

Raspberry Pi Intro – Hacking games with Python

TechResort Encounters #5138C_Games



What do I need?

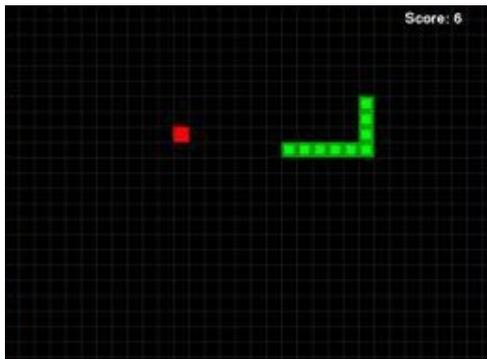
- A Raspberry Pi board and power supply
- An SD card with the operating system installed
- A monitor, HDMI cable and power supply
- A mouse and keyboard



What are we going to use the Raspberry Pi for?

Building on from [script #5138B](#), we're going to continue programming in Python to look at how games are made, and how we can customise them to our choosing.

As you've done in the previous sessions, power up your Pi. Remember to plug the power supply in last.



Last time, we had a look at the **wormy.py** file and added a difficulty selection system. Open up this file and familiarise yourself with the code again, and try running it.

We had a look at what some of the code did, especially parts that were necessary to modify for the difficulty menu. Remember to run it

from your folder inside **/home/pi** and not from the main menu.

Let's have a look at some of the other parts of the code, starting with the colour codes.

We'll skip some parts of the code for you to figure out yourself.

```
#           R     G     B
WHITE      = (255, 255, 255)
BLACK      = (  0,   0,   0)
RED        = (255,   0,   0)
GREEN      = (  0, 255,   0)
DARKGREEN  = (  0, 155,   0)
DARKGRAY   = ( 40,  40,  40)
BGCOLOR    = BLACK
```

This defines colours using their RGB (red green blue) values. This is a set of values that tells the computer how to display each colour, as otherwise the computer doesn't know how to show, for instance, the colour **purple** or **orange** or dark grey.

0 is the minimum, and 255 is the maximum – so if the R (red) value is 255, there the colour will be entirely red. However, if the R value is 0, no red at all will be displayed.

In Python (specifically, using *pygame* to make our games), these colours can be defined with each value separated by commas in brackets and saved into a variable; this is what the lines of code printed on the last page do.

For instance, pink is 255, 192, 203. Try saving this into a variable with the other colours.

Once you've done this, Python has a special variable name that can be used to change the colour of the background.

Can you try making this a colour of your choosing? Think of combining the RGB colours like mixing paint colours.

We now have our *main()* function. A function in Python is a bit of code that we write, usually at the top, so that we can run it later, without needing to repeat code again. They're also useful for organising code.

For instance, if we want a function that adds two numbers together, we can write some code like this. If you want, open a new IDLE file and try this.

```
def add(num1, num2):  
    result = num1+num2  
    return result  
  
print(add(2, 3))
```

We define functions (using *def*) and indent the code inside. They can be run by writing the name of the function (in this case, *add*) and then the **arguments** inside brackets. These are the numbers that we 'pass' to the function.

Here, they are 2 and 3, so that when you run the code, it will send them to the function, they will be added together, and it will then be **returned** so that the *print* function can be used to output the answer to the screen.

Not all functions will need **arguments** sent to them, but if they are, they need to match how they are defined. In the example above, there are two **parameters** (when they're in the definition, they're called **parameters**, and when they're called, on the final line, **arguments**), which should both be numbers and there should be two of them.

If you remove one or both of them when the function is called, you'll notice you get an error, as it will be expecting two.

Let's have a look now at the **main()** function.

```
def main():  
    global FPSLOCK, DISPLAYSURF, BASICFONT  
  
    pygame.init()  
    FPSLOCK = pygame.time.Clock()  
    DISPLAYSURF =  
pygame.display.set_mode((WINDOWWIDTH, WINDOWHEIGHT))  
    BASICFONT = pygame.font.Font('freesansbold.ttf',  
18)  
    pygame.display.set_caption('Wormy')  
  
    showStartScreen()  
  
    while True:  
        runGame()  
        showGameOverScreen()
```

Here, you can see several functions are called – as before, they're the name of the functions, plus brackets. If there are arguments, these will go inside the bracket. Some of these are controlling the **pygame** module, but the ones highlighted in bold above are defined in the **wormy.py** file.

In this session, we're going to have a go at using functions to build a leaderboard system, to save our scores later to a text file.

We're going to need to write our own function inside **main()** to manage saving a leaderboard. It'll need to go between the main game loop (**runGame**) and the game ending, so that it saves immediately after the game has ended.

- Try defining this function
- Remember to give it a sensible name, such as **leaderboard**
- Remember to define it in the correct place – it needs to be defined before it is called, and not inside another function

We may move it later, but let's start simple first.

We're going to use a separate file for saving our scores to a leaderboard.

Make a new blank text file in the same directory as your Python game, and call it **scores.txt**.

Firstly, we're going to want to 'open' it using code so that Python can add text to it. While it won't show on screen, Python needs to be told where the file is and what it needs to do with it.

At the top of your new function, write the following code.

```
def leaderboard():  
    file = open("scores.txt", "a+")
```

This line of code tells Python to open the **scores.txt** file, and the **a+** means "append", or "add onto the end of an existing file". As we've already created the file, this is the best option to use for the moment.

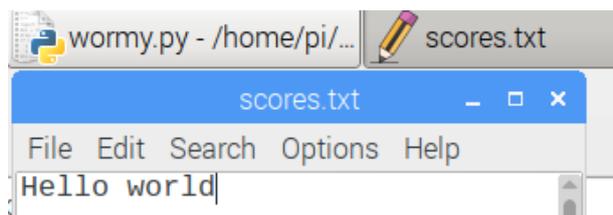
We've opened it, and can now refer to it as **file** further down in the code.

Let's just try writing some text into our file.

```
file.write("Hello world")  
file.close()
```

We should always write **file.close()** after dealing with files to prevent accidental modification afterwards.

Now make sure your function is called inside **main()**.



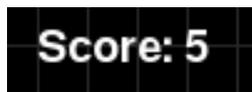
Try playing the game, and then opening up the file using a text editor (through File Manager) and see if the text appears in the file.

If it does, that's great! We've proven that we can use functions to write data to a text file, and we can now use this to save our score.

You can use the text editor to delete the "Hello world", and save it so it is an empty file again.

If it doesn't, think about where your function is called, using **leaderboard()**, and make sure the code above is indented inside the function. Also make sure the filename, **scores.txt**, is typed correctly.

Now let's see how we can add the scores the user gets into the text file.



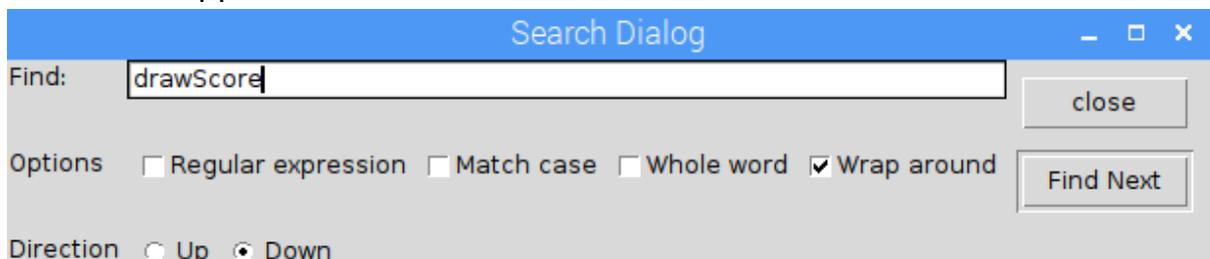
Firstly, we need to find out how the program calculates and displays the score. When you play the game, you'll notice it is displayed in the top corner of the screen.

Each of the on-screen items has a function that 'draws' it to the screen. Look in your Python file for where this is defined. You might need to scroll down a little bit to find it. We don't need to worry too much about how it displays it on-screen, but more about how it obtains the number.

```
def drawScore(score):
```

You should notice that when it is defined, it has one parameter, **score**. This is then used to show the score on screen. So we now need to look and see where this is called, and what data is 'passed' to the function – as this will be the score that we can save into the leaderboard.

To save you looking through hundreds of lines of code, we can just use the search function. Press the **Control key** (usually in the bottom left of the keyboard, written as CTRL) and **F** at the same time, and you should see a search box appear.



Look for the **drawScore** function being called using the search function.

You should find the following line of code inside of the **runGame** function.

```
drawScore(len(wormCoords) - 3)
```

This is passed to the function, so the value of `len(wormCoords) - 3` is the score that the user has got.

What does this mean?

Let's break it down and have a look at that bit of code.

len() is a way of getting the length of whatever is inside the brackets

wormCoords is a variable, which is using a *data type* called a dictionary.

-3 is subtracting 3 from the length value

What do we mean by data type?

In programming languages, we can set variables to store data. This data is of a certain data type. Some common ones include strings, integers and boolean values.

A string might be **"Python"** (note the quotation marks around it)

An integer might be **7** or **42** (it cannot have decimal points)

A boolean value can only be either **True** or **False**

What is a dictionary?

A dictionary is a way of storing data in programming languages, using a **key** and a **value** (much like an English dictionary has words and definitions).

wormCoords in our code is made up of the co-ordinates of where the worm is on screen.

The **key** here is X and Y (referring to co-ordinates on screen), and the **value** is the co-ordinate themselves. Together, the key and value make a **pair**, which counts as 1 entry in the dictionary.

We can have a look at what this looks like in our code, by adding a print statement just after the `drawScore(len(wormCoords) - 3)` line we found earlier.

Add underneath it:

```
print(wormCoords)
```

Now play your game. You might need to rearrange your windows so that you can see the shell output and the game itself.

```
[{'y': 19, 'x': 33}, {'y': 19, 'x': 32}, {'y': 19, 'x': 31}]\n[{'y': 19, 'x': 34}, {'y': 19, 'x': 33}, {'y': 19, 'x': 32}]\n[{'y': 19, 'x': 35}, {'y': 19, 'x': 34}, {'y': 19, 'x': 33}]\n[{'y': 19, 'x': 36}, {'y': 19, 'x': 35}, {'y': 19, 'x': 34}]\n[{'y': 19, 'x': 37}, {'y': 19, 'x': 36}, {'y': 19, 'x': 35}]\n[{'y': 18, 'x': 37}, {'y': 19, 'x': 37}, {'y': 19, 'x': 36}]\n[{'y': 17, 'x': 37}, {'y': 18, 'x': 37}, {'y': 19, 'x': 37}]\n[{'y': 16, 'x': 37}, {'y': 17, 'x': 37}, {'y': 18, 'x': 37}]\n[{'y': 15, 'x': 37}, {'y': 16, 'x': 37}, {'y': 17, 'x': 37}]\n[{'y': 14, 'x': 37}, {'y': 15, 'x': 37}, {'y': 16, 'x': 37}]\n[{'y': 14, 'x': 36}, {'y': 14, 'x': 37}, {'y': 15, 'x': 37}]\n[{'y': 14, 'x': 35}, {'y': 14, 'x': 36}, {'y': 14, 'x': 37}]\n[{'y': 14, 'x': 34}, {'y': 14, 'x': 35}, {'y': 14, 'x': 36}]\n[{'y': 14, 'x': 33}, {'y': 14, 'x': 34}, {'y': 14, 'x': 35}]\n[{'y': 14, 'x': 32}, {'y': 14, 'x': 33}, {'y': 14, 'x': 34}]\n[{'y': 14, 'x': 31}, {'y': 14, 'x': 32}, {'y': 14, 'x': 33}]\n[{'y': 14, 'x': 30}, {'y': 14, 'x': 31}, {'y': 14, 'x': 32}]\n[{'y': 14, 'x': 29}, {'y': 14, 'x': 30}, {'y': 14, 'x': 31}]\n[{'y': 14, 'x': 28}, {'y': 14, 'x': 29}, {'y': 14, 'x': 30}]\n[{'y': 14, 'x': 27}, {'y': 14, 'x': 28}, {'y': 14, 'x': 29}]\n[{'y': 15, 'x': 27}, {'y': 14, 'x': 27}, {'y': 14, 'x': 28}]\n[{'y': 16, 'x': 27}, {'y': 15, 'x': 27}, {'y': 14, 'x': 27}]\n[{'y': 17, 'x': 27}, {'y': 16, 'x': 27}, {'y': 15, 'x': 27}, {'y': 14, 'x': 27}]\n[{'y': 18, 'x': 27}, {'y': 17, 'x': 27}, {'y': 16, 'x': 27}, {'y': 15, 'x': 27}]\n[{'y': 19, 'x': 27}, {'y': 18, 'x': 27}, {'y': 17, 'x': 27}, {'y': 16, 'x': 27}]\n[{'y': 20, 'x': 27}, {'y': 19, 'x': 27}, {'y': 18, 'x': 27}, {'y': 17, 'x': 27}]\n[{'y': 21, 'x': 27}, {'y': 20, 'x': 27}, {'y': 19, 'x': 27}, {'y': 18, 'x': 27}]\n[{'y': 21, 'x': 26}, {'y': 21, 'x': 27}, {'y': 20, 'x': 27}, {'y': 19, 'x': 27}]\n[{'y': 21, 'x': 25}, {'y': 21, 'x': 26}, {'y': 21, 'x': 27}, {'y': 20, 'x': 27}]\n[{'y': 21, 'x': 24}, {'y': 21, 'x': 25}, {'y': 21, 'x': 26}, {'y': 21, 'x': 27}]\n[{'y': 21, 'x': 23}, {'y': 21, 'x': 24}, {'y': 21, 'x': 25}, {'y': 21, 'x': 26}]\n[{'y': 21, 'x': 22}, {'y': 21, 'x': 23}, {'y': 21, 'x': 24}, {'y': 21, 'x': 25}]\n[{'y': 21, 'x': 21}, {'y': 21, 'x': 22}, {'y': 21, 'x': 23}, {'y': 21, 'x': 24}]\n[{'y': 21, 'x': 20}, {'y': 21, 'x': 21}, {'y': 21, 'x': 22}, {'y': 21, 'x': 23}]\n[{'y': 21, 'x': 19}, {'y': 21, 'x': 20}, {'y': 21, 'x': 21}, {'y': 21, 'x': 22}]\n[{'y': 21, 'x': 18}, {'y': 21, 'x': 19}, {'y': 21, 'x': 20}, {'y': 21, 'x': 21}]\n[{'y': 21, 'x': 17}, {'y': 21, 'x': 18}, {'y': 21, 'x': 19}, {'y': 21, 'x': 20}]\n[{'y': 21, 'x': 16}, {'y': 21, 'x': 17}, {'y': 21, 'x': 18}, {'y': 21, 'x': 19}]\n[{'y': 21, 'x': 15}, {'y': 21, 'x': 16}, {'y': 21, 'x': 17}, {'y': 21, 'x': 18}]\n[{'y': 21, 'x': 14}, {'y': 21, 'x': 15}, {'y': 21, 'x': 16}, {'y': 21, 'x': 17}]\n[{'y': 21, 'x': 13}, {'y': 21, 'x': 14}, {'y': 21, 'x': 15}, {'y': 21, 'x': 16}]\n[{'y': 21, 'x': 12}, {'y': 21, 'x': 13}, {'y': 21, 'x': 14}, {'y': 21, 'x': 15}]\n[{'y': 21, 'x': 11}, {'y': 21, 'x': 12}, {'y': 21, 'x': 13}, {'y': 21, 'x': 14}]\n[{'y': 21, 'x': 10}, {'y': 21, 'x': 11}, {'y': 21, 'x': 12}, {'y': 21, 'x': 13}]\n[{'y': 21, 'x': 9}, {'y': 21, 'x': 10}, {'y': 21, 'x': 11}, {'y': 21, 'x': 12}]\n[{'y': 21, 'x': 8}, {'y': 21, 'x': 9}, {'y': 21, 'x': 10}, {'y': 21, 'x': 11}]\n[{'y': 21, 'x': 7}, {'y': 21, 'x': 8}, {'y': 21, 'x': 9}, {'y': 21, 'x': 10}]\n[{'y': 21, 'x': 6}, {'y': 21, 'x': 7}, {'y': 21, 'x': 8}, {'y': 21, 'x': 9}]\n[{'y': 21, 'x': 5}, {'y': 21, 'x': 6}, {'y': 21, 'x': 7}, {'y': 21, 'x': 8}]\n[{'y': 21, 'x': 4}, {'y': 21, 'x': 5}, {'y': 21, 'x': 6}, {'y': 21, 'x': 7}]\n[{'y': 21, 'x': 3}, {'y': 21, 'x': 4}, {'y': 21, 'x': 5}, {'y': 21, 'x': 6}]\n[{'y': 21, 'x': 2}, {'y': 21, 'x': 3}, {'y': 21, 'x': 4}, {'y': 21, 'x': 5}]\n[{'y': 21, 'x': 1}, {'y': 21, 'x': 2}, {'y': 21, 'x': 3}, {'y': 21, 'x': 4}]\n[{'y': 21, 'x': 0}, {'y': 21, 'x': 1}, {'y': 21, 'x': 2}, {'y': 21, 'x': 3}]\n[{'y': 20, 'x': 0}, {'y': 21, 'x': 0}, {'y': 20, 'x': 1}, {'y': 21, 'x': 1}]\n[{'y': 19, 'x': 0}, {'y': 20, 'x': 0}, {'y': 19, 'x': 1}, {'y': 20, 'x': 1}]\n[{'y': 18, 'x': 0}, {'y': 19, 'x': 0}, {'y': 18, 'x': 1}, {'y': 19, 'x': 1}]\n[{'y': 17, 'x': 0}, {'y': 18, 'x': 0}, {'y': 17, 'x': 1}, {'y': 18, 'x': 1}]\n[{'y': 16, 'x': 0}, {'y': 17, 'x': 0}, {'y': 16, 'x': 1}, {'y': 17, 'x': 1}]\n[{'y': 15, 'x': 0}, {'y': 16, 'x': 0}, {'y': 15, 'x': 1}, {'y': 16, 'x': 1}]\n[{'y': 14, 'x': 0}, {'y': 15, 'x': 0}, {'y': 14, 'x': 1}, {'y': 15, 'x': 1}]\n[{'y': 13, 'x': 0}, {'y': 14, 'x': 0}, {'y': 13, 'x': 1}, {'y': 14, 'x': 1}]\n[{'y': 12, 'x': 0}, {'y': 13, 'x': 0}, {'y': 12, 'x': 1}, {'y': 13, 'x': 1}]\n[{'y': 11, 'x': 0}, {'y': 12, 'x': 0}, {'y': 11, 'x': 1}, {'y': 12, 'x': 1}]\n[{'y': 10, 'x': 0}, {'y': 11, 'x': 0}, {'y': 10, 'x': 1}, {'y': 11, 'x': 1}]\n[{'y': 9, 'x': 0}, {'y': 10, 'x': 0}, {'y': 9, 'x': 1}, {'y': 10, 'x': 1}]\n[{'y': 8, 'x': 0}, {'y': 9, 'x': 0}, {'y': 8, 'x': 1}, {'y': 9, 'x': 1}]\n[{'y': 7, 'x': 0}, {'y': 8, 'x': 0}, {'y': 7, 'x': 1}, {'y': 8, 'x': 1}]\n[{'y': 6, 'x': 0}, {'y': 7, 'x': 0}, {'y': 6, 'x': 1}, {'y': 7, 'x': 1}]\n[{'y': 5, 'x': 0}, {'y': 6, 'x': 0}, {'y': 5, 'x': 1}, {'y': 6, 'x': 1}]\n[{'y': 4, 'x': 0}, {'y': 5, 'x': 0}, {'y': 4, 'x': 1}, {'y': 5, 'x': 1}]\n[{'y': 3, 'x': 0}, {'y': 4, 'x': 0}, {'y': 3, 'x': 1}, {'y': 4, 'x': 1}]\n[{'y': 2, 'x': 0}, {'y': 3, 'x': 0}, {'y': 2, 'x': 1}, {'y': 3, 'x': 1}]\n[{'y': 1, 'x': 0}, {'y': 2, 'x': 0}, {'y': 1, 'x': 1}, {'y': 2, 'x': 1}]\n[{'y': 0, 'x': 0}, {'y': 1, 'x': 0}, {'y': 0, 'x': 1}, {'y': 1, 'x': 1}]\n[{'y': 0, 'x': 1}, {'y': 0, 'x': 2}, {'y': 0, 'x': 3}, {'y': 0, 'x': 4}]\n[{'y': 0, 'x': 5}, {'y': 0, 'x': 6}, {'y': 0, 'x': 7}, {'y': 0, 'x': 8}]\n[{'y': 0, 'x': 9}, {'y': 0, 'x': 10}, {'y': 0, 'x': 11}, {'y': 0, 'x': 12}]\n[{'y': 0, 'x': 13}, {'y': 0, 'x': 14}, {'y': 0, 'x': 15}, {'y': 0, 'x': 16}]\n[{'y': 0, 'x': 17}, {'y': 0, 'x': 18}, {'y': 0, 'x': 19}, {'y': 0, 'x': 20}]\n[{'y': 0, 'x': 21}, {'y': 0, 'x': 22}, {'y': 0, 'x': 23}, {'y': 0, 'x': 24}]\n[{'y': 0, 'x': 25}, {'y': 0, 'x': 26}, {'y': 0, 'x': 27}, {'y': 0, 'x': 28}]\n[{'y': 0, 'x': 29}, {'y': 0, 'x': 30}, {'y': 0, 'x': 31}, {'y': 0, 'x': 32}]\n[{'y': 0, 'x': 33}, {'y': 0, 'x': 34}, {'y': 0, 'x': 35}, {'y': 0, 'x': 36}]\n[{'y': 0, 'x': 37}, {'y': 0, 'x': 38}, {'y': 0, 'x': 39}, {'y': 0, 'x': 40}]\n[{'y': 0, 'x': 41}, {'y': 0, 'x': 42}, {'y': 0, 'x': 43}, {'y': 0, 'x': 44}]\n[{'y': 0, 'x': 45}, {'y': 0, 'x': 46}, {'y': 0, 'x': 47}, {'y': 0, 'x': 48}]\n[{'y': 0, 'x': 49}, {'y': 0, 'x': 50}, {'y': 0, 'x': 51}, {'y': 0, 'x': 52}]\n[{'y': 0, 'x': 53}, {'y': 0, 'x': 54}, {'y': 0, 'x': 55}, {'y': 0, 'x': 56}]\n[{'y': 0, 'x': 57}, {'y': 0, 'x': 58}, {'y': 0, 'x': 59}, {'y': 0, 'x': 60}]\n[{'y': 0, 'x': 61}, {'y': 0, 'x': 62}, {'y': 0, 'x': 63}, {'y': 0, 'x': 64}]\n[{'y': 0, 'x': 65}, {'y': 0, 'x': 66}, {'y': 0, 'x': 67}, {'y': 0, 'x': 68}]\n[{'y': 0, 'x': 69}, {'y': 0, 'x': 70}, {'y': 0, 'x': 71}, {'y': 0, 'x': 72}]\n[{'y': 0, 'x': 73}, {'y': 0, 'x': 74}, {'y': 0, 'x': 75}, {'y': 0, 'x': 76}]\n[{'y': 0, 'x': 77}, {'y': 0, 'x': 78}, {'y': 0, 'x': 79}, {'y': 0, 'x': 80}]\n[{'y': 0, 'x': 81}, {'y': 0, 'x': 82}, {'y': 0, 'x': 83}, {'y': 0, 'x': 84}]\n[{'y': 0, 'x': 85}, {'y': 0, 'x': 86}, {'y': 0, 'x': 87}, {'y': 0, 'x': 88}]\n[{'y': 0, 'x': 89}, {'y': 0, 'x': 90}, {'y': 0, 'x': 91}, {'y': 0, 'x': 92}]\n[{'y': 0, 'x': 93}, {'y': 0, 'x': 94}, {'y': 0, 'x': 95}, {'y': 0, 'x': 96}]\n[{'y': 0, 'x': 97}, {'y': 0, 'x': 98}, {'y': 0, 'x': 99}, {'y': 0, 'x': 100}]\n[{'y': 1, 'x': 0}, {'y': 1, 'x': 1}, {'y': 1, 'x': 2}, {'y': 1, 'x': 3}]\n[{'y': 1, 'x': 4}, {'y': 1, 'x': 5}, {'y': 1, 'x': 6}, {'y': 1, 'x': 7}]\n[{'y': 1, 'x': 8}, {'y': 1, 'x': 9}, {'y': 1, 'x': 10}, {'y': 1, 'x': 11}]\n[{'y': 1, 'x': 12}, {'y': 1, 'x': 13}, {'y': 1, 'x': 14}, {'y': 1, 'x': 15}]\n[{'y': 1, 'x': 16}, {'y': 1, 'x': 17}, {'y': 1, 'x': 18}, {'y': 1, 'x': 19}]\n[{'y': 1, 'x': 20}, {'y': 1, 'x': 21}, {'y': 1, 'x': 22}, {'y': 1, 'x': 23}]\n[{'y': 1, 'x': 24}, {'y': 1, 'x': 25}, {'y': 1, 'x': 26}, {'y': 1, 'x': 27}]\n[{'y': 1, 'x': 28}, {'y': 1, 'x': 29}, {'y': 1, 'x': 30}, {'y': 1, 'x': 31}]\n[{'y': 1, 'x': 32}, {'y': 1, 'x': 33}, {'y': 1, 'x': 34}, {'y': 1, 'x': 35}]\n[{'y': 1, 'x': 36}, {'y': 1, 'x': 37}, {'y': 1, 'x': 38}, {'y': 1, 'x': 39}]\n[{'y': 1, 'x': 40}, {'y': 1, 'x': 41}, {'y': 1, 'x': 42}, {'y': 1, 'x': 43}]\n[{'y': 1, 'x': 44}, {'y': 1, 'x': 45}, {'y': 1, 'x': 46}, {'y': 1, 'x': 47}]\n[{'y': 1, 'x': 48}, {'y': 1, 'x': 49}, {'y': 1, 'x': 50}, {'y': 1, 'x': 51}]\n[{'y': 1, 'x': 52}, {'y': 1, 'x': 53}, {'y': 1, 'x': 54}, {'y': 1, 'x': 55}]\n[{'y': 1, 'x': 56}, {'y': 1, 'x': 57}, {'y': 1, 'x': 58}, {'y': 1, 'x': 59}]\n[{'y': 1, 'x': 60}, {'y': 1, 'x': 61}, {'y': 1, 'x': 62}, {'y': 1, 'x': 63}]\n[{'y': 1, 'x': 64}, {'y': 1, 'x': 65}, {'y': 1, 'x': 66}, {'y': 1, 'x': 67}]\n[{'y': 1, 'x': 68}, {'y': 1, 'x': 69}, {'y': 1, 'x': 70}, {'y': 1, 'x': 71}]\n[{'y': 1, 'x': 72}, {'y': 1, 'x': 73}, {'y': 1, 'x': 74}, {'y': 1, 'x': 75}]\n[{'y': 1, 'x': 76}, {'y': 1, 'x': 77}, {'y': 1, 'x': 78}, {'y': 1, 'x': 79}]\n[{'y': 1, 'x': 80}, {'y': 1, 'x': 81}, {'y': 1, 'x': 82}, {'y': 1, 'x': 83}]\n[{'y': 1, 'x': 84}, {'y': 1, 'x': 85}, {'y': 1, 'x': 86}, {'y': 1, 'x': 87}]\n[{'y': 1, 'x': 88}, {'y': 1, 'x': 89}, {'y': 1, 'x': 90}, {'y': 1, 'x': 91}]\n[{'y': 1, 'x': 92}, {'y': 1, 'x': 93}, {'y': 1, 'x': 94}, {'y': 1, 'x': 95}]\n[{'y': 1, 'x': 96}, {'y': 1, 'x': 97}, {'y': 1, 'x': 98}, {'y': 1, 'x': 99}]\n[{'y': 1, 'x': 100}, {'y': 1, 'x': 101}, {'y': 1, 'x': 102}, {'y': 1, 'x': 103}]\n[{'y': 1, 'x': 104}, {'y': 1, 'x': 105}, {'y': 1, 'x': 106}, {'y': 1, 'x': 107}]\n[{'y': 1, 'x': 108}, {'y': 1, 'x': 109}, {'y': 1, 'x': 110}, {'y': 1, 'x': 111}]\n[{'y': 1, 'x': 112}, {'y': 1, 'x': 113}, {'y': 1, 'x': 114}, {'y': 1, 'x': 115}]\n[{'y': 1, 'x': 116}, {'y': 1, 'x': 117}, {'y': 1, 'x': 118}, {'y': 1, 'x': 119}]\n[{'y': 1, 'x': 120}, {'y': 1, 'x': 121}, {'y': 1, 'x': 122}, {'y': 1, 'x': 123}]\n[{'y': 1, 'x': 124}, {'y': 1, 'x': 125}, {'y': 1, 'x': 126}, {'y': 1, 'x': 127}]\n[{'y': 1, 'x': 128}, {'y': 1, 'x': 129}, {'y': 1, 'x': 130}, {'y': 1, 'x': 131}]\n[{'y': 1, 'x': 132}, {'y': 1, 'x': 133}, {'y': 1, 'x': 134}, {'y': 1, 'x': 135}]\n[{'y': 1, 'x': 136}, {'y': 1, 'x': 137}, {'y': 1, 'x': 138}, {'y': 1, 'x': 139}]\n[{'y': 1, 'x': 140}, {'y': 1, 'x': 141}, {'y': 1, 'x': 142}, {'y': 1, 'x': 143}]\n[{'y': 1, 'x': 144}, {'y': 1, 'x': 145}, {'y': 1, 'x': 146}, {'y': 1, 'x': 147}]\n[{'y': 1, 'x': 148}, {'y': 1, 'x': 149}, {'y': 1, 'x': 150}, {'y': 1, 'x': 151}]\n[{'y': 1, 'x': 152}, {'y': 1, 'x': 153}, {'y': 1, 'x': 154}, {'y': 1, 'x': 155}]\n[{'y': 1, 'x': 156}, {'y': 1, 'x': 157}, {'y': 1, 'x': 158}, {'y': 1, 'x': 159}]\n[{'y': 1, 'x': 160}, {'y': 1, 'x': 161}, {'y': 1, 'x': 162}, {'y': 1, 'x': 163}]\n[{'y': 1, 'x': 164}, {'y': 1, 'x': 165}, {'y': 1, 'x': 166}, {'y': 1, 'x': 167}]\n[{'y': 1, 'x': 168}, {'y': 1, 'x': 169}, {'y': 1, 'x': 170}, {'y': 1, 'x': 171}]\n[{'y': 1, 'x': 172}, {'y': 1, 'x': 173}, {'y': 1, 'x': 174}, {'y': 1, 'x': 175}]\n[{'y': 1, 'x': 176}, {'y': 1, 'x': 177}, {'y': 1, 'x': 178}, {'y': 1, 'x': 179}]\n[{'y': 1, 'x': 180}, {'y': 1, 'x': 181}, {'y': 1, 'x': 182}, {'y': 1, 'x': 183}]\n[{'y': 1, 'x': 184}, {'y': 1, 'x': 185}, {'y': 1, 'x': 186}, {'y': 1, 'x': 187}]\n[{'y': 1, 'x': 188}, {'y': 1, 'x': 189}, {'y': 1, 'x': 190}, {'y': 1, 'x': 191}]\n[{'y': 1, 'x': 192}, {'y': 1, 'x': 193}, {'y': 1, 'x': 194}, {'y': 1, 'x': 195}]\n[{'y': 1, 'x': 196}, {'y': 1, 'x': 197}, {'y': 1, 'x': 198}, {'y': 1, 'x': 199}]\n[{'y': 1, 'x': 200}, {'y': 1, 'x': 201}, {'y': 1, 'x': 202}, {'y': 1, 'x': 203}]\n[{'y': 1, 'x': 204}, {'y': 1, 'x': 205}, {'y': 1, 'x': 206}, {'y': 1, 'x': 207}]\n[{'y': 1, 'x': 208}, {'y': 1, 'x': 209}, {'y': 1, 'x': 210}, {'y': 1, 'x': 211}]\n[{'y': 1, 'x': 212}, {'y': 1, 'x': 213}, {'y': 1, 'x': 214}, {'y': 1, 'x': 215}]\n[{'y': 1, 'x': 216}, {'y': 1, 'x': 217}, {'y': 1, 'x': 218}, {'y': 1, 'x': 219}]\n[{'y': 1, 'x': 220}, {'y': 1, 'x': 221}, {'y': 1, 'x': 222}, {'y': 1, 'x': 223}]\n[{'y': 1, 'x': 224}, {'y': 1, 'x': 225}, {'y': 1, 'x': 226}, {'y': 1, 'x': 227}]\n[{'y': 1, 'x': 228}, {'y': 1, 'x': 229}, {'y': 1, 'x': 230}, {'y': 1, 'x': 231}]\n[{'y': 1, 'x': 232}, {'y': 1, 'x': 233}, {'y': 1, 'x': 234}, {'y': 1, 'x': 235}]\n[{'y': 1, 'x': 236}, {'y': 1, 'x': 237}, {'y': 1, 'x': 238}, {'y': 1, 'x': 239}]\n[{'y': 1, 'x': 240}, {'y': 1, 'x': 241}, {'y': 1, 'x': 242}, {'y': 1, 'x': 243}]\n[{'y': 1, 'x': 244}, {'y': 1, 'x': 245}, {'y': 1, 'x': 246}, {'y': 1, 'x': 247}]\n[{'y': 1, 'x': 248}, {'y': 1, 'x': 249}, {'y': 1, 'x': 250}, {'y': 1, 'x': 251}]\n[{'y': 1, 'x': 252}, {'y': 1, 'x': 253}, {'y': 1, 'x': 254}, {'y': 1, 'x': 255}]\n[{'y': 1, 'x': 256}, {'y': 1, 'x': 257}, {'y': 1, 'x': 258}, {'y': 1, 'x': 259}]\n[{'y': 1, 'x': 260}, {'y': 1, 'x': 261}, {'y': 1, 'x': 262}, {'y': 1, 'x': 263}]\n[{'y': 1, 'x': 264}, {'y': 1, 'x': 265}, {'y': 1, 'x': 266}, {'y': 1, 'x': 267}]\n[{'y': 1, 'x': 268}, {'y': 1, 'x': 269}, {'y': 1, 'x': 270}, {'y': 1, 'x': 271}]\n[{'y': 1, 'x': 272}, {'y': 1, 'x': 273}, {'y': 1, 'x': 274}, {'y': 1, 'x': 275}]\n[{'y': 1, 'x': 276}, {'y': 1, 'x': 277}, {'y': 1, 'x': 278}, {'y': 1, 'x': 279}]\n[{'y': 1, 'x': 280}, {'y': 1, 'x': 281}, {'y': 1, 'x': 282}, {'y': 1, 'x': 283}]\n[{'y': 1, 'x': 284}, {'y': 1, 'x': 285}, {'y': 1, 'x': 286}, {'y': 1, 'x': 287}]\n[{'y': 1, 'x': 288}, {'y': 1, 'x': 289}, {'y': 1, 'x': 290}, {'y': 1, 'x': 291}]\n[{'y': 1, 'x': 292}, {'y': 1, 'x': 293}, {'y': 1, 'x': 294}, {'y': 1, 'x': 295}]\n[{'y': 1, 'x': 296}, {'y': 1, 'x': 297}, {'y': 1, 'x': 298}, {'y': 1, 'x': 299}]\n[{'y': 1, 'x': 300}, {'y': 1, 'x': 301}, {'y': 1, 'x': 302}, {'y': 1, 'x': 303}]\n[{'y': 1, 'x': 304}, {'y': 1, 'x': 305}, {'y': 1, 'x': 306}, {'y': 1, 'x': 307}]\n[{'y': 1, 'x': 308}, {'y': 1, 'x': 309}, {'y': 1, 'x': 310}, {'y': 1, 'x': 311}]\n[{'y': 1, 'x': 312}, {'y': 1, 'x': 313}, {'y': 1, 'x': 314}, {'y': 1, 'x': 315}]\n[{'y': 1, 'x': 316}, {'y': 1, 'x': 317}, {'y': 1, 'x': 318}, {'y': 1, 'x': 319}]\n[{'y': 1, 'x': 320}, {'y': 1, 'x': 321}, {'y': 1, 'x': 322}, {'y': 1, 'x': 323}]\n[{'y': 1, 'x': 324}, {'y': 1, 'x': 325}, {'y': 1, 'x': 326}, {'y': 1, 'x': 327}]\n[{'y': 1, 'x': 328}, {'y': 1, 'x': 329}, {'y': 1, 'x': 330}, {'y': 1, 'x': 331}]\n[{'y': 1, 'x': 332}, {'y': 1, 'x': 333}, {'y': 1, 'x': 334}, {'y': 1, 'x': 335}]\n[{'y': 1, 'x': 336}, {'y': 1, 'x': 337}, {'y': 1, 'x': 338}, {'y': 1, 'x': 339}]\n[{'y': 1, 'x': 340}, {'y': 1, 'x': 341}, {'y': 1, 'x': 342}, {'y': 1, 'x': 343}]\n[{'y': 1, 'x': 344}, {'y': 1, 'x': 345}, {'y': 1, 'x': 346}, {'y': 1, 'x': 347}]\n[{'y': 1, 'x': 348}, {'y': 1, 'x': 349}, {'y': 1, 'x': 350}, {'y': 1, 'x': 351}]\n[{'y': 1, 'x': 352}, {'y': 1, 'x': 353}, {'y': 1, 'x': 354}, {'y': 1, 'x': 355}]\n[{'y': 1, 'x': 356}, {'y': 1, 'x': 357}, {'y': 1, 'x': 358}, {'y': 1, 'x': 359}]\n[{'y': 1, 'x': 360}, {'y': 1, 'x': 361}, {'y': 1, 'x': 362}, {'y': 1, 'x': 363}]\n[{'y': 1, 'x': 364}, {'y': 1, 'x': 365}, {'y': 1, 'x': 366}, {'y': 1, 'x': 367}]\n[{'y': 1, 'x': 368}, {'y': 1, 'x': 369}, {'y': 1, 'x': 370}, {'y': 1, 'x': 371}]\n[{'y': 1, 'x': 372}, {'y': 1, 'x': 373}, {'y': 1, 'x': 374}, {'y': 1, 'x': 375}]\n[{'y': 1, 'x': 376}, {'y': 1, 'x': 377}, {'y': 1, 'x': 378}, {'y': 1, 'x': 379}]\n[{'y': 1, 'x': 380}, {'y': 1, 'x': 381}, {'y': 1, 'x': 382}, {'y': 1, 'x': 383}]\n[{'y': 1, 'x': 384}, {'y': 1, 'x': 385}, {'y': 1, 'x': 386}, {'y': 1, 'x': 387}]\n[{'y': 1, 'x': 388}, {'y': 1, 'x': 389}, {'y': 1, 'x': 390}, {'y': 1, 'x': 391}]\n[{'y': 1, 'x': 392}, {'y': 1, 'x': 393}, {'y': 1, 'x': 394}, {'y': 1, 'x': 395}]\n[{'y': 1, 'x': 396}, {'y': 1, 'x': 397}, {'y': 1, 'x': 398}, {'y': 1, 'x': 399}]\n[{'y': 1, 'x': 400}, {'y': 1, 'x': 401}, {'y': 1, 'x': 402}, {'y': 1, 'x': 403}]\n[{'y': 1, 'x': 404}, {'y': 1, 'x': 405}, {'y': 1, 'x': 406}, {'y': 1, 'x': 407}]\n[{'y': 1, 'x': 408}, {'y': 1, 'x': 409}, {'y': 1, 'x': 410}, {'y': 1, 'x': 411}]\n[{'y': 1, 'x': 412}, {'y': 1, 'x': 413}, {'y': 1, 'x': 414}, {'y': 1, 'x': 415}]\n[{'y': 1, 'x': 416}, {'y': 1, 'x': 417}, {'y': 1, 'x': 418}, {'y': 1, 'x': 419
```

You should notice that when you move around, the co-ordinates on screen also change to give a current indication of where the worm is.

You'll also see that when you eat the apple, to increase the score, the length of the dictionary you're printing on screen also changes (it may overflow onto another line), but each time the length of the worm increases, the length of the dictionary increases to store a co-ordinate (X and Y) value for each part of the worm.

This is how the game stores both the length and position of the worm – as the worm always starts out being 3 long, and the score starts at 0, this explains the **-3** we saw earlier in the code to calculate the final score.

If you have a look at the comments from the original programmer inside the **runGame** function you can also see how this dictionary (**wormCoords**) is used to check whether the worm has hit itself or the edge (causing the game to end) or has eaten an apple.

So that it is easier to reference later, let's save this `len(wormCoords) - 3` as a variable called *score*, so that we can pass it to our leaderboard function.

We can add this just before our print statement.

```
score = len(wormCoords) - 3
```

Now, let's go back to our **leaderboard** function. We'll want it to take the **score** as a parameter, and then save this to the file. You'll need to change a couple of lines, printed below. The opening and closing of the file don't need to change.

Once you've passed the score as an argument, you can then use it within the function to save it to the file.

We use the **str** to convert the score, an integer, back to a string so it can be written to the file properly.

```
def leaderboard(score):  
    file.write(str(score))
```

Try running your game now.

What happens after the 'Game Over' screen?

Hint: look in the shell output

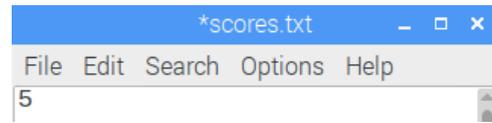
You probably got an error message that said something like:

```
NameError: name 'score' is not defined
```

Although we've declared our **score** variable, in most programming languages, variables only stay inside the function they are declared in.

This means that, because we have declared our **score** variable inside **runGame**, and are then trying to use it from within **main** (to call our leaderboard function), **main** doesn't know where **score** is, as it is not declared inside it.

To test this out, at the top of the **main** function, add a line that sets a variable called *score* to 5. Now run the code again, and get to the Game Over screen. You'll see that, regardless of your actual score, it will always save '5' to the text file as this is the only variable that **main** can 'see' and use.



Once you've done this, delete the *score* variable you've just created and we'll fix the problem using something called a **global** variable.

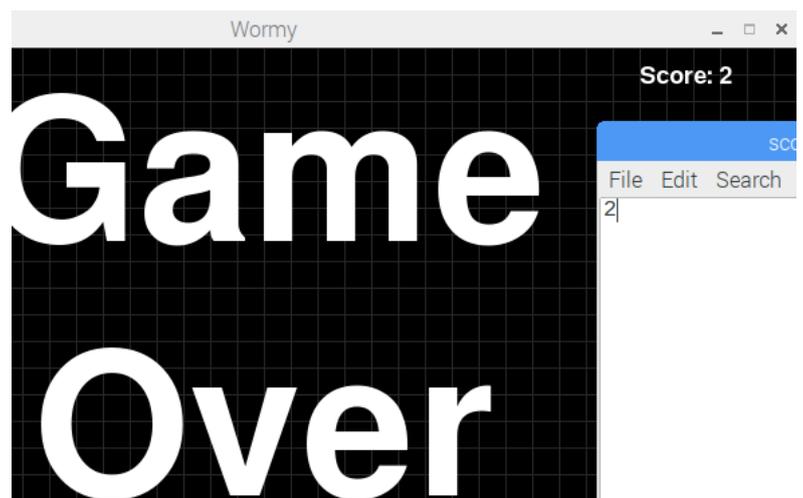
This is a special kind of variable which can be used by every function in the program. Although sometimes having variables confined to just one function can be a good thing (to prevent you accidentally changing something in one function where it is needed in another), in cases like this, global variables can be helpful. Making a variable global is really easy, too.

The line before it is assigned to `len(wormCoords) - 3`

should be `global score`

This tells Python that any function can now use this *score* variable.

Now try running your code, and make a note of your score. You should now see that your score is saved into the text file.



There are still a few ways we can improve our leaderboard.

For instance, if you run the game multiple times, you'll see that the scores are all on the same line. So if you score 2, 5 and 3, it'll look like your score is a (very impressive!) 253.

We can fix this by adding a new line every time the score is saved. This is the equivalent of pressing ENTER when writing a text file.

Look for the bit of code where you write the score to the text file. After the score, but inside the brackets, add:

```
+ '\n'
```

This is a special string that tells Python to insert a new line after the score. We use the plus symbol when dealing with strings to add strings together.

For instance, `'Tech'` + `'Resort'` would be `'TechResort'`

This can be used to add different pieces of information together and is called *string concatenation*.

Your challenge

Add a username to go with the score, so that the scoreboard will look something like:

David, 4

Ethan, 3

Sam, 6

Here's some hints...

- You'll need to ask the user for their name at the start of the game (before the difficulty selection) and save this as a variable
- You'll need to change the leaderboard function so that it will also add the name alongside the score
- Make sure you use a comma, so that we can separate the name from the score. **name + ',' + score**

Now play your game a few times and get a friend to play and save the score

Take a look at your leaderboard and see what your text file looks like now.

Is it saving the names and scores as you expected?

If not, think about your code. Is it returning an error message? Are your variables correctly defined and used properly?

Let's have a think about how we can improve our scoreboard further.

While we can see the **scores.txt** file by opening it up in the text editor and looking at it, we can also set it up to show the scores in a Python shell.

This will involve reading the text file and then outputting it. This works a bit like writing to it.

Write a new function called **showLeaderboard** or something similar.

We can then read our scoreboard and sort the data to see who has the highest score, before outputting it to the display.

This is a little bit fiddly, but the following code will read in each line, and convert it into a dictionary where the name is a string, and the score is an integer. It's important that a comma is used to separate the name and score.

```
def showLeaderboard():
    import operator
    d = {}
    with open("scores.txt") as f:
        for line in f:
            (key, val) = line.rstrip().split(',')
            d[str(key)] = int(val)

    sorted_scores = sorted(d.items(),
        key=operator.itemgetter(1))
```

Note the final line should just be one line, there is no need for a gap – a long line of code just doesn't quite fit onto one line on the page!

This will save a list of the scores, in order, to **sorted_scores**, which you can then print to the screen.

```
for s in reversed(sorted_scores):  
    print(s)
```

We use **reversed** as this will show the results from high to low.

You'll have to think about how you want to display the leaderboard – before or after the game?

How might you ask the player if they want to see the leaderboard?

You could use an **if** statement, similar to the one you used for the difficulty selection menu. If they do want to see it, then call the function to show the leaderboard. If they don't, carry on with the game as normal.

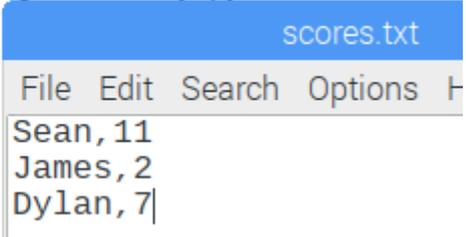
Your game should start by asking the user for 3 pieces of information using **input** and **if** statements.

- 1) Please enter your name
- 2) Would you like to see the leaderboard?
- 3) Please enter your difficulty selection

Think about which variables might need to be **global** and where you should define and call your **showLeaderboard** function.

It might look something a little like this once you're done:

```
===== RESTART: /home/pi/python_games/wormy.py =====  
Enter your name:Test  
Would you like to see the leaderboard?yes  
( 'Sean', 11)  
( 'Dylan', 7)  
( 'James', 2)  
Enter your difficulty level:
```



How else can we improve the leaderboard system?

Although we have our difficulty selection menu, the leaderboard isn't affected by this. For instance, a score of 3 on easy is a lot easier than a 3 on hard, but the leaderboard doesn't show it any differently, despite the increased challenge!

Your challenge

Combine the difficulty system with the leaderboard. You could:

Show the difficulty next to the player's name. This means it'll look like:

David (hard), 4

Ethan (easy), 3

Sam (medium), 6

Hint - You'll need to use the variable that takes the user's input on the difficulty selection screen and write this to the text file using string concatenation

You'll need **name + difficulty + comma to separate + score + new line**
and/or

Multiply the score depending on the difficulty – so playing on medium is worth double points, or playing on hard is worth triple points, for instance

Hint - You'll need to multiply the score variable before it is written to the text file. The symbol for multiplication is * (this can be found on your keyboard as Shift-8).

You may wish to adjust the difficulty in places as well to

To multiply variables in Python, use **a = b*c**, where **b** and **c** can be either variables that already exist or numbers.

```
scores.txt
File Edit Search Options Help
Sean (medium),6
Ethan (easy),8
James (hard),3
David (medium),4
Dylan (easy),7

Python 3.5.3 (default, Sep 27 2018, 17:25:
[GCC 6.3.0 20170516] on linux
Type "copyright", "credits" or "license()"
>>>
===== RESTART: /home/pi/pytho
Enter your name:Test
Would you like to see the leaderboard?yes
('Ethan (easy)', 8)
('Dylan (easy)', 7)
('Sean (medium)', 6)
('David (medium)', 4)
('James (hard)', 3)
Enter your difficulty level:
```

You can see our menu and leaderboard system now works as we expect it to – it takes our input, saves our scores, and then can order them and output them.

Extra Challenges

To make the game a bit harder, you can hide the grid that helps you see and line up where the worm and apple are. Look in the code for the function that shows the grid on screen – you can either stop it being called (and run) or change the colour of it.

Think about how you might combine this module with some of our other scripts – for instance, could you try the electronics module, and then make an LED light up whenever a point is scored? Or make a buzzer sound for ‘game over’? Or could you take a selfie using the Pi camera after each game?

Could you try tinkering with some of the other Python Games? You’ll need to read and follow the code closely so you can see exactly what is happening on each line of code, and what happens to different variables.