

Software Requirements and Design Document

For

Group <12>

Version 1.0

Authors:

Abigail Centers

David Song

Paul Santora

Jason Hamilton

1. Overview

We will use the Unity engine to build a 2D computer game with multiple levels of difficulty, a scoring system to measure the player's overall performance, and a leaderboard to store the top 10 highest scores. Every level will begin with the player avatar at the start position of a new maze, where the objective of each level is to find an exit to the maze by moving the player avatar with the arrow keys. If the exit cannot be found within the allotted time, the game will end. Additionally, the player must dodge any enemies that move randomly about the maze and could hurt the player's health. Collectable items will also be placed about each level. These items primarily include timers, health, and power-ups. Timers will add 5-10 seconds to the allotted time, health will allow the player to recover from an enemy encounter, and power-ups will increase the player's overall score.

2. Functional Requirements

Basic Features

1. A player character (avatar) that can move around freely in an environment (level).
 - a. High priority
2. Walls that limit where the players move, boundaries of the level.
 - a. High priority
3. A Goal, a time based race/an exit area.
 - a. High priority
 - b. Additional victory requirements (Trigger activated, key found, etc.)
4. A home screen where the player can start the game and find game instructions
 - a. High priority

Advanced Features

1. Challenges, Terrain types, Moving walls, Spikes, Basic enemies, Projectiles
 - a. Medium priority
2. Scoring System, records the player's progress.
 - a. Medium priority
3. Difficulty, Health, Timer, Increased enemy speed, harder level, checkpoints, power ups etc.
 - a. Medium priority
4. Settings (e.g. pause game, quit game), Sliders for Music, Basic Graphics, Controls.
 - a. Medium priority
5. Leaderboards, local or online, Highscores and first place etc. Submit score button
 - a. Medium priority
6. Music: level music and backgrounds
 - a. Medium priority

Extras (Only after all other features are completed)

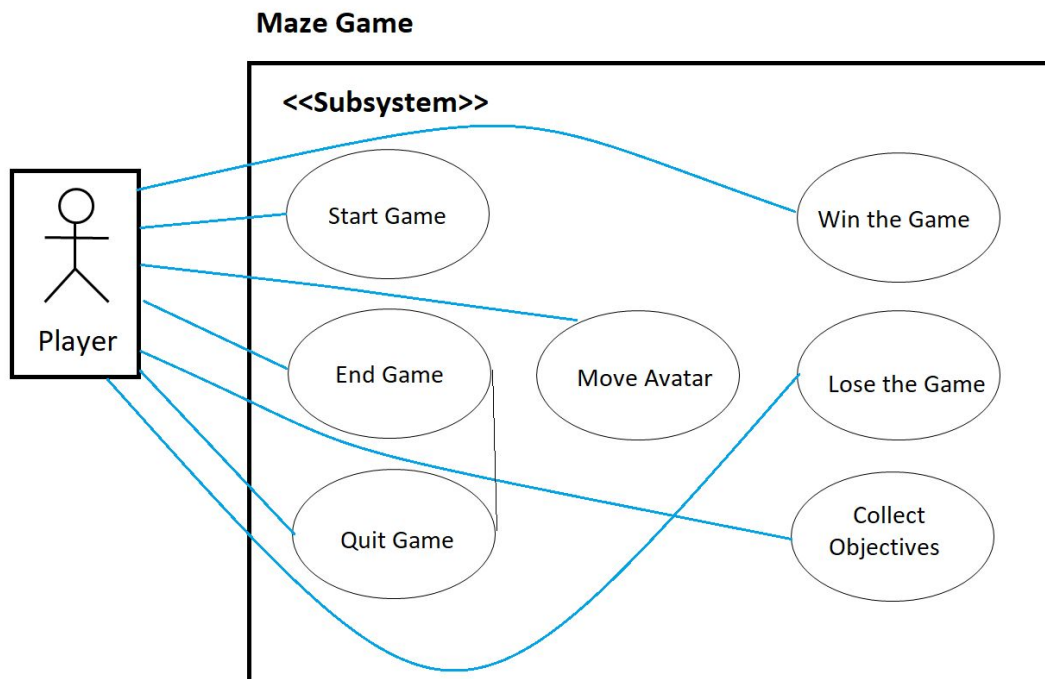
3. Multiplayer, local or online multiplayer activities
 - a. Low priority
4. Bosses/Bonus levels- unique levels with bosses or higher difficulty.
 - a. Low priority

5. Level Generating Algorithm- Automatically places walls and exits, infinite levels.
 - a. Low priority
6. Advanced Graphics- Textures, Animations, Sprites, Backgrounds
 - a. Low priority

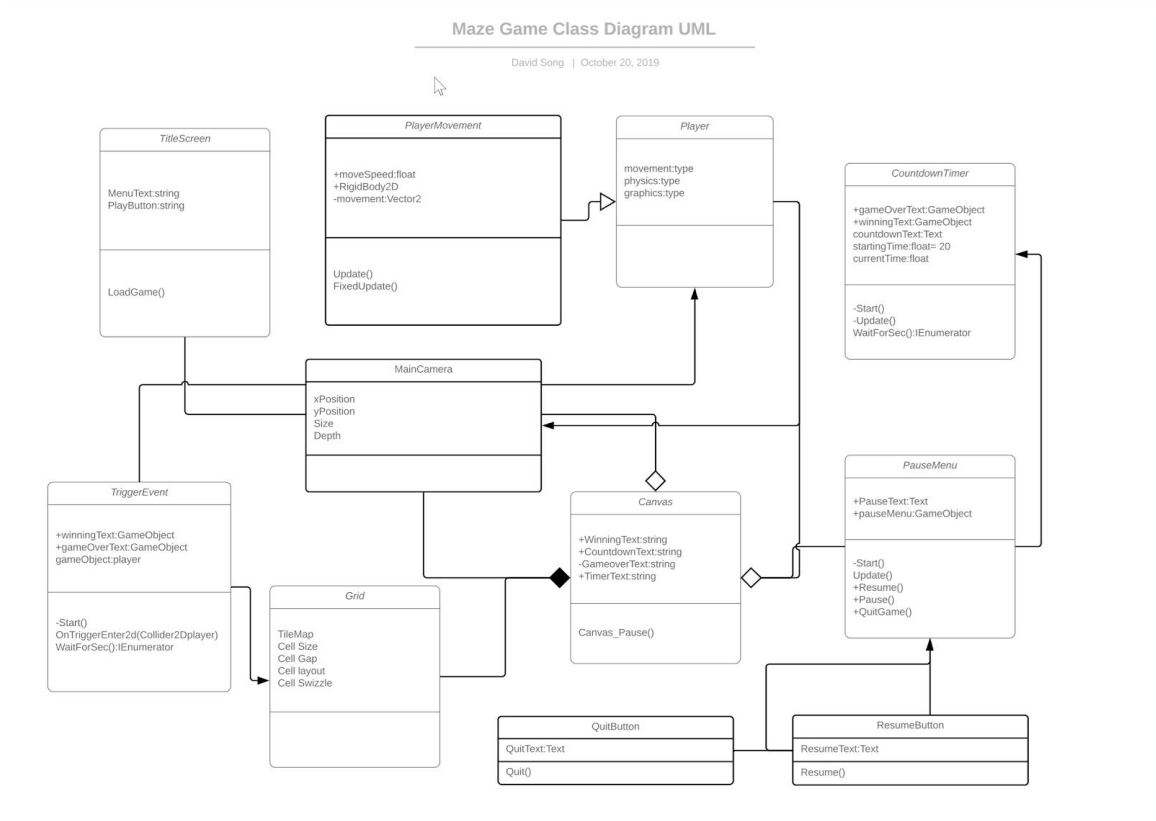
7. Non-functional Requirements

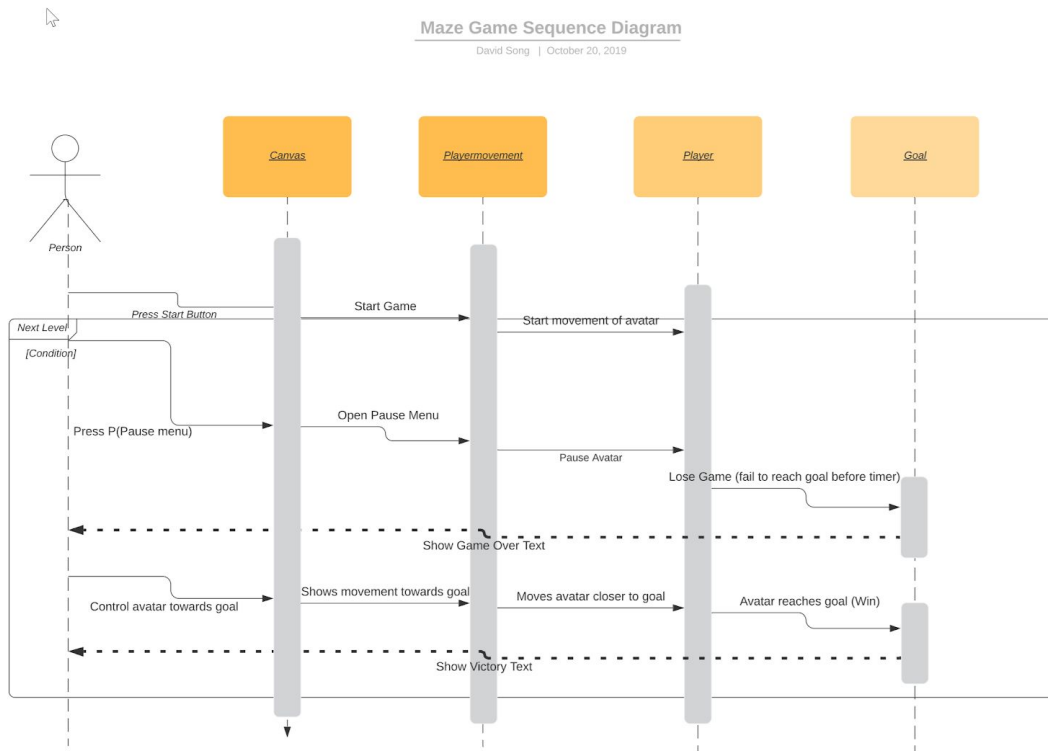
Minimum requirements for a system would be any level of graphics and player input capability. For security Unity can be used to compress and encrypt our application. For quality and performance standards our project will be designed to pursue a smooth gaming experience with no to minimal bugs. Our game would be able to run consistently and reliably without crashing.

8. Use Case Diagram



9. Class Diagram and/or Sequence Diagrams





10. Operating Environment

Unity allows us to port our game as an application. It allows us to support our project on Windows 10, macOS, Android, iOS.

11. Assumptions and Dependencies

Some factors that would be detrimental to our development can include compatibility issues (between mac and pc, different environments, and different Unity versions), Availability of group members (Time between busy schedules, work, exams, class). For external support on our project development there exists many game creation tutorials and free-use assets using Unity that can help us to develop our game. We are assuming that we will make consistent developments on our project and work towards completing our goals before each iteration submission. We are currently dependent on Unity's availability as a free software to develop our project.