User Interface

event

value	simple_direct_media_layer	unix_terminal	win32_console	x_window_system	win32_display	java	java_script	google_web_toolkit	description
keyboard	SDL_Keyboard	(interactive and mouse sequence)	KEY_EVENT			KEY_EVENT_MASK			
keyboard_press			WIN_KEYDOWN	XCB_KEY_PRESS			onkeydown	KeyDown	
keyboard_release			WIN_KEYUP	XCB_KEY_RELEASE			onkeyup	KeyUp	
							onkeypress	KeyPress	
mouse			MOUSE_EVENT			MOUSE_EVENT_MASK			
mouse_press			XCB_BUTTON_PRESS	WM_LBUTTONDOWN	MOUSE_PRESSED	onmousedown	MouseDown		
mouse_release			XCB_BUTTON_RELEASE	WM_LBUTTONUP	MOUSE_RELEASED	onmouseup	MouseUp		
mouse_click	SDL_MouseButton				MOUSE_CLICKED	onclick			
mouse_move	SDL_MouseMotion		XCB_MOTION_NOTIFY	WM_MOUSEMOVE	MOUSE_MOVED / MOUSE_MOTION_EVENT_MASK	onmousemove	MouseMove		
mouse_drag	SDL_MouseMotion				MOUSE_DRAGGED		Drag		
							DragEnd		
							DragEnterClick		
							DragEnter		
							DragLeave		
							DragOver		
							DragStart		
mouse_drop	SDL_DropEvent						Drop		request a file open by the user
mouse_enter				XCB_ENTER_NOTIFY	WM_MOUSEHOVER	MOUSE_ENTERED	onmouseover	MouseOver	
mouse_leave				XCB_LEAVE_NOTIFY	WM_MOUSELEAVE	MOUSE_EXITED	onmouseout	MouseOut	tracking of mouse hover behavior
					WM_CAPTURECHANGED				
					WM_LBUTTONDBLCLK		ondblclick		
				WM_ACTIVATE	WM_MOUSEACTIVATE				
mouse_wheel_horizontally					WM_MOUSEHWHEEL				
mouse_wheel	SDL_MouseWheel				WM_MOUSEWHEEL	MOUSE_WHEEL / MOUSE_WHEEL_EVENT_MASK		MouseWheel	
					WM_NCHITTEST				
joystick									
joystick_axis	SDL_JoyAxis								
joystick_ball	SDL_JoyBall								
joystick_hat	SDL_JoyHat								
joystick_button	SDL_JoyButton								
joystick_device	SDL_JoyDevice								
touchpad									send y-coordinate like mouse
touchpad_cancel								TouchCancel	
touchpad_end								TouchEnd	
touchpad_move								TouchMove	
touchpad_start	SDL_TouchFingerEvent							TouchStart	SDL_FINGERMOTION, SDL_FINGERDOWN, SDL_FINGERUP
gesture									rotation, scale, hover to get an extended view. up view of what's inside. swipe left and right to scroll. swipe up to accept a phone call, tilt head (dual-swipe)
gesture_change								GestureChange	
gesture_end	SDL_MultiGestureEvent							GestureEnd	x, y, theta, distance, number of fingers
gesture_start								GestureStart	
expose				XCB_EXPOSE					
window									
	SDL_Window		WINDOW_BUFFER_SIZE_EVENT					Resize	the event source is resized
window_close								Close	
								Window Closing	just before the browser window closes or navigates to a different site
								Scroll	
								Window Scroll	the browser window is scrolled
						WINDOW_EVENT_MASK			
						WINDOW_FOCUS_EVENT_MASK			
						WINDOW_STATE_EVENT_MASK			
miscellaneous	SDL_TextEditing								
	SDL_TextInput								
							onabort	Change	bei Abbruch
							onchange		beim Ändern
							onerror	Error	bei Fehlerfalle
							onload	Load	beim Laden einer Datei
							onreset		beim Zurücksetzen des Formulars
							onselect	SelectionEvent	beim Selektieren von Text
							onsubmit	FormPanelSubmitEvent	bis Form is submitted
							onunload		beim Verlassen der Datei
							javascript		bei Ereignissen
								AttachEvent	the event source is attached to the browser's document or detached from it
								BeforeSelection	
								Blur	
								CanPlayThrough	
								CellPreviewEvent	allows the previewing of events before they are fired to cells
								Click	
								ColumnSortEvent	
								ContextMenu	
								DoubleClick	
								Ended	
								EventSubmitPreview	preview of a native event
								FormPanelSubmitComplete	a form has been submitted successfully
								FormPanelSubmit	
								Highlight	
								Initialize	the event source is initialized
								LoadingStateChangeEvent	the data loading state has changed
								LoseCapture	
								Open	
								PlaceChangedEvent	the user has reached a new location in the app
								PlaceChangeRequestEvent	the user may go to a new place in the app, or tries to leave it
								Progress	
								RangeChangeEvent	
								RowCountChangeEvent	
								ScatteredEvent	
								SelectionChangeEvent	
								Selection	
								ShowRange	should be used when a widget displays a range of values to the user
								ValueChange	
			MENU_EVENT						
			FOCUS_EVENT			FOCUS_EVENT_MASK	onfocus	Focus	beim Aktivieren
							onblur		beim Verlassen
			WM_GETHOTKEY						Determines the hot key associated with a window.
			WM_SETHOTKEY						Associates a hot key with the window. When the user presses the hot key, the system activates the window.
			WM_APPCOMMAND						Notifies a window that the user generated an application command event, for example, by clicking an application command button using the mouse or typing an application command key on the keyboard.
			WM_CHAR						Posted to the window with the keyboard focus when a nonsystem key is pressed. A nonsystem key is a key that is pressed when the ALT key is not pressed.
			WM_DEADCHAR						Posted to the window with the keyboard focus when a nonsystem key is pressed. A nonsystem key is a key that is pressed when the ALT key is not pressed when the keyboard focus has the keyboard focus.
			WM_HOTKEY						Posted when the user presses a hot key registered by the RegisterHotKey function. The message is placed at the top of the message queue associated with the thread that registered the hot key.
			WM_KILLFOCUS						Sent to a window immediately before it loses the keyboard focus.
			WM_SETFOCUS						Sent to a window after it has gained the keyboard focus.
			WM_SYSDEADCHAR						Sent to the window with the keyboard focus when a WM_SYSKEYDOWN message is translated by the TranslateMessage function. WM_SYSDEADCHAR specifies the character code of a system dead key – that is, a dead key that is pressed while holding down the ALT key.
			WM_SYSKEYDOWN						Posted to the window with the keyboard focus when the user presses the F10 key (which activates the menu bar) or holds down the ALT key and then presses another key. It also occurs when no window currently has the keyboard focus; in this case, the WM_SYSKEYDOWN message is sent to the active window. The window that receives the message can distinguish between these two contexts by checking the context code in the lParam parameter.
			WM_SYSKEYUP						Posted to the window with the keyboard focus when the user releases a key that was pressed while the ALT key was held down. It also occurs when no window currently has the keyboard focus; in this case, the WM_SYSKEYUP message is sent to the active window. The window that receives the message can distinguish between these two contexts by checking the context code in the lParam parameter.
			WM_UNICHAR						Posted to the window with the keyboard focus when a WM_SYSKEYDOWN message is translated by the TranslateMessage function. TheWM_UNICHAR message contains the character code of the key that was pressed.
					ACTION_EVENT_MASK				
					ADJUSTMENT_EVENT_MASK				
					COMPONENT_EVENT_MASK				
					CONTAINER_EVENT_MASK				
					HIERARCHY_BOUNDS_EVENT_MASK				
					HIERARCHY_EVENT_MASK				
					INPUT_METHOD_EVENT_MASK				
					INVOCATION_EVENT_MASK				
					ITEM_EVENT_MASK				
					PAINT_EVENT_MASK				
					RESERVED_ID_MAX				
					TEXT_EVENT_MASK				

cyboi	unix_terminal	win32_console	x_window_system	win32_display	java_script	java	description
keyboard							
-		bKeyDown is true					
-		bKeyDown is false					
-		wRepeatCount					indicates that a key is being held down
-		wVirtualKeyCode	[still missing in xcb tutorial]				identifies key in a device-independent manner
key		wVirtualScanCode	detail (xcb_keycode_t)				physical key on the keyboard; device-dependent value generated by the keyboard hardware
character		uChar – UnicodeChar					translated Unicode character
mask		dwControlKeyState	state				mask of the buttons (or control keys) during the event
mouse							
-			response_type				type of event
button			mode				number of the mouse button that was clicked for enter/leave events
-			window				window identification
-			event_x				x coordinate where the mouse has been pressed in the window
-			event_y				y coordinate where the mouse has been pressed in the window
expose							
-			x				x coordinate of the top-left part of the window that needs to be redrawn
-			y				y coordinate of the top-left part of the window that needs to be redrawn
-			width				width coordinate of the top-left part of the window that needs to be redrawn
-			height				height coordinate of the top-left part of the window that needs to be redrawn

cyboi	unix_terminal	win32_console	x_window_system	win32_display	java_script	java	description
		ENHANCED_KEY 0x0100		-			The key is enhanced.
		LEFT_ALT_PRESSED 0x0002		-		ALT_DOWN_MASK	The left ALT key is pressed.
		NUMLOCK_ON 0x0020		-			The NUM LOCK light is on.
		RIGHT_ALT_PRESSED 0x0001		-		ALT_GRAPH_DOWN_MASK	The right ALT key is pressed.
		SCROLLLOCK_ON 0x0040		-			The SCROLL LOCK light is on.
		-		MK_LBUTTON 0x0001			The left mouse button is down.
		-		MK_RBUTTON 0x0002			The right mouse button is down.
		-		MK_MBUTTON 0x0010			The middle mouse button is down.
		-		MK_XBUTTON1 0x0020			The first X button is down.
		-		MK_XBUTTON2 0x0040			The second X button is down.
			XCB_BUTTON_MASK_1			BUTTON1_DOWN_MASK	
			XCB_BUTTON_MASK_2			BUTTON2_DOWN_MASK	
			XCB_BUTTON_MASK_3			BUTTON3_DOWN_MASK	
			XCB_BUTTON_MASK_4				
			XCB_BUTTON_MASK_5				
		SHIFT_PRESSED 0x0010	XCB_MOD_MASK_SHIFT	MK_SHIFT 0x0004		SHIFT_DOWN_MASK	The SHIFT key is pressed.
		CAPSLOCK_ON 0x0080	XCB_MOD_MASK_LOCK				The CAPS LOCK light is on.
		LEFT_CTRL_PRESSED 0x0008	XCB_MOD_MASK_CONTROL	MK_CONTROL 0x0008		CTRL_DOWN_MASK	The left CTRL key is pressed.
			XCB_MOD_MASK_1				Alt or Meta key, respectively
			XCB_MOD_MASK_2				
			XCB_MOD_MASK_3				
			XCB_MOD_MASK_4				
			XCB_MOD_MASK_5				
		RIGHT_CTRL_PRESSED 0x0004		MK_CONTROL 0x0008		CTRL_DOWN_MASK	The right CTRL key is pressed.
						META_DOWN_MASK	

cyboi	unix_terminal	win32_console	x_window_system	win32_display	java_script	java	description
				HTBORDER 18			In the border of a window that does not have a sizing border.
				HTBOTTOM 15			In the lower-horizontal border of a resizable window (the user can click the mouse to resize the window vertically).
				HTBOTTOMLEFT 16			In the lower-left corner of a border of a resizable window (the user can click the mouse to resize the window diagonally).
				HTBOTTOMRIGHT 17			In the lower-right corner of a border of a resizable window (the user can click the mouse to resize the window diagonally).
				HTCAPTION 2			In a title bar.
				HTCLIENT 1			In a client area.
				HTCLOSE 20			In a Close button.
				HTERROR -2			On the screen background or on a dividing line between windows (same asHTNOWHERE, except that the DefWindowProc function produces a system beep to indicate an error).
				HTGROWBOX 4			In a size box (same as HTSIZE).
				HTHELP 21			In a Help button.
				HTHSCROLL 6			In a horizontal scroll bar.
				HTLEFT 10			In the left border of a resizable window (the user can click the mouse to resize the window horizontally).
				HTMENU 5			In a menu.
				HTMAXBUTTON 9			In a Maximize button.
				HTMINBUTTON 8			In a Minimize button.
				HTNOWHERE 0			On the screen background or on a dividing line between windows.
				HTREDUCE 8			In a Minimize button.
				HTRIGHT 11			In the right border of a resizable window (the user can click the mouse to resize the window horizontally).
				HTSIZE 4			In a size box (same as HTGROWBOX).
				HTSYSMENU 3			In a window menu or in a Close button in a child window.
				HTTOP 12			In the upper-horizontal border of a window.
				HTTOPLEFT 13			In the upper-left corner of a window border.
				HTTOPRIGHT 14			In the upper-right corner of a window border.
				HTTRANSPARENT -1			In a window currently covered by another window in the same thread (the message will be sent to underlying windows in the same thread until one of them returns a code that is not HTTRANSPARENT).
				HTVSCROLL 7			In the vertical scroll bar.
				HTZOOM 9			In a Maximize button.

haptic_output

haptic	http://bobbens.dyndns.org/journal/2010/sdl_haptic/	remark
type	haptic effect type: SDL_HAPTIC_SINE, SDL_HAPTIC_SQUARE, SDL_HAPTIC_TRIANGLE, SDL_HAPTIC_SAWTOOTHUP, SDL_HAPTIC_SAWTOOTHDOWN	important
SDL_HapticDirection	direction of the effect	
replay_length	duration of haptic effect	important
replay_delay	delay before starting the effect	
trigger_button (LEGACY)	button that triggers haptic effect	legacy from old game controllers
trigger_interval (LEGACY)	How soon it can be triggered again after button	legacy from old game controllers
periodic_period	period of the wave	important
periodic_magnitude	peak value	important
periodic_offset	mean value of the wave	does not affect the effect much
periodic_phase	Horizontal shift given by hundredth of a cycle	does not affect the effect much
envelope_attack_length	duration of the attack	important
envelope_attack_level	level at the start of the attack	important
envelope_fade_length	duration of the fade	important
envelope_fade_level	level at the end of the fade	important