GETTING STARTED WITH PYGAME ON THE RASPBERRY PI

RASPBERRY JAM

MILTON KEYNES

Worksheet
And
Cheat Sheet

TechnoVisual education

www.technovisualeducation.co.uk

1. SETTING UP YOUR GAME LOOP

RASPBERRY JAM MILTON KEYNES

1. First open the Python 3 editor (IDLE) and make a new python file. Add import instructions to your file to load in the pygame and sys modules. Import the locals from pygame.

```
import pygame, sys
from pygame.locals import *
```

2. Initialise the pygame module and create a new pygame window, setting the display mode of the window. Then give it a caption.

```
pygame.init()
DISPLAYSURF = pygame.display.set_mode((400, 300))
pygame.display.set_caption('Pygame Number 1!')
```

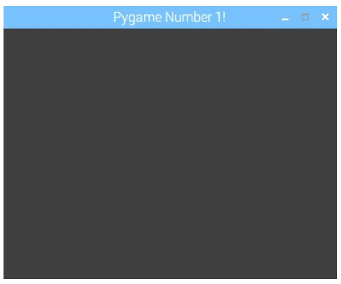
3. Create a game loop that will continue running throughout the game and check for events.

```
while True: # main game loop
    for event in pygame.event.get():
```

4. Check the event type for exit events like the close icon being clicked on the pygame window.

```
          if event.type == QUIT:
               pygame.quit()
               sys.exit()
```

5. Now to test your code. Save your code and then select 'Run' from the menu bar or press F5. You should see a window appear. If you click the close icon of the window, it should close and the programme exits.



Next: Defining Game Elements With Data

2. DEFINING GAME ELEMENTS WITH DATA

RASPBERRY JAM
MILTON KEYNES

1. To store the position of an alien ship define x and y co-ordinates by setting them to a starting value. Write this code near the top of your programme.

```
# Data
alienX = 200
alienY = 20
```

2. Define some constants that we can use for setting colours.

```
BLACK = (  0,   0,   0)
WHITE = (255, 255, 255)
RED   = (255,   0,   0)
GREEN = (  0, 255,   0)
BLUE  = (  0,   0, 255)
```

3. Now we can draw the alien ship onto the screen at the co-ordinates we have defined. Make a function called drawScreen and add a call to it in the game loop.

```
# This goes in the game loop
# Draw the screen
     drawScreen()
```

```
# This goes above the game loop
def drawScreen():
     DISPLAYSURF.fill(BLACK)
     drawAlien()
     pygame.display.update()
     return
```

```
# This goes above the drawScreen function
def drawAlien():
     x = alienX
     y = alienY
     points = [(x,y+10),(x-10,y-10),(x+10,y-10)]
     pygame.draw.lines(DISPLAYSURF,RED,True,points,1)
     return
```

Next: Animating the Alien Ship

3. ANIMATING THE ALIEN SHIP

RASPBERRY JAM MILTON KEYNES

1. Create a function to move the alien ship left and right.

```
def moveAlien():
     global alienDir, alienX, alienY
     if alienDir == 0:
          alienX = alienX - 1
          if alienX == 0:
               alienDir = 1
     if alienDir == 1:
          alienX = alienX + 1
          if alienX == 400:
               alienDir = 0
     return
```

2. We are using a new global variable called alienDir so we need to define that at the top of the code with our other data.

```
alienDir = 0
```

3. We also need to call the new moveAlien function from our game loop.

```
# Move the alien
     moveAlien()
```

4. If you try running the code, the alien ship will probably move too fast so we can slow down the animation like this:

```
# Put this before the game loop
gameclock = pygame.time.Clock()
```

```
# Put this just inside the game loop
     gameclock.tick(20)
```

Next: Adding A Player Ship

4. ADDING A PLAYER SHIP

RASPBERRY JAM
MILTON KEYNES

1. Add some global variables at the top of your code to hold the x and y co-ordinates of the player ship.

```
shipX = 200
shipY = 280
```

2. Define a function above your game loop to draw the player ship.

```
def drawShip():
     x = shipX
     y = shipY
     points = [(x,y-10),(x-10,y+10),(x+10,y+10)]
     pygame.draw.lines(DISPLAYSURF,GREEN,True,points,1)
     return
```

3. Then add a call to the drawShip function in the drawScreen function.

```
def drawScreen():
    DISPLAYSURF.fill(BLACK)
    drawAlien()
    drawShip()
    pygame.display.update()
    return
```

4. To listen for key presses we need to define a function called checkKeys and add a call to it from the game loop.

```
def checkKeys():
    global shipX, shipY
    if pygame.key.get_pressed()[pygame.K_LEFT] !=0 :
        shipX = shipX - 3
    if pygame.key.get_pressed()[pygame.K_RIGHT] !=0 :
        shipX = shipX + 3
    if pygame.key.get_pressed()[pygame.K_SPACE] !=0 :
        fireLaser()
    return
```

```
# Check keyboard input inside the game loop
    checkKeys()
```

Next: Make a Laser Element

5. MAKE A LASER ELEMENT

RASPBERRY JAM
MILTON KEYNES

1. Define a function called fireLaser that we called in the checkKeys function. Put this above the checkKeys function.

```
def fireLaser():
    global laserY,laserX
    if laserY <= 0 :
        laserY = shipY
        laserX = shipX
    return
```

2. We have used some new global variables so we need to make sure they are included in the data section at the beginning of the code.

```
laserX = 0
laserY = 0
```

3. Now define a function to draw the laser and add a call to the function in the drawScreen function.

```
def drawLaser():
    global laserX, laserY
    if laserY >= 1 :
        points = [(laserX,laserY+10),(laserX,laserY)]
        pygame.draw.lines(DISPLAYSURF,WHITE,False,points,1)
    return
```

```
# This goes in the drawScreen function
     drawLaser()
```

4. We will also need to define a function to move the laser up the screen when it has been fired and then add a call to it in the game loop.

```
def moveLaser():
    global laserY
    if laserY >=1 :
        laserY = laserY - 5
    return
```

```
# This goes in the game loop
     moveLaser()
```

Next: Check if The Laser Hit the Alien Ship

6. CHECK IF THE LASER HIT THE ALIEN SHIP

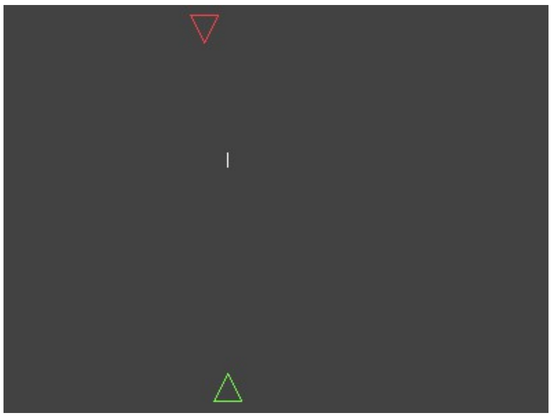
RASPBERRY JAM MILTON KEYNES

1. Define a function to check if the laser co-ordinates are inside the bounding box of the alien ship and add a call to the function in the game loop.

```
def checkLaser():
    global laserX,laserY,alienX,alienY
    if (laserX > alienX-10 and laserX < alienX+10
        and laserY > alienY-10 and laserY < alienY+10) :
        alienY = -20
    return
```

```
# This goes in the game loop
     checkLaser()
```

2. Now test your programme. You should see a moving red triangle and a green triangle at the bottom of the screen. If you use the arrow keys left and right the green (player) ship should move. If you hit the space bar you should see a laser fire from the player ship and move up the screen. If the laser hits the red ship it should disappear.



3. **Next Steps** : Have a go at coding the following items

- Multiple alien ships
- Alien ships shoot back
- Multiple lasers
- Score
- Alien ships move differently
- Use bitmap graphics for the ships
- Make explosion graphic when a ship is hit
- Make sounds when a laser is fired or a ship is hit

Extra resources for this project can be found at:
www.technovisualeducation.co.uk/pygame_workshop

Next: Python Cheat Sheet

PYTHON CHEAT SHEET

RASPBERRY JAM MILTON KEYNES

Python Keywords

def	*Define a function with optional variables.* Example: def fireLaser(laserNumber):
for in	*Make a loop to read an array.* Example: for value in array:
from	*Import a section of a module for use in the programme.* Example: from pygame.locals import *
global	*Enable changing the value of variables that have been defined outside a function.* Example: global x,y
if	*A condition statement. Run the following code if the equation is true.* Example: if laserY < 0:
import	*Import a module so that its functions can be used.* Example: import sys
return	*Return control of the programme back to where the function was called from.* Example: return
while	*Perform a loop until an equation is true.* Example: while (count < 10):

Useful PyGame Functions

pygame.display.update() - *refresh the window to display changes.*
pygame.display.set_caption(title) - *set the title of the window.*
DISPLAYSURFACE = pygame.display.set_mode((width,height)) - *create a pygame window.*
pygame.draw.lines(DISPLAYSURFACE,colour,closed,points,width) - *draw lines between a point list.*
pygame.event.get() - *get a list of events that have happened.*
pygame.init() - *Initialise the pygame module.*
DISPLAYSURFACE.fill(colour) - *fill the window with a colour.*
pygame.key.get_pressed()[key] - *see if a named key is being pressed.*
pygame.time.Clock() and tick(framespersecond) - *set the frame rate for a game loop.*
pygame.quit() - *close the pygame session.*

*For an **introduction to Python** on the Raspberry Pi, Mark Vanstone has produced a '**Getting Started**' series of videos that can be viewed at: http://thepimaker.online/getting-started/*

This resource produced by TechnoVisual Education 2017 – technovisualeducation.co.uk