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 (a python library) 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 library. 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. The software being used to produce this project is free, the purchase of a computer to produce the project is not required and no licence is required to develop this project.

Legal:

Data Protection:

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

Copyright, Designs and Patents Act:

* The copyright for the original character designs Is owned by taito (a Japanese game company), this may pose an issue.
* I will need to use public domain music and sound effects for the audio to avoid copyright issues.

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:

Items I will require to complete this project are:

* Pen and Paper
* Computer (with monitor, keyboard and mouse)
* A wide range of end users to survey
* SQA Project information resources
* Appropriate software:
  + Python
  + Pip
  + Pygame
  + Atom
  + Word
  + Excel
  + Balsamiq
  + Gantt Project
  + Github
* A web browser to research how to use the software I will be using.

Requirement Specification

Purpose:

The program will be a game of space invaders. The purpose of this game is to entertain the users, encourage an interest on older/classic video games and possibly provoke a feeling of nostalgia in older players.

Scope and Boundaries:

Scope:

* The users will be able to control the side to side movement of a game sprite at the bottom of the ‘game page’
* The user will be able to ‘shoot aliens’ using the game sprite under their control
* The ‘aliens’ will attack the player using slow and fast ‘projectiles’ that the player must avoid by moving out the way or by hiding behind barriers
* The barriers the player can hide behind will ‘wear away’ from being hit by projectiles from either side, increasing the difficulty of the game.
* There will be a special ‘alien’ that moves across the top of the ‘game page’ occasionally that awards the player more points for destroying
* As the aliens move down the screen their movement speed will increase, increasing the difficulty of the game.
* Once the player has ‘cleared the skies of aliens’ the ‘round’ will be over and the game will ‘reset’ with a now set of ‘aliens’ but the new ‘aliens’ will have a faster initial movement speed.
* There will be a scoreboard that keeps track of the 10 best scores achieved by players along with 3 characters they put in
* There implementing background music and the option to turn it off
* There will be multiple options for how the game looks, a classic version with the original graphics and a more modern version with higher quality graphics.

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’
* I will not be adding Language options

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. The users must have a basic knowledge of how to use a computer and of the layout of a standard qwerty keyboard.

User Requirements:

* The game must not ‘lag’
* The game must not be too difficult to play

Functional Requirements:

* There must be a page that explains how to play the game
* Must own a modern windows computer
* Must have a keyboard, mouse and monitor
* Must have python installed
* Must have Pygame installed

Inputs and Outputs:

Inputs (with resultant effects/outputs):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Section | Dynamic Element | Types | Input | Effect/Output |
| Home Page | Buttons | Start | Mouse click on button | Switches section to the game |
|  |  |  | Mouse hover | Enlarges the button image |
|  |  | Instructions | Mouse click on button | Switches section to Instructions page |
|  |  |  | Mouse hover | Enlarges button image |
|  |  | Settings | Mouse click on button | Switches section to the settings page |
|  |  |  | Mouse hover | Enlarges button image |
| Instructions Page | Buttons | Back | Mouse click on button | Switches section to the home page |
|  |  |  | Mouse hover | Enlarges button image |
| Settings Page | Buttons | Back | Mouse click on button | Switches section to the homepage |
|  |  |  | Mouse hover | Enlarges button image |
|  |  | Graphics | Mouse click on button | Changes the button text from/to ‘classic’/’modern’.  Changes the graphics used in the game |
|  |  |  | Mouse hover | Brings up setting options at the side with a description of what they are |
|  |  | Background music selection | Mouse click on button | Changes the button text from/to ‘song1’/’song2’/’song3’… and plays a short section of it. Changes the background music used in the game. |
|  |  |  | Mouse hover | Brings up setting options at the side with a description of what they are |
|  |  | Background music on/off | Mouse click on button | Changes the button text from/to ‘On’/’Off’.  Toggles whether or not the game will have background music. |
|  |  |  | Mouse hover | Brings up setting options at the side with a description of what they are |
|  |  | Sound effects on/off | Mouse click on button | Changes the button text from/to ‘on’/’off’.  Toggles whether or not the game will have sound effects. |
|  |  |  | Mouse hover | Brings up setting options at the side with a description of what they are |
| Game Page | Player sprite | Movement | A/D or /  Will only take in one input at a time | Changes the position of the player sprite in the game field |
|  |  | Shooting | ‘SpaceBar’ | Spawns a player shot above the centre of the player sprite. |
|  | Score display |  | High scores file | Will take in the high scores file to display the highest score on the top right of the page. |
|  | File check |  | High scores file | Will take in the high scores file to check current score against, if current score is higher than any score in the file then the page will switch to the high score input page otherwise it will skip to the high score display page. |
| High score input page | Initials input | Input | Alpha keys, only 3 characters will be taken in | The character of the key pressed will be placed in the selected initial place and the next initial place will be selected, |
|  |  |  | / | Will move the selected initial place appropriately. |
|  |  |  | Enter | Will start a validation on the entire initial set to make sure all 3 characters have been used. If valid will send initials to the save high score procedure and the high score file will be updated/created. |
|  |  |  | Backspace | Removes the character from the initial place selected, if place is already empty, nothing will happen |
| High score Display Page | Score Display |  | High scores file | Will take in high scores file to produce array of objects to be displaed |
|  | Buttons | Try again | Mouse click on button | Switches section to game page and resets the game page |
|  |  |  | Mouse hover | Enlarges button image |
|  |  | Home | Mouse click on button | Switches section to the home page and resets the game page |
|  |  |  | Mouse hover | Enlarges button image |
| All Pages | Buttons | Exit | Mouse click on button | The program will end |

Outputs:

|  |  |
| --- | --- |
| Section | Output |
| Home Page | All images will display appropriately e.g. background image |
| Instructions Page | All images will display appropriately e.g. background image |
| Settings Page | All images will display appropriately e.g. background image |
|  | Audio when changing the background music/sound effects |
| Game Page | All images will display appropriately e.g. background image |
|  | The current and top scores display at the top of the page |
|  | Background music |
|  | Sound effects |
| High score Input Page | All images will display appropriately e.g. background image |
|  | File containing high score details |
| High score Display Page | All images will display appropriately e.g. background image |

Test Plan

Home Page:

Functionality:

|  |  |  |
| --- | --- | --- |
| Function | Testing Method | Result? |
| Cursor should be visible. | Start the program and move mouse around to check if cursor is visible. |  |
| Images/Text should display correctly. | Start the program and check the pages against their wireframes to check positioning, and source images/text to check if the correct images/text are used and if the images/text look right. |  |
| Settings button should switch the page to the settings page when clicked. | Click on the button to check if the page changes and then use the wireframes to check that the correct page is now active. |  |
| Instructions button should switch the page to the instructions page when clicked. | Click on the button to check if the page changes and then use the wireframes to check that the correct page is now active. |  |
| Game button should switch the page to the game page when clicked. | Click on the button to check if the page changes and then use the wireframes to check that the correct page is now active. |  |
| Exit button should close the program when clicked. | Click on the button to check if the program stops running. |  |
| All buttons should enlarge slightly when the cursor is over them (not including the exit button). | Move the cursor over the buttons to check if the button gets larger. |  |

Usability:

A user survey will be taken on the clarity and usability of the layout.

|  |  |
| --- | --- |
| Aspect | Achieved? |
| Consistency: is the same font used, same look throughout the program? |  |
| Clarity: are the elements labelled in an understandable way? |  |
| Ease of use: are the buttons an appropriate size/do the buttons have appropriate positioning to make them easy to click? |  |

Settings Page:

|  |  |  |
| --- | --- | --- |
| Function | Testing Method | Result? |
| Cursor should be visible. | Start the program and move mouse around to check if cursor is visible. |  |
| Images/Text should display correctly. | Start the program and check the pages against their wireframes to check positioning, and source images/text to check if the correct images/text are used and if the images/text looks right. |  |
| Music Button should toggle whether the music is playing, when clicked.  The text on the button will also change to show the state of the music. | Code will be added that prints the state of the music to the console. Click on the button and check the console to see if the music state and the text on the button have changed as expected. |  |
| Music type Button should toggle the music that is playing, when clicked.  The text on the button will also change to show the music type selected. | Code will be added that prints the music file name to the console. Click on the button and check the console to see if the music file name and the text on the button have changed as expected. |  |
| Sound Effects Button should toggle whether the sound effects are used, when clicked.  The text on the button will also change to show the state of the sound effects. | Code will be added that prints the state of the sound effects to the console. Click on the button and check the console to see if the sound effects state and text on the button have changed as expected. |  |
| Graphics Button should toggle the images that are used, when clicked.  The text on the button will also change to show the state of the graphics. | Code will be added that prints the state of the graphics to the console. Click on the button and check the console to see if the graphics state and text on the button have changed as expected. |  |
| Back button should switch the page to the home page when clicked. | Click on the button to check if the page changes and then use the wireframes to check that the correct page is now active. |  |
| Exit button should close the program when clicked. | Click on the button to check if the program stops running. |  |
| All buttons should enlarge slightly when the cursor is over them (not including the exit button). | Move the cursor over the buttons to check if the button gets larger. |  |

Usability:

A user survey will be taken on the clarity and usability of the layout.

|  |  |
| --- | --- |
| Aspect | Achieved? |
| Consistency: is the same font used, same look throughout the program? |  |
| Clarity: are the elements labelled in an understandable way? |  |
| Ease of use: are the buttons an appropriate size/do the buttons have appropriate positioning to make them easy to click? |  |

Instructions Page:

Functionality:

|  |  |  |
| --- | --- | --- |
| Function | Testing Method | Result? |
| Cursor should be visible. | Start the program and move mouse around to check if cursor is visible. |  |
| Images/Text should display correctly. | Start the program and check the pages against their wireframes to check positioning, and source images/text to check if the correct images/text are used and if the images/text looks right. |  |
| Back button should switch the page to the home page when clicked. | Click on the button to check if the page changes and then use the wireframes to check that the correct page is now active. |  |
| Exit button should close the program when clicked. | Click on the button to check if the program stops running. |  |
| All buttons should enlarge slightly when the cursor is over them (not including the exit button). | Move the cursor over the buttons to check if the button gets larger. |  |

Usability:

A user survey will be taken on the clarity and usability of the layout.

|  |  |
| --- | --- |
| Aspect | Achieved? |
| Consistency: is the same font used, same look throughout the program? |  |
| Clarity: are the elements labelled in an understandable way. Are the instructions clear and easy to understand? |  |
| Ease of use: are the buttons an appropriate size/do the buttons have appropriate positioning to make them easy to click? |  |

Game Page:

Functionality:

|  |  |
| --- | --- |
| Function | Does it work? |
| Score display: should display the current score and the highest score at the top of the page |  |
| Alien types: there should be 3 different types of alien, each have a different point value, the lowest being on the lowest 2 rows of alien and giving 10 points, the middle being on the next 2 rows and giving 20 points and the highest being on the top row and giving 30 points. There will also be a mother ship alien that occasionally moves across the top of the page and gives either 50, 100, 150 or 300 points |  |
| Alien movement: The aliens should all move together, moving from side to side then when the first alien reaches an edge shifting down a row and moving to the other side and increasing the movement speed. |  |
| Alien projectiles: there will be 2 different types of projectile, fast(bolt) and slow(arrow). The bolt moves faster than the arrow and has a different image/animation. Both projectiles will instantly ‘kill’ the player when they come into contact (with the player). The colour of the projectiles changes depending on the y position, white while in the middle section, green while at the bottom section (from the top of the bunkers and bellow). |  |
| Player movement: the player will only be able to move side to side and will not be able to move past the edges of the page. The controls will be A/D or / . A and the left arrow key will move the player to the left. D and the right arrow key will move the player to the right. |  |
| Player projectile: there will only be one type of player projectile (shot), the shot will instantly ‘kill’ any alien it comes into contact with and will then disappear so as not to accidentally ‘kill’ more than one alien. The colour of the shot will change depending on the y position, green while in the bottom section, white while in the middle section and red in the top section. The space bar will ‘shoot’ the projectile. |  |
| Player death: when the player dies the game will pause while the player death animation plays and the player life count will decrement, then the player character will reappear in the ‘home’ position (x middle of the page) and the game will continue. When the player runs out of lives the game will end. |  |
| Bunker: There will be 4 bunkers, these bunkers will stop any of the projectiles but when a bunker stops a projectile the projectile will ‘blow up’ a section of the bunker. |  |
| End game: when the game is over the program will check if the current score is higher than any of the scores in the high score file, if so it will switch to the high score input page, if not it will switch to the high score display page. |  |
| Exit button click: halts the program. |  |
| Cursor: should not be visible |  |

Usability:

A user survey will be taken on the clarity and usability of the layout.

|  |  |
| --- | --- |
| Aspect | Achieved? |
| Consistency: same look throughout the program. |  |
| Ease of use: is there too much lag on the controls, are the aliens to small to hit, do the projectiles move too fast, is the player movement speed too fast or too slow. |  |

High Score Input Page:

Functionality:

|  |  |
| --- | --- |
| Function | Does it work? |
| Initials input: The currently selected initials place will be indicated but a flashing underline of the -place. alpha key presses will put that character into the currently selected initial place. The back space key will remove the character from the currently selected initials place. The enter key will enter the initials (see below). |  |
| Initials enter: checks to make sure 3 characters have been input. If the check is passed the sort function will be called to place the new high score and then it will be added to the file |  |
| Skip button click: will skip in putting a new high score and switch to the high score display page. |  |
| Exit button click: halts the program. |  |
| All buttons hover (excluding the exit button): enlarge image/text |  |
| Images: display correctly. |  |
| Cursor: should be visible |  |

Usability:

A user survey will be taken on the clarity and usability of the layout.

|  |  |
| --- | --- |
| Aspect | Achieved? |
| Consistency: is the same font used, same look throughout the program? |  |
| Clarity: are the elements labelled in an understandable way? Is the indication for the currently selected initial place clear enough. |  |
| Ease of use: are the buttons an appropriate size/do the buttons have appropriate positioning to make them easy to click? |  |

High Score Display Page:

Functionality:

|  |  |
| --- | --- |
| Function | Does it work? |
| New Game button click: resets the game page and changes page to game page. |  |
| Home button click: resets the game page and changes page to home page. |  |
| Exit button click: halts the program. |  |
| All buttons hover (excluding the exit button): enlarge image/text |  |
| Images: display correctly. |  |
| Cursor: should be visible |  |

Usability:

A user survey will be taken on the clarity and usability of the layout.

|  |  |
| --- | --- |
| Aspect | Achieved? |
| Consistency: is the same font used, same look throughout the program? |  |
| Clarity: are the elements labelled in an understandable way? |  |
| Ease of use: are the buttons an appropriate size/do the buttons have appropriate positioning to make them easy to click? |  |

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.