

GaeGebra

v0.1

Generated by Doxygen 1.10.0

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 _UIDropdownItem Struct Reference	5
3.1.1 Member Data Documentation	5
3.1.1.1 base	5
3.1.1.2 dropdown_index	5
3.1.1.3 mouse_state	5
3.1.1.4 parent_dropdown	5
3.1.1.5 text	6
3.2 _UISplitButtonItem Struct Reference	6
3.2.1 Member Data Documentation	6
3.2.1.1 base	6
3.2.1.2 mouse_state	6
3.2.1.3 parent_splitbutton	6
3.2.1.4 splitbutton_index	6
3.2.1.5 text	6
3.3 AppData Struct Reference	7
3.3.1 Member Data Documentation	7
3.3.1.1 delta_time	7
3.3.1.2 frame_start	7
3.3.1.3 last_frame_start	7
3.3.1.4 target_frame_time	7
3.3.1.5 windows	7
3.4 Circle Struct Reference	7
3.4.1 Member Data Documentation	8
3.4.1.1 base	8
3.4.1.2 center	8
3.4.1.3 perimeter_point	8
3.5 CoordinateSystem Struct Reference	8
3.5.1 Member Data Documentation	8
3.5.1.1 origin	8
3.5.1.2 position	9
3.5.1.3 shapes	9
3.5.1.4 size	9
3.5.1.5 zoom	9
3.6 Font Struct Reference	9
3.6.1 Member Data Documentation	9
3.6.1.1 font	9

3.6.1.2 size	9
3.7 InputData Struct Reference	10
3.7.1 Member Data Documentation	10
3.7.1.1 current_keyboard_state	10
3.7.1.2 current_mouse_button_state	10
3.7.1.3 current_mouse_position	10
3.7.1.4 key_count	10
3.7.1.5 mouse_wheel_delta	10
3.7.1.6 old_keyboard_state	10
3.7.1.7 old_mouse_button_state	10
3.7.1.8 old_mouse_position	11
3.8 IShape Struct Reference	11
3.8.1 Member Data Documentation	11
3.8.1.1 destroy	11
3.8.1.2 draw	11
3.8.1.3 is_defined_by	11
3.8.1.4 overlap_point	11
3.8.1.5 translate	11
3.9 Line Struct Reference	12
3.9.1 Member Data Documentation	12
3.9.1.1 base	12
3.9.1.2 p1	12
3.9.1.3 p2	12
3.10 Point Struct Reference	12
3.10.1 Member Data Documentation	12
3.10.1.1 base	12
3.10.1.2 coordinates	13
3.11 Texture Struct Reference	13
3.11.1 Member Data Documentation	13
3.11.1.1 height	13
3.11.1.2 texture	13
3.11.1.3 width	13
3.12 UIButton Struct Reference	13
3.12.1 Member Data Documentation	14
3.12.1.1 base	14
3.12.1.2 color	14
3.12.1.3 corner_radius	14
3.12.1.4 mouse_state	14
3.12.1.5 on_click	14
3.12.1.6 text	14
3.12.1.7 text_color	14
3.12.1.8 text_position	15

3.13 UICheckbox Struct Reference	15
3.13.1 Member Data Documentation	15
3.13.1.1 base	15
3.13.1.2 checked	15
3.13.1.3 checked_color	15
3.13.1.4 corner_radius	15
3.13.1.5 mouse_state	15
3.13.1.6 on_checked_changed	16
3.13.1.7 unchecked_color	16
3.14 UIConstraint Struct Reference	16
3.14.1 Member Data Documentation	16
3.14.1.1 constraint_type	16
3.14.1.2 recalculate	16
3.14.1.3 value	16
3.15 UIConstraints Struct Reference	16
3.15.1 Member Data Documentation	17
3.15.1.1 height	17
3.15.1.2 width	17
3.15.1.3 x	17
3.15.1.4 y	17
3.16 UIContainer Struct Reference	17
3.16.1 Member Data Documentation	17
3.16.1.1 base	17
3.16.1.2 children	18
3.16.1.3 on_size_changed	18
3.17 UIData Struct Reference	18
3.17.1 Member Data Documentation	18
3.17.1.1 backspace_pressed	18
3.17.1.2 expanded_splitbutton	18
3.17.1.3 main_container	18
3.17.1.4 mouse_captured	18
3.17.1.5 text_input	19
3.18 UIDropdownList Struct Reference	19
3.18.1 Member Data Documentation	19
3.18.1.1 base	19
3.18.1.2 color	19
3.18.1.3 corner_radius	19
3.18.1.4 expanded	19
3.18.1.5 items	19
3.18.1.6 on_selection_changed	20
3.18.1.7 selected_item	20
3.18.1.8 text_color	20

3.19 UIElement Struct Reference	20
3.19.1 Member Data Documentation	20
3.19.1.1 constraints	20
3.19.1.2 destroy	20
3.19.1.3 parent	20
3.19.1.4 position	21
3.19.1.5 recalculate	21
3.19.1.6 render	21
3.19.1.7 size	21
3.19.1.8 update	21
3.20 UIButton Struct Reference	21
3.20.1 Member Data Documentation	21
3.20.1.1 base	21
3.20.1.2 mouse_state	22
3.20.1.3 on_click	22
3.20.1.4 texture	22
3.21 UILabel Struct Reference	22
3.21.1 Member Data Documentation	22
3.21.1.1 base	22
3.21.1.2 color	22
3.21.1.3 text	22
3.22 UIPanel Struct Reference	23
3.22.1 Member Data Documentation	23
3.22.1.1 base	23
3.22.1.2 border_color	23
3.22.1.3 border_width	23
3.22.1.4 color	23
3.22.1.5 corner_radius	23
3.23 UISlider Struct Reference	23
3.23.1 Member Data Documentation	24
3.23.1.1 base	24
3.23.1.2 color	24
3.23.1.3 corner_radius	24
3.23.1.4 mouse_state	24
3.23.1.5 on_value_changed	24
3.23.1.6 slider_color	24
3.23.1.7 thickness	24
3.23.1.8 value	25
3.24 UISplitButton Struct Reference	25
3.24.1 Member Data Documentation	25
3.24.1.1 auto_dropdown	25
3.24.1.2 base	25

3.24.1.3 color	25
3.24.1.4 corner_radius	25
3.24.1.5 expanded	25
3.24.1.6 items	26
3.24.1.7 on_item_clicked	26
3.24.1.8 text_color	26
3.25 UITextbox Struct Reference	26
3.25.1 Member Data Documentation	26
3.25.1.1 base	26
3.25.1.2 color	26
3.25.1.3 corner_radius	26
3.25.1.4 focused	27
3.25.1.5 mouse_state	27
3.25.1.6 on_text_changed	27
3.25.1.7 text	27
3.25.1.8 text_color	27
3.26 Vector Struct Reference	27
3.26.1 Member Data Documentation	27
3.26.1.1 capacity	27
3.26.1.2 data	28
3.26.1.3 size	28
3.27 Vector2 Struct Reference	28
3.27.1 Member Data Documentation	28
3.27.1.1 x	28
3.27.1.2 y	28
3.28 Window Struct Reference	28
3.28.1 Member Data Documentation	29
3.28.1.1 close_requested	29
3.28.1.2 input_data	29
3.28.1.3 renderer	29
3.28.1.4 ui_data	29
3.28.1.5 window	29
4 File Documentation	31
4.1 src/app/app.c File Reference	31
4.1.1 Function Documentation	31
4.1.1.1 _app_add_window()	31
4.1.1.2 app_close()	32
4.1.1.3 app_get_delta_time()	32
4.1.1.4 app_get_time()	32
4.1.1.5 app_get_windows()	32
4.1.1.6 app_init()	32

4.1.1.7 app_render()	32
4.1.1.8 app_request_close()	32
4.1.1.9 app_set_target()	32
4.1.1.10 app_set_target_fps()	32
4.1.1.11 app_update()	32
4.1.2 Variable Documentation	33
4.1.2.1 app_data	33
4.2 src/app/app.h File Reference	33
4.2.1 Typedef Documentation	33
4.2.1.1 AppData	33
4.2.2 Function Documentation	33
4.2.2.1 _app_add_window()	33
4.2.2.2 app_close()	34
4.2.2.3 app_get_delta_time()	34
4.2.2.4 app_get_time()	34
4.2.2.5 app_get_windows()	34
4.2.2.6 app_init()	34
4.2.2.7 app_render()	34
4.2.2.8 app_request_close()	34
4.2.2.9 app_set_target()	34
4.2.2.10 app_set_target_fps()	34
4.2.2.11 app_update()	34
4.3 app.h	35
4.4 src/color/color.c File Reference	35
4.4.1 Function Documentation	35
4.4.1.1 color_clever_shift()	35
4.4.1.2 color_fade()	36
4.4.1.3 color_from_grayscale()	36
4.4.1.4 color_from_hex()	36
4.4.1.5 color_from_hsv()	36
4.4.1.6 color_from_rgb()	36
4.4.1.7 color_from_rgba()	36
4.4.1.8 color_shift()	36
4.5 src/color/color.h File Reference	37
4.5.1 Macro Definition Documentation	37
4.5.1.1 BLACK	37
4.5.1.2 BLUE	37
4.5.1.3 CYAN	37
4.5.1.4 DARK_GRAY	38
4.5.1.5 GRAY	38
4.5.1.6 GREEN	38
4.5.1.7 MAGENTA	38

4.5.1.8 RED	38
4.5.1.9 TRANSPARENT	38
4.5.1.10 WHITE	38
4.5.1.11 YELLOW	38
4.5.2 Typedef Documentation	38
4.5.2.1 Color	38
4.5.3 Function Documentation	39
4.5.3.1 color_clever_shift()	39
4.5.3.2 color_fade()	39
4.5.3.3 color_from_grayscale()	39
4.5.3.4 color_from_hex()	39
4.5.3.5 color_from_hsv()	39
4.5.3.6 color_from_rgb()	39
4.5.3.7 color_from_rgba()	39
4.5.3.8 color_shift()	40
4.6 color.h	40
4.7 src/font/font.c File Reference	40
4.7.1 Function Documentation	41
4.7.1.1 _font_close()	41
4.7.1.2 _font_init()	41
4.7.1.3 font_load()	41
4.7.2 Variable Documentation	41
4.7.2.1 fonts	41
4.8 src/font/font.h File Reference	41
4.8.1 Typedef Documentation	41
4.8.1.1 Font	41
4.8.2 Function Documentation	42
4.8.2.1 _font_close()	42
4.8.2.2 _font_init()	42
4.8.2.3 font_load()	42
4.9 font.h	42
4.10 src/geometry/coordinate_system/coordinate_system.c File Reference	42
4.10.1 Function Documentation	43
4.10.1.1 coordinate_system_create()	43
4.10.1.2 coordinate_system_destroy()	43
4.10.1.3 coordinate_system_draw()	44
4.10.1.4 coordinate_system_get_hovered_shape()	44
4.10.1.5 coordinate_system_is_hovered()	44
4.10.1.6 coordinate_system_translate()	45
4.10.1.7 coordinate_system_update_dimensions()	45
4.10.1.8 coordinate_system_zoom()	45
4.10.1.9 coordinates_to_screen()	45

4.10.1.10 screen_to_coordinates()	46
4.11 src/geometry/coordinate_system/coordinate_system.h File Reference	46
4.11.1 Macro Definition Documentation	47
4.11.1.1 INITIAL_ZOOM	47
4.11.2 Typedef Documentation	47
4.11.2.1 CoordinateSystem	47
4.11.3 Function Documentation	47
4.11.3.1 coordinate_system_create()	47
4.11.3.2 coordinate_system_destroy()	48
4.11.3.3 coordinate_system_draw()	48
4.11.3.4 coordinate_system_get_hovered_shape()	48
4.11.3.5 coordinate_system_is_hovered()	49
4.11.3.6 coordinate_system_translate()	49
4.11.3.7 coordinate_system_update_dimensions()	49
4.11.3.8 coordinate_system_zoom()	50
4.11.3.9 coordinates_to_screen()	50
4.11.3.10 screen_to_coordinates()	50
4.12 coordinate_system.h	51
4.13 src/geometry/shape/shape.c File Reference	51
4.13.1 Function Documentation	52
4.13.1.1 circle_create()	52
4.13.1.2 line_create()	52
4.13.1.3 point_create()	52
4.14 src/geometry/shape/shape.h File Reference	53
4.14.1 Macro Definition Documentation	53
4.14.1.1 OVERLAP_DISTANCE	53
4.14.2 Typedef Documentation	54
4.14.2.1 Circle	54
4.14.2.2 CoordinateSystem	54
4.14.2.3 IShape	54
4.14.2.4 Line	54
4.14.2.5 Point	54
4.14.3 Function Documentation	54
4.14.3.1 circle_create()	54
4.14.3.2 line_create()	55
4.14.3.3 point_create()	55
4.15 shape.h	55
4.16 src/geometry/vector2/vector2.c File Reference	56
4.16.1 Function Documentation	57
4.16.1.1 vector2_add()	57
4.16.1.2 vector2_angle()	57
4.16.1.3 vector2_create()	57

4.16.1.4	vector2_distance()	57
4.16.1.5	vector2_divide()	57
4.16.1.6	vector2_dot()	57
4.16.1.7	vector2_down()	57
4.16.1.8	vector2_from_point()	57
4.16.1.9	vector2_from_polar()	58
4.16.1.10	vector2_left()	58
4.16.1.11	vector2_length()	58
4.16.1.12	vector2_multiply()	58
4.16.1.13	vector2_negate()	58
4.16.1.14	vector2_normalize()	58
4.16.1.15	vector2_one()	58
4.16.1.16	vector2_reflect()	58
4.16.1.17	vector2_right()	58
4.16.1.18	vector2_rotate()	59
4.16.1.19	vector2_rotate90()	59
4.16.1.20	vector2_scale()	59
4.16.1.21	vector2_subtract()	59
4.16.1.22	vector2_up()	59
4.16.1.23	vector2_zero()	59
4.17	src/geometry/vector2/vector2.h File Reference	59
4.17.1	Typedef Documentation	60
4.17.1.1	Vector2	60
4.17.2	Function Documentation	60
4.17.2.1	vector2_add()	60
4.17.2.2	vector2_angle()	60
4.17.2.3	vector2_create()	60
4.17.2.4	vector2_distance()	61
4.17.2.5	vector2_divide()	61
4.17.2.6	vector2_dot()	61
4.17.2.7	vector2_down()	61
4.17.2.8	vector2_from_point()	61
4.17.2.9	vector2_from_polar()	61
4.17.2.10	vector2_left()	61
4.17.2.11	vector2_length()	61
4.17.2.12	vector2_multiply()	62
4.17.2.13	vector2_negate()	62
4.17.2.14	vector2_normalize()	62
4.17.2.15	vector2_one()	62
4.17.2.16	vector2_reflect()	62
4.17.2.17	vector2_right()	62
4.17.2.18	vector2_rotate()	62

4.17.2.19	vector2_rotate90()	62
4.17.2.20	vector2_scale()	63
4.17.2.21	vector2_subtract()	63
4.17.2.22	vector2_up()	63
4.17.2.23	vector2_zero()	63
4.18	vector2.h	63
4.19	src/includes.h File Reference	64
4.20	includes.h	64
4.21	src/input/input.c File Reference	64
4.21.1	Function Documentation	65
4.21.1.1	_input_close()	65
4.21.1.2	_input_handle_event()	65
4.21.1.3	_input_init()	65
4.21.1.4	_input_reset()	65
4.21.1.5	_input_set_target()	65
4.21.1.6	input_get_mouse_motion()	65
4.21.1.7	input_get_mouse_position()	66
4.21.1.8	input_get_mouse_wheel_delta()	66
4.21.1.9	input_is_key_down()	66
4.21.1.10	input_is_key_pressed()	66
4.21.1.11	input_is_key_released()	66
4.21.1.12	input_is_mouse_button_down()	66
4.21.1.13	input_is_mouse_button_pressed()	66
4.21.1.14	input_is_mouse_button_released()	66
4.21.2	Variable Documentation	66
4.21.2.1	target_input_data	66
4.22	src/input/input.h File Reference	67
4.22.1	Typedef Documentation	67
4.22.1.1	InputData	67
4.22.2	Function Documentation	67
4.22.2.1	_input_close()	67
4.22.2.2	_input_handle_event()	67
4.22.2.3	_input_init()	68
4.22.2.4	_input_reset()	68
4.22.2.5	_input_set_target()	68
4.22.2.6	input_get_mouse_motion()	68
4.22.2.7	input_get_mouse_position()	68
4.22.2.8	input_get_mouse_wheel_delta()	68
4.22.2.9	input_is_key_down()	68
4.22.2.10	input_is_key_pressed()	68
4.22.2.11	input_is_key_released()	68
4.22.2.12	input_is_mouse_button_down()	69

4.22.2.13 input_is_mouse_button_pressed()	69
4.22.2.14 input_is_mouse_button_released()	69
4.23 input.h	69
4.24 src/main.c File Reference	70
4.24.1 Detailed Description	71
4.24.2 Macro Definition Documentation	71
4.24.2.1 FPS	71
4.24.2.2 MOUSE_WHEEL_SENSITIVITY	71
4.24.3 Typedef Documentation	71
4.24.3.1 State	71
4.24.4 Enumeration Type Documentation	71
4.24.4.1 State	71
4.24.5 Function Documentation	72
4.24.5.1 main()	72
4.24.5.2 on_canvas_size_changed()	72
4.24.5.3 on_circle_clicked() [1/2]	72
4.24.5.4 on_circle_clicked() [2/2]	72
4.24.5.5 on_editmenu_clicked() [1/2]	72
4.24.5.6 on_editmenu_clicked() [2/2]	72
4.24.5.7 on_filemenu_clicked() [1/2]	72
4.24.5.8 on_filemenu_clicked() [2/2]	72
4.24.5.9 on_line_clicked() [1/2]	73
4.24.5.10 on_line_clicked() [2/2]	73
4.24.5.11 on_point_clicked() [1/2]	73
4.24.5.12 on_point_clicked() [2/2]	73
4.24.5.13 on_pointer_clicked() [1/2]	73
4.24.5.14 on_pointer_clicked() [2/2]	73
4.24.6 Variable Documentation	73
4.24.6.1 cs	73
4.24.6.2 state	73
4.25 src/renderer/renderer.c File Reference	74
4.25.1 Function Documentation	74
4.25.1.1 _renderer_set_target()	74
4.25.1.2 renderer_bind_framebuffer()	75
4.25.1.3 renderer_clear()	75
4.25.1.4 renderer_create_framebuffer()	75
4.25.1.5 renderer_draw_arc()	75
4.25.1.6 renderer_draw_bezier()	75
4.25.1.7 renderer_draw_circle()	75
4.25.1.8 renderer_draw_ellipse()	75
4.25.1.9 renderer_draw_filled_circle()	76
4.25.1.10 renderer_draw_filled_ellipse()	76

4.25.1.11 <code>renderer_draw_filled_pie()</code>	76
4.25.1.12 <code>renderer_draw_filled_polygon()</code>	76
4.25.1.13 <code>renderer_draw_filled_rect()</code>	76
4.25.1.14 <code>renderer_draw_filled_rounded_rect()</code>	76
4.25.1.15 <code>renderer_draw_filled_triangle()</code>	77
4.25.1.16 <code>renderer_draw_line()</code>	77
4.25.1.17 <code>renderer_draw_pie()</code>	77
4.25.1.18 <code>renderer_draw_pixel()</code>	77
4.25.1.19 <code>renderer_draw_polygon()</code>	77
4.25.1.20 <code>renderer_draw_rect()</code>	78
4.25.1.21 <code>renderer_draw_rounded_rect()</code>	78
4.25.1.22 <code>renderer_draw_text()</code>	78
4.25.1.23 <code>renderer_draw_texture()</code>	78
4.25.1.24 <code>renderer_draw_triangle()</code>	78
4.25.1.25 <code>renderer_query_text_size()</code>	78
4.25.1.26 <code>renderer_reset_clip_rect()</code>	79
4.25.1.27 <code>renderer_resize_framebuffer()</code>	79
4.25.1.28 <code>renderer_set_clip_rect()</code>	79
4.25.1.29 <code>renderer_set_default_font()</code>	79
4.25.2 Variable Documentation	79
4.25.2.1 <code>default_font</code>	79
4.25.2.2 <code>target_renderer</code>	79
4.26 <code>src/renderer/renderer.h</code> File Reference	79
4.26.1 Function Documentation	80
4.26.1.1 <code>_renderer_set_target()</code>	80
4.26.1.2 <code>renderer_bind_framebuffer()</code>	80
4.26.1.3 <code>renderer_clear()</code>	80
4.26.1.4 <code>renderer_create_framebuffer()</code>	81
4.26.1.5 <code>renderer_draw_arc()</code>	81
4.26.1.6 <code>renderer_draw_bezier()</code>	81
4.26.1.7 <code>renderer_draw_circle()</code>	81
4.26.1.8 <code>renderer_draw_ellipse()</code>	81
4.26.1.9 <code>renderer_draw_filled_circle()</code>	81
4.26.1.10 <code>renderer_draw_filled_ellipse()</code>	82
4.26.1.11 <code>renderer_draw_filled_pie()</code>	82
4.26.1.12 <code>renderer_draw_filled_polygon()</code>	82
4.26.1.13 <code>renderer_draw_filled_rect()</code>	82
4.26.1.14 <code>renderer_draw_filled_rounded_rect()</code>	82
4.26.1.15 <code>renderer_draw_filled_triangle()</code>	83
4.26.1.16 <code>renderer_draw_line()</code>	83
4.26.1.17 <code>renderer_draw_pie()</code>	83
4.26.1.18 <code>renderer_draw_pixel()</code>	83

4.26.1.19 <code>renderer_draw_polygon()</code>	83
4.26.1.20 <code>renderer_draw_rect()</code>	84
4.26.1.21 <code>renderer_draw_rounded_rect()</code>	84
4.26.1.22 <code>renderer_draw_text()</code>	84
4.26.1.23 <code>renderer_draw_texture()</code>	84
4.26.1.24 <code>renderer_draw_triangle()</code>	84
4.26.1.25 <code>renderer_query_text_size()</code>	84
4.26.1.26 <code>renderer_reset_clip_rect()</code>	85
4.26.1.27 <code>renderer_resize_framebuffer()</code>	85
4.26.1.28 <code>renderer_set_clip_rect()</code>	85
4.26.1.29 <code>renderer_set_default_font()</code>	85
4.27 <code>renderer.h</code>	85
4.28 <code>src/texture/texture.c</code> File Reference	86
4.28.1 Function Documentation	86
4.28.1.1 <code>_texture_add()</code>	86
4.28.1.2 <code>_texture_close()</code>	86
4.28.1.3 <code>_texture_init()</code>	86
4.28.1.4 <code>texture_load()</code>	86
4.28.2 Variable Documentation	87
4.28.2.1 <code>textures</code>	87
4.29 <code>src/texture/texture.h</code> File Reference	87
4.29.1 Typedef Documentation	87
4.29.1.1 <code>Texture</code>	87
4.29.2 Function Documentation	87
4.29.2.1 <code>_texture_add()</code>	87
4.29.2.2 <code>_texture_close()</code>	87
4.29.2.3 <code>_texture_init()</code>	87
4.29.2.4 <code>texture_load()</code>	88
4.30 <code>texture.h</code>	88
4.31 <code>src/ui/ui.c</code> File Reference	88
4.31.1 Function Documentation	89
4.31.1.1 <code>_ui_close()</code>	89
4.31.1.2 <code>_ui_get_target()</code>	89
4.31.1.3 <code>_ui_handle_event()</code>	89
4.31.1.4 <code>_ui_init()</code>	89
4.31.1.5 <code>_ui_render()</code>	89
4.31.1.6 <code>_ui_set_target()</code>	89
4.31.1.7 <code>_ui_update()</code>	89
4.31.2 Variable Documentation	89
4.31.2.1 <code>target_ui_data</code>	89
4.32 <code>src/ui/ui.h</code> File Reference	90
4.32.1 Typedef Documentation	90

4.32.1.1 UIData	90
4.32.2 Function Documentation	90
4.32.2.1 _ui_close()	90
4.32.2.2 _ui_get_target()	90
4.32.2.3 _ui_handle_event()	90
4.32.2.4 _ui_init()	91
4.32.2.5 _ui_render()	91
4.32.2.6 _ui_set_target()	91
4.32.2.7 _ui_update()	91
4.33 ui.h	91
4.34 src/ui/ui_constraint/ui_constraint.c File Reference	91
4.34.1 Function Documentation	92
4.34.1.1 constraints_from_string()	92
4.34.1.2 new_aspect_constraint()	92
4.34.1.3 new_center_constraint()	92
4.34.1.4 new_offset_constraint()	92
4.34.1.5 new_pixel_constraint()	92
4.34.1.6 new_relative_constraint()	92
4.35 src/ui/ui_constraint/ui_constraint.h File Reference	92
4.35.1 Typedef Documentation	93
4.35.1.1 ConstraintType	93
4.35.1.2 UIConstraint	93
4.35.1.3 UIConstraints	93
4.35.2 Enumeration Type Documentation	93
4.35.2.1 ConstraintType	93
4.35.3 Function Documentation	94
4.35.3.1 constraints_from_string()	94
4.35.3.2 new_aspect_constraint()	94
4.35.3.3 new_center_constraint()	94
4.35.3.4 new_offset_constraint()	94
4.35.3.5 new_pixel_constraint()	94
4.35.3.6 new_relative_constraint()	94
4.36 ui_constraint.h	94
4.37 src/ui/ui_element/ui_element.c File Reference	95
4.37.1 Typedef Documentation	95
4.37.1.1 UIDropdownItem	95
4.37.1.2 UISplitButtonItem	96
4.37.2 Function Documentation	96
4.37.2.1 _ui_container_destroy()	96
4.37.2.2 _ui_container_recalculate()	96
4.37.2.3 _ui_container_render()	96
4.37.2.4 _ui_container_update()	96

4.37.2.5 ui_create_button()	96
4.37.2.6 ui_create_checkbox()	96
4.37.2.7 ui_create_container()	97
4.37.2.8 ui_create_dropdown()	97
4.37.2.9 ui_create_imagebutton()	97
4.37.2.10 ui_create_label()	97
4.37.2.11 ui_create_panel()	97
4.37.2.12 ui_create_slider()	97
4.37.2.13 ui_create_splitbutton()	98
4.37.2.14 ui_create_textbox()	98
4.38 src/ui/ui_element/ui_element.h File Reference	98
4.38.1 Macro Definition Documentation	99
4.38.1.1 UITEXT_MAX_LENGTH	99
4.38.2 Typedef Documentation	100
4.38.2.1 MouseState	100
4.38.2.2 UIButton	100
4.38.2.3 UICheckbox	100
4.38.2.4 UIContainer	100
4.38.2.5 UIDropdownList	100
4.38.2.6 UIElement	100
4.38.2.7 UIImageButton	100
4.38.2.8 UILabel	100
4.38.2.9 UIPanel	100
4.38.2.10 UISlider	100
4.38.2.11 UISplitButton	101
4.38.2.12 UITextbox	101
4.38.3 Enumeration Type Documentation	101
4.38.3.1 MouseState	101
4.38.4 Function Documentation	101
4.38.4.1 _ui_container_destroy()	101
4.38.4.2 _ui_container_recalculate()	101
4.38.4.3 _ui_container_render()	101
4.38.4.4 _ui_container_update()	101
4.38.4.5 ui_create_button()	102
4.38.4.6 ui_create_checkbox()	102
4.38.4.7 ui_create_container()	102
4.38.4.8 ui_create_dropdown()	102
4.38.4.9 ui_create_imagebutton()	102
4.38.4.10 ui_create_label()	102
4.38.4.11 ui_create_panel()	103
4.38.4.12 ui_create_slider()	103
4.38.4.13 ui_create_splitbutton()	103

4.38.4.14 ui_create_textbox()	103
4.39 ui_element.h	103
4.40 src/utls/math/math.c File Reference	106
4.40.1 Function Documentation	106
4.40.1.1 check_collision_point_rect()	106
4.40.1.2 clamp()	106
4.40.1.3 deg_to_rad()	106
4.40.1.4 lerp()	106
4.40.1.5 map()	107
4.40.1.6 rad_to_deg()	107
4.41 src/utls/math/math.h File Reference	107
4.41.1 Macro Definition Documentation	107
4.41.1.1 HALF_PI	107
4.41.1.2 PI	107
4.41.1.3 TWO_PI	108
4.41.2 Function Documentation	108
4.41.2.1 check_collision_point_rect()	108
4.41.2.2 clamp()	108
4.41.2.3 deg_to_rad()	108
4.41.2.4 lerp()	108
4.41.2.5 map()	108
4.41.2.6 rad_to_deg()	108
4.42 math.h	109
4.43 src/utls/vector/vector.c File Reference	109
4.43.1 Function Documentation	109
4.43.1.1 vector_clear()	109
4.43.1.2 vector_contains()	109
4.43.1.3 vector_create()	110
4.43.1.4 vector_free()	110
4.43.1.5 vector_get()	110
4.43.1.6 vector_insert()	110
4.43.1.7 vector_pop_back()	110
4.43.1.8 vector_push_back()	110
4.43.1.9 vector_remove()	110
4.43.1.10 vector_remove_at()	110
4.43.1.11 vector_reserve()	111
4.43.1.12 vector_size()	111
4.44 src/utls/vector/vector.h File Reference	111
4.44.1 Typedef Documentation	111
4.44.1.1 Vector	111
4.44.2 Function Documentation	112
4.44.2.1 vector_clear()	112

4.44.2.2	vector_contains()	112
4.44.2.3	vector_create()	112
4.44.2.4	vector_free()	112
4.44.2.5	vector_get()	112
4.44.2.6	vector_insert()	112
4.44.2.7	vector_pop_back()	112
4.44.2.8	vector_push_back()	112
4.44.2.9	vector_remove()	113
4.44.2.10	vector_remove_at()	113
4.44.2.11	vector_reserve()	113
4.44.2.12	vector_size()	113
4.45	vector.h	113
4.46	src/window/window.c File Reference	113
4.46.1	Function Documentation	114
4.46.1.1	_window_close()	114
4.46.1.2	_window_handle_event()	114
4.46.1.3	_window_render()	114
4.46.1.4	_window_reset()	114
4.46.1.5	_window_update()	114
4.46.1.6	window_create()	114
4.46.1.7	window_focus()	115
4.46.1.8	window_get_main_container()	115
4.46.1.9	window_hide()	115
4.46.1.10	window_show()	115
4.47	src/window/window.h File Reference	115
4.47.1	Typedef Documentation	116
4.47.1.1	Window	116
4.47.2	Function Documentation	116
4.47.2.1	_window_close()	116
4.47.2.2	_window_handle_event()	116
4.47.2.3	_window_render()	116
4.47.2.4	_window_reset()	116
4.47.2.5	_window_update()	116
4.47.2.6	window_create()	116
4.47.2.7	window_focus()	116
4.47.2.8	window_get_main_container()	117
4.47.2.9	window_hide()	117
4.47.2.10	window_show()	117
4.48	window.h	117

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

_UIDropdownItem	5
_UISplitButtonItem	6
AppData	7
Circle	7
CoordinateSystem	8
Font	9
InputData	10
IShape	11
Line	12
Point	12
Texture	13
UIButton	13
UICheckbox	15
UIConstraint	16
UIConstraints	16
UIContainer	17
UIData	18
UIDropdownList	19
UIElement	20
UIImageButton	21
UILabel	22
UIPanel	23
UISlider	23
UISplitButton	25
UITextbox	26
Vector	27
Vector2	28
Window	28

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

src/ includes.h	64
src/ main.c	
This is the entry point of the application	70
src/app/ app.c	31
src/app/ app.h	33
src/color/ color.c	35
src/color/ color.h	37
src/font/ font.c	40
src/font/ font.h	41
src/geometry/coordinate_system/ coordinate_system.c	42
src/geometry/coordinate_system/ coordinate_system.h	46
src/geometry/shape/ shape.c	51
src/geometry/shape/ shape.h	53
src/geometry/vector2/ vector2.c	56
src/geometry/vector2/ vector2.h	59
src/input/ input.c	64
src/input/ input.h	67
src/renderer/ renderer.c	74
src/renderer/ renderer.h	79
src/texture/ texture.c	86
src/texture/ texture.h	87
src/ui/ ui.c	88
src/ui/ ui.h	90
src/ui/ui_constraint/ ui_constraint.c	91
src/ui/ui_constraint/ ui_constraint.h	92
src/ui/ui_element/ ui_element.c	95
src/ui/ui_element/ ui_element.h	98
src/utils/math/ math.c	106
src/utils/math/ math.h	107
src/utils/vector/ vector.c	109
src/utils/vector/ vector.h	111
src/window/ window.c	113
src/window/ window.h	115

Chapter 3

Class Documentation

3.1 _UIDropdownItem Struct Reference

Public Attributes

- [UIElement](#) base
- [UIDropdownList](#) * parent_dropdown
- [Sint32](#) dropdown_index
- [char](#) text [UITEXT_MAX_LENGTH+1]
- [MouseState](#) mouse_state

3.1.1 Member Data Documentation

3.1.1.1 base

[UIElement](#) _UIDropdownItem::base

3.1.1.2 dropdown_index

[Sint32](#) _UIDropdownItem::dropdown_index

3.1.1.3 mouse_state

[MouseState](#) _UIDropdownItem::mouse_state

3.1.1.4 parent_dropdown

[UIDropdownList](#)* _UIDropdownItem::parent_dropdown

3.1.1.5 text

```
char _UIDropdownItem::text [UITEXT_MAX_LENGTH+1]
```

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.c](#)

3.2 _UISplitButtonItem Struct Reference

Public Attributes

- [UIElement](#) base
- [UISplitButton](#) * parent_splitbutton
- Sint32 splitbutton_index
- char text [UITEXT_MAX_LENGTH+1]
- [MouseState](#) mouse_state

3.2.1 Member Data Documentation

3.2.1.1 base

```
UIElement _UISplitButtonItem::base
```

3.2.1.2 mouse_state

```
MouseState _UISplitButtonItem::mouse_state
```

3.2.1.3 parent_splitbutton

```
UISplitButton* _UISplitButtonItem::parent_splitbutton
```

3.2.1.4 splitbutton_index

```
Sint32 _UISplitButtonItem::splitbutton_index
```

3.2.1.5 text

```
char _UISplitButtonItem::text [UITEXT_MAX_LENGTH+1]
```

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.c](#)

3.3 AppData Struct Reference

```
#include <app.h>
```

Public Attributes

- [Vector](#) * [windows](#)
- [Uint32](#) [target_frame_time](#)
- [Uint32](#) [last_frame_start](#)
- [Uint32](#) [frame_start](#)
- [double](#) [delta_time](#)

3.3.1 Member Data Documentation

3.3.1.1 [delta_time](#)

```
double AppData::delta_time
```

3.3.1.2 [frame_start](#)

```
Uint32 AppData::frame_start
```

3.3.1.3 [last_frame_start](#)

```
Uint32 AppData::last_frame_start
```

3.3.1.4 [target_frame_time](#)

```
Uint32 AppData::target_frame_time
```

3.3.1.5 [windows](#)

```
Vector* AppData::windows
```

The documentation for this struct was generated from the following file:

- [src/app/app.h](#)

3.4 Circle Struct Reference

```
#include <shape.h>
```

Public Attributes

- [IShape base](#)
- [Point * center](#)
- [Point * perimeter_point](#)

3.4.1 Member Data Documentation

3.4.1.1 base

[IShape](#) Circle::base

3.4.1.2 center

[Point*](#) Circle::center

3.4.1.3 perimeter_point

[Point*](#) Circle::perimeter_point

The documentation for this struct was generated from the following file:

- [src/geometry/shape/shape.h](#)

3.5 CoordinateSystem Struct Reference

```
#include <coordinate_system.h>
```

Public Attributes

- [Vector2 position](#)
- [Vector2 size](#)
- [Vector2 origin](#)
- double [zoom](#)
- [Vector *](#) [shapes](#)

3.5.1 Member Data Documentation

3.5.1.1 origin

[Vector2](#) CoordinateSystem::origin

3.5.1.2 position

```
Vector2 CoordinateSystem::position
```

3.5.1.3 shapes

```
Vector* CoordinateSystem::shapes
```

3.5.1.4 size

```
Vector2 CoordinateSystem::size
```

3.5.1.5 zoom

```
double CoordinateSystem::zoom
```

The documentation for this struct was generated from the following file:

- [src/geometry/coordinate_system/coordinate_system.h](#)

3.6 Font Struct Reference

```
#include <font.h>
```

Public Attributes

- TTF_Font * [font](#)
- int [size](#)

3.6.1 Member Data Documentation

3.6.1.1 font

```
TTF_Font* Font::font
```

3.6.1.2 size

```
int Font::size
```

The documentation for this struct was generated from the following file:

- [src/font/font.h](#)

3.7 InputData Struct Reference

```
#include <input.h>
```

Public Attributes

- bool [current_mouse_button_state](#) [5]
- bool [old_mouse_button_state](#) [5]
- SDL_Point [current_mouse_position](#)
- SDL_Point [old_mouse_position](#)
- int [mouse_wheel_delta](#)
- Uint8 * [current_keyboard_state](#)
- Uint8 * [old_keyboard_state](#)
- int [key_count](#)

3.7.1 Member Data Documentation

3.7.1.1 current_keyboard_state

```
Uint8* InputData::current_keyboard_state
```

3.7.1.2 current_mouse_button_state

```
bool InputData::current_mouse_button_state[5]
```

3.7.1.3 current_mouse_position

```
SDL_Point InputData::current_mouse_position
```

3.7.1.4 key_count

```
int InputData::key_count
```

3.7.1.5 mouse_wheel_delta

```
int InputData::mouse_wheel_delta
```

3.7.1.6 old_keyboard_state

```
Uint8* InputData::old_keyboard_state
```

3.7.1.7 old_mouse_button_state

```
bool InputData::old_mouse_button_state[5]
```

3.7.1.8 old_mouse_position

```
SDL_Point InputData::old_mouse_position
```

The documentation for this struct was generated from the following file:

- [src/input/input.h](#)

3.8 IShape Struct Reference

```
#include <shape.h>
```

Public Attributes

- `void(* draw)(CoordinateSystem *cs, IShape *self)`
- `void(* translate)(CoordinateSystem *cs, IShape *self, Vector2 translation)`
- `void(* destroy)(CoordinateSystem *cs, IShape *self)`
- `bool(* overlap_point)(CoordinateSystem *cs, IShape *self, Vector2 point)`
- `bool(* is_defined_by)(IShape *self, IShape *shape)`

3.8.1 Member Data Documentation

3.8.1.1 destroy

```
void(* IShape::destroy) (CoordinateSystem *cs, IShape *self)
```

3.8.1.2 draw

```
void(* IShape::draw) (CoordinateSystem *cs, IShape *self)
```

3.8.1.3 is_defined_by

```
bool(* IShape::is_defined_by) (IShape *self, IShape *shape)
```

3.8.1.4 overlap_point

```
bool(* IShape::overlap_point) (CoordinateSystem *cs, IShape *self, Vector2 point)
```

3.8.1.5 translate

```
void(* IShape::translate) (CoordinateSystem *cs, IShape *self, Vector2 translation)
```

The documentation for this struct was generated from the following file:

- [src/geometry/shape/shape.h](#)

3.9 Line Struct Reference

```
#include <shape.h>
```

Public Attributes

- [IShape base](#)
- [Point * p1](#)
- [Point * p2](#)

3.9.1 Member Data Documentation

3.9.1.1 base

[IShape](#) Line::base

3.9.1.2 p1

[Point*](#) Line::p1

3.9.1.3 p2

[Point](#) * Line::p2

The documentation for this struct was generated from the following file:

- src/geometry/shape/[shape.h](#)

3.10 Point Struct Reference

```
#include <shape.h>
```

Public Attributes

- [IShape base](#)
- [Vector2 coordinates](#)

3.10.1 Member Data Documentation

3.10.1.1 base

[IShape](#) Point::base

3.10.1.2 coordinates

`Vector2 Point::coordinates`

The documentation for this struct was generated from the following file:

- `src/geometry/shape/shape.h`

3.11 Texture Struct Reference

```
#include <texture.h>
```

Public Attributes

- `SDL_Texture *` `texture`
- `int` `width`
- `int` `height`

3.11.1 Member Data Documentation

3.11.1.1 height

```
int Texture::height
```

3.11.1.2 texture

```
SDL_Texture* Texture::texture
```

3.11.1.3 width

```
int Texture::width
```

The documentation for this struct was generated from the following file:

- `src/texture/texture.h`

3.12 UIButton Struct Reference

```
#include <ui_element.h>
```

Public Attributes

- [UIElement](#) base
- char [text](#) [[UITEXT_MAX_LENGTH](#)+1]
- [SDL_Point](#) [text_position](#)
- [Color](#) color
- [Color](#) [text_color](#)
- [Uint32](#) [corner_radius](#)
- [MouseState](#) [mouse_state](#)
- void(* [on_click](#))([UIButton](#) *self)

3.12.1 Member Data Documentation

3.12.1.1 base

[UIElement](#) [UIButton::base](#)

3.12.1.2 color

[Color](#) [UIButton::color](#)

3.12.1.3 corner_radius

[Uint32](#) [UIButton::corner_radius](#)

3.12.1.4 mouse_state

[MouseState](#) [UIButton::mouse_state](#)

3.12.1.5 on_click

void(* [UIButton::on_click](#)) ([UIButton](#) *self)

3.12.1.6 text

char [UIButton::text](#) [[UITEXT_MAX_LENGTH](#)+1]

3.12.1.7 text_color

[Color](#) [UIButton::text_color](#)

3.12.1.8 text_position

```
SDL_Point UIButton::text_position
```

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.h](#)

3.13 UICheckbox Struct Reference

```
#include <ui_element.h>
```

Public Attributes

- [UIElement](#) base
- bool [checked](#)
- [Color](#) [checked_color](#)
- [Color](#) [unchecked_color](#)
- Uint32 [corner_radius](#)
- [MouseState](#) [mouse_state](#)
- void(* [on_checked_changed](#))(UICheckbox *self, bool [checked](#))

3.13.1 Member Data Documentation

3.13.1.1 base

```
UIElement UICheckbox::base
```

3.13.1.2 checked

```
bool UICheckbox::checked
```

3.13.1.3 checked_color

```
Color UICheckbox::checked_color
```

3.13.1.4 corner_radius

```
Uint32 UICheckbox::corner_radius
```

3.13.1.5 mouse_state

```
MouseState UICheckbox::mouse_state
```

3.13.1.6 on_checked_changed

```
void(* UICheckbox::on_checked_changed) (UICheckbox *self, bool checked)
```

3.13.1.7 unchecked_color

```
Color UICheckbox::unchecked_color
```

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.h](#)

3.14 UIConstraint Struct Reference

```
#include <ui_constraint.h>
```

Public Attributes

- double [value](#)
- [ConstraintType](#) [constraint_type](#)
- void(* [recalculate](#))(void *self)

3.14.1 Member Data Documentation

3.14.1.1 constraint_type

```
ConstraintType UIConstraint::constraint_type
```

3.14.1.2 recalculate

```
void(* UIConstraint::recalculate) (void *self)
```

3.14.1.3 value

```
double UIConstraint::value
```

The documentation for this struct was generated from the following file:

- [src/ui/ui_constraint/ui_constraint.h](#)

3.15 UIConstraints Struct Reference

```
#include <ui_constraint.h>
```

Public Attributes

- [UIConstraint x](#)
- [UIConstraint y](#)
- [UIConstraint width](#)
- [UIConstraint height](#)

3.15.1 Member Data Documentation

3.15.1.1 height

[UIConstraint](#) UIConstraints::height

3.15.1.2 width

[UIConstraint](#) UIConstraints::width

3.15.1.3 x

[UIConstraint](#) UIConstraints::x

3.15.1.4 y

[UIConstraint](#) UIConstraints::y

The documentation for this struct was generated from the following file:

- [src/ui/ui_constraint/ui_constraint.h](#)

3.16 UIContainer Struct Reference

```
#include <ui_element.h>
```

Public Attributes

- [UIElement base](#)
- [Vector](#) * children
- void(* [on_size_changed](#))(UIContainer *self, SDL_Point size)

3.16.1 Member Data Documentation

3.16.1.1 base

[UIElement](#) UIContainer::base

3.16.1.2 children

```
Vector* UIContainer::children
```

3.16.1.3 on_size_changed

```
void(* UIContainer::on_size_changed) (UIContainer *self, SDL_Point size)
```

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.h](#)

3.17 UIData Struct Reference

```
#include <ui.h>
```

Public Attributes

- [UIContainer](#) * [main_container](#)
- char [text_input](#) [SDL_TEXTINPUTEVENT_TEXT_SIZE]
- bool [backspace_pressed](#)
- bool [mouse_captured](#)
- [UISplitButton](#) * [expanded_splitbutton](#)

3.17.1 Member Data Documentation

3.17.1.1 backspace_pressed

```
bool UIData::backspace_pressed
```

3.17.1.2 expanded_splitbutton

```
UISplitButton* UIData::expanded_splitbutton
```

3.17.1.3 main_container

```
UIContainer* UIData::main_container
```

3.17.1.4 mouse_captured

```
bool UIData::mouse_captured
```

3.17.1.5 text_input

```
char UIData::text_input[SDL_TEXTINPUTEVENT_TEXT_SIZE]
```

The documentation for this struct was generated from the following file:

- [src/ui/ui.h](#)

3.18 UIDropdownList Struct Reference

```
#include <ui_element.h>
```

Public Attributes

- [UIElement](#) base
- [Vector](#) * items
- Uint32 selected_item
- bool expanded
- [Color](#) color
- [Color](#) text_color
- Uint32 corner_radius
- void(* on_selection_changed)(UIDropdownList *self, Sint32 index)

3.18.1 Member Data Documentation

3.18.1.1 base

```
UIElement UIDropdownList::base
```

3.18.1.2 color

```
Color UIDropdownList::color
```

3.18.1.3 corner_radius

```
Uint32 UIDropdownList::corner_radius
```

3.18.1.4 expanded

```
bool UIDropdownList::expanded
```

3.18.1.5 items

```
Vector* UIDropdownList::items
```

3.18.1.6 on_selection_changed

```
void(* UIDropdownList::on_selection_changed) (UIDropdownList *self, Sint32 index)
```

3.18.1.7 selected_item

```
Uint32 UIDropdownList::selected_item
```

3.18.1.8 text_color

```
Color UIDropdownList::text_color
```

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.h](#)

3.19 UIElement Struct Reference

```
#include <ui_element.h>
```

Public Attributes

- [UIElement](#) * [parent](#)
- [UIConstraints](#) [constraints](#)
- [SDL_Point](#) [position](#)
- [SDL_Point](#) [size](#)
- void(* [update](#))(UIElement *self)
- void(* [recalculate](#))(UIElement *sibling, UIElement *self)
- void(* [render](#))(UIElement *self)
- void(* [destroy](#))(UIElement *self)

3.19.1 Member Data Documentation

3.19.1.1 constraints

```
UIConstraints UIElement::constraints
```

3.19.1.2 destroy

```
void(* UIElement::destroy) (UIElement *self)
```

3.19.1.3 parent

```
UIElement* UIElement::parent
```


3.19.1.4 position

```
SDL_Point UIElement::position
```

3.19.1.5 recalculate

```
void(* UIElement::recalculate) (UIElement *sibling, UIElement *self)
```

3.19.1.6 render

```
void(* UIElement::render) (UIElement *self)
```

3.19.1.7 size

```
SDL_Point UIElement::size
```

3.19.1.8 update

```
void(* UIElement::update) (UIElement *self)
```

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.h](#)

3.20 UIImageButton Struct Reference

```
#include <ui_element.h>
```

Public Attributes

- [UIElement](#) base
- [Texture](#) * texture
- [MouseState](#) mouse_state
- void(* [on_click](#))(UIImageButton *self)

3.20.1 Member Data Documentation

3.20.1.1 base

```
UIElement UIImageButton::base
```

3.20.1.2 mouse_state

`MouseState UIImageButton::mouse_state`

3.20.1.3 on_click

`void(* UIImageButton::on_click) (UIImageButton *self)`

3.20.1.4 texture

`Texture* UIImageButton::texture`

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.h](#)

3.21 UILabel Struct Reference

```
#include <ui_element.h>
```

Public Attributes

- [UIElement](#) base
- char [text](#) [UITEXT_MAX_LENGTH+1]
- [Color](#) color

3.21.1 Member Data Documentation

3.21.1.1 base

`UIElement UILabel::base`

3.21.1.2 color

`Color UILabel::color`

3.21.1.3 text

`char UILabel::text [UITEXT_MAX_LENGTH+1]`

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.h](#)

3.22 UIPanel Struct Reference

```
#include <ui_element.h>
```

Public Attributes

- [UIElement](#) base
- [Color](#) color
- [Color](#) border_color
- [Uint32](#) border_width
- [Uint32](#) corner_radius

3.22.1 Member Data Documentation

3.22.1.1 base

```
UIElement UIPanel::base
```

3.22.1.2 border_color

```
Color UIPanel::border_color
```

3.22.1.3 border_width

```
Uint32 UIPanel::border_width
```

3.22.1.4 color

```
Color UIPanel::color
```

3.22.1.5 corner_radius

```
Uint32 UIPanel::corner_radius
```

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.h](#)

3.23 UISlider Struct Reference

```
#include <ui_element.h>
```

Public Attributes

- [UIElement](#) base
- double [value](#)
- [Color](#) color
- [Color](#) slider_color
- [UInt32](#) thickness
- [UInt32](#) corner_radius
- [MouseState](#) mouse_state
- void(* [on_value_changed](#))(UISlider *self, double [value](#))

3.23.1 Member Data Documentation

3.23.1.1 base

[UIElement](#) UISlider::base

3.23.1.2 color

[Color](#) UISlider::color

3.23.1.3 corner_radius

[UInt32](#) UISlider::corner_radius

3.23.1.4 mouse_state

[MouseState](#) UISlider::mouse_state

3.23.1.5 on_value_changed

void(* UISlider::on_value_changed) ([UISlider](#) *self, double [value](#))

3.23.1.6 slider_color

[Color](#) UISlider::slider_color

3.23.1.7 thickness

[UInt32](#) UISlider::thickness

3.23.1.8 value

```
double UISlider::value
```

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.h](#)

3.24 UISplitButton Struct Reference

```
#include <ui_element.h>
```

Public Attributes

- [UIElement](#) base
- [Vector](#) * items
- bool expanded
- [Color](#) color
- [Color](#) text_color
- Uint32 corner_radius
- void(* on_item_clicked)(UISplitButton *self, Sint32 index)
- bool auto_dropdown

3.24.1 Member Data Documentation

3.24.1.1 auto_dropdown

```
bool UISplitButton::auto_dropdown
```

3.24.1.2 base

```
UIElement UISplitButton::base
```

3.24.1.3 color

```
Color UISplitButton::color
```

3.24.1.4 corner_radius

```
Uint32 UISplitButton::corner_radius
```

3.24.1.5 expanded

```
bool UISplitButton::expanded
```

3.24.1.6 items

`Vector*` `UISplitButton::items`

3.24.1.7 on_item_clicked

`void(* UISplitButton::on_item_clicked) (UISplitButton *self, Sint32 index)`

3.24.1.8 text_color

`Color` `UISplitButton::text_color`

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.h](#)

3.25 UITextbox Struct Reference

```
#include <ui_element.h>
```

Public Attributes

- [UIElement](#) `base`
- `char` `text` [[UITEXT_MAX_LENGTH](#)+1]
- [Color](#) `color`
- [Color](#) `text_color`
- [Uint32](#) `corner_radius`
- `bool` `focused`
- [MouseState](#) `mouse_state`
- `void(* on_text_changed)(UITextbox *self, const char *text)`

3.25.1 Member Data Documentation

3.25.1.1 base

[UIElement](#) `UITextbox::base`

3.25.1.2 color

[Color](#) `UITextbox::color`

3.25.1.3 corner_radius

[Uint32](#) `UITextbox::corner_radius`

3.25.1.4 focused

```
bool UITextbox::focused
```

3.25.1.5 mouse_state

```
MouseState UITextbox::mouse_state
```

3.25.1.6 on_text_changed

```
void(* UITextbox::on_text_changed) (UITextbox *self, const char *text)
```

3.25.1.7 text

```
char UITextbox::text[UITEXT_MAX_LENGTH+1]
```

3.25.1.8 text_color

```
Color UITextbox::text_color
```

The documentation for this struct was generated from the following file:

- [src/ui/ui_element/ui_element.h](#)

3.26 Vector Struct Reference

```
#include <vector.h>
```

Public Attributes

- `size_t` [capacity](#)
- `size_t` [size](#)
- `void **` [data](#)

3.26.1 Member Data Documentation

3.26.1.1 capacity

```
size_t Vector::capacity
```

3.26.1.2 data

```
void** Vector::data
```

3.26.1.3 size

```
size_t Vector::size
```

The documentation for this struct was generated from the following file:

- [src/utils/vector/vector.h](#)

3.27 Vector2 Struct Reference

```
#include <vector2.h>
```

Public Attributes

- double [x](#)
- double [y](#)

3.27.1 Member Data Documentation

3.27.1.1 x

```
double Vector2::x
```

3.27.1.2 y

```
double Vector2::y
```

The documentation for this struct was generated from the following file:

- [src/geometry/vector2/vector2.h](#)

3.28 Window Struct Reference

```
#include <window.h>
```


Public Attributes

- `SDL_Window *` [window](#)
- `SDL_Renderer *` [renderer](#)
- `InputData` [input_data](#)
- `UIData` [ui_data](#)
- `bool` [close_requested](#)

3.28.1 Member Data Documentation

3.28.1.1 `close_requested`

```
bool Window::close_requested
```

3.28.1.2 `input_data`

```
InputData Window::input_data
```

3.28.1.3 `renderer`

```
SDL_Renderer* Window::renderer
```

3.28.1.4 `ui_data`

```
UIData Window::ui_data
```

3.28.1.5 `window`

```
SDL_Window* Window::window
```

The documentation for this struct was generated from the following file:

- `src/window/`[window.h](#)

Chapter 4

File Documentation

4.1 src/app/app.c File Reference

```
#include "app.h"
#include "../window/window.h"
#include "../renderer/renderer.h"
#include "../input/input.h"
#include "../ui/ui.h"
```

Functions

- void [app_init](#) ()
- void [app_update](#) ()
- void [app_render](#) ()
- void [app_request_close](#) ()
- void [app_close](#) ()
- void [app_set_target_fps](#) (Uint32 fps)
- void [app_set_target](#) ([Window](#) *window)
- [Vector](#) * [app_get_windows](#) ()
- double [app_get_time](#) ()
- double [app_get_delta_time](#) ()
- void [_app_add_window](#) ([Window](#) *window)

Variables

- [AppData](#) [app_data](#)

4.1.1 Function Documentation

4.1.1.1 _app_add_window()

```
void _app_add_window (  
    Window * window )
```

4.1.1.2 app_close()

```
void app_close ( )
```

4.1.1.3 app_get_delta_time()

```
double app_get_delta_time ( )
```

4.1.1.4 app_get_time()

```
double app_get_time ( )
```

4.1.1.5 app_get_windows()

```
Vector * app_get_windows ( )
```

4.1.1.6 app_init()

```
void app_init ( )
```

4.1.1.7 app_render()

```
void app_render ( )
```

4.1.1.8 app_request_close()

```
void app_request_close ( )
```

4.1.1.9 app_set_target()

```
void app_set_target (
    Window * window )
```

4.1.1.10 app_set_target_fps()

```
void app_set_target_fps (
    Uint32 fps )
```

4.1.1.11 app_update()

```
void app_update ( )
```

4.1.2 Variable Documentation

4.1.2.1 app_data

[AppData](#) app_data

4.2 src/app/app.h File Reference

```
#include "../window/window.h"
#include "../utils/vector/vector.h"
```

Classes

- struct [AppData](#)

Typedefs

- typedef struct [AppData](#) [AppData](#)

Functions

- void [app_init](#) ()
- void [app_update](#) ()
- void [app_render](#) ()
- void [app_request_close](#) ()
- void [app_close](#) ()
- void [app_set_target_fps](#) (Uint32 fps)
- void [app_set_target](#) ([Window](#) *window)
- [Vector](#) * [app_get_windows](#) ()
- double [app_get_time](#) ()
- double [app_get_delta_time](#) ()
- void [_app_add_window](#) ([Window](#) *window)

4.2.1 Typedef Documentation

4.2.1.1 AppData

```
typedef struct AppData AppData
```

4.2.2 Function Documentation

4.2.2.1 _app_add_window()

```
void _app_add_window (  
    Window * window )
```

4.2.2.2 app_close()

```
void app_close ( )
```

4.2.2.3 app_get_delta_time()

```
double app_get_delta_time ( )
```

4.2.2.4 app_get_time()

```
double app_get_time ( )
```

4.2.2.5 app_get_windows()

```
Vector * app_get_windows ( )
```

4.2.2.6 app_init()

```
void app_init ( )
```

4.2.2.7 app_render()

```
void app_render ( )
```

4.2.2.8 app_request_close()

```
void app_request_close ( )
```

4.2.2.9 app_set_target()

```
void app_set_target (
    Window * window )
```

4.2.2.10 app_set_target_fps()

```
void app_set_target_fps (
    Uint32 fps )
```

4.2.2.11 app_update()

```
void app_update ( )
```

4.3 app.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #ifdef _WIN32
00004     #include <SDL.h>
00005 #elif defined(__unix__) || defined(__linux__)
00006     #include <SDL2/SDL.h>
00007 #endif
00008
00009 #include "../window/window.h"
00010 #include "../utils/vector/vector.h"
00011
00012 typedef struct AppData
00013 {
00014     Vector* windows;
00015     Uint32 target_frame_time;
00016     Uint32 last_frame_start;
00017     Uint32 frame_start;
00018     double delta_time;
00019 } AppData;
00020
00021 void app_init();
00022 void app_update();
00023 void app_render();
00024 void app_request_close();
00025 void app_close();
00026 void app_set_target_fps(Uint32 fps);
00027
00028 void app_set_target(Window* window);
00029 Vector* app_get_windows();
00030 double app_get_time();
00031 double app_get_delta_time();
00032
00033 //internal functions
00034 void _app_add_window(Window* window);
```

4.4 src/color/color.c File Reference

```
#include "color.h"
```

Functions

- [Color color_from_hex](#) (int hex)
- [Color color_from_rgb](#) (int r, int g, int b)
- [Color color_from_rgba](#) (int r, int g, int b, int a)
- [Color color_from_hsv](#) (double h, double s, double v)
- [Color color_from_grayscale](#) (int value)
- [Color color_fade](#) (Color color, double fade)
- [Color color_shift](#) (Color color, int shift)
- [Color color_clever_shift](#) (Color color, int shift)

4.4.1 Function Documentation

4.4.1.1 color_clever_shift()

```
Color color_clever_shift (
    Color color,
    int shift )
```

4.4.1.2 color_fade()

```
Color color_fade (
    Color color,
    double fade )
```

4.4.1.3 color_from_grayscale()

```
Color color_from_grayscale (
    int value )
```

4.4.1.4 color_from_hex()

```
Color color_from_hex (
    int hex )
```

4.4.1.5 color_from_hsv()

```
Color color_from_hsv (
    double h,
    double s,
    double v )
```

4.4.1.6 color_from_rgb()

```
Color color_from_rgb (
    int r,
    int g,
    int b )
```

4.4.1.7 color_from_rgba()

```
Color color_from_rgba (
    int r,
    int g,
    int b,
    int a )
```

4.4.1.8 color_shift()

```
Color color_shift (
    Color color,
    int shift )
```


4.5 src/color/color.h File Reference

Macros

- `#define WHITE (Color) { 255, 255, 255, 255 }`
- `#define BLACK (Color) { 0, 0, 0, 255 }`
- `#define GRAY (Color) { 128, 128, 128, 255 }`
- `#define DARK_GRAY (Color) { 40, 40, 40, 255 }`
- `#define RED (Color) { 255, 0, 0, 255 }`
- `#define GREEN (Color) { 0, 255, 0, 255 }`
- `#define BLUE (Color) { 0, 0, 255, 255 }`
- `#define YELLOW (Color) { 255, 255, 0, 255 }`
- `#define MAGENTA (Color) { 255, 0, 255, 255 }`
- `#define CYAN (Color) { 0, 255, 255, 255 }`
- `#define TRANSPARENT (Color) { 0, 0, 0, 0 }`

Typedefs

- `typedef SDL_Color Color`

Functions

- `Color color_from_hex (int hex)`
- `Color color_from_rgb (int r, int g, int b)`
- `Color color_from_rgba (int r, int g, int b, int a)`
- `Color color_from_hsv (double h, double s, double v)`
- `Color color_from_grayscale (int value)`
- `Color color_fade (Color color, double fade)`
- `Color color_shift (Color color, int shift)`
- `Color color_clever_shift (Color color, int shift)`

4.5.1 Macro Definition Documentation

4.5.1.1 BLACK

```
#define BLACK (Color) { 0, 0, 0, 255 }
```

4.5.1.2 BLUE

```
#define BLUE (Color) { 0, 0, 255, 255 }
```

4.5.1.3 CYAN

```
#define CYAN (Color) { 0, 255, 255, 255 }
```

4.5.1.4 DARK_GRAY

```
#define DARK_GRAY (Color) { 40, 40, 40, 255 }
```

4.5.1.5 GRAY

```
#define GRAY (Color) { 128, 128, 128, 255 }
```

4.5.1.6 GREEN

```
#define GREEN (Color) { 0, 255, 0, 255 }
```

4.5.1.7 MAGENTA

```
#define MAGENTA (Color) { 255, 0, 255, 255 }
```

4.5.1.8 RED

```
#define RED (Color) { 255, 0, 0, 255 }
```

4.5.1.9 TRANSPARENT

```
#define TRANSPARENT (Color) { 0, 0, 0, 0 }
```

4.5.1.10 WHITE

```
#define WHITE (Color) { 255, 255, 255, 255 }
```

4.5.1.11 YELLOW

```
#define YELLOW (Color) { 255, 255, 0, 255 }
```

4.5.2 Typedef Documentation

4.5.2.1 Color

```
typedef SDL_Color Color
```

4.5.3 Function Documentation

4.5.3.1 color_clever_shift()

```
Color color_clever_shift (
    Color color,
    int shift )
```

4.5.3.2 color_fade()

```
Color color_fade (
    Color color,
    double fade )
```

4.5.3.3 color_from_grayscale()

```
Color color_from_grayscale (
    int value )
```

4.5.3.4 color_from_hex()

```
Color color_from_hex (
    int hex )
```

4.5.3.5 color_from_hsv()

```
Color color_from_hsv (
    double h,
    double s,
    double v )
```

4.5.3.6 color_from_rgb()

```
Color color_from_rgb (
    int r,
    int g,
    int b )
```

4.5.3.7 color_from_rgba()

```
Color color_from_rgba (
    int r,
    int g,
    int b,
    int a )
```

4.5.3.8 color_shift()

```
Color color_shift (
    Color color,
    int shift )
```

4.6 color.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #ifndef _WIN32
00004     #include <SDL.h>
00005 #elif defined(__unix__) || defined(__linux__)
00006     #include <SDL2/SDL.h>
00007 #endif
00008
00009 typedef SDL_Color Color;
00010
00011 #define WHITE (Color) { 255, 255, 255, 255 }
00012 #define BLACK (Color) { 0, 0, 0, 255 }
00013 #define GRAY (Color) { 128, 128, 128, 255 }
00014 #define DARK_GRAY (Color) { 40, 40, 40, 255 }
00015 #define RED (Color) { 255, 0, 0, 255 }
00016 #define GREEN (Color) { 0, 255, 0, 255 }
00017 #define BLUE (Color) { 0, 0, 255, 255 }
00018 #define YELLOW (Color) { 255, 255, 0, 255 }
00019 #define MAGENTA (Color) { 255, 0, 255, 255 }
00020 #define CYAN (Color) { 0, 255, 255, 255 }
00021 #define TRANSPARENT (Color) { 0, 0, 0, 0 }
00022
00023 Color color_from_hex(int hex);
00024 Color color_from_rgb(int r, int g, int b);
00025 Color color_from_rgba(int r, int g, int b, int a);
00026 Color color_from_hsv(double h, double s, double v);
00027 Color color_from_grayscale(int value);
00028 Color color_fade(Color color, double fade);
00029 Color color_shift(Color color, int shift);
00030 Color color_clever_shift(Color color, int shift);
```

4.7 src/font/font.c File Reference

```
#include "font.h"
#include "../utils/vector/vector.h"
```

Functions

- [Font * font_load](#) (const char *path, int size)
- [void _font_init](#) ()
- [void _font_close](#) ()

Variables

- [Vector * fonts](#)

4.7.1 Function Documentation

4.7.1.1 `_font_close()`

```
void _font_close ( )
```

4.7.1.2 `_font_init()`

```
void _font_init ( )
```

4.7.1.3 `font_load()`

```
Font * font_load (
    const char * path,
    int size )
```

4.7.2 Variable Documentation

4.7.2.1 `fonts`

```
Vector* fonts
```

4.8 src/font/font.h File Reference

Classes

- struct [Font](#)

Typedefs

- typedef struct [Font](#) [Font](#)

Functions

- [Font](#) * [font_load](#) (const char *path, int size)
- void [_font_init](#) ()
- void [_font_close](#) ()

4.8.1 Typedef Documentation

4.8.1.1 `Font`

```
typedef struct Font Font
```

4.8.2 Function Documentation

4.8.2.1 `_font_close()`

```
void _font_close ( )
```

4.8.2.2 `_font_init()`

```
void _font_init ( )
```

4.8.2.3 `font_load()`

```
Font * font_load (
    const char * path,
    int size )
```

4.9 font.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #ifdef _WIN32
00004     #include <SDL_ttf.h>
00005 #elif defined(__unix__) || defined(__linux__)
00006     #include <SDL2/SDL_ttf.h>
00007 #endif
00008
00009 typedef struct Font
00010 {
00011     TTF_Font* font;
00012     int size;
00013 } Font;
00014
00015 Font* font_load(const char* path, int size);
00016
00017 //internal functions
00018 void _font_init();
00019 void _font_close();
```

4.10 src/geometry/coordinate_system/coordinate_system.c File Reference

```
#include "coordinate_system.h"
#include "../renderer/renderer.h"
#include "../utils/math/math.h"
```

Functions

- `CoordinateSystem * coordinate_system_create (Vector2 position, Vector2 size, Vector2 origin)`
Creates a coordinate system.
- `void coordinate_system_destroy (CoordinateSystem *cs)`
Destroys a coordinate system.
- `Vector2 screen_to_coordinates (CoordinateSystem *cs, Vector2 point)`
Translates a point from the screen to the coordinate system.
- `Vector2 coordinates_to_screen (CoordinateSystem *cs, Vector2 point)`
Translates a point from the coordinate system to the screen.
- `bool coordinate_system_is_hovered (CoordinateSystem *cs, Vector2 point)`
Returns whether the coordinate system is hovered by the point.
- `IShape * coordinate_system_get_hovered_shape (CoordinateSystem *cs, Vector2 point)`
Returns the shape hovered by the point.
- `void coordinate_system_translate (CoordinateSystem *cs, Vector2 translation)`
Translates the coordinate system.
- `void coordinate_system_zoom (CoordinateSystem *cs, double zoom)`
Zooms into the coordinate system.
- `void coordinate_system_update_dimensions (CoordinateSystem *cs, Vector2 position, Vector2 size)`
Updates the dimensions of the coordinate system.
- `void coordinate_system_draw (CoordinateSystem *cs)`
Draws the coordinate system.

4.10.1 Function Documentation

4.10.1.1 coordinate_system_create()

```
CoordinateSystem * coordinate_system_create (
    Vector2 position,
    Vector2 size,
    Vector2 origin )
```

Creates a coordinate system.

Parameters

<i>position</i>	The position of the coordinate system in the screen
<i>size</i>	The size of the coordinate system (in pixels)
<i>origin</i>	The origin of the coordinate system (relative to the coordinate system (normalized))

Returns

CoordinateSystem* The created coordinate system

4.10.1.2 coordinate_system_destroy()

```
void coordinate_system_destroy (
    CoordinateSystem * cs )
```

Destroys a coordinate system.

Parameters

<i>cs</i>	The coordinate system to destroy
-----------	----------------------------------

4.10.1.3 coordinate_system_draw()

```
void coordinate_system_draw (
    CoordinateSystem * cs )
```

Draws the coordinate system.

Parameters

<i>cs</i>	The coordinate system to draw
-----------	-------------------------------

4.10.1.4 coordinate_system_get_hovered_shape()

```
IShape * coordinate_system_get_hovered_shape (
    CoordinateSystem * cs,
    Vector2 point )
```

Returns the shape hovered by the point.

Parameters

<i>cs</i>	The coordinate system to check
<i>point</i>	The point to check

Returns

*IShape** The hovered shape (NULL if none)

4.10.1.5 coordinate_system_is_hovered()

```
bool coordinate_system_is_hovered (
    CoordinateSystem * cs,
    Vector2 point )
```

Returns whether the coordinate system is hovered by the point.

Parameters

<i>cs</i>	The coordinate system to check
<i>point</i>	The point to check

4.10.1.6 coordinate_system_translate()

```
void coordinate_system_translate (
    CoordinateSystem * cs,
    Vector2 translation )
```

Translates the coordinate system.

Parameters

<i>cs</i>	The coordinate system to translate
<i>translation</i>	The translation vector (in pixels)

4.10.1.7 coordinate_system_update_dimensions()

```
void coordinate_system_update_dimensions (
    CoordinateSystem * cs,
    Vector2 position,
    Vector2 size )
```

Updates the dimensions of the coordinate system.

Parameters

<i>cs</i>	The coordinate system to update
<i>position</i>	The new position
<i>size</i>	The new size

4.10.1.8 coordinate_system_zoom()

```
void coordinate_system_zoom (
    CoordinateSystem * cs,
    double zoom )
```

Zooms into the coordinate system.

Parameters

<i>cs</i>	The coordinate system to zoom into
<i>zoom</i>	The zoom factor

4.10.1.9 coordinates_to_screen()

```
Vector2 coordinates_to_screen (
    CoordinateSystem * cs,
    Vector2 point )
```

Translates a point from the coordinate system to the screen.

Parameters

<i>cs</i>	The coordinate system to translate the point from
<i>point</i>	The point to translate

Returns

[Vector2](#) The translated point

4.10.1.10 screen_to_coordinates()

```
Vector2 screen_to_coordinates (
    CoordinateSystem * cs,
    Vector2 point )
```

Translates a point from the screen to the coordinate system.

Parameters

<i>cs</i>	The coordinate system to translate the point to
<i>point</i>	The point to translate

Returns

[Vector2](#) The translated point

4.11 src/geometry/coordinate_system/coordinate_system.h File Reference

```
#include "../shape/shape.h"
#include "../vector2/vector2.h"
#include "../../texture/texture.h"
#include "../../utils/vector/vector.h"
```

Classes

- struct [CoordinateSystem](#)

Macros

- #define [INITIAL_ZOOM](#) 20

Typedefs

- typedef struct [CoordinateSystem](#) [CoordinateSystem](#)

Functions

- [CoordinateSystem](#) * [coordinate_system_create](#) ([Vector2](#) position, [Vector2](#) size, [Vector2](#) origin)
Creates a coordinate system.
- void [coordinate_system_destroy](#) ([CoordinateSystem](#) *cs)
Destroys a coordinate system.
- [Vector2](#) [screen_to_coordinates](#) ([CoordinateSystem](#) *cs, [Vector2](#) point)
Translates a point from the screen to the coordinate system.
- [Vector2](#) [coordinates_to_screen](#) ([CoordinateSystem](#) *cs, [Vector2](#) point)
Translates a point from the coordinate system to the screen.
- bool [coordinate_system_is_hovered](#) ([CoordinateSystem](#) *cs, [Vector2](#) point)
Returns whether the coordinate system is hovered by the point.
- [IShape](#) * [coordinate_system_get_hovered_shape](#) ([CoordinateSystem](#) *cs, [Vector2](#) point)
Returns the shape hovered by the point.
- void [coordinate_system_translate](#) ([CoordinateSystem](#) *cs, [Vector2](#) translation)
Translates the coordinate system.
- void [coordinate_system_zoom](#) ([CoordinateSystem](#) *cs, double zoom)
Zooms into the coordinate system.
- void [coordinate_system_update_dimensions](#) ([CoordinateSystem](#) *cs, [Vector2](#) position, [Vector2](#) size)
Updates the dimensions of the coordinate system.
- void [coordinate_system_draw](#) ([CoordinateSystem](#) *cs)
Draws the coordinate system.

4.11.1 Macro Definition Documentation

4.11.1.1 INITIAL_ZOOM

```
#define INITIAL_ZOOM 20
```

4.11.2 Typedef Documentation

4.11.2.1 CoordinateSystem

```
typedef struct CoordinateSystem CoordinateSystem
```

4.11.3 Function Documentation

4.11.3.1 coordinate_system_create()

```
CoordinateSystem * coordinate\_system\_create (  
    Vector2 position,  
    Vector2 size,  
    Vector2 origin )
```

Creates a coordinate system.

Parameters

<i>position</i>	The position of the coordinate system in the screen
<i>size</i>	The size of the coordinate system (in pixels)
<i>origin</i>	The origin of the coordinate system (relative to the coordinate system (normalized))

Returns

CoordinateSystem* The created coordinate system

4.11.3.2 coordinate_system_destroy()

```
void coordinate_system_destroy (
    CoordinateSystem * cs )
```

Destroys a coordinate system.

Parameters

<i>cs</i>	The coordinate system to destroy
-----------	----------------------------------

4.11.3.3 coordinate_system_draw()

```
void coordinate_system_draw (
    CoordinateSystem * cs )
```

Draws the coordinate system.

Parameters

<i>cs</i>	The coordinate system to draw
-----------	-------------------------------

4.11.3.4 coordinate_system_get_hovered_shape()

```
IShape * coordinate_system_get_hovered_shape (
    CoordinateSystem * cs,
    Vector2 point )
```

Returns the shape hovered by the point.

Parameters

<i>cs</i>	The coordinate system to check
<i>point</i>	The point to check

Returns

IShape* The hovered shape (NULL if none)

4.11.3.5 coordinate_system_is_hovered()

```
bool coordinate_system_is_hovered (
    CoordinateSystem * cs,
    Vector2 point )
```

Returns whether the coordinate system is hovered by the point.

Parameters

<i>cs</i>	The coordinate system to check
<i>point</i>	The point to check

4.11.3.6 coordinate_system_translate()

```
void coordinate_system_translate (
    CoordinateSystem * cs,
    Vector2 translation )
```

Translates the coordinate system.

Parameters

<i>cs</i>	The coordinate system to translate
<i>translation</i>	The translation vector (in pixels)

4.11.3.7 coordinate_system_update_dimensions()

```
void coordinate_system_update_dimensions (
    CoordinateSystem * cs,
    Vector2 position,
    Vector2 size )
```

Updates the dimensions of the coordinate system.

Parameters

<i>cs</i>	The coordinate system to update
<i>position</i>	The new position
<i>size</i>	The new size

4.11.3.8 coordinate_system_zoom()

```
void coordinate_system_zoom (
    CoordinateSystem * cs,
    double zoom )
```

Zooms into the coordinate system.

Parameters

<i>cs</i>	The coordinate system to zoom into
<i>zoom</i>	The zoom factor

4.11.3.9 coordinates_to_screen()

```
Vector2 coordinates_to_screen (
    CoordinateSystem * cs,
    Vector2 point )
```

Translates a point from the coordinate system to the screen.

Parameters

<i>cs</i>	The coordinate system to translate the point from
<i>point</i>	The point to translate

Returns

Vector2 The translated point

4.11.3.10 screen_to_coordinates()

```
Vector2 screen_to_coordinates (
    CoordinateSystem * cs,
    Vector2 point )
```

Translates a point from the screen to the coordinate system.

Parameters

<i>cs</i>	The coordinate system to translate the point to
<i>point</i>	The point to translate

Returns

[Vector2](#) The translated point

4.12 coordinate_system.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include "../shape/shape.h"
00004 #include "../vector2/vector2.h"
00005 #include "../../texture/texture.h"
00006 #include "../../utils/vector/vector.h"
00007
00008 #define INITIAL_ZOOM 20
00009
00010 typedef struct CoordinateSystem
00011 {
00012     Vector2 position;
00013     Vector2 size;
00014     Vector2 origin;
00015     double zoom;
00016
00017     Vector* shapes;
00018 } CoordinateSystem;
00019
00028 CoordinateSystem* coordinate_system_create(Vector2 position, Vector2 size, Vector2 origin);
00034 void coordinate_system_destroy(CoordinateSystem* cs);
00035
00043 Vector2 screen_to_coordinates(CoordinateSystem* cs, Vector2 point);
00051 Vector2 coordinates_to_screen(CoordinateSystem* cs, Vector2 point);
00052
00059 bool coordinate_system_is_hovered(CoordinateSystem* cs, Vector2 point);
00067 IShape* coordinate_system_get_hovered_shape(CoordinateSystem* cs, Vector2 point);
00068
00075 void coordinate_system_translate(CoordinateSystem* cs, Vector2 translation);
00082 void coordinate_system_zoom(CoordinateSystem* cs, double zoom);
00090 void coordinate_system_update_dimensions(CoordinateSystem* cs, Vector2 position, Vector2 size);
00096 void coordinate_system_draw(CoordinateSystem* cs);
```

4.13 src/geometry/shape/shape.c File Reference

```
#include "shape.h"
#include "../coordinate_system/coordinate_system.h"
#include "../../renderer/renderer.h"
#include <math.h>
```

Functions

- [Point * point_create](#) (CoordinateSystem *cs, Vector2 coordinates)
Creates a point in the coordinate system.
- [Line * line_create](#) (CoordinateSystem *cs, Point *p1, Point *p2)
Creates a line in the coordinate system.
- [Circle * circle_create](#) (CoordinateSystem *cs, Point *center, Point *perimeter_point)
Creates a circle in the coordinate system.

4.13.1 Function Documentation

4.13.1.1 circle_create()

```
Circle * circle_create (
    CoordinateSystem * cs,
    Point * center,
    Point * perimeter_point )
```

Creates a circle in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the circle in
<i>center</i>	The center of the circle
<i>perimeter_point</i>	A point on the perimeter of the circle (has to be different from center)

Returns

Circle* The created circle

4.13.1.2 line_create()

```
Line * line_create (
    CoordinateSystem * cs,
    Point * p1,
    Point * p2 )
```

Creates a line in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the line in
<i>p1</i>	A point of the line
<i>p2</i>	Another point of the line (has to be different from p1)

Returns

Line* The created line

4.13.1.3 point_create()

```
Point * point_create (
    CoordinateSystem * cs,
    Vector2 coordinates )
```

Creates a point in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the point in
<i>coordinates</i>	The coordinates of the point

Returns

Point* The created point

4.14 src/geometry/shape/shape.h File Reference

```
#include <stdbool.h>
#include "../vector2/vector2.h"
```

Classes

- struct [IShape](#)
- struct [Point](#)
- struct [Line](#)
- struct [Circle](#)

Macros

- #define [OVERLAP_DISTANCE](#) 5

Typedefs

- typedef struct [CoordinateSystem](#) [CoordinateSystem](#)
- typedef struct [IShape](#) [IShape](#)
- typedef struct [Point](#) [Point](#)
- typedef struct [Line](#) [Line](#)
- typedef struct [Circle](#) [Circle](#)

Functions

- [Point](#) * [point_create](#) ([CoordinateSystem](#) *cs, [Vector2](#) coordinates)
Creates a point in the coordinate system.
- [Line](#) * [line_create](#) ([CoordinateSystem](#) *cs, [Point](#) *p1, [Point](#) *p2)
Creates a line in the coordinate system.
- [Circle](#) * [circle_create](#) ([CoordinateSystem](#) *cs, [Point](#) *center, [Point](#) *perimeter_point)
Creates a circle in the coordinate system.

4.14.1 Macro Definition Documentation

4.14.1.1 OVERLAP_DISTANCE

```
#define OVERLAP_DISTANCE 5
```

4.14.2 Typedef Documentation

4.14.2.1 Circle

```
typedef struct Circle Circle
```

4.14.2.2 CoordinateSystem

```
typedef struct CoordinateSystem CoordinateSystem
```

4.14.2.3 IShape

```
typedef struct IShape IShape
```

4.14.2.4 Line

```
typedef struct Line Line
```

4.14.2.5 Point

```
typedef struct Point Point
```

4.14.3 Function Documentation

4.14.3.1 circle_create()

```
Circle * circle_create (
    CoordinateSystem * cs,
    Point * center,
    Point * perimeter_point )
```

Creates a circle in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the circle in
<i>center</i>	The center of the circle
<i>perimeter_point</i>	A point on the perimeter of the circle (has to be different from center)

Returns

Circle* The created circle

4.14.3.2 line_create()

```
Line * line_create (
    CoordinateSystem * cs,
    Point * p1,
    Point * p2 )
```

Creates a line in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the line in
<i>p1</i>	A point of the line
<i>p2</i>	Another point of the line (has to be different from p1)

Returns

Line* The created line

4.14.3.3 point_create()

```
Point * point_create (
    CoordinateSystem * cs,
    Vector2 coordinates )
```

Creates a point in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the point in
<i>coordinates</i>	The coordinates of the point

Returns

Point* The created point

4.15 shape.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include <stdbool.h>
00004
00005 #include "../vector2/vector2.h"
00006
00007 #define OVERLAP_DISTANCE 5
00008
00009 typedef struct CoordinateSystem CoordinateSystem;
00010 typedef struct IShape IShape;
00011 typedef struct IShape
00012 {
00013     void (*draw)(CoordinateSystem* cs, IShape* self);
00014     void (*translate)(CoordinateSystem* cs, IShape* self, Vector2 translation);
00015     void (*destroy)(CoordinateSystem* cs, IShape* self);
```

```

00016     bool (*overlap_point)(CoordinateSystem* cs, IShape* self, Vector2 point);
00017     bool (*is_defined_by)(IShape* self, IShape* shape);
00018 } IShape;
00019
00020 typedef struct Point
00021 {
00022     IShape base;
00023     Vector2 coordinates;
00024 } Point;
00025
00026 typedef struct Line
00027 {
00028     IShape base;
00029     Point *p1, *p2;
00030 } Line;
00031
00032 typedef struct Circle
00033 {
00034     IShape base;
00035     Point* center;
00036     Point* perimeter_point;
00037 } Circle;
00038
00046 Point* point_create(CoordinateSystem* cs, Vector2 coordinates);
00055 Line* line_create(CoordinateSystem* cs, Point* p1, Point* p2);
00064 Circle* circle_create(CoordinateSystem* cs, Point* center, Point* perimeter_point);

```

4.16 src/geometry/vector2/vector2.c File Reference

```

#include "vector2.h"
#include <math.h>

```

Functions

- [Vector2 vector2_create](#) (double x, double y)
- [Vector2 vector2_from_polar](#) (double angle, double length)
- [Vector2 vector2_from_point](#) (SDL_Point point)
- [Vector2 vector2_zero](#) ()
- [Vector2 vector2_one](#) ()
- [Vector2 vector2_up](#) ()
- [Vector2 vector2_down](#) ()
- [Vector2 vector2_left](#) ()
- [Vector2 vector2_right](#) ()
- [Vector2 vector2_add](#) (Vector2 a, Vector2 b)
- [Vector2 vector2_subtract](#) (Vector2 a, Vector2 b)
- [Vector2 vector2_scale](#) (Vector2 a, double b)
- [Vector2 vector2_negate](#) (Vector2 a)
- [Vector2 vector2_multiply](#) (Vector2 a, Vector2 b)
- [Vector2 vector2_divide](#) (Vector2 a, Vector2 b)
- [double vector2_dot](#) (Vector2 a, Vector2 b)
- [double vector2_length](#) (Vector2 a)
- [double vector2_distance](#) (Vector2 a, Vector2 b)
- [double vector2_angle](#) (Vector2 a)
- [Vector2 vector2_normalize](#) (Vector2 a)
- [Vector2 vector2_rotate90](#) (Vector2 a)
- [Vector2 vector2_rotate](#) (Vector2 a, double angle)
- [Vector2 vector2_reflect](#) (Vector2 a, Vector2 normal)

4.16.1 Function Documentation

4.16.1.1 vector2_add()

```
Vector2 vector2_add (
    Vector2 a,
    Vector2 b )
```

4.16.1.2 vector2_angle()

```
double vector2_angle (
    Vector2 a )
```

4.16.1.3 vector2_create()

```
Vector2 vector2_create (
    double x,
    double y )
```

4.16.1.4 vector2_distance()

```
double vector2_distance (
    Vector2 a,
    Vector2 b )
```

4.16.1.5 vector2_divide()

```
Vector2 vector2_divide (
    Vector2 a,
    Vector2 b )
```

4.16.1.6 vector2_dot()

```
double vector2_dot (
    Vector2 a,
    Vector2 b )
```

4.16.1.7 vector2_down()

```
Vector2 vector2_down ( )
```

4.16.1.8 vector2_from_point()

```
Vector2 vector2_from_point (
    SDL_Point point )
```

4.16.1.9 vector2_from_polar()

```
Vector2 vector2_from_polar (
    double angle,
    double length )
```

4.16.1.10 vector2_left()

```
Vector2 vector2_left ( )
```

4.16.1.11 vector2_length()

```
double vector2_length (
    Vector2 a )
```

4.16.1.12 vector2_multiply()

```
Vector2 vector2_multiply (
    Vector2 a,
    Vector2 b )
```

4.16.1.13 vector2_negate()

```
Vector2 vector2_negate (
    Vector2 a )
```

4.16.1.14 vector2_normalize()

```
Vector2 vector2_normalize (
    Vector2 a )
```

4.16.1.15 vector2_one()

```
Vector2 vector2_one ( )
```

4.16.1.16 vector2_reflect()

```
Vector2 vector2_reflect (
    Vector2 a,
    Vector2 normal )
```

4.16.1.17 vector2_right()

```
Vector2 vector2_right ( )
```

4.16.1.18 vector2_rotate()

```
Vector2 vector2_rotate (
    Vector2 a,
    double angle )
```

4.16.1.19 vector2_rotate90()

```
Vector2 vector2_rotate90 (
    Vector2 a )
```

4.16.1.20 vector2_scale()

```
Vector2 vector2_scale (
    Vector2 a,
    double b )
```

4.16.1.21 vector2_subtract()

```
Vector2 vector2_subtract (
    Vector2 a,
    Vector2 b )
```

4.16.1.22 vector2_up()

```
Vector2 vector2_up ( )
```

4.16.1.23 vector2_zero()

```
Vector2 vector2_zero ( )
```

4.17 src/geometry/vector2/vector2.h File Reference

Classes

- struct [Vector2](#)

Typedefs

- typedef struct [Vector2](#) [Vector2](#)

Functions

- [Vector2 vector2_create](#) (double x, double y)
- [Vector2 vector2_from_polar](#) (double angle, double length)
- [Vector2 vector2_from_point](#) (SDL_Point point)
- [Vector2 vector2_zero](#) ()
- [Vector2 vector2_one](#) ()
- [Vector2 vector2_up](#) ()
- [Vector2 vector2_down](#) ()
- [Vector2 vector2_left](#) ()
- [Vector2 vector2_right](#) ()
- [Vector2 vector2_add](#) (Vector2 a, Vector2 b)
- [Vector2 vector2_subtract](#) (Vector2 a, Vector2 b)
- [Vector2 vector2_scale](#) (Vector2 a, double b)
- [Vector2 vector2_negate](#) (Vector2 a)
- [Vector2 vector2_multiply](#) (Vector2 a, Vector2 b)
- [Vector2 vector2_divide](#) (Vector2 a, Vector2 b)
- double [vector2_dot](#) (Vector2 a, Vector2 b)
- double [vector2_length](#) (Vector2 a)
- double [vector2_distance](#) (Vector2 a, Vector2 b)
- double [vector2_angle](#) (Vector2 a)
- [Vector2 vector2_normalize](#) (Vector2 a)
- [Vector2 vector2_rotate90](#) (Vector2 a)
- [Vector2 vector2_rotate](#) (Vector2 a, double angle)
- [Vector2 vector2_reflect](#) (Vector2 a, Vector2 normal)

4.17.1 Typedef Documentation

4.17.1.1 Vector2

```
typedef struct Vector2 Vector2
```

4.17.2 Function Documentation

4.17.2.1 vector2_add()

```
Vector2 vector2_add (  
    Vector2 a,  
    Vector2 b )
```

4.17.2.2 vector2_angle()

```
double vector2_angle (  
    Vector2 a )
```

4.17.2.3 vector2_create()

```
Vector2 vector2_create (  
    double x,  
    double y )
```


4.17.2.4 vector2_distance()

```
double vector2_distance (
    Vector2 a,
    Vector2 b )
```

4.17.2.5 vector2_divide()

```
Vector2 vector2_divide (
    Vector2 a,
    Vector2 b )
```

4.17.2.6 vector2_dot()

```
double vector2_dot (
    Vector2 a,
    Vector2 b )
```

4.17.2.7 vector2_down()

```
Vector2 vector2_down ( )
```

4.17.2.8 vector2_from_point()

```
Vector2 vector2_from_point (
    SDL_Point point )
```

4.17.2.9 vector2_from_polar()

```
Vector2 vector2_from_polar (
    double angle,
    double length )
```

4.17.2.10 vector2_left()

```
Vector2 vector2_left ( )
```

4.17.2.11 vector2_length()

```
double vector2_length (
    Vector2 a )
```

4.17.2.12 vector2_multiply()

```
Vector2 vector2_multiply (
    Vector2 a,
    Vector2 b )
```

4.17.2.13 vector2_negate()

```
Vector2 vector2_negate (
    Vector2 a )
```

4.17.2.14 vector2_normalize()

```
Vector2 vector2_normalize (
    Vector2 a )
```

4.17.2.15 vector2_one()

```
Vector2 vector2_one ( )
```

4.17.2.16 vector2_reflect()

```
Vector2 vector2_reflect (
    Vector2 a,
    Vector2 normal )
```

4.17.2.17 vector2_right()

```
Vector2 vector2_right ( )
```

4.17.2.18 vector2_rotate()

```
Vector2 vector2_rotate (
    Vector2 a,
    double angle )
```

4.17.2.19 vector2_rotate90()

```
Vector2 vector2_rotate90 (
    Vector2 a )
```

4.17.2.20 vector2_scale()

```
Vector2 vector2_scale (
    Vector2 a,
    double b )
```

4.17.2.21 vector2_subtract()

```
Vector2 vector2_subtract (
    Vector2 a,
    Vector2 b )
```

4.17.2.22 vector2_up()

```
Vector2 vector2_up ( )
```

4.17.2.23 vector2_zero()

```
Vector2 vector2_zero ( )
```

4.18 vector2.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #ifndef _WIN32
00004     #include <SDL.h>
00005 #elif defined(__unix__) || defined(__linux__)
00006     #include <SDL2/SDL.h>
00007 #endif
00008
00009 typedef struct Vector2
00010 {
00011     double x, y;
00012 } Vector2;
00013
00014 Vector2 vector2_create(double x, double y);
00015 Vector2 vector2_from_polar(double angle, double length);
00016 Vector2 vector2_from_point(SDL_Point point);
00017
00018 Vector2 vector2_zero();
00019 Vector2 vector2_one();
00020 Vector2 vector2_up();
00021 Vector2 vector2_down();
00022 Vector2 vector2_left();
00023 Vector2 vector2_right();
00024
00025 Vector2 vector2_add(Vector2 a, Vector2 b);
00026 Vector2 vector2_subtract(Vector2 a, Vector2 b);
00027 Vector2 vector2_scale(Vector2 a, double b);
00028 Vector2 vector2_negate(Vector2 a);
00029 Vector2 vector2_multiply(Vector2 a, Vector2 b);
00030 Vector2 vector2_divide(Vector2 a, Vector2 b);
00031 double vector2_dot(Vector2 a, Vector2 b);
00032 double vector2_length(Vector2 a);
00033 double vector2_distance(Vector2 a, Vector2 b);
00034 double vector2_angle(Vector2 a);
00035 Vector2 vector2_normalize(Vector2 a);
00036 Vector2 vector2_rotate90(Vector2 a);
00037 Vector2 vector2_rotate(Vector2 a, double angle);
00038 Vector2 vector2_reflect(Vector2 a, Vector2 normal);
```

4.19 src/includes.h File Reference

```
#include "app/app.h"
#include "color/color.h"
#include "font/font.h"
#include "geometry/coordinate_system/coordinate_system.h"
#include "geometry/shape/shape.h"
#include "geometry/vector2/vector2.h"
#include "input/input.h"
#include "renderer/renderer.h"
#include "texture/texture.h"
#include "ui/ui.h"
#include "ui/ui_constraint/ui_constraint.h"
#include "ui/ui_element/ui_element.h"
#include "utils/math/math.h"
#include "utils/vector/vector.h"
#include "window/window.h"
```

4.20 includes.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include "app/app.h"
00004 #include "color/color.h"
00005 #include "font/font.h"
00006 #include "geometry/coordinate_system/coordinate_system.h"
00007 #include "geometry/shape/shape.h"
00008 #include "geometry/vector2/vector2.h"
00009 #include "input/input.h"
00010 #include "renderer/renderer.h"
00011 #include "texture/texture.h"
00012 #include "ui/ui.h"
00013 #include "ui/ui_constraint/ui_constraint.h"
00014 #include "ui/ui_element/ui_element.h"
00015 #include "utils/math/math.h"
00016 #include "utils/vector/vector.h"
00017 #include "window/window.h"
```

4.21 src/input/input.c File Reference

```
#include "input.h"
#include "../app/app.h"
```

Functions

- bool [input_is_mouse_button_down](#) (int button)
- bool [input_is_mouse_button_pressed](#) (int button)
- bool [input_is_mouse_button_released](#) (int button)
- bool [input_is_key_down](#) (int key)
- bool [input_is_key_pressed](#) (int key)
- bool [input_is_key_released](#) (int key)
- SDL_Point [input_get_mouse_position](#) ()
- SDL_Point [input_get_mouse_motion](#) ()

- int [input_get_mouse_wheel_delta](#) ()
- void [_input_init](#) ([InputData](#) *input_data)
- void [_input_handle_event](#) ([InputData](#) *input_data, [SDL_Event](#) *event)
- void [_input_reset](#) ([InputData](#) *input_data)
- void [_input_close](#) ([InputData](#) *input_data)
- void [_input_set_target](#) ([InputData](#) *input_data)

Variables

- [InputData](#) * [target_input_data](#)

4.21.1 Function Documentation

4.21.1.1 [_input_close\(\)](#)

```
void _input_close (  
    InputData * input_data )
```

4.21.1.2 [_input_handle_event\(\)](#)

```
void _input_handle_event (  
    InputData * input_data,  
    SDL\_Event * event )
```

4.21.1.3 [_input_init\(\)](#)

```
void _input_init (  
    InputData * input_data )
```

4.21.1.4 [_input_reset\(\)](#)

```
void _input_reset (  
    InputData * input_data )
```

4.21.1.5 [_input_set_target\(\)](#)

```
void _input_set_target (  
    InputData * input_data )
```

4.21.1.6 [input_get_mouse_motion\(\)](#)

```
SDL\_Point input_get_mouse_motion ( )
```

4.21.1.7 input_get_mouse_position()

```
SDL_Point input_get_mouse_position ( )
```

4.21.1.8 input_get_mouse_wheel_delta()

```
int input_get_mouse_wheel_delta ( )
```

4.21.1.9 input_is_key_down()

```
bool input_is_key_down (
    int key )
```

4.21.1.10 input_is_key_pressed()

```
bool input_is_key_pressed (
    int key )
```

4.21.1.11 input_is_key_released()

```
bool input_is_key_released (
    int key )
```

4.21.1.12 input_is_mouse_button_down()

```
bool input_is_mouse_button_down (
    int button )
```

4.21.1.13 input_is_mouse_button_pressed()

```
bool input_is_mouse_button_pressed (
    int button )
```

4.21.1.14 input_is_mouse_button_released()

```
bool input_is_mouse_button_released (
    int button )
```

4.21.2 Variable Documentation

4.21.2.1 target_input_data

```
InputData* target_input_data
```

4.22 src/input/input.h File Reference

```
#include <stdbool.h>
```

Classes

- struct [InputData](#)

Typedefs

- typedef struct [InputData](#) [InputData](#)

Functions

- bool [input_is_mouse_button_down](#) (int button)
- bool [input_is_mouse_button_pressed](#) (int button)
- bool [input_is_mouse_button_released](#) (int button)
- bool [input_is_key_down](#) (int key)
- bool [input_is_key_pressed](#) (int key)
- bool [input_is_key_released](#) (int key)
- [SDL_Point](#) [input_get_mouse_position](#) ()
- [SDL_Point](#) [input_get_mouse_motion](#) ()
- int [input_get_mouse_wheel_delta](#) ()
- void [_input_init](#) ([InputData](#) *input_data)
- void [_input_handle_event](#) ([InputData](#) *input_data, [SDL_Event](#) *event)
- void [_input_reset](#) ([InputData](#) *input_data)
- void [_input_close](#) ([InputData](#) *input_data)
- void [_input_set_target](#) ([InputData](#) *input_data)

4.22.1 Typedef Documentation

4.22.1.1 InputData

```
typedef struct InputData InputData
```

4.22.2 Function Documentation

4.22.2.1 _input_close()

```
void _input_close (  
    InputData * input_data )
```

4.22.2.2 _input_handle_event()

```
void _input_handle_event (  
    InputData * input_data,  
    SDL\_Event * event )
```

4.22.2.3 `_input_init()`

```
void _input_init (
    InputData * input_data )
```

4.22.2.4 `_input_reset()`

```
void _input_reset (
    InputData * input_data )
```

4.22.2.5 `_input_set_target()`

```
void _input_set_target (
    InputData * input_data )
```

4.22.2.6 `input_get_mouse_motion()`

```
SDL_Point input_get_mouse_motion ( )
```

4.22.2.7 `input_get_mouse_position()`

```
SDL_Point input_get_mouse_position ( )
```

4.22.2.8 `input_get_mouse_wheel_delta()`

```
int input_get_mouse_wheel_delta ( )
```

4.22.2.9 `input_is_key_down()`

```
bool input_is_key_down (
    int key )
```

4.22.2.10 `input_is_key_pressed()`

```
bool input_is_key_pressed (
    int key )
```

4.22.2.11 `input_is_key_released()`

```
bool input_is_key_released (
    int key )
```


4.22.2.12 input_is_mouse_button_down()

```
bool input_is_mouse_button_down (
    int button )
```

4.22.2.13 input_is_mouse_button_pressed()

```
bool input_is_mouse_button_pressed (
    int button )
```

4.22.2.14 input_is_mouse_button_released()

```
bool input_is_mouse_button_released (
    int button )
```

4.23 input.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #ifndef _WIN32
00004     #include <SDL.h>
00005 #elif defined(__unix__) || defined(__linux__)
00006     #include <SDL2/SDL.h>
00007 #endif
00008
00009 #include <stdbool.h>
00010
00011 typedef struct InputData
00012 {
00013     //mouse
00014     bool current_mouse_button_state[5];
00015     bool old_mouse_button_state[5];
00016     SDL_Point current_mouse_position;
00017     SDL_Point old_mouse_position;
00018     int mouse_wheel_delta;
00019
00020     //keyboard
00021     Uint8* current_keyboard_state;
00022     Uint8* old_keyboard_state;
00023     int key_count;
00024 } InputData;
00025
00026 //API functions
00027 bool input_is_mouse_button_down(int button);
00028 bool input_is_mouse_button_pressed(int button);
00029 bool input_is_mouse_button_released(int button);
00030
00031 bool input_is_key_down(int key);
00032 bool input_is_key_pressed(int key);
00033 bool input_is_key_released(int key);
00034
00035 SDL_Point input_get_mouse_position();
00036 SDL_Point input_get_mouse_motion();
00037 int input_get_mouse_wheel_delta();
00038
00039 //internal functions
00040 void _input_init(InputData* input_data);
00041 void _input_handle_event(InputData* input_data, SDL_Event* event);
00042 void _input_reset(InputData* input_data);
00043 void _input_close(InputData* input_data);
00044 void _input_set_target(InputData* input_data);
```

4.24 src/main.c File Reference

This is the entry point of the application.

```
#include "includes.h"
#include "utils/vector/vector.h"
```

Macros

- `#define FPS 60`
- `#define MOUSE_WHEEL_SENSITIVITY 5`

Typedefs

- `typedef enum State State`

Enumerations

- `enum State {`
`STATE_POINTER, STATE_CS_DRAGGED, STATE_POINT, STATE_LINE,`
`STATE_LINE_POINT1_PLACED, STATE_CIRCLE, STATE_CIRCLE_CENTER_PLACED }`

Functions

- `void on_pointer_clicked (UIButton *self)`
- `void on_point_clicked (UIButton *self)`
- `void on_line_clicked (UIButton *self)`
- `void on_circle_clicked (UIButton *self)`
- `void on_filemenu_clicked (UISplitButton *self, Sint32 index)`
- `void on_editmenu_clicked (UISplitButton *self, Sint32 index)`
- `void on_canvas_size_changed (UIContainer *self, SDL_Point size)`
- `int main (void)`
- `void on_pointer_clicked (UIButton *self __attribute__((unused)))`
- `void on_point_clicked (UIButton *self __attribute__((unused)))`
- `void on_line_clicked (UIButton *self __attribute__((unused)))`
- `void on_circle_clicked (UIButton *self __attribute__((unused)))`
- `void on_filemenu_clicked (UISplitButton *self __attribute__((unused)), Sint32 index __attribute__((unused)))`
- `void on_editmenu_clicked (UISplitButton *self __attribute__((unused)), Sint32 index __attribute__((unused)))`

Variables

- `CoordinateSystem * cs`
- `State state = STATE_POINTER`

4.24.1 Detailed Description

This is the entry point of the application.

Author

Örs Mándli (mandliors@gmail.com)

Version

0.1

Date

2023-11-05

Copyright

Copyright (c) 2023

4.24.2 Macro Definition Documentation

4.24.2.1 FPS

```
#define FPS 60
```

4.24.2.2 MOUSE_WHEEL_SENSITIVITY

```
#define MOUSE_WHEEL_SENSITIVITY 5
```

4.24.3 Typedef Documentation

4.24.3.1 State

```
typedef enum State State
```

4.24.4 Enumeration Type Documentation

4.24.4.1 State

```
enum State
```

Enumerator

	STATE_POINTER	
	STATE_CS_DRAGGED	
Generated by Doxygen	STATE_POINT	
	STATE_LINE	
	STATE_LINE_POINT1_PLACED	
	STATE_CIRCLE	

4.24.5 Function Documentation

4.24.5.1 main()

```
int main (
    void )
```

4.24.5.2 on_canvas_size_changed()

```
void on_canvas_size_changed (
    UIContainer * self,
    SDL_Point size )
```

4.24.5.3 on_circle_clicked() [1/2]

```
void on_circle_clicked (
    UIButton *self __attribute__((unused)) )
```

4.24.5.4 on_circle_clicked() [2/2]

```
void on_circle_clicked (
    UIButton * self )
```

4.24.5.5 on_editmenu_clicked() [1/2]

```
void on_editmenu_clicked (
    UISplitButton *self __attribute__((unused)),
    Sint32 index __attribute__((unused)) )
```

4.24.5.6 on_editmenu_clicked() [2/2]

```
void on_editmenu_clicked (
    UISplitButton * self,
    Sint32 index )
```

4.24.5.7 on_filemenu_clicked() [1/2]

```
void on_filemenu_clicked (
    UISplitButton *self __attribute__((unused)),
    Sint32 index __attribute__((unused)) )
```

4.24.5.8 on_filemenu_clicked() [2/2]

```
void on_filemenu_clicked (
    UISplitButton * self,
    Sint32 index )
```

4.24.5.9 on_line_clicked() [1/2]

```
void on_line_clicked (
    UIButton *self  __attribute__((unused)) )
```

4.24.5.10 on_line_clicked() [2/2]

```
void on_line_clicked (
    UIButton * self )
```

4.24.5.11 on_point_clicked() [1/2]

```
void on_point_clicked (
    UIButton *self  __attribute__((unused)) )
```

4.24.5.12 on_point_clicked() [2/2]

```
void on_point_clicked (
    UIButton * self )
```

4.24.5.13 on_pointer_clicked() [1/2]

```
void on_pointer_clicked (
    UIButton *self  __attribute__((unused)) )
```

4.24.5.14 on_pointer_clicked() [2/2]

```
void on_pointer_clicked (
    UIButton * self )
```

4.24.6 Variable Documentation

4.24.6.1 cs

```
CoordinateSystem* cs
```

4.24.6.2 state

```
State state = STATE_POINTER
```

4.25 src/renderer/renderer.c File Reference

```
#include "renderer.h"
#include "../texture/texture.h"
#include "../font/font.h"
```

Functions

- void [renderer_set_default_font](#) ([Font](#) *font)
- void [renderer_set_clip_rect](#) (int x, int y, int width, int height)
- void [renderer_reset_clip_rect](#) ()
- [Texture](#) * [renderer_create_framebuffer](#) (int width, int height)
- void [renderer_resize_framebuffer](#) ([Texture](#) *framebuffer, int width, int height)
- void [renderer_bind_framebuffer](#) ([Texture](#) *framebuffer)
- void [renderer_clear](#) ([Color](#) color)
- void [renderer_draw_pixel](#) (int x, int y, [Color](#) color)
- void [renderer_draw_line](#) (int x1, int y1, int x2, int y2, int thickness, [Color](#) color)
- void [renderer_draw_rect](#) (int x, int y, int width, int height, [Color](#) color)
- void [renderer_draw_filled_rect](#) (int x, int y, int width, int height, [Color](#) color)
- void [renderer_draw_circle](#) (int x, int y, int radius, [Color](#) color)
- void [renderer_draw_filled_circle](#) (int x, int y, int radius, [Color](#) color)
- void [renderer_draw_ellipse](#) (int x, int y, int rx, int ry, [Color](#) color)
- void [renderer_draw_filled_ellipse](#) (int x, int y, int rx, int ry, [Color](#) color)
- void [renderer_draw_triangle](#) (int x1, int y1, int x2, int y2, int x3, int y3, [Color](#) color)
- void [renderer_draw_filled_triangle](#) (int x1, int y1, int x2, int y2, int x3, int y3, [Color](#) color)
- void [renderer_draw_rounded_rect](#) (int x, int y, int width, int height, int radius, [Color](#) color)
- void [renderer_draw_filled_rounded_rect](#) (int x, int y, int width, int height, int radius, [Color](#) color)
- void [renderer_draw_polygon](#) (const short *vx, const short *vy, int n, [Color](#) color)
- void [renderer_draw_filled_polygon](#) (const short *vx, const short *vy, int n, [Color](#) color)
- void [renderer_draw_arc](#) (int x, int y, int radius, int start, int end, [Color](#) color)
- void [renderer_draw_pie](#) (int x, int y, int radius, int start, int end, [Color](#) color)
- void [renderer_draw_filled_pie](#) (int x, int y, int radius, int start, int end, [Color](#) color)
- void [renderer_draw_bezier](#) (const short *vx, const short *vy, int n, int s, [Color](#) color)
- void [renderer_draw_texture](#) ([Texture](#) *texture, int x, int y, int width, int height)
- void [renderer_draw_text](#) (const char *text, int x, int y, [Color](#) color)
- [SDL_Point](#) [renderer_query_text_size](#) (const char *text)
- void [_renderer_set_target](#) ([SDL_Renderer](#) *renderer)

Variables

- [SDL_Renderer](#) * [target_renderer](#)
- [Font](#) * [default_font](#)

4.25.1 Function Documentation

4.25.1.1 _renderer_set_target()

```
void _renderer_set_target (
    SDL\_Renderer * renderer )
```

4.25.1.2 `renderer_bind_framebuffer()`

```
void renderer_bind_framebuffer (
    Texture * framebuffer )
```

4.25.1.3 `renderer_clear()`

```
void renderer_clear (
    Color color )
```

4.25.1.4 `renderer_create_framebuffer()`

```
Texture * renderer_create_framebuffer (
    int width,
    int height )
```

4.25.1.5 `renderer_draw_arc()`

```
void renderer_draw_arc (
    int x,
    int y,
    int radius,
    int start,
    int end,
    Color color )
```

4.25.1.6 `renderer_draw_bezier()`

```
void renderer_draw_bezier (
    const short * vx,
    const short * vy,
    int n,
    int s,
    Color color )
```

4.25.1.7 `renderer_draw_circle()`

```
void renderer_draw_circle (
    int x,
    int y,
    int radius,
    Color color )
```

4.25.1.8 `renderer_draw_ellipse()`

```
void renderer_draw_ellipse (
    int x,
    int y,
    int rx,
    int ry,
    Color color )
```

4.25.1.9 `renderer_draw_filled_circle()`

```
void renderer_draw_filled_circle (
    int x,
    int y,
    int radius,
    Color color )
```

4.25.1.10 `renderer_draw_filled_ellipse()`

```
void renderer_draw_filled_ellipse (
    int x,
    int y,
    int rx,
    int ry,
    Color color )
```

4.25.1.11 `renderer_draw_filled_pie()`

```
void renderer_draw_filled_pie (
    int x,
    int y,
    int radius,
    int start,
    int end,
    Color color )
```

4.25.1.12 `renderer_draw_filled_polygon()`

```
void renderer_draw_filled_polygon (
    const short * vx,
    const short * vy,
    int n,
    Color color )
```

4.25.1.13 `renderer_draw_filled_rect()`

```
void renderer_draw_filled_rect (
    int x,
    int y,
    int width,
    int height,
    Color color )
```

4.25.1.14 `renderer_draw_filled_rounded_rect()`

```
void renderer_draw_filled_rounded_rect (
    int x,
    int y,
    int width,
    int height,
    int radius,
    Color color )
```


4.25.1.15 `renderer_draw_filled_triangle()`

```
void renderer_draw_filled_triangle (
    int x1,
    int y1,
    int x2,
    int y2,
    int x3,
    int y3,
    Color color )
```

4.25.1.16 `renderer_draw_line()`

```
void renderer_draw_line (
    int x1,
    int y1,
    int x2,
    int y2,
    int thickness,
    Color color )
```

4.25.1.17 `renderer_draw_pie()`

```
void renderer_draw_pie (
    int x,
    int y,
    int radius,
    int start,
    int end,
    Color color )
```

4.25.1.18 `renderer_draw_pixel()`

```
void renderer_draw_pixel (
    int x,
    int y,
    Color color )
```

4.25.1.19 `renderer_draw_polygon()`

```
void renderer_draw_polygon (
    const short * vx,
    const short * vy,
    int n,
    Color color )
```

4.25.1.20 `renderer_draw_rect()`

```
void renderer_draw_rect (
    int x,
    int y,
    int width,
    int height,
    Color color )
```

4.25.1.21 `renderer_draw_rounded_rect()`

```
void renderer_draw_rounded_rect (
    int x,
    int y,
    int width,
    int height,
    int radius,
    Color color )
```

4.25.1.22 `renderer_draw_text()`

```
void renderer_draw_text (
    const char * text,
    int x,
    int y,
    Color color )
```

4.25.1.23 `renderer_draw_texture()`

```
void renderer_draw_texture (
    Texture * texture,
    int x,
    int y,
    int width,
    int height )
```

4.25.1.24 `renderer_draw_triangle()`

```
void renderer_draw_triangle (
    int x1,
    int y1,
    int x2,
    int y2,
    int x3,
    int y3,
    Color color )
```

4.25.1.25 `renderer_query_text_size()`

```
SDL_Point renderer_query_text_size (
    const char * text )
```

4.25.1.26 `renderer_reset_clip_rect()`

```
void renderer_reset_clip_rect ( )
```

4.25.1.27 `renderer_resize_framebuffer()`

```
void renderer_resize_framebuffer (
    Texture * framebuffer,
    int width,
    int height )
```

4.25.1.28 `renderer_set_clip_rect()`

```
void renderer_set_clip_rect (
    int x,
    int y,
    int width,
    int height )
```

4.25.1.29 `renderer_set_default_font()`

```
void renderer_set_default_font (
    Font * font )
```

4.25.2 Variable Documentation

4.25.2.1 `default_font`

```
Font* default_font
```

4.25.2.2 `target_renderer`

```
SDL_Renderer* target_renderer
```

4.26 src/renderer/renderer.h File Reference

```
#include "../color/color.h"
#include "../font/font.h"
#include "../texture/texture.h"
```

Functions

- void `renderer_set_default_font` (`Font *font`)
- void `renderer_set_clip_rect` (`int x`, `int y`, `int width`, `int height`)
- void `renderer_reset_clip_rect` ()
- `Texture *` `renderer_create_framebuffer` (`int width`, `int height`)
- void `renderer_resize_framebuffer` (`Texture *framebuffer`, `int width`, `int height`)
- void `renderer_bind_framebuffer` (`Texture *framebuffer`)
- void `renderer_clear` (`Color color`)
- void `renderer_draw_pixel` (`int x`, `int y`, `Color color`)
- void `renderer_draw_line` (`int x1`, `int y1`, `int x2`, `int y2`, `int thickness`, `Color color`)
- void `renderer_draw_rect` (`int x`, `int y`, `int width`, `int height`, `Color color`)
- void `renderer_draw_filled_rect` (`int x`, `int y`, `int width`, `int height`, `Color color`)
- void `renderer_draw_circle` (`int x`, `int y`, `int radius`, `Color color`)
- void `renderer_draw_filled_circle` (`int x`, `int y`, `int radius`, `Color color`)
- void `renderer_draw_ellipse` (`int x`, `int y`, `int rx`, `int ry`, `Color color`)
- void `renderer_draw_filled_ellipse` (`int x`, `int y`, `int rx`, `int ry`, `Color color`)
- void `renderer_draw_triangle` (`int x1`, `int y1`, `int x2`, `int y2`, `int x3`, `int y3`, `Color color`)
- void `renderer_draw_filled_triangle` (`int x1`, `int y1`, `int x2`, `int y2`, `int x3`, `int y3`, `Color color`)
- void `renderer_draw_rounded_rect` (`int x`, `int y`, `int width`, `int height`, `int radius`, `Color color`)
- void `renderer_draw_filled_rounded_rect` (`int x`, `int y`, `int width`, `int height`, `int radius`, `Color color`)
- void `renderer_draw_polygon` (`const short *vx`, `const short *vy`, `int n`, `Color color`)
- void `renderer_draw_filled_polygon` (`const short *vx`, `const short *vy`, `int n`, `Color color`)
- void `renderer_draw_arc` (`int x`, `int y`, `int radius`, `int start`, `int end`, `Color color`)
- void `renderer_draw_pie` (`int x`, `int y`, `int radius`, `int start`, `int end`, `Color color`)
- void `renderer_draw_filled_pie` (`int x`, `int y`, `int radius`, `int start`, `int end`, `Color color`)
- void `renderer_draw_bezier` (`const short *vx`, `const short *vy`, `int n`, `int s`, `Color color`)
- void `renderer_draw_texture` (`Texture *texture`, `int x`, `int y`, `int width`, `int height`)
- void `renderer_draw_text` (`const char *text`, `int x`, `int y`, `Color color`)
- `SDL_Point` `renderer_query_text_size` (`const char *text`)
- void `_renderer_set_target` (`SDL_Renderer *renderer`)

4.26.1 Function Documentation

4.26.1.1 _renderer_set_target()

```
void _renderer_set_target (
    SDL_Renderer * renderer )
```

4.26.1.2 renderer_bind_framebuffer()

```
void renderer_bind_framebuffer (
    Texture * framebuffer )
```

4.26.1.3 renderer_clear()

```
void renderer_clear (
    Color color )
```

4.26.1.4 `renderer_create_framebuffer()`

```
Texture * renderer_create_framebuffer (
    int width,
    int height )
```

4.26.1.5 `renderer_draw_arc()`

```
void renderer_draw_arc (
    int x,
    int y,
    int radius,
    int start,
    int end,
    Color color )
```

4.26.1.6 `renderer_draw_bezier()`

```
void renderer_draw_bezier (
    const short * vx,
    const short * vy,
    int n,
    int s,
    Color color )
```

4.26.1.7 `renderer_draw_circle()`

```
void renderer_draw_circle (
    int x,
    int y,
    int radius,
    Color color )
```

4.26.1.8 `renderer_draw_ellipse()`

```
void renderer_draw_ellipse (
    int x,
    int y,
    int rx,
    int ry,
    Color color )
```

4.26.1.9 `renderer_draw_filled_circle()`

```
void renderer_draw_filled_circle (
    int x,
    int y,
    int radius,
    Color color )
```

4.26.1.10 `renderer_draw_filled_ellipse()`

```
void renderer_draw_filled_ellipse (
    int x,
    int y,
    int rx,
    int ry,
    Color color )
```

4.26.1.11 `renderer_draw_filled_pie()`

```
void renderer_draw_filled_pie (
    int x,
    int y,
    int radius,
    int start,
    int end,
    Color color )
```

4.26.1.12 `renderer_draw_filled_polygon()`

```
void renderer_draw_filled_polygon (
    const short * vx,
    const short * vy,
    int n,
    Color color )
```

4.26.1.13 `renderer_draw_filled_rect()`

```
void renderer_draw_filled_rect (
    int x,
    int y,
    int width,
    int height,
    Color color )
```

4.26.1.14 `renderer_draw_filled_rounded_rect()`

```
void renderer_draw_filled_rounded_rect (
    int x,
    int y,
    int width,
    int height,
    int radius,
    Color color )
```

4.26.1.15 `renderer_draw_filled_triangle()`

```
void renderer_draw_filled_triangle (
    int x1,
    int y1,
    int x2,
    int y2,
    int x3,
    int y3,
    Color color )
```

4.26.1.16 `renderer_draw_line()`

```
void renderer_draw_line (
    int x1,
    int y1,
    int x2,
    int y2,
    int thickness,
    Color color )
```

4.26.1.17 `renderer_draw_pie()`

```
void renderer_draw_pie (
    int x,
    int y,
    int radius,
    int start,
    int end,
    Color color )
```

4.26.1.18 `renderer_draw_pixel()`

```
void renderer_draw_pixel (
    int x,
    int y,
    Color color )
```

4.26.1.19 `renderer_draw_polygon()`

```
void renderer_draw_polygon (
    const short * vx,
    const short * vy,
    int n,
    Color color )
```

4.26.1.20 `renderer_draw_rect()`

```
void renderer_draw_rect (
    int x,
    int y,
    int width,
    int height,
    Color color )
```

4.26.1.21 `renderer_draw_rounded_rect()`

```
void renderer_draw_rounded_rect (
    int x,
    int y,
    int width,
    int height,
    int radius,
    Color color )
```

4.26.1.22 `renderer_draw_text()`

```
void renderer_draw_text (
    const char * text,
    int x,
    int y,
    Color color )
```

4.26.1.23 `renderer_draw_texture()`

```
void renderer_draw_texture (
    Texture * texture,
    int x,
    int y,
    int width,
    int height )
```

4.26.1.24 `renderer_draw_triangle()`

```
void renderer_draw_triangle (
    int x1,
    int y1,
    int x2,
    int y2,
    int x3,
    int y3,
    Color color )
```

4.26.1.25 `renderer_query_text_size()`

```
SDL_Point renderer_query_text_size (
    const char * text )
```


4.26.1.26 renderer_reset_clip_rect()

```
void renderer_reset_clip_rect ( )
```

4.26.1.27 renderer_resize_framebuffer()

```
void renderer_resize_framebuffer (
    Texture * framebuffer,
    int width,
    int height )
```

4.26.1.28 renderer_set_clip_rect()

```
void renderer_set_clip_rect (
    int x,
    int y,
    int width,
    int height )
```

4.26.1.29 renderer_set_default_font()

```
void renderer_set_default_font (
    Font * font )
```

4.27 renderer.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #ifdef _WIN32
00004     #include <SDL.h>
00005     #include <SDL2_gfxPrimitives.h>
00006 #elif defined(__unix__) || defined(__linux__)
00007     #include <SDL2/SDL.h>
00008     #include <SDL2/SDL2_gfxPrimitives.h>
00009 #endif
00010
00011 #include "../color/color.h"
00012 #include "../font/font.h"
00013 #include "../texture/texture.h"
00014
00015 void renderer_set_default_font(Font* font);
00016 void renderer_set_clip_rect(int x, int y, int width, int height);
00017 void renderer_reset_clip_rect();
00018
00019 Texture* renderer_create_framebuffer(int width, int height);
00020 void renderer_resize_framebuffer(Texture* framebuffer, int width, int height);
00021 void renderer_bind_framebuffer(Texture* framebuffer);
00022
00023 void renderer_clear(Color color);
00024 void renderer_draw_pixel(int x, int y, Color color);
00025 void renderer_draw_line(int x1, int y1, int x2, int y2, int thickness, Color color);
00026 void renderer_draw_rect(int x, int y, int width, int height, Color color);
00027 void renderer_draw_filled_rect(int x, int y, int width, int height, Color color);
00028 void renderer_draw_circle(int x, int y, int radius, Color color);
00029 void renderer_draw_filled_circle(int x, int y, int radius, Color color);
00030 void renderer_draw_ellipse(int x, int y, int rx, int ry, Color color);
00031 void renderer_draw_filled_ellipse(int x, int y, int rx, int ry, Color color);
00032 void renderer_draw_triangle(int x1, int y1, int x2, int y2, int x3, int y3, Color color);
00033 void renderer_draw_filled_triangle(int x1, int y1, int x2, int y2, int x3, int y3, Color color);
00034 void renderer_draw_rounded_rect(int x, int y, int width, int height, int radius, Color color);
```

```

00035 void renderer_draw_filled_rounded_rect(int x, int y, int width, int height, int radius, Color color);
00036 void renderer_draw_polygon(const short* vx, const short* vy, int n, Color color);
00037 void renderer_draw_filled_polygon(const short* vx, const short* vy, int n, Color color);
00038 void renderer_draw_arc(int x, int y, int radius, int start, int end, Color color);
00039 void renderer_draw_pie(int x, int y, int radius, int start, int end, Color color);
00040 void renderer_draw_filled_pie(int x, int y, int radius, int start, int end, Color color);
00041 void renderer_draw_bezier(const short* vx, const short* vy, int n, int s, Color color);
00042 void renderer_draw_texture(Texture* texture, int x, int y, int width, int height);
00043 void renderer_draw_text(const char* text, int x, int y, Color color);
00044 SDL_Point renderer_query_text_size(const char* text);
00045
00046 void _renderer_set_target(SDL_Renderer* renderer);

```

4.28 src/texture/texture.c File Reference

```

#include "texture.h"
#include "../utils/vector/vector.h"

```

Functions

- [Texture](#) * [texture_load](#) (SDL_Renderer *renderer, const char *path)
- void [_texture_init](#) ()
- void [_texture_add](#) ([Texture](#) *texture)
- void [_texture_close](#) ()

Variables

- [Vector](#) * [textures](#)

4.28.1 Function Documentation

4.28.1.1 _texture_add()

```

void _texture_add (
    Texture * texture )

```

4.28.1.2 _texture_close()

```

void _texture_close ( )

```

4.28.1.3 _texture_init()

```

void _texture_init ( )

```

4.28.1.4 texture_load()

```

Texture * texture_load (
    SDL_Renderer * renderer,
    const char * path )

```

4.28.2 Variable Documentation

4.28.2.1 textures

`Vector*` textures

4.29 src/texture/texture.h File Reference

Classes

- struct `Texture`

Typedefs

- typedef struct `Texture` `Texture`

Functions

- `Texture*` `texture_load` (`SDL_Renderer*` renderer, `const char*` path)
- void `_texture_init` ()
- void `_texture_add` (`Texture*` texture)
- void `_texture_close` ()

4.29.1 Typedef Documentation

4.29.1.1 Texture

```
typedef struct Texture Texture
```

4.29.2 Function Documentation

4.29.2.1 _texture_add()

```
void _texture_add (  
    Texture * texture )
```

4.29.2.2 _texture_close()

```
void _texture_close ( )
```

4.29.2.3 _texture_init()

```
void _texture_init ( )
```

4.29.2.4 texture_load()

```
Texture * texture_load (
    SDL_Renderer * renderer,
    const char * path )
```

4.30 texture.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #ifdef _WIN32
00004     #include <SDL.h>
00005     #include <SDL_image.h>
00006 #elif defined(__unix__) || defined(__linux__)
00007     #include <SDL2/SDL.h>
00008     #include <SDL2/SDL_image.h>
00009 #endif
00010
00011 typedef struct Texture
00012 {
00013     SDL_Texture* texture;
00014     int width;
00015     int height;
00016 } Texture;
00017
00018 Texture* texture_load(SDL_Renderer* renderer, const char* path);
00019
00020 //internal functions
00021 void _texture_init();
00022 void _texture_add(Texture* texture);
00023 void _texture_close();
```

4.31 src/ui/ui.c File Reference

```
#include "ui.h"
#include "../app/app.h"
```

Functions

- void [_ui_init](#) (UIData *ui_data, int width, int height)
- void [_ui_handle_event](#) (UIData *ui_data, SDL_Event *event)
- void [_ui_update](#) (UIData *ui_data)
- void [_ui_render](#) (UIData *ui_data)
- void [_ui_close](#) (UIData *ui_data)
- void [_ui_set_target](#) (UIData *ui_data)
- UIData * [_ui_get_target](#) ()

Variables

- UIData * [target_ui_data](#) = NULL

4.31.1 Function Documentation

4.31.1.1 `_ui_close()`

```
void _ui_close (
    UIData * ui_data )
```

4.31.1.2 `_ui_get_target()`

```
UIData * _ui_get_target ( )
```

4.31.1.3 `_ui_handle_event()`

```
void _ui_handle_event (
    UIData * ui_data,
    SDL_Event * event )
```

4.31.1.4 `_ui_init()`

```
void _ui_init (
    UIData * ui_data,
    int width,
    int height )
```

4.31.1.5 `_ui_render()`

```
void _ui_render (
    UIData * ui_data )
```

4.31.1.6 `_ui_set_target()`

```
void _ui_set_target (
    UIData * ui_data )
```

4.31.1.7 `_ui_update()`

```
void _ui_update (
    UIData * ui_data )
```

4.31.2 Variable Documentation

4.31.2.1 `target_ui_data`

```
UIData* target_ui_data = NULL
```

4.32 src/ui/ui.h File Reference

```
#include "ui_element/ui_element.h"
```

Classes

- struct [UIData](#)

Typedefs

- typedef struct [UIData](#) [UIData](#)

Functions

- void [_ui_init](#) ([UIData](#) *ui_data, int width, int height)
- void [_ui_handle_event](#) ([UIData](#) *ui_data, [SDL_Event](#) *event)
- void [_ui_update](#) ([UIData](#) *ui_data)
- void [_ui_render](#) ([UIData](#) *ui_data)
- void [_ui_close](#) ([UIData](#) *ui_data)
- void [_ui_set_target](#) ([UIData](#) *ui_data)
- [UIData](#) * [_ui_get_target](#) ()

4.32.1 Typedef Documentation

4.32.1.1 UIData

```
typedef struct UIData UIData
```

4.32.2 Function Documentation

4.32.2.1 _ui_close()

```
void _ui_close (  
    UIData * ui_data )
```

4.32.2.2 _ui_get_target()

```
UIData * _ui_get_target ( )
```

4.32.2.3 _ui_handle_event()

```
void _ui_handle_event (  
    UIData * ui_data,  
    SDL\_Event * event )
```

4.32.2.4 _ui_init()

```
void _ui_init (
    UIData * ui_data,
    int width,
    int height )
```

4.32.2.5 _ui_render()

```
void _ui_render (
    UIData * ui_data )
```

4.32.2.6 _ui_set_target()

```
void _ui_set_target (
    UIData * ui_data )
```

4.32.2.7 _ui_update()

```
void _ui_update (
    UIData * ui_data )
```

4.33 ui.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include "ui_element/ui_element.h"
00004
00005 typedef struct UIData
00006 {
00007     UIContainer* main_container;
00008     char text_input[SDL_TEXTINPUTEVENT_TEXT_SIZE];
00009     bool backspace_pressed;
00010     bool mouse_captured;
00011     UISplitButton* expanded_splitbutton;
00012 } UIData;
00013
00014 //internal functions
00015 void _ui_init(UIData* ui_data, int width, int height);
00016 void _ui_handle_event(UIData* ui_data, SDL_Event* event);
00017 void _ui_update(UIData* ui_data);
00018 void _ui_render(UIData* ui_data);
00019 void _ui_close(UIData* ui_data);
00020 void _ui_set_target(UIData* ui_data);
00021 UIData* _ui_get_target();
```

4.34 src/ui/ui_constraint/ui_constraint.c File Reference

```
#include "ui_constraint.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Functions

- [UIConstraint new_pixel_constraint](#) (int value)
- [UIConstraint new_center_constraint](#) ()
- [UIConstraint new_relative_constraint](#) (double value)
- [UIConstraint new_offset_constraint](#) (double value)
- [UIConstraint new_aspect_constraint](#) (double value)
- [UIConstraints constraints_from_string](#) (const char *string)

4.34.1 Function Documentation

4.34.1.1 constraints_from_string()

```
UIConstraints constraints_from_string (  
    const char * string )
```

4.34.1.2 new_aspect_constraint()

```
UIConstraint new_aspect_constraint (  
    double value )
```

4.34.1.3 new_center_constraint()

```
UIConstraint new_center_constraint ( )
```

4.34.1.4 new_offset_constraint()

```
UIConstraint new_offset_constraint (  
    double value )
```

4.34.1.5 new_pixel_constraint()

```
UIConstraint new_pixel_constraint (  
    int value )
```

4.34.1.6 new_relative_constraint()

```
UIConstraint new_relative_constraint (  
    double value )
```

4.35 src/ui/ui_constraint/ui_constraint.h File Reference

Classes

- struct [UIConstraint](#)
- struct [UIConstraints](#)

Typedefs

- typedef enum [ConstraintType](#) [ConstraintType](#)
- typedef struct [UIConstraint](#) [UIConstraint](#)
- typedef struct [UIConstraints](#) [UIConstraints](#)

Enumerations

- enum [ConstraintType](#) {
[CT_PIXEL](#) = 0 , [CT_CENTER](#) , [CT_RELATIVE](#) , [CT_OFFSET](#) ,
[CT_ASPECT](#) }

Functions

- [UIConstraint new_pixel_constraint](#) (int value)
- [UIConstraint new_center_constraint](#) ()
- [UIConstraint new_relative_constraint](#) (double value)
- [UIConstraint new_offset_constraint](#) (double value)
- [UIConstraint new_aspect_constraint](#) (double value)
- [UIConstraints constraints_from_string](#) (const char *string)

4.35.1 Typedef Documentation

4.35.1.1 ConstraintType

```
typedef enum ConstraintType ConstraintType
```

4.35.1.2 UIConstraint

```
typedef struct UIConstraint UIConstraint
```

4.35.1.3 UIConstraints

```
typedef struct UIConstraints UIConstraints
```

4.35.2 Enumeration Type Documentation

4.35.2.1 ConstraintType

```
enum ConstraintType
```

Enumerator

CT_PIXEL	
CT_CENTER	
CT_RELATIVE	
CT_OFFSET	
CT_ASPECT	

4.35.3 Function Documentation

4.35.3.1 constraints_from_string()

```
UIConstraints constraints_from_string (
    const char * string )
```

4.35.3.2 new_aspect_constraint()

```
UIConstraint new_aspect_constraint (
    double value )
```

4.35.3.3 new_center_constraint()

```
UIConstraint new_center_constraint ( )
```

4.35.3.4 new_offset_constraint()

```
UIConstraint new_offset_constraint (
    double value )
```

4.35.3.5 new_pixel_constraint()

```
UIConstraint new_pixel_constraint (
    int value )
```

4.35.3.6 new_relative_constraint()

```
UIConstraint new_relative_constraint (
    double value )
```

4.36 ui_constraint.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 typedef enum ConstraintType
00004 {
00005     CT_PIXEL = 0,
00006     CT_CENTER,
00007     CT_RELATIVE,
00008     CT_OFFSET,
00009     CT_ASPECT
00010 } ConstraintType;
00011
00012 typedef struct UIConstraint
00013 {
00014     double value;
00015     ConstraintType constraint_type;
00016     void (*recalculate)(void* self);
00017 } UIConstraint;
00018
00019 typedef struct UIConstraints
00020 {
00021     UIConstraint x, y, width, height;
00022 } UIConstraints;
00023
00024 //API functions
00025 UIConstraint new_pixel_constraint(int value);
00026 UIConstraint new_center_constraint();
00027 UIConstraint new_relative_constraint(double value);
00028 UIConstraint new_offset_constraint(double value);
00029 UIConstraint new_aspect_constraint(double value);
00030 UIConstraints constraints_from_string(const char* string);
```

4.37 src/ui/ui_element/ui_element.c File Reference

```
#include "ui_element.h"
#include "../app/app.h"
#include "../renderer/renderer.h"
#include "../input/input.h"
#include "../utils/math/math.h"
#include <string.h>
```

Classes

- struct [_UIDropdownItem](#)
- struct [UISplitButtonItem](#)

Typedefs

- typedef struct [_UIDropdownItem](#) [_UIDropdownItem](#)
- typedef struct [UISplitButtonItem](#) [UISplitButtonItem](#)

Functions

- [UIContainer](#) * [ui_create_container](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, void(*on_size_changed)([UIContainer](#) *self, SDL_Point size))
- [UIPanel](#) * [ui_create_panel](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, [Color](#) color, [Color](#) border_color, Uint32 border_width, Uint32 roundness)
- [UILabel](#) * [ui_create_label](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, const char *text, [Color](#) color)
- [UIButton](#) * [ui_create_button](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, const char *text, [Color](#) color, [Color](#) text_color, void(*on_click)([UIButton](#) *self))
- [UIImageButton](#) * [ui_create_imagebutton](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, [Texture](#) *texture, void(*on_click)([UIImageButton](#) *self))
- [UITextbox](#) * [ui_create_textbox](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, const char *text, [Color](#) color, [Color](#) text_color, void(*on_text_changed)([UITextbox](#) *self, const char *text))
- [UICheckbox](#) * [ui_create_checkbox](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, [Color](#) checked_color, [Color](#) unchecked_color, void(*on_checked_changed)([UICheckbox](#) *self, bool checked))
- [UISlider](#) * [ui_create_slider](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, double value, [Color](#) color, [Color](#) slider_color, void(*on_value_changed)([UISlider](#) *self, double value))
- [UIDropdownList](#) * [ui_create_dropdown](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, char *items, [Color](#) color, [Color](#) text_color, void(*on_selection_changed)([UIDropdownList](#) *self, Sint32 index))
- [UISplitButton](#) * [ui_create_splitbutton](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, char *items, [Color](#) color, [Color](#) text_color, void(*on_item_clicked)([UISplitButton](#) *self, Sint32 index), bool auto_dropdown)
- void [_ui_container_update](#) ([UIElement](#) *self)
- void [_ui_container_recalculate](#) ([UIElement](#) *sibling, [UIElement](#) *self)
- void [_ui_container_render](#) ([UIElement](#) *self)
- void [_ui_container_destroy](#) ([UIElement](#) *self)

4.37.1 Typedef Documentation

4.37.1.1 _UIDropdownItem

```
typedef struct _UIDropdownItem _UIDropdownItem
```

4.37.1.2 _UISplitButtonItem

```
typedef struct _UISplitButtonItem _UISplitButtonItem
```

4.37.2 Function Documentation

4.37.2.1 _ui_container_destroy()

```
void _ui_container_destroy (
    UIElement * self )
```

4.37.2.2 _ui_container_recalculate()

```
void _ui_container_recalculate (
    UIElement * sibling,
    UIElement * self )
```

4.37.2.3 _ui_container_render()

```
void _ui_container_render (
    UIElement * self )
```

4.37.2.4 _ui_container_update()

```
void _ui_container_update (
    UIElement * self )
```

4.37.2.5 ui_create_button()

```
UIButton * ui_create_button (
    UIContainer * parent,
    UIConstraints constraints,
    const char * text,
    Color color,
    Color text_color,
    void(*) (UIButton *self) on_click )
```

4.37.2.6 ui_create_checkbox()

```
UICheckbox * ui_create_checkbox (
    UIContainer * parent,
    UIConstraints constraints,
    Color checked_color,
    Color unchecked_color,
    void(*) (UICheckbox *self, bool checked) on_checked_changed )
```

4.37.2.7 ui_create_container()

```
UIContainer * ui_create_container (
    UIContainer * parent,
    UIConstraints constraints,
    void(*) (UIContainer *self, SDL_Point size) on_size_changed )
```

4.37.2.8 ui_create_dropdown()

```
UIDropdownList * ui_create_dropdown (
    UIContainer * parent,
    UIConstraints constraints,
    char * items,
    Color color,
    Color text_color,
    void(*) (UIDropdownList *self, Sint32 index) on_selection_changed )
```

4.37.2.9 ui_create_imagebutton()

```
UIImageButton * ui_create_imagebutton (
    UIContainer * parent,
    UIConstraints constraints,
    Texture * texture,
    void(*) (UIImageButton *self) on_click )
```

4.37.2.10 ui_create_label()

```
UILabel * ui_create_label (
    UIContainer * parent,
    UIConstraints constraints,
    const char * text,
    Color color )
```

4.37.2.11 ui_create_panel()

```
UIPanel * ui_create_panel (
    UIContainer * parent,
    UIConstraints constraints,
    Color color,
    Color border_color,
    Uint32 border_width,
    Uint32 roundness )
```

4.37.2.12 ui_create_slider()

```
UISlider * ui_create_slider (
    UIContainer * parent,
    UIConstraints constraints,
    double value,
    Color color,
    Color slider_color,
    void(*) (UISlider *self, double value) on_value_changed )
```

4.37.2.13 ui_create_splitbutton()

```
UISplitButton * ui_create_splitbutton (
    UIContainer * parent,
    UIConstraints constraints,
    char * items,
    Color color,
    Color text_color,
    void(*) (UISplitButton *self, Sint32 index) on_item_clicked,
    bool auto_dropdown )
```

4.37.2.14 ui_create_textbox()

```
UITextbox * ui_create_textbox (
    UIContainer * parent,
    UIConstraints constraints,
    const char * text,
    Color color,
    Color text_color,
    void(*) (UITextbox *self, const char *text) on_text_changed )
```

4.38 src/ui/ui_element/ui_element.h File Reference

```
#include "../ui_constraint/ui_constraint.h"
#include "../../utils/vector/vector.h"
#include "../../color/color.h"
#include "../../texture/texture.h"
#include <stdbool.h>
```

Classes

- struct [UIElement](#)
- struct [UIContainer](#)
- struct [UIPanel](#)
- struct [UILabel](#)
- struct [UIButton](#)
- struct [UIImageButton](#)
- struct [UITextbox](#)
- struct [UICheckbox](#)
- struct [UISlider](#)
- struct [UIDropdownList](#)
- struct [UISplitButton](#)

Macros

- #define [UITEXT_MAX_LENGTH](#) 50

Typedefs

- typedef struct [UIElement](#) [UIElement](#)
- typedef struct [UIContainer](#) [UIContainer](#)
- typedef struct [UIPanel](#) [UIPanel](#)
- typedef struct [UILabel](#) [UILabel](#)
- typedef enum [MouseState](#) [MouseState](#)
- typedef struct [UIButton](#) [UIButton](#)
- typedef struct [UIImageButton](#) [UIImageButton](#)
- typedef struct [UITextbox](#) [UITextbox](#)
- typedef struct [UICheckbox](#) [UICheckbox](#)
- typedef struct [UISlider](#) [UISlider](#)
- typedef struct [UIDropdownList](#) [UIDropdownList](#)
- typedef struct [UISplitButton](#) [UISplitButton](#)

Enumerations

- enum [MouseState](#) { [MS_NONE](#) = 0 , [MS_HOVER](#) , [MS_PRESS](#) }

Functions

- [UIContainer](#) * [ui_create_container](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, void(*on_size_changed)([UIContainer](#) *self, [SDL_Point](#) size))
- [UIPanel](#) * [ui_create_panel](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, [Color](#) color, [Color](#) border_color, [Uint32](#) border_width, [Uint32](#) roundness)
- [UILabel](#) * [ui_create_label](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, const char *text, [Color](#) color)
- [UIButton](#) * [ui_create_button](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, const char *text, [Color](#) color, [Color](#) text_color, void(*on_click)([UIButton](#) *self))
- [UIImageButton](#) * [ui_create_imagebutton](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, [Texture](#) *texture, void(*on_click)([UIImageButton](#) *self))
- [UITextbox](#) * [ui_create_textbox](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, const char *text, [Color](#) color, [Color](#) text_color, void(*on_text_changed)([UITextbox](#) *self, const char *text))
- [UICheckbox](#) * [ui_create_checkbox](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, [Color](#) checked_color, [Color](#) unchecked_color, void(*on_checked_changed)([UICheckbox](#) *self, bool checked))
- [UISlider](#) * [ui_create_slider](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, double value, [Color](#) color, [Color](#) slider_color, void(*on_value_changed)([UISlider](#) *self, double value))
- [UIDropdownList](#) * [ui_create_dropdown](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, char *items, [Color](#) color, [Color](#) text_color, void(*on_selection_changed)([UIDropdownList](#) *self, [Sint32](#) index))
- [UISplitButton](#) * [ui_create_splitbutton](#) ([UIContainer](#) *parent, [UIConstraints](#) constraints, char *items, [Color](#) color, [Color](#) text_color, void(*on_item_clicked)([UISplitButton](#) *self, [Sint32](#) index), bool auto_dropdown)
- void [_ui_container_update](#) ([UIElement](#) *self)
- void [_ui_container_recalculate](#) ([UIElement](#) *sibling, [UIElement](#) *self)
- void [_ui_container_render](#) ([UIElement](#) *self)
- void [_ui_container_destroy](#) ([UIElement](#) *self)

4.38.1 Macro Definition Documentation

4.38.1.1 UITEXT_MAX_LENGTH

```
#define UITEXT_MAX_LENGTH 50
```

4.38.2 Typedef Documentation

4.38.2.1 MouseState

```
typedef enum MouseState MouseState
```

4.38.2.2 UIButton

```
typedef struct UIButton UIButton
```

4.38.2.3 UICheckbox

```
typedef struct UICheckbox UICheckbox
```

4.38.2.4 UIContainer

```
typedef struct UIContainer UIContainer
```

4.38.2.5 UIDropdownList

```
typedef struct UIDropdownList UIDropdownList
```

4.38.2.6 UIElement

```
typedef struct UIElement UIElement
```

4.38.2.7 UIImageButton

```
typedef struct UIImageButton UIImageButton
```

4.38.2.8 UILabel

```
typedef struct UILabel UILabel
```

4.38.2.9 UIPanel

```
typedef struct UIPanel UIPanel
```

4.38.2.10 UISlider

```
typedef struct UISlider UISlider
```


4.38.2.11 UISplitButton

```
typedef struct UISplitButton UISplitButton
```

4.38.2.12 UITextbox

```
typedef struct UITextbox UITextbox
```

4.38.3 Enumeration Type Documentation

4.38.3.1 MouseState

```
enum MouseState
```

Enumerator

MS_NONE	
MS_HOVER	
MS_PRESS	

4.38.4 Function Documentation

4.38.4.1 _ui_container_destroy()

```
void _ui_container_destroy (  
    UIElement * self )
```

4.38.4.2 _ui_container_recalculate()

```
void _ui_container_recalculate (  
    UIElement * sibling,  
    UIElement * self )
```

4.38.4.3 _ui_container_render()

```
void _ui_container_render (  
    UIElement * self )
```

4.38.4.4 _ui_container_update()

```
void _ui_container_update (  
    UIElement * self )
```

4.38.4.5 ui_create_button()

```
UIButton * ui_create_button (
    UIContainer * parent,
    UIConstraints constraints,
    const char * text,
    Color color,
    Color text_color,
    void(*) (UIButton *self) on_click )
```

4.38.4.6 ui_create_checkbox()

```
UICheckbox * ui_create_checkbox (
    UIContainer * parent,
    UIConstraints constraints,
    Color checked_color,
    Color unchecked_color,
    void(*) (UICheckbox *self, bool checked) on_checked_changed )
```

4.38.4.7 ui_create_container()

```
UIContainer * ui_create_container (
    UIContainer * parent,
    UIConstraints constraints,
    void(*) (UIContainer *self, SDL_Point size) on_size_changed )
```

4.38.4.8 ui_create_dropdown()

```
UIDropdownList * ui_create_dropdown (
    UIContainer * parent,
    UIConstraints constraints,
    char * items,
    Color color,
    Color text_color,
    void(*) (UIDropdownList *self, Sint32 index) on_selection_changed )
```

4.38.4.9 ui_create_imagebutton()

```
UIImageButton * ui_create_imagebutton (
    UIContainer * parent,
    UIConstraints constraints,
    Texture * texture,
    void(*) (UIImageButton *self) on_click )
```

4.38.4.10 ui_create_label()

```
UILabel * ui_create_label (
    UIContainer * parent,
    UIConstraints constraints,
    const char * text,
    Color color )
```

4.38.4.11 ui_create_panel()

```
UIPanel * ui_create_panel (
    UIContainer * parent,
    UIConstraints constraints,
    Color color,
    Color border_color,
    Uint32 border_width,
    Uint32 roundness )
```

4.38.4.12 ui_create_slider()

```
UISlider * ui_create_slider (
    UIContainer * parent,
    UIConstraints constraints,
    double value,
    Color color,
    Color slider_color,
    void(*) (UISlider *self, double value) on_value_changed )
```

4.38.4.13 ui_create_splitbutton()

```
UISplitButton * ui_create_splitbutton (
    UIContainer * parent,
    UIConstraints constraints,
    char * items,
    Color color,
    Color text_color,
    void(*) (UISplitButton *self, Sint32 index) on_item_clicked,
    bool auto_dropdown )
```

4.38.4.14 ui_create_textbox()

```
UITextbox * ui_create_textbox (
    UIContainer * parent,
    UIConstraints constraints,
    const char * text,
    Color color,
    Color text_color,
    void(*) (UITextbox *self, const char *text) on_text_changed )
```

4.39 ui_element.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #ifdef _WIN32
00004     #include <SDL.h>
00005 #elif defined(__unix__) || defined(__linux__)
00006     #include <SDL2/SDL.h>
00007 #endif
00008
00009 #include "../ui_constraint/ui_constraint.h"
```

```

00010 #include "../utils/vector/vector.h"
00011 #include "../color/color.h"
00012 #include "../texture/texture.h"
00013
00014 #include <stdbool.h>
00015
00016 #define UITEXT_MAX_LENGTH 50
00017
00018 typedef struct UIElement UIElement;
00019 typedef struct UIElement
00020 {
00021     UIElement* parent;
00022     UIConstraints constraints;
00023     SDL_Point position;
00024     SDL_Point size;
00025
00026     void (*update)(UIElement* self);
00027     void (*recalculate)(UIElement* sibling, UIElement* self);
00028     void (*render)(UIElement* self);
00029     void (*destroy)(UIElement* self);
00030 } UIElement;
00031
00032 typedef struct UIContainer UIContainer;
00033 typedef struct UIContainer
00034 {
00035     UIElement base;
00036
00037     Vector* children;
00038     void (*on_size_changed)(UIContainer* self, SDL_Point size);
00039 } UIContainer;
00040
00041 typedef struct UIPanel
00042 {
00043     UIElement base;
00044
00045     Color color;
00046     Color border_color;
00047     Uint32 border_width;
00048     Uint32 corner_radius;
00049 } UIPanel;
00050
00051 typedef struct UILabel
00052 {
00053     UIElement base;
00054
00055     char text[UITEXT_MAX_LENGTH + 1];
00056     Color color;
00057 } UILabel;
00058
00059 typedef enum MouseState { MS_NONE = 0, MS_HOVER, MS_PRESS } MouseState;
00060 typedef struct UIButton UIButton;
00061 typedef struct UIButton
00062 {
00063     UIElement base;
00064
00065     char text[UITEXT_MAX_LENGTH + 1];
00066     SDL_Point text_position;
00067     Color color;
00068     Color text_color;
00069     Uint32 corner_radius;
00070     MouseState mouse_state;
00071     void (*on_click)(UIButton* self);
00072 } UIButton;
00073
00074 typedef struct UIImageButton UIImageButton;
00075 typedef struct UIImageButton
00076 {
00077     UIElement base;
00078
00079     Texture* texture;
00080     MouseState mouse_state;
00081     void (*on_click)(UIImageButton* self);
00082 } UIImageButton;
00083
00084 typedef struct UITextbox UITextbox;
00085 typedef struct UITextbox
00086 {
00087     UIElement base;
00088
00089     char text[UITEXT_MAX_LENGTH + 1];
00090     Color color;
00091     Color text_color;
00092     Uint32 corner_radius;
00093     bool focused;
00094     MouseState mouse_state;
00095     void (*on_text_changed)(UITextbox* self, const char* text);
00096 } UITextbox;

```

```

00097
00098 typedef struct UICheckbox UICheckbox;
00099 typedef struct UICheckbox
00100 {
00101     UIElement base;
00102
00103     bool checked;
00104     Color checked_color;
00105     Color unchecked_color;
00106     Uint32 corner_radius;
00107     MouseState mouse_state;
00108     void (*on_checked_changed)(UICheckbox* self, bool checked);
00109 } UICheckbox;
00110
00111 typedef struct UISlider UISlider;
00112 typedef struct UISlider
00113 {
00114     UIElement base;
00115
00116     double value;
00117     Color color;
00118     Color slider_color;
00119     Uint32 thickness;
00120     Uint32 corner_radius;
00121     MouseState mouse_state;
00122     void (*on_value_changed)(UISlider* self, double value);
00123 } UISlider;
00124
00125 typedef struct UIDropdownList UIDropdownList;
00126 typedef struct UIDropdownList
00127 {
00128     UIElement base;
00129
00130     Vector* items;
00131     Uint32 selected_item;
00132     bool expanded;
00133     Color color;
00134     Color text_color;
00135     Uint32 corner_radius;
00136     void (*on_selection_changed)(UIDropdownList* self, Sint32 index);
00137 } UIDropdownList;
00138
00139 typedef struct UISplitButton UISplitButton;
00140 typedef struct UISplitButton
00141 {
00142     UIElement base;
00143
00144     Vector* items;
00145     bool expanded;
00146     Color color;
00147     Color text_color;
00148     Uint32 corner_radius;
00149     void (*on_item_clicked)(UISplitButton* self, Sint32 index);
00150     bool auto_dropdown;
00151 } UISplitButton;
00152
00153 //API functions
00154 UIContainer* ui_create_container(UIContainer* parent, UIConstraints constraints, void
(*on_size_changed)(UIContainer* self, SDL_Point size));
00155 UIPanel* ui_create_panel(UIContainer* parent, UIConstraints constraints, Color color, Color
border_color, Uint32 border_width, Uint32 roundness);
00156 UILabel* ui_create_label(UIContainer* parent, UIConstraints constraints, const char* text, Color
color);
00157 UIButton* ui_create_button(UIContainer* parent, UIConstraints constraints, const char* text, Color
color, Color text_color, void (*on_click)(UIButton* self));
00158 UIImageButton* ui_create_imagebutton(UIContainer* parent, UIConstraints constraints, Texture* texture,
void (*on_click)(UIImageButton* self));
00159 UITextbox* ui_create_textbox(UIContainer* parent, UIConstraints constraints, const char* text, Color
color, Color text_color, void (*on_text_changed)(UITextbox* self, const char* text));
00160 UICheckbox* ui_create_checkbox(UIContainer* parent, UIConstraints constraints, Color checked_color,
Color unchecked_color, void (*on_checked_changed)(UICheckbox* self, bool checked));
00161 UISlider* ui_create_slider(UIContainer* parent, UIConstraints constraints, double value, Color color,
Color slider_color, void (*on_value_changed)(UISlider* self, double value));
00162 UIDropdownList* ui_create_dropdown(UIContainer* parent, UIConstraints constraints, char* items, Color
color, Color text_color, void (*on_selection_changed)(UIDropdownList* self, Sint32 index));
00163 UISplitButton* ui_create_splitbutton(UIContainer* parent, UIConstraints constraints, char* items,
Color color, Color text_color, void (*on_item_clicked)(UISplitButton* self, Sint32 index), bool
auto_dropdown);
00164
00165 //internal functions
00166 void _ui_container_update(UIElement* self);
00167 void _ui_container_recalculate(UIElement* sibling, UIElement* self);
00168 void _ui_container_render(UIElement* self);
00169 void _ui_container_destroy(UIElement* self);

```

4.40 src/utils/math/math.c File Reference

```
#include "math.h"
```

Functions

- double [deg_to_rad](#) (double deg)
- double [rad_to_deg](#) (double rad)
- double [clamp](#) (double value, double min, double max)
- double [lerp](#) (double a, double b, double t)
- double [map](#) (double x, double min1, double max1, double min2, double max2)
- bool [check_collision_point_rect](#) (int px, int py, int rx, int ry, int rw, int rh)

4.40.1 Function Documentation

4.40.1.1 [check_collision_point_rect\(\)](#)

```
bool check_collision_point_rect (
    int px,
    int py,
    int rx,
    int ry,
    int rw,
    int rh )
```

4.40.1.2 [clamp\(\)](#)

```
double clamp (
    double value,
    double min,
    double max )
```

4.40.1.3 [deg_to_rad\(\)](#)

```
double deg_to_rad (
    double deg )
```

4.40.1.4 [lerp\(\)](#)

```
double lerp (
    double a,
    double b,
    double t )
```

4.40.1.5 map()

```
double map (
    double x,
    double min1,
    double max1,
    double min2,
    double max2 )
```

4.40.1.6 rad_to_deg()

```
double rad_to_deg (
    double rad )
```

4.41 src/utls/math/math.h File Reference

```
#include <stdbool.h>
```

Macros

- #define [PI](#) 3.14159265358979323846
- #define [TWO_PI](#) 6.28318530717958647692
- #define [HALF_PI](#) 1.57079632679489661923

Functions

- double [deg_to_rad](#) (double deg)
- double [rad_to_deg](#) (double rad)
- double [clamp](#) (double x, double min, double max)
- double [lerp](#) (double a, double b, double t)
- double [map](#) (double x, double min1, double max1, double min2, double max2)
- bool [check_collision_point_rect](#) (int px, int py, int rx, int ry, int rw, int rh)

4.41.1 Macro Definition Documentation

4.41.1.1 HALF_PI

```
#define HALF_PI 1.57079632679489661923
```

4.41.1.2 PI

```
#define PI 3.14159265358979323846
```

4.41.1.3 TWO_PI

```
#define TWO_PI 6.28318530717958647692
```

4.41.2 Function Documentation

4.41.2.1 check_collision_point_rect()

```
bool check_collision_point_rect (
    int px,
    int py,
    int rx,
    int ry,
    int rw,
    int rh )
```

4.41.2.2 clamp()

```
double clamp (
    double x,
    double min,
    double max )
```

4.41.2.3 deg_to_rad()

```
double deg_to_rad (
    double deg )
```

4.41.2.4 lerp()

```
double lerp (
    double a,
    double b,
    double t )
```

4.41.2.5 map()

```
double map (
    double x,
    double min1,
    double max1,
    double min2,
    double max2 )
```

4.41.2.6 rad_to_deg()

```
double rad_to_deg (
    double rad )
```


4.42 math.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include <stdbool.h>
00004
00005 #define PI 3.14159265358979323846
00006 #define TWO_PI 6.28318530717958647692
00007 #define HALF_PI 1.57079632679489661923
00008
00009 double deg_to_rad(double deg);
00010 double rad_to_deg(double rad);
00011 double clamp(double x, double min, double max);
00012 double lerp(double a, double b, double t);
00013 double map(double x, double min1, double max1, double min2, double max2);
00014
00015 bool check_collision_point_rect(int px, int py, int rx, int ry, int rw, int rh);
```

4.43 src/utils/vector/vector.c File Reference

```
#include "vector.h"
#include <stdio.h>
```

Functions

- [Vector](#) * [vector_create](#) (size_t capacity)
- void [vector_free](#) ([Vector](#) *vector)
- void * [vector_get](#) ([Vector](#) *vector, size_t idx)
- void [vector_push_back](#) ([Vector](#) *vector, void *value)
- void [vector_pop_back](#) ([Vector](#) *vector)
- void [vector_insert](#) ([Vector](#) *vector, size_t idx, void *value)
- bool [vector_contains](#) ([Vector](#) *vector, void *value)
- void [vector_remove_at](#) ([Vector](#) *vector, size_t idx)
- void [vector_remove](#) ([Vector](#) *vector, void *value)
- void [vector_reserve](#) ([Vector](#) *vector, size_t capacity)
- size_t [vector_size](#) ([Vector](#) *vector)
- void [vector_clear](#) ([Vector](#) *vector)

4.43.1 Function Documentation

4.43.1.1 [vector_clear\(\)](#)

```
void vector_clear (
    Vector * vector )
```

4.43.1.2 [vector_contains\(\)](#)

```
bool vector_contains (
    Vector * vector,
    void * value )
```

4.43.1.3 vector_create()

```
Vector * vector_create (
    size_t capacity )
```

4.43.1.4 vector_free()

```
void vector_free (
    Vector * vector )
```

4.43.1.5 vector_get()

```
void * vector_get (
    Vector * vector,
    size_t idx )
```

4.43.1.6 vector_insert()

```
void vector_insert (
    Vector * vector,
    size_t idx,
    void * value )
```

4.43.1.7 vector_pop_back()

```
void vector_pop_back (
    Vector * vector )
```

4.43.1.8 vector_push_back()

```
void vector_push_back (
    Vector * vector,
    void * value )
```

4.43.1.9 vector_remove()

```
void vector_remove (
    Vector * vector,
    void * value )
```

4.43.1.10 vector_remove_at()

```
void vector_remove_at (
    Vector * vector,
    size_t idx )
```

4.43.1.11 vector_reserve()

```
void vector_reserve (
    Vector * vector,
    size_t capacity )
```

4.43.1.12 vector_size()

```
size_t vector_size (
    Vector * vector )
```

4.44 src/utls/vector/vector.h File Reference

```
#include <stdlib.h>
#include <stdbool.h>
```

Classes

- struct [Vector](#)

Typedefs

- typedef struct [Vector](#) [Vector](#)

Functions

- [Vector](#) * [vector_create](#) (size_t capacity)
- void [vector_free](#) ([Vector](#) *vector)
- void * [vector_get](#) ([Vector](#) *vector, size_t idx)
- void [vector_push_back](#) ([Vector](#) *vector, void *value)
- void [vector_pop_back](#) ([Vector](#) *vector)
- void [vector_insert](#) ([Vector](#) *vector, size_t idx, void *value)
- bool [vector_contains](#) ([Vector](#) *vector, void *value)
- void [vector_remove_at](#) ([Vector](#) *vector, size_t idx)
- void [vector_remove](#) ([Vector](#) *vector, void *value)
- void [vector_reserve](#) ([Vector](#) *vector, size_t capacity)
- size_t [vector_size](#) ([Vector](#) *vector)
- void [vector_clear](#) ([Vector](#) *vector)

4.44.1 Typedef Documentation

4.44.1.1 Vector

```
typedef struct Vector Vector
```

4.44.2 Function Documentation

4.44.2.1 vector_clear()

```
void vector_clear (
    Vector * vector )
```

4.44.2.2 vector_contains()

```
bool vector_contains (
    Vector * vector,
    void * value )
```

4.44.2.3 vector_create()

```
Vector * vector_create (
    size_t capacity )
```

4.44.2.4 vector_free()

```
void vector_free (
    Vector * vector )
```

4.44.2.5 vector_get()

```
void * vector_get (
    Vector * vector,
    size_t idx )
```

4.44.2.6 vector_insert()

```
void vector_insert (
    Vector * vector,
    size_t idx,
    void * value )
```

4.44.2.7 vector_pop_back()

```
void vector_pop_back (
    Vector * vector )
```

4.44.2.8 vector_push_back()

```
void vector_push_back (
    Vector * vector,
    void * value )
```

4.44.2.9 vector_remove()

```
void vector_remove (
    Vector * vector,
    void * value )
```

4.44.2.10 vector_remove_at()

```
void vector_remove_at (
    Vector * vector,
    size_t idx )
```

4.44.2.11 vector_reserve()

```
void vector_reserve (
    Vector * vector,
    size_t capacity )
```

4.44.2.12 vector_size()

```
size_t vector_size (
    Vector * vector )
```

4.45 vector.h

[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include <stdlib.h>
00004 #include <stdbool.h>
00005
00006 typedef struct Vector {
00007     size_t capacity;
00008     size_t size;
00009     void** data;
00010 } Vector;
00011
00012 Vector* vector_create(size_t capacity);
00013 void vector_free(Vector* vector);
00014 void* vector_get(Vector* vector, size_t idx);
00015 void vector_push_back(Vector* vector, void* value);
00016 void vector_pop_back(Vector* vector);
00017 void vector_insert(Vector* vector, size_t idx, void* value);
00018 bool vector_contains(Vector* vector, void* value);
00019 void vector_remove_at(Vector* vector, size_t idx);
00020 void vector_remove(Vector* vector, void* value);
00021 void vector_reserve(Vector* vector, size_t capacity);
00022 size_t vector_size(Vector* vector);
00023 void vector_clear(Vector* vector);
```

4.46 src/window/window.c File Reference

```
#include <assert.h>
#include "window.h"
#include "../app/app.h"
#include "../input/input.h"
#include "../renderer/renderer.h"
```

Functions

- `Window * window_create` (const char *title, int width, int height, int flags)
- void `window_show` (Window *window)
- void `window_hide` (Window *window)
- void `window_focus` (Window *window)
- UIContainer * `window_get_main_container` (Window *window)
- void `_window_reset` (Window *window)
- void `_window_handle_event` (Window *window, SDL_Event *event)
- void `_window_update` (Window *window)
- void `_window_render` (Window *window)
- void `_window_close` (Window *window)

4.46.1 Function Documentation

4.46.1.1 `_window_close()`

```
void _window_close (  
    Window * window )
```

4.46.1.2 `_window_handle_event()`

```
void _window_handle_event (  
    Window * window,  
    SDL_Event * event )
```

4.46.1.3 `_window_render()`

```
void _window_render (  
    Window * window )
```

4.46.1.4 `_window_reset()`

```
void _window_reset (  
    Window * window )
```

4.46.1.5 `_window_update()`

```
void _window_update (  
    Window * window )
```

4.46.1.6 `window_create()`

```
Window * window_create (  
    const char * title,  
    int width,  
    int height,  
    int flags )
```

4.46.1.7 window_focus()

```
void window_focus (  
    Window * window )
```

4.46.1.8 window_get_main_container()

```
UIContainer * window_get_main_container (  
    Window * window )
```

4.46.1.9 window_hide()

```
void window_hide (  
    Window * window )
```

4.46.1.10 window_show()

```
void window_show (  
    Window * window )
```

4.47 src/window/window.h File Reference

```
#include <stdbool.h>  
#include "../input/input.h"  
#include "../ui/ui.h"  
#include "../ui/ui_element/ui_element.h"
```

Classes

- struct [Window](#)

Typedefs

- typedef struct [Window](#) [Window](#)

Functions

- [Window](#) * [window_create](#) (const char *title, int width, int height, int flags)
- void [window_show](#) ([Window](#) *window)
- void [window_hide](#) ([Window](#) *window)
- void [window_focus](#) ([Window](#) *window)
- [UIContainer](#) * [window_get_main_container](#) ([Window](#) *window)
- void [_window_reset](#) ([Window](#) *window)
- void [_window_handle_event](#) ([Window](#) *window, [SDL_Event](#) *event)
- void [_window_update](#) ([Window](#) *window)
- void [_window_render](#) ([Window](#) *window)
- void [_window_close](#) ([Window](#) *window)

4.47.1 Typedef Documentation

4.47.1.1 Window

```
typedef struct Window Window
```

4.47.2 Function Documentation

4.47.2.1 _window_close()

```
void _window_close (
    Window * window )
```

4.47.2.2 _window_handle_event()

```
void _window_handle_event (
    Window * window,
    SDL_Event * event )
```

4.47.2.3 _window_render()

```
void _window_render (
    Window * window )
```

4.47.2.4 _window_reset()

```
void _window_reset (
    Window * window )
```

4.47.2.5 _window_update()

```
void _window_update (
    Window * window )
```

4.47.2.6 window_create()

```
Window * window_create (
    const char * title,
    int width,
    int height,
    int flags )
```

4.47.2.7 window_focus()

```
void window_focus (
    Window * window )
```


4.47.2.8 window_get_main_container()

```
UIContainer * window_get_main_container (
    Window * window )
```

4.47.2.9 window_hide()

```
void window_hide (
    Window * window )
```

4.47.2.10 window_show()

```
void window_show (
    Window * window )
```

4.48 window.h

[Go to the documentation of this file.](#)

```
00001 #ifndef WINDOW_H
00002 #define WINDOW_H
00003
00004 #ifdef _WIN32
00005     #include <SDL.h>
00006 #elif defined(__unix__) || defined(__linux__)
00007     #include <SDL2/SDL.h>
00008 #endif
00009
00010 #include <stdbool.h>
00011 #include "../input/input.h"
00012 #include "../ui/ui.h"
00013 #include "../ui/ui_element/ui_element.h"
00014
00015 typedef struct Window
00016 {
00017     SDL_Window* window;
00018     SDL_Renderer* renderer;
00019     InputData input_data;
00020     UIData ui_data;
00021     bool close_requested;
00022 } Window;
00023
00024 Window* window_create(const char* title, int width, int height, int flags);
00025 void window_show(Window* window);
00026 void window_hide(Window* window);
00027 void window_focus(Window* window);
00028 UIContainer* window_get_main_container(Window* window);
00029
00030 //API functions
00031 void _window_reset(Window* window);
00032 void _window_handle_event(Window* window, SDL_Event* event);
00033 void _window_update(Window* window);
00034 void _window_render(Window* window);
00035 void _window_close(Window* window);
00036
00037 #endif
```


Index

- [_UIDropdownItem](#), 5
 - [base](#), 5
 - [dropdown_index](#), 5
 - [mouse_state](#), 5
 - [parent_dropdown](#), 5
 - [text](#), 5
 - [ui_element.c](#), 95
- [UISplitButtonItem](#), 6
 - [base](#), 6
 - [mouse_state](#), 6
 - [parent_splitbutton](#), 6
 - [splitbutton_index](#), 6
 - [text](#), 6
 - [ui_element.c](#), 95
- [_app_add_window](#)
 - [app.c](#), 31
 - [app.h](#), 33
- [_font_close](#)
 - [font.c](#), 41
 - [font.h](#), 42
- [_font_init](#)
 - [font.c](#), 41
 - [font.h](#), 42
- [_input_close](#)
 - [input.c](#), 65
 - [input.h](#), 67
- [_input_handle_event](#)
 - [input.c](#), 65
 - [input.h](#), 67
- [_input_init](#)
 - [input.c](#), 65
 - [input.h](#), 67
- [_input_reset](#)
 - [input.c](#), 65
 - [input.h](#), 68
- [_input_set_target](#)
 - [input.c](#), 65
 - [input.h](#), 68
- [_renderer_set_target](#)
 - [renderer.c](#), 74
 - [renderer.h](#), 80
- [_texture_add](#)
 - [texture.c](#), 86
 - [texture.h](#), 87
- [_texture_close](#)
 - [texture.c](#), 86
 - [texture.h](#), 87
- [_texture_init](#)
 - [texture.c](#), 86
 - [texture.h](#), 87
- [_ui_close](#)
 - [ui.c](#), 89
 - [ui.h](#), 90
- [_ui_container_destroy](#)
 - [ui_element.c](#), 96
 - [ui_element.h](#), 101
- [_ui_container_recalculate](#)
 - [ui_element.c](#), 96
 - [ui_element.h](#), 101
- [_ui_container_render](#)
 - [ui_element.c](#), 96
 - [ui_element.h](#), 101
- [_ui_container_update](#)
 - [ui_element.c](#), 96
 - [ui_element.h](#), 101
- [_ui_get_target](#)
 - [ui.c](#), 89
 - [ui.h](#), 90
- [_ui_handle_event](#)
 - [ui.c](#), 89
 - [ui.h](#), 90
- [_ui_init](#)
 - [ui.c](#), 89
 - [ui.h](#), 90
- [_ui_render](#)
 - [ui.c](#), 89
 - [ui.h](#), 91
- [_ui_set_target](#)
 - [ui.c](#), 89
 - [ui.h](#), 91
- [_ui_update](#)
 - [ui.c](#), 89
 - [ui.h](#), 91
- [_window_close](#)
 - [window.c](#), 114
 - [window.h](#), 116
- [_window_handle_event](#)
 - [window.c](#), 114
 - [window.h](#), 116
- [_window_render](#)
 - [window.c](#), 114
 - [window.h](#), 116
- [_window_reset](#)
 - [window.c](#), 114
 - [window.h](#), 116
- [_window_update](#)
 - [window.c](#), 114
 - [window.h](#), 116

- app.c
 - _app_add_window, 31
 - app_close, 31
 - app_data, 33
 - app_get_delta_time, 32
 - app_get_time, 32
 - app_get_windows, 32
 - app_init, 32
 - app_render, 32
 - app_request_close, 32
 - app_set_target, 32
 - app_set_target_fps, 32
 - app_update, 32
- app.h
 - _app_add_window, 33
 - app_close, 33
 - app_get_delta_time, 34
 - app_get_time, 34
 - app_get_windows, 34
 - app_init, 34
 - app_render, 34
 - app_request_close, 34
 - app_set_target, 34
 - app_set_target_fps, 34
 - app_update, 34
 - AppData, 33
- app_close
 - app.c, 31
 - app.h, 33
- app_data
 - app.c, 33
- app_get_delta_time
 - app.c, 32
 - app.h, 34
- app_get_time
 - app.c, 32
 - app.h, 34
- app_get_windows
 - app.c, 32
 - app.h, 34
- app_init
 - app.c, 32
 - app.h, 34
- app_render
 - app.c, 32
 - app.h, 34
- app_request_close
 - app.c, 32
 - app.h, 34
- app_set_target
 - app.c, 32
 - app.h, 34
- app_set_target_fps
 - app.c, 32
 - app.h, 34
- app_update
 - app.c, 32
 - app.h, 34
- AppData, 7
 - app.h, 33
 - delta_time, 7
 - frame_start, 7
 - last_frame_start, 7
 - target_frame_time, 7
 - windows, 7
- auto_dropdown
 - UISplitButton, 25
- backspace_pressed
 - UIData, 18
- base
 - _UIDropdownItem, 5
 - _UISplitButtonItem, 6
 - Circle, 8
 - Line, 12
 - Point, 12
 - UIButton, 14
 - UICheckbox, 15
 - UIContainer, 17
 - UIDropdownList, 19
 - UIImageButton, 21
 - UILabel, 22
 - UIPanel, 23
 - UISlider, 24
 - UISplitButton, 25
 - UITextbox, 26
- BLACK
 - color.h, 37
- BLUE
 - color.h, 37
- border_color
 - UIPanel, 23
- border_width
 - UIPanel, 23
- capacity
 - Vector, 27
- center
 - Circle, 8
- check_collision_point_rect
 - math.c, 106
 - math.h, 108
- checked
 - UICheckbox, 15
- checked_color
 - UICheckbox, 15
- children
 - UIContainer, 17
- Circle, 7
 - base, 8
 - center, 8
 - perimeter_point, 8
 - shape.h, 54
- circle_create
 - shape.c, 52
 - shape.h, 54
- clamp

- math.c, 106
- math.h, 108
- close_requested
 - Window, 29
- Color
 - color.h, 38
- color
 - UIButton, 14
 - UIDropdownList, 19
 - UILabel, 22
 - UIPanel, 23
 - UISlider, 24
 - UISplitButton, 25
 - UITextbox, 26
- color.c
 - color_clever_shift, 35
 - color_fade, 35
 - color_from_grayscale, 36
 - color_from_hex, 36
 - color_from_hsv, 36
 - color_from_rgb, 36
 - color_from_rgba, 36
 - color_shift, 36
- color.h
 - BLACK, 37
 - BLUE, 37
 - Color, 38
 - color_clever_shift, 39
 - color_fade, 39
 - color_from_grayscale, 39
 - color_from_hex, 39
 - color_from_hsv, 39
 - color_from_rgb, 39
 - color_from_rgba, 39
 - color_shift, 39
 - CYAN, 37
 - DARK_GRAY, 37
 - GRAY, 38
 - GREEN, 38
 - MAGENTA, 38
 - RED, 38
 - TRANSPARENT, 38
 - WHITE, 38
 - YELLOW, 38
- color_clever_shift
 - color.c, 35
 - color.h, 39
- color_fade
 - color.c, 35
 - color.h, 39
- color_from_grayscale
 - color.c, 36
 - color.h, 39
- color_from_hex
 - color.c, 36
 - color.h, 39
- color_from_hsv
 - color.c, 36
- color.h, 39
- color_from_rgb
 - color.c, 36
 - color.h, 39
- color_from_rgba
 - color.c, 36
 - color.h, 39
- color_shift
 - color.c, 36
 - color.h, 39
- constraint_type
 - UIConstraint, 16
- constraints
 - UIElement, 20
- constraints_from_string
 - ui_constraint.c, 92
 - ui_constraint.h, 94
- ConstraintType
 - ui_constraint.h, 93
- coordinate_system.c
 - coordinate_system_create, 43
 - coordinate_system_destroy, 43
 - coordinate_system_draw, 44
 - coordinate_system_get_hovered_shape, 44
 - coordinate_system_is_hovered, 44
 - coordinate_system_translate, 44
 - coordinate_system_update_dimensions, 45
 - coordinate_system_zoom, 45
 - coordinates_to_screen, 45
 - screen_to_coordinates, 46
- coordinate_system.h
 - coordinate_system_create, 47
 - coordinate_system_destroy, 48
 - coordinate_system_draw, 48
 - coordinate_system_get_hovered_shape, 48
 - coordinate_system_is_hovered, 49
 - coordinate_system_translate, 49
 - coordinate_system_update_dimensions, 49
 - coordinate_system_zoom, 49
 - coordinates_to_screen, 50
 - CoordinateSystem, 47
 - INITIAL_ZOOM, 47
 - screen_to_coordinates, 50
- coordinate_system_create
 - coordinate_system.c, 43
 - coordinate_system.h, 47
- coordinate_system_destroy
 - coordinate_system.c, 43
 - coordinate_system.h, 48
- coordinate_system_draw
 - coordinate_system.c, 44
 - coordinate_system.h, 48
- coordinate_system_get_hovered_shape
 - coordinate_system.c, 44
 - coordinate_system.h, 48
- coordinate_system_is_hovered
 - coordinate_system.c, 44
 - coordinate_system.h, 49

- coordinate_system_translate
 - coordinate_system.c, [44](#)
 - coordinate_system.h, [49](#)
- coordinate_system_update_dimensions
 - coordinate_system.c, [45](#)
 - coordinate_system.h, [49](#)
- coordinate_system_zoom
 - coordinate_system.c, [45](#)
 - coordinate_system.h, [49](#)
- coordinates
 - Point, [12](#)
- coordinates_to_screen
 - coordinate_system.c, [45](#)
 - coordinate_system.h, [50](#)
- CoordinateSystem, [8](#)
 - coordinate_system.h, [47](#)
 - origin, [8](#)
 - position, [8](#)
 - shape.h, [54](#)
 - shapes, [9](#)
 - size, [9](#)
 - zoom, [9](#)
- corner_radius
 - UIButton, [14](#)
 - UICheckbox, [15](#)
 - UIDropdownList, [19](#)
 - UIPanel, [23](#)
 - UISlider, [24](#)
 - UISplitButton, [25](#)
 - UITextbox, [26](#)
- cs
 - main.c, [73](#)
- CT_ASPECT
 - ui_constraint.h, [93](#)
- CT_CENTER
 - ui_constraint.h, [93](#)
- CT_OFFSET
 - ui_constraint.h, [93](#)
- CT_PIXEL
 - ui_constraint.h, [93](#)
- CT_RELATIVE
 - ui_constraint.h, [93](#)
- current_keyboard_state
 - InputData, [10](#)
- current_mouse_button_state
 - InputData, [10](#)
- current_mouse_position
 - InputData, [10](#)
- CYAN
 - color.h, [37](#)
- DARK_GRAY
 - color.h, [37](#)
- data
 - Vector, [27](#)
- default_font
 - renderer.c, [79](#)
- deg_to_rad
 - math.c, [106](#)
 - math.h, [108](#)
- delta_time
 - AppData, [7](#)
- destroy
 - IShape, [11](#)
 - UIElement, [20](#)
- draw
 - IShape, [11](#)
- dropdown_index
 - _UIDropdownItem, [5](#)
- expanded
 - UIDropdownList, [19](#)
 - UISplitButton, [25](#)
- expanded_splitbutton
 - UIData, [18](#)
- focused
 - UITextbox, [26](#)
- Font, [9](#)
 - font, [9](#)
 - font.h, [41](#)
 - size, [9](#)
- font
 - Font, [9](#)
- font.c
 - _font_close, [41](#)
 - _font_init, [41](#)
 - font_load, [41](#)
 - fonts, [41](#)
- font.h
 - _font_close, [42](#)
 - _font_init, [42](#)
 - Font, [41](#)
 - font_load, [42](#)
- font_load
 - font.c, [41](#)
 - font.h, [42](#)
- fonts
 - font.c, [41](#)
- FPS
 - main.c, [71](#)
- frame_start
 - AppData, [7](#)
- GRAY
 - color.h, [38](#)
- GREEN
 - color.h, [38](#)
- HALF_PI
 - math.h, [107](#)
- height
 - Texture, [13](#)
 - UIConstraints, [17](#)
- INITIAL_ZOOM
 - coordinate_system.h, [47](#)
- input.c

- [_input_close](#), [65](#)
 - [_input_handle_event](#), [65](#)
 - [_input_init](#), [65](#)
 - [_input_reset](#), [65](#)
 - [_input_set_target](#), [65](#)
 - [input_get_mouse_motion](#), [65](#)
 - [input_get_mouse_position](#), [65](#)
 - [input_get_mouse_wheel_delta](#), [66](#)
 - [input_is_key_down](#), [66](#)
 - [input_is_key_pressed](#), [66](#)
 - [input_is_key_released](#), [66](#)
 - [input_is_mouse_button_down](#), [66](#)
 - [input_is_mouse_button_pressed](#), [66](#)
 - [input_is_mouse_button_released](#), [66](#)
 - [target_input_data](#), [66](#)
- [input.h](#)
 - [_input_close](#), [67](#)
 - [_input_handle_event](#), [67](#)
 - [_input_init](#), [67](#)
 - [_input_reset](#), [68](#)
 - [_input_set_target](#), [68](#)
 - [input_get_mouse_motion](#), [68](#)
 - [input_get_mouse_position](#), [68](#)
 - [input_get_mouse_wheel_delta](#), [68](#)
 - [input_is_key_down](#), [68](#)
 - [input_is_key_pressed](#), [68](#)
 - [input_is_key_released](#), [68](#)
 - [input_is_mouse_button_down](#), [68](#)
 - [input_is_mouse_button_pressed](#), [69](#)
 - [input_is_mouse_button_released](#), [69](#)
 - [InputData](#), [67](#)
- [input_data](#)
 - [Window](#), [29](#)
- [input_get_mouse_motion](#)
 - [input.c](#), [65](#)
 - [input.h](#), [68](#)
- [input_get_mouse_position](#)
 - [input.c](#), [65](#)
 - [input.h](#), [68](#)
- [input_get_mouse_wheel_delta](#)
 - [input.c](#), [66](#)
 - [input.h](#), [68](#)
- [input_is_key_down](#)
 - [input.c](#), [66](#)
 - [input.h](#), [68](#)
- [input_is_key_pressed](#)
 - [input.c](#), [66](#)
 - [input.h](#), [68](#)
- [input_is_key_released](#)
 - [input.c](#), [66](#)
 - [input.h](#), [68](#)
- [input_is_mouse_button_down](#)
 - [input.c](#), [66](#)
 - [input.h](#), [68](#)
- [input_is_mouse_button_pressed](#)
 - [input.c](#), [66](#)
 - [input.h](#), [69](#)
- [input_is_mouse_button_released](#)
 - [input.c](#), [66](#)
 - [input.h](#), [69](#)
- [InputData](#), [10](#)
 - [current_keyboard_state](#), [10](#)
 - [current_mouse_button_state](#), [10](#)
 - [current_mouse_position](#), [10](#)
 - [input.h](#), [67](#)
 - [key_count](#), [10](#)
 - [mouse_wheel_delta](#), [10](#)
 - [old_keyboard_state](#), [10](#)
 - [old_mouse_button_state](#), [10](#)
 - [old_mouse_position](#), [10](#)
- [is_defined_by](#)
 - [IShape](#), [11](#)
- [IShape](#), [11](#)
 - [destroy](#), [11](#)
 - [draw](#), [11](#)
 - [is_defined_by](#), [11](#)
 - [overlap_point](#), [11](#)
 - [shape.h](#), [54](#)
 - [translate](#), [11](#)
- [items](#)
 - [UIDropdownList](#), [19](#)
 - [UISplitButton](#), [25](#)
- [key_count](#)
 - [InputData](#), [10](#)
- [last_frame_start](#)
 - [AppData](#), [7](#)
- [lerp](#)
 - [math.c](#), [106](#)
 - [math.h](#), [108](#)
- [Line](#), [12](#)
 - [base](#), [12](#)
 - [p1](#), [12](#)
 - [p2](#), [12](#)
 - [shape.h](#), [54](#)
- [line_create](#)
 - [shape.c](#), [52](#)
 - [shape.h](#), [54](#)
- [MAGENTA](#)
 - [color.h](#), [38](#)
- [main](#)
 - [main.c](#), [72](#)
- [main.c](#)
 - [cs](#), [73](#)
 - [FPS](#), [71](#)
 - [main](#), [72](#)
 - [MOUSE_WHEEL_SENSITIVITY](#), [71](#)
 - [on_canvas_size_changed](#), [72](#)
 - [on_circle_clicked](#), [72](#)
 - [on_editmenu_clicked](#), [72](#)
 - [on_filemenu_clicked](#), [72](#)
 - [on_line_clicked](#), [72](#), [73](#)
 - [on_point_clicked](#), [73](#)
 - [on_pointer_clicked](#), [73](#)
 - [State](#), [71](#)

- state, 73
- STATE_CIRCLE, 71
- STATE_CIRCLE_CENTER_PLACED, 71
- STATE_CS_DRAGGED, 71
- STATE_LINE, 71
- STATE_LINE_POINT1_PLACED, 71
- STATE_POINT, 71
- STATE_POINTER, 71
- main_container
 - UIData, 18
- map
 - math.c, 106
 - math.h, 108
- math.c
 - check_collision_point_rect, 106
 - clamp, 106
 - deg_to_rad, 106
 - lerp, 106
 - map, 106
 - rad_to_deg, 107
- math.h
 - check_collision_point_rect, 108
 - clamp, 108
 - deg_to_rad, 108
 - HALF_PI, 107
 - lerp, 108
 - map, 108
 - PI, 107
 - rad_to_deg, 108
 - TWO_PI, 107
- mouse_captured
 - UIData, 18
- mouse_state
 - _UIDropdownItem, 5
 - _UISplitButtonItem, 6
 - UIButton, 14
 - UICheckbox, 15
 - UIImageButton, 21
 - UISlider, 24
 - UITextbox, 27
- mouse_wheel_delta
 - InputData, 10
- MOUSE_WHEEL_SENSITIVITY
 - main.c, 71
- MouseState
 - ui_element.h, 100, 101
- MS_HOVER
 - ui_element.h, 101
- MS_NONE
 - ui_element.h, 101
- MS_PRESS
 - ui_element.h, 101
- new_aspect_constraint
 - ui_constraint.c, 92
 - ui_constraint.h, 94
- new_center_constraint
 - ui_constraint.c, 92
 - ui_constraint.h, 94
- new_offset_constraint
 - ui_constraint.c, 92
 - ui_constraint.h, 94
- new_pixel_constraint
 - ui_constraint.c, 92
 - ui_constraint.h, 94
- new_relative_constraint
 - ui_constraint.c, 92
 - ui_constraint.h, 94
- old_keyboard_state
 - InputData, 10
- old_mouse_button_state
 - InputData, 10
- old_mouse_position
 - InputData, 10
- on_canvas_size_changed
 - main.c, 72
- on_checked_changed
 - UICheckbox, 15
- on_circle_clicked
 - main.c, 72
- on_click
 - UIButton, 14
 - UIImageButton, 22
- on_editmenu_clicked
 - main.c, 72
- on_filemenu_clicked
 - main.c, 72
- on_item_clicked
 - UISplitButton, 26
- on_line_clicked
 - main.c, 72, 73
- on_point_clicked
 - main.c, 73
- on_pointer_clicked
 - main.c, 73
- on_selection_changed
 - UIDropdownList, 19
- on_size_changed
 - UIContainer, 18
- on_text_changed
 - UITextbox, 27
- on_value_changed
 - UISlider, 24
- origin
 - CoordinateSystem, 8
- OVERLAP_DISTANCE
 - shape.h, 53
- overlap_point
 - IShape, 11
- p1
 - Line, 12
- p2
 - Line, 12
- parent
 - UIElement, 20
- parent_dropdown

- [_UIDropdownItem](#), 5
- [parent_splitbutton](#)
 - [_UISplitButtonItem](#), 6
- [perimeter_point](#)
 - [Circle](#), 8
- [PI](#)
 - [math.h](#), 107
- [Point](#), 12
 - [base](#), 12
 - [coordinates](#), 12
 - [shape.h](#), 54
- [point_create](#)
 - [shape.c](#), 52
 - [shape.h](#), 55
- [position](#)
 - [CoordinateSystem](#), 8
 - [UIElement](#), 20
- [rad_to_deg](#)
 - [math.c](#), 107
 - [math.h](#), 108
- [recalculate](#)
 - [UIConstraint](#), 16
 - [UIElement](#), 21
- [RED](#)
 - [color.h](#), 38
- [render](#)
 - [UIElement](#), 21
- [renderer](#)
 - [Window](#), 29
- [renderer.c](#)
 - [_renderer_set_target](#), 74
 - [default_font](#), 79
 - [renderer_bind_framebuffer](#), 74
 - [renderer_clear](#), 75
 - [renderer_create_framebuffer](#), 75
 - [renderer_draw_arc](#), 75
 - [renderer_draw_bezier](#), 75
 - [renderer_draw_circle](#), 75
 - [renderer_draw_ellipse](#), 75
 - [renderer_draw_filled_circle](#), 75
 - [renderer_draw_filled_ellipse](#), 76
 - [renderer_draw_filled_pie](#), 76
 - [renderer_draw_filled_polygon](#), 76
 - [renderer_draw_filled_rect](#), 76
 - [renderer_draw_filled_rounded_rect](#), 76
 - [renderer_draw_filled_triangle](#), 76
 - [renderer_draw_line](#), 77
 - [renderer_draw_pie](#), 77
 - [renderer_draw_pixel](#), 77
 - [renderer_draw_polygon](#), 77
 - [renderer_draw_rect](#), 77
 - [renderer_draw_rounded_rect](#), 78
 - [renderer_draw_text](#), 78
 - [renderer_draw_texture](#), 78
 - [renderer_draw_triangle](#), 78
 - [renderer_query_text_size](#), 78
 - [renderer_reset_clip_rect](#), 78
 - [renderer_resize_framebuffer](#), 79
 - [renderer_set_clip_rect](#), 79
 - [renderer_set_default_font](#), 79
 - [target_renderer](#), 79
- [renderer.h](#)
 - [_renderer_set_target](#), 80
 - [renderer_bind_framebuffer](#), 80
 - [renderer_clear](#), 80
 - [renderer_create_framebuffer](#), 80
 - [renderer_draw_arc](#), 81
 - [renderer_draw_bezier](#), 81
 - [renderer_draw_circle](#), 81
 - [renderer_draw_ellipse](#), 81
 - [renderer_draw_filled_circle](#), 81
 - [renderer_draw_filled_ellipse](#), 81
 - [renderer_draw_filled_pie](#), 82
 - [renderer_draw_filled_polygon](#), 82
 - [renderer_draw_filled_rect](#), 82
 - [renderer_draw_filled_rounded_rect](#), 82
 - [renderer_draw_filled_triangle](#), 82
 - [renderer_draw_line](#), 83
 - [renderer_draw_pie](#), 83
 - [renderer_draw_pixel](#), 83
 - [renderer_draw_polygon](#), 83
 - [renderer_draw_rect](#), 83
 - [renderer_draw_rounded_rect](#), 84
 - [renderer_draw_text](#), 84
 - [renderer_draw_texture](#), 84
 - [renderer_draw_triangle](#), 84
 - [renderer_query_text_size](#), 84
 - [renderer_reset_clip_rect](#), 84
 - [renderer_resize_framebuffer](#), 85
 - [renderer_set_clip_rect](#), 85
 - [renderer_set_default_font](#), 85
- [renderer_bind_framebuffer](#)
 - [renderer.c](#), 74
 - [renderer.h](#), 80
- [renderer_clear](#)
 - [renderer.c](#), 75
 - [renderer.h](#), 80
- [renderer_create_framebuffer](#)
 - [renderer.c](#), 75
 - [renderer.h](#), 80
- [renderer_draw_arc](#)
 - [renderer.c](#), 75
 - [renderer.h](#), 81
- [renderer_draw_bezier](#)
 - [renderer.c](#), 75
 - [renderer.h](#), 81
- [renderer_draw_circle](#)
 - [renderer.c](#), 75
 - [renderer.h](#), 81
- [renderer_draw_ellipse](#)
 - [renderer.c](#), 75
 - [renderer.h](#), 81
- [renderer_draw_filled_circle](#)
 - [renderer.c](#), 75
 - [renderer.h](#), 81
- [renderer_draw_filled_ellipse](#)

- renderer.c, [76](#)
 - renderer.h, [81](#)
- renderer_draw_filled_pie
 - renderer.c, [76](#)
 - renderer.h, [82](#)
- renderer_draw_filled_polygon
 - renderer.c, [76](#)
 - renderer.h, [82](#)
- renderer_draw_filled_rect
 - renderer.c, [76](#)
 - renderer.h, [82](#)
- renderer_draw_filled_rounded_rect
 - renderer.c, [76](#)
 - renderer.h, [82](#)
- renderer_draw_filled_triangle
 - renderer.c, [76](#)
 - renderer.h, [82](#)
- renderer_draw_line
 - renderer.c, [77](#)
 - renderer.h, [83](#)
- renderer_draw_pie
 - renderer.c, [77](#)
 - renderer.h, [83](#)
- renderer_draw_pixel
 - renderer.c, [77](#)
 - renderer.h, [83](#)
- renderer_draw_polygon
 - renderer.c, [77](#)
 - renderer.h, [83](#)
- renderer_draw_rect
 - renderer.c, [77](#)
 - renderer.h, [83](#)
- renderer_draw_rounded_rect
 - renderer.c, [78](#)
 - renderer.h, [84](#)
- renderer_draw_text
 - renderer.c, [78](#)
 - renderer.h, [84](#)
- renderer_draw_texture
 - renderer.c, [78](#)
 - renderer.h, [84](#)
- renderer_draw_triangle
 - renderer.c, [78](#)
 - renderer.h, [84](#)
- renderer_query_text_size
 - renderer.c, [78](#)
 - renderer.h, [84](#)
- renderer_reset_clip_rect
 - renderer.c, [78](#)
 - renderer.h, [84](#)
- renderer_resize_framebuffer
 - renderer.c, [79](#)
 - renderer.h, [85](#)
- renderer_set_clip_rect
 - renderer.c, [79](#)
 - renderer.h, [85](#)
- renderer_set_default_font
 - renderer.c, [79](#)
- renderer.h, [85](#)
- screen_to_coordinates
 - coordinate_system.c, [46](#)
 - coordinate_system.h, [50](#)
- selected_item
 - UIDropdownList, [20](#)
- shape.c
 - circle_create, [52](#)
 - line_create, [52](#)
 - point_create, [52](#)
- shape.h
 - Circle, [54](#)
 - circle_create, [54](#)
 - CoordinateSystem, [54](#)
 - IShape, [54](#)
 - Line, [54](#)
 - line_create, [54](#)
 - OVERLAP_DISTANCE, [53](#)
 - Point, [54](#)
 - point_create, [55](#)
- shapes
 - CoordinateSystem, [9](#)
- size
 - CoordinateSystem, [9](#)
 - Font, [9](#)
 - UIElement, [21](#)
 - Vector, [28](#)
- slider_color
 - UISlider, [24](#)
- splitbutton_index
 - _UISplitButtonItem, [6](#)
- src/app/app.c, [31](#)
- src/app/app.h, [33](#), [35](#)
- src/color/color.c, [35](#)
- src/color/color.h, [37](#), [40](#)
- src/font/font.c, [40](#)
- src/font/font.h, [41](#), [42](#)
- src/geometry/coordinate_system/coordinate_system.c, [42](#)
- src/geometry/coordinate_system/coordinate_system.h, [46](#), [51](#)
- src/geometry/shape/shape.c, [51](#)
- src/geometry/shape/shape.h, [53](#), [55](#)
- src/geometry/vector2/vector2.c, [56](#)
- src/geometry/vector2/vector2.h, [59](#), [63](#)
- src/includes.h, [64](#)
- src/input/input.c, [64](#)
- src/input/input.h, [67](#), [69](#)
- src/main.c, [70](#)
- src/renderer/renderer.c, [74](#)
- src/renderer/renderer.h, [79](#), [85](#)
- src/texture/texture.c, [86](#)
- src/texture/texture.h, [87](#), [88](#)
- src/ui/ui.c, [88](#)
- src/ui/ui.h, [90](#), [91](#)
- src/ui/ui_constraint/ui_constraint.c, [91](#)
- src/ui/ui_constraint/ui_constraint.h, [92](#), [94](#)
- src/ui/ui_element/ui_element.c, [95](#)

- src/ui/ui_element/ui_element.h, 98, 103
- src/utils/math/math.c, 106
- src/utils/math/math.h, 107, 109
- src/utils/vector/vector.c, 109
- src/utils/vector/vector.h, 111, 113
- src/window/window.c, 113
- src/window/window.h, 115, 117
- State
 - main.c, 71
- state
 - main.c, 73
- STATE_CIRCLE
 - main.c, 71
- STATE_CIRCLE_CENTER_PLACED
 - main.c, 71
- STATE_CS_DRAGGED
 - main.c, 71
- STATE_LINE
 - main.c, 71
- STATE_LINE_POINT1_PLACED
 - main.c, 71
- STATE_POINT
 - main.c, 71
- STATE_POINTER
 - main.c, 71
- target_frame_time
 - AppData, 7
- target_input_data
 - input.c, 66
- target_renderer
 - renderer.c, 79
- target_ui_data
 - ui.c, 89
- text
 - _UIDropdownItem, 5
 - _UISplitButtonItem, 6
 - UIButton, 14
 - UILabel, 22
 - UITextbox, 27
- text_color
 - UIButton, 14
 - UIDropdownList, 20
 - UISplitButton, 26
 - UITextbox, 27
- text_input
 - UIData, 18
- text_position
 - UIButton, 14
- Texture, 13
 - height, 13
 - texture, 13
 - texture.h, 87
 - width, 13
- texture
 - Texture, 13
 - UIImageButton, 22
- texture.c
 - _texture_add, 86
 - _texture_close, 86
 - _texture_init, 86
 - texture_load, 86
 - textures, 87
- texture.h
 - _texture_add, 87
 - _texture_close, 87
 - _texture_init, 87
 - Texture, 87
 - texture_load, 87
- texture_load
 - texture.c, 86
 - texture.h, 87
- textures
 - texture.c, 87
- thickness
 - UISlider, 24
- translate
 - IShape, 11
- TRANSPARENT
 - color.h, 38
- TWO_PI
 - math.h, 107
- ui.c
 - _ui_close, 89
 - _ui_get_target, 89
 - _ui_handle_event, 89
 - _ui_init, 89
 - _ui_render, 89
 - _ui_set_target, 89
 - _ui_update, 89
 - target_ui_data, 89
- ui.h
 - _ui_close, 90
 - _ui_get_target, 90
 - _ui_handle_event, 90
 - _ui_init, 90
 - _ui_render, 91
 - _ui_set_target, 91
 - _ui_update, 91
 - UIData, 90
- ui_constraint.c
 - constraints_from_string, 92
 - new_aspect_constraint, 92
 - new_center_constraint, 92
 - new_offset_constraint, 92
 - new_pixel_constraint, 92
 - new_relative_constraint, 92
- ui_constraint.h
 - constraints_from_string, 94
 - ConstraintType, 93
 - CT_ASPECT, 93
 - CT_CENTER, 93
 - CT_OFFSET, 93
 - CT_PIXEL, 93
 - CT_RELATIVE, 93
 - new_aspect_constraint, 94
 - new_center_constraint, 94

- new_offset_constraint, 94
- new_pixel_constraint, 94
- new_relative_constraint, 94
- UIConstraint, 93
- UIConstraints, 93
- ui_create_button
 - ui_element.c, 96
 - ui_element.h, 101
- ui_create_checkbox
 - ui_element.c, 96
 - ui_element.h, 102
- ui_create_container
 - ui_element.c, 96
 - ui_element.h, 102
- ui_create_dropdown
 - ui_element.c, 97
 - ui_element.h, 102
- ui_create_imagebutton
 - ui_element.c, 97
 - ui_element.h, 102
- ui_create_label
 - ui_element.c, 97
 - ui_element.h, 102
- ui_create_panel
 - ui_element.c, 97
 - ui_element.h, 102
- ui_create_slider
 - ui_element.c, 97
 - ui_element.h, 103
- ui_create_splitbutton
 - ui_element.c, 97
 - ui_element.h, 103
- ui_create_textbox
 - ui_element.c, 98
 - ui_element.h, 103
- ui_data
 - Window, 29
- ui_element.c
 - _UIDropdownItem, 95
 - _UISplitButtonItem, 95
 - _ui_container_destroy, 96
 - _ui_container_recalculate, 96
 - _ui_container_render, 96
 - _ui_container_update, 96
 - ui_create_button, 96
 - ui_create_checkbox, 96
 - ui_create_container, 96
 - ui_create_dropdown, 97
 - ui_create_imagebutton, 97
 - ui_create_label, 97
 - ui_create_panel, 97
 - ui_create_slider, 97
 - ui_create_splitbutton, 97
 - ui_create_textbox, 98
- ui_element.h
 - _ui_container_destroy, 101
 - _ui_container_recalculate, 101
 - _ui_container_render, 101
 - _ui_container_update, 101
 - MouseState, 100, 101
 - MS_HOVER, 101
 - MS_NONE, 101
 - MS_PRESS, 101
 - ui_create_button, 101
 - ui_create_checkbox, 102
 - ui_create_container, 102
 - ui_create_dropdown, 102
 - ui_create_imagebutton, 102
 - ui_create_label, 102
 - ui_create_panel, 102
 - ui_create_slider, 103
 - ui_create_splitbutton, 103
 - ui_create_textbox, 103
 - UIButton, 100
 - UICheckbox, 100
 - UIContainer, 100
 - UIDropdownList, 100
 - UIElement, 100
 - UIImageButton, 100
 - UILabel, 100
 - UIPanel, 100
 - UISlider, 100
 - UISplitButton, 100
 - UITEXT_MAX_LENGTH, 99
 - UITextbox, 101
- UIButton, 13
 - base, 14
 - color, 14
 - corner_radius, 14
 - mouse_state, 14
 - on_click, 14
 - text, 14
 - text_color, 14
 - text_position, 14
 - ui_element.h, 100
- UICheckbox, 15
 - base, 15
 - checked, 15
 - checked_color, 15
 - corner_radius, 15
 - mouse_state, 15
 - on_checked_changed, 15
 - ui_element.h, 100
 - unchecked_color, 16
- UIConstraint, 16
 - constraint_type, 16
 - recalculate, 16
 - ui_constraint.h, 93
 - value, 16
- UIConstraints, 16
 - height, 17
 - ui_constraint.h, 93
 - width, 17
 - x, 17
 - y, 17
- UIContainer, 17

- base, 17
- children, 17
- on_size_changed, 18
- ui_element.h, 100
- UIData, 18
 - backspace_pressed, 18
 - expanded_splitbutton, 18
 - main_container, 18
 - mouse_captured, 18
 - text_input, 18
 - ui.h, 90
- UIDropdownList, 19
 - base, 19
 - color, 19
 - corner_radius, 19
 - expanded, 19
 - items, 19
 - on_selection_changed, 19
 - selected_item, 20
 - text_color, 20
 - ui_element.h, 100
- UIElement, 20
 - constraints, 20
 - destroy, 20
 - parent, 20
 - position, 20
 - recalculate, 21
 - render, 21
 - size, 21
 - ui_element.h, 100
 - update, 21
- UIImageButton, 21
 - base, 21
 - mouse_state, 21
 - on_click, 22
 - texture, 22
 - ui_element.h, 100
- UILabel, 22
 - base, 22
 - color, 22
 - text, 22
 - ui_element.h, 100
- UIPanel, 23
 - base, 23
 - border_color, 23
 - border_width, 23
 - color, 23
 - corner_radius, 23
 - ui_element.h, 100
- UISlider, 23
 - base, 24
 - color, 24
 - corner_radius, 24
 - mouse_state, 24
 - on_value_changed, 24
 - slider_color, 24
 - thickness, 24
 - ui_element.h, 100
 - value, 24
- UISplitButton, 25
 - auto_dropdown, 25
 - base, 25
 - color, 25
 - corner_radius, 25
 - expanded, 25
 - items, 25
 - on_item_clicked, 26
 - text_color, 26
 - ui_element.h, 100
- UITEXT_MAX_LENGTH
 - ui_element.h, 99
- UITextbox, 26
 - base, 26
 - color, 26
 - corner_radius, 26
 - focused, 26
 - mouse_state, 27
 - on_text_changed, 27
 - text, 27
 - text_color, 27
 - ui_element.h, 101
- unchecked_color
 - UICheckbox, 16
- update
 - UIElement, 21
- value
 - UIConstraint, 16
 - UISlider, 24
- Vector, 27
 - capacity, 27
 - data, 27
 - size, 28
 - vector.h, 111
- vector.c
 - vector_clear, 109
 - vector_contains, 109
 - vector_create, 109
 - vector_free, 110
 - vector_get, 110
 - vector_insert, 110
 - vector_pop_back, 110
 - vector_push_back, 110
 - vector_remove, 110
 - vector_remove_at, 110
 - vector_reserve, 110
 - vector_size, 111
- vector.h
 - Vector, 111
 - vector_clear, 112
 - vector_contains, 112
 - vector_create, 112
 - vector_free, 112
 - vector_get, 112
 - vector_insert, 112
 - vector_pop_back, 112
 - vector_push_back, 112

- vector_remove, 112
- vector_remove_at, 113
- vector_reserve, 113
- vector_size, 113
- Vector2, 28
 - vector2.h, 60
 - x, 28
 - y, 28
- vector2.c
 - vector2_add, 57
 - vector2_angle, 57
 - vector2_create, 57
 - vector2_distance, 57
 - vector2_divide, 57
 - vector2_dot, 57
 - vector2_down, 57
 - vector2_from_point, 57
 - vector2_from_polar, 57
 - vector2_left, 58
 - vector2_length, 58
 - vector2_multiply, 58
 - vector2_negate, 58
 - vector2_normalize, 58
 - vector2_one, 58
 - vector2_reflect, 58
 - vector2_right, 58
 - vector2_rotate, 58
 - vector2_rotate90, 59
 - vector2_scale, 59
 - vector2_subtract, 59
 - vector2_up, 59
 - vector2_zero, 59
- vector2.h
 - Vector2, 60
 - vector2_add, 60
 - vector2_angle, 60
 - vector2_create, 60
 - vector2_distance, 60
 - vector2_divide, 61
 - vector2_dot, 61
 - vector2_down, 61
 - vector2_from_point, 61
 - vector2_from_polar, 61
 - vector2_left, 61
 - vector2_length, 61
 - vector2_multiply, 61
 - vector2_negate, 62
 - vector2_normalize, 62
 - vector2_one, 62
 - vector2_reflect, 62
 - vector2_right, 62
 - vector2_rotate, 62
 - vector2_rotate90, 62
 - vector2_scale, 62
 - vector2_subtract, 63
 - vector2_up, 63
 - vector2_zero, 63
- vector2_add
 - vector2.c, 57
 - vector2.h, 60
- vector2_angle
 - vector2.c, 57
 - vector2.h, 60
- vector2_create
 - vector2.c, 57
 - vector2.h, 60
- vector2_distance
 - vector2.c, 57
 - vector2.h, 60
- vector2_divide
 - vector2.c, 57
 - vector2.h, 61
- vector2_dot
 - vector2.c, 57
 - vector2.h, 61
- vector2_down
 - vector2.c, 57
 - vector2.h, 61
- vector2_from_point
 - vector2.c, 57
 - vector2.h, 61
- vector2_from_polar
 - vector2.c, 57
 - vector2.h, 61
- vector2_left
 - vector2.c, 58
 - vector2.h, 61
- vector2_length
 - vector2.c, 58
 - vector2.h, 61
- vector2_multiply
 - vector2.c, 58
 - vector2.h, 61
- vector2_negate
 - vector2.c, 58
 - vector2.h, 62
- vector2_normalize
 - vector2.c, 58
 - vector2.h, 62
- vector2_one
 - vector2.c, 58
 - vector2.h, 62
- vector2_reflect
 - vector2.c, 58
 - vector2.h, 62
- vector2_right
 - vector2.c, 58
 - vector2.h, 62
- vector2_rotate
 - vector2.c, 58
 - vector2.h, 62
- vector2_rotate90
 - vector2.c, 59
 - vector2.h, 62
- vector2_scale
 - vector2.c, 59

- vector2.h, 62
- vector2_subtract
 - vector2.c, 59
 - vector2.h, 63
- vector2_up
 - vector2.c, 59
 - vector2.h, 63
- vector2_zero
 - vector2.c, 59
 - vector2.h, 63
- vector_clear
 - vector.c, 109
 - vector.h, 112
- vector_contains
 - vector.c, 109
 - vector.h, 112
- vector_create
 - vector.c, 109
 - vector.h, 112
- vector_free
 - vector.c, 110
 - vector.h, 112
- vector_get
 - vector.c, 110
 - vector.h, 112
- vector_insert
 - vector.c, 110
 - vector.h, 112
- vector_pop_back
 - vector.c, 110
 - vector.h, 112
- vector_push_back
 - vector.c, 110
 - vector.h, 112
- vector_remove
 - vector.c, 110
 - vector.h, 112
- vector_remove_at
 - vector.c, 110
 - vector.h, 113
- vector_reserve
 - vector.c, 110
 - vector.h, 113
- vector_size
 - vector.c, 111
 - vector.h, 113
- WHITE
 - color.h, 38
- width
 - Texture, 13
 - UIConstraints, 17
- Window, 28
 - close_requested, 29
 - input_data, 29
 - renderer, 29
 - ui_data, 29
 - window, 29
 - window.h, 116
- window
 - Window, 29
- window.c
 - _window_close, 114
 - _window_handle_event, 114
 - _window_render, 114
 - _window_reset, 114
 - _window_update, 114
 - window_create, 114
 - window_focus, 114
 - window_get_main_container, 115
 - window_hide, 115
 - window_show, 115
- window.h
 - _window_close, 116
 - _window_handle_event, 116
 - _window_render, 116
 - _window_reset, 116
 - _window_update, 116
 - Window, 116
 - window_create, 116
 - window_focus, 116
 - window_get_main_container, 116
 - window_hide, 117
 - window_show, 117
- window_create
 - window.c, 114
 - window.h, 116
- window_focus
 - window.c, 114
 - window.h, 116
- window_get_main_container
 - window.c, 115
 - window.h, 116
- window_hide
 - window.c, 115
 - window.h, 117
- window_show
 - window.c, 115
 - window.h, 117
- windows
 - AppData, 7
- x
 - UIConstraints, 17
 - Vector2, 28
- y
 - UIConstraints, 17
 - Vector2, 28
- YELLOW
 - color.h, 38
- zoom
 - CoordinateSystem, 9