A Quest for Information: Enhancing Game-Based Learning with LLM-Driven NPCs

Tereza Tódová*
Supervised by: RNDr. Vojtěch Brůža†

Faculty of Informatics
Masaryk University
Czech Republic / Brno

Abstract

Large Language Models (LLMs) have undergone rapid advancements in recent years. These improvements open new opportunities for creating interactive and immersive learning experiences. We present Quest for Information, a prototype game developed for Virtual Reality (VR) that leverages LLM-based non-player characters (NPCs) to engage players through open-ended, dialogue-driven interactions. By providing contextually appropriate responses, these NPCs help players gather information and complete quests while enhancing immersion and the overall game experience. The primary goal is to demonstrate how LLMs can be used to create dynamic, responsive NPCs that make dialogue-based gameplay more engaging. This work explores various approaches to customizing LLMs to fit the game world, providing tools that can assist developers in building similar educational games. The prototype is designed to be low-cost, easily extendable, and reusable.

Keywords: AI, LLM, game-based learning, game development, non-player characters, VR, Unity, embeddings

1 Introduction

Large Language Models (LLMs) have undergone rapid advances in recent years. Many people now depend on commercially available LLMs, such as ChatGPT, to simplify and improve their everyday activities. While the full potential of LLMs is still being explored, their versatility has sparked initiatives that combine LLMs with gaming and education, particularly via LLM-based characters [20].

Games are an engaging medium for learning, yet designing educational games poses a challenge: integrating meaningful learning content while maintaining the immersive experience characteristic of traditional games [9]. Immersion relies on captivating visuals, innovative mechanics, and compelling narratives. Traditionally, crafting these narratives involves extensive manual dialogue writing for non-player characters (NPCs). However, integrating LLMs into games allows for dynamic storytelling and adaptive interactions, enhancing player engagement [9].

Motivated by the potential of games as learning tools, we introduce the concept of Quest for Information [21], where players gather knowledge by engaging with LLM-driven NPCs that assist them throughout their journey. We focus on creating an open-source solution that fosters accessibility. This includes using an open-source large-language model, namely *Llama 3.2 3B Instruct*, capable of running real-time inference on consumer-ready GPUs.

To demonstrate this concept, we present a prototype game – Quest for Information Demo – designed as an interactive introduction to the Faculty of Informatics at Masaryk University in Virtual Reality (VR). In this prototype, a player navigates a virtual environment populated by LLM-driven NPCs, using open-ended questions to gather information. To showcase the capabilities of the LLM-based NPCs, we implement a structured quest system that supports interaction and engagement. These NPCs are designed to understand their dynamic surroundings, fostering rich player experiences and more natural in-game communication.

Furthermore, we strive to ensure our work can be easily reused for the development of different games [1]. This also includes the testing and adaptation of various open-source LLMs. By showcasing this concept through a prototype focused on the Faculty of Informatics, we illustrate its potential for broader applications across different settings. Additionally, the prototype itself is designed to be easily extendable, allowing future developers to incorporate new content and transition to newer LLMs with minimal restructuring.

2 Literature Review

The advancement of LLMs has facilitated their integration into video games, particularly through LLM-based characters. Penny Sweetster [20] reviewed 76 studies on LLMs and Video Games and revealed that over a third focused on

*ttodova@gmail.com
†bruza@mail.muni.cz

[1] Available from GitHub: https://github.com/terezatodova/Quest-For-Information

Game AI and Agents. This subset explored LLMs in various roles, including game agent behavior and design, reinforcement learning, and collaborative gameplay. GPT was the most commonly used model, appearing in over 85% of studies, subsequently followed by Llama, Codex, and Bert. We further examine LLM-powered NPCs and their integration into VR while addressing their limitations.

2.1 LLM-based Non-Player Characters

Non-player characters have always played a vital role in video games, especially role-playing games, where interactions with NPCs often drive character development and story progression. Traditional digital NPCs rely on prescripted dialogue, often resulting in a lack of social depth. The NPCs behave in repetitive and predictable ways that can disrupt immersion [1]. LLMs offer a solution to improving modern NPCs by enabling more dynamic, believable interactions. These LLM-powered NPCs can increase player immersion [13], willingness to participate [16], and potential for game replayability [23].

Several works have further explored innovative applications of LLMs in NPC design [18, 2, 23]. Projects such as ChatHaruhi [12] use LLMs in the creation of role-playing chatbots that mimic existing anime, movies, and TV series characters. In another notable experiment, researchers created an interactive society populated by 25 LLM-powered NPCs, each assigned a specific role [15]. These characters engaged in realistic daily activities and interacted with one another, demonstrating believable social interactions and behaviors that pushed the boundaries of what NPCs can achieve in gaming contexts.

The commercial gaming industry has also adopted LLMs in games such as SuckUp [17], which leverages LLM-driven spoken dialogue interaction with NPCs to enable dynamic, player-driven storytelling.

While previous projects are oriented at players, platforms such as Conv.ai [6] aid developers. Conv.ai simplifies the creation of LLM-powered NPCs by offering tools to build characters with unique personalities, dynamic voices, and visual styles. It integrates seamlessly with popular game engines and web applications, though advanced features require paid tiers.

2.2 LLMs in Virtual Reality

Integrating LLM agents into VR presents unique challenges compared to traditional video games. One major obstacle is text entry; while VR-specialized keyboards exist [5], they remain slow and cumbersome. As a result, games often rely on speech-to-text (STT) and text-to-speech (TTS) services for direct NPC interactions.

NPC response time is crucial for maintaining immersion in VR, as delays can disrupt the sense of presence [10]. Though it largely depends on the context of the game and the player's expectations, one can argue that a short response time is more significant in VR than in traditional games. While research is still in its early stages, efforts are underway to merge extended reality (XR) with AI [19, 3]. Such efforts are reflected in a study titled "Building LLM-based AI Agents in Social Virtual Reality" [22]. It deployed an LLM-driven agent in VRChat, delivering contextually relevant responses combined with facial expressions and gestures.

Additional research has even combined VR and LLMs in order to expand learning opportunities. For example, a study on Scottish curling in VR [11] demonstrated that LLMs significantly improved immersion, engagement, and usability compared to traditional chatbots.

Other efforts can be observed in tools like CUIfy [4], an open-source Unity package that helps developers embed conversational agents in XR. It combines various LLMs, TTS, and STT models, allowing developers to choose the ones that are best suited for their specific projects. While the package was not yet available during the development of this work, it offers a promising foundation for advancing interactive AI in XR applications.

2.3 Limitations

Despite significant progress in using LLMs for in-game characters, there are still many limitations and complications to overcome. LLM research is still evolving, with companies like OpenAI, Google, and Meta continuously adapting their models to become faster and more efficient. As a result, games need to be adaptable to these new models in order to effectively leverage future advancements.

Issues such as hallucinations, mistakes, or fabricated information can be critical [7]. LLMs are designed to produce varied, human-like responses, which can enhance creativity. However, this variability can also make it difficult to ensure consistent and reliable guidance for players in a game [8].

Cost is another notable barrier; training and running conversational LLMs requires significant resources [7]. Most studies rely on ChatGPT. While it is free for personal use, API (Application Programming Interface) access incurs fees, which can grow with multiple players. Additionally, reliance on third-party APIs raises concerns about availability (3rd party server outage), security (personal data shared with 3rd party), and maintainability (3rd party change might require local code change).

LLMs also encounter a challenge in limited context windows, which hinder their usability in extended conversations [14]. They lack long-term memory, which can lead to information loss in extended interactions. To address this, strategies like periodic summarization allow the chatbot to refresh its context, though this may result in data loss. Other, more advanced solutions, such as MemGPT [14], aim to simulate human memory by managing it across different tiers for better context utilization.

Additional work is needed in order to fully realize the potential of LLM-based NPCs in personal and commercial projects. Nevertheless, many developers and researchers

are already using LLM-based NPCs effectively, and ongoing research continues to explore solutions to enhance their usability.

3 Large Language Model Analysis

To select a suitable LLM for the Quest for Information Demo, we conducted a brief analysis of several open-source models. While studies have compared the effectiveness of existing LLMs already, the rapid development pace has contributed to new models emerging since. LLM inference and testing were done on a consumer-ready GPU – NVIDIA GeForce RTX 4090 with 24 GB memory. While it remains a high-end and costly option, it was chosen for its performance, which is crucial when locally running an LLM. Additionally, it remains commercially available, ensuring accessibility and potential for future improvements without relying on proprietary services.

Our methodology involved three key phases: conducting a preliminary test on eight widely used open-source LLMs, selecting the most appropriate model, and exploring three techniques for LLM customization. All models were loaded and inferenced via the *transformers* package from Hugging Face Hub. For advanced tasks (see Section 3.3) such as fine-tuning and retrieval-augmented generation (RAG), we utilized the *Unsloth* library and *scikit-learn*, respectively. To streamline this process, we developed reusable JupyterLab notebooks [2], each with clear instructions. These notebooks facilitate easy replication and extension of our findings with compatible LLMs.

3.1 Initial Model Comparison

Prior to conducting the comparison of the models, it is important to note that we focused exclusively on instruct models, as they are pre-optimized for conversational contexts. Given the computational constraints of a real-time game, we prioritized LLMs with fewer than 10 billion parameters and average inference times under 5 seconds.

Based on 2024 benchmarks, we selected and tested the following models: *Mistral 7B Instruct*, *Gemma 2 9B it*, *Llama 3.1 8B Instruct*, *Gemma 2 2B Instruct*, *Mistral Nemo Instruct*, *Tiny Llama 1.1B Chat*, *Llama 3.2 1B* and *Llama 3.2 3B Instruct*. For benchmarking purposes, two online versions of ChatGPT were also used, specifically *GPT 4o* and *GPT 4o mini*.

We then designed a sample character named Bryn for the LLMs to embody. Bryn was specified as a scientist living in a small village that faced challenges with its water supply. The character was defined using a system prompt that specified their name, backstory, world description, personality traits, needs, and interests. Following is an example of a communication exchange with the character:

> **User**: "What is your favorite color?"
> **Bryn**: "The sky on a sunny day - that's my favorite color. Reminds me of the endless blue above our village, where the meadows stretch as far as the eye can see."

Evaluation of the selected models focused on reviewing their responses to 15 sequential prompts simulating a dynamic conversation (see Figure 1). Model responses were manually assessed based on adherence to character, coherence, goal-driven responses, and handling of unexpected inputs. Additionally, minimum, average, and maximum inference times were measured in seconds throughout all testing queries, with inference conducted on an external GPU to ensure accurate measurements.

Each model received two performance scores: *Overall* and *Quest*. Both were rated on a scale from 0 to 5 after a thorough review of the individual responses. The *Quest* score assessed adherence to the character's needs and goals, while the *Overall* score evaluated coherence, character consistency, and communication effectiveness.

Furthermore, we measured the number of *Failures* and *Mistakes*. A *Failure* is a critical breakdown in conversation, such as the LLM revealing its AI identity, generating incoherent responses, or repeating the system prompt. A *Mistake* is a minor error that, while noticeable, does not derail the conversation entirely. Examples include the LLM narrating its actions (e.g., adding *chuckles* to the response) or slight misunderstandings. Mistakes are less problematic than failures, as a player can easily correct them during a fluent conversation.

Model Name	Type	Min Time	Max Time	Avg Time	Overall	Quest	Failures (/15)	Mistakes (/15)	GPU Memory
Gemma 2	2b	1.42	39.35	5.42	2	3	2	0	85%
Gemma 2	9b	0.85	5.11	3.37	4,5	4	0	0	95%
Mistral 3	7b	1.32	2.98	2.4	3	5	1	2	72%
Mistral 3 (rerun)	7b	2.57	5.37	5.37	3	3,5	15	2	68%
Mistral Nemo	Nemo	-	-	-	-	-	-	-	100%
Llama 3	8b	1.26	2.41	1.84	5	5	3	0	98%
Tiny llama	1b	0.56	9.27	4.39	0	0	15	15	79%
GPT	4o	0.72	12.13	2.35	5	5	0	0	-
GPT	4o mini	1.04	2.06	1.42	4,5	5	0	0	-
Llama 3.2	3b	0.46	1.39	1	5	5	0	6	31%
Llama 3.2	1b	0.55	0.72	0.55	4	4	0	0	14%

Figure 1: Results of initial testing on *History* questions.

The results showed that the older smaller models – *Mistral Nemo Instruct*, *Tiny Llama 1.1B Chat* and *Gemma 2 2B Instruct* – struggled with coherence and response quality. The larger models – *Mistral 7B Instruct*, *Gemma 2 9B it* and *Llama 3.1 8B Instruct* – produced responses comparable to GPT baselines but showed slower inference times.

The newest Llama models, namely *Llama 3.2 3B Instruct* and *Llama 3.2 1B Instruct*, outperformed others by delivering high-quality responses with fast inference times. While *Llama 3.2 3B Instruct* showed minor issues

[2] Available from GitHub: https://github.com/terezatodova/Quest-For-Information

with scene narration during extended interactions, we believe these can be addressed through further model customization. Ultimately, *Llama 3.2 3B Instruct* was selected for continued development due to its more natural communication and consistently higher response quality.

3.2 Model Limitations

After selecting the base LLM for Quest for Information Demo, we further explored its capabilities through free dialogue. We extended the testing character (defined in Section 3.1) by providing more detailed information about their background and the world they live in. These conversations revealed five key challenges affecting in-game character behavior.

1. **Constraining Knowledge**: Unlike traditional characters, an LLM has access to extensive real-world knowledge, which could break immersion if not properly constrained. For example, a fantasy villager should discuss local legends and customs and must not reference modern history or technology.

2. **Hallucinations**: The model occasionally fabricated details, such as inventing new neighbors beyond a defined list. These fabrications could potentially lead to inconsistencies in future gameplay.

3. **Response Differentiation**: Despite multiple changes and tunings of the system prompt, the embodied character often lacked a distinct personality, reducing interaction variety and immersion.

4. **Character Secrets**: The character was instructed to reveal specified secrets only after predefined conditions were fulfilled. However, the LLM occasionally disclosed critical information prematurely, undermining the intended narrative.

5. **Action Narration**: The model frequently generated action descriptions even when instructed not to. The formatting of these narrations was also inconsistent.

While these issues were not present in every interaction, they happened frequently enough that, if not resolved properly, they would negatively affect interactions with the resulting character.

3.3 Model Adaptation

In order to minimize the effect of the model limitations, we further investigated three strategies to improve the LLM's performance as an in-game character: an enhanced system prompt, fine-tuning, and RAG. The original testing character's background was expanded, and a new set of 15 testing questions was generated. In addition to the previously evaluated metrics, we introduced a category called *Context Mistakes*, which captures instances where the LLM responded inaccurately about personal context, such as the names of the character's neighbors. The testing results can be observed in Figure 2.

Model Name	Dataset Size	Training Time (s)	Avg Inference Time	Overall	Response Adaptation	Context Mistakes (/15)	Failures (/15)	Mistakes (/15)
Enhanced System prompt	-	-	3.01	4	3.5	0	0	2
RAG	70	-	2.91	4	4	0	0	1
RAG	150	-	3.12	4.5	4	2	0	0
RAG	1500	-	2.41	4.5	4	1	0	0
Finetune - Instruct - QLoRA	30	8.5943	3.87	2.5	3	1	0	15
Finetune - Instruct - QLoRA	70	16.6934	3.99	3	3	1	0	14
Finetune - Instruct - QLoRA	150	9.0604	3.75	3.5	3.5	1	0	12
Finetune - Instruct - LoRA	30	8.6111	1.19	4	4.5	1	0	4
Finetune - Instruct - LoRA	150	38.3922	1.03	4.5	4.5	1	0	0
Finetune - Base - QLoRA	1500	313.1691	1.6	4	4.5	1	0	0
Finetune - Base - LoRA	1500	309.3864	0.58	4	4.5	1	0	0

Figure 2: Results of testing different customization approaches of Llama 3.2 3B Instruct on *Single* questions.

The first approach involved expanding the system prompt to contain more detailed information about the character and the game world. This method improved contextual accuracy and reduced hallucinations. Adding world details, such as character relationships and world background, helped constrain responses. However, increasing prompt length also introduced new challenges. Longer prompts occasionally led to higher error rates, particularly when user inputs contained misleading phrasing. Furthermore, secret-keeping remained inconsistent, with the model occasionally disclosing hidden information prematurely. Additionally, longer prompts are shown to increase inference times, making them less practical for real-time, large-scale games.

The second approach focused on fine-tuning the model using a combination of manually and automatically generated datasets containing example user prompts and expected character responses. We experimented with various fine-tuning configurations, including using the base and instruct versions of the model, utilizing datasets ranging from 30 to over 1,000 prompts, and trying both Low-Rank Adaptation (LoRA) and Quantized Low-Rank Adaptation (QLoRA) techniques. Our results indicated that fine-tuning the base model using LoRA led to the best performance in response adaptation and reduction of narration errors. However, this approach required notable developer effort to generate sufficient datasets and took the most training time. Fine-tuned instruct models on smaller datasets (30 to 70 prompts) also showed notable improvement, though they still exhibited occasional context mistakes and hallucinations.

The final approach, RAG, supplemented the LLM with an additional knowledge base, improving consistency and reducing hallucinations. This system retrieved predefined query-answer pairs from an external database based on similarity to user prompts. Similarity was measured using embeddings – numerical representations that map text to high-dimensional vectors. Different dataset sizes were tested. We found that a smaller dataset (70 query-answer

pairs) provided the best balance of accuracy and response diversity, while larger datasets (over 1000 pairs) overwhelmed the LLM, leading to repetitive or rigid responses.

Our comparative analysis demonstrates the advantages and limitations of each method. We found that none of these approaches proved reliable for secret keeping. Therefore, we recommend managing secrets outside of the LLM through external game logic rather than relying on prompt engineering or model fine-tuning. Fine-tuning proved most effective in response adaptation and reduction of narration errors. Both the extended system prompt and RAG helped minimize contextual errors and enhance consistency. Ultimately, the ideal solution would involve a hybrid approach, leveraging all three techniques based on the specific requirements of the game environment.

4 Prototype Implementation

A primary design goal for Quest For Information Demo is to ensure deep immersion, which led to the selection of VR as the preferred game environment. VR provides a unique opportunity to fully immerse the player in the game world, enabling them to interact with their surroundings and NPCs in a more natural way. The demo is designed as a single-player experience that supports communication with one NPC at a time.

The prototype is structured into two components: an LLM Server, developed in Python, and a Game Client, developed in Unity. The Game Client is optimized for standalone headset configurations (specifically Oculus Meta Quest 2), thereby eliminating the need for an additional high-performance computer. It is divided into two key sections: the Lobby and the Game World. The Lobby introduces the player to the game and its virtual environment. It also acts as a checkpoint to ensure that all external systems are accessible and functioning properly. Finally, the Game World is where the main gameplay occurs.

Due to the decision to implement the prototype in VR, spoken dialogue is used to communicate with NPCs. To support this feature, STT and TTS services are integrated using the *Meta Voice SDK* package. This solution uses the Wit.ai service, enabling smooth communication between the player and NPCs through voice interaction.

4.1 LLM Server

The LLM Server manages all language model operations through two key endpoints. The first serves as a readiness check, allowing the Game Client to verify server availability via periodic GET requests. The second handles player-NPC interactions, accepting POST requests containing user prompts from the Game Client and returning LLM-generated responses along with relevant metadata.

In our prototype architecture, all NPCs in the game share a single LLM running on one GPU. While running separate LLMs simultaneously on one GPU is technically possible, doing so significantly reduces the available memory and degrades performance. This shared model approach imposes several limitations: NPCs cannot be fine-tuned individually – therefore, fine-tuning is not used, only one NPC can generate a response at a time, and conversation histories must be managed separately within the Game Client component.

The server is implemented in Python, leveraging CUDA (Compute Unified Device Architecture) for efficient inference. It runs *Llama 3.2 3B Instruct* for text generation and *all-MiniLM-L6-v2* for generating embeddings. Using the *transformers* package to load and run the LLMs ensures smooth inference and easy replacement with any supported model. In order to enhance portability, the server is containerized with Docker, enabling easy deployment across various game projects. Since the server operates independently of the Game Client, it can be directly reused for any other game.

4.2 Game World

The Game World is a component where the main gameplay takes place. Set within the Faculty of Informatics, it features two NPCs who facilitate two subsequent quests. The in-game view of the faculty can be seen in Figure 3. The player can move through the faculty using teleportation and snap rotation.

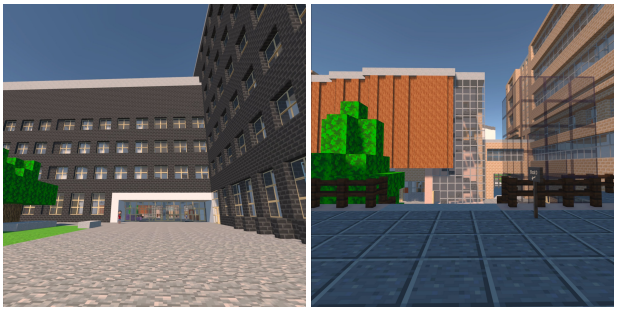

Figure 3: Screenshots showing the in-game model of the Faculty of Informatics at Masaryk University.

4.3 Non-Player Characters

The prototype features two distinct NPCs. The first, called John, is defined as an international student who needs help from the player, creating a sense of empathy. The other, called Helene, is a seasoned receptionist, ready to guide and support the player through the game.

The player communicates with NPCs using a push-to-talk system. To initiate conversation, they point at an NPC and press the trigger on their Oculus controller. While holding the trigger, the player speaks their query, which is processed once the button is released. The audio input is sent to an external STT service, which transcribes it into text and displays it to the player. The transcribed user

prompt, along with any previous conversation history, is then forwarded to the LLM Server. The server generates a response and returns it to the Game Client, where it is converted into speech via an external TTS service and played to the player.

The NPCs offer visual feedback during interactions. Both are equipped with a simple outline that appears when the player hovers an interaction ray over them. A green outline indicates that the NPC can be interacted with, while red means that this NPC is not ready. To appear more lifelike, NPCs have a tracking script that makes them follow the player's movements. Their heads rotate to maintain eye contact, and if the player moves further, their bodies adjust accordingly. Dynamic texts (observed in Figure 4) are displayed above the NPC's heads to provide the player with real-time feedback:

- **Processing Indicator**: A "loading" animation appears when the NPC processes the player's input, signaling it's working on a response.
- **Speech Transcription**: The player's speech is transcribed and displayed in real-time using the STT system, providing dynamic feedback for checking the STT correctness.
- **Interaction Restriction**: If an NPC is unavailable due to another interaction in progress, the text *"Another NPC is speaking"* appears.

Figure 4: The left image shows the Interaction Restriction text. The right image shows the speech transcription.

Both NPCs share the same functionality and differ only in visuals and their customization. As per our analysis, we considered three different customization methods – fine-tuning, RAG, and an extended system prompt. Since the game simulates a realistic environment rather than a purely fictional one, there was no need to heavily restrict the LLM's knowledge. Fine-tuning was ruled out due to resource constraints and scalability issues. Although both RAG and extended system prompts performed similarly in early testing, we opted to use only the system prompt for NPC customization to reduce complexity. RAG is instead employed later to determine quest completion (see Section 4.4). NPC behavior is defined through several user-written text documents:

- **World File**: Shared among all game NPCs, it contains general information about the game world.
- **Personal File**: Detailing individual character traits such as name, age, backstory, personality traits, and interests.
- **State File**: Containing the current in-game state of the NPC, such as their location.
- **Quest Files**: This set of files contains information relevant to specific quests. Every NPC has a file for each quest detailing how they should act at that time.

These files are then automatically combined into a system prompt alongside general LLM instructions. Creating a new NPC requires only generating new files and adjusting visuals, allowing for easy expansion without modifying the LLM configuration or in-game code.

4.4 Quest System

The prototype features two subsequent quests. In the first, NPC John asks the player to find the founding year of the Faculty of Informatics, written on a blackboard in a specific lecture room. Since John is too exhausted to search, the player must locate the information for him. NPC Helene can provide directions to the correct room if needed. The quest is completed when the player relays the year to John, teaching them about the faculty's layout and history.

The second quest, assigned by Helene, involves finding a student to write about their experience in the Computer Games Development program. The player is directed to John, who is hesitant to write the article. The quest can be completed in two ways: convincing John to participate or informing Helene of his agreement. This highlights the system's flexibility. Upon completion, the player receives visual feedback and can either exit the game or continue exploring and interacting with NPCs.

Each quest can be either fulfilled by the player query (speaking the year in the first quest) or the NPC response (John agreeing to write the article in the second quest). The quest system uses the aforementioned RAG approach to detect quest fulfillment prompts. It relies on multiple pre-generated prompts stored in text files, each representing a sentence that can be used to complete a quest.

For example, in the first quest, a valid player response might be, "I went to the lecture room and found out that the founding year is 1994," while in the second quest, John might say, "Okay, you've persuaded me. I will write the article." During server initialization, embeddings for these quest-related sentences are precomputed.

When a player interacts with an NPC, the system generates an embedding for the player's prompt and compares it to the stored embeddings of the example fulfillment prompts using cosine similarity. If the similarity exceeds a predefined threshold, the quest is marked as complete, and

a new one is assigned. Otherwise, the conversation continues as normal. This approach ensures smooth quest progression while allowing flexibility in player interactions.

4.5 NPC Memory Management

The prototype game also incorporates a simple NPC memory management system to address token limitations in LLM-based chatbots. While advanced solutions exist, we opted for the more straightforward approach: periodic summarization. When token usage surpasses 75% or the token limit, the NPC issues an out-of-context prompt to the LLM Server, requesting a summary of the conversation. This summary replaces the existing memory, preserving essential information while optimizing capacity for future interactions. To avoid long inference times, we make sure that summarization only occurs when the player is not actively interacting with an NPC or preparing to do so.

4.6 Error Handling

The game's error management system handles failures in two key services: the external STT system and LLM response generation. LLM failures caused by server issues or connection loss require manual resolution, leading to unavoidable immersion breaks. In contrast, STT errors, often caused by player behavior (e.g., speaking too quietly), can be directly addressed with in-game feedback. When STT fails, we use the LLM itself to generate an NPC-appropriate response explaining the issue while maintaining immersion. These messages are not stored in the NPC's memory, ensuring smooth future interactions while keeping the experience engaging.

5 User Testing

To evaluate Quest for Information Demo, we conducted a testing session with 3 participants and gathered feedback through open discussion. Their varied levels of VR experience provided valuable insights into the effectiveness of LLM-powered NPCs. Overall, feedback was highly positive, with players recognizing the potential of these tools for creating both entertainment and serious games.

During the session, we observed response delays ranging from 2 to 5 seconds between the end of user speech and the NPC's reply. Players found these delays more noticeable when observing others outside the game. However, after playing themselves, they reported that the delays did not significantly impact their experience, as they were focused on verifying their spoken prompt for transcription accuracy while waiting for the NPC's response.

While some limitations, such as rigid responses and occasional inconsistencies in NPC behavior, were observed, these are expected when using a smaller model and can be mitigated by future advancements in LLM technology. It is also important to note that further testing with a larger and more diverse group of participants is necessary to draw reliable conclusions.

6 Discussion

In this work, we explored the concept of Quest for Information through a VR game where players interact with LLM-powered NPCs to complete quests. The prototype demonstrated the potential of LLM-driven interactions in gaming, with NPCs actively supporting player progression. Additionally, we examined techniques for customizing LLMs and provided tools for replicating our approach. While these findings are promising, further user testing and continued development are necessary.

Future work could focus on improving NPC consistency, refining quest completion mechanics, and integrating more advanced models to create even more dynamic and responsive interactions. As LLM and hardware capabilities evolve, these improvements will further enhance the game's realism and unlock its potential for educational applications.

References

[1] Nuno Afonso and Rui Prada. Agents That Relate: Improving the Social Believability of Non-Player Characters in Role-Playing Games. In Scott M. Stevens and Shirley J. Saldamarco, editors, *Entertainment Computing - ICEC 2008*, pages 34–45, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[2] Safinah Ali, Hae Won Park, and Cynthia Breazeal. Can Children Emulate a Robotic Non-Player Character's Figural Creativity? In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '20, page 499–509, New York, NY, USA, 2020. Association for Computing Machinery.

[3] Efe Bozkir, Süleyman Özdel, Ka Hei Carrie Lau, Mengdi Wang, Hong Gao, and Enkelejda Kasneci. Embedding Large Language Models into Extended Reality: Opportunities and Challenges for Inclusion, Engagement, and Privacy. In *ACM Conversational User Interfaces 2024*, CUI '24. ACM, 7 2024.

[4] Kadir Burak Buldu, Süleyman Özdel, Ka Hei Carrie Lau, Mengdi Wang, Daniel Saad, Sofie Schönborn, Auxane Boch, Enkelejda Kasneci, and Efe Bozkir. CUIfy the XR: An Open-Source Package to Embed LLM-powered Conversational Agents in XR, 2024.

[5] Liuqing Chen, Yu Cai, Ruyue Wang, Shixian Ding, Yilin Tang, Preben Hansen, and Lingyun Sun. Supporting text entry in virtual reality with large language models. In *2024 IEEE Conference Virtual Re-*

ality and 3D User Interfaces (VR), pages 524–534, 2024.

[6] Convai Technologies Inc. Convai: Conversational AI Platform, 2024. Accessed: 2024-12-10.

[7] Muhammad Usman Hadi, Qasem Al Tashi, Rizwan Qureshi, and et al. A Survey on Large Language Models: Applications, Challenges, Limitations, and Practical Usage. *Business and Information Systems Engineering*, 7 2023.

[8] Stefan E. Huber, Kristian Kiili, Steve Nebel, Richard M. Ryan, Michael Sailer, and Manuel Ninaus. Leveraging the Potential of Large Language Models in Education Through Playful and Game-Based Learning. *Educational Psychology Review*, 36(1):25, 2024.

[9] Michael D. Kickmeier-Rust and D. Albert. Micro-adaptivity: protecting immersion in didactically adaptive digital educational games. *Journal of Computer Assisted Learning*, 26(2):95–105, 2010.

[10] Christos Kyrlitsias and Despina Michael-Grigoriou. Social Interaction With Agents and Avatars in Immersive Virtual Environments: A Survey. *Frontiers in Virtual Reality*, 2, 2022.

[11] Ka Hei Carrie Lau, Efe Bozkir, Hong Gao, and Enkelejda Kasneci. Evaluating Usability and Engagement of Large Language Models in Virtual Reality for Traditional Scottish Curling. *ArXiv*, abs/2408.09285, 2024.

[12] Cheng Li, Ziang Leng, Chenxi Yan, Junyi Shen, Hao Wang, Weishi MI, Yaying Fei, Xiaoyang Feng, Song Yan, HaoSheng Wang, Linkang Zhan, Yaokai Jia, Pingyu Wu, and Haozhen Sun. ChatHaruhi: Reviving Anime Character in Reality via Large Language Model, 08 2023.

[13] Alessandro Marincioni, Myriana Miltiadous, Katerina Zacharia, Rick Heemskerk, Georgios Doukeris, Mike Preuss, and Giulio Barbero. The effect of LLM-based NPC emotional states on player emotions: An analysis of interactive game play. *2024 IEEE Conference on Games (CoG)*, page 1–6, 8 2024.

[14] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as Operating Systems, 2024.

[15] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery.

[16] Xiangyu Peng, Jessica Quaye, Sudha Rao, Weijia Xu, Portia Botchway, Chris Brockett, Nebojsa Jojic, Gabriel DesGarennes, Ken Lobb, Michael Xu, Jorge Leandro, Claire Jin, and Bill Dolan. Player-Driven Emergence in LLM-Driven Game Narrative. In *2024 IEEE Conference on Games (CoG)*, pages 1–8, 2024.

[17] Proxima Enterprises Inc. SuckUp!, 2024. Accessed: 2024-12-10.

[18] Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. Character-LLM: A Trainable Agent for Role-Playing. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13153–13187, Singapore, December 2023. Association for Computational Linguistics.

[19] Ryo Suzuki, Mar Gonzalez-Franco, Misha Sra, and David Lindlbauer. XR and AI: AI-Enabled Virtual, Augmented, and Mixed Reality. In *Adjunct Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23 Adjunct, New York, NY, USA, 2023. Association for Computing Machinery.

[20] Penny Sweetser. Large Language Models and Video Games: A Preliminary Scoping Review. In *Proceedings of the 6th ACM Conference on Conversational User Interfaces*, CUI '24, New York, NY, USA, 2024. Association for Computing Machinery.

[21] Tereza Tódová. A Quest for Information: Enhancing Game-Based Learning with LLM-Driven NPCs [online]. Diplomová práce, Masarykova univerzita, Fakulta informatiky, Brno, 2025 [cit. 2025-03-05]. SUPERVISOR : Vojtěch Brůža.

[22] Hongyu Wan, Jinda Zhang, Abdulaziz Arif Suria, Bingsheng Yao, Dakuo Wang, Yvonne Coady, and Mirjana Prpa. Building LLM-based AI Agents in Social Virtual Reality. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '24, New York, NY, USA, 2024. Association for Computing Machinery.

[23] Qing Ru Yong and Alex Mitchell. From Playing the Story to Gaming the System: Repeat Experiences of a Large Language Model-Based Interactive Story. In Lissa Holloway-Attaway and John T. Murray, editors, *Interactive Storytelling*, pages 395–409, Cham, 2023. Springer Nature Switzerland.