Package: raylibr (via r-universe)

October 4, 2026

Title R Wrapper for 'Raylib'

Version 4.0.0.9001

Description This package wraps Raylib, a simple and easy-to-use library to enjoy videogames programming.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

LinkingTo Rcpp

Imports Rcpp, utils

URL https://github.com/jeroenjanssens/raylibr, https://jeroenjanssens.github.io/raylibr/

BugReports https://github.com/jeroenjanssens/raylibr/issues

Repository https://rogiersbart.r-universe.dev

Date/Publication 2024-05-31 08:12:59 UTC

RemoteUrl https://github.com/rogiersbart/raylibr

RemoteRef prefix-class-names

RemoteSha c5ee23dd8f1d6bca77cd5ea3508b8272dce3ff1c

Contents

as_color 10
begin_blend_mode 10
begin_drawing 11
begin_mode_2d 11
begin_mode_3d 12
begin_scissor_mode 13
begin_shader_mode 13
begin_texture_mode 14
bounding_box 15

camera_2d . . . 15
camera_3d . . . 16
change_directory . . . 17
check_collision_box_sphere . . . 18
check_collision_boxes . . . 19
check_collision_circle_rec . . . 19
check_collision_circles . . . 20
check_collision_lines . . . 21
check_collision_point_circle . . . 22
check_collision_point_line . . . 23
check_collision_point_rec . . . 24
check_collision_point_triangle . . . 25
check_collision_recs . . . 26
check_collision_spheres . . . 26
clear_background . . . 27
clear_directory_files . . . 28
clear_dropped_files . . . 28
clear_window_state . . . 29
close_audio_device . . . 29
close_window . . . 30
color . . . 30
color_alpha . . . 31
color_alpha_blend . . . 32
color_from_hsv . . . 33
color_from_normalized . . . 33
color_normalize . . . 34
color_to_hsv . . . 35
color_to_int . . . 36
directory_exists . . . 36
disable_cursor . . . 37
draw_billboard . . . 38
draw_bounding_box . . . 39
draw_circle . . . 40
draw_circle_3d . . . 41
draw_circle_gradient . . . 42
draw_circle_lines . . . 43
draw_circle_sector . . . 44
draw_circle_sector_lines . . . 46
draw_circle_v . . . 47
draw_cube . . . 48
draw_cube_texture . . . 49
draw_cube_texture_rec . . . 50
draw_cube_v . . . 52
draw_cube_wires . . . 53
draw_cube_wires_v . . . 54
draw_cylinder . . . 55
draw_cylinder_ex . . . 56
draw_cylinder_wires . . . 57

draw_cylinder_wires_ex . . . 58
draw_ellipse . . . 59
draw_ellipse_lines . . . 60
draw_fps . . . 61
draw_grid . . . 62
draw_line . . . 63
draw_line_3d . . . 64
draw_line_bezier . . . 65
draw_line_bezier_cubic . . . 66
draw_line_bezier_quad . . . 68
draw_line_ex . . . 69
draw_line_v . . . 70
draw_model . . . 71
draw_model_ex . . . 72
draw_model_wires . . . 74
draw_pixel . . . 75
draw_pixel_v . . . 76
draw_plane . . . 77
draw_point_3d . . . 78
draw_poly . . . 79
draw_poly_lines . . . 80
draw_poly_lines_ex . . . 81
draw_ray . . . 82
draw_rectangle . . . 83
draw_rectangle_gradient_ex . . . 84
draw_rectangle_gradient_h . . . 85
draw_rectangle_gradient_v . . . 86
draw_rectangle_lines . . . 88
draw_rectangle_lines_ex . . . 89
draw_rectangle_pro . . . 90
draw_rectangle_rec . . . 91
draw_rectangle_rounded . . . 92
draw_rectangle_rounded_lines . . . 94
draw_rectangle_v . . . 95
draw_ring . . . 96
draw_ring_lines . . . 97
draw_sphere . . . 99
draw_sphere_ex . . . 100
draw_sphere_wires . . . 101
draw_text . . . 102
draw_text_codepoint . . . 103
draw_text_ex . . . 104
draw_text_pro . . . 105
draw_texture . . . 107
draw_texture_ex . . . 108
draw_texture_npatch . . . 109
draw_texture_poly . . . 110
draw_texture_pro . . . 112

draw_texture_quad . . . 113
draw_texture_rec . . . 114
draw_texture_tiled . . . 115
draw_texture_v . . . 117
draw_triangle . . . 118
draw_triangle_3d . . . 119
draw_triangle_lines . . . 120
draw_triangle_strip_3d . . . 121
enable_cursor . . . 122
end_blend_mode . . . 123
end_drawing . . . 123
end_mode_2d . . . 124
end_mode_3d . . . 124
end_scissor_mode . . . 125
end_shader_mode . . . 125
end_texture_mode . . . 126
end_vr_stereo_mode . . . 126
export_image . . . 127
export_image_as_code . . . 128
export_wave . . . 129
export_wave_as_code . . . 129
fade . . . 130
file_exists . . . 131
gen_image_cellular . . . 131
gen_image_checked . . . 132
gen_image_color . . . 133
gen_image_gradient_h . . . 134
gen_image_gradient_radial . . . 135
gen_image_gradient_v . . . 136
gen_image_white_noise . . . 137
gen_texture_mipmaps . . . 138
get_camera_matrix . . . 139
get_camera_matrix_2d . . . 140
get_char_pressed . . . 140
get_collision_rec . . . 141
get_color . . . 142
get_current_monitor . . . 142
get_directory_path . . . 143
get_file_extension . . . 144
get_file_mod_time . . . 144
get_file_name . . . 145
get_file_name_without_ext . . . 146
get_font_default . . . 146
get_fps . . . 147
get_frame_time . . . 148
get_gamepad_axis_count . . . 148
get_gamepad_axis_movement . . . 149
get_gamepad_button_pressed . . . 150

get_gamepad_name . . . 150
get_gesture_detected . . . 151
get_gesture_drag_angle . . . 151
get_gesture_drag_vector . . . 152
get_gesture_hold_duration . . . 152
get_gesture_pinch_angle . . . 153
get_gesture_pinch_vector . . . 153
get_glyph_atlas_rec . . . 154
get_glyph_index . . . 155
get_glyph_info . . . 155
get_image_alpha_border . . . 156
get_image_color . . . 157
get_key_pressed . . . 158
get_model_bounding_box . . . 159
get_monitor_count . . . 159
get_monitor_height . . . 160
get_monitor_name . . . 161
get_monitor_physical_height . . . 161
get_monitor_physical_width . . . 162
get_monitor_position . . . 163
get_monitor_refresh_rate . . . 163
get_monitor_width . . . 164
get_mouse_delta . . . 165
get_mouse_position . . . 165
get_mouse_wheel_move . . . 166
get_mouse_x . . . 167
get_mouse_y . . . 167
get_music_time_length . . . 168
get_music_time_played . . . 169
get_pixel_data_size . . . 169
get_prev_directory_path . . . 170
get_random_value . . . 171
get_ray_collision_box . . . 171
get_ray_collision_mesh . . . 172
get_ray_collision_model . . . 173
get_ray_collision_quad . . . 174
get_ray_collision_sphere . . . 175
get_ray_collision_triangle . . . 176
get_screen_height . . . 177
get_screen_to_world_2d . . . 177
get_screen_width . . . 178
get_shader_location . . . 179
get_shader_location_attrib . . . 179
get_sounds_playing . . . 180
get_time . . . 181
get_touch_point_count . . . 181
get_touch_point_id . . . 182
get_touch_position . . . 182

get_touch_x . . . 183
get_touch_y . . . 184
get_window_position . . . 184
get_window_scale_dpi . . . 185
get_working_directory . . . 186
get_world_to_screen . . . 186
get_world_to_screen_2d . . . 187
get_world_to_screen_ex . . . 188
glyph_info . . . 189
hide_cursor . . . 190
image_alpha_clear . . . 190
image_alpha_crop . . . 191
image_alpha_mask . . . 192
image_alpha_premultiply . . . 193
image_clear_background . . . 194
image_color_brightness . . . 195
image_color_contrast . . . 196
image_color_grayscale . . . 197
image_color_invert . . . 198
image_color_replace . . . 199
image_color_tint . . . 200
image_copy . . . 201
image_crop . . . 202
image_dither . . . 203
image_draw . . . 204
image_draw_circle . . . 205
image_draw_circle_v . . . 206
image_draw_line . . . 208
image_draw_line_v . . . 209
image_draw_pixel . . . 211
image_draw_pixel_v . . . 212
image_draw_rectangle . . . 213
image_draw_rectangle_lines . . . 215
image_draw_rectangle_rec . . . 217
image_draw_rectangle_v . . . 218
image_draw_text . . . 219
image_draw_text_ex . . . 221
image_flip_horizontal . . . 222
image_flip_vertical . . . 223
image_format . . . 224
image_from_image . . . 225
image_mipmaps . . . 226
image_resize . . . 227
image_resize_canvas . . . 228
image_resize_nn . . . 229
image_rotate_ccw . . . 230
image_rotate_cw . . . 231
image_text . . . 232

image_text_ex . . . 233
image_to_pot . . . 234
init_audio_device . . . 235
init_window . . . 235
is_audio_device_ready . . . 236
is_audio_stream_playing . . . 237
is_audio_stream_processed . . . 238
is_cursor_hidden . . . 239
is_cursor_on_screen . . . 239
is_file_dropped . . . 240
is_file_extension . . . 240
is_gamepad_available . . . 241
is_gamepad_button_down . . . 242
is_gamepad_button_pressed . . . 242
is_gamepad_button_released . . . 243
is_gamepad_button_up . . . 244
is_gesture_detected . . . 245
is_key_down . . . 245
is_key_pressed . . . 246
is_key_released . . . 247
is_key_up . . . 247
is_mouse_button_down . . . 248
is_mouse_button_pressed . . . 249
is_mouse_button_released . . . 249
is_mouse_button_up . . . 250
is_music_stream_playing . . . 251
is_sound_playing . . . 252
is_window_focused . . . 252
is_window_fullscreen . . . 253
is_window_hidden . . . 254
is_window_maximized . . . 254
is_window_minimized . . . 255
is_window_ready . . . 256
is_window_resized . . . 256
is_window_state . . . 257
load_audio_stream . . . 258
load_file_text . . . 259
load_font . . . 259
load_font_ex . . . 260
load_font_from_image . . . 261
load_image . . . 262
load_image_from_screen . . . 263
load_image_from_texture . . . 264
load_image_raw . . . 265
load_model . . . 266
load_music_stream . . . 266
load_render_texture . . . 267
load_shader . . . 268

load_shader_from_memory . . . 269
load_sound . . . 269
load_sound_from_wave . . . 270
load_storage_value . . . 271
load_texture . . . 271
load_texture_from_image . . . 272
load_wave . . . 273
material_map . . . 274
maximize_window . . . 275
measure_text . . . 275
measure_text_ex . . . 276
minimize_window . . . 277
npatch_info . . . 277
pause_audio_stream . . . 278
pause_music_stream . . . 279
pause_sound . . . 280
play_audio_stream . . . 280
play_music_stream . . . 281
play_sound . . . 282
play_sound_multi . . . 283
ray . . . 283
ray_collision . . . 284
raylib_version . . . 285
rectangle . . . 285
render_texture . . . 286
restore_window . . . 287
resume_audio_stream . . . 287
resume_music_stream . . . 288
resume_sound . . . 289
save_storage_value . . . 289
seek_music_stream . . . 290
set_audio_stream_buffer_size_default . . . 291
set_audio_stream_pitch . . . 291
set_audio_stream_volume . . . 292
set_camera_alt_control . . . 293
set_camera_mode . . . 294
set_camera_move_controls . . . 294
set_camera_pan_control . . . 295
set_camera_smooth_zoom_control . . . 296
set_exit_key . . . 297
set_gamepad_mappings . . . 297
set_gestures_enabled . . . 298
set_master_volume . . . 298
set_mouse_cursor . . . 299
set_mouse_offset . . . 300
set_mouse_position . . . 300
set_mouse_scale . . . 301
set_music_pitch . . . 302

set_music_volume . . . 302
set_shader_value . . . 303
set_shader_value_matrix . . . 304
set_shader_value_texture . . . 304
set_shapes_texture . . . 305
set_sound_pitch . . . 306
set_sound_volume . . . 307
set_target_fps . . . 307
set_texture_filter . . . 308
set_texture_wrap . . . 309
set_trace_log_level . . . 309
set_window_icon . . . 310
set_window_min_size . . . 311
set_window_monitor . . . 311
set_window_position . . . 312
set_window_size . . . 313
set_window_state . . . 313
set_window_title . . . 314
show_cursor . . . 315
sound . . . 315
stop_audio_stream . . . 316
stop_music_stream . . . 317
stop_sound . . . 317
stop_sound_multi . . . 318
take_screenshot . . . 319
text_find_index . . . 319
text_insert . . . 320
text_replace . . . 321
text_to_integer . . . 321
text_to_lower . . . 322
text_to_pascal . . . 323
text_to_upper . . . 324
texture . . . 324
toggle_fullscreen . . . 325
transform . . . 326
unload_audio_stream . . . 326
unload_font . . . 327
unload_font_data . . . 328
unload_image . . . 328
unload_image_colors . . . 329
unload_image_palette . . . 330
unload_model . . . 331
unload_music_stream . . . 331
unload_render_texture . . . 332
unload_shader . . . 333
unload_sound . . . 333
unload_texture . . . 334
unload_wave . . . 335

update_camera 335
update_music_stream 336
vector_2_angle 337
vector_2_distance 337
vector_2_move_towards 338
vector_2_rotate 339
window_should_close 339

Index **341**

`as_color` *Coerce a string or a number to a color*

Description

Coerce a string or a number to a color

Usage

```
as_color(x)
```

Arguments

`x`	Object to be coerced. Can be a valid R color (see `colors()`) or a hexademical.

Value

A color

See Also

Other color functions: `color_alpha_blend()`, `color_alpha()`, `color_from_hsv()`, `color_from_normalized()`, `color_normalize()`, `color_to_hsv()`, `color_to_int()`, `color()`, `gen_image_color()`, `get_color()`, `get_image_color()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`

`begin_blend_mode` *Begin blend mode*

Description

Begin blending mode (alpha, additive, multiplied, subtract, custom).

Usage

```
begin_blend_mode(mode)
```

Arguments

mode	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void BeginBlendMode(int mode);
```

See Also

Other mode functions: begin_mode_2d(), begin_mode_3d(), begin_scissor_mode(), begin_shader_mode(), begin_texture_mode(), end_blend_mode(), end_mode_2d(), end_mode_3d(), end_scissor_mode(), end_shader_mode(), end_texture_mode(), end_vr_stereo_mode(), set_camera_mode()

---

begin_drawing *Begin drawing*

Description

Setup canvas (framebuffer) to start drawing.

Usage

```
begin_drawing()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void BeginDrawing(void);
```

---

begin_mode_2d *Begin mode 2d*

Description

Begin 2D mode with custom camera (2D).

Usage

```
begin_mode_2d(camera)
```

Arguments

camera	A camera_2d.

Note

This function has been auto-generated from the following Raylib function definition:

```
void BeginMode2D(Camera2D camera);
```

See Also

`camera_2d()` for creating camera_2d objects.

Other mode functions: `begin_blend_mode()`, `begin_mode_3d()`, `begin_scissor_mode()`, `begin_shader_mode()`, `begin_texture_mode()`, `end_blend_mode()`, `end_mode_2d()`, `end_mode_3d()`, `end_scissor_mode()`, `end_shader_mode()`, `end_texture_mode()`, `end_vr_stereo_mode()`, `set_camera_mode()`

---

begin_mode_3d *Begin mode 3d*

Description

Begin 3D mode with custom camera (3D).

Usage

```
begin_mode_3d(camera)
```

Arguments

camera	A camera_3d.

Note

This function has been auto-generated from the following Raylib function definition:

```
void BeginMode3D(Camera3D camera);
```

See Also

`camera_3d()` for creating camera_3d objects.

Other mode functions: `begin_blend_mode()`, `begin_mode_2d()`, `begin_scissor_mode()`, `begin_shader_mode()`, `begin_texture_mode()`, `end_blend_mode()`, `end_mode_2d()`, `end_mode_3d()`, `end_scissor_mode()`, `end_shader_mode()`, `end_texture_mode()`, `end_vr_stereo_mode()`, `set_camera_mode()`

begin_scissor_mode *Begin scissor mode*

Description

Begin scissor mode (define screen area for following drawing).

Usage

```
begin_scissor_mode(x, y, width, height)
```

Arguments

x	An integer.
y	An integer.
width	An integer.
height	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void BeginScissorMode(int x, int y, int width, int height);
```

See Also

Other mode functions: begin_blend_mode(), begin_mode_2d(), begin_mode_3d(), begin_shader_mode(), begin_texture_mode(), end_blend_mode(), end_mode_2d(), end_mode_3d(), end_scissor_mode(), end_shader_mode(), end_texture_mode(), end_vr_stereo_mode(), set_camera_mode()

begin_shader_mode *Begin shader mode*

Description

Begin custom shader drawing.

Usage

```
begin_shader_mode(shader)
```

Arguments

shader	A shader.

Note

This function has been auto-generated from the following Raylib function definition:

```
void BeginShaderMode(Shader shader);
```

See Also

shader() for creating shader objects.

Other shader functions: end_shader_mode(), get_shader_location_attrib(), get_shader_location(), load_shader_from_memory(), load_shader(), set_shader_value_matrix(), set_shader_value_texture(), set_shader_value(), unload_shader()

Other mode functions: begin_blend_mode(), begin_mode_2d(), begin_mode_3d(), begin_scissor_mode(), begin_texture_mode(), end_blend_mode(), end_mode_2d(), end_mode_3d(), end_scissor_mode(), end_shader_mode(), end_texture_mode(), end_vr_stereo_mode(), set_camera_mode()

---

begin_texture_mode *Begin texture mode*

Description

Begin drawing to render texture.

Usage

```
begin_texture_mode(target)
```

Arguments

target	A render_texture.

Note

This function has been auto-generated from the following Raylib function definition:

```
void BeginTextureMode(RenderTexture2D target);
```

See Also

Other texture functions: draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

Other mode functions: begin_blend_mode(), begin_mode_2d(), begin_mode_3d(), begin_scissor_mode(), begin_shader_mode(), end_blend_mode(), end_mode_2d(), end_mode_3d(), end_scissor_mode(), end_shader_mode(), end_texture_mode(), end_vr_stereo_mode(), set_camera_mode()

bounding_box *Bounding box*

Description

Create a new bounding_box object.

Usage

```
bounding_box(min, max)
```

Arguments

min	A numeric vector of length 3. Minimum vertex box-corner.
max	A numeric vector of length 3. Maximum vertex box-corner.

Value

A bounding_box

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct BoundingBox {
    Vector3 min;            // Minimum vertex box-corner
    Vector3 max;            // Maximum vertex box-corner
} BoundingBox;
```

camera_2d *Camera 2d*

Description

Create a new camera_2d object.

Usage

```
camera_2d(offset, target, rotation, zoom = 1)
```

Arguments

offset	A numeric vector of length 2. Camera offset (displacement from target).
target	A numeric vector of length 2. Camera target (rotation and zoom origin).
rotation	A number. Camera rotation in degrees.
zoom	A number. Camera zoom (scaling). Default: 1.

Value

A camera_2d

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct Camera2D {
    Vector2 offset;         // Camera offset (displacement from target)
    Vector2 target;         // Camera target (rotation and zoom origin)
    float rotation;         // Camera rotation in degrees
    float zoom;             // Camera zoom (scaling), should be 1.0f by default
} Camera2D;
```

---

camera_3d *Camera 3d*

Description

Create a new camera_3d object.

Usage

```
camera_3d(
  position,
  target = c(0, 0, 0),
  up = c(0, 1, 0),
  fovy = 70,
  projection = 0
)
```

Arguments

`position`	A numeric vector of length 3. Camera position.
`target`	A numeric vector of length 3. Camera target it looks-at. Default: `c(0, 0, 0)`.
`up`	A numeric vector of length 3. Camera up vector (rotation over its axis). Default: `c(0, 1, 0)`.
`fovy`	A number. Camera field-of-view apperture in Y (degrees) in perspective, used as near plane width in orthographic. Default: `70`.
`projection`	An integer. Camera projection: either `camera_projection$perspective` (0) or `camera_projection$orthographic` (1). Default: `0`.

Value

A camera_3d

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct Camera3D {
    Vector3 position;       // Camera position
    Vector3 target;         // Camera target it looks-at
    Vector3 up;             // Camera up vector (rotation over its axis)
   float fovy;         // Camera field-of-view apperture in Y (degrees) in perspective, used as near pla
   int projection;      // Camera projection: CAMERA_PERSPECTIVE or CAMERA_ORTHOGRAPHIC
} Camera3D;
```

---

change_directory *Change directory*

Description

Change working directory, return true on success.

Usage

```
change_directory(dir)
```

Arguments

dir	A string.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool ChangeDirectory(const char *dir);
```

See Also

Other directory functions: clear_directory_files(), directory_exists(), get_directory_path(), get_prev_directory_path(), get_working_directory()

`check_collision_box_sphere` *Check collision box sphere*

Description

Check collision between box and sphere.

Usage

```
check_collision_box_sphere(box, center, radius)
```

Arguments

`box`	A bounding_box.
`center`	A numeric vector of length 3.
`radius`	A number.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool CheckCollisionBoxSphere(BoundingBox box, Vector3 center, float radius);
```

See Also

Other collision functions: `check_collision_boxes()`, `check_collision_circle_rec()`, `check_collision_circles()`, `check_collision_lines()`, `check_collision_point_circle()`, `check_collision_point_line()`, `check_collision_point_rec()`, `check_collision_point_triangle()`, `check_collision_recs()`, `check_collision_spheres()`, `get_collision_rec()`, `get_ray_collision_box()`, `get_ray_collision_mesh()`, `get_ray_collision_model()`, `get_ray_collision_quad()`, `get_ray_collision_sphere()`, `get_ray_collision_tria`

check_collision_boxes *Check collision boxes*

Description

Check collision between two bounding boxes.

Usage

```
check_collision_boxes(box_1, box_2)
```

Arguments

box_1	A bounding_box.
box_2	A bounding_box.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool CheckCollisionBoxes(BoundingBox box1, BoundingBox box2);
```

See Also

Other collision functions: check_collision_box_sphere(), check_collision_circle_rec(), check_collision_circles(), check_collision_lines(), check_collision_point_circle(), check_collision_point_line(), check_collision_point_rec(), check_collision_point_triangle(), check_collision_recs(), check_collision_spheres(), get_collision_rec(), get_ray_collision_box(), get_ray_collision_mesh(), get_ray_collision_model(), get_ray_collision_quad(), get_ray_collision_sphere get_ray_collision_triangle()

check_collision_circle_rec *Check collision circle rec*

Description

Check collision between circle and rectangle.

Usage

```
check_collision_circle_rec(center, radius, rec)
```

Arguments

center	A numeric vector of length 2.
radius	A number.
rec	A rectangle.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool CheckCollisionCircleRec(Vector2 center, float radius, Rectangle rec);
```

See Also

rectangle() for creating rectangle objects.

Other collision functions: check_collision_box_sphere(), check_collision_boxes(), check_collision_circles(), check_collision_lines(), check_collision_point_circle(), check_collision_point_line(), check_collision_point_rec(), check_collision_point_triangle(), check_collision_recs(), check_collision_spheres(), get_collision_rec(), get_ray_collision_box(), get_ray_collision_mesh(), get_ray_collision_model(), get_ray_collision_quad(), get_ray_collision_sphere(), get_ray_collision_tria

Other circle functions: check_collision_point_circle(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), image_draw_circle_v(), image_draw_circle()

---

check_collision_circles

Check collision circles

Description

Check collision between two circles.

Usage

```
check_collision_circles(center_1, radius_1, center_2, radius_2)
```

Arguments

center_1	A numeric vector of length 2.
radius_1	A number.
center_2	A numeric vector of length 2.
radius_2	A number.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool CheckCollisionCircles(Vector2 center1, float radius1, Vector2 center2, float radius2);
```

See Also

Other collision functions: check_collision_box_sphere(), check_collision_boxes(), check_collision_circle_rec
check_collision_lines(), check_collision_point_circle(), check_collision_point_line(),
check_collision_point_rec(), check_collision_point_triangle(), check_collision_recs(),
check_collision_spheres(), get_collision_rec(), get_ray_collision_box(), get_ray_collision_mesh(),
get_ray_collision_model(), get_ray_collision_quad(), get_ray_collision_sphere(), get_ray_collision_tria

---

check_collision_lines *Check collision lines*

Description

Check the collision between two lines defined by two points each, returns collision point by reference.

Usage

```
check_collision_lines(
  start_pos_1,
  end_pos_1,
  start_pos_2,
  end_pos_2,
  collision_point
)
```

Arguments

start_pos_1	A numeric vector of length 2.
end_pos_1	A numeric vector of length 2.
start_pos_2	A numeric vector of length 2.
end_pos_2	A numeric vector of length 2.
collision_point	A numeric vector of length 2.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool CheckCollisionLines(Vector2 startPos1, Vector2 endPos1, Vector2 startPos2, Vector2 endPos2, Vecto
```

See Also

Other collision functions: check_collision_box_sphere(), check_collision_boxes(), check_collision_circle_rec
check_collision_circles(), check_collision_point_circle(), check_collision_point_line(),
check_collision_point_rec(), check_collision_point_triangle(), check_collision_recs(),
check_collision_spheres(), get_collision_rec(), get_ray_collision_box(), get_ray_collision_mesh(),
get_ray_collision_model(), get_ray_collision_quad(), get_ray_collision_sphere(), get_ray_collision_tria

---

check_collision_point_circle

Check collision point circle

Description

Check if point is inside circle.

Usage

```
check_collision_point_circle(point, center, radius)
```

Arguments

point	A numeric vector of length 2.
center	A numeric vector of length 2.
radius	A number.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool CheckCollisionPointCircle(Vector2 point, Vector2 center, float radius);
```

See Also

Other collision functions: check_collision_box_sphere(), check_collision_boxes(), check_collision_circle_rec
check_collision_circles(), check_collision_lines(), check_collision_point_line(),
check_collision_point_rec(), check_collision_point_triangle(), check_collision_recs(),
check_collision_spheres(), get_collision_rec(), get_ray_collision_box(), get_ray_collision_mesh(),
get_ray_collision_model(), get_ray_collision_quad(), get_ray_collision_sphere(), get_ray_collision_tria

Other circle functions: check_collision_circle_rec(), draw_circle_3d(), draw_circle_gradient(),
draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(),
draw_circle(), image_draw_circle_v(), image_draw_circle()

---

check_collision_point_line

Check collision point line

Description

Check if point belongs to line created between two points p1 and p2 with defined margin in pixels threshold.

Usage

```
check_collision_point_line(point, p_1, p_2, threshold)
```

Arguments

point	A numeric vector of length 2.
p_1	A numeric vector of length 2.
p_2	A numeric vector of length 2.
threshold	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool CheckCollisionPointLine(Vector2 point, Vector2 p1, Vector2 p2, int threshold);
```

See Also

Other collision functions: check_collision_box_sphere(), check_collision_boxes(), check_collision_circle_rec
check_collision_circles(), check_collision_lines(), check_collision_point_circle(),
check_collision_point_rec(), check_collision_point_triangle(), check_collision_recs(),
check_collision_spheres(), get_collision_rec(), get_ray_collision_box(), get_ray_collision_mesh(),
get_ray_collision_model(), get_ray_collision_quad(), get_ray_collision_sphere(), get_ray_collision_tria

Other line functions: draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(),
draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), image_draw_line_v(),
image_draw_line()

`check_collision_point_rec` *Check collision point rec*

Description

Check if point is inside rectangle.

Usage

```
check_collision_point_rec(point, rec)
```

Arguments

`point`	A numeric vector of length 2.
`rec`	A rectangle.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool CheckCollisionPointRec(Vector2 point, Rectangle rec);
```

See Also

`rectangle()` for creating rectangle objects.

Other collision functions: check_collision_box_sphere(), check_collision_boxes(), check_collision_circle_rec
check_collision_circles(), check_collision_lines(), check_collision_point_circle(),
check_collision_point_line(), check_collision_point_triangle(), check_collision_recs(),
check_collision_spheres(), get_collision_rec(), get_ray_collision_box(), get_ray_collision_mesh(),
get_ray_collision_model(), get_ray_collision_quad(), get_ray_collision_sphere(), get_ray_collision_tria

`check_collision_point_triangle` *Check collision point triangle*

Description

Check if point is inside a triangle.

Usage

```
check_collision_point_triangle(point, p_1, p_2, p_3)
```

Arguments

`point`	A numeric vector of length 2.
`p_1`	A numeric vector of length 2.
`p_2`	A numeric vector of length 2.
`p_3`	A numeric vector of length 2.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool CheckCollisionPointTriangle(Vector2 point, Vector2 p1, Vector2 p2, Vector2 p3);
```

See Also

Other collision functions: `check_collision_box_sphere()`, `check_collision_boxes()`, `check_collision_circle_rec`
`check_collision_circles()`, `check_collision_lines()`, `check_collision_point_circle()`,
`check_collision_point_line()`, `check_collision_point_rec()`, `check_collision_recs()`,
`check_collision_spheres()`, `get_collision_rec()`, `get_ray_collision_box()`, `get_ray_collision_mesh()`,
`get_ray_collision_model()`, `get_ray_collision_quad()`, `get_ray_collision_sphere()`, `get_ray_collision_tria`

check_collision_recs *Check collision recs*

Description

Check collision between two rectangles.

Usage

```
check_collision_recs(rec_1, rec_2)
```

Arguments

rec_1	A rectangle.
rec_2	A rectangle.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool CheckCollisionRecs(Rectangle rec1, Rectangle rec2);
```

See Also

rectangle() for creating rectangle objects.

Other collision functions: check_collision_box_sphere(), check_collision_boxes(), check_collision_circle_rec
check_collision_circles(), check_collision_lines(), check_collision_point_circle(),
check_collision_point_line(), check_collision_point_rec(), check_collision_point_triangle(),
check_collision_spheres(), get_collision_rec(), get_ray_collision_box(), get_ray_collision_mesh(),
get_ray_collision_model(), get_ray_collision_quad(), get_ray_collision_sphere(), get_ray_collision_tria

check_collision_spheres *Check collision spheres*

Description

Check collision between two spheres.

Usage

```
check_collision_spheres(center_1, radius_1, center_2, radius_2)
```

Arguments

center_1	A numeric vector of length 3.
radius_1	A number.
center_2	A numeric vector of length 3.
radius_2	A number.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool CheckCollisionSpheres(Vector3 center1, float radius1, Vector3 center2, float radius2);
```

See Also

Other collision functions: check_collision_box_sphere(), check_collision_boxes(), check_collision_circle_rec
check_collision_circles(), check_collision_lines(), check_collision_point_circle(),
check_collision_point_line(), check_collision_point_rec(), check_collision_point_triangle(),
check_collision_recs(), get_collision_rec(), get_ray_collision_box(), get_ray_collision_mesh(),
get_ray_collision_model(), get_ray_collision_quad(), get_ray_collision_sphere(), get_ray_collision_tria

---

clear_background *Clear background*

Description

Set background color (framebuffer clear color).

Usage

```
clear_background(color = "white")
```

Arguments

color	A color. Default: "white".

Note

This function has been auto-generated from the following Raylib function definition:

```
void ClearBackground(Color color);
```

See Also

color() for creating color objects.

clear_directory_files *Clear directory files*

Description

Clear directory files paths buffers (free memory).

Usage

```
clear_directory_files()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void ClearDirectoryFiles(void);
```

See Also

Other directory functions: `change_directory()`, `directory_exists()`, `get_directory_path()`, `get_prev_directory_path()`, `get_working_directory()`

clear_dropped_files *Clear dropped files*

Description

Clear dropped files paths buffer (free memory).

Usage

```
clear_dropped_files()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void ClearDroppedFiles(void);
```

clear_window_state *Clear window state*

Description

Clear window configuration state flags.

Usage

```
clear_window_state(flags)
```

Arguments

flags	A non-negative integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ClearWindowState(unsigned int flags);
```

See Also

Other window functions: close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

close_audio_device *Close audio device*

Description

Close the audio device and context.

Usage

```
close_audio_device()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void CloseAudioDevice(void);
```

See Also

Other audio functions: init_audio_device(), is_audio_device_ready(), is_audio_stream_playing(), is_audio_stream_processed(), load_audio_stream(), pause_audio_stream(), play_audio_stream(), resume_audio_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), unload_audio_stream()

---

close_window *Close window*

Description

Close window and unload OpenGL context.

Usage

```
close_window()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void CloseWindow(void);
```

See Also

Other window functions: clear_window_state(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

---

color *Color*

Description

Create a new color object.

Usage

```
color(r, g, b, a)
```

Arguments

r	A non-negative integer. Color red value.
g	A non-negative integer. Color green value.
b	A non-negative integer. Color blue value.
a	A non-negative integer. Color alpha value.

Value

A color

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct Color {
    unsigned char r;        // Color red value
    unsigned char g;        // Color green value
    unsigned char b;        // Color blue value
    unsigned char a;        // Color alpha value
} Color;
```

See Also

Other color functions: as_color(), color_alpha_blend(), color_alpha(), color_from_hsv(), color_from_normalized(), color_normalize(), color_to_hsv(), color_to_int(), gen_image_color(), get_color(), get_image_color(), image_color_brightness(), image_color_contrast(), image_color_grayscale(image_color_invert(), image_color_replace(), image_color_tint()

---

color_alpha *Color alpha*

Description

Get color with alpha applied, alpha goes from 0.0f to 1.0f.

Usage

```
color_alpha(color, alpha)
```

Arguments

color	A color.
alpha	A number.

Value

A color

Note

This function has been auto-generated from the following Raylib function definition:

```
Color ColorAlpha(Color color, float alpha);
```

See Also

`color()` for creating color objects.

Other color functions: `as_color()`, `color_alpha_blend()`, `color_from_hsv()`, `color_from_normalized()`, `color_normalize()`, `color_to_hsv()`, `color_to_int()`, `color()`, `gen_image_color()`, `get_color()`, `get_image_color()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`

---

color_alpha_blend *Color alpha blend*

Description

Get src alpha-blended into dst color with tint.

Usage

```
color_alpha_blend(dst, src, tint)
```

Arguments

`dst`	A color.
`src`	A color.
`tint`	A color.

Value

A color

Note

This function has been auto-generated from the following Raylib function definition:

```
Color ColorAlphaBlend(Color dst, Color src, Color tint);
```

See Also

`color()` for creating color objects.

Other color functions: `as_color()`, `color_alpha()`, `color_from_hsv()`, `color_from_normalized()`, `color_normalize()`, `color_to_hsv()`, `color_to_int()`, `color()`, `gen_image_color()`, `get_color()`, `get_image_color()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`

color_from_hsv *Color from hsv*

Description

Get a Color from HSV values, hue 0..360, saturation/value 0..1.

Usage

```
color_from_hsv(hue, saturation, value)
```

Arguments

hue	A number.
saturation	A number.
value	A number.

Value

A color

Note

This function has been auto-generated from the following Raylib function definition:

```
Color ColorFromHSV(float hue, float saturation, float value);
```

See Also

Other color functions: as_color(), color_alpha_blend(), color_alpha(), color_from_normalized(), color_normalize(), color_to_hsv(), color_to_int(), color(), gen_image_color(), get_color(), get_image_color(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint()

color_from_normalized *Color from normalized*

Description

Get Color from normalized values 0..1.

Usage

```
color_from_normalized(normalized)
```

Arguments

normalized	A numeric vector of length 4.

Value

A color

Note

This function has been auto-generated from the following Raylib function definition:

```
Color ColorFromNormalized(Vector4 normalized);
```

See Also

Other color functions: as_color(), color_alpha_blend(), color_alpha(), color_from_hsv(), color_normalize(), color_to_hsv(), color_to_int(), color(), gen_image_color(), get_color(), get_image_color(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint()

---

color_normalize *Color normalize*

Description

Get Color normalized as float 0..1.

Usage

```
color_normalize(color)
```

Arguments

color	A color.

Value

A numeric vector of length 4

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector4 ColorNormalize(Color color);
```

See Also

color() for creating color objects.

Other color functions: as_color(), color_alpha_blend(), color_alpha(), color_from_hsv(), color_from_normalized(), color_to_hsv(), color_to_int(), color(), gen_image_color(), get_color(), get_image_color(), image_color_brightness(), image_color_contrast(), image_color_grayscale(image_color_invert(), image_color_replace(), image_color_tint()

`color_to_hsv` *Color to hsv*

Description

Get HSV values for a Color, hue 0..360, saturation/value 0..1.

Usage

```
color_to_hsv(color)
```

Arguments

color	A color.

Value

A numeric vector of length 3

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector3 ColorToHSV(Color color);
```

See Also

color() for creating color objects.

Other color functions: as_color(), color_alpha_blend(), color_alpha(), color_from_hsv(), color_from_normalized(), color_normalize(), color_to_int(), color(), gen_image_color(), get_color(), get_image_color(), image_color_brightness(), image_color_contrast(), image_color_grayscale(image_color_invert(), image_color_replace(), image_color_tint()

color_to_int *Color to int*

Description

Get hexadecimal value for a Color.

Usage

```
color_to_int(color)
```

Arguments

color	A color.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int ColorToInt(Color color);
```

See Also

color() for creating color objects.

Other color functions: as_color(), color_alpha_blend(), color_alpha(), color_from_hsv(), color_from_normalized(), color_normalize(), color_to_hsv(), color(), gen_image_color(), get_color(), get_image_color(), image_color_brightness(), image_color_contrast(), image_color_grayscale(image_color_invert(), image_color_replace(), image_color_tint()

directory_exists *Directory exists*

Description

Check if a directory path exists.

Usage

```
directory_exists(dir_path)
```

Arguments

dir_path	A string.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool DirectoryExists(const char *dirPath);
```

See Also

Other directory functions: change_directory(), clear_directory_files(), get_directory_path(), get_prev_directory_path(), get_working_directory()

---

disable_cursor *Disable cursor*

Description

Disables cursor (lock cursor).

Usage

```
disable_cursor()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void DisableCursor(void);
```

See Also

Other cursor functions: enable_cursor(), hide_cursor(), is_cursor_hidden(), is_cursor_on_screen(), set_mouse_cursor(), show_cursor()

`draw_billboard` *Draw billboard*

Description

Draw a billboard texture.

Usage

```
draw_billboard(camera, texture, position, size, tint)
```

Arguments

`camera`	A camera_3d or a list of camera_3ds.
`texture`	A texture or a list of textures.
`position`	A numeric vector of length 3 or a numeric matrix of width 3.
`size`	A number or a vector of numbers.
`tint`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawBillboard(Camera camera, Texture2D texture, Vector3 position, float size, Color tint);
```

See Also

`camera_3d()` for creating camera_3d objects.

`texture()` for creating texture objects.

`color()` for creating color objects.

Other draw functions: `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_v()`, `draw_cube_wires_v()`, `draw_cube_wires()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse_lines()`, `draw_ellipse()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`, `draw_point_3d()`, `draw_poly_lines_ex()`, `draw_poly_lines()`, `draw_poly()`, `draw_ray()`, `draw_rectangle_gradient_ex()`, `draw_rectangle_gradient_h()`, `draw_rectangle_gradient_v()`, `draw_rectangle_lines_ex()`, `draw_rectangle_lines()`, `draw_rectangle_pro()`, `draw_rectangle_rec()`, `draw_rectangle_rounded_lines()`, `draw_rectangle_rounded()`, `draw_rectangle_v()`, `draw_rectangle()`, `draw_ring_lines()`, `draw_ring()`, `draw_sphere_ex()`, `draw_sphere_wires()`, `draw_sphere()`, `draw_text_codepoint()`, `draw_text_ex()`, `draw_text_pro()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`,

draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_bounding_box` *Draw bounding box*

Description

Draw bounding box (wires).

Usage

```
draw_bounding_box(box, color)
```

Arguments

box	A bounding_box or a list of bounding_boxs.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawBoundingBox(BoundingBox box, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(),

draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_circle` *Draw circle*

Description

Draw a color-filled circle.

Usage

```
draw_circle(center_x, center_y, radius, color)
```

Arguments

`center_x`	An integer or a vector of integers.
`center_y`	An integer or a vector of integers.
`radius`	A number or a vector of numbers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCircle(int centerX, int centerY, float radius, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(),

draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other circle functions: check_collision_circle_rec(), check_collision_point_circle(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), image_draw_circle_v(), image_draw_circle()

`draw_circle_3d` *Draw circle 3d*

Description

Draw a circle in 3D world space.

Usage

```
draw_circle_3d(center, radius, rotation_axis, rotation_angle, color)
```

Arguments

`center`	A numeric vector of length 3 or a numeric matrix of width 3.
`radius`	A number or a vector of numbers.
`rotation_axis`	A numeric vector of length 3 or a numeric matrix of width 3.
`rotation_angle`	A number or a vector of numbers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCircle3D(Vector3 center, float radius, Vector3 rotationAxis, float rotationAngle, Color color)
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(),

draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other circle functions: check_collision_circle_rec(), check_collision_point_circle(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), image_draw_circle_v(), image_draw_circle()

`draw_circle_gradient` *Draw circle gradient*

Description

Draw a gradient-filled circle.

Usage

```
draw_circle_gradient(center_x, center_y, radius, color_1, color_2)
```

Arguments

`center_x`	An integer or a vector of integers.
`center_y`	An integer or a vector of integers.
`radius`	A number or a vector of numbers.
`color_1`	A color or a list of colors.
`color_2`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCircleGradient(int centerX, int centerY, float radius, Color color1, Color color2);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other circle functions: check_collision_circle_rec(), check_collision_point_circle(), draw_circle_3d(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), image_draw_circle_v(), image_draw_circle()

---

draw_circle_lines *Draw circle lines*

Description

Draw circle outline.

Usage

```
draw_circle_lines(center_x, center_y, radius, color)
```

Arguments

center_x	An integer or a vector of integers.
center_y	An integer or a vector of integers.
radius	A number or a vector of numbers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCircleLines(int centerX, int centerY, float radius, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other circle functions: check_collision_circle_rec(), check_collision_point_circle(), draw_circle_3d(), draw_circle_gradient(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), image_draw_circle_v(), image_draw_circle()

---

draw_circle_sector *Draw circle sector*

Description

Draw a piece of a circle.

Usage

```
draw_circle_sector(center, radius, start_angle, end_angle, segments, color)
```

Arguments

center	A numeric vector of length 2 or a numeric matrix of width 2.
radius	A number or a vector of numbers.
start_angle	A number or a vector of numbers.
end_angle	A number or a vector of numbers.
segments	An integer or a vector of integers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCircleSector(Vector2 center, float radius, float startAngle, float endAngle, int segments, Col
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_v(), draw_circle(), draw_cube_texture_rec() draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other circle functions: check_collision_circle_rec(), check_collision_point_circle(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_v(), draw_circle(), image_draw_circle_v(), image_draw_circle()

`draw_circle_sector_lines` *Draw circle sector lines*

Description

Draw circle sector outline.

Usage

```
draw_circle_sector_lines(
  center,
  radius,
  start_angle,
  end_angle,
  segments,
  color
)
```

Arguments

`center`	A numeric vector of length 2 or a numeric matrix of width 2.
`radius`	A number or a vector of numbers.
`start_angle`	A number or a vector of numbers.
`end_angle`	A number or a vector of numbers.
`segments`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCircleSectorLines(Vector2 center, float radius, float startAngle, float endAngle, int segments
```

See Also

`color()` for creating color objects.

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_v()`, `draw_cube_wires_v()`, `draw_cube_wires()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse_lines()`, `draw_ellipse()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`,

draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other circle functions: check_collision_circle_rec(), check_collision_point_circle(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), image_draw_circle_v(), image_draw_circle()

`draw_circle_v` *Draw circle v*

Description

Draw a color-filled circle (Vector version).

Usage

```
draw_circle_v(center, radius, color)
```

Arguments

`center`	A numeric vector of length 2 or a numeric matrix of width 2.
`radius`	A number or a vector of numbers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCircleV(Vector2 center, float radius, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(),

draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other circle functions: check_collision_circle_rec(), check_collision_point_circle(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle(), image_draw_circle_v(), image_draw_circle()

`draw_cube` *Draw cube*

Description

Draw cube.

Usage

```
draw_cube(position, width, height, length, color)
```

Arguments

`position`	A numeric vector of length 3 or a numeric matrix of width 3.
`width`	A number or a vector of numbers.
`height`	A number or a vector of numbers.
`length`	A number or a vector of numbers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCube(Vector3 position, float width, float height, float length, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

---

draw_cube_texture *Draw cube texture*

Description

Draw cube textured.

Usage

```
draw_cube_texture(texture, position, width, height, length, color)
```

Arguments

texture	A texture or a list of textures.
position	A numeric vector of length 3 or a numeric matrix of width 3.
width	A number or a vector of numbers.
height	A number or a vector of numbers.
length	A number or a vector of numbers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCubeTexture(Texture2D texture, Vector3 position, float width, float height, float length, Colo
```

See Also

texture() for creating texture objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

---

draw_cube_texture_rec *Draw cube texture rec*

Description

Draw cube with a region of a texture.

Usage

```
draw_cube_texture_rec(texture, source, position, width, height, length, color)
```

Arguments

texture	A texture or a list of textures.
source	A rectangle or a list of rectangles.
position	A numeric vector of length 3 or a numeric matrix of width 3.
width	A number or a vector of numbers.
height	A number or a vector of numbers.
length	A number or a vector of numbers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCubeTextureRec(Texture2D texture, Rectangle source, Vector3 position, float width, float heigh
```

See Also

texture() for creating texture objects.

rectangle() for creating rectangle objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other texture functions: begin_texture_mode(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

`draw_cube_v` *Draw cube v*

Description

Draw cube (Vector version).

Usage

```
draw_cube_v(position, size, color)
```

Arguments

`position`	A numeric vector of length 3 or a numeric matrix of width 3.
`size`	A numeric vector of length 3 or a numeric matrix of width 3.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCubeV(Vector3 position, Vector3 size, Color color);
```

See Also

`color()` for creating color objects.

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_wires_v()`, `draw_cube_wires()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse_lines()`, `draw_ellipse()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`, `draw_point_3d()`, `draw_poly_lines_ex()`, `draw_poly_lines()`, `draw_poly()`, `draw_ray()`, `draw_rectangle_gradient_ex()`, `draw_rectangle_gradient_h()`, `draw_rectangle_gradient_v()`, `draw_rectangle_lines_ex()`, `draw_rectangle_lines()`, `draw_rectangle_pro()`, `draw_rectangle_rec()`, `draw_rectangle_rounded_lines()`, `draw_rectangle_rounded()`, `draw_rectangle_v()`, `draw_rectangle()`, `draw_ring_lines()`, `draw_ring()`, `draw_sphere_ex()`, `draw_sphere_wires()`, `draw_sphere()`, `draw_text_codepoint()`, `draw_text_ex()`, `draw_text_pro()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`, `draw_texture_rec()`, `draw_texture_tiled()`, `draw_texture_v()`, `draw_texture()`, `draw_text()`, `draw_triangle_3d()`, `draw_triangle_lines()`, `draw_triangle_strip_3d()`, `draw_triangle()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_l`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`

`draw_cube_wires` *Draw cube wires*

Description

Draw cube wires.

Usage

```
draw_cube_wires(position, width, height, length, color)
```

Arguments

`position`	A numeric vector of length 3 or a numeric matrix of width 3.
`width`	A number or a vector of numbers.
`height`	A number or a vector of numbers.
`length`	A number or a vector of numbers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCubeWires(Vector3 position, float width, float height, float length, Color color);
```

See Also

`color()` for creating color objects.

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_v()`, `draw_cube_wires_v()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse_lines()`, `draw_ellipse()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`, `draw_point_3d()`, `draw_poly_lines_ex()`, `draw_poly_lines()`, `draw_poly()`, `draw_ray()`, `draw_rectangle_gradient_ex()`, `draw_rectangle_gradient_h()`, `draw_rectangle_gradient_v()`, `draw_rectangle_lines_ex()`, `draw_rectangle_lines()`, `draw_rectangle_pro()`, `draw_rectangle_rec()`, `draw_rectangle_rounded_lines()`, `draw_rectangle_rounded()`, `draw_rectangle_v()`, `draw_rectangle()`, `draw_ring_lines()`, `draw_ring()`, `draw_sphere_ex()`, `draw_sphere_wires()`, `draw_sphere()`, `draw_text_codepoint()`, `draw_text_ex()`, `draw_text_pro()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`, `draw_texture_rec()`, `draw_texture_tiled()`, `draw_texture_v()`, `draw_texture()`, `draw_text()`, `draw_triangle_3d()`, `draw_triangle_lines()`, `draw_triangle_strip_3d()`, `draw_triangle()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_l`

image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

---

draw_cube_wires_v *Draw cube wires v*

Description

Draw cube wires (Vector version).

Usage

```
draw_cube_wires_v(position, size, color)
```

Arguments

position	A numeric vector of length 3 or a numeric matrix of width 3.
size	A numeric vector of length 3 or a numeric matrix of width 3.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCubeWiresV(Vector3 position, Vector3 size, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l

image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_cylinder` *Draw cylinder*

Description

Draw a cylinder/cone.

Usage

```
draw_cylinder(position, radius_top, radius_bottom, height, slices, color)
```

Arguments

`position`	A numeric vector of length 3 or a numeric matrix of width 3.
`radius_top`	A number or a vector of numbers.
`radius_bottom`	A number or a vector of numbers.
`height`	A number or a vector of numbers.
`slices`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCylinder(Vector3 position, float radiusTop, float radiusBottom, float height, int slices, Colo
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(),

draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_cylinder_ex` *Draw cylinder ex*

Description

Draw a cylinder with base at startPos and top at endPos.

Usage

```
draw_cylinder_ex(start_pos, end_pos, start_radius, end_radius, sides, color)
```

Arguments

`start_pos`	A numeric vector of length 3 or a numeric matrix of width 3.
`end_pos`	A numeric vector of length 3 or a numeric matrix of width 3.
`start_radius`	A number or a vector of numbers.
`end_radius`	A number or a vector of numbers.
`sides`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCylinderEx(Vector3 startPos, Vector3 endPos, float startRadius, float endRadius, int sides, Co
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(),

draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_cylinder_wires` *Draw cylinder wires*

Description

Draw a cylinder/cone wires.

Usage

```
draw_cylinder_wires(position, radius_top, radius_bottom, height, slices, color)
```

Arguments

`position`	A numeric vector of length 3 or a numeric matrix of width 3.
`radius_top`	A number or a vector of numbers.
`radius_bottom`	A number or a vector of numbers.
`height`	A number or a vector of numbers.
`slices`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCylinderWires(Vector3 position, float radiusTop, float radiusBottom, float height, int slices,
```

See Also

`color()` for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(),

draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_cylinder_wires_ex` *Draw cylinder wires ex*

Description

Draw a cylinder wires with base at startPos and top at endPos.

Usage

```
draw_cylinder_wires_ex(
  start_pos,
  end_pos,
  start_radius,
  end_radius,
  sides,
  color
)
```

Arguments

`start_pos`	A numeric vector of length 3 or a numeric matrix of width 3.
`end_pos`	A numeric vector of length 3 or a numeric matrix of width 3.
`start_radius`	A number or a vector of numbers.
`end_radius`	A number or a vector of numbers.
`sides`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawCylinderWiresEx(Vector3 startPos, Vector3 endPos, float startRadius, float endRadius, int side
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

---

draw_ellipse *Draw ellipse*

Description

Draw ellipse.

Usage

```
draw_ellipse(center_x, center_y, radius_h, radius_v, color)
```

Arguments

center_x	An integer or a vector of integers.
center_y	An integer or a vector of integers.
radius_h	A number or a vector of numbers.
radius_v	A number or a vector of numbers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawEllipse(int centerX, int centerY, float radiusH, float radiusV, Color color);
```

See Also

`color()` for creating color objects.

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_v()`, `draw_cube_wires_v()`, `draw_cube_wires()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse_lines()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`, `draw_point_3d()`, `draw_poly_lines_ex()`, `draw_poly_lines()`, `draw_poly()`, `draw_ray()`, `draw_rectangle_gradient_ex()`, `draw_rectangle_gradient_h()`, `draw_rectangle_gradient_v()`, `draw_rectangle_lines_ex()`, `draw_rectangle_lines()`, `draw_rectangle_pro()`, `draw_rectangle_rec()`, `draw_rectangle_rounded_lines()`, `draw_rectangle_rounded()`, `draw_rectangle_v()`, `draw_rectangle()`, `draw_ring_lines()`, `draw_ring()`, `draw_sphere_ex()`, `draw_sphere_wires()`, `draw_sphere()`, `draw_text_codepoint()`, `draw_text_ex()`, `draw_text_pro()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`, `draw_texture_rec()`, `draw_texture_tiled()`, `draw_texture_v()`, `draw_texture()`, `draw_text()`, `draw_triangle_3d()`, `draw_triangle_lines()`, `draw_triangle_strip_3d()`, `draw_triangle()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_l`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`

---

draw_ellipse_lines *Draw ellipse lines*

Description

Draw ellipse outline.

Usage

```
draw_ellipse_lines(center_x, center_y, radius_h, radius_v, color)
```

Arguments

center_x	An integer or a vector of integers.
center_y	An integer or a vector of integers.
radius_h	A number or a vector of numbers.
radius_v	A number or a vector of numbers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawEllipseLines(int centerX, int centerY, float radiusH, float radiusV, Color color);
```

See Also

`color()` for creating color objects.

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_v()`, `draw_cube_wires_v()`, `draw_cube_wires()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`, `draw_point_3d()`, `draw_poly_lines_ex()`, `draw_poly_lines()`, `draw_poly()`, `draw_ray()`, `draw_rectangle_gradient_ex()`, `draw_rectangle_gradient_h()`, `draw_rectangle_gradient_v()`, `draw_rectangle_lines_ex()`, `draw_rectangle_lines()`, `draw_rectangle_pro()`, `draw_rectangle_rec()`, `draw_rectangle_rounded_lines()`, `draw_rectangle_rounded()`, `draw_rectangle_v()`, `draw_rectangle()`, `draw_ring_lines()`, `draw_ring()`, `draw_sphere_ex()`, `draw_sphere_wires()`, `draw_sphere()`, `draw_text_codepoint()`, `draw_text_ex()`, `draw_text_pro()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`, `draw_texture_rec()`, `draw_texture_tiled()`, `draw_texture_v()`, `draw_texture()`, `draw_text()`, `draw_triangle_3d()`, `draw_triangle_lines()`, `draw_triangle_strip_3d()`, `draw_triangle()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_l`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`

---

draw_fps *Draw fps*

Description

Draw current FPS.

Usage

```
draw_fps(pos_x, pos_y)
```

Arguments

pos_x	An integer or a vector of integers.
pos_y	An integer or a vector of integers.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawFPS(int posX, int posY);
```

See Also

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other fps functions: get_fps(), set_target_fps()

---

draw_grid *Draw grid*

Description

Draw a grid (centered at (0, 0, 0)).

Usage

```
draw_grid(slices, spacing)
```

Arguments

slices	An integer or a vector of integers.
spacing	A number or a vector of numbers.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawGrid(int slices, float spacing);
```

See Also

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

---

draw_line *Draw line*

Description

Draw a line.

Usage

```
draw_line(start_pos_x, start_pos_y, end_pos_x, end_pos_y, color)
```

Arguments

start_pos_x	An integer or a vector of integers.
start_pos_y	An integer or a vector of integers.
end_pos_x	An integer or a vector of integers.
end_pos_y	An integer or a vector of integers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawLine(int startPosX, int startPosY, int endPosX, int endPosY, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other line functions: check_collision_point_line(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), image_draw_line_v(), image_draw_line()

---

draw_line_3d *Draw line 3d*

Description

Draw a line in 3D world space.

Usage

```
draw_line_3d(start_pos, end_pos, color)
```

Arguments

start_pos	A numeric vector of length 3 or a numeric matrix of width 3.
end_pos	A numeric vector of length 3 or a numeric matrix of width 3.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawLine3D(Vector3 startPos, Vector3 endPos, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other line functions: check_collision_point_line(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), image_draw_line_v(), image_draw_line()

---

draw_line_bezier *Draw line bezier*

Description

Draw a line using cubic-bezier curves in-out.

Usage

```
draw_line_bezier(start_pos, end_pos, thick, color)
```

Arguments

start_pos	A numeric vector of length 2 or a numeric matrix of width 2.
end_pos	A numeric vector of length 2 or a numeric matrix of width 2.
thick	A number or a vector of numbers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawLineBezier(Vector2 startPos, Vector2 endPos, float thick, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other line functions: check_collision_point_line(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_ex(), draw_line_v(), draw_line(), image_draw_line_v(), image_draw_line()

---

draw_line_bezier_cubic

Draw line bezier cubic

Description

Draw line using cubic bezier curves with 2 control points.

Usage

```
draw_line_bezier_cubic(
  start_pos,
  end_pos,
  start_control_pos,
  end_control_pos,
  thick,
  color
)
```

Arguments

start_pos	A numeric vector of length 2 or a numeric matrix of width 2.
end_pos	A numeric vector of length 2 or a numeric matrix of width 2.
start_control_pos	A numeric vector of length 2 or a numeric matrix of width 2.
end_control_pos	A numeric vector of length 2 or a numeric matrix of width 2.
thick	A number or a vector of numbers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawLineBezierCubic(Vector2 startPos, Vector2 endPos, Vector2 startControlPos, Vector2 endControl
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(),

image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l
image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(),
image_draw_text(), image_draw()

Other line functions: check_collision_point_line(), draw_line_3d(), draw_line_bezier_quad(),
draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), image_draw_line_v(),
image_draw_line()

`draw_line_bezier_quad` *Draw line bezier quad*

Description

Draw line using quadratic bezier curves with a control point.

Usage

```
draw_line_bezier_quad(start_pos, end_pos, control_pos, thick, color)
```

Arguments

start_pos	A numeric vector of length 2 or a numeric matrix of width 2.
end_pos	A numeric vector of length 2 or a numeric matrix of width 2.
control_pos	A numeric vector of length 2 or a numeric matrix of width 2.
thick	A number or a vector of numbers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawLineBezierQuad(Vector2 startPos, Vector2 endPos, Vector2 controlPos, float thick, Color color)
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(),
draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(),
draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(),
draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(),
draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(),
draw_line_bezier_cubic(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(),
draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(),
draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(),
draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(),
draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(),

draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other line functions: check_collision_point_line(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), image_draw_line_v(), image_draw_line()

`draw_line_ex` *Draw line ex*

Description

Draw a line defining thickness.

Usage

```
draw_line_ex(start_pos, end_pos, thick, color)
```

Arguments

`start_pos`	A numeric vector of length 2 or a numeric matrix of width 2.
`end_pos`	A numeric vector of length 2 or a numeric matrix of width 2.
`thick`	A number or a vector of numbers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawLineEx(Vector2 startPos, Vector2 endPos, float thick, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(),

draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other line functions: check_collision_point_line(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_v(), draw_line(), image_draw_line_v(), image_draw_line()

`draw_line_v` *Draw line v*

Description

Draw a line (Vector version).

Usage

```
draw_line_v(start_pos, end_pos, color)
```

Arguments

`start_pos`	A numeric vector of length 2 or a numeric matrix of width 2.
`end_pos`	A numeric vector of length 2 or a numeric matrix of width 2.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawLineV(Vector2 startPos, Vector2 endPos, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other line functions: check_collision_point_line(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line(), image_draw_line_v(), image_draw_line()

`draw_model` *Draw model*

Description

Draw a model (with texture if set).

Usage

```
draw_model(model, position, scale, tint)
```

Arguments

`model`	A model or a list of models.
`position`	A numeric vector of length 3 or a numeric matrix of width 3.
`scale`	A number or a vector of numbers.
`tint`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawModel(Model model, Vector3 position, float scale, Color tint);
```

See Also

model() for creating model objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other model functions: draw_model_ex(), draw_model_wires(), get_model_bounding_box(), get_ray_collision_model(), load_model(), unload_model()

`draw_model_ex` *Draw model ex*

Description

Draw a model with extended parameters.

Usage

```
draw_model_ex(model, position, rotation_axis, rotation_angle, scale, tint)
```

Arguments

model	A model or a list of models.
position	A numeric vector of length 3 or a numeric matrix of width 3.
rotation_axis	A numeric vector of length 3 or a numeric matrix of width 3.
rotation_angle	A number or a vector of numbers.
scale	A numeric vector of length 3 or a numeric matrix of width 3.
tint	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawModelEx(Model model, Vector3 position, Vector3 rotationAxis, float rotationAngle, Vector3 scal
```

See Also

model() for creating model objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other model functions: draw_model_wires(), draw_model(), get_model_bounding_box(), get_ray_collision_model(load_model(), unload_model()

`draw_model_wires` *Draw model wires*

Description

Draw a model wires (with texture if set).

Usage

```
draw_model_wires(model, position, scale, tint)
```

Arguments

`model`	A model or a list of models.
`position`	A numeric vector of length 3 or a numeric matrix of width 3.
`scale`	A number or a vector of numbers.
`tint`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawModelWires(Model model, Vector3 position, float scale, Color tint);
```

See Also

`model()` for creating model objects.

`color()` for creating color objects.

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_v()`, `draw_cube_wires_v()`, `draw_cube_wires()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse_lines()`, `draw_ellipse()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`, `draw_point_3d()`, `draw_poly_lines_ex()`, `draw_poly_lines()`, `draw_poly()`, `draw_ray()`, `draw_rectangle_gradient_ex()`, `draw_rectangle_gradient_h()`, `draw_rectangle_gradient_v()`, `draw_rectangle_lines_ex()`, `draw_rectangle_lines()`, `draw_rectangle_pro()`, `draw_rectangle_rec()`, `draw_rectangle_rounded_lines()`, `draw_rectangle_rounded()`, `draw_rectangle_v()`, `draw_rectangle()`, `draw_ring_lines()`, `draw_ring()`, `draw_sphere_ex()`, `draw_sphere_wires()`, `draw_sphere()`, `draw_text_codepoint()`, `draw_text_ex()`, `draw_text_pro()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`, `draw_texture_rec()`, `draw_texture_tiled()`, `draw_texture_v()`, `draw_texture()`, `draw_text()`, `draw_triangle_3d()`, `draw_triangle_lines()`, `draw_triangle_strip_3d()`, `draw_triangle()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_l`

image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other model functions: draw_model_ex(), draw_model(), get_model_bounding_box(), get_ray_collision_model(), load_model(), unload_model()

`draw_pixel` *Draw pixel*

Description

Draw a pixel.

Usage

```
draw_pixel(pos_x, pos_y, color)
```

Arguments

`pos_x`	An integer or a vector of integers.
`pos_y`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawPixel(int posX, int posY, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(),

draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_pixel_v` *Draw pixel v*

Description

Draw a pixel (Vector version).

Usage

```
draw_pixel_v(position, color)
```

Arguments

`position`	A numeric vector of length 2 or a numeric matrix of width 2.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawPixelV(Vector2 position, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(),

draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_plane` *Draw plane*

Description

Draw a plane XZ.

Usage

```
draw_plane(center_pos, size, color)
```

Arguments

`center_pos`	A numeric vector of length 3 or a numeric matrix of width 3.
`size`	A numeric vector of length 2 or a numeric matrix of width 2.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawPlane(Vector3 centerPos, Vector2 size, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(),

draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_point_3d` *Draw point 3d*

Description

Draw a point in 3D space, actually a small line.

Usage

```
draw_point_3d(position, color)
```

Arguments

`position`	A numeric vector of length 3 or a numeric matrix of width 3.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawPoint3D(Vector3 position, Color color);
```

See Also

`color()` for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(),

image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l
image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(),
image_draw_text(), image_draw()

`draw_poly` *Draw poly*

Description

Draw a regular polygon (Vector version).

Usage

```
draw_poly(center, sides, radius, rotation, color)
```

Arguments

`center`	A numeric vector of length 2 or a numeric matrix of width 2.
`sides`	An integer or a vector of integers.
`radius`	A number or a vector of numbers.
`rotation`	A number or a vector of numbers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawPoly(Vector2 center, int sides, float radius, float rotation, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(),

draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_poly_lines` *Draw poly lines*

Description

Draw a polygon outline of n sides.

Usage

```
draw_poly_lines(center, sides, radius, rotation, color)
```

Arguments

`center`	A numeric vector of length 2 or a numeric matrix of width 2.
`sides`	An integer or a vector of integers.
`radius`	A number or a vector of numbers.
`rotation`	A number or a vector of numbers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawPolyLines(Vector2 center, int sides, float radius, float rotation, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(),

draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_poly_lines_ex` *Draw poly lines ex*

Description

Draw a polygon outline of n sides with extended parameters.

Usage

```
draw_poly_lines_ex(center, sides, radius, rotation, line_thick, color)
```

Arguments

`center`	A numeric vector of length 2 or a numeric matrix of width 2.
`sides`	An integer or a vector of integers.
`radius`	A number or a vector of numbers.
`rotation`	A number or a vector of numbers.
`line_thick`	A number or a vector of numbers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawPolyLinesEx(Vector2 center, int sides, float radius, float rotation, float lineThick, Color co
```

See Also

`color()` for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(),

draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_ray` *Draw ray*

Description

Draw a ray line.

Usage

```
draw_ray(ray, color)
```

Arguments

`ray`	A ray or a list of rays.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRay(Ray ray, Color color);
```

See Also

`color()` for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(),

draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_rectangle` *Draw rectangle*

Description

Draw a color-filled rectangle.

Usage

```
draw_rectangle(pos_x, pos_y, width, height, color)
```

Arguments

`pos_x`	An integer or a vector of integers.
`pos_y`	An integer or a vector of integers.
`width`	An integer or a vector of integers.
`height`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRectangle(int posX, int posY, int width, int height, Color color);
```

See Also

`color()` for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(),

draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

`draw_rectangle_gradient_ex` *Draw rectangle gradient ex*

Description

Draw a gradient-filled rectangle with custom vertex colors.

Usage

```
draw_rectangle_gradient_ex(rec, col_1, col_2, col_3, col_4)
```

Arguments

rec	A rectangle or a list of rectangles.
col_1	A color or a list of colors.
col_2	A color or a list of colors.
col_3	A color or a list of colors.
col_4	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRectangleGradientEx(Rectangle rec, Color col1, Color col2, Color col3, Color col4);
```

See Also

rectangle() for creating rectangle objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_e draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

`draw_rectangle_gradient_h` *Draw rectangle gradient h*

Description

Draw a horizontal-gradient-filled rectangle.

Usage

```
draw_rectangle_gradient_h(pos_x, pos_y, width, height, color_1, color_2)
```

Arguments

pos_x	An integer or a vector of integers.
pos_y	An integer or a vector of integers.
width	An integer or a vector of integers.
height	An integer or a vector of integers.

color_1	A color or a list of colors.
color_2	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRectangleGradientH(int posX, int posY, int width, int height, Color color1, Color color2);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

---

draw_rectangle_gradient_v

Draw rectangle gradient v

Description

Draw a vertical-gradient-filled rectangle.

Usage

```
draw_rectangle_gradient_v(pos_x, pos_y, width, height, color_1, color_2)
```

Arguments

pos_x	An integer or a vector of integers.
pos_y	An integer or a vector of integers.
width	An integer or a vector of integers.
height	An integer or a vector of integers.
color_1	A color or a list of colors.
color_2	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRectangleGradientV(int posX, int posY, int width, int height, Color color1, Color color2);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

`draw_rectangle_lines` *Draw rectangle lines*

Description

Draw rectangle outline.

Usage

```
draw_rectangle_lines(pos_x, pos_y, width, height, color)
```

Arguments

`pos_x`	An integer or a vector of integers.
`pos_y`	An integer or a vector of integers.
`width`	An integer or a vector of integers.
`height`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRectangleLines(int posX, int posY, int width, int height, Color color);
```

See Also

`color()` for creating color objects.

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_v()`, `draw_cube_wires_v()`, `draw_cube_wires()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse_lines()`, `draw_ellipse()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`, `draw_point_3d()`, `draw_poly_lines_ex()`, `draw_poly_lines()`, `draw_poly()`, `draw_ray()`, `draw_rectangle_gradient_ex()`, `draw_rectangle_gradient_h()`, `draw_rectangle_gradient_v()`, `draw_rectangle_lines_ex()`, `draw_rectangle_pro()`, `draw_rectangle_rec()`, `draw_rectangle_rounded_lines()`, `draw_rectangle_rounded()`, `draw_rectangle_v()`, `draw_rectangle()`, `draw_ring_lines()`, `draw_ring()`, `draw_sphere_ex()`, `draw_sphere_wires()`, `draw_sphere()`, `draw_text_codepoint()`, `draw_text_ex()`, `draw_text_pro()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`, `draw_texture_rec()`, `draw_texture_tiled()`, `draw_texture_v()`, `draw_texture()`, `draw_text()`, `draw_triangle_3d()`, `draw_triangle_lines()`, `draw_triangle_strip_3d()`, `draw_triangle()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_l`

image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

`draw_rectangle_lines_ex` *Draw rectangle lines ex*

Description

Draw rectangle outline with extended parameters.

Usage

```
draw_rectangle_lines_ex(rec, line_thick, color)
```

Arguments

`rec`	A rectangle or a list of rectangles.
`line_thick`	A number or a vector of numbers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRectangleLinesEx(Rectangle rec, float lineThick, Color color);
```

See Also

rectangle() for creating rectangle objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(),

draw_rectangle_gradient_v(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

`draw_rectangle_pro` *Draw rectangle pro*

Description

Draw a color-filled rectangle with pro parameters.

Usage

```
draw_rectangle_pro(rec, origin, rotation, color)
```

Arguments

`rec`	A rectangle or a list of rectangles.
`origin`	A numeric vector of length 2 or a numeric matrix of width 2.
`rotation`	A number or a vector of numbers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRectanglePro(Rectangle rec, Vector2 origin, float rotation, Color color);
```

See Also

rectangle() for creating rectangle objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

---

draw_rectangle_rec *Draw rectangle rec*

Description

Draw a color-filled rectangle.

Usage

```
draw_rectangle_rec(rec, color)
```

Arguments

rec	A rectangle or a list of rectangles.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRectangleRec(Rectangle rec, Color color);
```

See Also

rectangle() for creating rectangle objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

---

draw_rectangle_rounded

Draw rectangle rounded

Description

Draw rectangle with rounded edges.

Usage

```
draw_rectangle_rounded(rec, roundness, segments, color)
```

Arguments

rec	A rectangle or a list of rectangles.
roundness	A number or a vector of numbers.
segments	An integer or a vector of integers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRectangleRounded(Rectangle rec, float roundness, int segments, Color color);
```

See Also

rectangle() for creating rectangle objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

`draw_rectangle_rounded_lines` *Draw rectangle rounded lines*

Description

Draw rectangle with rounded edges outline.

Usage

```
draw_rectangle_rounded_lines(rec, roundness, segments, line_thick, color)
```

Arguments

`rec`	A rectangle or a list of rectangles.
`roundness`	A number or a vector of numbers.
`segments`	An integer or a vector of integers.
`line_thick`	A number or a vector of numbers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRectangleRoundedLines(Rectangle rec, float roundness, int segments, float lineThick, Color col
```

See Also

`rectangle()` for creating rectangle objects.

`color()` for creating color objects.

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_v()`, `draw_cube_wires_v()`, `draw_cube_wires()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse_lines()`, `draw_ellipse()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`, `draw_point_3d()`, `draw_poly_lines_ex()`, `draw_poly_lines()`, `draw_poly()`, `draw_ray()`, `draw_rectangle_gradient_ex()`, `draw_rectangle_gradient_h()`, `draw_rectangle_gradient_v()`, `draw_rectangle_lines_ex()`, `draw_rectangle_lines()`, `draw_rectangle_pro()`, `draw_rectangle_rec()`, `draw_rectangle_rounded()`, `draw_rectangle_v()`, `draw_rectangle()`, `draw_ring_lines()`, `draw_ring()`, `draw_sphere_ex()`, `draw_sphere_wires()`, `draw_sphere()`, `draw_text_codepoint()`, `draw_text_ex()`, `draw_text_pro()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`, `draw_texture_rec()`, `draw_texture_tiled()`,

draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

`draw_rectangle_v` *Draw rectangle v*

Description

Draw a color-filled rectangle (Vector version).

Usage

```
draw_rectangle_v(position, size, color)
```

Arguments

`position`	A numeric vector of length 2 or a numeric matrix of width 2.
`size`	A numeric vector of length 2 or a numeric matrix of width 2.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRectangleV(Vector2 position, Vector2 size, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(),

draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

`draw_ring` *Draw ring*

Description

Draw ring.

Usage

```
draw_ring(
  center,
  inner_radius,
  outer_radius,
  start_angle,
  end_angle,
  segments,
  color
)
```

Arguments

`center`	A numeric vector of length 2 or a numeric matrix of width 2.
`inner_radius`	A number or a vector of numbers.
`outer_radius`	A number or a vector of numbers.
`start_angle`	A number or a vector of numbers.
`end_angle`	A number or a vector of numbers.
`segments`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRing(Vector2 center, float innerRadius, float outerRadius, float startAngle, float endAngle, i
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_ring_lines` *Draw ring lines*

Description

Draw ring outline.

Usage

```
draw_ring_lines(
  center,
  inner_radius,
  outer_radius,
  start_angle,
  end_angle,
  segments,
  color
)
```

Arguments

center	A numeric vector of length 2 or a numeric matrix of width 2.
inner_radius	A number or a vector of numbers.
outer_radius	A number or a vector of numbers.
start_angle	A number or a vector of numbers.
end_angle	A number or a vector of numbers.
segments	An integer or a vector of integers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawRingLines(Vector2 center, float innerRadius, float outerRadius, float startAngle, float endAng
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

draw_sphere *Draw sphere*

Description

Draw sphere.

Usage

```
draw_sphere(center_pos, radius, color)
```

Arguments

center_pos	A numeric vector of length 3 or a numeric matrix of width 3.
radius	A number or a vector of numbers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawSphere(Vector3 centerPos, float radius, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_sphere_ex` *Draw sphere ex*

Description

Draw sphere with extended parameters.

Usage

```
draw_sphere_ex(center_pos, radius, rings, slices, color)
```

Arguments

`center_pos`	A numeric vector of length 3 or a numeric matrix of width 3.
`radius`	A number or a vector of numbers.
`rings`	An integer or a vector of integers.
`slices`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawSphereEx(Vector3 centerPos, float radius, int rings, int slices, Color color);
```

See Also

`color()` for creating color objects.

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_v()`, `draw_cube_wires_v()`, `draw_cube_wires()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse_lines()`, `draw_ellipse()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`, `draw_point_3d()`, `draw_poly_lines_ex()`, `draw_poly_lines()`, `draw_poly()`, `draw_ray()`, `draw_rectangle_gradient_ex()`, `draw_rectangle_gradient_h()`, `draw_rectangle_gradient_v()`, `draw_rectangle_lines_ex()`, `draw_rectangle_lines()`, `draw_rectangle_pro()`, `draw_rectangle_rec()`, `draw_rectangle_rounded_lines()`, `draw_rectangle_rounded()`, `draw_rectangle_v()`, `draw_rectangle()`, `draw_ring_lines()`, `draw_ring()`, `draw_sphere_wires()`, `draw_sphere()`, `draw_text_codepoint()`, `draw_text_ex()`, `draw_text_pro()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`, `draw_texture_rec()`, `draw_texture_tiled()`, `draw_texture_v()`, `draw_texture()`, `draw_text()`, `draw_triangle_3d()`, `draw_triangle_lines()`, `draw_triangle_strip_3d()`, `draw_triangle()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_l`

image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_sphere_wires` *Draw sphere wires*

Description

Draw sphere wires.

Usage

```
draw_sphere_wires(center_pos, radius, rings, slices, color)
```

Arguments

`center_pos`	A numeric vector of length 3 or a numeric matrix of width 3.
`radius`	A number or a vector of numbers.
`rings`	An integer or a vector of integers.
`slices`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawSphereWires(Vector3 centerPos, float radius, int rings, int slices, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(),

draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_text` *Draw text*

Description

Draw text (using default font).

Usage

```
draw_text(text, pos_x, pos_y, font_size, color)
```

Arguments

`text`	A string or a vector of strings.
`pos_x`	An integer or a vector of integers.
`pos_y`	An integer or a vector of integers.
`font_size`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawText(const char *text, int posX, int posY, int fontSize, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(),

draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_find_index(), text_insert(), text_replace(), text_to_integer(), text_to_lower(), text_to_pascal(), text_to_upper()

`draw_text_codepoint` *Draw text codepoint*

Description

Draw one character (codepoint).

Usage

```
draw_text_codepoint(font, codepoint, position, font_size, tint)
```

Arguments

`font`	A font or a list of fonts.
`codepoint`	An integer or a vector of integers.
`position`	A numeric vector of length 2 or a numeric matrix of width 2.
`font_size`	A number or a vector of numbers.
`tint`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTextCodepoint(Font font, int codepoint, Vector2 position, float fontSize, Color tint);
```

See Also

font() for creating font objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other text functions: draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_find_index(), text_insert(), text_replace(), text_to_integer(), text_to_lower(), text_to_pascal(), text_to_upper()

`draw_text_ex` *Draw text ex*

Description

Draw text using font and additional parameters.

Usage

```
draw_text_ex(font, text, position, font_size, spacing, tint)
```

Arguments

font	A font or a list of fonts.
text	A string or a vector of strings.
position	A numeric vector of length 2 or a numeric matrix of width 2.
font_size	A number or a vector of numbers.
spacing	A number or a vector of numbers.
tint	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTextEx(Font font, const char *text, Vector2 position, float fontSize, float spacing, Color tint
```

See Also

`font()` for creating font objects.

`color()` for creating color objects.

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_v()`, `draw_cube_wires_v()`, `draw_cube_wires()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse_lines()`, `draw_ellipse()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`, `draw_point_3d()`, `draw_poly_lines_ex()`, `draw_poly_lines()`, `draw_poly()`, `draw_ray()`, `draw_rectangle_gradient_ex()`, `draw_rectangle_gradient_h()`, `draw_rectangle_gradient_v()`, `draw_rectangle_lines_ex()`, `draw_rectangle_lines()`, `draw_rectangle_pro()`, `draw_rectangle_rec()`, `draw_rectangle_rounded_lines()`, `draw_rectangle_rounded()`, `draw_rectangle_v()`, `draw_rectangle()`, `draw_ring_lines()`, `draw_ring()`, `draw_sphere_ex()`, `draw_sphere_wires()`, `draw_sphere()`, `draw_text_codepoint()`, `draw_text_pro()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`, `draw_texture_rec()`, `draw_texture_tiled()`, `draw_texture_v()`, `draw_texture()`, `draw_text()`, `draw_triangle_3d()`, `draw_triangle_lines()`, `draw_triangle_strip_3d()`, `draw_triangle()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_l`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`

Other text functions: `draw_text_codepoint()`, `draw_text_pro()`, `draw_text()`, `image_draw_text_ex()`, `image_draw_text()`, `image_text_ex()`, `image_text()`, `load_file_text()`, `measure_text_ex()`, `measure_text()`, `text_find_index()`, `text_insert()`, `text_replace()`, `text_to_integer()`, `text_to_lower()`, `text_to_pascal()`, `text_to_upper()`

---

draw_text_pro *Draw text pro*

Description

Draw text using Font and pro parameters (rotation).

Usage

```
draw_text_pro(font, text, position, origin, rotation, font_size, spacing, tint)
```

Arguments

font	A font or a list of fonts.
text	A string or a vector of strings.
position	A numeric vector of length 2 or a numeric matrix of width 2.
origin	A numeric vector of length 2 or a numeric matrix of width 2.
rotation	A number or a vector of numbers.
font_size	A number or a vector of numbers.
spacing	A number or a vector of numbers.
tint	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTextPro(Font font, const char *text, Vector2 position, Vector2 origin, float rotation, float fo
```

See Also

font() for creating font objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_find_index(), text_insert(), text_replace(), text_to_integer(), text_to_lower(), text_to_pascal(), text_to_upper()

`draw_texture` *Draw texture*

Description

Draw a Texture2D.

Usage

```
draw_texture(texture, pos_x, pos_y, tint)
```

Arguments

`texture`	A texture or a list of textures.
`pos_x`	An integer or a vector of integers.
`pos_y`	An integer or a vector of integers.
`tint`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTexture(Texture2D texture, int posX, int posY, Color tint);
```

See Also

`texture()` for creating texture objects.

`color()` for creating color objects.

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_v()`, `draw_cube_wires_v()`, `draw_cube_wires()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse_lines()`, `draw_ellipse()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`, `draw_point_3d()`, `draw_poly_lines_ex()`, `draw_poly_lines()`, `draw_poly()`, `draw_ray()`, `draw_rectangle_gradient_ex()`, `draw_rectangle_gradient_h()`, `draw_rectangle_gradient_v()`, `draw_rectangle_lines_ex()`, `draw_rectangle_lines()`, `draw_rectangle_pro()`, `draw_rectangle_rec()`, `draw_rectangle_rounded_lines()`, `draw_rectangle_rounded()`, `draw_rectangle_v()`, `draw_rectangle()`, `draw_ring_lines()`, `draw_ring()`, `draw_sphere_ex()`, `draw_sphere_wires()`, `draw_sphere()`, `draw_text_codepoint()`, `draw_text_ex()`, `draw_text_pro()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`, `draw_texture_rec()`, `draw_texture_tiled()`, `draw_texture_v()`, `draw_text()`, `draw_triangle_3d()`, `draw_triangle_lines()`, `draw_triangle_strip_3d()`, `draw_triangle()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`,

image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

`draw_texture_ex` *Draw texture ex*

Description

Draw a Texture2D with extended parameters.

Usage

```
draw_texture_ex(texture, position, rotation, scale, tint)
```

Arguments

`texture`	A texture or a list of textures.
`position`	A numeric vector of length 2 or a numeric matrix of width 2.
`rotation`	A number or a vector of numbers.
`scale`	A number or a vector of numbers.
`tint`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTextureEx(Texture2D texture, Vector2 position, float rotation, float scale, Color tint);
```

See Also

texture() for creating texture objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(),

draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

`draw_texture_npatch` *Draw texture npatch*

Description

Draws a texture (or part of it) that stretches or shrinks nicely.

Usage

```
draw_texture_npatch(texture, n_patch_info, dest, origin, rotation, tint)
```

Arguments

`texture`	A texture or a list of textures.
`n_patch_info`	A npatch_info or a list of npatch_infos.
`dest`	A rectangle or a list of rectangles.
`origin`	A numeric vector of length 2 or a numeric matrix of width 2.
`rotation`	A number or a vector of numbers.
`tint`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTextureNPatch(Texture2D texture, NPatchInfo nPatchInfo, Rectangle dest, Vector2 origin, float
```

See Also

texture() for creating texture objects.

rectangle() for creating rectangle objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_l image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

`draw_texture_poly` *Draw texture poly*

Description

Draw a textured polygon.

Usage

```
draw_texture_poly(texture, center, points, texcoords, point_count, tint)
```

Arguments

texture	A texture or a list of textures.
center	A numeric vector of length 2 or a numeric matrix of width 2.

points	A numeric vector of length 2 or a numeric matrix of width 2.
texcoords	A numeric vector of length 2 or a numeric matrix of width 2.
point_count	An integer or a vector of integers.
tint	A color or a list of colors.

Note

This function has been auto-generated from the following Raylib function definition:

```
void DrawTexturePoly(Texture2D texture, Vector2 center, Vector2 *points, Vector2 *texcoords, int point
```

See Also

texture() for creating texture objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

`draw_texture_pro` *Draw texture pro*

Description

Draw a part of a texture defined by a rectangle with 'pro' parameters.

Usage

```
draw_texture_pro(texture, source, dest, origin, rotation, tint)
```

Arguments

`texture`	A texture or a list of textures.
`source`	A rectangle or a list of rectangles.
`dest`	A rectangle or a list of rectangles.
`origin`	A numeric vector of length 2 or a numeric matrix of width 2.
`rotation`	A number or a vector of numbers.
`tint`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTexturePro(Texture2D texture, Rectangle source, Rectangle dest, Vector2 origin, float rotation
```

See Also

`texture()` for creating texture objects.

`rectangle()` for creating rectangle objects.

`color()` for creating color objects.

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`, `draw_circle()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_cube_v()`, `draw_cube_wires_v()`, `draw_cube_wires()`, `draw_cube()`, `draw_cylinder_ex()`, `draw_cylinder_wires_ex()`, `draw_cylinder_wires()`, `draw_cylinder()`, `draw_ellipse_lines()`, `draw_ellipse()`, `draw_fps()`, `draw_grid()`, `draw_line_3d()`, `draw_line_bezier_cubic()`, `draw_line_bezier_quad()`, `draw_line_bezier()`, `draw_line_ex()`, `draw_line_v()`, `draw_line()`, `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `draw_pixel_v()`, `draw_pixel()`, `draw_plane()`, `draw_point_3d()`, `draw_poly_lines_ex()`, `draw_poly_lines()`, `draw_poly()`, `draw_ray()`, `draw_rectangle_gradient_ex()`, `draw_rectangle_gradient_h()`, `draw_rectangle_gradient_v()`, `draw_rectangle_lines_ex()`, `draw_rectangle_lines()`, `draw_rectangle_pro()`, `draw_rectangle_rec()`, `draw_rectangle_rounded_lines()`, `draw_rectangle_rounded()`, `draw_rectangle_v()`, `draw_rectangle()`, `draw_ring_lines()`, `draw_ring()`, `draw_sphere_ex()`, `draw_sphere_wires()`, `draw_sphere()`, `draw_text_codepoint()`, `draw_text_ex()`, `draw_text_pro()`, `draw_texture_ex()`,

draw_texture_npatch(), draw_texture_poly(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

`draw_texture_quad` *Draw texture quad*

Description

Draw texture quad with tiling and offset parameters.

Usage

```
draw_texture_quad(texture, tiling, offset, quad, tint)
```

Arguments

`texture`	A texture or a list of textures.
`tiling`	A numeric vector of length 2 or a numeric matrix of width 2.
`offset`	A numeric vector of length 2 or a numeric matrix of width 2.
`quad`	A rectangle or a list of rectangles.
`tint`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTextureQuad(Texture2D texture, Vector2 tiling, Vector2 offset, Rectangle quad, Color tint);
```

See Also

texture() for creating texture objects.

rectangle() for creating rectangle objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

---

draw_texture_rec *Draw texture rec*

Description

Draw a part of a texture defined by a rectangle.

Usage

```
draw_texture_rec(texture, source, position, tint)
```

Arguments

texture	A texture or a list of textures.
source	A rectangle or a list of rectangles.
position	A numeric vector of length 2 or a numeric matrix of width 2.
tint	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTextureRec(Texture2D texture, Rectangle source, Vector2 position, Color tint);
```

See Also

texture() for creating texture objects.

rectangle() for creating rectangle objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

---

draw_texture_tiled *Draw texture tiled*

Description

Draw part of a texture (defined by a rectangle) with rotation and scale tiled into dest..

Usage

```
draw_texture_tiled(texture, source, dest, origin, rotation, scale, tint)
```

Arguments

texture	A texture or a list of textures.
source	A rectangle or a list of rectangles.
dest	A rectangle or a list of rectangles.
origin	A numeric vector of length 2 or a numeric matrix of width 2.
rotation	A number or a vector of numbers.
scale	A number or a vector of numbers.
tint	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTextureTiled(Texture2D texture, Rectangle source, Rectangle dest, Vector2 origin, float rotati
```

See Also

texture() for creating texture objects.

rectangle() for creating rectangle objects.

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(),

draw_texture_quad(), draw_texture_rec(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

`draw_texture_v` *Draw texture v*

Description

Draw a Texture2D with position defined as Vector2.

Usage

```
draw_texture_v(texture, position, tint)
```

Arguments

`texture`	A texture or a list of textures.
`position`	A numeric vector of length 2 or a numeric matrix of width 2.
`tint`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTextureV(Texture2D texture, Vector2 position, Color tint);
```

See Also

`texture()` for creating texture objects.

`color()` for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(),

draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

`draw_triangle` *Draw triangle*

Description

Draw a color-filled triangle (vertex in counter-clockwise order!).

Usage

```
draw_triangle(v_1, v_2, v_3, color)
```

Arguments

v_1	A numeric vector of length 2 or a numeric matrix of width 2.
v_2	A numeric vector of length 2 or a numeric matrix of width 2.
v_3	A numeric vector of length 2 or a numeric matrix of width 2.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTriangle(Vector2 v1, Vector2 v2, Vector2 v3, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(),

draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_triangle_3d` *Draw triangle 3d*

Description

Draw a color-filled triangle (vertex in counter-clockwise order!).

Usage

```
draw_triangle_3d(v_1, v_2, v_3, color)
```

Arguments

v_1	A numeric vector of length 3 or a numeric matrix of width 3.
v_2	A numeric vector of length 3 or a numeric matrix of width 3.
v_3	A numeric vector of length 3 or a numeric matrix of width 3.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTriangle3D(Vector3 v1, Vector3 v2, Vector3 v3, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(),

draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_triangle_lines` *Draw triangle lines*

Description

Draw triangle outline (vertex in counter-clockwise order!).

Usage

```
draw_triangle_lines(v_1, v_2, v_3, color)
```

Arguments

v_1	A numeric vector of length 2 or a numeric matrix of width 2.
v_2	A numeric vector of length 2 or a numeric matrix of width 2.
v_3	A numeric vector of length 2 or a numeric matrix of width 2.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void DrawTriangleLines(Vector2 v1, Vector2 v2, Vector2 v3, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`draw_triangle_strip_3d` *Draw triangle strip 3d*

Description

Draw a triangle strip defined by points.

Usage

```
draw_triangle_strip_3d(points, point_count, color)
```

Arguments

`points`	A numeric vector of length 3 or a numeric matrix of width 3.
`point_count`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function has been auto-generated from the following Raylib function definition:

```
void DrawTriangleStrip3D(Vector3 *points, int pointCount, Color color);
```

See Also

color() for creating color objects.

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

---

enable_cursor *Enable cursor*

Description

Enables cursor (unlock cursor).

Usage

```
enable_cursor()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void EnableCursor(void);
```

See Also

Other cursor functions: disable_cursor(), hide_cursor(), is_cursor_hidden(), is_cursor_on_screen(), set_mouse_cursor(), show_cursor()

end_blend_mode *End blend mode*

Description

End blending mode (reset to default: alpha blending).

Usage

```
end_blend_mode()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void EndBlendMode(void);
```

See Also

Other mode functions: `begin_blend_mode()`, `begin_mode_2d()`, `begin_mode_3d()`, `begin_scissor_mode()`, `begin_shader_mode()`, `begin_texture_mode()`, `end_mode_2d()`, `end_mode_3d()`, `end_scissor_mode()`, `end_shader_mode()`, `end_texture_mode()`, `end_vr_stereo_mode()`, `set_camera_mode()`

end_drawing *End drawing*

Description

End canvas drawing and swap buffers (double buffering).

Usage

```
end_drawing()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void EndDrawing(void);
```

end_mode_2d *End mode 2d*

Description

Ends 2D mode with custom camera.

Usage

```
end_mode_2d()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void EndMode2D(void);
```

See Also

Other mode functions: begin_blend_mode(), begin_mode_2d(), begin_mode_3d(), begin_scissor_mode(), begin_shader_mode(), begin_texture_mode(), end_blend_mode(), end_mode_3d(), end_scissor_mode(), end_shader_mode(), end_texture_mode(), end_vr_stereo_mode(), set_camera_mode()

end_mode_3d *End mode 3d*

Description

Ends 3D mode and returns to default 2D orthographic mode.

Usage

```
end_mode_3d()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void EndMode3D(void);
```

See Also

Other mode functions: begin_blend_mode(), begin_mode_2d(), begin_mode_3d(), begin_scissor_mode(), begin_shader_mode(), begin_texture_mode(), end_blend_mode(), end_mode_2d(), end_scissor_mode(), end_shader_mode(), end_texture_mode(), end_vr_stereo_mode(), set_camera_mode()

end_scissor_mode *End scissor mode*

Description

End scissor mode.

Usage

```
end_scissor_mode()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void EndScissorMode(void);
```

See Also

Other mode functions: begin_blend_mode(), begin_mode_2d(), begin_mode_3d(), begin_scissor_mode(), begin_shader_mode(), begin_texture_mode(), end_blend_mode(), end_mode_2d(), end_mode_3d(), end_shader_mode(), end_texture_mode(), end_vr_stereo_mode(), set_camera_mode()

end_shader_mode *End shader mode*

Description

End custom shader drawing (use default shader).

Usage

```
end_shader_mode()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void EndShaderMode(void);
```

See Also

Other shader functions: begin_shader_mode(), get_shader_location_attrib(), get_shader_location(), load_shader_from_memory(), load_shader(), set_shader_value_matrix(), set_shader_value_texture(), set_shader_value(), unload_shader()

Other mode functions: begin_blend_mode(), begin_mode_2d(), begin_mode_3d(), begin_scissor_mode(), begin_shader_mode(), begin_texture_mode(), end_blend_mode(), end_mode_2d(), end_mode_3d(), end_scissor_mode(), end_texture_mode(), end_vr_stereo_mode(), set_camera_mode()

end_texture_mode *End texture mode*

Description

Ends drawing to render texture.

Usage

```
end_texture_mode()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void EndTextureMode(void);
```

See Also

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

Other mode functions: begin_blend_mode(), begin_mode_2d(), begin_mode_3d(), begin_scissor_mode(), begin_shader_mode(), begin_texture_mode(), end_blend_mode(), end_mode_2d(), end_mode_3d(), end_scissor_mode(), end_shader_mode(), end_vr_stereo_mode(), set_camera_mode()

end_vr_stereo_mode *End vr stereo mode*

Description

End stereo rendering (requires VR simulator).

Usage

```
end_vr_stereo_mode()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void EndVrStereoMode(void);
```

See Also

Other mode functions: begin_blend_mode(), begin_mode_2d(), begin_mode_3d(), begin_scissor_mode(), begin_shader_mode(), begin_texture_mode(), end_blend_mode(), end_mode_2d(), end_mode_3d(), end_scissor_mode(), end_shader_mode(), end_texture_mode(), set_camera_mode()

`export_image` *Export image*

Description

Export image data to file, returns true on success.

Usage

```
export_image(image, file_name)
```

Arguments

`image`	An image.
`file_name`	A string.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool ExportImage(Image image, const char *fileName);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

`export_image_as_code` *Export image as code*

Description

Export image as code file defining an array of bytes, returns true on success.

Usage

```
export_image_as_code(image, file_name)
```

Arguments

`image`	An image.
`file_name`	A string.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool ExportImageAsCode(Image image, const char *fileName);
```

See Also

`image()` for creating image objects.

Other image functions: `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text_ex()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_from_texture()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

export_wave *Export wave*

Description

Export wave data to file, returns true on success.

Usage

```
export_wave(wave, file_name)
```

Arguments

wave	A wave.
file_name	A string.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool ExportWave(Wave wave, const char *fileName);
```

See Also

wave() for creating wave objects.

Other wave functions: export_wave_as_code(), load_sound_from_wave(), load_wave(), unload_wave()

export_wave_as_code *Export wave as code*

Description

Export wave sample data to code (.h), returns true on success.

Usage

```
export_wave_as_code(wave, file_name)
```

Arguments

wave	A wave.
file_name	A string.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool ExportWaveAsCode(Wave wave, const char *fileName);
```

See Also

`wave()` for creating wave objects.

Other wave functions: `export_wave()`, `load_sound_from_wave()`, `load_wave()`, `unload_wave()`

---

fade *Fade*

Description

Get color with alpha applied, alpha goes from 0.0f to 1.0f.

Usage

```
fade(color, alpha)
```

Arguments

`color`	A color.
`alpha`	A number.

Value

A color

Note

This function has been auto-generated from the following Raylib function definition:

```
Color Fade(Color color, float alpha);
```

See Also

`color()` for creating color objects.

file_exists *File exists*

Description

Check if file exists.

Usage

```
file_exists(file_name)
```

Arguments

file_name	A string.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool FileExists(const char *fileName);
```

See Also

Other file functions: get_file_extension(), get_file_mod_time(), get_file_name_without_ext(), get_file_name(), is_file_dropped(), is_file_extension(), load_file_text()

gen_image_cellular *Gen image cellular*

Description

Generate image: cellular algorithm, bigger tileSize means bigger cells.

Usage

```
gen_image_cellular(width, height, tile_size)
```

Arguments

width	An integer.
height	An integer.
tile_size	An integer.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image GenImageCellular(int width, int height, int tileSize);
```

See Also

Other image functions: export_image_as_code(), export_image(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

gen_image_checked *Gen image checked*

Description

Generate image: checked.

Usage

```
gen_image_checked(width, height, checks_x, checks_y, col_1, col_2)
```

Arguments

width	An integer.
height	An integer.
checks_x	An integer.
checks_y	An integer.
col_1	A color.
col_2	A color.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image GenImageChecked(int width, int height, int checksX, int checksY, Color col1, Color col2);
```

See Also

`color()` for creating color objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text_ex()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_from_texture()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

---

gen_image_color *Gen image color*

Description

Generate image: plain color.

Usage

```
gen_image_color(width, height, color)
```

Arguments

width	An integer.
height	An integer.
color	A color.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image GenImageColor(int width, int height, Color color);
```

See Also

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other color functions: as_color(), color_alpha_blend(), color_alpha(), color_from_hsv(), color_from_normalized(), color_normalize(), color_to_hsv(), color_to_int(), color(), get_color(), get_image_color(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint()

---

gen_image_gradient_h *Gen image gradient h*

Description

Generate image: horizontal gradient.

Usage

```
gen_image_gradient_h(width, height, left, right)
```

Arguments

width	An integer.
height	An integer.
left	A color.
right	A color.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image GenImageGradientH(int width, int height, Color left, Color right);
```

See Also

`color()` for creating color objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text_ex()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_from_texture()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

---

gen_image_gradient_radial

Gen image gradient radial

Description

Generate image: radial gradient.

Usage

```
gen_image_gradient_radial(width, height, density, inner, outer)
```

Arguments

`width`	An integer.
`height`	An integer.
`density`	A number.
`inner`	A color.
`outer`	A color.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image GenImageGradientRadial(int width, int height, float density, Color inner, Color outer);
```

See Also

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

gen_image_gradient_v *Gen image gradient v*

Description

Generate image: vertical gradient.

Usage

```
gen_image_gradient_v(width, height, top, bottom)
```

Arguments

width	An integer.
height	An integer.
top	A color.
bottom	A color.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image GenImageGradientV(int width, int height, Color top, Color bottom);
```

See Also

`color()` for creating color objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text_ex()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_from_texture()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

---

gen_image_white_noise *Gen image white noise*

Description

Generate image: white noise.

Usage

```
gen_image_white_noise(width, height, factor)
```

Arguments

width	An integer.
height	An integer.
factor	A number.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image GenImageWhiteNoise(int width, int height, float factor);
```

See Also

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

gen_texture_mipmaps *Gen texture mipmaps*

Description

Generate GPU mipmaps for a texture.

Usage

```
gen_texture_mipmaps(texture)
```

Arguments

texture	A texture.

Note

This function has been auto-generated from the following Raylib function definition:

```
void GenTextureMipmaps(Texture2D *texture);
```

See Also

texture() for creating texture objects.

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

---

get_camera_matrix *Get camera matrix*

Description

Get camera transform matrix (view matrix).

Usage

```
get_camera_matrix(camera)
```

Arguments

camera	A camera_3d.

Value

A numeric matrix of 4 by 4

Note

This function has been auto-generated from the following Raylib function definition:

```
Matrix GetCameraMatrix(Camera camera);
```

See Also

camera_3d() for creating camera_3d objects.

Other camera functions: get_camera_matrix_2d(), set_camera_alt_control(), set_camera_mode(), set_camera_move_controls(), set_camera_pan_control(), set_camera_smooth_zoom_control(), update_camera()

get_camera_matrix_2d *Get camera matrix 2d*

Description

Get camera 2d transform matrix.

Usage

```
get_camera_matrix_2d(camera)
```

Arguments

camera	A camera_2d.

Value

A numeric matrix of 4 by 4

Note

This function has been auto-generated from the following Raylib function definition:

```
Matrix GetCameraMatrix2D(Camera2D camera);
```

See Also

`camera_2d()` for creating camera_2d objects.

Other camera functions: `get_camera_matrix()`, `set_camera_alt_control()`, `set_camera_mode()`, `set_camera_move_controls()`, `set_camera_pan_control()`, `set_camera_smooth_zoom_control()`, `update_camera()`

get_char_pressed *Get char pressed*

Description

Get char pressed (unicode), call it multiple times for chars queued, returns 0 when the queue is empty.

Usage

```
get_char_pressed()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetCharPressed(void);
```

See Also

Other key functions: get_key_pressed(), is_key_down(), is_key_pressed(), is_key_released(), is_key_up(), set_exit_key()

---

get_collision_rec *Get collision rec*

Description

Get collision rectangle for two rectangles collision.

Usage

```
get_collision_rec(rec_1, rec_2)
```

Arguments

rec_1	A rectangle.
rec_2	A rectangle.

Value

A rectangle

Note

This function has been auto-generated from the following Raylib function definition:

```
Rectangle GetCollisionRec(Rectangle rec1, Rectangle rec2);
```

See Also

rectangle() for creating rectangle objects.

Other collision functions: check_collision_box_sphere(), check_collision_boxes(), check_collision_circle_rec
check_collision_circles(), check_collision_lines(), check_collision_point_circle(),
check_collision_point_line(), check_collision_point_rec(), check_collision_point_triangle(),
check_collision_recs(), check_collision_spheres(), get_ray_collision_box(), get_ray_collision_mesh(),
get_ray_collision_model(), get_ray_collision_quad(), get_ray_collision_sphere(), get_ray_collision_tria

get_color *Get color*

Description

Get Color structure from hexadecimal value.

Usage

```
get_color(hex_value)
```

Arguments

hex_value	A non-negative integer.

Value

A color

Note

This function has been auto-generated from the following Raylib function definition:

```
Color GetColor(unsigned int hexValue);
```

See Also

Other color functions: as_color(), color_alpha_blend(), color_alpha(), color_from_hsv(), color_from_normalized(), color_normalize(), color_to_hsv(), color_to_int(), color(), gen_image_color(), get_image_color(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint()

get_current_monitor *Get current monitor*

Description

Get current connected monitor.

Usage

```
get_current_monitor()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetCurrentMonitor(void);
```

See Also

Other monitor functions: get_monitor_count(), get_monitor_height(), get_monitor_name(), get_monitor_physical_height(), get_monitor_physical_width(), get_monitor_position(), get_monitor_refresh_rate(), get_monitor_width(), set_window_monitor()

---

get_directory_path *Get directory path*

Description

Get full path for a given fileName with path (uses static string).

Usage

```
get_directory_path(file_path)
```

Arguments

file_path	A string.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
const char *GetDirectoryPath(const char *filePath);
```

See Also

Other directory functions: change_directory(), clear_directory_files(), directory_exists(), get_prev_directory_path(), get_working_directory()

get_file_extension *Get file extension*

Description

Get pointer to extension for a filename string (includes dot: '.png').

Usage

```
get_file_extension(file_name)
```

Arguments

file_name	A string.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
const char *GetFileExtension(const char *fileName);
```

See Also

Other file functions: file_exists(), get_file_mod_time(), get_file_name_without_ext(), get_file_name(), is_file_dropped(), is_file_extension(), load_file_text()

get_file_mod_time *Get file mod time*

Description

Get file modification time (last write time).

Usage

```
get_file_mod_time(file_name)
```

Arguments

file_name	A string.

Value

A long

Note

This function has been auto-generated from the following Raylib function definition:

```
long GetFileModTime(const char *fileName);
```

See Also

Other file functions: file_exists(), get_file_extension(), get_file_name_without_ext(), get_file_name(), is_file_dropped(), is_file_extension(), load_file_text()

---

get_file_name *Get file name*

Description

Get pointer to filename for a path string.

Usage

```
get_file_name(file_path)
```

Arguments

file_path	A string.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
const char *GetFileName(const char *filePath);
```

See Also

Other file functions: file_exists(), get_file_extension(), get_file_mod_time(), get_file_name_without_ext(), is_file_dropped(), is_file_extension(), load_file_text()

get_file_name_without_ext *Get file name without ext*

Description

Get filename string without extension (uses static string).

Usage

```
get_file_name_without_ext(file_path)
```

Arguments

file_path	A string.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
const char *GetFileNameWithoutExt(const char *filePath);
```

See Also

Other file functions: file_exists(), get_file_extension(), get_file_mod_time(), get_file_name(), is_file_dropped(), is_file_extension(), load_file_text()

get_font_default *Get font default*

Description

Get the default Font.

Usage

```
get_font_default()
```

Value

A font

Note

This function has been auto-generated from the following Raylib function definition:

```
Font GetFontDefault(void);
```

See Also

Other font functions: load_font_ex(), load_font_from_image(), load_font(), unload_font_data(), unload_font()

---

get_fps *Get fps*

Description

Get current FPS.

Usage

```
get_fps()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetFPS(void);
```

See Also

Other fps functions: draw_fps(), set_target_fps()

get_frame_time *Get frame time*

Description

Get time in seconds for last frame drawn (delta time).

Usage

```
get_frame_time()
```

Value

A number

Note

This function has been auto-generated from the following Raylib function definition:

```
float GetFrameTime(void);
```

get_gamepad_axis_count *Get gamepad axis count*

Description

Get gamepad axis count for a gamepad.

Usage

```
get_gamepad_axis_count(gamepad)
```

Arguments

gamepad	An integer.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetGamepadAxisCount(int gamepad);
```

See Also

Other gamepad functions: get_gamepad_axis_movement(), get_gamepad_button_pressed(), get_gamepad_name(), is_gamepad_available(), is_gamepad_button_down(), is_gamepad_button_pressed(), is_gamepad_button_released(), is_gamepad_button_up(), set_gamepad_mappings()

`get_gamepad_axis_movement` *Get gamepad axis movement*

Description

Get axis movement value for a gamepad axis.

Usage

```
get_gamepad_axis_movement(gamepad, axis)
```

Arguments

gamepad	An integer.
axis	An integer.

Value

A number

Note

This function has been auto-generated from the following Raylib function definition:

```
float GetGamepadAxisMovement(int gamepad, int axis);
```

See Also

Other gamepad functions: get_gamepad_axis_count(), get_gamepad_button_pressed(), get_gamepad_name(), is_gamepad_available(), is_gamepad_button_down(), is_gamepad_button_pressed(), is_gamepad_button_releas
is_gamepad_button_up(), set_gamepad_mappings()

get_gamepad_button_pressed *Get gamepad button pressed*

Description

Get the last gamepad button pressed.

Usage

```
get_gamepad_button_pressed()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetGamepadButtonPressed(void);
```

See Also

Other gamepad functions: get_gamepad_axis_count(), get_gamepad_axis_movement(), get_gamepad_name(), is_gamepad_available(), is_gamepad_button_down(), is_gamepad_button_pressed(), is_gamepad_button_releas
is_gamepad_button_up(), set_gamepad_mappings()

get_gamepad_name *Get gamepad name*

Description

Get gamepad internal name id.

Usage

```
get_gamepad_name(gamepad)
```

Arguments

gamepad	An integer.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
const char *GetGamepadName(int gamepad);
```

See Also

Other gamepad functions: get_gamepad_axis_count(), get_gamepad_axis_movement(), get_gamepad_button_presse
is_gamepad_available(), is_gamepad_button_down(), is_gamepad_button_pressed(), is_gamepad_button_releas
is_gamepad_button_up(), set_gamepad_mappings()

---

get_gesture_detected *Get gesture detected*

Description

Get latest detected gesture.

Usage

```
get_gesture_detected()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetGestureDetected(void);
```

---

get_gesture_drag_angle *Get gesture drag angle*

Description

Get gesture drag angle.

Usage

```
get_gesture_drag_angle()
```

Value

A number

Note

This function has been auto-generated from the following Raylib function definition:

```
float GetGestureDragAngle(void);
```

`get_gesture_drag_vector`
Get gesture drag vector

Description

Get gesture drag vector.

Usage

```
get_gesture_drag_vector()
```

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 GetGestureDragVector(void);
```

`get_gesture_hold_duration`
Get gesture hold duration

Description

Get gesture hold time in milliseconds.

Usage

```
get_gesture_hold_duration()
```

Value

A number

Note

This function has been auto-generated from the following Raylib function definition:

```
float GetGestureHoldDuration(void);
```

`get_gesture_pinch_angle` *Get gesture pinch angle*

Description

Get gesture pinch angle.

Usage

```
get_gesture_pinch_angle()
```

Value

A number

Note

This function has been auto-generated from the following Raylib function definition:

```
float GetGesturePinchAngle(void);
```

`get_gesture_pinch_vector` *Get gesture pinch vector*

Description

Get gesture pinch delta.

Usage

```
get_gesture_pinch_vector()
```

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 GetGesturePinchVector(void);
```

`get_glyph_atlas_rec` *Get glyph atlas rec*

Description

Get glyph rectangle in font atlas for a codepoint (unicode character), fallback to '?' if not found.

Usage

```
get_glyph_atlas_rec(font, codepoint)
```

Arguments

`font`	A font.
`codepoint`	An integer.

Value

A rectangle

Note

This function has been auto-generated from the following Raylib function definition:

```
Rectangle GetGlyphAtlasRec(Font font, int codepoint);
```

See Also

`font()` for creating font objects.

get_glyph_index *Get glyph index*

Description

Get glyph index position in font for a codepoint (unicode character), fallback to '?' if not found.

Usage

```
get_glyph_index(font, codepoint)
```

Arguments

font	A font.
codepoint	An integer.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetGlyphIndex(Font font, int codepoint);
```

See Also

font() for creating font objects.

get_glyph_info *Get glyph info*

Description

Get glyph font info data for a codepoint (unicode character), fallback to '?' if not found.

Usage

```
get_glyph_info(font, codepoint)
```

Arguments

font	A font.
codepoint	An integer.

Value

A glyph_info

Note

This function has been auto-generated from the following Raylib function definition:

```
GlyphInfo GetGlyphInfo(Font font, int codepoint);
```

See Also

`font()` for creating font objects.

---

get_image_alpha_border

Get image alpha border

Description

Get image alpha border rectangle.

Usage

```
get_image_alpha_border(image, threshold)
```

Arguments

`image`	An image.
`threshold`	A number.

Value

A rectangle

Note

This function has been auto-generated from the following Raylib function definition:

```
Rectangle GetImageAlphaBorder(Image image, float threshold);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

get_image_color *Get image color*

Description

Get image pixel color at (x, y) position.

Usage

```
get_image_color(image, x, y)
```

Arguments

image	An image.
x	An integer.
y	An integer.

Value

A color

Note

This function has been auto-generated from the following Raylib function definition:

```
Color GetImageColor(Image image, int x, int y);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other color functions: as_color(), color_alpha_blend(), color_alpha(), color_from_hsv(), color_from_normalized(), color_normalize(), color_to_hsv(), color_to_int(), color(), gen_image_color(), get_color(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint()

---

get_key_pressed *Get key pressed*

Description

Get key pressed (keycode), call it multiple times for keys queued, returns 0 when the queue is empty.

Usage

```
get_key_pressed()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetKeyPressed(void);
```

See Also

Other key functions: get_char_pressed(), is_key_down(), is_key_pressed(), is_key_released(), is_key_up(), set_exit_key()

get_model_bounding_box *Get model bounding box*

Description

Compute model bounding box limits (considers all meshes).

Usage

```
get_model_bounding_box(model)
```

Arguments

model	A model.

Value

A bounding_box

Note

This function has been auto-generated from the following Raylib function definition:

```
BoundingBox GetModelBoundingBox(Model model);
```

See Also

`model()` for creating model objects.

Other model functions: `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `get_ray_collision_model()`, `load_model()`, `unload_model()`

get_monitor_count *Get monitor count*

Description

Get number of connected monitors.

Usage

```
get_monitor_count()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetMonitorCount(void);
```

See Also

Other monitor functions: `get_current_monitor()`, `get_monitor_height()`, `get_monitor_name()`, `get_monitor_physical_height()`, `get_monitor_physical_width()`, `get_monitor_position()`, `get_monitor_refresh_rate()`, `get_monitor_width()`, `set_window_monitor()`

---

get_monitor_height *Get monitor height*

Description

Get specified monitor height (max available by monitor).

Usage

```
get_monitor_height(monitor)
```

Arguments

monitor	An integer.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetMonitorHeight(int monitor);
```

See Also

Other monitor functions: `get_current_monitor()`, `get_monitor_count()`, `get_monitor_name()`, `get_monitor_physical_height()`, `get_monitor_physical_width()`, `get_monitor_position()`, `get_monitor_refresh_rate()`, `get_monitor_width()`, `set_window_monitor()`

get_monitor_name *Get monitor name*

Description

Get the human-readable, UTF-8 encoded name of the primary monitor.

Usage

```
get_monitor_name(monitor)
```

Arguments

monitor	An integer.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
const char *GetMonitorName(int monitor);
```

See Also

Other monitor functions: get_current_monitor(), get_monitor_count(), get_monitor_height(), get_monitor_physical_height(), get_monitor_physical_width(), get_monitor_position(), get_monitor_refresh_rate(), get_monitor_width(), set_window_monitor()

get_monitor_physical_height *Get monitor physical height*

Description

Get specified monitor physical height in millimetres.

Usage

```
get_monitor_physical_height(monitor)
```

Arguments

monitor	An integer.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetMonitorPhysicalHeight(int monitor);
```

See Also

Other monitor functions: get_current_monitor(), get_monitor_count(), get_monitor_height(), get_monitor_name(), get_monitor_physical_width(), get_monitor_position(), get_monitor_refresh_rate(), get_monitor_width(), set_window_monitor()

---

get_monitor_physical_width

Get monitor physical width

Description

Get specified monitor physical width in millimetres.

Usage

```
get_monitor_physical_width(monitor)
```

Arguments

monitor	An integer.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetMonitorPhysicalWidth(int monitor);
```

See Also

Other monitor functions: get_current_monitor(), get_monitor_count(), get_monitor_height(), get_monitor_name(), get_monitor_physical_height(), get_monitor_position(), get_monitor_refresh_rate(), get_monitor_width(), set_window_monitor()

get_monitor_position *Get monitor position*

Description

Get specified monitor position.

Usage

```
get_monitor_position(monitor)
```

Arguments

monitor	An integer.

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 GetMonitorPosition(int monitor);
```

See Also

Other monitor functions: get_current_monitor(), get_monitor_count(), get_monitor_height(), get_monitor_name(), get_monitor_physical_height(), get_monitor_physical_width(), get_monitor_refresh_ra get_monitor_width(), set_window_monitor()

get_monitor_refresh_rate *Get monitor refresh rate*

Description

Get specified monitor refresh rate.

Usage

```
get_monitor_refresh_rate(monitor)
```

Arguments

monitor	An integer.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetMonitorRefreshRate(int monitor);
```

See Also

Other monitor functions: get_current_monitor(), get_monitor_count(), get_monitor_height(), get_monitor_name(), get_monitor_physical_height(), get_monitor_physical_width(), get_monitor_position(), get_monitor_width(), set_window_monitor()

---

get_monitor_width *Get monitor width*

Description

Get specified monitor width (max available by monitor).

Usage

```
get_monitor_width(monitor)
```

Arguments

monitor	An integer.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetMonitorWidth(int monitor);
```

See Also

Other monitor functions: get_current_monitor(), get_monitor_count(), get_monitor_height(), get_monitor_name(), get_monitor_physical_height(), get_monitor_physical_width(), get_monitor_position(), get_monitor_refresh_rate(), set_window_monitor()

get_mouse_delta *Get mouse delta*

Description

Get mouse delta between frames.

Usage

```
get_mouse_delta()
```

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 GetMouseDelta(void);
```

See Also

Other mouse functions: `get_mouse_position()`, `get_mouse_wheel_move()`, `get_mouse_x()`, `get_mouse_y()`, `is_mouse_button_down()`, `is_mouse_button_pressed()`, `is_mouse_button_released()`, `is_mouse_button_up()`, `set_mouse_cursor()`, `set_mouse_offset()`, `set_mouse_position()`, `set_mouse_scale()`

get_mouse_position *Get mouse position*

Description

Get mouse position XY.

Usage

```
get_mouse_position()
```

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 GetMousePosition(void);
```

See Also

Other mouse functions: get_mouse_delta(), get_mouse_wheel_move(), get_mouse_x(), get_mouse_y(), is_mouse_button_down(), is_mouse_button_pressed(), is_mouse_button_released(), is_mouse_button_up(), set_mouse_cursor(), set_mouse_offset(), set_mouse_position(), set_mouse_scale()

---

get_mouse_wheel_move *Get mouse wheel move*

Description

Get mouse wheel movement Y.

Usage

```
get_mouse_wheel_move()
```

Value

A number

Note

This function has been auto-generated from the following Raylib function definition:

```
float GetMouseWheelMove(void);
```

See Also

Other mouse functions: get_mouse_delta(), get_mouse_position(), get_mouse_x(), get_mouse_y(), is_mouse_button_down(), is_mouse_button_pressed(), is_mouse_button_released(), is_mouse_button_up(), set_mouse_cursor(), set_mouse_offset(), set_mouse_position(), set_mouse_scale()

get_mouse_x *Get mouse x*

Description

Get mouse position X.

Usage

```
get_mouse_x()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetMouseX(void);
```

See Also

Other mouse functions: `get_mouse_delta()`, `get_mouse_position()`, `get_mouse_wheel_move()`, `get_mouse_y()`, `is_mouse_button_down()`, `is_mouse_button_pressed()`, `is_mouse_button_released()`, `is_mouse_button_up()`, `set_mouse_cursor()`, `set_mouse_offset()`, `set_mouse_position()`, `set_mouse_scale()`

get_mouse_y *Get mouse y*

Description

Get mouse position Y.

Usage

```
get_mouse_y()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetMouseY(void);
```

See Also

Other mouse functions: get_mouse_delta(), get_mouse_position(), get_mouse_wheel_move(), get_mouse_x(), is_mouse_button_down(), is_mouse_button_pressed(), is_mouse_button_released(), is_mouse_button_up(), set_mouse_cursor(), set_mouse_offset(), set_mouse_position(), set_mouse_scale()

---

get_music_time_length *Get music time length*

Description

Get music time length (in seconds).

Usage

```
get_music_time_length(music)
```

Arguments

music	A music.

Value

A number

Note

This function has been auto-generated from the following Raylib function definition:

```
float GetMusicTimeLength(Music music);
```

See Also

music() for creating music objects.

Other music functions: get_music_time_played(), is_music_stream_playing(), load_music_stream(), pause_music_stream(), play_music_stream(), resume_music_stream(), seek_music_stream(), set_music_pitch(), set_music_volume(), stop_music_stream(), unload_music_stream(), update_music_stream()

get_music_time_played *Get music time played*

Description

Get current music time played (in seconds).

Usage

```
get_music_time_played(music)
```

Arguments

music	A music.

Value

A number

Note

This function has been auto-generated from the following Raylib function definition:

```
float GetMusicTimePlayed(Music music);
```

See Also

music() for creating music objects.

Other music functions: get_music_time_length(), is_music_stream_playing(), load_music_stream(), pause_music_stream(), play_music_stream(), resume_music_stream(), seek_music_stream(), set_music_pitch(), set_music_volume(), stop_music_stream(), unload_music_stream(), update_music_stream()

get_pixel_data_size *Get pixel data size*

Description

Get pixel data size in bytes for certain format.

Usage

```
get_pixel_data_size(width, height, format)
```

Arguments

width	An integer.
height	An integer.
format	An integer.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetPixelDataSize(int width, int height, int format);
```

---

get_prev_directory_path

Get prev directory path

Description

Get previous directory path for a given path (uses static string).

Usage

```
get_prev_directory_path(dir_path)
```

Arguments

dir_path	A string.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
const char *GetPrevDirectoryPath(const char *dirPath);
```

See Also

Other directory functions: change_directory(), clear_directory_files(), directory_exists(), get_directory_path(), get_working_directory()

get_random_value *Get random value*

Description

Get a random value between min and max (both included).

Usage

```
get_random_value(min, max)
```

Arguments

min	An integer.
max	An integer.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetRandomValue(int min, int max);
```

get_ray_collision_box *Get ray collision box*

Description

Get collision info between ray and box.

Usage

```
get_ray_collision_box(ray, box)
```

Arguments

ray	A ray.
box	A bounding_box.

Value

A ray_collision

Note

This function has been auto-generated from the following Raylib function definition:

```
RayCollision GetRayCollisionBox(Ray ray, BoundingBox box);
```

See Also

Other collision functions: check_collision_box_sphere(), check_collision_boxes(), check_collision_circle_rec
check_collision_circles(), check_collision_lines(), check_collision_point_circle(),
check_collision_point_line(), check_collision_point_rec(), check_collision_point_triangle(),
check_collision_recs(), check_collision_spheres(), get_collision_rec(), get_ray_collision_mesh(),
get_ray_collision_model(), get_ray_collision_quad(), get_ray_collision_sphere(), get_ray_collision_tria

---

get_ray_collision_mesh

Get ray collision mesh

Description

Get collision info between ray and mesh.

Usage

```
get_ray_collision_mesh(ray, mesh, transform)
```

Arguments

ray	A ray.
mesh	A mesh.
transform	A numeric matrix of 4 by 4.

Value

A ray_collision

Note

This function has been auto-generated from the following Raylib function definition:

```
RayCollision GetRayCollisionMesh(Ray ray, Mesh mesh, Matrix transform);
```

See Also

mesh() for creating mesh objects.

Other collision functions: check_collision_box_sphere(), check_collision_boxes(), check_collision_circle_rec
check_collision_circles(), check_collision_lines(), check_collision_point_circle(),
check_collision_point_line(), check_collision_point_rec(), check_collision_point_triangle(),
check_collision_recs(), check_collision_spheres(), get_collision_rec(), get_ray_collision_box(),
get_ray_collision_model(), get_ray_collision_quad(), get_ray_collision_sphere(), get_ray_collision_tria

get_ray_collision_model *Get ray collision model*

Description

Get collision info between ray and model.

Usage

```
get_ray_collision_model(ray, model)
```

Arguments

ray	A ray.
model	A model.

Value

A ray_collision

Note

This function has been auto-generated from the following Raylib function definition:

```
RayCollision GetRayCollisionModel(Ray ray, Model model);
```

See Also

model() for creating model objects.

Other collision functions: check_collision_box_sphere(), check_collision_boxes(), check_collision_circle_rec
check_collision_circles(), check_collision_lines(), check_collision_point_circle(),
check_collision_point_line(), check_collision_point_rec(), check_collision_point_triangle(),
check_collision_recs(), check_collision_spheres(), get_collision_rec(), get_ray_collision_box(),
get_ray_collision_mesh(), get_ray_collision_quad(), get_ray_collision_sphere(), get_ray_collision_trian

Other model functions: draw_model_ex(), draw_model_wires(), draw_model(), get_model_bounding_box(),
load_model(), unload_model()

`get_ray_collision_quad` *Get ray collision quad*

Description

Get collision info between ray and quad.

Usage

```
get_ray_collision_quad(ray, p_1, p_2, p_3, p_4)
```

Arguments

`ray`	A ray.
`p_1`	A numeric vector of length 3.
`p_2`	A numeric vector of length 3.
`p_3`	A numeric vector of length 3.
`p_4`	A numeric vector of length 3.

Value

A ray_collision

Note

This function has been auto-generated from the following Raylib function definition:

```
RayCollision GetRayCollisionQuad(Ray ray, Vector3 p1, Vector3 p2, Vector3 p3, Vector3 p4);
```

See Also

Other collision functions: `check_collision_box_sphere()`, `check_collision_boxes()`, `check_collision_circle_rec`
`check_collision_circles()`, `check_collision_lines()`, `check_collision_point_circle()`,
`check_collision_point_line()`, `check_collision_point_rec()`, `check_collision_point_triangle()`,
`check_collision_recs()`, `check_collision_spheres()`, `get_collision_rec()`, `get_ray_collision_box()`,
`get_ray_collision_mesh()`, `get_ray_collision_model()`, `get_ray_collision_sphere()`, `get_ray_collision_tria`

`get_ray_collision_sphere` *Get ray collision sphere*

Description

Get collision info between ray and sphere.

Usage

```
get_ray_collision_sphere(ray, center, radius)
```

Arguments

`ray`	A ray.
`center`	A numeric vector of length 3.
`radius`	A number.

Value

A ray_collision

Note

This function has been auto-generated from the following Raylib function definition:

```
RayCollision GetRayCollisionSphere(Ray ray, Vector3 center, float radius);
```

See Also

Other collision functions: `check_collision_box_sphere()`, `check_collision_boxes()`, `check_collision_circle_rec`
`check_collision_circles()`, `check_collision_lines()`, `check_collision_point_circle()`,
`check_collision_point_line()`, `check_collision_point_rec()`, `check_collision_point_triangle()`,
`check_collision_recs()`, `check_collision_spheres()`, `get_collision_rec()`, `get_ray_collision_box()`,
`get_ray_collision_mesh()`, `get_ray_collision_model()`, `get_ray_collision_quad()`, `get_ray_collision_triang`

`get_ray_collision_triangle` *Get ray collision triangle*

Description

Get collision info between ray and triangle.

Usage

```
get_ray_collision_triangle(ray, p_1, p_2, p_3)
```

Arguments

`ray`	A ray.
`p_1`	A numeric vector of length 3.
`p_2`	A numeric vector of length 3.
`p_3`	A numeric vector of length 3.

Value

A ray_collision

Note

This function has been auto-generated from the following Raylib function definition:

```
RayCollision GetRayCollisionTriangle(Ray ray, Vector3 p1, Vector3 p2, Vector3 p3);
```

See Also

Other collision functions: `check_collision_box_sphere()`, `check_collision_boxes()`, `check_collision_circle_rec`
`check_collision_circles()`, `check_collision_lines()`, `check_collision_point_circle()`,
`check_collision_point_line()`, `check_collision_point_rec()`, `check_collision_point_triangle()`,
`check_collision_recs()`, `check_collision_spheres()`, `get_collision_rec()`, `get_ray_collision_box()`,
`get_ray_collision_mesh()`, `get_ray_collision_model()`, `get_ray_collision_quad()`, `get_ray_collision_sphere`

get_screen_height *Get screen height*

Description

Get current screen height.

Usage

```
get_screen_height()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetScreenHeight(void);
```

See Also

Other screen functions: `get_screen_to_world_2d()`, `get_screen_width()`, `get_world_to_screen_2d()`, `get_world_to_screen_ex()`, `get_world_to_screen()`, `is_cursor_on_screen()`, `load_image_from_screen()`

get_screen_to_world_2d *Get screen to world 2d*

Description

Get the world space position for a 2d camera screen space position.

Usage

```
get_screen_to_world_2d(position, camera)
```

Arguments

position	A numeric vector of length 2.
camera	A camera_2d.

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 GetScreenToWorld2D(Vector2 position, Camera2D camera);
```

See Also

camera_2d() for creating camera_2d objects.

Other screen functions: get_screen_height(), get_screen_width(), get_world_to_screen_2d(), get_world_to_screen_ex(), get_world_to_screen(), is_cursor_on_screen(), load_image_from_screen()

Other world functions: get_world_to_screen_2d(), get_world_to_screen_ex(), get_world_to_screen()

---

get_screen_width *Get screen width*

Description

Get current screen width.

Usage

```
get_screen_width()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetScreenWidth(void);
```

See Also

Other screen functions: get_screen_height(), get_screen_to_world_2d(), get_world_to_screen_2d(), get_world_to_screen_ex(), get_world_to_screen(), is_cursor_on_screen(), load_image_from_screen()

get_shader_location *Get shader location*

Description

Get shader uniform location.

Usage

```
get_shader_location(shader, uniform_name)
```

Arguments

shader	A shader.
uniform_name	A string.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetShaderLocation(Shader shader, const char *uniformName);
```

See Also

shader() for creating shader objects.

Other shader functions: begin_shader_mode(), end_shader_mode(), get_shader_location_attrib(), load_shader_from_memory(), load_shader(), set_shader_value_matrix(), set_shader_value_texture(), set_shader_value(), unload_shader()

get_shader_location_attrib *Get shader location attrib*

Description

Get shader attribute location.

Usage

```
get_shader_location_attrib(shader, attrib_name)
```

Arguments

shader	A shader.
attrib_name	A string.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetShaderLocationAttrib(Shader shader, const char *attribName);
```

See Also

shader() for creating shader objects.

Other shader functions: begin_shader_mode(), end_shader_mode(), get_shader_location(), load_shader_from_memory(), load_shader(), set_shader_value_matrix(), set_shader_value_texture(), set_shader_value(), unload_shader()

---

get_sounds_playing *Get sounds playing*

Description

Get number of sounds playing in the multichannel.

Usage

```
get_sounds_playing()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetSoundsPlaying(void);
```

get_time *Get time*

Description

Get elapsed time in seconds since InitWindow().

Usage

```
get_time()
```

Value

A double

Note

This function has been auto-generated from the following Raylib function definition:

```
double GetTime(void);
```

get_touch_point_count *Get touch point count*

Description

Get number of touch points.

Usage

```
get_touch_point_count()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetTouchPointCount(void);
```

See Also

Other touch functions: get_touch_point_id(), get_touch_position(), get_touch_x(), get_touch_y()

get_touch_point_id *Get touch point id*

Description

Get touch point identifier for given index.

Usage

```
get_touch_point_id(index)
```

Arguments

index	An integer.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetTouchPointId(int index);
```

See Also

Other touch functions: `get_touch_point_count()`, `get_touch_position()`, `get_touch_x()`, `get_touch_y()`

get_touch_position *Get touch position*

Description

Get touch position XY for a touch point index (relative to screen size).

Usage

```
get_touch_position(index)
```

Arguments

index	An integer.

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 GetTouchPosition(int index);
```

See Also

Other touch functions: `get_touch_point_count()`, `get_touch_point_id()`, `get_touch_x()`, `get_touch_y()`

---

get_touch_x *Get touch x*

Description

Get touch position X for touch point 0 (relative to screen size).

Usage

```
get_touch_x()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetTouchX(void);
```

See Also

Other touch functions: `get_touch_point_count()`, `get_touch_point_id()`, `get_touch_position()`, `get_touch_y()`

get_touch_y *Get touch y*

Description

Get touch position Y for touch point 0 (relative to screen size).

Usage

```
get_touch_y()
```

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int GetTouchY(void);
```

See Also

Other touch functions: `get_touch_point_count()`, `get_touch_point_id()`, `get_touch_position()`, `get_touch_x()`

get_window_position *Get window position*

Description

Get window position XY on monitor.

Usage

```
get_window_position()
```

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 GetWindowPosition(void);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

`get_window_scale_dpi` *Get window scale dpi*

Description

Get window scale DPI factor.

Usage

```
get_window_scale_dpi()
```

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 GetWindowScaleDPI(void);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

get_working_directory *Get working directory*

Description

Get current working directory (uses static string).

Usage

```
get_working_directory()
```

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
const char *GetWorkingDirectory(void);
```

See Also

Other directory functions: change_directory(), clear_directory_files(), directory_exists(), get_directory_path(), get_prev_directory_path()

get_world_to_screen *Get world to screen*

Description

Get the screen space position for a 3d world space position.

Usage

```
get_world_to_screen(position, camera)
```

Arguments

position	A numeric vector of length 3.
camera	A camera_3d.

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 GetWorldToScreen(Vector3 position, Camera camera);
```

See Also

camera_3d() for creating camera_3d objects.

Other world functions: get_screen_to_world_2d(), get_world_to_screen_2d(), get_world_to_screen_ex()

Other screen functions: get_screen_height(), get_screen_to_world_2d(), get_screen_width(), get_world_to_screen_2d(), get_world_to_screen_ex(), is_cursor_on_screen(), load_image_from_screen()

---

get_world_to_screen_2d

Get world to screen 2d

Description

Get the screen space position for a 2d camera world space position.

Usage

```
get_world_to_screen_2d(position, camera)
```

Arguments

position	A numeric vector of length 2.
camera	A camera_2d.

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 GetWorldToScreen2D(Vector2 position, Camera2D camera);
```

See Also

camera_2d() for creating camera_2d objects.

Other world functions: get_screen_to_world_2d(), get_world_to_screen_ex(), get_world_to_screen()

Other screen functions: get_screen_height(), get_screen_to_world_2d(), get_screen_width(), get_world_to_screen_ex(), get_world_to_screen(), is_cursor_on_screen(), load_image_from_screen()

`get_world_to_screen_ex` *Get world to screen ex*

Description

Get size position for a 3d world space position.

Usage

```
get_world_to_screen_ex(position, camera, width, height)
```

Arguments

`position`	A numeric vector of length 3.
`camera`	A camera_3d.
`width`	An integer.
`height`	An integer.

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 GetWorldToScreenEx(Vector3 position, Camera camera, int width, int height);
```

See Also

`camera_3d()` for creating camera_3d objects.

Other world functions: `get_screen_to_world_2d()`, `get_world_to_screen_2d()`, `get_world_to_screen()`

Other screen functions: `get_screen_height()`, `get_screen_to_world_2d()`, `get_screen_width()`, `get_world_to_screen_2d()`, `get_world_to_screen()`, `is_cursor_on_screen()`, `load_image_from_screen()`

`glyph_info` *Glyph info*

Description

Create a new glyph_info object.

Usage

```
glyph_info(value, offset_x, offset_y, advance_x, image)
```

Arguments

`value`	An integer. Character value (Unicode).
`offset_x`	An integer. Character offset X when drawing.
`offset_y`	An integer. Character offset Y when drawing.
`advance_x`	An integer. Character advance position X.
`image`	An image. Character image data.

Value

A glyph_info

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct GlyphInfo {
    int value;              // Character value (Unicode)
    int offsetX;            // Character offset X when drawing
    int offsetY;            // Character offset Y when drawing
    int advanceX;           // Character advance position X
    Image image;            // Character image data
} GlyphInfo;
```

hide_cursor *Hide cursor*

Description

Hides cursor.

Usage

```
hide_cursor()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void HideCursor(void);
```

See Also

Other cursor functions: `disable_cursor()`, `enable_cursor()`, `is_cursor_hidden()`, `is_cursor_on_screen()`, `set_mouse_cursor()`, `show_cursor()`

image_alpha_clear *Image alpha clear*

Description

Clear alpha channel to desired color.

Usage

```
image_alpha_clear(image, color, threshold)
```

Arguments

`image`	An image.
`color`	A color.
`threshold`	A number.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageAlphaClear(Image *image, Color color, float threshold);
```

See Also

image() for creating image objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

image_alpha_crop *Image alpha crop*

Description

Crop image depending on alpha value.

Usage

```
image_alpha_crop(image, threshold)
```

Arguments

image	An image.
threshold	A number.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageAlphaCrop(Image *image, float threshold);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

image_alpha_mask *Image alpha mask*

Description

Apply alpha mask to image.

Usage

```
image_alpha_mask(image, alpha_mask)
```

Arguments

image	An image.
alpha_mask	An image.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageAlphaMask(Image *image, Image alphaMask);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_premultiply(), image_clear_background(),

image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

`image_alpha_premultiply` *Image alpha premultiply*

Description

Premultiply alpha channel.

Usage

```
image_alpha_premultiply(image)
```

Arguments

`image`	An image.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageAlphaPremultiply(Image *image);
```

See Also

`image()` for creating image objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`,

image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

`image_clear_background` *Image clear background*

Description

Clear image background with given color.

Usage

```
image_clear_background(dst, color)
```

Arguments

`dst`	An image.
`color`	A color.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageClearBackground(Image *dst, Color color);
```

See Also

`image()` for creating image objects.

`color()` for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

image_color_brightness *Image color brightness*

Description

Modify image color: brightness (-255 to 255).

Usage

```
image_color_brightness(image, brightness)
```

Arguments

image	An image.
brightness	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageColorBrightness(Image *image, int brightness);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other color functions: as_color(), color_alpha_blend(), color_alpha(), color_from_hsv(), color_from_normalized(), color_normalize(), color_to_hsv(), color_to_int(), color(), gen_image_color(), get_color(), get_image_color(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint()

`image_color_contrast` *Image color contrast*

Description

Modify image color: contrast (-100 to 100).

Usage

```
image_color_contrast(image, contrast)
```

Arguments

`image`	An image.
`contrast`	A number.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageColorContrast(Image *image, float contrast);
```

See Also

`image()` for creating image objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text_ex()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_from_texture()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

Other color functions: `as_color()`, `color_alpha_blend()`, `color_alpha()`, `color_from_hsv()`, `color_from_normalized()`, `color_normalize()`, `color_to_hsv()`, `color_to_int()`, `color()`, `gen_image_color()`, `get_color()`, `get_image_color()`, `image_color_brightness()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`

`image_color_grayscale` *Image color grayscale*

Description

Modify image color: grayscale.

Usage

```
image_color_grayscale(image)
```

Arguments

`image`	An image.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageColorGrayscale(Image *image);
```

See Also

`image()` for creating image objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text_ex()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_from_texture()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

Other color functions: `as_color()`, `color_alpha_blend()`, `color_alpha()`, `color_from_hsv()`, `color_from_normalized()`, `color_normalize()`, `color_to_hsv()`, `color_to_int()`, `color()`, `gen_image_color()`, `get_color()`, `get_image_color()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`

`image_color_invert` *Image color invert*

Description

Modify image color: invert.

Usage

```
image_color_invert(image)
```

Arguments

`image`	An image.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageColorInvert(Image *image);
```

See Also

`image()` for creating image objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text_ex()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_from_texture()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

Other color functions: `as_color()`, `color_alpha_blend()`, `color_alpha()`, `color_from_hsv()`, `color_from_normalized()`, `color_normalize()`, `color_to_hsv()`, `color_to_int()`, `color()`, `gen_image_color()`, `get_color()`, `get_image_color()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_replace()`, `image_color_tint()`

image_color_replace *Image color replace*

Description

Modify image color: replace color.

Usage

```
image_color_replace(image, color, replace)
```

Arguments

image	An image.
color	A color.
replace	A color.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageColorReplace(Image *image, Color color, Color replace);
```

See Also

image() for creating image objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other color functions: as_color(), color_alpha_blend(), color_alpha(), color_from_hsv(), color_from_normalized(), color_normalize(), color_to_hsv(), color_to_int(), color(), gen_image_color(), get_color(), get_image_color(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_tint()

`image_color_tint` *Image color tint*

Description

Modify image color: tint.

Usage

```
image_color_tint(image, color)
```

Arguments

`image`	An image.
`color`	A color.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageColorTint(Image *image, Color color);
```

See Also

`image()` for creating image objects.

`color()` for creating color objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text_ex()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_from_texture()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

Other color functions: `as_color()`, `color_alpha_blend()`, `color_alpha()`, `color_from_hsv()`, `color_from_normalized()`, `color_normalize()`, `color_to_hsv()`, `color_to_int()`, `color()`, `gen_image_color()`, `get_color()`, `get_image_color()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`

image_copy *Image copy*

Description

Create an image duplicate (useful for transformations).

Usage

```
image_copy(image)
```

Arguments

image	An image.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image ImageCopy(Image image);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

image_crop *Image crop*

Description

Crop an image to a defined rectangle.

Usage

```
image_crop(image, crop)
```

Arguments

image	An image.
crop	A rectangle.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageCrop(Image *image, Rectangle crop);
```

See Also

image() for creating image objects.

rectangle() for creating rectangle objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

image_dither *Image dither*

Description

Dither image data to 16bpp or lower (Floyd-Steinberg dithering).

Usage

```
image_dither(image, r_bpp, g_bpp, b_bpp, a_bpp)
```

Arguments

image	An image.
r_bpp	An integer.
g_bpp	An integer.
b_bpp	An integer.
a_bpp	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageDither(Image *image, int rBpp, int gBpp, int bBpp, int aBpp);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

image_draw *Image draw*

Description

Draw a source image within a destination image (tint applied to source).

Usage

```
image_draw(dst, src, src_rec, dst_rec, tint)
```

Arguments

dst	An image.
src	An image.
src_rec	A rectangle.
dst_rec	A rectangle.
tint	A color.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageDraw(Image *dst, Image src, Rectangle srcRec, Rectangle dstRec, Color tint);
```

See Also

image() for creating image objects.

rectangle() for creating rectangle objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text()

`image_draw_circle` *Image draw circle*

Description

Draw circle within an image.

Usage

```
image_draw_circle(dst, center_x, center_y, radius, color)
```

Arguments

`dst`	An image.
`center_x`	An integer or a vector of integers.
`center_y`	An integer or a vector of integers.
`radius`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void ImageDrawCircle(Image *dst, int centerX, int centerY, int radius, Color color);
```

See Also

image() for creating image objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other circle functions: check_collision_circle_rec(), check_collision_point_circle(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), image_draw_circle_v()

---

image_draw_circle_v *Image draw circle v*

Description

Draw circle within an image (Vector version).

Usage

```
image_draw_circle_v(dst, center, radius, color)
```

Arguments

dst	An image.
center	A numeric vector of length 2 or a numeric matrix of width 2.
radius	An integer or a vector of integers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void ImageDrawCircleV(Image *dst, Vector2 center, int radius, Color color);
```

See Also

image() for creating image objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(),

draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other circle functions: check_collision_circle_rec(), check_collision_point_circle(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), image_draw_circle()

`image_draw_line` *Image draw line*

Description

Draw line within an image.

Usage

```
image_draw_line(dst, start_pos_x, start_pos_y, end_pos_x, end_pos_y, color)
```

Arguments

`dst`	An image.
`start_pos_x`	An integer or a vector of integers.
`start_pos_y`	An integer or a vector of integers.
`end_pos_x`	An integer or a vector of integers.
`end_pos_y`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void ImageDrawLine(Image *dst, int startPosX, int startPosY, int endPosX, int endPosY, Color color);
```

See Also

image() for creating image objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(),

image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other line functions: check_collision_point_line(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), image_draw_line_v()

`image_draw_line_v` *Image draw line v*

Description

Draw line within an image (Vector version).

Usage

```
image_draw_line_v(dst, start, end, color)
```

Arguments

dst	An image.
start	A numeric vector of length 2 or a numeric matrix of width 2.
end	A numeric vector of length 2 or a numeric matrix of width 2.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void ImageDrawLineV(Image *dst, Vector2 start, Vector2 end, Color color);
```

See Also

image() for creating image objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette() unload_image()

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(),

draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other line functions: check_collision_point_line(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), image_draw_line()

`image_draw_pixel` *Image draw pixel*

Description

Draw pixel within an image.

Usage

```
image_draw_pixel(dst, pos_x, pos_y, color)
```

Arguments

`dst`	An image.
`pos_x`	An integer or a vector of integers.
`pos_y`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void ImageDrawPixel(Image *dst, int posX, int posY, Color color);
```

See Also

image() for creating image objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(),

image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette() unload_image()

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`image_draw_pixel_v` *Image draw pixel v*

Description

Draw pixel within an image (Vector version).

Usage

```
image_draw_pixel_v(dst, position, color)
```

Arguments

`dst`	An image.
`position`	A numeric vector of length 2 or a numeric matrix of width 2.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void ImageDrawPixelV(Image *dst, Vector2 position, Color color);
```

See Also

image() for creating image objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette() unload_image()

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

`image_draw_rectangle` *Image draw rectangle*

Description

Draw rectangle within an image.

Usage

```
image_draw_rectangle(dst, pos_x, pos_y, width, height, color)
```

Arguments

dst	An image.
pos_x	An integer or a vector of integers.
pos_y	An integer or a vector of integers.
width	An integer or a vector of integers.
height	An integer or a vector of integers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void ImageDrawRectangle(Image *dst, int posX, int posY, int width, int height, Color color);
```

See Also

image() for creating image objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(),

draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), rectangle()

`image_draw_rectangle_lines` *Image draw rectangle lines*

Description

Draw rectangle lines within an image.

Usage

```
image_draw_rectangle_lines(dst, rec, thick, color)
```

Arguments

`dst`	An image.
`rec`	A rectangle or a list of rectangles.
`thick`	An integer or a vector of integers.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void ImageDrawRectangleLines(Image *dst, Rectangle rec, int thick, Color color);
```

See Also

image() for creating image objects.

rectangle() for creating rectangle objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette() unload_image()

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

`image_draw_rectangle_rec` *Image draw rectangle rec*

Description

Draw rectangle within an image.

Usage

```
image_draw_rectangle_rec(dst, rec, color)
```

Arguments

`dst`	An image.
`rec`	A rectangle or a list of rectangles.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void ImageDrawRectangleRec(Image *dst, Rectangle rec, Color color);
```

See Also

`image()` for creating image objects.

`rectangle()` for creating rectangle objects.

`color()` for creating color objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text_ex()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_from_texture()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

Other draw functions: `draw_billboard()`, `draw_bounding_box()`, `draw_circle_3d()`, `draw_circle_gradient()`, `draw_circle_lines()`, `draw_circle_sector_lines()`, `draw_circle_sector()`, `draw_circle_v()`,

draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_v(), image_draw_rectangle(), rectangle()

`image_draw_rectangle_v` *Image draw rectangle v*

Description

Draw rectangle within an image (Vector version).

Usage

```
image_draw_rectangle_v(dst, position, size, color)
```

Arguments

`dst`	An image.
`position`	A numeric vector of length 2 or a numeric matrix of width 2.
`size`	A numeric vector of length 2 or a numeric matrix of width 2.
`color`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void ImageDrawRectangleV(Image *dst, Vector2 position, Vector2 size, Color color);
```

See Also

image() for creating image objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette() unload_image()

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw()

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle(), rectangle()

`image_draw_text` *Image draw text*

Description

Draw text (using default font) within an image (destination).

Usage

```
image_draw_text(dst, text, pos_x, pos_y, font_size, color)
```

Arguments

dst	An image.
text	A string or a vector of strings.
pos_x	An integer or a vector of integers.
pos_y	An integer or a vector of integers.
font_size	An integer or a vector of integers.
color	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void ImageDrawText(Image *dst, const char *text, int posX, int posY, int fontSize, Color color);
```

See Also

image() for creating image objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(),

draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw()

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_find_index(), text_insert(), text_replace(), text_to_integer(), text_to_lower(), text_to_pascal(), text_to_upper()

`image_draw_text_ex` *Image draw text ex*

Description

Draw text (custom sprite font) within an image (destination).

Usage

```
image_draw_text_ex(dst, font, text, position, font_size, spacing, tint)
```

Arguments

`dst`	An image.
`font`	A font or a list of fonts.
`text`	A string or a vector of strings.
`position`	A numeric vector of length 2 or a numeric matrix of width 2.
`font_size`	A number or a vector of numbers.
`spacing`	A number or a vector of numbers.
`tint`	A color or a list of colors.

Note

This function is vectorized to allow for faster drawing. This function has been auto-generated from the following Raylib function definition:

```
void ImageDrawTextEx(Image *dst, Font font, const char *text, Vector2 position, float fontSize, float sp
```

See Also

image() for creating image objects.

font() for creating font objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other draw functions: draw_billboard(), draw_bounding_box(), draw_circle_3d(), draw_circle_gradient(), draw_circle_lines(), draw_circle_sector_lines(), draw_circle_sector(), draw_circle_v(), draw_circle(), draw_cube_texture_rec(), draw_cube_texture(), draw_cube_v(), draw_cube_wires_v(), draw_cube_wires(), draw_cube(), draw_cylinder_ex(), draw_cylinder_wires_ex(), draw_cylinder_wires(), draw_cylinder(), draw_ellipse_lines(), draw_ellipse(), draw_fps(), draw_grid(), draw_line_3d(), draw_line_bezier_cubic(), draw_line_bezier_quad(), draw_line_bezier(), draw_line_ex(), draw_line_v(), draw_line(), draw_model_ex(), draw_model_wires(), draw_model(), draw_pixel_v(), draw_pixel(), draw_plane(), draw_point_3d(), draw_poly_lines_ex(), draw_poly_lines(), draw_poly(), draw_ray(), draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), draw_ring_lines(), draw_ring(), draw_sphere_ex(), draw_sphere_wires(), draw_sphere(), draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), draw_text(), draw_triangle_3d(), draw_triangle_lines(), draw_triangle_strip_3d(), draw_triangle(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text(), image_draw()

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_find_index(), text_insert(), text_replace(), text_to_integer(), text_to_lower(), text_to_pascal(), text_to_upper()

`image_flip_horizontal` *Image flip horizontal*

Description

Flip image horizontally.

Usage

```
image_flip_horizontal(image)
```

Arguments

image	An image.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageFlipHorizontal(Image *image);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

image_flip_vertical *Image flip vertical*

Description

Flip image vertically.

Usage

```
image_flip_vertical(image)
```

Arguments

image	An image.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageFlipVertical(Image *image);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

image_format *Image format*

Description

Convert image data to desired format.

Usage

```
image_format(image, new_format)
```

Arguments

image	An image.
new_format	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageFormat(Image *image, int newFormat);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

image_from_image *Image from image*

Description

Create an image from another image piece.

Usage

```
image_from_image(image, rec)
```

Arguments

image	An image.
rec	A rectangle.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image ImageFromImage(Image image, Rectangle rec);
```

See Also

image() for creating image objects.

rectangle() for creating rectangle objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

image_mipmaps *Image mipmaps*

Description

Compute all mipmap levels for a provided image.

Usage

```
image_mipmaps(image)
```

Arguments

image	An image.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageMipmaps(Image *image);
```

See Also

`image()` for creating image objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text_ex()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_from_texture()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

---

image_resize *Image resize*

Description

Resize image (Bicubic scaling algorithm).

Usage

```
image_resize(image, new_width, new_height)
```

Arguments

`image`	An image.
`new_width`	An integer.
`new_height`	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageResize(Image *image, int newWidth, int newHeight);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

image_resize_canvas *Image resize canvas*

Description

Resize canvas and fill with color.

Usage

```
image_resize_canvas(image, new_width, new_height, offset_x, offset_y, fill)
```

Arguments

image	An image.
new_width	An integer.
new_height	An integer.
offset_x	An integer.
offset_y	An integer.
fill	A color.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageResizeCanvas(Image *image, int newWidth, int newHeight, int offsetX, int offsetY, Color fill)
```

See Also

image() for creating image objects.

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

image_resize_nn *Image resize nn*

Description

Resize image (Nearest-Neighbor scaling algorithm).

Usage

```
image_resize_nn(image, new_width, new_height)
```

Arguments

image	An image.
new_width	An integer.
new_height	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageResizeNN(Image *image, int newWidth,int newHeight);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

image_rotate_ccw *Image rotate ccw*

Description

Rotate image counter-clockwise 90deg.

Usage

```
image_rotate_ccw(image)
```

Arguments

image	An image.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageRotateCCW(Image *image);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(),

image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

`image_rotate_cw` *Image rotate cw*

Description

Rotate image clockwise 90deg.

Usage

```
image_rotate_cw(image)
```

Arguments

`image`	An image.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageRotateCW(Image *image);
```

See Also

`image()` for creating image objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_text_ex()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_from_texture()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

`image_text` *Image text*

Description

Create an image from text (default font).

Usage

```
image_text(text, font_size, color)
```

Arguments

`text`	A string.
`font_size`	An integer.
`color`	A color.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image ImageText(const char *text, int fontSize, Color color);
```

See Also

`color()` for creating color objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text_ex()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_from_texture()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

Other text functions: `draw_text_codepoint()`, `draw_text_ex()`, `draw_text_pro()`, `draw_text()`, `image_draw_text_ex()`, `image_draw_text()`, `image_text_ex()`, `load_file_text()`, `measure_text_ex()`, `measure_text()`, `text_find_index()`, `text_insert()`, `text_replace()`, `text_to_integer()`, `text_to_lower()`, `text_to_pascal()`, `text_to_upper()`

`image_text_ex` *Image text ex*

Description

Create an image from text (custom sprite font).

Usage

```
image_text_ex(font, text, font_size, spacing, tint)
```

Arguments

`font`	A font.
`text`	A string.
`font_size`	A number.
`spacing`	A number.
`tint`	A color.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image ImageTextEx(Font font, const char *text, float fontSize, float spacing, Color tint);
```

See Also

`font()` for creating font objects.

`color()` for creating color objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`,

load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_find_index(), text_insert(), text_replace(), text_to_integer(), text_to_lower(), text_to_pascal(), text_to_upper()

`image_to_pot` *Image to pot*

Description

Convert image to POT (power-of-two).

Usage

```
image_to_pot(image, fill)
```

Arguments

`image`	An image.
`fill`	A color.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ImageToPOT(Image *image, Color fill);
```

See Also

`image()` for creating image objects.

`color()` for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

init_audio_device *Init audio device*

Description

Initialize audio device and context.

Usage

```
init_audio_device()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void InitAudioDevice(void);
```

See Also

Other init functions: `init_window()`

Other audio functions: `close_audio_device()`, `is_audio_device_ready()`, `is_audio_stream_playing()`, `is_audio_stream_processed()`, `load_audio_stream()`, `pause_audio_stream()`, `play_audio_stream()`, `resume_audio_stream()`, `set_audio_stream_buffer_size_default()`, `set_audio_stream_pitch()`, `set_audio_stream_volume()`, `stop_audio_stream()`, `unload_audio_stream()`

init_window *Init window*

Description

Initialize window and OpenGL context.

Usage

```
init_window(width = 640, height = 480, title = "Raylibr")
```

Arguments

`width`	An integer. Default: `640`.
`height`	An integer. Default: `480`.
`title`	A string. Default: `"Raylibr"`.

Note

This function has been auto-generated from the following Raylib function definition:

```
void InitWindow(int width, int height, const char *title);
```

See Also

Other init functions: `init_audio_device()`

Other window functions: `clear_window_state()`, `close_window()`, `get_window_position()`, `get_window_scale_dpi()`, `is_window_focused()`, `is_window_fullscreen()`, `is_window_hidden()`, `is_window_maximized()`, `is_window_minimized()`, `is_window_ready()`, `is_window_resized()`, `is_window_state()`, `maximize_window()`, `minimize_window()`, `restore_window()`, `set_window_icon()`, `set_window_min_size()`, `set_window_monitor()`, `set_window_position()`, `set_window_size()`, `set_window_state()`, `set_window_title()`, `window_should_close()`

`is_audio_device_ready` *Is audio device ready*

Description

Check if audio device has been initialized successfully.

Usage

```
is_audio_device_ready()
```

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsAudioDeviceReady(void);
```

See Also

Other audio functions: `close_audio_device()`, `init_audio_device()`, `is_audio_stream_playing()`, `is_audio_stream_processed()`, `load_audio_stream()`, `pause_audio_stream()`, `play_audio_stream()`, `resume_audio_stream()`, `set_audio_stream_buffer_size_default()`, `set_audio_stream_pitch()`, `set_audio_stream_volume()`, `stop_audio_stream()`, `unload_audio_stream()`

`is_audio_stream_playing` *Is audio stream playing*

Description

Check if audio stream is playing.

Usage

```
is_audio_stream_playing(stream)
```

Arguments

`stream`	An audio_stream.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsAudioStreamPlaying(AudioStream stream);
```

See Also

`audio_stream()` for creating audio_stream objects.

Other audio functions: `close_audio_device()`, `init_audio_device()`, `is_audio_device_ready()`, `is_audio_stream_processed()`, `load_audio_stream()`, `pause_audio_stream()`, `play_audio_stream()`, `resume_audio_stream()`, `set_audio_stream_buffer_size_default()`, `set_audio_stream_pitch()`, `set_audio_stream_volume()`, `stop_audio_stream()`, `unload_audio_stream()`

Other stream functions: `is_audio_stream_processed()`, `is_music_stream_playing()`, `load_audio_stream()`, `load_music_stream()`, `pause_audio_stream()`, `pause_music_stream()`, `play_audio_stream()`, `play_music_stream()`, `resume_audio_stream()`, `resume_music_stream()`, `seek_music_stream()`, `set_audio_stream_buffer_size_default()`, `set_audio_stream_pitch()`, `set_audio_stream_volume()`, `stop_audio_stream()`, `stop_music_stream()`, `unload_audio_stream()`, `unload_music_stream()`, `update_music_stream()`

`is_audio_stream_processed` *Is audio stream processed*

Description

Check if any audio stream buffers requires refill.

Usage

```
is_audio_stream_processed(stream)
```

Arguments

`stream`	An audio_stream.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsAudioStreamProcessed(AudioStream stream);
```

See Also

`audio_stream()` for creating audio_stream objects.

Other audio functions: `close_audio_device()`, `init_audio_device()`, `is_audio_device_ready()`, `is_audio_stream_playing()`, `load_audio_stream()`, `pause_audio_stream()`, `play_audio_stream()`, `resume_audio_stream()`, `set_audio_stream_buffer_size_default()`, `set_audio_stream_pitch()`, `set_audio_stream_volume()`, `stop_audio_stream()`, `unload_audio_stream()`

Other stream functions: `is_audio_stream_playing()`, `is_music_stream_playing()`, `load_audio_stream()`, `load_music_stream()`, `pause_audio_stream()`, `pause_music_stream()`, `play_audio_stream()`, `play_music_stream()`, `resume_audio_stream()`, `resume_music_stream()`, `seek_music_stream()`, `set_audio_stream_buffer_size_default()`, `set_audio_stream_pitch()`, `set_audio_stream_volume()`, `stop_audio_stream()`, `stop_music_stream()`, `unload_audio_stream()`, `unload_music_stream()`, `update_music_stream()`

is_cursor_hidden *Is cursor hidden*

Description

Check if cursor is not visible.

Usage

```
is_cursor_hidden()
```

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsCursorHidden(void);
```

See Also

Other cursor functions: `disable_cursor()`, `enable_cursor()`, `hide_cursor()`, `is_cursor_on_screen()`, `set_mouse_cursor()`, `show_cursor()`

is_cursor_on_screen *Is cursor on screen*

Description

Check if cursor is on the screen.

Usage

```
is_cursor_on_screen()
```

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsCursorOnScreen(void);
```

See Also

Other cursor functions: disable_cursor(), enable_cursor(), hide_cursor(), is_cursor_hidden(), set_mouse_cursor(), show_cursor()

Other screen functions: get_screen_height(), get_screen_to_world_2d(), get_screen_width(), get_world_to_screen_2d(), get_world_to_screen_ex(), get_world_to_screen(), load_image_from_screen()

---

is_file_dropped *Is file dropped*

Description

Check if a file has been dropped into window.

Usage

```
is_file_dropped()
```

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsFileDropped(void);
```

See Also

Other file functions: file_exists(), get_file_extension(), get_file_mod_time(), get_file_name_without_ext(), get_file_name(), is_file_extension(), load_file_text()

---

is_file_extension *Is file extension*

Description

Check file extension (including point: .png, .wav).

Usage

```
is_file_extension(file_name, ext)
```

Arguments

file_name	A string.
ext	A string.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsFileExtension(const char *fileName, const char *ext);
```

See Also

Other file functions: file_exists(), get_file_extension(), get_file_mod_time(), get_file_name_without_ext(), get_file_name(), is_file_dropped(), load_file_text()

---

is_gamepad_available *Is gamepad available*

Description

Check if a gamepad is available.

Usage

```
is_gamepad_available(gamepad)
```

Arguments

gamepad	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsGamepadAvailable(int gamepad);
```

See Also

Other gamepad functions: get_gamepad_axis_count(), get_gamepad_axis_movement(), get_gamepad_button_presse
get_gamepad_name(), is_gamepad_button_down(), is_gamepad_button_pressed(), is_gamepad_button_released()
is_gamepad_button_up(), set_gamepad_mappings()

is_gamepad_button_down

Is gamepad button down

Description

Check if a gamepad button is being pressed.

Usage

```
is_gamepad_button_down(gamepad, button)
```

Arguments

gamepad	An integer.
button	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsGamepadButtonDown(int gamepad, int button);
```

See Also

Other gamepad functions: get_gamepad_axis_count(), get_gamepad_axis_movement(), get_gamepad_button_presse
get_gamepad_name(), is_gamepad_available(), is_gamepad_button_pressed(), is_gamepad_button_released(),
is_gamepad_button_up(), set_gamepad_mappings()

is_gamepad_button_pressed

Is gamepad button pressed

Description

Check if a gamepad button has been pressed once.

Usage

```
is_gamepad_button_pressed(gamepad, button)
```

Arguments

gamepad	An integer.
button	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsGamepadButtonPressed(int gamepad, int button);
```

See Also

Other gamepad functions: get_gamepad_axis_count(), get_gamepad_axis_movement(), get_gamepad_button_presse
get_gamepad_name(), is_gamepad_available(), is_gamepad_button_down(), is_gamepad_button_released(),
is_gamepad_button_up(), set_gamepad_mappings()

---

is_gamepad_button_released

Is gamepad button released

Description

Check if a gamepad button has been released once.

Usage

```
is_gamepad_button_released(gamepad, button)
```

Arguments

gamepad	An integer.
button	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsGamepadButtonReleased(int gamepad, int button);
```

See Also

Other gamepad functions: get_gamepad_axis_count(), get_gamepad_axis_movement(), get_gamepad_button_presse
get_gamepad_name(), is_gamepad_available(), is_gamepad_button_down(), is_gamepad_button_pressed(),
is_gamepad_button_up(), set_gamepad_mappings()

---

is_gamepad_button_up *Is gamepad button up*

Description

Check if a gamepad button is NOT being pressed.

Usage

```
is_gamepad_button_up(gamepad, button)
```

Arguments

gamepad	An integer.
button	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsGamepadButtonUp(int gamepad, int button);
```

See Also

Other gamepad functions: get_gamepad_axis_count(), get_gamepad_axis_movement(), get_gamepad_button_presse
get_gamepad_name(), is_gamepad_available(), is_gamepad_button_down(), is_gamepad_button_pressed(),
is_gamepad_button_released(), set_gamepad_mappings()

is_gesture_detected *Is gesture detected*

Description

Check if a gesture have been detected.

Usage

```
is_gesture_detected(gesture)
```

Arguments

gesture	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsGestureDetected(int gesture);
```

is_key_down *Is key down*

Description

Check if a key is being pressed.

Usage

```
is_key_down(key)
```

Arguments

key	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsKeyDown(int key);
```

See Also

Other key functions: `get_char_pressed()`, `get_key_pressed()`, `is_key_pressed()`, `is_key_released()`, `is_key_up()`, `set_exit_key()`

---

is_key_pressed *Is key pressed*

Description

Check if a key has been pressed once.

Usage

```
is_key_pressed(key)
```

Arguments

key	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsKeyPressed(int key);
```

See Also

Other key functions: `get_char_pressed()`, `get_key_pressed()`, `is_key_down()`, `is_key_released()`, `is_key_up()`, `set_exit_key()`

is_key_released *Is key released*

Description

Check if a key has been released once.

Usage

```
is_key_released(key)
```

Arguments

key	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsKeyReleased(int key);
```

See Also

Other key functions: get_char_pressed(), get_key_pressed(), is_key_down(), is_key_pressed(), is_key_up(), set_exit_key()

is_key_up *Is key up*

Description

Check if a key is NOT being pressed.

Usage

```
is_key_up(key)
```

Arguments

key	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsKeyUp(int key);
```

See Also

Other key functions: get_char_pressed(), get_key_pressed(), is_key_down(), is_key_pressed(), is_key_released(), set_exit_key()

---

is_mouse_button_down *Is mouse button down*

Description

Check if a mouse button is being pressed.

Usage

```
is_mouse_button_down(button)
```

Arguments

button	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsMouseButtonDown(int button);
```

See Also

Other mouse functions: get_mouse_delta(), get_mouse_position(), get_mouse_wheel_move(), get_mouse_x(), get_mouse_y(), is_mouse_button_pressed(), is_mouse_button_released(), is_mouse_button_up(), set_mouse_cursor(), set_mouse_offset(), set_mouse_position(), set_mouse_scale()

is_mouse_button_pressed

Is mouse button pressed

Description

Check if a mouse button has been pressed once.

Usage

```
is_mouse_button_pressed(button)
```

Arguments

button	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsMouseButtonPressed(int button);
```

See Also

Other mouse functions: get_mouse_delta(), get_mouse_position(), get_mouse_wheel_move(), get_mouse_x(), get_mouse_y(), is_mouse_button_down(), is_mouse_button_released(), is_mouse_button_up(), set_mouse_cursor(), set_mouse_offset(), set_mouse_position(), set_mouse_scale()

is_mouse_button_released

Is mouse button released

Description

Check if a mouse button has been released once.

Usage

```
is_mouse_button_released(button)
```

Arguments

button	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsMouseButtonReleased(int button);
```

See Also

Other mouse functions: get_mouse_delta(), get_mouse_position(), get_mouse_wheel_move(), get_mouse_x(), get_mouse_y(), is_mouse_button_down(), is_mouse_button_pressed(), is_mouse_button_up(), set_mouse_cursor(), set_mouse_offset(), set_mouse_position(), set_mouse_scale()

---

is_mouse_button_up *Is mouse button up*

Description

Check if a mouse button is NOT being pressed.

Usage

```
is_mouse_button_up(button)
```

Arguments

button	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsMouseButtonUp(int button);
```

See Also

Other mouse functions: get_mouse_delta(), get_mouse_position(), get_mouse_wheel_move(), get_mouse_x(), get_mouse_y(), is_mouse_button_down(), is_mouse_button_pressed(), is_mouse_button_release set_mouse_cursor(), set_mouse_offset(), set_mouse_position(), set_mouse_scale()

`is_music_stream_playing` *Is music stream playing*

Description

Check if music is playing.

Usage

```
is_music_stream_playing(music)
```

Arguments

`music`	A music.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsMusicStreamPlaying(Music music);
```

See Also

`music()` for creating music objects.

Other music functions: `get_music_time_length()`, `get_music_time_played()`, `load_music_stream()`, `pause_music_stream()`, `play_music_stream()`, `resume_music_stream()`, `seek_music_stream()`, `set_music_pitch()`, `set_music_volume()`, `stop_music_stream()`, `unload_music_stream()`, `update_music_stream()`

Other stream functions: `is_audio_stream_playing()`, `is_audio_stream_processed()`, `load_audio_stream()`, `load_music_stream()`, `pause_audio_stream()`, `pause_music_stream()`, `play_audio_stream()`, `play_music_stream()`, `resume_audio_stream()`, `resume_music_stream()`, `seek_music_stream()`, `set_audio_stream_buffer_size_default()`, `set_audio_stream_pitch()`, `set_audio_stream_volume()`, `stop_audio_stream()`, `stop_music_stream()`, `unload_audio_stream()`, `unload_music_stream()`, `update_music_stream()`

is_sound_playing *Is sound playing*

Description

Check if a sound is currently playing.

Usage

```
is_sound_playing(sound)
```

Arguments

sound	A sound.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsSoundPlaying(Sound sound);
```

See Also

`sound()` for creating sound objects.

Other sound functions: `load_sound_from_wave()`, `load_sound()`, `pause_sound()`, `play_sound_multi()`, `play_sound()`, `resume_sound()`, `set_sound_pitch()`, `set_sound_volume()`, `sound()`, `stop_sound_multi()`, `stop_sound()`, `unload_sound()`

is_window_focused *Is window focused*

Description

Check if window is currently focused (only PLATFORM_DESKTOP).

Usage

```
is_window_focused()
```

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsWindowFocused(void);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

---

is_window_fullscreen *Is window fullscreen*

Description

Check if window is currently fullscreen.

Usage

```
is_window_fullscreen()
```

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsWindowFullscreen(void);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

is_window_hidden *Is window hidden*

Description

Check if window is currently hidden (only PLATFORM_DESKTOP).

Usage

```
is_window_hidden()
```

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsWindowHidden(void);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

is_window_maximized *Is window maximized*

Description

Check if window is currently maximized (only PLATFORM_DESKTOP).

Usage

```
is_window_maximized()
```

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsWindowMaximized(void);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

---

is_window_minimized *Is window minimized*

Description

Check if window is currently minimized (only PLATFORM_DESKTOP).

Usage

```
is_window_minimized()
```

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsWindowMinimized(void);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

is_window_ready *Is window ready*

Description

Check if window has been initialized successfully.

Usage

```
is_window_ready()
```

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsWindowReady(void);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

is_window_resized *Is window resized*

Description

Check if window has been resized last frame.

Usage

```
is_window_resized()
```

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsWindowResized(void);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

---

is_window_state *Is window state*

Description

Check if one specific window flag is enabled.

Usage

```
is_window_state(flag)
```

Arguments

flag	A non-negative integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool IsWindowState(unsigned int flag);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

load_audio_stream *Load audio stream*

Description

Load audio stream (to stream raw audio pcm data).

Usage

```
load_audio_stream(sample_rate, sample_size, channels)
```

Arguments

sample_rate	A non-negative integer.
sample_size	A non-negative integer.
channels	A non-negative integer.

Value

An audio_stream

Note

This function has been auto-generated from the following Raylib function definition:

```
AudioStream LoadAudioStream(unsigned int sampleRate, unsigned int sampleSize, unsigned int channels);
```

See Also

Other audio functions: close_audio_device(), init_audio_device(), is_audio_device_ready(), is_audio_stream_playing(), is_audio_stream_processed(), pause_audio_stream(), play_audio_stream(), resume_audio_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), unload_audio_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_music_stream(), pause_audio_stream(), pause_music_stream(), play_audio_stream(), play_music_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), unload_music_stream(), update_music_stream()

load_file_text *Load file text*

Description

Load text data from file (read), returns a '\0' terminated string.

Usage

```
load_file_text(file_name)
```

Arguments

file_name	A string.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
char *LoadFileText(const char *fileName);
```

See Also

Other file functions: file_exists(), get_file_extension(), get_file_mod_time(), get_file_name_without_ext(), get_file_name(), is_file_dropped(), is_file_extension()

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), measure_text_ex(), measure_text(), text_find_index(), text_insert(), text_replace(), text_to_integer(), text_to_lower(), text_to_pascal(), text_to_upper()

load_font *Load font*

Description

Load font from file into GPU memory (VRAM).

Usage

```
load_font(file_name)
```

Arguments

file_name	A string.

Value

A font

Note

This function has been auto-generated from the following Raylib function definition:

```
Font LoadFont(const char *fileName);
```

See Also

Other font functions: get_font_default(), load_font_ex(), load_font_from_image(), unload_font_data(), unload_font()

---

load_font_ex *Load font ex*

Description

Load font from file with extended parameters.

Usage

```
load_font_ex(file_name, font_size)
```

Arguments

file_name	A string.
font_size	An integer.

Value

A font

Note

This function has been auto-generated from the following Raylib function definition:

```
Font LoadFontEx(const char *fileName, int fontSize, int *fontChars, int glyphCount);
```

See Also

Other font functions: get_font_default(), load_font_from_image(), load_font(), unload_font_data(), unload_font()

load_font_from_image *Load font from image*

Description

Load font from Image (XNA style).

Usage

```
load_font_from_image(image, key, first_char)
```

Arguments

image	An image.
key	A color.
first_char	An integer.

Value

A font

Note

This function has been auto-generated from the following Raylib function definition:

```
Font LoadFontFromImage(Image image, Color key, int firstChar);
```

See Also

image() for creating image objects.

color() for creating color objects.

Other font functions: get_font_default(), load_font_ex(), load_font(), unload_font_data(), unload_font()

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

load_image *Load image*

Description

Load image from file into CPU memory (RAM).

Usage

```
load_image(file_name)
```

Arguments

file_name	A string.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image LoadImage(const char *fileName);
```

See Also

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

load_image_from_screen *Load image from screen*

Description

Load image from screen buffer and (screenshot).

Usage

```
load_image_from_screen()
```

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image LoadImageFromScreen(void);
```

See Also

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

Other screen functions: get_screen_height(), get_screen_to_world_2d(), get_screen_width(), get_world_to_screen_2d(), get_world_to_screen_ex(), get_world_to_screen(), is_cursor_on_screen()

`load_image_from_texture` *Load image from texture*

Description

Load image from GPU texture data.

Usage

```
load_image_from_texture(texture)
```

Arguments

`texture`	A texture.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image LoadImageFromTexture(Texture2D texture);
```

See Also

`texture()` for creating texture objects.

Other image functions: `export_image_as_code()`, `export_image()`, `gen_image_cellular()`, `gen_image_checked()`, `gen_image_color()`, `gen_image_gradient_h()`, `gen_image_gradient_radial()`, `gen_image_gradient_v()`, `gen_image_white_noise()`, `get_image_alpha_border()`, `get_image_color()`, `image_alpha_clear()`, `image_alpha_crop()`, `image_alpha_mask()`, `image_alpha_premultiply()`, `image_clear_background()`, `image_color_brightness()`, `image_color_contrast()`, `image_color_grayscale()`, `image_color_invert()`, `image_color_replace()`, `image_color_tint()`, `image_copy()`, `image_crop()`, `image_dither()`, `image_draw_circle_v()`, `image_draw_circle()`, `image_draw_line_v()`, `image_draw_line()`, `image_draw_pixel_v()`, `image_draw_pixel()`, `image_draw_rectangle_lines()`, `image_draw_rectangle_rec()`, `image_draw_rectangle_v()`, `image_draw_rectangle()`, `image_draw_text_ex()`, `image_draw_text()`, `image_draw()`, `image_flip_horizontal()`, `image_flip_vertical()`, `image_format()`, `image_from_image()`, `image_mipmaps()`, `image_resize_canvas()`, `image_resize_nn()`, `image_resize()`, `image_rotate_ccw()`, `image_rotate_cw()`, `image_text_ex()`, `image_text()`, `image_to_pot()`, `load_font_from_image()`, `load_image_from_screen()`, `load_image_raw()`, `load_image()`, `load_texture_from_image()`, `unload_image_colors()`, `unload_image_palette()`, `unload_image()`

Other texture functions: `begin_texture_mode()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`, `draw_texture_rec()`, `draw_texture_tiled()`, `draw_texture_v()`, `draw_texture()`, `end_texture_mode()`, `gen_texture_mipmaps()`, `load_render_texture()`, `load_texture_from_image()`,

load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

---

load_image_raw *Load image raw*

Description

Load image from RAW file data.

Usage

```
load_image_raw(file_name, width, height, format, header_size)
```

Arguments

file_name	A string.
width	An integer.
height	An integer.
format	An integer.
header_size	An integer.

Value

An image

Note

This function has been auto-generated from the following Raylib function definition:

```
Image LoadImageRaw(const char *fileName, int width, int height, int format, int headerSize);
```

See Also

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette(), unload_image()

load_model *Load model*

Description

Load model from files (meshes and materials).

Usage

```
load_model(file_name)
```

Arguments

file_name	A string.

Value

A model

Note

This function has been auto-generated from the following Raylib function definition:

```
Model LoadModel(const char *fileName);
```

See Also

Other model functions: draw_model_ex(), draw_model_wires(), draw_model(), get_model_bounding_box(), get_ray_collision_model(), unload_model()

load_music_stream *Load music stream*

Description

Load music stream from file.

Usage

```
load_music_stream(file_name)
```

Arguments

file_name	A string.

Value

A music

Note

This function has been auto-generated from the following Raylib function definition:

```
Music LoadMusicStream(const char *fileName);
```

See Also

Other music functions: get_music_time_length(), get_music_time_played(), is_music_stream_playing(), pause_music_stream(), play_music_stream(), resume_music_stream(), seek_music_stream(), set_music_pitch(), set_music_volume(), stop_music_stream(), unload_music_stream(), update_music_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), pause_audio_stream(), pause_music_stream(), play_audio_stream(), play_music_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), unload_music_stream(), update_music_stream()

---

load_render_texture *Load render texture*

Description

Load texture for rendering (framebuffer).

Usage

```
load_render_texture(width, height)
```

Arguments

width	An integer.
height	An integer.

Value

A render_texture

Note

This function has been auto-generated from the following Raylib function definition:

```
RenderTexture2D LoadRenderTexture(int width, int height);
```

See Also

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

---

load_shader *Load shader*

Description

Load shader from files and bind default locations.

Usage

```
load_shader(vs_file_name, fs_file_name)
```

Arguments

vs_file_name	A string.
fs_file_name	A string.

Value

A shader

Note

This function has been auto-generated from the following Raylib function definition:

```
Shader LoadShader(const char *vsFileName, const char *fsFileName);
```

See Also

Other shader functions: begin_shader_mode(), end_shader_mode(), get_shader_location_attrib(), get_shader_location(), load_shader_from_memory(), set_shader_value_matrix(), set_shader_value_texture(), set_shader_value(), unload_shader()

load_shader_from_memory *Load shader from memory*

Description

Load shader from code strings and bind default locations.

Usage

```
load_shader_from_memory(vs_code, fs_code)
```

Arguments

vs_code	A string.
fs_code	A string.

Value

A shader

Note

This function has been auto-generated from the following Raylib function definition:

```
Shader LoadShaderFromMemory(const char *vsCode, const char *fsCode);
```

See Also

Other shader functions: begin_shader_mode(), end_shader_mode(), get_shader_location_attrib(), get_shader_location(), load_shader(), set_shader_value_matrix(), set_shader_value_texture(), set_shader_value(), unload_shader()

load_sound *Load sound*

Description

Load sound from file.

Usage

```
load_sound(file_name)
```

Arguments

file_name	A string.

Value

A sound

Note

This function has been auto-generated from the following Raylib function definition:

```
Sound LoadSound(const char *fileName);
```

See Also

Other sound functions: is_sound_playing(), load_sound_from_wave(), pause_sound(), play_sound_multi(), play_sound(), resume_sound(), set_sound_pitch(), set_sound_volume(), sound(), stop_sound_multi(), stop_sound(), unload_sound()

---

load_sound_from_wave *Load sound from wave*

Description

Load sound from wave data.

Usage

```
load_sound_from_wave(wave)
```

Arguments

wave	A wave.

Value

A sound

Note

This function has been auto-generated from the following Raylib function definition:

```
Sound LoadSoundFromWave(Wave wave);
```

See Also

wave() for creating wave objects.

Other sound functions: is_sound_playing(), load_sound(), pause_sound(), play_sound_multi(), play_sound(), resume_sound(), set_sound_pitch(), set_sound_volume(), sound(), stop_sound_multi(), stop_sound(), unload_sound()

Other wave functions: export_wave_as_code(), export_wave(), load_wave(), unload_wave()

---

load_storage_value *Load storage value*

Description

Load integer value from storage file (from defined position).

Usage

```
load_storage_value(position)
```

Arguments

position	A non-negative integer.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int LoadStorageValue(unsigned int position);
```

---

load_texture *Load texture*

Description

Load texture from file into GPU memory (VRAM).

Usage

```
load_texture(file_name)
```

Arguments

file_name	A string.

Value

A texture

Note

This function has been auto-generated from the following Raylib function definition:

```
Texture2D LoadTexture(const char *fileName);
```

See Also

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

---

load_texture_from_image

Load texture from image

Description

Load texture from image data.

Usage

```
load_texture_from_image(image)
```

Arguments

image	An image.

Value

A texture

Note

This function has been auto-generated from the following Raylib function definition:

```
Texture2D LoadTextureFromImage(Image image);
```

See Also

image() for creating image objects.

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), unload_image_colors(), unload_image_palette(), unload_image()

---

load_wave *Load wave*

Description

Load wave data from file.

Usage

```
load_wave(file_name)
```

Arguments

file_name	A string.

Value

A wave

Note

This function has been auto-generated from the following Raylib function definition:

```
Wave LoadWave(const char *fileName);
```

See Also

Other wave functions: `export_wave_as_code()`, `export_wave()`, `load_sound_from_wave()`, `unload_wave()`

`material_map` *Material map*

Description

Create a new material_map object.

Usage

```
material_map(texture, color, value)
```

Arguments

`texture`	A texture. Material map texture.
`color`	A color. Material map color.
`value`	A number. Material map value.

Value

A material_map

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct MaterialMap {
    Texture2D texture;      // Material map texture
    Color color;            // Material map color
    float value;            // Material map value
} MaterialMap;
```

maximize_window *Maximize window*

Description

Set window state: maximized, if resizable (only PLATFORM_DESKTOP).

Usage

```
maximize_window()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void MaximizeWindow(void);
```

See Also

Other window functions: `clear_window_state()`, `close_window()`, `get_window_position()`, `get_window_scale_dpi()`, `init_window()`, `is_window_focused()`, `is_window_fullscreen()`, `is_window_hidden()`, `is_window_maximized()`, `is_window_minimized()`, `is_window_ready()`, `is_window_resized()`, `is_window_state()`, `minimize_window()`, `restore_window()`, `set_window_icon()`, `set_window_min_size()`, `set_window_monitor()`, `set_window_position()`, `set_window_size()`, `set_window_state()`, `set_window_title()`, `window_should_close()`

measure_text *Measure text*

Description

Measure string width for default font.

Usage

```
measure_text(text, font_size)
```

Arguments

`text`	A string.
`font_size`	An integer.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int MeasureText(const char *text, int fontSize);
```

See Also

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), text_find_index(), text_insert(), text_replace(), text_to_integer(), text_to_lower(), text_to_pascal(), text_to_upper()

---

measure_text_ex *Measure text ex*

Description

Measure string size for Font.

Usage

```
measure_text_ex(font, text, font_size, spacing)
```

Arguments

font	A font.
text	A string.
font_size	A number.
spacing	A number.

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 MeasureTextEx(Font font, const char *text, float fontSize, float spacing);
```

See Also

font() for creating font objects.

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text(), text_find_index(), text_insert(), text_replace(), text_to_integer(), text_to_lower(), text_to_pascal(), text_to_upper()

minimize_window *Minimize window*

Description

Set window state: minimized, if resizable (only PLATFORM_DESKTOP).

Usage

```
minimize_window()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void MinimizeWindow(void);
```

See Also

Other window functions: `clear_window_state()`, `close_window()`, `get_window_position()`, `get_window_scale_dpi()`, `init_window()`, `is_window_focused()`, `is_window_fullscreen()`, `is_window_hidden()`, `is_window_maximized()`, `is_window_minimized()`, `is_window_ready()`, `is_window_resized()`, `is_window_state()`, `maximize_window()`, `restore_window()`, `set_window_icon()`, `set_window_min_size()`, `set_window_monitor()`, `set_window_position()`, `set_window_size()`, `set_window_state()`, `set_window_title()`, `window_should_close()`

npatch_info *Npatch info*

Description

Create a new npatch_info object.

Usage

```
npatch_info(source, left, top, right, bottom, layout)
```

Arguments

source	A rectangle. Texture source rectangle.
left	An integer. Left border offset.
top	An integer. Top border offset.
right	An integer. Right border offset.
bottom	An integer. Bottom border offset.
layout	An integer. Layout of the n-patch: 3x3, 1x3 or 3x1.

Value

A npatch_info

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct NPatchInfo {
    Rectangle source;       // Texture source rectangle
    int left;               // Left border offset
    int top;                // Top border offset
    int right;              // Right border offset
    int bottom;             // Bottom border offset
    int layout;             // Layout of the n-patch: 3x3, 1x3 or 3x1
} NPatchInfo;
```

---

pause_audio_stream *Pause audio stream*

Description

Pause audio stream.

Usage

```
pause_audio_stream(stream)
```

Arguments

stream	An audio_stream.

Note

This function has been auto-generated from the following Raylib function definition:

```
void PauseAudioStream(AudioStream stream);
```

See Also

audio_stream() for creating audio_stream objects.

Other audio functions: close_audio_device(), init_audio_device(), is_audio_device_ready(), is_audio_stream_playing(), is_audio_stream_processed(), load_audio_stream(), play_audio_stream(), resume_audio_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), unload_audio_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_music_stream(), play_audio_stream(),

play_music_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), unload_music_stream(), update_music_stream()

---

pause_music_stream *Pause music stream*

Description

Pause music playing.

Usage

```
pause_music_stream(music)
```

Arguments

music	A music.

Note

This function has been auto-generated from the following Raylib function definition:

```
void PauseMusicStream(Music music);
```

See Also

music() for creating music objects.

Other music functions: get_music_time_length(), get_music_time_played(), is_music_stream_playing(), load_music_stream(), play_music_stream(), resume_music_stream(), seek_music_stream(), set_music_pitch(), set_music_volume(), stop_music_stream(), unload_music_stream(), update_music_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_audio_stream(), play_audio_stream(), play_music_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), unload_music_stream(), update_music_stream()

pause_sound *Pause sound*

Description

Pause a sound.

Usage

```
pause_sound(sound)
```

Arguments

sound	A sound.

Note

This function has been auto-generated from the following Raylib function definition:

```
void PauseSound(Sound sound);
```

See Also

`sound()` for creating sound objects.

Other sound functions: `is_sound_playing()`, `load_sound_from_wave()`, `load_sound()`, `play_sound_multi()`, `play_sound()`, `resume_sound()`, `set_sound_pitch()`, `set_sound_volume()`, `sound()`, `stop_sound_multi()`, `stop_sound()`, `unload_sound()`

play_audio_stream *Play audio stream*

Description

Play audio stream.

Usage

```
play_audio_stream(stream)
```

Arguments

stream	An audio_stream.

Note

This function has been auto-generated from the following Raylib function definition:

```
void PlayAudioStream(AudioStream stream);
```

See Also

audio_stream() for creating audio_stream objects.

Other audio functions: close_audio_device(), init_audio_device(), is_audio_device_ready(), is_audio_stream_playing(), is_audio_stream_processed(), load_audio_stream(), pause_audio_stream(), resume_audio_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), unload_audio_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_audio_stream(), pause_music_stream(), play_music_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), unload_music_stream(), update_music_stream()

---

play_music_stream *Play music stream*

Description

Start music playing.

Usage

```
play_music_stream(music)
```

Arguments

music	A music.

Note

This function has been auto-generated from the following Raylib function definition:

```
void PlayMusicStream(Music music);
```

See Also

music() for creating music objects.

Other music functions: get_music_time_length(), get_music_time_played(), is_music_stream_playing(), load_music_stream(), pause_music_stream(), resume_music_stream(), seek_music_stream(), set_music_pitch(), set_music_volume(), stop_music_stream(), unload_music_stream(), update_music_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_audio_stream(), pause_music_stream(), play_audio_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), unload_music_stream(), update_music_stream()

---

play_sound *Play sound*

Description

Play a sound.

Usage

```
play_sound(sound)
```

Arguments

sound	A sound.

Note

This function has been auto-generated from the following Raylib function definition:

```
void PlaySound(Sound sound);
```

See Also

sound() for creating sound objects.

Other sound functions: is_sound_playing(), load_sound_from_wave(), load_sound(), pause_sound(), play_sound_multi(), resume_sound(), set_sound_pitch(), set_sound_volume(), sound(), stop_sound_multi(), stop_sound(), unload_sound()

play_sound_multi *Play sound multi*

Description

Play a sound (using multichannel buffer pool).

Usage

```
play_sound_multi(sound)
```

Arguments

sound	A sound.

Note

This function has been auto-generated from the following Raylib function definition:

```
void PlaySoundMulti(Sound sound);
```

See Also

`sound()` for creating sound objects.

Other sound functions: `is_sound_playing()`, `load_sound_from_wave()`, `load_sound()`, `pause_sound()`, `play_sound()`, `resume_sound()`, `set_sound_pitch()`, `set_sound_volume()`, `sound()`, `stop_sound_multi()`, `stop_sound()`, `unload_sound()`

ray *Ray*

Description

Create a new ray object.

Usage

```
ray(position, direction)
```

Arguments

position	A numeric vector of length 3. Ray position (origin).
direction	A numeric vector of length 3. Ray direction.

Value

A ray

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct Ray {
    Vector3 position;       // Ray position (origin)
    Vector3 direction;      // Ray direction
} Ray;
```

`ray_collision` *Ray collision*

Description

Create a new ray_collision object.

Usage

```
ray_collision(hit, distance, point, normal)
```

Arguments

`hit`	A logical. Did the ray hit something?.
`distance`	A number. Distance to nearest hit.
`point`	A numeric vector of length 3. Point of nearest hit.
`normal`	A numeric vector of length 3. Surface normal of hit.

Value

A ray_collision

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct RayCollision {
    bool hit;               // Did the ray hit something?
    float distance;         // Distance to nearest hit
    Vector3 point;          // Point of nearest hit
    Vector3 normal;         // Surface normal of hit
} RayCollision;
```

raylib_version *Raylib Version*

Description

Get the version of the Raylib library that is being used.

Usage

```
raylib_version()
```

Value

A string

rectangle *Rectangle*

Description

Create a new rectangle object.

Usage

```
rectangle(x, y, width, height)
```

Arguments

x	A number. Rectangle top-left corner position x.
y	A number. Rectangle top-left corner position y.
width	A number. Rectangle width.
height	A number. Rectangle height.

Value

A rectangle

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct Rectangle {
    float x;                // Rectangle top-left corner position x
    float y;                // Rectangle top-left corner position y
    float width;            // Rectangle width
    float height;           // Rectangle height
} Rectangle;
```

See Also

Other rectangle functions: draw_rectangle_gradient_ex(), draw_rectangle_gradient_h(), draw_rectangle_gradient_v(), draw_rectangle_lines_ex(), draw_rectangle_lines(), draw_rectangle_pro(), draw_rectangle_rec(), draw_rectangle_rounded_lines(), draw_rectangle_rounded(), draw_rectangle_v(), draw_rectangle(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle()

---

render_texture *Render texture*

Description

Create a new render_texture object.

Usage

```
render_texture(id, texture, depth)
```

Arguments

id	A non-negative integer. OpenGL framebuffer object id.
texture	A texture. Color buffer attachment texture.
depth	A texture. Depth buffer attachment texture.

Value

A render_texture

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct RenderTexture {
    unsigned int id;        // OpenGL framebuffer object id
    Texture texture;        // Color buffer attachment texture
    Texture depth;          // Depth buffer attachment texture
} RenderTexture;
```

restore_window *Restore window*

Description

Set window state: not minimized/maximized (only PLATFORM_DESKTOP).

Usage

```
restore_window()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void RestoreWindow(void);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

resume_audio_stream *Resume audio stream*

Description

Resume audio stream.

Usage

```
resume_audio_stream(stream)
```

Arguments

stream	An audio_stream.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ResumeAudioStream(AudioStream stream);
```

See Also

audio_stream() for creating audio_stream objects.

Other audio functions: close_audio_device(), init_audio_device(), is_audio_device_ready(), is_audio_stream_playing(), is_audio_stream_processed(), load_audio_stream(), pause_audio_stream(), play_audio_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), unload_audio_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_audio_stream(), pause_music_stream(), play_audio_stream(), play_music_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), unload_music_stream(), update_music_stream()

---

resume_music_stream *Resume music stream*

Description

Resume playing paused music.

Usage

```
resume_music_stream(music)
```

Arguments

music	A music.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ResumeMusicStream(Music music);
```

See Also

music() for creating music objects.

Other music functions: get_music_time_length(), get_music_time_played(), is_music_stream_playing(), load_music_stream(), pause_music_stream(), play_music_stream(), seek_music_stream(), set_music_pitch(), set_music_volume(), stop_music_stream(), unload_music_stream(), update_music_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_audio_stream(), pause_music_stream(), play_audio_stream(), play_music_stream(), resume_audio_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), unload_music_stream(), update_music_stream()

resume_sound *Resume sound*

Description

Resume a paused sound.

Usage

```
resume_sound(sound)
```

Arguments

sound	A sound.

Note

This function has been auto-generated from the following Raylib function definition:

```
void ResumeSound(Sound sound);
```

See Also

sound() for creating sound objects.

Other sound functions: is_sound_playing(), load_sound_from_wave(), load_sound(), pause_sound(), play_sound_multi(), play_sound(), set_sound_pitch(), set_sound_volume(), sound(), stop_sound_multi(), stop_sound(), unload_sound()

save_storage_value *Save storage value*

Description

Save integer value to storage file (to defined position), returns true on success.

Usage

```
save_storage_value(position, value)
```

Arguments

position	A non-negative integer.
value	An integer.

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool SaveStorageValue(unsigned int position, int value);
```

`seek_music_stream` *Seek music stream*

Description

Seek music to a position (in seconds).

Usage

```
seek_music_stream(music, position)
```

Arguments

`music`	A music.
`position`	A number.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SeekMusicStream(Music music, float position);
```

See Also

`music()` for creating music objects.

Other music functions: `get_music_time_length()`, `get_music_time_played()`, `is_music_stream_playing()`, `load_music_stream()`, `pause_music_stream()`, `play_music_stream()`, `resume_music_stream()`, `set_music_pitch()`, `set_music_volume()`, `stop_music_stream()`, `unload_music_stream()`, `update_music_stream()`

Other stream functions: `is_audio_stream_playing()`, `is_audio_stream_processed()`, `is_music_stream_playing()`, `load_audio_stream()`, `load_music_stream()`, `pause_audio_stream()`, `pause_music_stream()`, `play_audio_stream()`, `play_music_stream()`, `resume_audio_stream()`, `resume_music_stream()`, `set_audio_stream_buffer_size_default()`, `set_audio_stream_pitch()`, `set_audio_stream_volume()`, `stop_audio_stream()`, `stop_music_stream()`, `unload_audio_stream()`, `unload_music_stream()`, `update_music_stream()`

set_audio_stream_buffer_size_default

Set audio stream buffer size default

Description

Default size for new audio streams.

Usage

```
set_audio_stream_buffer_size_default(size)
```

Arguments

size	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetAudioStreamBufferSizeDefault(int size);
```

See Also

Other audio functions: close_audio_device(), init_audio_device(), is_audio_device_ready(), is_audio_stream_playing(), is_audio_stream_processed(), load_audio_stream(), pause_audio_stream(), play_audio_stream(), resume_audio_stream(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), unload_audio_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_audio_stream(), pause_music_stream(), play_audio_stream(), play_music_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), unload_music_stream(), update_music_stream()

set_audio_stream_pitch

Set audio stream pitch

Description

Set pitch for audio stream (1.0 is base level).

Usage

```
set_audio_stream_pitch(stream, pitch)
```

Arguments

stream	An audio_stream.
pitch	A number.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetAudioStreamPitch(AudioStream stream, float pitch);
```

See Also

audio_stream() for creating audio_stream objects.

Other audio functions: close_audio_device(), init_audio_device(), is_audio_device_ready(), is_audio_stream_playing(), is_audio_stream_processed(), load_audio_stream(), pause_audio_stream(), play_audio_stream(), resume_audio_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_volume(), stop_audio_stream(), unload_audio_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_audio_stream(), pause_music_stream(), play_audio_stream(), play_music_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), unload_music_stream(), update_music_stream()

---

set_audio_stream_volume

Set audio stream volume

Description

Set volume for audio stream (1.0 is max level).

Usage

```
set_audio_stream_volume(stream, volume)
```

Arguments

stream	An audio_stream.
volume	A number.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetAudioStreamVolume(AudioStream stream, float volume);
```

See Also

audio_stream() for creating audio_stream objects.

Other audio functions: close_audio_device(), init_audio_device(), is_audio_device_ready(), is_audio_stream_playing(), is_audio_stream_processed(), load_audio_stream(), pause_audio_stream(), play_audio_stream(), resume_audio_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), stop_audio_stream(), unload_audio_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_audio_stream(), pause_music_stream(), play_audio_stream(), play_music_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), unload_music_stream(), update_music_stream()

Other volume functions: set_master_volume(), set_music_volume(), set_sound_volume()

---

set_camera_alt_control

Set camera alt control

Description

Set camera alt key to combine with mouse movement (free camera).

Usage

```
set_camera_alt_control(key_alt)
```

Arguments

key_alt	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetCameraAltControl(int keyAlt);
```

See Also

Other camera functions: get_camera_matrix_2d(), get_camera_matrix(), set_camera_mode(), set_camera_move_controls(), set_camera_pan_control(), set_camera_smooth_zoom_control(), update_camera()

set_camera_mode *Set camera mode*

Description

Set camera mode (multiple camera modes available).

Usage

```
set_camera_mode(camera, mode)
```

Arguments

camera	A camera_3d.
mode	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetCameraMode(Camera camera, int mode);
```

See Also

camera_3d() for creating camera_3d objects.

Other camera functions: get_camera_matrix_2d(), get_camera_matrix(), set_camera_alt_control(), set_camera_move_controls(), set_camera_pan_control(), set_camera_smooth_zoom_control(), update_camera()

Other mode functions: begin_blend_mode(), begin_mode_2d(), begin_mode_3d(), begin_scissor_mode(), begin_shader_mode(), begin_texture_mode(), end_blend_mode(), end_mode_2d(), end_mode_3d(), end_scissor_mode(), end_shader_mode(), end_texture_mode(), end_vr_stereo_mode()

set_camera_move_controls *Set camera move controls*

Description

Set camera move controls (1st person and 3rd person cameras).

Usage

```
set_camera_move_controls(
  key_front,
  key_back,
  key_right,
  key_left,
  key_up,
  key_down
)
```

Arguments

key_front	An integer.
key_back	An integer.
key_right	An integer.
key_left	An integer.
key_up	An integer.
key_down	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetCameraMoveControls(int keyFront, int keyBack, int keyRight, int keyLeft, int keyUp, int keyDown
```

See Also

Other camera functions: get_camera_matrix_2d(), get_camera_matrix(), set_camera_alt_control(), set_camera_mode(), set_camera_pan_control(), set_camera_smooth_zoom_control(), update_camera()

---

set_camera_pan_control

Set camera pan control

Description

Set camera pan key to combine with mouse movement (free camera).

Usage

```
set_camera_pan_control(key_pan)
```

Arguments

key_pan	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetCameraPanControl(int keyPan);
```

See Also

Other camera functions: get_camera_matrix_2d(), get_camera_matrix(), set_camera_alt_control(), set_camera_mode(), set_camera_move_controls(), set_camera_smooth_zoom_control(), update_camera()

---

set_camera_smooth_zoom_control

Set camera smooth zoom control

Description

Set camera smooth zoom key to combine with mouse (free camera).

Usage

```
set_camera_smooth_zoom_control(key_smooth_zoom)
```

Arguments

key_smooth_zoom	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetCameraSmoothZoomControl(int keySmoothZoom);
```

See Also

Other camera functions: get_camera_matrix_2d(), get_camera_matrix(), set_camera_alt_control(), set_camera_mode(), set_camera_move_controls(), set_camera_pan_control(), update_camera()

set_exit_key *Set exit key*

Description

Set a custom key to exit program (default is ESC).

Usage

```
set_exit_key(key)
```

Arguments

key	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetExitKey(int key);
```

See Also

Other key functions: get_char_pressed(), get_key_pressed(), is_key_down(), is_key_pressed(), is_key_released(), is_key_up()

set_gamepad_mappings *Set gamepad mappings*

Description

Set internal gamepad mappings (SDL_GameControllerDB).

Usage

```
set_gamepad_mappings(mappings)
```

Arguments

mappings	A string.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int SetGamepadMappings(const char *mappings);
```

See Also

Other gamepad functions: get_gamepad_axis_count(), get_gamepad_axis_movement(), get_gamepad_button_presse
get_gamepad_name(), is_gamepad_available(), is_gamepad_button_down(), is_gamepad_button_pressed(),
is_gamepad_button_released(), is_gamepad_button_up()

---

set_gestures_enabled *Set gestures enabled*

Description

Enable a set of gestures using flags.

Usage

```
set_gestures_enabled(flags)
```

Arguments

flags	A non-negative integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetGesturesEnabled(unsigned int flags);
```

---

set_master_volume *Set master volume*

Description

Set master volume (listener).

Usage

```
set_master_volume(volume)
```

Arguments

volume	A number.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetMasterVolume(float volume);
```

See Also

Other volume functions: `set_audio_stream_volume()`, `set_music_volume()`, `set_sound_volume()`

`set_mouse_cursor` *Set mouse cursor*

Description

Set mouse cursor.

Usage

```
set_mouse_cursor(cursor)
```

Arguments

`cursor`	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetMouseCursor(int cursor);
```

See Also

Other mouse functions: `get_mouse_delta()`, `get_mouse_position()`, `get_mouse_wheel_move()`, `get_mouse_x()`, `get_mouse_y()`, `is_mouse_button_down()`, `is_mouse_button_pressed()`, `is_mouse_button_release` `is_mouse_button_up()`, `set_mouse_offset()`, `set_mouse_position()`, `set_mouse_scale()`

Other cursor functions: `disable_cursor()`, `enable_cursor()`, `hide_cursor()`, `is_cursor_hidden()`, `is_cursor_on_screen()`, `show_cursor()`

set_mouse_offset *Set mouse offset*

Description

Set mouse offset.

Usage

```
set_mouse_offset(offset_x, offset_y)
```

Arguments

offset_x	An integer.
offset_y	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetMouseOffset(int offsetX, int offsetY);
```

See Also

Other mouse functions: get_mouse_delta(), get_mouse_position(), get_mouse_wheel_move(), get_mouse_x(), get_mouse_y(), is_mouse_button_down(), is_mouse_button_pressed(), is_mouse_button_release is_mouse_button_up(), set_mouse_cursor(), set_mouse_position(), set_mouse_scale()

set_mouse_position *Set mouse position*

Description

Set mouse position XY.

Usage

```
set_mouse_position(x, y)
```

Arguments

x	An integer.
y	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetMousePosition(int x, int y);
```

See Also

Other mouse functions: get_mouse_delta(), get_mouse_position(), get_mouse_wheel_move(), get_mouse_x(), get_mouse_y(), is_mouse_button_down(), is_mouse_button_pressed(), is_mouse_button_release is_mouse_button_up(), set_mouse_cursor(), set_mouse_offset(), set_mouse_scale()

---

set_mouse_scale *Set mouse scale*

Description

Set mouse scaling.

Usage

```
set_mouse_scale(scale_x, scale_y)
```

Arguments

scale_x	A number.
scale_y	A number.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetMouseScale(float scaleX, float scaleY);
```

See Also

Other mouse functions: get_mouse_delta(), get_mouse_position(), get_mouse_wheel_move(), get_mouse_x(), get_mouse_y(), is_mouse_button_down(), is_mouse_button_pressed(), is_mouse_button_release is_mouse_button_up(), set_mouse_cursor(), set_mouse_offset(), set_mouse_position()

set_music_pitch *Set music pitch*

Description

Set pitch for a music (1.0 is base level).

Usage

```
set_music_pitch(music, pitch)
```

Arguments

music	A music.
pitch	A number.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetMusicPitch(Music music, float pitch);
```

See Also

music() for creating music objects.

Other music functions: get_music_time_length(), get_music_time_played(), is_music_stream_playing(), load_music_stream(), pause_music_stream(), play_music_stream(), resume_music_stream(), seek_music_stream(), set_music_volume(), stop_music_stream(), unload_music_stream(), update_music_stream()

set_music_volume *Set music volume*

Description

Set volume for music (1.0 is max level).

Usage

```
set_music_volume(music, volume)
```

Arguments

music	A music.
volume	A number.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetMusicVolume(Music music, float volume);
```

See Also

music() for creating music objects.

Other music functions: get_music_time_length(), get_music_time_played(), is_music_stream_playing(), load_music_stream(), pause_music_stream(), play_music_stream(), resume_music_stream(), seek_music_stream(), set_music_pitch(), stop_music_stream(), unload_music_stream(), update_music_stream()

Other volume functions: set_audio_stream_volume(), set_master_volume(), set_sound_volume()

---

set_shader_value *Set shader value*

Description

Set shader uniform value.

Usage

```
set_shader_value(shader, loc_index, value)
```

Arguments

shader	A shader.
loc_index	An integer.
value	A number or numerical vector of length 2, 3, or 4.

See Also

Other shader functions: begin_shader_mode(), end_shader_mode(), get_shader_location_attrib(), get_shader_location(), load_shader_from_memory(), load_shader(), set_shader_value_matrix(), set_shader_value_texture(), unload_shader()

set_shader_value_matrix *Set shader value matrix*

Description

Set shader uniform value (matrix 4x4).

Usage

```
set_shader_value_matrix(shader, loc_index, mat)
```

Arguments

shader	A shader.
loc_index	An integer.
mat	A numeric matrix of 4 by 4.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetShaderValueMatrix(Shader shader, int locIndex, Matrix mat);
```

See Also

shader() for creating shader objects.

Other shader functions: begin_shader_mode(), end_shader_mode(), get_shader_location_attrib(), get_shader_location(), load_shader_from_memory(), load_shader(), set_shader_value_texture(), set_shader_value(), unload_shader()

set_shader_value_texture *Set shader value texture*

Description

Set shader uniform value for texture (sampler2d).

Usage

```
set_shader_value_texture(shader, loc_index, texture)
```

Arguments

shader	A shader.
loc_index	An integer.
texture	A texture.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetShaderValueTexture(Shader shader, int locIndex, Texture2D texture);
```

See Also

shader() for creating shader objects.

texture() for creating texture objects.

Other shader functions: begin_shader_mode(), end_shader_mode(), get_shader_location_attrib(), get_shader_location(), load_shader_from_memory(), load_shader(), set_shader_value_matrix(), set_shader_value(), unload_shader()

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

---

set_shapes_texture *Set shapes texture*

Description

Set texture and rectangle to be used on shapes drawing.

Usage

```
set_shapes_texture(texture, source)
```

Arguments

texture	A texture.
source	A rectangle.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetShapesTexture(Texture2D texture, Rectangle source);
```

See Also

texture() for creating texture objects.

rectangle() for creating rectangle objects.

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture(), unload_texture()

---

set_sound_pitch *Set sound pitch*

Description

Set pitch for a sound (1.0 is base level).

Usage

```
set_sound_pitch(sound, pitch)
```

Arguments

sound	A sound.
pitch	A number.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetSoundPitch(Sound sound, float pitch);
```

See Also

sound() for creating sound objects.

Other sound functions: is_sound_playing(), load_sound_from_wave(), load_sound(), pause_sound(), play_sound_multi(), play_sound(), resume_sound(), set_sound_volume(), sound(), stop_sound_multi(), stop_sound(), unload_sound()

set_sound_volume *Set sound volume*

Description

Set volume for a sound (1.0 is max level).

Usage

```
set_sound_volume(sound, volume)
```

Arguments

sound	A sound.
volume	A number.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetSoundVolume(Sound sound, float volume);
```

See Also

sound() for creating sound objects.

Other sound functions: is_sound_playing(), load_sound_from_wave(), load_sound(), pause_sound(), play_sound_multi(), play_sound(), resume_sound(), set_sound_pitch(), sound(), stop_sound_multi(), stop_sound(), unload_sound()

Other volume functions: set_audio_stream_volume(), set_master_volume(), set_music_volume()

set_target_fps *Set target fps*

Description

Set target FPS (maximum).

Usage

```
set_target_fps(fps)
```

Arguments

fps	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetTargetFPS(int fps);
```

See Also

Other fps functions: `draw_fps()`, `get_fps()`

`set_texture_filter` *Set texture filter*

Description

Set texture scaling filter mode.

Usage

```
set_texture_filter(texture, filter)
```

Arguments

`texture`	A texture.
`filter`	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetTextureFilter(Texture2D texture, int filter);
```

See Also

`texture()` for creating texture objects.

Other texture functions: `begin_texture_mode()`, `draw_cube_texture_rec()`, `draw_cube_texture()`, `draw_texture_ex()`, `draw_texture_npatch()`, `draw_texture_poly()`, `draw_texture_pro()`, `draw_texture_quad()`, `draw_texture_rec()`, `draw_texture_tiled()`, `draw_texture_v()`, `draw_texture()`, `end_texture_mode()`, `gen_texture_mipmaps()`, `load_image_from_texture()`, `load_render_texture()`, `load_texture_from_image()`, `load_texture()`, `set_shader_value_texture()`, `set_shapes_texture()`, `set_texture_wrap()`, `texture()`, `unload_render_texture()`, `unload_texture()`

set_texture_wrap *Set texture wrap*

Description

Set texture wrapping mode.

Usage

```
set_texture_wrap(texture, wrap)
```

Arguments

texture	A texture.
wrap	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetTextureWrap(Texture2D texture, int wrap);
```

See Also

texture() for creating texture objects.

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), texture(), unload_render_texture(), unload_texture()

set_trace_log_level *Set trace log level*

Description

Set the current threshold (minimum) log level.

Usage

```
set_trace_log_level(log_level)
```

Arguments

log_level	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetTraceLogLevel(int logLevel);
```

`set_window_icon` *Set window icon*

Description

Set icon for window (only PLATFORM_DESKTOP).

Usage

```
set_window_icon(image)
```

Arguments

`image`	An image.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetWindowIcon(Image image);
```

See Also

`image()` for creating image objects.

Other window functions: `clear_window_state()`, `close_window()`, `get_window_position()`, `get_window_scale_dpi()`, `init_window()`, `is_window_focused()`, `is_window_fullscreen()`, `is_window_hidden()`, `is_window_maximized()`, `is_window_minimized()`, `is_window_ready()`, `is_window_resized()`, `is_window_state()`, `maximize_window()`, `minimize_window()`, `restore_window()`, `set_window_min_size()`, `set_window_monitor()`, `set_window_position()`, `set_window_size()`, `set_window_state()`, `set_window_title()`, `window_should_close()`

set_window_min_size *Set window min size*

Description

Set window minimum dimensions (for FLAG_WINDOW_RESIZABLE).

Usage

```
set_window_min_size(width, height)
```

Arguments

width	An integer.
height	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetWindowMinSize(int width, int height);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

set_window_monitor *Set window monitor*

Description

Set monitor for the current window (fullscreen mode).

Usage

```
set_window_monitor(monitor)
```

Arguments

monitor	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetWindowMonitor(int monitor);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_position(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

Other monitor functions: get_current_monitor(), get_monitor_count(), get_monitor_height(), get_monitor_name(), get_monitor_physical_height(), get_monitor_physical_width(), get_monitor_position(), get_monitor_refresh_rate(), get_monitor_width()

---

set_window_position *Set window position*

Description

Set window position on screen (only PLATFORM_DESKTOP).

Usage

```
set_window_position(x, y)
```

Arguments

x	An integer.
y	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetWindowPosition(int x, int y);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_size(), set_window_state(), set_window_title(), window_should_close()

set_window_size *Set window size*

Description

Set window dimensions.

Usage

```
set_window_size(width, height)
```

Arguments

width	An integer.
height	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetWindowSize(int width, int height);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_state(), set_window_title(), window_should_close()

set_window_state *Set window state*

Description

Set window configuration state using flags.

Usage

```
set_window_state(flags)
```

Arguments

flags	A non-negative integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetWindowState(unsigned int flags);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_title(), window_should_close()

---

set_window_title *Set window title*

Description

Set title for window (only PLATFORM_DESKTOP).

Usage

```
set_window_title(title)
```

Arguments

title	A string.

Note

This function has been auto-generated from the following Raylib function definition:

```
void SetWindowTitle(const char *title);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), window_should_close()

show_cursor *Show cursor*

Description

Shows cursor.

Usage

```
show_cursor()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void ShowCursor(void);
```

See Also

Other cursor functions: `disable_cursor()`, `enable_cursor()`, `hide_cursor()`, `is_cursor_hidden()`, `is_cursor_on_screen()`, `set_mouse_cursor()`

sound *Sound*

Description

Create a new sound object.

Usage

```
sound(stream, frame_count)
```

Arguments

`stream`	An audio_stream. Audio stream.
`frame_count`	A non-negative integer. Total number of frames (considering channels).

Value

A sound

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct Sound {
    AudioStream stream;         // Audio stream
    unsigned int frameCount;    // Total number of frames (considering channels)
} Sound;
```

See Also

Other sound functions: `is_sound_playing()`, `load_sound_from_wave()`, `load_sound()`, `pause_sound()`, `play_sound_multi()`, `play_sound()`, `resume_sound()`, `set_sound_pitch()`, `set_sound_volume()`, `stop_sound_multi()`, `stop_sound()`, `unload_sound()`

---

stop_audio_stream *Stop audio stream*

Description

Stop audio stream.

Usage

```
stop_audio_stream(stream)
```

Arguments

`stream`	An audio_stream.

Note

This function has been auto-generated from the following Raylib function definition:

```
void StopAudioStream(AudioStream stream);
```

See Also

`audio_stream()` for creating audio_stream objects.

Other audio functions: `close_audio_device()`, `init_audio_device()`, `is_audio_device_ready()`, `is_audio_stream_playing()`, `is_audio_stream_processed()`, `load_audio_stream()`, `pause_audio_stream()`, `play_audio_stream()`, `resume_audio_stream()`, `set_audio_stream_buffer_size_default()`, `set_audio_stream_pitch()`, `set_audio_stream_volume()`, `unload_audio_stream()`

Other stream functions: `is_audio_stream_playing()`, `is_audio_stream_processed()`, `is_music_stream_playing()`, `load_audio_stream()`, `load_music_stream()`, `pause_audio_stream()`, `pause_music_stream()`, `play_audio_stream()`, `play_music_stream()`, `resume_audio_stream()`, `resume_music_stream()`, `seek_music_stream()`, `set_audio_stream_buffer_size_default()`, `set_audio_stream_pitch()`, `set_audio_stream_volume()`, `stop_music_stream()`, `unload_audio_stream()`, `unload_music_stream()`, `update_music_stream()`

stop_music_stream *Stop music stream*

Description

Stop music playing.

Usage

```
stop_music_stream(music)
```

Arguments

music	A music.

Note

This function has been auto-generated from the following Raylib function definition:

```
void StopMusicStream(Music music);
```

See Also

music() for creating music objects.

Other music functions: get_music_time_length(), get_music_time_played(), is_music_stream_playing(), load_music_stream(), pause_music_stream(), play_music_stream(), resume_music_stream(), seek_music_stream(), set_music_pitch(), set_music_volume(), unload_music_stream(), update_music_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_audio_stream(), pause_music_stream(), play_audio_stream(), play_music_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), unload_audio_stream(), unload_music_stream(), update_music_stream()

stop_sound *Stop sound*

Description

Stop playing a sound.

Usage

```
stop_sound(sound)
```

Arguments

sound	A sound.

Note

This function has been auto-generated from the following Raylib function definition:

```
void StopSound(Sound sound);
```

See Also

sound() for creating sound objects.

Other sound functions: is_sound_playing(), load_sound_from_wave(), load_sound(), pause_sound(), play_sound_multi(), play_sound(), resume_sound(), set_sound_pitch(), set_sound_volume(), sound(), stop_sound_multi(), unload_sound()

---

stop_sound_multi *Stop sound multi*

Description

Stop any sound playing (using multichannel buffer pool).

Usage

```
stop_sound_multi()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void StopSoundMulti(void);
```

See Also

Other sound functions: is_sound_playing(), load_sound_from_wave(), load_sound(), pause_sound(), play_sound_multi(), play_sound(), resume_sound(), set_sound_pitch(), set_sound_volume(), sound(), stop_sound(), unload_sound()

take_screenshot *Take screenshot*

Description

Takes a screenshot of current screen (filename extension defines format).

Usage

```
take_screenshot(file_name)
```

Arguments

file_name	A string.

Note

This function has been auto-generated from the following Raylib function definition:

```
void TakeScreenshot(const char *fileName);
```

text_find_index *Text find index*

Description

Find first text occurrence within a string.

Usage

```
text_find_index(text, find)
```

Arguments

text	A string.
find	A string.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int TextFindIndex(const char *text, const char *find);
```

See Also

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_insert(), text_replace(), text_to_integer(), text_to_lower(), text_to_pascal(), text_to_upper()

---

text_insert *Text insert*

Description

Insert text in a position (WARNING: memory must be freed!).

Usage

```
text_insert(text, insert, position)
```

Arguments

text	A string.
insert	A string.
position	An integer.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
char *TextInsert(const char *text, const char *insert, int position);
```

See Also

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_find_index(), text_replace(), text_to_integer(), text_to_lower(), text_to_pascal(), text_to_upper()

text_replace *Text replace*

Description

Replace text string (WARNING: memory must be freed!).

Usage

```
text_replace(text, replace, by)
```

Arguments

text	A string.
replace	A string.
by	A string.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
char *TextReplace(char *text, const char *replace, const char *by);
```

See Also

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_find_index(), text_insert(), text_to_integer(), text_to_lower(), text_to_pascal(), text_to_upper()

text_to_integer *Text to integer*

Description

Get integer value from text (negative values not supported).

Usage

```
text_to_integer(text)
```

Arguments

text	A string.

Value

An integer

Note

This function has been auto-generated from the following Raylib function definition:

```
int TextToInteger(const char *text);
```

See Also

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_find_index(), text_insert(), text_replace(), text_to_lower(), text_to_pascal(), text_to_upper()

---

text_to_lower *Text to lower*

Description

Get lower case version of provided string.

Usage

```
text_to_lower(text)
```

Arguments

text	A string.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
const char *TextToLower(const char *text);
```

See Also

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_find_index(), text_insert(), text_replace(), text_to_integer(), text_to_pascal(), text_to_upper()

`text_to_pascal` *Text to pascal*

Description

Get Pascal case notation version of provided string.

Usage

```
text_to_pascal(text)
```

Arguments

`text`	A string.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
const char *TextToPascal(const char *text);
```

See Also

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_find_index(), text_insert(), text_replace(), text_to_integer(), text_to_lower(), text_to_upper()

text_to_upper *Text to upper*

Description

Get upper case version of provided string.

Usage

```
text_to_upper(text)
```

Arguments

text	A string.

Value

A string

Note

This function has been auto-generated from the following Raylib function definition:

```
const char *TextToUpper(const char *text);
```

See Also

Other text functions: draw_text_codepoint(), draw_text_ex(), draw_text_pro(), draw_text(), image_draw_text_ex(), image_draw_text(), image_text_ex(), image_text(), load_file_text(), measure_text_ex(), measure_text(), text_find_index(), text_insert(), text_replace(), text_to_integer(), text_to_lower(), text_to_pascal()

texture *Texture*

Description

Create a new texture object.

Usage

```
texture(id, width, height, mipmaps, format)
```

Arguments

id	A non-negative integer. OpenGL texture id.
width	An integer. Texture base width.
height	An integer. Texture base height.
mipmaps	An integer. Mipmap levels, 1 by default.
format	An integer. Data format (PixelFormat type).

Value

A texture

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct Texture {
    unsigned int id;        // OpenGL texture id
    int width;              // Texture base width
    int height;             // Texture base height
    int mipmaps;            // Mipmap levels, 1 by default
    int format;             // Data format (PixelFormat type)
} Texture;
```

See Also

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), unload_render_texture(), unload_texture()

---

toggle_fullscreen *Toggle fullscreen*

Description

Toggle window state: fullscreen/windowed (only PLATFORM_DESKTOP).

Usage

```
toggle_fullscreen()
```

Note

This function has been auto-generated from the following Raylib function definition:

```
void ToggleFullscreen(void);
```

transform *Transform*

Description

Create a new transform object.

Usage

```
transform(translation, rotation, scale)
```

Arguments

translation	A numeric vector of length 3. Translation.
rotation	A numeric vector of length 4. Rotation.
scale	A numeric vector of length 3. Scale.

Value

A transform

Note

This class has been auto-generated from the following Raylib struct definition:

```
typedef struct Transform {
    Vector3 translation;    // Translation
    Quaternion rotation;    // Rotation
    Vector3 scale;          // Scale
} Transform;
```

unload_audio_stream *Unload audio stream*

Description

Unload audio stream and free memory.

Usage

```
unload_audio_stream(stream)
```

Arguments

stream	An audio_stream.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadAudioStream(AudioStream stream);
```

See Also

audio_stream() for creating audio_stream objects.

Other audio functions: close_audio_device(), init_audio_device(), is_audio_device_ready(), is_audio_stream_playing(), is_audio_stream_processed(), load_audio_stream(), pause_audio_stream(), play_audio_stream(), resume_audio_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_audio_stream(), pause_music_stream(), play_audio_stream(), play_music_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_music_stream(), update_music_stream()

---

unload_font *Unload font*

Description

Unload Font from GPU memory (VRAM).

Usage

```
unload_font(font)
```

Arguments

font	A font.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadFont(Font font);
```

See Also

font() for creating font objects.

Other font functions: get_font_default(), load_font_ex(), load_font_from_image(), load_font(), unload_font_data()

unload_font_data *Unload font data*

Description

Unload font chars info data (RAM).

Usage

```
unload_font_data(chars, glyph_count)
```

Arguments

chars	A glyph_info.
glyph_count	An integer.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadFontData(GlyphInfo *chars, int glyphCount);
```

See Also

Other font functions: get_font_default(), load_font_ex(), load_font_from_image(), load_font(), unload_font()

unload_image *Unload image*

Description

Unload image from CPU memory (RAM).

Usage

```
unload_image(image)
```

Arguments

image	An image.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadImage(Image image);
```

See Also

image() for creating image objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image_palette()

---

unload_image_colors *Unload image colors*

Description

Unload color data loaded with LoadImageColors().

Usage

```
unload_image_colors(colors)
```

Arguments

colors	A color.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadImageColors(Color *colors);
```

See Also

color() for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(),

image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_palette(), unload_image()

`unload_image_palette` *Unload image palette*

Description

Unload colors palette loaded with LoadImagePalette().

Usage

```
unload_image_palette(colors)
```

Arguments

`colors`	A color.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadImagePalette(Color *colors);
```

See Also

`color()` for creating color objects.

Other image functions: export_image_as_code(), export_image(), gen_image_cellular(), gen_image_checked(), gen_image_color(), gen_image_gradient_h(), gen_image_gradient_radial(), gen_image_gradient_v(), gen_image_white_noise(), get_image_alpha_border(), get_image_color(), image_alpha_clear(), image_alpha_crop(), image_alpha_mask(), image_alpha_premultiply(), image_clear_background(), image_color_brightness(), image_color_contrast(), image_color_grayscale(), image_color_invert(), image_color_replace(), image_color_tint(), image_copy(), image_crop(), image_dither(), image_draw_circle_v(), image_draw_circle(), image_draw_line_v(), image_draw_line(), image_draw_pixel_v(), image_draw_pixel(), image_draw_rectangle_lines(), image_draw_rectangle_rec(), image_draw_rectangle_v(), image_draw_rectangle(), image_draw_text_ex(), image_draw_text(), image_draw(), image_flip_horizontal(), image_flip_vertical(), image_format(), image_from_image(), image_mipmaps(), image_resize_canvas(), image_resize_nn(), image_resize(), image_rotate_ccw(), image_rotate_cw(), image_text_ex(), image_text(), image_to_pot(), load_font_from_image(), load_image_from_screen(), load_image_from_texture(), load_image_raw(), load_image(), load_texture_from_image(), unload_image_colors(), unload_image()

unload_model *Unload model*

Description

Unload model (including meshes) from memory (RAM and/or VRAM).

Usage

```
unload_model(model)
```

Arguments

model	A model.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadModel(Model model);
```

See Also

`model()` for creating model objects.

Other model functions: `draw_model_ex()`, `draw_model_wires()`, `draw_model()`, `get_model_bounding_box()`, `get_ray_collision_model()`, `load_model()`

unload_music_stream *Unload music stream*

Description

Unload music stream.

Usage

```
unload_music_stream(music)
```

Arguments

music	A music.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadMusicStream(Music music);
```

See Also

music() for creating music objects.

Other music functions: get_music_time_length(), get_music_time_played(), is_music_stream_playing(), load_music_stream(), pause_music_stream(), play_music_stream(), resume_music_stream(), seek_music_stream(), set_music_pitch(), set_music_volume(), stop_music_stream(), update_music_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_audio_stream(), pause_music_stream(), play_audio_stream(), play_music_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), update_music_stream()

---

unload_render_texture *Unload render texture*

Description

Unload render texture from GPU memory (VRAM).

Usage

```
unload_render_texture(target)
```

Arguments

target	A render_texture.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadRenderTexture(RenderTexture2D target);
```

See Also

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_texture()

unload_shader *Unload shader*

Description

Unload shader from GPU memory (VRAM).

Usage

```
unload_shader(shader)
```

Arguments

shader	A shader.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadShader(Shader shader);
```

See Also

`shader()` for creating shader objects.

Other shader functions: `begin_shader_mode()`, `end_shader_mode()`, `get_shader_location_attrib()`, `get_shader_location()`, `load_shader_from_memory()`, `load_shader()`, `set_shader_value_matrix()`, `set_shader_value_texture()`, `set_shader_value()`

unload_sound *Unload sound*

Description

Unload sound.

Usage

```
unload_sound(sound)
```

Arguments

sound	A sound.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadSound(Sound sound);
```

See Also

sound() for creating sound objects.

Other sound functions: is_sound_playing(), load_sound_from_wave(), load_sound(), pause_sound(), play_sound_multi(), play_sound(), resume_sound(), set_sound_pitch(), set_sound_volume(), sound(), stop_sound_multi(), stop_sound()

---

unload_texture *Unload texture*

Description

Unload texture from GPU memory (VRAM).

Usage

```
unload_texture(texture)
```

Arguments

texture	A texture.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadTexture(Texture2D texture);
```

See Also

texture() for creating texture objects.

Other texture functions: begin_texture_mode(), draw_cube_texture_rec(), draw_cube_texture(), draw_texture_ex(), draw_texture_npatch(), draw_texture_poly(), draw_texture_pro(), draw_texture_quad(), draw_texture_rec(), draw_texture_tiled(), draw_texture_v(), draw_texture(), end_texture_mode(), gen_texture_mipmaps(), load_image_from_texture(), load_render_texture(), load_texture_from_image(), load_texture(), set_shader_value_texture(), set_shapes_texture(), set_texture_filter(), set_texture_wrap(), texture(), unload_render_texture()

unload_wave *Unload wave*

Description

Unload wave data.

Usage

```
unload_wave(wave)
```

Arguments

wave	A wave.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UnloadWave(Wave wave);
```

See Also

wave() for creating wave objects.

Other wave functions: export_wave_as_code(), export_wave(), load_sound_from_wave(), load_wave()

update_camera *Update camera*

Description

Update camera position for selected mode.

Usage

```
update_camera(camera)
```

Arguments

camera	A camera_3d.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UpdateCamera(Camera *camera);
```

See Also

camera_3d() for creating camera_3d objects.

Other camera functions: get_camera_matrix_2d(), get_camera_matrix(), set_camera_alt_control(), set_camera_mode(), set_camera_move_controls(), set_camera_pan_control(), set_camera_smooth_zoom_control

---

`update_music_stream` *Update music stream*

Description

Updates buffers for music streaming.

Usage

```
update_music_stream(music)
```

Arguments

`music`	A music.

Note

This function has been auto-generated from the following Raylib function definition:

```
void UpdateMusicStream(Music music);
```

See Also

music() for creating music objects.

Other music functions: get_music_time_length(), get_music_time_played(), is_music_stream_playing(), load_music_stream(), pause_music_stream(), play_music_stream(), resume_music_stream(), seek_music_stream(), set_music_pitch(), set_music_volume(), stop_music_stream(), unload_music_stream()

Other stream functions: is_audio_stream_playing(), is_audio_stream_processed(), is_music_stream_playing(), load_audio_stream(), load_music_stream(), pause_audio_stream(), pause_music_stream(), play_audio_stream(), play_music_stream(), resume_audio_stream(), resume_music_stream(), seek_music_stream(), set_audio_stream_buffer_size_default(), set_audio_stream_pitch(), set_audio_stream_volume(), stop_audio_stream(), stop_music_stream(), unload_audio_stream(), unload_music_stream()

vector_2_angle *Vector 2 angle*

Description

.

Usage

```
vector_2_angle(v_1, v_2)
```

Arguments

v_1	A numeric vector of length 2.
v_2	A numeric vector of length 2.

Value

A number

Note

This function has been auto-generated from the following Raylib function definition:

```
float Vector2Angle(Vector2 v1, Vector2 v2);
```

vector_2_distance *Vector 2 distance*

Description

.

Usage

```
vector_2_distance(v_1, v_2)
```

Arguments

v_1	A numeric vector of length 2.
v_2	A numeric vector of length 2.

Value

A number

Note

This function has been auto-generated from the following Raylib function definition:

```
float Vector2Distance(Vector2 v1, Vector2 v2);
```

`vector_2_move_towards` *Vector 2 move towards*

Description

.

Usage

```
vector_2_move_towards(v, target, max_distance)
```

Arguments

`v`	A numeric vector of length 2.
`target`	A numeric vector of length 2.
`max_distance`	A number.

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 Vector2MoveTowards(Vector2 v, Vector2 target, float maxDistance);
```

vector_2_rotate *Vector 2 rotate*

Description

.

Usage

```
vector_2_rotate(v, angle)
```

Arguments

v	A numeric vector of length 2.
angle	A number.

Value

A numeric vector of length 2

Note

This function has been auto-generated from the following Raylib function definition:

```
Vector2 Vector2Rotate(Vector2 v, float angle);
```

window_should_close *Window should close*

Description

Check if KEY_ESCAPE pressed or Close icon pressed.

Usage

```
window_should_close()
```

Value

A logical

Note

This function has been auto-generated from the following Raylib function definition:

```
bool WindowShouldClose(void);
```

See Also

Other window functions: clear_window_state(), close_window(), get_window_position(), get_window_scale_dpi(), init_window(), is_window_focused(), is_window_fullscreen(), is_window_hidden(), is_window_maximized(), is_window_minimized(), is_window_ready(), is_window_resized(), is_window_state(), maximize_window(), minimize_window(), restore_window(), set_window_icon(), set_window_min_size(), set_window_monitor(), set_window_position(), set_window_size(), set_window_state(), set_window_title()

Index

* **audio functions**
  close_audio_device, 29
  init_audio_device, 235
  is_audio_device_ready, 236
  is_audio_stream_playing, 237
  is_audio_stream_processed, 238
  load_audio_stream, 258
  pause_audio_stream, 278
  play_audio_stream, 280
  resume_audio_stream, 287
  set_audio_stream_buffer_size_default, 291
  set_audio_stream_pitch, 291
  set_audio_stream_volume, 292
  stop_audio_stream, 316
  unload_audio_stream, 326
* **bounding_box functions**
  bounding_box, 15
* **camera functions**
  get_camera_matrix, 139
  get_camera_matrix_2d, 140
  set_camera_alt_control, 293
  set_camera_mode, 294
  set_camera_move_controls, 294
  set_camera_pan_control, 295
  set_camera_smooth_zoom_control, 296
  update_camera, 335
* **camera_2d functions**
  camera_2d, 15
* **camera_3d functions**
  camera_3d, 16
* **circle functions**
  check_collision_circle_rec, 19
  check_collision_point_circle, 22
  draw_circle, 40
  draw_circle_3d, 41
  draw_circle_gradient, 42
  draw_circle_lines, 43
  draw_circle_sector, 44
  draw_circle_sector_lines, 46
  draw_circle_v, 47
  image_draw_circle, 205
  image_draw_circle_v, 206
* **collision functions**
  check_collision_box_sphere, 18
  check_collision_boxes, 19
  check_collision_circle_rec, 19
  check_collision_circles, 20
  check_collision_lines, 21
  check_collision_point_circle, 22
  check_collision_point_line, 23
  check_collision_point_rec, 24
  check_collision_point_triangle, 25
  check_collision_recs, 26
  check_collision_spheres, 26
  get_collision_rec, 141
  get_ray_collision_box, 171
  get_ray_collision_mesh, 172
  get_ray_collision_model, 173
  get_ray_collision_quad, 174
  get_ray_collision_sphere, 175
  get_ray_collision_triangle, 176
* **color functions**
  as_color, 10
  color, 30
  color_alpha, 31
  color_alpha_blend, 32
  color_from_hsv, 33
  color_from_normalized, 33
  color_normalize, 34
  color_to_hsv, 35
  color_to_int, 36
  gen_image_color, 133
  get_color, 142
  get_image_color, 157
  image_color_brightness, 195
  image_color_contrast, 196

image_color_grayscale, 197
image_color_invert, 198
image_color_replace, 199
image_color_tint, 200
∗ **cursor functions**
disable_cursor, 37
enable_cursor, 122
hide_cursor, 190
is_cursor_hidden, 239
is_cursor_on_screen, 239
set_mouse_cursor, 299
show_cursor, 315
∗ **directory functions**
change_directory, 17
clear_directory_files, 28
directory_exists, 36
get_directory_path, 143
get_prev_directory_path, 170
get_working_directory, 186
∗ **draw functions**
draw_billboard, 38
draw_bounding_box, 39
draw_circle, 40
draw_circle_3d, 41
draw_circle_gradient, 42
draw_circle_lines, 43
draw_circle_sector, 44
draw_circle_sector_lines, 46
draw_circle_v, 47
draw_cube, 48
draw_cube_texture, 49
draw_cube_texture_rec, 50
draw_cube_v, 52
draw_cube_wires, 53
draw_cube_wires_v, 54
draw_cylinder, 55
draw_cylinder_ex, 56
draw_cylinder_wires, 57
draw_cylinder_wires_ex, 58
draw_ellipse, 59
draw_ellipse_lines, 60
draw_fps, 61
draw_grid, 62
draw_line, 63
draw_line_3d, 64
draw_line_bezier, 65
draw_line_bezier_cubic, 66
draw_line_bezier_quad, 68
draw_line_ex, 69
draw_line_v, 70
draw_model, 71
draw_model_ex, 72
draw_model_wires, 74
draw_pixel, 75
draw_pixel_v, 76
draw_plane, 77
draw_point_3d, 78
draw_poly, 79
draw_poly_lines, 80
draw_poly_lines_ex, 81
draw_ray, 82
draw_rectangle, 83
draw_rectangle_gradient_ex, 84
draw_rectangle_gradient_h, 85
draw_rectangle_gradient_v, 86
draw_rectangle_lines, 88
draw_rectangle_lines_ex, 89
draw_rectangle_pro, 90
draw_rectangle_rec, 91
draw_rectangle_rounded, 92
draw_rectangle_rounded_lines, 94
draw_rectangle_v, 95
draw_ring, 96
draw_ring_lines, 97
draw_sphere, 99
draw_sphere_ex, 100
draw_sphere_wires, 101
draw_text, 102
draw_text_codepoint, 103
draw_text_ex, 104
draw_text_pro, 105
draw_texture, 107
draw_texture_ex, 108
draw_texture_npatch, 109
draw_texture_poly, 110
draw_texture_pro, 112
draw_texture_quad, 113
draw_texture_rec, 114
draw_texture_tiled, 115
draw_texture_v, 117
draw_triangle, 118
draw_triangle_3d, 119
draw_triangle_lines, 120
draw_triangle_strip_3d, 121
image_draw, 204
image_draw_circle, 205

image_draw_circle_v, 206
image_draw_line, 208
image_draw_line_v, 209
image_draw_pixel, 211
image_draw_pixel_v, 212
image_draw_rectangle, 213
image_draw_rectangle_lines, 215
image_draw_rectangle_rec, 217
image_draw_rectangle_v, 218
image_draw_text, 219
image_draw_text_ex, 221
∗ **file functions**
file_exists, 131
get_file_extension, 144
get_file_mod_time, 144
get_file_name, 145
get_file_name_without_ext, 146
is_file_dropped, 240
is_file_extension, 240
load_file_text, 259
∗ **font functions**
get_font_default, 146
load_font, 259
load_font_ex, 260
load_font_from_image, 261
unload_font, 327
unload_font_data, 328
∗ **fps functions**
draw_fps, 61
get_fps, 147
set_target_fps, 307
∗ **gamepad functions**
get_gamepad_axis_count, 148
get_gamepad_axis_movement, 149
get_gamepad_button_pressed, 150
get_gamepad_name, 150
is_gamepad_available, 241
is_gamepad_button_down, 242
is_gamepad_button_pressed, 242
is_gamepad_button_released, 243
is_gamepad_button_up, 244
set_gamepad_mappings, 297
∗ **glyph_info functions**
glyph_info, 189
∗ **image functions**
export_image, 127
export_image_as_code, 128
gen_image_cellular, 131
gen_image_checked, 132
gen_image_color, 133
gen_image_gradient_h, 134
gen_image_gradient_radial, 135
gen_image_gradient_v, 136
gen_image_white_noise, 137
get_image_alpha_border, 156
get_image_color, 157
image_alpha_clear, 190
image_alpha_crop, 191
image_alpha_mask, 192
image_alpha_premultiply, 193
image_clear_background, 194
image_color_brightness, 195
image_color_contrast, 196
image_color_grayscale, 197
image_color_invert, 198
image_color_replace, 199
image_color_tint, 200
image_copy, 201
image_crop, 202
image_dither, 203
image_draw, 204
image_draw_circle, 205
image_draw_circle_v, 206
image_draw_line, 208
image_draw_line_v, 209
image_draw_pixel, 211
image_draw_pixel_v, 212
image_draw_rectangle, 213
image_draw_rectangle_lines, 215
image_draw_rectangle_rec, 217
image_draw_rectangle_v, 218
image_draw_text, 219
image_draw_text_ex, 221
image_flip_horizontal, 222
image_flip_vertical, 223
image_format, 224
image_from_image, 225
image_mipmaps, 226
image_resize, 227
image_resize_canvas, 228
image_resize_nn, 229
image_rotate_ccw, 230
image_rotate_cw, 231
image_text, 232
image_text_ex, 233
image_to_pot, 234

load_font_from_image, 261
load_image, 262
load_image_from_screen, 263
load_image_from_texture, 264
load_image_raw, 265
load_texture_from_image, 272
unload_image, 328
unload_image_colors, 329
unload_image_palette, 330
* **init functions**
init_audio_device, 235
init_window, 235
* **key functions**
get_char_pressed, 140
get_key_pressed, 158
is_key_down, 245
is_key_pressed, 246
is_key_released, 247
is_key_up, 247
set_exit_key, 297
* **line functions**
check_collision_point_line, 23
draw_line, 63
draw_line_3d, 64
draw_line_bezier, 65
draw_line_bezier_cubic, 66
draw_line_bezier_quad, 68
draw_line_ex, 69
draw_line_v, 70
image_draw_line, 208
image_draw_line_v, 209
* **material_map functions**
material_map, 274
* **mesh functions**
get_ray_collision_mesh, 172
* **mode functions**
begin_blend_mode, 10
begin_mode_2d, 11
begin_mode_3d, 12
begin_scissor_mode, 13
begin_shader_mode, 13
begin_texture_mode, 14
end_blend_mode, 123
end_mode_2d, 124
end_mode_3d, 124
end_scissor_mode, 125
end_shader_mode, 125
end_texture_mode, 126
end_vr_stereo_mode, 126
set_camera_mode, 294
* **model functions**
draw_model, 71
draw_model_ex, 72
draw_model_wires, 74
get_model_bounding_box, 159
get_ray_collision_model, 173
load_model, 266
unload_model, 331
* **monitor functions**
get_current_monitor, 142
get_monitor_count, 159
get_monitor_height, 160
get_monitor_name, 161
get_monitor_physical_height, 161
get_monitor_physical_width, 162
get_monitor_position, 163
get_monitor_refresh_rate, 163
get_monitor_width, 164
set_window_monitor, 311
* **mouse functions**
get_mouse_delta, 165
get_mouse_position, 165
get_mouse_wheel_move, 166
get_mouse_x, 167
get_mouse_y, 167
is_mouse_button_down, 248
is_mouse_button_pressed, 249
is_mouse_button_released, 249
is_mouse_button_up, 250
set_mouse_cursor, 299
set_mouse_offset, 300
set_mouse_position, 300
set_mouse_scale, 301
* **music functions**
get_music_time_length, 168
get_music_time_played, 169
is_music_stream_playing, 251
load_music_stream, 266
pause_music_stream, 279
play_music_stream, 281
resume_music_stream, 288
seek_music_stream, 290
set_music_pitch, 302
set_music_volume, 302
stop_music_stream, 317
unload_music_stream, 331

update_music_stream, 336
* **npatch_info functions**
npatch_info, 277
* **ray functions**
ray, 283
* **ray_collision functions**
ray_collision, 284
* **rectangle functions**
draw_rectangle, 83
draw_rectangle_gradient_ex, 84
draw_rectangle_gradient_h, 85
draw_rectangle_gradient_v, 86
draw_rectangle_lines, 88
draw_rectangle_lines_ex, 89
draw_rectangle_pro, 90
draw_rectangle_rec, 91
draw_rectangle_rounded, 92
draw_rectangle_rounded_lines, 94
draw_rectangle_v, 95
image_draw_rectangle, 213
image_draw_rectangle_lines, 215
image_draw_rectangle_rec, 217
image_draw_rectangle_v, 218
rectangle, 285
* **render_texture functions**
render_texture, 286
* **screen functions**
get_screen_height, 177
get_screen_to_world_2d, 177
get_screen_width, 178
get_world_to_screen, 186
get_world_to_screen_2d, 187
get_world_to_screen_ex, 188
is_cursor_on_screen, 239
load_image_from_screen, 263
* **shader functions**
begin_shader_mode, 13
end_shader_mode, 125
get_shader_location, 179
get_shader_location_attrib, 179
load_shader, 268
load_shader_from_memory, 269
set_shader_value, 303
set_shader_value_matrix, 304
set_shader_value_texture, 304
unload_shader, 333
* **sound functions**
is_sound_playing, 252
load_sound, 269
load_sound_from_wave, 270
pause_sound, 280
play_sound, 282
play_sound_multi, 283
resume_sound, 289
set_sound_pitch, 306
set_sound_volume, 307
sound, 315
stop_sound, 317
stop_sound_multi, 318
unload_sound, 333
* **stream functions**
is_audio_stream_playing, 237
is_audio_stream_processed, 238
is_music_stream_playing, 251
load_audio_stream, 258
load_music_stream, 266
pause_audio_stream, 278
pause_music_stream, 279
play_audio_stream, 280
play_music_stream, 281
resume_audio_stream, 287
resume_music_stream, 288
seek_music_stream, 290
set_audio_stream_buffer_size_default, 291
set_audio_stream_pitch, 291
set_audio_stream_volume, 292
stop_audio_stream, 316
stop_music_stream, 317
unload_audio_stream, 326
unload_music_stream, 331
update_music_stream, 336
* **text functions**
draw_text, 102
draw_text_codepoint, 103
draw_text_ex, 104
draw_text_pro, 105
image_draw_text, 219
image_draw_text_ex, 221
image_text, 232
image_text_ex, 233
load_file_text, 259
measure_text, 275
measure_text_ex, 276
text_find_index, 319
text_insert, 320

text_replace, 321
text_to_integer, 321
text_to_lower, 322
text_to_pascal, 323
text_to_upper, 324
* **texture functions**
begin_texture_mode, 14
draw_cube_texture, 49
draw_cube_texture_rec, 50
draw_texture, 107
draw_texture_ex, 108
draw_texture_npatch, 109
draw_texture_poly, 110
draw_texture_pro, 112
draw_texture_quad, 113
draw_texture_rec, 114
draw_texture_tiled, 115
draw_texture_v, 117
end_texture_mode, 126
gen_texture_mipmaps, 138
load_image_from_texture, 264
load_render_texture, 267
load_texture, 271
load_texture_from_image, 272
set_shader_value_texture, 304
set_shapes_texture, 305
set_texture_filter, 308
set_texture_wrap, 309
texture, 324
unload_render_texture, 332
unload_texture, 334
* **touch functions**
get_touch_point_count, 181
get_touch_point_id, 182
get_touch_position, 182
get_touch_x, 183
get_touch_y, 184
* **transform functions**
transform, 326
* **volume functions**
set_audio_stream_volume, 292
set_master_volume, 298
set_music_volume, 302
set_sound_volume, 307
* **wave functions**
export_wave, 129
export_wave_as_code, 129
load_sound_from_wave, 270
load_wave, 273
unload_wave, 335
* **window functions**
clear_window_state, 29
close_window, 30
get_window_position, 184
get_window_scale_dpi, 185
init_window, 235
is_window_focused, 252
is_window_fullscreen, 253
is_window_hidden, 254
is_window_maximized, 254
is_window_minimized, 255
is_window_ready, 256
is_window_resized, 256
is_window_state, 257
maximize_window, 275
minimize_window, 277
restore_window, 287
set_window_icon, 310
set_window_min_size, 311
set_window_monitor, 311
set_window_position, 312
set_window_size, 313
set_window_state, 313
set_window_title, 314
window_should_close, 339
* **world functions**
get_screen_to_world_2d, 177
get_world_to_screen, 186
get_world_to_screen_2d, 187
get_world_to_screen_ex, 188
0..1, *33–35*
0..360, *33*, *35*

as_color, 10, *31–36*, *134*, *142*, *158*, *195–200*
audio_stream(), *237*, *238*, *278*, *281*, *288*, *292*, *293*, *316*, *327*

begin_blend_mode, 10, *12–14*, *123–127*, *294*
begin_drawing, 11
begin_mode_2d, 11, *11–14*, *123–127*, *294*
begin_mode_3d, *11*, 12, *12–14*, *123–127*, *294*
begin_scissor_mode, *11*, *12*, 13, *14*, *123–127*, *294*
begin_shader_mode, *11*, *12*, 13, *13*, *14*, *123–127*, *179*, *180*, *268*, *269*, *294*, *303–305*, *333*

begin_texture_mode, *11–13*, 14, *14*, *50, 51*, *108–111*, *113–116*, *118*, *123–127*, *139*, *264*, *268*, *272, 273*, *294*, *305, 306*, *308, 309*, *325*, *332*, *334*
bounding_box, 15

camera_2d, 15
camera_2d(), *12*, *140*, *178*, *187*
camera_3d, 16
camera_3d(), *12*, *38*, *139*, *187, 188*, *294*, *336*
change_directory, 17, *28*, *37*, *143*, *170*, *186*
check_collision_box_sphere, 18, *19–27*, *141*, *172–176*
check_collision_boxes, *18*, 19, *20–27*, *141*, *172–176*
check_collision_circle_rec, *18*, 19, *19*, *21–27*, *41–45*, *47, 48*, *141*, *172–176*, *206*, *208*
check_collision_circles, *18, 19*, 20, *20*, *22–27*, *141*, *172–176*
check_collision_lines, *18–20*, 21, *21*, *23–27*, *141*, *172–176*
check_collision_point_circle, *18–21*, 22, *22*, *24–27*, *41–45*, *47, 48*, *141*, *172–176*, *206*, *208*
check_collision_point_line, *18–22*, 23, *23–27*, *64–66*, *68–71*, *141*, *172–176*, *209*, *211*
check_collision_point_rec, *18–23*, 24, *24–27*, *141*, *172–176*
check_collision_point_triangle, *18–24*, 25, *26, 27*, *141*, *172–176*
check_collision_recs, *18–25*, 26, *27*, *141*, *172–176*
check_collision_spheres, *18–25*, 26, *26*, *141*, *172–176*
clear_background, 27
clear_directory_files, *17*, 28, *37*, *143*, *170*, *186*
clear_dropped_files, 28
clear_window_state, 29, *30*, *185*, *236*, *253–257*, *275*, *277*, *287*, *310–314*, *340*
close_audio_device, 29, *235–238*, *258*, *278*, *281*, *288*, *291–293*, *316*, *327*
close_window, *29*, 30, *185*, *236*, *253–257*, *275*, *277*, *287*, *310–314*, *340*
color, *10*, 30, *32–36*, *134*, *142*, *158*, *195–200*
color(), *27*, *32*, *35, 36*, *38–41*, *43–47*, *49–57*, *59–61*, *64–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–119*, *121, 122*, *130*, *133–137*, *191*, *194*, *199, 200*, *204*, *206–208*, *210, 211*, *213, 214*, *216, 217*, *219, 220*, *222*, *229*, *232–234*, *261*, *329, 330*
color_alpha, *10*, 31, *31–36*, *134*, *142*, *158*, *195–200*
color_alpha_blend, *10*, *31*, 32, *32–36*, *134*, *142*, *158*, *195–200*
color_from_hsv, *10*, *31, 32*, 33, *34–36*, *134*, *142*, *158*, *195–200*
color_from_normalized, *10*, *31, 32*, 33, *33*, *35, 36*, *134*, *142*, *158*, *195–200*
color_normalize, *10*, *31–33*, 34, *34–36*, *134*, *142*, *158*, *195–200*
color_to_hsv, *10*, *31–34*, 35, *35, 36*, *134*, *142*, *158*, *195–200*
color_to_int, *10*, *31–35*, 36, *134*, *142*, *158*, *195–200*

directory_exists, *17*, *28*, 36, *143*, *170*, *186*
disable_cursor, 37, *122*, *190*, *239, 240*, *299*, *315*
draw_billboard, 38, *39–41*, *43–47*, *49–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–119*, *121, 122*, *205–207*, *209, 210*, *212–214*, *216, 217*, *219, 220*, *222*
draw_bounding_box, *38*, 39, *40, 41*, *43–47*, *49–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–119*, *121, 122*, *205–207*, *209, 210*, *212–214*, *216, 217*, *219, 220*, *222*
draw_circle, *20*, *23*, *38, 39*, 40, *41–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–119*, *121, 122*, *205–210*, *212–214*, *216*, *218–220*, *222*
draw_circle_3d, *20*, *23*, *38–40*, 41, *41*, *43–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–119*, *121, 122*, *205–210*, *212–214*, *216, 217*, *219, 220*, *222*
draw_circle_gradient, *20*, *23*, *38–41*, 42,

42, *44–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–119*, *121, 122*, *205–210*, *212–214*, *216, 217*, *219, 220*, *222*

draw_circle_lines, *20*, *23*, *38–42*, 43, *43*, *45–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–119*, *121, 122*, *205–210*, *212–214*, *216, 217*, *219, 220*, *222*

draw_circle_sector, *20*, *23*, *38–43*, 44, *44*, *46–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–119*, *121, 122*, *205–210*, *212–214*, *216, 217*, *219, 220*, *222*

draw_circle_sector_lines, *20*, *23*, *38–45*, 46, *47–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–119*, *121, 122*, *205–210*, *212–214*, *216, 217*, *219, 220*, *222*

draw_circle_v, *20*, *23*, *38–46*, 47, *47*, *49–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–119*, *121, 122*, *205–210*, *212–214*, *216, 217*, *219, 220*, *222*

draw_cube, *38–41*, *43–46*, 48, *48*, *50–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–118*, *120–122*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*

draw_cube_texture, *14*, *38–41*, *43–47*, 49, *49*, *51–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–119*, *121, 122*, *126*, *139*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*, *264*, *268*, *272, 273*, *305, 306*, *308, 309*, *325*, *332*, *334*

draw_cube_texture_rec, *14*, *38–41*, *43–47*, *49*, 50, *50*, *52–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–119*, *121, 122*, *126*, *139*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*, *264*, *268*, *272, 273*, *305, 306*, *308, 309*, *325*, *332*, *334*

draw_cube_v, *38–41*, *43–47*, *49–51*, 52, *53–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–119*, *121, 122*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*

draw_cube_wires, *38–41*, *43–47*, *49–52*, 53, *54–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–118*, *120–122*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*

draw_cube_wires_v, *38–41*, *43–47*, *49–53*, 54, *55–57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–119*, *121, 122*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*

draw_cylinder, *38–41*, *43–46*, *48–54*, 55, *56, 57*, *59–69*, *71–82*, *84–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–118*, *120–122*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*

draw_cylinder_ex, *38–41*, *43–46*, *48–55*, 56, *57*, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–118*, *120–122*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*

draw_cylinder_wires, *38–41*, *43–46*, *48–56*, 57, *59–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–118*, *120–122*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*

draw_cylinder_wires_ex, *38–41*, *43–46*, *48–57*, 58, *60–69*, *71–83*, *85–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–118*, *120–122*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*

draw_ellipse, *38–41*, *43–46*, *48–57*, 59, *59*, *61–69*, *71–82*, *84–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–118*, *120–122*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*

draw_ellipse_lines, *38–41*, *43–46*, *48–57*, *59*, 60, *60*, *62–69*, *71–82*, *84–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–118*, *120–122*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*

draw_fps, *38–41*, *43–46*, *48–57*, *59, 60*, 61, *61*, *63–69*, *71–82*, *84–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–118*, *120–122*, *147*, *205–207*, *209, 210*, *212–214*, *216*, *218–220*, *222*, *308*

draw_grid, *38–41*, *43–46*, *48–57*, *59–61*, 62, *62*, *64–69*, *71–82*, *84–89*, *91–95*,

97–102, *104–108*, *110–112*, *114–118*, *120–122*, *205–207*, *209*, *210*, *212–214*, *216*, *218–220*, *222*
draw_line, *24*, *38–40*, *42–46*, *48–56*, *58–62*, 63, *63*, *65–80*, *82*, *84–89*, *91–95*, *97–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209–214*, *216*, *218*, *219*, *221*, *222*
draw_line_3d, *24*, *38–41*, *43–46*, *48–57*, *59–63*, 64, *64*, *66–82*, *84–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–118*, *120–122*, *205–207*, *209–214*, *216*, *218–220*, *222*
draw_line_bezier, *24*, *38–41*, *43–46*, *48–56*, *58–64*, 65, *65*, *67–82*, *84–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–117*, *119–122*, *205–207*, *209–214*, *216*, *218*, *219*, *221*, *222*
draw_line_bezier_cubic, *24*, *38–41*, *43–46*, *48–57*, *59–65*, 66, *66*, *68–82*, *84–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–117*, *119–122*, *205–207*, *209–214*, *216*, *218*, *219*, *221*, *222*
draw_line_bezier_quad, *24*, *38–41*, *43–46*, *48–56*, *58–67*, 68, *68*, *70–82*, *84–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–117*, *119–122*, *205–207*, *209–214*, *216*, *218*, *219*, *221*, *222*
draw_line_ex, *24*, *38–41*, *43–46*, *48–56*, *58–68*, 69, *69*, *71–82*, *84–89*, *91–95*, *97–102*, *104–108*, *110–112*, *114–117*, *119–122*, *205–207*, *209–214*, *216*, *218*, *219*, *221*, *222*
draw_line_v, *24*, *38–40*, *42–46*, *48–56*, *58–69*, 70, *70*, *72–80*, *82*, *84–89*, *91–95*, *97–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209–214*, *216*, *218*, *219*, *221*, *222*
draw_model, *38–40*, *42–46*, *48–56*, *58–68*, *70*, 71, *71*, *73–80*, *82*, *84–89*, *91–95*, *97–102*, *104–107*, *109–112*, *114–117*, *119–122*, *159*, *173*, *205–207*, *209*, *210*, *212–214*, *216*, *218*, *219*, *221*, *222*, *266*, *331*
draw_model_ex, *38–40*, *42–46*, *48–56*, *58–68*, *70*, *71*, 72, *72*, *74–80*, *82*, *84–89*, *91–95*, *97–102*, *104–107*, *109–112*, *114–117*, *119–122*, *159*, *173*, *205–207*, *209*, *210*, *212–214*, *216*, *218*, *219*, *221*, *222*, *266*, *331*
draw_model_wires, *38–40*, *42–46*, *48–56*, *58–68*, *70–73*, 74, *75–80*, *82*, *84–89*, *91–95*, *97–102*, *104–107*, *109–112*, *114–117*, *119–122*, *159*, *173*, *205–207*, *209*, *210*, *212–214*, *216*, *218*, *219*, *221*, *222*, *266*, *331*
draw_pixel, *38–40*, *42–46*, *48–56*, *58–68*, *70–74*, 75, *76–80*, *82–89*, *91–95*, *97–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212–214*, *216*, *218*, *219*, *221*, *222*
draw_pixel_v, *38–40*, *42–46*, *48–56*, *58–68*, *70–75*, 76, *77–80*, *82*, *84–89*, *91–95*, *97–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212–214*, *216*, *218*, *219*, *221*, *222*
draw_plane, *38–40*, *42–46*, *48–56*, *58–68*, *70–76*, 77, *78–80*, *82–89*, *91–95*, *97–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212–214*, *216*, *218*, *219*, *221*, *222*
draw_point_3d, *38–40*, *42–45*, *47–56*, *58–68*, *70–77*, 78, *79*, *80*, *82–89*, *91–95*, *97–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212–214*, *216*, *218*, *219*, *221*, *222*
draw_poly, *38–40*, *42–45*, *47–56*, *58–68*, *70–78*, 79, *80*, *82–89*, *91–94*, *96–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212–214*, *216*, *218*, *219*, *221*, *222*
draw_poly_lines, *38–40*, *42–45*, *47–56*, *58–68*, *70–79*, 80, *82–89*, *91–95*, *97–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212–214*, *216*, *218*, *219*, *221*, *222*
draw_poly_lines_ex, *38–40*, *42–45*, *47–56*, *58–68*, *70–80*, 81, *83–89*, *91–95*, *97–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*,

210, *212–214*, *216*, *218*, *219*, *221*, *222*
draw_ray, *38–40*, *42–45*, *47–56*, *58–68*, *70–80*, 82, *82*, *84–89*, *91–94*, *96–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212–214*, *216*, *218*, *219*, *221*, *222*
draw_rectangle, *38–40*, *42–45*, *47–55*, *57–67*, *69–79*, *81*, *82*, 83, *83*, *85–101*, *103–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *286*
draw_rectangle_gradient_ex, *38–40*, *42–45*, *47–56*, *58–68*, *70–80*, *82*, *83*, 84, *84*, *86–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212–216*, *218*, *219*, *221*, *222*, *286*
draw_rectangle_gradient_h, *38–40*, *42–45*, *47–56*, *58–68*, *70–80*, *82–84*, 85, *85*, *87–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212–216*, *218*, *219*, *221*, *222*, *286*
draw_rectangle_gradient_v, *38–40*, *42–45*, *47–56*, *58–68*, *70–80*, *82–85*, 86, *86*, *88–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *286*
draw_rectangle_lines, *38–40*, *42–45*, *47–55*, *57–68*, *70–80*, *82–87*, 88, *90–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *286*
draw_rectangle_lines_ex, *38–40*, *42–45*, *47–55*, *57–68*, *70–80*, *82–88*, 89, *89*, *91–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *286*
draw_rectangle_pro, *38–40*, *42–45*, *47–55*, *57–68*, *70–80*, *82–89*, 90, *90*, *92–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *286*
draw_rectangle_rec, *38–40*, *42–45*, *47–55*, *57–68*, *70–80*, *82–90*, 91, *91*, *93–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *286*
draw_rectangle_rounded, *38–40*, *42–45*, *47–55*, *57–67*, *69–79*, *81–91*, 92, *92*, *94–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *286*
draw_rectangle_rounded_lines, *38–40*, *42–45*, *47–55*, *57–67*, *69–79*, *81–93*, 94, *96–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *286*
draw_rectangle_v, *38–40*, *42–45*, *47–55*, *57–67*, *69–79*, *81–94*, 95, *95*, *97–102*, *104–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *286*
draw_ring, *38–40*, *42–45*, *47–55*, *57–67*, *69–79*, *81–88*, *90–94*, 96, *96*, *98–101*, *103–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*
draw_ring_lines, *38–40*, *42–45*, *47–55*, *57–67*, *69–79*, *81–88*, *90–94*, *96*, 97, *97*, *99–101*, *103–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*
draw_sphere, *38–40*, *42–45*, *47–55*, *57–67*, *69–79*, *81–88*, *90–94*, *96–98*, 99, *100*, *101*, *103–107*, *109–112*, *114–117*, *119–122*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*
draw_sphere_ex, *38–40*, *42–45*, *47–55*, *57–67*, *69–79*, *81–88*, *90–94*, *96–99*, 100, *101*, *103–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*,

210, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*
draw_sphere_wires, *38–40*, *42–45*, *47–55*, *57–67*, *69–79*, *81–88*, *90–94*, *96–100*, 101, *103–107*, *109–112*, *114–117*, *119–122*, *205–207*, *209*, *210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*
draw_text, *39–45*, *47–54*, *56–67*, *69–74*, *76–78*, *80–88*, *90–93*, *95–100*, 102, *102*, *104–107*, *109–111*, *113–116*, *118–122*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *232*, *234*, *259*, *276*, *320–324*
draw_text_codepoint, *38*, *39*, *41–45*, *47–55*, *57–67*, *69–79*, *81–88*, *90–94*, *96–101*, 103, *103*, *105–107*, *109–112*, *114–117*, *119–122*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *232*, *234*, *259*, *276*, *320–324*
draw_text_ex, *38*, *39*, *41–45*, *47–55*, *57–67*, *69–79*, *81–88*, *90–94*, *96–101*, *103*, 104, *104*, *106*, *107*, *109–112*, *114–117*, *119–122*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *232*, *234*, *259*, *276*, *320–324*
draw_text_pro, *38*, *39*, *41–45*, *47–55*, *57–67*, *69–79*, *81–88*, *90–94*, *96–101*, *103*, *104*, 105, *105*, *107*, *109–112*, *114–117*, *119–122*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *232*, *234*, *259*, *276*, *320–324*
draw_texture, *14*, *39–45*, *47–54*, *56–67*, *69–74*, *76–78*, *80–88*, *90–93*, *95–100*, *102–106*, 107, *109–111*, *113–122*, *126*, *139*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *264*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, *325*, *332*, *334*
draw_texture_ex, *14*, *38*, *39*, *41–45*, *47–55*, *57–67*, *69–79*, *81–88*, *90–94*, *96–101*, *103–107*, 108, *108*, *110–122*, *126*, *139*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *264*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, *325*, *332*, *334*
draw_texture_npatch, *14*, *38*, *39*, *41–45*, *47–55*, *57–67*, *69–79*, *81–88*, *90–94*, *96–101*, *103–108*, 109, *109*, *111*, *113–116*, *118–122*, *126*, *139*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *264*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, *325*, *332*, *334*
draw_texture_poly, *14*, *38*, *39*, *41–45*, *47–54*, *56–67*, *69–78*, *80–88*, *90–94*, *96–101*, *103–109*, 110, *110*, *113–116*, *118–122*, *126*, *139*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *264*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, *325*, *332*, *334*
draw_texture_pro, *14*, *38*, *39*, *41–45*, *47–54*, *56–67*, *69–78*, *80–88*, *90–94*, *96–101*, *103–111*, 112, *114–116*, *118–122*, *126*, *139*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *264*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, *325*, *332*, *334*
draw_texture_quad, *14*, *38*, *39*, *41–45*, *47–54*, *56–67*, *69–78*, *80–88*, *90–94*, *96–101*, *103–111*, 113, *113*, *115–122*, *126*, *139*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *264*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, *325*, *332*, *334*
draw_texture_rec, *14*, *39*, *41–45*, *47–54*, *56–67*, *69–78*, *80–88*, *90–94*, *96–101*, *103–111*, *113*, 114, *114*, *116–122*, *126*, *139*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *264*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, *325*, *332*, *334*
draw_texture_tiled, *14*, *39*, *41–45*, *47–54*, *56–67*, *69–78*, *80–88*, *90–94*, *96–101*, *103–111*, *113*, *114*, 115, *115*, *118–122*, *126*, *139*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *264*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, *325*, *332*, *334*
draw_texture_v, *14*, *39–45*, *47–54*, *56–67*, *69–74*, *76–78*, *80–88*, *90–93*, *95–100*, *102–111*, *113–116*, 117,

117, *119–122*, *126*, *139*, *205*, *206*, *208–210*, *212*, *213*, *215*, *216*, *218*, *219*, *221*, *222*, *264*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, *325*, *332*, *334*
draw_triangle, *39–45*, *47–54*, *56–67*, *69–74*, *76–78*, *80–88*, *90–93*, *95–100*, *102–107*, *109–111*, *113–116*, 118, *118*, *120–122*, *205*, *206*, *208*, *209*, *211–213*, *215*, *216*, *218*, *219*, *221*, *222*
draw_triangle_3d, *39–45*, *47–54*, *56–67*, *69–74*, *76–78*, *80–88*, *90–93*, *95–100*, *102–107*, *109–111*, *113–116*, *118*, 119, *119*, *121*, *122*, *205*, *206*, *208*, *209*, *211–213*, *215*, *216*, *218*, *219*, *221*, *222*
draw_triangle_lines, *39–45*, *47–54*, *56–67*, *69–74*, *76–78*, *80–88*, *90–93*, *95–100*, *102–107*, *109–111*, *113–116*, *118*, *119*, 120, *120*, *122*, *205*, *206*, *208*, *209*, *211–213*, *215*, *216*, *218*, *219*, *221*, *222*
draw_triangle_strip_3d, *39–45*, *47–54*, *56–67*, *69–74*, *76–78*, *80–88*, *90–93*, *95–100*, *102–107*, *109–111*, *113–116*, *118–120*, 121, *121*, *205*, *206*, *208*, *209*, *211–213*, *215*, *216*, *218*, *219*, *221*, *222*

enable_cursor, *37*, 122, *190*, *239*, *240*, *299*, *315*
end_blend_mode, *11–14*, 123, *124–127*, *294*
end_drawing, 123
end_mode_2d, *11–14*, *123*, 124, *124–127*, *294*
end_mode_3d, *11–14*, *123*, 124, *124–127*, *294*
end_scissor_mode, *11–14*, *123*, *124*, 125, *125–127*, *294*
end_shader_mode, *11–14*, *123*, *124*, 125, *125–127*, *179*, *180*, *268*, *269*, *294*, *303–305*, *333*
end_texture_mode, *11–14*, *50*, *51*, *108–111*, *113–115*, *117*, *118*, *123–125*, 126, *127*, *139*, *264*, *268*, *272*, *273*, *294*, *305*, *306*, *308*, *309*, *325*, *332*, *334*
end_vr_stereo_mode, *11–14*, *123–125*, 126, *126*, *294*
export_image, 127, *128*, *132–138*, *157*, *158*, *191–204*, *206–208*, *210*, *211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
export_image_as_code, *127*, 128, *132–138*, *157*, *158*, *191–204*, *206–208*, *210*, *211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
export_wave, 129, *130*, *271*, *274*, *335*
export_wave_as_code, 129, *129*, *271*, *274*, *335*

fade, 130
file_exists, 131, *144–146*, *240*, *241*, *259*
font(), *104–106*, *154–156*, *222*, *233*, *276*, *327*

gen_image_cellular, *127*, *128*, 131, *133–138*, *157*, *158*, *191–204*, *206–208*, *210*, *211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
gen_image_checked, *127*, *128*, 132, *132*, *134–138*, *157*, *158*, *191–204*, *206–208*, *210*, *211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
gen_image_color, *10*, *31–36*, *127*, *128*, *132*, 133, *133*, *135–138*, *142*, *157*, *158*, *191–204*, *206–208*, *210*, *211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
gen_image_gradient_h, *127*, *128*, *132*, *133*, 134, *134*, *136–138*, *157*, *158*, *191–204*, *206–208*, *210*, *211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
gen_image_gradient_radial, *127*, *128*, *132–134*, 135, *135*, *137*, *138*, *157*, *158*, *191–204*, *206–208*, *210*, *211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
gen_image_gradient_v, *127*, *128*, *132–135*, 136, *136*, *138*, *157*, *158*, *191–204*, *206–208*, *210*, *211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
gen_image_white_noise, *127*, *128*, *132–136*, 137, *137*, *157*, *158*, *191–204*, *206–208*, *210*, *211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*

gen_texture_mipmaps, *14*, *50, 51*, *108–111*, *113–115*, *117, 118*, *126*, 138, *264*, *268*, *272, 273*, *305, 306*, *308, 309*, *325*, *332*, *334*
get_camera_matrix, 139, *140*, *293–296*, *336*
get_camera_matrix_2d, *139*, 140, *293–296*, *336*
get_char_pressed, 140, *158*, *246–248*, *297*
get_collision_rec, *18–27*, 141, *172–176*
get_color, *10*, *31–36*, *134*, 142, *158*, *195–200*
get_current_monitor, 142, *160–164*, *312*
get_directory_path, *17*, *28*, *37*, 143, *170*, *186*
get_file_extension, *131*, 144, *145, 146*, *240, 241*, *259*
get_file_mod_time, *131*, 144, *144–146*, *240, 241*, *259*
get_file_name, *131*, *144*, 145, *145, 146*, *240, 241*, *259*
get_file_name_without_ext, *131*, *144, 145*, 146, *240, 241*, *259*
get_font_default, 146, *260, 261*, *327, 328*
get_fps, *62*, 147, *308*
get_frame_time, 148
get_gamepad_axis_count, 148, *149–151*, *241–244*, *298*
get_gamepad_axis_movement, 149, *149–151*, *241–244*, *298*
get_gamepad_button_pressed, *149*, 150, *151*, *241–244*, *298*
get_gamepad_name, *149*, 150, *150*, *241–244*, *298*
get_gesture_detected, 151
get_gesture_drag_angle, 151
get_gesture_drag_vector, 152
get_gesture_hold_duration, 152
get_gesture_pinch_angle, 153
get_gesture_pinch_vector, 153
get_glyph_atlas_rec, 154
get_glyph_index, 155
get_glyph_info, 155
get_image_alpha_border, *127, 128*, *132–138*, 156, *158*, *191–204*, *206–208*, *210, 211*, *213, 214*, *216, 217*, *219, 220*, *222–234*, *261–265*, *273*, *329, 330*
get_image_color, *10*, *31–36*, *127, 128*, *132–138*, *142*, 157, *157*, *191–204*, *206–208*, *210, 211*, *213, 214*, *216, 217*, *219, 220*, *222–234*, *261–265*, *273*, *329, 330*
get_key_pressed, *141*, 158, *246–248*, *297*
get_model_bounding_box, *72, 73*, *75*, 159, *173*, *266*, *331*
get_monitor_count, *143*, 159, *160–164*, *312*
get_monitor_height, *143*, 160, *160–164*, *312*
get_monitor_name, *143*, *160*, 161, *162–164*, *312*
get_monitor_physical_height, *143*, *160*, 161, *161–164*, *312*
get_monitor_physical_width, *143*, *160, 161*, 162, *162–164*, *312*
get_monitor_position, *143*, *160–162*, 163, *164*, *312*
get_monitor_refresh_rate, *143*, *160–162*, 163, *163, 164*, *312*
get_monitor_width, *143*, *160–163*, 164, *164*, *312*
get_mouse_delta, 165, *166–168*, *248–250*, *299–301*
get_mouse_position, 165, *165–168*, *248–250*, *299–301*
get_mouse_wheel_move, *165*, 166, *166–168*, *248–250*, *299–301*
get_mouse_x, *165, 166*, 167, *168*, *248–250*, *299–301*
get_mouse_y, *165, 166*, 167, *167*, *248–250*, *299–301*
get_music_time_length, 168, *169*, *251*, *267*, *279*, *282*, *288*, *290*, *302, 303*, *317*, *332*, *336*
get_music_time_played, *168*, 169, *251*, *267*, *279*, *282*, *288*, *290*, *302, 303*, *317*, *332*, *336*
get_pixel_data_size, 169
get_prev_directory_path, *17*, *28*, *37*, *143*, 170, *186*
get_random_value, 171
get_ray_collision_box, *18–27*, *141*, 171, *172–176*
get_ray_collision_mesh, *18–27*, *141*, 172, *172–176*
get_ray_collision_model, *18–27*, *72, 73*, *75*, *141*, *159*, *172*, 173, *174–176*, *266*, *331*

get_ray_collision_quad, *18–27*, *141*, *172*, *173*, 174, *175*, *176*
get_ray_collision_sphere, *18–27*, *141*, *172–174*, 175, *176*
get_ray_collision_triangle, *18–27*, *141*, *172–175*, 176
get_screen_height, 177, *178*, *187*, *188*, *240*, *263*
get_screen_to_world_2d, 177, *177*, *178*, *187*, *188*, *240*, *263*
get_screen_width, *177*, 178, *178*, *187*, *188*, *240*, *263*
get_shader_location, *14*, *125*, 179, *180*, *268*, *269*, *303–305*, *333*
get_shader_location_attrib, *14*, *125*, 179, *179*, *268*, *269*, *303–305*, *333*
get_sounds_playing, 180
get_time, 181
get_touch_point_count, 181, *182–184*
get_touch_point_id, *181*, 182, *183*, *184*
get_touch_position, *181*, 182, *182–184*
get_touch_x, *181*, *182*, 183, *183*, *184*
get_touch_y, *181–183*, 184
get_window_position, *29*, *30*, 184, *185*, *236*, *253–257*, *275*, *277*, *287*, *310–314*, *340*
get_window_scale_dpi, *29*, *30*, 185, *185*, *236*, *253–257*, *275*, *277*, *287*, *310–314*, *340*
get_working_directory, *17*, *28*, *37*, *143*, *170*, 186
get_world_to_screen, *177*, *178*, 186, *187*, *188*, *240*, *263*
get_world_to_screen_2d, *177*, *178*, 187, *187*, *188*, *240*, *263*
get_world_to_screen_ex, *177*, *178*, *187*, 188, *240*, *263*
glyph_info, 189

hide_cursor, *37*, *122*, 190, *239*, *240*, *299*, *315*

image(), *127*, *128*, *157*, *158*, *191–204*, *206–208*, *210*, *211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–231*, *234*, *261*, *273*, *310*, *329*
image_alpha_clear, *127*, *128*, *132–138*, *157*, *158*, 190, *192–204*, *206*, *207*, *209–211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
image_alpha_crop, *127*, *128*, *132–138*, *157*, *158*, 191, *191–204*, *206*, *207*, *209–211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
image_alpha_mask, *127*, *128*, *132–138*, *157*, *158*, *191*, 192, *192–204*, *206*, *207*, *209–211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
image_alpha_premultiply, *127*, *128*, *132–138*, *157*, *158*, *191*, *192*, 193, *194–204*, *206*, *207*, *209–211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
image_clear_background, *127*, *128*, *132–138*, *157*, *158*, *191–193*, 194, *195–204*, *206*, *207*, *209–211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
image_color_brightness, *10*, *31–36*, *127*, *128*, *132–138*, *142*, *157*, *158*, *191–194*, 195, *196–204*, *206*, *207*, *209–211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
image_color_contrast, *10*, *31–36*, *127*, *128*, *132–138*, *142*, *157*, *158*, *191–195*, 196, *197–204*, *206*, *207*, *209–211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
image_color_grayscale, *10*, *31–36*, *127*, *128*, *132–138*, *142*, *157*, *158*, *191–196*, 197, *198–204*, *206*, *207*, *209–211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
image_color_invert, *10*, *31–36*, *127*, *128*, *132–138*, *142*, *157*, *158*, *191–197*, 198, *199–204*, *206*, *207*, *209–211*, *213*, *214*, *216*, *217*, *219*, *220*, *222–234*, *261–265*, *273*, *329*, *330*
image_color_replace, *10*, *31–36*, *127*, *128*, *132–138*, *142*, *157*, *158*, *191–198*, 199, *200–204*, *206*, *207*, *209–211*, *213*, *214*, *216*, *217*, *219*, *220*,

222–234, 261–265, 273, 329, 330
image_color_tint, 10, 31–36, 127, 128, 132–138, 142, 157, 158, 191–199, 200, 201–204, 206, 207, 209–211, 213, 214, 216, 217, 219, 220, 222–234, 261–265, 273, 329, 330
image_copy, 127, 128, 132–138, 157, 158, 191–200, 201, 202–204, 206, 207, 209–211, 213, 214, 216, 217, 219, 220, 222–234, 261–265, 273, 329, 330
image_crop, 127, 128, 132–138, 157, 158, 191–201, 202, 203, 204, 206, 207, 209–211, 213, 214, 216, 217, 219, 220, 222–234, 261–265, 273, 329, 330
image_dither, 127, 128, 132–138, 157, 158, 191–202, 203, 204, 206, 207, 209–211, 213, 214, 216, 217, 219, 220, 222–234, 261–265, 273, 329, 330
image_draw, 39–45, 47–52, 54–66, 68–73, 75–87, 89–93, 95–99, 101–106, 108–111, 113–116, 118–122, 127, 128, 132–138, 157, 158, 191–203, 204, 206–234, 261–265, 273, 329, 330
image_draw_circle, 20, 23, 39–45, 47–54, 56–67, 69–74, 76–78, 80–88, 90–93, 95–100, 102–107, 109–111, 113–116, 118–122, 127, 128, 132–138, 157, 158, 191–204, 205, 205, 207–234, 261–265, 273, 329, 330
image_draw_circle_v, 20, 23, 39–45, 47–54, 56–67, 69–74, 76–78, 80–88, 90–93, 95–100, 102–107, 109–111, 113–116, 118–122, 127, 128, 132–138, 157, 158, 191–205, 206, 206, 209–234, 261–265, 273, 329, 330
image_draw_line, 24, 39–45, 47–54, 56–66, 68–74, 76–88, 90–93, 95–100, 102–107, 109–111, 113–116, 118–122, 127, 128, 132–138, 157, 158, 191–207, 208, 208, 210–234, 261–265, 273, 329, 330
image_draw_line_v, 24, 39–45, 47–54, 56–66, 68–74, 76–88, 90–93, 95–100, 102–107, 109–111, 113–116, 118–122, 127, 128, 132–138, 157, 158, 191–208, 209, 209, 211–234, 261–265, 273, 329, 330
image_draw_pixel, 39–45, 47–54, 56–66, 68–74, 76–88, 90–93, 95–100, 102–106, 108–111, 113–116, 118–122, 127, 128, 132–138, 157, 158, 191–210, 211, 211, 213–234, 261–265, 273, 329, 330
image_draw_pixel_v, 39–45, 47–54, 56–66, 68–74, 76–88, 90–93, 95–100, 102–107, 109–111, 113–116, 118–122, 127, 128, 132–138, 157, 158, 191–211, 212, 212, 214–234, 261–265, 273, 329, 330
image_draw_rectangle, 39–45, 47–52, 54–66, 68–73, 75–87, 89–93, 95–99, 101–106, 108–111, 113–116, 118–122, 127, 128, 132–138, 157, 158, 191–212, 213, 213, 216–234, 261–265, 273, 286, 329, 330
image_draw_rectangle_lines, 39–45, 47–54, 56–66, 68–74, 76–93, 95–100, 102–106, 108–111, 113–116, 118–122, 127, 128, 132–138, 157, 158, 191–214, 215, 215, 217–234, 261–265, 273, 286, 329, 330
image_draw_rectangle_rec, 39–45, 47–52, 54–66, 68–73, 75–87, 89–93, 95–99, 101–106, 108–111, 113–116, 118–122, 127, 128, 132–138, 157, 158, 191–216, 217, 219–234, 261–265, 273, 286, 329, 330
image_draw_rectangle_v, 39–45, 47–52, 54–66, 68–73, 75–87, 89–93, 95–99, 101–106, 108–111, 113–116, 118–122, 127, 128, 132–138, 157, 158, 191–217, 218, 218, 220–234, 261–265, 273, 286, 329, 330
image_draw_text, 39–45, 47–52, 54–66, 68–73, 75–87, 89–93, 95–99, 101–106, 108–111, 113–116, 118–122, 127, 128, 132–138, 157, 158, 191–218, 219, 219, 222–234,

259, *261–265*, *273*, *276*, *320–324*, *329*, *330*
image_draw_text_ex, *39–45*, *47–52*, *54–66*, *68–73*, *75–87*, *89–93*, *95–99*, *101–106*, *108–111*, *113–116*, *118–122*, *127*, *128*, *132–138*, *157*, *158*, *191–220*, 221, *221*, *223–234*, *259*, *261–265*, *273*, *276*, *320–324*, *329*, *330*
image_flip_horizontal, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209–211*, *213*, *214*, *216*, *217*, *219*, *220*, 222, *222*, *224–234*, *261–265*, *273*, *329*, *330*
image_flip_vertical, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222*, 223, *223*, *225–234*, *261–265*, *273*, *329*, *330*
image_format, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222*, *223*, 224, *224*, *226–234*, *261–265*, *273*, *329*, *330*
image_from_image, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–224*, 225, *225*, *227–234*, *261–265*, *273*, *329*, *330*
image_mipmaps, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–225*, 226, *226*, *228–234*, *261–265*, *273*, *329*, *330*
image_resize, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–226*, 227, *227*, *229–234*, *261–265*, *273*, *329*, *330*
image_resize_canvas, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–227*, 228, *228*, *230–234*, *261–265*, *273*, *329*, *330*
image_resize_nn, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–228*, 229, *229*, *231–234*, *261–265*, *273*, *329*, *330*
image_rotate_ccw, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–229*, 230, *230–234*, *261–265*, *273*, *329*, *330*
image_rotate_cw, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–230*, 231, *231–234*, *261–265*, *273*, *329*, *330*
image_text, *103–106*, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219–231*, 232, *233*, *234*, *259*, *261–265*, *273*, *276*, *320–324*, *329*, *330*
image_text_ex, *103–106*, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219–232*, 233, *234*, *259*, *261–265*, *273*, *276*, *320–324*, *329*, *330*
image_to_pot, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–233*, 234, *261–265*, *273*, *329*, *330*
init_audio_device, *30*, 235, *236–238*, *258*, *278*, *281*, *288*, *291–293*, *316*, *327*
init_window, *29*, *30*, *185*, 235, *235*, *253–257*, *275*, *277*, *287*, *310–314*, *340*
is_audio_device_ready, *30*, *235*, 236, *237*, *238*, *258*, *278*, *281*, *288*, *291–293*, *316*, *327*
is_audio_stream_playing, *30*, *235*, *236*, 237, *238*, *251*, *258*, *267*, *278*, *279*, *281*, *282*, *288*, *290–293*, *316*, *317*, *327*, *332*, *336*
is_audio_stream_processed, *30*, *235–237*, 238, *251*, *258*, *267*, *278*, *279*, *281*, *282*, *288*, *290–293*, *316*, *317*, *327*, *332*, *336*
is_cursor_hidden, *37*, *122*, *190*, 239, *240*, *299*, *315*
is_cursor_on_screen, *37*, *122*, *177*, *178*, *187*, *188*, *190*, 239, *239*, *263*, *299*, *315*
is_file_dropped, *131*, *144–146*, 240, *241*, *259*

is_file_extension, *131*, *144–146*, 240, *240*, *259*
is_gamepad_available, *149–151*, 241, *242–244*, *298*
is_gamepad_button_down, *149–151*, *241*, 242, *243*, *244*, *298*
is_gamepad_button_pressed, *149–151*, *241*, 242, *242*, *244*, *298*
is_gamepad_button_released, *149–151*, *241*, *242*, 243, *243*, *244*, *298*
is_gamepad_button_up, *149–151*, *241–243*, 244, *244*, *298*
is_gesture_detected, 245
is_key_down, *141*, *158*, 245, *246–248*, *297*
is_key_pressed, *141*, *158*, 246, *246–248*, *297*
is_key_released, *141*, *158*, *246*, 247, *248*, *297*
is_key_up, *141*, *158*, *246*, 247, *247*, *297*
is_mouse_button_down, *165–168*, 248, *249*, *250*, *299–301*
is_mouse_button_pressed, *165–168*, *248*, 249, *250*, *299–301*
is_mouse_button_released, *165–168*, *248*, 249, *249*, *250*, *299–301*
is_mouse_button_up, *165–168*, *248*, *249*, 250, *250*, *299–301*
is_music_stream_playing, *168*, *169*, *237*, *238*, 251, *258*, *267*, *278*, *279*, *281*, *282*, *288*, *290–293*, *302*, *303*, *316*, *317*, *327*, *332*, *336*
is_sound_playing, 252, *270*, *271*, *280*, *282*, *283*, *289*, *306*, *307*, *316*, *318*, *334*
is_window_focused, *29*, *30*, *185*, *236*, 252, *253–257*, *275*, *277*, *287*, *310–314*, *340*
is_window_fullscreen, *29*, *30*, *185*, *236*, 253, *253–257*, *275*, *277*, *287*, *310–314*, *340*
is_window_hidden, *29*, *30*, *185*, *236*, *253*, 254, *255–257*, *275*, *277*, *287*, *310–314*, *340*
is_window_maximized, *29*, *30*, *185*, *236*, *253*, 254, *254–257*, *275*, *277*, *287*, *310–314*, *340*
is_window_minimized, *29*, *30*, *185*, *236*, *253*, *254*, 255, *255–257*, *275*, *277*, *287*, *310–314*, *340*
is_window_ready, *29*, *30*, *185*, *236*, *253–255*, 256, *257*, *275*, *277*, *287*, *310–314*, *340*
is_window_resized, *29*, *30*, *185*, *236*, *253–255*, 256, *256*, *257*, *275*, *277*, *287*, *310–314*, *340*
is_window_state, *29*, *30*, *185*, *236*, *253–256*, 257, *257*, *275*, *277*, *287*, *310–314*, *340*

load_audio_stream, *30*, *235–238*, *251*, 258, *267*, *278*, *279*, *281*, *282*, *288*, *290–293*, *316*, *317*, *327*, *332*, *336*
load_file_text, *103–106*, *131*, *144–146*, *221*, *222*, *232*, *234*, *240*, *241*, 259, *276*, *320–324*
load_font, *147*, 259, *260*, *261*, *327*, *328*
load_font_ex, *147*, 260, *260*, *261*, *327*, *328*
load_font_from_image, *127*, *128*, *132–138*, *147*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–234*, *260*, 261, *262–265*, *273*, *327–330*
load_image, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–232*, *234*, *261*, 262, *263–265*, *273*, *329*, *330*
load_image_from_screen, *127*, *128*, *132–138*, *157*, *158*, *177*, *178*, *187*, *188*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–234*, *240*, *261*, *262*, 263, *264*, *265*, *273*, *329*, *330*
load_image_from_texture, *14*, *50*, *51*, *108–111*, *113–115*, *117*, *118*, *126–128*, *132–139*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–232*, *234*, *261–263*, 264, *265*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, *325*, *329*, *330*, *332*, *334*
load_image_raw, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–232*, *234*, *261–264*, 265, *273*, *329*, *330*
load_model, *72*, *73*, *75*, *159*, *173*, 266, *331*
load_music_stream, *168*, *169*, *237*, *238*, *251*,

258, 266, *278, 279*, *281, 282*, *288*, *290–293*, *302, 303*, *316, 317*, *327*, *332*, *336*
load_render_texture, *14*, *50, 51*, *108–111*, *113–115*, *117, 118*, *126*, *139*, *264*, 267, *272, 273*, *305, 306*, *308, 309*, *325*, *332*, *334*
load_shader, *14*, *125*, *179, 180*, 268, *269*, *303–305*, *333*
load_shader_from_memory, *14*, *125*, *179, 180*, *268*, 269, *303–305*, *333*
load_sound, *252*, 269, *271*, *280*, *282, 283*, *289*, *306, 307*, *316*, *318*, *334*
load_sound_from_wave, *129, 130*, *252*, 270, *270*, *274*, *280*, *282, 283*, *289*, *306, 307*, *316*, *318*, *334, 335*
load_storage_value, 271
load_texture, *14*, *50, 51*, *108–111*, *113–115*, *117, 118*, *126*, *139*, *265*, *268*, 271, *273*, *305, 306*, *308, 309*, *325*, *332*, *334*
load_texture_from_image, *14*, *50, 51*, *108–111*, *113–115*, *117, 118*, *126–128*, *132–139*, *157, 158*, *191–204*, *206, 207*, *209, 210*, *212–214*, *216, 217*, *219, 220*, *222–232*, *234*, *261–265*, *268*, 272, *272*, *305, 306*, *308, 309*, *325*, *329, 330*, *332*, *334*
load_wave, *129, 130*, *271*, 273, *335*

material_map, 274
maximize_window, *29, 30*, *185*, *236*, *253–257*, *275*, *277*, *287*, *310–314*, *340*
measure_text, *103–106*, *221, 222*, *232*, *234*, *259*, 275, *276*, *320–324*
measure_text_ex, *103–106*, *221, 222*, *232*, *234*, *259*, 276, *276*, *320–324*
mesh(), *172*
minimize_window, *29, 30*, *185*, *236*, *253–257*, *275*, 277, *287*, *310–314*, *340*
model(), *72–74*, *159*, *173*, *331*
music(), *168, 169*, *251*, *279*, *282*, *288*, *290*, *302, 303*, *317*, *332*, *336*

npatch_info, 277

p1, *23*
p2, *23*
pause_audio_stream, *30*, *235–238*, *251*, *258*, *267*, 278, *279*, *281, 282*, *288*, *290–293*, *316, 317*, *327*, *332*, *336*
pause_music_stream, *168, 169*, *237, 238*, *251*, *258*, *267*, *278*, 279, *281, 282*, *288*, *290–293*, *302, 303*, *316, 317*, *327*, *332*, *336*
pause_sound, *252*, *270, 271*, 280, *282, 283*, *289*, *306, 307*, *316*, *318*, *334*
play_audio_stream, *30*, *235–238*, *251*, *258*, *267*, *278, 279*, 280, *282*, *288*, *290–293*, *316, 317*, *327*, *332*, *336*
play_music_stream, *168, 169*, *237, 238*, *251*, *258*, *267*, *279*, 281, *281*, *288*, *290–293*, *302, 303*, *316, 317*, *327*, *332*, *336*
play_sound, *252*, *270, 271*, *280*, 282, *283*, *289*, *306, 307*, *316*, *318*, *334*
play_sound_multi, *252*, *270, 271*, *280*, *282*, 283, *289*, *306, 307*, *316*, *318*, *334*

ray, 283
ray_collision, 284
raylib_version, 285
rectangle, *84–87*, *89–93*, *95, 96*, *215, 216*, *218, 219*, 285
rectangle(), *20*, *24*, *26*, *51*, *85*, *89*, *91–94*, *110*, *112*, *114–116*, *141*, *202*, *204*, *216, 217*, *226*, *306*
render_texture, 286
restore_window, *29, 30*, *185*, *236*, *253–257*, *275*, *277*, 287, *310–314*, *340*
resume_audio_stream, *30*, *235–238*, *251*, *258*, *267*, *278, 279*, *281, 282*, 287, *288*, *290–293*, *316, 317*, *327*, *332*, *336*
resume_music_stream, *168, 169*, *237, 238*, *251*, *258*, *267*, *279*, *281, 282*, 288, *288*, *290–293*, *302, 303*, *316, 317*, *327*, *332*, *336*
resume_sound, *252*, *270, 271*, *280*, *282, 283*, 289, *306, 307*, *316*, *318*, *334*

save_storage_value, 289
seek_music_stream, *168, 169*, *237, 238*, *251*, *258*, *267*, *279*, *281, 282*, *288*, 290, *291–293*, *302, 303*, *316, 317*, *327*, *332*, *336*

set_audio_stream_buffer_size_default, *30*, *235–238*, *251*, *258*, *267*, *278, 279*, *281, 282*, *288*, *290*, 291, *292, 293*, *316, 317*, *327*, *332*, *336*
set_audio_stream_pitch, *30*, *235–238*, *251*, *258*, *267*, *278, 279*, *281, 282*, *288*, *290*, 291, *291*, *293*, *316, 317*, *327*, *332*, *336*
set_audio_stream_volume, *30*, *235–238*, *251*, *258*, *267*, *278, 279*, *281, 282*, *288*, *290, 291*, 292, *292*, *299*, *303*, *307*, *316, 317*, *327*, *332*, *336*
set_camera_alt_control, *139, 140*, 293, *294–296*, *336*
set_camera_mode, *11–14*, *123–127*, *139, 140*, *293*, 294, *295, 296*, *336*
set_camera_move_controls, *139, 140*, *293*, 294, *294*, *296*, *336*
set_camera_pan_control, *139, 140*, *293, 294*, 295, *295, 296*, *336*
set_camera_smooth_zoom_control, *139, 140*, *293–295*, 296, *296*, *336*
set_exit_key, *141*, *158*, *246–248*, 297
set_gamepad_mappings, *149–151*, *241–244*, 297
set_gestures_enabled, 298
set_master_volume, *293*, 298, *303*, *307*
set_mouse_cursor, *37*, *122*, *165–168*, *190*, *239, 240*, *248–250*, 299, *300, 301*, *315*
set_mouse_offset, *165–168*, *248–250*, *299*, 300, *301*
set_mouse_position, *165–168*, *248–250*, *299*, 300, *300, 301*
set_mouse_scale, *165–168*, *248–250*, *299, 300*, 301, *301*
set_music_pitch, *168, 169*, *251*, *267*, *279*, *282*, *288*, *290*, 302, *303*, *317*, *332*, *336*
set_music_volume, *168, 169*, *251*, *267*, *279*, *282*, *288*, *290*, *293*, *299*, 302, *302*, *307*, *317*, *332*, *336*
set_shader_value, *14*, *125*, *179, 180*, *268, 269*, 303, *304, 305*, *333*
set_shader_value_matrix, *14*, *125*, *179, 180*, *268, 269*, *303*, 304, *305*, *333*
set_shader_value_texture, *14*, *50, 51*, *108–111*, *113–115*, *117, 118*, *125, 126*, *139*, *179, 180*, *265*, *268, 269*, *272, 273*, *303*, 304, *304*, *306*, *308, 309*, *325*, *332–334*
set_shapes_texture, *14*, *50, 51*, *108–111*, *113–115*, *117, 118*, *126*, *139*, *265*, *268*, *272, 273*, 305, *305*, *308, 309*, *325*, *332*, *334*
set_sound_pitch, *252*, *270, 271*, *280*, *282, 283*, *289*, 306, *307*, *316*, *318*, *334*
set_sound_volume, *252*, *270, 271*, *280*, *282, 283*, *289*, *293*, *299*, *303*, *306*, 307, *316*, *318*, *334*
set_target_fps, *62*, *147*, 307
set_texture_filter, *14*, *50, 51*, *108–111*, *113–115*, *117, 118*, *126*, *139*, *265*, *268*, *272, 273*, *305, 306*, 308, *309*, *325*, *332*, *334*
set_texture_wrap, *14*, *50, 51*, *108–111*, *113–115*, *117, 118*, *126*, *139*, *265*, *268*, *272, 273*, *305, 306*, *308*, 309, *325*, *332*, *334*
set_trace_log_level, 309
set_window_icon, *29, 30*, *185*, *236*, *253–257*, *275*, *277*, *287*, 310, *311–314*, *340*
set_window_min_size, *29, 30*, *185*, *236*, *253–257*, *275*, *277*, *287*, *310*, 311, *312–314*, *340*
set_window_monitor, *29, 30*, *143*, *160–164*, *185*, *236*, *253–257*, *275*, *277*, *287*, *310*, 311, *311–314*, *340*
set_window_position, *29, 30*, *185*, *236*, *253–257*, *275*, *277*, *287*, *310, 311*, 312, *312–314*, *340*
set_window_size, *29, 30*, *185*, *236*, *253–257*, *275*, *277*, *287*, *310–312*, 313, *314*, *340*
set_window_state, *29, 30*, *185*, *236*, *253–257*, *275*, *277*, *287*, *310–312*, 313, *313, 314*, *340*
set_window_title, *29, 30*, *185*, *236*, *253–257*, *275*, *277*, *287*, *310–313*, 314, *314*, *340*
shader(), *14*, *179, 180*, *304, 305*, *333*
show_cursor, *37*, *122*, *190*, *239, 240*, *299*, 315
sound, *252*, *270, 271*, *280*, *282, 283*, *289*, *306, 307*, 315, *318*, *334*
sound(), *252*, *280*, *282, 283*, *289*, *306, 307*, *318*, *334*

stop_audio_stream, *30*, *235–238*, *251*, *258*, *267*, *278*, *279*, *281*, *282*, *288*, *290–293*, 316, *317*, *327*, *332*, *336*
stop_music_stream, *168*, *169*, *237*, *238*, *251*, *258*, *267*, *279*, *281*, *282*, *288*, *290–293*, *302*, *303*, *316*, 317, *327*, *332*, *336*
stop_sound, *252*, *270*, *271*, *280*, *282*, *283*, *289*, *306*, *307*, *316*, 317, *318*, *334*
stop_sound_multi, *252*, *270*, *271*, *280*, *282*, *283*, *289*, *306*, *307*, *316*, 318, *318*, *334*

take_screenshot, 319
text_find_index, *103–106*, *221*, *222*, *232*, *234*, *259*, *276*, 319, *320–324*
text_insert, *103–106*, *221*, *222*, *232*, *234*, *259*, *276*, 320, *320–324*
text_replace, *103–106*, *221*, *222*, *232*, *234*, *259*, *276*, *320*, 321, *322–324*
text_to_integer, *103–106*, *221*, *222*, *232*, *234*, *259*, *276*, *320*, 321, *321*, *323*, *324*
text_to_lower, *103–106*, *221*, *222*, *232*, *234*, *259*, *276*, *320*, *321*, 322, *322–324*
text_to_pascal, *103–106*, *221*, *222*, *232*, *234*, *259*, *276*, *320–322*, 323, *323*, *324*
text_to_upper, *103–106*, *221*, *222*, *232*, *234*, *259*, *276*, *320–323*, 324
texture, *14*, *50*, *51*, *108–111*, *113–115*, *117*, *118*, *126*, *139*, *265*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, 324, *332*, *334*
texture(), *38*, *50*, *51*, *107*, *108*, *110–112*, *114–117*, *139*, *264*, *305*, *306*, *308*, *309*, *334*
threshold, *23*
toggle_fullscreen, 325
transform, 326

unload_audio_stream, *30*, *235–238*, *251*, *258*, *267*, *278*, *279*, *281*, *282*, *288*, *290–293*, *316*, *317*, 326, *332*, *336*
unload_font, *147*, *260*, *261*, 327, *328*
unload_font_data, *147*, *260*, *261*, *327*, 328
unload_image, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–232*, *234*, *261–265*, *273*, 328, *330*
unload_image_colors, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–232*, *234*, *261–265*, *273*, 329, *329*, *330*
unload_image_palette, *127*, *128*, *132–138*, *157*, *158*, *191–204*, *206*, *207*, *209*, *210*, *212–214*, *216*, *217*, *219*, *220*, *222–232*, *234*, *261–265*, *273*, *329*, 330, *330*
unload_model, *72*, *73*, *75*, *159*, *173*, *266*, 331
unload_music_stream, *168*, *169*, *237*, *238*, *251*, *258*, *267*, *279*, *281*, *282*, *288*, *290–293*, *302*, *303*, *316*, *317*, *327*, 331, *336*
unload_render_texture, *14*, *50*, *51*, *108–111*, *113–115*, *117*, *118*, *126*, *139*, *265*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, *325*, 332, *334*
unload_shader, *14*, *125*, *179*, *180*, *268*, *269*, *303–305*, 333
unload_sound, *252*, *270*, *271*, *280*, *282*, *283*, *289*, *306*, *307*, *316*, *318*, 333
unload_texture, *14*, *50*, *51*, *108–111*, *113–115*, *117*, *118*, *126*, *139*, *265*, *268*, *272*, *273*, *305*, *306*, *308*, *309*, *325*, *332*, 334
unload_wave, *129*, *130*, *271*, *274*, 335
update_camera, *139*, *140*, *293–296*, 335
update_music_stream, *168*, *169*, *237*, *238*, *251*, *258*, *267*, *279*, *281*, *282*, *288*, *290–293*, *302*, *303*, *316*, *317*, *327*, *332*, 336

vector_2_angle, 337
vector_2_distance, 337
vector_2_move_towards, 338
vector_2_rotate, 339

wave(), *129*, *130*, *271*, *335*
window_should_close, *29*, *30*, *185*, *236*, *253–257*, *275*, *277*, *287*, *310–314*, 339