Software Requirements and Design Document

For

Group <24>

Version 2.0

Authors:
Adeline Belova
Samantha Rivera
Eric Pengili
Mitchell Thai
Jonathyn Coleman

1. Overview (5 points)

This system is a 2D top down dark fantasy action RPG built in Unity, with C# as the primary coding language. The game features four primary elemental dungeons– earth, air, fire, water– with a final boss at the end after completing each dungeon. A central "peaceful" area allows the player a reprieve between dungeons and will also house the final boss battle at the end. The protagonist will be able to gain skills in all four elements as they beat each dungeon to unlock a new element ability, as well as tools that affect health, combat, and defense. The goal of the game is for the player to beat all four dungeons, gaining new abilities each win to beat the final antagonist.

2. Functional Requirements (10 points)

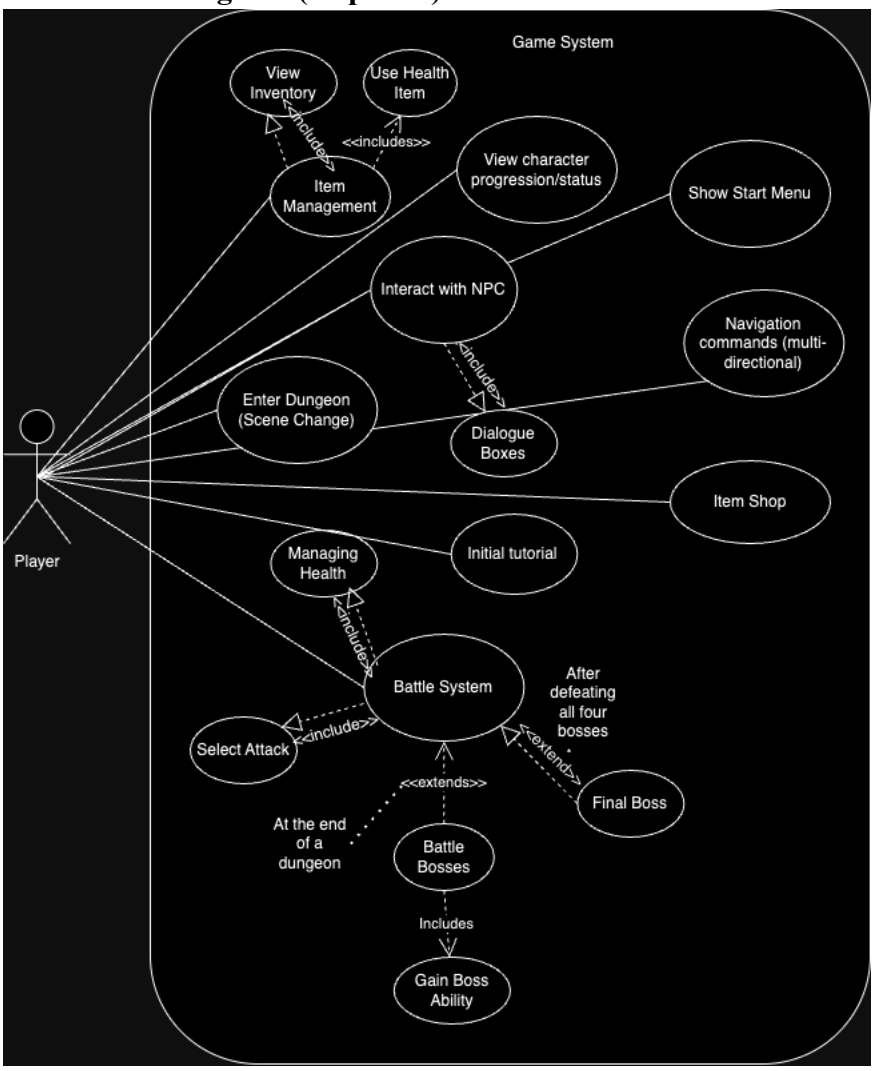
1. Allow the player to control a single main-character in a 2D top-down environment with keyboard input (high priority)
    a. This includes generating a script that allows for player keyboard input, as well as setting key-bindings in Unity
2. Allow the player to move between dungeons (separate scenes in Unity) in order to navigate the world. (high priority)
    a. This is achieved by the player walking over a door hitbox in the central area, triggering a script.
3. Implement central area NPC for the "tutorial" (high priority)
    a. Implement dialogue boxes
    b. NPC explains game mechanics and central focus to the player
4. Create an in-game combat system inspired by Pokemon (high priority)
    a. Turn-based 1v1
    b. Player has a list of actions to choose from
    c. Battle will take place in a designated "battle map" instead of the overworld.
5. Implement 5 main boss battles (high priority)
    a. four bosses for each dungeon area
    b. one central area final boss that is unlocked after completing all four dungeons
    c. Will follow the turn-based 1v1 combat as previously specified
6. Generate and distribute an executable file in Unity for the player to run the game (medium priority)
    a. Include a download link to the game in the Github
7. Implement staged small-enemy battles within the dungeons (high priority)
    a. Design enemy characters for the player to fight in dungeons before the final dungeon boss
8. Add skill books to find within dungeons (medium priority)
9. Add health items to find within dungeons (low priority)
10. Track the players health status for use in the battle system (low priority)
11. Allow the player to level up by managing experience points achieved in battles (medium priority)
12. Implement central "rest" area NPCs and item shop to allow the player to heal between dungeons and purchase healing items to use within the dungeons (medium priority)
13. A main home screen the player sees as the game starts (low priority)
14. Music in each primary area(low priority)
    a. Four dungeon pieces
    b. Combat theme
    c. Dungeon Boss Theme
    d. Final Boss Theme

3. Non-functional Requirements (10 points)

1. Shall have smooth and intuitive movement mechanics for the player, including proper boundary collision behaviors, no jitters on movement, and intended interaction with objects in the overworld.

2. Will include clear and high quality visual assets that are not warped or grainy.
3. Feature high-quality background music that dynamically changes based on the player's location (e.g., calm music in the central area and intense music during battles).
4. Shall maintain a consistent frame rate of at least 30 FPS during gameplay
5. Shall support the addition of more content such as puzzles, more dungeons, and quests in future updates without requiring major changes to the core game architecture.
6. Shall maintain minimal loading times transitioning between scenes (under 5 seconds)
7. The game's user interface shall be intuitive and easy to navigate, with clearly labeled menus and controls. Players should be able to access inventory and character status with minimal effort.
8. Shall be compatible with multiple systems, such as Windows and MacOS.
9. Shall be able to handle errors and unexpected user behaviors and actions.

4. Use Case Diagram (10 points)



Textual Descriptions:

Actors: Player

Preconditions:

- The user must have access to the game through either Unity or an executable file

- The user must have a keyboard
- The user must have a mouse

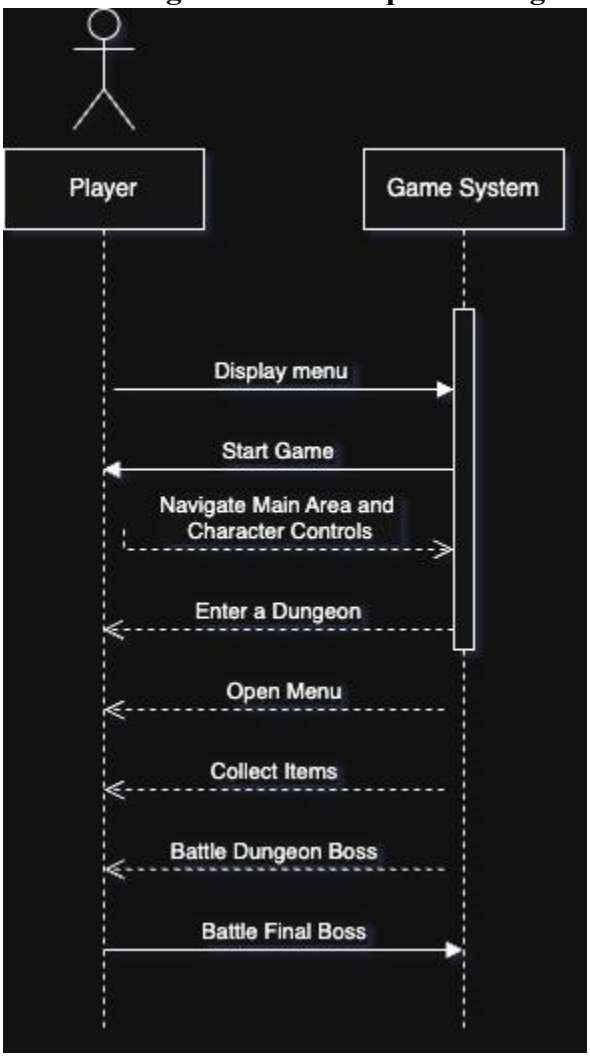
Postconditions:
- The user must be able to exit the game cleanly and smoothly

Flow of Events:
- The user enters the game system
- The user switches scenes for each dungeon
- The user accesses items through the item management system
- The user interacts with NPCs using dialogue boxes
- The user enters the battle system through enemy encounters
- The user can fight bosses
- The user can view character progress such as items and skills gained
- The user can access the item shop in the central area

Alternative Flows:
- If the user dies in battle, they return to the central area with one health.
- If the user does not beat all four dungeons, they cannot fight the final boss.
- If the user attempts to fight the dungeon boss before defeating enemies, they will be unable.

Special Requirements:
- All four dungeons must be won before fighting the final boss in the central area

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



6. Operating Environment (5 points)

Platforms: Windows, macOS
Game Engine: Unity v6000.0.23f1
Other Software: Git

7. Assumptions and Dependencies (5 points)

1. The player will have access to a keyboard for the movement of the character
2. All third-party assets are properly licensed for use
3. User utilizes intended operating system for the executable file, as well as the ability to download the executable from a link provided.