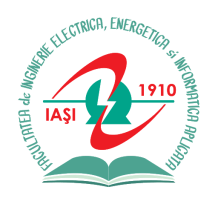
**UNIVERSITATEA TEHNICĂ „GHEORGHE ASACHI”**

**FACULTATEA DE INGINERIE ELECTRICĂ,**

**ENERGETICĂ ŞI INFORMATICĂ APLICATĂ**

**IAȘI**

**PROIECT LABORATOR**

**Programare Python**

**Semestrul I**

**Coordonator proiect: Student:**

**CONF.DR.ING.CĂTĂLIN DAMIAN CHIPERI MIHAI-IULIAN**

1. **INTRODUCERE**

Tema proiectului a fost Speech Recognition, iar prin acest proiect am dorit să consolidez abilitățiile de programare în limbajul Python.

Am dorit să creez prin acest proiect un asistent virtual care să poată înțelege anumite funcții de baza precum: căutarea pe Google, căutarea pe YouTube, generarea unor fișiere de tip .txt, deschiderea unor aplicații, etc.

1. **IDEI PENTRU PROIECT:**

**Obiective realizate:**

* Crearea unui mediu virtual (venv);
* Instalarea librariilor necesare pentru transformarea mesajului vocal în text;
* Crearea unor functii pentru transformarea mesajului vocal în text:
  + def inregistrare\_audio()
  + def inregistrare\_joc()
* Crearea unei funcții pentru a genera un răspuns adecvat mesajului vocal:
  + def raspunsuri()
* Realizarea unui joc în care utlizatorul trebuie să recunoasca steagurile lumii.
  + - Contorizarea punctajului
    - Utilizatorul are posibilitatea de a ieși oricând din joc.
    - Utilizatorul poate sa aleaga un continent anume.
    - Se afisează răspunsul corect.
* Crearea unei funcții cu ajutorul căreia mesajul programului va fi unul audio;
* Cautarea unei melodii si pornirea ei;
* Cautarea in libraria Wikipedia ;
* Countdown Timer
* Realizarea jocului - ghiceste numarul sau ghiceste fraza
* Generarea unei liste
* Realizarea unei functii prin care programul sa porneasca la un cuvant cheie
* Deschiderea unor programe din pc
* Stingerea calculatorului
* Crearea unei interfete grafice la jocuri

**Obiective în curs de realizare:**

**Obiective de realizat:**

1. **MOD DE FUNCȚIONARE**
2. **Module**

Proiectul se folosește de mai multe module pentru a putea transforma cuvintele rostite de utilizator, într-un format digital, care să satisfacă nevoile utilizatorului.

Module utilizate:

* pygame – acest modul a fost folosit pentru crearea unor interfețe grafice celor 2 jocuri.
* os – acest modul a fost utilizat pentru a putea utiliza funcționalități ale sistemului de operare (căutare, ștergere, accesare)
* glob – acest modul a fost folosit pentru a returna fișiere după un anumit criteriu.
* time – acest modul a fost folosit pentru a putea contoriza secundele la cronometru invers
* speech\_recognition – acest modul este cel de bază pentru realizarea recunoașterii vocale
* playsound – acest modul a fost folosit pentru a putea accesa sunetul de pornire înregistrare
* datetime – acest modul a fost folosit pentru a putea citi data si ora
* pywhatkit – acest modul a fost folosit pentru a căuta pe Google și pe YouTube
* wikipedia – acest modul a fost folosit pentru a căuta informatii pe wikipedia
* pyttsx3 – acest modul a fost folosit pentru a putea returna un răspuns audio

1. **Funcții**

Explicare pe scurt cu rol are fiecare funcție din program:

def inregistrare\_audio() – această funcție folosește microfonul pentru a putea înregistra si returna cuvintele rostite de utilizator. Pe durata la o secunda dupa apelarea functiei, microfonul analizează zgomotul de fundal pentru a putea să fie eliminat, ajutând astfel o comunicare mai bună între program și utilizator.

def inregistrare\_joc() – această funție are același rol ca funcția anterioară, diferența fiind că înregistreză pe durata a 5 secunde. Funcția este folosita în cadrul jocului „Ghicește steagul”, atunci când utlizatorul trebuie să ghiceastă care este steagul într-o durata de 5 secunde.

def voce\_bot() – această funcție se folosește în special de modulul pyttsx3, ajutând astfel la returnarea unui mesaj audio.

Această funcție se execută doar pe un sistem de operare Windows, deoarece se foloseste de funcția interna Speech to Text implementată de Microsoft in sistemul de operare Windows 10.

def locatie() – această funcție este baza jocului „Ghicește steagul”. Jocul are o mică fereastră în care se afisează steagul unei țări, numele țării respective se citește automat din folderul în care se află (numele fișierului .png trebuie să coincida cu numele țării respective). Dacă utlizatorul ghicește steagul, primește un mesaj vocal de confirmare răspuns corect și se trece la următorul steag.

La final utlizatorul primește un mesaj audio cu câte țări a ghicit.

Model afișare:

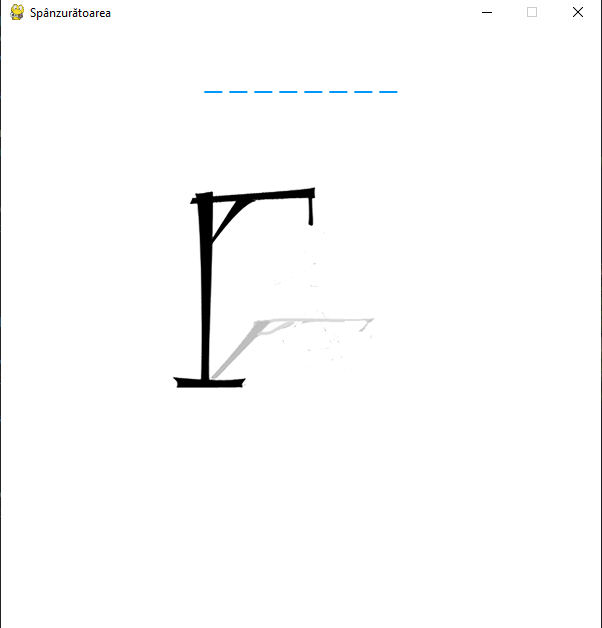


**Figura 1**

def set\_temporizator() – această funcție ajută la stabilirea timpilor de cronometrare inversă. Utilizatorul este întrebat câte minute și câte secunde dorește să se numere invers. Dacă utilizatorul spune un cuvânt care nu este un numar întreg, acesta primeste un mesaj audio („Te rog să introduci un număr valid!”), acest mesaj dispare în momentul în care utilizatorul introduce un număr valid. Dupa introducerea minutelor, acestea sunt transformate în secunde, iar împreună cu secundele introduse anterior, sunt adunate într-o variabilă.

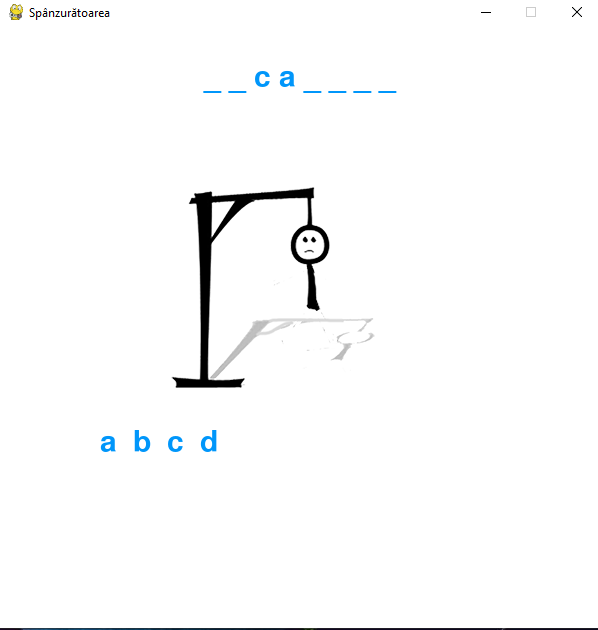
Această variabilă este apelata de către funcția def temporizator(), care se foloseste de modului time, scăzând din secundele totale câte o secunda, pâna când secundele totale ajung la 0. În momentul îndeplinirii acestei condiții utilizatorul primște un mesaj sonor.

def ghiceste\_cuvant() – în această funcție am încercat să reproduc jocul Spânzuratoarea. După apelarea funcției utilizatorul are un scurt mesaj audio, în care se expun categoriile din care poate să primească cuvântul. Daca utilizatorul spune o categorie care nu se află in joc, acesta primește un mesaj audio în care este rugat să aleagă o categorie care se află in joc. Odată ce categoria este aleasă, se deschide o fereastră cu jocul Spânzuratoarea.



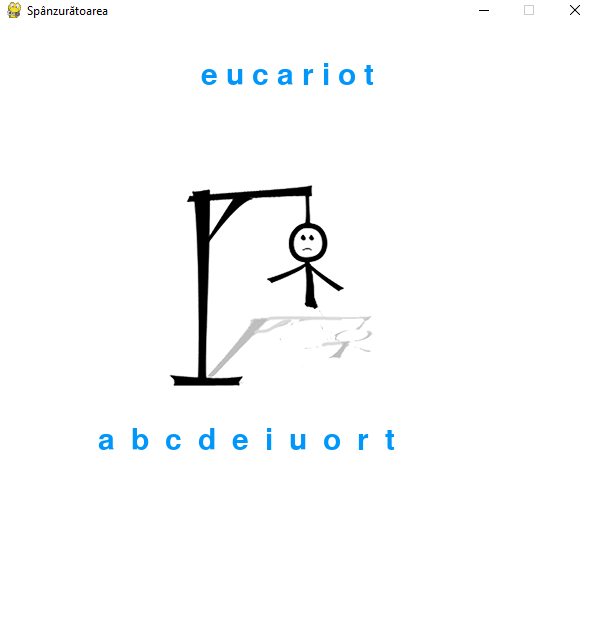
**Figura 2**

După afișarea ferestrei, utilizatorul trebuie să spună litera dorită, aceasta o sa fie introdusă sub forma unui mesaj vocal: „litera + litera dorită” (***din cauză că******modulul speech\_recognition nu este atât de bine optimizat pentru limba româna, de foarte multe ori încurca „litera e” cu „litera i” sau invers, așa ca am optat pentru a confirma literele respective***), dacă utilizatorul nu introduce sub acest format primește un mesaj audio că litera nu este corect introdusă. Dacă litera introdusă de utilizator se regăseste în cuvant, fereastra jocului se actualizează cu litera ghicită. Dacă utilizatorul nu ghicește litera, fereastra jocului incepe să deseneze acel „om”. Toate literele spuse de utilizator apar în fereastră, iar dacă utilizatorul spune de 2 ori aceeasi literă, acesta primește un mesaj audio în care este atentionat că litera este deja introdusă.



**Figura 3**

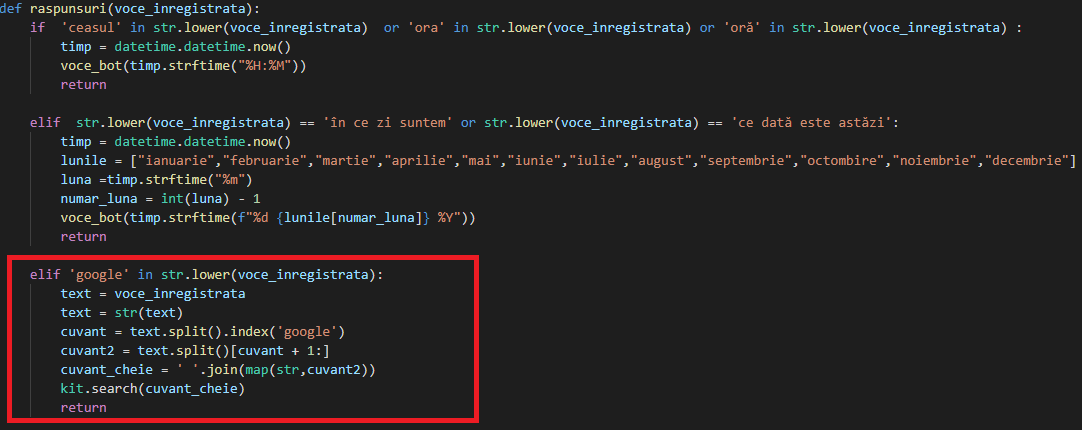
Dacă utilizatorul nu ghicește cuvantul, înainte să fie desenat tot omul, acesta pierde, la final primind un mesaj audio în care se spune că a pierdut și care a fost cuvântul, iar in caz contrar, utilizatorul câștigă.



**Figura 4**

def ghiceste\_cuvant() – această funcție ajută la oferirea de răspunsuri, în funcție de ce cere utilizatorul. Funcția este formata din if si elif astfel căutând cuvintele cheie și returnând un răspuns.

De exemplu, dacă utilizatorul spune: ”Caută pe google mașini”, programul folosește doar partea marcata cu roșu (figura 5), deoarece conține cuvântul cheie „Google”. Textul introdus de utilizator este împarțit in 2, așadar este facut un split, pentru a putea căuta ca și cuvant/cuvinte cheie pe www.google.ro doar cuvintele aflate după cuvantul cheie google introdus de utilizator.



**Figura 5**

def deschidere\_aplicatii() – acestă funcție are același principiu de funcționare precum funcția de sus, diferența fiind ca în această funcții se deschid programe precum: Google Chrome, camera de la laptop, Microsoft Excel, Microsoft Word.

def lista() – această funcție ajută la generarea unor liste cu activități pe care trebuie să le execute utilizatorul sau o listă de cumpărături. Aceasta lista este salvată într-un fișier de tip .txt. Numele listei este introdus de utilizator. Condiția de oprire scriere a listei este dată de cuvintele „listă finalizată” sau „gata”.

def prezentare() – această funcție este prima care se apelează, utilizatorul făcând o mică cunoștintă cu asistentul virtual. Utilizatorul își spune numele, iar in funcție de ora când deschide programul o să primească un mesaj audio de întâmpinare: Bună dimineața/ Bună ziua/ Bună seara + numele utilizatorului.

După ce se face prezentarea, programul apelează funcția def start1(), care ajută la căutarea unui raspuns adecvat cerinței puse de utilizator. După returnarea unui rezultat, programul apelează funcția def start2(), trecând astfel în așteptare, până când aude cuvantul cheie „hei” apelând din nou funcția def start1().

1. **SUSRSE DE DOCUMENTARE:**

* [**https://www.w3schools.com/python/default.asp**](https://www.w3schools.com/python/default.asp)
* [**https://realpython.com/**](https://realpython.com/)
* [**https://www.geeksforgeeks.org/python-programming-language/**](https://www.geeksforgeeks.org/python-programming-language/)
* **Imagini joc Spânzurătoarea:** [**https://drive.google.com/drive/folders/1VXFa0yNynZNFN-m6VoOM-Ri5gkSi89ym**](https://drive.google.com/drive/folders/1VXFa0yNynZNFN-m6VoOM-Ri5gkSi89ym)

Link GitHub:[**https://github.com/Chiperi2000/Proiect**](https://github.com/Chiperi2000/Proiect)