

Klasa Tabla

Predstavlja panel na kome ce biti dugmici koji ce moci da se pritiskaju i na kojima ce se videti trenutno stanje table.

Postoji privatna klasa Dugme koja predstavlja Button sa indeksom.

public Tabla(Panel pl) Funkcija koja na plocu dodaje 25 dugmadi i pravi osluskivace za njih.

public synchronized int koJePritisnut() Funkcija koja ceka da se pritisne neko od 25 dugmica na tabli.

private synchronized void dalje() funkcija koja signalizira da je dugme pritisnuto.

public void postaviOznaku(int ind, String oznaka) Postavlja oznaku na dugme.

public void prazni() Vraca tablu u prvobitno stanje.

public void oboji(int ind, int broj) Boji dugme na tabli u boju shodno visini tog polja. Svetlozelena je za visinu 1, malo jaca zelena za visinu 2, tamnoz zelena za visinu 3 i plava boja za polje sa kupolom.

Klasa Prikaz

Formira prozor i komponente na njemu da korisnik vidi sve potrebne objekte za pracenje igre.

Privatna klasa Izbor služi za biranje dubine pretrage u minimaks algoritmu.

Privatna klasa Sopro formira padajucu listu za biranje rezima rada kada igra racunar protiv racunara i za biranje da li ce se potezi citati iz datoteke.

Privatna klasa Dodatno predstavlja panel na kome se nalaze komponente na severnom delu prozora.

Privatna klasa Biranje formira igrace i parametre potrebne za njih ukoliko je neki od igraca racunar.

Privatna klasa Donje omogucava pravljenje panela na juznom delu prozora. Postoji dugme koje kada se pritisne formira igru i tada kreće da radi nit.

private void popuni() Metoda koja postavlja komponente na prozor.

public Prikaz() je konstruktor koji formira prozor i cini ga vidljivim.

Klasa Polje

Ideja je da ova klasa predstavlja po jedno polje na tabli. Pamti se da li je to polje zauzeto i ako jeste figura kog igraca je na njemu. Svako polje ima svoj indeks.

Klasa Igrac

Igrac je apstraktna klasa koja poseduje polja za oznaku tog igraca, tablu na kojoj taj igrac igra igru, ID igraca i dubinu do koje ce se vrsiti minimaks algoritam ako je igrac racunar. Labela je polje koje u koje se upisuje vrednost heuristicke funkcije ako je igrac racunar.

public int odigraj() throws InterruptedException je metoda koja služi za biranje polja.

Klasa Covek extends Igrac

Ova klasa nasledjuje metodu birajPolje() i vraca indeks pritisnutog polja u matrici.

Klasa Racunar extends Igrac

public int nadjiMetrinku(Polje[][] matrica, int ko, int gde, int gradnja) Metoda koja trazi metriku stanja matrice koja je prosledjena. Parametar ko predstavlja identifikator igraca koji racuna minimax funkciju. Parametar gde predstavlja indeks polja na koje bi figura bila pomerena, dok parametar gradnja

predstavlja indeks polja na koje bi se dodala pločica. Funkcija prvo pronalazi koordinate figura oba igrača i zatim računa heurističku funkciju po formuli $f=m+l$ gde je m visina polja na koje je pomaknuta figura a l proizvod trenutne visie gde se dodaje pločica i razlike rastojanja protivničkih figura i figura igrača za koji se poziva minimax.

`public int nadjiMetrinku1(Polje[][] matrica, int ko, int gde, int gradnja)` je metoda slična prethodnoj. Razlika je sto ako se nađe na situaciju gde je pobednik igrač koji je zvao minimax vraća se vrednost 500 ako je trenutni igrač max, odnosno -500 ako je trenutni igrač min. Ako nema protivničkih igrača u blizini polja gde je dodata kupola funkcija vraća 400, odnosno -400. Ako je barem jedna od protivničkih figura udaljena od obe figure za više od 5 polja funkcija vraća 300 odnosno -300. Ako nijedan od prethodnih uslova nije ispunjen ova metoda vraća potpuno isti rezultat kao prethodna.

`public int imalikuda(Polje[][] matrica1, int red)` Metoda koja proverava da neki igrač nije zarobljen tj. da li ima gde da pomeri barem jednu od svojih figura. Metoda vraća nulu ako igrač nema gde da se pomeri i to označava kraj igre.

`public int minimax(Polje[][] matrica, int maxDubina, int trenutnaDubina, int minilimax, int alfa, int beta)` Prvo se proverava da li je došlo do kraja igre i da li je trenutna dubina jednaka maksimalnoj dubini. Ako je to slučaj računa se metrika. Koja će se funkcija zvati zavisi od toga da li je u pitanju takmičarski igrač ili ne. Potom se izlazi iz funkcije. Ako to nije slučaj najbolja vrednost se postavlja na -1000000 ako je trenutni igrač max odnosno na 1000000 ako je trenutni igrač min. Potom se pronadju figure igrača koji je na potezu. Zatim se pronadju sva susedna polja koja su slobodna i ispunjavaju uslove da se na njih pomeri neka od figura. Zatim se za svako od tih polja pronadju susedna polja na kojim je dozvoljena gradnja. Za svaku trojku (figura, polje na koje će se figura pomeriti, polje gde će se graditi) se pravi nova tabla i za takvu tablu se poziva minimax funkcija. Ako je na potezu max i vraćena vrednost je bolja od najbolje vrednosti vraćena vrednost postaje najbolja vrednost. Ako je vraćena vrednost veća od donje granice za alfa-beta odsecanje donja granica postaje najbolja vrednost. Ako je najbolja vrednost veća od gornje granice izlazi se iz funkcije i vraća se najbolja vrednost. Ako je na potezu min i vraćena vrednost je manja od najbolje vrednosti vraćena vrednost postaje najbolja vrednost. Ako je vraćena vrednost manja od gornje granice za alfa-beta odsecanje gornja granica postaje najbolja vrednost. Ako je najbolja vrednost manja od donje granice izlazi se iz funkcije i vraća se najbolja vrednost.

`public int birajPolje()` throws InterruptedException je metoda koja se poziva da igrač odabere polje. Ako je u pitanju postavljanje figura polje se bira nasumično. Ako to nije slučaj, a u pitanju je prvi deo poteza gde samo treba odabrati figuru koja će se pomeriti, poziva se minimax funkcija. Drugi i treći deo poteza određuju se minimax algoritmom pa kada se dodje do njih samo se vraćaju prethodno izračunate vrednosti. Ako je izabran usporen režim rada pri svakom pozivu ove funkcije čeka se po jedan sekund pre početka izvršavanja.

Klasa Igra extends Thread

`public static int[] komsije(int ko)` pronalazi susedna polja polju čiji je ID prosledjen funkciji.

`public int imalikuda(int red)` proverava da li igrač čiji je ID prosledjen može barem jednu od svojih figura pomeriti negde. Ako ne može vraća se vrednost 0, u suprotnom se vraća vrednost 1.

`public boolean mozeli(int indeks)` je metoda koja proverava da li se figura koja se nalazi na polju sa rednim brojem indeks može pomeriti negde.

public String pretvori(int br) pretvara vrstu matrice u slovnu oznaku.

public int obrni(char s) pretvara slovnu oznaku u broj vrste.

public void run() Je metoda koja na pocetku prverava da li je potrebno poteze upisivati u datoteku ili citati iz datoteke. Zatim se proverava da li je kraj igre i ako nije odredjuje se ko je sledeci na potezu. Potom se proverava koja je faza poteza u pitanju i da li je korisnik odigrao ispravan potez. Prozor sa tablom se azurira pri svakoj promeni.

void prekini() je meroda koja prekida igru.