

## Introduction

Throughout the next 8 weeks we will be creating a game! Hopefully you'll all learn a lot about python.

The first step is to learn about the program we'll be using, there will be a lot of code written that you won't understand. Don't worry! hopefully by the end of it you'll have a better understanding of the ins and outs of python and you might be able to see how it all works.

The first file to understand is adventuregame.py it looks like this:

---

```
1 import pygame
2 from pygame.locals import *
3
4 from Engine.scenenode import *
5 from Engine.texturemanager import *
6 from Engine.inputmanager import *
7 from Engine.object import *
8 from player import *
9 from game import *
10
11
12 class AdventureGame(Game):
13
14     # This functions will create objects and add them to the scene.
15     def create_scene(self):
16         return
17
18 if __name__ == "__main__":
19     AdventureGame.start_game( AdventureGame())
```

---

On lines 1 to 9 we are importing code from other python files to use within this one. for example, importing pygame allows us to use the pygame library which is the backbone of the entire game.

Lines 12 to 16 define something called a class (which in this case is called AdventureGame) and within this class it creates a function called create\_scene. this function is called to add objects to the scene which will then be drawn to the screen.

The final two lines of this code start the game when the file is run.

The line 'if \_\_name\_\_ == "\_\_main\_\_":' checks whether the file has been run by python, don't worry if you don't understand the syntax of this. The final line calls the function start\_game() and passes an object of the AdventureGame class into it. again don't worry if you don't understand a

lot of this, hopefully you soon will.

## **Creating a Player**

Let's focus now on creating a player object and adding it to the scene. first we must create a player object, then we must add it to something called the 'scene graph'