I. Introduction

This document outlines the user requirements for a 2D platformer game with roguelike elements, centered around space exploration. Players explore procedurally-generated levels, defeat monsters, and inherit abilities to expand their combat options. The core gameplay focuses on combat and progression, allowing players to enhance their skills through powers gained from fallen enemies. The game will also include optional LAN multiplayer, enabling cooperative dungeon exploration, where players collaborate to defeat monsters and strategize using inherited abilities.

In addition, players will collect resources to upgrade and customize their hub base, which evolves as they progress. The hub serves as a functional home area, offering new capabilities over time. In multiplayer, the hub can be upgraded collaboratively, adding a new layer of strategic depth.

The primary stakeholder, the teaching assistant (TA), will oversee the project, ensuring it meets the requirements and delivers a high-quality final product.

II. User Requirements

Features:

1. **Good User Interface (UI) and Experience:** The game should provide a simple, intuitive interface that is easy to navigate and understand. Players should be able to easily control their characters, access menus, and track their progress. Information such as player health, available weapons, status ailments, and abilities should always be displayed. The interface will prioritize ease of use and clarity to ensure that both new and experienced players have a smooth gaming experience.
2. **Session Preservation and Save State:** Players must be able to preserve their game session when voluntarily exiting. This means that leaving the game will not reset progress; instead, the game should save the current state, including the player's position, upgrades, and resources. Upon returning, players should be able to resume from where they left off without having to restart their progress or the level. If a player is to forfeit or die along their run, however, this property will not be maintained and they will instead lose progress.
3. **Player Mechanics:** The game shall feature a clear and intuitive system for player mechanics, focusing on combat, health management, and ability customization. Players will have a health bar (HP) that depletes when they take damage and regenerates through in-game pickups or abilities. Combat will involve both melee and ranged weapons, with players able to switch back and forth between the two fighting styles at the press of a button. A key mechanic is the ability to inherit powers from defeated enemies, allowing players to customize and expand their skill set as they progress. The game should provide clear tutorials and prompts to ensure players understand how to navigate, fight, and use inherited abilities effectively.
4. **Visual Effects and Level Differentiation:** Each “world” in the game should feature distinct visual effects (VFX) to create a visually immersive experience. These effects will

help distinguish between different environments, enemy types, and special abilities. Unique visual effects should also accompany player actions like combat or ability usage, enhancing the overall gameplay experience.

Integrations:
At present, there are no specific integration needs with third-party systems or APIs. However, the optional LAN multiplayer mode may require integration with local network protocols to ensure smooth, secure, and stable multiplayer sessions.

User Interface:
The UI will prioritize simplicity and accessibility, displaying key elements like health, inventory, and abilities while stationary, and fading them during movement for clearer gameplay. Menus will be easy to navigate with minimal input, maintaining a consistent and user-friendly experience in both single-player and multiplayer modes. Visual feedback, such as flashing icons for low health or cooldown timers, will enhance clarity during gameplay.

Constraints:

1. **Timeline**: The project must be completed within the allocated time of the semester.
2. **Budget**: There is a limited budget for external resources, including any third-party software or services.
3. **Technology Stack**: The game will be developed using Godot, thus the team will need to work within the limitations of this engine.
4. **Resources**: The team consists of students with varying levels of experience, so development tasks must be divided according to skill sets to meet deadlines effectively.

III. Use Case

1. **Starting a New Game:** A player starts a new game, selecting a character and beginning their journey through the procedurally generated levels. The player learns how to navigate the game world and engage in combat.

Actors: Player
Steps:

- The player launches the game and selects "New Game" from the main menu.
- The player chooses a character and proceeds to the first level.
- In-game tutorials introduce basic movement controls (jumping, running, etc.).
- The player encounters an enemy, and the tutorial explains how to attack and use inherited abilities.
- The player defeats the enemy and proceeds to progress through the first level.

Expected Outcome: The player successfully begins their journey, understanding the core mechanics of movement and combat. The player can control their character, defeat enemies, and use their inherited abilities to progress through levels.

2. **Resuming a Preserved Session**: A player is mid-session, exploring a dungeon, but needs to leave the game. They exit the game and return later to resume from where they left off without losing progress.

Actors: Player

Steps:

- The player presses the pause button during a dungeon level.
- The player selects the "Save and Exit" option from the pause menu.
- The game stores the player's current state, including their position, HP, abilities, and collected resources.
- The player exits the game.
- Later, the player relaunches the game and selects "Continue Run" from the main menu.
- The game reloads the saved state, returning the player to the exact point where they left off.

Expected Outcome: The player can leave the game at any point and resume without losing any progress. Upon re-entry, the player finds themselves in the same state they were in when they exited, with all gameplay elements intact.

3. **Upgrading the Hub Base:** The player collects resources from dungeon runs and uses them to upgrade their hub base, unlocking new buildings, features, and cosmetic enhancements that affect gameplay.

Actors: Player
Steps:

- The player completes a dungeon run and collects resources (e.g., currency, materials) along the way.
- Upon returning to the hub, the player accesses the upgrade menu from the workbench area.
- The player selects a specific building or feature to upgrade (e.g., forge for new weapons, a resource farm, or cosmetic items for the base).
- The upgrade begins, and the player sees a progress bar or countdown timer.
- Once the upgrade is complete, the new building or feature becomes available for use.

Expected Outcome: The player can successfully upgrade their hub base, unlocking new functionalities (like crafting or resource generation) or enhancing the aesthetic appeal of their base. This progression allows for further customization and development of the hub area.

IV. Priorities & Milestones

The primary focus will be on the development of the player's platforming and combat physics, enemy design, and the ability inheritance system, alongside hardcoding initial levels to establish modular level generation and test the aforementioned features. Once these foundations are set, the priority will shift to implementing multiplayer functionality, a resource collection system, and an upgradable hub base. Afterward, additional worlds, levels, enemies, and power-ups will be developed to enhance the overall gameplay experience.

Milestones:

Month 1:

- Complete the development of core player mechanics (movement, combat).

- Implement initial enemy types with the ability inheritance system.
- Hardcode a couple of levels to prototype modular level generation, with basic VFX and world aesthetic shuffling.

Month 2.5:

- Implement and test LAN multiplayer functionality.
- Set up a basic resource system for dungeon loot and hub base upgrades.
- Begin developing a basic, upgradable hub area for players to return to between runs.

Remaining Timeline:

- Expand modular level generation with more worlds and aesthetic variations.
- Refine layout of hub and upgrade mechanics
- Add additional enemy types, abilities, and power-ups.
- Refine and balance gameplay, ensuring a cohesive player experience throughout.