

Dokumentacija projekta:

Pasijans

Autor: Domagoj Margeta, 4.5

Mentor: prof. Goran Boneta

Sadržaj:

1. UVOD	3
1.1 Pasijans	
1.2 O autoru	
1.3 Ideja za projekt	
2. DETALJNI OPIS RADA	
2.1 Menu	
2.2 Play	4
2.3 Postavljanje polja	5
2.4 Naredba nepreokrenutih karata u špil	6
2.5 Ostale naredbe	7
2.6 Obavijest o pobijedi	8
3. TEHNIČKE INFORMACIJE	9

1. Uvod

1.1 Pasijans:

Jedna od najprepoznatljivijih kartaških igra Pasijans (*patience*) ili Solitaire: Klondike postala je popularna u 19. st. tijekom zlatne groznice. Igra se s klasičnim špilom od 52 karte, njih 28 je poredano je u 7 redova tako da prvi red ima jednu kartu te svaki sljedeći ima jednu više od prijašnjeg, a ostale karte idu u špil te se onda vrte (detaljna pravila igra se nalaze u txt dokumentu koji je priložen u sklopu projekta). Pobjedili ste ako svaku od četiri vrste karata posložite od asa do kralja. Zapamtite mogućnost pobjede će odlučiti raspored karata koji dobijete.

1.2 O autoru:

Autor je Domagoj Margeta učenik završnog razreda Gimnazije Andrije Mohorovičić Rijeka te osoba koja sama dobiva veliko zadovoljstvo igrajući ovu igru te je na nju potrošila iznimno veliki broj sati.

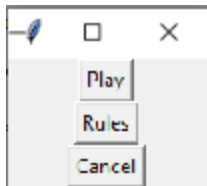
1.3 Ideja za projekt:

Kao što je već napisano ova igra autoru donosi veliko zadovoljstvo te je htio napraviti svoj program koji će korisniku lako omogućiti igranje Pasijansa bez postavljanja karata i upavljanje igrom jedino klikom miša. Osim toga ova verzija igra omogućava potpunu nasumičnost pri postavljanju karata, što je pri ručnom postavljanju teško izvedivo.

2. Detaljni opis rada

2.1 Menu:

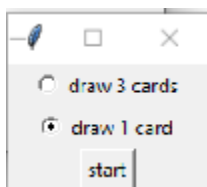
Ovo je glavni prozorčić u programu (Tk), svi ostali će biti objekta Toplevel. Ovaj mali prozorčić služi samo kao formalnost, u njemu ćete naći tri gumba. Prvi gumb *Play* će vas odvesti do novog istoimenog prozorčića *Play*, čija će uloga biti objašnjena kasnije. Gumb *Rules* će otvoriti txt dokument s detaljnim pravilima o igri, te posljednje gumb *Cancel* će zatvoriti glavni prozorčić.



```
class Menu(Tk, object):
    def __init__(self):
        self.root=Tk()
        self.root.title("Menu")
        self.frame=Frame(self.root)
        self.frame.pack()
        self.play=Button(self.frame, text="Play", command=self.play_choice)
        self.play.pack()
        self.rules=Button(self.frame, text="Rules", command=self.rules_book)
        self.rules.pack()
        self.cancel=Button(self.frame, text="Cancel", command=self.root.destroy)
        self.cancel.pack()
    def rules_book(self):
        #txt file s pravilima igre
        webbrowser.open("rules.txt")
    def play_choice(self):
        #novi prozor s odabirom načina igre
        play=Play()
```

2.2 Play:

Ovaj prozorčić, koji će se otvoriti nakon što stisnete *Play* u prijašnjem prozorčiću, ima funkciju odabira opcije igre. Odnosno postoji i način igre Pasijansa u kojem se umjesto jedne karte otvaraju tri iz špila te vam ovaj prozorčić omogućuje da odaberete tu opciju ako vam je to želja. Pošto samo jedna od dvije opcije može biti izabrana prozorčić sadrži dva *Radiobutton* te gumb *Start* koji otvara prozorčić za igru. Nakon što se prozorčić za igru pokrene ovaj prozorčić se briše.



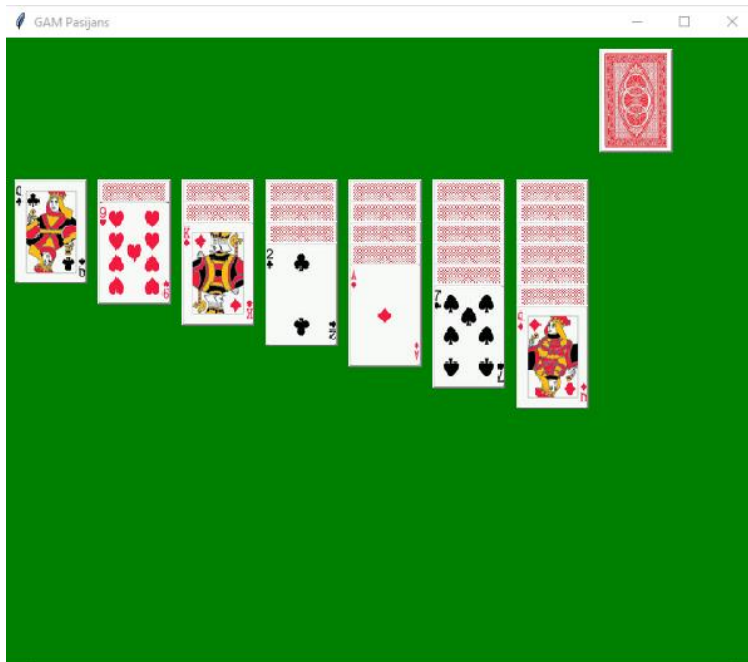
```
class Play(Toplevel, object):
    def __init__(self):
        self.root=Toplevel()
        self.root.title("Play")
        self.cards=IntVar(self.root, value=1)
        self.frame=Frame(self.root)
        self.frame.pack()
        #3 karte
        self.cards3=Radiobutton(self.frame, text="draw 3 cards", variable=self.cards, value=3)
        self.cards3.pack()
        #1 karta
        self.cards1=Radiobutton(self.frame, text="draw 1 card", variable=self.cards, value=1)
        self.cards1.pack()
        #gumb za početak igre
        self.start=Button(self.frame, text="start", command=self.start)
        self.start.pack()
    def start(self):
        #novi prozor u kojem je igra
        game=Game(self.cards, self.root)
```

2.3 Postavljanje polja:

Ovaj dio programa se zbiva u prozorčiću *GAM Pasijans* objekta *Game* pod `__init__`. Polje je klasične zelene boje, a polja se sastoji od tri glavne matrice: *self.deck*, *self.sets* i *self.rows*. Prvo što se odvija u programu je stvaranje liste *self.list* koja sadrži imena svih karti tako da prvo slovo označava boju karte, drugo vrstu karte i zadnja dva broja broj karte od 01 (as) do 13 (kralj). Uz to u projekt su

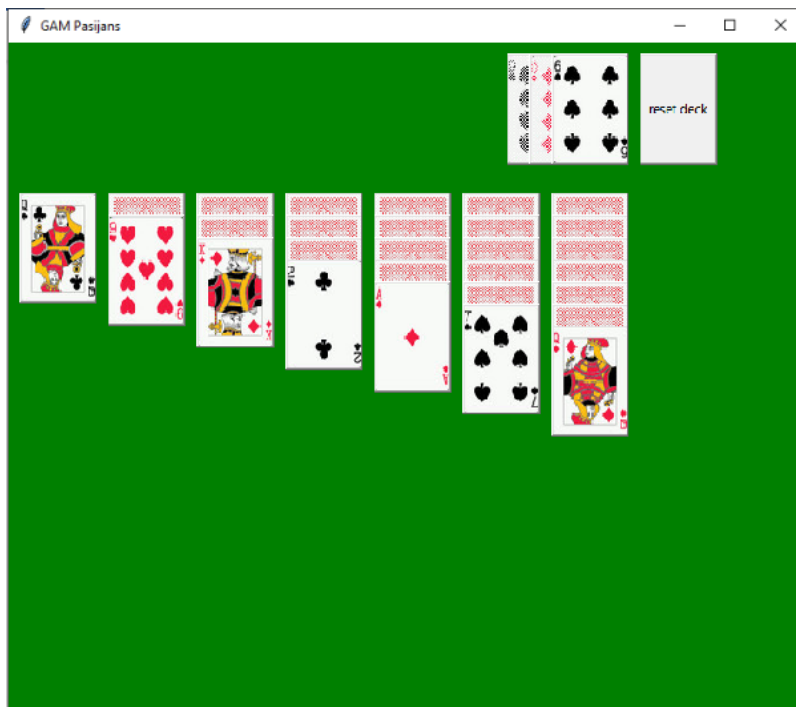
uklopljene slike svake karte s imenima koja odgovaraju imenima karata u programu te i slika poledine karte. Prva petlja stavlja karte (koje su zapravo gumbi sa slikom odgovarajuće karte) u matricu `self.rows` u listu koja odgovara redu u kojem je karta. Sve karte osim zadnje će biti isključene i imati sliku poledine. Ostale karte će biti stavljene u prvu listu matrice `self.deck` u koju je prije toga stavljen gumb za izvrtiti špil i koja služi za spremanje nepreokrenutih karata. Sve karte su nasumično poredane i na polju je uvijek jedna kopija svake karte. Pošto karte u svakoj matrici (redu, špil i setu) imaju posebne naredbe svaka od tih naredba je definirana u sljedećem djelu koda. Karte mijenjaju svoju naredbu kako prelaze iz jedne matrice u drugu.

```
class Game(Toplevel):
    def __init__(self, cards, play):
        play.destroy()
        self.master=Toplevel()
        self.cards=cards
        self.master.title("GDM Pasijansa")
        self.frame=Frame(self.master, width=720, height=600, bg="green")
        self.frame.pack()
        #lista svih karata: prvo slovo označava boju, drugo slovo označava grupu, a broj broj karte (asa do kralja)
        self.list=["hs01", "hs02", "hs03", "hs04", "hs05", "hs06", "hs07", "hs08", "hs09", "hs10", ".11", "hs12", "hs13", \
                  "rh01", "rh02", "rh03", "rh04", "rh05", "rh06", "rh07", "rh08", "rh09", "rh10", ".11", "rh12", "rh13", \
                  "rd01", "rd02", "rd03", "rd04", "rd05", "rd06", "rd07", "rd08", "rd09", "rd10", ".11", "rd12", "rd13", \
                  "bc01", "bc02", "bc03", "bc04", "bc05", "bc06", "bc07", "bc08", "bc09", "bc10", ".11", "bc12", "bc13"]
        #slike karte te okrenuta na poledinu
        self.cardb=Image.open("cardb.png")
        self.cardb=self.cardb.resize((70,100), Image.ANTIALIAS)
        # stvaranje podloge s mjerama i točno pozicioniranim kartama
        self.rows=[]
        for j in range(7):
            y=115
            for i in range(0, 7):
                b=7-i
                y+=20
                x=190
                for j in range(0, b):
                    k=random.choice(self.list)
                    self.list.remove(k)
                    #ime/tekst karte je povezan s nazivom slike te iste karte
                    img=Image.open(k[0:11]+".png")
                    image=ImageTk.PhotoImage(img, master=self.frame)
                    c=Button(self.frame, text=k, image=image, command=lambda l=k : self.rowscards[l])
                    c.image=image
                    c.place(x=x, y=y, height=100, width=70)
                    self.rows[j].append(c)
                    x+=80
            #isključivanje svih karata koje nisu na zadnjem mjestu
            for i in range(0, 7):
                for j in range( (len(self.rows[i]))-1):
                    if self.rows[i][j]!=self.rows[i][j-1]:
                        image=ImageTk.PhotoImage(self.cardb, master=self.frame)
                        self.rows[i][j].configure(state=DISABLED, image=image)
                        self.rows[i][j].image=image
        #4 sera za slaganje karata
        self.seta=[[] for i in range(4)]
        #špil s okrenutim i neokrenutim kartama
        self.deck=[[], []]
        #stvaranje špila
        self.re=place(x=570, y=10, height=100, width=70)
        self.deck[0].append(self.re)
        #povezivanje špila
        for i in range( (len(self.list))):
            k=random.choice(self.list)
            self.list.remove(k)
            image=ImageTk.PhotoImage(self.cardb, master=self.frame)
            c=Button(self.frame, text=k, image=image)
            c.image=image
            c.place(x=570, y=10, height=100, width=70)
            c.lift()
            self.deck[0].append(c)
            c.configure(command=lambda : self.deckn())
```



2.4 Naredba nepreokrenutih karata u špil:

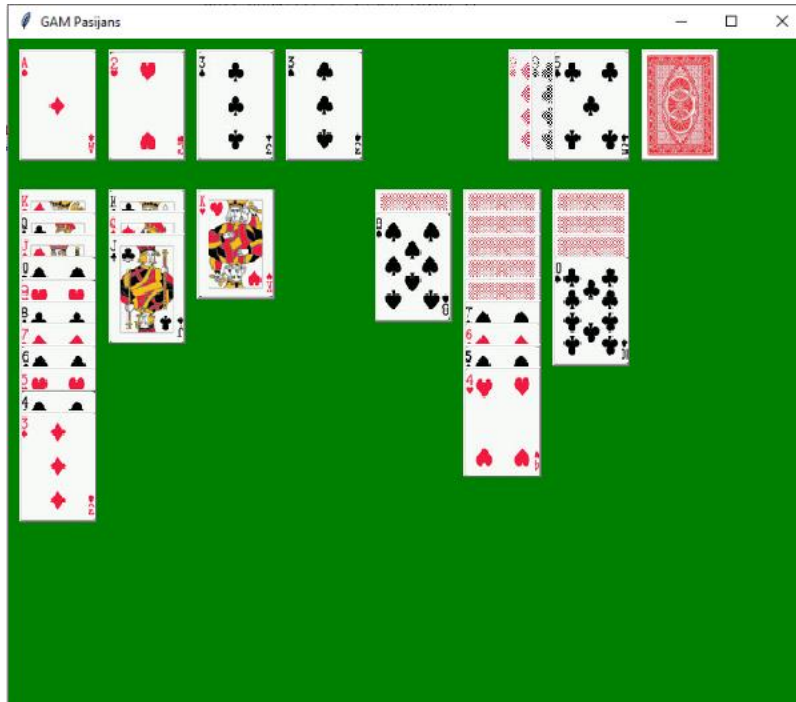
Naredba nepreokrenutih karata u špil (karte u prvoj listi matrice `self.deck`) je jednostavna. Kako kliknete na špil jedna ili tri karte će biti okrenute, dobit će svoju sliku i pomaknuti se lijevo od ostatka špila. Kako se špil okreće dvije prethodne karte koje su bile preokrenute (ako ih ima) će još uvijek biti vidljive tako da će njihov lijevi rub biti vidljiv no one će biti isključene. Kada je cijeli špil prokrenut na mjestu njega će biti gumb za preokrenut karte koji će onda sve preokrenute karte posložiti natrag u špil s istim redoslijedom kao na početku.



2.5 Ostale naredbe:

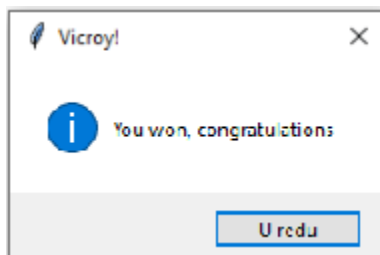
Ostale naredbe za karte su: prokrenute karte iz špila mogu ići u set ili u jedan od redova, karte iz redova mogu ići u set ili u jedan od drugih redova s time da se karte između redova mogu premiještati i ako one nisu zadnje u redu već karta koja paše na drugu premiješta i sve ostale karte ispod nje, a karte u setu mogu ići jedino u redove. S time da su karte as i kralj poseban slučaj, as ako je u špilu ili redu ima naredu da pronađe prvi prazan set (listu iz self.sets) i onda se premiješta na

njega, a kralj ako prelazi u jedan od redova mora ići na onaj koji je prazan (lista iz `self.rows`) uz to ako je kralj smješten u set pokreće se provjera je li igra gotova (kada su svi kraljevi smješteni u setove). Ako se desi slučaj da karta može preći u red i u set njezin je prioritet ići u set (jer je bit igre postaviti sve karte u set), no ako se onda ona opet klikne kada je u setu onda će ona preći u red. Ako karta iz reda nije preokrenuta ona se prokreće kad postane zadnja u redu. Karte u setu se slažu po vrsti od asa o kralja, a u redovima se karte slažu na druge koje su za jedan broj veće s time da karte moraju biti raličite boje. Kada karta dobije novu naredbu naredba prima njezin tekst (koji se nevidi zbog slke te služi kao identifikacija karte) te ju pomoću njega locira i pravilno upravlja. Ako se klikne karta koja ne može nigdje preći ništa se neće dogoditi.



2.6 Obavijest o pobjedi:

Kada složite sve kraljeve u setove pojaviti će se prozorčić s obavješću objekta *Win* da ste pobijedili nakon što kliknete *U redu* na tom prozorčiću program je završen.



```
class Win (Tk):
    def __init__(self):
        #poruka da ste pobijedili, program se prekida
        win=showinfo("Vicroy!", "You won, congratulations")
        exit()
```


3. Tehničke informacije

Ovaj program je napravljen i namijenjen za Python 3.8.1. Glavni modul koji preuzima je tkinte, modul za stvaranje programa s korisničkim sučeljem u Pythonu. Uz to koristi module random, webbrowser, PIL (modul za manipuliranje slika u tkinteru) i iz tkinter.messagebox uzima objekt showinfo za zadnji prozorčić o pobjedi.