### **About the game:**

**Name: Asteroids**

The game I have created using Python’s Pygame Engine as well as Visual Studio Code is, a recreation/almost identical, version of the Arcade Game ‘Asteroids’. When playing this game, the aim is to survive a constant barrage of asteroids, as well as accumulating a large sum of points from destroying the asteroids. If you collide with an asteroid regardless of its size, you “die” and restart the entire game from 0 points. Ultimately, I created the game for all ages aside from those who are unable or haven’t learned how to use a computer.

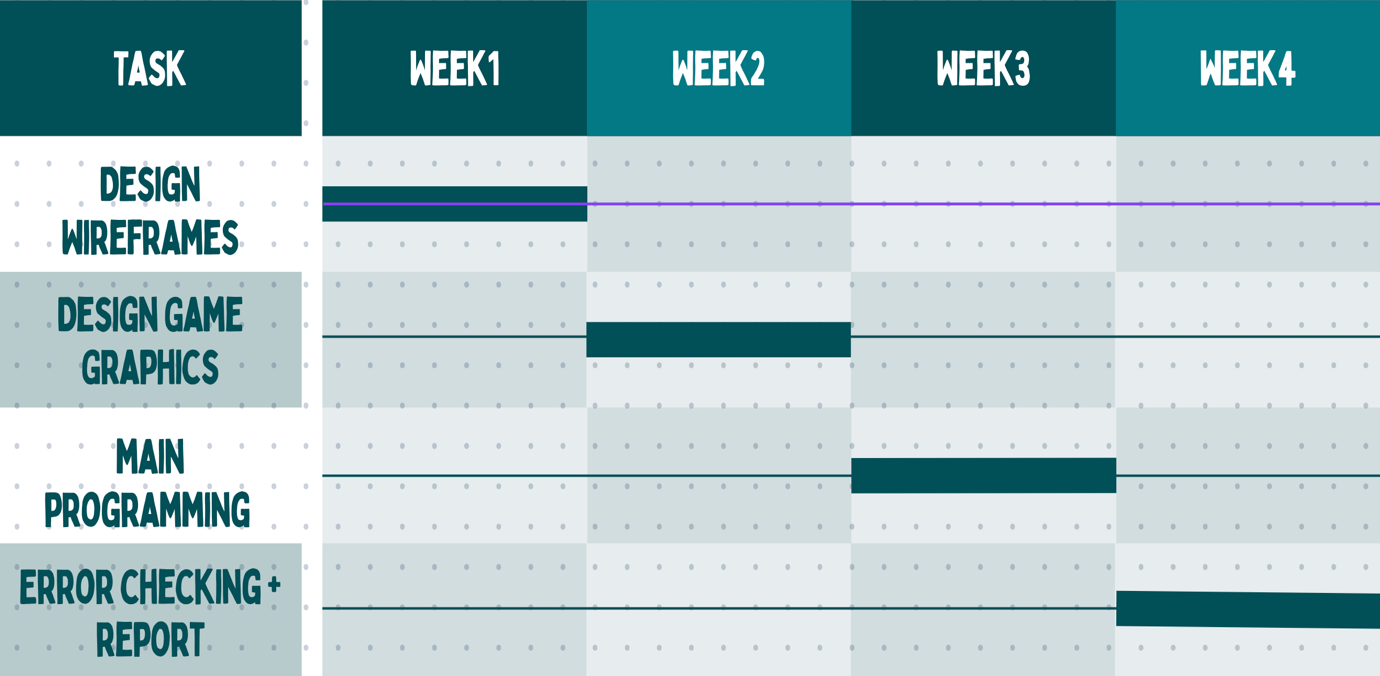
### **Creating the Game:**

To successfully develop my game, I had to follow specific steps and use methods as well as web-based resources to efficiently organise my time, with the purpose of fully developing the game by the due date.

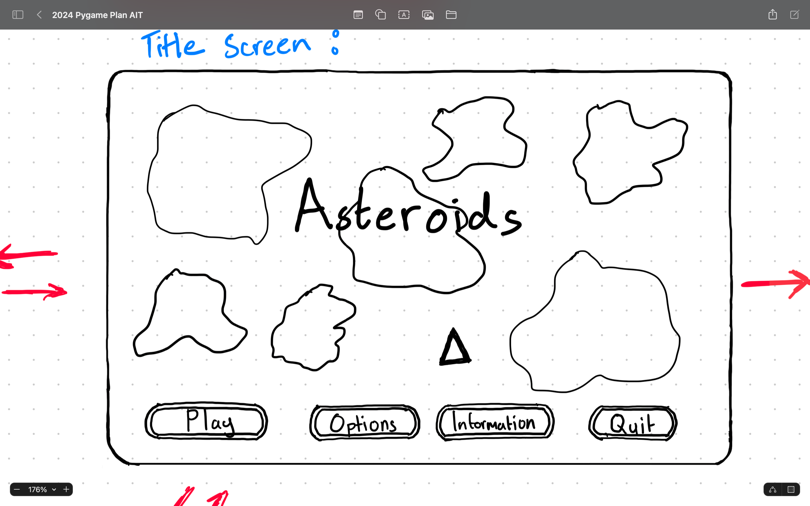
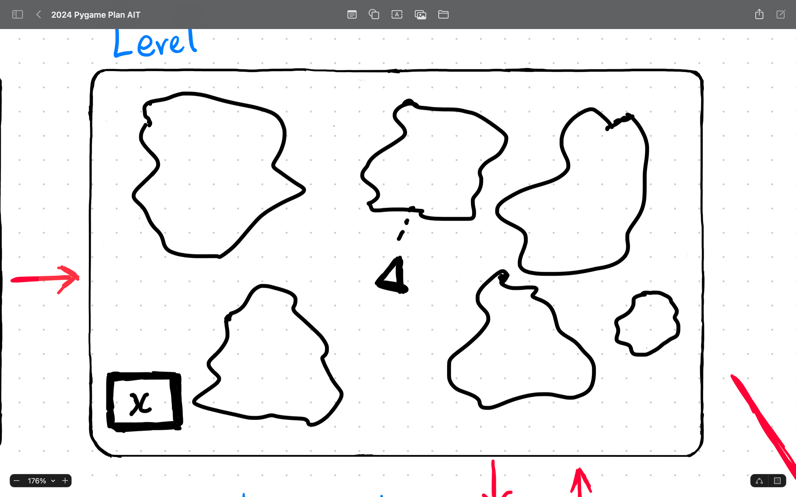
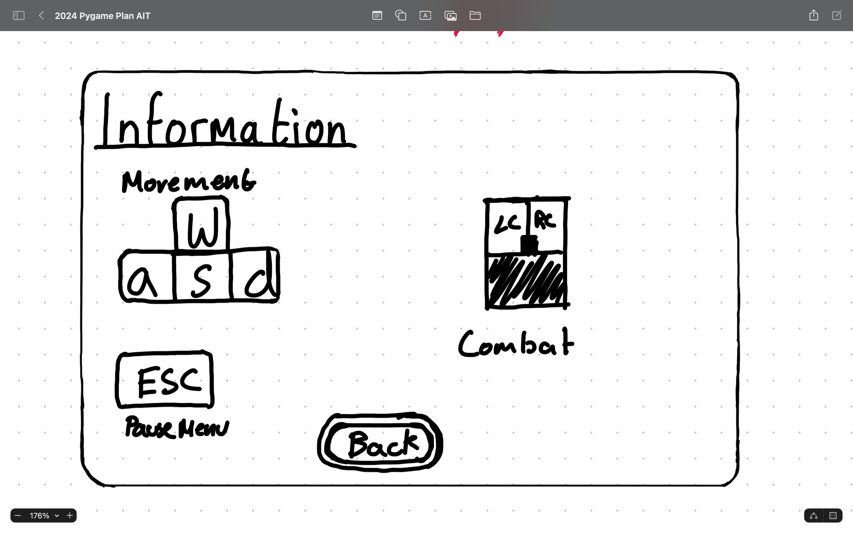
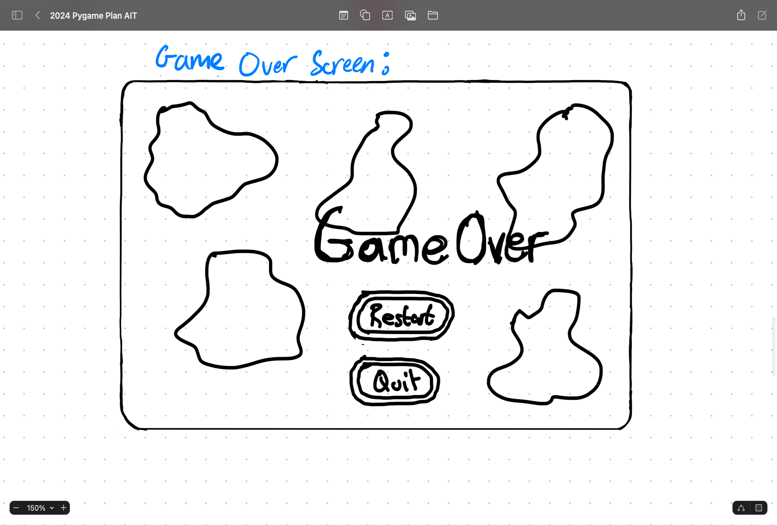
|  |  |
| --- | --- |
| **Week Number** | **Task/s** |
| **Week 1** | **Design Wireframe Sketches:** Over the course of week 1 (20 Feb 🡪 27 Feb) I used my iPad and the app called FreeForm to design the wireframe sketches. These sketches aren’t detailed but outline the main design of the game (interface especially) |
| **Week 2** | **Design Game Graphics:** Over the course of week 2, I utilised Piskel (online web-based pixel art creator) to draw the backgrounds for the game states as well as the Asteroid sprites for the main game. |
| **Week 3** | **Main Programming:** I used Visual Studio code to create/write the code for the game. I utilised Python’s Pygame game engine to create the game. During this phase I incorporated the sprites and visual designs that I drew. |
| **Week 4** | **Error Checking + Report:** I ran the code/game and tested for errors (none arose) however, the game over menu (the return to main menu button) wasn’t operating functionally. I finalised the project by writing my report and reflection. |

### **Gantt Chart:**

I utilised Gantt Charts which showed a simple breakdown of tasks for each week that I had to complete, as well as designing, brainstorming, and sketching the main aspects, ideas, and aesthetics of the game by hand in my journal.



### **Wireframing:**

Here are the sketches for the game that I drew using my iPad. The sketches are basic and non-detailed however the game itself is detailed.

### **Game Features:**

|  |  |
| --- | --- |
| **FEATURE** | **OPERATION** |
| Main Menu / Title Screen | * The main menu includes the play button, options button, information button, and the quit button. * The main menu displays the title of the game as well as a simple user interface to play the game or quit the game. |
| Play Button | * When clicked you enter the game. * I used Pygame.rect and Pygame.circle to draw the button. |
| Options Button | * When clicked transports the user to the options screen where the user can change the display settings of the game. |
| Instructions Button | * When the button is pressed, the user is transported to the information page that displays the key binds and aim of the game. |
| Quit Button | * When clicked, quits the game and exits the Pygame window. |
| Asteroids | * The Asteroids are entities/sprites that constantly move/travel across the screen in random orders. * If the player collides with one of these asteroids, you lose. |
| Player | * The player that is controlled by the player that is the main sprite used to play the game. * Made using Pygame.draw.polygon and Pygame.draw.line. |
| Pause Menu Screen | * The Pause menu activates when self.pause = True, otherwise it continues to update the game. * The Pause menu has a working Resume Button and Quit Button. Unfortunately, the Restart Button and the Options Button aren’t working. |
| Game Over Screen | * The screen (game state) that loads when the player collides with one of the asteroids to state that you have lost the game and needs to restart. |

### **User Manual:**

**Movement:**

* W key = move forward/upwards
* A key = move left
* S key = move backwards/downwards
* D key = move right

**User Interactions:**

* Escape key = toggle the pause menu (Restart and Options Buttons not functional)
* Mouse = used to press the buttons on the main game states to perform the essential actions of the game.

### **Reflection:**

Overall, the experience of creating this game was interesting, as it required me to learn skills that I had never even though about in the past. The creation of the game also required me to utilise AI tools such as BlackBox AI, ChatGPT, Microsoft Copilot, and Tabnine AI, which all provided me with notes and instructions on how to effectively build my game.

Creating the game took a substantial amount of time and effort that produced results that were close to what I had initially desired when first developing my game. I encountered many challenges such as, determining the idea that I wanted to use for my game, creating/drawing the graphic, applying motion physics to my game, and developing shooting mechanics for the game. There are many areas of the game that I can improve such as the “game over” screen, and the “pause menu” screen as both do not “fully” function and are some errors when interacting with them.

### **References/Bibliography:**

Microsoft (no date) *Microsoft copilot: Your everyday ai companion*, *Microsoft Copilot: Your everyday AI companion*. Available at: https://copilot.microsoft.com/ (Accessed: 25 March 2024).

(No date) *Chatgpt*. Available at: https://chat.openai.com (Accessed: 25 March 2024).

*Pygame v2.6.0 documentation* (no date) *pygame*. Available at: https://www.pygame.org/docs/ref/pygame.html (Accessed: 25 March 2024).

(No date b) *Pygame.font - pygame v2.6.0 documentation*. Available at: https://www.pygame.org/docs/ref/font.html (Accessed: 25 March 2024).

(No date a) *Pygame.rect - pygame v2.6.0 documentation*. Available at: https://www.pygame.org/docs/ref/rect.html (Accessed: 25 March 2024).