Design and Specification Outline

Project Tile: Procedural Map Generator Extension for Godot 2D
Group Members: Dorian Sanchez (C++ Extension Developer), Gareth Carew (Godot Scripting & Integration), Kristopher Thomas (Game Designer & QA Tester)

1. Executive Summary

- High Level Overview
    - Purpose of the extension
        - Create dynamic, procedurally generated, tile-based maps for 2D games in Godot
    - Game prototype concept
        - A top-down dungeon crawler with procedurally generated levels
- Key Technical Challenges
    - Ensure generated maps are playable
    - Optimize performance for large maps
    - Expose a clear and usable API for C# scripts

2. Extension Specification

- Extension Functionality
    - Generate 2D maps procedurally using configurable parameters:
        - Randomization seed (for reproducibility)
    - Map visual representation in godot
    - Possible Pathfinding for AI navigation

- Technical Architecture
    - Singleton Pattern: One global map generator instance
    - Factory Pattern: Dynamically create rooms

- API Documentation
    - ProceduralTileMapLayer
        - Properties
            - bool Collision_enabled
            - TileMapLayer tileMapLayer
            - Node2D Node_to_follow
            - Int Generation_radius
        - Functions Exposed to C#
            - void setNodeToFollow(Node2D nodeToFollow)
            - void setGenerationRadius(int radius)
            - void clear()
            - void remove_tile(Vector2i position)
            - void set_tile(Vector2i position, Vector2i atlasPosition)
- Integration Approach

    - Build a custom Godot 2D Extension
    - Use Godot Extension API

- Comparison with Existing Solutions
    - Godot supports tilemaps with algorithms to implement a procedural generation solution
    - Highlights
        - Easy API
        - Dynamic object placement
        - Optimized performance
        - Procedural generation

3. Game Design

- Game Concept and Genre
    - Type: Top down dungeon crawler
    - Player goal
        - Explore dungeons, defeat enemies, collect items

- Core Gameplay Mechanics
    - Basic WASD movement
    - Combat
        - Close and far ranged attacks
    - Objectives
        - Survive, collect items, kill enemies
    - Procedurally placed enemies, obstacles, items

- How the custom extension will improve game design
    - Map generator creates playable levels dynamically
    - AI uses pathinding that's generated by the extension
    - Rooms and paths will be logical and flow into each other

- Target audience and platform
    - Target audience: Roguelike players, casual players
    - Platform: PC

- Visual Style and Aesthetic
    - Tile based graphics using pixel art
    - Clear visual feedback for map generation

Technical Implementation

- Development timeline

1	Extension design, class architecture, API planning
2	C++ implementation of map generation

3	Game integration and gameplay functions
4	Polish and bug fixes

- Technology Stack and Dependencies
  - Godot 4.4 2D engine
  - C++ for extension development
  - C# for game implementation
  - Github for version control

- Risk assessment and mitigation

Risk	Solution
Performance issues	Optimize generation
Integration bugs	Regular testing

Team Collaboration Plan

- Role distribution
  - Each member tracks and oversees certain parts of the development process
  - C++ Extension Developer
  - Godot Integration and Scripting
  - Game design and testing

- Communication Plans
  - Regular meetings through discord or in person
  - Regular updates with each other and what we are working on
  - Shared github project

- Github workflow
  - Main branch: The stable working version
  - Feature branches
    - each major feature uses its own branch
  - Pull requests before merging
  - Issue tracking for bugs, features, and testing

- Conflict Resolution
  - Ask for assistance with tricky sections
  - Team voting for major decisions

References and Resources

- Godot engine docs
  - https://docs.godotengine.org/en/4.4/index.html
- C++ tutorials

- Procedural generation research papers
- Online tutorials