

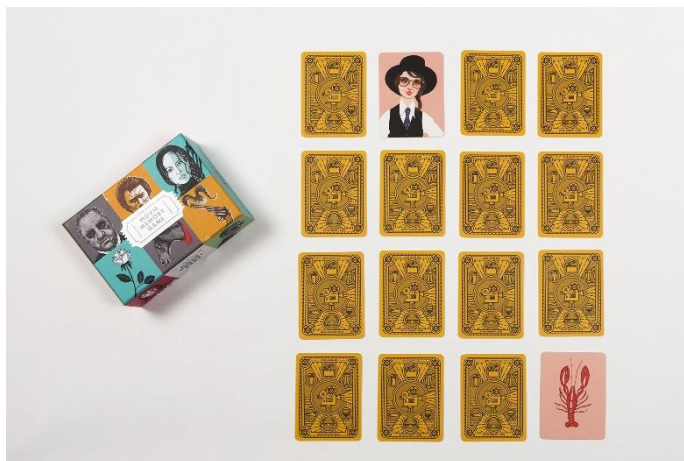
# Tehnicka dokumentacija: Memory

## 1. Uvod

---

### 1.1 . Klasična igra Memory-a

U današnje vrijeme više manje svi znaju za igru memory te kako je igrati. Ova svezremena igra se može igrati samostalno ili s više ljudi. Igrač (ili igrači) okreće dvije kartice (koje imaju prednju i stražnju stranu, sa stražnjom stranom, koja na svim karticama izgleda isto, okrenutom



prema gore, a prednjom stranom s posebnom slikom okrenutom prema dolje da je igrač/igrači ne vide) po rundi, pokušavajući u istoj rundi okrenuti dvije kartice s istom sličicom te ako to nije učinio upamtiti di je koja sličica. Ako kartice koje je igrač okrenuo imaju drugačiju sličicu s prednje strane tada igrač okreće kartice nazad kako su bile (stražnjom stranom prema gore), a ako su sličice iste, kartice ostaju okrenute prema gore. Kada igru igra više ljudi oni jedan po jedan okreću kartice te kada netko okrene dvije kartice sa istim sličicama ta osoba to sebi pribroji kao bod. Igra je zabava za sve od djece do staraca te do danas ima mnogo različitih verzija.

## 1.2. Ideja projekta



Ideja je bila napraviti jednostavni projekt koji uključuje vrijeme, buttone i slike, tako da učim raditi nešto što me interesira i inače, van projekta. Najduže je uzeo proces učenja, ali to je upravo ono što je ovdje bio najvažniji dio. Ovo je jednostavni mali memory za jednog igrača.

## 2. Detaljan opis rada

---

### 2.1. Određivanje svojstva ekrana

Za imati odvojeni ekran sa slikama u pythonu trebalo je programirati svojstva toga novog programa, kao što je veličina i boja ekrana. U prvoj vidljivoj liniji koda u slici ispod smo definirali programu što je bijela boja pomoću njenih kordinata u RGB sustavu boja. Zatim smo odredili širinu(x) i visinu(y) te nakon toga smo definirali ekran kao dis, tu kraticu sam uzela od riječi display. Te u zadnjem redu koda kojeg se vidi na fotografiji smo obojali taj definirani screen u bijelo.

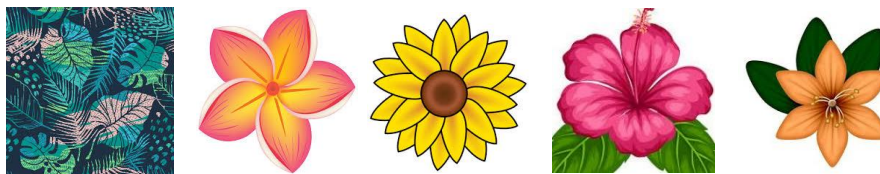
```
white=(250,250,250) #određivanje kordinata bjele boje u rgb-u  
#određivanje velicine ekrana  
X = 1140  
Y = 540  
dis=pygame.display.set_mode((X, Y ))  
dis.fill(white) #umetanje boje pozadine
```

### 2.2. Umetanje i mješanje slika

Prvo sam u programu dodala sliku za stražnju stranu kartice, koju sam označila imenom red jer je prvo to trebala biti samo crvena slika, ali sam kasnije odlučila da ću ipak umjesto toga staviti uzorak(1. sličica ispod koda).

Nakon toga sam umetnula sličice za prednju stranu kartica i dala im imena. Idući korak bio je rađanje liste(l) koja je imala po dva puta svaku od sličica s prednje strane kartice jer nam trebaju parovi kartica. Onda sam napravila for petlju koja tih osam sličica stavlja u random redosljedu u novu listu (li) iz koje cu kasnije uzimati jednu po jednu slicicu i uvrštavati ih na kordinate.

```
#umetanje slike za karticu za pokrivanje
red=pygame.image.load('red.jpg')
#umetanje slika
image0 = pygame.image.load('flower.jpg')
image1 = pygame.image.load('sun.jpg')
image2 = pygame.image.load('flower2.jpg')
image3 = pygame.image.load('flower3.jpg')
#random mjesanje slika
l=[image0, image1, image2, image3, image0, image1, image2, image3]
li=[]
for i in range(8):
    li.append(random.choice(l))
    l.remove(li[i])
```



## 2.3. Glavna funkcija programa

Da bi glavna funkcija koda radila, prvo smo trebali definirati par stvari. Prvo moramo odrediti listu u koju ćemo zapisati 'True' za sve stražnje kartice koje igrač trenutno vidi. Ja sam program napravila tako da se stražnja strana kartice miče tako da je to bilo važno

```
show=[True,True,True,True,True,True,True,True] #Pokazuju li se kartice
disabled=False #moze li se kartica okrenuti
open_card=False #je li kartica otvorena
forever_yeet=[True,True,True,True,True,True,True,True] #Hoce li kartica ostati otvorena
cardl=0 #broj kartice koju smo prvu okrenuli ce doci ovdje

def otvori(broj_kartice): #glavni program za otvanje kartice te provjeravanje jesu li kartice iste
    global open_card, disabled, begtime, cardl
    if open_card:
        open_card=False #oznacavanje da je kartica otvorena
        if li[broj_kartice]==li[cardl]: #provjeravanje jesu li kartice iste
            #micanje gornjih kartice
            forever_yeet[broj_kartice]=False
            forever_yeet[cardl]=False
        else:
            disabled=True
            begtime=time.time()
    else:
        open_card=True
        cardl=broj_kartice
        show[broj_kartice]=False
```

definirati. Nakon toga sam definirala da ako disabled govori za karticu da može biti maknuta (jer ne mogu biti više od dvije kartice okrenute u jednoj rundi), onda sam definirala da kartice po defaultu nisu otvorene. Zatim sam napravila listu koju sam nazvala forever\_yeet koja označuje da nijedna stražnja kartica još nije trajno bačena. card1 sam definirala samo da bi tu kasnije mogli uvrstiti prvu okrenutu karticu kako bi kasnije provjerili je li ista s drugom okrenutom karticom. Kad sam sve to definirala, krenula sam s izrađivanjem glavne funkcije koju sam nazvala 'otvori', pomoću if funkcije provjeravam imamo li već jednu okrenutu karticu te ako imamo drugim if-om provjeravamo jesu li sličice ispod maknutih kartica iste, ako jesu onda ih program zauvijek miče, a ako ne onda se počinje odbrojavati vrijeme (kojeg ću se uskoro pobliže dotaknuti) te nakon što zadano vrijeme prođe kartice vraćaju na njihovu prijašnju poziciju. Ako je odgovor na prvi if ne i ovo je prva kartica koju okrećemo onda tu karticu označujemo kao prvu okrenutu karticu kako bi njenu sliku usporedili s karticom koju ćemo okrenuti drugu tako da provjerimo jesu li te dvije kartice iste.

## 2.4. Uvođenje funkcije buttona i određivanje njegovih pozicija i slika

Sliku za stražnji dio kartice sam odmah umetnula pod button te sam mu odredila poziciju i veličinu i koja se funkcija izvršava ako je gumb stisnut. Funkciju s gumbom sam nazvala fun. Ispod toga sam definirala funkciju koja umeće sve slike prednjih stranica kartica te ih pokazuje na određenoj poziciji na prozoru od igrice. Funkcija button je iz library-a Engine kojeg je napravio Leonardo Šimunović.

```
#definiranje pozicije buttona i sto radi
def fun():
    Engine.button(dis, 10, 10, red.get_width(), red.get_height(),otvori,0,'',disabled, red)
def fun1():
    Engine.button(dis, 10, 300,red.get_width(), red.get_height(),otvori,1,'',disabled, red)
def fun2():
    Engine.button(dis,300, 10,red.get_width(), red.get_height(),otvori,2,'',disabled, red)
def fun3():
    Engine.button(dis,600, 10,red.get_width(), red.get_height(),otvori,3,'',disabled, red)
def fun4():
    Engine.button(dis,300, 300,red.get_width(), red.get_height(),otvori,4,'',disabled, red)
def fun5():
    Engine.button(dis,600, 300,red.get_width(), red.get_height(),otvori,5,'',disabled, red)
def fun6():
    Engine.button(dis,900, 10,red.get_width(), red.get_height(),otvori,6,'',disabled, red)
def fun7():
    Engine.button(dis,900, 300,red.get_width(), red.get_height(),otvori,7,'',disabled, red)

def slike_display(): #postavljanje izmjesanih slika na pozicije
    dis.blit(li[0], (10, 10))
    dis.blit(li[1], (10, 300))
    dis.blit(li[2], (300, 10))
    dis.blit(li[3], (600, 10))
    dis.blit(li[4], (300, 300))
    dis.blit(li[5], (600, 300))
    dis.blit(li[6], (900, 10))
    dis.blit(li[7], (900, 300))
```

## 2.5. Pokazuje li se stražnji dio kartice?

Ovaj dio programa (na slici ispod) provjerava je li stražnja kartica zauvijek maknuta i je li trenutčno maknuta kako bi program znao bi li trebao prikazivati gumb ili ne.

```
def kartice(): #umetanje kartica i provjeravanje jesu li upaljene
    global show
    if show[0] and forever_yeet[0]:#svaka kartica provjerava je li vec otvorena ili bacena
        fun()
    if show[1] and forever_yeet[1]:
        fun1()
    if show[2] and forever_yeet[2]:
        fun2()
    if show[3] and forever_yeet[3]:
        fun3()
    if show[4] and forever_yeet[4]:
        fun4()
    if show[5] and forever_yeet[5]:
        fun5()
    if show[6] and forever_yeet[6]:
        fun6()
    if show[7] and forever_yeet[7]:
        fun7()
```

## 2.6. Muzika

Zadnja stvar koju sam htjela dodati svojem programu bila je muzika, tako da sam nasla na YouTube-u pjesmu te sam istražila kako bi dodala glazbu u program. Pustila sam da se pjesma pušta samo jednom te napravila funkciju music da dio s muzikom bude što lakše ubaciti.

```
def music(): #dodavanje glazbe jer zasto ne, da bude bar malo manje basic
    pygame.mixer.music.load('music_just_for_fun.mp3')#ne znam zasto bas ov
    pygame.mixer.music.play()#glazba se pusta samo jednom ali ima 15 minut

#programa

music() #pustanje muzike da ljudi ne umru od dosade
```

## 2.7. Spajanje svih funkcija te više o vremenu

Dok program radi ako su kartice otvorene on mjeri jednu sekundu od kada je druga kartica otvorena prije nego sta opet pokrije karticu sa slicicom straznje strane kartice. Ako smo sličicu stražnje strane kartice preko nekog para zauvijek makli onda se ne čeka ta sekunda. Nakon toga smo samo ubacili sve prije objašnjene dijelove programa te dodali mogućnost izlaska pritiskom na crveni gumb x.

I to je kraj koda.

```
while True:
    if begtime+1<time.time() and not open_card: #kartice ako nisu zauvijek yeet
        disabled=False #opet mozes otvoriti karticu yay!!!
        show=[True,True,True,True,True,True,True,True]
        dis.fill(white) #bjela pozadina
        #postavljanje slika na pozicije
        slike_display()
        #postavljanje druge strane 'kartice' preko slika tako da ih igrac ne vidi
        kartice()
        pygame.display.update()
    for event in pygame.event.get() :
        if event.type==pygame.QUIT: #izlazak iz prozora igre kada stisnemo crv
            pygame.display.quit()
            sys.exit()
```



### 3. Tehničke informacije

---

#### 3.1. Korišteni library-i

Koristila sam library-e Engine, Pygame, Time, Sys i Random. Engine je library kojeg je napravio Leonardo Šimunović i u sebi sadrži button, u njemu je Leonardo koristio Pygame. Pygame je library s mnoštvom korisnih opcija pri izradi igara, kao što su opcije za dodavanje glazbe, slika, prozora za igru, mogućnosti dodavanja funkcija kada nešto stisnemo, library su napravili Lenard Lindstrom, René Dudfield, Pete Shinnars, Nicholas Dudfield, Thomas Kluyver i drugi (kako wikipedija kaže). Za library-e Time, Sys i Random nisam našla tko ih je napravio, Time dodaje vrijeme i opcije za rad s vremenom, Random razne opcije za nasumične opcije te Sys koji služi za opcije sistema. Kako bi program radio treba imati sve library-e instalirane te Engain treba biti u istoj mapi kao i program old Memory-a.

```
import Engine #umetanje Engine, napravio ga je Leonardo
import pygame #umetanje pygamea
import time #umetanje time
import sys #umetanje sysa
import random #umetanje random
```

#### 3.2. Koliko program troši

Finalna mapa programa sa slikama, muzikom i Engainom zauzima 15.5 MB prostora na računalu te ima 14 fileova. Igra zauzima oko 23% CPU-a za procesor Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz 2.30GHz.