

Mirrored Fantasy

CST 499 - Capstone

Andrew Bell

8/27/2020

Table of Contents

Executive Summary	3
Part 1	4
Introduction	4
Project Goals and Objectives	5
Environmental Scan	6
Stakeholders and Community	7
Approach and Methodology	8
Part 2	9
Ethical Considerations	9
Legal Considerations	11
Part 3	12
Timeline	12
Resources Needed	13
Milestones	14
Risks and Dependencies	15
Final Deliverables	16
Usability Testing and Evaluation	17
Team Members	18
References	19

Executive Summary

Mirrored Fantasy is a new take on the high speed team based combat from MOBAs, one of the most popular eSports genres, with more of a focus on: diverse map elements, ability interaction, and player versus environment. The main goal is to have interesting gameplay with isolated mirror lanes which no other MOBA has, with three dungeons of different types. A player versus player dungeon with the objective of this being the main interaction between teams. A player versus environment dungeon which shows player skill and knowledge of game mechanics. Last is a simple puzzle dungeon that has a mix between player versus environment and player versus player which can serve as a secondary point of contention between teams.

The purpose of this project is to fill a gap in the MOBA game genre with a greater focus on player versus environment elements and direct lockdowns of the other team's progress. Bringing player versus environment skill to a genre that normally keeps its focus in player versus player and a focus on the ability interaction that is common in RPGs. This is targeted at fans of both the RPG and MOBA genre as well as just gamers in general.

Part 1

Introduction

Mirrored Fantasy (working title) is a competitive moba style game where teams are separated in their own lane where each team must complete three dungeons. A player vs player dungeon where a team member must go through narrow corridors that is exposed to fire from the other team. A puzzle dungeon which the team must solve with light exposure to the other team. Finally a player vs environment dungeon with waves of enemies and a boss fight to complete. Each player will have their choice of magic based abilities ranging from a standard fireball to seeds that grow into plant turrets. Each of these skills will have a direct counter from another magic type. Each type of magic has two different skills, a projectile based skill and a terrain based skill.

The problem this project is solving is that the MOBA (multiplayer online battle arena) genre has grown stale with most games having the same formula of three lanes and a jungle where you just push towers till you destroy the other team's base. The solution to this problem that this project is going to implement is remove the standard map and put teams in isolated lanes where there is one focus on team interaction in the player vs player dungeon, and bring ability interaction and counters from popular RPGs such as Pokemon's type system (Pokemon Wiki) or Divinity Original Sin's environmental effects (Divinity Wiki).

Project Goals and Objectives

Goals	Objectives
Balanced gameplay	Ability type system with the base of a rock paper scissors balance (similar to Pokemon's type system) where each type has one that is good and bad against
Interesting map with mirrored lanes	A map with 3 dungeons with different objectives in each of the 2 mirrored lanes
Goal for the game is to be the team to finish all 3 dungeons first and put their objects on the victory monument.	A player vs player dungeon where the person running is very exposed to attacks from the other team
	A player vs environment dungeon where the players have to combat, most likely will be similar to a bullet hell or a raid boss style boss monster.
	A puzzle dungeon where the player must solve a basic puzzle to get the objective.
An easily extensible 2d Javascript game engine	A Vec2 class for holding all of the vector and point information
	A Map class that holds the map information, and the entities that are on the map
	A Collision engine that calculates when something collides with another object and calls the entities hit.
	A entity that holds all of the information about an entity (name, health, speed, position, look direction, hitbox, image source)
	A lot of other classes that build off of these
Server that feels quick and responsive.	NodeJS server running express to serve the client code and run the game server. This also encourages code reuse between the client and the server

Environmental Scan

This game's overall structure and mechanics is based off of Minecraft's Race for Wool (SkiTrip), where there are two mirrored lanes that will have teams race to be the first to complete three objectives and bring back the objective completion reward to the victory monument.

This project changes the formula by changing from a first person perspective to a top down perspective, where we will be taking the control scheme from Battlerite which "is an action-packed Team Arena Brawler focused on competitive PvP combat" (Battlerite) which has a standard MOBA (Massive Online Battle Arena) perspective of top down, except it has movement on "WASD" instead of mouse movement. We are going to have left click be a basic melee attack, right click be a bow charge style ranged attack, "Q" be a ranged projectile as a damaging ability based on one of the three types, and "E" a terrain placing ability again based on one of the types.

The project is going to have a rock paper scissors balanced in its abilities, with fire, water, and grass, which unintentionally lines up with the 3 main types of Pokemon, and like Pokemon we would like to expand types for abilities in the long term (Pokemon Wiki), however these types will only have two options a projectile that generally does damage and a secondary effect on hit and a terrain placement.

Stakeholders and Community

The stakeholders for this project would be the game developers. The amount of time and resources that will be used to create this game would determine how successful and how many features that would be included in this game. Further interest in this project can result in adding additional features as well as more content such as more maps and puzzles. There may also be more different game modes that use this game engine.

The community that would be affected by this project would be the players of this game and possibly community developers that plan to expand upon this project. Having a smooth and interesting game would help with the game's popularity and willingness of others to play it. Since this game touches upon the multiple aspects of competitive multiplayer while dealing with several obstacles and puzzles, this should appeal to multiple audiences. The popularity of the game will create a major factor of potential expansions of the game or even ideas for future games. Some of the possibilities includes player made maps, new game modes with new abilities and puzzles.

Approach and Methodology

In order to create this game, we will need to first take a look at games with similar features that we can use as an idea to implement into this game so that it would be intuitive and easy for potential players to get into quickly. Then we would take a deeper look at the different abilities that the players can use. These abilities should be able to counteract one another while balancing their effects and usages so that ideally there would not be one ability that is too strong.

Next step would be to determine what puzzles and obstacles that would be used in this game.

There are 3 different obstacles that each team must clear in order to win the game. The first one would have heavy influence with the other team where they can interact with and interfere. The second would be more difficult to interact with but still doable. And finally the last one would be free from opposing team's interference.

We would then need the images that would go with the player models, the abilities, and also the map resources in order to create the map.

Afterwards, we would have to work with the interactions with the different parts that the player model has with the map and abilities. This would include collision detection for the player models with the map walls and interactables that the player can work with to clear the objectives. The abilities that the players have should also affect the players. Some of the possible effects that should be considered are pushing, slowing, and freezing controls.

Finally, all these parts would be put together in order to create the game. Heavy testing would be required in order to test for bugs and check for improvements to the game.

Part 2

Ethical Considerations

One of the major concerns that involves video games is addiction. Addiction in video games mostly comes from either luck based mechanics or long term commitment in order to progress. Some of the luck based mechanics that are commonly used in larger games includes loot boxes and random loot drops. The other method of long term commitment would be like having a character saved onto an online server where progress is saved with long term goals of character growth and progression or having a ranking system where players compete to get the best score or ranking.

To mitigate the concerns about addiction, this game will focus on being a competitive game where it is easy for the players to pick up and play a quick game anytime without being concerned about dealing with progression, leaderboards, or loot boxes.

In terms of general accessibility this game is going to be an easy to run browser game so anyone with access to a computer should be able to play it. The goal is to make it light weight and well optimized while still being able to handle complex game mechanics and environmental interactions. This is part of the reason why the game is going to be 2d rather than 3d, as it greatly reduces the CPU load and gives greater flexibility when coming up with ability interactions that can be made.

As this is a game there will be lots of play testing needed so we will be keeping the privacy of our testers to a maximum and keep all feedback anonymous. Being unsure as to whether player

testing falls under human testing we will look into the policy on human subjects in research to be safe.

Legal Considerations

The major legal considerations for this project will be getting art and sprites that will display the player and abilities, as well as map layout. Our solution would be to use sites such as itch.io and opengameart.org with their selection of free assets. Finding assets for the game would be simple and quick to find in order to fit the theme of the game while giving credit to the artist of those art assets. Any other assets that would be needed for this project would be custom made using tools such as Express, Node, and ES6 Javascript. The other minor detail that should be addressed is checking if the name of the project is similar to other games.

After going through the environmental survey, this project should be unique enough so that this game does not infringe on copyright.

Part 3

Timeline

8/29/2020	2d engine with server integration and basic map design mocked up. Character interactions with basic features of the map.
9/5/2020	Three main abilities done with their interactions with each other. Ability interactions with characters.
9/12/2020	Three dungeons done and giving objectives with game win detection on victory monument (Start of playtesting)
9/19/2020	Response to initial round of playtesting
9/26/2020	Refine abilities and movement mechanics
10/3/2020	Refine dungeon designs
10/10/2020	Final fixes and stretch goals
10/17/2020	Last touch ups for Capstone presentation and preparing presentation

Resources Needed

A computer and a modern browser to run the client with. I will be using Visual Studio Code for development, but any text editor works fine for coding Javascript. A server to run NodeJS.

Milestones

- Basic 2d engine
- Server client communication and syncing of canvas
- Ability interactions and collision engine
- Dungeon areas defined
 - Giving the player the objective flag when they reach a tile and remove it when they get to the victory monument or die
 - Victory monument detects win when all three objectives are on it
- A puzzle dungeon
 - Design of puzzle and implementing it's logic
 - Gives objective when puzzle is solved
- A player vs environment dungeon
 - Creating boss assets and implementing it's AI
 - Gives objective when boss is defeated
- A player vs player dungeon
 - Platforming or dexterity challenge to make it more interesting than just running down a corridor
 - Gives objective when player reaches the end of the dungeon

Risks and Dependencies

Risks:

The biggest risk that I am worried about is having nodejs being fast enough to handle all of the collision calculations and communications when there isn't multithreading causing the server to be unresponsive. There is also a rush to get the minimal viable product as quickly as possible so that we can be running playtesting as soon as possible to make sure that balancing is good and the game is actually fun to play. Networking in a realtime game is also a concern in how it should be handled with client side prediction and if ping will cause too much rubberbanding.

Dependencies:

We need to get the 2d engine and collision detection in place before we can start making the abilities and movement of the players. Basic abilities and basic dungeon layout need to be done before we can start playtesting. A running NodeJS web server to distribute the client code.

Players playing the game need to be able to connect to the server in order to play the game on their browser. Currently it is planned for computer browsers only and mobile compatibility may or may not be added in the future.

Final Deliverables

NodeJS and ExpressJS server that runs the server side game loop as well as distributes the client side code. A full game engine, with consistent server client communication and 2d collision engine and map creator.

Usability Testing and Evaluation

Unit Testing will be run using Mocha and Chai run in node js to test classes, and integration testing will be using cypress to automate testing of multiple browsers and compatibility.

Getting at least 4 people together to play the game, have some of them stream or record their screens so that we can rewatch later and analyze what players movements and priorities look like. A satisfaction survey asking which parts of the game the players feel like could improve, any bugs they found, and which parts of the game they liked.

Team Members

This project will mostly be split up in weekly sprints where exact requirements will be laid out and requirements will be distributed among the team members based on which week in the timeline we are on.

Andrew Bell: Project Lead

Matthew Chan: Testing Lead and Quality Assurance

References

Battlerite. ““Battlerite Is the Best Team Fight You've Ever Had.”” *Battlerite*, Stunlock

Studios, 2019, arena.battlerite.com/.

Divinity Wiki. (2020, July 11). Environmental Effects (Divinity: Original Sin). Retrieved

August 27, 2020, from

[https://divinity.fandom.com/wiki/Environmental_Effects_\(Divinity:_Original_Sin\)](https://divinity.fandom.com/wiki/Environmental_Effects_(Divinity:_Original_Sin))

Pokemon Wiki. “Types.” *Pokémon Wiki*, 2020, pokemon.fandom.com/wiki/Types.

SkiTrip. *New? Here's How to Play*. 2017, wool.run/f/general/27.