# Követelmény, projekt, funkcionalitás

## Bevezetés

### Cél

A dokumentum célja a feladatkiírás szerinti „Csillagkapu” logikai játék specifikálása, munkafolyamatainak leírása. Későbbiekben a program tervének, megvalósításának és tesztelésének dokumentálása.

### Szakterület

A szoftver tulajdonképpen egy játék, melynek célja a szórakoztatás. Alkalmas logikai készségek fejlesztésére is. Célközönsége az egyszerű logikai játékkedvelők, minden korosztályban.

### Definíciók, rövidítések

|  |  |
| --- | --- |
| **kifejezés** | **magyarázat** |
| bemenet | A fájlból beolvasott adatok, vagy a felhasználó által nyújtott valós idejű információ. |
| bemutatás | Magyarázathoz kötött szemléltetés, mely bizonyos működések létezését és lépéseit demonstrálja. |
| billentyűzet | Egy bemeneti eszköz, mely a felhasználó által nyújtott információ valós idejű érzékelésére alkalmas, gombok segítségével. |
| háttértár | Hosszú távú adattárolásra alkalmas eszköz. |
| HSZK | Hallgatói Számítógép Központ |
| jar fájl | Olyan fájl, ami java programozási nyelvű elemeket tartalmaz. |
| játékmotor | Olyan programrészlet, mely a játékprogram bemeneteit kiértékeli. |
| Java Runtime Environment 1.8 | Programozási eszközök, környezet alkalmazások fejlesztéséhez Java programnyelven. |
| kiértékelés | Szélsőséges értékekkel tesztelés, működés számszerű értékelése, mérése. |
| kimenet | A program által előállított, a felhasználó számára közölt adatok. |
| monitor | Vizuális kimeneti eszköz. |
| parancssori fordítás | A program futtathatóvá alakításának módja. |
| standard I/O elemek | Java Runtime Environment 1.8-ban ki- és bemenetek kezelésére használt eszközkészlet. |

### Hivatkozások

Feladatkiírás:

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

Segédlet, hasznos tanácsok:

<https://wiki.sch.bme.hu/Szoftver_projekt_laborat%C3%B3rium>

Java Runtime Environment 1.8:

<http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

### Összefoglalás

Jelen dokumentum további részeiben lehetőség nyílik a feladatkiírás szerint elkészítendő program alapvető követelményeinek megismerésére. Ezentúl megismerkedhetünk a program leendő funkcióival, use-case-eivel, valamint a megvalósító csapat felépítésével és használt eszközeivel.

## Áttekintés

### Általános áttekintés

A szoftver a legmagasabb architekturális szinten 3 alrendszerből áll: Játékmotor, Pálya és Grafikus felület.

A szoftver legfontosabb alrendszere a Játékmotor, amiben zajlik maga a játék. A felhasználó utasításainak megfelelően mozgatja *O’Neill ezredest*, azaz az egyetlen játékos által irányított karaktert. Figyeli, hogy a két különböző színű csillagkapuból mindig csak egy, a legutoljára kilőtt legyen nyitva, illetve csak a megfelelő speciális falakon jöjjön létre. Valamint hogy két csillagkapu között létrejövő féregjárat mindkét irányba átjárható legyen. Nyomon követi a kilőtt lövedékek helyzetét. Gondoskodik róla, hogy az egyes ajtók csak a hozzájuk tartozó mérlegek lenyomásával nyíljanak ki, számon tartja a dobozokat, figyeli a mozgatásukat és az esetleges megsemmisülésüket. A pályát a Pálya alrendszertől kéri le, ami egy labirintus. A Grafikus felület számára szolgáltatja a szükséges információkat a játék pillanatnyi állásának kirajzolásához.

A Pálya alrendszer a pálya fájlokból olvassa be a pálya felépítésére vonatkozó információkat.

A Grafikus felület a Játékmotor által szolgáltatott információk alapján jeleníti meg a játék pillanatnyi állását. A felhasználó a Grafikus felület segítségével tud tájékozódni a labirintusban és ezáltal tudja *O’Neill ezredest* irányítani.

Hálózati elvárások nincsenek, ugyanis a szoftver nem használ hálózatot. A háttértáron jar fájl(ok)nak. a pálya fájloknak és a grafikus felület által felhasznált képeknek kell helyet biztosítani.

### Funkciók

A feladat egy labirintus játék elkészítése, amelyben a játékos célja a labirintusban található összes ZPM modul megszerzése, amelyek szétszórva helyezkednek el a pályán. A játékos O’Neill ezredes és Jaffa karakterét irányítja, akik a Földet akarják megvédeni Anubistól a gonosz goa’uldtól. Egy ZPM modult úgy tud megszerezni a játékos, ha odaviszi az ezredest vagy Jaffát, aki ha közvetlenül a modul előtt áll, akkor fel tudja venni azt.

A labirintust és a benne haladó utakat falak határolják. Két különböző típusú fal található a pályán. Az egyik közönséges fal, amin a játékos nem tud áthatolni. A másik pedig egy speciális fal, ami rendelkezik a közönséges fal minden tulajdonságával és az teszi speciálissá, hogy létre lehet hozni rajta csillagkaput. A két faltípus szemmel is jól megkülönböztethető.

A játékos nem csak az ezredes és Jaffa mozgatására képes, hanem használhatja azok speciális fegyverét is, amellyel csillagkapukat tudnak létrehozni. A fegyverrel egyenes vonalban tudnak lőni, a lövedék pedig addig halad előre, amíg falba vagy Replikátorba nem ütközik (aki a pályán véletlenszerűen mozog, és ha szakadékba lép, betemeti azt maga után). Ha a játékos a labirintus speciális falát találja el, akkor azon létrejön egy csillagkapu. Ha nem speciális falra lő, akkor a lövedék megsemmisül, és nem hoz létre csillagkaput. A fegyver kétféle lövedéket tud kilőni, amelyek a színükben térnek el, sárga és kék, illetve piros és zöld színűek. A felhasználó dönti el, hogy milyen színű lövedéket lő. A lövedék a színének megfelelő színű csillagkaput hoz létre.

Ha mindkét színű csillagkapu létrejött, akkor a kettő között kialakul egy féregjárat, amely mindkét irányban átjárható. Egyszerre csak egy féregjárat létezhet, így ha a játékos lő, miközben egy féregjárat nyitva van, akkor a lövedék színével megegyező színű korábban létrehozott csillagkapu bezárul. Ha sikeresen megnyílt az új csillagkapu, egy új féregjárat jön létre az új kapu és a megmaradt között.

A féregjáraton nem csak a játékos tud átmenni, hanem tárgyakat is magával tud vinni. Ha a játékos egy tárgyat cipel, akkor nem tud lőni a fegyverével. A labirintusban található tárgyak közül egyedül a dobozokat lehet felvenni és hordozni. A játékos egyszerre csak egy dobozt tud mozgatni. Csak akkor tudja felvenni a dobozt, ha közvetlenül előtte áll, a mozgás során a doboz végig az ezredessel megegyező mezőkön van és le is tuja tenni maga elé.

A pályák teljesítéshez szükség van a féregjáratok használatára, mert a labirintusban különböző akadályok zárjak el a játékost a ZPM moduloktól. Ezek az akadályok többfélék lehetnek: szakadékok, falak, dobozok és ajtók.

A szakadékokon a játékos nem tud közvetlenül átkelni, és ha túlmegy a szélén, akkor beleesik és meghal. Ha egy dobozt ejt a szakadékba, ami úgy lehetséges, hogy a szakadék fölé teszi le, vagy a dobozzal a kezében esik a szakadékba, akkor a doboz megsemmisül. A szakadékok felett át lehet lőni a fegyverrel.

A labirintusban találhatók még ajtók, amelyek a játék indulásakor mindig zárva vannak és falként viselkednek, azaz nem lehet áthatolni rajtuk semmilyen módon. A zárt ajtókon nem lehet elhelyezni csillagkapukat. Minden ajtóhoz tartozik egy mérleg, amivel ki lehet nyitni, még pedig úgy, hogy a mérlegre súlyt helyezünk, és ezzel lenyomjuk. Ez a súly lehet maga a játékos, vagy pedig egy doboz. Amíg súly van a mérlegen, addig a hozzá tartozó ajtó kinyílik, ekkor át lehet menni vagy lőni rajta, viszont amit lekerül róla, az ajtó bezárul. A mérlegek nem félétlenül a hozzájuk tartozó ajtók közvetlen közelében találhatóak.

A játék minden pályája teljesíthető. Az egyes pályák nehézsége változó. A játék a legkönnyebben megoldható pályától halad a legnehezebben megoldhatóig.

A játék akkor ér véget, ha a játékos összeszedte az összes ZPM modult, ami a pályán található vagy a játékos meghal. Illetve a felhasználó bármikor bezárhatja a programot, ezzel leállítva szoftver futását.

### Felhasználók

A felhasználóval szemben nincsenek előzetes követelmények. A szoftver használata gyorsan megtanulható és a fenti szabályok és definíciók megértése után pedig élvezhető minden felhasználó számára.

### Korlátozások

* A játéknak valós idejűnek kell lennie, nem fogadható el másodperces nagyságrendű késleltetés.
* A szoftver színes grafikai felülettel fog rendelkezni, amiről egyértelműen le kell tudni olvasni a játék aktuális állását.
* A játék billentyűzet segítségével irányítható. A billentyűzetkiosztás logikus legyen, pl. nyilakkal lehessen mozgatni a játékost.
* A szoftver fejlesztését dokumentálni kell.
* „*A forrássorok legalább 20 %-ának kommentárnak kell lennie.*”
* A játékban a pályák nehézsége szintenként fog nőni, így folyamatosan egyre nagyobb kihívást állítva a játékos elé.

### Feltételezések, kapcsolatok

Feladatkiírás:

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

Nem ismert kifejezések:

<https://hu.wikipedia.org/wiki/Goa%E2%80%99uld>

<https://hu.wikipedia.org/wiki/Csillagkapu_%28telev%C3%ADzi%C3%B3s_sorozat%29>

Felhasznált eszközök:

<https://bitbucket.org/>

<https://www.sourcetreeapp.com/>

<http://www.skype.com/hu/>

<https://www.facebook.com/>

## Követelmények

### Funkcionális követelmények

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Use-case** | **Komment** |
| F1. | Játékost lehessen a billentyűzet segítségével irányítani | bemutatás | alapvető | feladatkiírás | Ezredes irányítása, Readkey | A játékosnak (O’Neill ezredesnek) irányíthatónak kell lennie, különben az egész játék értelmét veszti. |
| F2. | A játékos fel tudja szedni a ZPM-eket, ha közvetlenül előttük áll | kiértékelés | alapvető | feladatkiírás | ZPM felvétele, Ezredes irányítása | Mindenképpen alapvető fontosságú, hiszen pontosan ez a játék célja. |
| F3. | A dobozok mozgathatóak, a játékos fel tudja venni őket, majd azokat egy tetszőleges másik helyen le tudja tenni | bemutatás | fontos | feladatkiírás | Doboz mozgatása, Ezredes irányítása | Nem alapvető fontosságú, hiszen elképzelhető olyan pálya, mely ajtók kinyitása nélkül is teljesíthető. |
| F4. | A falakon a játékos ne tudjon keresztülmenni | bemutatás | fontos | feladatkiírás | Ezredes irányítása | Fontos, de nem végzetes. A játékot jelentősen megkönnyítené a hiánya. |
| F5. | A mérlegek lenyomhatóak, melyek a nyomás időtartamáig egy ajtót nyitnak | bemutatás, kiértékelés | fontos | feladatkiírás | Doboz mozgatása, Ezredes irányítása | Fontos, de ugyanúgy, mint a dobozok mozgatásánál, itt is elképzelhető olyan pálya, mely e kritérium nélkül is teljesíthető marad. |
| F6. | Egy ajtót csak és kizárólag egy mérleggel lehet kinyitni | kiértékelés | opcionális | csapat | Doboz mozgatása | A feladatkiírás nem mondta ki explicit, hogy minden ajtóhoz csak egy mérleg tartozhat. |
| F7. | A játékos a fegyverrel két különböző színű csillagkaput tud nyitni | bemutatás | alapvető | feladatkiírás | Lövés fegyverrel | Sárga és kék portál nyitható a fegyverekkel, ha a lövedékek arra alkalmas falat érnek. (zárt ajtó nem minősül annak) |
| F8. | Az ellentétes színű csillagkapuk bármelyikén áthaladva a játékos a másik csillagkapunál jelenik meg | kiértékelés | alapvető | feladatkiírás | Lövés fegyverrel, Doboz mozgatása, Ezredes irányítása | Minden féreglyuk mindkét irányban átjárható, azokon tetszőleges tárgy (doboz) átvihető. |
| F9. | Bármelyik színű csillagkapuból mindig csak és kizárólag a legfrissebben nyitott aktív | kiértékelés | alapvető | feladatkiírás | Lövés fegyverrel | Ha ez nem teljesülne, nem lehetne egyértelműen eldönteni, hogy hova kerül a játékos. |
| F10. | A szakadék nem átjárható, de át lehet lőni rajta | bemutatás | fontos | feladatkiírás | Lövés fegyverrel, Ezredes irányítása | Ha a szakadék átjárható lenne, a játék csak akkor érhetne véget, ha a játékos az összes ZPM-et összeszedte. |
| F11. | A szakadékba esett játékos meghal | kiértékelés | fontos | feladatkiírás | Ezredes irányítása | A játék így is véget érhet. |
| F12. | A szakadékba esett tárgy megsemmisül | kiértékelés | fontos | feladatkiírás, csapat | Doboz mozgatása | A szakadékba esett doboz megsemmisül, nem kerül vissza az eredeti helyére, de ettől függetlenül a pálya továbbra is teljesíthető maradhat. (nem minden esetben) |
| F13. | Új játék kezdése | bemutatás | alapvető | feladatkiírás | Új játék kezdése | Alapvető funkció a menüben. |
| F14. | Kilépési lehetőség a játékból | bemutatás | fontos | csapat | Kilépés | Nem kritikus funkció, de kétségkívül hasznos. |
| F15. | A játéknak akkor van vége, ha a játékos összegyűjti az összes ZPM modult vagy meghal | kiértékelés | alapvető | feladatkiírás | Új játék kezdése | A feladatkiírás szerinti végkifejlet. |
| F16. | Ha a lövedék doboznak ütközik, megsemmisül | kiértékelés | opcionális | csapat | Lövés fegyverrel | De ZPM modul esetén nem semmisül meg. |
| F17. | A kilőtt lövedék színével megegyező színű csillagkapu megsemmisül a fegyver elsütésének pillanatában. | kiértékelés | fontos | csapat | Lövés fegyverrel | Amennyiben létezik. |
| F18. | A játékos dobozzal a kezében nem képes a fegyverével lőni | bemutatás | opcionális | csapat | Doboz mozgatása, Lövés fegyverrel | A nem tisztázott kérdésben a csapat így döntött. |
| F19. | Lövedék korlátlan számban lőhető, de egy időben csak egy lehet mozgásban egy színből | kiértékelés | fontos | feladatkiírás | Lövés fegyverrel | Soha nem fogy ki a lövedék. |
| F20. | Minden második ezredes által felvett ZPM modul után keletkezik egy új | bemutatás | fontos | feladatkiírás | Ezredes irányítása | Jaffa esetén nem |

### Erőforrásokkal kapcsolatos követelmények

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Komment** |
| E1. | Java Runtime Environment 1.8 | bemutatás | alapvető | feladatkiírás | Mivel a szoftvernek a HSZK gépein futtathatónak kell lenniük, ezért az említettnél frissebb és régebbi verzió használata nem ajánlott. |
| E2. | Grafikus megjelenítő (monitor) | bemutatás | fontos | feladatkiírás | A játék alapvetően grafikus felületen működik, az említett eszköz nélkül a játékélmény drasztikusan lecsökken. |
| E3. | Billentyűzet | bemutatás | alapvető | csapat | Bár a feladatkiírás nem mondja ki, hogy mivel kell a játékost irányítani, a csapat a hagyományos billentyűzet mellett döntött. |

### Átadással kapcsolatos követelmények

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Komment** |
| Á1. | A dokumentáció és a programrészletek átadására az előre meghatározott időpontokban kerül sor | bemutatás | alapvető | tárgy honlap | A pontos időpontok a dokumentum végén találhatók |
| Á2. | A program parancssori fordítás után futtatható | bemutatás | alapvető | tárgy honlap | Az előző pontban meghatározott kritériumok teljesítése esetén. |

### Egyéb nem funkcionális követelmények

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Komment** |
| NF1. | Újrafelhasználhatóság, objektumorientált elkészítés a későbbi bővítések/változtatások miatt | bemutatás | fontos | csapat | Törekedjünk a kódduplikáció elkerülésére. |
| NF2. | Az adatok központi tárolása | bemutatás | fontos | csapat | Lehetővé kell tenni a játék változtatását, minimális kódolással. |
| NF3. | A végső program tesztelését a részegységek individuális, majd pedig az összetartozó részek együttes tesztelésével végezzük el | bemutatás | fontos | tárgy honlap | A hibák elkerülése miatt muszáj a lehető legrészletesebben és legalaposabban az összes részegységet letesztelni. |
| NF4. | A játék egyszerűségéből fakadóan, a felhasználónak csak alapvető felhasználói ismeretekre legyen szüksége | bemutatás | opcionális | csapat | A játékhoz nincs szükség semmilyen szakmai előképzettségre. |

## Lényeges use-case-ek

### Use-case leírások

|  |  |
| --- | --- |
| **Use-case neve** | Új játék kezdése |
| **Rövid leírás** | Új játék kezdése |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A játékos kiválasztja az új játék menüpontot.  Betöltődik az 1. pálya, az ezredes kezdő pozícióba áll. |

|  |  |
| --- | --- |
| **Use-case neve** | Kilépés |
| **Rövid leírás** | Kilépés a játékból |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A játékos kiválasztja a kilépés menüpontot, a szoftver bezáródik. |

|  |  |
| --- | --- |
| **Use-case neve** | ZPM felvétele |
| **Rövid leírás** | Az ezredes felvesz egy ZPM modult. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | Az ezredes rálép egy ZPM modul melletti mezőre, ekkor azt automatikusan begyűjti. |

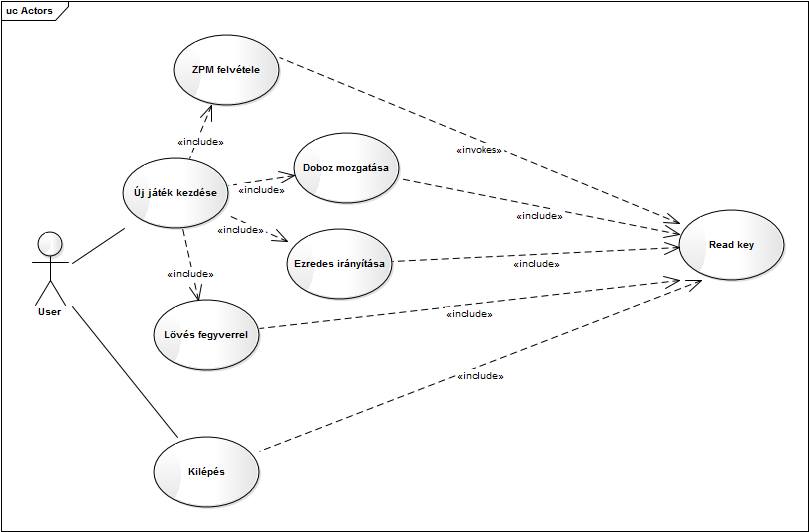
|  |  |
| --- | --- |
| **Use-case neve** | Doboz mozgatása |
| **Rövid leírás** | Az ezredes egy másik helyre helyez egy dobozt. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | Az ezredes egy doboz elé áll, és egy gomb megnyomására felveszi.  Utána a doboz és az ezredes együtt fog mozogni, e gomb újra megnyomásáig.  Ekkor leteszi maga elé az ezredes a dobozt, ami ott marad. |

|  |  |
| --- | --- |
| **Use-case neve** | Ezredes irányítása |
| **Rövid leírás** | Ezredes irányítása a pályán |
| **Aktorok** | Játékos |
| **Forgatókönyv** | Az ezredes billentyűlenyomás hatására elmozdul a pályán a megfelelő irányba. |

|  |  |
| --- | --- |
| **Use-case neve** | Lövés fegyverrel |
| **Rövid leírás** | Az ezredes féregjáratot nyit a falon keresztül a fegyverével |
| **Aktorok** | Játékos |
| **Forgatókönyv** | A játékos megnyom egy gombot, ennek hatására az ezredes lő egy lövedéket a fegyverével.   * Ha olyan falat vagy dobozt talál el, amin nem lehet féregjárat, akkor semmi nem történik. * Ha olyan falat talál el, amin lehet féregjárat   + Ha még nincs csillagkapu a falon, akkor elhelyez rajta egyet.   + Ha már meg volt egy helyen jelölve a fal egy másik színű csillagkapuval, akkor a két megjelölt fal rész között féregjárat fog keletkezni, amelyen keresztül az ezredes szabadon tud közlekedni.   + Ha már létrejött a féreg járat a régebbi csillagkapu felcserélődik azzal a csillagkapuval amelyet legutóbb hozott létre az ezredes. |

|  |  |
| --- | --- |
| **Use-case neve** | Readkey |
| **Rövid leírás** | Olvassuk a billentyűzetet. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | Standard I/O elemeket felhasználva olvassuk a billentyűzetet. |

### Use-case diagram



## Szótár

|  |  |
| --- | --- |
| **kifejezés** | **magyarázat** |
| ajtó | Egy olyan elem a pályán, melynek két állapota létezik: nyitott és zárt. Nyitott állapotban áthaladhat rajta O’Neill ezredes, a fegyvere és lövedékei, valamint dobozok is. Zárt állapotban közönséges falként működik. Az állapotát mindig pontosan egy mérleg segítségével lehet megváltoztatni. |
| alkalmas fal | lásd: speciális fal |
| Anubis | Egy gonosz goa'uld, aki a játékprogramban megvalósított történet alapján el akarja pusztítani a Földet, O’Neill ezredes főellensége. |
| átjárható | Egy pálya eleme átjárható, ha azo(ko)n a mező(kö)n, ami(ke)n az adott elem van, doboz, O’Neill ezredes és fegyvere vagy annak lövedéke is egyidejűleg helyet foglalhat. |
| csillagkapu | Egy olyan különleges elem, melyből minden időpillanatban maximum egy kék és egy sárga színű létezhet. Amennyiben csak egy csillagkapu létezik, az nem átjárható. Ha mindkét csillagkapu létrejött, a kettő között féregjárat alakul ki. |
| doboz | Egy olyan elem a pályán, melyet O’Neill ezredes, ha szomszédos mezőn áll, képes felvenni, és a pálya másik részeibe elmozdítani. |
| fal | Gyűjtőneve a közönséges és speciális falaknak. Nem átjárható elemek. |
| fegyver | O’Neill ezredes különleges eszköze, mellyel két különböző (sárga és kék) lövedékek lőhetőek. |
| féregjárat | Két különböző színű csillagkapu között kialakuló kapcsolat, mely biztosítja, hogy az egyik csillagkapuba beérkező mozgatható elemek (O’Neill ezredes, doboz) a másik csillagkapuból fognak kijönni. Ha az egyik csillagkapu megsemmisül, a féregjárat is. |
| goa'uld | Egy idegen bolygóról érkezett parazita faj, melyek legtöbbször emberi gazdatestet választanak. |
| Jaffa | A játékos által bemenetekkel irányított karakter, melynek megfelelő kezelésével teljesíthetőek a program által nyújtott logikai kihívások. |
| játékos | A játékprogramot futtató, és bemenetekkel irányító felhasználó. A kifejezést, ahol egyértelmű gyakran használjuk O’Neill ezredes helyett. |
| játékos meghal | A kezelt karakter végzetes cselekedetének eredménye, mely a játék végét eredményezi. |
| közönséges fal | Egy olyan elem a pályán, mely nem átjárható, a játékos által el nem mozdítható. |
| különleges fal | lásd: speciális fal |
| labirintus | lásd: pálya |
| lövedék | Egy olyan elem, mely a fegyvertől a választott irányban egyenesen halad az átjárható mezőkön, mindaddig, míg nem átjárhatóba ütközik. Amennyiben a nem átjárható mező speciális fal, a becsapódás helyén a lövedék színével (sárga vagy kék) egyező színű csillagkapu jön létre. Minden más esetben a lövedék megsemmisül. |
| mérleg | Egy olyan elem a pályán, melyre súlyt helyezve kinyílik pontosan egy pályán lévő ajtó, a súly mérlegről eltávolítása után pedig az ajtó visszazárul. |
| mező | A pálya olyan alapegysége, melyen vagy teljesen rajta van egy elem, vagy egyáltalán nem. |
| O’Neill ezredes | A játékos által bemenetekkel irányított karakter, melynek megfelelő kezelésével teljesíthetőek a program által nyújtott logikai kihívások. |
| pálya | A játékprogramban megvalósított logikai fejtörő, melyet a játékos O’Neill ezredes megfelelő irányításával oldhat meg. A pályán több, annak nehézségét és teljesíthetőségét befolyásoló elem található (pl.: falak, mérleg, doboz) |
| Replikátor | A gép által irányított karakter. |
| speciális fal | Egy olyan elem a pályán, mely nem átjárható, a játékos által el nem mozdítható. Ha ilyen típusú falba lövedék csapódik, csillagkapu nyílik rajta. |
| súly | Maga O’Neill ezredes vagy a pályán elhelyezett és elmozgatott doboz(ok) is szolgálhatnak súlyként. |
| szakadék | Egy olyan elem a pályán, melyre ha O’Neill ezredessel rálépünk, meghal, a pálya teljesítése nem sikerült. Amennyiben doboz kerül a szakadékkal azonos mezőre, a doboz megsemmisül. Lövedékek áthaladhatnak azokon a mezőkön, amelyen szakadék van. |
| ZPM modul | Energiaforrás, a játék célja ezek összegyűjtése. A ZPM modulok pályán elhelyezett elemek, melyek mindegyikét össze kell szednie a játékosnak a pálya teljesítéséhez. |

## Projekt terv

## Csapat

## A csapat 6 főből áll. A feladatot úgy próbáljuk felosztani, hogy lehetőleg mindenki azonos nehézségű részt kapjon, miközben az egyéni preferenciáit is betartjuk.

|  |  |
| --- | --- |
| **Név** | **Felelősségek** |
| Borbély László Benedek | kódolás, UML, teszt |
| Gáspár Vilmos | dokumentáció, UML, teszt |
| Makki Viktor | dokumentáció, kódolás, grafika |
| Papp Albert | kódolás, grafika, teszt |
| Ritter Alex | UML, kódolás, grafika |
| Zöld Máté | dokumentáció, UML, kódolás |

## Kommunikáció

## Verziókezelés: Mivel ezen a projekten többen dolgozunk, a dokumentumokat és programkódokat meg kell osztanunk egymással. Nem jó, ha valakinél még régebbi változat vagy hiányos fájlok vannak. Hogy minden egy helyen, a legújabb verziójában legyen.

**Facebook:** A csapatnak nyitottunk egy beszélgetést ahol a sürgősebb dolgokat kommunikálhatjuk le, illetve hogy ki hogy áll az adott feladatával.

**Megbeszélések:** Hetente kétszer tartunk megbeszélést. Az elsőn a feladatokat osztjuk ki, másodikon meg egyeztetjük őket illetve az esetleges problémákat orvosoljuk. Ha valaki nem tud részt venni ezeken a megbeszéléseken Skype segítségével hidaljuk át a problémát

**Levelezőlista:** Mindenkit felvettünk egy levelezőlistára ahol a találkozási időpontokat, helyszíneket és a kulcsfontosságú információkat osszuk meg egymással

## Használt programok

**Verziókezelés:** SourceTree programot használunk, bitbucket központi tárhellyel.

**Dokumentáció** Microsoft Wordöt használjuk, mert mindenki ismeri a csapatból  
Az UML szerkesztésére az Enterprise Architect programot használjuk

**Fejlesztőkörnyezet** Eclipse és IntelliJ IDEA szoftverekkel dolgozunk

## Feladat lépései, határidei

|  |  |
| --- | --- |
| **Dátum** | **Leírás** |
| febr. 29. | Követelmény, projekt, funkcionalitás - beadás |
| márc. 7. | Analízis modell kidolgozása 1. - beadás |
| márc. 14. | Analízis modell kidolgozása 2. - elektronikus feltöltés. A nyomtatott változatot a márc. 16-án kell a laborvezetőnek odaadni. |
| márc. 21. | Szkeleton tervezése - beadás |
| márc. 29. | **Szkeleton** - beadás |
| ápr. 4. | Prototípus koncepciója - beadás |
| ápr. 11. | Részletes tervek - beadás |
| ápr. 25. | **Prototípus** - beadás |
| máj. 2. | Grafikus felület specifikációja - beadás |
| máj. 17. | **Grafikus változat** - beadás |
| máj. 20. | Összefoglalás - beadás |

A feladat főbb lépései (vastagon jelölt) kifejtve (*forrás: tárgyhonlap*):

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 szkeletonnak alkalmasnak kell lenni arra, hogy a különböző forgatókönyvek és szekvencia diagramok ellenőrizhetők legyenek. Csak karakteres ernyőkezelés fogadható el, mert ez biztosítja a rendszer egyszerűségét.

A **prototípu**s 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 változat tervezési szempontból elkészült, az ütemezés, az aktív objektumok kezelése megoldott. A business objektumok - a megjelenítésre vonatkozó részeket kivéve - valamennyi metódusa a végleges algoritmusokat tartalmazza. A megjelenítés és működtetés egy alfanumerikus ernyőn követhető, ugyanakkor a megjelenítés fájlban is logolható, ezzel megteremtve a rendszer tesztelésének lehetőségét. Különös figyelmet kell fordítani az interfész logikájára, felépítésére, valamint arra, hogy az mennyiben tükrözi és teszi láthatóvá a program működését, a beavatkozások hatásait.

A teljes ( **grafikus** ) változat a prototípustól elvileg csak a kezelői felület minőségében különbözhet. Ennek változatnak az értékelésekor a hangsúlyt sokkal inkább a megvalósítás belső szerkezetére, semmint a külalakra kell helyezni.

***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2016.02.24. 12:00 | 1 óra | Gáspár, Borbély | Értekezlet.  Dokumentálás alapelvei |
| 2016.02.24. 19:30 | 1,5 óra | Gáspár, Borbély, Papp, Zöld, Ritter | Verziókezelő szoftverek előkészítése,  feladatok szétosztása |
| 2016.02.26. 17:30 | 4 óra | Borbély | Követelmények meghatározása |
| 2016.02.27. 20:00 | 2,5 óra | Ritter | Projekt terv |
| 2016.02.27. 21:00 | 2,5 óra | Zöld | Áttekintés, Funkciók, Felhasználók |
| 2016.02.27. 21:00 | 3 óra | Papp | Use-casek, korlátozások |
| 2016.02.28. 9:00 | 3,5 óra | Gáspár | Cél, Szótár, definíciók kitöltése |
| 2016.02.28. 16:00 | 3 óra | Gáspár, Borbély, Papp, Zöld, Makki, Ritter | Beadás előtti értekezlet, dokumentum egységesítése |

# Analízis modell kidolgozása

## Objektum katalógus

### Abyss

A pályán található szakadék. Azok az objektumok, amelyek beleesnek, mint a játékos vagy a doboz megsemmisülnek. Tetszőlegesen sok lehet belőle a pályán.

### Box

A pályán elhelyezkedő doboz.

### BoxInventory

A pályán elhelyezkedő mozgatható objektumokat tárolja. Ilyen objektum a Box. Ezeket az objektumokat a pályán elfoglalt koordinátájukkal együtt tárolja.

### Button

A mérleget megvalósító objektum, ami egy nyomógomb. Az ajtók számának megfelelő számú gomb található a pályán. Minden Button-hoz tartozik egy Door, a Button tartja nyilván ezt a hozzárendelés. Feladata, hogy az ajtó állapotát megváltoztassa.

### ButtonHandler

Kezeli mérlegek lenyomását.

### ButtonInventory

A pályán elhelyezkedő Button-okat tárolja a koordinátájukkal együtt.

### CollectableInventory

A pályán elhelyezkedő gyűjthető objektumokat tárolja. Ilyen objektum a ZPM. Ezeket az objektumokat a pályán elfoglalt koordinátájukkal együtt tárolja.

### Coordinate

Tárolja a pozíciót.

### CVector

Tárolja a pozíciót és irányt.

### DataAccessPoint

A StepHandler, a ShotHandler és a GrabHandler számára biztosít hozzáférést a FieldObjectInventory-hoz, BoxInventory-hoz és a CollectableInventory-hoz. Egy ilyen objektum van.

### Door

Ajtó, ami a pályán található. Két állapota van. Létrejöttekor csukott állapotban van és csak a hozzátartozó gomb lenyomásakor kerül nyitott állapotba. Feladata ezen állapotok nyilvántartása, azaz, hogy tudja, hogy átjárható-e vagy nem.

### FieldObjectInventory

Tárolja azokat az objektumokat, amik a pályát alkotják. Ezek közé tartozik a Wall, a SpecialWall, az Abyss, a Way és a Door. Az objektumokat a pályán elfoglalt koordinátájukkal együtt tárolja.

### GrabHandler

Kezeli a játékos egy objektum felvétele illetve lerakását.

### Player

Nyilvántartja a játékos állapotát (hely, irány, dobozkézben).

### ShotHandler

Kezeli a játékos lövéseit, csak akkor akar Csillagkaput létrehozni, ha az eltalált mezőn létrehozható csillagkapu.

### SpecialWall

Speciális fal, amelyen létrejöhet csillagkapu. Egy ilyen falon csak egy csillagkapu lehet.

### StarGate

Nyilvántartja a csillagkapu színét illetve pozícióját.

### StarGateHandler

Megszünteti és létrehozza a csillagkapukat.

### StarGateInventory

Tárolja a csillagkapukat, kezeli a helyüket és megmondja, hogy rálépve melyik mezőre kerülünk.

### StepHandler

Kezeli a játékos lépéseit, csak akkor lép az egyik irányba, ha ott nincs olyan mező illetve objektum, ami ezt nem engedné.

### Wall

Ez az objektum a pályán elhelyezkedő egységnyi fal. Tetszőleges számú lehet belőle. Feladata, hogy nem lehet áthatolni rajta semmilyen módon, így csillagkapu sem jöhet létre rajta, mert akkor féregjárattal át lehetne menni rajta.

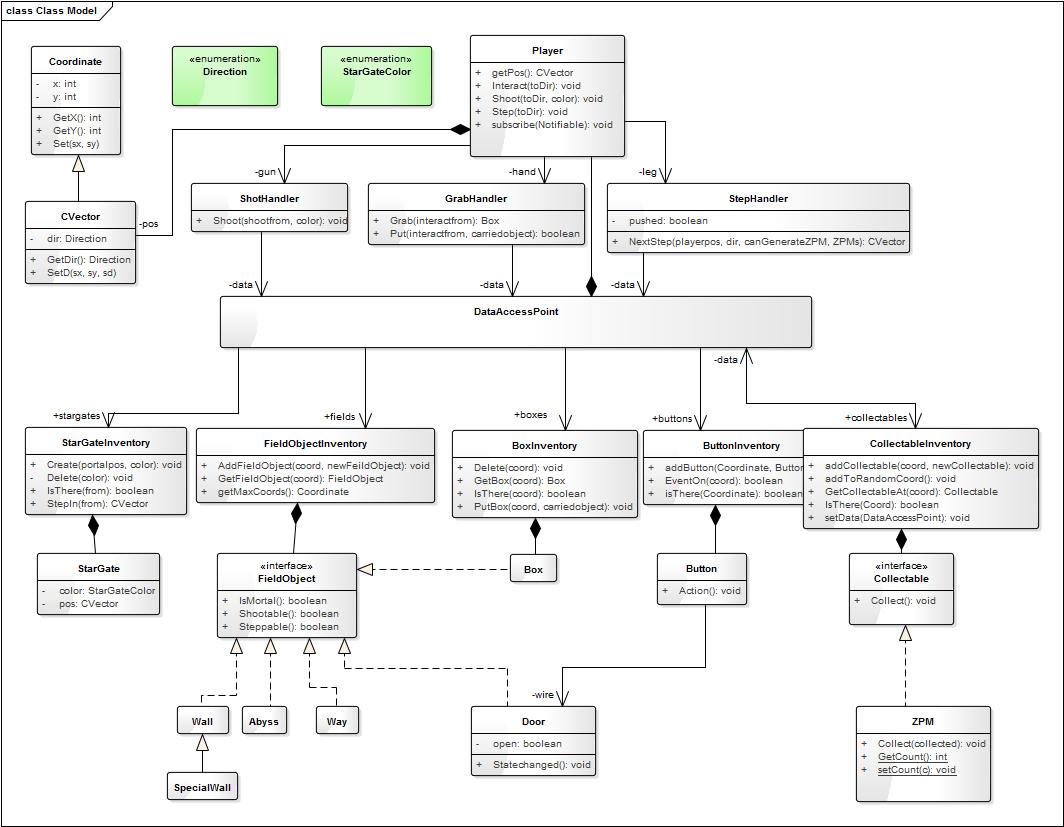
### Way

A pálya járható útja. Egyetlen feladata, hogy biztosítsa, hogy rá lehessen lépni. Szintén tetszőlegesen sok lehet belőle.

### ZPM

Összegyűjtendő objektum, amelyből tetszőleges számú lehet. A játék célja ezen objektumok összegyűjtése. Feladat, hogy biztosítsa a játékosnak, hogy feltudja venni.

## Statikus struktúra diagramok



## Osztályok leírása

### Abyss

#### Felelősség

Szakadékot megvalósító osztály. Átjárható, ezáltal át lehet lőni felett. Nem jöhet létre rajta csillagkapu. Halálos.

#### Ősosztályok: Nincs

#### Interfészek

* FieldObject

#### Attribútumok: Nincs

#### Metódusok: Nincs

### Box

#### Felelősség

A dobozt megvalósító osztály. Mozgatható, nem átjárható, nem lehet átlőni felette, nem halálos és nem jöhet létre rajta csillagkapu.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok: Nincs

#### Metódusok: Nincs

### BoxInventory

#### Felelősség

Tárolja a doboz objektumokat.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **boxes: HashMap<Coordinate, Box>**

#### Metódusok

* **void Delete(Coordinate coord)**: Törli a paraméterként átvett koordinátán található Box interface-t megvalósító objektumot.
* **Box GetBox(Coordinate coord)**: Visszaadja a paraméterként átvett koordinátán található objektumot a HashMap-ből, ha nincs azon a koordinátán objektum, akkor null-al tér vissza.
* **bool IsThere(Coordinate coord)**: Visszaadja, hogy van-e a keresett koordinátán objektum.
* **void addBox(Coordinate coord, Box newBox)**: A paraméterként átvett koordinátával elhelyezi a HashMap-ban a newBox-ban található objektumot, ha az nem null.

### Button

#### Felelősség

Megvalósítja a mérleget, ami egy nyomógomb. Feladata, hogy tudja, melyik ajtó tartozik hozzá, hogy annak meg tudja változtatni az állapotát.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **wire: Door**

#### Metódusok

* **void Action()**: Megváltoztatja a wire attribútumban referenciaként eltárolt ajtó állapotát, azzal, hogy meghívja az ajtó Statechanged() metódusát.

### ButtonHandler

#### Felelősség

*Kezeli azokat a pozíciókat ahová lépve mérlegek egyike lenyomódik.*

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **data**: DataAcessPoint: Adat-összeköttetést biztosít.

#### Metódusok

* **void EventOn(CVector coord)**: Kezeli, ha erre a pozícióra objektum kerül.

### ButtonInventory

#### Felelősség

Tárolja a pályán elhelyezkedő Button objektumokat.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **buttons: HashMap<Coordinate, Button>**

#### Metódusok

* **Button GetButtonOn(Coordinate: coord)**: A paraméterként átvett koordinátán található Buttont adja vissza, ha nincs ott Button, akkor null-al tér vissza.

### CollectableInventory

#### Felelősség

Tárolja a gyűjthető objektumokat, azaz azokat, amelyek megvalósítják a Collectable interface-t.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **col: HashMap<Coordinate, Collectable>**

#### Metódusok

* **Collectable GetCollectableAt(Coordinate coord)**: Visszaadja a paraméteként átvett koordinátán található Collectablet, ha nincs ott semmi, akkor null-t ad vissza.

### Collectable

#### Felelősség

Interface a gyűjthető osztályok számára.

#### Ősosztályok: Nincs

#### Metódusok

* **void Collect()**: Az interface-t megvalósító objektum begyűjtésekor hívódik meg.

### Coordinate

#### Felelősség

*Tárolja x, y koordinátákat.*

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **x**: int: x koordinátát tárolja.
* **y**: int: y koordinátát tárolja.

#### Metódusok

* **void Set(int sx, int sy)**: Beállítja erre a pozícióra koordinátákat.
* **int GetX()**: Visszaadja x értékét.
* **void GetY()**: Visszaadja y értékét.

### CVector

#### Felelősség

*Tárolja x, y koordinátákat, és az irányt ahova néz az adott objektum.*

#### Ősosztályok: Coordinate

#### Interfészek: Nincs

#### Attribútumok

* **dir**: Direction: Az irányt tárolja.

#### Metódusok

* **void SetDir(int sx, int sy, Direction sd)**: Beállítja erre a pozícióra koordinátákat.
* **Direction GetDir()**: Visszaadja az irányt.

### DataAccessPoint

#### Felelősség

Hozzáférés biztosítása a StepHandler, a ShotHandler és a GrabHandler számára a FieldObjectInventory-hoz, BoxInventory-hoz és a CollectableInventory-hoz.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **field**: FieldObjectInventory: Referencia a FieldObject-ek tárolójára
* **Box**: BoxInventory: Referencia a mozgatható objektumok tárolójára.
* **collectable**: CollectableInventory: Referencia gyűjthető objektumok tárolójára

#### Metódusok: Nincs

### Door

#### Felelősség

Az ajtót megvalósító osztály. Felelőssége, hogy tudja, hogy milyen állapotban van. Zárt állapotban nem lehet átmenni rajta, így átlőni sem. Nyílt állapotban pedig át lehet menni rajta és így át is lehet lőni a nyitott ajtón. Semelyik állapotában nem halálos és nem jöhet létre rajta csillagkapu.

#### Ősosztályok: Nincs

#### Interfészek

* FieldObject

#### Attribútumok

* **open: boolean**

#### Metódusok

* **void** **Statechanged()**: Megváltoztatja az ajtó állapotát. Ha nyitva volt, akkor becsukja, ha zárva volt, akkor kinyitja.

### FieldObject

#### Felelősség

Interface, amely gondoskodik arról, hogy azokról az osztályokról, amik megvalósítják, el lehessen dönteni, hogy átjárhatók-e, át lehet-e lőni felettük, létre lehet-e hozni felette csillagkaput, illetve halálosak-e.

#### Ősosztályok: Nincs

#### Metódusok

* **boolean IsMortal()**: Visszaad egy logikai értéket, hogy az objektumra lépés halálos-e. Azaz ha a játékos rálép vagy doboz kerül rá, akkor az megsemmisül-e.
* **boolean Shootable()**: Visszaad egy logikai értéke, hogy az objektumon létrejöhet-e csillagkapu.
* **boolean Steppable()**: Visszaad egy logikai értéket, hogy az objektumra rá lehet-e lépni. Azokon amikre rá lehet lépni, át is lehet lőni.

### FieldObjectInventory

#### Felelősség

FieldObject interface-et megvalósító objektumok tárolása.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **fieldobj: HashMap<Coordinate, FieldObject>**

#### Metódusok

* **FieldObject GetFieldObject(Coordinate coord)**: A paraméter listán átvett koordinátán található FieldObject-et visszaadja a HashMap-ből, ha nincs a megodott koordinátán FieldObject, akkor null-t ad vissza.

### GrabHandler

#### Felelősség

A mozgatható objektumok mozgatásáról gondoskodik.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **sgates**: StarGateInventory: Nyilvántartja a csillagkapukat és pozíciójukat.
* **data**: DataAcessPoint: Adat-összeköttetést biztosít.
* **weight:** ButtonHandler: Kezeli a gombok (mérlegek) lenyomását.

#### Metódusok

* **boolean Grab(CVector interactfrom)**: CVectorral adjuk meg, hogy honnan milyen irányból akarunk felvenni, true-t ad vissza, ha a felvétel sikerült.
* **boolean Put(CVector interactfrom)**: CVectorral adjuk meg, hogy honnan milyen irányból akarunk letenni, true-t ad vissza, ha a letételsikerült.

### Player

#### Felelősség

Tárolja a Player (O’Neill ezredes) állapotát: pozícióját, irányát és hogy van-e valami a kezében. Ezeket változtatja a megfelelő függvényekkel.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **handfull**: boolean: Van-e valami mozgatható a Player kezében.
* **pos:** CVector: Megadja a Player pozícióját és irányát.

#### Metódusok

* **CVector GetPos()**: Visszaadja a Player pozícióját, irányát.
* **void SetPos(CVector ppos)**: Megváltoztatja a Player pozícióját, irányát.
* **boolean GetInHand()**: Visszaadja a handfull attribútum értékét.
* **void SetInHand(boolean b)**: Megváltoztatja a handfull attribútum értékét.

### PlayerActionHandler

#### Felelősség

Bemenet szerint továbbítja az események vezérlését, melyek a játékoshoz köthetőek.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **leg**: StepHandler: Kezeli a Player lépéseit.
* **gun**: ShootHandler: Kezeli a Player lövését.
* **hand**: GrabHandler: Kezeli a mozgatható objektumok felszedését.
* **pushing**: ButtonHandler: Kezeli a gombok (mérlegek) lenyomását.
* **data:** Player: Tárolja a Player állapotát.

#### Metódusok

* **void Interact(Direction toDir)**: Megpróbál felvenni vagy lerakni abba az irányba.
* **void shoot(Direction toDir, StarGatColor color)**:Lő abba az irányba ilyen színű Csillagkaput.
* **void Step(Direction toDir)**: Megpróbál lépni abba az irányba.

### ShootHandler

#### Felelősség

Lekezeli a játékos lövéseit.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **sgates**: StarGateHandler: Kezeli a csillagkapukat.
* **data**: DataAcessPoint: Adat-összeköttetést biztosít.

#### Metódusok

* **void Shoot(CVector shootfrom, Direction color)**: CVectorral átadjuk a pozíciónkat és hogy onnan milyen irányból akarunk lőni. Kiszámolja a becsapódás helyét, ha shootable, vezérléstovábbadással gondoskodik az esetleges csillagkapukról.

### SpecialWall

#### Felelősség

Speciális falat megvalósító osztály. Biztosítja, hogy nem átjárható, nem lehet átlőni felette. Nem halálos. Abban különbözik a Wall-tól, hogy létrejöhet rajta csillagkapu.

#### Ősosztályok: Wall

#### Interfészek

* FieldObject

#### Attribútumok: Nincs

#### Metódusok: Nincs

### StarGate

#### Felelősség

Maga a csillagkapu objektum, nyilvántartja a színét.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **color**: StarGateColor: Tárolja a csillagkapuk színét.
* **pos:** CVector: Tárolja a csillagkapuk pozícióját.

#### Metódusok

### StarGateHandler

#### Felelősség

Gondoskodik a csillagkapuk létrejöttéről és törléséről.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **data**: StarGateInventory: Tárolja a csillagkapukat.
* **weight:** ButtonHandler: Kezeli a gombok (mérlegek) lenyomását.

#### Metódusok

* **void Create(CVector sgpos, StarGatColor color)**: Létrehoz adott pozíciójú, irányú és színű csillagkaput.
* **void Delete(StarGatColor color)**: Törli az adott színű csillagkaput.

### StarGateInventory

#### Felelősség

Nyilvántartja a csillagkapukat.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok

* **sgates**: StarGate: Tárolja a csillagkapukat.
* **weight:** ButtonHandler: Kezeli a gombok (mérlegek) lenyomását.

#### Metódusok

* **boolean IsThere(CVector from)**: Visszaadja, hogy az adott helyről érkezve van-e csillagkapu.
* **CVector StepIn(CVector from)**: Adott helyről belép a csillagkapuba, visszaadja a következő pozíciót.

### StepHandler

#### Felelősség

Lekezeli a játékos lépéseit, információt szolgáltat a mezők járhatóságáról.

#### Ősosztályok: Nincs

#### Interfészek: Nincs

#### Attribútumok:

* **sgates**: StarGateInventory: Nyilvántartja a csillagkapukat és pozíciójukat.
* **data**: DataAcessPoint: Adat-összeköttetést biztosít.

#### Metódusok

* **CVector Nextstep(CVector playerpos, Direction dir)**: Visszaadja a Player pozícióját, annak függvényében, hogy tudott-e lépni vagy sem.

### Wall

#### Felelősség

A falat megvalósító osztály. Biztosítja, hogy nem lehet átmenni, átlőni rajta. Nem halálos és nem jöhet létre rajta csillagkapu.

#### Ősosztályok: Nincs

#### Interfészek

* FieldObject

#### Attribútumok: Nincs

#### Metódusok: Nincs

### Way

#### Felelősség

Az utat megvalósító osztály. Átjárható, át lehet lőni felette, nem halálos és nem jöhet létre rajta csillagkapu.

#### Ősosztályok: Nincs

#### Interfészek

* FieldObject

#### Attribútumok: Nincs

#### Metódusok: Nincs

### ZPM

#### Felelősség

A játék során begyűjtendő ZPM-et megvalósító osztály.

#### Ősosztályok: Nincs

#### Interfészek

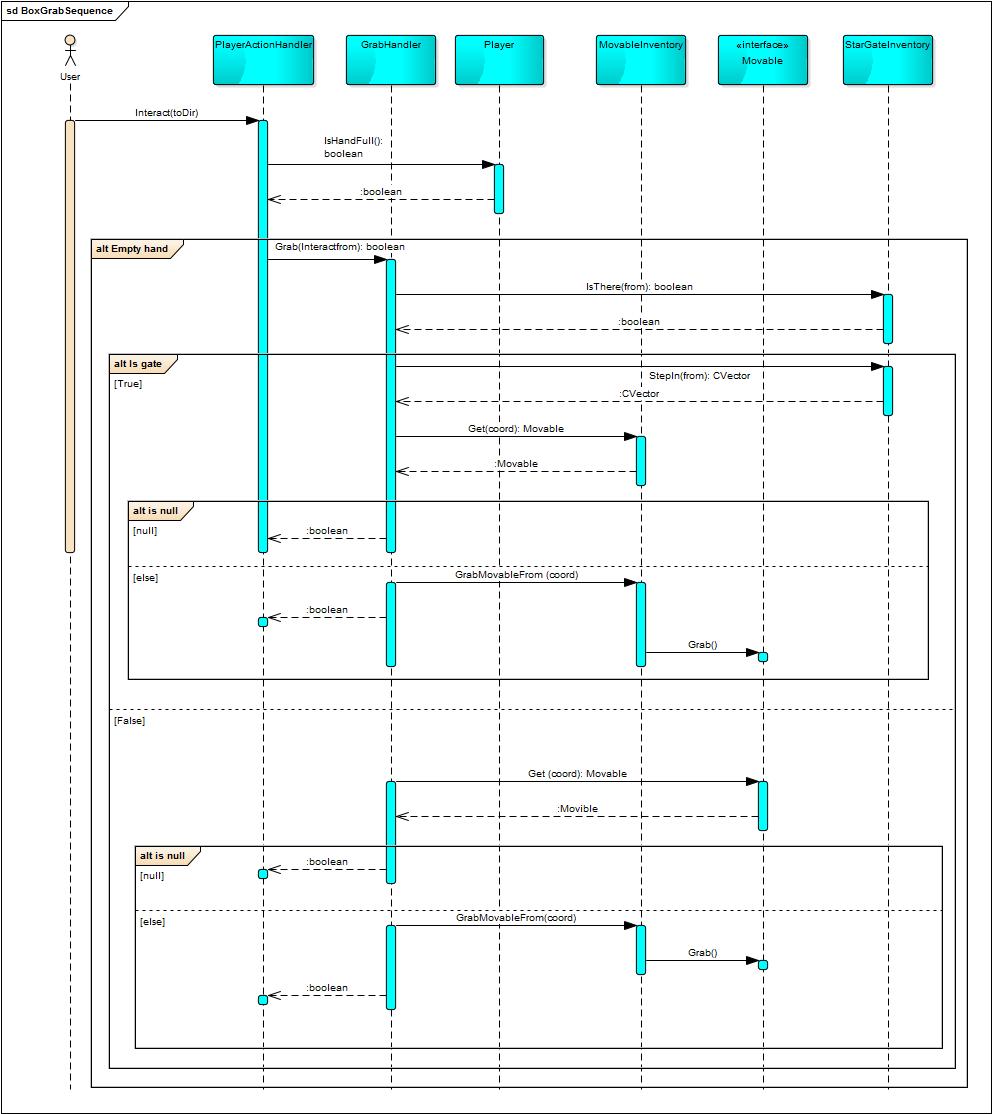
* Collectable

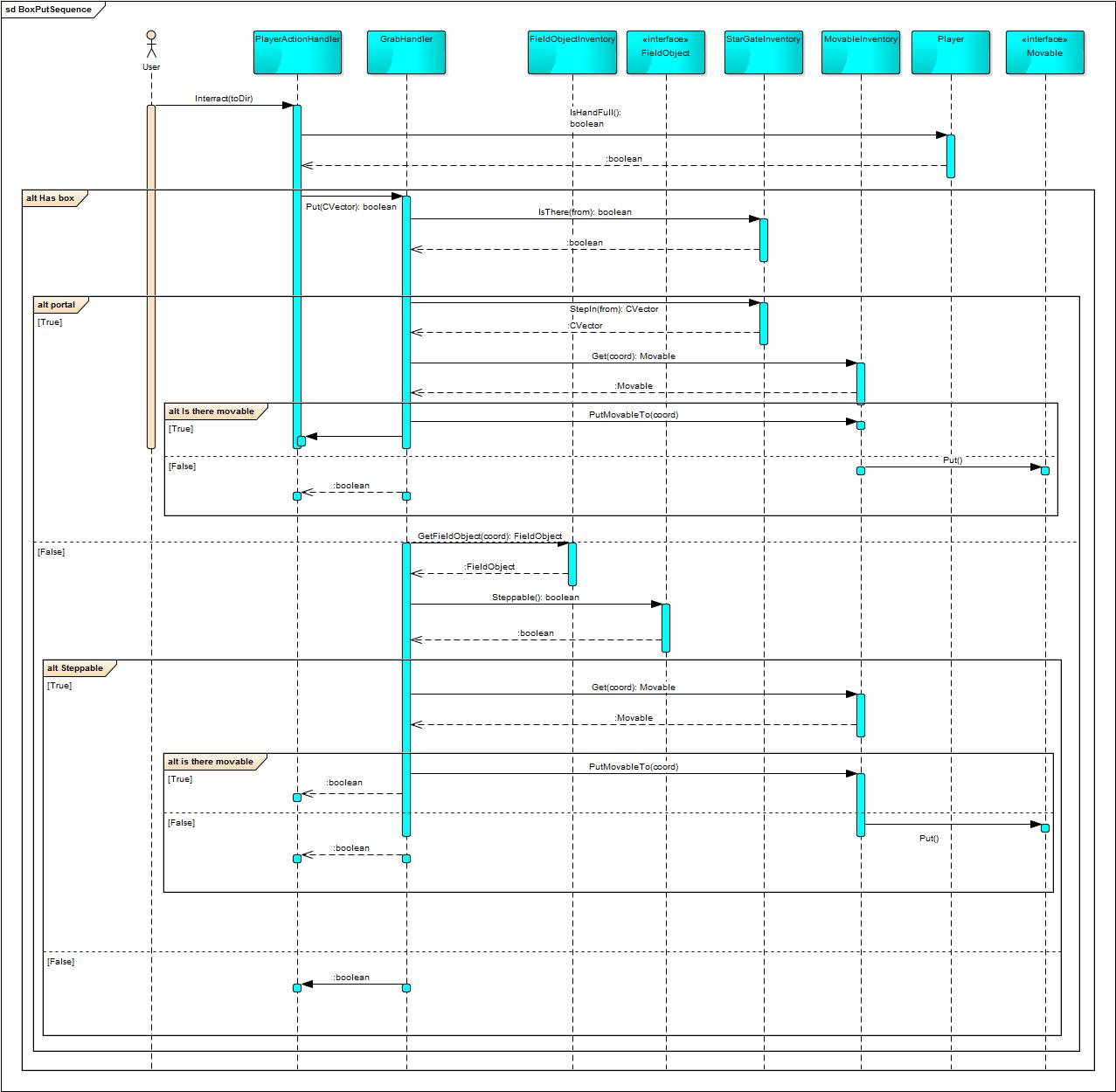
#### Attribútumok: Nincs

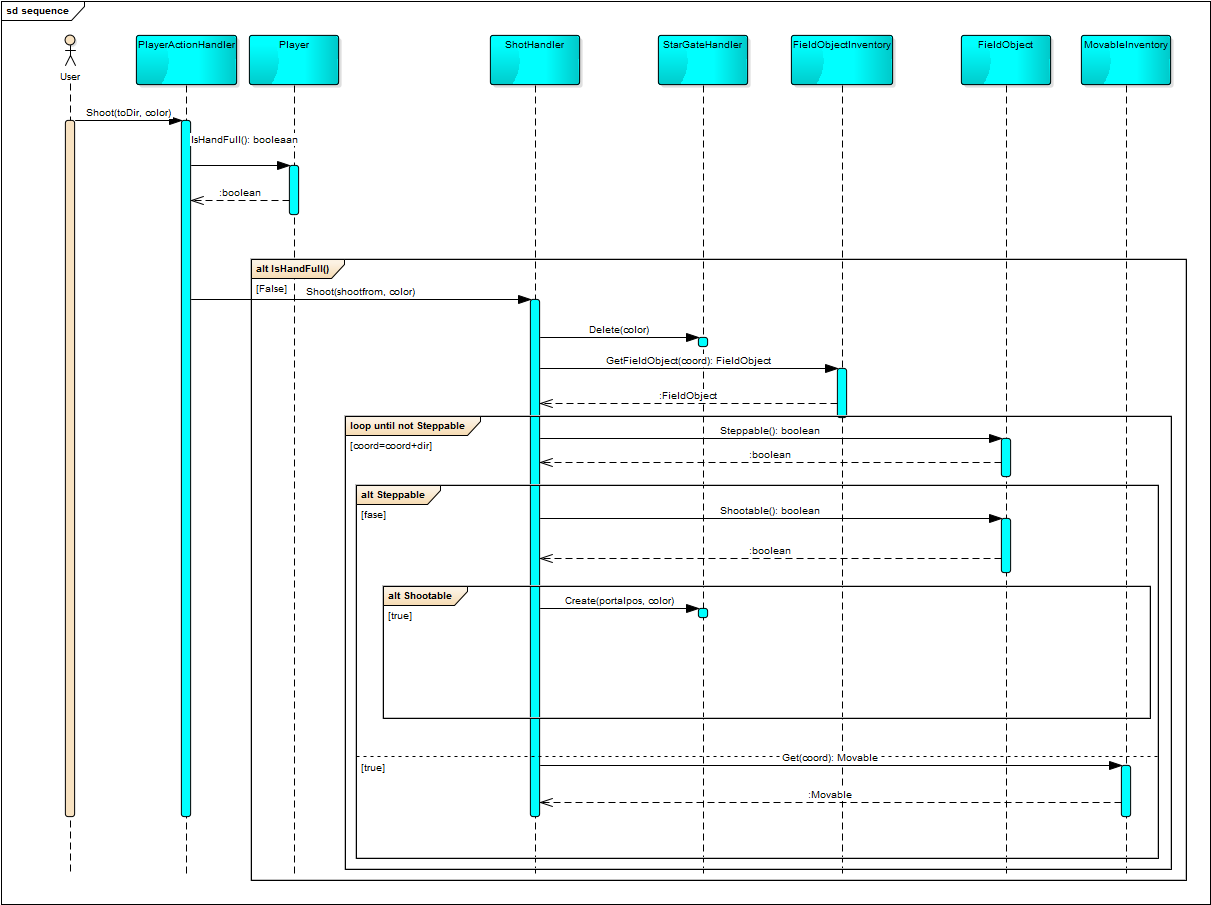
#### Metódusok: Nincs

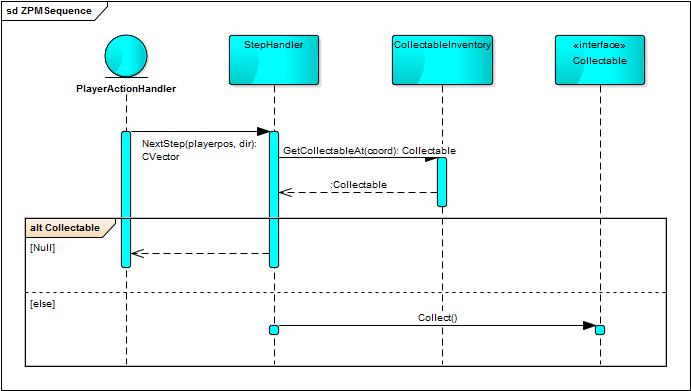
## Szekvencia diagramok

## StepSequence

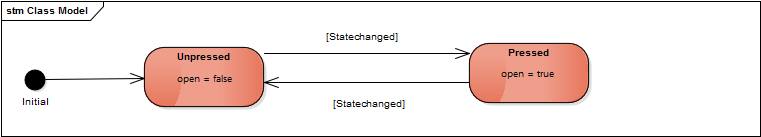








## State-chartok



## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2016.02.29 18:00 | 4 óra | Makki, Gáspár, Ritter, Papp, Borbély | Gondolkodás a felépítésen |
| 2016.03.03 16:30 | 3 óra | Makki, Ritter, Zöld, Gáspár, Papp | Osztálydiagram |
| 2016.03.04 14:30 | 4 óra | Ritter, Zöld, Gáspár, Papp | Osztálydiagram 2.0 |
| 2016.03.05 13:00 | 1 óra | Makki Zöld, Gáspár, Papp, Ritter | Osztálydiagram véglegesítése |
| 2016.03.05 14:00 | 5 óra | Gáspár | Osztálydiagram elkészítése, szekvenciák leírása |
| 2016.03.06 16:00 | 3 óra | Zöld | Objektuok és osztályok leírásai: DataAcsessPoint, FieldObjectInventory, BoxInventory, CollectableInventory, ButtonInventory, Button, FieldObject, Wall, SpecialWall, Abyss, Way, Door, Box, Box, Collectable, ZPM |
| 2016.03.06 16:00 | 6 óra | Makki | Sekvencia diagramok elkészítése |
| 2016.03.07.18:00 | 4 óra | Ritter | A maradék osztályok és objektumok leírásai |
| 2016.03.07 19:00 | 2 óra | Papp | Shoot szekvencia diagram |
| 2016.03.07.19:00 | 15 perc | Borbély | Állapotgép |