Software Requirements and Design Document

For

Group 6

Version 1.0

Authors:

Riley Arunakul

Gabriella Estevez

Braden Ford

Kevin Foughty

Mattan Pelah

1. Overview (5 points)

For our project, we will be making a simple 2d platformer with roguelike elements using the Unity engine. This 2d platformer will contain basic videogame functionality such as art, sound, objective based gameplay, options to tweak the experience, a point system, and more. There will be a finite number of levels - at this point in development the goal is 5 levels total - and replayability will come from various buffs and debuffs acquired from completing levels.

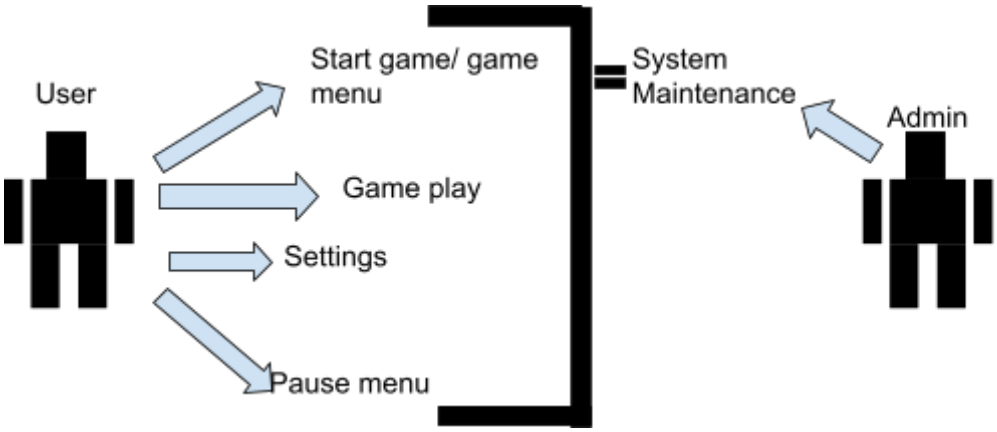
2. Functional Requirements (10 points)

1. The system allows players to start the game (H)
2. The system allows players to exit the game (H)
3. The system allows players to move the in-game character (H)
4. The system plays music upon startup (L)
5. The system plays sound effects when jumping, dying, etc. (L)
6. The system will have level transitions (H)
7. The system will allow a level to end (H)
8. The system will have animations for various actions (L)
9. The system will have a menu that allows users to restart the game upon death (H)
10. The system will have a main menu that allows users to start a level (H)
11. The system will have a options menu that allows users to tweak various options (M)
12. The system will have enemies that will patrol an area (M)
13. The system will have various buffs and debuffs assigned to the player upon completion of a level (M)
14. The system will detect a collision between player player and an enemy (H)
15. The system will detect a collision between the player and an inanimate object (H)
16. The system will allow users to deal damage to an enemy (H)
17. The system will have a camera that follows the player (H)
18. The system will end the game if the player does not have any more HP (H)
19. The system will have a timer that kills player character if time runs out (M)
20. The system will have a point system (M)

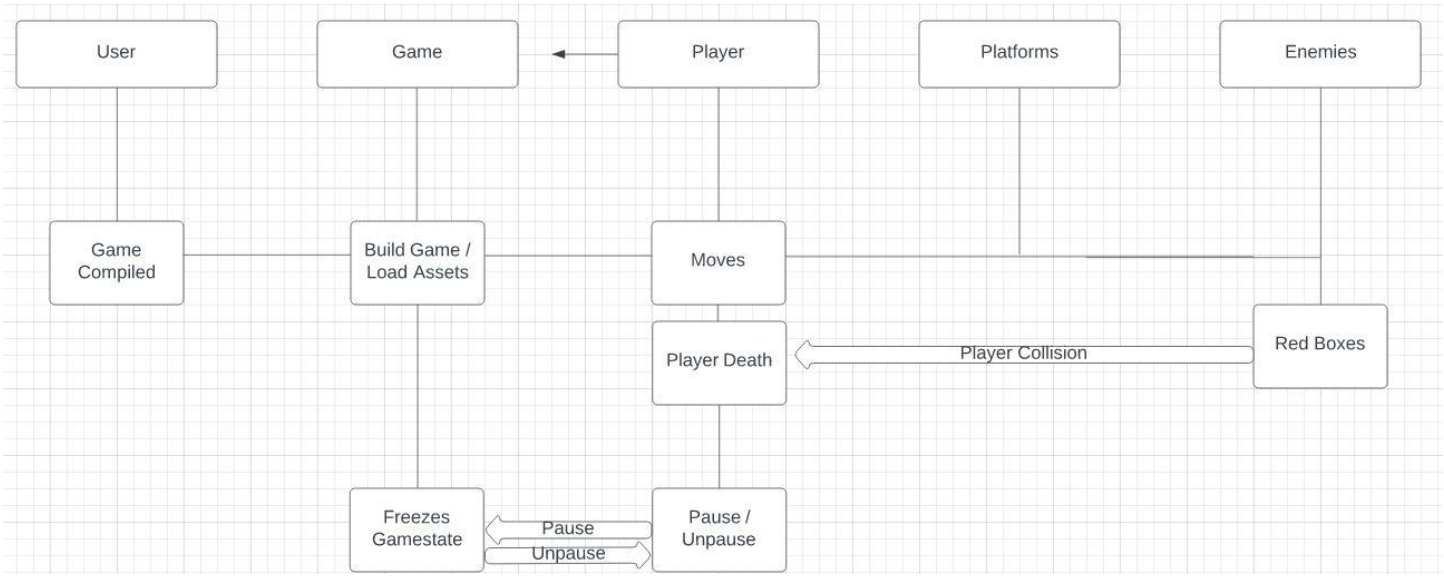
3. Non-functional Requirements (10 points)

1. Code should be kept clean and documented
2. Game should be able to run on Windows, MacOS, and Linux
3. Game should be able to run on systems regardless of hardware
4. Game should have fast response times
5. Game should not crash
6. The UI of the game should be easy to understand
7. Game should be fun

4. Use Case Diagram (10 points)



5. Class Diagram and/or Sequence Diagrams (15 points)



6. Operating Environment (5 points)

Our program is being developed in and for a Windows 10 environment including compatibility with later Windows versions as well. The hardware platform is personal desktop, however our program requires little graphical power and therefore should also work on personal laptops as well.

7. Assumptions and Dependencies (5 points)

We are assuming Unity and GitHub as softwares will not shut down as we use Unity as a game engine and GitHub for file sharing. Additionally, Unity tutorials were used for the basis of our code, so we are assuming that code is suitable for our game; note that we will update the code if problems occur.