

Veštačka Inteligencija

Izveštaj I faze projekta

Formulacija problema i interfejs

Slaganje

- Byte -

Naziv tima: A&A

Anđela Stojanović, 18406

Anastasija Trajković, 18456

U prvoj fazi projekta treba definisati način predstavljanja stanja problema tj. igre , osnovne funkcije igre i korisnički interfejs.

U ovoj fazi projekta implementirane su sledeće funkcije:

- Funkcija za prestavljanje stanja problema (igre)
- Funckija za unos početnih parametara igre
- Funkcija za proveru kraja igre
- Funkcija koja omogućava izbor ko će igrati prvi (čovek ili računar)
- Funkcija koja omogućava izbor ko će igrati prvi (X ili O)
- Funkcija za pravljenje inicijalnog stanja problema (igre)
- Funkcija za prikaz proizvoljnog stanja problema (igre)
- Funkcija za unos poteza
- Funkcija za proveru ispravnosti poteza

Funkcija za predstavljanje stanje problema (igre)

```
class Igra:
    def __init__(self):
        self.dimenzija = 0
        self.igrac = 'X'
        self.brojac_stack_x = 0
        self.brojac_stack_o = 0
```

```
class Tabla:
    def __init__(self, dimenzija) -> None:
        self.dimenzija = dimenzija
        self.matrica = []
        self.matrica_3x3 = [['.' for _ in range(2, -1, -1)] for _ in range(0, 3)]
```

Trenutno stanje igre prati se u klasi *Igra*. U klasi *Igra*, atribut *dimenzija* definiše veličinu table na kojoj se igra obavlja, i prosleđuje se klasi *Tabla* koja će iscrtati tablu. Definisani su atributi *igrac* koji je inicijalno postavljen na X, ali je moguća promena tog igrača. Imamo i attribute *brojac_stack_x* i *brojac_stack_o* koji služe da izbroje koliko stekova su igrači sakupili u toku igre, radi određivanja pobednika igre.

Klasi *Tabla* se prosleđuje parametar *dimenzija*, koja se koristi da predstavi tablu veličine *dimenzija x dimenzija*. U klasi *Tabla* pamtimo *matrica*, koja služi za predstavljanje kompletne table. Pored nje se pamti i *matrica_3x3* koja predstavlja polja unutar *matrice* (table) tj. stack-ove po kojima su poređane figure i po kojima je moguće odigrati potez. *Matrica_3x3* je na početku ispunjena tačkama, dok ne počne pomeranje figura.

Funkcija za unos početnih parametara igre

```
def postavi_velicinu_table(self)
```

Funkcija *postavi_velicinu_table(self)* definiše početno stanje na osnovu zadate veličine (dimenzije) table. Igrač definiše željenu veličinu table. Funkcija proverava ispravnost unosa, odnosno da li je dimenzija table deljiva sa 8 ili ima vrednost 10. Ako jeste kreira se tabla na osnovu unete dimenzije. Ako nije štampa se poruka o nevalidnom unosu i dozvoljava se igraču da ponovo unese dimenziju koja će zadovoljiti tražene uslove.

Funkcija za proveru kraja igre

```
def prover_i_kraj_igre(self, matrica):
```

Funkcija *proveri_kraj_igre(self, matrica)* proverava da li je igra gotova. Ovoj funkciji se prosleđuje tabla (tj. *matrica*), na osnovu koje se proverava kraj igre. *Brojac_figura* je postavljen na 0. Ispituje se da li je *matrica_3x3* (odnosno stack), popunjena figurama. Ako jeste povećava se *brojac_figura*.

Nakon toga se proverava da li je *brojac_figura* jednak 8, ako jeste ispituje se koja figura je poslednja stavljena u stack. Ako je poslednja figura u stack-u bila X onda taj stack odlazi igraču X i povećava se *brojac_stack_x*, u suprotnom se proverava da li je poslednja figura O, ako jeste povećava se *brojac_stack_o*.

Nakon toga funkcija proverava da li postoji igrač koji u svom vlasništvu ima više od polovine mogućih stekova. Ukoliko takav igrač postoji igra se završava i proglašava se pobednik, međutim ako takav igrač još uvek ne postoji, igra se nastavlja.

Funkcija koja omogućava izbor ko će igrati prvi (čovek ili računar)

```
def izbor_prvog_igraca(self):
```

Funkcija *izbor_prvog_igraca(self)* omogućava izbor ko će igrati prvi, računar ili čovek. Nakon izbora, igra počinje.

Funkcija koja omogućava izbor ko će igrati prvi (X ili O)

```
def postavi_igraca(self):
```

Funkcija *postavi_igraca(self)* omogućava da korisnik izabere koji je igrač, da li X ili O. Ovim izborom se određuje kojom figurom će igrati korisnik.

Funkcija koja obezbeđuje pravljenje inicijalnog stanja problema (igre)

```
def __init__(self, dimenzija):
```

U funkciji *def __init__(self, dimenzija)*, u klasi *Table*, se na osnovu definisanih atributa, koji su već objašnjeni, kreira matrica zadate dimenzije od strane igrača. U toj matrici se kreiraju polja koja mogu biti prazna ili mogu sadržati stekove (tj. matrice_3x3). U stack-

ove, se ređaju figure *X* ili *O*. Takođe, u ovoj funkciji su postavljene sve figure, *X* i *O*, na početnim mestima u stack-u, odnosno kreira se početno stanje igre.

Funkcija koja obezbeđuje prikaz proizvoljnog stanja problema (igre)

```
def iscrtaj_tablu(self):
```

```
def crtaj_tablu(self):
```

Funkcija ***iscrtaj_tablu(self)***, koja se nalazi u klasi *Tabla*, omogućava da se tabla i figure prikažu na traženi način koji je definisan za ovu igricu.

Funkcija ***crtaj_tablu(self)***, koja je deklarisan u klasi *Igra*, ima za cilj da pozove funkciju ***iscrtaj_tablu(self)*** iz klase *Tabla*, kako bi tabla i figure bile vidljive u ovoj klasi, jer se u njoj izvršava igrice.

Funkcija za unos poteza

```
def unos_poteza(self):
```

Funkcija ***unos_poteza(self)*** obezbeđuje unos poteza od strane igrača. Potez se sastoji od pozicije polja, mesta figure na steku i smeru pomeranja (GL, GD, DL, DD). Proverava se da li je igrač uneo vrstu kao string, kolonu kao broj, onako kako su obeležene na tabli, mesto kao broj ali samo u opsegu od 0 do 7, jer stack može posedovati maksimalno 8 figura, i smer jedan od četiri moguća.

Ukoliko jedan od ovih parametar nije unet onako kako bi trebalo, igrač se obaveštava o grešci, i pruža mu se mogućnost ponovnog unosa poteza. U ovoj funkciji se poziva funkcija ***ispravnost_poteza(self, potez)*** koja proverava da li je potez ispravan, ukoliko jeste vraća potez, i potez se odigrava (faza II).

Funkcija za proveru ispravnosti poteza

```
def ispravnost_poteza(self, potez):
```

Funkcija ***ispravnost_poteza(self, potez)*** proverava da li je unos poteza ispravan. Odnosno, proverava da li zadato polje postoji na tabli, zatim da li postoje figure na zadatom polju, da li postoji figure na zadatom mestu na steku na zadatom polju, i da li je smer jedan od četiri moguća (GL, GD, DL, DD).

Takođe, ova funkcija poziva funkciju: ***def vrati_mesto(self, element)***, koja ima za cilj da vrati indekse svih elemenata u matrici *_3x3*, kako bi mogla da proveriti da li figure na zadatom mestu, koje je igrač uneo, zaista postoje.

Ukoliko su svi definisani uslovi ispunjeni, odnosno vraćaju *True*, cela funkcija vraća *True*, i potez je ispravan.

