Project Title: **AI Dungeon Master: Consistent World-Building and Adaptive Storytelling**

Project Description

In this project, students will create an AI agent that serves as a Dungeon Master (DM) for a text-based version of Dungeons & Dragons (D&D). The AI will manage and evolve a virtual world, generating a map of locations, descriptions, and maintaining state across player sessions. As players interact with this world, the AI will remember their actions, ensuring that each location reflects any past changes. For example, if players unlock a door or defeat an NPC, those changes will persist when revisiting the area.

Students will design the AI agent using OpenAI's language models to dynamically generate descriptions and interactions based on D&D mechanics, world-building best practices, and simplified game logic. The AI should adaptively respond to players, narrate outcomes, and preserve continuity across the game world.

Project 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.

Core Mechanics

To simplify the AI's management of game logic, students will implement these basic mechanics:

1. **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.
2. **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.
3. **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.
4. **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

1. **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.
2. **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.

Learning Outcomes

1. **Understand State Persistence and Retrieval**: Students will learn to develop AI agents that store and recall state, essential for consistency in interactive applications.
2. **World-Building and Storytelling**: Students will apply narrative design principles to create immersive, dynamic worlds and ensure coherent story flow.
3. **Game Logic Implementation**: Students will gain experience in implementing rules-based decision making to enhance player interactions.
4. **AI as a Creative Tool**: Students will explore AI’s potential in generating creative content, balancing randomness and coherence for engaging interactions.

Deliverables

1. **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.
2. **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.
3. **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.

Additional Outcomes for Graduate Students

Graduate students are required to implement additional features to expand the AI's storytelling capabilities:

- **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 Criteria

1. **Consistency of World State** (20%)
   The AI must maintain coherent states across sessions, allowing players to revisit unchanged or modified locations.
2. **Creativity and Engagement** (20%)
   Descriptions should be imaginative, with narrative quality that enhances player immersion.
3. **Technical Implementation** (30%)
   The code should be efficient, well-documented, and demonstrate effective memory management for a stable game experience.
4. **Gameplay Adherence to Simplified Mechanics** (20%)
   The AI should apply the simplified mechanics (basic combat, skill checks, inventory) authentically for a streamlined, enjoyable gameplay experience.
5. **Clarity and Completeness of Final Report** (10%)
   The report should thoroughly discuss the AI's architecture, state management strategy, and limitations.

This project provides hands-on experience with AI-driven storytelling, state persistence, and interactive design, and challenges students to leverage AI in creating consistent, immersive worlds.