Project Name: Simple Roguelike Dungeon Crawler

Project Manager & Developer: Michael Vera

Executive Summary

A simple roguelike dungeon crawler game allows the player to fight and explore a small randomly generated dungeon through turn-based combat and resets after the dungeon is completed or upon death. This game will showcase expertise in Python, as it will be the only language used, as well as Object-Oriented Programming concepts. This also gives a rough and generalized experience with working within a project under industry standard workflows like scrum.

Project Objectives

- Completely develop a roguelike dungeon crawler game by the due date mentioned in the timeline.
- Showcase skills and knowledge in Python and apply concepts in Object-Oriented Programming.
- Use tools and workflows that are standard in production code and real-world projects.

Project Scope

- In Scope
- Developed in Python using Pygame and other necessary libraries
- Start menu
- Class choices (Fighter, Barbarian, Rogue)
- Character stats for player, enemies, and bosses (Inheritance)
- Turn-based combat system for each encounter
- Weapons & Items for the player to use in their inventory (Composition)
- Different types of weapons, items to be found and used (Polymorphism)
- Dungeon is randomly generated
- Dungeon has rooms that are either chest rooms or enemy encounters
- Victory or death screen
- Out of Scope
- Customizable player stats
- Equipment and Armor
- Min-Max stats playstyle with buff/debuff options on weapons or items
- Multiplayer

- Roguelite/Game save elements

Functional Requirements

ID	Requirement Description	Priority
FR-001	Player can run game and reach a start menu	High
FR-002	Dungeon is successfully generated with multiple rooms of either enemy encounters or chest encounters as well as one boss encounter.	High
FR-003	Players can interact with the main gameplay loop which is a turn-based combat system utilizing player and enemy stats as well as chance for hits and misses.	High
FR-004	Player can choose which room to go into and must fight if entered an enemy encounter room before leaving.	High
FR-005	Player can choose a class and successfully play class in their playthrough	High
FR-006	Player can choose which weapon to wield and which items to use during an enemy encounter	High
FR-007	Player can manipulate their inventory and choose which items to have or discard when opening a chest or not fighting	High
FR-008	Player can see game over screen upon death or victory screen upon defeating the boss	High

---

Stack and Tools

- Development

- Python (Main structure & game logic)

- Pygame (GUI)

- Management

- Git (Version Control and File Saving)

Scrum Implementation

- Sprint Duration: 1 day
- Sprint Planning: Morning
- Sprint Review: Evening
- Definition of Done
    - Planned sprint goals have been fully implemented.
    - Code achieving sprint goals has been successfully tested.
    - Previous backlogged sprint goals have been worked on/completed.

Timeline

Sprint Number	Duration	Goals

Sprint 1	8/4/25	- Overall project planning - Created file structure - Finalized scrum workflow - FR-001
Sprint 2	8/5/25	- FR-002 - FR-004
Sprint 3	8/6/25	- FR-005 - FR-003
Sprint 4	8/7/25	- FR-006 - FR-007
Sprint 5	8/8/25	- FR-008 - Backlog

Risks & Mitigation

Risk	Likelihood	Impact	Mitigation
Development teams' time on project	Medium	High	Avoid unnecessary time away from development. If absolutely necessary, then get it done efficiently.
Planning mistake/ unknown variables in development	High	Medium	Adapt and change as necessary. Avoid changing scrum and timeline unless necessary to keep up with timeline.
Development rut	Medium	High	Narrow down the problem on why progress is stalling and either take a new approach or research similar problems to find a solution.
Sprint backlog becomes overloaded as timeline proceeds	Medium	Very High	LOCK THE FUCK IN BROTHER.

Approvals

Name	Role	Initials
Michael Vera	Project Owner Project Manager Project Developer	M.V.