Programozási technológiák 3. beadandó.

YKMM49

1. Feladat

**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

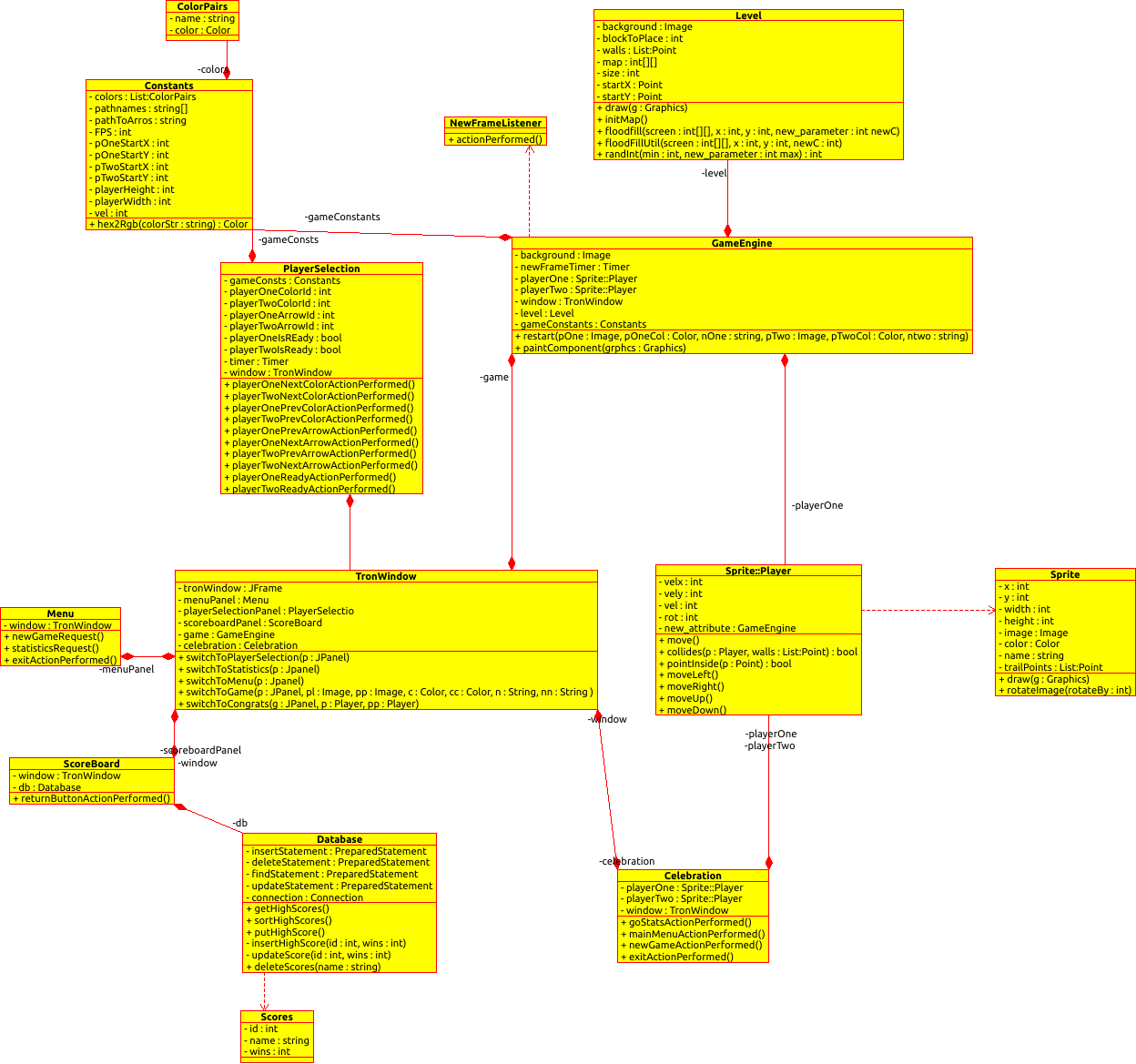
az adatbázisban. Ha a játékos még nem található meg az adatbázisban, úgy szúrjunk be

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.

2. UML

3. A program

A program egyetlen framre épül. Ez képes váltogatni a megjelenített paneleket, így lehet megoldani a navigálást. A menü a legfőbb pontokat mutatja meg játékindítás, statisztika megnázáse illetve az alkalmazás bezárása.

A játék indítása előtt a játékosoknak lehetőségük van rá, hogy kiválasszák a motorosuk színét és a motor stílusát. Ha kész vannak ezzel akkor egy visszaszámláló segítségével eljutnak a játékba.

A játéktér randomizált, tehát minden körben máshol jelennek meg akadályok a pályán. Az hogy a pálya játékképes-e egy floodFill nevű algoritmus segítségével ellnőrzöm.

A játékosok minden kört ugyan arról a pontrol kezdenek egymásra párhuyamosan. A játék addig tart amíg az egyik játékos neki nem megy valamelyik falnal,akadálynak vagy az ellenfél játékos vonalának.

Ha a játék véget ér a játékosok egz gratulációs ablakba jutnak, ahonnan el tudnak kalauzolni a főmenübe, a statisztikákhoz ahol a 10 legjobb játékost lehet megnézni, a játékos választóba vagz kiléphetnek a játékból.

4. Pálya generálás

A pálzagenerálás során úgy döntöttem, hogy 50x50-es négyzetekre bontom a játékteremet és ezekből a kockákból találomra választok néhányat. Amikor kiválasztottam a jövőbeli akadályaimat akkor készítek egy mátrixot ami a játékteret reprezentálja. A játékosok helyét -1 -re az akadályok és a falak helyét 1 -re és minden szabad helyet 0 -ra állítok. A pontok válogatásánál figzeltem hogy a játékos előtt maradt területet ne válasszam.

Amikor kész lett a mátrix kiválasztom rajta az első játékos előtti mezőt, hogy az összes onnan bejárható mező értékét állítsa át 2 -re.Miután lefutott ez a folyamat már csak azt kell megnéznem, hogy a második játékossal van-e olyan szomszédos terület aminek az értéke 2. Ha van ilyen akkor bejárható a terület ha nincs akkor újra generálom a területet addig, amíg nem járható be.

5. Esemény, eseménykezelő

A játék alatt történő eseménzeket az input map és az action map segítégével hajtom végre. Ha az első játékos a “wasd” gombok valamelyikét használja akkor a program meghívja az első játékos forgási függvényeit. Ezek a függvények ekkor átalakítják az x vagy az y tengelyen történő sebességet. Leginkább 0-ra vagy az alap sebességre és ellentettjére. Mivel képeket használok a motoroknak, így forgatáskor a képeket is forgatni kell. Ehhez folyton figyelni kell hogz most éppen milyen irányban áll és a szerint forgatni.

A második játékos ugyan ezt csinálja csak a nyilakkal.

6. Teszt

1. Minden GUI elem működése.

2. A játékos létrehozásakor minden adat elmentődik.

3. A játéktér létrejön és végig lehet játszani.

4. A játékosok mozognak.

5. A játék véget ér ha valamelyik játékos neki ütközik valami nem megfelelő dolognak.

6. A játékosok képesek mozgatni a figurákat.

7. Ha a játék véget ér a gratulációs ablak tölt be helyesen.

8. A statisztikák helyesen regisztrálódnak az adatbázisban és csak 10 sort mutat.

9. Az exit gombok bezárják az alkalmazást.