

pygamezerodemo.py ← Es werden keine Imports benötigt. Dafür muss die Datei mit 'pgzrun pygamezerodemo.py' aufgerufen werden.

TITLE = "Pygame Zero Demo"

WIDTH = 400

HEIGHT = 300

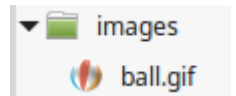
← Dimensionen und Titel des Fensters

class Ball():

def __init__(self):

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

Erzeugt einen Actor. Als Bild wird eine Datei 'images/ball.gif' erwartet. Andere Dateieendungen sind möglich.



self.speedx = 2

self.speedy = 2

← Geschwindigkeit in x- und y-Richtung

def update(self):

ball.actor.x += self.speedx

ball.actor.y += self.speedy

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)

← Richtung des Balls ändern, wenn er den Rand berührt.

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

if xdir:

self.speedx *= -1

if ydir:

self.speedy *= -1

← Kehre die Bewegung des Balls in x- oder y-Richtung um.

Update-Funktion, die für jeden Frame 60 mal pro Sekunde einmal aufgerufen wird. Hier wird die Spielwelt aktualisiert aber noch nichts gezeichnet.

def update():

ball.update()

def draw():

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

ball.actor.draw()

← Draw-Funktion, die nach dem Update den Bildschirm zeichnet.

def on_mouse_down(pos):

if ball.actor.collidepoint(pos):

ball.bounce()

← Eine von verschiedenen Event-Funktionen. Sie wird bei einem Mouseclick aufgerufen. Beim Aufruf wird die Position der Mouse als Parameter übergeben.

ball = Ball()

← Erzeugt einen Ball, der von den Funktionen update, draw und on_mouse_down genutzt wird.



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

