

2 Követelmény, projekt, funkcionalitás

2.1 Bevezetés

2.1.1 Cél

A dokumentum célja a szoftver projekt laboratórium csoportos házi feladat elkészítésének ütemezése, nyomon követése, a feladattal szemben támasztott követelmények pontos megfogalmazása, a program ismertetése, specifikálása. A dokumentumban leírtak támpontot nyújtanak a félévi munkához.

2.1.2 Szakterület

Jelen szoftver egy logikai elemekkel teletűzdelt játék. Ebből fakadóan nem speciális célra, csak szórakoztatásra készül.

2.1.3 Definíciók, rövidítések

ZPM: Zero point module. Egy energia tároló egység.

BME: Budapesti Műszaki és Gazdaságtudományi Egyetem

HSZK: Hallgatói Számítógép Központ

JAVA: Programozási nyelv

JRE: Java futtatókörnyezet

IIT: Irányítástechnika és Informatika Tanszék

UML: Unified Modeling Language

2.1.4 Hivatkozások

[1] <https://freedcamp.com>

[2] <https://www.iit.bme.hu/~projlab/>

[3] <https://www.iit.bme.hu/~projlab/feladat.shtml>

[4] <https://marketplace.eclipse.org/content/uml-java-generator>

[5] <https://github.com/>

[6] <https://eclipse.org/>

[7] https://www.google.com/intl/hu_hu/drive/

[8] <https://www.google.hu/intl/hu/docs/about/>

[9] www.skype.com

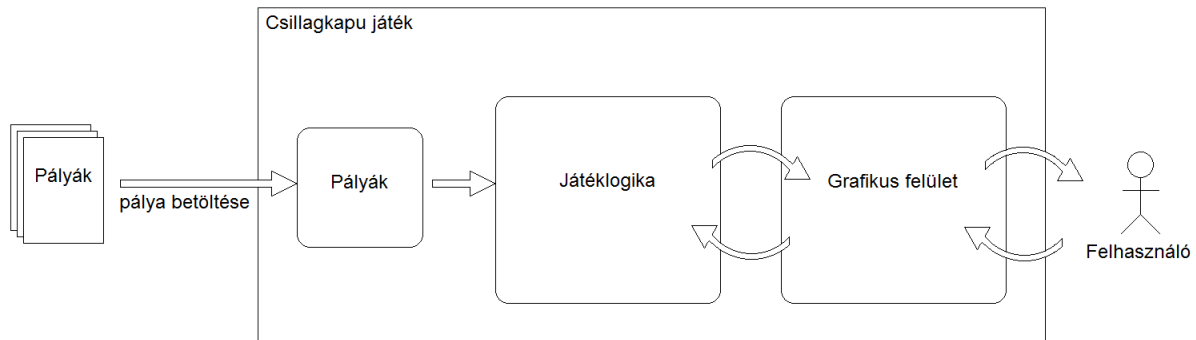
[10] <http://www.gimp.hu/>

2.1.5 Összefoglalás

A szoftver alapvető funkcióinak egy nagyon absztrakt megfogalmazása, alapvető funkcióinak ismertetése, továbbá a korlátozások meghatározása és a határidők megadása. A használt eszközök ismertetése.

2.2 Áttekintés

2.2.1 Általános áttekintés



A fontosabb alrendszerek a következők:

Grafikus felület: A program azon része, mellyel a felhasználó közvetlenül érintkezik. Ennek az alrendszernek a segítségével irányítja a játék menetét, navigálja O'Neill ezredes a pályán, veszi fel a ZPM-eket, mozgatja a dobozokat, csillagkapukat, féregjáratokat hoz létre.

Játéklógika: Ez az alrendszer a legfontosabb a játék szempontjából, ez futtatja magát a játékot. A betöltött pályán mozgatja a játékosokat. Ez az alrendszer tartja számon a még összegyűjtendő ZPM-ek számát és helyét a pályán, továbbá ez tartja számon a különböző színű kilőtt csillagkapuk helyét, felügyeli, hogy egy bizonyos színű csillagkapuból egyszerre csak egy legyen nyitva, ellenőrzi, hogy a kilőtt lövedék valóban speciális falon ér-e célba, és ezáltal létre jöhet-e csillagkapu. A Játéklógika felelőssége a csillagkapuk közt létrejövő féregjáratok kezelése: a féregjáraton - ha a felhasználó úgy szeretné - O'Neill ezredes, Jaffa és a Replikátor közlekedtetése, és féregjáratok nyitása, és bezárása. A Játéklógika alrendszer figyeli a szakadékokat is, és a bele kerülő tárgyak megsemmisítéséről is neki kell gondoskodnia, valamint a játék befejezéséről amennyiben az ezredes vagy Jaffa lép bele a szakadékba. Replikátor esetén a szakadék úttá alakítása. Figyelnie kell még a mérlegek állapotát, ha valamely mérlegre megfelelő nagyságú súly kerül, a Játéklógikának kell gondoskodnia a megfelelő ajtó nyitásáról, és zárásáról, amikor a súly lekerül a mérlegről. A Játéklógika feladatai közé tartozik még a pályán elhelyezett dobozok kezelése is.

Pályák: Betölti az előre elkészített és kiválasztott pályát, betöltés után a Játéklógika használja fel. A pályákat fájlrendszerben tároljuk.

2.2.2 Funkciók

A projekt célja egy többjátékos logikai játékszoftver elkészítése. A két játékos egy-egy karaktert (O'Neil ezredes és egy Jaffa) irányít felülnézetből egy kétdimenziós játéktérben, a céljuk pedig az, hogy a pályán található összes összegyűjtendő tárgyat (ZPM modul) megszerezzék. A játék akkor ér véget, ha valamelyik játékos karaktere meghal vagy az összes ZPM modult összegyűjtötték a pályáról. A nyertes az a játékos, akinek karaktere életben marad vagy amelyik több ZPM modult gyűjtött össze.

A játéktér a vízszintes és függőleges tengelyek mentén egyenlő méretű négyzetekre van osztva. A játék mechanikájának szempontjából a mező legfontosabb tulajdonsága, hogy a

játékos át tud-e rajta haladni vagy sem. Ezt a mező típusa és aktuális állapota minden pillanatban egyértelműen meghatározza.

Ezen kívül - ha a típusa megengedi - lehetnek a mezőn különböző tárgyak is. A játékos ezekkel interakcióba tud lépni, megváltoztatva azok pozícióját, állapotát és bizonyos esetekben rajtuk keresztül a játékban szereplő többi objektum állapotát is.

A fő mezőtípusok a következők:

- **Fal:** Áthatolhatatlan mező. Nem lehet rajta tárgy.
- **Út:** Átjárható mező. Lehetnek rajta tárgyak.
- **Szakadék:** Nem áthatolhatatlan, de a rátett tárgyak megsemmisülnek. Ha a játékos érkezik rá, akkor a játékos karaktere meghal és vége a játéknak. Ha Replikátor érkezik rá, akkor a szakadék úttá alakul át.
- **Ajtó:** Kétállapotú mezőtípus, lehet nyitott vagy zárt. Nyitott állapotában falként, zárt állapotában útként viselkedik. Állapotát egy speciális nyitómechanizmus (mérleg) segítségével befolyásolhatja a játékos. Amennyiben az ajtó bezárulásakor egy játékos áll az ajtóban, és rácsukodik a játékosra, akkor a játékos meghal, a játék pedig véget ér.
- **Mérleg:** Az ajtóhoz tartozó nyitómechanizmus. Ha a mezőn tartózkodó, súllyal rendelkező objektumok (a játékos karaktere, dobozok) összsúlya meghaladja a mérleg súlykorlátját, akkor a hozzá tartozó ajtó nyitott állapotba kerül. Ha az összsúly a súlykorlát alá csökken, az ajtó bezárul.
- **Speciális fal:** A fallal szinte minden szempontból megegyező mezőtípus, azonban a határoló élein a játékos csillagkapukat hozhat létre (lásd később).

A mezőkön lévő tárgyak a következő típusúak lehetnek:

- **Doboz:** Súllyal rendelkező tárgy, amelyet a mérlegre helyezve a játékos az ajtót nyitott állapotban tarthatja. A játékos képes a közvetlenül előtte lévő mezőn található dobozt felvenni. Ilyenkor a doboz a játékoskal együtt mozog, akár féregjáratokon keresztül is. A játékos hasonló módon képes a dobozt lerakni a közvetlenül előtte lévő mezőre, ha a mező típusa megengedi a tárgyak lehelyezését. A doboz lehelyezése féregjáraton keresztül a játékos mozgásával azonos logika alapján működik. A szakadékba lehelyezett vagy bezáródó ajtó által összenyomott doboz megsemmisül és az eredeti helyén egy új doboz keletkezik.
- **ZPM modul:** A játék célja ezeknek az összegyűjtése. Ha egy játékos egy ZPM modult tartalmazó mezőre lép, felszedi azt és a modul eltűnik – viszont az ezredes minden második ZPM felvételekor a pálya egy véletlenszerűen kiválasztott mezőjén egy új ZPM keletkezik. Ha a játékosok a pályán található összes ZPM modult összegyűjtötték, a játék véget ér, és az a játékos nyer aki több ZPM modult gyűjtött össze.

A labirintus felépítése nem garantálja, hogy bármely két mező között vezet átjárható út. A játékos ilyen esetekben egy különleges eszközt hívhat segítségül - a speciális fegyvert. A játékos ennek segítségével valamelyik irányban egy speciális töltényt lőhet ki. Az így kilőtt töltény vagy sikeresen létrehoz egy portált, vagy pedig megsemmisül attól függően, hogy a játékos pozíciójából kiindulva milyen típusú az első áthatolhatatlan mező. Amennyiben az

nem egy speciális fal, a csillagkapu nem jön létre. Ha viszont igen, a lövedék pályájára merőleges, játékoshoz közelebbi élén létrejön a portál. Ha a speciális lövedék egy olyan éllel ütközik, amelyen már van egy korábban létrehozott csillagkapu, a lövedék ugyanúgy megsemmisül, mintha csak fallal ütközött volna.

A játékosok speciális fegyvere egyenként kétféle (kék és sárga, illetve piros és zöld színű) lövedék kilövésére képes, amelyek meghatározzák a létrehozott csillagkapu színét. Egy színű portálból egyszerre csak egy létezhet a pályán, a következő, azonos színű csillagkapu létrejöttékor a régebben létrehozott kapu megsemmisül. A speciális fegyver töltényei nem fogynak el, bármennyit kilőhet a játékos.

Ha a pályán egyszerre egy színpár mindkét színéből létrejött egy-egy csillagkapu, a két portál között egy féregjárat jön létre. A játékos a féregjáraton keresztül tud haladni, mégpedig a következő logika szerint: ha a játékos az egyik csillagkapun keresztül mozog a speciális fal felé, a másik színű kapuval szomszédos, átjárható mezőre érkezik. A féregjáratba helyezett dobozok pozíciója is e logika alapján dől el.

A pályán kezdetben található továbbá egy Replikátor is. A Replikátor a pálya játékosok által is bejárható részein véletlenszerűen mozog, a játékosokkal és a pályán található tárgyakkal nem lép interakcióba. Azonban ha egy játékos speciális fegyverének lövedékével érintkezik, a Replikátor és a lövedék is megsemmisülnek. Továbbá ha a Replikátor szakadékba esik, betemeti a szakadékot, amely a továbbiakban útként fog viselkedni (a Replikátor ebben az esetben is megsemmisül).

2.2.3 Felhasználók

A szoftver használata alapvető számítógépes ismereteket igényel csupán, ezért gyakorlatilag bárki lehet felhasználó, aki képes elindítani az alkalmazást, kezelni a billentyűzetet és feldolgozni a grafikus interfészt.

2.2.4 Korlátozások

Az elkészítendő szoftverre vonatkozó előírások, korlátozások a következők: HSZK számítógépeinek megfelelő erőforrás követelmények kialakítása, illetve standard Java könyvtárkészlet használata.

2.2.5 Feltételezések, kapcsolatok

A Szoftver projekt labor feladat szövege: <https://www.iit.bme.hu/~projlab/feladat.shtml>

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azono- sító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-Case	Komment
F001	Új játék elindítása	Bemutatás	Alapvető	Megrendelő	Új játék	
F002	Pálya kiválasztás	Bemutatás	Opcionális	Csapat	Pálya kiválasztás	

F003	Játékból kilépés	Bemutató	Alapvető	Megrendelő	Kilépés	
F004	Játékos irányítása	Bemutató	Alapvető	Megrendelő	Mozgás	
F005	Doboz felvétel és letevés	Bemutató	Alapvető	Megrendelő	Doboz interakció	
F006	ZPM begyűjtése	Bemutató	Alapvető	Megrendelő	ZPM összeszedés	Ha az ezredes kettőt összeszedett, egy új ZPM kerül a pálya egy véletlen mezőjére.
F007	Kék, sárga, piros illetve zöld színű portál nyitás	Bemutató	Alapvető	Megrendelő	Lövés	Egy bizonyos színű portálból egyszerre csak egy lehet a pályán.
F008	Szakadékba mozgás az ezredes halálával jár	Bemutató	Alapvető	Megrendelő		
F009	Féregjáraton átmozgás	Bemutató	Alapvető	Megrendelő		Dobozzal együtt is. Kék és sárga, illetve piros és zöld csillagkapuk közt tud mozogni az irányított játékos.
F010	Ha egy doboz a szakadékba esik, megsemmisül	Bemutató	Alapvető	Megrendelő		
F011	Ha súlyt helyezünk a mérlegre, kinyílik a hozzátartozó ajtó.	Bemutató	Alapvető	Megrendelő		A súly lehet az ezredes maga, Jaffa vagy egy doboz (utóbbi 1, előbbiek 2 egységnyi súlyúak). Legalább 2 egységnyi súly kell a nyitáshoz.
F012	A pályák kialakítása olyan, hogy egyes mezők közvetlen nem megközelíthetők	Bemutató	Alapvető	Megrendelő		
F013	A játékot az nyeri, aki több ZPM-et gyűjt, vagy tovább életben marad	Bemutató	Alapvető	Megrendelő		
F014	Ha egy doboz megsemmisül, egy kis idő múlva újra megjelenik az eredeti pozíciójában	Bemutató	Alapvető	Csapat		
F015	Ha a Replikátor szakadékba esik, akkor a szakadék mezőből út lesz	Bemutató	Alapvető	Megrendelő		
F016	Ajtó megöli a játékost és a dobozt	Bemutató	Alapvető	Csapat		Az ajtó bezáruláskor megöli a játékost,

						illetve megsemmisíti a dobozt ha azok éppen akkor rajta tartózkodnak.
F017	Replikátor véletlenszerűen mozog	Bemutatás	Alapvető	Megrendelő		
F018	Ha a Replikátort eltalálja egy speciális lövedék, a Replikátor és a speciális lövedék is megsemmisül.	Bemutatás	Alapvető	Megrendelő, Csapat		

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
E01	HSZK-s számítógép	kiértékelés	alapvető	megrendelő	JRE 1.6+
E02	monitor	nincs	alapvető	csapat	
E03	billentyűzet	nincs	alapvető	csapat	
E04	egér	nincs	alapvető	csapat	
E05	Git	nincs	alapvető	csapat	verziókezelő
E06	Microsoft Word	nincs	alapvető	csapat	dokumentáció szerkesztésére
E07	Eclipse	nincs	alapvető	csapat	
E08	Gimp	nincs	alapvető	csapat	a játék grafikai interfészének elkészítéséhez
E09	Skype	nincs	alapvető	csapat	
E10	UML to Java Generator (Eclipse plugin)	nincs	alapvető	csapat	https://marketplace.eclipse.org/content/uml-java-generator
E11	Google Docs	nincs	alapvető	csapat	
E12	Google Drive	nincs	alapvető	csapat	
E13	Freedcamp	nincs	alapvető	csapat	

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
A001	Követelmény, projekt, funkcionalitás	bemutató	alapvető	megrendelő	Határidő: 2016. február 29.
A002	Analízis modell kidolgozása 1.	bemutató	alapvető	megrendelő	Határidő: 2016.március 7.
A003	Analízis modell kidolgozása 2.	bemutató	alapvető	megrendelő	Határidő: 2016. március 14.
A004	Szkeleton tervek beadása	bemutató	alapvető	megrendelő	Határidő: 2016. március 21.
A005	Szkeleton bemutatása	bemutató	alapvető	megrendelő	Határidő: 2016. március 29.
A006	Prototípus koncepció kidolgozása	bemutató	alapvető	megrendelő	Határidő: 2016. április 4.
A007	Részletes tervek beadása	bemutató	alapvető	megrendelő	Határidő: 2016. április 11.
A008	Prototípus bemutatása	bemutató	alapvető	megrendelő	Határidő: 2016. április 25.
A009	Grafikus felület specifikációjának beadása	bemutató	alapvető	megrendelő	Határidő: 2016. május 2.
A010	Grafikus változat bemutatása	bemutató	alapvető	megrendelő	Határidő: 2016. május 17.
A011	A projekt végeredményének kiértékelése, végső beadás	bemutató	alapvető	megrendelő	Határidő: 2016. május 20.

2.3.4 Egyéb nem funkcionális követelmények

Nincs egyéb nem funkcionális követelmény.

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	Új játék
Rövid leírás	Új játék kezdése
Aktorok	Játékos
Forgatókönyv	A pálya betöltése után egy új játék kezdése.

Use-case neve	Kilépés
Rövid leírás	Kilépés a játékból
Aktorok	Játékos
Forgatókönyv	A program bezárása a jobb felső sarokban található kilépés gombbal.

Use-case neve	Mozgás
Rövid leírás	Mozgás a játékosal
Aktorok	Játékos
Forgatókönyv	Egy gomb lenyomására a megfelelő játékos, a lenyomott gombnak megfelelő irányú szomszédos mezőre lép, amely nem fal vagy speciális fal.

Use-case neve	Doboz interakció
Rövid leírás	Dobozokkal való műveletek öse
Aktorok	Játékos
Forgatókönyv	Dobozokkal való műveletek öse

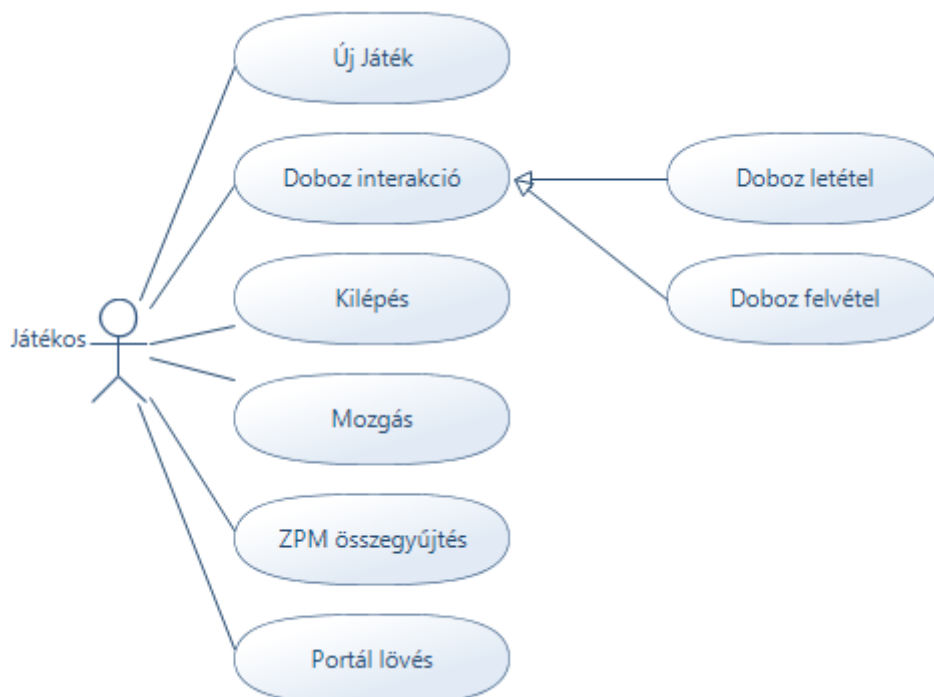
Use-case neve	Doboz letétel
Rövid leírás	Doboz lehelyezése egy mezőre
Aktorok	Játékos
Forgatókönyv	Egy gomb megnyomására egy doboz letétele a megfelelő játékos előtt lévő mezőre, ha az adott játékosnál van doboz.

Use-case neve	Doboz felvétel
Rövid leírás	Doboz felvétele egy mezőről
Aktorok	Játékos
Forgatókönyv	Egy gomb megnyomására a megfelelő játékos előtti mezőn lévő doboz felvétele.

Use-case neve	ZPM összegyűjtés
Rövid leírás	ZPM összegyűjtése egy mezőről
Aktorok	Játékos
Forgatókönyv	A megfelelő játékkal egy mezőn lévő ZPM összegyűjtése.

Use-case neve	Portál Lövés
Rövid leírás	Töltény színe szerinti portál lövése
Aktorok	Játékos
Forgatókönyv	Egy gomb lenyomására a megfelelő játékos egyenes vonalban lő egy portált abba az irányba, amerre éppen néz.

2.4.2 Use-case diagram



2.5 Szótár

Ajtó: Speciális *Mező*, amely viselkedhet *Falként* vagy *Útként*.

Ajtó bezárul: Egy *Ajtó* típusú *Mező* állapotváltozása, melynek hatására nem lesz *Átjárható*. Az éppen ajtóban álló *Tárgyakat* megsemmisíti, a játékosokat megöli.

Ajtó kinyílik: Egy *Ajtó* típusú *Mező* állapotváltozása, melynek hatására *Átjárható* lesz.

Átjárható: Egy *Mezőnek* azon tulajdonsága, amely azt határozza meg, hogy a játékos vagy bizonyos tárgyak tartózkodhatnak-e a *Mezőn*.

Csillagkapu: lásd *Portál*

Doboz: *Mozgatható*, *Tömeggel* rendelkező objektum, amelyet a játékos magához vehet és le is rakhat.

Játékos meghal: A játékos által irányított karakter életét veszti. Ilyen esetben a játéknak vége, az életben maradt játékos nyer.

Facebook: Közösségi oldal.

Fal: Olyan *Mező*, amin a játékos nem tartózkodhat

Féregjárát: Azon élek, amiken a két különböző *Portál*fajta található. Összekapcsolódnak és úgy viselkednek, mintha egy és ugyanazon él lennének - vagyis *Mozgással* átléphetünk az egyik *Portállal* határolt átjárható *Mezőről* a másik *Portállal* határolt átjárható *Mezőre* abban az esetben, ha mindkét színű *Portálból* létezik egy-egy példány.

Freedcamp: Projektvezetési eszköz.

Git: Verziókezelő eszköz.

Google Docs: Párhuzamos dokumentumkezelésre alkalmas eszköz.

Google Drive: Közös elérhető dokumentumok tárolására alkalmas eszköz

Jaffa: Az egyik játékos által irányított karakter neve

Megrendelő: IIT tanszék.

Mérleg: Az *Ajtók* zármechanizmusa. Minden *Ajtóhoz* pontosan egy *Mérleg* tartozik és minden *Mérleghez* pontosan egy *Ajtó*. Ha *Tömeggel* rendelkező objektum van vele azonos *Mezőn*, akkor a hozzá tartozó *Ajtó* nyitott állapotba kerül.

Mező: A pálya atomi alap egysége.

Microsoft Word: Dokumentum szerkesztő eszköz

Mozgás: Egy *Mezőről* egy vele közös élű *Mezőre* való átlépés.

Mozgatható: Olyan objektum, melynek pozícióját a játékos megváltoztathatja.

O'Neill ezredes: A másik játékos által irányított karakter neve.

Pálya: Egy négyzetráccsal *Mezőkre* osztott felület.

Portál: Speciális *fal*, amit eltalált egy *Speciális lövedék*.

Portál lövés: Egy játékbeli cselekvés, aminek során a játékos saját pozíciójában létrehoz egy egyenes vonalban mozgó *Speciális töltényt*, amely abba az irányba mozog, amerre az ezredes éppen néz.

Replikátor: Egy játékbeli objektum, amely a *Pálya* átjárható részein véletlenszerűen mozog. A játékoskal vagy *Tárgyakkal* nem lép interakcióba, de *Speciális lövedékkel* való ütközés hatására megsemmisül és ha *Szakadékba* esik, azt *Úttá* alakítja.

Speciális fal: Olyan *Mező*, amin a játékos nem tartózkodhat. Az *Úttal* határos éleire *Portál* lőhető.

Speciális fegyver: A játékos fegyvere, amellyel kilőhetőek a *Speciális lövedékek*.

Speciális lövedék: A *Speciális fegyver* által lőtt lövedék. Négy különböző fajtája létezik, ami meghatározza, hogy kék, sárga, piros vagy zöld *Portált* hozunk létre.

Speciális töltény: A *Speciális fegyver* tölténye.

Skype: Internetes telefonhívásra alkalmas eszköz.

Súlyt helyez: Egy *Mezőre Tömeggel* rendelkező objektum kerül. Ez lehet egy játékos vagy egy *Doboz*.

Szakadék: Olyan *Mező*, ami elpusztítja a tárgyakat és a játékosokat. A *Replikátor Úttá* alakítja.

Tárgy: Játékbeli objektum, amely egy *Mezőhöz* van kötve és különböző interakcióival a játék menetét befolyásolja. Rendelkezhet *Tömeggel* és lehet *Mozgatható*.

Tömeg: Egy objektum fizikai tulajdonsága.

Út: Olyan *Mező*, amelyen a játékos tartózkodhat és tárgyak is lehetnek rajta.

ZPM: A játékban összegyűjtendő tárgy.

2.6 Projekt terv

2.6.1 A csapat felépítése

A megalakított csapat 5 főből áll, melynek tagjai, illetve a tagok felelőssége a következő táblázatban szerepel. A csapat minden tagja a lehetőségekhez mérten a legtöbb feladat típusból kiveszi a részét.

Név	Beosztás, feladatok
Csóka Máté	Csapatvezetés, kódolás, dokumentáció
Rosner Roland	UML készítés, tesztelés, dokumentáció
Sándor Mátyás Márton	UML tervezés, kódolás, dokumentáció
Szalisznyó Ivett	UML készítés, tesztelés, dokumentáció
Vándor Dávid	Kódolás, tesztelés, dokumentáció

Ahogy a fenti táblázatban is látható a csapat minden tagja köteles saját munkáját dokumentálni. Az elvégzendő feladatokat három fő csoportba osztottuk, melyek az UML készítéssel kapcsolatos feladatok, Java kód elkészítése és tesztelés. A megjelölt tagok felelősek az adott feladatok megfelelő végrehajtásáért, azonban ez nem jelenti azt, hogy ne kaphatnának más típusú részfeladatot.

2.6.2 A projekt időbeosztása heti bontásban

Az alábbi táblázatban láthatók a projekt feladatai, illetve az azokhoz köthető határidők heti bontásban. A mérföldkövek a táblázatban vastagon szedettek.

Határidő	Feladat
2016. 02. 29.	Követelmény, projekt, funkcionalitás
2016. 03. 07.	Analízis modell kidolgozása 1.
2016. 03. 14.	Analízis modell kidolgozása 2.
2016. 03. 21.	Szkeleton tervezése
2016. 03. 29.	Skeleton
2016. 04. 04.	Prototípus koncepciója
2016. 04. 11.	Részletes tervek
2016. 04. 25.	Prototípus
2016. 05. 02.	Grafikus felület specifikációja
2016. 05. 17.	Grafikus változat
2016. 05. 20.	Összefoglalás

2.6.3 A projekt heteinek beosztása

Minden **hétfőn** bemutatjuk az addig elkészítendő munkát a megrendelőnek.

A csapat tagjainak **kedd** éjfélig van lehetőségük egy közösen szerkeszthető dokumentumban feltenni az adott heti feladattal kapcsolatos kérdéseiket, melyeket másnap a megrendelővel tisztázhatunk.

Szerdán a csapat néhány tagja találkozik a megrendelővel a bizonytalan kérdések egyeztetése érdekében. A csapat minden tagja szerda este személyesen találkozik, ahol megvitatják a megvalósítási lehetőségeket, kijelölik az adott heti feladatokat, illetve azok felelőseit.

A csapat tagjainak **csütörtöktől vasárnap** kora délutánig van lehetőségük feladataik elkészítésére. Vasárnap délután a csapat tagjai áttekintik a többiek munkáját, hiba esetén jelzik azt a felelős személynek. Vasárnap este egy rövid megbeszélés keretében értékeljük a heti munkát.

2.6.4 A mérföldkövek, azok jelentősége

A projekt megvalósítására három mérföldkövet jelölt ki a megrendelő, melyek a következők:

- Skeleton
- Prototípus
- Grafikus

A **szkeleton** változat célja annak bizonyítása, hogy az objektum és dinamikus modellek a definiált feladat egy modelljét alkotják. A szkeleton egy program, amelyben már valamennyi,

a végső rendszerben is szereplő business objektum szerepel. Az objektumoknak csak az interfésze definiált. Valamennyi metódus az indulás pillanatában az ernyőre szöveges változatban kiírja a saját nevét, majd meghívja azon metódusokat, amelyeket a szolgáltatás végrehajtása érdekében meg kell hívnia. Amennyiben a metódusból valamely feltétel fennállása esetén hívunk meg más metódusokat, akkor a feltételre vonatkozó kérdést interaktívan az ernyőn fel kell tenni és a kapott válasz alapján kell a továbbiakban eljárni.

A **prototípus** program célja annak demonstrálása, hogy a program elkészült, helyesen működik, valamennyi feladatát teljesíti. A prototípus változat egy elkészült program kivéve a kifejlett grafikus interfészt.

A teljes (**grafikus**) változat a prototípustól elvileg csak a kezelői felület minőségében különbözhet.

2.6.5 Kommunikációs csatornák

Négy csatornát határoztunk meg az esetleges problémák, illetve megbeszélések lebonyolítására:

- **Személyes megbeszélés:** a csapat minden héten szerdán megbeszélésen vesz részt, ahol áttekintjük, kiosztjuk a feladatokat
- **Skype:** a vasárnapi heti összegző megbeszélést Skype konferenciahívással bonyolítjuk le
- **Facebook csoport:** létrehoztunk egy csoportot, ahol a mindekit érintő eseti problémákat vitatjuk meg
- **Telefon:** egy - egy személyt érintő eseti problémák megvitatására

2.6.6 A projekt megvalósítására felhasznált szoftverek, eszközök

Projektvezetési eszközként a **Freedcampet** használjuk, amely online érhető el. Segítségével a létrehozott projektben fel tudjuk venni a szükséges feladatokat, melyekhez személyt tudunk rendelni. A személyek feladatonként könyvelik a ráordított időt. A projekt bármely szakaszában könnyen visszakövethetjük a felelős személyeket, illetve a ráfordított időt.

A verziókezelésre használt eszköz a **Git** lesz. Funkcionalitását tekintve kielégítette az általunk szabott követelményeket, segítségével könnyen végezhető párhuzamosan a munka, visszakövethető, hogy ki mit csinált, illetve probléma esetén könnyen visszaállhatunk egy korábbi verzióra.

A dokumentumok párhuzamos szerkesztésére a **Google Docs** online dokumentumszerkesztőt választottuk, melynek az azonos időben történő munkavégzésen kívül előnye a szerkesztettség automatikus nyomonkövetése, illetve kommentezhetőség. Az esetleges formázási hibákat **Microsoft Worddel** javítjuk. Az elkészült dokumentumokat **Google Driveon** tároljuk.

Szoftverfejlesztésre az **Eclipse** fejlesztőkörnyezetet használjuk, melyhez telepíthető az **UML To Java Generator** beépülő modul, amely alkalmas mind modellezésre, mind az elkészített modellből Java kód generálására.

A grafikus felület elemeinek elkészítését **Gimppel** fogjuk elvégezni.

3 Analízis modell kidolgozása 1

3.1 Objektum katalógus

3.1.1 Box

A dobozt leíró osztály. Van egy kezdő pozíciója, melyre azért van szükség, mert megsemmisülése esetén (**Gap** mezőre kerül) a kezdő pozíciójának megfelelő helyen újra megjelenik.

3.1.2 Door

Az ajtó típusú mezőt leíró osztály. Alapesetben zárva van, azonban minden ajtóhoz tartozik egy mérleg (**Scale**) melyre ha súly kerül, kinyílik az ajtó. Amikor a súlyt eltávolítják a mérlegről, az ajtó bezárul. Egy ajtóhoz pontosan egy mérleg tartozik. Az ajtóba nyitott állapotában tudunk dobozt lehelyezni, az ajtó bezárulásakor a doboz megsemmisül és eredeti helyén megjelenik. Ajtón csak nyitott állapotban tudunk áthaladni.

3.1.3 Game

A játék indulásáért és a pontok számolásáért felelős osztály.

3.1.4 Gap

A szakadék típusú mezőt leíró osztály. Ha erre érkezik egy **Movable** típusú objektum, akkor megsemmisül. **Movable** típusú objektum lehet maga az ezredes (ezzel véget ér a játék) vagy egy doboz (ez megsemmisül és azonnal a kezdeti helyén újból megjelenik).

3.1.5 Map

Maga a labirintus. Négyzetrácsos szerkezetű, 120 négyzetből fog állni. Minden egyes négyzetről tárolni fogjuk, hogy milyen típus, és a pályán hol található.

3.1.6 MapBuilder

A játék indulásakor létrehozza és felépíti a **Map** osztályt.

3.1.7 Player

Az ezredes leíró osztály. A felhasználó őt irányítja a billentyűzet segítségével. Tud speciális speciális lövedéket lőni és dobozokat felvenni, mozogni dobozzal együtt és dobozt letenni. Ha olyan mezőre érkezik, ahol ZPM van, azt felveszi (a ZPM megsemmisül és a Game osztályban növekszik a pontszámláló értéke. Ha szakadékba lép (Gap), a játék véget ér.

3.1.8 Projectile

A lövedék megvalósító osztály. Az ezredes hozza létre. Miután létrejött, elindul abba az irányba, amerre az ezredes néz. Ha **Wall**ba vagy **Door**ba (ami zárva van, tehát a hozzá

tartozó **Scale**-en nincs tömeg) ütközik, akkor megsemmisül. Ha **SpecialWall**ba ütközik és annak az adott élén még nincs **Stargate**, akkor létrehoz egyet, különben megsemmisül.

3.1.9 Road

Az út típusú mezőt leíró osztály. Ezeken a pályaelemeken az ezredes és a lövedék mindig át tud haladni. Lehet rajta **ZPM** és doboz.

3.1.10 Scale

A mérleg típusú mezőt leíró osztály. Minden mérleghez tartozik pontosan egy ajtó, amelyet kinyit ha olyan objektum érkezik rá, aminek van súlya. Csak a doboznak (**Box**) és az ezredesnek (**Player**) van tömege, ami aktiválja a mérleget. Alapesetben (nincs rajta súly) zárva tartja az ajtót.

3.1.11 SpecialWall

A speciális fal típusú mezőt leíró osztály. Alapesetben nem tudunk áthaladni rajta, falként (**Wall**) viselkedik. Annnyiban különbözik a **Wall** típustól, hogy ezen tudunk csillagkaput nyitni, melyen át is haladhatunk. Létezik olyan eshetőség, amikor egy **SpecialWall**on több csillagkapu is található, ezt is kezeli.

3.1.12 Stargate

A csillagkaput megvalósító osztály. Kezeli a játékos és a doboz féregjáraton való áthaladását.

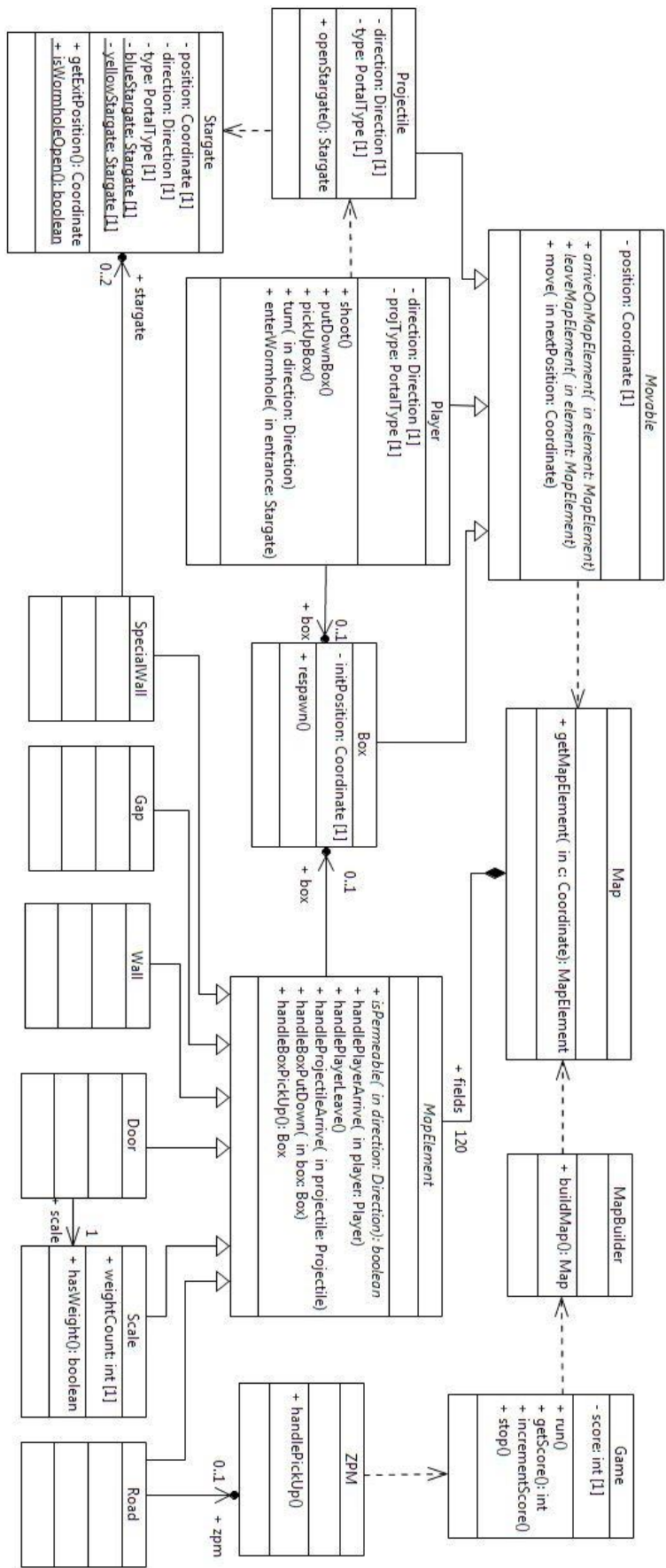
3.1.13 Wall

A fal típusú mezőt leíró osztály. Nem tudunk áthaladni rajta és nem tudunk csillagkaput nyitni rajta.

3.1.14 ZPM

A játékos célja ezeknek az objektumoknak a fölszedése. Csak **Road** típusú mezőn helyezkedhetnek el.

3.2 Statikus struktúra diagramok



3.3 Osztályok leírása

3.3.1 Box

- **Felelősség**
A Dobozok kezelése. Tudnia kell a Doboz pozícióját és tudnia kell lekezelnie, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább.
- **Ősosztályok**
Movable
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **Coordinate initPosition:** Ebben tárolja a doboz kezdeti pozícióját.
- **Metódusok**
 - **void respawn():** A doboz megsemmisülése után hívódik meg.

3.3.2 Door

- **Felelősség**
Az ajtó típusú mező megvalósításáért felel. Mindegyik **Door**hoz tartozik pontosan egy **Scale**, mely képes az ajtó mezőt átjárhatóvá tenni.
- **Ősosztályok**
MapElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **Scale scale:** A Mezőhöz tartozó mérlegre mutató referencia
- **Metódusok**
Az ősosztálytól örökölt metódusokon kívül nem deklarál újat.

3.3.3 Game

- **Felelősség**
Az osztály felelőssége a játék elindítása és az összegyűjtött ZPM-ek számontartása.
- **Ősosztályok**
Nincs ősosztálya.

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **int score**: Az összegyűjtött ZPM-ek száma.

- **Metódusok**

- **void run()**: Ez a főciklus, melyben maga a játék működik.
- **int getScore()**: Visszaadja, hogy mennyi pontot gyűjtött össze a játékos.
- **void incrementScore()**: Megnöveli az összegyűjtött pontok értékét eggyel.
- **void stop()**: A játék vége.

3.3.4 Gap

- **Felelősség**

A szakadék mező megvalósításáért felelős. Ha a játékos vagy egy doboz érkezik rá, akkor azt megsemmisíti.

- **Ősosztályok**

MapElement

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

Az őszosztálytól örökölt attribútumokon kívül mást nem definiál.

- **Metódusok**

Az őszosztálytól örökölt metódusokon kívül nem deklarál újat.

3.3.5 Map

- **Felelősség**

Felelőssége, hogy tárolja a **MapElement** osztályból álló tömböt és visszaadja, hogy melyik koordinátán milyen típusú mező van.

- **Ősosztályok**

Nincs

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **MapElement fields[10][12]**: A pályát felépítő mezők tömbje.

- **Metódusok**

- **MapElement getMapElement(Coordinate coordinate)**: Visszaadja a paraméterül kapott koordinátán lévő mezőt.

3.3.6 MapBuilder

- **Felelősség**
A **Map** osztály létrehozása és pályaelemekkel való feltöltése.
- **Ősosztályok**
Nincs őszosztálya
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Az őszosztálytól örökölt attribútumokon kívül mást nem definiál.
- **Metódusok**
 - **Map buildMap()**: Létrehozza és felépíti a térképet.

3.3.7 MapElement

- **Felelősség**
A Mező típusokat egybefonó őszosztály. Szerepe, hogy a konkrét mezőket a dinamikus típusuk ismerete nélkül tudjuk heterogén kollekcióként kezelni.
- **Ősosztályok**
Nincs
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Box box: A Mezőn elhelyezkedő dobozra mutató referencia.
- **Metódusok**
 - **boolean isPermeable()**: Visszaadja, hogy átjárható-e a Mező.
 - **void handlePlayerArrive(Player player)**: Ez a függvény kezeli le az ezredes érkezését az adott mezőre.
 - **void handlePlayerLeave()**: Ez a függvény kezeli le az ezredes távozását az adott mezőről.
 - **void handleProjectileArrive(Projectile projectile)**: Ez a függvény kezeli le a speciális lövedék érkezését az adott mezőre
 - **void handleBoxPutDown(Box box)**: Ez a függvény kezeli le egy doboz lerakását az adott mezőre.
 - **Box handleBoxPickUp()**: Ez a függvény kezeli le egy doboz felvételét az adott mezőről.

3.3.8 Movable

- **Felelősség**
A mozgatható objektumokat egybefonó absztrakt őszosztály. Szerepe, hogy a mozgatható objektumokat egységesen tudjuk kezelni.
- **Őszosztályok**
Nincs
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **Coordinate position:** Ebben tárolja az objektum a pozícióját.
- **Metódusok**
 - **void arriveOnMapElement(MapElement mapElement):** Ez a függvény kezeli le egy objektum mezőre való érkezését.
 - **void leaveMapElement(MapElement mapElement):** Ez a függvény kezeli le egy objektum mezőről való távozását.
 - **void move(Coordinate nextPosition):** Ez a függvény kezeli le egy objektum mozgását egy adott koordinátára.

3.3.9 Player

- **Felelősség**
Az ezredes kezelése. Tudnia kell az ezredes pozícióját, tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább.
- **Őszosztályok**
Movable
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **Direction direction:** Az irány, amerre az ezredes néz.
 - **PortalType projType:** A kilőtt speciális lövedék színe.
- **Metódusok**
 - **void shoot():** Ez a függvény kezeli a lövést.
 - **void pickUpBox():** Ez a függvény kezeli a doboz felvételt.
 - **void putDownBox():** Ez a függvény kezeli a doboz letevését.
 - **void turn(Direction direction):** Ez a függvény kezeli a játékos irányváltását.
 - **void enterWormhole(Stargate entrance):** Ez a függvény kezeli a játékos csillagkapuba lépését.

3.3.10 Projectile

- **Felelősség**
A Speciális lövedékek kezelése. Tudnia kell a Speciális lövedék pozícióját, tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább. Speciális fallal történő ütközés esetén létrehoz egy **Stargate** objektumot.
- **Ősosztályok**
Moveable
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **Direction direction:** Ebben tárolja a lövedék irányát.
 - **PortalType type:** Ebben tárolja a lövedék típusát.
- **Metódusok**
 - **Stargate openStargate():** Ez a függvény kezeli a lövedék speciális fallal való találkozásokor keletkező csillagkapu nyitását.

3.3.11 Road

- **Felelősség**
Az út típusú mezők megvalósításáért felelős.
- **Ősosztályok**
MapElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **ZPM zpm:** A Mezőn elhelyezkedő ZPM-re mutató referencia.
- **Metódusok**
Az ősosztálytól örökölt metódusokon kívül nem deklarálnak újat.

3.3.12 Scale

- **Felelősség**
A mérleg típusú mezők megvalósításáért felel. Mindenképpen egy ajtóhoz tartozik. Amennyiben rákerül a játékos vagy egy doboz, akkor az ajtót átjárhatóvá teszi.
- **Ősosztályok**
MapElement

- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **int weightCount**: Azt tartjuk számon benne, hogy hány olyan objektum van a mezőn, aminek van súlya.
- **Metódusok**
 - **bool hasWeight()**: Visszaadja, hogy van-e súly a mérlegen.

3.3.13 SpecialWall

- **Felelősség**
Felelőssége olyan fal megvalósítása, mely alapesetben átjárhatatlan, de ha **Projectile** éri, akkor egy **Stargate** nyílik rajta, így szomszédos mezőnek tekinthető azzal a **MapElement**tel, mely szintén egy **SpecialWall** és van rajta egy másik **Stargate**.
- **Ősosztályok**
MapElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **List<Stargate> stargate**: A Mezőn elhelyezkedő csillagkapura egy referencia.
- **Metódusok**
Az ősosztálytól örökölt metódusokon kívül nem deklarál újat.

3.3.14 Stargate

- **Felelősség**
A **Projectile** ha egy **SpecialWall**ba ütközik, akkor jön létre a **Stargate**. Két **Stargate** közötti kapcsolatért felelős ez az osztály.
- **Ősosztályok**
Nincs ősosztálya
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **Coordinate position**: A csillagkapu koordinátájának tárolására van.
 - **Direction direction**: A csillagkapu irányát tároljuk benne.
 - **PortalType type**: A csillagkapu típusát tároljuk benne.
 - **Stargate blueStargate**: A kék csillagkapu

- **Stargate yellowStargate**: A sárga csillagkapu
- **Metódusok**
 - **Coordinate getExitPosition()**: A függvény visszaadja a féregjárat kijáratának koordinátáit.
 - **boolean isWormholeOpen()**: Visszatérési értékben megmondja, hogy van-e két csillagkapu a pályán.

3.3.15 Wall

- **Felelősség**

A fal típusú mezők megvalósításáért felelős. Nem átjárható. Ezekre a mezőkre nem kerülhet semmilyen **Movable** objektum.
- **Ősosztályok**

MapElement
- **Interfészek**

Nem valósít meg interfészt.
- **Attribútumok**

Az őszosztálytól örökölt attribútumokon kívül mást nem definiál.
- **Metódusok**

Az őszosztálytól örökölt metódusokon kívül nem deklarál újat.

3.3.16 ZPM

- **Felelősség**

A játék célja, hogy az összes ilyen objektumot felszedjük.
- **Ősosztályok**

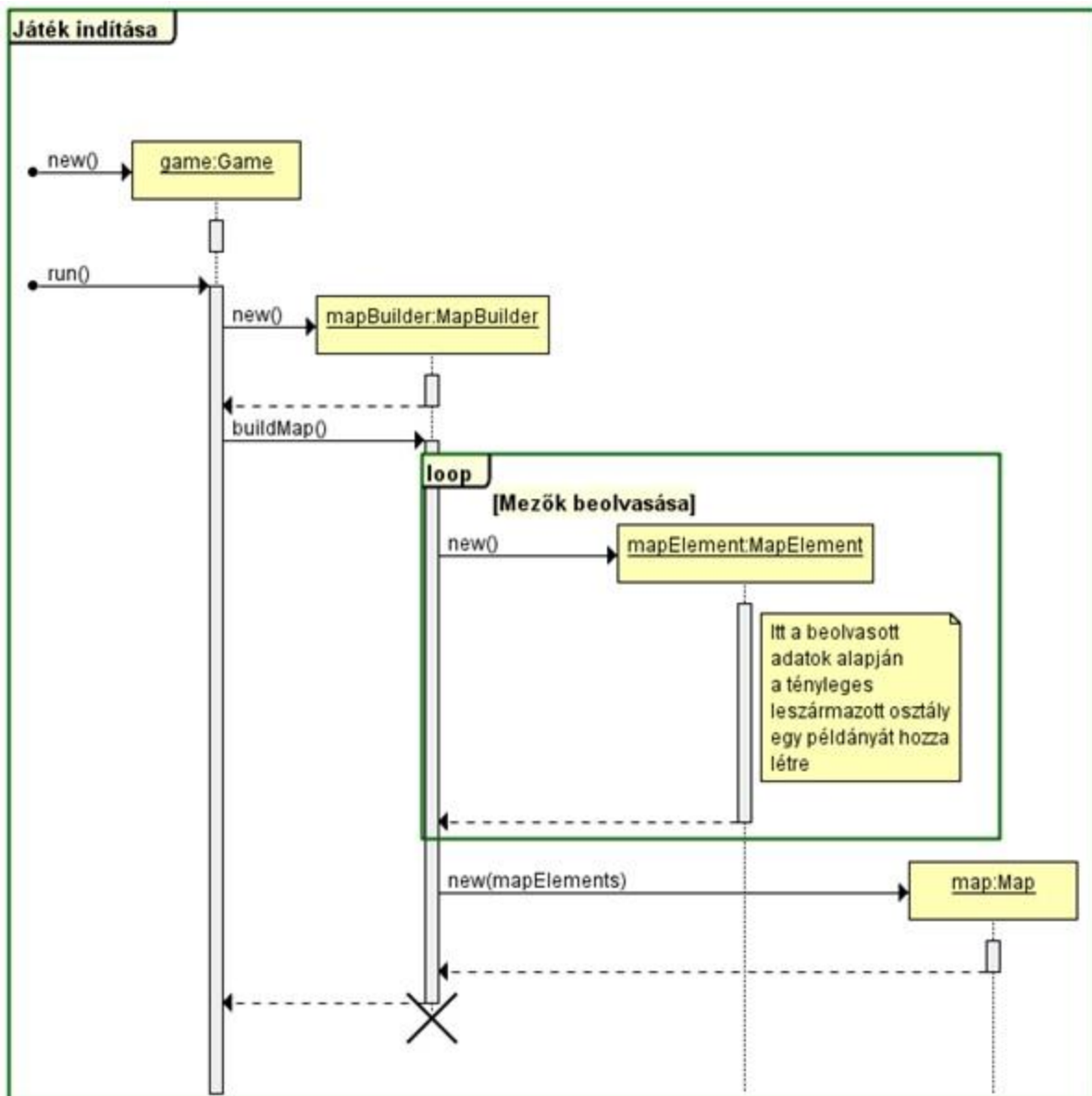
Nincs őszosztálya.
- **Interfészek**

Nem valósít meg interfészt.
- **Attribútumok**

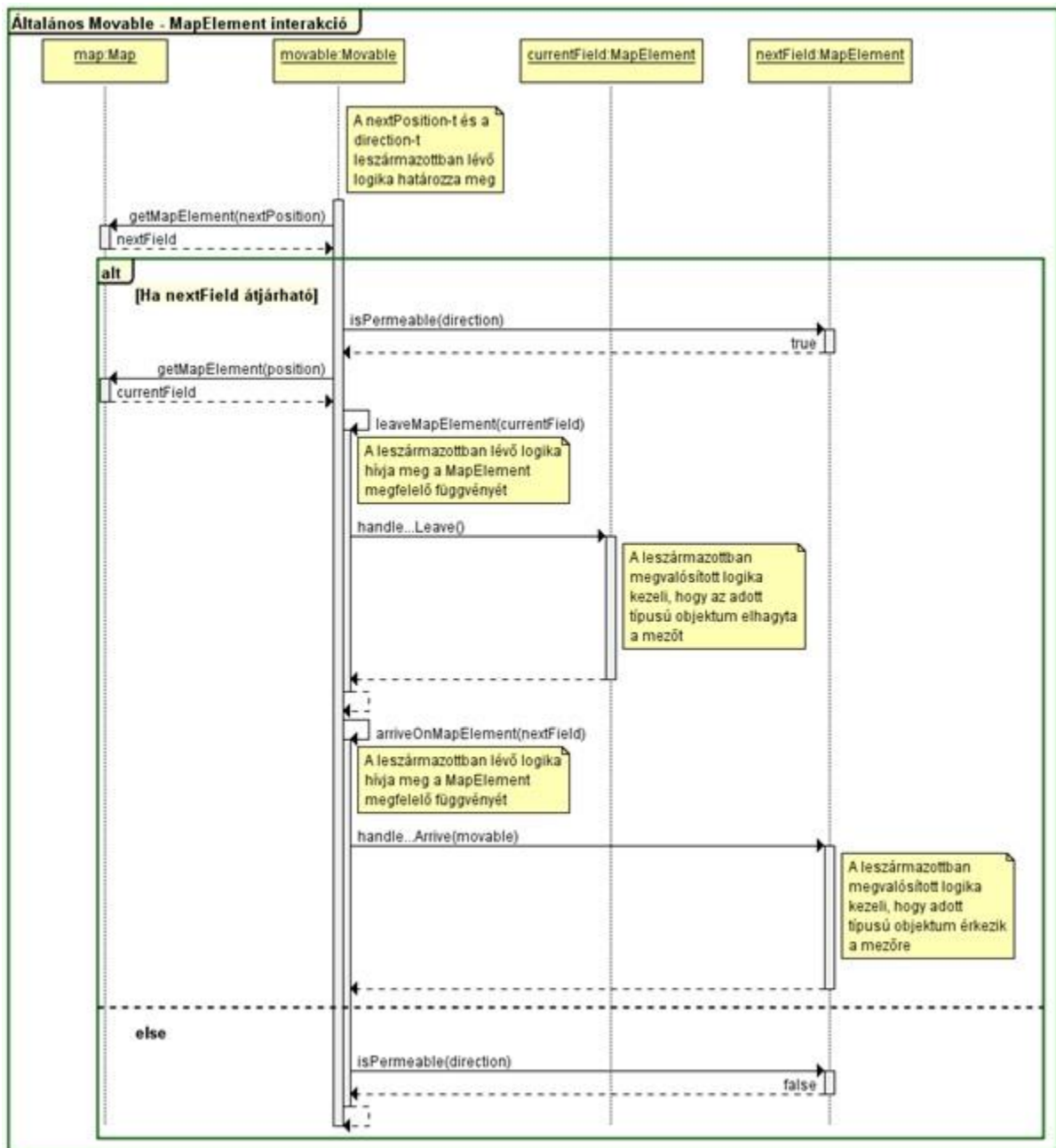
Nincsenek attribútumai
- **Metódusok**
 - **void handlePickUp()**: Ez a függvény kezeli le, ha a játékos rálép egy olyan mezőre, ahol ZPM van.

3.4 Szekvencia diagramok

3.4.1 Játék indítása



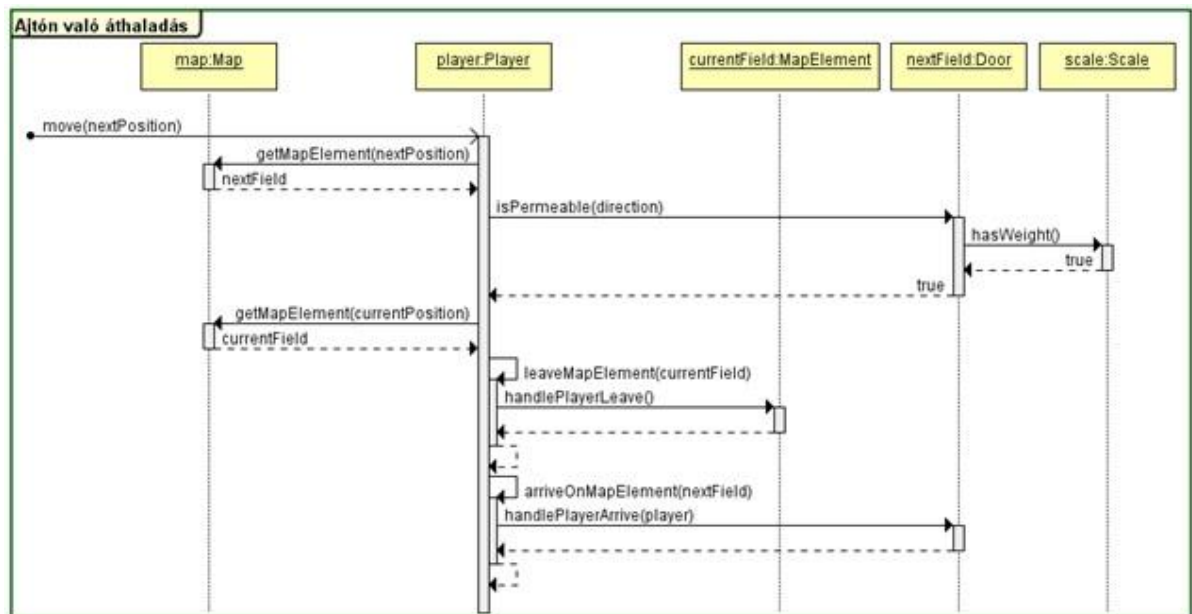
3.4.2 Általános Movable - MapElement interakció



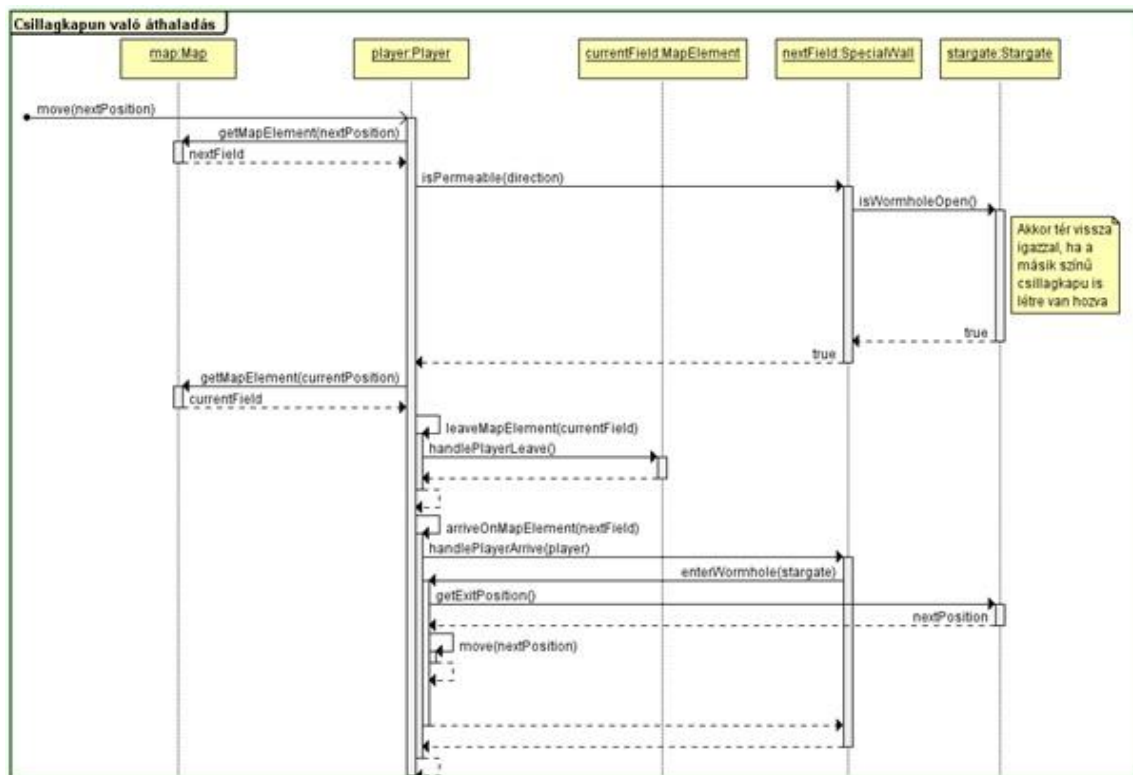
3.4.3 ZPM felvétele



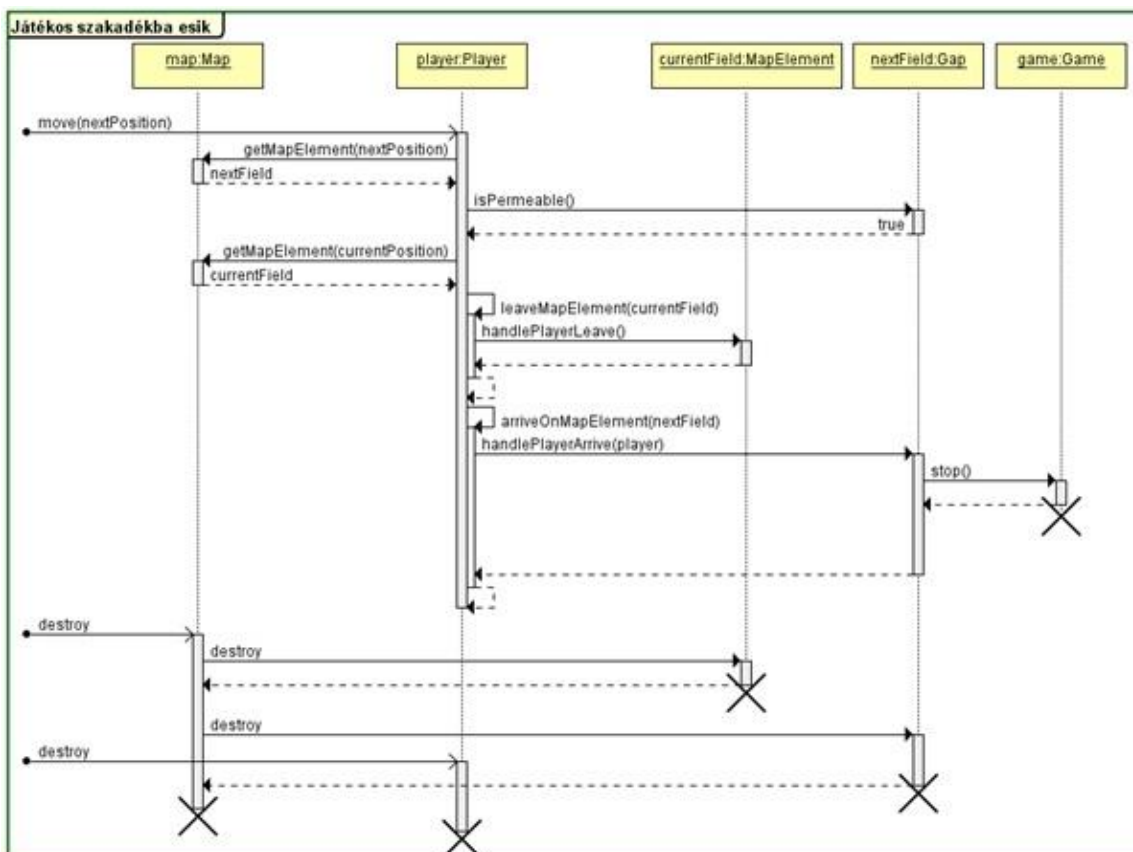
3.4.4 Ajtón való áthaladás



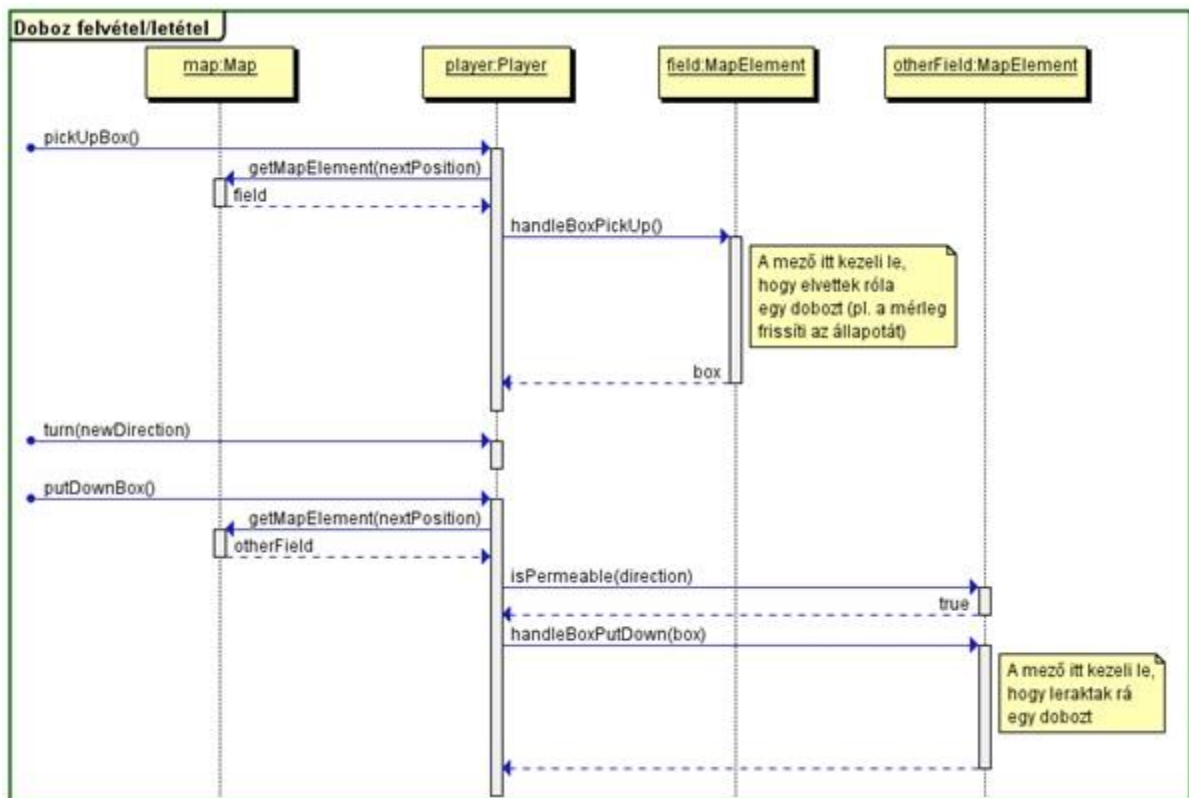
3.4.5 Csillagkapun való áthaladás



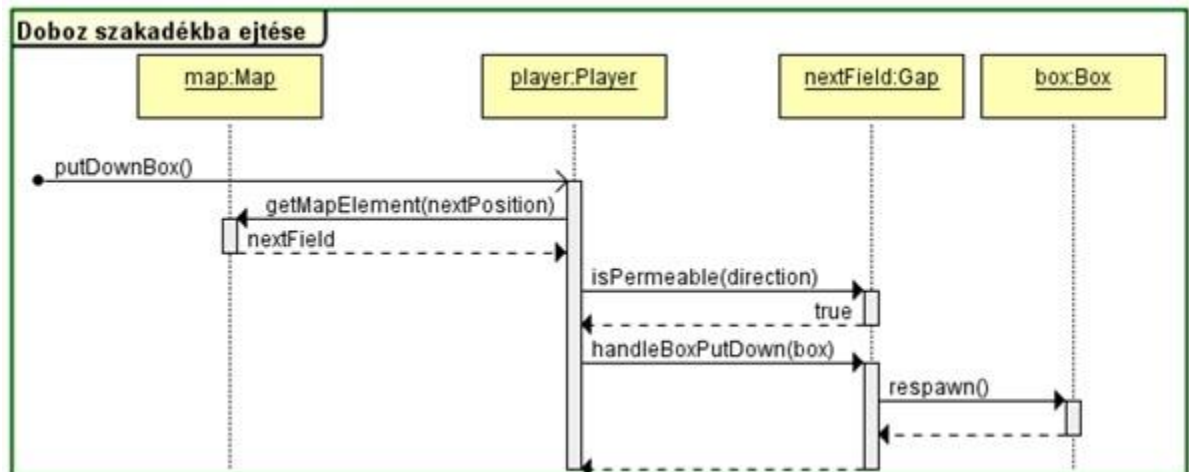
3.4.6 Játékos szakadékba esik



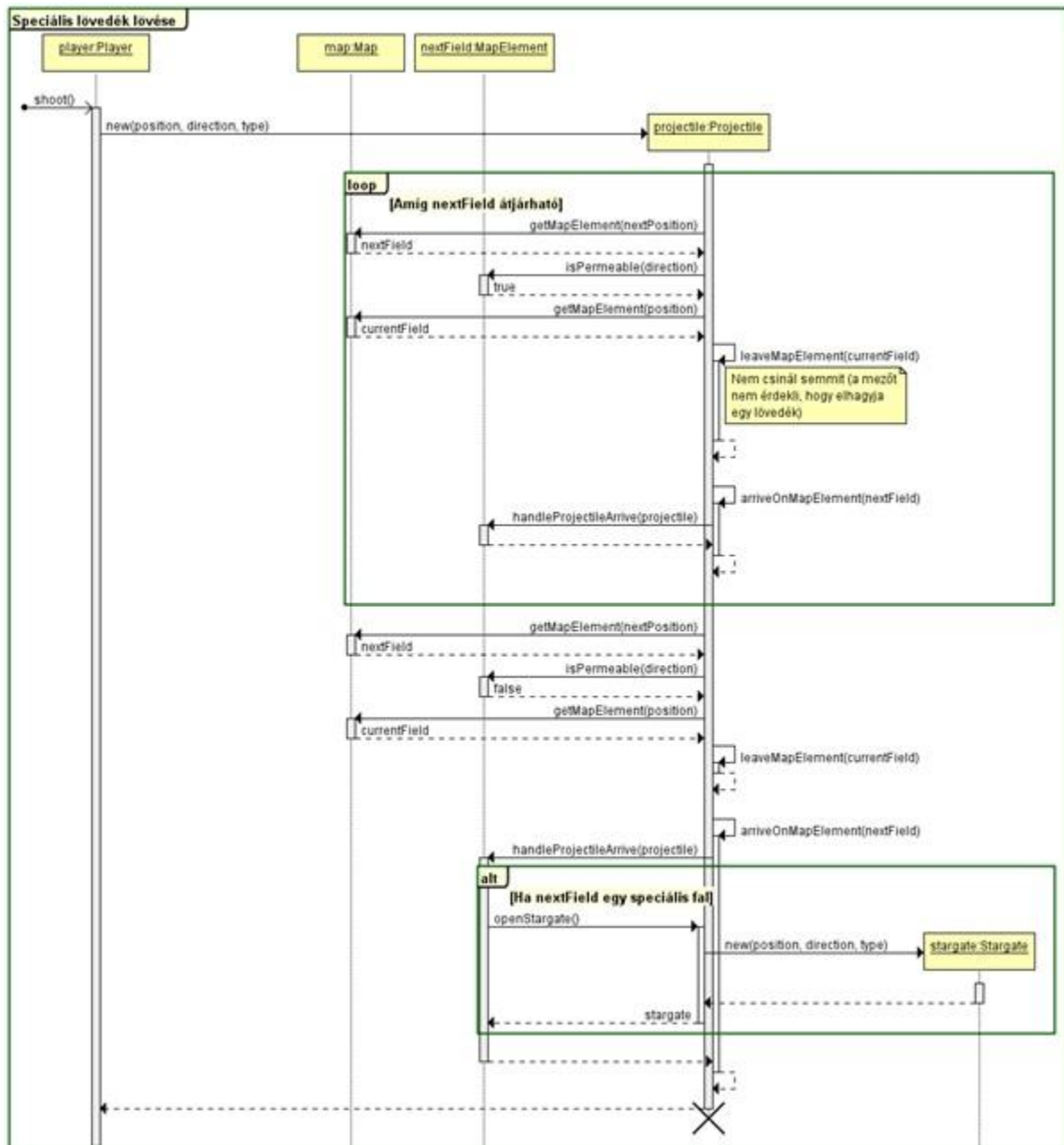
3.4.7 Doboz interakciók



3.4.8 Doboz szakadékba ejtése

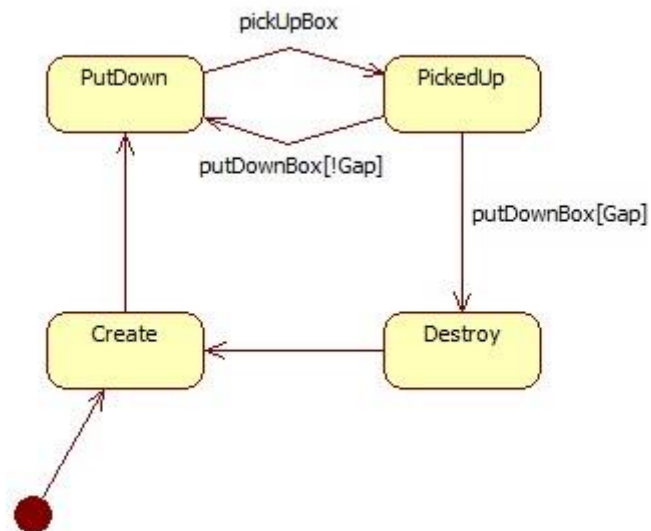


3.4.9 Speciális lövedék lövése

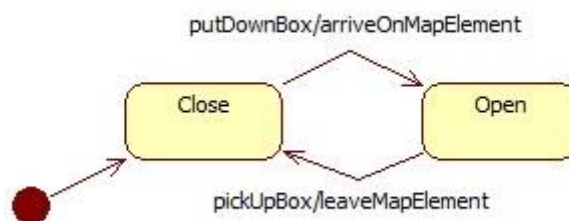


3.5 State-chartok

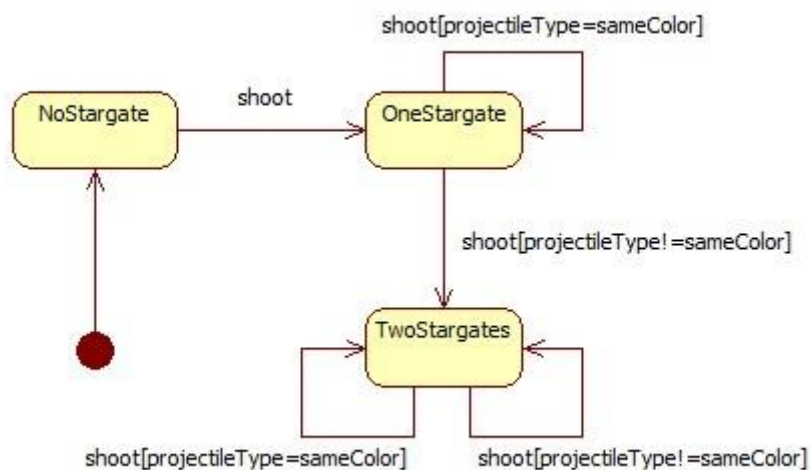
3.5.1 Box



3.5.2 Door



3.5.3 Stargate



4 Analízis modell kidolgozása 2

4.1 Objektum katalógus

4.1.1 Box

Van egy kezdő pozíciója, melyre azért van szükség, mert megsemmisülése esetén (szakadék (**Gap**) mezőre kerül vagy ajtó rácsukódik) a kezdő pozíciójának megfelelő helyen újra megjelenik.

4.1.2 Door

Alapesetben zárva van, azonban minden ajtóhoz tartozik pontosan egy mérleg (**Scale**) melyre ha megfelelő nagyságú súly kerül, kinyílik az ajtó. Amikor a súlyt eltávolítják a mérlegről, az ajtó bezárul. Az ajtóba nyitott állapotában tudunk dobozt (**Box**) lehelyezni, az ajtó bezárulásakor a doboz (**Box**) megsemmisül és eredeti helyén megjelenik. Ajtón csak nyitott állapotban tudunk áthaladni.

4.1.3 Gap

Ha erre érkezik egy játékos (**Player**) vagy egy doboz (**Box**) akkor megsemmisül, játékos (**Player**) esetén a játéknak is vége lesz. A lövedék (**Projectile**) áthalad rajta. Ha Replikátor érkezik rá, úttá (**Road**) alakul át.

4.1.4 Jaffa

A második felhasználó őt irányítja. Tud speciális lövedéket (**Projectile**) lőni és dobozt (**Box**) felvenni, mozogni dobozzal együtt és dobozt letenni. Ha olyan mezőre érkezik, ahol **ZPM** van, azt felveszi (a **ZPM** megsemmisül és növekszik a pontszámlálójának értéke. Ha szakadékba lép (**Gap**) vagy rácsukódik egy ajtó, a játék véget ér és a másik felhasználó nyer.

4.1.5 O'Neill

Az első felhasználó őt irányítja. Tud speciális lövedéket (**Projectile**) lőni és dobozt (**Box**) felvenni, mozogni dobozzal együtt és dobozt letenni. Ha olyan mezőre érkezik, ahol **ZPM** van, azt felveszi (a **ZPM** megsemmisül és növekszik a pontszámlálójának értéke. Ha szakadékba lép (**Gap**) vagy rácsukódik egy ajtó, a játék véget ér és a másik felhasználó nyer.

4.1.6 Projectile

Egy játékos hozza létre. Miután létrejött, elindul abba az irányba, amerre a játékos néz. Ha falba (**Wall**) vagy ajtóba (**Door**) (ami zárva van, tehát a hozzá tartozó mérlegen (**Scale**) nincs tömeg) ütközik, akkor megsemmisül. Ha speciális falba (**SpecialWall**) ütközik és annak az adott élén még nincs csillagkapu (**Stargate**), akkor létrehoz egyet, különben megsemmisül. A lövedék színét a játékos (**Player**) állítja kékre, sárgára pirosra vagy zöldre.

4.1.7 Replikátor

Egy játékbeli objektum, amely a pálya átjárható részein véletlenszerűen mozog. A játékoskal vagy tárgyakkal nem lép interakcióba, de Speciális lövedékekkel való ütközés hatására megsemmisül és ha Szakadékba **(Gap)** esik, azt Úttá **(Road)** alakítja.

4.1.8 Road

Ezek a pályaelemek játékos és a lövedék mindig át tud haladni. Lehet rajta **ZPM** és doboz **(Box)**.

4.1.9 Scale

Minden mérleghez tartozik pontosan egy ajtó, amelyet kinyit ha olyan objektum érkezik rá, aminek van súlya és a mérlegen lévő tárgyak összsúlya eléri az előre megadott súlyhatárt. Csak a doboznak **(Box)** és a játékosoknak **(Jaffa, ONeill)**, illetve a replikátornak van súlya, ami aktiválja a mérleget. Alapesetben (nincs rajta súly) zárva tartja az ajtót.

4.1.10 SpecialWall

Alapesetben nem tudunk áthaladni rajta, falként **(Wall)** viselkedik. Annyiban különbözik a **Wall** típustól, hogy ezen tudunk csillagkaput **(Stargate)** nyitni, melyen át is haladhatunk. Létezik olyan eshetőség, amikor egy speciális falon **(SpecialWall)** több csillagkapu **(Stargate)** is található, ezt is kezeli.

4.1.11 Stargate

Kezeli a játékosok **(Player)** és a doboz **(Box)** féregjáraton való áthaladását. Egyszerre egy kék, egy sárga, egy piros és egy zöld színű csillagkapu **(Stargate)** létezhet, a létrehozott kapu színe a lövedék színével megegyező lesz.

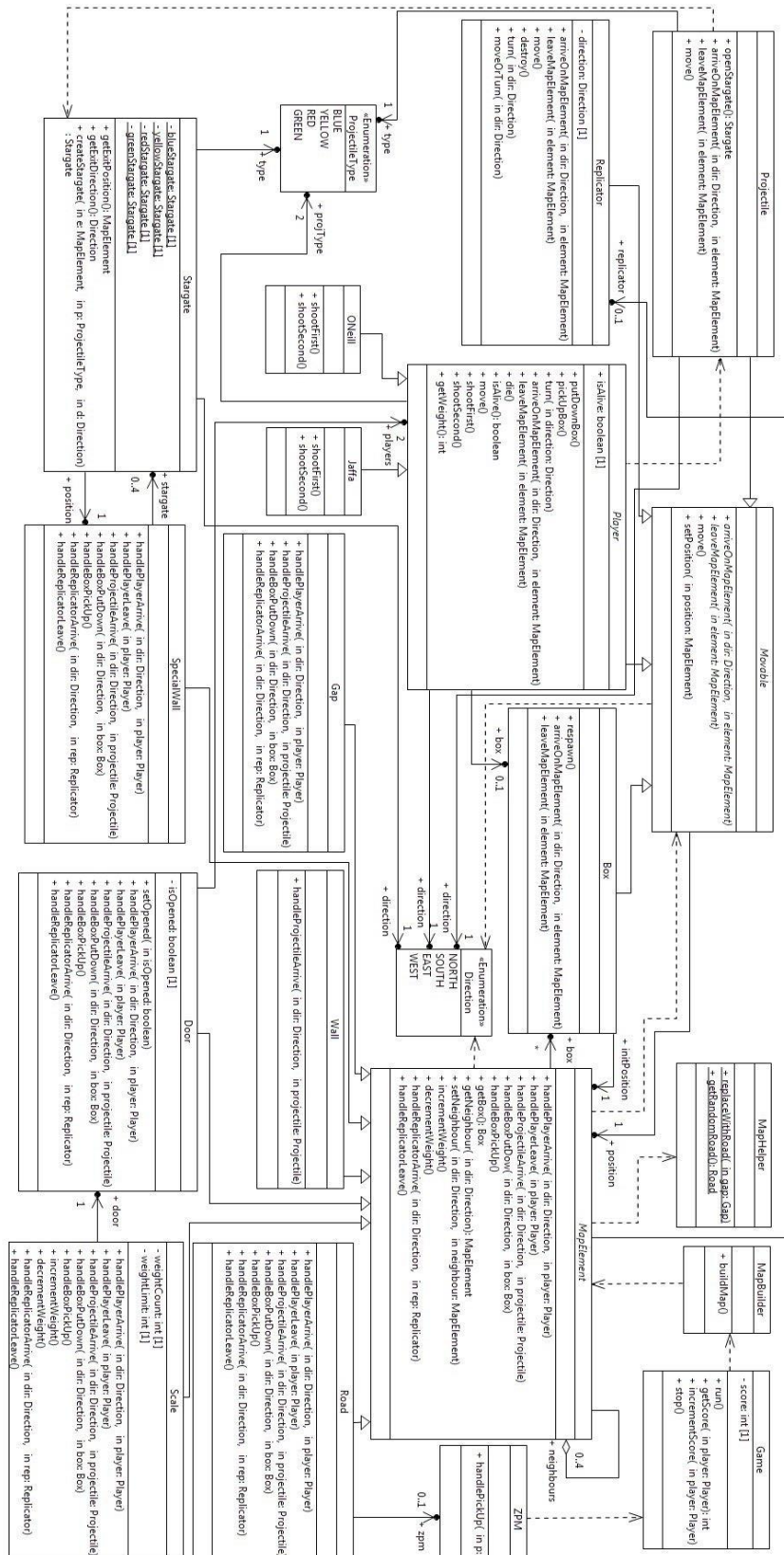
4.1.12 Wall

Nem tudunk áthaladni rajta és nem lehet csillagkaput **(Stargate)** nyitni rajta.

4.1.13 ZPM

A játékosok célja ezeknek a fölszedése. Csak út **(Road)** típusú mezőn helyezkedhet el.

4.2 Statikus struktúra diagramok



4.3 Osztályok leírása

4.3.1 Box

- **Felelősség**
A Dobozok kezelése. Tudnia kell a Doboz pozícióját és tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányba haladna tovább.
- **Ősosztályok**
Movable
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **MapElement initPosition**: Ebben tárolja a doboz kezdeti pozícióját.
- **Metódusok**
 - **void respawn()**: A doboz megsemmisülése után hívódik meg.
 - **void arriveOnMapElement(MapElement mapElement)**: Ez a függvény kezeli le egy doboz mezőre való letételét.
 - **void leaveMapElement(MapElement mapElement)**: Ez a függvény kezeli le egy doboz mezőről való felvételét.

4.3.2 Door

- **Felelősség**
Az ajtó típusú mező megvalósításáért felel. Mindegyik **Door**hoz tartozik pontosan egy **Scale**, mely képes az ajtó mezőt átjárhatóvá tenni.
- **Ősosztályok**
MapElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **boolean isOpened**: Az ajtó útként (true) vagy falként (false) viselkedik
 - **List<Player> players**: Az ajtó mezőn álló játékosok listája.
- **Metódusok**
 - **void handlePlayerArrive(Direction dir, Player player)**: Ez a függvény kezeli le az ezredes érkezését az ajtó mezőre.
 - **void handlePlayerLeave()**: Ez a függvény kezeli le az ezredes távozását az ajtó mezőről.
 - **void handleProjectileArrive(Direction dir, Projectile projectile)**: Ez a függvény kezeli le a speciális lövedék érkezését az ajtó mezőre

- **void handleBoxPutDown(Box box):** Ez a függvény kezeli le egy doboz lerakását az ajtó mezőre.
- **void handleBoxPickUp():** Ez a függvény kezeli le egy doboz felvételét az ajtó mezőről.
- **void setOpen(ed(boolean opened):** Ez a függvény állítja be, hogy az ajtó nyitva vagy csukva van.
- **void handleReplicatorArrive(Direction dir, Replicator rep):** Ez a függvény kezeli le a replikátor érkezését a mezőre.
- **void handleReplicatorLeave():** Ez a függvény kezeli le a replikátor távozását a mezőről.

4.3.3 Game

- **Felelősség**
Az osztály felelőssége a játék elindítása, leállítása és az összegyűjtött ZPM-ek számontartása.
- **Ősosztályok**
Nincs ősosztálya.
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **int score:** Az összegyűjtött ZPM-ek száma.
- **Metódusok**
 - **void run():** Ez a főciklus, melyben maga a játék működik.
 - **int getScore(Player player):** Visszaadja, hogy mennyi pontot gyűjtött össze a játékos.
 - **void incrementScore(Player player):** Megnöveli a játékos által összegyűjtött pontok értékét eggyel.
 - **void stop():** A játék vége.

4.3.4 Gap

- **Felelősség**
A szakadék mező megvalósításáért felelős. Ha a játékos vagy egy doboz érkezik rá, akkor azt megsemmisíti.
- **Ősosztályok**
MapElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Az ősosztálytól örökölt attribútumokon kívül mást nem definiál.

- **Metódusok**

- **void handlePlayerArrive(Direction dir, Player player):** Ez a függvény kezeli le a **játékosok érkezését** a szakadék mezőre.
- **void handleProjectileArrive(Direction dir, Projectile projectile):** Ez a függvény kezeli le a speciális lövedék érkezését a szakadék mezőre
- **void handleBoxPutDown(Direction dir, Box box):** Ez a függvény kezeli le egy doboz lerakását a szakadék mezőre.
- **void handleReplicatorArrive(Direction dir, Replicator rep):** Ez a függvény kezeli le a replikátor érkezését a mezőre.

4.3.5 Jaffa

- **Felelősség**

Jaffa kezelése. Tudnia kell Jaffa pozícióját, tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább.

- **Ősosztályok**

Player

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **Direction direction:** Az irány, amerre a Jaffa néz.
- **Set<ProjectileType> projType:** A kilőhető speciális lövedékek színe.
- **Box box:** Jaffánál lévő doboz referenciája (ha van nála).
- **boolean isAlive:** Igaz, ha még él a Jaffa.

- **Metódusok**

- **void pickUpBox():** Ez a függvény kezeli a doboz felvételt.
- **void putDownBox():** Ez a függvény kezeli a doboz letevését.
- **void turn(Direction direction):** Ez a függvény kezeli Jaffa irányváltását.
- **void arriveOnMapElement(MapElement mapElement):** Ez a függvény kezeli le Jaffa mezőre való érkezését.
- **void leaveMapElement(MapElement mapElement):** Ez a függvény kezeli le Jaffa mezőről való távozását.
- **void die():** Az isAlive attribútumot hamisra állítja.
- **boolean isAlive():** Getter az isAlive attribútumra.
- **void move():** Jaffa mozgatása.
- **void shootFirst():** Lövés az egyik típusú lövedékkal.
- **void shootSecond():** Lövés a másik típusú lövedékkal.
- **int getWeight():** Visszatér Jaffa súlyával.

4.3.6 MapBuilder

- **Felelősség**
A pálya beolvasása, a megfelelő **MapElement** példányok létrehozása és a szomszédos pályaelemek összekapcsolása.
- **Ősosztályok**
Nincs őszosztálya
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Az őszosztálytól örökölt attribútumokon kívül mást nem definiál.
- **Metódusok**
 - **void buildMap()**: Létrehozza és felépíti a térképet.

4.3.7 MapElement

- **Felelősség**
A Mező típusokat egybefonó őszosztály. Szerepe, hogy a konkrét mezőket a dinamikus típusuk ismerete nélkül tudjuk heterogén kollekcióként kezelni.
- **Ősosztályok**
Nincs őszosztálya.
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **Stack<Box> box**: A mezőn elhelyezkedő dobozok tárolását végzi.
 - **HashMap<Direction, MapElement> neighbours**: A Mező szomszédait tárolja.
 - **Replicator replicator**: Mutató a mezőn elhelyezkedő replikátorra.
- **Metódusok**
 - **MapElement getNeighbour(Direction dir)**: Visszaadja a mező megfelelő oldali szomszédját.
 - **Box getBox()**: Visszaadja a mezőn található dobozt.
 - **void setNeighbour(Direction dir, MapElement mapElement)**: Beállítja a mező megfelelő oldali szomszédját.
 - **void handlePlayerArrive(Direction dir, Player player)**: Ez a függvény kezeli le a **játékosok** érkezését az adott mezőre.
 - **void handlePlayerLeave()**: Ez a függvény kezeli le az **játékosok** távozását az adott mezőről.
 - **void handleProjectileArrive(Direction dir, Projectile projectile)**: Ez a függvény kezeli le a speciális lövedék érkezését az adott mezőre
 - **void handleBoxPutDown(Direction dir, Box box)**: Ez a függvény kezeli le egy doboz lerakását az adott mezőre.

- **void handleBoxPickUp():** Ez a függvény kezeli le egy doboz felvételét az adott mezőről.
- **void incrementWeight():** A mezőn elhelyezkedő súlyt növeli.
- **void decrementWeight():** A mezőn elhelyezkedő súlyt csökkenti.
- **void handleReplicatorArrive(Direction dir, Replicator rep):** Ez a függvény kezeli le a replikátor érkezését a mezőre.
- **void handleReplicatorLeave():** Ez a függvény kezeli le a replikátor távozását a mezőről.

4.3.8 MapHelper

- **Felelősség**
A pálya átépítésében nyújt segítséget.
- **Ősosztályok**
Nincs őszosztálya.
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Nincs attribútuma.
- **Metódusok**
 - **void replaceWithRoad(Gap gap):** A megadott szakadék helyére utat épít.
 - **Road getRandomRoad():** Visszatér egy véletlenszerűen választott úttal a pályáról.

4.3.9 Movable

- **Felelősség**
A mozgatható objektumokat egybefonó absztrakt őszosztály. Szerepe, hogy a mozgatható objektumokat egységesen tudjuk kezelni.
- **Ősosztályok**
Nincs őszosztálya.
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **MapElement position:** Ebben tárolja az objektum a pozícióját.
- **Metódusok**
 - **void arriveOnMapElement(MapElement mapElement):** Ez a függvény kezeli le egy objektum mezőre való érkezését.
 - **void leaveMapElement(MapElement mapElement):** Ez a függvény kezeli le egy objektum mezőről való távozását.

- **void move():** Ez a függvény kezeli le egy objektum mozgását egy adott koordinátára.
- **void setPosition(MapElement position):** Az objektum pozícióját állítja be.

4.3.10 ONeill

- **Felelősség**
ONeill kezelése. Tudnia kell ONeill pozícióját, tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább.
- **Ősosztályok**
Player
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **Direction direction:** Az irány, amerre ONeill néz.
 - **Set<PortalType> projType:** A kilőhető speciális lövedékek színe.
 - **Box box:** ONeillnél lévő doboz referenciája (ha van nála).
 - **boolean isAlive:** Igaz, ha Oneill még él.
- **Metódusok**
 - **void pickUpBox():** Ez a függvény kezeli a doboz felvételt.
 - **void putDownBox():** Ez a függvény kezeli a doboz letevését.
 - **void turn(Direction direction):** Ez a függvény kezeli ONeill irányváltását.
 - **void arriveOnMapElement(MapElement mapElement):** Ez a függvény kezeli le ONeill mezőre való érkezését.
 - **void leaveMapElement(MapElement mapElement):** Ez a függvény kezeli le ONeill mezőről való távozását.
 - **void die():** Az isAlive attribútumot hamisra állítja.
 - **boolean isAlive():** Getter az isAlive attribútumra.
 - **void move():** ONeill mozgatása.
 - **void shootFirst():** Lövés az egyik típusú lövedékkel.
 - **void shootSecond():** Lövés a másik típusú lövedékkel.
 - **int getWeight():** Visszatér ONeill súlyával.

4.3.11 Player

- **Felelősség**
A játékos kezelése. Tudnia kell a játékos pozícióját, tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább.
- **Ősosztályok**
Movable

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **Direction direction**: Az irány, amerre a játékos néz.
- **Set<PortalType> projType**: A kilőhető speciális lövedékek színe.
- **Box box**: A játékosnál lévő doboz referenciája (ha van nála).
- **boolean isAlive**: Igaz, ha még él a játékos.

- **Metódusok**

- **void pickUpBox()**: Ez a függvény kezeli a doboz felvételt.
- **void putDownBox()**: Ez a függvény kezeli a doboz letevését.
- **void turn(Direction direction)**: Ez a függvény kezeli a játékos irányváltását.
- **void arriveOnMapElement(MapElement mapElement)**: Ez a függvény kezeli le a játékos mezőre való érkezését.
- **void leaveMapElement(MapElement mapElement)**: Ez a függvény kezeli le a játékos mezőről való távozását.
- **void die()**: Az isAlive attribútumot hamisra állítja.
- **boolean isAlive()**: Getter az isAlive attribútumra.
- **void move()**: Játékos mozgatása.
- **void shootFirst()**: Lövés az egyik típusú lövedékkel.
- **void shootSecond()**: Lövés a másik típusú lövedékkel.
- **int getWeight()**: Visszatér a játékos súlyával.

4.3.12 Projectile

- **Felelősség**

A Speciális lövedékek kezelése. Tudnia kell a Speciális lövedék pozícióját, tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább. Speciális fallal történő ütközés esetén létrehoz egy **Stargate** objektumot.

- **Ősosztályok**

Movable

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **Direction direction**: Ebben tárolja a lövedék irányát.
- **PortalType type**: Ebben tárolja a lövedék típusát.

- **Metódusok**

- **Stargate openStargate()**: Ez a függvény kezeli a lövedék speciális fallal való találkozásakor keletkező csillagkapu nyitását.
- **void arriveOnMapElement(MapElement mapElement)**: Ez a függvény kezeli le a lövedék mezőre való érkezését.

- **void leaveMapElement(MapElement mapElement):** Ez a függvény kezeli le a lövedék mezőről való távozását.
- **void move():** A lövedék mozgását megvalósító függvény.

4.3.13 Replicator

- **Felelősség**
A replikátor kezelése. Tudnia kell a pozícióját, tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább.
- **Ősosztályok**
Movable
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **MapElement position:** Aktuális pozíció.
 - **Direction direction:** Aktuális irány.
- **Metódusok**
 - **void arriveOnMapElement(MapElement mapElement):** Ez a függvény kezeli le a replikátor mezőre való érkezését.
 - **void leaveMapElement(MapElement mapElement):** Ez a függvény kezeli le a replikátor mezőről való távozását.
 - **void move():** Replikátor mozgatása.
 - **void destroy():** Megsemmisíti a replikátort.
 - **void turn(Direction dir):** Replikátor forgatása a megadott irányba.
 - **void moveOrTurn(Direction dir):** Replikátor mozgatása vagy forgatása a megadott irányba.

4.3.14 Road

- **Felelősség**
Az út típusú mezők megvalósításáért felelős.
- **Ősosztályok**
MapElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **ZPM zpm:** A Mezőn elhelyezkedő ZPM-re mutató referencia.

- **Metódusok**

- **void handlePlayerArrive(Direction dir, Player player):** Ez a függvény kezeli le az ezredes érkezését az út mezőre.
- **void handlePlayerLeave():** Ez a függvény kezeli le az ezredes távozását az út mezőről.
- **void handleProjectileArrive(Direction dir, Projectile projectile):** Ez a függvény kezeli le a speciális lövedék érkezését az út mezőre
- **void handleBoxPutDown(Direction dir, Box box):** Ez a függvény kezeli le egy doboz lerakását az út mezőre.
- **void handleBoxPickUp():** Ez a függvény kezeli le egy doboz felvételét az út mezőről.
- **void handleReplicatorArrive(Direction dir, Replicator rep):** Ez a függvény kezeli le a replikátor érkezését a mezőre.
- **void handleReplicatorLeave():** Ez a függvény kezeli le a replikátor távozását a mezőről.

4.3.15 Scale

- **Felelősség**

A mérleg típusú mezők megvalósításáért felel. Mindenképpen egy ajtóhoz tartozik. Amennyiben rákerül **legalább egy** játékos vagy **két** doboz, akkor az ajtót átjárhatóvá teszi.

- **Össztályok**

MapElement

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **int weightCount:** Azt tartjuk számon benne, hogy hány olyan objektum van a mezőn, aminek van súlya.
- **Door door:** A Mérleghez tartozó ajtóra mutató referencia.

- **Metódusok**

- **void handlePlayerArrive(Direction dir, Player player):** Ez a függvény kezeli le a **játékosok** érkezését a mérleg mezőre.
- **void handlePlayerLeave():** Ez a függvény kezeli le a **játékosok** távozását a mérlegmezőről.
- **void handleProjectileArrive(Direction dir, Projectile projectile):** Ez a függvény kezeli le a speciális lövedék érkezését a mérleg mezőre
- **void handleBoxPutDown(Direction dir, Box box):** Ez a függvény kezeli le egy doboz lerakását a mérleg mezőre.
- **void handleBoxPickUp():** Ez a függvény kezeli le egy doboz felvételét a mérleg mezőről.
- **void incrementWeight():** A mérlegen elhelyezkedő súlyt növeli.
- **void decrementWeight():** A mérlegen elhelyezkedő súlyt csökkenti.

- **void handleReplicatorArrive(Direction dir, Replicator rep):** Ez a függvény kezeli le a replikátor érkezését a mezőre.
- **void handleReplicatorLeave():** Ez a függvény kezeli le a replikátor távozását a mezőről.

4.3.16 SpecialWall

- **Felelősség**
Felelőssége olyan fal megvalósítása, mely alapesetben átjárhatatlan, de ha **Projectile** éri, akkor egy **Stargate** nyílik rajta, így szomszédos mezőnek tekinthető azzal a **MapElement**tel, mely szintén egy **SpecialWall** és van rajta egy másik **Stargate**.
- **Ősosztályok**
MapElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **HashMap<Direction, Stargate> stargate:** A Mezőn elhelyezkedő csillagkapura egy referencia.
- **Metódusok**
 - **void handlePlayerArrive(Direction dir, Player player):** Ez a függvény kezeli le a **játékosok** érkezését a speciális fal mezőre.
 - **void handlePlayerLeave():** Ez a függvény kezeli le a **játékosok** távozását a speciális fal mezőről.
 - **void handleProjectileArrive(Direction dir, Projectile projectile):** Ez a függvény kezeli le a speciális lövedék érkezését a speciális fal mezőre
 - **void handleBoxPutDown(Direction dir, Box box):** Ez a függvény kezeli le egy doboz lerakását a speciális fal mezőre.
 - **void handleBoxPickUp():** Ez a függvény kezeli le egy doboz felvételét a speciális fal mezőről.
 - **void handleReplicatorArrive(Direction dir, Replicator rep):** Ez a függvény kezeli le a replikátor érkezését a mezőre.
 - **void handleReplicatorLeave():** Ez a függvény kezeli le a replikátor távozását a mezőről.

4.3.17 Stargate

- **Felelősség**
A **Projectile** ha egy **SpecialWall**ba ütközik, akkor jön létre a **Stargate**. Két **Stargate** közötti kapcsolatért felelős ez az osztály.
- **Ősosztályok**
Nincs őszotállya

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **SpecialWall position**: A csillagkapu koordinátájának tárolására van.
- **Direction direction**: A csillagkapu irányát tároljuk benne.
- **PortalType type**: A csillagkapu típusát tároljuk benne.
- **Stargate blueStargate**: A kék csillagkapu
- **Stargate yellowStargate**: A sárga csillagkapu
- **Stargate redStargate**: A piros csillagkapu
- **Stargate greenStargate**: A zöld csillagkapu

- **Metódusok**

- **MapElement getExitPosition()**: A függvény visszaadja a féregjárat kijáratának pozícióját.
- **Direction getExitDirection()**: A függvény visszaadja a féregjárat kijáratának irányát.
- **Stargate createStargate(MapElement e, ProjectileType p, Direction d)**: Elkészíti a megfelelő csillagkaput a paraméterek szerint.

4.3.18 Wall

- **Felelősség**

A fal típusú mezők megvalósításáért felelős. Nem átjárható. Ezekre a mezőkre nem kerülhet semmilyen **Movable** objektum.

- **Ősosztályok**

MapElement

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

Az őszosztálytól örökölt attribútumokon kívül mást nem definiál.

- **Metódusok**

- **void handleProjectileArrive(Direction dir, Projectile projectile)**: Ez a függvény kezeli le a speciális lövedék érkezését a fal mezőre

4.3.19 ZPM

- **Felelősség**

A játék célja, hogy az összes ilyen objektumot felszedjük.

- **Ősosztályok**

Nincs őszosztálya.

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

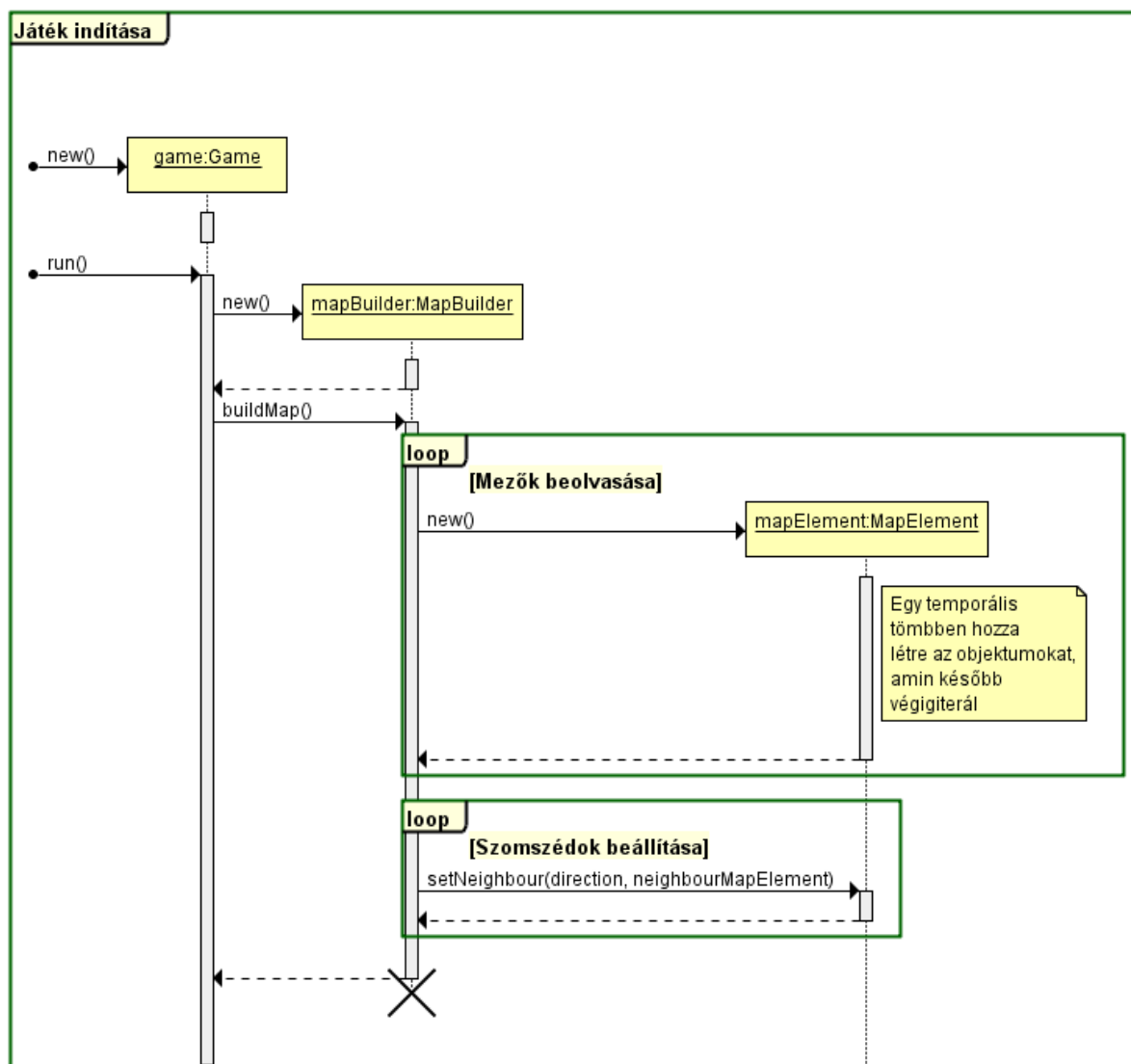
Nincsenek attribútumai

- **Metódusok**

- **void handlePickUp(Player p):** Ez a függvény kezeli le, ha **egy** játékos rálép egy olyan mezőre, ahol ZPM van.

4.4 Szekvencia diagramok

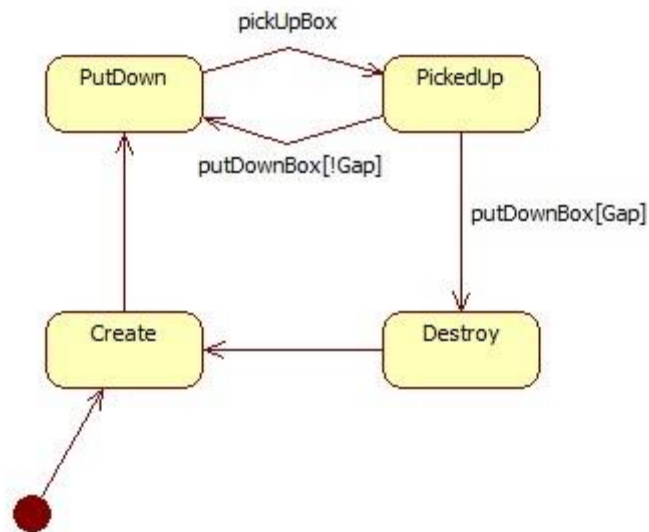
4.4.1 Játék indítása



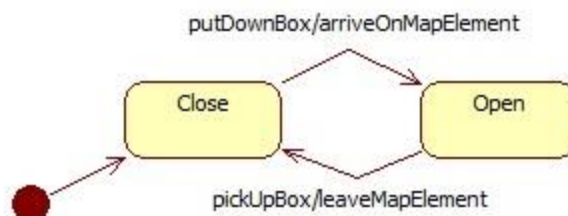
A 4. fejezetben beadott szekvencia diagramok az 5. fejezetben módosításra kerültek, a játék indításán kívül mindegyiket módosítottuk, bővítettük, részleteztük. További szekvencia diagramok az 5.3 *Szekvencia diagramok belső működése* című fejezetben találhatóak.

4.5 State-chartok

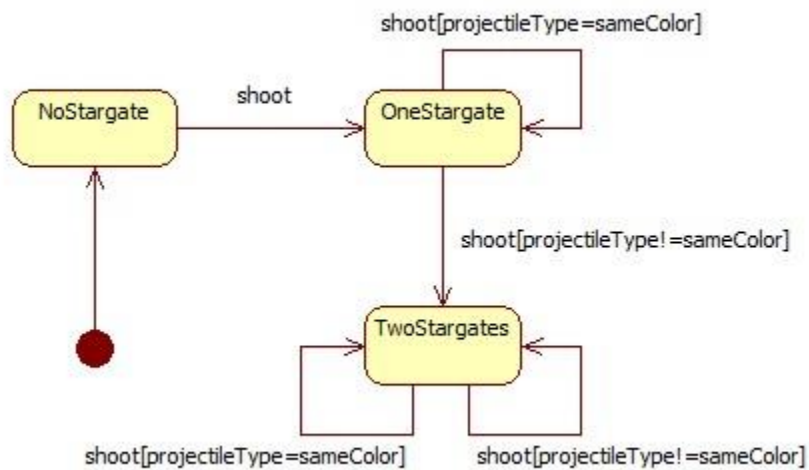
4.5.1 Box



4.5.2 Door



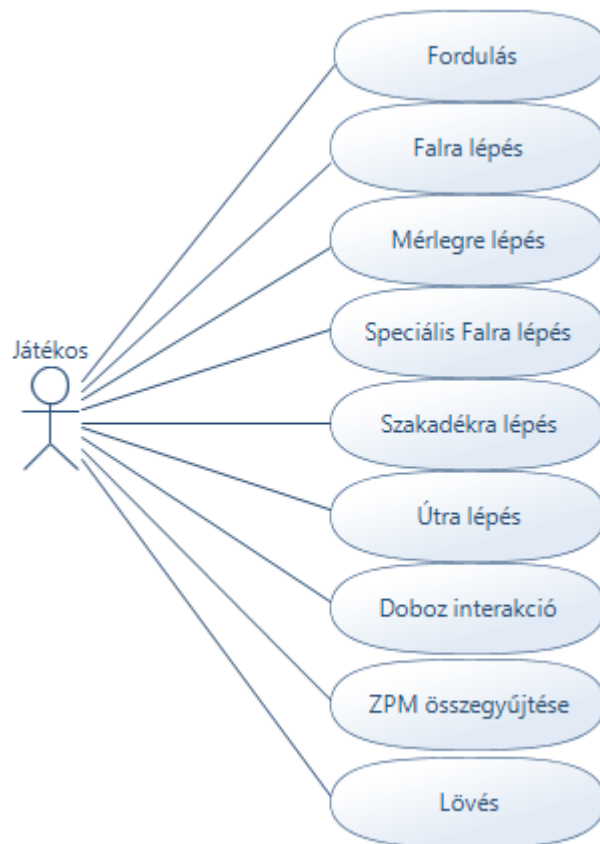
4.5.3 Stargate



5 Szkeleton tervezése

5.1 A szkeleton modell valóságos use-case-ei

5.1.1 Use-case diagram



5.1.2 Use-case leírások

Use-case neve	Falra lépés
Rövid leírás	A következő fal mezőre lépés
Aktorok	Játékos
Forgatókönyv	A játékos visszapattan a fal előtti mezőre.

Use-case neve	Mérlegre lépés
Rövid leírás	A következő mérleg mezőre lépés
Aktorok	Játékos
Forgatókönyv	Kinyitja a hozzá tartozó ajtót.

Use-case neve	Speciális falra lépés
Rövid leírás	A következő speciális fal mezőre lépés
Aktorok	Játékos
Forgatókönyv	Ha található az érkezési iránynak megfelelő oldalon csillagkapu és a másik fajta csillagkapu is létezik, akkor az ezredes áthalad a féregjáraton.

Use-case neve	Szakadékra lépés
Rövid leírás	A következő szakadék mezőre lépés
Aktorok	Játékos
Forgatókönyv	Az ezredes meghal.

Use-case neve	Útra lépés
Rövid leírás	A következő út mezőre lépés
Aktorok	Játékos
Forgatókönyv	A játékos átlép a következő mezőre.

Use-case neve	Doboz interakció
Rövid leírás	Dobozokkal való műveletek őse
Aktorok	Játékos
Forgatókönyv	A játékos felvesz vagy letesz egy dobozt.

Use-case neve	Fordul
Rövid leírás	Ezredes irányának megváltoztatása
Aktorok	Játékos
Forgatókönyv	Ezredes irányának megváltoztatása a lenyomott gombnak megfelelően.

Use-case neve	ZPM összegyűjtése
Rövid leírás	ZPM modul összegyűjtése
Aktorok	Játékos
Forgatókönyv	Az ezredessel egy mezőn lévő ZPM modul összegyűjtése

Use-case neve	Lövés
Rövid leírás	Egy lövedék kilövése
Aktorok	Játékos
Forgatókönyv	A speciális töltény színének megfelelő lövedék lövése a játékos irányával megegyezően egyenes vonalban.

5.2 A szkeleton kezelői felületének terve, dialógusok

A szkeleton tesztelésére szolgáló parancssoros felület a szöveges kimenetre írja a választható menüpontokat, majd a felhasználó által kiválasztott menüponthoz tartozó kódrészletet futtatja.

A menü két szintből áll, az első szinten találhatjuk a fő menüpontokat, melyek kategóriákba rendezik a tesztelendő kódrészleteket, ezeket kiválasztva, jutunk el az adott kategóriába tartozó tesztesetekhez, melyeket szintén a felhasználó választhat ki, majd ezeket a program automatikusan futtatja. A teszteset futtatása után újra a főmenüben találja magát a felhasználó. Az egyes teszteseteket a korábbi use-case diagramnak megfelelően alakítottuk ki.

A felhasználó a lent felsorolt menüpontok előtti szám beírásával tudja azokat kiválasztani.

A főmenü felépítése:

- (1) Játékos mozgatása
- (2) Doboz interakció
- (3) Lövés
- (4) ZPM felvétele
- (5) Kilépés

A főmenü egyes pontjaihoz tartozó tesztesetek:

- Játékos mozgatása
 - (1) Játékos mozgatása útra
 - (2) Játékos mozgatása mérlegre
 - (3) Játékos mozgatása falra
 - (4) Játékos mozgatása speciális falra (nyitott portál)
 - (5) Játékos mozgatása szakadékba
- Doboz interakció
 - (1) Felvétel útról
 - (2) Felvétel mérlegről
 - (3) Felvétel portálon átnyúlva
 - (4) Lerakás útra
 - (5) Lerakás mérlegre
 - (6) Lerakás szakadékba
 - (7) Lerakás falra
 - (8) Lerakás portálon keresztül
- Lövés
 - (1) Lövés falra
 - (2) Lövés speciális falra
- ZPM felvétel
 - (1) Utolsó ZPM felvétele

A teszteset kiválasztása után a program betölti a megfelelő pályát és lefuttatja a szükséges parancsokat.

A játék állapotára vonatkozó információk szöveges formában jelennek meg a parancssor standard kimenetén, melyen visszakövethetőek lesznek az egyes metódushívások, azok paraméterei, illetve a hívó fél is. Így a kimenet könnyen összeköthető lesz a szekvencia - diagramokkal.

A tesztesetek kimeneteinek formai követelményei:

Az egyes metódushívások a következő formátumban jelennek meg a kimenten:
Osztály.Metódus(Param1, Param2, ...)

A visszatérési értékeket pedig a következő formátumban jelenítjük meg:
Osztály.Metódus returns VisszatérésiÉrték

Játékos mozgatása útra:

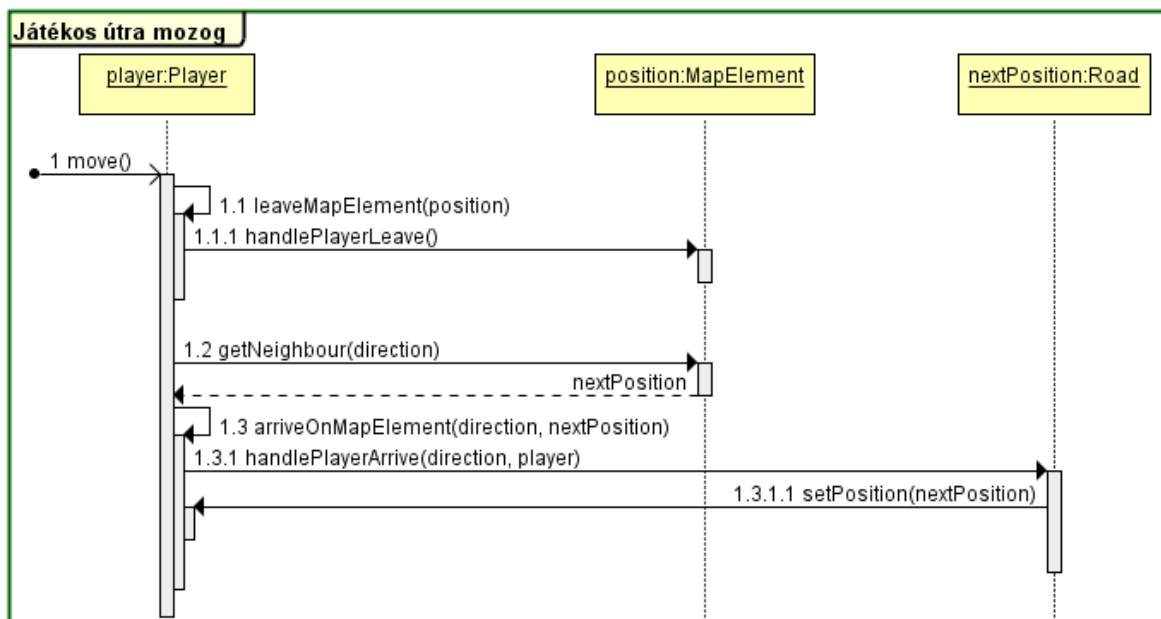
```
Player.move()  
    Player.leaveMapElement(Road)  
        Road.handlePlayerLeave()  
    Road.getNeighbour(SOUTH)  
        return Road  
    Player.arriveOnMapElement(SOUTH, Road)  
        Road.handlePlayerArrive(SOUTH, Player)  
        Player.setPosition(Road)
```

A teszteset lefutása után, ha a felhasználó entert nyom újra megjelenik a menü.

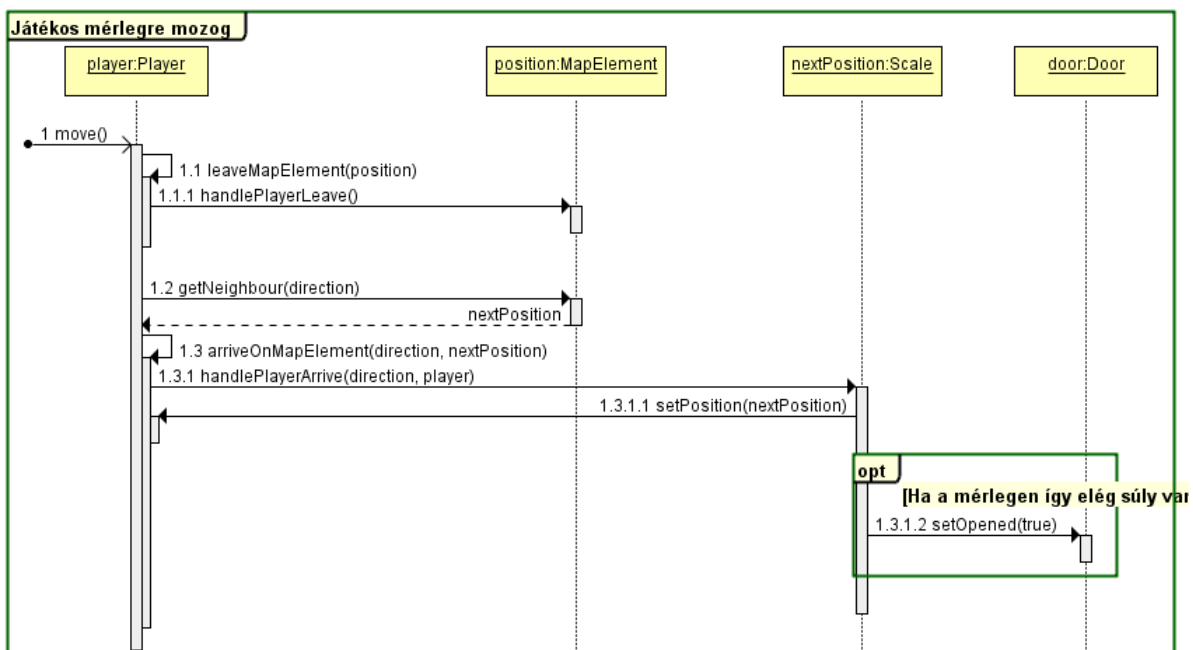
5.3 Szekvencia diagramok belső működésre

5.3.1 Játékos mozgása

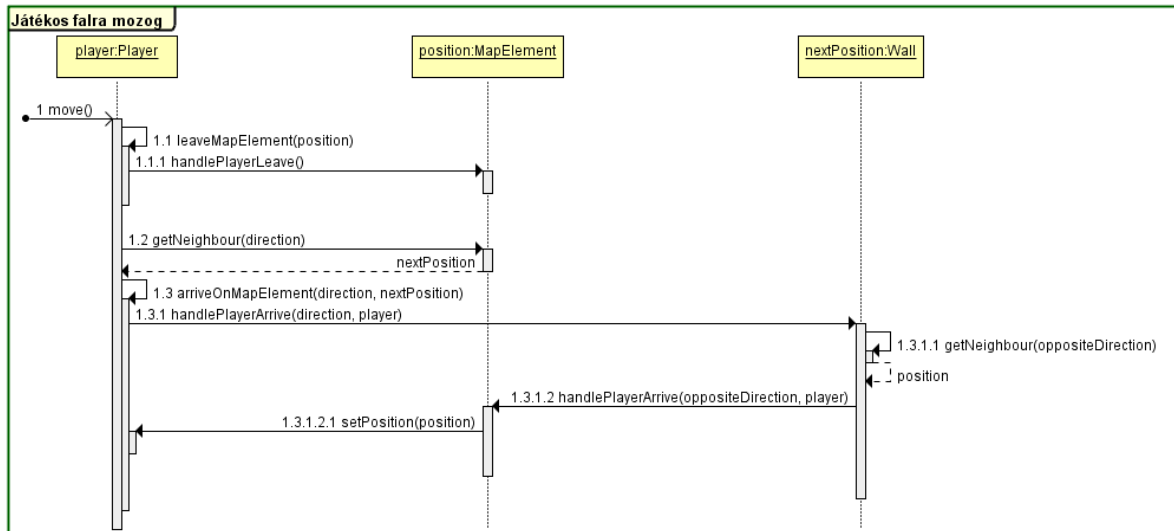
5.3.1.1 Játékos útra mozog



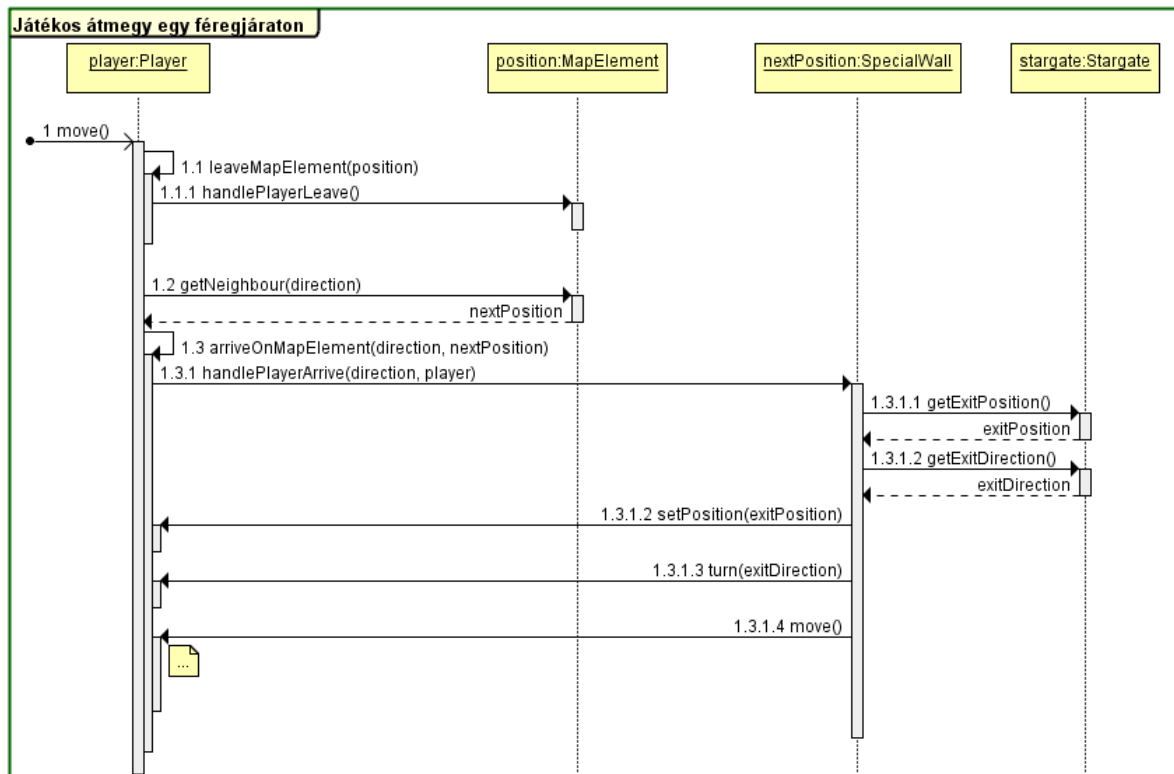
5.3.1.2 Játékos mérlegre mozog



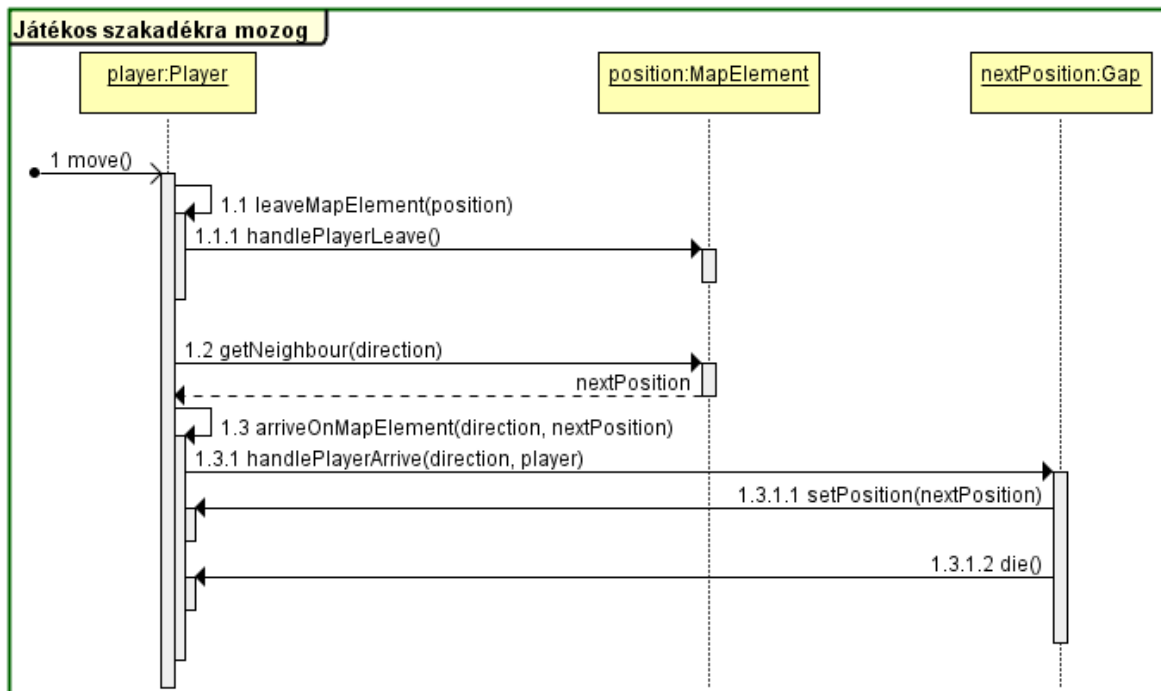
5.3.1.3 Játékos falra mozog



5.3.1.4 Játékos átmegy egy féregjáraton

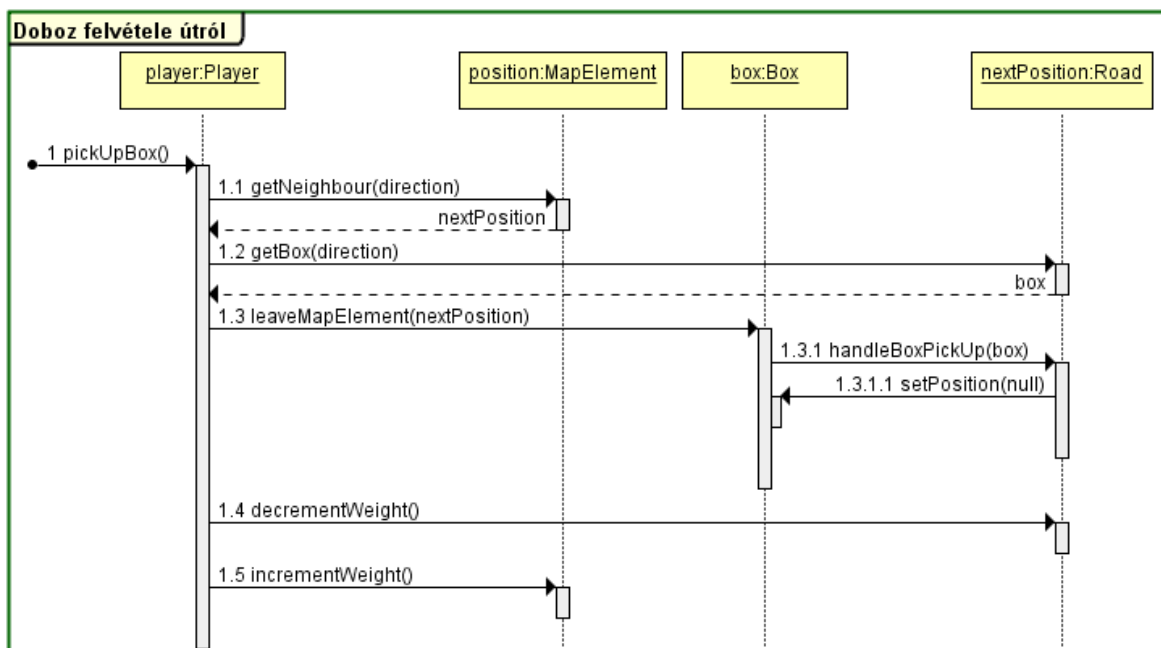


5.3.1.5 Játékos szakadékba mozog

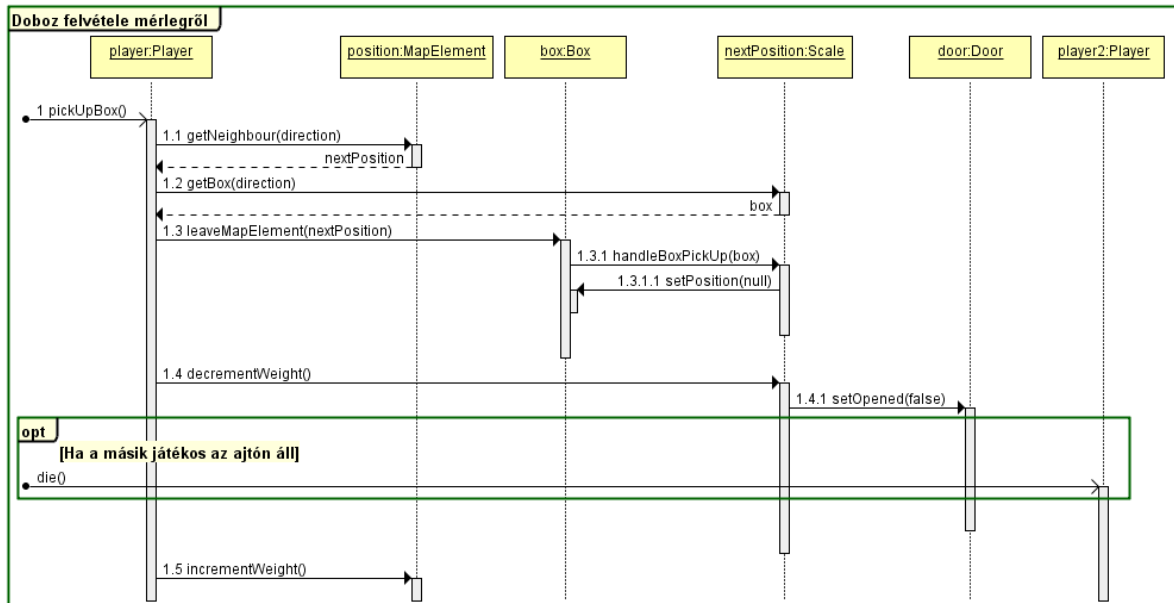


5.3.2 Doboz iterakciók

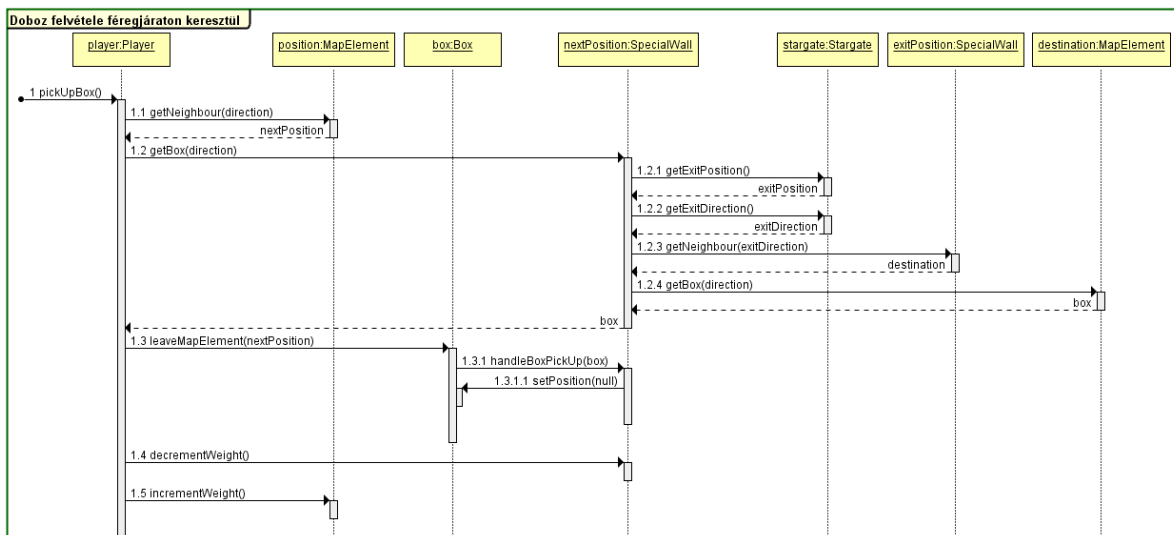
5.3.2.1 Doboz felvétele útról



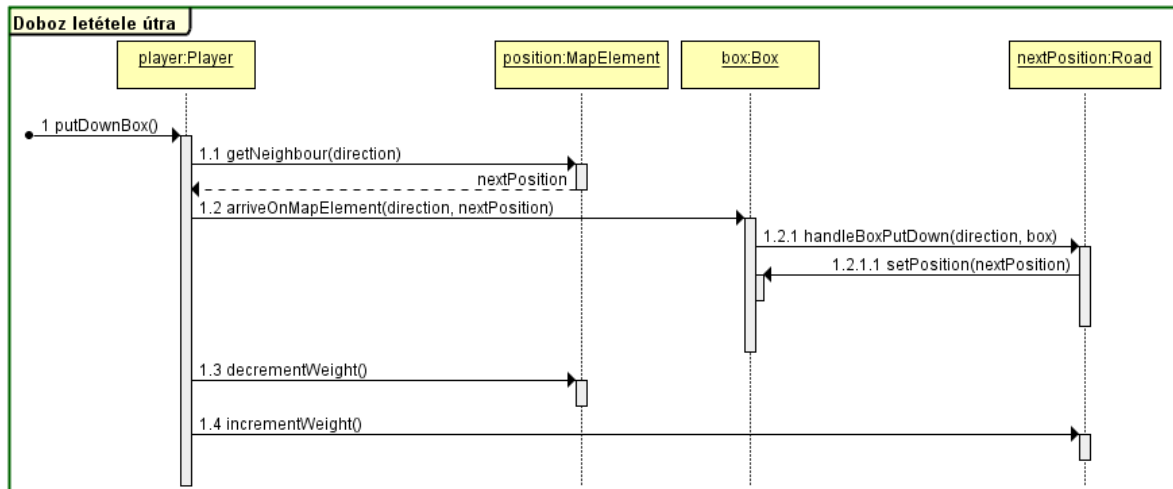
5.3.2.2 Doboz felvétele mérlegről



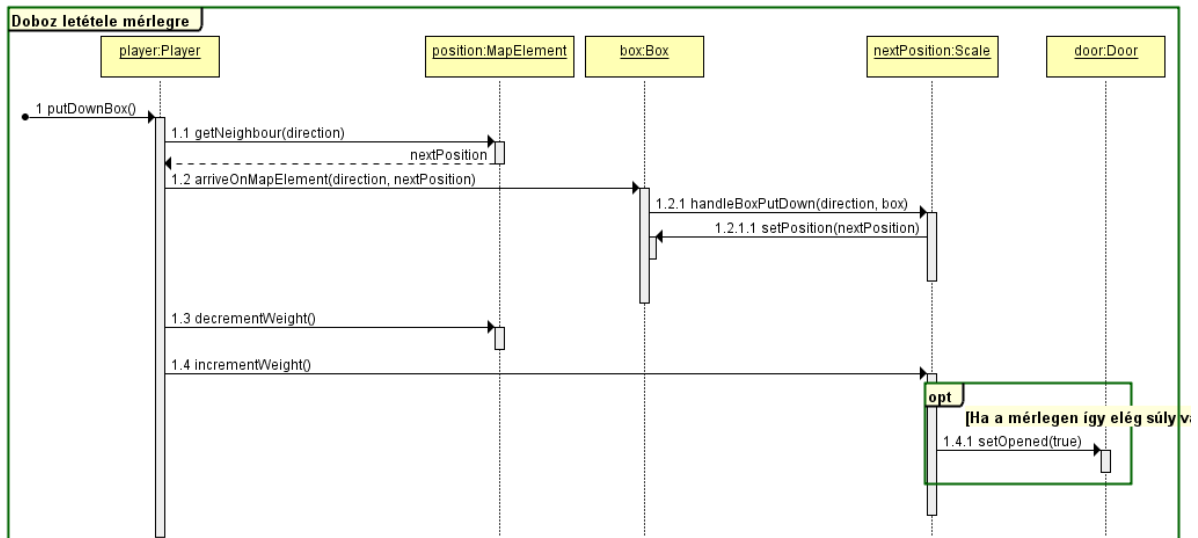
5.3.2.3 Doboz felvétele féregjáraton keresztül



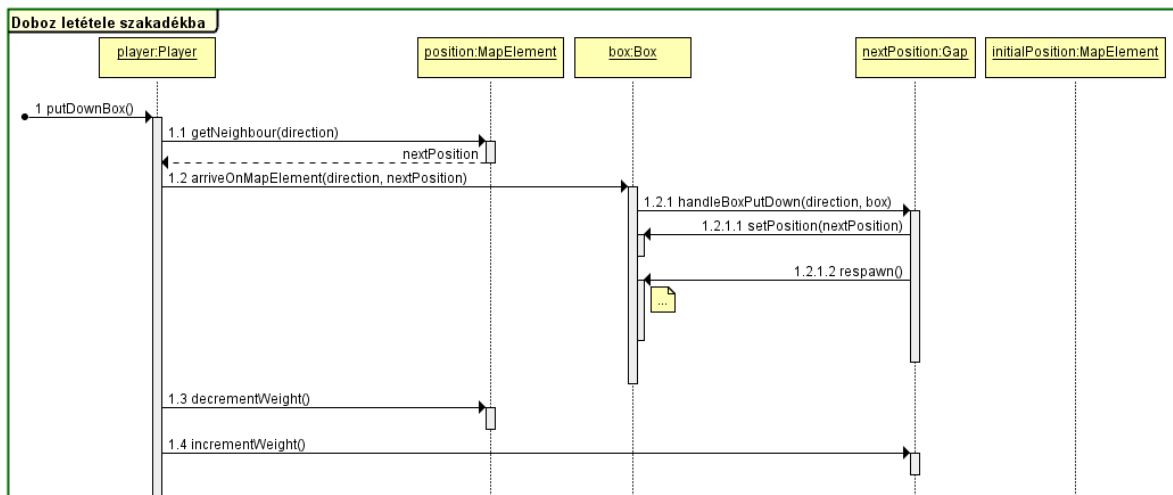
5.3.2.4 Doboz letétele útra



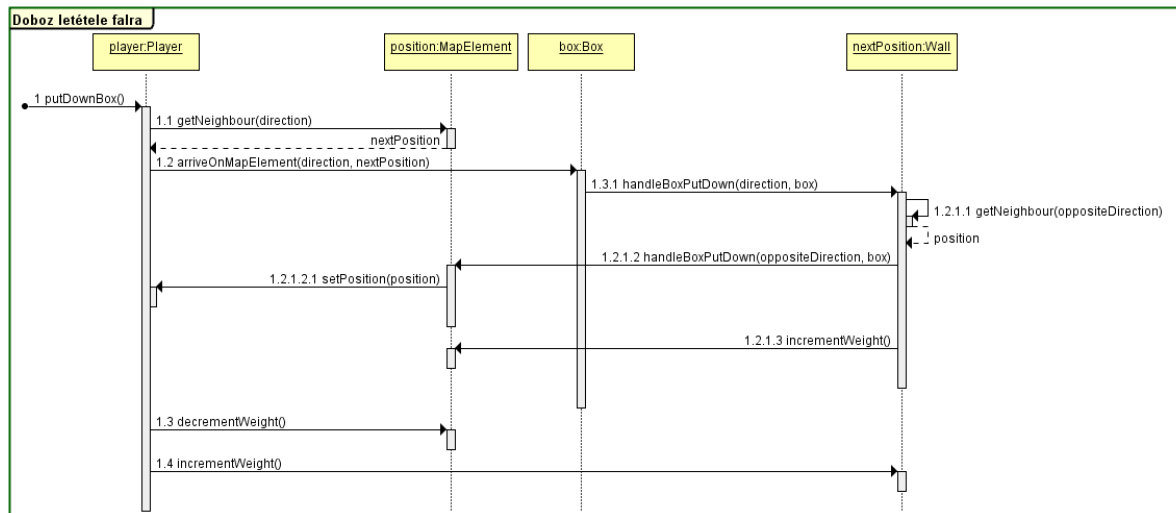
5.3.2.5 Doboz letétele mérlegre



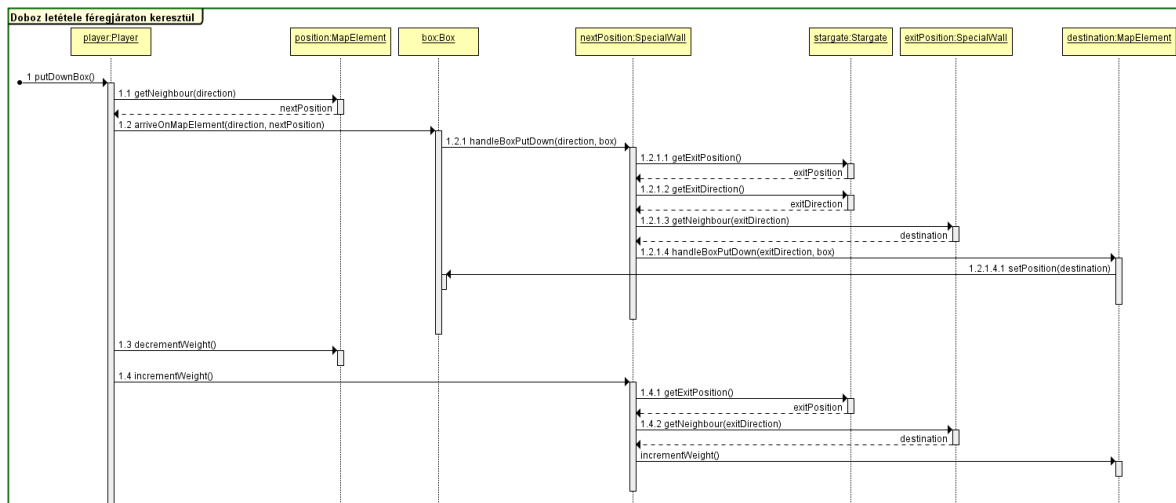
5.3.2.6 Doboz letétele szakadékba



5.3.2.7 Doboz letétele falra

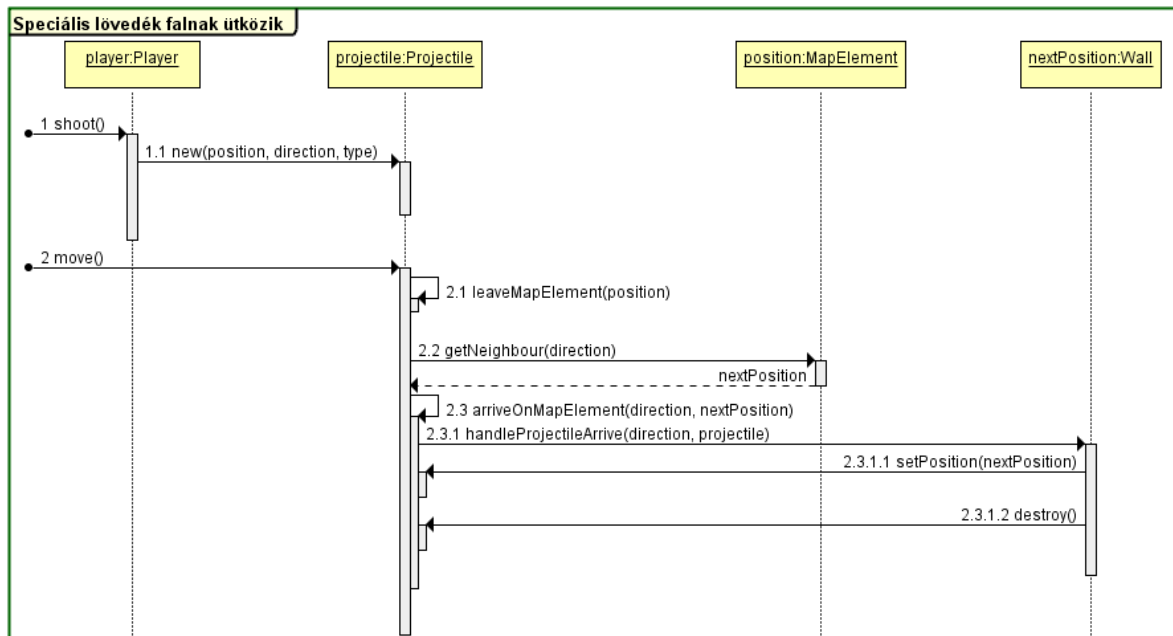


5.3.2.8 Doboz letétele féregjáraton keresztül

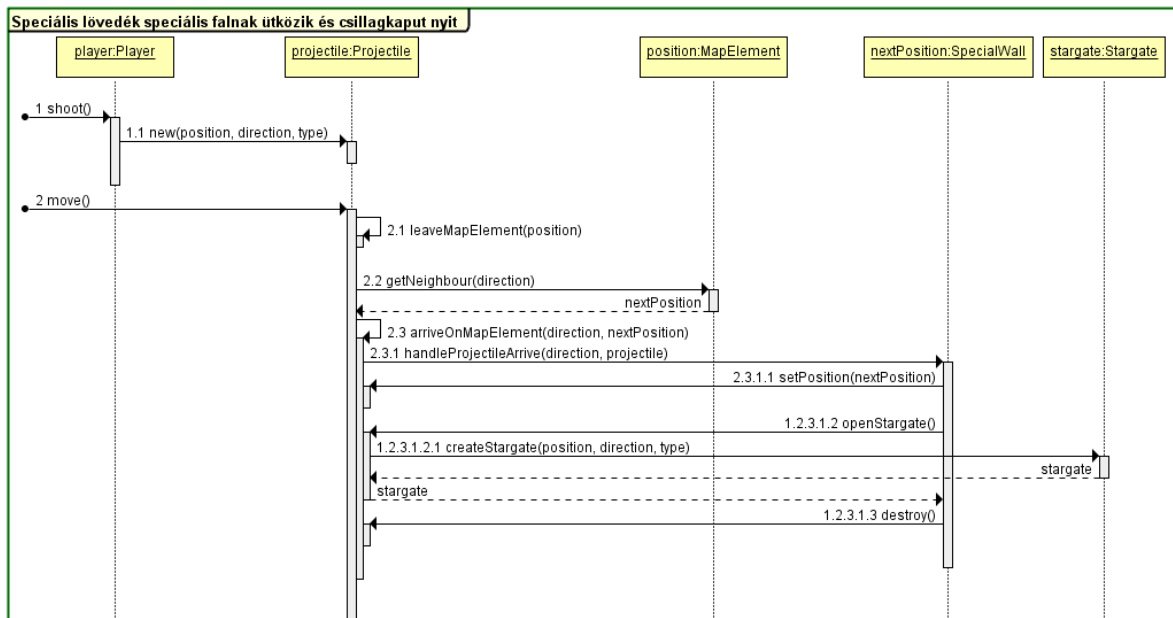


5.3.3 Lövés

5.3.3.1 Speciális lövedék falnak ütközik



5.3.3.2 Speciális lövedék speciális falnak ütközik és csillagkaput nyit

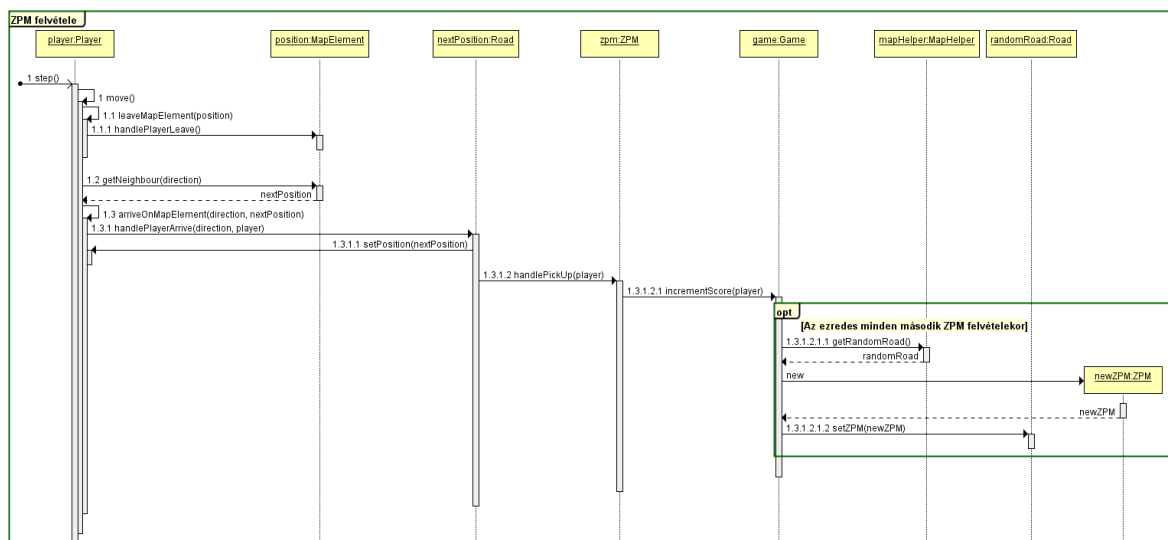


5.3.3.3 Speciális lövedék Replikátornak ütközik



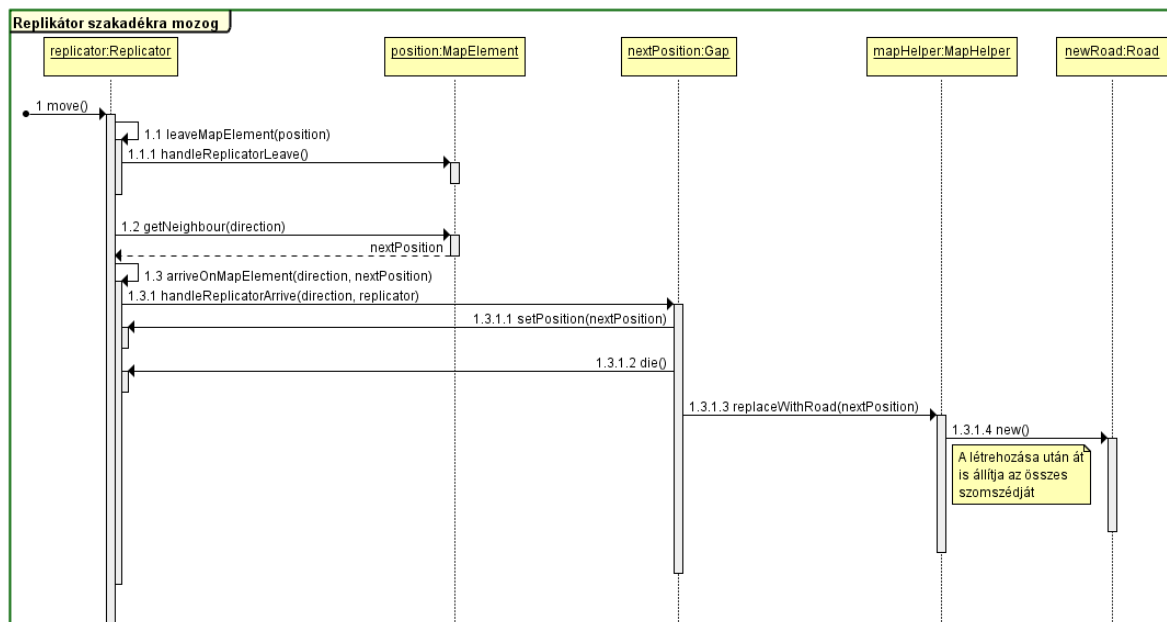
5.3.4 ZPM felvétele

5.3.4.1 ZPM felvétele



5.3.5 Replikátor mozgatása

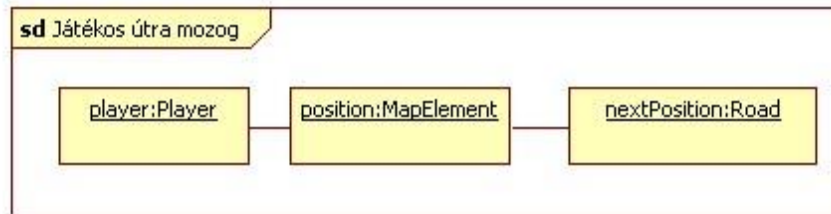
5.3.5.1 Replikátor szakadékra mozog



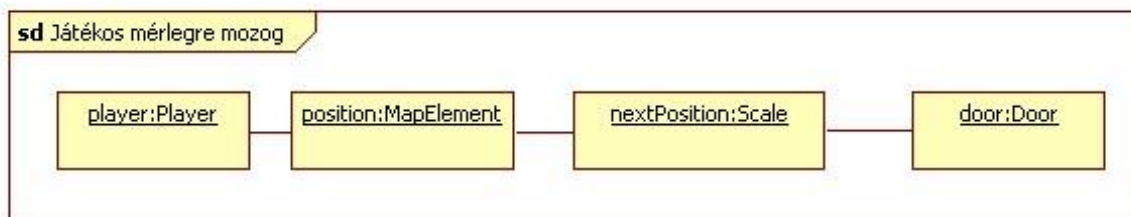
5.4 Kommunikációs diagramok

5.4.1 Játékos mozgatása

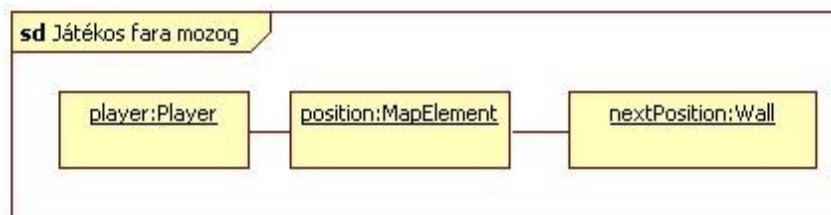
5.4.1.1 Játékos útra mozog



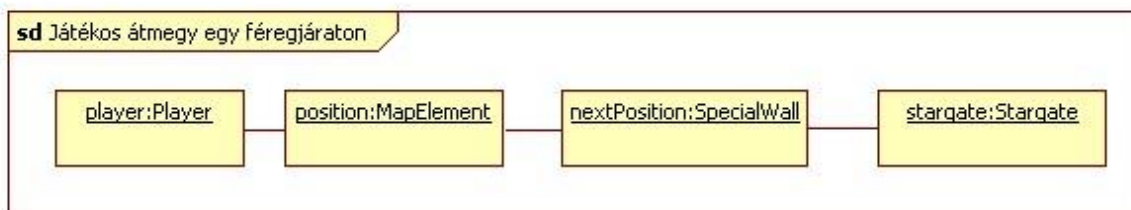
5.4.1.2 Játékos mérlegre mozog



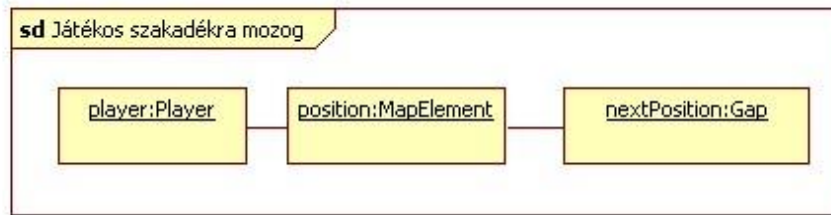
5.4.1.3 Játékos falra mozog



5.4.1.4 Játékos átmegy egy féregjáraton

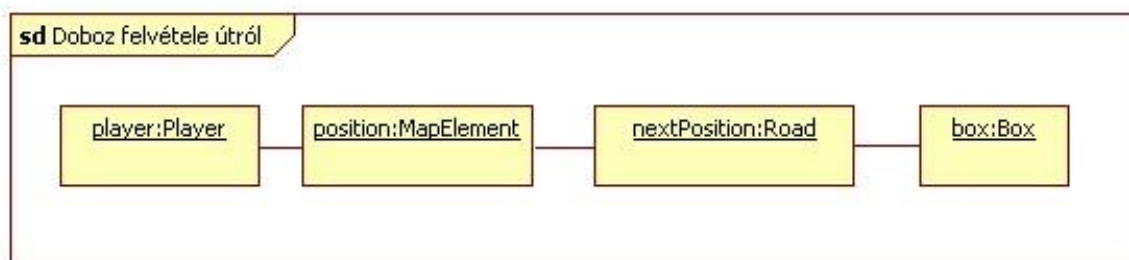


5.4.1.5 Játékos szakadékba mozog

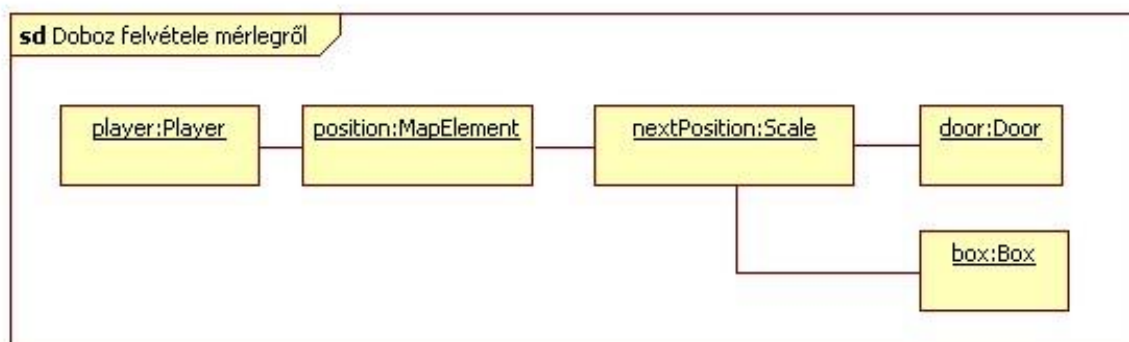


5.4.2 Doboz iterakciók

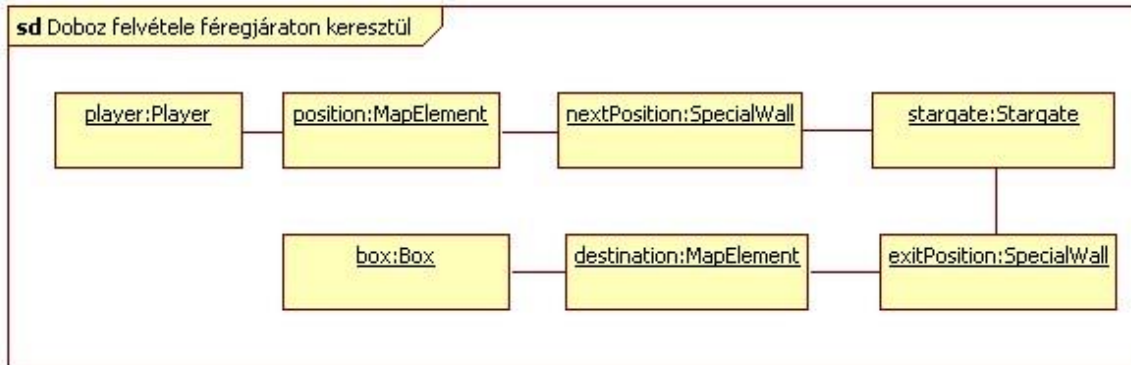
5.4.2.1 Doboz felvétele útról



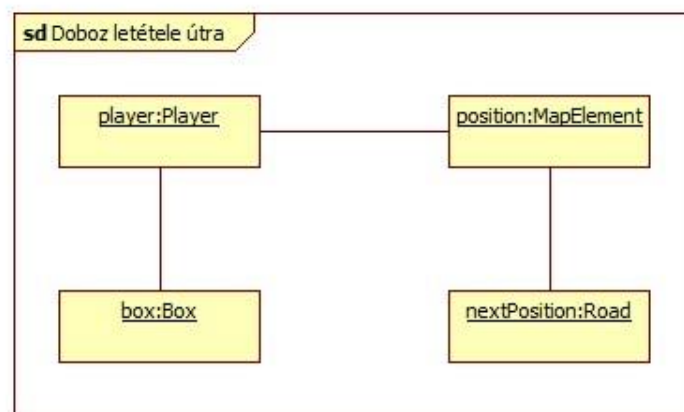
5.4.2.2 Doboz felvétele mérlegről



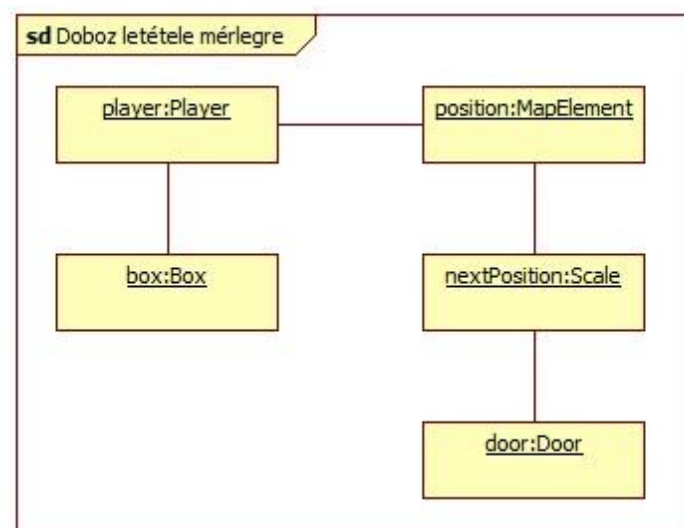
5.4.2.3 Doboz felvétele féregjáraton keresztül



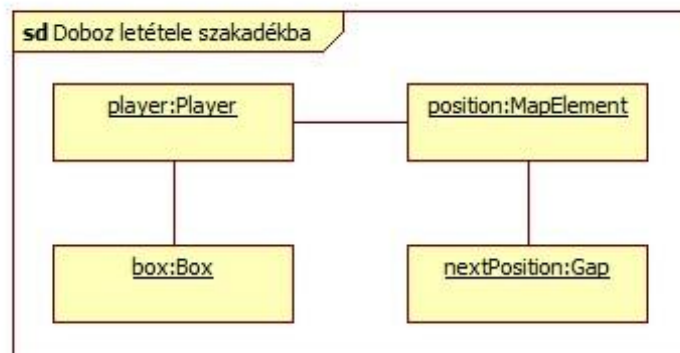
5.4.2.4 Doboz letétele útra



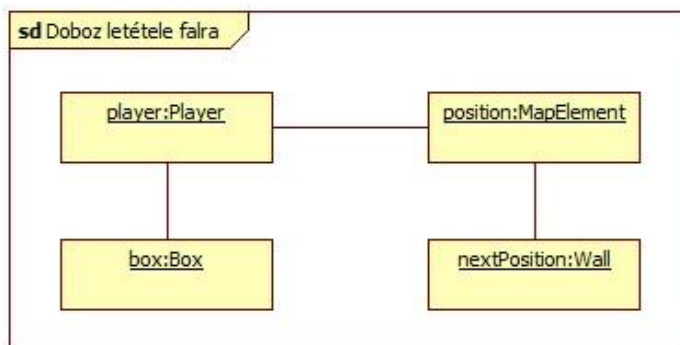
5.4.2.5 Doboz letétele mérlegre



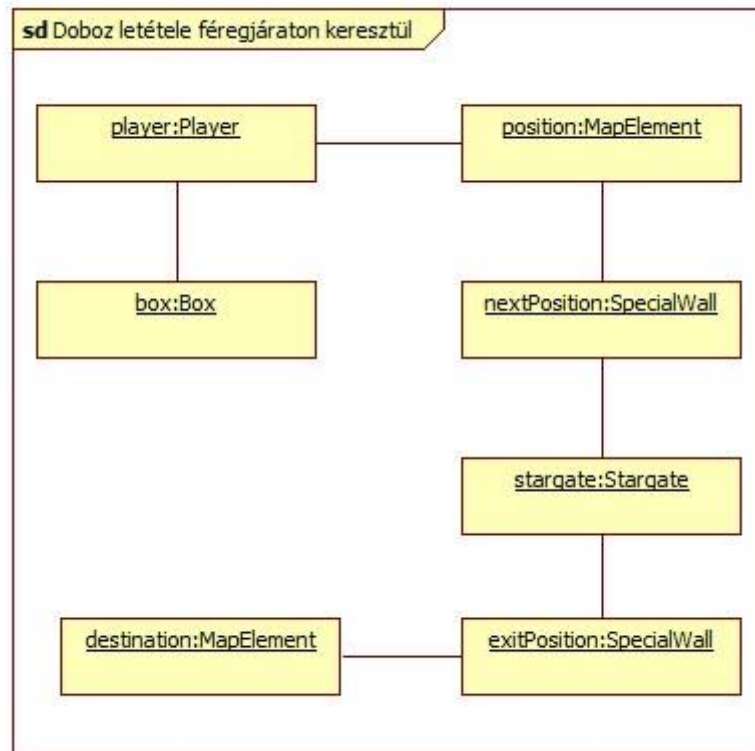
5.4.2.6 Doboz letétele szakadékba



5.4.2.7 Doboz letétele falra

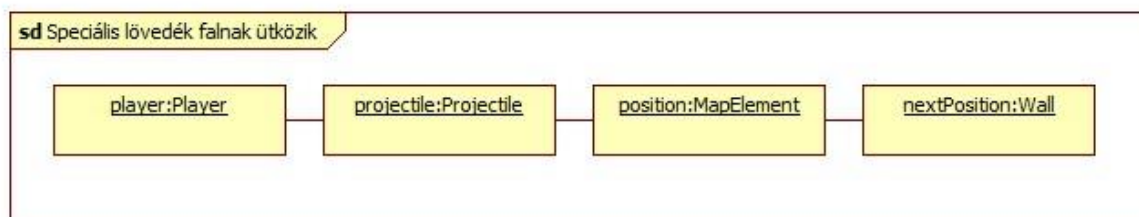


5.4.2.8 Doboz letétele féregjáraton keresztül

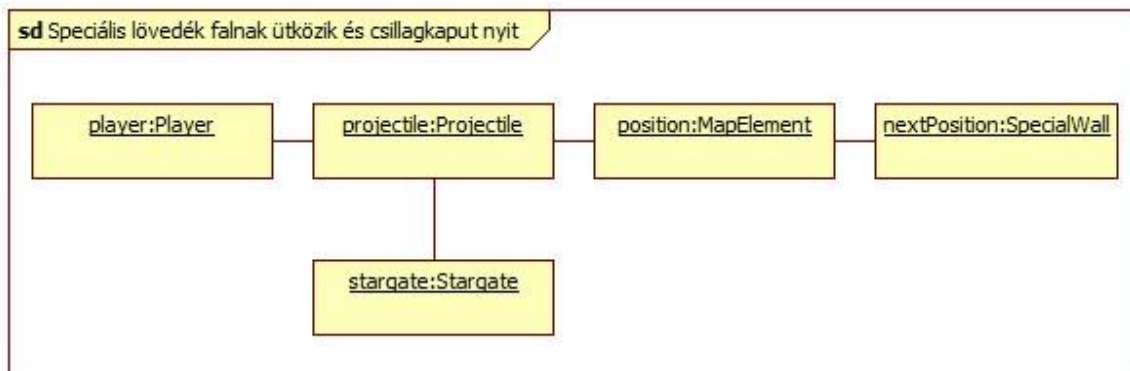


5.4.3 Lövés

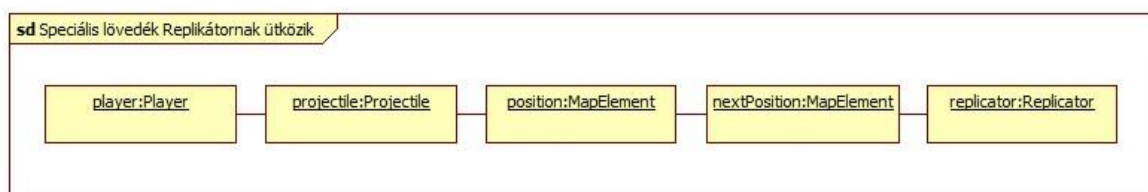
5.4.3.1 Speciális lövedék falnak ütközik



5.4.3.2 Speciális lövedék speciális falnak ütközik és csillagkaput nyit

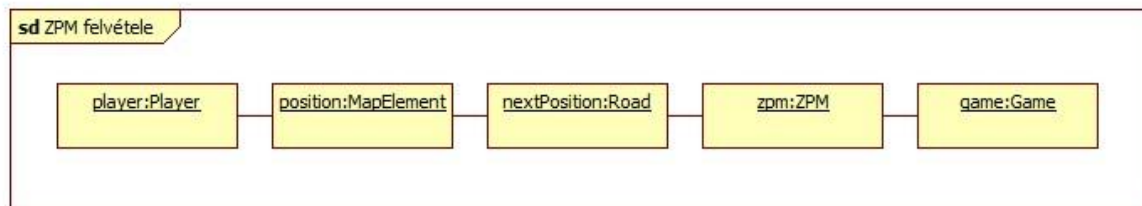


5.4.3.3 Speciális lövedék Replikátornak ütközik



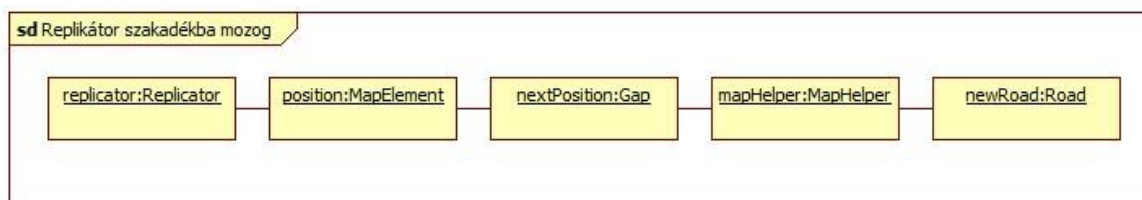
5.4.4 ZPM felvétele

5.4.4.1 ZPM felvétele



5.4.5 Replikátor mozgatása

5.4.5.1 Replikátor szakadékokra mozog



6 Szkeleton beadás

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Controller.java	2440 byte	2016.03.29. 1:12	Controller osztály.
Game.java	1270 byte	2016.03.29. 1:42	Game osztály. Számon tartja a játék pontszámát.
MapBuilder.java	3010 byte	2016.03.28. 22:54	MapBuilder osztály. Felépíti a pályát, létrehozza és elhelyezi rajta a játékost, a ZPM-et és a dobozt.
SkeletonLogger.java	1910 byte	2016.03.28. 3:14	SkeletonLogger osztály. Paraméter függvényhívások és visszatérési értékek hierarchikus kiírása.
Box.java	1590 byte	2016.03.29. 1:04	Box osztály.
Direction.java	375 byte	2016.03.29. 0:25	Direction osztály.
Door.java	3140 byte	2016.03.29. 1:04	Door osztály.
Gap.java	1040 byte	2016.03.28. 21:50	Gap osztály.
MapElement.java	2700 byte	2016.03.29. 1:04	Mapelemt osztály.
Movable.java	962 byte	2016.03.29. 2:18	Moveable osztály.
Player.java	3400 byte	2016.03.28. 21:35	Player osztály.
Projectile.java	2080 byte	2016.03.29. 1:04	Projectile osztály.
ProjectileType.java	259 byte	2016.03.28. 11:02	ProjectileType osztály.
Road.java	1490 byte	2016.03.29. 1:04	Road osztály.
Scale.java	2050 byte	2016.03.29. 1:04	Scale osztály.
SpecialWall.java	4150 byte	2016.03.29. 1:04	SpecialWall osztály.
Stargate.java	1900 byte	2016.03.28. 22:47	Stargate osztály.
Wall.java	447 byte	2016.03.28. 21:50	Wall osztály.
ZPM.java	508 byte	2016.03.29. 1:04	ZPM osztály.
IProjectileStateListener.java	326 byte	2016.03.28. 13:31	A Projectile keletkezésével és megsemmisülésével kapcsolatos üzeneteket fogadó osztályok interfésze
IZPMPickedUpListener.java	179 byte	2016.03.28. 13:31	A ZPM és felvételével kapcsolatos üzeneteket fogadó osztályok interfésze
ModelEventSource.java	1970 byte	2016.03.28. 23:23	A játékmodel változásairól küldött üzeneteket küldő osztály

BoxPickUpRoadTest.java	487 byte	2016.03.29. 1:04	BoxPickUpRoadTest osztály. Doboz felvétele útról akció tesztelése.
BoxPickUpScaleTest.java	709 byte	2016.03.29. 1:04	BoxPickUpScaleTest osztály. Doboz felvétele mérlegről akció tesztelése.
BoxPickUpWormholeTest.java	855 byte	2016.03.29. 1:04	BoxPickUpWormholeTest osztály. Doboz felvétele féregjáraton keresztül akció tesztelésle.
BoxPutDownGapTest.java	670 byte	2016.03.29. 1:04	BoxPutDownGapTest osztály. Doboz letétele szakadékba akció tesztelése.
BoxPutDownRoadTest.java	606 byte	2016.03.29. 1:04	BoxPutDownRoadTest osztály. Doboz letétele útra akció tesztelése.
BoxPutDownScaleTest.java	738 byte	2016.03.29. 1:04	BoxPutDownScaleTest osztály. Doboz letétele mérlegre akció tesztelése.
BoxPutDownWallTest.java	679 byte	2016.03.29. 1:04	BoxPutDownWallTest osztály. Doboz letétele falra akció tesztelése.
BoxPutDownWormholeTest.java	1019 byte	2016.03.29. 1:58	BoxPutDownWormholeTest osztály. Doboz letétele féregjáraton keresztül akció tesztelése.
PlayerMoveGapTest.java	433 byte	2016.03.29. 1:04	PlayerMoveGapTest osztály. A játékos szakadékba mozog akció tesztelése.
PlayerMoveRoadTest.java	448 byte	2016.03.29. 1:04	PlayerMoveRoadTest osztály. A játékos út mezőre mozog akció tesztelése.
PlayerMoveScaleTest.java	384 byte	2016.03.29. 1:04	PlayerMoveScaleTest osztály. A játékos mérlegre mozog akció tesztelése.
PlayerMoveWallTest.java	936 byte	2016.03.29. 1:04	PlayerMoveWallTest osztály. A játékos falra mozog akció tesztelése.
PlayerMoveWormholeTest.java	676 byte	2016.03.29. 1:04	PlayerMoveWormholeTest osztály. A játékos féregjáraton keresztül mozog akció tesztelése.
ShootSpecialWallTest.java	500 byte	2016.03.29. 1:04	ShootSpecialWallTest osztály. Lövedék speciális falra érkezése akció tesztelése.
ShootWallTest.java	552 byte	2016.03.29. 1:04	ShootWallTest osztály. Lövedék falra érkezése akció tesztelése.
Test.java	427 byte	2016.03.28. 14:57	Test osztály. Minden tesztelésre használt osztály őosztálya.
TestFactory.java	1440 byte	2016.03.28. 21:35	TestFactory osztály. Visszaadja az adott teszt típusnak megfelelő tesztet.
TestRunner.java	792 byte	2016.03.29. 1:04	TestRunner osztály. Elindít egy új játékot. Legyártja a tesztek és futtatja azokat.
TestType.java	306 byte	2016.03.28. 14:57	TestType osztály. Teszt típusok.
ZPMTTest.java	914 byte	2016.03.29. 1:04	ZPMTTest osztály. ZPM felvétel akció tesztelése.
Main.java	11800 byte	2016.03.29. 1:25	Main osztály. Menü. A program belépési pontja.

6.1.2 Fordítás

A beadott könyvtárstruktúra gyökerében található *compile.bat* szkriptet futtatva jönnek létre a class fájlok.Futtatás

6.1.3 Futtatás

A beadott könyvtárstruktúra gyökerében található *run.bat* szkriptet futtatva indítható el a program (természetesen csak sikeres fordítás után).

A szkeletonnak a következőkre van szüksége a futáshoz:

Azon.	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
E01	HSZK-s számítógép	kiértékelés	alapvető	megrendelő	JRE 1.6+
E02	monitor	nincs	alapvető	csapat	
E03	billentyűzet	nincs	alapvető	csapat	

Ezekon kívül szükséges, hogy a futtató gépén a megfelelő verziójú JRE és JDK szerepeljen a környezeti változók között.

6.2 Értékelés

Tag neve	Munka százalékban
Csóka Máté	20
Rosner Roland	20
Sándor Mátyás Márton	20
Szalisznyó Ivett	20
Vándor Dávid	20

7 Prototípus koncepciója

7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A prototípus a felhasználóval való kommunikációra a szabványos be- és kimeneti csatornákat használja, így a prototípus megfelelően felparaméterezett indításával a program be- és kimenete a parancssor helyett lehet akár egy tetszőleges szöveges fájl is.

7.1.2 Bemeneti nyelv

A pályákat leíró nyelv emberi szem számára is könnyen értelmezhető. A pálya leírása egy csv-szerű szöveges formátumot követ. A mezők típusát angol nevük kezdőbetűjével jelöljük, a betűk egy soron belül vesszőkkel vannak elválasztva. A mező típusát opcionálisan követheti egy alulvonás karakter, majd ezután egymás után a mezőn tartózkodó egyéb pályaelemeket jelölő szimbólumok. A mérleget és a hozzá tartozó ajtót a leírásban megadott azonosító alapján kapcsoljuk össze.

Az érvényes jelölések

R	Út típusú mező
W	Fal típusú mező
G	Szakadék típusú mező
S:1	Mérleg típusú mező, 1-es azonosítóval
D:1	Ajtó típusú mező, 1-es azonosítóval
SW	Speciális fal típusú mező

Mezőkön található egyéb objektumok jelölése

R_B	Út típusú mező egy doboz szal
R_BBB	Út típusú mező három doboz szal
R_Z	Út típusú mező egy ZPM -mel
R_O	Út típusú mező, melyen O'Neill ezredes áll. Az ezredes kezdetben északra néz.
R_J	Út típusú mező, melyen a Jaffa áll. A Jaffa kezdetben északra néz.
R_R	Út típusú mező, melyen a replikátor tartózkodik.

Példa egy pálya leírására

```
W,      G,      G,      W,      W,  
SW,    R_B,    G,      R_O,    SW,  
W,     D:1,    G,      R_R,    W,  
W,     R_Z,    G,      S:1,    SW,  
W,     SW,    R_JB,    R_B,    W,  
W,     W,     SW,     W,      W,
```

A példán jól látszik, hogy a whitespace-eknek (space, tab) nincsen jelentősége, jelenlétük egyedül az olvashatóságot javítja. Látszik továbbá az is, hogy amennyiben azt a mező típusa megengedi, azon több, különböző típusú objektum is szerepelhet - a leírásban ezek sorrendje lényegtelen.

Amennyiben a pálya leírásában szintaktikai (ismeretlen karakter, stb...) hiba van, azt a program a pálya beolvasásakor (az adott teszteset futtatása elején) a szabványos kimeneten jelezni fogja. Amennyiben azonban a pálya a megadott szintaxisnak megfelel, viszont a játék logikájának ellentmondó modellt tartalmaz (nincs lehelyezett játékos, doboz egy falon van, stb...), nem garantált, hogy az esetleges hibát a pálya beolvasásakor kiszűri a program!

A tesztlépések nyelve

A tesztek bemeneti nyelvének parancsai egy vagy két elemből épülnek fel: a cselekvő objektum megadásából és opcionálisan a cselekvés típusának leírásából.

restart

Leírás: A prototípus újraindítása.

oneill

Leírás: Paraméter nélkül az ezredes állapotát írja ki a szabványos kimenetre (pozíció, szomszéd mezők típusai, birtokolt doboz, pontok száma, stb...)

Opciók:

- **north / west / south / east:** Az ezredest a megadott irányba fordítja, vagy ha már abba az irányba néz, akkor el is mozgatja.
- **proj1 / proj2:** Az ezredes a speciális fegyveréből a megfelelő színű (sárga vagy kék) lövedéket lő abba az irányba, amerre éppen néz.
- **pickup:** Az ezredes, ha még nincsen nála, felvesz egy dobozt az előtte lévő mezőről.
- **putdown:** Az ezredes, ha van nála doboz, leteszi azt az előtte lévő mezőre.

jaffa

Leírás: A Jaffa állapotát írja ki a szabványos kimenetre (pozíció, szomszéd mezők típusai, birtokolt doboz, pontok száma, stb...)

Opciók:

- **north / west / south / east:** A Jaffát a megadott irányba fordítja, vagy ha már abba az irányba néz, akkor el is mozgatja.
- **proj1/proj2:** A Jaffa a speciális fegyveréből a megfelelő színű (piros vagy zöld) lövedéket lő abba az irányba, amerre éppen néz.
- **pickup:** A Jaffa, ha még nincsen nála, felvesz egy dobozt az előtte lévő mezőről.
- **putdown:** A Jaffa, ha van nála doboz, leteszi azt az előtte lévő mezőre.

replicator

Leírás: A Replikátor állapotát írja ki a szabványos kimenetre (pozíció, szomszéd mezők típusai, stb...)

Opciók:

- **north / west / south / east:** A Replikátort a megadott irányba fordítja, vagy ha már abba az irányba néz, akkor el is mozgatja.
- **random:** A Replikátort addig mozgatja véletlenszerűen választott irányokba, amíg az szakadékba nem esik.

zpm_random

Leírás: Az új ZPM-ek keletkezésének

Opciók:

- **on / off:** A Replikátort a megadott irányba fordítja, vagy ha már abba az irányba néz, akkor el is mozgatja.

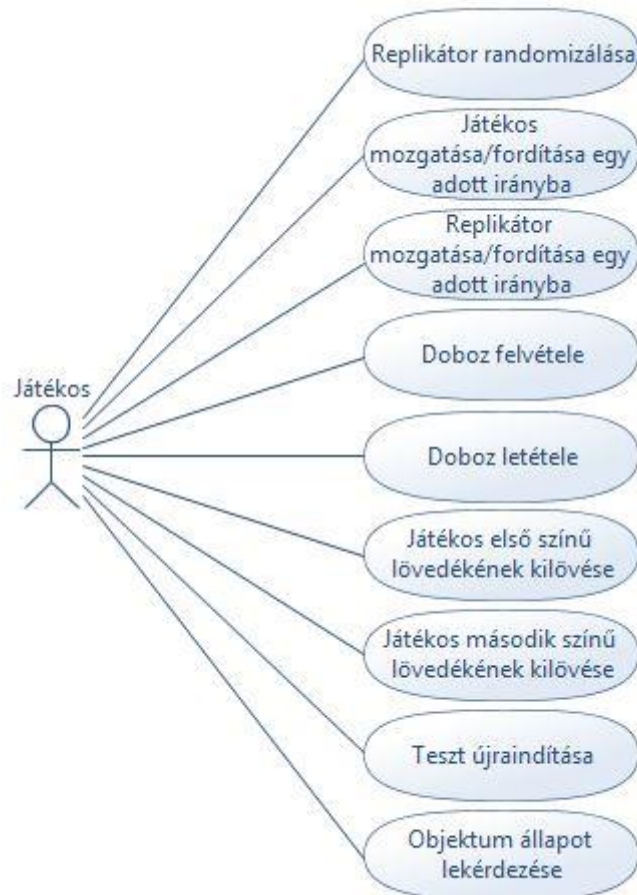
7.1.3 Kimeneti nyelv

Az egyes parancsok utáni kimenet három részre osztható: először minden kiadott parancs után kirajzolódik az egész térkép.

Ezalatt a kiadott parancs hatására bekövetkezett változások íródnak ki (pl. a kilőtt speciális lövedék speciális falnak ütközött és ott egy csillagkaput hozott létre).

A harmadik rész pedig a játék megkülönböztetett figyelemben részesített objektumairól (Ezredes, Jaffa, dobozok, ZPM-ek, stb...) nyújt információkat (aktuális pozíció, irány, van-e nála doboz, stb...).

7.2 Összes részletes use-case



Use-case neve	Replikátor randomizálása
Rövid leírás	Replikátor randomizálásának bekapcsolása
Aktorok	Játékos
Forgatókönyv	Bekapcsolt esetben a Replikátor véletlenszerűen mozog a pályán, amíg el nem pusztl.

Use-case neve	Játékos mozgatása/fordítása egy adott irányba
Rövid leírás	Játékos mozgatása/fordítása egy adott irányba
Aktorok	Játékos
Forgatókönyv	Ha a Játékos már abba az irányba néz, akkor a következő mezőre léptetése, egyébként pedig az adott irányba fordítása.

Use-case neve	Replikátor mozgatása/fordítása egy adott irányba
Rövid leírás	Replikátor mozgatása/fordítása egy adott irányba
Aktorok	Játékos
Forgatókönyv	Ha a Replikátor már abba az irányba néz, akkor a következő mezőre léptetése, egyébként pedig az adott irányba fordítása.

Use-case neve	Doboz felvétele
Rövid leírás	Doboz felvétele a következő mezőről
Aktorok	Játékos
Forgatókönyv	Ha a játékosnál nem található doboz, akkor a játékos előtti mezőről felveszi a dobozt.

Use-case neve	Doboz letétele
Rövid leírás	Doboz letétele a következő mezőre
Aktorok	Játékos
Forgatókönyv	Ha a játékosnál található doboz, akkor a játékos előtti mezőre leteszi a dobozt.

Use-case neve	Játékos első színű lövedékének kilövése
Rövid leírás	Speciális lövedék kilövése
Aktorok	Játékos
Forgatókönyv	A játékos irányának megfelelő irányban, a játékos első színű (sárga vagy piros) lövedékének lövése.

Use-case neve	Játékos második színű lövedékének kilövése
Rövid leírás	Speciális lövedék kilövése
Aktorok	Játékos
Forgatókönyv	A játékos irányának megfelelő irányban, a játékos második színű (kék vagy zöld) lövedékének lövése.

Use-case neve	Teszt újraindítása
Rövid leírás	Teszt előlről kezdése
Aktorok	Játékos
Forgatókönyv	Pálya betöltése, tárgyak visszahelyezése a kezdőpozíciójukra.

Use-case neve	Objektum állapot lekérdezése
Rövid leírás	Objektum adatainak lekérdezése
Aktorok	Játékos
Forgatókönyv	Egy objektumhoz tartozó információk lekérdezése

7.3 Tesztelési terv

Teszt-eset neve	Doboz felvétele útról
Rövid leírás	A játékos felvesz egy dobozt az előtte található út típusú mezőről.
Teszt célja	A teszt azt vizsgálja, hogy a játékos fel tud-e venni egy dobozt az előtte található út típusú mezőről.

Teszt-eset neve	Doboz felvétele mérlegről
Rövid leírás	A játékos felvesz egy dobozt az előtte található mérleg típusú mezőről.
Teszt célja	A teszt az vizsgálja, hogy a játékos fel tud-e venni egy dobozt az előtte található mérleg típusú mezőről, és a mérleghez tartozó ajtó bezárul-e mikor a megfelelő mennyiségű súly lekerül a mérlegről.

Teszt-eset neve	Doboz felvétele portálon átnyúlva
Rövid leírás	A játékos felvesz egy dobozt az előtte található nyitott portál másik végén található mezőről.
Teszt célja	A teszt az vizsgálja, hogy a játékos fel tud-e venni egy dobozt az előtte található nyitott portál másik végén található mezőről.

Teszt-eset neve	Doboz lerakása útra
Rövid leírás	A játékos letesz egy dobozt az előtte található út típusú mezőre.
Teszt célja	A teszt azt vizsgálja, hogy a játékos le tud-e tenni egy dobozt az előtte található út típusú mezőre.

Teszt-eset neve	Doboz lerakása mérlegre
Rövid leírás	A játékos letesz egy dobozt az előtte található mérleg típusú mezőre.
Teszt célja	A teszt azt vizsgálja, hogy a játékos le tud-e tenni egy dobozt az előtte található mérleg típusú mezőre és a mérleghez tartozó ajtó megfelelő súly hatására kinyílik-e.

Teszt-eset neve	Doboz lerakása szakadékba
Rövid leírás	A játékos letesz egy dobozt az előtte található szakadék típusú mezőre.
Teszt célja	A teszt azt vizsgálja, hogy a játékos le tud-e tenni egy dobozt az előtte található szakadék típusú mezőre és a doboz megsemmisül-e és megjelenik-e újból az eredeti helyén a pályán.

Teszt-eset neve	Doboz lerakása falra
Rövid leírás	A játékos megpróbál letenni egy dobozt az előtte található fal típusú mezőre. A falról a doboz "visszapattan" és a játékos mezőjére kerül.
Teszt célja	A teszt azt vizsgálja, hogy a játékos le tud-e tenni egy dobozt az előtte található fal típusú mezőre (nem tud).

Teszt-eset neve	Doboz lerakása portálon keresztül
Rövid leírás	A játékos letesz egy dobozt az előtte található nyitott portál másik végén található mezőre.
Teszt célja	A teszt azt vizsgálja, hogy a játékos le tud-e tenni egy dobozt az előtte található speciális falon lévő nyitott portálon keresztül, a portál másik végén található mezőre.

Teszt-eset neve	Játékos mozgatása útra
Rövid leírás	A játékos ellép az aktuális mezőjéről és az előtte található út típusú mezőre lép.
Teszt célja	A teszt az vizsgálja, hogy a játékos rá tud-e lépni az előtte található út típusú mezőre.

Teszt-eset neve	Játékos mozgatása mérlegre
Rövid leírás	A játékos ellép az aktuális mezőjéről és az előtte található mérleg típusú mezőre lép.
Teszt célja	A teszt azt vizsgálja, hogy a játékos rá tud-e lépni az előtte található mérleg típusú mezőre és a mérleghez tartozó ajtó kinyílik-e.

Teszt-eset neve	Játékos mozgatása falra
Rövid leírás	A játékos megpróbál ellépni az aktuális fal mezőjéről és az előtte található fal típusú mezőre lépni.
Teszt célja	A teszt azt vizsgálja, hogy a játékos rá tud-e lépni az előtte található fal típusú mezőre (nem tud).

Teszt-eset neve	Játékos mozgatása speciális falra (nyitott portál)
Rövid leírás	A játékos ellép az aktuális mezőjéről és az előtte található portálon keresztül a portál túoldalán található mezőre lép.
Teszt célja	A teszt az vizsgálja, hogy a játékos rá tud-e lépni az előtte található speciális falra, melyen nyitott portál található, át tud-e haladni a portálon keresztül és a portál túoldalán található mezőre lépni.

Teszt-eset neve	Játékos mozgatása szakadékba
Rövid leírás	A játékos ellép az aktuális mezőjéről és az előtte található szakadék típusú mezőre lép. A játékos meghal, a játék véget ér.
Teszt célja	A teszt azt vizsgálja, hogy a játékos rá tud-e lépni az előtte található szakadék típusú mezőre, ennek hatására a játékos meghal-e, és véget ér-e a játék.

Teszt-eset neve	Lövés falra
Rövid leírás	A játékos lő egyet, a lövedék fal típusú mezőn ér célba.
Teszt célja	A teszt azt vizsgálja, hogy mi történik egy lövedékkel, ha az fal típusú mezőn ér célba. Az elvárás az, hogy semmisüljön meg.

Teszt-eset neve	Lövés speciális falra
Rövid leírás	A játékos lő egyet, a lövedék speciális fal típusú mezőn ér célba és a lövedék színével megegyező színű csillagkaput nyit.
Teszt célja	A teszt azt vizsgálja, hogy mi történik egy lövedékkel, ha az speciális fal típusú mezőn ér célba. Az elvárás az, hogy a kilőtt lövedék színével megegyező csillagkaput hozzon létre.

Teszt-eset neve	Utolsó ZPM felvétele
Rövid leírás	A játékos ellép az aktuális mezőjéről, és az előtte lévő mezőre lép, melyen a pályán lévő utolsó ZPM modul található. A játékos felszedi a ZPM modult. A játék véget ér.
Teszt célja	A teszt az vizsgálja, hogy a játékos rá tud-e lépni az előtte található mezőre, melyen a pályán lévő utolsó ZPM modul található. A ZPM modult felszedve a játék véget ér-e.

Teszt-eset neve	Egyik játékos egy (nyitott állapotú) ajtó típusú mezőn áll, mikor az ajtó becsukódik
Rövid leírás	Egyik játékos egy (nyitott állapotú) ajtó típusú mezőn áll, mikor az ajtó becsukódik. Az ajtó megöli a játékost. A játék véget ér.
Teszt célja	A teszt azt vizsgálja, hogy az ajtó becsukódásakor az azon tartózkodó játékos elpusztul-e.

Teszt-eset neve	Új ZPM teremtdődik
Rövid leírás	Minden második ZPM után, amit az ezredes vesz fel, új ZPM teremtdődik a pálya egy véletlen mezőjén.
Teszt célja	A teszt azt vizsgálja, hogy minden páros sorszámú ZPM ezredes általi felvételekor létrejön-e egy új ZPM egy véletlenszerűen kiválasztott úton.

Teszt-eset neve	Lövedék eltalálja a replikátort
Rövid leírás	Lövedék eltalálja a Replikátort. A lövedék és a Replikátor is megsemmisül.
Teszt célja	A teszt azt vizsgálja, hogy mi történik, ha a Replikátor egy speciális lövedékkel konfrontálódik. Az elvárás az, hogy semmisüljön meg a lövedék is és a Replikátor is.

Teszt-eset neve	Több doboz letétele ugyan arra a mezőre
Rövid leírás	Egy mezőre több doboz is lerakható stack-szerűen.
Teszt célja	A teszt azt vizsgálja, hogy mi történik, ha a játékos olyan mezőre szeretne letenni egy dobozt, amin már van egy másik doboz. Az elvárt eredmény, hogy le tudja tenni a dobozt.

Teszt-eset neve	Replikátor szakadékba esik
Rövid leírás	A replikátor szakadékba esik, a replikátor megsemmisül, a szakadék típusú mező pedig út típusú mezővé változik.
Teszt célja	A teszt azt vizsgálja, hogy a replikátor rá tud-e lépni egy szakadék típusú mezőre, ilyenkor a szakadék típusú mező úttá változik-e.

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

Az egyes tesztekhez előre megadjuk a szükséges bemeneti fájlokat és az elvárt kimenetet. A tesztelés futása után az elvárt kimenet alapján a program leellenőrzi, hogy teljesül(nek)-e az adott teszthez tartozó sikerkritérium(ok) és a teszt eredményéről a szabványos kimeneten tájékoztatja a felhasználót a program.

Az ellenőrzés a program menüjéből külön is elérhető lesz, így nem kell minden alkalommal újra lefuttatni a tesztet.

8 Részletes tervek

8.1 Osztályok és metódusok tervei

8.1.1 Box

- **Felelősség**
A Dobozok kezelése. Tudnia kell a Doboz pozícióját és tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányba haladna tovább.
- **Ősosztályok**
Movable
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **-MapElement initPosition:** Ebben tárolja a doboz kezdeti pozícióját; megsemmisülés esetén erre a pozícióra kerül.
- **Metódusok**
 - **-void respawn():** A doboz elhagyja a MapElementet amin van. Létrejön a kezdeti pozícióján a doboz, majd megérkezik az adott MapElementre.
 - **-void arriveOnMapElement(Direction dir, MapElement element):** Meghívja a paraméterül kapott MapElement handleBoxPutDown függvényét.
 - **-void leaveMapElement(MapElement element):** A paraméterül kapott MapElement kezeli a doboz felvételét.

8.1.2 Controller

- **Felelősség**
Menedzseli a játék folyamatát és ellenőrzi annak állapotát.
- **Ősosztályok**
Nincs ősosztálya.
- **Interfészek**
 - IZPMPickedUpListener
 - IProjectileStateListener
- **Attribútumok**
 - **-Game game:** Az aktuális játékot reprezentáló játék példánya.
 - **-Jaffa jaffa:** Jaffa karakterét reprezentáló objektum referenciája.
 - **-Oneill oneill:** Az ezredes karakterét reprezentáló objektum referenciája.
 - **-HashSet<ZPM> zpmSet:** Pályán lévő ZPMek halmaza.

- **-boolean isProjectileMoving:** Az aktuális Projectile állapotának tárolására alkalmas.
- **-Replicator replicator:** Referencia a replikátort reprezentáló objektumra.
- **Metódusok**
 - **+void unsubscribe():** Leiratkozás A ZPMet és lövedéket kezelő listenerekről.
 - **+void moveOrTurnMovable(Moveable moveable, Direction dir):** A játékos/replikátor fordítása vagy mozgatása egy irányba a jelenlegi irány függvényében. Ha abba az irányba áll, amit paraméterül is kap, akkor a mozgást végző függvény hívódik meg; ellenkező esetben marad az aktuális MapElementen, és elfordul a kapott irányba.
 - **+void shootFirst(Player player):** A paraméterül kapott játékos első színű lövedékének kilövése.
 - **+void shootSecond(Player player):** A paraméterül kapott játékos második színű lövedékének kilövése.
 - **+void pickUpBox(Player player):** A játékos felvesz egy dobozt az előtte lévő mezőről.
 - **+void putDownBox(Player player):** A játékos letesz egy dobozt az előtte lévő mezőre.
 - **+void onZMPickedUp(ZPM zpm, Player player):** Játékos pontszámának növelése és a ZPM eltávolítása. Ha nincs több ZPM akkor a játéknak vége.
 - **+void onProjectileCreated(Projectile projectile):** Projectile állapotának beállítása mozgásra, majd amíg ez az állapot nem változik, addig mozog.
 - **+void onProjectileDestroyed(Projectile projectile):** Lövedék mozgásának leállítása.

8.1.3 Door

- **Felelősség**

Az ajtó típusú mező megvalósításáért felel. Mindegyik Doorhoz tartozik pontosan egy Scale, mely képes az ajtó mezőt átjárhatóvá tenni.
- **Össztályok**

MapElement
- **Interfészek**

Nem valósít meg interfészt.
- **Attribútumok**
 - **-boolean isOpened:** Az ajtó útként (true) vagy falként (false) viselkedik
 - **-List<Player> players:** Az ajtó mezőn álló játékosok listája.
- **Metódusok**
 - **+void handlePlayerArrive(Direction dir, Player player):** Ha a mező ajtóként viselkedik, akkor a játékos megérkezik és a mező kezeli ezt, ha falként, akkor a mező azt az esetet kezeli.
 - **+void handlePlayerLeave(Player player):** Ez a függvény kezeli le a játékos távozását az ajtó mezőről.

- **+void handleProjectileArrive(Direction dir, Projectile projectile):** Ez a függvény kezeli le a speciális lövedék érkezését az ajtó mezőre. Ha volt itt egy Replikátor, akkor az elpusztul. Ha az ajtó zárva van akkor a Projectile megsemmisül.
- **+void handleBoxPutDown(Direction dir, Box box):** Ez a függvény kezeli le egy doboz lerakását az ajtó mezőre. Ha az ajtó nyitva van, akkor erre a mezőre kerül, ha zárva, akkor oda, ahonnan az ajtó mezőre tettük volna.
- **+void handleBoxPickUp():** Ez a függvény kezeli le egy doboz felvételét az ajtó mezőről. A doboz pozícióját kinullázzuk.
- **+void setOpen(booleen isOpened):** Ez a függvény állítja be, hogy az ajtó nyitva vagy csukva van. Ha egy játékos áll az ajtóban és az ajtó bezárul, az meghal. Ha Replikátor van rajta és az ajtó bezárul, az elpusztul. Ha doboz van rajta, és az ajtó bezárul, akkor a doboz átkerül a kezdeti pozíciójára.
- **+void handleReplicatorArrive(Direction dir, Replicator rep):** Ez a függvény kezeli le a replikátor érkezését a mezőre.
- **+void handleReplicatorLeave():** Ez a függvény kezeli le a replikátor távozását a mezőről.

8.1.4 Game

- **Felelősség**
Az osztály felelőssége a játék elindítása, leállítása és az összegyűjtött ZPM-ek számontartása.
- **Összostályok**
Nincs őssosztálya.
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **-int scoreJaffa:** Jaffa által összegyűjtött ZPM-ek száma.
 - **-int scoreOneill:** Oneill által összegyűjtött ZPM-ek száma.
- **Metódusok**
 - **+void run():** Ez a főciklus, melyben maga a játék működik. Ez hozza létre a MapBuildert és a Controllert, előbbi konstruktor paraméterben kapja meg a térképet tároló fájl elérési útvonalát.
 - **+int getScore(Player player):** Visszaadja, hogy mennyi pontot gyűjtött össze a játékos.
 - **+void incrementScore(Player player):** Megnöveli a játékos által összegyűjtött pontok értékét eggyel.
 - **+void stop():** A játék vége. Ha úgy lett vége a játéknak, hogy nincs több ZPM a pályán, a több pontot összegyűjtött játékos nyer. Ha úgy lett vége, hogy valamelyik játékos meghalt, akkor az életben maradt játékos nyer.

8.1.5 Gap

- **Felelősség**
A szakadék mező megvalósításáért felelős. Ha a játékos vagy egy doboz érkezik rá, akkor azt megsemmisíti. Ha a projectile érkezik rá, akkor út mezővé változik
- **Ősosztályok**
MapElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Az őosztálytól örökölt attribútumokon kívül mást nem definiál.
- **Metódusok**
 - **+void handlePlayerArrive(Direction dir, Player player):** Ez a függvény kezeli le a játékos érkezését a szakadék mezőre, melynek során meghal.
 - **+void handleProjectileArrive(Direction dir, Projectile projectile):** Ez a függvény kezeli le a speciális lövedék érkezését a szakadék mezőre, melynek során gond nélkül áthalad rajta.
 - **+void handleBoxPutDown(Direction dir, Box box):** Ez a függvény kezeli le egy doboz lerakását a szakadék mezőre, melynek során a doboz átkerül a kezdeti pozíciójára.
 - **+void handleReplicatorArrive(Direction dir, Replicator rep):** Ez a függvény kezeli le a replikátor érkezését a mezőre. A Replikátor elpusztul és a szakadék helyén út jelenik meg.

8.1.6 Jaffa

- **Felelősség**
Jaffa kezelése. Tudnia kell Jaffa pozícióját, tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább.
- **Ősosztályok**
Moveable → *Player*
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **-Direction direction:** Az irány, amerre a Jaffa néz.
 - **-Set<PortalType> projType:** A kilőhető speciális lövedékek színe.
 - **-Box box:** Jaffánál lévő doboz referenciája (ha van nála).
 - **-boolean isAlive:** Igaz, ha még él a Jaffa.

- **Metódusok**

- **+void pickUpBox():** Ez a függvény kezeli a doboz felvételt. Ha van nála doboz, akkor semmi nem történik. Ha nincs, akkor arról a MapElementről próbál fölvenni egy dobozt, amerre néz. Ha ott van legalább egy doboz, akkor azt fölveszi, ezután a doboz elhagyja a mezőt. Ezután megnöveljük a tömegünket.
- **+void putDownBox():** Ez a függvény kezeli a doboz letevését. Ha nincs nálunk, akkor nem történik semmi. Ha van akkor azt a következő mezőre tesszük.
- **+void turn(Direction direction):** Ez a függvény kezeli Jaffa irányváltását; a paraméterül kapott irányba fog nézni.
- **+void arriveOnMapElement(Direction direction, MapElement mapElement):** Ez a függvény kezeli le a Jaffa egy mezőre való érkezését, melyet a következő MapElement old meg.
- **+void leaveMapElement(MapElement mapElement):** Ez a függvény kezeli le a Jaffa egy mezőről való távozását, melyet a jelenlegi MapElement old meg.
- **+void die():** Az isAlive attribútumot hamisra állítja.
- **+boolean isAlive():** Getter az isAlive attribútumra.
- **+void move():** Jaffa mozgatása. A jelenlegi mezőt elhagyja és megérkezik a következőre. Ezeket az adott MapElementek kezelik.
- **+void shootFirst():** Lövés az egyik típusú lövedékkel, melynek során létrejön egy Projectile.
- **+void shootSecond():** Lövés a másik típusú lövedékkel, melynek során létrejön egy Projectile.
- **+int getWeight():** Visszatér Jaffa súlyával, ami 2 egységnyi. Ha van nálunk doboz, akkor a doboz 1 egységnyi súlya miatt 3 egységgel térünk vissza.

8.1.7 MapBuilder

- **Felelősség**

A pálya beolvasása, a megfelelő MapElement példányok létrehozása és a szomszédos pályaelemek összekapcsolása.

- **Össztályok**

Nincs osztály

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **-final String mapPath:** A pálya felépítését tartalmazó fájl elérését tárolja

- **Metódusok**

- **+Controller createController(Game game):** Controller létrehozása.
- **+void buildMap():** Létrehozza és felépíti a fájlból beolvasott térképet, az elérési útvonal valid. Ennek során először a fájlt feldaraboljuk és az egyes MapElementeket egy átmeneti mátrixban tároljuk (egyelőre stringként). Ezután a mátrix minden elemén végig megyünk, és kialakítjuk a kapcsolatokat az újonnan létrehozott MapElementek között (szomszédok, ajtó-mérleg összerendelés).

Végül elhelyezzük a pályán a játékosokat, a Replikátort, a dobozokat és a ZPM-eket.

8.1.8 MapElement

- **Felelősség**

A Mező típusokat egybefonó őszosztály. Szerepe, hogy a konkrét mezőket a dinamikus típusuk ismerete nélkül tudjuk heterogén kollekcióként kezelni.

- **Őszosztályok**

Nincs őszosztálya.

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **-Stack<Box> box:** A mezőn elhelyezkedő dobozok tárolását végzi.
- **-HashMap<Direction, MapElement> neighbours:** A Mező szomszédait tárolja.
- **-Replicator replicator:** Mutató a mezőn elhelyezkedő replikátorra.

- **Metódusok**

- **+MapElement getNeighbour(Direction dir):** Visszaadja a mező megfelelő oldali szomszédját. Paraméterül kapja a megfelelő oldali szomszéd irányát.
- **+void setNeighbour(Direction dir, MapElement neighbour):** Beállítja a mező megfelelő oldali szomszédját. Paraméterül kapja a szomszédos mezőt, és annak irányát az aktuális mezőhöz képest.
- **+Box getBox():** Ha a mezőn található doboz, akkor visszaadja azt.
- **+void handlePlayerArrive(Direction dir, Player player):** Paraméterül megkapja, hogy melyik játékos, milyen irányból érkezik a mezőre. Ez a függvény kezeli egy játékos érkezését az adott mezőre. Default viselkedése: visszalöki a játékost oda, ahonnan az érkezett (a megfelelő szomszéd megállapítása után a szomszéd kezeli a játékos érkezését).
- **+void handlePlayerLeave():** Ez a függvény kezeli le a játékos távozását az adott mezőről. Default viselkedése: nem csinál semmit.
- **+void handleProjectileArrive(Direction dir, Projectile projectile):** Ez a függvény kezeli le a speciális lövedék érkezését az adott mezőre. Paraméterül kapja magát a lövedéket, és az irányt, ahonnan a lövedék érkezik a mezőre. Default viselkedése: nem csinál semmit.
- **+void handleBoxPutDown(Direction dir, Box box):** Ez a függvény kezeli le egy doboz lerakását az adott mezőre. Paraméterül kapja az irányt, ahonnan a doboz érkezik, és magát a dobozt. Default viselkedése: arra a mezőre teszi a dobozt, ahonnan az érkezett (a megfelelő szomszéd megállapítása után a szomszéd kezeli a doboz érkezését).
- **+void handleBoxPickUp():** Ez a függvény kezeli le egy doboz felvételét az adott mezőről. Default viselkedése: nem csinál semmit.
- **+void incrementWeight():** A mezőn elhelyezkedő súlyt növeli. Default viselkedése: nem csinál semmit.

- **+void decrementWeight():** A mezőn elhelyezkedő súlyt csökkenti. Default viselkedése: nem csinál semmit.
- **+void handleReplicatorArrive(Direction dir, Replicator rep):** Ez a függvény kezeli le a replikátor érkezését a mezőre. Paraméterül kapja az irányt, ahonnan a replikátor érkezik, valamint magát a replikátort. Default viselkedése: visszalöki a replikátort arra a mezőre, ahonnan érkezett (a megfelelő szomszéd megállapítása után a szomszéd kezeli a replikátor érkezését).
- **+void handleReplicatorLeave():** Ez a függvény kezeli le a replikátor távozását a mezőről. Default viselkedése: nem csinál semmit.

8.1.9 MapHelper

- **Felelősség**
A pálya átépítésében nyújt segítséget.
- **Ősosztályok**
Nincs őszosztálya.
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Nincs attribútuma.
- **Metódusok**
 - **+void replaceWithRoad(Gap gap):** A megadott hoz egy új típusú mezőt a szakadék helyén. Frissíti a szomszédokat.
 - **+Road getRandomRoad():** Visszatér egy véletlenszerűen választott úttal a pályáról.

8.1.10 ModelEventSource

- **Felelősség**
A modell osztályai által generált események megfigyelőinek kezelése, illetve ezek értesítése az adott eseményről.
- **Ősosztályok**
Nincs őszosztálya.
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **-static Set<IZPMPickedUpListener> zpmListeners:** A feliratkozott ZPM listenerek halmaza.
 - **-static Set<IProjectileStateListener> projectileListeners:** A feliratkozott lövedék listenerek halmaza.
- **Metódusok**
 - **+static void subscribe(IZPMPickedUpListener listener):** Új ZPM listener felvétele.
 - **+static void subscribe(IProjectileStateListener listener):** Új lövedék listener felvétele.
 - **+static void unsubscribe(IZPMPickedUpListener listener):** ZPM listener leiratkozását támogató metódus.
 - **+static void unsubscribe(IProjectileStateListener listener):** Lövedék listener leiratkozását támogató metódus.
 - **+static void clear() :** Minden feliratkozó törlése

- **+static void notifyZMPickedUp(ZPM zpm):** A feliratkozók értesítése egy ZPM felvételéről.
- **+static void notifyProjectileCreated(Projectile projectile):** A feliratkozók értesítése lövedék létrejöveteléről.
- **+static void notifyProjectileDestroyed(Projectile projectile):** A feliratkozók értesítése egy lövedék megsemmisüléséről.

8.1.11 Movable

- **Felelősség**
A mozgatható objektumokat egybefonó absztrakt őssosztály. Szerepe, hogy a mozgatható objektumokat egységesen tudjuk kezelni.
- **Őssosztályok**
Nincs őssosztálya.
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **#MapElement position:** Ebben tárolja az objektum a pozícióját.
- **Metódusok**
 - **+void arriveOnMapElement(Direction dir, MapElement element):** Ez a függvény kezeli le egy objektum mezőre való érkezését. A konkrét működést a leszármazottjai valósítják meg.
 - **+void leaveMapElement(MapElement element):** Ez a függvény kezeli le egy objektum mezőről való távozását. A konkrét űködést a leszármazottjai valósítják meg.
 - **+void move():** Ez a függvény kezeli le egy objektum mozgását egy adott koordinátára. A konkrét űködést a leszármazottjai valósítják meg.
 - **+void setPosition(MapElement position):** Az objektum pozícióját állítja be. A paraméterül kapott mező pozícióját állítja be saját pozíciójául.

8.1.12 ONeill

- **Felelősség**
ONeill kezelése. Tudnia kell ONeill pozícióját, tudnia kell lekezeln, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább.
- **Őssosztályok**
Moveable → Player
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **-Direction direction:** Az irány, amerre a ONeill néz.

- **-Set<PortalType> projType:** A kilőhető speciális lövedékek színe.
- **-Box box:** ONeillnél lévő doboz referenciája (ha van nála).
- **-boolean isAlive:** Igaz, ha még él a ONeill.
- **Metódusok**
 - **+void pickUpBox():** Ez a függvény kezeli a doboz felvételt. Ha van nála doboz, akkor semmi nem történik. Ha nincs, akkor arról a MapElementről próbál fölvenni egy dobozt, amerre néz. Ha ott van legalább egy doboz, akkor azt fölveszi, ezután a doboz elhagyja a mezőt. Ezután megnöveljük a tömegünket.
 - **+void putDownBox():** Ez a függvény kezeli a doboz letevését. Ha nincs nálunk, akkor nem történik semmi. Ha van akkor azt a következő mezőre tesszük.
 - **+void turn(Direction direction):** Ez a függvény kezeli ONeill irányváltását; a paraméterül kapott irányba fogunk nézni
 - **+void arriveOnMapElement(Direction direction, MapElement mapElement):** Ez a függvény kezeli le a ONeill egy mezőre való érkezését, melyet a következő MapElement old meg.
 - **+void leaveMapElement(MapElement mapElement):** Ez a függvény kezeli le a ONeill egy mezőről való távozását, melyet a jelenlegi MapElement old meg.
 - **+void die():** Az isAlive attribútumot hamisra állítja.
 - **+boolean isAlive():** Getter az isAlive attribútumra.
 - **+void move():** ONeill mozgatása. A jelenlegi mezőt elhagyja és megérkezik a következőre. Ezeket az adott MapElementek kezelik.
 - **+void shootFirst():** Lövés az egyik típusú lövedékkal, melynek során létrejön egy Projectile.
 - **+void shootSecond():** Lövés a másik típusú lövedékkal, melynek során létrejön egy Projectile
 - **+int getWeight():** Visszatér ONeill súlyával, , ami 2 egységnyi. Ha van nálunk doboz, akkor a doboz 1 egységnyi súlya miatt 3 egységgel térünk vissza.

8.1.13 Player

- **Felelősség**

A játékos kezelése. Tudnia kell a játékos pozícióját, tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább.
- **Ősosztályok**

Movable
- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **-Direction direction:** Az irány, amerre a játékos néz. Alap beállítás szerint a játékos NORTH (észak) irányba néz.
- **-Set<ProjectileType> projType:** A kilőhető speciális lövedékek színe.
- **-Box box:** A játékosnál lévő doboz referenciája (ha van nála).
- **-boolean isAlive:** Igaz, ha még él a játékos, hamis ha már nem.

- **Metódusok**

- **+void pickUpBox():** Ez a függvény kezeli a doboz felvételét a játékos előtt található mezőről, amennyiben a játékosnál még nincs doboz. Ha a játékos előtt található mezőn van doboz, akkor lekezeli, hogy a doboz elhagyja a mezőt és meghívja a mezőre az incrementWeight() metódust, ami mezőtípusonként különbözően viselkedik.
- **+void putDownBox():** Ez a függvény kezeli a doboz letevését a játékos előtt található mezőre, amennyiben a játékosnál van doboz. Meghívja a mezőre a decrementWeight() metódust, ami mezőtípusonként különbözően viselkedik, majd a játékosnál lévő doboz referenciáját nullára állítja.
- **+void turn(Direction direction):** Ez a függvény kezeli a játékos irányváltását. A játékos új iránya, a paraméterül kapott irány lesz.
- **+void arriveOnMapElement(Direction direction, MapElement mapElement):** Ez a függvény kezeli le a játékos mezőre való érkezését. A paraméterül kapott mezőre, a paraméterül kapott irányból érkezik a játékos.
- **+void leaveMapElement(MapElement mapElement):** Ez a függvény kezeli le a játékos mezőről való távozását. A paraméterül kapott mezőről távozik a játékos.
- **+void die():** Az isAlive attribútumot hamisra állítja.
- **+boolean isAlive():** Getter az isAlive attribútumra.
- **+void move():** Játékos mozgatása. Megállapítja a célmezőt, amire a játékos lépni szeretne, ez mindig a játékos előtt található mező. Ha a játékos előtt nincs mező, akkor a lépés nem hajtodik végre. Ha a játékos előtti mező érvényes mező, akkor a játékos elhagyja az aktuális mezőjét, és a célmező kezeli a játékos megérkezését.
- **+void shootFirst():** Lövés az egyik típusú lövedékkel. Létrehoz egy lövedéket az egyik lövedéktípusból. A létrehozott lövedéknek van pozíciója, iránya és típusa (színe). Pozíciója a játékos pozíciója, iránya a játékos iránya.
- **+void shootSecond():** Lövés a másik típusú lövedékkel. Létrehoz egy lövedéket a másik lövedéktípusból. A létrehozott lövedéknek van pozíciója, iránya és típusa (színe). Pozíciója a játékos pozíciója, iránya a játékos iránya.
- **+int getWeight():** Visszatér a játékos súlyával. Ha van a játékosnál doboz, akkor a játékos súlya a doboz súlyával együtt értendő.

8.1.14 Projectile

- **Felelősség**

A Speciális lövedékek kezelése. Tudnia kell a Speciális lövedék pozícióját, tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább. Speciális fallal történő ütközés esetén létrehoz egy Stargate objektumot.

- **Össztályok**

Movable

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **-Direction direction:** Ebben tárolja a lövedék irányát.
- **-ProjectileType type:** Ebben tárolja a lövedék típusát.

- **Metódusok**

- **+Stargate openStargate():** Ez a függvény kezeli a lövedék speciális fallal való találkozásakor keletkező csillagkapu nyitását. A metódus megállapítja a lövés irányával ellentétes irányt és (amikor a lövedék speciális fallal találkozik), a speciális fal lövés irányával ellentétes irányú oldalán létrehoz egy csillagkaput.
- **+void arriveOnMapElement(Direction dir, MapElement element):** Ez a függvény kezeli le a lövedék mezőre való érkezését. paraméterül kapott mezőre, a paraméterül kapott irányból érkezik a lövedék.
- **+void leaveMapElement(MapElement element):** Ez a függvény kezeli le a lövedék mezőről való távozását. A paraméterül kapott mezőről távozik a lövedék, amennyiben van még előtte mező. Ha már nincs előtte mező, a lövedék megsemmisül.
- **+void move():** A lövedék mozgását megvalósító függvény. Elhagyja a mezőt, amin aktuálisan tartózkodik, és az előtte lévő szomszédos mezőre lép. A lövedék érkezését a szomszédos mező kezeli le. Ha már nincs előtte több mező, a lövedék megsemmisül.

8.1.15 Replicator

- **Felelősség**

A replikátor kezelése. Tudnia kell a pozícióját, tudnia kell lekezelni, hogy mi történik vele, ha egy mezőre érkezik, ha egy mezőről távozik, ha egy adott irányban haladna tovább.

- **Ősosztályok**
Movable
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **-Direction direction:**A replikátor iránya. Alapértelmezetten a replikátor NORTH (észak) irányba néz.
- **Metódusok**
 - **+void arriveOnMapElement(Direction direction, MapElement mapElement):** Ez a függvény kezeli le a replikátor egy mezőre való érkezését. A paraméterül kapott mezőre érkezik a paraméterül kapott irányból.
 - **+void leaveMapElement(MapElement mapElement):** Ez a függvény kezeli le a replikátor egy mezőről való távozását. A paraméterül kapott mezőt hagyja a replikátor.
 - **+void move():** Replikátor mozgatása. Elhagyja a mezőt, amin aktuálisan tartózkodik, és az előtte lévő szomszédos mezőre lép. A replikátor érkezését a szomszédos mező kezeli le.
 - **+void destroy():** Megsemmisíti a replikátort.
 - **+void turn(Direction dir):** Replikátor forgatása a megadott irányba. A replikátor új iránya a paraméterül kapott irány lesz.

8.1.16 Road

- **Felelősség**
Az út típusú mezők megvalósításáért felelős.
- **Ősosztályok**
MapElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **-ZPM zpm:** A Mezőn elhelyezkedő ZPM-re mutató referencia.
- **Metódusok**
 - **+void handlePlayerArrive(Direction dir, Player player):**Ez a függvény kezeli le a játékos érkezését az út mezőre. Beállítja a játékos pozícióját a saját pozíciójára. Ha van az aktuális pozíciójú út típusú mezőn ZPM, akkor azt felszedi a játékos.
 - **+void handlePlayerLeave():**Ez a függvény kezeli le a játékos távozását az út mezőről. Ugyan úgy működik, mint az ősosztályának handlePlayerLeave() meódusa.

- **+void handleProjectileArrive(Direction dir, Projectile projectile):** Ez a függvény kezeli le a speciális lövedék érkezését az út mezőre. Beállítja a lövedék pozícióját a saját pozíciójára. Ha az aktuális pozíciójú út típusú mezőn perlikátor tartózkodik, akkor megsemmisíti azt és maga a lövedék is megsemmisül.
- **+void handleBoxPutDown(Direction dir, Box box):** Ez a függvény kezeli le egy doboz lerakását az út mezőre. Beállítja a doboz pozícióját a saját pozíciójára. A dobozt behelyezi a Stack-be, amiben az aktuális mezőn elhelyezkedő dobozokat tároljuk.
- **+void handleBoxPickUp():** Ez a függvény kezeli le egy doboz felvételét az út mezőről, amennyiben a mezőn található doboz. A doboz pozícióját nullára állítja.
- **+void handleReplicatorArrive(Direction dir, Replicator rep):** Ez a függvény kezeli le a replikátor érkezését a mezőre. Beállítja a replikátor pozícióját a saját pozíciójára. A referenciát, amely a mezőn elhelyezkedő replikátorra mutat, a paraméterül kapott replikátorral rendeli össze.
- **+void handleReplicatorLeave():** Ez a függvény kezeli le a replikátor távozását a mezőről. A referenciát, amely a mezőn elhelyezkedő replikátorra mutat, nullára állítja.

8.1.17 Scale

- **Felelősség**

A mérleg típusú mezők megvalósításáért felel. Mindenképpen egy ajtóhoz tartozik. Amennyiben rákerül a játékos vagy egy doboz, akkor az ajtót átjárhatóvá teszi.

- **Össztályok**

MapElement

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **-int weightCount:** A mezőn elhelyezkedő objektumok összsúlya.
- **-int weightLimit:** Azt tartjuk számon benne, hogy mekkora súly fölött kell kinyílnia a mérleghez tartozó ajtónak.
- **-Door door:** A Mérleghez tartozó ajtóra mutató referencia.

- **Metódusok**

- **+void handlePlayerArrive(Direction dir, Player player):** Ez a függvény kezeli le egy játékos érkezését a mérleg mezőre. Beállítja a játékos pozícióját a saját pozíciójára. Lekéri a játékos súlyát, és a weightCount változóját megnöveli a játékos súlyával. Amennyiben a mérlegen található objektumok összsúlya eléri a mérleghez tartozó súlykorlátot, akkor a mérleghez tartozó ajtó kinyílik.
- **+void handlePlayerLeave(Player player):** Ez a függvény kezeli le egy játékos távozását a mérlegmezőről. Lekéri a játékos súlyát és a weightCount változóját csökkenti a játékos súlyával. Amennyiben a mérlegen található objektumok

összsúlya a mérleghez tartozó súlykorlát alá csökken, akkor a mérleghez tartozó ajtó bezárul.

- **+void handleProjectileArrive(Direction dir, Projectile projectile):** Ez a függvény kezeli le a speciális lövedék érkezését a mérleg mezőre. Beállítja a lövedék pozícióját a saját pozíciójára. Ha az aktuális pozíciójú mérleg típusú mezőn perlikátor tartózkodik, akkor megsemmisíti azt.
- **+void handleBoxPutDown(Direction dir, Box box):** Ez a függvény kezeli le egy doboz lerakását a mérleg mezőre. Beállítja a doboz pozícióját a saját pozíciójára. A dobozt behelyezi a Stack-be, amiben az aktuális mezőn elhelyezkedő dobozokat tároljuk. És meghívja a mezőn elhelyezkedő objektumok összsúlyát számontartó attribútumot (weightCount) növelő függvényt (incrementWeight()).
- **+void handleBoxPickUp():** Ez a függvény kezeli le egy doboz felvételét a mérleg mezőről, amennyiben található doboz az adott mezőn. Beállítja a doboz pozícióját nullára. És meghívja a mezőn elhelyezkedő objektumok összsúlyát számontartó attribútumot (weightCount) csökkentő függvényt (decrementWeight()).
- **+void incrementWeight():** A mérlegen elhelyezkedő súlyt növeli eggyel. Ha az ajtó még nincs nyitva, akkor kinyitja a mérleghez tartozó ajtót, amennyiben a mérlegen található objektumok összsúlya nagyobb, mint a súlykorlát.
- **+void decrementWeight():** A mérlegen elhelyezkedő súlyt csökkenti eggyel. A mérlegen található objektumok összsúlya nem lehet negatív. Ha mérleghez tartozó ajtó még nincs zárva és a súlykorlátnál kevesebb összsúly található a mérlegen, akkor a megfelelő ajtó bezárul.
- **+void handleReplicatorArrive(Direction dir, Replicator rep):** Ez a függvény kezeli le a replikátor érkezését a mezőre. Beállítja a replikátor pozícióját a saját pozíciójára. A referenciát, amely a mezőn elhelyezkedő replikátorra mutat, a paraméterül kapott replikátorral rendeli össze.
- **+void handleReplicatorLeave():** Ez a függvény kezeli le a replikátor távozását a mezőről. A referenciát, amely a mezőn elhelyezkedő replikátorra mutat, nullára állítja.

8.1.18 SpecialWall

- **Felelősség**
Felelőssége olyan fal megvalósítása, mely alapesetben átjárhatatlan, de ha Projectile éri, akkor egy Stargate nyílik rajta, így szomszédos mezőnek tekinthető azzal a MapElementtel, mely szintén egy SpecialWall és van rajta egy másik (párjának megfelelő) Stargate.
- **Ősosztályok**
MapElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **-HashMap<Direction, Stargate> stargate:** A Mezőn elhelyezkedő csillagkapura egy referencia relatív pozíciójuk alapján.
- **Metódusok**
 - **+void handlePlayerArrive(Direction dir, Player player):** Ez a függvény kezeli le egy játkos érkezését a speciális fal mezőre. Ennek során, ha abba az irányba van nyitva csillagkapu, akkor a játékos átkerül a másik csillagkapu melletti mezőre, iránya pedig a kijáratirányával megegyező lesz. Ha nincs nyitva a speciális falon csillagkapu, akkor a játékos visszakerül az előző mezőre.
 - **+void handleProjectileArrive(Direction dir, Projectile projectile):** Ez a függvény kezeli le a speciális lövedék érkezését a speciális fal mezőre. Ha már van nyitott csillagkapu azon az oldalon a falnak, ahova a lövedék érkezik, akkor a lövedék megsemmisül, ellenkező esetben a speciális fal azon oldalán, ahova a lövedék érkezett egy csillagkapu nyílik, és utána a lövedék megsemmisül.
 - **+void handleBoxPutDown(Direction dir, Box box):** Ez a függvény kezeli le egy doboz lerakását a speciális fal mezőre. Ha van nyitva csillagkapu abból az irányból ahonnan a doboz érkezik, akkor átkerül a kijárat melletti mezőre. Ha nincs csillagkapu, akkor visszakerül arra a mezőre ahol épp állunk.
 - **+void handleBoxPickUp():** Ha van nyitva csillagkapu, akkor a féregjárat túloldalán lévő dobozt felvesszünk magunkhoz.
 - **+void handleReplicatorArrive(Direction dir, Replicator rep):** Ez a függvény kezeli le a replikátor érkezését a mezőre, mely nyitott csillagkapu át tud mozogni.

8.1.19 Stargate

- **Felelősség**
Ha Projectile egy SpecialWallba ütközik, jön létre a Stargate. Két Stargate közötti kapcsolatért felelős ez az osztály.
- **Ősosztályok**
Nincs ősosztálya
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **-SpecialWall position:** A csillagkapu koordinátájának tárolása.
 - **-Direction direction:** A csillagkapu irányát tároljuk benne.
 - **-ProjectileType type:** A csillagkapu típusát tároljuk benne.
 - **-Stargate blueStargate:** A kék csillagkapu
 - **-Stargate yellowStargate:** A sárga csillagkapu
 - **-Stargate redStargate:** A piros csillagkapu
 - **-Stargate greenStargate:** A zöld csillagkapu
- **Metódusok**
 - **+MapElement getExitPosition():** A függvény visszaadja a féregjárat ellenkező oldali kijáratának pozícióját, amennyiben létezik ilyen.
 - **+Direction getExitDirection():** A függvény visszaadja a féregjárat ellenkező oldali kijáratának irányát, amennyiben létezik ilyen.
 - **+Stargate createStargate(MapElement e, ProjectileType p, Direction d):**
Elkészíti a megfelelő csillagkaput a paraméterek szerint. Létrehoz egy csillagkaput a paraméterül kapott pozíción, a paraméterül kapott iránnyal és lövedék típussal.

8.1.20 Wall

- **Felelősség**
A fal típusú mezők megvalósításáért felelős. Nem átjárható. Ezekre a mezőkre nem kerülhet semmilyen Movable objektum.
- **Ősosztályok**
MapElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Az őszosztálytól örökölt attribútumokon kívül mást nem definiál.
- **Metódusok**
 - **+void handleProjectileArrive(Direction dir, Projectile projectile):** Ez a függvény kezeli le a speciális lövedék érkezését a fal mezőre. Beállítja a lövedék pozícióját a saját pozíciójára, majd megsemmisíti a lövedéket.

8.1.21 ZPM

- **Felelősség**
A játék célja, hogy az összes ilyen objektumot felszedjük.
- **Ősosztályok**
Nincs őszosztálya.
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Nincsenek attribútumai
- **Metódusok**
 - **+void handlePickUp(Player p):** Ez a függvény kezeli le, ha egy játékos rálép egy olyan mezőre, ahol ZPM van, melynek során értesíti erről a feliratkozót.

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

8.2.1 Játékos mozgatása útra

- **Leírás**

A játékos ellép az aktuális mezőjéről és az előtte található út típusú mezőre lép.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt az vizsgálja, hogy a játékos rá tud-e lépni az előtte található út típusú mezőre. A mező felelős a saját szomszédaival való kommunikációért és a játékos pozíciójának beállításáért. Hiba esetén a játékos pozíciója nem fog megváltozni.

- **Bemenet**

Pálya

R_O, R

Parancsok

```
oneill east  
oneill east
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba
ONeill a(z) A1 (Road) mezőről a(z) A2 (Road) mezőre próbál lépni
Sikeresen átlépett a következő mezőre: A2 (Road)

8.2.2 Játékos mozgatása mérlegre

- **Leírás**

A játékos ellép az aktuális mezőjéről és az előtte található mérleg típusú mezőre lép.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy a játékos rá tud-e lépni az előtte található mérleg típusú mezőre és a mérleghez tartozó ajtó kinyílik-e. A mérlegnek ellenőriznie kell, hogy a ráérkező játékos súlya meghaladja-e a saját súlykorlátját, majd ha ez teljesül, akkor szólania kell a hozzá tartozó ajtónak. Az ajtónak ezután ki kell nyílnia.

- **Bemenet**

Pálya

R_O, S:1, D:1

Parancsok

```
oneill east  
oneill east
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba

ONeill a(z) A1 (Road) mezőről a(z) A2 (Scale) mezőre próbál lépni

Sikeresen átlépett a következő mezőre: A2 (Scale)

A(z) A2 (Scale) mezőre 2 egységnyi súly került, összesen 2 súly van rajta

Mivel legalább 2 súly van a mérlegen, a(z) A3 (Door) kinyílt

8.2.3 Játékos mozgatása falra

- **Leírás**

A játékos ellép az aktuális mezőjéről és az előtte található fal típusú mezőre lép.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy a játékos rá tud-e lépni az előtte található fal típusú mezőre (nem tud). A fal, ahogy értesül arról, hogy egy játékos megpróbál rálépni, vissza kell dobnia a játékost arra a mezőre, ahonnan érkezett.

- **Bemenet**

Pálya

```
R_O, W
```

Parancsok

```
oneill east  
oneill east
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba

ONeill a(z) A1 (Road) mezőről a(z) A2 (Wall) mezőre próbál lépni

Sikertelen lépés: ONeill visszalöködött az előző A1 (Road) mezőre

Sikeresen átlépett a következő mezőre: A1 (Road)

8.2.4 Játékos mozgatása speciális falra (nyitott portál)

- **Leírás**

A játékos nyit két különböző színű csillagkaput, majd az egyikre rálépve a másikon kijön.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy a játékos át tud-e haladni a nyitott féregjáraton. A speciális fal, amikor arról értesül, hogy egy játékos rá akar lépni, megvizsgálja, hogy a megfelelő élen

van-e nyitott csillagkapu és ha igen, akkor létezik-e neki párja. Ha ezek a feltételek teljesülnek, akkor a rálépő játékost áthelyezi a másik csillagkapu elé, egyébként falként viselkedik (tehát visszadobja a játékost arra a mezőre, ahonnan érkezett).

- **Bemenet**

Pálya

SW, R, R_O, SW

Parancsok

```
oneill west  
oneill proj1  
oneill east  
oneill proj2  
oneill east
```

- **Elvárt kimenet**

ONeill elfordult WEST irányba

ONeill YELLOW lövedéket lőtt WEST irányba

YELLOW csillagkaput nyitott A1 (SpecialWall) mező EAST oldalán

YELLOW lövedék megsemmisült a(z) A1 (SpecialWall) mezőn

ONeill elfordult EAST irányba

ONeill BLUE lövedéket lőtt EAST irányba

BLUE csillagkaput nyitott A4 (SpecialWall) mező WEST oldalán

BLUE lövedék megsemmisült a(z) A4 (SpecialWall) mezőn

ONeill a(z) A3 (Road) mezőről a(z) A4 (SpecialWall) mezőre próbál lépni

A4 (SpecialWall) mező WEST oldalán BLUE csillagkapu van nyitva

A csillagkapu kijárata A1 (SpecialWall) EAST oldala

ONeill áthaladt a csillagkapun

ONeill elfordult EAST irányba

ONeill a(z) A1 (SpecialWall) mezőről a(z) A2 (Road) mezőre próbál lépni

Sikeresen átlépett a következő mezőre: A2 (Road)

8.2.5 Játékos mozgatása szakadékra

- **Leírás**

A játékos ellép az aktuális mezőjéről és az előtte található szakadék típusú mezőre lép. A játékos meghal, a játék véget ér.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy a játékos bele tud-e lépni egy szakadékba. A szakadék, amikor arról értesül, hogy egy játékos rá akar lépni, jelzi a játékos által irányított karakternek, hogy az megsemmisül. Ezután a játék menetét irányító osztály, mikor legközelebb megkérdezi a játékost, hogy él-e még, a tagadó válasz után leállítja a játékot.

- **Bemenet**

Pálya

R_O, G

Parancsok

```
oneill east  
oneill east
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba
ONeill a(z) A1 (Road) mezőről a(z) A2 (Gap) mezőre próbál lépni
Sikeresen átlépett a következő mezőre: A2 (Gap)
ONeill meghalt a(z) A2 (Gap) mezőn
A játék véget ért, nincsen győztes

8.2.6 Doboz felvétele útról

- **Leírás**

A játékos felvesz egy dobozt az előtte található út típusú mezőről.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy a játékos fel tud-e venni egy dobozt az előtte található út típusú mezőről. Ha a programban hiba van, előfordulhat, hogy a felvett doboz eltűnik a mezőről, de a játékoshoz sem kerül (eltűnik a játékból) vagy pedig a játékoshoz kerül, de a mezőn is rajtamarad (két helyen létezik tovább).

- **Bemenet**

Pálya

R_O, R_B

Parancsok

```
oneill east  
oneill pickup
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba
ONeill megpróbál felvenni egy dobozt a(z) A2 (Road) mezőről
Sikeres dobozfelvétel a(z) A2 (Road) mezőről

8.2.7 Doboz felvétele mérlegről

- **Leírás**

A játékos felvesz egy dobozt az előtte található mérleg típusú mezőről. A mérleghez tartozó ajtó bezárul.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt az vizsgálja, hogy a játékos fel tud-e venni egy dobozt az előtte található mérleg típusú mezőről, és a mérleghez tartozó ajtó bezárul-e mikor a megfelelő mennyiségű súly lekerül a mérlegről.

- **Bemenet**

Pálya

R_O, S:1_BB, D:1

Parancsok

```
oneill east  
oneill pickup
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba

ONeill megpróbál felvenni egy dobozt a(z) A2 (Scale) mezőről

Sikeres dobozfelvétel a(z) A2 (Scale) mezőről

A(z) A2 (Scale) mezőről 1 egységnyi súly lekerült, összesen 1 súly van rajta

Mivel nincsen legalább 2 súly a mérlegen, a(z) A3 (Door) bezárul

8.2.8 Doboz felvétele portálon átnyúlva

- **Leírás**

A játékos felvesz egy dobozt az előtte található nyitott portál másik végén található mezőről.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt az vizsgálja, hogy a játékos fel tud-e venni egy dobozt az előtte található nyitott portál másik végén található mezőről.

- **Bemenet**

Pálya

SW, R_B, R_O, SW

Parancsok

```
oneill west  
oneill proj1  
oneill east  
oneill proj2  
oneill pickup
```

- **Elvárt kimenet**

ONeill elfordult WEST irányba
ONeill YELLOW lövedéket lőtt WEST irányba
YELLOW csillagkaput nyitott A1 (SpecialWall) mező EAST oldalán
YELLOW lövedék megsemmisült a(z) A1 (SpecialWall) mezőn
ONeill elfordult EAST irányba
ONeill BLUE lövedéket lőtt EAST irányba
BLUE csillagkaput nyitott A4 (SpecialWall) mező WEST oldalán
BLUE lövedék megsemmisült a(z) A4 (SpecialWall) mezőn
ONeill megpróbál felvenni egy dobozt a(z) A4 (SpecialWall) mezőről
A4 (SpecialWall) mező WEST oldalán BLUE csillagkapu van nyitva
A csillagkapu kijárata A1 (SpecialWall) EAST oldala
A játékos átnyúlt a csillagkapun
Sikeres dobozfelvétel a(z) A2 (Road) mezőről

8.2.9 Doboz lerakása útra

- **Leírás**

A játékos letesz egy dobozt az előtte található út típusú mezőre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy a játékos le tud-e tenni egy dobozt az előtte található út típusú mezőre. Hibátlan működés esetén a játékos kezéből eltűnik a doboz, az előtte lévő mezőn viszont megjelenik.

- **Bemenet**

Pálya

```
R_O, R_B
```

Parancsok

```
oneill east  
oneill pickup  
oneill putdown
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba

ONeill megpróbál felvenni egy dobozt a(z) A2 (Road) mezőről

Sikeres dobozfelvétel a(z) A2 (Road) mezőről

ONeill megpróbál letenni egy dobozt a(z) A2 (Road) mezőre

Sikeres dobozletétel a(z) A2 (Road) mezőre

8.2.10 Doboz lerakása mérlegre

- **Leírás**

A játékos letesz egy dobozt az előtte található mérleg típusú mezőre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy a játékos le tud-e tenni egy dobozt az előtte található út típusú mezőre. Hibátlan működés esetén a játékos kezéből eltűnik a doboz, az előtte lévő mezőn viszont megjelenik. A mérlegen már van egy doboz, így egy újabb doboz lerakásakor a hozzá tartozó ajtónak ki kell nyílnia.

- **Bemenet**

Pálya

R_O, R_B, S:1_B, D:1

Parancsok

oneill east

oneill pickup

oneill east

oneill putdown

- **Elvárt kimenet**

ONeill elfordult EAST irányba

ONeill megpróbál felvenni egy dobozt a(z) A2 (Road) mezőről

Sikeres dobozfelvétel a(z) A2 (Road) mezőről

ONeill a(z) A1 (Road) mezőről a(z) A2 (Road) mezőre próbál lépni

Sikeresen átlépett a következő mezőre: A2 (Road)

ONeill megpróbál letenni egy dobozt a(z) A3 (Scale) mezőre

Sikeres dobozletétel a(z) A3 (Scale) mezőre

A(z) A3 (Scale) mezőre 1 egységnyi súly került, összesen 2 súly van rajta

Mivel legalább 2 súly van a mérlegen, a(z) A4 (Door) kinyílt

8.2.11 Doboz lerakása szakadékba

- **Leírás**

A játékos letesz egy dobozt az előtte található szakadék típusú mezőre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy a szakadékba érkező doboz megsemmisül-e és megjelenik-e újból az eredeti helyén.

- **Bemenet**

Pálya

R_O, R_B, G

Parancsok

```
oneill east  
oneill pickup  
oneill east  
oneill putdown
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba
ONeill megpróbál felvenni egy dobozt a(z) A2 (Road) mezőről
Sikeres dobozfelvétel a(z) A2 (Road) mezőről
ONeill a(z) A1 (Road) mezőről a(z) A2 (Road) mezőre próbál lépni
Sikeresen átlépett a következő mezőre: A2 (Road)
ONeill megpróbál letenni egy dobozt a(z) A3 (Gap) mezőre
Sikeres dobozletétel a(z) A3 (Gap) mezőre
Egy doboz elpusztult a(z) A3 (Gap) mezőn
Egy doboz létrejön a(z) A2 (Road) mezőn
Sikeres dobozletétel a(z) A2 (Road) mezőre

8.2.12 Doboz lerakása falra

- **Leírás**

A játékos megpróbál letenni egy dobozt az előtte található fal típusú mezőre. A falról a doboz "visszapattan" és a játékos mezőjére kerül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy a játékos le tud-e tenni egy dobozt az előtte található fal típusú mezőre (nem tud).

- **Bemenet**

Pálya

R_O, R_B, W

Parancsok

```
oneill east
oneill pickup
oneill east
oneill putdown
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba
ONeill megpróbál felvenni egy dobozt a(z) A2 (Road) mezőről
Sikeres dobozfelvétel a(z) A2 (Road) mezőről
ONeill a(z) A1 (Road) mezőről a(z) A2 (Road) mezőre próbál lépni
Sikeresen átlépett a következő mezőre: A2 (Road)
ONeill megpróbál letenni egy dobozt a(z) A3 (Wall) mezőre
Sikertelen dobozletétel: A doboz visszalökődik az előző A2 (Road) mezőre
Sikeres dobozletétel a(z) A2 (Road) mezőre

8.2.13 Doboz lerakása portálon keresztül

- **Leírás**

A játékos letesz egy dobozt az előtte található nyitott portál másik végén található mezőre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy a játékos le tud-e tenni egy dobozt az előtte található speciális falon lévő nyitott portálon keresztül, a portál másik végén található mezőre. A speciális falon lévő portál felelős a doboz áthelyezéséért (vagy adott esetben - a falhoz hasonlóan - a doboz visszapattintásáért).

- **Bemenet**

Pálya

SW, R_B, R_O, SW

Parancsok

```
oneill west
oneill proj1
oneill pickup
oneill east
oneill proj2
oneill putdown
```

- **Elvárt kimenet**

ONeill elfordult WEST irányba

ONeill YELLOW lövedéket lőtt WEST irányba

YELLOW csillagkaput nyitott A1 (SpecialWall) mező EAST oldalán

YELLOW lövedék megsemmisült a(z) A1 (SpecialWall) mezőn

ONeill megpróbál felvenni egy dobozt a(z) A2 (Road) mezőről

Sikeres dobozfelvétel a(z) A2 (Road) mezőről

ONeill elfordult EAST irányba

ONeill BLUE lövedéket lőtt EAST irányba

BLUE csillagkaput nyitott A4 (SpecialWall) mező WEST oldalán

BLUE lövedék megsemmisült a(z) A4 (SpecialWall) mezőn

ONeill megpróbál letenni egy dobozt a(z) A4 (SpecialWall) mezőre

A4 (SpecialWall) mező WEST oldalán BLUE csillagkapu van nyitva

A csillagkapu kijárata A1 (SpecialWall) EAST oldala

A doboz áthaladt a csillagkapun

Sikeres dobozletétel a(z) A2 (Road) mezőre

8.2.14 Lövés falra

- **Leírás**

A játékos lő egyet, a lövedék fal típusú mezőn ér célba.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy mi történik egy lövedékkel, ha az fal típusú mezőn ér célba. Az elvárás az, hogy semmisüljön meg.

- **Bemenet**

Pálya

R_O, W

Parancsok

```
oneill east
```

```
oneill proj1
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba

ONeill YELLOW lövedéket lőtt EAST irányba

YELLOW lövedék megsemmisült a(z) A2 (Wall) mezőn

8.2.15 Lövés speciális falra

- **Leírás**

A játékos lő egyet, a lövedék speciális fal típusú mezőn ér célba.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy mi történik egy lövedékkel, ha az speciális fal típusú mezőn ér célba. Az elvárás az, hogy megsemmisül és létrehoz egy megfelelő színű csillagkaput.

- **Bemenet**

Pálya

R_O, SW

Parancsok

```
oneill east  
oneill proj1
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba

ONeill YELLOW lövedéket lőtt EAST irányba

YELLOW csillagkaput nyitott A2 (SpecialWall) mező WEST oldalán

YELLOW lövedék megsemmisült a(z) A2 (SpecialWall) mezőn

8.2.16 Lövés replikátorra

- **Leírás**

A játékos lő egyet, a lövedék egy olyan mezőre érkezik, ahol egy replikátor tartózkodik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy mi történik egy lövedékkel, ha az egy replikátorral ütközik. Az elvárás az, hogy a lövedék és a replikátor is megsemmisülnek.

- **Bemenet**

Pálya

R_O, R_R

Parancsok

```
oneill east  
oneill proj1
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba

ONeill YELLOW lövedéket lőtt EAST irányba

Replicator megsemmisült a(z) A2 (Road) mezőn

YELLOW lövedék megsemmisült a(z) A2 (Road) mezőn

8.2.17 Utolsó ZPM felvétele

- **Leírás**

A játékos felszedi a pályán található egyetlen ZPM modult.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az elvárt működés, hogy a ZPM modul eltűnik, a játékos pontszáma nő eggyel és a játéknak vége lesz.

- **Bemenet**

Pálya

R_O, R_Z

Parancsok

```
oneill east  
oneill east
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba
ONeill a(z) A1 (Road) mezőről a(z) A2 (Road) mezőre próbál lépni
Sikeresen átlépett a következő mezőre: A2 (Road)
ONeill felvett egy ZPM-et
A játék véget ért, O'Neill nyert

8.2.18 O'Neill ezredes felvesz két ZPM-et

- **Leírás**

Az ezredes felszedi a pályán található két ZPM modult.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az elvárt működés, hogy a ZPM modulok eltűnnek, az ezredes pontszáma nő és a második ZPM felvétele után egy újabb ZPM keletkezik az egyik mezőn..

- **Bemenet**

Pálya

R_O, R_Z, R_Z

Parancsok

```
zpm_random off  
oneill east  
oneill east  
oneill east
```

- **Elvárt kimenet**

ONeill elfordult EAST irányba

ONeill a(z) A1 (Road) mezőről a(z) A2 (Road) mezőre próbál lépni

Sikeresen átlépett a következő mezőre: A2 (Road)

ONeill felvett egy ZPM-et

ONeill a(z) A2 (Road) mezőről a(z) A3 (Road) mezőre próbál lépni

Sikeresen átlépett a következő mezőre: A3 (Road)

ONeill felvett egy ZPM-et

Egy új ZPM generálódott a(z) A1 (Road) mezőn

8.2.19 Egyik játékos megöli a másikat

- **Leírás**

Az ezredes rááll a mérlegre, ami kinyitja a pályán található ajtót. A Jaffa ráál az ajtóral. Az ezredes ezután leszáll az ajtóhoz tartozó mérlegről, így az ajtó becsukódik és a Jaffa meghal. A játéknak vége lesz, az ezredes nyer.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az elvárt működés, hogy az ajtó bezárásakor értesítse a Jaffát annak haláláról. A játék ezután az ezredes győzelmével ér véget.

- **Bemenet**

Pálya

R_O, S:1, R_J, D:1

Parancsok

oneill east

oneill east

jaffa east

jaffa east

oneill east

- **Elvárt kimenet**

ONeill elfordult EAST irányba

ONeill a(z) A1 (Road) mezőről a(z) A2 (Scale) mezőre próbál lépni

Sikeresen átlépett a következő mezőre: A2 (Scale)

A(z) A2 (Scale) mezőre 2 egységnyi súly került, összesen 2 súly van rajta

Mivel legalább 2 súly van a mérlegen, a(z) A4 (Door) kinyílt

Jaffa elfordult EAST irányba

Jaffa a(z) A3 (Road) mezőről a(z) A4 (Door) mezőre próbál lépni

Sikeresen átlépett a következő mezőre: A4 (Door)

ONeill a(z) A2 (Scale) mezőről a(z) A3 (Road) mezőre próbál lépni

A(z) A2 (Scale) mezőről 2 egységnyi súly lekerült, összesen 0 súly van rajta

Mivel nincsen legalább 2 súly a mérlegen, a(z) A4 (Door) bezárul

Jaffa meghalt a(z) A4 (Door) mezőn

Sikeresen átlépett a következő mezőre: A3 (Road)
A játék véget ért, O'Neill nyert

8.2.20 Replikátor szakadékba esik

- **Leírás**

A replikátor szakadékba esik, a replikátor megsemmisül, a szakadék típusú mező pedig út típusú mezővé változik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt azt vizsgálja, hogy a replikátor rá tud-e lépni egy szakadék típusú mezőre, ilyenkor a szakadék típusú mező úttá változik-e.

- **Bemenet**

Pálya

R_OR, G

Parancsok

```
replicator east  
replicator east
```

- **Elvárt kimenet**

Replicator elfordult EAST irányba

Replicator a(z) A1 (Road) mezőről a(z) A2 (Gap) mezőre próbál lépni

Sikeresen átlépett a következő mezőre: A2 (Gap)

Replicator megsemmisült a(z) A2 (Gap) mezőn

A(z) A2 (Gap) mező Road mezővé alakult

8.3 A tesztelést támogató programok tervei

A tesztelést támogató szoftver a játék erre a célra elkészített változata lesz, ahol a tesztesetek futtatását és ellenőrzését a játék menüjéből lehet kiválasztani egy külön funkcióként. Ezt a funkciót választva a program az összes specifikált tesztesetet lefuttatja és a kimenetüket menet közben összehasonlítja az elvárt eredménnyel.

Az alkalmazás az előző pontban specifikált pályát és parancsokat kapja meg tesztesetenként külön fájlokban. A pálya leírását egy szöveges fájl fogja tartalmazni, a korábban ismertetett bemeneti nyelv szerint. A teszteset lefutásához szükséges parancsokat egy másik szöveges állomány fogja tartalmazni. A fájlok nevei a tesztek neveinek angol összefoglalói lesznek.

Példa a bemeneti fájlokra és azok tartalmára:

Egy tesztet (8.2.1.) futása és ellenőrzése a következőképpen néz ki:

Az alkalmazás a következő bemeneti fájlokat olvassa be:

- /maps/PlayerMoveRoad.txt, amelynek tartalma:
R_O, R
- /testinputs/PlayerMoveRoad.txt, amelynek tartalma:
oneill east
oneill east

A bemeneti fájlok az alkalmazás gyökérkönyvtárában található 'res' mappában lesznek elhelyezve, de a program fordításakor ezek a binárisok mellé másolódnak és a futó program a fájloknak ezt a változatát fogja használni.

További bemenetei az alkalmazásnak az elvárt kimeneteket tartalmazó fájlok, amelyekkel a generált szöveges fájlokat hasonlítja össze a szoftver. Ezeket az 'res/expectedresults' mappában találhatjuk majd meg a projektben (a fenti példában PlayerMoveRoad.txt lesz az aktuális elvárt kimenetet tartalmazó szöveges fájl neve).

A program a tesztek elején beolvassa a megadott térképet, felépíti azt, majd végrehajtja a megadott parancsokat. Az ellenőrző funkció futása során a program szöveges kimenete a szabványos kimenet helyett egy ellenőrző kimenetre kerül, ahol minden sor kiírásakor ellenőrzi, hogy az éppen kiírni kívánt sor megegyezik-e az elvárt kimenettel.

Ha az elvárt és a kapott kimenet között eltérés van, a program mindkettőt kiírja a szabványos kimenetre. Ezen kívül tesztenkénti és teljes összesítést is kapunk a hibás sorok számáról.

10 Prototípus beadása

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
.\src\controller\Controller.java	5874 byte	2016.03.27 16:38	Controller osztály.
.\src\controller\Game.java	2884 byte	2016.03.24 13:20	Game osztály. Számon tartja a játékosok pontszámát.
.\src\controller\InputProcessor.java	4242 byte	2016.04.22 16:53	Saját parancsok fordítását végző osztály.
.\src\controller\MapBuilder.java	8209 byte	2016.03.24 13:18	MapBuilder osztály. Felépíti a pályát, létrehozza és elhelyezi rajta a játékost, a ZPM-et és a dobozt.
.\src\controller\MapHelper.java	1270 byte	2016.04.13 15:42	Segédosztály a szakadék úttá építéséhez, illetve egy véletlenszerű út visszaadásához.
.\src\controller\PlayerType.java	69 byte	2016.04.13 16:16	Játékosok enumerációja.
.\src\debug\ProtoLogger.java	1307 byte	2016.03.27 16:53	Logolást végző osztály.
.\src\debug\RandomGenerator.java	758 byte	2016.04.22 18:39	A véletlenszerűség kezeléséért felelő osztály.
.\src\gamemodel\Box.java	974 byte	2016.03.24 13:09	Box osztály.
.\src\gamemodel\Direction.java	705 byte	2016.03.24 12:54	Direction osztály.
.\src\gamemodel\Door.java	3126 byte	2016.03.24 13:37	Door osztály.
.\src\gamemodel\Gap.java	1424 byte	2016.03.24 13:28	Gap osztály.
.\src\gamemodel\Jaffa.java	264 byte	2016.04.13 14:01	Jaffa osztály.
.\src\gamemodel\MapElement.java	3629 byte	2016.03.24 12:47	MapElement osztály.
.\src\gamemodel\Movable.java	774 byte	2016.03.24 12:55	Movable osztály.
.\src\gamemodel\ONeill.java	268 byte	2016.04.13 13:58	ONeill osztály.
.\src\gamemodel\Player.java	3857 byte	2016.03.24 12:58	Player osztály.
.\src\gamemodel\Projectile.java	1855 byte	2016.03.24 13:08	Projectile osztály.
.\src\gamemodel\ProjectileType.java	333 byte	2016.03.24 13:01	ProjectileType osztály.
.\src\gamemodel\Replicator.java	1868 byte	2016.04.13 15:32	Replicator osztály.
.\src\gamemodel\Road.java	2019 byte	2016.03.24 13:35	Road osztály.
.\src\gamemodel\Scale.java	4443 byte	2016.03.24 13:29	Scale osztály.

.\src\gamemodel\SpecialWall.java	4591 byte	2016.03.24 13:24	SpecialWall osztály.
.\src\gamemodel\Stargate.java	1534 byte	2016.03.24 13:15	Stargate osztály.
.\src\gamemodel\Wall.java	374 byte	2016.03.24 13:32	Wall osztály.
.\src\gamemodel\ZPM.java	389 byte	2016.03.24 13:37	ZPM osztály.
.\src\gamemodel\events\IProjectileStateListener.java	326 byte	2016.03.28 3:50	A Projectile keletkezésével és megsemmisülésével kapcsolatos üzeneteket fogadó osztályok interfésze.
.\src\gamemodel\events\IReplicatorDestroyedListener.java	222 byte	2016.04.22 18:33	A replikátor megsemmisülésével kapcsolatos üzeneteket fogadó osztályok interfésze.
.\src\gamemodel\events\IZPMPickedUpListener.java	220 byte	2016.03.28 3:52	A ZPM és felvételével kapcsolatos üzeneteket fogadó osztályok interfésze.
.\src\gamemodel\events\ModelEventSource.java	2446 byte	2016.03.28 3:58	A játékmód változásairól küldött üzeneteket küldő osztály.
.\src\test\PrototypeValidator.java	3223 byte	2016.04.24 17:44	A tesztesetek kimenetének ellenőrzésére szolgáló osztály.
.\src\test\TestRunner.java	901 byte	2016.03.27 17:21	TestRunner osztály. Elindít egy új játékot és egy InputProcessort a tesztesetek feldolgozásához.
.\src\test\TestType.java	894 byte	2016.03.28 13:55	TestType osztály. Teszt típusok.
.\src\userinterface\Main.java	16119 byte	2016.03.24 13:40	Main osztály. Menü. A program belépési pontja
.\res\expectedresults\BoxPickUpRoad.txt	147 byte	2016.04.24 18:41	BoxPickUpRoad teszt elvárt kimenete.
.\res\expectedresults\BoxPickUpScale.txt	306 byte	2016.04.24 18:41	BoxPickUpScale teszt elvárt kimenete.
.\res\expectedresults\BoxPickUpWormhole.txt	693 byte	2016.04.24 18:41	BoxPickUpWormhole teszt elvárt kimenete.
.\res\expectedresults\BoxPutDownGap.txt	518 byte	2016.04.24 18:41	BoxPutDownGap teszt elvárt kimenete.
.\res\expectedresults\BoxPutDownRoad.txt	256 byte	2016.04.24 18:41	BoxPutDownRoad teszt elvárt kimenete.
.\res\expectedresults\BoxPutDownScale.txt	535 byte	2016.04.24 18:41	BoxPutDownScale teszt elvárt kimenete.
.\res\expectedresults\BoxPutDownWall.txt	465 byte	2016.04.24 18:41	BoxPutDownWall teszt elvárt kimenete.
.\res\expectedresults\BoxPutDownWormhole.txt	798 byte	2016.04.24 18:41	BoxPutDownWormhole teszt elvárt kimenete.
.\res\expectedresults\LastZPM.txt	225 byte	2016.04.24 18:41	LastZPM teszt elvárt kimenete.
.\res\expectedresults\PlayerKillsOtherPlayer.txt	830 byte	2016.04.24 18:41	PlayerKillsOtherPlayer teszt elvárt kimenete.
.\res\expectedresults\PlayerMoveGap.txt	237 byte	2016.04.24 18:41	PlayerMoveGap teszt elvárt kimenete.
.\res\expectedresults\PlayerMoveRoad.txt	160 byte	2016.04.24 18:41	PlayerMoveRoad teszt elvárt kimenete.
.\res\expectedresults\PlayerMoveScale.txt	311 byte	2016.04.24 18:41	PlayerMoveScale teszt elvárt kimenete.
.\res\expectedresults\PlayerMoveWall.txt	236 byte	2016.04.24 18:41	PlayerMoveWall teszt elvárt kimenete.

.res\expectedresults\PlayerMoveWormhole.txt	812 byte	2016.04.24 18:41	PlayerMoveWormhole teszt elvárt kimenete.
.res\expectedresults\ReplicatorMoveGap.txt	258 byte	2016.04.24 18:41	ReplicatorMoveGap teszt elvárt kimenete.
.res\expectedresults\ShootReplicator.txt	183 byte	2016.04.24 18:41	ShootReplicator teszt elvárt kimenete.
.res\expectedresults\ShootSpecialWall.txt	208 byte	2016.04.24 18:41	ShootSpecialWall teszt elvárt kimenete.
.res\expectedresults\ShootWall.txt	134 byte	2016.04.24 18:41	ShootWall teszt elvárt kimenete.
.res\expectedresults\TwoZPMs.txt	395 byte	2016.04.24 18:41	TwoZPMs teszt elvárt kimenete.
.res\maps\BoxPickUpRoad.txt	8 byte	2016.04.22 19:19	BoxPickUpRoad teszt pályáját leíró fájl.
.res\maps\BoxPickUpScale.txt	16 byte	2016.04.22 19:19	BoxPickUpScale teszt pályáját leíró fájl.
.res\maps\BoxPickUpWormhole.txt	16 byte	2016.04.22 19:19	BoxPickUpWormhole teszt pályáját leíró fájl.
.res\maps\BoxPutDownGap.txt	11 byte	2016.04.22 19:19	BoxPutDownGap teszt pályáját leíró fájl.
.res\maps\BoxPutDownRoad.txt	8 byte	2016.04.22 19:19	BoxPutDownRoad teszt pályáját leíró fájl.
.res\maps\BoxPutDownScale.txt	20 byte	2016.04.22 19:19	BoxPutDownScale teszt pályáját leíró fájl.
.res\maps\BoxPutDownWall.txt	11 byte	2016.04.22 19:19	BoxPutDownWall teszt pályáját leíró fájl.
.res\maps\BoxPutDownWormhole.txt	16 byte	2016.04.22 19:19	BoxPutDownWormhole teszt pályáját leíró fájl.
.res\maps\LastZPM.txt	8 byte	2016.04.22 19:19	LastZPM teszt pályáját leíró fájl.
.res\maps\PlayerKillsOtherPlayer.txt	18 byte	2016.04.22 19:19	PlayerKillsOtherPlayer teszt pályáját leíró fájl.
.res\maps\PlayerMoveGap.txt	6 byte	2016.04.22 19:19	PlayerMoveGap teszt pályáját leíró fájl.
.res\maps\PlayerMoveRoad.txt	6 byte	2016.04.22 13:39	PlayerMoveRoad teszt pályáját leíró fájl.
.res\maps\PlayerMoveScale.txt	13 byte	2016.04.22 18:59	PlayerMoveScale teszt pályáját leíró fájl.
.res\maps\PlayerMoveWall.txt	6 byte	2016.04.22 19:19	PlayerMoveWall teszt pályáját leíró fájl.
.res\maps\PlayerMoveWormhole.txt	14 byte	2016.04.22 19:19	PlayerMoveWormhole teszt pályáját leíró fájl.
.res\maps\ReplicatorMoveGap.txt	7 byte	2016.04.22 19:19	ReplicatorMoveGap teszt pályáját leíró fájl.
.res\maps\ShootReplicator.txt	8 byte	2016.04.22 19:19	ShootReplicator teszt pályáját leíró fájl.
.res\maps\ShootSpecialWall.txt	7 byte	2016.04.22 19:19	ShootSpecialWall teszt pályáját leíró fájl.
.res\maps\ShootWall.txt	6 byte	2016.04.22 19:19	ShootWall teszt pályáját leíró fájl.
.res\maps\TwoZPMs.txt	13 byte	2016.04.22 19:19	TwoZPMs teszt pályáját leíró fájl.
.res\testinputs\BoxPickUpRoad.txt	26 byte	2016.04.22 19:19	BoxPickUpRoad teszt elvégzéséhez szükséges parancsok.
.res\testinputs\BoxPickUpScale.txt	26 byte	2016.04.22 19:19	BoxPickUpScale teszt elvégzéséhez szükséges parancsok.
.res\testinputs\BoxPickUpWormhole.txt	67 byte	2016.04.22 19:19	BoxPickUpWormhole teszt elvégzéséhez szükséges parancsok.

.\res\testinputs\BoxPutDownGap.txt	55 byte	2016.04.22 19:19	BoxPutDownGap teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\BoxPutDownRoad.txt	42 byte	2016.04.22 19:19	BoxPutDownRoad teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\BoxPutDownScale.txt	55 byte	2016.04.22 19:19	BoxPutDownScale teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\BoxPutDownWall.txt	55 byte	2016.04.22 19:19	BoxPutDownWall teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\BoxPutDownWormhole.txt	83 byte	2016.04.22 19:19	BoxPutDownWormhole teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\LastZPM.txt	24 byte	2016.04.22 19:19	LastZPM teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\PlayerKillsOtherPlayer.txt	61 byte	2016.04.22 19:19	PlayerKillsOtherPlayer teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\PlayerMoveGap.txt	24 byte	2016.04.22 19:19	PlayerMoveGap teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\PlayerMoveRoad.txt	24 byte	2016.04.22 14:36	PlayerMoveRoad teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\PlayerMoveScale.txt	24 byte	2016.04.22 18:59	PlayerMoveScale teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\PlayerMoveWall.txt	24 byte	2016.04.22 19:19	PlayerMoveWall teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\PlayerMoveWormhole.txt	65 byte	2016.04.22 19:19	PlayerMoveWormhole teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\ReplicatorMoveGap.txt	32 byte	2016.04.22 19:19	ReplicatorMoveGap teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\ShootReplicator.txt	25 byte	2016.04.22 19:19	ShootReplicator teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\ShootSpecialWall.txt	25 byte	2016.04.22 19:19	ShootSpecialWall teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\ShootWall.txt	25 byte	2016.04.22 19:19	ShootWall teszt elvégzéséhez szükséges parancsok.
.\res\testinputs\TwoZPMs.txt	53 byte	2016.04.22 19:19	TwoZPMs teszt elvégzéséhez szükséges parancsok.
.\compile.bat	456 byte	2016.03.29 2:46	Fordítást végző batch fájl.
.\run.bat	118 byte	2016.03.29 3:17	Futtatást végző batch fájl.

10.1.2 Fordítás

A beadott könyvtárstruktúra gyökerében található *compile.bat* szkriptet futtatva jönnek létre a class fájlok és másolódnak be a resource-ok a binárisok mellé.

10.1.3 Futtatás

A beadott könyvtárstruktúra gyökerében található *run.bat* szkriptet futtatva indítható el a program (természetesen csak sikeres fordítás után).

A prototípusnak a következőkre van szüksége a futáshoz:

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
E01	HSZK-s számítógép	kiértékelés	alapvető	megrendelő	JRE 1.6+
E02	monitor	nincs	alapvető	csapat	
E03	billentyűzet	nincs	alapvető	csapat	

Ezekon kívül szükséges, hogy a futtató gépén a megfelelő verziójú JRE és JDK szerepeljen a környezeti változók között.

10.2 Tesztek jegyzőkönyvei

10.2.1 Játékos mozgatása útra

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 10:58

10.2.2 Játékos mozgatása mérlegre

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:00

10.2.3 Játékos mozgatása falra

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:02

10.2.4 Játékos mozgatása speciális falra (nyitott portál)

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:04

10.2.5 Játékos mozgatása szakadékba

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:08

10.2.6 Felvétel útról

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:09

10.2.7 Felvétel mérlegről

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:10

10.2.8 Felvétel portálon átnyúlva

Tesztelő neve	Csóka Máté
Teszt időpontja	2016.04.23. 16:15

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:40
Teszt eredménye	<p>ONEill elfordult WEST irányba ONEill YELLOW lövedéket lőtt WEST irányba YELLOW csillagkaput nyitott A1 (SpecialWall) mező EAST oldalán YELLOW lövedék megsemmisült a(z) A1 (SpecialWall) mezőn ONEill elfordult EAST irányba ONEill BLUE lövedéket lőtt EAST irányba BLUE csillagkaput nyitott A4 (SpecialWall) mező WEST oldalán BLUE lövedék megsemmisült a(z) A4 (SpecialWall) mezőn ONEill megpróbál felvenni egy dobozt a(z) A4 (SpecialWall) mezőről A4 (SpecialWall) mező WEST oldalán BLUE csillagkapu van nyitva A csillagkapu kijárata A1 (SpecialWall) EAST oldala A játékos átnyúlt a csillagkapun</p>
Lehetséges hibaok	<p>Player osztályban a pickUpBox fv-ben az alábbi kódrészlet található: if (box != null) { box.leaveMapElement(nextPosition);....} Amikor egy feregjáraton áthalad akkor nem a közvetlen szomszadját kell átadni paraméterként a leaveMapElement fv-ben, hanem a feregjárat kijárata mezőt.</p>
Változtatások	Player pickUpBox fv-ében box.leaveMapElement(nextPosition); helyett box.leaveMapElement(box.position);

10.2.9 Lerakás útra

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:15

10.2.10 Lerakás mérlegre

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:16

10.2.11 Lerakás szakadékba

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:18

10.2.12 Lerakás falra

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:20

10.2.13 Lerakás portálon keresztül

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:21

10.2.14 Lövés falra

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:23

10.2.15 Lövés speciális falra

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 11:25

10.2.16 Lövés replikátorra

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 12:05

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 12:00
Teszt eredménye	<p>ONeill elfordult EAST irányba ONeill YELLOW lövedéket lőtt EAST irányba Replicator megsemmisült a(z) A2 (Road) mezőn YELLOW lövedék megsemmisült a(z) A2 (Road) mezőn !!! Projectile tried to move on a MapElement that does not exist !!! VAGY EZ: ONeill elfordult EAST irányba ONeill YELLOW lövedéket lőtt EAST irányba Replicator megsemmisült a(z) A2 (Road) mezőn !!! Projectile tried to move on a MapElement that does not exist !!! YELLOW lövedék megsemmisült a(z) A2 (Road) mezőn</p>
Lehetséges hibaok	A lövedék nem semmisül meg a replikátorral való találkozáskor és továbbhalad a következő mezőre (Ami nem is létezik)
Változtatások	<p>Az alábbi kódrészlet beszúrása a Road osztály handleProjectileArrive fv-ébe, az if (replicator != null) {}-en belülre.</p> <pre>projectile.setPosition(this); projectile.destroy();</pre>

10.2.17 Utolsó ZPM felvétele

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 12:28

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 12:20
Teszt eredménye	<p>ONeill elfordult EAST irányba ONeill a(z) A1 (Road) mezőről a(z) A2 (Road) mezőre próbál lépni Sikeresen átlépett a következő mezőre: A2 (Road) A játék véget ért, O'Neill nyert</p>
Lehetséges hibaok	Ha a játékos felvesz egy ZPM-et, akkor az nincs kiírva.
Változtatások	<p>Road osztály handlePlayerArrive fv-ébe az if (zpm != null) {}-en belülre az alábbi kódrészletet kell beilleszteni:</p> <pre>ProtoLogger.log(player.toString() + " felvett egy ZPM-et");</pre>

10.2.18 Két ZPM felvétele

Tesztelő neve	Csóka Máté
Teszt időpontja	2016.04.23. 16:00

10.2.19 Egyik játékos megöli a másikat

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 13:00

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 12:40
Teszt eredménye	<p>ONeill elfordult EAST irányba</p> <p>ONeill a(z) A1 (Road) mezőről a(z) A2 (Scale) mezőre próbál lépni</p> <p>Sikeresen átlépett a következő mezőre: A2 (Scale)</p> <p>A(z) A2 (Scale) mezőre 2 egységnyi súly került, összesen 2 súly van rajta</p> <p>Mivel legalább 2 súly van a mérlegen, a(z) A4 (Door) kinyílt</p> <p>Jaffa elfordult EAST irányba</p> <p>Jaffa a(z) A3 (Road) mezőről a(z) A4 (Door) mezőre próbál lépni</p> <p>Sikeresen átlépett a következő mezőre: A4 (Door)</p> <p>ONeill a(z) A2 (Scale) mezőről a(z) A3 (Road) mezőre próbál lépni</p> <p>A(z) A2 (Scale) mezőről 2 egységnyi súly lekerült, összesen 0 súly van rajta</p> <p>Mivel nincsen legalább 2 súly a mérlegen, a(z) A4 (Door) bezárul</p> <p>Sikeresen átlépett a következő mezőre: A3 (Road)</p>
Lehetséges hibaok	A Door osztály setOpened fv. if(isOpened=false) feltétele hibás
Változtatások	Az új feltétel: isOpened==false

10.2.20 Replikátor szakadékba esik

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 13:10

Tesztelő neve	Rosner Roland
Teszt időpontja	2016.04.23. 13:00
Teszt eredménye	<p>Replicator elfordult EAST irányba</p> <p>Replicator a(z) A1 (Road) mezőről a(z) A2 (Gap) mezőre próbál lépni</p> <p>Sikeresen átlépett a következő mezőre: A2 (Gap)</p> <p>Replicator megsemmisült a(z) A2 (Gap) mezőn</p>
Lehetséges hibaok	Nincsen kiíratva, amikor egy Gap mező Road mezővé alakul.
Változtatások	A Gap osztály handleReplicatorArrive fv-ébe az alábbi kódrészletet kell beszúrni: ProtoLogger.log("A(z) " + this.toString() + " mező Road mezővé alakult");

10.3 Értékelés

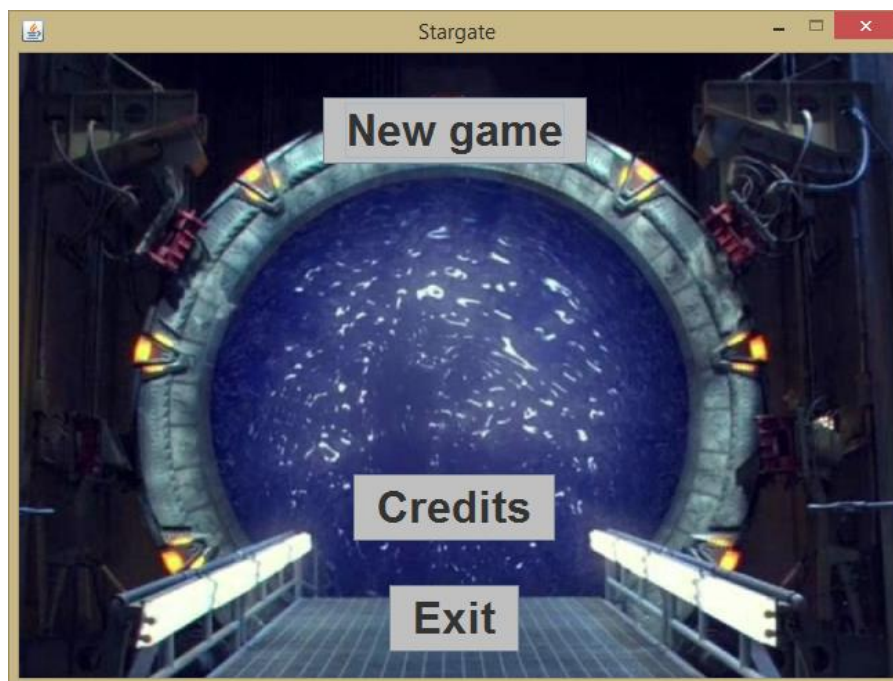
Tag neve	Munka százalékban
Csóka Máté	18
Rosner Roland	18
Sándor Mátyás Márton	28
Szalisznyó Ivett	18
Vándor Dávid	18

11 Grafikus felület specifikációja

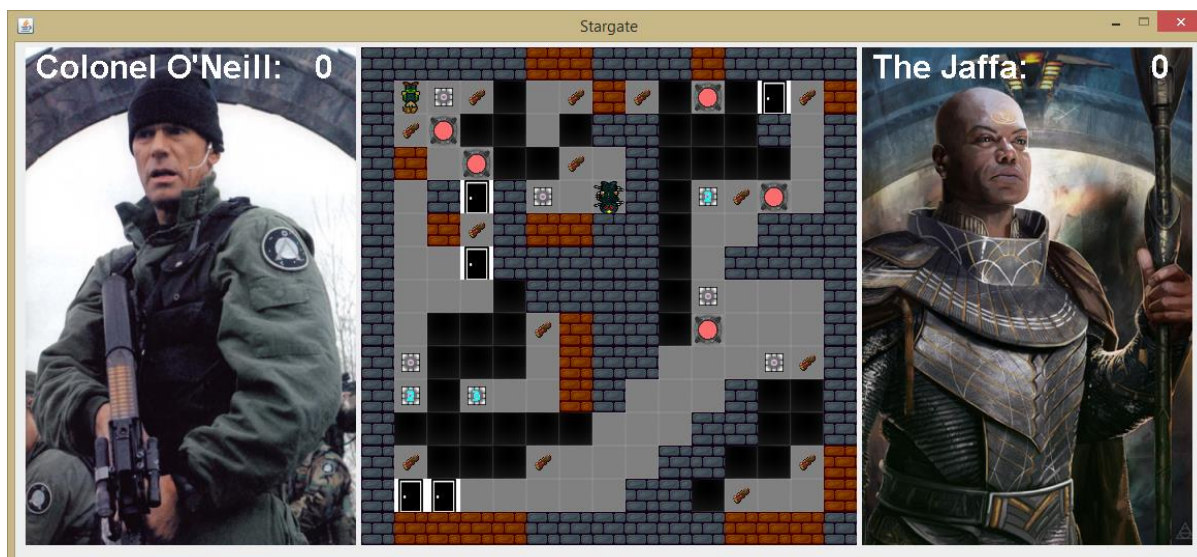
11.1 A grafikus interfész

A játék futtatását követően egy főmenü(11.1.1) jelenik meg, ahol el lehet indítani egy játékot(New Game - 11.1.2) vagy megtekinteni a fejlesztő csapatot(Credits - 11.1.4), illetve kilépni(Exit). Amikor vége van egy játéknak, mert egy vagy mindkét játékos meghalt vagy már nincs több ZPM a pályán, akkor egy ablakban(11.1.3) dr. László Zoltán tanár úr hirdeti ki a nyertest.

11.1.1 Főmenü



11.1.2 Játéktér



A felületen lévő ikonok és jelentésük:

						
O'Neill	Jaffa	Replikátor	Doboz*	Mozgó lövedék	Csillagkapu	ZPM
						
Út	Szakadék	Fal	Speciális fal	Mérleg	Nyitott ajtó	Zárt ajtó

*Ha több doboz található egymáson akkor ezt egy számmal jelzi.

11.1.3 Irányítás

A játékos a saját karakterét a játékot működtető billentyűzet segítségével tudja irányítani.

O'Neill: **W** - északi irány

A - nyugati irány

S - déli irány

D - keleti irány

SPACE -doboz felvétel és letétel

F - sárga színű lövedék lövése

G -kék színű lövedék lövése

Jaffa: ↑ -északi irány

← -nyugati irány

↓ -déli irány

→ -keleti irány

3 - doboz felvétel és letétel

1 -piros színű lövedék lövése

2 -zöld színű lövedék lövése

11.1.4 Játék vége



11.1.5 Fejlesztők és a Tanár úr



11.2 A grafikus rendszer architektúrája

11.2.1 A felület működési elve

A felhasználói felület megvalósítását **push** alapú modell szerint, az *EventSource* - *Observer* design pattern alapján készítjük el. A *controller* rétegben található *Observer*ek a *model* réteg lényeges eseményeit gyűjtik össze és dolgozzák fel. A *controller* réteg is publikál egy *EventSource*-ot, amin keresztül a *user interface* réteg értesül azokról a változásokról, amelyek a *user interface* frissítését vonják maguk után (ikoncsere, új vagy megsemmisült

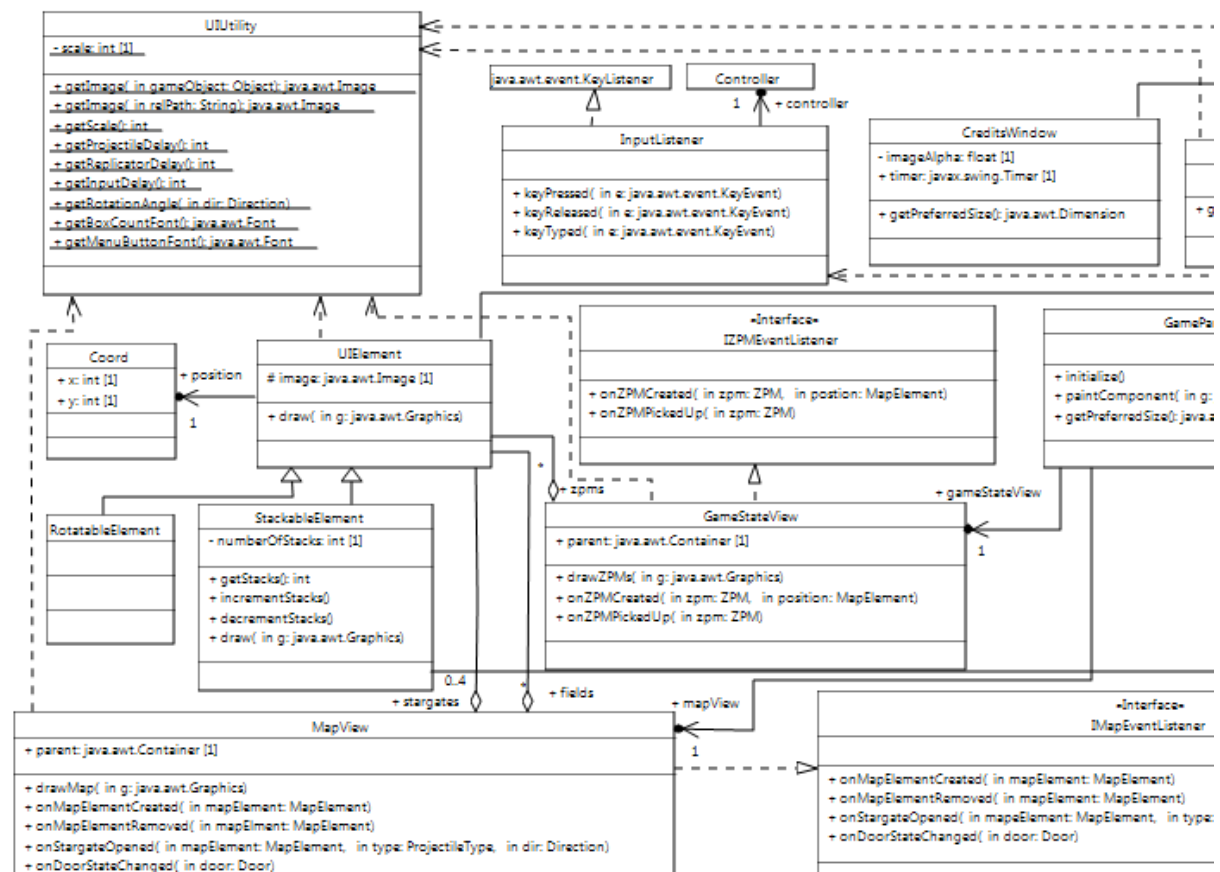
objektum). Ennek megtervezésekor a *user interface*-nek a többi rétegtől való minél nagyobb függetlenségére törekedtünk - az itt publikált események *model* rétegbeli objektumokat szinte kizárólag azok referenciájának kulcsként való használata érdekében adnak át.

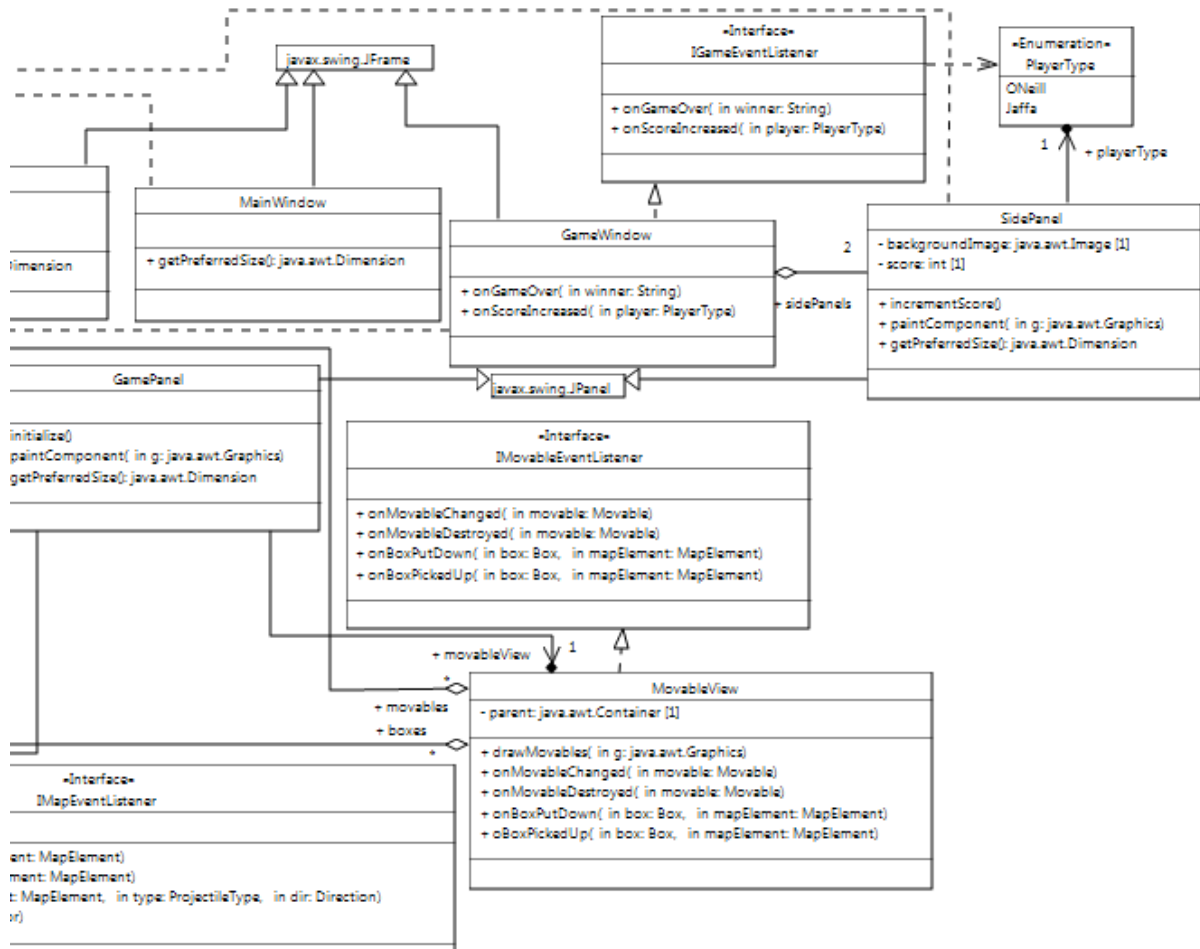
A *user interface* réteg építőelemei egy több lépcsőből álló hierarchiát alkotnak. Legalul állnak a *userinterface.elements* package tagjai. Az ebben szereplő osztályok példányai a játék egy-egy alapvető elemét reprezentálják a *user interface*-en. Ezek feladata, hogy kérésre a megfelelő képet a megfelelő pozícióban a képernyőre rajzolják.

Egy szinttel feljebb a *userinterface.viewobjects* package osztályai találhatók. Ezek feladata, hogy fogadják a *controller* rétegből érkező üzeneteket és azok alapján karbantartsák az aktuálisan létező user interface elemek listáját és szükség esetén értesítsék az őket tartalmazó ablakot, hogy rajzolja újra tartalmát.

Efelett pedig a `userinterface.containers` package található. Ebben a különböző ablakok és az ablakokon belüli panelek szerepelnek, melyek a *swing* keretrendszerbe illeszkednek. Ezek felelnek a megfelelő view objectek létrehozásáért és a kirajzolást végző függvények meghívásáért.

11.2.2 A felület osztály-struktúrája





11.3 A grafikus objektumok felsorolása

11.3.1 Controller

- **Felelősség**
A játékban szereplő objektumokat irányítja, ellenőrzi azok állapotát és továbbítja az eseményeket a user interface réteg felé.
- **Ősosztályok**
Nincs ősosztálya.
- **Interfészek**
IModelEventListener
- **Attribútumok**
 - **-HashMap<PlayerType, Player> players:** A játszható karaktereket reprezentáló osztály
 - **-boolean isReplicatorMoving:** A replikátor állapotát tárolja, *volatile* változó. Alap esetben hamis.

- **Metódusok**

- **+List<Player> getPlayers():** Visszaadja a létező karakterek referenciáját.
- **+void gameOver():** Minden változót visszaállít alaphelyzetbe a Controller osztályban.
- **+void moveOrTurnPlayer(PlayerType playerType, Direction dir):** A játékos fordítása vagy mozgatása egy irányba a jelenlegi iránya függvényében. Ha abba az irányba áll, amit paraméterül is kap, akkor a mozgást végző függvény hívódik meg; ellenkező esetben marad az aktuális MapElementen, és elfordul a kapott irányba.
- **+void moveOrTurnReplicator(Direction dir):** A replikátor fordítása vagy mozgatása egy irányba a jelenlegi iránya függvényében. Ha abba az irányba áll, amit paraméterül is kap, akkor a mozgást végző függvény hívódik meg; ellenkező esetben marad az aktuális MapElementen, és elfordul a kapott irányba.
- **+void shootFirst(PlayerType playerType):** A paraméterül kapott játékos első színű lövedékének kilövése.
- **+void shootSecond(PlayerType playerType):** A paraméterül kapott játékos második színű lövedékének kilövése.
- **+void pickUpOrPutDownBox(PlayerType playerType):** A paraméterül kapott játékos megpróbál felvenni egy dobozt az előtte lévő területről; ha nem tud, akkor ha van nála azt lerakja az előtte lévő mezőre, melyről minden feliratkozott objektum értesül.
- **+void moveReplicatorUntilDeath():** Egy külön szálát felhasználva mozgatjuk a replikátort véletlenszerűen.
- **+void onStargateOpened(MapElement mapElement, ProjectileType type, Direction direction):** A ControllerEventSource feliratkozóit értesíti egy csillagkapú kinyílásáról.
- **+void onReplicatorDestroyed(Replicator replicator):** Ha a replikátor elpusztul, akkor töröljük a rá mutató referenciákat és értesítjük a feliratkozókat.
- **+void onMovableChanged(Movable movable):** Értesítjük a feliratkozókat egy Movable megváltozásáról.
- **+void onDoorStateChanged(Door door):** Értesítjük a feliratkozókat egy ajtó kinyitásáról vagy becsukásáról.

11.3.2 ControllerEventSource

- **Felelősség**

Kezeli a feliratkozó objektumokat, fel- és leiratkozást valamint az értesítést a feliratkozóknak.

- **Ősosztályok**

Nincs ősosztálya.

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **+Set<IMapEventListener> mapEventListeners:** Az IMapEventListenerre feliratkozók halmaza

- **+Set<IMovableEventListener> movableEventListeners:** Az IMovableEventListenerre feliratkozók halmaza
- **+Set<IZPMEventListener> zpmEventListeners:** Az IZPMEventListenerre feliratkozók halmaza
- **+Set<IGameEventListener> gameEventListeners:** Az IGameEventListenerre feliratkozók halmaza
- **Metódusok**
 - **-void subscribe(IMapEventListener listener):** Hozzáadunk egy feliratkozót az IMapEventListenerhez.
 - **-void subscribe(IMovableEventListener listener):** Hozzáadunk egy feliratkozót az IMovableEventListenerhez.
 - **-void subscribe(IZPMEventListener listener):** Hozzáadunk egy feliratkozót az IZPMEventListenerhez.
 - **-void subscribe(IGameEventListener listener):** Hozzáadunk egy feliratkozót az IGameEventListenerhez.
 - **-void clear():** Töröljük az összes feliratkozót.
 - **-void notifyMapElementCreated(MapElement mapElement):** Értesíti a feliratkozókat egy MapElement létrejöttéről.
 - **-void notifyMapElementRemoved(MapElement mapElement):** Értesíti a feliratkozókat egy MapElement eltávolításáról.
 - **-void notifyStargateOpened(MapElement mapElement, ProjectileType type, Direction direction):** Értesíti a feliratkozókat egy csillagkapu nyitásáról.
 - **-void notifyDoorStateChanged(Door door):** Értesíti a feliratkozókat egy ajtó kinyitásáról vagy becsukásáról..
 - **-void notifyMovableChanged(Movable movable):** Értesíti a feliratkozókat egy Movable állapotának megváltozásáról..
 - **-void notifyMovableDestroyed(Movable movable):** Értesíti a feliratkozókat egy Movable elpusztulásáról.
 - **-void notifyBoxPutDown(Box box, MapElement mapElement):** Értesíti a feliratkozókat egy doboz letételéről.
 - **-void notifyBoxPickedUp(Box box, MapElement mapElement):** Értesíti a feliratkozókat egy doboz felvételéről.
 - **-void notifyZPMCreated(ZPM zpm, MapElement position):** Értesíti a feliratkozókat egy ZPM létrejöttéről.
 - **-void notifyZMPickedUp(PlayerType player, ZPM zpm):** Értesíti a feliratkozókat egy ZPM felvételéről valamelyik játékos által.
 - **-void notifyGameOver(String winner):** Értesíti a feliratkozókat a játék végétől.
 - **-void assertIsOnUIThread():** Ellenőrzi, hogy melyik szálon futtatják és naplózza, ha nem a UI threaden (debuggolás céljából hasznos).

11.3.3 CreditsWindow

- **Felelősség**

Ha a felhasználó a játék főmenüjében rákattint a "Credits" felíratra, akkor ez az osztály valósítja meg azt, hogy ezen az ablakon mi jelenjen meg és hogyan.

- **Ősosztályok**
JFrame
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **+float imageAlpha**: A felületen lévő kép átlátszósága.
 - **+Timer timer**: Az animációhoz használt időzítő.
 - **Metódusok**
 - **-Dimension getPreferredSize()**: Visszaadja az ablak méreteit.

11.3.4 GamePanel

- **Felelősség**
Az új játék képernyőjén lévő középső panel kirajzolásáért felel; ezen van a térkép az egyes elemeivel
- **Ősosztályok**
 - JPanel
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **+MapView mapView**: A komponens ami kirajzolja a térképet.
 - **+MovableView movableView**: A komponens ami kirajzolj egy mozgó objektumot.
 - **+GameStateView gameStateView**: Ezen keresztül értesülünk a játék eseményeiről.
- **Metódusok**
 - **-void initialize()**: Inicializálja az osztály változóit.
 - **-void paintComponent(Graphics g)**: Kirajzolja a térképet, ZPM-eket és minden mozgó objektumot a panelra.
 - **-Dimension getPreferredSize()**: Visszaadja az ablak méreteit.

11.3.5 GameStateView

- **Felelősség**
ZPM-ek megjelenítéséért felelős osztály.
- **Ősosztályok**
Nincs ősosztálya.
- **Interfészek**
 - IZPMEventListener

- **Attribútumok**
 - **#Container parent**: a MapView konténere
 - **-Map<ZPM, UIElement> zpms**: az aktuális játékhoz tartozó UIElement-ek halmaza
- **Metódusok**
 - **+void drawZPMs(Graphics g)**: kirajzolja a ZPM-eket a felhasználói felületre
 - **+void on ZPMCreated(ZPM zpm, MapElement position)**: létrehoz egy új UIElement-et a ZPM számára, és beleteszi a halmazba
 - **+void onZMPickedUp(ZPM zpm)**: törli a ZPM-et a halmazból

11.3.6 GameWindow

- **Felelősség**

Ha a felhasználó a főképernyőről egy új játékot szeretne indítani, akkor egy új ablakban jelenik meg a játék. A képernyő három részből tevődik össze, a középsőn van a labirintus, a két szélsőn pedig a két játékosnak szóló információk a játékról. Az ablak felépítése után a Controller elindítja a játékot.
- **Ősosztályok**

JPanel
- **Interfészek**

IGameEventListener
- **Attribútumok**
 - **+Map<PlayerType, SidePanel> sidePanels**: A főablakon megjelenő oldalsó panelek halmaza.
- **Metódusok**
 - **-void onGameOver(String winner)**: A játék végeztével egy felugró ablakban értesülünk az eredményről.
 - **-void onScoreIncreased(PlayerType player)**: Ha valamelyik játékos összegyűjt egy ZPM-et (tehát pontot szerez) akkor az oldalsó paneljén (játékosnak megfelelően) növeljük a szerzett pontszámok összegét eggyel.

11.3.7 InputListener

- **Felelősség**

A billentyűzetről érkező inputok feldolgozása.
- **Ősosztályok**

Nincs ősosztálya
- **Interfészek**

KeyListener

- **Attribútumok**
 - **-Controller controller:** A Controller objektum ami feldolgozza a beérkező utasításokat.
- **Metódusok**
 - **-void keyPressed(KeyEvent e):** O'Neill és Jaffa irányításához a lenyomott billentyűk feldolgozása és a Controllernek való átadása. Az egyes esetek: O'Neill: W (északi irány); A (nyugati irány); S (déli irány); D (keleti irány); SPACE (doboz felvétel és letétel); F (sárga színű lövedék lövése); G (kék színű lövedék lövése) Jaffa: ↑ (északi irány); ← (nyugati irány); ↓ (déli irány); → (keleti irány); 3 (doboz felvétel és letétel); 1 (piros színű lövedék lövése); 2 (zöld színű lövedék lövése)

11.3.8 MainWindow

- **Felelősség**

A játék elindulása után a főképernyő kirajzolása az elemeivel (gombok), valamint az egyes gombokra való kattintás után a gomboknak megfelelő utasítás megvalósítása; új játék (New game) vagy a készítőк megtekintése (Credits).
- **Ősosztályok**

JFrame
- **Interfészek**

Nem valósít meg interfészt.
- **Attribútumok**

Nincs attribútuma.
- **Metódusok**
 - **-Dimension getPreferredSize():** Visszaadja az ablak méreteit.

11.3.9 MapBuilder

- **Felelősség**

A pálya beolvasása, a megfelelő MapElement példányok létrehozása és a szomszédos pályaelemek összekapcsolása.
- **Ősosztályok**

Nincs ősosztálya
- **Interfészek**

Nem valósít meg interfészt.
- **Attribútumok**
 - **+HashSet<ZPM> zpmSet:** A labirintusban elhelyezkedő ZPM-ek halmaza.
 - **+Player oneill:** O'Neill játszható karaktere a játékban.
 - **+Player jaffa:** Jaffa játszható karaktere a játékban.
 - **+Replicator replicator:** A replikátor karakter a játékban.

- **Metódusok**

- **-void buildMapFromFile(String mapFilePath):** A res mappában található térképet beolvasva összeállítja a térképet. Ennek során először a fájlt feldaraboljuk és az egyes MapElementeket egy átmeneti mátrixban tároljuk (egyelőre stringként). Ezután a mátrix minden elemén végig megyünk, és kialakítjuk a kapcsolatokat az újonnan létrehozott MapElementek között (szomszédok, ajtó-mérleg összerendelés). Végül elhelyezzük a pályán a játékosokat, a Replikátort, a dobozokat és a ZPM-eket.
- **+Controller createController(Game game):** Létrehozza a Controllert az aktuális játékhoz.

11.3.10 MapView

- **Felelősség**

Felelőssége a pálya és elemeinek kirajzolása a GamePanelen.

- **Ősosztályok**

Nincs ősosztálya.

- **Interfészek**

- IMapEventListener

- **Attribútumok**

- **#Container parent:** a MapView konténere
- **-Map<MapElement, UIElement> fields:** az aktuális pályához tartozó UIElement-ek halmaza
- **-Map<ProjectileType, UIElement> stargates:** csillagkapukhoz tartozó UIElementek halmaza

- **Metódusok**

- **+void drawMap(Graphics g):** kirajzolja a pályát
- **+void onMapElementCreate(MapElement mapElement):** létrehoz egy új UIElement-et és eltárolja a halmazban
- **+void onMapElementRemoved(MapElement mapElement):** törli az UIElement-et a halmazból
- **+void onStargateOpened(MapElement mapElement, ProjectileType type, Direction direction):** létrehoz egy új RotatableElement-et a csillagkapunak és eltárolja azt a halmazban
- **+onDoorStateChanged(Door door):** frissíti az ajtóhoz tartozó UIElement-et

11.3.11 MovableView

- **Felelősség**

A mozgatható elemek megjelenítéséért felelős osztály

- **Ősosztályok**

Nincs ősosztálya.

- **Interfészek**
 - `IMovableEventListener`
- **Attribútumok**
 - **#Container parent**: a `GameObjectView` konténere
 - **-Map<Movable, UIElement> movables**: az aktuális játékhoz tartozóan, a `movables`-hez tartozó `UIElement`-ek halmaza
 - **-Map<MapElement, StackableElement> boxes**: az aktuális játékhoz tartozóan, a dobozokhoz tartozó `UIElement`-ek halmaza
- **Metódusok**
 - **+void drawMovables(Graphics g)**: a paraméterül objektum segítségével felrajzolja a `movables`-eket.
 - **+void onMovableChanged(Movable movable)**: létrehoz egy új `UIElement`-et és eltárolja azt a halmazban. Az újonnan létrehozott `UIElement` fogja jelképezni a megváltozott játékelemet. Eltárolja az új `UIElement`-et, és értesíti a `GamePanel`-t, hogy a tartalma megváltozott, újrarajzolás szükséges.
 - **+void onMovableDestroyed(Movable movable)**: törli az `UIElement`-et a halmazból, értesíti a `GamePanel`-t, hogy a tartalma megváltozott, újrarajzolás szükséges.
 - **+void onBoxPutDown(Box box MapElement mapElement)**: ha nincs doboz a `MapElement`-en, létrehoz egy új `StackableElement`-et és eltárolja azt a halmazban. Máskülönben csak növeli eggyel a már létező elem `stack`-jének számát. Értesíti a `GamePanel`-t, hogy a tartalma megváltozott, újrarajzolás szükséges.
 - **+void onBoxPickedUp(Box box, MapElement mapElement)**: csökkenti a az adott `MapElement`-hez tartozó `StackableElement` `stack`-jének számát, amennyiben a `stack` mérete eléri a 0-t, törli az elemet a halmazból. Értesíti a `GamePanel`-t, hogy a tartalma megváltozott, újrarajzolás szükséges.

11.3.12 UIElement

- **Felelősség**

A felhasználói felület elemeinek függvényeit összefoglaló osztály.
- **Ősosztályok**

Nincs ősosztály.
- **Interfészek**

Nem valósít meg interfészt.
- **Attribútumok**
 - **#Image image**: a játék elemeihez kapcsolódó képek
 - **#Coord position**: a játék eleminek pozíciója a képernyőn
- **Metódusok**
 - **+void draw(Graphics g)**: kirajzolja a felhasználói felületre a `MapElement`-hez tartozó képet.

11.3.13 UIUtility

- **Felelősség**
A felhasználói felülethez kapcsolódó segéd függvényeket összefogó osztály.
- **Ősosztályok**
Nincs ősosztálya
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **+int scale**: A térkép egy egységének mérete pixelben. Alapértéke 32, a változó *final*.
- **Metódusok**
 - **-Image getImage(Object gameObject)**: Visszaad egy képet ami egy objektumot reprezentál a felhasználói felületen.
 - **-Image getImage(String relPath)**: Visszaad egy képet ami egy objektumot reprezentál a felhasználói felületen.
 - **-int getScale()**: Getter a scale attribútumnak.
 - **-int getProjectileDelay()**: Visszaadja, hogy mennyi idő telik el amíg egy lövedék az egyik mezőről a másikra ér ms-ban.
 - **-int getReplicatorDelay()**: Visszaadja, hogy mennyi idő telik el amíg a replikátor az egyik mezőről a másikra ér ms-ban.
 - **-int int getRotationAngle(Direction dir)**: Visszaadja, hogy mekkora szöggel kell elforgatni valamit, hogy a játékban megfelelő irányba nézzen.
 - **-Font getBoxCountFont()**: Visszaadja, hogy milyen betűtípust használjon az egymáson elhelyezkedő dobozok darabszámának megjelenítéséhez.
 - **-Font getMenuButtonFont()**: Visszaadja, hogy milyen betűtípust használjon a menü megjelenítéséhez.

11.3.14 RotatableElement

- **Felelősség**
Az elforgatható elemekért felelős osztály. A kapott képből létrehoz egy BufferedImage-t és elforgatja azt a kapott forgatási szöggel, majd eltárolja az elforgatott képet.
- **Ősosztályok**
 - UIElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Nincs attribútuma.

- **Metódusok**
Nincs metódusa.

11.3.15 SidePanel

- **Felelősség**
Az új játék képernyőjén elhelyezkedő két szélső panel kirajzolásáért felelős osztály. A két oldalpanel két különböző játékosnak szól, így más van a két panelen.
- **Ősosztályok**
 - JPanel
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **+PlayerType playerType**: A játékos akire vonatkozik az adott panel
 - **+Image backgroundImage**: A panel háttérében lévő kép.
 - **+int score**: Az eddig elért pontszám.
- **Metódusok**
 - **-void incrementScore()**: Növeljük a pontszámot és újrarajzoljuk.
 - **-void paintComponent(Graphics g)**: A háttérkép és a pontszám kirajzolásáért felelős függvény.
 - **-Dimension getPreferredSize()**: Visszaadja az ablak méreteit.

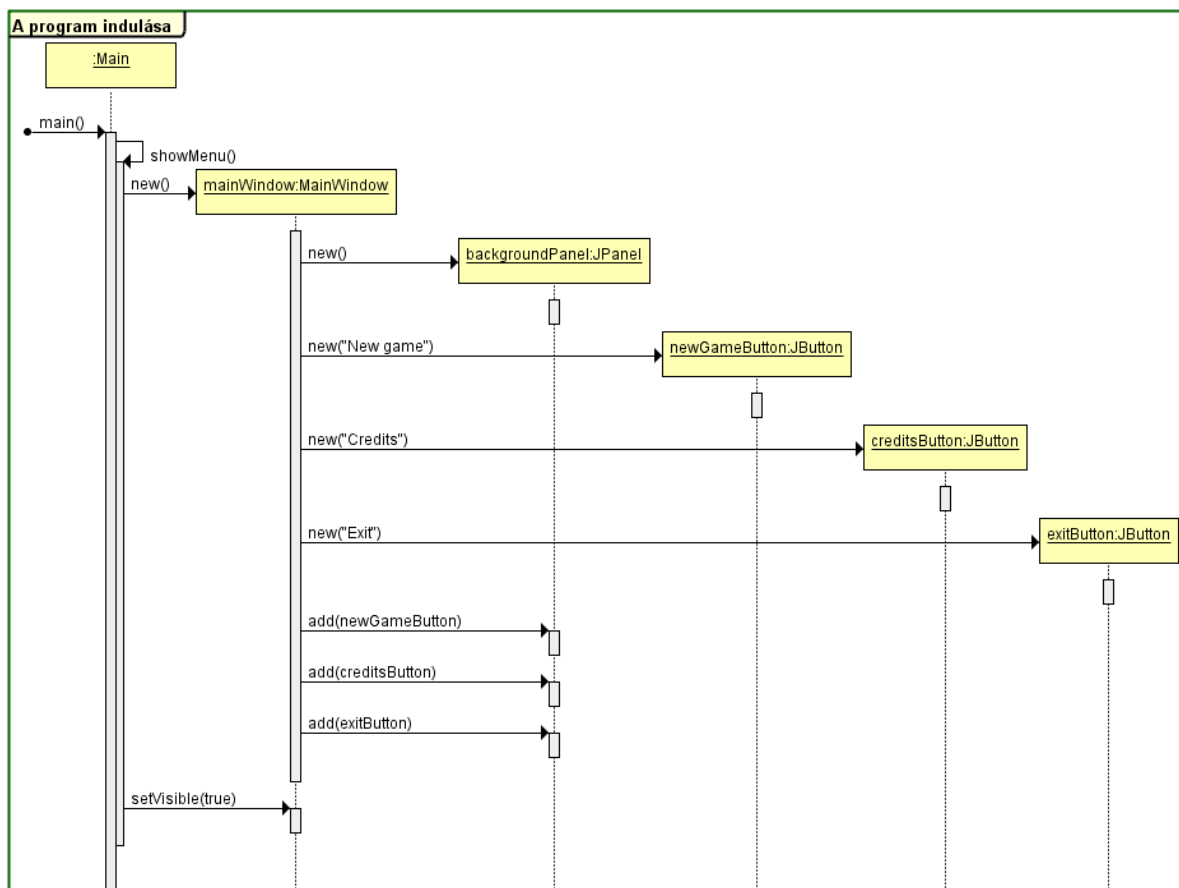
11.3.16 StackableElement

- **Felelősség**
A stackbe pakolható MapElementek megjelenítéséért felelős osztály.
- **Ősosztályok**
UIElement
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
 - **-int numberOfStacks**: a stackek száma
- **Metódusok**
 - **+int getStacks()**: visszaadja a numberOfStacks változó értékét
 - **+void incrementStacks()**: növeli a numberOfStacks változó értékét eggyel
 - **+void decrementStacks()**: csökkenti a numberOfStacks változó értékét eggyel
 - **+void draw(Graphics g)**: kirajzolja a felhasználói felületre a MapElementhez tartozó képet, amennyiben több azonos típusú objektum található egy mezőn, azt a képre írt számmal jelzi

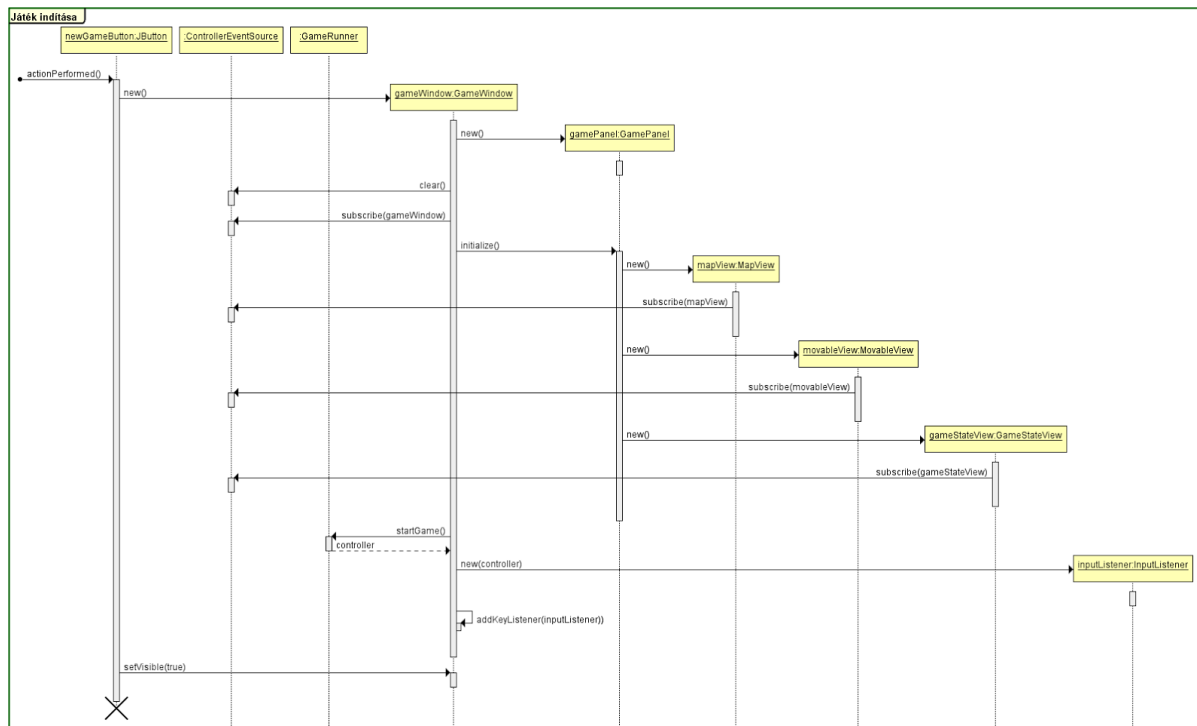
11.4 Kapcsolat az alkalmazói rendszerrel

11.4.1 Létrehozást bemutató diagramok

11.4.1.1 A program indulása

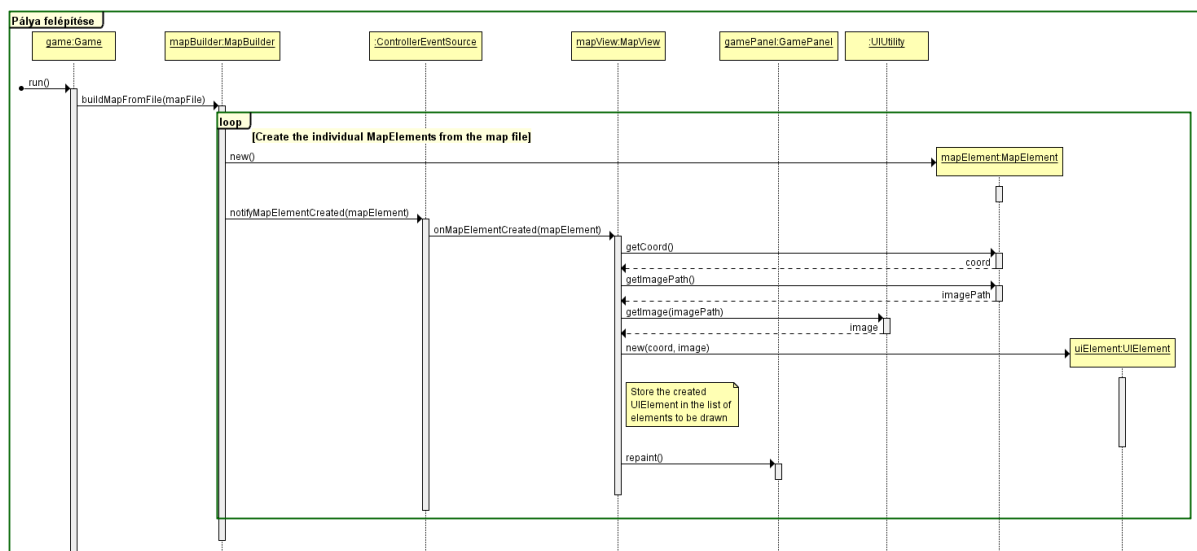


11.4.1.2 Játék indítása

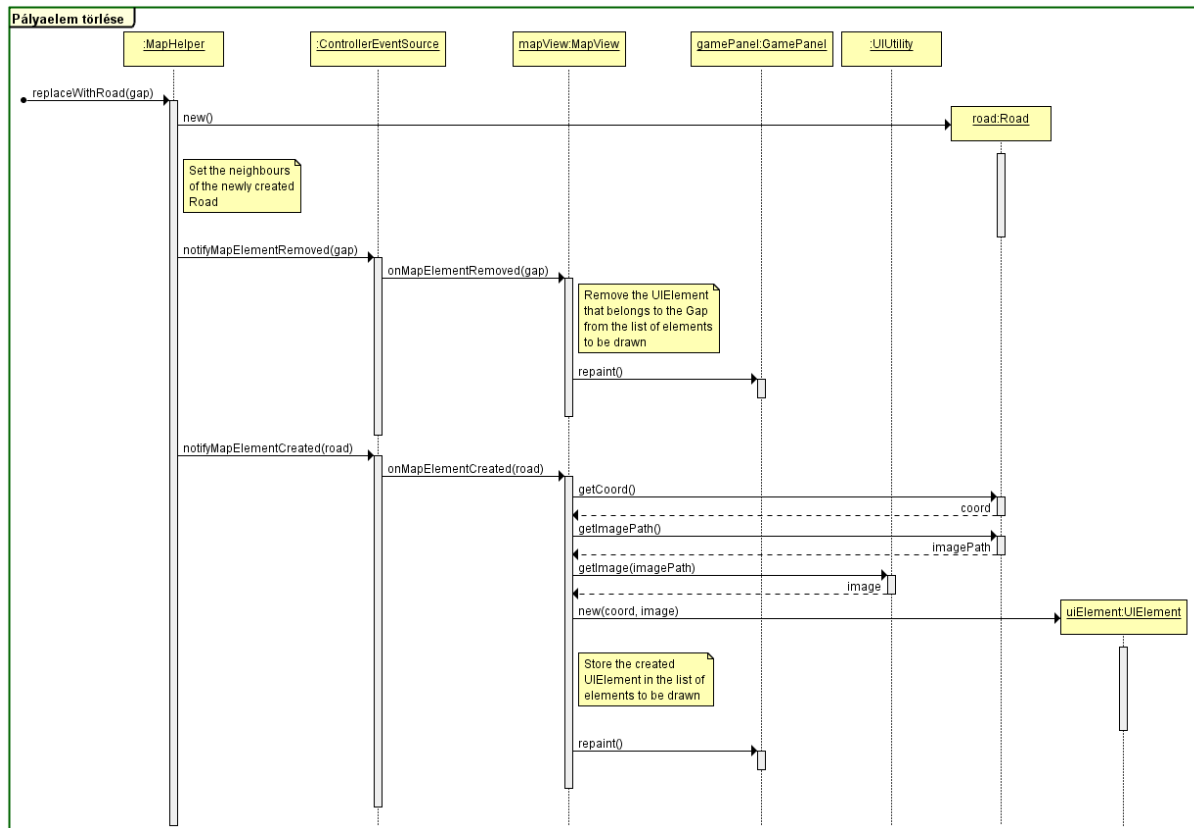


11.4.2 Értesítéseket bemutató diagramok

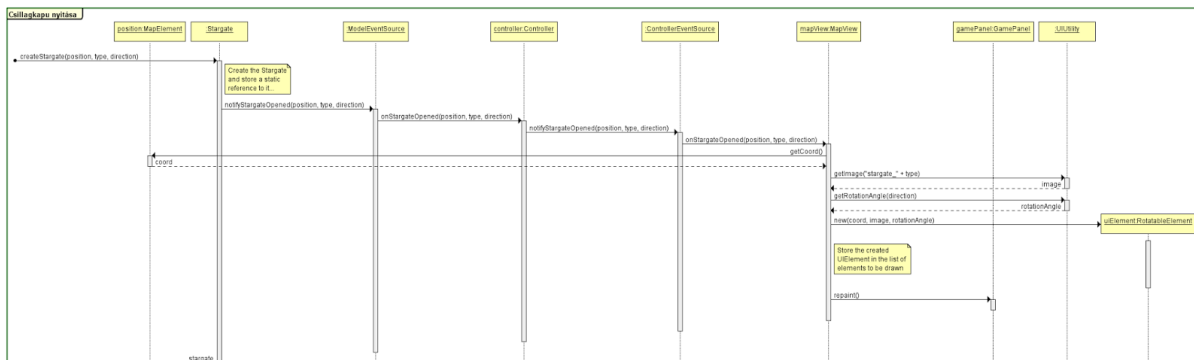
11.4.2.1 Pálya felépítése



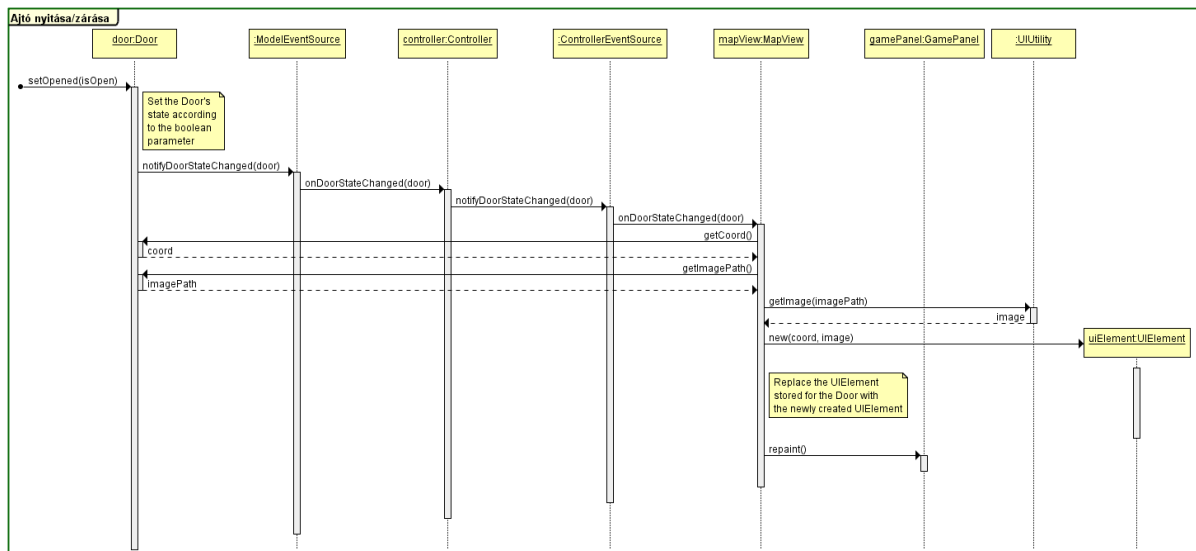
11.4.2.2 Pályaelem törlése



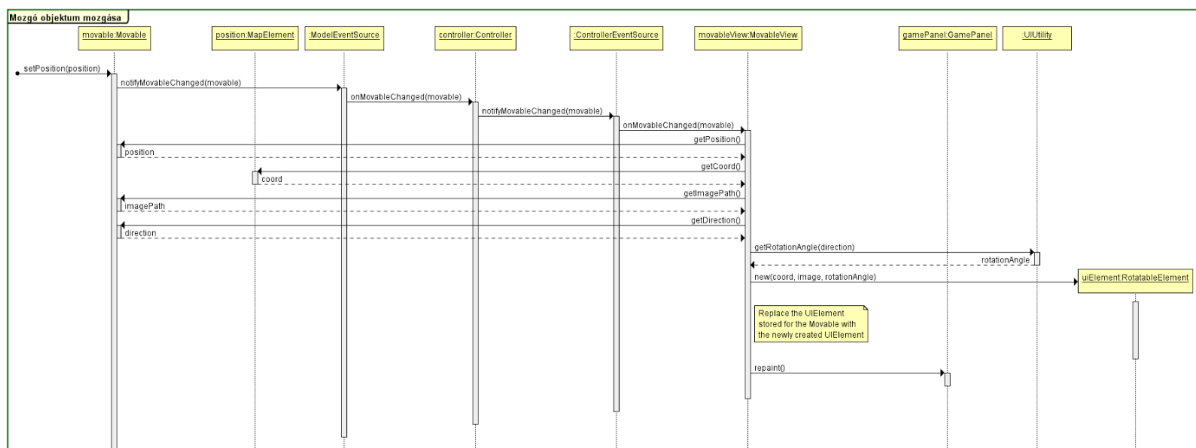
11.4.2.3 Csillagkapu nyitása



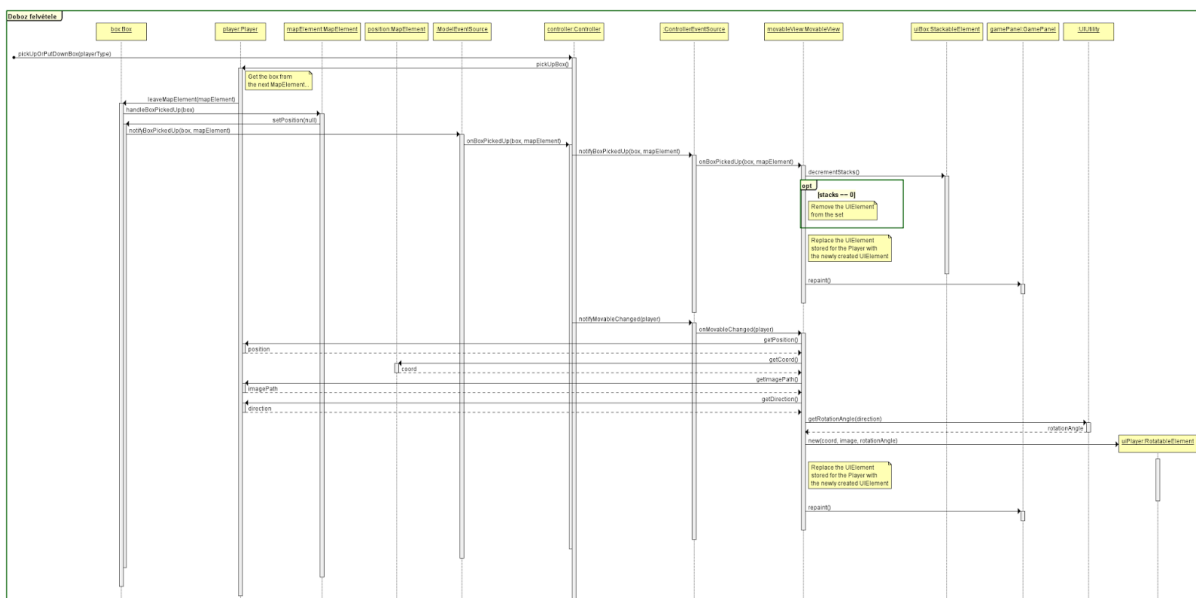
11.4.2.4 Ajtó nyitása/zárása



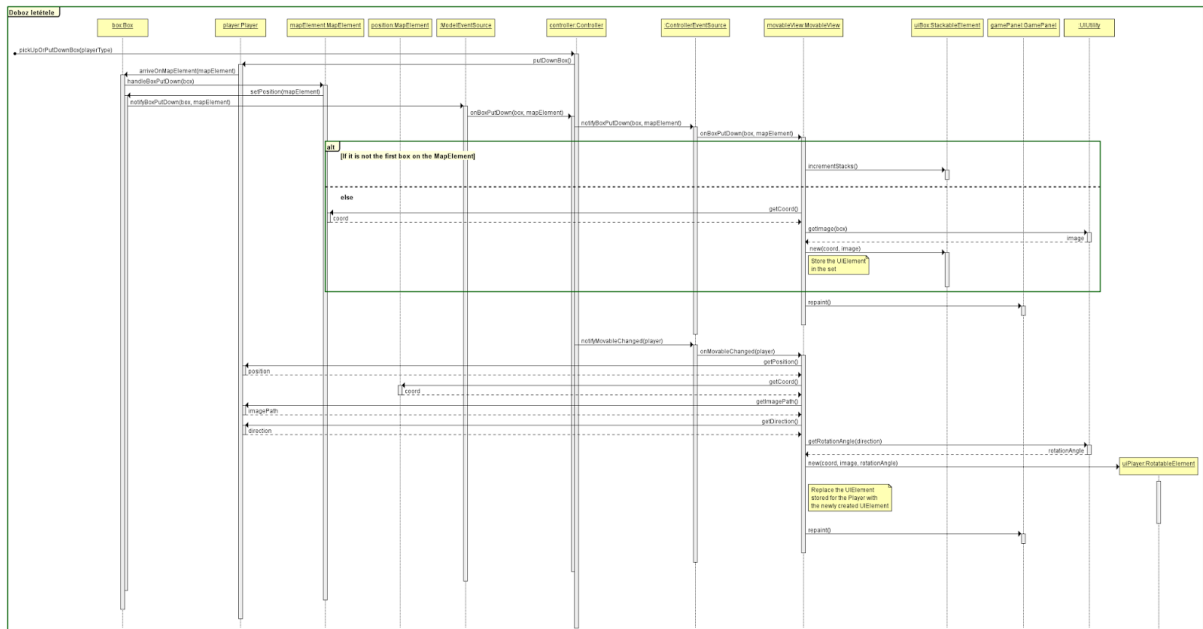
11.4.2.5 Mozgó objektum mozgatása



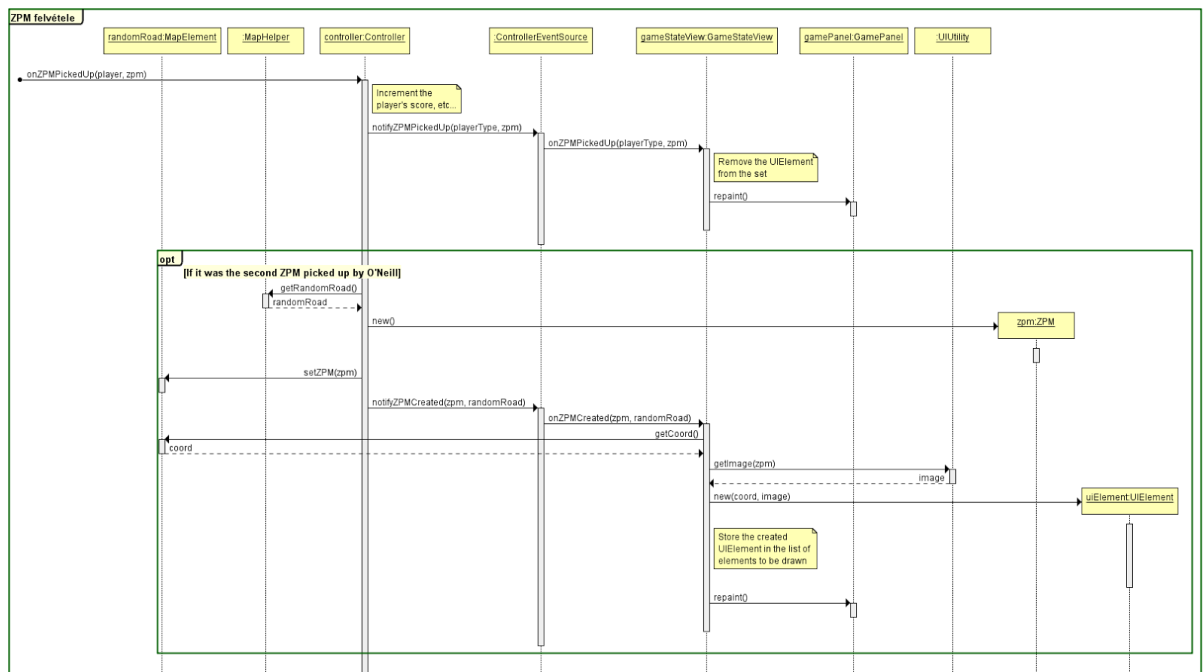
11.4.2.6 Doboz felvétele



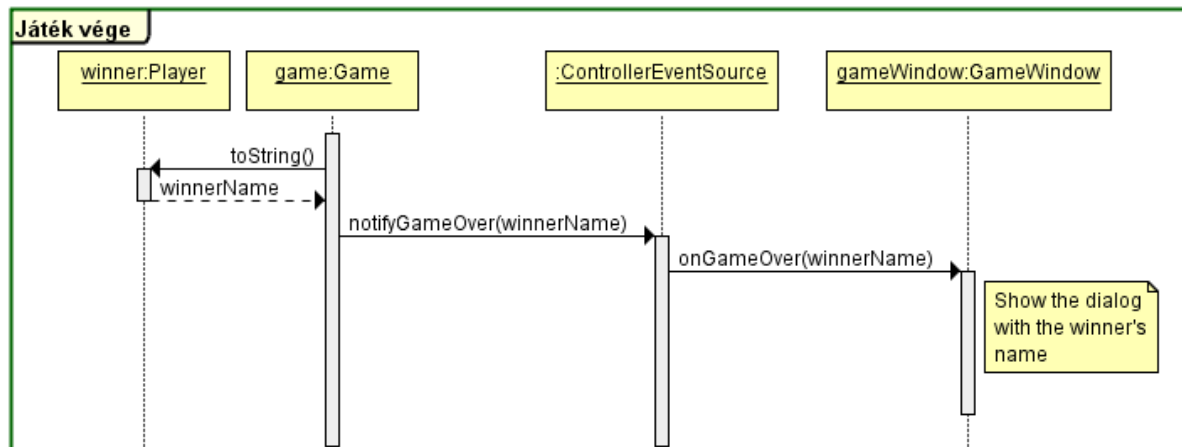
11.4.2.7 Doboz letétele



11.4.2.8 ZPM felvétele

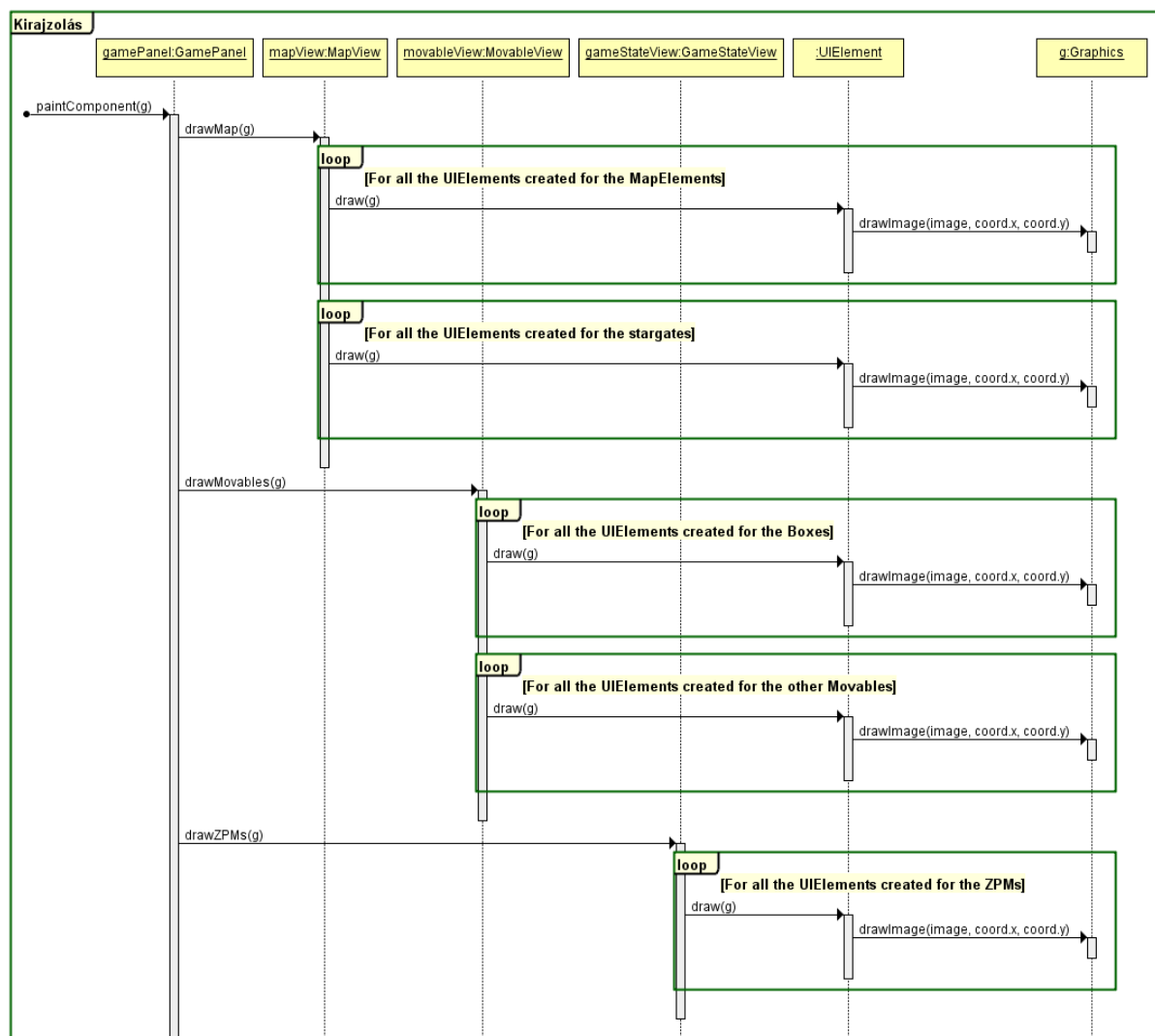


11.4.2.9 Játék vége



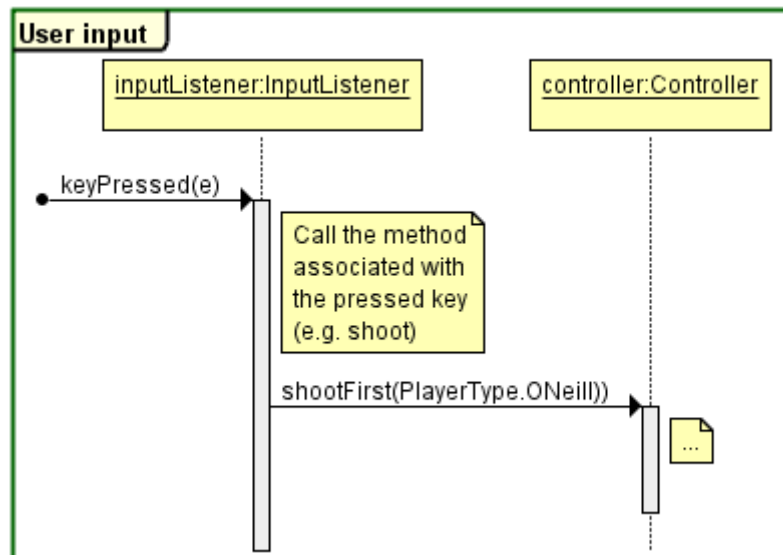
11.4.3 11.4.3. Kirajzolást bemutató diagram

11.4.3.1 Kirajzolás



11.4.4 A felhasználói input kezelését bemutató diagram

11.4.4.1 User input



13 Grafikus változat beadása

13.1 Fordítási és futtatási útmutató

13.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
.\src\controller\Controller.java	9519 byte	2016.03.27 16:38	Controller osztály.
.\src\controller\Game.java	3014 byte	2016.03.24 13:20	Game osztály. Számon tartja a játékosok pontszámát.
.\src\controller\GameRunner.java	617 byte	2016.03.27 17:21	GameRunner osztály. Elindítja a játékot.
.\src\controller\MapBuilder.java	9253 byte	2016.03.24 13:18	MapBuilder osztály. Felépíti a pályát, létrehozza és elhelyezi rajta a játékost, a ZPM-et és a dobozt.
.\src\controller\MapHelper.java	1487 byte	2016.04.13 15:42	Segédosztály a szakadék úttá építéséhez, illetve egy véletlenszerű út visszaadásához.
.\src\controller\PlayerType.java	324 byte	2016.04.13 16:16	Játékosok enumerációja.
.\src\controller\events\ControllerEventSource.java	5251 byte	2016.04.27 15:50	GUI: Eventek kezelése
.\src\controller\events\IGameEventListener.java	282 byte	2016.04.28 11:05	Játékosok pontjainak kijelzése a GUI-n
.\src\controller\events\IMapEventListener.java	644 byte	2016.04.27 15:51	GUI: MapElementek változásait kezelő eventek
.\src\controller\events\IMovableEventListener.java	564 byte	2016.04.27 15:51	GUI: Moveable-ök változásait kezelő eventek
.\src\controller\events\IZPMEEventListener.java	312 byte	2016.04.27 15:51	GUI: ZPMeket kezelő eventek
.\src\debug\GameLogger.java	1686 byte	2016.03.27 16:53	Logolást végző osztály.
.\src\debug\RandomGenerator.java	758 byte	2016.04.22 18:39	A véletlenszerűség kezeléséért felelő osztály.
.\src\debug\UILogger.java	631 byte	2016.04.27 18:55	Logolás
.\src\gamemodel\Box.java	1393 byte	2016.03.24 13:09	Box osztály.
.\src\gamemodel\Coord.java	364 byte	2016.04.27 15:45	Koordináta osztály.
.\src\gamemodel\Direction.java	705 byte	2016.03.24 12:54	Direction osztály.
.\src\gamemodel\Door.java	3463 byte	2016.03.24 13:37	Door osztály.
.\src\gamemodel\Gap.java	1502 byte	2016.03.24 13:28	Gap osztály.
.\src\gamemodel\Jaffa.java	264 byte	2016.04.13 14:01	Jaffa osztály.

.src\gamemodel\MapElement.java	3790 byte	2016.03.24 12:47	MapElement osztály.
.src\gamemodel\Movable.java	1130 byte	2016.03.24 12:55	Moveable osztály.
.src\gamemodel\ONeill.java	268 byte	2016.04.13 13:58	ONeill osztály.
.src\gamemodel\Player.java	4390 byte	2016.03.24 12:58	Player osztály.
.src\gamemodel\Projectile.java	2542 byte	2016.03.24 13:08	Projectile osztály.
.src\gamemodel\ProjectileType.java	333 byte	2016.03.24 13:01	ProjectileType osztály.
.src\gamemodel\Replicator.java	2164 byte	2016.04.13 15:32	Replicator osztály.
.src\gamemodel\Road.java	2097 byte	2016.03.24 13:35	Road osztály.
.src\gamemodel\Scale.java	4426 byte	2016.03.24 13:29	Scale osztály.
.src\gamemodel\SpecialWall.java	4700 byte	2016.03.24 13:24	SpecialWall osztály.
.src\gamemodel\Stargate.java	2029 byte	2016.03.24 13:15	Stargate osztály.
.src\gamemodel\Wall.java	373 byte	2016.03.24 13:32	Wall osztály.
.src\gamemodel\ZPM.java	389 byte	2016.03.24 13:37	ZPM osztály.
.src\gamemodel\events\IModelEventListener.java	1339 byte	2016.04.28 15:34	IModelEventListener osztály.
.src\gamemodel\events\ModelEventSource.java	3134 byte	2016.03.28 3:58	ModelEventSource.
.src\userinterface\InputListener.java	2340 byte	2016.04.28 10:00	InputListener osztály.
.src\userinterface\Main.java	1134 byte	2016.03.24 13:40	Main osztály. Menü. A program belépési pontja
.src\userinterface\UIUtility.java	2307 byte	2016.04.28 9:14	UIUtility osztály.
.src\userinterface\containers\CreditsWindow.java	1980 byte	2016.04.28 22:28	CreditsWindow osztály.
.src\userinterface\containers\GamePanel.java	1123 byte	2016.04.27 16:01	GamePanel osztály.
.src\userinterface\containers\GameWindow.java	2658 byte	2016.04.27 15:59	GameWindow osztály.
.src\userinterface\containers\MainWindow.java	3430 byte	2016.04.28 19:14	MainWindow osztály.
.src\userinterface\containers\SidePanel.java	1406 byte	2016.04.27 16:01	SidePanel osztály.
.src\userinterface\elements\RotatableElement.java	1050 byte	2016.04.28 11:39	Rotatable Element osztály.
.src\userinterface\elements\StackableElement.java	1305 byte	2016.04.28 15:11	StackableElement osztály.
.src\userinterface\elements\UIElement.java	942 byte	2016.04.27 17:32	UIElement osztály.
.src\userinterface\viewobjects\GameStateView.java	1636 byte	2016.04.28 10:35	GameStateView osztály.
.src\userinterface\viewobjects\MapView.java	3006 byte	2016.04.27 16:10	MapView osztály.
.src\userinterface\viewobjects\MovableView.java	3767 byte	2016.04.27 17:37	MovableView osztály.

.\res\maps\GameMap 1.txt	784 byte	2016.05.16 14:56	1. pálya
.\res\maps\GameMap 2.txt	770 byte	2016.05.16 14:56	2. pálya
.\res\maps\GameMap 3.txt	794 byte	2016.05.16 14:56	3. pálya
.\res\maps\GameMap 4.txt	762 byte	2016.05.16 14:56	4. pálya
.\res\images\credits.png	548461 byte	2016.04.28 22:34	Fejlesztő csapat képe
.\res\images\jaffa_static.png	269203 byte	2016.04.28 18:09	A Jaffa oldalt lévő képe
.\res\images\LZ.png	30092 byte	2016.04.29 0:04	Tanár úr
.\res\images\mainmenu.png	394860 byte	2016.04.28 18:02	Főmenü háttere
.\res\images\oneill_static.png	214245 byte	2016.04.28 18:09	O'Neill oldalt lévő képe
.\res\images\gameobjects\box.png	1020 byte	2016.04.27 19:17	Box képe
.\res\images\gameobjects\door_closed.png	315 byte	2016.04.28 15:26	Zárt Door képe
.\res\images\gameobjects\door_open.png	412 byte	2016.04.28 15:29	Nyitott Door képe
.\res\images\gameobjects\gap.png	3208 byte	2016.04.27 19:33	Gap képe
.\res\images\gameobjects\jaffa.png	2014 byte	2016.04.28 21:47	A Jaffa képe
.\res\images\gameobjects\jaffa_box.png	2080 byte	2016.04.29 13:53	A Jaffa dobozal a kezében kép
.\res\images\gameobjects\oneill.png	1807 byte	2016.04.28 21:47	O'Neill képe
.\res\images\gameobjects\oneill_box.png	2049 byte	2016.04.29 13:55	O'Neill dobozzal a kezében kép
.\res\images\gameobjects\projectile_blue.png	1227 byte	2016.04.28 13:24	Kék lövedék képe
.\res\images\gameobjects\projectile_green.png	1368 byte	2016.04.28 13:28	Zöld lövedék képe
.\res\images\gameobjects\projectile_red.png	1074 byte	2016.04.28 13:28	Piros lövedék képe
.\res\images\gameobjects\projectile_yellow.png	1373 byte	2016.04.28 13:27	Sárga lövedék képe
.\res\images\gameobjects\replicator.png	549 byte	2016.04.28 9:24	Replicator képe
.\res\images\gameobjects\road.png	168 byte	2016.04.27 19:29	Road képe
.\res\images\gameobjects\scale.png	1727 byte	2016.04.27 19:35	Scale képe
.\res\images\gameobjects\specialwall.png	441 byte	2016.04.27 19:31	SpecialWall képe
.\res\images\gameobjects\stargate_blue.png	623 byte	2016.04.28 12:58	Kék porál képe
.\res\images\gameobjects\stargate_green.png	620 byte	2016.04.28 13:07	Zöld portál képe
.\res\images\gameobjects\stargate_red.png	537 byte	2016.04.28 13:08	Piros portál képe
.\res\images\gameobjects\stargate_yellow.png	538 byte	2016.04.28 13:10	Sárga portál képe

.\res\images\gameobjects\wall.png	461 byte	2016.04.27 19:30	Wall képe
.\res\images\gameobjects\zpm.png	629 byte	2016.04.28 10:44	ZPM képe
.\compile.bat	1769 byte	2016.03.29 2:46	Fordítást végző batch fájl
.\run.bat	255 byte	2016.03.29 3:17	Futtatást végző batch fájl.

13.1.2 Fordítás és telepítés

A beadott könyvtárstruktúra gyökerében található *compile.bat* szkriptet futtatva jönnek létre a class fájlok és másolódnak be a resource-ok a binárisok mellé.

13.1.3 Futtatás

A beadott könyvtárstruktúra gyökerében található *run.bat* szkriptet futtatva indítható el a program (természetesen csak sikeres fordítás után).

13.2 Értékelés

Tag neve	Munka százalékban
Csóka Máté	18
Rosner Roland	18
Sándor Mátyás Márton	28
Szalisznyó Ivett	18
Vándor Dávid	18