

Amorph Othello Kötelező Program

A feladat az othello (reversi) jétékot játszó ágens implementálása.

A játék során adott egy négyzet alakú tábla, amelyen a két játékos felváltva helyezi el a korongjait a megfelelő üres mezőkön. Egy korong lehelyezése esetén a már meglévő saját korongok által közrefogott (folytonos egyvonalba eső, vízszintes, függőleges vagy átlós) az ellenfélhez tartozó korongok saját korongokká válnak. A cél, hogy minnél több korongunk legyen a táblán. Korong csak olyan helyre helyezhető el, ahol annak hatására átfordul az ellenfél tulajdonában lévő korong. A játék véget ér, ha már nem helyezhető el újabb korong a táblán (betelik a tábla vagy a következő játékos nem tud átfordítani egy korongot sem).

További információ a játék Wikipédi oldalán magyarul, angolul

Szabályok

Az eredeti játékhoz képest a következő módosításokat alkalmazzuk:

- 10x10-es a pálya mérete
- vannak a pálya szélein **lyukak**, ahova nem helyezhető el korong

Keretrendszer

A megoldást **Java** nyelven kell megvalósítani, egy általunk definiált absztrakt osztály megvalósítása által (részletek később). Az ehhez szükséges keretrendszer a coospace felületről letölthető, használatát pedig a továbbiakban részletezzük.

Szükséges a **Java sdk 6** vagy újabb telepítése a fordításhoz és a kiértékeléshez.

Játék indítása vizualizációs felülettel

```
java -jar game_engine.jar 10 game.oth.OthelloGame 1234567890 10 3
2000 game.oth.players.GreedyPlayer game.oth.players.GreedyPlayer
```

Paraméterek:

- 10: fps / debug paraméter, ami a játék illetve a megjelenítés sebességét állítja; speciális értéke a 0 (nulla), ami a kiértékelési üzemmód, itt megjelenítés nélkül, maximális sebességgel történik a futtatás
- **game.oth.OthelloGame**: Játék osztály, ez mindig ugyanez kell legyen
- 1234567890: random seed
- 10: a tábla mérete (10x10)
- 3: táblán lévő **lyukak** generálásához használt iterációk száma
- 2000: maximális gondolkodási idő (ms)
- **game.oth.players.GreedyPlayer** **game.oth.players.RandomPlayer**: játékos osztályok:
 - **game.oth.players.GreedyPlayer**: mohó megvalósítás
 - **game.oth.players.RandomPlayer**: véletlen cellát választó ágens

A kezdő tábla a következő (a lyukak a seed hatására változhatnak):

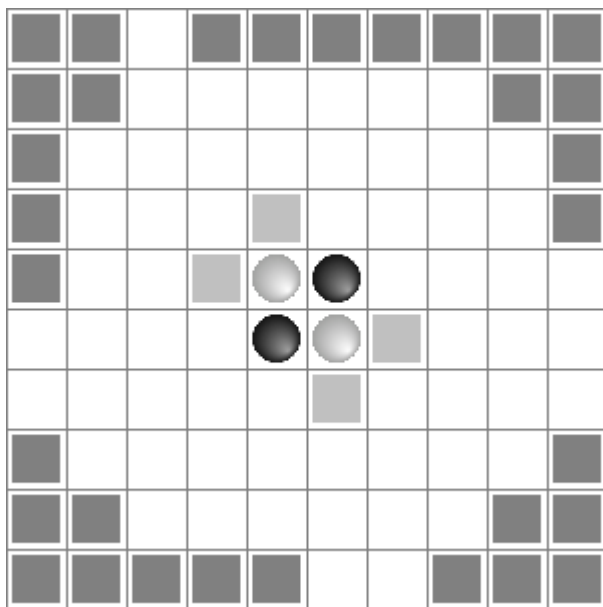
Konzol nézetben:

```

      0 1 2 3 4 5 6 7 8 9
    *-----*
0 |# # . # # # # # # /
1 |# # . . . . . # # /
2 |# . . . . . . . # /
3 |# . . . . . . . # /
4 |# . . . O X . . . |
5 |. . . . X O . . . |
6 |. . . . . . . . . |
7 |# . . . . . . . # /
8 |# # . . . . . # # /
9 |# # # # # . . # # /
    *-----*

```

Grafikus nézetben:



Saját ágens készítése:

- Hozzuk létre egy `SamplePlayer.java` állományt, a következő tartalommal:

```

import java.util.Random;

import game.oth.OthelloAction;
import game.oth.OthelloPlayer;

```

```

public class SamplePlayer extends OthelloPlayer {

    public SamplePlayer(int color, int[][] board, Random random) {
        super(color, board, random);
    }

    @Override
    public OthelloAction getAction(OthelloAction prevAction, long[] remainingTimes) {
        int i = random.nextInt(board.length);
        int j = random.nextInt(board[i].length);
        return new OthelloAction(i, j);
    }
}

```

- Fordítsuk le a file-t: `javac -cp game_engine.jar SamplePlayer.java`
- Értékeljük ki a GreedyPlayer ellen: `java -jar game_engine.jar 0 game.oth.OthelloGame 1234567890 10 3 2000 game.oth.players.GreedyPlayer SamplePlayer`
- Kimenet az output csatornán:

```

logfile: gameplay_XXXXXXXXX.data
0 game.oth.players.GreedyPlayer 0 4.0 2000000000
1 SamplePlayer 1 -1.0 2000000000

```

- Kimenet error csatornán (mivel ebben a játékban a véletlenszerű lépés valószínűleg nem helyes):

```
INVALID ACTION: 12 2 OF: SamplePlayer 1
```

- A játék kézi játékos módja a GreedyPlayer ellen: `java -jar game_engine.jar 1 game.oth.OthelloGame 1234567890 10 3 2000 game.oth.players.GreedyPlayer game.oth.players.HumanPlayer`
- Egy játék visszanezése a logfile alapján: `java -jar game_engine.jar 1 gameplay_XXXXXXXXX.data`

Kiértékelés

A feladat beadása a coospace-en keresztül történik majd, a beadáshoz egyetlen java file feltöltése szükséges ami a fentiek szerint a stratégia megvalósítását tartalmazza. A keretrendszer nem használ véletlen döntéseket, tehát a random seed kizárólag a saját megvalósítás esetleges véletlen döntéseit befolyásolja.

Korlátok, határidők, követelmények

- Maximális gondolkodási idő: 2000 ms
- Maximálisan felhasználható memória: 2G
- A teljesítéshez legalább 8/10 arányban le kell győzni a GreedyPlayer-t
- 10 próbálkozás áll rendelkezésre

- Beküldési határidő: 2021. december 6. 23:59

A fenti korlátoknak megfelelő futtatási paraméterezés lehet a következő: `java -Xmx2G -jar game_engine.jar 0 game.oth.OthelloGame 1234567890 10 3 2000 game.oth.players.GreedyPlayer SamplePlayer`

A kiértékelés során 5 véletlen inicializáció lesz használva (random seed), mindegyik esetén két-két futás (egyikben a feltöltött ágens kezdi a játékot, a másikban az ellenfél).

További követelmények a megoldással szemben

A megoldásnak saját munkának kell lennie. Konzultáció, közös ötletelés megengedett, de a megvalósítás önálló kell legyen. A megoldást tartalmazó forráskódnak minden körülmények között ki kell elégítenie a következő követelményeket:

- A megoldás nem állhat előre legyártott lépéssorozat visszajátszásából
- A forráskódot `Agent.java` néven kell feltölteni
- A feltöltött forráskódnak le kell fordulnia és hibamentesen le kell futnia
- A feltöltött fájlt az `iconv -f ascii -c` paranccsal ASCII-vé konvertáljuk a fordítás előtt. Emiatt az ékezetes betűk és minden más nem-ascii karakter ki lesznek vágva, tehát jobb ezeket eleve kerülni. Javasolt az UTF8 kódolás.
- A megoldást tartalmazó osztálynak a `game.oth.OthelloPlayer`-ből kell származnia, ami a keretrendszer részét képezi
- Véletlen számok használata esetén kizárólag az örökölt `random` mezőt szabad használni, és a seed átállítása tilos
- A megoldást tartalmazó osztálynak részletes magyar osztálydokumentációt kell tartalmaznia, javadoc formátumban, illetve a kód dokumentációja is magyar kell, hogy legyen
- A kód nem használhat a keretrendszeren kívül semmilyen más osztálykönyvtárat (természetesen a JDK osztályain kívül)
- A megoldást tartalmazó osztály nem lehet csomagban
- A megoldásban nem lehet képernyőre írás
- A megoldás nem nyithat meg fájlt, nem indíthat új szálakat
- Az implementált metódusoknak minden esetben vissza kell térniük (nem szerepelhet benne `exit` hívás például)
- A forráskód első sorában megadható egy nicknév és egy értesítési emailcím a következő formátumban:

///Nicknevem,Vezeteknev.Keresztnev@stud.u-szeged.hu

Ha meg van adva, a nicknév jelenik meg a ranglistában, egyébként pedig a Neptun azonosító. Ha meg van adva emailcím, egy tájékoztató emailt küldünk az ágens kiértékelése után, mely a `{DATE}_out.txt` (a program kimenete), `{DATE}_log.txt` (játék logja), és `meta.txt` (eddig beküldések státusza) állományok elérhetőségét tartalmazza. Emailcím megadása nélkül is megtekinthető a ranglistában a pontszám és a játék visszajátszható. Lehetőség van arra is, hogy nicknevet ne, csak emailt adjunk meg, ebben az esetben az első paramétert üresen kell hagyni, majd a vessző után az emailcímet megadni:

///,Vezeteknev.Keresztnev@stud.u-szeged.hu

Az email értesítő esetén érdemes hivatalos egyetemi emailcímet használni. (A gmail pl. spam folderbe teheti az értesítést.)

- Fenntartjuk a jogot, hogy bármilyen, fent nem listázott, de az etika szabályai ellen történő vétséget szankcionáljunk; ha bárkinek kételyei vannak egy konkrét dologgal kapcsolatban, inkább kérdezzen rá időben