**Logika implementacije I faze projekta Triggle**

Ovaj dokument obuhvata celokupno objašnjenje logike koja je primenjena za implementaciju I faze projekta Triggle. U prvoj fazi je bilo potrebno odraditi sledeće funkcionalnosti :

1. Omogućiti izbor ko igra prvi (čovek ili računar).

2. Omogućiti izbor simbola igrača koji igra prvi (X ili O).

3. Unos dimenzija table i validacija unosa.

4. Generisanje početnog stanja igre na osnovu unosa.

5. Prikaz trenutnog stanja table sa pozicijama čvorova i poteza.

6. Unos poteza sa validacijom unosa.

7. Provera kraja igre na osnovu popunjenosti table ili broja zauzetih trouglića.

8. Definisati način za predstavljanje stanja problema (igre)

* Predstavljanje pozicija razvučenih gumica i zauzetih trouglića.

9. Napisati funkciju za postavljanje početnog stanja

* Definiše se na osnovu zadate veličine table.

10. Napisati funkcije za proveru kraja igre

* Tabla je popunjena razvučenim gumicama ili je igrač zauzeo više od polovine

Trouglića.

Radi bolje organizovanosti i preglednosti koda sve funkcije za izradu prve faze su podeljene u sledeće fajlove:

* **board.py** : sadrži sve funkcije za rad sa tablom
* **game\_logic.py**: sadrži sve funkcije vezane za logiku same igre, poteza i provere validnosti, kraja igre…
* **utilities.py**: sadrži sve pomoćne funkcije
* **main.py**: glavni deo programa

U nastavku dokumenta sledi opis ideje koja je primenjena za svaku od funkcionalnosti.

Inicijalizacija table za igru Triggle

U našoj implementaciji, funkcija **draw\_hexagon(side\_length)** generiše matricu koja predstavlja heksagonalnu tablu.

* Korisnik na početku unosi dimenziju table kao veličinu stranice šestougla (side\_length) , nakon čega se kreira matrica sa tačkama (stubovima u igri) označenim kao '●' .
* računaju se visina i širina matrice i kreira se matrica sa praznim poljima
* popunjava se matrica tačkama uporedno i simetrično odozgo i odozdo ka sredini
* srednji red se popunjava u narednoj petlji
* svaka tačka se pamti u čvoru (nodes) koji je implementiran kao dictionary koji čuva par x,y koordinata
* obezbedjuje se da svaki red kao i kolona sadrži dovoljno prostora za potege i trougliće
* funkcija vraća iscrtanu tablu kao i sve čvorove table.

Prikaz table

Funkcija **print\_board(matrix, nodes)** prikazuje trenutno stanje table na osnovu matrice matrix.

* red first\_row omogućava da brojevi kolona počinju sa leve strane ali da budu poravnati sa stubovima matrice za gornju polovinu matrice, kao i last\_row za donji deo
* brojevi se zatim dodaju na svakoj šestoj poziciji i tako omogućujemo da se odgovarajuć broj kolone nadje u liniji sa odgovarajućom kolonom.
* prolazi se kroz svaki red matrice i ukoliko je on deljiv sa 3 dodaje oznaku reda kao veliko slovo azbuke (dobijeno pomoću ASCII vrednosti, dodavanje u svakom trećem redu jer su tu čvorovi). Pre ispisivanja smo sve elemente pretvorili u stringove zbog korišćenja join funkcije.

Unos poteza igrača i validacija poteza

Unos poteza igrača omogućujemo kroz funkciju **play\_move(matrix, nodes, start, direction)**. Potez uključuje početnu poziciju (npr. A1) i smer razvlačenja gumice (D - desno, DD - dijagonalno desno, DL - dijagonalno levo).

* na početku funkcije definisani su svi validni pravci i izvršena je provera da ukoliko nije unet dozvoljen pravac ili ukoliko stub oko kog želimo da krenemo razvlačenje gumice nije u nodes, potez je nevalidan.
* u positions listi se čuvaju sve pozicije matrice koje treba ažurirati potezom. U zavisnosti od direction-a u position listu se dodaju pozicije na kojima bi trebale da budu gumice.
* pre nego da se potez ažurira, vršimo proveru da li su sve pozicije validne (funkcija **is\_valid\_move(matrix, x, y)**, proverava da li je koordinata x,y unutar granica matrice i da li postoji čvor) i da li je na poslednjoj poziciji stub(gumice obuhvataju 4 stuba)
* ukoliko jesu, dodaje se poteg u matrici, a ukoliko nisu onda je nevalidan potez.

Zauzimanje trouglića (tj. x ili o) na tabli i provera za kraj igre

Ovaj deo zadatka smo implementirale pomoću funkcije **draw\_triangle(matrix, symbol, count)**.

* Prolazimo kroz svaku poziciju matrice pomoću dvostruke petlje i ako je trenutna pozicija u matrici prazna, onda se prvo vrši provera da li je moguće formirati trougao. Postoje dve opcije:
* “–” na dnu(pozicija gore (i+1, j) mora biti validna i sadržati "-", pozicija gore-desno (i-1, j+1) mora biti validna i sadržati "\", pozicija gore-levo (i-1, j-1) mora biti validna i sadržati "/",
* “–” na vrhu (pozicija dole (i-1, j) mora biti validna i sadržati "-", pozicija dole-levo(i+1, j-1) mora biti validna i sadržati "\", pozicija dole-desno (i+1, j+1) mora biti validna i sadržati "/", )
* i ako jeste dodajemo ga i označavamo simbolom koji predstavlja igrača (x ili o) i povećavamo count
* Funkcija vraća broj nacrtanih trouglova.

Promena igrača (+započet deo II faze)

Pomoću funkcije **switch\_player(current\_player)**, omogućeno je da se menja koji igrač je trenutno na potezu (računar ili čovek).

Izbor koji igrač igra prvi (čovek ili računar)

Ovu funkcionalnost smo implementirale u Main delu programa. Od korisnika se zahteva da unese ko igra prvi (čovek ili računar) i da ukoliko navede neku drugu opciju od te dve se ponovo vrati i izabere jednu od dve moguće.

Izbor simbola igrača koji igra prvi (X ili O)

I ovu funkcionalnost smo implementirale u Main delu programa. Od korisnika se zahteva da unese koji simbol bira i da ukoliko navede neku drugu opciju od te dve se ponovo vrati i izabere jednu od dve moguće ( X ili O). Nakon njegovog odabira za drugog igrača se postavlja suprotan simbol i pravi se dictionary symbols za čuvanje informacija o simbolima.

Prikaz trenutnog stanja tabele na osnovu prosledjenog poteza

Funkcija **parse\_matrix(matrix\_str)** pretvara stanje prosledjeno stringom u matricu sa tabelom koja sadrži tablu sa već odigranim potezima (priprema za obradu u narednoj fazi).

Kraj igre

Funkcija za kraj igre **end\_of\_game(matrix, count, side\_length)** koristi pomoćnu funkciju **max\_connections(side\_length)** koja računa koliko potega je moguće povući kroz tablu i završava igru ako:

* tabla popunjena (svi potezi izmedju čvorova povučeni)
* neki od igrača je popunio više od pola trouglića i tako pobedio.