Univerzitet Crne Gore

ELEKTROTEHNIČKI FAKULTET

Studije primijenjenog računarstva

**Softversko inženjerstvo**

-dokumentacija mini-projekta-

**Tema: Memory Game**

Projekat radili:

1. Tijana Bošković 59/23
2. Nikolina Fatić 63/23
3. Andrej Kastratović 83/23
4. Luka Tomanović 39/23

Podgorica, 10.06.2025. godine

# Kratak opis projekta:

Cilj ovog projektnog zadatka bio je da se implementira edukativna i zabavna igra memorije koristeći programski jezik **Python** i GUI biblioteku **Tkinter**. Igra omogućava korisniku da razvija pažnju i pamćenje kroz pronalaženje parova skrivenih kartica.

Korisniku su ponuđeni različiti **nivoi**, od lakog (2x2) do teškog (6x6), uključujući i specijalne nivoe kao što su:

* **Hard Mode** sa vremenskim ograničenjem,
* **Blind Mode**, gdje su kartice prikazane na kratko, nakon čega igrač mora igrati "naslijepo".

Dodana su funkcionalna poboljšanja kao što su:

* brojanje pokušaja,
* vizuelno otkrivanje i skrivanje kartica,
* automatski povratak na početni ekran nakon pobede ili isteka vremena.

# Opis zadataka i doprinosa članova projektnog tima

## Nikolina Fatic:

Kreiranje klase, prikaz pocetnog ekrana kao i funkcionisanje kartica. Objašnjenje koda.

## Tijana Boskovic:

Kreiranje klase, kao i kreiranje modova(hard mod i blind mod). Objašnjenje koda.

## Andrej Kastratovic:

Izrada dokumentacije. Izrada funkcionalnih zahtjeva, testova prihvatanja, use case-ova.

## Luka Tomanovic:

Izrada use case dijagrama, kao i dijagrama sekvence i aktivnosti.

# Opis opštih funkcionalnosti softvera

Igra Memory Game je klasična logička igra u kojoj je cilj pronaći sve parove istih brojeva skrivenih iza kartica.

### **Osnovna pravila:**

1. Igra se odvija na mreži dimenzija 2x2, 4x4 ili 6x6, u zavisnosti od izabranog nivoa.
2. Svaka kartica krije broj. Igrač klikom otkriva dvije kartice.
3. Ako su brojevi isti, par ostaje otkriven.
4. Ako nisu isti, kartice se zatvaraju nakon kratkog vremena.
5. Igra se nastavlja dok se svi parovi ne pronađu.

### **Specijalni režimi:**

* **Hard Mode:**  
  Igra traje 60 sekundi. Ako ne pronađeš sve parove na vrijeme – gubiš.
* **Blind Mode:**  
  Kartice su prikazane na 3 sekunde na početku, nakon čega se igra nastavlja bez vizuelnog otkrivanja vrijednosti.

### **Cilj igre:**

Pronaći sve parove sa što manje pokušaja,a u nekim režimima i u što kraćem vremenskom roku.

Funkcionalni zahtjevi

|  |  |  |
| --- | --- | --- |
| Identifikator | PW | Zahtjev |
| Req1 | 5 | Sistem prikazuje početni ekran sa naslovom i dugmetom za početak igre. |
| Req2 | 5 | Sistem omogućava izbor između više nivoa težine (2x2, 4x4, 6x6, Hard Mode, Blind Mode). |
| Req3 | 5 | Sistem nasumično raspoređuje parove brojeva na polja i omogućava okretanje kartica. |
| Req4 | 2 | Sistem pamti broj pokušaja i prikazuje ga tokom igre. |
| Req5 | 3 | U režimu "Hard Mode", uključuje se tajmer koji odbrojava vrijeme. |
| Req6 | 3 | U režimu "Blind Mode", kartice se kratko otkriju na početku i zatim igra u potpunosti bez vidljivih vrednosti. |
| Req7 | 5 | Kada se svi parovi pronađu, prikazuje se poruka o pobedi i korisnik se vraća na izbor nivoa. |
| Req8 | 4 | Dugme za povratak na meni omogućava restart igre bez gašenja aplikacije. |

Slučajevi testa prihvatanja

* REQ1 – Početni ekran sa naslovom i dugmetom

**Koraci:**

1. Pokreni igru.
2. Provjeri da li se prikazuje naslov "Memory Game".
3. Provjeri da li postoji dugme "Počni igru".

* REQ2 – Odabir nivoa težine

**Koraci:**

1. Klikni na "Počni igru".
2. Provjeri da li se prikazuju svi ponuđeni nivoi:
   * Lako (2x2)
   * Srednje (4x4)
   * Teško (6x6)
   * Hard Mode
   * Blind Mode
3. Klikom na bilo koji nivo, provjeri da li igra kreće sa odgovarajućom veličinom table.

* REQ3 – Kartice se miješaju i mogu se okretati

**Koraci:**

1. Odaberi bilo koji nivo (npr. 4x4).
2. Klikni na bilo koje dvije kartice.
3. Proveri da li se vrijednosti otkrivaju.
4. Provjeri da li, u slučaju para, kartice ostaju otkrivene; ako nije par – ponovo se zatvaraju.

* REQ4 – Prikaz broja pokušaja

**Koraci:**

1. Odaberi nivo i započni igru.
2. Okreni dvije kartice.
3. Provjeri da li se broj pokušaja povećao za 1 i prikazuje pored teksta "Pokušaji".

* REQ5 – Tajmer u Hard Mode režimu

**Koraci:**

1. Izaberi nivo “Hard Mode”.
2. Provjeri da li se prikazuje tajmer sa 60s.
3. Sačekaj 60 sekundi bez pronalaženja svih parova.
4. Provjeri da li se pojavljuje poruka "Vrijeme isteklo".

* REQ6 – Blind Mode

**Koraci:**

1. Izaberi "Blind Mode".
2. Provjeri da li su sve kartice kratko prikazane (oko 3s).
3. Nakon toga, provjeri da li se karte zatvaraju i više se ne otvaraju klikom (osim prilikom pokušaja para).
4. Kada klikneš dva puta, proveri da li sistem upoređuje interne vrijednosti i samo potvrđuje parove.

* REQ7 – Završetak igre i povratak na meni

**Koraci:**

1. Igraj dok ne pronađeš sve parove.
2. Provjeri da li se pojavljuje poruka o pobjedi (sa brojem pokušaja).
3. Provjeri da li te sistem vraća na meni za izbor nivoa.

* REQ8 – Dugme za povratak

**Koraci:**

1. U bilo kom trenutku igre, klikni na dugme "Nazad na izbor nivoa".
2. Provjeri da li se vratiš na ekran sa izborom nivoa.

Use Case

Use Case: Pokretanje igre memorije

**Opis:**  
Ovaj Use Case opisuje proces pokretanja igre, uključujući prikaz početnog ekrana, izbor nivoa težine i pripremu kartica.  
**Glavni tok događaja:**

* Korisnik pokreće igru.
* Sistem prikazuje početni ekran sa dugmetom za početak.
* Nakon pritiska, korisniku se nudi izbor nivoa težine.
* Po izboru nivoa, sistem prikazuje mrežu kartica i pokreće igru.  
  **Izvedeno iz zahtjeva:** REQ1

Use Case: Izbor kartice i pokušaj pronalaska para

**Opis:**  
Ovaj Use Case opisuje korisnikov pokušaj da pronađe dva ista broja klikom na kartice.  
**Glavni tok događaja:**

* Korisnik klikne na jednu karticu – kartica se otkriva.
* Korisnik zatim klikne na drugu karticu.
* Ako su brojevi isti, kartice ostaju otkrivene.
* Ako nisu, kartice se vraćaju u početno stanje nakon kratkog vremena.  
  **Izvedeno iz zahtjeva:** REQ2

Use Case: Brojanje pokušaja i praćenje igre

**Opis:**  
Ovaj Use Case opisuje sistemsko brojanje pokušaja korisnika i označavanje toka igre.  
**Glavni tok događaja:**

* Sistem pamti svaki par klikova (pokušaj).
* Broj pokušaja se prikazuje u realnom vremenu na ekranu.  
  **Izvedeno iz zahtjeva:** REQ3

Use Case: Prikaz svih kartica na početku (Blind/Hard Mode)

**Opis:**  
Ovaj Use Case opisuje kako se u specijalnim režimima igre sve kartice kratko otkriju na početku.  
**Glavni tok događaja:**

* Po pokretanju igre u "Blind" ili "Hard" režimu, sistem prikazuje sve kartice na 3 sekunde.
* Zatim ih automatski zatvara i igra može da počne.  
  **Izvedeno iz zahtjeva:** REQ4

Use Case: Tajmer i ograničeno vreme (Hard Mode)

**Opis:**  
Ovaj Use Case opisuje proces odbrojavanja vremena kada je aktiviran režim sa tajmerom.  
**Glavni tok događaja:**

* Po pokretanju igre u "Hard Mode", sistem uključuje tajmer (npr. 60 sekundi).
* Tajmer se prikazuje u gornjem dijelu ekrana.
* Ako vrijeme istekne pre nego što su svi parovi pronađeni, igra se automatski završava porukom.  
  **Izvedeno iz zahtjeva:** REQ5

Use Case: Pobjeda u igri

**Opis:**  
Ovaj Use Case opisuje završetak igre kada korisnik pronađe sve parove.  
**Glavni tok događaja:**

* Korisnik pronađe sve parove kartica.
* Sistem prikazuje poruku o pobjedi, zajedno sa brojem pokušaja.
* Igra se resetuje ili se korisnik vraća na izbor nivoa.  
  **Izvedeno iz zahtjeva:** REQ6, REQ7

Use Case: Ponovno pokretanje igre

**Opis:**  
Ovaj Use Case opisuje ponovni početak igre putem dugmeta.  
**Glavni tok događaja:**

* Korisnik klikne na dugme “⬅️ Nazad na izbor nivoa”.
* Sistem briše trenutno stanje i prikazuje izbor nivoa.  
  **Izvedeno iz zahtjeva:** REQ8

Use Case dijagram

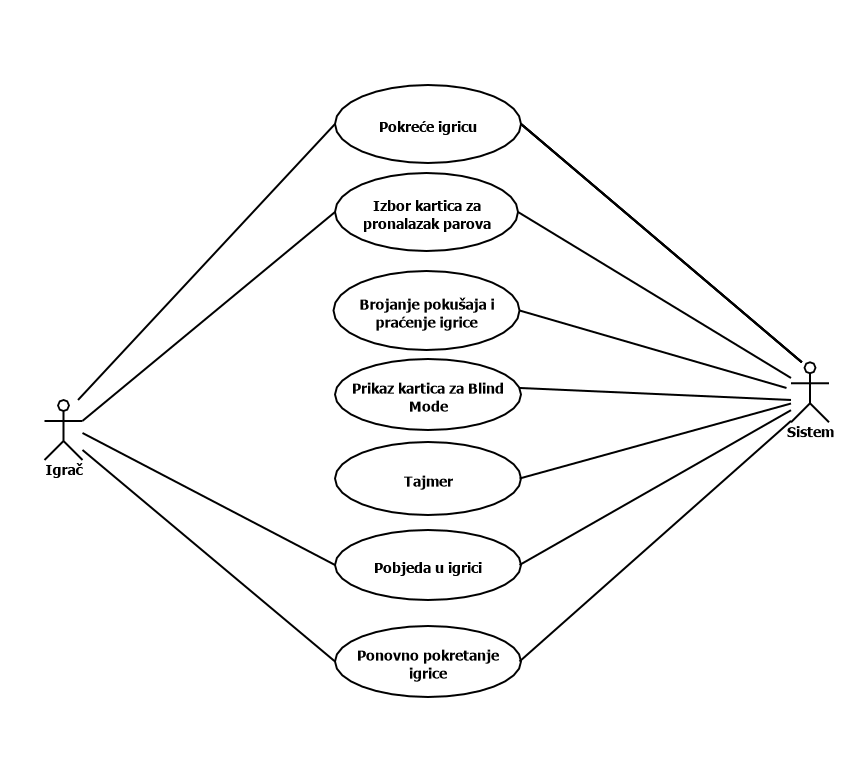
Use Case dijagram prikazuje osnovne funkcionalnosti aplikacije, kroz perspektivu korisnika sistema. Dijagram služi za vizuelno predstavljanje odnosa između **korisnika** i **glavnih slučajeva upotrebe (use case-ova)**, koji definišu ponašanje sistema na najvišem nivou apstrakcije.

Cilj dijagrama je da jasno ilustruje sve dostupne interakcije koje korisnik može ostvariti sa aplikacijom, uključujući izbor nivoa, igru, praćenje vremena, ponovno pokretanje igre i pregled rezultata.

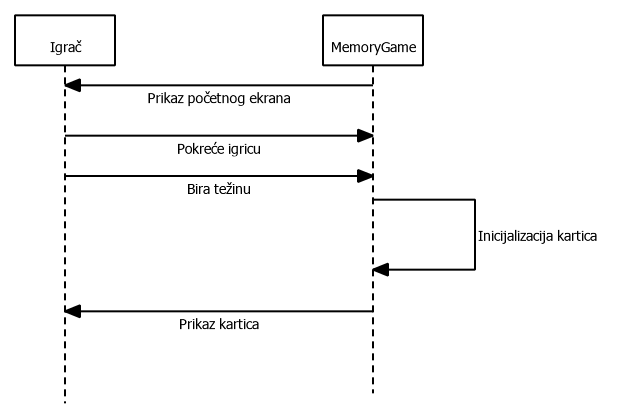
### Dijagram uključuje sledeće funkcionalnosti:

* **Pokretanje igre** i izbor nivoa težine
* **Otkrivanje i uparivanje kartica**
* **Brojanje pokušaja**
* **Specijalni režimi** kao što su Blind Mode i Hard Mode
* **Vremensko ograničenje (tajmer)**
* **Pobjeda i kraj igre**
* **Ponovno pokretanje igre**

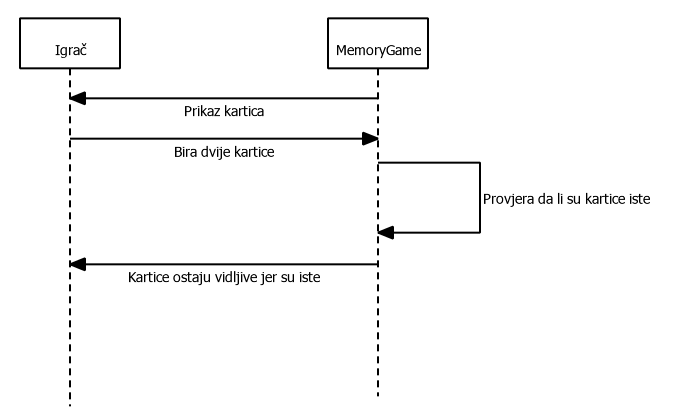
Svi ovi slučajevi upotrebe su izvedeni iz funkcionalnih zahtjeva sistema (REQ1 – REQ8) i predstavljaju osnovne tokove ponašanja aplikacije iz ugla krajnjeg korisnika.



Dijagram sekvence za početak igrice



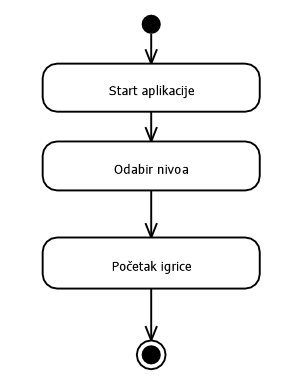
Dijagram sekvence za pogađanje karata

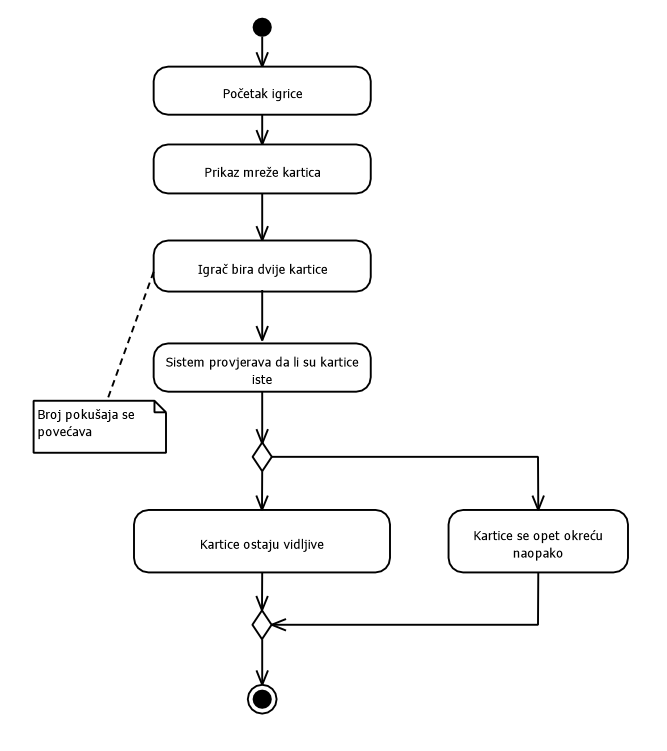


Dijagram klase



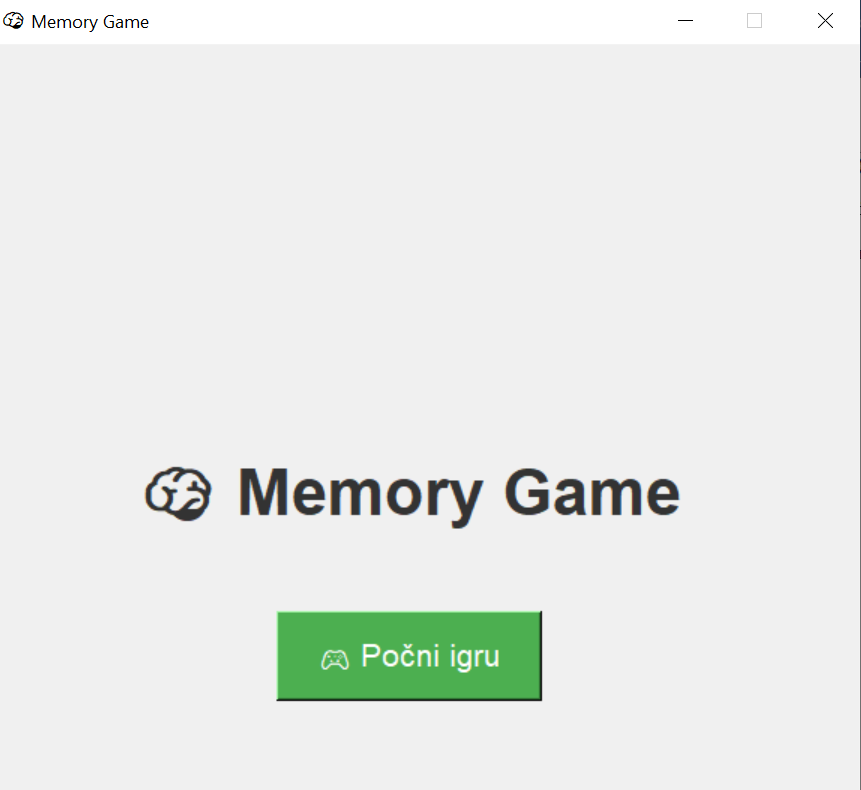
Dijagram aktivnosti





Interfejs

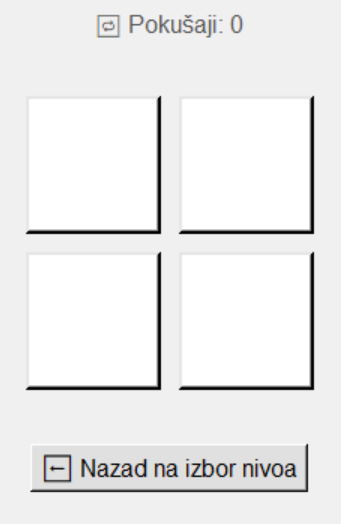
Početni ekran:



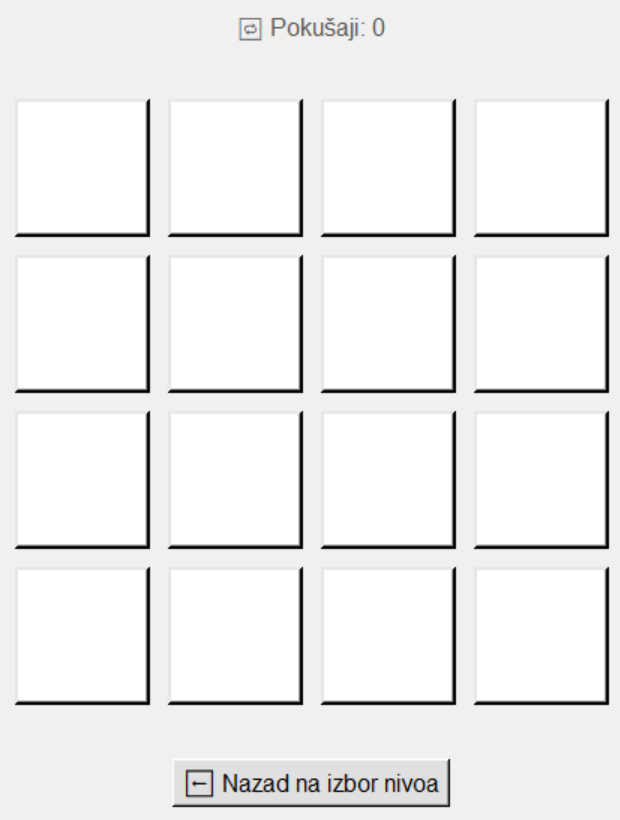
Odabir težine:



2x2



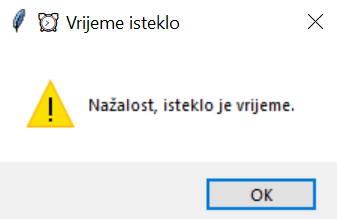
4x4



Blind mod



Obavještenje



Ključne funkcije i dijelovi koda

U konstrutkoru klase postavljamo osnovne postavke prozora, definišemo nivoe igre i inicijalizujemo sve pomoćne varijable koje prate stanje igre (koliko je pokušaja bilo, koje kartice su već okrenute, mod igre i slično). Na kraju se poziva funkcija koja prikazuje početni ekran.

Inicijalizacija klase

def \_\_init\_\_(self, root):

self.root = root

self.root.title("🧠 Igra Pamćenja")

self.root.configure(bg="#f0f0f0")

self.nivoi = {

"🟢 Lako (2x2)": 2,

"🟡 Srednje (4x4)": 4,

"🔴 Teško (6x6)": 6,

"🔥 Hard Mode (4x4 + tajmer)": 4,

"🕶️ Blind Mode (4x4 memorija)"}

self.prvi\_izabrani = None

self.drugi\_izabrani = None

self.dugmad = []

self.vrijednosti = []

self.okrenute = []

self.broj\_pokusaja = 0

self.oznaka\_tajmera = None

self.preostalo\_vrijeme = 30

self.hard\_mode = False

self.blind\_mode = False

self.prikazi\_pocetak()

Funkcija prikazuje početni meni sa naslovom i dugmetom za početak igre. IKada igrač klikne na dugme, poziva se funkcija odabir\_nivoa koja omogućava izbor težine.

Prikaz početnog ekrana

def prikazi\_pocetak(self):

self.obrisi\_sve()

okvir = tk.Frame(self.root, bg="#f0f0f0")

okvir.pack(expand=True)

naslov = tk.Label(okvir, text="Igra Pamćenja", font=("Helvetica", 32, "bold"), fg="#333", bg="#f0f0f0")

naslov.pack(pady=30)

dugme\_pocni = tk.Button(okvir, text="Počni igru", font=("Helvetica", 16), bg="#4CAF50", fg="white",

activebackground="#45a049", padx=20, pady=10, command=self.odabir\_nivoa)

dugme\_pocni.pack(pady=20)

Igrač bira jedan od nivoa igre. Klikom na neki nivo poziva se funkcija pocetak\_igre sa odgovarajućom veličinom mreže i imenom nivoa.

Izbor nivoa težine

def odabir\_nivoa(self):

self.obrisi\_sve()

self.hard\_mode = False

self.blind\_mode = False

okvir = tk.Frame(self.root, bg="#f0f0f0")

okvir.pack(expand=True)

labela = tk.Label(okvir, text="Odaberi nivo težine:", font=("Helvetica", 20, "bold"), bg="#f0f0f0", fg="#333")

labela.pack(pady=20)

for nivo, velicina in self.nivoi.items():

dugme = tk.Button(okvir, text=nivo, width=30, font=("Helvetica", 14),

bg="#2196F3", fg="white", activebackground="#1976D2",

command=lambda n=nivo, v=velicina: self.pocetak\_igre(v, n))

dugme.pack(pady=8)

Pokretanje igre

def pocetak\_igre(self, velicina, ime\_nivoa):

self.velicina = velicina

self.ukupan\_broj\_kartica = velicina \* velicina

self.obrisi\_sve()

self.prvi\_izabrani = None

self.drugi\_izabrani = None

self.dugmad = []

self.vrijednosti = []

self.okrenute = []

self.broj\_pokusaja = 0

self.broj\_uklopljenih = 0

self.preostalo\_vrijeme = 60

self.hard\_mode = "Hard Mode" in ime\_nivoa

self.blind\_mode = "Blind Mode" in ime\_nivoa

brojevi = list(range(1, self.ukupan\_broj\_kartica // 2 + 1)) \* 2

random.shuffle(brojevi)

self.vrijednosti = brojevi

self.okvir\_info = tk.Frame(self.root, bg="#f0f0f0")

self.okvir\_info.pack(pady=10)

self.oznaka\_pokusaja = tk.Label(self.okvir\_info, text="🔁 Pokušaji: 0", font=("Helvetica", 12), bg="#f0f0f0", fg="#555")

self.oznaka\_pokusaja.pack(side=tk.LEFT, padx=10)

if self.hard\_mode:

self.oznaka\_tajmera = tk.Label(self.okvir\_info, text=f"⏱️ Vrijeme: {self.preostalo\_vrijeme}s", font=("Helvetica", 12), bg="#f0f0f0", fg="#c62828")

self.oznaka\_tajmera.pack(side=tk.RIGHT, padx=10)

self.azuriraj\_tajmer()

self.okvir\_karte = tk.Frame(self.root, bg="#f0f0f0")

self.okvir\_karte.pack(pady=20)

for i in range(self.ukupan\_broj\_kartica):

dugme = tk.Button(self.okvir\_karte, text=" ", width=6, height=3, font=("Helvetica", 16, "bold"),

relief="raised", bg="#ffffff", fg="#333", bd=3,

command=lambda idx=i: self.okreni\_karticu(idx))

dugme.grid(row=i // velicina, column=i % velicina, padx=6, pady=6)

self.dugmad.append(dugme)

self.dugme\_nazad = tk.Button(self.root, text="⬅️ Nazad na izbor nivoa", font=("Helvetica", 12),

bg="#e0e0e0", command=self.odabir\_nivoa)

self.dugme\_nazad.pack(pady=10)

if self.hard\_mode or self.blind\_mode:

self.prikazi\_sve\_karte()

Kada igrač klikne na karticu, ona se okreće tako što se prikaže broj koji ima ta kartica. Pamte se indeksi kliknutih kartica. Kada se klikne na drugu karticu, poziva se funkcija za provjeru poklapanja brojeva.

Okretanje kartica

def okreni\_karticu(self, idx):

if idx in self.okrenute or self.drugi\_izabrani is not None:

return

if self.blind\_mode:

if self.prvi\_izabrani is None:

self.prvi\_izabrani = idx

elif self.prvi\_izabrani != idx and self.drugi\_izabrani is None:

self.drugi\_izabrani = idx

self.root.after(400, self.provjeri\_poklapanje)

return

dugme = self.dugmad[idx]

dugme.config(text=str(self.vrijednosti[idx]), state="disabled", disabledforeground="black", bg="#BBDEFB")

if self.prvi\_izabrani is None:

self.prvi\_izabrani = idx

elif self.prvi\_izabrani != idx and self.drugi\_izabrani is None:

self.drugi\_izabrani = idx

self.root.after(400, self.provjeri\_poklapanje)

Ako su izabrane kartice iste, one ostaju okrenute i označene kao par. Ako su različite, one se ponovo okreću naopako. Kada se upare sve kartice, igra se završava.

Provjera kartica

def provjeri\_poklapanje(self):

self.broj\_pokusaja += 1

self.oznaka\_pokusaja.config(text=f"Pokušaji: {self.broj\_pokusaja}")

vrijednost\_prve = self.vrijednosti[self.prvi\_izabrani]

vrijednost\_druge = self.vrijednosti[self.drugi\_izabrani]

if vrijednost\_prve == vrijednost\_druge:

self.okrenute.extend([self.prvi\_izabrani, self.drugi\_izabrani])

self.broj\_uklopljenih += 1

if self.blind\_mode:

self.dugmad[self.prvi\_izabrani].config(text=str(vrijednost\_prve), state="disabled", bg="#C8E6C9")

self.dugmad[self.drugi\_izabrani].config(text=str(vrijednost\_druge), state="disabled", bg="#C8E6C9")

if self.broj\_uklopljenih == self.ukupan\_broj\_kartica // 2:

messagebox.showinfo("Čestitamo!", f"Pobijedili ste za {self.broj\_pokusaja} pokušaja!")

self.odabir\_nivoa()

else:

if not self.blind\_mode:

self.dugmad[self.prvi\_izabrani].config(text=" ", state="normal", bg="#ffffff")

self.dugmad[self.drugi\_izabrani].config(text=" ", state="normal", bg="#ffffff")

if self.blind\_mode and vrijednost\_prve != vrijednost\_druge:

pass

self.prvi\_izabrani = None

self.drugi\_izabrani = None

Tajmer za hard mod

def azuriraj\_tajmer(self):

if not self.hard\_mode:

return

self.preostalo\_vrijeme -= 1

if self.oznaka\_tajmera:

self.oznaka\_tajmera.config(text=f"⏱️ Vrijeme: {self.preostalo\_vrijeme}s")

if self.preostalo\_vrijeme == 0:

messagebox.showwarning("⏰ Vrijeme isteklo", "Nažalost, isteklo je vrijeme.")

self.odabir\_nivoa()

else:

self.root.after(1000, self.azuriraj\_tajmer)

# Pregled literature, gotovih rješenja i softverske dokumentacije

<https://www.geeksforgeeks.org/python-gui-tkinter/>

https://www.simplilearn.com/tutorials/python-tutorial/python-graphical-user-interface-gui