**Documentatie 2**

***Prerequisites si configurare:***

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

Se deschide fisierul hangman.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 hangman.py**.

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

***Explicarea codului hangman.py:***

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

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

**print("Hai să ne jucăm Spânzurătoarea! Succes!")** : afisam un mesaj de inceput.

**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.

**while turns > 0** :incepe bucla care se repeta atata timp cat jucatorul mai are incercari.

**failed = 0** : se initializeaza variabila „failed” cu 0. Aceasta reprezinta cate litere din cuvant nu au fost inca ghicite.

**for char in word:**

**if char in guesses:**

**print(char, end=" ")**

**else:**

**print("\_", end=" ")**

**failed += 1**

: se parcurge fiecare litera a cuvantului ales. Daca litera a fost ghicita se afiseaza. Daca nu, se afiseaza caracterul „\_” pentru a arata ca litera nu a fost ghicita inca si se incrementeaza variabila „failed”.

**print (”\n”)** : se adauga o linie goala.

**if failed == 0:**

**print("Felicitări, ai câștigat! Cuvântul era:", word)**

**break**

: se verifica daca variabila „failed” este 0. Daca aceasta este 0, inseamna ca jucatorul a castigat , se printeaza mesajul de final si se iese din bucla.

**print(hangman\_stages[len(hangman\_stages) - 1 - turns])** : se afiseaza starea curenta a jocului.

**guess = input("Ghiceste o literă: ").lower()** : se soliciteaza jucatorului sa ghiceasca o litera si se converteste intr-o litera mica.

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

**print("Te rog introdu o singură literă validă!")**

**continue**

: se verifica daca caracterul introdus de catre jucator este o singura litera si este o litera a alfabetului. Daca nu indeplineste aceste conditii, se afiseaza un mesaj de eroare si se solicita o alta intrare.

**if guess in guesses:**

**print("Ai încercat deja această literă. Încearcă alta!")**

**continue**

: se verifica daca litera ghicita a fost deja introdusa. Daca da, se afiseaza un mesaj de avertizare si se continua cu urmatoare iteratie a buclei.

**guesses += guess** : se adauga litera ghicita la lista de litere ghicite „guesses”.

**if guess not in word:**

**turns -= 1**

**print("Gresit! Litera nu este în cuvânt.")**

**print("Mai ai", turns, "șanse.")**

**if turns == 0:**

**print(hangman\_stages[-1])**

**print("Ai pierdut! Cuvântul era:", word)**

: daca litera ghicita nu se afla in cuvant, se decrementeaza numarul de incercari „turns” si se afiseaza un mesaj. Daca numarul de incercari este 0, inseamna ca jucatorul a pierdut, se afiseaza ultima etapa a jocului si se afiseaza mesajul de final, impreuna cu cuvantul ales.

***Bibliografie:***

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

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

***Link Github :***

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