


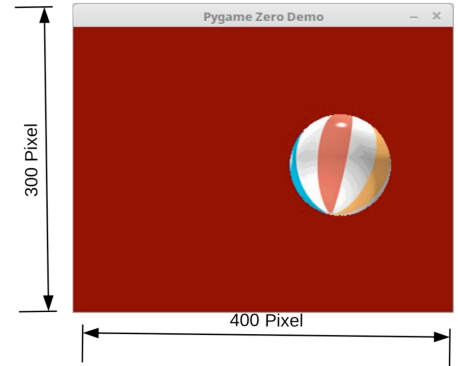
pygamezerodemo.py  No imports are required! In order for this to work, you have to run the program with "pgzrun pygamezerodemo.py".

TITLE = "Pygame Zero Demo"

WIDTH = 400

HEIGHT = 300

 Title and dimensions of the window




class Ball():

def __init__(self):

self.actor = Actor("ball", (150,150))

self.speedx = 2

self.speedy = 2

 Creates an Actor object. The file "images/ball.gif" is expected as actor's image. Other file endings are possible.

def update(self):

ball.actor.x += self.speedx

ball.actor.y += self.speedy


 Sets the speed in the x and y directions.

if self.actor.left < 0 or self.actor.right > WIDTH:

self.bounce(xdir=True, ydir=False)

if self.actor.top < 0 or self.actor.bottom > HEIGHT:

self.bounce(xdir=False, ydir=True)

 Changes the direction of the ball if it touches a boundary.

def bounce(self, xdir=True, ydir=False):

if xdir:

self.speedx *= -1


if ydir:

self.speedy *= -1

 Reverses the direction of the ball in the x or y direction.

def update():


ball.update()

 Update function which is called 60 times per second for each frame. This updates the game world but doesn't draw it to the screen.

def draw():

screen.fill((128,0,0))


ball.actor.draw()

 Draw function, which renders to the screen after the update.


def on_mouse_down(pos):

if ball.actor.collidepoint(pos):

ball.bounce()

 One of many different event functions. It is called when the mouse is clicked. When called, the position of the mouse is passed as a parameter.

ball = Ball()

 Creates a Ball object that will be used by the update, draw, and on_mouse_down functions.



<http://pygame-zero.readthedocs.io>