

Holistički pristup interakciji s NAO robotom: od detekcije zvuka do govorne povratne informacije koristeći OpenAI tehnologiju

Čolić, Fran

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, School of Applied Mathematics and Informatics / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet primijenjene matematike i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:126:052177>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-02-22**



mathos

Repository / Repozitorij:

[Repository of School of Applied Mathematics and Informatics](#)





SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET PRIMIJENJENE MATEMATIKE I INFORMATIKE

Sveučilišni prijediplomski studij Matematika i računarstvo

Holistički pristup interakciji s NAO robotom: Od detekcije zvuka do govorne povratne informacije koristeći OpenAI tehnologiju

ZAVRŠNI RAD

Mentor:

doc. dr. sc. Domagoj Ševerdija

Student:

Fran Čolić

Osijek, 2024.

SAŽETAK

U ovoj dokumentaciji završnog rada opisat ćemo način na koji upravljamo humanoidnim NAO robotom te kako pohranjujemo i obrađujemo zvučne zapise snimljene tijekom interakcije. Prikazat ćemo razvoj ideje, tijekom implementacije te proces testiranja. Detaljno ćemo objasniti metodu koju koristimo kako bismo prepoznali zvučne zapise te pretvorili ih u odgovarajuću povratnu informaciju od strane robota. Ovisno o tipu prepoznatog signala, robot će odgovoriti zvučnom povratnom informacijom koja se reproducira putem ugrađenih zvučnika. Spajanje i upravljanje robotom odvijat će se u programskom softveru Choregraphe [1], dok će obrada zvučnih signala biti provedena unutar Python [2] programskog jezika. Tijekom implementacije, spojeni smo na stvarnog robota te se cijeli proces odvija u interakciji s robotom. Na kraju dokumentacije ćemo prikazati primjere i zaključke do kojih smo došli.

Ključne riječi: humanoidni robot, NAO, prepoznavanje zvuka, povratna informacija, openAI

A HOLISTIC APPROACH TO INTERACTION WITH THE NAO ROBOT: FROM SOUND DETECTION TO SPOKEN FEEDBACK USING OPENAI TECHNOLOGY

In this documentation of the final project, we will describe how we control the humanoid NAO robot and how we store and process audio recordings captured during interaction. We will show the development of the idea, the implementation process, and the testing process. We will explain in detail the method we use to recognize audio recordings and convert them into appropriate feedback from the robot. Depending on the type of recognized signal, the robot will respond with audio feedback, which is played through its built-in speakers. Connecting and controlling the robot will be done through the Choregraphe software [1], while audio signal processing will be carried out using the Python [2] programming language. During implementation, we are connected to the actual robot, and the entire process takes place in interaction with the robot. At the end of the documentation, we will present examples and conclusions we have reached.

Keywords: humanoid robot, NAO, sound recognition, feedback, openAI

Sadržaj

1	Uvod	1
2	Pokretanje i spajanje robota	2
3	Korištenje softvera i alata	4
3.1	Python	4
3.1.1	Instalacija	4
3.2	Choregraphe	6
3.2.1	Instalacija	6
3.2.2	Korištenje	6
4	Softveri treće strane	11
4.1	OpenAI	11
4.2	Flask	12
5	Implementacija	13
5.1	Skripta <i>Detekcija i pohrana zvuka</i>	14
5.2	Skripta <i>Reprodukcija zvuka</i>	15
5.3	Skripta <i>Server</i>	16
5.4	Flask Server i Obrada Zvuka	17
5.4.1	Ključne Funkcionalnosti Skripte	17
6	Rezultati	18
6.1	Analiza logova	18
	Literatura	22

1 | Uvod

Ideja ovog završnog projekta bila je obuhvatiti holistički pristup interakciji s humanoidnim NAO robotom, koristeći detekciju zvuka i govornu povratnu informaciju služeći se OpenAI tehnologijom kako bi se omogućila prirodna komunikacija između korisnika i robota. Cilj je bio primijeniti tehnologije prepoznavanja i obrade zvuka u stvarnom vremenu, u kombinaciji s metodama upravljanja robotom putem softvera Choregraphe i programskog jezika Python.

Kroz Choregraphe platformu, razvijene su Python skripte koje omogućuju da NAO robot detektira zvukove iz okoline. Kad robot prepozna zvuk, korisnik mu može postaviti pitanje, a robot bilježi zvučni zapis pitanja nakon detektirane pauze. Ovaj zvučni zapis potom se šalje serveru, gdje se koristi OpenAI tehnologija za transkripciju zvuka u tekst. Nakon toga, OpenAI generira tekstualni odgovor na postavljeno pitanje, koji se zatim prevodi u audio zapis na hrvatskom jeziku i vraća robotu.

Robot reproducira primljeni audio zapis putem svojih ugrađenih zvučnika, čime korisniku daje govorni odgovor na postavljeno pitanje. Ovaj cjeloviti proces omogućuje dvosmjernu komunikaciju s robotom te pridonosi prirodnijem i intuitivnijem iskustvu interakcije s humanoidnim robotom. U ovoj dokumentaciji detaljno će se prikazati koraci implementacije i korištenje tehnologija koje omogućuju ovakvu interakciju, kao i evaluacija postignutih rezultata i moguće smjernice za daljnji razvoj.

2 | Pokretanje i spajanje robota

Humanoidni NAO robot je iznimno lak za spajanje i rukovanje. Detaljnu dokumentaciju za rukovanje robotom možemo pronaći na [5]. Nakon pažljivog vađenja iz zaštitnog kofera, postavljamo ga u čučajući položaj i spajamo ga na punjač.

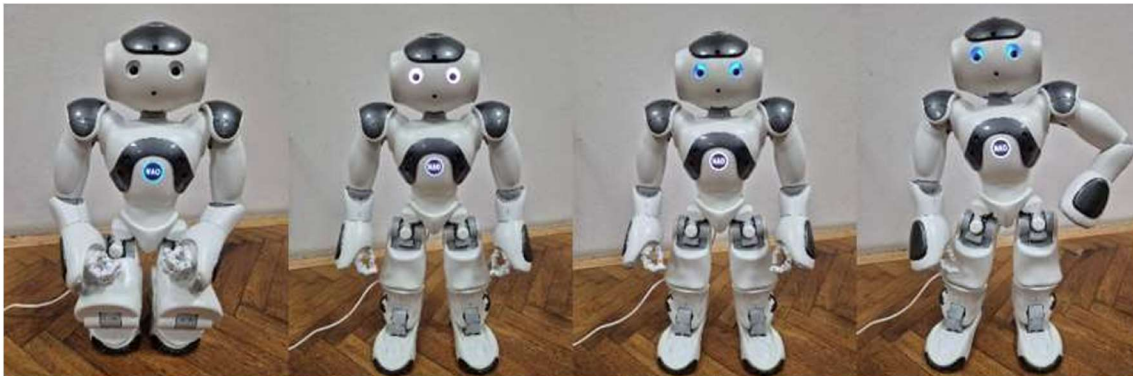


Slika 2.1: NAO robot u položaju čučnja



Slika 2.2: Spajanje robota na punjač

Nakon što se robot ponešto napuni, palimo ga tako što tri sekunde držimo NAO tipku. Robot se budi i ustaje.



Slika 2.3: Proces buđenja

Kako bismo robota spojili na računalo, spajamo mrežni kabel u glavu robota te u računalo.



Slika 2.4: Spajanje robota na računalo

3 | Korištenje softvera i alata

U ovom poglavlju objasnit ćemo kako smo koristili programski jezik Python za upravljanje NAO robotom, uključujući instalaciju potrebnih biblioteka koje su kompatibilne s robotom.

Nakon toga, objasnit ćemo način rada s Choregraphe programskim softverom, koji smo koristili za kontrolu pokreta, interakciju i testiranje robota. Navest ćemo specifične opcije koje smo koristili te izvore s kojih smo preuzeli softver.

3.1 Python

Upravljanje i programiranje NAO robota odvija se pomoću programskog jezika Python, koji pruža fleksibilnost i mogućnosti integracije s raznim API-jima. Važno je napomenuti da je NAO robot dostupan na tržištu od 2011. godine, a nakon toga prestalo je bilo kakvo službeno ažuriranje i razvoj softvera za ovaj model. Zbog toga je NAO robot kompatibilan samo s određenim starijim verzijama Python-a i specifičnim verzijama biblioteka koje odgovaraju vremenu njegovog razvoja.

3.1.1 Instalacija

Odlaskom na internetsku stranicu [9] možemo preuzeti verziju 2.7.18 Python-a, koja je jedina za koju je, prema našim testiranjima, potvrđeno da funkcionalno radi na NAO robotu i ne uzrokuje sukob verzija različitih biblioteka i paketa koje smo koristili.

Nakon klasične instalacije Python-a, potrebno je instalirati SDK – Software Development Kit [7] koji pruža kompanija SoftBank Robotics. To je skup alata koji omogućava uspostavljanje veze s NAO robotom te izgradnju funkcija za upravljanje i programiranje modula robota.

Nadalje, opisat ćemo kako povezati SDK unutar Python okruženja.

Ukoliko je Python27 instaliran na C disku, pristupamo datoteci na putanji `C:\Python27\Lib\site-packages` te ondje pohranjujemo SDK datoteku koju preimenujemo u `pynaoqi`.

Zatim je potrebno u PATH varijablu okruženja dodati nekoliko putanja:

- C:\Python27\Lib
- C:\Python27\Lib\Scripts

Nakon toga trebamo dodati novu varijablu okruženja:

- **Ime varijable:** PYTHONPATH
- **Vrijednost varijable:** C:\Python27\Lib\site-packages\pynaoqi\lib

Sada bi import naoqi trebao raditi unutar Pythona.

Potrebne Python biblioteke su:

- pip install openai
- pip install pydub
- pip install SpeechRecognition
- pip install playsound
- pip install requests
- pip install Flask

3.2 Choregraphe

Choregraphe je programski softver u vlasništvu tvrtke SoftBank Robotics [?]. Koristi se za kreiranje animacija te za kontrolu i interakciju s NAO robotom. Softver omogućuje korisnicima upravljanje robotom putem grafičkog sučelja koje sadrži blokovske funkcije za animaciju i kontrolu pokreta, kao i mogućnost pisanja Python skripti za napredne funkcionalnosti.

Osim za korisnike koji posjeduju robota, Choregraphe je također koristan i za one koji žele raditi sa simulacijom robota u vizualnom okruženju, pružajući realističan prikaz pokreta i interakcije robota. Sučelje nudi razne informacije o stanju robota, uključujući pozicije svih udova, napunjenost baterije, glasnoću govora i slične parametre.

Većinu navedenih funkcija opisat ćemo u poglavlju ??, gdje ćemo detaljnije objasniti osnovne i napredne opcije Choregraphe-a.

3.2.1 Instalacija

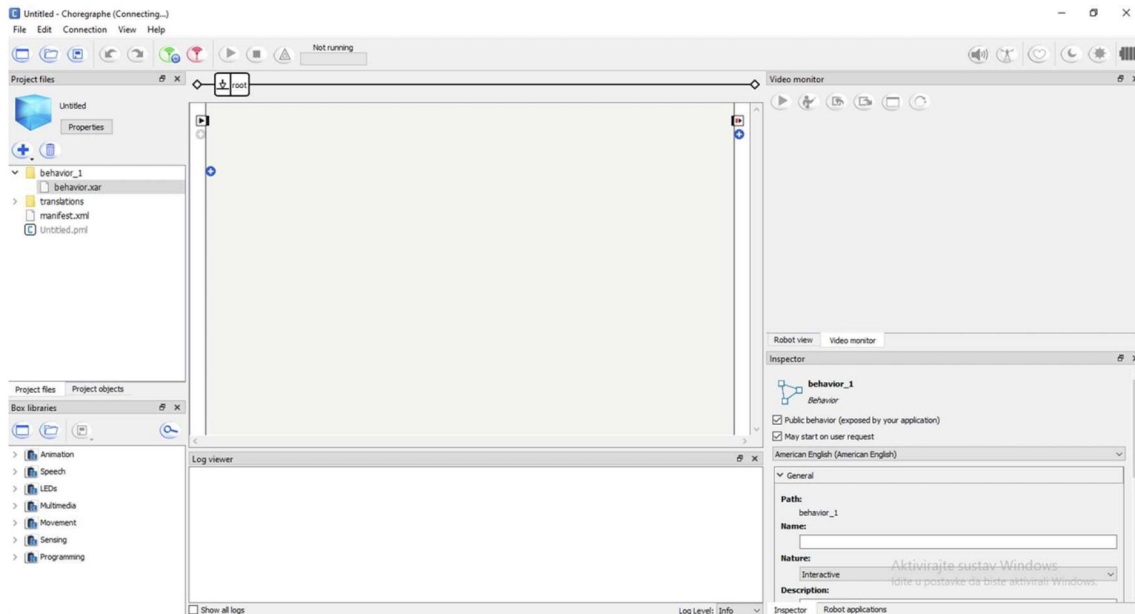
Preuzimanje softvera Choregraphe vršimo putem poveznice [3]. Odabiremo operativni sustav koji posjedujemo na računalu te preuzimamo odgovarajuću verziju instalacijskog paketa. Instalaciju provodimo na uobičajen način prateći korake instalacijskog čarobnjaka. Nakon dovršetka instalacije, možemo pokrenuti instalirani softver i povezati se s virtualnim NAO robotom ili pravim NAO robotom putem mrežne veze.

3.2.2 Korištenje

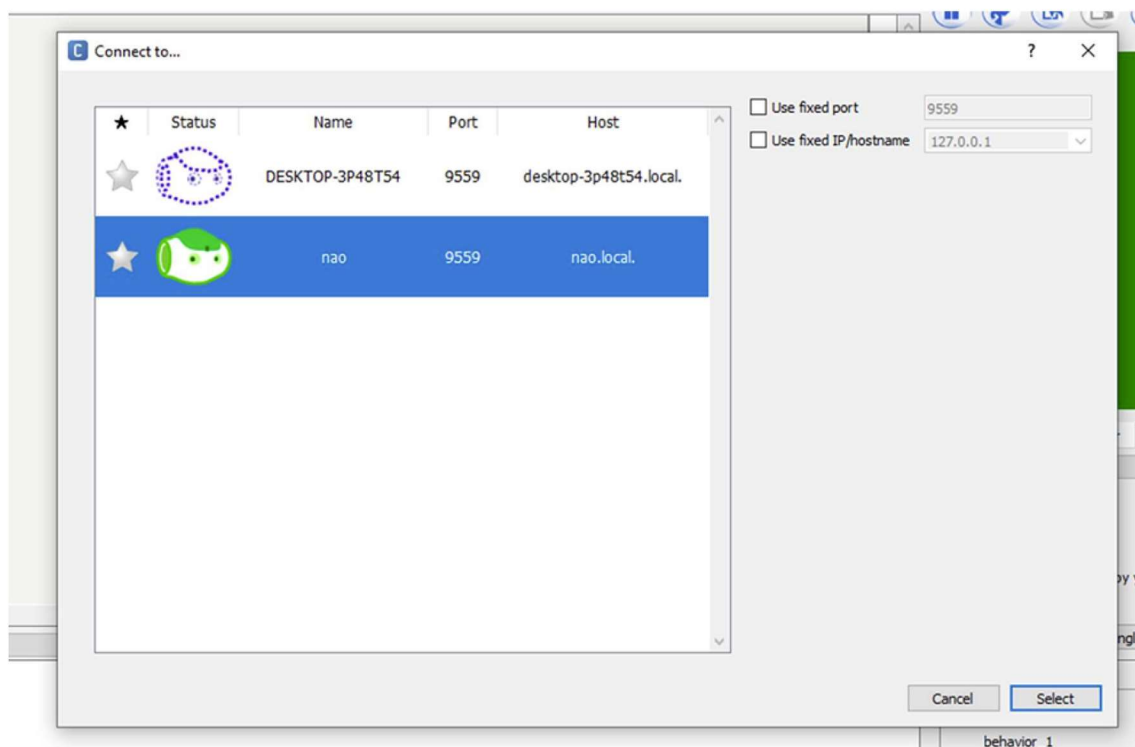
Prije samog opisa načina na koji se koristi Choregraphe programski softver, želimo naglasiti da ostale funkcionalnosti koje nisu korištene unutar ovog projekta nećemo dodatno opisivati jer se detaljna dokumentacija za Choregraphe softver može pronaći na internetskoj stranici [8]. Tu smo dokumentaciju koristili kako bismo se upoznali s funkcionalnostima koje su nam bile potrebne, a njihovo korištenje nadalje opisujemo.

Unutar Choregraphe softvera, pritiskom na ikonu za bežično povezivanje (vidi sliku 3.1) prikazuje se izbornik u kojem možemo odabrati plavog, "DESKTOP" robota za simulaciju ili zelenog "NAO" robota za fizički uređaj. Plavi robot označava simulaciju robota, dok zeleni predstavlja fizičkog robota.

U poglavlju 2 Pokretanje i spajanje robota, detaljno je opisan postupak spajanja fizičkog NAO robota s računalom. Ako su svi koraci ispravno izvedeni, zeleni robot bi trebao biti vidljiv i spreman za interakciju.



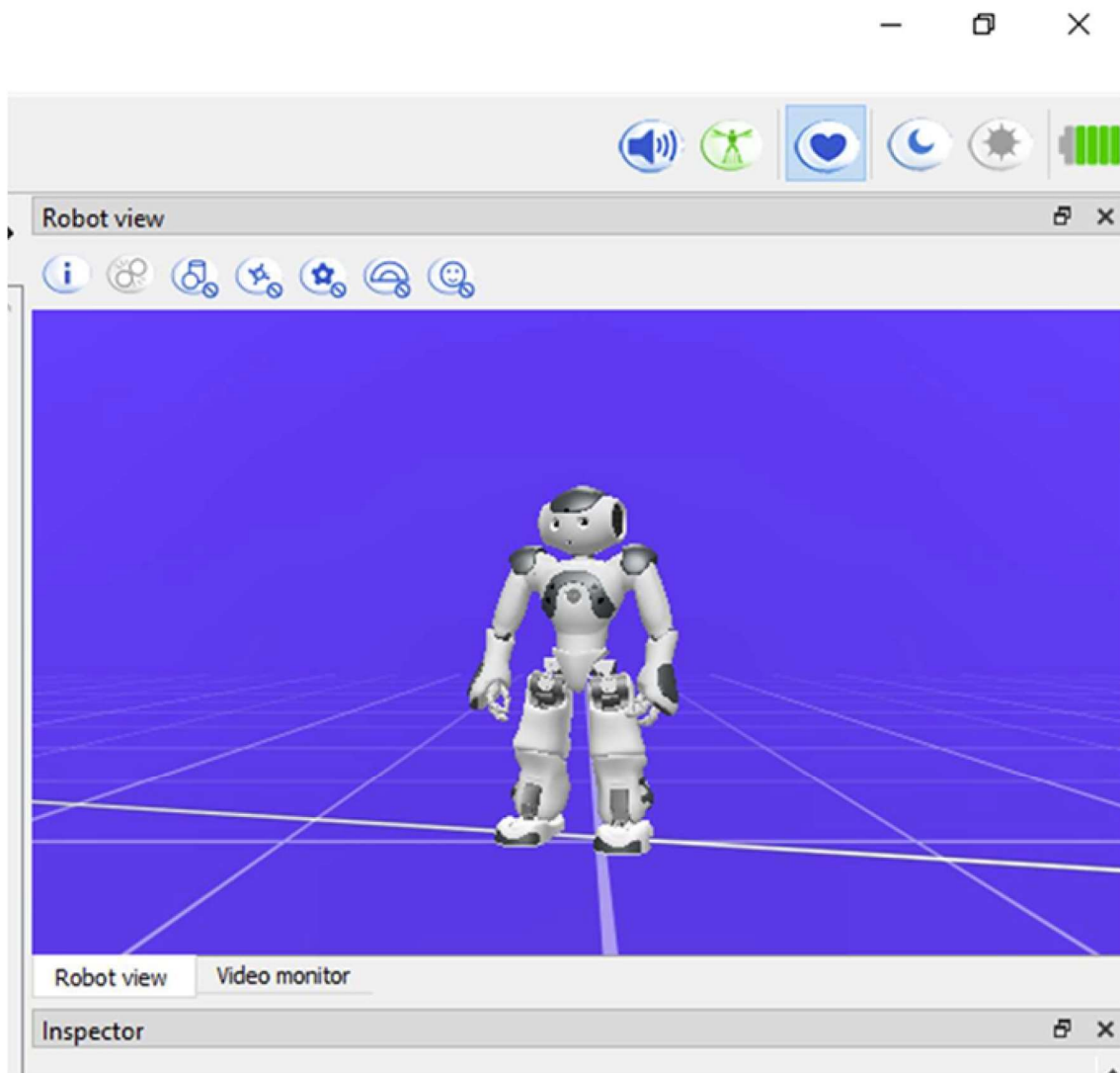
Slika 3.1: Choregraphe programski softver



Slika 3.2: Ponudene opcije spajanja

Ukoliko je sve ispravno povezano, u desnom bismo dijelu softvera trebali moći vidjeti robota koji ima jednake pozicije udova kao fizički robot (vidi sliku 3.3).

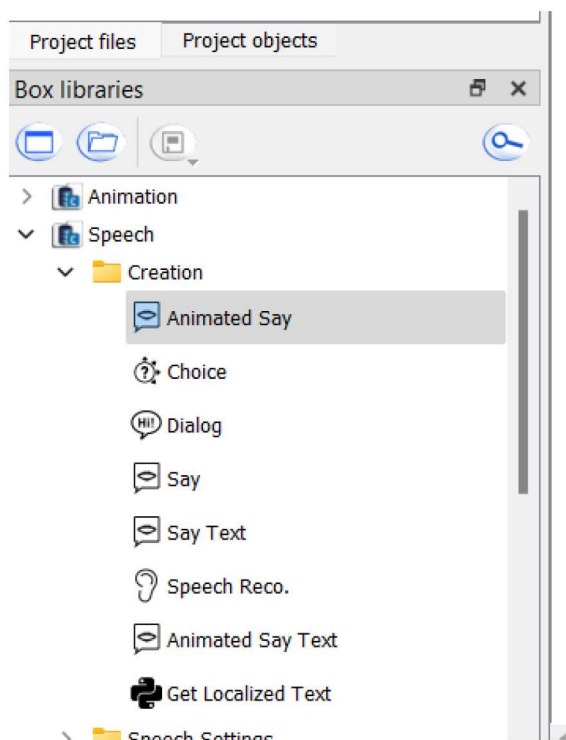
Kao što se može vidjeti, prikazane su razne informacije o robotu, uključujući razinu baterije, položaj udova, glasnoću govora i slične parametre. Također funkcionalnosti koje smo koristiti, te za koje smatramo da su korisne, jesu blokovske funkcije. Softver Choregraphe ima već veliki broj ugrađenih funkcija koje se



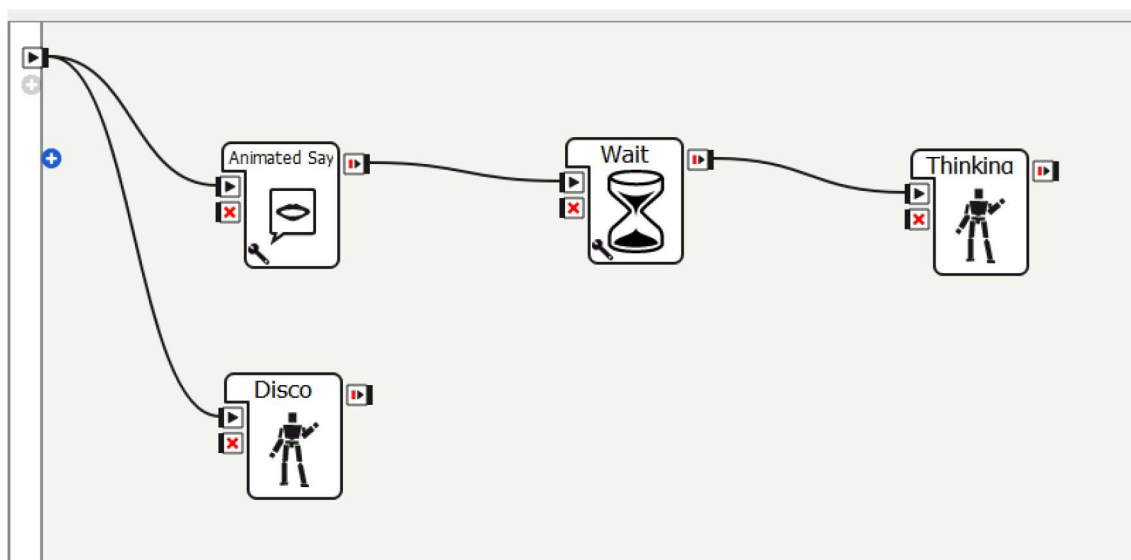
Slika 3.3: Simulacija robota u prostoru

moгу slagati i pozivati u obliku blokova. Tako, primjerice, možemo pokretati različite dijelove tijela, upravljati animacijama, sensorima ili dati robotu da izgovori rečenicu (vidi sliku 3.4).

Na slici 3.5 je prikazano kako iz izbornika u donjem lijevom dijelu softvera (slika 3.4) možemo odabrati blokovsku funkciju i povući ju pokazivačem u središnji dio ekrana. Takve funkcije pokrećemo klikom na dugme „play“ ili pritiskom na tipku F5 na tipkovnici.

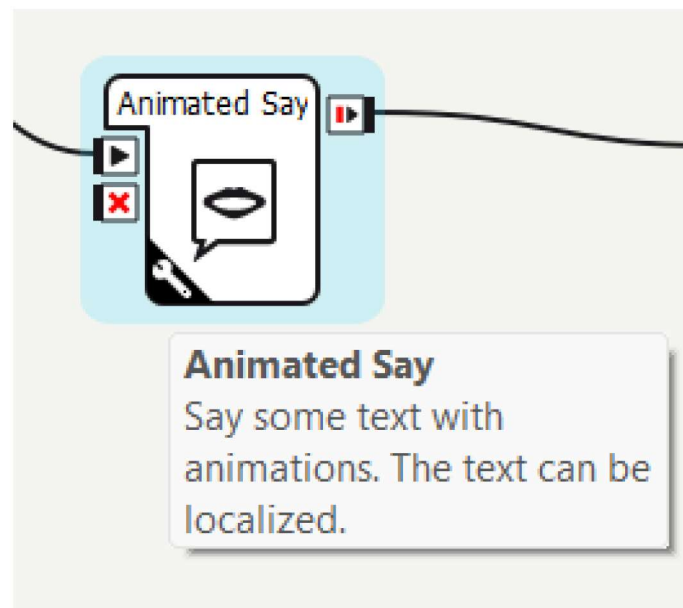


Slika 3.4: Odabir blokovskih funkcija



Slika 3.5: Primjer blokovskih funkcija

Klikom na ikonu u donjem lijevom kutu blokovske funkcije (vidi sliku 3.6) možemo postavljati različite parametre blokovske funkcije, a brzim dvoklikom na blokovsku funkciju nam se otvara "Script editor" (vidi sliku 3.7) u kojemu možemo samo stalno mjenjati skriptu ukoliko želimo nešto promijeniti.



Slika 3.6: Postavljanje parametara u blokovskoj funkciji

```

Script editor
Animated Say x Wait x

1 import time
2
3 class MyClass(GeneratedClass):
4     def __init__(self):
5         GeneratedClass.__init__(self, False)
6
7     def onLoad(self):
8         self.tts = self.session().service('ALAnimatedSpeech')
9         self.ttsStop = self.session().service('ALAnimatedSpeech') #Create
another service as wait is blocking if audioout is remote
10        self.bIsRunning = False
11        self.ids = []
12
13    def onUnload(self):
14        for id in self.ids:
15            try:
16                self.ttsStop.stop(id)
17            except:
18                pass
19        while( self.bIsRunning ):
20            time.sleep( 0.2 )
21
22    def onInput_onStart(self):
23        self.bIsRunning = True
24        try:
25            movement = self.getParameter("Speaking movement mode")
26            textParam = self.getParameter("Text")
27            if movement == "disabled":

```

Slika 3.7: "Script editor"

4 | Softveri treće strane

U ovom poglavlju želimo prikazati vanjske tehnologije koje smo koristili u projektu za postizanje naprednih funkcionalnosti potrebnih za interakciju s NAO robotom. Ove third-party tehnologije omogućile su nam pristup alatima i API-jima za obradu prirodnog jezika, prepoznavanje govora, te kreiranje lokalnog web poslužitelja za komunikaciju s robotom. Glavne tehnologije koje ćemo opisati su OpenAI i Flask, koje su ključne za ostvarivanje dvosmjerne komunikacije između korisnika i robota.

4.1 OpenAI

Želja nam je ukratko spomenuti OpenAI tehnologiju kao ključni dio projekta. OpenAI pruža napredne modele za obradu prirodnog jezika, prepoznavanje govora i generiranje odgovora, što omogućuje interaktivnu komunikaciju između korisnika i NAO robota. U projektu smo koristili OpenAI API kako bismo analizirali i interpretirali govorne unose korisnika, generirali odgovarajuće tekstualne odgovore te ih pretvorili u govor koji robot reproducira.

Integracija s OpenAI API-jem omogućuje naprednu analizu prirodnog jezika, što je ključno za humanoidne robote koji komuniciraju na razumljiv i prirodan način. OpenAI tehnologija nam je izuzetno bitna te nam koristi prilikom prepoznavanja govora, obrade prirodnog jezika te generiranja odgovora. Također, OpenAI nam omogućava vrhunsku konverziju generiranog odgovora u audio zapis na hrvatskom jeziku. Za više informacija o ovoj tehnologiji, pogledajte [10].

4.2 Flask

Ovdje vam želimo predstaviti korištenje Flask vanjske tehnologije, koju smo implementirali za postavljanje lokalnog web poslužitelja radi omogućavanja komunikacije između NAO robota i računala. Flask je lagan i jednostavan Python web okvir koji nam omogućuje brzo postavljanje HTTP poslužitelja, što je bilo ključno za primanje i slanje audio datoteka između robota i našeg sustava za obradu [1].

U projektu smo koristili Flask za postavljanje API-ja koji prima audio datoteke snimljene od strane robota, šalje ih na obradu pomoću OpenAI tehnologije te vraća generirani odgovor. Korištenjem Flask servera postigli smo dvosmjernu komunikaciju u stvarnom vremenu, gdje robot može slati podatke na obradu te primiti povratne informacije u obliku govorne datoteke [2].

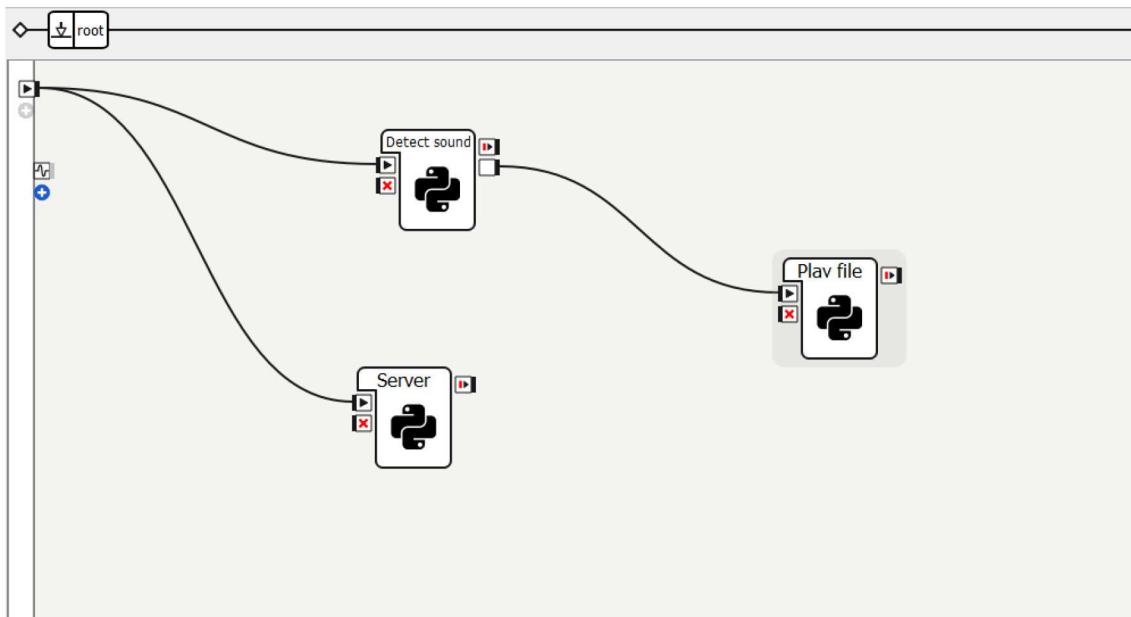
5 | Implementacija

U ovom poglavlju detaljno ćemo objasniti kako smo proveli implementaciju sustava za interakciju s humanoidnim NAO robotom korištenjem detekcije zvuka, obrade prirodnog jezika, i generiranja govorne povratne informacije. Projekt uključuje povezivanje više tehnologija koje omogućuju robotu prepoznavanje i interpretaciju govora, slanje podataka na obradu putem lokalnog servera, te primanje generiranog odgovora koji se reproducira u stvarnom vremenu.

Prva faza implementacije fokusirana je na konfiguraciju i korištenje programskog okruženja *Choregraphe* za kontrolu robota, uključujući skripte za detekciju zvuka i reprodukciju audio datoteka. Korištenje Pythona omogućilo je pristup vanjskim bibliotekama kao što su *speech recognition* za prepoznavanje govora, *pydub* za obradu zvučnih datoteka, te *OpenAI* za generiranje odgovora korištenjem modela umjetne inteligencije. Također, postavili smo *Flask* server za dvosmjernu komunikaciju između robota i računala.

Kroz dijagram u *Choregraphe*-u vidjet ćemo kako se funkcionalnosti detekcije zvuka, obrade i generiranja govora povezuju u jedinstven sustav interakcije. Svaki dio implementacije bit će detaljno opisan u nastavku, uključujući korištene funkcije i algoritme za uspješno provođenje komunikacije između korisnika i robota.

Na slici 6.1 prikazujemo implementaciju dijagrama toka i Python skripti unutar okruženja *Choregraphe*, koji omogućava detekciju zvuka, slanje snimke na server i reprodukciju odgovora na NAO robotu. Na slici 6.1 prikazan je dijagram toka sa tri Python skripte (koje također možemo koristiti kao blokovske funkcije): *Detekcija i pohrana zvuka*, *Reprodukcija zvuka* i *Server*. Svaka od njih ima specifičnu ulogu u procesu interakcije s robotom.



Slika 5.1: Dijagram blokovskih funkcija

5.1 Skripta *Detekcija i pohrana zvuka*

Skripta *Detekcija i pohrana zvuka* odgovorna je za detekciju zvuka u okolini robota i započinjanje snimanja kada detektira zvuk iznad određenog praga glasnoće. Proces se sastoji od sljedećih koraka:

- **Detekcija zvuka:** NAO robot koristi mikrofone za mjerenje energije zvuka u svojoj okolini. Kada detektira zvuk s razinom energije većom od praga (npr. 1400), skripta pokreće snimanje.
- **Praćenje tišine:** Nakon što započne snimanje, skripta prati razinu buke. Ako razina buke padne ispod praga (npr. 700) i ostane tišina dulje od jedne sekunde, snimanje prestaje.
- **Pohrana snimke:** Snimljeni zvuk pohranjuje se kao *recording.wav* na NAO-ovom računalu.
- **Slanje datoteke:** Nakon završetka snimanja, zvučna datoteka se šalje na postavljenu *Flask* server za daljnju obradu pomoću POST zahtjeva.

5.2 Skripta *Reprodukcija zvuka*

Skripta *Reprodukcija zvuka* omogućava reprodukciju odgovora na robotu pomoću *ALAudioPlayer* modula. Glavni koraci u ovoj skripti uključuju:

- **Učitavanje audio datoteke:** Skripta traži audio datoteku *returned_audio.wav*, koju je prethodno preuzeo s *Flask* servera.
- **Reprodukcija zvuka:** Datoteka se reproducira pomoću *ALAudioPlayer* modula, čime robot izgovara odgovor korisniku.

5.3 Skripta *Server*

Skripta *Server* služi za pokretanje jednostavnog HTTP poslužitelja unutar Choregraphe okruženja, čime omogućujemo komunikaciju između robota NAO i vanjskog sustava putem mreže. Ovaj poslužitelj je osmišljen tako da može primiti zahtjeve, prvenstveno u obliku POST zahtjeva, koji prenose audio datoteke ili druge podatke između robota i računala. Ova skripta omogućava da robot prima audio datoteke, koje se koriste za daljnju obradu govora u vanjskom sustavu.

Važno je naglasiti da je za spajanje s robotom, direktno putem Pythona, potreban IP robota. IP robota dobijemo na način da pritisnemo „NAO“ tipku na robotu. On će na glas izreći svoj IP.



Slika 5.2: Robot izgovara svoj IP

5.4 Flask Server i Obrada Zvuka

Ova skripta implementira Flask server koji omogućava dvosmjernu komunikaciju između robota i računala te obavlja sve ključne funkcionalnosti vezane za prepoznavanje govora, generiranje odgovora putem OpenAI API-ja i slanje rezultirajućeg audio zapisa natrag na robota. Server omogućava jednostavno slanje audio datoteka robota na obradu, generiranje odgovora, i vraćanje odgovora kao audio zapisa.

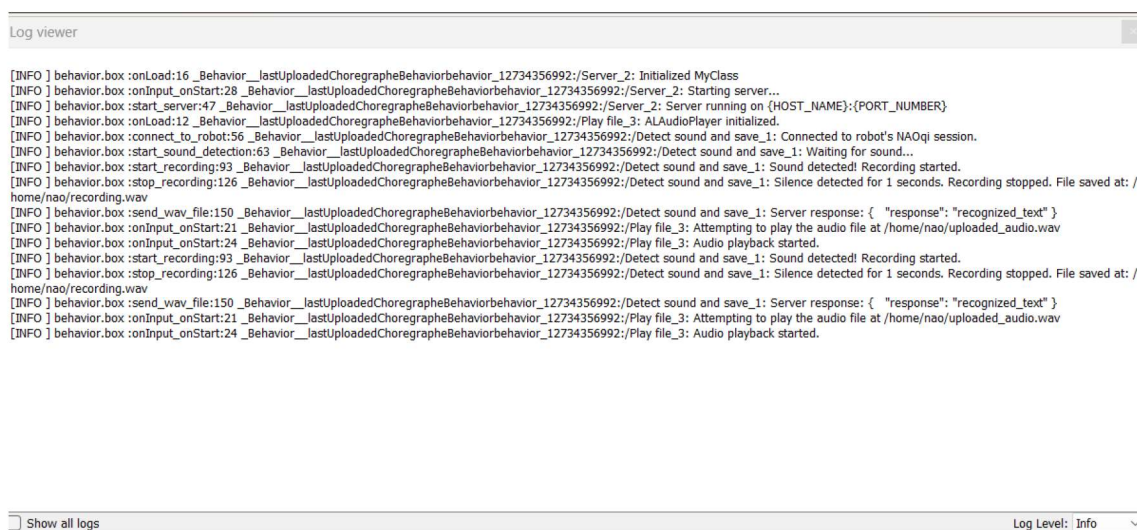
5.4.1 Ključne Funkcionalnosti Skripte

- **Pokretanje Flask Servera:** Flask server se pokreće pomoću `app.run()` komande, gdje server sluša zahtjeve na `0.0.0.0` IP adresi i portu `5000`. Ova konfiguracija omogućava da server prima POST zahtjeve iz lokalne mreže, čime omogućuje prijenos audio datoteka između robota i računala.
- **Prepoznavanje Govora:** Funkcija `convert_audio_to_text()` koristi biblioteku `speech_recognition` za prepoznavanje govora u primljenim audio datotekama. Audio datoteka se prvo konvertira u mono format, a zatim se koristi Google-ov API za prepoznavanje govora na hrvatskom jeziku.
- **Komunikacija s OpenAI API-jem:** Nakon prepoznavanja govora, generirani tekst se šalje OpenAI API-ju kako bi se generirao odgovor temeljen na prepoznatom tekstu. Skripta koristi OpenAI-jev model za prirodni jezik kako bi omogućila logičan i kontekstualno relevantan odgovor robota na hrvatskom jeziku.
- **Generiranje i Slanje Audio Odgovora:** Generirani odgovor iz OpenAI API-ja se zatim pretvara u audio zapis na hrvatskom jeziku. Audio zapis se sprema kao `output.wav` i šalje natrag robotu pomoću funkcije `send_file_to_server()`. Ova funkcija koristi POST zahtjev za prijenos audio datoteke na definiranu IP adresu i port robota.
- **Prijem Audio Datoteka:** PUT zahtjev `upload_file()` omogućava serveru da prima audio datoteke poslane od robota. Ova funkcija pohranjuje audio datoteku kao `recording.wav` i pokreće prepoznavanje govora na primljenoj datoteci.

Ova skripta omogućava složenu interakciju između robota i korisnika putem prirodnog jezika, omogućavajući prepoznavanje govora, generiranje odgovora putem OpenAI API-ja te vraćanje odgovora kao audio zapis za daljnje korištenje u robotu.

6 | Rezultati

6.1 Analiza logova



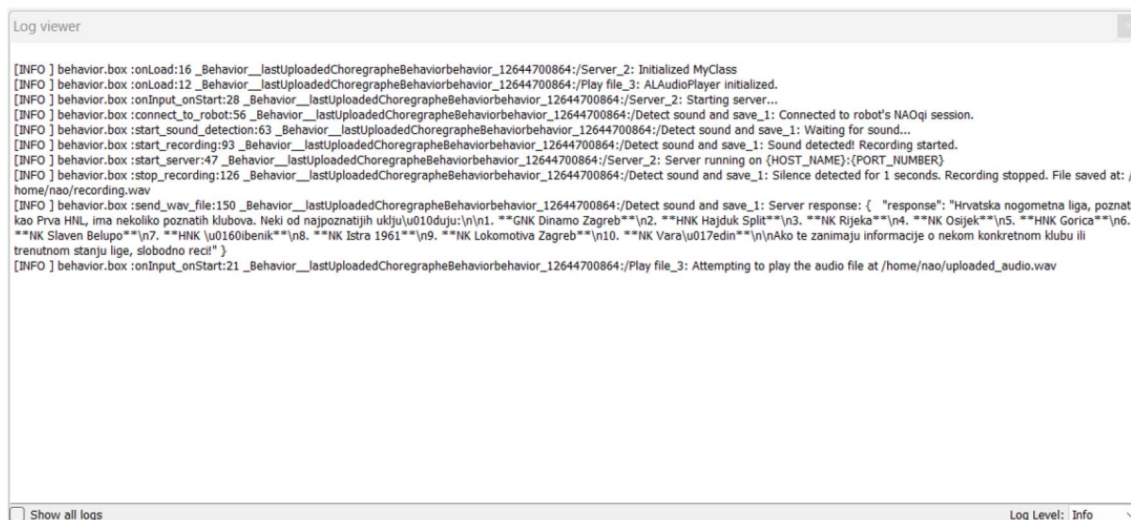
Slika 6.1: Prikaz log-a

Na slici je prikazan log koji pokazuje tijek rada skripte unutar *Choregraphe* softvera za kontrolu NAO robota. Kroz logove možemo pratiti ključne korake u procesu:

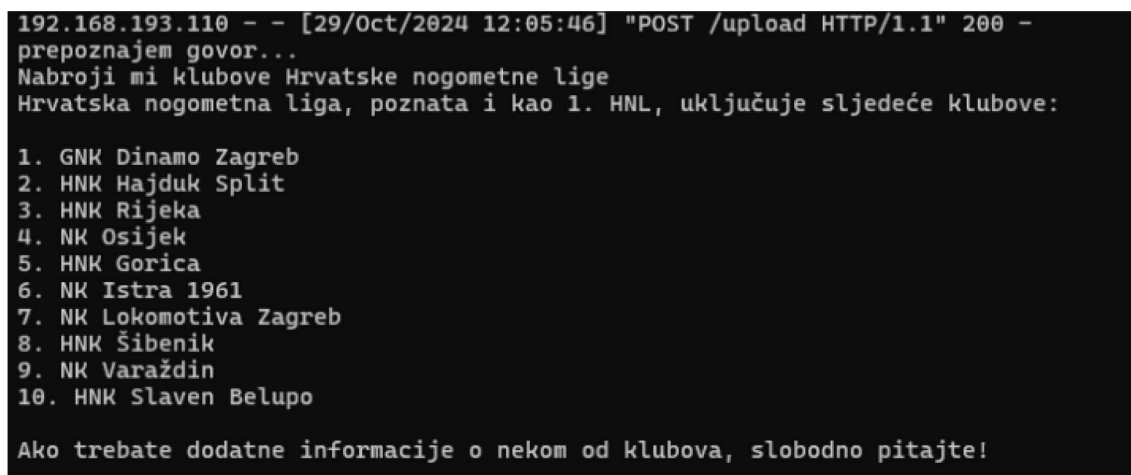
- **Inicijalizacija Servera:** Server je pokrenut pomoću metode `start_server()` i uspješno je aktiviran na navedenom hostu i portu.
- **Detekcija Zvuka:** Nakon inicijalizacije, aktivira se funkcionalnost detekcije zvuka (Detekcija i pohrana zvuka). Kada robot prepozna zvuk koji prelazi zadani prag, započinje snimanje.
- **Snimanje i Pohrana Zvuka:** Zapisuje se informacija o početku snimanja te o završetku snimanja nakon što je detektirana tišina od 1 sekunde. Audio datoteka je pohranjena na lokaciji `/home/nao/recording.wav`.
- **Slanje i Obrada Zvuka na Serveru:** Snimljeni zvuk se šalje na server, gdje se obrađuje i prepoznaje tekst iz zvuka. Server odgovara s tekstualnim odgovorom koji se prikazuje u logu kao `Server response: "recognized_text"`.

- **Reprodukcija Odgovora:** Na kraju, NAO robot reproducira primljeni audio zapis pomoću ALAudioPlayer modula, što je zabilježeno u logovima kao Audio playback started.

Ovaj log omogućava praćenje rada sustava i potvrdu da sve funkcionalnosti, od detekcije zvuka do reprodukcije odgovora, ispravno funkcioniraju.

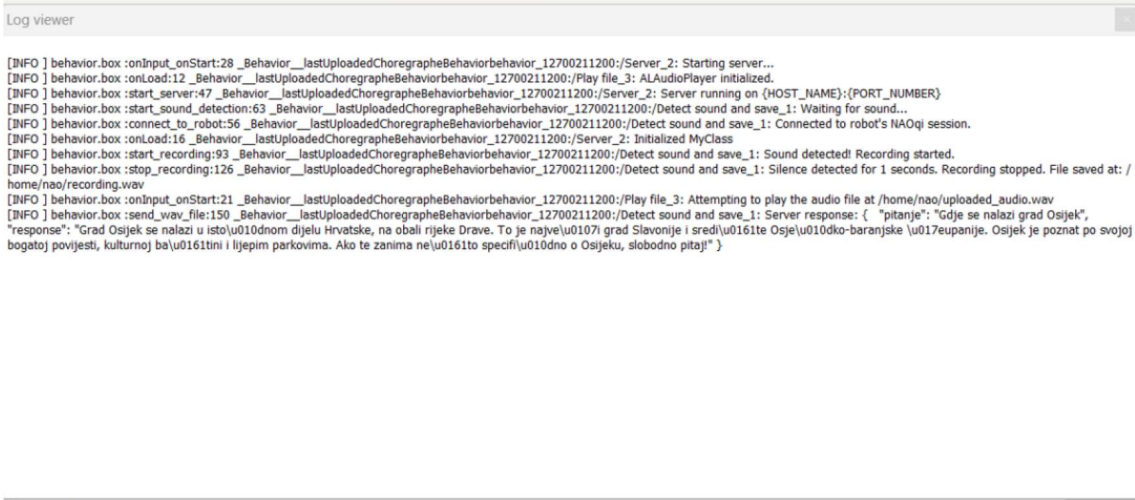


Slika 6.2: Choregraphe "Log viewer"



Slika 6.3: Server log

Na slici 6.2 eksplicitno vidimo odgovor koji smo dobili u "Log viewer" dijelu Choregraphe softvera dok na slici 6.3 vidimo točno ono što je server dobio iz audio zapisa koji je dobio od robota.



```

[INFO ] behavior.box :onInput_onStart:28 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12700211200;/Server_2: Starting server...
[INFO ] behavior.box :onLoad:12 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12700211200;/Play file_3: ALAudioPlayer initialized.
[INFO ] behavior.box :start_server:47 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12700211200;/Server_2: Server running on (HOST_NAME):(PORT_NUMBER)
[INFO ] behavior.box :start_sound_detection:63 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12700211200;/Detect sound and save_1: Waiting for sound...
[INFO ] behavior.box :connect_to_robot:56 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12700211200;/Detect sound and save_1: Connected to robot's NAOqi session.
[INFO ] behavior.box :onLoad:16 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12700211200;/Server_2: Initialized MyClass
[INFO ] behavior.box :start_recording:93 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12700211200;/Detect sound and save_1: Sound detected! Recording started.
[INFO ] behavior.box :stop_recording:126 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12700211200;/Detect sound and save_1: Silence detected for 1 seconds. Recording stopped. File saved at: /home/nao/recording.wav
[INFO ] behavior.box :onInput_onStart:21 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12700211200;/Play file_3: Attempting to play the audio file at /home/nao/uploaded_audio.wav
[INFO ] behavior.box :send_wav_file:150 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12700211200;/Detect sound and save_1: Server response: { "pitanje": "Gdje se nalazi grad Osijek", "response": "Grad Osijek se nalazi u isto\u010dnom dijelu Hrvatske, na obali rijeke Drave. To je najve\u0107i grad Slavonije i sredi\u0161te Osje\u010dsko-baranjske (u017eupanje. Osijek je poznat po svojoj bogatoj povijesti, kulturnoj ba\u0161tini i lijepim parkovima. Ako te zanima ne\u010di\u0161to specifi\u010dno o Osijeku, slobodno pitaj!" }

```

Show all logs Log Level: Info

Slika 6.4: Gdje se nalazi grad Osijek?



```

[INFO ] behavior.box :onInput_onStart:28 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Server_2: Starting server...
[INFO ] behavior.box :connect_to_robot:56 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Detect sound and save_1: Connected to robot's NAOqi session.
[INFO ] behavior.box :onLoad:12 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Play file_3: ALAudioPlayer initialized.
[INFO ] behavior.box :start_server:47 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Server_2: Server running on (HOST_NAME):(PORT_NUMBER)
[INFO ] behavior.box :start_sound_detection:63 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Detect sound and save_1: Waiting for sound...
[INFO ] behavior.box :start_recording:93 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Detect sound and save_1: Sound detected! Recording started.
[INFO ] behavior.box :stop_recording:126 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Detect sound and save_1: Silence detected for 1 seconds. Recording stopped. File saved at: /home/nao/recording.wav
[INFO ] behavior.box :onInput_onStart:21 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Play file_3: Attempting to play the audio file at /home/nao/uploaded_audio.wav
[INFO ] behavior.box :send_wav_file:150 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Detect sound and save_1: Server response: { "pitanje": "reci mi ne\u0161to o Japanu", "response": "Japan je zemlja koja se sastoji od niza otoka na istoku Azije, a najpoznatiji su otoci Honshu, Hokkaido, Kyushu i Shikoku. Glavni grad Japana je Tokio, koji je jedan od najnaseljenijih gradova na svijetu. Japan je poznat po svojoj bogatoj kulturi, tradiciji, kao i naprednoj tehnologiji.\n\nOva zemlja ima jedinstvenu povijest, koja obuhva\u0107a feudalno razdoblje, period samuraja, kao i Meiji restauraciju koja je donijela brze promjene i modernizaciju. Japan je tako\u0111er poznat po svojim tradicionalnim umjetnostima kao \u0161to su haiku poezija, \u0161to su haiku poezija, \u0161to su haiku poezija, i kaligrafija.\n\nJapanska kuhinja, uklju\u010duju\u0107i sushi, ramen, i tempuru, u\u0107eiva veliku popularnost diljem svijeta. Osim toga, Japan je poznat i po svojim prirodnim ljepotama, uklju\u010duju\u0107i planinu Fuji i razne nacionalne parkove.\n\nTehnologija i inovacije su tako\u0111er ključni dijelovi japanskog dru\u0161tva, s mnogim poznatim kompanijama koje rade u industrijama poput elektronike, automobilske proizvodnje i robotike. \n\nJapan je tako\u0111er poznat po svojim festivalima, anime i manga kulturi, koji su postali popularni \u0161to od svijeta. Sve u svemu, Japan je fascinantna zemlja koja nudi spoj tradicije i modernosti." }
[INFO ] behavior.box :onInput_onStart:24 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Play file_3: Audio playback started.
[INFO ] behavior.box :start_recording:93 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Detect sound and save_1: Sound detected! Recording started.
[INFO ] behavior.box :stop_recording:126 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Detect sound and save_1: Silence detected for 1 seconds. Recording stopped. File saved at: /home/nao/recording.wav
[INFO ] behavior.box :send_wav_file:150 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Detect sound and save_1: Server response: { "pitanje": "reci mi ne\u0161to o republici Hrvatskoj", "response": "Republika Hrvatska je dr\u0161tvo na istoku, Bosnom i Hercegovinom na jugoistoku, na raskri\u0161ju srednje i jugoisto\u010dne Europe. Grani\u0107i se Italijom na zapadu, Slovenijom na sjeveru, Ma\u0161arskom na sjeveroistoku, Srbijom na istoku, Bosnom i Hercegovinom na jugu. Ima izlaz na Jadransko more, koje je poznato po svojoj ljepoti i brojnim otocima.\n\nGlavni grad Hrvatske je Zagreb, koji je i najve\u0107i grad u zemlji. Hrvatska ima bogatu povijest, kulturnu ba\u0161tinu i raznoliku prirodnu ljepotu, uklju\u010duju\u0107i nacionalne parkove poput Plitvi\u010dkih jezera i Krke. Hrvatska je poznata po svom turizmu, s brojnim turistima koji dolaze u Hrvatsku kao \u0161to su Dubrovnik, Split, i Hvar.\n\nRepublika Hrvatska (\u010dlanica je Europske unije od 1. srpnja 2013. godine, a koristi euro kao svoju slu\u0161ebenu valutu od 1. sije\u010dnja 2023. godine. Jezik koji se govori u Hrvatskoj je hrvatski, koji koristi latinicu. Hrvatske tradicije i obiluje tako\u0111er su vrlo raznoliki i variraju od regije do regije.\n\nAko \u0161to saznati ne\u0161to specifi\u010dno o Hrvatskoj, slobodno pitaj!" }
[INFO ] behavior.box :onInput_onStart:21 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Play file_3: Attempting to play the audio file at /home/nao/uploaded_audio.wav
[INFO ] behavior.box :onInput_onStart:24 _Behavior__lastUploadedChoregrapheBehaviorbehavior_12608869056;/Play file_3: Audio playback started.

```

Show all logs Log Level: Info

Slika 6.5: Primjer s više upita

Na prikazanoj slici 6.5 možemo vidjeti primjer loga u kojem su postavljena dva uzastopna pitanja. Robot je uspješno obradio oba upita koristeći *OpenAI* tehnologiju za generiranje odgovora i vratio rezultate. Iz ovoga možemo zaključiti da u realnom vremenu možemo nastaviti razgovarati odnosno postavljati pitanja robotu. Prvi odgovor se odnosi na pitanje o Japanu, dok se drugi odgovor odnosi na pitanje o Republici Hrvatskoj.

- **Odgovor na prvo pitanje "reci mi nešto o Japanu":** Japan je zemlja koja se sastoji od nizova otoka na istoku Azije, a najpoznatiji su otoci Honshu, Hokkaido, Kyushu i Shikoku. Glavni grad Japana je Tokio, koji je jedan od najnaseljenijih gradova na svijetu. Japan je poznat po svojoj bogatoj kulturi, tradiciji, kao i naprednoj tehnologiji. Ovo zemlja ima jedinstvenu povijest, koja obuhvaća doba feudalnog razdoblja, period samuraja, kao i Meiji restauraciju koja je donijela velike promjene i modernizaciju. Japan je također poznat po svojim tradicionalnim umjetnostima kao što su haiku poezija, kaligrafija i ikebana, kao i japanska kuhinja, uključujući sushi, ramen, i tempuru, uživa veliku popularnost diljem svijeta. Osim toga, Japan je poznat i po svojim prirodnim ljepotama, uključujući planinu Fuji i razne nacionalne parkove. Inovacija i tehnologija u Japanu uključuju divove japanskog društva, s mnogim poznatim kompanijama koje rade u industrijama poput elektronike, automobilske proizvodnje i robotike. Japan je također izuzetno poznat po svojim festivalima, anime i manga kulturi, koji su postali popularni diljem svijeta. Sve u svemu, Japan je fascinantna zemlja koja nudi spoj tradicije i modernosti.
- **Odgovor na drugo pitanje "reci mi nešto o Republici Hrvatskoj":** Republika Hrvatska je država smještena u jugoistočnoj Europi, na raskrižju srednje i jugoistočne Europe. Graniči s Italijom na zapadu, Slovenijom na sjeveru, Mađarskom na sjeveroistoku, Srbijom na istoku, Bosnom i Hercegovinom na jugoistoku, te Crnom Gorom na jugu. Ima izlaz na Jadransko more, koje je poznato po svojoj ljepoti i brojnim otocima. Glavni grad Hrvatske je Zagreb, koji je i najveći grad u zemlji. Hrvatska je poznata po svojoj bogatoj povijesti, kulturi i običajima i raznolika prirodnim ljepotama poput Plitvičkih jezera i Krke. Hrvatska je poznata po svom turizmu, s brojnim turističkim atrakcijama kao što su Dubrovnik, Split i Hvar. Republika Hrvatska članica je Europske unije od 1. srpnja 2013. godine, a koristi euro kao svoju valutu od 1. siječnja 2023. godine. Jezik koji se govori u Hrvatskoj je hrvatski, koji koristi latinicu. Hrvatske tradicije i običaji variraju u različitim dijelovima zemlje, a koristi autohtona jela poput pršuta i raznih vrsta ribe. Ako želite saznati nešto specifično o Hrvatskoj, slobodno pitaj!

Ovi odgovori prikazani su u logu kao dio interakcije između robota i korisnika te potvrđuju uspješnu dvosmjernu komunikaciju kroz postavljanje pitanja i generiranje odgovora.

Literatura

- [1] Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.
- [2] Massé, M. (2011). *REST API Design Rulebook*. O'Reilly Media.
- [3] *Choregraphe Overview*, dostupno na http://doc.aldebaran.com/1-14/software/choregraphe/choregraphe_overview.html
- [4] *Python - About*, dostupno na <https://www.python.org/about/>
- [5] *Home NAO*, dostupno na http://doc.aldebaran.com/2-1/home_ao.html
- [6] *NAO 6 Support - Downloads*, dostupno na <https://www.aldebaran.com/en/support/nao-6/downloads-softwares>
- [7] *SoftBank Robotics*, dostupno na <https://www.softbankrobotics.com/>
- [8] *Choregraphe Index*, dostupno na <http://doc.aldebaran.com/1-14/software/choregraphe/index.html>
- [9] *Python - Downloads*, dostupno na <https://www.python.org/downloads/>
- [10] *OpenAI - API*, dostupno na <https://platform.openai.com/docs/api-reference/introduction>
- [11] *Flask Documentation*, dostupno na <https://flask.palletsprojects.com/en/stable/>