

Dokumentáció

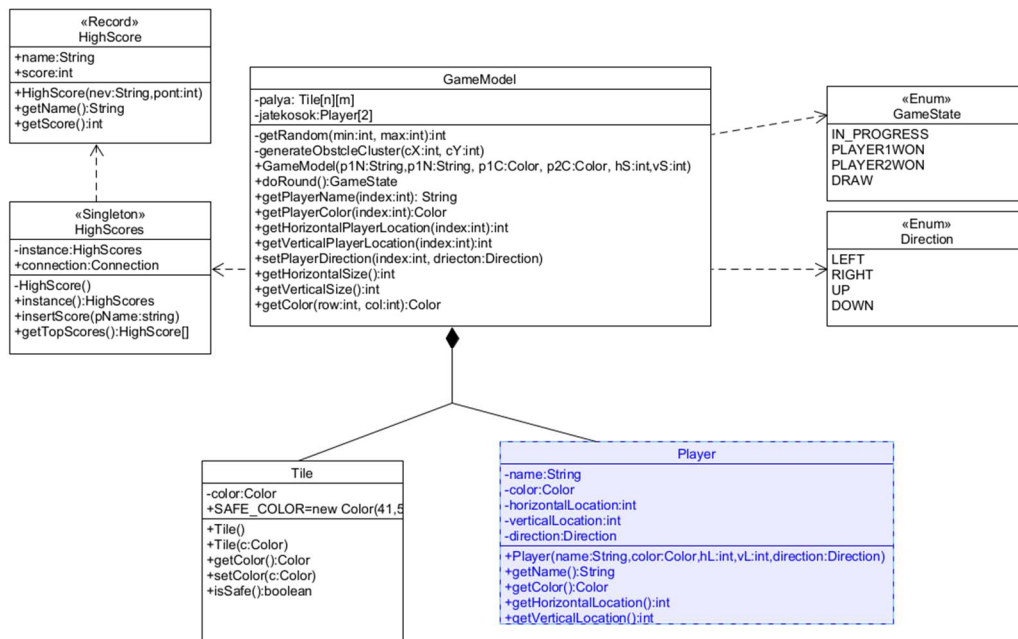
Feladat leírás:

4. Tron

Készítsünk programot, amellyel a Tronból ismert fény-motor párbajt játszhatjuk felülnézetből. Két játékos játszik egymás ellen egy-egy olyan motorral, amely fénycsíkot húz maga mögött a képernyőn. A motor minden másodpercben a legutoljára beállított irányba halad feltéve, hogy a játékos nem változtatja meg azt egy megfelelő billentyű lenyomásával. (WASD az első játékos, nyilak a második játékos.)

Az a játékos veszít, aki előbb neki ütközik a másik játékos fénycsíkjának vagy a képernyő szélének. A játék elején kérjük el a játékosok nevét és engedjük meg, hogy maguk válasszák ki a fényük színét. A játék végekor a győztes játékos eredményét növeljük meg az adatbázisban. Ha a játékos még nem található meg az adatbázisban, úgy szúrunk be egy új sort. Egy menüpontban legyen lehetőségünk a 10 legjobb eredménnyel rendelkező játékost megtekinteni, az elért pontszámukkal, továbbá lehessen bármikor új játékot indítani egy másik menüből.

UML diagram:



Metódusok rövid leírása:

Direction:

Az irányítás konstansok segítségével történő leírása a játékosok mozgásának irányát reprezentálja. A lehetséges értékek a következők: FEL (UP), BALRA (LEFT), LE (DOWN), JOBBRA (RIGHT).

Player:

A játékosokat a nevükkel, színükkel, vízszintes és függőleges pozíciójukkal, valamint mozgásuk irányával jelenítjük meg. A játékosok attribútumai lekérdezhetőek, és képesek irányt váltani, valamint a jelenlegi irányban haladni.

Tile:

A Tile osztály egy játékterület egy celláját reprezentálja, és egyetlen adattaggal rendelkezik, ami a cella színét tárolja. A cella biztonságos vagy nem biztonságos állapotát ennek a színnek az alapján határozzuk meg. A biztonságos cellák színe az (41, 53, 66) RGB kóddal van meghatározva, és ezt a színt egy osztályszintű konstansban, SAFE_COLOR néven tároljuk. Az összes egyéb cellát nem biztonságosnak tekintjük.

GameModel:

A GameModel egy játékmodellt reprezentál, amely tárolja a játékosokat, a játékteret, és egy adatbázissal kommunikáló objektumot. A modell előkészíti a játékteret a következő algoritmus alapján:

1. Kiválaszt egy koordinátát a térkép bal felső negyedéből, beleértve a két negyed közötti cellákat is.
2. Előállít egy gyűjteményt, amely tartalmazza a kiválasztott koordináta körüli és magát a koordinátát.
3. Összekeveri a gyűjteményt.
4. Generál egy $n \in [3..9]$ számot, majd kiválasztja a gyűjtemény első n elemét, a többit eldobja.
5. Színezi feketére a játékteret azokon a koordinátákon, amelyek a gyűjteményben szereplők és azok tükrözött pontjain.
6. Ismétli a 2-5. lépéseket egy jobb felső negyedbeli koordinátával.
7. A játékosok körüli 3×3 -as területet biztonságossá teszi, kivéve a játékterület szélein lévő cellákat.

A GameModel osztály doRound() metódusa szimulálja a játék egy körét, és egy GameState-et ad vissza.

GameState: A GameState egy enumeráció, amely a játék aktuális állapotát reprezentálja. A lehetséges értékei: FOLYAMATBAN (IN_PROGRESS), JÁTÉKOS1_NYERT (PLAYER1 WON), JÁTÉKOS2_NYERT (PLAYER2 WON), DÖNTETLEN (DRAW).

HighScore: A HighScore egy rekord, amely egy nevet (String) és egy pontszámot (int) tárol.

HighScores: A HighScores osztály egy olyan osztály, amely kommunikál a játék adatbázisával.

GameView: A GameView egy olyan osztály, amely a JFrame osztályból származik, és felelős a játék ablakának megvalósításáért. Ezen osztály hozza létre az alkalmazás többi menüpontjának példányait, valamint végzi ezek közti váltások implementációját.

MainMenu: A MainMenu osztály egy JPanel, ami a játék főmenüjét alkotja. A főmenüből lehetőségünk van testreszabni a játékosokat, új játékot kezdeni, megadni a pályaméretet, megtekinteni az eredményeket és kilépni a játékból.

PlayerCustomizationPanel: A PlayerCustomizationPanel osztály példányai a játékosok testreszabási lehetőségeit valósítják meg. Ezen a panelen lehetőség van megadni a játékos nevét és fényküldőjének színét. A menüben továbbá megjelenik, hogy az adott játékos mely billentyűvel irányítható.

GameMenu: A GameMenu egy konkrét játékmenüt reprezentál. Amikor az alkalmazás nézetét erre a panelre vált, elindul egy időzítő, amely minden másodpercben előidézik egy eseményt, ami lépteti a játék állapotát. Az eseménykezelői a következők:

- Időzítő: Másodpercenként előreviszi a játék állapotát a `this.gameModel` objektum `doRound()` metódusának meghívásával. Ha ennek a metódusnak a hatására a játék véget ér, megjeleníti a győztest, és visszalép a főmenübe. Ha a `gameModel` nem tud kapcsolódni az adatbázishoz, hibaüzenetet jelenít meg.

- Escape megnyomása: Kilép a játékból és megállítja az időzítőt.

- W megnyomása: Az egyes játékos iránya "fel"-re vált.

- D megnyomása: Az egyes játékos iránya "jobbra"-ra vált.

- S megnyomása: Az egyes játékos iránya "le"-re vált.

- A megnyomása: Az egyes játékos iránya "balra"-ra vált.

- ↑ megnyomása: A kettes játékos iránya "fel"-re vált.

- → megnyomása: A kettes játékos iránya "jobbra"-ra vált.

- ↓ megnyomása: A kettes játékos iránya "le"-re vált.

- ← megnyomása: A kettes játékos iránya "bal"-ra vált.

HighScoreMenu: A HighScoreMenu egy JPanel, amely az adatbázisból lekérdezett adatokból álló táblázatot tartalmazza. Egyetlen eseménykezelője van, amely a füln található "vissza" gomb hatására a program nézetét visszalépteti a főmenüre.

Tesztelés:

Tevékenység	elvárt eredmény
adatbázis kapcsolat nem tud létrejönni	SQLException
játékosnevek megegyeznek	IllegalArgumentException
egyik játékosnév üres	IllegalArgumentException
játékosok színe megegyezik	IllegalArgumentException
egyik játékosok a játék számára fenntartott szint választ	IllegalArgumentException
pályaméret túl kicsi	IllegalArgumentException
pályaméret egyik oldala páros	IllegalArgumentException
játékparaméterek megfelelőek	létrejön a GameModel objektum
játék véget ér	adatbázis frissül