

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Domagoj Rojnić

Diplomski studij

ZADAVANJE NAREDBI WINDOWS SUSTAVU GLASOM

Raspoznavanje uzoraka i strojno učenje

Osijek, 2021.

SADRŽAJ

1. Uvod	1
1.1. Opis projektnog zadatka	1
2. Programsko rješenje.....	2
2.1. SpeechRecognition.....	2
2.2. PyAutoGUI	3
3. ZAKLJUČAK	5
4. LITERATURA	6

1. Uvod

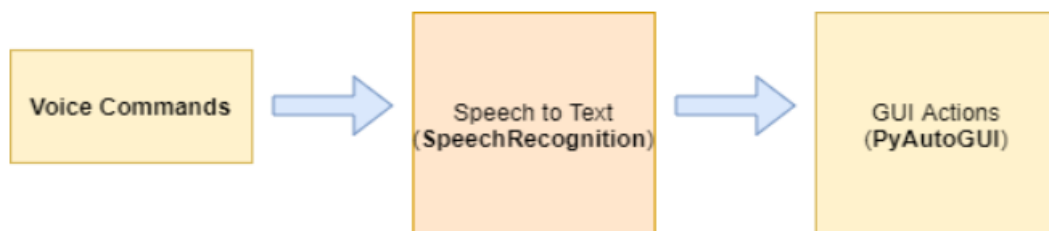
Windows sustav pruža veliki niz mogućnosti korisnicima. Daje jednostavno grafičko sučelje pomoću kojeg je lako izvoditi niz naredbi. Osim „klasičnog“ klikanja i pokretanja programa, moguće je izvođenje naredbi putem glasa, što je i cilj ovoga projekta.

1.1. Opis projektnog zadatka

Cilj ovog projektnog zadatka bio je implementirati sustav koji omogućuje izvođenje Windows naredbi putem glasa. Odabrana je nekolicina naredbi koje se izvršavaju pomoću različitih biblioteka putem Python skripte.

2. Programsko rješenje

Prethodno naveden cilj projekta također se može prikazati dijagramom na slici 2.1. Korisnik kao ulaz daje glasovnu naredbu putem mikrofona, koji može biti ili ugrađen u Windows računalo ili eksterno povezan, nakon čega se glasovna naredba pretvara u text koristeći metode SpeechRecognition biblioteke. Konvertirani tekst je tada mapiran pojedinim GUI zadacima, poput upravljanja mišem ili tipkovnicom, prilikom čega su aktivnosti izvedene pomoću PyAutoGUI biblioteke.



Slika 2.1. Dijagram zadatka

2.1. SpeechRecognition

SpeechRecognition Python biblioteka nudi veliki niz mogućnosti u različitim aspektima kućnih automatskih sustava, učenja jezika pa sve do digitalnih asistenata. Korištenjem ove biblioteke omogućena je pretvorba govora u tekst. Podupire nekolicinu API-ja, *online* te čak i *offline*. U ovome zadatku, korišten je Online Google API. Moguće je primjetiti zaostajanje sustava prilikom konverzije govora u tekst, no odabran API se pokazao najtočnijim od ostalih mogućih rješenja. Instalacija biblioteke u projekt izvodi se kako je prikazano na slici 2.1.1. prilikom čega se također instalira pyaudio biblioteka kako bi se moglo pristupiti mikrofону.

```
pip install SpeechRecognition
pip install pyaudio
```

Slika 2.1.1. Instaliranje biblioteka

Sljedećom slikom, vidljiv je kod kojim se uvodi biblioteka te inicijalizira recognizer objekt koji sluša glas putem mikrofona i pretvara ga u tekst koristeći `recognize_google` funkciju. Pozivom funkcije `recognizer.adjust_for_ambient_noise()` je opcionalno, ali vrlo korisno. Ova funkcija sluša ulazni zvuk i mijenja vrijednost `energy_threshold` ovisno o količini buke koja je prisutna u sobi. Ovom funkcijom se mjeri „osjetljivost“ recognizer objekta. Dobivena vrijednost varira u rasponu od 50 do 4000 gdje veće vrijednosti predstavljaju više buke i manju osjetljivost na ulazni signal govora.

```
import speech_recognition

recognizer = speech_recognition.Recognizer()

with speech_recognition.Microphone() as src:
    try:
        audio = recognizer.adjust_for_ambient_noise(src)
        print("Threshold Value After calibration:" +
              str(recognizer.energy_threshold))
        print("Please speak:")
        audio = recognizer.listen(src)
        speech_to_txt = recognizer.recognize_google(audio).lower()
        print(speech_to_txt)
    except Exception as ex:
        print("Sorry. Could not understand.")
```

Slika 2.1.2. Početni dio koda

2.2. PyAutoGUI

Python nudi biblioteku PyAutoGUI za GUI automatizaciju koja simulira klikanje miša i pritiskanje tipki na tipkovnici kao da ih i sam čovjek obavlja. Na primjer, moguće je sve od simuliranja kretnje miša, povećavanja i smanjivanja otvorenih prozora pa sve do mijenjanje razine zvuka i otvaranje instaliranih aplikacija. Instaliranje je prikazano na slici 2.2.1. te se izvodi na isti način kao i kod SpeechRecognition biblioteke.

```
pip install PyAutoGUI
```

Slika 2.2.1. Instaliranje PyAutoGUI biblioteke

Kretnje miša simuliraju se na način da se dohvaćaju x i y koordinate na zaslonu. Zaslon se može zamisliti kao 2D prostor čije se (0, 0) koordinate nalaze u gornjem lijevom kutu. Kretnjom udesno povećava se x vrijednost te kretanjem prema dolje povećava se y vrijednost. Funkcijom `pyautogui.size()` dohvaća se dimenzija zaslona.

Metodom `pyautogui.locateOnScreen()` danu sliku pronalazi na zaslonu tako da traži točan raspored piksela kao na slici. Pronalaskim ikonice moguće je odrediti x i y koordinate te simulirati lijevi klik miša na te koordinate metodom `pyautogui.click()`. Na taj način ostvareno je otvaranje Google Chrome i Notepad aplikacija.

Simulacija pritiskanja tipki odrađena je funkcijom `pyautogui.press()` kojoj se kao argument zadaje ime radnje koju tipka izvodi.

Na poslijetku, odrađeno je mapiranje funkcija koje se izvode na određene glasovne naredbe kako je prikazano slikom 2.2.2.

Glasovna naredba	Funkcija
Scroll up	Pomicanje dokumenta prema gore
Scroll down	Pomicanje dokumenta prema dolje
Volume up	Povećanje razine zvuka
Volume down	Smanjenje razine zvuka
Mute/Unmute	Isključivanje/uključivanje zvuka
Open Chrome	Otvaranje Google Chrome preglednika
Open Notepad	Otvaranje Notepad uređivača teksta
Quit/Exit	Prekid izvođenja

Slika 2.2.2. Naredbe aplikacije

3. ZAKLJUČAK

Ovim projektom vidljivo je kako postoji više mogućnosti upravljanja Windows sustavom te je prikazana i korištena mogućnost izvođenja naredbi govorom. Osim korištenog Google API-ja, dostupni su IBM, Microsoft Bing, Wit.ai te mnogi drugi API čiju je implementaciju potrebno dodatno istražiti i implementirati u budućim projektima.

4. LITERATURA

<https://pypi.org/project/SpeechRecognition/>

<https://pypi.org/project/PyAudio/>

<https://pypi.org/project/PyAutoGUI/>

<https://towardsdatascience.com/audio-deep-learning-made-simple-automatic-speech-recognition-asr-how-it-works-716cfce4c706>

<https://towardsdatascience.com/build-a-voice-controlled-mouse-keyboard-in-5-minutes-952bc8f101fc>