Projekat Domineering

**I faza:**

U ovoj fazi projekat je smesten u dva fajla (*main.py* i *board\_funcs.py*).

Fajl *main.py* sadrzi main funkciju koja realizuje game loop. U ovoj funkciji se pozivaju sve funkcije za iscrtavanje table, domina i koordinata koje se nalaze u fajlu *board\_funcs.py*. U njoj se nalaze i event handleri za sve interakcije sa programom.

Fajl *board\_funcs.py* sadrzi sledece funkcije:

**draw\_squares**(screen,board,LIGHT,GREY) – funkcija koja crta tablu *board* na povrsini *screen* a boje kvadratica su *LIGHT* i *GREY* (rgb vrednosti) i naizmenicno se menjaju

**draw\_o**(screen,color,x,y) – funkcija koja crta O dominu na zadatu poziciju

**draw\_x**(screen,color,x,y) – funkcija koja crta X dominu na zadatu poziciju

**draw\_domine**(board,screen,w,h) – funkcija koja iscrtava sve postavljene domine na tabli *board*

**draw\_coords**(screen,font,w,h) – funkcija koja crta koordinate oko table

**resize\_board**(new\_w,new\_h,w,h) – funkcija koja menja velicinu table

**init\_board**(x,y) – funkcija koja pravi tablu zadate velicine

**find\_square**(x,y) – funkcija koja pronalazi kvadrat u tabli u zavisnosti od prosledjenih koordinata prilikom klika misem na ekranu (pronalazi stvarno polje u matrici)

**check\_end**(board,player) – funkcija koja proverava da li je ispunjen uslov kraja igre

**II faza:**

U ovoj fazi obezbedjeno je odigravanje igre izmedju 2 coveka, implementriane su funkcije za promenu stanja, proveru validnosti poteza i nalazenje svih mogucih narednih stanja na osnovu prethodnog stanja i igraca koji je na potezu. To je implementirano kroz:

**check\_if\_valid**(turn, true\_target, board\_width , board\_height) – metoda koja na osnovu igraca koji je na potezu *turn* i polja na koje je igrac kliknuo *true\_target* proverava da li je uneti potez validan tj. da li je moguce postaviti igracevu plocicu na tablu sirine *board\_width* i visine *board\_height.*

**change\_state**(board,target,turn) – metoda koja na osnovu trenutnog igraca *turn* i odigranog poteza *target* menja stanje na tabli *board* ako su izabrana polja slobodna i prepusta potez sledecem igracu. Ako polja nisu slobodna stanje ostaje nepromenjeno.

**find\_all\_possible\_states**(turn,board) – metoda koja na osnovu igraca koji je na potezu *turn* i trenutnog stanja na tabli *board* pronalazi sva moguca naredna stanja (sve moguce table).

**III faza:**

U ovoj fazi obezbedjeno je odigravanje partije izmedju coveka i kompjutera, implementriane su funkcije za odigravanje poteza kompjutera (minimax algoritam) i sve pratece funkcije koje omogucavaju odigravanje partije. Implementirane su sledece funkcije:

**minimax\_alpha\_beta**(stanje, dubina, moj\_potez, player,alpha=(None, -1000), beta=(None, 1000) ) –

funkcija koja na osnovu toga da li je na potezu max ili min(parametar *moj\_potez*: True | False) igrac poziva na izvrsenje funkcije *max\_value* ili *min\_value*.

**max\_value**(stanje, dubina, alpha, beta, player, old\_state,cpu\_on\_move,next\_state =None) –

funkcija minimax algoritma koja odredjuje najbolje moguce stanje za max igraca upotrebom alfa-beta odsecanja

**min\_value**(stanje, dubina, alpha, beta, player , old\_state,cpu\_on\_move, next\_state=None) –

funkcija minimax algoritma koja odredjuje najbolje moguce stanje min igraca upotrebom alfa-beta odsecanja.

U ovoj funkciji nalazi se zakomentarisani deo koda koji sluzi da bi se algoritam ubrzao pri odigravanju partije na tabli cija je sirina ili visina veca od 5. Kompjuter bi tada odigravao najvise ogranicavajuce poteze (oduzima 4 poteza protivniku) dok je to moguce bez pokretanja minimax algoritma

**oceni**(stanje,turn,prevState,cpu\_on\_move) – funkcija koja vraca numericku ocenu stanja (heuristicka funkcija)

**DD**(state,oponent,oldNumOfOpStates) – funkcija koja proverava da li stanje odnosno odigrani potez najvise ogranicava protivnika (na osnovu toga koliko poteza oduzima protivniku, implementirano najmanje 4)

**\***Koriscenjem fajla *checkbox.py* omogucena je upotreba checkbox-ova u pygame biblioteci tako da je graficki omogucen izbor odigravanja prvog poteza izmedju igraca i kompjutera.

Dimitrije Marinkovic 17739

Ivan Markovic 17746