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 for each level. 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 some amount of 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 - High priority

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

Advanced Features - Medium priority

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

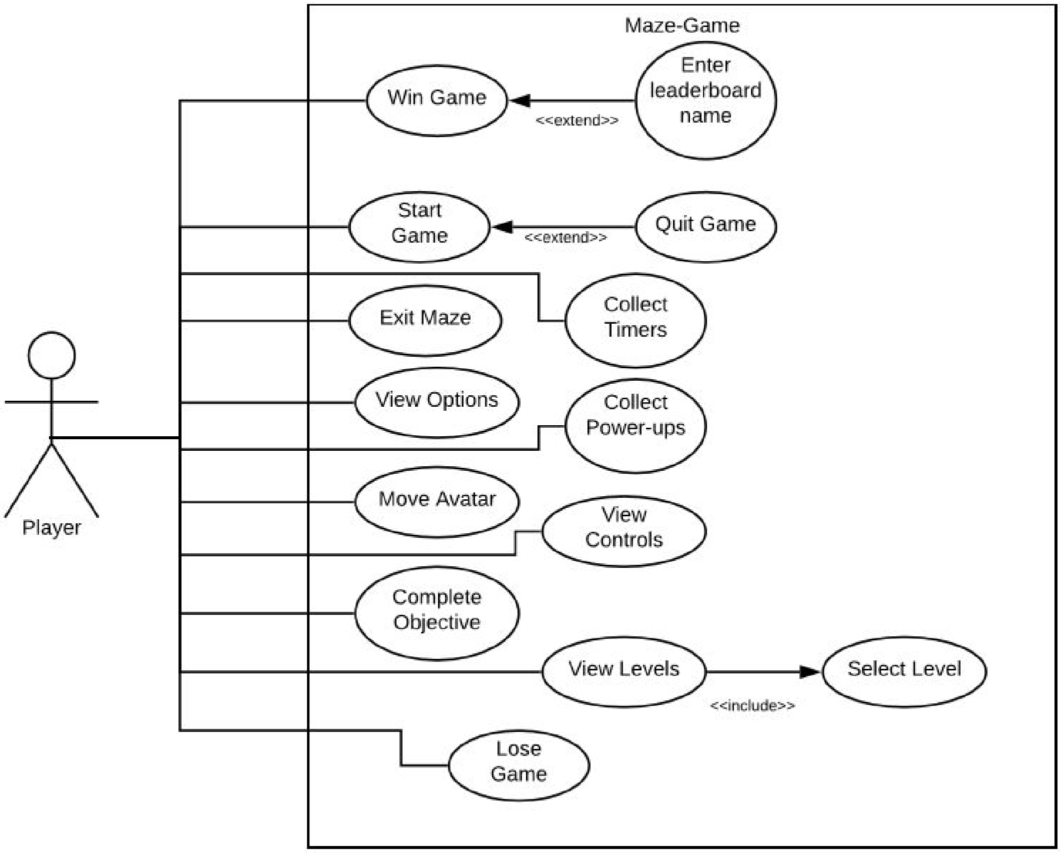
Extras (Only if high and medium priority features are completed) - Low priority

1. Advanced Graphics- Textures, Animations, Sprites, Backgrounds
2. Unique features- Enemies specific to a level, etc.

3. Non-functional Requirements

Leaderboard scores must persist between sessions.
Successive levels must have more objectives required to complete level.
For security Unity compresses and encrypts our application.

4. Use Case Diagram



Textual Descriptions

Name: Lose Game
Actors: Player
Entry conditions: Player is playing level
Exit conditions: Display dead menu
Flow of events: Start Game, Start Level, Press arrows keys, Run out of time, Run out of Hp, Lose Game
Special Conditions: Quit Game Early (Force close game)

Name: View Levels
Actors: Player
Entry Conditions: Enter Main Menu
Exit Conditions: Select a level choice, Quit Game
Flow of Events: Start Game, Enter Main menu, View Levels
Special Conditions:

Name: Complete Objective
Actors:Player
Entry Conditions: Reach one Lever
Exit Conditions: Reach all Levers
Flow of Events: Start Game, Play Level, Complete Objectives
Special Conditions: Fail to complete, Game Over

Name: View Controls
Actors: Player
Entry Conditions: Click on the Controls button in Main menu
Exit Conditions: Click on the Exit button
Flow of Events: Start Game, View Controls button
Special Conditions:

Name: Move Avatar
Actors: Player
Entry Conditions: Start Game, Select Level
Exit Conditions: Lose Game
Flow of Events: Start the Game or Select a Level
Special Conditions: Game freezes or Quits

Name: Collect Power-Up
Actors: Player
Entry Conditions: Start game
Exit Conditions: Power-up collected and score is increased by a constant amount (different for each level)
Flow of Events: Player starts game, begins level, and collects power-ups to increase his/her score
Special Conditions:

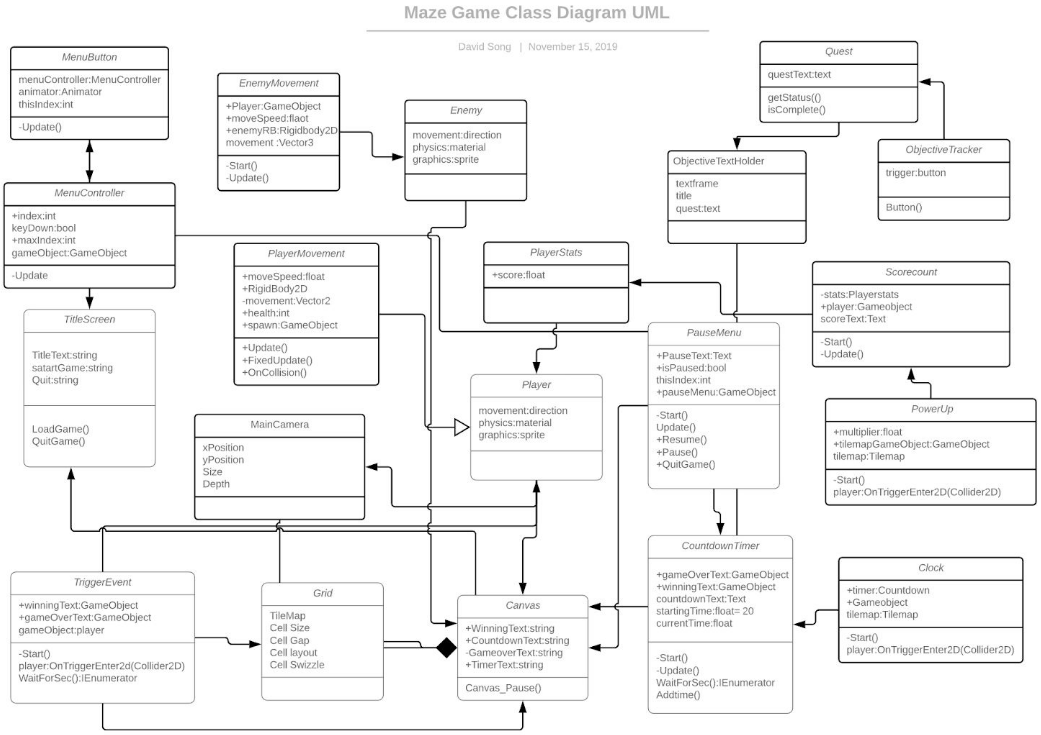
Name: Collect Timers
Actors: Player
Entry Conditions: Start game
Exit Conditions: Timer collected and 5 seconds added to the game clock for that level
Flow of Events: Player starts game, begins level, and collects timers to add time to the clock
Special Conditions:

Name: Start Game
Actors: Player
Entry Conditions: on the title screen
Exit Conditions: player clicks start game
Flow of Events: player enters title menu, navigate to start game and clicks on it
Special Conditions: While on level select, the player can jump to any level they wish by clicking on the level’s name

Name: Win Game
Actors: Player
Entry Conditions: player in level
Exit Conditions: player touches goal
Flow of Events: player enters level, and completes all objectives before time runs out and without touching an enemy, then touches goal
Special Conditions: If player has not completed objectives and touches goal, nothing will happen

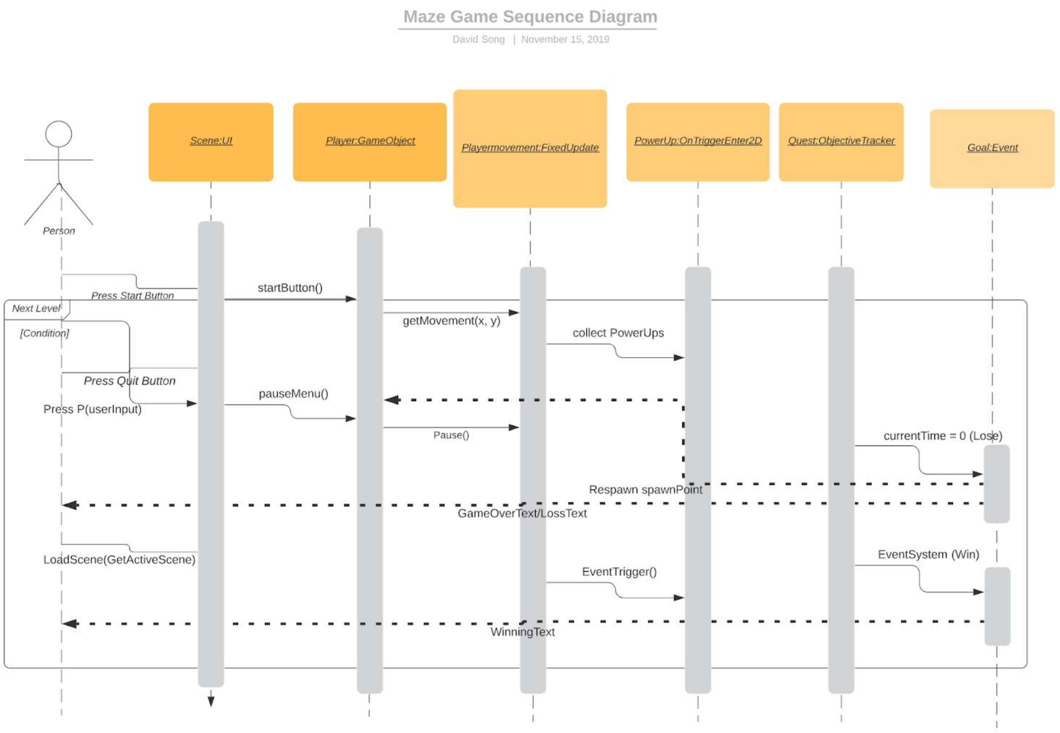
Name: Exit Maze
Actors: Player
Entry Conditions: In level
Exit Conditions: Click on quit game
Flow of Events: Player is in level, player pauses game, player selects quit game
Special Conditions: If the player dies, can select quit game on dead menu

5. Class Diagram and/or Sequence Diagrams



Maze Game Class Diagram UML

David Song | November 15, 2019



6. 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.

7. 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.