**a. Game Concept**

* Define the purpose of the game: What is the goal? (e.g., exploring a dungeon, finding treasure, escaping a maze)

Answers: Focusing on exploring a dungeon and escaping the maze. Both features will have a mini map to show the player current location

* Decide on the map structure: Is it a grid, graph, or procedurally generated map?

It will use a grid for easy implementation.

**b. Core Components**

1. **Map Design**
   * Represent the map as a 2D grid, a dictionary, or a graph.
   * Define rooms, paths, or locations.
   * Add details to each location (e.g., description, items, enemies).

Answers:

The map representation is a 2D grid. It is similar to create a n x n table but not every block is used

It would be better to have a collection of rooms that will have: id, location on maps, description, directions that connect to other rooms.

First, let’s create an initial map room based on my documents. Then, we could create a room generation that will automatically create rooms and environments related to it.

Each map generation will have a hidden item called “Full Map Unlock” which will shows the entire map for the players, which is useful for finding the exit.

1. **Player**
   * Track the player's current position.
   * Include an inventory system, health, or other stats if needed.

Answers:

At the moment, we just need to track the player’s current position. We also include an inventory system which will store some hints and key components (ex: Full Map Unlock items)

1. **Actions**
   * Define player actions like move, look, take item, use item, etc.
   * Define commands and how the game interprets them (e.g., go north, inspect, pick up).

Answer:

Player actions will only have to move and take items at the moment. But we should leave some room for future improvements.

1. **Game Loop**
   * Continuously accept input from the player.
   * Process actions and update the game state.
   * Check for winning or losing conditions.

Answers:

Follow your game loop ideas

1. **Events and Obstacles**
   * Include random or fixed events like encountering enemies, finding treasures, or solving puzzles.
   * Define interactions between objects in the environment and the player.

Answers: Based on the three game modes, but initially we mostly solve the puzzles. Such as some simple math problems and calculation, find the abnormal symbols among a list of strings, etc. You could help me with these simple puzzles.

There is no need for interactions between environments at the moment.

1. **Victory and End States**
   * Define conditions for winning or losing the game.
   * Add optional story progression.

Answers: For exploring the dungeons and escaping the maze, as long as they reached the exit room. It is a win condition.

No need for optional story progression but we could add that features in the future.