

# Verbalna komunikacija afektivnog virtualnog agenta temeljena na chatbot usluzi

---

**Majetić, Domagoj**

**Master's thesis / Diplomski rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:964815>

*Rights / Prava:* [Attribution-NonCommercial-NoDerivatives 4.0 International/Imenovanje-Nekomercijalno-Bez prerada 4.0 međunarodna](#)

*Download date / Datum preuzimanja:* **2024-12-19**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

**Domagoj Majetić**

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentor:

Izv. prof. dr.sc. Tomislav Stipančić, dipl. ing.

Student:

Domagoj Majetić

Zagreb, 2024.

Izjavljujem da sam ovaj rad izradio samostalno koristeći stečena znanja tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru, izv. prof. dr. sc. Tomislavu Stipančiću, na svim korisnim savjetima, potpori i pomoći prilikom izrade i pisanja ovog diplomskog rada.

Također, posebno se zahvaljujem ocu Zlatku, majci Silvani, sestri Luciji, djevojci Dolores i prijateljima koji su mi uljepšali vrijeme studiranja i bili podrška tijekom studija.

Domagoj Majetić



SVEUČILIŠTE U ZAGREBU  
**FAKULTET STROJARSTVA I BRODOGRADNJE**



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za diplomske ispite studija strojarstva za smjerove:  
Proizvodno inženjerstvo, inženjerstvo materijala, industrijsko inženjerstvo i menadžment,  
mehatronika i robotika, autonomni sustavi i računalna inteligencija

Sveučilište u Zagrebu	
Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 - 04 / 24 - 06 / 1	
Ur.broj: 15 - 24 -	

## DIPLOMSKI ZADATAK

Student: **Domagoj Majetić** JMBAG: 0035215314

Naslov rada na hrvatskom jeziku: **Verbalna komunikacija afektivnog virtualnog agenta temeljena na chatbot usluzi**

Naslov rada na engleskom jeziku: **Verbal communication of an affective virtual agent based on a chatbot service**

Opis zadatka:

U području virtualne stvarnosti i afektivne robotike često je fokus istraživanja razvoj virtualnih modela ljudi. Na virtualne modele se mogu implementirati različiti obrasci ponašanja i djelovanja kako bi se ostvarilo njihovo što vjernije djelovanje. Razvoj tehnologije omogućuje korištenje postojećih chatbot sustava kao što su ChatGPT ili Bard da ostvare verbalnu komuniciraju virtualnih agenata s ljudima. Afektivni virtualni agent PLEA se razvija u sklopu Laboratorija za projektiranje izradbenih i montažnih sustava te se koristi za proučavanje interakcije ljudi i robota. PLEA omogućuje analizu i procjenu emocionalnog stanja osobe za vrijeme interakcije što potom koristi kako bi vodila interakciju. U radu je potrebno povezati ChatGPT uslugu s PLEA virtualnim agentom čime bi robot dobio sposobnost verbalne komunikacije.

U sklopu rada je potrebno:

- istražiti te povezati ChatGPT uslugu koristeći odgovarajuću programsku podršku s Unreal Engine 5 razvojnim simulacijskim okruženjem u kojem je modeliran PLEA virtualni agent
- implementirati programsku podršku koja omogućuje „speech to text“ te „tekst to speech“ funkcionalnosti
- izraditi programsku podršku za precizno animiranje izražaja lica usklađenih s trenutnim govorom
- testirati razvijeno rješenje te dati kritički osvrt na razvijeno programsko rješenje.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

7. ožujka 2024.

Zadatak zadao:

Izv. prof. dr. sc. Tomislav Stipančić

Datum predaje rada:

9. svibnja 2024.

Predviđeni datum obrane:

13. – 17. svibnja 2024.

Predsjednik Povjerenstva:

Prof. dr. sc. Ivica Garašić

# Sadržaj

<b>Popis slika.....</b>	<b>I</b>
<b>Sažetak.....</b>	<b>III</b>
<b>Summary .....</b>	<b>IV</b>
<b>1. Uvod.....</b>	<b>1</b>
<b>2. Obrada prirodnog jezika .....</b>	<b>2</b>
<b>2.1 Metode obrade prirodnog jezika.....</b>	<b>4</b>
2.1.1 Simbolički NLP.....	4
2.1.2 Statistički NLP – Statističke metode i modeli.....	5
2.1.3 Neuronski NLP – Neuronske mreže.....	6
<b>2.2 Primjena NLP-a.....</b>	<b>10</b>
2.2.1 Pretvaranje teksta u govor i govora u tekst.....	12
2.2.2 „Chatbotovi“ .....	15
<b>3. Alat za obradu prirodnog jezika ChatGPT .....</b>	<b>17</b>
<b>4. Programski alat za razvoj videoigara Unreal Engine.....</b>	<b>18</b>
<b>4.1 Virtualna stvarnost.....</b>	<b>19</b>
<b>4.2 Virtualni modeli ljudi „Metahumans“.....</b>	<b>21</b>
4.2.1 Audio2Face aplikacija.....	23
4.2.2 MetahumanSDK dodatak.....	24
<b>5. Povezivanje ChatGPT-a sa virtualnim bićem kroz UE5 okolinu .....</b>	<b>25</b>
<b>5.1 Aplikacijsko programsko sučelje.....</b>	<b>25</b>
5.1.1 Restful API.....	27
5.1.2 WebSocket API.....	28

<b>5.2</b>	<b><i>Unreal Engine dodaci.....</i></b>	<b>28</b>
<b>5.3</b>	<b><i>Mogućnosti povezivanja Chat GPT-a s Unreal Engine-om.....</i></b>	<b>29</b>
5.3.1	<i>Razvoj novog Unreal Engine dodatka pomoću API-ja.....</i>	29
5.3.2	<i>Korištenje postojećih Unreal Engine dodataka .....</i>	30
5.3.3	<i>Integracija putem vanjske aplikacije.....</i>	31
<b>6.</b>	<b><i>Izvedba zadatka.....</i></b>	<b>32</b>
<b>6.1</b>	<b><i>Virtualni agent s jednostavnim animacijama govora .....</i></b>	<b>32</b>
6.1.1	<i>Implementacija ChatGPT-a, STT-a i TTS-a .....</i>	37
6.1.2	<i>Animiranje lica virtualnog agenta.....</i>	42
<b>6.2</b>	<b><i>Virtualni agent s preciznim animacijama govora .....</i></b>	<b>49</b>
6.2.1	<i>Implementacija ChatGPT-a, STT-a i TTS-a .....</i>	51
6.2.2	<i>Animiranje lica virtualnog agenta.....</i>	54
<b>7.</b>	<b><i>Zaključak .....</i></b>	<b>58</b>
<b>8.</b>	<b><i>Literatura .....</i></b>	<b>59</b>
<b>9.</b>	<b><i>Prilozi.....</i></b>	<b>62</b>

## Popis slika

Slika 1 Ilustrativni prikaz komunikacije čovjeka i računala koristeći NLP [3] ....	3
Slika 2 Shematski prikaz 3 glavne metode NLP-a [4] .....	4
Slika 3 Prikaz korištenja SHRDLU-a [6].....	5
Slika 4 Shematski prikaz modela jednostavne neuronske mreže [12].....	7
Slika 5 Primjer neuronske mreže s jednosmjernim prijenosom [13] .....	8
Slika 6 Primjer neuronske mreže s povratnom propagacijom [14].....	9
Slika 7 Primjer konvolucijske neuronske mreže [15] .....	9
Slika 8 Automatsko dovršavanje rečenica u Google tražilici [17].....	11
Slika 9 Analiza sentimenta recenzija neke usluge [18].....	12
Slika 10 Shematski prikaz načina rada TTS-a [19].....	14
Slika 11 Prototip afektivne humanoidne robotske glave PLEA [26] .....	19
Slika 12 Prikaz korištenja VR naočala [28] .....	20
Slika 13 PLEA <i>Metahuman</i> model.....	21
Slika 14 Upravljačka ploča za lice <i>Metahuman-a</i> [29].....	22
Slika 15 Princip rada A2F aplikacije [30] .....	23
Slika 16 Shematski prikaz načina rada API-ja [33] .....	26
Slika 17 Shematski prikaz načina rada REST API-ja [35].....	27
Slika 18 Dijagram koji opisuje vezu pomoću WebSocket-a [37] .....	28
Slika 19 Sučelje UE5 – virtualni agent u virtualnoj stvarnosti .....	34
Slika 20 AzSpeech postavke .....	36
Slika 21 Open AI Api ključ .....	37
Slika 22 Funkcija za pretvaranje govora u tekst i poziv funkcije „Call GPT“ ...	38
Slika 23 Slanje upita ChatGPT-u.....	39
Slika 24 Pretvaranje generiranog teksta u govor.....	40
Slika 25 Generiranje zvuka na temelju generiranog teksta od strane ChatGPT-a .....	41



Slika 26 Petlja kojom se ponovno pokreće program.....	41
Slika 27 Sekvencer i <i>ControlRig</i> .....	43
Slika 28 Funkcija „Is Character Talking“ .....	44
Slika 29 Poziv „Is Character Talking“ funkcije .....	45
Slika 30 Pokretanje animacije virtualnog agenta prema stanju „State Machine-a“ .....	46
Slika 31 Postavke „State Machine-a“.....	46
Slika 32 Uvjet za pokretanje animacije govora.....	47
Slika 33 Uvjet za pokretanje animacije mirovanja .....	47
Slika 34 <i>Metahuman</i> postavke lica.....	48
Slika 35 UE5 Sučelje sa postavljenim virtualnim agentom.....	50
Slika 36 MetahumanSDK postavke .....	51
Slika 37 Open AI Api ključ .....	52
Slika 38 Funkcija za pretvaranje govora u tekst i poziv funkcije „Call GPT“ ...	52
Slika 39 Slanje upita ChatGPT-u.....	53
Slika 40 Pretvaranje generiranog teksta u govor.....	54
Slika 41 MetahumanSDK funkcije u shematskom planu dio 1. ....	55
Slika 42 MetahumanSDK funkcije u shematskom planu dio 2. ....	56
Slika 43 MetahumanSDK funkcije u shematskom planu dio 3. ....	56

## Sažetak

Tema ovog rada je integracija ChatGPT usluge sa virtualnim agentom radi ostvarivanja verbalne komunikacije u području virtualne stvarnosti i afektivne robotike. U teorijskom dijelu opisuju se metode obrade prirodnog jezika te njihova primjena i analiziraju se načini povezivanja ChatGPT usluge s Unreal Engine 5 razvojnim okruženjem u kojem je modeliran PLEA virtualni agent. Praktični dio rada obuhvaća integraciju ChatGPT usluge sa virtualnim agentom, što uključuje implementaciju programske podrške za pretvorbu govora u tekst i teksta u govor te animiranje izražaja lica, virtualnog modela čovjeka, sukladno govoru.

Ključne riječi: ChatGPT, virtualni agent, verbalna komunikacija, virtualna stvarnost, afektivna robotika, obrada prirodnog jezika, Unreal Engine, animiranje lica

## **Summary**

The topic of this paper is the integration of the ChatGPT service with a virtual agent to achieve verbal communication in the field of virtual reality and affective robotics. The theoretical part describes Natural Language Processing methods, their application, and analyzes ways to connect the ChatGPT service with the Unreal Engine 5 development environment. The practical part of the paper involves integrating the ChatGPT service with the virtual agent, including implementing speech-to-text and text-to-speech conversion, as well as animating facial expressions of a virtual human model in accordance with speech.

**Keywords:** ChatGPT, virtual agent, verbal communication, virtual reality, affective robotics, Natural Language Processing, Unreal Engine, facial animation

## 1. Uvod

Umjetna inteligencija (UI, prema eng. akronimu AI, od *Artificial Intelligence*), dio je računalne znanosti koji se bavi razvojem sposobnosti računala da obavljaju zadaće za koje je potreban neki oblik inteligencije, tj. da se mogu snalaziti u novom okruženju, učiti nove koncepte, donositi zaključke, razumjeti prirodni jezik, raspoznavati prizore i dr. [1]. Jedna od primjena umjetne inteligencije je obrada prirodnog jezika (eng. akronim NLP, od *Natural Language Processing*). NLP omogućava računalu da razumije tekst ili govor stvoren od strane čovjeka. Sposobnost procesuiranja takvih informacija je čovjeku jednostavan zadatak, no računalima su potrebni kompleksni programi kako bi se ulazni podaci mogli obrađivati na sličan način [2].

U sklopu ovoga rada analizira se jedan od alata za obradu prirodnog jezika - ChatGPT te njegova primjena u virtualnoj okolini u sklopu programskog paketa za razvoj videoigara Unreal Engine 5 (krat. UE5). Cilj ovoga rada je razvijanje programske podrške za povezivanje ChatGPT usluge s virtualnim bićem PLEA, razvijenim na Fakultetu strojarstva i brodogradnje, unutar UE5 okoline. Pored integracije ChatGPT usluge, u radu je opisana i implementacija programske podrške za pretvaranje govora u tekst i teksta u govor uz animiranje lica virtualnog agenta na temelju izgovorenih odgovora.

## 2. Obrada prirodnog jezika

Obrada prirodnog jezika je sposobnost računalnog programa da razumije ljudski govor, kako onaj koji je izgovoren, tako i onaj pisani. NLP postoji već više od 50 godina i ima različite primjene u stvarnom svijetu u brojnim područjima [2].

NLP se postiže brojnim različitim pravilima zasnovanim ili strojno učenim tehnikama, kako bi računala mogla "razumjeti" prirodni jezik. Baš kao što ljudi imaju različite senzore, poput ušiju za slušanje i oči za gledanje, računala koriste mikrofone za prikupljanje zvuka, kamere za prikupljanje slika i programe za čitanje. Također kao što ljudi imaju mozak za obradu tog ulaza, računala imaju program za obradu svojih odgovarajućih ulaza. U nekom trenutku obrade, ulaz se pretvara u kod koji računalo može razumjeti. Postoje dvije glavne faze obrade prirodnog jezika, a to su pred-obrada podataka i razvoj algoritama [2].

Pred-obrada podataka uključuje pripremu i čišćenje tekstualnih podataka kako bi ih strojevi mogli analizirati. Podaci se pojednostavljaju u oblik koji daje mogućnost algoritmima da ih jednostavnije procesuiraju. Postoji nekoliko načina na koje se to može učiniti, uključujući sljedeće:

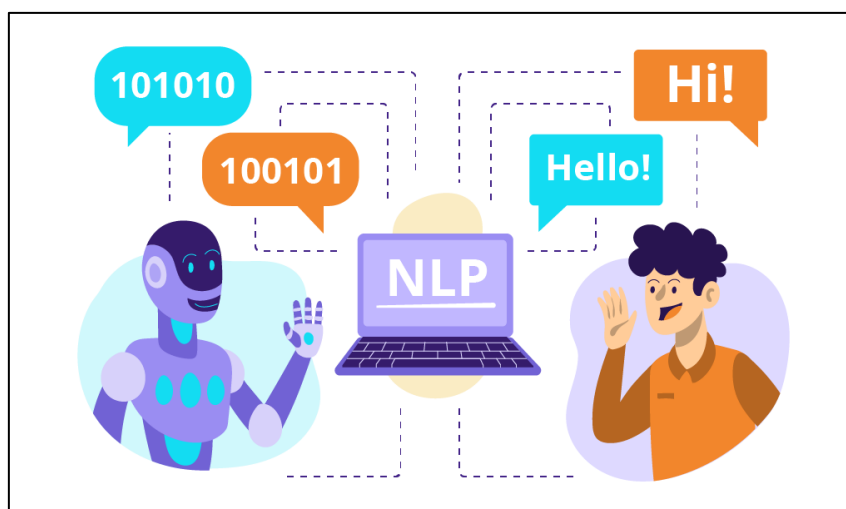
- Tokenizacija (eng. *Tokenization*) – zamjenjuju se osjetljive informacije s neosjetljivim informacijama, ili tokenom. Tokenizacija se često koristi u platnim transakcijama kako bi se zaštitili podaci o kreditnoj kartici.
- Uklanjanje beskorisnih riječi (eng. *Stop word removal*) - česte riječi poput veznika, čestica, priloga i prijedloga se uklanjaju iz teksta, tako da ostanu riječi koje pružaju najviše informacija o tekstu.
- Lemmatizacija (eng. *Lemmatization*) – grupiraju se različito izvedeni oblici istih riječi. Na primjer, riječ "raditi" bi se svela na svoj korijenski oblik ili osnovu "rad" kako bi se obradila.

- Označavanje vrsta riječi u rečenici (eng. akronim POS, od *Part-of-speech tagging*) - riječi se označavaju na temelju toga kojoj vrsti pripadaju poput imenica, glagola ili pridjeva u svrhu daljnje obrade [2].

Kada su podaci pred-obrađeni, razvija se algoritam za njihovu obradu. Postoji mnogo različitih algoritama obrade prirodnog jezika, ali se često koriste sljedeće dvije glavne vrste:

- Sustav na temelju zadanih pravila - ovaj sustav koristi set pažljivo osmišljenih i kompleksnih lingvističkih pravila koje unosi čovjek. Ovakav princip bio je korišten u najranijim fazama razvoja obrade prirodnog jezika te se još uvijek koristi.
- Sustav na temelju strojnog učenja - algoritmi strojnog učenja koriste statističke metode ili neuronske mreže. Algoritmi su trenirani obavljati zadatke na temelju podataka za učenje koji su im dani i prilagođavaju svoje metode kako se obrađuje više podataka [2].

Na slici [Slika 1.] nalazi se ilustrativni prikaz komunikacije čovjeka i računala koristeći obradu prirodnog jezika.



Slika 1 Ilustrativni prikaz komunikacije čovjeka i računala koristeći NLP [3]

## 2.1 Metode obrade prirodnog jezika

Postoji nekoliko metoda, odnosno vrsta modela obrade prirodnog jezika, a tri glavne uključuju:

- Simbolički NLP,
- Statistički NLP i
- Neuralni NLP [4].

Navedene metode opisane su u nastavku ovoga rada, a na slici [Slika 2.] može se vidjeti shematski prikaz navedenih metoda.

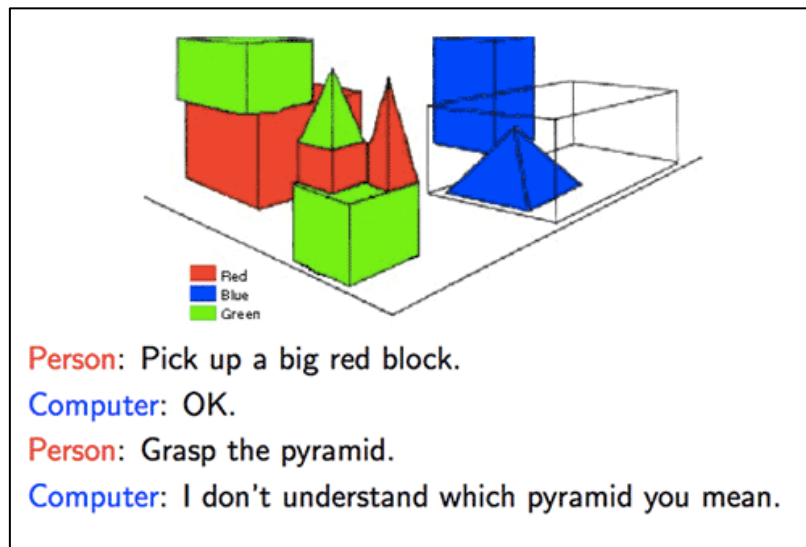


Slika 2 Shematski prikaz 3 glavne metode NLP-a [4]

### 2.1.1 Simbolički NLP

Simbolički NLP pojavio se u ranim 1950-ima i bio je relevantan sve do ranih 1990-ih. Temeljna ideja ovog pristupa bila je podučiti računala kako bi razumjela jezik na način na koji ljudi uče čitati i pisati, koristeći se skupinom unaprijed definiranih pravila koja su se unosila od strane čovjeka. U 1960-ima razvijeni su neki od prvih uspješnih sustava obrade prirodnog jezika. Jedan od najranijih takvih programa bio je SHRDLU, koji je omogućavao interakciju između čovjeka i računala. Nakon toga, pojavili su se i prvi *chatbotovi*, softveri koji omogućavaju

automatiziranu komunikaciju između ljudi i računala. Takvi programi su se prilagođavali porukama korisnika i smisleno odgovarali na njih [5]. Na slici [Slika 3.] prikazan je rad SHRDLU programa.



Slika 3 Prikaz korištenja SHRDLU-a [6]

### 2.1.2 Statistički NLP – Statističke metode i modeli

Prije 1980-ih većina NLP sustava temeljila se na složenim setovima pravila koje je postavljao čovjek. Početkom 1990-ih došlo je do značajnog preokreta s pojavom statističkog NLP-a i strojnog učenja [5]. Strojno učenje, kao grana umjetne inteligencije, omogućava računalima da postanu preciznija u predviđanju i donošenju odluka, bez potrebe direktnog programiranja [7]. Statistički NLP pridonio je automatizaciji obrade prirodnog jezika, omogućujući računalima da čitaju i razumiju značenje ljudskih jezika.

Statistički model je matematički model koji objedinjuje skup statističkih pretpostavki o podacima koji se generiraju. Takav model predstavlja proces stvaranja podataka u idealiziranom obliku. Ovaj model često se definira kao



matematički odnos između različitih varijabli, od kojih su neke slučajne, dok su druge neslučajne [8]. Ovi modeli mogu predstaviti više mogućih rješenja, pružajući tako pouzdanije rezultate kada su uključeni u veće sustave. Često se koriste u sustavima za prepoznavanje govora, a neki od takvih sustava zahtijevaju trening, gdje korisnik čita tekst ili određene riječi iz vokabulara kako bi se podaci pohranili u sustav [9]. Iako su neuronske mreže zamijenile statističke metode u većoj mjeri, statistički modeli i dalje su relevantni u kontekstima gdje je važna statistička interpretacija i transparentnost.

Primjeri primjene statističkog NLP-a nalaze se u nastavku:

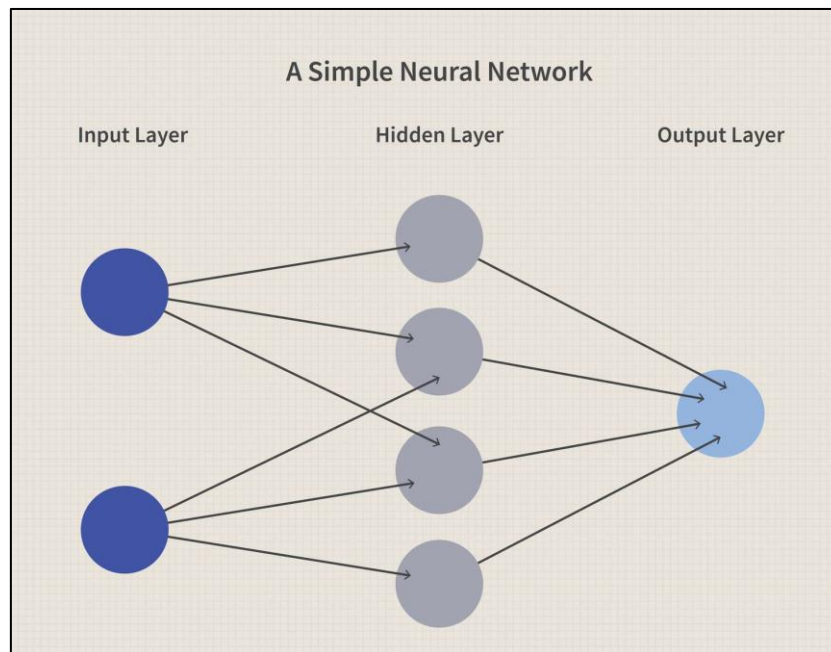
- Predlaganje riječi prilikom upisivanja teksta,
- Uočavanje i ispravljanje gramatičkih grešaka prilikom upisivanja teksta,
- Prepoznavanje rukopisa unatoč nečitko napisanom tekstu,
- Sažimanje teksta i
- Kategorizacija teksta [10].

### 2.1.3 Neuronski NLP – Neuronske mreže

Struktura neuronske mreže funkcionira putem procesa obuke, to je tehnologija umjetne inteligencije primijenjena na neuro-lingvističko programiranje [11].

U osnovi, neuronska mreža predstavlja modeliranje načina na koji ljudski mozak funkcionira prilikom obrade informacija. To je prilagodljiv sustav koji, kroz obuku, uči i neprekidno poboljšava modeliranje odnosa i uzoraka između podataka, pružajući rješenja za kompleksne probleme i simulira obradu prirodnog jezika. Takva mreža sadrži neurone ili čvorove (osnovne jedinice) međusobno povezane i organizirane u slojevima. Postoji ulazni sloj koji prima ulazne podatke, jedan ili više skrivenih slojeva koji propagiraju podatke, te izlazni sloj koji šalje

rezultat analize ili obrade prirodnog jezika [11]. Na slici [Slika 4.] nalazi se shematski prikaz modela jednostavne neuronske mreže.



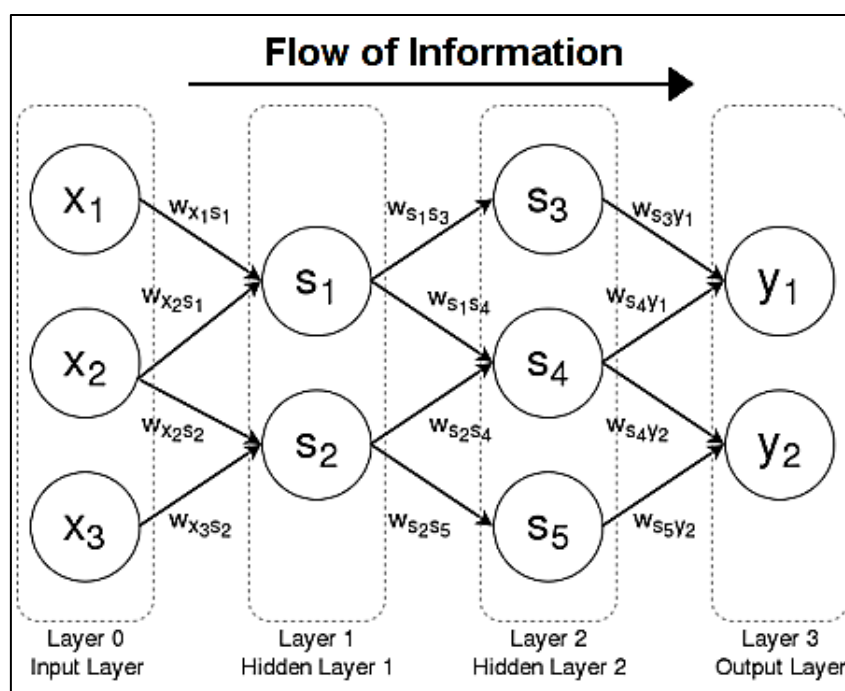
Slika 4 Shematski prikaz modela jednostavne neuronske mreže [12]

Da bi imitirala ljudski mozak, neuronska mreža ispituje vrijednosti ili podatke koje prima u svom ulaznom sloju, predviđajući i propagirajući rezultat. Ulazni sloj šalje podatke drugom sloju, koji ih potom prenosi u sukcesivne skrivene slojeve. U drugom sloju, neuroni ili čvorovi otkrivaju i filtriraju obrasce visoke relevantnosti te kombiniraju podatke. Svaka ulazna vrijednost dodjeljuje se težini koja mijenja težinu ulaza. Težine u neuronskoj mreži su numeričke vrijednosti povezane s vezama između neurona u različitim slojevima mreže. Svaka veza između dva neurona iz različitih slojeva ima povezanu težinu koja označava snagu i smjer utjecaja jednog neurona na drugi. Rezultirajuće vrijednosti težina i ulaznih podataka se potom sumiraju, a na tu sumu se primjenjuje aktivacijska funkcija. Aktivacijska funkcija odlučuje treba li se neuron aktivirati ili ne, svrha aktivacijske funkcije je unijeti nelinearnost u neuronsku mrežu. Da ne postoje

aktivacijske funkcije, neuronska mreža bi konstantno učila linearno. Ovakav sustav učenja temelji se na pretpostavci da u svakom skupu parametara postoji način kombiniranja kako bi se predvidio određeni rezultat. Neuronska mreža odgovorna je za postizanje najbolje moguće kombinacije parametara za određeni problem i primjenu iste, tj. obavljanje predikcije i pružanje izlaza [11].

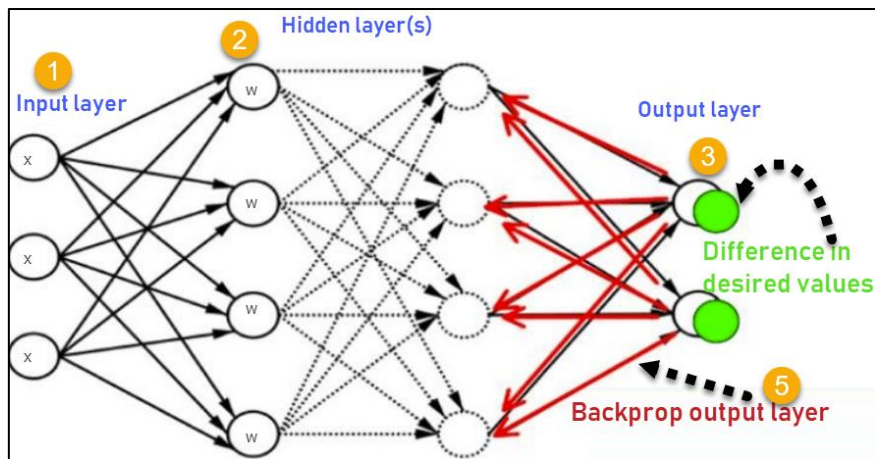
Ovisno o tome kako podaci teku ili se prenose od ulaznih čvorova do izlaznih čvorova, postoji više vrste neuronskih mreža, najčešće korištene su:

- Neuronske mreže s jednosmjernim prijenosom - u ovoj strukturi mreže, podaci se obrađuju samo u jednom smjeru, od ulaznog čvora do izlaznog čvora. Na slici [Slika 5.] prikazan je primjer takve mreže.



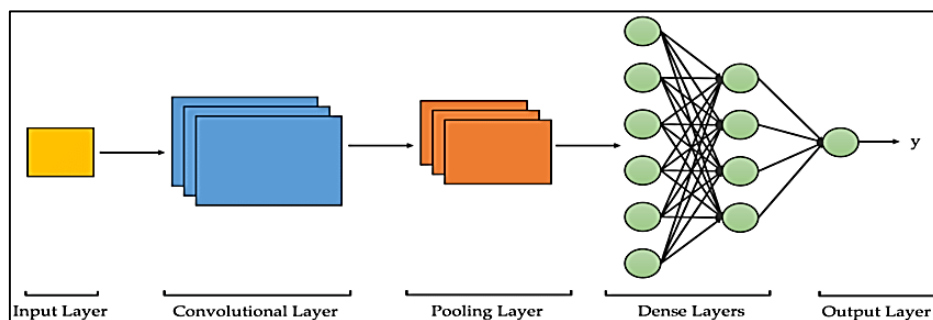
Slika 5 Primjer neuronske mreže s jednosmjernim prijenosom [13]

- Neuronske mreže s povratnom propagacijom - u ovom slučaju, vrijednosti se također prenose od ulaznog čvora do izlaznog čvora, ali putuju različitim putevima u mreži. Samo jedan od tih puteva je ispravan, a mreža ga otkriva kroz rad povratne petlje. Na slici [Slika 6.] prikazan je primjer takve mreže.



Slika 6 Primjer neuronske mreže s povratnom propagacijom [14]

- Konvolucijske neuronske mreže - ova vrsta mreže ima nekoliko skrivenih slojeva obučeni za izvođenje konvolucije, matematičke operacije koja omogućuje povezivanje dva skupa informacija. Vrlo su korisne u klasifikaciji i prepoznavanju slika [11]. Na slici [Slika 7.] prikazan je primjer takve mreže.



Slika 7 Primjer konvolucijske neuronske mreže [15]

Neuronska mreža se trenira prilagođavanjem vrijednosti težina ulaznih vrijednosti kako bi ponudila najtočniji odgovor na neki problem. Drugim riječima, neuronska mreža se trenira korištenjem procesa učenja.

Tijekom treninga, neuronska mreža može koristiti različite metode učenja:

- Nadgledano ili kontrolirano učenje - vanjski agent pokazuje sustavu obrasce i rezultat koji treba generirati. Na taj način, mreža će izvršiti izračune i kombinacije kako bi njen izlaz odgovarao očekivanom rezultatu.
- Nenadgledano učenje (bez vanjskog utjecaja) - u ovom procesu, izlaz je nepoznat. Mreža se temelji samo na promatranjima napravljenim na ulaznim vrijednostima.
- Pojačano učenje - u ovom mehanizmu učenja, mreža samostalno provodi analizu, ali se izlazi kasnije procjenjuju. Svaki točan izlaz pozitivno se potiče, dok su pogrešni izlazi odbačeni.
- Hibridno učenje - ova metoda učenja kombinira gore navedene mehanizme [11].

Neuronske mreže, uz kombinaciju nadgledanog i nenadgledanog učenja, se koriste u tehnologiji obrade prirodnog jezika kako bi omogućile računalima uspješno obavljanje procesa obrade tekstova ili dokumenata. [11].

Neuronske mreže su NLP modelima pružile ogromne kapacitete za razumijevanje i simulaciju ljudskog jezika. Omogućile su strojevima da predviđaju riječi i obrađuju teme koje nisu bile dio procesa učenja. Da bi postigle ovu izvedbu u procesima NLP-a, neuronske mreže moraju biti trenirane s velikim količinama dokumenata, odnosno korpusima, prema vrsti teksta ili jezika koji se obrađuje. U jezičnim modelima NLP-a, neuronske mreže djeluju u ranim fazama, pretvarajući riječi vokabulara u vektore. Djeluju na temelju principa da se u tekstu značenje određene riječi povezuje s riječima koje se nalaze oko nje. [11].

## 2.2 Primjena NLP-a

NLP danas ima široku primjenu te se može prepoznati u raznim svakodnevnim situacijama i u uslugama koje ljudi koriste. U svrhu izrade ovog rada detaljnije će

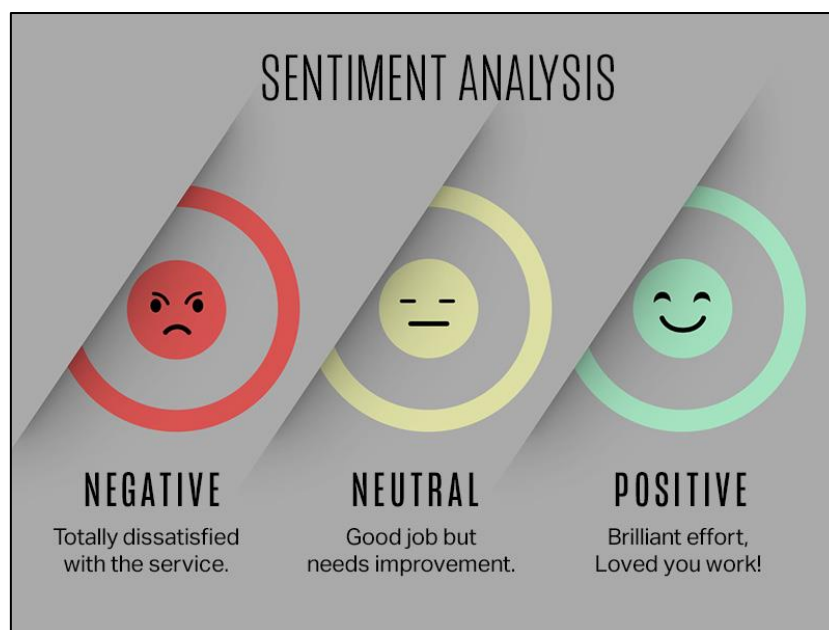
se analizirati pretvaranja teksta u govor i obratno te *chatbotovi*. Osim navedenih u nastavku su spomenuti još neki učestali primjeri primjene NLP-a:

- Automatsko dovršavanje rečenica (eng. *Autocomplete*),
- Prevođenje jezika (eng. *Language translator*),
- Analiza sentimenta (eng. *Sentiment analysis*),
- Provjera pravopisa (eng. *Grammar checker*),
- Filtriranje e-mailova (eng. *Email classification and filtering*) [16].

Na slikama [Slika 8.] i [Slika 9.] prikazani su primjeri primjene NLP-a.



Slika 8 Automatsko dovršavanje rečenica u Google tražilici [17]



Slika 9 Analiza sentimenta recenzija neke usluge [18]

### 2.2.1 Pretvaranje teksta u govor i govora u tekst

Pretvaranje teksta u govor (eng. akronim TTS, od *Text to speech*) se koristi za pretvaranje pisanog teksta u izgovorene riječi, dok se pretvaranje govora u tekst (eng. akronim STT, od *Speech to text*) koristi za suprotno. TTS se koristi kako bi se pisani sadržaj učinio čujnim, djelujući kao pomoćnik za osobe s oštećenjima vida ili poteškoćama u učenju. STT, s druge strane, bilježi govorni jezik i pretvara ga u pisani prijepis, koristan za diktiranje i glasovne naredbe [19].

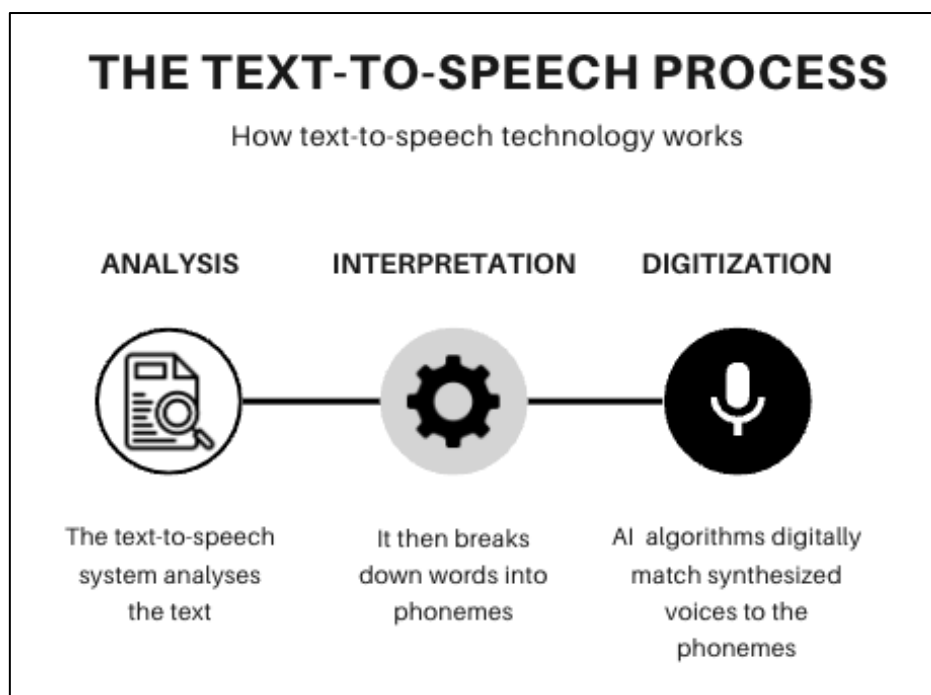
TTS se obično integrira u e-čitače, sustave za javno oglašavanje i virtualne asistente za pružanje audio izlaza. STT nalazi svoju primjenu u uslugama prijepisa, aplikacijama s glasovnim upravljanjem i titlovima u stvarnom vremenu za osobe oštećena sluha. Kontekst upotrebe TTS-a prvenstveno je vođen izlazom, fokusirajući se na zvučnu isporuku informacija. Nasuprot tome, STT je usmjeren na unos, fokusirajući se na snimanje i obradu govornog jezika [19].

TTS tehnologija uključuje analizu teksta, obradu jezika i sintezu govora. Kako bi TTS sustav bio efikasan ključno je prenijeti nijanse govornog jezika, uključujući intonaciju i ritam. STT zahtijeva napredne mogućnosti prepoznavanja glasa za točnu transkripciju različitih naglasaka, dijalekata i govornih obrazaca, često u stvarnom vremenu [19].

#### 2.2.1.1 Princip rada TTS-a

U početku, TTS sustav rastavlja tekst, segmentira ga na foneme odnosno najmanje zvučne jedinice u bilo kojem jeziku. Ova segmentacija je ključna za sposobnost sustava da točno izgovara različite riječi. Nakon fonemske segmentacije, sustav nastavlja pretvarati te zvukove u digitalni govor. Zatim koristeći AI algoritme obučene na opsežnim skupovima podataka govornog jezika, sustav proizvodi govor koji nalikuje ljudskim tonovima i ritmovima. Ovaj generirani govor se zatim usklađuje s identificiranim fonemima, što kulminira ispisom prirodnog zvuka. Zahvaljujući napretku u umjetnoj inteligenciji i strojnom učenju, moderne TTS tehnologije su postale iznimno napredne. Pomoću TTS tehnologija danas je moguće razumjeti kontekstualne nijanse, prilagoditi se više jezika i donekle oponašati emocionalan kontekst napisanog teksta. Ova su poboljšanja značajno humanizirala govorni izlaz, što je dovelo do prirodnije i zanimljivije interakcije s digitalnim uređajima [19]. Na slici [Slika 10.] nalazi se shematski prikaz načina rada TTS-a.





Slika 10 Shematski prikaz načina rada TTS-a [19]

### 2.2.1.2 Princip rada STT-a

Tehnologija pretvaranja govora u tekst pretvara govorni jezik u pisani tekst kroz složen proces u više koraka. Prvo, počinje snimanjem izgovorenih riječi, obično putem mikrofona. Ovaj audio ulaz se zatim pretvara u digitalni format koji sustav može obraditi. Srž STT-a leži u njegovoj sposobnosti da analizira digitalni zvuk. Koriste se sofisticirani algoritmi za rastavljanje govora na manje, prepoznatljive segmente. Ti segmenti su fonemi - najmanje jedinice zvuka u govoru. STT sustav uspoređuje ove foneme s unaprijed definiranim lingvističkim modelom za prepoznavanje riječi i fraza. Ovaj korak je ključan za razumijevanje različitih naglasaka, dijalekata i varijacija u govoru. Zatim, sustav primjenjuje tehnike obrade prirodnog jezika. Korištenjem NLP-a se ostvaruje razumijevanje konteksta i sintakse govornog jezika, omogućujući točniju transkripciju. Također omogućuje se sustavu rukovanje složenim rečeničnim strukturama i žargonom specifičnim za industriju. U naprednim STT sustavima koristi se strojno učenje i

algoritmi dubokog učenja, koji se poboljšavaju integriranjem velikih količina podataka u njih. Ove tehnologije omogućuju sustavu da s vremenom uči iz novih govornih obrazaca, naglasaka, pa čak i jezika, povećavajući njegovu točnost i učinkovitost. Ukratko, STT tehnologija uključuje snimanje zvuka, fonemsku analizu, lingvističko modeliranje i NLP, sve poduprto strojnim učenjem, za učinkovito pretvaranje govora u tekst [19].

### 2.2.2 „Chatbotovi“

*Chatbot* je računalni program koji simulira ljudski razgovor s krajnjim korisnikom. Nisu svi *chatbotovi* opremljeni umjetnom inteligencijom, ali moderni *chatbotovi* sve više koriste konverzijske AI tehnike kao što je obrada prirodnog jezika za razumijevanje korisničkih pitanja i automatiziranje odgovora na njih [20].

Najraniji *chatbotovi* su bili interaktivni programi, koji su se oslanjali na ograničen skup uobičajenih pitanja s unaprijed napisanim odgovorima. Takvi rudimentarni, tradicionalni *chatbotovi* ne mogu obraditi složena pitanja niti odgovoriti na jednostavna pitanja koja programeri nisu predvidjeli. S vremenom su algoritmi *chatbota* postali sposobni za složenije programiranje temeljeno na pravilima, pa čak i za obradu prirodnog jezika, omogućujući da se upiti korisnika izraze razgovornim putem. To je dovelo do nove vrste *chatbota*, kontekstualno svjesnog i opremljenog strojnim učenjem kako bi kontinuirano optimizirao svoju sposobnost ispravne obrade i predviđanja upita kroz interakciju sa ljudskim jezikom. Moderni AI *chatbotovi* sada koriste razumijevanje prirodnog jezika da razaznaju značenje otvorenog korisničkog unosa, prevladavajući sve, od tipfelera do problema s prijevodom. Napredni AI alati zatim preslikavaju to značenje na specifičnu namjeru prema kojoj korisnik želi da *chatbot* djeluje i koriste AI za razgovor kako bi formulirali odgovarajući odgovor. Ove AI tehnologije

iskorištavaju i strojno učenje i duboko učenje, s nekim nijansiranim razlikama za razvoj sve preciznije baze znanja pitanja i odgovora utemeljenih na interakcijama korisnika. Ova sofisticiranost, oslanjajući se na nedavni napredak u velikim jezičnim modelima (eng. akronim LLM, od *Large language model*), dovela je do povećanog zadovoljstva korisnika i svestranijih primjena *chatbotova*. Vrijeme potrebno za izradu AI *chatbota* može varirati ovisno o razvojnim alatima koji se koriste, složenosti *chatbota*, željenim značajkama, dostupnosti podataka i treba li se integrirati s drugim sustavima, bazama podataka ili platformama. [20].

ChatGPT je bio prvi naširoko korišten AI *chatbot*, ali se u međuvremenu pojavila velika konkurencija. Ovisno o preferencijama korisnika trenutno postoji značajan broj AI *chatbotova* koji uvelike olakšavaju svakodnevnicu svim ljudima, a neki od najpopularnijih su:

- ChatGPT – Osmišljen za AI generiranje teksta,
- Google Bard – Najčešće korišten za produktivnost,
- Microsoft Bing Chat - Osmišljen za korisnike Binga i Microsoft Edgea i
- Jasper - Najčešće korišten za marketing [21].

### 3. Alat za obradu prirodnog jezika ChatGPT

ChatGPT (eng. akronim GPT, od *Generative Pre-training Transformer*) je alat za obradu prirodnog jezika, razvijen u korporaciji OpenAI, baziran na umjetnoj inteligenciji i procesima dubokog učenja koji omogućava razgovor korisnika sa *chatbotom*. ChatGPT je u ulozi virtualnog asistenta koji može dati odgovor na svako postavljeno pitanje, no odgovor koji generira nije nužno točan ili smislen. Također ChatGPT je sposoban održavati razgovor sa korisnikom, ima mogućnost spremanja svih prijašnje vođenih razgovora te se lako nadovezuje na iste. Takav jezični model može pomoći korisniku sa zadacima kao što su sastavljanje e-pošte, eseja iz raznih područja života te može pomoći korisniku kod kreiranja ili razumijevanja koda za razvoj nekog programa [22].

To je generativni jezični model temeljen na treniranoj bazi podataka s interneta, obučen da predviđa sljedeću riječ u zadanoj rečenici te koristi transformer model koji je vrsta neuronske mreže posebno dizajnirane za obradu prirodnog jezika. Konkretno GPT-3 verzija ChatGPT-a sačinjena je od 175 milijardi parametara, što ga čini najvećim jezičnim modelom ikada obučenim. Model GPT-3 uvježban je na skupu teksta koji je uključivao više od 8 milijuna dokumenata i više od 10 milijardi riječi. Nakon što je model dobro uvježban usavršen je finim podešavanjem pod nadzorom ljudskih AI trenera [23].

#### 4. Programski alat za razvoj videoigara Unreal Engine

Unreal Engine (krat. UE) je paket alata, razvijen u tvrtki Epic Games, koji se koristi za razvoj igara, virtualne stvarnosti (eng. akronim VR, od *Virtual reality*), proširene stvarnosti (eng. akronim AR, od *Augmented reality*), stvaranje linearnog filmskog i televizijskog sadržaja, produkciju emitiranja događaja uživo, obuku i simulaciju te druge primjene u stvarnom vremenu. UE je besplatan za preuzimanje i dolazi potpuno opremljen i spreman za korištenje sa svim uključenim značajkama i potpunim pristupom izvornom kodu. UE se može besplatno koristiti za učenje i razvoj internih projekata, također omogućuje distribuciju mnogih komercijalnih projekata bez plaćanja naknada Epic Gamesu, uključujući prilagođene projekte isporučene klijentima, linearni sadržaj i bilo koji proizvod koji ne ostvaruje prihod [24].

Unreal Engine je poznat po svojoj fleksibilnosti, skalabilnosti i visokim performansama. Omogućuje programerima da razvijaju igre u 2D ili 3D okruženju te podržava napredne funkcionalnosti poput zakona fizike, umjetne inteligencije, animacija i simulacija [24].

Dodatno, Unreal Engine je korišten u razvoju PLEA virtualnog bića koje „živi“ u *MetaVerse* kibernetičkom prostoru i može posjećivati različite okoline u realnom svijetu kroz interakcijska sučelja. PLEA je razvijena na Fakultetu strojarstva i brodogradnje u sklopu projekta pod vodstvom dr. sc. Tomislava Stipančića [25]. Na slici [ Slika 11.] prikazan je prototip afektivne humanoidne robotske glave PLEA.



Slika 11 Prototip afektivne humanoidne robotske glave PLEA [26]

#### 4.1 Virtualna stvarnost

Virtualna stvarnost (eng. akronim VR, od *Virtual Reality*) je simulacija okoline generirana računalom, pomoću koje korisnik može doživjeti privid boravka, kretanja i percepcije unutar nje. Ovo trodimenzionalno multimedijско okruženje se stvara korištenjem posebnih računalnih periferija i programa. Korisniku se prikazuju slike stvarnog ili imaginarnog okruženja na zaslonu računala ili kroz specijalne stereoskopske uređaje poput naočala ili kacige s dva ugrađena LCD zaslona. Iskustvo se nadopunjuje zvukovima putem slušalica ili zvučnika, vibracijama te mogućnošću simulacije taktilnih i mirisnih osjeta. Interakcija između korisnika i računala postiže se kroz standardne ulazne uređaje poput tipkovnice ili miša, ali i kroz specijalne uređaje poput rukavica sa sensorima za praćenje položaja ruku i pokreta prstiju te druge uređaje za praćenje kretanja. Virtualna stvarnost ostvaruje se osnovnim programskim jezikom VRML (akronim od engl. *Virtual Reality Modelling Language*: jezik za modeliranje prividne

stvarnosti). VR omogućava korisnicima novu razinu interakcije s računalom, što je posebno korisno u edukativne svrhe zbog uštede vremena i resursa pri obuci za složene zadatke poput upravljanja vozilima ili borbene taktike. Također se koristi za prezentaciju projekata, umjetničke izraze, kao pomoć u snalaženju u nepoznatim okruženjima te za zabavu i računalne igre. Određeni VR sustavi omogućuju trodimenzionalne prikaze korisnika te povezivanje s drugim korisnicima, omogućujući prijenos pokreta i izraza lica u istoj virtualnoj stvarnosti. Također je moguće stvoriti alternativni identitet putem virtualnih avatara. Ovi prostori predstavljaju spoj društva i tehnologije, stvarajući novo značenje susreta i identiteta. Virtualna stvarnost donosi nove utjecaje na ovisnost o internetu te na razvoj virtualnih identiteta, a puni opseg tih utjecaja na društvene i psihološke aspekte pojedinca i društva tek će se otkriti [27].

Unreal Engine 5 pruža alate za izradu visokokvalitetnih virtualnih svjetova i iskustava virtualne stvarnosti. S naprednim tehnologijama poput *Nanite* i *Lumen*, omogućuje stvaranje realističnih okruženja s detaljnom grafikom i dinamičnim osvjetljenjem. Razvojni programeri mogu koristiti ove alate za kreiranje impresivnih VR iskustava prilagođenih različitim VR uređajima i kontrolerima. Na slici [Slika 12.] prikazan je način korištenja VR naočala.



Slika 12 Prikaz korištenja VR naočala [28]

## 4.2 Virtualni modeli ljudi „Metahumans“

MetaHuman je alat za kreiranje visokokvalitetnih modela ljudi koji se koriste unutar Unreal Engine-a. Proces izrade virtualnih ljudi je izuzetno prilagodljiv, omogućavajući podešavanje izgleda lika sve od generalnih obilježja pa do najsitnijih detalja poput boje očiju i oblika trepavica. Izrađeni lik dolazi s ugrađenim kontrolama za animaciju lica i tijela, čime se izbjegava potreba za dodatnim softverima koji bi to trebali omogućiti. Na slici [Slika 13.] prikazan je virtualni model čovjeka koji predstavlja PLEA virtualnog agenta ranije spomenutog u ovom radu.



Slika 13 PLEA *Metahuman* model

U Unreal Engine-u moguće je precizno animirati lice ili tijelo *Metahuman-a*, to se može postići kombinacijom funkcija Sekvencera (eng. *Sequencer*), *Control Rig-a* i *IK Rig-a* (eng. akronim IK, od *Inverse kinematics*). Sekvencer pruža sučelje temeljeno na vremenskoj crti za orkestriranje animacija, omogućujući



preciznu kontrolu izraza lica ili pokreta tijela u određenom trenutku. *Control Rig* omogućuje postavljanje i podešavanje kontrola nad pokretima skeleta, karakteristikama lica, pa čak i deformacijama mišića *Metahuman-a*. Ta razina kontrole omogućuje složene prilagodbe izraza lica, osiguravajući živopisne i ekspresivne izvedbe. Dodatno, *IK Rigging* (tehnika inverzne kinematike) igra ključnu ulogu u održavanju prirodnog kretanja i interakcije između dijelova tijela lika i okoline. Implementacijom IK rješenja, animatori mogu osigurati da se dijelovi tijela lika dinamično odazivaju na promjene ili vanjske sile, poboljšavajući realizam. Kombinirajući ove moćne alate unutar Unreal Engine-a, kreatori mogu stvoriti zanimljive i dinamične animacije za *MetaHuman* likove, oživljavajući ih fluidnim pokretima, nijansiranim izrazima lica i realističnim interakcijama u virtualnim okruženjima [29]. Na slici [Slika 14.] prikazana je upravljačka ploča odnosno *Control Rig* za lice *MetaHuman-a*.



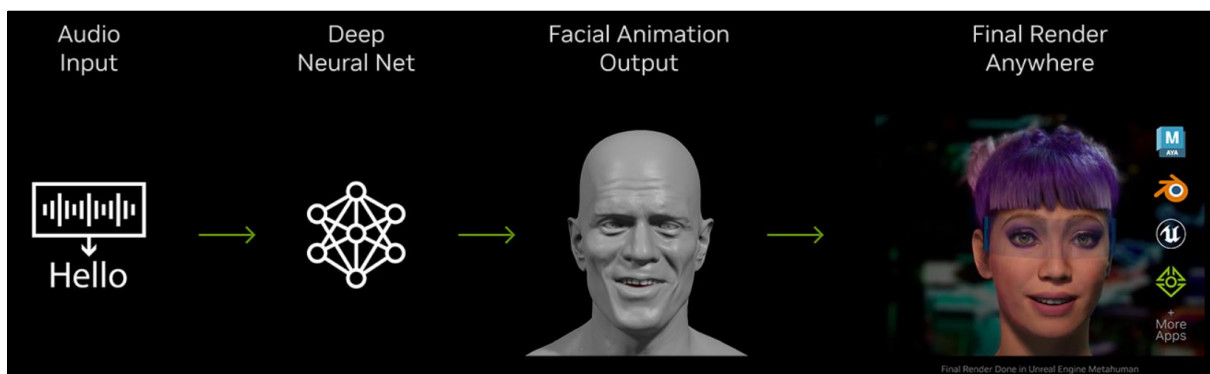
Slika 14 Upravljačka ploča za lice *Metahuman-a* [29]

Lice *Metahuman-a* može se animirati ručno, koristeći sekvencer i *Control Rig*, ili koristeći vanjske aplikacije i dodatke (eng. *plugins*), kompatibilne s Unreal Engine-om, koji su namijenjeni za to. Neki od popularnijih su aplikacija

Audio2Face i dodatak MetahumanSDK, koji su detaljnije opisani u nastavku ovog rada.

#### 4.2.1 Audio2Face aplikacija

Audio2Face (krat. A2F) je aplikacija koju je razvila Nvidia Corporation kao dio Omniverse kolaboracijske platforme. Ova aplikacija se ističe prvenstveno zbog svoje sposobnosti interaktivne animacije u stvarnom vremenu. Nadalje, aplikacija je primjenjiva na širok spektar jezika, iako je trenutno u razvoju s očekivanjem da će podržavati sve jezike u budućnosti. A2F dolazi s unaprijed definiranim likovima koji imaju svoje kontrole za pokretanje (eng. *rig*), što omogućuje jednostavnu primjenu na bilo koji 3D model, uključujući i neljudske likove. Korisnicima se omogućuje animacija lica koja je vođena govorom, kao i ručno upravljanje emocijama i njihovim intenzitetom. Konačno, aplikacija podržava prijenos animacije uživo, što znači da se može integrirati s vanjskim softverima i programima [30]. Princip pretvorbe zvučnog ulaza u animaciju lica provodi se korištenjem dubokih neuronskih mreža, a na slici [Slika 15.] prikazan je pojednostavljen izgled principa rada aplikacije.



Slika 15 Princip rada A2F aplikacije [30]

#### 4.2.2 *MetahumanSDK* dodatak

MetaHumanSDK je skup alata za stvaranje realistične interakcije s digitalnim ljudima. Pomoću te usluge stvaraju se animacije lica temeljene na glasovnom ili tekstualnom ulazu koristeći se dubokim neuronskim mrežama. Dodatak uključuje module povezivanja za stvaranje sintetiziranog glasa putem Google-a ili Azure-a te se nudi mogućnost stvaranja interaktivnog razgovora s povezivanjem na Google-ov „Dialogflow“ stvarajući mogućnost kreiranja dijaloga s digitalnim ljudima [31].

Značajke ovog dodatka nalaze se u nastavku:

- Realistična sinkronizacija usana na temelju zvukovnog ulaza,
- Podrška za različite emocije lica,
- Potpuna integracija s Unreal Engine *MetaHuman-ima* [31].

## 5. Povezivanje ChatGPT-a sa virtualnim bićem kroz UE5 okolinu

Svrha povezivanja ChatGPT-a sa virtualnim bićem je u tome da se navedeni alat za obradu prirodnog jezika u obliku *chatbota* dodatno „oživi“ na način da odgovore koje inače ispisuje tekstualno putem sučelja za *chat* sada producira putem virtualnog bića s kojim ga povežemo. Time bi se omogućilo da komunikacija između čovjeka i virtualnog bića nalikuje onoj između dva čovjeka. Kako je ChatGPT relativno nov alat koji se neprestano nadograđuje uz Unreal Engine koji također dobiva nadogradnje i proširenja dolazi do mnogo izmjena i novih rješenja za njihovo međusobno povezivanje. Za vrijeme pisanja ovog rada aktualan je ChatGPT-4 koji je dostupan uz određenu pretplatu te Unreal Engine 5 koji je dostupan svima i besplatan. Njihovo međusobno povezivanje aktualan je problem za koji postoji nekolicina rješenja te nije u potpunosti istražen.

Trenutno postoji nekoliko načina za povezivanje ChatGPT-a sa virtualnim bićem kroz UE5 okolinu, a oni su sljedeći:

- Razvoj novog Unreal Engine dodatka (eng. *plugin*) pomoću REST ili WebSocket API-ja,
- Korištenje postojećih Unreal Engine dodataka (eng. *plugins*),
- Integracija putem vanjske aplikacije.

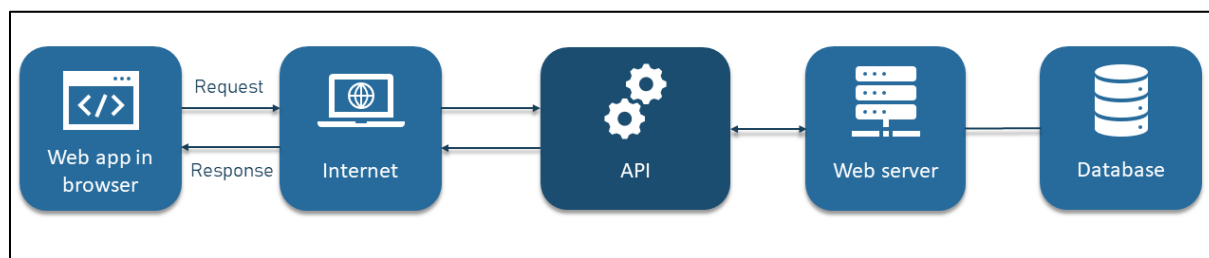
### 5.1 Aplikacijsko programsko sučelje

Aplikacijsko programsko sučelje (eng. akronim API, od *Application programming interface*) je softverski posrednik koji omogućuje da dvije aplikacije međusobno komuniciraju. API-ji su pristupačan način izdvajanja i dijeljenja podataka unutar i između sustava te se nalaze svuda oko nas. Primjena API-ja je široka, koristi se kod svih aplikacija za naručivanje određene usluge te

kod mobilnog plaćanja. Kada korisnik koristi jednu od takvih aplikacija one se povezuju na internet i šalju podatke na poslužitelj. Poslužitelj zatim dohvaća te podatke, tumači ih, izvodi potrebne radnje i šalje natrag na korisnikov telefon. Aplikacija zatim interpretira te podatke i na čitljiv način predstavlja informacije koje korisnik želi [32].

Tijekom godina API je preuzeo neke jedinstvene karakteristike. Moderni API-ji pridržavaju se specifičnih standarda (obično HTTP i REST), koji omogućavaju da API-ji budu prilagođeni programerima, samoopisani, lako dostupni i lako shvatljivi. Osim toga, danas se API-ji više tretiraju kao proizvod nego kao kod. Osmišljeni su za upotrebu za određenu publiku (npr. mobilne programere), a dokumentirani su i imaju verzije na način koji korisnicima omogućuje jasna očekivanja o njihovom održavanju i životnom ciklusu. Budući da su API-ji više standardizirani, oni se mogu nadzirati i njima se može upravljati [32].

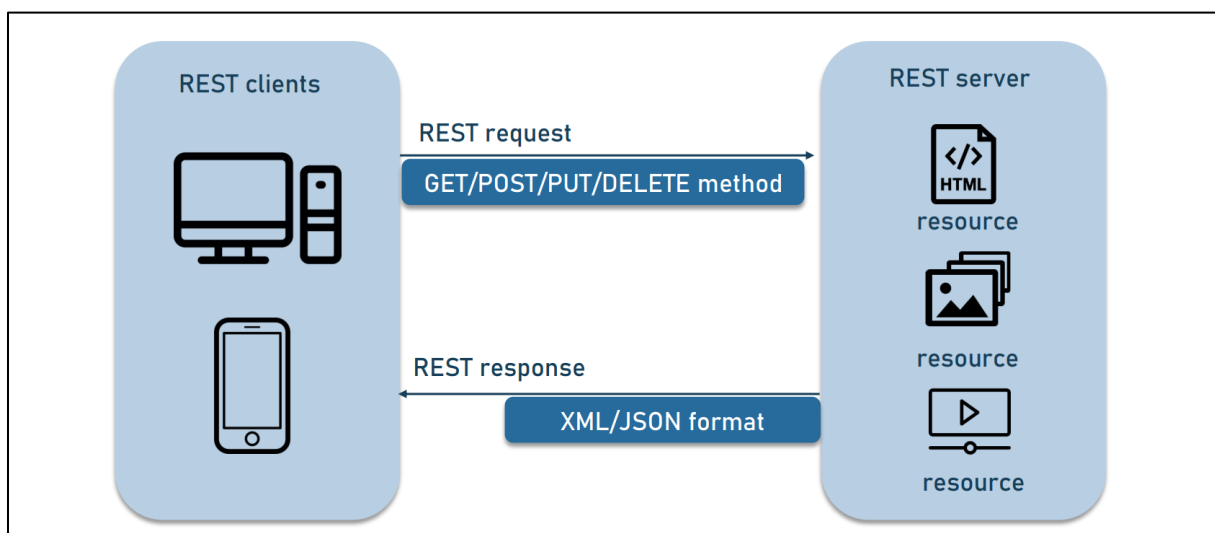
Koncept API-ja se može zamisliti kao naručivanje hrane u restoranu. Gost u restoranu pomoću jelovnika bira jelo po izboru, kaže konobaru što želi jesti te mu konobar iz kuhinje donese željeno jelo. U ovakvom scenariju konobar predstavlja Aplikacijsko programsko sučelje (API) koji je zapravo posrednik između gosta i kuhinje restorana [32]. Na slici [Slika 16.] prikazan je pojednostavljen način rada API-ja.



Slika 16 Shematski prikaz načina rada API-ja [33]

### 5.1.1 Restful API

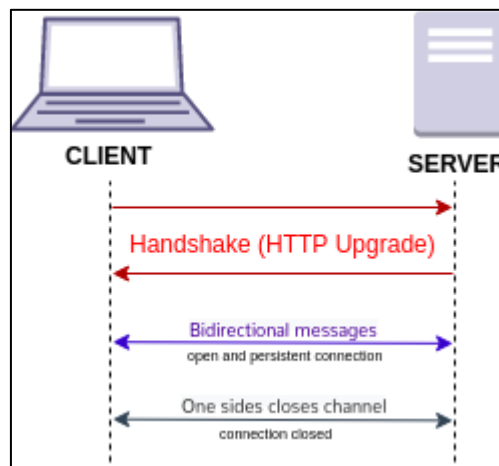
RESTful API je arhitektonski stil za sučelje aplikacijskog programa (API) koji koristi HTTP zahtjeve za pristup i korištenje podataka. Ti se podaci mogu koristiti za tipove podataka „GET“, „PUT“, „POST“ i „DELETE“, koji se odnose na čitanje, ažuriranje, stvaranje i brisanje operacija koje se tiču resursa. API za web stranicu je kod koji dvama softverskim programima omogućuje međusobnu komunikaciju. API navodi ispravan način na koji programer može napisati program koji zahtijeva usluge od operativnog sustava ili druge aplikacije. RESTful API, koji se također naziva RESTful web usluga ili REST API, temelji se na reprezentativnom prijenosu stanja (prema eng. akronimu REST, od *representational state transfer*), što je arhitektonski stil i pristup komunikaciji koji se često koristi u razvoju web usluga. REST tehnologija općenito ima prednost u odnosu na druge slične tehnologije. To je obično slučaj jer REST koristi manje propusnosti, što ga čini prikladnijim za učinkovito korištenje interneta. RESTful API-ji također se mogu izgraditi s programskim jezicima kao što su JavaScript ili Python [34]. Na slici [Slika 17.] prikazan je način rada REST API-ja.



Slika 17 Shematski prikaz načina rada REST API-ja [35]

### 5.1.2 WebSocket API

WebSocket API je napredna tehnologija koja omogućuje otvaranje dvosmjerne interaktivne komunikacijske sesije između korisničkog preglednika i poslužitelja, za razliku od REST API-ja koji prima i odgovara na zahtjeve. Pomoću ovog API-ja se mogu slati poruke poslužitelju i primiti odgovore vođene događajima bez potrebe da se traži odgovor od poslužitelja [36]. Na slici [Slika 18.] prikazan je dijagram koji opisuje vezu pomoću WebSocket-a.



Slika 18 Dijagram koji opisuje vezu pomoću WebSocket-a [37]

## 5.2 Unreal Engine dodaci

Unreal Engine dodaci (eng. *plugins*) su zbirke kodova i podataka koje programeri mogu lako omogućiti ili onemogućiti unutar sučelja za uređivanje projekta za svaki pojedinačni projekt. Ti dodaci mogu dodati funkcionalnosti tijekom korištenja Unreal Engine-a, modificirati ili dodati nove ugrađene značajke, stvoriti nove vrste datoteka i proširiti mogućnosti uređivača s novim izbornicima,

naredbama alatne trake i načinima rada. Mnogi postojeći UE podsustavi dizajnirani su da budu proširivi pomoću dodataka [38].

Postoje mnogobrojni dodaci koji se mogu koristiti u Unreal Engine-u te su dostupni u trgovini (eng. *marketplace*) u sklopu Epic Games aplikacije. U svrhu povezivanja ChatGPT-a s Unreal Engine-om dostupni su idući dodaci:

- HttpGPT,
- OpenAI API Plugin.

HttpGPT je Unreal Engine dodatak koji olakšava integraciju s OpenAI GPT uslugama (ChatGPT i DALL-E) putem asinkronih *REST* zahtjeva, olakšavajući programerima komunikaciju s tim uslugama. HttpGPT također uključuje uređivačke alate za integraciju Chat GPT i DALL-E sustava za generiranje slika izravno u *Engine* [39].

OpenAI API Plugin je dodatak koji omogućava programerima pristup OpenAI API-ju u Unreal Engineu. Kompatibilan je s 4.26, 4.27, 5.0, 5.1, 5.2 i 5.3 verzijama Unreal Engine-a. To je dodatak koji je kreirala zajednica te nema veze sa tvrtkom OpenAI [40].

Ukoliko je potrebno da virtualno biće u okolini UE5 dobiva izgovorene upite od korisnika te izgovara dobivene odgovore od ChatGPT-a, potrebno je koristiti i dodatke koji omogućuju pretvaranje govora u pisani tekst i obrnuto. Za takve potrebe postoji nekolicina dodataka kao što su AzSpeech i Amazon Polly.

## 5.3 Mogućnosti povezivanja Chat GPT-a s Unreal Engine-om

### 5.3.1 Razvoj novog Unreal Engine dodatka pomoću API-ja

Jedna od opcija za povezivanje Chat GPT-a s UE je stvaranje prilagođenog dodatka unutar Unreal Engine-a koji bi omogućivao direktnu komunikaciju



između ChatGPT API-ja i sustava dijaloga virtualnog agenta te direktnu komunikaciju između korisnika i virtualnog agenta. To bi bilo moguće izvesti korištenjem REST ili WebSocket API-ja.

Moguće je razviti RESTful API koji bi omogućavao komunikaciju između ChatGPT-a i virtualnog agenta u Unreal Engine-u. Kroz ovaj pristup mogao bi se uspostaviti poslužitelj koji prima zahtjeve iz UE5 okoline i šalje ih ChatGPT modelu. Virtualni agent u UE5 može koristiti *HTTP* zahtjeve prema API-ju kako bi slao korisničke upite i primao odgovore.

Druga opcija je uspostaviti *WebSocket* vezu između UE5 okruženja i ChatGPT-a. U ovom pristupu, razvija se *WebSocket* poslužitelj koji omogućuje komunikaciju. Virtualni agent u UE5 može se povezati na *WebSocket* poslužitelj i slati poruke ChatGPT-u radi obrade. Poslužitelj prima poruke od virtualnog agenta, šalje ih ChatGPT modelu i vraća odgovore u UE5.

Pomoću ove dvije opcije omogućilo bi se povezivanje ChatGPT-a i Unreal Engine-a, no ukoliko je potrebno da virtualni agent prima glasovne naredbe te izgovara dobivene odgovore potrebno je uz navedeno omogućiti i pretvaranje govora u tekst te teksta u govor. Kroz razvoj dodatka, tj. *plugina*, može se stvoriti most kojim se olakšava komunikacija između ChatGPT-a i virtualnog agenta u UE5. Za ovakvu opciju potrebno je posjedovati znanje u području razvoja API-ja.

### 5.3.2 Korištenje postojećih Unreal Engine dodataka

Alternativno razvoju novog UE dodatka, moguće je koristiti već razvijene dodatke, tj. *pluginove*, dostupne za Unreal Engine 5, kao što su HttpGPT ili OpenAI API Plugin. Ovakvi tipovi dodataka pružaju sustav dijaloga za virtualnog agenta i nude mogućnosti integracije s raznim platformama, u ovom slučaju sa ChatGPT-om. Kroz konfiguraciju takvog *plugina*, moguće je povezati virtualnog

agenta u UE5 okolini sa ChatGPT-om. Ova opcija je najpristupačnija od svih navedenih te se na najbrži i najjednostavniji način može dobiti željeni rezultat. Uz navedene dodatke, tj. *pluginove*, da bi se u potpunosti ostvario razgovor između čovjeka i virtualnog agenta potrebno je koristiti i AzSpeech ili Amazon Polly dodatke za pretvaranje teksta u govor i obrnuto.

Najveća mana korištenja gotovih *pluginova* je u njihovoj sinkronizaciji sa trenutno dostupnim verzijama ChatGPT-a i Unreal Engine-a jer u slučaju određenih ažuriranja ili stvaranja novih verzija može doći do prestanka rada navedenih *pluginova*. Unatoč tomu, *pluginovi* također dobivaju novije verzije i ažuriranja te bi se s vremenom trebali ažurirati u ovisnosti o promjenama.

### 5.3.3 Integracija putem vanjske aplikacije

Ova opcija uključuje razvoj zasebne aplikacije koja povezuje UE5 okolinu i ChatGPT. Vanjska aplikacija bi trebala imati mogućnost komunicirati sa ChatGPT-om putem dostupnih API-ja ili biblioteka. Zatim bi se trebao uspostaviti komunikacijski kanal između virtualnog agenta u UE5 okolini i vanjske aplikacije, kao što su međuprocena komunikacija ili mrežni protokoli. Tako razvijena vanjska aplikacija bi imala mogućnost upravljati integracijom, omogućujući prijenos poruka između Unreal Engine-a i ChatGPT-a. Također u aplikaciju bi trebalo implementirati mogućnost pretvaranja teksta u govor i obrnuto. Ovo je ujedno i najsloženija opcija od svih navedenih te za razvoj takve aplikacije treba najviše vremena i potrebnih znanja poput programiranja u pythonu, C++ ili sličnom programskom jeziku te znanja iz mrežnog programiranja uz razumijevanje TCP/IP protokola.

## 6. Izvedba zadatka

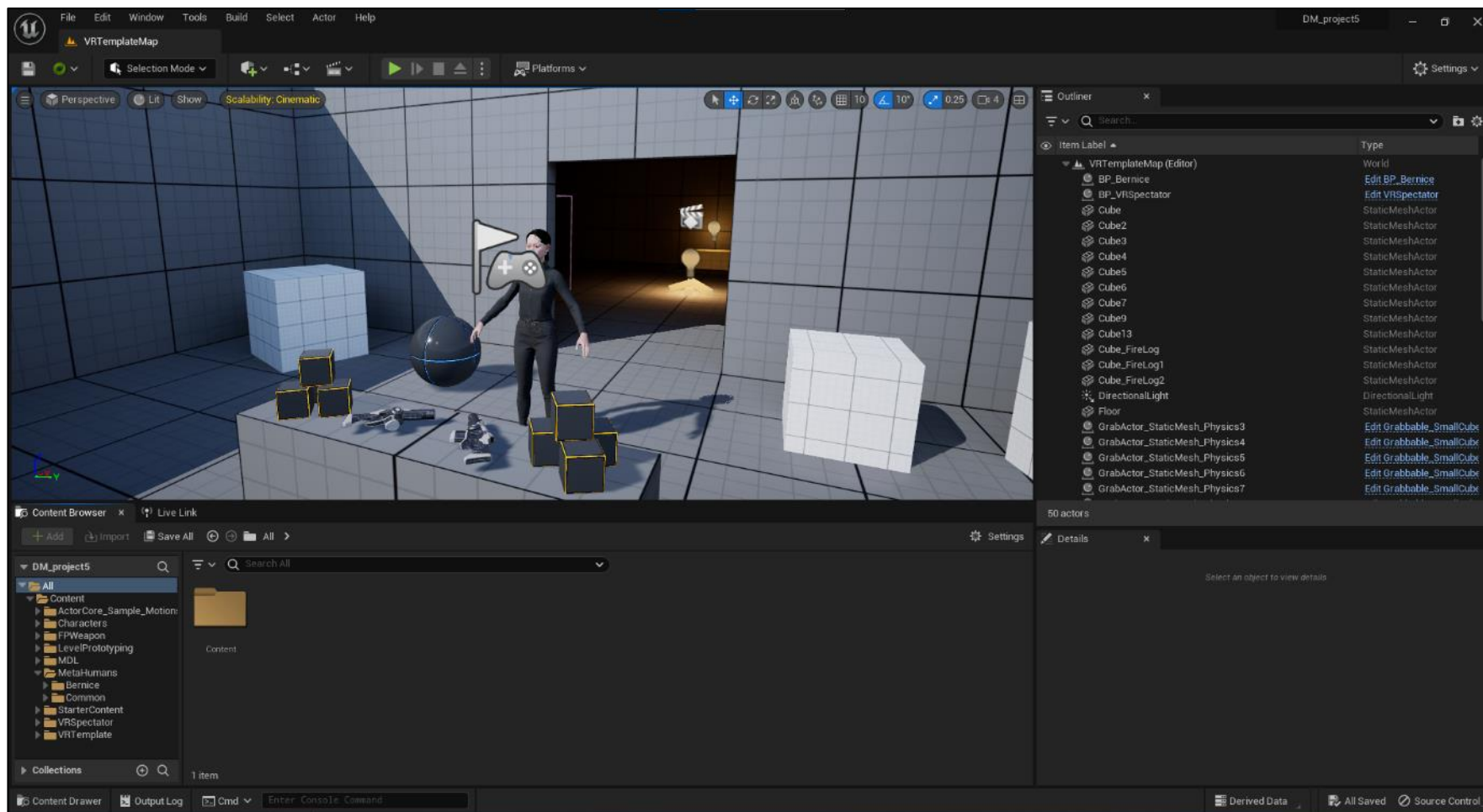
U svrhu izvedbe ovog zadatka bilo je potrebno povezati ChatGPT s UE5 okolinom, omogućiti TTS i STT te animirati lice virtualnog agenta na temelju izgovorenog. Za implementaciju ChatGPT-a korišten je OpenAI dodatak. OpenAI je odabran zbog toga što je u potpunosti besplatan te nudi sve funkcije koje su potrebne za izradu ovog rada. Za pretvaranje govora u tekst i teksta u govor korišten je AzSpeech od Microsoft Azure-a jer nudi podršku za sve jezike i pruža mogućnosti kreiranja prirodnog glasa koji zvuči potpuno kao ljudski. Ovaj dodatak ima ograničeno korištenje u sklopu besplatne pretplate, ali pruža dovoljno resursa za testiranje virtualnog agenta na mjesečnoj bazi. Za animiranje lica virtualnog agenta korištena su dva rješenja, u jednom je animacija stvorena ručno na način da se imitira govor jednostavnim animacijama otvaranja i zatvaranja usta, dok je u drugom implementirana programska podrška MetahumanSDK dodatka koji omogućava precizne animacije lica na temelju izgovorenog. Kod testiranja funkcionalnosti navedenih projekata korišteni su različiti *Metahumani*, jedan iz biblioteke već unaprijed generiranih *Metahumana* te jedan ručno kreiran. Svi navedeni postupci u nastavku primjenjivi su i na ranije spomenutog PLEA virtualnog agenta razvijenog na Fakultetu strojarstva i brodogradnje u Zagrebu. U nastavku ovog rada detaljnije je opisan postupak izvedbe zadatka.

### 6.1 Virtualni agent s jednostavnim animacijama govora

Unatoč tome što postoje određene aplikacije i dodaci poput ranije spomenutih A2F i MetahumanSDK, pojavljuju se određeni nedostaci koji onemogućavaju njihovo korištenje. A2F aplikacija se prvenstveno koristi za animiranje lica virtualnih ljudi s unaprijed definiranim audio zapisima te trenutno nema

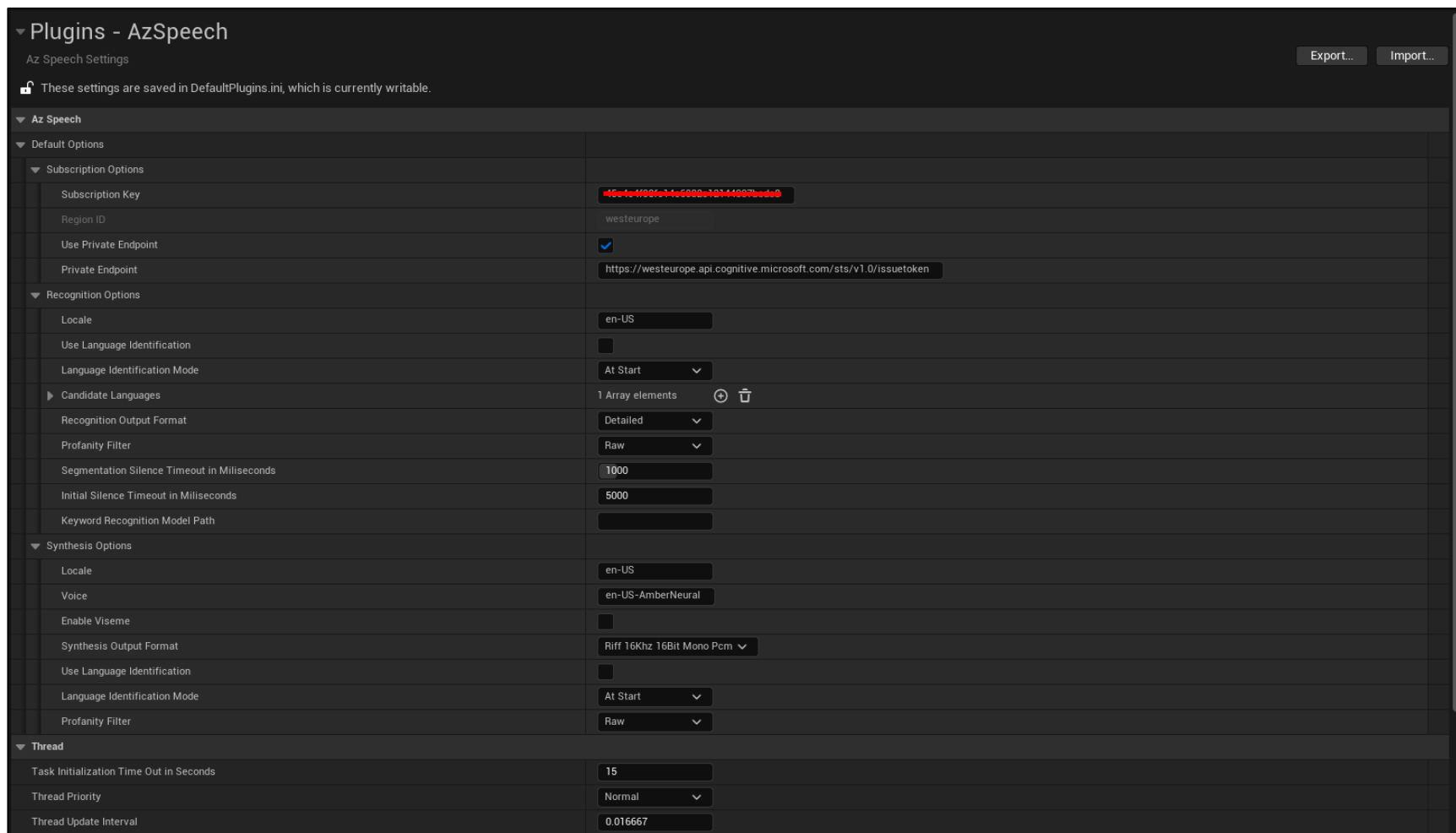
razrađenih opcija za povezivanje s trenutno stvorenim glasom od strane *chatbota*. Dok, s druge strane, *MetahumanSDK* dodatak posjeduje tu mogućnost, ali ima sustav naplate korištenja usluge prema trajanju korištenja animacija. Zbog navedenih razloga, u svrhu potpuno besplatne konverzacije sa virtualnim agentom, razvijena je i ova verzija virtualnog agenta koji prilikom izgovaranja odgovora koristi jednostavne animacije otvaranja i zatvaranja usta.

Za početak, potrebno je postaviti okruženje unutar *Unreal Engine-a*, dodati *Metahuman-a* koji predstavlja virtualnog agenta te namjestiti sve postavke implementiranih dodataka. Na slici [Slika 19.] prikazano je sučelje programa s postavljenim virtualnim agentom unutar poligona virtualne stvarnosti.



Slika 19 Sučelje UE5 – virtualni agent u virtualnoj stvarnosti

Dodaci koji su korišteni u ovom projektu su OpenAI i AzSpeech. OpenAI se instalira ručno, ubacivanjem datoteka unutar mape od projekta, dok se AzSpeech dodatak instalira pomoću Epic Games softvera. Da bi projekt funkcionirao, potrebno je podesiti određene postavke AzSpeech dodatka unutar Unreal Engine-a. U postavkama se određuje jezik koji se koristi prilikom prepoznavanja govora za STT i TTS te se bira glas agenta prilikom izgovaranja dobivenih odgovora. Uz to potrebno je unijeti i ključ s kojim se povezuje na online servis od Microsoft Azure-a. Na slici [Slika 20.] prikazane su unesene postavke za AzSpeech dodatak.



Slika 20 AzSpeech postavke

### 6.1.1 Implementacija ChatGPT-a, STT-a i TTS-a

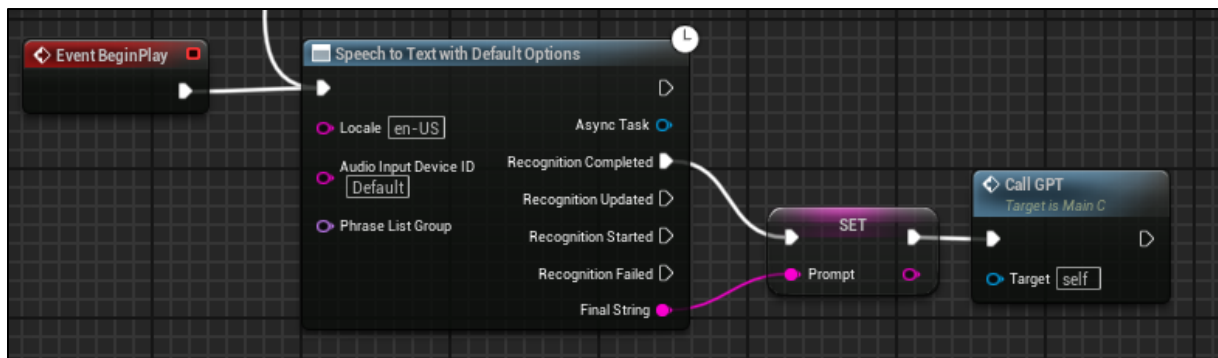
Nakon što su uspješno implementirani dodaci i podešene sve potrebne postavke, potrebno je unutar shematskih planova (eng. *blueprints*) postaviti i isprogramirati na funkcionalan način poveznice između različitih funkcija odabranih dodataka. Za početak potrebno je unijeti OpenAI Api ključ koji povezuje servis ChatGPT-a sa Unreal Engine-om. Na slici [Slika 21.] prikazano je postavljanje Api ključa.



Slika 21 Open AI Api ključ

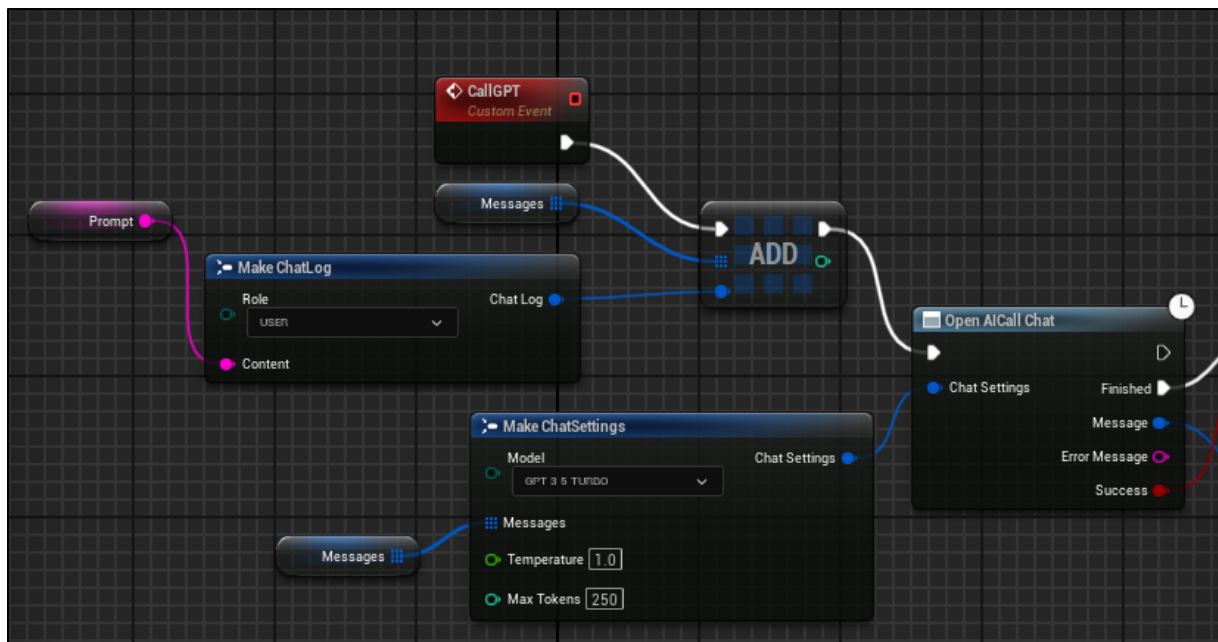
Nakon postavljanja Api ključa, u novom shematskom planu potrebno je povezati sve funkcionalnosti od pretvorbe govora u tekst, pozivanja ChatGPT-a do pretvorbe teksta u govor te na kraju pokretanje animacija lica virtualnog agenta. Na slici [Slika 22.] prikazan je poziv funkcije za pretvaranje govora u tekst, određivanje jezika koji je potrebno prepoznati unutar funkcije, postavljanje nove varijable „Prompt“, koja označava tekst generiran na temelju govora, te pozivanje „Call GPT“ funkcije koja je detaljnije opisana u nastavku.





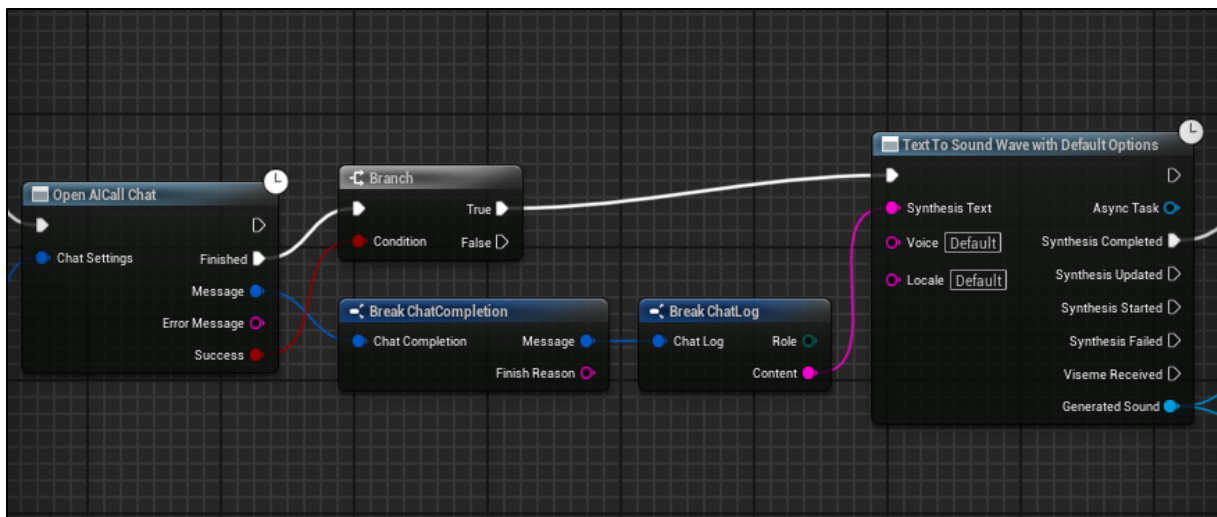
Slika 22 Funkcija za pretvaranje govora u tekst i poziv funkcije „Call GPT“

Nakon što se generira tekst prepoznat od strane zvuka korisnika, potrebno je pohraniti taj tekst u obliku upita za ChatGPT putem „Make ChatLog“ funkcije. Uz tekst primljen od strane korisnika, moguće je i dodati unaprijed generirana pravila i upute koje virtualni agent može slijediti putem varijable „Messages“. Varijable „Prompt“ i „Messages“ se pohranjuju te zajedno šalju prema ChatGPT-u koji zatim daje određeni odgovor na upit korisnika. Na slici [Slika 23.] Prikazano je definiranje „CallGPT“ funkcije te pohranjivanje poruka koje se šalju prema ChatGPT-u putem funkcija „Add“, „Make ChatLog“, „Make ChatSettings“ i „Open AiCall Chat“. Unutar „Make ChatSettings“ funkcije definira se verzija ChatGPT-a koji se koristi, u ovom slučaju to je verzija 3.5.



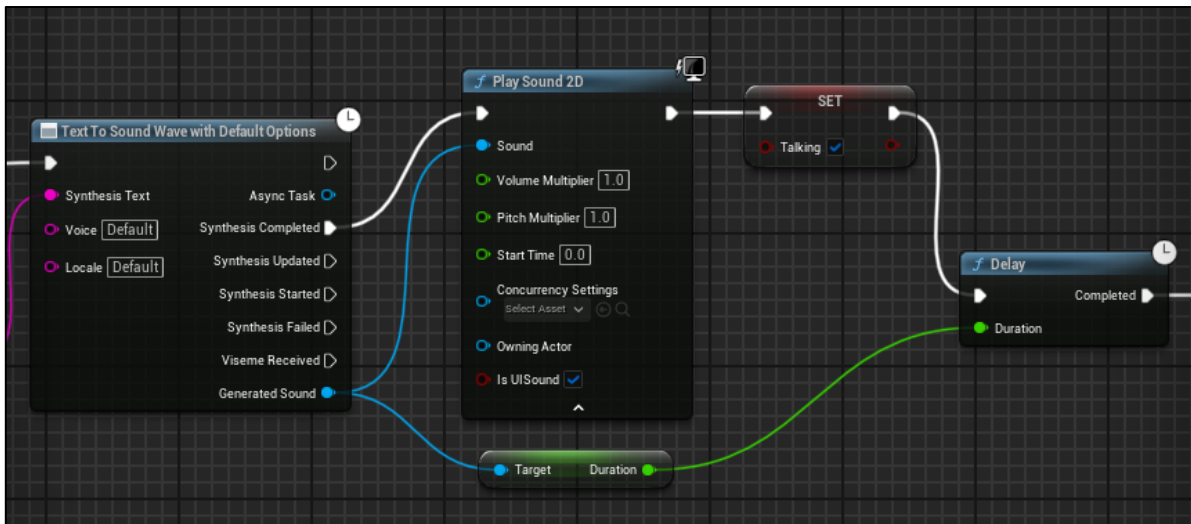
Slika 23 Slanje upita ChatGPT-u

Nakon što je poslan upit ChatGPT-u potrebno je „Branch“ funkcijom provjeriti je li završilo generiranje odgovora od strane ChatGPT-a te ukoliko je, proslijediti tu poruku u funkciju za pretvorbu teksta u govor. Na slici [Slika 24.] prikazane su funkcije procesuiranja i pohranjivanja odgovorenih poruka od strane ChatGPT-a „Break ChatCompletion“ i „Break ChatLog“ te funkcija pretvorbe generiranog teksta u zvuk pomoću „Text to Sound Wave“ funkcije.



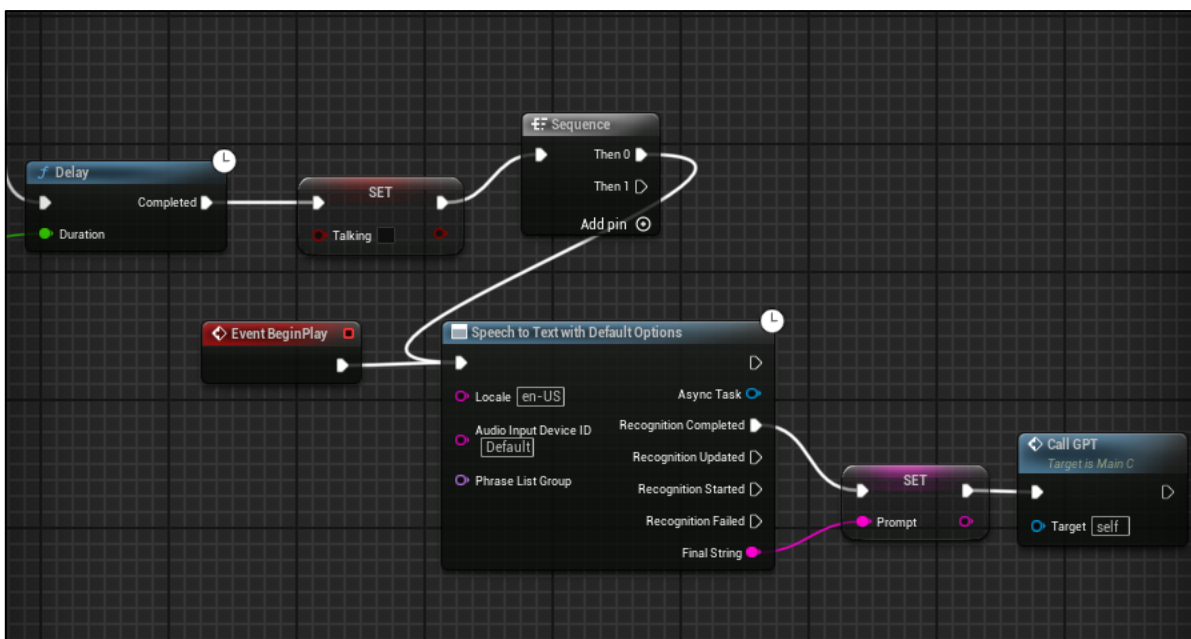
Slika 24 Pretvaranje generiranog teksta u govor

Nakon što je generiran zvuk koji odgovara tekstu generiranog od strane ChatGPT-a potrebno je pokrenuti taj zvuk kao odgovor korisniku. Taj dio odrađuje se funkcijom „Play Sound 2D“ . Dodatno stvara se nova varijabla „Talking“ koja definira da li je zvuk pokrenut ili nije, što će se kasnije koristiti za pokretanje ili zaustavljanje animacije govora virtualnog agenta. Varijabla „Talking“ se postavlja u stanje „True“ te se na kraju dodaje funkcija „Delay“ koja odgađa daljnje akcije dok se ne izgovori cijeli tekst stvoren od strane Chat GPT-a. Na slici [Slika 25.] Prikazane su funkcije za pretvaranje teksta u govor, pokretanje zvuka te funkcija kojom se odgađaju daljnje akcije dok se ne izgovori tekst u potpunosti.



Slika 25 Generiranje zvuka na temelju generiranog teksta od strane ChatGPT-a

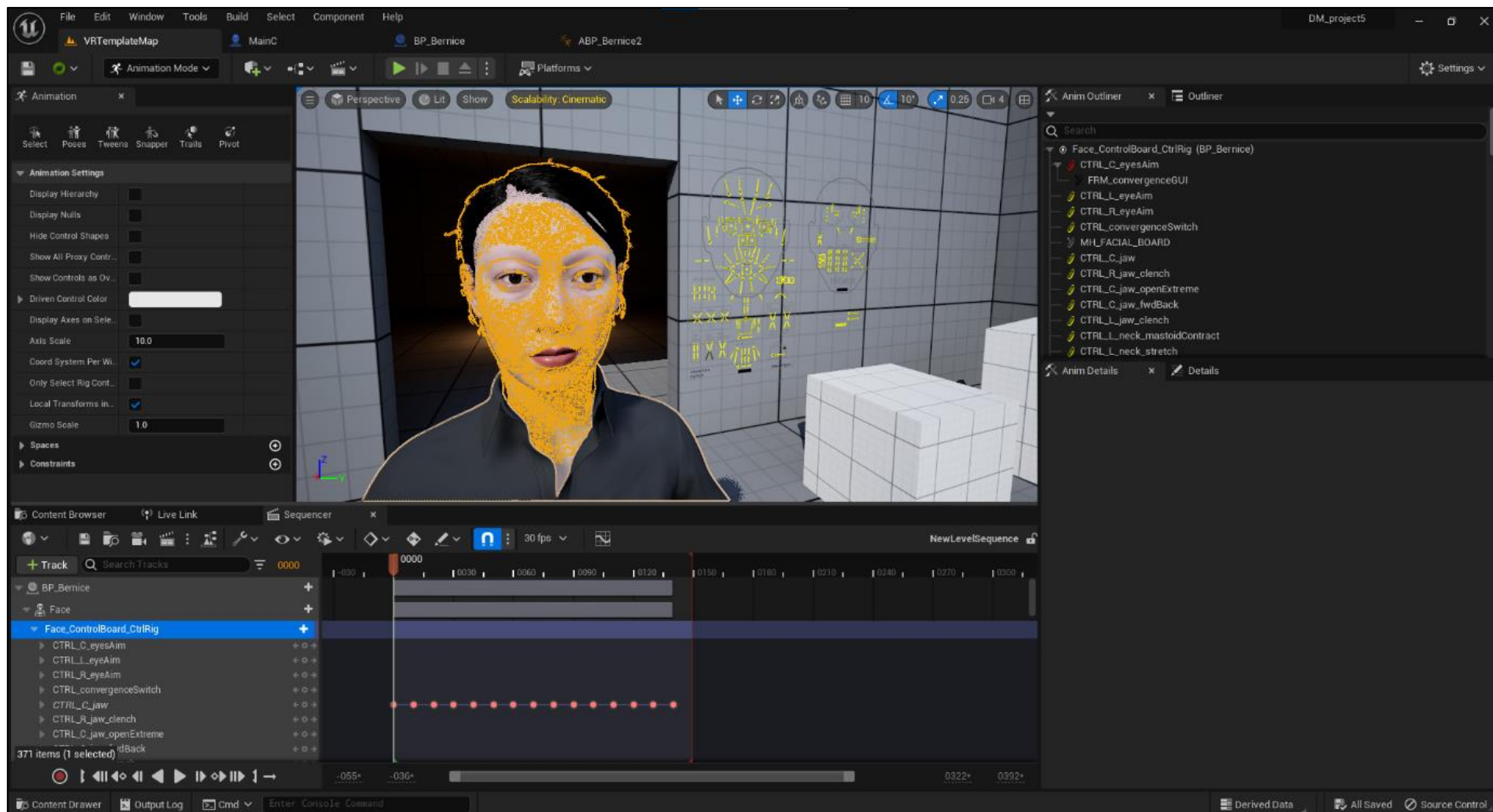
Kada virtualni agent završi s glasovnim odgovorom, „Talking“ varijabla se postavlja u stanje „False“ i program se pokreće ponovno te je spreman preuzeti nove glasovne naredbe od korisnika. Na slici [Slika 26.] prikazan je završetak izvođenja programa te ponovno pokretanje u svrhu preuzimanja glasovnih naredbi.



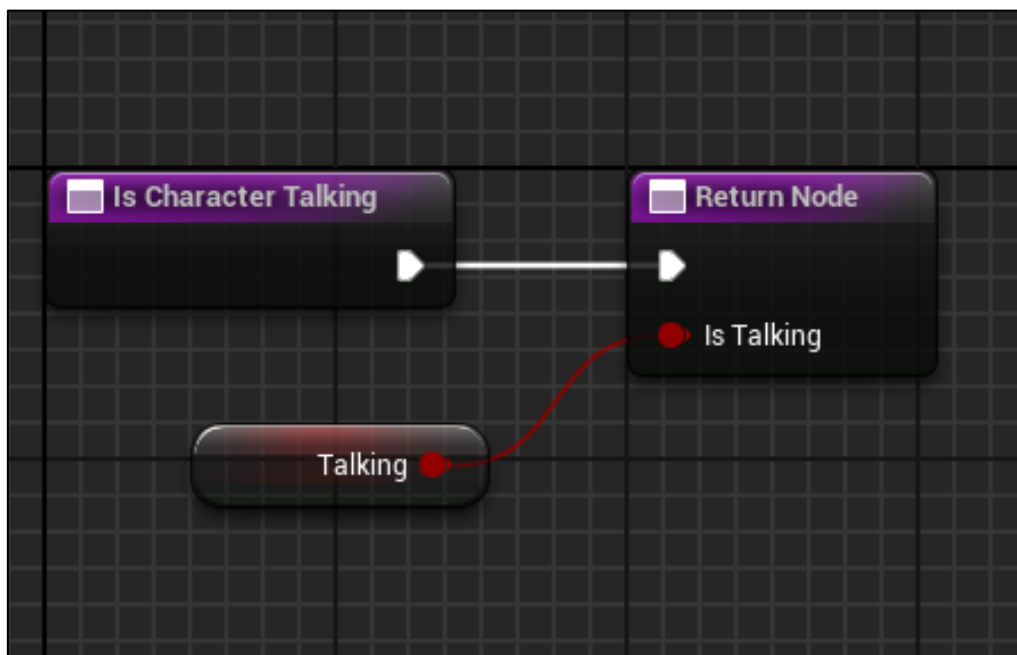
Slika 26 Petlja kojom se ponovno pokreće program

### 6.1.2 Animiranje lica virtualnog agenta

Da bi se pokrenule animacije govora tokom izgovaranja odgovora virtualnog agenta, potrebno je kreirati jednostavne animacije otvaranja i zatvaranja usta pomoću Unreal Engine sekvencera i *Control Rig-a*. Na slici [Slika 27.] prikazan je pokrenut sekvencer i *Control Rig* lica *Metahumana*. Unutar sekvencera se podešava, u određenom vremenskom rasponu, animacije kontrolnih točaka lica u svakom trenutku trajanja animacije.

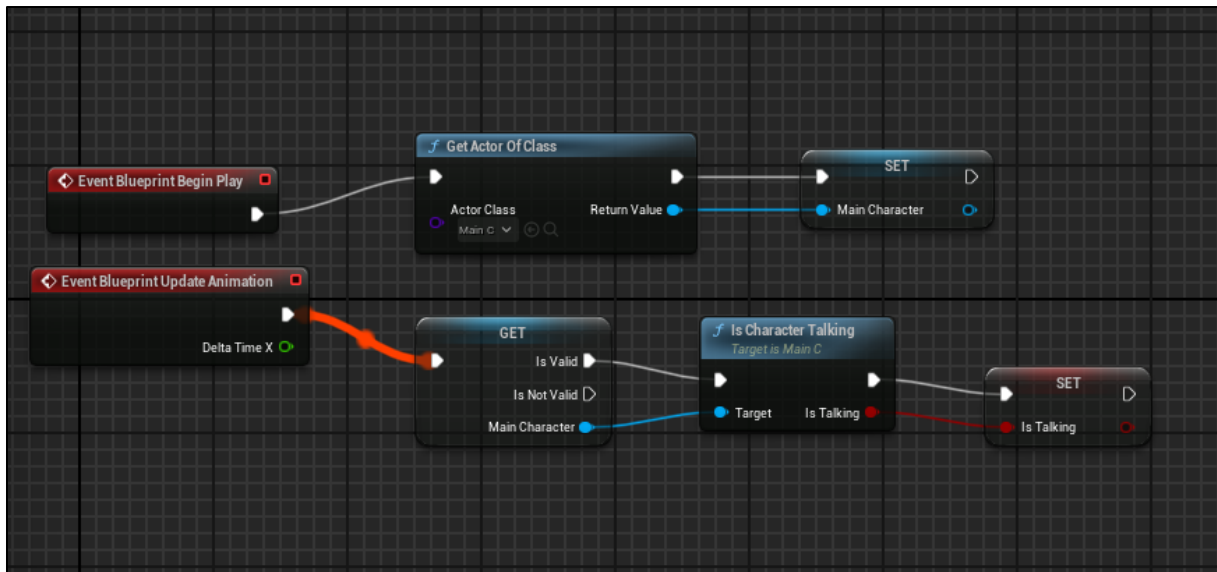
Slika 27 Sekvencer i *ControlRig*

Nakon što je spremljena jednostavna animacija otvaranja i zatvaranja usta potrebno je u shematskim planovima stvoriti novu funkciju „Is Character Talking“ u kojoj se preuzima stanje varijable „Talking“ te se kasnije poziva unutar animacijskih shematskih planova za svrhe pokretanja ili zaustavljanja animacija govora *Metahumana*. Na slici [Slika 28.] prikazana je funkcija „Is Character Talking“.



Slika 28 Funkcija „Is Character Talking“

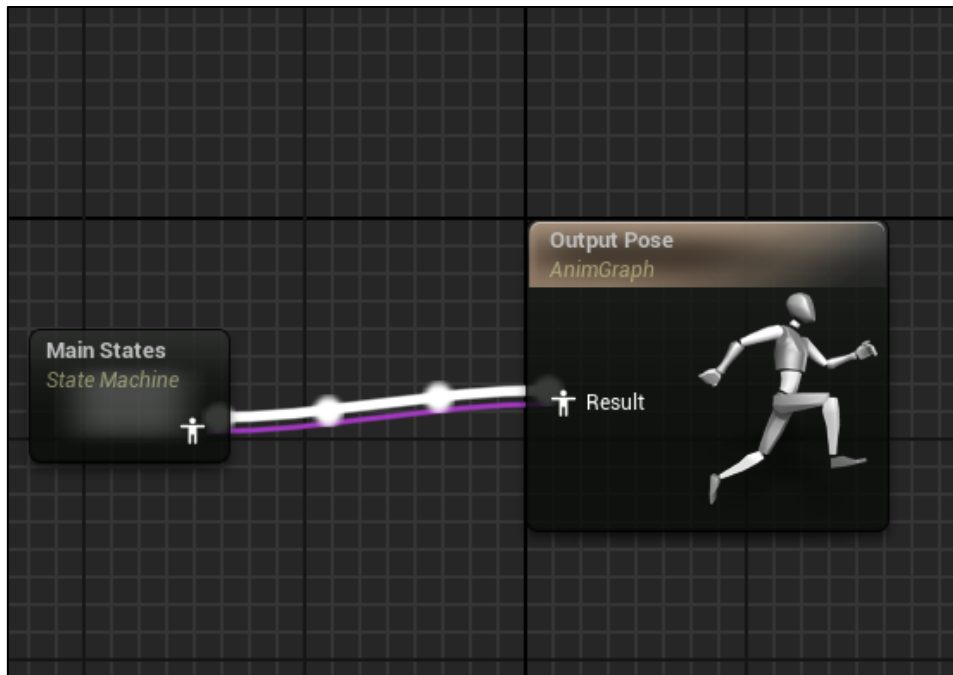
Pored klasičnih shematskih planova, moguće je koristiti i animacijske shematske planove, čija je svrha pokretanje i upravljanje animacijama unutar Unreal Engine simulacije. Na slici [Slika 29.] prikazan je poziv funkcije „Is Character Talking“ unutar animacijskog shematskog plana. Uz to, prikazan je uvjet koji određuje kada se pokreće provjera govori li virtualni agent. Taj uvjet je definiran pomoću „Get Actor Of Class“ funkcije koja sugerira programu da se pojavio korisnik unutar virtualnog okruženja.



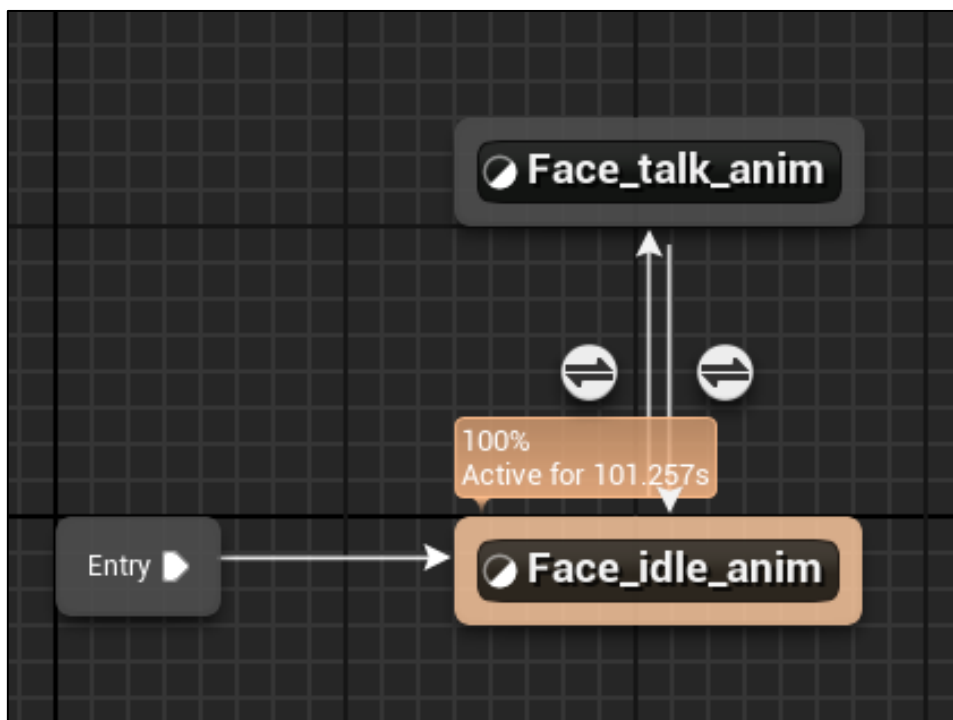
Slika 29 Poziv „Is Character Talking“ funkcije

Unutar animacijskih shematskih planova potrebno je postaviti „State Machine“ opciju kojom se upravlja stanjima *Metahumana* u određenim trenucima. Unutar „State Machine-a“ pozivaju se definirane animacije koje su kreirane u sekvenceru uz određene uvjete kada se točno koja poziva. U slučaju da je prijašnje spomenuta varijabla „Talking“ u stanju „True“, pokreće se animacija govora i obrnuto. Na slikama [Slika 30.], [Slika 31.], [Slika 32.] i [Slika 33.] prikazan je način funkcioniranja „State Machine-a“ te uvjeti u kojima se pokreće animacija govora ili mirovanja virtualnog agenta.

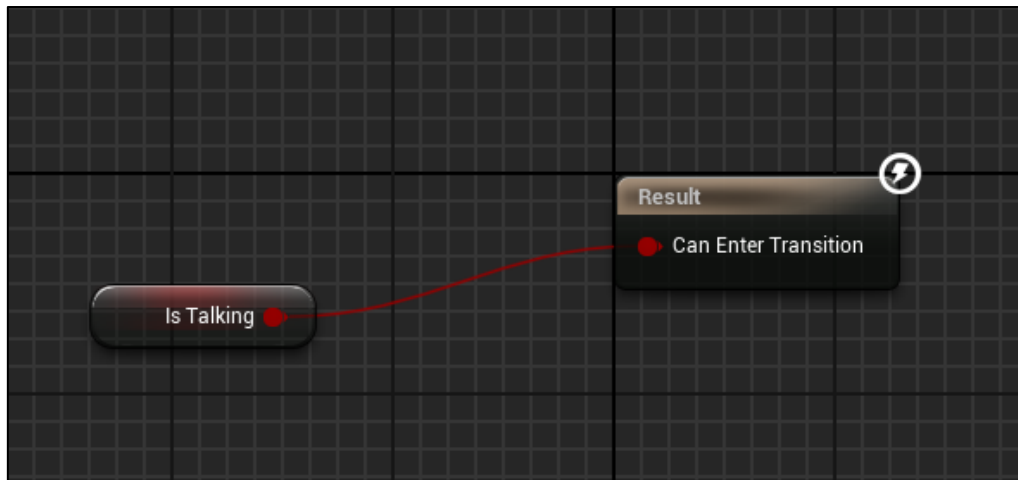




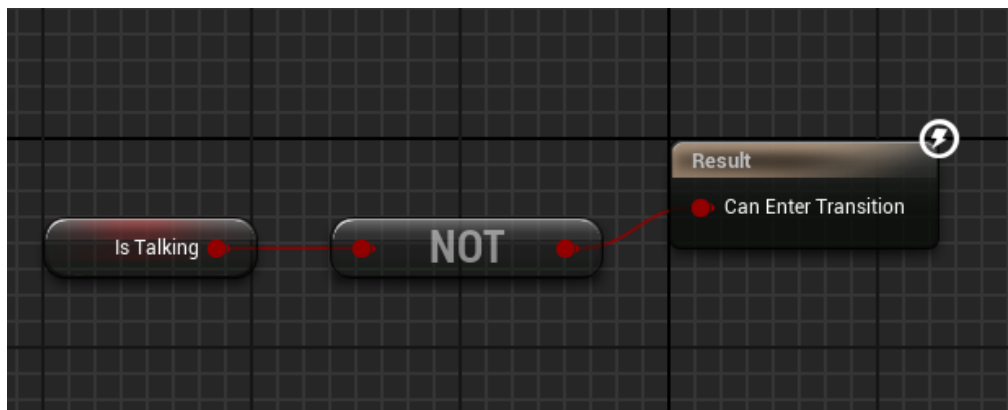
Slika 30 Pokretanje animacije virtualnog agenta prema stanju „State Machine-a“



Slika 31 Postavke „State Machine-a“

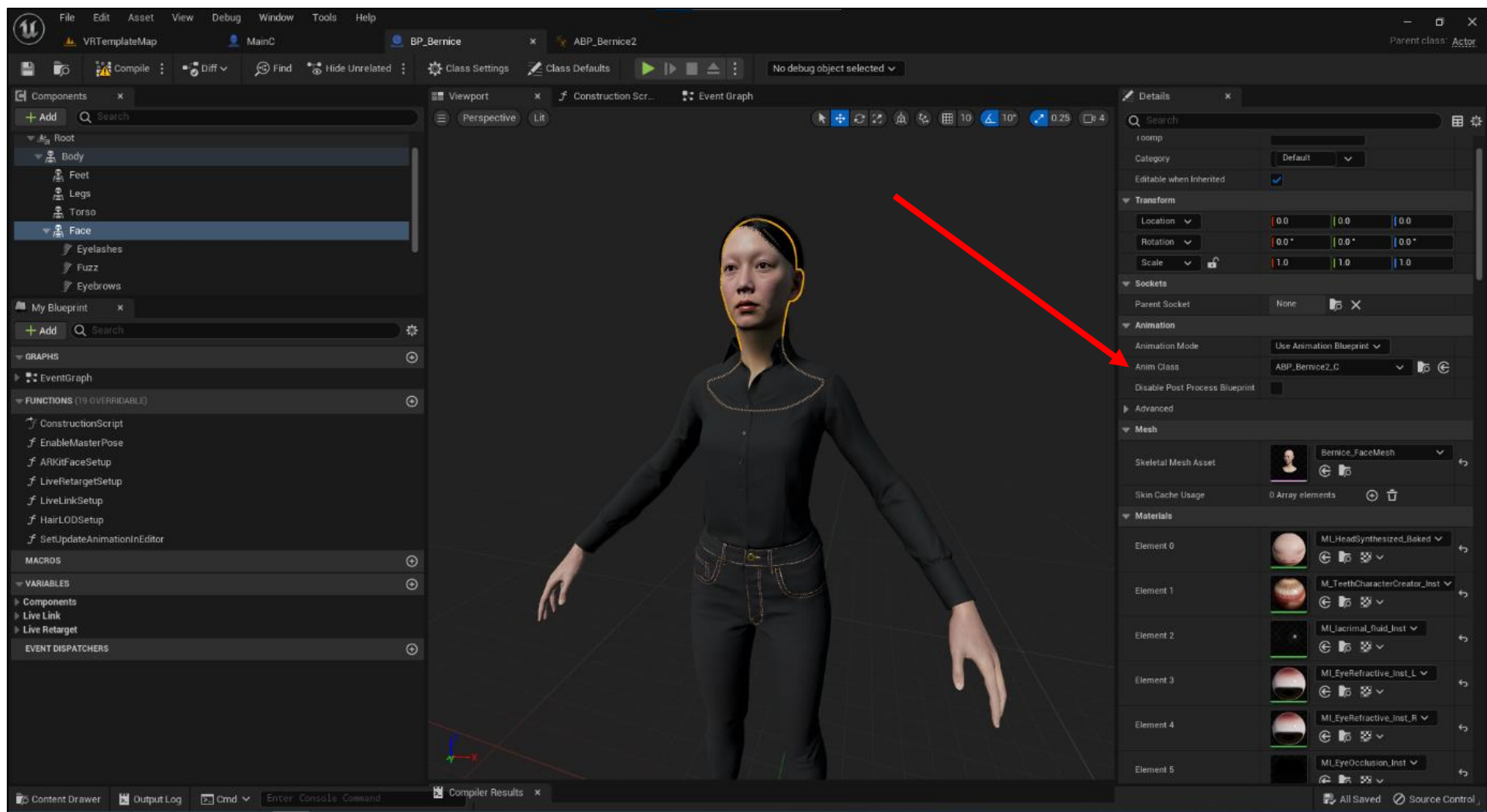


Slika 32 Uvjet za pokretanje animacije govora



Slika 33 Uvjet za pokretanje animacije mirovanja

Nakon što je u potpunosti postavljen animacijski shematski plan, potrebno je u postavkama *Metahumana* primijeniti tu animaciju uz njegovo pripadajuće lice. Preciznije, pod postavkom „Anim Class“, potrebno je odabrati animacijski shematski plan u kojem smo definirali uvjete govora i mirovanja *Metahumana*. Na slici [Slika 34.] prikazane su postavke lica *Metahumana*.



Slika 34 Metahuman postavke lica

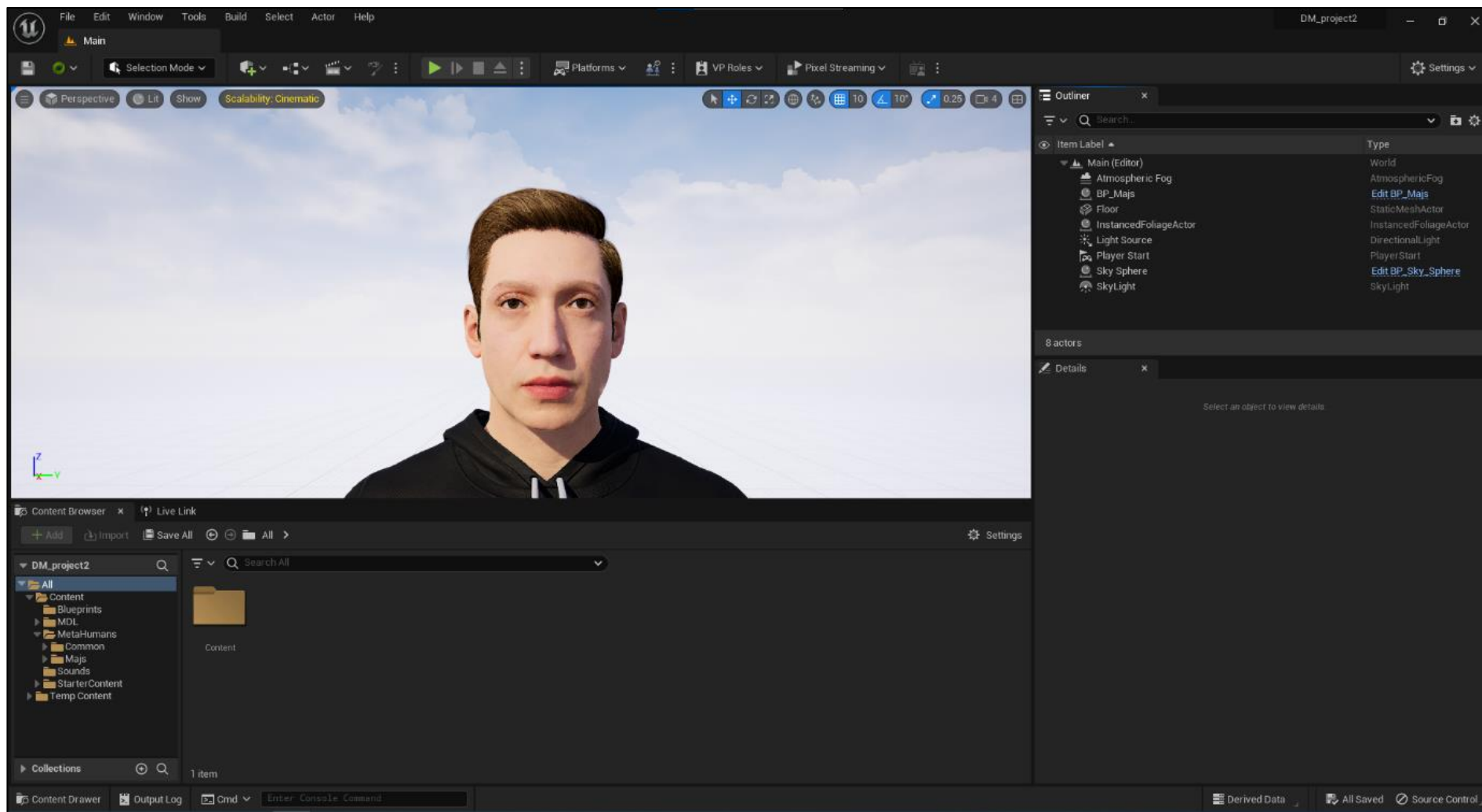
Nakon što su svi shematski planovi pohranjeni zajedno sa svim potrebnim postavkama, pokretanjem simulacije moguće je komunicirati verbalno sa virtualnim agentom koji prilikom izgovaranja odgovora koristi jednostavne animacije otvaranja i zatvaranja usta.

Na idućoj poveznici prikazan je test konverzacije između čovjeka i virtualnog agenta: <https://www.youtube.com/watch?v=KHJs9sFHDe0>.

## 6.2 Virtualni agent s preciznim animacijama govora

Pored ranije prikazane izvedbe zadatka s jednostavnim animacijama govora, razvijen je i virtualni agent koji ima mogućnost preciznog izgovaranja riječi. Na taj način može se samo s iščitavanjem sa usana prepoznati što virtualni agent govori. Kao i u prethodnom slučaju za implementaciju ChatGPT-a korišten je OpenAI dodatak, za STT i TTS korišten je AzSpeech dodatak, te u ovom slučaju dodatno, za animiranje lica virtualnog agenta korišten je MetahumanSDK dodatak.

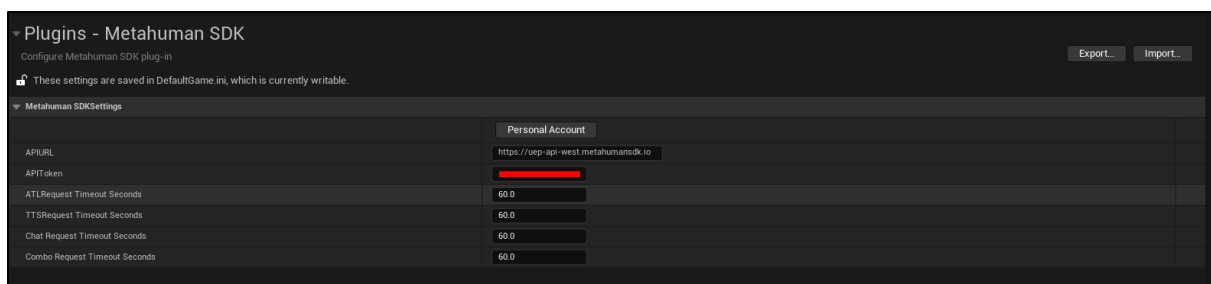
Za početak, potrebno je kao i u prethodnom projektu, instalirati potrebne dodatke, postaviti okruženje unutar Unreal Engine-a, dodati *Metahuman-a* koji predstavlja virtualnog agenta te namjestiti sve postavke implementiranih dodataka. Na slici [Slika 35.] prikazano je sučelje programa s postavljenim virtualnim agentom.



Slika 35 UE5 Sučelje sa postavljenim virtualnim agentom

Da bi projekt funkcionirao, potrebno je podesiti određene postavke AzSpeech dodatka unutar Unreal Engine-a na isti način kao u prethodnom projektu.

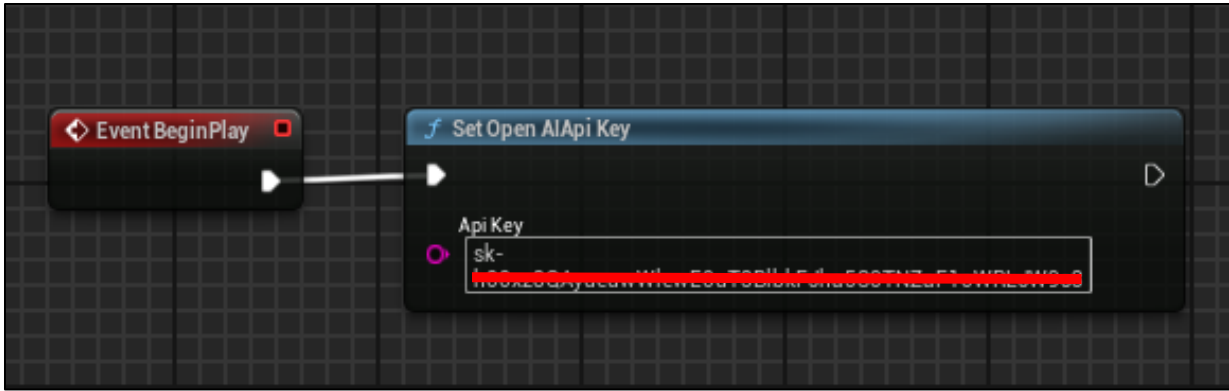
Pored AzSpeech dodatka, potrebno je postaviti postavke za MetahumanSDK dodatak, odnosno unijeti ključ kojim se povezuje s aktiviranom pretplatom od tog online plaćenog servisa. Na slici [Slika 36.] prikazane su unesene postavke za MetahumanSDK dodatak.



Slika 36 MetahumanSDK postavke

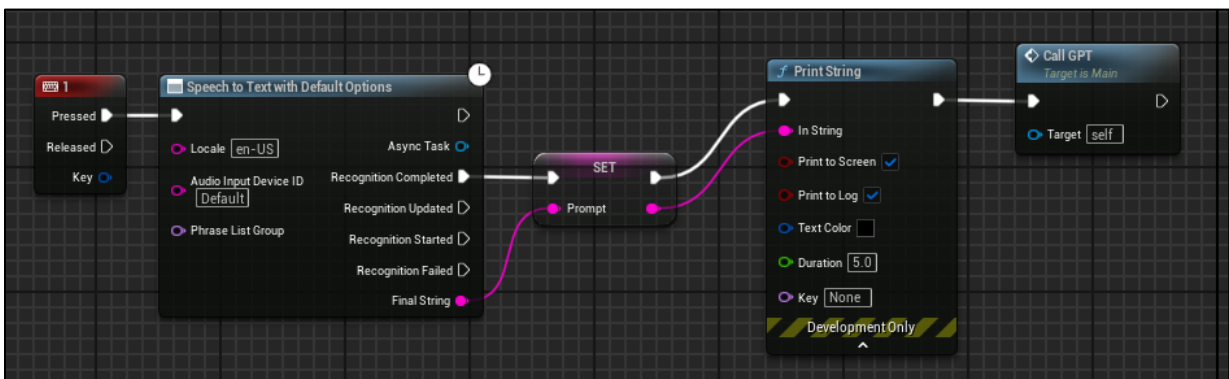
### 6.2.1 Implementacija ChatGPT-a, STT-a i TTS-a

Nakon što su uspješno implementirani dodaci i podešene sve potrebne postavke, potrebno je unutar shematskih planova postaviti i isprogramirati na funkcionalan način poveznice između različitih funkcija odabranih dodataka. Za početak potrebno je, poput prethodnog projekta, unijeti OpenAI Api ključ koji povezuje servis ChatGPT-a sa Unreal Engine-om. Na slici [Slika 37.] prikazano je postavljanje Api ključa.



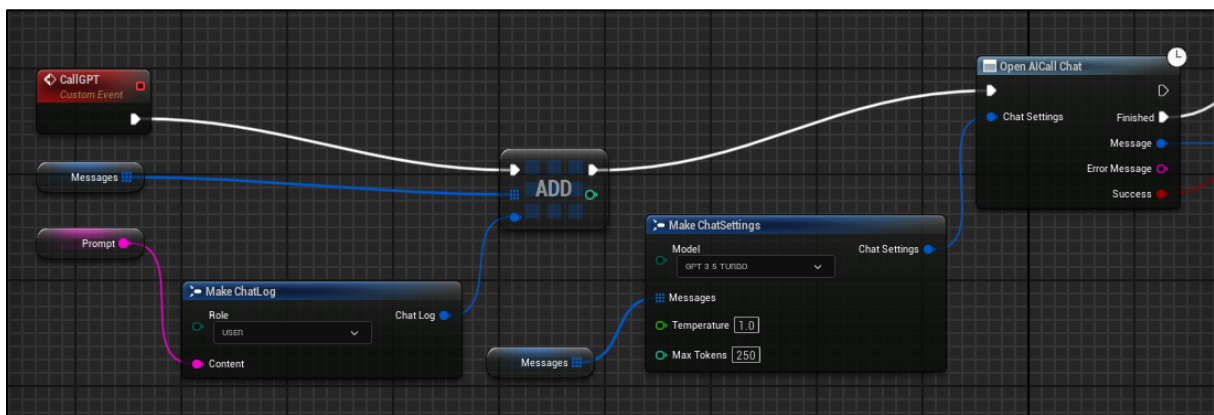
Slika 37 Open AI Api ključ

Nakon postavljanja Api ključa potrebno je povezati sve funkcionalnosti od pretvorbe govora u tekst, pozivanja ChatGPT-a do pretvorbe teksta u govor te na kraju animiranja lica virtualnog agenta. Na slici [Slika 38.] prikazan je poziv funkcije za pretvaranje govora u tekst, određivanje jezika koji je potrebno prepoznati unutar funkcije, postavljanje nove varijable „Prompt“, koja označava tekst generiran na temelju govora, ispisivanje prepoznatog govora u obliku teksta na ekranu pomoću funkcije „Print String“ te pozivanje „Call GPT“ funkcije koja je detaljnije opisana u nastavku. Pošto je ovaj projekt izveden u klasičnom sučelju spremnom za korištenje na računalu, a ne u virtualnoj stvarnosti, postavljena je funkcija za davanje glasovnih naredbi u trenutku kada se na tipkovnici pritisne tipka „1“.



Slika 38 Funkcija za pretvaranje govora u tekst i poziv funkcije „Call GPT“

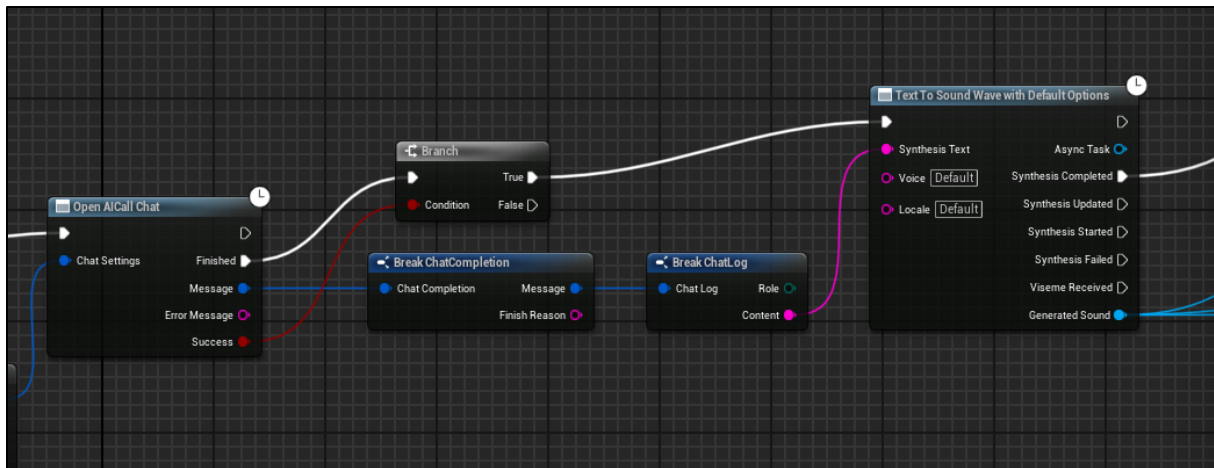
Nakon što se generira tekst prepoznat od strane zvuka korisnika, potrebno je pohraniti taj tekst u obliku upita za ChatGPT putem „Make ChatLog“ funkcije. Uz tekst primljen od strane korisnika, moguće je i dodati unaprijed generirana pravila i upute koje virtualni agent može slijediti putem varijable „Messages“. Varijable „Prompt“ i „Messages“ se pohranjuju te zajedno šalju prema ChatGPT-u koji zatim daje određeni odgovor na upit korisnika. Na slici [Slika 39.] Prikazano je definiranje „CallGPT“ funkcije te pohranjivanje poruka koje se šalju prema ChatGPT-u putem funkcija „Add“, „Make ChatLog“, „Make ChatSettings“ i „Open AiCall Chat“. Unutar „Make ChatSettings“ definirana je 3.5 verzija ChatGPT-a koja se koristi.



Slika 39 Slanje upita ChatGPT-u

Nakon što je poslan upit ChatGPT-u potrebno je „Branch“ funkcijom provjeriti je li završilo generiranje odgovora od strane ChatGPT-a te ukoliko je, proslijediti tu poruku u funkciju za pretvorbu teksta u govor. Na slici [Slika 40.] prikazane su funkcije procesuiranja i pohranjivanja odgovorenih poruka od strane ChatGPT-a „Break ChatCompletion“ i „Break ChatLog“ te funkcija pretvorbe generiranog teksta u zvuk pomoću „Text to Sound Wave“ funkcije.

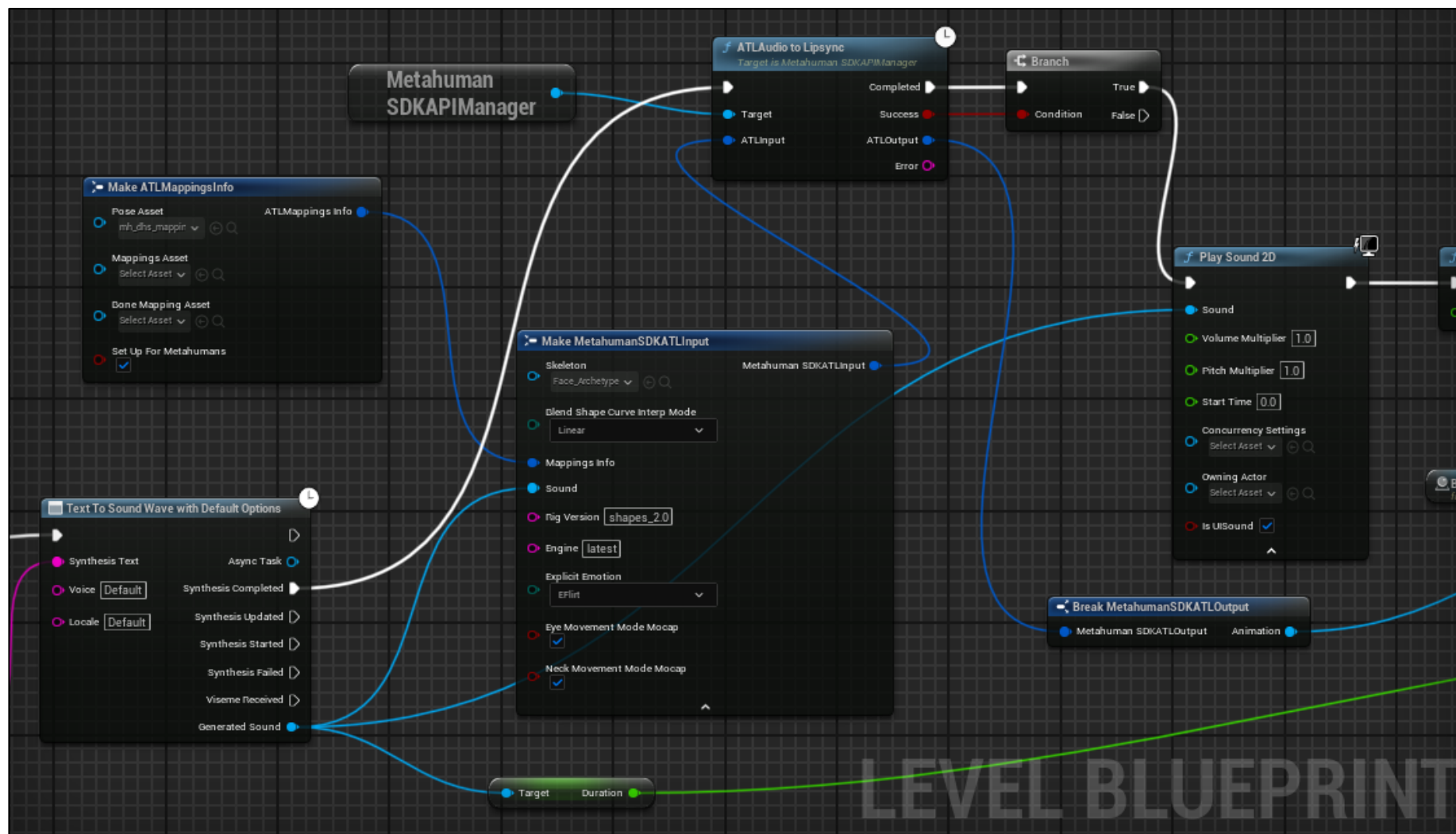




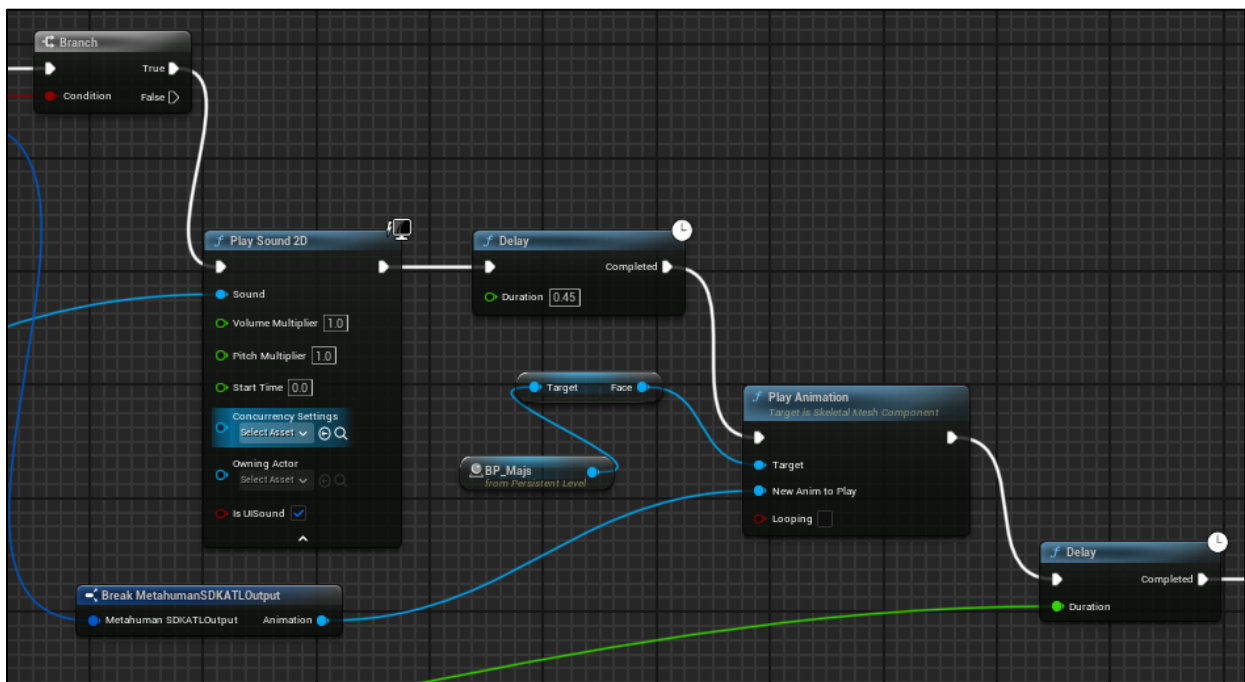
Slika 40 Pretvaranje generiranog teksta u govor

### 6.2.2 Animiranje lica virtualnog agenta

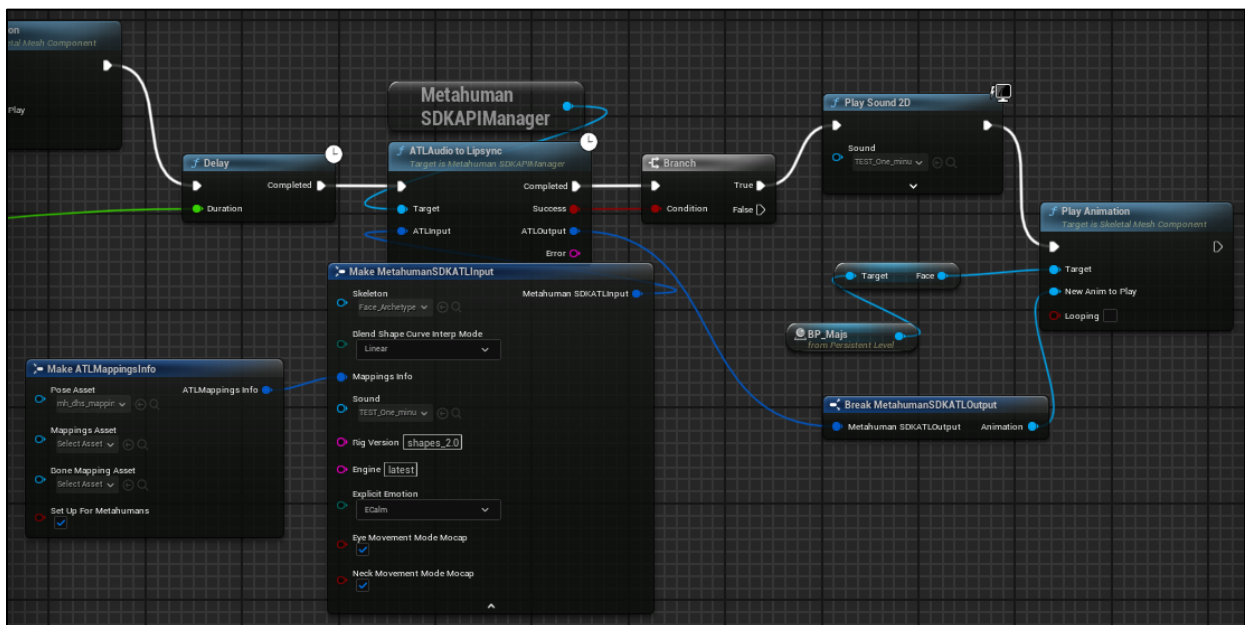
Nakon što je generiran zvuk koji odgovara tekstu generiranog od strane ChatGPT-a potrebno je pokrenuti taj zvuk kao odgovor korisniku. Taj dio odrađuje se funkcijom „Play Sound 2D“. Uz pokretanje generiranog zvuka, u ovoj inačici projekta, potrebno je postaviti sve funkcije od dodatka MeathumanSDK kojima se omogućuje animiranje lica virtualnog agenta sukladno pohranjenom zvuku. Nakon animiranja lica virtualnog agenta tokom izgovaranja dobivenog odgovora, bilo je potrebno animirati lice i u stanju dok ne govori, odnosno kada čeka upit od korisnika i miruje. Animacija mirovanja se postigla pomoću ubacivanja bezvučnog audio zapisa unutar „Play Sound 2D“ funkcije te se na taj način postiglo da virtualni agent djeluje „življe“ dok čeka idući upit od korisnika. Pomoću funkcije „Delay“ se odgađa animacija mirovanja dok virtualni agent ne izgovori cijeli odgovor. Na slikama [Slika 41.], [Slika 42.] i [Slika 43.] prikazane su funkcije za pretvaranje teksta u govor, pokretanje zvuka te funkcije potrebne za pokretanje animiranja lica virtualnog agenta koje su dostupne unutar MetahumanSDK dodatka.



Slika 41 MetahumanSDK funkcije u shematskom planu dio 1.



Slika 42 MetahumanSDK funkcije u shematskom planu dio 2.



Slika 43 MetahumanSDK funkcije u shematskom planu dio 3.

Nakon što je shematski plan pohranjen, zajedno sa svim potrebnim postavkama, pokretanjem simulacije moguće je komunicirati verbalno sa virtualnim agentom

koji prilikom izgovaranja odgovora koristi precizne zvukom slijeđenje animacije otvaranja i zatvaranja usta, na način kojim ljudi govore.

Na idućoj poveznici prikazan je test konverzacije između čovjeka i virtualnog agenta: <https://www.youtube.com/watch?v=6Ws9LJZzLM>.

U videu se može očitovati da postoji određeni vremenski razmak između postavljenog pitanja korisnika i odgovora virtualnog agenta. Ta problematika proizlazi iz toga što je potrebno određeno vrijeme da se odgovor generira unutar ChatGPT-a te određeno vrijeme da se kreira animacija pomoću vanjskog dodatka MetahumanSDK. Ovaj problem bi se mogao riješiti na način da se virtualnom agentu omogući direktno izgovaranje teksta, rečenicu po rečenicu, a da se ne čeka potpuni odgovor generiran od strane ChatGPT-a. Pored toga bilo bi poželjno da se omogući vanjskim dodacima ili aplikacijama animiranje lica virtualnog agenta bez potrebe za povezivanjem s online servisima, odnosno izvan mrežno (eng. *offline*).

## 7. Zaključak

U ovom radu istražena je problematika i principi izrade modela virtualnih agenata koji su sposobni voditi verbalnu komunikaciju s ljudima uz generiranje animacija lica vođenih zvukom. Kroz rad je obrađena teorijska osnova i navedene su primjene obrade prirodnog jezika te princip rada korištenih alata Unreal Engine i ChatGPT. Poznavanjem teorijskih osnova omogućeno je shvaćanje načina na koji se povezuju navedeni alati u svrhu izvedbe danog zadatka.

Zadatak je izveden na dva načina s obzirom na način generiranja animacija lica virtualnih agenata. Svaki od načina ima određene prednosti i nedostatke u odnosu jednoga prema drugome. Kod korištenja jednostavnih animacija imitacije govora otvaranjem i zatvaranjem usta, najveća prednost je brzina pokretanja animacije koja je direktno generirana unutar Unreal Engine-a te dostupnost korisnicima pošto je ovakav način u potpunosti besplatan. Dok, s druge strane, korištenjem preciznih animacija govora dobivamo puno realnijeg virtualnog agenta koji nalikuje po ponašanju pravom čovjeku. Glavni nedostatak kod korištenja preciznih animacija je u tome što se mora koristiti neka vanjska aplikacija ili dodatak koji trenutno dolaze uz određene naplate ili poteškoće prilikom povezivanja s Unreal Engine-om, poput brzine generiranja animacija nakon dobivenog odgovora od strane ChatGPT-a.

Ovaj rad doprinosi širem razumijevanju mogućnosti primjene umjetne inteligencije, posebno u kontekstu razvoja virtualnih okruženja i ljudi. Nadalje, pruža se temelj za daljnja istraživanja i razvoj u području interakcije i razvoja inteligentnih virtualnih agenata. U budućnosti, moguće je proširiti funkcionalnosti i poboljšati performanse ovakvog sustava na način da se u potpunosti smanji vrijeme čekanja dobivenih odgovora od strane virtualnog agenta uz potpuno realistične animacije govora.

## PODRŠKA HRZZ

Ovaj rad je sufinancirala Hrvatska zaklada za znanost projektom "AMICORC - Afektivna multimodalna interakcija temeljena na konstruiranoj robotskoj spoznaji" (UIP-2020-02-7184).

## 8. Literatura

- [1] *Umjetna inteligencija*, <https://enciklopedija.hr/natuknica.aspx?ID=63150> (dostupno dana 17.03.2024.)
- [2] *Obrada prirodnog jezika*, <https://www.techtarget.com/searchenterpriseai/definition/natural-language-processing-nlp> (dostupno dana 19.03.2024.)
- [3] *Komunikacija čovjeka i računala*, <https://revolveai.com/natural-language-processing-techniques/> (dostupno dana 19.03.2024.)
- [4] *Glavne metode NLP-a*, [https://www.splunk.com/en\\_us/blog/learn/natural-language-processing-nlp.html](https://www.splunk.com/en_us/blog/learn/natural-language-processing-nlp.html) (dostupno dana 19.03.2024.)
- [5] *Simbolički NLP*, [https://en.wikipedia.org/wiki/Natural\\_language\\_processing#Neural\\_NLP\\_\(present\)](https://en.wikipedia.org/wiki/Natural_language_processing#Neural_NLP_(present)) (dostupno dana 19.03.2024.)
- [6] *SHRDLU*, <https://www.freecodecamp.org/news/how-natural-language-processing-powers-chatbots-4-common-approaches-a077a4de04d4/> (dostupno dana 19.03.2024.)
- [7] *Strojno učenje*, <https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML> (dostupno dana 19.03.2024.)
- [8] *Statistički NLP*, [https://en.wikipedia.org/wiki/Statistical\\_model](https://en.wikipedia.org/wiki/Statistical_model) (dostupno dana 19.03.2024.)
- [9] *Prepoznavanje govora*, [https://en.wikipedia.org/wiki/Speech\\_recognition](https://en.wikipedia.org/wiki/Speech_recognition) (dostupno dana 19.03.2024.)
- [10] *Primjena Statističkog NLP-a*, <https://proxet.com/blog/fundamentals-of-statistical-natural-language-processing/> (dostupno dana 19.03.2024.)
- [11] *Neuronske mreže*, <https://blog.pangeanic.com/neural-networks-and-how-they-work-in-natural-language-processing> (dostupno dana 19.03.2024.)

- [12] *Model neuronske mreže*, <https://www.investopedia.com/terms/n/neuralnetwork.asp> (dostupno dana 19.03.2024.)
- [13] *Neuronska mreža s jednosmjernim prijenosom*, <https://brilliant.org/wiki/feedforward-neural-networks/> (dostupno dana 19.03.2024.)
- [14] *Neuronska mreža s povratnom propagacijom*, <https://www.guru99.com/backpropogation-neural-network.html> (dostupno dana 19.03.2024.)
- [15] *Konvolucijska neuronska mreža*, <https://medium.com/swlh/an-overview-on-convolutional-neural-networks-ea48e76fb186> (dostupno dana 19.03.2024.)
- [16] *Primjena NLP-a*, <https://www.geeksforgeeks.org/top-7-applications-of-natural-language-processing/> (dostupno dana 20.03.2024.)
- [17] *Automatsko dovršavanje riječi u rečenici*, <https://blog.google/products/search/how-google-autocomplete-works-search/> (dostupno dana 20.03.2024.)
- [18] *Analiza sentimenta*, <https://imerit.net/sentiment-analysis/> (dostupno dana 20.03.2024.)
- [19] *Pretvaranje teksta u govor i obratno*, <https://elevenlabs.io/blog/text-to-speech-vs-speech-to-text/> (dostupno dana 20.03.2024.)
- [20] *Chatbotovi*, <https://www.ibm.com/topics/chatbots> (dostupno dana 21.03.2024.)
- [21] *Primjeri chatbotova*, <https://www.pcmag.com/picks/the-best-ai-chatbots> (dostupno dana 22.03.2024.)
- [22] *ChatGPT*, <https://www.zdnet.com/article/what-is-chatgpt-and-why-does-it-matter-heres-everything-you-need-to-know/> (dostupno dana 22.03.2024.)
- [23] *Princip rada ChatGPT-a*, <https://www.atriainnovation.com/en/how-does-chat-gpt-work/> (dostupno dana 22.03.2024.)
- [24] *Unreal Engine*, <https://www.unrealengine.com/en-US/faq> (dostupno dana 22.03.2024.)
- [25] *PLEA*, <https://zimo.dnevnik.hr/clanak/zakoracite-u-metasvemir-na-fsb-u-i-komunicirajte-s-virtualnim-bicem-plea---691838.html> (dostupno dana 22.03.2024.)
- [26] *Prototip afektivne humanoidne glave*, <https://robotika.fsb.hr/> (dostupno dana 22.03.2024.)

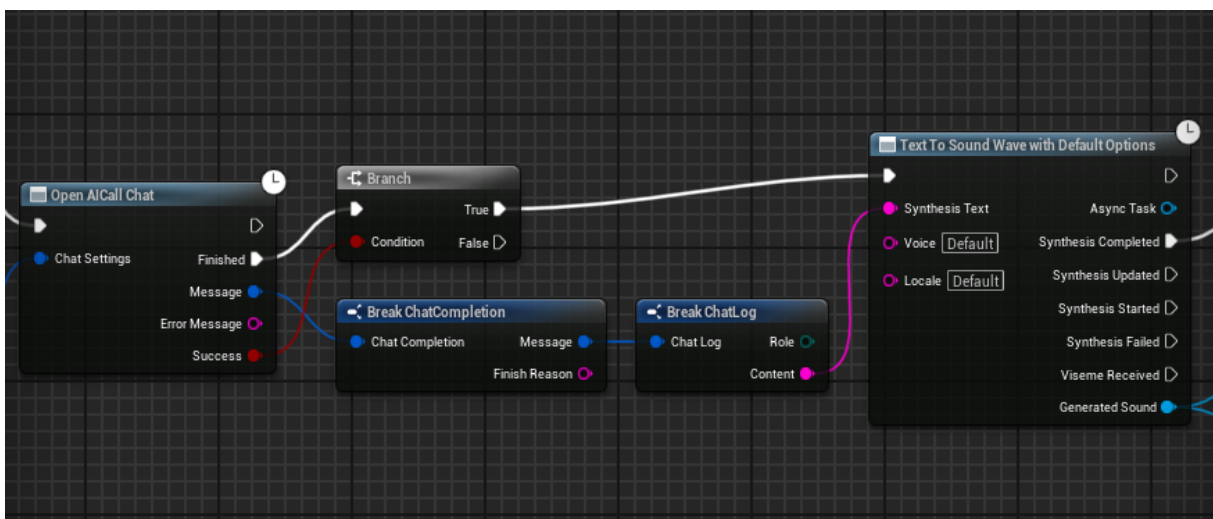
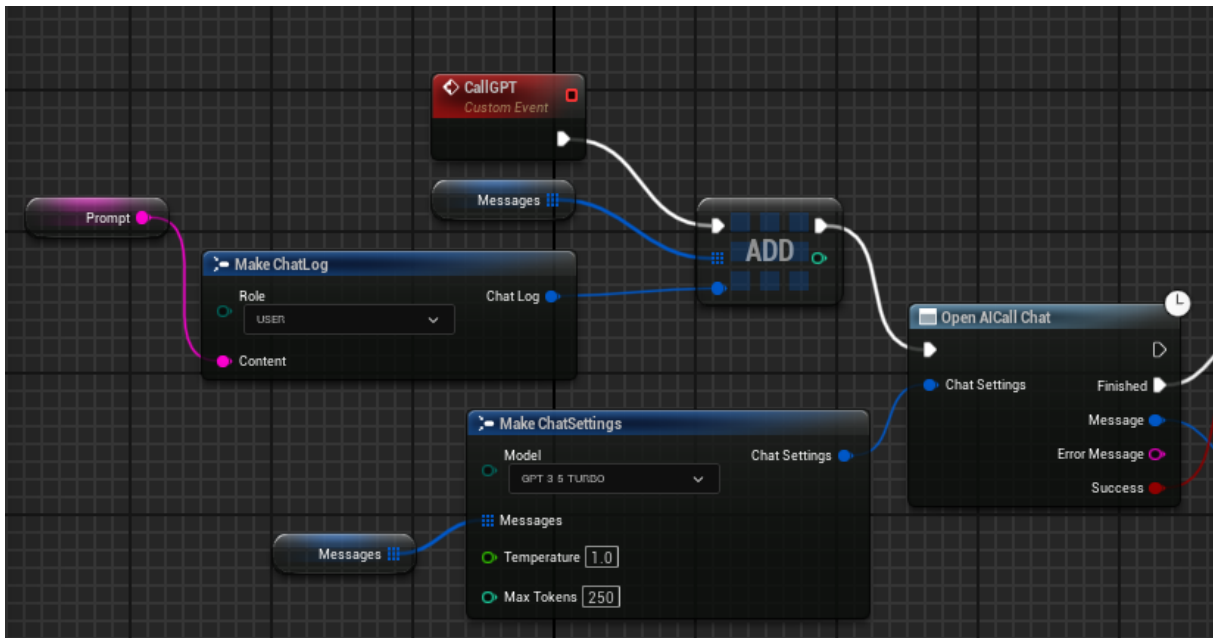
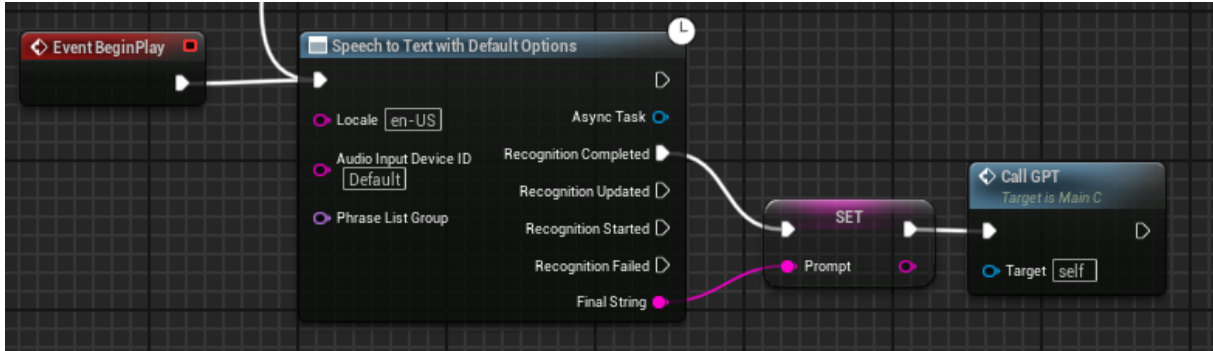
- [27] *Virtualna stvarnost*, <https://www.enciklopedija.hr/clanak/virtualna-stvarnost> (dostupno dana 22.03.2024.)
- [28] *Korištenje VR-a*, <https://medium.com/imeshup/all-in-the-immersive-games-of-vr-1937c950f955> (dostupno dana 22.03.2024.)
- [29] *Animiranje Metahumana*, <https://dev.epicgames.com/documentation/en-us/metahuman/animating-metahumans/sequencer> (dostupno dana 26.03.2024.)
- [30] *Audio2Face aplikacija*, <https://www.nvidia.com/en-us/ai-data-science/audio2face/> (dostupno dana 26.03.2024.)
- [31] *MetahumanSDK dodatak*, <https://docs.metahumansdk.io/metahuman-sdk> (dostupno dana 27.03.2024.)
- [32] *API*, <https://www.mulesoft.com/resources/api/what-is-an-api> (dostupno dana 27.03.2024.)
- [33] *Način rada API-ja*, <https://www.altexsoft.com/blog/engineering/what-is-api-definition-types-specifications-documentation/> (dostupno dana 27.03.2024.)
- [34] *RESTful API*, <https://www.techtarget.com/searchapparchitecture/definition/RESTful-API> (dostupno dana 28.03.2024.)
- [35] *Način rada REST API-ja*, <https://www.altexsoft.com/blog/rest-api-design/> (dostupno dana 28.03.2024.)
- [36] *WebSocket API*, [https://developer.mozilla.org/en-US/docs/Web/API/WebSockets\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API) (dostupno dana 29.03.2024.)
- [37] *Način rada WebSocket API-ja*, <https://en.wikipedia.org/wiki/WebSocket> (dostupno dana 29.03.2024.)
- [38] *Unreal Engine dodaci*, <https://docs.unrealengine.com/4.27/enUS/ProductionPipelines/Plugins/> (dostupno dana 29.03.2024.)
- [39] *HttpGPT dodatak*, <https://www.unrealengine.com/marketplace/en-US/product/httpgpt-chatgpt-integrated-in-the-engine?sessionInvalidated=true> (dostupno dana 30.03.2024.)
- [40] *OpenAI API dodatak*, <https://github.com/KellanM/OpenAI-Api-Unreal> (dostupno dana 30.03.2024.)

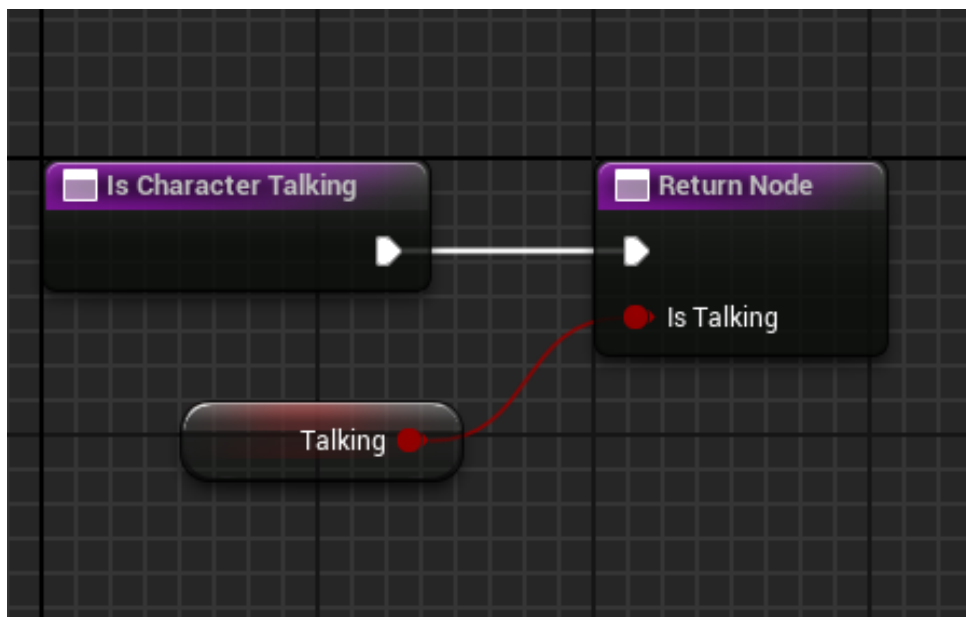
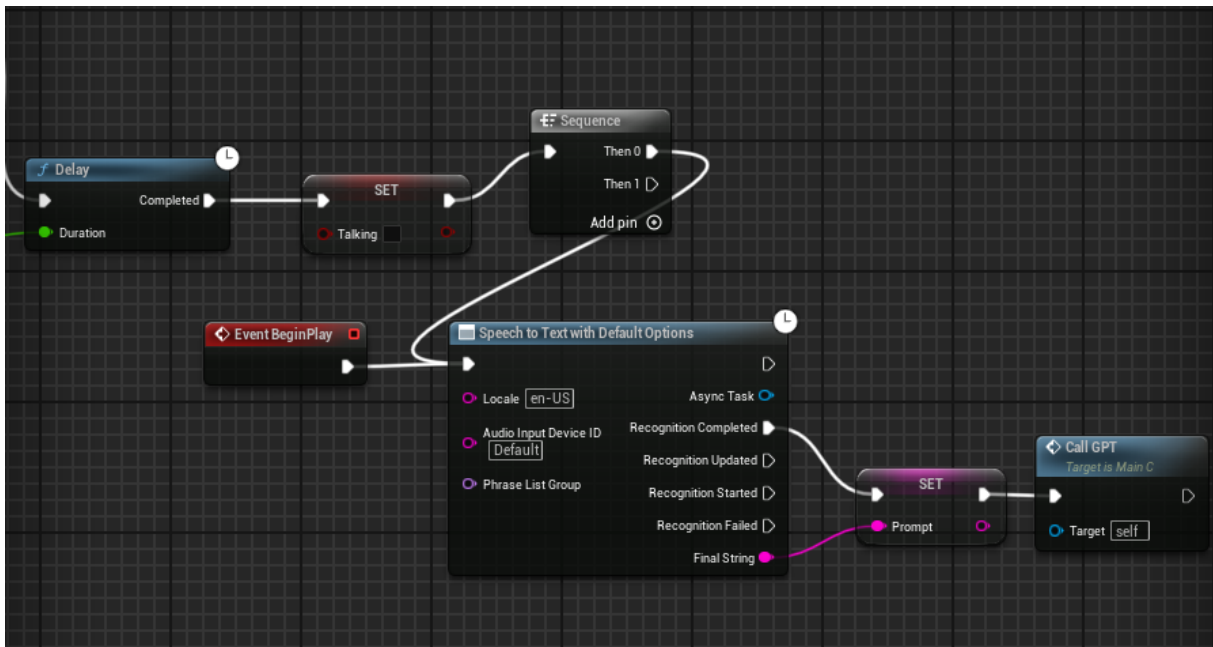
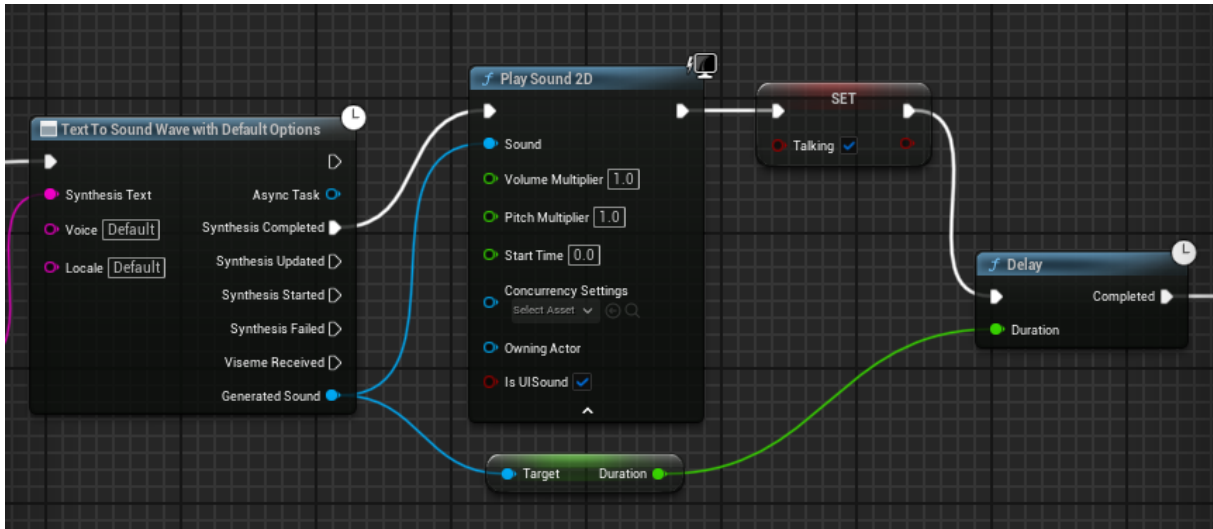


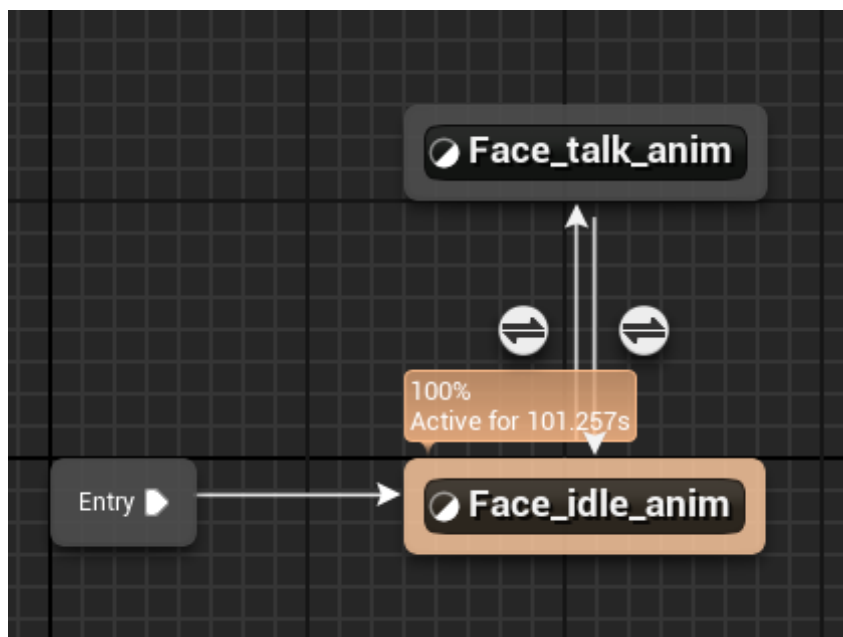
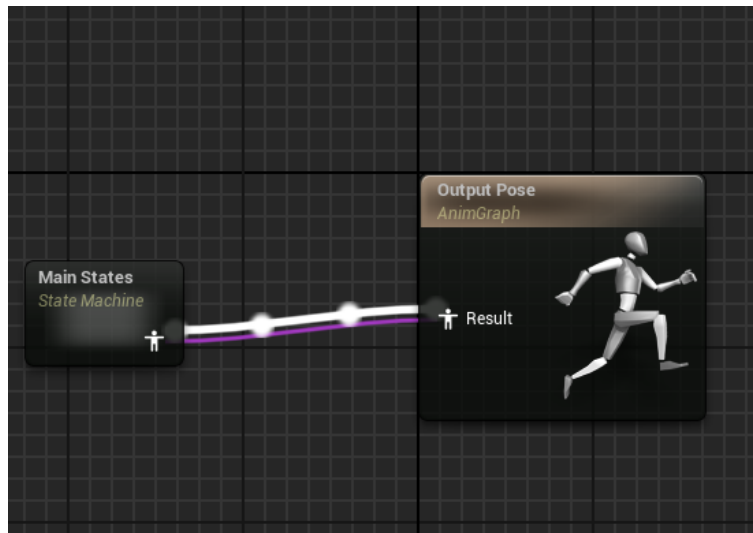
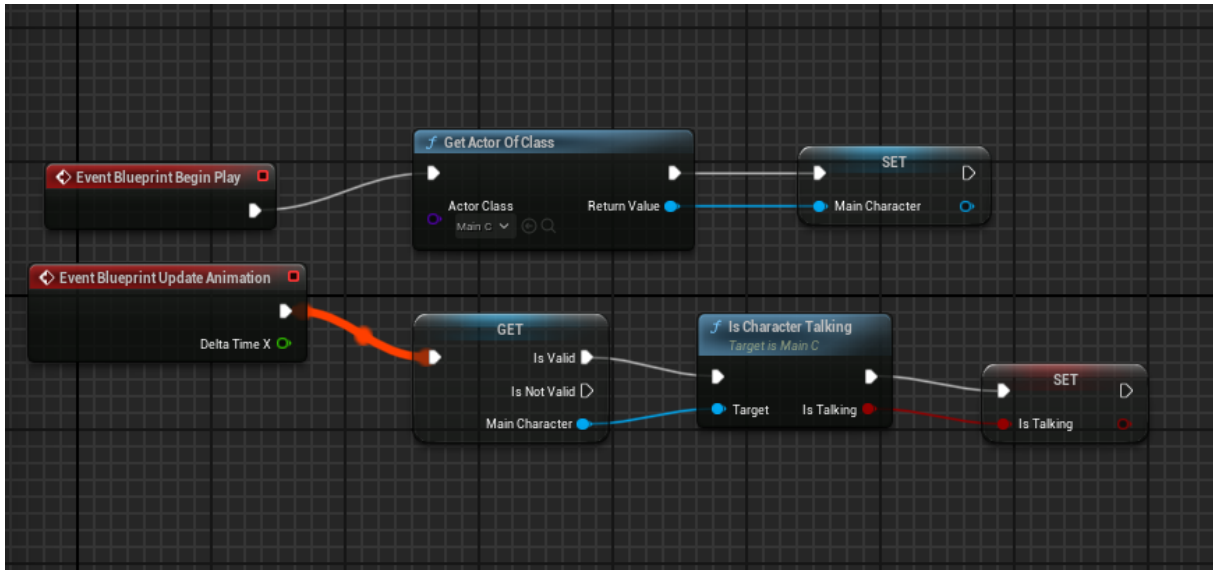
## **9. Prilozi**

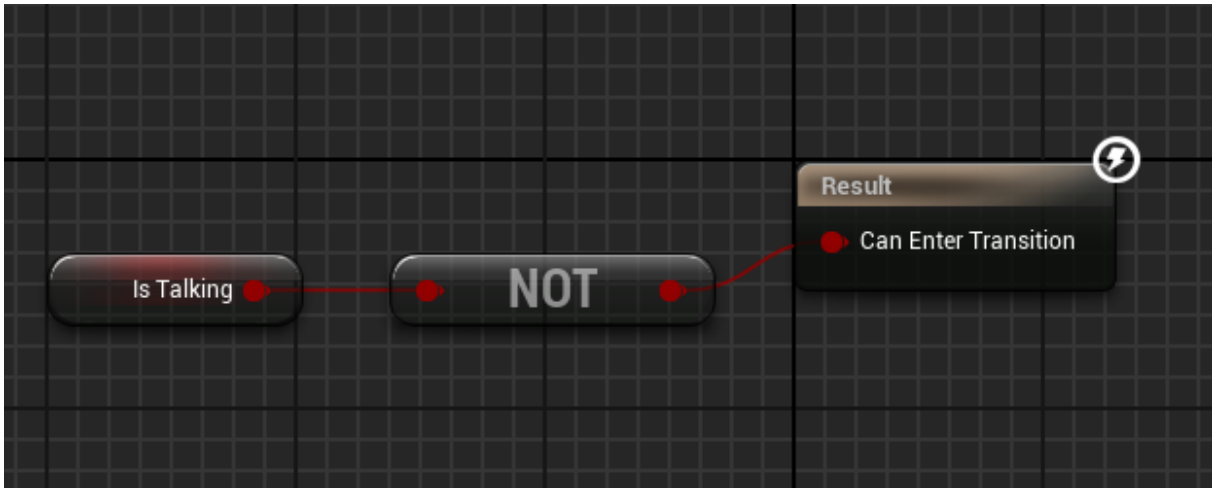
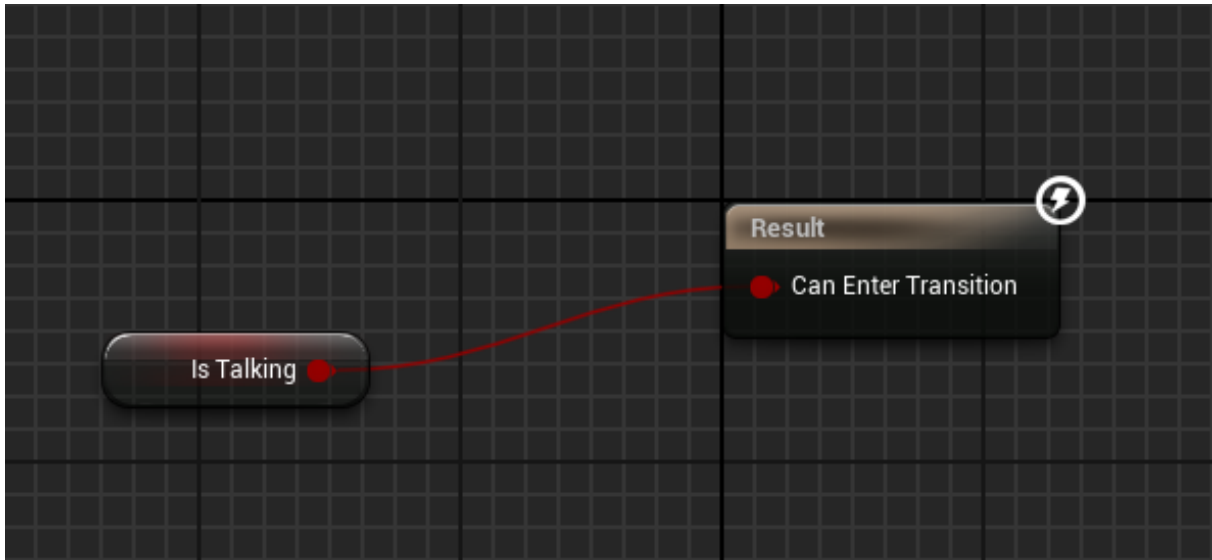
- I. Shematski planovi (eng. *Blueprints*) – Virtualni agent s jednostavnim animacijama govora
- II. Shematski planovi (eng. *Blueprints*) – Virtualni agent s preciznim animacijama govora

# SHEMATSKI PLANOVI (ENG. BLUEPRINTS) – VIRTUALNI AGENT S JEDNOSTAVNIM ANIMACIJAMA GOVORA









# SHEMATSKI PLANOVI (ENG. *BLUEPRINTS*) – VIRTUALNI AGENT S PRECIZNIM ANIMACIJAMA GOVORA

