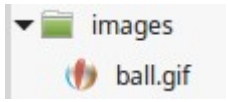
```
# pygamezerodemo.py

TITLE = "Pygame Zero Demo"
WIDTH = 400
HEIGHT = 300

class Ball():
    def __init__(self):
        self.actor = Actor("ball", (150,150))

        self.speed = [2, 2]

    def update(self):
        ball.actor.x += self.speed[0]
        ball.actor.y += self.speed[1]

        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)

    def bounce(self, xdir=True, ydir=False):
        if xdir:
            self.speed[0] *= -1
        if ydir:
            self.speed[1] *= -1

def update():
    ball.update()

def draw():
    screen.fill((128,0,0))
    ball.actor.draw()

def on_mouse_down(pos):
    if ball.actor.collidepoint(pos):
        ball.bounce()

ball = Ball()
```

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

Dimensionen und Titel des Fensters

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



Geschwindigkeit in x- und y-Richtung

Richtung des Balles ändern, wenn der Rand berührt wird

Kehre die X- oder Y-Richtung des Balles um.

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

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

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

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