## Moving around



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

Before calling the <code>update</code> and <code>draw</code> functions, PyGame Zero also calls a few "event handling hooks". The <code>on\_key\_up</code> function can be used to check which key has been pressed:

```
WIDTH = 640
2
   HEIGHT = 480
3
   bee = Actor('bee')
   def update(dt):
7
   pass
8
9
   def on_key_up(key, mod):
10
       if key == keys.LEFT:
       bee.x -= 10
11
       elif key == keys.RIGHT:
12
13
       bee.x += 10
14
       elif key == keys.UP:
       bee.y -= 10
15
       elif key == keys.DOWN:
16
17
        bee.y += 10
18
19 def draw():
2.0
       screen.fill((0, 210, 255))
       bee.draw()
21
```

Every "Actor" automatically has a position that is initialized to (0,0), the top left corner. We move the bee around, by increasing or decreasing its x and y coordinates.

## Moving around



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

Before calling the <code>update</code> and <code>draw</code> functions, PyGame Zero also calls a few "event handling hooks". The <code>on\_key\_up</code> function can be used to check which key has been pressed:

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

Every "Actor" automatically has a position that is initialized to (0,0), the top left corner. We move the bee around, by increasing or decreasing its x and y coordinates.

## Moving around



You can use the arrow keys – or other keys like  $_{\rm a}$ ,  $_{\rm d}$ ,  $_{\rm w}$ ,  $_{\rm s}$  – to move around a character.

Before calling the <code>update</code> and <code>draw</code> functions, PyGame Zero also calls a few "event handling hooks". The <code>on\_key\_up</code> function can be used to check which key has been pressed:

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

Every "Actor" automatically has a position that is initialized to (0,0), the top left corner. We move the bee around, by increasing or decreasing its x and y coordinates.

## Moving around



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

Before calling the update and draw functions, PyGame Zero also calls a few "event handling hooks". The on\_key\_up function can be used to check which key has been pressed:

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

Every "Actor" automatically has a position that is initialized to (0,0), the top left corner. We move the bee around, by increasing or decreasing its x and y coordinates.