

I'm Tiago,

I

- Software Development Consultancy.
- Python Training for Professionals.
- Product Management and Lead Dev @Promptar.

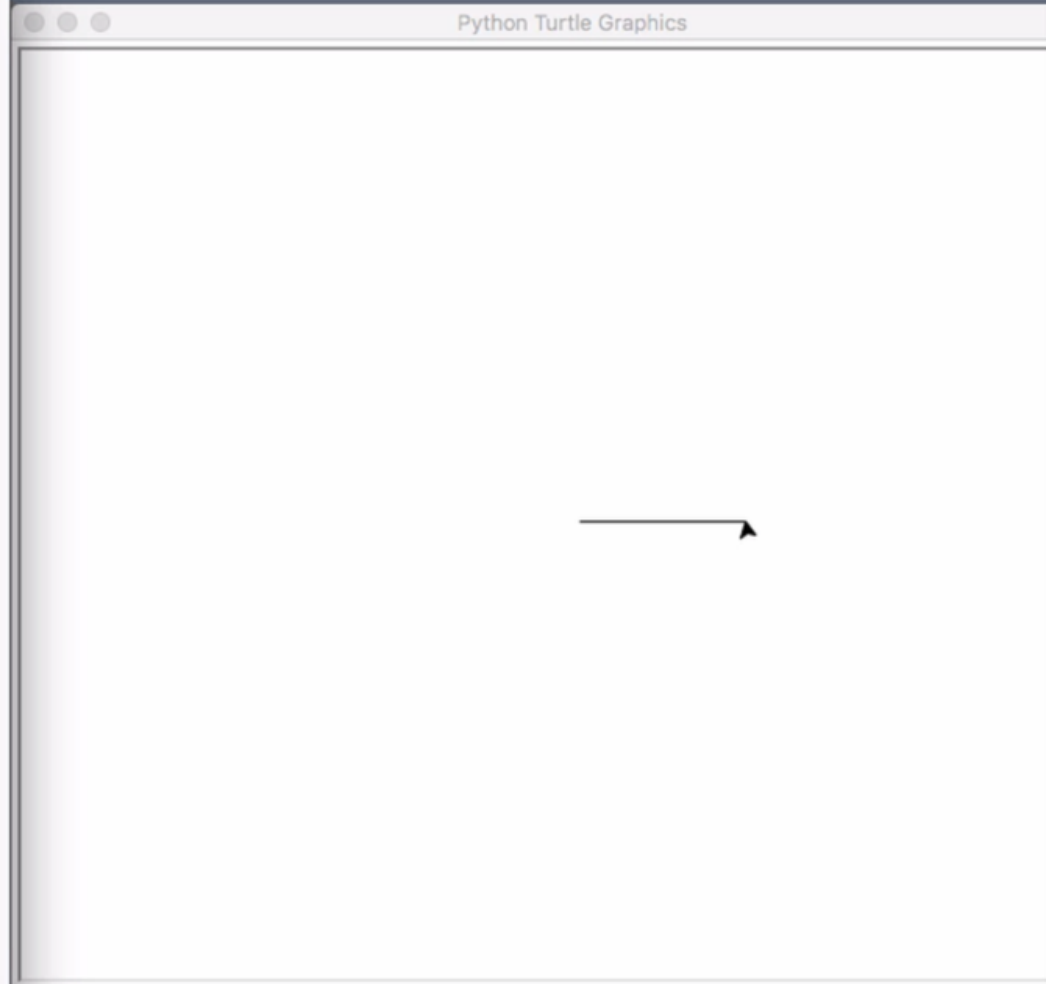
<https://tmont.es/>

@setnomt

\$ █

```
>>> import turtle
>>> turtle.clear()
>>> turtle.forward(100)
>>> turtle.left(100)
>>> █
```

I



```
>>> import turtle
>>> turtle.clear()
>>> turtle.forward(100)
>>> turtle.left(100)
>>> turtle.forward(100)
>>> turtle.up()
>>> turtle.forward(100)
>>> █
```

I



```
>>> import turtle
>>> turtle.clear()
>>> turtle.forward(100)
>>> turtle.left(100)
>>> turtle.forward(100)
>>> turtle.up()
>>> turtle.forward(100)
>>> turtle.forward(-100)
>>> turtle.down()
>>> turtle.goto(0, 0)
>>> █
```

I

Python Turtle Graphics



05:16

dont-do-this-at-work-ep2019.mp4

-40:31



39:27



```
$ clerar  
$ which python3.7  
/usr/local/bin/python3.7  
$ which python3.7 > game.py  
$ █
```

```
#!/usr/local/bin/python3.7
```

```
import turtle
```

```
turtle.mainloop()
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
::
```

```
$ clrar  
$ which python3.7  
/usr/local/bin/python3.7  
$ which python3.7 >  
$ chmod 750 game.py  
$ vi game.py
```



```
#!/usr/local/bin/python3.7
```

```
import turtle
```

```
turtle.Screen().setup(startx=600, width=640, height=640)
```

```
turtle.bgcolor('orange')
```

```
turtle.mainloop()
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

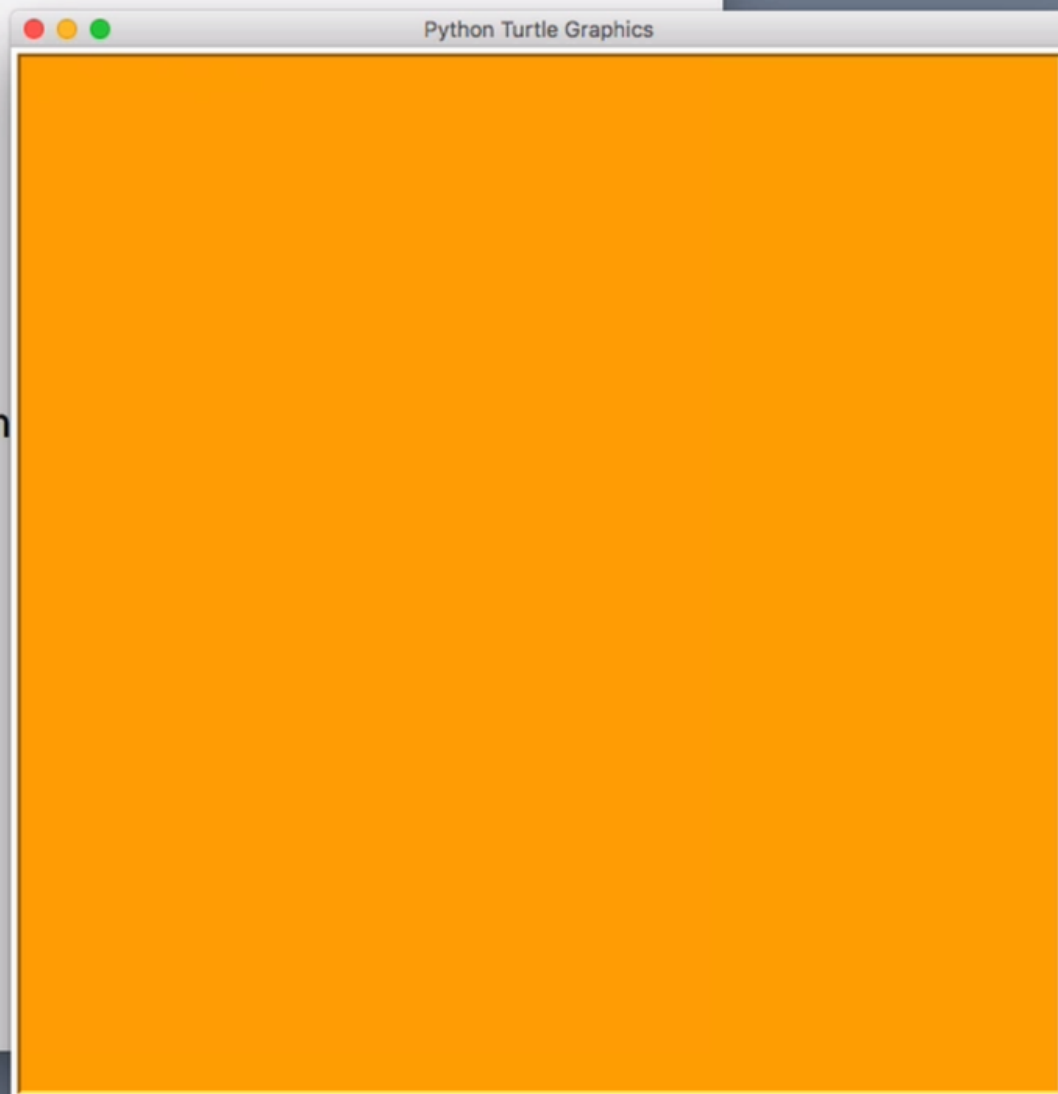
```
~
```

```
~
```

```
~
```

```
$ clrar  
$ which python3.7  
/usr/local/bin/python3.7  
$ which python3.7 > game.py  
$ chmod 750 game.py  
$ vi game.py
```

Press ENTER or type command to continue



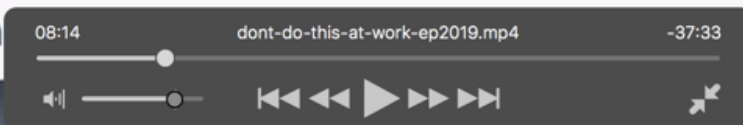
```
turtle.bgcolor('orange')

GRID_SIZE = 80
GRID_SPAN = 3
GRID_MAX = GRID_SIZE * GRID_SPAN

def line(x1, y1, x2, y2):
    turtle.up()
    turtle.goto(x1, y1)
    turtle.down()
    turtle.goto(x2, y2)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)

turtle.mainloop()
~
~
"game.py" 22L, 437C written
```

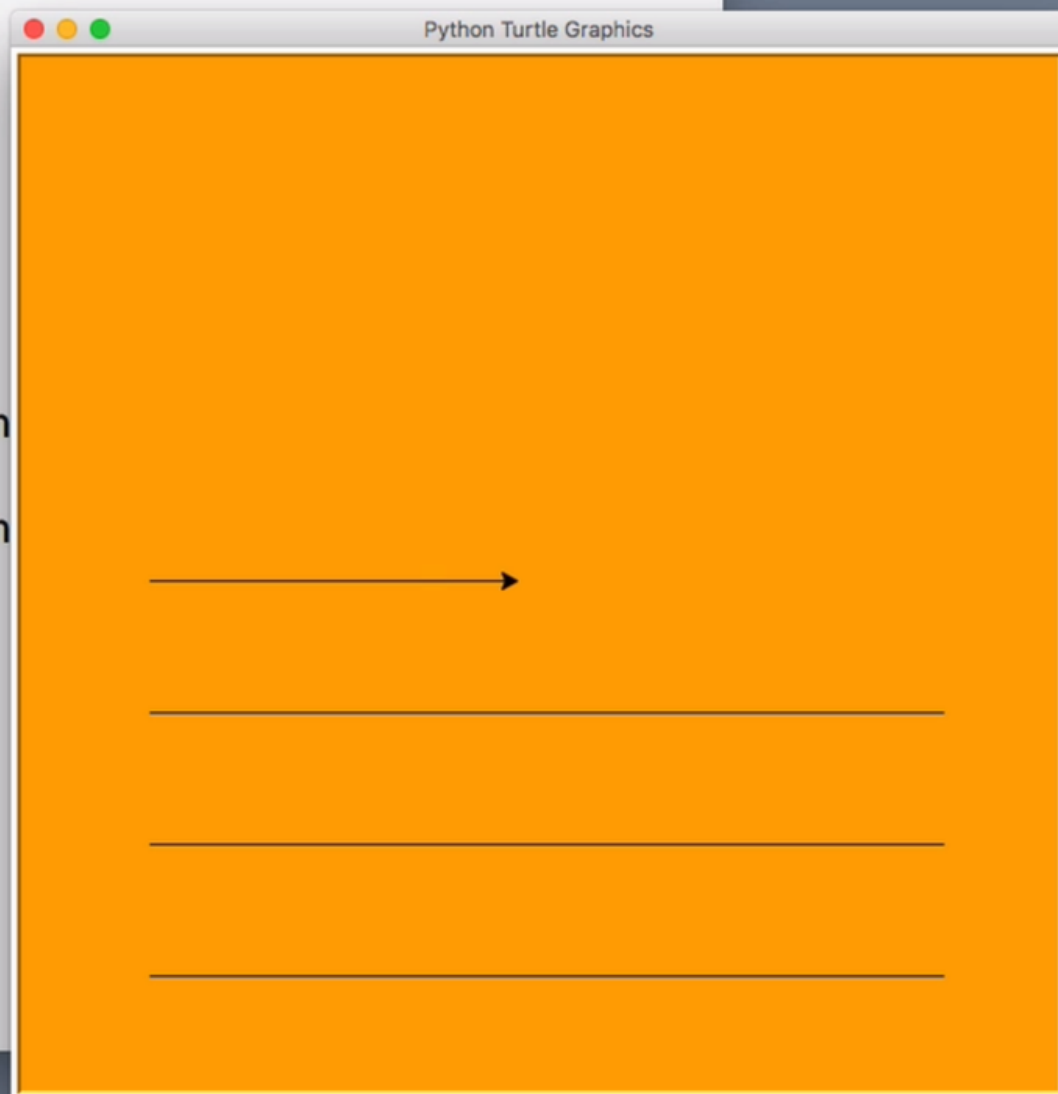


```
$ clerar  
$ which python3.7  
/usr/local/bin/python3.7  
$ which python3.7 > game.py  
$ chmod 750 game.py  
$ vi game.py
```

Press ENTER or type command to continue

Press ENTER or type command to continue

□



```
GRID_SPAN = 3
GRID_MAX = GRID_SIZE * GRID_SPAN

def line(x1, y1, x2, y2):
    turtle.up()
    turtle.goto(x1, y1)
    turtle.down()
    turtle.goto(x2, y2)

turtle.speed(0)
turtle.hideturtle()
turtle.color('dark orange')
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)
```

-- INSERT --

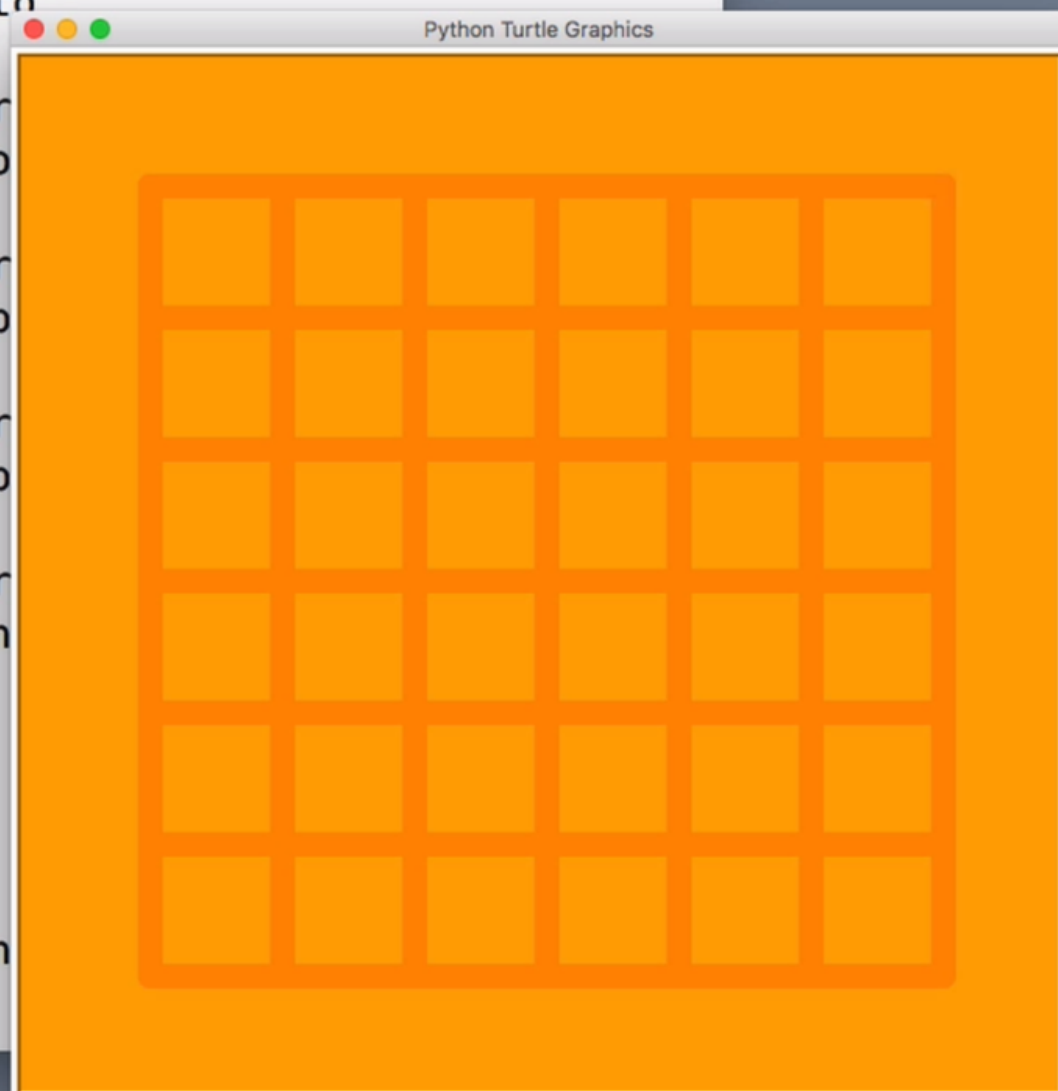


```
thon3.7/turtle.py", line 1776, in goto
    self._goto(Vec2D(x, y))
File "/Library/Frameworks/Python.framework/Versions/3.7/turtle.py", line 3195, in _goto
    self._update() #count=True)
File "/Library/Frameworks/Python.framework/Versions/3.7/turtle.py", line 2660, in _update
    self._update_data()
File "/Library/Frameworks/Python.framework/Versions/3.7/turtle.py", line 2646, in _update_data
    self.screen._incrementudc()
File "/Library/Frameworks/Python.framework/Versions/3.7/turtle.py", line 1292, in _incrementudc
    raise Terminator
turtle.Terminator

shell returned 1

Press ENTER or type command to continue

```



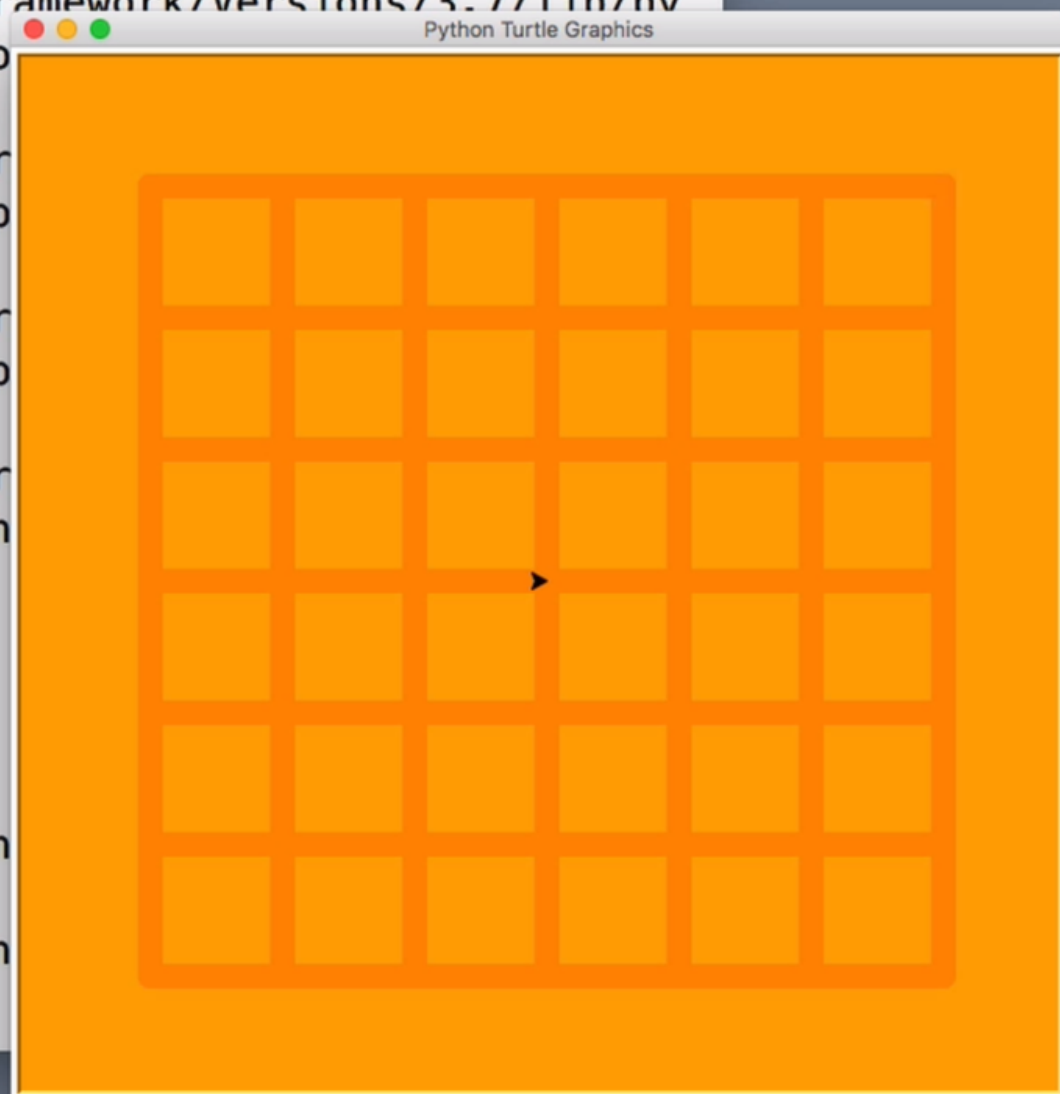
```
def line(x1, y1, x2, y2):  
    turtle.up()  
    turtle.goto(x1, y1)  
    turtle.down()  
    turtle.goto(x2, y2)  
  
turtle.speed(0)  
turtle.hideturtle()  
turtle.color('dark orange')  
turtle.width(15)  
  
for i in range(-GRID_SPAN, GRID_SPAN+1):  
    scaled_i = i * GRID_SIZE  
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)  
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)  
  
player = turtle.Turtle()
```

```
File "/Library/Frameworks/Python.framework/Versions/3.7/lib/nv
thon3.7/turtle.py", line 3195, in _go
    self._update() #count=True)
File "/Library/Frameworks/Python.framework/Versions/3.7/lib/nv
thon3.7/turtle.py", line 2660, in _up
    self._update_data()
File "/Library/Frameworks/Python.framework/Versions/3.7/lib/nv
thon3.7/turtle.py", line 2646, in _up
    self.screen._incrementudc()
File "/Library/Frameworks/Python.framework/Versions/3.7/lib/nv
thon3.7/turtle.py", line 1292, in _in
    raise Terminator
turtle.Terminator

shell returned 1

Press ENTER or type command to continue

Press ENTER or type command to continue
█
```



```
turtle.color('dark orange')
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

player = turtle.Turtle()
turtle.register_shape('player.gif')
player.shape('player.gif')
```

```
turtle.mainloop()
```

```
~
~
~
~
~
~
```

```
self._update() #count=True)
File "/Library/Frameworks/Python.frameworks/Python3.7/turtle.py", line 2660, in _update
    self._update_data()
File "/Library/Frameworks/Python.frameworks/Python3.7/turtle.py", line 2646, in _update
    self.screen._incrementturtles()
File "/Library/Frameworks/Python.frameworks/Python3.7/turtle.py", line 1292, in _incrementturtles
    raise Terminator
turtle.Terminator
```

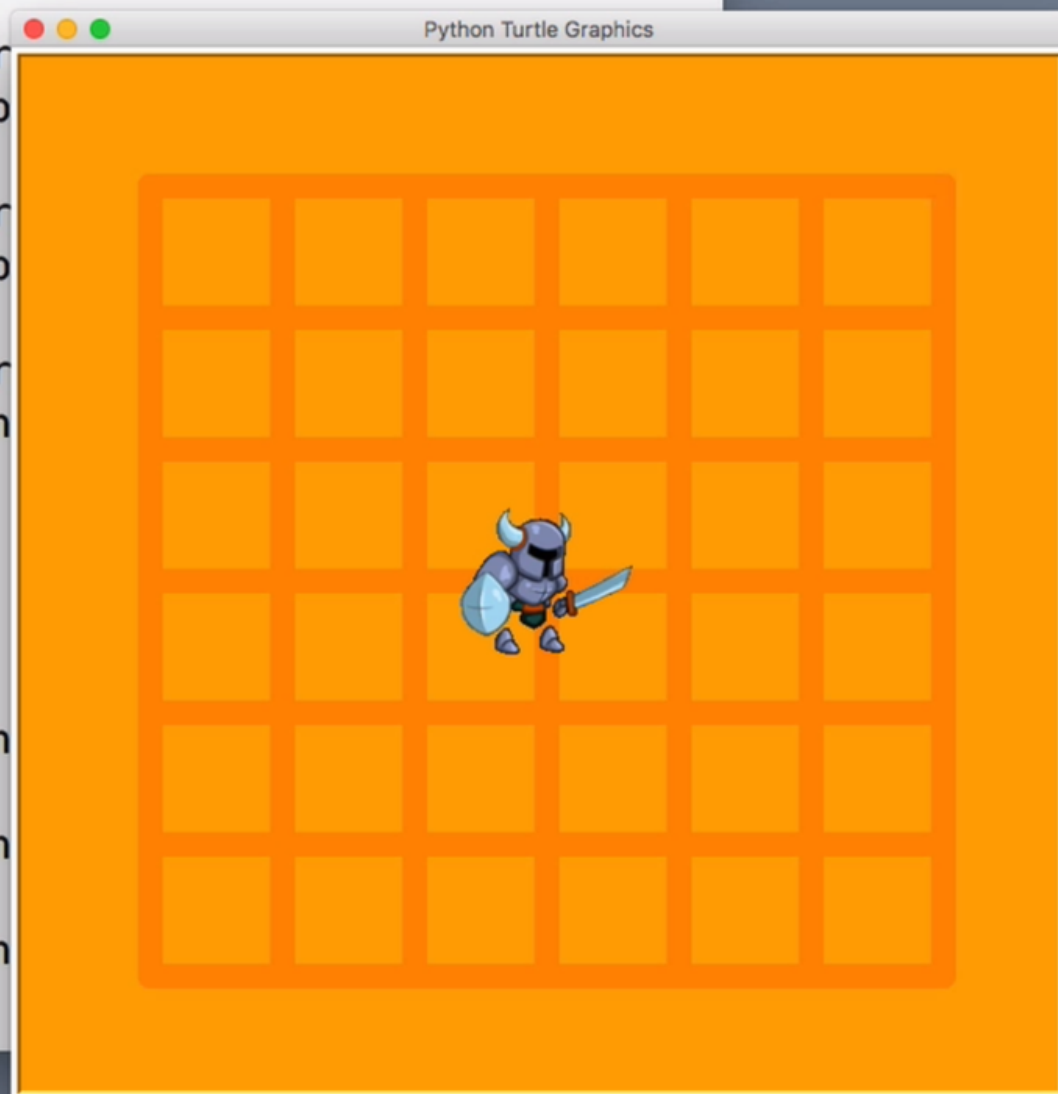
shell returned 1

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□



```
bash-3.2$ ls *gif
beast.gif      player.gif
bash-3.2$ █
```

Author statement of CC0 (public domain)

<https://opengameart.org/content/some-characters>

Assets downloaded from [opengamegraphics.com](http://opengamegraphics.com)

bash-3.2\$ █



```
turtle.color('dark orange')
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

player = turtle.Turtle()
turtle.register_shape('player.gif')
player.shape('player.gif')

turtle.listen()

turtle.onkey(lambda: move_player(0, 1), 'Up')
turtle.onkey(lambda: move_player(0, -1), 'Down')
turtle.onkey(lambda: move_player(-1, 0), 'Left')
turtle.onkey(lambda: move_player(1, 0), 'Right')
```

"game.py" 39L, 869C written



```
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

player = turtle.Turtle()
turtle.register_shape('player.gif')
player.shape('player.gif')

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)

turtle.listen()

turtle.onkey(lambda: move_player(0, 1), 'Up')
turtle.onkey(lambda: move_player(0, -1), 'Down')
"game.py" 43L, 976C written
```

`https://opengameart.org/content/s`

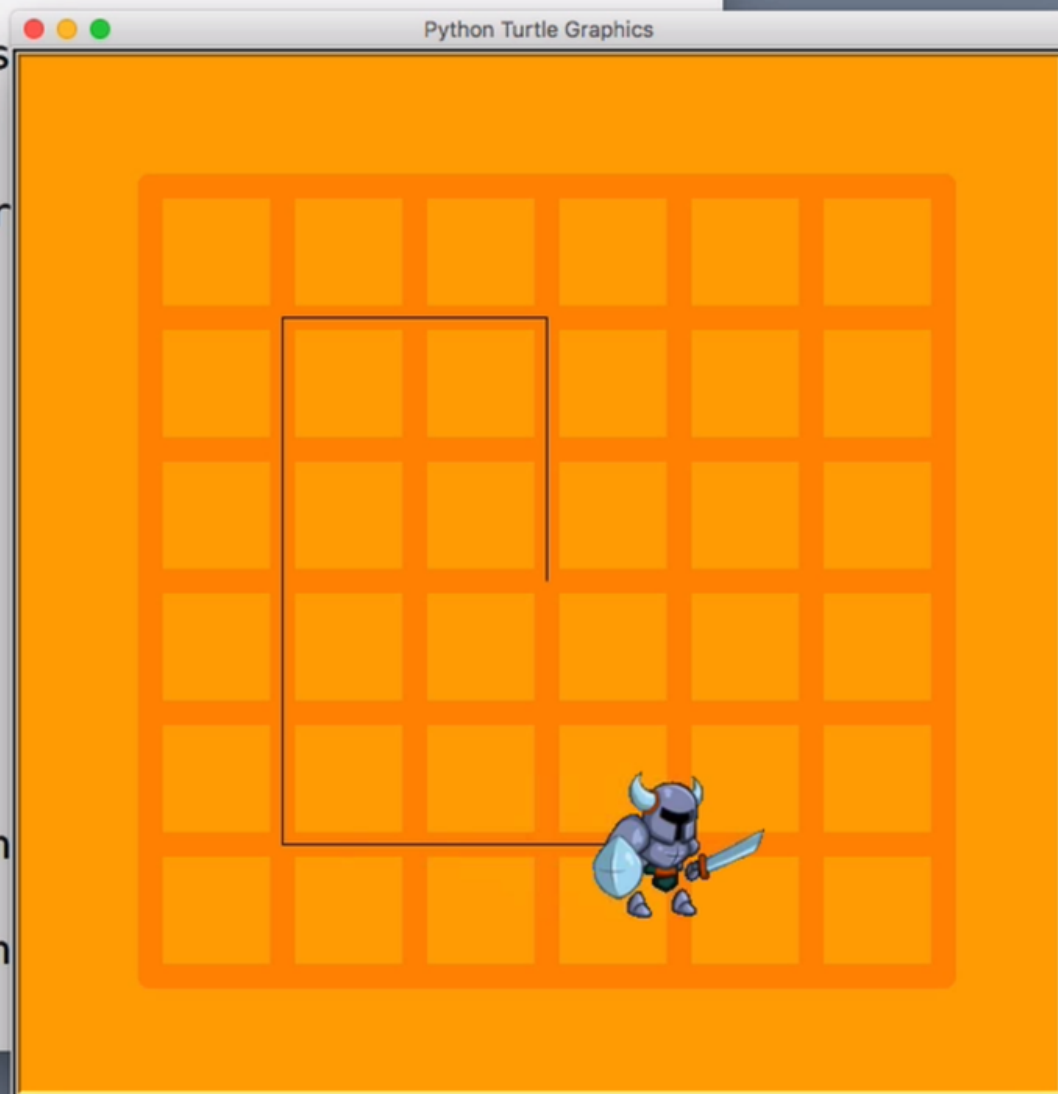
Assets downloaded from opengamegr

`bash-3.2$ exit`

Press ENTER or type command to contin

Press ENTER or type command to contin

□



`https://opengameart.org/content/s`

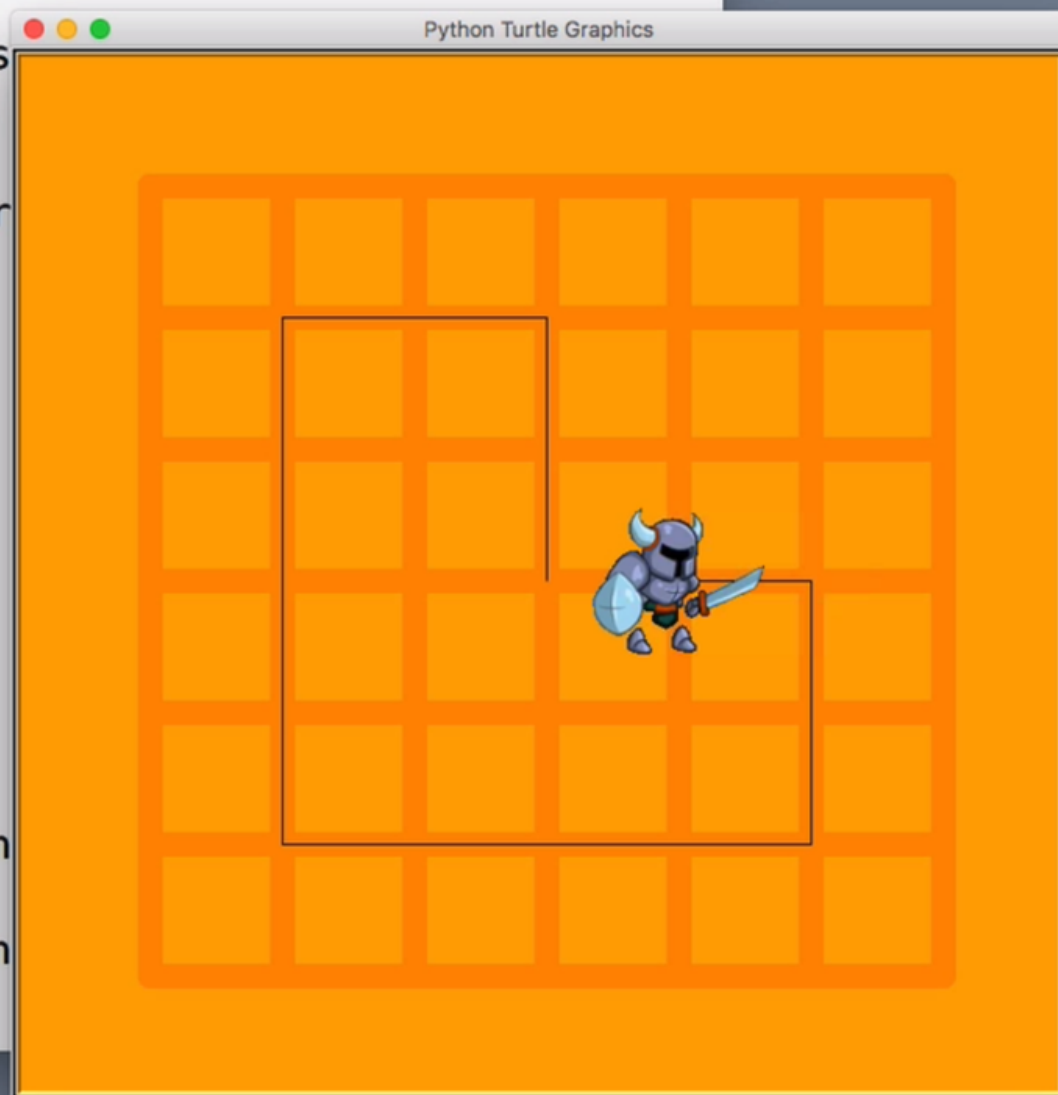
Assets downloaded from opengamegr

`bash-3.2$ exit`

Press ENTER or type command to contin

Press ENTER or type command to contin

□



```
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

player = turtle.Turtle()
turtle.register_shape('player.gif')
player.shape('player.gif')
player.up()

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)

turtle.listen()

turtle.onkey(lambda: move_player(0, 1), 'Up')
"game.py" 44L, 988C written
```

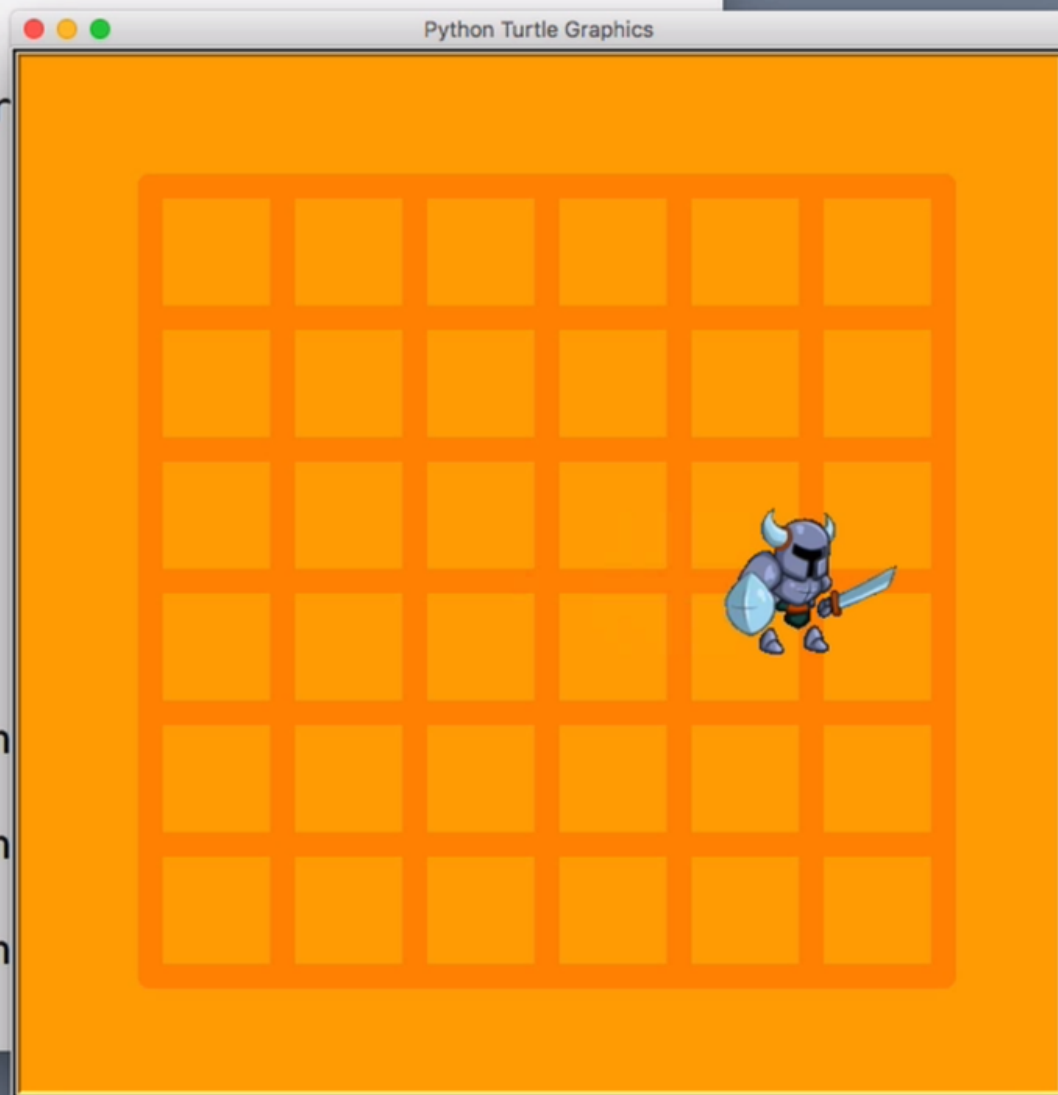
Assets downloaded from opengamegr

```
bash-3.2$ exit
```

Press ENTER or type command to contin

Press ENTER or type command to contin

Press ENTER or type command to contin



```
turtle.width(15)

for i in range(-GRID_SPAN, GRID_SPAN+1):
    scaled_i = i * GRID_SIZE
    line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
    line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)

def actor(image):
    t = turtle.Turtle()
    turtle.register_shape(image)
    t.shape(image)
    t.up()
    return t

beast = actor('beast.gif')
player = actor('player.gif')

def move_player(dx, dy):
    x, y = player.position()
```

Assets downloaded from [opengamegraphics.com](http://opengamegraphics.com)

```
bash-3.2$ exit
```

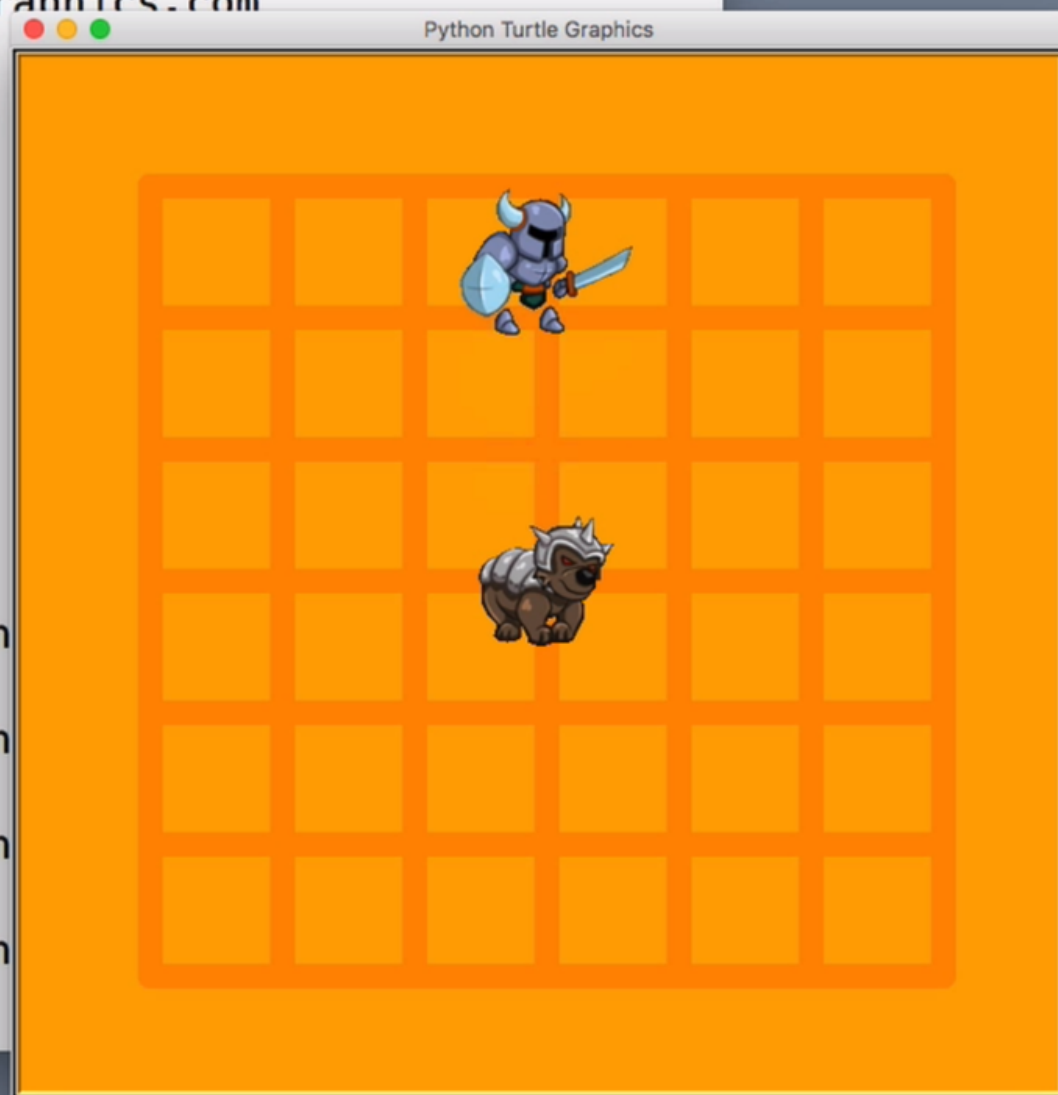
Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□



```
def actor(image):
    t = turtle.Turtle()
    turtle.register_shape(image)
    t.shape(image)
    t.up()
    return t

beast = actor('beast.gif')
player = actor('player.gif')

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)

def move_beast():
    x = random.randint(-GRID_SPAN, GRID_SPAN)
    y = random.randint(-GRID_SPAN, GRID_SPAN)
    beast.goto(x*GRID_SIZE, y*GRID_SIZE)
```

-- INSERT --



```
bash-3.2$ exit
```

```
Press ENTER or type command to continue
```

```
Press ENTER or type command to continue
```

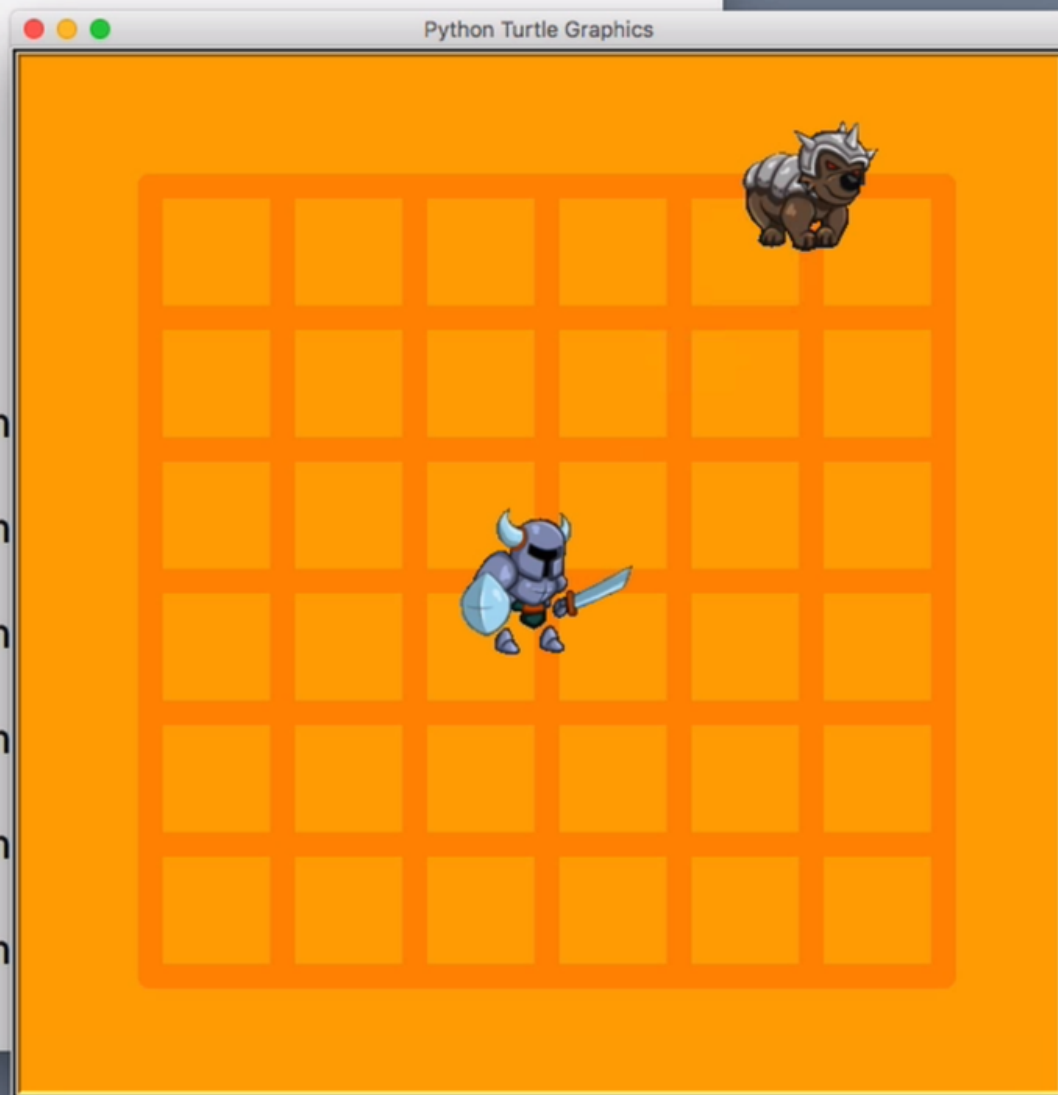
```
Press ENTER or type command to continue
```

```
Press ENTER or type command to continue
```

```
Press ENTER or type command to continue
```

```
Press ENTER or type command to continue
```

```
█
```



```
bash-3.2$ exit
```

Press ENTER or type command to continue

Press ENTER or type command to continue

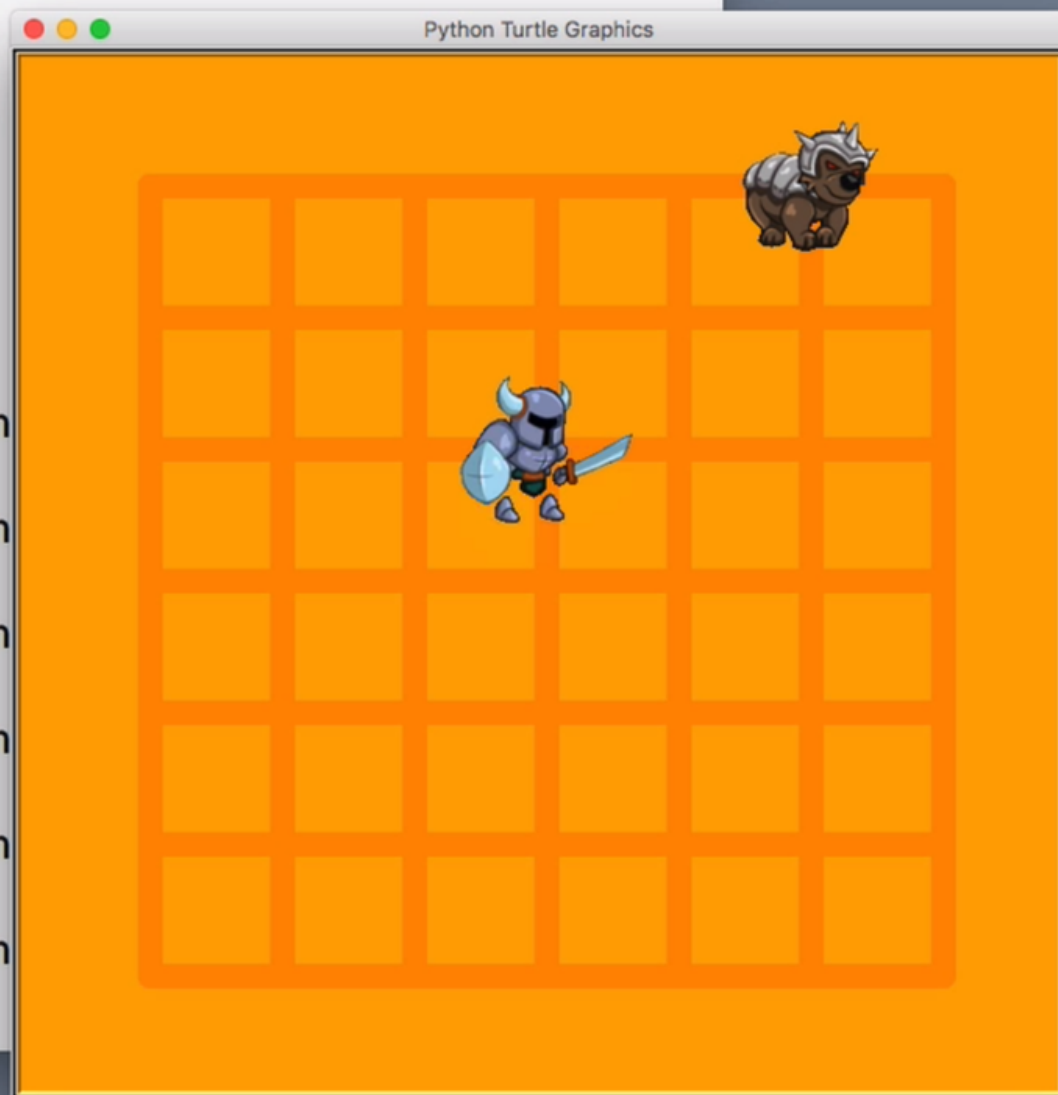
Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□



```
t = turtle.Turtle()
turtle.register_shape(image)
t.shape(image)
t.up()
return t
```

```
beast = actor('beast.gif')
player = actor('player.gif')
```

```
def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
    attempt_capture()
```

```
def attempt_capture():
    if player.position() == beast.position():
        beast.circle(20, steps=7)
        move_beast()
```

"game.py" 63L, 1390C written

```
player = actor('player.gif')

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
    attempt_capture()

def attempt_capture():
    if player.position() == beast.position():
        beast.circle(20, steps=7)
        move_beast()
    else:
        if random.random() < MOVE_BEAST_ODDS:
            move_beast()

def move_beast():
    x = random.randint(-GRID_SPAN, GRID_SPAN)
    y = random.randint(-GRID_SPAN, GRID_SPAN)
    beast.goto(x*GRID_SIZE, y*GRID_SIZE)
```

"game.py" 66L, 1471C written

```
#!/usr/local/bin/python3.7
```

```
import random  
import turtle
```

```
turtle.Screen().setup(startx=600, width=640, height=640)  
turtle.bgcolor('orange')
```

```
GRID_SIZE = 80  
GRID_SPAN = 3  
GRID_MAX = GRID_SIZE * GRID_SPAN
```

```
MOVE_BEAST_ODDS = 0.2
```

```
def line(x1, y1, x2, y2):  
    turtle.up()  
    turtle.goto(x1, y1)  
    turtle.down()  
    turtle.goto(x2, y2)
```

```
bash-3.2$ exit
```

```
Press ENTER or type command to contin
```

```
Press ENTER or type command to contin
```

```
Press ENTER or type command to contin
```

```
Press ENTER or type command to contin
```

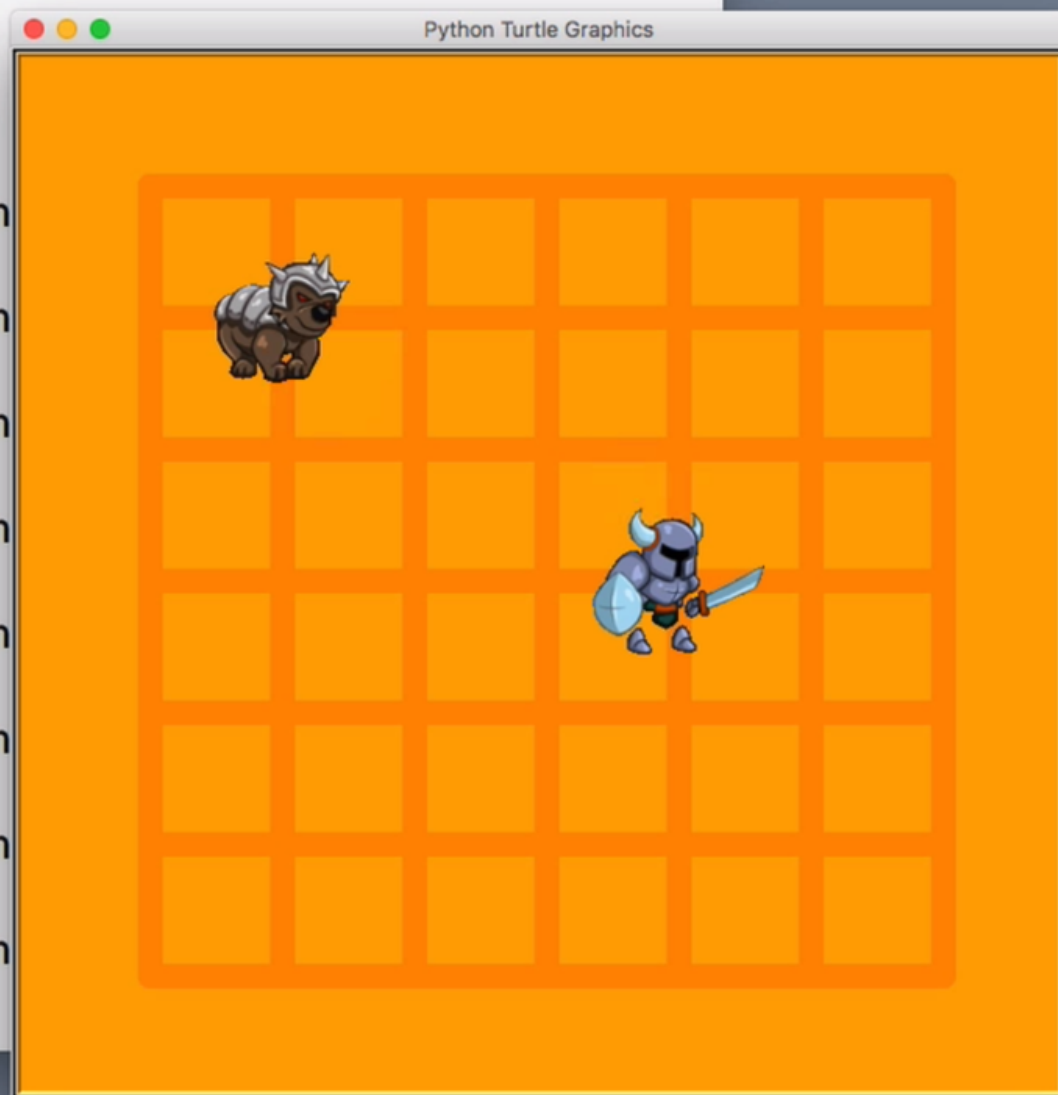
```
Press ENTER or type command to contin
```

```
Press ENTER or type command to contin
```

```
Press ENTER or type command to contin
```

```
Press ENTER or type command to contin
```

```
█
```



```
def actor(image):
    t = turtle.Turtle()
    if image.endswith('.gif'):
        turtle.register_shape(image)
    t.shape(image)
    t.up()
    return t

beast = actor('beast.gif')
player = actor('player.gif')
karma = actor('circle')

def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
    attempt_capture()

def attempt_capture():
```



Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

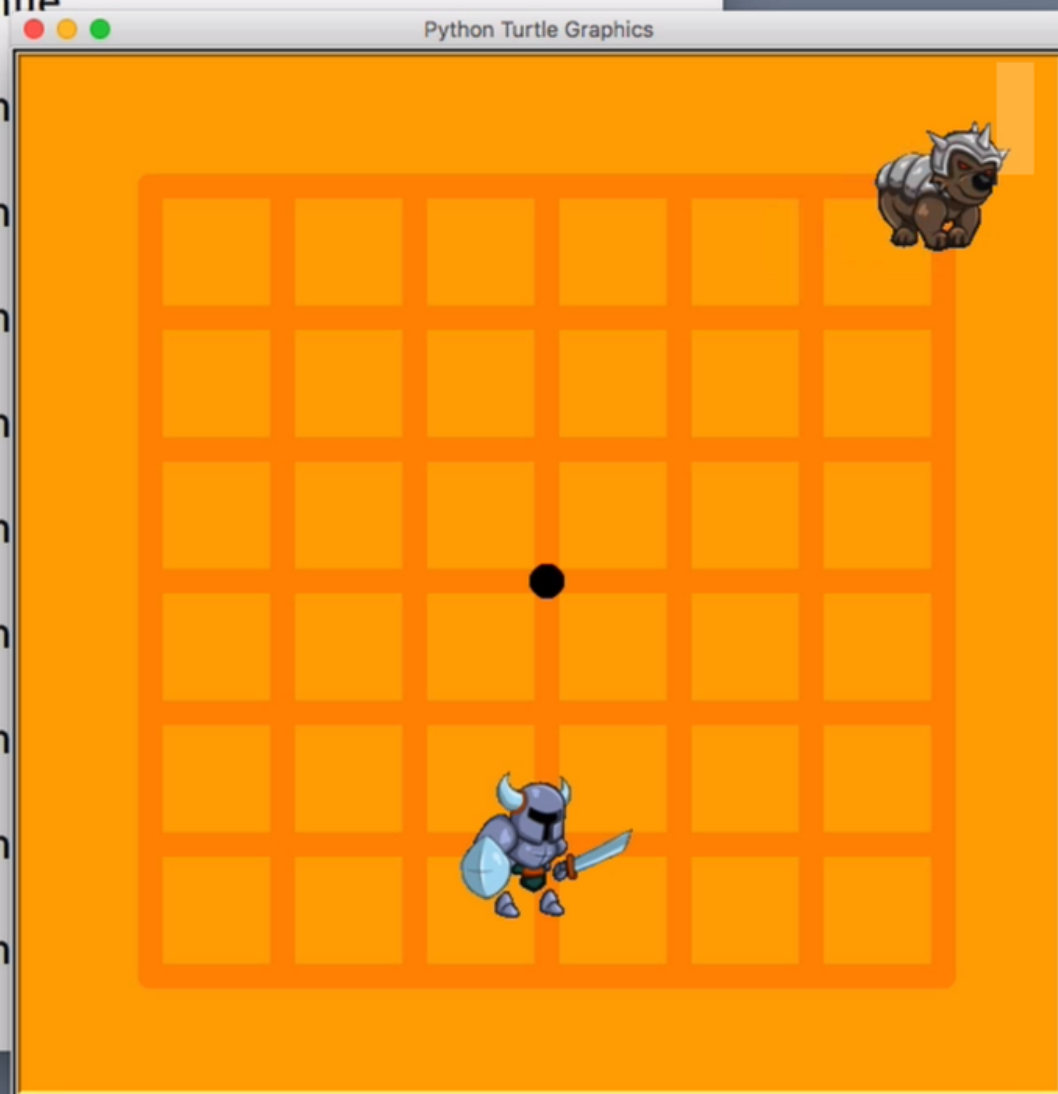
Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□





```
def actor(image, *, color='black'):
    t = turtle.Turtle()
    if image.endswith('.gif'):
        turtle.register_shape(image)
    t.shape(image)
    t.color(color)
    t.up()
    return t
```

```
beast = actor('beast.gif')
player = actor('player.gif')
karma = actor('circle', color='gold')
karma.goto(0, GRID_MAX+42)
```

```
def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
    attempt_capture()
-- INSERT --
```

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

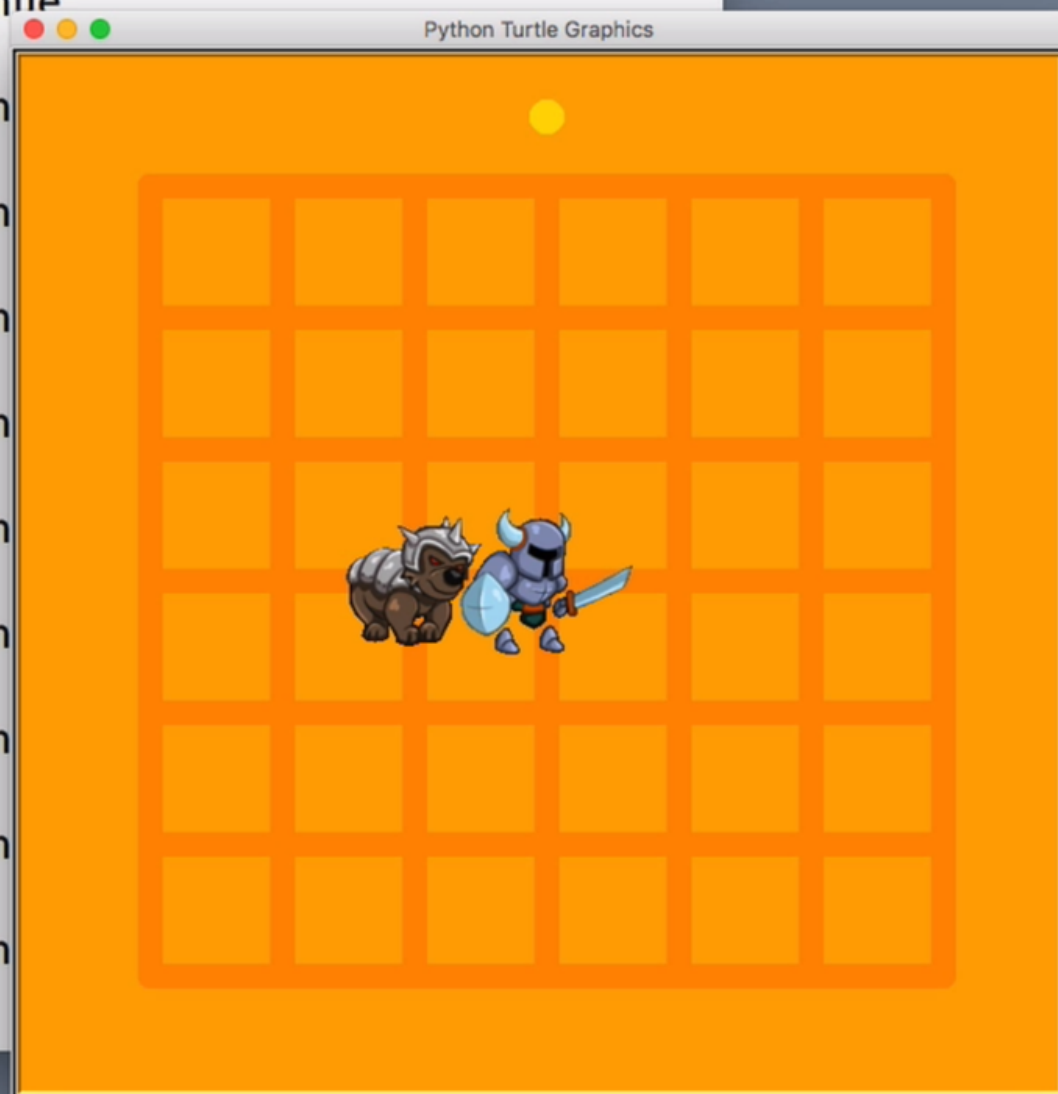
Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□



```
player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
attempt_capture()
```

```
def attempt_capture():
    if player.position() == beast.position():
        beast.circle(20, steps=7)
        update_karma(KARMA_CAPTURE)
        move_beast()
    else:
        update_karma(KARMA_MOVE)
        if random.random() < MOVE_BEAST_ODDS:
            move_beast()
```

```
def update_karma(dk):
    karma.forward(dk)
```

```
def move_beast():
    x = random.randint(-GRID_SPAN, GRID_SPAN)
    y = random.randint(-GRID_SPAN, GRID_SPAN)
```

```
#!/usr/local/bin/python3.7

import random
import turtle

turtle.Screen().setup(startx=600, width=640, height=640)
turtle.bgcolor('orange')

GRID_SIZE = 80
GRID_SPAN = 3
GRID_MAX = GRID_SIZE * GRID_SPAN

MOVE_BEAST_ODDS = 0.2
KARMA_MOVE = -20
KARMA_CAPTURE = 100

def line(x1, y1, x2, y2):
    turtle.up()
    turtle.goto(x1, y1)
"game.py" 79L, 1782C written
```

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

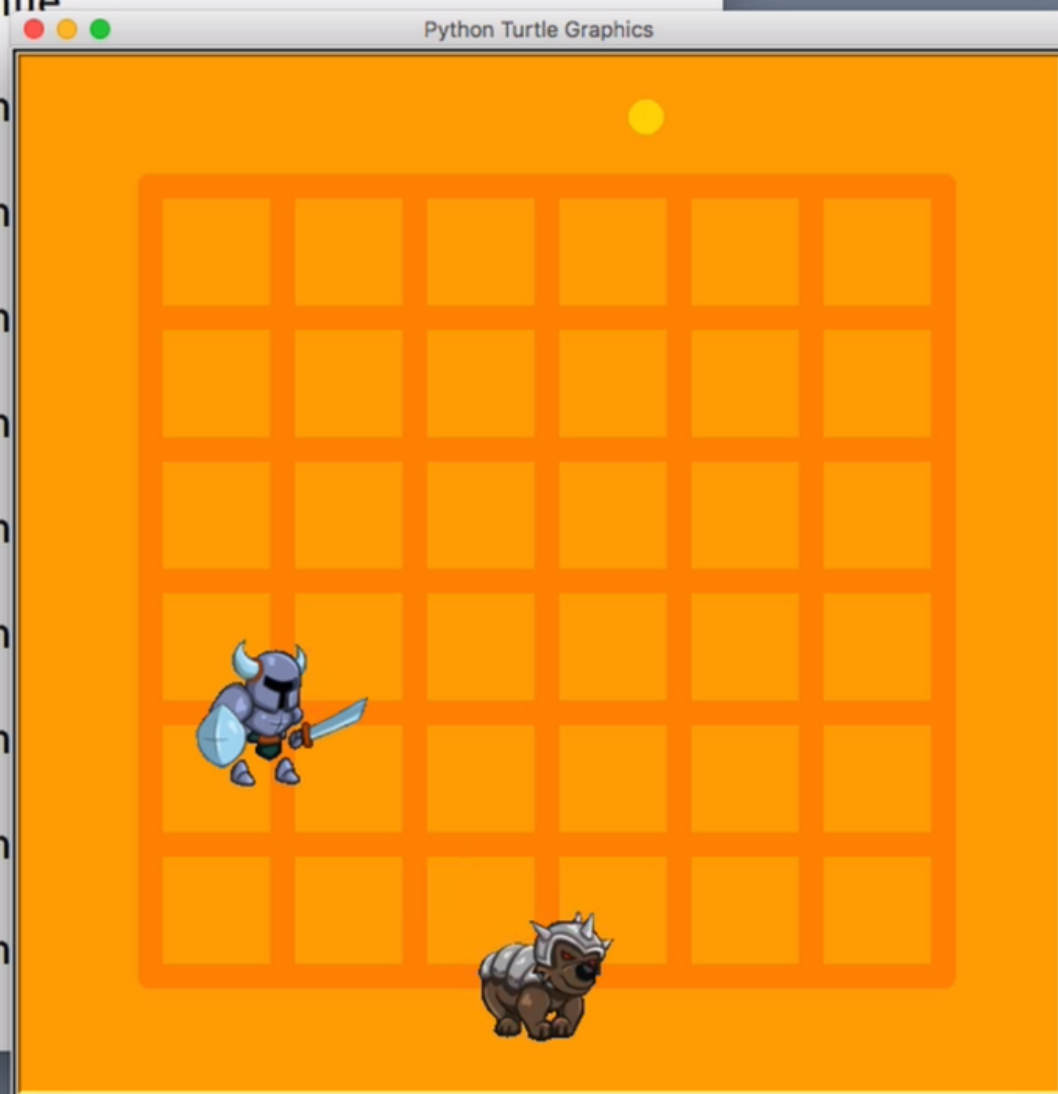
Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□



```
    if random.random() < MOVE_BEAST_ODDS:  
        move_beast()
```

```
def update_karma(dk):  
    karma.forward(dk)  
    karma_x, _ = karma.position()  
    if karma_x >= GRID_MAX:  
        end_game('Victory!')  
    elif karma_x <= -GRID_MAX:  
        end_game('Defeat...')  
  
def end_game(message):  
    player.write(  
        message,  
        align='center',  
        font=('Helvetica', 64, 'bold'),  
    )
```

```
def move_beast():
```



Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□



Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□

Python Turtle Graphics

**Victory!**

**Victory!**

**Victory!**

**Victory!**





```
def end_game(message):  
    player.write(  
        message,  
        align='center',  
        font=('Helvetica', 64, 'bold'),  
    )  
    for key in ('Up', 'Down', 'Left', 'Right'):  
        turtle.onkey(None, key)
```

```
def move_beast():  
    x = random.randint(-GRID_SPAN, GRID_SPAN)  
    y = random.randint(-GRID_SPAN, GRID_SPAN)  
    beast.goto(x*GRID_SIZE, y*GRID_SIZE)
```

```
turtle.listen()
```

```
turtle.onkey(lambda: move_player(0, 1), 'Up')  
turtle.onkey(lambda: move_player(0, -1), 'Down')  
turtle.onkey(lambda: move_player(-1, 0), 'Left')  
"game.py" 93L, 2148C written
```

```
t = turtle.Turtle()
if image.endswith('.gif'):
    turtle.register_shape(image)
t.shape(image)
t.color(color)
t.up()
return t
```

```
beast = actor('beast.gif')
player = actor('player.gif', color='white')
karma = actor('circle', color='gold')
karma.goto(0, GRID_MAX+42)
```

```
def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
    attempt_capture()
```

```
def attempt_capture():
```

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

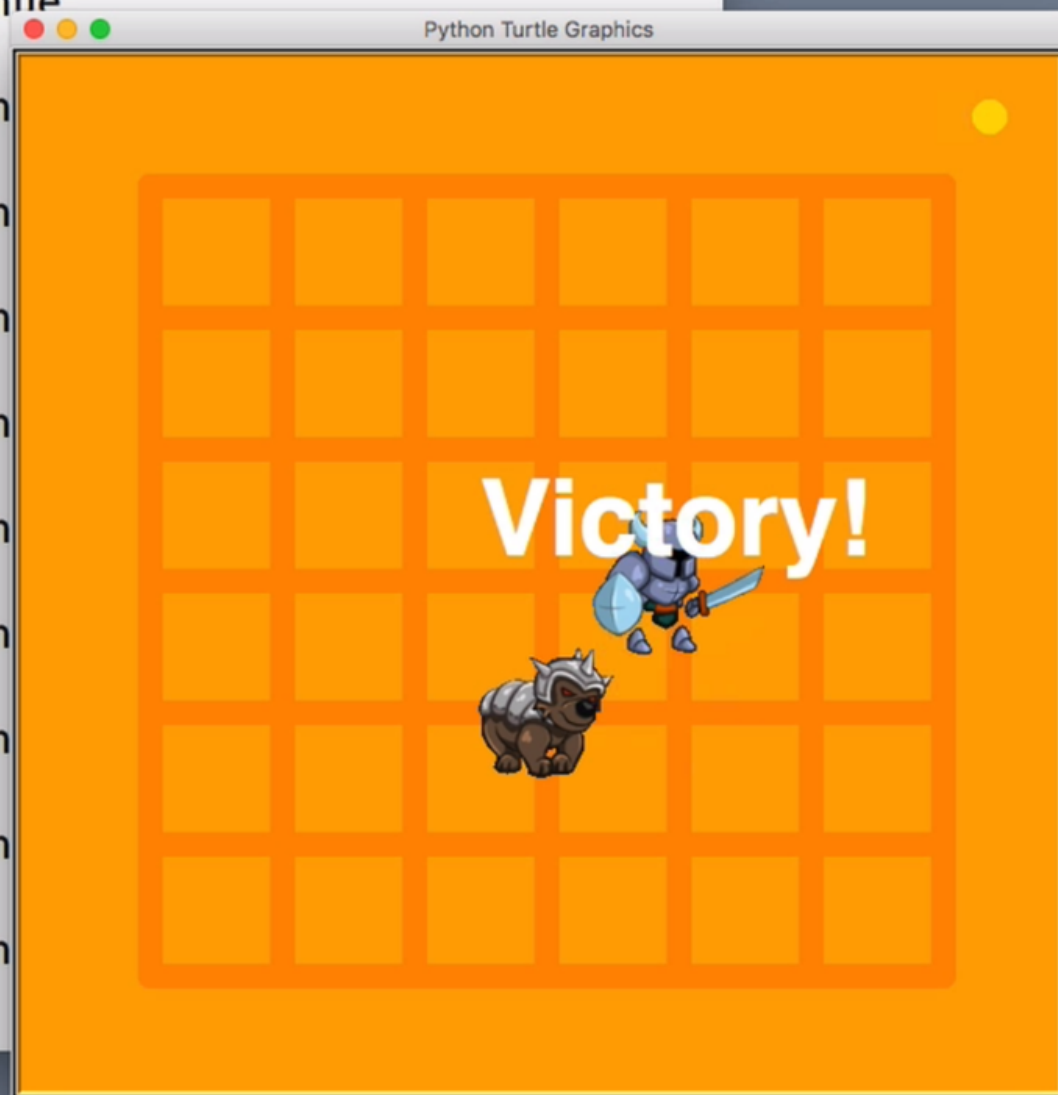
Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□



Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

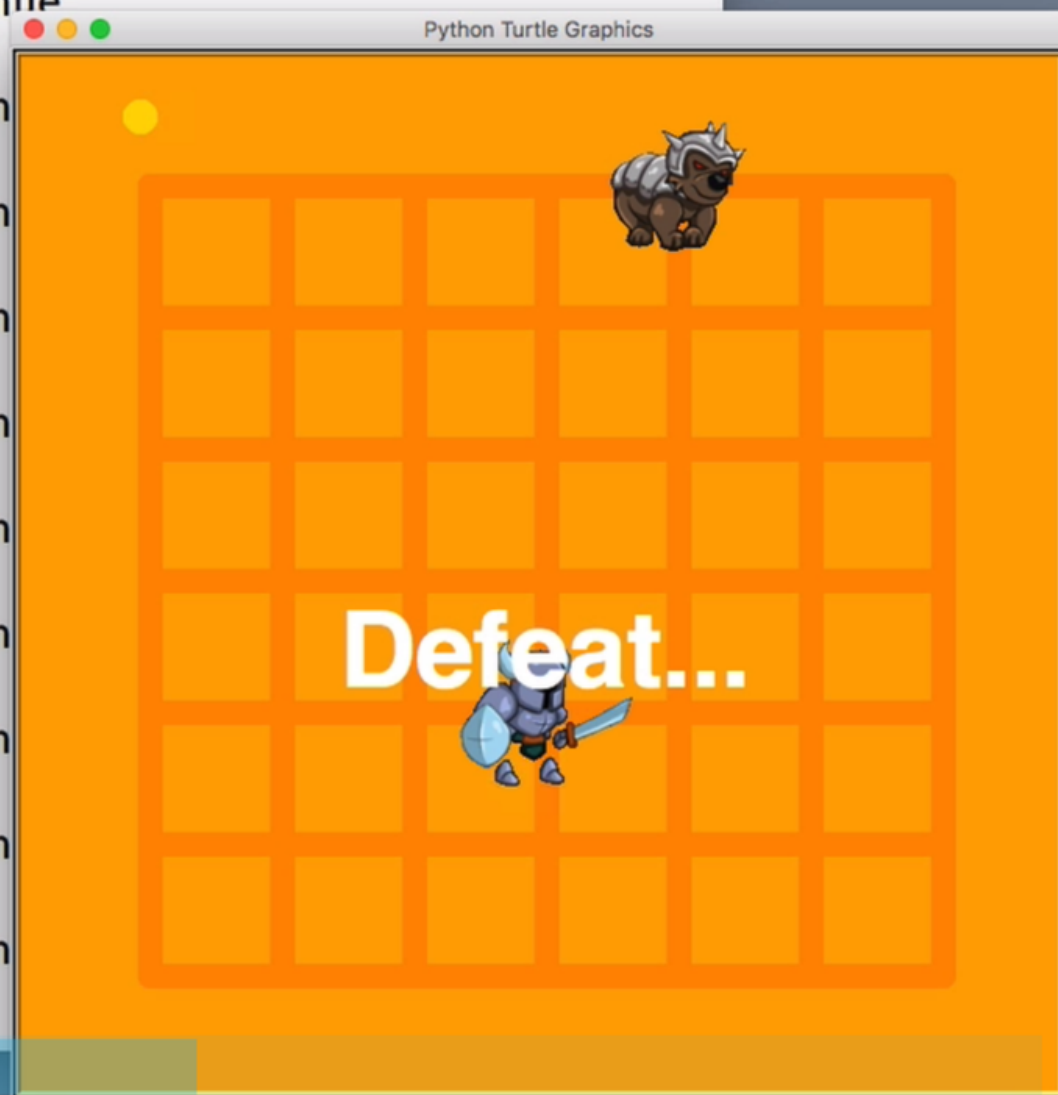
Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□



```
t = turtle.Turtle()
if image.endswith('.gif'):
    turtle.register_shape(image)
t.shape(image)
t.color(color)
t.up()
return t
```

```
beast = actor('beast.gif')
player = actor('player.gif', color='white')
player.steps = 0
karma = actor('circle', color='gold')
karma.goto(0, GRID_MAX+42)
```

```
def move_player(dx, dy):
    x, y = player.position()
    player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
    player.steps += 1
    attempt_capture()
```

"game.py" 95L, 2203C written

```
def update_karma(dk):
    karma.forward(dk)
    karma_x, _ = karma.position()
    if karma_x >= GRID_MAX:
        end_game('Victory!')
    elif karma_x <= -GRID_MAX:
        end_game('Defeat...')

def end_game(message):
    player.write(
        message,
        align='center',
        font=('Helvetica', 64, 'bold'),
    )
    karma.write(
        f'{player.steps} steps',
        align='center',
        font=('Helvetica', 32, 'bold'),
    )
```

"game.py" 100L, 2323C written



Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

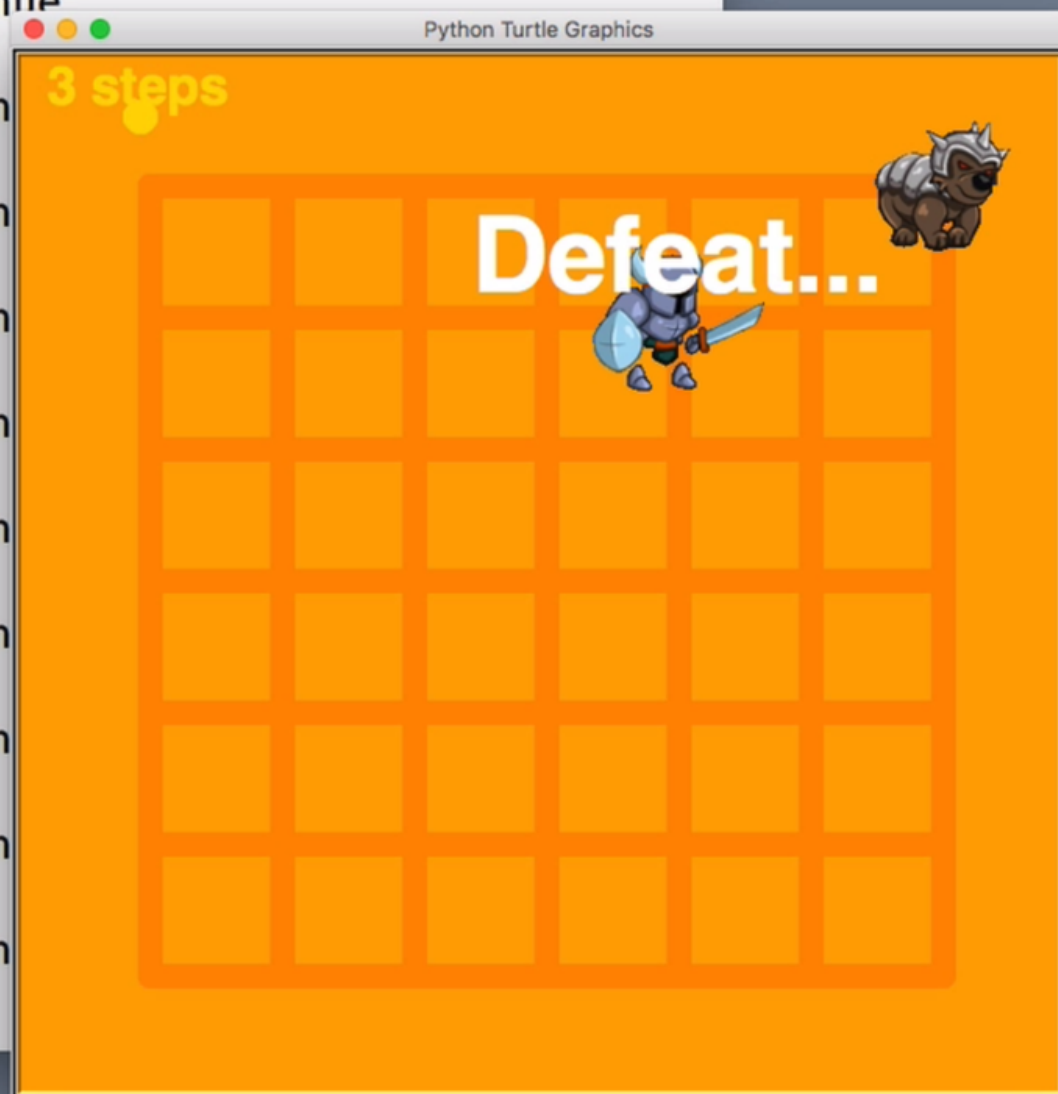
Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□



Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

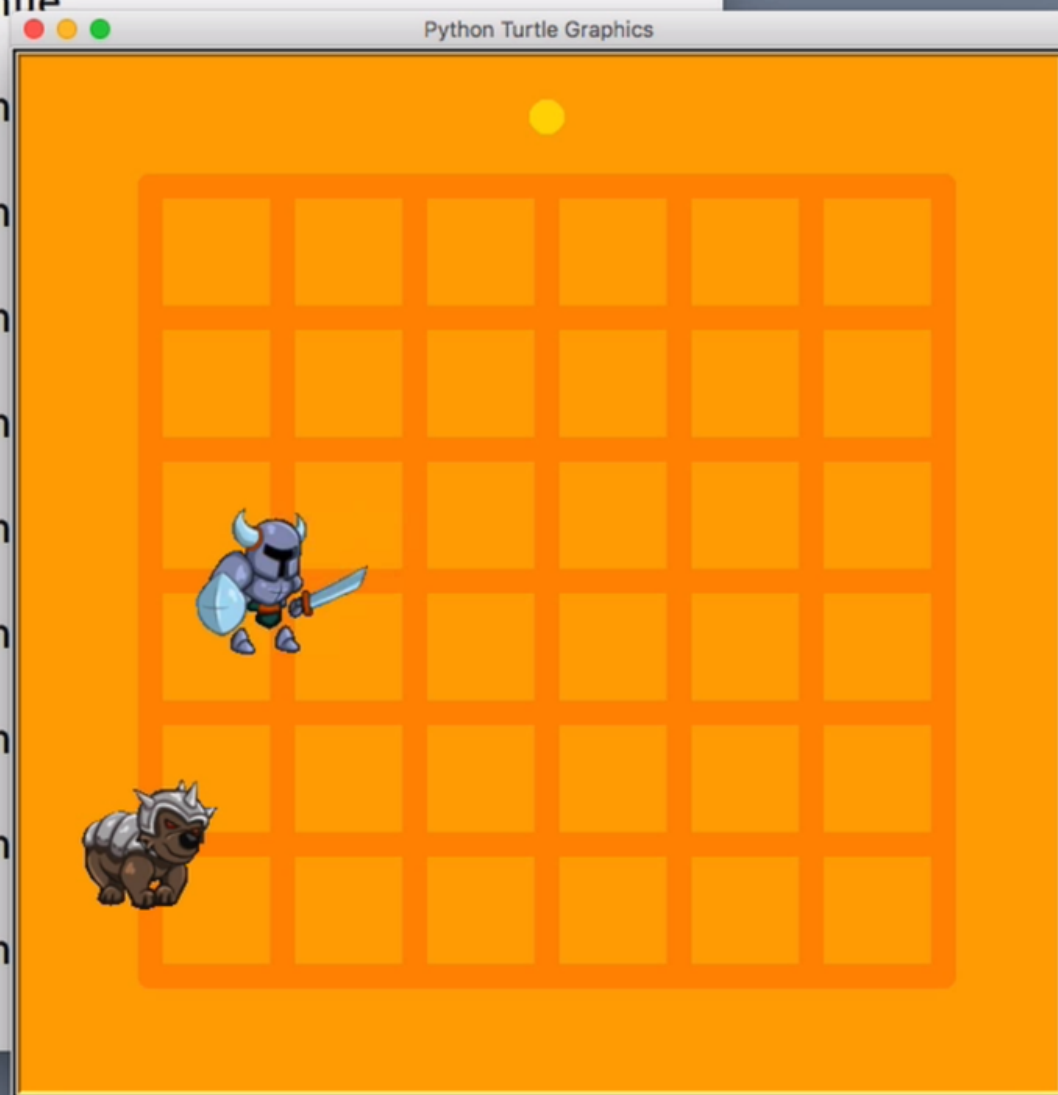
Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□





```
82     )
83     for key in ('Up', 'Down', 'Left', 'Right'):
84         turtle.onkey(None, key)
85
86 def move_beast():
87     x = random.randint(-GRID_SPAN, GRID_SPAN)
88     y = random.randint(-GRID_SPAN, GRID_SPAN)
89     beast.goto(x*GRID_SIZE, y*GRID_SIZE)
90
91 turtle.listen()
92
93 turtle.onkey(lambda: move_player(0, 1), 'Up')
94 turtle.onkey(lambda: move_player(0, -1), 'Down')
95 turtle.onkey(lambda: move_player(-1, 0), 'Left')
96 turtle.onkey(lambda: move_player(1, 0), 'Right')
97
98 move_beast()
99
100 turtle.mainloop()
```

```
1 #!/usr/local/bin/python3.7
2
3 import random
4 import turtle
5
6 turtle.Screen().setup(startx=600, width=640, height=640)
7 turtle.bgcolor('orange')
8
9 GRID_SIZE = 50
10 GRID_SPAN = 5
11 GRID_MAX = GRID_SIZE * GRID_SPAN
12
13 MOVE_BEAST_ODDS = 0.2
14 KARMA_MOVE = -20
15 KARMA_CAPTURE = 100
16
17 def line(x1, y1, x2, y2):
18     turtle.up()
19     turtle.goto(x1, y1)
```

1 change; before #177 3 seconds ago

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

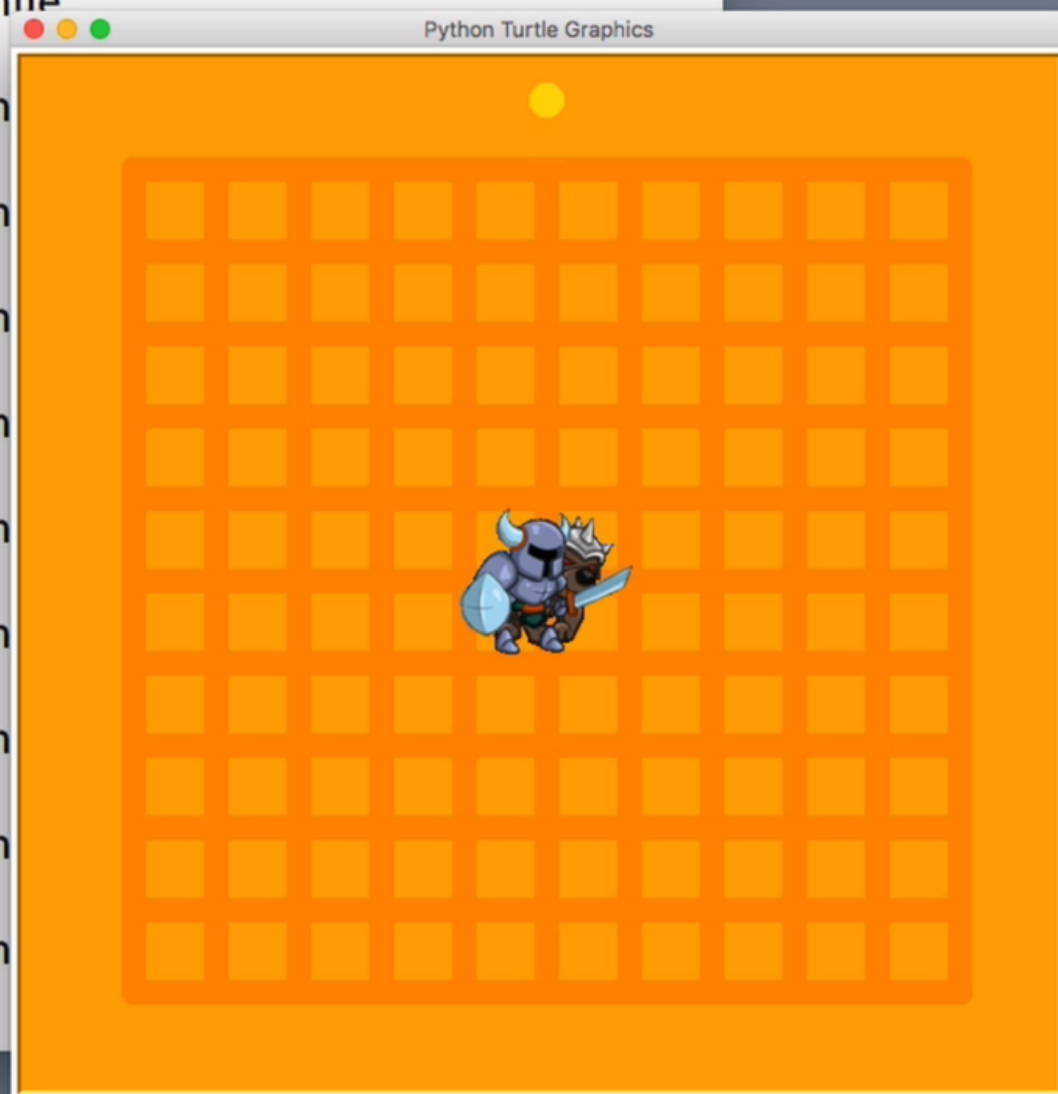
Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□



Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

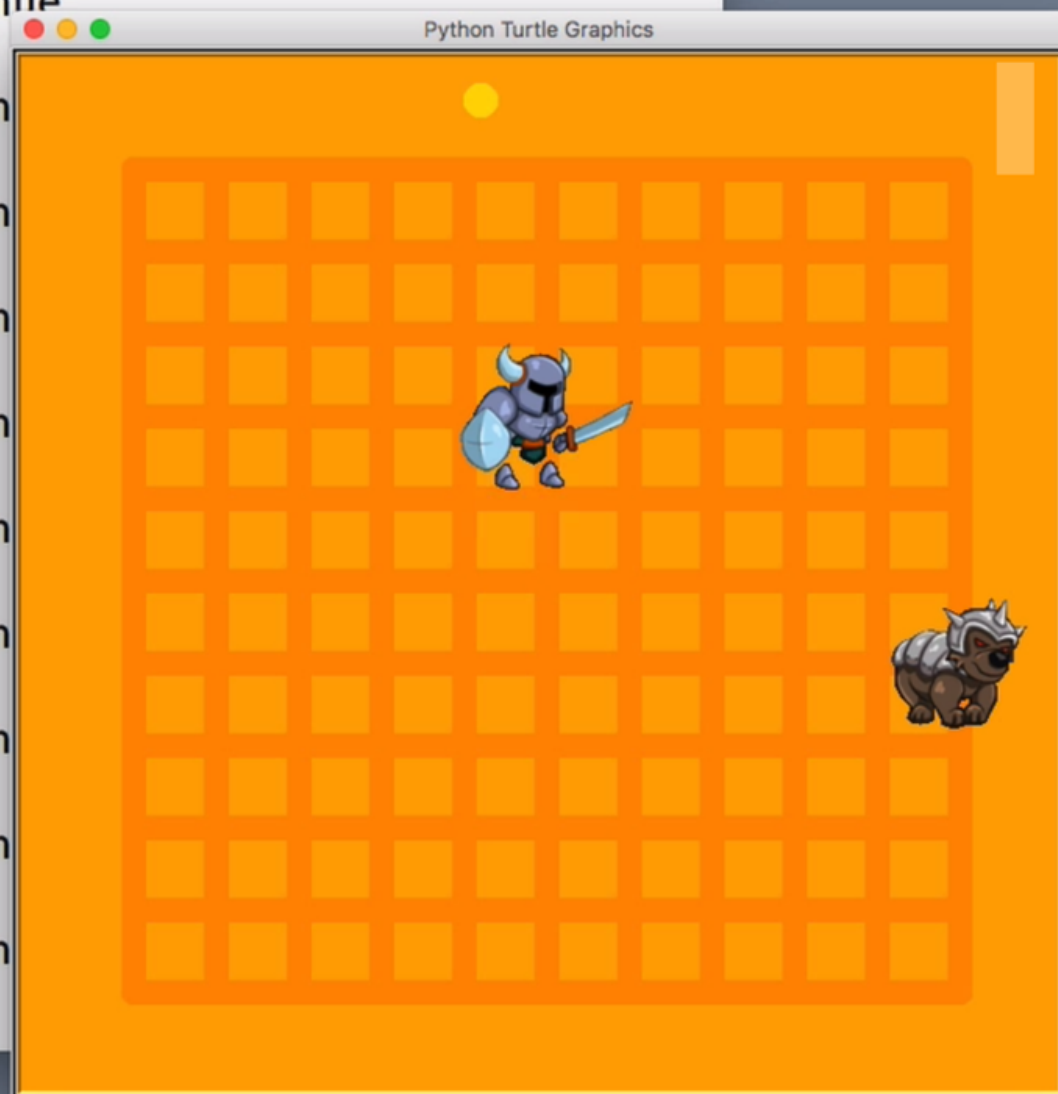
Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

Press ENTER or type command to continue

□



```
23 turtle.speed(0)
24 turtle.hideturtle()
25 turtle.color('dark orange')
26 turtle.width(15)
27
28 for i in range(-GRID_SPAN, GRID_SPAN+1):
29     scaled_i = i * GRID_SIZE
30     line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
31     line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)
32
33 def actor(image, *, color='black'):
34     t = turtle.Turtle()
35     if image.endswith('.gif'):
36         turtle.register_shape(image)
37     t.shape(image)
38     t.color(color)
39     t.up()
40     return t
41
```

-- VISUAL LINE --

```
30     line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
31     line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)
32
33 def actor(image, *, color='black'):
34     t = turtle.Turtle()
35
36
37
38
39     if image.endswith('.gif'):
40
41
42
43         if image[-4:] == '.gif':
44
45
46
47
48             turtle.register_shape(image)
```

```
30     line(-GRID_MAX, scaled_i, GRID_MAX, scaled_i)
31     line(scaled_i, -GRID_MAX, scaled_i, GRID_MAX)
32
33 def actor(image, *, color='black'):
34     t = turtle.Turtle()
35
36
37
38
39     if image.endswith('.jpeg'):
40
41
42
43         if image[-4:] == '.jpeg':
44
45
46
47
48             turtle.register_shape(image)
```



```
35     if image.endswith('.gif'):
36         turtle.register_shape(image)
37     t.shape(image)
38     t.color(color)
39     t.up()
40     return t
41
42     beast = actor('beast.gif')
43     player = actor('player.gif', color='white')
44     player.steps = 0
45     karma = actor('circle', color='gold')
46     karma.goto(0, GRID_MAX+42)
47
48     def move_player(dx, dy):
49         x, y = player.position()
50         player.goto(x + dx*GRID_SIZE, y + dy*GRID_SIZE)
51         player.steps += 1
52         attempt_capture()
53
```

-- VISUAL LINE --



```
62         move_beast()
63
64 def update_karma(dk):
65     karma.forward(dk)
66
67
68     karma_x, _ = karma.position()
69
70
71
72     karma_x = karma.position()[0]
73
74
75
76
77
78     if karma_x >= GRID_MAX:
79         end_game('Victory!')
80     elif karma_x <= -GRID_MAX:
```

"game.py" 111L, 2361C written

```
62         move_beast()
63
64 def update_karma(dk):
65     karma.forward(dk)
66
67
68     karma_x, _ = karma.position()
69     # rhs is iterable (think list-like)
70     # two elements
71     # we care about the first
72
73
74     karma_x = karma.position()[0]
75     # [42]?
76     # ['key']?...
77
78
79
80
```

```
81         font=('Helvetica', 32, 'bold'),
82     )
83     for key in ('Up', 'Down', 'Left', 'Right'):
84         turtle.onkey(None, key)
85
86 def move_beast():
87     x = random.randint(-GRID_SPAN, GRID_SPAN)
88     y = random.randint(-GRID_SPAN, GRID_SPAN)
89     beast.goto(x*GRID_SIZE, y*GRID_SIZE)
90
91 turtle.listen()
92
93 turtle.onkey(lambda: move_player(0, 1), 'Up')
94 turtle.onkey(lambda: move_player(0, -1), 'Down')
95 turtle.onkey(lambda: move_player(-1, 0), 'Left')
96 turtle.onkey(lambda: move_player(1, 0), 'Right')
97
98 move_beast()
99
```

-- VISUAL LINE --

## Thoughts

- Python's Standard Library often surprises you.
- Direct, in-your-face Python is effective and fun.
- Having fun is one great way of learning new things.
- The kids did learn something.

Don't do this at work?

-----

Extending the game:

- Distributed network-based play?
- Create an event loop for async turtles?
- Learn about, and fix race-conditions?
- Have an AI/ML-powered beast?

What could we learn?

Learning

Growing

Having Fun

\$

