pygame documentation

search

Most useful stuff:
Color | display | draw | event | font | image | key | locals | mixer | mouse | Rect | Surface | time | music | pygame

Advanced stuff: cursors | joystick | mask | sprite | transform | BufferProxy | freetype | gfxdraw | midi | PixelArray | pixelcopy | sndarray | surfarray | math

Other: camera | controller | examples | fastevent | scrap | tests | touch | version

Pygame Front Page

Quick start

Welcome to pygame! Once you've got pygame installed (`pip install pygame` or `pip3 install pygame` for most people), the next question is how to get a game loop running. Pygame, unlike some other libraries, gives you full control of program execution. That freedom means it is easy to mess up in your initial steps.

Here is a good example of a basic setup (opens the window, updates the screen, and handles events)--

```
# Example file showing a basic pygame "game loop"
import pygame

# pygame setup
pygame.init()
screen = pygame.display.set_mode((1280, 720))
clock = pygame.time.Clock()
running = True

while running:
    # poll for events
    # pygame.QUIT event means the user clicked X to close your windo
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
```

```
    # RENDER YOUR GAME HERE

    # flip() the display to put your work on screen
    pygame.display.flip()

    clock.tick(60)  # limits FPS to 60

pygame.quit()
```

Here is a slightly more fleshed out example, which shows you how to move something (a circle in this case) around on screen--

```
# Example file showing a circle moving on screen
import pygame

# pygame setup
pygame.init()
screen = pygame.display.set_mode((1280, 720))
clock = pygame.time.Clock()
running = True
dt = 0

player_pos = pygame.Vector2(screen.get_width() / 2, screen.get_height

while running:
    # poll for events
    # pygame.QUIT event means the user clicked X to close your windo
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    # fill the screen with a color to wipe away anything from last fr
    screen.fill("purple")

    pygame.draw.circle(screen, "red", player_pos, 40)

    keys = pygame.key.get_pressed()
    if keys[pygame.K_w]:
        player_pos.y -= 300 * dt
    if keys[pygame.K_s]:
        player_pos.y += 300 * dt
    if keys[pygame.K_a]:
        player_pos.x -= 300 * dt
    if keys[pygame.K_d]:
```

```
    # limits FPS to 60
    # dt is delta time in seconds since last frame, used for framerat
    # independent physics.
    dt = clock.tick(60) / 1000

pygame.quit()
```

For more in depth reference, check out the Tutorials section below, check out a video tutorial (I'm a fan of this one), or reference the API documentation by module.

Documents

Readme
: Basic information about pygame: what it is, who is involved, and where to find it.

Install
: Steps needed to compile pygame on several platforms. Also help on finding and installing prebuilt binaries for your system.

File Path Function Arguments
: How pygame handles file system paths.

Pygame Logos
: The logos of Pygame in different resolutions.

LGPL License
: This is the license pygame is distributed under. It provides for pygame to be distributed with open source and commercial software. Generally, if pygame is not changed, it can be used with any type of program.

Tutorials

Import and Initialize

The beginning steps on importing and initializing pygame. The pygame package is made of several modules. Some modules are not included on all platforms.

How do I move an Image?

A basic tutorial that covers the concepts behind 2D computer animation. Information about drawing and clearing objects to make them appear animated.

Chimp Tutorial, Line by Line

The pygame examples include a simple program with an interactive fist and a chimpanzee. This was inspired by the annoying flash banner of the early 2000s. This tutorial examines every line of code used in the example.

Sprite Module Introduction

Pygame includes a higher level sprite module to help organize games. The sprite module includes several classes that help manage details found in almost all games types. The Sprite classes are a bit more advanced than the regular pygame modules, and need more understanding to be properly used.

Surfarray Introduction

Pygame used the NumPy python module to allow efficient per pixel effects on images. Using the surface arrays is an advanced feature that allows custom effects and filters. This also examines some of the simple effects from the pygame example, arraydemo.py.

Camera Module Introduction

Pygame, as of 1.9, has a camera module that allows you to capture images, watch live streams, and do some basic computer vision. This tutorial covers those use cases.

Newbie Guide

game.

Display Modes
Getting a display surface for the screen.

한국어 튜토리얼 (Korean Tutorial)
빨간블록 검은블록

Reference

Index
A list of all functions, classes, and methods in the pygame package.

pygame.BufferProxy
An array protocol view of surface pixels

pygame.Color
Color representation.

pygame.cursors
Loading and compiling cursor images.

pygame.display
Configure the display surface.

pygame.draw
Drawing simple shapes like lines and ellipses to surfaces.

pygame.event
Manage the incoming events from various input devices and the windowing platform.

pygame.examples
Various programs demonstrating the use of individual pygame modules.

pygame.freetype
: Enhanced pygame module for loading and rendering font faces.

pygame.gfxdraw
: Anti-aliasing draw functions.

pygame.image
: Loading, saving, and transferring of surfaces.

pygame.joystick
: Manage the joystick devices.

pygame.key
: Manage the keyboard device.

pygame.locals
: Pygame constants.

pygame.mixer
: Load and play sounds

pygame.mouse
: Manage the mouse device and display.

pygame.mixer.music
: Play streaming music tracks.

pygame
: Top level functions to manage pygame.

pygame.PixelArray
: Manipulate image pixel data.

pygame.Rect
: Flexible container for a rectangle.

pygame.sndarray
: Manipulate sound sample data.

pygame.sprite
: Higher level objects to represent game images.

pygame.Surface
: Objects for images and the screen.

pygame.surfarray
: Manipulate image pixel data.

pygame.tests
: Test pygame.

pygame.time
: Manage timing and framerate.

pygame.transform
: Resize and move images.

pygame C API
: The C api shared amongst pygame extension modules.

Search Page
: Search pygame documents by keyword.

Edit on GitHub