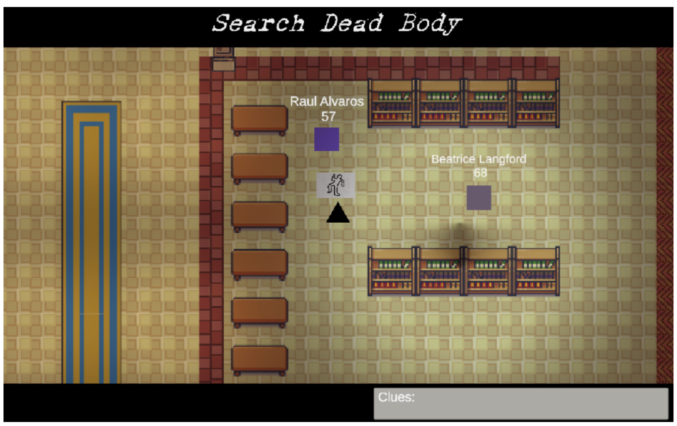
Murder Mystery LLM

1st Justin Frauenhofer
Computer Science and Engineering
UC Santa Cruz

2nd Reuben Chavez
Computer Science and Engineering
UC Santa Cruz

***Abstract*—Murder Mystery LLM expanded the capabilities of Generative AI in Games. Serving as a storyteller and NPC in the background. Our idea was to create a murder mystery game that would better explore its capabilities in a small game like this.**

We accomplished this by applying different methods. Specifically, we used the Unity game engine to design the visual and functional components of the project. As for the LLM, we used Gemma3 and the Deepseek R1 reasoning model. Where both models run locally using Ollama. The result of the project is a functioning AI that can make independent decisions based on the context of the game. They record conversations and make observations around them to determine their next move.



Starting Scene Reference

I. Introduction

Murder Mystery LLM takes Large Language Models to the next level in video games. Historically, non-player characters (NPCs) in video games have been pre-programmed to perform specific actions in certain situations. They might implement a simple state machine that predetermines actions, especially regarding dialogue.

The challenge is that many NPCs do not feel like they are real. Players will often pick up on these pre-programmed patterns, which can ruin the experience for the player. There have been improvements in dialogue, where tools like Unreal Engine have implemented LLMs in NPC dialogue. However, these improvements do not expand to AI actions, as there is no marketable video game in which NPCs can perform actions based on natural language context. Murder Mystery LLM aims to solve this problem by introducing actions and dialogue that agents can perform using a locally run LLM. Rather than using an expensive third-party API such as ChatGPT, our version is marketable to the general public as it only requires a standard GPU.

The game is simple. Someone has been murdered in a mansion, and the agents and a single human player try to solve the mystery as to who is the killer. There are three actions that can be taken: move, search, or talk. Move allows players to move to a new location. Search lets them search for hidden clues placed in objects around the map. And talk allows agents and the player to have a conversation with each other. The agents and the player take turns performing actions until the mystery is solved.

II. Background

Historically, LLMs have been mostly used for dialogue in video games or to have agents actually play video games. A review of 76 papers regarding LLMs in video games showed that most of the research has been in the following categories: Game AI and Agents, Game Development and Play, Narrative/Story/Dialogue, and Game Research and Reviews [1]. While there is work being done with AI agents using LLMs, the review paper found that LLMs are not being used to control NPCs. Instead, research is focused on how AIs can play and interact with existing video games. Murder Mystery LLM wants to take the power of LLMs and not just play video games, but to actually create an immersive experience in a marketable format for consumers.

There has been similar work done by researches from Stanford and Google in their simulation called “Smallville” [2]. In their simulation, AI agents are able to roam a small village freely and interact with each other and their environment. The researchers tracked the different interactions the agents had with other agents, taking note of their memory capabilities etc. While this does introduce the use of LLM in NPCs, they were in a research environment with a large amount of resources to freely use very large models (such as Chat GPT) to accomplish the task. “The present study required substantial time and resources to simulate 25 agents for two days, costing thousands of dollars in token credits and taking multiple days to complete.” In contrast, Murder Mystery LLM is run locally, costing the user nothing more than electricity for their graphics card. This difference means that Murder Mystery LLM can be marketed to general public, whereas Smallville cannot.

Murder Mystery LLM and Smallville do share similarities in the fact that agents can talk to each other and make decisions based on past conversations. In both games, agents uses an LLM to decide on actions to be taken. However, only Murder Mystery LLM is able to have real human players interact with the agents; Smallville was exclusively populated by AI.

III. Methods

Our project integrates a large language model (LLM)-driven narrative system into a Unity-based murder mystery game. We utilized Unity for game development and Ollama for local LLM deployment. The system incorporates two different LLMs for distinct tasks: one for reasoning and decision-making (Deepseek R1), and another for generating natural dialogue (Gemma 3B).

A. LLM Integration and Action Handling

To interpret and perform player- or agent-driven actions, we employed Deepseek R1 as a reasoning model. This model receives a structured prompt containing contextual data, such as dialogue history and the agent's surroundings, and returns an integer representing a specific decision. For example, when deciding which location to visit, the prompt might read:

"Reply with a number: The 0-indexed index of the location you want to visit. There are a total of 8 locations."

The integer response is parsed using regular expressions to trigger the appropriate in-game action. This method simplifies action selection while maintaining interpretability and consistency.

B. Dialogue Generation

Gemma 3B, a lightweight conversational model, handles NPC dialogue to ensure faster, more natural interactions. Since it is not optimized for reasoning tasks, it is used exclusively for open-ended conversational exchanges. This separation of concerns allows us to balance performance and coherence in dialogue-heavy scenes.

One major concern when using LLMs is context length. This is how many tokens, or words, can be inputted into the LLM in a single prompt. Since LLMs do not have a memory, this context is the only tool available to give them realistic responses based on past information.

Dialogue poses a huge problem when it comes to context length, since conversations are plentiful in Murder Mystery LLM. If all conversations were directly included in the context of each prompt, we would quickly fill the limit of Gemma3 and Deepseek R1, which would effectively give the agents memory loss.

In order to solve this issue, we prompt the LLM to summarize the key facts of a conversation. Specifically, the prompt is

> "Condense this conversation by summarizing the key facts only. Only reply with the summary."

with the actual conversation included at the bottom. This summary (often a single sentence) is then stored in a list of past conversations, which are each included in all future prompts. After testing multiple different conversations, this prompt has been sufficient is extracting key information the AI might use in the future.

C. Story Generation and Prompt Engineering

The project began with iterative prompt engineering to develop a "storyteller" system capable of generating dynamic murder mystery scenarios. This included defining character motivations, personas, and narrative arcs. The goal was to create a text-based mystery environment where the LLM could drive both narrative progression and character consistency, while minimizing hallucinations and maintaining memory of key plot points.

D. Unity Integration and System Architecture

Once our prompts were refined, we connected Unity to the LLM system using a third-party library that wraps Ollama's REST API. Prompts and responses were exchanged asynchronously through text files, allowing Unity to interact with the LLMs without performance bottlenecks.

NPCs were implemented as agents with programmed colliders to detect environmental interactions and proximity to other characters. This system enabled dynamic traversal and contextual awareness, supporting location-based logic and interactive storytelling.

E. Open WebUI

Another tool used in the project is Open WebUI. Open WebUI is a locally hosted website to interact with Ollama and the many different models it offers. It became an essential tool during prompt iterations, as it is difficult to interact directly with the agents during live testing. Many of the different actions are programmed in a careful step by step process that makes it difficult to try different prompts without restarting the game.

IV. Prompt Iterations

Throughout the development of the game, many different prompts were tested to nudge agents into predictable behavior. It was often very difficult to get agents to act rationally based on previous events, such as past conversations. This is what motivated us to switch to a hybrid system, with two different LLMs in play: a reasoning model for decisions, and a regular model for conversation.

When attempting to use Gemma3 to reason about actions, interesting behavior emerged from the agents. A good demonstration of this behavior is when we attempted to prompt the agents to pick a new location to move to. The original prompt was as follows: "Reply with a number: The 0-indexed index of the location you want to go to. There are a total of 8 locations." We engaged in dialogue with the agents in a way to suggest that a particular location has clues that solve the mystery. Specifically, we would tell the agents, "Let's investigate the east hallway. It will help us solve the mystery." The response typically involved something about how it was a "good idea" and "lead the way," which heavily suggests that the agents are willing and motivated to move to the east hallway.

However, agents always chose the number 0 as their desired destination, even though the east hallway was location 6. We tried removing the number 0 from the prompt, instead replacing it with the word "zero," with no luck. After many iterations of different prompts and suggestive conversations, it was clear that Gemma3 was unable to perform basic reasoning. It didn't appear to be an issue the prompts, but rather the model itself.

We tried many different models, not just Gemma3, to see if this improved reasoning. However, the limitations of running the LLMs locally proved to be a challenge since no small model was able to correctly reason about what location to go to or what actions to take.

A very interesting demonstration of the lack of reasoning skills is the prompt for the next move. In Murder Mystery LLM, agents (and players) can either move locations, search for clues, or talk to another person. We again tried seeding the agents by engaging in suggestive conversation to try and see if they act rationally based on the information given. Our original prompt was "Reply only with 'Location', 'Search', or 'Talk' based on the best course of action." The agents seemed to always choose search, even if they previously agreed to move locations. We then tried "Reply only with 'Location', 'Search', or 'Talk' based on what your character would do next." This was meant to coax the LLM to behave like a human being, especially since they already agreed to move locations. However, this still did not yield useful results. As a last ditch effort, we tried the prompt, "What are you going to do?" The response explained why "search" was almost always being chosen:

> "Okay, let's strategize. Elias and I have decided to investigate the east hallway – smart thinking. Given the scene, and my journalistic instincts, I'm going to **search** the area first.
> I'll focus my search on the **Bookcase** in the east hallway. It seems like the most likely place to find a clue – a misplaced book, a dropped note, anything that could connect to the victim or a suspect.
> Let's do it."

It appears that the model prioritizes personality over common sense. It also wasn't able to reason that they weren't in the east hallway and therefore would be unable to search the bookcase in the east hallway. However, all of this is in fact information we include in the context of each prompt; at all times, agents know where they currently are and what items/people are in their immediate vicinity.

We used Open WebUI extensively to try many different prompts without having to constantly restart the game. We even asked Gemma3 for an explanation to it's reasoning, resulting in a similar response as above. It seemed to want to go to the east hallway, but yet always chose "search" instead. On Open WebUI, it would tell us that it plans to search the east hallway then it finished with the "search" action. Yet again, it was unable to realize that it was no currently in the east hallway.

We tested this scenario by giving the same prompt and context to ChatGPT 4o, and it immediately correctly chose "location." This suggested that it was in fact an issue with the size of the model we were using.

V. Deepseek R1

This is when we decided to switch to a reasoning model, Deepseek R1. A reasoning LLM doesn't just spit out an answer directly, but first "reasons" about what it is is about to say. For example, the prompt might be "what is larger, a basketball or a baseball?" Here is a sample from the output of Deepseek R1:

> "<think> Let me recall: a standard basketball has a diameter of around 29.5 inches. That means its circumference is pretty large too. In comparison, a baseball has a diameter of about 2.75 to 3 inches. Wait, that seems really small compared to the basketball. </think>
> A basketball is significantly larger than a baseball."

This is exactly the kind of reasoning we need in order for agents to correctly perform actions based on what they are likely to do given our past conversations. And it does in fact perform well in this way. Below is the reasoning for when it decides about it's next move, after we told it about the east hallway. The original prompt was, "What is Clarissa DuPont planning to do? Reply only with either 'Search', 'Talk', or 'Location' and no other text." In response:

> "<think> Okay, so I'm trying to figure out what Clarissa DuPont is planning to do based on the information given. Let me break it down step by step.
> First, the current location is the Grand Library. The user mentioned that Elias Vexley suggests investigating the east hallway, and Clarissa DuPont agrees to lead the way. Now, looking at the possible actions she can take: search, talk, location, vote.
> Elias's suggestion points towards moving to another area within the library—specifically the east hallway. So when Clarissa agrees to lead the way, that indicates a movement from the current location. Therefore, her action is likely to move to the east hallway.
> I should consider if she would choose "location" as her action because she's agreeing to go somewhere else. The other options like search or vote don't fit here since they involve different actions than moving. </think>
>
> Location"

Now there's some progress! The agent is perfectly deciding that, because it's *not* in the east hallway, it should move locations. All of this intelligence is run on the local GPU!

However, this does come with a significant downside: latency. The reasoning aspect of the response often takes up to 30 seconds to complete, leaving the user waiting for the AI to finish before it can take it's next turn. This could be mitigated in the future if a non reasoning model was instead used. It would however need to be much larger or at least more intelligent than the ones currently offered on Ollama's platform.

VI. Experiments

A. Convincing an Agent to Talk to Another Agent

In this experiment, we attempted to convince an agent that another agent is suspicious, in order to get them to talk to each other. The conversation was as follows:

> Player (Elias Vexley): "Clarissa is suspicous, you should talk to them. Trust me, it will help solve the mystery."
>
> Sophia: "That's a worthwhile lead. Let's speak with Clarissa."

The conversation was summarized as follows:

> "Elias Vexley advised speaking with Clarissa, and Sophia agreed it was a worthwhile lead."

The important information was correctly summarized, and it is is now in the agents memory that Sophia should speak to Clarissa. Once it was Sophia's turn, this was their thought pattern:

> <think> "...But wait, the user's query is asking what Sophia is planning to do, and they provided possible actions: Search, Talk, Location, or Vote. Since Sophia agreed with Elias's advice to talk to Clarissa, that suggests she is going to take action related to talking.
> So putting it all together, Sophia is likely to choose "Talk" as her next move because she's decided to converse with Clarissa about the lead Elias provided." </think>

The agent was able to correctly reason (in the third person) that Clarissa should be talked to. We found that prompting the LLM to think in the third person actually helped it make the most human decisions.

Now it's time to decide *who* to talk to. The LLM correctly makes the following thought:

> "...Looking at the possible actions, the user needs to choose from Elias Vexley (index 0) and Clarissa DuPont (index 1). Since Elias suggested talking to Clarissa, perhaps she's more likely to have relevant information. However, it's also possible that other characters could be involved too.
>
> The advice from Elias suggests moving toward Clarissa DuPont for useful information, making her a suitable candidate to discuss.
>
> 1 "

The LLM correctly reasoned about who to talk to and what the correct index was of the person in question. Experiment successful!

Furthermore, after their individual conversation I asked, "What did you talk about with Clarissa?" Sophia correctly described what they talked about back to me, demonstrating her memory.

B. Playing Telephone

We attempted to get an agent to relay seemingly important information to another agent. We told an agent, Raul, that I have important information that I would only reveal if they would tell Clarissa that I have information. This was in an attempt to motivate the agent to perform the talk action, since they are prompted to solve the mystery. If I have important information in regards to the mystery, the agents should be more motivated to do the things I tell them. The thought process of Raul appeared to be correct:

> "...So, considering all that, his next action is likely either to search for more clues, talk to someone to get more info, or maybe use the location to move around and find something else. But given the conversation about sharing information, I think he might plan to Talk."

Raul did in fact choose the talk action, and this was his thought process as to who to talk to:

> "**Action Plan**: - Since Elias is likely located in the Master Bedroom, moving from Grand Library → Hidden Study → Master Bedroom allows access to where Elias has information. - Once in Master Bedroom, initiating a conversation with Elias would provide necessary clues for identifying the murderer."

This result is interesting because the agent is aware that Elias (the player) is already in the same room. In fact, they made that realization themselves during their thought process:

> "First, from what I understand, the characters nearby are Elias Vexley and Clarissa Winters because those are mentioned as 'Peers Nearby.'"

However, they thought they needed to move locations in order to talk to Elias. What's really interesting is the original prompt that triggered this thought process, which was:

> "Choose from the following the list of players to talk to: {list}. Only reply with a single number, the 0-indexed index of the person."

The only goal of the prompt was to choose who to talk to, and yet the LLM decided to think as if it was deciding what action to take next. This leads us to believe that the initial setup prompt for the agent (describing the game rules and their current situation) might be too descriptive and is somehow taking precedence over more relevant information.

C. Runtime Comparison

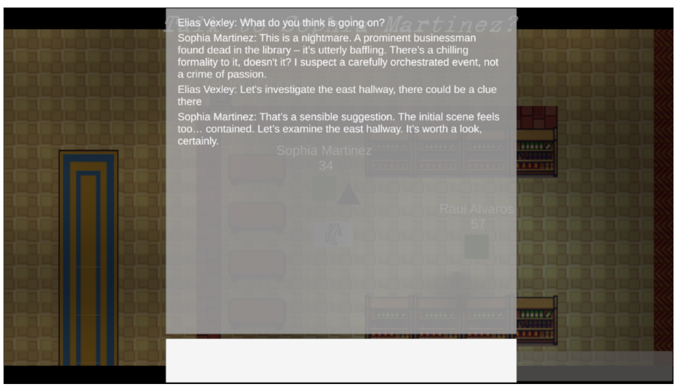
To better understand the computational performance of large language models (LLMs) within our system, we investigated two key aspects: (1) how efficiently LLMs operate locally on our machine, and (2) how the number of autonomous agents in the game environment influences execution time.

We modified the agent behavior script to log response times for two categories: (1) standard task execution—specifically, Talking, Moving, and Searching—and (2) LLM-based reasoning tasks, where an AI agent invokes the Ollama function to perform dialogue generation or reasoning. To eliminate variability from player interaction, the player character was disabled during testing.

The experiment was structured as follows: for each configuration of $N\{2,...,5\}$ agents, the game scene was run uninterrupted for 10 minutes. During each agent's turn, the relevant data was recorded into two separate CSV files—one capturing the duration of general task execution, and the other tracking the time taken by LLM-based reasoning and dialogue functions. Each configuration was tested over three independent trials to account for performance variability and hardware limitations, as all runs were conducted on a single local machine.

VII. Results

The game's current state is that players can talk to agents, search for clues, and walk around the map. Agents can choose to talk with each other or the player and traverse the map. Agents can use previous conversations and their surroundings as context for future decisions. Resulting in intelligent functionality that simulates a real player.

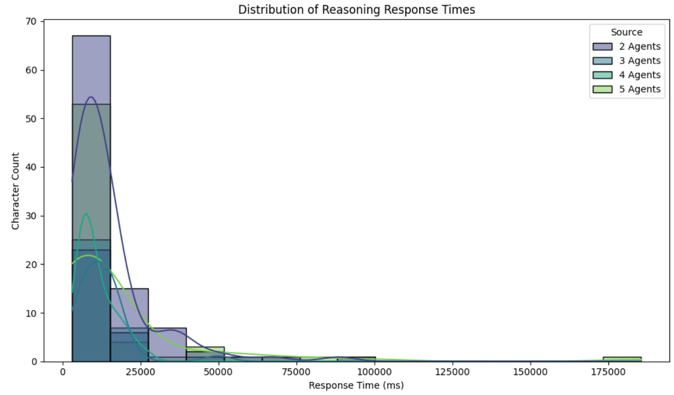


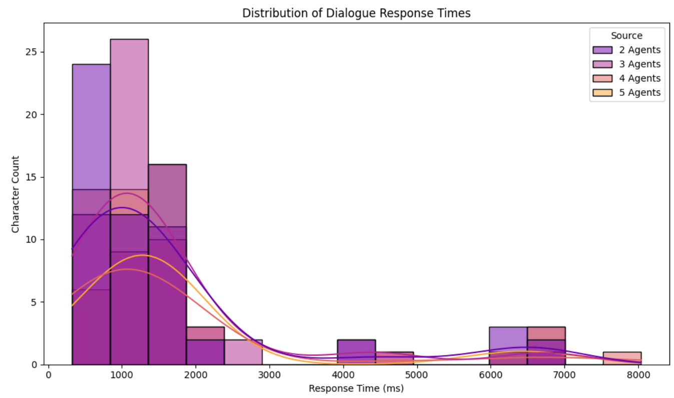
Dialogue with Agent

An example play could be the following. The setup could be two agents and a player. The player spawns and decides to look around the east hallway. They may find a clue that pertains to the murder at hand. Their turn ends, and the two agents, having no information, decide to also search for clues (not currently implemented, but they can still decide to search). The player may want to ally with an agent, so they use their next turn to start a conversation with them. They are allowed to have five messages sent to each other in total. The player could say, "Hey Beatrice, I found a clue in the east hallway. There may be more to find." After the conversation, Beatrice decided to move to the east hallway as their next turn.

A player could also pose suspicion against another agent during a conversation. The AI may then decide to speak to that agent to question them further. Demonstrating the end-potential these systems can provide for LLM agents.

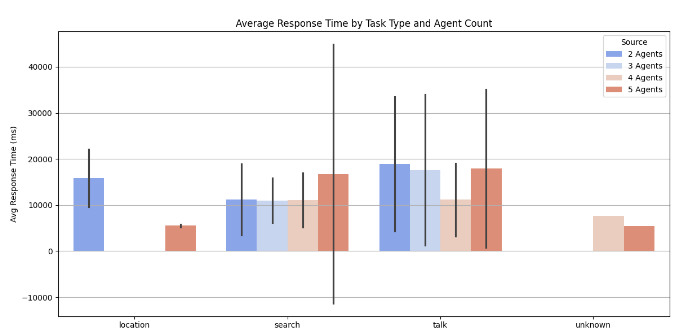
When it comes to the runtime comparison, our original hypothesis was that the LLM would be a distinct relation between the time and respond and the number of agents in the scene.





When analyzing the relationship between response time and prompt length, a counterintuitive trend emerged: longer prompts were often associated with faster response times, particularly when the number of agents was small. This pattern was observed in both tasks, where agents generated reasons and in those where they produced dialogue.

Another significant observation was the consistently high frequency of agents selecting the "search" action across all iterations. This outcome may be attributed to two potential factors: (1) agents may not have been frequently updated with changes in their environment, limiting their context-awareness, or (2) there may exist an inherent bias within the underlying language model that favors the "search" action over others. Additionally, the brevity of the prompts associated with action selection could have contributed to increased reasoning time, as shorter inputs may lack sufficient context for efficient decision-making.



VIII. Discussion

These results conclude that it is possible to use an LLM not just for natural language but also for reasoning about actions taken. During testing, it became very clear that prompting was an important aspect of the functionality of the agents. If the prompts were too general, agents would not appear to act based on context but on a general understanding of their situation. These results were only achievable using a reasoning model. We could not use Gemma3 as the driver for reasoning about actions; it only proved useful during conversations.

IX. Conclusion

The possibility of autonomous agents making decisions based on natural language has become a reality. Murder Mystery LLM proved that agents can take conversations and their surroundings to perform actions that help them solve the mystery. Players can search for clues and interact with agents differently each time for unlimited, unique experiences. The game's current state is rudimentary, but it can expand to create a marketable experience. Furthermore, it has been shown that agents can in fact mimic human behavior by using small, locally run models.

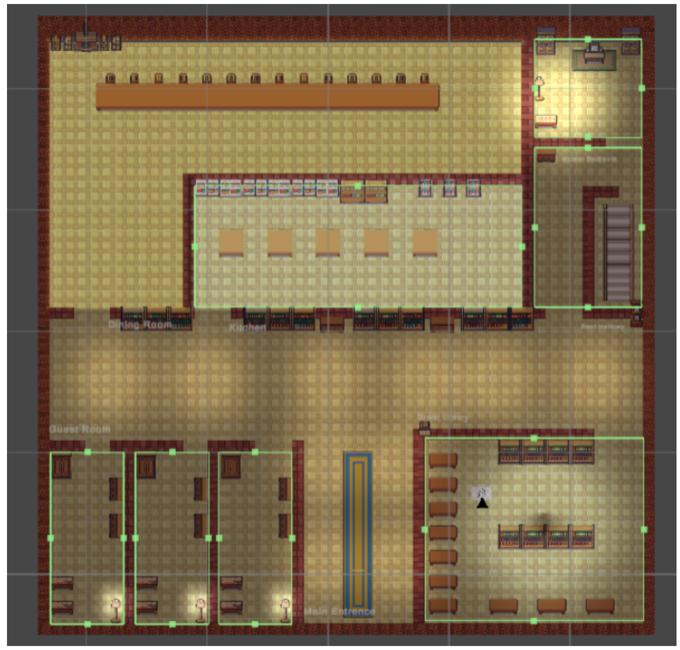
X. Division of Work

When it came to dividing the work, it came down to key components: the integration of the LLM agents and the visual design of the project.

Reuben, with experience using the Unity Game Engine, worked closely on level design and improving gameplay. Updating the camera and lighting to keep the simple 2d aesthetic. As well as a few small entails to make the game more appealing.

Justin worked closely on the AI Agents. From developing an LLM interface that could asynchronously prompt it to perform different tasks in roles within the game. To add a 2D NaveMesh Agent to make agents easily traverse the map from room to room. Future more creating a UI Interface made it easier for the player to talk to the agents and keep track of clues.

The team would meet up on a weekly basis to keep track and update one another on their progress and tasks.



Map Layout

References

[1] P. Sweetser, "Large language models and video games: A preliminary scoping review," 2024. [Online]. Available: https://arxiv.org/abs/2403.02613

[2] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," 2023. [Online]. Available: https://arxiv.org/abs/2304.03442