Interactive Media Design

Assignment 1

Fantasy Zone

Evaldas Cesna X00112310

David Kavanagh X00109296

Introduction

The game chosen for the CA is Fantasy Zone, the game was released by sega in 1986. It was ported to many systems including the Sega Master System and the NES. The game plays similarly in all versions however some versions like the Master systems omitted features like the radar. The game was remade for the PlayStation 2 and later received a digital re-release on the Nintendo virtual console in 2008. [1]

The main character of the game is a sentient spaceship name Opa-Opa and is considered to be one of Sega’s first mascots. Today he is remembered with games such as project X Zone with him being a playable character.

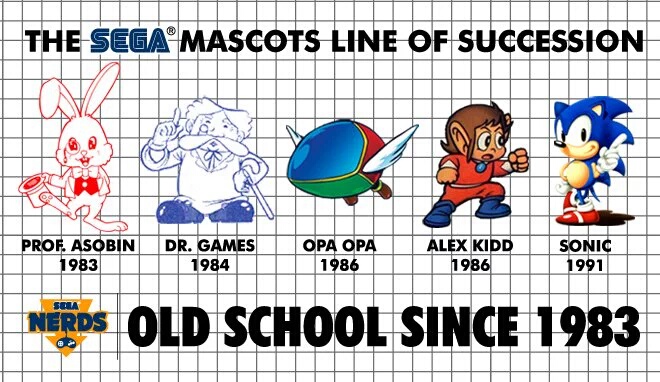


Fig 1 Sega Mascots

**Approach Taken**

The Approach taken to recreating the game was to pick a version of the game and try to recreate its first level with the resources available to us. The site “Spriters resource” was used to acquire the sprites needed. Sounds and music were taken from Youtube and “Sounds Resource”.

For the project our approach was to recreate the game at more than surface level, the game's first level and gameplay was recreated, but despite its age Fantasy Zone is quite modern with an upgrade system and unique enemy AI, these features were recreated to show Fantasy Zone’s very modern features that add a lot of complexity to the gameplay.

To get a feel for the gameplay we used a emulator to simulate the Master System version of the game, this gave us insight into the game's difficulty, pacing and atmosphere. We wanted to recreate this a closely as possible to show the game's unique charm.



Fig 1.1 Player Sprites

**Gameplay**

The Gameplay of fantasy zone is based around reaction speed and timing, the player must avoid enemies and destroy the enemy bases in order to spawn the boss, when defeated the player can move to the next level.

The player can shoot projectiles and bombs to destroy the game's many enemies, these enemies spawn infinitely until the player destroys the bases.

The player has a variety of tools to help beat the game, this is shown by the game's upgrade store, the player can purchase several upgrades such as permanent upgrades to the player’s speed or temporary buffs with projectiles and bombs, these provide an extra challenge and layer to gameplay as the player must learn the enemy patterns and decide which upgrades to buy accordingly.

To implement these functions we used several scripts that control the player's movement and ability, a unique feature to Fantasy Zone is the player’s ability to land this was recreated by setting a collider on the ground which triggers an animation switch to Opa-Opa’s standing position.

To implement the game's upgrade system we implemented a scriptable object system that allowed us to create unique objects that take in a prefab and a set of values allowing us control over the stat values.[2]

To acquire upgrades the player must first collect a set amount of coins to spawn the upgrade balloon, enemies drop coins however the low drop rate makes it unlikely that the player will use the more expensive upgrades more than once a playthrough.

Items such as the rocket engine also come with drawbacks, in the rockets case the speed value is too high for the player to control making it not worth the price unless the player is exceptionally skilled.



Fig 1.2 Upgrade Menu

**Goals**

The goals of the game are simple, the player must destroy the bases and collect coins to increase Opa-Opa’s speed and firepower for when the challenge ramps up. The goal of the first level is to defeat the bases and summon the boss. The base’s are of varying height and spread throughout the level when the player defeats them the first boss is spawned.

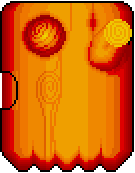
Fantasy Zone base dorarinfura.gif

Fig 1.3 Boss & Base for level 1

To implement these goals we statically assigned the bases to their in game locations upon death of the last base a trigger is set that sets the boss to be spawned sending the player to the bosses location and locking his movement to face the boss.

**Player Interaction and Perspective**

The players interaction is solely as the ship Opa-Opa the games view is limited to a set amount behind and in front of the ship the player has a limited view of the game, enemies come in from both sides of the screen to surround the player.

The game’s objects such as bullets and enemies disappear when not in the camera's view, however more enemies spawn from the camera’s edge meaning the screen is almost always filled with enemies the exception to this is the bases who stay in place when off screen or when the player loses a life, the player can also destroy the bases to stop them from spawning their special type of enemy..

The players perspective is that of a side scroller, while in the air the player floats from left to right but if the player lands on the ground the scrolling stops. The player can also choose what direction to fly in.



Fig 1.4 Opa-Opa landing

To implement this we used a script that ties the camera to the player, and another script to check if an object is rendered on screen and destroy the object if necessary.

**Genre**

The Genre of Fantasy Zone is known as a Shoot em up also known as Shmup or STG, the genre first started in 1978 with space invaders, the game featured one dimension the player could only move left or right making at an on rails shmup,

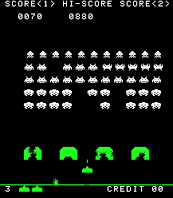


Fig 1.5 Space Invaders the First Shmup

Where fantasy zone deviates from the genre norm is allowing the player to choose up, down,left and right as well as the ability to land.

The game's art style is also a major deviation from the normal of shoot em ups as the game's design is noticeably cute compared to most sci-fi based shmups, this was known as the “Cute-em-up” subgenre. [3]

This genre was known for its bright color pallette and cute enemy designs.

Part 2:

**Evaluation**

In the modern context the shoot em up genre continues to this day with games like Geometry wars and Ikaruga. By modern standards Fantasy zone holds up quite well with a variety of enemies and features like the upgrade system found in most mobile games, however the genre has continued to grow games like Ikaruga that take the bullet hell approach filling the screen with bullets for the player to dodge, Geometry wars takes another approach by adding a 3D grid element for the player to fly around as well as online modes.

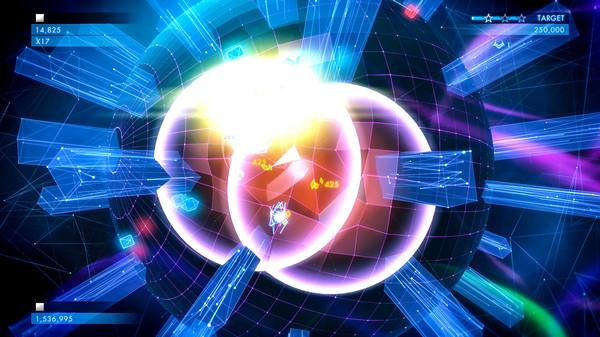
By modern standards the games art and music direction is lacking compared to modern HD graphics and music but makes up for it with a unique art style and catchy melodies.

Fig 1.6 Geometry wars Example of a modern Shmup

**Improvements**

**1.Online Co-Op**

One improvement that could be made to Fantasy Zone is the addition of online features, many modern games feature some form of online mode, even games within the genre such as Geometry wars, An online feature would give the game more variety like restricting the player’s ability to earn coins but adding an assistant to help with enemies or put the players in competition to one another to see who collects more coins.

With Unity this feature can be easily implemented, Unity provides the framework to add a network manager to the game, as well as provide developers with servers to host games.

To implement the player adds the network controller to the game and implements Host and Join Buttons, to ensure that the players do not control both characters a unique identity needs to be applied to each player on the network with the “isLocal” player functions.

The player’s actions will also need to be synced up with the same functions,adjusting for latency, and loading in prefabs into the network manager.

**2.Particle Effects**

To recreate the player death animation we used the unity particle system, however most animations are done with the games 2D sprites, items like the rocket engine produce a flame effect with the animator to improve upon this and modernize the graphics we could use flame particles and add this to the player prefab rather than a 2D sprite based animation.

Particle Effects could also be used to generate more complex bullet patterns that the player has to dodge improving the challenge of the gameplay as well as the game's graphic quality.

Particle Effects could also be used to add moving liquids or cloud effects to the game, unity’s particle system allows us to easily edit the particles size,shape and materials.

Some additional particle effects would make the game seem less static and improve the game's visuals by adding fluidity and motion.

**3.Player Customization**

Another feature that is quite common in modern gaming is player customization, games such as Overwatch and Team Fortress 2 encourage player uniqueness with unlockable skins and hats to add to the player’s model which make the player feel more connected to the game with their own unique avatar increase the player's interactions with the game.

To implement this in unity we can use more scriptable objects that create cosmetics of different types such as skins or unlockable decorations and rocket trails for the player’s ship, the scriptable objects would take in a prefab that is assigned a sprite and some values, the player can access these through a shop UI interface which accesses an array of that upgrade type and changes the current value based on what the player purchases.

**4.Loadouts & Random Generation**

A trend in gaming that is gaining popularity is the return of the roguelike genre ,this genre consists of randomly generated objects, either player's builds, enemies, loot found or any combination.

To implement this in Unity we could take the existing upgrade shop and replace it with a random number generator assigned to the player,this would mean the player can spawn with any available weapon type, more weapons could be added to further improve this feature adding more variety, the enemy bases could also spawn in random locations leading to a different level each time the player revisits.

Each enemy could also spawn a powerup that buffs the player leading each play of the game to have a different build and load out by the end of the game.

These improvements to gameplay would bring Fantasy zone more in line with current trends with popular games such as the Binding of Isaac or Rogue legacy.

**5. Scrolling Backgrounds**

A common feature in side scrollers and still widely used among retro styled indie games such as Shovel Knight is scrolling backgrounds. Techniques such as parallax scrolling are common in modern 2D games.

In Unity this can be implemented with the moving of background layers at different speeds giving the illusion of depth to the player. In Unity we use and Orthographic camera and a script that is linked to the camera that moves the camera.

This basic feature is a must have for a 2D shoot em up adding a layer of depth to the graphics and especially important for representing clouds moving in the sky or meteors depending on a shoot em ups setting.

**References**

**[1]**<http://strategywiki.org/wiki/Fantasy_Zone>

**[2]**<https://unity3d.com/learn/tutorials/modules/intermediate/live-training-archive/creating-an-in-game-shop>

**[3]**<https://classes.soe.ucsc.edu/cmps080k/Winter07/lectures/shmups.pdf>

**Sources:**

**Sounds -** <https://www.sounds-resource.com/nes/fantasyzonesunsoft/sound/3979/>

**Sprites -** <https://www.spriters-resource.com/master_system/fantasyzone/>

**Game Reference-**

[https://play.emuparadise.me/playonline/SMS/Fantasy\_Zone\_(World)\_(v1.1)\_(Beta](https://play.emuparadise.me/playonline/SMS/Fantasy_Zone_(World)_(v1.1)_(Beta)/sound/89001)

[)/sound/89001](https://play.emuparadise.me/playonline/SMS/Fantasy_Zone_(World)_(v1.1)_(Beta)/sound/89001)