Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
University 8 May 1945 , - Guelma -
Faculty of Mathematics, Computer Science and Sciences of Matter

Department of Computer Science

Final Year Dissertation- Master's Degree

Field : Computer Science
Option : Informatics Systems

Topic

AI-Powered Virtual Campus Exploration : Integrating LLM-Driven Interactions within a 3D Virtual Reality Model

Supervised by :
Dr. Chohra Chemseddine

Realized by :
Abdellaoui Salim

June 2025

Acknowledgments

Alhamdulillah, I have finally reached this milestone and achieved one of my life's greatest goals. First and foremost, I thank Allah for granting me the strength, knowledge, perseverance, and opportunity to complete this research work successfully.

I would like to express my deepest gratitude to my supervisor, Dr. Chohra Chemseddine, for His invaluable guidance, scientific expertise, patience, and continuous encouragement throughout this journey. His insightful feedback and unwavering support were instrumental in shaping this work.

My sincere appreciation also goes to Dr. Zineddine Kouahla for his mentorship, wisdom, and motivation, which greatly influenced my academic growth and mindset during my years at the university.

A special and heartfelt thanks goes to my CREATIVA Family your endless inspiration, love, and belief in me have been my driving force, I've known you for two months now, but it feels like years have passed. No matter how much I thank God for you, it would never be enough.

I am forever grateful to my beloved mother, my dear sister, and my brother Abdenour for their sacrifices, prayers, and unconditional support. Your presence in my life is my greatest blessing.

I extend my thanks to all the professors of the Computer Science Department at the University of May 8, 1945, Guelma, for their knowledge, dedication, and guidance. I was fortunate to learn from such esteemed educators, whose lessons will remain with me beyond academia.

I also owe much to my friends Nabil, Zaki, Wassim, and Adem, as well as my colleagues and the entire Class of 2025, for their camaraderie and encouragement.

Finally, to my family no words can truly express my gratitude. You have stood by me at every step, offering love, patience, and motivation. May Allah reward you with health, happiness, and prosperity, and may I always make you proud.

ABSTRACT

As higher education institutions increasingly adopt digital platforms to enhance learning and outreach, many universities have begun offering virtual tours and 3D representations of their campuses. However, despite advances in 3D modeling and immersive technologies, these virtual environments often remain limited in their interactivity and adaptability. The majority of existing virtual campus systems rely on pre-scripted dialogue, static user pathways, and minimal personalization, resulting in experiences that are visually rich but cognitively shallow.

At the same time, breakthroughs in Artificial Intelligence particularly in the form of Large Language Models (LLMs) have opened new possibilities for creating intelligent, responsive, and conversational agents. Yet, these powerful models have not been fully leveraged within virtual reality (VR) environments, especially in educational contexts where natural dialogue, contextual understanding, and adaptive guidance could significantly improve user engagement and learning outcomes.

This gap raises a critical question : How can we design and implement a virtual campus environment that not only allows immersive exploration but also enables meaningful, real-time interaction through AI-powered dialogue ? Furthermore, what are the technical, design, and educational considerations for integrating LLMs into VR spaces to create dynamic, responsive, and personalized experiences for students ?

RÉSUMÉ

Alors que les établissements d'enseignement supérieur adoptent de plus en plus des plateformes numériques pour améliorer l'apprentissage et l'accès, de nombreuses universités ont commencé à proposer des visites virtuelles et des représentations 3D de leurs campus. Cependant, malgré les avancées dans la modélisation 3D et les technologies immersives, ces environnements virtuels restent souvent limités en termes d'interactivité et d'adaptabilité. La majorité des systèmes de campus virtuels actuels reposent sur des dialogues préprogrammés, des parcours utilisateurs statiques et une personnalisation minimale, ce qui donne lieu à des expériences visuellement riches mais cognitivement pauvres.

Parallèlement, les percées en intelligence artificielle, notamment sous la forme de grands modèles de langage (LLMs), ont ouvert de nouvelles possibilités pour créer des agents intelligents, réactifs et conversationnels. Pourtant, ces puissants modèles ne sont pas encore pleinement exploités dans les environnements de réalité virtuelle (VR), en particulier dans les contextes éducatifs où le dialogue naturel, la compréhension contextuelle et les conseils adaptatifs pourraient améliorer considérablement l'engagement des utilisateurs et les résultats d'apprentissage.

Cette lacune soulève une question cruciale : comment concevoir et mettre en œuvre un environnement de campus virtuel qui permette non seulement une exploration immersive, mais aussi une interaction significative en temps réel grâce à un dialogue

alimenté par l'IA ? De plus, quelles sont les considérations techniques, conceptuelles et pédagogiques pour intégrer les LLMs dans les espaces VR afin de créer des expériences dynamiques, réactives et personnalisées pour les étudiants ?

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GENERAL INTRODUCTION

The rapid advancement of digital technologies has revolutionized the way educational institutions engage with students. Among these innovations, Virtual Reality (VR) and Artificial Intelligence (AI) stand out as transformative tools capable of creating immersive, interactive, and personalized experiences. Universities worldwide have begun adopting virtual campus tours [1, 2, 3] to showcase their facilities, attract prospective students, and enhance remote learning. However, despite their visual appeal, these virtual environments often lack dynamic interactivity, adaptability, and meaningful user engagement.

A critical limitation of existing virtual campus solutions is their reliance on prescripted interactions and static pathways, which restrict users to passive exploration. While 3D modeling and VR platforms like Unity and Unreal Engine have enabled highly realistic representations of physical spaces, the integration of intelligent, conversational agents remains underdeveloped. This gap presents an opportunity to leverage Large Language Models (LLMs), such as GPT and LLaMA, to create AI-driven virtual assistants that can engage users in natural, context-aware dialogues within VR environments.

This dissertation addresses these challenges by proposing and implementing an AI-powered virtual campus for the University of May 8, 1945, Guelma. The project combines 3D modeling, VR immersion, and LLM-based dialogue systems to deliver a smart, interactive, and personalized exploration experience. Key objectives include :

— Designing a high-fidelity 3D model of the university campus using Blender and Unity.

— Integrating LLM-driven NPCs (Non-Player Characters) to serve as virtual guides, capable of answering questions and providing real-time assistance.

— Developing intuitive navigation systems, including pathfinding and voice-controlled interactions, to enhance usability.

— Ensuring performance optimization for seamless VR rendering while maintaining visual realism.

By bridging the gap between immersive VR and conversational AI, this work aims to redefine virtual campus experiences, making them more engaging, adaptive, and educationally impactful. The findings and methodologies presented here contribute to the broader discourse on AI-enhanced virtual learning environments, offering a replicable framework for future innovations in educational technology.

0.1 Problematic

While several universities have begun exploring virtual reality (VR) to create immersive campus tours or learning environments, the majority of these initiatives remain limited in two fundamental ways. First, they predominantly rely on pre-scripted Non-Player Characters (NPCs) that follow fixed dialogue trees and lack the ability to engage in natural, adaptive conversations. This static interaction model fails to utilize the capabilities of modern Artificial Intelligence, especially Large Language Models (LLMs), which could enable NPCs to function as intelligent virtual teachers, guides, and assistants capable of understanding context, remembering past interactions, and providing dynamic responses tailored to each user.

Secondly, the visual quality of the virtual environments presented in many of these projects is often subpar. Models are frequently low in detail, poorly textured, or lack realistic lighting—factors which significantly reduce immersion and engagement, particularly in educational contexts where clarity and realism are important for spatial awareness and user experience. As a result, these virtual campuses often appear more like simplistic game prototypes than fully-fledged educational platforms.

This project seeks to address both of these shortcomings by building a highly detailed and optimized 3D representation of the university campus and integrating AI-powered NPCs capable of real-time dialogue and pedagogical assistance. Through the combination of high-fidelity visuals and LLM-driven interaction, the proposed system aims to deliver a more engaging, intelligent, and immersive experience that goes beyond traditional virtual tours.

CHAPITRE 1
VIRTUAL REALITY AND ARTIFICIAL INTELLIGENCE IN EDUCATION

1.1 Introduction

The integration of Virtual Reality (VR) and Artificial Intelligence (AI) is reshaping the landscape of digital interaction, particularly in educational environments. As immersive technologies mature, they offer transformative possibilities for how students explore, interact with, and learn from virtual spaces. This chapter delves into the foundational technologies that power AI-driven VR systems, focusing on how these tools can simulate realistic campus environments and enable intelligent, human-like communication through Large Language Models (LLMs).

We begin by defining Virtual Reality and outlining its key characteristics, especially in the context of immersive education. Then, we explore the primary platforms for VR development, including Unity and Unreal Engine, highlighting their respective strengths. Following this, we examine the role of 3D modeling and simulation in constructing virtual environments, emphasizing the technologies and techniques used to ensure realism and performance.

The chapter also investigates how Artificial Intelligence, particularly LLMs, enhances virtual environments through realism, interactivity, and personalization. This includes a breakdown of dialogue systems, NLP mechanisms, and deployment models such as API-based versus on-device and how they impact performance, privacy, and responsiveness. Finally, we address the design and ethical considerations of human-AI interaction within VR, setting the stage for building intelligent, believable, and inclusive virtual experiences.

1.2 Virtual Reality (VR)

Virtual Reality (VR) refers to a computer-generated simulation that enables users to interact with a three-dimensional environment using special electronic equipment such as headsets and motion controllers, or only the computer ans the mouse. Unlike traditional interfaces, VR places the user inside an experience, allowing for full immersion. The goal is to simulate a real or imagined world where users can look around, move, and interact as if they were physically present in that environment [4].

1.2.1 Characteristics of VR

The characteristics of Virtual Reality (VR) in education highlight its transformative potential for enhancing learning experiences. VR provides an immersive learning environment, fostering student engagement and interaction with complex subjects. By offering 3D visualizations, it improves understanding of intricate structures and interactions that traditional methods struggle to convey. The integration of gamification motivates students to explore and learn at their own pace, while interactivity allows for hands-on manipulation of virtual models, deepening comprehension. Accessibility through smartphones and low-cost headsets makes VR flexible and available outside the classroom. Additionally, VR can be adapted to meet diverse learning needs, with high acceptance rates among students who recognize its benefits. Research shows that VR enhances memory retention and understanding [5, 6] compared to traditional approaches. Overall, the potential for VR in education is significant, yet further studies are essential to optimize its application across various educational contexts and ensure it meets evolving student needs [7] .

1.2.2 VR Platforms : Unity and Unreal Engine

There are several platforms used for developing VR experiences, with Unity and Unreal Engine being the most popular.

Unity

Unity is a widely-used VR development platform known for its flexibility and user-friendliness. It supports a variety of VR SDKs such as Oculus SDK and OpenXR, and it's compatible with many VR headsets. Unity uses the C# programming language and features a large asset store that allows developers to integrate pre-made 3D models, animations, and scripts. Due to its strong community support and extensive documentation, Unity is a preferred platform for both beginners and professionals [8].

Unreal Engine

Unreal Engine is renowned for its high-end graphics and is often used in the development of AAA games and photorealistic VR applications. It supports both C++ programming and a visual scripting system called Blueprints, which allows developers to create complex logic without writing code. Unreal Engine is especially powerful for applications that require cinematic visuals, such as architectural visualization or medical simulations [9].

FIGURE 1.1 – Unity And Unreal Engine Logos

1.2.3 Educational Uses of VR

The educational applications of Virtual Reality (VR) present significant opportunities for enhancing learning across various fields. VR technology allows for the creation of immersive environments where learners can actively engage with content in ways that traditional methods cannot achieve. By simulating realistic experiences, VR promotes deeper understanding and retention of knowledge. It facilitates exploration and interaction, enabling students to manipulate virtual objects and environments, thus enhancing their problem-solving skills.

Moreover, VR can be particularly beneficial in remote or underserved areas, providing access to educational resources and experiences that might otherwise be unavailable. The ability to customize VR applications for specific educational contexts ensures that they meet the unique needs of diverse learners. As educational institutions increasingly recognize the value of VR, it holds the potential to transform teaching and learning, making education more engaging, interactive, and accessible for all [10].

1.3 3D Modeling and Simulation

3D modeling is the process of creating a three-dimensional representation of any object or surface using specialized software. This process is fundamental in building virtual environments for VR applications. In education-focused VR projects, 3D models help represent buildings, classrooms, objects, and characters that students can interact with. The creation of a 3D model usually begins with basic shapes (called primitives) such as cubes, spheres, and cylinders, which are then manipulated and refined to form complex structures. This process involves several key concepts :

— Vertices, Edges, and Faces : These are the core building blocks of a 3D mesh. Vertices define points in space, edges connect those points, and faces form the visible surfaces of the model.

— Modeling Techniques : Includes box modeling, extrusion, and sculpting, depending on the complexity and desired detail of the object.

— Texturing and Materials : After the shape is created, materials and textures are applied to give the model color, patterns, and realistic surface properties.

— Lighting and Environment : Scene creation also includes the placement of virtual lights and environmental settings to make the space feel real and immersive.

— Optimization : Essential for VR performance ; it ensures that models have the lowest polygon count possible without sacrificing visual quality [11].

.

1.4 Artificial Intelligence in Virtual Environments

Artificial Intelligence (AI) plays a crucial role in virtual environments by enhancing the realism, interactivity, and overall user experience. AI agents, whether they represent characters, systems, or environmental elements, contribute significantly to making virtual worlds feel dynamic and responsive rather than static and predictable [12, 13].

1.4.1 Enhancing Realism

AI-driven agents can simulate realistic behaviors that mirror how real-world entities would act. For example, non-player characters (NPCs) in a virtual university campus could autonomously navigate spaces, engage in natural conversations, or react logically to user interactions. These behaviors are often powered by decision trees, machine learning models, or reinforcement learning techniques that allow agents to adapt and respond in complex ways.

1.4.2 Increasing Interactivity

AI agents offer users meaningful interactions beyond scripted responses. Through techniques like natural language processing, computer vision, and behavior modeling, AI can interpret user actions, understand voice commands, and modify the environment dynamically. This responsiveness helps maintain immersion, keeping users engaged and making the virtual world feel alive.

1.4.3 Personalization and Adaptation

Advanced AI systems can personalize experiences based on user behavior. In educational VR platforms, AI agents could adapt explanations, guidance, or challenges to match a learner's level and preferences, offering a more tailored and effective experience.

1.4.4 Examples

- Virtual tour guides that answer students' questions about university departments in real time.
- AI-driven students and professors who simulate academic life and help new users understand university culture.

— Intelligent environment management systems that modify weather, lighting, or events in response to user activities.

In summary, AI agents are fundamental to building believable and engaging virtual environments, particularly in educational applications where interaction quality is critical for learning and exploration.

1.5 Large Language Models (LLMs)

1.5.1 Overview

Large Language Models (LLMs) are advanced artificial intelligence systems designed to understand and generate human language. They are trained on massive datasets containing diverse text sources, allowing them to predict and produce coherent, contextually relevant text. Popular examples of LLMs include OpenAI's GPT series (such as GPT-3.5 and GPT-4), Meta's LLaMA models [14] (Large Language Model Meta AI), and Mistral models [15], which focus on efficiency and performance for open-weight deployment. These models use deep learning techniques, particularly transformer architectures, to process and generate language with a high degree of fluency and contextual understanding. Their capabilities include answering questions, writing essays, generating dialogue, summarizing information, and even reasoning across complex topics [16].

1.5.2 Mechanisms of LLM-based Dialogue Systems

Dialogue systems based on LLMs operate by predicting the next most appropriate token (word or character) in a conversation based on the previous input. They use attention mechanisms to weigh the importance of different parts of the conversation history, allowing them to maintain coherence across long dialogues. Fine-tuning and prompt engineering techniques can guide these models to specialize in specific tasks, such as customer support, tutoring, or conversational companions. In more advanced systems, LLMs can be integrated with memory modules to remember past interactions, personalization layers to adapt to the user's style, and retrieval-augmented generation (RAG) models to access external databases for factual accuracy [17].

1.5.3 API-Based vs. On-Device Deployments

API-Based Deployment

In this model, the LLM runs on a remote server (usually hosted by the model provider, such as OpenAI or Anthropic. Applications interact with the model via APIs (Application Programming Interfaces) over the internet. This approach allows access to powerful models without needing substantial local computing resources, but it depends on network availability and raises considerations around data privacy, latency, and cost [18].

On-Device Deployment

Here, the model runs directly on the user's local hardware, such as a personal computer, VR headset, or smartphone. This method ensures greater data privacy, reduces reliance on external servers, and can minimize latency. However, on-device deployment is limited by the device's computational power, which often restricts the size and complexity of the models that can be run locally. Recent advancements, like quantization and model distillation, are helping make LLMs smaller and more efficient for on-device use. In virtual reality environments, the choice between API-based and on-device deployment can significantly impact user experience, particularly regarding interaction speed, offline availability, and security.

1.6 Natural Language Processing (NLP) for Dialogue

Natural Language Processing (NLP) is essential in enabling AI systems to understand and respond to human language in a meaningful way. In virtual environments, particularly those used for educational purposes, NLP is crucial for creating interactive dialogue systems that can communicate effectively with users [19].

1.6.1 Intent Recognition, Contextualization, and Response Generation

Intent Recognition

Intent recognition involves identifying the purpose behind a user's input, such as a question, request, or command. In a virtual university environment, this could mean recognizing when a user is asking for directions, inquiring about a course, or looking for help with a task. The goal is to classify the input into predefined categories (e.g., asking for help, providing feedback, initiating a task).

Contextualization

Context is key for understanding the meaning of a user's input. NLP systems need to consider the context of previous interactions, the user's current goals, and the ongoing conversation to generate relevant and coherent responses. This allows for a more fluid and natural conversation. For instance, if a user previously asked about a specific department, the system could provide more detailed follow-up information in subsequent exchanges.

Response Generation

Once intent is recognized and context is established, the system must generate a response that is both informative and engaging. Modern dialogue systems, particularly those powered by LLMs, use deep learning techniques to generate human-like responses. These responses can be tailored to the user's language style, level of understanding, and previous interactions, making the AI appear more empathetic and knowledgeable.

1.7 User Engagement Strategies : Gamification, UI Design, and Accessibility

Maintaining user engagement in VR tours requires thoughtful design strategies that balance functionality and immersion. Gamification has proven effective in educational VR [20, 21], where elements like interactive quizzes, scavenger hunts, or achievement systems encourage exploration and retention. For example, a virtual campus tour might reward users for visiting key locations or completing mini-tasks, transforming passive observation into an active learning experience.

Equally important is intuitive UI design, which minimizes distractions while ensuring ease of navigation. Many successful VR tours employ diegetic interfaces UI elements that exist naturally within the virtual world, such as floating signs or in-world maps to reduce cognitive load. Additionally, accessibility considerations, such as adjustable text sizes, voice narration, and alternative input methods (e.g., gaze-based selection), ensure inclusivity for diverse users. Despite these advancements, some projects still suffer from clunky menus or over-reliance on controller inputs, which can disrupt immersion.

1.8 Clunky Interfaces and Lack of Personalized Guidance

While VR tours have grown increasingly sophisticated, several persistent challenges hinder their effectiveness. Many systems still rely on unintuitive interfaces, such as complex button combinations or unnatural menu navigation, which can frustrate users unfamiliar with VR controls. Furthermore, most existing tours offer generic, one-size-fits-all guidance, lacking adaptive features that tailor content to individual interests or learning paces. For instance, a virtual campus tour might provide the same scripted narration to all users, missing opportunities to highlight areas relevant to a student's specific field of study.

1.9 Human-AI Interaction in VR : Challenges and Design Considerations

Creating effective human-AI interactions in virtual reality (VR) environments presents unique challenges that differ from traditional applications. The immersive nature of VR amplifies the need for intuitive, seamless interactions between users and AI agents.

1.9.1 Realism and Believability

For AI agents in VR to be effective, they must behave in ways that are realistic and believable. Users expect AI agents to have lifelike movements, natural language communication, and logical behavior that fits within the virtual world. Any unnatural or inconsistent behavior can break immersion, which is crucial for VR experiences.

1.9.2 User Experience Design

The design of AI interactions should consider the user's physical and emotional state within the VR environment. For example, VR users may experience motion sickness, which could be triggered by overly complex or rapid movements of AI agents. Designers must account for such factors and ensure that AI responses and actions are comfortable and intuitive for the user.

1.9.3 Natural and Non-Intrusive Interactions

In VR, AI agents should interact with users in a non-intrusive manner. Overly frequent or demanding interactions can disrupt the user's experience. AI agents must be able to gauge when and how to engage the user, ensuring the interaction is timely and relevant to the task at hand. Furthermore, users should have control over the AI's involvement, such as the ability to pause or end interactions if needed.

1.9.4 Ethical Considerations

As AI agents become more capable of simulating human-like interactions, ethical considerations also come into play [22]. In educational VR, AI agents should respect privacy, avoid bias, and ensure fairness in their responses. Designers must be cautious in how data is collected and used, as well as ensure that AI agents are designed to act in a way that aligns with ethical guidelines, especially when interacting with minors.

1.10 Conclusion

In this chapter we have established the technological and conceptual groundwork for creating intelligent, immersive virtual environments. We have explored how VR platforms like Unity, supported by tools such as Blender and Materialize, provide the backbone for realistic 3D simulations. Coupled with AI techniques particularly LLMs and NLP these environments are no longer limited to passive exploration but can support dynamic, personalized, and context-aware interactions.

Furthermore, we have addressed the critical aspects of deploying LLMs, the challenges of natural language dialogue in VR, and the ethical responsibilities tied to building AI-powered agents. This comprehensive understanding paves the way for the following chapters, where these technologies are applied to build an interactive virtual campus, bridging the gap between technological capability and educational utility.

CHAPITRE 2
REVIEW OF EXISTING VR CAMPUS SOLUTIONS AND AI INTEGRATION GAPS

2.1 Introduction

The rapid development of immersive technologies has led to the emergence of various virtual campus applications across academic institutions. However, while these implementations demonstrate the potential of VR in education, they often fall short in terms of interactivity, personalization, and adaptive communication. This chapter provides a comprehensive review of existing VR campus systems, focusing on their features, interaction methods, and levels of AI integration.

We begin by surveying current VR use cases in universities such as the University of West London, Stanford, and Illinois, analyzing their strengths and limitations. A comparative analysis highlights the common reliance on scripted interactions and static content, which limits user engagement and flexibility. From there, we explore the evolution of AI-driven non-player characters (NPCs), assessing prior work in dialogue systems and the growing but still underutilized role of Large Language Models in educational VR.

The chapter also identifies critical technical and pedagogical gaps in current systems including issues with clunky interfaces, lack of real-time personalization, and latency in LLM responses. By mapping out the challenges and shortcomings of existing approaches, this chapter establishes a clear justification for a new, more dynamic framework that integrates advanced AI within VR to transform the user experience.

2.2 Existing VR Campus Implementations

Several universities have developed VR environments for specialized training, with the University of West London (UWL) emerging as a notable example in healthcare education. UWL's VR platform trains nursing students through immersive medical simulations, allowing them to practice procedures, interact with virtual patients, and respond to dynamic clinical scenarios. Unlike traditional methods, UWL's system emphasizes real-time decision-making for instance, students must assess patient conditions, administer correct dosages, and communicate effectively all within a controlled virtual space.

Other institutions have taken different approaches :

— **University of Illinois :** Developed VR medical labs primarily for surgical training, leveraging haptic feedback for precision but lacking AI-driven patient interaction [23].

— **Stanford University :** Focused on collaborative virtual labs for STEM fields, enabling remote teamwork but without adaptive NPCs or dialogue systems [24].

While these implementations demonstrate VR's educational potential, most rely on pre-scripted scenarios rather than AI-generated content a limitation my project seeks to address.

2.3 Comparative Analysis : Features, Interaction Methods, and AI Integration

When examining prior VR campus projects, key differences emerge in design priorities and technological depth :

Features

- **UWL :** Prioritizes healthcare realism (e.g., accurate medical equipment, patient vitals monitoring).
- **Illinois/Stanford :** Focus on technical skill-building (surgery, lab experiments) or collaboration tools.

Interaction Methods

- **UWL :** Employs voice commands and hand-tracking for natural nurse-patient interactions.
- **Others :** Use controller-based inputs (e.g., Stanford's virtual whiteboards).

AI Integration

Most projects, including UWL's, use scripted NPCs with fixed dialogue. None fully leverage LLMs for dynamic patient responses creating an opportunity for my work to innovate.

2.4 Critical Gap

Current systems lack personalized, AI-driven adaptability (e.g., an LLM-powered patient who recalls prior interactions or adjusts symptoms based on student decisions). my thesis could bridge this by integrating real-time NLP into UWL's framework.

2.5 AI-Driven NPCs in Virtual Environments

2.5.1 Prior Work on AI Avatars

Early virtual environments relied on scripted NPCs (Non-Player Characters) with pre-defined dialogue trees and behaviors, limiting interactivity. Recent advancements have introduced AI-driven avatars, enhancing realism through machine learning and natural language processing. Examples include :

- **Rule-Based NPCs :** Used in medical simulations (e.g., UWL's nursing VR) for structured interactions.
- **Chatbot-Style Avatars :** Projects like Virtual Humans employed early NLP for basic conversation.
- **LLM-Powered Characters :** Modern experiments (e.g., AI Dungeon, Meta's Codec Avatars) integrate GPT-like models for dynamic responses.

However, most educational VR projects still use static NPCs, missing opportunities for adaptive learning.

2.5.2 Dialogue Systems in VR

Current approaches fall into two categories :

Rule-Based Systems

- Fixed dialogue options (e.g., multiple-choice responses).

— Used in VR therapy (e.g., treating social anxiety) due to predictability.

Generative AI Systems

— Leverage LLMs (e.g., OpenAI, Llama) for open-ended conversations.

— Rare in education due to latency and lack of control over outputs.

VR-Specific Challenges :

— **Voice Recognition :** Background noise and accent variability reduce accuracy.

— **Context Retention :** LLMs often "forget" prior interactions in long sessions.

2.5.3 Limitations

Performance Issues

— Cloud-based LLMs introduce latency, breaking immersion.

— On-device models (e.g., Llama.cpp) sacrifice response quality for speed.

Ethical Concerns

— needs in training data may lead to inappropriate NPC behavior.

— Lack of content moderation in real-time generative dialogue.

Design Trade-offs

— **Realism vs. Control :** Dynamic NPCs are engaging but harder to align with educational goals.

— **User Fatigue :** Overly verbose LLM responses can overwhelm learners.

Research Gap : Few studies test LLM-driven NPCs in VR education our Work could pioneer this by balancing adaptability with pedagogical safety.

2.6 Navigation and Interaction in VR Tours

2.6.1 Pathfinding Techniques : NavMesh

Effective navigation is essential for creating seamless virtual environments, particularly in educational and exploratory VR applications. The most common approach, NavMesh (Navigation Mesh), is widely used due to its efficiency in static environments. NavMesh predefines walkable areas within a 3D space, allowing characters and users to follow optimized routes without computational overhead. This method is particularly popular in virtual campus tours and museum simulations, where the layout remains consistent. However, NavMesh lacks adaptability when faced with dynamic obstacles, such as moving objects or user-altered paths, limiting its use in more interactive scenarios [25].

Another notable limitation is the absence of intelligent NPC guides. While some projects incorporate basic avatars to direct users, these characters typically follow rigid scripts, unable to answer nuanced questions or adjust their guidance based on user behavior. Integrating AI-driven assistants, capable of dynamic dialogue and personalized recommendations, could address this gap, bridging the divide between static tours and truly interactive experiences.

2.7 Current Implementations of LLM-VR Integration

Recent advances in large language models (LLMs) have enabled dynamic AI interactions in virtual reality, with several pioneering projects demonstrating this potential. VR experiences now utilize models like OpenAI's GPT and Meta's LLaMA to power conversational NPCs capable of real-time responsive dialogue. Educational applications show particular promise, with virtual tutors in classroom environments adapting explanations to student needs. Narrative-driven experiences have also emerged, such as interactive storytelling companions that generate plot branches dynamically. These implementations reveal both the transformative possibilities of

LLM integration and the current limitations in achieving seamless immersion. Notably, projects like AI-driven virtual patients for medical training and language learning assistants highlight the technology's potential to revolutionize specialized education domains.

2.7.1 Technical Challenges in LLM-VR Deployment

Significant hurdles remain in effectively implementing LLMs within VR systems. Latency presents a critical barrier, as even minor delays in cloud-based API responses (typically 200–1500 ms) can disrupt VR's essential sense of presence. Developers have attempted mitigation through on-device deployments of optimized models, though these solutions currently force difficult trade-offs between conversational depth (limited by smaller model sizes) and system responsiveness.

The computational demands of maintaining both linguistic sophistication and real-time performance create ongoing design challenges, particularly for applications requiring nuanced dialogue. Additional technical obstacles include the lack of standardized frameworks for natural language interaction in 3D environments and insufficient research into hybrid architectures that might optimally balance cloud and local processing.

These challenges are further compounded in educational applications, where requirements for accuracy and pedagogical effectiveness introduce additional constraints on system design.

2.8 Comparative Analysis of VR Campus Implementations and the Role of AI-Driven Interaction

The following table summarizes the key features, strengths, and limitations of VR campus projects at three universities, while all three projects demonstrate VR's potential in education, they rely heavily on pre-defined scripts, static interfaces, and lack

TABLE 2.1 – Comparison of VR Campus Projects at Three Universities

University	Use Case	Strengths	Limitations
UWL	Immersive medical simulations for training nursing students	— Real-time decision-making scenarios — Use of voice and hand-tracking	— Pre-scripted NPCs. — No dynamic dialogue or memory.
Illinois	VR medical labs focused on surgical training with haptic feedback	— High precision skill development — Realistic procedural environments	— No intelligent interaction with virtual patients — No LLM or adaptive AI integration
Stanford	Collaborative virtual labs for STEM education	— Supports teamwork in shared virtual space	— Lacks AI characters — Limited interaction with static tools

natural, human-like interactions. These limitations point to a clear opportunity for improvement through AI-driven solutions.

2.8.1 Proposed Solution in This Project :

- Integration of Large Language Models (LLMs) to enable dynamic, real-time dialogue with NPCs.
- Personalization of responses based on user behavior and context.
- Enhanced immersion through voice-based interaction and lifelike animations.
- Bridges the gap between static VR tours and truly intelligent, adaptive learning experiences.

2.9 Conclusion

This chapter has revealed that while current VR campus implementations provide immersive educational environments, they are largely limited by static design and constrained AI capabilities. Most platforms rely on pre-defined scripts and offer little adaptability, resulting in generic, non-personalized experiences. Through a comparative review, we identified a significant opportunity for innovation in the integration of LLMs and real-time AI dialogue within VR systems.

The analysis underscores a clear gap : the need for intelligent NPCs that can adapt, remember, and respond contextually to users in an educational setting. Addressing this gap through LLM-driven interaction opens new possibilities for engagement, learning, and guidance within virtual campuses. These insights directly inform the next stage of the project the methodology for building a dynamic and interactive AI-powered virtual campus that offers a more authentic and personalized experience for students.

CHAPITRE 3 METHODOLOGY FOR A SMART INTERACTIVE VIRTUAL CAMPUS OF 8 MAI 1945 UNIVERSITY

3.1 Introduction

This chapter presents the complete development methodology used to design and implement the AI-powered virtual campus for 8 Mai 1945 University. The goal was to construct a visually immersive, interactive, and optimized 3D environment that allows users to navigate the campus and engage in natural conversations with intelligent virtual characters. The methodology combines elements of 3D modeling, game development, user interface design, AI integration, and optimization.

Beginning with the collection of visual references and satellite imagery, the workflow proceeds through stages such as blockout, detailed modeling in Blender, texturing with tools like Photoshop and Materialize, and importing into Unity. The chapter also describes the use of Unity's High Definition Render Pipeline (HDRP) to achieve high-fidelity graphics and performance. Special attention is given to key interactive components, including the third-person controller, NPC development, voice-based interaction system, and LLM server integration.

By outlining the technical steps and design choices in detail, this chapter serves as the practical foundation of the project, demonstrating how theoretical concepts are transformed into a functional and immersive educational VR experience.

3.2 Methodology

This section outlines the methodologies employed in the design and development of the interactive university environment for 8 Mai 1945 University, encompassing the creation of a 3D map showcase and a virtual tour for new students, both developed using the Unity game engine.

3.2.1 Map Design

The objective of the map design phase was to create a realistic, visually appealing, and highly optimized 3D representation of 8 Mai 1945 University for use as the primary map in the Unity-based showcase and virtual tour.

Objectives

— To develop a realistic and visually appealing 3D representation of 8 Mai 1945 University as the map.

— To ensure accurate depiction of key landmarks, buildings, and the overall campus layout.

— To optimize the 3D model for efficient performance within the Unity engine.

Requirements

— Utilization of 3D modeling software, specifically Blender, for map creation.

— Acquisition of high-quality textures, materials, and assets for buildings, terrain, and vegetation to enhance visual quality.

— Optimization of the 3D model's geometry and textures for optimal rendering performance in Unity.

— Adherence to the specified hardware and software requirements of the target platforms (e.g., PC, mobile).

— Inclusion of key features of 8 Mai 1945 University, such as buildings, roads, walkways, and prominent landmarks.

— Ensuring compatibility of the 3D map with the Unity game engine.

— Performing iterative testing and optimization to address any performance or visual issues during development.

Creating The 3D Map

The process of creating the 3D map involved a series of structured steps to ensure accuracy, visual fidelity, and optimal performance.

3.3 Reference Gathering

The initial phase involved extensively collecting maps, photographs, and other relevant references of 8 Mai 1945 University. These resources formed the foundation for creating an accurate and detailed 3D map. Meticulous examination of this material provided a comprehensive understanding of the university's layout, architectural intricacies, and unique features, enabling a faithful representation of key elements in the 3D model. This comprehensive reference gathering ensured the resulting 3D map captured the university's essence and authenticity, providing players with an immersive experience.

3.4 Blockout and Layout

To begin the 3D map creation, a preliminary blockout was developed using simple shapes and primitives in Blender. This established the overall campus layout, carefully considering the placement of roads, walkways, and buildings. Emphasis was

FIGURE 3.1 – satalite image of the university

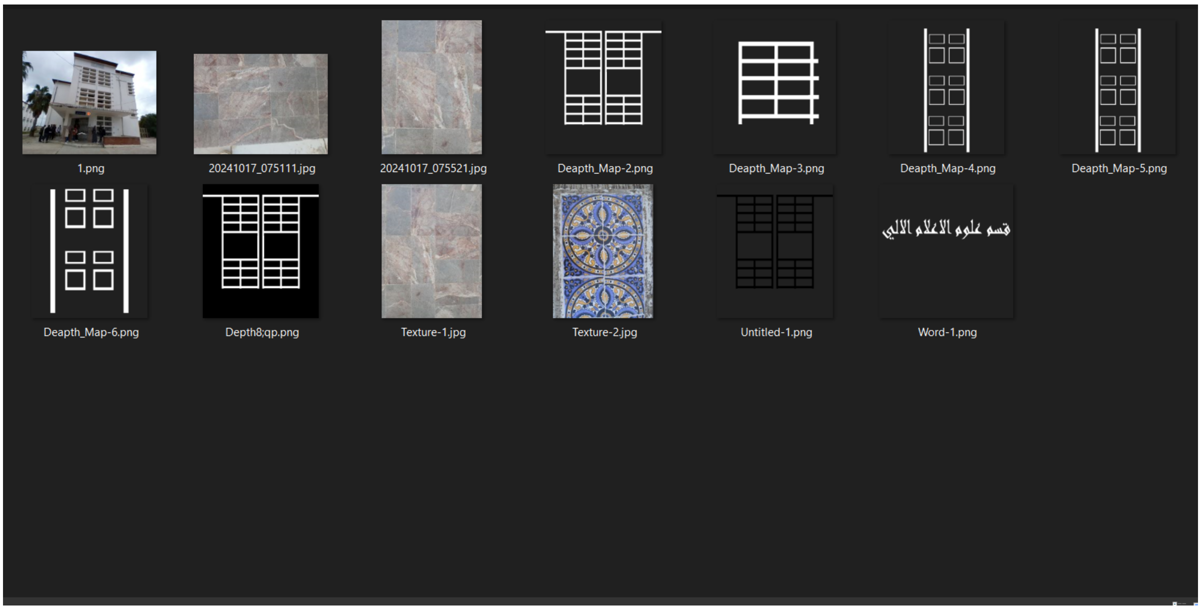

FIGURE 3.2 – some of the reference photos

placed on achieving precise proportions and spatial relationships between elements to accurately reflect the real-world layout, allowing for authentic and familiar navigation.

3.5 Building Modeling

Each building within the university map was meticulously modeled individually in Blender, with close reference to the collected source materials. This involved a careful examination of architectural details, including windows, doors, and other distinctive features, to faithfully replicate the real-world structures. This attention to detail contributed significantly to the overall realism and authenticity of the game environment.

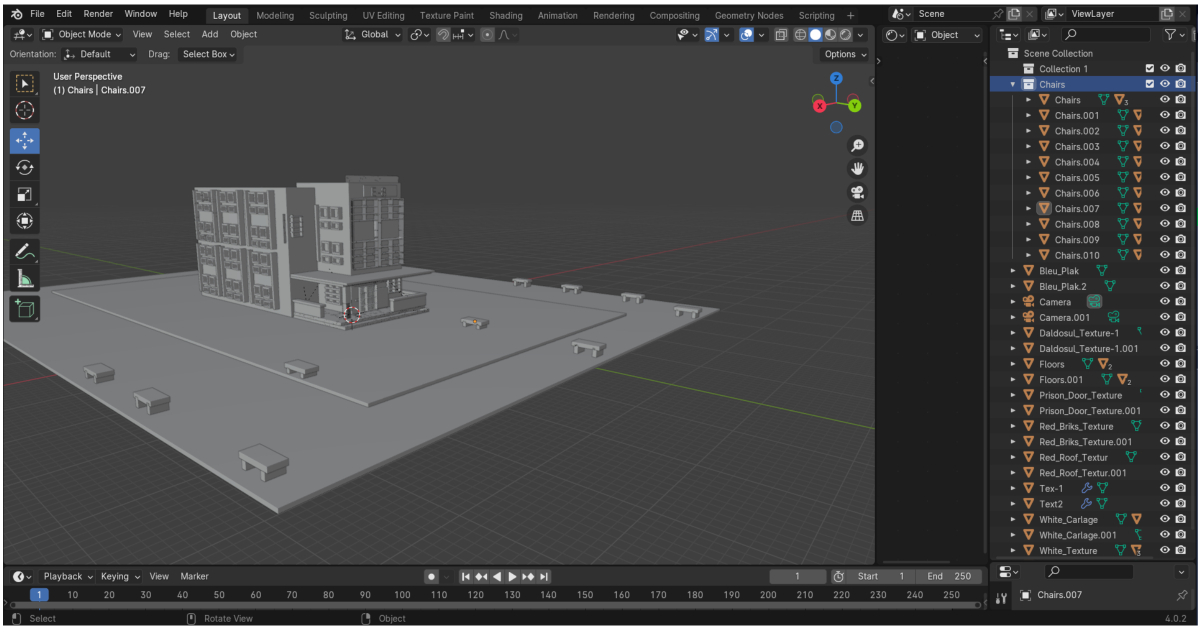

FIGURE 3.3 – 3d model of Computer Department

3.6 Texturing and Materials

To capture the authentic look and feel of the real-world university materials, a series of photographs were taken. These served as valuable references for accurately replicating textures in the 3D model, capturing color, texture, and surface details. This meticulous material reproduction added an extra layer of authenticity to the 3D map.

3.6.1 Technologies Used in The Project

Blender

is a free and powerful 3D creation tool. It's used for modeling, animation, rendering, and even video editing. Because it's open-source and supports the entire 3D pipeline, it's popular with both hobbyists and professional studios [26].

Photoshop

is the industry-standard software for image editing. Designers and artists use it for photo manipulation, digital painting, and preparing textures for 3D models [27].

Materialize

is a specialized tool for turning images into game-ready materials. It helps developers create realistic textures for 3D environments quickly and efficiently [28].

Visual Studio

is a powerful coding tool from Microsoft. Programmers use it to write, debug, and test software, including games, websites, and mobile apps. It supports many programming languages [29].

C# (C Sharp)

is a modern programming language often used with Unity. It's easy to learn, works well for game development, and is great for building Windows applications. Its automatic memory management and strong performance make it a top choice for developers [30].

FIGURE 3.4 – pictuers taken for real materials

Significant attention was also dedicated to preparing materials for seamless tiling across surfaces. This ensured materials could be repeated without visible seams or discontinuities, resulting in a cohesive and realistic appearance across the entire map.

To further enhance visual details and realism, Materialize software was employed. This tool facilitated the addition of intricate details using normal maps, height maps, and roughness maps. Normal maps simulated fine surface details and imperfections, adding depth ; height maps created subtle height variations ; and roughness maps

FIGURE 3.5 – seamless and not seamless texture

controlled glossiness or reflectivity. Leveraging Materialize imbued the 3D map's materials with enhanced realism and intricacy.

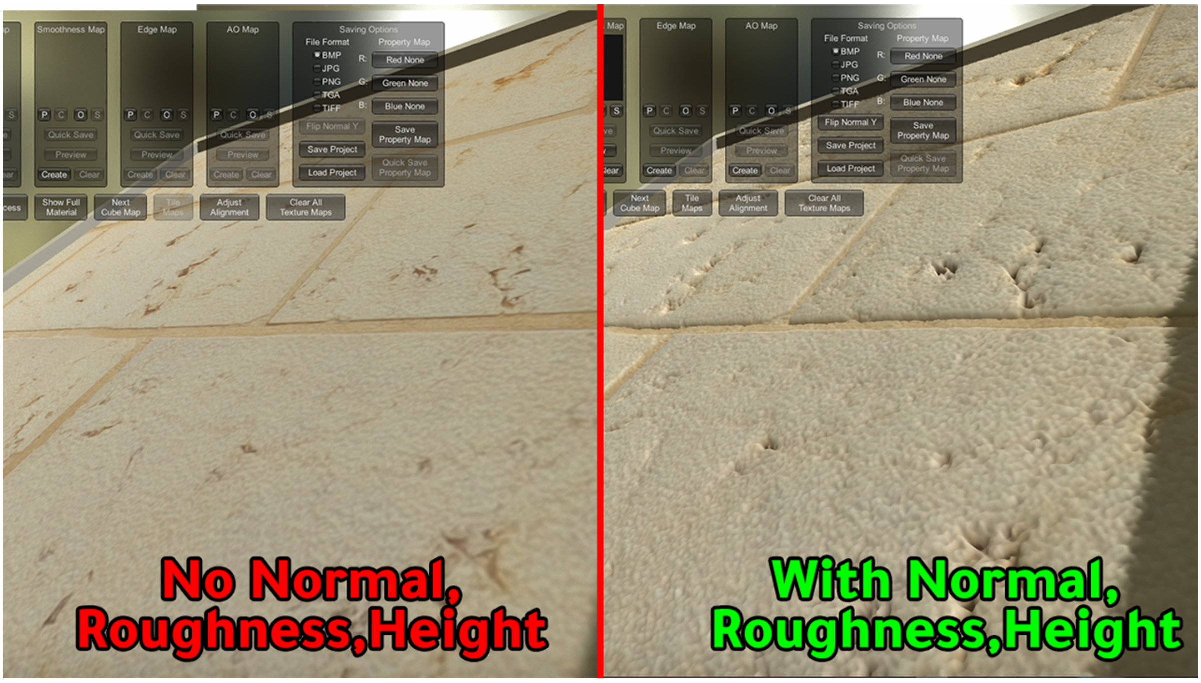

FIGURE 3.6 – adding details to the matireal

3.7 Optimization and Render Pipeline Selection

To ensure optimal performance and visual quality, the project leveraged Unity's High-Definition Render Pipeline (HDRP), chosen for its advanced rendering capabilities and optimization features. HDRP excels in delivering cinematic-quality graphics while maintaining efficiency through techniques like [31] :

- Distance-based culling : Automatically avoids rendering faraway objects, reducing GPU overhead.
- Dynamic resolution scaling : Adjusts render quality in real-time to preserve performance.
- Advanced lighting and shading : Utilizes physically based rendering (PBR) for realistic materials without excessive resource cost.

HDRP vs. URP : Key Differences

- Higher Visual Fidelity : HDRP supports ray tracing, volumetric effects, and complex post-processing, unlike URP, which prioritizes broader hardware compatibility.
- Optimized for High-End Hardware : HDRP balances quality and performance via GPU-driven occlusion culling and LOD systems.
- Scalability : HDRP's settings can be fine-tuned to prioritize either realism or performance.

Model Optimization The 3D assets were further optimized by :

- Polygon reduction : Streamlining geometry without sacrificing detail.
- Removing hidden surfaces : Eliminating unseen portions of buildings to minimize unnecessary rendering.

This combination of HDRP's built-in optimizations and manual asset refinement ensured a seamless, visually immersive experience tailored to high-end platforms.

ar design for scalability.

FIGURE 3.7 – The Defrence Between HDRP and URP

3.8 Unity Showcase

The Unity showcase leverages Unity engine to present a highly detailed 3D model map of 8 Mai 1945 University. The primary goal is to demonstrate Unity's graphical capabilities by presenting the map in a visually immersive and realistic environment. By utilizing Unity's advanced rendering features, such as High Definition Render Pipeline (HDRP) and real-time lighting, the project highlights the map's intricate details and captivating design, providing players with an engaging experience.

3.9 Third-Person Controller

To improve player interaction and control, a third-person controller was integrated into the game. This enables seamless control of the player character from a third-person perspective, allowing players to navigate and engage with the virtual environment effectively. The addition of this controller ensures intuitive and responsive controls, creating a more immersive and dynamic gameplay experience.

FIGURE 3.8 – Third Person Controller

3.10 Navigation Menu and Pathfinding System

To enhance the accessibility and interactivity of the virtual tour for 8 Mai 1945 University, a navigation menu was integrated into the Unity-based project, allowing students to select a department and receive real-time directions to their chosen destination. The menu, designed using Unity's UI system, presents a clear and intuitive interface listing all campus departments, ensuring ease of use for new students. Upon selecting a department, Unity's NavMesh Agent calculates an optimal path from the player's current location to the target, using a pre-generated NavMesh that defines walkable areas like pathways while avoiding obstacles such as buildings. Visual path indicators, such as highlighted routes or waypoints, are rendered on the 3D map to guide users clearly. This system was optimized for performance within Unity's High-Definition Render Pipeline (HDRP), ensuring smooth navigation and compatibility across PC and mobile platforms. Iterative testing refined the pathfinding accuracy and menu responsiveness, creating an engaging and efficient tool for students to explore the campus.

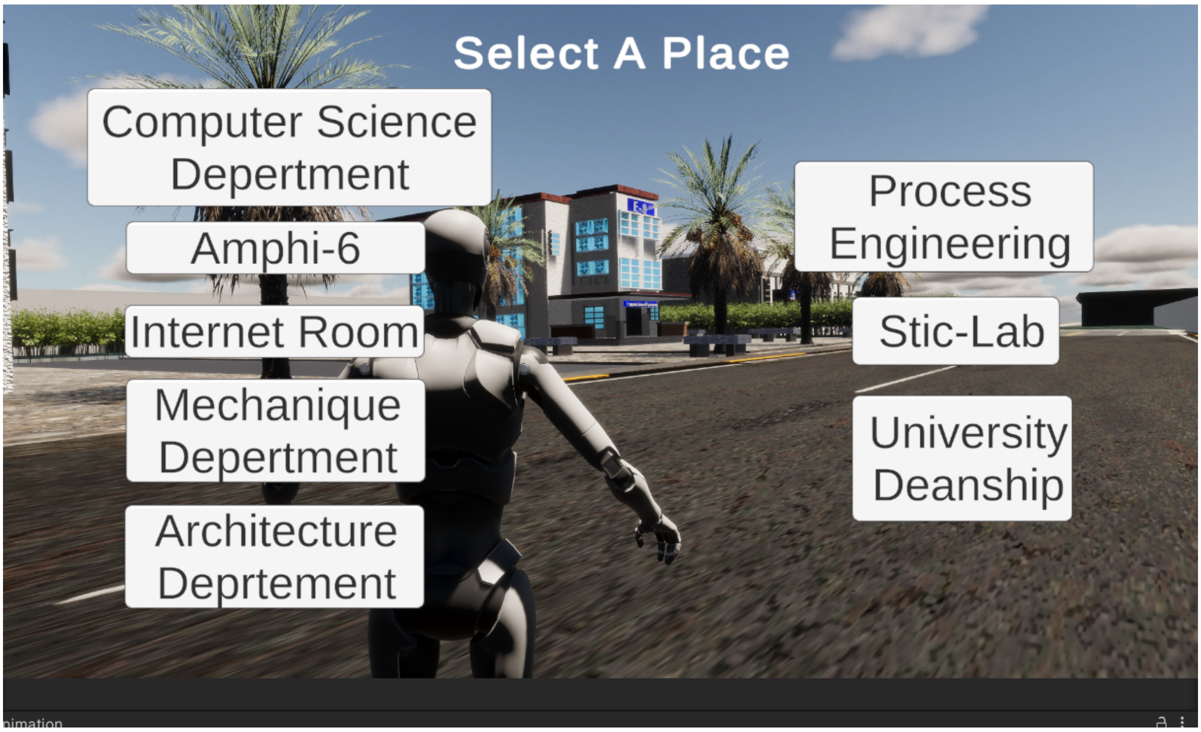

FIGURE 3.9 – destinations selection

3.11 NPC Character Creation

A 3D character model was designed to serve as the NPC, representing a virtual Anatomy Teeacher for the university . The character was crafted using Blender, with detailed attention to visual aesthetics to align with the realistic style of the 3D map. The model was rigged to support animations and imported into Unity, where it was configured with a defined role (e.g., campus guide) to interact meaningfully with users, enhancing the authenticity of the virtual tour.

3.12 LLM Integration via Server

a dynamic Non-Player Character (NPC) capable of natural language interaction within a virtual environment. The NPC was developed in Unity and connected to LM Studio, which served as a bridge to a large language model (LLM). The interaction is entirely text-based on the backend : we created a lightweight application that converts

FIGURE 3.10 – Npc Character

the user's voice into text, sends that text to the AI, receives a textual response, and then transforms that response into voice output. This enables the NPC to appear as if it's "talking." Additionally, we synchronized certain facial and body animations with the voice playback to make the NPC's behavior appear more lifelike and immersive.

3.13 Animation Tree and Dynamic Behavior

An animation tree was implemented in Unity to control the NPC's movements and expressions based on dialogue states. Using switch statements, the system dynamically selects animations (e.g., talking,

gesturing, or idle poses) in response to the LLM's output or user interactions. This ensures the NPC's visual behavior aligns with the conversation flow, creating a seamless and lifelike interaction that enhances the immersive quality of the virtual tour.

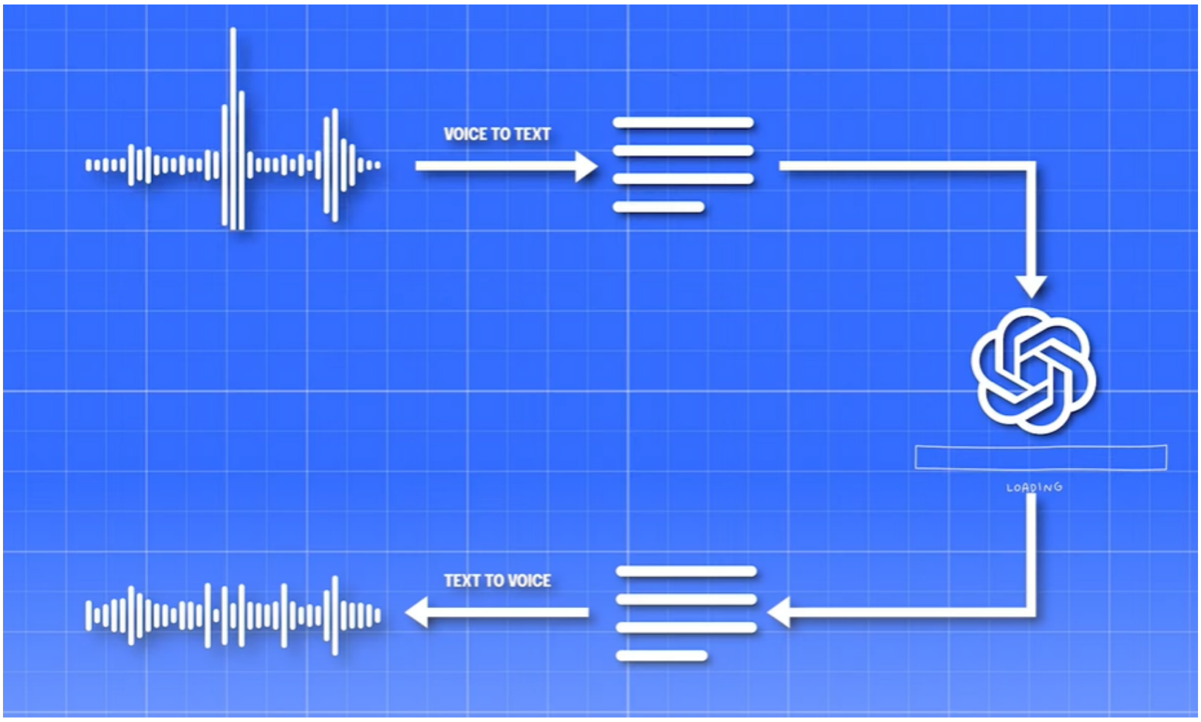

FIGURE 3.11 – How Ai Character Works

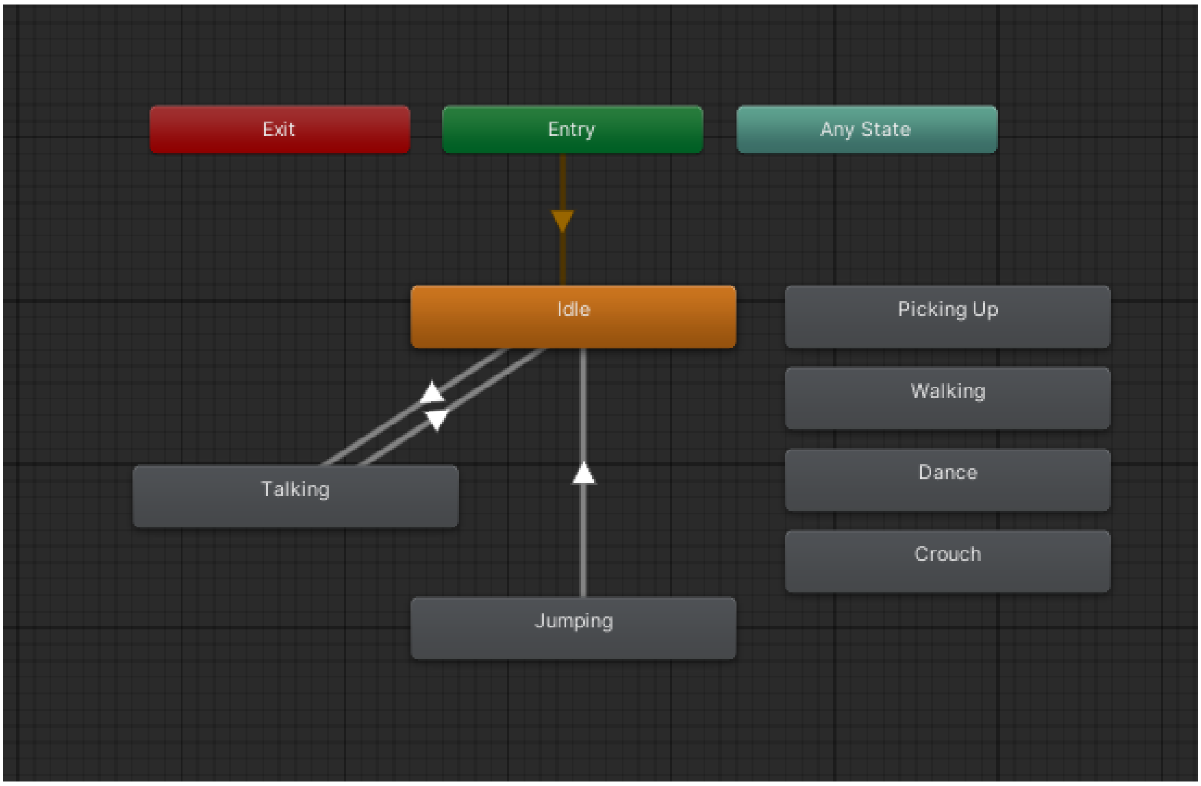

FIGURE 3.12 – Npc Animation Tree

3.14 sequence Diagram

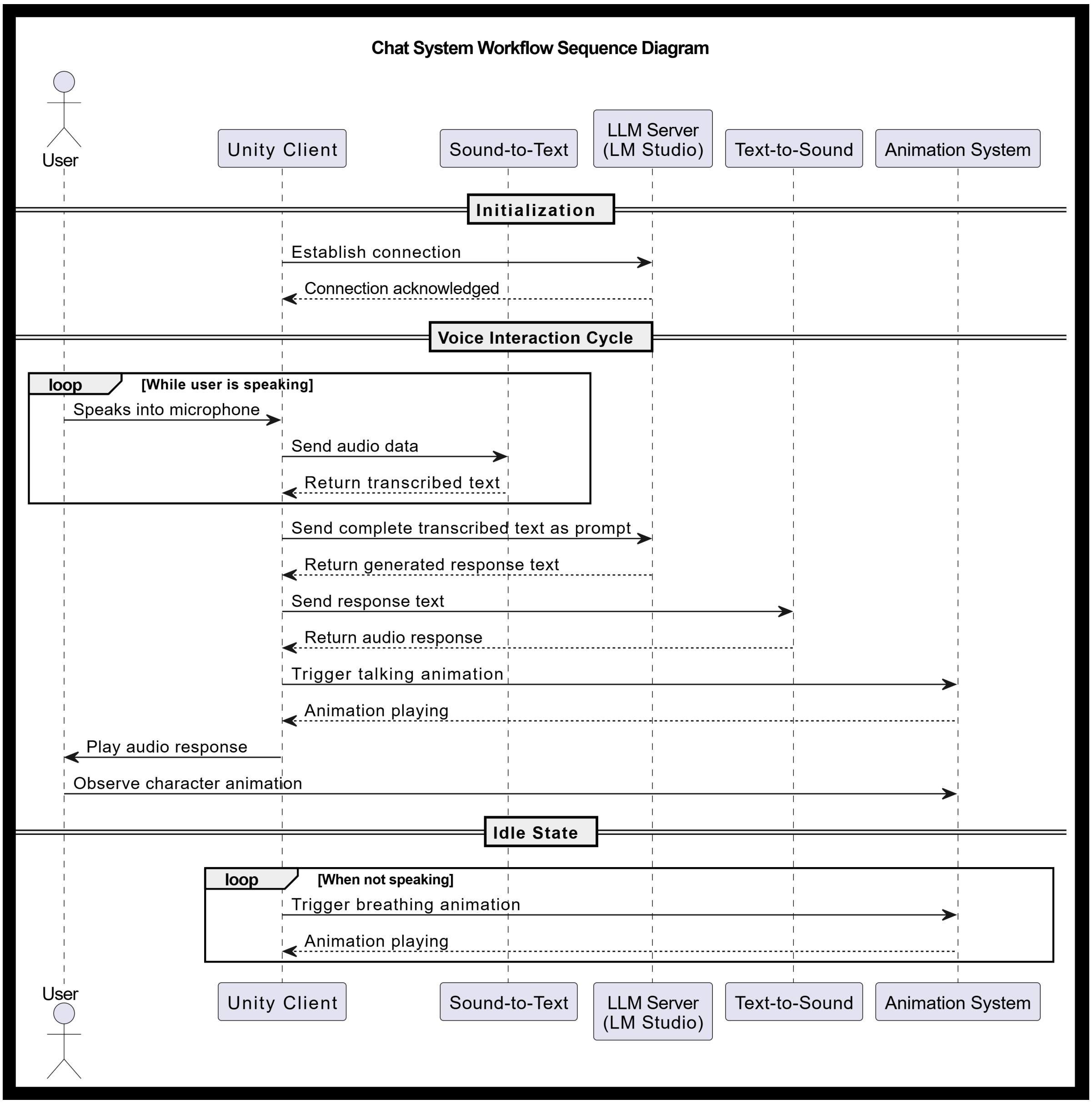

FIGURE 3.13 – Sequence Diagram

3.15 Explanation of the Sequence Diagram

The interaction flow between the user, the Unity environment, LM Studio, and the AI-driven NPC is designed to simulate a natural conversation within a virtual 3D campus. This interaction begins when the user initiates a prompt and ends when a spoken response is delivered by an animated character, creating a lifelike AI-driven experience. The system follows a structured sequence that ensures real-time communication, dynamic animation, and immersive feedback.

The process starts when the user presses the T key on the keyboard. This input acts as a trigger to establish a live connection between Unity and the locally hosted LM Studio, which runs a Large Language Model (LLM). As long as the user keeps holding the T key, the system remains active and listening. The user's voice is captured using a speech recognition (voice-to-text) application, which transcribes the spoken input into text format. This transcribed text represents the user's question or command.

Once the text prompt is ready, Unity sends it to LM Studio through a network call. LM Studio processes the prompt using the embedded LLM (such as LLaMA 3) and generates a textual response. This response is then sent back to Unity, where it is handled in multiple ways.

First, the text is forwarded to a text-to-speech (TTS) module, which synthesizes the content into an audio clip. This voice output is played through Unity's audio system, giving the impression that the AI character is speaking directly to the user. At the same time, a 3D animated NPC character—designed as a virtual teacher or guide—is triggered to perform specific animations, such as idle poses, facial gestures, or subtle body movements, to reinforce the illusion of a living, responsive being. These animations are controlled via Unity's animation tree, which dynamically selects appropriate states based on whether the character is "listening", "speaking", or "idle".

The interaction remains active until the user releases the T key. Once the key is released, Unity detects the input change and gracefully terminates the connection with LM Studio. This mechanism ensures that the system only listens and processes user inputs during deliberate interaction periods, conserving resources and preserving conversational context.

Through this seamless flow—initiated by a key press, driven by AI reasoning, enriched with audio feedback, and visualized by real-time 3D animation—the system delivers a rich, immersive, and intelligent virtual interaction. It marks a significant advancement over conventional pre-scripted NPCs by enabling adaptive, voice-controlled conversations within an educational VR setting.

3.16 Methods of Integrating AI Models into the Virtual Campus

Integrating AI into a virtual campus environment can be achieved through several architectural methods, each offering different trade-offs in terms of performance, cost, complexity, and scalability. In this section, we explore three main methods for deploying AI models : API-based integration, local (on-device) deployment, and hybrid approaches. Each of these methods has distinct advantages and disadvantages that must be considered depending on the educational goals and technical constraints of the system.

API-Based Integration

This method involves connecting Unity to an external AI service via an internet-based API. The user's prompt is sent to a remote server , which processes the request and returns a response from a hosted language model. Advantages :

Access to powerful, state-of-the-art models that are regularly updated and fine-tuned.

No need for heavy local computing resources.

Simple to implement and maintain, with well-documented SDKs and support.

Disadvantages :

Often subject to pricing models based on the number of tokens processed, which can become expensive over time.

Requires a stable internet connection, which can introduce latency and reduce responsiveness in real-time VR scenarios.

Raises concerns about data privacy and security, particularly in academic settings where sensitive information may be involved.

Local (On-Device) Deployment

With local deployment, the AI model is hosted directly on the user's machine. Tools like LM Studio can run open-source models (e.g., LLaMA, Mistral) without requiring cloud infrastructure.

Advantages : No ongoing token-based costs—after initial setup, usage is free. Better data privacy and control over the entire processing pipeline.

Lower latency compared to cloud services, enhancing immersion and real-time responsiveness.

Disadvantages : Requires significant hardware capabilities, particularly in terms of CPU, GPU, and RAM.

May not support very large or complex models due to device limitations.

Can be more complex to set up and optimize, especially for users without AI deployment experience. Convai offers a specialized solution for integrating conversational AI into virtual environments. It provides a ready-to-use platform that allows developers to assign pre-trained AI personalities to 3D characters in Unity, which can then engage in real-time conversations based on custom data or uploaded content.

In this project, Convai was used to train an AI agent on an 80-page academic chapter. The results were highly responsive, with average response times of approximately 0.8 seconds, even for complex or long queries. Convai also supports voice input and output, animation triggers, and memory capabilities, making it ideal for rich NPC-based educational applications.

Advantages :

Extremely fast response time suitable for real-time interaction.

Supports character animations, gestures, and emotional responses.

Can be trained on custom documents (e.g., textbooks, PDFs), enabling deep, contextual conversations.

Requires minimal setup and integrates easily with Unity.

Disadvantages :

Requires a Convai account and API key ; advanced features may have usage limits or pricing tiers.

Less flexible in model selection compared to self-hosted solutions.

Limited control over model internals and fine-tuning options.

In conclusion, the method chosen for AI integration depends heavily on the priorities of the project. For real-time, cost-effective, and private interaction, local deployment is ideal. For high-quality, always-available intelligence, cloud APIs are preferred. And for systems requiring flexibility, the hybrid method presents a promising middle ground. This project primarily focuses on local deployment using LM Studio, while remaining open to future extensions via hybrid strategies.

3.16.1 Integration via Local LLMs Using LM Studio and Unity

In this method, I explored the integration of a locally hosted large language model (LLM) using LM Studio llama-3.2 as an example, in combination with Unity. LM Studio acts as a local inference server, allowing developers to host open-source models directly on their machine without relying on external APIs. This approach is especially valuable for offline use cases and full control over model behavior and data privacy.

The implementation process included the following steps :

Setting up the LM Studio local server and hosting a compatible open-source LLM.

Establishing HTTP communication between Unity and the local server via UnityWebRequest.

Sending prompts from Unity to LM Studio and receiving generated responses in real time.

Training and fine-tuning the model using smaller custom datasets to align the responses with the educational context of the virtual campus.

While this method proved functional and enabled prompt-based interaction with the model, it had some significant limitations. The primary challenge was the limited token capacity of the local model, which constrained the complexity and length of the prompts and responses. As a result, only short and simple interactions were feasible, making it less suitable for extended or highly contextual conversations.

Despite its limitations, this method served as a valuable proof of concept for locally powered AI integration within Unity.

3.16.2 Integration via Online APIs (DeepSeek and Qwen)

The second method explored the use of online AI inference APIs for integrating language models into the virtual campus system. Unlike the LM Studio approach, which relies on the user's local hardware capabilities, these APIs offer access to cloud-hosted large language models with significantly improved response speed and performance consistency. and i used the website OpenRouter To Find Those Free LLM's.

In this implementation, I connected Unity to publicly available LLMs via HTTP APIs. Two models were tested extensively :

DeepSeek : A high-performance open-source LLM accessible through online endpoints.

Qwen : An AI model developed by Alibaba, known for its language understanding and reasoning abilities.

These models were accessed through hosted API endpoints, allowing real-time text communication between Unity and the LLMs. The setup involved :

Registering for free access to the model providers.

Securing API keys and configuring UnityWebRequest to send JSON-formatted prompts.

Receiving and processing AI-generated responses in Unity's runtime environment.

This method demonstrated several advantages over the local LM Studio solution :

Faster response times, regardless of the user's machine specifications.

Higher model stability and lower maintenance effort, since the models are hosted externally.

Better overall output quality, especially in terms of coherence and natural language flow.

However, a major constraint remained — similar to the LM Studio approach — in the form of token and prompt size limitations. Due to the nature of free-tier APIs, I was unable to send long or highly structured prompts that would define the desired behavior and context in depth. This restricted the AI's ability to fully understand its role or simulate long-form conversation without losing context.

Despite these limitations, the online API method offered a more scalable and consistent integration path, especially for lightweight, web-connected AI experiences in Unity.

3.16.3 Integration via Convai Unity Plugin

The third method involved the use of Convai, a free Unity plugin available on the Unity Asset Store. Unlike previous methods that required manual setup for API calls or local hosting, 12AI offers a streamlined and user-friendly integration of AI functionalities directly within the Unity environment.

ConvaiAI stood out due to several significant advantages :

Low latency : The response time was noticeably faster and more consistent compared to both LM Studio and free-tier web APIs.

Support for custom training : A standout feature of 12AI is its ability to train the AI using custom data, including PDF files. I was able to upload documents relevant to the virtual campus (e.g., academic program guides, department brochures, or university policies), and the AI automatically fine-tuned itself based on this input.

Structured and reliable output : Regardless of whether the user's question was short or required a long and detailed answer, the AI provided well-organized and contextually appropriate responses. This made it particularly suitable for educational use cases, such as simulating virtual advisors or AI-powered information booths within the campus environment.

The implementation process was also remarkably simple :

Import the ConvaiAI package from the Unity Asset Store.

Configure the plugin within the Unity Inspector using the provided tools.

Upload relevant training documents (e.g., PDFs) through their interface or associated backend.

Connect the plugin to scene objects (e.g., NPCs) and enable interaction.

This method effectively addressed many of the limitations encountered in earlier approaches. There was no need to manage token sizes manually, and the quality of responses remained high across different types of queries. Additionally, the ability to contextually train the model using domain-specific documents opened new possibilities for personalized and intelligent virtual campus assistants,And Convai Didnt Put any resource about which LLM model They Use.

Overall, ConvAI provided a robust, scalable, and developer-friendly solution for integrating advanced conversational AI into Unity-based educational applications.

TABLE 3.1 – Comparison of AI Integration Methods

Feature / Method	**LM Studio (Local)**	**Online APIs (DeepSeek, Qwen)**	**Convai (Unity Plugin)**
Hosting	Local (runs on user's PC)	Cloud (external servers)	Cloud (integrated with Unity)
Integration Complexity	Moderate (manual API setup)	Moderate (API key setup & scripting)	Easy (drag-and-drop Unity plugin)
Performance	Varies by machine specs (can be slow)	Fast and stable (depends on server load)	Very fast and consistent
Training / Custom Knowledge	Limited (manual tuning or fine-tuning)	No (prompt-based only)	Yes (upload PDFs or documents for contextual training)
Prompt Size Limit	Yes (limited by model size and memory)	Yes (token limit on free tiers)	No (handles both long and short prompts effectively)
Quality of Responses	Medium (limited by model and prompt size)	Good (but not deeply personalized)	High (structured, accurate, and well-contextualized)
Latency	High (local computation delay)	Medium (depends on internet speed)	Low (near real-time cloud responses)
Scalability	Poor (dependent on user hardware)	Moderate (limited by API tier)	High (cloud-optimized and Unity-ready)
Best Use Case	Offline prototyping and experimentation	Basic online conversations without training	Advanced AI NPCs with personalized knowledge

3.17 Performance Comparison of Various Large Language Models

During the development of the AI-powered virtual campus system, multiple Large Language Models (LLMs) were tested and evaluated based on their response times, handling of prompt sizes, and integration flexibility. This comparison aimed to identify the most suitable model for real-time VR interaction, particularly in educational settings where responsiveness and consistency are critical. Below is a summary of the main models tested and the outcomes observed.

Jamba-1.5-Large (Cloud API Deployment)

Jamba-1.5-Large was accessed via a cloud-based API. It demonstrated strong performance with short prompts, returning answers in an average of 0.9 seconds. However, with longer prompts of approximately 230 tokens, the average response time increased significantly to around 20 seconds. While the model was accurate and capable of handling complex queries, the latency in long-form inputs made it less suitable for highly interactive real-time use cases in VR.

LLaMA 3.2B (Local Integration via LM Studio)

For on-device deployment, the LLaMA 3.2B model was tested using LM Studio. With short prompts, it achieved an average response time of around 1.7 seconds, which is acceptable for most VR dialogue interactions. However, when processing longer prompts such as 230 tokens, the model either failed to respond or returned errors, likely due to memory limitations or performance bottlenecks on the local machine. This makes the smaller version of LLaMA more appropriate for lightweight, real-time scenarios with limited prompt complexity. not that my computer has 32 GB RAM and a NVIDIA Quadro T1000 GPU with 4gb vram.

Convai (Pre-trained Conversational AI Platform)

Convai was evaluated as a third solution, particularly for NPC-style conversational agents trained on custom datasets. The model used was pre-trained on a large dataset (approximately 80 pages of a specific academic chapter). Despite the size and depth of the training data, Convai responded rapidly—averaging 0.8 seconds per prompt regardless of prompt length. This makes Convai an excellent candidate for scenarios requiring fast, knowledge-grounded responses in immersive environments.

3.17.1 Summary of Observations

Cloud-based LLMs offer the highest flexibility and accuracy but suffer from high latency and pricing issues with long prompts.

Local LLMs are responsive and cost-effective but struggle with long or complex prompts unless smaller, optimized models are used.

Convai offers an excellent middle ground, with low latency, custom training capabilities, and consistent performance, making it highly effective for educational NPC design in VR.

This comparative study played a crucial role in selecting the appropriate AI backend for the virtual campus, balancing cost, speed, and contextual reliability.

TABLE 3.2 – Comparison Table of LLMs

Model	Deployment	Short Prompt Time	Long Prompt Time	Notes
Jamba-1.5-Large	Cloud API	0.9 sec	20 sec (230 tokens)	High latency on long prompts
LLaMA 3.2B	Local (LM Studio)	1.7 sec	Error / Fail (230 tokens)	Fast for small prompts, unstable for long ones
Convai	Cloud / Hosted	0.8 sec	0.8 sec (trained on 80 pages)	Consistent speed, trained on domain-specific data

3.18 AI Integration Types

When integrating Artificial Intelligence into Virtual Reality (VR), the complexity and interactivity of the system can vary significantly depending on the type of AI used. In this section, we distinguish between three main categories of AI-driven systems that can be implemented in VR experiences. Each type reflects a different level of immersion, interactivity, and technical depth.

3.18.1 Basic Question-Answer AI

This is the most common and straightforward type of AI integration. In this setup, the AI functions as a passive chatbot : the user inputs a text or voice question, and the AI responds with a simple text or audio answer. There is no dynamic behavior or visual feedback from the virtual environment. The interaction is limited to a one-way exchange between the user and the AI, without any physical representation in the 3D world. Although useful for basic informational purposes, this method lacks immersion and emotional engagement.

3.18.2 Interactive AI with Animation

This type goes one step further by combining AI-generated responses with animated 3D characters in the virtual world. When the user asks a question, the AI not only answers but also triggers specific animations, such as talking gestures, hand movements, or facial expressions. This makes the AI character appear more lifelike and responsive, enhancing the sense of presence in the VR environment. The animations are typically managed using an animation tree in Unity or Unreal, synchronized with the AI's response behavior. and this Project Uses This Type

3.18.3 Autonomous NPCs with AI-Driven Social Interaction

The most advanced form of AI in VR involves Non-Player Characters (NPCs) that can interact not only with the user but also with each other. These NPCs are powered by AI models that simulate autonomous behavior and group dynamics. For example, two or more characters might carry on a conversation with each other, discuss different topics, or respond collectively to the user's presence. This creates a living, dynamic environment where AI agents exhibit believable social behavior, making the virtual world feel more authentic and immersive. Implementing this type of interaction requires complex logic, contextual awareness, and careful synchronization between dialogue systems, animation, and environmental triggers.

3.19 Conclusion

Chapter 3 detailed the comprehensive and structured development process for creating the AI-enhanced virtual campus. From the initial planning and modeling in Blender to the integration of realistic textures and the implementation of Unity-based navigation, every stage was carefully executed to ensure immersion, visual quality, and performance.

Key innovations included the use of HDRP for cinematic rendering, the integration of an intelligent NPC connected to a server-hosted LLM, and the creation of a responsive pathfinding system. These components work together to provide a seamless user experience where students can explore the campus naturally and engage with lifelike AI characters in real time.

The methodology outlined here lays the groundwork for the technical implementation of intelligent interactions, which will be explored further in the next chapter as we integrate the backend systems, LLMs, and dialogue logic that power the conversational experience.

CHAPITRE 4
SYSTEM IMPLEMENTATION : DEVELOPMENT TOOLS, WORKFLOW, AND APPLICATION SHOWCASE

4.1 Introduction

This chapter presents a comprehensive account of the end-to-end implementation of an AI-powered virtual campus, translating the theoretical framework outlined in Chapter 3 into a fully functional deployment. Unlike traditional virtual reality (VR) systems reliant on static, pre-scripted interactions, this project integrates real-time dialogue generation using Large Language Models (LLMs) within a dynamically rendered 3D environment. The implementation leverages a robust toolchain—Unity for VR rendering, Blender for 3D asset creation, and locally hosted LLMs (Llama 3) for intelligent non-player character (NPC) interactions—to achieve a balance of performance, visual realism, and interactivity. This chapter is structured to detail the development tools, system architecture, workflow, performance metrics, and a showcase of the final application. Screenshots, code snippets, and quantitative evaluations are embedded to substantiate technical claims and illustrate the system's usability. The chapter concludes with a discussion of challenges, solutions, and future directions, positioning this work as a replicable foundation for immersive educational platforms.

4.2 Development Tools and Frameworks

4.2.1 Unity Engine with HDRP

The project utilized Unity 2022.3 Long-Term Support (LTS) with the High-Definition Render Pipeline (HDRP) to deliver photorealistic visuals optimized for VR. HDRP's physically based rendering (PBR) facilitated dynamic lighting effects, such as sunlight reflections on glass facades, modulated by a time-of-day simulation. To ensure compatibility with mid-range hardware (e.g., NVIDIA RTX 3060), HDRP's Dynamic Resolution Scaling maintained a target frame rate of 40 FPS by adjusting render resolution in real-time. Unity's NavMesh system enabled efficient pathfinding, generating walkable surfaces for both users and NPCs across the virtual campus. Custom shaders, developed via Shader Graph, ensured seamless texture tiling for materials like brick walls and cobblestone paths, eliminating visual artifacts. A sample shader configuration for a brick material is shown below

4.2.2 Blender for 3D Asset Creation

All 3D assets, including campus buildings, foliage, and NPC avatars, were crafted using Blender 3.6. The workflow began with blockout modeling to establish spatial proportions, followed by high-resolution sculpting for architectural details (e.g., ornate columns, window frames). To optimize performance, the Decimate Modifier reduced polygon counts by approximately 60 while preserving visual fidelity, achieving an average of 10,000–20,000 triangles per building. NPC avatars were rigged with a humanoid skeleton, enabling animations such as walking and gesturing. Facial animations employed blend shapes for lip-syncing, with 22 visemes mapped to phonemes for realistic speech synchronization. Texture maps (albedo, normal, roughness) were generated using Substance Painter, ensuring PBR compatibility.

4.2.3 LM Studio with Llama 3

Dynamic NPC dialogues were powered by Llama 3 (7B parameter model), hosted locally using LM Studio to minimize latency and eliminate dependency on cloud APIs. A custom system prompt constrained NPC responses to campus-related topics, such as academic programs or facility locations.

4.3 System Architecture

The system architecture integrates three core components : VR Rendering, AI Dialogue System, and User Interaction Layer.

VR Rendering : Unity's HDRP renders the 3D campus, handling real-time lighting, shadows, and physics-based interactions. The NavMesh system manages NPC and user navigation. AI Dialogue System : Llama 3 processes user queries via LM Studio, interfacing with Unity through a C# script that handles intent recognition and response generation. Pre-loaded campus data (e.g., course catalogs) enhances response accuracy User Interaction Layer : A diegetic UI, voice/text input, and locomotion options (teleportation or smooth movement) ensure intuitive user engagement. The components communicate via a modular API, enabling scalability and ease of maintenance. For instance, the dialogue system can be swapped with an alternative LLM without altering the rendering pipeline.

4.4 Application Show-Case

4.4.1 UI Main Menu Design in Unity

To enhance the user experience and provide intuitive navigation within the VR application, I designed and implemented a Main Menu UI using Unity's UI system. This menu serves as the first interaction point for users, offering options such as "Start" and "Exit" to either launch the virtual campus or close the application.

The UI was developed using Unity's Canvas system, which allows for the creation of 2D interface elements that are rendered in screen space or world space. For this project, a world-space canvas was used to ensure the UI elements are anchored within the VR environment, allowing users to interact with them naturally using head or controller movement. Unity's Button component was utilized for the Start and Exit buttons, each connected to appropriate scripts that trigger scene loading or application termination.

By carefully positioning the canvas and optimizing its interaction settings for VR, the Main Menu provides a smooth and immersive entry point into the virtual experience, aligning with the principles of accessibility and user-centered design.

FIGURE 4.1 – Vr Main Menu

4.4.2 3D Modeling of Key University Locations

As part of the development of the immersive VR application, several important locations from the university were carefully modeled in 3D to replicate the real-world campus environment. These models serve as the core scenes users can explore, aiming to provide both spatial orientation and interactive educational potential. The modeled locations include :Dean's Office,Internet Room,Amphi 6,Mechanical Engineering Department,Laboratory STIQ,Architecture Department Each space was designed in Blender with attention to architectural detail and optimized geometry to ensure smooth performance within Unity. The models were then imported into the Unity engine, where materials, lighting, and collision components were added to enhance realism and interactivity in the VR experience.

This modeling process not only brings authenticity to the application but also allows users—especially new students—to become familiar with the campus layout in a virtual environment before engaging with it physically.

FIGURE 4.2 – Campus Places

4.4.3 Integration of AI-Powered NPC Character as a Virtual Teacher

To introduce a layer of interactivity and intelligent assistance within the VR campus, I integrated a Non-Player Character (NPC) driven by Artificial Intelligence to act as a virtual teacher. The character was designed to interact with students and answer educational queries in real time, specifically focused on the domain of Anatomy.

The 3D model of the character was created and animated in Blender, then imported into Unity. Using Unity's Animator and interaction systems, I enabled the character to engage with users through gestures and head movement to simulate a natural presence.

The AI capabilities were established by connecting Unity to a Large Language Model via LM Studio, allowing real-time text communication. A lightweight application was also developed to convert user voice input to text, which is then sent to the AI model. The AI's response is returned as text and optionally converted back into speech using a TTS (Text-To-Speech) system, enabling full conversational interaction.

The character was further trained or prompted with structured knowledge related to human anatomy, enabling it to simulate the role of an instructor. It can explain anatomical structures, answer questions, and guide the student through educational scenes—all within the immersive VR space.

This approach demonstrates the potential of combining conversational AI with immersive environments to create intelligent virtual agents capable of enhancing digital learning experiences.

FIGURE 4.3 – Anatomy Teacher

4.4.4 Department Selection Menu for Guided Navigation

To enhance the usability and personalization of the VR campus experience, I implemented a Department Selection Menu within the application. This interactive menu allows users to choose the specific department they wish to visit, enabling a targeted and efficient exploration of the virtual university.

The menu was designed using Unity's Canvas system and presented in world-space to integrate naturally into the VR environment. Each department is represented by a dedicated button, and upon selection, the system teleports the user to the corresponding 3D scene, such as the Architecture Department, Mechanical Engineering Department, or Laboratory STIQ.

This feature streamlines navigation, supports self-guided learning, and simulates the freedom of real-life movement across a campus. It also enables modular scalability, allowing more departments to be added in the future with minimal changes to the UI structure.

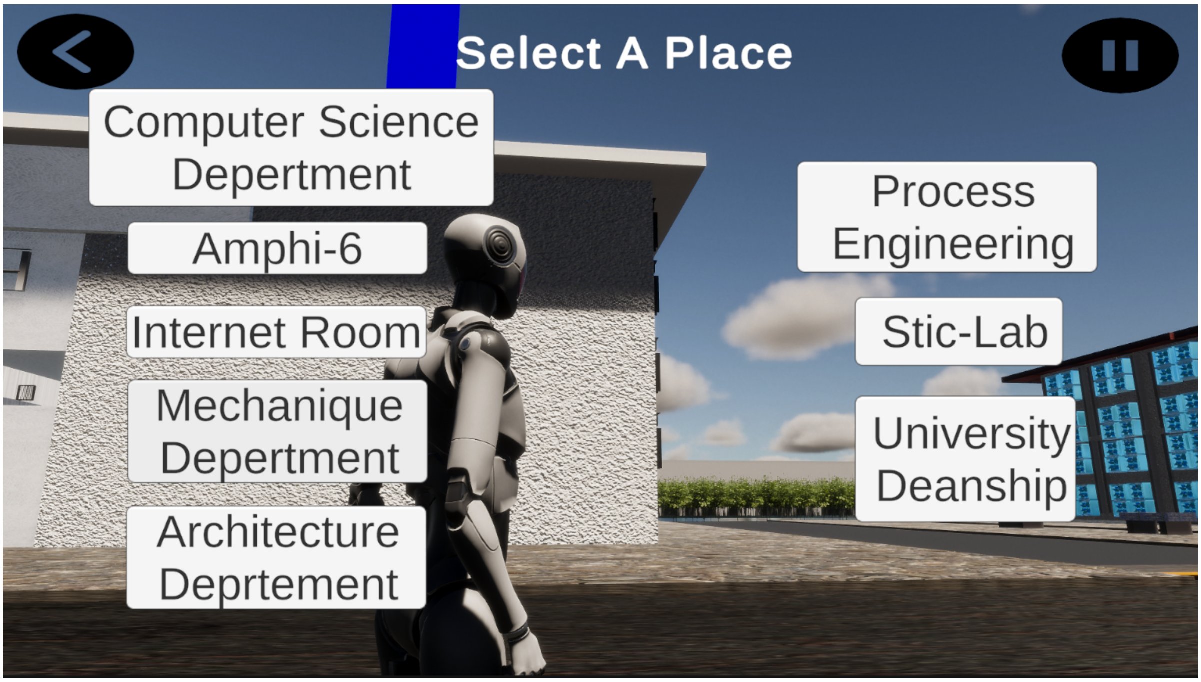

FIGURE 4.4 – Selection Menu

4.4.5 Directional Guidance System

To improve orientation and ensure users can easily find their way within the virtual campus, I implemented a Directional Guidance System using a 3D arrow that dynamically points towards the selected department's location. This feature was crucial for enhancing spatial awareness in the VR environment, especially for first-time users.

When a user selects a department from the Department Selection Menu, a navigation arrow appears in the scene, continuously pointing in the direction of the corresponding location. The arrow is programmed to update its rotation in real-time based on the player's current position and the destination coordinates. This ensures that users are always guided accurately toward their target.

This system was developed using Unity's Transform and Vector3 calculations, allowing the arrow to adjust smoothly and intuitively. It acts as a subtle yet effective visual cue, making exploration more natural and reducing confusion in complex scenes.

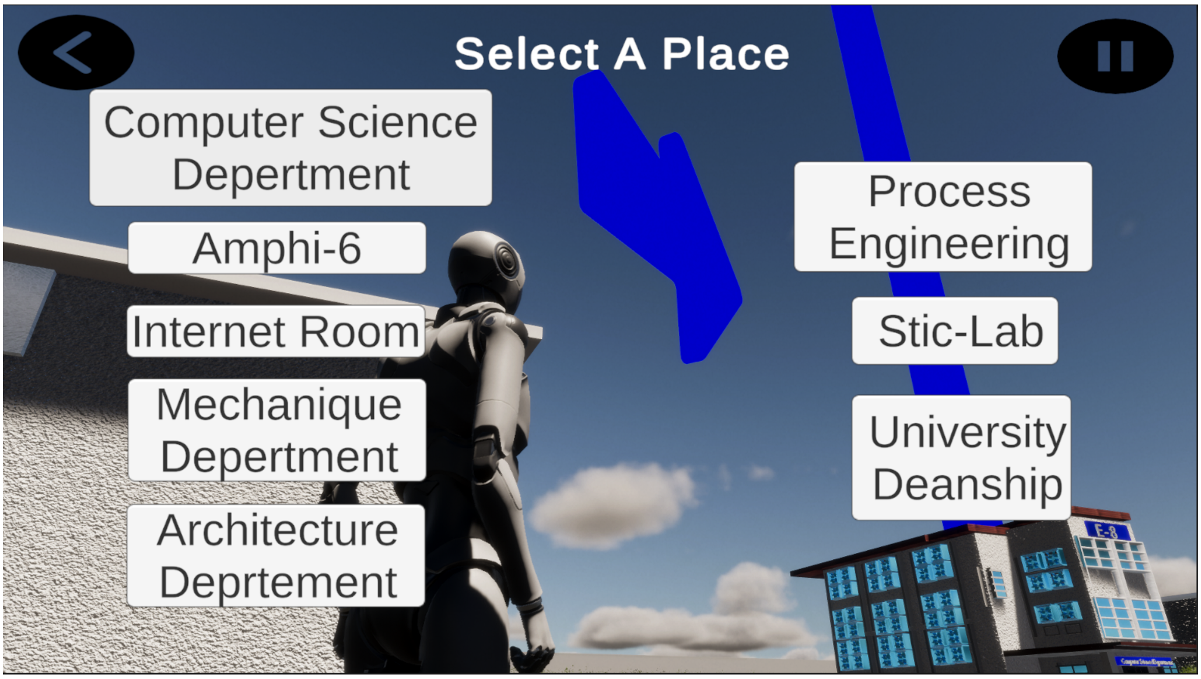

FIGURE 4.5 – Guidence System

4.5 Conclusion

This chapter presented the complete technical implementation of the AI-powered virtual campus, transforming the conceptual design outlined in Chapter 3 into a fully functional, interactive experience. By leveraging a robust combination of development tools, Unity for real-time rendering, Blender for accurate 3D modeling, LM Studio for local language model integration, and supporting tools like Materialize and Visual Studio, the project achieved a high level of realism, interactivity, and performance.

The use of a locally hosted LLM (Llama 3) enabled responsive and context-aware conversations with non-player characters, enhancing the educational value of the virtual tour. Advanced techniques such as diegetic UI design, optimized animation controllers, and voice-driven input pipelines contributed to a natural and engaging user experience. Visual fidelity was maintained through HDRP rendering and efficient optimization strategies, ensuring smooth performance across both desktop and

VR platforms.

Through detailed workflow explanations and application showcases, this chapter demonstrated not only how each technical component was implemented, but also how they came together to serve the core objective : providing an immersive, intelligent, and accessible exploration of a real university campus. This foundation now sets the stage for evaluating user experience and educational impact in the next chapter.campuses to transform academic engagement.

GENERALE CONCLUSION

This thesis explored the design, development, and implementation of an AI-enhanced virtual reality campus, aiming to redefine how students interact with educational environments. At its core, the project addressed a critical gap in existing virtual campus systems : the lack of meaningful interactivity, adaptability, and personalized user experiences. By integrating Large Language Models (LLMs) into a fully navigable 3D simulation of 8 Mai 1945 University, the system moved beyond traditional scripted tours toward a dynamic, conversational learning platform.

The research began by analyzing current VR campus solutions and identifying their technical and pedagogical limitations. A key insight was the potential of conversational AI, particularly LLMs, to create natural, responsive, and engaging educational experiences. Building on this, the thesis outlined a comprehensive methodology for constructing a smart virtual campus, incorporating 3D modeling, voice-based interaction, AI-driven non-player characters (NPCs), and real-time environment optimization.

The implementation phase demonstrated how modern development tools—such as Unity, Blender, Materialize, and LM Studio—can work together to build an immersive, scalable, and responsive VR environment. Through a detailed workflow, the project showcased the integration of locally hosted LLMs to support real-time dialogue, personalization, and natural language understanding, even in offline settings. The resulting application provided a highly interactive and accessible user experience, featuring voice-controlled navigation, NPC guidance, and contextualized responses to user inquiries.

This work contributes significantly to the growing body of research on AI and immersive technologies in education. It offers a practical and replicable model for future systems that combine intelligent agents with immersive environments. The system not only demonstrates the feasibility of intelligent, voice-enabled virtual campuses powered by LLMs but also highlights their potential as transformative tools for improving student orientation, engagement, and learning in digital spaces.

While the current system focuses on single-user interaction and limited academic data, it lays a robust foundation for future enhancements. Planned improvements include transitioning to cloud-hosted LLMs to support multi-user environments while maintaining low latency, implementing networked sessions for collaborative campus exploration, and integrating reinforcement learning to enable NPCs to adapt their behavior based on user interactions. Additionally, optimizing the system for standalone VR headsets, such as the Oculus Quest 3, will broaden accessibility and ensure cross-platform compatibility. These advancements will further enhance the system's scalability, interactivity, and alignment with institutional information systems, paving the way for more dynamic and data-driven educational experiences.

In conclusion, this thesis establishes that intelligent, voice-enabled virtual campuses powered by LLMs are not only feasible but also highly promising for the future of education. With continued refinement and the incorporation of these future enhancements, such systems have the potential to revolutionize how students engage with digital learning environments, fostering more personalized, collaborative, and immersive educational experiences.

BIBLIOGRAPHIE

[1] Katashi NAGAO. « Virtual reality campuses as new educational metaverses ». In : *IEICE TRANSACTIONS on Information and Systems* 106.2 (2023), p. 93-100.

[2] Liu YALI et Zhang HUIJIE. « Three-dimensional campus 360-degree video encoding VR technology based on OpenGL ». In : *Multimedia Tools and Applications* 79.7 (2020), p. 5099-5107.

[3] Bruce DAMER et al. « Steps toward Learning in Virtual World Cyberspace : TheU Virtual University and BOWorld ». In : *Interactions in Virtual Worlds. : A. Nijholt, OA Donk, EMAG van Dijk (eds.) : University Twente, Enschede* (1999), p. 31-43.

[4] Steven M LAVALLE. *Virtual reality*. Cambridge university press, 2023.

[5] Ishan MAHESHWARI et Piyush MAHESHWARI. « Effectiveness of immersive VR in STEM education ». In : *2020 seventh international conference on information technology trends (ITT)*. IEEE. 2020, p. 7-12.

[6] Abdullah M AL-ANSI et al. « Analyzing augmented reality (AR) and virtual reality (VR) recent development in education ». In : *Social Sciences & Humanities Open* 8.1 (2023), p. 100532.

[7] Jannat FALAH et al. « Identifying the characteristics of virtual reality gamification for complex educational topics ». In : *Multimodal Technologies and Interaction* 5.9 (2021), p. 53.

[8] Cosmina ISAR. « A glance into virtual reality development using unity ». In : *Informatica Economica* 22.3 (2018), p. 14-22.

[9] Mahmoud AL-BITAR et al. « The Use of Unreal Engine to Create Virtual Realty Fitness and Gaming Application ». In : *2019 IEEE 10th GCC Conference & Exhibition (GCC)*. IEEE. 2019, p. 1-5.

[10] Nikoletta ZOUBOULA, M FAKIDES et Costas TSOLAKIDIS. « Educational uses of virtual reality : Constructing a VR museum ». In : *International conference on interactice collaborative learning proceedings*. 2008, p. 24-26.

[11] Yuk Ming TANG et Ho Lun HO. « 3D modeling and computer graphics in virtual reality ». In : *mixed reality and three-dimensional computer graphics*. IntechOpen, 2020.

[12] Narayan H GANDEDKAR, Matthew T WONG et M Ali DARENDELILER. « Role of virtual reality (VR), augmented reality (AR) and artificial intelligence (AI) in tertiary education and research of orthodontics : An insight ». In : *Seminars in Orthodontics*. T. 27. 2. Elsevier. 2021, p. 69-77.

[13] Omer AYDIN, Enis KARAARSLAN et Nida Gokce NARIN. « Artificial intelligence, vr, ar and metaverse technologies for human resources management ». In : *arXiv preprint arXiv :2406.15383* (2024).

[14] Hugo TOUVRON et al. « Llama : Open and efficient foundation language models ». In : *arXiv preprint arXiv :2302.13971* (2023).

[15] Albert Q. JIANG et al. *Mistral 7B*. 2023. arXiv : `2310.06825 [cs.CL]`. URL : `https://arxiv.org/abs/2310.06825`.

[16] Yupeng CHANG et al. « A survey on evaluation of large language models ». In : *ACM transactions on intelligent systems and technology* 15.3 (2024), p. 1-45.

[17] Fabian GLOECKLE et al. « Better & faster large language models via multi-token prediction ». In : *arXiv preprint arXiv :2404.19737* (2024).

[18] Noor NASHID et al. « Contextual api completion for unseen repositories using llms ». In : *arXiv preprint arXiv :2405.04600* (2024).

[19] KR1442 CHOWDHARY et KR CHOWDHARY. « Natural language processing ». In : *Fundamentals of artificial intelligence* (2020), p. 603-649.

[20] Nurshamine NOR, Mohd SUNAR et Azyan KAPI. « A review of gamification in virtual reality (VR) sport ». In : *EAI Endorsed Transactions on Creative Technologies* 6.21 (2020).

[21] Felicitas STEIN et al. « Leveling up coolness : how VR gamification drives experiential values ». In : *Journal of Brand Management* (2025), p. 1-20.

[22] Elayne RUANE, Abeba BIRHANE et Anthony VENTRESQUE. « Conversational AI : Social and Ethical Considerations. » In : *AICS* 2563 (2019), p. 104-115.

[23] G Michael LEMOLE JR et al. « Virtual reality in neurosurgical education : part-task ventriculostomy simulation with dynamic visual and haptic feedback ». In : *Neurosurgery* 61.1 (2007), p. 142-149.

[24] Andrea Stevenson WON. « The virtual human interaction lab Stanford, California ». In : *XRDS : Crossroads, The ACM Magazine for Students* 22.1 (2015), p. 64-65.

[25] Jong-Hyun KIM. « Efficient Controlling Trajectory of NPC with Accumulation Map based on Path of User and NavMesh in Unity3D ». In : *Journal of The Korea Society of Computer and Information* 25.4 (2020), p. 55-61.

[26] *blender.org - Home of the Blender project - Free and Open 3D Creation Software*. `https://www.blender.org`. Accessed : 2025-06-04.

[27] *What is Photoshop ? Definition from WhatIs.com*. https ://www.techtarget.com/whatis/definit

[28] *What is it ?* http ://www.boundingboxsoftware.com/materialize/.

[29] *Visual Studio : IDE and Code Editor for Software Developers and Teams*. https ://visualstudio.microsoft.com.

[30] *C# Tutorial (C Sharp)*. https ://www.w3schools.com/cs/index.php.

[31] Saku ERÄNIITTY. « Creating a game scene in Unity ». In : (2020).