Yannik Nelson AH Project

This is the Advanced Higher project of Yannik Nelson.

The project is a game, in this case space invaders.

**Mechanics:**

Aliens:

There will be 5 layers of aliens.

3 Types of aliens:

10pts:

These constitute the lower 2 layers.

20pts:

These constitute the 2 layers above the 10pts aliens.

30pts:

These constitute the top layer.

There will be 11 aliens on each row.

There will also be a bonus point 'mother ship' that will occasionally

travel across the top of the screen, above all the aliens this will be worth 100pts.

Bombs:

There will be 2 types of projectile, bolt (fast) and arrow (slow).

Movement:

The aliens will all move across the screen, then when they reach the end they will

shift down a bit and their movement speed will increase.

Player:

The player will only be able to move right and left. Shots will only come from the centre of

the player character, this makes the game harder. The player will have 3 'lives' in total,

each time the player 'dies' there will be a short explosion animation.

Barricade:

There will be 4 barricades in total, each time a shot hits a barricade a 'chunk' will be taken

out of it.

Projectiles:

There will be 3 types of Projectiles in total:

The 2 bomb types of the Aliens, mentioned earlier.

The last type is the type the player will be able to 'shoot'.

Score-board:

Throughout the game the score will be kept track of and shown at the very top of the

'canvas'. Once the game is over the user will be asked to input 3 letters to be their player

name and once they submit the name the score will be compared to a file containing the

top 5 scores, if it is higher than any of the scores it will replace them/slot in and shift

down the list, removing the new 11th highest score. Any repetitions of the scores will

cause the names to be added together in a list style, e.g. YAN, BEN. This will not occur if

the name if the same, if so no change will be made.

**Implementation**:

This project will be made using the object-oriented programming language Python.

To make the GUI I will be using the package Pyganim/Pygame.

Aliens:

There will be 3 types of aliens, but they will all have the same underlying behaviour.

This means the best way to implement this is by having an 'Alien' super-class, containing

the code that controls the behaviour that appears in all the alien (movement, shooting,

hit detection, death animation, etc...), and then using this class to create all 3 subclasses

which will contain the type specific info (points awarded sprite images, etc...).

There will also be a 'mothership' sprite this will have its own class as its behaviour is

different to the other aliens.

Player:

The player will have its own class that will be completely self-contained.

The player will be able to move right and left using the arrow keys and shoot by pressing

the spacebar.

The death animation will consist of 2 images switching back and forth a couple times

before the sprite disappears and the respawn method is called.

Barricades:

The barricades will be part of the background image (coloured green), then when the

projectile detect they're touching the colour green 'above' the co-ordinates of the player,

the image of the projectile will change to one of 2 black masks and will stop moving, thus

making part of the barricade black.

Projectiles:

There will be 3 types of projectile in total but they will all have the same underlying

behaviour, therefore I will make a super-class called Projectiles containing the code that

controls movement, then I will have 3 sub classes containing the code controlling the

direction, speed, image, etc...

Score-board:

The scores will be stored in a plain text file and will be sorted using an array of objects.

To check scores the find minimum, search and sort algorithms will be used.

**Requirements Met?:**

Validating inputs:

The player will only move on left and right arrow presses and shoot on spacebar presses.

The score-board will only allow alphabetical characters.

Interfacing with stored data:

The program will store the top 5 in a text file and will taking in that data and manipulate it

when displaying the score-board it will then also save the top 5 in the same file.

Binary search or sorting algorithms:

A Binary search and sorting algorithm will be used on the scores when considering adding

a new score.

Array of objects:

An array of objects will be used to store the high score data while being manipulated.