

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
2 HEIGHT = 480
3
4 bee = Actor('bee')
5
6 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:
14        bee.x += 10
15    elif key == keys.UP:
16        bee.y -= 10
17    elif key == keys.DOWN:
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 `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
2 HEIGHT = 480
3
4 bee = Actor('bee')
5
6 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:
14        bee.x += 10
15    elif key == keys.UP:
16        bee.y -= 10
17    elif key == keys.DOWN:
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 `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
2 HEIGHT = 480
3
4 bee = Actor('bee')
5
6 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:
14        bee.x += 10
15    elif key == keys.UP:
16        bee.y -= 10
17    elif key == keys.DOWN:
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 `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
2 HEIGHT = 480
3
4 bee = Actor('bee')
5
6 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:
14        bee.x += 10
15    elif key == keys.UP:
16        bee.y -= 10
17    elif key == keys.DOWN:
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.