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')
5
   def update(dt):
7
    pass
8
9
   def on_key_up(key, mod):
10
       print(key)
11
       if key == keys.LEFT:
12
           bee.x -= 10
13
       elif key == keys.RIGHT:
           bee.x += 10
        elif key == keys.UP:
15
           bee.y -= 10
16
17
        elif key == keys.DOWN:
18
           bee.y += 10
19
2.0
   def draw():
21
       screen.fill((0, 210, 255))
22
       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 <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
       print (key)
       if key == keys.LEFT:
11
           bee.x -= 10
12
13
       elif key == keys.RIGHT:
14
       elif key == keys.UP:
15
           bee.y -= 10
16
       elif key == keys.DOWN:
17
18
           bee.y += 10
19
20
   def draw():
21
       screen.fill((0, 210, 255))
22
       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 <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
4
   bee = Actor('bee')
   def update(dt):
7
    pass
9
   def on_key_up(key, mod):
10
       print(key)
11
    if key == keys.LEFT:
12
           bee.x -= 10
13
       elif key == keys.RIGHT:
14
           bee.x += 10
       elif key == keys.UP:
15
          bee.y -= 10
16
17
       elif key == keys.DOWN:
18
           bee.y += 10
19
20 def draw():
21
    screen.fill((0, 210, 255))
       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
       print(key)
11
       if key == keys.LEFT:
           bee.x -= 10
12
13
       elif key == keys.RIGHT:
14
       elif key == keys.UP:
15
           bee.y -= 10
16
       elif key == keys.DOWN:
17
18
           bee.y += 10
19
20 def draw():
21
       screen.fill((0, 210, 255))
       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.