

API

Interfejs `morris.Player`

```
package morris;

public interface Player {
    public Move getMove(Board board, Move move, long timeForMove);
    public String getMessage();
}
```

`getMove()` – treba da vrati sledeći potez, koji je odigrao AI program.

`getMessage()` – treba da vrati poruku koja će biti prikazana u delu za informacije glavnog prozora igre. Treba da opiše odigrani potez. (Proizvoljno odrediti nivo detalja, ali da bude dovoljno informativno).

Klasa `morris.Move`

Predstavlja podatak o potezu.

```
public static final int PLACE = 0;
public static final int MOVE = 1;
public static final int MILL = 2;
public int type;
public int player;
public int piece, dest;
```

Sva polja klase su *public* da bi se uštedelo vreme prilikom pristupa poljima.

Statička polja služe da opišu potez koji treba da bude odigran. Po pravilima igre, potez može da bude postavljanje figure na tablu (prva faza), pomeranje figure (druga faza) i mica, odnosno uklanjanje protivničke figure (i prva i druga faza).

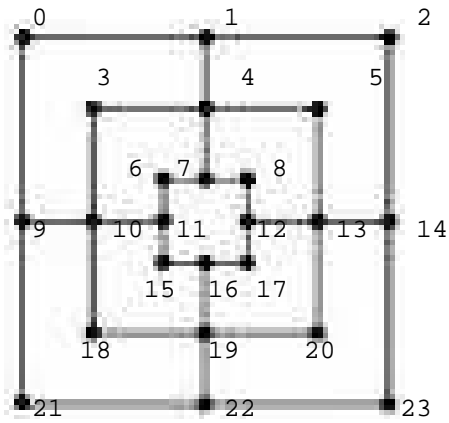
`type` – vrsta sledećeg poteza iz opsega koji je opisan u prethodnom pasusu.

`player` – „boja“ igrača koji je sledeći na potezu. 0 za crvenog igrača, 1 je plavog.

`piece, dest` – kada je potez `PLACE`, `piece` predstavlja polje na koje se postavlja figura, `dest` se ne koristi. Ako je potez `MOVE`, `piece` predstavlja polje na kojem se nalazi figura koja se pomera, `dest` je destinacija pomeranja. Za tip `MILL`, `piece` predstavlja polje na kojem se nalazi protivnikova figura koji želimo da uklonimo, `dest` se ne koristi.

Klasa `morris.Board`

Predstavlja trenutno stanje table u svakom potezu. Za predstavljanje table, koristi se niz celih brojeva koji označavaju redni broj polja na tabli. Polja su numerisana na sledeći način:



Konstante:

```
public static final int RED = 0;
public static final int BLUE = 1;
public static final int EMPTY = -1;
```

služe za označavanje zauzetosti svakog polja. EMPTY pokazuje da je polje prazno, RED označava da je polje zauzeto crvenom figurom, a BLUE da je polje zauzeto plavom figurom.

```
public int[] fields;
public int redHolding, blueHolding;
public int player;
```

`[]fields` – predstavlja sva polja table.

`redHolding, blueHolding` – broj figura koje još uvek nisu postavljene na tablu crvenog, odnosno plavog igrača.

`player` – boja igrača koji je trenutno na potezu

`public int[] getAdjacent(int square)` – rezultat je niz koji sadrži redne brojeve susednih polja, polju koje je prosleđeno kao argument metode.

`public boolean inMill(int square, int color)` – rezultat je `true` ako je figura boje `color` na polju `square` u mici.

`public int countPieces(int color)` – rezultat je broj figura boje `color` na tabli.

`public int getWinner()` – rezultat je boja pobednika (0 za RED, 1 za BLUE), ili -1 ako je igra nije završena.

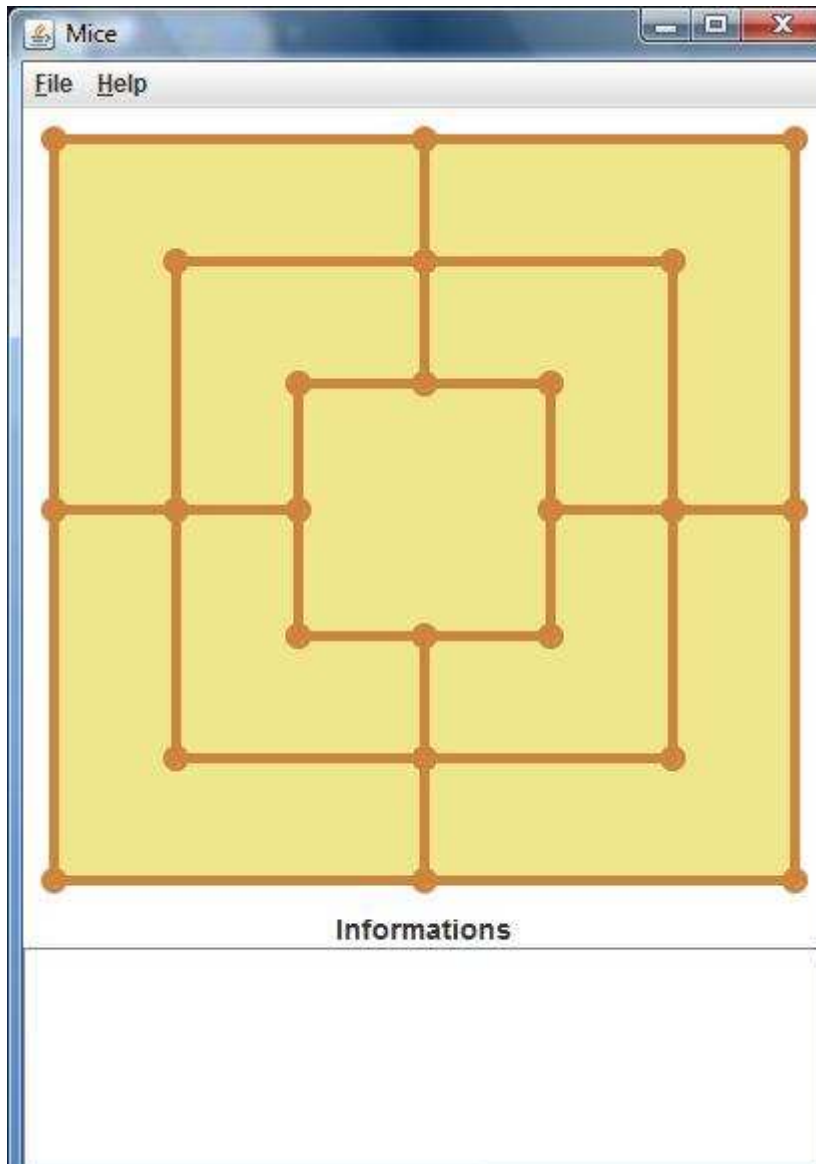
Uputstvo za pokretanje

Da bi AI igrač mogao da učestvuje u igri, implementacija interfejsa *Player* mora da bude dostupna u classpath-u prilikom pokretanja igre. Igra se pokreće pokretanjem .jar fajla *mice.jar*.

Ako je implementacija AI igrača *MyPlayer.java* u paketu *myPlayer*, i paket je upakovan u arhivu *myJar.jar*, igru možete pokrenuti na sledeći način (ako se arhive *mice.jar* i *myJar.jar* nalaze na disku E):

```
C:\Users\John Smith\Desktop>java -classpath E:\mice.jar;E:\myJar.jar morris.Game
```

Po pokretanju igre, pojaviće se glavni prozor:



Igra počinje tek kada se iz menija *File* izabere opcija *New Game*, nakon čega se pojavljuje sledeći prozor:



U polje *Time limit (s)* se unosi dozvoljeno vreme za jedan potez AI igrača. Ako se ništa ne unese, podrazumevana vrednost je 10 sekundi. Postavlja se svaki put kada se izabere nova partija.

Polja *Red Player* treba da sadrži ime AI igrača, tj. ime klase koja je implementirala interfejs *Player*. Ime treba da bude puno ime java klase, uključujući i ime paketa. Ako je ime klase *MyPlayer* i ako se klasa nalazi u paketu *myPackage*, puno ime klase je *myPackage.MyPlayer*. Klasa *MyPlayer* mora biti u classpath-u prilikom pokretanja igre. Da bi se potvrdilo da je igrač AI, potrebno je štiklirati odgovarajući checkbox pored imena klase. Sve navedeno za *Red Player*, važi i za *Blue Player*.

Ukoliko se ne štiklira odgovarajuće checkbox polje, tip tog igrača će biti *Human*, odnosno Čovek. Igru mogu igrati dva Human igrača, jedan Human i jedan AI, i dva AI igrača.

Novu igru je moguće pokrenuti u bilo kom trenutku, pri čemu se tekuća prekida.

Primer AI igrača

Primer implementacije AI igrača dat je u arhivi *myJar.jar*.