Hattori

"A vertical-scrolling space-shooter"

Edward Eldridge (G00337490)

September 19, 2018

Hattori will attempt to re-invent the traditional vertical scrolling space shooter genre by retaining the core gameplay while introducing more interactivity and decision making with the addition of a resource management system, on-screen gestures and a detailed upgrade system.

Contents

1 Overview

Traditionally, a vertical scrolling space shooter consists of the player controlling a spaceship while 'aliens' or enemies attempt to destroy the players ship. While this kind of gameplay can be fun, for me personally it quickly loses it's charm as aside from controlling the ship and shooting, there is little else for the player to do. With Hattori, I intend to add an extra layer of depth to the core gameplay of this genre.

1. **Resource management**
    - Link the player's score with the use of special abilities.
    - Introduces risk/reward. The player must decide if it is necessary to use a special ability. Overuse of abilites will result in a lower score but not enough use could result in the player losing. Player must find a balance.
2. **Gesture-based combat**
    - Rather than the traditional method of defeating enemies through just moving and firing, I intend to introduce abilities that require on-screen gestures to be performed. For example, drawing a circle could represent a bomb, or a cross a sword strike.
    - This addition will provide more depth to the game and give the player an opportunity to interact more with the game. Maybe the player is poor at managing his energy/score but can quickly perform gestures. giving them an advantage vs. other players.
3. **Player upgrade**
    - To keep players interested, I intend to introduce a way of augmenting or upgrading the player's ship.
    - Upgrade systems and character development are very important aspects to any game. They provide the player with a feeling of accomplishment and progress. They also keep the game fresh and exciting as more ways to play the game are introduced as the player progresses through the game.

2 References

Improving Player Choices by Tracy Fullerton, Chris Swain and Steve Hoffman

3 Specification

3.1 Genre

Hattori will be a score-based, endless, vertical-scrolling shooter.

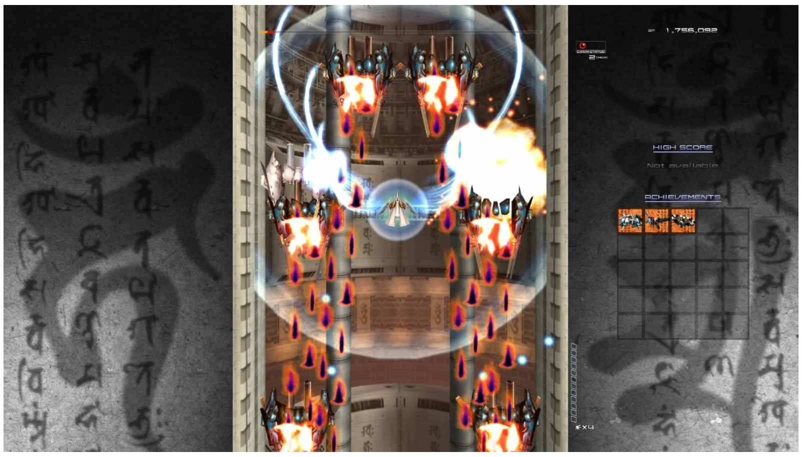

Figure 3.1: Gameplay from Ikagura(1998)

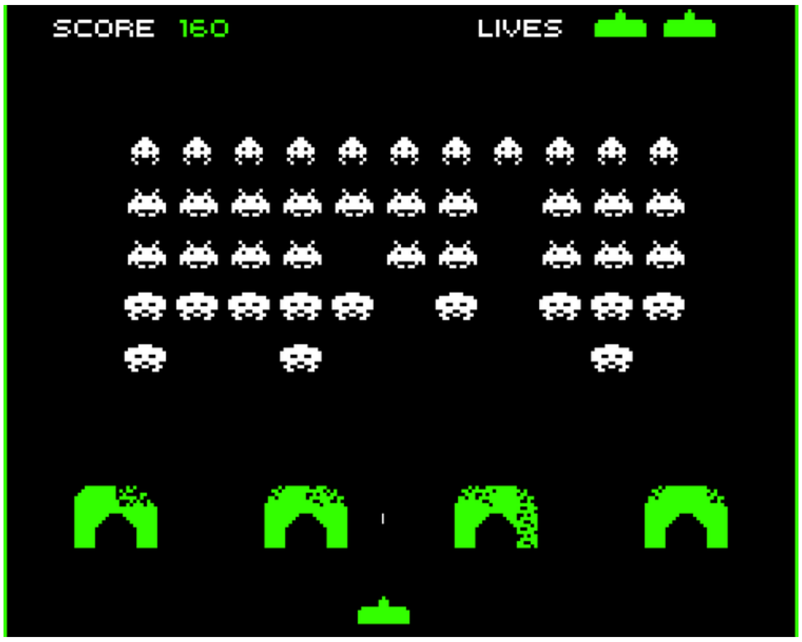

Figure 3.2: Gameplay from Space Invaders(1978)

3.2 Art Style

The art style of the game will be mainly sci-fi/space focused but will take inspiration from lots of different sources.

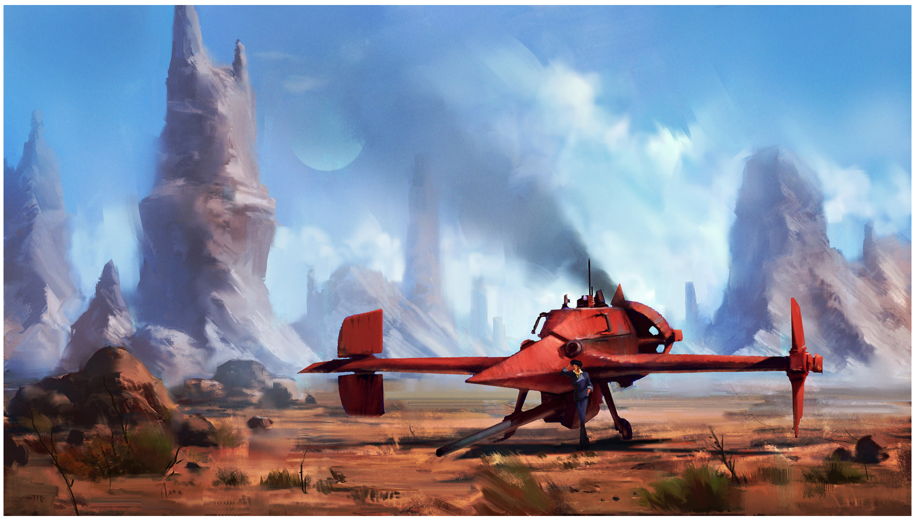

Figure 3.3: 'Cowboy Bebop' concept art

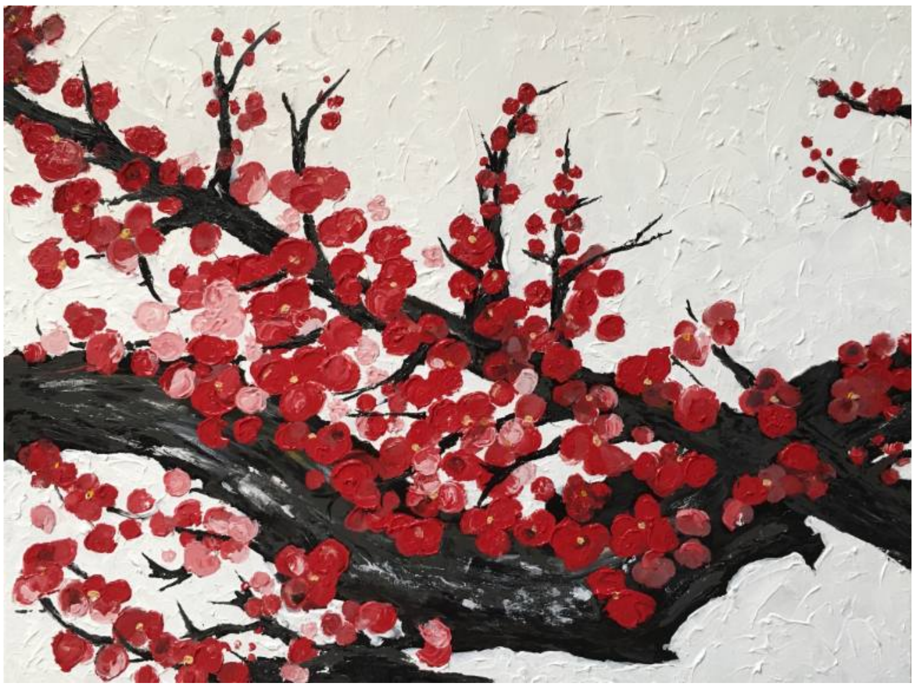

Figure 3.4: Cherry blossom painting

4 Gameplay and Game Setting

4.1 Story

The story of the game will be kept simple to instead focus on gameplay. While exploring space in a ship called the 'Hattori', the player is teleported to an unknown location in space. The player is then confronted by horde of enemies they must defeat to survive.

4.2 World/Environment

The game will be set in space. As such, the background and enviroment may feel a bit bland, however due to the nature of the story I can expand on this and hope to make the player feel like they are making some kind of progress through space instead of feeling like a static object on a scrolling background.

Figure 4.1: Art from Ikagura(1998)

4.3 User Interface

The user interface of the game will be broken down into 3 sections.

- A large health bar so the player will be able to know their current health without moving their eyes away from the action.
- The general gameplay area, where the player will be controlling the ship and where enemies will be spawning.
- A third area for additional information such as the player's score, upgrades, special abilities etc.

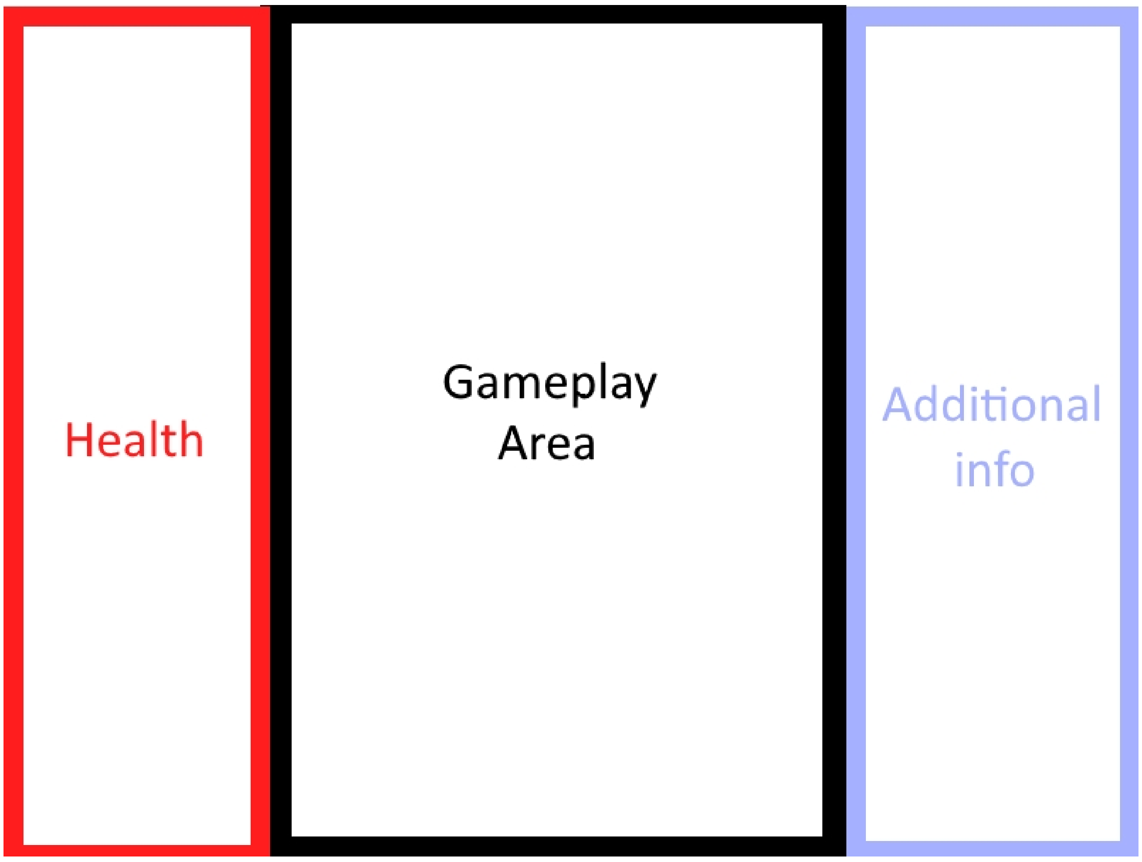

Figure 4.2: UI Example

4.4 Main Objective

The main objective of the game is to stay alive and complete all levels. At the end of each round/wave, the player will have a chance to rest and take count of his resources, health and current upgrades. This will give the player a break to either stop/playing continue. This allows the player to not feel like playing a round is a time commitment and makes the game generally more accessible to a wider audience.

4.5 Core Mechanics

The core mechanics of the game will involve the player controlling a spaceship and shooting at enemies to kill them, before they kill the player. Additional mechanics such as a resource management system, upgrades and gesture-based attacks will be added onto this core mechanic to keep things interesting and varied. Simple things like being able to upgrade your ship, for me personally, add a lot of depth to a game. It makes me want to continue playing, knowing that the experience I have now, might not be the same experience I will have after say an hour of gameplay.

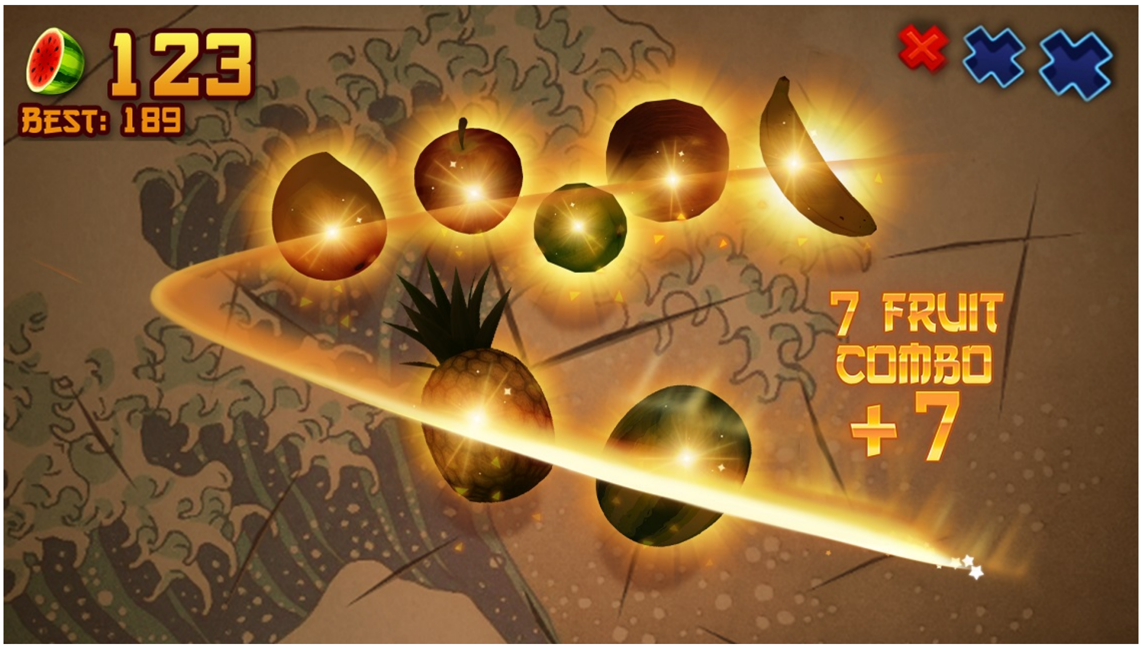

Figure 4.3: Gesture-based gameplay from 'Fruit Ninja'(2010)

4.6 Controls

I intend for the game to be controlled with mouse and keyboard. However, I would like to build the game for Android and for the Universal Windows Platform also.

I want to combine mouse with keyboard for this game as traditionally, vertical shooters just utilize the keyboard. However, this leaves the player with a free hand. To remedy this, I propose the player use the keyboard to control and fire from the ship and use the mouse to perform on-screen gestures.

This adds another layer of interactivity for the player, making the gameplay feel more engaging and fast-paced.

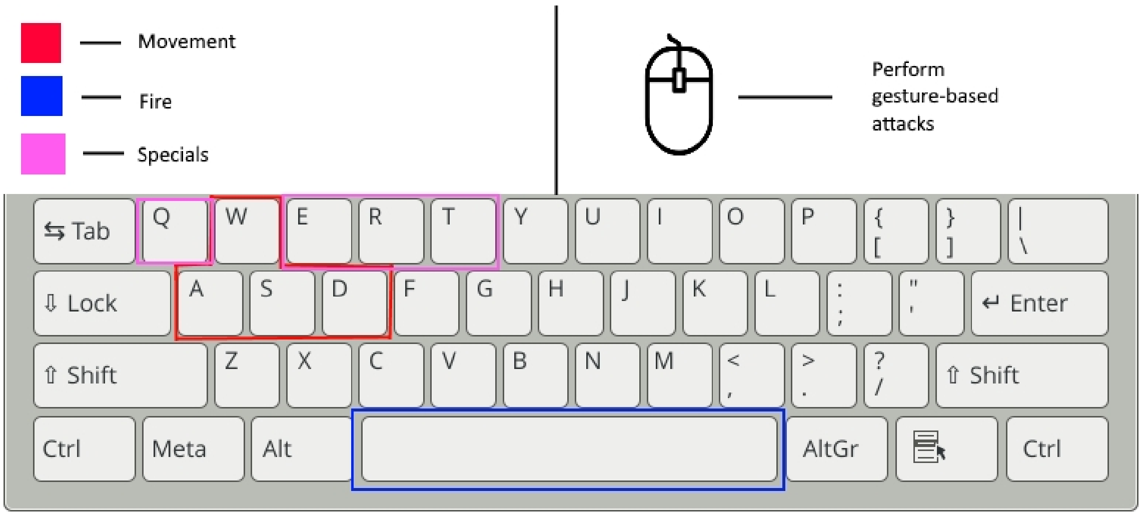

Figure 4.4: Mock control scheme

5 Front End

description of front end such as start screen, menu screens,..

5.1 Start Screen

5.2 Menus

5.3 End Screen

6 Technology

This game is designed for the Universal Windows Platform but will hopefully be available in both desktop and mobile versions.

6.1 Target Systems

Android, Windows desktop and Universal Windows Platform.

6.2 Hardware

Mouse and keyboard or an accelerometer/touch screen device.

6.3 Development Systems/Tools

1. **Programming**
    - The game will be developed in the **Unity** engine, using the Unity editor.
    - **C#** wil be the main programming language, using **Visual Studio 2017/Visual Studio Code** as an IDE.
2. **Design**
    - **Paint.net** will be used to to design the art/assets.
    - **Audacity** and royalty-free music/sound effects will be used to create the audio component.
3. **Source control/Documentation**
    - **Git** and **GitHub** are the technologies being utilized for source control.
    - This design document and documentation will be created with a combination of **LaTeX** and **Markdown**.

7 Timeline

Milestone	**Description**	**Date**
	Official Start Date	12.09.2018
1	Basic gameplay	31.09.2018
1	Control scheme	7.10.2018
2	Upgrade/resource system	14.10.2018
3	User interface/menus	31.10.2018
4	Music/SFX	7.11.2018
5	Bugfixing	21.11.2018
5	Documentation	24.11.2018
	End of Project	31.11.2018