

Készítette

Kis Gergely Domonkos (VMT982)

E-mail: tianarath30@gmail.com, vmt982@inf.elte.hu

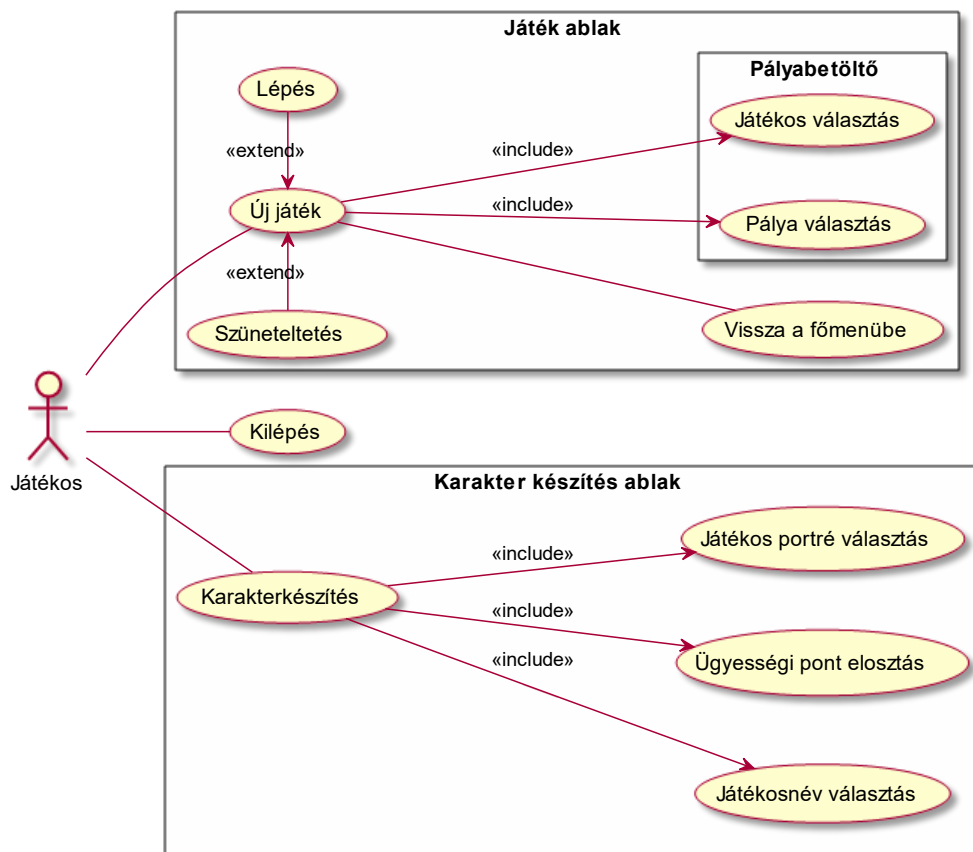
Csoportszám: 6

Feladat

Készítsünk programot, amellyel a következő játékot játszhatjuk. Adott egy $n \times n$ elemből álló **játékpálya**, amely **labirintusként** épül fel, azaz fal, illetve padló mezők találhatóak benne, illetve egy kijárat a jobb felső sarokban. A játék célja az, hogy egy játékost minél előbb kivezessünk a labirintusból a bal alsó sarokból indulva. 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 programot legalább 3 különböző méretű pályával lehessen használni. A program biztosítson lehetőséget új játék kezdetére 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.

A feladat elemzése

A játékban kizárólag egy játékos vesz részt, aki az alkalmazás indítását követően a **főmenüben** alábbi tevékenységeket végezheti el:



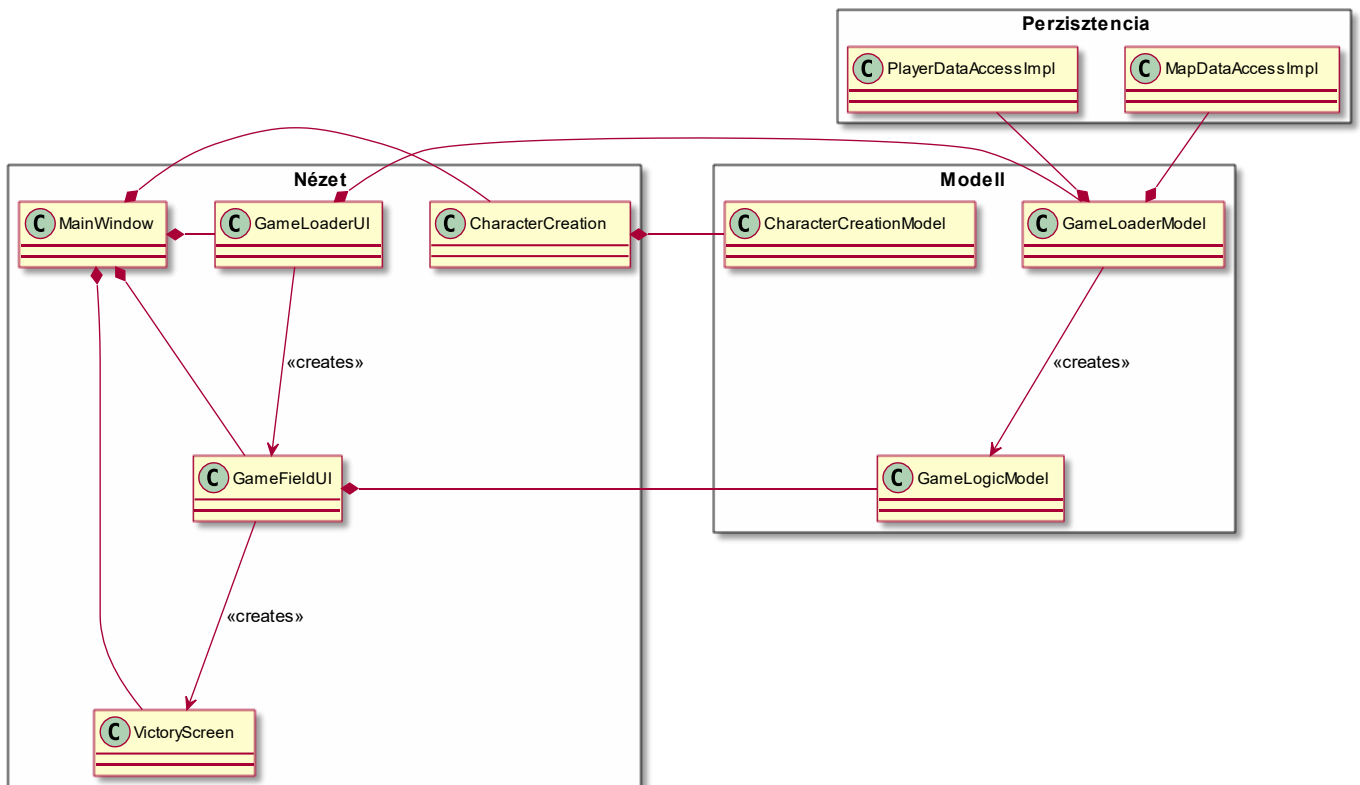
A felhasználói tevékenységek során az alábbi esetek következhetnek be.

1	Alkalmazás indítása	GIVEN:	Az alkalmazás telepítve van
		WHEN:	Az alkalmazás indítása
		THEN:	Megjelenik a főmenü
2	Kilépés	GIVEN:	A főmenü aktív
		WHEN:	A „ Kilépés ” feliratú gombra kattintunk
		THEN:	Az alkalmazás bezáródik
3a	Új játék indítása	GIVEN:	A főmenü aktív
		WHEN:	Az „ Új Játék ” feliratú gombra kattintunk
		THEN:	Megnyílik a játék betöltő képernyő
3b	Új játék indítása (Pálya választás)	GIVEN:	A játékbetöltő képernyő aktív
		WHEN:	A pályaválasztó listában egy mezőre kattintunk
		THEN:	Kiválasztódik a pálya a játékunkra
3c	Új játék indítása (Játékos választás)	GIVEN:	A játékbetöltő képernyő aktív
		WHEN:	Az aktív játékos alatti „>” vagy „<” feliratú gombokra kattintunk vagy a billentyűzet jobb és bal nyíl gombjait lenyomjuk
		THEN:	Az aktív játékos a következő/előző játékos lesz a játékosok közül
4d	Új játék indítása (véglegesítés)	GIVEN:	A játékbetöltő képernyő aktív, van kiválasztott játékos és pálya
		WHEN:	A „ Játék ” feliratú gombra kattintunk
		THEN:	Becsukódik a játékbetöltő képernyő és megnyílik a játéktér
5	Lépés	GIVEN:	A játéktér aktív, és a játék nincs szüneteltetve
		WHEN:	<ul style="list-style-type: none"> A navigációs felület egy gombjára kattintunk, vagy az azoknak megfelelő gyorsgombokat lenyomjuk (billentyűzetten a nyilak) Az elmozdulás irányában nincs a játékos előtt fal blokk és nem lépne ki a játéktérről
		THEN:	A játékos az adott irányba lép

5b	Lépés	GIVEN:	A játéktér aktív, és a játék nincs szüneteltetve
		WHEN:	<ul style="list-style-type: none"> A navigációs felület egy gombjára kattintunk, vagy az azoknak megfelelő gyorsgombokat lenyomjuk (billentyűzetten a nyilak) Az elmozdulás irányában fal blokk vagy a játéktér széle van
		THEN:	A játékos helyben marad, nem lép semerre
5c	Lépés	GIVEN:	A játéktér aktív, és a játék nincs szüneteltetve
		WHEN:	<ul style="list-style-type: none"> A navigációs felület egy gombjára kattintunk, vagy az azoknak megfelelő gyorsgombokat lenyomjuk (billentyűzetten a nyilak) Az elmozdulás irányában a kijárat blokk van
		THEN:	A játékos megnyerte a játékot, a játéktér bezáródik, és megnyílik a játék vége képernyő
6	Szüneteltetés	GIVEN:	A játéktér aktív
		WHEN:	A „PAUSE” gombot megnyomjuk a billentyűzetten
		THEN:	A játék szünetel, a játékidőzítő leáll, és nem lehet lépni egészen addig, amíg újra le nem nyomjuk a „PAUSE” billentyűt és újraindítva a játékot
7	Vissza a főmenübe	GIVEN:	A játék vége képernyő aktív
		WHEN:	A „Vissza a főmenübe” feliratú gombra kattintunk
		THEN:	Az alkalmazás visszatér a főmenübe

Architektúra

Az alkalmazást három rétegű (nézet-modell-perzisztencia) architektúrában valósítjuk meg. Minden ablaknak megfelel egy osztály, amely modell-nézet rétegekre bomlik.

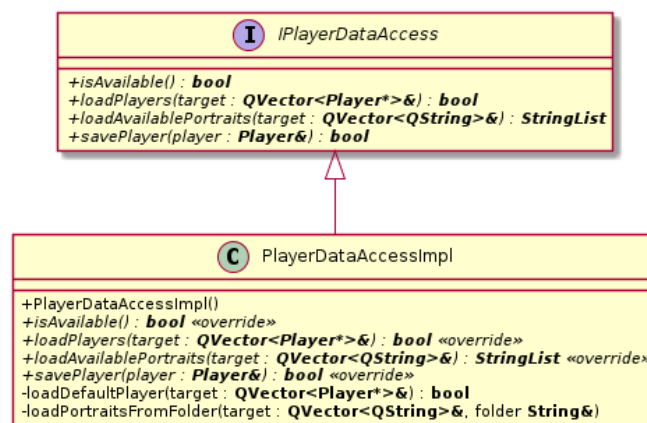


Perzisztencia

Az adatelérési réteg 2 interfészből és az őket implementáló egy-egy osztályból áll. A pályák adatait egy-egy szöveges fájlban, a játékosok adatait pedig egy szöveges fájlban és egy hozzátartozó képfájlban tároljuk. A modellnek nem kell tudnia a tárolás mikéntjéről, bármely az interfészeket helyesen megvalósító osztályt be lehet injektálni a játéketöltő (`GameLoaderModel`) osztályba.

PlayerDataAccessImpl

Osztálydiagram

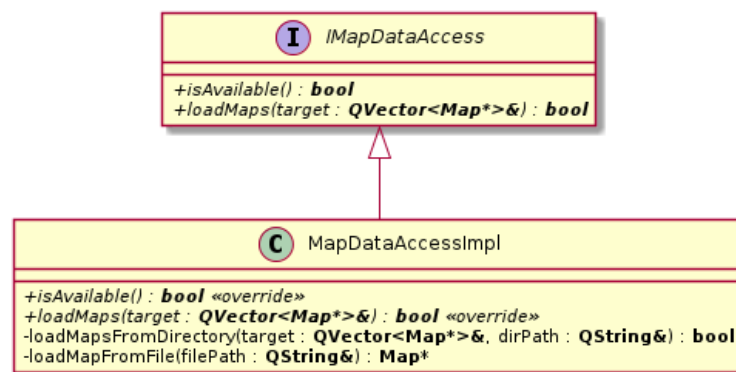


Metódusok

- Az „**isAvailable()**” metódus jelen esetben mindig igazat ad vissza hiszen az erőforrás (saját számítógépünk) mindig elérhető
- A „**loadPlayers()**” metódus betölti az összes „**./Players/**” mappában található játékost és a beégetett alapértelmezett játékost a cél konténerbe.
- A „**loadAvailablePortraits()**” betölti az összes „**./Players/Portraits/**” mappában található (és alapértelmezetten beégetett) játékos portré kép elérési útvonalát a célkonténerbe.
- A „**savePlayer()**” metódus az adott játékos objektumot fájlba menti a „**./Players/**” mappában a játékos nevével megegyező nevű fájlba
- A „**loadDefaultPlayer()**” metódus betölti a célkonténerbe a beégetett alapértelmezett játékost.
- A „**loadPortraitsFromFolder()**” metódus betölti a játékos portrék elérési útvonalait a paraméterként kapott mappából egy célkonténerbe

MapDataAccessImpl

Osztálydiagram



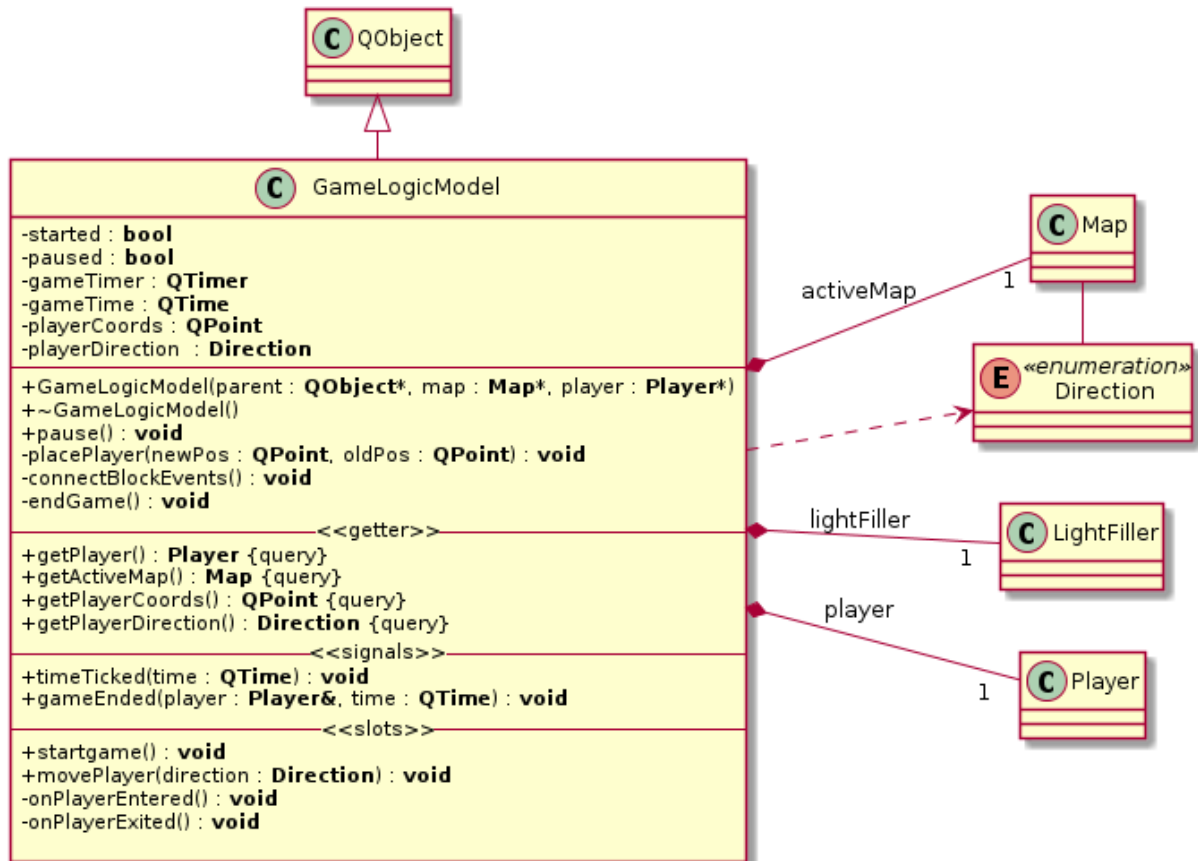
Metódusok

- Az „**isAvailable()**” metódus jelen esetben mindig igazat ad vissza hiszen az erőforrás (saját számítógépünk) mindig elérhető
- A „**loadMaps()**” metódus betölti a „**./Maps/**” mappából és a beégetett mappából a talált pályákat a célkonténerbe.
- A „**LoadMapsFromFirectory()**” metódus betölti a paraméterként kapott mappából a fájlokat a célkonténerbe
- A „**LoadMapFromFile()**” metódus egy konkrét fájlból tölt be egy pályát.

Modell

GameLogicModel

Osztálydiagram



Adattagok

- Eltároljuk az adott játékost (kiegészítve aktuális pozíciójával és irányával), az aktuális pályát, valamint számon tartjuk a játék szüneteltettségét és elindultságát.
- Adattagként felveszünk egy LightFiller típusú objektumot ami egy adott pálya világítását képes végezni

Metódusok

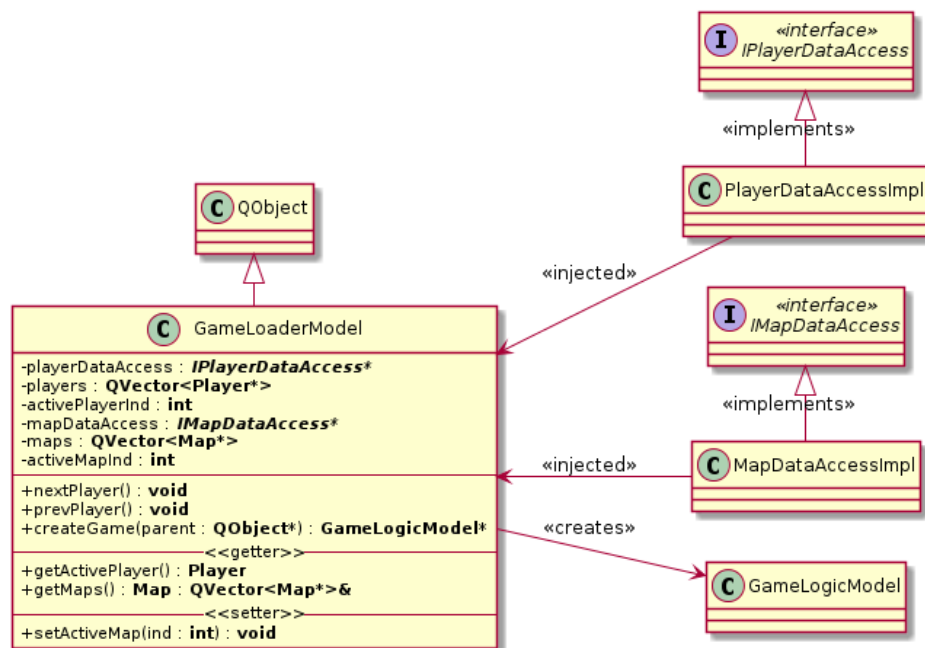
- A „**PlacePlayer()**” metódus egy régi pozícióról egy újra helyezi a játékost amennyiben a célmező engedi a belépést.
- A „**connectBlockEvents()**” metódus összeköti a játéklemezők eseményeit (belépés, kilépés) a játék ezeknek megfelelő eseménykezelőivel (pl.: győzelem ellenőrzése)
- Kifelé a „**movePlayer()**” és a „**pause()**” metódus látszik, ezek elegendőek a logika nézettel való kommunikációjára
- A további metódusok nem szorulnak magyarázatra

Eseménykezelés

- A játéklelogika a játéklemezők játék belépés és kilépés eseményeit kezelik/kezelhetik le.

GameLoaderModel

Osztálydiagram

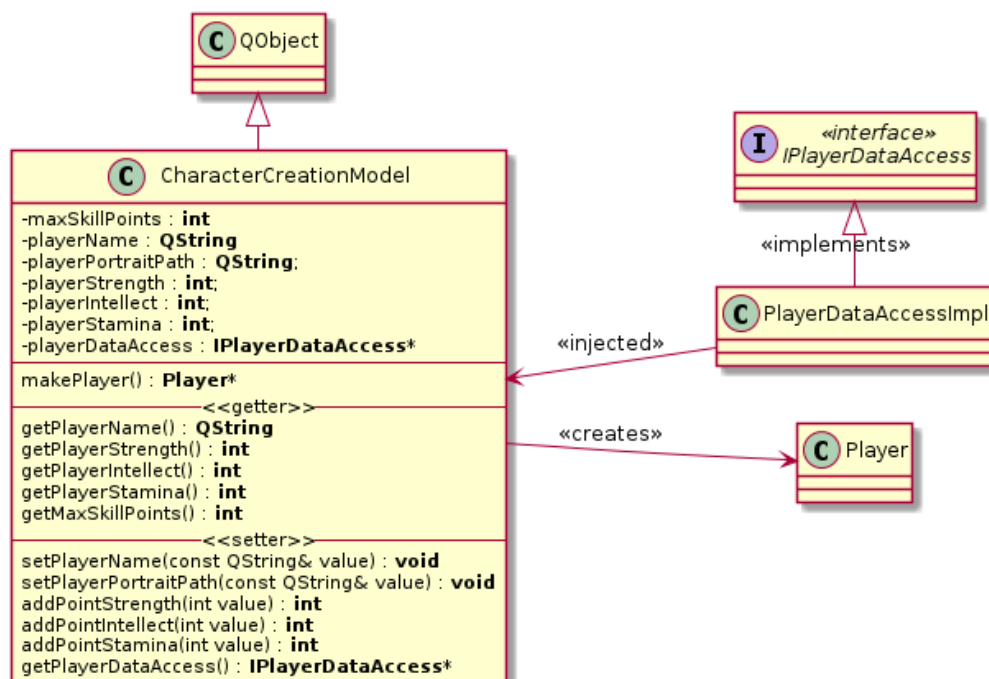


Metódusok

- Az osztály főleg Data Binding (összekapcsolt getter-setter) metódusokat tartalmaz ezek magyarázatra nem szorulnak.
- Az egyetlen lényeges metódus a „**createGame()**” gyártófüggvény, amely egy GameLogicModel típusú játék példányt példányosít a betöltött adatokból.

CharacterCreationModel

Osztálydiagram

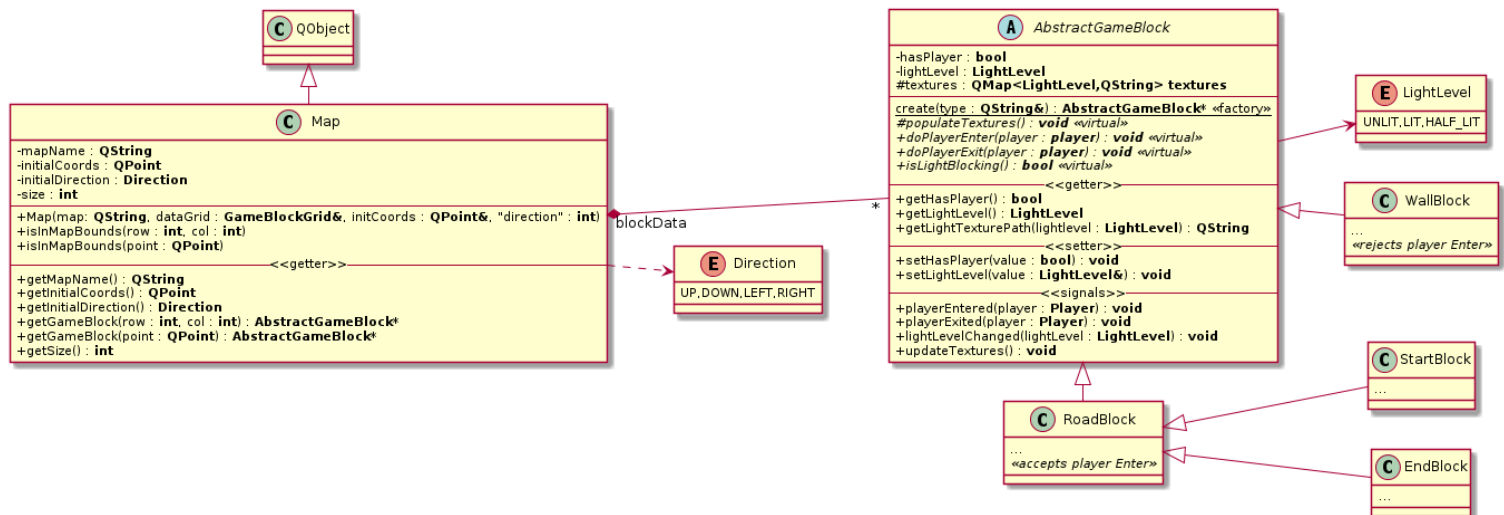


Metódusok

- Az osztály főleg Data Binding (összekapcsolt getter-setter) metódusokat tartalmaz ezek magyarázatra nem szorulnak.
- Az egyetlen lényeges metódus a „**makePlayer()**” gyártófüggvény, amely egy Player típusú játékos példányt példányosít, (majd ment a játékos könyvtárba).

„Map” és a hozzákapcsolódó mezőtípusok

Osztálydiagram

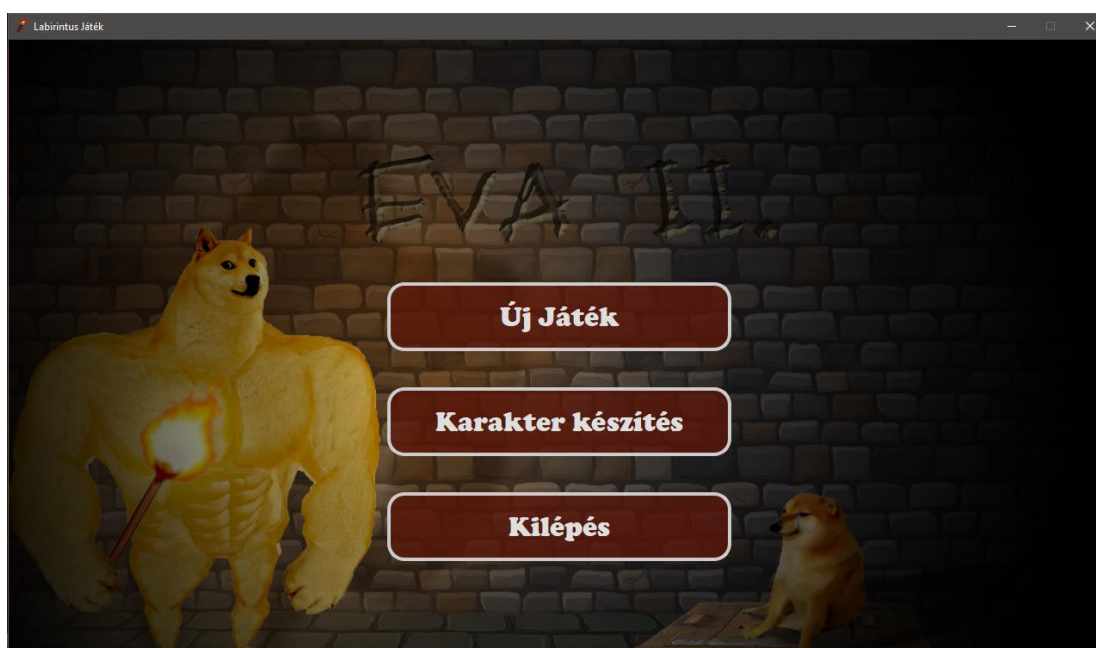


- Az osztályok egyszerű adatkezelő metódusokkal és adattagokkal rendelkeznek, ezek a kommentelt kód alapján és azonosítóik alapján könnyen megérthetők.

Nézet

A felhasználói felület terve:

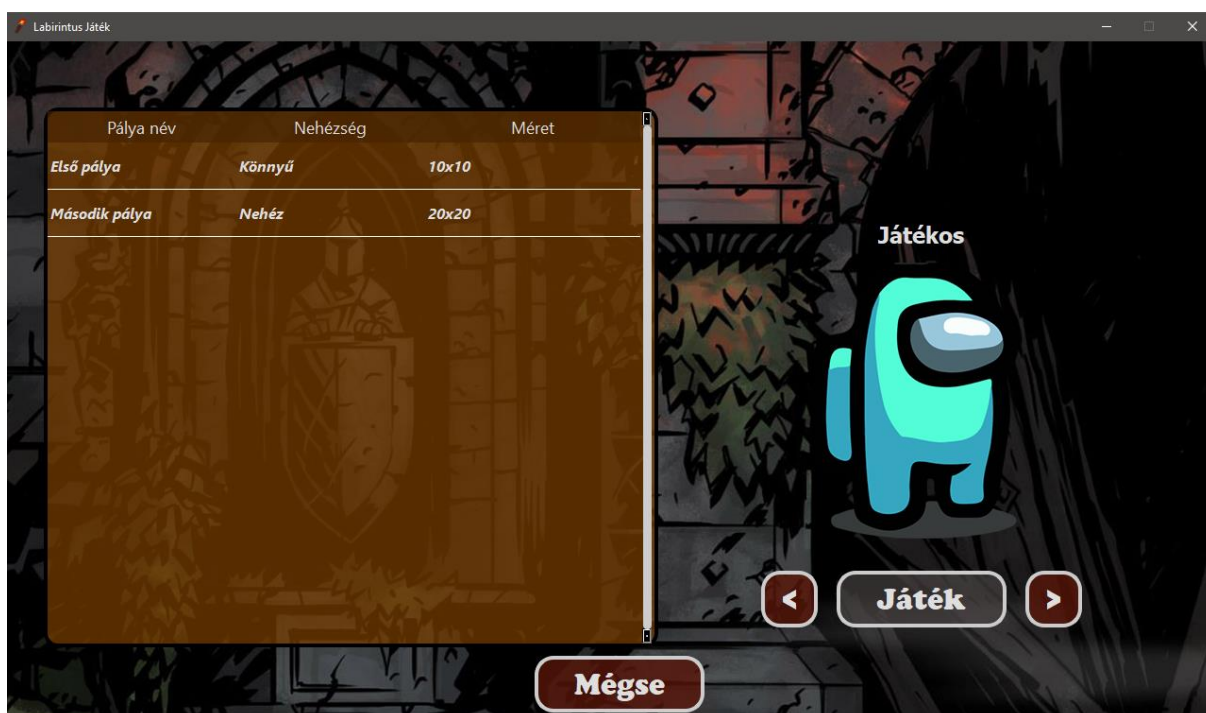
Főmenü



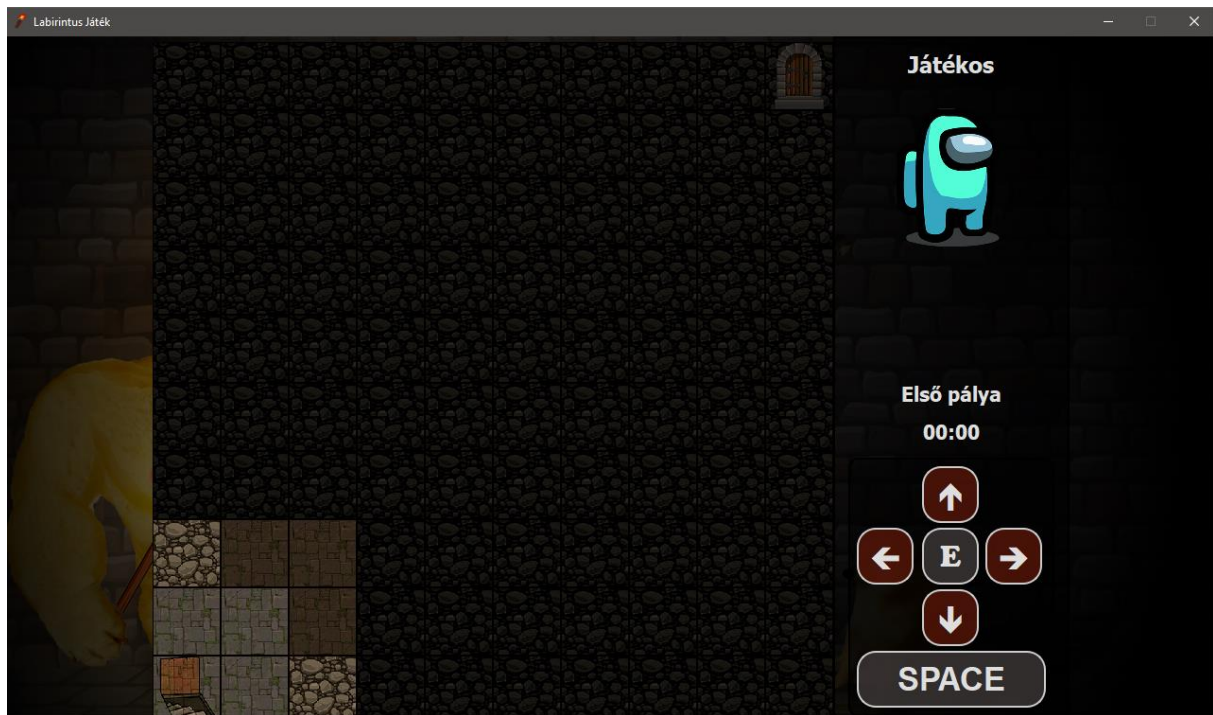
Karakterkészítő ablak



Játékbetöltő képernyő



Játéktér



Eseménykezelés

A nyíl feliratú gombok használhatóak a billentyűzet nyíl billentyűivel, valamint a játék közben a PAUSE gomb megnyomásával lehet szüneteltetni a játékot

<i>sender</i>	<i>signal</i>	<i>receiver</i>	<i>slot</i>
newgameButton	clicked	MainWindow	openNewGame()
charCreateButton	clicked	MainWindow	openCharacterCreation()
quitButton	clicked	MainWindow	close()
mapsTable	clicked	gameLoaderModel	setActiveMap()
playBtn	clicked	gameLoaderModel	create()
game	gameended	gamefieldUI	endgame()
game	timerTick	gameFieldUI	updateTime()
AbstractGameBlock	playerEntered	game	onPlayerEntered()
AbstractGameBlock	playerEntered	AbstractGameBlockWidget	onPlayerEntered()
AbstractGameBlock	playerExited	game	onPlayerExited()
GameLoaderModel	gameLoaded	MainWindow	openGameFieldUI()
charCreateBtn	clicked	playerDataAccess	savePlayer()
GamefieldUI	navBtn	game	movePlayer()
game	startGame	gameFieldUI	onStartGame()

Végfelhasználói tesztesetek:

	Teszteset	Elvárt hatás
1	Csupa út pálya (+start+end) helyes inicializálása	Pálya blokkok megfelelőek
2	Csupa út pálya helyes kivilágítása	Minden blokk maximális világítású
3	Csupa út pálya sikeres elmozdulás (lépés)	A játékos a megfelelő irányba mozog
4	Csupa út pálya sikertelen elmozdulás (lépés) pályahatáron	A játékos nem mozog
5	Csupa fal pálya helyes inicializálás	Pálya blokkok megfelelőek
6	Csupa fal pálya helyes kivilágítása	Csak a játékos körüli blokkok világosak a többi sötét
7	Csupa fal pálya sikertelen elmozdulás (falba ütközés)	A játékos nem mozog
8	Játék vége tesztelése	endGame() szignál kiváltva