Beadandó feladat dokumentáció

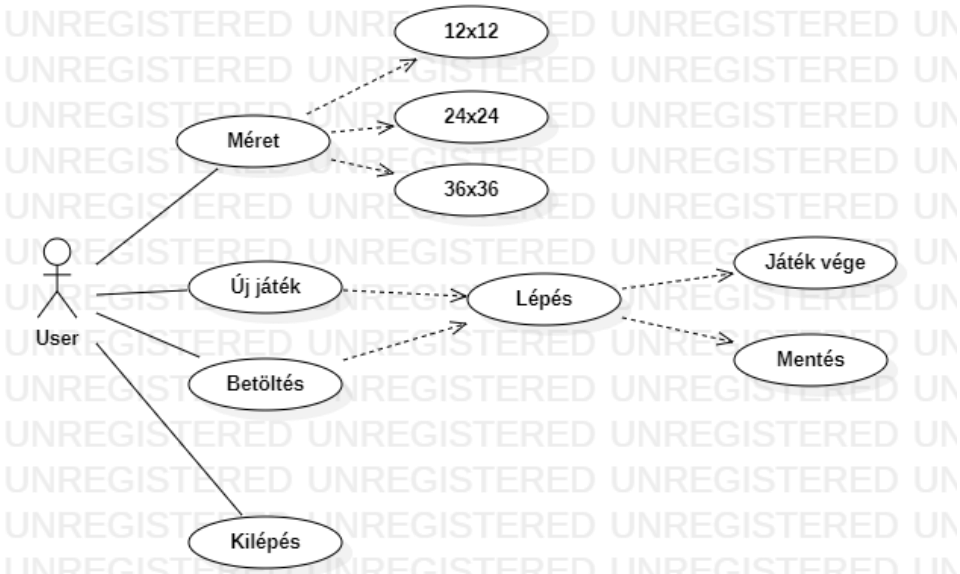
Készítette:
Hallgató: Szöllősi Ádám
Neptun-kód: DELG87
e-mail: szollosi.adam@icloud.com

Feladat:
Fénymotor párbaj Készítsük programot, amellyel a Tronból ismert fénymotor párbajt játszhatjuk. Adott egy $n \times n$ elemből álló játékpálya. A két játékos a bal, illetve jobb oldal közepén indul egy-egy fénymotorral, amely egyenesen halad (rögzített időközönként) a legutoljára beállított irányba (függőlegesen, vagy vízszintesen). A motorokkal lehetőség van balra, illetve jobbra fordulni. A fénymotor mozgás közben fénycsíkot húz, ami a játék végéig ott marad. Az a játékos veszít, aki előbb nekiütközik a másik játékos motorjának, bármelyikük fénycsíkjának vagy a pálya szélének. A program biztosítson lehetőséget új játék kezdésére a pályaméret megadásával (12 × 12, 24 × 24, 36 × 36), valamint játék szüneteltetésére (ekkor nem telik az idő, és nem mozognak a motorok). Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött. Ezen felül szüneteltetés alatt legyen lehetőség a játék elmentésére, valamint betöltésére.

Elemzés:

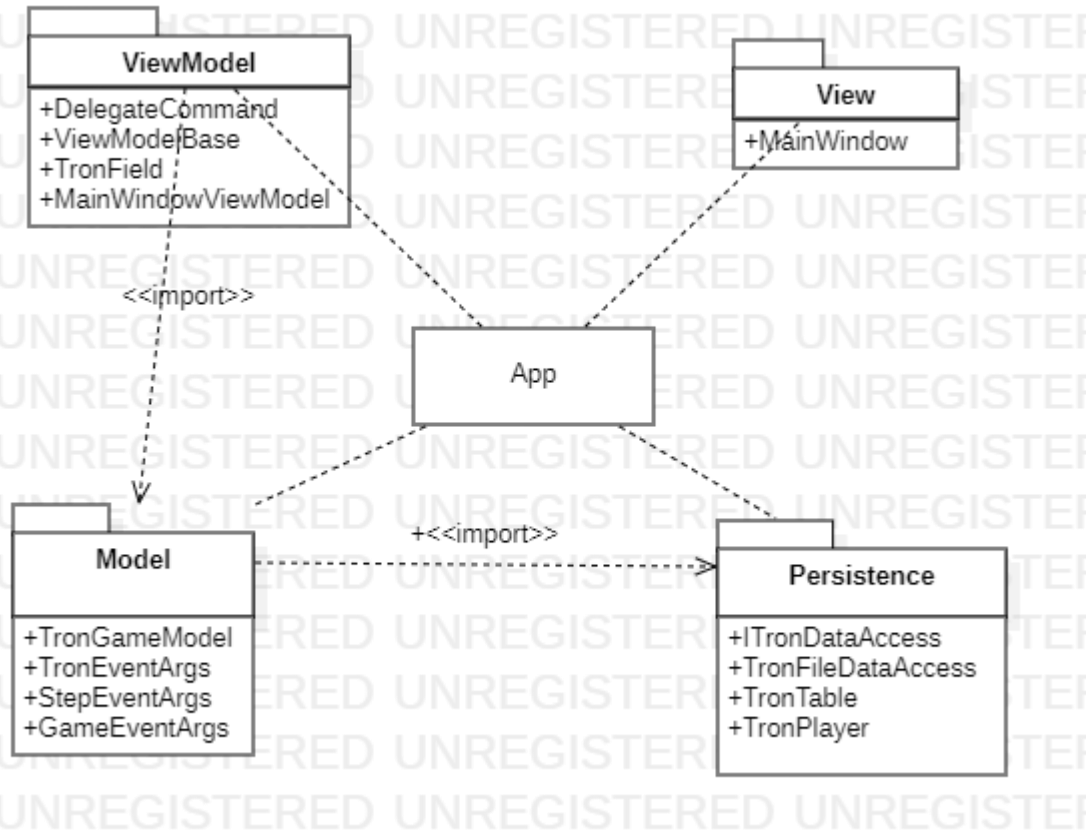
- A játék elején 3 pálya méret közül választhatunk: (12x12),(24x24),(36x36). A játékot a ’start’ gomb segítségével tudjunk elindítani, de további lehetőségünk van még a játék egy régebbi mentésének a betöltésére vagy az alkalmazás bezárására.
- Ha elindult a játék, a játéktér egy UniformGrid jelenik meg az ablak közepén.
- A játékosoknak lehetőségük van jobbra és balra fordulni az (’A’,’D’) és a (’⇦’,’⇨’) gombok segítségével és minden ’timer_Tick’-re lépnek a játékosok a megadott irányba és átszíneződnek a pálya négyzetei a megfelelő színre.
- A felhasználóknak lehetőségük van a játék megállítására a ’Space’ gomb segítségével, amellyel az időzítők, így a játék haladása is megáll.
- Amikor valamelyik játékos nyer, automatikusan megjelenik egy dialógusablak ami leírja hogy melyik játékos nyert és egy ‘OK’ gomb, aminek a megnyomásával az alkalmazás visszalép a fő menübe.

Felhasználói eset diagramm:



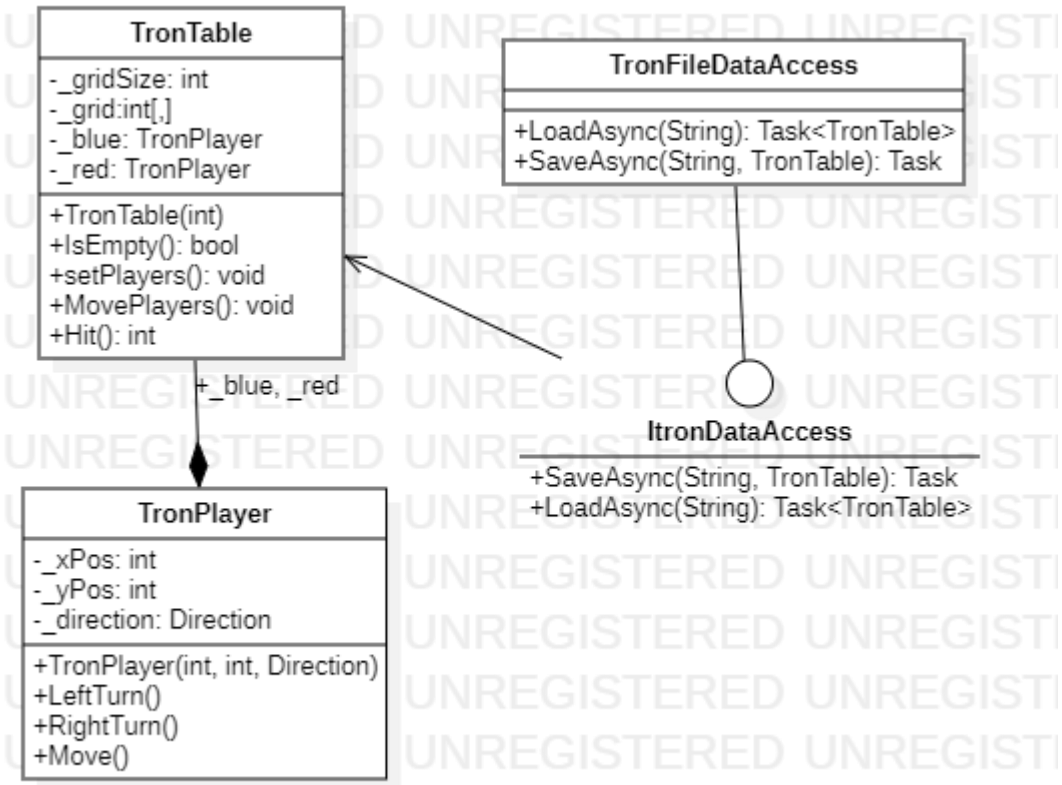
Tervezés:

A programot MVVM architektúrában valósítjuk meg. A megjelenítés a View, a megjelnités logikája a ViewModel, a modell a Model, míg a perzisztencia a Persistence névtérben helyezkedik el.



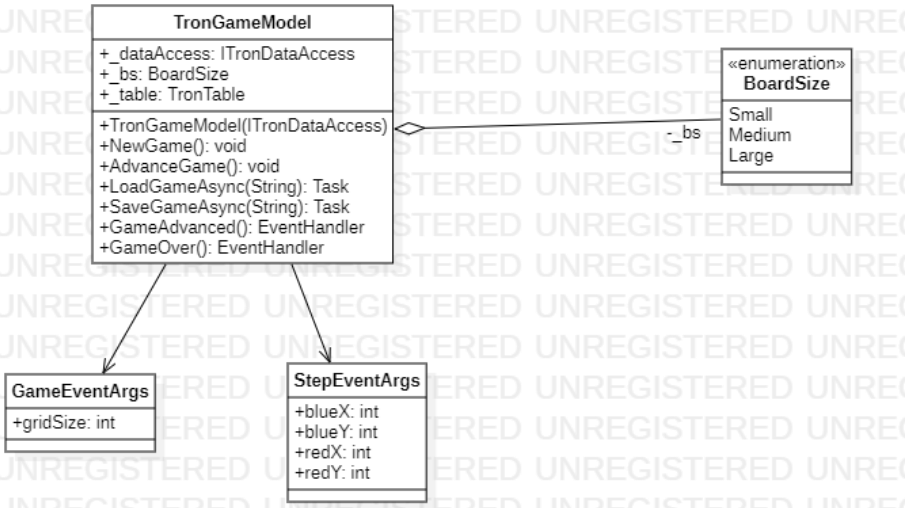
Perzisztencia:

- Feladata a játékkal kapcsolatos információk elmentése, tárolása és betöltése
- A játék elmentését a SaveAsync() míg a betöltését a LoadAsync() függvény valósítja meg aszinkron módon.
- A mentési fájl első sorában sorra: pályaméret, aktuális szint, kék játékos x pozíciója, kék játékos y pozíciója, kék játékos iránya, piros játékos x pozíciója, piros játékos y pozíciója, piros játékos iránya
- A fájl többi része a táblának az elmentett állását tartalmazza, a játékosok eddigi pozícióival.
  (0-Üres, 1-Kék játékos, 2-Piros játékos)
- Tartalmaz egy Player osztályt is amely tartalmazza, a játékosok aktuális x és y pozícióját illetve irányát. Itt valósul meg a játékosok, irányváltoztatása (RightTurn(), LeftTurn()) és léptetése (Move()) az irány alapján.



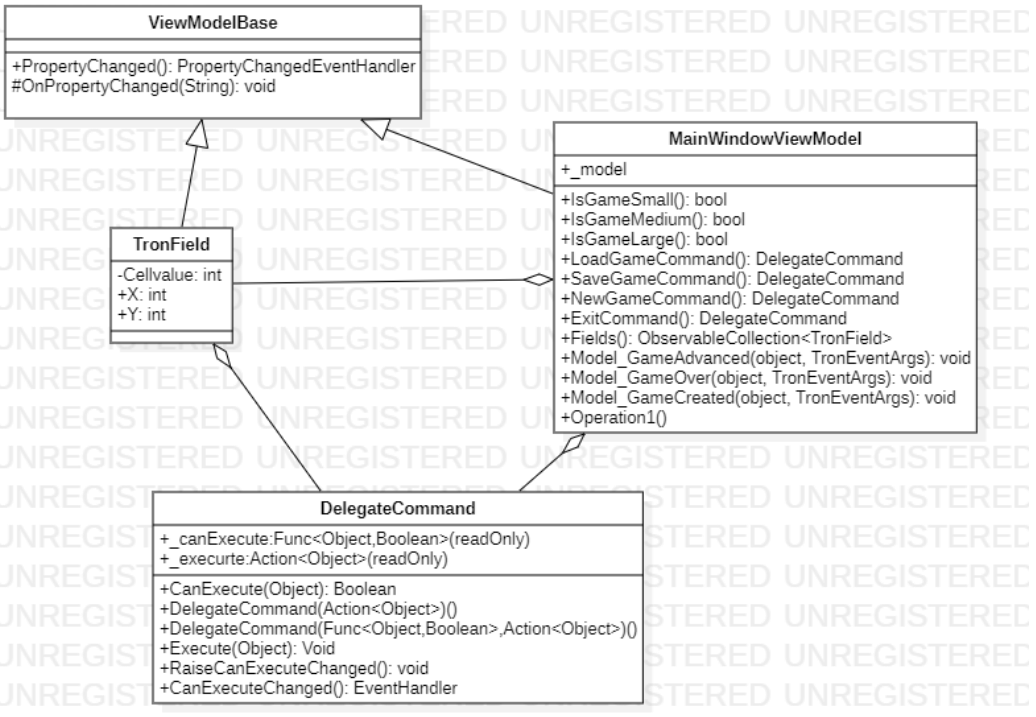
Modell:

- A modell lényegi részét a GameModel osztály valósítja meg, amely lehetőséget ad lekérdezni, hogy volt-e ütközés, lépni (AdvanceGame()), illetve a játék végének jelzésére (GameOver()).



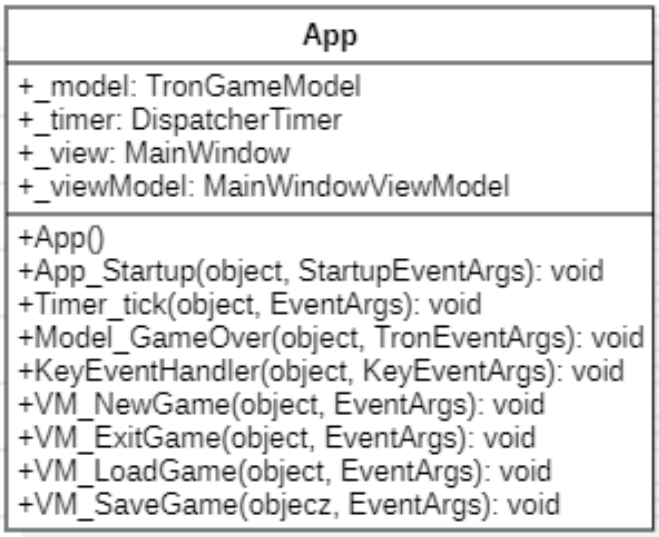
NézetModell:

- A nézetet a Main_Window osztály biztosítja
- A játékmező egy ItemsControl ami dinamikus épít fel egy UniformGridet, ami négyszögekből áll, amelyek színét és pozicióját adatkötéssel határozunk meg
- A fájl betöltését és mentését dialógusablakok segítségével végezzük.



App:

- Feladata az egyes rétegek példányosítása és a modell eseményeinek kezelése, játék szabályozása.
- A játék léptetését egy időzitő(_timer) végzi, amely kezdetben 800 ms-enként, lép és csökenti az intevallumát 4ms-al.



Tesztelés:

A modell funkcionalitása egységtesztek segítségével lett ellenőrizve a GameModelTest osztályban.
Tesztesetek:

- Player_TurnLeft();
- Player_TurnRight(): Azt ellenőrzi, hogy a Player osztály Left és RightTurn metódusai megfelelően állítják-e be a játékos irányát.
- Player_Move(): Azt ellenőrzi, hogy a Player osztály Move() metódusa megfelelően növeli vagy csökkenti a játékos x vagy y pozícióját a megadott irány alapján.
- GameModel_CorrectGridSize():Azt ellenőrzi, hogy a modell konstruktorában megfelelően állítódik-e be a pálya mérete.
- GameModel_BluePlayerWins(),
- GameModel_RedPlayerWins(): Azt ellenőrzi, hogy ha bármely játékos ütközik valamivel, akkor le áll-e a játék.
- FileDataAccess_SaveandLoad(): A program Save és Load metódusát teszteli