

Beadandó III

Magyar Tamás | RNYR2F

(4. Feladat)

Feladat Leírása

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.

Feladat Elemzése

A feladatban egy fix méretű pályán kell lerakni a két játékost és minden frame-ben egyel „előre” lép és a régi részeit megtartja. A játékosokhoz hozzá kell rendelni az irányító gombokat és találni kell egy módot arra, hogy begyűjtsd a nevét és a választott színét. A győztes ellenőrzésekor elég a fény „fejét” ellenőrizni, hogy tartalmazza-e a másik fénycsíkot, ha igen akkor veszített a játékos, illetve a pálya méretét kell ellenőrizni.

Esemény/Eseménykezelő

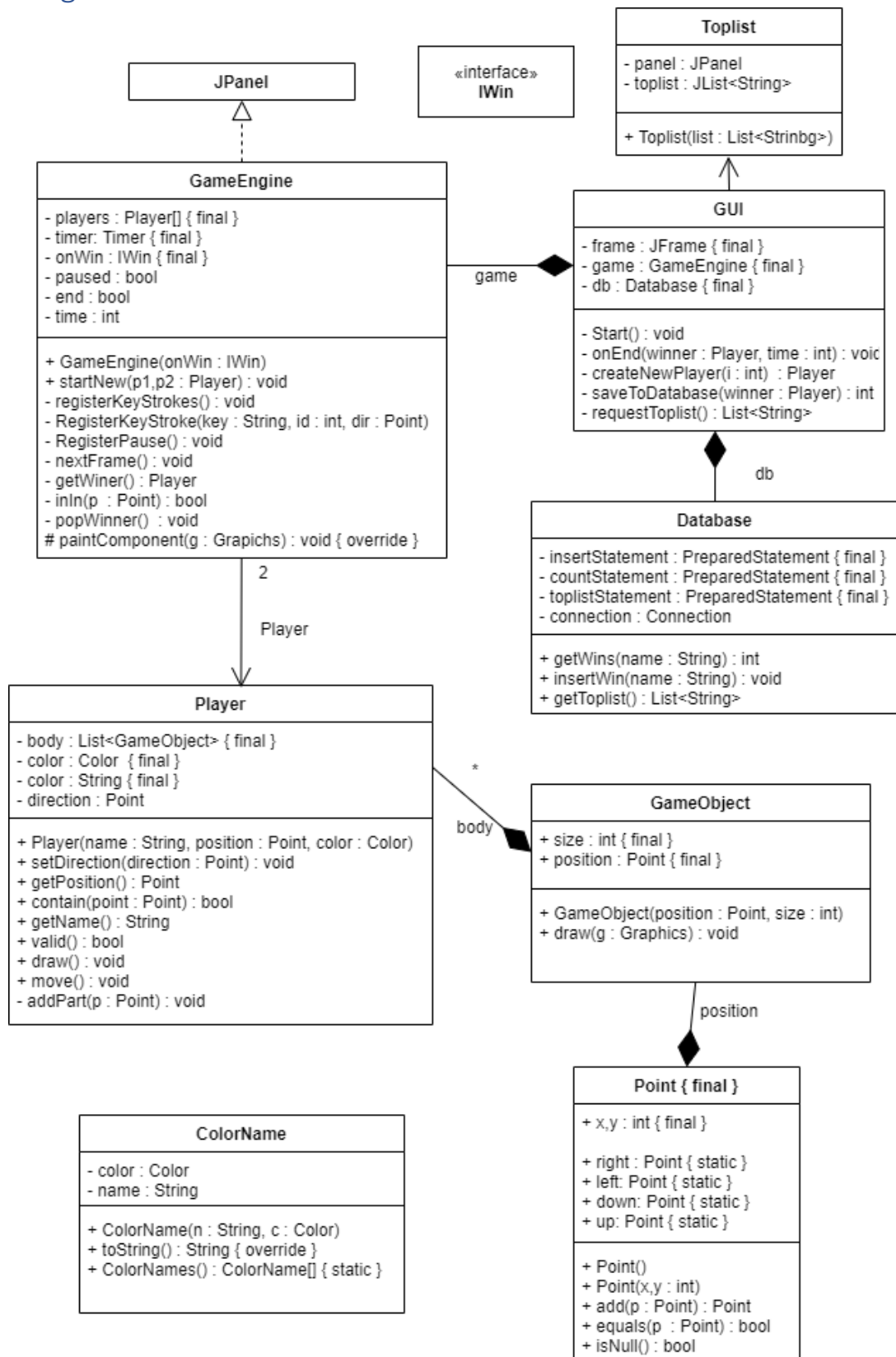
- Indítás -> új játék indítása
- Új játék -> név, szín bekérése (2x)
- Irányító gombok (aktív játék) (játékosunkként) -> motor irányba alítása
- Játék vége -> új játékos indítása **vagy** játék bezárása
- Bezárás -> játék bezárása

Implementációs részlet

A játék végét ellenőrző függvény azaz a `getWinner()` minkét játékosra ellenőrzi azt a feltelent hogy a Játékos „feje” a játéktérben van-e és ellenőrzi a másik játékos `body` -ját amit egy `List<GameObject>` reprezentál ezek a `GameObject` valósítják meg a fény részeit. Az ellenőrzéskor az ellenfél összes `GameObject` összehasonlításra kerül és ebből derül ki, hogy ütközött-e a játékos.

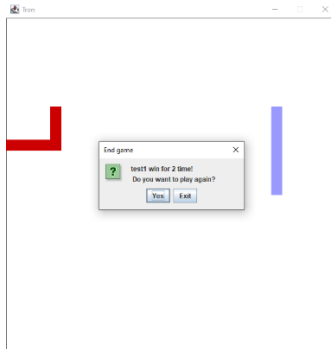
A `GameObject` gyakorlatilag egy `Point` határoz meg ez egy immutable osztály aminek van egy `x,y` mezője ami a pályán való elhelyezkedést határozza meg.

UML Diagram

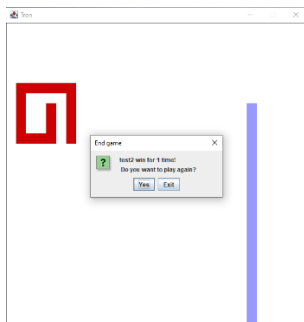


Tesztetek

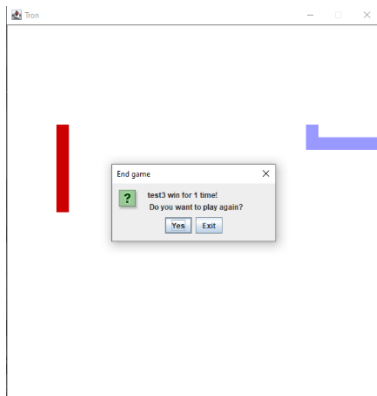
$X < 0$



$Y > size$



$X > size$



Normal

