CSE 434/534: Dungeon Master AI Agent

Allie Marcum, Steven Harvey, Sam Lothrop, Kimberly Hamilton

December 4th, 2024

AI Architecture

The AI Agent Dungeon Master is built upon a sophisticated architecture that leverages various advanced tools to create an immersive and dynamic Dungeons & Dragons experience in a text-based model. At its core, the system utilizes ChatGPT gpt-4o-mini, which serves as the primary language model responsible for generating rich descriptions, engaging dialogues, and managing the underlying game logic.

Crucially, the system employs OpenAI's function calling capabilities, allowing the AI to seamlessly interact with custom-defined functions. This enables the AI to perform specific tasks such as updating the game state, rolling dice, or retrieving information from the game's database. For instance, when a player attempts to pick a lock, the AI can call a function to perform a skill check, determining the outcome based on the game's rules.

This foundational model is complemented by the OpenAI API, which enables text-to-speech (TTS) functionality. This integration enhances player immersion by providing voiced narrations of character dialogues and environmental descriptions, allowing players to experience the narrative in a more engaging manner.

Additionally, Stability.AI is incorporated into the architecture to facilitate image generation. This tool creates visual representations of locations and key events within the game, enriching the storytelling experience by providing players with visual cues that complement the textual narratives.

State Management Strategy

To create a coherent gaming experience, the project employs JSON files for state management. This approach allows for lightweight storage solutions that can efficiently handle complex game states. The state management system tracks various elements crucial to gameplay, including location states—such as whether doors are unlocked or chests have been opened—as well as NPC statuses, which encompass health points, positions, and interaction histories.

Furthermore, player inventory and attributes are maintained within this system, allowing for seamless tracking of items collected, skills acquired, and character progression over time. The map structure is also stored in JSON format, detailing the layout and connections between different areas of the game world.

Dynamic Content Generation

Text descriptions are generated by ChatGPT relying on current game state and player actions, ensuring that narratives are contextually appropriate and responsive to player choices. This adaptability allows for unique storytelling experiences tailored to individual gameplay sessions.

The integration of OpenAI's TTS API transforms these text descriptions into spoken narration, adding an auditory layer that further immerses players in the game world. The auditory experience is complemented by visual elements produced by Stability.AI's image generation model, which creates images that represent scenes, characters, and items within the game.

Challenges Encountered

The team faced several significant challenges that tested our ability to integrate components effectively. One of the primary hurdles was the use of Google Colab as our development environment. While Colab offers powerful features for collaborative coding and access to resources, it also introduced complexities that hindered our workflow. From this we would have liked to start work on another platform, such as VS code's live share extension.

Maintaining collaborative coding efforts without inadvertently overwriting each other's work was another major challenge. As multiple team members contributed code simultaneously, implementing effective version control became essential. We utilized verbal communication for managing contributions.

Integrating individually created code components into a cohesive system also presented difficulties. Each team member developed their modules independently, leading to compatibility issues when merging these components. To address this, we emphasized regular communication and established integration sessions to test and refine our combined codebase.

We encountered technical issues related to kernel stability during development in Colab. These disruptions occasionally halted progress and required devoted time for troubleshooting to identify and resolve underlying problems. Such issues often stemmed from conflicts between different libraries within the Colab environment.

Improvements

Some implementations that currently exist in our AI agent but could use improvements are as follows.

Enhanced dynamics for content generation - The AI currently creates a good base of dynamic descriptions based on user input but it would be an improvement if this included developing more open-ended narrative structures.

More advanced TTS Options - The current TTS has a defined voice that we chose out of a few that were offered. Ideally, to improve this if there was an option to give more of a humanistic feel to the voice as far as age or accent would increase the users gameplay.

GUI Map Option - The map currently is a visual representation that changes based on text input by the user and the ai can call the move function or create diverse descriptions for the rooms. As an improvement in user-friendliness, it would be nice to change this to an optional clickable GUI map instead of a text input required.

Received image files from PNG instead of JSON - Currently we are receiving the image information in JSON format, but have the option to receive the PNG strait from Stable Diffusion. This may decrease the playtime of our system, but will require a rework of storage and interaction with long-term retrieval of the pictures.

Fog of War - In the map of the dungeon we currently show the entirety of the available rooms for exploration and how all of the rooms are connected. This makes it to where the player can make a bee-line for the exit. We would like to implement a fog of war where it only showed your current room, as well as the rooms that you have been in. This will add a sense of danger to exploration as you do not know how close you are to the exit.

Streaming - Load speeds are a bit slow currently in the dungeon location. Increasing speeds would be required for increased user-friendliness. This will require a full refactor of our code base.

Future Extensions

Looking ahead, there are numerous opportunities for extending the capabilities of the AI Dungeon Master.

Expanded Dungeons - Multi-level dungeons provide a larger play space for the players adding to the complexity and allowing us to provide more intricate quests and interactions with the world.

Party Creation - TTRPGs are best when played with a group. Currently the system is set up as a solo experience with the AI DM. Utilizing a party mechanic with the ability to handle more advanced combat, world interactions with party conflicts of interest, and individual advancement would be ideal.

Background Music/Sound Effects - While not needed for gameplay, adding a loop of background music as well as sound effects, possibly accompanying the image generation, would lend an ambiance to the game, and heighten the immersiveness of the interactions.

Experience/Leveling System - RPGs are, at their core, about the advancement of the player's character and the interaction between themselves and the world around them. Advancing your abilities, skills, and options are tantamount to that goal. Giving experience points(XP) as a reward for quest completion or destroying various monsters allows us to utilize a leveling system and allows us to temper the speed at which that advancement occurs.

Save/Load Sessions - This improvement would allow players to resume their adventures seamlessly. The system would automatically transport players to a designated save location, likely the tavern, upon loading a saved game. At the start of each gameplay session, the AI would prompt users to load their previous game state from a JSON file, enabling a smooth continuation of their adventure.

Conclusion

In conclusion, the AI Dungeon Master project represents an innovative approach to AI-driven storytelling within tabletop role-playing games. By combining state-of-the-art language models with effective state management strategies and dynamic content generation techniques, it offers a unique gaming experience that evolves based on player interactions. While challenges remain in refining these systems and addressing limitations, ongoing improvements and future extensions promise to enhance user engagement and gameplay depth significantly.