

Dokumentacija

Santorini

Nikola Kesic

16/0642

Main:

// Inicijalizacija pocetnih vrednosti

private void initElements(Scene scene);

// Citanje stanja igre iz datoteke

private void readGameFromFile(File f);

// OnClick listener

private void onClick(Canvas canvas, double x, double y);

// Pozivanje odredjene f-je zavisno od kliknutog dugmeta

private void takeButtonActions(int width,int height,double x,double y);

// Sta je kliknuto

private Action getAction(int width,int height,double x,double y);

// Crtanje svega

private void draw(Canvas canvas);

// Crtanje table

private void drawBoard(GraphicsContext gc, int width,int height);

// Crtanje igraca

private void drawPlayers(GraphicsContext gc, int width,int height);

// Crtanje dugmica

```
private void drawButtons(GraphicsContext gc,int width,int height);
```

```
// Crtanje informacija
```

```
private void drawStats(GraphicsContext gc, int width,int height);
```

```
-----
```

Game.java:

```
// Pokretanje igre
```

```
public void startGame();
```

```
// Promena tezine igre
```

```
public void changeDifficulty();
```

```
// Promena nacina( PvsP/PvsB/BvsB )
```

```
public void changeGameMode();
```

```
// Sta se desi na klik
```

```
public void clicked(int row, int column);
```

```
// Obrada kad se igra PvsB
```

```
private void playerVsBot(int clickedRow, int clickedCol);
```

```
// Obrada kad se igra PvsP
```

```
private void playerVsPlayer(int clickedRow,int clickedCol);
```

```
// Obrada kad se igra BvsB
```

```
private void botVsBot(int row,int col);
```

```
// Da li je kraj igre
```

```
private void checkForGameOver();
```

```
// Upis poteza u datoteku
```

```
private void writeMove(Move move);

// Da li je pomeranje moguće

private boolean possibleMove(int row,int col);

// -||-

public static boolean isPossibleMove(Board board, Position from, Position to);

// Može li se staviti pločica

public static boolean canPutAPlate(Board board, Position from, Position to);

// Odigraj potez

public void playTheMove(Move mov);

// Citanje igre iz fajla

public void readGameFromFile(File file);
```

Board:

```
// Da li je ne stoji niko na polju

public boolean emptyField(int r, int c);

// Funkcija evaluacije data u postavci

public int evaluateMinimaxMove(Position destination,Position plateDestination, int pid);

// Unapredjena funkcija evaluacije

public int improvedEvaluation(Position destination, Position plateDestination,int pid);

// Da li je polje nedostizno

private boolean unreachable(Position plate, Position p1,Position p2);

// Lista svih poteza

public ArrayList<Move> listOfAllMoves(int playerId);
```

```
// Tabla nakon odigranog poteza

public Board newBoardAfterPlayedMove(Move move);

// Da li igrac moze da se pomeri

public boolean canMove(int pid);

// Da li je gotova igra

public boolean isGameOver(int pid);

// Pozicija

class Position;

// Potez

class Move;

-----

Bot:

// Da li sam na potezu

protected boolean isMyTurn();

// Procena stanja igre

protected int evaluateBoard(Board prevBoard, Move move);

// Unapredjena procena stanja igre

protected int improvedEvaluation(Board game, Move move);

// Odigraj potez

public abstract void makeTheMove();
```

MinimaxBot:

// Odigraj potez

public void makeTheMove();

// Minimax

private int minimax(int initialDepth, int depth, Board game, boolean isMaximisingPlayer, int playerId, Board prevBoard, Move played);

AlphaBetaBot:

// Odigraj potez

public void makeTheMove();

// AlphaBeta Minimax

private int alphaBetaMinimax(int alpha, int beta, int initialDepth, int depth, Board game, boolean isMaximisingPlayer, int playerId, Board prevBoard, Move played);
