**Trouglići (TRIGGLE)**

**Faza I – Formulacija problema i interfejs**

TIM: ENT Grupa  
1. Ljiljana Stojičić 18942  
2. Mihajlo Ranđelović 18872  
3. Vladimir Stojmenović 18947

**Za ovu fazu definisali smo sledeće bitne funkcije:**  
Konstruktor \_\_init\_\_ inicijalizuje instancu klase za igru. Postavlja da li računar učestvuje u igri (isComputerPlaying), definiše veličinu table (boardSize), određuje koji igrač počinje igru (firstPlay), te postavlja početne vrednosti za poene oba igrača (oPoints i xPoints). Takođe, računa broj poena potrebnih za pobedu (halfPoints), inicijalizuje praznu listu za tabelu igre (board) i prazne rečnike za praćenje stanja grana (branchState) i trouglova (triggleState), kao i promenjivu gameFinished ukazuje na to da li je igra završena.



Funkcija initialize generiše početnu strukturu table igre u obliku dvodimenzionalnog trougla i poziva funkciju initialState koja postavlja početno stanje table. Za svaki red iznad i ispod sredine, računa odgovarajući broj praznina i zvezdica kako bi kreirala pravilnu geometrijsku formu trougla. Prazni redovi se dodaju između glavnih redova radi pravilnog razmaka. Nakon što je tabla kreirana, funkcija poziva initializeState kako bi postavila početno stanje temena, grana i trouglova.

Funkcija print ispisuje trenutno stanje table igre. Prolazi kroz svaki red u internoj reprezentaciji table (self.board) i ispisuje ga u konzoli, omogućavajući pregled trenutnog rasporeda elemenata na tabli.

Funkcija position\_to\_matrix konvertuje poziciju na tabli u formatu slovo-broj (npr. "A1") u odgovarajuće koordinate u matrikskoj reprezentaciji (red i kolona). Na osnovu slova određuje odgovarajući red, dok broj koristi za izračunavanje kolone. Ako je red iznad ili ispod centralnog reda, primenjuje pomak (offset) kako bi prilagodila kolonu u skladu s geometrijom table. Na kraju vraća izračunate matrikske koordinate kao par (row, col).

Funkcija matrix\_to\_position konvertuje koordinate u matrikskoj reprezentaciji table (red i kolona) u poziciju na tabli u formatu slovo-broj (npr. "A1"). Izračunava odgovarajuće slovo na osnovu reda u matrici, dok broj zavisi od kolone i pozicije reda u odnosu na srednji deo table. Ako je red iznad ili ispod centralnog reda, koristi pomak (offset) kako bi pravilno mapirala broj, uzimajući u obzir geometrijsku strukturu table. Rezultat vraća kao tekstualnu oznaku pozicije.

Funkcija initializeState postavlja početno stanje igre identifikujući temena na tabli, njihove pozicije i međusobne veze. Temena se pronalaze na osnovu oznake "\*" u matrikskoj reprezentaciji, a zatim se kreiraju grane između svakog temena i njegovih validnih suseda (donji, desni i donji-desni). Svaka grana dobija jedinstveni ID i postavlja se kao nezauzeta. Na osnovu povezanih temena i njihovih grana formiraju se trouglovi, koji se takođe mapiraju sa pripadajućim ID-ovima grana i središtem trougla. Trouglovi se inicijalno postavljaju kao nezauzeti, spremni za ažuriranje tokom igre.

Funkcija draw\_and\_update služi za crtanje poteza, validaciju i ažuriranje stanja igre. Prvo proverava validnost unetog poteza (slovo, broj i smer) kako bi se osiguralo da je potez u okviru pravila i dimenzija table. Na osnovu smera poteza identifikuje grane koje treba zauzeti i ažurira njihovo stanje kao zauzeto. Zatim ažurira tablu tako što dodaje linije koje predstavljaju potez (horizontalne, dijagonalne levo ili desno). Na kraju, funkcija proverava da li su zauzeti svi delovi nekog trougla i, ako jesu, dodeljuje ga trenutnom igraču, ažurira bodove i vizuelno prikazuje osvajanje na tabli. Funkcija vraca True ukoliko je potez uspešno odigran, a False ukoliko nije došlo do promene usled nevalidnosti.

**Screenshot_1**

Funkcija check\_win služi da proveri da li je jedan od igrača pobedio nakon odigrane igre. Pobednik se odlučuje na osnovu stanja poena koji se ažuriraju nakon formiranja svakog trouglića.

Screenshot_2

Funkcija check\_position služi za proveru validnosti unosa pozicije. Koristi se u funkciji draw\_and\_update da proveri ulazne parametre i spreči sprovođenje runde sa nevalidnom pozicijom. Ako dođe do nevalidnosti igra se ne prekida već se samo ispisuje poruka sa greškom.

TEST:

**test.py fajl**: implementacija testa za prvu fazu projekta (*testRound* - primer odigrane runde; *testGame* - primer završene igre)