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 very basic, easy-to-learn controls. The ideal game that I want to create is a battle-royale style game (many enter but there is only one winner). The game would include melee weapons and ranged weapons along the map. The entire game will be played in a 2D assortment of buildings that are connected through wooden planks. There will be different accessible rooms, with a vision aspect where the player will not be able to see through walls or through doors and such.

**Project Deliverables**

* Server up and running
* Game Sprites fully functional
* Multiplayer 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.

**Reference Material**

**Definitions and Acronyms**

**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.

High Priority Functionality!!

**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.

Think a little more outside of the school project: As if you were actually creating this for a client

**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
* Think Client-wise and commercial

**Methods, Tools and Techniques**

Tools:

* Piskel – Free Software to create pixel-art sprites for playable characters
* Tiled – Free Software to create pixel-art styled maps for the gameplay itself
* Github – a means of sorting and maintaining my proper workflow
* Coding Language – C#, Javascript, HTML (possibly)
* Unity – Game Developing Software Used Throughout a lot of companies

**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** | User Joins The Game |
| **Descriptions** | User enters the landing page, creates an account or plays as a guest, joins a game (either random or from a list) |
| **Actors** | User |
| **Organizational Benefits** | Being able to access game and user registration |
| **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 |
| **Main Course** |  |
| **Exceptions** | 400 – 500 internal errors  Login Failure |

Alternative Courses for the different user styles (profiles)