Frequently Asked Questions Pygame

**1. Setting up a Pygame Application**

[1.1 What is the minimum amount of code needed to make a Pygame window appear?](#question1_1)

[1.2 I want to make a full game project, what is a good main.py file startup?](#question1_2)

**2. Drawing Shapes with pygame.draw**

[2.1 How do I draw a rectangle on my screen?](#question2_1)

[2.2 How do I draw a polygon on my screen?](#question2_2)

[2.3 How do I draw a circle or an ellipse on my screen?](#question2_3)

[2.4 How do I draw a line on my screen?](#question2_4)

[2.5 How do I draw shapes to surface that is not my screen?](#question2_5)

[2.6 I drew a shape on a surface, now how can I make the outside of my shape transparent?](#question2_6)

**3. Creating and Using Sprites**

3.1 How to I create a Sprite I can use in my game?

1.1 What is the minimum amount of code needed to make a Pygame window appear?

The following snippet of code will create a window 800 pixels wide by 600 pixels tall. Note that the set\_mode function requires width and height be a tuple.

import pygame

pygame.init()

WIDTH = 800

HEIGHT = 600

screen = pygame.display.set\_mode( (WIDTH, HEIGHT) )

running = True

while running:

for event in pygame.event.get():

if event.type == pygame.QUIT:

running = False

# flip the display updates user screen with images

pygame.display.flip()

pygame.quit()

1.2 I want to make a full game project, what is a good main.py file startup?

import pygame

# these constants may be stored in a setting.py module

WIDTH = 800

HEIGHT = 600

FPS = 60

WHITE = (255, 255, 255)

###

class Application:

def \_\_init\_\_(self):

# create the screen

self.screen = pygame.display.set\_mode( (WIDTH, HEIGHT) )

# create a group for all sprites

self.sprites = pygame.sprite.Group()

# create a clock object to manage frame rate and ticks

self.clock = pygame.time.Clock()

# a bool value to end the game loop

self.running = True

def gameloop(self):

while self.running:

# maintain FPS rate and measure delta time

self.dt = self.clock.tick(FPS) / 1000

# handle all user events since last frame

self.update\_events()

# update sprite logic and positions

self.sprites.update()

# update the screen

self.update\_screen()

def update\_screen(self):

# fill with background image or color

self.screen.fill(WHITE)

# draw sprites on screen

self.sprites.draw(self.screen)

# flip screen from previous frame to new frame

pygame.display.flip()

def update\_events(self):

for event in pygame.event.get():

if event.type == pygame.QUIT:

self.running = False

# additional inputs can be handled here.

def main():

pygame.init()

app = Application()

app.gameloop()

pygame.quit()

if \_\_name\_\_ == '\_\_main\_\_':

main()

2.1 How do I draw a rectangle on my screen?

Use pygame.draw.rect function.

pygame.draw.rect(*surface*, *color*, *rect*, *width* = 0)

*surface* is surface you want to draw on. If you are drawing on your screen, then this is your screen variable.

color is a tuple represeing RGB values. This is often a constant you have made.

rect is either a Rect object of a tuple of 4 values (x, y, width, height)

The optional argument width defaults to 0. When 0, the shape is filled in. Otherwise it is hollow.

There are several ways to use this function.

Example: draw a filled red rectangle on screen at (x,y) = (100, 200), width = 50, height = 25

RED = (255, 0, 0)

pygame.draw.rect(screen, RED, (100, 200, 50, 25))

Example: draw a hollow blue rect at x= 500, y = 400, width =100, height = 20 with edge width size of 1

BLUE = (0, 0, 255)

pygame.draw.rect(screen, BLUE, (500, 400, 100, 20), width = 1)

Example: create a Rect object, then draw that object

GREEN = (0, 255, 0)

my\_rect = pygame.Rect(150, 550, 40, 40)

pygame.draw.rect(screen, GREEN, my\_rect)

2.2 How do I draw a polygon on my screen?

pygame.draw.rect(*surface*, *color*, *rect*, *width* = 0)

*surface* is surface you want to draw on.

*color* is a tuple represeing RGB values.

*points* is a list or tuple of points which are in tuple form.

*width* = 0 is default, making the polygon filled.

Example

GREEN = (0, 255, 0)

points = [(50, 100), (50, 200), (100, 150)]

pygame.draw.polygon(screen, GREEN, points) # filled

pygame.draw.polygon(screen, RED, points, width = 2) # hollow

The above example will make a green filled triangle that is outlined in red.

2.3 How do I draw a circle or an ellipse on my screen?

A circle can be drawin with either the circle or ellipse function.

pygame.draw.circle(surface, color, center, radius, width = 0)

*surface* is surface you want to draw on.

*color* is a tuple represeing RGB values.

*center* is a tuple for (x, y) or a vector.

*radius* is the radius of the circle.

*width* = 0 (optional) is used for line thickness. width = 0 is filled; width > 0 is hollow.

pygame.draw.ellipse(surface, color, rect, width = 0)

*surface* is surface you want to draw on.

*color* is a tuple represeing RGB values.

*rect* is either a Rect object or a tuple (x, y, width, height) for a bounding rectangle

*width* = 0 (optional) is used for line thickness. width = 0 is filled; width > 0 is hollow.

Examples

RED = (255, 0, 0)

BLUE = (0, 0, 255)

GREEN = (0, 255, 0)

# draw a blue circle at (400, 300) with radius 50

pygame.draw.circle(screen, BLUE, (400, 300), 50)

# draw a hollow circle with red outline

pygame.draw.circle(screen, RED, (100, 400), 25, width = 1)

# creating Rect object at (550, 200) with width = height = 75

rect1 = pygame.Rect(550, 200, 75, 75)

# drawing a circle that inscribes rect1

pygame.draw.ellipse(screen, GREEN, rect1)

# creating a Rect that is 100 pixels long but only 50 high

rect2 = pygame.Rect(200, 40, 100, 50)

# drawing the ellipse bound by the rect2 object.

pygame.draw.ellipse(screen, BLUE, rect2)

2.4 How do I draw a line on my screen?

Use the line function.

pygame.draw.line(surface, color, start, end, width = 1)

*surface* is surface you want to draw on.

*color* is a tuple represeing RGB values.

*start* is a tuple for (x, y) for one endpoint of the line

*end* is a tupe for the other endpoint of the line.

*width* = 1 (optional) is used for line thickness. width = 1 is default.

pygame.draw.line(screen, BLUE, (100, 200), (600, 50))

pygame.draw.line(screen, GREEN, (200, 300), (400, 400), width = 3)

2.5 How do I draw shapes to surface that is not my screen?

Shapes can be drawn onto any surface object in pygame.

You can create your own instances of the Surface class or, since an image is also a surface, load a surface with pygame.image.load.

In this example, we create an image object by making a surface, then drawing to the surface a circle.

RED = (255, 0, 0)

# create a screen surface

screen = pygame.display.set\_mode( (WIDTH, HEIGHT) )

# create an image surface

image = pygame.Surface( (100, 100) )

# calculate center of our image surface

x, y = image.get\_width() / 2, image.get\_height() / 2

# draw red circle in center of the image

pygame.draw.circle(image, RED, (x, y), 20)

# draw the circle image on the screen.

screen.blit(image, (400, 300))

2.6 I drew a shape on a surface, now how can I make the outside of my shape transparent?

Use the set\_colorkey(*color*) method that belongs to a Surface object to set the color you want to be transparent.

In this example, we make a red circle on an image surface and then make the part of the surface that is outside of the circle transparent (instead of the black color it is by default).

RED = (255, 0, 0)

BLACK = (0, 0, 0)

WHITE = (255, 255, 255)

# create a screen

screen = pygame.display.set\_mode((800, 600))

# make the screen a white background

screen.fill(WHITE)

# create a surface that will perfectly fit a circle with radius 50

radius = 50

image = pygame.Surface( (radius \* 2, radius \* 2) )

x, y = radius, radius

pygame.draw.circle(image, RED, (x, y), radius)

# make BLACK transparent

image.set\_colorkey(BLACK)

# draw the circle image on the screen.

screen.blit(image, (400, 300))