

3.Beadandó feladat dokumentáció

Készítette:
Nagy Rebeka
NZONGK

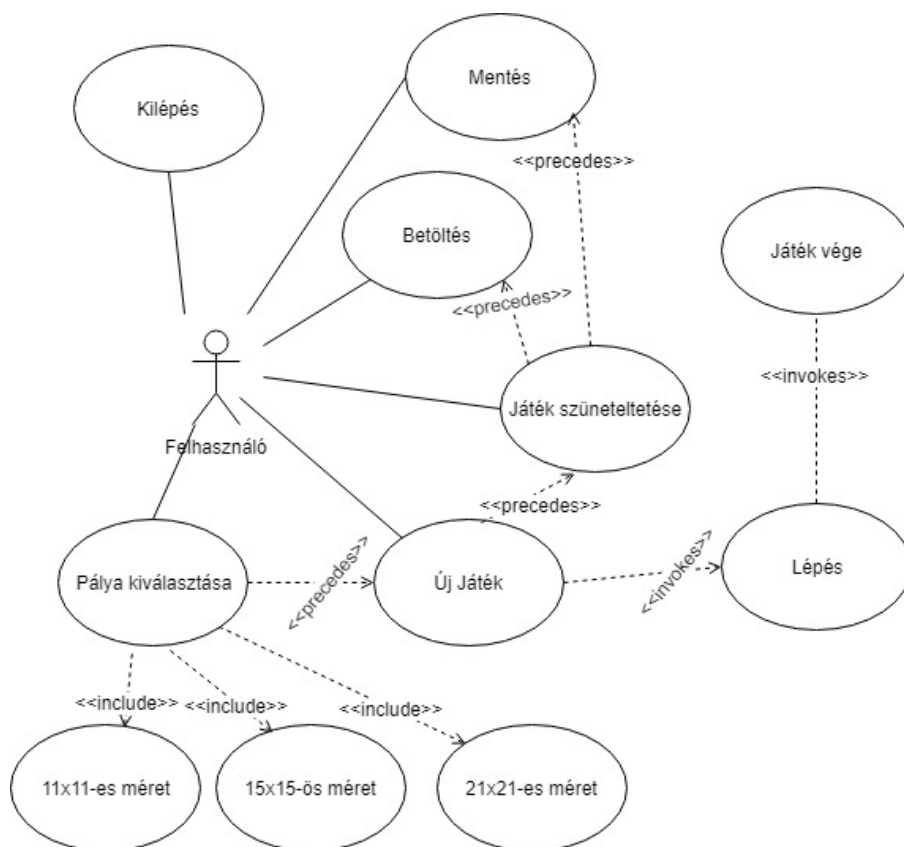
Feladat:

Készítsünk programot, amellyel a következő játékot játszhatjuk. Adott egy $n \times n$ elemből álló játékpálya, ahol a játékos két üldöző elől próbál menekülni, illetve próbálja őket aknára csalni. Kezdetben a játékos játékpálya felső sorának közepén helyezkedik el, a két üldöző pedig az alsó két sarokban. Az ellenfelek adott időközönként lépnek egy mezőt a játékos felé haladva úgy, hogy ha a függőleges távolság a nagyobb, akkor függőlegesen, ellenkező esetben vízszintesen mozognak a játékos felé. A pályán véletlenszerű pozíciókban aknák is elhelyezkednek, amelyekbe az ellenfelek könnyen beleléphetnek, ekkor eltűnnek (az akna megmarad). A játékos vízszintesen, illetve függőlegesen mozoghat (egyesével) a pályán, és célja, hogy az ellenfeleket aknára csalja, miközben ő nem lép aknára. Ha sikerül minden üldözőt aknára csálnia, akkor győzött, ha valamely ellenfél elkapja (egy pozíciót foglal el vele), vagy aknára lép, akkor veszített. A program biztosítson lehetőséget új játék kezdésére a pályaméret megadásával (11×11 , 15×15 , 21×21), valamint játék szüneteltetésére (ekkor nem telik az idő, és nem léphet senki). Ismerje fel, ha vége a játéknak, és jelenítse meg, hogy győzött, vagy veszített-e a játékos. Ezen felül szüneteltetés alatt legyen lehetőség a játék elmentésére, valamint betöltésére. A program játék közben folyamatosan jelezze ki a játékidőt.

Elemzés:

- A játékot három különböző méretű játéktáblával játszhatjuk (11×11 , 15×15 , 21×21). A játék addig nem indul el, amíg a játékos nem választja ki a kívánt méretet.
- A feladatot egyablakos asztali alkalmazásként Windows Presentation Foundation grafikus felülettel valósítjuk meg.
- Az ablakban elhelyezünk egy menüt a következő menüpontokkal: New Game(11×11 , 15×15 , 21×21), Pause, Load, Save. Az ablak alján megjelenítünk egy státuszsort, amely a lépések számát, illetve a hátralévő időt jelzi.
- A játéktáblát a táblamérettől függő számosságú panelek reprezentálják. A játékost egy zöld négyzet jelöli, az ellenfeleket piros, az aknákat sárga. A megfelelő billentyűzet lenyomása következtében a játékos annak megfelelő irányba mozdul el. A játéktábláról nem lehet „lelépni”.
- A játék automatikusan feldob egy dialógusablakot, amikor vége a játéknak, melynek szövege annak függvényében változik, hogy nyertünk, vagy veszítettünk. Szintén dialógusablakkal végezzük el a mentést, illetve a betöltést, a fájlneveket a felhasználó adja meg.

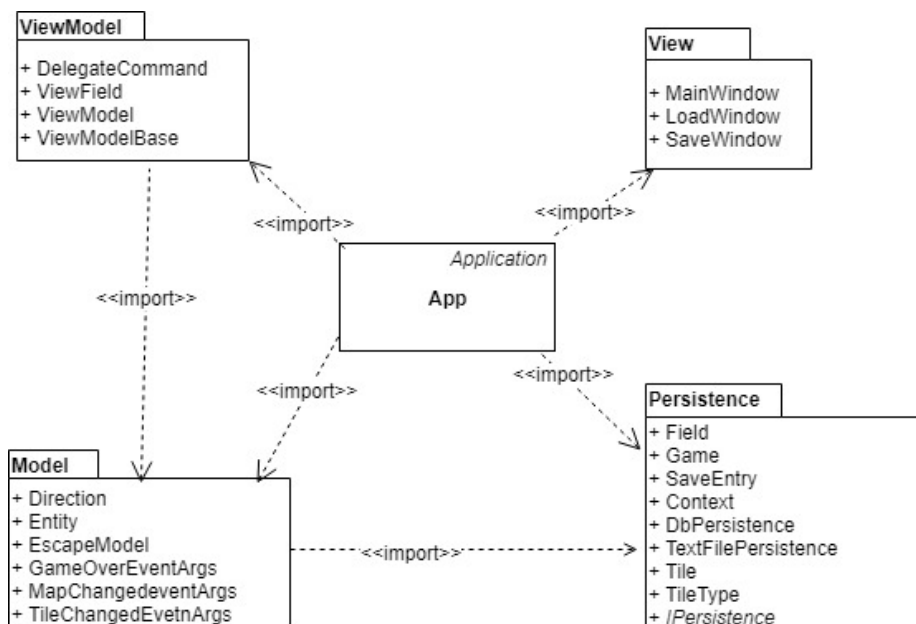
Felhasználói esetek:



Tervezés:

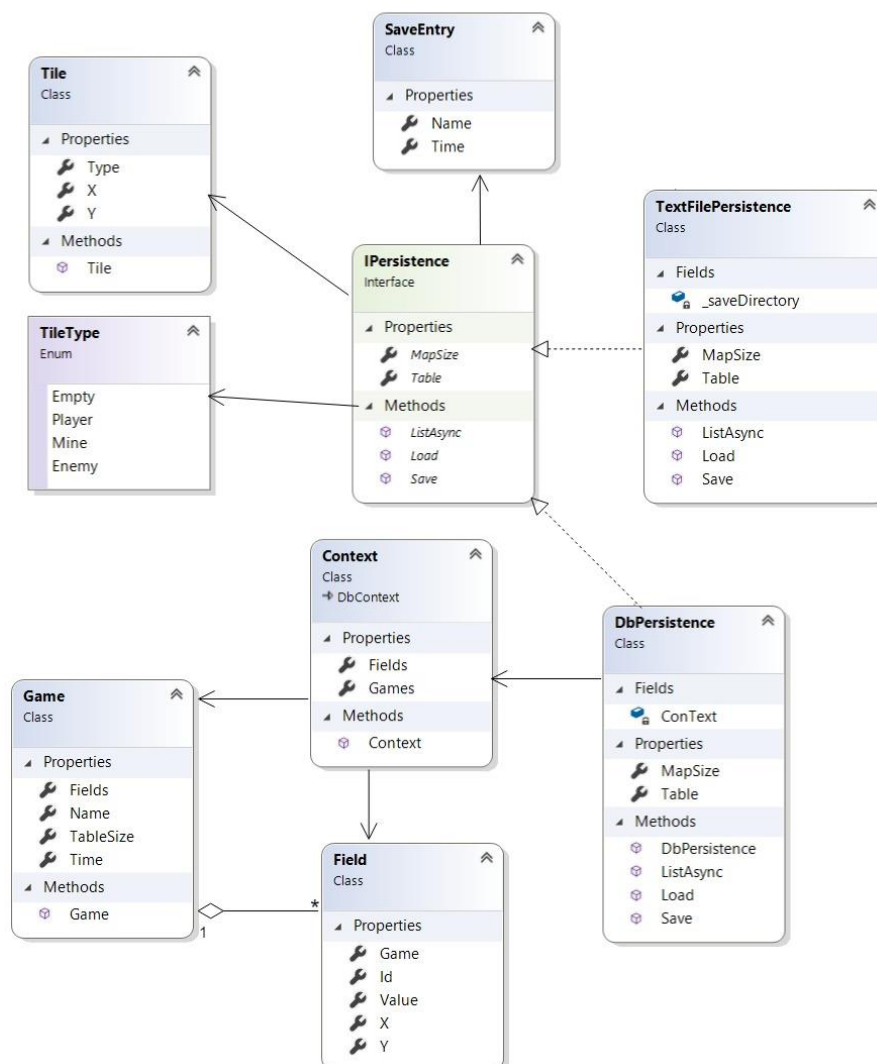
- **Programszerkezet:**

- A programot MVVM architektúrában valósítjuk meg, ennek megfelelően View, Model, ViewModel és Persistence névtereket valósítunk meg az alkalmazáson belül. A program környezetét az alkalmazás osztály (App) végzi, amely példányosítja a modellt, a nézetmodellt és a nézetet, biztosítja a kommunikációt, valamint felügyeli az adatkezelést.



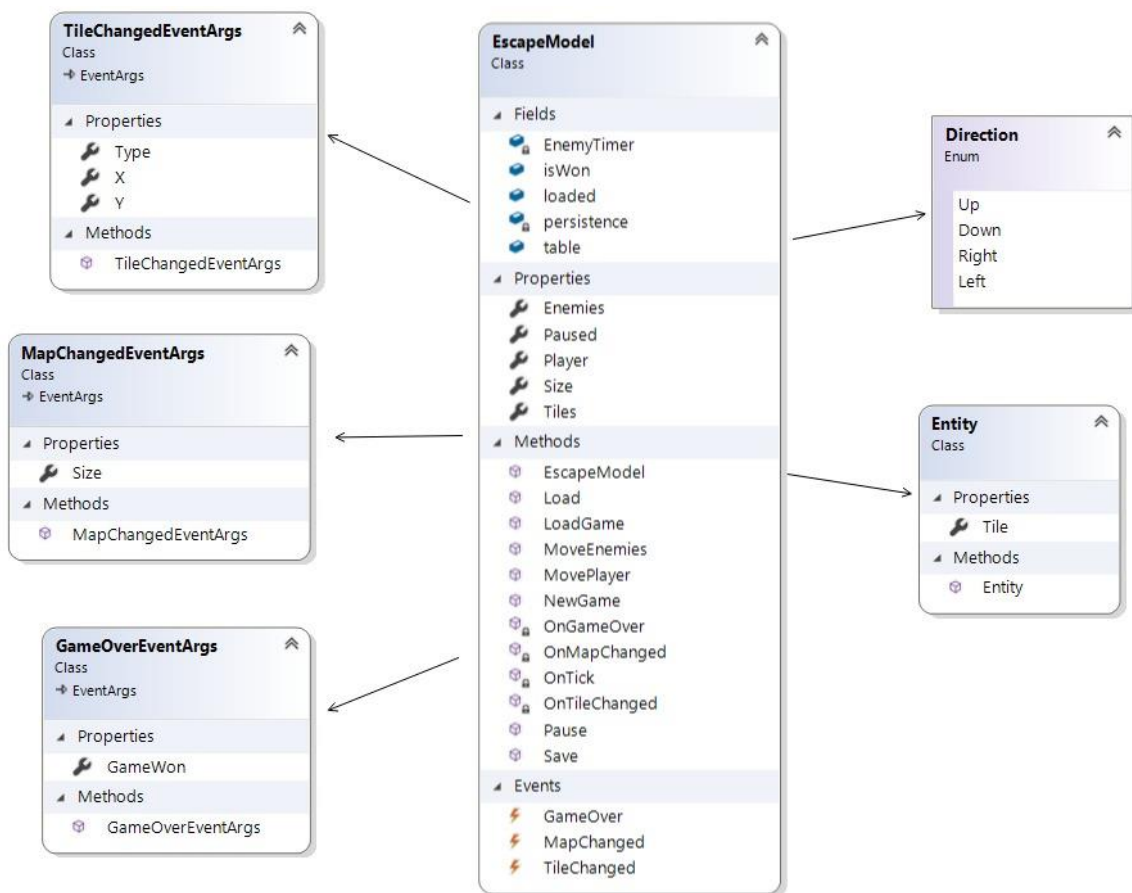
- **Perzisztencia:**

- Az adatkezelés feladata az Escape táblával kapcsolatos információk tárolása, valamint betöltés/mentés biztosítása.
- A Tile osztály tartalmazza a táblán megtalálható mezők tulajdonságait.
- A TileType egy enum, mely 4 féle lehet: Player, Enemy, Mine és Empty. Ennek segítségével tudja egy mező, hogy milyen objektum van „rajta”.
- Az IPersistence interfész adja meg a hosszú távú adattárolás lehetőségeit. Lehetőséget ad a betöltésre (Load) és mentésre (Save). A műveleteket hatékonysági okokból aszinkron módon valósítjuk meg.
- Az interfészt szöveges fájl alapú adatkezelésre a TextFilePersistence osztály valósítja meg.
- Az interfészt adatbázis alapú adatkezelésre a DbPersistence osztály valósítja meg. Az adatbáziskezelés Entity Framework használatával a Field és Game entitás típusokkal és Context adatbázis kontextussal történik.
- A program az adatokat szöveges fájlként tudja eltárolni, vagy adatbázisban. Ezeket az adatokat a programban bármikor be lehet tölteni, illetve ki lehet menteni az aktuális állást.



- **Modell:**

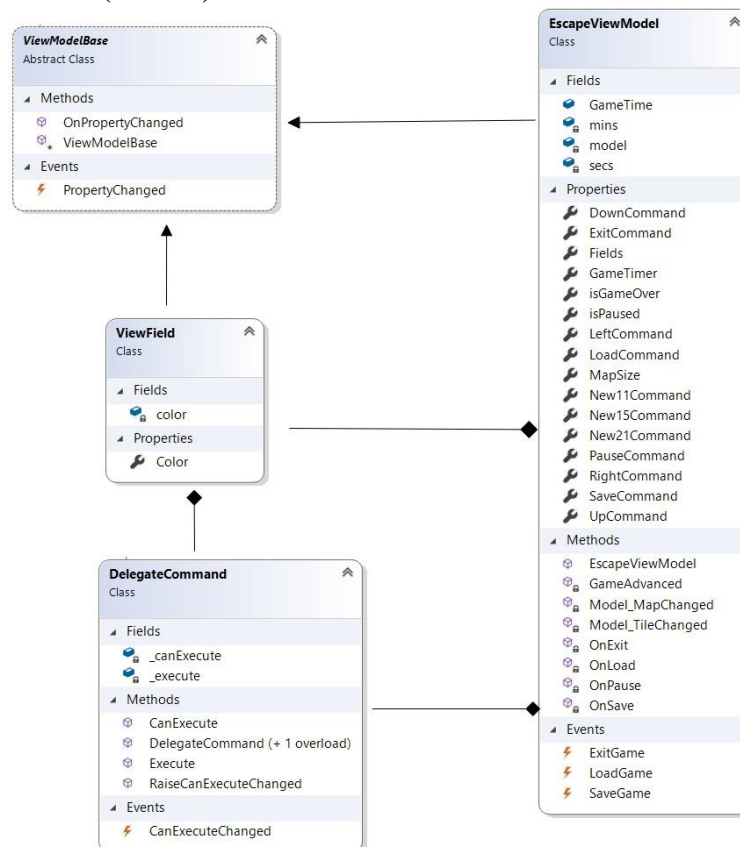
- A modell lényegi részét az EscapeModel osztály valósítja meg, amely szabályozza a tábla tevékenységeit, valamint a játék egyéb paramétereit, úgymint az ellenségek mozgását(MoveEnemies), a játékos léptetését(MovePlayer), az új játék kezdését(NewGame), betöltött játék folytatását(LoadGame), az ellenségek mozgásához egy timert(EnemyTimer).
- A játéktábla változásait két esemény vezérli, a méretének megváltozását a MapChanged, az adott mezők megváltozásait pedig a TileChanged események.
- A játék kimeneteleire szintén egy esemény van létrehozva, aminek a neve GameOver.
- A modell példányosításkor megkapja az adatkezelés felületét, amelynek segítségével lehetőséget ad betöltésre (Load), és mentésre(Save).



- **Nézetmodell:**

- A nézetmodell megvalósításához felhasználunk egy általános utasítás(DelegateCommand), valamint egy ős változásjelző (ViewModelBase) osztályt.
- A nézetmodell feladatait a ViewModel látja el, amely parancsokat biztosít az új játék kezdéséhez, játék betöltéséhez, mentéséhez, szüneteltetéséhez valamint kilépéshez. A parancsokhoz eseményeket kötünk, amelyek a parancs lefutását jelzik a vezérlónek. A nézetmodell tárolja a modell egy hivatkozását (model), de csupán információkat kér le tőle, illetve a játéktábla méretét szabályozza. Direkt nem avatkozik a játék futtatásába.

- A játéklemező számára egy külön mezőt biztosítunk (ViewField), amely eltárolja az adott mező színét. A mezőket egy felügyelt gyűjteménybe helyezzük a nézetmodellbe (Fields).

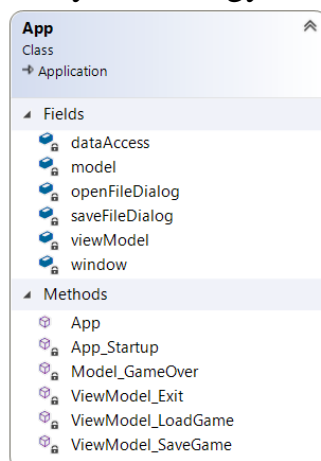


• Nézet:

- A nézet csak egy képernyőt tartalmaz, a MainWindow osztályt. Egy rácsban tárolja a játéklemezőt, a menüt és a státuszsort. Minden adatot adatkötéssel kapcsolunk a felülethez, továbbá azon keresztül szabályozzuk a mezők színét is.
- A fájlnev bekérését betöltéskor és mentéskor, valamint a figyelmeztető üzenetek megjelenését beépített dialógusablakok segítségével végezzük.

• Környezet:

- Az App osztály feladata az egyes rétegek példányosítása (App_Startup), összekötése, a nézetmodell, valamint a modell eseményeinek lekezelése, és ezáltal a játék, az adatkezelés, valamint a nézetek szabályozása.
- A játék előrehaladásának megjelenítéséhez tárol egy időzítőt is (GameTime), amelynek állítását is szabályozza az egyes funkciók hatására.



Tesztelés:

- A modell funkcionalitása egységtesztek segítségével eltt ellenőrizve az EscapeTest osztályban.
- Az alábbi tesztesetek kerültek megvalósításra:
 - EscapeNewGameTest: Új játék megfelelő generálása, a játékos első sor középső mezőre, az ellenségek pedig a tábla két alsó sarkáakerülése. A timer elindulása.
 - EscapeMovePlayerTest: A játékos mozgatása megfelelően működésének ellenőrzése. A tábla szélein túlmozdulás nem lehetséges, az adott leütött billentyűk hatására a megfelelő irányba elmozdulás.
 - EscapeMoveEnemiesTest: A timer hatására megfelelő irányba elmozdulás.
 - EscapeGameWonTest: Az ellenségek aknára vezetése után a játék valóban felismeri a játék végét, és a játékos nyereségét.
 - EscapeGameOverTest: A játékos aknára lépése után a játék valóban felismeri a játék végét, és a játékos vereségét.