

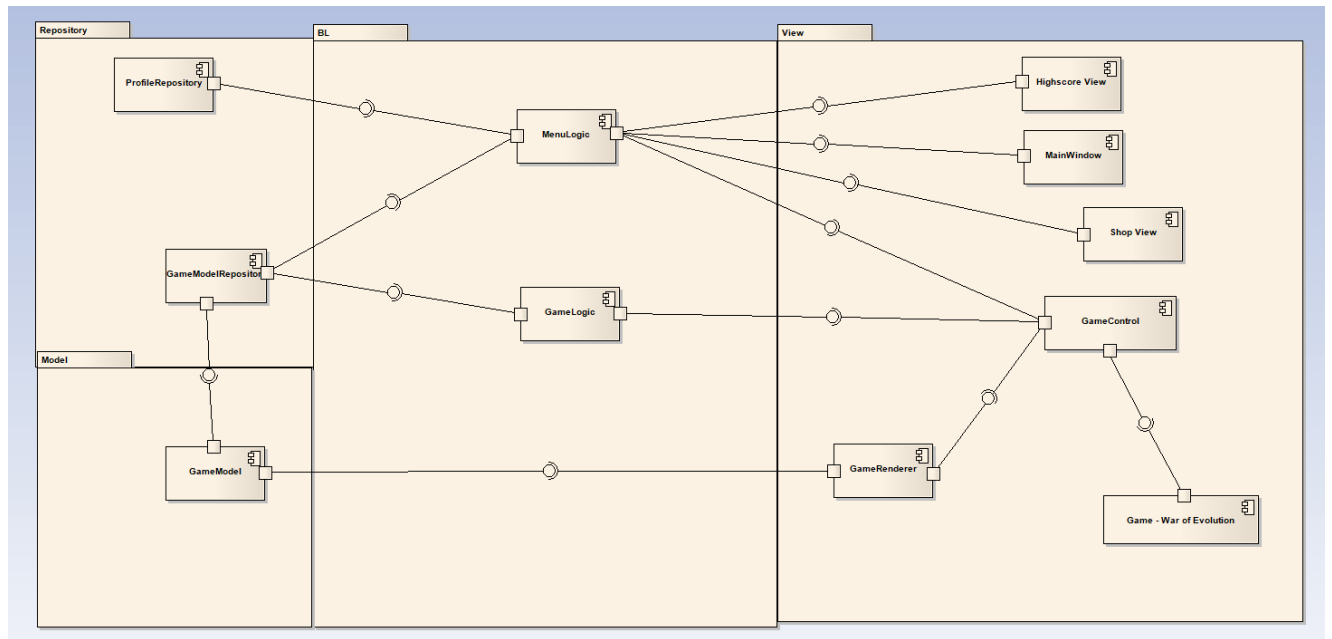
# War of Evolution

*Csuzdi Bence (BJ0R1Q) & Nagy Noel (UR1OPG)*

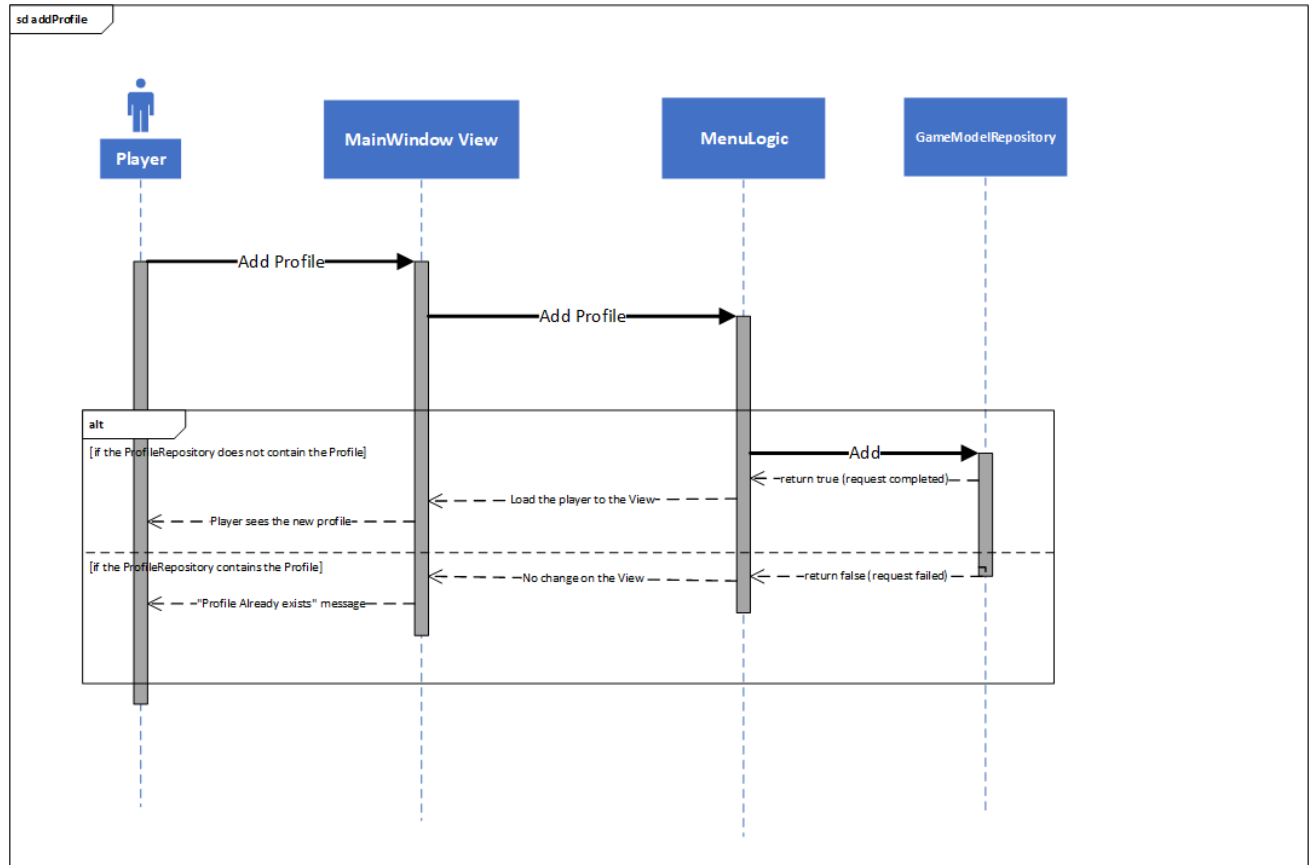
Fejlesztői dokumentáció

- A játékos egy általa kiválasztott profillal játszik, mely, ha még nem szerepel a játékban létre tudja hozni, saját névvel, és a játék elejére mutató kezdőértékkel.
- A játékot elindítva az első pályára kerül a játékos a karakterével.
- A karakternek a kiválasztott profiltól függően van fegyvere minden pályán, valamint pályától függően eredménye.
- A karakter mozgatása:
  - A billentyű: a karakterrel való bal irányú mozgás
  - D billentyű: a karakterrel való jobb irányú mozgás
  - Space billentyű: a karakterrel való ugrás, mely dupla is lehet
  - E billentyű: a fegyver váltása a következő elérhetőre
- A pályán ellenségek lelhetőek fel, amelyeknek két fajta fegyver közül van az egyik: közelharcos, távolharcos
  - Közelharcos esetén: Addig közeledik az ellenség a játékoshoz, amíg nem tudja a fegyverével megsebezni.
  - Távolharcos esetén: Addig közeledik az ellenség, amíg a fegyverét el nem tudja a játékos fele hajítani.
- A karakter fegyverének használata:
  - Közelharcos fegyver esetén: Az egér bal gombjának lenyomásával történik a fegyver használata, és ha olyan pozícióban van, az ellenség sebzése.
  - Távolharcos fegyver esetén: Az egér jobb gombjának lenyomásával történik a fegyver használata, ha eltalál egy ellenséget, akkor az az ellenség sebződik, és a játékos kezébe visszakerül a fegyver.
- Minden egyes megölt ellenségért pontot kap a játékos.
- A pályákon fellelhető akadályok a kilőtt fegyvert blokkolják, amikor ez sorra kerül, a kilőtt fegyver visszakerül a játékos kezébe.
- Az pálya végén pedig egy portál található, melynek elérése jelenti a pálya sikeres teljesítését.
- Ha a játékos élete elfogy, a játéknak vége.

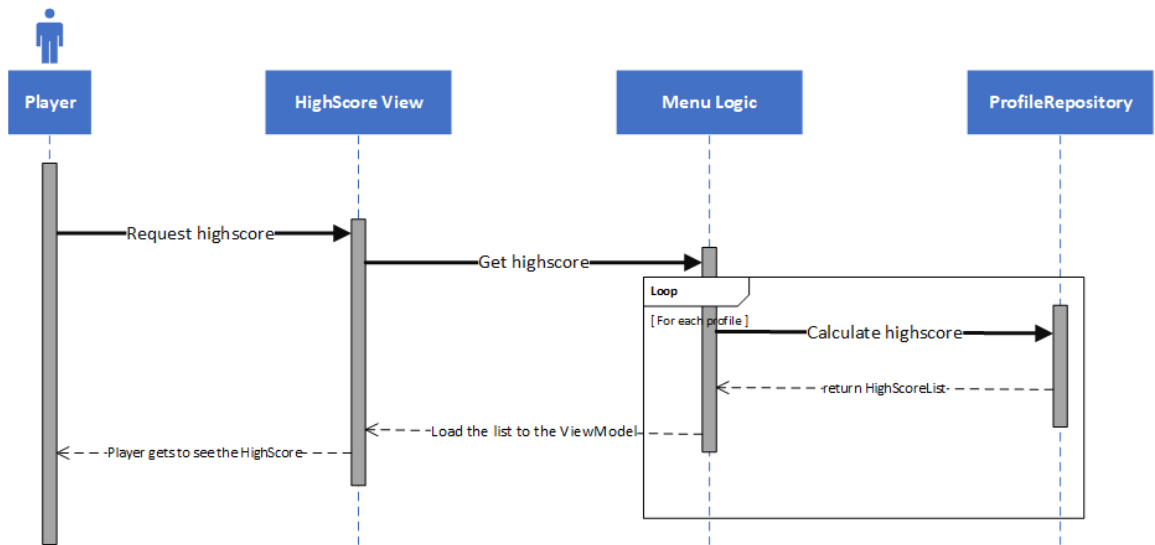
## Component diagram:

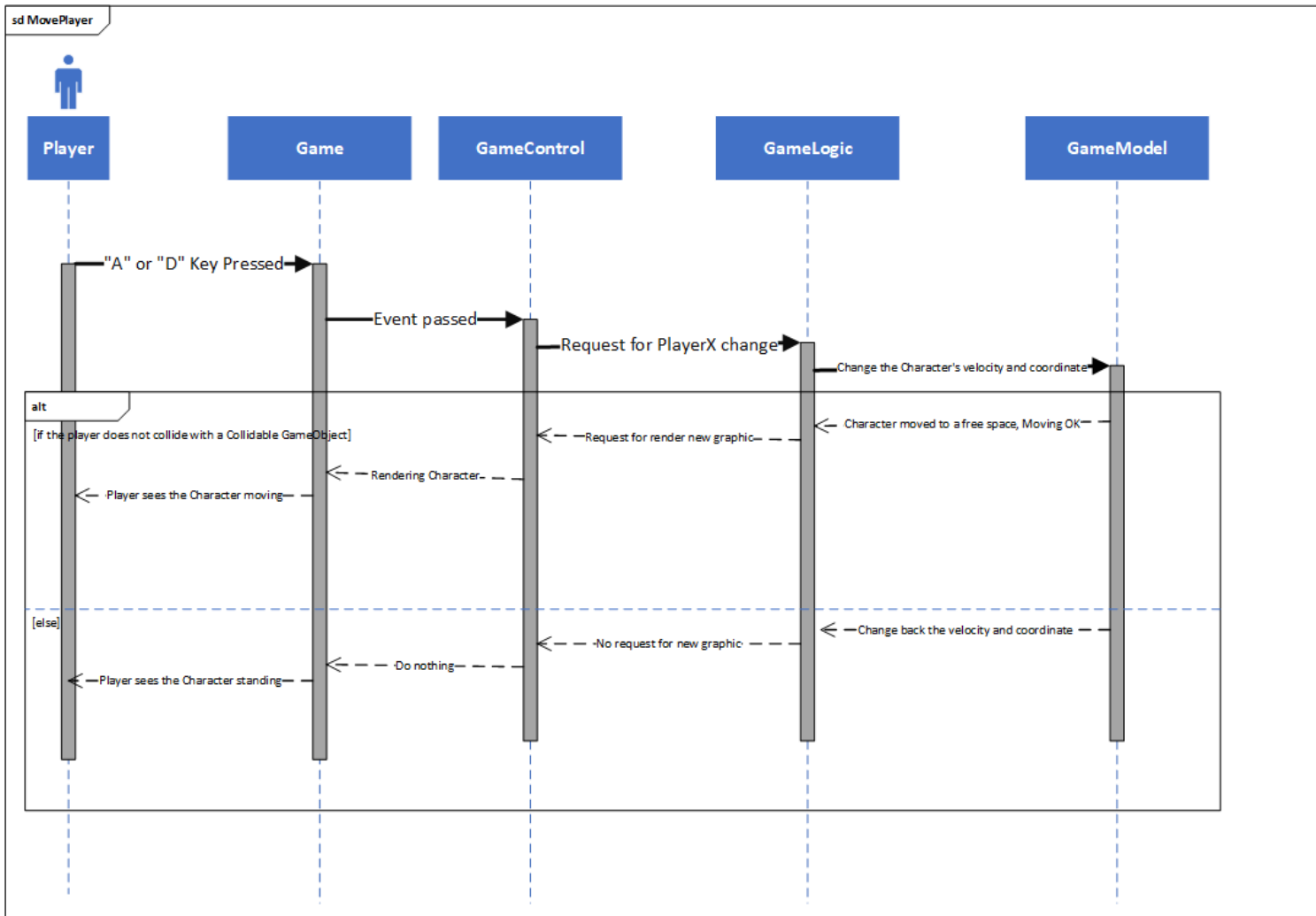


## Sequence Diagrams:



sd GetHighscore





Interface-k (A Bitbucket repo-ban is megtalálhatóak – merge commit 2bb1942)

Model:

- IGameModel
  - Player Player { `get`; `set`; }
  - ICollection<Enemy> Enemies { `get`; `set`; }
  - Level Level { `get`; `set`; }
  - `double` GameWidth { `get`; `set`; }
  - `double` GameHeight { `get`; `set`; }

A model osztályok létre lettek hozva, implementálva.

Repository:

- IRepository<T>
  - ICollection<T> GetItems { `get`; }
  - `void` LoadData();

Ez egy olyan generikus interface, mely bármilyen adat olvasására szolgál (írásra viszont nem).

- IWritableRepository<T> : IRepository<T>
  - `void` WriteData(ICollection<T> collection);

Megvalósítja az olvasást, kiegészül egy írással. Ez a későbbi esetleges bővítéseket segítette. Ezt az interface-t a ProfileRepository valósítja meg, illetve a játékon kívül is el kell érni, így független a játék logikájától.

- IGameModelRepository
  - IGameModel LoadGameModel(MyProfile profile, `int` levelNumber);
  - `void` SaveGameModel(IGameModel gameModel);
  - `void` GenerateEnemies(GameModel gameModel);

Dedikált GameModel olvasó interface, játékállapot betöltésére szolgál. A GameModelRepository egy Profil és pálya alapján tud majd betölteni egy GameModel-t és az visszaadódik majd a Repository-n keresztül.

Logic:

- IGameLogic
  - void PlayerMoveHorizontal();
  - void MoveEnemy(Enemy enemy);
  - void PlayerMoveVertical();
  - bool IsEndOfGame();
  - bool PlayerAlive();
  - bool AnyoneAlive();
  - void ReleaseMeleeWeapon();
  - void StopPlayer();
  - void PlayerTick();
  - void PlayerThrow();
  - void AddGold(int bonusGold);
  - void GenerateEnemies();
  - void MoveToRight();
  - void SaveProfileData(int level);
  - void MoveToLeft();
  - void ChangeWeapon();
  - string WidhtHelper(Enemy enemy);

A játék logikáját írja le, később még változhat, implementálás közben felmerülhet új feature.

- IMenuLogic
  - List<MyProfile> CalculateHighScores();
  - bool IsBuyable(MyProfile myProfile, Weapon weapon);
  - List<MyProfile> ListAllProfiles();

Játék logikájától függetlenített menü logikai interface.



## Class diagrams:

