Abprallen

Die Figur bewegt sich und prallt an den Ränder zurück.

Wir erstellen einen Ball, der sich mit einer Geschwindigkeit von 150 Pixel pro Sekunde auf dem Bildschirm bewegt. Wenn es den Rand erreicht, prallt es zurück

Der Ball hat eine Geschwindigkeit mit zwei Komponenten: eine für jede Achse. Wenn der Ball den Rand berührt, wird eine der beiden Komponenten “invertiert".

```
WIDTH = 640
HEIGHT = 480

ball = Actor('ball', anchor=('left', 'top'))
velocity = {'x': 150, 'y': 150}

def update(dt):
    if ball.x + ball.width >= WIDTH:
        velocity['x'] = -velocity['x']
    if ball.x < 0:
        velocity['x'] = -velocity['x']
    if ball.y + ball.height >= HEIGHT:
        velocity['y'] = -velocity['y']
    if ball.y < 0:
        velocity['y'] = -velocity['y']
    ball.x += velocity['x'] * dt
    ball.y += velocity['y'] * dt

def draw():
    screen.fill((255, 255, 153))
    ball.draw()
```

In der `update`-Funktion überprüfen wir die aktuelle Position und vergleichen sie mit der Fenstergrösse.

Beim Erstellen des `ball`s definieren wir, dass der Actor in der oberen linken Ecke verankert werden soll (die Standard-einstellung ist die Mitte). Das macht den Vergleich mit dem Rand einfacher: Wir können den oberen und den linken Rand mit `0` vergleichen. Für den rechten und den unteren Rand müssen wir die Breite oder Höhe des Actors hinzufügen, um die Position mit `WIDTH` und `HEIGHT` vergleichen zu können.

Die `x` und `y`-Komponente der `ball`-Position werden durch Addition der entsprechenden Geschwindigkeitskomponente ausgerechnet, beide mit `dt` multipliziert.
`dt` ist die Zeit, die seit dem letzten Aufruf der `update`-Funktion vergangen ist.
Wenn wir die Geschwindigkeit nicht mit `dt` multiplizieren würden, würde sich der Ball,

bei jedem Aufruf der update Funktion (sehr, sehr häufig!) um 150 Pixel bewegen, und nicht jeder Sekunde.

Wenn du die "Play" Taste drückst, solltest du einen Ball sehen, der um das Fenster hüpft.