Project Proposal

Concept:

This project will be a game of space invaders. A game where you control a tank at the bottom of the screen and you have to use the tank to destroy aliens that are moving from side to side and down before they reach the bottom of the screen, each time they move down, their movement speed will increase. You must also avoid the bombs and shots from the aliens that will take one of your lives away (you will only have 3), to help with this there will be 4 barriers you can hide behind, but they will slowly be destroyed as they get hit by more and more shots. Once a level has been cleared the player will gain a life and the next level will load (which will be the same but harder by having the aliens move faster).

Target Audience:

Age: 8-16

Gender: Any

Interests: playing video games, computers, classic arcade games.

Other: Must own a computer, will have to be relatively skilled with their computer as the game uses python to run so they will have to have python installed or be able to install it.

Requirements Met:

Validating inputs:

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

The scoreboard will only allow alphabetical characters.

Interfacing with stored data:

The program will store the top 5 in a text file and will take in that data and manipulate it when displaying the scoreboard 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.

Research

Feasibility study:

Technical:

* The game will need to be object orientated both to make it easier to program and to meet some of the requirements. Python will be used for this as it is object orientated, simple to install and I already know the syntax.
* The game will need to have a graphical interface, for this Pygame and Pyganim (python libraries) will be used, this requires python and Pim (a software installation package) to be installed. This is feasible as Pim comes included in the most recent python package and installing Pygame and Pyganim only takes a one-line command.
* I will also require resources detailing how to use the Pygame and Pyganim libraries. This is feasible as there are many websites detailing the methods and variables provided by the libraries and how to use them. For this, I will be using the Pygame home website.

Economic:

* The Project is economically feasible as it will not cost anything to produce.

Legal:

Data Protection:

* I will be collecting high-scores
* I will be collecting names (3 characters max)

Copyright, Designs and Patents Act:

* I will need to use a different name for the game due to copyright
* I will need to use different graphics due to copyright
* I will need to use different audio due to copyright

Schedule:

* The project is of suitable complexity to be achievable in the allocated timeframe.

Survey:

Analysis of survey results:

Analysis:

Knowing Spaces invaders:

The majority of the people surveyed

knew about scape invaders, about 83%,

and only about 17% of the people

surveyed either didn’t know about the

game or were unsure if they knew the

game.

Playing Video Games:

Boys:

The majority of the boys surveyed play

video games, 60%, with 36% who don’t

play video games and 4% who were

unsure.

Girls:

The majority of the girls surveyed don’t

play video games, 76%, with only 24% who do play video games.

Conclusion:

While there was little to no difference between the genders when it came to knowing about the game, the difference when it came to playing video games suggests that the majority of the users for this game will be male (this does not exclude girls from the end user group).

Analysis:

Wanting high-score table:

83% of the people surveyed want a

high score table with 14% being unsure

and only 3% saying they didn’t want a

high score table.

Places on the Highscore Table:

The majority of the people surveyed

wanted 10 places on the table, 52%,

and of the people who didn’t want 10

places half wanted 5 places and the rest

wanted an other number.

Conclusion:

A high score table will be implemented with 10 places.

Analysis:

The majority of the people surveyed wanted background music, 86%, with

12% not wanting background music and 2% being unsure. Of the 86% who

do want background music a couple people noted to make sure that the

music was not too irritating.

Conclusion:

I will be implementing background music but must be careful what music I

will be using, I will be running another survey on the music choices.

Analysis:

The majority of the people surveyed wanted an option to play the game

with the original arcade graphics, 50%, with 31% not wanting the original

graphics and 19% not minding.

Conclusion:

I will be attempting to implement an option to play the game with original

arcade graphics, this may present some copyright issues.

Analysis:

Laptop:

The majority of laptop users use

windows.

Phone:

The majority of smartphone users use

an iPhone.

Desktop:

The desktop users were split 50/50

between windows and apple.

Table:

The majority of Tablet users use iPads.

Device Used:

The majority of people surveyed would use their smartphone to play the game, 49%, 36% would use their laptop, 10% would use their tablets, 2% would use their console and 3% have not device preference.

Conclusion:

The majority of people surveyed use apple products and would use their smartphone to play the game.

This suggests the I should write the program to work on an iPhone but this would require me to learn a new language and how to use a new programming environment reducing the feasibility of the project.

The second device the people surveyed said they would use was their laptop and as the percentage of laptop users who used windows was 78%, this suggests that I should write the program to work on a windows laptop, this is more feasible as it would not require me to learn a new language or how to use a new programming environment.

Project Plan

Gantt Chart:

Resources:

Requirement Specification

Purpose:

Scope and Boundaries:

Scope:

* I will be implementing background music and the option to turn it off
* I will be implementing multiple graphics options
* I will be implementing a scoreboard
* I will be implementing increasing difficulty (with unlimited levels)
* I will be implementing the ‘mothership’ at the top of the ‘game page’

Boundaries:

* I will not be implementing boss levels
* I will not be implementing multiplayer options
* I will not be implementing upgrades to the player character/power-ups
* I will not be implementing multiple player characters
* I will not be making the game ‘open world’
* I will not be adding ‘Easter eggs’

End Users:

The users will be children/teenagers from the ages of 8-17 who enjoy video/arcade games and have an interest in old games. Nostalgic 50-year-olds may also be part of the end user group as the original space invaders game came out in 1978.

User Requirements (of the program):

* The game must not ‘lag’
* The game must not be too difficult to play
* There must be a page that explains how to play the game

User Requirements (of the user):

* Must own a windows computer
* Must have a keyboard and mouse
* Must have python installed
* Must have Pygame installed

Functional Requirements:

Inputs and Outputs:

Inputs:

* Alpha Keys (including spacebar)
* Arrow Keys
* Mouse movement/position
* Mouse
* Clicks

Outputs:

* Highscore file
* Graphical interface
* Background music audio

Functionality:

Aliens:

There will be 5 layers of aliens.

The aliens will be held in a 2D array of objects.

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'.

Scoreboard:

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 coordinates 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 projectiles 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 subclasses containing the code controlling the direction, speed, image, etc...

Scoreboard:

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.