

## Beadandó feladat dokumentáció

**Készítette :** Kovács Máté

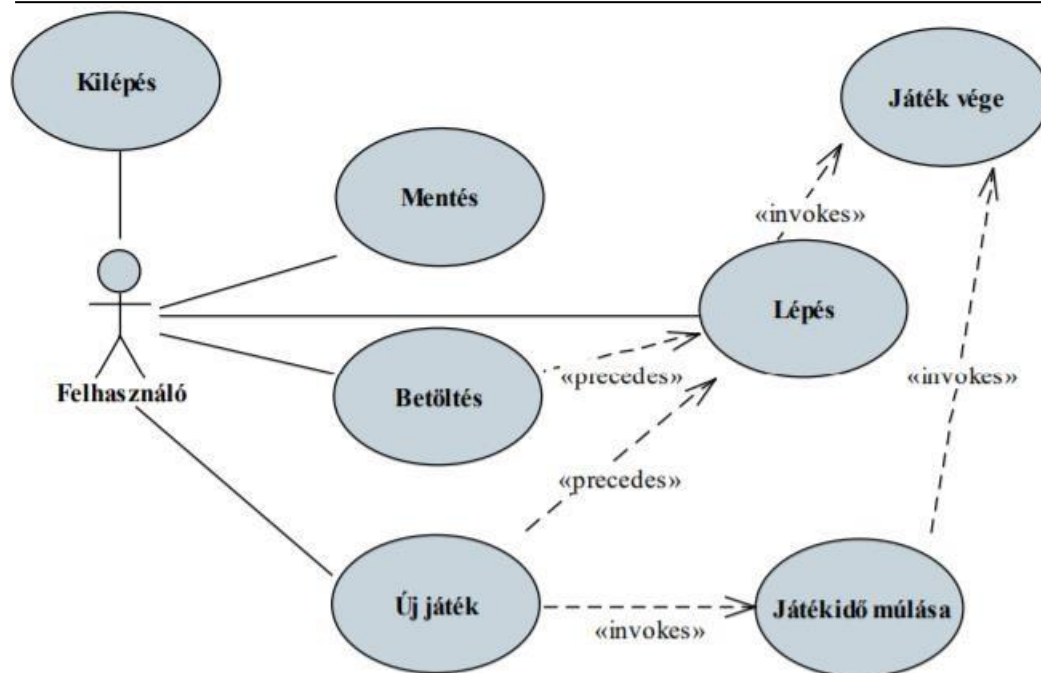
### Feladat:

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.

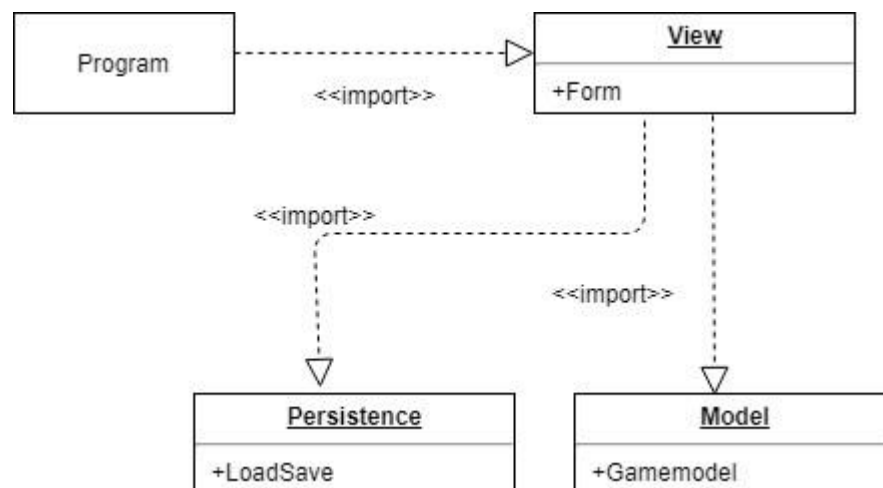
A program biztosítson lehetőséget új játék kezdésére a pályaméret megadásával ( $12 \times 12$ ,  $24 \times 24$ ,  $36 \times 36$ ), valamint játék szüneteltetésére (ekkor nem telik az idő, és nem mozognak a motorok). Ismerje fel, ha vége a játéknak, és jelenítse meg, melyik játékos győzött. Ezen felül szüneteltetés alatt legyen lehetőség a játék elmentésére, valamint betöltésére.

### Elemzés:

- A játékot három mérettel játszhatjuk: kicsi ( $12 \times 12$ ), közepes ( $24 \times 24$ ), nagy ( $36 \times 36$ ). A program indításkor kis méretet állít be, és automatikusan új játékot indít.
- A feladatot egyablakos asztali alkalmazásként Windows Forms grafikus felülettel valósítjuk meg.
- Az ablakban elhelyezünk egy méretválasztó menüt, egy Start , Mentés, Betöltés, és egy Pause gombot, azonkívül egy kijelzőt ami az időt mutatja.
- A játéktáblát egy  $n \times n$  nyomógombokból álló rács reprezentálja. A pályán két "fénymotor" közlekedik, maguk után falat húzva. Ha a motorok nekiütköznek, akár a saját faluknak is ,akkor a játékot a másik játékos nyeri
- A játék automatikusan feldob egy dialógusablakot, amikor vége a játéknak (Valamelyik fénymotor fallal ütközik).
- A felhasználói esetek a következő ábrán láthatóak.



## Tervezés:



### • Programszerkezet:

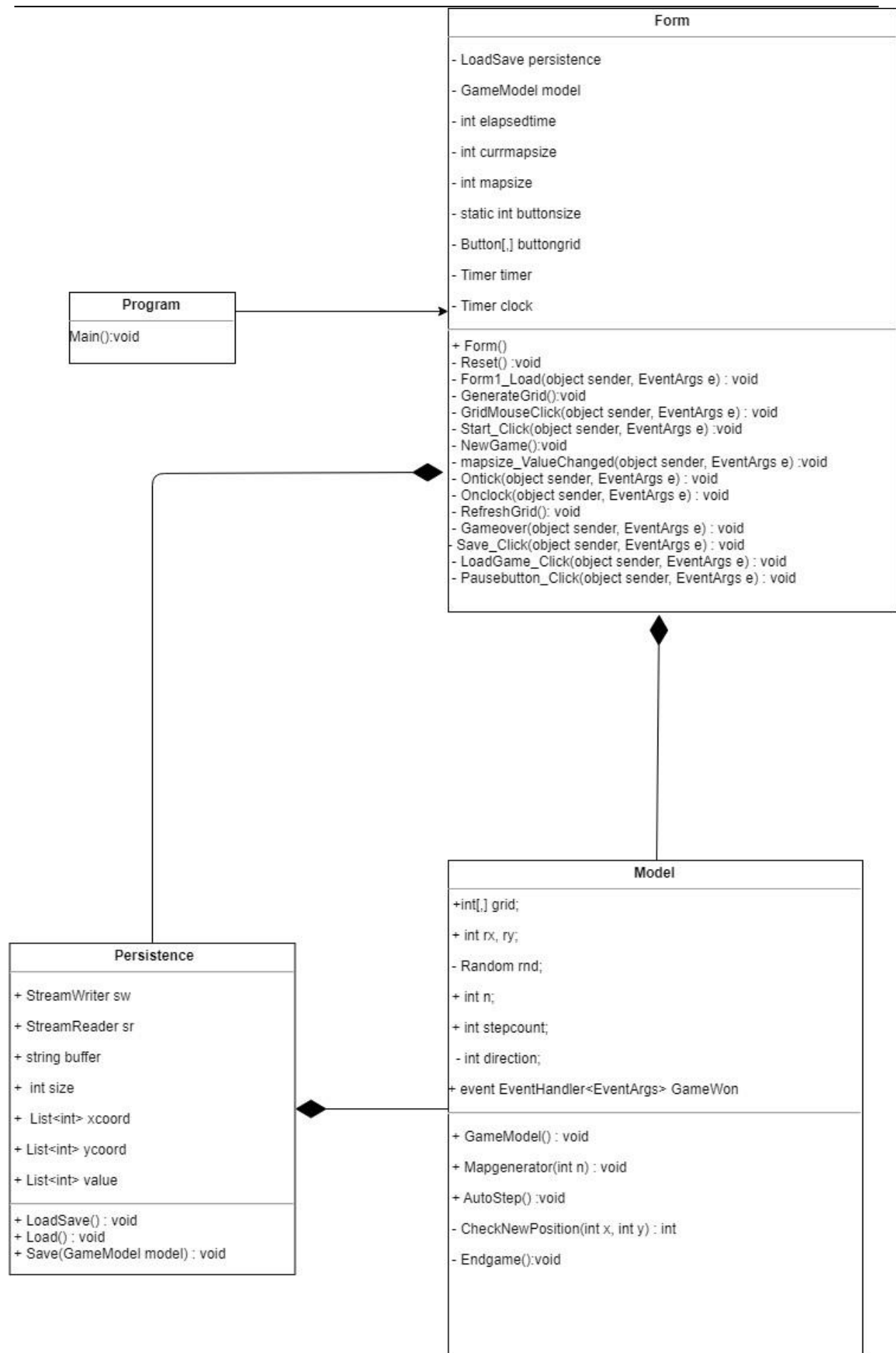
- A programot háromrétegű architektúrában valósítjuk meg. A megjelenítés a View, a modell a Model, míg a perzisztencia a Persistence névtérben helyezkedik..
- **Perzisztencia:**
- Az adatkezelés feladata a táblával kapcsolatos információk tárolása, valamint a betöltés/mentés biztosítása. A Load metódus a betöltést, míg a Save a mentést végzi. A mentés-betöltés folyamata egy txt file-ba történik, az első sor a pálya méretét, míg az alatta levő sorok a pályán elhelyezkedő objektumokat (falak, fénymotorok) és azok koordinátáit tárolják.

- **Modell:**

- A modell egy osztályból, a Gamemodel osztályból áll. Ez tartalmazza a pályát, ami egy 2dimenziós integer matrixot, a játékosok jelenlegi pozícióját, az irányt amibe tartanak.
- A Mapgenerator() hozza létre a pályát, előre megadott méret alapján, berakja a játékosokat a kezdő pozícióba.
- Az step() végzi játékosok mozgatását. Ezenkívül, itt történik a játék végének ellenőrzése is, azaz az EndGame() fv. Meghívása.
- Az EndGame() ellenőrzi le, hogy a játék véget ért-e, azaz bármelyik játékos nekiment bármelyik falnak, legyen az fénymotor által húzott, vagy a pálya széle.

- **Nézet:**

- A nézet egy osztályból, a Form osztályból áll. Ez tartalmaz egy model és egy persistence objektumot, az eltelt időt, a jelenlegi pálya méretet, egy aktuális pályaméretet, egy gombmátrixot, és két időzítőt, egyet ami a lépésekért felel, és egyet ami az eltelt időt számolja.
- A Reset() függvény meghívódik minden játék elején. Ez felel azért, hogy a gombmátrix, model, időzítők, stb. Mind törlődjének, és ne vigyenek át értéket az új játékba.
- A GenerateGrid() osztály hozza létre a gombmátrixot.
  - A NewGame osztály meghívja a Reset() és GenerateGrid() fveket, beállítja a pályaméretet az aktuális pályaméretre, meghívja a model Mapgenerator() fvt, létrehozza az időzítőket, majd elindítja azokat.



**Tesztesetek:**

Generator(): Pályagenerálás tesztelése

Lepes():Játékos lépésének tesztelése

**További fejlesztésre lehetőség:**

- A játék nagyobb pályaméreteken lassú, másszóval nem optimalizált erre a felhasználásra(ez részben a winforms hibája, részben saját)
- Grafikus fejlesztés, képek behelyezése, stb.