

### 3. Beadandó feladat dokumentáció

Készítette: Pataki Mátyás Balázs

E-mail: [dc88on@inf.elte.hu](mailto:dc88on@inf.elte.hu)

Feladat:

#### 8. Aszteroidák

Készítsünk programot, amellyel az aszteroidák játékot játszhatjuk.

A feladatunk az, hogy egy űrhajó segítségével átnavigáljuk egy aszteroidamezőt.

Az űrhajóval a képernyő alsó sorában tudunk balra, illetve jobbra navigálni. A képernyő felső sorában meghatározott időközönként véletlenszerű pozícióban jelennek meg az aszteroidák, amelyek folyamatosan közelednek állandó sebességgel a képernyő alja felé. Az idő múlásával egyre több aszteroida jelenik meg egyszerre, így idővel elkerülhetetlenné válik az ütközés. A játék célja az, hogy az űrhajó minél tovább elkerülje az ütközést.

A program biztosítson lehetőséget új játék kezdésére, valamint játék szüneteltetésére (ekkor nem telik az idő, és nem mozog semmi a játékban).

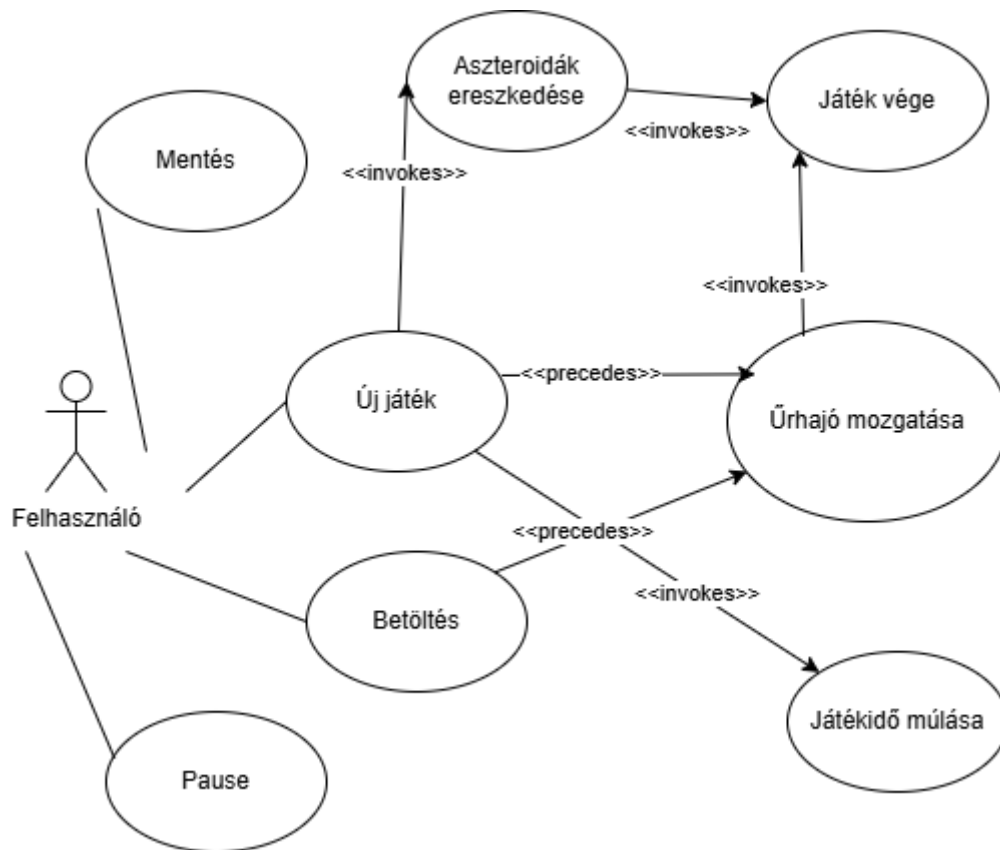
Ismerje fel, ha vége a játéknak, és jelenítse meg, mennyi volt a játékidő. Ezen felül szüneteltetés alatt legyen lehetőség a játék elmentésére, valamint betöltésére.

1.ábra : Felhasználói esetek diagramja

Elemzés:

- A feladatot egyablakos asztali és mobil alkalmazásként Avalonia grafikus felülettel valósítjuk meg.
- A játék alaphelyzetében az űrhajó a játéktér közepén áll és egy aszteroida sincsen. Az időmérő pedig még nincs elindítva.
- Az ablakban elhelyezünk egy menüt is, ahol a felhasználó új játékot tud indítani (New Game), le tudja menteni a játék állását (Save Game) és be tudja tölteni a régebben elmentett állásait (Load Game). Emellett megjelenítünk egy stopperórát is, ami mutatja a felhasználó számára, hogy mennyi ideje megy az adott játékmenet.
- A játék közben a bal és jobb nyíl lenyomásával tudja a játékos irányítani az űrhajót. A space billentyűvel tudja szüneteltetni a játékot (a menü csak szüneteltetett állapotban elérhető).

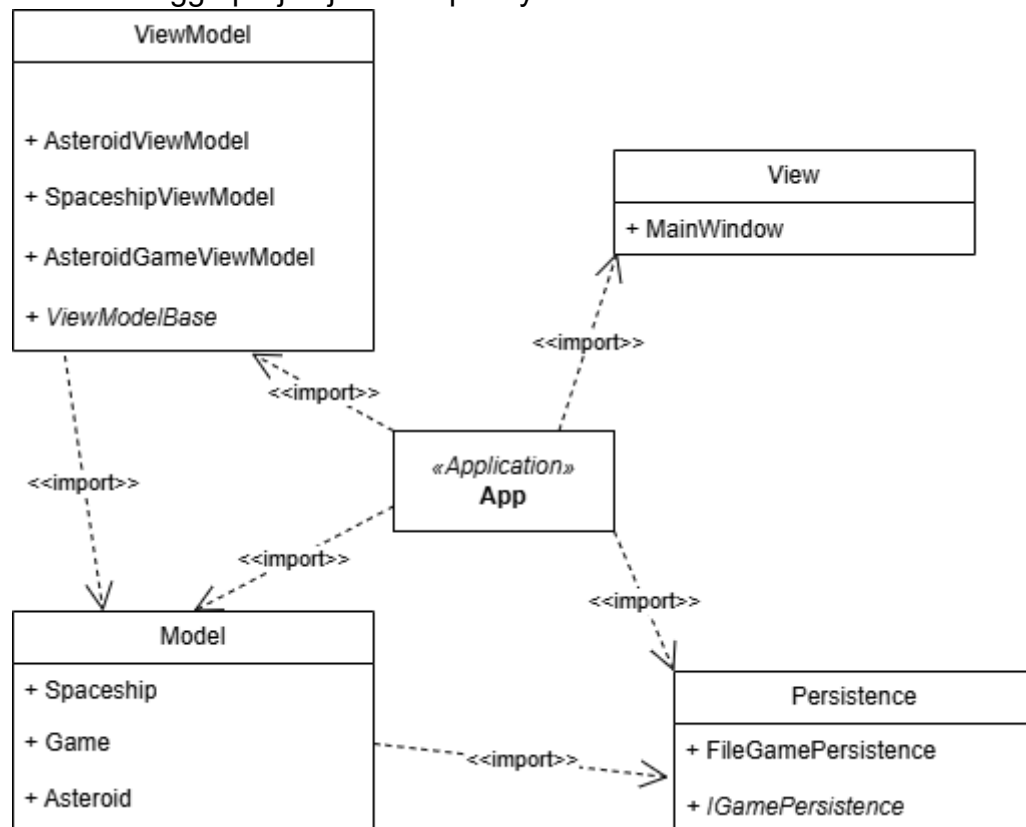
- A képernyő felső sorából folyamatosan közelednek az aszteroidák és folyamatosan ellenőrizzük azt, hogy ütközött-e valamelyik az űrhajóval. Ha igen akkor leállítjuk a stopper, küldünk egy üzenetet a felhasználó számára, hogy „Game Over” a futott idővel együtt.



## 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. A program csomagszerkezete a 2. ábrán látható.
  - A program szerkezetét két projektre osztjuk implementációs megfontolásból: a Persistence és Model csomagok a program felületfüggetlen projektjében, míg a ViewModel és View csomag az

Avalonia függő projektjében kap helyet.



2.ábra: AsteroidGame csomagdiagramja

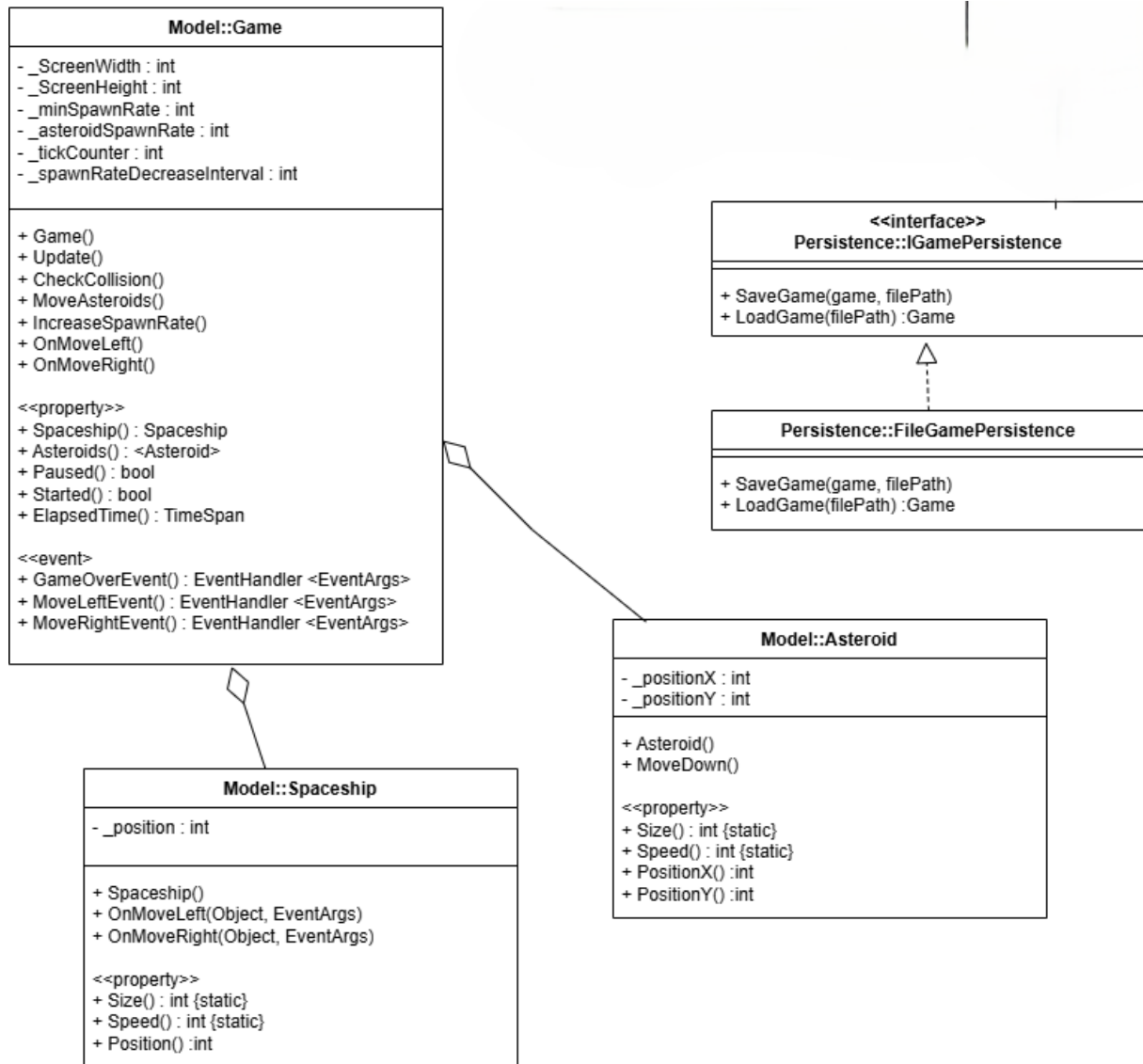
- **Perzisztencia:**

- A hosszú távú adattárolás lehetőségeit az **IGamePersistence** interfész adja meg, amely lehetőséget ad egy játékállás betöltésére (**LoadGameAsync**), valamint mentésére (**SaveGameAsync**). A műveleteket hatékonysági okokból aszinkron módon valósítjuk meg.
- Az interfészt szöveges fájl alapú adatkezelésre a **FileGamePersistence** osztály valósítja meg. A fájlkezelés során fellépő hibákat kivételekkel jelezzük.
- A program az adatokat szöveges fájlként tudja eltárolni, melyek a json kiterjesztést kapják. Ezeket az adatokat a programban bármikor be lehet tölteni, illetve ki lehet menteni az aktuális állást.

- **Modell:**

- A modell lényegi részét a **Game** osztály valósítja meg, amely szabályozza az űrhajó és aszteroidák tevékenységeit, valamint a játék egyéb paramétereit. Ezen az osztályon belül történik a játék állapotának frissítése (**Update**), azaz az űrhajó mozgatása és az aszteroidák ereszkedése. Ezen felül itt ellenőrizzük az esetleges ütközést (**CheckCollision**). A játékmenetben töltött időt is itt tartjuk számon (**ElapsedTime**). A konstruktor pedig az alapértelmezett beállításokat állítja be (pl. **\_asteroidSpawnRate**).

- A Spaceship osztály tárolja az űrhajó pozícióját (**\_position**) és az általános tulajdonságait egy űrhajónak (**Speed**, **Size**). A pozícióját pedig a balra (**OnMoveLeft**) és jobbra (**OnMoveRight**) tudja változtatni.
- Az Asteroid osztály felel egy aszteroida pozíciójáért (**\_positionX**, **\_positionY**) és az aszteroidák általános tulajdonságaiért (**Speed**, **Size**). Az aszteroidák ereszkedése itt történik meg (**MoveDown**)



3.ábra: Model & Perzisztencia

- **Nézet:**

- A nézet tartalmaz egy MainWindow és egy MainView osztályt. A nézet egy Canvas-en fogja kirajzolni az aszteroidákat és az űrhajót. A Canvas mellet pedig egy StackPanel fogja tartalmazni. Minden adatot adatkötéssel kapcsolunk a felülethez.
- Parancsokkal kezeljük a menügombok megnyomását (**NewGame**, **SaveGame**, **LoadGame**), amik csak szüneteltetett állapotba



lekezelése, és ezáltal a játék, az adatkezelés, valamint a nézetek szabályozása.

| <b>App</b> <i>Application</i>  |
|--|
| - _model: Game<br>- _viewModel: ViewModel<br>- _view: MainWindow<br>- _timer: DispatcherTimer  |
| + App()<br>+ InitializeGame(Game?)<br>- App_StartUp(object, StartupEventArgs)<br>- View_Loaded(object, RoutedEventArgs)<br>- Timer_Tick(object, EventArgs)<br>- Model_GameOver(object, EventArgs) {async}<br>- ViewModel_TogglePause(object, EventArgs)<br>- ViewModel_OnMoveRight(object, EventArgs)<br>- ViewModel_OnMoveLeft(object, EventArgs)<br>- ViewModel_NewGame(object, EventArgs)<br>- ViewModel_SaveGame(object, EventArgs) {async}<br>- ViewModel_LoadGame(object, EventArgs) {async} |

#### Project:

| AsteroidGame   |
|--|
| <b>Model:</b><br>- Asteroid.cs<br>- SpaceShip.cs<br>- Game.cs<br><br><b>Persistence:</b><br>- GamePersistence.cs<br>- FileGamePersistence.cs |

| AsteroidGameAvalonia   |
|--|
| <b>ViewModel:</b><br>- ViewModelBase.cs<br>- AsteroidViewModel.cs<br>- SpaceshipViewModel.cs<br>- AsteroidGameViewModel.cs<br><br><b>View:</b><br>- MainWindow.axaml<br>- MainView.axaml |

| AsteroidGameAvalonia.Desktop |
|------------------------------|
| -program.cs                  |

| AsteroidGameAvalonia.Android |
|------------------------------|
| -MainActivity.cs             |

| AsteroidGameWinForms         |
|------------------------------|
| <b>View:</b><br>-GameForm.cs |

| AsteroidGameWPF   |
|---|
| <b>ViewModel:</b><br>-DelegateCommand.cs<br>-ViewModelBase.cs<br>-AsteroidViewModel.cs<br>-SpaceshipViewModel.cs<br>-AsteroidGameViewModel.cs<br><br><b>View:</b><br>-MainWindow.xaml |

## Tesztelés:

- A modell funkcionalitása egységtesztek segítségével lett ellenőrizve a **GameModelTest** osztályban.
- Az alábbi tesztek kerültek megvalósításra:
  - **InicializesCorrectly** – Az alapértelmezett értékek helyességét ellenőrzi mikor létrehozunk egy Game objektumot
  - **MovesLeft** – Az űrhajó balra mozgását teszteli
  - **MovesRight** – Az űrhajó jobbra mozgását teszteli
  - **CannotMoveBeyondLeftBoundary** – Azt ellenőrzi, hogy az űrhajó nem tud a bal oldali határon túlmenni
  - **CannotMoveBeyondRightBoundary** – Azt ellenőrzi, hogy az űrhajó nem tud a jobb oldali határon túlmenni
  - **AsteroidSpawn** – Az aszteroidák helyes spawnolását ellenőrzi
  - **GameEndsWithCollision** – Azt teszteli, hogy mikor egy aszteroida koordinátája az aszteroida hitbox-án belül van akkor a játék véget ér.
  - **CannotMoveWhenPaused** – Teszteli, hogy szüneteltetés közben tudunk-e mozogni.