

Vally

Generated by Doxygen 1.9.4



<b>1 Module Index</b>	<b>1</b>
1.1 Modules	1
<b>2 Data Structure Index</b>	<b>3</b>
2.1 Data Structures	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Module Documentation</b>	<b>7</b>
4.1 Keyboard keys	7
4.1.1 Detailed Description	9
4.1.2 Macro Definition Documentation	9
4.1.2.1 KEY_0	9
4.1.2.2 KEY_1	9
4.1.2.3 KEY_2	10
4.1.2.4 KEY_3	10
4.1.2.5 KEY_4	10
4.1.2.6 KEY_5	10
4.1.2.7 KEY_6	10
4.1.2.8 KEY_7	10
4.1.2.9 KEY_8	10
4.1.2.10 KEY_9	10
4.1.2.11 KEY_A	11
4.1.2.12 KEY_APOSTROPHE	11
4.1.2.13 KEY_B	11
4.1.2.14 KEY_BACKSLASH	11
4.1.2.15 KEY_BACKSPACE	11
4.1.2.16 KEY_C	11
4.1.2.17 KEY_CAPS_LOCK	11
4.1.2.18 KEY_COMMA	11
4.1.2.19 KEY_D	12
4.1.2.20 KEY_DELETE	12
4.1.2.21 KEY_DOWN	12
4.1.2.22 KEY_E	12
4.1.2.23 KEY_END	12
4.1.2.24 KEY_ENTER	12
4.1.2.25 KEY_EQUAL	12
4.1.2.26 KEY_ESCAPE	12
4.1.2.27 KEY_F	13
4.1.2.28 KEY_F1	13
4.1.2.29 KEY_F10	13
4.1.2.30 KEY_F11	13

4.1.2.31 KEY_F12	13
4.1.2.32 KEY_F13	13
4.1.2.33 KEY_F14	13
4.1.2.34 KEY_F15	13
4.1.2.35 KEY_F16	14
4.1.2.36 KEY_F17	14
4.1.2.37 KEY_F18	14
4.1.2.38 KEY_F19	14
4.1.2.39 KEY_F2	14
4.1.2.40 KEY_F20	14
4.1.2.41 KEY_F21	14
4.1.2.42 KEY_F22	14
4.1.2.43 KEY_F23	15
4.1.2.44 KEY_F24	15
4.1.2.45 KEY_F25	15
4.1.2.46 KEY_F3	15
4.1.2.47 KEY_F4	15
4.1.2.48 KEY_F5	15
4.1.2.49 KEY_F6	15
4.1.2.50 KEY_F7	15
4.1.2.51 KEY_F8	16
4.1.2.52 KEY_F9	16
4.1.2.53 KEY_G	16
4.1.2.54 KEY_GRAVE_ACCENT	16
4.1.2.55 KEY_H	16
4.1.2.56 KEY_HOME	16
4.1.2.57 KEY_I	16
4.1.2.58 KEY_INSERT	16
4.1.2.59 KEY_J	17
4.1.2.60 KEY_K	17
4.1.2.61 KEY_KP_0	17
4.1.2.62 KEY_KP_1	17
4.1.2.63 KEY_KP_2	17
4.1.2.64 KEY_KP_3	17
4.1.2.65 KEY_KP_4	17
4.1.2.66 KEY_KP_5	17
4.1.2.67 KEY_KP_6	18
4.1.2.68 KEY_KP_7	18
4.1.2.69 KEY_KP_8	18
4.1.2.70 KEY_KP_9	18
4.1.2.71 KEY_KP_ADD	18
4.1.2.72 KEY_KP_DECIMAL	18

4.1.2.73 KEY_KP_DIVIDE . . . . .	18
4.1.2.74 KEY_KP_ENTER . . . . .	18
4.1.2.75 KEY_KP_EQUAL . . . . .	19
4.1.2.76 KEY_KP_MULTIPLY . . . . .	19
4.1.2.77 KEY_KP_SUBTRACT . . . . .	19
4.1.2.78 KEY_L . . . . .	19
4.1.2.79 KEY_LEFT . . . . .	19
4.1.2.80 KEY_LEFT_ALT . . . . .	19
4.1.2.81 KEY_LEFT_BRACKET . . . . .	19
4.1.2.82 KEY_LEFT_CONTROL . . . . .	19
4.1.2.83 KEY_LEFT_SHIFT . . . . .	20
4.1.2.84 KEY_LEFT_SUPER . . . . .	20
4.1.2.85 KEY_M . . . . .	20
4.1.2.86 KEY_MENU . . . . .	20
4.1.2.87 KEY_MINUS . . . . .	20
4.1.2.88 KEY_N . . . . .	20
4.1.2.89 KEY_NUM_LOCK . . . . .	20
4.1.2.90 KEY_O . . . . .	20
4.1.2.91 KEY_P . . . . .	21
4.1.2.92 KEY_PAGE_DOWN . . . . .	21
4.1.2.93 KEY_PAGE_UP . . . . .	21
4.1.2.94 KEY_PAUSE . . . . .	21
4.1.2.95 KEY_PERIOD . . . . .	21
4.1.2.96 KEY_PRINT_SCREEN . . . . .	21
4.1.2.97 KEY_Q . . . . .	21
4.1.2.98 KEY_R . . . . .	21
4.1.2.99 KEY_RIGHT . . . . .	22
4.1.2.100 KEY_RIGHT_ALT . . . . .	22
4.1.2.101 KEY_RIGHT_BRACKET . . . . .	22
4.1.2.102 KEY_RIGHT_CONTROL . . . . .	22
4.1.2.103 KEY_RIGHT_SHIFT . . . . .	22
4.1.2.104 KEY_RIGHT_SUPER . . . . .	22
4.1.2.105 KEY_S . . . . .	22
4.1.2.106 KEY_SCROLL_LOCK . . . . .	22
4.1.2.107 KEY_SEMICOLON . . . . .	23
4.1.2.108 KEY_SLASH . . . . .	23
4.1.2.109 KEY_SPACE . . . . .	23
4.1.2.110 KEY_T . . . . .	23
4.1.2.111 KEY_TAB . . . . .	23
4.1.2.112 KEY_U . . . . .	23
4.1.2.113 KEY_UP . . . . .	23
4.1.2.114 KEY_V . . . . .	23

4.1.2.115 KEY_W . . . . .	24
4.1.2.116 KEY_WORLD_1 . . . . .	24
4.1.2.117 KEY_WORLD_2 . . . . .	24
4.1.2.118 KEY_X . . . . .	24
4.1.2.119 KEY_Y . . . . .	24
4.1.2.120 KEY_Z . . . . .	24
4.1.2.121 KEYS_MAX_KEYS . . . . .	24
4.2 Mouse buttons . . . . .	25
4.2.1 Detailed Description . . . . .	25
4.2.2 Macro Definition Documentation . . . . .	25
4.2.2.1 MOUSE_BUTTON_1 . . . . .	25
4.2.2.2 MOUSE_BUTTON_2 . . . . .	25
4.2.2.3 MOUSE_BUTTON_3 . . . . .	25
4.2.2.4 MOUSE_BUTTON_4 . . . . .	25
4.2.2.5 MOUSE_BUTTON_5 . . . . .	26
4.2.2.6 MOUSE_BUTTON_6 . . . . .	26
4.2.2.7 MOUSE_BUTTON_7 . . . . .	26
4.2.2.8 MOUSE_BUTTON_8 . . . . .	26
4.2.2.9 MOUSE_BUTTON_LEFT . . . . .	26
4.2.2.10 MOUSE_BUTTON_MAX . . . . .	26
4.2.2.11 MOUSE_BUTTON_MIDDLE . . . . .	26
4.2.2.12 MOUSE_BUTTON_RIGHT . . . . .	26
<b>5 Data Structure Documentation . . . . .</b>	<b>27</b>
5.1 animation Struct Reference . . . . .	27
5.1.1 Field Documentation . . . . .	27
5.1.1.1 first_frame . . . . .	27
5.1.1.2 frametime . . . . .	27
5.1.1.3 last_frame . . . . .	27
5.1.1.4 texture . . . . .	28
5.1.1.5 tiling . . . . .	28
5.2 animator Struct Reference . . . . .	28
5.2.1 Field Documentation . . . . .	28
5.2.1.1 animation . . . . .	28
5.2.1.2 current_frame . . . . .	28
5.2.1.3 playing . . . . .	29
5.2.1.4 time . . . . .	29
5.3 atlas Struct Reference . . . . .	29
5.3.1 Field Documentation . . . . .	29
5.3.1.1 sprite_count . . . . .	29
5.3.1.2 sprites_per_row . . . . .	29
5.3.1.3 tiling . . . . .	30

5.3.1.4 uvs	30
5.4 event_context Struct Reference	30
5.4.1 Detailed Description	30
5.4.2 Field Documentation	30
5.4.2.1 c	31
5.4.2.2	31
5.4.2.3 f32	31
5.4.2.4 f64	31
5.4.2.5 i16	31
5.4.2.6 i32	31
5.4.2.7 i64	31
5.4.2.8 i8	32
5.4.2.9 u16	32
5.4.2.10 u32	32
5.4.2.11 u64	32
5.4.2.12 u8	32
5.5 image Struct Reference	32
5.5.1 Field Documentation	33
5.5.1.1 buffer	33
5.5.1.2 channels	33
5.5.1.3 height	33
5.5.1.4 width	33
5.6 resource Struct Reference	33
5.6.1 Field Documentation	33
5.6.1.1 data	34
5.6.1.2 destructor	34
5.7 spriterenderer Struct Reference	34
5.7.1 Field Documentation	34
5.7.1.1 atlas	34
5.7.1.2 entity	34
5.7.1.3 size	35
5.7.1.4 texture	35
5.7.1.5 uvs	35
5.8 texture Struct Reference	35
5.8.1 Detailed Description	35
5.8.2 Field Documentation	35
5.8.2.1 height	36
5.8.2.2 id	36
5.8.2.3 width	36
5.9 transform Struct Reference	36
5.9.1 Field Documentation	36
5.9.1.1 entity	36

5.9.1.2 position	36
5.9.1.3 scale	37
5.10 uvs Struct Reference	37
5.10.1 Field Documentation	37
5.10.1.1 uv	37
5.11 vector2f Struct Reference	37
5.11.1 Field Documentation	37
5.11.1.1 x	38
5.11.1.2 y	38
5.12 vector2i Struct Reference	38
5.12.1 Field Documentation	38
5.12.1.1 x	38
5.12.1.2 y	38
5.13 vector3f Struct Reference	38
5.13.1 Field Documentation	39
5.13.1.1 x	39
5.13.1.2 y	39
5.13.1.3 z	39
5.14 vector3i Struct Reference	39
5.14.1 Field Documentation	39
5.14.1.1 x	40
5.14.1.2 y	40
5.14.1.3 z	40
<b>6 File Documentation</b>	<b>41</b>
6.1 include/vally/config.h File Reference	41
6.1.1 Detailed Description	42
6.1.2 Macro Definition Documentation	42
6.1.2.1 BIT_MASK	42
6.1.2.2 FALSE	42
6.1.2.3 MAX	42
6.1.2.4 MIN	42
6.1.2.5 NULL	43
6.1.2.6 TRUE	43
6.1.2.7 VALLY_API	43
6.1.3 Typedef Documentation	43
6.1.3.1 b32	43
6.1.3.2 b8	43
6.1.3.3 f32	43
6.1.3.4 f64	44
6.1.3.5 i16	44
6.1.3.6 i32	44



6.1.3.7 i64 . . . . .	44
6.1.3.8 i8 . . . . .	44
6.1.3.9 u16 . . . . .	44
6.1.3.10 u32 . . . . .	44
6.1.3.11 u64 . . . . .	45
6.1.3.12 u8 . . . . .	45
6.2 config.h . . . . .	45
6.3 include/vally/core/engine.h File Reference . . . . .	46
6.3.1 Typedef Documentation . . . . .	47
6.3.1.1 engine_render . . . . .	47
6.3.1.2 engine_start . . . . .	47
6.3.1.3 engine_update . . . . .	47
6.3.2 Function Documentation . . . . .	47
6.3.2.1 engine_create() . . . . .	47
6.3.2.2 engine_run() . . . . .	48
6.4 engine.h . . . . .	48
6.5 include/vally/core/event.h File Reference . . . . .	49
6.5.1 Typedef Documentation . . . . .	49
6.5.1.1 on_event_fun . . . . .	49
6.5.2 Enumeration Type Documentation . . . . .	50
6.5.2.1 engine_event_code . . . . .	50
6.5.3 Function Documentation . . . . .	50
6.5.3.1 event_init() . . . . .	50
6.5.3.2 event_publish() . . . . .	50
6.5.3.3 event_subscribe() . . . . .	51
6.5.3.4 event_terminate() . . . . .	51
6.5.3.5 event_unsubscribe() . . . . .	51
6.6 event.h . . . . .	52
6.7 include/vally/core/input.h File Reference . . . . .	52
6.7.1 Function Documentation . . . . .	55
6.7.1.1 input_init() . . . . .	55
6.7.1.2 input_key_pressed() . . . . .	56
6.7.1.3 input_key_released() . . . . .	56
6.7.1.4 input_mouse_button_pressed() . . . . .	56
6.7.1.5 input_mouse_button_released() . . . . .	57
6.7.1.6 input_mouse_position() . . . . .	57
6.7.1.7 input_mouse_prev_position() . . . . .	57
6.7.1.8 input_terminate() . . . . .	58
6.8 input.h . . . . .	58
6.9 include/vally/ecs/components/animatorm.h File Reference . . . . .	60
6.9.1 Function Documentation . . . . .	60
6.9.1.1 animation_create() . . . . .	60

6.9.1.2	<a href="#">ecs_animator_add()</a>	60
6.9.1.3	<a href="#">ecs_animator_play()</a>	61
6.9.1.4	<a href="#">ecs_animator_stop()</a>	61
6.9.1.5	<a href="#">ecs_animator_update()</a>	61
6.10	<a href="#">animator.h</a>	61
6.11	<a href="#">include/vally/ecs/components/spriterenderer.h</a> File Reference	62
6.11.1	Function Documentation	62
6.11.1.1	<a href="#">ecs_spriterenderer_add()</a>	62
6.11.1.2	<a href="#">ecs_spriterenderer_add_from_atlas()</a>	62
6.11.1.3	<a href="#">ecs_spriterenderer_atlas_select()</a>	62
6.11.1.4	<a href="#">ecs_spriterenderer_get()</a>	63
6.11.1.5	<a href="#">ecs_spriterenderer_update()</a>	63
6.12	<a href="#">spriterenderer.h</a>	63
6.13	<a href="#">include/vally/ecs/components/transform.h</a> File Reference	63
6.13.1	Function Documentation	64
6.13.1.1	<a href="#">ecs_transform_add()</a>	64
6.13.1.2	<a href="#">ecs_transform_get()</a>	64
6.14	<a href="#">transform.h</a>	64
6.15	<a href="#">include/vally/ecs/ecs.h</a> File Reference	64
6.15.1	Macro Definition Documentation	65
6.15.1.1	<a href="#">ecs_is_alive</a>	65
6.15.1.2	<a href="#">ECS_MAX_COMPONENTS</a>	65
6.15.1.3	<a href="#">ECS_MAX_ENTITIES</a>	65
6.15.1.4	<a href="#">ECS_NULL_ENTITY</a>	65
6.15.2	Function Documentation	65
6.15.2.1	<a href="#">ecs_component_add()</a>	66
6.15.2.2	<a href="#">ecs_component_has()</a>	66
6.15.2.3	<a href="#">ecs_component_remove()</a>	66
6.15.2.4	<a href="#">ecs_entity_create()</a>	66
6.15.2.5	<a href="#">ecs_get_last()</a>	66
6.15.2.6	<a href="#">ecs_init()</a>	66
6.16	<a href="#">ecs.h</a>	67
6.17	<a href="#">include/vally/ecs/ecs_types.h</a> File Reference	67
6.17.1	Typedef Documentation	67
6.17.1.1	<a href="#">entity</a>	67
6.17.2	Enumeration Type Documentation	67
6.17.2.1	<a href="#">component_mask</a>	67
6.18	<a href="#">ecs_types.h</a>	68
6.19	<a href="#">include/vally/math/math.h</a> File Reference	68
6.19.1	Macro Definition Documentation	69
6.19.1.1	<a href="#">LERP</a>	69
6.19.1.2	<a href="#">V_2PI</a>	70

---

6.19.1.3 V_PI	70
6.19.1.4 V_PI_2	70
6.19.1.5 V_PI_3	70
6.19.1.6 V_PI_4	70
6.19.1.7 V_SQRT2	70
6.19.1.8 vector2f_create	70
6.19.1.9 vector2f_down	71
6.19.1.10 vector2f_left	71
6.19.1.11 vector2f_one	71
6.19.1.12 vector2f_right	71
6.19.1.13 vector2f_up	71
6.19.1.14 vector2f_zero	71
6.19.1.15 vector2i_create	71
6.19.1.16 vector3f_create	72
6.19.1.17 vector3i_create	72
6.19.2 Function Documentation	72
6.19.2.1 vabs()	72
6.19.2.2 vacos()	72
6.19.2.3 vasin()	72
6.19.2.4 vatan()	72
6.19.2.5 vcos()	73
6.19.2.6 vector2f_add()	73
6.19.2.7 vector2f_distance()	73
6.19.2.8 vector2f_div()	73
6.19.2.9 vector2f_divs()	73
6.19.2.10 vector2f_length()	73
6.19.2.11 vector2f_mul()	74
6.19.2.12 vector2f_muls()	74
6.19.2.13 vector2f_normalize()	74
6.19.2.14 vector2f_normalized()	74
6.19.2.15 vector2f_sub()	74
6.19.2.16 vector2i_add()	74
6.19.2.17 vector2i_sub()	75
6.19.2.18 vector3f_add()	75
6.19.2.19 vector3f_cross()	75
6.19.2.20 vector3f_distance()	75
6.19.2.21 vector3f_div()	75
6.19.2.22 vector3f_divs()	75
6.19.2.23 vector3f_dot()	76
6.19.2.24 vector3f_length()	76
6.19.2.25 vector3f_mul()	76
6.19.2.26 vector3f_muls()	76

6.19.2.27 vector3f_normalize()	76
6.19.2.28 vector3f_normalized()	76
6.19.2.29 vector3f_sub()	77
6.19.2.30 vector3i_add()	77
6.19.2.31 vector3i_sub()	77
6.19.2.32 vsin()	77
6.19.2.33 vsqrt()	77
6.19.2.34 vtan()	77
6.20 math.h	78
6.21 include/vally/math/math_types.h File Reference	79
6.22 math_types.h	80
6.23 include/vally/renderer/texture.h File Reference	80
6.23.1 Function Documentation	81
6.23.1.1 texture_bind()	81
6.23.1.2 texture_create()	81
6.23.1.3 texture_destroy()	81
6.23.1.4 texture_unbind()	81
6.23.1.5 texture_white_create()	81
6.24 texture.h	82
6.25 include/vally/resources/image.h File Reference	82
6.25.1 Function Documentation	82
6.25.1.1 image_load()	83
6.25.1.2 image_unload()	83
6.26 image.h	83
6.27 include/vally/resources/resource_types.h File Reference	83
6.27.1 Typedef Documentation	83
6.27.1.1 destructor_fun	83
6.28 resource_types.h	84
6.29 include/vally/vally.h File Reference	84
6.30 vally.h	84
<b>Index</b>	<b>85</b>

# Chapter 1

## Module Index

### 1.1 Modules

Here is a list of all modules:

Keyboard keys . . . . .	7
Mouse buttons . . . . .	25



## Chapter 2

# Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">animation</a>	27
<a href="#">animator</a>	28
<a href="#">atlas</a>	29
<a href="#">event_context</a>	
Context send alongside the event	30
<a href="#">image</a>	32
<a href="#">resource</a>	33
<a href="#">spriterenderer</a>	34
<a href="#">texture</a>	
Texture struct	35
<a href="#">transform</a>	36
<a href="#">uvs</a>	37
<a href="#">vector2f</a>	37
<a href="#">vector2i</a>	38
<a href="#">vector3f</a>	38
<a href="#">vector3i</a>	39





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

include/vally/ <a href="#">config.h</a>	
This header file contains main declarations and macros	41
include/vally/ <a href="#">vally.h</a>	84
include/vally/core/ <a href="#">engine.h</a>	46
include/vally/core/ <a href="#">event.h</a>	49
include/vally/core/ <a href="#">input.h</a>	52
include/vally/ecs/ <a href="#">ecs.h</a>	64
include/vally/ecs/ <a href="#">ecs_types.h</a>	67
include/vally/ecs/components/ <a href="#">animator.h</a>	60
include/vally/ecs/components/ <a href="#">spriterenderer.h</a>	62
include/vally/ecs/components/ <a href="#">transform.h</a>	63
include/vally/math/ <a href="#">math.h</a>	68
include/vally/math/ <a href="#">math_types.h</a>	79
include/vally/renderer/ <a href="#">texture.h</a>	80
include/vally/resources/ <a href="#">image.h</a>	82
include/vally/resources/ <a href="#">resource_types.h</a>	83



## Chapter 4

# Module Documentation

### 4.1 Keyboard keys

#### Macros

- `#define KEY_SPACE` 32
- `#define KEY_APOSTROPHE` 39
- `#define KEY_COMMA` 44
- `#define KEY_MINUS` 45
- `#define KEY_PERIOD` 46
- `#define KEY_SLASH` 47
- `#define KEY_0` 48
- `#define KEY_1` 49
- `#define KEY_2` 50
- `#define KEY_3` 51
- `#define KEY_4` 52
- `#define KEY_5` 53
- `#define KEY_6` 54
- `#define KEY_7` 55
- `#define KEY_8` 56
- `#define KEY_9` 57
- `#define KEY_SEMICOLON` 59
- `#define KEY_EQUAL` 61
- `#define KEY_A` 65
- `#define KEY_B` 66
- `#define KEY_C` 67
- `#define KEY_D` 68
- `#define KEY_E` 69
- `#define KEY_F` 70
- `#define KEY_G` 71
- `#define KEY_H` 72
- `#define KEY_I` 73
- `#define KEY_J` 74
- `#define KEY_K` 75
- `#define KEY_L` 76
- `#define KEY_M` 77
- `#define KEY_N` 78
- `#define KEY_O` 79

- #define [KEY\\_P](#) 80
- #define [KEY\\_Q](#) 81
- #define [KEY\\_R](#) 82
- #define [KEY\\_S](#) 83
- #define [KEY\\_T](#) 84
- #define [KEY\\_U](#) 85
- #define [KEY\\_V](#) 86
- #define [KEY\\_W](#) 87
- #define [KEY\\_X](#) 88
- #define [KEY\\_Y](#) 89
- #define [KEY\\_Z](#) 90
- #define [KEY\\_LEFT\\_BRACKET](#) 91
- #define [KEY\\_BACKSLASH](#) 92
- #define [KEY\\_RIGHT\\_BRACKET](#) 93
- #define [KEY\\_GRAVE\\_ACCENT](#) 96
- #define [KEY\\_WORLD\\_1](#) 161
- #define [KEY\\_WORLD\\_2](#) 162
- #define [KEY\\_ESCAPE](#) 256
- #define [KEY\\_ENTER](#) 257
- #define [KEY\\_TAB](#) 258
- #define [KEY\\_BACKSPACE](#) 259
- #define [KEY\\_INSERT](#) 260
- #define [KEY\\_DELETE](#) 261
- #define [KEY\\_RIGHT](#) 262
- #define [KEY\\_LEFT](#) 263
- #define [KEY\\_DOWN](#) 264
- #define [KEY\\_UP](#) 265
- #define [KEY\\_PAGE\\_UP](#) 266
- #define [KEY\\_PAGE\\_DOWN](#) 267
- #define [KEY\\_HOME](#) 268
- #define [KEY\\_END](#) 269
- #define [KEY\\_CAPS\\_LOCK](#) 280
- #define [KEY\\_SCROLL\\_LOCK](#) 281
- #define [KEY\\_NUM\\_LOCK](#) 282
- #define [KEY\\_PRINT\\_SCREEN](#) 283
- #define [KEY\\_PAUSE](#) 284
- #define [KEY\\_F1](#) 290
- #define [KEY\\_F2](#) 291
- #define [KEY\\_F3](#) 292
- #define [KEY\\_F4](#) 293
- #define [KEY\\_F5](#) 294
- #define [KEY\\_F6](#) 295
- #define [KEY\\_F7](#) 296
- #define [KEY\\_F8](#) 297
- #define [KEY\\_F9](#) 298
- #define [KEY\\_F10](#) 299
- #define [KEY\\_F11](#) 300
- #define [KEY\\_F12](#) 301
- #define [KEY\\_F13](#) 302
- #define [KEY\\_F14](#) 303
- #define [KEY\\_F15](#) 304
- #define [KEY\\_F16](#) 305
- #define [KEY\\_F17](#) 306
- #define [KEY\\_F18](#) 307
- #define [KEY\\_F19](#) 308

- `#define KEY_F20 309`
- `#define KEY_F21 310`
- `#define KEY_F22 311`
- `#define KEY_F23 312`
- `#define KEY_F24 313`
- `#define KEY_F25 314`
- `#define KEY_KP_0 320`
- `#define KEY_KP_1 321`
- `#define KEY_KP_2 322`
- `#define KEY_KP_3 323`
- `#define KEY_KP_4 324`
- `#define KEY_KP_5 325`
- `#define KEY_KP_6 326`
- `#define KEY_KP_7 327`
- `#define KEY_KP_8 328`
- `#define KEY_KP_9 329`
- `#define KEY_KP_DECIMAL 330`
- `#define KEY_KP_DIVIDE 331`
- `#define KEY_KP_MULTIPLY 332`
- `#define KEY_KP_SUBTRACT 333`
- `#define KEY_KP_ADD 334`
- `#define KEY_KP_ENTER 335`
- `#define KEY_KP_EQUAL 336`
- `#define KEY_LEFT_SHIFT 340`
- `#define KEY_LEFT_CONTROL 341`
- `#define KEY_LEFT_ALT 342`
- `#define KEY_LEFT_SUPER 343`
- `#define KEY_RIGHT_SHIFT 344`
- `#define KEY_RIGHT_CONTROL 345`
- `#define KEY_RIGHT_ALT 346`
- `#define KEY_RIGHT_SUPER 347`
- `#define KEY_MENU 348`
- `#define KEYS_MAX_KEYS KEY_MENU`

### 4.1.1 Detailed Description

### 4.1.2 Macro Definition Documentation

#### 4.1.2.1 KEY\_0

```
#define KEY_0 48
```

#### 4.1.2.2 KEY\_1

```
#define KEY_1 49
```

#### 4.1.2.3 KEY\_2

```
#define KEY_2 50
```

#### 4.1.2.4 KEY\_3

```
#define KEY_3 51
```

#### 4.1.2.5 KEY\_4

```
#define KEY_4 52
```

#### 4.1.2.6 KEY\_5

```
#define KEY_5 53
```

#### 4.1.2.7 KEY\_6

```
#define KEY_6 54
```

#### 4.1.2.8 KEY\_7

```
#define KEY_7 55
```

#### 4.1.2.9 KEY\_8

```
#define KEY_8 56
```

#### 4.1.2.10 KEY\_9

```
#define KEY_9 57
```

#### 4.1.2.11 KEY\_A

```
#define KEY_A 65
```

#### 4.1.2.12 KEY\_APOSTROPHE

```
#define KEY_APOSTROPHE 39
```

#### 4.1.2.13 KEY\_B

```
#define KEY_B 66
```

#### 4.1.2.14 KEY\_BACKSLASH

```
#define KEY_BACKSLASH 92
```

#### 4.1.2.15 KEY\_BACKSPACE

```
#define KEY_BACKSPACE 259
```

#### 4.1.2.16 KEY\_C

```
#define KEY_C 67
```

#### 4.1.2.17 KEY\_CAPS\_LOCK

```
#define KEY_CAPS_LOCK 280
```

#### 4.1.2.18 KEY\_COMMA

```
#define KEY_COMMA 44
```

**4.1.2.19 KEY\_D**

```
#define KEY_D 68
```

**4.1.2.20 KEY\_DELETE**

```
#define KEY_DELETE 261
```

**4.1.2.21 KEY\_DOWN**

```
#define KEY_DOWN 264
```

**4.1.2.22 KEY\_E**

```
#define KEY_E 69
```

**4.1.2.23 KEY\_END**

```
#define KEY_END 269
```

**4.1.2.24 KEY\_ENTER**

```
#define KEY_ENTER 257
```

**4.1.2.25 KEY\_EQUAL**

```
#define KEY_EQUAL 61
```

**4.1.2.26 KEY\_ESCAPE**

```
#define KEY_ESCAPE 256
```



**4.1.2.27 KEY\_F**

```
#define KEY_F 70
```

**4.1.2.28 KEY\_F1**

```
#define KEY_F1 290
```

**4.1.2.29 KEY\_F10**

```
#define KEY_F10 299
```

**4.1.2.30 KEY\_F11**

```
#define KEY_F11 300
```

**4.1.2.31 KEY\_F12**

```
#define KEY_F12 301
```

**4.1.2.32 KEY\_F13**

```
#define KEY_F13 302
```

**4.1.2.33 KEY\_F14**

```
#define KEY_F14 303
```

**4.1.2.34 KEY\_F15**

```
#define KEY_F15 304
```

**4.1.2.35 KEY\_F16**

```
#define KEY_F16 305
```

**4.1.2.36 KEY\_F17**

```
#define KEY_F17 306
```

**4.1.2.37 KEY\_F18**

```
#define KEY_F18 307
```

**4.1.2.38 KEY\_F19**

```
#define KEY_F19 308
```

**4.1.2.39 KEY\_F2**

```
#define KEY_F2 291
```

**4.1.2.40 KEY\_F20**

```
#define KEY_F20 309
```

**4.1.2.41 KEY\_F21**

```
#define KEY_F21 310
```

**4.1.2.42 KEY\_F22**

```
#define KEY_F22 311
```

**4.1.2.43 KEY\_F23**

```
#define KEY_F23 312
```

**4.1.2.44 KEY\_F24**

```
#define KEY_F24 313
```

**4.1.2.45 KEY\_F25**

```
#define KEY_F25 314
```

**4.1.2.46 KEY\_F3**

```
#define KEY_F3 292
```

**4.1.2.47 KEY\_F4**

```
#define KEY_F4 293
```

**4.1.2.48 KEY\_F5**

```
#define KEY_F5 294
```

**4.1.2.49 KEY\_F6**

```
#define KEY_F6 295
```

**4.1.2.50 KEY\_F7**

```
#define KEY_F7 296
```

**4.1.2.51 KEY\_F8**

```
#define KEY_F8 297
```

**4.1.2.52 KEY\_F9**

```
#define KEY_F9 298
```

**4.1.2.53 KEY\_G**

```
#define KEY_G 71
```

**4.1.2.54 KEY\_GRAVE\_ACCENT**

```
#define KEY_GRAVE_ACCENT 96
```

**4.1.2.55 KEY\_H**

```
#define KEY_H 72
```

**4.1.2.56 KEY\_HOME**

```
#define KEY_HOME 268
```

**4.1.2.57 KEY\_I**

```
#define KEY_I 73
```

**4.1.2.58 KEY\_INSERT**

```
#define KEY_INSERT 260
```

**4.1.2.59 KEY\_J**

```
#define KEY_J 74
```

**4.1.2.60 KEY\_K**

```
#define KEY_K 75
```

**4.1.2.61 KEY\_KP\_0**

```
#define KEY_KP_0 320
```

**4.1.2.62 KEY\_KP\_1**

```
#define KEY_KP_1 321
```

**4.1.2.63 KEY\_KP\_2**

```
#define KEY_KP_2 322
```

**4.1.2.64 KEY\_KP\_3**

```
#define KEY_KP_3 323
```

**4.1.2.65 KEY\_KP\_4**

```
#define KEY_KP_4 324
```

**4.1.2.66 KEY\_KP\_5**

```
#define KEY_KP_5 325
```

**4.1.2.67 KEY\_KP\_6**

```
#define KEY_KP_6 326
```

**4.1.2.68 KEY\_KP\_7**

```
#define KEY_KP_7 327
```

**4.1.2.69 KEY\_KP\_8**

```
#define KEY_KP_8 328
```

**4.1.2.70 KEY\_KP\_9**

```
#define KEY_KP_9 329
```

**4.1.2.71 KEY\_KP\_ADD**

```
#define KEY_KP_ADD 334
```

**4.1.2.72 KEY\_KP\_DECIMAL**

```
#define KEY_KP_DECIMAL 330
```

**4.1.2.73 KEY\_KP\_DIVIDE**

```
#define KEY_KP_DIVIDE 331
```

**4.1.2.74 KEY\_KP\_ENTER**

```
#define KEY_KP_ENTER 335
```

**4.1.2.75 KEY\_KP\_EQUAL**

```
#define KEY_KP_EQUAL 336
```

**4.1.2.76 KEY\_KP\_MULTIPLY**

```
#define KEY_KP_MULTIPLY 332
```

**4.1.2.77 KEY\_KP\_SUBTRACT**

```
#define KEY_KP_SUBTRACT 333
```

**4.1.2.78 KEY\_L**

```
#define KEY_L 76
```

**4.1.2.79 KEY\_LEFT**

```
#define KEY_LEFT 263
```

**4.1.2.80 KEY\_LEFT\_ALT**

```
#define KEY_LEFT_ALT 342
```

**4.1.2.81 KEY\_LEFT\_BRACKET**

```
#define KEY_LEFT_BRACKET 91
```

**4.1.2.82 KEY\_LEFT\_CONTROL**

```
#define KEY_LEFT_CONTROL 341
```

#### 4.1.2.83 KEY\_LEFT\_SHIFT

```
#define KEY_LEFT_SHIFT 340
```

#### 4.1.2.84 KEY\_LEFT\_SUPER

```
#define KEY_LEFT_SUPER 343
```

#### 4.1.2.85 KEY\_M

```
#define KEY_M 77
```

#### 4.1.2.86 KEY\_MENU

```
#define KEY_MENU 348
```

#### 4.1.2.87 KEY\_MINUS

```
#define KEY_MINUS 45
```

#### 4.1.2.88 KEY\_N

```
#define KEY_N 78
```

#### 4.1.2.89 KEY\_NUM\_LOCK

```
#define KEY_NUM_LOCK 282
```

#### 4.1.2.90 KEY\_O

```
#define KEY_O 79
```



**4.1.2.91 KEY\_P**

```
#define KEY_P 80
```

**4.1.2.92 KEY\_PAGE\_DOWN**

```
#define KEY_PAGE_DOWN 267
```

**4.1.2.93 KEY\_PAGE\_UP**

```
#define KEY_PAGE_UP 266
```

**4.1.2.94 KEY\_PAUSE**

```
#define KEY_PAUSE 284
```

**4.1.2.95 KEY\_PERIOD**

```
#define KEY_PERIOD 46
```

**4.1.2.96 KEY\_PRINT\_SCREEN**

```
#define KEY_PRINT_SCREEN 283
```

**4.1.2.97 KEY\_Q**

```
#define KEY_Q 81
```

**4.1.2.98 KEY\_R**

```
#define KEY_R 82
```

**4.1.2.99 KEY\_RIGHT**

```
#define KEY_RIGHT 262
```

**4.1.2.100 KEY\_RIGHT\_ALT**

```
#define KEY_RIGHT_ALT 346
```

**4.1.2.101 KEY\_RIGHT\_BRACKET**

```
#define KEY_RIGHT_BRACKET 93
```

**4.1.2.102 KEY\_RIGHT\_CONTROL**

```
#define KEY_RIGHT_CONTROL 345
```

**4.1.2.103 KEY\_RIGHT\_SHIFT**

```
#define KEY_RIGHT_SHIFT 344
```

**4.1.2.104 KEY\_RIGHT\_SUPER**

```
#define KEY_RIGHT_SUPER 347
```

**4.1.2.105 KEY\_S**

```
#define KEY_S 83
```

**4.1.2.106 KEY\_SCROLL\_LOCK**

```
#define KEY_SCROLL_LOCK 281
```

**4.1.2.107 KEY\_SEMICOLON**

```
#define KEY_SEMICOLON 59
```

**4.1.2.108 KEY\_SLASH**

```
#define KEY_SLASH 47
```

**4.1.2.109 KEY\_SPACE**

```
#define KEY_SPACE 32
```

**4.1.2.110 KEY\_T**

```
#define KEY_T 84
```

**4.1.2.111 KEY\_TAB**

```
#define KEY_TAB 258
```

**4.1.2.112 KEY\_U**

```
#define KEY_U 85
```

**4.1.2.113 KEY\_UP**

```
#define KEY_UP 265
```

**4.1.2.114 KEY\_V**

```
#define KEY_V 86
```

**4.1.2.115 KEY\_W**

```
#define KEY_W 87
```

**4.1.2.116 KEY\_WORLD\_1**

```
#define KEY_WORLD_1 161
```

**4.1.2.117 KEY\_WORLD\_2**

```
#define KEY_WORLD_2 162
```

**4.1.2.118 KEY\_X**

```
#define KEY_X 88
```

**4.1.2.119 KEY\_Y**

```
#define KEY_Y 89
```

**4.1.2.120 KEY\_Z**

```
#define KEY_Z 90
```

**4.1.2.121 KEYS\_MAX\_KEYS**

```
#define KEYS_MAX_KEYS KEY\_MENU
```

## 4.2 Mouse buttons

### Macros

- `#define` [MOUSE\\_BUTTON\\_1](#) 0
- `#define` [MOUSE\\_BUTTON\\_2](#) 1
- `#define` [MOUSE\\_BUTTON\\_3](#) 2
- `#define` [MOUSE\\_BUTTON\\_4](#) 3
- `#define` [MOUSE\\_BUTTON\\_5](#) 4
- `#define` [MOUSE\\_BUTTON\\_6](#) 5
- `#define` [MOUSE\\_BUTTON\\_7](#) 6
- `#define` [MOUSE\\_BUTTON\\_8](#) 7
- `#define` [MOUSE\\_BUTTON\\_MAX](#) [MOUSE\\_BUTTON\\_8](#)
- `#define` [MOUSE\\_BUTTON\\_LEFT](#) [MOUSE\\_BUTTON\\_1](#)
- `#define` [MOUSE\\_BUTTON\\_RIGHT](#) [MOUSE\\_BUTTON\\_2](#)
- `#define` [MOUSE\\_BUTTON\\_MIDDLE](#) [MOUSE\\_BUTTON\\_3](#)

### 4.2.1 Detailed Description

### 4.2.2 Macro Definition Documentation

#### 4.2.2.1 [MOUSE\\_BUTTON\\_1](#)

```
#define MOUSE_BUTTON_1 0
```

#### 4.2.2.2 [MOUSE\\_BUTTON\\_2](#)

```
#define MOUSE_BUTTON_2 1
```

#### 4.2.2.3 [MOUSE\\_BUTTON\\_3](#)

```
#define MOUSE_BUTTON_3 2
```

#### 4.2.2.4 [MOUSE\\_BUTTON\\_4](#)

```
#define MOUSE_BUTTON_4 3
```

#### 4.2.2.5 MOUSE\_BUTTON\_5

```
#define MOUSE_BUTTON_5 4
```

#### 4.2.2.6 MOUSE\_BUTTON\_6

```
#define MOUSE_BUTTON_6 5
```

#### 4.2.2.7 MOUSE\_BUTTON\_7

```
#define MOUSE_BUTTON_7 6
```

#### 4.2.2.8 MOUSE\_BUTTON\_8

```
#define MOUSE_BUTTON_8 7
```

#### 4.2.2.9 MOUSE\_BUTTON\_LEFT

```
#define MOUSE_BUTTON_LEFT MOUSE\_BUTTON\_1
```

#### 4.2.2.10 MOUSE\_BUTTON\_MAX

```
#define MOUSE_BUTTON_MAX MOUSE\_BUTTON\_8
```

#### 4.2.2.11 MOUSE\_BUTTON\_MIDDLE

```
#define MOUSE_BUTTON_MIDDLE MOUSE\_BUTTON\_3
```

#### 4.2.2.12 MOUSE\_BUTTON\_RIGHT

```
#define MOUSE_BUTTON_RIGHT MOUSE\_BUTTON\_2
```

## Chapter 5

# Data Structure Documentation

### 5.1 animation Struct Reference

```
#include <animator.h>
```

#### Data Fields

- [texture \\* texture](#)
- [vector2f tiling](#)
- [u32 first\\_frame](#)
- [u32 last\\_frame](#)
- [f32 frametime](#)

#### 5.1.1 Field Documentation

##### 5.1.1.1 first\_frame

```
u32 animation::first_frame
```

##### 5.1.1.2 frametime

```
f32 animation::frametime
```

##### 5.1.1.3 last\_frame

```
u32 animation::last_frame
```

#### 5.1.1.4 texture

```
texture* animation::texture
```

#### 5.1.1.5 tiling

```
vector2f animation::tiling
```

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

- include/vally/ecs/components/[animator.h](#)

## 5.2 animator Struct Reference

```
#include <animator.h>
```

### Data Fields

- [b8 playing](#)
- [animation](#) \* [animation](#)
- [u32 current\\_frame](#)
- [f32 time](#)

### 5.2.1 Field Documentation

#### 5.2.1.1 animation

```
animation* animator::animation
```

#### 5.2.1.2 current\_frame

```
u32 animator::current_frame
```



### 5.2.1.3 playing

```
b8 animator::playing
```

### 5.2.1.4 time

```
f32 animator::time
```

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

- include/vally/ecs/components/[animator.h](#)

## 5.3 atlas Struct Reference

```
#include <spriterenderer.h>
```

### Data Fields

- [vector2f tiling](#)
- [u32 sprite\\_count](#)
- [u32 sprites\\_per\\_row](#)
- [uvs \\* uvs](#)

### 5.3.1 Field Documentation

#### 5.3.1.1 sprite\_count

```
u32 atlas::sprite_count
```

#### 5.3.1.2 sprites\_per\_row

```
u32 atlas::sprites_per_row
```

### 5.3.1.3 tiling

```
vector2f atlas::tiling
```

### 5.3.1.4 uvs

```
uvs* atlas::uvs
```

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

- include/vally/ecs/components/[spriterenderer.h](#)

## 5.4 event\_context Struct Reference

Context send alongside the event.

```
#include <event.h>
```

### Data Fields

- union {
    - [i64 i64](#) [2]
    - [u64 u64](#) [2]
    - [f64 f64](#) [2]
    - [i32 i32](#) [4]
    - [u32 u32](#) [4]
    - [f32 f32](#) [4]
    - [i16 i16](#) [8]
    - [u16 u16](#) [8]
    - [i8 i8](#) [8]
    - [u8 u8](#) [8]
    - char [c](#) [16]
- } [data](#)

*Stores 128 bytes of data.*

### 5.4.1 Detailed Description

Context send alongside the event.

### 5.4.2 Field Documentation

#### 5.4.2.1 c

```
char event_context::c[16]
```

#### 5.4.2.2

```
union { ... } event_context::data
```

Stores 128 bytes of data.

#### 5.4.2.3 f32

```
f32 event_context::f32[4]
```

#### 5.4.2.4 f64

```
f64 event_context::f64[2]
```

#### 5.4.2.5 i16

```
i16 event_context::i16[8]
```

#### 5.4.2.6 i32

```
i32 event_context::i32[4]
```

#### 5.4.2.7 i64

```
i64 event_context::i64[2]
```

#### 5.4.2.8 i8

```
i8 event_context::i8[8]
```

#### 5.4.2.9 u16

```
u16 event_context::u16[8]
```

#### 5.4.2.10 u32

```
u32 event_context::u32[4]
```

#### 5.4.2.11 u64

```
u64 event_context::u64[2]
```

#### 5.4.2.12 u8

```
u8 event_context::u8[8]
```

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

- `include/vally/core/event.h`

## 5.5 image Struct Reference

```
#include <resource_types.h>
```

### Data Fields

- `i8 * buffer`
- `i32 width`
- `i32 height`
- `i32 channels`

## 5.5.1 Field Documentation

### 5.5.1.1 buffer

`i8* image::buffer`

### 5.5.1.2 channels

`i32 image::channels`

### 5.5.1.3 height

`i32 image::height`

### 5.5.1.4 width

`i32 image::width`

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

- `include/vally/resources/resource_types.h`

## 5.6 resource Struct Reference

```
#include <resource_types.h>
```

### Data Fields

- `void * data`
- `destructor_fun destructor`

## 5.6.1 Field Documentation

#### 5.6.1.1 data

```
void* resource::data
```

#### 5.6.1.2 destructor

```
destructor_fun resource::destructor
```

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

- [include/vally/resources/resource\\_types.h](#)

## 5.7 spriterenderer Struct Reference

```
#include <spriterenderer.h>
```

### Data Fields

- [entity entity](#)
- [texture \\* texture](#)
- [vector2f size](#)
- [uvs uvs](#)
- [atlas atlas](#)

### 5.7.1 Field Documentation

#### 5.7.1.1 atlas

```
atlas spriterenderer::atlas
```

#### 5.7.1.2 entity

```
entity spriterenderer::entity
```

### 5.7.1.3 size

```
vector2f spriterenderer::size
```

### 5.7.1.4 texture

```
texture* spriterenderer::texture
```

### 5.7.1.5 uvs

```
uvs spriterenderer::uvs
```

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

- [include/vally/ecs/components/spriterenderer.h](#)

## 5.8 texture Struct Reference

Texture struct.

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

### Data Fields

- [u32 id](#)
- [i32 width](#)
- [i32 height](#)

### 5.8.1 Detailed Description

Texture struct.

Contains OpenGL texture and its dimensions.

INFO: Recommended declaring a pointer.

### 5.8.2 Field Documentation

#### 5.8.2.1 height

`i32 texture::height`

#### 5.8.2.2 id

`u32 texture::id`

#### 5.8.2.3 width

`i32 texture::width`

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

- `include/vally/renderer/texture.h`

## 5.9 transform Struct Reference

```
#include <transform.h>
```

### Data Fields

- `entity entity`
- `vector2f position`
- `vector2f scale`

### 5.9.1 Field Documentation

#### 5.9.1.1 entity

`entity transform::entity`

#### 5.9.1.2 position

`vector2f transform::position`



### 5.9.1.3 scale

```
vector2f transform::scale
```

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

- include/vally/ecs/components/[transform.h](#)

## 5.10 uvs Struct Reference

```
#include <spriterenderer.h>
```

### Data Fields

- [vector2f uv](#) [4]

### 5.10.1 Field Documentation

#### 5.10.1.1 uv

```
vector2f uvs::uv[4]
```

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

- include/vally/ecs/components/[spriterenderer.h](#)

## 5.11 vector2f Struct Reference

```
#include <math_types.h>
```

### Data Fields

- [f32 x](#)
- [f32 y](#)

### 5.11.1 Field Documentation

#### 5.11.1.1 x

```
f32 vector2f::x
```

#### 5.11.1.2 y

```
f32 vector2f::y
```

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

- include/vally/math/[math\\_types.h](#)

## 5.12 vector2i Struct Reference

```
#include <math_types.h>
```

### Data Fields

- [i32 x](#)
- [i32 y](#)

### 5.12.1 Field Documentation

#### 5.12.1.1 x

```
i32 vector2i::x
```

#### 5.12.1.2 y

```
i32 vector2i::y
```

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

- include/vally/math/[math\\_types.h](#)

## 5.13 vector3f Struct Reference

```
#include <math_types.h>
```

## Data Fields

- [f32 x](#)
- [f32 y](#)
- [f32 z](#)

### 5.13.1 Field Documentation

#### 5.13.1.1 x

[f32](#) `vector3f::x`

#### 5.13.1.2 y

[f32](#) `vector3f::y`

#### 5.13.1.3 z

[f32](#) `vector3f::z`

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

- `include/vally/math/math\_types.h`

## 5.14 vector3i Struct Reference

```
#include <math_types.h>
```

## Data Fields

- [i32 x](#)
- [i32 y](#)
- [i32 z](#)

### 5.14.1 Field Documentation

**5.14.1.1 x**

```
i32 vector3i::x
```

**5.14.1.2 y**

```
i32 vector3i::y
```

**5.14.1.3 z**

```
i32 vector3i::z
```

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

- `include/vally/math/math_types.h`

## Chapter 6

# File Documentation

### 6.1 include/vally/config.h File Reference

This header file contains main declarations and macros.

```
#include <stdint.h>
```

#### Macros

- `#define VALLY_API`  
*Identify OS.*
- `#define TRUE 1`  
*Boolean values.*
- `#define FALSE 0`
- `#define NULL ((void *)0)`
- `#define BIT_MASK(n) (1 << n)`  
*Creates a bit mask with certain bit toggled on.*
- `#define MIN(a, b) (((a)<(b))?(a):(b))`
- `#define MAX(a, b) (((a)>(b))?(a):(b))`

#### Typedefs

- `typedef uint8_t u8`  
*Fixed-size unsigned types.*
- `typedef uint16_t u16`
- `typedef uint32_t u32`
- `typedef uint64_t u64`
- `typedef int8_t i8`  
*Fixed-size integer types.*
- `typedef int16_t i16`
- `typedef int32_t i32`
- `typedef int64_t i64`
- `typedef float f32`  
*Fixed-size decimal types.*
- `typedef double f64`
- `typedef int32_t b32`  
*Custom booleans.*
- `typedef uint8_t b8`

### 6.1.1 Detailed Description

This header file contains main declarations and macros.

#### Author

Dmytro Zykov

#### Date

12 Jan 2022

### 6.1.2 Macro Definition Documentation

#### 6.1.2.1 BIT\_MASK

```
#define BIT_MASK(  
    n ) (1 << n)
```

Creates a bit mask with certain bit toggled on.

#### 6.1.2.2 FALSE

```
#define FALSE 0
```

#### 6.1.2.3 MAX

```
#define MAX(  
    a,  
    b ) ((a) > (b)) ? (a) : (b)
```

#### 6.1.2.4 MIN

```
#define MIN(  
    a,  
    b ) ((a) < (b)) ? (a) : (b)
```

#### 6.1.2.5 NULL

```
#define NULL ((void *)0)
```

#### 6.1.2.6 TRUE

```
#define TRUE 1
```

Boolean values.

#### 6.1.2.7 VALLY\_API

```
#define VALLY_API
```

Identify OS.

Define import and export macros.

### 6.1.3 Typedef Documentation

#### 6.1.3.1 b32

```
typedef int32_t b32
```

Custom booleans.

#### 6.1.3.2 b8

```
typedef uint8_t b8
```

#### 6.1.3.3 f32

```
typedef float f32
```

Fixed-size decimal types.

#### 6.1.3.4 f64

```
typedef double f64
```

#### 6.1.3.5 i16

```
typedef int16_t i16
```

#### 6.1.3.6 i32

```
typedef int32_t i32
```

#### 6.1.3.7 i64

```
typedef int64_t i64
```

#### 6.1.3.8 i8

```
typedef int8_t i8
```

Fixed-size integer types.

#### 6.1.3.9 u16

```
typedef uint16_t u16
```

#### 6.1.3.10 u32

```
typedef uint32_t u32
```



### 6.1.3.11 u64

```
typedef uint64_t u64
```

### 6.1.3.12 u8

```
typedef uint8_t u8
```

Fixed-size unsigned types.

## 6.2 config.h

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

```
1 /*****
2  * config.h
3  *
4  * Copyright (c) 2022 Dmytro Zykov
5  *
6  * This file is a part of the vally project, and may only be used,
7  * modified and distributed under the terms of the MIT License,
8  * LICENSE.md. By continuing to use, modify and distribute this file
9  * you indicate that you have read the license and accept it fully.
10 *****/
11
12 #ifndef VALLY_CONFIG_H_
13 #define VALLY_CONFIG_H_
14
15 #include <stdint.h>
16
17 #if defined(_WIN32)
18     // Windows
19     #define VALLY_PLATFORM_WINDOWS
20
21 #elif defined(__linux__)
22     // Linux
23     #define VALLY_PLATFORM_LINUX
24
25 #else
26     #error This OS is not supported by Vally
27 #endif
28
29 #if !defined(VALLY_STATIC)
30     #if defined(VALLY_PLATFORM_WINDOWS)
31         // Special keywords for Windows
32         #if defined(VALLY_BUILD_DLL)
33             #define VALLY_API __declspec(dllexport)
34         #else
35             #define VALLY_API __declspec(dllimport)
36         #endif
37     #if defined(_MSC_VER)
38         #pragma warning(disable: 4251)
39     #endif
40 #else // Linux
41     #if defined(VALLY_BUILD_DLL)
42         #define VALLY_API __attribute__((visibility("default")))
43     #endif
44 #endif
45 #endif
```

```

74
75     #else
76
77         #define VALLY_API
78
79     #endif
80
81 #endif
82
83 #else
84
85     // Static library does not require import and export macros
86     #define VALLY_API
87
88 #endif
89
90 typedef uint8_t u8;
91 typedef uint16_t u16;
92 typedef uint32_t u32;
93 typedef uint64_t u64;
94
95 typedef int8_t i8;
96 typedef int16_t i16;
97 typedef int32_t i32;
98 typedef int64_t i64;
99
100 typedef float f32;
101 typedef double f64;
102
103 typedef int32_t b32;
104 typedef uint8_t b8;
105
106 #define TRUE 1
107 #define FALSE 0
108
109 #define NULL ((void *)0)
110
111 #define BIT_MASK(n) (1 << n)
112
113 #define MIN(a,b) ((a)<(b))?(a):(b)
114 #define MAX(a,b) ((a)>(b))?(a):(b)
115
116 #endif // VALLY_CONFIG_H_

```

## 6.3 include/vally/core/engine.h File Reference

```
#include <vally/config.h>
```

### Typedefs

- typedef [b8](#)(\* [engine\\_start](#)) (void)  
*Client API function pointer.*
- typedef [b8](#)(\* [engine\\_update](#)) ([f32](#) delta\_time)  
*Client API function pointer.*
- typedef [b8](#)(\* [engine\\_render](#)) ([f32](#) delta\_time)  
*Client API function pointer.*

### Functions

- [VALLY\\_API b8 engine\\_create](#) ([i16](#) width, [i16](#) height, const char \*title)  
*Initializes the vally engine.*
- [VALLY\\_API b8 engine\\_run](#) ([engine\\_start](#) start, [engine\\_update](#) update, [engine\\_render](#) render)  
*Runs the vally engine.*

## 6.3.1 Typedef Documentation

### 6.3.1.1 engine\_render

```
typedef b8(* engine_render) (f32 delta_time)
```

Client API function pointer.

Called inside main loop in rendering phase.

### 6.3.1.2 engine\_start

```
typedef b8(* engine_start) (void)
```

Client API function pointer.

Called after engine is initialized, before main loop.

### 6.3.1.3 engine\_update

```
typedef b8(* engine_update) (f32 delta_time)
```

Client API function pointer.

Called inside main loop before rendering phase.

## 6.3.2 Function Documentation

### 6.3.2.1 engine\_create()

```
VALLY_API b8 engine_create (  
    i16 width,  
    i16 height,  
    const char * title )
```

Initializes the vally engine.

This function initializes the vally engine. Must be called before any other vally function calls.

#### Parameters

<i>width</i>	Target window width
<i>height</i>	Target window height
<i>title</i>	Target window title

**Returns**

TRUE - if success, FALSE - in case of fatal errors

**6.3.2.2 engine\_run()**

```
VALLY_API b8 engine_run (
    engine_start start,
    engine_update update,
    engine_render render )
```

Runs the vally engine.

This function runs the vally engine.

**Parameters**

<i>start</i>	Function that will be called before main loop
<i>update</i>	Function that will be called before rendering
<i>render</i>	Function that will be called in rendering

**Returns**

TRUE - if success, FALSE - in case of fatal errors

**6.4 engine.h**

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

```
1  /*****
2  * engine.h
3  *
4  * Copyright (c) 2022 Dmytro Zykov
5  *
6  * This file is a part of the vally project, and may only be used,
7  * modified and distributed under the terms of the MIT License,
8  * LICENSE.md. By continuing to use, modify and distribute this file
9  * you indicate that you have read the license and accept it fully.
10 * *****/
11
12 #ifndef VALLY_ENGINE_H_
13 #define VALLY_ENGINE_H_
14
15 #include <vally/config.h>
16
22 typedef b8 (*engine_start)(void);
23
24
30 typedef b8 (*engine_update)(f32 delta_time);
31
37 typedef b8 (*engine_render)(f32 delta_time);
38
52 VALLY_API b8 engine_create(i16 width, i16 height, const char *title);
53
66 VALLY_API b8 engine_run(engine_start start, engine_update update, engine_render render);
67
68 #endif // VALLY_ENGINE_H_
```

## 6.5 include/vally/core/event.h File Reference

```
#include <vally/config.h>
```

### Data Structures

- struct [event\\_context](#)  
*Context send alongside the event.*

### Typedefs

- typedef [b8](#)(\* [on\\_event\\_fun](#)) ([u16](#) code, void \*sender, void \*listener, [event\\_context](#) context)  
*Pointer to event function.*

### Enumerations

- enum [engine\\_event\\_code](#) {  
[EVENT\\_CODE\\_APP\\_QUIT](#) , [EVENT\\_CODE\\_KEY\\_PRESSED](#) , [EVENT\\_CODE\\_KEY\\_RELEASED](#) ,  
[EVENT\\_CODE\\_MOUSE\\_BUTTON\\_PRESSED](#) ,  
[EVENT\\_CODE\\_MOUSE\\_BUTTON\\_RELEASED](#) , [EVENT\\_CODE\\_MOUSE\\_MOVED](#) , [EVENT\\_CODE\\_MOUSE\\_WHEEL](#)  
, [EVENT\\_CODE\\_WINDOW\\_RESIZED](#) ,  
[EVENT\\_CODE\\_CAMERA\\_MOVED](#) , [MAX\\_EVENT\\_CODE](#) = 255 }  
*Event type codes.*

### Functions

- [b8 event\\_init](#) ()
- void [event\\_terminate](#) ()
- [VALLY\\_API b8 event\\_subscribe](#) ([u16](#) code, void \*listener, [on\\_event\\_fun](#) callback)  
*Subscribes for certain event types.*
- [VALLY\\_API b8 event\\_unsubscribe](#) ([u16](#) code, void \*listener, [on\\_event\\_fun](#) callback)  
*Unsubscribes from certain event types.*
- [VALLY\\_API b8 event\\_publish](#) ([u16](#) code, void \*sender, [event\\_context](#) context)  
*Publishes event of certain type.*

#### 6.5.1 Typedef Documentation

##### 6.5.1.1 on\_event\_fun

```
typedef b8(* on_event_fun) (u16 code, void *sender, void *listener, event\_context context)
```

Pointer to event function.

## 6.5.2 Enumeration Type Documentation

### 6.5.2.1 engine\_event\_code

enum `engine_event_code`

Event type codes.

User defined event codes must have code larger then 255

Enumerator

EVENT_CODE_APP_QUIT	
EVENT_CODE_KEY_PRESSED	
EVENT_CODE_KEY_RELEASED	
EVENT_CODE_MOUSE_BUTTON_PRESSED	
EVENT_CODE_MOUSE_BUTTON_RELEASED	
EVENT_CODE_MOUSE_MOVED	
EVENT_CODE_MOUSE_WHEEL	
EVENT_CODE_WINDOW_RESIZED	
EVENT_CODE_CAMERA_MOVED	
MAX_EVENT_CODE	

## 6.5.3 Function Documentation

### 6.5.3.1 event\_init()

`b8 event_init ( )`

### 6.5.3.2 event\_publish()

```
VALLY_API b8 event_publish (
    u16 code,
    void * sender,
    event_context context )
```

Publishes event of certain type.

Parameters

<i>code</i>	Event code
<i>sender</i>	Object that published the event
<i>context</i>	Context information sent alongside the event

**Returns**

TRUE - if success, FALSE - in case of fatal errors

**6.5.3.3 event\_subscribe()**

```
VALLY_API b8 event_subscribe (
    u16 code,
    void * listener,
    on_event_fun callback )
```

Subscribes for certain event types.

Executes the callback function when event is published

**Parameters**

<i>code</i>	Event code
<i>listener</i>	Object that subscribes for an event
<i>callback</i>	Event callback function

**Returns**

TRUE - if success, FALSE - in case of fatal errors

**6.5.3.4 event\_terminate()**

```
void event_terminate ( )
```

**6.5.3.5 event\_unsubscribe()**

```
VALLY_API b8 event_unsubscribe (
    u16 code,
    void * listener,
    on_event_fun callback )
```

Unsubscribes from certain event types.

Stops executing the callback function when event is published

**Parameters**

<i>code</i>	Event code
<i>listener</i>	Object that unsubscribes from an event
<i>callback</i>	Event callback function

**Returns**

TRUE - if success, FALSE - in case of fatal errors

**6.6 event.h**

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

```

1 //=====
2 // event.h
3 //
4 // Copyright (c) 2022 Dmytro Zykov
5 //
6 // This file is a part of the vally project, and may only be used,
7 // modified and distributed under the terms of the MIT License,
8 // LICENSE.md. By continuing to use, modify and distribute this file
9 // you indicate that you have read the license and accept it fully.
10 //=====
11
12 #ifndef VALLY_EVENT_H_
13 #define VALLY_EVENT_H_
14
15 #include <vally/config.h>
16
21 typedef struct {
26     union {
27         i64 i64[2];
28         u64 u64[2];
29         f64 f64[2];
30
31         i32 i32[4];
32         u32 u32[4];
33         f32 f32[4];
34
35         i16 i16[8];
36         u16 u16[8];
37
38         i8 i8[8];
39         u8 u8[8];
40
41         char c[16];
42     } data;
43 } event_context;
44
49 typedef b8 (*on_event_fun)(u16 code, void *sender, void *listener, event_context context);
50
51 b8 event_init();
52 void event_terminate();
53
66 VALLY_API b8 event_subscribe(u16 code, void *listener, on_event_fun callback);
67
80 VALLY_API b8 event_unsubscribe(u16 code, void *listener, on_event_fun callback);
81
92 VALLY_API b8 event_publish(u16 code, void *sender, event_context context);
93
100 typedef enum {
101     EVENT_CODE_APP_QUIT,
102     EVENT_CODE_KEY_PRESSED,
103     EVENT_CODE_KEY_RELEASED,
104     EVENT_CODE_MOUSE_BUTTON_PRESSED,
105     EVENT_CODE_MOUSE_BUTTON_RELEASED,
106     EVENT_CODE_MOUSE_MOVED,
107     EVENT_CODE_MOUSE_WHEEL,
108     EVENT_CODE_WINDOW_RESIZED,
109     EVENT_CODE_CAMERA_MOVED,
110
111     MAX_EVENT_CODE = 255
112 } engine_event_code;
113
114 #endif // VALLY_EVENT_H_

```

**6.7 include/vally/core/input.h File Reference**

```
#include <vally/config.h>
```



## Macros

- #define [KEY\\_SPACE](#) 32
- #define [KEY\\_APOSTROPHE](#) 39
- #define [KEY\\_COMMA](#) 44
- #define [KEY\\_MINUS](#) 45
- #define [KEY\\_PERIOD](#) 46
- #define [KEY\\_SLASH](#) 47
- #define [KEY\\_0](#) 48
- #define [KEY\\_1](#) 49
- #define [KEY\\_2](#) 50
- #define [KEY\\_3](#) 51
- #define [KEY\\_4](#) 52
- #define [KEY\\_5](#) 53
- #define [KEY\\_6](#) 54
- #define [KEY\\_7](#) 55
- #define [KEY\\_8](#) 56
- #define [KEY\\_9](#) 57
- #define [KEY\\_SEMICOLON](#) 59
- #define [KEY\\_EQUAL](#) 61
- #define [KEY\\_A](#) 65
- #define [KEY\\_B](#) 66
- #define [KEY\\_C](#) 67
- #define [KEY\\_D](#) 68
- #define [KEY\\_E](#) 69
- #define [KEY\\_F](#) 70
- #define [KEY\\_G](#) 71
- #define [KEY\\_H](#) 72
- #define [KEY\\_I](#) 73
- #define [KEY\\_J](#) 74
- #define [KEY\\_K](#) 75
- #define [KEY\\_L](#) 76
- #define [KEY\\_M](#) 77
- #define [KEY\\_N](#) 78
- #define [KEY\\_O](#) 79
- #define [KEY\\_P](#) 80
- #define [KEY\\_Q](#) 81
- #define [KEY\\_R](#) 82
- #define [KEY\\_S](#) 83
- #define [KEY\\_T](#) 84
- #define [KEY\\_U](#) 85
- #define [KEY\\_V](#) 86
- #define [KEY\\_W](#) 87
- #define [KEY\\_X](#) 88
- #define [KEY\\_Y](#) 89
- #define [KEY\\_Z](#) 90
- #define [KEY\\_LEFT\\_BRACKET](#) 91
- #define [KEY\\_BACKSLASH](#) 92
- #define [KEY\\_RIGHT\\_BRACKET](#) 93
- #define [KEY\\_GRAVE\\_ACCENT](#) 96
- #define [KEY\\_WORLD\\_1](#) 161
- #define [KEY\\_WORLD\\_2](#) 162
- #define [KEY\\_ESCAPE](#) 256
- #define [KEY\\_ENTER](#) 257
- #define [KEY\\_TAB](#) 258

- `#define KEY_BACKSPACE` 259
- `#define KEY_INSERT` 260
- `#define KEY_DELETE` 261
- `#define KEY_RIGHT` 262
- `#define KEY_LEFT` 263
- `#define KEY_DOWN` 264
- `#define KEY_UP` 265
- `#define KEY_PAGE_UP` 266
- `#define KEY_PAGE_DOWN` 267
- `#define KEY_HOME` 268
- `#define KEY_END` 269
- `#define KEY_CAPS_LOCK` 280
- `#define KEY_SCROLL_LOCK` 281
- `#define KEY_NUM_LOCK` 282
- `#define KEY_PRINT_SCREEN` 283
- `#define KEY_PAUSE` 284
- `#define KEY_F1` 290
- `#define KEY_F2` 291
- `#define KEY_F3` 292
- `#define KEY_F4` 293
- `#define KEY_F5` 294
- `#define KEY_F6` 295
- `#define KEY_F7` 296
- `#define KEY_F8` 297
- `#define KEY_F9` 298
- `#define KEY_F10` 299
- `#define KEY_F11` 300
- `#define KEY_F12` 301
- `#define KEY_F13` 302
- `#define KEY_F14` 303
- `#define KEY_F15` 304
- `#define KEY_F16` 305
- `#define KEY_F17` 306
- `#define KEY_F18` 307
- `#define KEY_F19` 308
- `#define KEY_F20` 309
- `#define KEY_F21` 310
- `#define KEY_F22` 311
- `#define KEY_F23` 312
- `#define KEY_F24` 313
- `#define KEY_F25` 314
- `#define KEY_KP_0` 320
- `#define KEY_KP_1` 321
- `#define KEY_KP_2` 322
- `#define KEY_KP_3` 323
- `#define KEY_KP_4` 324
- `#define KEY_KP_5` 325
- `#define KEY_KP_6` 326
- `#define KEY_KP_7` 327
- `#define KEY_KP_8` 328
- `#define KEY_KP_9` 329
- `#define KEY_KP_DECIMAL` 330
- `#define KEY_KP_DIVIDE` 331
- `#define KEY_KP_MULTIPLY` 332
- `#define KEY_KP_SUBTRACT` 333

- #define [KEY\\_KP\\_ADD](#) 334
- #define [KEY\\_KP\\_ENTER](#) 335
- #define [KEY\\_KP\\_EQUAL](#) 336
- #define [KEY\\_LEFT\\_SHIFT](#) 340
- #define [KEY\\_LEFT\\_CONTROL](#) 341
- #define [KEY\\_LEFT\\_ALT](#) 342
- #define [KEY\\_LEFT\\_SUPER](#) 343
- #define [KEY\\_RIGHT\\_SHIFT](#) 344
- #define [KEY\\_RIGHT\\_CONTROL](#) 345
- #define [KEY\\_RIGHT\\_ALT](#) 346
- #define [KEY\\_RIGHT\\_SUPER](#) 347
- #define [KEY\\_MENU](#) 348
- #define [KEYS\\_MAX\\_KEYS](#) [KEY\\_MENU](#)
- #define [MOUSE\\_BUTTON\\_1](#) 0
- #define [MOUSE\\_BUTTON\\_2](#) 1
- #define [MOUSE\\_BUTTON\\_3](#) 2
- #define [MOUSE\\_BUTTON\\_4](#) 3
- #define [MOUSE\\_BUTTON\\_5](#) 4
- #define [MOUSE\\_BUTTON\\_6](#) 5
- #define [MOUSE\\_BUTTON\\_7](#) 6
- #define [MOUSE\\_BUTTON\\_8](#) 7
- #define [MOUSE\\_BUTTON\\_MAX](#) [MOUSE\\_BUTTON\\_8](#)
- #define [MOUSE\\_BUTTON\\_LEFT](#) [MOUSE\\_BUTTON\\_1](#)
- #define [MOUSE\\_BUTTON\\_RIGHT](#) [MOUSE\\_BUTTON\\_2](#)
- #define [MOUSE\\_BUTTON\\_MIDDLE](#) [MOUSE\\_BUTTON\\_3](#)

## Functions

- [b8 input\\_init](#) ()
- void [input\\_terminate](#) ()
- [VALLY\\_API b8 input\\_key\\_pressed](#) (u16 key)  
*Checks if keyboard key is pressed.*
- [VALLY\\_API b8 input\\_key\\_released](#) (u16 key)  
*Checks if keyboard key is released.*
- [VALLY\\_API b8 input\\_mouse\\_button\\_pressed](#) (u16 button)  
*Checks if mouse button is pressed.*
- [VALLY\\_API b8 input\\_mouse\\_button\\_released](#) (u16 button)  
*Checks if mouse button is released.*
- [VALLY\\_API void input\\_mouse\\_position](#) (f64 \*x, f64 \*y)  
*Returns current mouse position.*
- [VALLY\\_API void input\\_mouse\\_prev\\_position](#) (f64 \*x, f64 \*y)  
*Returns current previous position.*

### 6.7.1 Function Documentation

#### 6.7.1.1 input\_init()

```
b8 input_init ( )
```

### 6.7.1.2 input\_key\_pressed()

```
VALLY_API b8 input_key_pressed (  
    u16 key )
```

Checks if keyboard key is pressed.

#### Parameters

<i>key</i>	Keyboard key code
------------	-------------------

#### Returns

TRUE - if pressed, FALSE - if released

### 6.7.1.3 input\_key\_released()

```
VALLY_API b8 input_key_released (  
    u16 key )
```

Checks if keyboard key is released.

#### Parameters

<i>key</i>	Keyboard key code
------------	-------------------

#### Returns

TRUE - if released, FALSE - if pressed

### 6.7.1.4 input\_mouse\_button\_pressed()

```
VALLY_API b8 input_mouse_button_pressed (  
    u16 button )
```

Checks if mouse button is pressed.

#### Parameters

<i>button</i>	Mouse button code
---------------	-------------------

#### Returns

TRUE - if pressed, FALSE - if released

#### 6.7.1.5 input\_mouse\_button\_released()

```
VALLY_API b8 input_mouse_button_released (
    u16 button )
```

Checks if mouse button is released.

##### Parameters

<i>button</i>	Mouse button code
---------------	-------------------

##### Returns

TRUE - if released, FALSE - if pressed

#### 6.7.1.6 input\_mouse\_position()

```
VALLY_API void input_mouse_position (
    f64 * x,
    f64 * y )
```

Returns current mouse position.

##### Parameters

<i>x</i>	pointer where mouse x coordinate will be written
<i>y</i>	pointer where mouse y coordinate will be written

#### 6.7.1.7 input\_mouse\_prev\_position()

```
VALLY_API void input_mouse_prev_position (
    f64 * x,
    f64 * y )
```

Returns current previous position.

##### Parameters

<i>x</i>	pointer where mouse x coordinate will be written
<i>y</i>	pointer where mouse y coordinate will be written

### 6.7.1.8 input\_terminate()

```
void input_terminate ( )
```

## 6.8 input.h

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

```
1 //=====
2 // input.h
3 //
4 // Copyright (c) 2022 Dmytro Zykov
5 //
6 // This file is a part of the vally project, and may only be used,
7 // modified and distributed under the terms of the MIT License,
8 // LICENSE.md. By continuing to use, modify and distribute this file
9 // you indicate that you have read the license and accept it fully.
10 //=====
11
12 #ifndef VALLY_INPUT_H_
13 #define VALLY_INPUT_H_
14
15 #include <vally/config.h>
16
17
18
19
20 #define KEY_SPACE           32
21 #define KEY_APOSTROPHE     39
22 #define KEY_COMMA          44
23 #define KEY_MINUS          45
24 #define KEY_PERIOD         46
25 #define KEY_SLASH          47
26 #define KEY_0              48
27 #define KEY_1              49
28 #define KEY_2              50
29 #define KEY_3              51
30 #define KEY_4              52
31 #define KEY_5              53
32 #define KEY_6              54
33 #define KEY_7              55
34 #define KEY_8              56
35 #define KEY_9              57
36 #define KEY_SEMICOLON      59
37 #define KEY_EQUAL          61
38 #define KEY_A              65
39 #define KEY_B              66
40 #define KEY_C              67
41 #define KEY_D              68
42 #define KEY_E              69
43 #define KEY_F              70
44 #define KEY_G              71
45 #define KEY_H              72
46 #define KEY_I              73
47 #define KEY_J              74
48 #define KEY_K              75
49 #define KEY_L              76
50 #define KEY_M              77
51 #define KEY_N              78
52 #define KEY_O              79
53 #define KEY_P              80
54 #define KEY_Q              81
55 #define KEY_R              82
56 #define KEY_S              83
57 #define KEY_T              84
58 #define KEY_U              85
59 #define KEY_V              86
60 #define KEY_W              87
61 #define KEY_X              88
62 #define KEY_Y              89
63 #define KEY_Z              90
64 #define KEY_LEFT_BRACKET  91
65 #define KEY_BACKSLASH     92
66 #define KEY_RIGHT_BRACKET 93
67 #define KEY_GRAVE_ACCENT  96
68 #define KEY_WORLD_1       161
69 #define KEY_WORLD_2       162
70 #define KEY_ESCAPE        256
71 #define KEY_ENTER         257
72 #define KEY_TAB           258
73 #define KEY_BACKSPACE     259
74 #define KEY_INSERT        260
75 #define KEY_DELETE        261
```

```

76 #define KEY_RIGHT          262
77 #define KEY_LEFT           263
78 #define KEY_DOWN           264
79 #define KEY_UP              265
80 #define KEY_PAGE_UP        266
81 #define KEY_PAGE_DOWN      267
82 #define KEY_HOME           268
83 #define KEY_END             269
84 #define KEY_CAPS_LOCK       280
85 #define KEY_SCROLL_LOCK     281
86 #define KEY_NUM_LOCK        282
87 #define KEY_PRINT_SCREEN    283
88 #define KEY_PAUSE           284
89 #define KEY_F1              290
90 #define KEY_F2              291
91 #define KEY_F3              292
92 #define KEY_F4              293
93 #define KEY_F5              294
94 #define KEY_F6              295
95 #define KEY_F7              296
96 #define KEY_F8              297
97 #define KEY_F9              298
98 #define KEY_F10             299
99 #define KEY_F11             300
100 #define KEY_F12             301
101 #define KEY_F13             302
102 #define KEY_F14             303
103 #define KEY_F15             304
104 #define KEY_F16             305
105 #define KEY_F17             306
106 #define KEY_F18             307
107 #define KEY_F19             308
108 #define KEY_F20             309
109 #define KEY_F21             310
110 #define KEY_F22             311
111 #define KEY_F23             312
112 #define KEY_F24             313
113 #define KEY_F25             314
114 #define KEY_KP_0            320
115 #define KEY_KP_1            321
116 #define KEY_KP_2            322
117 #define KEY_KP_3            323
118 #define KEY_KP_4            324
119 #define KEY_KP_5            325
120 #define KEY_KP_6            326
121 #define KEY_KP_7            327
122 #define KEY_KP_8            328
123 #define KEY_KP_9            329
124 #define KEY_KP_DECIMAL      330
125 #define KEY_KP_DIVIDE       331
126 #define KEY_KP_MULTIPLY     332
127 #define KEY_KP_SUBTRACT     333
128 #define KEY_KP_ADD          334
129 #define KEY_KP_ENTER        335
130 #define KEY_KP_EQUAL        336
131 #define KEY_LEFT_SHIFT      340
132 #define KEY_LEFT_CONTROL    341
133 #define KEY_LEFT_ALT        342
134 #define KEY_LEFT_SUPER      343
135 #define KEY_RIGHT_SHIFT     344
136 #define KEY_RIGHT_CONTROL   345
137 #define KEY_RIGHT_ALT       346
138 #define KEY_RIGHT_SUPER     347
139 #define KEY_MENU            348
140 #define KEYS_MAX_KEYS       KEY_MENU
146 #define MOUSE_BUTTON_1      0
147 #define MOUSE_BUTTON_2      1
148 #define MOUSE_BUTTON_3      2
149 #define MOUSE_BUTTON_4      3
150 #define MOUSE_BUTTON_5      4
151 #define MOUSE_BUTTON_6      5
152 #define MOUSE_BUTTON_7      6
153 #define MOUSE_BUTTON_8      7
154 #define MOUSE_BUTTON_MAX    MOUSE_BUTTON_8
155 #define MOUSE_BUTTON_LEFT    MOUSE_BUTTON_1
156 #define MOUSE_BUTTON_RIGHT   MOUSE_BUTTON_2
157 #define MOUSE_BUTTON_MIDDLE   MOUSE_BUTTON_3
160 b8 input_init();
161 void input_terminate();
162
171 VALLY_API b8 input_key_pressed(u16 key);
172
181 VALLY_API b8 input_key_released(u16 key);
182
191 VALLY_API b8 input_mouse_button_pressed(u16 button);
192
201 VALLY_API b8 input_mouse_button_released(u16 button);

```

```

202
210 VALLY_API void input_mouse_position(f64 *x, f64 *y);
211
219 VALLY_API void input_mouse_prev_position(f64 *x, f64 *y);
220
221 #endif // VALLY_INPUT_H_
222

```

## 6.9 include/vally/ecs/components/animator.h File Reference

```

#include <vally/config.h>
#include <vally/renderer/texture.h>
#include <vally/ecs/ecs.h>

```

### Data Structures

- struct [animation](#)
- struct [animotor](#)

### Functions

- [VALLY\\_API animation animation\\_create](#) (texture \*texture, vector2f tiling, u32 first\_frame, u32 last\_frame, u32 framerate)
- [VALLY\\_API void ecs\\_animator\\_add](#) (entity entity)
- [VALLY\\_API void ecs\\_animator\\_play](#) (entity entity, animation \*animation)
- [VALLY\\_API void ecs\\_animator\\_stop](#) (entity entity)
- void [ecs\\_animator\\_update](#) (f32 dt)

### 6.9.1 Function Documentation

#### 6.9.1.1 animation\_create()

```

VALLY_API animation animation_create (
    texture * texture,
    vector2f tiling,
    u32 first_frame,
    u32 last_frame,
    u32 framerate )

```

#### 6.9.1.2 ecs\_animator\_add()

```

VALLY_API void ecs_animator_add (
    entity entity )

```



### 6.9.1.3 ecs\_animator\_play()

```
VALLY_API void ecs_animator_play (
    entity entity,
    animation * animation )
```

### 6.9.1.4 ecs\_animator\_stop()

```
VALLY_API void ecs_animator_stop (
    entity entity )
```

### 6.9.1.5 ecs\_animator\_update()

```
void ecs_animator_update (
    f32 dt )
```

## 6.10 animator.h

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

```
1 #ifndef VALLY_ECS_COMPONENTS_ANIMATOR_H_
2 #define VALLY_ECS_COMPONENTS_ANIMATOR_H_
3
4 #include <vally/config.h>
5
6 #include <vally/renderer/texture.h>
7 #include <vally/ecs/ecs.h>
8
9 typedef struct {
10     texture *texture;
11     vector2f tiling;
12     u32 first_frame;
13     u32 last_frame;
14     f32 frametime;
15 } animation;
16
17 typedef struct {
18     b8 playing;
19     animation *animation;
20     u32 current_frame;
21     f32 time;
22 } animator;
23
24 VALLY_API animation animation_create(texture *texture, vector2f tiling, u32 first_frame, u32 last_frame,
    u32 framerate);
25
26 VALLY_API void ecs_animator_add(entity entity);
27
28 VALLY_API void ecs_animator_play(entity entity, animation *animation);
29
30 VALLY_API void ecs_animator_stop(entity entity);
31
32 void ecs_animator_update(f32 dt);
33
34 #endif // !VALLY_ECS_COMPONENTS_ANIMATOR_H_
```

## 6.11 include/vally/ecs/components/spriterenderer.h File Reference

```
#include <vally/config.h>
#include <vally/renderer/texture.h>
#include <vally/ecs/ecs.h>
#include <vally/math/math.h>
```

### Data Structures

- struct [uvs](#)
- struct [atlas](#)
- struct [spriterenderer](#)

### Functions

- [VALLY\\_API](#) void [ecs\\_spriterenderer\\_add](#) (entity entity, texture \*texture)
- [VALLY\\_API](#) void [ecs\\_spriterenderer\\_add\\_from\\_atlas](#) (entity entity, texture \*texture, vector2f tiling)
- [VALLY\\_API](#) void [ecs\\_spriterenderer\\_atlas\\_select](#) (entity entity, u32 index)
- [VALLY\\_API](#) [spriterenderer](#) \* [ecs\\_spriterenderer\\_get](#) (entity entity)
- void [ecs\\_spriterenderer\\_update](#) ()

### 6.11.1 Function Documentation

#### 6.11.1.1 [ecs\\_spriterenderer\\_add\(\)](#)

```
VALLY\_API void ecs\_spriterenderer\_add (  
    entity entity,  
    texture * texture )
```

#### 6.11.1.2 [ecs\\_spriterenderer\\_add\\_from\\_atlas\(\)](#)

```
VALLY\_API void ecs\_spriterenderer\_add\_from\_atlas (  
    entity entity,  
    texture * texture,  
    vector2f tiling )
```

#### 6.11.1.3 [ecs\\_spriterenderer\\_atlas\\_select\(\)](#)

```
VALLY\_API void ecs\_spriterenderer\_atlas\_select (  
    entity entity,  
    u32 index )
```

## 6.11.1.4 ecs\_spriterenderer\_get()

```
VALLY_API spriterenderer * ecs_spriterenderer_get (
    entity entity )
```

## 6.11.1.5 ecs\_spriterenderer\_update()

```
void ecs_spriterenderer_update ( )
```

## 6.12 spriterenderer.h

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

```
1 #ifndef VALLY_ECS_COMPONENTS_SPRITERENDERER_H_
2 #define VALLY_ECS_COMPONENTS_SPRITERENDERER_H_
3
4 #include <vally/config.h>
5 #include <vally/renderer/texture.h>
6 #include <vally/ecs/ecs.h>
7 #include <vally/math/math.h>
8
9 typedef struct {
10     vector2f uv[4];
11 } uvs;
12
13 typedef struct {
14     vector2f tiling;
15     u32 sprite_count;
16     u32 sprites_per_row;
17     uvs *uvs;
18 } atlas;
19
20 typedef struct {
21     entity entity;
22     texture *texture;
23     vector2f size;
24     uvs uvs;
25     atlas atlas;
26 } spriterenderer;
27
28 VALLY_API void ecs_spriterenderer_add(entity entity, texture *texture);
29
30 VALLY_API void ecs_spriterenderer_add_from_atlas(entity entity, texture *texture, vector2f tiling);
31
32 VALLY_API void ecs_spriterenderer_atlas_select(entity entity, u32 index);
33
34 VALLY_API spriterenderer *ecs_spriterenderer_get(entity entity);
35
36 void ecs_spriterenderer_update();
37
38 #endif // !VALLY_ECS_COMPONENTS_SPRITERENDERER_H_
```

## 6.13 include/vally/ecs/components/transform.h File Reference

```
#include <vally/config.h>
#include <vally/ecs/ecs.h>
#include <vally/math/math.h>
```

## Data Structures

- struct [transform](#)

## Functions

- [VALLY\\_API](#) void `ecs_transform_add` (`entity` `entity`, `vector2f` `position`, `vector2f` `scale`)
- [VALLY\\_API](#) `transform` \* `ecs_transform_get` (`entity` `entity`)

### 6.13.1 Function Documentation

#### 6.13.1.1 `ecs_transform_add()`

```
VALLY_API void ecs_transform_add (
    entity entity,
    vector2f position,
    vector2f scale )
```

#### 6.13.1.2 `ecs_transform_get()`

```
VALLY_API transform * ecs_transform_get (
    entity entity )
```

## 6.14 `transform.h`

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

```
1 #ifndef VALLY_ECS_COMPONENTS_TRANSFORM_H_
2 #define VALLY_ECS_COMPONENTS_TRANSFORM_H_
3
4 #include <vally/config.h>
5 #include <vally/ecs/ecs.h>
6 #include <vally/math/math.h>
7
8 typedef struct {
9     entity entity;
10    vector2f position;
11    vector2f scale;
12 } transform;
13
14 VALLY_API void ecs_transform_add(entity entity, vector2f position, vector2f scale);
15
16 VALLY_API transform *ecs_transform_get(entity entity);
17
18 #endif // !VALLY_ECS_COMPONENTS_TRANSFORM_H_
```

## 6.15 `include/vally/ecs/ecs.h` File Reference

```
#include <vally/config.h>
#include <vally/ecs/ecs_types.h>
#include <vally/ecs/components/spriterenderer.h>
#include <vally/ecs/components/transform.h>
#include <vally/ecs/components/animatorm.h>
```

## Macros

- #define `ECS_NULL_ENTITY` -1
- #define `ECS_MAX_ENTITIES` 1000
- #define `ECS_MAX_COMPONENTS` `ECS_MAX_ENTITIES`
- #define `ecs_is_alive(entity)` `ecs_component_has(entity, ECS_ALIVE)`

## Functions

- `b8 ecs_init()`
- `VALLY_API entity ecs_entity_create()`
- `VALLY_API void ecs_component_add(entity entity, component_mask mask)`
- `VALLY_API void ecs_component_remove(entity entity, component_mask mask)`
- `b8 ecs_component_has(entity entity, component_mask mask)`
- `entity ecs_get_last()`

### 6.15.1 Macro Definition Documentation

#### 6.15.1.1 `ecs_is_alive`

```
#define ecs_is_alive(  
    entity ) ecs_component_has(entity, ECS_ALIVE)
```

#### 6.15.1.2 `ECS_MAX_COMPONENTS`

```
#define ECS_MAX_COMPONENTS ECS_MAX_ENTITIES
```

#### 6.15.1.3 `ECS_MAX_ENTITIES`

```
#define ECS_MAX_ENTITIES 1000
```

#### 6.15.1.4 `ECS_NULL_ENTITY`

```
#define ECS_NULL_ENTITY -1
```

### 6.15.2 Function Documentation

#### 6.15.2.1 ecs\_component\_add()

```
VALLY_API void ecs_component_add (
    entity entity,
    component_mask mask )
```

#### 6.15.2.2 ecs\_component\_has()

```
b8 ecs_component_has (
    entity entity,
    component_mask mask )
```

#### 6.15.2.3 ecs\_component\_remove()

```
VALLY_API void ecs_component_remove (
    entity entity,
    component_mask mask )
```

#### 6.15.2.4 ecs\_entity\_create()

```
VALLY_API entity ecs_entity_create ( )
```

#### 6.15.2.5 ecs\_get\_last()

```
entity ecs_get_last ( )
```

#### 6.15.2.6 ecs\_init()

```
b8 ecs_init ( )
```

## 6.16 ecs.h

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

```

1 #ifndef VALLY_ECS_ECS_H_
2 #define VALLY_ECS_ECS_H_
3
4 #include <vally/config.h>
5 #include <vally/ecs/ecs_types.h>
6 #include <vally/ecs/components/spriterenderer.h>
7 #include <vally/ecs/components/transform.h>
8 #include <vally/ecs/components/animotor.h>
9
10 #define ECS_NULL_ENTITY -1
11 #define ECS_MAX_ENTITIES 1000
12 #define ECS_MAX_COMPONENTS ECS_MAX_ENTITIES
13
14 b8 ecs_init();
15
16 VALLY_API entity ecs_entity_create();
17
18 VALLY_API void ecs_component_add(entity entity, component_mask mask);
19
20 VALLY_API void ecs_component_remove(entity entity, component_mask mask);
21
22 b8 ecs_component_has(entity entity, component_mask mask);
23
24 entity ecs_get_last();
25
26 #define ecs_is_alive(entity) ecs_component_has(entity, ECS_ALIVE)
27
28 #endif // !VALLY_ECS_ECS_H_

```

## 6.17 include/vally/ecs/ecs\_types.h File Reference

```
#include <vally/config.h>
```

### Typedefs

- typedef [i32](#) [entity](#)

### Enumerations

- enum [component\\_mask](#) {  
[ECS\\_ALIVE](#) = [BIT\\_MASK](#)(0) , [ECS\\_TRANSFORM\\_COMPONENT](#) = [BIT\\_MASK](#)(1) , [ECS\\_SPRITERENDERER\\_COMPONENT](#) = [BIT\\_MASK](#)(2) , [ECS\\_RIGIDBODY\\_COMPONENT](#) = [BIT\\_MASK](#)(3) ,  
[ECS\\_ANIMATOR\\_COMPONENT](#) = [BIT\\_MASK](#)(4) }

### 6.17.1 Typedef Documentation

#### 6.17.1.1 entity

```
typedef i32 entity
```

### 6.17.2 Enumeration Type Documentation

#### 6.17.2.1 component\_mask

```
enum component\_mask
```

## Enumerator

ECS_ALIVE	
ECS_TRANSFORM_COMPONENT	
ECS_SPRITERENDERER_COMPONENT	
ECS_RIGIDBODY_COMPONENT	
ECS_ANIMATOR_COMPONENT	

## 6.18 ecs\_types.h

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

```

1 #ifndef VALLY_ECS_ECS_COMPONENTS_H_
2 #define VALLY_ECS_ECS_COMPONENTS_H_
3
4 #include <vally/config.h>
5
6 typedef i32 entity;
7
8 typedef enum {
9     ECS_ALIVE = BIT_MASK(0),
10    ECS_TRANSFORM_COMPONENT = BIT_MASK(1),
11    ECS_SPRITERENDERER_COMPONENT = BIT_MASK(2),
12    ECS_RIGIDBODY_COMPONENT = BIT_MASK(3),
13    ECS_ANIMATOR_COMPONENT = BIT_MASK(4),
14 } component_mask;
15
16 #endif // !VALLY_ECS_ECS_COMPONENTS_H_

```

## 6.19 include/vally/math/math.h File Reference

```

#include <vally/config.h>
#include <vally/math/math_types.h>

```

### Macros

- #define [V\\_PI](#) 3.141592653589793F
- #define [V\\_2PI](#) [V\\_PI](#) \* 2.0F
- #define [V\\_PI\\_2](#) [V\\_PI](#) / 2.0F
- #define [V\\_PI\\_3](#) [V\\_PI](#) / 3.0F
- #define [V\\_PI\\_4](#) [V\\_PI](#) / 4.0F
- #define [V\\_SQRT2](#) 1.41421356237F
- #define [LERP](#)(w, v1, v2) ((1.0 - (w)) \* (v1) + (w) \* (v2))
- #define [vector2f\\_create](#)(x, y) (([vector2f](#)){x, y})
- #define [vector3f\\_create](#)(x, y, z) (([vector3f](#)){x, y, z})
- #define [vector2i\\_create](#)(x, y) (([vector2i](#)){x, y})
- #define [vector3i\\_create](#)(x, y, z) (([vector3i](#)){x, y, z})
- #define [vector2f\\_zero](#) [vector2f\\_create](#)(0.0f, 0.0f)
- #define [vector2f\\_one](#) [vector2f\\_create](#)(1.0f, 1.0f)
- #define [vector2f\\_right](#) [vector2f\\_create](#)(1.0f, 0.0f)
- #define [vector2f\\_left](#) [vector2f\\_create](#)(-1.0f, 0.0f)
- #define [vector2f\\_up](#) [vector2f\\_create](#)(0.0f, 1.0f)
- #define [vector2f\\_down](#) [vector2f\\_create](#)(0.0f, -1.0f)



## Functions

- [VALLY\\_API f32 vsin \(f32 x\)](#)
- [VALLY\\_API f32 vcos \(f32 x\)](#)
- [VALLY\\_API f32 vtan \(f32 x\)](#)
- [VALLY\\_API f32 vasin \(f32 x\)](#)
- [VALLY\\_API f32 vacos \(f32 x\)](#)
- [VALLY\\_API f32 vatan \(f32 x\)](#)
- [VALLY\\_API f32 vsqrt \(f32 x\)](#)
- [VALLY\\_API f32 vabs \(f32 x\)](#)
- [VALLY\\_INLINE vector2f vector2f\\_add \(vector2f a, vector2f b\)](#)
- [VALLY\\_INLINE vector2f vector2f\\_sub \(vector2f a, vector2f b\)](#)
- [VALLY\\_INLINE vector2f vector2f\\_mul \(vector2f a, vector2f b\)](#)
- [VALLY\\_INLINE vector2f vector2f\\_muls \(vector2f vector, f32 scalar\)](#)
- [VALLY\\_INLINE vector2f vector2f\\_divs \(vector2f vector, f32 scalar\)](#)
- [VALLY\\_INLINE vector2f vector2f\\_div \(vector2f a, vector2f b\)](#)
- [VALLY\\_INLINE f32 vector2f\\_length \(vector2f vector\)](#)
- [VALLY\\_INLINE vector2f vector2f\\_normalized \(vector2f vector\)](#)
- [VALLY\\_INLINE void vector2f\\_normalize \(vector2f \\*vector\)](#)
- [VALLY\\_INLINE f32 vector2f\\_distance \(vector2f a, vector2f b\)](#)
- [VALLY\\_INLINE vector3f vector3f\\_add \(vector3f a, vector3f b\)](#)
- [VALLY\\_INLINE vector3f vector3f\\_sub \(vector3f a, vector3f b\)](#)
- [VALLY\\_INLINE vector3f vector3f\\_cross \(vector3f a, vector3f b\)](#)
- [VALLY\\_INLINE f32 vector3f\\_dot \(vector3f a, vector3f b\)](#)
- [VALLY\\_INLINE vector3f vector3f\\_mul \(vector3f a, vector3f b\)](#)
- [VALLY\\_INLINE vector3f vector3f\\_div \(vector3f a, vector3f b\)](#)
- [VALLY\\_INLINE vector3f vector3f\\_muls \(vector3f vector, f32 scalar\)](#)
- [VALLY\\_INLINE vector3f vector3f\\_divs \(vector3f vector, f32 scalar\)](#)
- [VALLY\\_INLINE f32 vector3f\\_length \(vector3f vector\)](#)
- [VALLY\\_INLINE vector3f vector3f\\_normalized \(vector3f vector\)](#)
- [VALLY\\_INLINE void vector3f\\_normalize \(vector3f \\*vector\)](#)
- [VALLY\\_INLINE f32 vector3f\\_distance \(vector3f a, vector3f b\)](#)
- [VALLY\\_INLINE vector2i vector2i\\_add \(vector2i a, vector2i b\)](#)
- [VALLY\\_INLINE vector3i vector3i\\_add \(vector3i a, vector3i b\)](#)
- [VALLY\\_INLINE vector2i vector2i\\_sub \(vector2i a, vector2i b\)](#)
- [VALLY\\_INLINE vector3i vector3i\\_sub \(vector3i a, vector3i b\)](#)

### 6.19.1 Macro Definition Documentation

#### 6.19.1.1 LERP

```
#define LERP(
    w,
    v1,
    v2 ) ((1.0 - (w)) * (v1) + (w) * (v2))
```

#### 6.19.1.2 V\_2PI

```
#define V_2PI V_PI * 2.0F
```

#### 6.19.1.3 V\_PI

```
#define V_PI 3.141592653589793F
```

#### 6.19.1.4 V\_PI\_2

```
#define V_PI_2 V_PI / 2.0F
```

#### 6.19.1.5 V\_PI\_3

```
#define V_PI_3 V_PI / 3.0F
```

#### 6.19.1.6 V\_PI\_4

```
#define V_PI_4 V_PI / 4.0F
```

#### 6.19.1.7 V\_SQRT2

```
#define V_SQRT2 1.41421356237F
```

#### 6.19.1.8 vector2f\_create

```
#define vector2f_create(  
    x,  
    y ) ((vector2f){x, y})
```

#### 6.19.1.9 vector2f\_down

```
#define vector2f_down vector2f_create(0.0f, -1.0f)
```

#### 6.19.1.10 vector2f\_left

```
#define vector2f_left vector2f_create(-1.0f, 0.0f)
```

#### 6.19.1.11 vector2f\_one

```
#define vector2f_one vector2f_create(1.0f, 1.0f)
```

#### 6.19.1.12 vector2f\_right

```
#define vector2f_right vector2f_create(1.0f, 0.0f)
```

#### 6.19.1.13 vector2f\_up

```
#define vector2f_up vector2f_create(0.0f, 1.0f)
```

#### 6.19.1.14 vector2f\_zero

```
#define vector2f_zero vector2f_create(0.0f, 0.0f)
```

#### 6.19.1.15 vector2i\_create

```
#define vector2i_create(  
    x,  
    y ) ((vector2i){x, y})
```

### 6.19.1.16 vector3f\_create

```
#define vector3f_create(  
    x,  
    y,  
    z ) ((vector3f){x, y, z})
```

### 6.19.1.17 vector3i\_create

```
#define vector3i_create(  
    x,  
    y,  
    z ) ((vector3i){x, y, z})
```

## 6.19.2 Function Documentation

### 6.19.2.1 vabs()

```
VALLY_API f32 vabs (  
    f32 x )
```

### 6.19.2.2 vacos()

```
VALLY_API f32 vacos (  
    f32 x )
```

### 6.19.2.3 vasin()

```
VALLY_API f32 vasin (  
    f32 x )
```

### 6.19.2.4 vatan()

```
VALLY_API f32 vatan (  
    f32 x )
```

#### 6.19.2.5 vcos()

```
VALLY_API f32 vcos (
    f32 x )
```

#### 6.19.2.6 vector2f\_add()

```
VALLY_INLINE vector2f vector2f_add (
    vector2f a,
    vector2f b )
```

#### 6.19.2.7 vector2f\_distance()

```
VALLY_INLINE f32 vector2f_distance (
    vector2f a,
    vector2f b )
```

#### 6.19.2.8 vector2f\_div()

```
VALLY_INLINE vector2f vector2f_div (
    vector2f a,
    vector2f b )
```

#### 6.19.2.9 vector2f\_divs()

```
VALLY_INLINE vector2f vector2f_divs (
    vector2f vector,
    f32 scalar )
```

#### 6.19.2.10 vector2f\_length()

```
VALLY_INLINE f32 vector2f_length (
    vector2f vector )
```

#### 6.19.2.11 vector2f\_mul()

```
VALLY_INLINE vector2f vector2f_mul (
    vector2f a,
    vector2f b )
```

#### 6.19.2.12 vector2f\_muls()

```
VALLY_INLINE vector2f vector2f_muls (
    vector2f vector,
    f32 scalar )
```

#### 6.19.2.13 vector2f\_normalize()

```
VALLY_INLINE void vector2f_normalize (
    vector2f * vector )
```

#### 6.19.2.14 vector2f\_normalized()

```
VALLY_INLINE vector2f vector2f_normalized (
    vector2f vector )
```

#### 6.19.2.15 vector2f\_sub()

```
VALLY_INLINE vector2f vector2f_sub (
    vector2f a,
    vector2f b )
```

#### 6.19.2.16 vector2i\_add()

```
VALLY_INLINE vector2i vector2i_add (
    vector2i a,
    vector2i b )
```

#### 6.19.2.17 vector2i\_sub()

```
VALLY_INLINE vector2i vector2i_sub (
    vector2i a,
    vector2i b )
```

#### 6.19.2.18 vector3f\_add()

```
VALLY_INLINE vector3f vector3f_add (
    vector3f a,
    vector3f b )
```

#### 6.19.2.19 vector3f\_cross()

```
VALLY_INLINE vector3f vector3f_cross (
    vector3f a,
    vector3f b )
```

#### 6.19.2.20 vector3f\_distance()

```
VALLY_INLINE f32 vector3f_distance (
    vector3f a,
    vector3f b )
```

#### 6.19.2.21 vector3f\_div()

```
VALLY_INLINE vector3f vector3f_div (
    vector3f a,
    vector3f b )
```

#### 6.19.2.22 vector3f\_divs()

```
VALLY_INLINE vector3f vector3f_divs (
    vector3f vector,
    f32 scalar )
```

**6.19.2.23 vector3f\_dot()**

```
VALLY_INLINE f32 vector3f_dot (
    vector3f a,
    vector3f b )
```

**6.19.2.24 vector3f\_length()**

```
VALLY_INLINE f32 vector3f_length (
    vector3f vector )
```

**6.19.2.25 vector3f\_mul()**

```
VALLY_INLINE vector3f vector3f_mul (
    vector3f a,
    vector3f b )
```

**6.19.2.26 vector3f\_muls()**

```
VALLY_INLINE vector3f vector3f_muls (
    vector3f vector,
    f32 scalar )
```

**6.19.2.27 vector3f\_normalize()**

```
VALLY_INLINE void vector3f_normalize (
    vector3f * vector )
```

**6.19.2.28 vector3f\_normalized()**

```
VALLY_INLINE vector3f vector3f_normalized (
    vector3f vector )
```



### 6.19.2.29 vector3f\_sub()

```
VALLY_INLINE vector3f vector3f_sub (
    vector3f a,
    vector3f b )
```

### 6.19.2.30 vector3i\_add()

```
VALLY_INLINE vector3i vector3i_add (
    vector3i a,
    vector3i b )
```

### 6.19.2.31 vector3i\_sub()

```
VALLY_INLINE vector3i vector3i_sub (
    vector3i a,
    vector3i b )
```

### 6.19.2.32 vsin()

```
VALLY_API f32 vsin (
    f32 x )
```

### 6.19.2.33 vsqrt()

```
VALLY_API f32 vsqrt (
    f32 x )
```

### 6.19.2.34 vtan()

```
VALLY_API f32 vtan (
    f32 x )
```

## 6.20 math.h

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

```

1 #ifndef VALLY_MATH_VMATH_H_
2 #define VALLY_MATH_VMATH_H_
3
4 #include <vally/config.h>
5 #include <vally/math/math_types.h>
6
7 #define V_PI 3.141592653589793F
8 #define V_2PI V_PI * 2.0F
9 #define V_PI_2 V_PI / 2.0F
10 #define V_PI_3 V_PI / 3.0F
11 #define V_PI_4 V_PI / 4.0F
12 #define V_SQRT2 1.41421356237F
13
14 #define LERP(w, v1, v2) ((1.0 - (w)) * (v1) + (w) * (v2))
15
16 #define vector2f_create(x, y) ((vector2f){x, y})
17 #define vector3f_create(x, y, z) ((vector3f){x, y, z})
18 #define vector2i_create(x, y) ((vector2i){x, y})
19 #define vector3i_create(x, y, z) ((vector3i){x, y, z})
20
21 #define vector2f_zero vector2f_create(0.0f, 0.0f)
22 #define vector2f_one vector2f_create(1.0f, 1.0f)
23 #define vector2f_right vector2f_create(1.0f, 0.0f)
24 #define vector2f_left vector2f_create(-1.0f, 0.0f)
25 #define vector2f_up vector2f_create(0.0f, 1.0f)
26 #define vector2f_down vector2f_create(0.0f, -1.0f)
27
28 VALLY_API f32 vsin(f32 x);
29 VALLY_API f32 vcos(f32 x);
30 VALLY_API f32 vtan(f32 x);
31 VALLY_API f32 vasin(f32 x);
32 VALLY_API f32 vacos(f32 x);
33 VALLY_API f32 vatan(f32 x);
34 VALLY_API f32 vsqrt(f32 x);
35 VALLY_API f32 vabs(f32 x);
36
37 /* vector2f */
38 VALLY_INLINE vector2f vector2f_add(vector2f a, vector2f b) {
39     return vector2f_create(a.x + b.x, a.y + b.y);
40 }
41
42 VALLY_INLINE vector2f vector2f_sub(vector2f a, vector2f b) {
43     return vector2f_create(a.x - b.x, a.y - b.y);
44 }
45
46 VALLY_INLINE vector2f vector2f_mul(vector2f a, vector2f b) {
47     return vector2f_create(a.x * b.x, a.y * b.y);
48 }
49
50 VALLY_INLINE vector2f vector2f_muls(vector2f vector, f32 scalar) {
51     return vector2f_create(vector.x * scalar, vector.y * scalar);
52 }
53
54 VALLY_INLINE vector2f vector2f_divs(vector2f vector, f32 scalar) {
55     return vector2f_create(vector.x / scalar, vector.y / scalar);
56 }
57
58 VALLY_INLINE vector2f vector2f_div(vector2f a, vector2f b) {
59     return vector2f_create(a.x / b.x, a.y / b.y);
60 }
61
62 VALLY_INLINE f32 vector2f_length(vector2f vector) {
63     return vsqrt(vector.x * vector.x + vector.y * vector.y);
64 }
65
66 VALLY_INLINE vector2f vector2f_normalized(vector2f vector) {
67     const f32 length = vector2f_length(vector);
68     if (length != 0.0f) {
69         return vector2f_divs(vector, vector2f_length(vector));
70     } else {
71         return vector;
72     }
73 }
74
75 VALLY_INLINE void vector2f_normalize(vector2f *vector) {
76     const f32 length = vector2f_length(*vector);
77     if (length != 0.0f) {
78         vector->x /= length;
79         vector->y /= length;
80     }
81 }
82

```

```

83 VALLY_INLINE f32 vector2f_distance(vector2f a, vector2f b) {
84     return vector2f_length(vector2f_sub(a, b));
85 }
86
87 /* vector3f */
88 VALLY_INLINE vector3f vector3f_add(vector3f a, vector3f b) {
89     return vector3f_create(a.x + b.x, a.y + b.y, a.z + b.z);
90 }
91
92 VALLY_INLINE vector3f vector3f_sub(vector3f a, vector3f b) {
93     return vector3f_create(a.x - b.x, a.y - b.y, a.z - b.z);
94 }
95
96 VALLY_INLINE vector3f vector3f_cross(vector3f a, vector3f b) {
97     return vector3f_create((a.y * b.z - a.z * b.y), (a.z * b.x - a.x * b.z), (a.x * b.y - a.y * b.x));
98 }
99
100 VALLY_INLINE f32 vector3f_dot(vector3f a, vector3f b) {
101     return a.x * b.x + a.y * b.y + a.z * b.z;
102 }
103
104 VALLY_INLINE vector3f vector3f_mul(vector3f a, vector3f b) {
105     return vector3f_create(a.x * b.x, a.y * b.y, a.z * b.z);
106 }
107
108 VALLY_INLINE vector3f vector3f_div(vector3f a, vector3f b) {
109     return vector3f_create(a.x / b.x, a.y / b.y, a.z / b.z);
110 }
111
112 VALLY_INLINE vector3f vector3f_muls(vector3f vector, f32 scalar) {
113     return vector3f_create(vector.x * scalar, vector.y * scalar, vector.z * scalar);
114 }
115
116 VALLY_INLINE vector3f vector3f_divs(vector3f vector, f32 scalar) {
117     return vector3f_create(vector.x / scalar, vector.y / scalar, vector.z / scalar);
118 }
119
120 VALLY_INLINE f32 vector3f_length(vector3f vector) {
121     return vsqrt(vector.x * vector.x + vector.y * vector.y + vector.z * vector.z);
122 }
123
124 VALLY_INLINE vector3f vector3f_normalized(vector3f vector) {
125     return vector3f_divs(vector, vector3f_length(vector));
126 }
127
128 VALLY_INLINE void vector3f_normalize(vector3f *vector) {
129     *vector = vector3f_normalized(*vector);
130 }
131
132 VALLY_INLINE f32 vector3f_distance(vector3f a, vector3f b) {
133     return vector3f_length(vector3f_sub(a, b));
134 }
135
136 VALLY_INLINE vector2i vector2i_add(vector2i a, vector2i b) {
137     return vector2i_create(a.x + b.x, a.y + b.y);
138 }
139
140 VALLY_INLINE vector3i vector3i_add(vector3i a, vector3i b) {
141     return vector3i_create(a.x + b.x, a.y + b.y, a.z + b.z);
142 }
143
144 VALLY_INLINE vector2i vector2i_sub(vector2i a, vector2i b) {
145     return vector2i_create(a.x - b.x, a.y - b.y);
146 }
147
148 VALLY_INLINE vector3i vector3i_sub(vector3i a, vector3i b) {
149     return vector3i_create(a.x - b.x, a.y - b.y, a.z - b.z);
150 }
151
152 #endif // !VALLY_MATH_VMATH_H_

```

## 6.21 include/vally/math/math\_types.h File Reference

```
#include <vally/config.h>
```

### Data Structures

- struct [vector2f](#)

- struct [vector3f](#)
- struct [vector2i](#)
- struct [vector3i](#)

## 6.22 math\_types.h

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

```
1 #ifndef VALLY_MATH_VMATH_TYPES_H_
2 #define VALLY_MATH_VMATH_TYPES_H_
3
4 #include <vally/config.h>
5
6 typedef struct {
7     f32 x;
8     f32 y;
9 } vector2f;
10
11 typedef struct {
12     f32 x;
13     f32 y;
14     f32 z;
15 } vector3f;
16
17 typedef struct {
18     i32 x;
19     i32 y;
20 } vector2i;
21
22 typedef struct {
23     i32 x;
24     i32 y;
25     i32 z;
26 } vector3i;
27
28 #endif // !VALLY_MATH_VMATH_TYPES_H_
```

## 6.23 include/vally/renderer/texture.h File Reference

```
#include <vally/config.h>
#include <vally/resources/image.h>
```

### Data Structures

- struct [texture](#)  
*Texture struct.*

### Functions

- [VALLY\\_API texture \\* texture\\_create \(image \\*image\)](#)  
*Creates new texture from image.*
- [texture \\* texture\\_white\\_create \(\)](#)
- void [texture\\_destroy \(texture \\*texture\)](#)
- void [texture\\_bind \(u32 unit, texture \\*texture\)](#)
- void [texture\\_unbind \(\)](#)

## 6.23.1 Function Documentation

### 6.23.1.1 texture\_bind()

```
void texture_bind (
    u32 unit,
    texture * texture )
```

### 6.23.1.2 texture\_create()

```
VALLY_API texture * texture_create (
    image * image )
```

Creates new texture from image.

#### Parameters

<i>img</i>	image source
------------	--------------

#### Returns

textureS

### 6.23.1.3 texture\_destroy()

```
void texture_destroy (
    texture * texture )
```

### 6.23.1.4 texture\_unbind()

```
void texture_unbind ( )
```

### 6.23.1.5 texture\_white\_create()

```
texture * texture_white_create ( )
```

## 6.24 texture.h

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

```

1 //=====
2 // Vally 0.1indev
3 //-----
4 // Copyright (c) 2022 Dmytro Zykov
5
6 // Permission is hereby granted, free of charge, to any person obtaining a copy
7 // of this software and associated documentation files (the "Software"), to deal
8 // in the Software without restriction, including without limitation the rights
9 // to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
10 // copies of the Software, and to permit persons to whom the Software is
11 // furnished to do so, subject to the following conditions:
12
13 // The above copyright notice and this permission notice shall be included in all
14 // copies or substantial portions of the Software.
15
16 // THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
17 // IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
18 // FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
19 // AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
20 // LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
21 // OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
22 // SOFTWARE.
23 //=====
24
25 #pragma once
26
27 #include <vally/config.h>
28 #include <vally/resources/image.h>
29
30 typedef struct {
31     u32 id;
32     i32 width;
33     i32 height;
34 } texture;
35
36 VALLY_API texture *texture_create(image *image);
37
38 // Creates white texture 1x1.
39 // Needed for colored quads "without texture".
40 texture *texture_white_create();
41
42 // Simple destructor
43 void texture_destroy(texture *texture);
44
45 // Binds OpenGL texture to given texture slot
46 void texture_bind(u32 unit, texture *texture);
47
48 // Unbinds texture (Actually, it's not used, but maybe needed in the future)
49 void texture_unbind();

```

## 6.25 include/vally/resources/image.h File Reference

```

#include <vally/resources/resource_types.h>
#include <vally/config.h>

```

### Functions

- [VALLY\\_API image \\* image\\_load](#) (const char \*path)
- void [image\\_unload](#) (image \*img)

### 6.25.1 Function Documentation

### 6.25.1.1 image\_load()

```
VALLY_API image * image_load (
    const char * path )
```

### 6.25.1.2 image\_unload()

```
void image_unload (
    image * img )
```

## 6.26 image.h

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

```
1 #ifndef VALLY_RESOURCES_IMAGE_H_
2 #define VALLY_RESOURCES_IMAGE_H_
3
4 #include <vally/resources/resource_types.h>
5 #include <vally/config.h>
6
7 VALLY_API image *image_load(const char *path);
8
9 void image_unload(image *img);
10
11
12 #endif // !VALLY_RESOURCES_IMAGE_H_
```

## 6.27 include/vally/resources/resource\_types.h File Reference

```
#include <vally/config.h>
```

### Data Structures

- struct [image](#)
- struct [resource](#)

### Typedefs

- typedef void(\* [destructor\\_fun](#)) (void \*)

### 6.27.1 Typedef Documentation

#### 6.27.1.1 destructor\_fun

```
typedef void(* destructor_fun) (void *)
```

## 6.28 resource\_types.h

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

```
1 #ifndef VALLY_RESOURCES_RECOURCE_TYPES_H_
2 #define VALLY_RESOURCES_RECOURCE_TYPES_H_
3
4 #include <vally/config.h>
5
6 typedef void (*destructor_fun)(void *);
7
8 typedef struct {
9     i8 *buffer;
10    i32 width;
11    i32 height;
12    i32 channels;
13 } image;
14
15 typedef struct {
16     void *data;
17     destructor_fun destructor;
18 } resource;
19
20 #endif // !VALLY_RESOURCES_RECOURCE_TYPES_H_
```

## 6.29 include/vally/vally.h File Reference

```
#include <vally/core/engine.h>
#include <vally/core/input.h>
#include <vally/ecs/ecs.h>
```

## 6.30 vally.h

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

```
1 /*****
2  * vally.h
3  *
4  * Copyright (c) 2022 Dmytro Zykov
5  *
6  * This file is a part of the vally project, and may only be used,
7  * modified and distributed under the terms of the MIT License,
8  * LICENSE.md. By continuing to use, modify and distribute this file
9  * you indicate that you have read the license and accept it fully.
10 *****/
11
12 #ifndef VALLY_H_
13 #define VALLY_H_
14
15 #include <vally/core/engine.h>
16 #include <vally/core/input.h>
17 #include <vally/ecs/ecs.h>
18
19 #endif // VALLY_H_
```



# Index

- animation, [27](#)
  - animator, [28](#)
  - first\_frame, [27](#)
  - frametime, [27](#)
  - last\_frame, [27](#)
  - texture, [27](#)
  - tiling, [28](#)
- animation\_create
  - animator.h, [60](#)
- animator, [28](#)
  - animation, [28](#)
  - current\_frame, [28](#)
  - playing, [28](#)
  - time, [29](#)
- animator.h
  - animation\_create, [60](#)
  - ecs\_animator\_add, [60](#)
  - ecs\_animator\_play, [60](#)
  - ecs\_animator\_stop, [61](#)
  - ecs\_animator\_update, [61](#)
- atlas, [29](#)
  - sprite\_count, [29](#)
  - spriterenderer, [34](#)
  - sprites\_per\_row, [29](#)
  - tiling, [29](#)
  - uvs, [30](#)
- b32
  - config.h, [43](#)
- b8
  - config.h, [43](#)
- BIT\_MASK
  - config.h, [42](#)
- buffer
  - image, [33](#)
- c
  - event\_context, [30](#)
- channels
  - image, [33](#)
- component\_mask
  - ecs\_types.h, [67](#)
- config.h
  - b32, [43](#)
  - b8, [43](#)
  - BIT\_MASK, [42](#)
  - f32, [43](#)
  - f64, [43](#)
  - FALSE, [42](#)
  - i16, [44](#)
  - i32, [44](#)
  - i64, [44](#)
  - i8, [44](#)
  - MAX, [42](#)
  - MIN, [42](#)
  - NULL, [42](#)
  - TRUE, [43](#)
  - u16, [44](#)
  - u32, [44](#)
  - u64, [44](#)
  - u8, [45](#)
  - VALLY\_API, [43](#)
- current\_frame
  - animator, [28](#)
- data
  - event\_context, [31](#)
  - resource, [33](#)
- destructor
  - resource, [34](#)
- destructor\_fun
  - resource\_types.h, [83](#)
- ecs.h
  - ecs\_component\_add, [65](#)
  - ecs\_component\_has, [66](#)
  - ecs\_component\_remove, [66](#)
  - ecs\_entity\_create, [66](#)
  - ecs\_get\_last, [66](#)
  - ecs\_init, [66](#)
  - ecs\_is\_alive, [65](#)
  - ECS\_MAX\_COMPONENTS, [65](#)
  - ECS\_MAX\_ENTITIES, [65](#)
  - ECS\_NULL\_ENTITY, [65](#)
- ECS\_ALIVE
  - ecs\_types.h, [68](#)
- ecs\_animator\_add
  - animator.h, [60](#)
- ECS\_ANIMATOR\_COMPONENT
  - ecs\_types.h, [68](#)
- ecs\_animator\_play
  - animator.h, [60](#)
- ecs\_animator\_stop
  - animator.h, [61](#)
- ecs\_animator\_update
  - animator.h, [61](#)
- ecs\_component\_add
  - ecs.h, [65](#)
- ecs\_component\_has
  - ecs.h, [66](#)

- ecs\_component\_remove
  - ecs.h, 66
- ecs\_entity\_create
  - ecs.h, 66
- ecs\_get\_last
  - ecs.h, 66
- ecs\_init
  - ecs.h, 66
- ecs\_is\_alive
  - ecs.h, 65
- ECS\_MAX\_COMPONENTS
  - ecs.h, 65
- ECS\_MAX\_ENTITIES
  - ecs.h, 65
- ECS\_NULL\_ENTITY
  - ecs.h, 65
- ECS\_RIGIDBODY\_COMPONENT
  - ecs\_types.h, 68
- ecs\_spriterenderer\_add
  - spriterenderer.h, 62
- ecs\_spriterenderer\_add\_from\_atlas
  - spriterenderer.h, 62
- ecs\_spriterenderer\_atlas\_select
  - spriterenderer.h, 62
- ECS\_SPRITERENDERER\_COMPONENT
  - ecs\_types.h, 68
- ecs\_spriterenderer\_get
  - spriterenderer.h, 62
- ecs\_spriterenderer\_update
  - spriterenderer.h, 63
- ecs\_transform\_add
  - transform.h, 64
- ECS\_TRANSFORM\_COMPONENT
  - ecs\_types.h, 68
- ecs\_transform\_get
  - transform.h, 64
- ecs\_types.h
  - component\_mask, 67
  - ECS\_ALIVE, 68
  - ECS\_ANIMATOR\_COMPONENT, 68
  - ECS\_RIGIDBODY\_COMPONENT, 68
  - ECS\_SPRITERENDERER\_COMPONENT, 68
  - ECS\_TRANSFORM\_COMPONENT, 68
  - entity, 67
- engine.h
  - engine\_create, 47
  - engine\_render, 47
  - engine\_run, 48
  - engine\_start, 47
  - engine\_update, 47
- engine\_create
  - engine.h, 47
- engine\_event\_code
  - event.h, 50
- engine\_render
  - engine.h, 47
- engine\_run
  - engine.h, 48
- engine\_start
  - engine.h, 47
- engine\_update
  - engine.h, 47
- entity
  - ecs\_types.h, 67
  - spriterenderer, 34
  - transform, 36
- event.h
  - engine\_event\_code, 50
  - EVENT\_CODE\_APP\_QUIT, 50
  - EVENT\_CODE\_CAMERA\_MOVED, 50
  - EVENT\_CODE\_KEY\_PRESSED, 50
  - EVENT\_CODE\_KEY\_RELEASED, 50
  - EVENT\_CODE\_MOUSE\_BUTTON\_PRESSED, 50
  - EVENT\_CODE\_MOUSE\_BUTTON\_RELEASED, 50
  - EVENT\_CODE\_MOUSE\_MOVED, 50
  - EVENT\_CODE\_MOUSE\_WHEEL, 50
  - EVENT\_CODE\_WINDOW\_RESIZED, 50
  - event\_init, 50
  - event\_publish, 50
  - event\_subscribe, 51
  - event\_terminate, 51
  - event\_unsubscribe, 51
  - MAX\_EVENT\_CODE, 50
  - on\_event\_fun, 49
- EVENT\_CODE\_APP\_QUIT
  - event.h, 50
- EVENT\_CODE\_CAMERA\_MOVED
  - event.h, 50
- EVENT\_CODE\_KEY\_PRESSED
  - event.h, 50
- EVENT\_CODE\_KEY\_RELEASED
  - event.h, 50
- EVENT\_CODE\_MOUSE\_BUTTON\_PRESSED
  - event.h, 50
- EVENT\_CODE\_MOUSE\_BUTTON\_RELEASED
  - event.h, 50
- EVENT\_CODE\_MOUSE\_MOVED
  - event.h, 50
- EVENT\_CODE\_MOUSE\_WHEEL
  - event.h, 50
- EVENT\_CODE\_WINDOW\_RESIZED
  - event.h, 50
- event\_context, 30
  - c, 30
  - data, 31
  - f32, 31
  - f64, 31
  - i16, 31
  - i32, 31
  - i64, 31
  - i8, 31
  - u16, 32
  - u32, 32
  - u64, 32

- u8, [32](#)
- event\_init
  - event.h, [50](#)
- event\_publish
  - event.h, [50](#)
- event\_subscribe
  - event.h, [51](#)
- event\_terminate
  - event.h, [51](#)
- event\_unsubscribe
  - event.h, [51](#)
- f32
  - config.h, [43](#)
  - event\_context, [31](#)
- f64
  - config.h, [43](#)
  - event\_context, [31](#)
- FALSE
  - config.h, [42](#)
- first\_frame
  - animation, [27](#)
- frametime
  - animation, [27](#)
- height
  - image, [33](#)
  - texture, [35](#)
- i16
  - config.h, [44](#)
  - event\_context, [31](#)
- i32
  - config.h, [44](#)
  - event\_context, [31](#)
- i64
  - config.h, [44](#)
  - event\_context, [31](#)
- i8
  - config.h, [44](#)
  - event\_context, [31](#)
- id
  - texture, [36](#)
- image, [32](#)
  - buffer, [33](#)
  - channels, [33](#)
  - height, [33](#)
  - width, [33](#)
- image.h
  - image\_load, [82](#)
  - image\_unload, [83](#)
- image\_load
  - image.h, [82](#)
- image\_unload
  - image.h, [83](#)
- include/vally/config.h, [41](#), [45](#)
- include/vally/core/engine.h, [46](#), [48](#)
- include/vally/core/event.h, [49](#), [52](#)
- include/vally/core/input.h, [52](#), [58](#)
- include/vally/ecs/components/animatorm.h, [60](#), [61](#)
- include/vally/ecs/components/spriterenderer.h, [62](#), [63](#)
- include/vally/ecs/components/transform.h, [63](#), [64](#)
- include/vally/ecs/ecs.h, [64](#), [67](#)
- include/vally/ecs/ecs\_types.h, [67](#), [68](#)
- include/vally/math/math.h, [68](#), [78](#)
- include/vally/math/math\_types.h, [79](#), [80](#)
- include/vally/renderer/texture.h, [80](#), [82](#)
- include/vally/resources/image.h, [82](#), [83](#)
- include/vally/resources/resource\_types.h, [83](#), [84](#)
- include/vally/vally.h, [84](#)
- input.h
  - input\_init, [55](#)
  - input\_key\_pressed, [55](#)
  - input\_key\_released, [56](#)
  - input\_mouse\_button\_pressed, [56](#)
  - input\_mouse\_button\_released, [57](#)
  - input\_mouse\_position, [57](#)
  - input\_mouse\_prev\_position, [57](#)
  - input\_terminate, [57](#)
- input\_init
  - input.h, [55](#)
- input\_key\_pressed
  - input.h, [55](#)
- input\_key\_released
  - input.h, [56](#)
- input\_mouse\_button\_pressed
  - input.h, [56](#)
- input\_mouse\_button\_released
  - input.h, [57](#)
- input\_mouse\_position
  - input.h, [57](#)
- input\_mouse\_prev\_position
  - input.h, [57](#)
- input\_terminate
  - input.h, [57](#)
- KEY\_0
  - Keyboard keys, [9](#)
- KEY\_1
  - Keyboard keys, [9](#)
- KEY\_2
  - Keyboard keys, [9](#)
- KEY\_3
  - Keyboard keys, [10](#)
- KEY\_4
  - Keyboard keys, [10](#)
- KEY\_5
  - Keyboard keys, [10](#)
- KEY\_6
  - Keyboard keys, [10](#)
- KEY\_7
  - Keyboard keys, [10](#)
- KEY\_8
  - Keyboard keys, [10](#)
- KEY\_9
  - Keyboard keys, [10](#)
- KEY\_A
  - Keyboard keys, [10](#)

KEY\_APOSTROPHE  
    Keyboard keys, [11](#)  
KEY\_B  
    Keyboard keys, [11](#)  
KEY\_BACKSLASH  
    Keyboard keys, [11](#)  
KEY\_BACKSPACE  
    Keyboard keys, [11](#)  
KEY\_C  
    Keyboard keys, [11](#)  
KEY\_CAPS\_LOCK  
    Keyboard keys, [11](#)  
KEY\_COMMA  
    Keyboard keys, [11](#)  
KEY\_D  
    Keyboard keys, [11](#)  
KEY\_DELETE  
    Keyboard keys, [12](#)  
KEY\_DOWN  
    Keyboard keys, [12](#)  
KEY\_E  
    Keyboard keys, [12](#)  
KEY\_END  
    Keyboard keys, [12](#)  
KEY\_ENTER  
    Keyboard keys, [12](#)  
KEY\_EQUAL  
    Keyboard keys, [12](#)  
KEY\_ESCAPE  
    Keyboard keys, [12](#)  
KEY\_F  
    Keyboard keys, [12](#)  
KEY\_F1  
    Keyboard keys, [13](#)  
KEY\_F10  
    Keyboard keys, [13](#)  
KEY\_F11  
    Keyboard keys, [13](#)  
KEY\_F12  
    Keyboard keys, [13](#)  
KEY\_F13  
    Keyboard keys, [13](#)  
KEY\_F14  
    Keyboard keys, [13](#)  
KEY\_F15  
    Keyboard keys, [13](#)  
KEY\_F16  
    Keyboard keys, [13](#)  
KEY\_F17  
    Keyboard keys, [14](#)  
KEY\_F18  
    Keyboard keys, [14](#)  
KEY\_F19  
    Keyboard keys, [14](#)  
KEY\_F2  
    Keyboard keys, [14](#)  
KEY\_F20  
    Keyboard keys, [14](#)  
KEY\_F21  
    Keyboard keys, [14](#)  
KEY\_F22  
    Keyboard keys, [14](#)  
KEY\_F23  
    Keyboard keys, [14](#)  
KEY\_F24  
    Keyboard keys, [15](#)  
KEY\_F25  
    Keyboard keys, [15](#)  
KEY\_F3  
    Keyboard keys, [15](#)  
KEY\_F4  
    Keyboard keys, [15](#)  
KEY\_F5  
    Keyboard keys, [15](#)  
KEY\_F6  
    Keyboard keys, [15](#)  
KEY\_F7  
    Keyboard keys, [15](#)  
KEY\_F8  
    Keyboard keys, [15](#)  
KEY\_F9  
    Keyboard keys, [16](#)  
KEY\_G  
    Keyboard keys, [16](#)  
KEY\_GRAVE\_ACCENT  
    Keyboard keys, [16](#)  
KEY\_H  
    Keyboard keys, [16](#)  
KEY\_HOME  
    Keyboard keys, [16](#)  
KEY\_I  
    Keyboard keys, [16](#)  
KEY\_INSERT  
    Keyboard keys, [16](#)  
KEY\_J  
    Keyboard keys, [16](#)  
KEY\_K  
    Keyboard keys, [17](#)  
KEY\_KP\_0  
    Keyboard keys, [17](#)  
KEY\_KP\_1  
    Keyboard keys, [17](#)  
KEY\_KP\_2  
    Keyboard keys, [17](#)  
KEY\_KP\_3  
    Keyboard keys, [17](#)  
KEY\_KP\_4  
    Keyboard keys, [17](#)  
KEY\_KP\_5  
    Keyboard keys, [17](#)  
KEY\_KP\_6  
    Keyboard keys, [17](#)  
KEY\_KP\_7  
    Keyboard keys, [18](#)  
KEY\_KP\_8  
    Keyboard keys, [18](#)

- KEY\_KP\_9
  - Keyboard keys, [18](#)
- KEY\_KP\_ADD
  - Keyboard keys, [18](#)
- KEY\_KP\_DECIMAL
  - Keyboard keys, [18](#)
- KEY\_KP\_DIVIDE
  - Keyboard keys, [18](#)
- KEY\_KP\_ENTER
  - Keyboard keys, [18](#)
- KEY\_KP\_EQUAL
  - Keyboard keys, [18](#)
- KEY\_KP\_MULTIPLY
  - Keyboard keys, [19](#)
- KEY\_KP\_SUBTRACT
  - Keyboard keys, [19](#)
- KEY\_L
  - Keyboard keys, [19](#)
- KEY\_LEFT
  - Keyboard keys, [19](#)
- KEY\_LEFT\_ALT
  - Keyboard keys, [19](#)
- KEY\_LEFT\_BRACKET
  - Keyboard keys, [19](#)
- KEY\_LEFT\_CONTROL
  - Keyboard keys, [19](#)
- KEY\_LEFT\_SHIFT
  - Keyboard keys, [19](#)
- KEY\_LEFT\_SUPER
  - Keyboard keys, [20](#)
- KEY\_M
  - Keyboard keys, [20](#)
- KEY\_MENU
  - Keyboard keys, [20](#)
- KEY\_MINUS
  - Keyboard keys, [20](#)
- KEY\_N
  - Keyboard keys, [20](#)
- KEY\_NUM\_LOCK
  - Keyboard keys, [20](#)
- KEY\_O
  - Keyboard keys, [20](#)
- KEY\_P
  - Keyboard keys, [20](#)
- KEY\_PAGE\_DOWN
  - Keyboard keys, [21](#)
- KEY\_PAGE\_UP
  - Keyboard keys, [21](#)
- KEY\_PAUSE
  - Keyboard keys, [21](#)
- KEY\_PERIOD
  - Keyboard keys, [21](#)
- KEY\_PRINT\_SCREEN
  - Keyboard keys, [21](#)
- KEY\_Q
  - Keyboard keys, [21](#)
- KEY\_R
  - Keyboard keys, [21](#)
- KEY\_RIGHT
  - Keyboard keys, [21](#)
- KEY\_RIGHT\_ALT
  - Keyboard keys, [22](#)
- KEY\_RIGHT\_BRACKET
  - Keyboard keys, [22](#)
- KEY\_RIGHT\_CONTROL
  - Keyboard keys, [22](#)
- KEY\_RIGHT\_SHIFT
  - Keyboard keys, [22](#)
- KEY\_RIGHT\_SUPER
  - Keyboard keys, [22](#)
- KEY\_S
  - Keyboard keys, [22](#)
- KEY\_SCROLL\_LOCK
  - Keyboard keys, [22](#)
- KEY\_SEMICOLON
  - Keyboard keys, [22](#)
- KEY\_SLASH
  - Keyboard keys, [23](#)
- KEY\_SPACE
  - Keyboard keys, [23](#)
- KEY\_T
  - Keyboard keys, [23](#)
- KEY\_TAB
  - Keyboard keys, [23](#)
- KEY\_U
  - Keyboard keys, [23](#)
- KEY\_UP
  - Keyboard keys, [23](#)
- KEY\_V
  - Keyboard keys, [23](#)
- KEY\_W
  - Keyboard keys, [23](#)
- KEY\_WORLD\_1
  - Keyboard keys, [24](#)
- KEY\_WORLD\_2
  - Keyboard keys, [24](#)
- KEY\_X
  - Keyboard keys, [24](#)
- KEY\_Y
  - Keyboard keys, [24](#)
- KEY\_Z
  - Keyboard keys, [24](#)
- Keyboard keys, [7](#)
  - KEY\_0, [9](#)
  - KEY\_1, [9](#)
  - KEY\_2, [9](#)
  - KEY\_3, [10](#)
  - KEY\_4, [10](#)
  - KEY\_5, [10](#)
  - KEY\_6, [10](#)
  - KEY\_7, [10](#)
  - KEY\_8, [10](#)
  - KEY\_9, [10](#)
  - KEY\_A, [10](#)
  - KEY\_APOSTROPHE, [11](#)
  - KEY\_B, [11](#)

KEY\_BACKSLASH, 11  
KEY\_BACKSPACE, 11  
KEY\_C, 11  
KEY\_CAPS\_LOCK, 11  
KEY\_COMMA, 11  
KEY\_D, 11  
KEY\_DELETE, 12  
KEY\_DOWN, 12  
KEY\_E, 12  
KEY\_END, 12  
KEY\_ENTER, 12  
KEY\_EQUAL, 12  
KEY\_ESCAPE, 12  
KEY\_F, 12  
KEY\_F1, 13  
KEY\_F10, 13  
KEY\_F11, 13  
KEY\_F12, 13  
KEY\_F13, 13  
KEY\_F14, 13  
KEY\_F15, 13  
KEY\_F16, 13  
KEY\_F17, 14  
KEY\_F18, 14  
KEY\_F19, 14  
KEY\_F2, 14  
KEY\_F20, 14  
KEY\_F21, 14  
KEY\_F22, 14  
KEY\_F23, 14  
KEY\_F24, 15  
KEY\_F25, 15  
KEY\_F3, 15  
KEY\_F4, 15  
KEY\_F5, 15  
KEY\_F6, 15  
KEY\_F7, 15  
KEY\_F8, 15  
KEY\_F9, 16  
KEY\_G, 16  
KEY\_GRAVE\_ACCENT, 16  
KEY\_H, 16  
KEY\_HOME, 16  
KEY\_I, 16  
KEY\_INSERT, 16  
KEY\_J, 16  
KEY\_K, 17  
KEY\_KP\_0, 17  
KEY\_KP\_1, 17  
KEY\_KP\_2, 17  
KEY\_KP\_3, 17  
KEY\_KP\_4, 17  
KEY\_KP\_5, 17  
KEY\_KP\_6, 17  
KEY\_KP\_7, 18  
KEY\_KP\_8, 18  
KEY\_KP\_9, 18  
KEY\_KP\_ADD, 18  
KEY\_KP\_DECIMAL, 18  
KEY\_KP\_DIVIDE, 18  
KEY\_KP\_ENTER, 18  
KEY\_KP\_EQUAL, 18  
KEY\_KP\_MULTIPLY, 19  
KEY\_KP\_SUBTRACT, 19  
KEY\_L, 19  
KEY\_LEFT, 19  
KEY\_LEFT\_ALT, 19  
KEY\_LEFT\_BRACKET, 19  
KEY\_LEFT\_CONTROL, 19  
KEY\_LEFT\_SHIFT, 19  
KEY\_LEFT\_SUPER, 20  
KEY\_M, 20  
KEY\_MENU, 20  
KEY\_MINUS, 20  
KEY\_N, 20  
KEY\_NUM\_LOCK, 20  
KEY\_O, 20  
KEY\_P, 20  
KEY\_PAGE\_DOWN, 21  
KEY\_PAGE\_UP, 21  
KEY\_PAUSE, 21  
KEY\_PERIOD, 21  
KEY\_PRINT\_SCREEN, 21  
KEY\_Q, 21  
KEY\_R, 21  
KEY\_RIGHT, 21  
KEY\_RIGHT\_ALT, 22  
KEY\_RIGHT\_BRACKET, 22  
KEY\_RIGHT\_CONTROL, 22  
KEY\_RIGHT\_SHIFT, 22  
KEY\_RIGHT\_SUPER, 22  
KEY\_S, 22  
KEY\_SCROLL\_LOCK, 22  
KEY\_SEMICOLON, 22  
KEY\_SLASH, 23  
KEY\_SPACE, 23  
KEY\_T, 23  
KEY\_TAB, 23  
KEY\_U, 23  
KEY\_UP, 23  
KEY\_V, 23  
KEY\_W, 23  
KEY\_WORLD\_1, 24  
KEY\_WORLD\_2, 24  
KEY\_X, 24  
KEY\_Y, 24  
KEY\_Z, 24  
KEYS\_MAX\_KEYS, 24  
KEYS\_MAX\_KEYS  
    Keyboard keys, 24  
last\_frame  
    animation, 27  
LERP  
    math.h, 69  
math.h

- LERP, [69](#)
- V\_2PI, [69](#)
- V\_PI, [70](#)
- V\_PI\_2, [70](#)
- V\_PI\_3, [70](#)
- V\_PI\_4, [70](#)
- V\_SQRT2, [70](#)
- vabs, [72](#)
- vacos, [72](#)
- vasin, [72](#)
- vatan, [72](#)
- vcos, [72](#)
- vector2f\_add, [73](#)
- vector2f\_create, [70](#)
- vector2f\_distance, [73](#)
- vector2f\_div, [73](#)
- vector2f\_divs, [73](#)
- vector2f\_down, [70](#)
- vector2f\_left, [71](#)
- vector2f\_length, [73](#)
- vector2f\_mul, [73](#)
- vector2f\_muls, [74](#)
- vector2f\_normalize, [74](#)
- vector2f\_normalized, [74](#)
- vector2f\_one, [71](#)
- vector2f\_right, [71](#)
- vector2f\_sub, [74](#)
- vector2f\_up, [71](#)
- vector2f\_zero, [71](#)
- vector2i\_add, [74](#)
- vector2i\_create, [71](#)
- vector2i\_sub, [74](#)
- vector3f\_add, [75](#)
- vector3f\_create, [71](#)
- vector3f\_cross, [75](#)
- vector3f\_distance, [75](#)
- vector3f\_div, [75](#)
- vector3f\_divs, [75](#)
- vector3f\_dot, [75](#)
- vector3f\_length, [76](#)
- vector3f\_mul, [76](#)
- vector3f\_muls, [76](#)
- vector3f\_normalize, [76](#)
- vector3f\_normalized, [76](#)
- vector3f\_sub, [76](#)
- vector3i\_add, [77](#)
- vector3i\_create, [72](#)
- vector3i\_sub, [77](#)
- vsin, [77](#)
- vsqrt, [77](#)
- vtan, [77](#)
- MAX
  - config.h, [42](#)
- MAX\_EVENT\_CODE
  - event.h, [50](#)
- MIN
  - config.h, [42](#)
- Mouse buttons, [25](#)
- MOUSE\_BUTTON\_1, [25](#)
- MOUSE\_BUTTON\_2, [25](#)
- MOUSE\_BUTTON\_3, [25](#)
- MOUSE\_BUTTON\_4, [25](#)
- MOUSE\_BUTTON\_5, [25](#)
- MOUSE\_BUTTON\_6, [26](#)
- MOUSE\_BUTTON\_7, [26](#)
- MOUSE\_BUTTON\_8, [26](#)
- MOUSE\_BUTTON\_LEFT, [26](#)
- MOUSE\_BUTTON\_MAX, [26](#)
- MOUSE\_BUTTON\_MIDDLE, [26](#)
- MOUSE\_BUTTON\_RIGHT, [26](#)
- MOUSE\_BUTTON\_1
  - Mouse buttons, [25](#)
- MOUSE\_BUTTON\_2
  - Mouse buttons, [25](#)
- MOUSE\_BUTTON\_3
  - Mouse buttons, [25](#)
- MOUSE\_BUTTON\_4
  - Mouse buttons, [25](#)
- MOUSE\_BUTTON\_5
  - Mouse buttons, [25](#)
- MOUSE\_BUTTON\_6
  - Mouse buttons, [26](#)
- MOUSE\_BUTTON\_7
  - Mouse buttons, [26](#)
- MOUSE\_BUTTON\_8
  - Mouse buttons, [26](#)
- MOUSE\_BUTTON\_LEFT
  - Mouse buttons, [26](#)
- MOUSE\_BUTTON\_MAX
  - Mouse buttons, [26](#)
- MOUSE\_BUTTON\_MIDDLE
  - Mouse buttons, [26](#)
- MOUSE\_BUTTON\_RIGHT
  - Mouse buttons, [26](#)
- NULL
  - config.h, [42](#)
- on\_event\_fun
  - event.h, [49](#)
- playing
  - animator, [28](#)
- position
  - transform, [36](#)
- resource, [33](#)
  - data, [33](#)
  - destructor, [34](#)
- resource\_types.h
  - destructor\_fun, [83](#)
- scale
  - transform, [36](#)
- size
  - spriterenderer, [34](#)
- sprite\_count

- atlas, 29
- spriterenderer, 34
  - atlas, 34
  - entity, 34
  - size, 34
  - texture, 35
  - uvs, 35
- spriterenderer.h
  - ecs\_spriterenderer\_add, 62
  - ecs\_spriterenderer\_add\_from\_atlas, 62
  - ecs\_spriterenderer\_atlas\_select, 62
  - ecs\_spriterenderer\_get, 62
  - ecs\_spriterenderer\_update, 63
- sprites\_per\_row
  - atlas, 29
- texture, 35
  - animation, 27
  - height, 35
  - id, 36
  - spriterenderer, 35
  - width, 36
- texture.h
  - texture\_bind, 81
  - texture\_create, 81
  - texture\_destroy, 81
  - texture\_unbind, 81
  - texture\_white\_create, 81
- texture\_bind
  - texture.h, 81
- texture\_create
  - texture.h, 81
- texture\_destroy
  - texture.h, 81
- texture\_unbind
  - texture.h, 81
- texture\_white\_create
  - texture.h, 81
- tiling
  - animation, 28
  - atlas, 29
- time
  - animator, 29
- transform, 36
  - entity, 36
  - position, 36
  - scale, 36
- transform.h
  - ecs\_transform\_add, 64
  - ecs\_transform\_get, 64
- TRUE
  - config.h, 43
- u16
  - config.h, 44
  - event\_context, 32
- u32
  - config.h, 44
  - event\_context, 32
- u64
  - config.h, 44
  - event\_context, 32
- u8
  - config.h, 45
  - event\_context, 32
- uv
  - uvs, 37
- uvs, 37
  - atlas, 30
  - spriterenderer, 35
  - uv, 37
- V\_2PI
  - math.h, 69
- V\_PI
  - math.h, 70
- V\_PI\_2
  - math.h, 70
- V\_PI\_3
  - math.h, 70
- V\_PI\_4
  - math.h, 70
- V\_SQRT2
  - math.h, 70
- vabs
  - math.h, 72
- vacos
  - math.h, 72
- VALLY\_API
  - config.h, 43
- vasin
  - math.h, 72
- vatan
  - math.h, 72
- vcos
  - math.h, 72
- vector2f, 37
  - x, 37
  - y, 38
- vector2f\_add
  - math.h, 73
- vector2f\_create
  - math.h, 70
- vector2f\_distance
  - math.h, 73
- vector2f\_div
  - math.h, 73
- vector2f\_divs
  - math.h, 73
- vector2f\_down
  - math.h, 70
- vector2f\_left
  - math.h, 71
- vector2f\_length
  - math.h, 73
- vector2f\_mul
  - math.h, 73
- vector2f\_muls



- math.h, [74](#)
- vector2f\_normalize
  - math.h, [74](#)
- vector2f\_normalized
  - math.h, [74](#)
- vector2f\_one
  - math.h, [71](#)
- vector2f\_right
  - math.h, [71](#)
- vector2f\_sub
  - math.h, [74](#)
- vector2f\_up
  - math.h, [71](#)
- vector2f\_zero
  - math.h, [71](#)
- vector2i, [38](#)
  - x, [38](#)
  - y, [38](#)
- vector2i\_add
  - math.h, [74](#)
- vector2i\_create
  - math.h, [71](#)
- vector2i\_sub
  - math.h, [74](#)
- vector3f, [38](#)
  - x, [39](#)
  - y, [39](#)
  - z, [39](#)
- vector3f\_add
  - math.h, [75](#)
- vector3f\_create
  - math.h, [71](#)
- vector3f\_cross
  - math.h, [75](#)
- vector3f\_distance
  - math.h, [75](#)
- vector3f\_div
  - math.h, [75](#)
- vector3f\_divs
  - math.h, [75](#)
- vector3f\_dot
  - math.h, [75](#)
- vector3f\_length
  - math.h, [76](#)
- vector3f\_mul
  - math.h, [76](#)
- vector3f\_muls
  - math.h, [76](#)
- vector3f\_normalize
  - math.h, [76](#)
- vector3f\_normalized
  - math.h, [76](#)
- vector3f\_sub
  - math.h, [76](#)
- vector3i, [39](#)
  - x, [39](#)
  - y, [40](#)
  - z, [40](#)
- vector3i\_add
  - math.h, [77](#)
- vector3i\_create
  - math.h, [72](#)
- vector3i\_sub
  - math.h, [77](#)
- vsin
  - math.h, [77](#)
- vsqrt
  - math.h, [77](#)
- vtan
  - math.h, [77](#)
- width
  - image, [33](#)
  - texture, [36](#)
- x
  - vector2f, [37](#)
  - vector2i, [38](#)
  - vector3f, [39](#)
  - vector3i, [39](#)
- y
  - vector2f, [38](#)
  - vector2i, [38](#)
  - vector3f, [39](#)
  - vector3i, [40](#)
- z
  - vector3f, [39](#)
  - vector3i, [40](#)