Programming Project Evaluation and Final Code

By Farhan Mirza

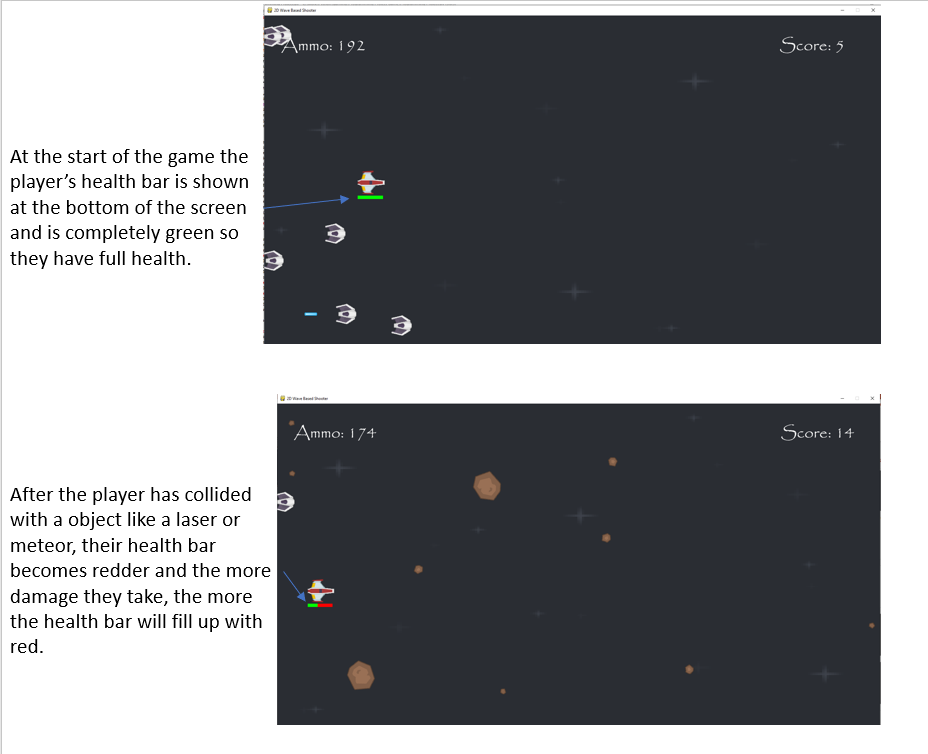
Success Criteria Achieved

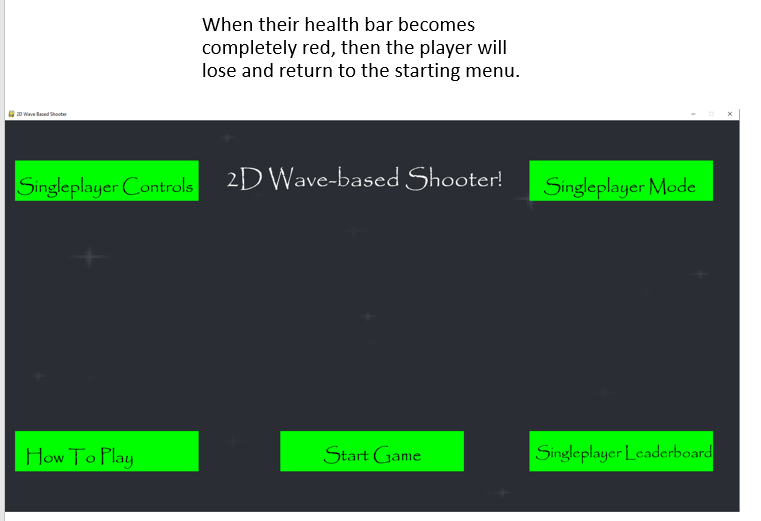
|  |  |
| --- | --- |
| **Functional Requirements** | **Was it achieved?** |
| Use a health system in the game that makes the player lose health when they collide with hazards | It was fully achieved. |
| Make the game playable with another player | It was fully achieved. |
| The game will also need to fit two sets of controls on one keyboard | It was fully achieved. |
| Introduce multiple different enemy types, including bosses, overtime as the game progresses | It was fully achieved. |
| Use a scoring system for the enemies destroyed and record the score and time the player survived for | Recording the score was achieved, but recording the time wasn’t achieved since the client didn’t think it would be useful information to have on a leaderboard. |
| Have an interactive User Interface that lets players start a game, check the high scores and check the controls | It was fully achieved. |
| Have a consistent design and layout | It was fully achieved. |
| Allow the user to choose the height and width of the screen | This was changed to basing the size of the game screen on the resolution of the monitor in order to simplify this process for the user. |
| Include powerups (like extra health and ammo) in the game | It was fully achieved. |

|  |  |
| --- | --- |
| **Non-functional Requirements** | **Was it achieved?** |
| Allow a user to play with another player online | This was not achieved. |
| Enable controller support for the game | It was fully achieved. |
| Include sound effects for lasers, collisions, getting powerups etc. | It was fully achieved. |
| Have the background of the game change overtime | It was fully achieved. |
| Have an AI player that can fight alongside the player. | This was not achieved. |

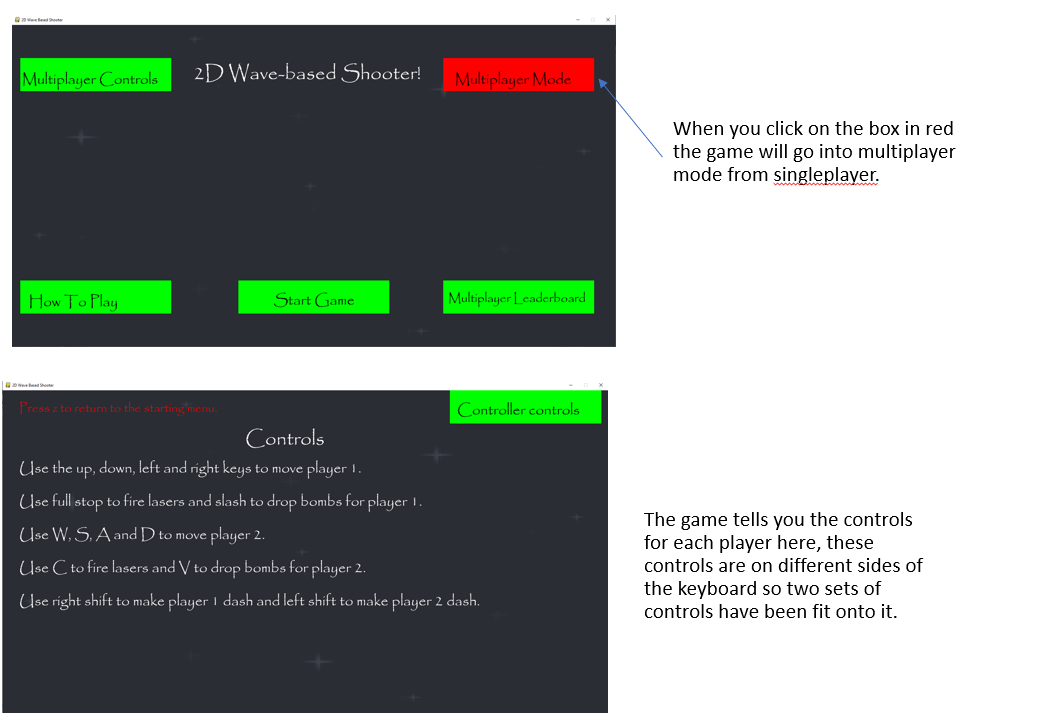
Evidence for the Success Criteria

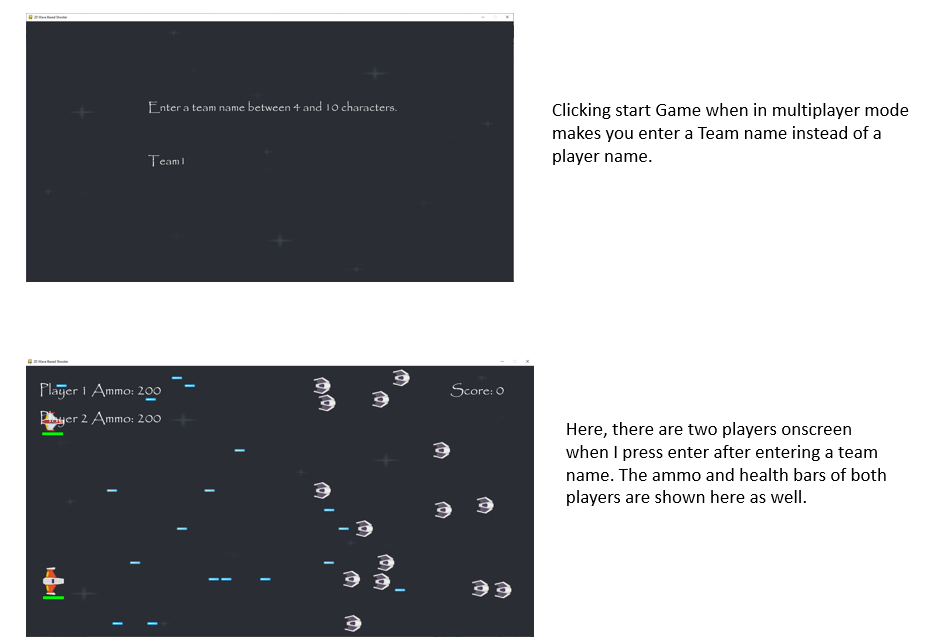
Evidence of a health bar and health loss:

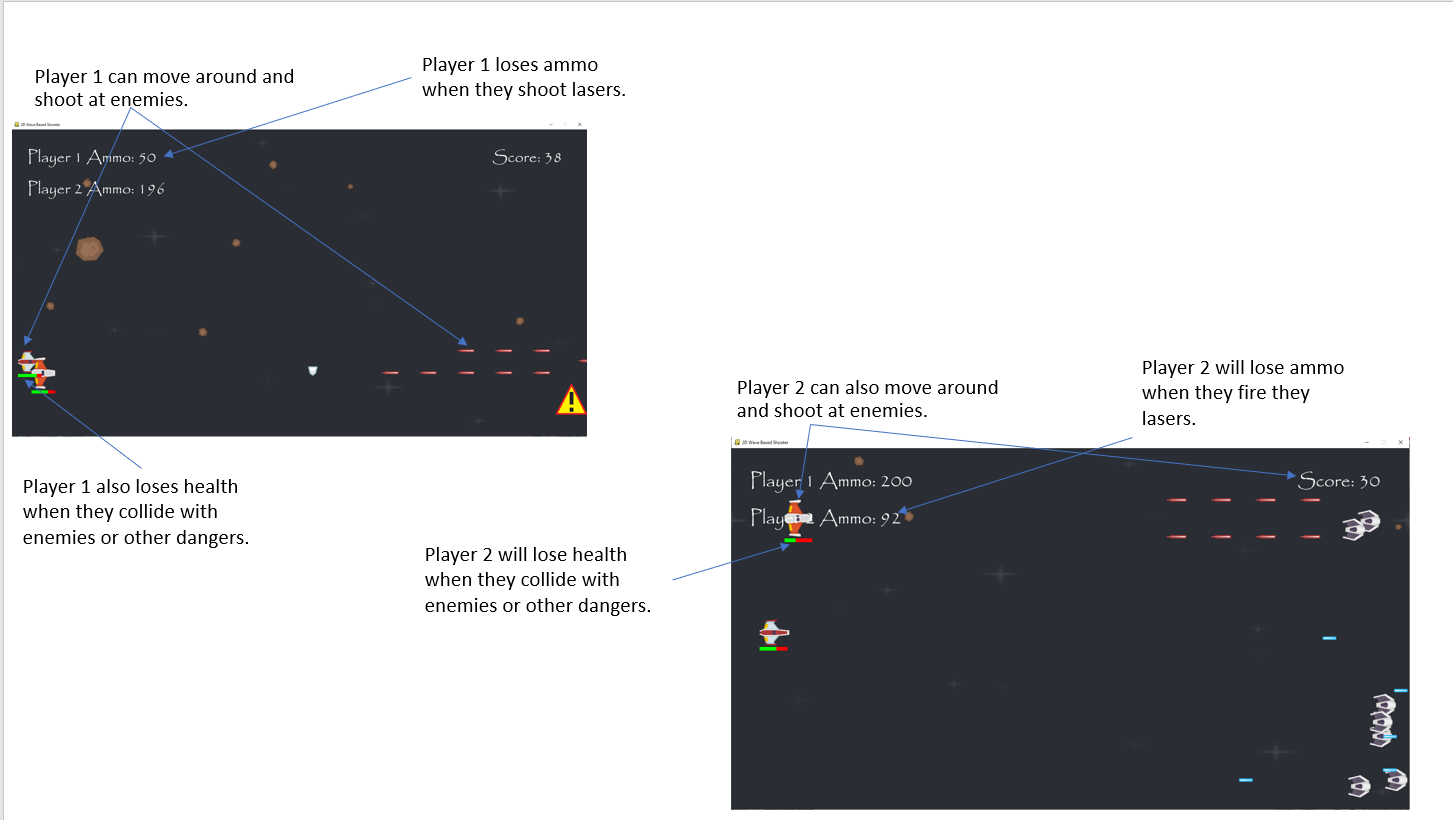




Evidence of multiplayer functionality:

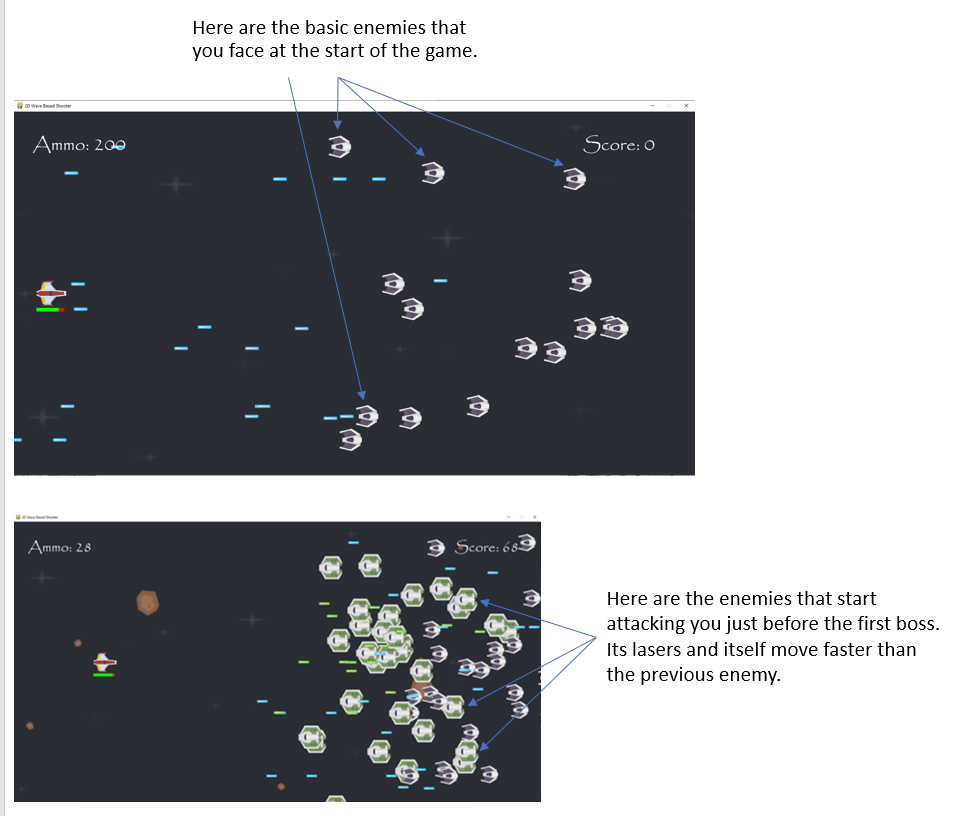






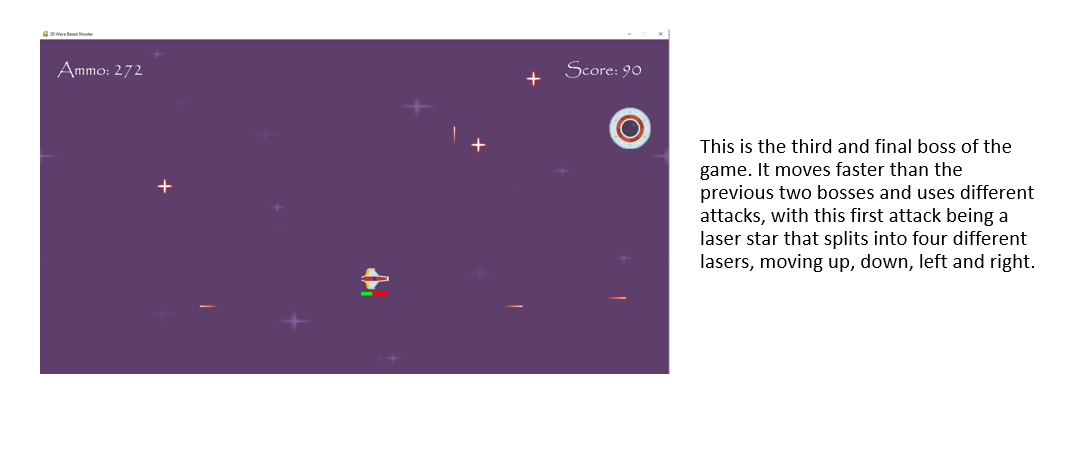


Evidence of Different Enemy Types and Bosses:

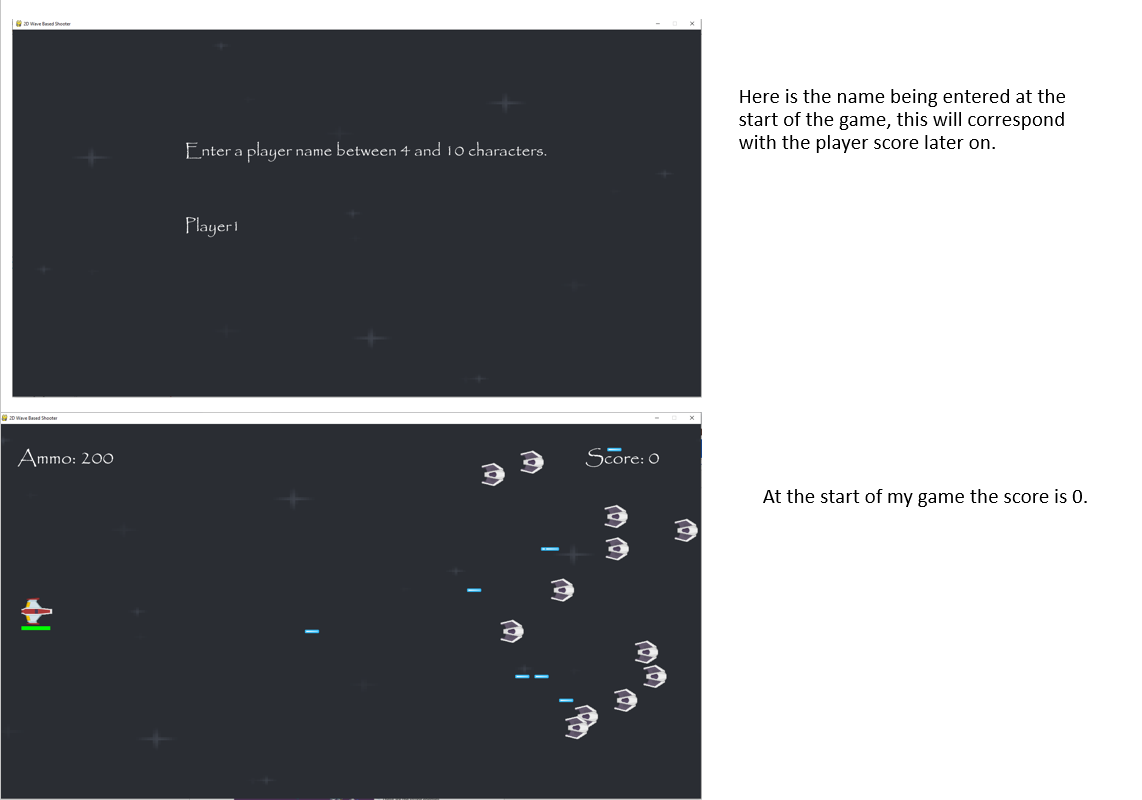


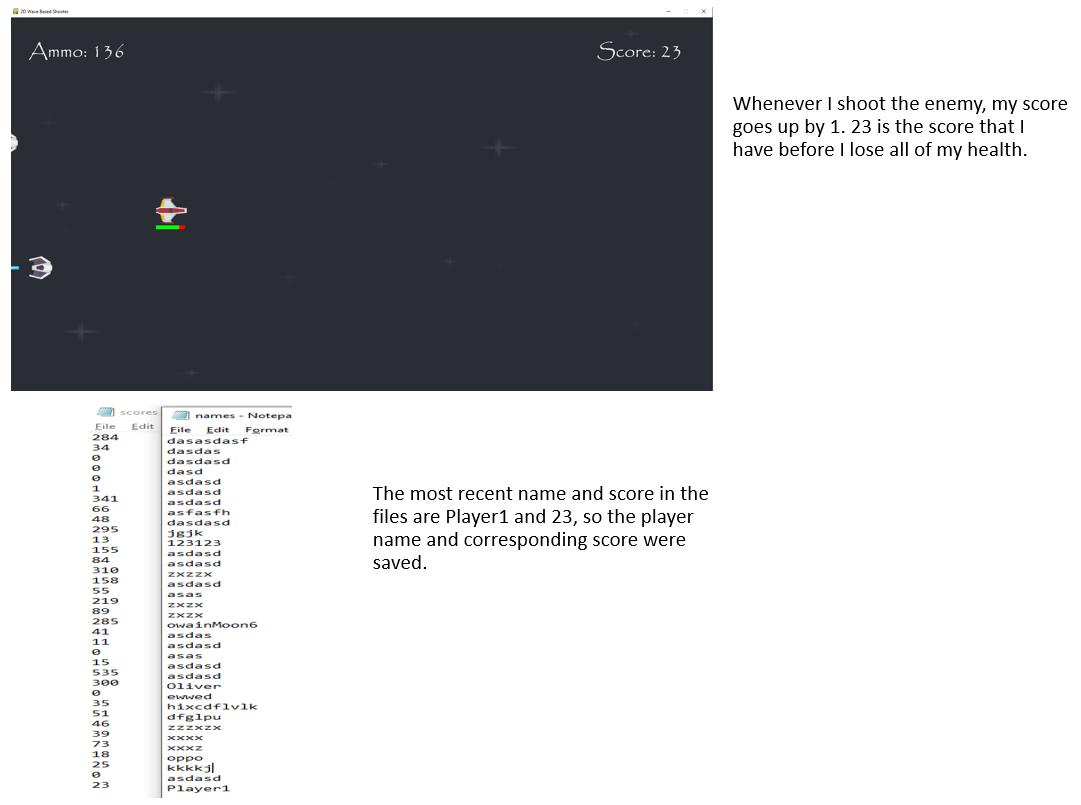


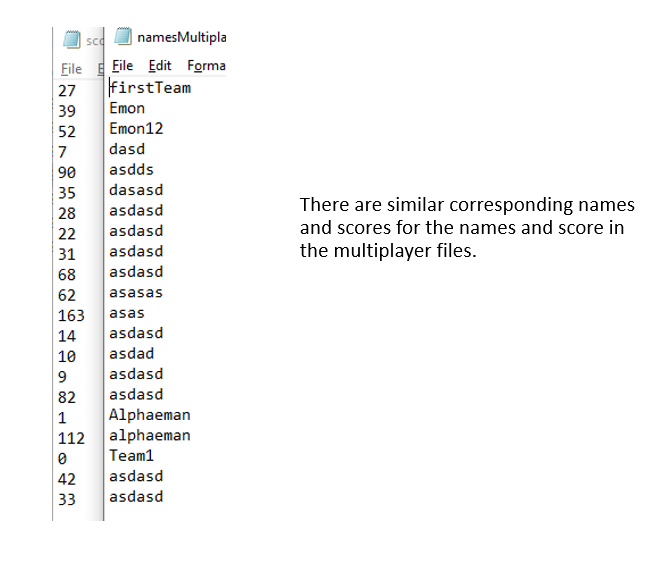




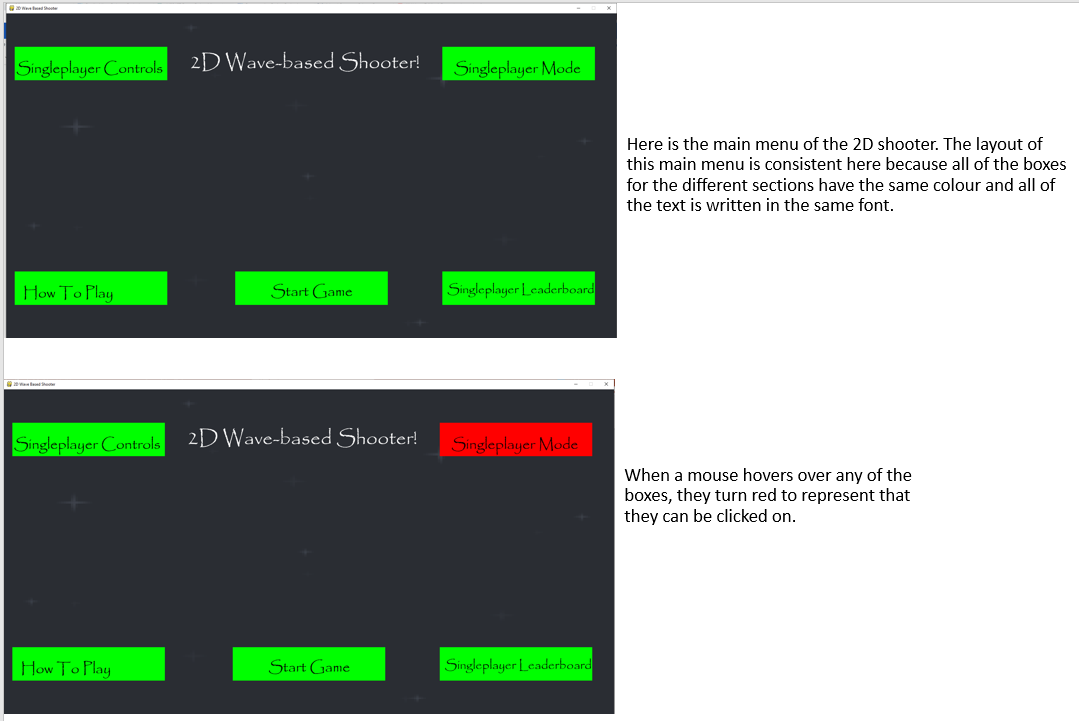
Evidence of a scoring system:

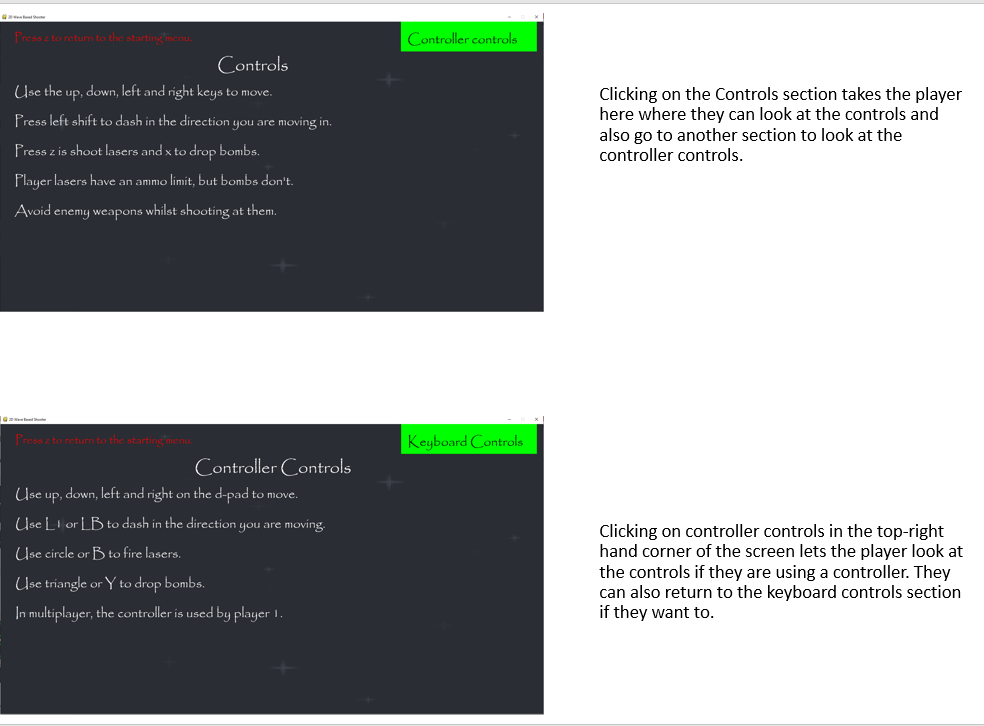


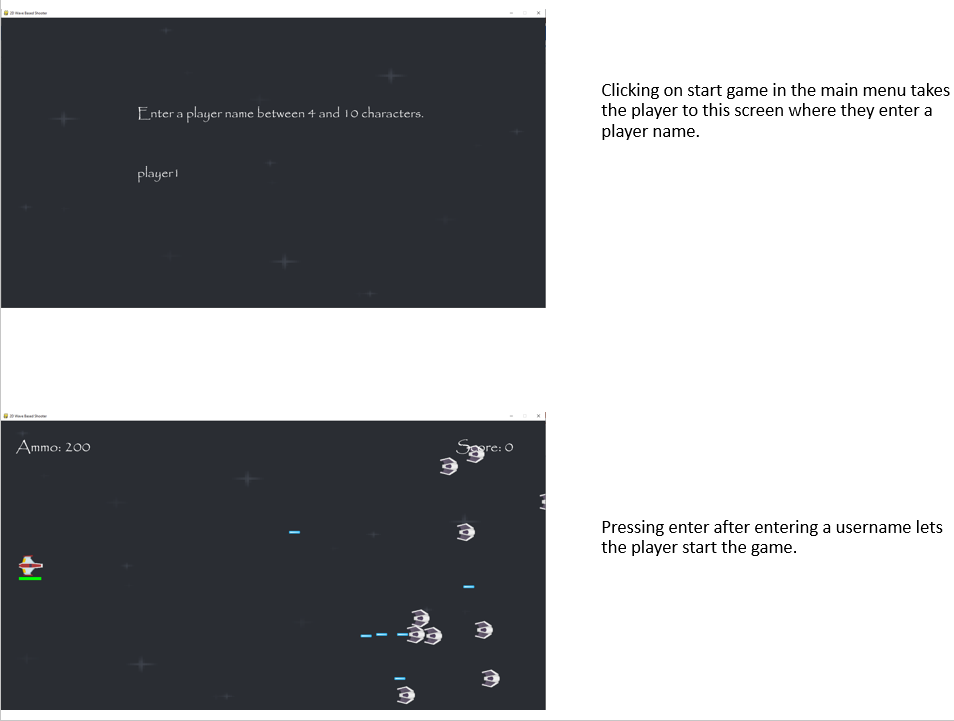


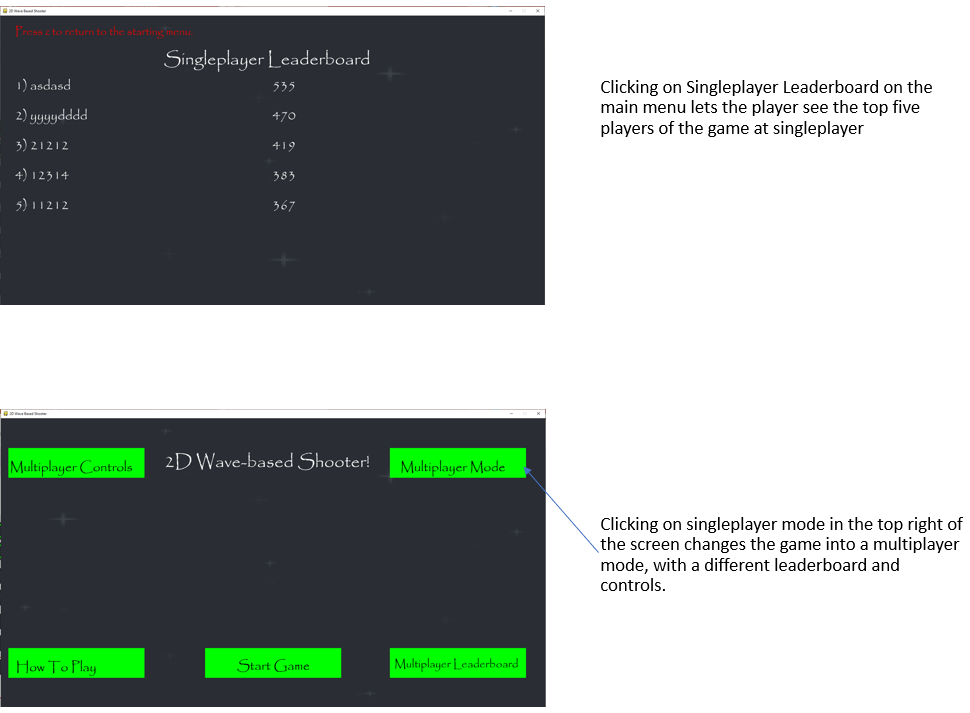


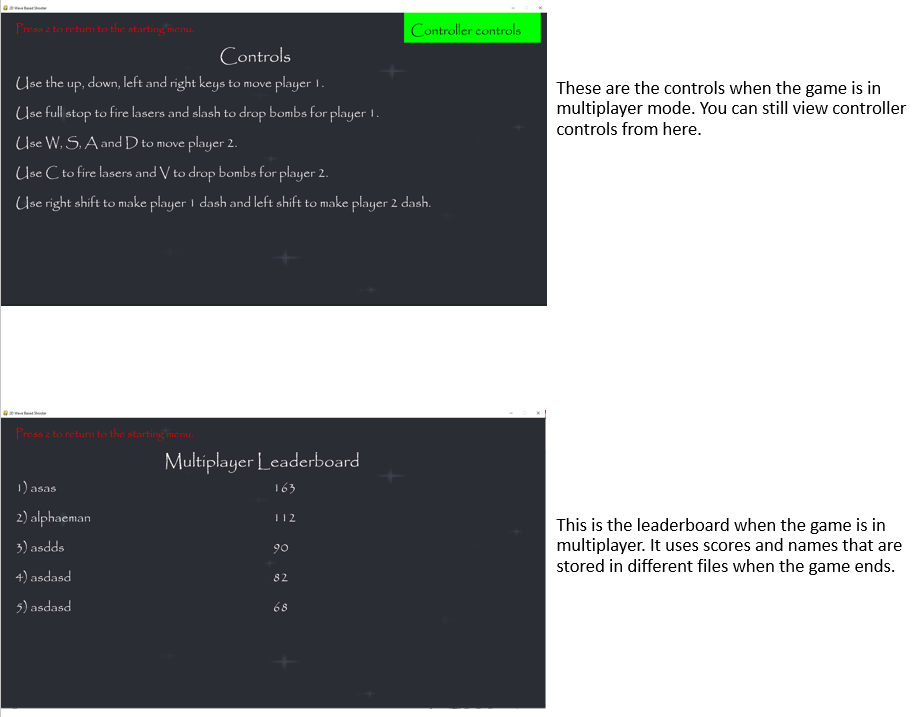
Evidence of an interactive User Interface and Consistent Layout:





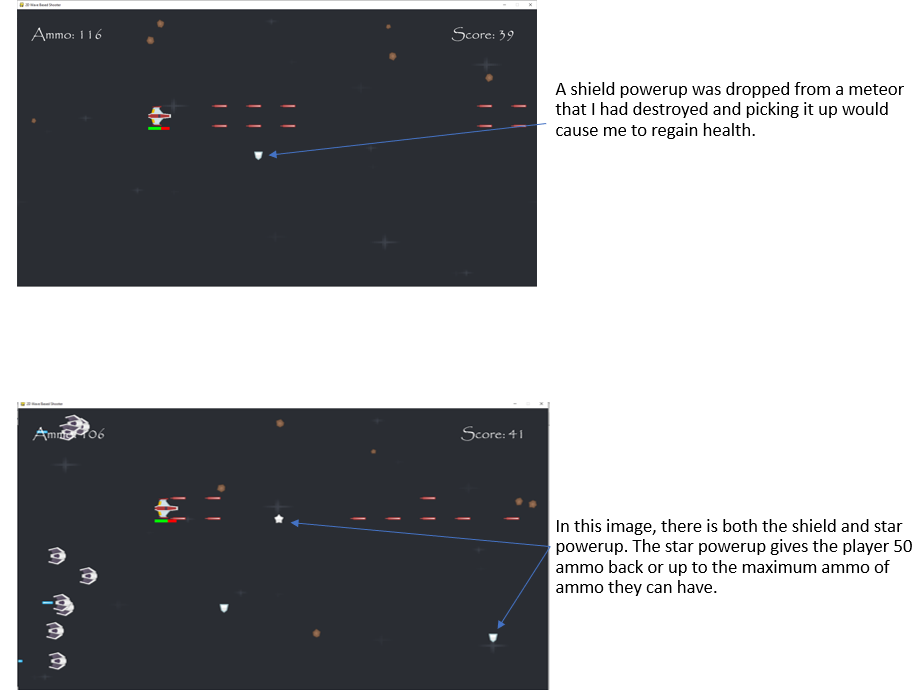




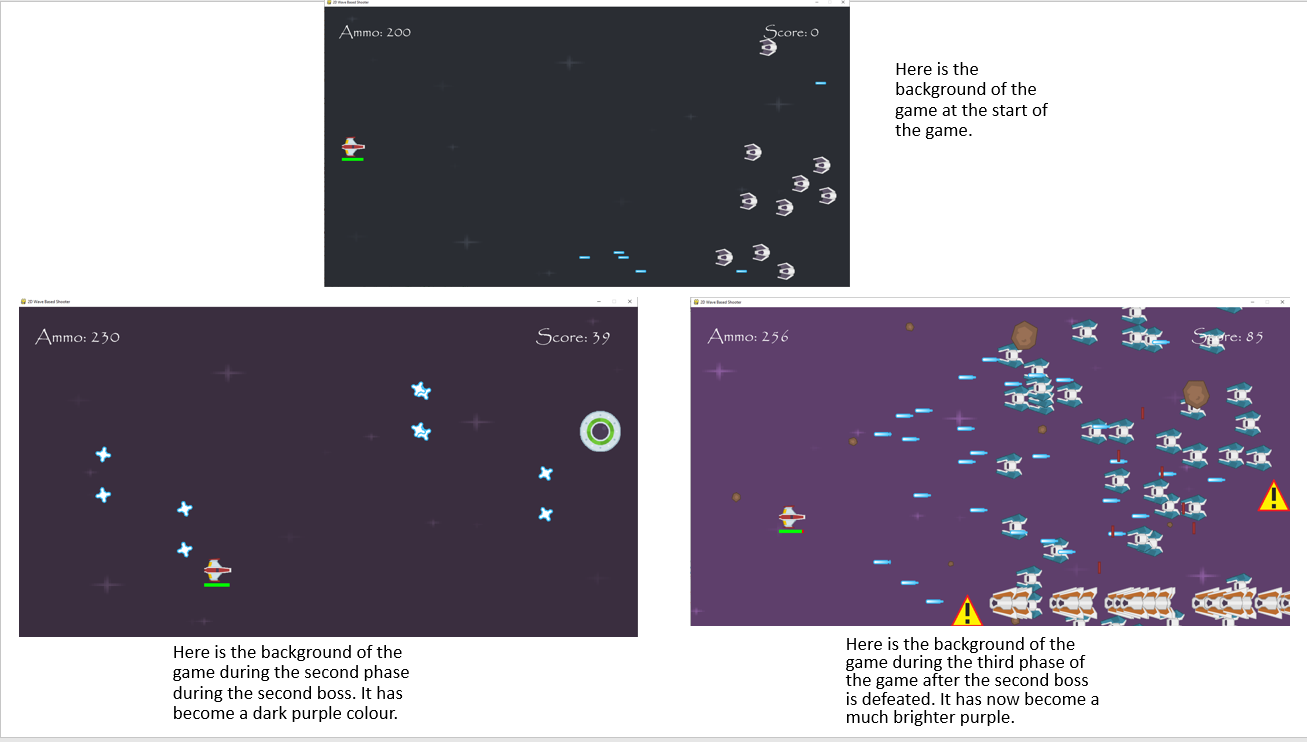


The layout of the main menu and other menus is very consistent as all the boxes are green most of the time. But when a cursor hovers over them, they become red to signify that they can be clicked on. The boxes for the different sections are also laid out consistently whether in singleplayer or multiplayer mode and they are all aligned with each-other in some way on the x and y axis.

Evidence of Powerups in the Game:

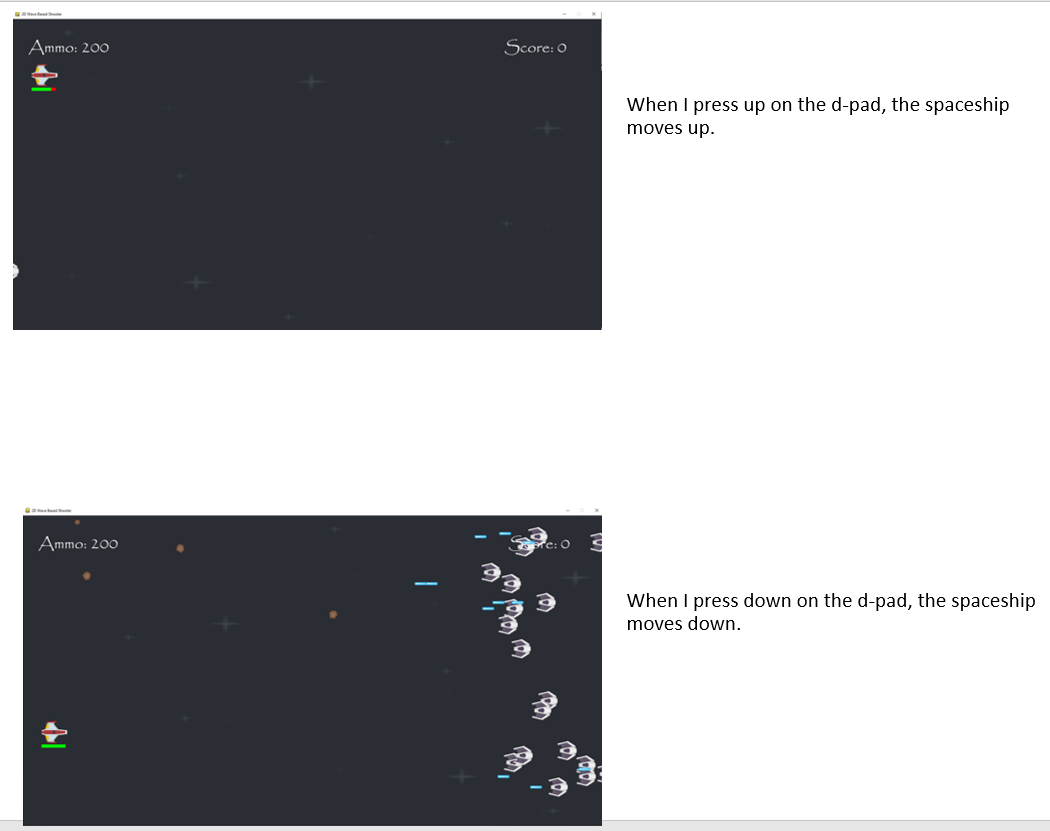


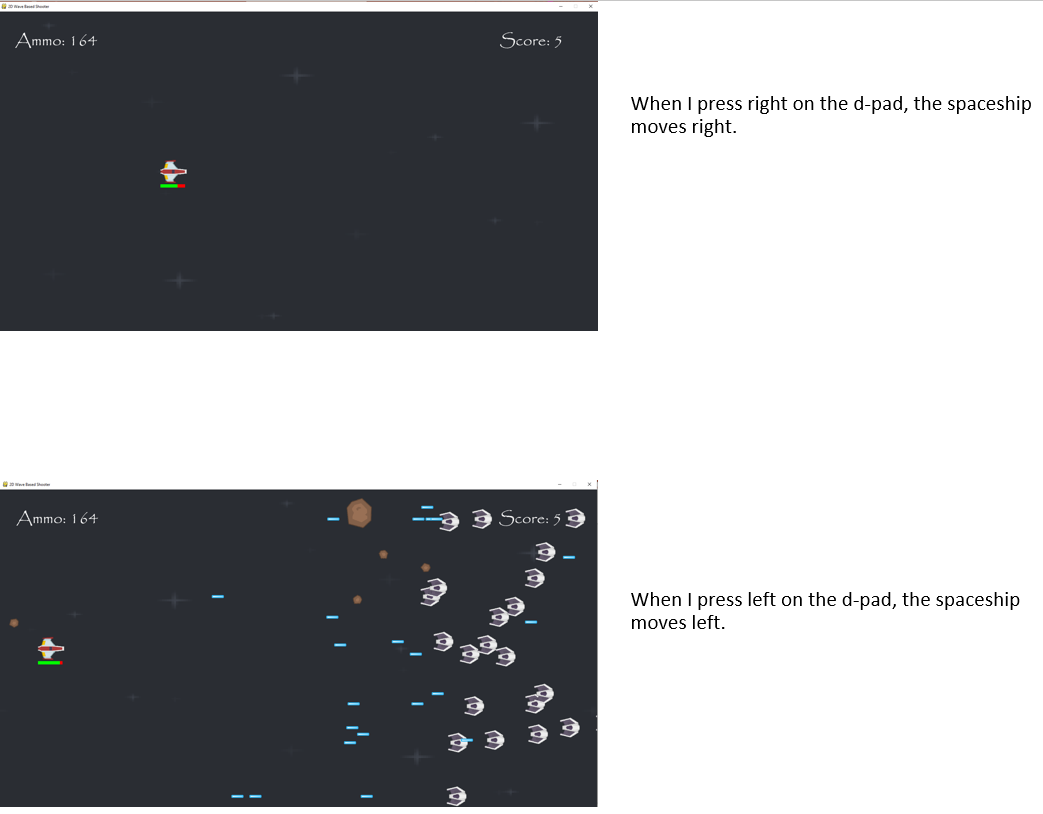
Evidence that the Game’s background changes over time:



Evidence that the Game can is played using a controller:







Evidence of Sound Effects in Game:

Unfortunately, it is impossible for me to show evidence that there are sound effects in the game through a word document.

Unachieved Requirements

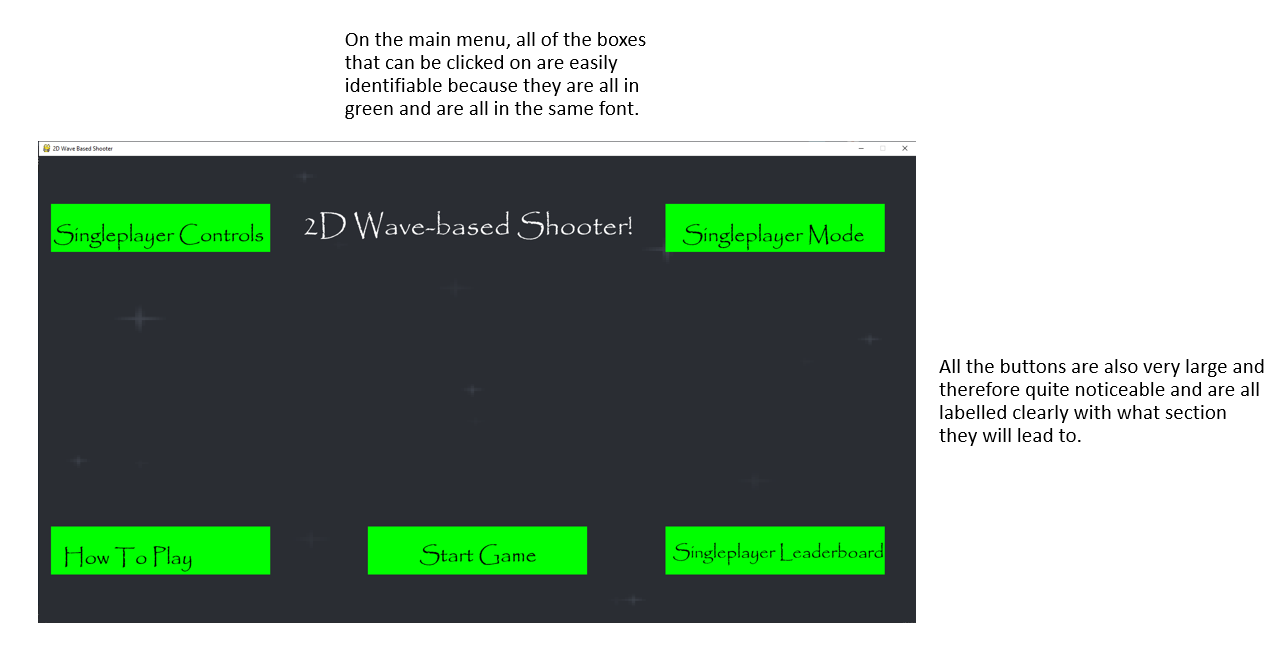
There were three parts of the success criteria that weren’t achieved. The first one was allowing the user to choose the height and width of the screen. This was originally in the game but it was removed since it would be simpler for the user if the height and width of the screen were based on the resolution of the monitor.

The second requirement that wasn’t achieved was adding online play into the game where the player would be able to play the game alongside another person who was playing on another computer. This would have been an option that could be selected from the main menu when in multiplayer mode and the player would have to connect to the other player’s computer, probably through a LAN connection, before they would be able to play the game together.

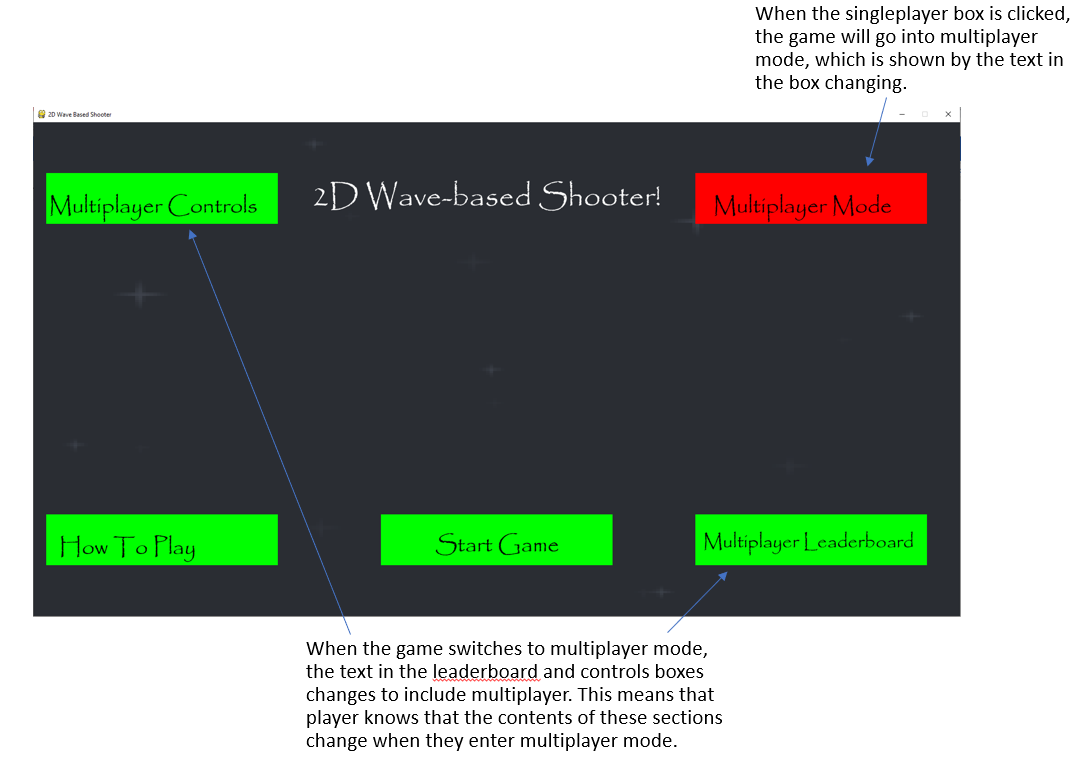
The final unachieved requirement was to have an AI player that could play alongside the player as well if they couldn’t play the game in multiplayer with another person. This would have been selected from a similar section on the main menu when in multiplayer mode and when playing the game this AI partner would have shot at the enemy spaceships and also tried to avoid their attacks to help the player out.

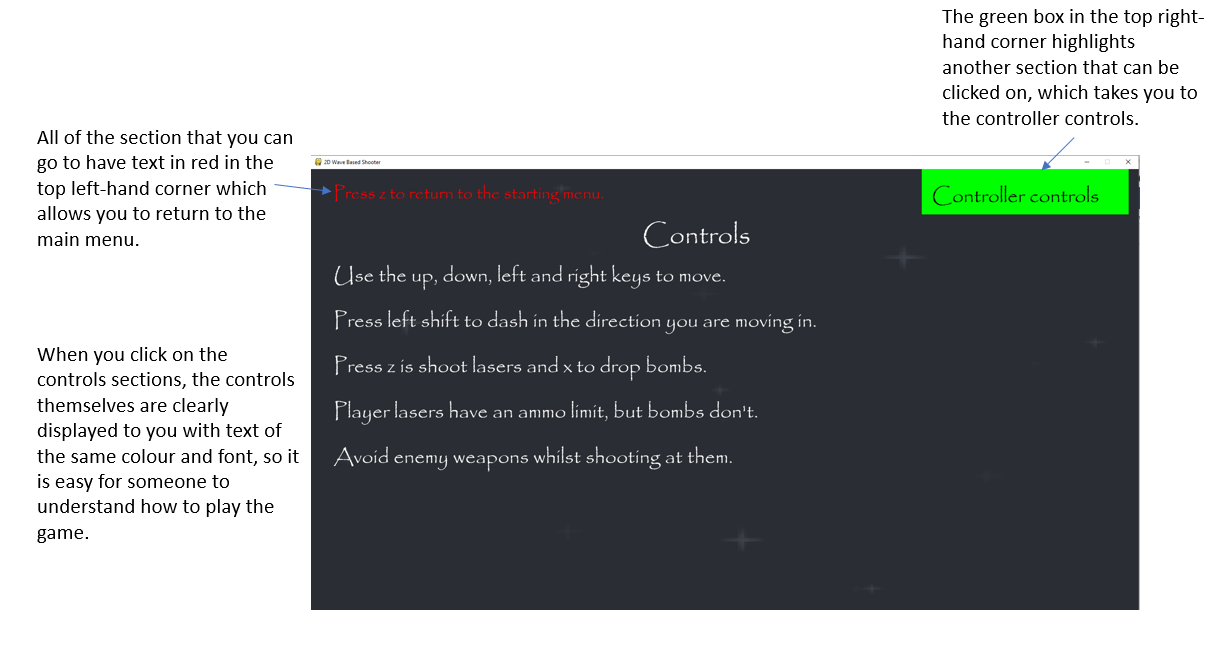
The reason why the second and third requirements weren’t achieved was because of time constraints; when I got to the end of the project’s development, I realised that adding an online component and an AI player would require me to learn a lot of new code and then write and document all of that code. This would have taken me a very long time and I didn’t have that much time and had already achieved all of the primary success criteria, so I decided against including them in the game.

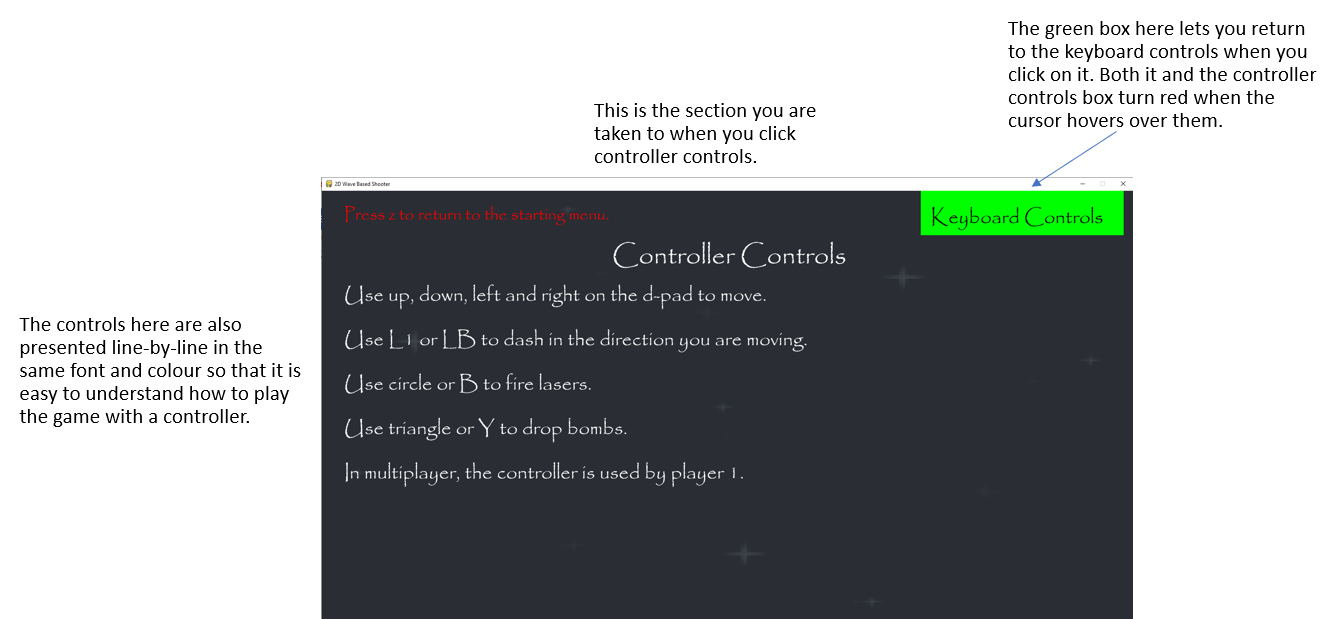
Usability Features

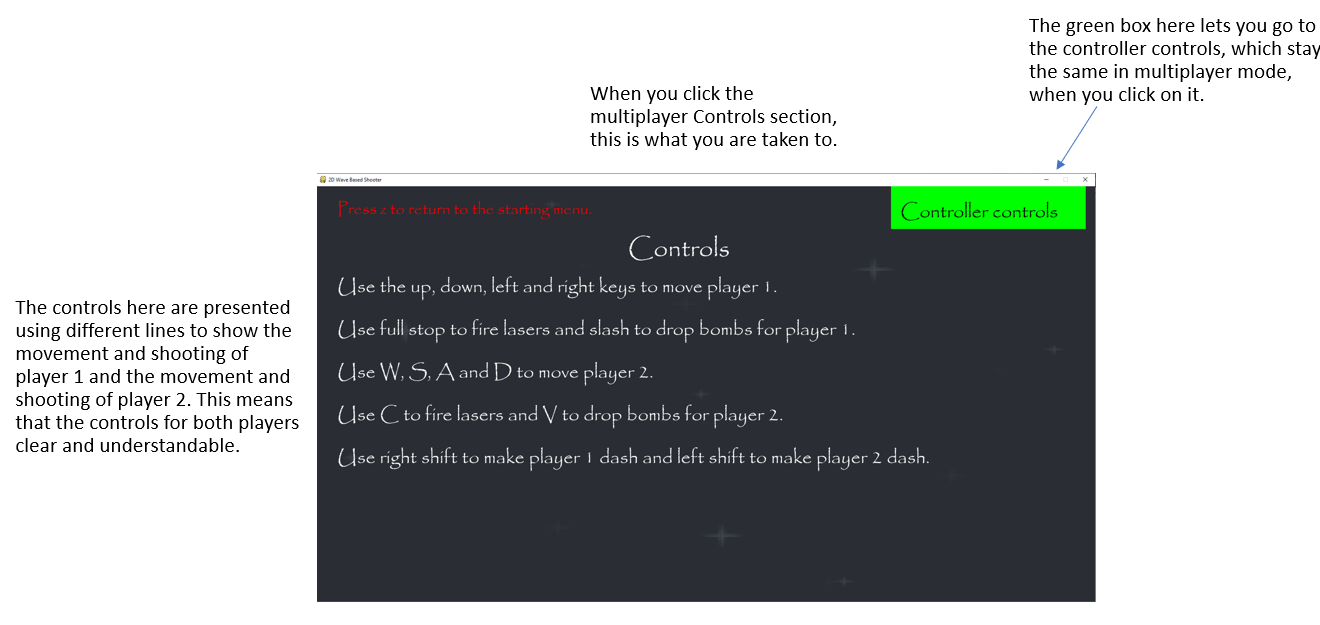


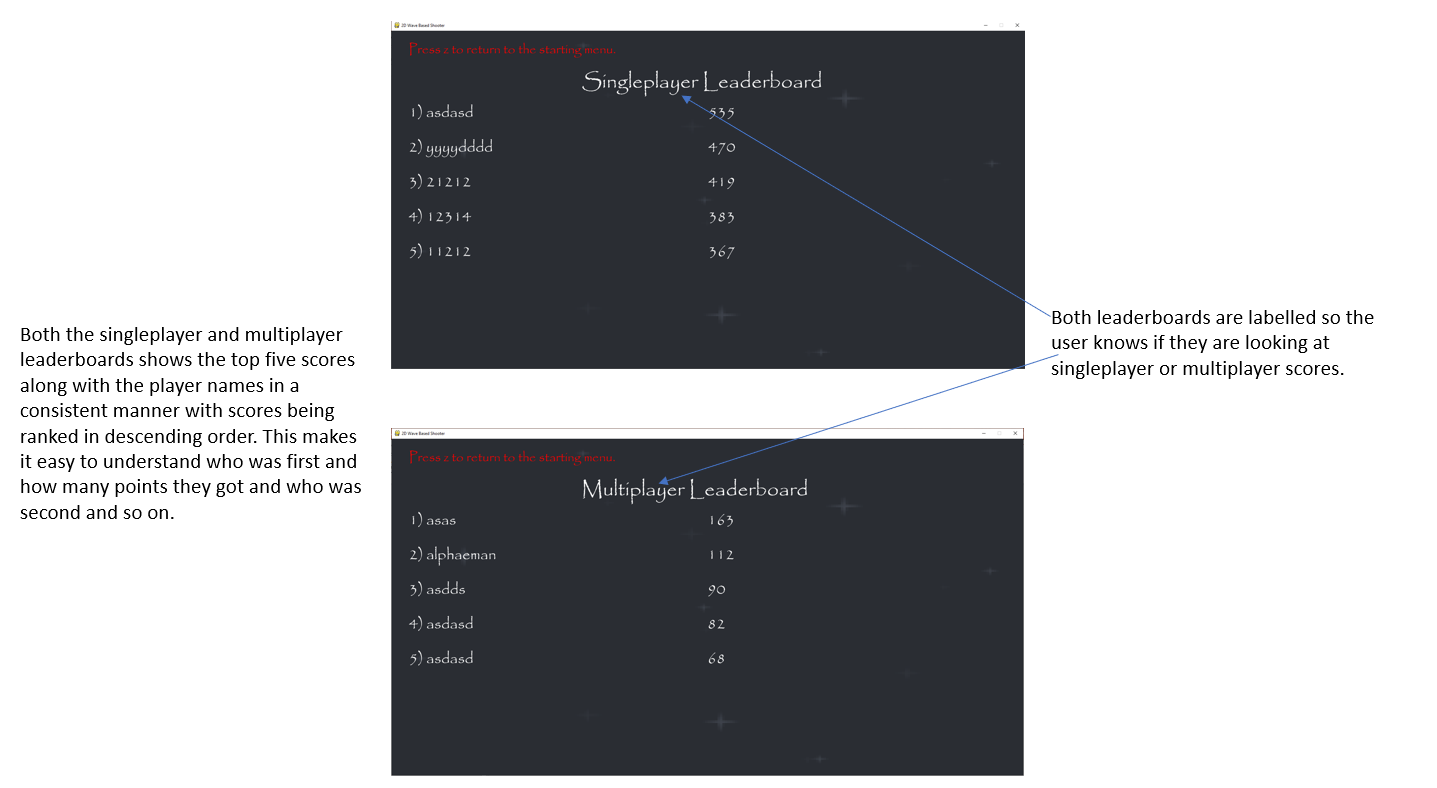


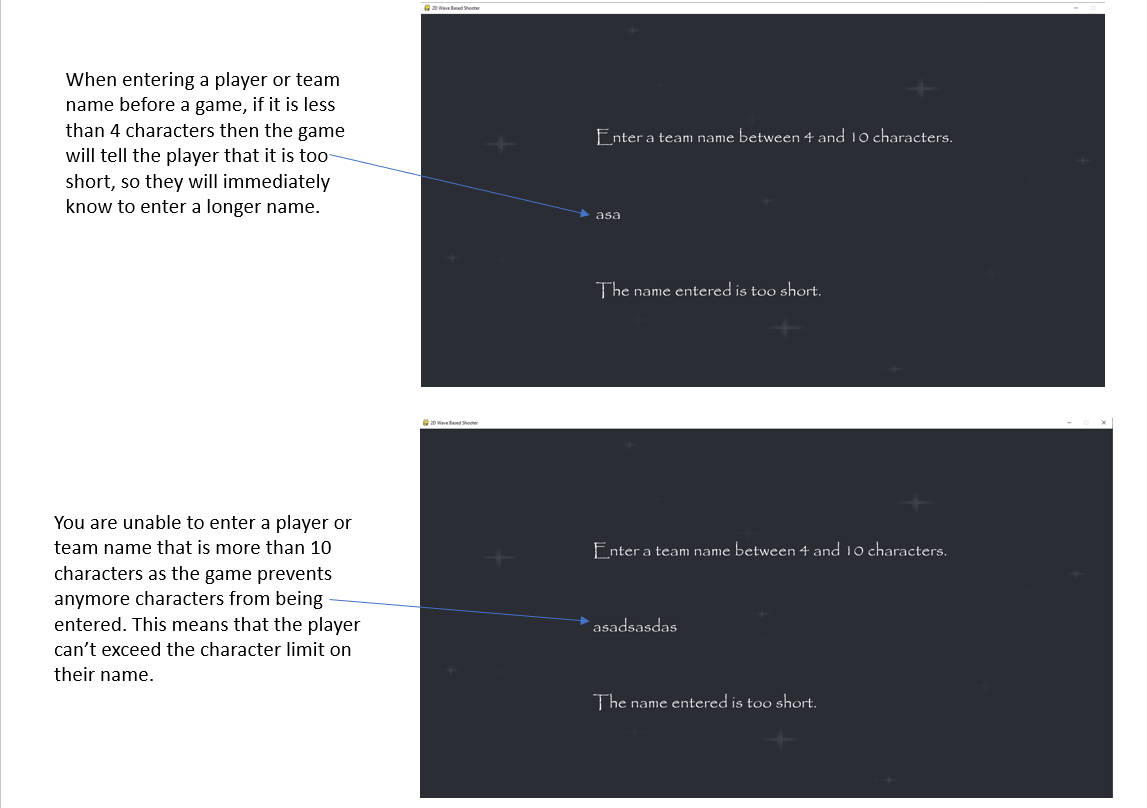












Usability is also used in the controls of the game since in singleplayer the player will use the arrow keys to move around which would be very easy for a player to learn and get used to. It is also the movement controls used in some other videogames which means that the controls will come naturally to many players. This is also done with the controller controls since the d-pad controls movement in some 2D games so using the d-pad for movement in my game makes it consistent with those other videogames, so the controls are more useable and easier to learn. In multiplayer, player 1 will use the arrow keys to move but player 2 uses the WASD keys instead. This is because many videogames, particularly 3D ones, will also use WASD keys to move the player so this would be a natural controls system that would be easy for the players to learn. The keys are player 1 and player 2’s controls are also far apart from each-other on the keyboard so that their hands are less likely to hit each-other whilst playing the game, but this might still happen. This system could be improved in multiplayer by advising one of the players to use a controller, or by letting the players change the controls to something that works for them.

Limitations

One of the limitations for this game was the lack of an online mode for the game. This would have meant that two people playing the game on different computers would have been able to play together in a co-op game and this couldn’t be coded because I would have had to spend a lot of time learning how to incorporate online features into pygame software. This ties into the main cause of the limitations which was the fact that I simply didn’t have enough time to add in all the features that I wanted as coding them and learning how to code them would have taken too long.

Another limitation is the lack of any change in gameplay depending on the screen size. What this means is that the number of enemies displayed onscreen is constant regardless of the screen’s height and width which means that the game will be harder for people playing the game with a smaller screen since they will have less space to move around in to avoid enemy weapons and hazards and if the screen is too small then there may be instances where the player is destroyed since they were completely there was no way they could have avoided the enemy attacks. The same thing applies to multiplayer since the game doesn’t get any harder when there are two people playing the game which makes the game significantly easier than it would be otherwise, especially when the game is played with a big screen.

In addition, I couldn’t add in an AI player that could fight alongside the player spaceship because it would have been incredibly difficult to code an ai player that could dodge enemy attacks to an extent and fire back at enemies whilst knowing which weapon to use without making the AI player either way to powerful thus making the game unplayable for the human player, or way too weak thus causing them to die instantly.

One final limitation was the lack of a pause menu which would have allowed the player to pause the game whilst they were playing if there was something else that they had to quickly do. This means that, instead, players are forced to continue playing until they are defeated which isn’t an ideal situation.

Fixing the limitations

Having more development time would have allowed me to add an online mode so that the player could play with someone on another computer. This would involve me learning how to code online features in pygame. Changing the number of enemies depending on the screen size would be much easier to implement as it would involve me simply having change the number of enemies in a wave depending on the size of the screen, which would be determined from multiplying its height and width together. This size would then determine the number of enemies in a wave. For a multiplayer game, I would just take the number of enemies determined for the height and width in a wave, and simply double that number in order to compensate for two players being onscreen at once. I would also double each boss’s health so that the difficulty remains similar in multiplayer.

Adding in an AI player would be another objective that would require a lot of time to achieve but ultimately the AI player would be able to know where enemy attacks are and how best to avoid them. They would also be able to determine when to use lasers and when to use bombs on an enemy and when to dash. However, it would also have a chance of not doing these things when attacking or when attacking so that it isn’t too powerful relative to the player character. In addition, the game wouldn’t end when it is destroyed since otherwise that would be annoying for the player since the game would have ended for something that wasn’t their fault.

Implementing a pause menu would involve the pause menu itself as well as allowing the player to return to the main menu from that pause menu and letting them return to the game when they want to. The pause menu would be achieved by freezing everything onscreen whilst the game being paused and gameOver would become True if the player wanted to go back to the main menu so that the game would finish.

Maintenance

If I made the number of enemies in a wave dependent on the resolution of the screen, then the program might have to be updated in the future. This is because the size of the screen is dependent on the resolution of the monitor, and as time goes on bigger and bigger resolutions are being put onto smaller screens. This means that in the future, a person playing the game on a 4K laptop and a 4K PC would both face the same number of enemies, but it would be harder for the person on the laptop since that would have the smaller screen so there would be less room to move the spaceship around in. This means that in the future, the way that the number of enemies in a wave is decided would be by looking at the size of the screen itself.

For further maintenance, when new games consoles come out in the future; they will use new controllers that might not be able to function with my game, so I would have to add code so that the game recognises these new controller inputs, and I would also have to update the controls section on the main menu to tell people how to play using these new controllers.

Another feature that could be added would be the ability to change the buttons used for the controls, so the player could assign different keys to movement and shooting controls. This would be done with controls in singleplayer and for both players in multiplayer. In addition, you would also be able to change the controller inputs that are used for movement and shooting.

If I were to continue making the game into the future, I would also add in more phases that introduce new bosses, enemies and hazards. These would be harder than previous ones with new attacks and more health so that the game would still get more difficult over time and there would be more of a game to play so people would play it for longer. New powerups would also be introduced that have different effects on the player, like maybe making them move faster temporarily or giving them infinite ammo for a short period of time.

Because of the object-oriented structure of my code, it would be easy for stakeholders or other developers to add or change different characters in the game. This is because the attributes and methods for each object are held in the same class and I made sure to give different attributes and classes sensible names that would make them easy to identify for the stakeholders or coders. This also means that individual classes can be changed without changing other parts of the program, so even if a mistake is made, it only affects that class since the classes are mostly isolated from each-other. A lot of the rest of the code is written using sub-routines which can be changed individually in the same way.

I have also included many comments in my code to lay out what different lines of code and sections of code do. I have also used comments to describe what a lot of the important variables and attributes are used for and how they might change. This means that it would be quite easy for another programmer to make changes to the code if they want to as it would be easy to understand.

Final Code for the Programming Project

