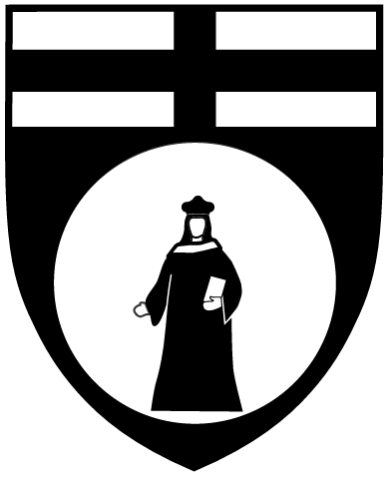

Dibris Dipartimento di Informatica, Bioingegneria,
Robotica e Ingegneria dei Sistemi

Virtual Reality for Robotics - a.y. 2023-2024

Assignment 3

AI NPC: Artificial Intelligence applied to a Non-Playable Character

---

Nicholas Attolino

UNIVERSITY OF GENOA, NOVEMBER 2024

Contents

1 Introduction **3**

2 Integrate AI Characters with MetaHuman Avatars in Unreal Engine 5 **4**

2.1 Setting Up the ConvAI Plugin 4
2.2 Adding MetaHumans to Scene 5
2.3 Adding Animations 6
2.4 Creating Character AI Personality 8
2.5 Enabling Actions 10
2.6 Scene Perception 12

3 Conversation with the Character **13**

3.1 Push-to-Talk 13
3.2 Goal of the project 14
3.3 Flip-Flop 14
3.4 Button-less Communication 15

4 Results **16**

5 Conclusion and Future prospects **16**

1 Introduction

A non-player character (NPC), also called a non-playable character, is a character in a game that is not controlled by a player. The origin of this term comes from board games such as " Dungeons & Dragons " as it referred to characters managed by the GameMaster and not by other players[3]. Within video games, an NPC is a character managed by the computer and involves a number of behaviours that will influence the gaming experience but will not necessarily be the product of true artificial intelligence[6]. With the arrival of LLMs, video game companies are working on implementing them to NPCs to allow them to behave more like humans, consequently more interactive; an example could be through verbal communication or through facial or behavioural expressions based on user interaction[4]. Thus the idea for this project was born, namely to realise an NPC capable of communicating with the user through the use of LLMs. The development environment was Unreal Engine 5.4.4 [2] and the plug-ins used were *AI for NPC, MetaHuman - Dialog, actions and general intelligence - by ConvAI* and *MetaHumans on QuixelBridge* [5].

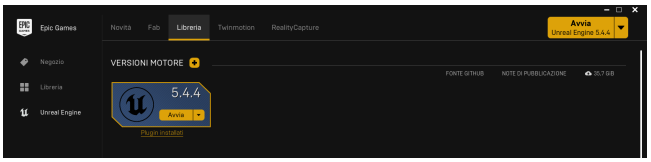

Figure 1: Unreal Engine 5.4.4 as environment

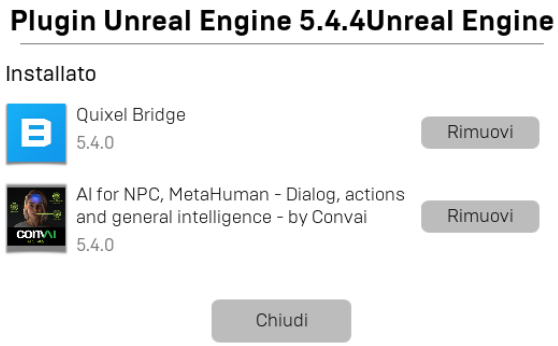

Figure 2: ConvAI Plug-in and MetaHumans on QuixelBridge

2 Integrate AI Characters with MetaHuman Avatars in Unreal Engine 5

2.1 Setting Up the ConvAI Plugin

Initially, the ConvAI Plug-in was set up by downloading and installing it from the EpicGames launcher marketplace:

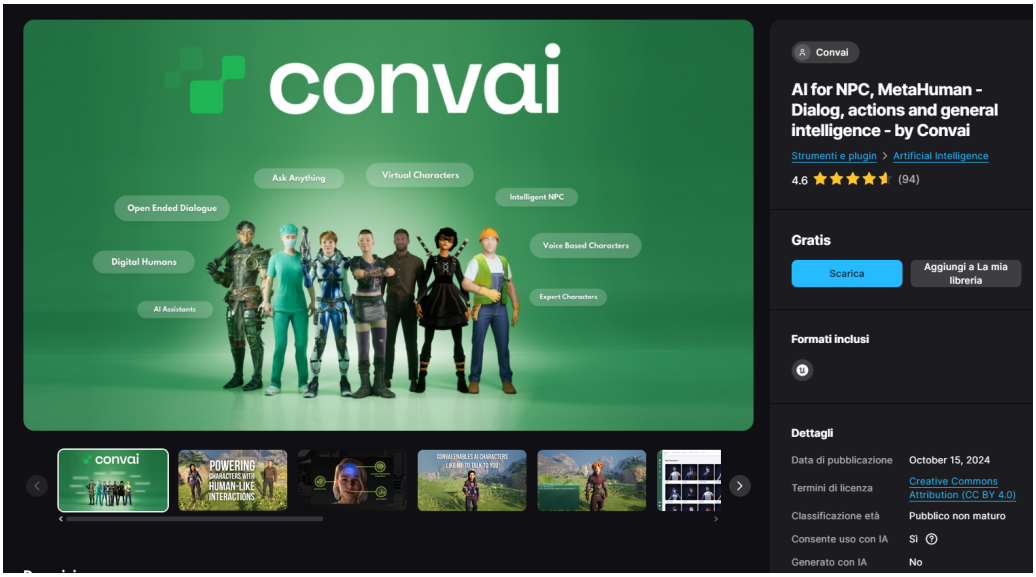

Figure 3: ConvAI Plug-in

Once this was done, the plug-in was enabled by selecting Plugins from the Edit menu, searching for ConvAI and ticking the box:

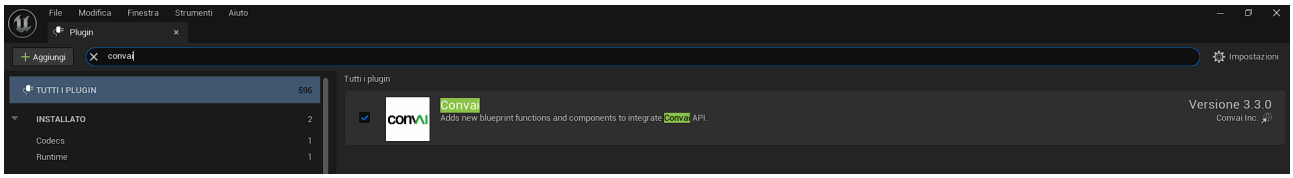

Figure 4: Edit -> Plugin -> search "ConvAI"

2.2 Adding MetaHumans to Scene

A MetaHuman character was added from QuixelBridge by selecting it from the relevant menu:

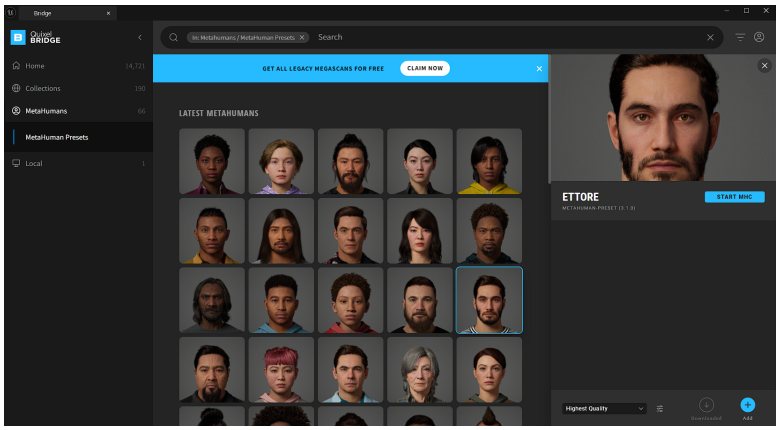

Figure 5: Window -> QuixelBridge -> MetaHumans -> Add after downloaded

Inside the Content folder will appear the MetaHumans folder, which in turn will contain the character that simply needs to be picked up and dragged into the environment to add it:

Figure 6: Content -> MetaHumans -> Drag and Drop Blueprint Ettore into the environment

2.3 Adding Animations

By opening the character's blueprint, animations were added to the body and face by selecting the respective AnimClass provided by ConvAI:

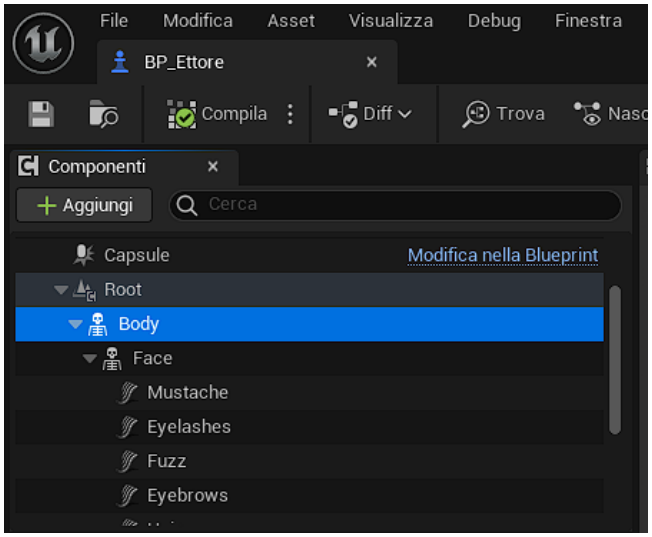

Figure 7: Components Panel

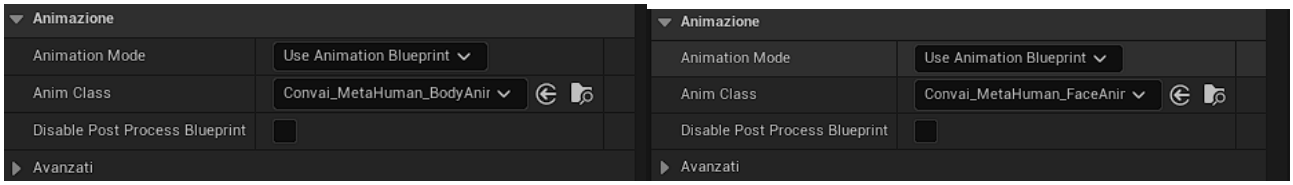

Figure 8: Details Panel -> Anim Class -> Type "ConvAI" and Click rispectively "Convai MetaHuman BodyAnim" and "Convai MetaHuman FaceAnim"

It is also important to change the parent class to ConvAI Base Character:

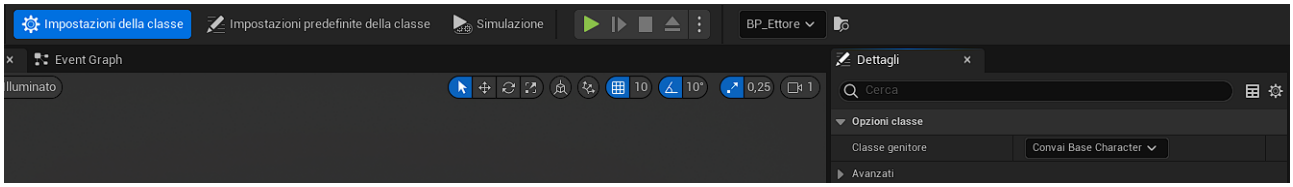

Figure 9: Class Settings -> Details Panel -> Class Parent -> Select "ConvAI Base Character"

ConvAI Base Character is a simple blueprint containing a few functions designed to make the development process a bit faster.

Then the Lip Sync component was added by selecting ConvaiFaceSync from the component panel:

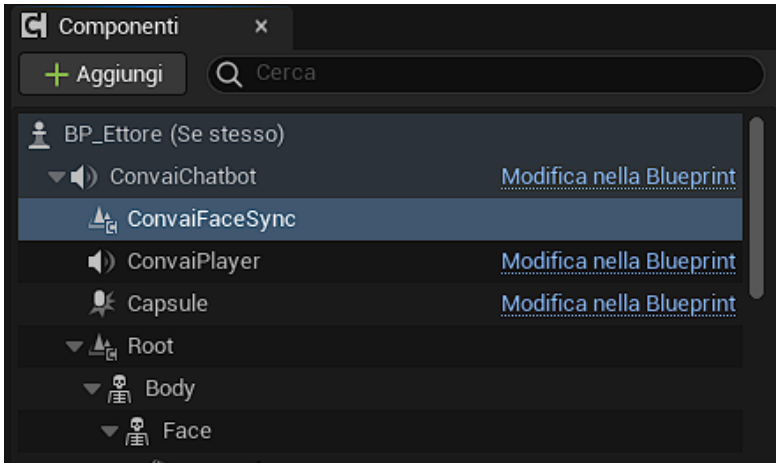

Figure 10: Components Panel -> Add component -> Type "ConvAI" -> Select "ConvAI Face Sync"

Now that the Character has been prepared, assuming that you are working on a project where the player is in the first person (this is the case dealt with, however any other type is also possible), you need to change the parent class of the blueprint of the player to ConvAI Base Player:

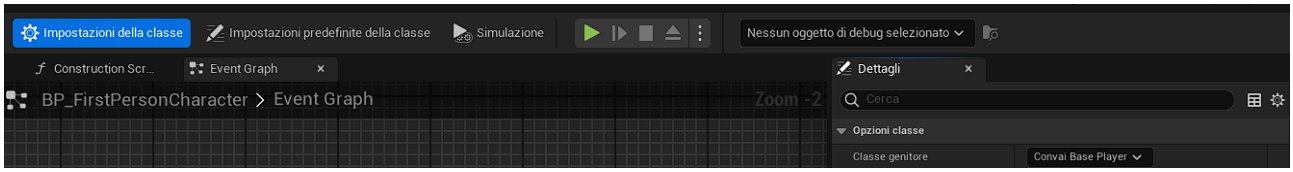

Figure 11: Contents Panel -> First Person -> Blueprint First Person -> Class Settings -> Class Parent -> Type "ConvAI" -> Select "ConvAI Base Player"

2.4 Creating Character AI Personality

The next step is to create an AI personality for the character, and to do this you need to create an account on the ConvAI website[1]. Once you have created your account and logged in, you can see the characters available:

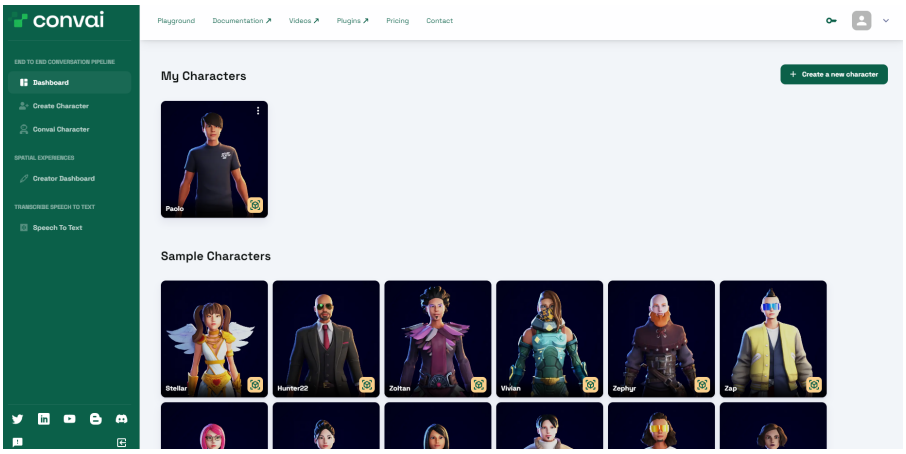

Figure 12: ConvAI Dashboard

Before creating the character, you must copy your personal API key and enter it into Unreal Engine:

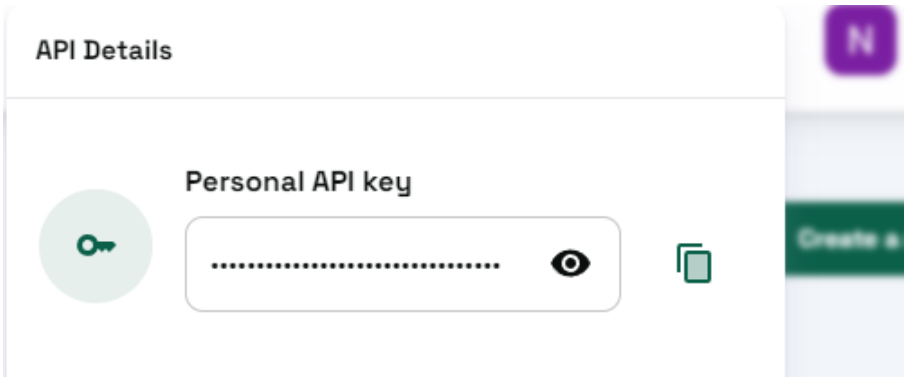

Figure 13: Personal API Key

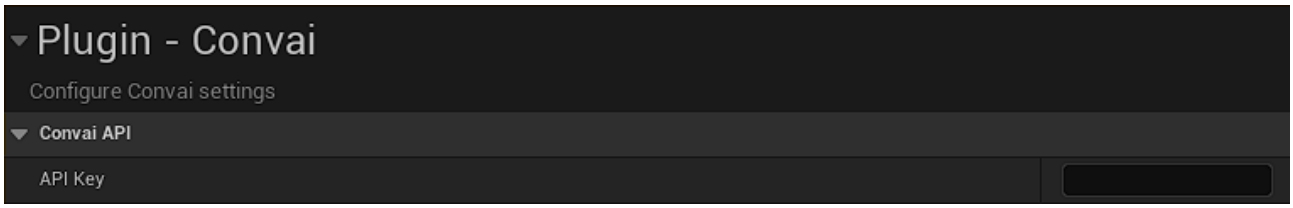

Figure 14: Edit -> Project Settings -> Scroll down until ConvAI -> Click on it and put in the API key

Moving on to character creation, in the new character description you can see a field to enter a name; below you can enter a backstory which describes the personality or characteristics of the character. The next step is to choose a voice for the character: on top of the list there are voices from ConvAI which are fast response voices that have low latency; there are several GCP voices, Azure voices, OpenAI voices and Elevenlabs voices (these latter are not available in the free plan). After the choice, click on " Create Character " and, as you can see, you can test the backstory by talking to the character on the site itself.

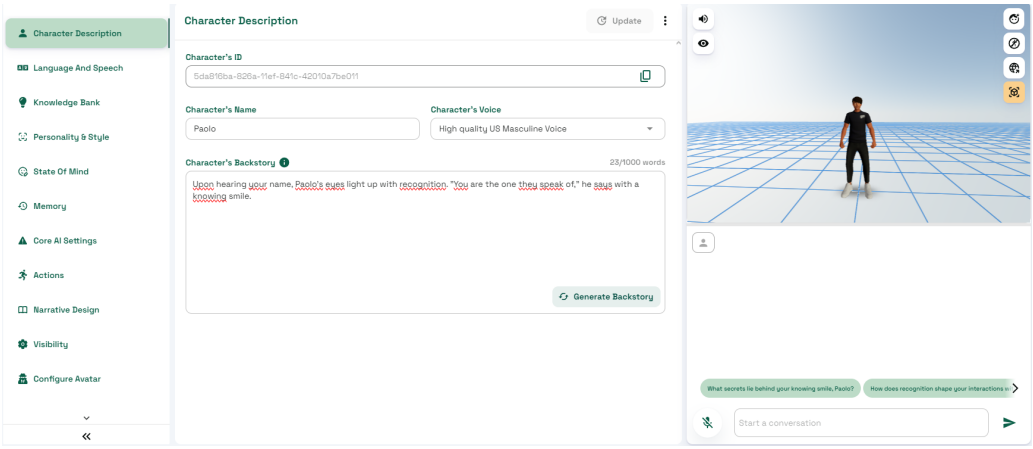

Figure 15: Character created

It is also possible to copy the character ID and insert it into Unreal Engine:

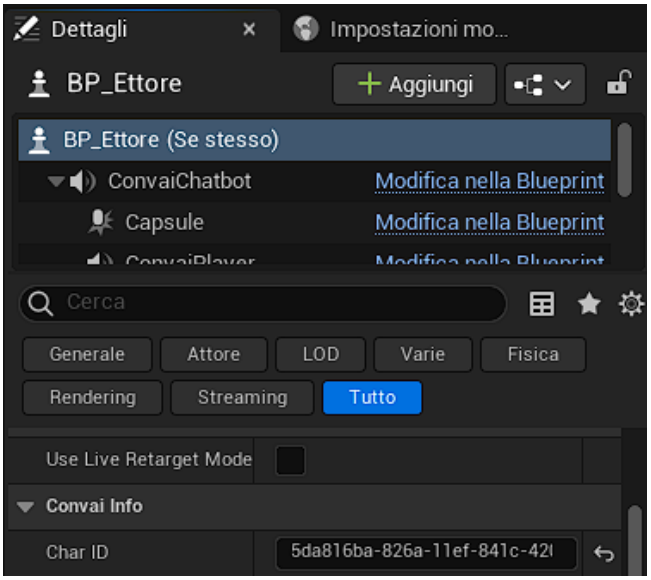

Figure 16: Character's ID into Character's ID on Unreal Engine

2.5 Enabling Actions

By default, each character has certain actions available:

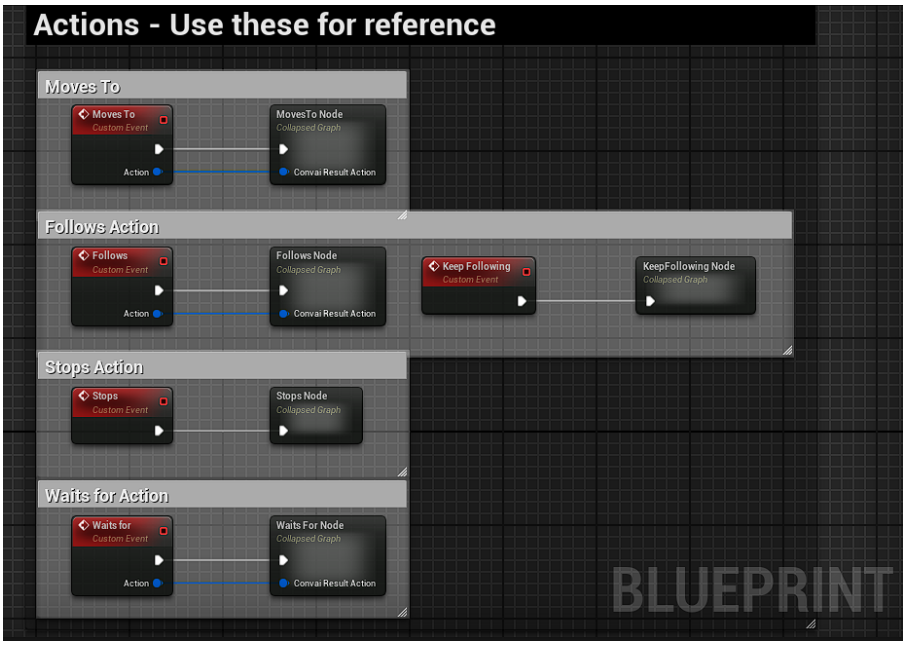

Figure 17: Default actions for each character

The following were tested in this case:

- **Follow**, to follow the User;
- **Move**, to send the character towards an object in the environment and come back;
- **Stop**, to end the follow movement.

To allow movement in the environment, a Navigation Mesh must be implemented and to do so, the following steps must be followed:

> Window -> Place Actors -> search "Nav" -> Drag and Drop "Nav Mesh Bounds Volume" into the scene -> Make it big enough so that it covers the whole area (in this case you can see shaded areas which indicate where the character can move around)

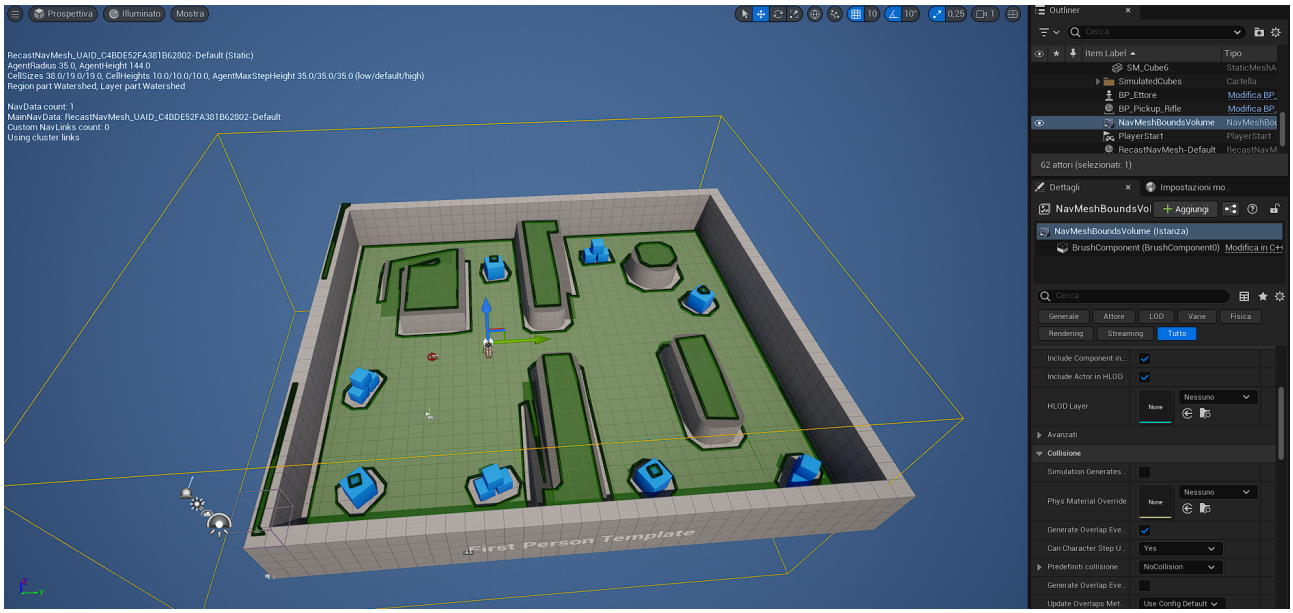

Figure 18: Nav Mesh implemented

2.6 Scene Perception

It is also possible to allow the character to interact with objects in the environment:

> Blueprint Ettore -> Scroll down details panel until ConvAI Info -> Objects -> Click "+" to add a new object -> Use the eyedropper to select an object into the scene and then give it a name (is also possible to give a description to provide a more contextual knowledge for the character)

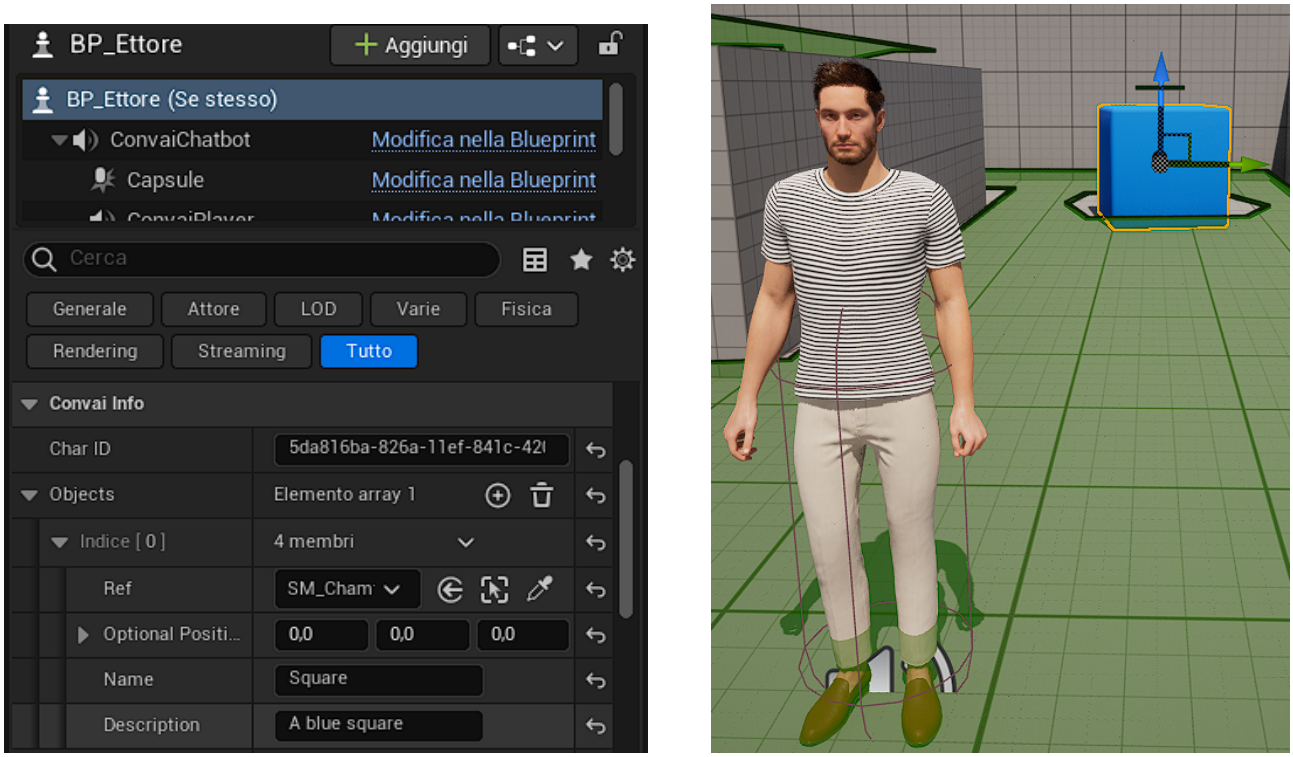

Figure 19: Set as blue square object that one behind the character

If there is interest in the implementation but you encounter problems following this description, you can follow the tutorial or you can read the documentation.

3 Conversation with the Character

3.1 Push-to-Talk

In the ConvAI plugin, Push-to-Talk is the only method of communication and is managed by pressing the T key on the keyboard: when pressed and held, it starts communication from the user to the character; when released, it stops communication from the user, allowing the character to process a reply and consequently respond. In addition, the communication is noted in a chat in the bottom left-hand corner.

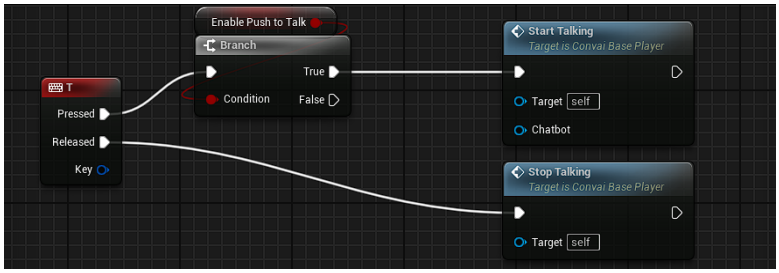

Figure 20: Push-to-Talk nodes

Figure 21: Communication into the Chatbox

3.2 Goal of the project

The goal of this project was to create a more "natural" communication between the User and the character, and to do so it was necessary to create a communication method different from Push-to-Talk. Initially, it was thought to use a "sphere collision" that, when the User enters the range of action, starts the communication; however, there was no way to create something that would allow the User to end the communication to the character (i.e. something that would indicate that the user has finished speaking so that the character would process the message and respond accordingly) because it was discovered that the plug-in in the source code is based only and exclusively on push-to-talk so this choice was not the best.

3.3 Flip-Flop

Subsequently, we thought of using a Flip-Flop node to roughly simulate push-to-talk, also allowing the use of a button in the field of view, assuming that the simulation takes place in Virtual Reality (of course, it is also possible to set any key of the Virtual Reality commands as the trigger key). With the Flip-Flop, communication occurs by pressing a single button both to start and to end it, and thus makes it more "natural" than that via push-to-talk. To use the Flip-Flop communication is important to set a new Input key:

> Settings -> Project Settings -> Scroll down until "Input" -> Click on "+" next to "Action Mapping" to add a new action that represents the new Input key (in this case the action was renamed "Start Stop talking with chatbot" and set "G" as Input key)

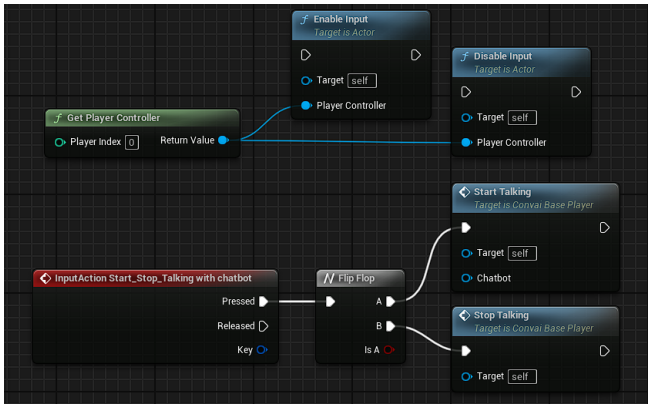

Figure 22: Communication with Flip-Flop

Figure 23: Setting the Input key for Flip-Flop

3.4 Button-less Communication

As a final implementation there was a completely trigger-free communication method via keys, which uses the recognition of audio coming from the microphone (Thanks to 'IRANpower' of ConvAI community on Discord for the help):

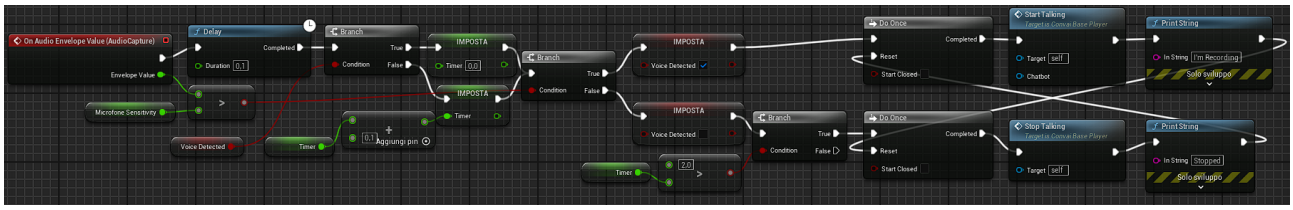

Figure 24: Communication with only sound from microphone

How it works: when the user starts speaking, the sound triggers an event involving an audio value from the microphone that is synced with a delay and sets a timer when it detects the voice. It then starts the communication and prints the string "I'm recording" to confirm that the User is speaking and the character is listening; when the user stops speaking, it checks within a certain timer that there is no more sound so that it can end the communication from the User and print the string "Stopped" to confirm that the User has stopped speaking and allow the character to process and respond.

4 Results

The results obtained were excellent:

- **Communication with Flip-Flop** represents the most precise version as a communication method because, simulating the push-to-talk via physical button or button in the field of view, it is capable of correctly performing the communication between User and character;
- **Button-less communication**, an issue was found regarding the correct recognition of the message from the User to the character, that is, during operation, the first word pronounced by the User is never interpreted due to the underlying system on which the communication in the ConvAI plugin is based since it considers the first word as the trigger of the event to start the communication with the character.

5 Conclusion and Future prospects

In this project it was seen how to implement artificial intelligence on an NPC (Non-Playable Character) through the use of:

- ConvAI plug-in
- QuixelBridge plug-in
- Unreal Engine 5.4.4 as a development environment for this project

Different methods of communication between User and NPC were tested, resulting in the most precise being the communication with Flip-Flop while the most suitable for the project's goal was the communication without buttons despite the problems encountered in its operation due to a deeper logic due to the ConAI plug-in.

As for future prospects, it could be possible to modify the deeper logic that manages communication in the ConvAI plug-in to solve the problem encountered in the communication without buttons or directly wait for progress from ConvAI on their entire plug-in in the hope that the system receives an update that improves the communication part; furthermore it could be interesting to implement additional actions for the characters, for example custom actions such as dancing and gesticulating or even interacting more with the environment surrounding the NPC.

References

[1] ConvAI. *AI for NPC, MetaHuman - Dialog, actions and general intelligence - by ConvAI*. [online] Available: https://convai.com.

[2] Epic Games. *Unreal Engine 5.4.4*. [online] Available: https://www.unrealengine.com/en-US/download, Published in June 2024.

[3] Wizard of the Coast. *Dungeons and Dragons*. [online] Available: https://dnd.wizards.com/it.

[4] OpenAI. *ChatGPT*. [online] Available: https://platform.openai.com/docs/overview.

[5] Quixel. *MetaHumans on QuixelBridge*. [online] Available: https://quixel.com/blog/2021/4/14/metahumans-have-a-home-in-bridge.

[6] Wikipedia. *Non-player character*. [online] Available: https://en.wikipedia.org/wiki/Non-player_character.