

## Amőba kötelező program

A feladat az amőba (gomoku) játékot játszó ágens implementálása.

A játék során adott egy négyzet alakú tábla, amelyen a két játékos felváltva helyezi el a szimbólumait az üres mezőkön. A cél a játékosok számára, hogy sikerüljön összefüggően, megszakítás nélkül (egy sorban, oszlopban vagy átlósan), egymás mellé legalább 5 azonos szimbólumot lerakni. Akinek ez először sikerül, az megnyeri a játékot.

További információ a játék Wikipédi oldalán (magyarul, angolul).

## Szabályok

Változások az `alap` szabályokhoz képest:

- A tábla mérete 15x15, végtelenített (tórusz) alakú
- Nincs második szakasz, a szimbólumok addig rakhatóak le, amíg be nem telik a tábla, vagy az egyik játékos nem nyer
- Betelt tábla esetén döntetlen a végeredmény
- A táblán vannak **lyukak**, ahova nem lehet lerakni szimbólumot senkinek sem

## Keretrendszer

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

Szükséges a Java sdk 8 vagy újabb telepítése a fordításhoz és a kiértékeléshez. Továbbá a kényelmesebb tesztelés érdekében hasznos lehet JavaFX telepítése (pl. Oracle Java 8 sdk használatával).

## Játék indítása vizualizációs felülettel

```
java -jar game_engine.jar 10 game.gmk.GomokuGame 1234567890 15 15  
0.1 2000 game.gmk.players.GreedyPlayer game.gmk.players.RandomPlayer
```

Megfelelően telepített JavaFX hiányában csak konzolos megjelenítés lehetséges.

## 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.gmk.GomokuGame`: Játék osztály, ez mindig ugyanez kell legyen
- 1234567890: random seed
- 15 15: a tábla mérete (15x15)
- 0.1: táblán lévő lyukak aránya

- 2000: maximális gondolkodási idő (ms)
- `game.gmk.players.GreedyPlayer` `game.gmk.players.RandomPlayer`: játékos osztályok:
  - `game.gmk.players.GreedyPlayer`: mohó megvalósítás
  - `game.gmk.players.RandomPlayer`: véletlen cellát választó ágens

Az fps (frame per sec) paraméter ha nullánál nagyobb, és egyúttal a JavaFX is rendesen telepítve van, akkor a vizualizációs felület is megjelenik. A `game.gmk.players.HumanPlayer` mint játékos megadásakor kézzel játszható a játék.

### Játék indítása vizualizációs felület nélkül

```
java -jar game_engine.jar 0 game.gmk.GomokuGame 1234567890 15 15
0.1 2000 game.gmk.players.GreedyPlayer game.gmk.players.RandomPlayer
```

- a konzol error csatornáján esetleges hibaüzenetek íródhatnak ki.
- a konzol output csatornáján pedig: pl.

```
logfile: gameplay_xxxxxxxx.data
0 game.gmk.players.GreedyPlayer 0 190.0 2000000000
1 game.gmk.players.RandomPlayer 1 -190.0 2000000000
```

ahol:

- logfile: `gameplay_xxxxxxxx.data`: a játék visszajátszására szolgáló log file a következő sorokban:
- 0/1: játékos azonosítója
- `game.gmk.players.GreedyPlayer`/`game.gmk.players.RandomPlayer`: játékos osztály
- 0/1: játékos azonosító / szín
- 190.0/-190.0: üresen maradt mezők száma (a negatív előjel jelöli a vesztes játékost)
- 2000000000/2000000000: megmaradt gondolkodási idő (ns)
- egy játék visszajátszása logfile alapján: `java -jar game_engine.jar 10 gameplay_xxxxxxxx.data`

### Saját ágens fejlesztése

Véletlenszerűen választott cella jelölésére szolgáló megoldás. Figyelem: nincs vizsgálva, hogy a cella üres-e!!!

Hozzuk létre a `SamplePlayer.java` fület a következő tartalommal:

```
import java.util.Random;

import game.gmk.GomokuAction;
```

```

import game.gmk.GomokuPlayer;

public class SamplePlayer extends GomokuPlayer {

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

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

• Fordítsuk le az osztályt: javac -cp game_engine.jar SamplePlayer.java
• Értékeljük ki a GreedyPlayer ellen (vizualizációval 1fps): java -jar
  game_engine.jar 1 game.gmk.GomokuGame 1234567890 15 15 0.1
  2000 game.gmk.players.GreedyPlayer SamplePlayer

```

## 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.

## 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.gmk.GomokuPlayer`-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 szálát
- 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

### Korlátok és határidők

- 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: 2020. december 7. 23:59