pygame$_{sdl2}$ Documentation

Release latest

Nov 19, 2018

Contents

There isn't much pygame_sdl2-specific documentation at the moment. Since pygame_sdl2 tries to follow pygame as closely as possible, the best reference is the pygame documentation:

http://www.pygame.org/docs/

The list of what has been implemented can be found in the README:

https://github.com/renpy/pygame_sdl2

An Android packaging example can be found at:

https://github.com/renpytom/rapt-pygame-example

There have been a few additions to the Pygame API, documented below.

CHAPTER 1

Importing as pygame

```
import pygame_sdl2
pygame_sdl2.import_as_pygame()
```

Will modify sys.modules so that pygame_sdl2 modules are used instead of their pygame equivalents. For example, after running the code above, the code:

```
import pygame.image
img = pygame.image.load("logo.png")
```

will use pygame_sdl2 to load the image, instead of pygame. (This is intended to allow code to run on pygame_sdl2 or pygame.)

CHAPTER 2

Mobile

Pygame_sdl2 exposes the SDL2 application lifecycle events, which are used to pause and resume the application on mobile platforms. The most useful events are:

APP_WILLENTERBACKGROUND Generated when the app will enter the background. The app should immediately stop drawing the screen and playing sounds. It should save its state, as the application may be killed at any time after this event has been handled.

APP_DIDENTERFOREGROUND Generated when the app will enter the foreground. The app should delete the saved state (as it is no longer needed), and resume drawing the screen and playing sounds.

In addition, the set of keycodes now include the SDL2 application control keyboard codes. Most notably, pygame_sdl2.K_AC_BACK is the code for the Android back button.

CHAPTER 3

Text Input

Several functions have been added to allow more complex text input.

`pygame_sdl2.key.`**`start_text_input`**`()`
Starts text input. If an on-screen keyboard is supported, it is shown.

`pygame_sdl2.key.`**`stop_text_input`**`()`
Stops text input and hides the on-screen keyboard.

`pygame_sdl2.key.`**`set_text_input_rect`**`(`*rect*`)`
Sets the text input rectangle. This is used by input methods and, on some platforms, to ensure the text is not blocked by an on-screen keyboard.

`pygame_sdl2.key.`**`has_screen_keyboard_support`**`()`
Returns true of the platform supports an on-screen keyboard.

`pygame_sdl2.key.`**`is_screen_keyboard_shown`**`(`*Window window=None*`)`
Returns true if the on-screen keyboard is shown.

During text input, the unicode field of the KEYDOWN object is not set. Instead, two new events are generated:

TEXTINPUT Generated when text has been added.

text The text that has been added.

TEXTEDITING Used when text is being edited by an input method (IME).

text The text that is being edited. This is usually displayed underlined.

start*, *length Used by IMEs to display text being actively edited. This is generaly displayed with a thicker underline.

CHAPTER 4

Mouse Wheel

pygame.event.**set_mousewheel_buttons**(*flag*)

When *flag* is true (the default), the vertical mouswheel is mapped to buttons 4 and 5, with mousebuttons 4 and greater being offset by 2.

When flag is false, the mousebuttons retain their numbers, and MOUSEWHEEL events are generated.

pygame.event.**get_mousewheel_buttons**()

Returns the mousewheel buttons flag.

MOUSEWHEEL Generated by mousewheel motion.

x The amount of motion of the mousewheel in the x axis.

y The amount of motion of the mousewheel in the y axis.

CHAPTER 5

Multiple Mice

The mouse events (MOUSEBUTTONDOWN, MOUSEMOTION, MOUSEBUTTONUP, and MOUSEWHEEL) have a *which* field that identifies the mouse that generated the event. When equal to pygame_sdl2.TOUCH_MOUSEID, the event was generated by a touch of the screen.

CHAPTER 6

HighDPI/Retina

When the pygame.WINDOW_ALLOW_HIGHDPI flag is passed to pygame.display.set_mode, opengl surfaces can be created in HighDPI/Retina mode. When this occurs, the drawable size of a window will be larger than the size of the window.

pygame.display.**get_drawable_size**()

Gets the drawable size of the window created with pygame.display.set_mode()

Index

P