



Jesenji semestar, 2022/23

PREDMET: CS324 - Skripting jezici

Projektni Zadatak

Igra rešavanja anagrama

Ime i prezime: **Aleksa Cekić**

Broj indeksa: **4173**

Datum izrade: **13.01.2023**

Sadržaj:

[Uvod](#)

[Struktura aplikacije](#)

[Tkinter](#)

[Prikaz aplikacije](#)

[Početna strana](#)

[Glavna strana](#)

[Funkcionalnosti](#)

[Čuvanje podataka u fajlu](#)

[Zaključak](#)

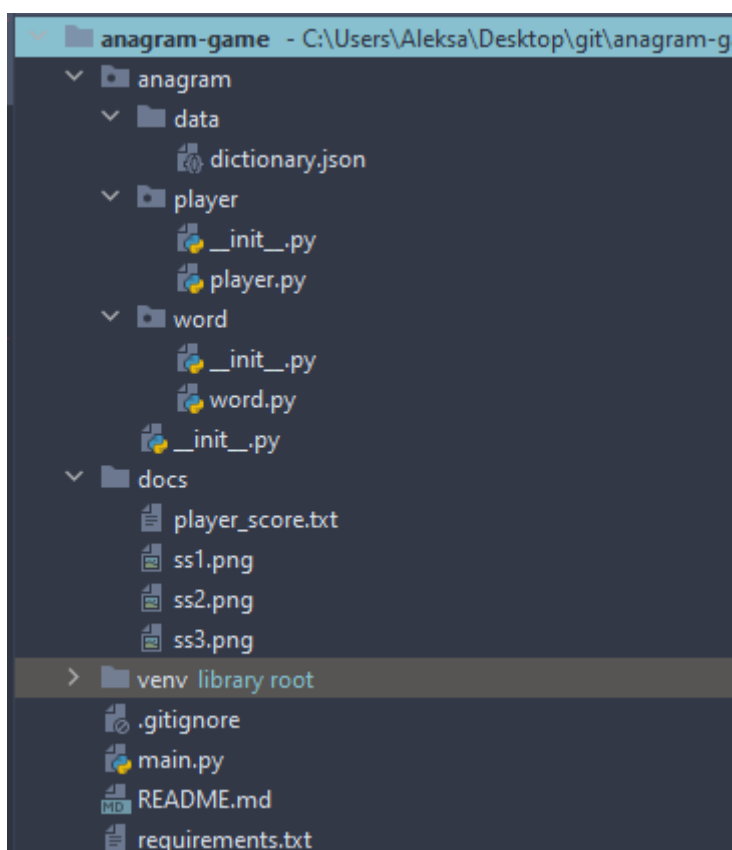
[Literatura](#)

Uvod

Tema ovog projektnog zadatka je Igra rešavanja anagrama. Igra je dobila inspiraciju od tv emisije "Slagalice". Aplikacija funkcioniše tako što korisnik na početku unosi svoj naziv. Nakon toga igra počinje i na ekranu se korisniku prikazuje reč koja je izmešana. Igrač treba da reši tu reč kucanjem u polje njeno pravo značenje. Ukoliko igrač pogodi koja je reč onda dobija poen, u slučaju da pogreši igraču se poen skida sa skora. Ukoliko Igrač ne vidi rešenje, postoji šansa prikazivanja hinta klikom na dugme za to. Prekidom igre, u fajlu se čuva igračev naziv i ukupan broj skora koji je imao.

Struktura aplikacije

Aplikacija je realizovana u python jeziku sa GUI-jem (Grafičkim Korisničkim Interfejsom) preko biblioteke tkinter. Za strukturu projekta korišćen je [Hitchhiker-ov guide](#).

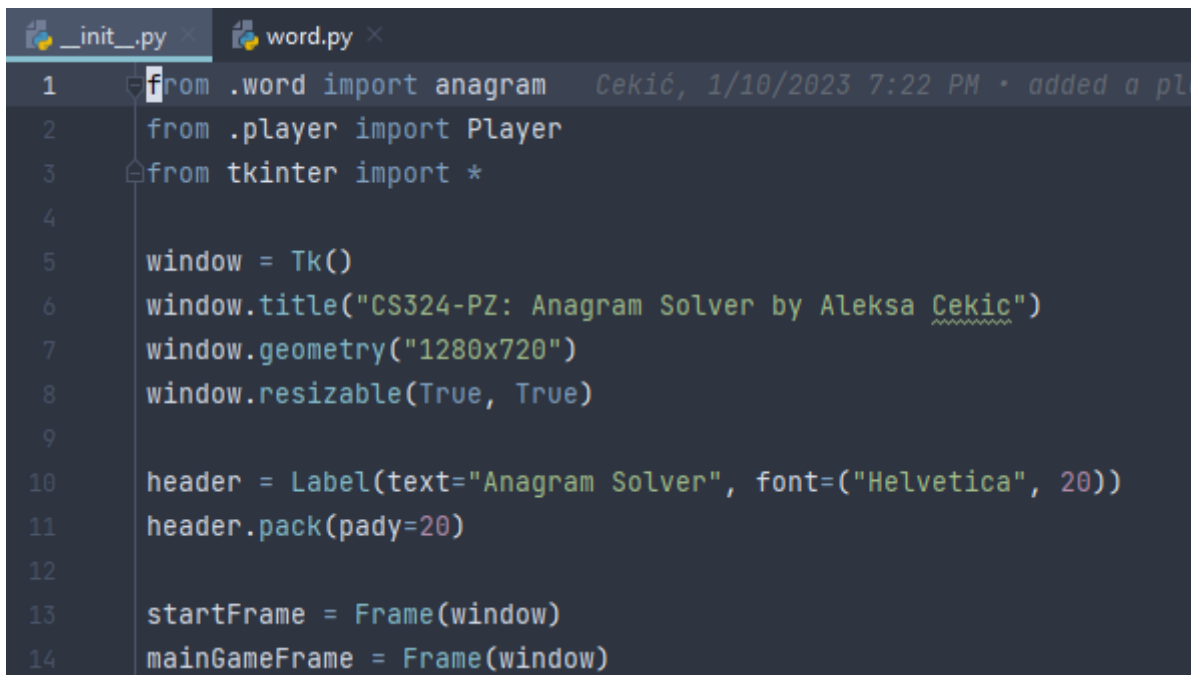


Tkinter

[Tkinter](#) je open-source grafički korisnički interfejs koji se koristi u Python jeziku. Ova biblioteka se dodaje u projekat instaliranjem preko pajtonovog PIP menadžera paketa. Koristi slojni stil prilikom dizajniranja i implementiran je u C jeziku. Prilikom importovanja tkinter, komponente na aplikaciju moguće je prikazati jedan preko drugog korišćenjem pack() metode, ili preko grid metode koja pozicionira komponente u stilu matrice sa parametrima reda i kolone. Postoje razne komponente koje se mogu dodati i stilizovati programerski a neke od njih su:

- Buttons (dugmići),
- Entries (Text polja),
- Labels (Nazivi),
- Dropdown menus (Meniji)
- Frames (Okviri), itd.

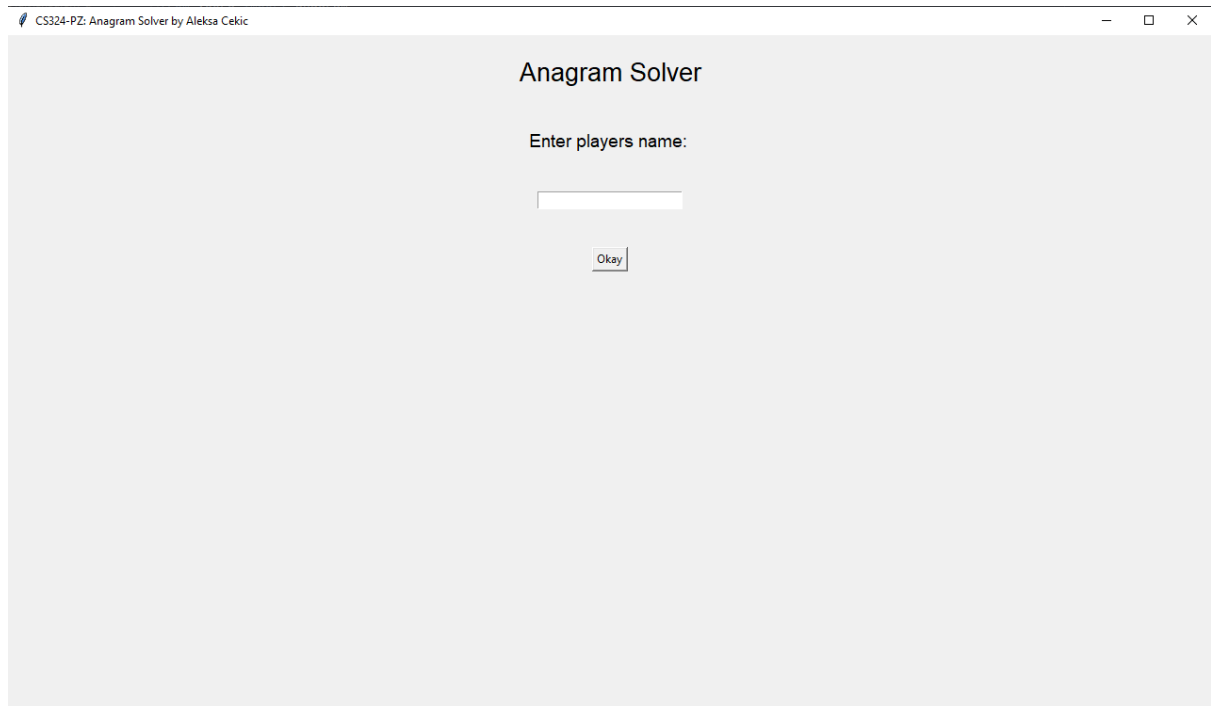
Biblioteku možete instalirati preko requirements.txt fajla. A da kreirate GUI koristite pozivom Tk() konstruktora.



```
1 from .word import anagram
2 from .player import Player
3 from tkinter import *
4
5 window = Tk()
6 window.title("CS324-PZ: Anagram Solver by Aleksa Cekic")
7 window.geometry("1280x720")
8 window.resizable(True, True)
9
10 header = Label(text="Anagram Solver", font=("Helvetica", 20))
11 header.pack(pady=20)
12
13 startFrame = Frame(window)
14 mainGameFrame = Frame(window)
```

Prikaz aplikacije

Početna strana



Početna strana je karakteristična po tome što se na njoj nalazi text polje gde korisnik unosi naziv igrača. Ukoliko igrač ne unese ništa u polje, igraču će se dodeliti osnovno ime "Player". Klikom na okej dugme se prikazuje novi okvir (Frame) na kome igrač rešava anagrame.

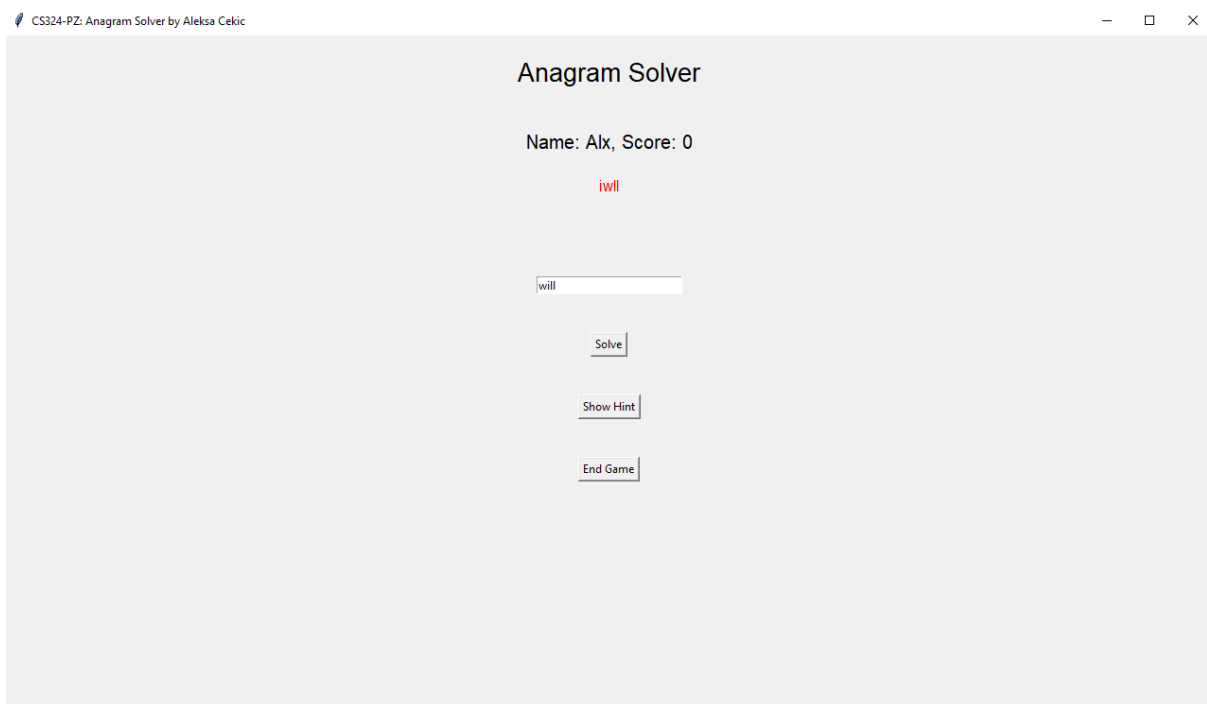
```

19 def go_to_game_frame():
20     mainGameFrame.pack(fill='both', expand=1)
21     startFrame.pack_forget()
22
23
24 def player_input(text, frame):
25     if text.get() != "":
26         p.set_name(text.get())
27         clear_frame(frame)
28         game_frame()
29
30
31 def start():
32     startFrame.pack()
33     label1 = Label(startFrame, text="Enter players name: ", font=("Helvetica", 14))
34     label1.pack(pady=20)
35
36     input_field = Entry(startFrame, width=25)
37     input_field.pack(pady=20)
38
39     btn = Button(startFrame, text="Okay", command=lambda: player_input(input_field, startFrame))
40     btn.pack(pady=20)

```

Promena stranice je realizovana tako što klikom na dugme, korisnik briše prethodni okvir, i dodaje novi okvir na prozoru.

Glavna strana



Korisniku se prikazuje strana na kojoj se odvija igra. Na vrhu posle naziva projekta se nalazi naziv korisnika i njegov score. Score će se povećavati za 1 svaki put kada odgovori tačno na odgovor, a smanjivaće se uvek kada ne odgovori ili ne odgovori tačno na odgovor.

CS324-PZ: Anagram Solver by Aleksa Cekic

— □ ×

Anagram Solver

Name: Alx, Score: 0

pdoo

A Russian weight, equal to forty Russian pounds or about thirty-six English pounds avoirdupois.

Solve

Show Hint

End Game

Ukoliko je igraču potrebna pomoć, on uvek može klikom na dugme “Show Hint” prikazati Opis te reči koju treba da reši.

Name: Alx, Score: 1

inad

Solve

Show Hint

End Game

Prikaz kada korisnik dobije poen

Anagram Solver

Name: Alx, Score: 0

lawk

Solve

Show Hint

End Game

Prikaz u slučaju gubitka poena (score se vratio na 0)

Funkcionalnosti

```
def pick_word(data, length):
    words = list(data.keys())
    while True:
        word = random.choice(words)
        if length > len(word) > 2:
            return word

Aleksa Cekić

def shuffle(word):
    characters = list(word)
    shuffled = word
    while True:
        random.shuffle(characters)
        shuffled = ''.join(characters)
        if shuffled != word:
            return shuffled

Aleksa Cekić

def anagram(length=5):
    data = json.load(open('anagram/data/dictionary.json'))

    word = pick_word(data, length)
    question = shuffle(word)
    hint = data[word]

    question = question.lower()

    return Anagram(question, hint, word)
```

Aplikacija funkcioniše tako što se iz preuzetog dictionera učitavaju reči dužine 5. Nakon čega se redovi slova mešaju i na kraju pakuju sa hintom i reč sa normalnim karakterima (koji nisu izmešani) u klasi Anagram odakle se prikazuju korisniku na ekranu.

```

# Aleksa Cekic
def game_frame():
    go_to_game_frame()

    player_name = Label(mainGameFrame, text="Name: {}, Score: {}".format(p.get_name(), p.get_score()),
                        font=("Helvetica", 15))
    player_name.pack(pady=20)

    a = anagram(5)

    question_label = Label(mainGameFrame, text="".join(a.get_question()), font=("Helvetica", 12), fg="Red")
    question_label.pack()

    hint = Label(mainGameFrame, text="", font=("Helvetica", 12), fg="Blue")
    hint.pack(pady=20)

    inpt = Entry(mainGameFrame, width=25)
    inpt.pack(pady=20)

    btn_solve = Button(mainGameFrame, text="Solve", command=lambda: solve(a.get_answer(), inpt.get(), mainGameFrame))
    btn_solve.pack(pady=20)

    btn_show = Button(mainGameFrame, text="Show Hint", command=lambda: show(hint, a.get_hint()))
    btn_show.pack(pady=20)

    btn_show = Button(mainGameFrame, text="End Game", command=lambda: save_stats(p, window))
    btn_show.pack(pady=20)

```

Prikazivanje podataka preko tkintera.

```

def solve(answer, inpt, frame):
    if answer == inpt.lower():
        count = p.get_score() + 1
        p.set_score(count)
        clear_frame(frame)
        game_frame()
    else:
        if p.get_score() > 0:
            count = p.get_score() - 1
            p.set_score(count)
            clear_frame(frame)
            game_frame()

```

Logika rešavanja anagrama

```

Aleksa Cekić
class Player:
    Aleksa Cekić
    def __init__(self, player_name="Player", score=0) -> None:
        self._player_name = player_name
        self._score = score

    Aleksa Cekić
    def set_name(self, name):
        self._player_name = name

    Aleksa Cekić
    def set_score(self, score):
        self._score = score

    Aleksa Cekić
    def get_name(self):
        return self._player_name

    Aleksa Cekić
    def get_score(self):
        return self._score

    Aleksa Cekić
    def __repr__(self):
        return "Player: {}, has a score of {} points!".format(self.get_name(), self.get_score())

```

Klasa Igrača

Čuvanje podataka u fajlu

```

Player: aleksa, has a score of 0 points!
Player: asd, has a score of 0 points!
Player: Test1, has a score of 3 points!
Player: Alex, has a score of 1 points!
Player: a, has a score of 1 points!
Player: Test2, has a score of 1 points!
Player: Alx, has a score of 3 points!

```

Klikom na dugme "End Game" igrač gasi aplikaciju i čuva naziv i maksimalni scor koji je dobio u fajlu.

```
👤 Aleksa Cekić  
def save_stats(plyr, win):  
    # saves to file  
    file = open("docs/player_score.txt", "a")  
    file.write("\n" + plyr.__repr__())  
    file.close()  
    win.destroy()
```

Nakon čuvanja podataka u fajl poziva se metoda `destroy()` koja zatvara prozor i gasi aplikaciju.

Zaključak

Ovaj projekat je osmišljen tako da igračima daje zadatak da reše anagram koji se slučajno izvlači i meša iz dictionerija koji je preuzet sa ovog [linka](#). Ceo projekat možete preuzeti sa mog GitHub [repozitorijuma](#).

Tkinter biblioteka je dosta drugačija od ostalih biblioteka za prikaz podataka preko Grafičkog Korisničkog Interfejsa sa kojim sam se susreo do sada. Neke funkcionalnosti koje sam prvobitno smislio sam morao drugačije da realizujem iz razloga zato što Tkinter ne podržava korišćenje time funkcionalnosti. Funkcioniše tako da zamrzne ekran na vreme koje se dodaje pozivom `sleep()` metode.

Realizacija ovakvog projekta je omogućena stečenim znanjem na ovom predmetu.

Literatura

Login - LAMS :: Learning activity management system. (n.d.).

Metropolitan.Ac.Rs:8080. Retrieved January 13, 2023, from

<http://lams.metropolitan.ac.rs:8080/lams/index.do>

Stack Overflow - where developers learn, share, & build careers. (n.d.). Stack

Overflow. Retrieved January 13, 2023, from <https://stackoverflow.com/>

Python, R. (2022, March 30). *Python GUI programming with Tkinter.*

Realpython.com; Real Python. <https://realpython.com/python-gui-tkinter/>

Structuring your project — the hitchhiker's guide to python. (n.d.). Python-guide.org.

Retrieved January 13, 2023, from <https://docs.python-guide.org/writing/structure/>