COSC 412 Individual SPMP

Jordan Whittington

**Project Overview**

The purpose of my project is to create an easily accessible browser game that people can play and enjoy. The game will feature a username selection screen and the game features itself will include considerably basic, easy-to-learn controls. The game that I am going to develop is going to be a game where you traverse from left to right through the screens and defeat the final boss to win. Upon winning, you will receive your final score and that will be displayed on the global leaderboard.

**Project Deliverables**

* Server up and running
* Game Sprites fully functional
* Global Leaderboard fully functional
* Gameplay fully functional
* User registration fully functional
* Use Cases
* Requirements
* System Test
* Final Prototype

**Evolution of the SPMP**

I expect some gameplay mechanics to be explored and changed from the original idea as well as the overall goal of the game to possibly change as well. I expect for the overall powerups that I plan to implement and some of the key bindings to be changed. I foresee some different aspects of the game being completely scrapped or left for later, last minute implementation depending on the overall difficulty of the game itself.

**Management Objectives and Priorities**

Handle the easy stuff in tandem with the more time-consuming implementation i.e. creating sprites around the same time as creating a server. Seeing as how the gameplay is not dedicated on the development of the sprites, because a placeholder can easily be used for gameplay aspects early on.

I plan to tackle the gameplay early on as well as moving into how to implement a game using unity onto a website. I will need to start looking into hosting around this time, so I have enough time to get everything functioning appropriately

I will then move into the things of a little lesser priority such as the sprite creation and animation aspects of the game, as well as the overall art for the game will come a lot later in the development timeline.

**Assumptions, Dependencies and Constraints**

The project completely hinges on my ability to understand and be able to completely get a server up and running to have the game be played on. Gameplay mechanics and basic game development things are not too foreign for me.

There is a budgetary constriction surrounding game development. In terms of sprites and assets for the game itself, I would not like to spend any amount on this game for right now. I want to make this game something that anyone who has a computer and access to the internet can play because the game should not demand too much from the client users. To run the game, Unity Web Player will be used but it should not have to be downloaded onto the user’s computer at all.

**Risk Management**

* If I feel as if the project is becoming larger or is already larger than I originally expected, then I can try to reduce the overall scope of the project. By that I mean, reducing some gameplay mechanics, remove powerups, or even reduce the overall play area.
* The same as project size, if I notice that the complexity of the project begins to escalate and multiply then I will try to locate the source of the multiplication and see if there is a way for me to reduce the complexity and make it more simple while maintaining the same playability.
* Computer Issues: Convert to my laptop until my desktop issues are resolved
* Coding Issues: Use past documentation and other references to assist the coding process
* Every piece of the game will be tested multiple times and beta tested to remove any possible glitches in the code or anything like that.
* Using Risk Assessment charts to assess each different feature to be implemented into the game and the website

**Methods, Tools and Techniques**

Tools:

* Piskel – Free Software to create pixel-art sprites for playable characters
* Github – a means of sorting and maintaining my proper workflow
* Coding Language – C#, Javascript, HTML (possibly), CSS (possibly)
* Unity – Game Developing Software Used Throughout a lot of companies
  + An assortment of different Unity Asset Packages

**Techniques:**

I am going to follow the Agile Development Model because of its ability to adjust to the constant changes of the environment and such. Developing a game brings many intricate issues that can easily bounce off each other and create a completely inoperable project, so I would want a model that can adjust to those circumstances as well as adjust with the different ideas that I come up with for the direction of the game.

**Work Breakdown Structures**

**Use Cases**

|  |  |
| --- | --- |
| **Name** | Regular Commercial User Joins The Game |
| **Descriptions** | User enters the game and enters in a nickname to display their final score on the global leaderboard. There will be no registration |
| **Actors** | User |
| **Organizational Benefits** | Being able to access game |
| **Frequency of Use** | Quite common |
| **Triggers** | User enters website, |
| **Preconditions** | User has web browser and a stable internet connectivity |
| **Postconditions** | Access to the game and the assets associated with the game are loaded |
| **Main Course** |  |
| **Exceptions** | 400 – 500 internal errors  Login Failure |

|  |  |
| --- | --- |
| **Name** | Admin User Enters the Website |
| **Descriptions** | User can enter the leaderboard and either remove certain user’s scores or reset the leaderboard entirely |
| **Actors** | User |
| **Organizational Benefits** | Being able to access game and the leaderboard |
| **Frequency of Use** | Quite common |
| **Triggers** | User enters website, |
| **Preconditions** | User has web browser and a stable internet connectivity |
| **Postconditions** | Access to the game, the assets associated with the game, and the leaderboard that corresponds to the game scores |
| **Main Course** |  |
| **Exceptions** | 400 – 500 internal errors  Login Failure |