

VA in VWLE: Virtual Assistant in Virtual World Learning Environment

Pedro Andre Heckler Guidobono

Supervisor: Matthew Montebello

June 2024

*Submitted in partial fulfilment of the requirements
for the degree of **B.Sc. ICT in Artificial Intelligence (Hons.)***



L-Università ta' Malta
Faculty of Information &
Communication Technology

Abstract

Do you remember or have you heard about Microsoft Office's Clippy , the virtual assistant infamous for causing frustration rather than providing help? Advances in technology now enable the training of more effective virtual assistants. Despite its removal in 2007, a resurgence in the coming years would not be surprising, given the current AI Era we are living in nowadays.

You may not be familiar with Clippy, but if you have seen the movie Iron Man, you have likely imagined the convenience of having a Jarvis-like assistant. What if we could turn this science fiction into reality? With current AI tools, achieving similar goals is within reach. The same virtual assistant can be used in different scenarios. The idea here is to bring a virtual assistant into a virtual reality designed to education and expose the benefits it can bring to the learner.

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to my dear wife, who has shown immense patience and support throughout my time at the University of Malta. Before proceeding, I must also acknowledge my parents, who, despite being over 9,000 km away, have always been readily available to offer their help, just as they always have. I am equally grateful to CCBill for their flexibility with my work schedule and their constant encouragement to prioritize my studies. Additionally, I extend my appreciation to my friends, who have consistently been a source of encouragement and positivity. Last but not least, I owe a special thank you to my supervisor, Prof. Matthew Montebello, who has consistently provided insightful guidance and promptly responded to my inquiries, even on weekends.

Contents

Abstract	ii
Acknowledgements	iii
Contents	iv
List of Figures	vi
List of Tables	viii
List of Abbreviations.....	ix
1. Introduction.....	1
1.1 Problem Definition	1
1.2 Motivation	2
1.3 Aims and Objectives.....	3
1.4 Thesis Outline.....	4
2. Background and Literature Review.....	5
2.1 Virtual World Learning Environment	5
2.2 Deep Learning and Artificial Neuron Network	7
2.2.1 Artificial Neural Networks (ANNs)	7
2.2.2 Deep Learning (DL)	8
2.2.3 Supervised Learning / Unsupervised Learning	9
2.2.4 Natural Language Processing	9
2.2.5 Transformers.....	10
2.3 OpenAI.....	10
2.3.1 ChatGPT.....	11
2.3.2 APIs.....	11
2.4 Speech Technology	12
2.4.1 Speech-To-Text.....	12
2.4.2 Text-To-Speech.....	13
2.4.3 Libraries (APIs)	13
2.5 Unity	14
2.5.1 VR.....	14
2.6 Summary	14
3. Methodology	15
3.1 Subsection 1.....	15
3.1.1 Approaches	15
3.1.2 Decision taken	16
3.2 Subsection 2.....	17
3.2.1 setup environment.....	17
3.2.2 ChatGPT (OpenAI API).....	19
3.2.3 Speech To Text (STT)	20
3.2.4 Text To Speech (TTS)	24
3.2.5 Base of VA.....	27
3.3 Summary	29
4. Evaluation	30

4.1 Silent surrounding.....	31
4.2 Noisy surrounding.....	32
4.3 Outside.....	33
4.4 Summary	35
5. Conclusions.....	36
5.1 Revisiting the Objectives.....	36
References	38
Appendix A	40
A.1 ChatGPT communication script	40
A.2 Links to this project resources	41
Appendix B Outdoor evaluation	42

List of Figures

Figure 1.1 - Clippy Office Assistant images snipped from [1].....	1
Figure 2.1 - Overview of VR Usage in Higher Education: Statistical Perspective [3]....	6
Figure 2.2 - Complementary perspective to Figure 2.1 from different source [4].....	6
Figure 2.3 - Representation of perceptron with and without bias [8].....	7
Figure 2.4 - Visual representation of deep neural network [7].....	8
Figure 2.5 – API free tier rate limit [22]......	12
Figure 2.6 - Spectrogram images as features, text as target [25]......	13
Figure 3.1 – Creating new Unity project (3D URP).	17
Figure 3.2 - XR Plugin Management.....	18
Figure 3.3 - XR Interaction Toolkit importing Sample.....	18
Figure 3.4 – Project Settings, Player I.	18
Figure 3.5 - Project Settings, Player II.	18
Figure 3.6 – Basic VR env scene.....	19
Figure 3.7 - Package Manager OpenUPM.....	19
Figure 3.8 – Script and whole project available digitally.	20
Figure 3.9 – Meta Asset.	20
Figure 3.10 – Importing sample from dictation package.....	21
Figure 3.11 – Adding XR canvas to hold sample's objects.....	21
Figure 3.12 – Coping objects from Dictation Sample.	22
Figure 3.13 – Ensure OnClick() kept settings.....	22
Figure 3.14 – User Interface (UI) of STT sample.....	22
Figure 3.15 – Canvas Transform values.....	22
Figure 3.16 – Transcription Inspector.	23
Figure 3.17 – AI Inspector.	23
Figure 3.18 – STT and OpenAI API.....	23
Figure 3.19 – Textbox to edit VA's role.	24
Figure 3.20 – Installing TTS sample.	25
Figure 3.21 - TTS sample with 2 panels.	25
Figure 3.22 – TTS Voices Sample.....	25
Figure 3.23 – Diagram to prepare TTS.....	26

Figure 3.24 – Substituting Display Text in AI object.....	26
Figure 3.25 – Clear transcription when “Speak” is clicked.....	27
Figure 3.26 – All in. Base of VA.....	27
Figure 3.27 – First look.....	28
Figure 3.28 – How to access the VA.....	28
Figure 3.29 – Final look of the VA	28
Figure 3.30 – Read Response.....	28
Figure 4.1 – Location for Evaluation 4.3.....	34
Figure 4.2 – Graph overall performance.....	35

List of Tables

Table 1 – Performance on silent surrounding.....	31
Table 2 - Performance on noisy surrounding.....	32
Table 3 - Performance Outside.....	33

List of Abbreviations

VA	Virtual Assistant
VWLE	Virtual World Learning Environment
STT	Speech to Text
TTS	Text to Speech
VR	Virtual Reality
AR	Augmented Reality
NLP	Natural Language Processing
GPT	Generative Pre-trained Transformer
ANN	Artificial Neural Network
DNN	Deep Neural Network

1. Introduction

In educational environments, the need for quick access to information and prompt responses to queries is critical. Yet, many virtual learning environments fall short in providing effective solutions, often leading to disengagement and diminished educational efficacy.

My early experiences with Microsoft's Clippy, the animated virtual assistant introduced in the late 1990s and removed in 2007, highlighted both the potential and limitations of early artificial intelligence (AI) helpers. Although Clippy was often more distracting than helpful, I must admit, I still miss its company during my writing sessions. These experiences underscore the necessity for a more sophisticated and intuitive system.

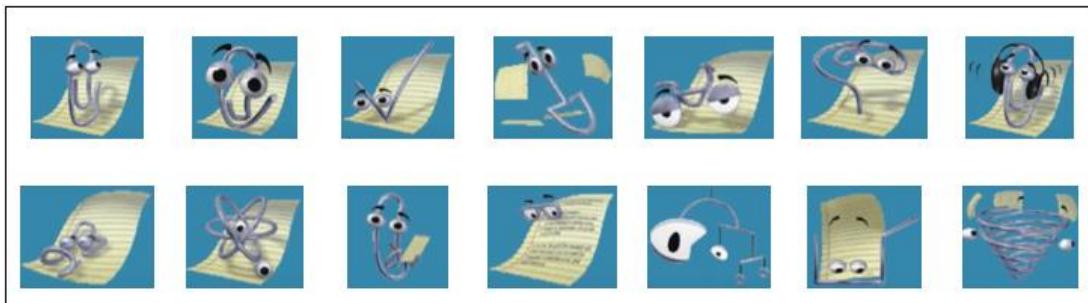


Figure 1.1 - Clippy Office Assistant images snipped from [1].

With the current advancements in AI tools, this dissertation aims to transform cinematic fantasies, like the capabilities seen in Jarvis from 'Iron Man,' into practical educational tools. By proposing the development of a Virtual Assistant (VA) within a Virtual World Learning Environment (VWLE), this project seeks to address these shortcomings by leveraging advanced AI technologies for dynamic interaction and support.

1.1 Problem Definition

Current virtual learning platforms often fall short in providing interactive and intuitive assistance, making it challenging for students to obtain quick and relevant information. This shortfall can lead to disengagement and negatively impact learning outcomes, particularly for individuals with ADHD. For instance, my personal

experience with using dual monitors, one dedicated to ChatGPT for real-time assistance, has proven significantly more effective in maintaining my engagement compared to using conventional search engines like Google, Bing, or Yahoo. These traditional methods often require additional filtering to retrieve appropriate answers, increasing the likelihood of distractions and diverting attention away from the task at hand.

With this context in mind, this project aims to develop a user-friendly virtual assistant (VA) that operates effectively within Virtual World Learning Environments (VWLEs), thereby enhancing both Virtual Reality (VR) and Augmented Reality (AR) educational applications.

1.2 Motivation

Several sources inspired the motivation for this project:

- Inspiration from Fictional Models: The iconic representation of virtual assistants like Jarvis in the fictional work Iron Man, serves as inspiration. The desire to turn such futuristic concepts into reality is a driving force, motivating the exploration of innovative solutions for educational technology.
- Enhanced Learning Experience: Many virtual learning environments are static, offering limited interaction possibilities. Introducing a sophisticated Virtual Assistant (VA) can lead to more engaging and personalized learning experiences, thus improving comprehension and maintaining attention, which is particularly beneficial for individuals with ADHD.
- Technological Advancements: The current era of advanced AI technologies, such as transformer-based generative models like ChatGPT, Speech-To-Text (STT) a model which transcribes a speech into text, and Text-To-Speech (TTS), provides an opportune moment to explore and implement cutting-edge solutions in education. The company OpenAI, which owns ChatGPT, provides APIs in order for developers to create new applications that are able to use functionalities from ChatGPT and other models provided by them. And by harnessing these technologies, the project aspires to contribute to the ongoing evolution of AI applications in learning environments.

- Efficiency and Accessibility: The envisioned virtual assistant seeks to streamline access to information and support, providing students with quick responses to queries and facilitating efficient navigation within the virtual learning environment. This would contribute to a more efficient learning process.

1.3 Aims and Objectives

The primary aim of this dissertation is to enhance the interactivity and effectiveness of virtual learning environments through the development of a sophisticated Virtual Assistant (VA) within a Virtual World Learning Environment (VWLE). This VA will leverage the latest advancements in AI technology to create a more dynamic, intuitive, and engaging educational experience. The specific objectives to achieve this aim are as follows:

- O1.** Conduct thorough research into AI technologies that can be integrated into virtual learning environments. This includes exploring transformer-based generative models, speech-to-text, and text-to-speech technologies to enhance the functionality and responsiveness of the virtual assistant.
- O2.** Design and develop a simplified virtual reality (VR) framework that serves as a foundation for deploying the virtual assistant (VA). This framework should be easily adaptable, allowing the virtual assistant to be integrated seamlessly into various VR environments. The design will prioritize modularity and flexibility, ensuring that the virtual assistant can enhance interactivity and personalized learning experiences across different virtual reality platforms.
- O3.** Develop a prototype of the virtual assistant that can understand and respond to user queries in real-time, providing immediate assistance and support within the VWLE. This prototype will be tested for its effectiveness in improving user engagement and learning outcomes.
- O4.** Evaluate the quality and efficiency of the virtual assistant to determine its effectiveness in assisting users with inquiries during navigation within a virtual reality (VR) environment. The goal is to analyse if the virtual assistant indeed enhances the user's ability to obtain relevant information promptly and effectively, thereby improving the overall experience within the VR.

This dissertation will explore these objectives, aiming to demonstrate the feasibility and educational benefits of integrating such a virtual assistant into virtual learning environments.

1.4 Thesis Outline

In the subsequent chapters, you will explore all the knowledge and steps necessary to achieve each objective. Chapter 2 presents a background and literature review of the theoretical AI techniques and tools applicable to this project. Chapter 3 is divided into two subsections detailing the methodology: Subsection 1 briefly reviews all approaches attempted, noting both failures and partial successes, while Subsection 2 provides a detailed step-by-step guide on how to create your own Virtual Assistant for VR. Chapter 4 outlines the methods used to evaluate the Virtual Assistant, including the tests performed and a graph comparing the results. Finally, Chapter 6 concludes the report and discusses potential future work that could further enhance the system.

2. Background and Literature Review

Building on O1, this chapter explores existing research and technologies that are pertinent to this project. It delves into prior studies and assesses various technological advancements to identify tools and methodologies that could enhance the project's outcomes.

2.1 Virtual World Learning Environment

"Throughout human history to date, it has been the environments, not the human beings, that have run up against limitations." [2] said in 1968.

From my experience, I believe that this phrase was still valid until the 90's. However, a lot has been changing with the advent of computers and the easy access to information through the internet.

A Virtual World Environment, synonymous with Virtual Reality (VR) or artificial environment, employs advanced technology to create artificial environments that simulate real-world sensory experiences. Users engage with these environments in real-time, often through specialized equipment like VR headsets, data gloves, and input devices like keyboards and mice, enabling immersive interaction within these digitally constructed spaces. This technology allows for a blend of visual, auditory, and sometimes haptic feedback, enhancing the sense of presence in the virtual world [3]. The use of VR and educational games in VWLEs is transforming the educational landscape. A study reviewing 31 articles found a predominance of Oculus Rift headsets and HTC Vive in Educational Virtual Reality Games (EVRGs). These games are being developed for all education levels and primarily focus on out-of-class use, with healthcare education being a dominant topic [4].

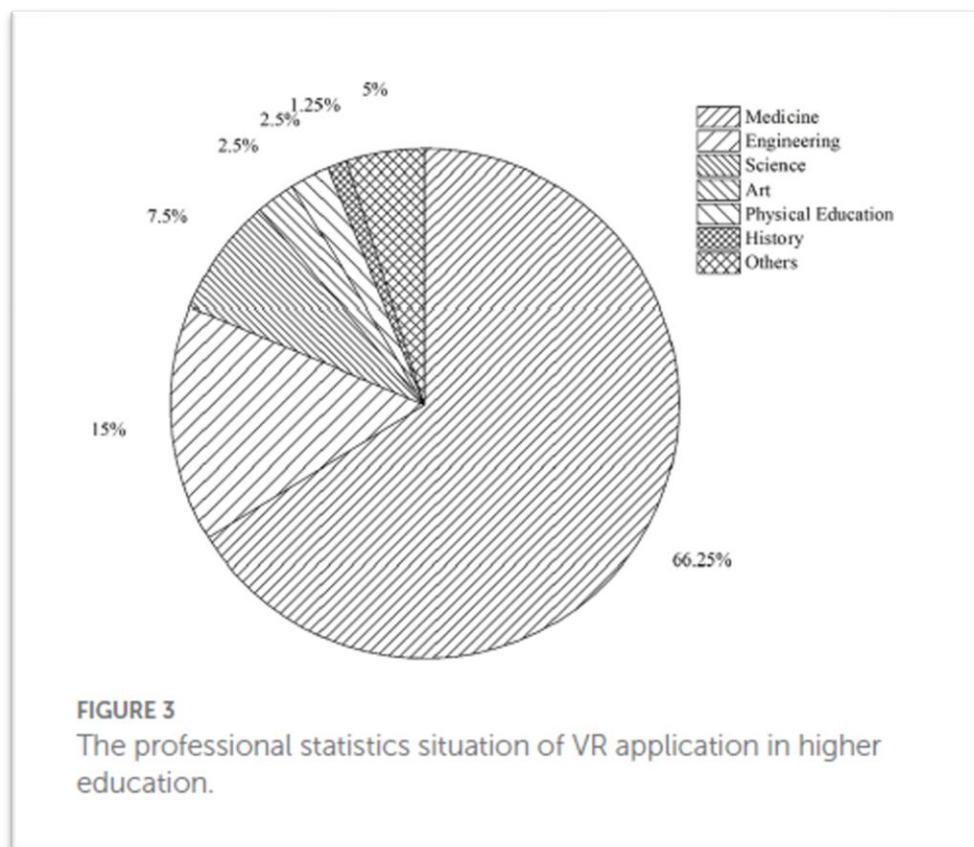


Figure 2.1 - Overview of VR Usage in Higher Education: Statistical Perspective [3].

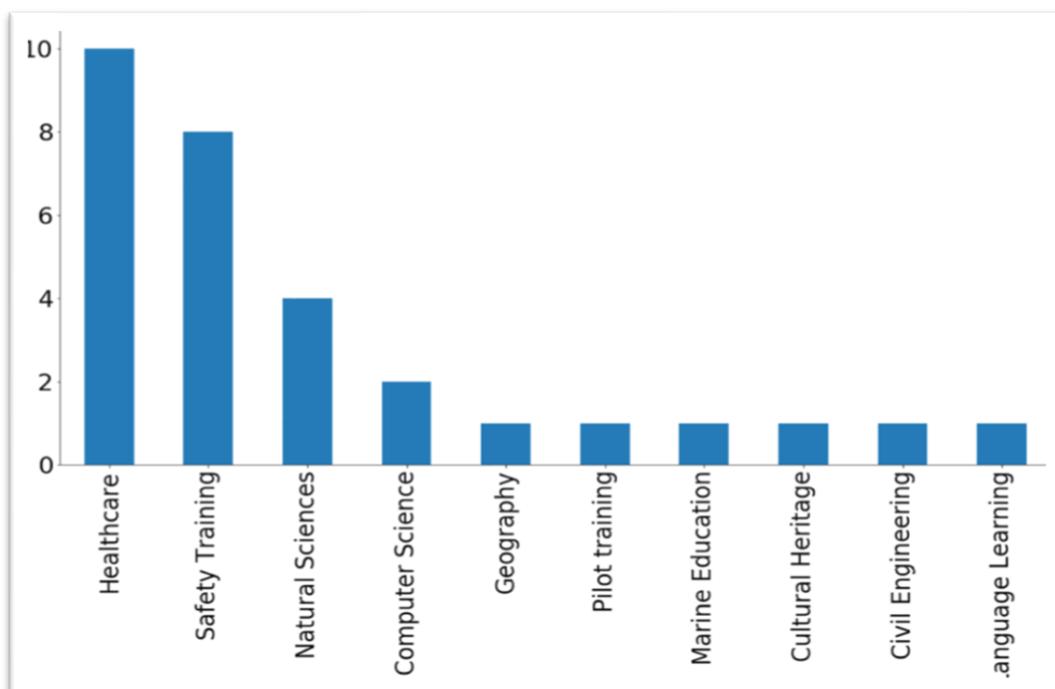


Figure 2.2 - Complementary perspective to Figure 2.1 from different source [4].

2.2 Deep Learning and Artificial Neuron Network

2.2.1 Artificial Neural Networks (ANNs)

ANNs find their origins in the concept of computational models inspired by the human brain's neural structure. McCulloch and Pitts introduced the foundational idea of an artificial neuron in 1943, simulating the basic functionality of a biological neuron [5]. These early developments paved the way for the perceptron model proposed by Frank Rosenblatt in 1958 [6].

Neural networks, commonly referred to as artificial neural networks or simulated neural networks (SNNs), form a crucial part of machine learning and the core of deep learning algorithms. Drawing inspiration from the human brain, they are designed to replicate how biological neurons transmit signals to each other [7].

ANNs consist of several layers of nodes: an input layer, multiple hidden layers, and an output layer. Each node, also known as an artificial neuron, is linked to others, each with a specific weight (w_1, w_2, \dots, w_n) and threshold. A node activates and transmits data to the next layer if its output exceeds this threshold. If not, it doesn't pass on any data [7].

The perceptron represents the most basic form of an ANN, consisting of just one input layer and a single output node [8].

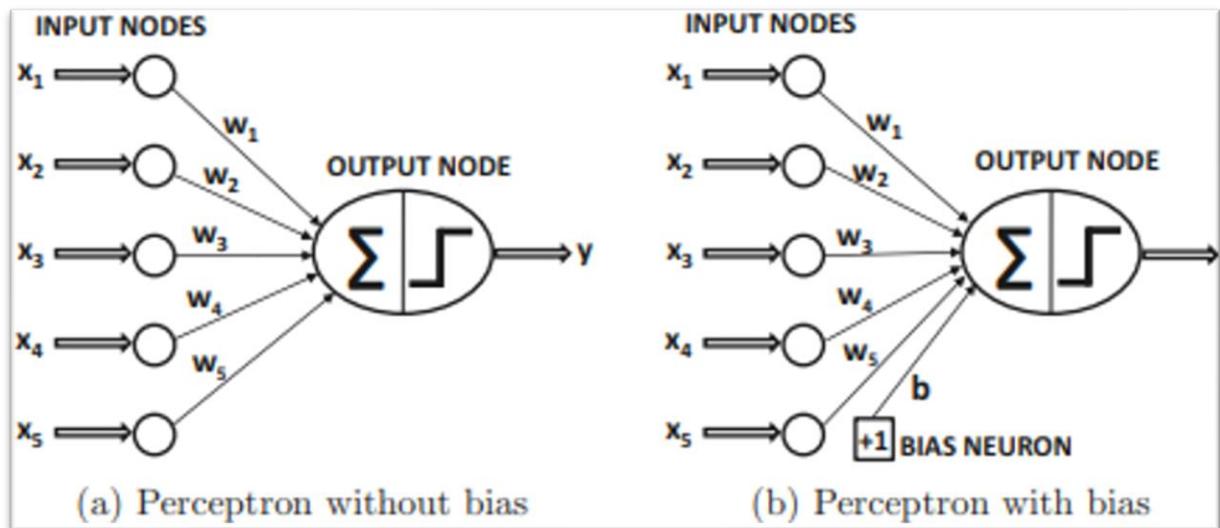


Figure 2.3 - Representation of perceptron with and without bias [8].

2.2.2 Deep Learning (DL)

DL represents a significant advancement in the realm of artificial intelligence, building upon the principles of Artificial Neural Networks. Geoffrey Hinton, along with Yoshua Bengio and Yann LeCun, played pivotal roles in the development of deep learning techniques, especially convolutional neural networks (CNNs) and recurrent neural networks (RNNs) [9]. Unlike traditional ANNs, DL involves the training of neural networks with multiple layers, enabling the extraction of intricate features and patterns from complex datasets. The depth of these networks enhances their ability to understand hierarchical representations, making them more adept at tasks like image recognition, natural language processing, and speech recognition [10].

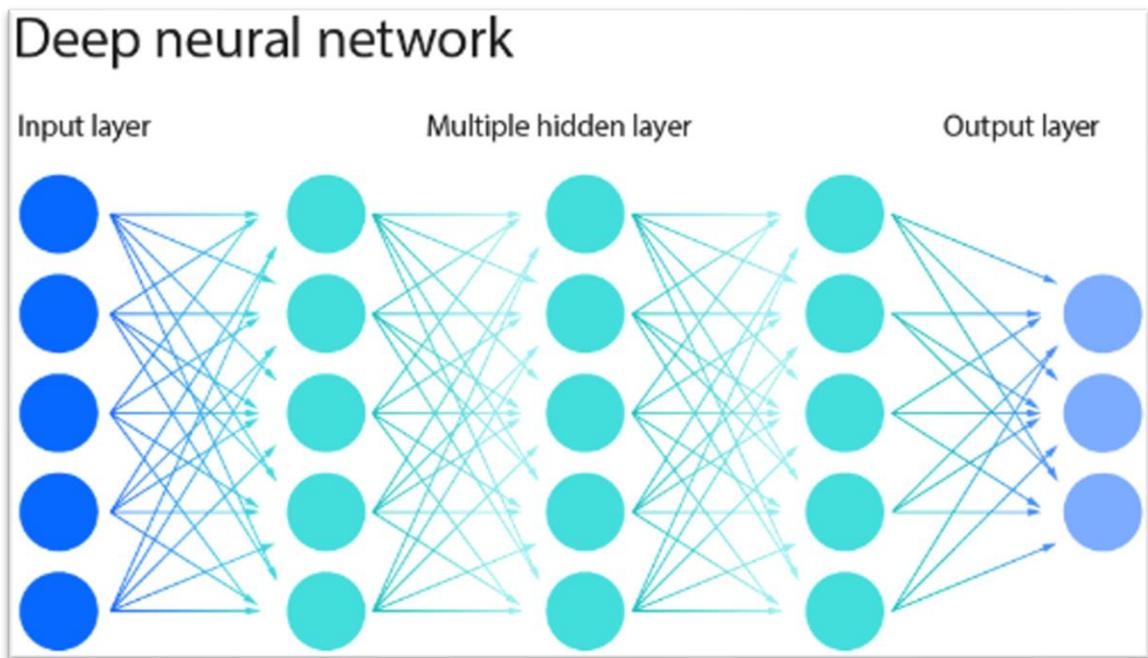


Figure 2.4 - Visual representation of deep neural network [7].

Connection and Difference:

While Artificial Neural Networks serve as the foundation for Deep Learning, the key distinction lies in the depth of the network architecture. Deep Learning involves training networks with multiple layers, allowing for the extraction of hierarchical representations, and has demonstrated remarkable success in various domains. Both ANNs and DL rely on the fundamental principle of learning from data

through adjustments to weights, but DL's increased depth enables it to model more complex relationships within the data.

2.2.3 Supervised Learning / Unsupervised Learning

In the realm of artificial intelligence and machine learning, two fundamental approaches exist: supervised learning and unsupervised learning. The key distinction between them lies in the utilization of labelled data for making predictions in one case and the absence of such labels in the other [11].

Supervised learning uses labelled datasets to train algorithms for accurate classification and prediction. It includes classification (sorting data into categories) and regression (predicting numerical values). Examples of algorithms in supervised learning include decision trees, linear classifiers, and support vector machines [11].

Unsupervised learning analyses unlabelled datasets to discover hidden patterns without human intervention. It encompasses clustering (grouping similar data), association (finding relationships between variables), and dimensionality reduction (simplifying high-dimensional data). K-means clustering is a well-known clustering algorithm [11].

2.2.4 Natural Language Processing

Natural Language Processing (NLP) is a subset of computer science, particularly within the realm of AI, dedicated to equipping computers with the capability to comprehend both written text and spoken language, mirroring the way humans process and understand linguistic input [12].

Pioneering works in NLP date back to the late 1940s [13], However, it was the advent of machine learning techniques that brought about transformative changes. Notably, the application of Hidden Markov Models (HMMs) in speech recognition [14] and the development of rule-based systems for language translation marked significant strides. The evolution continued with the introduction of neural network-based approaches, particularly recurrent neural networks (RNNs) and convolutional neural networks (CNNs), enabling more robust and context-aware language understanding [15]. Contemporary breakthroughs, such as the widespread adoption of transformer architectures, exemplified by models like BERT [16] and GPT [17],

have propelled NLP into new frontiers, achieving unprecedented performance in tasks like sentiment analysis, machine translation, and text summarization.

2.2.5 Transformers

The Transformer architecture, introduced by [18] in 2017, has revolutionized natural language processing (NLP) and sequential data processing. It provides a scalable and parallelizable structure, overcoming limitations of traditional recurrent and convolutional models.

Transformers utilize a self-attention mechanism, allowing simultaneous processing of all elements in a sequence. With an encoder-decoder setup, they excel in capturing long-range dependencies and context, proving effective in machine translation, sentiment analysis, and text summarization [18]. In other words, Transformers have a special feature that lets them look at and work with all parts of a sentence at the same time. They're designed with two main parts, one that reads and understands the context and another that puts the information to use. This setup helps them to be really good at seeing how different parts of a sentence are connected, no matter how far apart they are. This makes them excellent tools for tasks like translating languages, figuring out feelings in text, and summarizing long articles.

Bidirectional Encoder Representations from Transformers (BERT), introduced by [16] in 2018, extended Transformer capabilities with pre-training on large corpora. BERT's pre-trained models have set benchmarks in transfer learning for various NLP tasks. Generative Pre-trained Transformers (GPT), developed by OpenAI, employs an autoregressive approach for natural language generation. GPT has become a standard for coherent and contextually relevant text generation [16].

The Transformer architecture stands as a cornerstone in deep learning, offering versatility and effectiveness across diverse applications, pushing the boundaries of artificial intelligence.

2.3 OpenAI

OpenAI, established in 2015, is an AI research and deployment company. They have a mission focused on ensuring that artificial general intelligence (AGI) benefits all of

humanity. They construct their generative models utilizing deep learning technology, harnessing substantial datasets to train an AI system for task performance [19]. The company has developed various groundbreaking products, including ChatGPT which uses GPT (Generative Pre-trained Transformer) model to interact and generate responses based on user inputs. Also, image generation models like DALL-E, and Whisper, which focuses on speech recognition and translation [20].

2.3.1 ChatGPT

ChatGPT, one of OpenAI's most notable creations, is a conversational AI model. It was developed using transformer-based architectures to generate human-like text responses. The model has undergone several iterations, improving its capabilities with each version. ChatGPT has been trained on a diverse range of internet text and is designed to provide informative, relevant, and contextually appropriate responses [21].

2.3.2 APIs

OpenAI offers various APIs that allow developers to integrate its AI models into different applications. These APIs support multiple programming languages and provide access to technologies like deep learning, computer vision, and natural language processing. The DALL-E API for image generation, the Codex API for building custom AI models, and the Safety API for ensuring the security and ethical alignment of AI systems are among these offerings. These APIs can automate manual tasks, save time and costs, and improve accuracy and efficiency in various applications. However, developers should consider technical complexity, cost, and data privacy concerns.

OpenAI frequently updates its policies, and during the course of this project, the 'gpt-3.5-turbo' model was available for free for a limited time following the creation of an OpenAI account. Pricing and usage limits vary and are detailed in [22], which also describes the different service tiers OpenAI offers. These tiers are referred to as 'Tiers,' and the free tier's rate limits is illustrated in Figure 2.5. Rate limits are measured in several ways: RPM (requests per minute), RPD (requests per day), and TPM (tokens per minute). The first paid tier (Tier 1) starts at \$5 and includes a specific usage allowance for the GPT-4 model which is an evolutionary step forward from

GPT-3.5, offering enhanced capabilities and refinements that improve its utility and interaction quality.

MODEL	RPM	RPD	TPM	BATCH QUEUE LIMIT
gpt-3.5-turbo	3	200	40,000	200,000
text-embedding-3-large	3,000	200	1,000,000	3,000,000
text-embedding-3-small	3,000	200	1,000,000	3,000,000
text-embedding-ada-002	3,000	200	1,000,000	3,000,000
whisper-1	3	200	-	-
tts-1	3	200	-	-
dall-e-2	5 img/min	-	-	-
dall-e-3	1 img/min	-	-	-

Figure 2.5 – API free tier rate limit [22].

2.4 Speech Technology

The advancements in artificial intelligence have significantly improved machine learning, NLP, and deep learning. These improvements have opened new possibilities in specialized fields, especially in speech technology. AI is now widely used in various aspects of speech technology, including speech analysis, modelling, and representation. It also plays a crucial role in recognizing and understanding spoken language, affective speech recognition, interpretation, synthesis, designing interfaces, human factors engineering, emotion recognition technologies, and integrating audio-visual aspects in speech processing [23].

2.4.1 Speech-To-Text

Speech-to-text (STT), also known as speech recognition or computer speech recognition, is a software that utilizes computational linguistics to convert spoken language into written text [24].

The training classification is done by using audio of spoken words as input features (X), while text from the transcript of the same audio serves as target (y) [25].

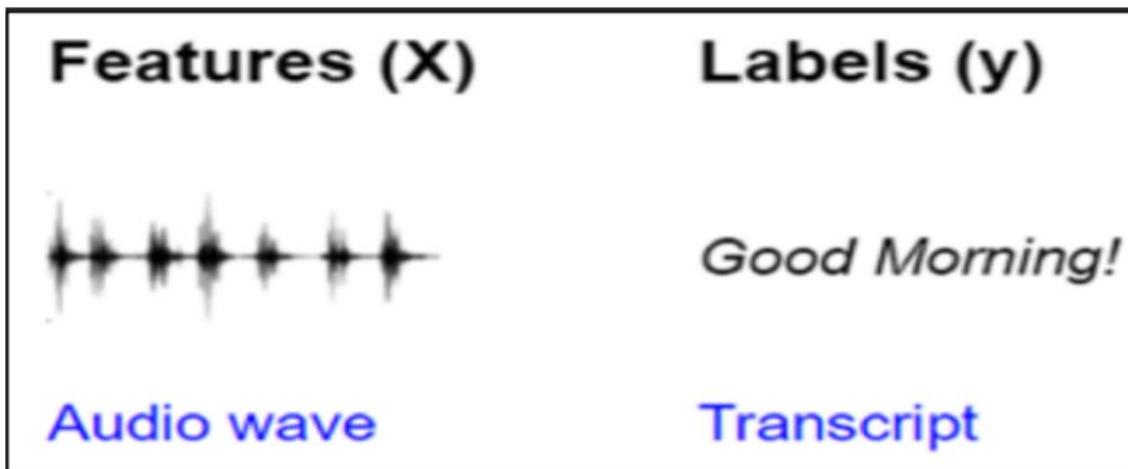


Figure 2.6 - Spectrogram images as features, text as target [25].

For sound signal classification, the process involves transforming raw audio waves into Mel spectrogram images. This approach aids in extracting the user's sound signal while reducing noise. "Librosa", is a widely used deep learning Python library employed for this purpose [25].

2.4.2 Text-To-Speech

Text-to-speech (TTS) technology transforms text into artificial speech. Over recent decades, this technology has advanced significantly, especially with the integration of deep learning. This progression allows for the creation of highly natural-sounding speech, complete with variations in pitch, speed, pronunciation, and intonation.

Today, computer-generated speech is widely used in various user interfaces (UIs). It has a range of applications including news reading, gaming, public announcement systems, e-learning, telephony, IoT devices, and personal digital assistants, among others [26].

2.4.3 Libraries (APIs)

Numerous libraries exist for tasks like TTS and STT, offering various customization levels. Google's API provides a monthly free tier, with the first 1 million characters for WaveNet (a deep neural network for generating raw audio) voices and the first 4 million characters for Standard voices free each month. Beyond these limits, charges apply per million characters processed [27]. While it might be sufficient for this projects, free alternatives are available online, including alternatives that work offline.

2.5 Unity

Unity is a versatile engine and framework for building both 2D and 3D game or application scenes. It's used not only for games but also for various applications like training simulations and business tools. Unity offers an interactive system combining coding with visual elements, supporting export to major mobile platforms and beyond, free of charge (with a more feature-rich pro version available). It's compatible with major 3D apps and audio formats, even supporting Photoshop (.psd files.) Within Unity, users can import assets, write scripts, incorporate animations, and utilize a sophisticated animation system, among other features [28].

2.5.1 VR

Unity enables the creation of immersive VR experiences, including engaging games and interactive storytelling. Its multiplatform support extends to various headsets like SteamVR, **Meta Quest**, and PlayStation VR. This broad compatibility is enhanced with optimized graphics and profiling tools, and an XR Interaction toolkit that simplifies and accelerates development across platforms [29]. Meta provides a good range of packages in Unity Store which will be explored to help in the development of my virtual assistant.

2.6 Summary

This chapter has provided a thorough review of the core AI technologies and virtual environment concepts essential for developing advanced virtual assistants within a Virtual Reality. It highlighted the importance of a VWLE, the most recent advancements in AI, such as deep learning, neural networks, speech processing technologies and tools available which can help in deploying them.

This foundation is crucial as we move forward to the practical aspects of the dissertation. The next chapters will delve into the methodology used to implement these technologies.

3. Methodology

This chapter is divided into two subsections. The first describes the approaches taken during the trials and failures, and the decision of which approaches were chosen to be used. The second provides a detailed, step-by-step guide on how to build the base for a Virtual Assistant using **Unity Editor version 2022.3.10f1** and deploy it on Oculus Meta Quest 2.

The main tools needed to make this Virtual Assistant (VA) are: STT, OpenAI API to send text and retrieve response, and TTS.

3.1 Subsection 1

After weeks spent exploring different ways to implement these features, most attempts were unsuccessful. Although most solutions worked when accessed from a computer or mobile device, the speech-to-text (STT) and text-to-speech (TTS) functionalities did not operate properly after being compiled and tested on the Oculus Meta Quest 2.

3.1.1 Approaches

Different approaches were taken, including:

- **APPROACH_1** - A Python implementation that worked well on Windows using libraries such as:
 - a. ‘speech_recognition’ for speech-to-text tasks.
 - b. ‘pyttsx3’ for text-to-speech conversion.
 - c. ‘OpenAI’ to send text and retrieve responses for creating messages.

This approach works perfectly in a Windows environment. However, to implement this with Unity, I would need to set up a server to continuously run the Python script, which the Oculus would need to communicate with. Therefore, I decided to explore other alternatives.
- **APPROACH_2** - Given that Unity primarily utilizes the C# programming language, inspired by [30],[31], I implemented a similar script in C# instead of Python and used it in a Unity project for VR. I had to fix the script in order to use updated OpenAI methods, although it worked correctly when executed in the Unity Editor

and also worked as a mobile app tested on Android, it was not successful when tested on the Meta Quest 2. However, the communication with OpenAI was successful from the Meta device. Therefore, the remaining issues to address were with the speech-to-text (STT) and text-to-speech (TTS) functionalities.

- **APPROACH_3** - I experimented with a Meta library from the Unity asset store, which enables the use of the system keyboard. This allowed me to utilize the keyboard's microphone for speech-to-text tasks. However, to use this feature, I had to activate the "Voice Command" setting in my Oculus, which is only available in the US. I overcame this restriction by installing a VPN to simulate a US location, enabling the use of "Voice Command." While this allowed the keyboard's microphone to function within my app, "Voice Command" is still a BETA feature only available in the US. Consequently, I continued my search for more reliable approaches.
- **APPROACH_4** - I researched existing implementations of TTS and STT for Oculus. I found numerous resources providing instructions on how to resolve the TTS issue using the Oculus package integrated with the wit.ai API.
- **APPROACH_5** - The only issue that remained was STT, for which I could not find any solutions through standard searches. However, after exploring Meta's library on my own, I came across a feature called 'Voice SDK – Dictation.' Initially, I dismissed it as irrelevant because I had not seen it recommended for STT in any instructions, and many people seemed to be looking for it, judging from the comments I had read. Nevertheless, I decided to give it a try. Surprisingly, it had exactly what I needed, and I wouldn't need to use the 'Voice Command' feature nor VPN at all.

3.1.2 Decision taken

For communication with ChatGPT, part of APPROACH_2 was employed, and the library utilized was OpenUPM with scopes: com.openai and com.utilities. For the STT task, the asset from Unity AssetStore **Meta XR Voice SDK – Dictation** developed by Meta was used as discovered in APPROACH_5. This package automatically installs **Meta - Voice SDK - Immersive Voice Commands** as dependency. Following APPROACH_4 I imported a TTS Voice Sample from **Meta - Voice SDK - Immersive**

Voice Commands. In order to use these packages, a key from <https://wit.ai> must be created If samples' keys do not work.

By simply extracting the main canvas from the samples available in the last two packages mentioned above, it is easy to work with *speech-to-text* and *text-to-speech*. And with the ChatGPT script from the 2º Approach mentioned in **3.1.1**, everything needed to make a fully functional virtual assistant is in place.

3.2 Subsection 2

Here, I provide a detailed step-by-step explanation of how Objectives **O2** and **O3** were implemented. The solution for Objective **O2** is explained in the subsection 3.2.1, and the subsequent subsections detail the steps taken to complete Objective **O3**.

3.2.1 setup environment

Create a new project 3D URP.

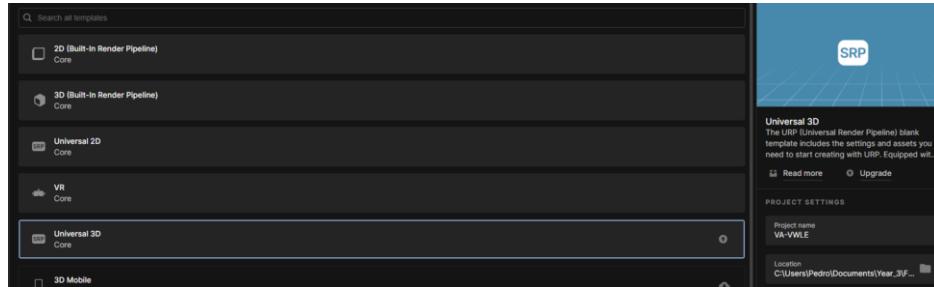


Figure 3.1 – Creating new Unity project (3D URP).

There are plenty guides and documentation explaining how to set the environment to be used to develop apps for VR. One that I found easy to follow can be found in [32]. Switch to Android Environment in “File/Build Settings...”. Select “Android” (Make sure it is installed) and click in “Switch Environment”. Note that Texture Compression worked only with ETC2 (GLES 3.0) in the Build Settings window.

In “Window/Package Manager”, **XR Plugin Management** and **XR Interaction Toolkit** were installed. And in “Edit/project Settings/XR Plug-in Management” Oculus was selected.

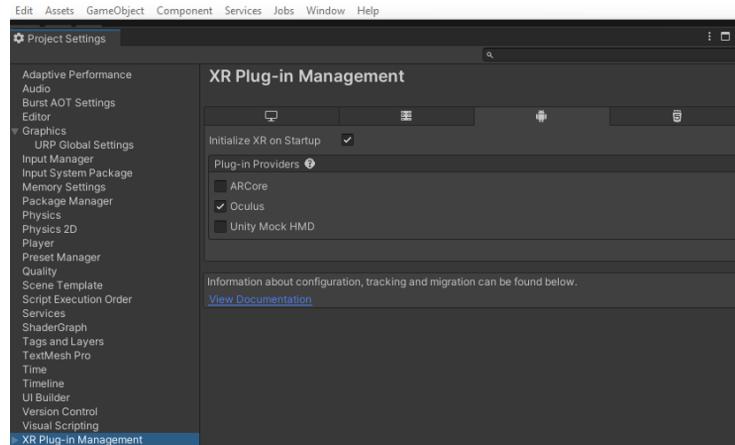


Figure 3.2 - XR Plugin Management.

Back in Package Manager, the “Starter Assets” sample from **XR Interaction Toolkit** was also installed.

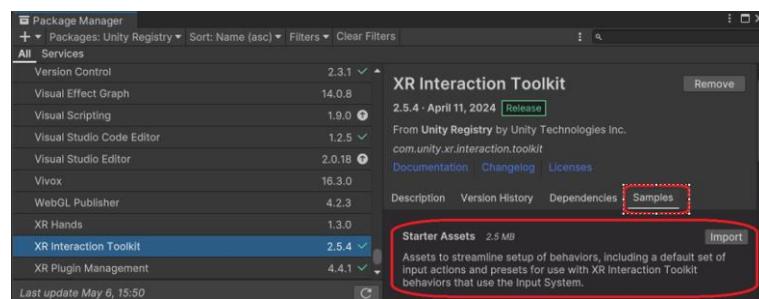


Figure 3.3 - XR Interaction Toolkit importing Sample.

In “Edit/Project Settings/Player” I respected the following configuration:

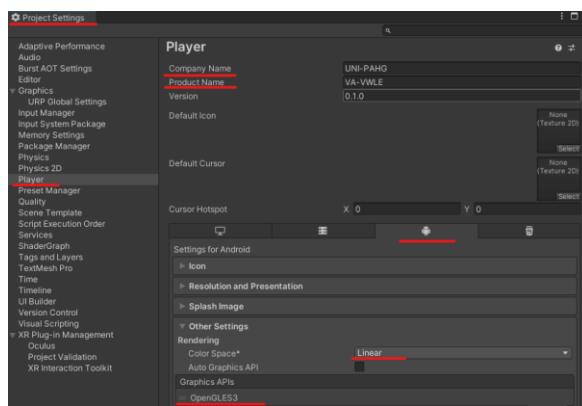


Figure 3.4 – Project Settings, Player I.

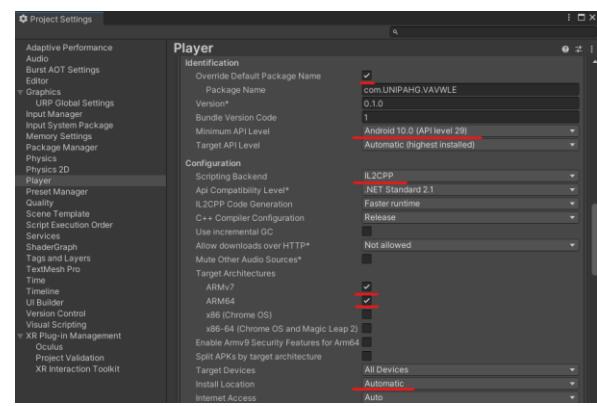


Figure 3.5 - Project Settings, Player II.

And in order to have a quick VR environment ready with controls and other trivial settings ready, so I could focus on the Virtual Assistant implementation only, I deleted the main Camera Object and copied all objects from a DemoSample scene to my scene.

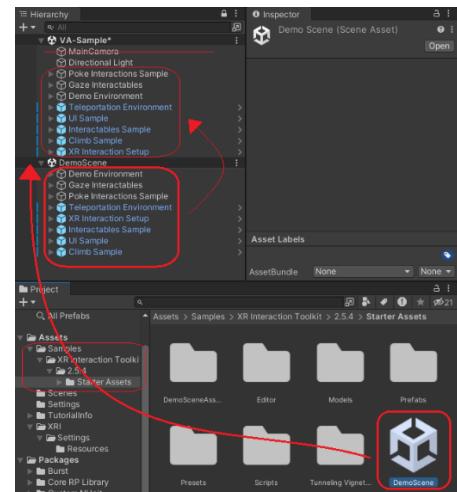


Figure 3.6 – Basic VR env scene.

At this point, this project can be built, and you can walk around the scene and use the controls.

3.2.2 ChatGPT (OpenAI API)

Settled OpenUPM package following the steps from their website [33]. I simply went to “Edit/Project settings/Package Manager” fill up the Name, URL and Scopes as below and saved.

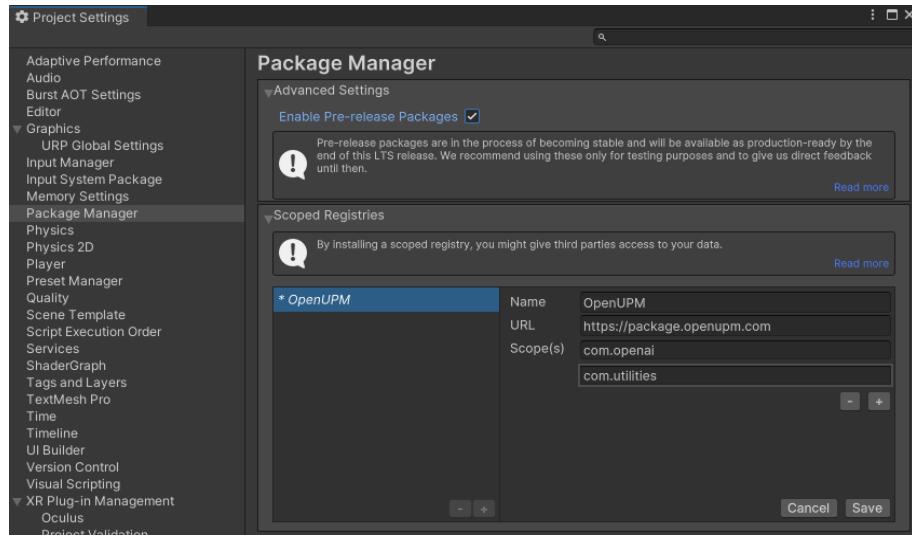


Figure 3.7 - Package Manager OpenUPM.

Once it is saved, the new package can be installed from “Window/Package Manager” under the “My Registries”. Select OpenAI and install it. The subsequent scopes will be installed automatically as its dependencies.

Errors may occur along the way like some .dll file missing. It is usually solved by closing the Editor and opening it again. Remember to save it!

In the Hierarchy, I have created empty objects just to name it differently in order to separate each part of the project. In the “-----GPT” area in the Editor hierarchy, an empty object called “AI” was created and add a script called **ChatGPT.cs** to it.

You can find the main part of the script in **Appendix A** which is short and well commented.

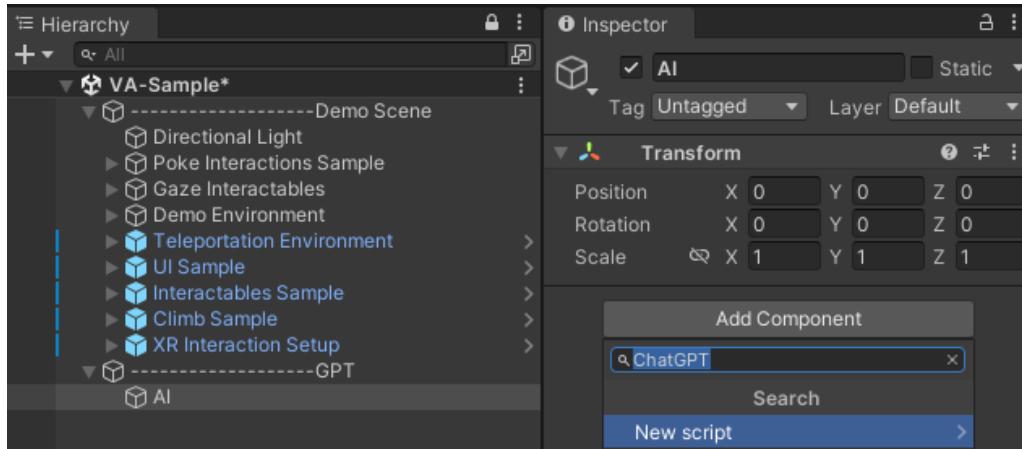


Figure 3.8 – Script and whole project available digitally.

The script was inspired by [30] where a mobile app which communicates with ChatGPT was developed. The script from [30] is deprecated though and some adjustments were needed since the methods used there were decommissioned and the new method expects a list instead of string.

3.2.3 Speech To Text (STT)

In order to implement this technology into the VR, the Unity asset **Meta XR Voice SDK – Dictation** by Oculus [34] was installed. It automatically installs another Meta asset as dependency, the **Meta – Voice SDK – Immersive Voice Commands**.

Followed this, a sample is imported from the Voice Dictation package. See Figure 3.10

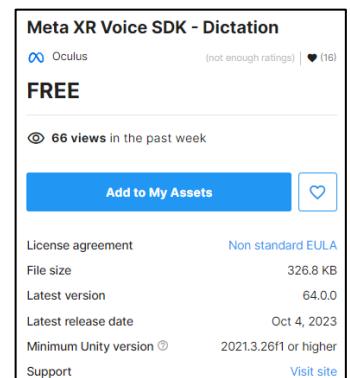


Figure 3.9 – Meta Asset.

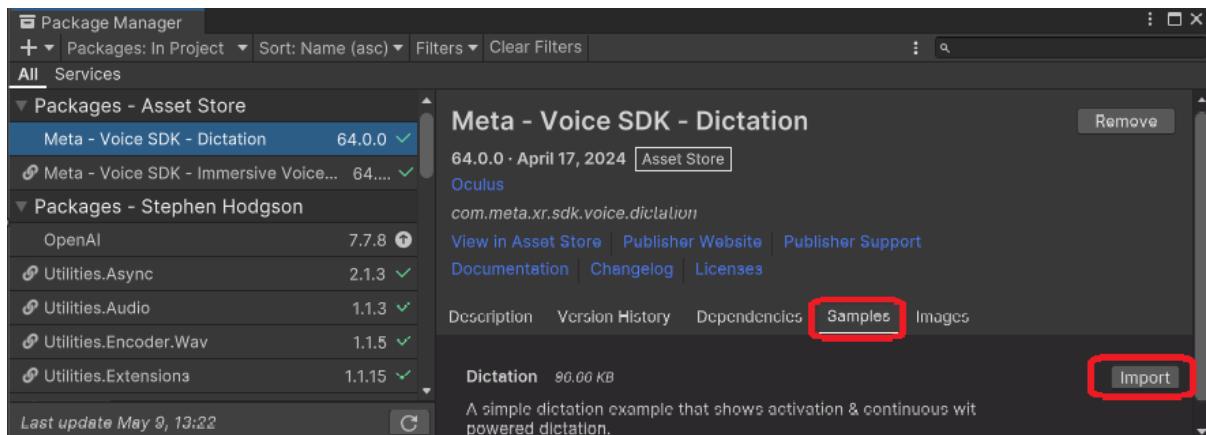


Figure 3.10 – Importing sample from dictation package.

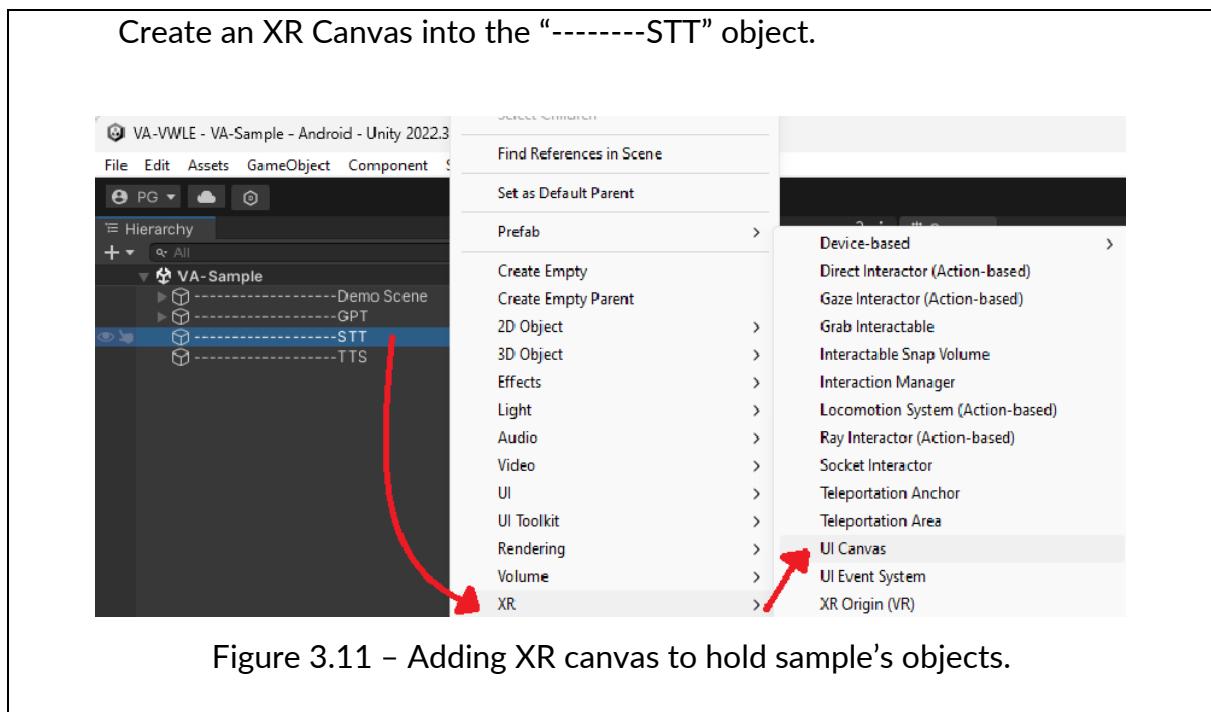


Figure 3.11 – Adding XR canvas to hold sample's objects.

Copy the objects depicted from Dictation sample scene.

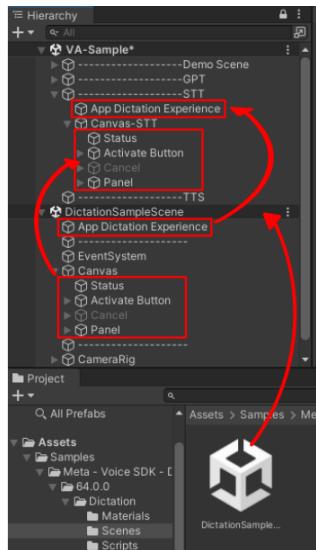


Figure 3.12 – Coping objects from Dictation Sample.

Make sure it kept the configuration in the **Activate Button**. Copying all together should keep the configuration. On Click uses **App Dictation Experience** object to define the method.

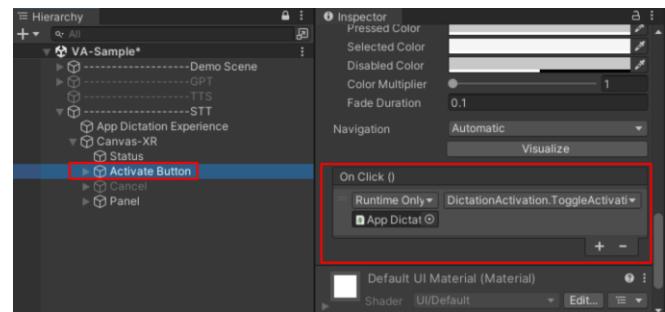


Figure 3.13 – Ensure OnClick() kept settings.

At this point, “**TextMesh Pro**” package is needed, and once this sample scene is brought to the hierarchy, a window should pop-up to import it, otherwise, you can always import it manually from Package Manager.

If texts are on the opposite side, rotate canvas 180° degrees. Also set canvas Render Mode to **World Space** and Decrease its scale. See Figure 3.15.

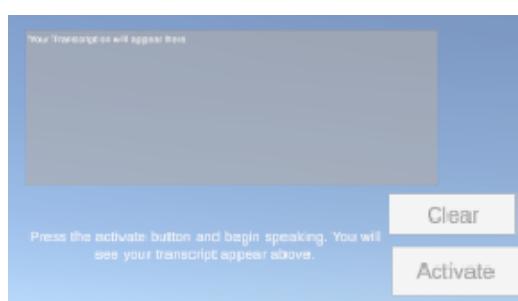


Figure 3.14 – User Interface (UI) of STT sample.

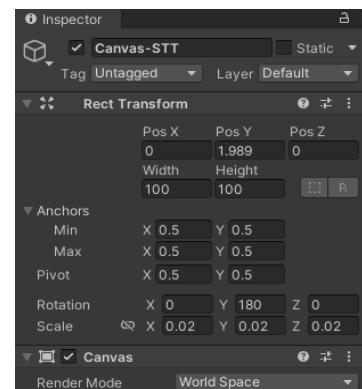


Figure 3.15 – Canvas Transform values.

By pressing play on Unity Editor, it can be tested. “AI” should not be active for this test since it is missing assigning some instances to it. This will be defined later.

In Activation Separator replace “;” with “.”
And use 1 Line between Activation instead
of 2.

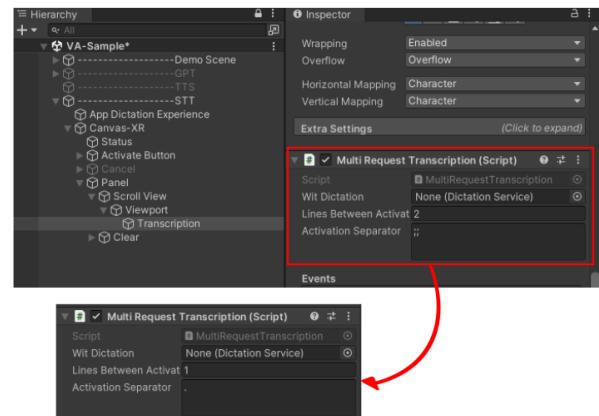


Figure 3.16 – Transcription Inspector.

To test the communication of the progress so far, we can add a new XR Canvas with a button, and assign this button, as well as the transcription to the designated areas into the ChatGPT script located in AI object. See Figure 3.17.

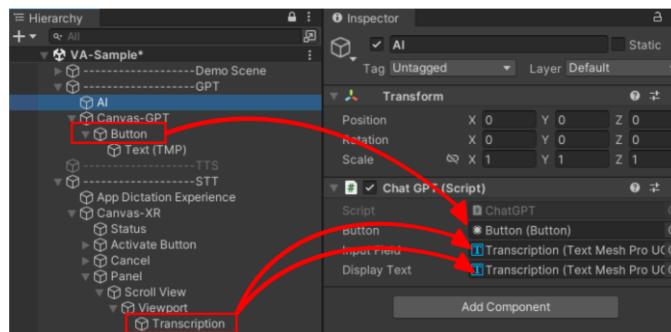


Figure 3.17 – AI Inspector.

The Figure 3.18 shows the STT sample, and a button created in the “-----GPT” in other to test the speech to text feature and the communication with OpenAI. The response from ChatGPT will be displayed in the same text area of the STT sample.

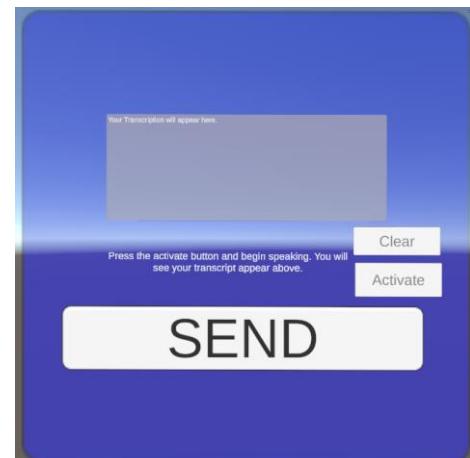


Figure 3.18 – STT and OpenAI API.

Transcribe speech and get response from ChatGPT.

I have also added a variable in the script to hold a text to be sent to ChatGPT at the first call. This way, it can be settled to keep short answers. I have it Serialized (which means that it is editable in Unity, see Figure 3.19) and added the following text: *You are a Virtual Assistant. Please, ensure to keep responses short, favourably with one line but make sure to always have less than 100 words unless asked for more details. If you understand, say 'hello, how can I Help?'* I have also added a field for the API key.

Once we open the VA for the first time, if everything works as expected, the user should listen the VA saying “hello, how can I Help?“.

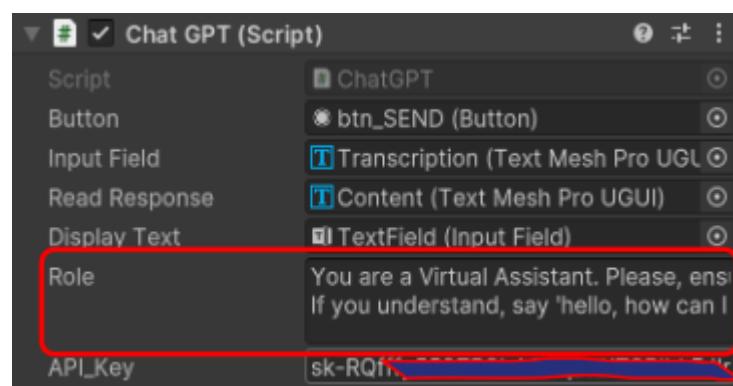
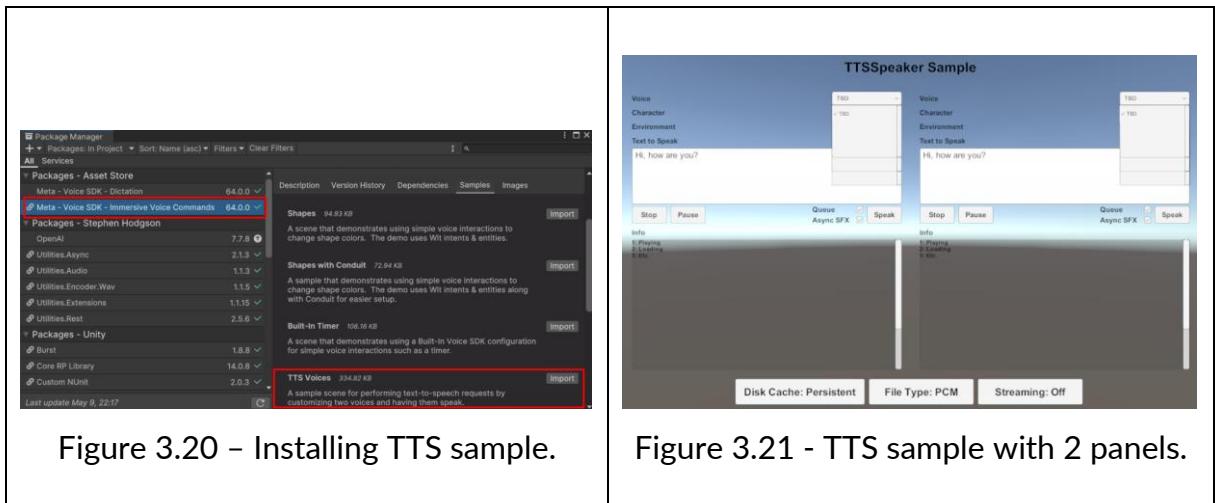


Figure 3.19 – Textbox to edit VA’s role.

3.2.4 Text To Speech (TTS)

Last, the conversion of the text response from ChatGPT to Speech. To do this, go into Package Manager and look for the package already in the project called **Meta - Voice SDK - Immersive Voice Commands**, click on **Samples** and import **TTS Voices**. See Figure 3.20.



This sample contain a scene with two panels which one can be deleted and the whole Canvas as well as the TTSSwitService objects can be copied to our scene. Figure 3.21 depicts how the sample looks like as a whole.

The following Figure 3.22 and Figure 3.23 depicts a diagram to prepare the base for the TTS using one panel from the sample displayed in Figure 3.21.

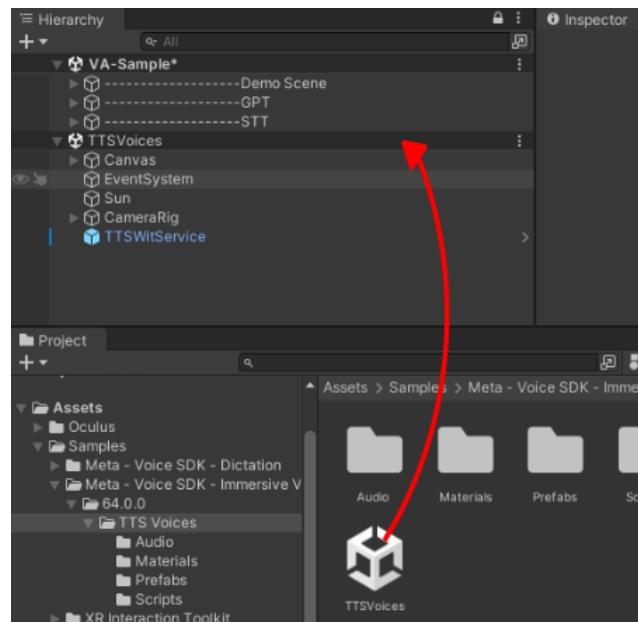


Figure 3.22 – TTS Voices Sample.

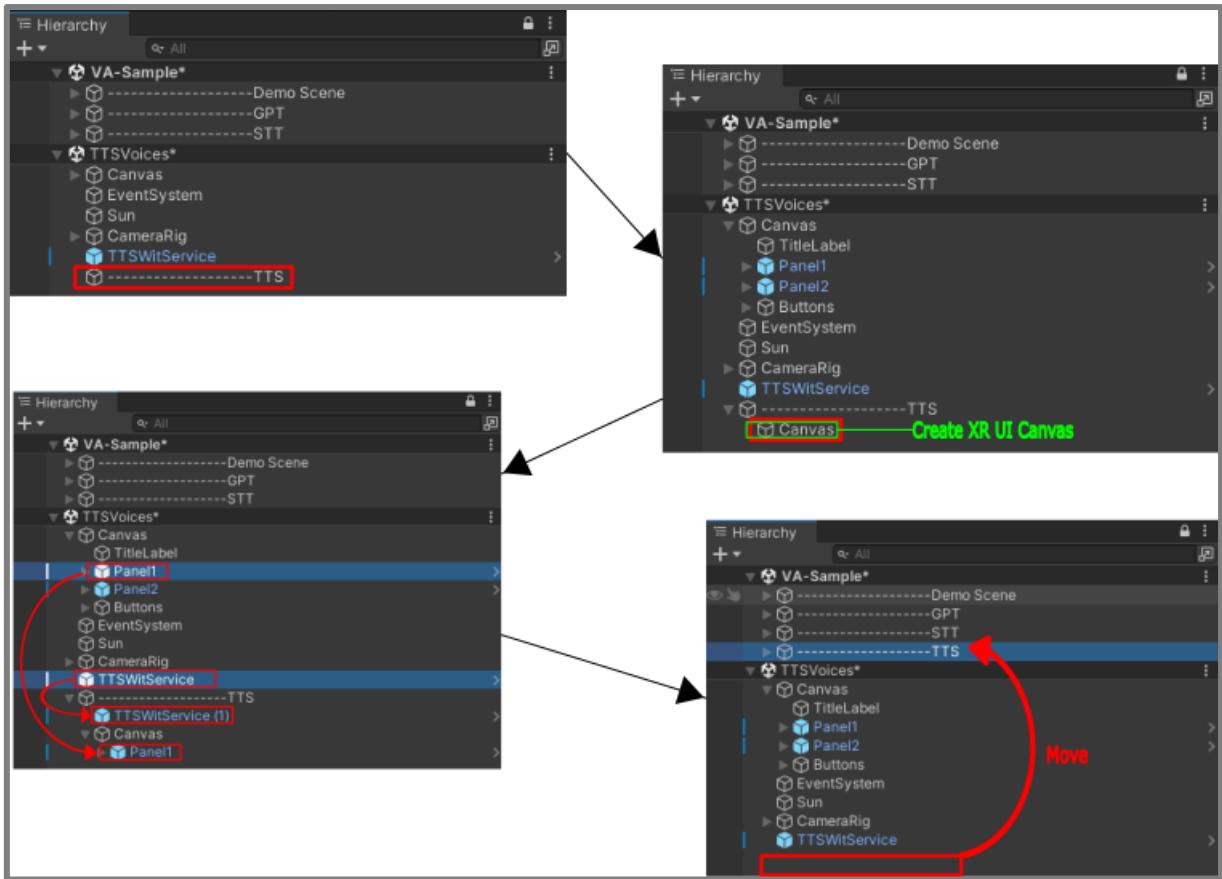


Figure 3.23 – Diagram to prepare TTS.

After modifying the script to use “`InputField`” as the type of the variable “`displayText`”, the `TextField` object from TTS can be dragged and assigned to `displayText`.

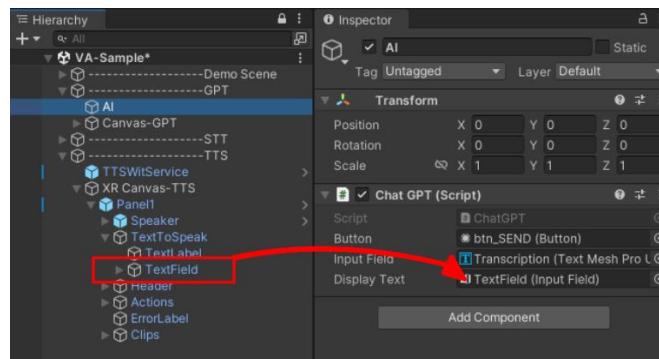


Figure 3.24 – Substituting Display Text in AI object.

In order to remove transcription from last speech, I decided to use the method `Clear()` from the transcription script once the `Speak()` is activated. See Figure 3.25.

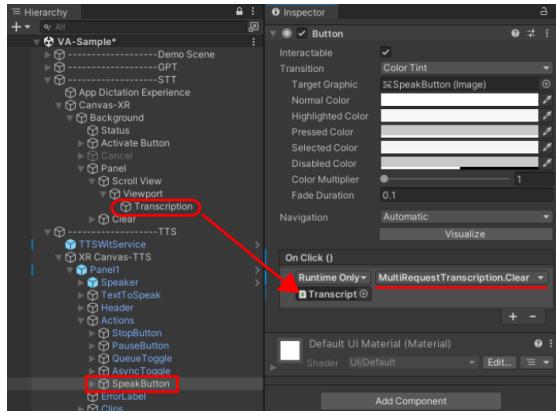


Figure 3.25 – Clear transcription when “Speak” is clicked.

3.2.5 Base of VA

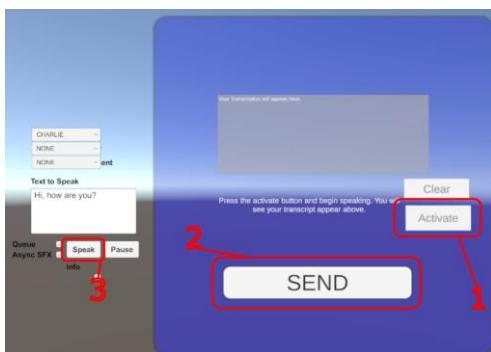


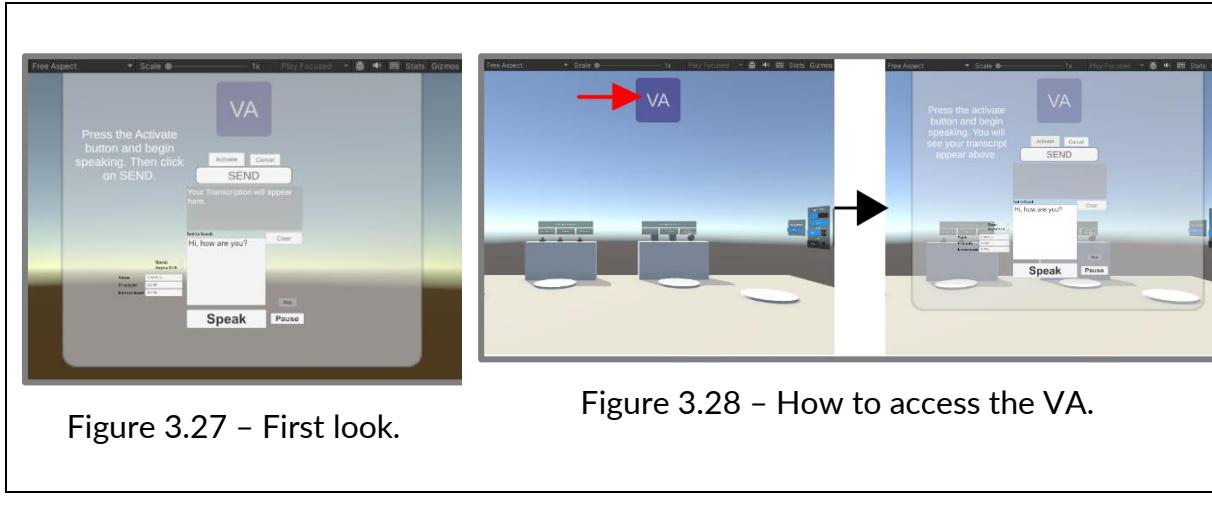
Figure 3.26 – All in. Base of VA.

1. Activate speech-to-text and say something. It will convert your speech into text.
2. Send to OpenAI and receive a response which will appear in the text box on the left.
3. Click on “Speak” and the text in that box will be converted into speech.

Done! This forms the base for the Virtual Assistant. The next steps involve refining the user interface (UI) to make it more user-friendly by removing redundant buttons and hiding unnecessary settings to prevent cluttering the screen with too much information.

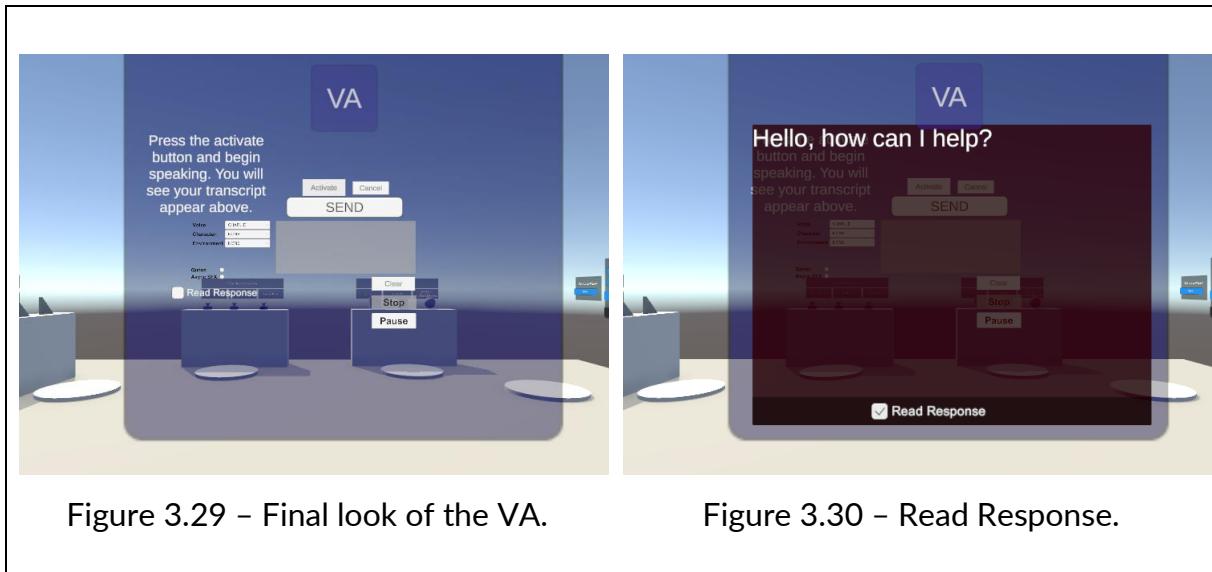
Lastly, I reorganized the objects and concealed the entire VA panel, leaving only a small button visible within the camera's view. By pressing this button, the UI panel of the VA becomes visible. For an enhanced user experience (UX), activating the VA will automatically start the listening mode, eliminating the need for an additional click. Following the same logic, I configured the Speak function to activate whenever there is a change in the text box of the TTS section, unless it is empty or contains only “.....”. This way, the user needs to click only once to send a query to ChatGPT, and as soon as the response arrives, the VA will vocalize it immediately.

Buttons to pause, stop, clear are also present in case the user or the transcription made some mistakes. And the button “Activate” also present to start/stop the dictation again and again.



Final appearance of the Virtual Assistance UI. I have also hidden the response text which can be popped-up once the “Read Response” box is toggled, see Figure 3.30.

To start a new query, just press the Activate button, by pressing again it stops listening or just press SEND, Once the response is ready, the VA will convert it to speech automatically.



When creating a new VWLE, developers can easily access the VA object from this project and position it under their project’s camera. Required packages must be installed and imported if not already present.

3.3 Summary

This chapter outlines the methods used to create and test a Virtual Assistant (VA) in a Virtual World Learning Environment, using the Unity Editor with Oculus Meta Quest 2. It is split into two parts: the first part reviews different approaches that were tried, highlighting both failures and partial successes, especially with the speech-to-text (STT) and text-to-speech (TTS) functionalities.

The second part provides a detailed guide on setting up and configuring the Unity environment to ensure the VA operates smoothly, incorporating OpenAI's APIs to enhance the VA's interactive capabilities. This section also details the successful implementation of TTS and STT within the system.

Together, these segments detail the practical steps and some refinements made during the development process, setting the stage for subsequent evaluations of the virtual assistant's performance in real-world scenarios.

The Unity exported package with scenes and scripts can be found on github¹ link and the full project can be found on Google Drive² link. Note that exported package in github does not include the libraries imported, therefore, it is important to install the assets and packages presented in this methodology.

¹ <https://github.com/University-of-Malta/Final-Year-Project>

² <https://drive.google.com/drive/folders/1Qa0G3K6oBA2Cexo2RQg2qb4CRSv9jy1W?usp=sharing>

4. Evaluation

In this chapter, I describe the evaluation strategy for the final objective **O4**, which aims to assess the usability of the virtual assistant feature across different scenarios.

For this evaluation, I utilized the VA to pose a series of questions, analysing the accuracy of the speech-to-text model's transcriptions.

Concurrently, I evaluated ChatGPT's responses, noting the accuracy of these responses even when transcriptions were incorrect.

Lastly, I examined the text-to-speech model, which was expected to exhibit the fewest errors as it does not depend on the surrounding environment.

Overall, I assessed the VA's performance in various locations using the Oculus. The test phrases included:

1. Hey, what's up?
2. Can you help me with some doubts I have about VR?
3. Why should I study from a virtual reality instead of real world?
4. If I were in a virtual reality dentist office, can you help me if I have some questions?
5. How many teeth humans have? And why?
6. Where is El Dorado?
7. I heard it is close to Portugal, have you ever heard anything about it?
8. Who invented the airplane?
9. Wasn't it Alberto Santos-Dumont?
10. Why would anyone believe that Alberto Santos-Dumont was the first to fly an airplane?

The colours below will aid in interpreting the results in the tables.



Failed



Enough to succeed



Good



perfect

4.1 Silent surrounding

	STT	Response	TTS Correct?	Overall
1.	Hey, Whatsapp!	"Hello, how can I assist you today?"	yes	Good
2.	Equal	"Sure, I'd be happy to help with your VR queries. What would you like to know?"	yes	Perfect
3.	Why should I study for me Victoria reality instead of real world	Assuming you meant "Virtual Reality" (VR), it provides unique immersive learning experiences that can enhance understanding and retention. It's not about choosing VR over the real world, but using VR as a tool to augment real-world learning.	yes	Enough
4.	If I were in the future reality dentist office. Can you help me if I have some questions	Absolutely, I can try my best to provide answers to your questions about a future virtual reality dentist office. What would you like to know?	yes	Enough
5.	How many teeth, .You must have and why	A full set of adult teeth consists of 32 teeth. This includes 8 incisors, 4 canines, 8 premolars, and 12 molars, which includes 4 third molars commonly known as 'wisdom teeth'. Teeth perform important functions like biting, chewing, and peaking clearly.	yes	Enough
6.	Equal	"El Dorado is a legendary city of gold said to be in South America. However, it has never been found, and its precise location remains unknown.	Yes	Perfect
7.	Equal	El Dorado is a mythological city of gold, reportedly in South America, not Europe. It's not linked to Portugal geographically. The legend, however, was pursued by many European explorers including the Portuguese in the 16th and 17th centuries.	yes	Perfect
8.	Equal	The airplane was invented by Orville and Wilbur Wright, also known as the Wright brothers.	yes	Very Good
9.	Wasn't it Alberto Santos demon	Alberto Santos-Dumont was a Brazilian inventor who is known for his significant contribution to the development of airships and airplanes. While he did make several important advances in aviation, the Wright Brothers are generally credited with inventing and flying the first successful airplane.	yes	Good
10.	Why would anyone believe that Alberto Santos Demont was First to fly in airplane	Some believe Alberto Santos Dumont was first because he made his flight in Paris, 1906, where it was public and widely observed, unlike the Wright Brothers' private, less documented flight in 1903.	yes	Good

Table 1 – Performance on silent surrounding.

Based on the 10 examples displayed in Table 1, it was concluded that there is an average of good performance, although improvements in speech-to-text

transcription are necessary. While not all transcriptions were perfectly accurate, those that diverged were still close enough to allow ChatGPT to understand the context of the questions being asked, thereby enabling it to provide concise responses.

4.2 Noisy surrounding

STT		Response	TTS	Overall Correct?
1.	Hey, WhatsApp	I'm an AI and don't use WhatsApp. How can I assist you today?	yes	Enough
2.	Equal	Absolutely! Please ask your questions about VR. I'm here to help	yes	Perfect
3.	Equal	Studying in virtual reality can enable immersive and interactive experiences, potentially improving understanding and retention.	yes	Perfect
4.	Hey dentist can you help me if I have some questions	I'm an AI, not a dentist, but I can answer general dental questions. Go ahead!	yes	Enough
5.	How many teeth humans have in line	Adult humans typically have 32 teeth, with 8 incisors, 4 canines, 8 premolars, and 12 molars	Yes	Enough
6.	Equal	El Dorado is a mythical city of gold, long sought by explorers, believed to be in south America. Its exact location remains unknown.	Yes	Perfect
7.	Equal	As far as historical records and myths are concerned, El Dorado is not near Portugal. It's associated with South America.	Yes	Perfect
8.	Equal	The Wright Brothers, Orville and Wilbur, are credited with inventing the first successful airplane in 1903.	Yes	Perfect
9.	Wasn't it Alberto Santos the month	Alberto Santos-Dumont was a pioneering Brazilian aviator, but the invention of the first successful airplane is credited to the Wright Brothers	Yes	Good
10.	Why would anyone believe that .Alberto Santos Demont was the first to fly in airplane	Alberto Santos-Dumont made significant contributions to aviation and his flights in Europe were well-documented and public, leading some to argue that he was the first to fly. The Wright Brothers' initial flight in the US were less publicized at the time.	Yes	Good

Table 2 - Performance on noisy surrounding.

To conduct this test, questions were asked while loud music played in the background. Both evaluations initially misunderstood the phrase 'What's up,' transcribing it as 'Whatapp'. However, in a silent environment, the speech-to-text (STT) technology transcribed it as 'Whatsapp,' and in the second evaluation, it was transcribed as 'WhatsApp' with a capital 'A', which altered the response from ChatGPT. Despite this, the response was sufficient to continue the conversation,

since it ended with 'How can I assist you today?' For the remaining sentences, the performance was overall consistent with test 4.1.

4.3 Outside

STT		Response	TTS Correct?	Overall
1.	Equal	I'm here to assist you, how can I help you today	Yes	Perfect
2.	Equal	of course I'd be happy to help with your VR questions, what would you like to know	Yes	Perfect
3.	Equal	studying in virtual reality can be interactive and immersive making learning more engaging it can also simulate experiences and environments that may not be acceptable in the real world	Yes	Perfect
4.	Equal	absolutely I can help answer any questions you might have while in a virtual dentist office, what do you need to know	Yes	Perfect
5.	Equal	adult humans typically have 32 teeth including incisors canines premolars and molars they are used for biting tearing grinding and chewing food aiding in speech and maintaining facial structure	Yes	Perfect
6.	Equal	El Dorado is a legendary city of gold said to be in South America but its exact location remains unknown as it's a myth and has never become	Yes	Perfect
7.	Equal	El Dorado is a mythical city and there is no concrete evidence of its existence or location it's not associated with Portugal it's historically tied to legends from South America	Yes	Perfect
8.	Equal	the airplane was invented by Orville and Wilbur Wright also known as the Wright Brothers in 1903	Yes	Perfect
9.	Wasn't it Alberto Santos Demont	Alberto Santos-Dumont was a significant figure in the development of aviation. However, the Wright brothers are credited with inventing and flying the first airplane. Santos-Dumont made important contributions especially in regard to controlled flights in Europe	Yes	Good
10.	Why would anyone believe that Alberto Santos demont was the first to fly an airplane?	Alberto Santos-Dumont is revered in Brazil as the first to fly an airplane because his flights were public, well documented and met the criteria of the Fédération Aéronautique Internationale at that time. The Wright Brothers' flights were done in relative privacy which contributes to this perception.	Yes	Good

Table 3 - Performance Outside.

For this evaluation, I have performed the same test as the previous two, but this time in a busy outside environment. To be more precise, it was located beside the roundabout in Wiji, Malta. See Figure 4.1. This was conducted on a Monday at 5 PM, specifically chosen during rush hour to capture as much ambient noise as possible.

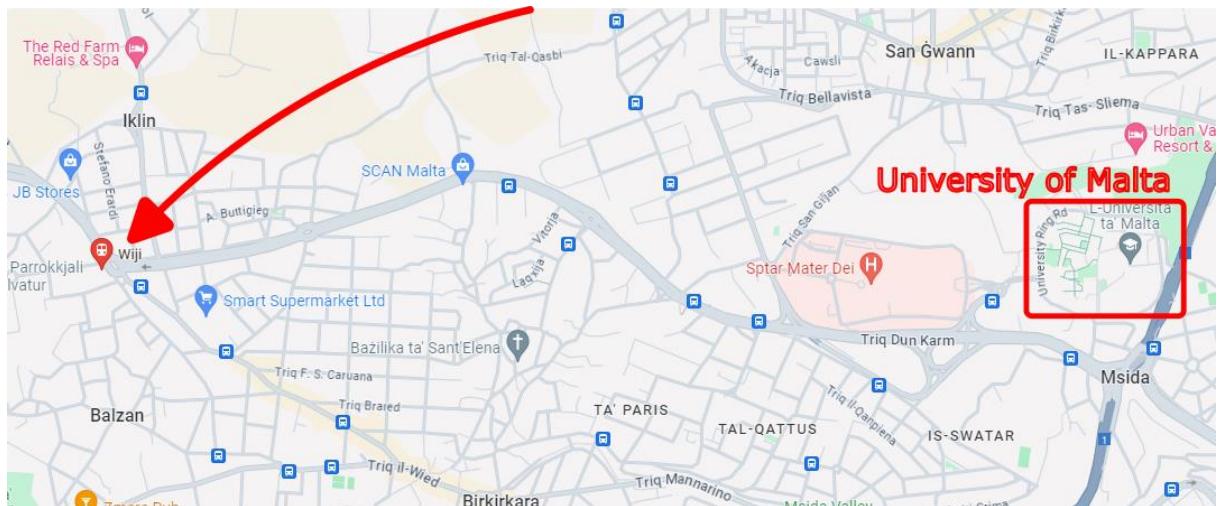


Figure 4.1 – Location for Evaluation 4.3.

Surprisingly, the performance in this evaluation was superior to the previous two. The noisier the environment, the better the transcription results were. It's important to note that it wasn't flawless, primarily due to an unusual surname, a mistake that might have occurred even with human transcription. However, using VR in such a noisy outdoor environment might not be comfortable. Nonetheless, if the user finds this acceptable, the virtual assistant will function impeccably. Furthermore, I believe that Augmented Reality (AR) would be more suitable for outdoor use. Should this virtual assistant be implemented in an AR application, this test indicates that it would likely perform well.

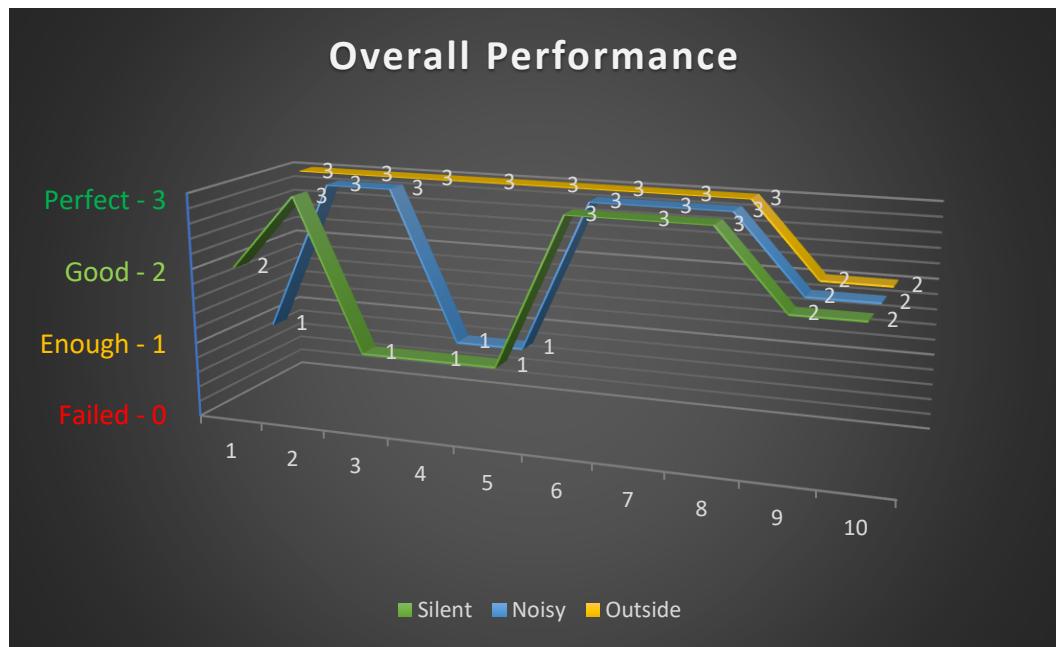


Figure 4.2 – Graph overall performance.

4.4 Summary

In this chapter, the usability of the virtual assistant was rigorously tested across different environments to assess its performance under various conditions. The VA was evaluated by asking a series of predetermined questions, with the accuracy of the speech-to-text (STT) transcriptions and the appropriateness of ChatGPT's responses being critically analysed. The text-to-speech (TTS) functionality was also tested, as it is crucial for the VA's interaction with users.

The evaluation was structured into three distinct settings: silent, noisy, and outdoor environments. Results showed that while the VA performed well in silent and moderately noisy conditions, its functionality improved significantly in highly noisy outdoor settings. This was illustrated by the graph in Figure 4.2, which depicted the VA's performance across these scenarios. Despite some challenges with unusual vocabulary or names, the VA generally provided accurate and contextually appropriate responses.

Overall, the evaluation confirmed that the VA could effectively handle real-world applications, with its performance being robust even in challenging environments. These findings suggest that the VA could be particularly effective in outdoor or noisy settings, which indicates good prospects for potential augmented reality (AR) applications.

5. Conclusions

From the evaluations conducted, it is evident that the virtual assistant developed performs exceptionally well, exceeding expectations, particularly in noisy environments during rush hours, a finding that was both unexpected and insightful. This capability underscores the robustness of the speech-to-text and text-to-speech technologies employed, confirming that the virtual assistant facilitates efficient access to relevant information. Indeed, the presence of such technology can significantly enhance engagement and participation in learning processes. However, it is crucial for users to remain focused and not divert their attention with irrelevant inquiries.

Looking ahead, several enhancements and areas of research emerge. Currently, the ChatGPT model, integral to this virtual assistant, shows limitations in answering real-time, context-specific questions, such as those about the weather or current events; due to the static nature of its training data, incorporating commands that allow real-time updates or integrating other functionalities from wit.ai API may provide a viable workaround. Additionally, expanding the virtual assistant's capabilities to include interacting with images through the implementation of the DALL-E API could enrich virtual learning environments, offering even more dynamic and engaging user experience.

Refining the user interface further and enhancing the user experience (UX) are essential, with potential contributions from experienced designers to elevate the system's usability. Moreover, broadening the testing framework to include users from diverse linguistic backgrounds could ensure the virtual assistant's utility across various accents and dialects, enhancing its inclusiveness and accessibility.

By advancing these areas, the functionality and applicability of virtual assistants in educational settings can continue be improved, making these tools even more effective and responsive to the needs of a global user base.

5.1 Revisiting the Objectives

As we conclude this study, it's important to review the objectives outlined at the start to evaluate their fulfilment:

Objective 1: Conduct thorough research into AI technologies that can be integrated into virtual learning environments, focusing on transformer-based generative models, speech-to-text, and text-to-speech technologies. Throughout the dissertation, comprehensive research was conducted and applied, enhancing the functionality and responsiveness of the virtual assistant within the virtual learning environment. This objective was met as the foundational AI technologies were thoroughly explored and integrated, proving critical for the development of the virtual assistant. However, I believe that OpenAI deserves a greater literature review since it contains many features for the APIs which were not explored in this project.

Objective 2: Develop a flexible and adaptable virtual reality (VR) framework for deploying the virtual assistant across various VR platforms. The framework created was both modular and adaptable, detailed in the Methodology chapter, which allowed the virtual assistant to integrate smoothly into different VR environments. This goal was successfully met using a sample environment which contained the necessary configurations for the basic use of a VR. With more time, one could create an actual learning environment to better demonstrate the VA.

Objective 3: Build a prototype of the virtual assistant that can understand and respond to user queries in real time. The prototype tested showed effective real-time interaction, enhancing engagement and facilitating the search for information. This confirms that the prototype met its intended purpose. However, it is important to highlight that this VA was developed using Meta Packages and was not tested with different VR devices.

Objective 4: Evaluate the virtual assistant's quality and efficiency in a VR environment. Through rigorous testing in various settings, the virtual assistant proved to enhance users' ability to quickly and effectively access relevant information, thus improving the overall experience in VR.

In summary, all objectives were met with considerable success, demonstrating the potential and efficacy of virtual assistants in educational environments. Future efforts will aim to expand the virtual assistant's capabilities and improve its user interactions, as suggested by the enhancements discussed earlier.

References

- [1] "(248) Clippy Office Assistant Animation Compilation, Clippit - YouTube." Accessed: May 04, 2024. [Online]. Available: https://www.youtube.com/watch?v=c2gp8mWXNoE&ab_channel=JohnDaniels
- [2] G. B. Leonard, *Education and ecstasy*. Delacorte P., 1968.
- [3] X. Ding and Z. Li, "A review of the application of virtual reality technology in higher education based on Web of Science literature data as an example," *Front Educ (Lausanne)*, vol. 7, Nov. 2022, doi: 10.3389/feduc.2022.1048816.
- [4] S. S. Oyelere, N. Bouali, R. Kaliisa, G. Obaido, A. A. Yunusa, and E. R. Jimoh, "Exploring the trends of educational virtual reality games: a systematic review of empirical studies," *Smart Learning Environments*, vol. 7, no. 1, p. 31, Dec. 2020, doi: 10.1186/s40561-020-00142-7.
- [5] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull Math Biophys*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.
- [6] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol Rev*, vol. 65, no. 6, pp. 386–408, 1958, doi: 10.1037/h0042519.
- [7] "What are Neural Networks? | IBM." Accessed: Dec. 16, 2023. [Online]. Available: <https://www.ibm.com/topics/neural-networks>
- [8] C. C. Aggarwal and others, "Neural networks and deep learning," *Springer*, vol. 10, no. 978, p. 3, 2018.
- [9] G. Hinton *et al.*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Process Mag*, vol. 29, no. 6, pp. 82–97, Nov. 2012, doi: 10.1109/MSP.2012.2205597.
- [10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [11] "Supervised vs. Unsupervised Learning: What's the Difference? - IBM Blog." Accessed: Dec. 12, 2023. [Online]. Available: <https://www.ibm.com/blog/supervised-vs-unsupervised-learning/>
- [12] "What is Natural Language Processing? | IBM." Accessed: Dec. 10, 2023. [Online]. Available: <https://www.ibm.com/topics/natural-language-processing>
- [13] K. S. Jones, "Natural Language Processing: A Historical Review," in *Current Issues in Computational Linguistics: In Honour of Don Walker*, Dordrecht: Springer Netherlands, 1994, pp. 3–16. doi: 10.1007/978-0-585-35958-8_1.
- [14] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989, doi: 10.1109/5.18626.
- [15] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. 6, pp. 1137–1155, 2003.
- [16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct. 2018.
- [17] T. B. Brown *et al.*, "Language Models are Few-Shot Learners," May 2020.
- [18] A. Vaswani *et al.*, "Attention Is All You Need," Jun. 2017.

- [19] “About.” Accessed: Dec. 12, 2023. [Online]. Available: <https://openai.com/about>
- [20] “Product.” Accessed: Dec. 12, 2023. [Online]. Available: <https://openai.com/product>
- [21] “ChatGPT.” Accessed: Dec. 12, 2023. [Online]. Available: <https://openai.com/chatgpt>
- [22] “Rate limits - OpenAI API.” Accessed: May 28, 2024. [Online]. Available: <https://platform.openai.com/docs/guides/rate-limits/usage-tiers?context=tier-free>
- [23] A. Dev, S. S. Agrawal, and A. Sharma, Eds., *Artificial Intelligence and Speech Technology*, vol. 1546. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-95711-7.
- [24] “What is Speech to Text? - Speech to Text Explained - AWS.” Accessed: Dec. 16, 2023. [Online]. Available: <https://aws.amazon.com/what-is/speech-to-text/>
- [25] K. Sharma and S. R. N. Reddy, “Spectrogram Analysis and Text Conversion of Sound Signal for Query Generation to Give Input to Audio Input Device,” 2022, pp. 171–181. doi: 10.1007/978-3-030-95711-7_15.
- [26] “What is Text to Speech? - TTS Explained - AWS.” Accessed: Dec. 16, 2023. [Online]. Available: <https://aws.amazon.com/polly/what-is/text-to-speech/>
- [27] “Text-to-Speech AI: Lifelike Speech Synthesis | Google Cloud.” Accessed: Dec. 17, 2023. [Online]. Available: <https://cloud.google.com/text-to-speech?hl=en>
- [28] “Unity - Developing Your First Game with Unity and C# | Microsoft Learn.” Accessed: Dec. 17, 2023. [Online]. Available: <https://learn.microsoft.com/en-us/archive/msdn-magazine/2014/august/unity-developing-your-first-game-with-unity-and-csharp>
- [29] “Build VR experiences with Unity.” Accessed: Dec. 17, 2023. [Online]. Available: <https://unity.com/solutions/vr>
- [30] “(262) Making ChatGPT for Android using Unity! - YouTube.” Accessed: May 09, 2024. [Online]. Available: https://www.youtube.com/watch?v=6I3gFp_WXPs&list=PL4g4CxkYXn3tugt8eXqz_9yBKcTXtYaeh&index=8&ab_channel=immersiveinsiders
- [31] “(316) Using Text to Speech with ChatGPT! - YouTube.” Accessed: May 29, 2024. [Online]. Available: https://www.youtube.com/watch?v=321h9wDT-gM&ab_channel=immersiveinsiders
- [32] “(253) Beginner’s Guide to Unity: Building Unity Projects for Oculus Quest 2 - YouTube.” Accessed: May 06, 2024. [Online]. Available: https://www.youtube.com/watch?v=NWBc0y95obc&ab_channel=CircuitStream
- [33] “OpenAI | com.openai.unity | Unity Package (UPM) | OpenUPM.” Accessed: May 06, 2024. [Online]. Available: <https://openupm.com/packages/com.openai.unity/#completions>
- [34] “Meta XR Voice SDK - Dictation | Integration | Unity Asset Store.” Accessed: May 09, 2024. [Online]. Available: <https://assetstore.unity.com/packages/tools/integration/meta-voice-sdk-dictation-265013>

Appendix A

A.1 ChatGPT communication script

```
using System;
using System.Collections;
using System.Collections.Generic;
using NUnit.Framework;
using UnityEngine;
using UnityEngine.UI;

using Button = UnityEngine.UI.Button;
using OpenAI;
using OpenAI.Chat;
using OpenAI.Models;
using TMPro;

/* 6 asset usages
public class ChatGPT : MonoBehaviour
{
    [SerializeField] private Button button; // Button SEND *Changed in 2 assets
    [SerializeField] private TMP_Text inputField; // Text from STT *Changed in 2 assets
    [SerializeField] public TMP_Text readResponse; // For Text panel *Content (TextMeshProUGUI)
    [SerializeField] private InputField displayText; // For TTS Text field (Hidden) *Changed in 2 assets
    [Multiline]
    [SerializeField] private string _role = String.Empty; // To describe VA's role *Unchanged
    [SerializeField] private string API_Key = String.Empty; // "sk-RQffy559Z...<>...XTjT93H6RQcai"

    private string userInput; // To store text from inputField
    private List<Message> chatHistory = new List<Message>(); // Sabe all tokens
    private OpenAIClient api; // OpenAI library
}

private void Start()
{
    api = new OpenAIClient(new OpenAIAuthentication(API_Key)); // Configure api passing the key
    button.onClick.AddListener(AskAI); // Read when SEND btn in Clicked

    inputField.text = _role; // Send _role on first call to AskAI()
    AskAI();
}

private async void AskAI()
{
    button.enabled = false; // Disable SEND button while no response from last SEND pressed
    inputField.enabled = false; // Avoid new text input during communication with OpenAI API

    userInput = inputField.text; // Retrieve Text from STT text box
    chatHistory.Add(item: new Message(Role.User, userInput)); // Append text into chatHistory

    displayText.text = "....."; // This will be in the "Read Response panel"
    inputField.text = ""; // as well as into TTS text field

    var chatRequest = new ChatRequest(chatHistory, (string)Model.GPT3_5_Turbo ); // Create a chatRequest variable with
        // chatHistory and GPT Model to use

    var result:ChatResponse = await api.ChatEndpoint.GetCompletionAsync(chatRequest); // Retrieve response by
        // sending chatRequest

    displayText.text = result.Choices[0].Message.ToString(); // Add text into TTS text field
    readResponse.text = displayText.text; // Add text into Read Response Panel

    chatHistory.Add(item: new Message(Role.Assistant, displayText.text)); // Append GPT response into chatHistory
    DisplayMessages(chatHistory); //Method created For debugging in Unity console

    button.enabled = true; // Enable back the features
    inputField.enabled = true;
}
```

A.2 Links to this project resources

Github

<https://github.com/University-of-Malta/Final-Year-Project>

Google Drive (Contains complete Unity project)

[https://drive.google.com/drive/folders/1Qa0G3K6oBA2Cexo2RQg2qb4CRSv9jy1W
?usp=sharing](https://drive.google.com/drive/folders/1Qa0G3K6oBA2Cexo2RQg2qb4CRSv9jy1W?usp=sharing)

Appendix B Outdoor evaluation

