

H446/03 Programming Project

Abby Stott

Centre Number: 37363
Candidate Number: 1856



Contents

Analysis	7
Initial Ideas	7
Inspiration for my Game	7
Computer Sustainability	8
Computational Methods	8
Problem Recognition	8
Problem Decomposition	8
Abstraction	8
Visualisation	8
Backtracking	9
Concurrency	9
The Original Game	9
History and Popularity	9
Planning Based on Research	9
First Version Plan	10
Limitations	11
Flowchart	11
Stakeholders	12
Questionnaire #1	13
Responses to Questionnaire #1	14
Evaluation of responses to Questionnaire #1	17
Questionnaire #2	18
Section 1	18
Section 2	20
Responses to questionnaire 2	21
Section 1	21
Section 2	25
Evaluation of responses to questionnaire 2	26
Section 1	26
Section 2	27
Hardware and Software Requirements	28
Success Criteria	29
Stakeholder Signatures	31
Design	32
Top Down Modular Design	32

Layer 1	33
Layer 2	33
Layer 3	33
Layer 4+	33
Top Down Modular Chart Modules	33
Weapon Menu	33
Main Menu	33
Leaderboard Menu	34
Movement	34
Black Box Test Tables	34
Post Development Testing	64
Algorithms	66
Controls	66
Player Jump	66
Game Escape	67
Player Fire Weapon	67
Enemy movement	67
Player and enemy removal	67
Auto Scrolling Screen	68
Weapon collision	68
Design Objectives	72
User Interface Design	72
Main Menu Screen	72
Play Game Menu	73
Leaderboards Menu	74
Aesthetic Considerations	74
Input Considerations	75
Processing Considerations	76
Output Considerations	76
UML Diagrams	76
Attributes	76
Methods	77
Class Diagrams	77
Validation plan	77
Data Dictionary	78
Stakeholder Signatures	81
Development	82
Early Planning and Practice	82
Version 1 - Creating the Screen	82

Tests	85
Top down Modular Update	85
Success Criteria	86
Version 2 - Scrolling Screen	89
Tests	90
Tests 2	93
Top Down Modular Update	95
Success Criteria	95
Version 3 - Jumping Player	97
Tests	99
Top Down Modular Update	100
Success Criteria	100
Version 4 - Enemies	103
Tests	104
Top Down Modular Update	105
Success Criteria	105
Version 5 - Health	108
Tests	111
Top Down Modular Update	112
Success Criteria	112
Version 6 - Respawning Enemies	115
Tests	115
Top Down Modular Update	116
Success Criteria	116
Version 7 - Bullets	119
Tests	120
Tests 2	122
Top Down Modular Update	123
Success Criteria	123
Version 8 - Random Enemy	126
Tests	127
Top Down Modular Update	128
Success Criteria	128
Version 9 - Different Weapons	130
Tests	133
Tests 2	135
Top Down Modular Update	136
Success Criteria	136
Version 10 - Score	139
Tests	140

Top Down Modular Update	142
Success Criteria	142
Version 11 - Menu	144
Tests	147
Tests 2	149
Top Down Modular Update	150
Success Criteria	150
Version 12 - Customisation Menu	153
Tests	155
Top Down Modular Update	157
Success Criteria	157
Version 13 - Fixing Health Bar	159
Tests	160
Top Down Modular Update	161
Success Criteria	161
Version 14 - Game Over and Validation	164
Tests	165
Top Down Modular Update	167
Success Criteria	167
Version 15 - Leaderboard	169
Tests	172
Tests 2	176
Top Down Modular Update	177
Success Criteria	177
Version 16 - Chosen Weapon	180
Tests	181
Top Down Modular Update	184
Success Criteria	184
Version 17 - Sound	186
Tests	190
Top Down Modular Update	192
Success Criteria	192
Version 18 - Difficulties	195
Tests	198
Top Down Modular Update	226
Success Criteria	226
Version 19 - Different Players	228
Tests	231
Top Down Modular Update	233
Success Criteria	233

Version 20 - Cleaning up Stakeholder Signatures	236
Evaluation	236
Post Development Test Table	236
Post Development Test Video	238
Evaluation of Post development Tests	238
Final Questionnaire	238
Responses to final questionnaire	240
Evaluation of responses to final questionnaire	251
Evaluation of Success Criteria	254
Auto Scrolling Screen	257
Player	257
Enemy	258
Difficulties	258
Menu	258
Power - Ups	258
Weapons	259
Score	259
Sound	259
Further Development	259
Maintainability	260
Appendix	261
Final Code	261

Analysis

Initial Ideas

For my programming project I have decided that I am going to code a game. Once I decided to create a game I had to think about the game I wanted to create and think about the time that I have to program it. When I first decided to code a game there were many game ideas that came to mind. One of the game ideas that came to mind was tetris however I was going to add my own feature that enables the user to change the block that is falling down into a block of their choice by spending points. However, this game idea seemed too complex and I didn't believe that I would be able to program this game within the time that I have to do it. I also decided that I want to create a game with a bit more originality that includes the supernatural theme because that is more interesting to me, so I have decided that I want to create a horizontal scrolling platformer game based on the CW's American TV series "Supernatural".

The game will be similar to dino chrome because the main character is in the centre of the screen and enemies come from the right hand side and there is a scrolling screen. However, in my game you can also choose to shoot monsters to kill them by selecting the correct weapon. If you kill the monster you get more points than you would if you jumped over it. Jumping over the monster will take some of the health off the innocent and when the innocents health gets to 0 you die. If you fail to jump over the monster or kill it you also die. If you choose the wrong weapon then you will have to jump over the monster because certain weapons kill certain monsters.

Inspiration for my Game

The main inspiration for my game is CW's "Supernatural". The objective of my game is to hunt monsters to save the life of the innocent, which is what happens in every episode of "Supernatural". However, to make it easier to code instead of walking around a map to find the monsters I have decided to make it a side scrolling game because this is also easier to play. The hunters in my game (the characters that the users are) are based off some of the hunters that are in "Supernatural" except instead of having images of them I am going to create pixelated images of them. I have also chosen the monsters in my game to be from "Supernatural" and the weapons that you use to kill them are the weapons that are used to kill them in the show because in the show only certain weapons can kill certain monsters. Having this concept will also make my game more challenging to play.

Computer Sustainability

Computational Methods

To plan and program my game there are many computational methods that I am going to have to use in the process. The computational methods that I have chosen to use are Problem recognition, Problem decomposition.

Problem Recognition

To plan my game, the first computational method I am using is problem recognition. Problem recognition is defining what the problem is. You have to be able to do this before you can actually solve the problem because it means that you have an understanding of what the problem is, therefore enabling you to figure out what you are going to do.

Problem Decomposition

After problem recognition the next computational method that I am going to use is problem decomposition. Problem decomposition is when you break down a larger problem into smaller subproblems that are much easier to handle. I am going to use this method because it means that I can work on tiny sections of the game at a time so therefore I can make sure each section works before moving on to the next making it easier to find errors within my code. For problem decomposition I am going to make a divide and conquer diagram.

Abstraction

Abstraction is when you remove unnecessary detail to focus on the important parts. I am going to use abstraction when I am programming my game because I will be using a lot of functions that come with python but I am not focusing on all of the code that is used to make them work, therefore I am removing the unnecessary detail. I am just using the functions, therefore I am focusing on the important parts. I am also using abstraction for the graphics that I create because instead of creating loads of different detailed clouds and detailed monsters, I am creating a cloud and just copying it and I am not adding too much detail to the clouds and every witch that I make will all look the same. Using abstraction in my game means that it won't take as long to program therefore I am more likely to get it programmed in the time.

Visualisation

Visualisation is when you make use of graphical representation of data in order to find solutions. I will use visualisation when I collect my questionnaire results from my stakeholders because it will be easier to analyse the results of the questionnaire if they are represented as bar charts and pie charts. I will also be using visualisation when I create flow charts and class diagrams before starting to program my game. This will make it easier to program because if you can visually see how the game is going to go then it will be easier to program it and also it will be easier to program the class if you can visually see what your class is going to be like.

Backtracking

Backtracking is when you “try” an approach to solve a problem and continue to do it until you don’t succeed. When you meet a dead end you will backtrack to the last known success and continue with a different approach. I may use backtracking when I am programming my game because if I want to add a feature in my game and I write code and keep making errors when trying to fix it I will go back to the last version of the code that was successful therefore I will backtrack to my last known success.

Concurrency

When two things are happening at the same time they are happening concurrently. I will be using this within my game because I am going to have a scrolling screen that will be happening at the same time as bullets are going to be shot at the monsters.

The Original Game

Jump Bug is the first side-scrolling platform game released and was developed by Alpha Denshi. The character goes from the left side of the screen to the right and to play the game you control a car that is driving along through different environments and you shoot all of the enemies that appear. To gain points in the game you have to collect treasure, kill the enemies and jump on the clouds.

There are multiple different objects in the game that the car can interact with. The treasure in the game will have collision detection so that if the car collides with the treasure then your points increase. There will also be collision detection to detect if the bottom of the car has collided with the top of the clouds and if this is true then your points will also increase. The final collision detection that is in the game will be with the enemies. The first one for the enemies will be if the bullet collides with the enemies then your points will increase and the enemy object will be killed. Also if the car object collides with the enemy then the car object will be killed.

The game jump bug had a pixelated design because it was run on a low resolution screen but this also made the game cheaper to run and also meant that the game didn’t use up as much storage space therefore also making the game faster.

History and Popularity

The first scrolling platformer game that was released to the public was Jump Bug. This game was released back in 1981. Since then people have made other scrolling platform games such as pac-land in 1984 which is just 3 years later. Scrolling games evolved into Super Mario Bros which was first released in 1985 and had sold over 40 million copies by 1999.

Planning Based on Research

Upon researching side scrolling platform games I found one that a user has created using python and pygame called “Jungle Run”. This game that I found at

<https://github.com/moseshsu01/Jungle-Run/blob/master/game.py> uses object oriented programming and imports pygame and random.

After looking at the code for this game I have also decided to use object oriented programming because it means that I can have an enemy class that has attributes and methods and then I can create many different enemy objects from that class which enables the program to have a smaller file size and it will reduce the time that it takes to program the game. Classes are templates for objects meaning that the enemy objects will get all of the methods and attributes from the enemy class. Encapsulation is when variables are kept hidden within a class so they are private. This means that they are stored more securely which therefore means my game is less likely to have an error because they cannot be accidentally modified.

I will be importing the pygame library so that I can create a 2d game and the pygame library will mean that I can actually have graphics in my game. I will also import the random library because this means that I can have a random enemy appear on the screen every time so then it varies therefore the user can't memorise the order in which the monsters will appear.

First Version Plan

The programming language I am going to use to code my game is python. I have chosen this language because it is a language that I am already familiar with because I have programmed with python in the past. It also has access to different libraries that I can use in my game like Pygame and random. Pygame is what I am going to use as my graphics driver.

For my game I am going to use object oriented programming instead of procedural programming because this enables me to create many objects from the same class because all of my enemies are going to have the same attributes. This means that my program will use less lines of code therefore meaning it won't have as large of a file size and it means that it won't take as long to code meaning that it will be easier to program it all within the time that I have.

Many different scrolling platformers have been created in the past and each one has unique features. My scrolling game is going to have clear scrolling platform features however I am also going to make my own because my main inspiration for my game is CW's "Supernatural".

I will ask my stakeholders about the features that are going to be in the game however, the features I am thinking about using from other scrolling platformers are:

- Pixelated design
- 2D game
- Screen that auto scrolls (including the background).
- Ability to make your character jump

- Ability to kill enemies

The features that I am thinking about adding to my game that isn't common in scrolling platformers are:

- Choice of weapons
- Choosing your character
- Certain weapons killing certain enemies
- Difficulty choice at the start:
 - Easy - If the user chooses this game mode then the game will start off slower at the start
 - Medium - If the user chooses this game mode then the game will be faster than easy but slower than hard
 - Hard - If the user chooses this game mode then the game will start off faster.

Limitations

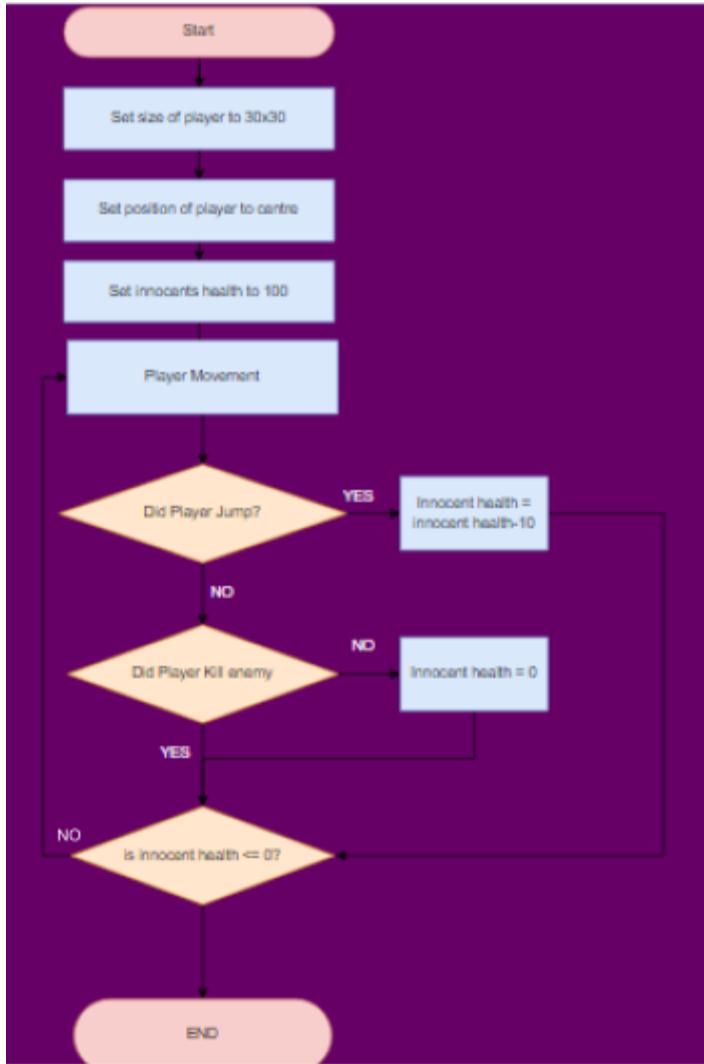
Whilst programming my game there are going to be a few limitations that I will face.

- In the past I have coded using python however I don't have much experience using the pygame add-on so it is going to be challenging to code this game because of the chosen features. In order to overcome this limitation I can go on sites like pygame documentation to help me out with anything I am struggling with.
- Another limitation I may have is time. There are many features that I want in my game however I may not have enough time to implement them all , meaning I will have an incomplete game. To overcome this limitation I am going to create the basic game then if I have more time I will add one more extra feature that I want in it so that I have a game that works at the end if I run out of time.
- Graphics is another limitation I have because I am wanting pixelated characters based on the characters from supernatural however I will struggle creating them because I am not a graphical artist so it is going to take some time to create them. To overcome this limitation I can look online for characters that people have already created because I am not actually distributing the game so copyright is not going to affect it.

Flowchart

To make my game easier to code I am going to create a flowchart of how I want the game to go. For this I am using abstraction and problem decomposition. The flowchart represents the basic game features that I am going to have.

The first three blue boxes shown set some variables that will need to be set. The fourth blue box sets the movement of the player. After that there are multiple decisions to see if health needs to be taken away and to see if the game needs to end.



Stakeholders

The target audience for my game is “Supernatural” fans aged 15 years and above. I agree with this estimate because it is based on CW’s “Supernatural” which is a 15 and the game requires some knowledge of “Supernatural” to be able to play it. The game also includes some violence because of the killing monsters therefore making it unsuitable for younger ages.

Because my age range is 15+ I am going to have a mature audience therefore when I collect feedback from my target audience I am likely to get honest responses. I am going to send my questionnaire to people that are in the age range that have watched “Supernatural”.

My market research that I am going to carry out won’t be as accurate as I would like it because I am limited to who I can ask and what sampling techniques I can use. If I was able to do my research properly I would have a sampling frame of people who have

watched supernatural and then I would collect a stratified sample of these people to make my research more accurate and also this reduces bias.

Throughout the production of the game I will have some of my stakeholders test the game and then gain informal feedback from them in order for my game to fit the needs of my target audience as much as possible.

Questionnaire #1

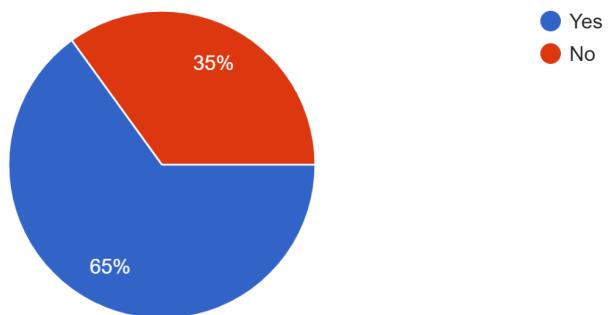
I have made this questionnaire so that I have an insight to what features my stakeholders like in games. This will help me add or remove features from my game idea to make it better suited to my stakeholders.

1. Do you like to play games based off shows? - Multiple Choice
2. In games do you prefer pixelated graphics or cartoon graphics? - Multiple Choice
3. Do you like power-ups in games? - Multiple Choice
4. Which difficulties would you like the option of? - Checkbox
 - a. Very Easy
 - b. Easy
 - c. Medium
 - d. Hard
 - e. Impossible
 - f. None
5. Do you like leaderboards in your game? - Multiple Choice
6. Do you like your score to be displayed on screen or get told at the end? - Multiple Choice
7. Do you like levels in games or endless mode? - Multiple Choice
8. Do you like games that get slightly harder the longer you play it? - Multiple Choice

Responses to Questionnaire #1

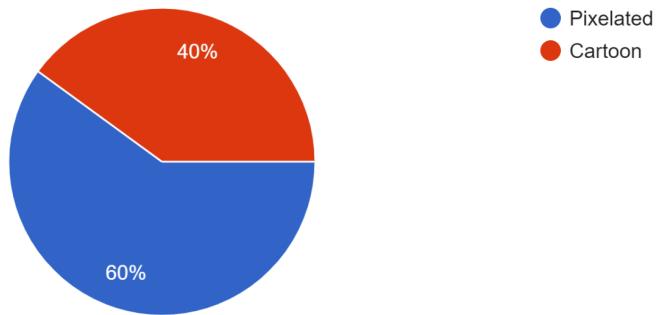
Do you like to play games based off shows?

20 responses



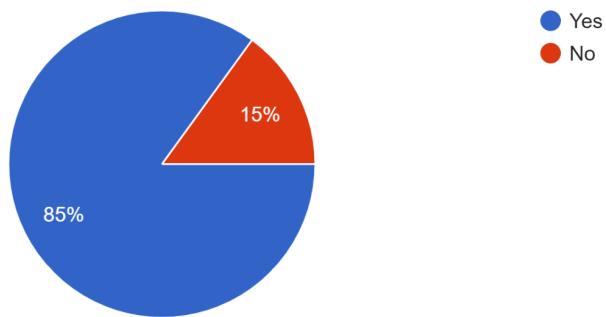
In games do you prefer pixelated graphics or cartoon graphics?

20 responses



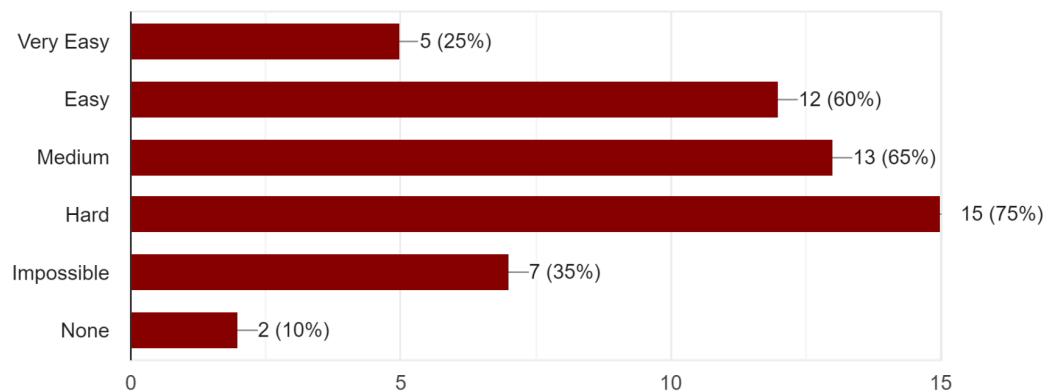
Do you like power-ups in games?

20 responses



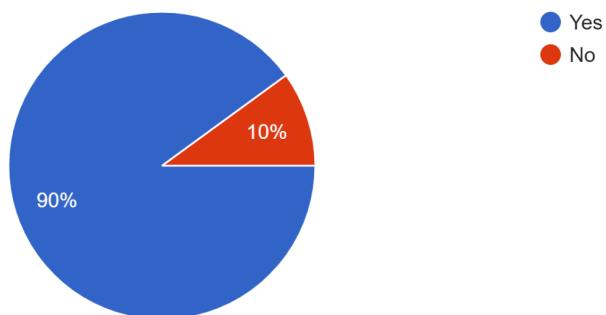
Which difficulties would you like the option of?

20 responses



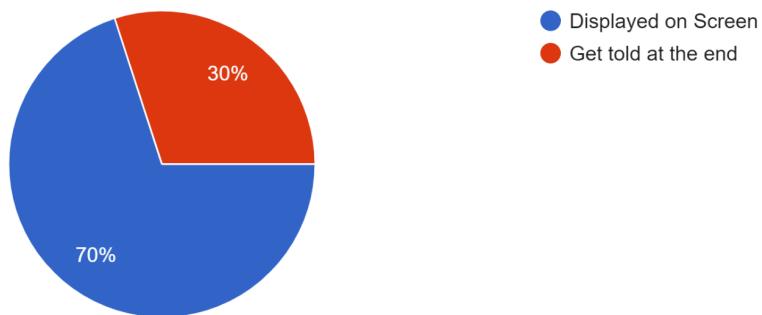
Do you like leaderboards in your game?

20 responses



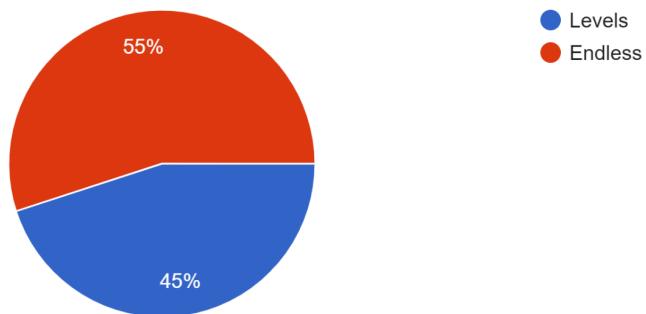
Do you like your score to be displayed on screen or get told at the end?

20 responses



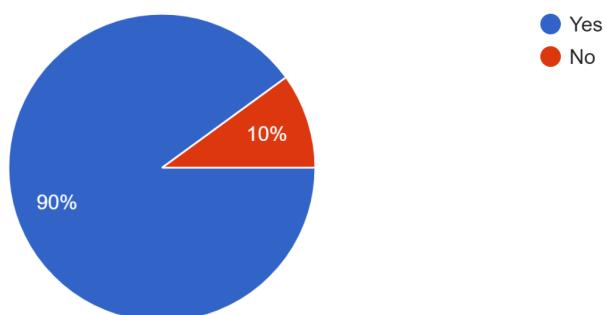
Do you like levels in games or endless mode?

20 responses



Do you like games that get slightly harder the longer you play it?

20 responses



Evaluation of responses to Questionnaire #1

For questionnaire 1 I asked all closed end questions because the results will be much easier to analyse because people can't type whatever they want meaning all answers have to be one of the options available. I also asked closed end questions because it means that my stakeholders are more likely to answer my questionnaire because it is quick and easy to complete meaning that I am more likely to get more responses making my results more reliable.

For this questionnaire I got 20 responses which should be enough data to see roughly what features my stakeholders like, especially because my questions are closed end questions therefore I don't have 20 different responses for 1 question.

Question 1 - 65% of the people that took my questionnaire said that they like to play games that are based on shows. This is reflective of my game idea because my whole game is based on the television series "Supernatural". This means that I can create my game idea and a lot of people are likely to play it. There are no follow up questions on this piece of data.

Question 2 - My research shows that 60 % of people like to play games with pixelated graphics. This means that I can use a pixel art site and make all of the graphics for my game in order to fit the needs of my target audience. Because I will be creating all of the graphics myself it means that I am able to make all of the characters myself. More data will be needed to see what characters my target audience want me to create.

Question 3 - With asking this question I have discovered that 85% of my stakeholders like to have power-ups in games that they play. This means that I am going to have to think of some upgrades that I could put into my game and also I am going to have to add a coin system in order for people to purchase the power-ups or I am going to have to add a time in so that they can only activate it after a certain amount of time. I am going to have to collect more data from this information with 2 different follow up questions. One of the follow up questions will be a choice of having a coin system or having to wait some time before activating the powerup, and the second follow up question will be asking which power ups they prefer out of a list of available options.

Question 4 - This data shows that the majority of my stakeholders like the choice of difficulty at the start of the game. This reflects my game idea because I was planning on having a difficulty choice at the start of my game. The results of the question also show that easy, medium and hard are the most popular choices to have so I am going to let the user choose between easy medium and hard at the start of my game. There are no follow up questions based on this piece of data.

Question 5 - The results from this question show that 90% of people like to have leaderboards in games that they play. This is reflective of what I was going to do in my game because I was going to already include a leaderboard in my game because it means that the people who play it can compare their score with others that play it and

also it is likely to increase the amount of people who play it because of friendly competition. There are no follow up questions based on this piece of data.

Question 6 - This shows that the stakeholders of my game like to see the score displayed on the screen whilst they play it. 30% like to be told the score at the end. In this scenario I can program my game to fit all of my stakeholders because I can display the score on the screen whilst the game is being played and I can also show the player their score at the end so that if they don't see their final score they can still find out what their final score is. There are no follow up questions based on this piece of data.

Question 7 - This data shows that 55% of my stakeholders like to play games in endless mode. This is representative of my initial ideas for this game. Programming my game in endless mode is going to mean that my game won't take as long to code therefore I am more likely to have my game coded within the time limit that I have to do it in. There are no follow up questions based on this piece of data.

Question 8 - The final question that I asked shows that most of my stakeholders like to play games that get slightly harder the longer they play it . This is reflective of my original plans for the game. I am going to have to ask a follow up question on this piece of data because I want to find out if my stakeholders would prefer for the game to get slightly faster or if they would prefer for monsters to come along more frequently.

Questionnaire #2

I created this questionnaire to get more detail about certain features in my game that the stakeholders like. This will help me to work out the finer details of my game so that my stakeholders are more likely to enjoy it.

Section 1

This section is about the characters and the design of the game.

1. What Supernatural hunters would you like to be included in the game? -

Checkbox

- a. Sam Winchester
- b. Dean Winchester
- c. Mary Winchester/ Mary Campbell
- d. John Winchester
- e. Jack Klein
- f. Claire Novak
- g. Castiel
- h. Bobby Singer
- i. Rowena Macleod
- j. Charlie Bradbury
- k. Jo Harvelle
- l. Ellen Harvelle
- m. Rufus Turner

- n. Samuel Campbell
 - o. Henry Winchester
 - p. Jody Mills
 - q. Other
2. If you chose other, which hunter would you like to be in the game? - Short Answer
3. Which monsters would you like in the game? - Checkbox
- a. Demon
 - b. Vampire
 - c. Werewolf
 - d. Witch
 - e. Spirits
 - f. Hellhounds
 - g. Leviathans
 - h. Angels
 - i. Death
 - j. Wendigo
 - k. Djinn
 - l. Shapeshifter
 - m. Skinwalker
 - n. Dragon
 - o. Fairies
 - p. Ghoul
 - q. Wraith
 - r. Banshees
 - s. Siren
 - t. Other
4. If you chose other, what monsters would you like included? - Short Answer
5. Which style of pixel art would you pick? - Multiple Choice
- a. Style 1 (There was an image with this choice)
 - b. Style 2 (There was an image with this choice)
6. Where would you like your score displayed while playing? - Multiple Choice Grid
- a.

	Left	Between Left and Middle	Middle	Between Middle and Right	Right
Top					
Between					

Top and Middle					
Middle					
Between Middle and Bottom					
Bottom					

7. Where would you like your power-ups to be displayed? - Multiple Choice Grid

a.

	Left	Between Left and Middle	Middle	Between Middle and Right	Right
Top					
Between Top and Middle					
Middle					
Between Middle and Bottom					
Bottom					

Section 2

This section is about the gameplay of the game.

1. How would you like to activate your power-ups in the game? - Multiple choice
 - a. Buy with coins
 - b. Activate after waiting time
2. Which Power up would you prefer? - Checkbox
 - a. Screen Slows Down
 - b. Enemies appear less frequently
 - c. Health Boost (You can Jump over enemies without losing any health)
 - d. Score Boost (Score increases faster)

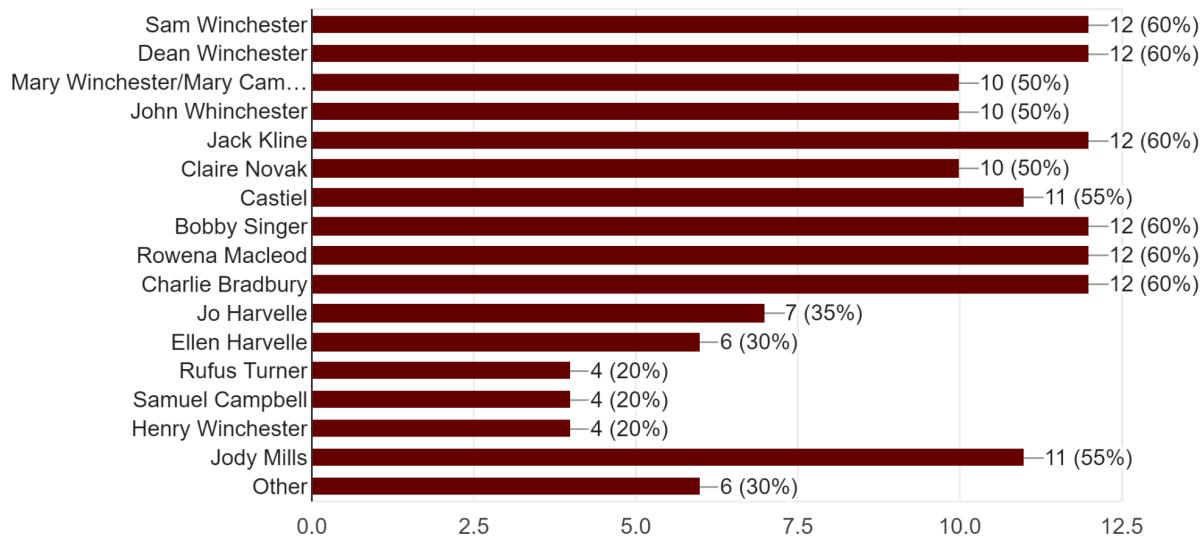
3. How would you like the game to get slightly harder? - Multiple Choice
- Gets slightly faster
 - Enemies appear more often

Responses to questionnaire 2

Section 1

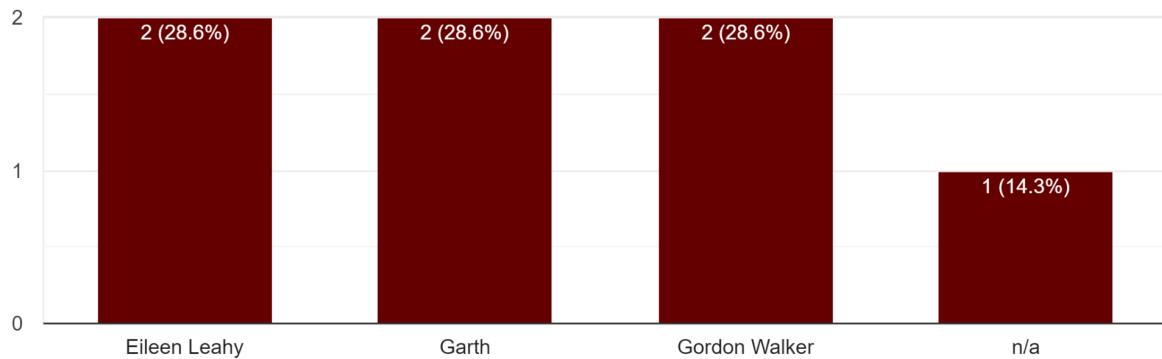
What Supernatural hunters would you like to be included in the game?

20 responses



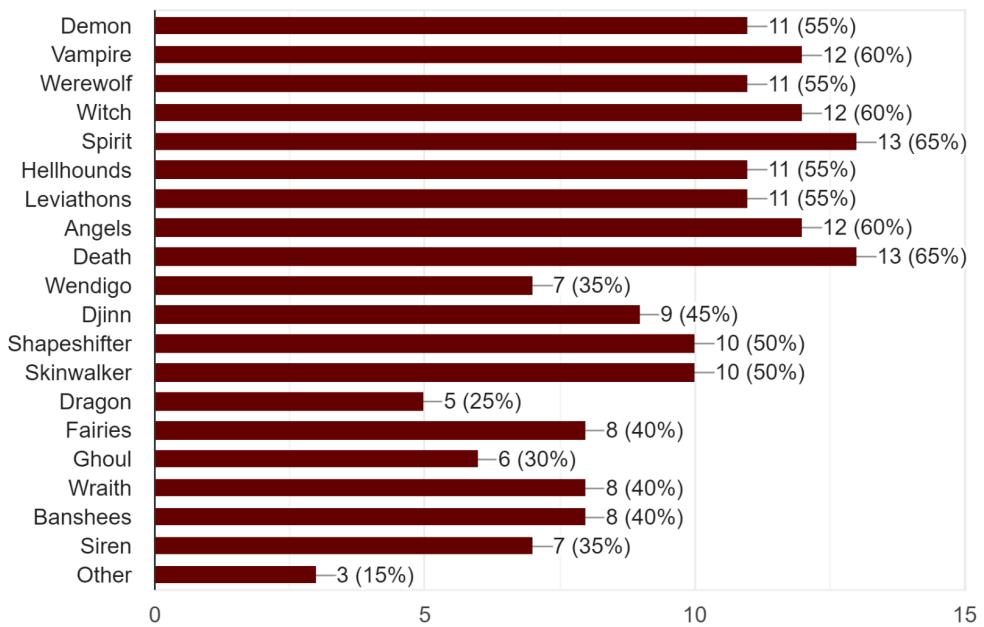
If you chose other, which hunter would you like to be in the game?

7 responses



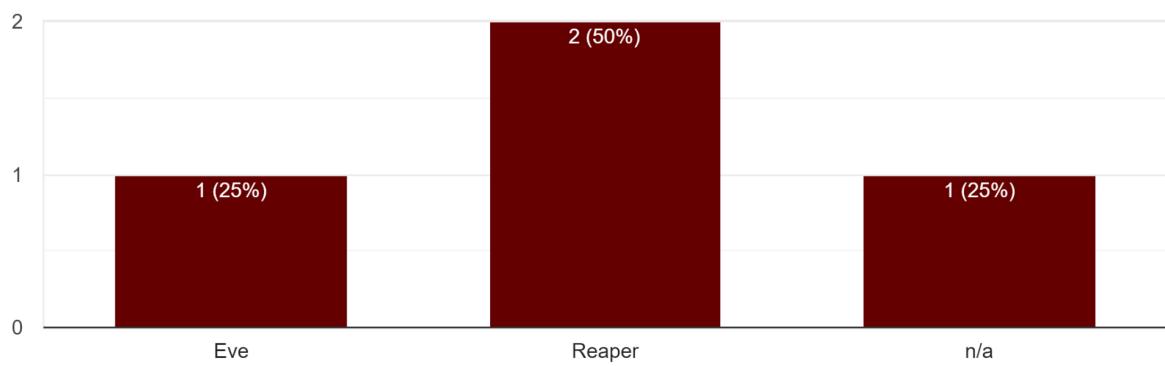
Which monsters would you like in the game?

20 responses



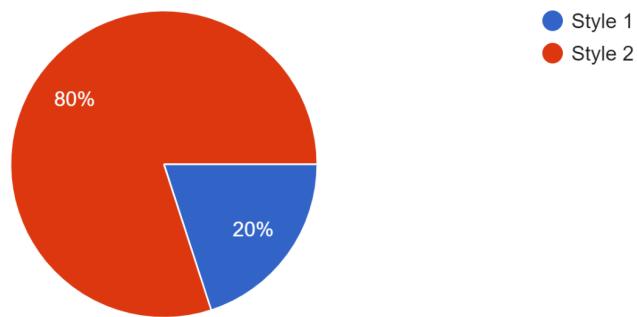
If you chose other, What other monsters would you like included?

4 responses

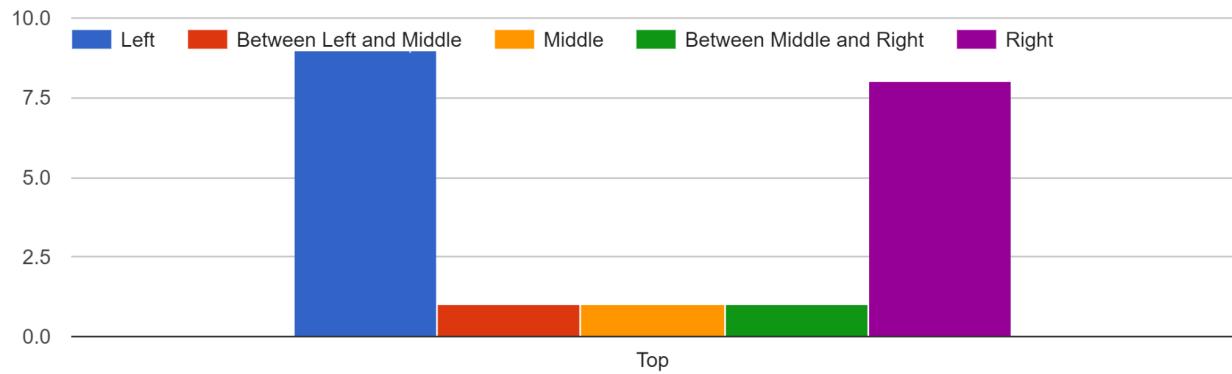


Which Style of pixel art would you pick?

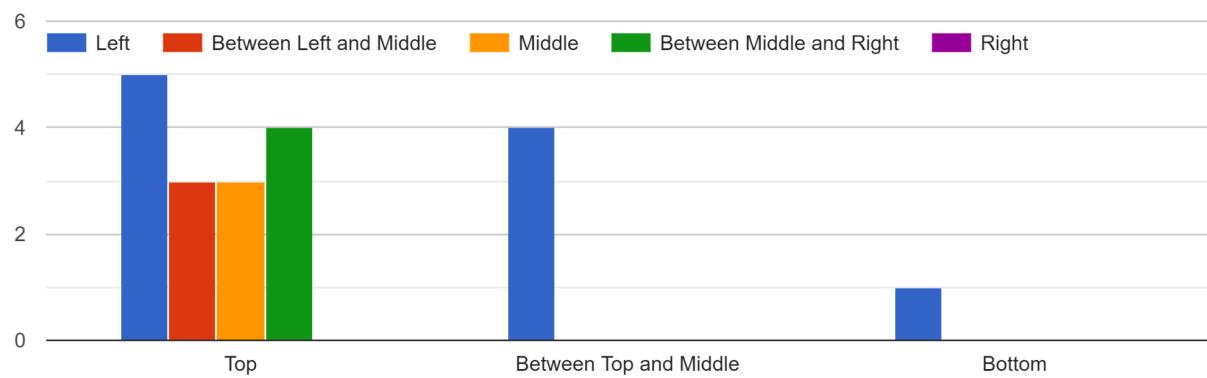
20 responses



Where would you like your score displayed whilst playing?



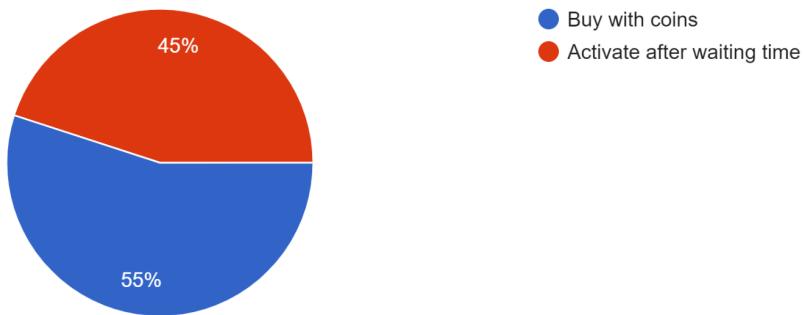
Where would you like your power-ups to be displayed?



Section 2

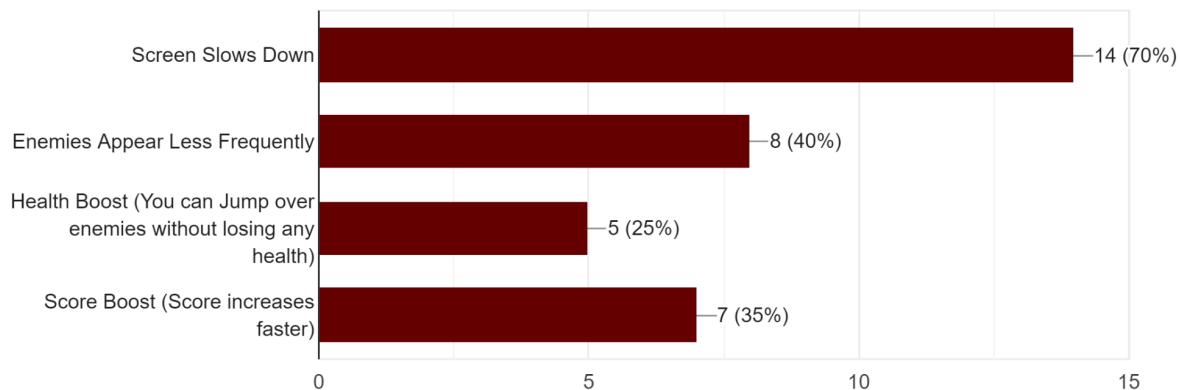
How would you like to activate your power-ups in the game?

20 responses



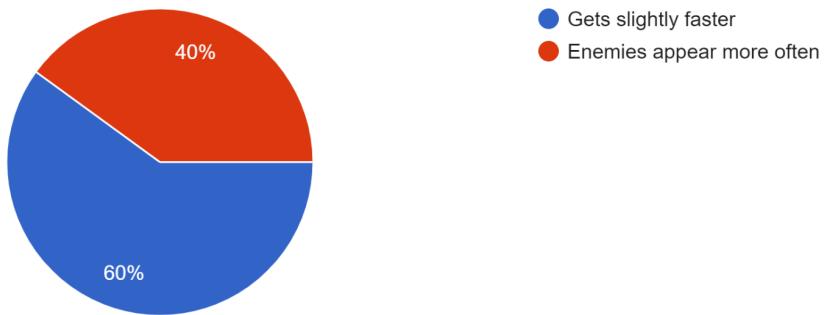
Which power up would you prefer?

20 responses



How would you like the game to get slightly harder?

20 responses



Evaluation of responses to questionnaire 2

Section 1

Question 1 - I asked this question because people might want to be different hunters to others so I decided to see which ones people wanted to be and I will create the most popular one. The results of this question shows that 60% of people chose: Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury. Due to these being the favourite of my stakeholders, I am going to let people who play my game decide which one of these they want to play so that they have a range of options. If I have enough time I will also add Castiel and Jody mills because they had the second highest number of people wanting to play as them with 55%.

Question 2 - I asked this question because people might want to play as hunters that I didn't include in the previous question so I gave them the chance to choose any hunter that is in supernatural. The results of this question show that a few people want other hunters but the other hunters aren't as popular as the ones in the previous question. This question doesn't change who I am going to let people have a choice of, due to the results.

Question 3 - I asked this question because I wanted to know which were the most popular monsters so then I know which ones to include in my game. The most popular monsters were spirits and death each with 65% of my stakeholders choosing them so these are definitely going to be in my game. There are only two monsters that are the most popular so I am going to only have these two monsters in the easy mode but as the difficulty mode increases I am going to include more monsters. For the medium mode that I am going to create, I will also include the second most popular monsters in my game so that there are more variations of monsters therefore making the game slightly harder. The results show that the second most popular monsters with 60% of my stakeholders choosing them are vampires, witches and angels. For the hard mode in

my game I am also going to include the third most popular monsters which 55% of my stakeholders chose. These are demons, werewolves, hellhounds and leviathans.

Question 4 - I decided to include this question because people might want monsters in the game that I didn't include in the previous question so I gave them the chance to recommend any other monsters. The results of this question show us that a few people wanted other monsters that I didn't include in the previous question, however they aren't as popular as the ones in the previous question therefore this question isn't going to change which monsters I include in my game.

Question 5 - I asked this question because I wanted to know which was the preferred pixel art style for my game because in the previous questionnaire people said that they preferred pixel art graphics. The results of my questionnaire show that style 2 was the most popular pixel art style with 80% of users choosing that style so I am going to make my pixel art as close to that style as possible so that my game is more appealing for my stakeholders to look at.

Question 6 - I asked this question because I wanted to know where people would prefer the score to be displayed whilst playing the game because I needed to know the location where people would be able to see it easier without them dying in the game. The results of the questionnaire show that users would prefer the score to be displayed on the top left with 9 of my stakeholders choosing that option. I am going to have the score on the top left because this is the preferred placement of the score for my stakeholders.

Question 7 - I asked this question because I wanted to know where the most convenient place for the power ups would be so that it is easier for the user to access power ups without the game ending for them. The most popular placement for the power ups in my game is the top left with 5 of my stakeholders choosing this option. The previous question shows that 9 people wanted the score in the top left corner so I am unable to have the powerups in the top left corner. This means that I will choose the second favourite option for the powerups with 4 votes which is either on the top between the middle right or between the top and middle on the left. I am going to display the powerups on the left between the middle and the top because this option is the closest to the top left corner so it is more likely to fit the needs of my stakeholders.

Section 2

Question 1 - I asked this question because in the previous questionnaire people said that they wanted power ups in the game , however I didn't know how they wanted to activate the power ups. The results of this question show that 55% of my stakeholders would prefer to activate the power ups by using coins. This means that I am going to add a coin object and class so that the user is able to collect coins and then I am going to add power ups that can only be used if the coins have been spent.

Question 2 - I asked this question because in the previous questionnaire people said they wanted power ups in the game however, I don't know which power ups they would

prefer to use. The results of this question show that people would prefer to use a power up that slows the screen down because 70% of my stakeholders chose this option. This means that I am going to have this as the powerup that my users are able to purchase with the coins.

Question 3 - I asked this question because in the previous questionnaire people said they want the game to get slightly harder as they play it but I need to know how they would prefer it to get harder. The results of this question show that people would prefer the screen to get faster for the game to get harder with 60% of my stakeholders choosing this option so when I create my game I am going to make the screen go slightly faster after a certain length of time.

Hardware and Software Requirements

The tables below show the minimum requirements that the PC needs in order for my game to be run. The requirements are similar or the same as the requirements to run python because that is the programming language that my game is going to be coded in.

Software Requirements	Justification
Python 3.4.1	This is the software required to run my game.
GUIZero	This is the software required to run my game menu.
Pygame	This is the software required to use the pygame module.
OS: Windows 7 or higher Mac OS X 10.11 or higher, 64-bit Linux: RHEL 6/7, 64-bit	This is the minimum operating system requirements to run python

Hardware Requirements	Justification
Keyboard	This is a piece of hardware used to make the character jump and is used to input your name.
Mouse	This piece of hardware is used to click on the buttons to start the game.
Monitor	This is used so that you can see the game.

Processor that is 486DX . 66 MHz or higher	This is the minimum processing power required for python.
32 MB RAM	This is the minimum requirements in order to run python
50MB hard drive space	This is so that you have enough storage to run the game.

Success Criteria

Following the responses from both questionnaires I have created a list of success criteria that fits the needs of the majority of my stakeholders.

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.
 - A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death

- Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power - ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision
 - Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision

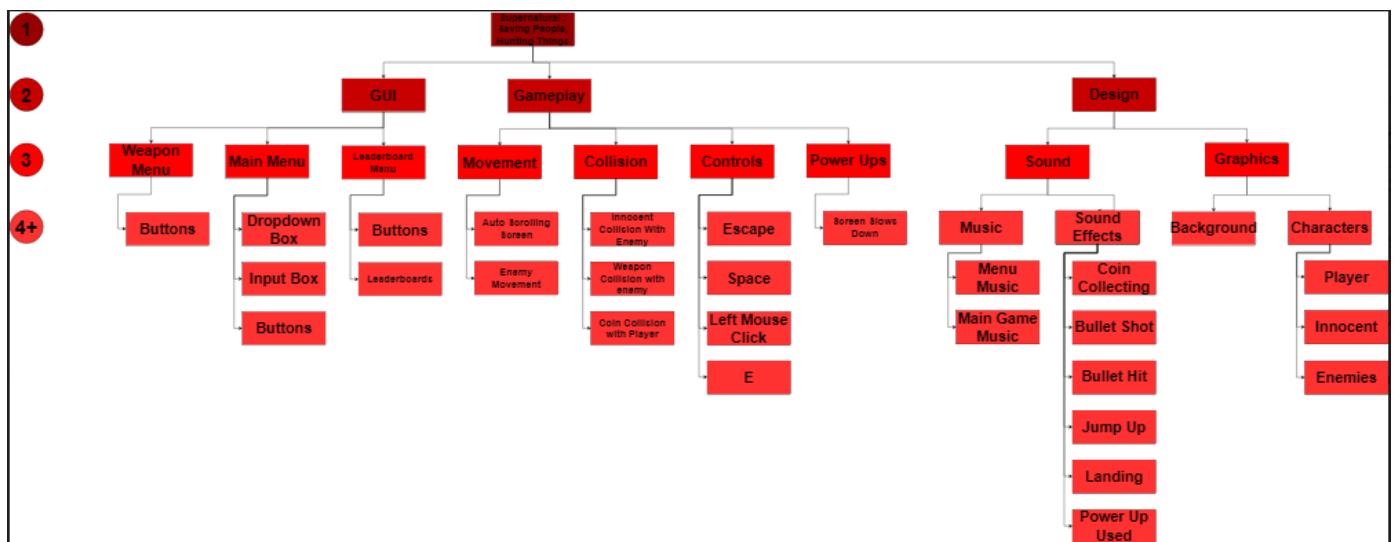
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
 - Silver Bullet
 - Kills Werewolves upon collision
 - Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Bone
 - Kills Leviathans upon collision
 - Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Stakeholder Signatures



Design

Top Down Modular Design



I created this top down modular design using problem decomposition. I looked at my main problem and broke it down into much smaller problems that are easier to solve. Creating this design is beneficial for the creation of my game because I can cross one of the smaller problems off at a time until eventually they are all done making it easier to focus on one part at a time. The only way to complete section 1 is to complete all of the section 4+ tasks meaning that the game is complete when section 1 is complete. I can use this top down modular chart as a checklist of what I have to complete in order for the game to be finished.

Layer 1

I created this layer as a starting point of the bigger picture. This layer isn't going to help me create the game but it is a starting point for the graph and also helps me to see my main goal.

Layer 2

This layer contains the three main categories that all the other modules can be grouped inside. This layer isn't very helpful but does allow me to split the different modules into smaller categories.

Layer 3

This layer splits down layer 2 into more specific categories that are slightly more helpful but still slightly vague.

Layer 4+

This layer splits down layer 3 into more specific tasks which can now be easily completed. I can use this layer as a checklist of the things that need to be completed to make the basic necessities of my game.

Top Down Modular Chart Modules

Weapon Menu

The weapon menu is a crucial part of the game for the user to be able to play it because they need to choose the correct weapon to kill the monster. The layout of the weapon menu will be basic with a title saying weapons at the top with all the choice of weapons below. Coming off the weapon menu module is another module that says buttons.

- Buttons - I have this smaller module coming off the larger weapon menu module because the buttons are what the user will press to select the weapon of their choice. I will have to have a weapon variable and when the user selects a button the name of the weapon will be what is stored in the weapon variable.

Main Menu

The main menu is important for the game and sets the image for the rest of the game as it is the first thing that the user sees when they go to play it.

The main menu is going to have a basic design because you don't want it to be too overwhelming because it needs to be easy to use. Coming from the main menu module are three smaller modules: Dropdown Box, Input Box and Buttons.

- Dropdown Box - This smaller module is included in the modular chart because I will need to include a dropdown box with all the names of the characters that the player has the option of playing as. This feature is important because it gives the user more choice and allows them to play as their favourite supernatural hunters.
- Input Box - This smaller module is included in the modular chart because the user needs to be able to input their name so that they can see their score on the leaderboard. This allows them to compete with others to be the best therefore it is likely to keep users more intrigued.

- Buttons - This smaller module is included in the module chart because the user needs the option to select whether they want to see the leaderboards, quit the game or play the game. This is essential for the user to be able to play the game.

Leaderboard Menu

The leaderboard menu is included in the chart because users want to see their score to see if they are the best player. This is also going to be a basic design because all that needs to be included is the choice of which leaderboard they want to see and the option to go back to the main menu. Coming off this module are two smaller modules: Buttons and Leaderboards.

- Buttons - This smaller module is included in the module chart because it is important to give the user the option to choose which leaderboard that they want to view. Clicking one of these buttons will close the leaderboard menu window and open another window with the required leaderboard on it, or it will close the window and open the app if you click on the back button.
- Leaderboards - This smaller module is included because the buttons will lead to the leaderboards. This smaller module is going to take longer to program than the others because there is going to be a different leaderboard for each difficulty and I will need to get the players scores onto a file in order for the leaderboard to be created.

Movement

Movement is crucial for most games so I included it in my layer three and then for layer four I split it into the different things that move within my game. The game wouldn't be playable without things that move. Coming from this larger module are two smaller modules: Auto Scrolling Screen and Enemy movement.

- Auto Scrolling Screen - The auto scrolling screen will give the illusion that the player is walking along a path and the faster the screen scrolls means that the player is walking faster. This is simple to code because it is only a few lines of code that will just loop.
- Enemy Movement - The enemies in my game will move from the right side of the screen to the left side of the screen. This will also be simple to program because it will have a movement function within the enemy class that will get called within the game loop.

Black Box Test Tables

Function	Test Data	Type of Data	Expected Outcome	Justification
			Sound	

Menu Music	Being on the menu	Valid	The music for the menu should be playing	These tests are important to carry out because you want the different music to play at the correct time to ensure that gameplay is smooth.
Game Music	Loading the Menu	Borderline	The music for the menu should be playing	
	Playing the Game	Valid	The music for the game should be playing	
	Loading the game	Borderline	The music for the game should be playing	
Collision with Coins Audio	Player coordinates == coin coordinates	Valid	The audio for the collection of coins should play	This is a test that needs to be carried out because the sound will confirm to the player that they have collected coins
	Players coordinates != coin coordinates	Invalid	The audio for the coin collection should not play	This test needs to be carried out because you don't want the coin collection audio to play if they haven't collected coins because this can cause confusion to the player
	Enemies coordinates == coin coordinates	Invalid	The audio for the coin collection should not play	This test needs to be carried out because you don't want the coin collecting audio to play when the enemy collides with the coins because

				this may confuse the player into thinking that the enemy collected the coins
Bullet Shot	Bullet is created and drawn onto the screen	Valid	Bullet shot audio is played	This test is important to carry out because having the bullet shot audio play when the bullet is shot confirms to the user that the bullet was shot
Bullet Hit	Bullet coordinate s == Enemy coordinate s	Valid	Bullet Hit audio is played	This test is important to carry out because having this sound play at the correct time confirms to the user that they just hit the enemy with their bullet
	Bullet coordinate s != Enemy coordinate s	Invalid	Bullet Hit audio is not played	This test is important to carry out because if the audio plays when the bullet hasn't collided with the enemy then the player is going to get confused.
Jump Audio	Space Bar is pressed	Valid	Jump audio is played	This test is important to carry out because having this sound play when the space bar has been pressed confirms to the user that they have just jumped
	Space Bar isn't pressed	Invalid	Jump audio is not played	This test is important to carry out because if this sound plays when the player hasn't jumped they

				will get confused
Landing Audio	Player coordinate s == original position but space hasn't been pressed	Invalid	Landing audio is not played	This test is important to carry out because you don't want the landing audio to play continuously throughout the game only once it has landed after jumping
	Player coordinate s == original coordinates just after space being pressed	Valid	Landing audio is played	This is important to carry out because you want the audio to play once the player has landed to confirm to the player that they have just landed
	Player coordinate s != original position	Invalid	Landing audio is not played	This is important to carry out because you don't want the landing audio to play whilst they are mid air because this can cause confusion
Power up used Audio	Mouse coordinate s == powerup button coordinates and mouse has been left clicked	Valid	Power up audio is played	This test is important because having this sound play when the power up button is clicked verifies to the user that the power up has been activated
	Mouse coordinate s == powerup button	Invalid	Power up audio is no played	This test is important because you don't want the power up audio to play when the right button has

	coordinates and mouse has been right clicked			been clicked because the user may be confused thinking that they activated the power up.
	Mouse coordinates == powerup button coordinates and mouse hasn't clicked	Invalid	Power up audio is not played	This test is important because the audio shouldn't be played when the mouse is hovering over the button because this isn't how the power up is going to be activated
	Mouse coordinates != power up button coordinates	Invalid	Power up audio is not played	This is important to carry out because you don't want the audio to play all the time. You only want it to play the second that the power up is activated
	Mouse coordinates != power up button coordinates and left button is clicked	Invalid	Power up audio is not played	This is important because the powerup isn't going to be activated when you click anywhere on the screen so if you click somewhere else on the screen and the power up audio is played then the player will get confused
Power up used	Mouse coordinates ==	Valid	The screen should slow down	This test is important because the power up chosen is for the

	powerup button coordinate s and mouse has been left clicked			screen to slow down and if it doesn't slow down it means that the power up doesn't work.
	Mouse coordinate s == powerup button coordinate s and mouse has been right clicked	Invalid	The screen should stay the same speed	This test is important because the screen isn't meant to slow down when it has been right clicked so if it does it means that there is a bug within my game.
	Mouse coordinate s == powerup button coordinate s and mouse hasn't clicked	Invalid	The screen should stay the same speed	This test is important because the screen isn't meant to slow down when the mouse hasn't been clicked so if it does it means that there is a bug within my game.
	Mouse coordinate s != power up button coordinate s	Invalid	The screen should stay the same speed	This test is important because the screen isn't meant to slow down when the mouse isn't on the power up button so if it does it means that there is a bug within my game.
	Mouse coordinate s != power up button coordinate	Invalid	The screen should stay the same speed	This test is important because the screen isn't meant to slow down when the mouse isn't on the

	s and left button is clicked			power up button but the left button has been clicked so if it does it means that there is a bug within my game.
Buttons				
Exit Game	Escape key pressed	Valid	The game should exit	This test is important because the game is played in full screen so there is no cross for the user to click therefore they need some other way to exit the game.
Character Jump	Space Key pressed	Valid	The player should jump	This test is important because my character has the ability to jump and because games usually use the space button to jump, this button is the one in my game used to jump because it makes the game simpler to play.
Weapon Menu	E Key pressed	Valid	Weapon Menu pops up	This test is important because the player has to choose the correct weapon and usually the button e is the inventory button so I chose this button as the menu weapon button and it needs to work otherwise they are stuck with the same weapon

				throughout meaning they won't be able to get a score as high.
Collisions				
Player Collision with enemy	Player coordinate s + 30 >= enemy coordinate s	Valid	Player Health decreases to 0 and the player object is deleted off the screen.	This test is important because if the player collides it means they have died but if the player continues to move then the player hasn't died and it will be impossible to end the game. It adds 30 because you want the collision to happen with the right of the player and not the left.
	Player coordinate s +30 <= enemy coordinate s	Invalid	Doesn't have any effect on the game , the game carries on as normal.	This test is important because if the player doesn't collide with the enemy you don't want it to affect the rest of the game.
Scythe Collision with enemy	Scythe Coordinate s == Death Coordinate s	Valid	Death health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the scythe kills death so when it collides you want death to disappear to give the illusion he's dead and you want the bullet to disappear because it has hit death.
	Scythe Coordinate s == Spirits coordinate s	Invalid	The weapon disappears off the screen.	This test is important because the scythe doesn't do any damage to the spirit however, it has still struck the spirit so it

			needs to disappear.
Scythe Coordinate s == Vampire coordinate s	Invalid	The weapon disappears off the screen.	This test is important because the scythe doesn't do any damage to the vampire however, it has still struck the vampire so it needs to disappear.
Scythe Coordinate s == Witch coordinate s	Invalid	The weapon disappears off the screen.	This test is important because the scythe doesn't do any damage to the witch however, it has still struck the witch so it needs to disappear.
Scythe Coordinate s == Angel coordinate s	Invalid	The weapon disappears off the screen.	This test is important because the scythe doesn't do any damage to the angel however, it has still struck the angel so it needs to disappear.
Scythe Coordinate s == Demon coordinate s	Invalid	The weapon disappears off the screen.	This test is important because the scythe doesn't do any damage to the demon however, it has still struck the demon so it needs to disappear.
Scythe Coordinate s == Werewolf coordinate s	Invalid	The weapon disappears off the screen.	This test is important because the scythe doesn't do any damage to the werewolf however, it has still struck the werewolf so it needs to disappear.
Scythe	Invalid	The weapon	This test is important

	Coordinate s == Hellhound coordinate s		disappears off the screen.	because the scythe doesn't do any damage to the hellhound however, it has still struck the hellhound so it needs to disappear.
	Scythe Coordinate s == Leviathan coordinate s	Invalid	The weapon disappears off the screen.	This test is important because the scythe doesn't do any damage to the leviathan however, it has still struck the leviathan so it needs to disappear.
Salt Collision with enemy	Salt Coordinate s == Death Coordinate s	Invalid	The weapon disappears off the screen.	This test is important because the salt doesn't do any damage to death however, it has still struck death so it needs to disappear.
	Salt Coordinate s == Spirit Coordinate s	Valid	Spirit health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the salt kills spirits so when it collides you want the spirit to disappear to give the illusion it's dead and you want the bullet to disappear because it has hit the spirit.
	Salt Coordinate s == Vampire Coordinate s	Invalid	The weapon disappears off the screen.	This test is important because the salt doesn't do any damage to the vampire however, it has still struck the vampire so it needs to disappear.

	Salt Coordinate s == Witch Coordinate s	Invalid	The weapon disappears off the screen.	This test is important because the salt doesn't do any damage to the witch however, it has still struck the witch so it needs to disappear.
	Salt Coordinate s == Angel Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the salt doesn't do any damage to the angel however, it has still struck the angel so it needs to disappear.
	Salt Coordinate s == Demon Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the salt doesn't do any damage to the demon however, it has still struck the demon so it needs to disappear.
	Salt Coordinate s == Werewolf Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the salt doesn't do any damage to the werewolf however, it has still struck the werewolf so it needs to disappear.
	Salt Coordinate s == Hellhound Coordinate s Hellhound health>10	Valid	The weapon disappears off the screen and the hellhounds health decreases by 10	This test is important because the salt harms the hellhound but can never kill it so if the health of the hellhound is above ten then 10 health can get taken off the hellhounds health, it has still struck the hellhound

				so the weapon needs to disappear.
	Salt Coordinate s == Hellhound Coordinates Hellhound Health<=10	Invalid	The weapon disappears off the screen.	This test is important because the salt can't kill a hellhound however it has still struck the hellhound so it needs to disappear
	Salt Coordinates == Leviathan Coordinates	Invalid	The weapon disappears off the screen	This test is important because the salt doesn't do any damage to the Leviathan however, it has still struck the Leviathan so it needs to disappear.
Machete Collision with enemy	Machete Coordinates == Death Coordinates	Invalid	The weapon disappears off the screen.	This test is important because the machete doesn't do any damage to death however, it has still struck death so it needs to disappear.
	Machete Coordinates == Spirit Coordinates	Invalid	The weapon disappears off the screen.	This test is important because the machete doesn't do any damage to the spirit however, it has still struck the spirit so it needs to disappear.
	Machete Coordinates == Vampire Coordinates	Valid	Vampire health decreases to 0 and gets removed off the screen. The weapon also	This test is important because the machete kills vampires so when it collides you want

	s		disappears off the screen.	the vampire to disappear to give the illusion it's dead and you want the bullet to disappear because it has hit the vampire.
	Machete Coordinate s == Witch Coordinate s	Invalid	The weapon disappears off the screen.	This test is important because the machete doesn't do any damage to the witch however, it has still struck the witch so it needs to disappear.
	Machete Coordinate s == Angel Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the machete doesn't do any damage to the angel however, it has still struck the angel so it needs to disappear.
	Machete Coordinate s == Demon Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the machete doesn't do any damage to the demon however, it has still struck the demon so it needs to disappear.
	Machete Coordinate s == Werewolf Coordinate s	Valid	Werewolf health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the machete kills werewolves so when it collides you want the werewolf to disappear to give the illusion it's dead and you want the bullet to disappear

				because it has hit the werewolf.
	Machete Coordinates == Hellhound Coordinates	Invalid	The weapon disappears off the screen	This test is important because the machete doesn't do any damage to the hellhound however, it has still struck the hellhound so it needs to disappear.
	Machete Coordinates == Leviathan Coordinates	Invalid	The weapon disappears off the screen	This test is important because the machete doesn't do any damage to the Leviathan however, it has still struck the leviathan so it needs to disappear.
WitchBullet Collision with Enemy	Witch Bullet coordinates == Death Coordinates	Invalid	The weapon disappears off the screen	This test is important because the Witch Bullet doesn't do any damage to death however, it has still struck death so it needs to disappear.
	Witch Bullet Coordinates == Spirit Coordinates	Invalid	The weapon disappears off the screen	This test is important because the Witch Bullet doesn't do any damage to the spirit however, it has still struck the spirit so it needs to disappear.
	Witch Bullet Coordinates == Vampire Coordinates	Invalid	The weapon disappears off the screen	This test is important because the Witch Bullet doesn't do any damage to the vampire however, it has still struck the

	s			vampire so it needs to disappear.
	Witch Bullet Coordinate s == Witch Coordinates	Valid	Witch health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the witch bullet kills witches so when it collides you want the witch to disappear to give the illusion it's dead and you want the bullet to disappear because it has hit the witch.
	Witch Bullet Coordinates == Angel Coordinates	Invalid	The weapon disappears off the screen	This test is important because the Witch Bullet doesn't do any damage to the angel however, it has still struck the angel so it needs to disappear.
	Witch Bullet Coordinates == Demon Coordinates	Invalid	The weapon disappears off the screen	This test is important because the Witch Bullet doesn't do any damage to the demon however, it has still struck the demon so it needs to disappear.
	Witch Bullet Coordinates == Werewolf Coordinates	Invalid	The weapon disappears off the screen	This test is important because the Witch Bullet doesn't do any damage to the werewolf however, it has still struck the werewolf so it needs to disappear.
	Witch Bullet Coordinates ==	Invalid	The weapon disappears off the screen	This test is important because the Witch Bullet doesn't do any damage to the

	Hellhound Coordinates			hellhound however, it has still struck the hellhound so it needs to disappear.
	Witch Bullet Coordinates == Leviathan Coordinates	Invalid	The weapon disappears off the screen	This test is important because the Witch Bullet doesn't do any damage to the leviathan however, it has still struck the leviathan so it needs to disappear.
Angel blade Collision with enemy	Angel blade Coordinates == Death Coordinates	Invalid	The weapon disappears off the screen	This test is important because the angel blade doesn't do any damage to death however, it has still struck death so it needs to disappear.
	Angel blade Coordinates == Spirits coordinates	Invalid	The weapon disappears off the screen	This test is important because the angel blade doesn't do any damage to the spirit however, it has still struck the spirit so it needs to disappear.
	Angel blade Coordinates == Vampire coordinates	Invalid	The weapon disappears off the screen	This test is important because the angel blade doesn't do any damage to the vampire however, it has still struck the vampire so it needs to disappear.
	Angel blade Coordinates == Witch coordinates	Invalid	The weapon disappears off the screen	This test is important because the angel blade doesn't do any damage to the witch however, it has still struck the witch so it needs to disappear.

	Angel blade Coordinate s == Angel coordinate s	Valid	angel health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the angel blade kills angels so when it collides you want the angel to disappear to give the illusion it's dead and you want the angel blade to disappear because it has hit the angel.
	Angel blade Coordinate s == Demon coordinate s	Valid	Demon health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the angel blade kills demons so when it collides you want the demon to disappear to give the illusion it's dead and you want the angel blade to disappear because it has hit the demon.
	Angel blade Coordinate s == Werewolf coordinate s	Valid	Werewolf health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the angel blade kills werewolves so when it collides you want the werewolf to disappear to give the illusion it's dead and you want the angel blade to disappear because it has hit the werewolf.
	Angel blade Coordinate s == Hellhound coordinate s	Valid	Hellhound health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the angel blade kills hellhounds so when it collides you want the hellhound to disappear to give the illusion it's dead

				and you want the angel blade to disappear because it has hit the hellhound.
	Angel blade Coordinates == Leviathan coordinates	Invalid	The weapon disappears off the screen	This test is important because the angel blade doesn't do any damage to the leviathan, however it has still struck the leviathan so it needs to disappear.
Demon blade Collision with enemy	Demon blade Coordinates == Death Coordinates	Invalid	The weapon disappears off the screen	This test is important because the demon blade doesn't do any damage to death however, it has still struck death so it needs to disappear.
	Demon blade Coordinates == Spirit Coordinates	Invalid	The weapon disappears off the screen	This test is important because the demon blade doesn't do any damage to the spirit, however it has still struck the spirit so it needs to disappear.
	Demon blade Coordinates == Vampire Coordinates	Invalid	The weapon disappears off the screen	This test is important because the demon blade doesn't do any damage to the vampire however, it has still struck the vampire so it needs to disappear.
	Demon blade Coordinates == Witch Coordinates	Invalid	The weapon disappears off the screen	This test is important because the demon blade doesn't do any damage to the witch, however it has still struck the witch so it

			needs to disappear.
Demon blade Coordinate s == Angel Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the demon blade doesn't do any damage to the angel however, it has still struck the angel so it needs to disappear.
Demon blade Coordinate s == Demon Coordinate s	Valid	Demon health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the demon blade kills demons so when it collides you want the demon to disappear to give the illusion it's dead and you want the demon blade to disappear because it has hit the demon.
Demon blade Coordinate s == Werewolf Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the demon blade doesn't do any damage to the werewolf, however it has still struck the werewolf so it needs to disappear.
Demon blade Coordinate s == Hellhound Coordinate s	Valid	Hellhound health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the demon blade kills hellhounds so when it collides you want the hellhound to disappear to give the illusion it's dead and you want the demon blade to disappear because it has hit the hellhound.
Demon	Invalid	The weapon	This test is important

	blade Coordinate s == Leviathan Coordinate s		disappears off the screen	because the demon blade doesn't do any damage to the leviathan however, it has still struck the leviathan so it needs to disappear.
Silver Bullet Collision with enemy	Silver Bullet Coordinate s == Death Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the silver bullet doesn't do any damage to death, however it has still struck death so it needs to disappear.
	Silver Bullet Coordinate s == Spirit Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the silver bullet doesn't do any damage to the spirit however, it has still struck the spirit so it needs to disappear.
	Silver Bullet Coordinate s == Vampire Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the silver bullet doesn't do any damage to the vampire however, it has still struck the vampire so it needs to disappear.
	Silver Bullet Coordinate s == Witch Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the silver bullet doesn't do any damage to the witch, however it has still struck the witch so it needs to disappear.
	Silver Bullet Coordinate s == Angel Coordinate	Invalid	The weapon disappears off the screen	This test is important because the silver bullet doesn't do any damage to the angel, however it

	s			has still struck the angel so it needs to disappear.
	Silver Bullet Coordinate s == Demon Coordinates	Invalid	The weapon disappears off the screen	This test is important because the silver bullet doesn't do any damage to the demon, however it has still struck the demon so it needs to disappear.
	Silver Bullet Coordinate s == Werewolf Coordinates	Valid	Werewolf health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the silver bullet kills werewolves so when it collides you want the werewolf to disappear to give the illusion it's dead and you want the silver bullet to disappear because it has hit the werewolf.
	Silver Bullet Coordinate s == Hellhound Coordinates	Invalid	The weapon disappears off the screen	This test is important because the silver bullet doesn't do any damage to the hellhound however, it has still struck the hellhound so it needs to disappear.
	Silver Bullet Coordinate s == Leviathan Coordinates	Invalid	The weapon disappears off the screen	This test is important because the silver bullet doesn't do any damage to the leviathan however, it has still struck the leviathan so it needs to disappear.
Colt Collision	Colt coordinate	Invalid	The weapon disappears off the	This test is important because the colt

with Enemy	s == Death Coordinate s		screen	doesn't do any damage to death, however it has still struck death so it needs to disappear.
	Colt Coordinate s == Spirit Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the colt doesn't do any damage to the spirit, however it has still struck the spirit so it needs to disappear.
	Colt Coordinate s == Vampire Coordinate s	Valid	Vampire health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the colt kills vampires so when it collides you want the vampire to disappear to give the illusion it's dead and you want the colt to disappear because it has hit the vampire.
	Colt Coordinate s == Witch Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the colt doesn't do any damage to the witch, however it has still struck the witch so it needs to disappear.
	Colt Coordinate s == Angel Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the colt doesn't do any damage to the angel, however it has still struck the angel so it needs to disappear.
	Colt Coordinate s ==	Valid	Demon health decreases to 0 and gets removed off	This test is important because the colt kills demons so when it

	Demon Coordinates		the screen. The weapon also disappears off the screen.	collides you want the demon to disappear to give the illusion it's dead and you want the colt to disappear because it has hit the demon.
	Colt Coordinates == Werewolf Coordinates	Valid	Werewolf health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the colt kills werewolves so when it collides you want the werewolf to disappear to give the illusion it's dead and you want the colt to disappear because it has hit the werewolf.
	Colt Coordinates == Hellhound Coordinates	Valid	Hellhound health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the colt kills hellhounds so when it collides you want the hellhound to disappear to give the illusion it's dead and you want the colt to disappear because it has hit the hellhound.
	Colt Coordinates == Leviathan Coordinates	Invalid	The weapon disappears off the screen	This test is important because the colt doesn't do any damage to the leviathan however, it has still struck the leviathan so it needs to disappear.
Bone Collision with enemy	Bone Coordinates == Death	Invalid	The weapon disappears off the screen	This test is important because the bone doesn't do any

	Coordinate s			damage to death however, it has still struck death so it needs to disappear.
	Bone Coordinate s == Spirit Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the bone doesn't do any damage to the spirit however, it has still struck the spirit so it needs to disappear.
	Bone Coordinate s == Vampire Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the bone doesn't do any damage to the vampire however, it has still struck the vampire so it needs to disappear.
	Bone Coordinate s == Witch Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the bone doesn't do any damage to the witch however, it has still struck the witch so it needs to disappear.
	Bone Coordinate s == Angel Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the bone doesn't do any damage to the angel however, it has still struck the angel so it needs to disappear.
	Bone Coordinate s == Demon Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the bone doesn't do any damage to the demon however, it has still struck the demon so it needs to disappear.

	Bone Coordinate s == Werewolf Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the bone doesn't do any damage to the werewolf however, it has still struck the werewolf so it needs to disappear.
	Bone Coordinate s == Hellhound Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the bone doesn't do any damage to the hellhound however, it has still struck the hellhound so it needs to disappear.
	Bone Coordinate s == Leviathan Coordinate s	Valid	Leviathan health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the bone kills leviathans so when it collides you want the leviathan to disappear to give the illusion it's dead and you want the bone to disappear because it has hit the leviathan.
Borax Collision with Enemy	Borax coordinate s == Death Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the borax doesn't do any damage to death however, it has still struck death so it needs to disappear.
	Borax Coordinate s == Spirit Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the borax doesn't do any damage to the spirit, however it has still struck the spirit so it needs to disappear.

	Borax Coordinate s == Vampire Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the borax doesn't do any damage to the vampire, however it has still struck the vampire so it needs to disappear.
	Borax Coordinate s == Witch Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the borax doesn't do any damage to the witch, however it has still struck the witch so it needs to disappear.
	Borax Coordinate s == Angel Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the borax doesn't do any damage to the angel, however it has still struck the angel so it needs to disappear.
	Borax Coordinate s == Demon Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the borax doesn't do any damage to the demon, however it has still struck the demon so it needs to disappear.
	Borax Coordinate s == Werewolf Coordinate s	Invalid	The weapon disappears off the screen	This test is important because the borax doesn't do any damage to the werewolf, however it has still struck the werewolf so it needs to disappear.
	Borax Coordinate	Invalid	The weapon disappears off the	This test is important because the borax

	s == Hellhound Coordinate s		screen	doesn't do any damage to the hellhound however, it has still struck the hellhound so it needs to disappear.
	Borax Coordinate s == Leviathan Coordinate s Leviathan health > 10	Valid	Leviathan health decreases by 10. The weapon also disappears off the screen.	This test is important because borax damages leviathans and kills them eventually so when it collides you want the leviathans health to decrease by 10 and you want the bone to disappear because it has hit the leviathan.
	Borax coordinate s == Leviathan coordinate s Leviathan health <= 10	Valid	Leviathan health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	This test is important because the borax will kill the leviathan if the health is less than or equal to 10 so when it collides you want the leviathan to disappear to give the illusion it's dead and you want the borax to disappear because it has hit the leviathan.
	Bullet on Screen	Valid	The x coordinate of the bullet increases	This test is important because you want the bullet to look like it was fired at the enemy and you want the bullet to eventually hit the enemy.
Bullet Hit	Bullet	Valid	Bullets disappear	This test is important

	coordinate s == Enemy coordinate s		and enemy disappears	to carry out because you want the enemy to disappear when the bullets hit them because otherwise it will look like they are still alive,
	Player health decreases by 10	Valid	The health bar should be the width of the players health making part of the health bar green and part red	This test is important because you want the health bar to be proportional to the health so if the size of the bar is the same as the health then it is proportional and the user can visually see how high their health is.
	Player health gets to zero	Valid	The health bar should be the width of the player health therefore it should all be red	This test is important because the user might get a bit confused if their character disappears so the health bar being all red will let them see they died.
Score				
	Enemy is removed off screen	Valid	The players score increases by 100	This test is important because you want the score to increase when you kill an enemy.
	Player x coordinate s == enemy x coordinate s and player y	Valid	The players score increases by 10	This test is important because if the player chooses to jump over the enemy the score should still increase because they didn't die and

	coordinate s!= enemy y coordinate s			they got past the enemy.
	Bullet hits enemy but enemy health remains the same	Valid	The players score decreases by 10	This test is important because you want the score to decrease if they use the wrong bullet because you don't want them to spam bullets.
	Bullet hits enemy and enemy health decreases	Valid	The players score increases by the amount of damage taken off the enemy	This test is important because some weapons only damage enemies but you still want the score to increase because hey didn't use a weapon that takes off zero damage

Menu

	Running the code initially	Valid	The menu should be displayed	This test is important because you want the menu to load as soon as the code is run so that the user can use the menu to select what they want to do.
	Mouse is hovering over the play button and the left button is clicked.	Valid	The game should load	This test is important because if the user pressed play you want them to be able to play the game.
	Mouse is	Invalid	This shouldn't	This test is important

	hovering over the play button and the right button is clicked		affect anything on the game because the button hasn't been pressed.	because the game should only load if the left button is clicked, not the right.
	Mouse is hovering over the how to play button and the left button is clicked.	Valid	The how to play screen should load	This test is important because if the user pressed how to play you want them to be able to find out how to play the game.
	Mouse is hovering over the how to play button and the right button is clicked	Invalid	This shouldn't affect anything on the game because the button hasn't been pressed.	This test is important because the how to play menu should only load if the left button is clicked, not the right.
	Mouse is hovering over the leaderboar d button and the left button is clicked.	Valid	The leaderboards should load.	This test is important because if the user pressed leaderboards, you want them to be able to view the game's leaderboards.
	Mouse is hovering over the leaderboar d button and the right button is clicked	Invalid	This shouldn't affect anything on the game because the button hasn't been pressed.	This test is important because the leaderboards should only load if the left button is clicked, not the right.
	The user types a three letter	Borderline	The game should load	This test is important because you want the game to load if

	name into the input text box And presses play			the user inputs a three letter name.
	The user types a ten letter name into the input text box And presses play	Borderline	The game should load	This test is important because you want the game to load if the user inputs a ten letter name.
	The user types a two letter name into the input text box And presses play	Invalid	An error should pop up telling the user that the name has to be between 3 and 10 characters.	This test is important because you don't want the game to load if the user inputs a two letter name.
	The user types an eleven letter name into the input text box And presses play	Invalid	An error should pop up telling the user that the name has to be between 3 and 10 characters.	This test is important because you don't want the game to load if the user inputs an eleven letter name.

Post Development Testing

Function	Test Data	Expected Result	Justification for test
Player - Movement	The player presses	The player moves upwards then	This test ensures that the game is playable. Without having the ability

	space to jump	downwards giving the illusion that the player is jumping	to jump a huge feature of my game has been left out. The game is built around this feature and a few others.
Player - Collision	The player collides with the different objects	Enemy collision - The player object gets deleted and the health of the player goes to zero. Coin collision - The coin object gets deleted and the money variable increases	This test ensures that the game is playable. Without the player responding correctly to the enemies then there is going to be a huge error in my game making it unplayable and the game would go on forever with no way of being able to win it. The game is built around this feature and a few others.
Enemy - Collision	The different types of enemies collide with the different types of weapons.	Enemy collides with the weapon that kills it - The enemy health goes to zero and the enemy object gets deleted and a coin takes its place. Enemy collides with the weapon that damages it - The enemies health decreases by a bit but it doesn't go to zero unless the health is already low. Enemy collides with a weapon that does no damage to it - The enemies health remains the same and the enemy remains the same.	This test ensures that the game is playable. Without the enemies responding to the weapons correctly the game won't be playable because the enemies never die meaning the game will end straight away. The game is built around this feature and a few others.
GUI - Main Menu	The different buttons are clicked	The right feature is used when the buttons are clicked.	This test is important because it enables my game to run more smoothly and because it is the first thing that a player sees when they load the game, it is crucial that

			there are no mistakes within it because it sets a bad impression for the rest of the game.
GUI - Weapon Menu	The “e” button is clicked and the different buttons are clicked when on the menu.	The game pauses and a menu weapon is displayed with the different weapons on. The different buttons on the menu work correctly	This test is important because this is an important function within my game. This enables the game to be played properly because it enables the user to select the correct weapon.

Algorithms

There are many features implemented within my game that need to make use of algorithms. These algorithms have to be efficient enough so that memory isn't wasted and it will run quicker.

Below I have included pseudocode algorithms that I am going to use for the different features within my game. They are going to slightly differ from what I include in my game because these are just rough plans.

Controls

These algorithms are the algorithms that control the keyboard input from the user. This is crucial because these are the main controls that make the game playable by the user. It is essential to plan these algorithms because if there are any bugs within these sections of code then the user won't be able to play my game.

Player Jump

When this algorithm is run if the player isn't already jumping and they press space then the variable jump will be set to true to indicate that the player is jumping and then every game loop it runs this algorithm so the player will keep going upwards until it has finished its jump and then it will go downwards after. The variables are reset at the end to ensure the player is at the start position again.

```

02    keys IS EQUAL TO pygame.key.get_pressed()
03    IF NOT JUMP THEN
04        IF keys[pygame.K_SPACE] THEN
05            Jump IS EQUAL TO TRUE
06        END IF
07    ELSE THEN
08        IF jumpCount IS GREATER THAN OR EQUAL TO -20 THEN
09            Player.rect.y IS EQUAL TO Player.rect.y MINUS jumpCount
10            jumpCount IS EQUAL TO jumpCount -2
11        ELSE THEN

```

```
12      jumpCount IS EQUAL TO 20
13      Jump IS EQUAL TO FALSE
14      Player.rect.y IS EQUAL TO (infoObject.current_h MULTIPLIED
BY(107 DIVIDED BY 144)) MINUS 30
15      END IF
16  END IF
```

Game Escape

When this algorithm is run if the escape key is pressed then the game exit variable is set to true and pygame quits itself.

```
01  FOR event IN pygame.event.get THEN
02      IF EventType IS EQUAL TO KEYDOWN THEN
03          IF EventKey IS EQUAL TO EscapeKey THEN
04              gameExit EQUALS True
05              pygame.quit()
06              quit()
```

Player Fire Weapon

In this algorithm it checks to see if a key has been pressed and if e has been pressed then it creates a bullet object from the bullet class and displays it from the players centre. It also adds this bullet object to the bullet group.

```
01  FOR event IN pygame.event.get THEN
02      IF EventType IS EQUAL TO KEYDOWN THEN
03          IF EventKey IS EQUAL TO pygame.K_e THEN
04              bulletObject EQUALS Bullet(Player.rect.centerX,
Player.rect.centerY,20,5)
05              BulletGroup.add(bulletObject)
06          END IF
07      END IF
08  END FOR
```

Enemy movement

An important feature for the game is that the enemy moves so to make the enemy move this one algorithm is used to decrease the x coordinate of the enemy by the speed of the enemy.

```
01  self.rect.x EQUALS self.rect.x MINUS self.speed
```

Player and enemy removal

Without player and enemy collision the game would be endless so it is important to plan out this collision to ensure that this feature works correctly. It is also important to do their checks so that the player and enemy are removed when needed

In this algorithm it checks to see if any enemy has collided with a player and if it has the player's health gets set to zero. It also checks to see if the enemy has gone off the

screen and if it has then it will remove the enemy and increase the enemy count which will cause another enemy to spawn.

```
01 FOR eachEnemy IN EnemyGroup THEN
02     Enemyhits EQUALS pygame.sprite.spritecollide( eachEnemy,
SpriteGroup, TRUE)
03     IF eachEnemy.rect.x IS LESS THAN 0 THEN
04         EnemyGroup.remove(eachEnemy)
05         EnemyCount EQUALS EnemyCount ADD 1
06     END IF
07     IF eachEnemy.rect.centerx IS LESS THAN OR EQUAL TO
Player.rect.centerx ADD 5 AND eachEnemy.rect.centerx IS GREATER THAN OR
EQUAL TO Player.rect.centerx MINUS 1 THEN
08         score EQUALS score ADD 10
09         Player.health EQUALS Player.health MINUS 10
10     END IF
11     IF Enemyhits THEN
12         Player.health EQUALS zero
13     END IF
14 END FOR
```

Auto Scrolling Screen

In my game to give the illusion that the hunter is running towards the player I am going to implement an auto scrolling screen So i need to ensure that I plan an algorithm for this so that this feature works.

```
01 self.bgX1 EQUALS self.bgX1 MINUS self.movingSpeed
02 self.bgX2 EQUALS self.bgX2 MINUS self.movingSpeed
03 IF self.bgX1 IS LESS THAN OR EQUAL TO -self.rectBGimg.width THEN
04     self.bgX1 EQUALS self.rectBGimg.width
05 END IF
06 IF self.bgX2 IS LESS THAN OR EQUAL TO -self.rectBGimg.width THEN
07     self.bgX2 EQUALS self.rectBGimg.width
08 END IF
09 Screen.blit(self.bgimage, (self.bgX1, self.bgY1))
10 Screen.blit(self.bgimage, (self.bgX2, self.bgY2))
```

Weapon collision

This is the largest algorithm that I am including within my game so it is important that I plan this algorithm out carefully. This algorithm checks the weapons against then enemy to see if it is going to kill the enemy or not.

```
01 FOR eachBullet IN BulletGroup THEN
02     IF eachBullet.rect.x IS EQUAL TO infoObject.curren_w THEN
03         BulletGroup.remove(eachBullet)
```

```

04      END IF
05      IF eachBullet.bullet IS EQUAL TO "Scythe" AND enemy.monster IS
EQUAL TO "death" THEN
06          BulletHits EQUALS pygame.sprite.spritecollide(eachBullet,
EnemyGroup, True)
07          IF BulletHits THEN
08              EnemyCount EQUALS EnemyCount ADD 1
09              score EQUALS score ADD 100
10          END IF
11          ELSE IF eachBullet.bullet IS EQUAL TO "Salt" AND enemy.monster IS
EQUAL TO "spirit" THEN
12              BulletHits EQUALS pygame.sprite.spritecollide(eachBullet,
EnemyGroup, True)
13              IF BulletHits THEN
14                  EnemyCount EQUALS EnemyCount ADD 1
15                  score EQUALS score ADD 100
16              END IF
17              ELSE IF eachBullet.bullet IS EQUAL TO "AngelBlade" AND
enemy.monster IS EQUAL TO "angel" THEN
18                  BulletHits EQUALS pygame.sprite.spritecollide(eachBullet,
EnemyGroup, True)
19                  IF BulletHits THEN
20                      EnemyCount EQUALS EnemyCount ADD 1
21                      score EQUALS score ADD 100
22                  END IF
23                  ELSE IF (eachBullet.bullet IS EQUAL TO "Machete" OR eachBullet.bullet
IS EQUAL TO "Colt") AND enemy.monster IS EQUAL TO "vampire" THEN
24                      BulletHits EQUALS pygame.sprite.spritecollide(eachBullet,
EnemyGroup, True)
25                      IF BulletHits THEN
26                          EnemyCount EQUALS EnemyCount ADD 1
27                          score EQUALS score ADD 100
28                      END IF
29                      ELSE IF eachBullet.bullet IS EQUAL TO "WitchBullet" AND
enemy.monster IS EQUAL TO "witch" THEN
30                          BulletHits EQUALS pygame.sprite.spritecollide(eachBullet,
EnemyGroup, True)
31                          IF BulletHits THEN
32                              EnemyCount EQUALS EnemyCount ADD 1
33                              score EQUALS score ADD 100

```

```

34         END IF
35     ELSE IF enemy.monster IS EQUAL TO "leviathan" AND (eachBullet.bullet
IS EQUAL TO "Bone" OR eachBullet.bullet IS EQUAL TO "Borax" THEN
36         IF eachBullet.bullet IS EQUAL TO "Bone" THEN
37             BulletHits EQUALS pygame.sprite.spritecollide(eachBullet,
EnemyGroup, True)
38             IF BulletHits THEN
39                 EnemyCount EQUALS EnemyCount ADD 1
40                 score EQUALS score ADD 100
41             END IF
42         ENDIF
43         IF eachBullet.bullet IS EQUAL TO "Borax" THEN
44             IF enemy.health IS GREATER THAN 10 THEN
45                 BulletHits EQUALS
pygame.sprite.spritecollide(eachBullet, EnemyGroup, False)
46             IF BulletHits THEN
47                 enemy.health EQUALS enemy.health
MINUS 10
48                 score EQUALS score ADD 10
49             END IF
50         ELSE IF enemy.health IS LESS THAN OR EQUAL TO 10
THEN
51             BulletHits EQUALS
pygame.sprite.spritecollide(eachBullet, EnemyGroup, True)
52             IF BulletHits THEN
53                 enemy.health EQUALS enemy.health
MINUS 10
54                 score EQUALS score ADD 10
55                 EnemyCount EQUALS EnemyCount ADD 1
56             END IF
57             END IF
58         END IF
59         ELSE IF enemy.monster IS EQUAL TO "hellhound" AND (eachBullet.bullet
IS EQUAL TO "AngelBlade" OR eachBullet.bullet IS EQUAL TO "DemonBlade" OR
eachBullet.bullet IS EQUAL TO "Colt" OR eachBullet.bullet IS EQUAL TO "Salt")THEN
60             IF eachBullet.bullet IS EQUAL TO "AngelBlade" OR
eachBullet.bullet IS EQUAL TO "DemonBlade" OR eachBullet.bullet IS EQUAL TO
"Colt" THEN
61                 BulletHits EQUALS pygame.sprite.spritecollide(each Bullet,
EnemyGroup, True)

```

```

62      IF BulletHits THEN
63          EnemyCount EQUALS EnemyCount ADD 1
64          score EQUALS score ADD 100
65      END IF
66  END IF
67  IF eachBullet.bullet IS EQUAL TO "Salt" THEN
68      IF enemy.health IS GREATER THAN 10 THEN
69          BulletHits EQUALS
pygame.sprite.spritecollide(eachBullet, EnemyGroup, False)
70      IF BulletHits THEN
71          enemy.health EQUALS enemy.health
MINUS 10
72          score EQUALS score ADD 10
73      END IF
74  ELSE THEN
75      BulletHits EQUALS
pygame.sprite.spritecollide(eachBullet, EnemyGroup, False)
76      IF BulletHits THEN
77          score EQUALS score MINUS 10
78      END IF
79  END IF
80  END IF
81  ELSE IF (eachBullet.bullet IS EQUAL TO "Machete" OR eachBulet.bullet
IS EQUAL TO "AngelBlade" OR eachBullet.bullet IS EQUAL TO "SilverBullet" OR
eachBullet.bullet IS EQUAL TO "Colt") AND enemy.monster IS EQUAL TO "werewolf"
THEN
82      BulletHits EQUALS pygame.sprite.spritecollide(eachBullet,
EnemyGroup, True)
83      IF BulletHits THEN
84          EnemyCount EQUALS EnemyCount ADD 1
85          score EQUALS score ADD 100
86      END IF
87  ELSE IF (eachBullet.bullet IS EQUAL TO "AngelBlade" OR
eachBullet.bulet IS EQUAL TO "DemonBlade" OR eachBullet.bullet IS EQUAL TO
"Colt") AND enemy.monster IS EQUAL TO "Demon" THEN
88      BulletHits EQUALS pygame.sprite.spritecollide(eachBullet,
EnemyGroup, True)
89      IF BulletHits THEN
90          EnemyCount EQUALS EnemyCount ADD 1
91          score EQUALS score ADD 100

```

```

92      END IF
93  ELSE THEN
94      BulletHits EQUALS pygame.sprite.spritecollide(eachBullet,
EnemyGroup, False)
95      IF BulletHits THEN
96          score EQUALS score MINUS 10
97      END IF
98  END IF
99  IF BulletHits THEN
100      BulletGroup.remove(eachBullet)
101  END IF
102 END FOR

```

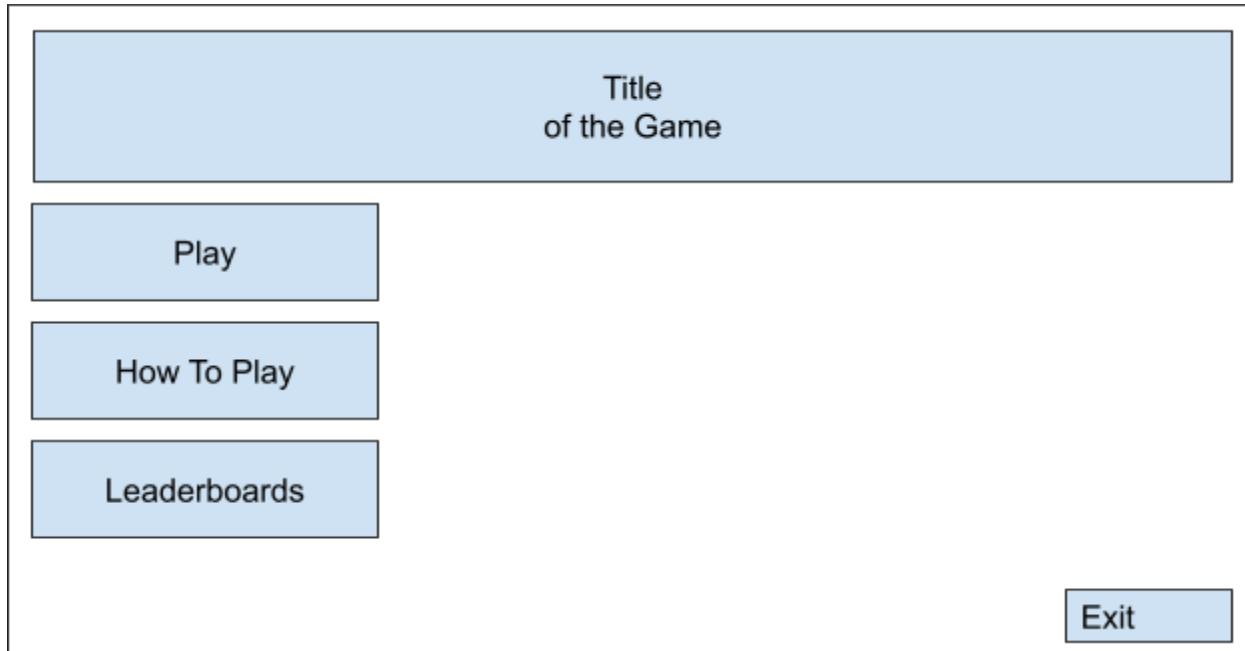
Design Objectives

In order to start developing my game I need to design the menus for the game. For this I am going to have to consider a few things like the inputs, processes and outputs and how I can make my game and menus more accessible.

User Interface Design

When I create my GUI's they will be displayed as full screen on the users screen. This means that for my design I will create a rectangle and when I actually create my GUI it will have to use maths so that it is proportionate to the size of the user's screen.

Main Menu Screen

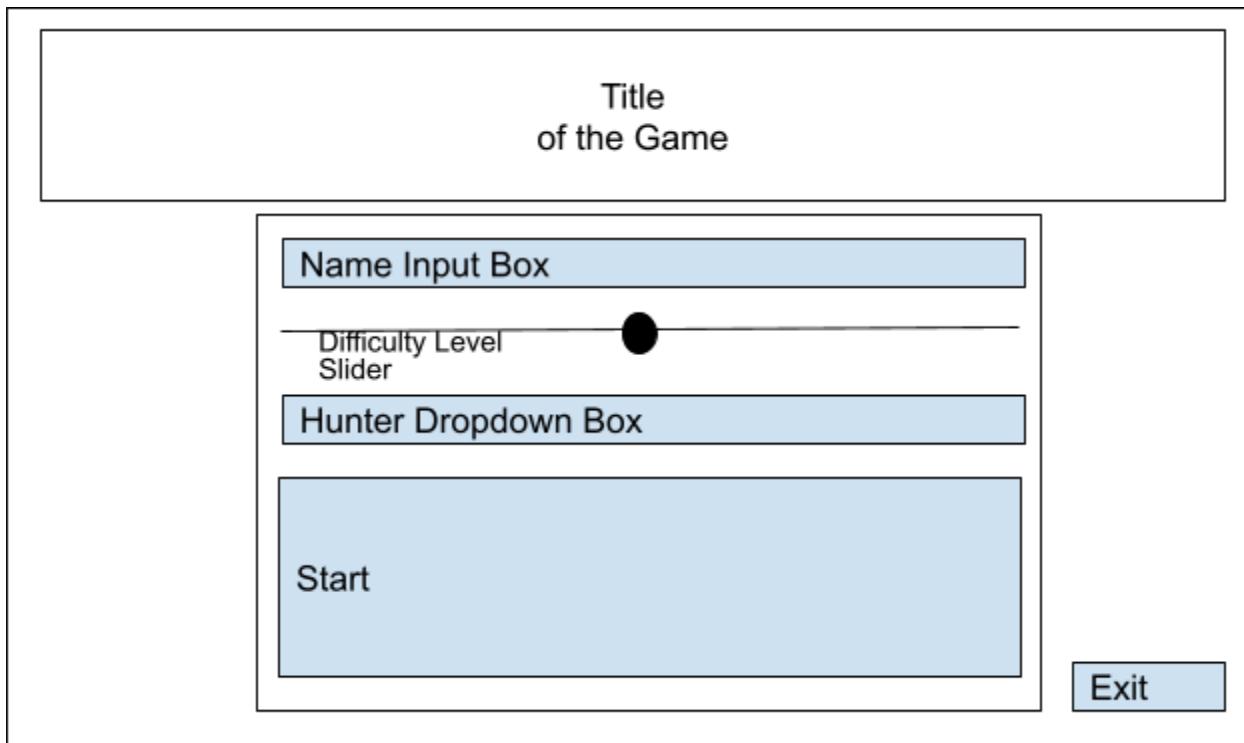


The main menu of my screen has a very simple layout because it needs to be easy to navigate. The title of the game is displayed in large at the top because it needs to be memorable for the user because if they like it then it means that they will know what it is

called. The play button will take the user to another screen that enables them to input their name, choose a difficulty and choose which hunter they want to be. I have chosen for these options to be displayed on another menu because you don't want the main menu to be full of different buttons because this can get confusing for a user.

The "how to play" button will take the user to a screen that tells the user how to play the game because when someone first plays the game they won't know how to play it. The "leaderboards" button will take the user to the leaderboards. The exit button is going to exit the game and it is in the bottom right of the screen so then people don't see it when they go to click play meaning they are more likely to continue playing.

Play Game Menu



The play game menu has all of the settings to start the game. The title of the game is still displayed at large at the top because it means it will still remain memorable and it means the design is similar to the design of the main menu therefore there will be more familiarity with the user. The name input box will be clearly labelled so that the user knows what they have to type in that box and it will have validation rules applied to it so that the name can't be too long or too short. There will be a slider so that the user can choose their difficulty level and this slider will be clearly labelled to ensure that the user knows what it is there for. Below that is a hunter dropdown box so that the user is able to play as their favourite hunter meaning that more people are likely to play the game because more people will be able to use the character that they want to choose. The start button is below this which will take the user to the game. The start button is slightly larger because this makes it stand out a bit more. The exit button is still displayed in the

bottom right corner of the screen because this therefore makes the game more accessible to the user.

Leaderboards Menu

Leaderboards					
Easy:	Medium:	Hard:			
01 AAA 02 BBB 03 CCC 04 DDD 05 EEE 06 FFF 07 GGG 08 HHH 09 III 10 JJJ	999 888 777 666 555 444 333 222 111 000	01 AAA 02 BBB 03 CCC 04 DDD 05 EEE 06 FFF 07 GGG 08 HHH 09 III 10 JJJ	999 888 777 666 555 444 333 222 111 000	01 AAA 02 BBB 03 CCC 04 DDD 05 EEE 06 FFF 07 GGG 08 HHH 09 III 10 JJJ	999 888 777 666 555 444 333 222 111 000

Exit

The leaderboard displays all three leaderboards on one screen. It is laid out so that easy is to the left, Medium is in the middle and hard is to the right. This is more accessible because the difficulties are in a clear order rather than just being random. The exit button is still displayed in the bottom right because this is where it is displayed in all of the other menus so therefore it is still accessible.

Aesthetic Considerations

- The resolution of the game is going to be set to the resolution of the user's screen. This means that I am going to have to place things three quarters of the way down the screen rather than having a fixed position on the screen for items. Things are also going to have a different size depending on the size of the screen.
- The font is going to match the theme of the supernatural because this is what my game is about. This is going to make my game more appealing to the user. Below are a few examples of fonts that I might use. My favourite out of these fonts is Horor Drama because this matches when they engrave their initials in the table in the bunker so therefore this font matches the series.

Another Danger by The Branded Quotes ↗

Bloodsoul ↗ by Kong Font ↗

SUPERNATURAL

SUPERNATURAL

Melted Monster ↗ by Dim Letter Studio - Dimas Prasetyo ↗

Doctor Terror by Woodcutter ↗

SUPERNATURAL SUPERNATURAL

Horror Drama by Noah Type ↗

SUPERNATURAL

- The background of my menus are going to be black and the background of my game is going to be a dark red. These colours match the supernatural theme and the dark red also gives the illusion that the characters are fighting monsters in hell which is what I want my game to look like.
- The text colour on the menus and the game should be white because the background of the game and menus are dark colours you will need a lighter colour for the text so that the user can read it.
- At the top of the main menu and the customisation menu the title of the game will clearly be displayed and it will match the theme of the background and text colour because this will make it more appealing to the user and therefore they are more likely to play the game.

Input Considerations

- On the main menu the user should be able to press buttons that allow them to access the customisation menu, “how to play” menu and the leaderboards.
- On all GUI’s there should be an exit button in the bottom right corner. The one on the main menu exits the game and the ones on the Customiser menu, “how to play” menu and leaderboards take you back to the main menu.
- On the customiser menu the user should be able to input a name that is between three and ten characters which will be stored in the database with their score.
- On the customiser menu the user should be able to drag a slider left and right to select the difficulty that they want to play at.
- On the Customiser menu the user should be able to click a dropdown box that displays a list of hunters and then they can select which one they want to play as.
- Whilst in game there should be no lagging so when the user presses space the player will move upwards and back down again to give the illusion that they are jumping.

- In game pressing the letter “E” will fire the bullet that was last selected by the user.

Processing Considerations

- The resolution of the user's screen should be detected and then the size of everything and positioning of everything will be determined based on the user's screen size. The game and menu will always be displayed full screen on the user's device.
- The users name and score will be sent to a database that will be displayed in the form of a leaderboard on the leaderboards screen
- A random number generator will be used to determine what monster comes onto the screen next.
- The monster's health will be set to full at the start and if the monster is hit by a specific weapon then their health will decrease to zero and other weapons can decrease the health by smaller increments at a time. When a monster's health is zero they are removed from the screen and the user's score increases.
- The player's health will be set to full at the start and if they get hit by a monster their health will go to zero. If they jump over the monster then their health will decrease by 10. When the player's health hits zero, the game is over and you get returned to the main menu.
- The leaderboard's GUI should access the database where the score and names are displayed and it should sort them and only display the top ten for each difficulty.

Output Considerations

- Sounds should play when the player dies, jumps, lands and collects coins, fires a weapon, clicks a button and sound should play when the weapon hits the enemy.
- Music should be playing on any menus and on the main game at all times.
- The score should be updated on the screen every time it increases and decreases.
- The health bar should go down everytime health is taken off the enemy.
- The player should be visible at all times during the game unless they have died and it switches to the menu.

UML Diagrams

Within my program I will have to include different classes that can have many different objects coming from it each with slightly different features. With objects I will need many different types to be created and it will take more lines of code if I don't use classes so using a class will save time and memory. It will also make my code more readable.

Attributes

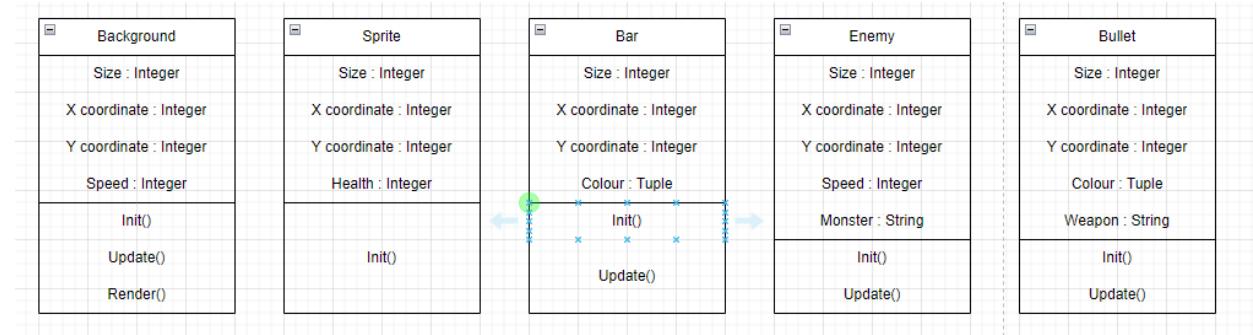
For all of the classes they need a size, x-coordinate and y-coordinate attribute so that the position on the screen can be set for the objects and also the size of them so then

things aren't too small or too large. All the objects that move across the screen also need a speed attribute that sets how fast or slow it moves. The enemy and the sprite need health attributes because these attributes affect if the game continues and the score of the game. The enemy class has a monster attribute and the weapon class has a weapon attribute because these are the attributes that are going to be used to control whether the enemy can be killed by the weapon.

Methods

All of the classes have init methods because it initialises all of the attributes that are going to be used. Most of them also have update methods so that they can move whilst the program is running. The background then has a render method so that the graphics are rendered because it is constantly scrolling left.

Class Diagrams



Validation plan

I want my game to be able to run correctly without any bugs within it so I am going to have to create a validation plan to ensure that this happens.

- The player spawns to the left of the screen a quarter of the way across and above the ground so that the feet are on the ground. - I have to include this in my validation plan because you want the player to spawn far enough to the left so that they have time to select the correct weapon and kill the player and you also want it to spawn on the ground so that it appears like the player is walking. To make sure this works correctly I will get the resolution of the screen and divide it by 4 to get the x - coordinate and for the y - coordinate I will use maths to figure out where the players are going to be spawned.
- The enemy spawns to the right just off the screen and above the ground so that the feet are on the ground - I have to include this within my validation plan because you don't want the enemy to just appear on the screen, you want it to appear like it is moving onto the screen. To make sure that this works correctly I will set the top left corner of the enemy to spawn at the width of the screen and I will do the same maths as I did for the player to get the y - coordinate.

- The enemy is removed when it goes off screen - I have to include this within my validation plan because if the enemy isn't removed there will be too many enemies still moving which will therefore make the game run slower because there are too many graphics which mean the game won't run smoothly. To make sure that this runs correctly I will add an if statement within the game loop that checks whether the x - coordinate of the enemy is zero and if it is then the enemy is deleted.
- The user can only have up to a ten letter name - I have to include this within my validation plan because we don't want the user to spam load of characters as their name because this will make the file size much larger and also it means that if they are in the first ten it won't fit everyone else's names on the leaderboard. To make sure that this runs correctly I will use .len to do a length check and the user will get an error message if the length of the name doesn't meet the requirements.

Data Dictionary

All of the variables mentioned in the table below will be implemented within my code because they are crucial for the game to properly function. This is not all of the variables that will be used because it isn't possible to know all of the variables I will use at this current time but these are the main ones that I will need for the game to run.

Variable Name	Data Type	Data Range	Example Data	Justification
Black, Purple, Red, DarkRed, Green	RGB colour set	(0,0,0) to (255,255,255)	(133, 245, 97)	These are variables that store the rgb colour values which I can use later on to change the colours of the background, buttons and text.
Jump	Boolean	True/False	True	This variable is used to determine whether the character should jump. If it is true it will run an if statement within the game loop so it is important to include this variable within my game.
JumpCount	Integer	0+	10	This is a variable that determines how high the

				player is going to jump so it is important because jumping is a key feature of my game.
Score	Integer	0+	456	This is a variable that determines how well the user is playing the game. This variable starts at 0 when the user first loads the game but as they play the game it increases and then they can compare it to other users scores on the leaderboard.
EnemyCount	Integer	0+	1	This is a variable that determines how many enemies need to be spawned on the screen. This can be changed throughout to increase the number of enemies on screen and is also used so that when an enemy dies another one is spawned.
infoObject	Integer	Resolution of the screen	infoObject.current_w = 8000 infoObject.current_h = 2000	This variable is used to store the width and height of the screen so that the game can be played in full screen.
Clock	Float	0+	30	This variable determines the FPS of the game. This means that the game can be slowed down and sped up.
gameExit	Boolean	True/False	True	This variable is used to determine whether or not the game loop should be running. This is a crucial

				part of the game because without it the game can't be run.
event	list	Multiple Data Types	unsure	This variable comes with the pygame library and takes input in from the user. This can be modified to detect if certain keys have been pressed. Without this variable the user wouldn't be able to play the game because the program wouldn't be able to take input.
ImageSpeed	Integer	1+	5	This variable determines how fast the screen scrolls to the left. This variable isn't necessary but makes the game more appealing to the user because it gives the illusion that the player is running.
PlayerHealth	Integer	0-100	45	This variable determines the health of the player. This variable is necessary because it helps to determine when the game has ended.
EnemyHealth	Integer	0-100	79	This variable determines the health of the enemy. This variable is necessary because it means that the enemy is then able to die.
PlayerX, PlayerY	Integer	0+	400, 200	These are variables that set the coordinates for the player on the screen. These variables are updated whenever the

				player jumps.
EnemyX EnemyY	Integer	0+	800, 200	These are variables that set the coordinates for the enemy on the screen. These variables are updated every second because the enemy moves from the left to the right.

Stakeholder Signatures



Development

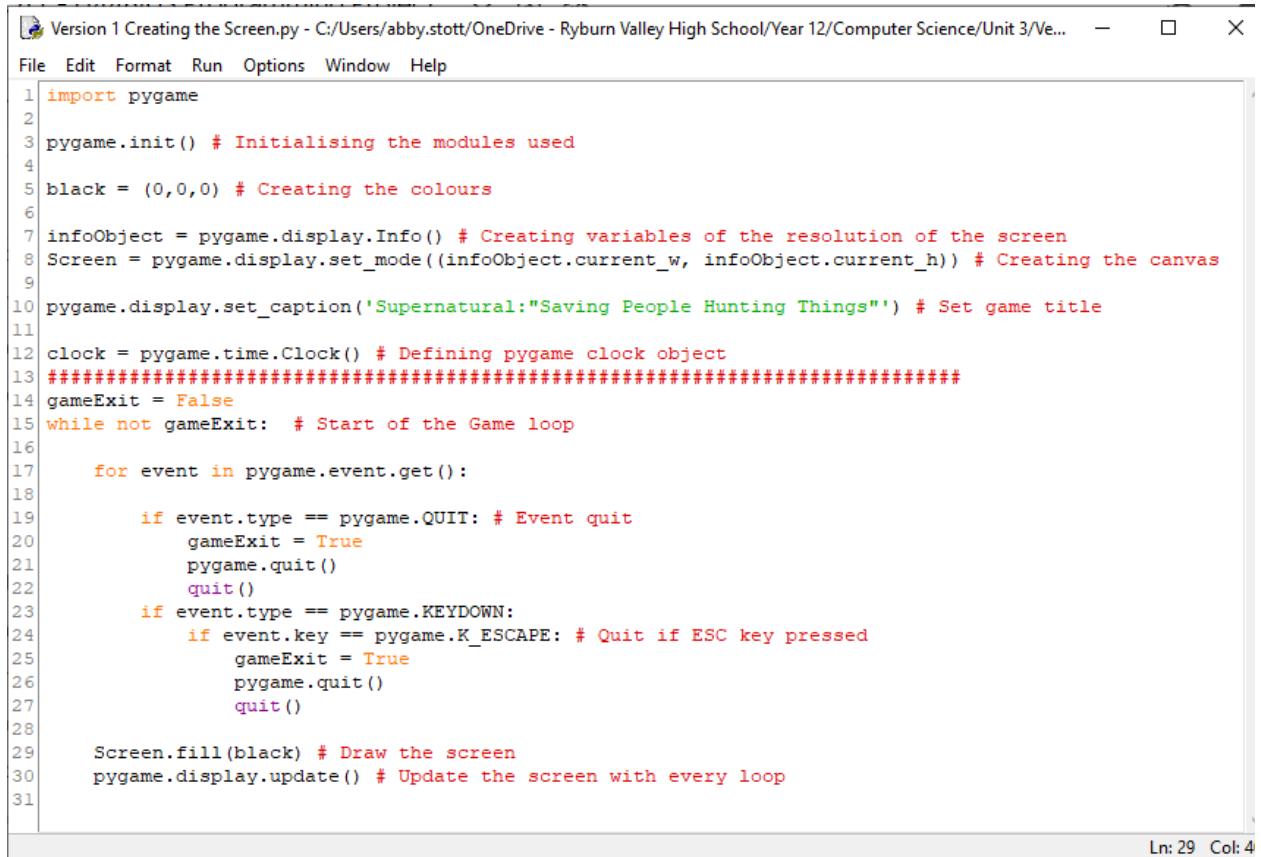
Early Planning and Practice

I have now designed my game so I have all the information required so I can go on to actually programming it. I may need to spend a bit more time on the more difficult things like having certain bullets killing the different types of monsters. I will also need to spend a bit of time on the GUI because the layout needs to suit every screen. So then the player can't predict which enemy is going to come onto the screen next, I will program it so that it is a random character. Throughout my programming I will also carry out a series of tests to ensure that my program works correctly. Before starting to make the game I have learned how to create the screen and draw things to the screen and movement to learn the basics before making my project because I have never used the pygame library before so it is useful to have a little practice beforehand.

Version 1 - Creating the Screen

To first start creating my game I need to create the screen so that the game is displayed on the users screen. In order for this to happen I have to import a python library called

pygame and initialise it. I am also having to set the parameters of the window.

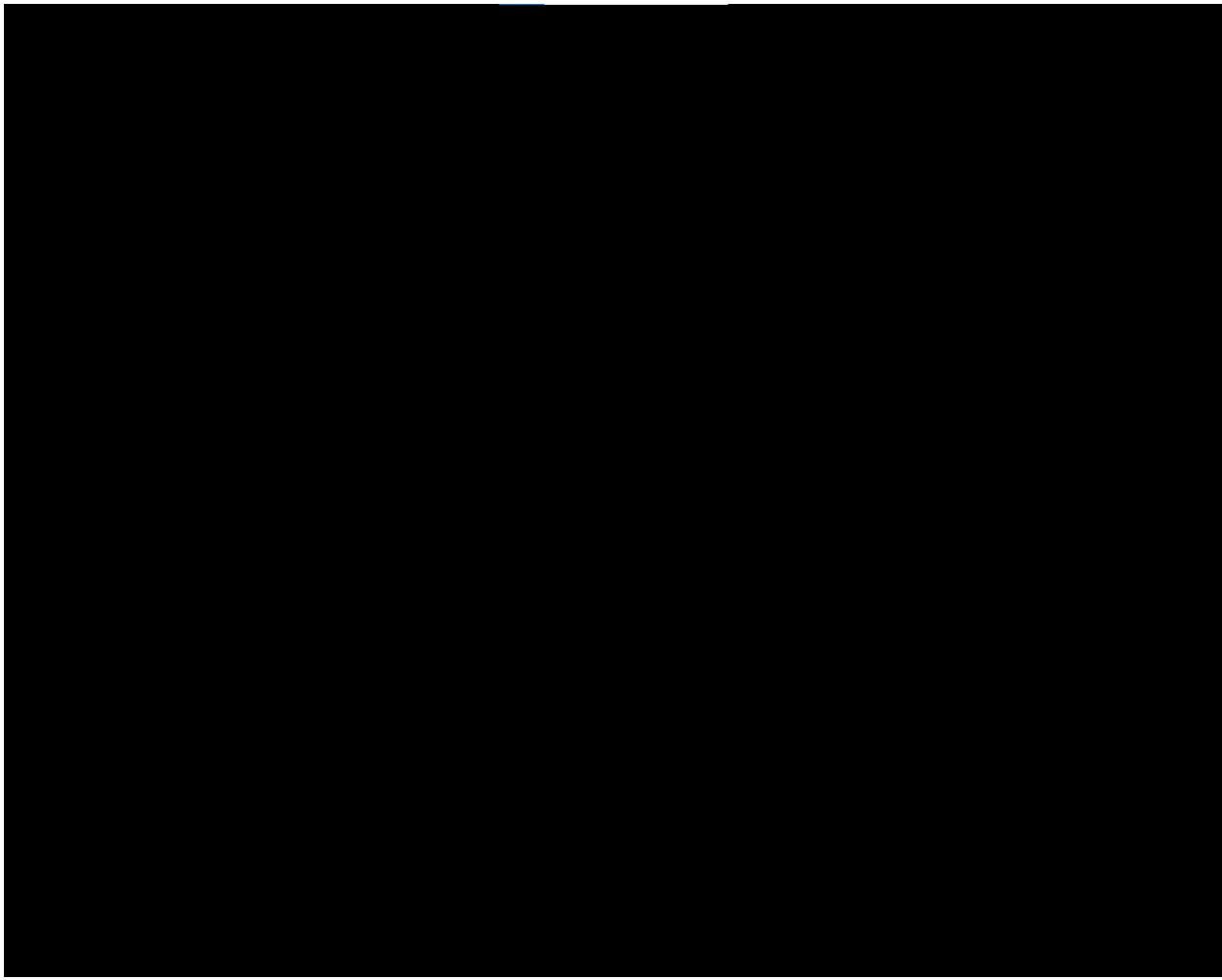


The screenshot shows a code editor window titled "Version 1 Creating the Screen.py - C:/Users/abby.stott/OneDrive - Ryburn Valley High School/Year 12/Computer Science/Unit 3/Ve...". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code itself is a Python script:

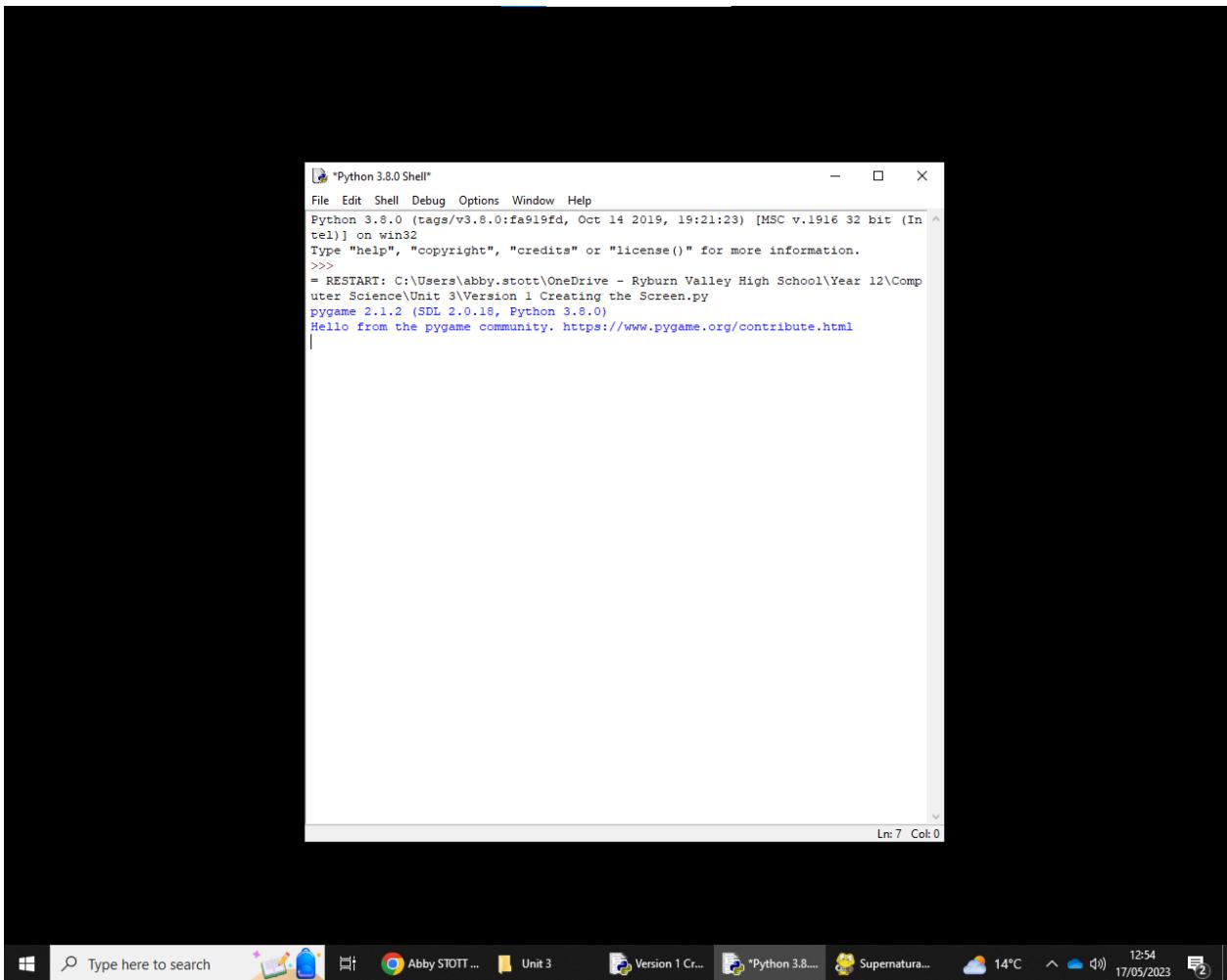
```
1 import pygame
2
3 pygame.init() # Initialising the modules used
4
5 black = (0,0,0) # Creating the colours
6
7 infoObject = pygame.display.Info() # Creating variables of the resolution of the screen
8 Screen = pygame.display.set_mode((infoObject.current_w, infoObject.current_h)) # Creating the canvas
9
10 pygame.display.set_caption('Supernatural:"Saving People Hunting Things"') # Set game title
11
12 clock = pygame.time.Clock() # Defining pygame clock object
13 ######
14 gameExit = False
15 while not gameExit: # Start of the Game loop
16
17     for event in pygame.event.get():
18
19         if event.type == pygame.QUIT: # Event quit
20             gameExit = True
21             pygame.quit()
22             quit()
23
24         if event.type == pygame.KEYDOWN:
25             if event.key == pygame.K_ESCAPE: # Quit if ESC key pressed
26                 gameExit = True
27                 pygame.quit()
28                 quit()
29
30     Screen.fill(black) # Draw the screen
31     pygame.display.update() # Update the screen with every loop
```

Ln: 29 Col: 4

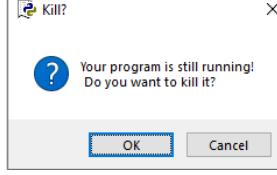
The first two lines of code imports the pygame library and initialises the modules that are going to be used so that we can use all the different pygame functions within the game. Without these two lines of code the game wouldn't work at all so these two lines are crucial to the game. Next I have created a variable **black** which has an rgb value assigned to it so that the screen has a colour. **infoObject** is a variable that stores the values of the resolution of the screen. **Screen** stores the value for the screen and **set_mode** will set the size of the screen to the parameters that are passed into it which is going to be the resolution of the user's screen that is obtained in the previous line. On the next line I have set the title of the game to what I have decided to call my game. The variable **clock** is created to control the FPS of the screen. **gameExit** is the variable that contains the boolean value that allows the game loop to work. This is important because it allows the user to end the game when they want and also when I have created my menu it ensures that the game will only load when the play button has been pressed. Below this is the game loop. The program will run through this loop until the escape key is pressed. Line 29 draws the screen and colours it in black and below that we update the screen within every loop to make the game playable.



The image above is a screenshot of the screen that has been drawn and below are the screenshots of the python shell open above the screen to show it is running and the tab on the taskbar to show the title of the game.



Tests

	Input	Test Type	Expected Outcome	Actual Outcome	Comments
Exit Game	Escape key pressed	Valid	The game should exit		The game exited as expected so this feature works correctly

Top down Modular Update

I have now started to complete things that are on my top down modular chart.



Success Criteria

Currently I haven't completed any of the features on my success criteria.

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.

- A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power-ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision

- Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
- Silver Bullet
 - Kills Werewolves upon collision
- Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
- Bone
 - Kills Leviathans upon collision
- Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 2 - Scrolling Screen

One of the main aspects of my game is that it auto scrolls to give the illusion that the character is running so I decided to make this the next thing that I add to my game. In order to create the scrolling screen I made a class for the background with an update method and a render method.

```
class Background():
    def __init__(self,image,speed):
        self.bgimage = pygame.image.load(image) # Loading the image
        self.bgimage = pygame.transform.scale(self.bgimage,((infoObject.current_w, infoObject.current_h))) # Fitting the image to the size of the screen
        self.rectBGimg = self.bgimage.get_rect()

        self.bgX1 = 0
        self.bgX2 = 0

        self.bgY1 = 0
        self.bgY2 = self.rectBGimg.width

        self.movingSpeed = speed # setting the speed of the background

    def update(self):
        self.bgX1 -= self.movingSpeed
        self.bgX2 -= self.movingSpeed
        if self.bgX1 <= -self.rectBGimg.width: #Resetting the coordinates of the image if it goes off screen
            self.bgX1 = self.rectBGimg.width
        if self.bgX2 <= -self.rectBGimg.width:
            self.bgX2 = self.rectBGimg.width
    def render(self):
        Screen.blit(self.bgimage, (self.bgX1, self.bgY1)) # Drawing the background to the screen
        Screen.blit(self.bgimage, (self.bgX2, self.bgY2))
```

The first line is creating the class for the background. From lines 2 to 14 I am creating all of the attributes that the background uses. Some of the values of the attributes are set when you create the object through parameters. The attributes load the image and set it to the size of the user's screen and also sets the x and y coordinates of the image. Lines 15 - 21 are creating the update method. The x coordinates change by taking away the speed that you set so that every loop the screen moves left by the same number of pixels. The two if statements below that are saying that if the image goes off the screen then the x coordinates will reset to the width of the screen so the image comes in from the right again to complete the loop. Lines 22-24 are creating the **render** method. In this method the images are getting redrawn with the new x coordinates that were changed within the **update** method.

```
backgrnd = Background("pixil-frame-0.png", 1) # Creating the background objects
clouds = Background("pixil-layer-1.png", 1)
clouds2 = Background("pixil-layer-2.png", 3)
ground = Background("pixil-layer-3.png", 5)
```

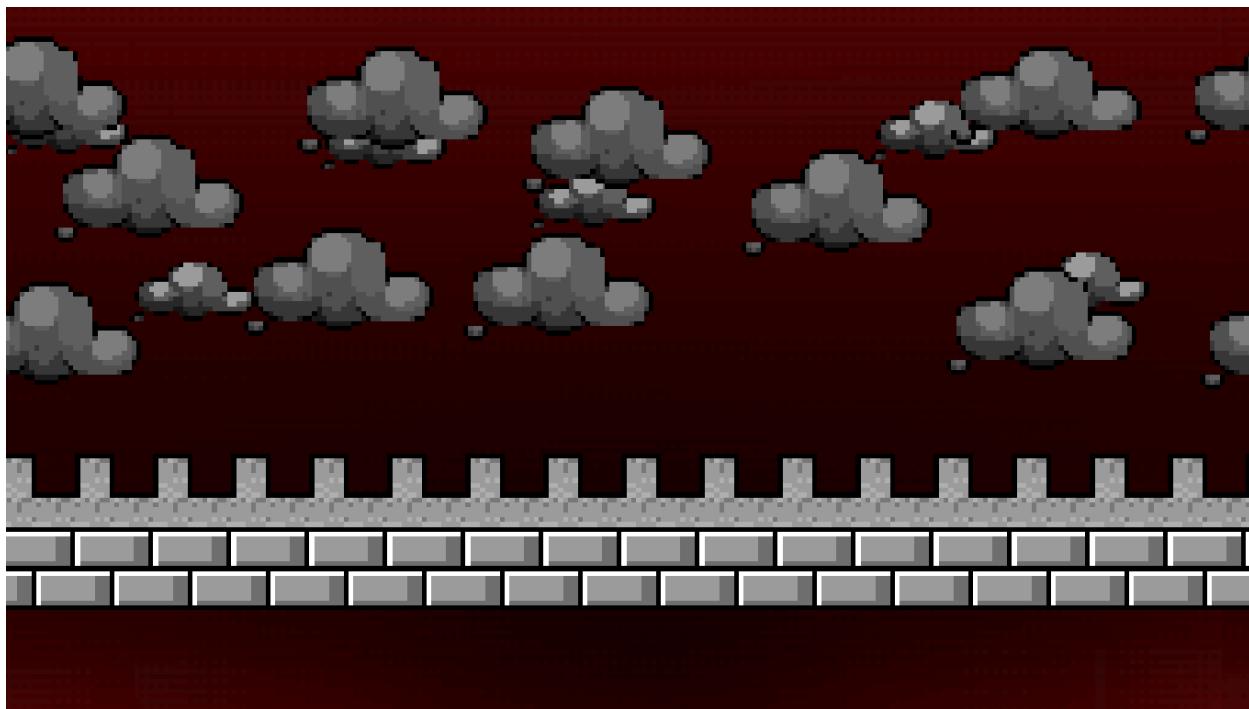
These four lines of code are creating objects for each layer of the background. They have all got different images and speeds. The backgrounds that have a slower speed are going to appear further away and the backgrounds that have a higher speed will appear closer.

```

backrnd.update()      # Running the class methods
backrnd.render()
clouds.update()
clouds.render()
clouds2.update()|
clouds2.render()
ground.update()
ground.render()

```

These lines of code are within the game loop and they are calling the methods from within the class. The **update** method changes the x coordinates and the **render** method redraws the images at the new x coordinates.



This is an image of what the screen looks like. The ground and clouds are moving at different speeds.

Tests

	Input	Test Type	Expected Outcome	Actual Outcome	Comments
Auto scrolling screen	Loading the game	Valid	x - coordinates decrease and reset once hit -(resolution	The width of the screen is 1360	The x coordinates decreased as expected so the images

		of the screen)	<pre> background xl is -1351 background xl is -1352 background xl is -1353 background xl is -1354 background xl is -1355 background xl is -1356 background xl is -1357 background xl is -1358 background xl is -1359 background xl is 1360 background xl is 1359 background xl is 1358 background xl is 1357 background xl is 1356 background xl is 1355 background xl is 1354 background xl is 1353 background xl is 1352 background xl is 1351 background xl is 1350 background xl is 1349 background xl is 1348 background xl is 1347 background xl is 1346 background xl is 1345 </pre> <pre> background x2 is -1352 background x2 is -1353 background x2 is -1354 background x2 is -1355 background x2 is -1356 background x2 is -1357 background x2 is -1358 background x2 is -1359 background x2 is 1360 background x2 is 1359 background x2 is 1358 background x2 is 1357 background x2 is 1356 background x2 is 1355 background x2 is 1354 background x2 is 1353 background x2 is 1352 background x2 is 1351 background x2 is 1350 background x2 is 1349 background x2 is 1348 background x2 is 1347 background x2 is 1346 background x2 is 1345 background x2 is 1344 background x2 is 1343 background x2 is 1342 background x2 is 1341 background x2 is 1340 </pre>	moved automatically.
Loading the game	Valid	X1 and X2 coordinates are different for each object when x coordinates change	<pre> background xl is -1 clouds 1 xl is -1 clouds 2 xl is -3 ground xl is -5 background xl is -2 clouds 1 xl is -2 clouds 2 xl is -6 ground xl is -10 background xl is -3 clouds 1 xl is -3 clouds 2 xl is -9 ground xl is -15 </pre>	The different backgrounds are at different x coordinates therefore everything moves at different speeds successfully.

				<pre>background x2 is 1359 clouds 1 x2 is 1359 clouds 2 x2 is 1357 ground x2 is 1355 background x2 is 1358 clouds 1 x2 is 1358 clouds 2 x2 is 1354 ground x2 is 1350 background x2 is 1357 clouds 1 x2 is 1357 clouds 2 x2 is 1351 ground x2 is 1345</pre>	
	Loading the game	Valid	The background moves the the left smoothly (no jittering)	<pre><Clock(fps=6.00)></pre>	The background was going really slow and was jittery so I need to improve my background so that it moves faster.

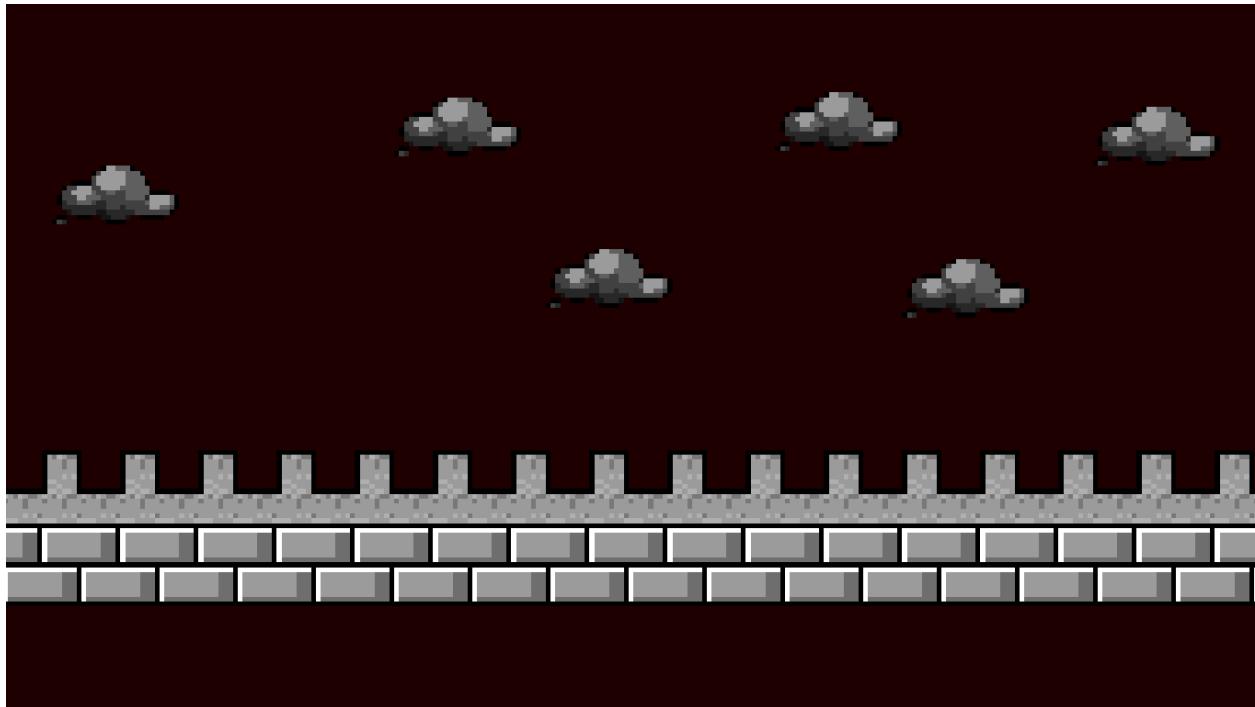
The background isn't moving as smoothly as I like so I am going to have to edit my program so therefore it is running smoother.

To improve my program I am going to remove all the layers of the background and have just one layer that moves so therefore I am not rendering as many graphics and hopefully this will speed up the game.

```
Screen.fill(DarkRed)
```

```
ground.update()
ground.render()
```

This code is within the game loop. To speed up the game I filled in the background with a dark red and then I deleted all of the other things in the background like clouds and I just rendered the ground because these are the most important graphics that are being rendered.



This is an image of the new background. This one is a lot more basic because I had to cut out the other clouds and the background image to speed it up.

Tests 2

	Input	Test Type	Expected Outcome	Actual Outcome	Comments
Auto scrolling screen	Loading the game	Valid	x - coordinates decrease and reset once hit -(resolution of the screen)	<pre>The width of the screen is 1360 background xl is -1351 background xl is -1352 background xl is -1353 background xl is -1354 background xl is -1355 background xl is -1356 background xl is -1357 background xl is -1358 background xl is -1359 background xl is 1360 background xl is 1359 background xl is 1358 background xl is 1357 background xl is 1356 background xl is 1355 background xl is 1354 background xl is 1353 background xl is 1352 background xl is 1351 background xl is 1350 background xl is 1349 background xl is 1348 background xl is 1347 background xl is 1346 background xl is 1345</pre>	The x coordinates decreased as expected so the images moved automatically.

				<pre>background x2 is -1352 background x2 is -1353 background x2 is -1354 background x2 is -1355 background x2 is -1356 background x2 is -1357 background x2 is -1358 background x2 is -1359 background x2 is 1360 background x2 is 1359 background x2 is 1358 background x2 is 1357 background x2 is 1356 background x2 is 1355 background x2 is 1354 background x2 is 1353 background x2 is 1352 background x2 is 1351 background x2 is 1350 background x2 is 1349 background x2 is 1348 background x2 is 1347 background x2 is 1346 background x2 is 1345 background x2 is 1344 background x2 is 1343 background x2 is 1342 background x2 is 1341 background x2 is 1340</pre>	
Loading the game	Valid	The background moves the the left smoothly (no jittering)		<Clock(fps=20.96)>	The background is moving more smoothly in this new piece of code so I am going to keep my background this way.

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.

- A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power - ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision

- Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
- Silver Bullet
 - Kills Werewolves upon collision
- Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
- Bone
 - Kills Leviathans upon collision
- Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 3 - Jumping Player

To make the game playable I need a character in there that the player can control so in this version I am going to add a character that has the ability to jump if you press the

space key. In order to create a character that can jump I am going to add a sprite class and some variables that will control the jumping movement.

```
Jump = False # Creating the variables for the Jump
jumpCount = 10
```

These 2 lines are creating the variables **Jump** and **jumpCount** which is what I will use to make the character jump.

```
class Sprite(pygame.sprite.Sprite):

    def __init__(self,x,y,a,b):
        super().__init__()
        self.image = pygame.Surface([a,b]) # Sets the size
        self.image.fill(purple)
        self.rect = self.image.get_rect() # Hitbox
        self.rect.x = x# Sets the position on the screen
        self.rect.y = y
```

This section of code is creating the sprite class so that the user has a character that they can play as. At the moment the character is getting filled in with just a colour because I need to make the graphics for the different characters that the users can choose from but that will be added at a later date.

```
Player = Sprite(infoObject.current_w/4, (infoObject.current_h*(107/144))-30,30,30) # Creating the player object
SpriteGroup = pygame.sprite.Group() # Create a group
SpriteGroup.add(Player) # Add object to group
```

Below the class I added these three lines. This will create a Player object and a sprite group and then the player object will get added to the sprite group.

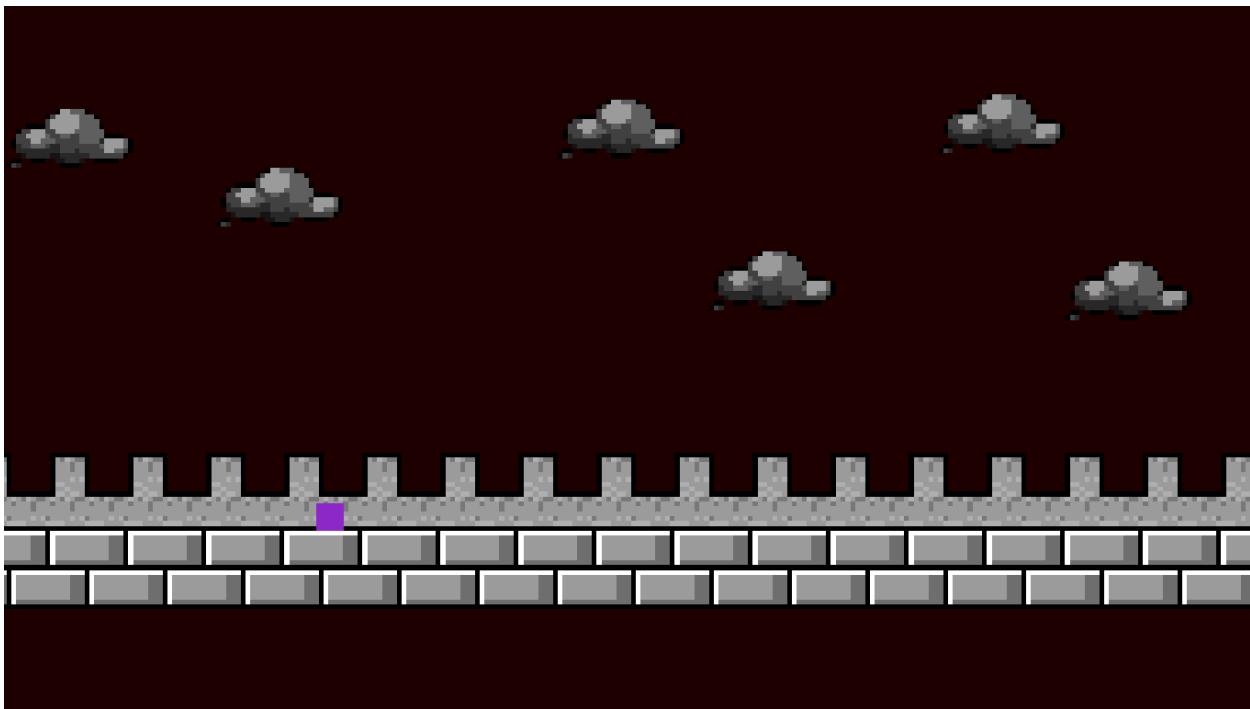
```
keys = pygame.key.get_pressed()
if not(Jump):
    if keys[pygame.K_SPACE]: # Changing Jump to true if space key is pressed
        Jump = True
else:
    if jumpCount >=-10:
        Player.rect.y -= jumpCount
        jumpCount -= 1
    else: # This will execute if our jump is finished
        jumpCount = 10
        Jump = False
        Player.rect.y = (infoObject.current_h*(107/144))-30
        # Resetting our Variables
```

This section of code is within the game loop and controls the jumping movement of the player. A variable is created that checks if a button is being pressed. If a button is pressed **keys** would equal true and if a button isn't pressed **keys** would equal false. If **Jump** is False and the space key has been pressed (this is checked by seeing if the variable for **keys** is true for the space bar) then the variable **Jump** is set to true so that when it executes the loop again it will execute the else section of the if statement. When it executes the else part it will check if the jump count is larger than or equal to -10. If this is the case, the y coordinate value decreases by jump count so that when jump

count is positive the player starts going up and when it is negative the player goes back down again therefore completing a jump. Jump count decreases by one so that the character eventually goes back down. All the variables are then reset that allow the player to jump so that when the space key is pressed the player can jump again.

```
SpriteGroup.draw(Screen)
```

This line of code is drawing the sprite group onto the screen.



This is what the game looks like with a character in it. In future versions the character will have an image but for now it is just a purple block

Tests

	Input	Test Type	Expected Outcome	Actual Outcome	Comments
--	-------	-----------	------------------	----------------	----------

Character Jump	Space Key pressed	Valid	The player should jump - The Y coordinate of the player decreases then increases until it is at its original value at the start of the game.	<pre> Y coordinate of player is 532 Y coordinate of player is 522 Y coordinate of player is 513 Y coordinate of player is 505 Y coordinate of player is 498 Y coordinate of player is 492 Y coordinate of player is 487 Y coordinate of player is 483 Y coordinate of player is 480 Y coordinate of player is 478 Y coordinate of player is 477 Y coordinate of player is 477 Y coordinate of player is 478 Y coordinate of player is 480 Y coordinate of player is 483 Y coordinate of player is 487 Y coordinate of player is 492 Y coordinate of player is 498 Y coordinate of player is 505 Y coordinate of player is 513 Y coordinate of player is 522 Y coordinate of player is 532 </pre>	The y coordinate of the player decreased and increased as expected therefore the jumping feature works correctly.
----------------	-------------------	-------	--	--	---

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.

- Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.
 - A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power - ups which you buy with coins:

- Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision
 - Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
 - Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
 - Silver Bullet
 - Kills Werewolves upon collision
 - Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Bone
 - Kills Leviathans upon collision
 - Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound

- Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
- Main game Music - When the player is playing the main game the main game music will play on loop in the background.
- Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 4 - Enemies

Like most games I am going to need enemies in my game, something that the players can defeat. In this version I am going to add enemies. In order to add the enemies I am going to have to create another class but this time it is an enemy class.

```
class Enemy(pygame.sprite.Sprite):

    def __init__(self,x,y,a,b,enemyspeed):
        super().__init__()
        self.image = pygame.Surface([a,b]) # Sets the size
        self.image.fill(red)
        self.rect = self.image.get_rect() # Hitbox
        self.rect.x = x# Sets the position on the screen
        self.rect.y = y
        self.speed = enemyspeed

    def update(self,):
        self.rect.x-=self.speed
```

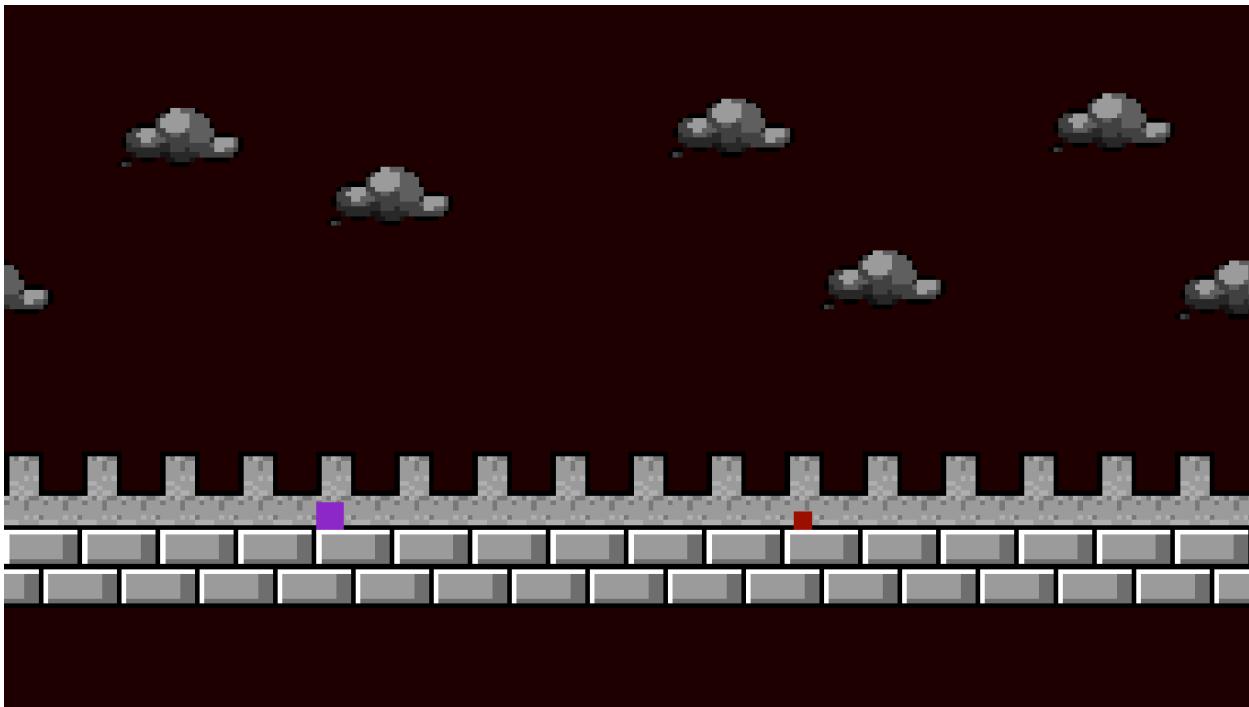
This section of code is creating the class for the enemy. The attributes: set the size of the enemy, the position on the screen and also you can set the speed of the enemy. The enemies are made smaller than the player so that the player can jump over the enemies easier and the enemies will have different speeds so that when I have multiple different types of enemies they can all move at different speeds. The enemy gets filled in, in red until I add images for the different enemies. The enemy has an update method which will make the enemy move from the right of the screen to the left of the screen by decreasing the x coordinate by the speed of the enemy.

```
Player = Sprite(infoObject.current_w/4, (infoObject.current_h*(107/144))-30,30,30)
Enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-20,20,20,10)
SpriteGroup = pygame.sprite.Group() # Create a group
EnemyGroup = pygame.sprite.Group()
SpriteGroup.add(Player) # Add object to group
EnemyGroup.add(Enemy)
```

In this section of code I added in lines 2, 4 and 6. These lines create the enemy object from the enemy class and also creates an enemy group. It then adds the enemy object into the enemy group.

```
EnemyGroup.draw(Screen)  
EnemyGroup.update()
```

These two lines are in the game loop. These lines draw the enemy group onto the screen and then updates it which causes the enemy to move closer to the player.



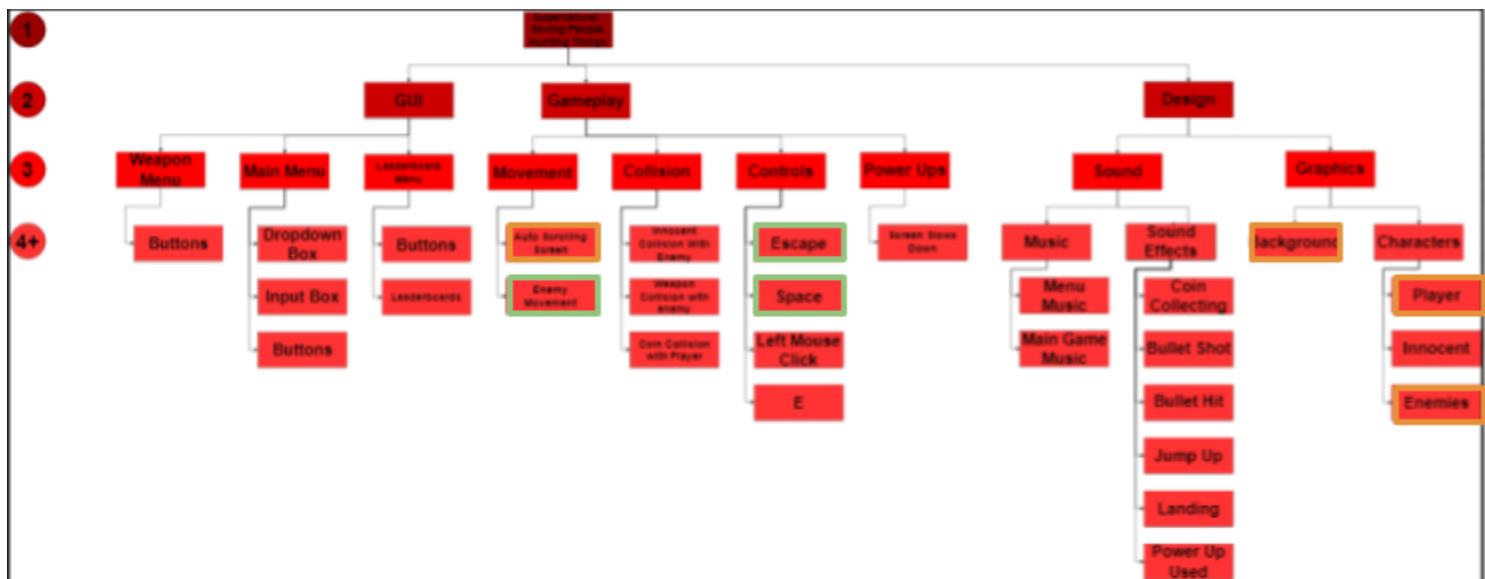
This image is now what the game looks like with the enemy also on the screen.

Tests

	Input	Test Type	Expected outcome	Actual Outcome	Comments

Enemy Movement	Loading the game	Valid	The enemy moves from the right of the screen to the left - the x - coordinate decreases	<pre>Enemy x coordinate is 1340 Enemy x coordinate is 1330 Enemy x coordinate is 1320 Enemy x coordinate is 1310 Enemy x coordinate is 1300 Enemy x coordinate is 1290 Enemy x coordinate is 1280 Enemy x coordinate is 1270 Enemy x coordinate is 1260 Enemy x coordinate is 1250 Enemy x coordinate is 1240 Enemy x coordinate is 1230 Enemy x coordinate is 1220 Enemy x coordinate is 1210 Enemy x coordinate is 1200 Enemy x coordinate is 1190 Enemy x coordinate is 1180 Enemy x coordinate is 1170 Enemy x coordinate is 1160 Enemy x coordinate is 1150 Enemy x coordinate is 1140 Enemy x coordinate is 1130 Enemy x coordinate is 1120 Enemy x coordinate is 1110 Enemy x coordinate is 1100 Enemy x coordinate is 1090 Enemy x coordinate is 1080 Enemy x coordinate is 1070 Enemy x coordinate is 1060 Enemy x coordinate is 1050 Enemy x coordinate is 1040 Enemy x coordinate is 1030 Enemy x coordinate is 1020 Enemy x coordinate is 1010 Enemy x coordinate is 1000 Enemy x coordinate is 990 Enemy x coordinate is 980 Enemy x coordinate is 970 Enemy x coordinate is 960</pre>	The enemy moves from the right of the screen to the left so I don't need to edit this feature however, when the enemy goes off screen another one doesn't come along so i'll need to add that in.
----------------	------------------	-------	---	---	---

Top Down Modular Update



Success Criteria

- Auto scrolling screen

- Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.
 - A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.

- Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power-ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision
 - Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
 - Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
 - Silver Bullet
 - Kills Werewolves upon collision
 - Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Bone
 - Kills Leviathans upon collision
 - Borax
 - Damages Leviathans (kills them eventually) upon collision

- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 5 - Health

Most games have something that will make the game end whether it's completing a level or dying. My game is no different so the next feature I am going to add is the players health. When the health is zero the game is over. To add health into my game I am going to add another attribute into the sprite class for health and I'm going to create another class for the health bar.

```
self.health = 100
```

I have added this line of code to the sprite class with all the other attributes and sets the health of the player to 100 when the character is first created so that we can decrease the health but it will reset when we reset the game.

```
class Bar(pygame.sprite.Sprite):

    def __init__(self,x,y,a,b,colour):
        super().__init__()
        self.image = pygame.Surface([a,b]) # Sets the size
        self.image.fill(colour)
        self.rect = self.image.get_rect() # Hitbox
        self.rect.x = x# Sets the position on the screen
        self.rect.y = y

    def update(self):
        self.image = pygame.Surface([Player.health,10])
        self.image.fill(Green)
```

So that the user can visually see what the health of their character is , I have created a class for the health bar. I haven't set the colour of the health bar with all the attributes

because there will be two different bars to create the health bar. One will be red and the other one will be green . The health bar has an update method that redraws the health bar to be the width of the player's health and it will get filled in green.

```
Player = Sprite(infoObject.current_w/4, (infoObject.cu  
Enemy = Enemy(infoObject.current_w, (infoObject.curren  
HealthBarPt1 = Bar(10,10,100,10,Red)  
HealthBarPt2 = Bar(10,10,Player.health,10,Green)  
  
SpriteGroup = pygame.sprite.Group() # Create a group  
EnemyGroup = pygame.sprite.Group()  
HealthBarGroup = pygame.sprite.Group()  
  
SpriteGroup.add(Player) # Add object to group  
EnemyGroup.add(Enemy)  
HealthBarGroup.add(HealthBarPt1)  
HealthBarGroup.add(HealthBarPt2)
```

In this section of code the lines that were added were lines 3, 4, 8, 11 and 12. These lines create the health bar objects and a health bar group. It then adds the two health bar objects into the health bar group.

```
for eachEnemy in EnemyGroup:  
    Enemyhits = pygame.sprite.spritecollide(Enemy,SpriteGroup, True)  
    if eachEnemy.rect.x<0:  
        EnemyGroup.remove(eachEnemy)  
    if Enemyhits:  
        Player.health= 0
```

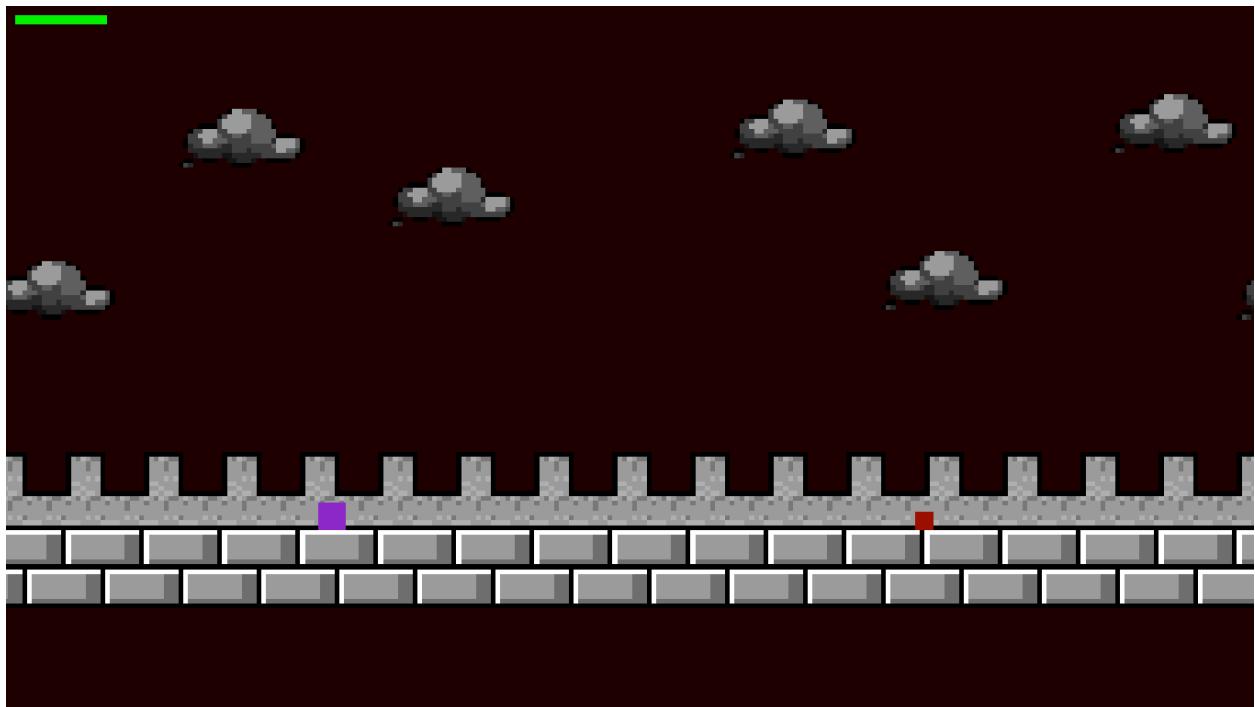
This section of code has been added into the game loop. It checks to see if each enemy collides with the sprite group and if it does the health of the player is set to zero and the sprite is deleted off the screen. It also removes the enemy object when it goes off the screen.

```
    HealthBarGroup.draw(Screen)  
    HealthBarPt2.update()
```

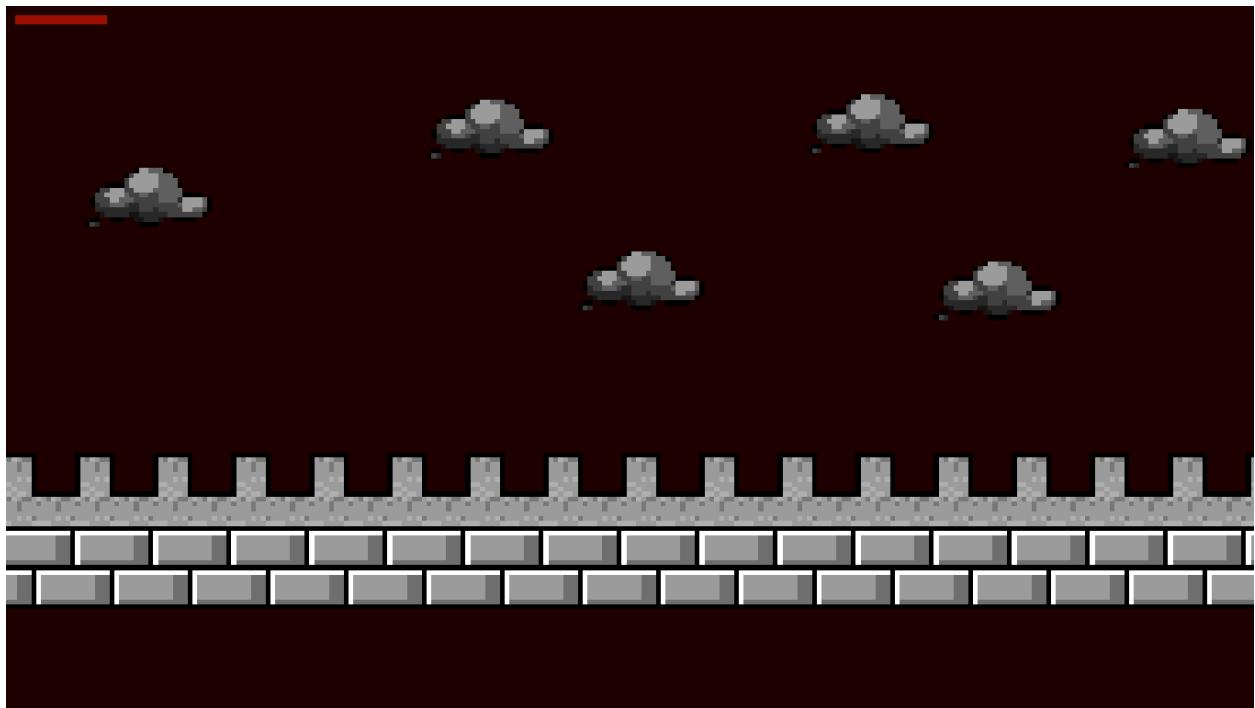
These two lines are further down the game loop. It draws the health bar on the screen and updates the second one so that when the health goes down the width of the green bar gets smaller so the player can see how much health they have left.

```
    if Player.health == 0:  
        SpriteGroup.remove()  
        EnemyGroup.remove(eachEnemy)
```

This if statement is at the very end of the game loop. It checks to see if the health of the player is equal to zero and if it is then it will remove the sprite group and the enemy group off the screen to signal that it is game over.



This is an image of what the game looks like when the player hasn't collided with the enemy. In the top left corner the health bar is still full.



This is an image of what the game looks like when the player has collided with the enemy. In the top left corner there is no more green on the health bar because the player is dead and the player and enemy have been removed off the screen

Tests

	Input	Test Type	Expected Outcome	Actual Outcome	Comments
Player Collision with enemy	Player coordinates +30 >= enemy coordinates	Valid	Player Health decreases to 0 and the player object is deleted off the screen.	<pre> Player x coordinate: 340 Enemy x coordinate: 350 Player health: 0 </pre>	This feature works because the health went to zero when the player and enemy collided
	Player coordinates + 30 <= enemy coordinates	Invalid	Doesn't have any effect on the game , the game carries on as normal.	<pre> Player x coordinate: 340 Enemy x coordinate: 440 Player health: 100 Player x coordinate: 340 Enemy x coordinate: 430 Player health: 100 Player x coordinate: 340 Enemy x coordinate: 420 Player health: 100 </pre>	This works because when the enemy hasn't collided the health of the player remains at 100

	Input	Test Type	Expected Outcome	Actual Outcome	Comments
Health bar	Player health is 100	Valid	The health bar should be the width of the players health making it all green	<pre> Health bar width <Surface(100x10x3)> Player health: 100 </pre>	The width of the health bar is the same as the player health so this works fine.
	Player health gets to zero	Valid	The health bar should be the width of the player health therefore it should all be red	<pre> Health bar width <Surface(0x10x32 SW)> Player health: 0 </pre>	The width of the health bar is the same as the player health so this works fine.

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns

- A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.
 - A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power - ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision

- Witch Bullet
 - Kills Witches upon collision
- Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
- Silver Bullet
 - Kills Werewolves upon collision
- Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
- Bone
 - Kills Leviathans upon collision
- Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 6 - Respawning Enemies

So far in my game you have an enemy that goes across the screen however, once it's gone it doesn't come back. This means the game is over too quickly so therefore in this version I am respawning the enemies when there are no more. To do this I'm adding an enemy count.

```
| EnemyCount = 1
```

This line has been added to the list of variables at the start of the game and we are going to use this variable to control the number of enemies on the screen.

```
Player = Sprite(infoObject.current_w/4, (infoObject.current_h*(107/144))-30, 30, 30)
HealthBarPt1 = Bar(10,10,100,10,Red)
HealthBarPt2 = Bar(10,10,Player.health,10,Green)
```

I used to create the enemy object within this section of code before but I have deleted that line and now I am going to make it within the game loop because it needs to be within the loop for the enemies to keep respawning.

```
if EnemyCount>0:
    enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-20, 20, 20, 10)
    EnemyGroup.add(enemy)
    EnemyCount -=1
```

This section of code has been added into the game loop. It checks to see if EnemyCount is larger than zero and if it is it will create an enemy object and add it to the enemy group. It will then take one off the enemy count so then you don't get an endless number of enemies spawning on the screen.

```
for eachEnemy in EnemyGroup:
    Enemyhits = pygame.sprite.spritecollide(enemy,SpriteGroup, True)
    if eachEnemy.rect.x<0:
        EnemyGroup.remove(eachEnemy)
        EnemyCount+=1
    if Enemyhits:
        Player.health= 0
```

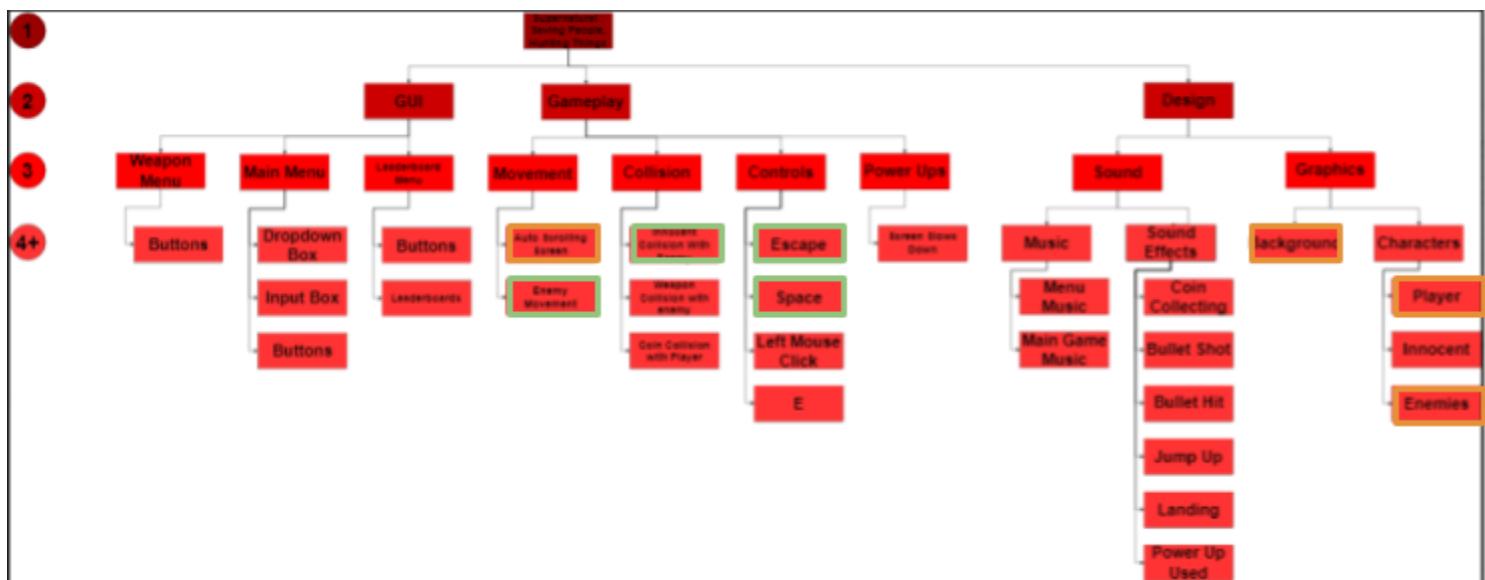
Line 5 has been added to this for loop. This line adds one to EnemyCount when the enemy goes off the other side of the screen. This means that the if statement above will run again therefore adding another enemy to the screen, which means that the enemies will keep coming onto the screen making the game last longer. This will cause the game to be more enjoyable because it doesn't end the game after one enemy.

Tests

	Input	Test Type	Expected Outcome	Actual Outcome	Comments

Enemies Respawning	Enemy coordinates < 0	Valid	An enemy should come from the right side of the screen.	<pre>Enemy x coordinate: 0 EnemyCount: 0 Enemy x coordinate: -10 EnemyCount: 0 Enemy x coordinate: -10 EnemyCount: 1 Enemy x coordinate: 1350 EnemyCount: 0</pre>	This feature worked well because when the enemy went off screen EnemyCount went up causing another enemy to appear on the right and EnemyCount to go back to zero again.
--------------------	-----------------------	-------	---	---	--

Top Down Modular Update



I haven't boxed off anything on my top down modular chart because I didn't have enemies respawning on there but thought it was a crucial feature to have so I added it in.

Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed

- Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.
 - A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.

- Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power - ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision
 - Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
 - Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
 - Silver Bullet
 - Kills Werewolves upon collision
 - Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Bone
 - Kills Leviathans upon collision
 - Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy

- Decrease by 10 when wrong bullet hits enemy
- Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 7 - Bullets

What use is it to have enemies when you can't kill them? All you can do is jump over them. In this version I am going to add bullets so that the player can kill the enemy. To do this I am going to add a weapon class that will let you make lots of different weapons from it.

```
class Bullet(pygame.sprite.Sprite):

    def __init__(self,x,y,a,b,color):
        super().__init__()
        self.image = pygame.Surface([a,b]) # Sets the size
        self.image.fill(color)
        self.rect = self.image.get_rect() # Hitbox
        self.rect.x = x# Sets the position on the screen
        self.rect.y = y

    def update(self):
        self.rect.x += 5
```

The block of code is creating a bullet class that all of the weapon objects will be created from. This class contains an update method that will cause the x coordinate of the weapon to increase by 5 every game loop

```
BulletGroup = pygame.sprite.Group()
```

This line is added to where all the other groups are created and this line creates the group for the bullets that we will make within the game loop.

```

        .
        .
if event.type == pygame.KEYDOWN:
    if event.key == pygame.K_ESCAPE: # Quit if ESC key pressed
        gameExit = True
        pygame.quit()
        quit()
    if event.key == pygame.K_e:
        pressed = True

if event.type == pygame.KEYUP:
    pressed = False

```

This section of code is within the game loop and I have added in lines 6 - 10. It checks to see if the letter e has been pressed and if it has it sets the value of the variable **pressed** to True. It also checks to see if the key has been lifted and if it has the variable **pressed** is set to False. The **pressed** variable is what we are using to determine when a bullet object is going to be created.

```

if pressed == True:
    bullet = Bullet(Player.rect.centerx, Player.rect.centery, 20, 5, Red)
    BulletGroup.add(bullet) # Add bullet to bullet group

for eachBullet in BulletGroup:
    if eachBullet.rect.x > infoObject.current_w:
        BulletGroup.remove(eachBullet) #Kills the bullet if it goes off screen
    BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
    if BulletHits:
        BulletGroup.remove(eachBullet)
        EnemyCount += 1

```

This section of code is within the game loop. If pressed is equal to true which only happens when e is pressed, a bullet object is going to be created from the bullet class and this object will be added to the bullet group. It then iterates through every bullet within the bullet group checking to see if it goes off screen and if it does then it will remove that bullet object. After that it checks to see if the bullet has collided with the enemy and if it has the bullet is removed and then enemy count increases by one so that another enemy can spawn.

```

SpriteGroup.draw(Screen)
EnemyGroup.draw(Screen)
HealthBarGroup.draw(Screen)
BulletGroup.draw(Screen)
HealthBarPt2.update()
EnemyGroup.update()
BulletGroup.update()

```

In this section of code which is near the end of the game loop I am drawing the bullet group onto the screen and then updating the bullet group so it moves across the screen.

Tests

	Input	Test Type	Expected	Actual Outcome	Comments
--	-------	-----------	----------	----------------	----------

					This is what should happen when the enemy gets killed so this is working as it should do.
--	--	--	--	--	---

To fix the error with multiple bullets spawning, I am going to remove the pressed variable and I am going to spawn the bullet where the variable **presses** was set to true because it only ran that code once even if the button was held and the variable **pressed** is just going to be removed overall.

```
if event.key == pygame.K_e:  
    bullet = Bullet(Player.rect.centerx, Player.rect.centery, 20, 5, Red)  
    BulletGroup.add(bullet) # Add bullet to bullet group
```

This section of code is within the game loop and it checks to see if the button e is pressed and if e is pressed then the object **bullet** is made from the **Bullet** class.

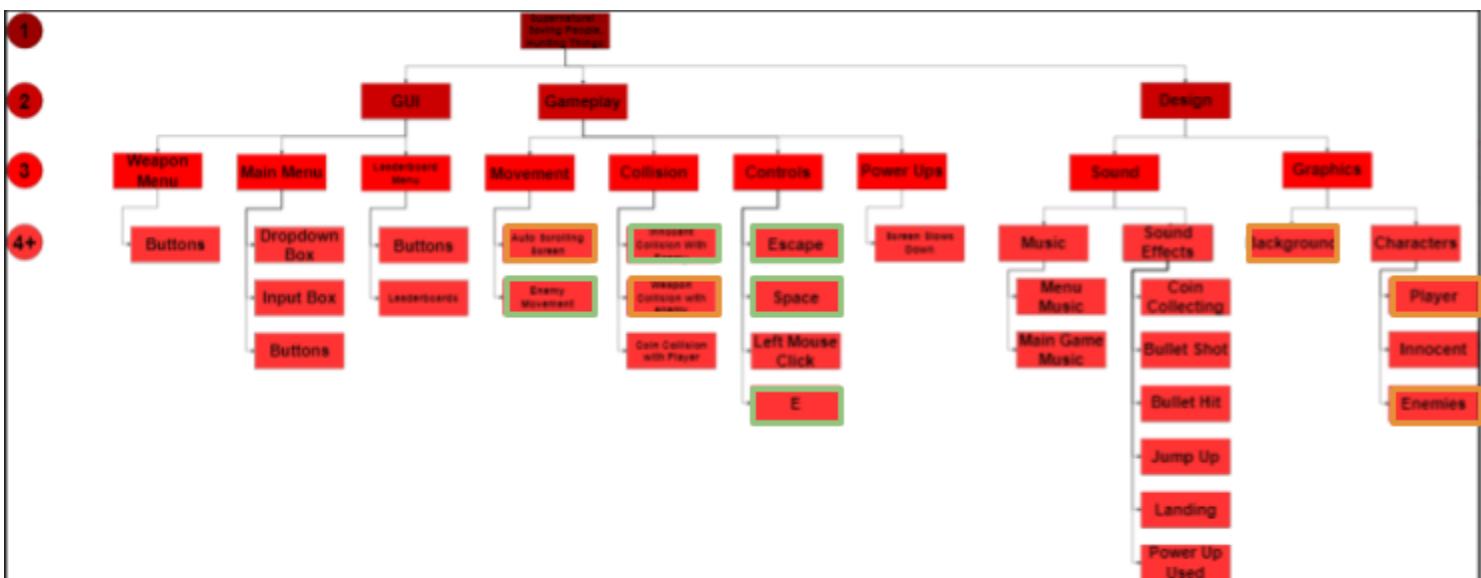
Tests 2

I am going to run the bullets tests again to see if the bullets work now.

	Input	Test Type	Expected Outcome	Actual Outcome	Comments
Bullet movement	E button is pressed	Valid	One bullet should spawn on the screen	E pressed Bullet Spawner	This feature now works because when e is pressed only one bullet spawns on the screen.
	Bullet on Screen	Valid	The x coordinate of the bullet increases	Bullet X coordinate: 375 Bullet X coordinate: 380 Bullet X coordinate: 385 Bullet X coordinate: 390 Bullet X coordinate: 395 Bullet X coordinate: 400 Bullet X coordinate: 405 Bullet X coordinate: 410 Bullet X coordinate: 415 Bullet X coordinate: 420 Bullet X coordinate: 425 Bullet X coordinate: 430 Bullet X coordinate: 435 Bullet X coordinate: 440 Bullet X coordinate: 445 Bullet X coordinate: 450 Bullet X coordinate: 455 " " " " " "	The bullet moves across the screen as expected.

Bullet Hit	Bullet coordinates == Enemy coordinates	Valid	Bullets disappear and enemy disappears	<pre>Enemy x coordinate: 650 Bullet X coordinate: 640 Enemy died</pre>	The bullet hits the enemy so another enemy spawns and the bullet is removed. This is what should happen when the enemy gets killed so this is working as it should do.
------------	---	-------	--	--	--

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar

- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.
 - A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power - ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster

- Different Weapons

- Scythe
 - Kills Death upon collision
- Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
- Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
- Witch Bullet
 - Kills Witches upon collision
- Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
- Silver Bullet
 - Kills Werewolves upon collision
- Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
- Bone
 - Kills Leviathans upon collision
- Borax
 - Damages Leviathans (kills them eventually) upon collision

- Score

- Increases by 100 when enemy killed
- Increase by 10 when player jumps over enemy
- Decrease by 10 when wrong bullet hits enemy
- Increases by 10 when some damage taken off enemy

- Sound

- Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound

- will play and when the player and enemy collide a player died sound will play
- Main game Music - When the player is playing the main game the main game music will play on loop in the background.
- Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 8 - Random Enemy

One of the main features of my game is that there are going to be multiple different types of enemies that can only be killed by certain weapons so the next thing that I am going to add into my game is multiple different enemies. To do this I am going to use a random number generator and have it generate a random number between 1 and the max number of different enemies each difficulty is going to have

```
import random
```

This line is added at the start of the program and it imports random so we can use a random number generator in order to make which enemy appears on the screen random.

```
class Enemy(pygame.sprite.Sprite):

    def __init__(self,x,y,a,b,enemyspeed, image):
        super().__init__()
        self.image = pygame.image.load(image)
        self.image = pygame.transform.scale(self.image, [a,b]) # Sets the size
        self.rect = self.image.get_rect() # Hitbox

        self.rect.x = x# Sets the position on the screen
        self.rect.y = y
        self.speed = enemyspeed

    def update(self,):
        self.rect.x-=self.speed
```

I have edited the enemy class so it takes in an image so that each enemy looks different and the player can visually see what the enemy is.. The image is transformed to the size of the enemy. Each enemy has a different size.

```
* RESETTING OUR VARIABLES

if EnemyCount>0:
    enemyChoice = random.randint(1,2)
    if enemyChoice == 1:
        enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-50,30,50,10, "spr_ghost.png")
    if enemyChoice == 2:
        enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-70,30,70,10, "spr_death.png")

    EnemyGroup.add(enemy)
    EnemyCount -=1
```

This if statement has been edited so that if there are no enemies on the screen a random number is generated between 1 and 2(because I am just working on easy mode at the moment). Depending on what number is generated a ghost object is

created or a death object is created. If a 1 is generated it will be a ghost and if a 2 is generated it will be death.

Tests

I didn't include these tests in my design section but I still think it is important to carry them out.

	Input	Test Type	Expected Outcome	Actual Outcome	Comments
Random Enemy	No enemies on screen so a one is generated	Valid	A ghost appears on the screen	<pre>Random number generated: 1 Ghost</pre> <pre>Random number generated: 2 Death</pre>	This is working as expected because when a one was generated the ghost was spawned on the screen
	No enemies on screen so a two is generated	Valid	Death appears on screen	<pre>Random number generated: 1 Ghost</pre> <pre>Random number generated: 2 Death</pre>	This is working as expected because when a two was generated death spawned on the screen.

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.

- A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power-ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision

- Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
- Silver Bullet
 - Kills Werewolves upon collision
- Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
- Bone
 - Kills Leviathans upon collision
- Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 9 - Different Weapons

In the television show “Supernatural” different weapons kill the different monsters so I wanted to implement this into my game to make it more like the show. To do this I am

going to use the number keys and each number is going to select a different weapon. I will use the bullet class that I have already created for the different weapons.

```
playerweapon = "Scythe"
weaponcolour = (0,0,0)
```

These two variables are created at the start of the program. These are the default weapon type and colour of weapon so that if the user hasn't selected a weapon there won't be any errors.

```
weapon_list = pygame.image.load("spr_weapons.png")
weapon_list = pygame.transform.scale(weapon_list, (infoObject.current_w, (infoObject.current_h/489)*21))
```

These two lines are written just before all the classes are created and they load an image that has all the weapons on it and what number key they correspond to so the users don't have to memorise it. The image is then transformed to fit the width of the screen and then some maths was done for the height of the image to keep the same scaling of the image.

```
class Enemy(pygame.sprite.Sprite):

    def __init__(self,x,y,a,b,enemyspeed, image, monster):
        super().__init__()
        self.image = pygame.image.load(image)
        self.image = pygame.transform.scale(self.image, [a,b]) # Sets the size
        self.rect = self.image.get_rect() # Hitbox

        self.rect.x = x# Sets the position on the screen
        self.rect.y = y
        self.speed = enemyspeed
        self.monster = monster
```

I added a line into the enemy class that says what monster each enemy is. This is so I can compare the monster type with the weapon type so then only certain weapons will damage certain enemies. The monster is passed up as a parameter when the object is created.

```
class Bullet(pygame.sprite.Sprite):

    def __init__(self,x,y,a,b,color, weapon):
        super().__init__()
        self.image = pygame.Surface([a,b]) # Sets the size
        self.image.fill(color)
        self.rect = self.image.get_rect() # Hitbox
        self.rect.x = x# Sets the position on the screen
        self.rect.y = y

        self.weapon = weapon

    def update(self):
        self.rect.x += 5
```

I added a line into the weapon class that says what type of weapon each object is. This is so I can compare the monster type with the weapon type so then only certain weapons will damage certain enemies. The weapon type is passed up as a parameter when the weapon object is created.

```

if event.type == pygame.KEYDOWN:
    if event.key == pygame.K_ESCAPE: # Quit if ESC key pressed
        gameExit = True
        pygame.quit()
        quit()
    if event.key == pygame.K_e:
        bullet = Bullet(Player.rect.centerx, Player.rect.centery, 20, 5, weaponcolour, playerweapon)
        BulletGroup.add(bullet) # Add bullet to bullet group
    if event.key == pygame.K_1:
        playerweapon = "Scythe"
        weaponcolour = Black
    if event.key == pygame.K_2:
        playerweapon = "Salt"
        weaponcolour = White
    if event.key == pygame.K_3:
        weapon = "Machette"
    if event.key == pygame.K_4:
        weapon = "WitchBullet"
    if event.key == pygame.K_5:
        weapon = "AngelBlade"
    if event.key == pygame.K_6:
        weapon = "DemonBlade"
    if event.key == pygame.K_7:
        weapon = "SilverBullet"
    if event.key == pygame.K_8:
        weapon = "Colt"
    if event.key == pygame.K_9:
        weapon = "Bone"
    if event.key == pygame.K_0:
        weapon = "Borax"

```

This series of if statements was added into the game loop. It checks to see if a number key has been pressed and if it has then it will change the variables **playerweapon** and **weaponcolour** to the correct values. When the bullet objects are then created the variables are passed up as parameters.

```

for eachBullet in BulletGroup:
    if eachBullet.rect.x > infoObject.current_w:
        BulletGroup.remove(eachBullet) #Kills the bullet if it goes off screen
    if eachBullet.weapon == "Scythe" and enemy.monster == "death":
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
        if BulletHits:
            BulletGroup.remove(eachBullet)
            EnemyCount += 1
    if eachBullet.weapon == "Salt" and enemy.monster == "spirit":
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
        if BulletHits:
            BulletGroup.remove(eachBullet)
            EnemyCount += 1

```

This for loop is within the game loop and I edited it so that it compares the weapon type with the monster type and if the correct weapon type is used to kill the monster then it will run the collision detection and kill the enemy, remove the bullet off the screen and increase **EnemyCount** by one so that another enemy can spawn on the screen.

```
Screen.blit(weapon_list, (l, infoObject.current_h - ((infoObject.current_h/489)*21)))
```

This line is added near the end of the game loop and it draws on the image of the list of weapons that we created at the start of the program.

Tests

	Input	Test Type	Expected Outcome	Actual Outcome	Comments
Weapon Selected	One is pressed	Valid	Scythe is selected	One pressed - Scythe selected	This feature works as expected so we can test the scythe with the enemies.
	Two is pressed	Valid	Salt is selected	Two pressed - Salt selected	This feature works as expected so we can test the salt with the enemies.
Scythe Collision with enemy	Scythe Coordinates == Death Coordinates	Valid	Death health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre> death x coordinates 586 Scythe x coordinate 517 death x coordinates 576 Scythe x coordinate 522 death x coordinates 566 Scythe x coordinate 527 death x coordinates 556 Scythe x coordinate 532 death x coordinates 546 Scythe x coordinate 537 death x coordinates 546 death x coordinates 1356 death x coordinates 1346 death x coordinates 1336 ... </pre>	This feature works as expected because as the scythe and death collided the scythe was removed and the enemy was killed so a new one spawned on the right of the screen.

	Scythe Coordinates == Spirits coordinates	Invalid	The weapon disappears off the screen.	<pre>Scythe x coordinate 492 spirit x coordinates 596 Scythe x coordinate 497 spirit x coordinates 586 Scythe x coordinate 502 spirit x coordinates 576 Scythe x coordinate 507 spirit x coordinates 566 Scythe x coordinate 512 spirit x coordinates 556 Scythe x coordinate 517 spirit x coordinates 546 Scythe x coordinate 522 spirit x coordinates 536 Scythe x coordinate 527 spirit x coordinates 526 Scythe x coordinate 532 spirit x coordinates 516 Scythe x coordinate 537</pre>	This feature doesn't work at the moment.
Salt Collision with enemy	Salt Coordinates == Death Coordinates	Invalid	The weapon disappears off the screen.	<pre>Salt x coordinate 457 death x coordinates 466 Salt x coordinate 462 death x coordinates 456 Salt x coordinate 467 death x coordinates 446 Salt x coordinate 472 death x coordinates 436</pre>	This feature doesn't work at the moment.
	Salt Coordinates == Spirit Coordinates	Valid	Spirit health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre>Salt x coordinate 527 spirit x coordinates 576 Salt x coordinate 532 spirit x coordinates 566 Salt x coordinate 537 spirit x coordinates 556 Salt x coordinate 542 spirit x coordinates 556 death x coordinates 1356 death x coordinates 1346 death x coordinates 1336 death x coordinates 1326</pre>	This feature works as expected because as the salt and spirit collided the salt was removed and the enemy was killed so a new one spawned on the right of the screen.

The bullets didn't work exactly how I wanted them to so when I looked back at the code it's because there is no collision detection if the weapon doesn't match the monster so I am going to change the two if statements into one if statement that has elifs and then I am going to add an else that does collision detection for any weapons that don't kill the monster.

```

for eachBullet in BulletGroup:
    if eachBullet.rect.x == infoObject.current_w:
        BulletGroup.remove(eachBullet) #Kills the bullet if it goes off screen
    if eachBullet.weapon == "Scythe" and enemy.monster == "death":
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
        if BulletHits:
            EnemyCount += 1
    elif eachBullet.weapon == "Salt" and enemy.monster == "spirit":
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
        if BulletHits:
            EnemyCount += 1
    else:
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, False) #Allows the bullets to collide
    if BulletHits:
        BulletGroup.remove(eachBullet)

```

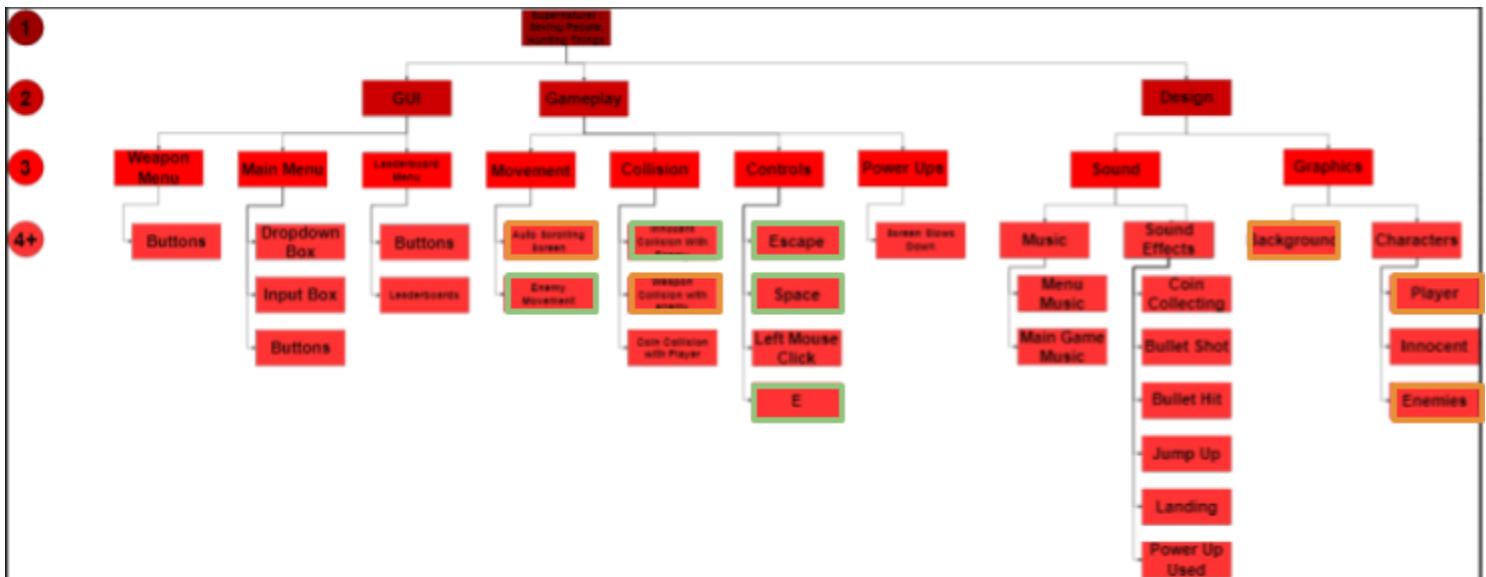
I changed the two if statements into one if statement and then added an else that checks for a collision with any other monster and weapon combinations but this time it says false because I don't want the enemy getting deleted off the screen because the else section is for the weapons that have no effect on the monster. Instead of having the bullet removed in every single section that checks for collision I added an if statement to the end that checks to see if a collision has occurred and if it has then the bullet is removed off the screen. This will make my code more efficient overall because less lines are being used.

Tests 2

	Input	Test Type	Expected Outcome	Actual Outcome	Comments
Scythe Collision with enemy	Scythe Coordinates == Spirits coordinates	Invalid	The weapon disappears off the screen.	Scythe x coordinates 570 spirit x coordinate 630 Scythe x coordinates 575 spirit x coordinate 620 Scythe x coordinates 580 spirit x coordinate 610 Scythe x coordinates 585 spirit x coordinate 600 spirit x coordinate 590 spirit x coordinate 580 spirit x coordinate 570	This feature works as expected because when the spirit and the scythe passed each other the spirit carried on moving however the scythe was removed off the screen.
Salt Collision with enemy	Salt Coordinates == Death Coordinates	Invalid	The weapon disappears off the screen.	Salt x coordinates 635 death x coordinate 660 Salt x coordinates 640 death x coordinate 650 death x coordinate 640 death x coordinate 630 death x coordinate 620	This feature works as expected because

					when death and the salt passed each other the death carried on moving however the salt was removed off the screen.
--	--	--	--	--	--

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy

- Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.
 - A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power-ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons

- Scythe
 - Kills Death upon collision
- Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
- Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
- Witch Bullet
 - Kills Witches upon collision
- Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
- Silver Bullet
 - Kills Werewolves upon collision
- Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
- Bone
 - Kills Leviathans upon collision
- Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound

- will play and when the player and enemy collide a player died sound will play
- Main game Music - When the player is playing the main game the main game music will play on loop in the background.
- Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 10 - Score

It's pointless having a game if there is no way for a user to measure how good they are and compare it against other players because they will get bored of the game so in this version I am adding a score to the game. To do this I am going to add a variable called **score** to the start of my game and have it increase when an enemy dies.

```
score = 0
```

This line is at the start of my game and is just creating a variable that I am going to store the score in.

```
font = pygame.font.SysFont(None, 25) # Import the font
ScoreBoardFont = pygame.font.Font("fnt_HelpMe.ttf", 25)
```

These two lines are at the start of the code and they are importing a ttf font and setting the size of the font so that I can display the font onto the screen.

```
def ScoreText(Text, Colour, x, y): # Text procedure
    ScreenText = ScoreBoardFont.render(Text, True, Colour)
    Screen.blit(ScreenText, [x, y])
```

This procedure is stored just below the classes and it is the procedure I am going to use to display the users score on the screen. This procedure renders the text and displays it at the coordinates passed up.

```
for eachEnemy in EnemyGroup:
    Enemyhits = pygame.sprite.spritecollide(eachEnemy, SpriteGroup, True)
    if eachEnemy.rect.x<0:
        EnemyGroup.remove(eachEnemy)
        EnemyCount+=1
    if eachEnemy.rect.centerx == Player.rect.centerx:
        score +=10
        Player.health-=10
    if Enemyhits:
        Player.health= 0
```

I increased the score here when the player jumps over the enemy so that if the only thing the player does is jump over the enemy then the player can still have a score.

```

for eachBullet in BulletGroup:
    if eachBullet.rect.x == infoObject.current_w:
        BulletGroup.remove(eachBullet) #Kills the bullet if it goes off screen
    if eachBullet.weapon == "Scythe" and enemy.monster == "death":
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
        if BulletHits:
            EnemyCount += 1
            score+=100
    elif eachBullet.weapon == "Salt" and enemy.monster == "spirit":
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
        if BulletHits:
            EnemyCount += 1
            score+=100
    else:
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, False)
        if BulletHits:
            score-=10
    if BulletHits:
        BulletGroup.remove(eachBullet)

```

In this section of code the score increases by 100 if the enemy is killed and if the player shoots the wrong bullet then the score decreases by ten. This is so that the player doesn't spam a load of random bullets because if they did the game could go on forever if they alternate between the bullets.

```

ScoreText("Score:",White,10,30)
ScoreText(str(score),White,110,30)

```

These lines of text are at the end of the game loop and it uses the score text procedure to display the user's score on the screen.

Tests

	Input	Test Type	Expected Outcome	Actual Outcome	Comments
	Enemy is removed off screen	Valid	The players score increases by 100	<pre> spirit x coordinator 610 score 0 spirit x coordinator 600 score 0 spirit x coordinator 600 score 100 death x coordinator 1350 score 100 death x coordinator 1340 score 100 </pre>	When the spirit was killed the score increased by 100 and a new enemy spawned so this feature works as expected.

	Player goes past enemy but they never collide	Valid	The players score increases by 10	<pre> death x coordinater 340 player x coordinate 340 Score 0 death x coordinater 330 player x coordinate 340 Score 10 death x coordinater 320 player x coordinate 340 Score 10 </pre>	When the player jumped over the enemy the score decreased by 10 so this feature works as expected.
	Bullet hits enemy but enemy isn't removed	Valid	The players score decreases by 10	<pre> Bullet x coordinate 590 death x coordinater 620 Score 0 Bullet x coordinate 595 death x coordinater 610 Score 0 Bullet x coordinate 600 death x coordinater 600 Score -10 death x coordinater 590 Score -10 </pre>	When the bullet hit the enemy it was removed off screen but the enemy stayed so the score decreased by 10. This feature works as expected.

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.

- A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power-ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision

- Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
- Silver Bullet
 - Kills Werewolves upon collision
- Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
- Bone
 - Kills Leviathans upon collision
- Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 11 - Menu

Like every game, my game is going to have a menu that has buttons that allow the user to click on what they want to view. The buttons are going to be **Play**, **How to Play** and

Leaderboards a well as an exit button in the bottom right. To do this I am going to import a python module called **guizero**.

```
from guizero import App, Box, Text, TextBox, PushButton, Picture
```

This line is added to the very start of the program and it imports guizero and all the guizero modules that we are going to use for our menu.

```
*****#
def Game():
    i = 0
    Jump = False
    jumpCount = 20
    score = 0
    EnemyCount = 1
    playerweapon = "Scythe"
    weaponcolour = (0,0,0)

    gameExit = False
    pressed = False
    while not gameExit:
```

I have created a procedure called game and I have stored the game loop inside it and also I have created the variables within the game loop instead of at the start so that if the user plays the game multiple times in a row then the variables will reset too.

```
if Player.health == 0:
    SpriteGroup.remove()
    EnemyGroup.remove(eachEnemy)
    gameExit = True
    Menu.show()
    pygame.display.update()
```

At the end of the game loop, I added the **Menu.show()** so that when the game is over it shows the main menu.

```

#####
def playGame():
    Menu.hide()
    Game()

Menu = App(title="Supernatural: Saving People Hunting Things", bg = Black)
Menu.full_screen = True

MenuHeight = Menu.height
MenuWidth = Menu.width

TitleBox = Box(Menu, width="fill", align="top")
Background = Picture(Menu, image = "bgd_menu.PNG")

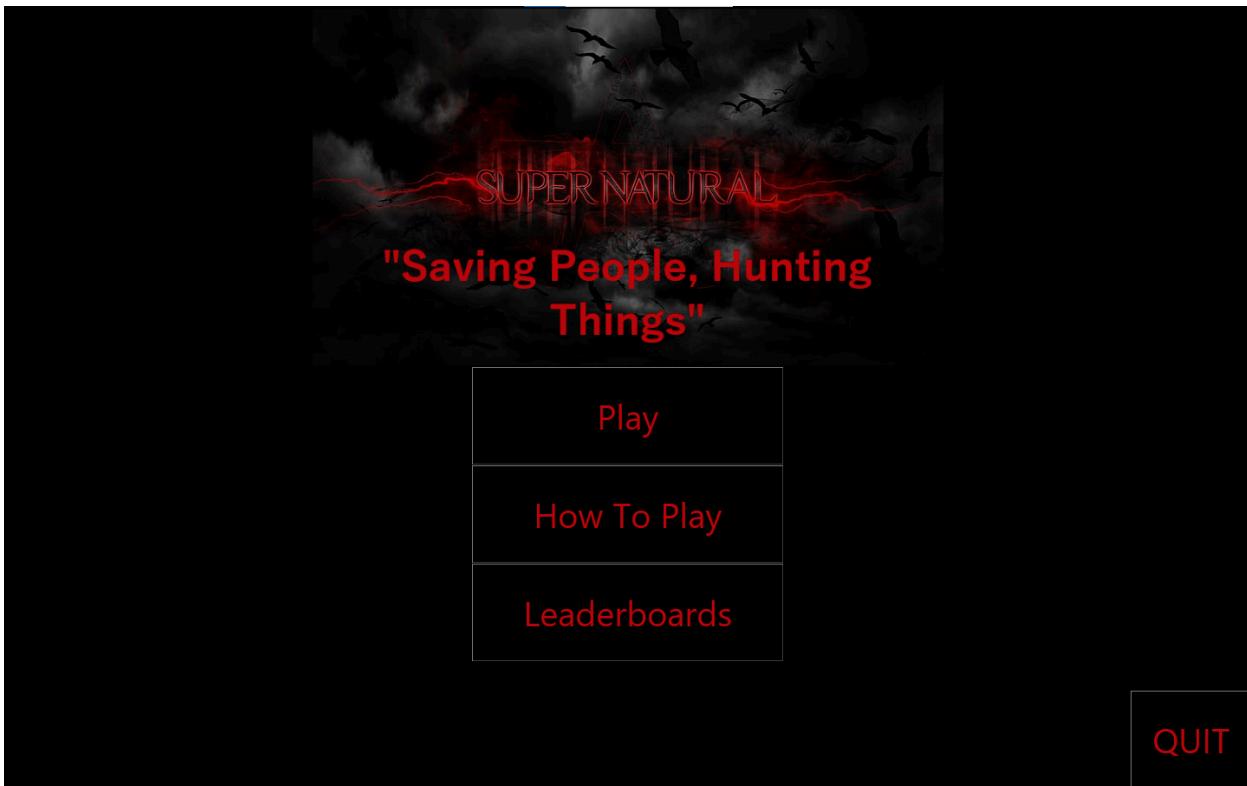
ButtonBox = Box(Menu, width = int(MenuWidth/4), height="fill")
button = PushButton(ButtonBox, text ="Play", width = "fill", command=playGame)
button.text_size = int(MenuHeight/30)
button.text_color = DarkRed2
button = PushButton(ButtonBox, text="How To Play", width = "fill")
button.text_size = int(MenuHeight/30)
button.text_color = DarkRed2
button = PushButton(ButtonBox, text="Leaderboards", width = "fill")
button.text_size = int(MenuHeight/30)
button.text_color = DarkRed2

ExitBox = Box(Menu, align="bottom", width = "fill")
button = PushButton(ExitBox, text ="QUIT", align = "right")
button.text_size = int(MenuHeight/30)
button.text_color = DarkRed2
#####
Menu.display()

```

This section of code has been added at the very end of the program. I have created a procedure called **playGame** and if this procedure is run it runs the **Game** procedure that we created and hides the menu so that we can actually see the game that is being run. The menu is being created with the guizero model **App** and I have set the title of the menu as the title of the screen. The menu has the background set to black and it is then set to full screen. I create variables of the width and height of the menu so that I can use maths to ensure the layout remains the same on screens of different sizes. In the menu there is a title box that is displayed at the top and set to the width of the screen and within that box I have used **Picture** to insert an image that says the title of the game. Below the title box I have created a buttons box that is going to store the buttons in and it is set to the height of the screen. The play button has a command **playGame** which will run the procedure that has this name. All the buttons are created from **PushButton** and stored in **ButtonBox**. The colour of the text is set to dark red and the size of the text is set to one thirtieth of the screen's height so that if the user has a smaller screen the text is smaller and if the user has a larger screen the text is larger. On the bottom of

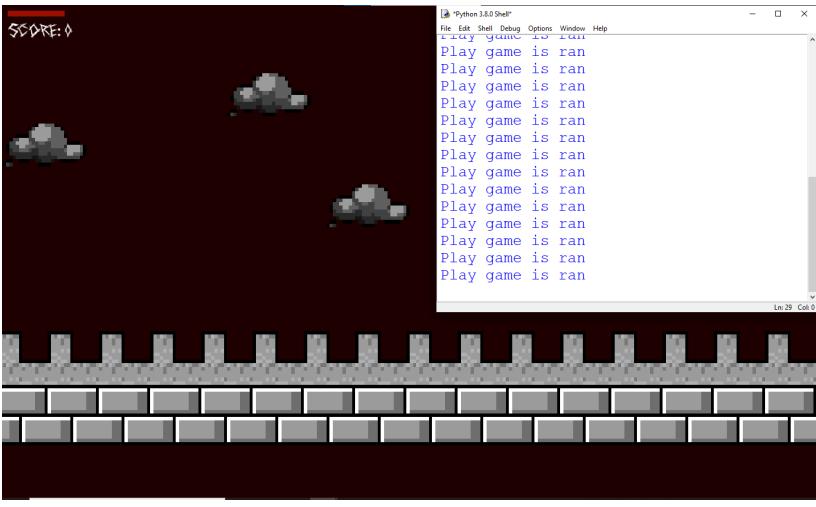
the menu I have added an exit box and in the box is an exit button that is aligned to the right of the screen.

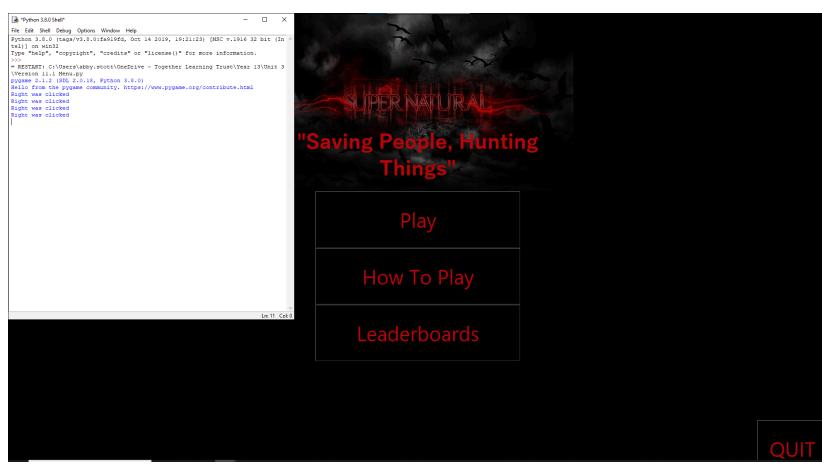


This is an image of how my menu looks when it is displayed on the screen. The menu will always be displayed in full screen.

Tests

Input	Test Type	Expected Outcome	Actual Outcome	Comments
Running the code initially	Valid	The menu should be displayed	<p>A screenshot of a game menu titled "SUPER NATURAL" with a dark, moody background. The menu has three options: "Play", "How To Play", and "Leaderboards". In the bottom right corner is a "QUIT" button. An overlaid Python 3.8.0 Shell window shows the following text:</p> <pre> Python 3.8.0 (tags/v3.8.0:fa919fd, Oct 1 4 2019, 19:21:23) [MSC v.1916 32 bit (In tel)] on win32 Type "help", "copyright", "credits" or " license()" for more information. >>> = RESTART: C:\Users\abby.stott\OneDrive - Together Learning Trust\Year 13\Unit 3 \Version 11.1 Menu.py pygame 2.1.2 (SDL 2.0.18, Python 3.8.0) Hello from the pygame community. https:// /www.pygame.org/contribute.html </pre>	The menu loaded when the code was initially run so therefore this feature works correctly.

Mouse is hovering over the play button and the left button is clicked.	Valid	The game should load	 	<p>This feature works when you first press the play button however when you die and click play again the variables don't reset so you can't actually play the game again so I am going to fix this so that it is possible to replay the game.</p>
Mouse is hovering over the play button and the right button is clicked	Invalid	This shouldn't affect anything on the game because the button hasn't been pressed.	<pre> def rightclicked(): print("Right was clicked") Menu = App(title="Supernatural: Saving People Hunting Thing") Menu.full_screen = True MenuHeight = Menu.height MenuWidth = Menu.width TitleBox = Box(Menu, width="fill", align="top") Background = Picture(Menu, image = "bgd_menu.PNG") ButtonBox = Box(Menu, width = int(MenuWidth/4), height="fill") button = PushButton(ButtonBox, text ="Play", width = "fill") button.text_size = int(MenuHeight/30) button.text_color = DarkRed2 button.when_right_button_pressed = rightclicked </pre>	<p>I added the function right clicked in that only runs when the right button on the mouse is pressed whilst it's hovering over the play button. When the right button was clicked whilst the mouse was hovering over the play button it still stayed on the menu so this feature works as it should.</p>



When play is clicked for a second the time the game doesn't replay. This is because I didn't rewrite the variables within the game loop so they don't restart when someone tries to replay so I am going to fix this by creating the variable within the game loop.

```
def Game():
    i = 0
    Jump = False
    jumpCount = 20
    score = 0
    EnemyCount = 1
    playerweapon = "Scythe"
    weaponcolour = (0,0,0)
    #####
    Player = Sprite(infoObject.current_w/4, (infoObject.current_h*(107/144))-30,30,30)
    HealthBarPtl = Bar(10,10,100,10,Red)
    HealthBarPt2 = Bar(10,10,Player.health,10,Green)

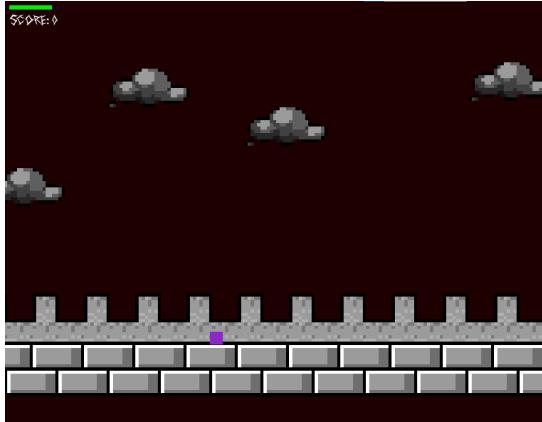
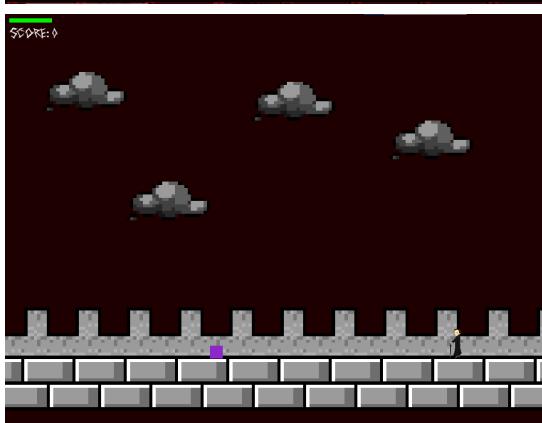
    SpriteGroup = pygame.sprite.Group() # Create a group
    EnemyGroup = pygame.sprite.Group()
    HealthBarGroup = pygame.sprite.Group()
    BulletGroup = pygame.sprite.Group()

    SpriteGroup.add(Player) # Add object to group
    HealthBarGroup.add(HealthBarPtl)
    HealthBarGroup.add(HealthBarPt2)

    ######GAMELOOP#####
```

I created the game's variables and objects within the game procedure so now when play is pressed the player should reset and the user should be able to play the game again.

Tests 2

Mouse is hovering over the play button and the left button is clicked.	Valid	The game should load	 	<p>This feature works when you first press the play button and when you press the play button again it also works so therefore this feature works correctly overall.</p>
--	-------	----------------------	---	--

Top Down Modular Update



Success Criteria

- Auto scrolling screen

- Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.
 - A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.

- Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power-ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision
 - Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
 - Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
 - Silver Bullet
 - Kills Werewolves upon collision
 - Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Bone
 - Kills Leviathans upon collision
 - Borax
 - Damages Leviathans (kills them eventually) upon collision

- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 12 - Customisation Menu

In my game I want the user to pick a few options before they actually start the game like: what hunter they are going to use, what their name is and what difficulty they want to play. I thought the best way to do this would be to create a menu that pops up when the user clicks play instead of overcrowding the main menu because it makes it easier to navigate with the two different menus.

```
from guizero import App, Box, Combo, Text, TextBox, PushButton, Picture, Window, Slider
import random
```

In order to have the different selection tools I had to import combo, window and slider so that the menu will actually work so I added these into the first line of code that imports all of the other guizero modules.

```
def customGame():
    Menu.hide()
    Customiser.show()

Menu = App(title="Supernatural: Saving People Hunting Things", bg = Black)
Menu.full_screen = True

MenuHeight = Menu.height
MenuWidth = Menu.width

TitleBox = Box(Menu, width="fill", align="top")
Background = Picture(Menu, image = "bgd_menu.PNG")

ButtonBox = Box(Menu, width = int(MenuWidth/4), height="fill")
button = PushButton(ButtonBox, text ="Play", width = "fill", command=customGame)
```

When the play button is pressed on the main menu it used to run a procedure called **playGame** however I changed the command for the button to **customGame** and then I

changed the **playGame** procedure into **customGame** and the **customGame** procedure hides the main menu and displays the customisation menu.

```
def GameInformation():
    Name = NameInput.value
    print(Name)
    Customiser.hide()
    Game()

Customiser = Window(Menu, title = "Customisation Menu", bg = Black)
Customiser.full_screen = True

CustomiserHeight = Customiser.height
CustomiserWidth = Customiser.width

TitleBox2 = Box(Customiser, width="fill", align="top")
Background2 = Picture(Customiser, image = "bgd_menu.PNG")

InputsBox = Box(Customiser, width = int(Customiser.width/4), height = "fill")
NameText = Text(InputsBox, text = "Enter your name:")
NameText.text_color = White
NameText.text_size = 20

NameInput = TextBox(InputsBox, width = "fill")
NameInput.text_color = White
NameInput.text_size = 20

DifficultyInput = Slider(InputsBox, start = 1, end = 1, width = "fill")
DifficultyInput.text_color = White
DifficultyInput.text_size = 20

HunterInput = Combo(InputsBox, width = "fill")

StartButton = PushButton(InputsBox, text = "START", width = "fill", command = GameInformation)

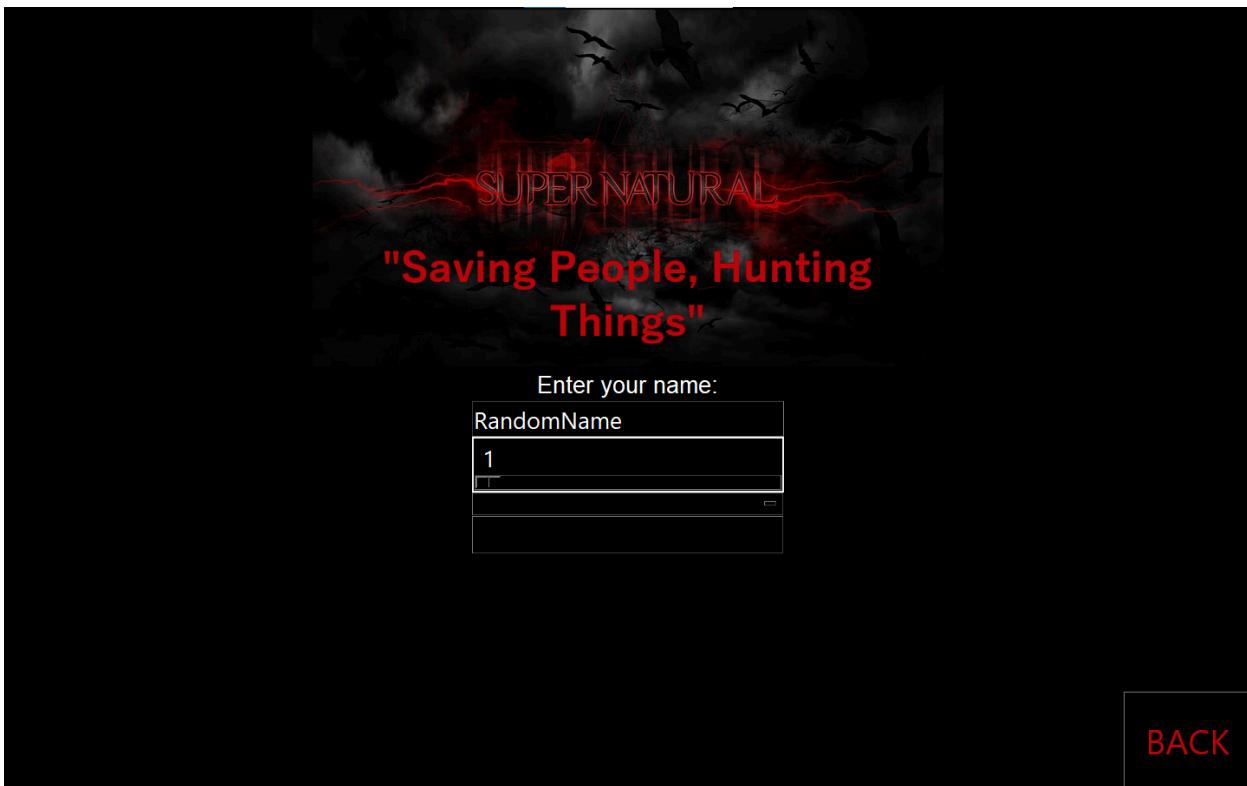
BackBox = Box(Customiser, align="bottom", width = "fill")
button = PushButton(BackBox, text ="BACK", align = "right")
button.text_size = int(MenuHeight/30)
button.text_color = DarkRed2
```

This section of code is what is used to create the customisation menu at the top if a procedure called **GameInformation** that will save the user's name so that when I have my leaderboards working, people can have their score stored on the leaderboard. The procedure also hides the customisation menu and displays the game. The customisation menu is set as a window from the main app. The background is set to black and the size of the screen is set to full screen. I make variables of the width and the height so that I can use maths to make everything different sizes so the game can be played on screens with different resolutions. Displayed at the top of the customisation menu is a title box that has the same image in it as the one displayed on the main menu because it is the name of my game. Below the title box is an inputs box that holds a text box that the user can type their name into, a slider so the user can select their difficulty and a combo that will allow the user to select which hunter they want to play as. A start button is then at the bottom of the inputs box that will run the **GameInformation** procedure that was created just above it. At the very bottom right of

the customisation menu is a back button that will allow the user to go back to the main menu.

```
| Customiser.hide()
```

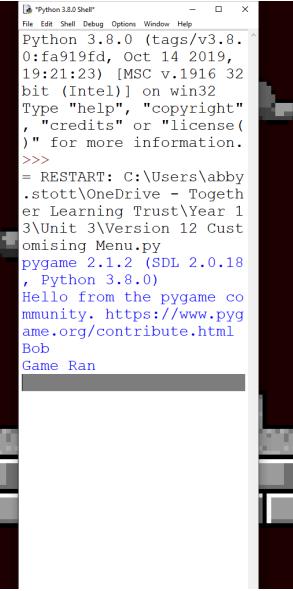
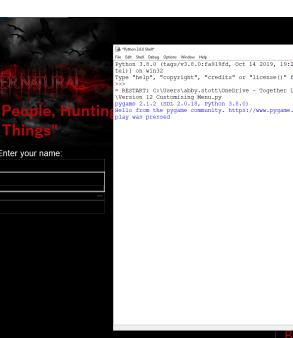
At the very end of the code is this single line that hides the customisation menu until it is displayed in the **customGame** procedure.



This is an image of what the customisation screen looks like.

Tests

	Input	Test Type	Expected Outcome	Actual Outcome	Comments

Customer	The user types a three letter name into the input text box And presses play	Borderline	The game should load			The name bob was used which is three letters and the game still ran as normal so this feature works correctly.
	The user types a ten letter name into the input text box And presses play	Borderline	The game should load			The name Winchester was used which is ten letters and the game still ran as normal so this feature works as normal.
	The user presses the play button on the main menu	Valid	The customisation screen should load			The game customisation menu was displayed when the play button on the main menu was pressed so this feature works as expected.

I haven't added any validation yet so in a future version I will add validation and test different lengths of names again.

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.

- A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power-ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision

- Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
- Silver Bullet
 - Kills Werewolves upon collision
- Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
- Bone
 - Kills Leviathans upon collision
- Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 13 - Fixing Health Bar

Whilst testing my game I discovered that since adding the menu in my players health bar doesn't work so In this version I am going to get my health bar working again.

```

class Bar(pygame.sprite.Sprite):

    def __init__(self,x,y,a,b,colour):
        super().__init__()
        self.image = pygame.Surface([a,b]) # Sets the size
        self.image.fill(colour)
        self.rect = self.image.get_rect() # Hitbox
        self.rect.x = x# Sets the position on the screen
        self.rect.y = y

    def update(self, Health):
        self.image = pygame.Surface([Health,10])
        self.image.fill(Green)

```

In the bar class in the update method, the health is getting passed up as a parameter and the width of the bar is set to the value of the health parameter that was passed up.

```
HealthBarPt2.update(Player.health)
```

I edited the line at the end of the game loop that updates the health bar by passing up the player's health as a parameter.

Tests

	Input	Test Type	Expected Outcome	Actual Outcome	Comments
Health bar	Player health is 100	Valid	The health bar should be the width of the players health making it all green	<code>Health bar width <Surface(100x10x32)> Player health: 100</code>	The width of the health bar is the same as the player health so this works fine.
	Player health gets to zero	Valid	The health bar should be the width of the player health therefore it should all be red	<code>Health bar width <Surface(0x10x32 SW)> Player health: 0</code>	The width of the health bar is the same as the player health so this works fine.

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.

- A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power-ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision

- Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
- Silver Bullet
 - Kills Werewolves upon collision
- Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
- Bone
 - Kills Leviathans upon collision
- Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 14 - Game Over and Validation

It's nice to have a screen pop up saying game over when you have died in a game so in this version I am adding a game over screen. My game also needs some validation checks so I am going to check the length of the name that has been input so that people don't put a name in that is too long or too short and I am going to have a pop up asking the user if they are sure they want to quit the game after they press the quit button and if they say yes another popup will show that thanks the user for playing the game.

```
if Player.health == 0:  
    SpriteGroup.remove()  
    EnemyGroup.remove(eachEnemy)  
    gameExit = True  
    GameOver.show()  
    pygame.display.update()  
.#####
```

At the end of the game loop instead of taking the user back to the main menu I changed the **menu.show()** line to **GameOver.show()** so that the user doesn't get confused as to why they automatically go to the main menu.

```
def endgame():  
    endchoice = Menu.yesno("QUIT?", "Are you sure you want to quit?")  
    if endchoice == True:  
        Menu.warn("Thanks for Playing", "Thank you for playing Supernatural:\\\"Saving People, Hunting Things\\\"")  
        Menu.destroy()  
        pygame.quit()
```

Underneath the **customGame** procedure created in a previous version, I have created an **endgame** function that has a pop up that double checks if the user wants to end the game and if they do there is another pop up that thanks the player for playing the game. And then the menu is destroyed and pygame quits.

```
ExitBox = Box(Menu, align="bottom", width = "fill")  
button = PushButton(ExitBox, text ="QUIT", align = "right", command = endgame)  
button.text_size = int(MenuHeight/30)  
button.text_color = DarkRed2
```

I have added a command to the exit button that runs the **endgame** procedure that I created.

```
.....  
def GameInformation():  
    Name = NameInput.value  
    print(Name)  
    if len(Name) < 3:  
        Customiser.error("Name is too short", "The name you have entered is too short. Your name must be 3 characters - 10 characters long.")  
    elif len(Name) >10:  
        Customiser.error("Name is too long", "The name you have entered is too long. Your name must be 3 characters - 10 characters long.")  
    else:  
        Customiser.hide()  
        Game()
```

Within the **GameInformation** procedure, I have added a length check on the name by using an if statement so that if the name is shorter than three characters there is a pop up that tells the user it is too short and it needs to be between three and ten characters.

It then checks to see if the name is longer than ten characters and if it is there is a popup that tells the user that the name needs to be between three and ten characters. If the name is the right length then it will hide the customisation game and run the game procedure so the user can play the game.

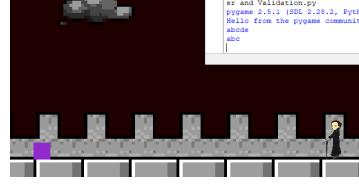
```
def MainMenu():
    GameOver.hide()
    Menu.show()
GameOver = Window(Menu, title = "Game Over", bg = DarkRed)
GameOver.full_screen = True
GameOverText = Text(GameOver, text = "Game Over!", width = "fill", height = "fill")
GameOver.text_size = 100
GameOver.text_color = White
GameOverButton = PushButton(GameOver , text="Go back to Main Menu", command = MainMenu)
```

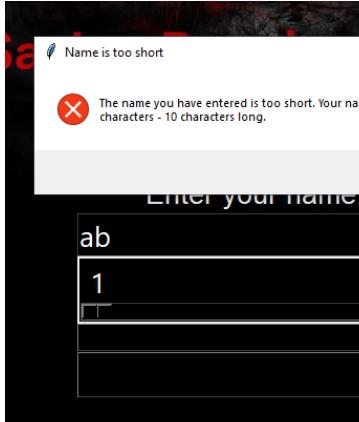
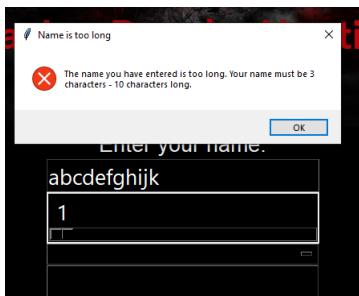
At the bottom of the program, I have added a main menu procedure that will hide the game over screen and show the main menu. Below that I have created the game over window that will display when the user has died in the game. It is displayed in full screen and has the words game over at the top of the screen. Along the bottom of the screen is a **Go back to Main Menu** button that runs the **MainMenu** procedure.

```
GameOver.hide()
```

At the very bottom of the code I hide the **GameOver** window until it is needed.

Tests

The user types a five letter name into the input box and presses play	Valid	The game should load		When I used the name "abcde" the game loaded so this is working how it should.
The user types a three letter name into the input text box And presses play	Borderline	The game should load		When I used the name "abc" the game loaded so this is working how it should.

	The user types a ten letter name into the input text box And presses play	Borderline	The game should load		When I used the name "abcdefghijkl" the game loaded so this is working how it should.
	The user types a two letter name into the input text box And presses play	Invalid	An error should pop up telling the user that the name has to be between 3 and 10 characters.		When a two letter name was entered an error popped up saying the name was too short and that it needed to be between three and ten characters so this feature works as normal.
	The user types an eleven letter name into the input text box And presses play	Invalid	An error should pop up telling the user that the name has to be between 3 and 10 characters.		When an eleven letter name was entered an error popped up saying the name was too long and that it needed to be between three and ten characters so this feature works as normal.

After completing all the tests on the name length it's clear to see that the validation check on the length of the name works correctly.

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.

- A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power - ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision

- Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
- Silver Bullet
 - Kills Werewolves upon collision
- Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
- Bone
 - Kills Leviathans upon collision
- Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 15 - Leaderboard

One of the most important parts of most games is a leaderboard because it keeps people interested because they want to beat others to become the best. In this version I

I am going to create my leaderboard and allow the user to display it onto the screen. To do this I am going to create a leaderboard database with a table that stores the user's scores. I am going to do this by using **DB browser** and I will write and read from the table by using **SQL**.

```
import sqlite3
```

I added this line to the start of my program and it imports sqlite3 so that I can use SQL later on in my program to write the users score to the leaderboard and then i can also use it to read from the leaderboard table and display it onto the screen for the user so then the user can see their score if they have one of the top ten scores for ny of the difficulties.

```
-----  
def Game(Name, Difficulty):  
    i = 0  
    .Timer = False
```

When the game procedure gets called it has the users name and chosen difficulty passed up as parameters so that when the scores get written to the leaderboard you can see who got what score and the difficulty was passed up so then I can have a different leaderboard for each difficulty.

```
def AddToLeaderboard(Name, Score, Difficulty):  
    con = sqlite3.connect('Leaderboard.db')  
    cursorObj = con.cursor()  
  
    entities = (Name, Score, Difficulty)  
    cursorObj.execute('INSERT INTO UsersScores (Name, Score, Difficulty) VALUES(?, ?, ?)', entities)  
  
    con.commit()  
  
    cursorObj.close()
```

This procedure was added to the start of the game loop and it takes the name, difficulty and score in as parameters and then the leaderboard database is opened and the name, score and difficulty are written to the leaderboard table. The database is closed because it isn't needed after it has been written to but it will reopen again when needed.

```
if Player.health == 0:  
    SpriteGroup.remove()  
    EnemyGroup.remove(eachEnemy)  
    AddToLeaderboard(Name, score, Diffi  
    gameExit = True  
    GameOver.show()  
    pygame.display.update()
```

At the end of the game loop, I added in a line that runs the **AddToLeaderboard** procedure passing the name, score and difficulty up as parameters.

```
def ShowLeaderboard():  
    Menu.hide()  
    Leaderboard.show()
```

I created this procedure above the code that creates the main menu and this procedure hides the main menu and displays the leaderboard menu.

```

308| button.text_color = darkgrey
309| button = PushButton(ButtonBox, text="Leaderboards", width = "fill", command = ShowLeaderboard)
310| button.text_size = int(MenuHeight/20)

```

To this line I added **command = ShowLeaderboard** and this will run the **ShowLeaderboard** procedure.

```

def GameInformation():
    Name = NameInput.value
    Difficulty = DifficultyInput.value
    print(Name)
    if len(Name) < 3:
        Customiser.error("Name is too short")
    elif len(Name) > 10:
        Customiser.error("Name is too long")
    else:
        Customiser.hide()
        Game(Name, Difficulty)

```

Within the GameInformation procedure which runs the game, I created a variable called difficulty that takes the value from the difficulty slider. This variable then gets passed up as a parameter to the game procedure so it can be written to the leaderboard.

```

def BackToMain():
    Leaderboard.hide()
    Menu.show()

```

I added this procedure just before creating the leaderboard menu and this procedure hides the leaderboard menu and displays the main menu.

```

con = sqlite3.connect('Leaderboard.db')
cursorObj = con.cursor()

Leaderboard = Window(Menu, title="Supernatural: Saving People Hunting Things", bg = Black)
Leaderboard.full_screen = True

LeaderboardHeight = Leaderboard.height
LeaderboardWidth = Leaderboard.width

TitleBox = Box(Leaderboard, width="fill", align="top")
Background = Picture(TitleBox, image = "Bgd_Leaderboard.PNG", width = LeaderboardWidth)

```

In this section of code, the leaderboard database is opened so that the leaderboards can be displayed to the users screens. The leaderboard menu is then created with a black background and it is set to full screen. A title box is then drawn at the top of the screen and in the title box an image is displayed which is the same one that is displayed for the main menu however it has the words **Easy**, **Medium** and **Hard** written on it to show which leaderboard is for which difficulty.

```

EasyBox = Box(Leaderboard, width = int(LeaderboardWidth/4), height="fill", align = "left")
EasyBoxNumbers = Box(EasyBox, width = int(LeaderboardWidth/4), height="fill", align = "left")
for i in range (1,11):
    PlaceText = Text(EasyBoxNumbers, text = i)
    PlaceText.text_size = int(LeaderboardHeight/30)
    PlaceText.text_color = White
strSQL = "SELECT Name, Score FROM UsersScores WHERE Difficulty = 1 ORDER BY Score DESC LIMIT 10"
cursorObj.execute(strSQL)
rows = cursorObj.fetchall()

EasyBoxStats = Box(EasyBox, width = int(LeaderboardWidth/4), height="fill", align = "left")
for row in rows:
    EasyText = Text(EasyBoxStats, text = row)
    EasyText.text_size = int(LeaderboardHeight/30)
    EasyText.text_color = White

MediumBox = Box(Leaderboard, width = int(LeaderboardWidth/4), height="fill", align = "left")
MediumBoxNumbers = Box(MediumBox, width = int(LeaderboardWidth/4), height="fill", align = "left")
for i in range (1,11):
    PlaceText = Text(MediumBoxNumbers, text = i)
    PlaceText.text_size = int(LeaderboardHeight/30)
    PlaceText.text_color = White

HardBox = Box(Leaderboard, width = int(LeaderboardWidth/4), height="fill", align = "left")
HardBoxNumbers = Box(HardBox, width = int(LeaderboardWidth/4), height="fill", align = "left")
for i in range (1,11):
    PlaceText = Text(HardBoxNumbers, text = i)
    PlaceText.text_size = int(LeaderboardHeight/30)
    PlaceText.text_color = White

ExitBox = Box(Leaderboard, align="bottom", width = "fill")
button = PushButton(ExitBox, text ="QUIT", align = "right", command = BackToMain)
button.text_size = int(LeaderboardHeight/30)
button.text_color = White

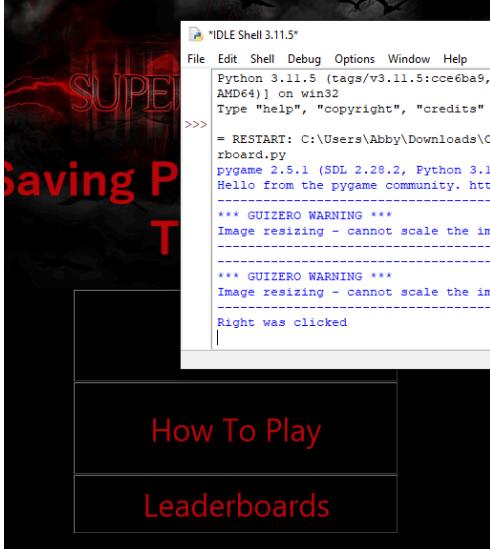
cursorObj.close()
Leaderboard.hide()

```

This section of code is written directly below the code that I have just explained and it makes a box for **Easy**, **Medium** and **Hard** and then these boxes are split into two, one for the numbers and one for the names with the score. To display the numbers, I have used a for loop that iterate from 1 to 11 and creates text for the number and it also sets the colour of the text to white and sets the size of the text to 1/30th of the height of the users screen so that the text is smaller for smaller screens and larger for larger screens. Because the easy mode is the only mode that the user can play at the moment, I have only added the code for the easy stats. To display the scores I created a variable that will get the top 10 scores in descending order from the database table. I then created a variable called rows that will store all this information and then to display it to the user's screen I iterated through the rows and displayed it as text to the users screen and the colour of this text is set to white and the size of the text is also set to 1/30th of the users screen so that it lines up perfectly with the numbers.

Tests

	Mouse is hovering over the leaderboard button and the left button is clicked.	Valid	The leaderboards should load and the top ten scores for each difficulty are displayed in descending order with the numbers 1 to 10 lined up to the left of the players name	<table border="1"> <thead> <tr> <th></th> <th>Name</th> <th>Score</th> <th>Difficulty</th> </tr> </thead> <tbody> <tr><td>1</td><td>Abby</td><td>200</td><td>1</td></tr> <tr><td>2</td><td>Abby</td><td>400</td><td>1</td></tr> <tr><td>3</td><td>Abby</td><td>500</td><td>1</td></tr> <tr><td>4</td><td>fred</td><td>0</td><td>1</td></tr> <tr><td>5</td><td>abby</td><td>0</td><td>1</td></tr> <tr><td>6</td><td>abby</td><td>370</td><td>1</td></tr> <tr><td>7</td><td>abby</td><td>810</td><td>1</td></tr> <tr><td>8</td><td>Katie</td><td>0</td><td>1</td></tr> <tr><td>9</td><td>Katie</td><td>100</td><td>1</td></tr> <tr><td>10</td><td>Katie</td><td>0</td><td>1</td></tr> <tr><td>11</td><td>katie</td><td>0</td><td>1</td></tr> <tr><td>12</td><td>katie</td><td>0</td><td>1</td></tr> <tr><td>13</td><td>katie</td><td>1730</td><td>1</td></tr> <tr><td>14</td><td>Abby</td><td>0</td><td>1</td></tr> <tr><td>15</td><td>ruby</td><td>-50</td><td>1</td></tr> <tr><td>16</td><td>chloe</td><td>10</td><td>1</td></tr> <tr><td>17</td><td>Abby</td><td>90</td><td>1</td></tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="3">Leaderboards</th> </tr> <tr> <th>Easy</th> <th>Medium</th> <th>Hard</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>2</td><td>2</td></tr> <tr><td>3</td><td>3</td><td>3</td></tr> <tr><td>4</td><td>4</td><td>4</td></tr> <tr><td>5</td><td>5</td><td>5</td></tr> <tr><td>6</td><td>6</td><td>6</td></tr> <tr><td>7</td><td>7</td><td>7</td></tr> <tr><td>8</td><td>8</td><td>8</td></tr> <tr><td>9</td><td>9</td><td>9</td></tr> <tr><td>10</td><td>10</td><td>10</td></tr> </tbody> </table> <p style="text-align: right;">QUIT</p>		Name	Score	Difficulty	1	Abby	200	1	2	Abby	400	1	3	Abby	500	1	4	fred	0	1	5	abby	0	1	6	abby	370	1	7	abby	810	1	8	Katie	0	1	9	Katie	100	1	10	Katie	0	1	11	katie	0	1	12	katie	0	1	13	katie	1730	1	14	Abby	0	1	15	ruby	-50	1	16	chloe	10	1	17	Abby	90	1	Leaderboards			Easy	Medium	Hard	1	1	1	2	2	2	3	3	3	4	4	4	5	5	5	6	6	6	7	7	7	8	8	8	9	9	9	10	10	10	When the left button was clicked the leaderboard menu loaded however the scores didn't load even though there was data on the database so i need to fix this error.
	Name	Score	Difficulty																																																																																																														
1	Abby	200	1																																																																																																														
2	Abby	400	1																																																																																																														
3	Abby	500	1																																																																																																														
4	fred	0	1																																																																																																														
5	abby	0	1																																																																																																														
6	abby	370	1																																																																																																														
7	abby	810	1																																																																																																														
8	Katie	0	1																																																																																																														
9	Katie	100	1																																																																																																														
10	Katie	0	1																																																																																																														
11	katie	0	1																																																																																																														
12	katie	0	1																																																																																																														
13	katie	1730	1																																																																																																														
14	Abby	0	1																																																																																																														
15	ruby	-50	1																																																																																																														
16	chloe	10	1																																																																																																														
17	Abby	90	1																																																																																																														
Leaderboards																																																																																																																	
Easy	Medium	Hard																																																																																																															
1	1	1																																																																																																															
2	2	2																																																																																																															
3	3	3																																																																																																															
4	4	4																																																																																																															
5	5	5																																																																																																															
6	6	6																																																																																																															
7	7	7																																																																																																															
8	8	8																																																																																																															
9	9	9																																																																																																															
10	10	10																																																																																																															

Mouse is hovering over the leaderboard button and the right button is clicked	Invalid	This shouldn't affect anything on the game because the button hasn't been pressed.	<pre> def rightclicked(): print("Right was clicked") Menu = App(title="Supernatural: Saving People Human") Menu.full_screen = True MenuHeight = Menu.height MenuWidth = Menu.width TitleBox = Box(Menu, width="fill", align="top") Background = Picture(Menu, image = "bgd_menu.PNG") ButtonBox = Box(Menu, width = int(MenuWidth/4), button = PushButton(ButtonBox, text ="Play", width= int(MenuWidth/4)) button.text_size = int(MenuHeight/30) button.text_color = DarkRed2 button = PushButton(ButtonBox, text="How To Play") button.text_size = int(MenuHeight/30) button.text_color = DarkRed2 button = PushButton(ButtonBox, text="Leaderboard") button.text_size = int(MenuHeight/30) button.text_color = DarkRed2 button.when_right_button_pressed = rightclicked </pre> 	When the mouse was hovering over the leaderboard button and the right button was clicked nothing happened so this worked how it was meant to.

The leaderboard isn't displaying on the screen even though there is data in the database so I need to find the logic error within my code to fix the leaderboards.

```

EasyBoxStats = Box(EasyBox, width = int(LeaderboardWidth))
for row in rows:
    print(row)
    EasyText = Text(EasyBoxStats, text = row)
    EasyText.text_size = int(LeaderboardHeight/30)
    EasyText.text_color = White

```

The image shows a game board with the word "Easy" at the top and a list of names and scores. To the right is a Python IDLE Shell window displaying the code and output.

Game Board Data:

Row	Column 1	Column 2
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10

IDLE Shell Output:

```
*IDLE Shell 3.11.5*
File Edit Shell Debug Options Win
Python 3.11.5 (tags/v3.11-  
AMD64) ] on win32
Type "help", "copyright",
>>>
= RESTART: C:\Users\Abby\  
rboard.py
pygame 2.5.1 (SDL 2.28.2,  
Hello from the pygame com
-----
*** GUIZERO WARNING ***
Image resizing - cannot s
-----
-----
*** GUIZERO WARNING ***
Image resizing - cannot s
-----
-----
('Abby', 7500)
('Alessandro', 5090)
('edward', 2280)
('katie', 1730)
('Harrison', 1280)
('abby', 810)
('Alessandro', 770)
('Abby', 500)
('Abby', 400)
('abby', 370)
|
```

My first thought was that it wasn't iterating through the rows properly to display it but as you can see from the images above I printed rows and it displayed the top 10 scores within the shell.

I had a closer look at the code again and noticed that the error in my code was to do with the width that I had set the boxes to.

```

EasyBox = Box(Leaderboard, width = int(LeaderboardWidth/4), height = int(LeaderboardHeight/4))
EasyBoxNumbers = Box(EasyBox, width = int(LeaderboardWidth/4), height = int(LeaderboardHeight/4))
for i in range (1,11):
    PlaceText = Text(EasyBoxNumbers, text = i)
    PlaceText.text_size = int(LeaderboardHeight/30)
    PlaceText.text_color = White
strSQL = "SELECT Name, Score FROM UsersScores WHERE Difficulty = :Difficulty"
cursorObj.execute(strSQL)
rows = cursorObj.fetchall()

EasyBoxStats = Box(EasyBox, width = int(LeaderboardWidth/4), height = int(LeaderboardHeight/4))
for row in rows:
    EasyText = Text(EasyBoxStats, text = row)
    EasyText.text_size = int(LeaderboardHeight/30)
    EasyText.text_color = White

MediumBox = Box(Leaderboard, width = int(LeaderboardWidth/4), height = int(LeaderboardHeight/4))
MediumBoxNumbers = Box(MediumBox, width = int(LeaderboardWidth/4), height = int(LeaderboardHeight/4))
for i in range (1,11):
    PlaceText = Text(MediumBoxNumbers, text = i)
    PlaceText.text_size = int(LeaderboardHeight/30)
    PlaceText.text_color = White

```

As you can see in this image the width of the boxes are all a quarter of the width of the screen but if the box for the easy stats is the same as the width of the numbers column then there is no room for the data to fit and that is why it wasn't displayed to the user's screen.

```

EasyBox = Box(Leaderboard, width = int(LeaderboardWidth/4), height = int(LeaderboardHeight/4))
EasyBoxNumbers = Box(EasyBox, width = int(LeaderboardWidth/8), height = int(LeaderboardHeight/4))
for i in range (1,11):
    PlaceText = Text(EasyBoxNumbers, text = i)
    PlaceText.text_size = int(LeaderboardHeight/30)
    PlaceText.text_color = White
strSQL = "SELECT Name, Score FROM UsersScores WHERE Difficulty = :Difficulty"
cursorObj.execute(strSQL)
rows = cursorObj.fetchall()

EasyBoxStats = Box(EasyBox, width = int(LeaderboardWidth/8), height = int(LeaderboardHeight/4))
for row in rows:
    EasyText = Text(EasyBoxStats, text = row)
    EasyText.text_size = int(LeaderboardHeight/30)
    EasyText.text_color = White

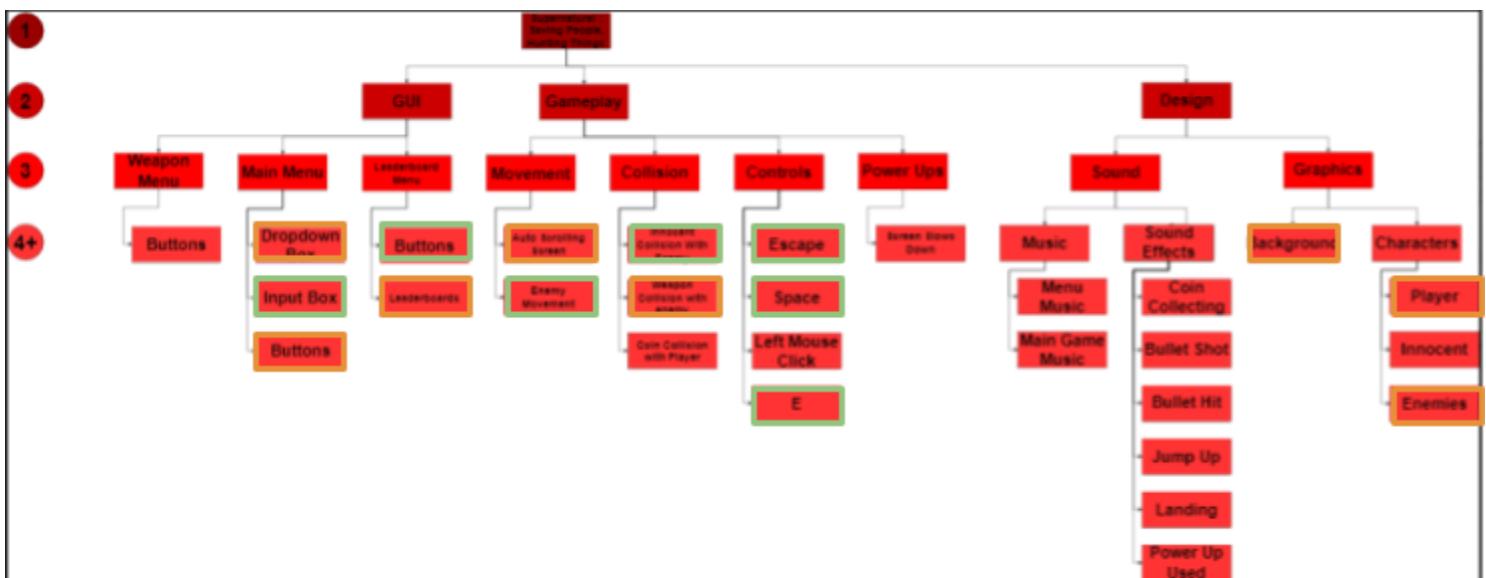
```

I kept the width of the main easy box the same but i halved the width of the other two boxes so they fit.

Tests 2

	Mouse is hovering over the leaderboard button and the left button is clicked.	Valid	The leaderboards should load and the top ten scores for each difficulty are displayed in descending order with the numbers 1 to 10 lined up to the left of the players name		When the left button was clicked the leaderboard menu loaded and the scores were displayed in descending order next to the numbers so this works how it is meant to.
--	---	-------	---	--	--

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed

- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.
 - A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.

- Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
- Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power - ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision
 - Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
 - Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
 - Silver Bullet
 - Kills Werewolves upon collision
 - Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Bone
 - Kills Leviathans upon collision
 - Borax
 - Damages Leviathans (kills them eventually) upon collision

- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 16 - Chosen Weapon

When I've been testing features of my game I have noticed that sometimes it's hard to remember which weapon has been selected so in this version I am going to have the weapon that is currently selected displayed onto the screen.

```
ScoreBoardFont = pygame.font.Font("fnt_HelpMe.ttf", 25)
WeaponFont = pygame.font.Font("fnt_HelpMe.ttf", 50)
```

The first thing I did was add a font for the weapon choice near the start of the program. It is the same font used for displaying the users score on the screen however it is just a different size. I kept the same font to keep the design consistent.

```
def WeaponChoiceText(Text, Colour): # Text procedure
    ScreenText = WeaponFont.render(Text, True, Colour)
    textWidth = ScreenText.get_width()
    Screen.blit(ScreenText,[infoObject.current_w/2 - textWidth/2, infoObject.current_h/4])
*****
```

I added this procedure in just below where all the classes are and it displays the text onto the screen. The text that is going to be displayed and the colour of the text are passed up as parameters. For where the text is positioned onto the screen the program does some maths. For the x coordinate it is half of the width of the screen i half the text width so that the text is always displayed halfway along the users screen no matter how long the text is. For the y coordinate it is just set to the screen's height divided by four so then it is displayed a quarter of the way down the user's screen making it easier for the user to see without covering the main game.

```
WeaponChoiceText(playerweapon,White)
```

Near the end of the game loop I added this line so that the text can be displayed. It passes the name of the weapon up and refreshes every game loop so if the player changes weapon it'll be updated on the screen.

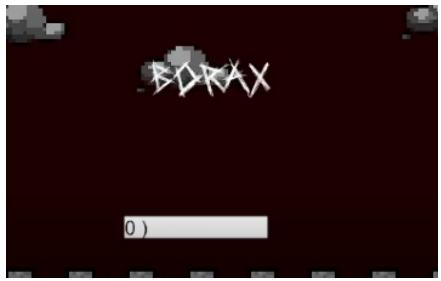
Tests

For this series of tests I have used a keylogger to show what button has been pressed .

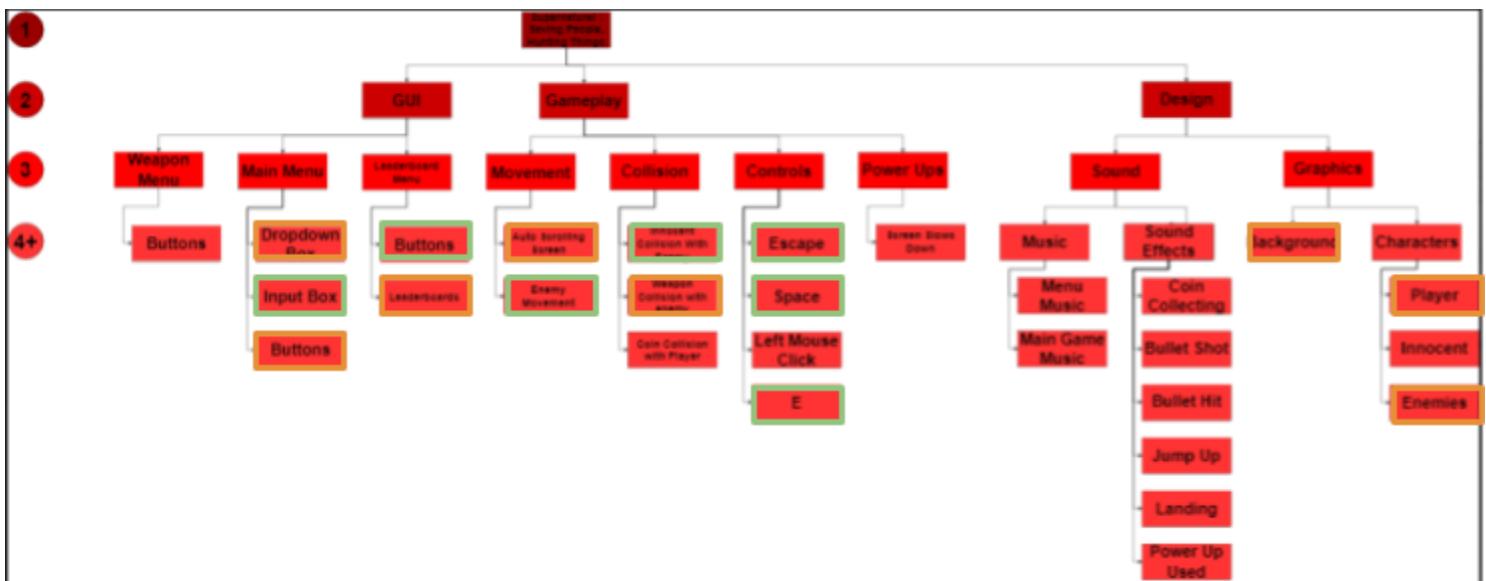
The white box shown is the key logger box.

	User presses 1	Valid	SCYTHE should be displayed on the screen.		When one was pressed it displayed SCYTHE on the screen so this works how it should do.
	User presses 2	Valid	SALT should be displayed on the screen.		When two was pressed it displayed SALT on the screen so this works how it should do.
	User presses 3	Valid	MACHETTE should be displayed on the screen.		When three was pressed it displayed MACHETTE on the screen so this works how it should do.

	User presses 4	Valid	WITCHBULLET should be displayed on the screen.		When four was pressed it displayed WITCHBULLET on the screen so this works how it should do.
	User presses 5	Valid	ANGELBLADE should be displayed on the screen		When five was pressed it displayed ANGELBLADE on the screen so this works how it should do.
	User presses 6	Valid	DEMONBLADE should be displayed on the screen.		When six was pressed it displayed DEMONBLADE on the screen so this works how it should do.
	User presses 7	Valid	SILVERBULLET should be displayed on the screen.		When seven was pressed it displayed SILVERBULLET on the screen so this works how it should do.

	User presses 8	Valid	COLT should be displayed on the screen.		When eight was pressed it displayed COLT on the screen so this works how it should do.
	User presses 9	Valid	BONE should be displayed on the screen.		When nine was pressed it displayed BONE on the screen so this works how it should do.
	User presses 0	Valid	BORAX should be displayed on the screen		When zero was pressed it displayed BORAX on the screen so this works how it should do.

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.

- A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power - ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision

- Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
- Silver Bullet
 - Kills Werewolves upon collision
- Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
- Bone
 - Kills Leviathans upon collision
- Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound
 - Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
 - Main game Music - When the player is playing the main game the main game music will play on loop in the background.
 - Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 17 - Sound

In most games, they make use of sound to make the game more interesting because if you're playing a game in a quiet room you might want the sound on so it isn't silent so in

this version I am going to add sound effects and music to my game. To do this I am going to initialise pygame mixer.

```
pygame.mixer.init() # to play sound
buttonPressedSound = pygame.mixer.Sound("Snd_buttonpressed.WAV") # Imports sound
enemyDiedSound = pygame.mixer.Sound("Snd_dead.WAV")
jumpSound = pygame.mixer.Sound("Snd_jump.WAV")
landSound = pygame.mixer.Sound("Snd_land.WAV")
shootSound = pygame.mixer.Sound("Snd_shot.WAV")
gameMusic = pygame.mixer.music
menuMusic = pygame.mixer.music
```

This section of code is near the start of the program. First it initialises the pygame mixer so that I am able to play the sound and then it imports all the different sounds I am going to use within my game. Some of the sounds are pygame.mixer.sound because these are the different sound effects that are going to be used but at the bottom there are two that are pygame.mixer.music and these are the music that are going to be used. One is for the menu and one is for the actual game.

```
-----  
    shootSound.play()
```

I added this line in the section that runs when the e button is pressed so that it plays the shoot sound as the bullet is fired because you want the sound to sync up with the firing of the bullet.

```
keys = pygame.key.get_pressed()
if not(Jump):
    if keys[pygame.K_SPACE]:
        jumpSound.play()
        Jump = True
else:
    if jumpCount >=-20:
        Player.rect.y -= jumpCount
        jumpCount -= 2
    else: # This will execute if our jump is finished
        landSound.play()
        jumpCount = 20
        Jump = False
    Player.rect.y = (infoObject.current_h*(107/144))-30
    # Resetting our Variables
```

In this section of code I added two lines of code. The first line I added is in the if statement that runs when the space has been pressed and it plays the jump sound so that when the player first jumps there is a sound effect there and the second line I added is in the section of code that executes when the jump is finished and it plays the landing sound here so it aligns up with when the players jump variables are reset because this means it's the end of the jump.

```

bulletHits = pygame.sprite.sprite
if BulletHits:
    enemyDiedSound.play()
    EnemyCount += 1
    score+=100
for eachBullet in weapon == "Salt" and
bulletHits = pygame.sprite.sprite
if BulletHits:
    enemyDiedSound.play()
    EnemyCount += 1
    score+=100
:

```

In this section of code, if the correct bullet hits the enemy the enemy gets killed so i added in a line that adds the enemy killed sound i imported above to further indicate that the enemy has died.

```

def customGame():
    buttonPressedSound.play()
    Menu.hide()
    Customiser.show()
def endgame():
    buttonPressedSound.play()
    endchoice = Menu.yesno("QUIT?", "Are you sure you wa
    if endchoice == True:
        Menu.warn("Thanks for Playing", "Thank you for p
        Menu.destroy()
        pygame.quit()

def ShowLeaderboard():
    buttonPressedSound.play()
    Menu.hide()
    Leaderboard.show()

```

In each of these procedures, it plays a button pressed sound to indicate that a button has been pressed and if any more buttons get added it will also be added to all them buttons so that it is consistent throughout the whole game.

```

def GameInformation():
    buttonPressedSound.play()
    Name = NameInput.value
    Difficulty = DifficultyInput.value
    print(Name)
    if len(Name) < 3:
        Customiser.error("Name is too short")
    elif len(Name) > 10:
        Customiser.error("Name is too long")
    else:
        Customiser.hide()
        gameMusic.load("Snd_GameMusic.WAV")
        time.sleep(1)
        gameMusic.play(-1)
        Game(Name, Difficulty)

def Back():
    buttonPressedSound.play()
    Customiser.hide()
    Menu.show()

```

In these two procedures I added lines so that it plays the button pressed sound however, I also loaded the game music and played it just before running the game procedure so that it plays the game music whilst the game is running and it is set to loop so if the player is on the game when the music ends it loops the music so that the music never stops playing whilst someone is playing the game.

```

def MainMenu():
    buttonPressedSound.play()
    GameOver.hide()
    menuMusic.load("Snd_MenuMusic.WAV")
    time.sleep(1)
    menuMusic.play(-1)
    Menu.show()

```

In this procedure I added three lines. The first one plays the button pressed sound and then it loads the menu music and plays it because this procedure is run when you click to go onto the main menu.

```

*****  

def BackToMain():
    buttonPressedSound.play()
    Leaderboard.hide()
    Menu.show()

```

In this procedure I added in a line that plays the button pressed sound effect.

```

menuMusic.load("Snd_MenuMusic.WAV")
menuMusic.play(-1)
Menu.display()

```

At the very end of the program I added in a line that loads the menu music and then I added a line that plays the menu music just loaded because these two lines are executed just before loading the menu.

Tests

https://drive.google.com/file/d/1zMhPHzOjxDP4ePrOyU_htXg74mTP8geu/view?usp=sharing

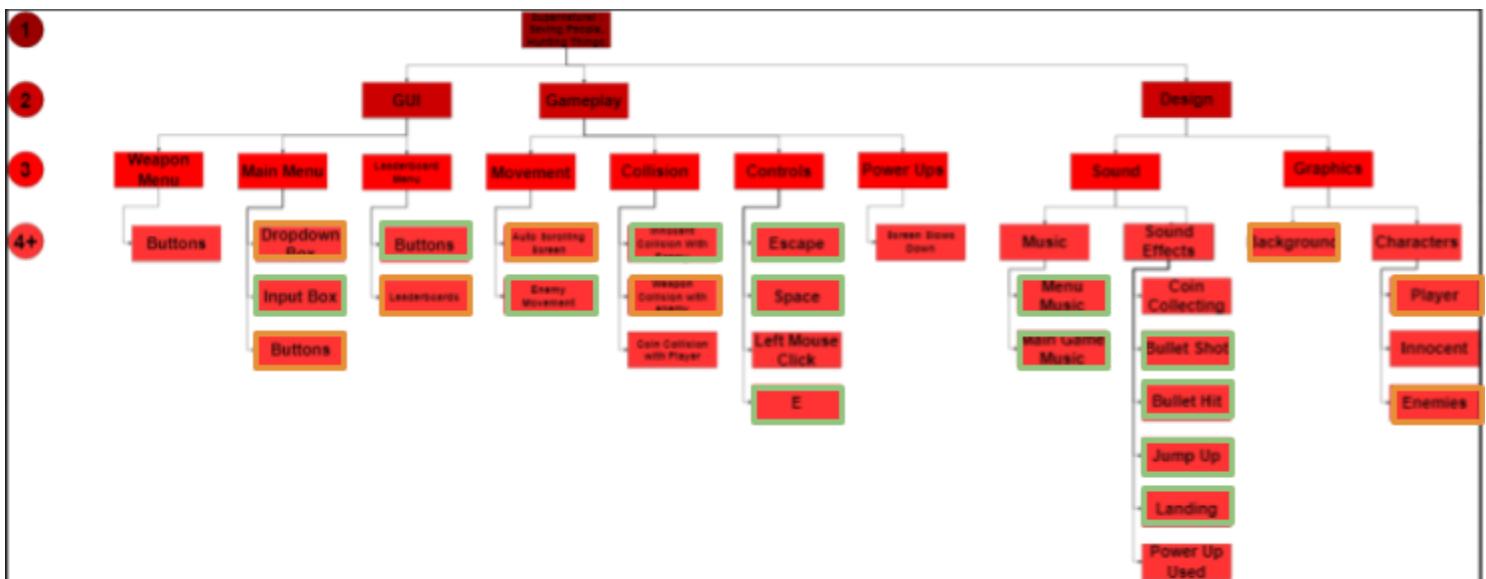
Above is a link to the video tests the different sounds used within my game.

	Test Data	Test Type	Expected Outcome	Actual Outcome	Comments
Menu Music	Being on the menu	Valid	The music for the menu should be playing	0:01	All the music and sound effects work when they are supposed to so no changes need to be made for the sound.
Game Music	Playing the Game	Valid	The music for the game should be playing	0:23	
Bullet Shot	Bullet is created and drawn onto the screen	Valid	Bullet shot audio is played	0:25	
Bullet Hit	Bullet coordinates == Enemy coordinates	Valid	Bullet Hit audio is played	0:29	
Jump Audio	Space Bar is pressed	Valid	Jump audio is played	0:46	
Landing Audio	Player coordinates == original coordinates	Valid	Landing audio is played	0:48	

	tes just after space being pressed				
Button Pressed Audio	Play button is left clicked	Valid	Button pressed sound should play	0:13	
	Leaderboard button is left clicked	Valid	Button pressed sound should play	0:03	
	Back button on leaderboard is left clicked	Valid	Button pressed sound should play	0:07	
	Quit button on main menu is left clicked	Valid	Button pressed sound should play	0:10	
	Back button on customisation menu is left clicked	Valid	Button pressed sound should play	1:25	
	Go back to main menu button on	Valid	Button pressed sound should play	1:34	

	game over screen is left clicked				
	Start button on game customisation menu is left clicked	Valid	Button pressed sound should play	0:20	

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.

- Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.
 - A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power - ups which you buy with coins:

- Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision
 - Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
 - Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
 - Silver Bullet
 - Kills Werewolves upon collision
 - Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Bone
 - Kills Leviathans upon collision
 - Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound

- Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
- Main game Music - When the player is playing the main game the main game music will play on loop in the background.
- Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 18 - Difficulties

One of the main features of my game like in many other games, I am going to have multiple different difficulties. Each difficulty will have a different number of types of enemies, and each difficulty will have its own leaderboard displayed on the leaderboard menu.

```
class Enemy(pygame.sprite.Sprite):

    def __init__(self,x,y,a,b,enemyspeed, image, monster):
        super().__init__()
        self.image = pygame.image.load(image)
        self.image = pygame.transform.scale(self.image, [a,b])
        self.rect = self.image.get_rect() # Hitbox

        self.rect.x = x# Sets the position on the screen
        self.rect.y = y
        self.speed = enemyspeed
        self.monster = monster
        self.health = 100
```

Before adding the different difficulties and more types of enemies I added health to the enemy because some weapons only take some health off the enemy and not all the health.

```
if Difficulty ==1:
    typesOfEnemies = 2
elif Difficulty == 2:
    typesOfEnemies = 5
elif Difficulty == 3:
    typesOfEnemies = 9
```

At the start of the game procedure I added an if statement that checks if the difficulty was set to one, two or three and then depending on what difficulty it sets the number of types of enemies that are going to be included within the game.

```

if EnemyCount>0:
    enemyChoice = random.randint(1,typesOfEnemies)
    if enemyChoice == 1:
        enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-50,30,50,10, "spr_ghost.png", "spirit")
    if enemyChoice == 2:
        enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-70,30,70,10, "spr_death.png", "death")
    if enemyChoice == 3:
        enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-50,30,50,10, "spr_angel.png", "angel")
    if enemyChoice == 4:
        enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-50,30,50,10, "spr_vampire.png", "vampire")
    if enemyChoice == 5:
        enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-50,30,50,10, "spr_witch.png", "witch")
    if enemyChoice == 6:
        enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-50,30,50,10, "spr_leviathan.png", "leviathan")
    if enemyChoice == 7:
        enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-50,30,50,10, "spr_hellhound.png", "hellhound")
    if enemyChoice == 8:
        enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-50,30,50,10, "spr_werewolf.png", "werewolf")
    if enemyChoice == 9:
        enemy = Enemy(infoObject.current_w, (infoObject.current_h*(107/144))-70,20,70,10, "spr_demon.png", "demon")
    EnemyGroup.add(enemy)
    EnemyCount -=1

```

In this if statement that only runs when there are no enemies on screen, it runs a random number generator to decide what enemy is going to spawn and the maximum number for the random number generator is the variable **typeOfEnemies** that was set in the if statement at the top of the game procedure.

```

for eachBullet in BulletGroup:
    if eachBullet.rect.x == infoObject.current_w:
        BulletGroup.remove(eachBullet)#Kills the bullet if it goes off screen
    if eachBullet.weapon == "Scythe" and enemy.monster == "death":
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
        if BulletHits:
            enemyDiedSound.play()
            EnemyCount += 1
            score+=100
    elif eachBullet.weapon == "Salt" and enemy.monster == "spirit":
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
        if BulletHits:
            enemyDiedSound.play()
            EnemyCount += 1
            score+=100
    elif eachBullet.weapon == "AngelBlade" and enemy.monster == "angel":
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
        if BulletHits:
            enemyDiedSound.play()
            EnemyCount += 1
            score+=100
    elif (eachBullet.weapon == "Machette" or eachBullet.weapon == "Colt") and enemy.monster == "vampire":
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
        if BulletHits:
            enemyDiedSound.play()
            EnemyCount += 1
            score+=100
    elif eachBullet.weapon == "WitchBullet" and enemy.monster == "witch":
        BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
        if BulletHits:
            enemyDiedSound.play()
            EnemyCount += 1
            score+=100
    elif enemy.monster == "leviathan" and (eachBullet.weapon == "Bone" or eachBullet.weapon == "Borax"):
        if eachBullet.weapon == "Bone":
            BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
            if BulletHits:
                enemyDiedSound.play()
                EnemyCount += 1
                score+=100

```

```

        elif eachBullet.weapon == "Borax":
            if enemy.health > 10:
                BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, False) #Allows the bullets to collide
                if BulletHits:
                    enemy.health -=10
                    score+=10
            elif enemy.health <=10:
                BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
                if BulletHits:
                    enemy.health -=10
                    score+=10
                    EnemyCount+=1
            elif enemy.monster == "hellhound" and (eachBullet.weapon == "AngelBlade" or eachBullet.weapon == "DemonBlade" or eachBullet.weapon == "Colt" or eachBullet.weapon == "Machette" or eachBullet.weapon == "Salt" or eachBullet.weapon == "SilverBullet") and enemy.health > 10:
                BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, False) #Allows the bullets to collide
                if BulletHits:
                    enemy.health -=10
                    score+=10
            else:
                BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, False) #Allows the bullets to collide
                if BulletHits:
                    score-=10
        elif (eachBullet.weapon == "Machette" or eachBullet.weapon == "AngelBlade" or eachBullet.weapon == "SilverBullet" or eachBullet.weapon == "Colt") and enemy.health > 10:
            BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
            if BulletHits:
                enemyDiedSound.play()
                EnemyCount += 1
                score+=100
        elif (eachBullet.weapon == "AngelBlade" or eachBullet.weapon == "DemonBlade" or eachBullet.weapon == "Colt") and enemy.monster == "demon":
            BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, True) #Allows the bullets to collide
            if BulletHits:
                enemyDiedSound.play()
                EnemyCount += 1
                score+=100
        else:
            BulletHits = pygame.sprite.spritecollide(eachBullet, EnemyGroup, False) #Allows the bullets to collide
            if BulletHits:
                score-=10
        if BulletHits:
            BulletGroup.remove(eachBullet)
    
```

In this giant if statement that is located within the game loop, I am checking to see if a valid monster and weapon combination is chosen and if it is when the bullet collides it will kill the monster, the score will increase and both the monster and the bullet will be removed from the screen.

```

DifficultyText = Text(InputsBox, text = "Difficulty:")
DifficultyText.text_color = White
DifficultyText.text_size = 20
DifficultyInput = Slider(InputsBox, start = 1, end = 3, width = "fill")
DifficultyInput.text_color = White
DifficultyInput.text_size = 20
    
```

In the section of code that creates the customisation screen, I have edited the slider so that it can go from 1 to 3 with one being the easiest difficulty and 3 being the hardest difficulty.

```

t MediumBox = Box(Leaderboard, width = int(LeaderboardWidth/3), height="fill", align = "left")
MediumBoxNumbers = Box(MediumBox, width = int(LeaderboardWidth/24), height="fill", align = "left")
for i in range (1,11):
    PlaceText = Text(MediumBoxNumbers, text = i)
    PlaceText.text_size = int(LeaderboardHeight/30)
    PlaceText.text_color = White
strSQLMedium = "SELECT Name, Score FROM UsersScores WHERE Difficulty = 2 ORDER BY Score DESC LIMIT 10"
cursorObj.execute(strSQLMedium)
rows = cursorObj.fetchall()

Y MediumBoxStats = Box(MediumBox, width = int(LeaderboardWidth/3), height="fill", align = "right")
for row in rows:
    MediumText = Text(MediumBoxStats, text = row)
    MediumText.text_size = int(LeaderboardHeight/30)
    MediumText.text_color = White

HardBox = Box(Leaderboard, width = int(LeaderboardWidth/3), height = "fill", align = "left")
HardBoxNumbers = Box(HardBox, width = int(LeaderboardWidth/24), height = "fill", align = "left")
for i in range (1,11):
    PlaceText = Text(HardBoxNumbers, text = i)
    PlaceText.text_size = int(LeaderboardHeight/30)
    PlaceText.text_color = White
strSQLHard = "SELECT Name, Score FROM UsersScores WHERE Difficulty = 3 ORDER BY Score DESC LIMIT 10"
cursorObj.execute(strSQLHard)
rows = cursorObj.fetchall()

HardBoxStats = Box(HardBox, width = int(LeaderboardWidth/3), height = "fill", align = "right")
for row in rows:
    HardText = Text(HardBoxStats, text = row)
    HardText.text_size = int(LeaderboardHeight/30)
    HardText.text_color = White

```

On the leaderboard menu I have added i code that will also now display the medium and hard scores too. For each difficulty it filters the database by difficulty and then orders the top 10 by decreasing score to then display it onto the leaderboard.

Tests

	Test Data	Test Type	Expected Outcome	Actual Outcome	Comments
Scythe Collision with enemy	Scythe Coordinates == Death Coordinates	Valid	Death health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	Scythe x coordinate is 595 death x coordinate is 610 Scythe x coordinate is 600 death x coordinate is 610 death x coordinate is 1350	As death and the scythe collided the scythe was deleted off the screen and death appeared at the right therefore was killed so this feature works how it is supposed to.

	Scythe Coordinates == Spirits coordinates	Invalid	The weapon disappear s off the screen.	<pre>Scythe x coordinate i 575 spirit x coordinate is 590 Scythe x coordinate i 580 spirit x coordinate is 580 spirit x coordinate is 570</pre>	As the spirit and scythe collided the scythe disappeared but the spirit carried on so this feature works how it is supposed to.
	Scythe Coordinates == Vampire coordinates	Invalid	The weapon disappear s off the screen.	<pre>Scythe x coordinate i 580 vampire x coordinate is 600 Scythe x coordinate i 585 vampire x coordinate is 590 vampire x coordinate is 580</pre>	As the vampire and scythe collided the scythe disappeared but the vampire carried on so this feature works how it is supposed to.
	Scythe Coordinates == Witch coordinates	Invalid	The weapon disappear s off the screen.	<pre>Scythe x coordinate i 505 witch x coordinate is 520 Scythe x coordinate i 510 witch x coordinate is 510 witch x coordinate is 500</pre>	As the witch and scythe collided the scythe disappeared but the witch carried on so this feature works how it is supposed to.
	Scythe Coordinates == Angel coordinates	Invalid	The weapon disappear s off the screen.	<pre>Scythe x coordinate i 580 angel x coordinate is 590 Scythe x coordinate i 585 angel x coordinate is 580 angel x coordinate is 570</pre>	As the angel and scythe collided the scythe disappeared but the angel carried on so this feature works how it is supposed to.

	Scythe Coordinates == Demon coordinates	Invalid	The weapon disappears off the screen.	Scythe x coordinate i 610 demon x coordinate is 620 Scythe x coordinate i 615 demon x coordinate is 610 demon x coordinate is 600	As the demon and scythe collided the scythe disappeared but the demon carried on so this feature works how it is supposed to.
	Scythe Coordinates == Werewolf coordinates	Invalid	The weapon disappears off the screen.	Scythe x coordinate i 580 werewolf x coordinate is 600 Scythe x coordinate i 585 werewolf x coordinate is 590 werewolf x coordinate is 580	As the werewolf and scythe collided the scythe disappeared but the werewolf carried on so this feature works how it is supposed to.
	Scythe Coordinates == Hellhound coordinates	Invalid	The weapon disappears off the screen.	Scythe x coordinate i 590 hellhound x coordinate is 610 Scythe x coordinate i 595 hellhound x coordinate is 600 hellhound x coordinate is 590	As the hellhound and scythe collided the scythe disappeared but the hellhound carried on so this feature works how it is supposed to.
	Scythe Coordinates == Leviathan coordinates	Invalid	The weapon disappears off the screen.	Scythe x coordinate i 570 leviathan x coordinate is 580 Scythe x coordinate i 575 leviathan x coordinate is 570 leviathan x coordinate is 560	As the leviathan and scythe collided the scythe disappeared but the leviathan carried on so this feature works how it is

					supposed to.
Salt Collision with enemy	Salt Coordinates == Death Coordinates	Invalid	The weapon disappears off the screen.	<pre> Salt x coordinate i 635 death x coordinate is 650 Salt x coordinate i 640 death x coordinate is 640 death x coordinate is 630 </pre>	As death and the salt collided the salt disappeared but death carried on so this feature works how it is supposed to.
	Salt Coordinates == Spirit Coordinates	Valid	Spirit health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre> Salt x coordinate i 380 spirit x coordinate is 390 Salt x coordinate i 385 spirit x coordinate is 390 spirit x coordinate is 1350 </pre>	As spirit and the salt collided the salt was deleted off the screen and the spirit appeared at the right therefore was killed so this feature works how it is supposed to.
	Salt Coordinates == Vampire Coordinates	Invalid	The weapon disappears off the screen.	<pre> Salt x coordinate i 640 vampire x coordinate is 660 Salt x coordinate i 645 vampire x coordinate is 650 vampire x coordinate is 640 </pre>	As the vampire and the salt collided the salt disappeared but the vampire carried on so this feature works how it is supposed to.
	Salt Coordinates == Witch Coordinates	Invalid	The weapon disappears off the screen.	<pre> Salt x coordinate i 375 witch x coordinate is 390 Salt x coordinate i 380 witch x coordinate is 380 witch x coordinate is 370 </pre>	As the witch and the salt collided the salt disappeared but the witch carried on so this feature works how it is

				supposed to.
Salt Coordi nates == Angel Coordi nates	Invalid	The weapon disappear s off the screen	<pre> Salt x coordinate i 590 angel x coordinate is 600 Salt x coordinate i 595 angel x coordinate is 590 angel x coordinate is 580 </pre>	As the angel and the salt collided the salt disappeared but the angel carried on so this feature works how it is supposed to.
Salt Coordi nates == Demon Coordi nates	Invalid	The weapon disappear s off the screen	<pre> Salt x coordinate i 620 demon x coordinate is 640 Salt x coordinate i 625 demon x coordinate is 630 demon x coordinate is 620 </pre>	As the demon and the salt collided the salt disappeared but the demon carried on so this feature works how it is supposed to.
Salt Coordi nates == Werewol f Coordi nates	Invalid	The weapon disappear s off the screen	<pre> Salt x coordinate i 390 werewolf x coordinate is 410 Salt x coordinate i 395 werewolf x coordinate is 400 werewolf x coordinate is 390 </pre>	As the werewolf and the salt collided the salt disappeared but the werewolf carried on so this feature works how it is supposed to.
Salt Coordi nates == Hellhou nd Coordi nates Hellhou nd health>1 0	Valid	The weapon disappear s off the screen and the hellhound s health decreases by 10	<pre> Salt x coordinate i 620 hellhound x coordinate is 630 hellhound's health is 20 Salt x coordinate i 625 hellhound x coordinate is 620 hellhound's health is 10 hellhound x coordinate is 610 hellhound's health is 10 </pre>	As the hellhound and the salt collided the salt was deleted and the hellhound's health decreased by 10 o this feature works how it is meant to.

	Salt Coordinates == Hellhound Coordinates Hellhound Health<= 10	Invalid	The weapon disappears off the screen.	<pre> Salt x coordinate i 400 hellhound x coordinate is 420 hellhound's health is 10 salt x coordinate i 405 hellhound x coordinate is 410 hellhound's health is 10 hellhound x coordinate is 400 hellhound's health is 10 </pre>	As the salt and the hellhound passed each other the health remained the same and the salt was deleted off the screen so this feature works how it is meant to.
	Salt Coordinates == Leviathan Coordinates	Invalid	The weapon disappears off the screen	<pre> Salt x coordinate i 555 leviathan x coordinate is 570 Salt x coordinate i 560 leviathan x coordinate is 560 leviathan x coordinate is 550 </pre>	As the leviathan and the salt collided the salt disappeared but the leviathan carried on so this feature works how it is supposed to.
Machete Collision with enemy	Machete Coordinates == Death Coordinates	Invalid	The weapon disappears off the screen.	<pre> Machette x coordinate i 405 death x coordinate is 420 Machette x coordinate i 410 death x coordinate is 410 death x coordinate is 400 </pre>	As death and the machete collided the machete disappeared but death carried on so this feature works how it is supposed to.
	Machete Coordinates == Spirit Coordinates	Invalid	The weapon disappears off the screen.	<pre> Machette x coordinate i 390 spirit x coordinate is 410 Machette x coordinate i 395 spirit x coordinate is 400 spirit x coordinate is 390 </pre>	As the spirit and the machete collided the machete disappeared but the spirit carried on so this feature works how it is supposed to.

				supposed to.
Machete Coordinates == Vampire Coordinates	Valid	Vampire health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre> Machette x coordinate i 395 vampire x coordinate is 410 Machette x coordinate i 400 vampire x coordinate is 410 vampire x coordinate is 1350 </pre>	As the vampire and the machete collided the machete was deleted off the screen and the vampire appeared at the right therefore was killed so this feature works how it is supposed to.
Machete Coordinates == Witch Coordinates	Invalid	The weapon disappears off the screen.	<pre> Machette x coordinate i 640 witch x coordinate is 660 Machette x coordinate i 645 witch x coordinate is 650 witch x coordinate is 640 </pre>	As the witch and the machete collided the machete disappeared but the witch carried on so this feature works how it is supposed to.
Machete Coordinates == Angel Coordinates	Invalid	The weapon disappears off the screen	<pre> Machette x coordinate i 540 angel x coordinate is 560 Machette x coordinate i 545 angel x coordinate is 550 angel x coordinate is 540 </pre>	As the angel and the machete collided the machete disappeared but the angel carried on so this feature works how it is supposed to.
Machete Coordinates == Demon	Invalid	The weapon disappears off the	<pre> Machette x coordinate i 650 demon x coordinate is 660 Machette x coordinate i 655 demon x coordinate is 650 demon x coordinate is 640 </pre>	As the demon and the machete collided the

	Coordinates		screen		machete disappeared but the demon carried on so this feature works how it is supposed to.
	Machete Coordinates == Werewolf Coordinates	Valid	Werewolf health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre> Machette x coordinate i 640 werewolf x coordinate is 650 Machette x coordinate i 645 werewolf x coordinate is 650 werewolf x coordinate is 1350 </pre>	As the werewolf and the machete collided the machete was deleted off the screen and the werewolf appeared at the right therefore was killed so this feature works how it is supposed to.
	Machete Coordinates == Hellhound Coordinates	Invalid	The weapon disappears off the screen	<pre> Machette x coordinate i 605 hellhound x coordinate is 620 Machette x coordinate i 610 hellhound x coordinate is 610 hellhound x coordinate is 600 </pre>	As the hellhound and the machete collided the machete disappeared but the hellhound carried on so this feature works how it is supposed to.
	Machete Coordinates == Leviathan Coordinates	Invalid	The weapon disappears off the screen	<pre> Machette x coordinate i 615 leviathan x coordinate is 630 Machette x coordinate i 620 leviathan x coordinate is 620 leviathan x coordinate is 610 </pre>	As the leviathan and the machete collided the machete disappeared but the leviathan carried on so

					this feature works how it is supposed to.
WitchBullet Collision with Enemy	Witch Bullet coordinates == Death Coordinates	Invalid	The weapon disappears off the screen	<pre> WitchBullet x coordinate i 515 death x coordinate is 530 WitchBullet x coordinate i 520 death x coordinate is 520 death x coordinate is 510 </pre>	As death and the witch bullet collided the witch bullet disappeared but death carried on so this feature works how it is supposed to.
	Witch Bullet Coordinates == Spirit Coordinates	Invalid	The weapon disappears off the screen	<pre> WitchBullet x coordinate i 375 spirit x coordinate is 390 WitchBullet x coordinate i 380 spirit x coordinate is 380 spirit x coordinate is 370 </pre>	As the spirit and the witch bullet collided the witch bullet disappeared but the spirit carried on so this feature works how it is supposed to.
	Witch Bullet Coordinates == Vampire Coordinates	Invalid	The weapon disappears off the screen	<pre> WitchBullet x coordinate i 620 vampire x coordinate is 630 WitchBullet x coordinate i 625 vampire x coordinate is 620 vampire x coordinate is 610 </pre>	As the vampire and the witch bullet collided the witch bullet disappeared but the vampire carried on so this feature works how it is supposed to.
	Witch Bullet Coordinates == Witch Coordinates	Valid	Witch health decreases to 0 and gets removed off the screen.	<pre> WitchBullet x coordinate i 415 witch x coordinate is 430 WitchBullet x coordinate i 420 witch x coordinate is 430 witch x coordinate is 1350 </pre>	As the witch and the witch bullet collided the witch bullet was deleted off the screen and the witch appeared at the

			The weapon also disappears off the screen.		right therefore was killed so this feature works how it is supposed to.
Witch Bullet Coordinates == Angel Coordinates	Invalid		The weapon disappears off the screen	<pre> WitchBullet x coordinate i 570 angel x coordinate is 580 WitchBullet x coordinate i 575 angel x coordinate is 570 angel x coordinate is 560 </pre>	As the angel and the witch bullet collided the witch bullet disappeared but the angel carried on so this feature works how it is supposed to.
Witch Bullet Coordinates == Demon Coordinates	Invalid		The weapon disappears off the screen	<pre> WitchBullet x coordinate i 635 demon x coordinate is 650 WitchBullet x coordinate i 640 demon x coordinate is 640 demon x coordinate is 630 </pre>	As the demon and the witch bullet collided the witch bullet disappeared but the demon carried on so this feature works how it is supposed to.
Witch Bullet Coordinates == Werewolf Coordinates	Invalid		The weapon disappears off the screen	<pre> WitchBullet x coordinate i 620 werewolf x coordinate is 640 WitchBullet x coordinate i 625 werewolf x coordinate is 630 werewolf x coordinate is 620 </pre>	As the werewolf and the witch bullet collided the witch bullet disappeared but the werewolf carried on so this feature works how it is supposed to.
Witch Bullet Coordinates ==	Invalid		The weapon disappears off the	<pre> WitchBullet x coordinate i 605 hellhound x coordinate is 620 WitchBullet x coordinate i 610 hellhound x coordinate is 610 hellhound x coordinate is 600 </pre>	As the hellhound and the witch bullet collided the

	Vampire coordinates		screen		disappeared but the vampire carried on so this feature works how it is supposed to.
	Angel blade Coordinates == Witch coordinates	Invalid	The weapon disappears off the screen	<pre>AngelBlade x coordinate i 645 witch x coordinate is 660 AngelBlade x coordinate i 650 witch x coordinate is 650 witch x coordinate is 640</pre>	As the witch and the angel blade collided the angel blade disappeared but the witch carried on so this feature works how it is supposed to.
	Angel blade Coordinates == Angel coordinates	Valid	angel health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre>AngelBlade x coordinate i 385 angel x coordinate is 400 AngelBlade x coordinate i 390 angel x coordinate is 400 angel x coordinate is 1350</pre>	As the angel and the angel blade collided the angel blade was deleted off the screen and the angel appeared at the right therefore was killed so this feature works how it is supposed to.
	Angel blade Coordinates == Demon coordinates	Valid	Demon health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre>AngelBlade x coordinate i 630 demon x coordinate is 640 AngelBlade x coordinate i 635 demon x coordinate is 640 demon x coordinate is 1350 . . .</pre>	As the demon and the angel blade collided the angel blade was deleted off the screen and the demon appeared at the right therefore was killed so this feature works how it is supposed to.

		s off the screen.		supposed to.
Angel blade Coordinates == Werewolf coordinates	Valid	Werewolf health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre>AngelBlade x coordinate i 625 werewolf x coordinate is 640 AngelBlade x coordinate i 630 werewolf x coordinate is 640 werewolf x coordinate is 1350</pre>	As the werewolf and the angel blade collided the angel blade was deleted off the screen and the werewolf appeared at the right therefore was killed so this feature works how it is supposed to.
Angel blade Coordinates == Hellhound coordinates	Valid	Hellhound health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre>AngelBlade x coordinate i 640 hellhound x coordinate is 650 AngelBlade x coordinate i 645 hellhound x coordinate is 650 hellhound x coordinate is 1350</pre>	As the hellhound and the angel blade collided the angel blade was deleted off the screen and the spirit appeared at the right therefore was killed so this feature works how it is supposed to.
Angel blade Coordinates == Leviathan coordinates	Invalid	The weapon disappears off the screen	<pre>AngelBlade x coordinate i 570 leviathan x coordinate is 580 AngelBlade x coordinate i 575 leviathan x coordinate is 570 leviathan x coordinate is 560 - - - - -</pre>	As the leviathan and the angel blade collided the angel blade disappeared but the leviathan carried on so this feature works how it is supposed to.

Demon blade Collision with enemy	Demon blade Coordinates == Death Coordinates	Invalid	The weapon disappears off the screen	<pre>DemonBlade x coordinate is 600 death x coordinate is 610 DemonBlade x coordinate is 605 death x coordinate is 600 death x coordinate is 590</pre>	As death and the demon blade collided the demon blade disappeared but death carried on so this feature works how it is supposed to.
	Demon blade Coordinates == Spirit Coordinates	Invalid	The weapon disappears off the screen	<pre>DemonBlade x coordinate is 390 spirit x coordinate is 410 DemonBlade x coordinate is 395 spirit x coordinate is 400 spirit x coordinate is 390 spirit x coordinate is 390</pre>	As the spirit and the demon blade collided the demon blade disappeared but the spirit carried on so this feature works how it is supposed to.
	Demon blade Coordinates == Vampire Coordinates	Invalid	The weapon disappears off the screen	<pre>DemonBlade x coordinate is 610 vampire x coordinate is 630 DemonBlade x coordinate is 615 vampire x coordinate is 620 vampire x coordinate is 610</pre>	As the vampire and the demon blade collided the demon blade disappeared but the vampire carried on so this feature works how it is supposed to.
	Demon blade Coordinates == Witch Coordinates	Invalid	The weapon disappears off the screen	<pre>DemonBlade x coordinate is 610 witch x coordinate is 630 DemonBlade x coordinate is 615 witch x coordinate is 620 witch x coordinate is 610 witch x coordinate is 610</pre>	As the witch and the demon blade collided the demon blade disappeared but the witch carried on so this feature

				works how it is supposed to.
Demon blade Coordinates == Angel Coordinates	Invalid	The weapon disappears off the screen	<pre>DemonBlade x coordinate is 610 angel x coordinate is 630 DemonBlade x coordinate is 615 angel x coordinate is 620 angel x coordinate is 610</pre>	As the angel and the demon blade collided the demon blade disappeared but the angel carried on so this feature works how it is supposed to.
Demon blade Coordinates == Demon Coordinates	Valid	Demon health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre>DemonBlade x coordinate is 630 demon x coordinate is 650 DemonBlade x coordinate is 635 demon x coordinate is 650 demon x coordinate is 1350</pre>	As the demon and the demon blade collided the demon blade was deleted off the screen and the demon appeared at the right therefore was killed so this feature works how it is supposed to.
Demon blade Coordinates == Werewolf Coordinates	Invalid	The weapon disappears off the screen	<pre>DemonBlade x coordinate is 620 werewolf x coordinate is 640 DemonBlade x coordinate is 625 werewolf x coordinate is 630 werewolf x coordinate is 620 -- -- -- -- --</pre>	As the werewolf and the demon blade collided the demon blade disappeared but the werewolf carried on so this feature works how it is supposed to.

	Demon blade Coordinates == Hellhound Coordinates	Valid	Hellhound health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre>DemonBlade x coordinate i 630 hellhound x coordinate is 640 DemonBlade x coordinate i 635 hellhound x coordinate is 640 hellhound x coordinate is 1350</pre>	As the hellhound and the demon blade collided the demon blade was deleted off the screen and the hellhound appeared at the right therefore was killed so this feature works how it is supposed to.
	Demon blade Coordinates == Leviathan Coordinates	Invalid	The weapon disappears off the screen	<pre>DemonBlade x coordinate i 630 leviathan x coordinate is 640 DemonBlade x coordinate i 635 leviathan x coordinate is 630 leviathan x coordinate is 620</pre>	As the leviathan and the demon blade collided the demon blade disappeared but the leviathan carried on so this feature works how it is supposed to.
Silver Bullet Collision with enemy	Silver Bullet Coordinates == Death Coordinates	Invalid	The weapon disappears off the screen	<pre>SilverBullet x coordinate i 500 death x coordinate is 520 SilverBullet x coordinate i 505 death x coordinate is 510 death x coordinate is 500</pre>	As death and the silver bullet collided the silver bullet disappeared but death carried on so this feature works how it is supposed to.
	Silver Bullet Coordinates == Spirit	Invalid	The weapon disappears off the screen	<pre>SilverBullet x coordinate i 560 spirit x coordinate is 570 SilverBullet x coordinate i 565 spirit x coordinate is 560 spirit x coordinate is 550</pre>	As the spirit and the silver bullet collided the silver bullet disappeared

	Coordinates				but the spirit carried on so this feature works how it is supposed to.
Silver Bullet Coordinates == Vampire Coordinates	Invalid	The weapon disappears off the screen	<pre>SilverBullet x coordinate i 630 vampire x coordinate is 640 SilverBullet x coordinate i 635 vampire x coordinate is 630 vampire x coordinate is 620</pre>	As the vampire and the silver bullet collided the silver bullet disappeared but the vampire carried on so this feature works how it is supposed to.	
Silver Bullet Coordinates == Witch Coordinates	Invalid	The weapon disappears off the screen	<pre>SilverBullet x coordinate i 550 witch x coordinate is 560 SilverBullet x coordinate i 555 witch x coordinate is 550 witch x coordinate is 540</pre>	As the witch and the silver bullet collided the silver bullet disappeared but the witch carried on so this feature works how it is supposed to.	
Silver Bullet Coordinates == Angel Coordinates	Invalid	The weapon disappears off the screen	<pre>SilverBullet x coordinate i 600 angel x coordinate is 620 SilverBullet x coordinate i 605 angel x coordinate is 610 angel x coordinate is 600</pre>	As the angel and the silver bullet collided the silver bullet disappeared but the angel carried on so this feature works how it is supposed to.	
Silver Bullet Coordinates == Demon Coordinates	Invalid	The weapon disappears off the screen	<pre>SilverBullet x coordinate i 640 demon x coordinate is 660 SilverBullet x coordinate i 645 demon x coordinate is 650 demon x coordinate is 640</pre>	As the demon and the silver bullet collided the silver bullet disappeared but the demon	

Colt Collision with Enemy	Colt coordinates == Death Coordinates	Invalid	The weapon disappears off the screen	<pre>Colt x coordinate i 590 death x coordinate is 610 Colt x coordinate i 595 death x coordinate is 600 death x coordinate is 590</pre>	As death and the colt collided the colt disappeared but death carried on so this feature works how it is supposed to.
	Colt Coordinates == Spirit Coordinates	Invalid	The weapon disappears off the screen	<pre>Colt x coordinate i 380 spirit x coordinate is 400 Colt x coordinate i 385 spirit x coordinate is 390 spirit x coordinate is 380</pre>	As the spirit and the colt collided the colt disappeared but the spirit carried on so this feature works how it is supposed to.
	Colt Coordinates == Vampire Coordinates	Valid	Vampire health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre>Colt x coordinate i 595 vampire x coordinate is 610 Colt x coordinate i 600 vampire x coordinate is 610 vampire x coordinate is 1350</pre>	As the vampire and the colt collided the colt was deleted off the screen and the vampire appeared at the right therefore was killed so this feature works how it is supposed to.
	Colt Coordinates == Witch Coordinates	Invalid	The weapon disappears off the screen	<pre>Colt x coordinate i 585 witch x coordinate is 600 Colt x coordinate i 590 witch x coordinate is 590 witch x coordinate is 580</pre>	As the witch and the colt collided the colt disappeared but the witch carried on so this feature works how it is supposed to.

	Colt Coordinates == Angel Coordinates	Invalid	The weapon disappears off the screen	<pre>Colt x coordinate is 600 angel x coordinate is 620 Colt x coordinate is 605 angel x coordinate is 610 angel x coordinate is 600</pre>	As the angel and the colt collided the colt disappeared but the angel carried on so this feature works how it is supposed to.
	Colt Coordinates == Demon Coordinates	Valid	Demon health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre>Colt x coordinate is 620 demon x coordinate is 630 Colt x coordinate is 625 demon x coordinate is 630 demon x coordinate is 1350</pre>	As the demon and the colt collided the colt was deleted off the screen and the demon appeared at the right therefore was killed so this feature works how it is supposed to.
	Colt Coordinates == Werewolf Coordinates	Valid	Werewolf health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre>Colt x coordinate is 590 werewolf x coordinate is 610 Colt x coordinate is 595 werewolf x coordinate is 610 werewolf x coordinate is 1350</pre>	As the werewolf and the colt collided the colt was deleted off the screen and the werewolf appeared at the right therefore was killed so this feature works how it is supposed to.
	Colt Coordinates == Hellhound	Valid	Hellhound health decreases to 0 and gets	<pre>Colt x coordinate is 595 hellhound x coordinate is 610 Colt x coordinate is 600 hellhound x coordinate is 610 hellhound x coordinate is 1350</pre>	As the hellhound and the colt collided the colt was deleted off the

	Coordinates		removed off the screen. The weapon also disappears off the screen.		screen and the hellhound appeared at the right therefore was killed so this feature works how it is supposed to.
	Colt Coordinates == Leviathan Coordinates	Invalid	The weapon disappears off the screen	Colt x coordinate i 630 leviathan x coordinate is 640 Colt x coordinate i 635 leviathan x coordinate is 630 leviathan x coordinate is 620	As the leviathan and the colt collided the colt disappeared but the leviathan carried on so this feature works how it is supposed to.
Bone Collision with enemy	Bone Coordinates == Death Coordinates	Invalid	The weapon disappears off the screen	Bone x coordinate i 610 death x coordinate is 630 Bone x coordinate i 615 death x coordinate is 620 death x coordinate is 610	As death and the bone collided the bone disappeared but death carried on so this feature works how it is supposed to.
	Bone Coordinates == Spirit Coordinates	Invalid	The weapon disappears off the screen	Bone x coordinate i 610 spirit x coordinate is 620 Bone x coordinate i 615 spirit x coordinate is 610 spirit x coordinate is 600	As the spirit and the bone collided the bone disappeared but the spirit carried on so this feature works how it is supposed to.

	Bone Coordinates == Vampire Coordinates	Invalid	The weapon disappears off the screen	<pre> Bone x coordinate is 595 vampire x coordinate is 610 Bone x coordinate is 600 vampire x coordinate is 600 vampire x coordinate is 590 </pre>	As the vampire and the bone collided the bone disappeared but the vampire carried on so this feature works how it is supposed to.
	Bone Coordinates == Witch Coordinates	Invalid	The weapon disappears off the screen	<pre> Bone x coordinate is 610 witch x coordinate is 630 Bone x coordinate is 615 witch x coordinate is 620 witch x coordinate is 610 </pre>	As the witch and the bone collided the bone disappeared but the witch carried on so this feature works how it is supposed to.
	Bone Coordinates == Angel Coordinates	Invalid	The weapon disappears off the screen	<pre> Bone x coordinate is 585 angel x coordinate is 600 Bone x coordinate is 590 angel x coordinate is 590 angel x coordinate is 580 </pre>	As the angel and the bone collided the bone disappeared but the angel carried on so this feature works how it is supposed to.
	Bone Coordinates == Demon Coordinates	Invalid	The weapon disappears off the screen	<pre> Bone x coordinate is 620 demon x coordinate is 630 Bone x coordinate is 625 demon x coordinate is 620 demon x coordinate is 610 </pre>	As the demon and the bone collided the bone disappeared but the demon carried on so this feature works how it is supposed to.

	Bone Coordinates == Werewolf Coordinates	Invalid	The weapon disappears off the screen	<pre>Bone x coordinate is 610 werewolf x coordinate is 630 Bone x coordinate is 615 werewolf x coordinate is 620 werewolf x coordinate is 610</pre>	As the werewolf and the bone collided the bone disappeared but the werewolf carried on so this feature works how it is supposed to.
	Bone Coordinates == Hellhound Coordinates	Invalid	The weapon disappears off the screen	<pre>Bone x coordinate is 585 hellhound x coordinate is 600 Bone x coordinate is 590 hellhound x coordinate is 590 hellhound x coordinate is 580</pre>	As the hellhound and the bone collided the bone disappeared but the hellhound carried on so this feature works how it is supposed to.
	Bone Coordinates == Leviathan Coordinates	Valid	Leviathan health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre>Bone x coordinate is 420 leviathan x coordinate is 430 Bone x coordinate is 425 leviathan x coordinate is 430 leviathan x coordinate is 1350</pre>	As the leviathan and the bone collided the bone was deleted off the screen and the leviathan appeared at the right therefore was killed so this feature works how it is supposed to.
Borax Collision with Enemy	Borax coordinates == Death Coordinates	Invalid	The weapon disappears off the screen	<pre>Borax x coordinate is 610 death x coordinate is 620 Borax x coordinate is 615 death x coordinate is 610 death x coordinate is 600</pre>	As death and the borax collided the borax disappeared

	ates			but death carried on so this feature works how it is supposed to.
	Borax Coordinates == Spirit Coordinates	Invalid	The weapon disappears off the screen	Borax x coordinate i 390 spirit x coordinate is 400 Borax x coordinate i 395 spirit x coordinate is 390 spirit x coordinate is 380
	Borax Coordinates == Vampire Coordinates	Invalid	The weapon disappears off the screen	Borax x coordinate i 385 vampire x coordinate is 400 Borax x coordinate i 390 vampire x coordinate is 390 vampire x coordinate is 380 ^--
	Borax Coordinates == Witch Coordinates	Invalid	The weapon disappears off the screen	Borax x coordinate i 595 witch x coordinate is 610 Borax x coordinate i 600 witch x coordinate is 600 witch x coordinate is 590
	Borax Coordinates == Angel Coordinates	Invalid	The weapon disappears off the screen	Borax x coordinate i 640 angel x coordinate is 650 Borax x coordinate i 645 angel x coordinate is 640 angel x coordinate is 630

				carried on so this feature works how it is supposed to.
Borax Coordinates == Demon Coordinates	Invalid	The weapon disappears off the screen	<pre>Borax x coordinate i 405 demon x coordinate is 420 Borax x coordinate i 410 demon x coordinate is 410 demon x coordinate is 400</pre>	As the demon and the borax collided the borax disappeared but the demon carried on so this feature works how it is supposed to.
Borax Coordinates == Werewolf Coordinates	Invalid	The weapon disappears off the screen	<pre>Borax x coordinate i 640 werewolf x coordinate is 660 Borax x coordinate i 645 werewolf x coordinate is 650 werewolf x coordinate is 640</pre>	As the werewolf and the borax collided the borax disappeared but the werewolf carried on so this feature works how it is supposed to.
Borax Coordinates == Hellhound Coordinates	Invalid	The weapon disappears off the screen	<pre>Borax x coordinate i 600 hellhound x coordinate is 610 Borax x coordinate i 605 hellhound x coordinate is 600 hellhound x coordinate is 590</pre>	As the hellhound and the borax collided the borax disappeared but the hellhound carried on so this feature works how it is supposed to.

	Borax Coordinates == Leviathan Coordinates Leviathan health > 10	Valid	Leviathan health decreases by 10. The weapon also disappears off the screen.	<pre>Borax x coordinate i 575 leviathan x coordinate is 590 leviathan's health is 20 Borax x coordinate i 580 leviathan x coordinate is 580 leviathan's health is 10 leviathan x coordinate is 570 leviathan's health is 10 leviathan x coordinate is 560</pre>	As the borax and the leviathan collided the health of the leviathan decreased by ten and the borax disappeared off the screen so this feature works how it is supposed to
	Borax coordinates == Leviathan coordinates Leviathan health <= 10	Valid	Leviathan health decreases to 0 and gets removed off the screen. The weapon also disappears off the screen.	<pre>Borax x coordinate i 405 leviathan x coordinate is 420 leviathan's health is 10 Borax x coordinate i 410 leviathan x coordinate is 420 leviathan's health is 0 leviathan x coordinate is 1350 leviathan's health is 20 leviathan x coordinate is 1340</pre>	As the borax and the leviathan collided the leviathans health decreased to zero and a new one spawned on the right and the borax was deleted off the screen so this feature works how it is supposed to.

	Enemy is removed off screen	Valid	The players score increases by 100	<pre>Bone x coordinate i 400 leviathan x coordinate is 420 leviathan's health is 90 Score 1490 Bone x coordinate i 405 leviathan x coordinate is 420 leviathan's health is 90 Score 1590 werewolf x coordinate is 1350 werewolf's health is 100 Score 1590</pre>	The enemy was removed off the screen and another came from the right and the score also increased by 100 so this feature works how it is
--	-----------------------------	-------	------------------------------------	--	--

					supposed to.
	Bullet hits enemy but enemy health remains the same	Valid	The players score decreases by 10	<pre> Scythe x coordinate is 620 vampire x coordinate is 630 vampire's health is 100 Score 1690 Scythe x coordinate is 625 vampire x coordinate is 620 vampire's health is 100 Score 1680 vampire x coordinate is 610 vampire's health is 100 Score 1680 </pre>	The scythe was removed off the screen but the enemies health remained the same and the score decreased by 100 so this feature works how it is meant to.
	Bullet hits enemy and enemy health decreases	Valid	The players score increases by the amount of damage taken off the enemy	<pre> Borax x coordinate is 575 leviathan x coordinate is 590 leviathan's health is 100 Score 1480 Borax x coordinate is 580 leviathan x coordinate is 580 leviathan's health is 90 Score 1490 </pre>	The enemies health decreased by 10 and the score increased by 10 so this feature works how it is supposed to
Difficulty Slider	Difficulty Slider is on one	Valid	There should be two different types of enemies	Difficulty is 1 so there are 2 different types of enemies	When the difficulty slider was on one there were only two different types of enemies so this works how it is meant to.
	Difficulty Slider is on two	Valid	There should be five different types of enemies	Difficulty is 2 so there are 5 different types of enemies	When the difficulty slider was on two there were five different types of enemies so this works how

					it is supposed to.																																	
	Difficulty Slider is on three	Valid	There should be nine different types of enemies	Difficulty is 3 so there are 9 different types of enemies	When the difficulty slider was on three there were nine different types of enemies so this works how it is supposed to.																																	
Leaderboards	The user clicks the leaderboard button	Valid	Three leaderboards should be displayed on the screen with the top ten scores in descending order	<p style="text-align: center;">Leaderboards</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; background-color: black; color: white;">Easy</th> <th style="text-align: center; background-color: black; color: white;">Medium</th> <th style="text-align: center; background-color: black; color: white;">Hard</th> </tr> </thead> <tbody> <tr><td>1 Abby 7500</td><td>1 Abby 2170</td><td>1 Abby 3990</td></tr> <tr><td>2 Alessandro 5090</td><td>2 Abby 1790</td><td>2 Abby 2150</td></tr> <tr><td>3 edward 2280</td><td>3 Olivia 1710</td><td>3 Abby 1710</td></tr> <tr><td>4 katie 1730</td><td>4 Abby 1390</td><td>4 test 1680</td></tr> <tr><td>5 Harrison 1280</td><td>5 Abby 1310</td><td>5 nsme 1170</td></tr> <tr><td>6 abby 1020</td><td>6 Abby 1090</td><td>6 ALS 980</td></tr> <tr><td>7 Abby 920</td><td>7 Abby 690</td><td>7 Abby 920</td></tr> <tr><td>8 Test 850</td><td>8 Abby 200</td><td>8 Abby 890</td></tr> <tr><td>9 Abby 850</td><td>9 Abby 0</td><td>9 Abby 700</td></tr> <tr><td>10 abby 810</td><td>10 (Bob Marley) 0</td><td>10 Harrison 610</td></tr> </tbody> </table> <p style="text-align: right; margin-top: -10px;">QUIT</p>	Easy	Medium	Hard	1 Abby 7500	1 Abby 2170	1 Abby 3990	2 Alessandro 5090	2 Abby 1790	2 Abby 2150	3 edward 2280	3 Olivia 1710	3 Abby 1710	4 katie 1730	4 Abby 1390	4 test 1680	5 Harrison 1280	5 Abby 1310	5 nsme 1170	6 abby 1020	6 Abby 1090	6 ALS 980	7 Abby 920	7 Abby 690	7 Abby 920	8 Test 850	8 Abby 200	8 Abby 890	9 Abby 850	9 Abby 0	9 Abby 700	10 abby 810	10 (Bob Marley) 0	10 Harrison 610	All three leaderboards were different and all scores were displayed in descending order so this works how it is supposed to.
Easy	Medium	Hard																																				
1 Abby 7500	1 Abby 2170	1 Abby 3990																																				
2 Alessandro 5090	2 Abby 1790	2 Abby 2150																																				
3 edward 2280	3 Olivia 1710	3 Abby 1710																																				
4 katie 1730	4 Abby 1390	4 test 1680																																				
5 Harrison 1280	5 Abby 1310	5 nsme 1170																																				
6 abby 1020	6 Abby 1090	6 ALS 980																																				
7 Abby 920	7 Abby 690	7 Abby 920																																				
8 Test 850	8 Abby 200	8 Abby 890																																				
9 Abby 850	9 Abby 0	9 Abby 700																																				
10 abby 810	10 (Bob Marley) 0	10 Harrison 610																																				

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar
- Enemy
 - Moves from the right to left at a constant speed
 - Is removed from the screen when the health is zero
 - When an enemy has died another will spawn on the right of the screen
 - Another enemy reappears on the right if an enemy goes off the screen to the left
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.

- A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
 - Exit - There will be an exit button in the bottom right hand corner that will quit the game.
- Power - ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision
 - Angel Blade

- Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
- Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
- Silver Bullet
 - Kills Werewolves upon collision
- Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
- Bone
 - Kills Leviathans upon collision
- Borax
 - Damages Leviathans (kills them eventually) upon collision

- Score

- Increases by 100 when enemy killed
- Increase by 10 when player jumps over enemy
- Decrease by 10 when wrong bullet hits enemy
- Increases by 10 when some damage taken off enemy

- Sound

- Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
- Main game Music - When the player is playing the main game the main game music will play on loop in the background.
- Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 19 - Different Players

For the last main edit of my game I am going to add a few features to it like different playable hunters and the how to play menu.

```

class Sprite(pygame.sprite.Sprite):

    def __init__(self,x,y,a,b,startImage):
        super().__init__()
        self.image = pygame.image.load(startImage) # Sets the size
        self.image = pygame.transform.scale(self.image,[a,b])
        self.rect = self.image.get_rect() # Hitbox
        self.rect.x = x# Sets the position on the screen
        self.rect.y = y
        self.health = 100

```

To create the different playable characters I edited the sprite class so that instead of having it as one colour I have an image passed up and set the image to the image that is passed up.

```

def Game(Name, Difficulty, Hunter):
    i = 0
    Jump = False
    jumpCount = 20
    score = 0
    EnemyCount = 1
    playerweapon = "Scythe"
    weaponcolour = (0,0,0)
    if Difficulty ==1:
        typesOfEnemies = 2
    elif Difficulty == 2:
        typesOfEnemies = 5
    elif Difficulty == 3:
        typesOfEnemies = 9

    if Hunter == "Dean Winchester":
        startImage = "spr_dean.png"
        imageList = ["spr_dean.png", "spr_dean2.png", "spr_dean.png", "spr_dean3.png"]
    if Hunter == "Sam Winchester":
        startImage = "spr_sam.png"
        imageList = ["spr_sam.png", "spr_sam2.png", "spr_sam.png", "spr_sam3.png"]

```

I pass up the hunter as a parameter and I have two if statements that set the starting image and the image list depending on what hunter the user chose.

```

if Player.health == 0:
    SpriteGroup.remove()
    EnemyGroup.remove(eachEnemy)
    AddToLeaderboard(Name, score, Di
    gameExit = True
    GameOver(Name, score)
    pygame.display.update()

```

At the end of the game loop where it checks if the player is dead I edited a line so that it takes the user to a game over screen instead that displays the user's score to them before they then click to go back to the main menu.

```

    Customiser()
def loadHowToPlay():
    buttonPressedSound.play()
    Menu.hide()
    HowToPlay()

```

I added a how to play procedure that runs when you click the how to play button and this procedure plays the button pressed sound then hides the main menu and shows the how to play menu.

```

def GameInformation():
    buttonPressedSound.play()
    Name = NameInput.value
    Difficulty = DifficultyInput.value
    Hunter = HunterInput.value
    print(Name)
    if len(Name) < 3:
        Customiser.error("Name is too short", "The")
    elif len(Name) > 10:
        Customiser.error("Name is too long", "The")
    else:
        Customiser.hide()
        gameMusic.load("Snd_GameMusic.WAV")
        gameMusic.play(-1)
        Game(Name, Difficulty, Hunter)

```

In this procedure I added in a line that creates a variable called hunter and sets it to the value of the combo box and then this value gets passed up to the game procedure as a parameter.

```
HunterInput = Combo(InputsBox, options = ["Dean Winchester", "Sam Winchester"], width = "fill")
```

I edited this line to give the user the options of Dean winchester or sam winchester for the hunter they would like to play as.

```

def HowToPlay():
    def BackToMain():
        buttonPressedSound.play()
        HowToPlay.hide()
        Menu.show()
    HowToPlay = Window(Menu, title = "Customisation Menu", bg = Black)
    HowToPlay.full_screen = True

    HowToPlayHeight = HowToPlay.height
    HowToPlayWidth = HowToPlay.width

    TitleBox3 = Box(HowToPlay, width="fill", align="top")
    Background2 = Picture(TitleBox3, image = "bgd_menu.PNG")

    HowToPlay1 = Picture(HowToPlay, image = "menu_UseCustomisation.PNG", align = "left")
    HowToPlay2 = Picture(HowToPlay, image = "menu_PlayMainGame.PNG", align = "left")

    BackBox = Box(HowToPlay, align="bottom", width = "fill")
    button = PushButton(BackBox, text ="BACK", align = "right", command = BackToMain)
    button.text_size = int(MenuHeight/30)
    button.text_color = DarkRed2
    .....

```

I added in a hw to play menu. At the start there is a procedure that will run when the user presses the back button and this procedure plays the button pressed sound, hides the how to play menu and shows the main menu. The how to play menu is set as a window for the main menu and the background is black. It is set to be displayed in full screen. The values of the width and the height of the how to play menu are then set as variables. Along the top I have created a title box that holds the image displayed on the main menu because I wanted to keep the design consistent. For the instructions on how to play I created two images and then I displayed these images to the left of the menu. In the bottom right corner I have added a back button that runs the procedure created above the how to play menu when pressed.

```

def GameOver(Name, Score):
    def MainMenu():
        buttonPressedSound.play()
        GameOver.hide()
        menuMusic.load("Snd_MenuMusic.WAV")
        menuMusic.play(-1)
        Menu.show()

    ScoreText = (Name, "your score is", Score)

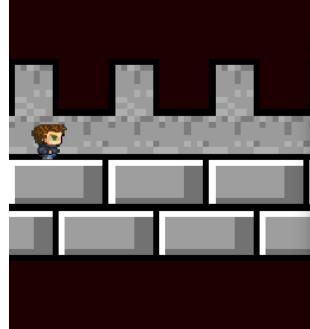
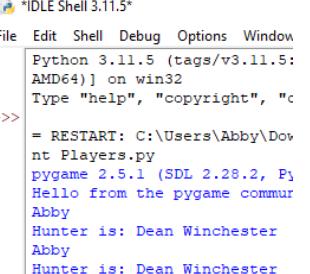
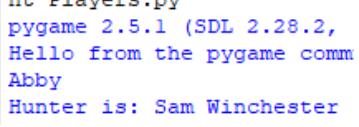
    GameOver = Window(Menu, title = "Game Over", bg = DarkRed)
    GameOver.full_screen = True
    GameOverText = Text(GameOver, text = "Game Over!", width = "fill", height = "fill")
    GameOver.text_size = 50
    GameOver.text_color = White
    GameOverText = Text(GameOver, text = ScoreText, width = "fill", height = "fill")
    GameOver.text_size = 100
    GameOver.text_color = White
    GameOverButton = PushButton(GameOver , text="Go back to Main Menu", command = MainMenu)

```

Along with all the other menus I created a game over menu that displays the user's score to them so it takes in the name and score as parameters. The procedure at the top runs when the user presses the Back to Main Menu button and his procedure plays the button pressed sound, Hides the game over screen, plays the menu music and displays the main menu. I then created a string to store the text that I want displayed on the game over screen. I then proceed to actually creating the game over screen and this is created as a window of the main menu and the background is set to dark red. The game over screen is set to full screen. At the top I created a title that says game over and the size of this text is set to 50 and the colour of this text is set to white. In the middle of the game over screen I am displaying the string that we created just before creating the game over screen and the size of this text is set to 100 with the colour also being white. At the bottom of the game over screen I created a button that says go back to the main menu and it runs the procedure created at the start of this section of code.

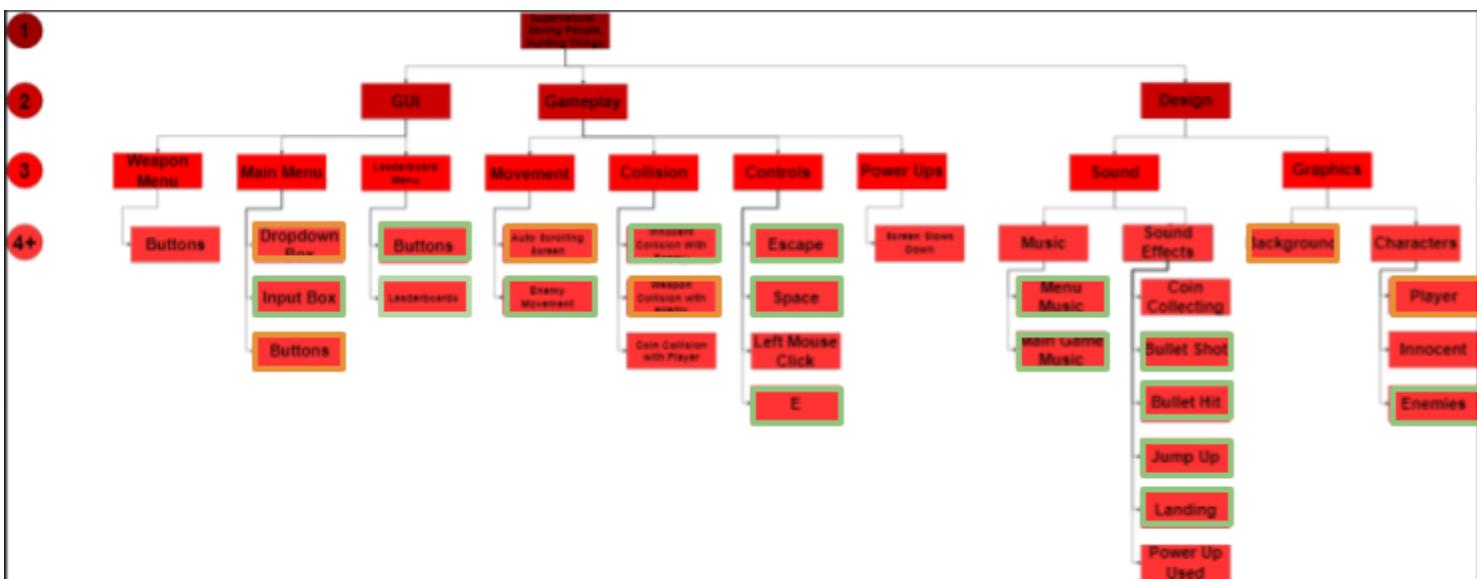
Tests

	Test Data	Test Type	Expected Outcome	Actual Outcome	Comments
--	-----------	-----------	------------------	----------------	----------

Hunter selection	Dean winchester is selected from the hunter list	Valid	Dean winchester is displayed on the screen	 	When dean winchester was selected as the hunter dean winchester was displayed on the screen so this works as expected.
	Sam winchester is selected from the hunter list	Valid	Sam winchester is displayed on the screen	 	When sam winchester was selected as the hunter sam winchester was displayed on the screen so this works how it is supposed to.
Game over	The game over screen is displayed	Valid	It shows the user their score	 <p>Abby Score: 430</p>	When the user died it told them their score therefore this feature works how it is supposed to.
How to Play Menu	The mouse is hovering over the how to play menu button	Valid	The how to play menu is displayed		When the how to play button is clicked it displayed the how to play menu therefore this fulture works how it is meant to.

	The user is on the how to play menu and presses the back button	Valid	The main menu is displayed		When on the how to play menu and the back button got pressed the main menu was displayed therefore this works how it is supposed to.
--	---	-------	----------------------------	--	--

Top Down Modular Update



Success Criteria

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed
- Player
 - Jumps at a constant height when the space bar is pressed
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same.
 - Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the

player collides with the enemy and is displayed on the screen in the form of a health bar

- Enemy

- Moves from the right to left at a constant speed
- Is removed from the screen when the health is zero
- When an enemy has died another will spawn on the right of the screen
- Another enemy reappears on the right if an enemy goes off the screen to the left

- Three different difficulty modes:

- An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns
- A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns.
- A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns.

- Different types of enemies depending on the difficulty choice:

- Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death
- Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel.
- Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan.

- Menu with different options:

- Name Input - there is a text box that the player is able to type their name into
- Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play.
- Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard.
- Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right.
- Exit - There will be an exit button in the bottom right hand corner that will quit the game.

- Power - ups which you buy with coins:

- Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision
 - Salt
 - Kills Spirits upon collision
 - Harms Hellhounds but never kills upon collision
 - Machete
 - Kills vampires upon collision
 - Kills Werewolves upon collision
 - Witch Bullet
 - Kills Witches upon collision
 - Angel Blade
 - Kills Angels upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Kills Hellhounds upon collision
 - Demon Blade
 - Kills Demons upon collision
 - Kills Hellhounds upon collision
 - Silver Bullet
 - Kills Werewolves upon collision
 - Colt
 - Kills vampires upon collision
 - Kills Hellhounds upon collision
 - Kills Demons upon collision
 - Kills Werewolves upon collision
 - Bone
 - Kills Leviathans upon collision
 - Borax
 - Damages Leviathans (kills them eventually) upon collision
- Score
 - Increases by 100 when enemy killed
 - Increase by 10 when player jumps over enemy
 - Decrease by 10 when wrong bullet hits enemy
 - Increases by 10 when some damage taken off enemy
- Sound

- Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play
- Main game Music - When the player is playing the main game the main game music will play on loop in the background.
- Main Menu Music - When the player is on a menu the main menu music will play in the background on loop

Version 20 - Cleaning up

In this version I am just cleaning up the code so if there are any variables with confusing names then I will change the names into meaningful names and also I will be adding comments throughout my code explaining what everything does. The final code after the cleaning is done is available in the appendix.

Stakeholder Signatures



Evaluation

Post Development Test Table

Below is my final test table so that I can be certain that my game works correctly. It isn't as detailed as the development tests because I've already tested all the tiny details, I

just need to ensure that my game works properly how a game should work and also check that when editing code it doesn't stop something else from working.

The timings in the test table are where each test has been carried out within my post development test video

Function	Test Data	Expected Result	Actual Result
Player - Movement	The player presses space to jump	The player moves upwards then downwards giving the illusion that the player is jumping	4:12
Player - Collision	The player collides with the different objects	Enemy collision - The player object gets deleted and the health of the player goes to zero. Coin collision - The coin object gets deleted and the money variable increases	5:17
Enemy - Collision	The different types of enemies collide with the different types of weapons.	Enemy collides with the weapon that kills it - The enemy health goes to zero and the enemy object gets deleted and a coin takes its place. Enemy collides with the weapon that damages it - The enemies health decreases by a bit but it doesn't go to zero unless the health is already low. Enemy collides with a weapon that does no damage to it - The enemies health remains the same and the enemy	3:48 9:44 4:34

		remains the same.	
GUI - Main Menu	The different buttons are clicked	The right feature is used when the buttons are clicked.	0:27
GUI - Weapon Menu	The “e” button is clicked and the different buttons are clicked when on the menu.	The game pauses and a menu weapon is displayed with the different weapons on. The different buttons on the menu work correctly	I didn’t end up adding this feature in so instead at 4:32 it shows how the weapon system actually works.

Post Development Test Video

To ensure that my game works correctly I did a test video and all the time included in my post development test table are an indication as to where in the video that test is carried out. Below is the link to my test video.

<https://drive.google.com/file/d/15tXttEK9xkXkPl4kqp6nbBRid0mUci-i/view?usp=sharing>

Evaluation of Post development Tests

In my tests everything that I added had worked correctly and there were no issues with how the game worked. I didn’t add the weapon menu because it would be a smoother gameplay without however the weapon system that I didn’t include worked how it should do. I also didn’t add any power-ups because of time but if I had more time I would have added this feature into my game.

Although there were no bugs within my game it does need to be improved because the gameplay was too easy which was very easily noticeable when I played the game properly whilst carrying out my post development tests.

My game isn’t complete and there are features that I am wanting to edit however, my game is robust so is playable by the stakeholders.

Final Questionnaire

In order to find out if my stakeholders liked my game and what parts they liked and disliked, I created a final questionnaire. All stakeholders played the game before filling out the questionnaire. The questionnaire is going to give me an insight into if my game properly works because other people playing the game might come across bugs that I haven’t and it helps me to figure out how I can improve my game to make it more enjoyable to stakeholders. These are the questions that I asked in my stakeholders:

1. How old are you? - Multiple choice
 - a. 10
 - b. 11
 - c. 12
 - d. 13
 - e. 14
 - f. 15
 - g. 16
 - h. 17
 - i. 18+
2. Is the menu easy to navigate? - Multiple Choice
 - a. Yes
 - b. No
3. Which of the menus can you read the text on? - Checkbox
 - a. Main Menu
 - b. Game Customisation
 - c. How to Play
 - d. Leaderboards
 - e. Game Over
4. Is the game easy to control? - Multiple Choice
 - a. Yes
 - b. No
5. How difficult was the game? - Linear Scale
 - a. 1 - 5 with 1 being not difficult and 5 being very difficult
6. How many stars would you give this game - Linear Scale
 - a. 1-5
7. Was the colour scheme consistent throughout the menus and game? - Multiple Choice
 - a. Yes
 - b. No
8. Did the colour scheme match the theme of the game (Supernatural)? - Multiple Choice
 - a. Yes
 - b. No
9. Did you like the layout of the game? - Multiple Choice
 - a. Yes
 - b. No
10. Did you like the layout of the menu? - Multiple Choice
 - a. Yes
 - b. No

11. Which graphics did you like? - Checkbox

- a. Character
- b. Enemy
- c. Background
- d. Title
- e. None

12. What did you like about my game? - Short Answer

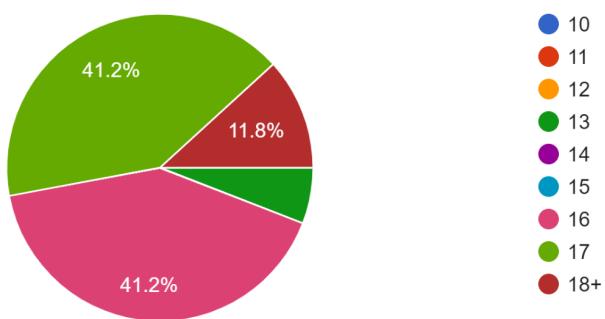
13. What didn't you like about my game? - Short Answer

14. How can I make my game better? - Short Answer

Responses to final questionnaire

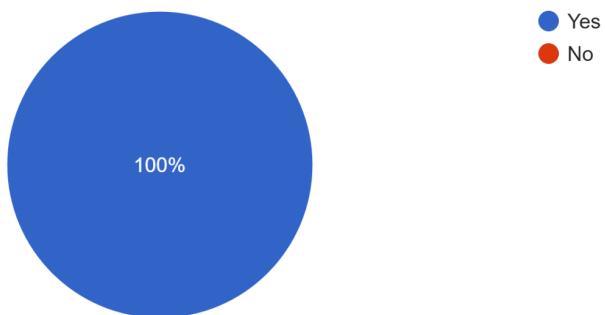
How old are you?

17 responses



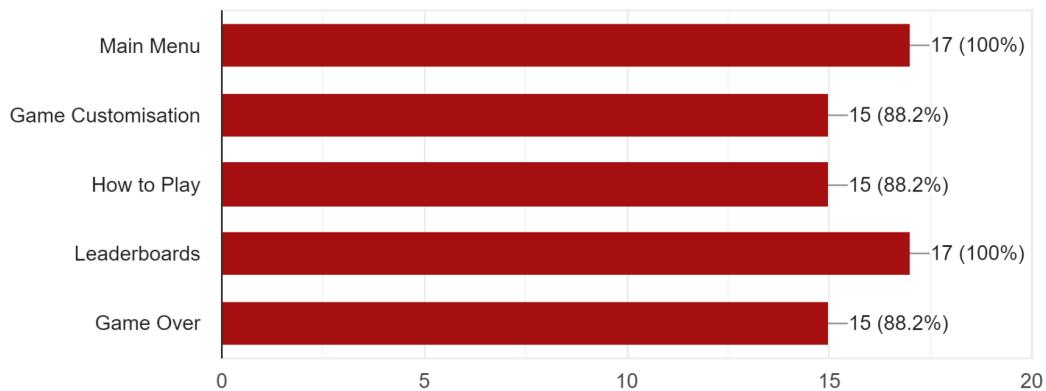
Is the menu easy to navigate?

17 responses



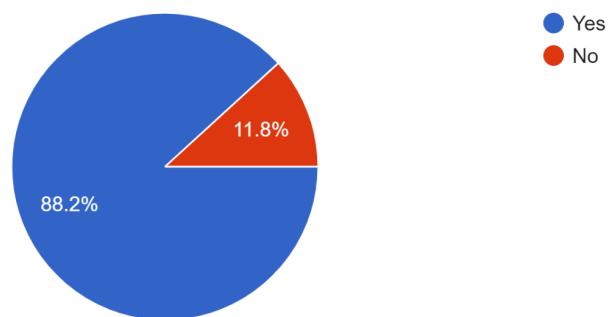
Which of the menu's can you read the text on?

17 responses



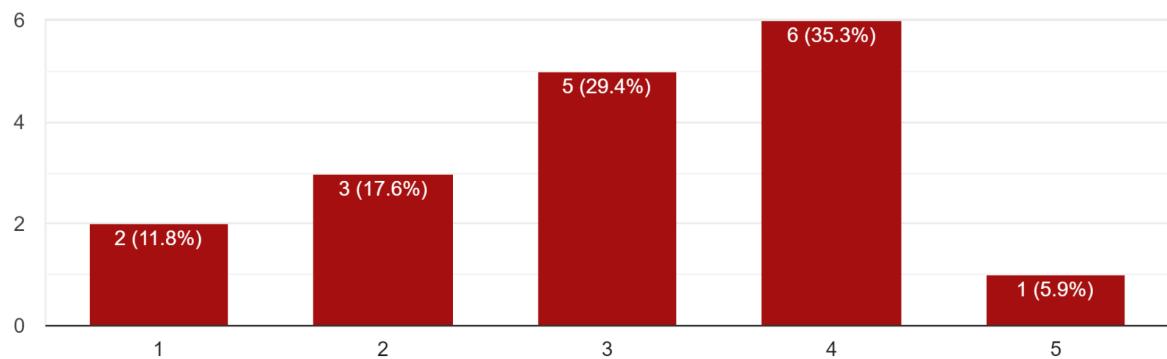
Is the game easy to control?

17 responses



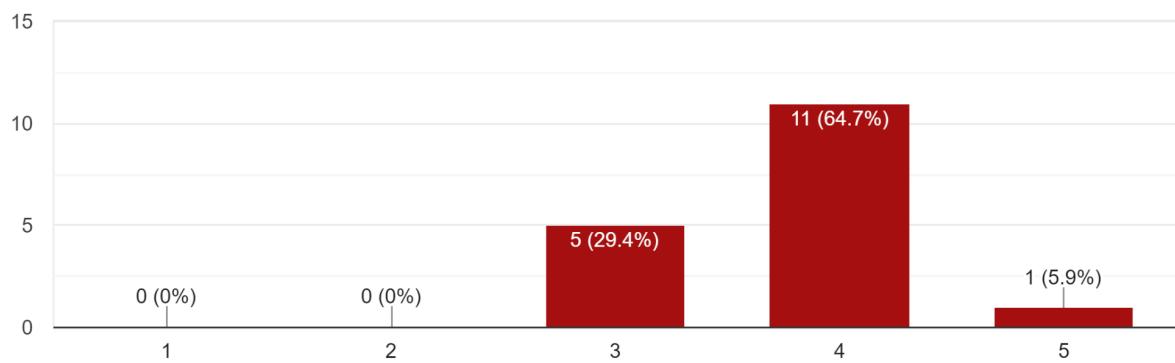
How difficult was the game?

17 responses



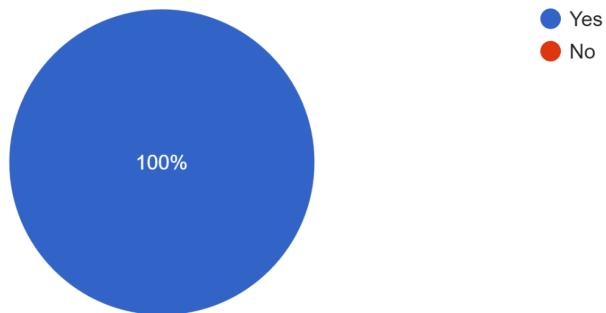
How many stars would you give this game?

17 responses



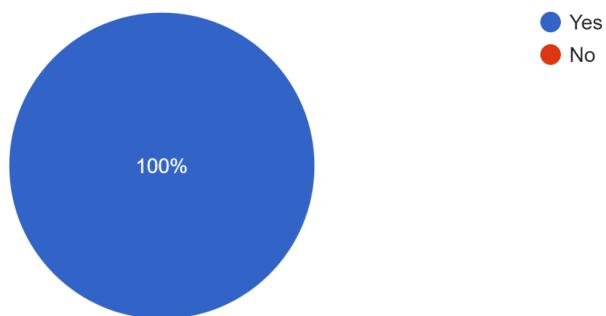
Was the colour scheme consistent throughout the menus and game?

17 responses



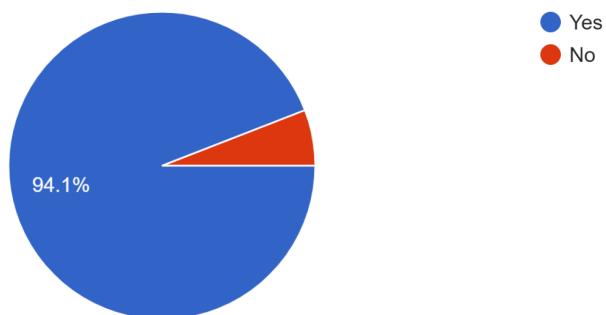
Did the colour scheme match the theme of the game (Supernatural)?

17 responses



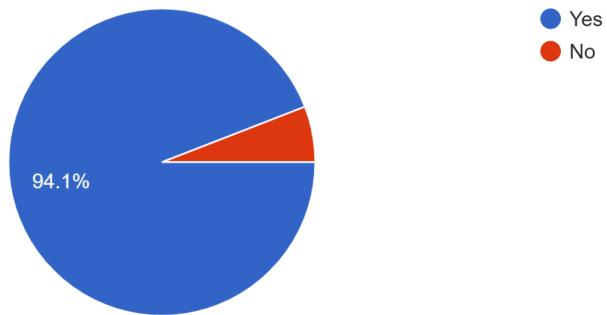
Did you like the layout of the game?

17 responses



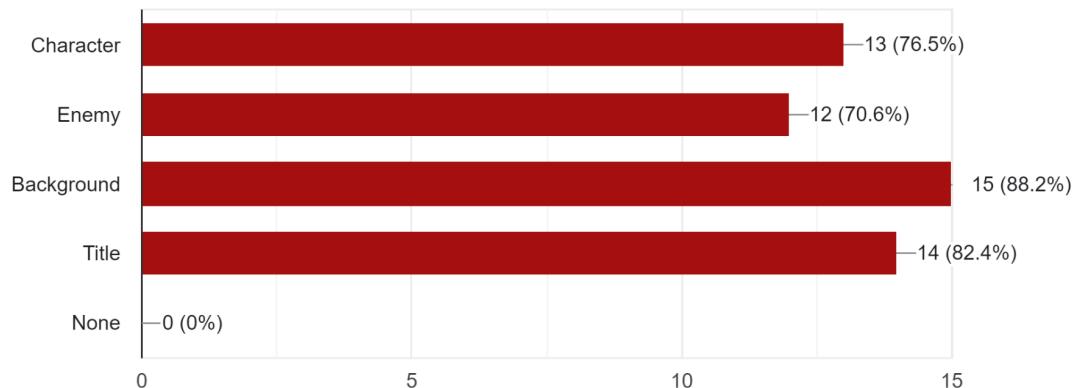
Did you like the layout of the menu?

17 responses



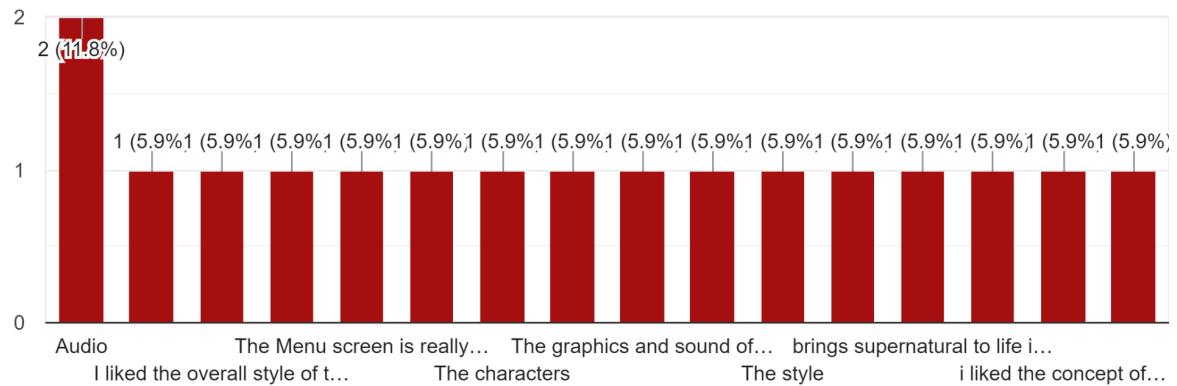
Which graphics did you like?

17 responses



What did you like about my game?

17 responses



It is difficult to see what people have said because this was represented as a bar chart due to 2 people saying “Audio” so below I will post each response individually.

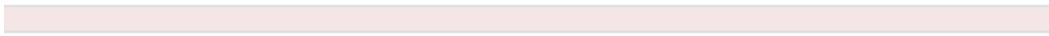
Audio

[2 responses](#)



brings supernatural to life in an engaging game experience

[1 response](#)



fun

[1 response](#)



The different weapons for different enemies.

[1 response](#)

Good speed and music

1 response



The style

1 response



The Menu screen is really well designed and implemented

1 response



The sound effects and control of the game

1 response

Variation of weapons

1 response

The graphics and sound of the bullets.

1 response

The background

1 response

Style

1 response

The characters

1 response

I liked the overall style of the game, the graphics and sound effects were nice

1 response

i liked the concept of changing characters through the game

1 response

intuituve iunterface

1 response

What didn't you like about my game?

17 responses

Hard to know which enemies are defeated with each weapon

i didnt like the controls having to shoot with E and pressing 123456 to change wepons

The character art is very sim[;e

How wordy the explanation of how to play is

Maybe too easy at the start.

Lack of information about what weapons are most effective against which enemies

Too many weapons

the medium difficulty was too easy

Could get a negative score

Score system

i didnt understand how to swithc between buttons

Score system too harsh

fun

It was a bit easy (played on medium) and there was no explanation for the enemies or what the weapons did

Lack of difficulty

Individual weapon attack effects

The size of text

How can i make my game better?

17 responses

make it so you know which weapons defeat each enemy

i noticed when adding my name i did not know if i was clicked on the box or not you could add something to show the curser when typing your name

more detailed pixel art for characters

Maybe at the start of the game for 20 seconds, show what number the player has to press to defeat the enemy with the correct weapon

Increase the difficulty slightly.

Include information about what weapons are most effective against which enemies

Make it more clear which weapons are for what enemies

have a health bar on enemies and confused how there was negative points

Explain different weapons better

Brief description on what to do when starting game

Fix health bar

Info on weapons and what they do

more fun

add information about enemies and weapons

Make it harder

More unique effects for each weapon

Bigger text,different hunters like bobby

Evaluation of responses to final questionnaire

Question 1 - I asked this question to confirm that the people playing the game were in my target market age range. Most of the people said they were 16 or 17 and a few said they were 18. One person said they were 13 so most of the people who answered my questionnaire were in the age range of my target market so therefore all the results in the rest of the questionnaire are valid because it was a stakeholder in my target market that gave the response.

Question 2 - I asked this question because you want people to be able to navigate the menu easily in order to be able to play the game because people are less likely to play a game when they can't navigate a menu. Everyone who played my game said that the menu was easy to navigate which means that nothing has to change about my menu system.

Question 3 - I asked this question because I needed to ensure that people can read the text on the menus so they understand what is going on. It is important to be able to read all the menus because, for the main menu if you can't read it you can't even navigate to be able to play the game, for the game customisation menu you need to be able to read the menu to input your name, difficulty and hunter, for the how to play menu you need to be able to read it so you can understand how to play the game, for the leaderboards you need to be able to read it in order to compare your score with other players and for the game over screen you need to be able to read it to see how well you did in that game. Everyone who answered my questionnaire said they were able to read the text on both the main menu and the leaderboards. Most people said they were able to read the text on the game customisation, game over and how to play. If i had more time to edit it to ensure everyone can read the text on the the how to play and game customisation I could make the text a bit larger because it can't be to do with the colour of the text compared to the colour of the background because they are the same as the main menu and leaderboard menu which everyone can read. To make the text readable to everyone on the game over menu I can change the text of the colour and background so it matches the other menus because the size of the text is slightly larger on this menu and it uses the same font however the colour scheme on this menu was slightly different.

Question 4 - I asked this question because it is important to understand if the user can actually control the game. 15 people said that they found the game easy to control and 2 people said they didn't. In a later question someone said they didn't know how to switch between buttons and i should add a how to explain at the start of the game so maybe the people that didn't know how to control the game didn't read the how to play or found it difficult to understand so if i had more time i could improve my game by making the how to play more simple and clear to understand.

Question 5 - I asked this question because I wanted to know if my game was too easy or too difficult for users. Only one person thought it was very difficult and only two people thought it was very easy. On average the difficulty of my game was 3.06 to three

significant figures which means that overall my game isn't too easy or too hard so i don't think anything needs to be changed about the difficulty of my game.

Question 6 - I asked this question because I wanted to know how successful my game was on average and a lot of games are ranked by their star rating . no one said that my game was 1 or 2 stars which means that my game was somewhat successful. Most of my stakeholders gave my game 4 stars which is a really good rating because I know my game does have a lot of things to improve on so couldn't be a five star game. On average my game is rated 3.76/5 to 3 significant figures but if i were to improve my game using my stakeholder feedback it may increase my average rating.

Question 7 - I asked this question because consistency is a key feature to any game you play for it to look more appealing and so then you don't confuse the users.

Everyone who answered my questionnaire said that the colour scheme was consistent throughout the game so I don't need to change anything about the colour scheme.

Question 8 - I asked this question because I wanted the colour scheme to match the theme of the game so the user doesn't get confused. Everyone said that the colour scheme matched the theme therefore the colour scheme I have chosen doesn't need to change.

Question 9 - I asked this question because games need a layout that is simple so that the user understands what is going on. If a user likes how the game is layed out, it means that the user isn't getting confused by the layout. Everyone but one person liked the layout of the game and in the things I needed to improve, the one person who said they didn't like the layout of the game didn't inform me how it can be improved to better suit them so nothing needs to be changed about the layout of the game.

Question 10 - I asked this question because you don't want the layout of the menu because you want people to be able to access it more easily and you want it to be accessible to all users. Everyone but one person likes the layout of the menu of my game which makes sense because everyone said the menu is easy to navigate. No improvements need to be made to how the menu is laid out.

Question 11 - I asked this question because the design of a game is just as important as the game play because you want the user to find the game appealing to the eye because otherwise they might not want to play the gaem.No one said that they didn't like any of the graphics, however for each of the different graphics categories someone disliked the graphics however, I had a time limit to make my game so I couldn't create the best graphics, so if i was to have more time I would improve the graphics for my game.

Question 12 - I asked this question because I wanted to know what people actually like about my game because with the other multiple choice questions it is difficult to actually understand what features are the favourite of the users and also if you created a giant questionnaire asking about every feature a user is less likely to want to answer it. Many people that answered my questionnaire said they like the music and sound effects that I

used within my game. People are also saying that they like the style of the game and they like how multiple different weapons can be used throughout the game. This means that if I had more time to edit my game I wouldn't need to edit the sound effects or the style of the game.

Question 13 - I asked this question because it's useful to see what people don't like about your game so that you can improve upon it to make the game fit your target market more easily. A few people said that the game was too easy on the first two difficulties so to improve on this I could have the number of enemies that appear on the screen increase as the player gets a higher score. Some people also said that they don't know which enemies get killed by which weapons so in the how to play I could add information about the effectiveness of each weapon against each enemy. There were also a few comments about the scoring system because people said that they didn't like how you could get a negative score so to fix this instead of decreasing the score by ten everytime you use the wrong weapon I can have some sort of count that decreases every time you use the wrong weapon so that people don't get a negative score and also people still can't spam loads of weapons in hopes one of them is correct.

Question 14 - I asked this question because I can think of ways to improve my game based on what my stakeholders don't like, however it is also useful to see how they think I can improve the game. A lot of people were saying to add information about which weapons are effective against which enemies and a few people said that they wanted the game to be a bit more difficult so if I had more time I would make the game difficulty increase as the game progresses and I would have a section on the how to play menu that informs the user of what weapon to use against each enemy.

Evaluation of Success Criteria

Below I have copied and pasted my success criteria and highlighted it to show what has been completed, partially completed and not completed. If the criteria is highlighted in green it means that I have completed that piece of criteria. If the criteria is highlighted in orange it means that that piece of criteria has not been finished. If the criteria is highlighted in red it means that that piece of criteria hasn't been included in the game at all.

- Auto scrolling screen
 - Screen that constantly scrolls at a constant speed - version 2
- Player
 - Jumps at a constant height when the space bar is pressed - version 3
 - Fires a weapon when the letter E is pressed that spawns in the middle of the player and moves at a constant speed. The size of all weapons are the same. - version 7, version 18

- Health that decreases by 10 when the wrong weapon is used, decreases by 10 when the player jumps over the enemies, goes to zero when the player collides with the enemy and is displayed on the screen in the form of a health bar - version 5, version 13,
- Enemy
 - Moves from the right to left at a constant speed - version 4
 - Is removed from the screen when the health is zero - version 7
 - When an enemy has died another will spawn on the right of the screen - version 7,
 - Another enemy reappears on the right if an enemy goes off the screen to the left - version 6
- Three different difficulty modes:
 - An easy mode that contains two different types of enemies that can randomly spawn when an enemy spawns - version 18
 - A medium mode that contains 5 different types of enemies that can spawn when an enemy spawns. - version 18
 - A hard mode that contains 9 different types of enemies that can spawn when an enemy spawns. - version 18
- Different types of enemies depending on the difficulty choice:
 - Easy - When an enemy spawns on the right of the screen it is going to be either a spirit or death - version 8
 - Medium - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch or an angel - version 18
 - Hard - When an enemy spawns on the right of the screen it is going to be either a spirit, death, a vampire, a witch, an angel, a demon, a werewolf, a hellhound or a leviathan. - version 18
- Menu with different options:
 - Name Input - there is a text box that the player is able to type their name into - version 12, version 14
 - Character Input - There is a dropdown box with the choices of Sam Winchester, Dean Winchester, Jack Klein, Bobby Singer, Rowena Macleod and Charlie Bradbury and the character that is chosen will be displayed on the screen when the user presses play. - version 19
 - Play - There will be a difficulty slider where the user can slide to difficulty 1, 2 or 3. 1 is easy, 2 is medium and 3 is hard. - version 18
 - Leaderboard - There will be a leaderboard button that takes the user to a menu that displays the leaderboards. The easy will be on the left, the medium will be in the middle and the hard will be on the right. - version 15, version 18

- Exit - There will be an exit button in the bottom right hand corner that will quit the game.- version 14
- Power - ups which you buy with coins:
 - Screen Slows Down - The screen starts going at a constant slower speed for 10 seconds and then goes back to its normal speed after the ten seconds.
- Every time your score reaches a multiple of 100 the screen moves faster
- Different Weapons
 - Scythe
 - Kills Death upon collision - version 9
 - Salt
 - Kills Spirits upon collision - version 9
 - Harms Hellhounds but never kills upon collision - version 18
 - Machete
 - Kills vampires upon collision - version 18
 - Kills Werewolves upon collision - version 18
 - Witch Bullet
 - Kills Witches upon collision - version 18
 - Angel Blade
 - Kills Angels upon collision - version 18
 - Kills Demons upon collision - version 18
 - Kills Werewolves upon collision - version 18
 - Kills Hellhounds upon collision - version 18
 - Demon Blade
 - Kills Demons upon collision - version 18
 - Kills Hellhounds upon collision - version 18
 - Silver Bullet
 - Kills Werewolves upon collision - version 18
 - Colt
 - Kills vampires upon collision - version 18
 - Kills Hellhounds upon collision - version 18
 - Kills Demons upon collision - version 18
 - Kills Werewolves upon collision - version 18
 - Bone
 - Kills Leviathans upon collision - version 18
 - Borax
 - Damages Leviathans (kills them eventually) upon collision - version 18
- Score
 - Increases by 100 when enemy killed - version 10

- Increase by 10 when player jumps over enemy - version 10
- Decrease by 10 when wrong bullet hits enemy - version 10
- Increases by 10 when some damage taken off enemy - version 18

- Sound

- Sound effects - different sound effects will happen when things happen in game so when the player jumps a player jump sound will play, when the player lands a player land sound will play, when the player fires a weapon a weapon fire sound will play, when the enemy dies an enemy died sound will play and when the player and enemy collide a player died sound will play - version 17
- Main game Music - When the player is playing the main game the main game music will play on loop in the background. - version 17
- Main Menu Music - When the player is on a menu the main menu music will play in the background on loop - version 17

Most of my success criteria is highlighted in green which shows that most of my initial plans for the game were implemented however, there are some reds and oranges so I wasn't able to add in everything.

Auto Scrolling Screen

I added this feature to my game in version 2 and it did work as shown by the tests however the game was going really slow because it was rendering too many graphics so I had to decrease the number of graphics that I was rendering which lead to my auto scrolling screen not being how I wanted it to be therefore I highlighted this criteria in orange.

Player

Jumping - Having the player be able to jump was a feature that my game would have been able to survive without if I wasn't able to implement it properly however in version 3 I added this feature to my game and the tests in version 3 shows that it worked how it was supposed to without needing to be edited later on.

Firing a weapon - This was an important feature to my game so I had to ensure that I got it right. I implemented and tested this in version 7 and it worked how it was supposed to, however for my game I need different weapons to kill different enemies so I had to edit this in version 18 when I added all the enemies and the tests show that the weapon firing worked in version 8 too.

Health - I added the health bar for the player in version 5 and I tested it and it worked and it was crucial that this works because the user needs to be able to see how much health they have left, however when I put the game loop within a procedure I noticed the health bar didn't work anymore so I had to edit the health bar in version 13 and it was tested and worked how it was supposed to and didn't need changing in any other versions after that.

Enemy

Movement - In version 4 I added the enemy into my game and the first thing i did was create the movement towards the player. This was important for my game so I had to ensure that this was implemented near the start and worked. The tests in this version show that the enemy movement worked as intended and I didn't need to edit the movement in later versions.

Respawn - In version 6 I added the feature of the enemy respawning when it went off the screen because if the player jumped over the enemy then no more enemies would respawn and the game would never end. I tested this feature within the game and it worked how it was meant to work and no further improvements needed to be made to this feature. Also in version 7 I worked on the respawning of the enemies when the enemy dies because when the enemy dies you want the enemy to be deleted off the screen and then respawn and in this version this also worked with no further improvements needed. This was crucial to get right because you want some way for the game to end.

Difficulties

Having different difficulties is one of the most important features of my game because some people might want to play an easier game than others so it was important that I got this feature right. I am working on the easy version all the way up to version 18 and all the tests have worked and then in version 18 because the only difference between the difficulties is the number of enemies I added in all the enemies in version 18 and when I did all my tests for the different enemies they all worked as expected.

Menu

I didn't create the menu how i planned originally but i have a series of menus instead. I added a name input in so that the user can input their name for the leaderboard and I tested this system within versions 12 and 14 and it worked so no improvements need to be made. For the character input I made sure that I had some choice of characters, however I didn't have enough time to add in all of the characters that I wanted to add. The character input was implemented and tested within version 19. If I had more time I would add the other hunters. For the difficulty selection I couldn't add the decision in until I had all of the different difficulties so I added this feature into version 18 and tested it and it worked how it was expected to so no change needed to be made. I created the leaderboard in version 15 and tested it and i had to tweak something in my code and then it worked and then in version 18 I finished off the leaderboards and tested them and they also worked as expected.I added the exit button on version 14 when adding validation and this wasn't in my original plan for the validation but i got it working anyway.

Power - Ups

This was a feature that I wanted in my game originally and so did my stakeholders however I didn't have enough time to implement this within my game so I had to

highlight it in red. If I had more time to improve my game I would definitely include power ups.

Weapons

In version 9 I added two of the weapons and tested them and they worked with no further improvements needed and then in version 18 when I added more enemies I needed more weapons so I added the rest of the weapons in version 18 and I tested all the different weapons on all the different enemies and it worked how it was expected to which was important because the weapon algorithm was the largest and one of the most important algorithms within my game.

Score

I added the score system within my game in version 10 and I tested it and it all worked as expected with no further improvements needed which is important because you want the scoring system to work correctly for the leaderboards so you can compete with other players. In version 18 I added to the code that controls the score because the increase in types of enemies had some enemies only have a certain amount of health go up so I also had to have the score go up according to that but I tested the new score increase within this version and everything worked how it was expected. If I had more time I would edit the score system so that it couldn't go into negatives because people didn't like that it could go into negative numbers.

Sound

I added all the sounds into version 17 and when I tested them they all played how they were meant to. If I had more time I would add in more sound effects like a running sound effect however no improvements were needed to be made when I tested this feature.

Further Development

If I had time to further develop my game there would be numerous features that I would like to add to my game or improve on based on the evaluation of the success criteria and the results from the final questionnaire.

Score - The score was able to go into negative numbers because in order for me to ensure people weren't spamming random weapons, if you used the wrong weapon then the score would decrease. In the questionnaire it became clear that people didn't like this happening so if I had more time I would have a weapon count of 5 so that you could only use the wrong weapon 5 different times so then the score would never decrease so therefore would never go into negative numbers and I am still ensuring that people don't spam random weapons in order to become top of the leaderboard.

How to Play - People have said that the how to play is wordy and they don't have all the information that they want to be able to play my game so if I had more time to

improve my game, I would decrease the length of the how to play and I would add in information about which weapons are effective against which enemies.

Power - ups - This is a feature that many people have said they would like within the original questionnaires however I didn't have time to implement it within my game so even though it wasn't in what i should improve in my final questionnaire, if I had more time I would add power ups to my game using an in-game currency to purchase the power-ups.

Difficulty - I have had responses that my game is too easy on the easier modes so if I had time to improve my game I would gradually increase the difficulty as the game progresses by increasing the maximum number of enemies on the screen at once and I would also use the power ups with this feature to decrease the maximum number of enemies on the screen. I would also implement bullet angles I.to my game if I had more time and have things like ghosts moving up and down and witches disappearing and reappearing to make my game slightly more difficult for users to play.

Maintainability

If you don't make your code maintainable it means that when you come back to it at a different time it could be difficult to read the code or understand what is going on.

Throughout my code to ensure it is easier to read I have added comments because it makes it easier to understand what is going on when it is written clearly as a comment. I have also sectioned out the code which made it easier to read because if you wanted to look for a certain class to edit it then they are all together in one section. Because all the code took a while to program it is also important to have meaningful variable names that explain what the variable is because then when you come back to the code at a later time you can easily find the variables that you want to use. I used this within my code so all of the variable names are clear descriptions of what the variables are. For example i have **enemyCount** for the count of how many enemies need to be displayed on the screen. With making my code maintainable it has ensured that I have been able to come back to the code and also it means that other people can look at the code and understand what is going on.

Another thing i did to ensure maintainability is that I documented the stages of my development which explains what is going on throughout and also when I added new things to the code I saved it as a new version so that if I couldn't get the new thing to work I could go back to the last version which would have been tested to see if it works so then I didn't have to start all over again.

I also added a length check on the name input so that people didn't put really long names or really short names and this makes my program more maintainable because it ensures that the scores of the top ten users can be displayed onto the screen and ait also means that the database doesn't take up too much storage by people including names that are really long.

Appendix

Final Code

Below is my final code.

```
## Importing the modules used throughout the code
from guizero import App, Box, Combo, Text, TextBox, PushButton, Picture, Window,
Slider, ListBox
import sqlite3
import pygame
import random
import time

pygame.init() # Initialising the modules used

Black = (0,0,0) # Creating the Colours
DramaticalRed = (156, 18, 0)
DwarfFortress = (30, 0, 0)
RedSentinel = (188, 7, 12)
DelightfulGreen = (0, 240, 0)
GameBoyContrast = (17, 59, 17)
White = (255, 255, 255)
RushmoreGrey = (181, 177, 165)
Silver = (192, 192, 192)
DeadPixel = (60, 57, 57)
BrilliantLiquorice = (84, 84, 84)
Nero = (38, 38, 38)
Skull = (227, 218, 201)
EmpirePorcelain = (223, 219, 211)

infoObject = pygame.display.Info() # Creating variables of the resolution of the screen
Screen = pygame.display.set_mode((infoObject.current_w, infoObject.current_h)) # Creating the canvas

pygame.display.set_caption('Supernatural:"Saving People Hunting Things") # Set game title

clock = pygame.time.Clock() # Defining pygame clock object

font = pygame.font.SysFont(None, 25) # Import the font
```

```

ScoreBoardFont = pygame.font.Font("fnt_HelpMe.ttf", 25) # Creatinge fonts used
throughout the program
WeaponFont = pygame.font.Font("fnt_HelpMe.ttf",50)

bg_img = pygame.image.load("pixil-frame-0.png") # Loading the image for the
background
bg_img = pygame.transform.scale(bg_img,(infoObject.current_w, infoObject.current_h))
# Scaling the image to fit the screen

# Importing the sounds used throughout the game
pygame.mixer.init() # to play sound
buttonPressedSound = pygame.mixer.Sound("Snd_buttonpressed.WAV") # Imports
sound
enemyDiedSound = pygame.mixer.Sound("Snd_dead.WAV")
jumpSound = pygame.mixer.Sound("Snd_jump.WAV")
landSound = pygame.mixer.Sound("Snd_land.WAV")
shootSound = pygame.mixer.Sound("Snd_shot.WAV")
gameMusic = pygame.mixer.music
menuMusic = pygame.mixer.music

weapon_list = pygame.image.load("spr_weapons.png") # Importing the weapon list
image that is displayed at the bottom of the screen
weapon_list = pygame.transform.scale(weapon_list, (infoObject.current_w,
(infoObject.current_h/489)*21)) # Scaling the image to fit the width of the screen
#####
#####CLASSES#####
#####

class Background():
    def __init__(self,image,speed):
        self.bgimage = pygame.image.load(image) # Loading the image
        self.bgimage = pygame.transform.scale(self.bgimage,((infoObject.current_w,
infoObject.current_h))) # Scaling the image to fit the size of the screen
        self.rectBGimg= self.bgimage.get_rect()

        self.bgY1 = 0
        self.bgX1 = 0

        self.bgY2 = 0
        self.bgX2 = self.rectBGimg.width # Setting the initial x an y coordinates of the two
background images

```

```

    self.movingSpeed = speed # setting the speed of the background

def update(self):
    self.bgX1 -= self.movingSpeed
    self.bgX2 -= self.movingSpeed
    if self.bgX1 <= -self.rectBGimg.width: #Resetting the coordinates of the image if it
goes off screen
        self.bgX1 = self.rectBGimg.width
    if self.bgX2 <= -self.rectBGimg.width: # Moving two different backgrounds because
one background is the size of the users screen so when it moves left you want another
one right next to it so that the background is covered at all times
        self.bgX2 = self.rectBGimg.width
def render(self):
    Screen.blit(self.bgimage, (self.bgX1, self.bgY1)) # Drawing the background to the
screen
    Screen.blit(self.bgimage, (self.bgX2, self.bgY2))

class Sprite(pygame.sprite.Sprite):

    def __init__(self,xCoordinate,yCoordinate,width,height,startImage):
        super().__init__()
        self.image = pygame.image.load(startImage) # Sets the size
        self.image = pygame.transform.scale(self.image,[width,height]) # scaling the image
to the width and height passed up
        self.rect = self.image.get_rect() # Hitbox
        self.rect.x = xCoordinate# Sets the position on the screen
        self.rect.y = yCoordinate
        self.health = 100

class Bar(pygame.sprite.Sprite):

    def __init__(self,xCoordinate,yCoordinate,width,height,colour):
        super().__init__()
        self.image = pygame.Surface([width,height]) # Sets the size
        self.image.fill(colour)
        self.rect = self.image.get_rect() # Hitbox
        self.rect.x = xCoordinate# Sets the position on the screen
        self.rect.y = yCoordinate

```

```

def update(self, Health):
    self.image = pygame.Surface([Health,10]) # Updating the width of the green health
bar so it changes size if the users health goes down
    self.image.fill(DelightfulGreen)

class Enemy(pygame.sprite.Sprite):

    def __init__(self,xCoordinate,yCoordinate,width,height,enemyspeed, image,
monster):
        super().__init__()
        self.image = pygame.image.load(image)
        self.image = pygame.transform.scale(self.image,[width,height]) # Sets the size
        self.rect = self.image.get_rect() # Hitbox

        self.rect.x = xCoordinate# Sets the position on the screen
        self.rect.y = yCoordinate
        self.speed = enemyspeed
        self.monster = monster
        self.health = 100

    def update(self):
        self.rect.x-=self.speed # Moves the enemy left

class Weapon(pygame.sprite.Sprite):

    def __init__(self,xCoordinate,yCoordinate,width,height,colour, weapon):
        super().__init__()
        self.image = pygame.Surface([width,height]) # Sets the size
        self.image.fill(colour)
        self.rect = self.image.get_rect() # Hitbox
        self.rect.x = xCoordinate# Sets the position on the screen
        self.rect.y = yCoordinate

        self.weapon = weapon

    def update(self):
        self.rect.x += 5 # Moves the bullet right by five pixels

def ScoreText(Text, Colour, xCoordinate, yCoordinate): # Text procedure
    ScreenText = ScoreBoardFont.render(Text, True, Colour)

```

```

Screen.blit(ScreenText,[xCoordinate, yCoordinate])

def WeaponChoiceText(Text, Colour): # Text procedure
    ScreenText = WeaponFont.render(Text, True, Colour)
    textWidth = ScreenText.get_width()
    Screen.blit(ScreenText,[infoObject.current_w/2 - textWidth/2, infoObject.current_h/4])
#####
#####
#####
Player= Sprite(infoObject.current_w/4,
(infoObject.current_h*(107/144))-30,30,30,"spr_dean.png") # Creating the defauult
player object

ground = Background("Bgd_path.png", 5)
#####
#####GAMELOOP#####
#####
#####

def Game(Name, Difficulty, Hunter):
    i = 0 # Creating the variables used for the game
    Jump = False
    jumpCount = 20
    score = 0
    EnemyCount = 1
    playerweapon = "Scythe"
    weaponcolour = (0,0,0)
    if Difficulty ==1:      # Setting the number of types of enemies based on the
difficulty chosen by the user
        typesOfEnemies = 2
    elif Difficulty == 2:
        typesOfEnemies = 5
    elif Difficulty == 3:
        typesOfEnemies = 9

    if Hunter == "Dean Winchester": # Creating the image based on what hunter was
chosen
        startImage = "spr_dean.png"
        imageList = ["spr_dean.png", "spr_dean2.png", "spr_dean.png", "spr_dean3.png"]
# Creating an array of images to implement character animation (didn't have enough
time to complete)
    if Hunter == "Sam Winchester":

```

```

startImage = "spr_sam.png"
imageList = ["spr_sam.png", "spr_sam2.png", "spr_sam.png", "spr_sam3.png"]

#####
#####OBJECTS#####
#####

Player= Sprite(infoObject.current_w/4, (infoObject.current_h*(107/144))-30,30,30,
startImage) # Creating the objects
HealthBarPt1 = Bar(10,10,100,10,DramaticalRed)
HealthBarPt2 = Bar(10,10,Player.health,10,DelightfulGreen)

SpriteGroup = pygame.sprite.Group() # Create a group
EnemyGroup = pygame.sprite.Group()
HealthBarGroup = pygame.sprite.Group()
WeaponGroup = pygame.sprite.Group()

SpriteGroup.add(Player) # Add object to group
HealthBarGroup.add(HealthBarPt1)
HealthBarGroup.add(HealthBarPt2)

def AddToLeaderboard(Name,Score,Difficulty): # Creating the add to leaderboard
procedure
    con = sqlite3.connect('Leaderboard.db') # Opening the leaderboard
    cursorObj = con.cursor()

    entities = (Name, Score, Difficulty)
    cursorObj.execute('INSERT INTO UsersScores (Name, Score, Difficulty)
VALUES(?, ?, ?)',entities) # Adding the data into the leaderboard

    con.commit()

    cursorObj.close()

#####
#####GAMELOOP#####
#####

gameExit = False
pressed = False
while not gameExit:

```

```

for event in pygame.event.get():

    if event.type == pygame.QUIT: # Event quit
        gameExit = True
        pygame.quit()
        quit()

    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_ESCAPE: # Quit if ESC key pressed
            gameExit = True
            pygame.quit()
            quit()
        if event.key == pygame.K_e:
            shootSound.play() # Playing shoot sound
            weaponObject =
                Weapon(Player.rect.centerx,Player.rect.centery,20,5,weaponcolour, playerweapon) #
                Firing weapon
                WeaponGroup.add(weaponObject) # Add weapon to weapon group
        if event.key == pygame.K_1:
            playerweapon = "Scythe" # setting the weapon and the colour of the
            weapon based on the the number pressed
            weaponcolour = Black
        if event.key == pygame.K_2:
            playerweapon = "Salt"
            weaponcolour = White
        if event.key == pygame.K_3:
            playerweapon = "Machete"
            weaponcolour = DeadPixel
        if event.key == pygame.K_4:
            playerweapon = "WitchBullet"
            weaponcolour = GameBoyContrast
        if event.key == pygame.K_5:
            playerweapon = "AngelBlade"
            weaponcolour = RushmoreGrey
        if event.key == pygame.K_6:
            playerweapon = "DemonBlade"
            weaponcolour = BrilliantLiquorice
        if event.key == pygame.K_7:
            playerweapon = "SilverBullet"

```

```

        weaponcolour = Silver
    if event.key == pygame.K_8:
        playerweapon = "Colt"
        weaponcolour = Nero
    if event.key == pygame.K_9:
        playerweapon = "Bone"
        weaponcolour = Skull
    if event.key == pygame.K_0:
        playerweapon = "Borax"
        weaponcolour = EmpirePorcelain

keys = pygame.key.get_pressed()
if not(Jump):
    if keys[pygame.K_SPACE]:
        jumpSound.play()
        Jump = True
else:
    if jumpCount >=-20:
        Player.rect.y -= jumpCount
        jumpCount -= 2
    else: # This will execute if our jump is finished
        landSound.play()
        jumpCount = 20
        Jump = False
        Player.rect.y = (infoObject.current_h*(107/144))-30
        # Resetting our Variables

if EnemyCount>0:
    enemyChoice = random.randint(1,typesOfEnemies) # random number generator
so that a random enemy spawns and then a giant if statement setting the enemy object
depending on the number from the random number generator
    if enemyChoice == 1:
        enemy = Enemy(infoObject.current_w,
(infoObject.current_h*(107/144))-50,30,50,10, "spr_ghost.png", "spirit")
    if enemyChoice == 2:
        enemy = Enemy(infoObject.current_w,
(infoObject.current_h*(107/144))-70,30,70,10, "spr_death.png", "death")
    if enemyChoice == 3:
        enemy = Enemy(infoObject.current_w,
(infoObject.current_h*(107/144))-50,30,50,10, "spr_angel.png", "angel")

```

```

if enemyChoice == 4:
    enemy = Enemy(infoObject.current_w,
(infoObject.current_h*(107/144))-50,30,50,10, "spr_vampire.png", "vampire")
if enemyChoice == 5:
    enemy = Enemy(infoObject.current_w,
(infoObject.current_h*(107/144))-50,30,50,10, "spr_witch.png", "witch")
if enemyChoice == 6:
    enemy = Enemy(infoObject.current_w,
(infoObject.current_h*(107/144))-50,30,50,10, "spr_leviathan.png", "leviathan")
if enemyChoice == 7:
    enemy = Enemy(infoObject.current_w,
(infoObject.current_h*(107/144))-50,30,50,10, "spr_hellhound.png", "hellhound")
if enemyChoice == 8:
    enemy = Enemy(infoObject.current_w,
(infoObject.current_h*(107/144))-50,30,50,10, "spr_werewolf.png", "werewolf")
if enemyChoice == 9:
    enemy = Enemy(infoObject.current_w,
(infoObject.current_h*(107/144))-70,20,70,10, "spr_demon.png", "demon")
    EnemyGroup.add(enemy)
    EnemyCount -=1

for eachEnemy in EnemyGroup:
    Enemyhits = pygame.sprite.spritecollide(eachEnemy,SpriteGroup, True)
    if eachEnemy.rect.x<0: # Checking if the enemy goes off screen
        EnemyGroup.remove(eachEnemy)
        EnemyCount+=1
    if eachEnemy.rect.centerx <= Player.rect.centerx + 5 and
eachEnemy.rect.centerx >= Player.rect.centerx -1: # Checking if the player has jumped
over the enemy
        score +=10
        Player.health-=10
    if Enemyhits: # Checking if the enemy and player collide
        Player.health= 0

for eachWeapon in WeaponGroup:
    if eachWeapon.rect.x == infoObject.current_w: # Giant if statement to check if
the correct weapon is getting used against the correct enemy
        WeaponGroup.remove(eachWeapon)#Kills the Weapon if it goes off screen
    if eachWeapon.weapon == "Scythe" and enemy.monster == "death":

```

```

WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup, True)
#Allows the Weapons to collide
    if WeaponHits:
        enemyDiedSound.play()
        EnemyCount += 1
        score+=100
    elif eachWeapon.weapon == "Salt" and enemy.monster == "spirit":
        WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup, True)
#Allows the Weapons to collide
        if WeaponHits:
            enemyDiedSound.play()
            EnemyCount += 1
            score+=100
        elif eachWeapon.weapon == "AngelBlade" and enemy.monster == "angel":
            WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup, True)
#Allows the Weapons to collide
            if WeaponHits:
                enemyDiedSound.play()
                EnemyCount += 1
                score+=100
            elif (eachWeapon.weapon == "Machete" or eachWeapon.weapon == "Colt") and
enemy.monster == "vampire":
                WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup, True)
#Allows the Weapons to collide
                if WeaponHits:
                    enemyDiedSound.play()
                    EnemyCount += 1
                    score+=100
                elif eachWeapon.weapon == "WitchBullet" and enemy.monster == "witch":
                    WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup, True)
#Allows the Weapons to collide
                    if WeaponHits:
                        enemyDiedSound.play()
                        EnemyCount += 1
                        score+=100
                    elif enemy.monster == "leviathan" and (eachWeapon.weapon == "Bone" or
eachWeapon.weapon == "Borax"):
                        if eachWeapon.weapon == "Bone":
                            WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup,
True) #Allows the Weapons to collide

```

```

if WeaponHits:
    enemyDiedSound.play()
    EnemyCount += 1
    score+=100
if eachWeapon.weapon == "Borax":
    if enemy.health > 10:
        WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup,
False) #Allows the Weapons to collide
        if WeaponHits:
            enemy.health -=10
            score+=10
    elif enemy.health <=10:
        WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup,
True) #Allows the Weapons to collide
        if WeaponHits:
            enemy.health -=10
            score+=10
            EnemyCount+=1
    elif enemy.monster == "hellhound" and (eachWeapon.weapon == "AngelBlade"
or eachWeapon.weapon == "DemonBlade" or eachWeapon.weapon == "Colt" or
eachWeapon.weapon == "Salt"):
        if eachWeapon.weapon == "AngelBlade" or eachWeapon.weapon ==
"DemonBlade" or eachWeapon.weapon == "Colt":
            WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup,
True) #Allows the Weapons to collide
            if WeaponHits:
                enemyDiedSound.play()
                EnemyCount += 1
                score+=100
        if eachWeapon.weapon == "Salt":
            if enemy.health > 10:
                WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup,
False) #Allows the Weapons to collide
                if WeaponHits:
                    enemy.health -=10
                    score+=10
            else:
                BulletHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup,
False) #Allows the bullets to collide
                if BulletHits:

```

```

        score-=10
    elif (eachWeapon.weapon == "Machete" or eachWeapon.weapon ==
"AngelBlade" or eachWeapon.weapon == "SilverWeapon" or eachWeapon.weapon ==
"Colt") and enemy.monster == "werewolf":
        WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup, True)
#Allows the Weapons to collide
    if WeaponHits:
        enemyDiedSound.play()
        EnemyCount += 1
        score+=100
    elif (eachWeapon.weapon == "AngelBlade" or eachWeapon.weapon ==
"DemonBlade" or eachWeapon.weapon == "Colt") and enemy.monster == "demon":
        WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup, True)
#Allows the Weapons to collide
    if WeaponHits:
        enemyDiedSound.play()
        EnemyCount += 1
        score+=100
    else:
        WeaponHits = pygame.sprite.spritecollide(eachWeapon, EnemyGroup, False)
#Allows the Weapons to collide
    if WeaponHits:
        score-=10
    if WeaponHits:
        WeaponGroup.remove(eachWeapon)

Screen.fill(DwarfFortress)

ground.update()
ground.render() # running the background methods for the auto scrolling
background

SpriteGroup.draw(Screen)
EnemyGroup.draw(Screen)
HealthBarGroup.draw(Screen)
WeaponGroup.draw(Screen)
HealthBarPt2.update(Player.health)
EnemyGroup.update()
WeaponGroup.update()

```

```

    Screen.blit(weapon_list,(1, infoObject.current_h-((infoObject.current_h/489)*21))) #
Displaying the weapon list on the bottom of the screen

    clock.tick(30) # setting the FPS

    ScoreText("Score:",White,10,30)
    ScoreText(str(score),White,110,30) # Displaying the score onto the screen
    WeaponChoiceText(playerweapon,White) # Displaying the equipped weapon on
the screen

if Player.health == 0:
    SpriteGroup.remove()
    EnemyGroup.remove(eachEnemy)
    AddToLeaderboard(Name, score, Difficulty) # Adding the score to the
leaderboard
    gameExit = True
    GameOver(Name, score)
    pygame.display.update() # ensuring all the objects update on the screen after
every loop
#####
#####MENU#####
#####

def customGame(): # Procedure to hide the main menu and show the customisation
menu
    buttonPressedSound.play()
    Menu.hide()
    Customiser()

def loadHowToPlay(): # procedure to hide the main menu and display the how to play
menu
    buttonPressedSound.play()
    Menu.hide()
    HowToPlay()

def endgame(): # procedure to end the game
    buttonPressedSound.play()
    endchoice = Menu.yesno("QUIT?", "Are you sure you want to quit?")
    if endchoice == True:
        Menu.warn("Thanks for Playing", "Thank you for playing Supernatural:\\\"Saving
People, Hunting Things\\\"")
        Menu.destroy()

```

```

pygame.quit()

def ShowLeaderboard(): # procedure to hide the main menu and display the
leaderboard menu
    buttonPressedSound.play()
    Menu.hide()
    displayLeaderboard()

Menu = App(title="Supernatural: Saving People Hunting Things", bg = Black) # Creating
the main menu
Menu.full_screen = True

MenuHeight = Menu.height
MenuWidth = Menu.width

TitleBox = Box(Menu, width="fill", align="top") # Creating the title box
Background = Picture(Menu, image = "bgd_menu.PNG")

ButtonBox = Box(Menu, width = int(MenuWidth/4), height="fill") # creating all of the
buttons displayed on the main menu
button = PushButton(ButtonBox, text ="Play", width = "fill", command=customGame)
button.text_size = int(MenuHeight/30)
button.text_color = RedSentinel
button = PushButton(ButtonBox, text="How To Play", width = "fill",
command=loadHowToPlay)
button.text_size = int(MenuHeight/30)
button.text_color = RedSentinel
button = PushButton(ButtonBox, text="Leaderboards", width = "fill", command =
ShowLeaderboard)
button.text_size = int(MenuHeight/30)
button.text_color = RedSentinel

ExitBox = Box(Menu, align="bottom", width = "fill") # creating the exit box
button = PushButton(ExitBox, text ="QUIT", align = "right",command = endgame)
button.text_size = int(MenuHeight/30)
button.text_color = RedSentinel
#####
#####
#####
#####
def Customiser(): # procedure for the customisation menu

```

```

def GameInformation(): # procedure to hide the customisation menu and display the game
    buttonPressedSound.play()
    Name = NameInput.value
    Difficulty = DifficultyInput.value
    Hunter = HunterInput.value
    print(Name)
    if len(Name) < 3:
        Customiser.error("Name is too short", "The name you have entered is too short.  
Your name must be 3 characters - 10 characters long.")
    elif len(Name) > 10:
        Customiser.error("Name is too long", "The name you have entered is too long.  
Your name must be 3 characters - 10 characters long.")
    else:
        Customiser.hide()
        gameMusic.load("Snd_GameMusic.WAV")
        gameMusic.play(-1)
        Game(Name, Difficulty, Hunter)

def Back(): # procedure to hide the customisation menu and load the main menu
    buttonPressedSound.play()
    Customiser.hide()
    Menu.show()
    Customiser = Window(Menu, title = "Customisation Menu", bg = Black) # creating the customisation menu
    Customiser.full_screen = True

    CustomiserHeight = Customiser.height
    CustomiserWidth = Customiser.width

    TitleBox2 = Box(Customiser, width="fill", align="top") # creating the title box
    Background2 = Picture(Customiser, image = "bgd_menu.PNG")

    InputsBox = Box(Customiser, width = int(Customiser.width/4), height = "fill") # taking in all of the inputs from the user
    NameText = Text(InputsBox, text = "Enter your name:")
    NameText.text_color = White
    NameText.text_size = 15

    NameInput = TextBox(InputsBox, width = "fill")

```

```

NameInput.text_color = White
NameInput.text_size = 15

DifficultyText = Text(InputsBox, text = "Difficulty:")
DifficultyText.text_color = White
DifficultyText.text_size = 15
DifficultyInput = Slider(InputsBox, start = 1, end = 3, width = "fill")
DifficultyInput.text_color = White
DifficultyInput.text_size = 15

HunterText = Text(InputsBox, text = "Hunter:")
HunterText.text_color = White
HunterText.text_size = 15
HunterInput = Combo(InputsBox, options = ["Dean Winchester", "Sam Winchester"],
width = "fill")
HunterInput.text_color = White
HunterInput.text_size = 15

StartButton = PushButton(InputsBox, text = "START", width = "fill", command =
GameInformation) # creating the button to start the game
StartButton.text_size = int(MenuHeight/30)
StartButton.text_color = White

BackBox = Box(Customiser, align="bottom", width = "fill") # creating the button to go
back to main menu
button = PushButton(BackBox, text ="BACK", align = "right", command = Back)
button.text_size = int(MenuHeight/30)
button.text_color = RedSentinel
#####
#####
def HowToPlay(): # creating the how to play procedure
    def BackToMain(): # creating a procedure that hides the how to play menu and shows
the main menu
        buttonPressedSound.play()
        HowToPlay.hide()
        Menu.show()
    HowToPlay = Window(Menu, title = "Customisation Menu", bg = Black) # creating the
how to play menu
    HowToPlay.full_screen = True

```

```

HowToPlayHeight = HowToPlay.height
HowToPlayWidth = HowToPlay.width

TitleBox3 = Box(HowToPlay, width="fill", align="top")
Background2 = Picture(TitleBox3, image = "bgd_menu.PNG")

HowToPlay1 = Picture(HowToPlay, image = "menu_UseCustomisation.PNG", align =
"left") # displaying the images that instruct the user on how to play
HowToPlay2 = Picture(HowToPlay, image = "menu_PlayMainGame.PNG", align =
"left")

BackBox = Box(HowToPlay, align="bottom", width = "fill") # creating the button that
takes the user back to the main menu
button = PushButton(BackBox, text ="BACK", align = "right", command =
BackToMain)
button.text_size = int(MenuHeight/30)
button.text_color = RedSentinel
#####
#####
def GameOver(Name, Score): # creating the game over menu procedure
    def MainMenu(): # reating a procedure that hides the game over menu and displays
the main menu
        buttonPressedSound.play()
        GameOver.hide()
        menuMusic.load("Snd_MenuMusic.WAV")
        menuMusic.play(-1)
        Menu.show()

ScoreText = (Name,"your score is", Score) # creating the string to display the score

GameOver = Window(Menu, title = "Game Over", bg = DwarfFortress)
GameOver.full_screen = True
GameOverText = Text(GameOver, text = "Game Over!", width = "fill", height = "fill")
GameOverText.text_size = 100
GameOverText.text_color = White
GameOverTextScore = Text(GameOver, text = ScoreText, width = "fill", height = "fill")
# Displaying the score
GameOverTextScore.text_size = 50
GameOverTextScore.text_color = White

```

```

GameOverButton = PushButton(GameOver , text="Go back to Main Menu",
command = MainMenu) # creating the button that takes the user back to the main menu
    GameOverButton.text_size = 100
    GameOverButton.text_color = White
#####
#####
def displayLeaderboard(): # creating the leaderboard menu procedure
    def BackToMain(): # creating a procedure that hides the leaderboard menu and
displays the main menu
        buttonPressedSound.play()
        Leaderboard.hide()
        Menu.show()

con = sqlite3.connect('Leaderboard.db') # opeing the database that stores the users
scores
cursorObj = con.cursor()

Leaderboard = Window(Menu, title="Supernatural: Saving People Hunting Things",
bg = Black)
Leaderboard.full_screen = True

LeaderboardHeight = Leaderboard.height
LeaderboardWidth = Leaderboard.width

TitleBox = Box(Leaderboard, width="fill", align="top")
Background = Picture(TitleBox, image = "Bgd_Leaderboard.PNG", width =
LeaderboardWidth)

EasyBox = Box(Leaderboard, width = int(LeaderboardWidth/3),
height=int(LeaderboardWidth*0.8), align = "left")
EasyBoxNumbers = Box(EasyBox,width = int(LeaderboardWidth/24), height="fill",
align = "left") # changed width of numbers box and stats box
for i in range (1,11): # for loop that displays the numbers 1-10
    PlaceText = Text(EasyBoxNumbers, text = i)
    PlaceText.text_size = int(LeaderboardHeight/30)
    PlaceText.text_color = White
strSQLEasy = "SELECT Name, Score FROM UsersScores WHERE Difficulty = 1
ORDER BY Score DESC LIMIT 10" # creating a string of the top 10 easy scores
cursorObj.execute(strSQLEasy)

```

```

rows = cursorObj.fetchall()

EasyBoxStats = Box(EasyBox, width = int(LeaderboardWidth/3), height="fill", align =
"left")
for row in rows: # Iterating through the easy scores string and displaying it
    EasyText = Text(EasyBoxStats, text = row)
    EasyText.text_size = int(LeaderboardHeight/30)
    EasyText.text_color = White

MediumBox = Box(Leaderboard, width = int(LeaderboardWidth/3),
height=int(LeaderboardWidth*0.8), align = "left")
MediumBoxNumbers = Box(MediumBox, width = int(LeaderboardWidth/24),
height="fill", align = "left")
for i in range (1,11): # for loop that displays the numbers 1 to 10
    PlaceText = Text(MediumBoxNumbers, text = i)
    PlaceText.text_size = int(LeaderboardHeight/30)
    PlaceText.text_color = White
strSQLMedium = "SELECT Name, Score FROM UsersScores WHERE Difficulty = 2
ORDER BY Score DESC LIMIT 10" # creating a string that holds the top 10 medium
scores
cursorObj.execute(strSQLMedium)
rows = cursorObj.fetchall()

MediumBoxStats = Box(MediumBox, width = int(LeaderboardWidth/3), height="fill",
align = "left")
for row in rows: # iterating through the medium scores string and displaying it to the
screen
    MediumText = Text(MediumBoxStats, text = row)
    MediumText.text_size = int(LeaderboardHeight/30)
    MediumText.text_color = White

HardBox = Box(Leaderboard, width = int(LeaderboardWidth/3),
height=int(LeaderboardWidth*0.8), align = "left")
HardBoxNumbers = Box(HardBox, width = int(LeaderboardWidth/24), height = "fill",
align = "left")
for i in range (1,11): # for loop to display the numbers 1 to 10
    PlaceText = Text(HardBoxNumbers, text = i)
    PlaceText.text_size = int(LeaderboardHeight/30)
    PlaceText.text_color = White

```

```

strSQLHard = "SELECT Name, Score FROM UsersScores WHERE Difficulty = 3
ORDER BY Score DESC LIMIT 10" # creating a string with the top 10 hard difficulty
scores
cursorObj.execute(strSQLHard)
rows = cursorObj.fetchall()

HardBoxStats = Box(HardBox, width = int(LeaderboardWidth/3), height = "fill", align =
"left")
for row in rows: # iterating through the hard score string and displaying it onto the
screen
    HardText = Text(HardBoxStats, text = row)
    HardText.text_size = int(LeaderboardHeight/30)
    HardText.text_color = White

button = PushButton(HardBoxStats, text ="QUIT", align ="right", command =
BackToMain) # creating the button that takes the user back to the main menu
button.text_size = int(LeaderboardHeight/30)
button.text_color = White

cursorObj.close() # closing the database that stores the scores
#####
#####
menuMusic.load("Snd_MenuMusic.WAV") # playing the main menu music
menuMusic.play(-1) # looping the music
Menu.display() # displaying the main menu

```