Software Requirements and Design Document

For

Group 12

Version 1.0

Authors:
Nathan O
Amisaday S
Isabela T
Bria W
Matthew G

1. Overview (5 points)

The game we're making is for now still unnamed as we want it to be meaningful. Our group is making a 2D, top down, game centered around a girl named Luna trying to get out of a maze. The goal is to, with limited light and scant weapons randomly scattered through each level, navigate through the labyrinth-esc area, and proceed to the next level. This is done by both finding a key and a door. The player will need to avoid varied traps and enemies, all dependent on whichever area/realm the player has made it into and has also has the potential to find coins for a shop. The shop will likely be between a certain amount of levels.

2. Functional Requirements (10 points)

1.) **Levels:** The game will boast some where between 20-25 levels, and four or five different biomes with varied enemies, traps, and aesthetics. It would be comparable to Mario: after doing a few levels, you go to a new world. Priority for this is medium: having the mechanics for the game created is more important than getting all of the levels designed.

2.) **RNG Aspects:** Weapons, coins, the key respective to each door, some enemy placement, and possibly even player placement, will be randomized for every level. The levels themselves will not change and the size and complexity of each will effect what entities spawns where. This is medium priority: while we intend to implement this, if for some reason time doesn't allow for it, the item placement can be set instead of randomized.

3.) **Player movement:** Luna will be responsive to both keyboard and controller schemes. She will be able to move around, jump, attack, and quite possibly though we're not sure yet, sprint/dash. This is a high priority as the player must be able to navigate the maze to beat the game.

4.) **Light mechanic:** The player will have limited vision restricted to a small magical star that will provide them scant light in the maze. High priority: this was one of the founding ideas for the game. It makes the mazes more difficult to navigate as much as it adds a sense of forebode.

5.) **Key and door:** The player will need to locate both a door that can allow them to proceed to the next level as well as a key to unlock said door. This is a high priority as it triggers the next level. The doors may change aesthetically but will all function the same.

6.) **Enemies and Traps:** Enemies will traverse the maze, posing a threat along side the dormant traps. Some will follow a predefined path and can be easily avoided, some will be more sporadic in their patrols, some are stationary, and some will chase the player. Each new area, not every new level, will introduce new enemies. While enemies are a high priority their variety is debatable as a low priority. Second to this, traps are also a high priority and will also vary per-level, it is unlikely they will ever be randomized. They will always be in the same tactful place in the same level, both as a deadly hinderance and as a potential landmark.

7.) **Weapons:** Weapons are a temporary solution to enemies or maybe even barricades in the environment. They spawn randomly and will very seldom, if ever, be needed to complete a level. They're a low priority for this reason: they'd be useful for the player, but not necessary to progress the game.

8.) **Currency and Shop:** Coins for the moment are a medium priority. There is intended use to have them be used for currency for some form of exchange at a shop, one likely between levels, or quite possibly, randomly hidden in some levels. Integration of a shop is low priority for now; if it isn't added later, coins can still function as collectables that enhances the player's score.

9.) **HUD:** The HUD will likely be minimal, showing the player's health, the weapon they have equipped, the amount of coins they have, and whether or not they have the required key(s) to complete the level. An empty key slot in the HUD bar will also denote how many keys are needed to complete the level. This is high priority as it conveys to the player the length of their objective, how many keys they must acquire and how many they have, in case they've forgotten, as well as their health.

10.) **Feedback:** This is already achieved through the HUD, though this is extra means to communicate to the player what's happening. For example, if Luna takes damage, we may want to incorporate some screen shake and some sort of sound effect. We may play a stinger when a key is collected or a door is unlocked. This is ultimately a low priority though, as previously stated, the necessities are already communicated through the HUD.

11.) **Menu:** The in-game menu will allow players to pause, resume, exit, and adjust settings, such as volumes, control layouts, and other potential accessibility features. This is a high priority as it allows the game to be saved and the player to safely turn it off so that they may later pick up where they left off.

3. Non-functional Requirements (10 points)

1.) **Smooth gameplay and Optimization:** The game engine will allow for consistent FPS as well as fast load times across machines. The complexity of areas should bear little to no effect on framerate.

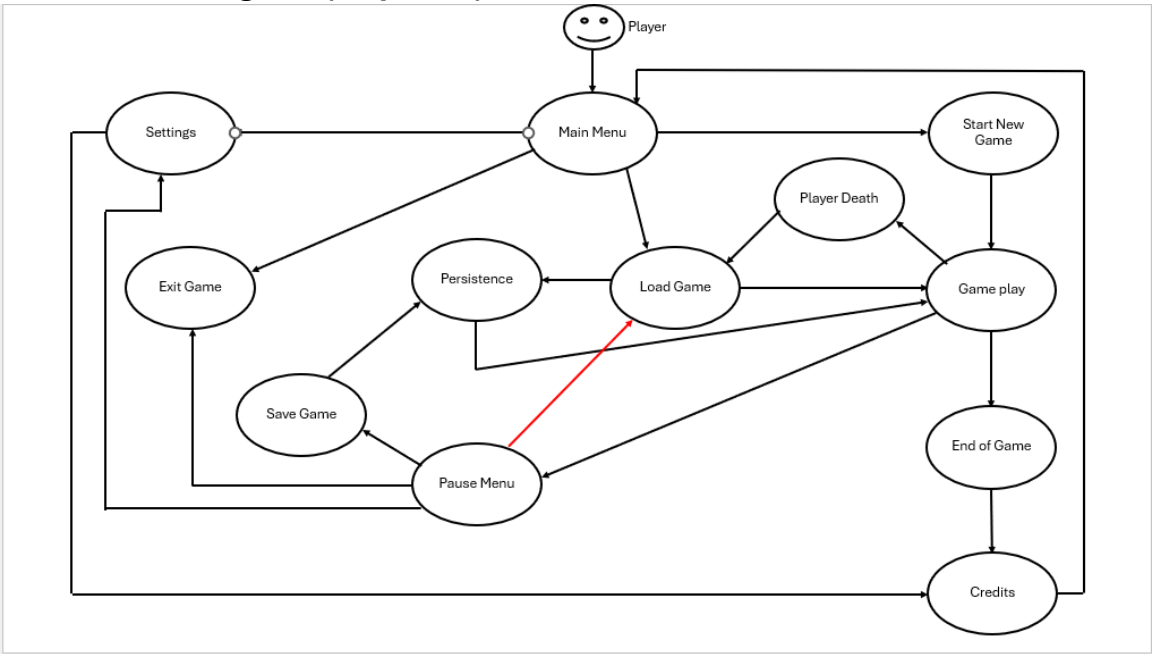
2.) **Intuitive Controls:** The control scheme across both the keyboard layout and controller will be easy for players to grasp. The controls can be viewed from the in game menu and there is potential for hints to be enabled when relevant in game.

3.) **Robust Testing:** Excluding the five people making it, the game maybe tested by others to ensure it runs perfectly and that any unseen bugs that otherwise wouldn't have been found might be squashed.

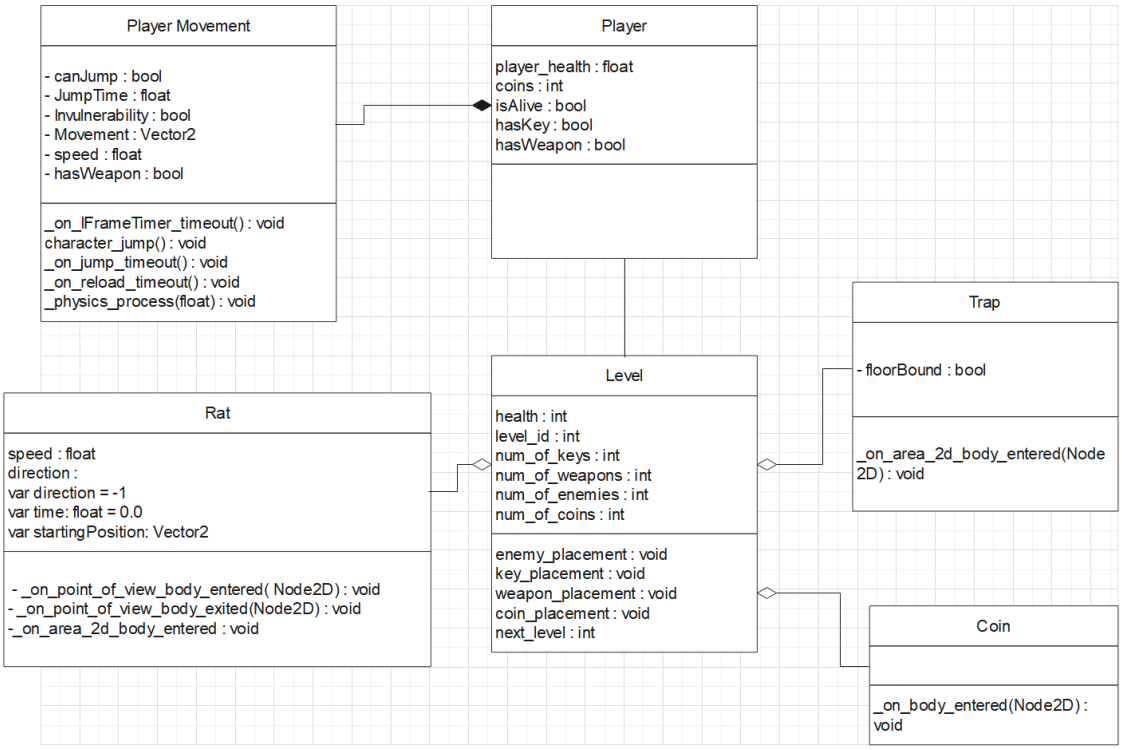
4.) **Modular Design:** The game engine, entities and scripts within it, have been and will continually built in a modular manner to ensure the code is easy to navigate, fix, and build upon.

5.) **Consistent Style:** To maintain immersion and mood, the art style will remain consistent throughout, employing art from the same source(s), or art that doesn't conflict with said source(s).

4. Use Case Diagram (10 points)

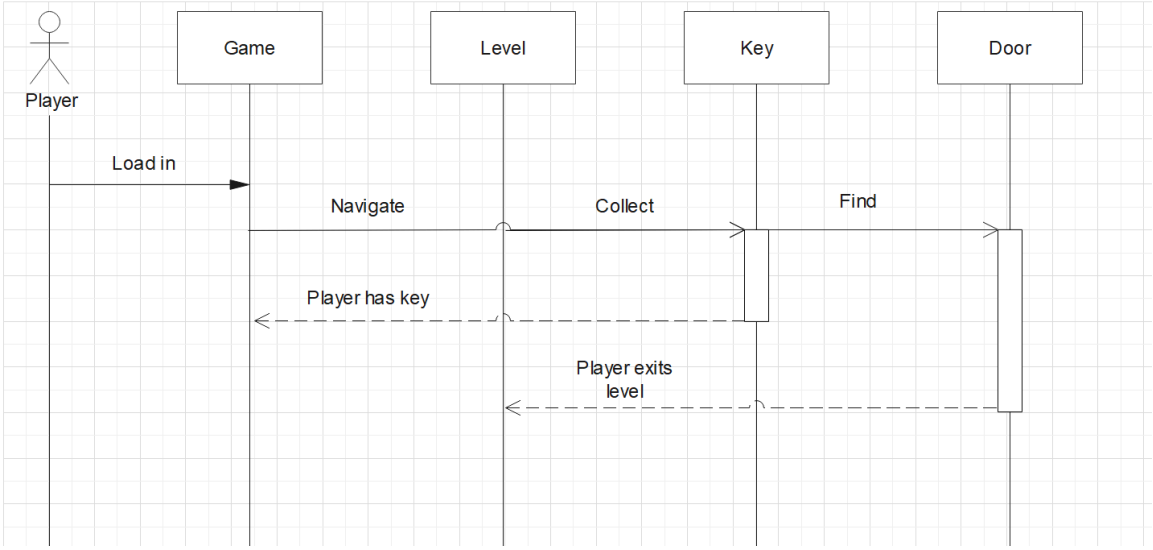


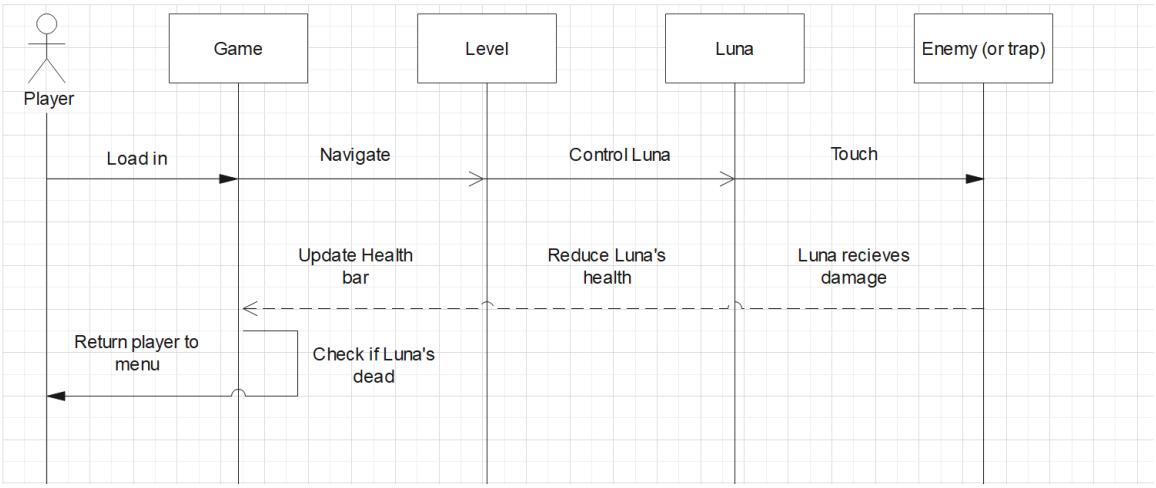
5. Class Diagram and/or Sequence Diagrams (15 points)

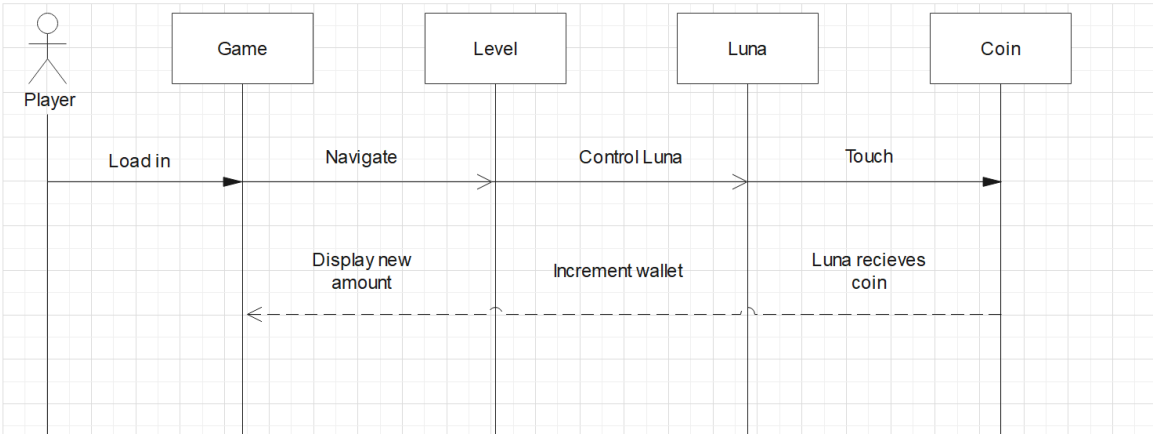


CD Notes: There currently isn't an actual level class as none have been designed yet, the attributes and variables are just assumed as to what maybe in one. Similarly, there is no official enemy class, only a rat enemy, though by next increment it's likely it'll be replaced with an enemy class
with much broader attributes and methods.

Getting to the next level:







SD Notes: These are subject to change or even be replaced as the project continues. The coin sequence diagram will likely be replaced with a shop functionality diagram when this is later discussed with the team in increment two.

6. Operating Environment (5 points)

The game will operate on both recent Windows and MACOS systems, and potentially more as the system the game is made with, Godot, is cross platform. It should run well on these systems and should utilize, without error, dependencies shared between them.

7. Assumptions and Dependencies (5 points)

The game has been built from scratch using the Godot engine and depends on no outside sources when in regard to coding, with the exception of a free-use template employed for the movement system, one that has been built off of. For a time, the game will use a sprite pack from itch.io (https://pixelfight.itch.io/retro-character-pack), though this only for temporary use. As of increment one and the foreseeable future, the bulk of sprites will be made by one of the team member's friends, Gabbie. She will be drawing much of the sprites for the character Luna, monsters, traps, and potentially the environment. Similarly, the soundscape for both music and general sound effects will be made by a sound engineer, also a friend of a team member.