# Moving around



You can use the arrow keys – or other keys like a, d, w, s – to move around a character.

At the start of the code, we initialize an "Actor" with its start position (320, 240), the center of the window.

On each cycle, before calling the update and draw functions, PyGame Zero calls a few "event handling hooks".

One of them is the <code>on\_key\_up</code> function, which you can used to check which key has been pressed.

In on\_key\_up, we move the bee around, by increasing or decreasing its  $\mathbf x$  and  $\mathbf y$  coordinates.

## Moving around



You can use the arrow keys – or other keys like a, d, w, s – to move around a character.

At the start of the code, we initialize an "Actor" with its start position (320, 240), the center of the window.

On each cycle, before calling the update and draw functions, PyGame Zero calls a few "event handling hooks".

One of them is the <code>on\_key\_up</code> function, which you can used to check which key has been pressed.

In on\_key\_up, we move the bee around, by increasing or decreasing its  ${\bf x}$  and  ${\bf y}$  coordinates.



## Moving around



You can use the arrow keys – or other keys like a, d, w, s – to move around a character.

At the start of the code, we initialize an "Actor" with its start position (320, 240), the center of the window.

On each cycle, before calling the update and draw functions, PyGame Zero calls a few "event handling hooks".

One of them is the <code>on\_key\_up</code> function, which you can used to check which key has been pressed.

In on\_key\_up, we move the bee around, by increasing or decreasing its  $\mathbf x$  and  $\mathbf y$  coordinates.

# Moving around



You can use the arrow keys – or other keys like a, d, w, s – to move around a character.

At the start of the code, we initialize an "Actor" with its start position (320, 240), the center of the window.

On each cycle, before calling the update and draw functions, PyGame Zero calls a few "event handling hooks".

One of them is the <code>on\_key\_up</code> function, which you can used to check which key has been pressed.

In on\_key\_up, we move the bee around, by increasing or decreasing its  $\mathbf x$  and  $\mathbf y$  coordinates.





```
1
   WIDTH = 640
2
   HEIGHT = 480
3
   bee = Actor('bee', (320, 240))
4
5
   def update(dt):
6
7
     pass
8
9
   def on_key_up(key, mod):
       if key == keys.LEFT:
10
         bee.x -= 10
11
       elif key == keys.RIGHT:
12
13
       bee.x += 10
       elif key == keys.UP:
14
          bee.y -= 10
15
       elif key == keys.DOWN:
16
         bee.y += 10
17
18
   def draw():
19
20
       screen.fill((0, 210, 255))
21
    bee.draw()
```

🕈 🕮 Ale Rimoldi, Coderdojo Zürich

```
🕈 🕮 Ale Rimoldi, Coderdojo Zürich
```

def draw():

WIDTH = 640

HEIGHT = 480

def update(dt):

pass

bee = Actor('bee', (320, 240))

def on\_key\_up(key, mod):

if key == keys.LEFT:

elif key == keys.UP:

bee.y -= 10

bee.v += 10

bee.draw()

elif key == keys.DOWN:

screen.fill((0, 210, 255))

elif key == keys.RIGHT:

bee.x -= 10

bee.x += 10

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

```
1
   WIDTH = 640
2
   HEIGHT = 480
3
   bee = Actor('bee', (320, 240))
4
5
   def update(dt):
6
7
      pass
8
9
   def on_key_up(key, mod):
       if key == keys.LEFT:
10
      bee.x -= 10
11
       elif key == keys.RIGHT:
12
       bee.x += 10
13
       elif key == keys.UP:
14
      bee.y -= 10
15
       elif key == keys.DOWN:
16
         bee.y += 10
17
18
19
   def draw():
       screen.fill((0, 210, 255))
20
21
       bee.draw()
```

```
WIDTH = 640
2
   HEIGHT = 480
3
4
   bee = Actor('bee', (320, 240))
5
6
   def update(dt):
7
       pass
8
9
   def on_key_up(key, mod):
       if key == keys.LEFT:
10
       bee.x -= 10
11
       elif key == keys.RIGHT:
12
       bee.x += 10
13
       elif key == keys.UP:
14
       bee.y -= 10
15
       elif key == keys.DOWN:
16
           bee.y += 10
17
18
19
   def draw():
       screen.fill((0, 210, 255))
20
21
       bee.draw()
```