

# RaceTrack Kötelező Program

A feladat a RaceTrack játékot játszó ágens implementálása.

A játék során egy autót kell a versenypályán irányítani a kezdő pozícióból a célt jelző területre. A versenypályát egy téglalap alakú négyzetekre osztott mátrix reprezentálja, az irányítandó autó pedig ennek egy celláján helyezkedik el. A jármű mozgását a hozzá tartozó sebességvektor reprezentálja, amely lépésről lépésre történő módosításával valósul meg az irányítás. A pálya területét falak határolják, ezek között kell megoldani a navigálást. A feladat az autót irányító ágens implementálása.

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

## Szabályok

Az autó kezdő pozíciója a pálya bal felső részén, a cél pedig a jobb felső részén helyezkedik el. A pályán lévő falak labirintus szerűen helyezkednek el. A játékban használt koordináta rendszer 'mátrix címzés' jellegű, azaz a bal felső sarokban van a zérus. Az  $(i, j)$  mező pedig az  $i$ -edik sor  $j$ -edik oszlopában van. Az autó sebességvektora  $(v_i, v_j)$  egész értékeket tartalmazó vektor, ahol a  $v_i$  az  $i$ , a  $v_j$  a  $j$  indexhez tartozó sebességérték. Az autót irányító ágens iterációnként változtathatja a sebességvektor koordinátáit  $\{-1, 0, 1\}$  értékekkel, azaz csökkentheti, tarthatja vagy növelheti a sebességvektor értékét. Az autó pozíciója minden iterációban a sebességvektorának megfelelően változik meg:  $i = i + v_i$ ,  $j = j + v_j$ , kivéve ha ezen a pozíción fal található. Ebben az esetben az autó a fal melletti helyre kerül.

A játékos pontszáma iterációnként növekszik eggyel. A pályán elhelyezett érmék felvételével csökken a pontszám az érme értékével. A cél minél kisebb költséggel eljutni a cél területre.

Érmék felvétele azok feletti elhaladással lehetséges, azaz az autó pozíciója és a sebességvektora által meghatározott szakasz alatti érmék lesznek számításba véve.

A játékos kezdeti pontszáma a kezdőponttól a célig vezető legrövidebb út hossza negatív előjellel. Tehát, ha a játékos pontosan követi ezt az utat, akkor a pontszáma 0 lesz. Az érmék értéke ettől a legrövidebb úttól vette minimális manhattan távolság háromszorosa.

## 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

```
java -jar game_engine.jar 10 game.racetrack.RaceTrackGame 11 27 5 0.1 10 1234567890 1000
game.racetrack.players.RandomPlayer
```

## 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.racetrack.RaceTrackGame`: Játék osztály, ez mindig ugyanez kell legyen
- 11: a pálya magassága
- 27: a pálya szélessége
- 5: a pálya vonalának és a falak szélessége, azaz a pálya tényleges mérete: 55x135
- 1234567890: random seed
- 1000: maximális gondolkodási idő (ms)
- `game.racetrack.players.RandomPlayer`: az autót irányító véletlen irányokat választó ágens

### Saját ágens készítése:

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

```
import java.util.Random;

import game.racetrack.Direction;
import game.racetrack.RaceTrackGame;
import game.racetrack.RaceTrackPlayer;
import game.racetrack.utils.Coin;
import game.racetrack.utils.PlayerState;

public class SamplePlayer extends RaceTrackPlayer {

    public SamplePlayer(PlayerState state, Random random, int[][] track, Coin[] coins, int color) {
        super(state, random, track, coins, color);
    }

    @Override
    public Direction getDirection(long remainingTime) {
        return RaceTrackGame.DIRECTIONS[random.nextInt(RaceTrackGame.DIRECTIONS.length)];
    }
}
```

- Fordítsuk le a file-t: `javac -cp game_engine.jar SamplePlayer.java`
- Értékeljük ki: `java -jar game_engine.jar 0 game.racetrack.RaceTrackGame 11 27 5 0.1 10 1234567890 1000 SamplePlayer`
- Kimenet az output csatornán:

```
logfile: gameplay_xxxxxxxx.data
0 X SamplePlayer p:(3, 126) v:(-3, -1) 3496.0 937500000
```

- Egy játék visszanezése a logfile alapján: `java -jar game_engine.jar 1 gameplay_xxxxxxxx.data`

A játék kimenete:

- 0: játékos index
- X: autó karakteres reprezentációja
- SamplePlayer: játékot irányító ágens
- p:(3, 126) v:(-3, -1): autó pozíció és sebességvektor
- 3496.0: játékos pontszáma
- 937500000: rendelkezésre álló idő (nanomásodperc)

### 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: 1000 ms
- Maximálisan felhasználható memória: 2G
- A teljesítéshez legalább 8/10 arányban negatív pontszámmal kell végezni
- 10 próbálkozás áll rendelkezésre
- Beküldési határidő: 2023. december 6. 23:59

A fenti korlátoknak megfelelő futtatási paraméterezés lehet a következő: `java -Xmx2G -jar game_engine.jar 0 game.racetrack.RaceTrackGame 11 27 5 0.1 10 1234567890 1000 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.racetrack.RaceTrackPlayer`-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