Python Games

with Pygame Zero

FRAZER-NASH CONSULTANCY

W. H. Bell
http://www.whbell.net/

http://www.fnc.co.uk

Raspberry Pi Day
University of Strathclyde
12/12/2015

Outline

- Teaching with Python
- Writing games in Python
- Pictures and directory structure
- Controlling a sprite
- Handling collisions
- Using the timer functions

Teaching with Python

- Scratch provides a great environment to learn the process of building a program.
  - Most programming languages are typed in and require careful debugging.
- Python is a simple language that provides the next step in complexity.
  - Can encourage programming using hardware examples and by writing simple games.

Writing games in Python

- Pygame provides a diverse set of tools for creating games. http://www.pygame.org/
  - Can use SDL and OpenGL graphics libraries, for two and three dimensional graphics respectively.
  - However, to write a simple game a lot of Pygame specific boilerplate has to be written.
- Pygame Zero provides a layer around Pygame, to allow simple games to be written quickly.
  - Can be used to encourage students to get off the ground, without having to worry about the details of Pygame API.

https://pygame-zero.readthedocs.org/en/latest/index.html

Getting started

- Create png sprite images using Gimp or a similar graphics editor.
  - Then copy them into an images/ directory, in the same directory as the games.

```
python/images/rock_destroyed.png
python/images/rock.png
python/images/spacecraft_destroyed.png
python/images/spacecraft.png
python/laser.py
python/oneRock.py
python/spacecraft.py
```

Controlling a spacecraft

```
WIDTH = 600
HEIGHT = 500
```

Set the size of the screen

```
spacecraft = Actor('spacecraft')
spacecraft.pos = WIDTH/2, HEIGHT/2
```

Create a sprite, using spacecraft.png
Set it to be in the middle of the screen

```
def draw():
  screen.clear()
  spacecraft.draw()
```

Clear the screen and draw the spacecraft
draw() is only called when needed. (c.f. paint() in Java)

```
def update():
  if keyboard.left and spacecraft.left > 2:
    spacecraft.x -= 2
  if keyboard.right and spacecraft.right < WIDTH+2:
    spacecraft.x += 2
  if keyboard.down and spacecraft.bottom < HEIGHT+2:
    spacecraft.y += 2
  if keyboard.up and spacecraft.top > 2:
    spacecraft.y -= 2
```

The update() function is called with a default frequency.
This version moves the spacecraft around, but does not allow it to leave the screen.

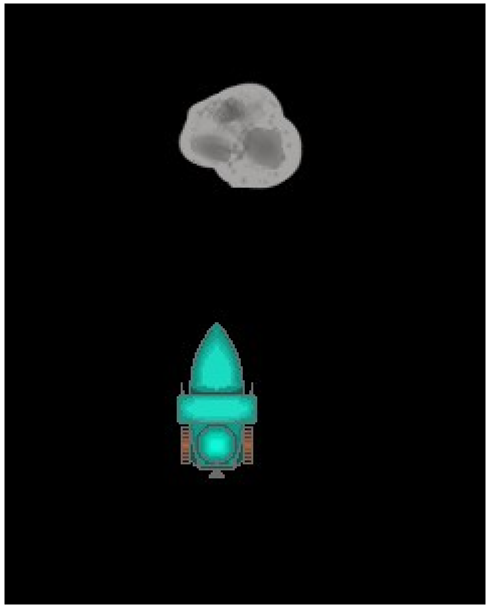
```
git clone https://github.com/williamhbell/PygameZero.git
pgzrun PygameZero/python/spacecraft.py
```

Adding a rock

Collisions are handled using pygame.Rect.colliderect()

```
import random
WIDTH = 600
HEIGHT = 500
spacecraft = Actor('spacecraft')
rock = Actor('rock')
centre_x = WIDTH/2
centre_y = HEIGHT/2

def startingPosition():
  return (random.randint(0, WIDTH), 0)

def startingVelocity():
  return (random.randint(-5, 5), random.randint(1, 5))

def update():
  global gameOver
  if not gameOver:
    updateSpacecraft()
    updateRock()
  else:
    if keyboard.space:
      initialPositions()
      gameOver = False
```

```
collision = spacecraft.colliderect(rock)

if collision:
  rock.image = 'rock_destroyed'
  spacecraft.image = 'spacecraft_destroyed'
  gameOver = True
```



```
git clone https://github.com/williamhbell/PygameZero.git
pgzrun PygameZero/python/oneRock.py
```

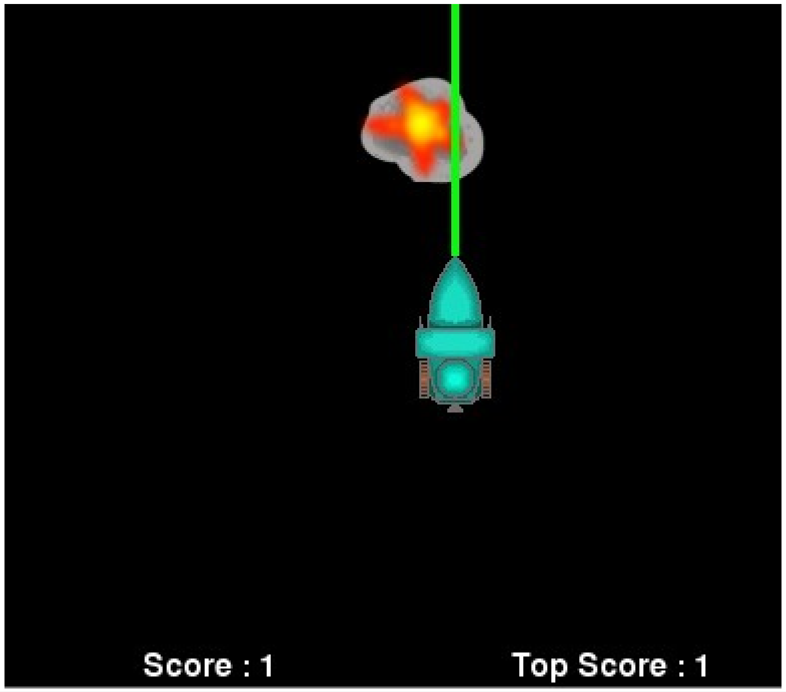
Adding a laser

Events can be scheduled using the internal clock.

```
def laserFiringComplete():
  global laserFiring
  laserFiring = False
  clock.schedule(laserChargingComplete, 1.0)

def laserChargingComplete():
  global laserCharged
  laserCharged = True

def updateSpacecraft():
  global laserFiring
  global laserCharged
  global laser
  if keyboard.left and spacecraft.left > 2:
    spacecraft.x -= 2
  if keyboard.right and spacecraft.right < WIDTH+2:
    spacecraft.x += 2
  if keyboard.down and spacecraft.bottom < HEIGHT+2:
    spacecraft.y += 2
  if keyboard.up and spacecraft.top > 2:
    spacecraft.y -= 2
```

```
  if keyboard.space and laserCharged:
    laserCharged = False
    laserFiring = True
    clock.schedule(laserFiringComplete, 0.3)

  if laserFiring:
    laser = Rect((spacecraft.x-2,0),
(4,spacecraft.top))
```

```
git clone https://github.com/williamhbell/PygameZero.git
pgzrun PygameZero/python/laser.py
```


Score : 1
Top Score : 1