

Space Walker Report

For my game I decided to build a space-themed level-based runner game. The player takes on the role of a space explorer who has crash landed in the outer rim of the galaxy and they must progress through each level by overcoming various physical obstacles on the planet they have crash landed on. In order to go home, the player must repair their ship, the StarSparrow, by purchasing different components from the store. The player gains credits by collecting coins found along each level. Once the ship is complete, the player can do the final mission of flying through the asteroid belt to return home.

How-To-Play:

- Select the level you want to play
- Move along the path and dodge any oncoming obstacles.
- Purchase ship parts from the store as you progress through each level to gain buffs (stats bonuses) for your character
- When all ship parts have been purchased, start the final level
- For the final level, fly through the asteroid belt and dodge or shoot any incoming asteroids.
- Once you make it to the end of the asteroid belt, the game is complete!

Controls:

PC:

- Movement: W/A/S/D or Arrow Keys.
- Jump (Not available in Final Level): Spacebar.
- Shoot (Final Level Only): Left Click or Spacebar.

Mobile:

- Movement: Slide your finger along the screen.
- Jump (Not available in Final Level): tap the screen once
- Shoot (Final Level Only): Press the shoot button highlighted on the screen.

Character Buffs (Player Stat Bonuses):

- **Extra Coins:** For every coin collected you gain a 2x bonus.
- **Speed Boost:** You run at 1.5x speed.
- **Jump Boost:** You jump at 1.5x height.

FEATURES:

Cutscenes:

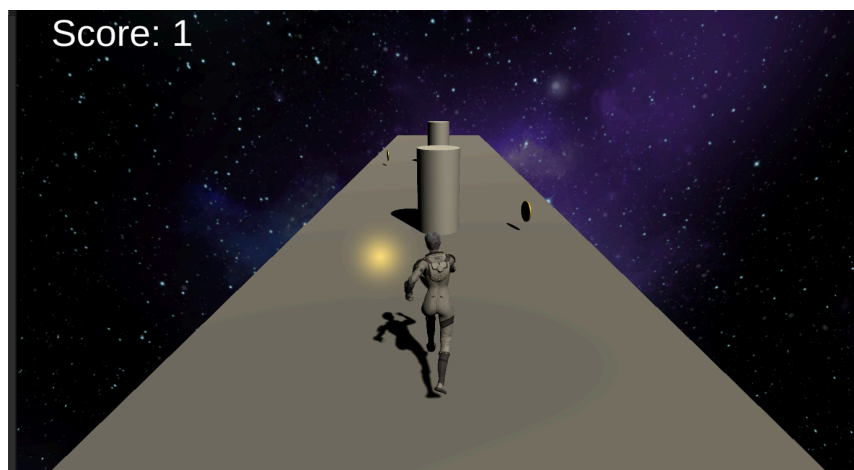
The game takes advantage of Unity's built-in timeline editor to provide 2 cutscenes that tell the story of the game. The first one happens when the player clicks the "Play" button for the first time. This shows the beginning of the story and how the character has ended up in this situation. (See figure 1 below) The second cutscene plays after the final mission and shows the character escaping and then rolls through the credits.



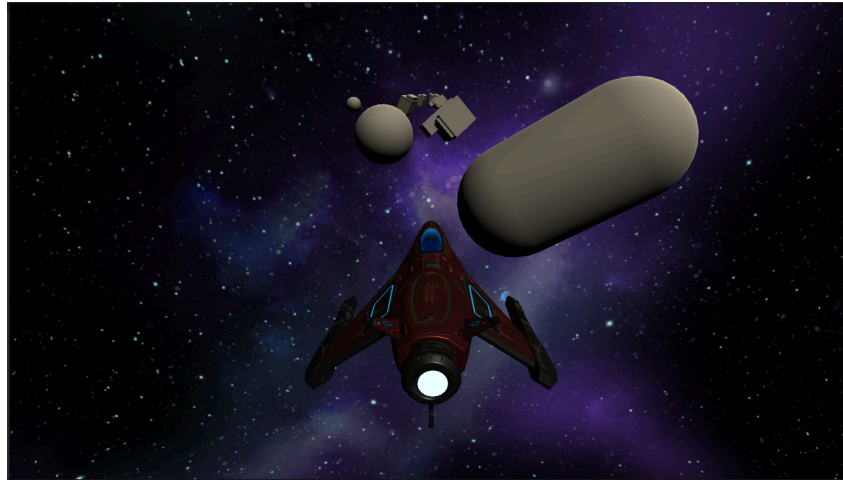
[Figure 1 - The opening cutscene showing the player character crawling away from their crashed ship]

Levels:

The game features 4 main levels along with a 5th final level that is unlocked upon purchasing all ship parts. The 4 main levels are instantly available to the player and can be completed in any order, however they are designed to get the player used to the game so sequential order is highly recommended. For example, the first level is very short and just allows the player to learn the basic movement to dodge some small cylinders in the centre. (See figure 2 below). The second level is a little longer and features a gap in the middle which requires the player to jump. The 3rd and 4th levels are more advanced, requiring the player to use everything they have learned to that point. The final level changes the flow of the game completely and the player must control a spaceship and dodge/shoot at incoming asteroids in order to escape (See figure 3 below).



[Figure 2 - The opening level featuring 2 relatively simple obstacles that the player must avoid]

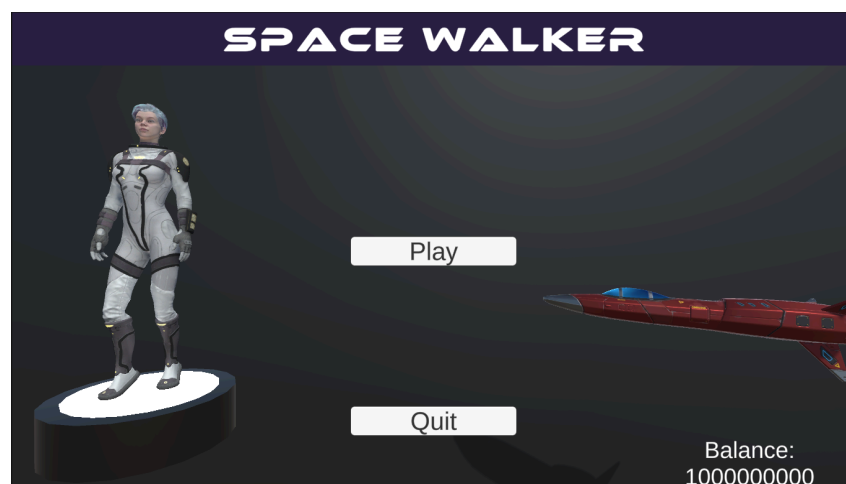


[Figure 3 - The final level featuring the player controlling the StarSparrow ship]

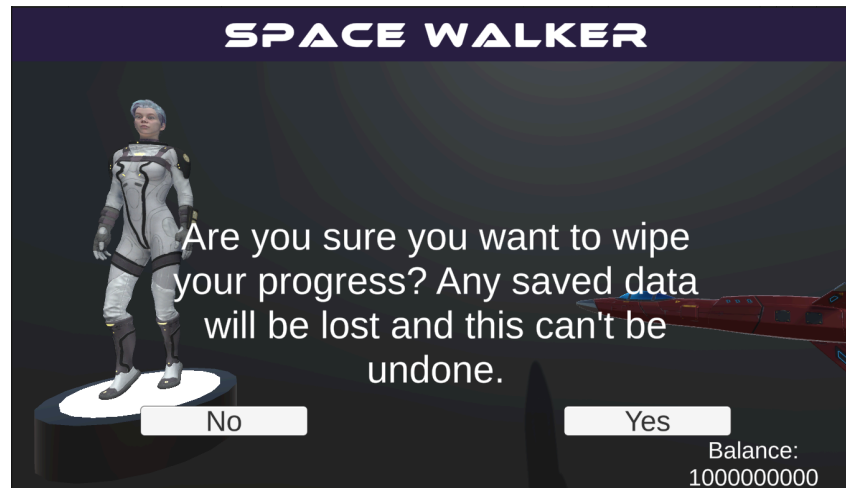
Menu:

The game has an opening menu that shows the character on the left as well as the player's progress on repairing the ship on the right (See figure 4 below). Every time they purchase a new part, it is added to the display on the right. The player can initially choose to play or exit the game. If they choose to play then they can choose the level selector, wipe their progress or visit the store.

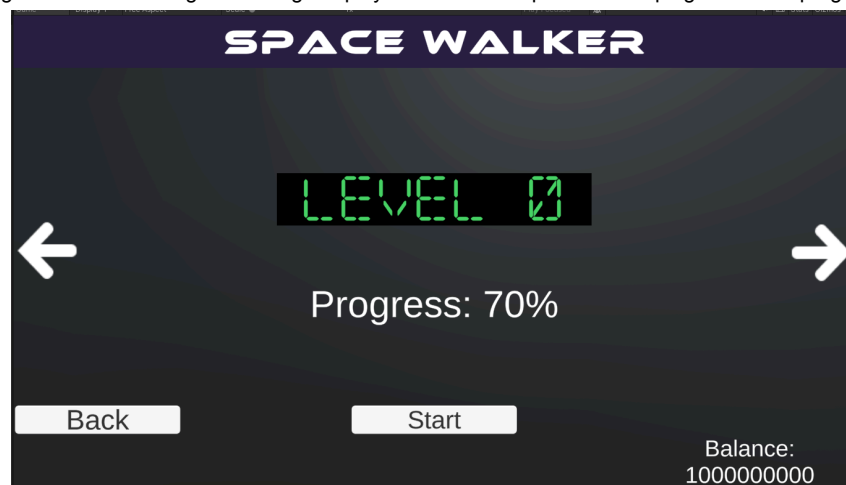
The button to wipe their progress will ask them if they are sure about the decision and warn them of the consequences (See figure 5 below). The level selector features a GUI for the different levels that displays the name, a start button as well as the player's progress for that level (i.e. how many coins collected) (See figure 6 below). If the ship is not complete, the final level's start button will be faded out and there will be a message informing them that they must have the ship fully repaired in order to start. If the ship is fully repaired, then the progress is displayed as either complete or incomplete since there are no coins to collect in this level.



[Figure 4 - The first section of the main menu that the player sees when starting the game or arriving at the main menu from the store]



[Figure 5 - The message informing the player of the consequences of wiping their save progress]



[Figure 6 - The level selector GUI, showing the level name and how much progress the player has made on it]

Store:

The store is an area where the players can browse the available ship parts. They can use the buttons on either side to scroll through the options and they can make a purchase using the purchase button below each one that displays the price. (See figure 7 below) If the player can't afford an item the purchase button will be faded out. If the player already owns the item, the button will be faded out and specifically state that it is already purchased. The player's current credits are displayed in the top right corner. This area features retro store background music as well as a spotlight sequence whenever the player enters the store.



[Figure 7 - The store area showing the different ship parts the player can purchase along with a description and price of each one.]

Issues:

Several issues were encountered during the development process of this game, some of which I managed to overcome, others required the feature to be removed or implemented differently.

For example, the core initially offered the player an extra life bonus, which would move the player back to their last safe checkpoint if they ever died. However, this led to some problems as changing the position of a moving gameobject was very inconsistent and didn't always work. I tried several different approaches for this with no success from any. I ultimately decided to change the buff to something else to avoid wasting time on it.

The speed boost and jump boost initially were too much and during play testing, the players found it hindered their gameplay. I had to manually adjust the values for these buffs to something that felt more comfortable.

Porting the game to mobile was very difficult. While Unity has great tools to do this, making new controls for a mobile game is not a light challenge. I followed several tutorials online and did research into how to detect touch screen taps before I finally arrived at something I was happy with. The jump button was the most difficult but I decided to have it so the jump would be triggered by a single tap. For the ship's guns in the final level, I felt that using the same approach as jumping would just make dodging too difficult and the two controls would overlap too much. I decided it was much simpler to just add a button on the side that only appears if the game is run on a mobile device.

Additional Credits:

Ebal Studios on the Unity Asset Store - Ship models

Mixamo - Player Character avatar

"Wee Free Music" on YouTube - Level 3 music, final level music

"Eon" on YouTube - Main Menu music

Onyys FR - Final Credits Music

"Dave / Game Development" on YouTube - Gun Script for the ship

Ayomide Sola-Ayodele - Game design ideas

Liwia Ejsmond - Game design ideas + Play Tester

Michael Roberts - Game Design ideas + Play Tester