

„Eseményvezérelt Alkalmazások”

3.Beadandó dokumentáció

4.feladat

Készítette: Székely Regő

Neptun-azonosító: H5LJFS

E-mail: h5ljfs@inf.elte.hu

2023.február 5.

Feladat

4. Labirintus

Készítsünk programot, amellyel a következő játékot játszhatjuk.

Adott egy $n \times n$ elemből álló játékpálya, amely labirintusként épül fel, azaz fal, illetve padló mezők találhatók benne, illetve egy kijárat a jobb felső sarokban. A játékos célja, hogy a bal alsó sarokból indulva minél előbb kijusson a labirintusból.

A labirintusban nincs világítás, csak egy fáklyát visz a játékos, amely a 2 szomszédos mezőt világítja meg (azaz egy 5×5 -ös négyzetet), de a falakon nem tud átvilágítani.

A játékos figurája kezdetben a bal alsó sarokban helyezkedik el, és vízszintesen, illetve függőlegesen mozoghat (egyesével) a pályán.

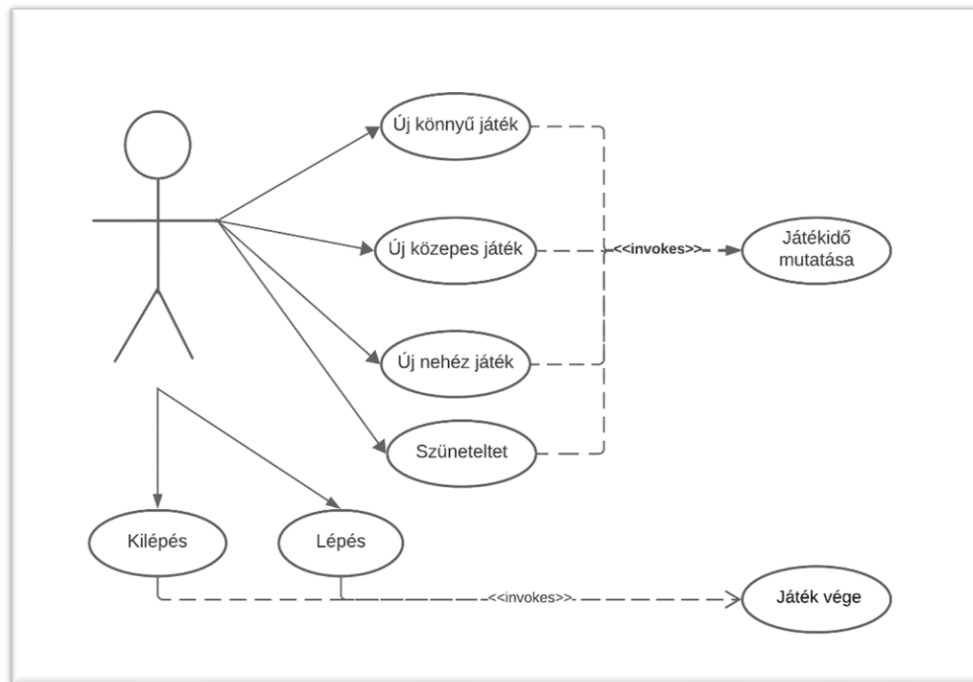
A pályák méretét, illetve felépítését (falak, padlók) tároljuk fájlban. A program legalább 3 különböző méretű pályát tartalmazzon.

A program biztosítson lehetőséget új játék kezdésére a pálya kiválasztásával, valamint játék szüneteltetésére (ekkor nem telik az idő, és nem léphet a játékos), továbbá ismerje fel, ha vége a játéknak. A program játék közben folyamatosan jelezze ki a játékidőt.

Elemzés:

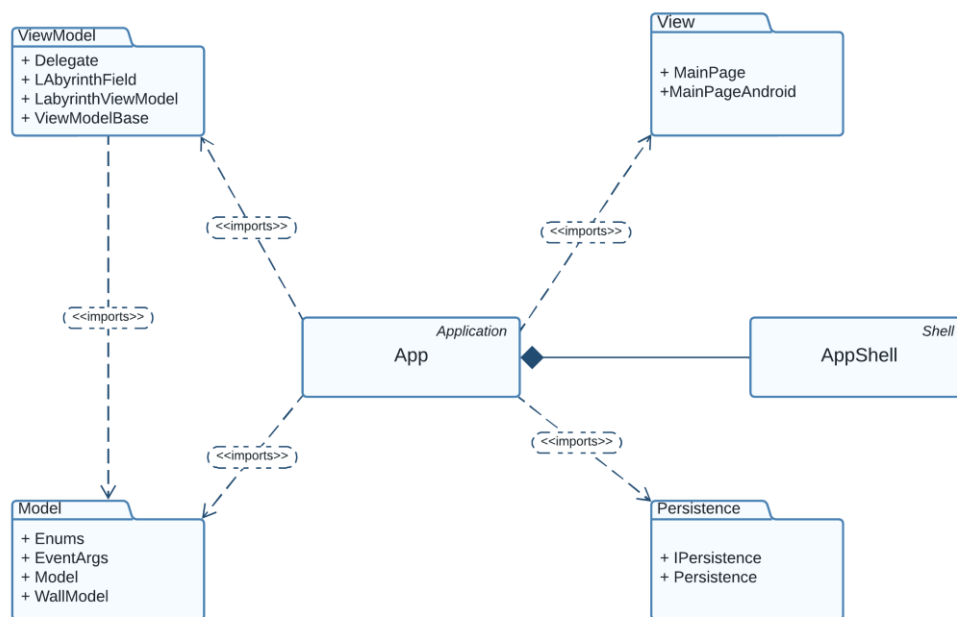
- A játékot három kiválasztott pályamérettel használhatjuk: 10×10 , 15×15 , 20×20 . Ezeket az „Easy”, „Medium” és „Hard” jelzőkkel választhatjuk ki. Érdemes megjegyezni azt, hogy ebben az esetben minél nagyobb a pálya annál nehezebb a játék hiszen a több fallal rendelkező pályák bonyolultabb utakat eredményeznek. A program indításkor New Game gombbal indítható a játék.
- A feladatot .NET MAUI alkalmazásként, elsődlegesen Windows és Android platformon valósítjuk meg.
- Az ablak aljában elhelyezem státuszsort, ami tartalmazza a szüneteltetés (Pause) gombot és az eltel idő jelzésére szolgáló számlálót. A felső menüsávban található a lenyíló menüpont, ahol a következők találhatók: új játék (New Easy/Medium=Hard game) és kilépés (Exit Game).
- A játéktábla a kiválasztott méret alapján generálódik le.
- A játékos mozgása a „Húzás” gesztusokkal történik (Fel-Bal-Le-Jobb).
- A játék automatikusan feldob egy dialógusablakot, amikor vége a játéknak rajta megfelelő üzenettel.

- Az ábrán láthatóak a felhasználói esetek:

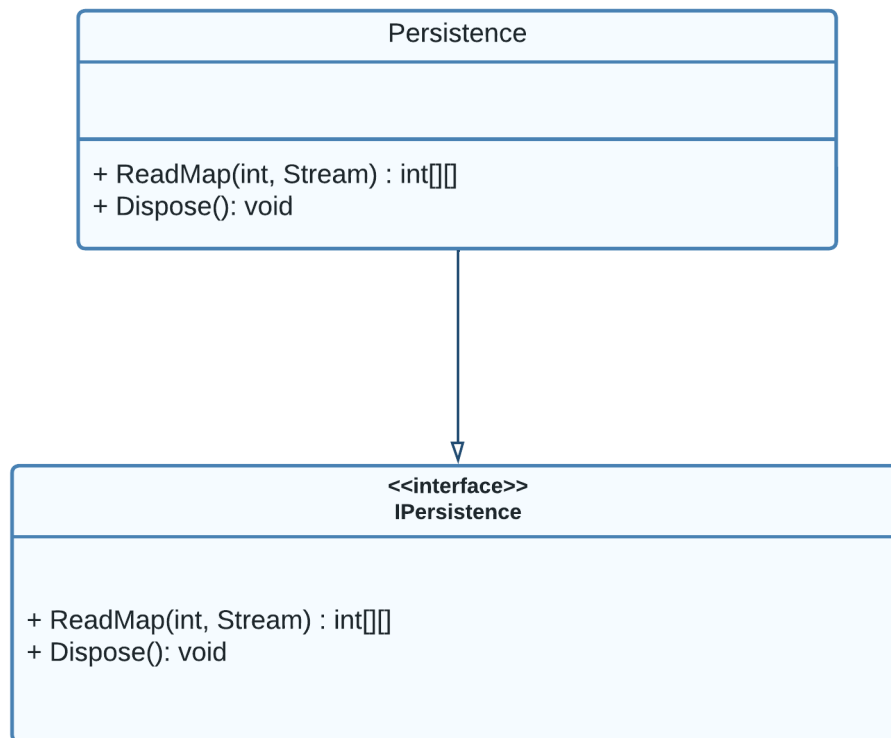


Tervezés:

- *Programszerkezet:*
 - A szoftvert két projektből építjük fel: a modellt és a perzisztenciát tartalmazó osztálykönyvtárból (.NET Standard Class Library), valamint a .NET MAUI többplatformos projektből, amelyet Windows és Android operációs rendszerre is le tudunk fordítani.

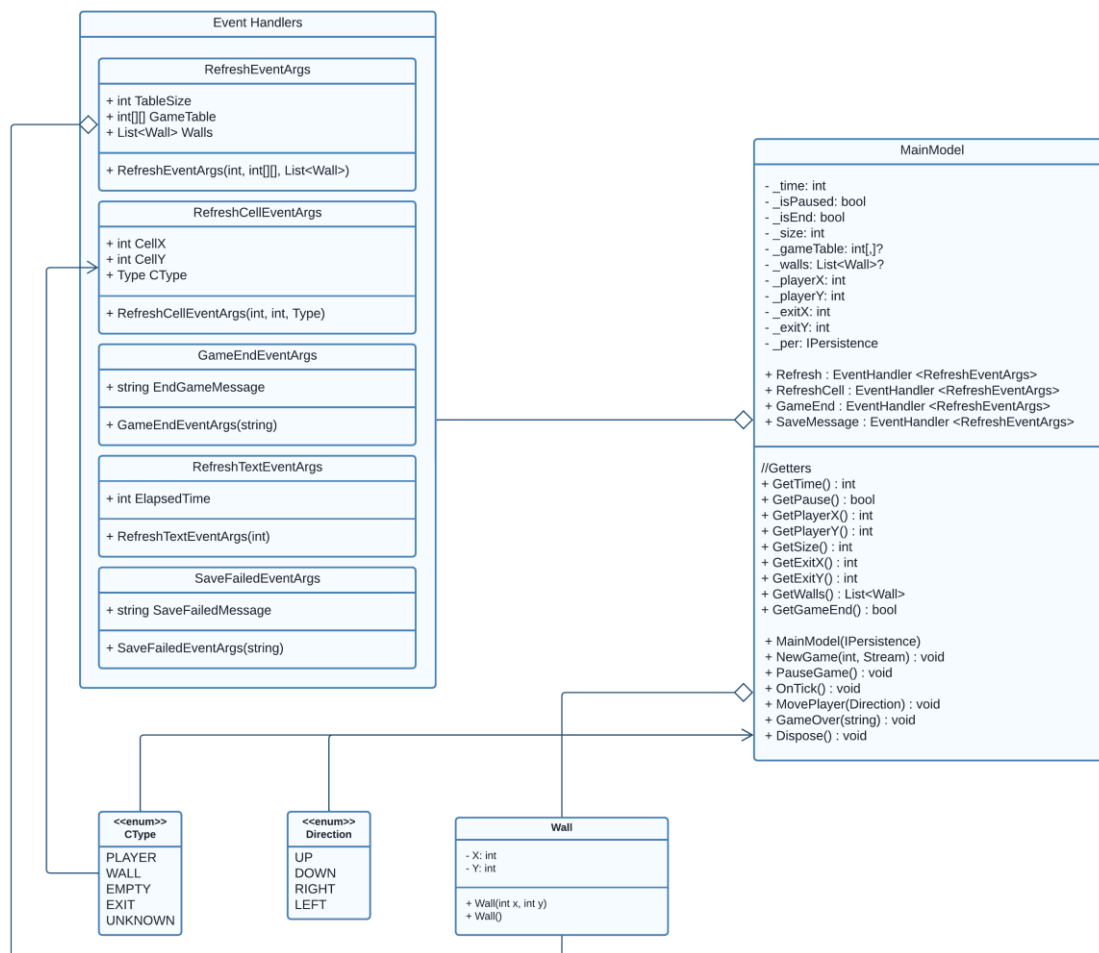


- *Perzisztencia:*
 - A betöltés feladatát látja el. Illetve saját hibát is dob, ha sikertelen a betöltés.
 - A program az adatokat egy szövegfájlban tárolja, amit a megadott mentési útvonalon bármikor el lehet menteni. Hiba lép fel, ha nem létezik a megadott útvonal.

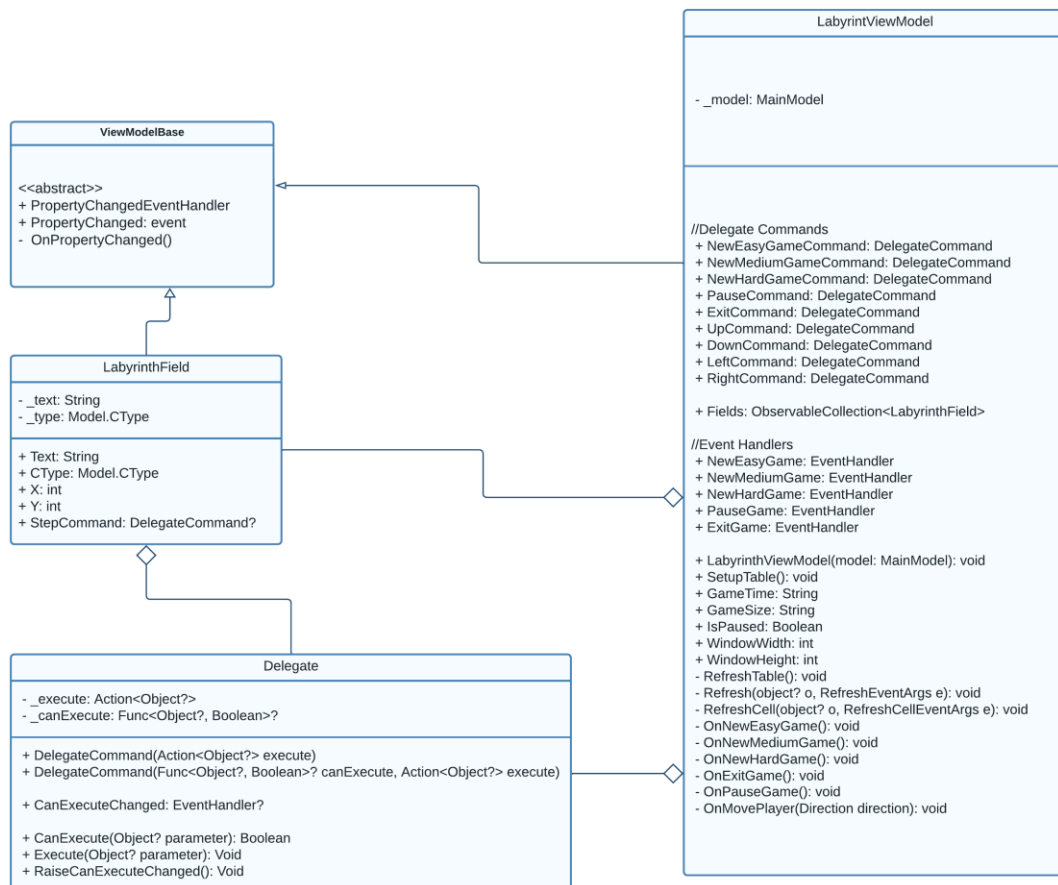


• *Modell:*

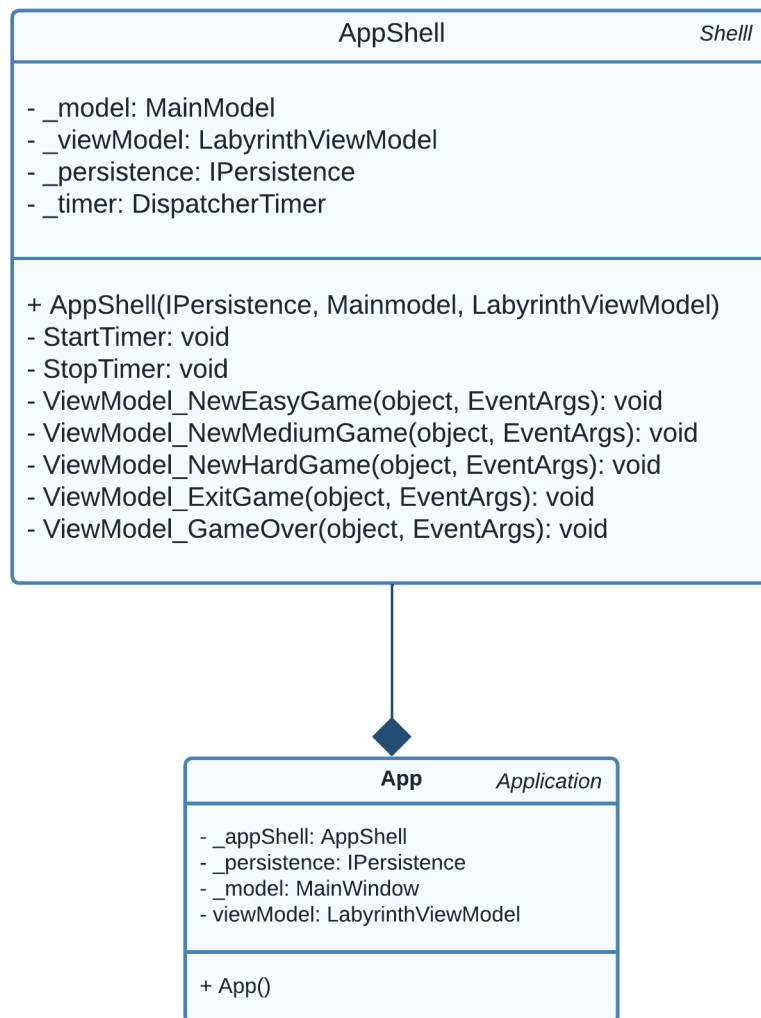
- A modell lényegi részét a MainModel osztály valósítja meg, amely szabályozza a tábla tevékenységeit, valamint a játék paramétereit úgy, mint az idő(_time), játékos pozíciója (_playerX és _playerY) stb. A típus lehetőséget ad új játék kezdésére (NewGame), valamint lépésre (MovePlayer). Új játéknál a kiválasztott méret alapján legenerálódnak a falak, a játékos és a kijárat. Az idő haladását időbeli lépések végzése (OnTick) teszi lehetővé.
- A játékalapot változásáról az OnTick és a MovePlayer esemény gondoskodik, hiszen a kettő egymástól függetlenül működik. A játék végét a GameOver esemény kezeli.
- A játéktér előre el van tárolva.
- A játék időbeli kezelését egy időzítő végzi (_timer), amelyet mindig aktiválunk játék során, illetve inaktiváljuk, amennyiben a játék megállításra kerül.



- **Nézet:**
 - A nézetet navigációs lapok segítségével építjük fel. A MainPage osztály tartalmazza a játéktáblát Windows rendszerhez, amelyet egy Grid segítségével valósítunk meg, amelyben Button elemeket helyezünk el.
 - A nézetet navigációs lapok segítségével építjük fel. A MainPageAndroid osztály tartalmazza a játéktáblát Android rendszerhez, amelyet egy Grid segítségével valósítunk meg, amelyben Button elemeket helyezünk el.
- **NézetModell:**
 - A nézetmodell megvalósításához felhasználunk egy általános utasítás (DelegateCommand), valamint egy ős változásjelző (ViewModelBase) osztályt.
 - A nézetmodell feladatait a LabyrinthViewModel osztály látja el, amely parancsokat biztosít az új játék kezdéséhez, játék betöltéséhez, valamint a kilépéshez. A parancsokhoz eseményeket kötünk, amelyek a parancs lefutását jelzik a vezérlőnek. A nézetmodell tárolja a modell egy hivatkozását (_model), de csupán információkat kér le tőle, illetve a játéknehézséget szabályozza. Direkt nem avatkozik a játék futtatásába.



- **Vezérlés:**
 - Az App osztály feladata az alkalmazás vezérlése, a rétegek példányosítása és az események feldolgozása.
 - A CreateWindow metódus felüldefiniálásával kezeljük az alkalmazás életciklusát a megfelelő eseményekre történő feliratkozással. Így az alkalmazás felfüggesztéskor (Stopped) elmentjük az aktuális játékállást (SuspendedGame), míg folytatáskor vagy újraindításkor (Activated) pedig folytatjuk, amennyiben történt mentés.
 - Az alkalmazás lapjait egy AppShell keretben helyezzük el. Ez az osztály felelős a lapok közötti navigációk megvalósításáért.



Tesztelési terv:

- A modell működését egységtesztek segítségével lett vizsgálva (GameTest osztály).
- Megvalósított tesztesetek:
 - NewGameTest1: Új játék létrehozásának elemzése és annak változói ellenőrzése 10 x 10-es tábla esetén.
 - NewGameTest2: Új játék létrehozásának elemzése és annak változói ellenőrzése 15 x 15-es tábla esetén.
 - NewGameTest3: Új játék létrehozásának elemzése és annak változói ellenőrzése 20 x 20-es tábla esetén.
 - PlayerMovementTest: Játékos mozgásának ellenőrzése minden irányba, illetve, hogy nem mozdul pályán kívülre.
 - GameEndTest: Játék végállapotának vizsgálata indítás előtt.
 - TimeTest: Állapotvizsgálatok, hogy az időzítő tényleges rendben működik.
 - EndGameTest: A játék végállapotának vizsgálata
 - HitWallTest: A játékos pozíciójának vizsgálata falnak ütközés esetén.