Dokumentacija

**Algoritam:** Primenjeni su algoritmi minimax i minimax sa alfa-beta odsecanjem. U slučaju odsecanja, algoritam je vidno brži, pogotovo pri dubljem grananju. Sam algoritam nije značajno drugačiji u odnosu na onaj predstavljen na časovima predavanja i vežbi, odnosno koji se može naći u literaturi za predmet. Razlike su prisutne zbog same implementacije, a način funkcionisanja nije izmenjen.

**Klase:** Navedene su sve klase i njihove značajnije metode:

Algorithm – osnovna klasa koju nasledjuju Minimax i AlphaBeta.

public final int[] search(Heaps heaps);

protected abstract int searchImpl(Heaps heaps);

AlphaBeta – nasleđuje Algorithm i implementira minimax sa alfa-beta odsecanjem.

protected abstract int searchImpl(Heaps heaps);

private int alphabeta(Node node, int currentDepth, int alpha, int beta, boolean maximizing);

BetterStaticFunction – nasleđuje StaticFunction i realizuje njenu najvažniju metodu (evaluate). Omogućava bolju procenu situacije u igri u odnosu na SimpleStaticFunction.

Commands – predstavlja komandnu tablu na kojoj se izdaju naređenja i vide podaci relevantni za igru (a da nije u pitanju grafički prikaz).

public synchronized int[] getCommandParameters() throws InterruptedException;

Computer – nasleđuje Player i realizuje njenu jedinu apstraktnu metodu. To je igrač-računar.

Game – nit (nasleđuje Thread) koja predstavlja igru, odlučuje čiji je potez i izdaje zahtev za izvođenje poteza, odlučuje pobednika.

public void run();

HeapException – nasleđuje Exception.

Heaps – glavna klasa koja predstavlja same stubove (gomile) i žetone.

public Heaps(int maxCheckersPerHeap, int[] heaps) throws IllegalArgumentException, HeapException;

public static void checkParameters(int maxCheckersPerHeap, int[] heaps) throws IllegalArgumentException, HeapException;

public boolean removeCheckers(int heapNumber, int checkers) throws IllegalArgumentException, HeapException;

public boolean legalMoveExists();

public static boolean firstMoveExists(int maxCheckersPerHeap, int[] heaps);

private static boolean heapStatesLegalAfterRemoval(int[] heaps, int heapNumber, int checkersToRemove);

public boolean moveIsLegal(int heapNumber, int checkersToRemove) throws IllegalArgumentException, HeapException;

public int[] allowedToRemoveFromHeap(int i);

public Heaps copy();

HeapsGraphics – nasleđuje Heaps i poseduje objekte tipa Canvas za vizuelno predstavljanje igre.

public HeapsGraphics(int maxCheckersPerHeap, int[] heaps, Panel p) throws IllegalArgumentException, HeapException;

Human – nasleđuje Player i realizuje njenu jedinu apstraktnu metodu. To je igrač-čovek.

MainFrame – nasleđuje Frame i to je glavna klasa za GUI. Poseduje main metodu.

public static void main(String[] args);

Minimax – nasleđuje Algorithm i implementira minimax.

protected abstract int searchImpl(Heaps heaps);

private int minimax(Node node, int currentDepth, boolean maximizing);

Node – čvor u grananju minimax algoritma. Čuva podatke o predviđenoj situaciji i potezu koji je doveo do nje i omogućava ekspanziju čvora.

public int getRemovedFrom();

public int getCheckersRemoved() ;

public List<Node> expand();

Player – osnovna klasa koja predstavlja igrača. Nju nasleđuju Computer i Human i implementiraju njenu apstraktnu metodu.

protected abstract int[] selectHeapAndAmount() throws InterruptedException;

public void playMove() throws InterruptedException;

SimpleStaticFunction – nasleđuje StaticFunction. Metoda koju implementira radi kao što je navedeno u tekstu zadatka.

StaticFunction – osnovna klasa za statičke funkcije.

public abstract int evaluate(Heaps heaps);