

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 AngleBisector Struct Reference	5
3.1.1 Member Data Documentation	5
3.1.1.1 base	5
3.1.1.2 line1	5
3.1.1.3 line2	5
3.2 AppData Struct Reference	5
3.2.1 Member Data Documentation	6
3.2.1.1 delta_time	6
3.2.1.2 frame_start	6
3.2.1.3 last_frame_start	6
3.2.1.4 target_frame_time	6
3.2.1.5 windows	6
3.3 Circle Struct Reference	6
3.3.1 Member Data Documentation	6
3.3.1.1 base	6
3.3.1.2 center	6
3.3.1.3 perimeter_point	7
3.4 CoordinateSystem Struct Reference	7
3.4.1 Member Data Documentation	7
3.4.1.1 intersection_points	7
3.4.1.2 origin	7
3.4.1.3 position	7
3.4.1.4 shapes	7
3.4.1.5 size	7
3.4.1.6 zoom	7
3.5 Font Struct Reference	8
3.5.1 Member Data Documentation	8
3.5.1.1 font	8
3.5.1.2 size	8
3.6 InputData Struct Reference	8
3.6.1 Member Data Documentation	8
3.6.1.1 current_keyboard_state	8
3.6.1.2 current_mouse_button_state	8
3.6.1.3 current_mouse_position	8
3.6.1.4 key_count	8
3.6.1.5 mouse_wheel_delta	9

3.6.1.6 old_keyboard_state	9
3.6.1.7 old_mouse_button_state	9
3.6.1.8 old_mouse_position	9
3.7 Line Struct Reference	9
3.7.1 Member Data Documentation	9
3.7.1.1 base	9
3.7.1.2 p1	9
3.7.1.3 p2	9
3.8 Parallel Struct Reference	10
3.8.1 Member Data Documentation	10
3.8.1.1 base	10
3.8.1.2 line	10
3.8.1.3 point	10
3.9 Perpendicular Struct Reference	10
3.9.1 Member Data Documentation	10
3.9.1.1 base	10
3.9.1.2 line	10
3.9.1.3 point	10
3.10 Point Struct Reference	11
3.10.1 Member Data Documentation	11
3.10.1.1 base	11
3.10.1.2 coordinates	11
3.11 Shape Struct Reference	11
3.11.1 Member Data Documentation	11
3.11.1.1 dragged	11
3.11.1.2 selected	11
3.11.1.3 type	11
3.12 Tangent Struct Reference	12
3.12.1 Member Data Documentation	12
3.12.1.1 base	12
3.12.1.2 circle	12
3.12.1.3 point	12
3.13 Texture Struct Reference	12
3.13.1 Member Data Documentation	12
3.13.1.1 height	12
3.13.1.2 texture	12
3.13.1.3 width	12
3.14 UIButton Struct Reference	13
3.14.1 Member Data Documentation	13
3.14.1.1 base	13
3.14.1.2 color	13
3.14.1.3 corner_radius	13

3.14.1.4 mouse_state	13
3.14.1.5 on_click	13
3.14.1.6 text	13
3.14.1.7 text_color	13
3.14.1.8 text_position	13
3.15 UICheckbox Struct Reference	14
3.15.1 Member Data Documentation	14
3.15.1.1 base	14
3.15.1.2 checked	14
3.15.1.3 checked_color	14
3.15.1.4 corner_radius	14
3.15.1.5 mouse_state	14
3.15.1.6 on_checked_changed	14
3.15.1.7 unchecked_color	14
3.16 UIConstraint Struct Reference	14
3.16.1 Member Data Documentation	15
3.16.1.1 constraint_type	15
3.16.1.2 value	15
3.17 UIConstraints Struct Reference	15
3.17.1 Member Data Documentation	15
3.17.1.1 height	15
3.17.1.2 width	15
3.17.1.3 x	15
3.17.1.4 y	15
3.18 UIContainer Struct Reference	15
3.18.1 Member Data Documentation	16
3.18.1.1 base	16
3.18.1.2 children	16
3.18.1.3 on_size_changed	16
3.19 UIData Struct Reference	16
3.19.1 Member Data Documentation	16
3.19.1.1 backspace_pressed	16
3.19.1.2 expanded_splitbutton	16
3.19.1.3 main_container	16
3.19.1.4 mouse_captured	16
3.19.1.5 text_input	17
3.20 UIDropdownList Struct Reference	17
3.20.1 Member Data Documentation	17
3.20.1.1 base	17
3.20.1.2 color	17
3.20.1.3 corner_radius	17
3.20.1.4 expanded	17

3.20.1.5 items	17
3.20.1.6 on_selection_changed	17
3.20.1.7 selected_item	17
3.20.1.8 text_color	18
3.21 UIElement Struct Reference	18
3.21.1 Member Data Documentation	18
3.21.1.1 constraints	18
3.21.1.2 destroy	18
3.21.1.3 parent	18
3.21.1.4 position	18
3.21.1.5 recalculate	18
3.21.1.6 render	18
3.21.1.7 shown	18
3.21.1.8 size	19
3.21.1.9 update	19
3.22 UIButton Struct Reference	19
3.22.1 Member Data Documentation	19
3.22.1.1 base	19
3.22.1.2 mouse_state	19
3.22.1.3 on_click	19
3.22.1.4 texture	19
3.23 UILabel Struct Reference	19
3.23.1 Member Data Documentation	20
3.23.1.1 base	20
3.23.1.2 color	20
3.23.1.3 text	20
3.24 UIPanel Struct Reference	20
3.24.1 Member Data Documentation	20
3.24.1.1 base	20
3.24.1.2 border_color	20
3.24.1.3 border_width	20
3.24.1.4 color	20
3.24.1.5 corner_radius	21
3.25 UISlider Struct Reference	21
3.25.1 Member Data Documentation	21
3.25.1.1 base	21
3.25.1.2 color	21
3.25.1.3 corner_radius	21
3.25.1.4 mouse_state	21
3.25.1.5 on_value_changed	21
3.25.1.6 slider_color	21
3.25.1.7 thickness	21

3.25.1.8 value	22
3.26 UISplitButton Struct Reference	22
3.26.1 Member Data Documentation	22
3.26.1.1 auto_dropdown	22
3.26.1.2 base	22
3.26.1.3 color	22
3.26.1.4 corner_radius	22
3.26.1.5 expanded	22
3.26.1.6 items	22
3.26.1.7 on_item_clicked	22
3.26.1.8 text_color	23
3.27 UITextbox Struct Reference	23
3.27.1 Member Data Documentation	23
3.27.1.1 base	23
3.27.1.2 color	23
3.27.1.3 corner_radius	23
3.27.1.4 focused	23
3.27.1.5 mouse_state	23
3.27.1.6 on_text_changed	23
3.27.1.7 text	23
3.27.1.8 text_color	24
3.28 Vector Struct Reference	24
3.28.1 Member Data Documentation	24
3.28.1.1 capacity	24
3.28.1.2 data	24
3.28.1.3 size	24
3.29 Vector2 Struct Reference	24
3.29.1 Member Data Documentation	24
3.29.1.1 x	24
3.29.1.2 y	25
3.30 Window Struct Reference	25
3.30.1 Member Data Documentation	25
3.30.1.1 close_requested	25
3.30.1.2 input_data	25
3.30.1.3 renderer	25
3.30.1.4 ui_data	25
3.30.1.5 window	25
4 File Documentation	27
4.1 src/app/app.c File Reference	27
4.1.1 Function Documentation	27
4.1.1.1 _app_add_window()	27

4.1.1.2 app_close()	27
4.1.1.3 app_get_delta_time()	27
4.1.1.4 app_get_target()	27
4.1.1.5 app_get_time()	27
4.1.1.6 app_get_windows()	27
4.1.1.7 app_init()	28
4.1.1.8 app_render()	28
4.1.1.9 app_request_close()	28
4.1.1.10 app_set_target()	28
4.1.1.11 app_set_target_fps()	28
4.1.1.12 app_update()	28
4.2 src/app/app.h File Reference	28
4.2.1 Typedef Documentation	28
4.2.1.1 AppData	28
4.2.2 Function Documentation	28
4.2.2.1 _app_add_window()	28
4.2.2.2 app_close()	29
4.2.2.3 app_get_delta_time()	29
4.2.2.4 app_get_target()	29
4.2.2.5 app_get_time()	29
4.2.2.6 app_get_windows()	29
4.2.2.7 app_init()	29
4.2.2.8 app_render()	29
4.2.2.9 app_request_close()	29
4.2.2.10 app_set_target()	29
4.2.2.11 app_set_target_fps()	29
4.2.2.12 app_update()	30
4.3 app.h	30
4.4 src/color/color.c File Reference	30
4.4.1 Function Documentation	30
4.4.1.1 color_clever_shift()	30
4.4.1.2 color_fade()	30
4.4.1.3 color_from_grayscale()	31
4.4.1.4 color_from_hex()	31
4.4.1.5 color_from_hsv()	31
4.4.1.6 color_from_rgb()	31
4.4.1.7 color_from_rgba()	31
4.4.1.8 color_shift()	31
4.5 src/color/color.h File Reference	31
4.5.1 Macro Definition Documentation	31
4.5.1.1 BLACK	31
4.5.1.2 BLUE	32

4.5.1.3 CYAN	32
4.5.1.4 DARK_GRAY	32
4.5.1.5 GRAY	32
4.5.1.6 GREEN	32
4.5.1.7 MAGENTA	32
4.5.1.8 RED	32
4.5.1.9 TRANSPARENT	32
4.5.1.10 WHITE	32
4.5.1.11 YELLOW	32
4.5.2 Typedef Documentation	33
4.5.2.1 Color	33
4.5.3 Function Documentation	33
4.5.3.1 color_clever_shift()	33
4.5.3.2 color_fade()	33
4.5.3.3 color_from_grayscale()	33
4.5.3.4 color_from_hex()	33
4.5.3.5 color_from_hsv()	33
4.5.3.6 color_from_rgb()	33
4.5.3.7 color_from_rgba()	34
4.5.3.8 color_shift()	34
4.6 color.h	34
4.7 src/font/font.c File Reference	34
4.7.1 Function Documentation	34
4.7.1.1 _font_close()	34
4.7.1.2 _font_init()	34
4.7.1.3 font_load()	35
4.8 src/font/font.h File Reference	35
4.8.1 Typedef Documentation	35
4.8.1.1 Font	35
4.8.2 Function Documentation	35
4.8.2.1 _font_close()	35
4.8.2.2 _font_init()	35
4.8.2.3 font_load()	35
4.9 font.h	35
4.10 src/geometry/coordinate_system/coordinate_system.c File Reference	36
4.10.1 Function Documentation	36
4.10.1.1 coordinate_system_clear()	36
4.10.1.2 coordinate_system_create()	36
4.10.1.3 coordinate_system_delete_selected_shapes()	37
4.10.1.4 coordinate_system_deselect_shape()	37
4.10.1.5 coordinate_system_deselect_shapes()	37
4.10.1.6 coordinate_system_destroy()	38

4.10.1.7	<code>coordinate_system_destroy_shape()</code>	38
4.10.1.8	<code>coordinate_system_drag_selected_shapes()</code>	38
4.10.1.9	<code>coordinate_system_draw()</code>	38
4.10.1.10	<code>coordinate_system_get_hovered_shape()</code>	39
4.10.1.11	<code>coordinate_system_get_selected_shapes()</code>	39
4.10.1.12	<code>coordinate_system_is_hovered()</code>	39
4.10.1.13	<code>coordinate_system_load()</code>	40
4.10.1.14	<code>coordinate_system_save()</code>	40
4.10.1.15	<code>coordinate_system_select_all_shapes()</code>	40
4.10.1.16	<code>coordinate_system_select_shape()</code>	40
4.10.1.17	<code>coordinate_system_translate()</code>	41
4.10.1.18	<code>coordinate_system_update()</code>	41
4.10.1.19	<code>coordinate_system_update_dimensions()</code>	41
4.10.1.20	<code>coordinate_system_zoom()</code>	42
4.10.1.21	<code>coordinates_to_screen()</code>	42
4.10.1.22	<code>screen_to_coordinates()</code>	42
4.11	<code>src/geometry/coordinate_system/coordinate_system.h</code> File Reference	43
4.11.1	Macro Definition Documentation	43
4.11.1.1	<code>INITIAL_ZOOM</code>	43
4.11.2	Typedef Documentation	43
4.11.2.1	<code>CoordinateSystem</code>	43
4.11.3	Function Documentation	43
4.11.3.1	<code>coordinate_system_clear()</code>	43
4.11.3.2	<code>coordinate_system_create()</code>	44
4.11.3.3	<code>coordinate_system_delete_selected_shapes()</code>	44
4.11.3.4	<code>coordinate_system_deselect_shape()</code>	44
4.11.3.5	<code>coordinate_system_deselect_shapes()</code>	45
4.11.3.6	<code>coordinate_system_destroy()</code>	45
4.11.3.7	<code>coordinate_system_destroy_shape()</code>	45
4.11.3.8	<code>coordinate_system_drag_selected_shapes()</code>	45
4.11.3.9	<code>coordinate_system_draw()</code>	46
4.11.3.10	<code>coordinate_system_get_hovered_shape()</code>	46
4.11.3.11	<code>coordinate_system_get_selected_shapes()</code>	46
4.11.3.12	<code>coordinate_system_is_hovered()</code>	46
4.11.3.13	<code>coordinate_system_load()</code>	47
4.11.3.14	<code>coordinate_system_save()</code>	47
4.11.3.15	<code>coordinate_system_select_all_shapes()</code>	47
4.11.3.16	<code>coordinate_system_select_shape()</code>	48
4.11.3.17	<code>coordinate_system_translate()</code>	48
4.11.3.18	<code>coordinate_system_update()</code>	48
4.11.3.19	<code>coordinate_system_update_dimensions()</code>	48
4.11.3.20	<code>coordinate_system_zoom()</code>	49

4.11.3.21 <code>coordinates_to_screen()</code>	49
4.11.3.22 <code>screen_to_coordinates()</code>	49
4.12 <code>coordinate_system.h</code>	50
4.13 <code>src/geometry/intersection/intersection.c</code> File Reference	50
4.13.1 Macro Definition Documentation	50
4.13.1.1 EPSILON	50
4.13.2 Function Documentation	50
4.13.2.1 <code>intersection_get()</code>	50
4.14 <code>src/geometry/intersection/intersection.h</code> File Reference	51
4.14.1 Function Documentation	51
4.14.1.1 <code>intersection_get()</code>	51
4.15 <code>intersection.h</code>	51
4.16 <code>src/geometry/shape/shape.c</code> File Reference	51
4.16.1 Macro Definition Documentation	51
4.16.1.1 EPSILON	51
4.16.2 Function Documentation	51
4.16.2.1 <code>angle_bisector_create()</code>	51
4.16.2.2 <code>circle_create()</code>	52
4.16.2.3 <code>line_create()</code>	52
4.16.2.4 <code>parallel_create()</code>	53
4.16.2.5 <code>perpendicular_create()</code>	53
4.16.2.6 <code>point_create()</code>	53
4.16.2.7 <code>shape_destroy()</code>	54
4.16.2.8 <code>shape_draw()</code>	54
4.16.2.9 <code>shape_is_defined_by()</code>	54
4.16.2.10 <code>shape_overlap_point()</code>	54
4.16.2.11 <code>shape_translate()</code>	54
4.16.2.12 <code>shape_update()</code>	54
4.16.2.13 <code>tangent_create()</code>	54
4.16.3 Variable Documentation	55
4.16.3.1 <code>shape_destroy_funcs</code>	55
4.16.3.2 <code>shape_draw_funcs</code>	55
4.16.3.3 <code>shape_is_defined_by_funcs</code>	55
4.16.3.4 <code>shape_overlap_point_funcs</code>	55
4.16.3.5 <code>shape_translate_funcs</code>	55
4.17 <code>src/geometry/shape/shape.h</code> File Reference	56
4.17.1 Macro Definition Documentation	56
4.17.1.1 OVERLAP_DISTANCE	56
4.17.2 Typedef Documentation	56
4.17.2.1 AngleBisector	56
4.17.2.2 Circle	56
4.17.2.3 CoordinateSystem	56

4.17.2.4 Line	56
4.17.2.5 Parallel	57
4.17.2.6 Perpendicular	57
4.17.2.7 Point	57
4.17.2.8 Shape	57
4.17.2.9 ShapeDestroy	57
4.17.2.10 ShapeDraw	57
4.17.2.11 ShapesDefinedBy	57
4.17.2.12 ShapeOverlapPoint	57
4.17.2.13 ShapeTranslate	57
4.17.2.14 ShapeType	57
4.17.2.15 Tangent	57
4.17.3 Enumeration Type Documentation	57
4.17.3.1 ShapeType	57
4.17.4 Function Documentation	58
4.17.4.1 angle_bisector_create()	58
4.17.4.2 circle_create()	58
4.17.4.3 line_create()	59
4.17.4.4 parallel_create()	59
4.17.4.5 perpendicular_create()	59
4.17.4.6 point_create()	60
4.17.4.7 shape_destroy()	60
4.17.4.8 shape_draw()	60
4.17.4.9 shape_is_defined_by()	60
4.17.4.10 shape_overlap_point()	61
4.17.4.11 shape_translate()	61
4.17.4.12 shape_update()	61
4.17.4.13 tangent_create()	61
4.18 shape.h	61
4.19 src/geometry/vector2/vector2.c File Reference	63
4.19.1 Function Documentation	63
4.19.1.1 vector2_add()	63
4.19.1.2 vector2_angle()	63
4.19.1.3 vector2_create()	63
4.19.1.4 vector2_distance()	63
4.19.1.5 vector2_divide()	63
4.19.1.6 vector2_dot()	63
4.19.1.7 vector2_down()	64
4.19.1.8 vector2_from_point()	64
4.19.1.9 vector2_from_polar()	64
4.19.1.10 vector2_left()	64
4.19.1.11 vector2_length()	64

4.19.1.12 vector2_multiply()	64
4.19.1.13 vector2_negate()	64
4.19.1.14 vector2_normalize()	64
4.19.1.15 vector2_one()	64
4.19.1.16 vector2_reflect()	65
4.19.1.17 vector2_right()	65
4.19.1.18 vector2_rotate()	65
4.19.1.19 vector2_rotate90()	65
4.19.1.20 vector2_scale()	65
4.19.1.21 vector2_subtract()	65
4.19.1.22 vector2_up()	65
4.19.1.23 vector2_zero()	65
4.20 src/geometry/vector2/vector2.h File Reference	66
4.20.1 Typedef Documentation	66
4.20.1.1 Vector2	66
4.20.2 Function Documentation	66
4.20.2.1 vector2_add()	66
4.20.2.2 vector2_angle()	66
4.20.2.3 vector2_create()	66
4.20.2.4 vector2_distance()	66
4.20.2.5 vector2_divide()	66
4.20.2.6 vector2_dot()	67
4.20.2.7 vector2_down()	67
4.20.2.8 vector2_from_point()	67
4.20.2.9 vector2_from_polar()	67
4.20.2.10 vector2_left()	67
4.20.2.11 vector2_length()	67
4.20.2.12 vector2_multiply()	67
4.20.2.13 vector2_negate()	67
4.20.2.14 vector2_normalize()	67
4.20.2.15 vector2_one()	68
4.20.2.16 vector2_reflect()	68
4.20.2.17 vector2_right()	68
4.20.2.18 vector2_rotate()	68
4.20.2.19 vector2_rotate90()	68
4.20.2.20 vector2_scale()	68
4.20.2.21 vector2_subtract()	68
4.20.2.22 vector2_up()	68
4.20.2.23 vector2_zero()	68
4.21 vector2.h	69
4.22 src/input/input.c File Reference	69
4.22.1 Function Documentation	69

4.22.1.1	_input_close()	69
4.22.1.2	_input_handle_event()	69
4.22.1.3	_input_init()	69
4.22.1.4	_input_reset()	70
4.22.1.5	_input_set_target()	70
4.22.1.6	input_get_mouse_motion()	70
4.22.1.7	input_get_mouse_position()	70
4.22.1.8	input_get_mouse_wheel_delta()	70
4.22.1.9	input_is_key_down()	70
4.22.1.10	input_is_key_pressed()	70
4.22.1.11	input_is_key_released()	70
4.22.1.12	input_is_mouse_button_down()	70
4.22.1.13	input_is_mouse_button_pressed()	71
4.22.1.14	input_is_mouse_button_released()	71
4.23	src/input/input.h File Reference	71
4.23.1	Typedef Documentation	71
4.23.1.1	InputData	71
4.23.2	Function Documentation	71
4.23.2.1	_input_close()	71
4.23.2.2	_input_handle_event()	71
4.23.2.3	_input_init()	71
4.23.2.4	_input_reset()	71
4.23.2.5	_input_set_target()	72
4.23.2.6	input_get_mouse_motion()	72
4.23.2.7	input_get_mouse_position()	72
4.23.2.8	input_get_mouse_wheel_delta()	72
4.23.2.9	input_is_key_down()	72
4.23.2.10	input_is_key_pressed()	72
4.23.2.11	input_is_key_released()	72
4.23.2.12	input_is_mouse_button_down()	72
4.23.2.13	input_is_mouse_button_pressed()	72
4.23.2.14	input_is_mouse_button_released()	73
4.24	input.h	73
4.25	src/main.c File Reference	73
4.25.1	Detailed Description	73
4.25.2	Macro Definition Documentation	74
4.25.2.1	FPS	74
4.25.2.2	MOUSE_WHEEL_SENSITIVITY	74
4.25.3	Typedef Documentation	74
4.25.3.1	State	74
4.25.4	Enumeration Type Documentation	74
4.25.4.1	State	74

4.25.5 Function Documentation	75
4.25.5.1 main()	75
4.25.5.2 on_angle_bisector_clicked() [1/2]	75
4.25.5.3 on_angle_bisector_clicked() [2/2]	75
4.25.5.4 on_cancel_button_clicked()	75
4.25.5.5 on_canvas_size_changed()	75
4.25.5.6 on_circle_clicked() [1/2]	75
4.25.5.7 on_circle_clicked() [2/2]	75
4.25.5.8 on_editmenu_clicked() [1/2]	75
4.25.5.9 on_editmenu_clicked() [2/2]	76
4.25.5.10 on_filemenu_clicked() [1/2]	76
4.25.5.11 on_filemenu_clicked() [2/2]	76
4.25.5.12 on_line_clicked() [1/2]	76
4.25.5.13 on_line_clicked() [2/2]	76
4.25.5.14 on_open_button_clicked()	76
4.25.5.15 on_parallel_clicked() [1/2]	76
4.25.5.16 on_parallel_clicked() [2/2]	76
4.25.5.17 on_perpendicular_clicked() [1/2]	77
4.25.5.18 on_perpendicular_clicked() [2/2]	77
4.25.5.19 on_point_clicked() [1/2]	77
4.25.5.20 on_point_clicked() [2/2]	77
4.25.5.21 on_pointer_clicked() [1/2]	77
4.25.5.22 on_pointer_clicked() [2/2]	77
4.25.5.23 on_save_button_clicked()	77
4.25.5.24 on_tangent_clicked() [1/2]	77
4.25.5.25 on_tangent_clicked() [2/2]	77
4.25.6 Variable Documentation	78
4.25.6.1 cs	78
4.25.6.2 state	78
4.26 src/renderer/renderer.c File Reference	78
4.26.1 Function Documentation	78
4.26.1.1 _renderer_set_target()	78
4.26.1.2 renderer_bind_framebuffer()	78
4.26.1.3 renderer_clear()	78
4.26.1.4 renderer_create_framebuffer()	78
4.26.1.5 renderer_draw_arc()	78
4.26.1.6 renderer_draw_bezier()	79
4.26.1.7 renderer_draw_circle()	79
4.26.1.8 renderer_draw_ellipse()	79
4.26.1.9 renderer_draw_filled_circle()	79
4.26.1.10 renderer_draw_filled_ellipse()	79
4.26.1.11 renderer_draw_filled_pie()	79

4.26.1.12	renderer_draw_filled_polygon()	80
4.26.1.13	renderer_draw_filled_rect()	80
4.26.1.14	renderer_draw_filled_rounded_rect()	80
4.26.1.15	renderer_draw_filled_triangle()	80
4.26.1.16	renderer_draw_line()	80
4.26.1.17	renderer_draw_pie()	81
4.26.1.18	renderer_draw_pixel()	81
4.26.1.19	renderer_draw_polygon()	81
4.26.1.20	renderer_draw_rect()	81
4.26.1.21	renderer_draw_rounded_rect()	81
4.26.1.22	renderer_draw_text()	81
4.26.1.23	renderer_draw_texture()	82
4.26.1.24	renderer_draw_triangle()	82
4.26.1.25	renderer_query_text_size()	82
4.26.1.26	renderer_reset_clip_rect()	82
4.26.1.27	renderer_resize_framebuffer()	82
4.26.1.28	renderer_set_clip_rect()	82
4.26.1.29	renderer_set_default_font()	82
4.27	src/renderer/renderer.h File Reference	83
4.27.1	Function Documentation	83
4.27.1.1	_renderer_set_target()	83
4.27.1.2	renderer_bind_framebuffer()	83
4.27.1.3	renderer_clear()	83
4.27.1.4	renderer_create_framebuffer()	83
4.27.1.5	renderer_draw_arc()	83
4.27.1.6	renderer_draw_bezier()	83
4.27.1.7	renderer_draw_circle()	84
4.27.1.8	renderer_draw_ellipse()	84
4.27.1.9	renderer_draw_filled_circle()	84
4.27.1.10	renderer_draw_filled_ellipse()	84
4.27.1.11	renderer_draw_filled_pie()	84
4.27.1.12	renderer_draw_filled_polygon()	84
4.27.1.13	renderer_draw_filled_rect()	85
4.27.1.14	renderer_draw_filled_rounded_rect()	85
4.27.1.15	renderer_draw_filled_triangle()	85
4.27.1.16	renderer_draw_line()	85
4.27.1.17	renderer_draw_pie()	85
4.27.1.18	renderer_draw_pixel()	86
4.27.1.19	renderer_draw_polygon()	86
4.27.1.20	renderer_draw_rect()	86
4.27.1.21	renderer_draw_rounded_rect()	86
4.27.1.22	renderer_draw_text()	86

4.27.1.23 <code>renderer_draw_texture()</code>	86
4.27.1.24 <code>renderer_draw_triangle()</code>	87
4.27.1.25 <code>renderer_query_text_size()</code>	87
4.27.1.26 <code>renderer_reset_clip_rect()</code>	87
4.27.1.27 <code>renderer_resize_framebuffer()</code>	87
4.27.1.28 <code>renderer_set_clip_rect()</code>	87
4.27.1.29 <code>renderer_set_default_font()</code>	87
4.28 <code>renderer.h</code>	88
4.29 <code>src/texture/texture.c</code> File Reference	88
4.29.1 Function Documentation	88
4.29.1.1 <code>_texture_add()</code>	88
4.29.1.2 <code>_texture_close()</code>	88
4.29.1.3 <code>_texture_init()</code>	88
4.29.1.4 <code>texture_load()</code>	89
4.30 <code>src/texture/texture.h</code> File Reference	89
4.30.1 Typedef Documentation	89
4.30.1.1 Texture	89
4.30.2 Function Documentation	89
4.30.2.1 <code>_texture_add()</code>	89
4.30.2.2 <code>_texture_close()</code>	89
4.30.2.3 <code>_texture_init()</code>	89
4.30.2.4 <code>texture_load()</code>	89
4.31 <code>texture.h</code>	90
4.32 <code>src/ui/ui.c</code> File Reference	90
4.32.1 Function Documentation	90
4.32.1.1 <code>_ui_close()</code>	90
4.32.1.2 <code>_ui_get_target()</code>	90
4.32.1.3 <code>_ui_handle_event()</code>	90
4.32.1.4 <code>_ui_init()</code>	90
4.32.1.5 <code>_ui_render()</code>	91
4.32.1.6 <code>_ui_set_target()</code>	91
4.32.1.7 <code>_ui_update()</code>	91
4.32.2 Variable Documentation	91
4.32.2.1 <code>target_ui_data</code>	91
4.33 <code>src/ui/ui.h</code> File Reference	91
4.33.1 Typedef Documentation	91
4.33.1.1 <code>UIData</code>	91
4.33.2 Function Documentation	91
4.33.2.1 <code>_ui_close()</code>	91
4.33.2.2 <code>_ui_get_target()</code>	91
4.33.2.3 <code>_ui_handle_event()</code>	92
4.33.2.4 <code>_ui_init()</code>	92

4.33.2.5 <code>_ui_render()</code>	92
4.33.2.6 <code>_ui_set_target()</code>	92
4.33.2.7 <code>_ui_update()</code>	92
4.34 <code>ui.h</code>	92
4.35 <code>src/ui/ui_constraint/ui_constraint.c</code> File Reference	93
4.35.1 Function Documentation	93
4.35.1.1 <code>constraints_from_string()</code>	93
4.35.1.2 <code>new_aspect_constraint()</code>	93
4.35.1.3 <code>new_center_constraint()</code>	93
4.35.1.4 <code>new_offset_constraint()</code>	93
4.35.1.5 <code>new_pixel_constraint()</code>	93
4.35.1.6 <code>new_relative_constraint()</code>	93
4.36 <code>src/ui/ui_constraint/ui_constraint.h</code> File Reference	93
4.36.1 Typedef Documentation	94
4.36.1.1 <code>ConstraintType</code>	94
4.36.1.2 <code>UIConstraint</code>	94
4.36.1.3 <code>UIConstraints</code>	94
4.36.2 Enumeration Type Documentation	94
4.36.2.1 <code>ConstraintType</code>	94
4.36.3 Function Documentation	94
4.36.3.1 <code>constraints_from_string()</code>	94
4.36.3.2 <code>new_aspect_constraint()</code>	94
4.36.3.3 <code>new_center_constraint()</code>	94
4.36.3.4 <code>new_offset_constraint()</code>	95
4.36.3.5 <code>new_pixel_constraint()</code>	95
4.36.3.6 <code>new_relative_constraint()</code>	95
4.37 <code>ui_constraint.h</code>	95
4.38 <code>src/ui/ui_element/ui_element.c</code> File Reference	95
4.38.1 Typedef Documentation	95
4.38.1.1 <code>_UIDropdownItem</code>	95
4.38.1.2 <code>_UISplitButtonItem</code>	96
4.38.2 Function Documentation	96
4.38.2.1 <code>_ui_container_destroy()</code>	96
4.38.2.2 <code>_ui_container_recalculate()</code>	96
4.38.2.3 <code>_ui_container_render()</code>	96
4.38.2.4 <code>_ui_container_update()</code>	96
4.38.2.5 <code>ui_create_button()</code>	96
4.38.2.6 <code>ui_create_checkbox()</code>	96
4.38.2.7 <code>ui_create_container()</code>	97
4.38.2.8 <code>ui_create_dropdown()</code>	97
4.38.2.9 <code>ui_create_imagebutton()</code>	97
4.38.2.10 <code>ui_create_label()</code>	97

4.38.2.11 ui_create_panel()	97
4.38.2.12 ui_create_slider()	97
4.38.2.13 ui_create_splitbutton()	98
4.38.2.14 ui_create_textbox()	98
4.38.2.15 ui_hide_element()	98
4.38.2.16 ui_show_element()	98
4.39 src/ui/ui_element/ui_element.h File Reference	98
4.39.1 Macro Definition Documentation	99
4.39.1.1 UITEXT_MAX_LENGTH	99
4.39.2 Typedef Documentation	99
4.39.2.1 MouseState	99
4.39.2.2 UIButton	99
4.39.2.3 UIButtonClick	99
4.39.2.4 UICheckbox	99
4.39.2.5 UICheckboxCheckedChanged	99
4.39.2.6 UIContainer	99
4.39.2.7 UIContainerSizeChanged	99
4.39.2.8 UIDropdownList	99
4.39.2.9 UIDropdownListSelectionChanged	100
4.39.2.10 UIElement	100
4.39.2.11 UIElementDestroy	100
4.39.2.12 UIElementRecalculate	100
4.39.2.13 UIElementRender	100
4.39.2.14 UIElementUpdate	100
4.39.2.15 UIImageButton	100
4.39.2.16 UIImageButtonClick	100
4.39.2.17 UILabel	100
4.39.2.18 UIPanel	100
4.39.2.19 UISlider	101
4.39.2.20 UISliderValueChanged	101
4.39.2.21 UISplitButton	101
4.39.2.22 UISplitButtonClicked	101
4.39.2.23 UITextbox	101
4.39.2.24 UITextboxTextChanged	101
4.39.3 Enumeration Type Documentation	101
4.39.3.1 MouseState	101
4.39.4 Function Documentation	101
4.39.4.1 _ui_container_destroy()	101
4.39.4.2 _ui_container_recalculate()	102
4.39.4.3 _ui_container_render()	102
4.39.4.4 _ui_container_update()	102
4.39.4.5 ui_create_button()	102

4.39.4.6 ui_create_checkbox()	102
4.39.4.7 ui_create_container()	102
4.39.4.8 ui_create_dropdown()	102
4.39.4.9 ui_create_imagebutton()	103
4.39.4.10 ui_create_label()	103
4.39.4.11 ui_create_panel()	103
4.39.4.12 ui_create_slider()	103
4.39.4.13 ui_create_splitbutton()	103
4.39.4.14 ui_create_textbox()	104
4.39.4.15 ui_hide_element()	104
4.39.4.16 ui_show_element()	104
4.40 ui_element.h	104
4.41 src/utils/math/math.c File Reference	107
4.41.1 Function Documentation	107
4.41.1.1 check_collision_point_rect()	107
4.41.1.2 clamp()	107
4.41.1.3 deg_to_rad()	107
4.41.1.4 lerp()	107
4.41.1.5 map()	107
4.41.1.6 rad_to_deg()	107
4.42 src/utils/math/math.h File Reference	108
4.42.1 Macro Definition Documentation	108
4.42.1.1 HALF_PI	108
4.42.1.2 PI	108
4.42.1.3 TWO_PI	108
4.42.2 Function Documentation	108
4.42.2.1 check_collision_point_rect()	108
4.42.2.2 clamp()	108
4.42.2.3 deg_to_rad()	108
4.42.2.4 lerp()	108
4.42.2.5 map()	109
4.42.2.6 rad_to_deg()	109
4.43 math.h	109
4.44 src/utils/vector/vector.c File Reference	109
4.44.1 Function Documentation	109
4.44.1.1 vector_clear()	109
4.44.1.2 vector_contains()	109
4.44.1.3 vector_create()	109
4.44.1.4 vector_destroy()	110
4.44.1.5 vector_get()	110
4.44.1.6 vector_index_of()	110
4.44.1.7 vector_insert()	110

4.44.1.8	vector_pop_back()	110
4.44.1.9	vector_push_back()	110
4.44.1.10	vector_remove()	110
4.44.1.11	vector_remove_at()	110
4.44.1.12	vector_reserve()	111
4.44.1.13	vector_size()	111
4.45	src/utlis/vector/vector.h File Reference	111
4.45.1	Typedef Documentation	111
4.45.1.1	Vector	111
4.45.2	Function Documentation	111
4.45.2.1	vector_clear()	111
4.45.2.2	vector_contains()	111
4.45.2.3	vector_create()	111
4.45.2.4	vector_destroy()	112
4.45.2.5	vector_get()	112
4.45.2.6	vector_index_of()	112
4.45.2.7	vector_insert()	112
4.45.2.8	vector_pop_back()	112
4.45.2.9	vector_push_back()	112
4.45.2.10	vector_remove()	112
4.45.2.11	vector_remove_at()	112
4.45.2.12	vector_reserve()	113
4.45.2.13	vector_size()	113
4.46	vector.h	113
4.47	src/window/window.c File Reference	113
4.47.1	Function Documentation	113
4.47.1.1	_window_close()	113
4.47.1.2	_window_handle_event()	113
4.47.1.3	_window_render()	114
4.47.1.4	_window_reset()	114
4.47.1.5	_window_update()	114
4.47.1.6	window_create()	114
4.47.1.7	window_focus()	114
4.47.1.8	window_get_main_container()	114
4.47.1.9	window_hide()	114
4.47.1.10	window_show()	114
4.48	src/window/window.h File Reference	115
4.48.1	Typedef Documentation	115
4.48.1.1	Window	115
4.48.2	Function Documentation	115
4.48.2.1	_window_close()	115
4.48.2.2	_window_handle_event()	115

4.48.2.3 _window_render()	115
4.48.2.4 _window_reset()	115
4.48.2.5 _window_update()	115
4.48.2.6 window_create()	116
4.48.2.7 window_focus()	116
4.48.2.8 window_get_main_container()	116
4.48.2.9 window_hide()	116
4.48.2.10 window_show()	116
4.49 window.h	116

Index	119
--------------	------------

Chapter 1

Class Index

1.1 Class List

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

AngleBisector	5
AppData	5
Circle	6
CoordinateSystem	7
Font	8
InputData	8
Line	9
Parallel	10
Perpendicular	10
Point	11
Shape	11
Tangent	12
Texture	12
UIButton	13
UICheckbox	14
UIConstraint	14
UIConstraints	15
UIContainer	15
UIData	16
UIDropdownList	17
UIElement	18
UIImageButton	19
UILabel	19
UIPanel	20
UISlider	21
UISplitButton	22
UITextbox	23
Vector	24
Vector2	24
Window	25

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

src/main.c	
This is the entry point of the application	73
src/app/app.c	27
src/app/app.h	28
src/color/color.c	30
src/color/color.h	31
src/font/font.c	34
src/font/font.h	35
src/geometry/coordinate_system/coordinate_system.c	36
src/geometry/coordinate_system/coordinate_system.h	43
src/geometry/intersection/intersection.c	50
src/geometry/intersection/intersection.h	51
src/geometry/shape/shape.c	51
src/geometry/shape/shape.h	56
src/geometry/vector2/vector2.c	63
src/geometry/vector2/vector2.h	66
src/input/input.c	69
src/input/input.h	71
src/renderer/renderer.c	78
src/renderer/renderer.h	83
src/texture/texture.c	88
src/texture/texture.h	89
src/ui/ui.c	90
src/ui/ui.h	91
src/ui/ui_constraint/ui_constraint.c	93
src/ui/ui_constraint/ui_constraint.h	93
src/ui/ui_element/ui_element.c	95
src/ui/ui_element/ui_element.h	98
src/utils/math/math.c	107
src/utils/math/math.h	108
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 AngleBisector Struct Reference

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

3.1.1 Member Data Documentation

3.1.1.1 base

```
Shape AngleBisector::base
```

3.1.1.2 line1

```
Line* AngleBisector::line1
```

3.1.1.3 line2

```
Line* AngleBisector::line2
```

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

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

3.2 AppData Struct Reference

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

3.2.1 Member Data Documentation

3.2.1.1 `delta_time`

```
double AppData::delta_time
```

3.2.1.2 `frame_start`

```
UInt32 AppData::frame_start
```

3.2.1.3 `last_frame_start`

```
UInt32 AppData::last_frame_start
```

3.2.1.4 `target_frame_time`

```
UInt32 AppData::target_frame_time
```

3.2.1.5 `windows`

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

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

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

3.3 Circle Struct Reference

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

3.3.1 Member Data Documentation

3.3.1.1 `base`

```
Shape Circle::base
```

3.3.1.2 `center`

```
Point* Circle::center
```

3.3.1.3 perimeter_point

```
Point* Circle::perimeter_point
```

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

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

3.4 CoordinateSystem Struct Reference

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

3.4.1 Member Data Documentation

3.4.1.1 intersection_points

```
Vector* CoordinateSystem::intersection_points
```

3.4.1.2 origin

```
Vector2 CoordinateSystem::origin
```

3.4.1.3 position

```
Vector2 CoordinateSystem::position
```

3.4.1.4 shapes

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

3.4.1.5 size

```
Vector2 CoordinateSystem::size
```

3.4.1.6 zoom

```
double CoordinateSystem::zoom
```

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

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

3.5 Font Struct Reference

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

3.5.1 Member Data Documentation

3.5.1.1 font

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

3.5.1.2 size

```
int Font::size
```

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

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

3.6 InputData Struct Reference

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

3.6.1 Member Data Documentation

3.6.1.1 current_keyboard_state

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

3.6.1.2 current_mouse_button_state

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

3.6.1.3 current_mouse_position

```
SDL_Point InputData::current_mouse_position
```

3.6.1.4 key_count

```
int InputData::key_count
```

3.6.1.5 mouse_wheel_delta

```
int InputData::mouse_wheel_delta
```

3.6.1.6 old_keyboard_state

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

3.6.1.7 old_mouse_button_state

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

3.6.1.8 old_mouse_position

```
SDL_Point InputData::old_mouse_position
```

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

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

3.7 Line Struct Reference

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

3.7.1 Member Data Documentation

3.7.1.1 base

```
Shape Line::base
```

3.7.1.2 p1

```
Point* Line::p1
```

3.7.1.3 p2

```
Point * Line::p2
```

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

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

3.8 Parallel Struct Reference

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

3.8.1 Member Data Documentation

3.8.1.1 base

```
Shape Parallel::base
```

3.8.1.2 line

```
Line* Parallel::line
```

3.8.1.3 point

```
Point* Parallel::point
```

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

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

3.9 Perpendicular Struct Reference

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

3.9.1 Member Data Documentation

3.9.1.1 base

```
Shape Perpendicular::base
```

3.9.1.2 line

```
Line* Perpendicular::line
```

3.9.1.3 point

```
Point* Perpendicular::point
```

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

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

3.10 Point Struct Reference

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

3.10.1 Member Data Documentation

3.10.1.1 base

```
Shape 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 Shape Struct Reference

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

3.11.1 Member Data Documentation

3.11.1.1 dragged

```
bool Shape::dragged
```

3.11.1.2 selected

```
bool Shape::selected
```

3.11.1.3 type

```
ShapeType Shape::type
```

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

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

3.12 Tangent Struct Reference

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

3.12.1 Member Data Documentation

3.12.1.1 base

```
Shape Tangent::base
```

3.12.1.2 circle

```
Circle* Tangent::circle
```

3.12.1.3 point

```
Point* Tangent::point
```

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

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

3.13 Texture Struct Reference

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

3.13.1 Member Data Documentation

3.13.1.1 height

```
int Texture::height
```

3.13.1.2 texture

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

3.13.1.3 width

```
int Texture::width
```

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

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

3.14 UIButton Struct Reference

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

3.14.1 Member Data Documentation

3.14.1.1 base

[UIElement](#) UIButton::base

3.14.1.2 color

[Color](#) UIButton::color

3.14.1.3 corner_radius

[Uint32](#) UIButton::corner_radius

3.14.1.4 mouse_state

[MouseState](#) UIButton::mouse_state

3.14.1.5 on_click

[UIButtonClick](#) UIButton::on_click

3.14.1.6 text

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

3.14.1.7 text_color

[Color](#) UIButton::text_color

3.14.1.8 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](#)

3.15 UICheckbox Struct Reference

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

3.15.1 Member Data Documentation

3.15.1.1 base

[UIElement](#) UICheckbox::base

3.15.1.2 checked

bool UICheckbox::checked

3.15.1.3 checked_color

[Color](#) UICheckbox::checked_color

3.15.1.4 corner_radius

UInt32 UICheckbox::corner_radius

3.15.1.5 mouse_state

[MouseState](#) UICheckbox::mouse_state

3.15.1.6 on_checked_changed

[UICheckboxCheckedChanged](#) UICheckbox::on_checked_changed

3.15.1.7 unchecked_color

[Color](#) UICheckbox::unchecked_color

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

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

3.16 UIConstraint Struct Reference

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

3.16.1 Member Data Documentation

3.16.1.1 `constraint_type`

`ConstraintType` `UIConstraint::constraint_type`

3.16.1.2 `value`

`double` `UIConstraint::value`

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

- `src/ui/ui_constraint/ui_constraint.h`

3.17 UIConstraints Struct Reference

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

3.17.1 Member Data Documentation

3.17.1.1 `height`

`UIConstraint` `UIConstraints::height`

3.17.1.2 `width`

`UIConstraint` `UIConstraints::width`

3.17.1.3 `x`

`UIConstraint` `UIConstraints::x`

3.17.1.4 `y`

`UIConstraint` `UIConstraints::y`

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

- `src/ui/ui_constraint/ui_constraint.h`

3.18 UIContainer Struct Reference

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

3.18.1 Member Data Documentation

3.18.1.1 base

`UIElement` `UIContainer::base`

3.18.1.2 children

`Vector*` `UIContainer::children`

3.18.1.3 on_size_changed

`UIContainerSizeChanged` `UIContainer::on_size_changed`

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

- `src/ui/ui_element/ui_element.h`

3.19 UIData Struct Reference

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

3.19.1 Member Data Documentation

3.19.1.1 backspace_pressed

`bool` `UIData::backspace_pressed`

3.19.1.2 expanded_splitbutton

`UISplitButton*` `UIData::expanded_splitbutton`

3.19.1.3 main_container

`UIContainer*` `UIData::main_container`

3.19.1.4 mouse_captured

`bool` `UIData::mouse_captured`

3.19.1.5 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](#)

3.20 UIDropdownList Struct Reference

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

3.20.1 Member Data Documentation

3.20.1.1 base

```
UIElement UIDropdownList::base
```

3.20.1.2 color

```
Color UIDropdownList::color
```

3.20.1.3 corner_radius

```
Uint32 UIDropdownList::corner_radius
```

3.20.1.4 expanded

```
bool UIDropdownList::expanded
```

3.20.1.5 items

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

3.20.1.6 on_selection_changed

```
UIDropdownListSelectionChanged UIDropdownList::on_selection_changed
```

3.20.1.7 selected_item

```
Uint32 UIDropdownList::selected_item
```

3.20.1.8 text_color

`Color UIDropdownList::text_color`

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

- `src/ui/ui_element/ui_element.h`

3.21 UIElement Struct Reference

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

3.21.1 Member Data Documentation

3.21.1.1 constraints

`UIConstraints UIElement::constraints`

3.21.1.2 destroy

`UIElementDestroy UIElement::destroy`

3.21.1.3 parent

`UIElement* UIElement::parent`

3.21.1.4 position

`SDL_Point UIElement::position`

3.21.1.5 recalculate

`UIElementRecalculate UIElement::recalculate`

3.21.1.6 render

`UIElementRender UIElement::render`

3.21.1.7 shown

`bool UIElement::shown`

3.21.1.8 size

```
SDL_Point UIElement::size
```

3.21.1.9 update

```
UIElementUpdate UIElement::update
```

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

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

3.22 UIButton Struct Reference

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

3.22.1 Member Data Documentation

3.22.1.1 base

```
UIElement UIButton::base
```

3.22.1.2 mouse_state

```
MouseState UIButton::mouse_state
```

3.22.1.3 on_click

```
UIButtonOnClick UIButton::on_click
```

3.22.1.4 texture

```
Texture* UIButton::texture
```

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

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

3.23 UILabel Struct Reference

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

3.23.1 Member Data Documentation

3.23.1.1 base

`UIElement UILabel::base`

3.23.1.2 color

`Color UILabel::color`

3.23.1.3 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`

3.24 UIPanel Struct Reference

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

3.24.1 Member Data Documentation

3.24.1.1 base

`UIElement UIPanel::base`

3.24.1.2 border_color

`Color UIPanel::border_color`

3.24.1.3 border_width

`UInt32 UIPanel::border_width`

3.24.1.4 color

`Color UIPanel::color`

3.24.1.5 corner_radius

```
UInt32 UIPanel::corner_radius
```

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

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

3.25 UISlider Struct Reference

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

3.25.1 Member Data Documentation

3.25.1.1 base

```
UIElement UISlider::base
```

3.25.1.2 color

```
Color UISlider::color
```

3.25.1.3 corner_radius

```
UInt32 UISlider::corner_radius
```

3.25.1.4 mouse_state

```
MouseState UISlider::mouse_state
```

3.25.1.5 on_value_changed

```
UISliderValueChanged UISlider::on_value_changed
```

3.25.1.6 slider_color

```
Color UISlider::slider_color
```

3.25.1.7 thickness

```
UInt32 UISlider::thickness
```

3.25.1.8 value

```
double UISlider::value
```

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

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

3.26 UISplitButton Struct Reference

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

3.26.1 Member Data Documentation

3.26.1.1 auto_dropdown

```
bool UISplitButton::auto_dropdown
```

3.26.1.2 base

```
UIElement UISplitButton::base
```

3.26.1.3 color

```
Color UISplitButton::color
```

3.26.1.4 corner_radius

```
UInt32 UISplitButton::corner_radius
```

3.26.1.5 expanded

```
bool UISplitButton::expanded
```

3.26.1.6 items

```
Vector* UISplitButton::items
```

3.26.1.7 on_item_clicked

```
UISplitButtonClicked UISplitButton::on_item_clicked
```

3.26.1.8 text_color

`Color UISplitButton::text_color`

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

- `src/ui/ui_element/ui_element.h`

3.27 UITextbox Struct Reference

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

3.27.1 Member Data Documentation

3.27.1.1 base

`UIElement UITextbox::base`

3.27.1.2 color

`Color UITextbox::color`

3.27.1.3 corner_radius

`UInt32 UITextbox::corner_radius`

3.27.1.4 focused

`bool UITextbox::focused`

3.27.1.5 mouse_state

`MouseState UITextbox::mouse_state`

3.27.1.6 on_text_changed

`UITextboxTextChanged UITextbox::on_text_changed`

3.27.1.7 text

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

3.27.1.8 text_color

```
Color UITextbox::text_color
```

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

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

3.28 Vector Struct Reference

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

3.28.1 Member Data Documentation

3.28.1.1 capacity

```
size_t Vector::capacity
```

3.28.1.2 data

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

3.28.1.3 size

```
size_t Vector::size
```

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

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

3.29 Vector2 Struct Reference

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

3.29.1 Member Data Documentation

3.29.1.1 x

```
double Vector2::x
```

3.29.1.2 y

```
double Vector2::y
```

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

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

3.30 Window Struct Reference

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

3.30.1 Member Data Documentation

3.30.1.1 close_requested

```
bool Window::close_requested
```

3.30.1.2 input_data

```
InputData Window::input_data
```

3.30.1.3 renderer

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

3.30.1.4 ui_data

```
UIData Window::ui_data
```

3.30.1.5 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

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_target()

```
Window * app_get_target ( )
```

4.1.1.5 app_get_time()

```
double app_get_time ( )
```

4.1.1.6 app_get_windows()

```
Vector * app_get_windows ( )
```

4.1.1.7 app_init()

```
void app_init ( )
```

4.1.1.8 app_render()

```
void app_render ( )
```

4.1.1.9 app_request_close()

```
void app_request_close ( )
```

4.1.1.10 app_set_target()

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

4.1.1.11 app_set_target_fps()

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

4.1.1.12 app_update()

```
void app_update ( )
```

4.2 src/app/app.h File Reference

Classes

- struct [AppData](#)

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_target()

```
Window * app_get_target ( )
```

4.2.2.5 app_get_time()

```
double app_get_time ( )
```

4.2.2.6 app_get_windows()

```
Vector * app_get_windows ( )
```

4.2.2.7 app_init()

```
void app_init ( )
```

4.2.2.8 app_render()

```
void app_render ( )
```

4.2.2.9 app_request_close()

```
void app_request_close ( )
```

4.2.2.10 app_set_target()

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

4.2.2.11 app_set_target_fps()

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

4.2.2.12 app_update()

```
void app_update ( )
```

4.3 app.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 "../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 Window* app_get_target();
00030 Vector* app_get_windows();
00031 double app_get_time();
00032 double app_get_delta_time();
00033
00034 //internal functions
00035 void _app_add_window(Window* window);
```

4.4 src/color/color.c File Reference

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

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

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 )
```

4.8 src/font/font.h File Reference

Classes

- struct [Font](#)

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

Functions

- `CoordinateSystem * coordinate_system_create` (`Vector2` position, `Vector2` size, `Vector2` origin)
- void `coordinate_system_clear` (`CoordinateSystem *cs`)
- void `coordinate_system_destroy` (`CoordinateSystem *cs`)
- void `coordinate_system_save` (`CoordinateSystem *cs`, const char *path)
- `CoordinateSystem * coordinate_system_load` (const char *path)
- `Vector2 screen_to_coordinates` (`CoordinateSystem *cs`, `Vector2` point)
- `Vector2 coordinates_to_screen` (`CoordinateSystem *cs`, `Vector2` point)
- bool `coordinate_system_is_hovered` (`CoordinateSystem *cs`, `Vector2` point)
- void `coordinate_system_select_shape` (`CoordinateSystem *cs`, `Shape *shape`)
- void `coordinate_system_deselect_shape` (`CoordinateSystem *cs`, `Shape *shape`)
- void `coordinate_system_select_all_shapes` (`CoordinateSystem *cs`)
- void `coordinate_system_drag_selected_shapes` (`CoordinateSystem *cs`, bool drag)
- `Shape * coordinate_system_get_hovered_shape` (`CoordinateSystem *cs`, `Vector2` point)
- `Vector * coordinate_system_get_selected_shapes` (`CoordinateSystem *cs`)
- void `coordinate_system_deselect_shapes` (`CoordinateSystem *cs`)
- void `coordinate_system_delete_selected_shapes` (`CoordinateSystem *cs`)
- void `coordinate_system_translate` (`CoordinateSystem *cs`, `Vector2` translation)
- void `coordinate_system_zoom` (`CoordinateSystem *cs`, double zoom)
- void `coordinate_system_update` (`CoordinateSystem *cs`)
- void `coordinate_system_draw` (`CoordinateSystem *cs`)
- void `coordinate_system_update_dimensions` (`CoordinateSystem *cs`, `Vector2` position, `Vector2` size)
- void `coordinate_system_destroy_shape` (`CoordinateSystem *cs`, `Shape *shape`)

4.10.1 Function Documentation

4.10.1.1 coordinate_system_clear()

```
void coordinate_system_clear (
    CoordinateSystem * cs )
```

Clears a coordinate system (removes all the shapes)

Parameters

<code>cs</code>	
-----------------	--

4.10.1.2 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

4.10.1.3 coordinate_system_delete_selected_shapes()

```
void coordinate_system_delete_selected_shapes (
    CoordinateSystem * cs )
```

Deletes all the selected shapes.

Parameters

<i>cs</i>	The coordinate system to delete the shapes in
-----------	---

4.10.1.4 coordinate_system_deselect_shape()

```
void coordinate_system_deselect_shape (
    CoordinateSystem * cs,
    Shape * shape )
```

Deselects a shape.

Parameters

<i>cs</i>	The coordinate system to deselect the shape in
<i>shape</i>	The shape to deselect

4.10.1.5 coordinate_system_deselect_shapes()

```
void coordinate_system_deselect_shapes (
    CoordinateSystem * cs )
```

Deselects all the selected shapes.

Parameters

<i>cs</i>	The coordinate system to deselect the shapes in
-----------	---

4.10.1.6 coordinate_system_destroy()

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

Destroys a coordinate system.

Parameters

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

4.10.1.7 coordinate_system_destroy_shape()

```
void coordinate_system_destroy_shape (
    CoordinateSystem * cs,
    Shape * shape )
```

Destroys a shape and removes it from the coordinate system (as well as the shapes it defined)

Parameters

<i>cs</i>	The coordinate system to remove the shape from
<i>shape</i>	The shape to remove

4.10.1.8 coordinate_system_drag_selected_shapes()

```
void coordinate_system_drag_selected_shapes (
    CoordinateSystem * cs,
    bool drag )
```

Sets the dragged shape.

Parameters

<i>cs</i>	The coordinate system to set the dragged shape in
<i>shape</i>	The shape to set as dragged

4.10.1.9 coordinate_system_draw()

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

Draws the coordinate system.

Parameters

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

4.10.1.10 coordinate_system_get_hovered_shape()

```
Shape * 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

Shape* The hovered shape (NULL if none)

4.10.1.11 coordinate_system_get_selected_shapes()

```
Vector * coordinate_system_get_selected_shapes (
    CoordinateSystem * cs )
```

Returns the selected shapes.

Parameters

<i>cs</i>	The coordinate system to retrieve to selected shapes from
-----------	---

Returns

Vector* A vector of the selected shapes

4.10.1.12 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

4.10.1.13 coordinate_system_load()

```
CoordinateSystem * coordinate_system_load (
    const char * path )
```

Loads a coordinate system from a file (loads the shapes from a .gae file)

Parameters

<i>path</i>	The path to load the coordinate system from
-------------	---

Returns

CoordinateSystem* The loaded coordinate system

4.10.1.14 coordinate_system_save()

```
void coordinate_system_save (
    CoordinateSystem * cs,
    const char * path )
```

Saves a coordinate system to a file (saves the shapes into a .gae file)

Parameters

<i>cs</i>	The coordinate system to save
<i>path</i>	The path to save the coordinate system to

4.10.1.15 coordinate_system_select_all_shapes()

```
void coordinate_system_select_all_shapes (
    CoordinateSystem * cs )
```

Selects all the shapes.

Parameters

<i>cs</i>	The coordinate system to select the shapes in
-----------	---

4.10.1.16 coordinate_system_select_shape()

```
void coordinate_system_select_shape (
    CoordinateSystem * cs,
    Shape * shape )
```

Selects a shape.

Parameters

<i>cs</i>	The coordinate system to select the shape in
<i>shape</i>	The shape to select

4.10.1.17 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)

4.10.1.18 coordinate_system_update()

```
void coordinate_system_update (
    CoordinateSystem * cs )
```

Updates the coordinate system and calculates the intersections.

Parameters

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

4.10.1.19 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

4.10.1.20 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

4.10.1.21 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

4.10.1.22 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

Classes

- struct [CoordinateSystem](#)

Functions

- [CoordinateSystem](#) * [coordinate_system_create](#) ([Vector2](#) position, [Vector2](#) size, [Vector2](#) origin)
- void [coordinate_system_clear](#) ([CoordinateSystem](#) *cs)
- void [coordinate_system_destroy](#) ([CoordinateSystem](#) *cs)
- void [coordinate_system_save](#) ([CoordinateSystem](#) *cs, const char *path)
- [CoordinateSystem](#) * [coordinate_system_load](#) (const char *path)
- [Vector2](#) [screen_to_coordinates](#) ([CoordinateSystem](#) *cs, [Vector2](#) point)
- [Vector2](#) [coordinates_to_screen](#) ([CoordinateSystem](#) *cs, [Vector2](#) point)
- bool [coordinate_system_is_hovered](#) ([CoordinateSystem](#) *cs, [Vector2](#) point)
- void [coordinate_system_select_shape](#) ([CoordinateSystem](#) *cs, [Shape](#) *shape)
- void [coordinate_system_deselect_shape](#) ([CoordinateSystem](#) *cs, [Shape](#) *shape)
- void [coordinate_system_select_all_shapes](#) ([CoordinateSystem](#) *cs)
- void [coordinate_system_drag_selected_shapes](#) ([CoordinateSystem](#) *cs, bool drag)
- [Shape](#) * [coordinate_system_get_hovered_shape](#) ([CoordinateSystem](#) *cs, [Vector2](#) point)
- [Vector](#) * [coordinate_system_get_selected_shapes](#) ([CoordinateSystem](#) *cs)
- void [coordinate_system_deselect_shapes](#) ([CoordinateSystem](#) *cs)
- void [coordinate_system_delete_selected_shapes](#) ([CoordinateSystem](#) *cs)
- void [coordinate_system_translate](#) ([CoordinateSystem](#) *cs, [Vector2](#) translation)
- void [coordinate_system_zoom](#) ([CoordinateSystem](#) *cs, double zoom)
- void [coordinate_system_update](#) ([CoordinateSystem](#) *cs)
- void [coordinate_system_draw](#) ([CoordinateSystem](#) *cs)
- void [coordinate_system_update_dimensions](#) ([CoordinateSystem](#) *cs, [Vector2](#) position, [Vector2](#) size)
- void [coordinate_system_destroy_shape](#) ([CoordinateSystem](#) *cs, [Shape](#) *shape)

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_clear()

```
void coordinate\_system\_clear (  
    CoordinateSystem * cs )
```

Clears a coordinate system (removes all the shapes)

Parameters

<i>cs</i>	
-----------	--

4.11.3.2 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

4.11.3.3 coordinate_system_delete_selected_shapes()

```
void coordinate_system_delete_selected_shapes (
    CoordinateSystem * cs )
```

Deletes all the selected shapes.

Parameters

<i>cs</i>	The coordinate system to delete the shapes in
-----------	---

4.11.3.4 coordinate_system_deselect_shape()

```
void coordinate_system_deselect_shape (
    CoordinateSystem * cs,
    Shape * shape )
```

Deselects a shape.

Parameters

<i>cs</i>	The coordinate system to deselect the shape in
<i>shape</i>	The shape to deselect

4.11.3.5 coordinate_system_deselect_shapes()

```
void coordinate_system_deselect_shapes (
    CoordinateSystem * cs )
```

Deselects all the selected shapes.

Parameters

<i>cs</i>	The coordinate system to deselect the shapes in
-----------	---

4.11.3.6 coordinate_system_destroy()

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

Destroys a coordinate system.

Parameters

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

4.11.3.7 coordinate_system_destroy_shape()

```
void coordinate_system_destroy_shape (
    CoordinateSystem * cs,
    Shape * shape )
```

Destroys a shape and removes it from the coordinate system (as well as the shapes it defined)

Parameters

<i>cs</i>	The coordinate system to remove the shape from
<i>shape</i>	The shape to remove

4.11.3.8 coordinate_system_drag_selected_shapes()

```
void coordinate_system_drag_selected_shapes (
    CoordinateSystem * cs,
    bool drag )
```

Sets the dragged shape.

Parameters

<i>cs</i>	The coordinate system to set the dragged shape in
<i>shape</i>	The shape to set as dragged

4.11.3.9 coordinate_system_draw()

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

Draws the coordinate system.

Parameters

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

4.11.3.10 coordinate_system_get_hovered_shape()

```
Shape * 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

Shape* The hovered shape (NULL if none)

4.11.3.11 coordinate_system_get_selected_shapes()

```
Vector * coordinate_system_get_selected_shapes (
    CoordinateSystem * cs )
```

Returns the selected shapes.

Parameters

<i>cs</i>	The coordinate system to retrieve to selected shapes from
-----------	---

Returns

Vector* A vector of the selected shapes

4.11.3.12 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

4.11.3.13 coordinate_system_load()

```
CoordinateSystem * coordinate_system_load (  
    const char * path )
```

Loads a coordinate system from a file (loads the shapes from a .gae file)

Parameters

<i>path</i>	The path to load the coordinate system from
-------------	---

Returns

CoordinateSystem* The loaded coordinate system

4.11.3.14 coordinate_system_save()

```
void coordinate_system_save (  
    CoordinateSystem * cs,  
    const char * path )
```

Saves a coordinate system to a file (saves the shapes into a .gae file)

Parameters

<i>cs</i>	The coordinate system to save
<i>path</i>	The path to save the coordinate system to

4.11.3.15 coordinate_system_select_all_shapes()

```
void coordinate_system_select_all_shapes (  
    CoordinateSystem * cs )
```

Selects all the shapes.

Parameters

<i>cs</i>	The coordinate system to select the shapes in
-----------	---

4.11.3.16 coordinate_system_select_shape()

```
void coordinate_system_select_shape (
    CoordinateSystem * cs,
    Shape * shape )
```

Selects a shape.

Parameters

<i>cs</i>	The coordinate system to select the shape in
<i>shape</i>	The shape to select

4.11.3.17 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)

4.11.3.18 coordinate_system_update()

```
void coordinate_system_update (
    CoordinateSystem * cs )
```

Updates the coordinate system and calculates the intersections.

Parameters

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

4.11.3.19 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

4.11.3.20 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

4.11.3.21 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

4.11.3.22 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     Vector* intersection_points;
00019 } CoordinateSystem;
00020
00029 CoordinateSystem* coordinate_system_create(Vector2 position, Vector2 size, Vector2 origin);
00035 void coordinate_system_clear(CoordinateSystem* cs);
00041 void coordinate_system_destroy(CoordinateSystem* cs);
00042
00049 void coordinate_system_save(CoordinateSystem* cs, const char* path);
00056 CoordinateSystem* coordinate_system_load(const char* path);
00057
00065 Vector2 screen_to_coordinates(CoordinateSystem* cs, Vector2 point);
00073 Vector2 coordinates_to_screen(CoordinateSystem* cs, Vector2 point);
00074
00081 bool coordinate_system_is_hovered(CoordinateSystem* cs, Vector2 point);
00088 void coordinate_system_select_shape(CoordinateSystem* cs, Shape* shape);
00095 void coordinate_system_deselect_shape(CoordinateSystem* cs, Shape* shape);
00101 void coordinate_system_select_all_shapes(CoordinateSystem* cs);
00108 void coordinate_system_drag_selected_shapes(CoordinateSystem* cs, bool drag);
00116 Shape* coordinate_system_get_hovered_shape(CoordinateSystem* cs, Vector2 point);
00123 Vector* coordinate_system_get_selected_shapes(CoordinateSystem* cs);
00129 void coordinate_system_deselect_shapes(CoordinateSystem* cs);
00135 void coordinate_system_delete_selected_shapes(CoordinateSystem* cs);
00142 void coordinate_system_translate(CoordinateSystem* cs, Vector2 translation);
00149 void coordinate_system_zoom(CoordinateSystem* cs, double zoom);
00155 void coordinate_system_update(CoordinateSystem* cs);
00161 void coordinate_system_draw(CoordinateSystem* cs);
00169 void coordinate_system_update_dimensions(CoordinateSystem* cs, Vector2 position, Vector2 size);
00176 void coordinate_system_destroy_shape(CoordinateSystem* cs, Shape* shape);

```

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

4.13.1 Macro Definition Documentation

4.13.1.1 EPSILON

```
#define EPSILON 0.0001
```

4.13.2 Function Documentation

4.13.2.1 intersection_get()

```

Vector * intersection_get (
    Shape * shape1,
    Shape * shape2 )

```


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

4.14.1 Function Documentation

4.14.1.1 intersection_get()

```
Vector * intersection_get (
    Shape * shape1,
    Shape * shape2 )
```

4.15 intersection.h

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

```
00001 #pragma once
00002
00003 #include "../shape/shape.h"
00004 #include "../utils/vector/vector.h"
00005
00006 Vector* intersection_get(Shape* shape1, Shape* shape2);
```

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

Functions

- [Point * point_create](#) (CoordinateSystem *cs, Vector2 coordinates)
- [Line * line_create](#) (CoordinateSystem *cs, Point *p1, Point *p2)
- [Circle * circle_create](#) (CoordinateSystem *cs, Point *center, Point *perimeter_point)
- [Parallel * parallel_create](#) (CoordinateSystem *cs, Line *line, Point *point)
- [Perpendicular * perpendicular_create](#) (CoordinateSystem *cs, Line *line, Point *point)
- [AngleBisector * angle_bisector_create](#) (CoordinateSystem *cs, Line *line1, Line *line2)
- [Tangent * tangent_create](#) (CoordinateSystem *cs, Circle *circle, Point *point)

4.16.1 Macro Definition Documentation

4.16.1.1 EPSILON

```
#define EPSILON 0.0001
```

4.16.2 Function Documentation

4.16.2.1 angle_bisector_create()

```
AngleBisector * angle_bisector_create (
    CoordinateSystem * cs,
    Line * line1,
    Line * line2 )
```

Creates an angle bisector in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the angle bisector in
<i>line1</i>	The first line to create the angle bisector to
<i>line2</i>	The second line to create the angle bisector to

Returns

AngleBisector* The created angle bisector

4.16.2.2 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

4.16.2.3 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

4.16.2.4 parallel_create()

```
Parallel * parallel_create (
    CoordinateSystem * cs,
    Line * line,
    Point * point )
```

Creates a parallel line in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the parallel line in
<i>line</i>	The line to create the parallel line to
<i>point</i>	The point the parallel line goes through

Returns

Parallel* The created parallel line

4.16.2.5 perpendicular_create()

```
Perpendicular * perpendicular_create (
    CoordinateSystem * cs,
    Line * line,
    Point * point )
```

Creates a perpendicular line in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the perpendicular line in
<i>line</i>	The line to create the perpendicular line to
<i>point</i>	The point the perpendicular line goes through

Returns

Perpendicular* The created perpendicular line

4.16.2.6 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

4.16.2.7 shape_destroy()

```
void shape_destroy (
    CoordinateSystem * cs,
    Shape * self )
```

4.16.2.8 shape_draw()

```
void shape_draw (
    CoordinateSystem * cs,
    Shape * self )
```

4.16.2.9 shape_is_defined_by()

```
bool shape_is_defined_by (
    Shape * self,
    Shape * shape )
```

4.16.2.10 shape_overlap_point()

```
bool shape_overlap_point (
    CoordinateSystem * cs,
    Shape * self,
    Vector2 point )
```

4.16.2.11 shape_translate()

```
void shape_translate (
    CoordinateSystem * cs,
    Shape * self,
    Vector2 translation )
```

4.16.2.12 shape_update()

```
void shape_update (
    CoordinateSystem * cs,
    Shape * self )
```

4.16.2.13 tangent_create()

```
Tangent * tangent_create (
    CoordinateSystem * cs,
    Circle * circle,
    Point * point )
```

Creates a tangent to a circle in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the tangent in
<i>circle</i>	The circle to create the tangent to
<i>point</i>	The point the tangent goes through

Returns

Tangent* The created tangent

4.16.3 Variable Documentation

4.16.3.1 shape_destroy_funcs

```
ShapeDestroy shape_destroy_funcs[ST_COUNT] = {_point_destroy, _line_destroy, _circle_destroy,
_parallel_destroy, _perpendicular_destroy, _angle_bisector_destroy, _tangent_destroy}
```

4.16.3.2 shape_draw_funcs

```
ShapeDraw shape_draw_funcs[ST_COUNT] = {_point_draw, _line_draw, _circle_draw, _parallel_draw,
_perpendicular_draw, _angle_bisector_draw, _tangent_draw}
```

4.16.3.3 shape_is_defined_by_funcs

```
ShapeIsDefinedBy shape_is_defined_by_funcs[ST_COUNT] = {_point_is_defined_by, _line_is_↵
defined_by, _circle_is_defined_by, _parallel_is_defined_by, _perpendicular_is_defined_by, _↵
angle_bisector_is_defined_by, _tangent_is_defined_by}
```

4.16.3.4 shape_overlap_point_funcs

```
ShapeOverlapPoint shape_overlap_point_funcs[ST_COUNT] = {_point_overlap, _line_overlap, _↵
circle_overlap, _parallel_overlap, _perpendicular_overlap, _angle_bisector_overlap, _tangent↵
_overlap}
```

4.16.3.5 shape_translate_funcs

```
ShapeTranslate shape_translate_funcs[ST_COUNT] = {_point_translate, _line_translate, _circle↵
_translate, _parallel_translate, _perpendicular_translate, _angle_bisector_translate, _↵
tangent_translate}
```

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

Classes

- struct [Shape](#)
- struct [Point](#)
- struct [Line](#)
- struct [Circle](#)
- struct [Parallel](#)
- struct [Perpendicular](#)
- struct [AngleBisector](#)
- struct [Tangent](#)

Functions

- [Point](#) * [point_create](#) ([CoordinateSystem](#) *cs, [Vector2](#) coordinates)
- [Line](#) * [line_create](#) ([CoordinateSystem](#) *cs, [Point](#) *p1, [Point](#) *p2)
- [Circle](#) * [circle_create](#) ([CoordinateSystem](#) *cs, [Point](#) *center, [Point](#) *perimeter_point)
- [Parallel](#) * [parallel_create](#) ([CoordinateSystem](#) *cs, [Line](#) *line, [Point](#) *point)
- [Perpendicular](#) * [perpendicular_create](#) ([CoordinateSystem](#) *cs, [Line](#) *line, [Point](#) *point)
- [AngleBisector](#) * [angle_bisector_create](#) ([CoordinateSystem](#) *cs, [Line](#) *line1, [Line](#) *line2)
- [Tangent](#) * [tangent_create](#) ([CoordinateSystem](#) *cs, [Circle](#) *circle, [Point](#) *point)

4.17.1 Macro Definition Documentation

4.17.1.1 OVERLAP_DISTANCE

```
#define OVERLAP_DISTANCE 5
```

4.17.2 Typedef Documentation

4.17.2.1 AngleBisector

```
typedef struct AngleBisector AngleBisector
```

4.17.2.2 Circle

```
typedef struct Circle Circle
```

4.17.2.3 CoordinateSystem

```
typedef struct CoordinateSystem CoordinateSystem
```

4.17.2.4 Line

```
typedef struct Line Line
```

4.17.2.5 Parallel

```
typedef struct Parallel Parallel
```

4.17.2.6 Perpendicular

```
typedef struct Perpendicular Perpendicular
```

4.17.2.7 Point

```
typedef struct Point Point
```

4.17.2.8 Shape

```
typedef struct Shape Shape
```

4.17.2.9 ShapeDestroy

```
typedef void(* ShapeDestroy) (struct CoordinateSystem *cs, struct Shape *self)
```

4.17.2.10 ShapeDraw

```
typedef void(* ShapeDraw) (struct CoordinateSystem *cs, struct Shape *self)
```

4.17.2.11 ShapeIsDefinedBy

```
typedef bool(* ShapeIsDefinedBy) (struct Shape *self, struct Shape *shape)
```

4.17.2.12 ShapeOverlapPoint

```
typedef bool(* ShapeOverlapPoint) (struct CoordinateSystem *cs, struct Shape *self, Vector2  
point)
```

4.17.2.13 ShapeTranslate

```
typedef void(* ShapeTranslate) (struct CoordinateSystem *cs, struct Shape *self, Vector2 translation)
```

4.17.2.14 ShapeType

```
typedef enum ShapeType ShapeType
```

4.17.2.15 Tangent

```
typedef struct Tangent Tangent
```

4.17.3 Enumeration Type Documentation

4.17.3.1 ShapeType

```
enum ShapeType
```

Enumerator

ST_POINT	
ST_LINE	
ST_CIRCLE	
ST_PARALLEL	
ST_PERPENDICULAR	
ST_ANGLE_BISECTOR	
ST_TANGENT	
ST_COUNT	

4.17.4 Function Documentation

4.17.4.1 angle_bisector_create()

```
AngleBisector * angle_bisector_create (
    CoordinateSystem * cs,
    Line * line1,
    Line * line2 )
```

Creates an angle bisector in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the angle bisector in
<i>line1</i>	The first line to create the angle bisector to
<i>line2</i>	The second line to create the angle bisector to

Returns

AngleBisector* The created angle bisector

4.17.4.2 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

4.17.4.3 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

4.17.4.4 parallel_create()

```
Parallel * parallel_create (
    CoordinateSystem * cs,
    Line * line,
    Point * point )
```

Creates a parallel line in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the parallel line in
<i>line</i>	The line to create the parallel line to
<i>point</i>	The point the parallel line goes through

Returns

Parallel* The created parallel line

4.17.4.5 perpendicular_create()

```
Perpendicular * perpendicular_create (
    CoordinateSystem * cs,
    Line * line,
    Point * point )
```

Creates a perpendicular line in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the perpendicular line in
<i>line</i>	The line to create the perpendicular line to
<i>point</i>	The point the perpendicular line goes through

Returns

Perpendicular* The created perpendicular line

4.17.4.6 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

4.17.4.7 shape_destroy()

```
void shape_destroy (
    CoordinateSystem * cs,
    Shape * self )
```

4.17.4.8 shape_draw()

```
void shape_draw (
    CoordinateSystem * cs,
    Shape * self )
```

4.17.4.9 shape_is_defined_by()

```
bool shape_is_defined_by (
    Shape * self,
    Shape * shape )
```

4.17.4.10 shape_overlap_point()

```
bool shape_overlap_point (
    CoordinateSystem * cs,
    Shape * self,
    Vector2 point )
```

4.17.4.11 shape_translate()

```
void shape_translate (
    CoordinateSystem * cs,
    Shape * self,
    Vector2 translation )
```

4.17.4.12 shape_update()

```
void shape_update (
    CoordinateSystem * cs,
    Shape * self )
```

4.17.4.13 tangent_create()

```
Tangent * tangent_create (
    CoordinateSystem * cs,
    Circle * circle,
    Point * point )
```

Creates a tangent to a circle in the coordinate system.

Parameters

<i>cs</i>	The coordinate system to create the tangent in
<i>circle</i>	The circle to create the tangent to
<i>point</i>	The point the tangent goes through

Returns

Tangent* The created tangent

4.18 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 Shape Shape;
00011
00012 typedef void (*ShapeDraw)(struct CoordinateSystem* cs, struct Shape* self);
00013 typedef void (*ShapeTranslate)(struct CoordinateSystem* cs, struct Shape* self, Vector2 translation);
00014 typedef void (*ShapeDestroy)(struct CoordinateSystem* cs, struct Shape* self);
00015 typedef bool (*ShapeOverlapPoint)(struct CoordinateSystem* cs, struct Shape* self, Vector2 point);
00016 typedef bool (*ShapeIsDefinedBy)(struct Shape* self, struct Shape* shape);
00017
00018 typedef enum ShapeType
00019 {
00020     ST_POINT,
00021     ST_LINE,
00022     ST_CIRCLE,
00023
00024     ST_PARALLEL,
00025     ST_PERPENDICULAR,
00026     ST_ANGLE_BISECTOR,
00027     ST_TANGENT,
00028
00029     ST_COUNT
00030 } ShapeType;
00031
00032 typedef struct Shape
00033 {
00034     ShapeType type;
00035     bool selected;
00036     bool dragged;
00037 } Shape;
00038
00039 typedef struct Point
00040 {
00041     Shape base;
00042     Vector2 coordinates;
00043 } Point;
00044
00045 typedef struct Line
00046 {
00047     Shape base;
00048     Point *p1, *p2;
00049 } Line;
00050
00051 typedef struct Circle
00052 {
00053     Shape base;
00054     Point* center;
00055     Point* perimeter_point;
00056 } Circle;
00057
00058 typedef struct Parallel
00059 {
00060     Shape base;
00061     Line* line;
00062     Point* point;
00063 } Parallel;
00064
00065 typedef struct Perpendicular
00066 {
00067     Shape base;
00068     Line* line;
00069     Point* point;
00070 } Perpendicular;
00071
00072 typedef struct AngleBisector
00073 {
00074     Shape base;
00075     Line* line1;
00076     Line* line2;
00077 } AngleBisector;
00078
00079 typedef struct Tangent
00080 {
00081     Shape base;
00082     Circle* circle;
00083     Point* point;
00084 } Tangent;
00085
00093 Point* point_create(CoordinateSystem* cs, Vector2 coordinates);
00102 Line* line_create(CoordinateSystem* cs, Point* p1, Point* p2);
00111 Circle* circle_create(CoordinateSystem* cs, Point* center, Point* perimeter_point);
00120 Parallel* parallel_create(CoordinateSystem* cs, Line* line, Point* point);
00129 Perpendicular* perpendicular_create(CoordinateSystem* cs, Line* line, Point* point);
00138 AngleBisector* angle_bisector_create(CoordinateSystem* cs, Line* line1, Line* line2);
00147 Tangent* tangent_create(CoordinateSystem* cs, Circle* circle, Point* point);
00148
00149 void shape_draw(CoordinateSystem* cs, Shape* self);
00150 void shape_update(CoordinateSystem* cs, Shape* self);

```

```
00151 void shape_translate(CoordinateSystem* cs, Shape* self, Vector2 translation);
00152 void shape_destroy(CoordinateSystem* cs, Shape* self);
00153 bool shape_overlap_point(CoordinateSystem* cs, Shape* self, Vector2 point);
00154 bool shape_is_defined_by(Shape* self, Shape* shape);
```

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

4.19.1 Function Documentation

4.19.1.1 vector2_add()

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

4.19.1.2 vector2_angle()

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

4.19.1.3 vector2_create()

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

4.19.1.4 vector2_distance()

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

4.19.1.5 vector2_divide()

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

4.19.1.6 vector2_dot()

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

4.19.1.7 vector2_down()

```
Vector2 vector2_down ( )
```

4.19.1.8 vector2_from_point()

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

4.19.1.9 vector2_from_polar()

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

4.19.1.10 vector2_left()

```
Vector2 vector2_left ( )
```

4.19.1.11 vector2_length()

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

4.19.1.12 vector2_multiply()

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

4.19.1.13 vector2_negate()

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

4.19.1.14 vector2_normalize()

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

4.19.1.15 vector2_one()

```
Vector2 vector2_one ( )
```

4.19.1.16 vector2_reflect()

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

4.19.1.17 vector2_right()

```
Vector2 vector2_right ( )
```

4.19.1.18 vector2_rotate()

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

4.19.1.19 vector2_rotate90()

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

4.19.1.20 vector2_scale()

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

4.19.1.21 vector2_subtract()

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

4.19.1.22 vector2_up()

```
Vector2 vector2_up ( )
```

4.19.1.23 vector2_zero()

```
Vector2 vector2_zero ( )
```

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

Classes

- struct [Vector2](#)

4.20.1 Typedef Documentation

4.20.1.1 Vector2

```
typedef struct Vector2 Vector2
```

4.20.2 Function Documentation

4.20.2.1 vector2_add()

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

4.20.2.2 vector2_angle()

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

4.20.2.3 vector2_create()

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

4.20.2.4 vector2_distance()

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

4.20.2.5 vector2_divide()

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


4.20.2.6 vector2_dot()

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

4.20.2.7 vector2_down()

```
Vector2 vector2_down ( )
```

4.20.2.8 vector2_from_point()

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

4.20.2.9 vector2_from_polar()

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

4.20.2.10 vector2_left()

```
Vector2 vector2_left ( )
```

4.20.2.11 vector2_length()

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

4.20.2.12 vector2_multiply()

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

4.20.2.13 vector2_negate()

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

4.20.2.14 vector2_normalize()

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

4.20.2.15 vector2_one()

```
Vector2 vector2_one ( )
```

4.20.2.16 vector2_reflect()

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

4.20.2.17 vector2_right()

```
Vector2 vector2_right ( )
```

4.20.2.18 vector2_rotate()

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

4.20.2.19 vector2_rotate90()

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

4.20.2.20 vector2_scale()

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

4.20.2.21 vector2_subtract()

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

4.20.2.22 vector2_up()

```
Vector2 vector2_up ( )
```

4.20.2.23 vector2_zero()

```
Vector2 vector2_zero ( )
```

4.21 vector2.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 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);

```

4.22 src/input/input.c File Reference

4.22.1 Function Documentation

4.22.1.1 _input_close()

```

void _input_close (
    InputData * input_data )

```

4.22.1.2 _input_handle_event()

```

void _input_handle_event (
    InputData * input_data,
    SDL_Event * event )

```

4.22.1.3 _input_init()

```

void _input_init (
    InputData * input_data )

```

4.22.1.4 `_input_reset()`

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

4.22.1.5 `_input_set_target()`

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

4.22.1.6 `input_get_mouse_motion()`

```
SDL_Point input_get_mouse_motion ( )
```

4.22.1.7 `input_get_mouse_position()`

```
SDL_Point input_get_mouse_position ( )
```

4.22.1.8 `input_get_mouse_wheel_delta()`

```
int input_get_mouse_wheel_delta ( )
```

4.22.1.9 `input_is_key_down()`

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

4.22.1.10 `input_is_key_pressed()`

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

4.22.1.11 `input_is_key_released()`

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

4.22.1.12 `input_is_mouse_button_down()`

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

4.22.1.13 input_is_mouse_button_pressed()

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

4.22.1.14 input_is_mouse_button_released()

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

4.23 src/input/input.h File Reference

Classes

- struct [InputData](#)

4.23.1 Typedef Documentation

4.23.1.1 InputData

```
typedef struct InputData InputData
```

4.23.2 Function Documentation

4.23.2.1 _input_close()

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

4.23.2.2 _input_handle_event()

```
void _input_handle_event (
    InputData * input_data,
    SDL_Event * event )
```

4.23.2.3 _input_init()

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

4.23.2.4 _input_reset()

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

4.23.2.5 `_input_set_target()`

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

4.23.2.6 `input_get_mouse_motion()`

```
SDL_Point input_get_mouse_motion ( )
```

4.23.2.7 `input_get_mouse_position()`

```
SDL_Point input_get_mouse_position ( )
```

4.23.2.8 `input_get_mouse_wheel_delta()`

```
int input_get_mouse_wheel_delta ( )
```

4.23.2.9 `input_is_key_down()`

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

4.23.2.10 `input_is_key_pressed()`

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

4.23.2.11 `input_is_key_released()`

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

4.23.2.12 `input_is_mouse_button_down()`

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

4.23.2.13 `input_is_mouse_button_pressed()`

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

4.23.2.14 input_is_mouse_button_released()

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

4.24 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);
```

4.25 src/main.c File Reference

4.25.1 Detailed Description

Author

Örs Mándli (mandliors@gmail.com)

Version

0.1

Date

2023-11-05

Copyright

Copyright (c) 2023

4.25.2 Macro Definition Documentation**4.25.2.1 FPS**

```
#define FPS 60
```

4.25.2.2 MOUSE_WHEEL_SENSITIVITY

```
#define MOUSE_WHEEL_SENSITIVITY 5
```

4.25.3 Typedef Documentation**4.25.3.1 State**

```
typedef enum State State
```

4.25.4 Enumeration Type Documentation**4.25.4.1 State**

```
enum State
```

Enumerator

STATE_POINTER	
STATE_CS_DRAGGED	
STATE_POINT	
STATE_LINE	
STATE_LINE_POINT1_PLACED	
STATE_CIRCLE	
STATE_CIRCLE_CENTER_PLACED	
STATE_PARALLEL	
STATE_PARALLEL_LINE_SELECTED	
STATE_PERPENDICULAR	
STATE_PERPENDICULAR_LINE_SELECTED	
STATE_ANGLE_BISECTOR	
STATE_ANGLE_BISECTOR_LINE1_SELECTED	
STATE_TANGENT	
STATE_TANGENT_LINE_SELECTED	
STATE_OPENING	
STATE_SAVEING	

4.25.5 Function Documentation

4.25.5.1 main()

```
int main (
    void )
```

4.25.5.2 on_angle_bisector_clicked() [1/2]

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

4.25.5.3 on_angle_bisector_clicked() [2/2]

```
void on_angle_bisector_clicked (
    UIButton * self )
```

4.25.5.4 on_cancel_button_clicked()

```
void on_cancel_button_clicked (
    UIButton * self )
```

4.25.5.5 on_canvas_size_changed()

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

4.25.5.6 on_circle_clicked() [1/2]

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

4.25.5.7 on_circle_clicked() [2/2]

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

4.25.5.8 on_editmenu_clicked() [1/2]

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

4.25.5.9 on_editmenu_clicked() [2/2]

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

4.25.5.10 on_filemenu_clicked() [1/2]

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

4.25.5.11 on_filemenu_clicked() [2/2]

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

4.25.5.12 on_line_clicked() [1/2]

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

4.25.5.13 on_line_clicked() [2/2]

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

4.25.5.14 on_open_button_clicked()

```
void on_open_button_clicked (
    UIButton * self )
```

4.25.5.15 on_parallel_clicked() [1/2]

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

4.25.5.16 on_parallel_clicked() [2/2]

```
void on_parallel_clicked (
    UIButton * self )
```

4.25.5.17 on_perpendicular_clicked() [1/2]

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

4.25.5.18 on_perpendicular_clicked() [2/2]

```
void on_perpendicular_clicked (
    UIButton * self )
```

4.25.5.19 on_point_clicked() [1/2]

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

4.25.5.20 on_point_clicked() [2/2]

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

4.25.5.21 on_pointer_clicked() [1/2]

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

4.25.5.22 on_pointer_clicked() [2/2]

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

4.25.5.23 on_save_button_clicked()

```
void on_save_button_clicked (
    UIButton * self )
```

4.25.5.24 on_tangent_clicked() [1/2]

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

4.25.5.25 on_tangent_clicked() [2/2]

```
void on_tangent_clicked (
    UIButton * self )
```

4.25.6 Variable Documentation

4.25.6.1 cs

```
CoordinateSystem* cs
```

4.25.6.2 state

```
State state = STATE_POINTER
```

4.26 src/renderer/renderer.c File Reference

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 src/renderer/renderer.h File Reference

4.27.1 Function Documentation

4.27.1.1 `_renderer_set_target()`

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

4.27.1.2 `renderer_bind_framebuffer()`

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

4.27.1.3 `renderer_clear()`

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

4.27.1.4 `renderer_create_framebuffer()`

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

4.27.1.5 `renderer_draw_arc()`

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

4.27.1.6 `renderer_draw_bezier()`

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

4.27.1.7 `renderer_draw_circle()`

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

4.27.1.8 `renderer_draw_ellipse()`

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

4.27.1.9 `renderer_draw_filled_circle()`

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

4.27.1.10 `renderer_draw_filled_ellipse()`

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

4.27.1.11 `renderer_draw_filled_pie()`

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

4.27.1.12 `renderer_draw_filled_polygon()`

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

4.27.1.13 `renderer_draw_filled_rect()`

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

4.27.1.14 `renderer_draw_filled_rounded_rect()`

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

4.27.1.15 `renderer_draw_filled_triangle()`

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

4.27.1.16 `renderer_draw_line()`

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

4.27.1.17 `renderer_draw_pie()`

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

4.27.1.18 `renderer_draw_pixel()`

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

4.27.1.19 `renderer_draw_polygon()`

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

4.27.1.20 `renderer_draw_rect()`

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

4.27.1.21 `renderer_draw_rounded_rect()`

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

4.27.1.22 `renderer_draw_text()`

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

4.27.1.23 `renderer_draw_texture()`

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

4.27.1.24 `renderer_draw_triangle()`

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

4.27.1.25 `renderer_query_text_size()`

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

4.27.1.26 `renderer_reset_clip_rect()`

```
void renderer_reset_clip_rect ( )
```

4.27.1.27 `renderer_resize_framebuffer()`

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

4.27.1.28 `renderer_set_clip_rect()`

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

4.27.1.29 `renderer_set_default_font()`

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

4.28 renderer.h

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

```
00001 #pragma once
00002
00003 #ifndef _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);
```

4.29 src/texture/texture.c File Reference

4.29.1 Function Documentation

4.29.1.1 _texture_add()

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

4.29.1.2 _texture_close()

```
void _texture_close ( )
```

4.29.1.3 _texture_init()

```
void _texture_init ( )
```

4.29.1.4 texture_load()

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

4.30 src/texture/texture.h File Reference

Classes

- struct [Texture](#)

4.30.1 Typedef Documentation

4.30.1.1 Texture

```
typedef struct Texture Texture
```

4.30.2 Function Documentation

4.30.2.1 _texture_add()

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

4.30.2.2 _texture_close()

```
void _texture_close ( )
```

4.30.2.3 _texture_init()

```
void _texture_init ( )
```

4.30.2.4 texture_load()

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

4.31 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();
```

4.32 src/ui/ui.c File Reference

4.32.1 Function Documentation

4.32.1.1 _ui_close()

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

4.32.1.2 _ui_get_target()

```
UIData * _ui_get_target ( )
```

4.32.1.3 _ui_handle_event()

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

4.32.1.4 _ui_init()

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


4.32.1.5 `_ui_render()`

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

4.32.1.6 `_ui_set_target()`

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

4.32.1.7 `_ui_update()`

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

4.32.2 Variable Documentation

4.32.2.1 `target_ui_data`

```
UIData* target_ui_data = NULL
```

4.33 src/ui/ui.h File Reference

Classes

- struct `UIData`

4.33.1 Typedef Documentation

4.33.1.1 `UIData`

```
typedef struct UIData UIData
```

4.33.2 Function Documentation

4.33.2.1 `_ui_close()`

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

4.33.2.2 `_ui_get_target()`

```
UIData * _ui_get_target ( )
```

4.33.2.3 `_ui_handle_event()`

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

4.33.2.4 `_ui_init()`

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

4.33.2.5 `_ui_render()`

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

4.33.2.6 `_ui_set_target()`

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

4.33.2.7 `_ui_update()`

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

4.34 `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();
```

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

4.35.1 Function Documentation

4.35.1.1 constraints_from_string()

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

4.35.1.2 new_aspect_constraint()

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

4.35.1.3 new_center_constraint()

```
UIConstraint new_center_constraint ( )
```

4.35.1.4 new_offset_constraint()

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

4.35.1.5 new_pixel_constraint()

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

4.35.1.6 new_relative_constraint()

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

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

Classes

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

4.36.1 Typedef Documentation

4.36.1.1 ConstraintType

```
typedef enum ConstraintType ConstraintType
```

4.36.1.2 UIConstraint

```
typedef struct UIConstraint UIConstraint
```

4.36.1.3 UIConstraints

```
typedef struct UIConstraints UIConstraints
```

4.36.2 Enumeration Type Documentation

4.36.2.1 ConstraintType

```
enum ConstraintType
```

Enumerator

CT_PIXEL	
CT_CENTER	
CT_RELATIVE	
CT_OFFSET	
CT_ASPECT	

4.36.3 Function Documentation

4.36.3.1 constraints_from_string()

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

4.36.3.2 new_aspect_constraint()

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

4.36.3.3 new_center_constraint()

```
UIConstraint new_center_constraint ( )
```

4.36.3.4 new_offset_constraint()

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

4.36.3.5 new_pixel_constraint()

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

4.36.3.6 new_relative_constraint()

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

4.37 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 UIConstraint;
00013 typedef struct UIConstraint
00014 {
00015     double value;
00016     ConstraintType constraint_type;
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);
```

4.38 src/ui/ui_element/ui_element.c File Reference**4.38.1 Typedef Documentation****4.38.1.1 _UIDropdownItem**

```
typedef struct _UIDropdownItem _UIDropdownItem
```

4.38.1.2 _UISplitButtonItem

```
typedef struct _UISplitButtonItem _UISplitButtonItem
```

4.38.2 Function Documentation

4.38.2.1 _ui_container_destroy()

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

4.38.2.2 _ui_container_recalculate()

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

4.38.2.3 _ui_container_render()

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

4.38.2.4 _ui_container_update()

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

4.38.2.5 ui_create_button()

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

4.38.2.6 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 )
```

4.38.2.7 ui_create_container()

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

4.38.2.8 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 )
```

4.38.2.9 ui_create_imagebutton()

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

4.38.2.10 ui_create_label()

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

4.38.2.11 ui_create_panel()

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

4.38.2.12 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 )
```

4.38.2.13 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 )
```

4.38.2.14 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 )
```

4.38.2.15 ui_hide_element()

```
void ui_hide_element (
    UIElement * self )
```

4.38.2.16 ui_show_element()

```
void ui_show_element (
    UIElement * self )
```

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

Classes

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

4.39.1 Macro Definition Documentation

4.39.1.1 UITEXT_MAX_LENGTH

```
#define UITEXT_MAX_LENGTH 50
```

4.39.2 Typedef Documentation

4.39.2.1 MouseState

```
typedef enum MouseState MouseState
```

4.39.2.2 UIButton

```
typedef struct UIButton UIButton
```

4.39.2.3 UIButtonClick

```
typedef void(* UIButtonClick) (UIButton *self)
```

4.39.2.4 UICheckbox

```
typedef struct UICheckbox UICheckbox
```

4.39.2.5 UICheckboxCheckedChanged

```
typedef void(* UICheckboxCheckedChanged) (UICheckbox *self, bool checked)
```

4.39.2.6 UIContainer

```
typedef struct UIContainer UIContainer
```

4.39.2.7 UIContainerSizeChanged

```
typedef void(* UIContainerSizeChanged) (UIContainer *self, SDL_Point size)
```

4.39.2.8 UIDropdownList

```
typedef struct UIDropdownList UIDropdownList
```

4.39.2.9 UIDropdownListSelectionChanged

```
typedef void(* UIDropdownListSelectionChanged) (UIDropdownList *self, Sint32 index)
```

4.39.2.10 UIElement

```
typedef struct UIElement UIElement
```

4.39.2.11 UIElementDestroy

```
typedef void(* UIElementDestroy) (UIElement *self)
```

4.39.2.12 UIElementRecalculate

```
typedef void(* UIElementRecalculate) (UIElement *sibling, UIElement *self)
```

4.39.2.13 UIElementRender

```
typedef void(* UIElementRender) (UIElement *self)
```

4.39.2.14 UIElementUpdate

```
typedef void(* UIElementUpdate) (UIElement *self)
```

4.39.2.15 UIImageButton

```
typedef struct UIImageButton UIImageButton
```

4.39.2.16 UIImageButtonClick

```
typedef void(* UIImageButtonClick) (UIImageButton *self)
```

4.39.2.17 UILabel

```
typedef struct UILabel UILabel
```

4.39.2.18 UIPanel

```
typedef struct UIPanel UIPanel
```

4.39.2.19 UISlider

```
typedef struct UISlider UISlider
```

4.39.2.20 UISliderValueChanged

```
typedef void(* UISliderValueChanged) (UISlider *self, double value)
```

4.39.2.21 UISplitButton

```
typedef struct UISplitButton UISplitButton
```

4.39.2.22 UISplitButtonClicked

```
typedef void(* UISplitButtonClicked) (UISplitButton *self, Sint32 index)
```

4.39.2.23 UITextbox

```
typedef struct UITextbox UITextbox
```

4.39.2.24 UITextboxTextChanged

```
typedef void(* UITextboxTextChanged) (UITextbox *self, const char *text)
```

4.39.3 Enumeration Type Documentation

4.39.3.1 MouseState

```
enum MouseState
```

Enumerator

MS_NONE	
MS_HOVER	
MS_PRESS	

4.39.4 Function Documentation

4.39.4.1 _ui_container_destroy()

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

4.39.4.2 _ui_container_recalculate()

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

4.39.4.3 _ui_container_render()

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

4.39.4.4 _ui_container_update()

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

4.39.4.5 ui_create_button()

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

4.39.4.6 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 )
```

4.39.4.7 ui_create_container()

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

4.39.4.8 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 )
```

4.39.4.9 ui_create_imagebutton()

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

4.39.4.10 ui_create_label()

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

4.39.4.11 ui_create_panel()

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

4.39.4.12 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 )
```

4.39.4.13 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 )
```

4.39.4.14 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 )
```

4.39.4.15 ui_hide_element()

```
void ui_hide_element (
    UIElement * self )
```

4.39.4.16 ui_show_element()

```
void ui_show_element (
    UIElement * self )
```

4.40 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 void (*UIElementUpdate) (UIElement* self);
00020 typedef void (*UIElementRecalculate) (UIElement* sibling, UIElement* self);
00021 typedef void (*UIElementRender) (UIElement* self);
00022 typedef void (*UIElementDestroy) (UIElement* self);
00023
00024 typedef struct UIElement
00025 {
00026     UIElement* parent;
00027     UIConstraints constraints;
00028     SDL_Point position;
00029     SDL_Point size;
00030     bool shown;
00031
00032     UIElementUpdate update;
00033     UIElementRecalculate recalculate;
00034     UIElementRender render;
00035     UIElementDestroy destroy;
00036 } UIElement;
00037
00038 typedef struct UIContainer UIContainer;
00039 typedef void (*UIContainerSizeChanged) (UIContainer* self, SDL_Point size);
00040
00041 typedef struct UIContainer
00042 {
00043     UIElement base;
```

```

00044
00045     Vector* children;
00046     UIContainerSizeChanged on_size_changed;
00047 } UIContainer;
00048
00049 typedef struct UIPanel
00050 {
00051     UIElement base;
00052
00053     Color color;
00054     Color border_color;
00055     Uint32 border_width;
00056     Uint32 corner_radius;
00057 } UIPanel;
00058
00059 typedef struct UILabel
00060 {
00061     UIElement base;
00062
00063     char text[UITEXT_MAX_LENGTH + 1];
00064     Color color;
00065 } UILabel;
00066
00067 typedef enum MouseState { MS_NONE = 0, MS_HOVER, MS_PRESS } MouseState;
00068 typedef struct UIButton UIButton;
00069 typedef void (*UIButtonClick) (UIButton* self);
00070
00071 typedef struct UIButton
00072 {
00073     UIElement base;
00074
00075     char text[UITEXT_MAX_LENGTH + 1];
00076     SDL_Point text_position;
00077     Color color;
00078     Color text_color;
00079     Uint32 corner_radius;
00080     MouseState mouse_state;
00081     UIButtonClick on_click;
00082 } UIButton;
00083
00084 typedef struct UIImageButton UIImageButton;
00085 typedef void (*UIImageButtonClick) (UIImageButton* self);
00086
00087 typedef struct UIImageButton
00088 {
00089     UIElement base;
00090
00091     Texture* texture;
00092     MouseState mouse_state;
00093     UIImageButtonClick on_click;
00094 } UIImageButton;
00095
00096 typedef struct UITextbox UITextbox;
00097 typedef void (*UITextboxTextChanged) (UITextbox* self, const char* text);
00098
00099 typedef struct UITextbox
00100 {
00101     UIElement base;
00102
00103     char text[UITEXT_MAX_LENGTH + 1];
00104     Color color;
00105     Color text_color;
00106     Uint32 corner_radius;
00107     bool focused;
00108     MouseState mouse_state;
00109     UITextboxTextChanged on_text_changed;
00110 } UITextbox;
00111
00112 typedef struct UICheckbox UICheckbox;
00113 typedef void (*UICheckboxCheckedChanged) (UICheckbox* self, bool checked);
00114
00115 typedef struct UICheckbox
00116 {
00117     UIElement base;
00118
00119     bool checked;
00120     Color checked_color;
00121     Color unchecked_color;
00122     Uint32 corner_radius;
00123     MouseState mouse_state;
00124     UICheckboxCheckedChanged on_checked_changed;
00125 } UICheckbox;
00126
00127 typedef struct UISlider UISlider;
00128 typedef void (*UISliderValueChanged) (UISlider* self, double value);
00129
00130 typedef struct UISlider

```

```

00131 {
00132     UIElement base;
00133
00134     double value;
00135     Color color;
00136     Color slider_color;
00137     Uint32 thickness;
00138     Uint32 corner_radius;
00139     MouseState mouse_state;
00140     UISliderValueChanged on_value_changed;
00141 } UISlider;
00142
00143 typedef struct UIDropdownList UIDropdownList;
00144 typedef void (*UIDropdownListSelectionChanged)(UIDropdownList* self, Sint32 index);
00145
00146 typedef struct UIDropdownList
00147 {
00148     UIElement base;
00149
00150     Vector* items;
00151     Uint32 selected_item;
00152     bool expanded;
00153     Color color;
00154     Color text_color;
00155     Uint32 corner_radius;
00156     UIDropdownListSelectionChanged on_selection_changed;
00157 } UIDropdownList;
00158
00159 typedef struct UISplitButton UISplitButton;
00160 typedef void (*UISplitButtonClicked)(UISplitButton* self, Sint32 index);
00161
00162 typedef struct UISplitButton
00163 {
00164     UIElement base;
00165
00166     Vector* items;
00167     bool expanded;
00168     Color color;
00169     Color text_color;
00170     Uint32 corner_radius;
00171     UISplitButtonClicked on_item_clicked;
00172     bool auto_dropdown;
00173 } UISplitButton;
00174
00175 //API functions
00176 UIContainer* ui_create_container(UIContainer* parent, UIConstraints constraints, void
(*on_size_changed)(UIContainer* self, SDL_Point size));
00177 UIPanel* ui_create_panel(UIContainer* parent, UIConstraints constraints, Color color, Color
border_color, Uint32 border_width, Uint32 roundness);
00178 UILabel* ui_create_label(UIContainer* parent, UIConstraints constraints, const char* text, Color
color);
00179 UIButton* ui_create_button(UIContainer* parent, UIConstraints constraints, const char* text, Color
color, Color text_color, void (*on_click)(UIButton* self));
00180 UIImageButton* ui_create_imagebutton(UIContainer* parent, UIConstraints constraints, Texture* texture,
void (*on_click)(UIImageButton* self));
00181 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));
00182 UICheckbox* ui_create_checkbox(UIContainer* parent, UIConstraints constraints, Color checked_color,
Color unchecked_color, void (*on_checked_changed)(UICheckbox* self, bool checked));
00183 UISlider* ui_create_slider(UIContainer* parent, UIConstraints constraints, double value, Color color,
Color slider_color, void (*on_value_changed)(UISlider* self, double value));
00184 UIDropdownList* ui_create_dropdown(UIContainer* parent, UIConstraints constraints, char* items, Color
color, Color text_color, void (*on_selection_changed)(UIDropdownList* self, Sint32 index));
00185 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);
00186
00187 void ui_show_element(UIElement* self);
00188 void ui_hide_element(UIElement* self);
00189
00190 //internal functions
00191 void _ui_container_update(UIElement* self);
00192 void _ui_container_recalculate(UIElement* sibling, UIElement* self);
00193 void _ui_container_render(UIElement* self);
00194 void _ui_container_destroy(UIElement* self);

```


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

4.41.1 Function Documentation

4.41.1.1 check_collision_point_rect()

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

4.41.1.2 clamp()

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

4.41.1.3 deg_to_rad()

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

4.41.1.4 lerp()

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

4.41.1.5 map()

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

4.41.1.6 rad_to_deg()

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

4.42 src/utils/math/math.h File Reference

4.42.1 Macro Definition Documentation

4.42.1.1 HALF_PI

```
#define HALF_PI 1.57079632679489661923
```

4.42.1.2 PI

```
#define PI 3.14159265358979323846
```

4.42.1.3 TWO_PI

```
#define TWO_PI 6.28318530717958647692
```

4.42.2 Function Documentation

4.42.2.1 check_collision_point_rect()

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

4.42.2.2 clamp()

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

4.42.2.3 deg_to_rad()

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

4.42.2.4 lerp()

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

4.42.2.5 map()

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

4.42.2.6 rad_to_deg()

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

4.43 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);
```

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

4.44.1 Function Documentation

4.44.1.1 vector_clear()

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

4.44.1.2 vector_contains()

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

4.44.1.3 vector_create()

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

4.44.1.4 vector_destroy()

```
void vector_destroy (
    Vector * vector )
```

4.44.1.5 vector_get()

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

4.44.1.6 vector_index_of()

```
int vector_index_of (
    Vector * vector,
    void * value )
```

4.44.1.7 vector_insert()

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

4.44.1.8 vector_pop_back()

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

4.44.1.9 vector_push_back()

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

4.44.1.10 vector_remove()

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

4.44.1.11 vector_remove_at()

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

4.44.1.12 vector_reserve()

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

4.44.1.13 vector_size()

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

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

Classes

- struct Vector

4.45.1 Typedef Documentation

4.45.1.1 Vector

```
typedef struct Vector Vector
```

4.45.2 Function Documentation

4.45.2.1 vector_clear()

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

4.45.2.2 vector_contains()

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

4.45.2.3 vector_create()

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

4.45.2.4 vector_destroy()

```
void vector_destroy (
    Vector * vector )
```

4.45.2.5 vector_get()

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

4.45.2.6 vector_index_of()

```
int vector_index_of (
    Vector * vector,
    void * value )
```

4.45.2.7 vector_insert()

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

4.45.2.8 vector_pop_back()

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

4.45.2.9 vector_push_back()

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

4.45.2.10 vector_remove()

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

4.45.2.11 vector_remove_at()

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

4.45.2.12 vector_reserve()

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

4.45.2.13 vector_size()

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

4.46 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_destroy(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 int vector_index_of(Vector* vector, void* value);
00020 void vector_remove_at(Vector* vector, size_t idx);
00021 void vector_remove(Vector* vector, void* value);
00022 void vector_reserve(Vector* vector, size_t capacity);
00023 size_t vector_size(Vector* vector);
00024 void vector_clear(Vector* vector);
```

4.47 src/window/window.c File Reference**4.47.1 Function Documentation****4.47.1.1 _window_close()**

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

4.47.1.2 _window_handle_event()

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

4.47.1.3 `_window_render()`

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

4.47.1.4 `_window_reset()`

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

4.47.1.5 `_window_update()`

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

4.47.1.6 `window_create()`

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

4.47.1.7 `window_focus()`

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

4.47.1.8 `window_get_main_container()`

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

4.47.1.9 `window_hide()`

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

4.47.1.10 `window_show()`

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


4.48 src/window/window.h File Reference

Classes

- struct [Window](#)

4.48.1 Typedef Documentation

4.48.1.1 Window

```
typedef struct Window Window
```

4.48.2 Function Documentation

4.48.2.1 _window_close()

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

4.48.2.2 _window_handle_event()

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

4.48.2.3 _window_render()

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

4.48.2.4 _window_reset()

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

4.48.2.5 _window_update()

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

4.48.2.6 window_create()

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

4.48.2.7 window_focus()

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

4.48.2.8 window_get_main_container()

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

4.48.2.9 window_hide()

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

4.48.2.10 window_show()

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

4.49 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](#)
 - [ui_element.c, 95](#)
 - [_UISplitButtonItem](#)
 - [ui_element.c, 95](#)
 - [_app_add_window](#)
 - [app.c, 27](#)
 - [app.h, 28](#)
 - [_font_close](#)
 - [font.c, 34](#)
 - [font.h, 35](#)
 - [_font_init](#)
 - [font.c, 34](#)
 - [font.h, 35](#)
 - [_input_close](#)
 - [input.c, 69](#)
 - [input.h, 71](#)
 - [_input_handle_event](#)
 - [input.c, 69](#)
 - [input.h, 71](#)
 - [_input_init](#)
 - [input.c, 69](#)
 - [input.h, 71](#)
 - [_input_reset](#)
 - [input.c, 69](#)
 - [input.h, 71](#)
 - [_input_set_target](#)
 - [input.c, 70](#)
 - [input.h, 71](#)
 - [_renderer_set_target](#)
 - [renderer.c, 78](#)
 - [renderer.h, 83](#)
 - [_texture_add](#)
 - [texture.c, 88](#)
 - [texture.h, 89](#)
 - [_texture_close](#)
 - [texture.c, 88](#)
 - [texture.h, 89](#)
 - [_texture_init](#)
 - [texture.c, 88](#)
 - [texture.h, 89](#)
 - [_ui_close](#)
 - [ui.c, 90](#)
 - [ui.h, 91](#)
 - [_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, 102](#)
 - [_ui_container_update](#)
 - [ui_element.c, 96](#)
 - [ui_element.h, 102](#)
 - [_ui_get_target](#)
 - [ui.c, 90](#)
 - [ui.h, 91](#)
 - [_ui_handle_event](#)
 - [ui.c, 90](#)
 - [ui.h, 91](#)
 - [_ui_init](#)
 - [ui.c, 90](#)
 - [ui.h, 92](#)
 - [_ui_render](#)
 - [ui.c, 90](#)
 - [ui.h, 92](#)
 - [_ui_set_target](#)
 - [ui.c, 91](#)
 - [ui.h, 92](#)
 - [_ui_update](#)
 - [ui.c, 91](#)
 - [ui.h, 92](#)
 - [_window_close](#)
 - [window.c, 113](#)
 - [window.h, 115](#)
 - [_window_handle_event](#)
 - [window.c, 113](#)
 - [window.h, 115](#)
 - [_window_render](#)
 - [window.c, 113](#)
 - [window.h, 115](#)
 - [_window_reset](#)
 - [window.c, 114](#)
 - [window.h, 115](#)
 - [_window_update](#)
 - [window.c, 114](#)
 - [window.h, 115](#)
 - [angle_bisector_create](#)
 - [shape.c, 51](#)
 - [shape.h, 58](#)
 - [AngleBisector, 5](#)
 - [base, 5](#)
 - [line1, 5](#)
 - [line2, 5](#)
 - [shape.h, 56](#)
 - [app.c](#)
 - [_app_add_window, 27](#)

- app_close, 27
- app_get_delta_time, 27
- app_get_target, 27
- app_get_time, 27
- app_get_windows, 27
- app_init, 27
- app_render, 28
- app_request_close, 28
- app_set_target, 28
- app_set_target_fps, 28
- app_update, 28
- app.h
 - _app_add_window, 28
 - app_close, 28
 - app_get_delta_time, 29
 - app_get_target, 29
 - app_get_time, 29
 - app_get_windows, 29
 - app_init, 29
 - app_render, 29
 - app_request_close, 29
 - app_set_target, 29
 - app_set_target_fps, 29
 - app_update, 29
 - AppData, 28
- app_close
 - app.c, 27
 - app.h, 28
- app_get_delta_time
 - app.c, 27
 - app.h, 29
- app_get_target
 - app.c, 27
 - app.h, 29
- app_get_time
 - app.c, 27
 - app.h, 29
- app_get_windows
 - app.c, 27
 - app.h, 29
- app_init
 - app.c, 27
 - app.h, 29
- app_render
 - app.c, 28
 - app.h, 29
- app_request_close
 - app.c, 28
 - app.h, 29
- app_set_target
 - app.c, 28
 - app.h, 29
- app_set_target_fps
 - app.c, 28
 - app.h, 29
- app_update
 - app.c, 28
 - app.h, 29
- AppData, 5
 - app.h, 28
 - delta_time, 6
 - frame_start, 6
 - last_frame_start, 6
 - target_frame_time, 6
 - windows, 6
- auto_dropdown
 - UISplitButton, 22
- backspace_pressed
 - UIData, 16
- base
 - AngleBisector, 5
 - Circle, 6
 - Line, 9
 - Parallel, 10
 - Perpendicular, 10
 - Point, 11
 - Tangent, 12
 - UIButton, 13
 - UICheckbox, 14
 - UIContainer, 16
 - UIDropdownList, 17
 - UIImageButton, 19
 - UILabel, 20
 - UIPanel, 20
 - UISlider, 21
 - UISplitButton, 22
 - UITextbox, 23
- BLACK
 - color.h, 31
- BLUE
 - color.h, 31
- border_color
 - UIPanel, 20
- border_width
 - UIPanel, 20
- capacity
 - Vector, 24
- center
 - Circle, 6
- check_collision_point_rect
 - math.c, 107
 - math.h, 108
- checked
 - UICheckbox, 14
- checked_color
 - UICheckbox, 14
- children
 - UIContainer, 16
- Circle, 6
 - base, 6
 - center, 6
 - perimeter_point, 6
 - shape.h, 56
- circle
 - Tangent, 12

- circle_create
 - shape.c, 52
 - shape.h, 58
- clamp
 - math.c, 107
 - math.h, 108
- close_requested
 - Window, 25
- Color
 - color.h, 33
- color
 - UIButton, 13
 - UIDropdownList, 17
 - UILabel, 20
 - UIPanel, 20
 - UISlider, 21
 - UISplitButton, 22
 - UITextbox, 23
- color.c
 - color_clever_shift, 30
 - color_fade, 30
 - color_from_grayscale, 30
 - color_from_hex, 31
 - color_from_hsv, 31
 - color_from_rgb, 31
 - color_from_rgba, 31
 - color_shift, 31
- color.h
 - BLACK, 31
 - BLUE, 31
 - Color, 33
 - color_clever_shift, 33
 - color_fade, 33
 - color_from_grayscale, 33
 - color_from_hex, 33
 - color_from_hsv, 33
 - color_from_rgb, 33
 - color_from_rgba, 33
 - color_shift, 34
 - CYAN, 32
 - DARK_GRAY, 32
 - GRAY, 32
 - GREEN, 32
 - MAGENTA, 32
 - RED, 32
 - TRANSPARENT, 32
 - WHITE, 32
 - YELLOW, 32
- color_clever_shift
 - color.c, 30
 - color.h, 33
- color_fade
 - color.c, 30
 - color.h, 33
- color_from_grayscale
 - color.c, 30
 - color.h, 33
- color_from_hex
 - color.c, 31
 - color.h, 33
- color_from_hsv
 - color.c, 31
 - color.h, 33
- color_from_rgb
 - color.c, 31
 - color.h, 33
- color_from_rgba
 - color.c, 31
 - color.h, 33
- color_shift
 - color.c, 31
 - color.h, 34
- constraint_type
 - UIConstraint, 15
- constraints
 - UIElement, 18
- constraints_from_string
 - ui_constraint.c, 93
 - ui_constraint.h, 94
- ConstraintType
 - ui_constraint.h, 94
- coordinate_system.c
 - coordinate_system_clear, 36
 - coordinate_system_create, 36
 - coordinate_system_delete_selected_shapes, 37
 - coordinate_system_deselect_shape, 37
 - coordinate_system_deselect_shapes, 37
 - coordinate_system_destroy, 37
 - coordinate_system_destroy_shape, 38
 - coordinate_system_drag_selected_shapes, 38
 - coordinate_system_draw, 38
 - coordinate_system_get_hovered_shape, 39
 - coordinate_system_get_selected_shapes, 39
 - coordinate_system_is_hovered, 39
 - coordinate_system_load, 39
 - coordinate_system_save, 40
 - coordinate_system_select_all_shapes, 40
 - coordinate_system_select_shape, 40
 - coordinate_system_translate, 41
 - coordinate_system_update, 41
 - coordinate_system_update_dimensions, 41
 - coordinate_system_zoom, 41
 - coordinates_to_screen, 42
 - screen_to_coordinates, 42
- coordinate_system.h
 - coordinate_system_clear, 43
 - coordinate_system_create, 44
 - coordinate_system_delete_selected_shapes, 44
 - coordinate_system_deselect_shape, 44
 - coordinate_system_deselect_shapes, 44
 - coordinate_system_destroy, 45
 - coordinate_system_destroy_shape, 45
 - coordinate_system_drag_selected_shapes, 45
 - coordinate_system_draw, 46
 - coordinate_system_get_hovered_shape, 46
 - coordinate_system_get_selected_shapes, 46

- coordinate_system_is_hovered, 46
- coordinate_system_load, 47
- coordinate_system_save, 47
- coordinate_system_select_all_shapes, 47
- coordinate_system_select_shape, 47
- coordinate_system_translate, 48
- coordinate_system_update, 48
- coordinate_system_update_dimensions, 48
- coordinate_system_zoom, 49
- coordinates_to_screen, 49
- CoordinateSystem, 43
- INITIAL_ZOOM, 43
- screen_to_coordinates, 49
- coordinate_system_clear
 - coordinate_system.c, 36
 - coordinate_system.h, 43
- coordinate_system_create
 - coordinate_system.c, 36
 - coordinate_system.h, 44
- coordinate_system_delete_selected_shapes
 - coordinate_system.c, 37
 - coordinate_system.h, 44
- coordinate_system_deselect_shape
 - coordinate_system.c, 37
 - coordinate_system.h, 44
- coordinate_system_deselect_shapes
 - coordinate_system.c, 37
 - coordinate_system.h, 44
- coordinate_system_destroy
 - coordinate_system.c, 37
 - coordinate_system.h, 45
- coordinate_system_destroy_shape
 - coordinate_system.c, 38
 - coordinate_system.h, 45
- coordinate_system_drag_selected_shapes
 - coordinate_system.c, 38
 - coordinate_system.h, 45
- coordinate_system_draw
 - coordinate_system.c, 38
 - coordinate_system.h, 46
- coordinate_system_get_hovered_shape
 - coordinate_system.c, 39
 - coordinate_system.h, 46
- coordinate_system_get_selected_shapes
 - coordinate_system.c, 39
 - coordinate_system.h, 46
- coordinate_system_is_hovered
 - coordinate_system.c, 39
 - coordinate_system.h, 46
- coordinate_system_load
 - coordinate_system.c, 39
 - coordinate_system.h, 47
- coordinate_system_save
 - coordinate_system.c, 40
 - coordinate_system.h, 47
- coordinate_system_select_all_shapes
 - coordinate_system.c, 40
 - coordinate_system.h, 47
- coordinate_system_select_shape
 - coordinate_system.c, 40
 - coordinate_system.h, 47
- coordinate_system_translate
 - coordinate_system.c, 41
 - coordinate_system.h, 48
- coordinate_system_update
 - coordinate_system.c, 41
 - coordinate_system.h, 48
- coordinate_system_update_dimensions
 - coordinate_system.c, 41
 - coordinate_system.h, 48
- coordinate_system_zoom
 - coordinate_system.c, 41
 - coordinate_system.h, 49
- coordinates
 - Point, 11
- coordinates_to_screen
 - coordinate_system.c, 42
 - coordinate_system.h, 49
- CoordinateSystem, 7
 - coordinate_system.h, 43
 - intersection_points, 7
 - origin, 7
 - position, 7
 - shape.h, 56
 - shapes, 7
 - size, 7
 - zoom, 7
- corner_radius
 - UIButton, 13
 - UICheckbox, 14
 - UIDropdownList, 17
 - UIPanel, 20
 - UISlider, 21
 - UISplitButton, 22
 - UITextbox, 23
- cs
 - main.c, 78
- CT_ASPECT
 - ui_constraint.h, 94
- CT_CENTER
 - ui_constraint.h, 94
- CT_OFFSET
 - ui_constraint.h, 94
- CT_PIXEL
 - ui_constraint.h, 94
- CT_RELATIVE
 - ui_constraint.h, 94
- current_keyboard_state
 - InputData, 8
- current_mouse_button_state
 - InputData, 8
- current_mouse_position
 - InputData, 8
- CYAN
 - color.h, 32
- DARK_GRAY

- color.h, 32
- data
 - Vector, 24
- deg_to_rad
 - math.c, 107
 - math.h, 108
- delta_time
 - AppData, 6
- destroy
 - UIElement, 18
- dragged
 - Shape, 11
- EPSILON
 - intersection.c, 50
 - shape.c, 51
- expanded
 - UIDropdownList, 17
 - UISplitButton, 22
- expanded_splitbutton
 - UIData, 16
- focused
 - UITextbox, 23
- Font, 8
 - font, 8
 - font.h, 35
 - size, 8
- font
 - Font, 8
- font.c
 - _font_close, 34
 - _font_init, 34
 - font_load, 34
- font.h
 - _font_close, 35
 - _font_init, 35
 - Font, 35
 - font_load, 35
- font_load
 - font.c, 34
 - font.h, 35
- FPS
 - main.c, 74
- frame_start
 - AppData, 6
- GRAY
 - color.h, 32
- GREEN
 - color.h, 32
- HALF_PI
 - math.h, 108
- height
 - Texture, 12
 - UIConstraints, 15
- INITIAL_ZOOM
 - coordinate_system.h, 43
- input.c
 - _input_close, 69
 - _input_handle_event, 69
 - _input_init, 69
 - _input_reset, 69
 - _input_set_target, 70
 - input_get_mouse_motion, 70
 - input_get_mouse_position, 70
 - input_get_mouse_wheel_delta, 70
 - input_is_key_down, 70
 - input_is_key_pressed, 70
 - input_is_key_released, 70
 - input_is_mouse_button_down, 70
 - input_is_mouse_button_pressed, 70
 - input_is_mouse_button_released, 71
- input.h
 - _input_close, 71
 - _input_handle_event, 71
 - _input_init, 71
 - _input_reset, 71
 - _input_set_target, 71
 - input_get_mouse_motion, 72
 - input_get_mouse_position, 72
 - input_get_mouse_wheel_delta, 72
 - input_is_key_down, 72
 - input_is_key_pressed, 72
 - input_is_key_released, 72
 - input_is_mouse_button_down, 72
 - input_is_mouse_button_pressed, 72
 - input_is_mouse_button_released, 72
 - InputData, 71
- input_data
 - Window, 25
- input_get_mouse_motion
 - input.c, 70
 - input.h, 72
- input_get_mouse_position
 - input.c, 70
 - input.h, 72
- input_get_mouse_wheel_delta
 - input.c, 70
 - input.h, 72
- input_is_key_down
 - input.c, 70
 - input.h, 72
- input_is_key_pressed
 - input.c, 70
 - input.h, 72
- input_is_key_released
 - input.c, 70
 - input.h, 72
- input_is_mouse_button_down
 - input.c, 70
 - input.h, 72
- input_is_mouse_button_pressed
 - input.c, 70
 - input.h, 72

- input_is_mouse_button_released
 - input.c, [71](#)
 - input.h, [72](#)
- InputData, [8](#)
 - current_keyboard_state, [8](#)
 - current_mouse_button_state, [8](#)
 - current_mouse_position, [8](#)
 - input.h, [71](#)
 - key_count, [8](#)
 - mouse_wheel_delta, [8](#)
 - old_keyboard_state, [9](#)
 - old_mouse_button_state, [9](#)
 - old_mouse_position, [9](#)
- intersection.c
 - EPSILON, [50](#)
 - intersection_get, [50](#)
- intersection.h
 - intersection_get, [51](#)
- intersection_get
 - intersection.c, [50](#)
 - intersection.h, [51](#)
- intersection_points
 - CoordinateSystem, [7](#)
- items
 - UIDropdownList, [17](#)
 - UISplitButton, [22](#)
- key_count
 - InputData, [8](#)
- last_frame_start
 - AppData, [6](#)
- lerp
 - math.c, [107](#)
 - math.h, [108](#)
- Line, [9](#)
 - base, [9](#)
 - p1, [9](#)
 - p2, [9](#)
 - shape.h, [56](#)
- line
 - Parallel, [10](#)
 - Perpendicular, [10](#)
- line1
 - AngleBisector, [5](#)
- line2
 - AngleBisector, [5](#)
- line_create
 - shape.c, [52](#)
 - shape.h, [59](#)
- MAGENTA
 - color.h, [32](#)
- main
 - main.c, [75](#)
- main.c
 - cs, [78](#)
 - FPS, [74](#)
 - main, [75](#)
- MOUSE_WHEEL_SENSITIVITY, [74](#)
 - on_angle_bisector_clicked, [75](#)
 - on_cancel_button_clicked, [75](#)
 - on_canvas_size_changed, [75](#)
 - on_circle_clicked, [75](#)
 - on_editmenu_clicked, [75](#)
 - on_filemenu_clicked, [76](#)
 - on_line_clicked, [76](#)
 - on_open_button_clicked, [76](#)
 - on_parallel_clicked, [76](#)
 - on_perpendicular_clicked, [76, 77](#)
 - on_point_clicked, [77](#)
 - on_pointer_clicked, [77](#)
 - on_save_button_clicked, [77](#)
 - on_tangent_clicked, [77](#)
 - State, [74](#)
 - state, [78](#)
 - STATE_ANGLE_BISECTOR, [74](#)
 - STATE_ANGLE_BISECTOR_LINE1_SELECTED, [74](#)
 - STATE_CIRCLE, [74](#)
 - STATE_CIRCLE_CENTER_PLACED, [74](#)
 - STATE_CS_DRAGGED, [74](#)
 - STATE_LINE, [74](#)
 - STATE_LINE_POINT1_PLACED, [74](#)
 - STATE_OPENING, [74](#)
 - STATE_PARALLEL, [74](#)
 - STATE_PARALLEL_LINE_SELECTED, [74](#)
 - STATE_PERPENDICULAR, [74](#)
 - STATE_PERPENDICULAR_LINE_SELECTED, [74](#)
 - STATE_POINT, [74](#)
 - STATE_POINTER, [74](#)
 - STATE_SAVEING, [74](#)
 - STATE_TANGENT, [74](#)
 - STATE_TANGENT_LINE_SELECTED, [74](#)
- main_container
 - UIData, [16](#)
- map
 - math.c, [107](#)
 - math.h, [108](#)
- math.c
 - check_collision_point_rect, [107](#)
 - clamp, [107](#)
 - deg_to_rad, [107](#)
 - lerp, [107](#)
 - map, [107](#)
 - rad_to_deg, [107](#)
- math.h
 - check_collision_point_rect, [108](#)
 - clamp, [108](#)
 - deg_to_rad, [108](#)
 - HALF_PI, [108](#)
 - lerp, [108](#)
 - map, [108](#)
 - PI, [108](#)
 - rad_to_deg, [109](#)
 - TWO_PI, [108](#)
- mouse_captured

- UIData, 16
- mouse_state
 - UIButton, 13
 - UICheckbox, 14
 - UIImageButton, 19
 - UISlider, 21
 - UITextbox, 23
- mouse_wheel_delta
 - InputData, 8
- MOUSE_WHEEL_SENSITIVITY
 - main.c, 74
- MouseState
 - ui_element.h, 99, 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, 93
 - ui_constraint.h, 94
- new_center_constraint
 - ui_constraint.c, 93
 - ui_constraint.h, 94
- new_offset_constraint
 - ui_constraint.c, 93
 - ui_constraint.h, 94
- new_pixel_constraint
 - ui_constraint.c, 93
 - ui_constraint.h, 95
- new_relative_constraint
 - ui_constraint.c, 93
 - ui_constraint.h, 95
- old_keyboard_state
 - InputData, 9
- old_mouse_button_state
 - InputData, 9
- old_mouse_position
 - InputData, 9
- on_angle_bisector_clicked
 - main.c, 75
- on_cancel_button_clicked
 - main.c, 75
- on_canvas_size_changed
 - main.c, 75
- on_checked_changed
 - UICheckbox, 14
- on_circle_clicked
 - main.c, 75
- on_click
 - UIButton, 13
 - UIImageButton, 19
- on_editmenu_clicked
 - main.c, 75
- on_filemenu_clicked
 - main.c, 76
- on_item_clicked
 - UISplitButton, 22
- on_line_clicked
 - main.c, 76
- on_open_button_clicked
 - main.c, 76
- on_parallel_clicked
 - main.c, 76
- on_perpendicular_clicked
 - main.c, 76, 77
- on_point_clicked
 - main.c, 77
- on_pointer_clicked
 - main.c, 77
- on_save_button_clicked
 - main.c, 77
- on_selection_changed
 - UIDropdownList, 17
- on_size_changed
 - UIContainer, 16
- on_tangent_clicked
 - main.c, 77
- on_text_changed
 - UITextbox, 23
- on_value_changed
 - UISlider, 21
- origin
 - CoordinateSystem, 7
- OVERLAP_DISTANCE
 - shape.h, 56
- p1
 - Line, 9
- p2
 - Line, 9
- Parallel, 10
 - base, 10
 - line, 10
 - point, 10
 - shape.h, 56
- parallel_create
 - shape.c, 52
 - shape.h, 59
- parent
 - UIElement, 18
- perimeter_point
 - Circle, 6
- Perpendicular, 10
 - base, 10
 - line, 10
 - point, 10
 - shape.h, 57
- perpendicular_create
 - shape.c, 53
 - shape.h, 59
- PI
 - math.h, 108
- Point, 11
 - base, 11

- coordinates, 11
- shape.h, 57
- point
 - Parallel, 10
 - Perpendicular, 10
 - Tangent, 12
- point_create
 - shape.c, 53
 - shape.h, 60
- position
 - CoordinateSystem, 7
 - UIElement, 18
- rad_to_deg
 - math.c, 107
 - math.h, 109
- recalculate
 - UIElement, 18
- RED
 - color.h, 32
- render
 - UIElement, 18
- renderer
 - Window, 25
- renderer.c
 - _renderer_set_target, 78
 - renderer_bind_framebuffer, 78
 - renderer_clear, 78
 - renderer_create_framebuffer, 78
 - renderer_draw_arc, 78
 - renderer_draw_bezier, 78
 - renderer_draw_circle, 79
 - renderer_draw_ellipse, 79
 - renderer_draw_filled_circle, 79
 - renderer_draw_filled_ellipse, 79
 - renderer_draw_filled_pie, 79
 - renderer_draw_filled_polygon, 79
 - renderer_draw_filled_rect, 80
 - renderer_draw_filled_rounded_rect, 80
 - renderer_draw_filled_triangle, 80
 - renderer_draw_line, 80
 - renderer_draw_pie, 80
 - renderer_draw_pixel, 81
 - renderer_draw_polygon, 81
 - renderer_draw_rect, 81
 - renderer_draw_rounded_rect, 81
 - renderer_draw_text, 81
 - renderer_draw_texture, 81
 - renderer_draw_triangle, 82
 - renderer_query_text_size, 82
 - renderer_reset_clip_rect, 82
 - renderer_resize_framebuffer, 82
 - renderer_set_clip_rect, 82
 - renderer_set_default_font, 82
- renderer.h
 - _renderer_set_target, 83
 - renderer_bind_framebuffer, 83
 - renderer_clear, 83
 - renderer_create_framebuffer, 83
 - renderer_draw_arc, 83
 - renderer_draw_bezier, 83
 - renderer_draw_circle, 83
 - renderer_draw_ellipse, 84
 - renderer_draw_filled_circle, 84
 - renderer_draw_filled_ellipse, 84
 - renderer_draw_filled_pie, 84
 - renderer_draw_filled_polygon, 84
 - renderer_draw_filled_rect, 85
 - renderer_draw_filled_rounded_rect, 85
 - renderer_draw_filled_triangle, 85
 - renderer_draw_line, 85
 - renderer_draw_pie, 85
 - renderer_draw_pixel, 85
 - renderer_draw_polygon, 86
 - renderer_draw_rect, 86
 - renderer_draw_rounded_rect, 86
 - renderer_draw_text, 86
 - renderer_draw_texture, 86
 - renderer_draw_triangle, 86
 - renderer_query_text_size, 87
 - renderer_reset_clip_rect, 87
 - renderer_resize_framebuffer, 87
 - renderer_set_clip_rect, 87
 - renderer_set_default_font, 87
- renderer_bind_framebuffer
 - renderer.c, 78
 - renderer.h, 83
- renderer_clear
 - renderer.c, 78
 - renderer.h, 83
- renderer_create_framebuffer
 - renderer.c, 78
 - renderer.h, 83
- renderer_draw_arc
 - renderer.c, 78
 - renderer.h, 83
- renderer_draw_bezier
 - renderer.c, 78
 - renderer.h, 83
- renderer_draw_circle
 - renderer.c, 79
 - renderer.h, 83
- renderer_draw_ellipse
 - renderer.c, 79
 - renderer.h, 84
- renderer_draw_filled_circle
 - renderer.c, 79
 - renderer.h, 84
- renderer_draw_filled_ellipse
 - renderer.c, 79
 - renderer.h, 84
- renderer_draw_filled_pie
 - renderer.c, 79
 - renderer.h, 84
- renderer_draw_filled_polygon
 - renderer.c, 79
 - renderer.h, 84

- renderer_draw_filled_rect
 - renderer.c, [80](#)
 - renderer.h, [84](#)
- renderer_draw_filled_rounded_rect
 - renderer.c, [80](#)
 - renderer.h, [85](#)
- renderer_draw_filled_triangle
 - renderer.c, [80](#)
 - renderer.h, [85](#)
- renderer_draw_line
 - renderer.c, [80](#)
 - renderer.h, [85](#)
- renderer_draw_pie
 - renderer.c, [80](#)
 - renderer.h, [85](#)
- renderer_draw_pixel
 - renderer.c, [81](#)
 - renderer.h, [85](#)
- renderer_draw_polygon
 - renderer.c, [81](#)
 - renderer.h, [86](#)
- renderer_draw_rect
 - renderer.c, [81](#)
 - renderer.h, [86](#)
- renderer_draw_rounded_rect
 - renderer.c, [81](#)
 - renderer.h, [86](#)
- renderer_draw_text
 - renderer.c, [81](#)
 - renderer.h, [86](#)
- renderer_draw_texture
 - renderer.c, [81](#)
 - renderer.h, [86](#)
- renderer_draw_triangle
 - renderer.c, [82](#)
 - renderer.h, [86](#)
- renderer_query_text_size
 - renderer.c, [82](#)
 - renderer.h, [87](#)
- renderer_reset_clip_rect
 - renderer.c, [82](#)
 - renderer.h, [87](#)
- renderer_resize_framebuffer
 - renderer.c, [82](#)
 - renderer.h, [87](#)
- renderer_set_clip_rect
 - renderer.c, [82](#)
 - renderer.h, [87](#)
- renderer_set_default_font
 - renderer.c, [82](#)
 - renderer.h, [87](#)
- screen_to_coordinates
 - coordinate_system.c, [42](#)
 - coordinate_system.h, [49](#)
- selected
 - Shape, [11](#)
- selected_item
 - UIDropdownList, [17](#)
- Shape, [11](#)
 - dragged, [11](#)
 - selected, [11](#)
 - shape.h, [57](#)
 - type, [11](#)
- shape.c
 - angle_bisector_create, [51](#)
 - circle_create, [52](#)
 - EPSILON, [51](#)
 - line_create, [52](#)
 - parallel_create, [52](#)
 - perpendicular_create, [53](#)
 - point_create, [53](#)
 - shape_destroy, [54](#)
 - shape_destroy_funcs, [55](#)
 - shape_draw, [54](#)
 - shape_draw_funcs, [55](#)
 - shape_is_defined_by, [54](#)
 - shape_is_defined_by_funcs, [55](#)
 - shape_overlap_point, [54](#)
 - shape_overlap_point_funcs, [55](#)
 - shape_translate, [54](#)
 - shape_translate_funcs, [55](#)
 - shape_update, [54](#)
 - tangent_create, [54](#)
- shape.h
 - angle_bisector_create, [58](#)
 - AngleBisector, [56](#)
 - Circle, [56](#)
 - circle_create, [58](#)
 - CoordinateSystem, [56](#)
 - Line, [56](#)
 - line_create, [59](#)
 - OVERLAP_DISTANCE, [56](#)
 - Parallel, [56](#)
 - parallel_create, [59](#)
 - Perpendicular, [57](#)
 - perpendicular_create, [59](#)
 - Point, [57](#)
 - point_create, [60](#)
 - Shape, [57](#)
 - shape_destroy, [60](#)
 - shape_draw, [60](#)
 - shape_is_defined_by, [60](#)
 - shape_overlap_point, [60](#)
 - shape_translate, [61](#)
 - shape_update, [61](#)
 - ShapeDestroy, [57](#)
 - ShapeDraw, [57](#)
 - ShapeIsDefinedBy, [57](#)
 - ShapeOverlapPoint, [57](#)
 - ShapeTranslate, [57](#)
 - ShapeType, [57](#)
 - ST_ANGLE_BISECTOR, [58](#)
 - ST_CIRCLE, [58](#)
 - ST_COUNT, [58](#)
 - ST_LINE, [58](#)
 - ST_PARALLEL, [58](#)

- ST_PERPENDICULAR, 58
- ST_POINT, 58
- ST_TANGENT, 58
- Tangent, 57
- tangent_create, 61
- shape_destroy
 - shape.c, 54
 - shape.h, 60
- shape_destroy_funcs
 - shape.c, 55
- shape_draw
 - shape.c, 54
 - shape.h, 60
- shape_draw_funcs
 - shape.c, 55
- shape_is_defined_by
 - shape.c, 54
 - shape.h, 60
- shape_is_defined_by_funcs
 - shape.c, 55
- shape_overlap_point
 - shape.c, 54
 - shape.h, 60
- shape_overlap_point_funcs
 - shape.c, 55
- shape_translate
 - shape.c, 54
 - shape.h, 61
- shape_translate_funcs
 - shape.c, 55
- shape_update
 - shape.c, 54
 - shape.h, 61
- ShapeDestroy
 - shape.h, 57
- ShapeDraw
 - shape.h, 57
- ShapeIsDefinedBy
 - shape.h, 57
- ShapeOverlapPoint
 - shape.h, 57
- shapes
 - CoordinateSystem, 7
- ShapeTranslate
 - shape.h, 57
- ShapeType
 - shape.h, 57
- shown
 - UIElement, 18
- size
 - CoordinateSystem, 7
 - Font, 8
 - UIElement, 18
 - Vector, 24
- slider_color
 - UISlider, 21
- src/app/app.c, 27
- src/app/app.h, 28, 30
- src/color/color.c, 30
- src/color/color.h, 31, 34
- src/font/font.c, 34
- src/font/font.h, 35
- src/geometry/coordinate_system/coordinate_system.c, 36
- src/geometry/coordinate_system/coordinate_system.h, 43, 50
- src/geometry/intersection/intersection.c, 50
- src/geometry/intersection/intersection.h, 51
- src/geometry/shape/shape.c, 51
- src/geometry/shape/shape.h, 56, 61
- src/geometry/vector2/vector2.c, 63
- src/geometry/vector2/vector2.h, 66, 69
- src/input/input.c, 69
- src/input/input.h, 71, 73
- src/main.c, 73
- src/renderer/renderer.c, 78
- src/renderer/renderer.h, 83, 88
- src/texture/texture.c, 88
- src/texture/texture.h, 89, 90
- src/ui/ui.c, 90
- src/ui/ui.h, 91, 92
- src/ui/ui_constraint/ui_constraint.c, 93
- src/ui/ui_constraint/ui_constraint.h, 93, 95
- src/ui/ui_element/ui_element.c, 95
- src/ui/ui_element/ui_element.h, 98, 104
- src/utils/math/math.c, 107
- src/utils/math/math.h, 108, 109
- src/utils/vector/vector.c, 109
- src/utils/vector/vector.h, 111, 113
- src/window/window.c, 113
- src/window/window.h, 115, 116
- ST_ANGLE_BISECTOR
 - shape.h, 58
- ST_CIRCLE
 - shape.h, 58
- ST_COUNT
 - shape.h, 58
- ST_LINE
 - shape.h, 58
- ST_PARALLEL
 - shape.h, 58
- ST_PERPENDICULAR
 - shape.h, 58
- ST_POINT
 - shape.h, 58
- ST_TANGENT
 - shape.h, 58
- State
 - main.c, 74
- state
 - main.c, 78
- STATE_ANGLE_BISECTOR
 - main.c, 74
- STATE_ANGLE_BISECTOR_LINE1_SELECTED
 - main.c, 74
- STATE_CIRCLE

- main.c, [74](#)
- STATE_CIRCLE_CENTER_PLACED
 - main.c, [74](#)
- STATE_CS_DRAGGED
 - main.c, [74](#)
- STATE_LINE
 - main.c, [74](#)
- STATE_LINE_POINT1_PLACED
 - main.c, [74](#)
- STATE_OPENING
 - main.c, [74](#)
- STATE_PARALLEL
 - main.c, [74](#)
- STATE_PARALLEL_LINE_SELECTED
 - main.c, [74](#)
- STATE_PERPENDICULAR
 - main.c, [74](#)
- STATE_PERPENDICULAR_LINE_SELECTED
 - main.c, [74](#)
- STATE_POINT
 - main.c, [74](#)
- STATE_POINTER
 - main.c, [74](#)
- STATE_SAVEING
 - main.c, [74](#)
- STATE_TANGENT
 - main.c, [74](#)
- STATE_TANGENT_LINE_SELECTED
 - main.c, [74](#)
- Tangent, [12](#)
 - base, [12](#)
 - circle, [12](#)
 - point, [12](#)
 - shape.h, [57](#)
- tangent_create
 - shape.c, [54](#)
 - shape.h, [61](#)
- target_frame_time
 - AppData, [6](#)
- target_ui_data
 - ui.c, [91](#)
- text
 - UIButton, [13](#)
 - UILabel, [20](#)
 - UITextbox, [23](#)
- text_color
 - UIButton, [13](#)
 - UIDropdownList, [17](#)
 - UISplitButton, [22](#)
 - UITextbox, [23](#)
- text_input
 - UIData, [16](#)
- text_position
 - UIButton, [13](#)
- Texture, [12](#)
 - height, [12](#)
 - texture, [12](#)
 - texture.h, [89](#)
 - width, [12](#)
- texture
 - Texture, [12](#)
 - UIButton, [19](#)
- texture.c
 - _texture_add, [88](#)
 - _texture_close, [88](#)
 - _texture_init, [88](#)
 - texture_load, [88](#)
- texture.h
 - _texture_add, [89](#)
 - _texture_close, [89](#)
 - _texture_init, [89](#)
 - Texture, [89](#)
 - texture_load, [89](#)
- texture_load
 - texture.c, [88](#)
 - texture.h, [89](#)
- thickness
 - UISlider, [21](#)
- TRANSPARENT
 - color.h, [32](#)
- TWO_PI
 - math.h, [108](#)
- type
 - Shape, [11](#)
- ui.c
 - _ui_close, [90](#)
 - _ui_get_target, [90](#)
 - _ui_handle_event, [90](#)
 - _ui_init, [90](#)
 - _ui_render, [90](#)
 - _ui_set_target, [91](#)
 - _ui_update, [91](#)
 - target_ui_data, [91](#)
- ui.h
 - _ui_close, [91](#)
 - _ui_get_target, [91](#)
 - _ui_handle_event, [91](#)
 - _ui_init, [92](#)
 - _ui_render, [92](#)
 - _ui_set_target, [92](#)
 - _ui_update, [92](#)
 - UIData, [91](#)
- ui_constraint.c
 - constraints_from_string, [93](#)
 - new_aspect_constraint, [93](#)
 - new_center_constraint, [93](#)
 - new_offset_constraint, [93](#)
 - new_pixel_constraint, [93](#)
 - new_relative_constraint, [93](#)
- ui_constraint.h
 - constraints_from_string, [94](#)
 - ConstraintType, [94](#)
 - CT_ASPECT, [94](#)
 - CT_CENTER, [94](#)
 - CT_OFFSET, [94](#)
 - CT_PIXEL, [94](#)

- CT_RELATIVE, 94
- new_aspect_constraint, 94
- new_center_constraint, 94
- new_offset_constraint, 94
- new_pixel_constraint, 95
- new_relative_constraint, 95
- UIConstraint, 94
- UIConstraints, 94
- ui_create_button
 - ui_element.c, 96
 - ui_element.h, 102
- 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, 103
- ui_create_panel
 - ui_element.c, 97
 - ui_element.h, 103
- 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, 25
- 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_hide_element, 98
 - ui_show_element, 98
- ui_element.h
 - _ui_container_destroy, 101
 - _ui_container_recalculate, 101
 - _ui_container_render, 102
 - _ui_container_update, 102
 - MouseState, 99, 101
 - MS_HOVER, 101
 - MS_NONE, 101
 - MS_PRESS, 101
 - ui_create_button, 102
 - ui_create_checkbox, 102
 - ui_create_container, 102
 - ui_create_dropdown, 102
 - ui_create_imagebutton, 102
 - ui_create_label, 103
 - ui_create_panel, 103
 - ui_create_slider, 103
 - ui_create_splitbutton, 103
 - ui_create_textbox, 103
 - ui_hide_element, 104
 - ui_show_element, 104
 - UIButton, 99
 - UIButtonClick, 99
 - UICheckbox, 99
 - UICheckboxCheckedChanged, 99
 - UIContainer, 99
 - UIContainerSizeChanged, 99
 - UIDropdownList, 99
 - UIDropdownListSelectionChanged, 99
 - UIElement, 100
 - UIElementDestroy, 100
 - UIElementRecalculate, 100
 - UIElementRender, 100
 - UIElementUpdate, 100
 - UIImageButton, 100
 - UIImageButtonClick, 100
 - UILabel, 100
 - UIPanel, 100
 - UISlider, 100
 - UISliderValueChanged, 101
 - UISplitButton, 101
 - UISplitButtonClicked, 101
 - UITEXT_MAX_LENGTH, 99
 - UITextbox, 101
 - UITextboxTextChanged, 101
- ui_hide_element
 - ui_element.c, 98
 - ui_element.h, 104
- ui_show_element
 - ui_element.c, 98
 - ui_element.h, 104
- UIButton, 13
 - base, 13
 - color, 13
 - corner_radius, 13
 - mouse_state, 13
 - on_click, 13

- text, 13
- text_color, 13
- text_position, 13
- ui_element.h, 99
- UIButtonClick
 - ui_element.h, 99
- UICheckbox, 14
 - base, 14
 - checked, 14
 - checked_color, 14
 - corner_radius, 14
 - mouse_state, 14
 - on_checked_changed, 14
 - ui_element.h, 99
 - unchecked_color, 14
- UICheckboxCheckedChanged
 - ui_element.h, 99
- UIConstraint, 14
 - constraint_type, 15
 - ui_constraint.h, 94
 - value, 15
- UIConstraints, 15
 - height, 15
 - ui_constraint.h, 94
 - width, 15
 - x, 15
 - y, 15
- UIContainer, 15
 - base, 16
 - children, 16
 - on_size_changed, 16
 - ui_element.h, 99
- UIContainerSizeChanged
 - ui_element.h, 99
- UIData, 16
 - backspace_pressed, 16
 - expanded_splitbutton, 16
 - main_container, 16
 - mouse_captured, 16
 - text_input, 16
 - ui.h, 91
- UIDropdownList, 17
 - base, 17
 - color, 17
 - corner_radius, 17
 - expanded, 17
 - items, 17
 - on_selection_changed, 17
 - selected_item, 17
 - text_color, 17
 - ui_element.h, 99
- UIDropdownListSelectionChanged
 - ui_element.h, 99
- UIElement, 18
 - constraints, 18
 - destroy, 18
 - parent, 18
 - position, 18
 - recalculate, 18
 - render, 18
 - shown, 18
 - size, 18
 - ui_element.h, 100
 - update, 19
- UIElementDestroy
 - ui_element.h, 100
- UIElementRecalculate
 - ui_element.h, 100
- UIElementRender
 - ui_element.h, 100
- UIElementUpdate
 - ui_element.h, 100
- UIImageButton, 19
 - base, 19
 - mouse_state, 19
 - on_click, 19
 - texture, 19
 - ui_element.h, 100
- UIImageButtonClick
 - ui_element.h, 100
- UILabel, 19
 - base, 20
 - color, 20
 - text, 20
 - ui_element.h, 100
- UIPanel, 20
 - base, 20
 - border_color, 20
 - border_width, 20
 - color, 20
 - corner_radius, 20
 - ui_element.h, 100
- UISlider, 21
 - base, 21
 - color, 21
 - corner_radius, 21
 - mouse_state, 21
 - on_value_changed, 21
 - slider_color, 21
 - thickness, 21
 - ui_element.h, 100
 - value, 21
- UISliderValueChanged
 - ui_element.h, 101
- UISplitButton, 22
 - auto_dropdown, 22
 - base, 22
 - color, 22
 - corner_radius, 22
 - expanded, 22
 - items, 22
 - on_item_clicked, 22
 - text_color, 22
 - ui_element.h, 101
- UISplitButtonClicked
 - ui_element.h, 101

- UITEXT_MAX_LENGTH
 - ui_element.h, 99
- UITextbox, 23
 - base, 23
 - color, 23
 - corner_radius, 23
 - focused, 23
 - mouse_state, 23
 - on_text_changed, 23
 - text, 23
 - text_color, 23
 - ui_element.h, 101
- UITextboxTextChanged
 - ui_element.h, 101
- unchecked_color
 - UICheckbox, 14
- update
 - UIElement, 19
- value
 - UIConstraint, 15
 - UISlider, 21
- Vector, 24
 - capacity, 24
 - data, 24
 - size, 24
 - vector.h, 111
- vector.c
 - vector_clear, 109
 - vector_contains, 109
 - vector_create, 109
 - vector_destroy, 109
 - vector_get, 110
 - vector_index_of, 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, 111
 - vector_contains, 111
 - vector_create, 111
 - vector_destroy, 111
 - vector_get, 112
 - vector_index_of, 112
 - vector_insert, 112
 - vector_pop_back, 112
 - vector_push_back, 112
 - vector_remove, 112
 - vector_remove_at, 112
 - vector_reserve, 112
 - vector_size, 113
- Vector2, 24
 - vector2.h, 66
 - x, 24
 - y, 24
- vector2.c
 - vector2_add, 63
 - vector2_angle, 63
 - vector2_create, 63
 - vector2_distance, 63
 - vector2_divide, 63
 - vector2_dot, 63
 - vector2_down, 63
 - vector2_from_point, 64
 - vector2_from_polar, 64
 - vector2_left, 64
 - vector2_length, 64
 - vector2_multiply, 64
 - vector2_negate, 64
 - vector2_normalize, 64
 - vector2_one, 64
 - vector2_reflect, 64
 - vector2_right, 65
 - vector2_rotate, 65
 - vector2_rotate90, 65
 - vector2_scale, 65
 - vector2_subtract, 65
 - vector2_up, 65
 - vector2_zero, 65
- vector2.h
 - Vector2, 66
 - vector2_add, 66
 - vector2_angle, 66
 - vector2_create, 66
 - vector2_distance, 66
 - vector2_divide, 66
 - vector2_dot, 66
 - vector2_down, 67
 - vector2_from_point, 67
 - vector2_from_polar, 67
 - vector2_left, 67
 - vector2_length, 67
 - vector2_multiply, 67
 - vector2_negate, 67
 - vector2_normalize, 67
 - vector2_one, 67
 - vector2_reflect, 68
 - vector2_right, 68
 - vector2_rotate, 68
 - vector2_rotate90, 68
 - vector2_scale, 68
 - vector2_subtract, 68
 - vector2_up, 68
 - vector2_zero, 68
- vector2_add
 - vector2.c, 63
 - vector2.h, 66
- vector2_angle
 - vector2.c, 63
 - vector2.h, 66
- vector2_create
 - vector2.c, 63

- vector2.h, 66
- vector2_distance
 - vector2.c, 63
 - vector2.h, 66
- vector2_divide
 - vector2.c, 63
 - vector2.h, 66
- vector2_dot
 - vector2.c, 63
 - vector2.h, 66
- vector2_down
 - vector2.c, 63
 - vector2.h, 67
- vector2_from_point
 - vector2.c, 64
 - vector2.h, 67
- vector2_from_polar
 - vector2.c, 64
 - vector2.h, 67
- vector2_left
 - vector2.c, 64
 - vector2.h, 67
- vector2_length
 - vector2.c, 64
 - vector2.h, 67
- vector2_multiply
 - vector2.c, 64
 - vector2.h, 67
- vector2_negate
 - vector2.c, 64
 - vector2.h, 67
- vector2_normalize
 - vector2.c, 64
 - vector2.h, 67
- vector2_one
 - vector2.c, 64
 - vector2.h, 67
- vector2_reflect
 - vector2.c, 64
 - vector2.h, 68
- vector2_right
 - vector2.c, 65
 - vector2.h, 68
- vector2_rotate
 - vector2.c, 65
 - vector2.h, 68
- vector2_rotate90
 - vector2.c, 65
 - vector2.h, 68
- vector2_scale
 - vector2.c, 65
 - vector2.h, 68
- vector2_subtract
 - vector2.c, 65
 - vector2.h, 68
- vector2_up
 - vector2.c, 65
 - vector2.h, 68
- vector2_zero
 - vector2.c, 65
 - vector2.h, 68
- vector_clear
 - vector.c, 109
 - vector.h, 111
- vector_contains
 - vector.c, 109
 - vector.h, 111
- vector_create
 - vector.c, 109
 - vector.h, 111
- vector_destroy
 - vector.c, 109
 - vector.h, 111
- vector_get
 - vector.c, 110
 - vector.h, 112
- vector_index_of
 - 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, 112
- vector_reserve
 - vector.c, 110
 - vector.h, 112
- vector_size
 - vector.c, 111
 - vector.h, 113
- WHITE
 - color.h, 32
- width
 - Texture, 12
 - UIConstraints, 15
- Window, 25
 - close_requested, 25
 - input_data, 25
 - renderer, 25
 - ui_data, 25
 - window, 25
 - window.h, 115
- window
 - Window, 25
- window.c
 - _window_close, 113

- [_window_handle_event](#), 113
 - [_window_render](#), 113
 - [_window_reset](#), 114
 - [_window_update](#), 114
 - [window_create](#), 114
 - [window_focus](#), 114
 - [window_get_main_container](#), 114
 - [window_hide](#), 114
 - [window_show](#), 114
- window.h
 - [_window_close](#), 115
 - [_window_handle_event](#), 115
 - [_window_render](#), 115
 - [_window_reset](#), 115
 - [_window_update](#), 115
 - [Window](#), 115
 - [window_create](#), 115
 - [window_focus](#), 116
 - [window_get_main_container](#), 116
 - [window_hide](#), 116
 - [window_show](#), 116
- window_create
 - [window.c](#), 114
 - [window.h](#), 115
- window_focus
 - [window.c](#), 114
 - [window.h](#), 116
- window_get_main_container
 - [window.c](#), 114
 - [window.h](#), 116
- window_hide
 - [window.c](#), 114
 - [window.h](#), 116
- window_show
 - [window.c](#), 114
 - [window.h](#), 116
- windows
 - [AppData](#), 6
- x
 - [UIConstraints](#), 15
 - [Vector2](#), 24
- y
 - [UIConstraints](#), 15
 - [Vector2](#), 24
- YELLOW
 - [color.h](#), 32
- zoom
 - [CoordinateSystem](#), 7