

Check collision



With the right arrow key, you make a shoe move: when it hits the ball, the ball "jumps" to the right.

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 shoe = Actor('shoe', (80, 200))
5 ball = Actor('ball', (230, 200)) 
6
7 def on_key_up(key, mod):
8     if key == keys.RIGHT:
9         shoe.x += 10
10
11 def update(dt):
12     if ball.colliderect(shoe):
13         ball.x += 50
14
15 def draw():
16     screen.fill((255, 255, 153))
17     shoe.draw()
18     ball.draw()
```

We need two actors, `shoe` and `ball`, with a matching image in the `images` directory. When creating the actor, we also pass the width and height of each image: in our case, the shoe is 80 pixels wide and 200 pixels high.

We use the `on_key_up` hook to bind the right arrow key with the shoe movement.

In the `update` function, we check if the ball touches (`collides`) shoe. If it's the case, the ball moves horizontally by 50 pixels.

Check collision



With the right arrow key, you make a shoe move: when it hits the ball, the ball "jumps" to the right.

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 shoe = Actor('shoe', (80, 200))
5 ball = Actor('ball', (230, 200)) 
6
7 def on_key_up(key, mod):
8     if key == keys.RIGHT:
9         shoe.x += 10
10
11 def update(dt):
12     if ball.colliderect(shoe):
13         ball.x += 50
14
15 def draw():
16     screen.fill((255, 255, 153))
17     shoe.draw()
18     ball.draw()
```

We need two actors, `shoe` and `ball`, with a matching image in the `images` directory. When creating the actor, we also pass the width and height of each image: in our case, the shoe is 80 pixels wide and 200 pixels high.

We use the `on_key_up` hook to bind the right arrow key with the shoe movement.

In the `update` function, we check if the ball touches (`collides`) shoe. If it's the case, the ball moves horizontally by 50 pixels.

Check collision



With the right arrow key, you make a shoe move: when it hits the ball, the ball "jumps" to the right.

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 shoe = Actor('shoe', (80, 200))
5 ball = Actor('ball', (230, 200)) 
6
7 def on_key_up(key, mod):
8     if key == keys.RIGHT:
9         shoe.x += 10
10
11 def update(dt):
12     if ball.colliderect(shoe):
13         ball.x += 50
14
15 def draw():
16     screen.fill((255, 255, 153))
17     shoe.draw()
18     ball.draw()
```

We need two actors, `shoe` and `ball`, with a matching image in the `images` directory. When creating the actor, we also pass the width and height of each image: in our case, the shoe is 80 pixels wide and 200 pixels high.

We use the `on_key_up` hook to bind the right arrow key with the shoe movement.

In the `update` function, we check if the ball touches (`collides`) shoe. If it's the case, the ball moves horizontally by 50 pixels.

Check collision



With the right arrow key, you make a shoe move: when it hits the ball, the ball "jumps" to the right.

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 shoe = Actor('shoe', (80, 200))
5 ball = Actor('ball', (230, 200)) 
6
7 def on_key_up(key, mod):
8     if key == keys.RIGHT:
9         shoe.x += 10
10
11 def update(dt):
12     if ball.colliderect(shoe):
13         ball.x += 50
14
15 def draw():
16     screen.fill((255, 255, 153))
17     shoe.draw()
18     ball.draw()
```

We need two actors, `shoe` and `ball`, with a matching image in the `images` directory. When creating the actor, we also pass the width and height of each image: in our case, the shoe is 80 pixels wide and 200 pixels high.

We use the `on_key_up` hook to bind the right arrow key with the shoe movement.

In the `update` function, we check if the ball touches (`collides`) shoe. If it's the case, the ball moves horizontally by 50 pixels.