

Snake Kötelező Program

A feladat a Snake (kígyó) játékot játszó ágens implementálása.

A játék során egy kígyót kell irányítani egy négyzetrács szerű táblán az azon elhelyezett étel felé. Az étel felvételével a kígyó mérete egyel növekszik. A cél minél több étel felvétele anélkül, hogy a kígyó a pálya falába vagy önmagába ütközzön.

További információ a játék Wikipédia oldalán [angolul](#)

Szabályok

A játék kezdetekor a kígyó feje a pálya közepén helyezkedik el, a teste pedig attól jobbra még 2 egységnyi és balra indul el. A kígyó folyamatosan egyesével halad a négyzetrácson. Az irányítás a mozgás irányának módosításával oldható meg, a lehetséges irányok: bal, fel, jobb, le. Minden lépés előtt módosítható a kígyó fejének iránya valamelyik a jelenlegi irányra merőleges irányban (tehát ha a kígyó balra megy, akkor a lehetséges irányváltások: fel vagy le), ellenkező esetben (nem megfelelő módosítás esetén) az iránya nem változik.

A játék véget ér, ha a kígyó a falnak (pálya széle) vagy önmagának ütközik, ha lejár a játékos gondolkodási ideje (lépésenkénti eltel idők összege elér egy előre definiált felső korlátot) vagy kellően sokáig 'tétlenkedik' (nem vesz fel egyetlen ételt sem lépéseken keresztül).

Pontozás

A játékos pontszáma 0-ról indul és lépésenként csökken 1-el, valamint minden étel elfogyasztása esetén növekszik 100-al.

Célok és Paraméterek

- Cél: minél több étel felvétele, minél kevesebb lépéssel
- Pálya mérete: 15x25, azaz 15 sor és 25 oszlop
- Lehetséges akciók:
 - `SnakeGame.LEFT`: bal
 - `SnakeGame.UP`: fel
 - `SnakeGame.RIGHT`: jobb
 - `SnakeGame.DOWN`: le
- Maximális tétlen lépésszám: 2x15x25, a pálya méretének duplája
- Pálya lehetséges elemei:
 - `SnakeGame.EMPTY`: üres (space)
 - `SnakeGame.FOOD`: étel (0)
 - `SnakeGame.SNAKE`: kígyó (#)
 - A kígyó feje külön karakterrel van jelölve (@), de ez a pályáról nem kérhető le, hanem csak a `gameState`-ből: `gameState.snake.peekFirst()`

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 8 vagy újabb telepítése a fordításhoz és a kiértékeléshez.

Játék indítása vizualizációs felülettel

- Véletlenszerű irányt választó ágenssel: `java -jar game_engine.jar 10 game.snake.SnakeGame 1234567890 15 25 10000 game.snake.players.RandomPlayer`

- Mohó (az ételhez legközelebbi cellát eredményező) irányt választó ágenssel: `java -jar game_engine.jar 10 game.snake.SnakeGame 1234567890 15 25 10000 game.snake.players.GreedyPlayer`

Paraméterek:

- 10: debug paraméter (x frames/sec, 0: nincs gui, -x csak konzol kimenet)
- `game.snake.SnakeGame`: játék logikát megvalósító osztály
- 1234567890: random seed
- 15: pálya magassága
- 25: pálya szélessége
- 10000: rendelkezésre álló összidő (millisec)
- `game.snake.players.RandomPlayer`: kígyót vezérlő osztály

Saját ágens készítése:

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

```
import java.util.Random;

import game.snake.Direction;
import game.snake.SnakeGame;
import game.snake.SnakePlayer;
import game.snake.utils.SnakeGameState;

public class SamplePlayer extends SnakePlayer {

    public SamplePlayer(SnakeGameState gameState, int color, Random random) {
        super(gameState, color, random);
    }

    @Override
    public Direction getAction(long remainingTime) {
        Direction action = SnakeGame.DIRECTIONS[random.nextInt(SnakeGame.DIRECTIONS.length)];
        return action;
    }
}
```

- Fordítsuk le a file-t: `javac -cp game_engine.jar SamplePlayer.java`
- Értékeljük ki: `java -jar game_engine.jar 0 game.snake.SnakeGame 1234567890 15 25 10000 SamplePlayer`
- Kimenet az output csatornán:

```
logfile: gameplay_xxxxxxxx.data
0 SamplePlayer -80.0 10000000000
```

- Egy játék visszanezése a logfile alapján (25fps): `java -jar game_engine.jar 25 gameplay_xxxxxxxx.data`

A játék kimenete:

- 0: játékos azonosító
- `SamplePlayer`: játékos implementáló osztály neve
- -80.0: elért pontszám
- 10000000000: megmaradt gondolkodási idő nanomásodpercben

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 használ véletlen döntéseket,

tehát a random seed a saját megvalósítás esetleges véletlen döntéseit befolyásolja és attól függetlenül a keretrendszerét is.

Korlátok, határidők, követelmények

- Maximális gondolkodási idő: 10000 ms
- Maximálisan felhasználható memória: 2G
- A teljesítéshez legalább 3000 pontot kell elérni a 10 játékból legalább 8 esetben
- 10 próbálkozás áll rendelkezésre
- Beküldési határidő: 2024. december 2. 23:59

A fenti korlátoknak megfelelő futtatási paraméterezés lehet a következő: `java -Xmx2G -jar game_engine.jar 0 game.snake.SnakeGame 1234567890 15 25 10000 SamplePlayer`

A kiértékelés során 10 véletlen inicializáció lesz használva (random seed).

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.snake.SnakePlayer`-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.