**Documentatie 2**

***Prerequisites si configurare:***

Este nevoie să aveți instalată o versiune a Python pe calculator, un mediu IDE si biblioteca Tkinter.

Se deschide fisierul joc.py. Se ruleaza codul fie intrand in meniul „Run” si apansand pe butonul „Run Module”, fie apasand tasta „F5”.

Alternativ, jocul se poate rula intrand in consola si folosind comanda **python joc.py**.

(locatia folosita in terminal trebuie sa coincida cu path-ul fisierului).

***Explicarea Aplicatiei :***

Aplicatia deschide o fereastra interactiva, alege un cuvant aleatoriu din lista de cuvinte si afiseaza prima poza a spanzuratorii, numarul de incercari ramase si cuvantul reprezentat de un sir de linii, fiecare linie reprezentand o litera a cuvantului,

Utilizatorul trebuie sa introduca o litera in casuta aplicatiei interactive si sa apese butonul „Ghiceste o litera”. Aceeasi litera nu poate fi ghicita de doua ori.

Daca aceasta litera se regaseste in cuvantul ales, programul compleataza pozitiile unde aceasta apare.

Daca litera nu se regaseste in cuvantul ales, animatia jocului progreasa cu o parte a corpului.

Jocul se termina cand toate literele din cuvantul ales au fost ghicite, utilizatorul castigand jocul, sau desenul este completat, utilizatorul pierzand jocul.

***Explicarea codului joc.py:***

**Import random :** modulul care ajuta sa alegem un cuvant aleatoriu din lista de cuvinte.

**Import tkinter as tk** : biblioteca Python pentru a crea o interfata grafica.

**hangman\_stages = [...]** : este o lista care contine animatiile pentru fiecare etapa a jocului de „Spanzuratoarea”.

**words = [...]** : lista de cuvinte din care din care se va alege unul aleatoriu.

**word = random.choice(words)** : functia care selecteaza un cuvant aleatoriu din lista „words”.

**guesses = ’ ’** : sirul care stocheaza literele ghicite de catre jucator, este initializat ca un sir gol.

**turns = len(hangman\_stages) - 1** : este numarul de incercari pe care le are jucatorul.

**def update\_game():**

**global guesses, turns**

**display\_word = ""**

**failed = 0**

**for char in word:**

**if char in guesses:**

**display\_word += char + " "**

**else:**

**display\_word += "\_ "**

**failed += 1**

**word\_label.config(text=display\_word.strip())**

: Functia care actualizeaza starea jocului. Afiseaza cuvantul cu literele ghicite si cele necunoscute si actulizeaza starea jocului. Variabila „failed” numara cate litere din cuvant nu au fost ghicite inca.

**if failed == 0:**

**result\_label.config(text="Felicitări, ai câștigat! Cuvântul era: " + word)**

**disable\_game()**

**return**

: daca toate literele au fost ghicite, inseamna ca jucatorul a castigat si se afiseaza mesajul de sfarsit.

**if turns == 0:**

**hangman\_label.config(text=hangman\_stages[-1])**

**result\_label.config(text="Ai pierdut! Cuvântul era: " + word)**

**chances\_label.config(text="Mai ai 0 șanse.")**

**disable\_game()**

**return**

: daca numarul de incercari ajunge la 0, inseamna ca jucatorul a pierdut si se afiseaza figura completa si mesajul de infrangere.

**hangman\_label.config(text=hangman\_stages[len(hangman\_stages) - 1 - turns])** : se actualizeaza animatia jocului.

**if turns == 1:**

**chances\_label.config(text="Mai ai 1 șansă.")**

**else:**

**chances\_label.config(text=f"Mai ai {turns} șanse.")**

: se afiseaza numarul de incarcari ramase.

**def guess\_letter():**

**global guesses, turns**

**guess = guess\_entry.get().lower()**

: functia care este apelata atunci cand o litera este introdusa de catre jucator si apasa butonul „Ghiceste o litera”. Variabila „guess” preia litera introdusa si o converteste in litera mica.

**if len(guess) != 1 or not guess.isalpha():**

**result\_label.config(text="Te rog introdu o singură literă validă!")**

**return**

: se verifica daca caracterul introdus este valabil, adica daca acesta este un singur caracter si apartine alfabetului.

**if guess in guesses:**

**result\_label.config(text="Ai încercat deja această literă. Încearcă alta!")**

**return**

: se verifica daca litera a fost deja ghicita.

**guesses += guess** : se adauga litera ghicita in lista „guesses”.

**if guess not in word:**

**turns -= 1**

**result\_label.config(text=f"Greșit! Litera '{guess}' nu este în cuvânt.")**

**else:**

**result\_label.config(text=f"Ai ghicit corect! Litera '{guess}' este în cuvânt.")**

: daca litera nu se afla in cuvant, se decrementeaza variabila „turns” si se afiseaza mesajul de esec, daca aceasta este se afiseaza mesajul de succes.

**guess\_entry.delete(0, tk.END)** : functia care goleste campul de text.

**update\_game()** : functia care actualizeaza starea jocului.

**def disable\_game():**

**guess\_entry.config(state="disabled")**

**guess\_button.config(state="disabled")**

: funcita care dezactiveaza campul de text si butonul de ghicire, pentru a preveni introducerea de litere dupa ce jocul s-a incheiat.

**root = tk.Tk()** : se creeaza fereastra principala a jocului.

**root.title("Jocul Spânzurătoarea") :** se seteaza titlul jocului.

**word\_label = tk.Label(root, text="\_ " \* len(word), font=("Helvetica", 20))**

**word\_label.pack()** : se creeaza un label pentru a afisa cuvantul ales cu liniute.

**hangman\_label = tk.Label(root, text=hangman\_stages[0], font=("Courier", 12), justify="left")**

**hangman\_label.pack()** : se creeaza un label pentru a afisa animatia jocului.

**chances\_label = tk.Label(root, text=f"Mai ai {turns} șanse.", font=("Helvetica", 14))**

**chances\_label.pack()** : se creeaza un label pentru a afisa numarul de incercari ramase.

**guess\_entry = tk.Entry(root, font=("Helvetica", 14))**

**guess\_entry.pack()** : se creeaza un camp de text in care jucatorul poate introduce litera ghicita.

**guess\_button = tk.Button(root, text="Ghiceste o literă", font=("Helvetica", 14), command=guess\_letter)**

**guess\_button.pack()** : se creeaza un buton care, atunci cand este apasat, apeleaza functia „guess\_letter()”

**result\_label = tk.Label(root, text="", font=("Helvetica", 14))**

**result\_label.pack()** : creaza un label unde se afiseaza mesajele.

**update\_game()** : functia care actualizeaza starea jocului inainte de a incepe.

**root.mainloop()** : ruleaza bucla principala a aplicatiei grafice.

***Bibliografie:***

<https://www.youtube.com/watch?v=ag8NtD1e0Kc>

[inventwithpython.com/invent4thed/chapter8.html](https://inventwithpython.com/invent4thed/chapter8.html)

<https://www.geeksforgeeks.org/python-tkinter-tutorial/>

<https://www.youtube.com/watch?v=l4-wydm2Jkg>

***Link Github :***

[Hangman/joc.py at main · stefan-petcu/Hangman · GitHub](https://github.com/stefan-petcu/Hangman/blob/main/joc.py)