Software Requirements and Design Document

For

Group 8

Version 1.0

Authors:
Zak K.
Taj B.
Ash C.
Taylor R.
John M.

1. Overview (5 points)

- Our group is creating a video game that falls into the genre of a metroidvania. There will be a map the player can explore, find items, and interact with NPC's, including a shop where the player can buy upgrades or items. The goal for the player is to ascend something in the very end.

2. Functional Requirements (10 points)

- ***Coin Requirements***
  1. COIN: Users should be able to pick up coins by walking into them. The application should use collision detection from the coin and the player. (HIGH)
  2. COIN: Users should be able to use coins. The application should keep track of how many coins the player has and accurately change its value. (MEDIUM)

- ***Enemy Requirements***
  1. ENEMY Collision: The system should be able to recognize that the enemy character has entered the user's hitbox. (HIGH)
  2. ENEMY State Machine: The system should be able to accurately switch the enemy's state correctly and in a sensible manner (i.e., idle state, patrol state, action state, etc.) (HIGH)
  3. ENEMY Enhanced Mechanics: The system should allow the user to interact with enemy in a unique way (i.e., ride an enemy across a gap, turn an enemy into a weapon) (LOW)

- ***Game Over Screen Requirement***
  1. Game Over Message: The screen should display a "Game Over" message when the player dies. (HIGH)
  2. Retry Button: On the same screen as the "Game Over" message, a button labeled "Retry?" should appear and reset the scene when pressed. (HIGH)

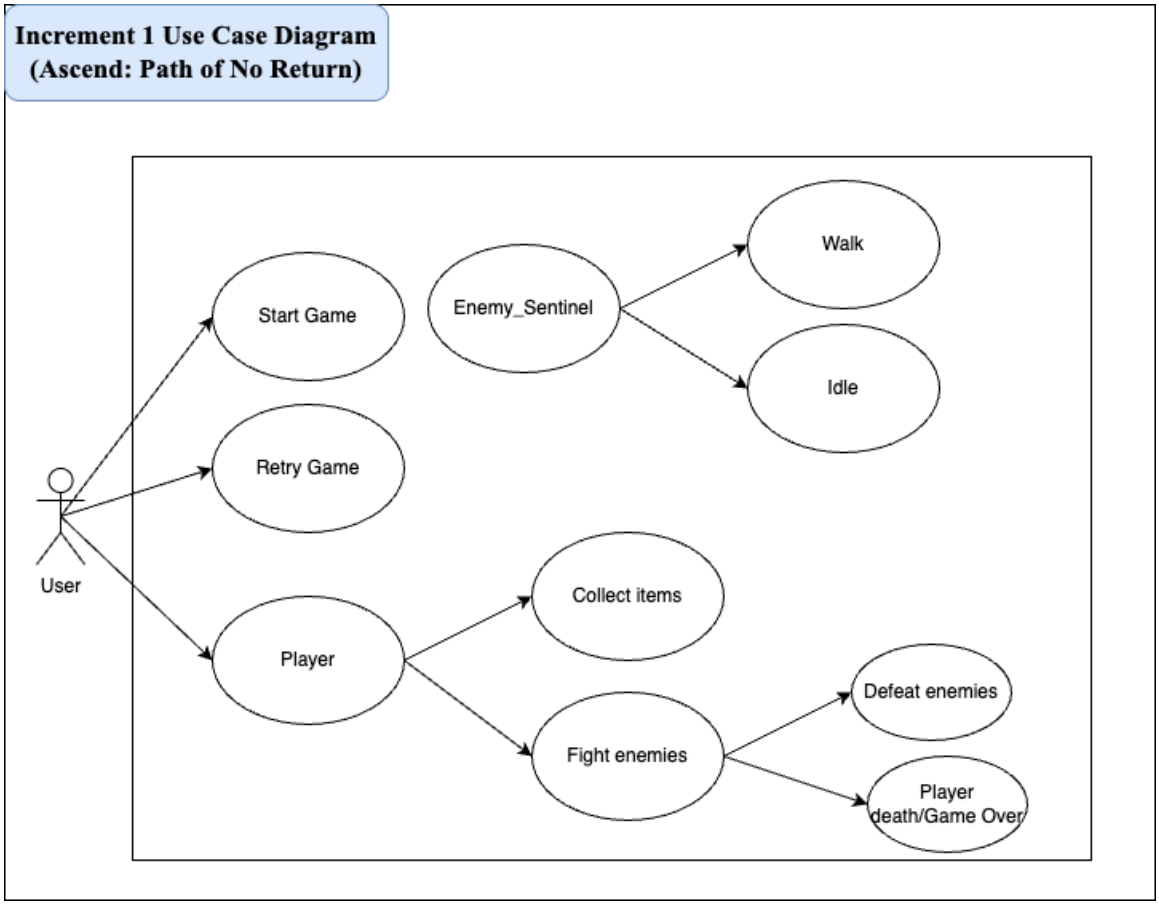
- ***Player Requirements***
  1. PLAYER Movement: User interacts with the player by using appropriate keys (L & R arrows plus spacebar for jump). (HIGH)
  2. PLAYER Collision: Player plays appropriate death animation once collided with enemy. (HIGH)

- ***Map Requirements***
  1. MAP Navigation: The player should be able to walk around the map, with map collision properly set to handle player movement and interactions with items. (HIGH)
  2. MAP Tileset: A map tileset has been set up, allowing for future additions such as obstacles and other interactive elements. (HIGH)

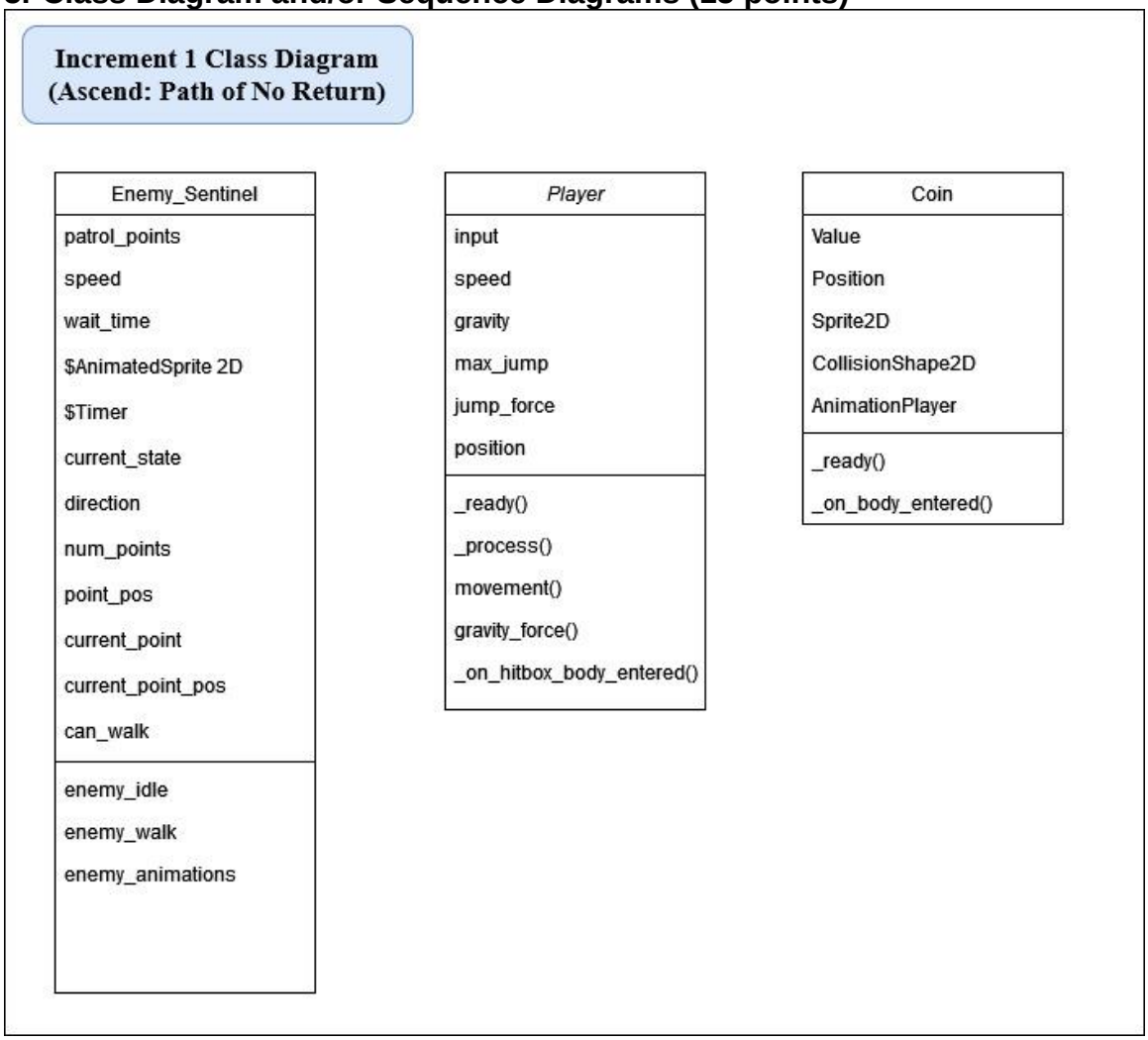
3. Non-functional Requirements (10 points)

1. The game should maintain a frame rate of at least 30 FPS
2. The game should load the entire map within 5 seconds
3. The game should be responsive to user input with little to no delay
4. The camera should follow the player smoothly with no sudden jumps
5. The game should save the location of the player at checkpoints
6. The game-over screen should load immediately upon death
7. The replay button should function without delay

4. Use Case Diagram (10 points)



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



6. Operating Environment (5 points)

- The software will operate only in a Windows 10/11 environment.

7. Assumptions and Dependencies (5 points)

- It is assumed that the player is using a keyboard to play the game. We are relying on Godot's engine and features.