Single Player Game Design and Development

CI328 – Internet Game Design and Development Module

Author - Jay Massey

Student Number - 17818608

Submitted 27/01/2020

Game Summary

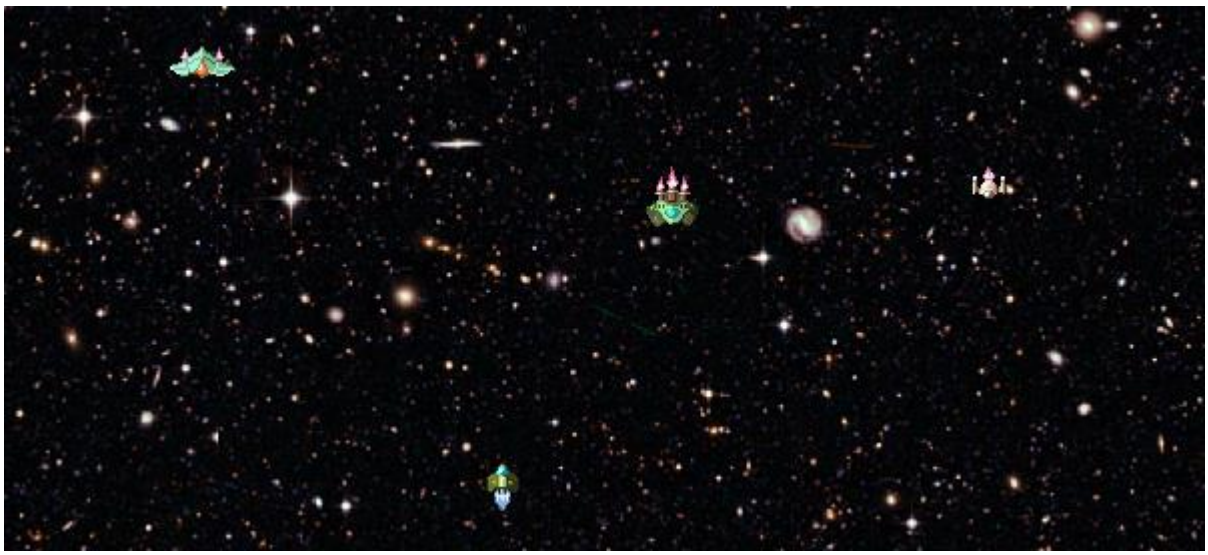
The aim of the game is to gain points by shooting enemy ships that appear on the screen, flying the screen at different speeds. The more points you obtain, the faster the game becomes. If you hit an enemy ship with your own, your ship is destroyed. The game then slows down back to its original speed, before respawning your ship. The goal is to have the highest score possible before dying.

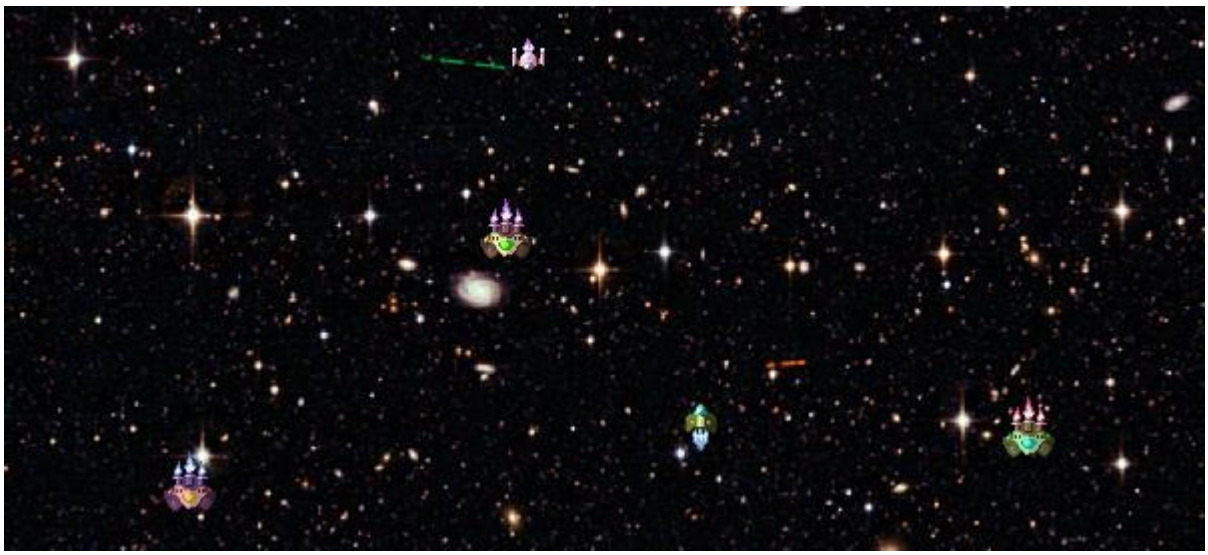
In order to move your ship, you must press the arrow keys in the direction you desire to move. To fire at the enemy ships, you must spam the space bar; the faster you spam, the more beams fired. If you ship is destroyed, your high score continues to be displayed until you land your first shot on an enemy after respawning. This is done to allow you to see your score before starting the game again.

Game Link:

https://jaymassey98.github.io/CI328-Demo/

Storyboard





Implementation Specification

The code is split into three main sections. These are **assets**, **js** and **libs**. **Assets** is further divided into five sub- folders, including **font**, **icons**, **images**, **sounds** and **spritesheets**. Font contains an image of text that is used by the Phaser engine to display the score. **Icons** contains all the favicons. **Images** contains all the image files used in the game itself. **Sound** contains the audio files that play during the game. **Spritesheets** contains various spritesheets, such as an explosion animation. All the art is laid out this way so that the project is easy to navigate, allowing for a cleanly developed structure.

All the JavaScript code for this project is located in the **js** folder for the same reason, with exception to the Phaser library. This is located in the **libs** folder, for easy distinction. There are four JavaScript files in **js**, which are *game.js*, *bootScene.js*, *playScene.js* and *systems.js*. When the game is loaded, a config file located in *game.js* causes the game to begin with predetermined values. The game loads *bootScene.js*, which loads in all the textures and audio that the game will then trigger when needed in *playScene.js*. The two scenes are separated to allow for a clearer distinction of phases. *Systems.js* contains two important classes, these are Beam and Explosion. These two particular classes are separated from *playScene.js* as they are called multiple times throughout the entire game.

Research

The area that I decided to focus on was game physics. Due to this, I chose to create an arcade shooter, so that I could create different effects in the game. When the player's ship fires a beam that hits an enemy ship, it is destroyed, and the player's score increases. When the player's ship collides with an enemy ship, the player's ship explodes, and the game is reset. The greater player's score, the faster the enemy ships move left and right, in order to make it harder to hit them.

Critical Review

The game runs smoothly when opened in a browser, allowing the player to move their ship, as well as shoot at incoming enemy ships that continuously spawn in. The game in its current state could do with some extra challenges, as it is definitely on the easy side compared to the majority of games. One such feature could be enemies shooting back at the player. Another feature could be that certain enemies take multiple hits until they are destroyed.

Assets

The background of the game is taken from a Wikipedia entry of a galaxy known as Abell315. I chose to use it for my game as it is free to use, share and modify, whilst fitting the theme perfectly. One problem I had with the original image, however, was that it was very obvious where the image repeated when the screen scrolled down. This was initially distracting whilst playing the game.

In order to overcome this, I copied the image into Photoshop and overlaid the image on top of itself twice, once at the top and once at the bottom. I then blurred the multiple overlapping portions of the images, then cropped the image back to its original size. This allowed for a seamless transition when the screen scrolls down during the game, stopping the player from being distracted.

3 of the 10 ship textures were taken from an existing GitHub example (see references for details) and modified to fit the theme of the game. The remaining 7 ship textures are custom made, using the other ships as basic templates. I also added custom favicon images, displaying icons based on where the game was being run from, whether it be on an Apple, Android or Windows platform.

I downloaded a free to use “1 Hour of Space Music” in order to add some ambience to my game. The file size for this however was extortionate in comparison, so I used audacity to trim and fade the audio to create a smaller, more reasonable loop of audio.

References

GitHub Example:

https://github.com/ansimuz/getting-started-with-phaser/tree/master/part%206%20-%20Physics

Background Image:

https://it.m.wikipedia.org/wiki/File:Abell315_(ESO).jpg

Background Audio:

https://soundcloud.com/host7/1-hour-of-epic-space-music-cosmos-volume-2-grv-megamix