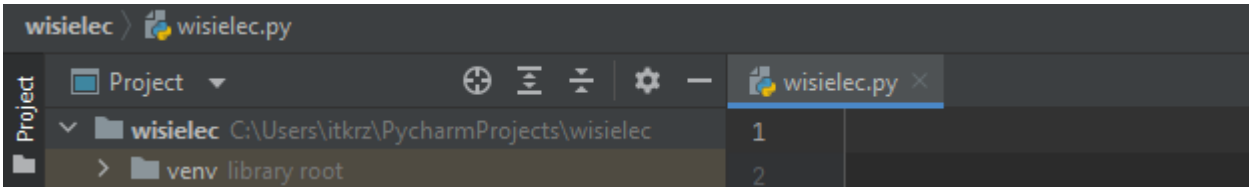


Python – gra Wisielec

Python umożliwia wykonywanie ciekawych projektów. Przykład gry w Wisielca:

Tworzymy nowy projekt i zmieniamy nazwę pliku głównego:



Zmienne początkowe:

```
3  # --- zmienne globalne programu
4  probyOdgadnienia = 5      # --- ilość prób odgadnięcia
5  word = "krzysztof"        # --- słowo do odgadnięcia
6  uzyteLitery = []          # --- tablica użytych liter
7  literyUzytkownika = []    # --- tablica podanych liter
```

Dodajemy główną pętlę programu:

```
# --- główna pętla programu
while True:
    litera = input('Podaj literę którą chcesz odgadnąć: ')
```

Dodajemy pętlę zamieniającą słowo do odgadnięcia na znaki podkreślenia:

```
8
9  # --- lista odgadywanego/uzupełnianego słowa
10 for i in word:
11     literyUzytkownika.append('_')
12
13 # --- główna pętla programu
14 while True:
15     litera = input('Podaj literę którą chcesz odgadnąć: ')
```

Dopisujemy w pętli litery podawane przez użytkownika:

```
14 while True:
15     litera = input('Podaj literę którą chcesz odgadnąć: ')
16     uzyteLitery.append(litera)
17     # --- kontrola tworzonej listy z użytymi literami
18     print(uzyteLitery)
```

Sprawdzamy, czy użyta litera znajduje się w słowie:

```
18     print(uzyteLitery)
19     ## --- sprawdzenie liter - metoda nie sprawdza się przy większej ilości takich samych liter
20     print(word.index(litera))
```

Własna funkcja znajdująca powtarzające się litery:

```
11 # 5. --- funkcja znajdująca indeksy duplikowanych liter
12 def znajdzDuplikatyLiter(word, litera):
13     indeksyLiter = []
14     # numerujemy litery
15     for i, literyWyrazu in enumerate(word):
16         if litera == literyWyrazu:
17             indeksyLiter.append(i)
18     return indeksyLiter # --- zwraca tablice
```

Wywołanie kontrolne powyższej funkcji:

```
wisielec.py x
23 while True:
24     litera = input('Podaj literę którą chcesz odgadnąć: ')
25     uzyteLiter.append(litera)
26     # --- kontrola tworzonej listy z użytymi literami
27     print(uzyteLiter)
28     ## --- sprawdzenie liter - metoda nie sprawdza się przy większej ilości takich samych liter
29     print(word.index(litera))
30     # --- kontrolne sprawdzenie indeksów danej liter
31     print(znajdzDuplikatyLiter(word, litera))
```

Sprawdzenie stanu gry:

```
22 # 3. --- główna pętla programu
23 while True:
24     litera = input('Podaj literę którą chcesz odgadnąć: ')
25     uzyteLiter.append(litera)
26     # --- kontrola tworzonej listy z użytymi literami
27     print(uzyteLiter)
28     ## --- sprawdzenie liter - metoda nie sprawdza się przy większej ilości takich samych liter
29     print(word.index(litera))
30     # --- kontrolne sprawdzenie indeksów danej liter
31     print(znajdzDuplikatyLiter(word, litera))
32     # --- znajdujemy indeksy i zamieniamy '-' na litery w odgadywanym słowie
33     znajdzIndeksyLiter = znajdzDuplikatyLiter(word, litera)
34     if len(znajdzIndeksyLiter) == 0:
35         print("Nie udało się. Nie ma takiej litery.")
36         probyOdgadniecia -= 1 # --- zmniejszamy ilość życ
37         # --- sprawdzenie, czy mamy jeszcze życia do dyspozycji
38         if probyOdgadniecia == 0:
39             print("Koniec gry :)")
```

Zakończenie gry gdy nie mamy już prób odgadnięcia jest możliwe dzięki zaimportowaniu modułu 'sys':

```
wisielec.py x
1 # 1. --- Wisielec gra terminalowa
2 # 6. --- importowane moduły
3 import sys
4
```

Oraz dopisujemy wyjście z gry:

```
40     if probyOdgadnienia == 0:
41         print("Koniec gry :)")
42         sys.exit(0)
```

Dodajemy informację o ilości pozostałych prób:

```
36     if len(znajdzIndeksyLiter) == 0:
37         print("Nie udało się. Nie ma takiej litery.")
38         probyOdgadnienia -= 1 # --- zmniejszamy ilość żyć
39         print('Pozostało prób:', probyOdgadnienia)
```

Sytuacja gdy odgadniemy słowo (else):

```
41     if probyOdgadnienia == 0:
42         print("Koniec gry :)")
43         sys.exit(0)
44     else:
45         # --- zamieniamy podkreślenia na odgadnięte litery
46         for i in znajdzIndeksyLiter:
47             literyUzytkownika[i] = litera
48         # --- sprawdzamy czy słowo zostało odgadnięte
49         if ''.join(literyUzytkownika) == word:
50             print("Brawo, to jest to słowo!")
51             sys.exit(0)
```

Podgląd odgadywanego słowa:

```
45         # --- zamieniamy podkreślenia na odgadnięte litery
46         for i in znajdzIndeksyLiter:
47             literyUzytkownika[i] = litera
48         print(literyUzytkownika) # wyświetlanie odgadniętych liter
49         # --- lub
50         print(''.join(literyUzytkownika))
```

Dodanie funkcji, której zadaniem będzie informowanie gracza o ilości pozostałych prób:

```
50         print(''.join(literyUzytkownika))
51         print('Pozostało prób:', probyOdgadnienia)
```

KONIEC

Zadanie:

zmodyfikować grę aby słowa były losowane oraz aby użytkownik zdecydował czy chce grać czy zakończyć grę

