# Main Requirements

* **~~World Memory~~**~~: Implement persistent storage of state for each location. The AI must recognize when players revisit a location and present an updated description reflecting prior actions.~~
* **~~Map Generation and Navigation~~**~~: The AI should create and maintain a coherent map of connected locations, ensuring logical placement and connections.~~
* **~~Dynamic Descriptions and Interactions~~**~~: Generate unique descriptions and manage NPC or item states dynamically, adapting based on player actions.~~
* **~~Randomized Location Descriptions and Encounters~~**~~: When players enter a new location, the AI generates a fresh, random description consistent with nearby areas and may spawn new NPCs or items. This allows each exploration to feel novel and immersive while maintaining a coherent world.~~
* **~~Simplified D&D Mechanics~~**~~: Use a streamlined set of D&D-inspired mechanics (detailed below) to guide the AI’s decisions and narration for combat, exploration, and interactions.~~

# Game Mechanics

* **~~NPC Interaction and Combat~~**~~NPCs have two attributes:~~ **~~Health Points (HP)~~** ~~and~~ **~~Attack Power~~**~~.~~
  + **~~Combat~~**~~: When players engage in combat, each side rolls a virtual 1d6 (a six-sided die) and adds it to their Attack Power to determine damage for that turn.~~
  + **~~Victory Conditions~~**~~: The player wins if the NPC’s HP reaches zero, and loses if their own HP does.~~
* **~~Skill Checks and Random Outcomes~~**~~When players attempt a risky action (e.g., unlocking a door), the AI rolls a 1d10 (a ten-sided die) for a~~ **~~skill check~~**~~.~~
  + ~~For simple tasks, a roll of~~ **~~3 or higher~~** ~~succeeds.~~
  + ~~For challenging tasks, a roll of~~ **~~6 or higher~~** ~~is needed.~~
  + ~~Descriptions should vary based on success or failure, giving the interaction a creative touch.~~
* **~~Inventory and Simple Item Use~~**~~Players can pick up and use basic items like keys or healing potions.~~
  + **~~Items~~**~~: Each item has a single effect—e.g., a potion heals 10 HP, a key unlocks a specific door.~~
  + ~~The AI remembers which items have been used or taken to manage inventory without complex tracking.~~
* **~~Experience Points and Leveling (Optional)~~**~~Players can gain~~ **~~Experience Points (XP)~~** ~~for completing significant actions or defeating NPCs.~~
  + ~~When players reach 50 XP, they "level up," gaining a small HP increase.~~
  + ~~This provides a sense of progression with minimal complexity.~~

# State Persistence and Map Representation

* **~~State Persistence~~**~~The AI must track and remember states for each location and NPC to maintain continuity. This includes:~~
  + **~~Location State~~**~~: Track if a door is unlocked, a chest is opened, or an item has been taken.~~
  + **~~NPC Status~~**~~: Remember if NPCs have been defeated, moved, or interacted with.~~
  + **~~Inventory~~**~~: Track items players possess and mark items as “used” once they’re applied (e.g., a used key).~~
  + **~~Player Stats~~**~~: Keep player attributes like HP, XP, and inventory status persistent across moves.~~
* **~~Map Representation~~**~~The map is a logical grid or node-based structure, where each “node” represents a unique location.~~
  + **~~Location Connections~~**~~: Each node includes connections to adjacent locations (e.g., north, south, east, west).~~
  + **~~Unique Descriptions and Random Encounters~~**~~: Each time players enter a new node, the AI generates a fresh description that is contextually consistent with nearby areas. The AI may also spawn random NPCs or items, creating an immersive experience while ensuring coherence.~~
  + **~~NPC and Item Placement~~**~~: Each node may contain NPCs or items, which the AI can add or remove based on player actions.~~

# Deliverables

* **~~Demonstration Video~~**~~Students will record a brief video demonstrating the AI Dungeon Master in action. This video should showcase:~~
  + ~~Consistent world-building, including persistent changes when revisiting locations.~~
  + ~~Interactive storytelling with dynamically generated descriptions.~~
  + ~~Simplified D&D mechanics in action, such as skill checks and combat.~~
* **~~Code Repository~~**~~Upload a GitHub or GitLab repository containing:~~
  + ~~Well-documented code, with clear sections explaining how the AI handles state persistence, dynamic descriptions, and game mechanics.~~
  + ~~Instructions on running the code and reproducing the project demonstration.~~
  + ~~An optional README section describing any additional features implemented.~~
* **~~Final Report~~**~~Submit a final report summarizing:~~
  + ~~The AI’s architecture, including key design decisions related to state management and dynamic content generation.~~
  + ~~Challenges encountered and solutions implemented.~~
  + ~~Potential improvements and future extensions.~~

# Grad Specific Outcomes

* **~~Text-to-Speech (TTS)~~**~~: Add a TTS system to narrate the AI-generated descriptions and interactions, enhancing the immersive experience.~~
* **~~Image Generation~~**~~: Integrate a simple image generation model to create visual representations of locations or key events, adding a visual storytelling element to the text-based game.~~

# Grading Percentages

| Deliverable | Percentage |
| --- | --- |
| Consistency of World State | 20 |
| Creativity and Engagement | 20 |
| Technical Implementation | 30 |
| Gameplay Adherence to Simplified Mechanics | 20 |
| Clarity and Completeness of the Final Report | 10 |