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.
* Webistes:
  + <http://www.cogsci.rpi.edu/~destem/gamedev/pygame.pdf>
  + The Pygame Wiki

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 |
|  | File |  | Settings file | Takes in and reads the settings stored in the file mapping it to a set of variables. |
| 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 |
|  |  | Exit | The escape button is clicked | 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 |
|  | Settings file with changed settings (if nessasary) |
| 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:

This page is event driven reducing the need for input validation.

|  |  |  |
| --- | --- | --- |
| 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 page against its wireframe 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 and escape key should close the program when clicked. | Click on the button to check if the program stops running. Press the escape key 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. |  |
| Should take in settings file and read the content into a set of variables. Should validate the files existence | Use a break point to have a look at the values of the variables. Delete the file to makesure a new file is created in its place |  |

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:

Functionality:

This page is event driven reducing the need for input validation.

|  |  |  |
| --- | --- | --- |
| 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 page against its wireframe 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 and escape key should close the program when clicked. | Click on the button to check if the program stops running. Press the escape key 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. |  |
| Should take in settings file and read the content into a set of variables. Should validate the files existence | Use a break point to have a look at the values of the variables. Delete the file to makesure a new file is created in its place |  |
| Settings file should be output (only when a setting has been changed) when the back button is clicked | Change a setting, click the back button and then look at the file to ensure that the corresponding setting has changed. |  |

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:

This page is event driven reducing the need for input validation.

|  |  |  |
| --- | --- | --- |
| 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 page against its wireframe 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 and escape key should close the program when clicked. | Click on the button to check if the program stops running. Press the escape key 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:

This page is event driven reducing the need for input validation, the required validation will be mentioned.

|  |  |  |
| --- | --- | --- |
| 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 page against its wireframe to check positioning, and source images/text to check if the correct images/text are used and if the images/text looks right. |  |
| The current and highest scores should be displayed at the top of the screen | Start the game and check against the wireframe to check positioning, input a high score into the scores file and print the current score to a console to check the scores being displayed are correct. |  |
| There should be 3 different types of alien each section will hold a different type of alien, the lowest 2 rows of aliens and give 10 points, the next 2 rows give 20 points and the top row gives 30 points. | Create a version of the game where the aliens do not move or ‘fight back’. ‘kill’ a column of aliens checking the current score increases appropriately for each alien ‘death’. |  |
| There should also be a mother ship alien that occasionally moves across the top of the page and gives either 50, 100, 150 or 300 points. | Create a version of the game where the aliens are replaced with mother ships that do not move. ‘kill’ the motherships checking that the current score increases appropriately. |  |
| The aliens should all move together, moving from side to side until any alien reaches an edge at which point the aliens will shift down a row and change the direction of their movement (side to side). The movement speed should increase. When the aliens reach the ‘bunkers’ the game should end. | Create a version of the game where the aliens cannot ‘fight back’. Start the game and observe the aliens moving making sure the movement is correct. Check that the game does end when the first alien reaches the bunker. |  |
| There will be 2 different types of projectile the aliens can fire. Fast(bolt) and slow(arrow) projectiles. The different projectiles will have different images and movement speeds (they move down at this speed). Both projectiles will instantly ‘kill’ the player when they come into contact (with the player) decreasing the lives count by one. 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 below). | Start the game and observe the alien projectiles to ensure they behave as expected. |  |
| There will be a system in place that makes the aliens’ projectiles are shot accurately as opposed to randomly, although there will be some randomness involved. | Start and observe the game to ensure the aliens shoot the projectiles as expected, moving the player around to ensure the behaviour is consistent. |  |
| 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. Only one input will be allowed at a time ‘A’ or ‘D’ or left or right, all other inputs will be ignored. | Create a version of the game without aliens. Use the ‘A’ and ‘D’ keys to move the player character from edge to edge ensuring the player character stops as expected. Attempt to use both ‘A’ and ‘D’ at the same to make sure the player character does not ‘glitch’. Do the same using the left and right arrow keys. Use both the ‘A’ and left arrow key at the same time to ensure the speed of movement does not change. Do the same with ‘D’ and the right arrow key. Press any other keys (excluding the escape key) to ensure nothing happens. |  |
| There will only be one type of player projectile (shot), the shot will move upward at a constant speed and instantly ‘kill’ any alien it comes into contact with and will then disappear so as not to accidentally ‘kill’ more than one alien, the shot will also disappear when in contact with a bunker. 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. When shot the x co-ordinate of the projectile will be that of the player at the time of shooting. There will be a 1 second shot ‘cool down’ between shots although no limit to the number of shots. The space bar will ‘shoot’ the projectile. | Create version of the game where the aliens don’t move or ‘fight back’. Press the spacebar to ensure the projectile appears in the expect place, move the player around and do the same. Attempt to ‘shoot’ as frequently as possible to check the cool-down time works and that there is no limit to the number of shots. Shoot an alien to ensure the shot disappears when in contact and the alien ‘dies’. Shoot a bunker to ensure it the shot disappears. |  |
| 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. | Start the game and get hit by a projectile to ensure the correct behaviour is carried out at player death. Repeat until the lives counter is 0 to ensure the game ends when expected. |  |
| There will be 4 ‘bunkers’. These bunkers will stop any type of projectiles (player shots included) but when a bunker stops a projectile the projectile will ‘blow up’ a section of the bunker. | Start the game and move the player character underneath a bunker, shoot the bunker to ensure the projectile stops and the bunker has an appropriate section ‘blown up’. Create a version of the game where a projectile spawns where the mouse is clicked, spawn all three types of projectile over a bunker to ensure all disappear and ‘blow up’ and appropriate section of the bunker. |  |
| 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. | Create a version of the game where the current score is input able and the ‘game’ can be ended using a button, set a high score in the high score file and then input a lower value to check the correct page is switched to then a higher value to check the correct page is switched to. Use the wireframes to check that the correct page is now active. |  |
| Exit button and escape key should close the program when clicked. | Click on the button to check if the program stops running. Press the escape key to check if the program stops running. |  |
| Should take in settings file and read the content into a set of variables. | Use a break point to have a look at the values of the variables. |  |

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 or aliens move too fast, is the player movement speed too fast or too slow. |  |

High Score Input 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 page against its wireframe to check positioning, and source images/text to check if the correct images/text are used and if the images/text looks right. |  |
| Skip button should switch the page to the high score display 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. |  |
| The currently selected initials place will be indicated but a flashing underline. Only alpha key presses will put a character into the currently selected initial place. The back space key will remove the character from the currently selected initials place. The left and right arrow keys will change the selected initials place | On the page use the right and left arrow keys to select an initial place and ensure the underline is flashing. Press any Alpha key to ensure the correct character is input in the selected initials place (expected). Press any non-alpha keys to ensure the character is not input (extreme).Not done yet |  |
| Exit button and escape key should close the program when clicked. | Click on the button to check if the program stops running. Press the escape key 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? 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 | 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 page against its wireframe to check positioning, and source images/text to check if the correct images/text are used and if the images/text looks right. |  |
| Home 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. |  |
| New Game button should reset the game page then 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 and that the game page has been reset. |  |
| Exit button and escape key should close the program when clicked. | Click on the button to check if the program stops running. Press the escape key 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:

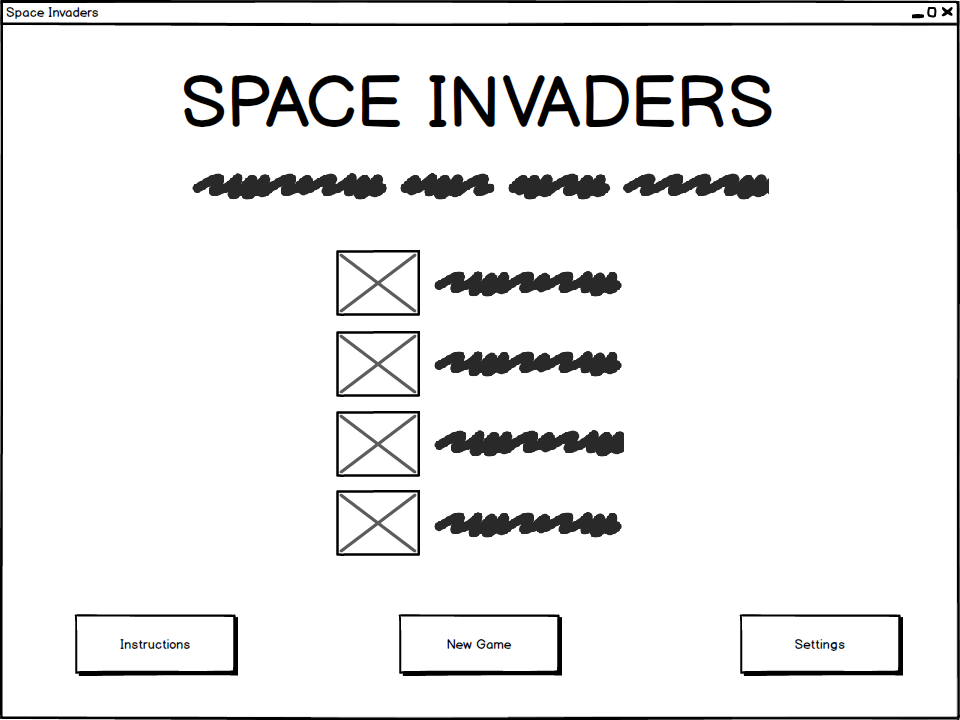
A user survey will be taken on the clarity and usability of the layout.

|  |  |
| --- | --- |
| 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? |  |

UI Design

Wireframes:

Home Page:



Title

Sub-title

Alien images

Alien details

Background

Buttons

Title:

Font-size: 72px

Colour: white

Font-family: cosmic-aliens (ca)

Position (corner): (180, 65)

Content: “SPACE INCADERS”

Sub-title:

Font-size: 32px

Colour: white

Font-family: cosmic-aliens (ca)

Position(corner): (192, 173)

Content: description of below (what is being shown)

Alien images:

Size: 86x66

Number: 4

Positions(centre): (380, 283), (380, 364), (380, 444), (380, 523)

Content: respective alien sprite images

Alien details:

Font-size: 28px

Colour: white

Font-family: cosmic-aliens (ca)

Positions(centre): (434, 271), (434, 351), (434, 420), (434, 512)

Content: the point value of respective aliens

Background:

Size: 960x720px

Image: black with white spots to mimic stars

Buttons:

Size: 164x62px (normal), 184x78px (large)

Positions(centre): (156, 645), (548, 645), (821, 645)

Content: 1-2 word description of function.

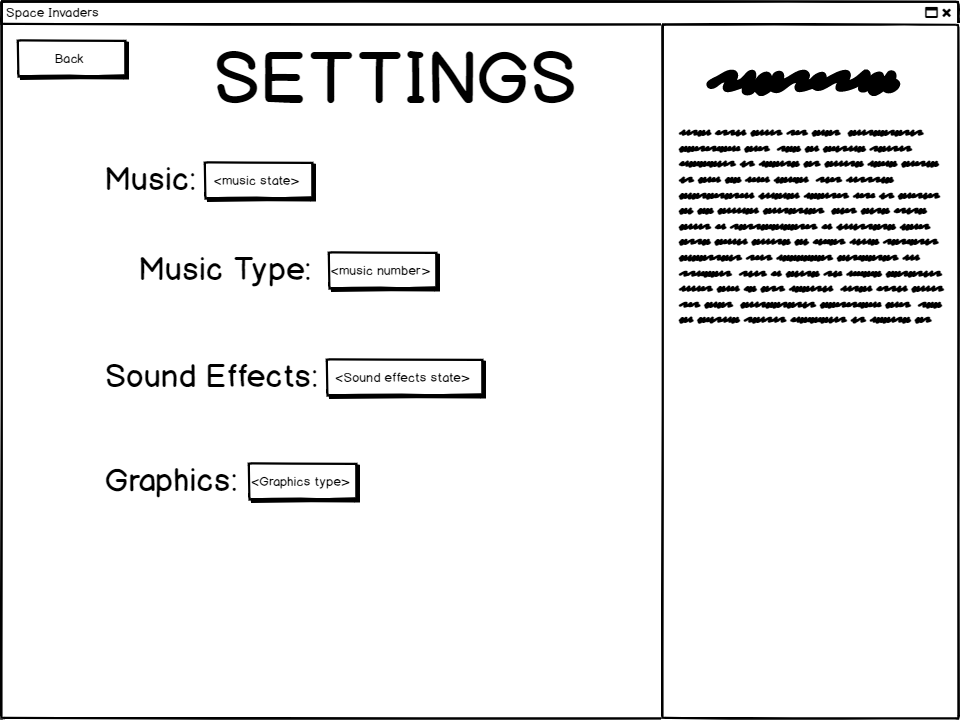
Font-size: 13px

Font-family: cosmic-aliens (ca)

Input: click detection, hover detection.

Output: size will change size when cursor is hovering over it, will change the page when clicked (to appropriate respective page).

Settings Page:



Title

Buttons Description Title

Labels Description Text

Background

Title:

Font-size: 72px

Colour: white

Font-family: cosmic-aliens (ca)

Position(corner): (213, 42)

Content: “SETTINGS”

Buttons:

Size: 113x42px (normal), 133x58px (large)

Positions(centre): (72, 59), (259, 181), (383, 271), (381, 378), (303, 482)

Content: 1-2 word description of function.

Font-size: 13px

Font-family: cosmic-aliens (ca)

Input: click detection, hover detection.

Output: size will change size when cursor is hovering over it, will change the sate of its respective setting/ change the page to the homepage (back button).

Labels:

Font-size: 32px

Position (corner): (105, 161), (139, 251), (105, 358), (105, 462)

Colour: white

Font-family: cosmic-aliens (ca)

Content: name of respective setting

Background:

Size: 960x720

Image: black with white spots to mimic stars and white bar 299px from the left.

Description Title:

Font-size: 28px

Position (corner): (706, 69)

Colour: white

Font-family: cosmic-aliens (ca)

Content: name of respective setting

Description Text:

Font-size: 16px

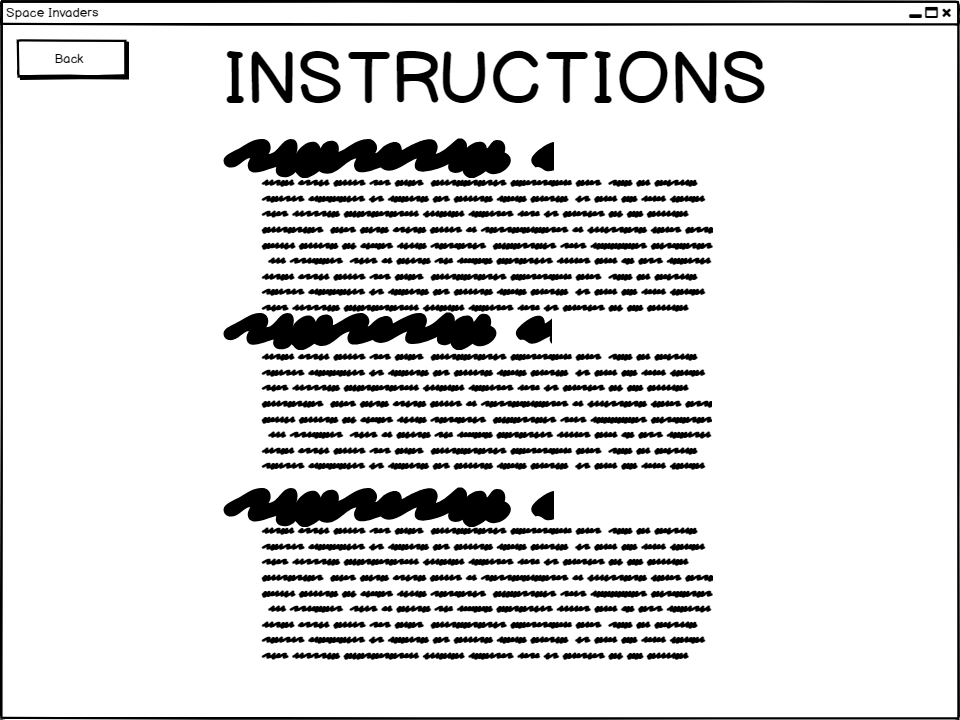
Position (corner): (676, 126)

Colour: white

Font-family: cosmic-aliens (ca)

Content: description of respective setting states

Instructions Page:



Title

Button

Instruction headers

Instruction body texts

Background

Title:

Font-size: 72px

Position (corner): (223, 42)

Colour: white

Font-family: cosmic-aliens (ca)

Content: “INSTRUCTIONS”

Button:

Size: 113x41px (normal), 133x57px (large)

Position (centre): (72, 59)

Content: “Back”

Font-size: 13px

Font-family: cosmic-aliens (ca)

Input: click detection, hover detection.

Output: changes the page to the homepage.

Instruction Headers:

Font-size: 32px

Positions (corner): (223, 138), (223, 312), (223, 287)

Colour: white

Font-family: cosmic-aliens (ca)

Content: summary of what is being explained below.

Instruction Body Texts:

Font-size: 18px

Positions (corner): (259, 176), (259, 350), (259, 524)

Colour: white

Font-family: cosmic-aliens (ca)

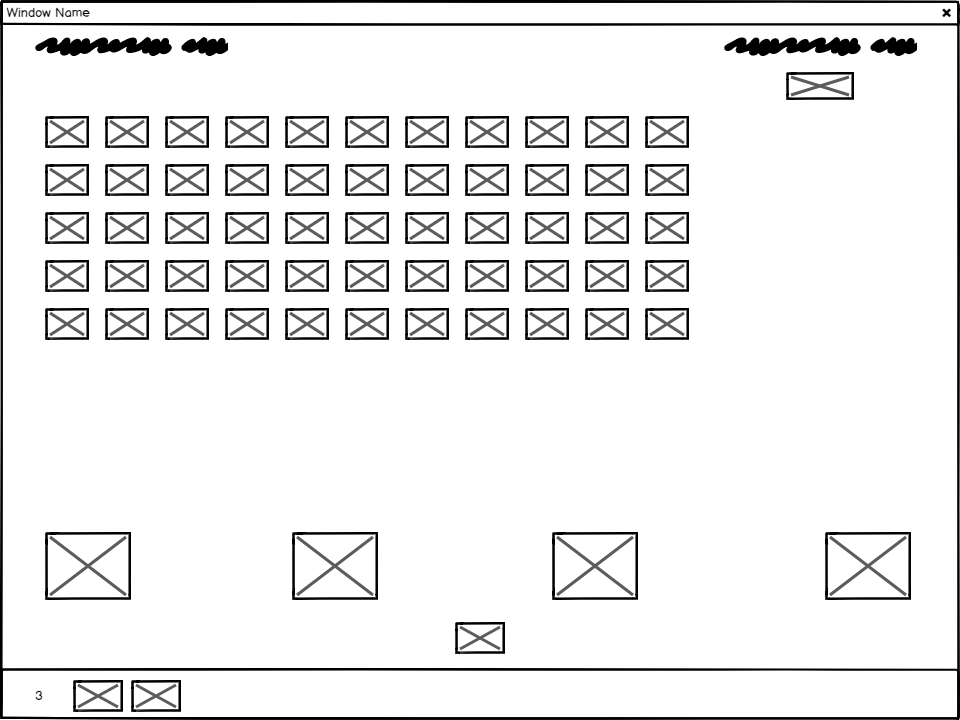
Content: explanation of controls or game mechanic.

Background:

Size: 960x720px

Image: black with white spots to mimic stars

Game Page:



Current score Highest score

Mothership

Aliens Background

Bunkers

Player Sprite

Lives indicator Bottom Bar

Current score:

Font-size: 16px

Colour: white

Font-family: cosmic-aliens (ca)

Content: displays the current score

Highest score:

Font-size: 16px

Colour: white

Font-family: cosmic-aliens (ca)

Content: displays the highest score

Input: high score file

Background:

Size: 960x720px

Image: black with the bunkers on it

Input: settings variables

Output: background music

Bunkers:

Size: 86x68px

Special: will be part of the background for easier implementation

Mothership:

Size: 68x28px

Image: mothership sprite image

Output: sound effects

Aliens:

Size: 44x32px

Total number: 55

Separation (X): 16px

Separation (Y): 21px

Output: sound effects

Player Sprite:

Size: 50x32px

Image: player sprite image

Input: ‘A’, ’D’, left key and right key

Output: sound effects

Player Shot (not shown):

Size: 10x40px

Image: player shot

Output: sound effects

Lives indicator:

Font-size: 16px

Colour: white

Font-family: cosmic-aliens (ca)

Content: the number of lives left

Image Size: 50x32px

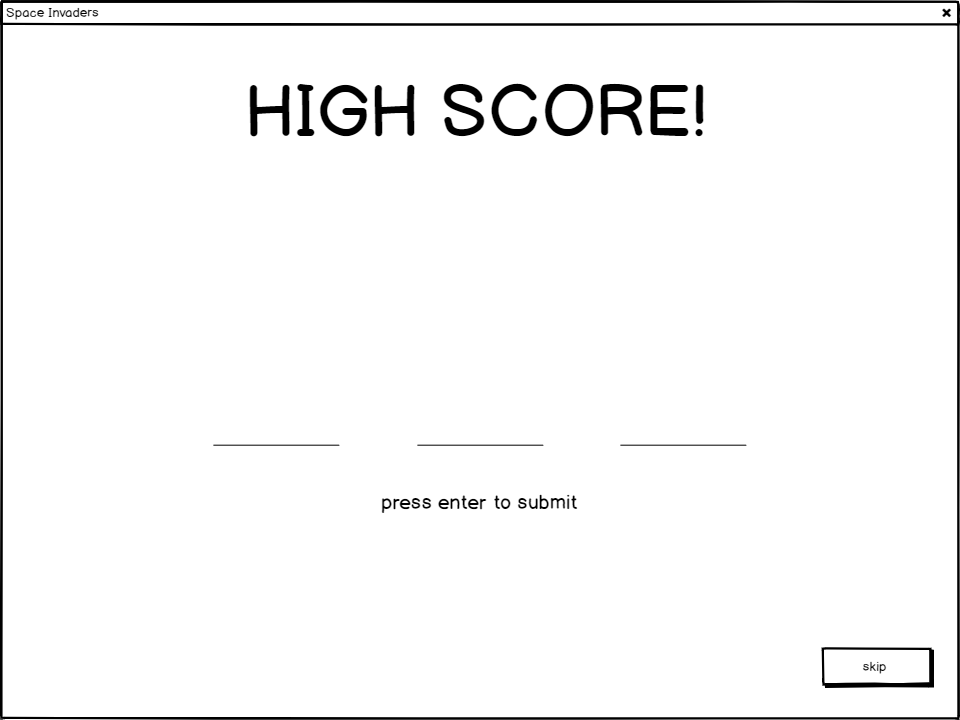
Number of images: number of lives left

Bottom Bar:

Size: 960x52px

Image: white bat at top, the rest black

High Score Input Page:



Title

Background

Initial spaces

Enter text

Skip button

Title:

Font-size: 72px

Position (corner): (245, 75)

Colour: white

Font-family: cosmic-aliens (ca)

Content: “HIGH SCORE!”

Background:

Size: 960x720px

Image: black with white spots to mimic stars

Input: enter key press, high score file

Output: high score file

Initial spaces:

Font-size: 180px

Positions (underlines) (centre): (275, 439), (480, 439), (683,439)

Font-family: cosmic-aliens (ca)

Underline size (width): 125px

Colour: white

Input: alpha key presses

Enter Text:

Font-size: 18px

Position (corner): (380, 489)

Colour: white

Font-family: cosmic-aliens (ca)

Content: “press enter to submit”

Skip Button:

Size: 113x42px (normal), 133x58px (large)

Position (centre): (876, 667)

Content: “skip”

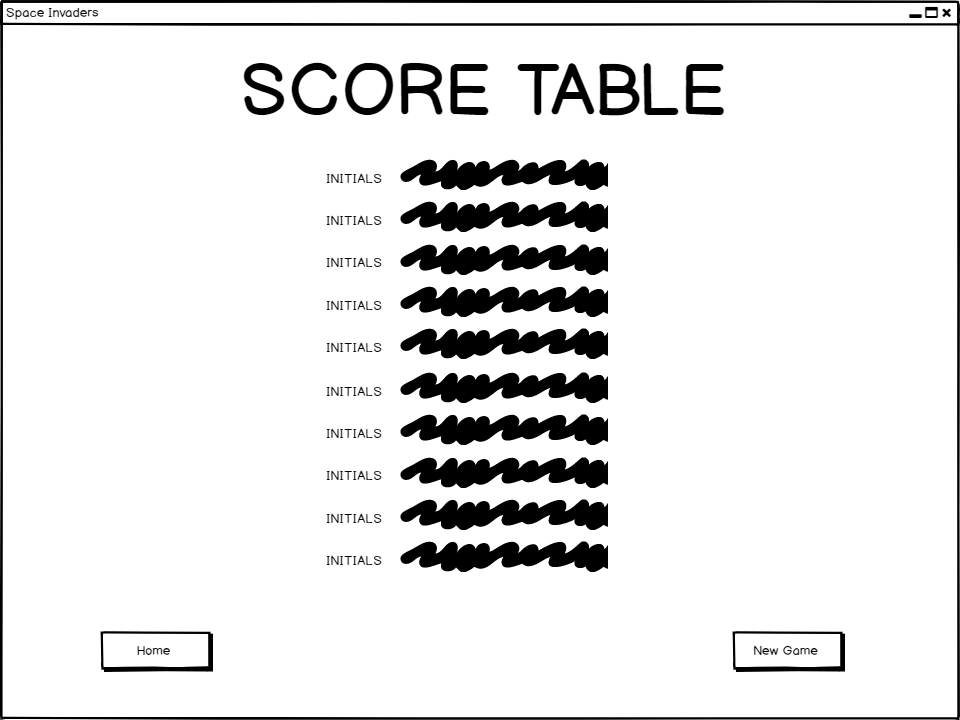
Font-size: 13px

Font-family: cosmic-aliens (ca)

Input: click detection, hover detection.

Output: changes the page to the high score display page.

High Score Display Page:



Title

High score table

Background

Buttons

Title:

Font-size: 72px

Position (corner): (240, 54)

Colour: white

Font-family: cosmic-aliens (ca)

Content: “HIGH SCORE!”

High Score Table:

Font-size: 13px

Font-family: cosmic-aliens (ca)

Content: a sorted list of the top ten highest score along

with their corresponding initials

Background:

Size: 960x720px

Image: black with white spots to mimic stars

Input: high score file

Buttons:

Size: 113x42px (normal), 184x78px (large)

Positions (centre): (156, 651), (788, 651)

Content: 1-2 word description of function.

Font-size: 13px

Font-family: cosmic-aliens (ca)

Input: click detection, hover detection.

Output: size will change size when cursor is hovering over it, will change the page when clicked (to appropriate respective page).

Psuedocode

Button Class:

Instance Variable:

Declare Position() As INTEGER

Declare Size() As INTEGER

Declare Content() As INTEGER

Methods:

hover\_Detect()

click\_Detect()

enlarge()

load\_Page()

Home Page:

1. Load Home Page
2. While Home page is loaded

3 btn\_Settings.click\_Detect()

4 btn\_Instructions.click\_Detect()

5 btn\_Start\_Game.click\_Detect()

6 btn\_Settings.hover\_Detect()

7 btn\_Instructions.hover\_Detect()

8 btn\_Start\_Game.hover\_Detect()

9 End While

1 Load Home Page:

* 1. load form
  2. load background
  3. load text and images
  4. load buttons

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.