**Szoftverprojekt laboratórium  
Végleges dokumentáció**

12 - ch33kybr33ky\_labz

Konzulens:

Bodó Zsófia

Csapattagok

|  |  |  |
| --- | --- | --- |
| **Frontó András** | andrasfronto@gmail.com | **G392CQ** |
| **Czetter Gábor** | cz3tt1@gmail.com | **CP6OG3** |
| **Tomori Dániel** | tomori.daniel@gmail.com | **TDWSF5** |
| **Verasztó Bálint** | balintveraszto@gmail.com | **VSDRBZ** |

**02. Követelmény, projekt, funkcionalitás**

12 - ch33kybr33ky\_labz

Konzulens:

Bodó Zsófia

Csapattagok

|  |  |  |
| --- | --- | --- |
| **Frontó András** | andrasfronto@gmail.com | **G392CQ** |
| **Czetter Gábor** | cz3tt1@gmail.com | **CP6OG3** |
| **Tomori Dániel** | tomori.daniel@gmail.com | **TDWSF5** |
| **Verasztó Bálint** | balintveraszto@gmail.com | **VSDRBZ** |

1. **Követelmény, projekt, funkcionalitás**
   1. ***Bevezetés***
      1. **Cél**

A dokumentum célja hogy megismertesse az olvasót a játék felépítésével,szabályaival, követelményeivel és szerkezetével.

* + 1. **Szakterület**

Egy több játékos által játszható ügyességi és logikai játék, mely kikapcsolódás az egész családnak.

* + 1. **Definíciók, rövidítések**

***Definíciók:***

**Eclipse** - Fejlesztőkörnyezet, főként Java programozási nyelvhez

**Facebook** - Internetes közösségi oldal

**fejlesztőkörnyezet** - Olyan program vagy programok amelyek lehetővé teszik a fejlesztői munka végrehajtását

**funkció** - A program működésének egy külön megfogalmazható része.

**Git** - Verziókezelő rendszer

**GitHub** - A Git verziókezelőre épülő internetes szolgáltatás

**Google Drive** - Felhő alapú tárhely, beépített szövegszerkesztővel

**HSZK** - Hallgatói Számítógép Központ.

**háttértár** - A számítógép olyan tárhelye amely kikapcsolás után is tárolja az adatokat

**Skype** - Interneten keresztüli telefonálást lehetővé tevő program

**szoftver** - Számítógépen futtatható program.

**UMLet** - UML diagram készítő program

**UML** - Unified Modeling Language rövidítése, egy rendszermodellező eszköz

**verziókezelő -** Számítógépes program amellyel eltárolható a fájloknak a régebbi verziói, így nyomon követhetők a változások

**JAVA** - A Java általános célú, objektumorientált programozási nyelv, amelyet a Sun Microsystems fejlesztett. A Java alkalmazásokat jellemzően bájtkód formátumra alakítják, a bájtkód futtatása a Java virtuális géppel történik.

**IDE** - Integrált Fejlesztői Környezet, a szoftverfejlesztéshez használatos. Tartalmaz egy szövegszerkesztőt, nyelvi értelmezőt, fordítót és egyéb eszközöket, hogy megkönnyítse a fejlesztést.

**Microsoft Word** - Szövegszerkesztő program a végleges formázáshoz.

***Rövidítések:***

**bem**. - Bemutatás

* + 1. **Hivatkozások**

*https://hu.wikipedia.org/wiki/Java\_(programoz%C3%A1si\_nyelv)*

*https://www.iit.bme.hu/targyak/BMEVIIIAB02*

* + 1. **Összefoglalás**

Az alábbi dokumentumban ismertetjük a játékra vonatkozó alapszabályokat, korlátozásokat, lehetőségeket, valamint az elénk állított követelményeket és a saját megvalósítási ötleteinket.

* 1. **Áttekintés**
     1. **Általános áttekintés**

A szoftver egy grafikus megjelenítőből, menüből illetve egy játékmotorból áll.

A játékmotor generálja a pályákat illetve ez tartja számon a pályán történő változásokat. (Ládák helyét, játékosok helyét stb.) Itt figyel a játék az ütközésekre eltolásokra, lyukkal való érintkezésekre.

A játékmotor mellet van egy menü rendszer amiben a játék indítható, játékosok száma kiválasztható és a program bezárható.

Mindezeket a grafikus megjelenítő jeleníti meg a felhasználó számára. A programnak a bementet is ezen keresztül ad a felhasználó.

A program nem igényel hálózatot. A megjelenítéshez használt képek és a program saját fájljai a háttértáron találhatók.

* + 1. **Funkciók**

**Játék célja:**

A játékot több felhasználó játszhatja egyszerre. A lényeg a többi játékos megölése vagy a dobozok helyre tolása. Mindkét esetben a játék véget ér, de míg az előbbi esetben a játékos kieséses módszerrel szerezi meg a győzelmet, addig az utóbbi esetben az a játékos nyer aki a játék vége előtt (amikor már nincs mozgatható doboz) a legtöbb ládát mozgatta a helyére.

**A tábla felépítése:**

A pálya négyzetekre van osztva. Ezeken a négyzeteken lépkednek a játékosok és ugyan itt tologatják a dobozokat is.

*Speciális mezők:*

A legtöbb négyzet nem rendelkezik semmilyen komolyabb tulajdonsággal. A játékosok rájuk léphetnek minden következmény nélkül és a dobozok is rájuk csúszhatnak bármilyen mellékhatás nélkül. Azonban néhány mező, könnyen megtréfálhatja a játékosokat.

* *falak:*

A pályát határolják, semmilyen módon nem lehet rajtuk áthaladni vagy ládát áttolni.

* *oszlopok:*

Ezeken a játékosok nem képesek áthaladni a dobozokat pedig nem lehet rajtuk áttolni, tehát úgy viselkednek mint egy különálló kis falrész.

* *lyukak:*

Egyes mezők alatt beszakadt a padló és most egy hatalmas lyuk tátong rajta. Ha egy játékos vagy egy doboz lép rá erre a mezőre, beleesik a lyukba. A játékos és a doboz is megsemmisül ebben az esetben.

* *gombok és csapóajtók:*

A gombok a pálya egy bizonyos mezőjét aktiválják és kinyitnak rajta egy csapóajtót. A csapóajtó ugyan úgy viselkedik mint a korábban említett lyuk. Bármi ami beleesik megsemmisül. Azonban a gombokat nem olyan könnyű benyomni. A játékos súlya nem elég hozzá. Így a gomb csak akkor nyomódik be, ha egy dobozt tolunk arra a mezőre. Amint a dobozt eltoljuk a mezőről a gomb deaktiválódik és a csapóajtó ismét bezárul és egyszerű mezőként viselkedik. A csapóajtók végtelen sokszor kinyithatók és bezárhatók.

* *cél*

Az a mező a táblán, ahova a játékosok igyekeznek eltolni a ládákat, hogy pontot kaphassanak cserébe. A cél csak ládára reagál, ha egy játékos lép rá nem fog semmi történni.

**A játékososok tulajdonságai:**

A játékosok egyszerre több ládát is el tudnak tolni, hogyha az keresztezi útjukat. Viszont amilyen erősek, olyan könnyen véget is lehet vetni az életüknek. Ha egy játékos eltol egy dobozt és a tolás irányába egy másik játékos áll, akkor a toló játékos a tolás irányába eltolja egy mezővel a kockát és a másik játékost egyaránt. Viszont ha a kocka mögött álló játékos, akit épp el akarnak tolni, nem képes egy mezővel arrébb lépni, mivel útját egy fal, vagy oszlop gátolja, összenyomódik és számára a játék végetér.

Ugyan ez történik ha nem fal, hanem egy lyuk van az eltolt játékos irányába. A játékos a lyukba esik, így meghal.

Csábító lehetőség lehet, hogy mi magunk toljuk bele a másik játékost egy lyukba, így hozzásegítve magunkat a korai győzelemhez, azonban ez sajnos nem lehetséges. A munkások ugyanolyan erősek, így eltolni nem tudják egymást, továbbá egy mező nem elég széles ahhoz, hogy egyszerre két munkás megálljon rajta, így egy négyzeten egyszerre csak egy játékos tartózkodhat. Tehát arra a mezőre addig nem lehet lépni, amíg az adott játékos ott áll.

A játékosok pontot szereznek ha egy kockát a kijelölt helyre tolnak. Az a játékos kapja a pontot aki a végső lökést adta a kockának a célig. Így könnyen elbitorolható a pont az ügyetlenebb játékosoktól.

**A kockák tulajdonságai:**

Akár több is tolható belőlük egyszerre egy sorban. Két kocka nem tolható össze semmilyen módon, valamint a falnál és az oszlopoknál megakadnak.

A kockák elég nehezek ahhoz, hogy benyomjanak egy gombot. Amíg a kocka egy gombon tartózkodik a gombhoz tartozó csapóajtó nyitva marad.

Egy kocka megsemmisül, ha egy játékos belöki a célba. Ilyenkor a végső lökést adó játékos pontot kap. Vagy ha lyukba esik. Ebben az esetben senkinek nem jár pont.

**A játék vége:**

A játéknak több módon is vége lehet.

* Csak egy játékos maradt a pályán, mivel a többit vagy összenyomták, vagy lyukba vagy csapóajtóba lépett. Ilyenkor az utolsó játékos a nyertes.
* Nem maradt tolható kocka a pályán. (Ez bele foglalja azt az esetet is, ha minden kockát a játékosok a lyukba hajítottak.) Ilyenkor az a játékos nyer aki él, és a legtöbb pontja van. Előfordulhat, hogy a pontok száma megegyezik kettő vagy több játékosnál, ilyenkor a játék eredménye döntetlen.
  + 1. **Felhasználók**

A szoftver felhasználói számára nincs szükség különösebb előképzettségre, a használata rövid időn belül elsajátítható, és a játék élvezhető bárki számára.

A felhasználók mindegyike egy külön játékost irányít. Minden játékos egyenrangú. Csak egy felhasználó nyerhet, tehát egymás ellen versenyeznek.

* + 1. **Korlátozások**
* Az alkalmazásnak a fejlesztőkörnyezet által adott standard Java csomagokat használhat csak.
  + 1. **Feltételezések, kapcsolatok**

[*https://hu.wikipedia.org/wiki/Java\_(programoz%C3%A1si\_nyelv)*](https://hu.wikipedia.org/wiki/Java_(programoz%C3%A1si_nyelv)) *-* A Java definíciója

[*https://www.iit.bme.hu/targyak/BMEVIIIAB02*](https://www.iit.bme.hu/targyak/BMEVIIIAB02) *-* A feladat kiírás itt található

* 1. ***Követelmények***
     1. **Funkcionális követelmények**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Use-case** | **Komment** |
| 1.01 | Ládákat lehet tologatni | bem. | alapvető | megrendelő | ManageObject | Egyszerre csak egy mezőnyit |
| 1.02 | A játékosok igyekeznek minél több ládát a célba tolni | bem. | alapvető | megrendelő | ManageObject, MovePlayer |  |
| 1.03 | Több játékos van, több felhasználó irányítja őket | bem. | fontos | megrendelő | MovePlayer | Mindenki a sajátját |
| 1.04 | A falban és az oszlopban megakad a láda | bem. | alapvető | megrendelő | MovePlayer, ManageObject |  |
| 1.05 | A ládák egymást eltoják | bem. | alapvető | megrendelő | ManageObjects |  |
| 1.06 | A láda eltolhatja a játékost ha eltolható (nincs mögötte fal) | bem. | alapvető | megrendelő | MovePlayer, ManageObjects |  |
| 1.07 | A láda megöli a játékost ha nem tudja eltolni | bem. | alapvető | megrendelő | MovePlayer, ManageObjects |  |
| 1.08 | A láda nem nyomható össze | bem. | alapvető | megrendelő | ManageObjects |  |
| 1.09 | A padlón lyukak találhatók | bem. | alapvető | megrendelő | ViewLevel, CreateLevel |  |
| 1.10 | A láda lyukra tolva eltűnik | bem. | alapvető | megrendelő | ManageObjects |  |
| 1.11 | A munkás lyukra lépve meghal | bem. | alapvető | megrendelő | MovePlayer |  |
| 1.12 | A padlón kapcsolómezők találhatók | bem. | alapvető |  | ViewLevel, CreateLevel |  |
| 1.13 | A kapcsolómezők csapóajtókat kapcsolnak | bem. | alapvető | megrendelő | ViewLevel, CreateLeve, ManageObjects | Egy kapcsolóhoz egy csapóajtó tartozik |
| 1.14 | Kapcsolót csak láda kapcsolhat | bem. | alapvető | megrendelő | ManageObjects |  |
| 1.15 | A játék véget ér, ha minden láda a helyén van | bem. | alapvető | megrendelő | ManageObjects |  |
| 1.16 | A játék véget ér ha nem lehet tolni többet | bem. | alapvető | megrendelő | *ManageObjects* |  |
| 1.17 | A játékos pontot kap ha ládát a célba tol | bem. | alapvető | megrendelő | *ManageObjects* |  |
| 1.18 | A legtöbb ponttal rendelkező játékos nyer | bem. | alapvető | megrendelő | ViewLeve, ManageObjects |  |
| 1.19 | A játékosok nem tolják el egymást | bem. | alapvető | csapat | MovePlayer, ManageObjects |  |
| 1.20 | Aki meghal, kiesik és nem nyerhet | bem. | fontos | csapat | *MovePlayer* |  |
| 1.21 | A játékból ki lehet lépni | bem. | alapvető | csapat | ChooseOption, Quit |  |
| 1.22 | A játékosok száma a játék indításakor megadható | bem. | fontos | csapat | ChooseOption, PlayerNumber, NewGame |  |
| 1.23 | A játék indításakor a pályák random generálódnak | bem. | fontos | csapat | CreateLevel |  |
| 1.24 | A játékot el lehet indítani | bem. | alapvető | megrendelő | ChooseOption |  |
| 1.25 | Egy gomb egy csapóajtót nyit | bem. | fontos | csapat | ManageObjects,CreateLevel |  |
| 1.26 | A pálya négyzet alakú mezőkből áll | bem. | alapvető | csapat | ViewLevel |  |
| 1.27 | Csak egy cél van a pályán | bem. | alapvető | csapat | ViewLevel, CreateLevel |  |

* + 1. **Erőforrásokkal kapcsolatos követelmények**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Komment** |
| 2.01 | Git | nincs | alapvető | csapat | Elosztott verziókezelő |
| 2.02 | Github account | nincs | alapvető | csapat | Git tárhely |
| 2.03 | Eclipse | nincs | opcionális | csapat | Java IDE |
| 2.04 | Monitor | nincs | alapvető | csapat |  |
| 2.05 | Billentyűzet | nincs | alapvető | csapat |  |
| 2.06 | HSZK-ban találhatókkal azonos vagy jobb teljesítményű PC | bem. | alapvető | megrendelő |  |
| 2.07 | UMLet | nincs | opcionális | csapat | UML modellező |
| 2.08 | Egér | nincs | alpvető | csapat |  |

* + 1. **Átadással kapcsolatos követelmények**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Komment** |
| 3.01 | Szkeleton átadás | bem. | alapvető | megrendelő |  |
| 3.02 | Proto átadás | bem. | alapvető | megrendelő |  |
| 3.03 | Teljes program átadása | bem. | alapverő | megrendelő |  |
| 3.04 | Útmutató alapján telepíthető, külső segítség nélkül | bem. | fontos | megrendelő |  |
| 3.05 | BME HSZK gépeken is működnie kell | bem. | alapvető | megrendelő |  |

* + 1. **Egyéb nem funkcionális követelmények**

Nincs egyéb nem funkciónális követelmény.

* 1. ***Lényeges use-case-ek***
     1. **Use-case leírások**

|  |  |
| --- | --- |
| **Use-case neve** | Új Játék (NewGame) |
| **Rövid leírás** | Új játékot indít. |
| **Aktorok** | Menu |
| **Forgatókönyv** | A játékot elindítja megszabott számú munkással egy random pályán. |

|  |  |
| --- | --- |
| **Use-case neve** | Játékosok számának kiválasztása (PlayerNumber) |
| **Rövid leírás** | Kiválasztjuk, hogy hányan szeretnénk játszani. |
| **Aktorok** | Menu |
| **Forgatókönyv** | Beveszi hogy hány játékossal akarjuk elindítani a játékot. |

|  |  |
| --- | --- |
| **Use-case neve** | Kilépés (Quit) |
| **Rövid leírás** | Kilépünk a programból |
| **Aktorok** | Menu |
| **Forgatókönyv** | A gomb megnyomására a program kilép. |

|  |  |
| --- | --- |
| **Use-case neve** | Opció választás (ChoseOption) |
| **Rövid leírás** | A felhasználó választ a menüpontok közül |
| **Aktorok** | User |
| **Forgatókönyv** | Kiválasztunk egyet a menüpontok közül |

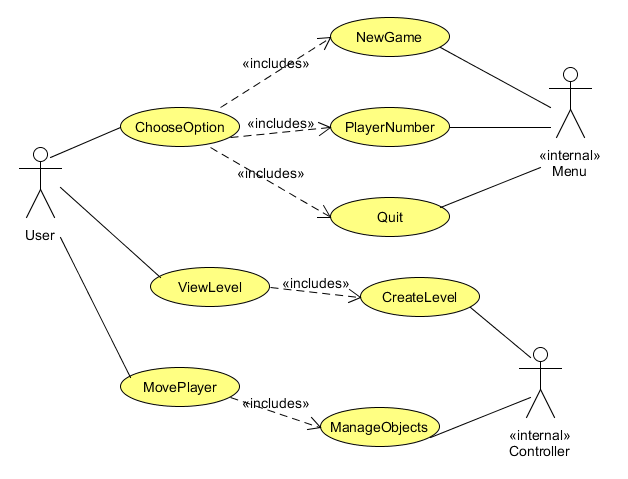
|  |  |
| --- | --- |
| **Use-case neve** | Pálya megtekintése (ViewLevel) |
| **Rövid leírás** | A felhasználó látja a pályát |
| **Aktorok** | User |
| **Forgatókönyv** | A játék indítása után a felhasználó látja a pályát. |

|  |  |
| --- | --- |
| **Use-case neve** | Játékos mozgatása (MovePlayer) |
| **Rövid leírás** | A felhazsnáló irányítja a saját játékosát. |
| **Aktorok** | User |
| **Forgatókönyv** | A játék indítása után a megadott gombokkal tudja mozgatni a felhasználó a játékost a pályán. |

|  |  |
| --- | --- |
| **Use-case neve** | Obijektumok kezelése(ManageObjects) |
| **Rövid leírás** | A falak, ládák, kapcsolók, cél és lyukak kezelése. |
| **Aktorok** | Controller |
| **Forgatókönyv** | Számontartja az obijektumok pozícióját, állapotát, és lekezeli a kölcsönhatásaikat. |

|  |  |
| --- | --- |
| **Use-case neve** | Pálya generálás (CreateLevel) |
| **Rövid leírás** | Véletlenszerűen generálódik egy pálya. |
| **Aktorok** | Controller |
| **Forgatókönyv** | Amikor indul a játék egy random pályát generál a játék. |

* + 1. **Use-case diagram**



* 1. **Szótár**

|  |  |
| --- | --- |
| **Aktiválás** | Az a cselekvés amit egy gomb végez egy csapóajtó felé. Vagy lyukká teszi vagy sima mezővé változtatja. |
| **Áthaladás** | A játékos illetve a láda képes az adott mezőn lévő objektumon átlépni, illetve áttolódni.  (Áthaladható például egy gomb, nem áthaladható például egy fal) |
| **Áthatolhatatlan** | Azok a speciális mezők, amelyekre nem lehet lépni. |
| **Benyomás** | Egy doboz hatása egy gombra, ha egy mezőn vannak. Benyomás hatására a gomb aktiválódik. |
| **Cél** | Egy speciális mező a pályán, ami a rátolt ládát elnyeli és pontot ad a végső tolást végző játékosnak. |
| **Csapóajtó** | Egy speciális lyuk, mely csak akkor nyílik ki, ha a hozzá tartozó gombját egy láda aktiválta.. |
| **Doboz** | Láda szinonímája. |
| **Döntetlen** | Nem maradt több mozgatható láda és kettő vagy több játékos pontszáma megegyezik. |
| **Fal** | Áthatolhatatlan mező típus, ami a pályát határolja. |
| **Gomb** | Speciális mező, melyen a játékos áthaladhat, de ha egy ládát tolnak rá, kinyit egy csapóajtót. |
| **Győzelem** | A játékot nem lehet folytatni és a nyertes egyértelmű. |
| **Halál** | A játékos elveszíti az eddig megszerzett pontjait eltűnik a pályáról, és nem nyerhet. |
| **Játékos** | A játékban szereplő karakter akit a felhasználó irányít. |
| **Kieséses módszer** | Egy kivételével minden játékos meghal a pályán. Ilyenkor az életben maradt a nyertes. |
| **Láda** | A pályán lévő objektum amit, a játékosok képesek mozgatni, célba juttatni vagy megsemmisíteni. |
| **Lépkedés** | A játékosok mozgása két mező között. |
| **Lyuk** | Speciális mező, melyre ha egy ember, vagy láda kerül azonnal megsemmisül. |
| **Megakadás** | Amikor egy játékos vagy egy doboz egy áthatolhatatlan mezőbe ütközik.  Ugyanabban az irányba tovább nem tolható. |
| **Megsemmisül** | A játékos és a doboz számára a halált jelenti. Kitörlődnek a játékból és már nem lehet őket mozgatni. |
| **Mező** | Kis négyzetes területek, amikből a pálya épül fel. |
| **Mozgatható** | Egy ládára vett tulajdonság, amikor van olyan irály melyre a ládát tovább lehet tolni. |
| **Munkás** | Játékos szinonímája |
| **Négyzet** | A mező szinonímája. |
| **Oszlop** | A pálya közepén fekvő egy mező méretű fal. |
| **Összenyomódás** | Egyfajta halál típus. Akkor következik be, ha egy játékos egy láda és egy áthatolhatatlan tárgy között áll és egy második játékos az első munkás irányába tolja a ládát. Ilyenkor az első játékos meghal. |
| **Pálya** | Az a terület, ahol a játék zajlik. |
| **Pont** | Egy játékos kapja, ha sikerül a célba juttatnia egy dobozt. |
| **Speciális mező** | Egy olyan mező melyen egy olyan objektum van elhelyezve (nem játékos, nem láda),  mely befolyással bír a játék alakulására. |
| **Tábla** | A pálya szinonímája. |
| **Tolható** | Mozgatható szinonímája. |
| **Tologatás** | A pályán lévő ládák mozgatása a játékos, vagy egy másik láda által. |

* 1. ***Projekt terv***

***2.6.1*** *Csapat*

|  |  |
| --- | --- |
| **Név** | **Felelősség** |
| Frontó András | kód,dokumentáció |
| Czetter Gábor | kód,dokumentáció |
| Tomori Dániel | kód,dokumentáció |
| Verasztó Bálint | kód,dokumentáció |

***2.6.2* Kommunikáció**

**Verziókezelés:** Mivel csapatban dolgozunk ezért fontos, hogy mindenki elérje a legfrisseb fileokat, forráskódokat. Ezt valamilyen módon el kell juttani mindenkinek, ezért választottuk a Git elosztott verziókezelő szoftvert. Ehhez az ingyenes tárhelyet a GitHub biztosítja.

Azért erre esett a választásunk, mert így könnyen nyomonkövethető, hogy ki mivel dolgozik, valamint a korábbi feladataink során már használtuk, tehát némi tapasztalatunk már van a programmal.

A dokumentáció közös szerkesztésére más megoldást választottunk, ezt a Google Drive segítségével közösen tudjuk szerkeszteni.

**Facebook:** A csapatnak van egy közös privát Facebook csoportja ahol a fontosabb információkat tesszük közzé.

**Skype:** Amennyiben nincs lehetőségünk találkozni, Skype beszélgetés során folytatjuk a közös munkát.

**Megbeszélések:**  A heti konzultáción kívül, hetente változó időpontban közösen dolgozunk.

**2.6.3 Használt Programok**

**Verziókezelés**: A fent említett Git.

**Dokumentáció**: A fent emltett Google Drive. A formázásnál használjuk a Microsoft Word programot is.

**Fejlesztőkörnyezet**: A forráskodot Eclipse fejlesztőkörnyezetben készítjük.

**2.6.4 Határidők**

|  |  |
| --- | --- |
| ***Határidő*** | ***Név*** |
| febr. 19. | *02. Követelmény, projekt, funkcionalitás* |
| *febr. 26.* | *03. Analízis modell kidolgozása 1.* |
| *márc. 5.* | *04. Analízis modell kidolgozása 2.* |
| *márc. 12.* | *05. Szkeleton tervezése* |
| *márc. 19.* | *06. Szkeleton* |
| *márc. 26.* | *07. Prototípus koncepciója* |
| *ápr. 9.* | *08. Részletes tervek* |
| *ápr. 23.* | *10. Prototípus* |
| *máj. 2.* | *11. Grafikus felület specifikációja* |
| *máj. 14.* | *13. Grafikus változat* |
| *máj. 18.* | *14. Összefoglalás* |

A feladatot 3 részletben oldjuk meg: Szkeleton, Prototípus, Grafikius.

* 1. **Napló**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.02.16 | 3 óra | Czetter  Frontó  Verasztó | Közös megbeszélésen a dokumentum nagyvonalakban elkészült. |
| 2018.02.17 | 2 óra | Tomori | A dokumentum újbóli átolvasása hiányosságok pótlása, megjegyzések |
| 2018.02.18 | 2óra | Czetter  Frontó  Tomori  Verasztó | Skypeos egyeztetés mindenki átfutotta a dokumentumot és ellenőriztük egymás munkáját, össze hangoltuk a különböző részeket. |
| 2018.02.18 | 1 óra | Tomori | A dokumentum végleges formázása Microsoft Word segítségével. |

**03. Analízis modell kidolgozása 1.**

12 - ch33kybr33ky\_labz

Konzulens:

Bodó Zsófia

Csapattagok

|  |  |  |
| --- | --- | --- |
| **Frontó András** | **G392CQ** | andrasfronto@gmail.com |
| **Czetter Gábor** | **CP6OG3** | cz3tt1@gmail.com |
| **Tomori Dániel** | **TDWSF5** | tomori.daniel@gmail.com |
| **Verasztó Bálint** | **VSDRBZ** | balintveraszto@gmail.com |

1. **Analízis modell kidolgozása**
   1. ***Objektum katalógus***
      1. **Player**

A Player-eket a felhasználók irányítják a pálya falakkal határolt mezőin.  Képesek Box-okat tolni, de nem tudnak átmenni a Wall vagy Pillar Field-eken. A Button-ok nem aktiválódnak, ha rájuk lépnek viszont a TrapDoor-ok és a Hole-ok képesek megölni őket. Továbbá lépés hiányába, ha rájuk tolódik egy doboz akkor meghal.

* + 1. **Box**

A Box-okat a Playerek tologatják a Map-en. Nem lehet Wall-ba vagy Pillar-ba tolni őket, de cserébe képesek aktiválni a Button-al ellátott Field-eket. Ha Hole-al vagy TrapDoor-al rendelkező Field-re lépnek akkor meghalnak. Ha a Target mezőbe csúszik be egy Box, az pontot ad annak a Player-nek, aki utoljára hozzáért, majd eltűnik.

* + 1. **Map**

A játékban lévő összes Field-et összefoglalja, továbbá felelős azok elhelyezkedéséért és specialitásuk kiosztásáért.

* + 1. **Field**

A játék mezői, amik együtt a Map-ot alkotják. Field-eknek vannak speciális esetük, amik mind máshogy reagálnak arra, ha egy Player vagy egy Box kerül rájuk.

* + 1. **Wall**

A Filed speciális esete, ami határolja a Map mozgásterét. Se Player se Box nem képes átmenni rajta.

* + 1. **Pillar**

A Wall egy speciális esete, ami a Map azon részén helyezkedik el, ahol a Player-ek és a Box-ok mozognak. Lényegében ez egy egyedül álló falrész.

* + 1. **Hole**

A Field egy speciális esete, ha Player vagy Box kerül rá eltűnik.

* + 1. **Button**

A Field egy speciális esete. Csak akkor tud benyomódni, aktiválódni, ha egy Box kerül fölé. Ekkor a hozzá tartozó TrapDoor, valahol a Map-on kinyílik és ami rajta van eltűnik.

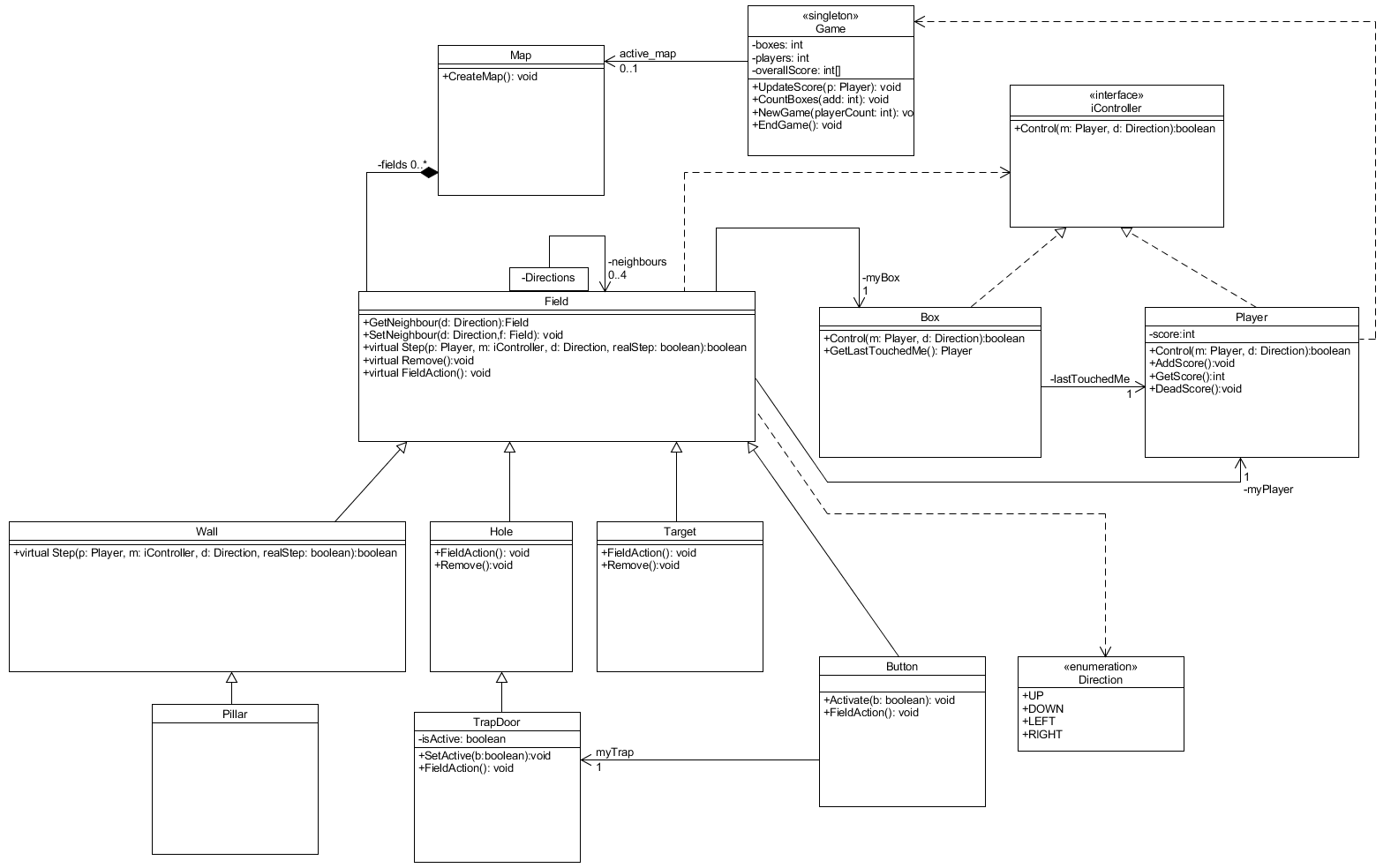
* + 1. **TrapDoor**

Ha a hozzá tartozó Button aktív, kinyílik és a rajta lévő Box, vagy Player beleesik. Ha a Button-ról eltűnik a Box, bezárul és újra egy egyszerű Field lesz.

* + 1. **Target**

Cél mező a Field egy speciális esete, ha Box kerül rá eltűnik és pontot kap a végső lökést adó játékos. Ha játékos lép rá, nem történik semmi.

* 1. ***Statikus struktúra diagramok***

**

* 1. ***Osztályok leírása***

Azért hogy leegyszerűsítsük az osztály leírásokat azokat a virtuális függvényeket amiket a leszármazott nem ír felül nem írtuk ki még egyszer (pl.: Button.Step,Button.Remove stb).

* + 1. **Box**
       - **Felelősség**

Az iController-t megvalósító osztály. A Player-ek Box-okat tolnak a Field-eken keresztül. Box-ok képesek gombokat benyomni, és a Target-be lökésükért pont jár a végős lökést adó Player-nek.

* + - * **Ősosztályok**

Nincs

* + - * **Interfészek**

iController

* + - * **Attribútumok**
* **Player lastTouchedMe**
  + - * **Metódusok**
* **Player GetLastTouchedMe:**.Visszaadja azt a player-t aki legutoljára ért hozzá.
* **boolean Controll(Player p, Direction d):** Mozdulási szándékot intéz egy Field felé, és visszaadja, hogy sikerült-e a mozdulás. Első paramétere megmondja, hogy ki tolta, a második pedig, hogy milyen irányba.
  + 1. **Button**
       - **Felelősség**

A Field-ből öröklődik. Felelős azért, hogy ha egy Box kerül rá, ami képes benyomni, akkor a hozzá tartozó TrapDoor kinyíljon, továbbá ha az lekerül róla, bezáródjon. Ha egy Player lép rá, nem történik semmi, a csapó ajtó csukva marad.

* + - * **Ősosztályok**

Field → Button

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **TrapDoor myTrap**
* **Field[] neighbours**
* **Box myBox**
* **Player myPlayer**
  + - * **Metódusok**
* **void Activate(Boolean b)**: Box hatására aktiválja a gombot, Player hatására nem történik semmi.
* **void FieldAction():** A gomb aktiválódását segíti elő. Figyeli, hogy beszorul-e a láda.
* **Field GetNeighbour(Direction d)**: A megadott irányba lévő Field-et adja vissza.
* **void SetNeighbour(Direction d, Field f):** Beállítja az adott irányba a paraméterül kapott Field-et.
  + 1. **Field**
       - **Felelősség**

Ezekből épül fel a Map. Egy Field mező felelős a rajta lévő Player-ért vagy Box-ért. Ha az a külső hatás miatt el szeretne mozdulni a Field felelős a lépéséért vagy az esetleges haláláért. A szomszédos mezőket is ő tartja számon.

* + - * **Ősosztályok**

Nincs

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **Field[] neighbours**
* **Box myBox**
* **Player myPlayer**
  + - * **Metódusok**
* **Field GetNeighbour(Direction d)**: A megadott irányba lévő Field-et adja vissza.
* **void SetNeighbour(Direction d, Field f):** Beállítja az adott irányba a paraméterül kapott Field-et.
* **virtual boolean Step(Player p, m iController, Direction d, boolean realStep)**: Tovább lépést indít egy megadott irányba. Visszatérési értéke azt jelzi, hogy a lépést sikeresen megtörtént-e vagy valami miatt meghiúsult. Az első paramétere azt mutatja, hogy ki tolta az objektumot, ennek később lesz majd hatása majd, amikor több Box-ot tolunk egyszerre. A második az a Box vagy Player aki a mezőre szeretne lépni. A harmadik pedig az irány, amerre az egész megindul. Az utolsó paraméter pedig ha igaz akkor tényleg lép,  ha hamis akkor csak egy ellenőrzés hogy léphetne e. Ez a beszorulás vizsgálatához kell.
* **virtual void Remove():** Az egyszerű Field Remove() függvénye csak és kizárólag Player-t tud eltűntetni, mivel csak az tud meghalni attól, ha rátolnak egy Box-ot. Box-okat nem lehet összetolni.
* **virtual void FieldAction():** Minden Field-hez az egyedi tulajdonságot hajtja végre ami kötelezően meghívódik, ha valaki rálép vagy rátolódik. Figyeli, hogy beszorul e a láda.
  + 1. **Direction**

Egy enumeráció, ami a lehetséges irányokat tartalmazza. (UP, DOWN, LEFT, RIGHT)

* + 1. **Game**
       - **Felelősség**

Egy singleton osztály. Feladata a játék elkezdése és lezárása, továbbá ezeknek a feltételeknek az ellenőrzése. Folyamatosan számon tartja hány Boksz mozgatható, így ha ez a szám nullára csökken a játék automatikusan véget ér, valamint a Player-ek pontjait és számát is ő tartja nyilván.

* + - * **Ősosztályok**

Nincs

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **int boxes**
* **int players**
* **int[] overallScore**
  + - * **Metódusok**
* **void UpdateScore(Player p):** Frissíti az adott Player pontszámát. Vagy növeli, vagy nulláza abban az esetben ha meghalt. Ha kiesik 3 játékos befejezi a játékot.
* **void CountBoxes(int add):** Számlálja a mozgatható Box-okat a Map-en. Amikor elfogynak befejezi a játékot.
* **void NewGame(int playerCount):** Új játékot indít a megadott számú játékossal.
* **void EndGame():** Véget ér a játék, ha vagy mozgatható Box, vagy 1-nél több irányítható Player nincs már a Map-en.
  + 1. **Hole**
       - **Felelősség**

A Field-ből öröklődik, ha bármi rákerül, eltűnik és törlődik a játékból. Ha Box kerül bele nem jár érte senkinek pont, ha Player, meghal és elveszíti az eddig összegyűjtött pontjait.

* + - * **Ősosztályok**

Field → Hole

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **Field[] neighbours**
* **Box myBox**
* **Player myPlayer**
  + - * **Metódusok**
* **void FieldAction():** Ebben az esetben, bármi ami rákerül, leesik, tehát törlődik a játékból.
* **void Remove():** A Hole Remove() függvénye már nem csak Player-t, de Box-ot is képes eltűntetni, hiszen tök mindegy mi lépett rá, az rögtön törlődik a játékból.
* **Field GetNeighbour(Direction d)**: A megadott irányba lévő Field-et adja vissza.
* **void SetNeighbour(Direction d, Field f):** Beállítja az adott irányba a paraméterül kapott Field-et.
  + 1. **iController**
       - **Felelősség** Egy interface, amit a Player és a Box osztály valósít meg. Felelős az őt megvalósító osztályok lépéseiért.
       - **Ősosztályok**

Nincs

* + - * **Metódusok**
* **boolean Control(Player p, Direction d)**: Mozdulási szándékot intéz egy Field felé, és visszaadja, hogy sikerült-e a mozdulás. Első paramétere megmondja, hogy ki tolta, a második pedig, hogy milyen irányba. Dobozra hívva a beállítja a lastTouched paramétert is.
  + 1. **Map**
       - **Felelősség**

A pálya szabályos kialakításáért felelős, ezzel együtt raktározza a Field-eket.

* + - * **Ősosztályok**

Nincs

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **Field[] fields**
* **Game active\_map**
  + - * **Metódusok**
* **void CreateMap()**: Megadott algoritmusok szerint alkotja meg random a pályát az egyszerű Field-ek és a speciális Field-ek használatával. Lerakja a Boxokat és a Playereket is.
  + 1. **Pillar**
       - **Felelősség**

A Pillar-ok ugyan úgy viselkednek, mint a falak attól eltekintve, hogy a Map közepén helyezkednek el. Nem lehet rajtuk áthaladni, vagy Box-okat áttolni.

* + - * **Ősosztályok**

Field → Wall → Pillar

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **Field[] neighbours**
* **Box myBox**
* **Player myPlayer**
  + - * **Metódusok**
* **virtual boolean Step(Player p, m iController, Direction d, boolean realStep)**: Ebben az esetben a Step függvény nem engedi a lépést és rögtön elutasítja, ha valaki rá szeretne kerülni bármelyik irányból is érkezik a kérés a lépésre.
* **Field GetNeighbour(Direction d)**: A megadott irányba lévő Field-et adja vissza.
* **void SetNeighbour(Direction d, Field f):** Beállítja az adott irányba a paraméterül kapott Field-et.
  + 1. **Player**
       - **Felelősség**

Az iController megvalósító osztály. Felelős a Playerek mozgatásáért és a pontjaiért.

* + - * **Ősosztályok**

Nincs

* + - * **Interfészek**

iController

* + - * **Attribútumok**
* **int score**
  + - * **Metódusok**
* **void AddScore():** Egyet hozzáad a Player pontjaihoz.
* **int GetScore():** Lekérdezi az adott Player pontját.
* **boolean Controll(Player m, Direction d):** Mozdulási szándékot intéz egy Field felé, és visszaadja, hogy sikerült-e a mozdulás. Első paramétere megmondja, hogy ki tolta, a második pedig, hogy milyen irányba.
* **void DeadScore():** -1-re állítja az adott Player pontjait, majd a Game UpdateScrore() függvénye kiveteti a játékból.
  + 1. **Target**
       - **Felelősség**

A Fieldből származó osztály. Csak a Box-okra aktiválódik a Player-eket figyelmen kívül hagyja. Ha egy Box rákerül az eltűnik és az utolsó lökést adó játékosnak pontot ad. Természetesen a Target mezőn is ugyan úgy összenyomódhat a Player mint bárhol máshol.

* + - * **Ősosztályok**

Field → Target

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **Directions[] neighbours**
* **Box myBox**
* **Player myPlayer**
  + - * **Metódusok**
* **void FieldAction():** Rátoláskor a Box eltűnik és az utolsó Playernek aki hozzá ért pont írást kezdeményez.
* **void Remove():** A szokásos Remove() függvény. Ha meghívődik törli a Playert vagy a Boxot a Field-ről.
* **Field GetNeighbour(Direction d)**: A megadott irányba lévő Field-et adja vissza.
* **void SetNeighbour(Direction d, Field f):** Beállítja az adott irányba a paraméterül kapott Field-et.

**3.3.10 TrapDoor**

* + - * **Felelősség**

A Field leszármazottja. Egy hozzá tartozó gomb utasítására nyílik ki és viselkedik úgy, mint egy Hole, vagy csukódik be és egy sima Field-ként létezik.

* + - * **Ősosztályok**

Field → Hole → TrapDoor

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **boolean isActive**
* **Field[] neighbours**
* **Box myBox**
* **Player myPlayer**
  + - * **Metódusok**
* **void SetActive(boolean: b):** Átállítja a TrapDoor állását. Nyitottból csukott, csukottból nyitott.
* **void FieldAction():** Aktiválódáskor kinyílik, és ha éppen egy Player vagy egy Box volt rajta, akkor azt törli. Ha a hozzá tartozó Button-ról letolják a Box-ot akkor bezárul és sima Field lesz amire midnen következmény nélkül rá lehet lépni vagy tolni.
* **Field GetNeighbour(Direction d)**: A megadott irányba lévő Field-et adja vissza.
* **void SetNeighbour(Direction d, Field f):** Beállítja az adott irányba a paraméterül kapott Field-et.

**3.3.11 Wall**

* + - * **Felelősség**

A Field-ből öröklődik. Sem a Player-ek sem a Box-ok nem képesek áthaladni rajta. Áthatolhatatlanok a játék kezdetétől a végéig és ezt már senki nem veheti eltőlük.

* + - * **Ősosztályok**

Field → Wall

* + - * **Interfészek**

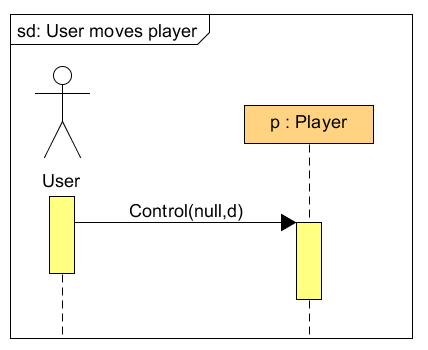
Nincs

* + - * **Attribútumok**
* **Field[] neighbours**
* **Box myBox**
* **Player myPlayer**
  + - * **Metódusok**
* **virtual boolean Step(Player p, m iController, Direction d)**: Ebben az esetben a Step függvény nem engedi a lépést és rögtön elutasítja, ha valaki rá szeretne kerülni.
* **Field GetNeighbour(Direction d)**: A megadott irányba lévő Field-et adja vissza.
* **void SetNeighbour(Direction d, Field f):** Beállítja az adott irányba a paraméterül kapott Field-et.

***Szekvencia diagramok***

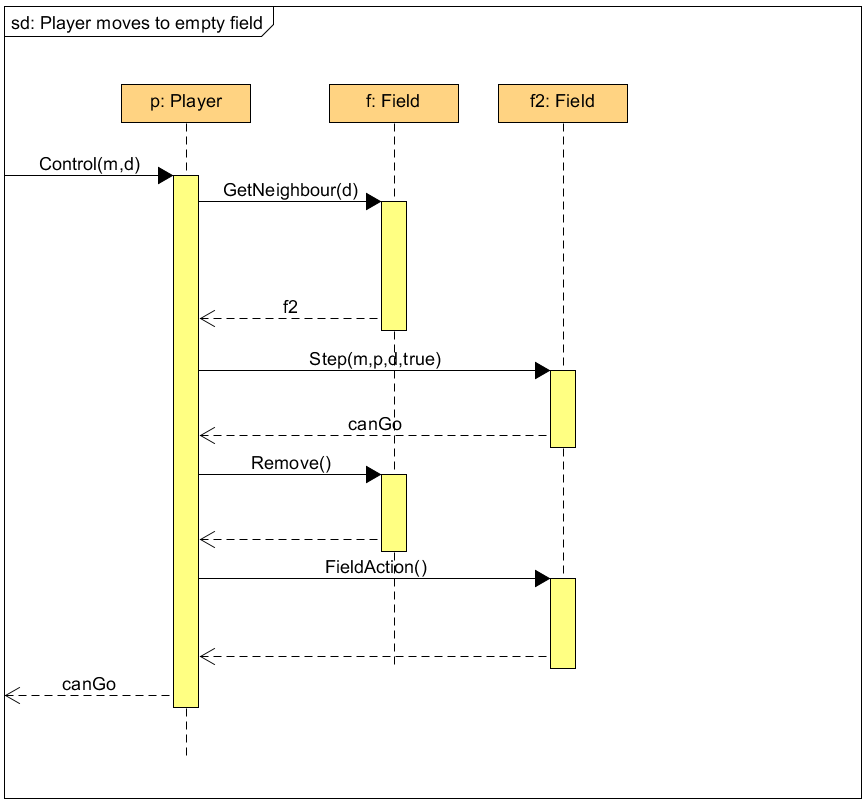
***3.4.1 User moves player***

A felhasználó mozgatja az egyik játékost. Fontos, hogy itt nem egy másik térképen lévő objektum “löki” meg a játékost - ezért a Controll első átvett értéke *null*.

******

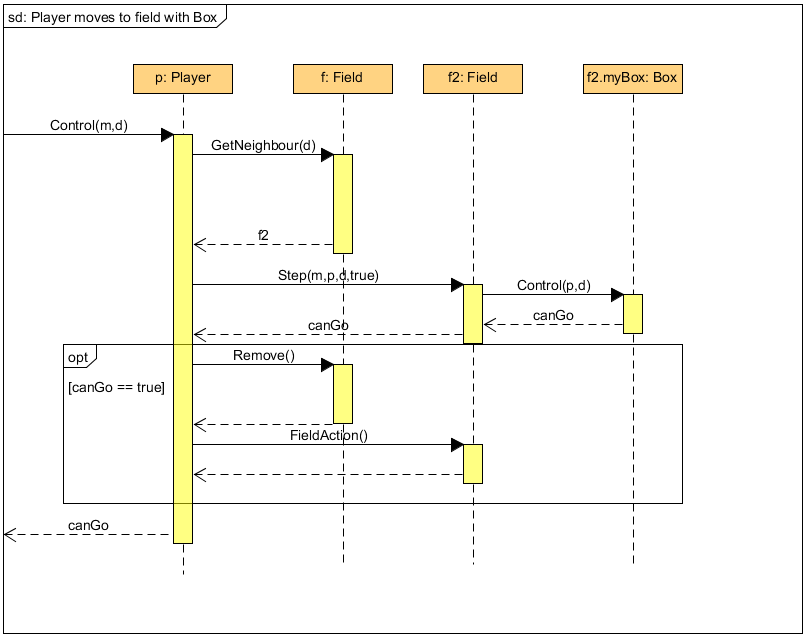
***3.4.2 Player moves to empty field***

A játékost valami (akár a user akár pl egy doboz) elmozgatja egy irányba, ahol üres mező taláható. Az ábrán nincs lekezelve az az eset, amikor van valami a szomszéd mezőn - ezt külön diagramon kezeljük. Az m értéke vagy a mozgást kezdeményező Player vagy null ha a játékos kezdte.

******

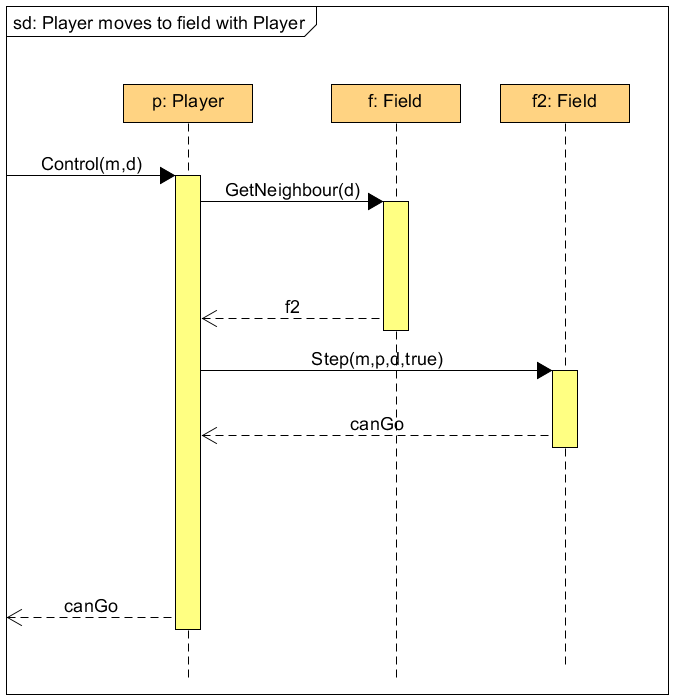
***3.4.3 Player moves to field with box***

A játékos olyan mezőre lép, amin van egy doboz. Ha a játékost a felhasználó mozgatta, akkor a Control első paramétere null, ha valamilyen másik objektum, akkor a mozgást kezdeményező Player. Ha a láda által visszaadott canGo érték igaz, akkor a játékos eltolhatja a dobozt és a lépés végbemegy. A doboznak saját magát adja a Controll-ban, így a dobozok mindig arról a Player-ről tudnak aki “utoljára ért bele” a mozgásba.

******

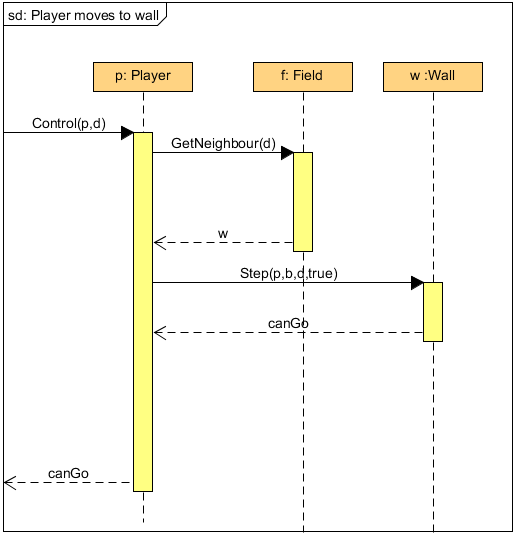
***3.4.4 Player moves to field with player***

A játékost a felhasználó vagy egy másik objektum egy olyan mezőre mozgatná, amin egy másik játékos áll. (Itt is a Controll függvény első paramétere ettől függően vagy null vagy a “megtoló” Player.)

******

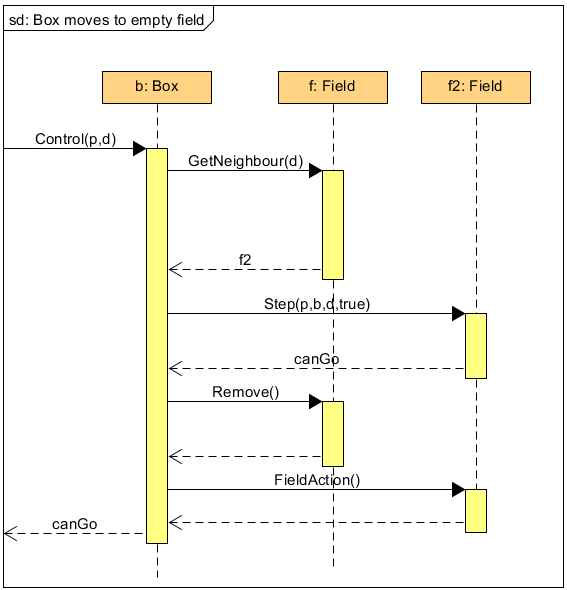
***3.4.5 Player moves to wall***

A Player egy falra akar lépni amit nem tehet, itt a Palyer lép Player-re esethez hasonlóannem fog oda lépni. (Itt is a Controll függvény első paramétere ettől függően vagy null vagy a “megtoló” Player.)



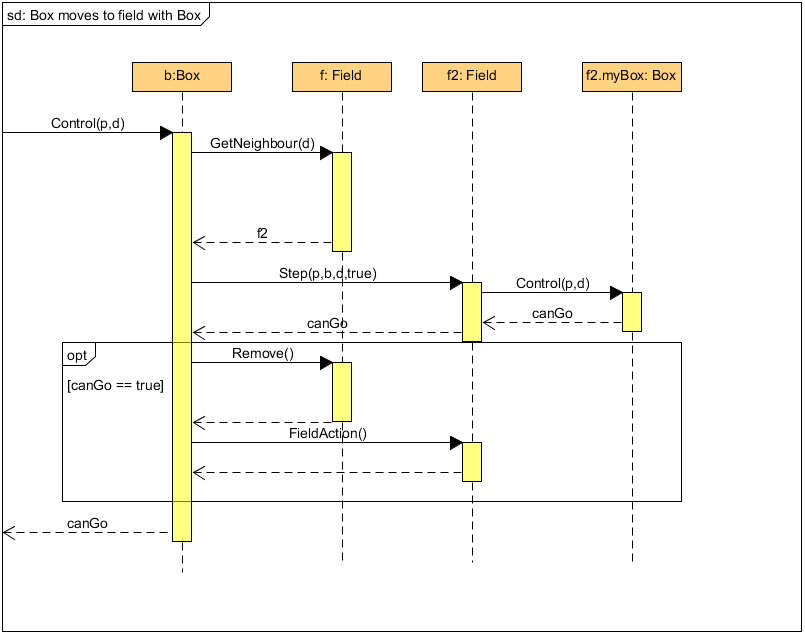
***3.4.6 Box moves to empty field***

Egy dobozt egy szomszédos üres mezőre tolják. Ebben a diagramban csak azt az esetet kezeljük, amikor a szomszéd mező üres.

******

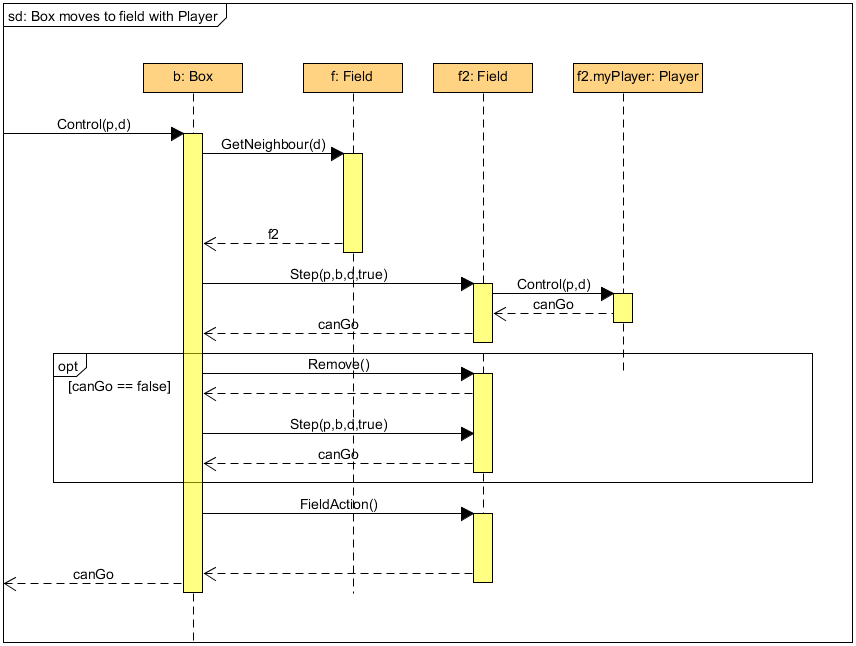
***3.4.7 Box moves to field with box***

A dobozt egy olyan mezőre tolják, amin szintén egy doboz van. Rekurzíve megnézi, hogy tud-e lépni az adott irányba. Ha igen, akkor lép.

******

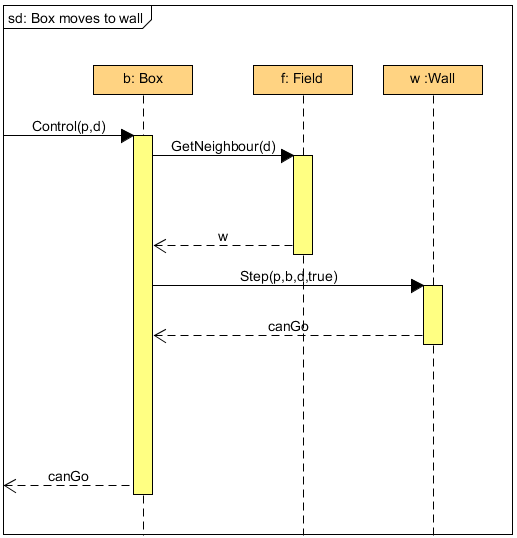
***3.4.8 Box moves to field with player***

A dobozt egy olyan mezőre tolják, ahol egy másik játékos van. Ha nem tud menni a játékostól “összenyomja” azt.

******

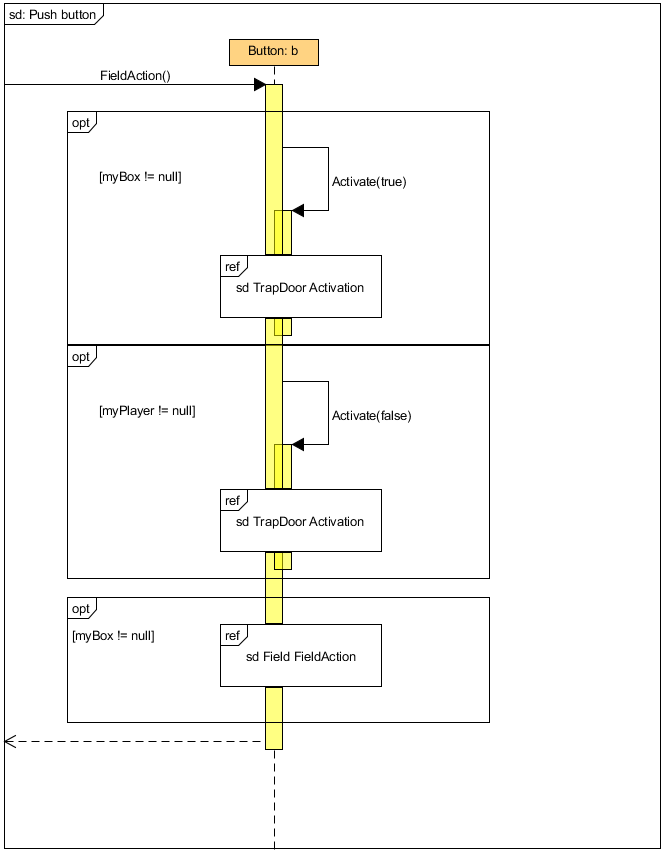
***3.4.9 Boksz moves to wall***

A doboz egy falra akar lépni amit nem tehet tehát itt a canGo false-al jön vissza és nem lép oda.



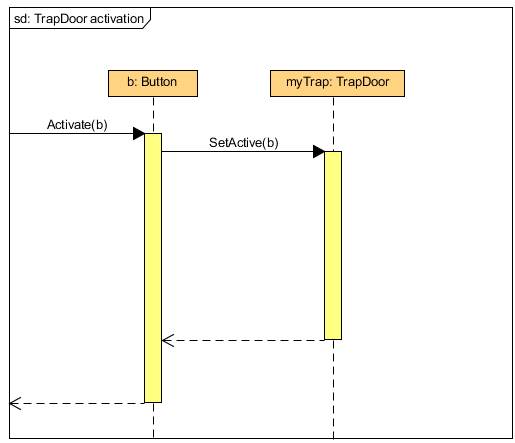
***3.4.10 Box pushes button***

A doboz egy olyan mezőre kerül, amin egy gomb van. Meghívódik a Button Activate függvénye a rajta lévő obijektumnak megfelelő boolean értékkel. (a Box megnyomja a Player nem)



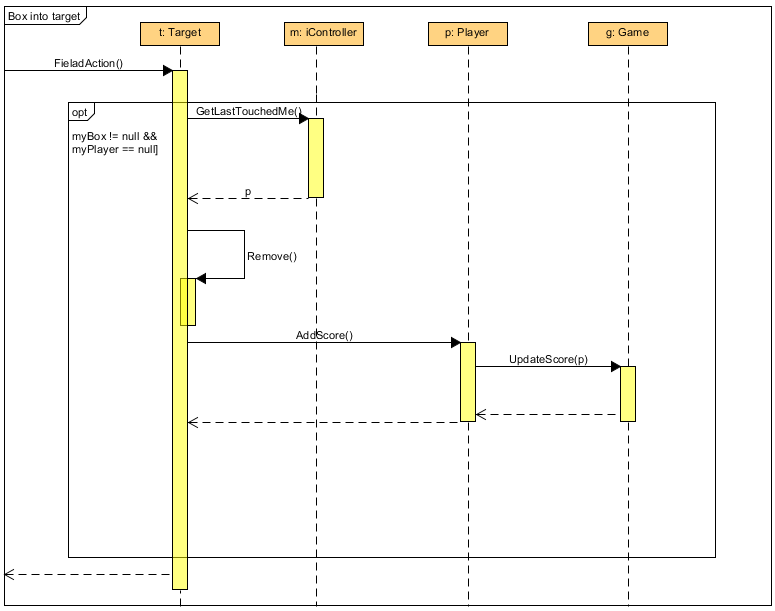
***3.4.11 TrapDoor activation***

Ha egy gombra rátolnak egy dobozt, az aktivál egy TrapDoort. Beállítja a gombon az aktivitást, ami meghívja az általa tartalmazott TrapDoor aktiváló függvényét, ami ez után kinyílik és Hole-ként viselkedik.

******

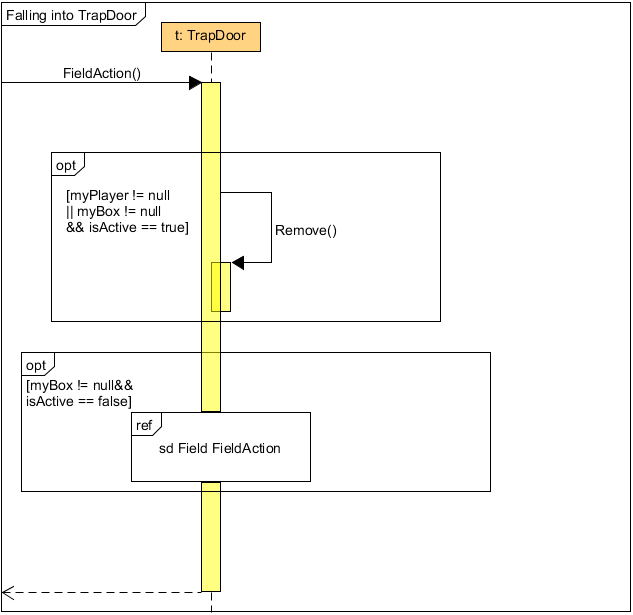
***3.4.12 Move box into target***

Egy doboz célba tolása. Megnézi az adott Field, hogy Box került-e rá. Ha igen, akkor lekérdezi, hogy ki volt az a player aki az adott mozgást indította. Ezután a Target Fieldről eltávolítja a Boxot, és az adott Playernek jóváírja a pontot.

******

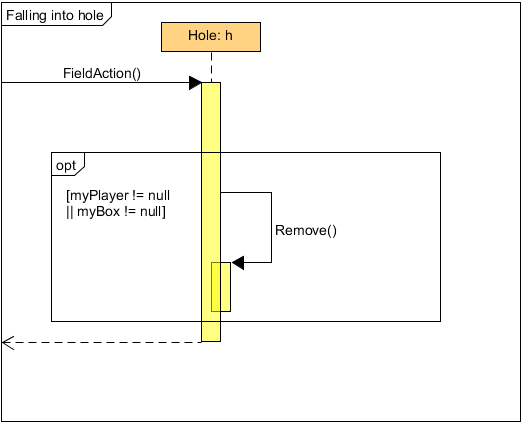
***3.4.13 Falling into TrapDoor***

Vagy egy Player vagy egy Box lép egy nyitott TrapDoor-ra. Ekkor az adott objektum leesik és eltűnik.

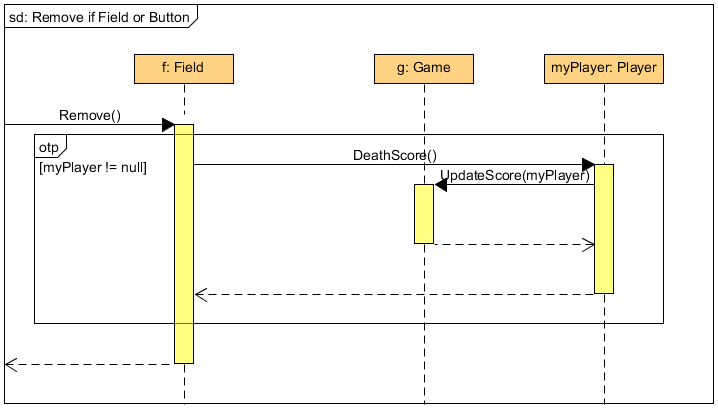
******

***3.4.14 Falling into hole***

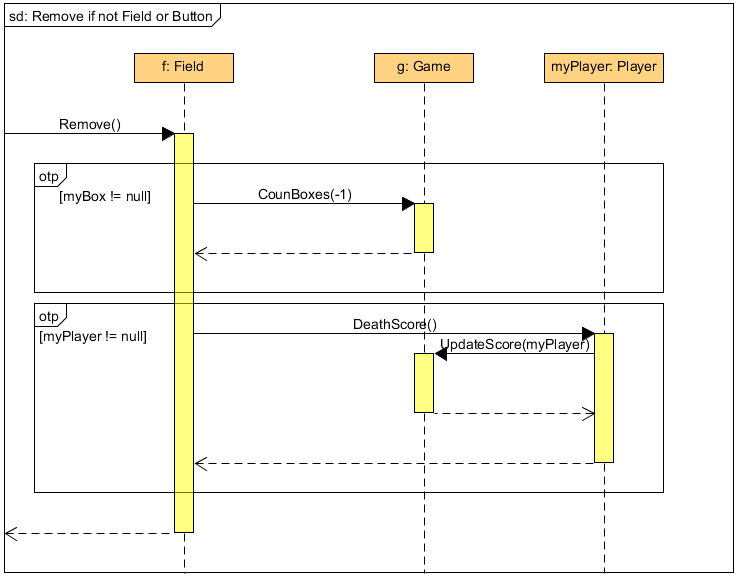
Egy Player vagy egy Box ha Hole-ra lép, leesik és eltűnik.

******

***3.4.15 Field és Button mező Remove***

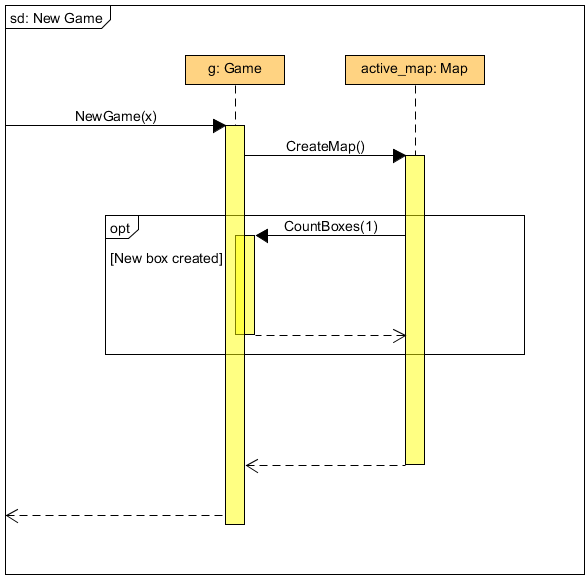
******

***3.4.16 Minden más Remove***

******

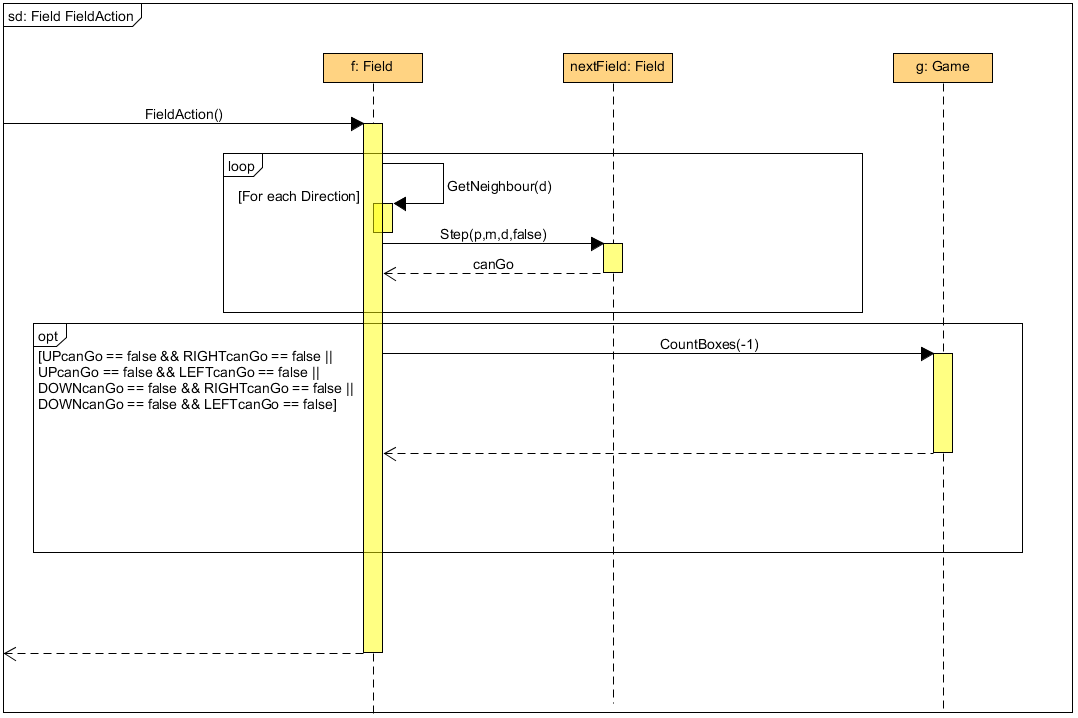
***3.4.17 New Game***

Új játék indításakor léterjön egy Map, ami feltöltődik Fieldekkel és Boxokkal.

******

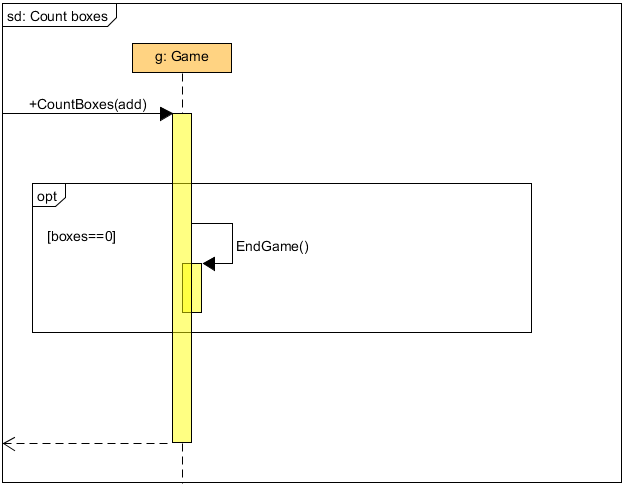
***3.4.18 Field Field Action***

Minden irányba megnézzük hogy tolható e. A Step ha false az utolsó paramétere akkor nem lép csak rekurzívan lekérdezi hogy arra lehet e menni. Ha a doboz “beszorult” akkor csökkenti a pályán lévő dobozok számát. Amikor a dobozok elfogynak a játéknak vége.

******

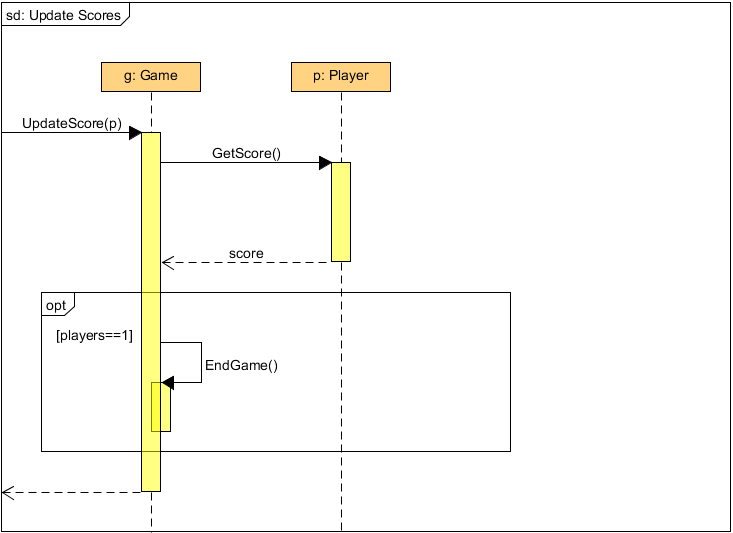
***3.4.19 Count Boxes***

A dobozok számát mindig a megadott értékkel “növeli” ha a dobozok elfogynak, tehát nincs több mozgatható, akkor a játék véget ér.

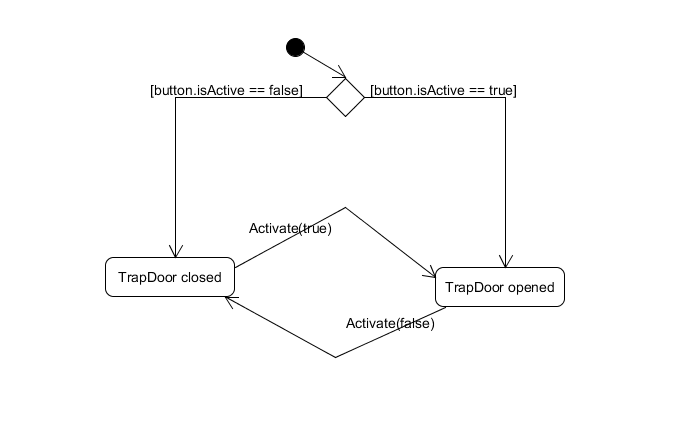
******

***3.4.20 Update Scores***

A játékosok pontját a DeathScore() -1-re állítja, ha ezt érzékeli az UpdateScore akkor csökkenti a Playerek számát. Ha a Playerek száma 1 akkor az utolsó játékos nyer és a játék véget ér.

******

* 1. ***State-chartok***



* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.02.20 11:00 | 40 perc | Czetter  Frontó  Tomori  Verasztó | Megbeszélés:  Objektumkatalógus megvitatása. Feladatok kiosztása |
| 2018.02.20 12:20 | 80 perc | Frontó | 3.2 Statikus struktúra diagramok |
| 2018.02.20 12:20 | 80 perc | Tomori | 3.2 Statikus struktúra diagramok |
| 2018.02.20 12:20 | 80 perc | Czetter | 3.2 Statikus struktúra diagramok |
| 2018.02.20 12:20 | 80 perc | Verasztó | 3.2 Statikus struktúra diagramok |
| 2018.02.20 14:00 | 2 óra | Frontó | 3.3 Szekvencia diagramok |
| 2018.02.20 14:00 | 2 óra | Tomori | 3.3 Szekvencia diagramok |
| 2018.02.20 14:00 | 2 óra | Czetter | 3.3 Szekvencia diagramok |
| 2018.02.20 14:00 | 2 óra | Verasztó | 3.3 Szekvencia diagramok |
| 2018.02.21 10:00 | 2 óra | Frontó | 3.3 Szekvencia diagramok |
| 2018.02.21 10:00 | 2 óra | Czetter | 3.3 Szekvencia diagramok |
| 2018.02.21 10:00 | 2 óra | Verrasztó | 3.3 Szekvencia diagramok |
| 2018.02.21 10:00 | 2 óra | Tomori | 3.3 Szekvencia diagramok |
| 2018.02.21 13:00 | 1 óra | Czetter | 3.3 Szekvencia diagramok |
| 2018.02.21 13:00 | 1 óra | Frontó | 3.3 Szekvencia diagramok |
| 2018.02.21 13:00 | 1 óra | Tomori | 3.3 Szekvencia diagramok |
| 2018.02.21 13:00 | 1 óra | Verrasztó | 3.3 Szekvencia diagramok |
| 2018.02.22 10:00 | 3,5 óra | Czetter  Frontó | 3.2 Osztály diagramok |
| 2018.02.22 10:00 | 3,5 óra | Verrasztó  Tomori | 3.3 Szekvencia diagramok |
| 2018.02.23 | 3 óra | Czetter  Frontó  Verrasztó | 3.3 Szekvencia diagramok |
| 2018.02.25 15:00 | 6 óra | Frontó  Czetter  Verrasztó  Tomori | Források integrálása, dokumentum formázása. |

**04. Analízis modell kidolgozása 2.**

12 - ch33kybr33ky\_labz

Konzulens:

Bodó Zsófia

Csapattagok

|  |  |  |
| --- | --- | --- |
| **Frontó András** | **G392CQ** | andrasfronto@gmail.com |
| **Czetter Gábor** | **CP6OG3** | cz3tt1@gmail.com |
| **Tomori Dániel** | **TDWSF5** | tomori.daniel@gmail.com |
| **Verasztó Bálint** | **VSDRBZ** | balintveraszto@gmail.com |

1. **Analízis modell kidolgozása**
   1. ***Objektum katalógus***
      1. **Player**

A Player-eket a felhasználók irányítják a pálya falakkal határolt mezőin.  Képesek Box-okat tolni, de nem tudnak átmenni a Wall-okon. A Button-ok nem aktiválódnak, ha rájuk lépnek viszont a TrapDoor-ok és a Hole-ok képesek megölni őket. Továbbá lépés hiányába, ha rájuk tolódik egy doboz akkor meghalnak.

* + 1. **Box**

A Box-okat a Playerek tologatják a Map-en. Nem lehet Wall-ba tolni őket, de cserébe képesek aktiválni a Button-al ellátott Field-eket. Ha Hole-al vagy TrapDoor-al rendelkező Field-re tolódnak akkor meghalnak. Ha a Target mezőbe csúszik be egy Box, az pontot ad annak a Player-nek, aki utoljára hozzáért, majd eltűnik.

* + 1. **Field**

A játék mezői, amik a Wall-okkal határolva alkotják a játszható területet. Field-eken a játek elhelyez különböző tárgyakat, akadályokat, hogy izgalmasabbá tegye a felhasználói élményt. Ezek a tárgyak vagy akadályok, mind máshogy reagálnak arra, hogy Player, vagy Box került-e rájuk.

* + 1. **Wall**

Határolja a játszható terület. Se Player se Box nem képes átmenni rajta.

* + 1. **Hole**

Ez egy olyan mező, melyen egy hatalmas lyuk tátong. Ha Player vagy Box kerül rá eltűnik és törlődik a játékból. Ha Player esik bele, az addig megszerzett pontjai nullázódnak.

* + 1. **Button**

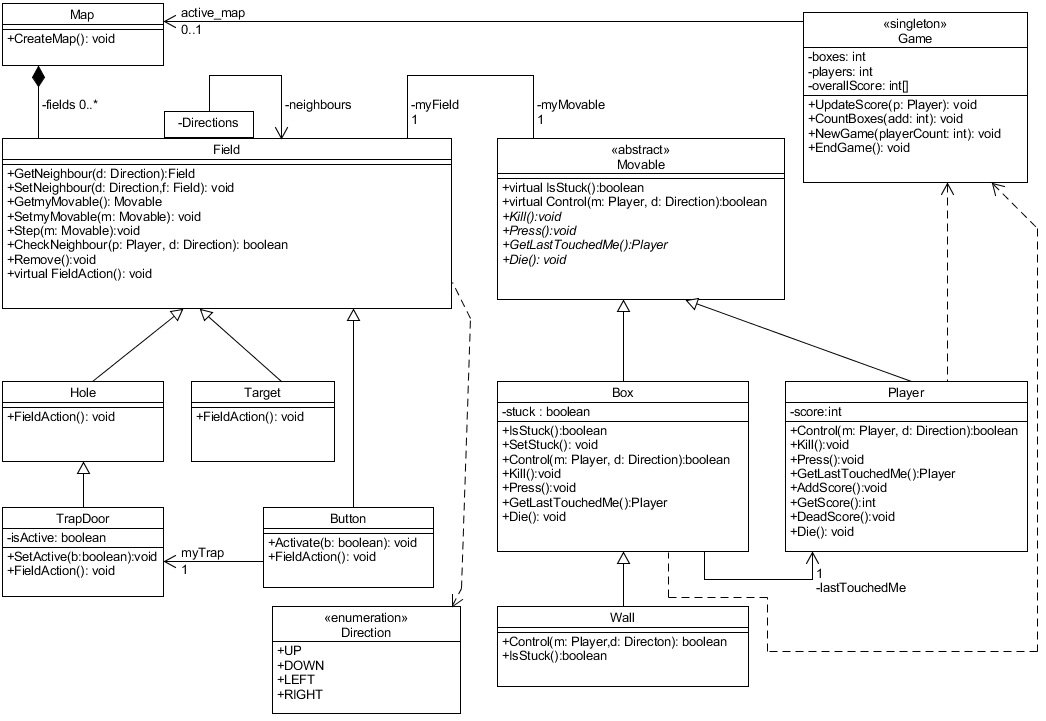
Egy mező, melyre a játék egy gombot helyezett el. Csak akkor tud benyomódni, aktiválódni, ha egy Box kerül fölé. Ekkor a hozzá tartozó TrapDoor, valahol egy másik mezőn kinyílik és ami rajta van eltűnik.

* + 1. **TrapDoor**

Ha a hozzá tartozó Button aktív, kinyílik és a rajta lévő Box, vagy Player beleesik. Ha a Button-ról eltűnik a Box, bezárul és újra egy egyszerű mező lesz.

* + 1. **Target**

A cél mező. Ha Box kerül rá eltűnik és pontot kap a végső lökést adó játékos. Ha játékos lép rá, nem történik semmi.

* 1. ***Statikus struktúra diagramok***
  2. ***Osztályok leírása***

Azért hogy leegyszerűsítsük az osztály leírásokat azokat a virtuális függvényeket amiket a leszármazott nem ír felül nem írtuk ki még egyszer (pl.: Button.Step,Button.Remove stb).

* + 1. **Box**
       - **Felelősség**

A Moveable ősosztályból származó osztály. A Player-ek Box-okat tolnak a Field-eken keresztül. Box-ok képesek gombokat benyomni, és a Target-be lökésükért pont jár a végső lökést adó Player-nek, amit a lastTouchedMe változójában tárol. A stuck változója pedig azt jelzi ha a doboz végleg mozdíthatatlanná vált.

* + - * **Ősosztályok**

Moveable → Box

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **Player lastTouchedMe**
* **boolean stuck**
  + - * **Metódusok**
* **Player GetLastTouchedMe:**.Visszaadja azt a player-t aki legutoljára ért hozzá.
* **boolean Controll(Player p, Direction d):** Mozdulási szándékot intéz egy Field felé, és visszaadja, hogy sikerült-e a mozdulás. Első paramétere megmondja, hogy ki tolta, a második pedig, hogy milyen irányba. Annyiban tér el az ős Controll függvényétől, hogy ha nem tud menni egy Field-re, mert azon egy Moveable helyezkedik el akkor Kill-t hív rá ami, ha Player volt akkor megöli.
* **boolean IsStuck():** Visszaadja, hogy be van-e szorulva a doboz.
* **void SetStuck():** A stuck változót állítja, ha a doboz beszorult, vagy tolhatóvá vált.
* **void Kill():** Ha Kill-t hívnak rá nem csinál semmit mert a doboz nem össze nyomható.
* **voidPress():** Ha gombra tolódik, aktiválja azt.
* **void Die():** Amikor törlik akkor fut le, el intéz mindent ami a doboz játékból való kivételéhez kell.
  + 1. **Button**
       - **Felelősség**

A Field-ből öröklődik. Felelős azért, hogy ha egy Box kerül rá, ami képes benyomni, akkor a hozzá tartozó TrapDoor kinyíljon, továbbá ha az lekerül róla, bezáródjon. Ha egy Player lép rá, nem történik semmi, a csapó ajtó csukva marad.

* + - * **Ősosztályok**

Field → Button

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **TrapDoor myTrap**
* **Field[] neighbours**
* **Moveable myMoveable**
  + - * **Metódusok**
* **void Activate(Boolean b)**: Box hatására aktiválja a gombot, Player hatására nem történik semmi.
* **void FieldAction():** A gomb aktiválódását segíti elő. Figyeli, hogy beszorul e a láda.
  + 1. **Field**
       - **Felelősség**

Ezekből épül fel a Map. Egy Field mező felelős a rajta lévő Moveable-ért. Ha az a külső hatás miatt el szeretne mozdulni a Field felelős az átlépésért. A szomszédos mezőket is ő tartja számon.

* + - * **Ősosztályok**

-

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **Field[] neighbours**
* **Moveable myMoveable**
  + - * **Metódusok**
* **Field GetNeighbour(Direction d)**: A megadott irányba lévő Field-et adja vissza.
* **void SetNeighbour(Direction d, Field f):** Beállítja az adott irányba a paraméterül kapott Field-et.
* **Moveable GetmyMoveable()**: Visszaadja a rajta lévő Moveable objektumot. Ha nincs rajta, a visszatérési értéke null.
* **void SetmyMoveable(Moveable m)**: Beállít az adott Field rá egy Moveable objektumot.
* **boolean CheckNeighbour(Player p, Direction d)**: Tovább lépési szándékot jelez a megadott Player-től az adott irányba. Ha a visszatérési értéke igaz, akkor szabadon léphet tovább, egyébként marad az adott Field-en.
* **void Step(Moveable m)**: A megadott Moveable objektummal elvégzi a lépést az adott Field-re.
* **void Remove():** Az épp rajta álló Moveablet eltünteti.
* **virtual void FieldAction():** Minden Field-hez az egyedi tulajdonságot hajtja végre ami kötelezően meghívódik, ha valaki rálép vagy rátolódik. Figyeli, hogy beszorul e a láda.
  + 1. **Direction**

Egy enumeratio, ami a lehetséges irányokat tartalmazza. (UP, DOWN, LEFT, RIGHT)

* + 1. **Game**
       - **Felelősség**

Egy singleton osztály. Feladata a játék elkezdése és lezárása, továbbá ezeknek a feltételeknek az ellenőrzése. Folyamatosan számon tartja hány Box mozgatható még, így ha ez a szám nullára csökken a játék autómatikusan véget ér, valamint a Player-ek pontjait és számát is ő tartja nyilván.

* + - * **Ősosztályok**

Nincs

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **int boxes**
* **int players**
* **int[] overallScore**
  + - * **Metódusok**
* **void UpdateScore(Player p):** Frissíti az adott Player pontszámát. Vagy növeli, vagy nulláza abban az esetben ha meghalt. Ha kiesik 3 játékos befejezi a játékot.
* **void CountBoxes(int add):** Számlálja a mozgatható Box-okat a Map-en. Amikor elfogynak befejezi a játékot.
* **void NewGame(int playerCount):** Új játékot indít a megadott számú játékossal.
* **void EndGame():** Véget ér a játék, ha vagy mozgatható Box, vagy 1 nél több irányítható Player nincs már a Map-en.
  + 1. **Hole**
       - **Felelősség**

A Field-ből öröklődik, ha bármi rákerül, eltűnik és törlődik a játékból. Ha Box kerül bele nem jár érte senkinek pont, ha Player, meghal és elveszíti az eddig összegyűjtött pontjait.

* + - * **Ősosztályok**

Field → Hole

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **Field[] neighbours**
* **Moveable myMoveable**
  + - * **Metódusok**
* **void FieldAction():** Ebben az esetben, bármi ami rákerül, leesik, tehát törlődik a játékból.
  + 1. **Map**
       - **Felelősség**

A pálya szabályos kialakításáért felelős, ezzel együtt raktározza a Field-eket.

* + - * **Ősosztályok**

Nincs

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **Field[] fields**
* **Game active\_map**
  + - * **Metódusok**
* **void CreateMap()**: Megadott algoritmusok szerint alkottja meg random a pályát az egyszerű Field-ek és a speciális Field-ek használatával. Lerakja a Box-okat, Wall-okat és a Player-eket is.
  + 1. **Moveable**
       - **Felelősség**

A Moveable egy olyan abstract osztály melyből a Box és a Player is egyaránt származik. Egy Field csak egy Moveable-t taralmazhat és a Moveable is eltárolja a Field-et amin épp elhelyezkedik.

* + - * **Ősosztályok**

Nincs

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **Field myField**

* + - * **Metódusok**

(Az abstract függvényeket nem vettük fel a jobb átláthatóság miatt)

* **virtual boolean Controll(Player p, Direction d):** Mozdulást intéz egy Field felé, és visszaadja, hogy sikerült-e a mozdulás. Első paramétere megmondja, hogy ki tolta, a második pedig, hogy milyen irányba.
* **virtual boolean IsStuck():** Alapvetően false-val tér vissza.
  + 1. **Player**
       - **Felelősség**

A Moveable-ből származó osztály. Felelős a Playerek mozgatásáért és a pontjaiért.

* + - * **Ősosztályok**

Field → Player

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **int score**
  + - * **Metódusok**
* **void AddScore():** Egyet hozzáad a Player pontjaihoz.
* **int GetScore():** Lekérdezi az adott Player pontját.
* **boolean Controll(Player m, Direction d):** Mozdulást intéz egy Field felé, és visszaadja, hogy sikerült-e a mozdulás. Első paramétere megmondja, hogy ki tolta, a második pedig, hogy milyen irányba.
* **void DeadScore():** -1-re állítja az adott Player pontjait, majd a Game UpdateScrore() függvénye kiveteti a játékból.
* **void Kill():** Ha meghívják meghívja a saját Field-je Remove-jat így gyakorlatilag “meghal”.
* **void Press():** Ez a függvény false-val hívja a gomb Activate() függvényét, mert ha player lép a gombra ő nem aktiválja.
* **Player GetLastTouchedMe():** Saját magát adja vissza.
* **void Die():** Meghívja a saját DeathScore() függvényét így megindítja a halálával járó eseményeket.
  + 1. **Target**
       - **Felelősség**

A Fieldből származó osztály. Csak a Box-okra aktiválódik a Player-eket figyelmen kívül hagyja. Ha egy Box rákerül az eltűnik és az utolsó lökést adó játékosnak pontot ad. Természetesen a Target mezőn is ugyan úgy összenyomódhat a Player mint bárhol máshol.

* + - * **Ősosztályok**

Field → Target

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **Field[] neighbours**
* **Moveable myMoveable**
  + - * **Metódusok**
* **void FieldAction():** Rátoláskor a Box eltűnik és az utolsó Playernek aki hozzá ért pont írást kezdeményez.

**4.3.11 TrapDoor**

* + - * **Felelősség**

A Field leszármazottja. Egy hozzá tartozó gomb utasítására nyílik ki és viselkedik úgy, mint egy Hole, vagy csukódik be és egy sima Field-ként létezik.

* + - * **Ősosztályok**

Field → Hole → TrapDoor

* + - * **Interfészek**

Nincs

* + - * **Attribútumok**
* **boolean isActive**
* **Field[] neighbours**
* **Moveable myMoveable**
  + - * **Metódusok**
* **void SetActive(boolean: b):** Átállítja a TrapDoor állását. Nyitottból csukott, csukottból nyitott.
* **void FieldAction():** Aktiválódáskor kinyílik, és ha éppen egy Player vagy egy Box volt rajta, akkor azt törli. Ha a hozzá tartozó Button-ról letolják a Box-ot akkor bezárul és sima Field lesz amire midnen következmény nélkül rá lehet lépni vagy tolni.

**4.3.12 Wall**

* + - * **Felelősség**

A Box-ból öröklődik. Sem a Player-ek sem a Box-ok nem képesek áthaladni rajta. Áthatolhatatlanok a játék kezdetétől a végéig. Úgy viselkednek mint egy Box ami mindíg stuck.

* + - * **Ősosztályok**

Moveable → Box → Wall

* + - * **Interfészek**

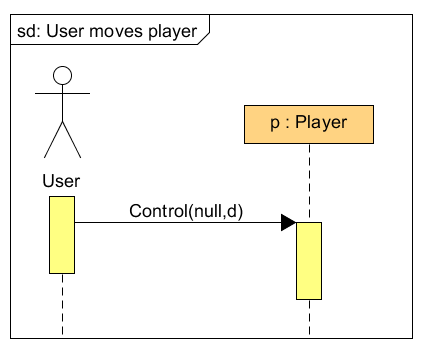
Nincs

* + - * **Attribútumok**
* **boolean stuck**
  + - * **Metódusok**
* **boolean Control(Player m, Direction d):** Mindig biztosan false-val tér vissza hiszen nem lehet rajta át menni és eltolhatatlan is.
* **boolean IsStuck():** Mindig biztosan true-val tér vissza hiszen folyton mozdíthatatlan.

***Szekvencia diagramok***

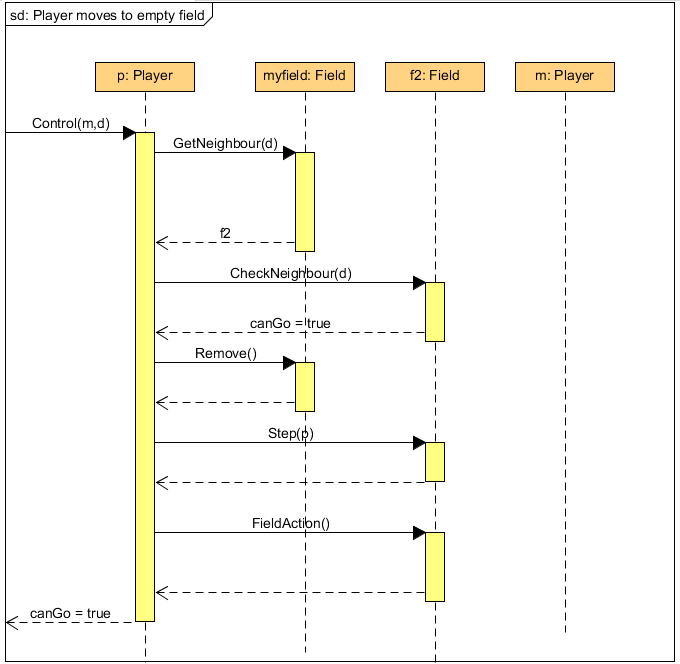
***4.4.1 User moves player***

A felhasználó mozgatja az egyik játékost. Fontos, hogy itt nem egy másik térképen lévő objektum “löki” meg a játékost - ezért a Controll első átvett értéke *null*.

******

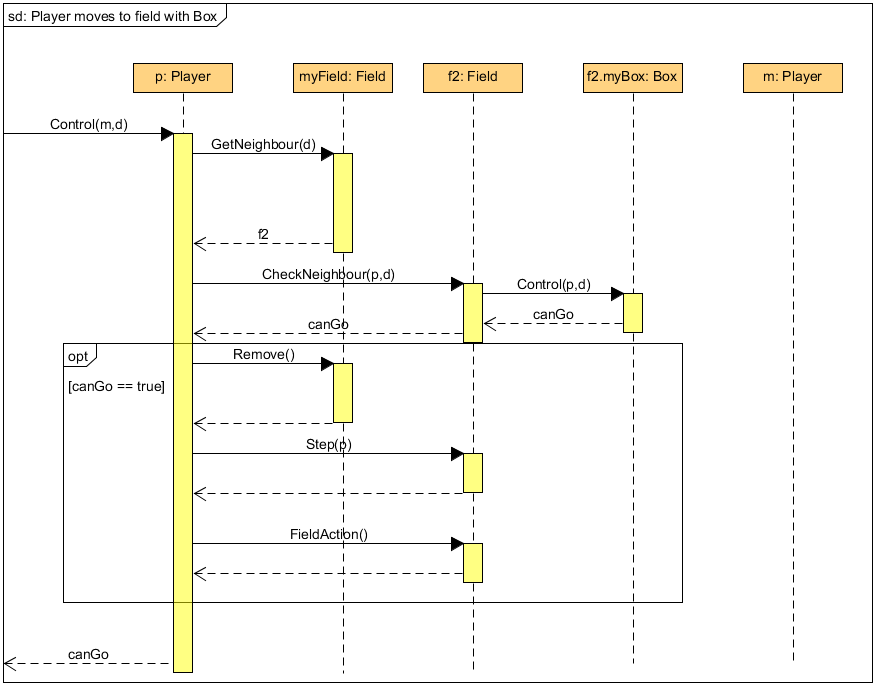
***4.4.2 Player moves to empty field***

A játékost valami (akár a user akár pl egy doboz) elmozgatja egy irányba, ahol üres mező taláható. Az ábrán nincs lekezelve az az eset, amikor van valami a szomszéd mezőn - ezt külön diagramon kezeljük. Az m értéke vagy a mozgást kezdeményező Player vagy null ha a játékos kezdte.

******

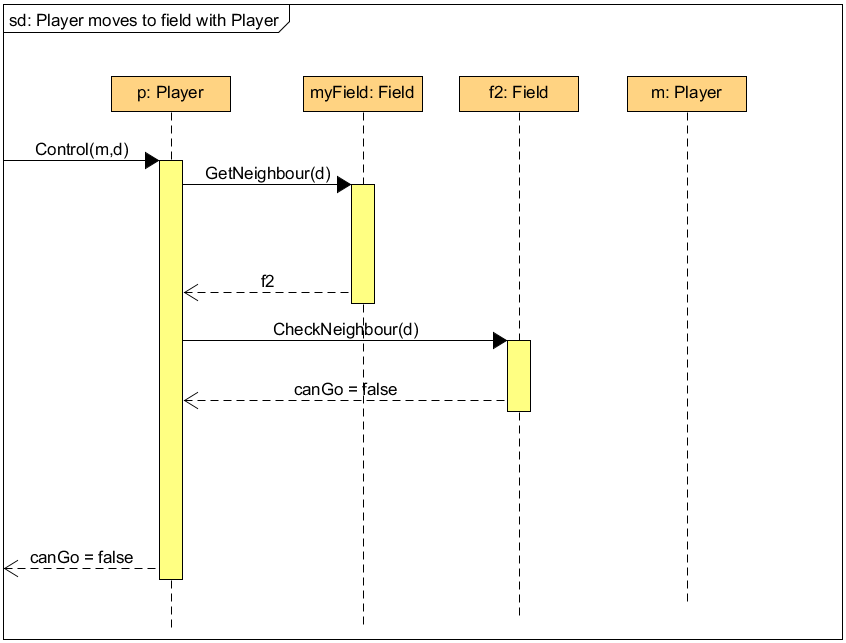
***4.4.3 Player moves to field with box***

A játékos olyan mezőre lép, amin van egy doboz. Ha a játékost a felhasználó mozgatta, akkor a Control első paramétere null, ha valamilyen másik objektum, akkor a mozgást kezdeményező Player. Ha a láda által visszaadott canGo érték igaz, akkor a játékos eltolhatja a dobozt és a lépés végbemegy. A doboznak saját magát adja a Controll-ban, így a dobozok mindig arról a Player-ről tudnak aki “utoljára ért bele” a mozgásba.

******

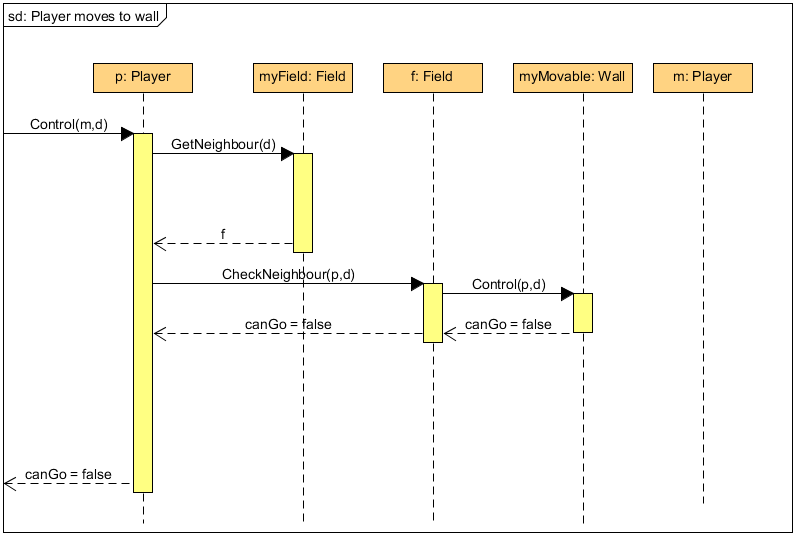
***4.4.4 Player moves to field with player***

A játékost a felhasználó vagy egy másik objektum egy olyan mezőre mozgatná, amin egy másik játékos áll. (Itt is a Controll függvény első paramétere ettől függően vagy null vagy a “megtoló” Player.)

******

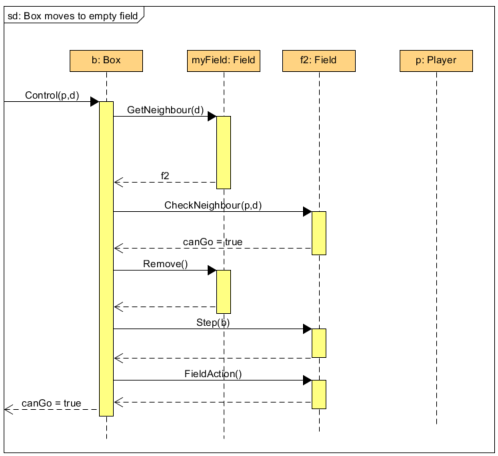
***4.4.5 Player moves to wall***

A Player egy falra akar lépni amit nem tehet, itt a Palyer lép Player-re esethez hasonlóannem fog oda lépni. (Itt is a Controll függvény első paramétere ettől függően vagy null vagy a “megtoló” Player.)



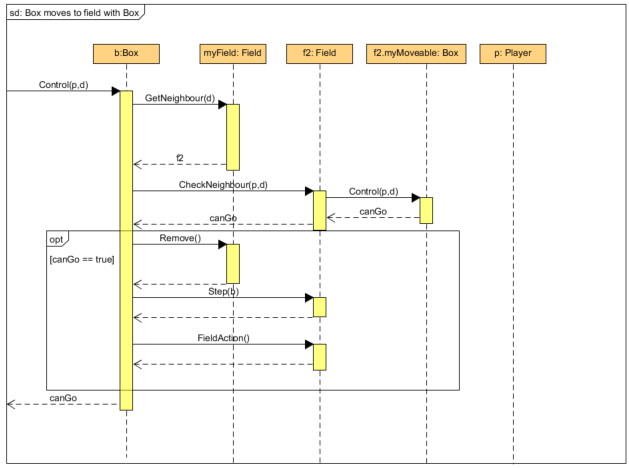
***4.4.6 Box moves to empty field***

Egy dobozt egy szomszédos üres mezőre tolják. Ebben a diagramban csak azt az esetet kezeljük, amikor a szomszéd mező üres. Mivel egy dobozt csak egy objektum tolhat el, ezért p értéke sose null.

******

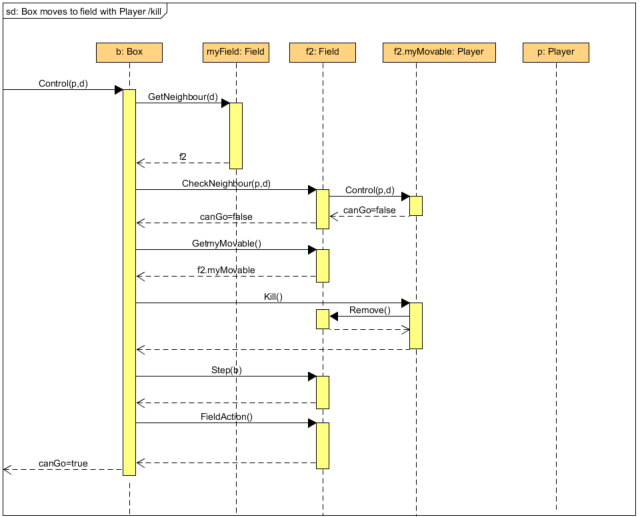
***4.4.7 Box moves to field with box***

A dobozt egy olyan mezőre tolják, amin szintén egy doboz van. Rekurzíve megnézi, hogy tud-e lépni az adott irányba. Ha igen, akkor lép. Mivel egy dobozt csak egy objektum tolhat el, ezért p értéke sose null.

******

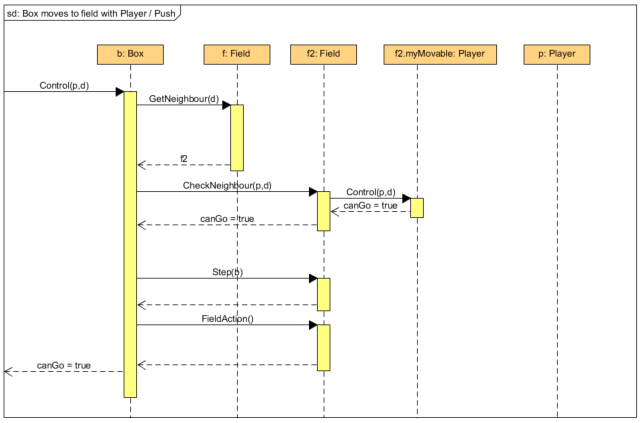
***4.4.8 Box moves to field with player that can not move on***

A dobozt egy olyan mezőre tolják, ahol egy másik játékos van. Az adott játékos, viszont nem tud mozogni arra, amerre tolják, ezért a láda összenyomja és meghal.

******

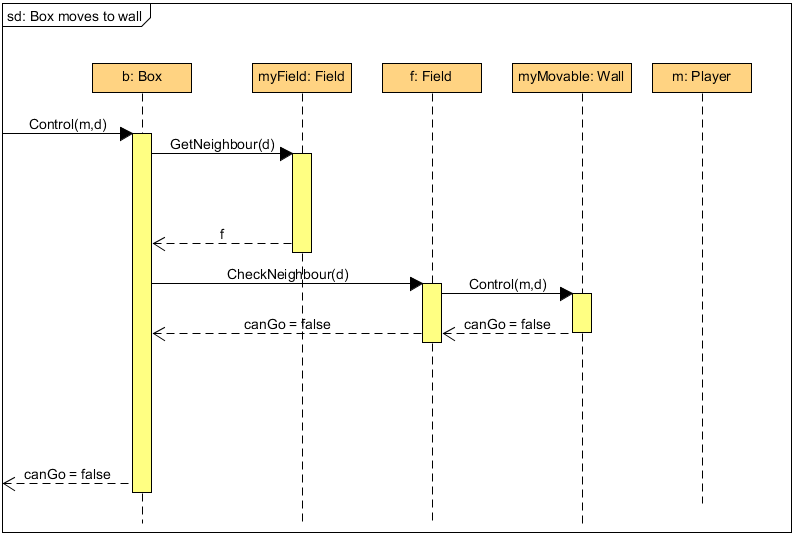
***4.4.9 Box moves to field with Player that can move on***

A Box-ot egy olyan Field-re tolják, amin van egy játékos. A játékos az aktuális irányba tovább tud mozogni, így a doboz eltolja.



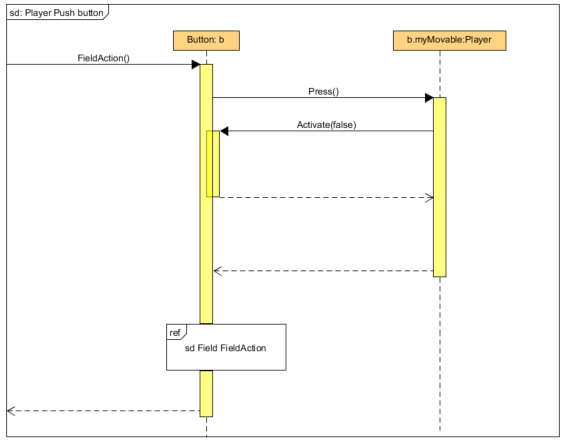
***4.4.8 Box moves to wall***

A dobozt egy falra akarják tolni - amit nem tehet - tehát itt a canGo false-al jön vissza és a lépés nem megy végbe.



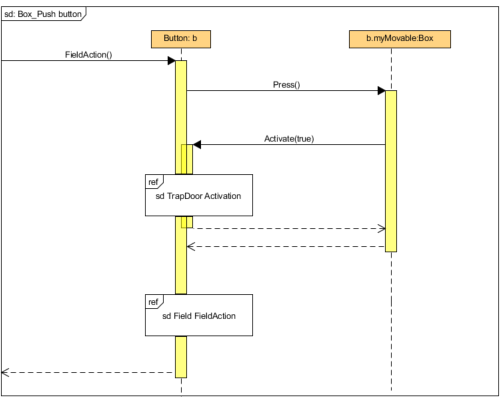
***4.4.9 Player pushes button***

Egy Player lép egy Button-ra. Meghívja a Button a Player press függvényét. Mivel egy Player nem tud megnyomni egy Buttont ezért “nem történik semmi”, nem aktiválódik a gomb (az Activatet false paraméterrel hívja).



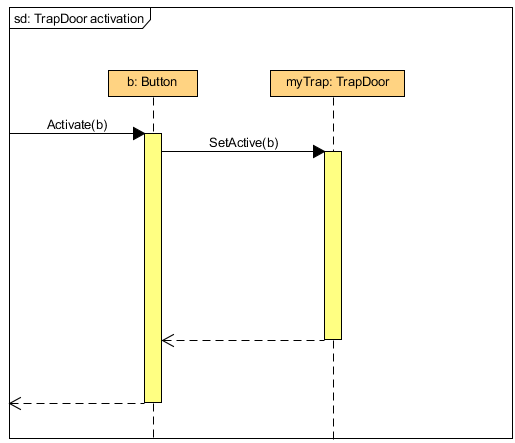
***4.4.10 Box pushes button***

A doboz egy olyan mezőre kerül, amin egy gomb van. Meghívódik a Button Activate függvénye a rajta lévő objektumnak megfelelő boolean értékkel. (a Box megnyomja a Player nem.)



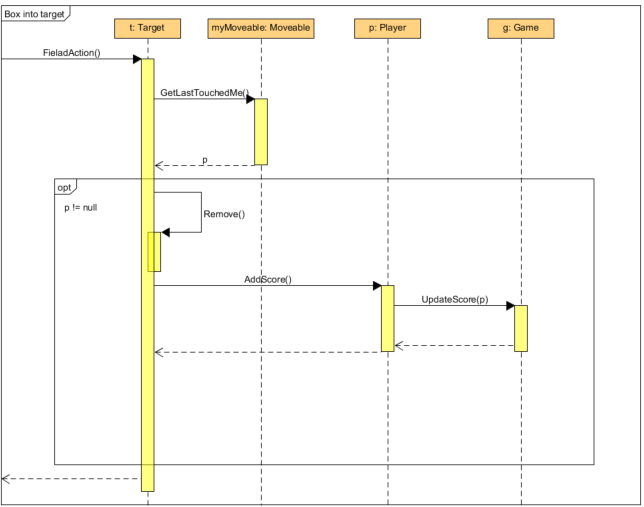
***4.4.11 TrapDoor activation***

Ha egy gombra rátolnak egy dobozt, az aktivál egy TrapDoort. Beállítja a gombon az aktivitást, ami meghívja az általa tartalmazott TrapDoor aktiváló függvényét, ami ez után kinyílik és Hole-ként viselkedik. Ha a doboz le kerül a csapó ajtó be zárul.

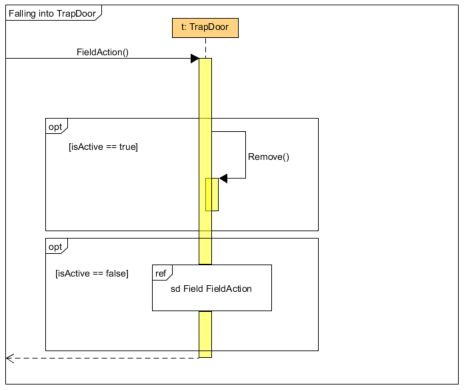
******

***4.4.12 Move box into target***

Egy doboz célba tolása. Megnézi a Field ki tolta meg az adott objektumot. Ha ez az objektum egy doboz, akkor azt eltünteti a pályáról, illetve az azt megtoló játékosnak jóváírja a pontot.

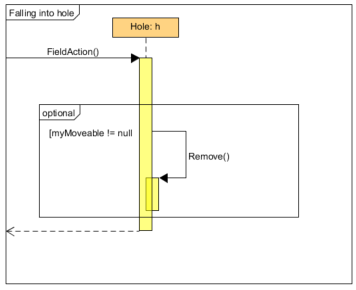
******

***4.4.13 Falling into TrapDoor***

******

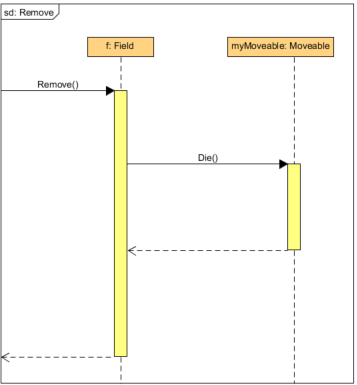
***4.4.14 Falling into hole***

Mikor egy objektum egy Hole típusú Fieldre lép, akkor”leesik” és eltűnik a pályáról.

******

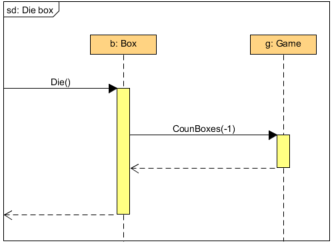
***4.4.15 Objektum eltüntetése Fieldről***

A Fieldre meghívjuk a Remove() függvényt, ami a rajta tartózkodó Moveable objektum Die függvényét hívja meg és kiveszi azt a játékból.

******

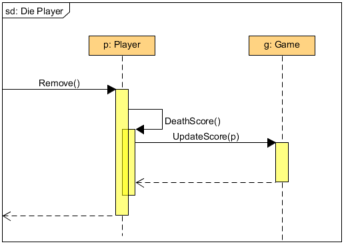
***4.4.16 Box’s Die function***

Csökkentjük az adott játékban lévő dobozok számát ha a doboz “meghal”.



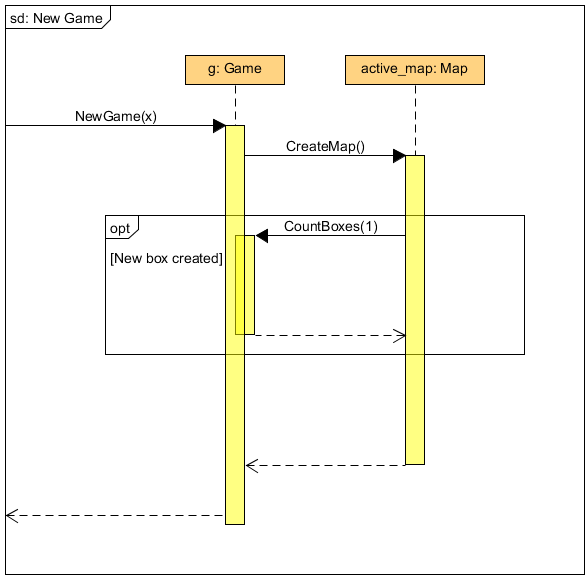
***4.4.17 Player’s Die function***

A játékosnak meghívjuk a DeathScore függvényét ezzel a pontszámát -1 -re állítva. Az UpdateScore függvény ekkor törli a játékost az adott játékból.

******

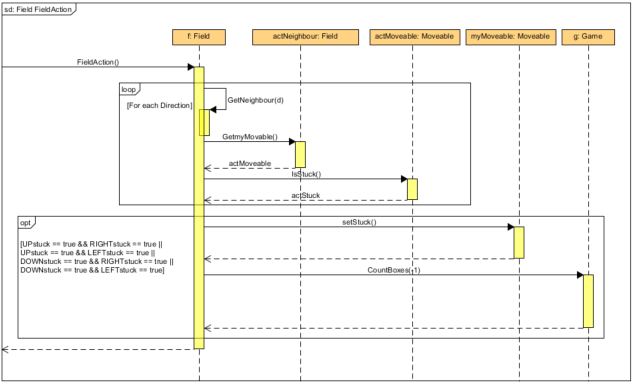
***4.4.18 New Game***

Létrehoz egy játékot, legenerálja a Map elemeit.

******

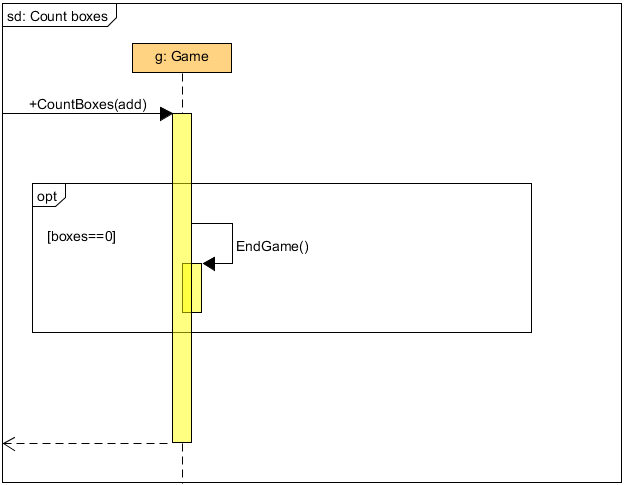
***4.4.19 Field Field Action***

Minden irányba megnézzük hogy tolható-e. Ha a doboz “beszorult” akkor csökkenti a pályán lévő dobozok számát. Amikor a dobozok elfogynak a játéknak vége.

******

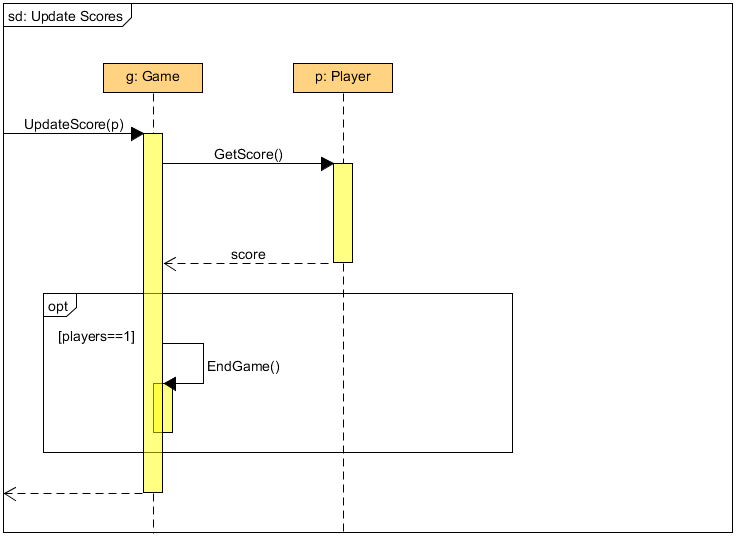
***4.4.20 Count Boxes***

A dobozok számát mindig a megadott értékkel “növeli” ha a dobozok elfogynak tehát nincs több mozgatható akkor a játék véget ér.

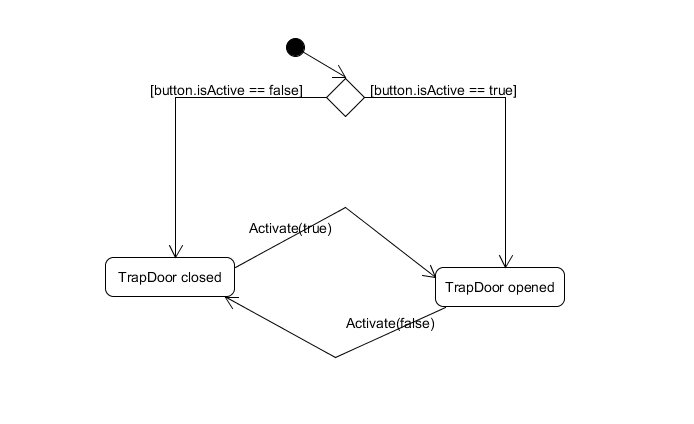
******

***4.4.21 Update Scores***

A játékosok pontját a DeathScore() -1-re állítja, ha ezt érzékeli az UpdateScore akkor csökkenti a Playerek számát. Ha a playerek száma 1 akkor az utolsó játékos nyer és a játék véget ér.

******

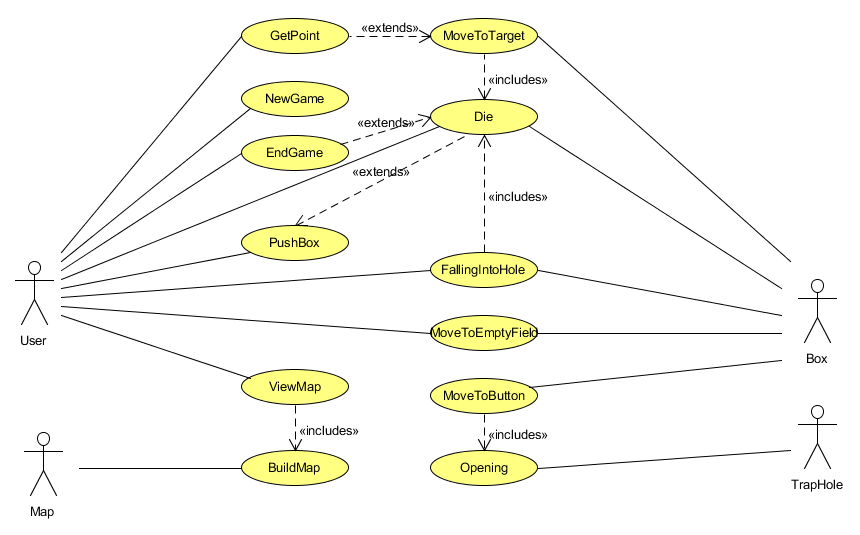
* 1. ***State-chartok***



***4.5 Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.03.01 10:00 | 1 óra | Czetter  Frontó  Tomori  Verasztó | Megbeszélés:  Objektumkatalógus megvitatása. Feladatok kiosztása |
| 2018.03.01 11:00 | 3 óra | Czetter  Frontó  Tomori  Verasztó | Osztályok és szekvencia diagrammok átgondolása, újratervezése. |
| 2018.03.02 12:00 | 2 óra | Czetter  Frontó  Tomori  Verasztó | Osztályok és szekvencia diagrammok összehangolása, javítása. |
| 2018.03.04 15:00 | 5 óra | Czetter  Frontó  Tomori  Verasztó | Ellenőrzés, javítás dokumentáció írása és lektorálása. |

1. **Szkeleton tervezése**
   1. ***A szkeleton modell valóságos use-case-ei***
      1. **Use-case diagram**



* + 1. **Use-case leírások**

|  |  |
| --- | --- |
| **Use-case neve** | NewGame |
| **Rövid leírás** | Új játékot indít. |
| **Aktorok** | User |
| **Forgatókönyv** | Miután megadtuk a játékosokot számát, egy teljesen új játék indul a frissen generált pályán. |

|  |  |
| --- | --- |
| **Use-case neve** | EndGame |
| **Rövid leírás** | A játék véget ér. |
| **Aktorok** | User |
| **Forgatókönyv** | Ha a játékban csak egy játékos maradt, vagy az utolsó dobozt is olyan helyre tolták, ahonnan már nem lehet kiszedni, a játék befejeződik és az utóbbi esetben a legtöbb ponttal rendelkező Player nyer. |
| **Use-case neve** | Die |
| **Rövid leírás** | Valamilyen oknál fogva bekövetkezik a halál. |
| **Aktorok** | User, Box |
| **Forgatókönyv** | User halála bekövetkezhet úgy, hogy leesik egy Hole-ba vagy egy TrapDoor-ba, továbbá ha rátolnak egy másik Box-ot úgy, hogy Ő maga nem tud hová tolódni. A Box-ok pedig meghalnak úgyan úgy, ha Hola-ba, TrapDoor-ba vagy a Target-be esnek. |

|  |  |
| --- | --- |
| **Use-case neve** | GetPoint |
| **Rövid leírás** | A Player pontot kap. |
| **Aktorok** | User |
| **Forgatókönyv** | Mivel egy Box-ot sikeresen betolt a célba, ezért pont jár az adott Player-nek. |

|  |  |
| --- | --- |
| **Use-case neve** | MoveToTarget |
| **Rövid leírás** | Egy Box a Target-be ér. |
| **Aktorok** | Box |
| **Forgatókönyv** | Egy Box-ot a Target-be toltak ilyenkor eltűnik és az eltűnésével együtt pont jár az utolsó lökést adó Playernek. |

|  |  |
| --- | --- |
| **Use-case neve** | PushBox |
| **Rövid leírás** | Egy Player megtol egy Box-ot |
| **Aktorok** | User |
| **Forgatókönyv** | Egy Player egy adott irányba próbál lökni egy Box-ot. Ez sikerülhet neki ha nem fal van a Box előtt, továbbá akár egy másik Player-t is megölhet vele, ha az nem tud arrébb lépni. |

|  |  |
| --- | --- |
| **Use-case neve** | FallingIntoHole |
| **Rövid leírás** | Beleesnek egy Hole-ba vagy TrapDoor-ba. |
| **Aktorok** | User, Box |
| **Forgatókönyv** | Vagy a Player szabad akaratából vagy közvetve lökés miatt beleesik egy lyukba. Ilyenkor meghal és az elért Score-ja törlődik. Egy Box ugyanúgy meghal, ha Hole-ba esik. |

|  |  |
| --- | --- |
| **Use-case neve** | MoveToEmptyField |
| **Rövid leírás** | Egy olyan Field-re lépnek ami számukra nincs hatással. |
| **Aktorok** | User, Box |
| **Forgatókönyv** | Egy sima egyszerű lépés vagy tolás, a megadott irányba. Amire kerülnek Field, az se a Player-re se a Box-ra semmilyen játékbeli hatással nincs. |

|  |  |
| --- | --- |
| **Use-case neve** | MoveToButton |
| **Rövid leírás** | Benyomja a Field-en lévő Button-t. |
| **Aktorok** | Box |
| **Forgatókönyv** | Egy olyan Field-re kerül, ahol egy Button-on van. Ilyenkor az benyomódik és a hozzá tartozó TrapDoor kinyílik. A TrapDoor Hole-ként viselkedik. |

|  |  |
| --- | --- |
| **Use-case neve** | ViewMap |
| **Rövid leírás** | Látja a Map-et. |
| **Aktorok** | User |
| **Forgatókönyv** | Látható számára az egész generált Map. Minden Box, Hole és Button továbbá látható számára a többi játékos is és a Target. |

|  |  |
| --- | --- |
| **Use-case neve** | BuildMap |
| **Rövid leírás** | Map építés. |
| **Aktorok** | Map |
| **Forgatókönyv** | A Map, előre meghatározott paraméterek alapján, minden egyes indításkor random generálja a pályát. Random az elhelyezkedése a Map-en megjelenő Box-oknak, Button-oknak Hole-oknak és a Target-nek is. |

|  |  |
| --- | --- |
| **Use-case neve** | Opening |
| **Rövid leírás** | TrapDoor nyílás. |
| **Aktorok** | TrapDoor |
| **Forgatókönyv** | A Button-hoz tartozó TrapDoor kinyílik. |

* 1. ***A szkeleton kezelői felületének terve, dialógusok***

1. Menü választás
   1. Mit akarunk tenni? Új Játék/ Kilépés
2. Játékosok számának meghatározása. (1-től 4-ig)
3. Pálya mezőinek generálása.
4. Mit akarsz rakni? Field/Hole/Button/TrapDoor/Target
   1. Mit akarsz rakni? Field/Hole/Button/TrapDoor/Target
      1. Mi legyen a mezőn? Box/Player/Wall/Üres
5. Játékos lép
   1. Hova lép? (UP,DOWN,LEFT,RIGHT)
      1. Mi van itt? Field/Hole/Button/TrapDoor/Target
      2. Van rajta valaki? Box/Player/Wall/Üres
      3. Ha doboz eltudom tolni? I/N \*
      4. Ha TrapDoor  nyitva van e? I/N\*
6. Játékos tol
   1. Mit tolok? Box/Player/Wall
      1. Ha doboz hova tolom? Field/Hole/Button/TrapDoor/Target \*
         1. Van rajta valaki? Box/Player/Wall/Üres
            1. Ha doboz eltudom tolni? I/N
            2. Ha játékos eltudom tolni? I/N
7. Játék befejezése
   1. Van 1-nél több játékos a pályán? I/N
   2. Van tolható doboz a pályán? I/N

A \* jelölést olyan kérdések kapták melyeknek a megjelenése csak bizonyos esetekben fordul elő, ha az előző kérés olyan választ kapott.

Az alábbi példa a játék indítását és a pálya generálásának első lépését mutatja be. Felhasználó választhatja ki a menüpontot (új játék, vagy kilépés). Új játék után megadhatjuk a játékosok számát, majd a pályagenerálást szimuláljuk. Kiválasztjuk, hogy melyik eset működését szeretnénk megnézni. Minden sor elején jelöljük, hogy az adott sor milyen interakciót jelent.

*create        -->Game():*

*create            -->Map():*

*return            <--Map()*

*return        <--Game()*

*-         1. Uj Jatek*

*-         2. Kilepes*

*choice    Mit akarsz csinalni? (1-2):1*

*choice    Hany jatekos legyen? (1-4):2*

*call          -->Game.NewGame(2):*

*call              -->Map.CreateMap(2):*

*-                 1. Field*

*-                 2. Hole*

*-                 3. TrapDoor*

*-                 4. Button*

*-                 5. Target*

*-                 6. Vegeztem az epitessel*

*choice            Milyen mezot akarsz rakni? (1-6):*

*•’choice’: kérdés, felhasználói interakcióra vár*

*• ’-’: opció*

*• ’call’: metódus hívás*

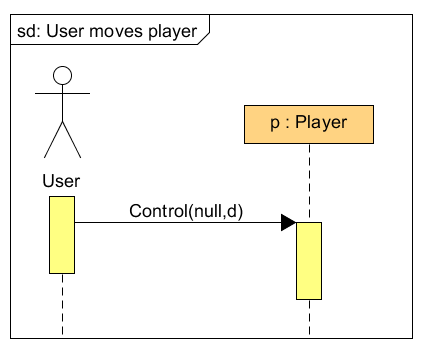
*• ’return’: metódusból visszatérés*

*• 'create‘: konstruktor hívás*

* 1. ***Szekvencia diagramok a belső működésre***

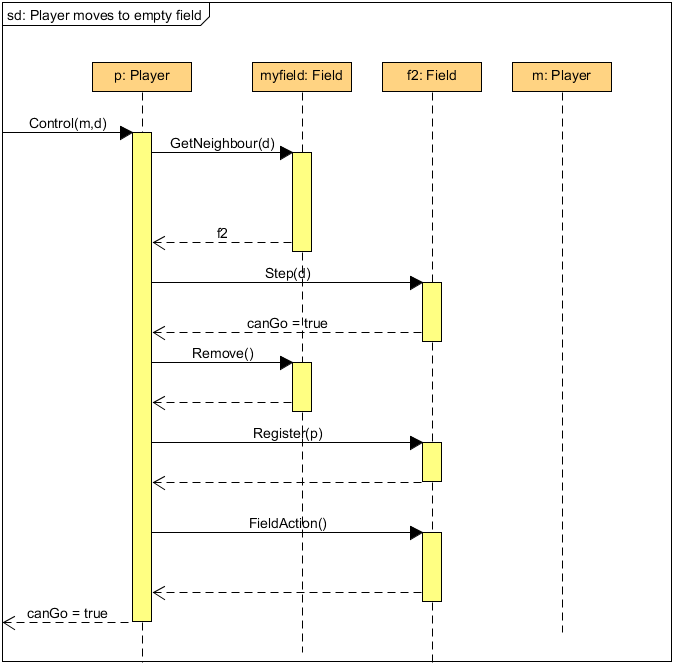
***5.3.1 User moves player***

A felhasználó mozgatja az egyik játékost. Fontos, hogy itt nem egy másik térképen lévő objektum “löki” meg a játékost - ezért a Controll első átvett értéke *null*.

******

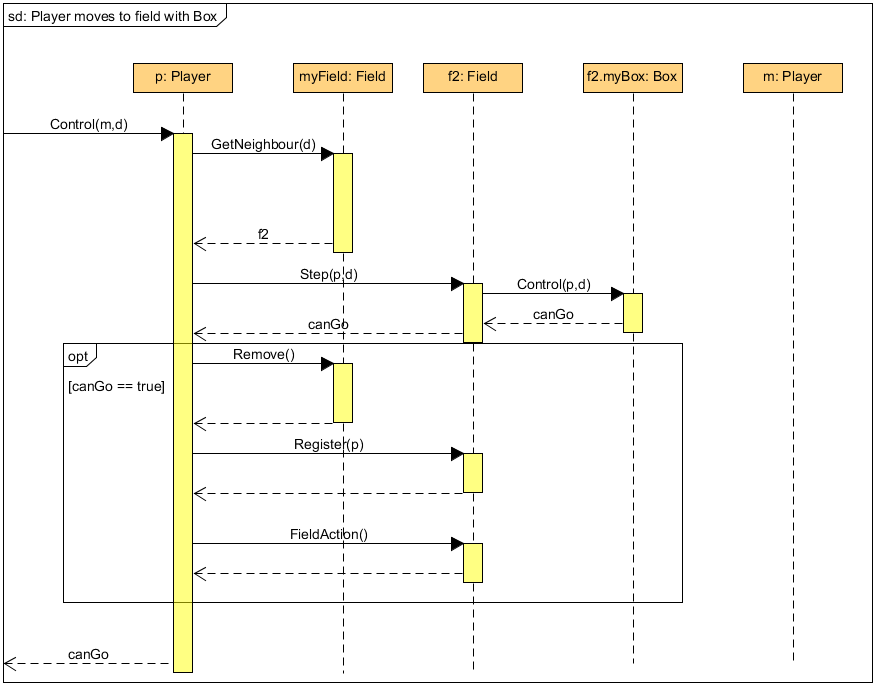
***5.3.2 Player moves to empty field***

A játékost valami (akár a user akár pl egy doboz) elmozgatja egy irányba, ahol üres mező taláható. Az ábrán nincs lekezelve az az eset, amikor van valami a szomszéd mezőn - ezt külön diagramon kezeljük. Az m értéke vagy a mozgást kezdeményező Player vagy null ha a játékos kezdte.

******

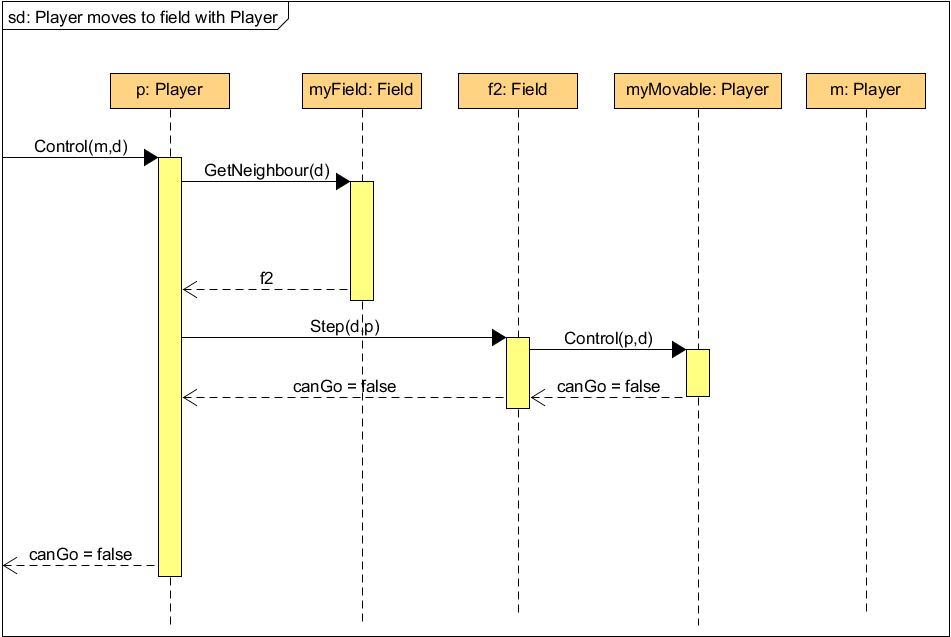
***5.3.3 Player moves to field with box***

A játékos olyan mezőre lép, amin van egy doboz. Ha a játékost a felhasználó mozgatta, akkor a Control első paramétere null, ha valamilyen másik objektum, akkor a mozgást kezdeményező Player. Ha a láda által visszaadott canGo érték igaz, akkor a játékos eltolhatja a dobozt és a lépés végbemegy. A doboznak saját magát adja a Controll-ban, így a dobozok mindig arról a Player-ről tudnak aki “utoljára ért bele” a mozgásba.

******

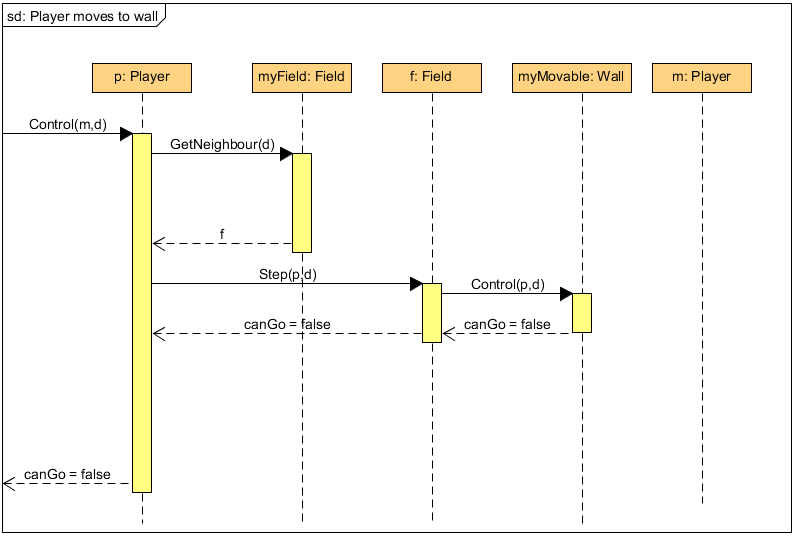
***5.3.4 Player moves to field with player***

A játékost a felhasználó vagy egy másik objektum egy olyan mezőre mozgatná, amin egy másik játékos áll. (Itt is a Controll függvény első paramétere ettől függően vagy null vagy a “megtoló” Player.)

******

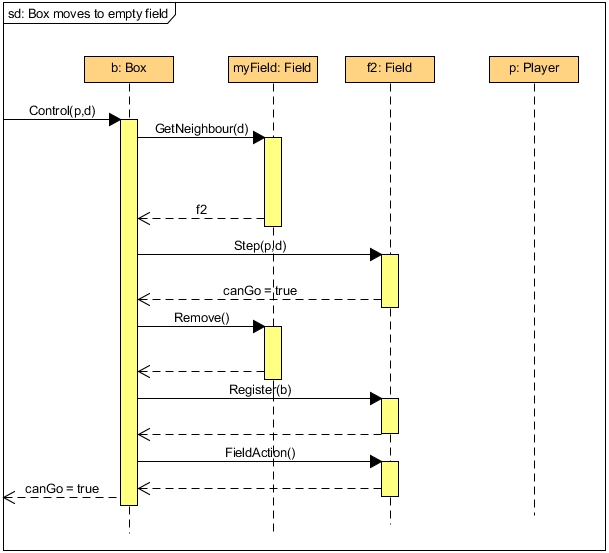
***5.3.5 Player moves to wall***

A Player egy falra akar lépni amit nem tehet, itt a Palyer lép Player-re esethez hasonlóannem fog oda lépni. (Itt is a Controll függvény első paramétere ettől függően vagy null vagy a “megtoló” Player.)



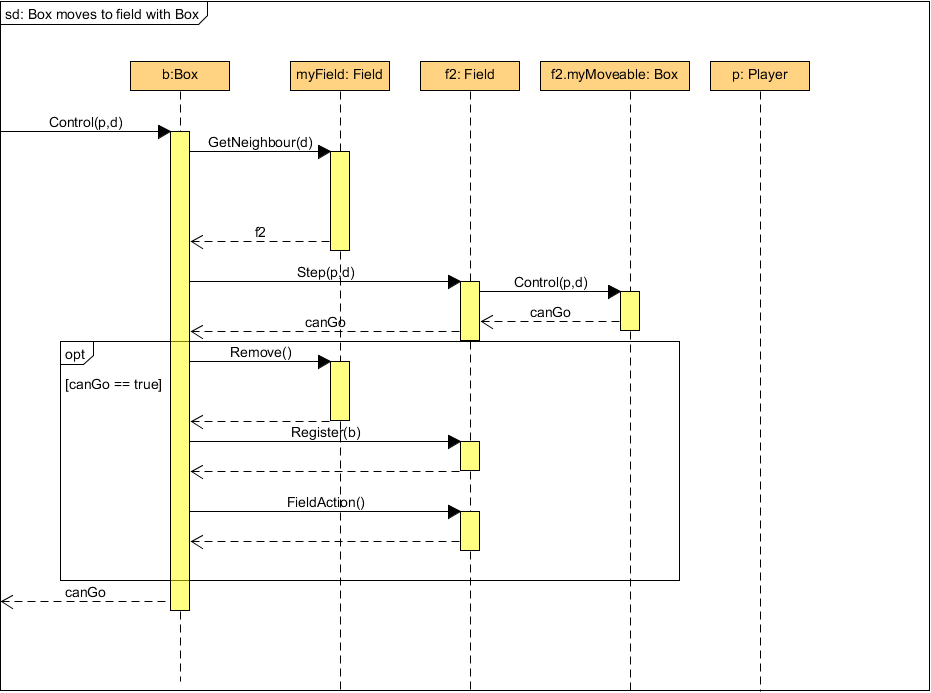
***5.3.6 Box moves to empty field***

Egy dobozt egy szomszédos üres mezőre tolják. Ebben a diagramban csak azt az esetet kezeljük, amikor a szomszéd mező üres. Mivel egy dobozt csak egy objektum tolhat el, ezért p értéke sose null.

******

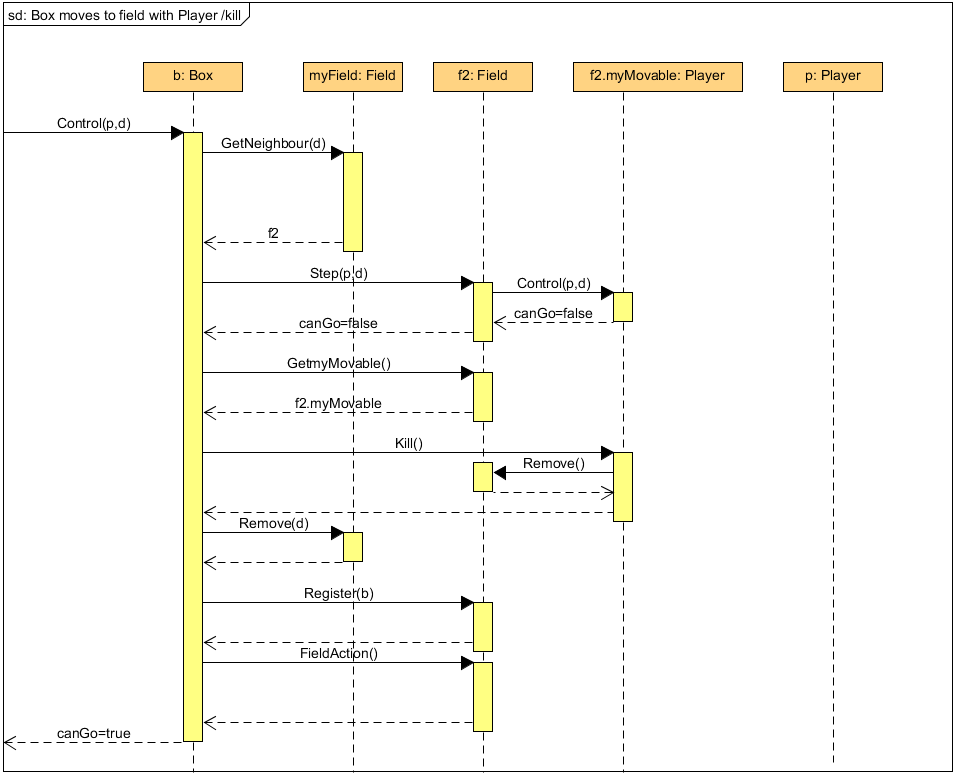
***5.3.7 Box moves to field with box***

A dobozt egy olyan mezőre tolják, amin szintén egy doboz van. Rekurzíve megnézi, hogy tud-e lépni az adott irányba. Ha igen, akkor lép. Mivel egy dobozt csak egy objektum tolhat el, ezért p értéke sose null.

******

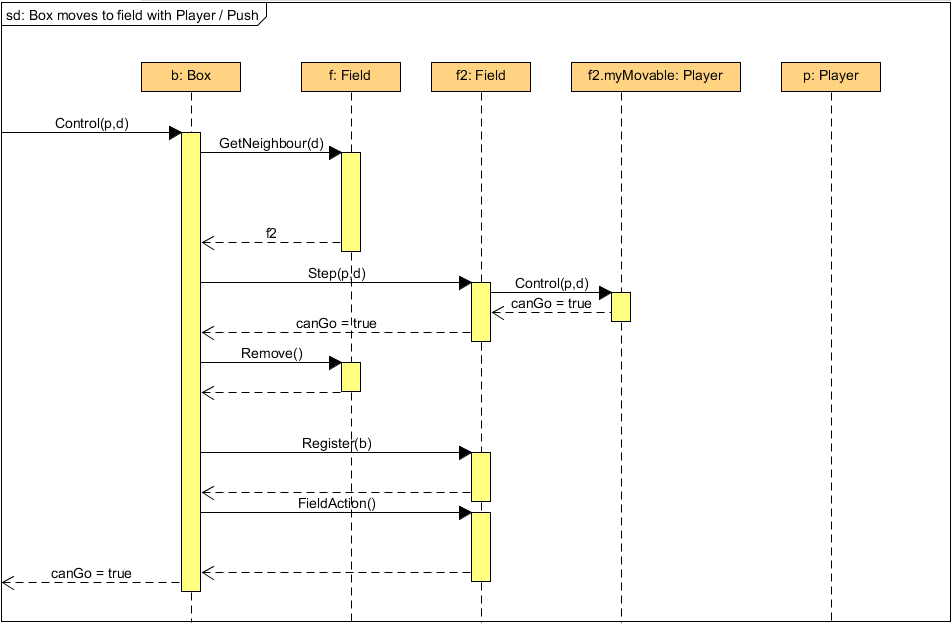
***5.3.8 Box moves to field with player that can not move on***

A dobozt egy olyan mezőre tolják, ahol egy másik játékos van. Az adott játékos, viszont nem tud mozogni arra, amerre tolják, ezért a láda összenyomja és meghal.

******

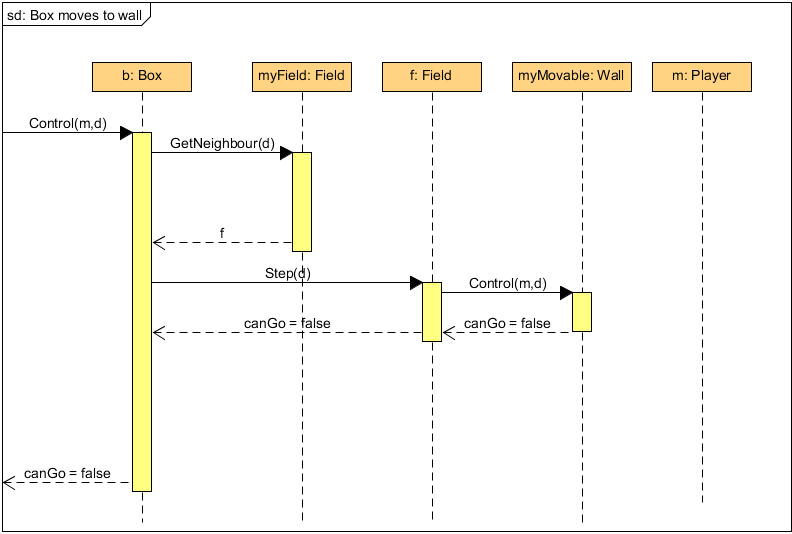
***5.3.9 Box moves to field with Player that can move on***

A Box-ot egy olyan Field-re tolják, amin van egy játékos. A játékos az aktuális irányba tovább tud mozogni, így a doboz eltolja.



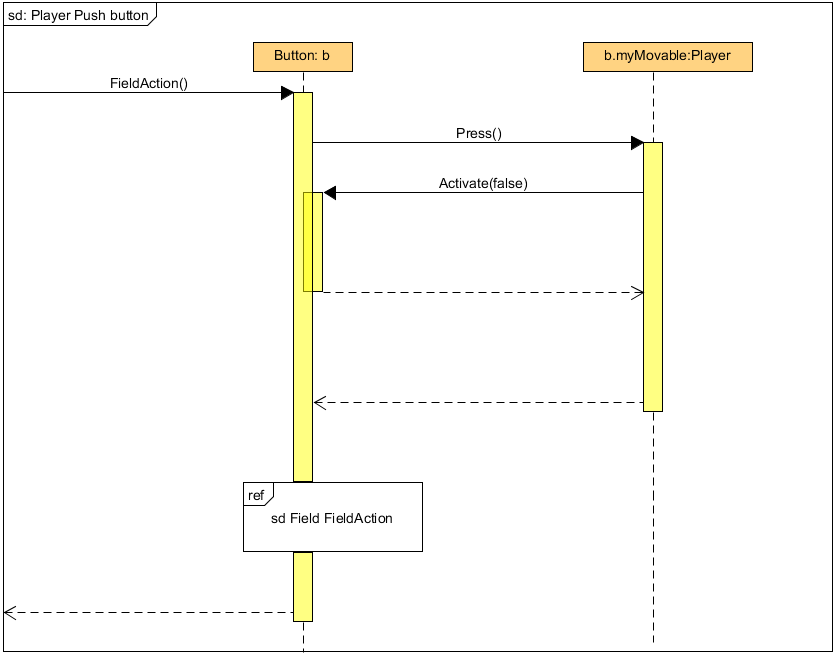
***5.3.8 Box moves to wall***

A dobozt egy falra akarják tolni - amit nem tehet - tehát itt a canGo false-al jön vissza és a lépés nem megy végbe.



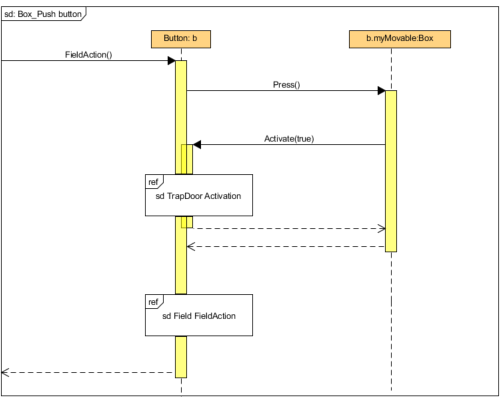
***5.3.9 Player pushes button***

Egy Player lép egy Button-ra. Meghívja a Button a Player press függvényét. Mivel egy Player nem tud megnyomni egy Buttont ezért “nem történik semmi”, nem aktiválódik a gomb (az Activatet false paraméterrel hívja).



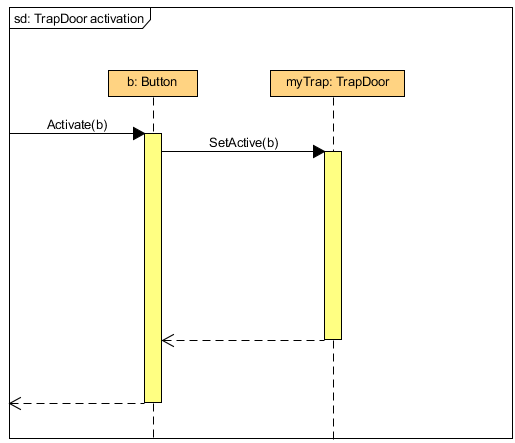
***5.3.10 Box pushes button***

A doboz egy olyan mezőre kerül, amin egy gomb van. Meghívódik a Button Activate függvénye a rajta lévő objektumnak megfelelő boolean értékkel. (a Box megnyomja a Player nem.)



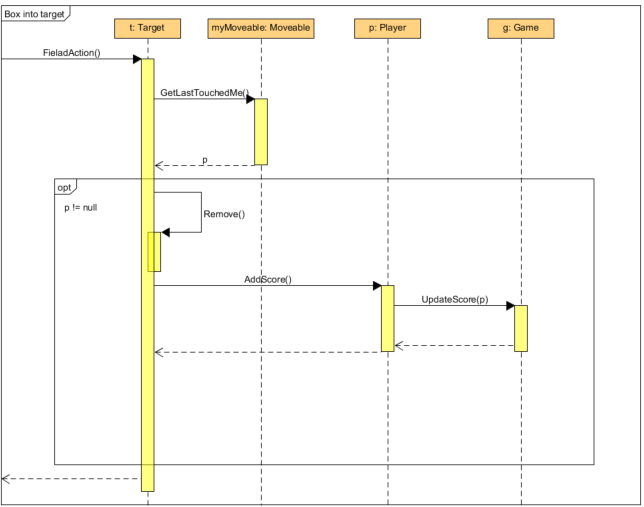
***5.3.11 TrapDoor activation***

Ha egy gombra rátolnak egy dobozt, az aktivál egy TrapDoort. Beállítja a gombon az aktivitást, ami meghívja az általa tartalmazott TrapDoor aktiváló függvényét, ami ez után kinyílik és Hole-ként viselkedik. Ha a doboz le kerül a csapó ajtó be zárul.

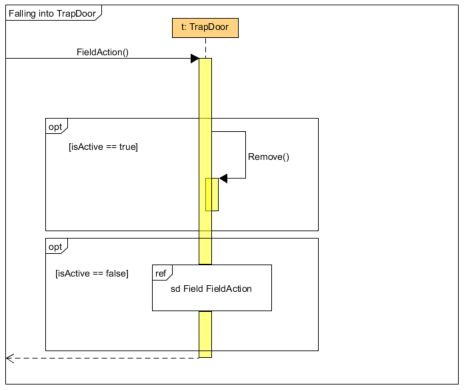
******

***5.3.12 Move box into target***

Egy doboz célba tolása. Megnézi a Field ki tolta meg az adott objektumot. Ha ez az objektum egy doboz, akkor azt eltünteti a pályáról, illetve az azt megtoló játékosnak jóváírja a pontot.

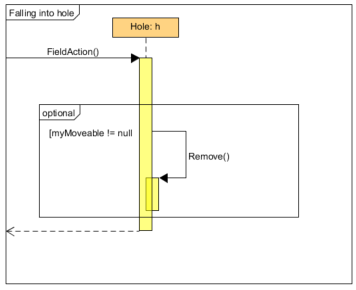
******

***5.3.13 Falling into TrapDoor***

******

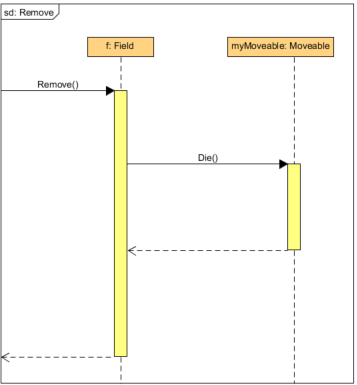
***5.3.14 Falling into hole***

Mikor egy objektum egy Hole típusú Fieldre lép, akkor”leesik” és eltűnik a pályáról.

******

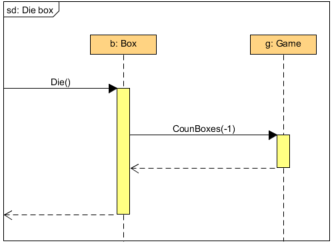
***5.3.15 Objektum eltüntetése Fieldről***

A Fieldre meghívjuk a Remove() függvényt, ami a rajta tartózkodó Moveable objektum Die függvényét hívja meg és kiveszi azt a játékból.

******

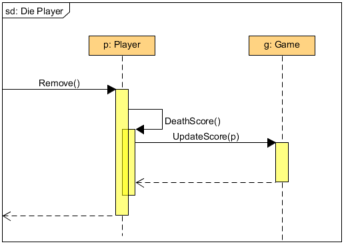
***5.3.16 Box’s Die function***

Csökkentjük az adott játékban lévő dobozok számát ha a doboz “meghal”.



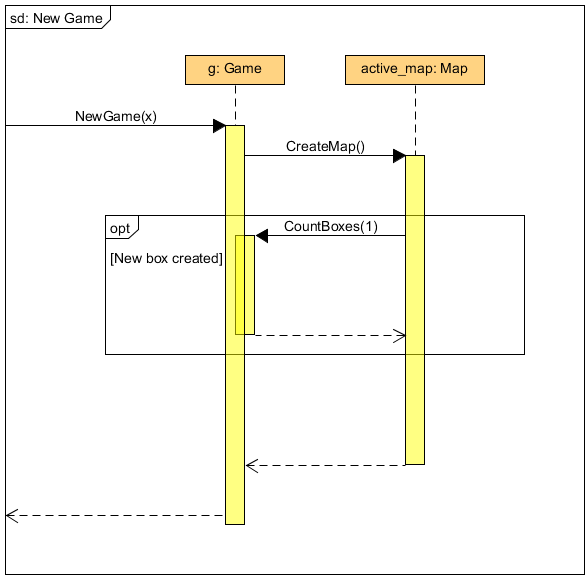
***5.3.17 Player’s Die function***

A játékosnak meghívjuk a DeathScore függvényét ezzel a pontszámát -1 -re állítva. Az UpdateScore függvény ekkor törli a játékost az adott játékból.

******

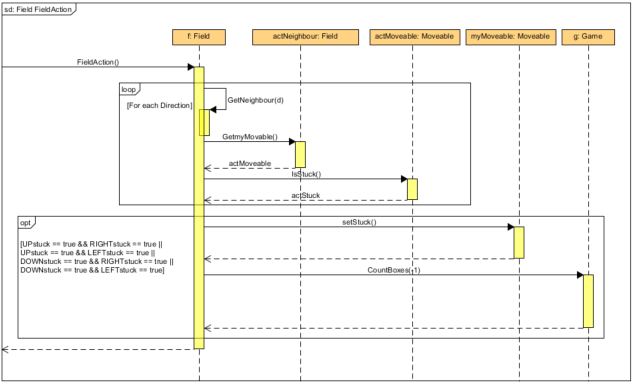
***5.3.18 New Game***

Létrehoz egy játékot, legenerálja a Map elemeit.

******

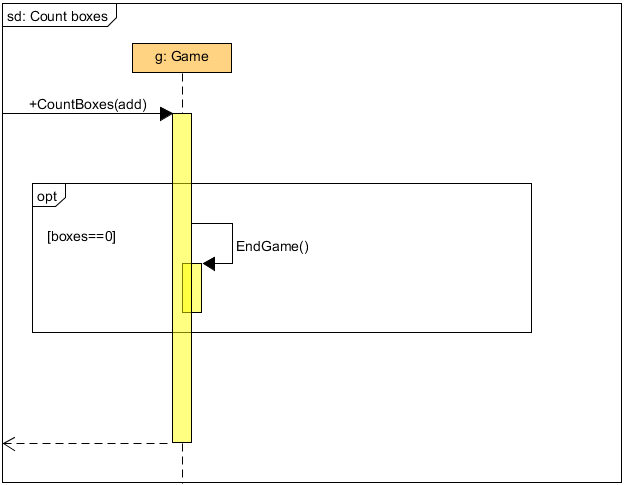
***5.3.19 Field Field Action***

Minden irányba megnézzük hogy tolható-e. Ha a doboz “beszorult” akkor csökkenti a pályán lévő dobozok számát. Amikor a dobozok elfogynak a játéknak vége.

******

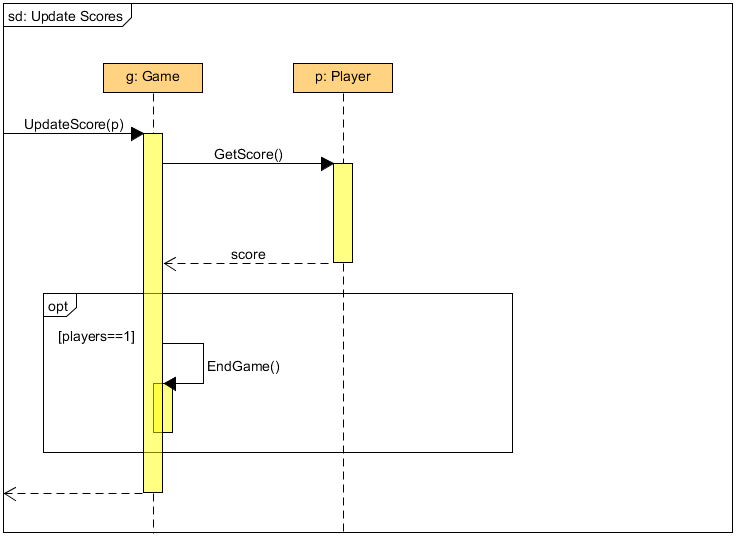
***5.3.20 Count Boxes***

A dobozok számát mindig a megadott értékkel “növeli” ha a dobozok elfogynak tehát nincs több mozgatható akkor a játék véget ér.

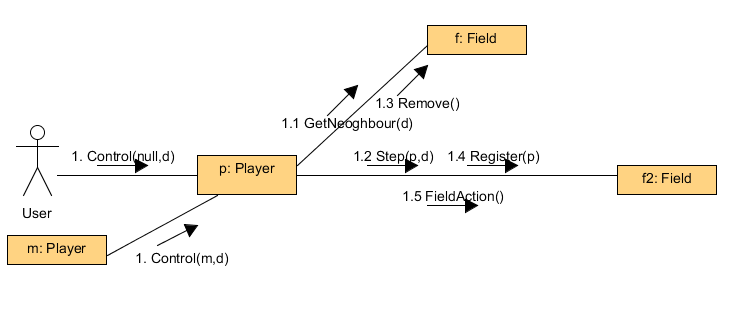
******

***5.3.21 Update Scores***

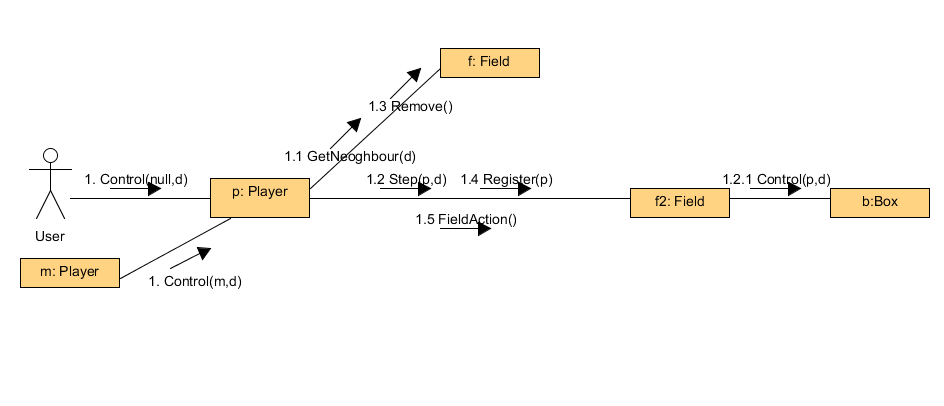
A játékosok pontját a DeathScore() -1-re állítja, ha ezt érzékeli az UpdateScore akkor csökkenti a Playerek számát. Ha a playerek száma 1 akkor az utolsó játékos nyer és a játék véget ér.

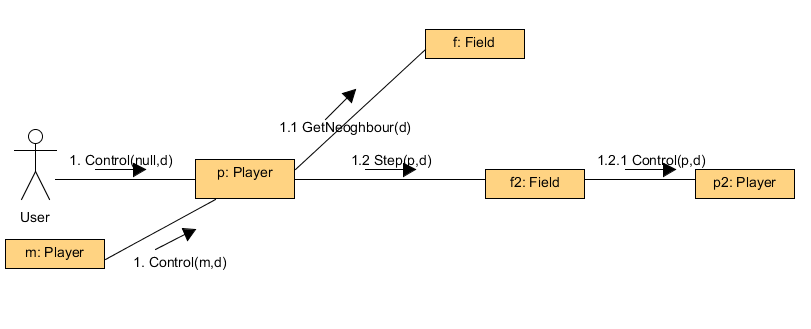
******

* 1. ***Kommunikációs diagramok***

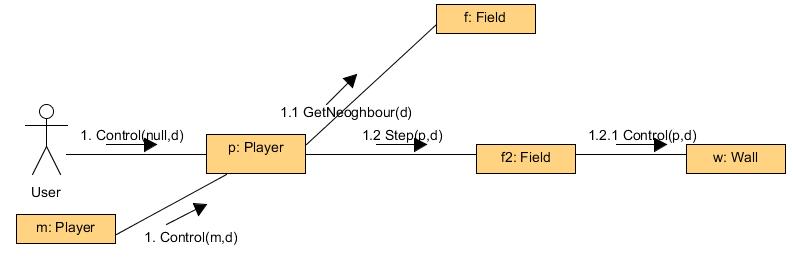


5.4.1 ábra: A Player üres Field-re lép.

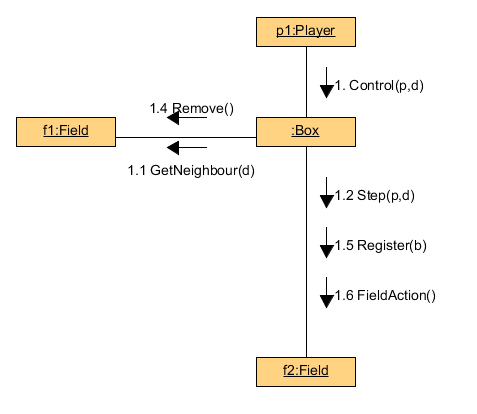
5.4.2 ábra: A Player olyan Fieldre lép, ahol Box áll, így tovább tolja.



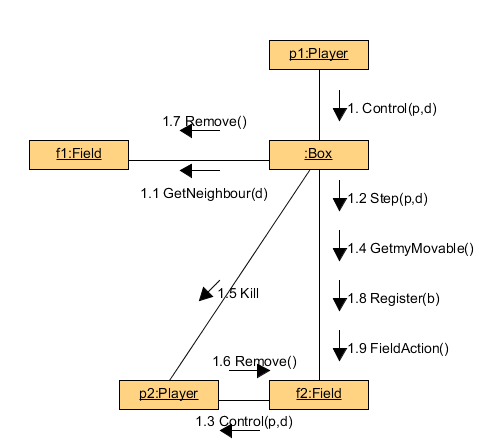
5.4.3 ábra: A Player olyan Field-re próbál lépni ahol egy másik Player áll.



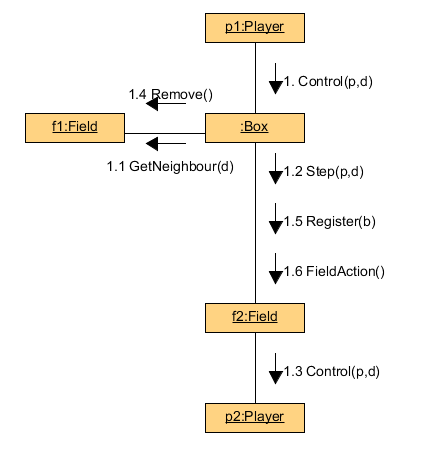
5.4.4 ábra: A Player Wall-nak ütközik.



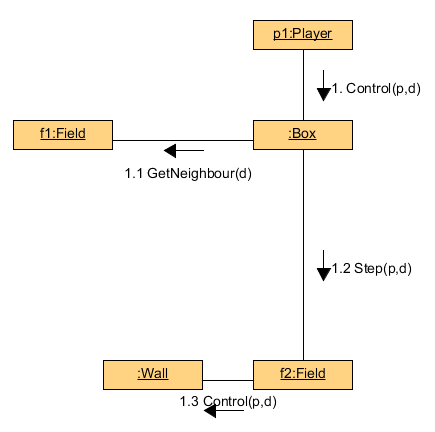
5.4.5 ábra: Box-ot tolnak üres Field-re



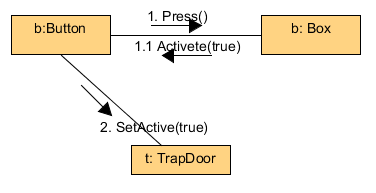
5.4.6 ábra: Box egy másik Player-nek ütközik és megöli.



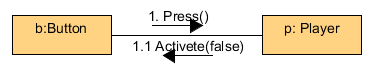
5.4.7 ábra: Box egy Player-nek ütközik és eltolja.



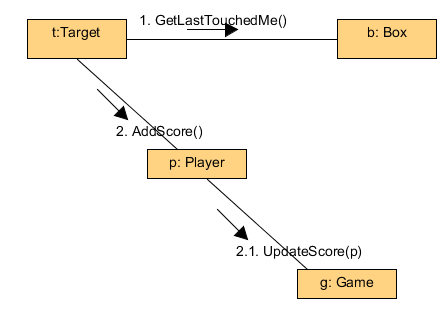
5.4.8 ábra: Box Wall-nak ütközik.



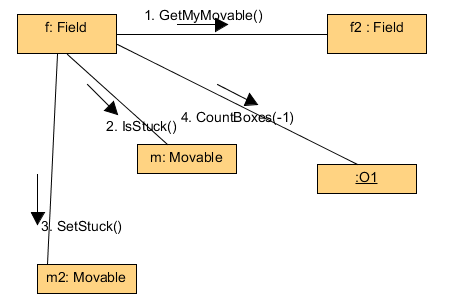
5.4.9 ábra: Box benyomja a Button-t és aktiválja a TrapDoor-t



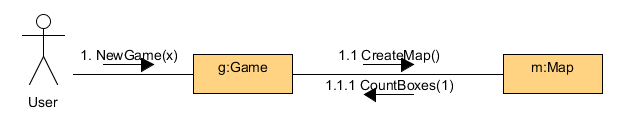
5.4.10 ábra: Player lép Button-ra.



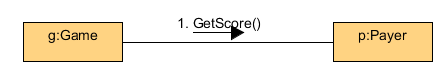
5.4.11 ábra: Valaki betol egy Box-ot a Target-be.



5.4.12 ábra: FieldAction függvény



5.4.13 ábra: Új játék



5.4.14: Update Score függvény

***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.03.08. 10:00 | 3 óra | Czetter  Frontó  Verasztó  Tomori | Megbeszélés.  Feladatok felosztása, dokumentum áttekintése |
| 2018.03.09. 10:00 | 3 óra | Czetter  Frontó  Verasztó  Tomori | Use-case és sequencia diagramok megalkotása és dokumentálása. |
| 2018.03.10. 18:00 | 2 óra | Czetter  Frontó  Verasztó  Tomori | Kommunikációs diagrammok készítése. |
| 2018.03.10. 15:00 | 5 óra | Czetter  Frontó  Verasztó  Tomori | Különböző részek összerakása, összehangolása és formázása. |

**06. Szkeleton beadás**

12 - ch33kybr33ky\_labz

Konzulens:

Bodó Zsófia

Csapattagok

|  |  |  |
| --- | --- | --- |
| **Frontó András** | **G392CQ** | andrasfronto@gmail.com |
| **Czetter Gábor** | **CP6OG3** | cz3tt1@gmail.com |
| **Tomori Dániel** | **TDWSF5** | tomori.daniel@gmail.com |
| **Verasztó Bálint** | **VSDRBZ** | balintveraszto@gmail.com |

1. **Szkeleton beadás**
   1. ***Fordítási és futtatási útmutató***
      1. **Fájllista**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fájl neve** | **Méret** | **Keletkezés ideje** | **Tartalom** |
| Box.java | 3 KB | 2018.03.18. | Box osztály |
| Button.java | 2 KB | 2018.03.18 | Button osztály |
| Direction.java | 1 KB | 2018.03.18 | Irányok |
| Field.java | 4 KB | 2018.03.18 | Field osztály |
| Game.java | 6 KB | 2018.03.18 | Game osztály és egyéb segéd függvények a Skeletonhoz |
| Hole.java | 1 KB | 2018.03.18 | Hole osztály |
| Map.java | 3 KB | 2018.03.18 | Map osztály |
| Moveable.java | 1 KB | 2018.03.18 | Moveable osztály |
| Player.java | 3 KB | 2018.03.18 | Player osztály |
| Target.java | 1 KB | 2018.03.18 | Target osztály |
| TrapDoor.java | 2 KB | 2018.03.18 | TrapDoor osztály |
| Wall.java | 1 KB | 2018.03.18 | Wall osztály |
| KillerSokoban.jar | 14.8 KB | 2018.03.18 | A jar-ba összecsomagolt program. |

* + 1. **Fordítás**

Fordítani **javac [tartalmazó mappa]/\*.java** paranccsal lehet ez az összes java kiterjesztésű fájlt lefordítja a mappában. Ezek után futtatható **java killer\_sokoban.Game** parancsal, de előre össze készítettünk egy .jar fájlt is, hogy elég legyen azt elindítani.

* + 1. **Futtatás**

Két kattintás a .jar fájlra és indul. Ha ez nem sikerülne akkor **java -jar KillerSokoban.jar** parancsal is futtatható.

A 3 menüpontból az 1. megmutatja hogyan készül egy pálya. A másodikban pedig a játékos lép. Ott rakható össze hogy hova lép. Illetve ha a dobozok és a játékosok interakcióját akarjuk tesztelni, hogy mit tol az is itt található. Az utolsó menüpont pedig kilép.

* 1. ***Értékelés***

|  |  |
| --- | --- |
| **Tag neve** | **Munka százalékban** |
| Czetter Gábor | 25% |
| Frontó András | 25% |
| Tomori Dániel | 25% |
| Verasztó Bálint | 25% |

* 1. **Módosítások és megállapítások**
     1. **Név módosítások**

Kérésre a CheckNeighbour() metódust Step()-re, a régi Step() metódust pedig Register()-re cseréltük. Így a nevük jobban tükrözi, miért felelősek maguk a függvények.

* + 1. **Játékmenetbeli módosítások**

Mostantól Player halálakor nem a Remove() hívja meg a Die() függvényt, mert a Remove() az áthelyezéskor is lefut, helyette a tényleges halálkor hívódik függetlenül.

Mostantól a Player a Control- ban vissza kérdez, hogy a Player aki a mozgást indította szomszédja e. Ha igen akkor Player tol Player-t interakció miatt elakadás lesz.

* + 1. **Megállapítások a tesztelésre nézve**

Mivel a teszt eset során, nincsen konkrét játék környezetünk, ahol tárolhatnánk a falakat és a meglévő dobozokat, hanem minden a futás során keletkezik és hajtódik végre, ezért nincs meg a megfelelő körülmény ahhoz, hogy a dobozok beszorulását és így az isStuck() függvényt teszteljük. (A megfelelő teszt eset ki alakítása feleslegesen macerássá tenné a tesztelést) Ezt sajnos későbbre kell halasztanunk, amikor a program eléri az adott fejlettségi szintet.

***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.03.15. 15:00 | 4 óra | Czetter  Frontó  Tomori  Verasztó | Feladatok kiosztása, skeleton elkezdése |
| 2018.03.17. 20:00 | 4 óra | Czetter  Frontó  Tomori  Verasztó | Skeleton folytatása, tesztesetek összeírása |
| 2018.03.18.12:00 | 5 óra | Czetter  Frontó  Tomori  Verasztó | Egyeztetés, tesztelés a felírt esetekre, dokumentáció |

1. **Prototípus koncepciója**
   1. ***Változtatások a megrendelő által***

Hosszas beszélgetések után kiderült, hogy a megrendelőnek ez a játék túlságosan egyszerű. Szívesen csavarna egyet a szabályokon, valamint a játékosoknak is nagyobb hatalmat adna, hogy ne csak a dobozok tologatásával tudják bosszantani egymást, hanem saját erejükből is. Így megszületett a döntés, hogy a játékosok csak bizonyos számú dobozt legyenek képesek eltolni, továbbá képesek legyenek a tolást megnehezíteni az ellenfeleik számára, és megkönnyíteni saját maguk számára. A megrendelő azt szeretné, hogy a játékosok képesek legyenek olajat vagy mézet helyezni a mezőre, amivel befolyásolják annak csúszósságát.

**Megbeszélt változtatások:**

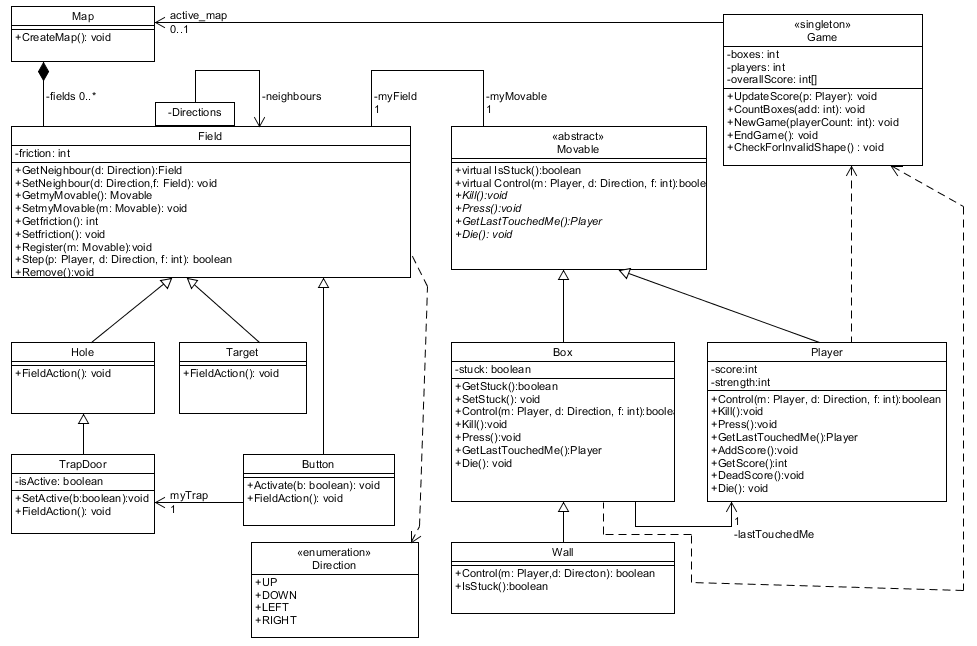
* A Player-ek fix erővel rendelkeznek ami azt jelképezi mennyi dobozt képesek eltolni egy vonalban.
* A Player-ek a mezőkre mézet és olajat tehetnek le amik a tolást könnyítik illetve nehezítik.
  1. ***Változás hatása a specifikációra***
     1. ***Player***

A Playerek egyszerre, csak adott számú Box-ot képesek eltolni. Ha esetleg a dobozokkal együtt egy másik Player tolása is bekövetkezne, akkor az a Player is nehezíti a tolást, tehát hozzáadódik az eltoltak számához. Ellenben, ha az eltolni kívánt Player szintén két doboz között van, akkor ezután ő próbálja meg eltolni az előtte lévő dobozokat. Ha a tolás sikeres előre lép egyet és az indító Player is szintén lép. Viszont ha az eltolni kívánt Player tolása meghiúsul, akár azért mert Wall-t szeretne tolni, akár azért mert elfogyott az ereje, a tolást indító Player összenyomja az eltolni kívánt Player-t.

* + 1. ***Honey és Oil***

A Player egy mezőre állva megtudja változtatni annak a tapadási súrlódását. Ezt a játék során egy Player bárhányszor megteheti. Ha kedve tartja és a többi játékos hagyják számára, akár az egész Map-et átváltoztathatja egy méztengerré vagy olajtócsává. Ezt úgy tudja elérnil, hogy egy Field-re lépve képes váltogatni az állapotok között. Sima Fieldről Honey Field-re képes váltani, Honey Field-ről Oil Field-re, és onnan pedig ismét vissza sima Field-re. Ha egy ilyen mezőn próbálunk Box-ot tolni azok mostantól befolyásolják a súrlódást. A sima Field 1 súrlódási értékkel rendelkezik, míg a Honey Field 2-essel, az Oil Field pedig 0-ással.

* 1. ***Nem deffiniált tényezőkre hozott döntések***
* A játékosok erejét a játék random osztja szét.
* A játékosok végtelenszer változtathatják a mező tapadási súrlódását.
* A Honey és az Oil a Player-ek lökését is megnehezíti illetve megkönnyíti.
  1. ***Változás hatása a modellre***
     1. **Módosult osztálydiagram**

**

* + 1. **Új vagy megváltozó metódusok**
       1. **Field**

int Getfriction(): Visszaadja a Field, pillanatnyi súrlódását.

void Setfriction(): Lépteti az adott Field súrlódását.

boolean Step(p: Player, d: Direction, f: int): A működése maradt továbbra is ugyan az, csak továbbadja, hogy az adott Player toló erejéből még mennyi maradt.

* + - 1. **Moveable**

boolean Control(m: Player, d: Direction, f: int): Ugyan azt csinálja mint eddig, csak ha a tovább adott i érték, ami a továbbadott erőt jelzi,ha elérte a nullát, élből visszadobja false-al.

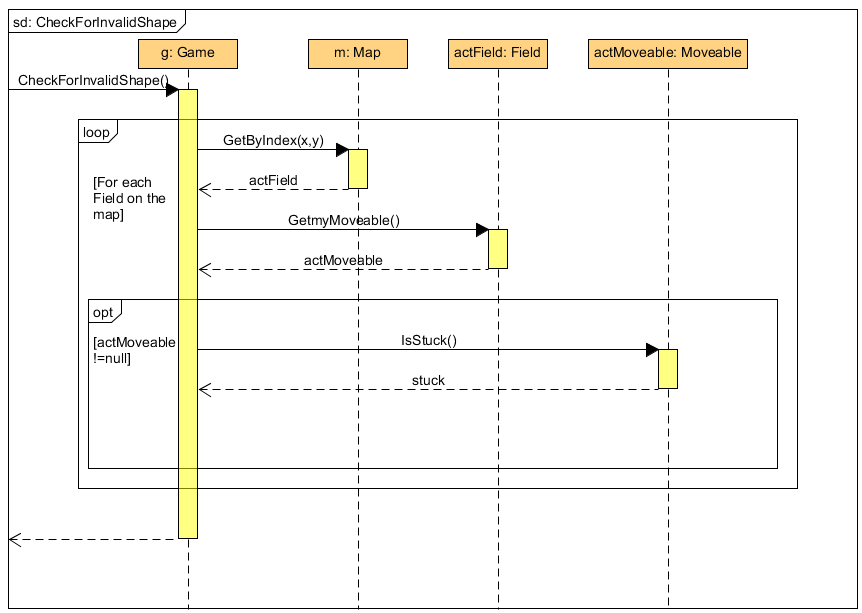
* + - 1. **Game**

void CheckForInvalidShape(): Ez a függvény átnézi a Map-et és eldönti hogy van e benne tolhatatlanná össze állt doboz sziget ha talál ilyet kivesz a játékból ennek megfelelő számú ládát.

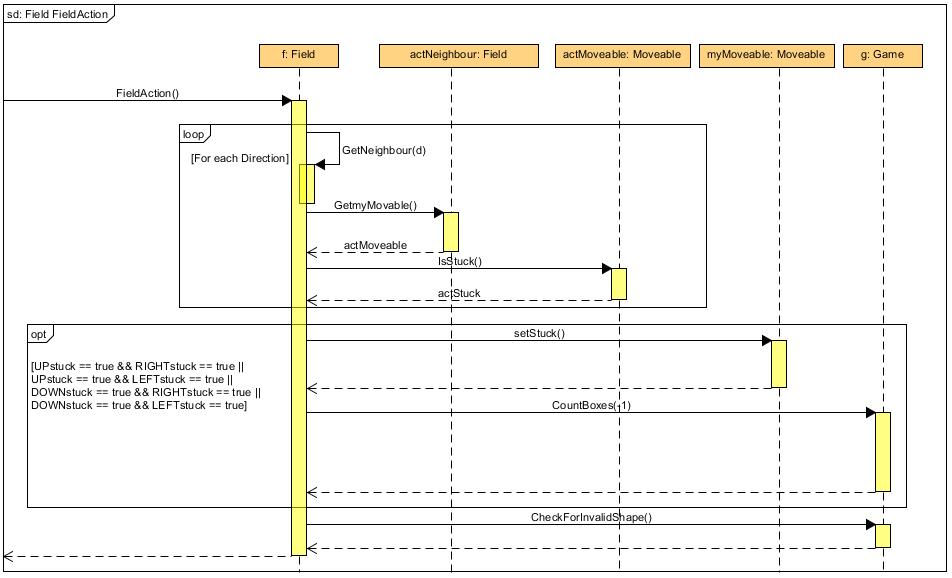
* + - 1. **Map**

Field GetByIndex(x: int,y: int): Azért kell hogy a rossz alakzatokat a játék megtalálja. Ehhez pedig a mezőket sorra vizsgálhassa.

* + 1. **Szekvencia-diagramok**
       1. **Check for Invalid shape**

****

* + - 1. **Field FieldAction**

****

A további diagrammok is változtak igazából az összes diagramm ami tartalmazta a Controll() vagy a Step() függvényeket viszont ezek az egy egy paraméterbeli változások nem indokolták, hogy ide bekerüljenek.

* 1. ***Prototípus interface-definíciója***
     1. **Az interfész általános leírása**

A program szabványos bemeneten és kimeneten kommunikál, parancs sorból kezelhető. Az általunk létrehozott scriptnyelv segítségével tesztelhető. Az egyes parancsok script fájlba fűzhetők és scriptként futtathatók. Vannak előre definiált teszt-esetek melyekhez előre elkészített scriptek tartoznak, szintén az általunk definiált parancsokból össze gyúrva.

* + 1. **Bemeneti nyelv egyszerű specifikációja**

A parancsokat egymás után, enter elválasztásával lehet beírni.

A “create Map” parancs, egy egyedülálla parancs. Egy kész Map-et generál, utána már csak a Player-ekkel kell lépegetni, a többi “create” és “connect” parancs nem használható.

Ha azt akarjuk, hogy amit létrehoztunk Player-t és Box-ot használni is lehessen mindenképp “connect” paranccsal kell őket egy adott Fieldhez kötni. Utána a “step” parancs kiadásával már szabadon irányítható az adott Player. Ugyan ez vonatkozik a Button-TrapDoor kapcsolatra. Az adott Button, csak az után nyitja ki az adott TrapDoor-t miután használtuk rajtuk a “connect” parancsot.

Figyelem! A Wall egy elmozdíthatatlan doboz nem Field. Ugyan úgy le kell helyezni egy Field-re (“connect” parancs) mint egy egyszerű Boxot.

* + 1. **Bemeneti nyelv**

***create Map***

**Leírás: Legenerálja a Map-ot.**

**Opciók: -**

***create Field “Field neve”***

**Leírás: Létrehoz egy sima Field-et**

**Opciók: A létrehozandó Field neve.**

***create Wall “Wall neve”***

**Leírás: Létrehoz egy Wall-t.**

**Opciók: A létrehozandó Wall neve.**

***create Hole “Hole neve”***

**Leírás: Létrehoz egy Hole-t.**

**Opciók: A létrehozandó Hole neve.**

***create Button “Button neve”***

**Leírás: Létrehoz egy Button-t.**

**Opciók: A létrehozandó Button neve.**

***create TrapDoor “TrapDoor neve”***

**Leírás: Létrehoz egy TrapDoor-t.**

**Opciók: A létrehozandó TrapDoor neve.**

***create Target “Target neve”***

**Leírás: Létrehoz egy Target-et.**

**Opciók: A létrehozandó Target neve.**

***create Player “Player neve”***

**Leírás: Létrehoz egy Player-t.**

**Opciók: A létrehozandó Player neve.**

***create Box “Box neve”***

**Leírás: Létrehoz egy Box-ot.**

**Opciók: A létrehozandó Box neve.**

***connect Neighbour “Field1 neve” “Field2 neve” “Direction”***

**Leírás: Field1 és a Field2 között oda-vissza szomszéd viszonyt állít.**

**Opciók: Az első és a második Field neve valamint az irány.**

***connect Moveable “Field neve” “Moveable neve”***

**Leírás: Ráhelyezi a megadott Moveable-t az adott Field-re.**

**Opciók: Field és Moveable neve.**

***connect Button-TrapDoor “Button neve” “TrapDoor neve”***

**Leírás: Logikailag összekapcsol egy Button-t és egy TrapDoor-t.**

**Opciók: Button és a TrapDoor neve.**

***choose “Player neve”***

**Leírás: Kiválasztja a Player-t és ezentúl őt tudjuk irányítani.**

**Opciók: Player neve.**

**--choose után--**

***step “Direction”***

**Leírás: Az adott irányba lépteti a Player-t.**

**Opciók: Direction.**

***listNeighbour***

**Leírás: Kilistázza a Field szomszédait.**

**Opciók: -**

***putSomeThing***

**Leírás: Megváltoztatja a Field súrlódását.**

**Opciók: -**

* + 1. **Kimeneti nyelv**

Kimenetet adó parancsok:

***step***

<sikerült-e a lépés>, <további lépés események>

***listNeighbour***

<szomszédos Field neve>, <szomszédes Field iránya>

* 1. ***Összes részletes use-case***

|  |  |
| --- | --- |
| **Use-case neve** | create Map |
| **Rövid leírás** | Legenerálja a Map-ot. |
| **Aktorok** | Game és Map osztályok |
| **Forgatókönyv** | Előre deffiniált algoritmus alapján random legenerálja a pályát, Player-ekkel, Box-okkal és minden Field típussal. |

|  |  |
| --- | --- |
| **Use-case neve** | create Field |
| **Rövid leírás** | Létrehoz egy sima Field-et. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Létrehoz a Tesztelő által egy Field-et, amivel speciális eseteket lehet tesztelni. |

|  |  |
| --- | --- |
| **Use-case neve** | create Wall |
| **Rövid leírás** | Létrehoz egy Wall-t. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Létrehoz a Tesztelő által egy Wall-t, amivel speciális eseteket lehet tesztelni. |

|  |  |
| --- | --- |
| **Use-case neve** | create Hole |
| **Rövid leírás** | Létrehoz egy Hole-t. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Létrehoz a Tesztelő által egy Hole-t, amivel speciális eseteket lehet tesztelni. |

|  |  |
| --- | --- |
| **Use-case neve** | create Button |
| **Rövid leírás** | Létrehoz egy Button-t. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Létrehoz a Tesztelő által egy Button-t, amivel speciális eseteket lehet tesztelni. |

|  |  |
| --- | --- |
| **Use-case neve** | create TrapDoor |
| **Rövid leírás** | Létrehoz egy TrapDoor-t. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Létrehoz a Tesztelő által egy TrapDoor-t, amivel speciális eseteket lehet tesztelni. |

|  |  |
| --- | --- |
| **Use-case neve** | create Target |
| **Rövid leírás** | Létrehoz egy Target-et. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Létrehoz a Tesztelő által egy Target-et, amivel speciális eseteket lehet tesztelni. |

|  |  |
| --- | --- |
| **Use-case neve** | create Player |
| **Rövid leírás** | Létrehoz egy Player-t. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Létrehoz a Tesztelő által egy Target-et, amivel speciális eseteket lehet tesztelni. |

|  |  |
| --- | --- |
| **Use-case neve** | create Box |
| **Rövid leírás** | Létrehoz egy Box-ot |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Létrehoz a Tesztelő által egy Box-ot, amivel speciális eseteket lehet tesztelni. |

|  |  |
| --- | --- |
| **Use-case neve** | connect Neighbour |
| **Rövid leírás** | Szomszédságot állít be. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Két megadott mező között, az adott irányba kölcsönös szomszédságot állít |

|  |  |
| --- | --- |
| **Use-case neve** | connect Moveable |
| **Rövid leírás** | Moveable-t helyez a Fieldre. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Egy megadott Moveable-t helyez egy adott Fieldre. |

|  |  |
| --- | --- |
| **Use-case neve** | connect Button-TrapDoor |
| **Rövid leírás** | Összekapcsol egy Button-t és egy TrapDoor-t |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Logikailag összekapcsol egy Button-t és egy TrapDoor-t. Ezentúl Button aktiválásra az a TrapdDoor nyílik. |

|  |  |
| --- | --- |
| **Use-case neve** | choose “Player neve” |
| **Rövid leírás** | Kiválaszt egy Player-t |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Kiválasztja az adott Player-t és ezentúl őt tudjuk irányítani. |

|  |  |
| --- | --- |
| **Use-case neve** | step “Direction” |
| **Rövid leírás** | Lépés |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Az adott irányba lép egyet az irányított Player-rel |

|  |  |
| --- | --- |
| **Use-case neve** | listNeighbour |
| **Rövid leírás** | Kilistázza a Field szomszédait. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Ha a Field-nek van beállított szomszédai, akkor kilistázza azokat. |

|  |  |
| --- | --- |
| **Use-case neve** | listBox |
| **Rövid leírás** | Kilistázza a játékban lévő Box-okat. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Kilistázza a még játékban lévő (tehát a nem leesett) Box-okat |

|  |  |
| --- | --- |
| **Use-case neve** | listPlayer |
| **Rövid leírás** | Kilistázza a játékban lévő Playereket. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Kilistázza az összes Player-t. Azokat is akik Hole-ba estek. |

|  |  |
| --- | --- |
| **Use-case neve** | putSomeThing |
| **Rövid leírás** | Megváltoztatja a Field súrlódását. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A Player indítja és folyamatosan változtatja a Field-je súrlódását. (Folyamatosan görget a sima, olajos vagy mézes verzió között.) |

* 1. ***Tesztelési terv***

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tol dobozt - sikerül |
| **Rövid leírás** | Alap mozgások tesztelése |
| **Teszt célja** | Teszteli, hogy a tolás sikeres lesz-e, olyan feltételek mellett, amelyek biztosítani igyekeznek a sikerességet. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tol dobozt - nem sikerül |
| **Rövid leírás** | Alap mozgások tesztelése |
| **Teszt célja** | Teszteli, hogy a tolás sikeres lesz-e, olyan feltételek mellett, amelyek biztosítani igyekeznek a bukást. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tolni próbál embert - nem sikerül |
| **Rövid leírás** | Alap mozgások tesztelése |
| **Teszt célja** | Teszteli, hogy a tolás sikeres lesz-e, a korábban megfogalmazott döntéseink alapján, nem lesz sikeres. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tol dobozt emberre - ember eltolódik |
| **Rövid leírás** | Alap mozgások tesztelése |
| **Teszt célja** | Amennyiben nincs az ember mögött akadály, vagy bármi ami akadályozni tudja az eltolás, az ember nem hal meg, hanem arréb lép. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tol dobozt emberre - ember meghal |
| **Rövid leírás** | Alap mozgások / halál tesztelése |
| **Teszt célja** | Amennyiben a léptetés nem lehetséges, a hátrébb lévő munkás meghal. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tol dobozt a célba |
| **Rövid leírás** | Alap mozgások / cél mező  tesztelése |
| **Teszt célja** | Amennyiben egy doboz bekerül a célba annak az embernek pont jár aki utoljára hozzáért. Teszteljük a cél működését valamint a pontozást is. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember célra lép |
| **Rövid leírás** | Alap mozgások / cél mező  tesztelése |
| **Teszt célja** | Cél mező tesztelése, ember hatására. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember lyukra lép |
| **Rövid leírás** | Alap mozgások / lyuk mező  tesztelése |
| **Teszt célja** | lyuk mező tesztelése, ember hatására. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember gombra lép |
| **Rövid leírás** | Alap mozgások / gombra mező  tesztelése |
| **Teszt célja** | gombra mező tesztelése, ember hatására. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember inaktív csapdára lép |
| **Rövid leírás** | Alap mozgások / csapda / gomb mező  tesztelése |
| **Teszt célja** | csapda / gombra mező tesztelése, ember hatására. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tol dobozt lyukba |
| **Rövid leírás** | Alap mozgások / lyuk  mező tesztelése |
| **Teszt célja** | Amennyiben doboz lyukra tolódik, megsemmisül. A lyuk működését és a doboz eltüntetését teszteljük. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tol dobozt csapdára |
| **Rövid leírás** | Alap mozgások / csapda  mező tesztelése |
| **Teszt célja** | Amennyiben doboz csapda mezőre tolódik és a csapda aktív, megsemmisül. A csapda működését és a doboz eltüntetését teszteljük. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember - doboz - ember - doboz sorozat utolsó elemének célba tolása |
| **Rövid leírás** | Alap mozgások / pontozás /cél mező tesztelése. |
| **Teszt célja** | Ebben a tesztesetben a dokumentáltak szerinz az kapja a pontot aki utoljára ért a dobozhoz, akkor is ha kényszerítve volt a mozgásra. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tol dobozt kapcsolóra. |
| **Rövid leírás** | Alap mozgások / gomb mező  tesztelése |
| **Teszt célja** | Amennyiben doboz gomb mezőre tolódik, a gombhoz tartozó csapda aktív lesz. A gomb működését és a csapda aktiválását teszteljük. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tol dobozt kapcsolóra, a kapcsolóhoz tartozó csapdán ember van. |
| **Rövid leírás** | Alap mozgások / gomb mező / csapda / ember  tesztelése |
| **Teszt célja** | Egy összetettebb teszteset. Amennyiben doboz gomb mezőre tolódik, a gombhoz tartozó csapda aktív lesz. A csapda működéséből adódóan az ember meghal. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tol dobozt kapcsolóra, a kapcsolóhoz tartozó csapdán doboz van. |
| **Rövid leírás** | Alap mozgások / gomb mező / csapda / doboz  tesztelése |
| **Teszt célja** | Egy összetettebb teszteset. Amennyiben doboz gomb mezőre tolódik, a gombhoz tartozó csapda aktív lesz. A csapda működéséből adódóan a doboz eltűnik. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tol dobozt ami beltolja a mellette lévő embert egy lyukba |
| **Rövid leírás** | Alap mozgások / lyuk / ember  tesztelése |
| **Teszt célja** | Az ember lyuk általi megölését teszteljük. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Ember tol dobozt ami beltolja a mellette lévő dobozt egy lyukba |
| **Rövid leírás** | Alap mozgások / lyuk / doboz  tesztelése |
| **Teszt célja** | Az doboz lyuk általi megsemmisítését teszteljük. |

* 1. ***Tesztelést támogató segéd- és fordítóprogramok specifikálása***

Az általunk definiált nyelv segítségével, a program tesztelése meglepően egyszerűvé és átláthatóvá válik. Képesek leszünk előre definiált teszt-eseteket lefuttatni, amik végeredményül az alapértelmezett kimenetre írják ki, hogy a teszt során milyen változások mentek végbe. Továbbá minden eszköz adott ahhoz, hogy szabadjára engedjük a képzeletünket és saját magunknak hozzunk létre logikailag nyakatekertebb teszteket. A program ebben az esetben is az alapértelmezett kimenetre írja ki a teszt-eset során meghívott metódusokat, így ellenőrízhetővé téve a folyamatot. Illetve a program egyesével kiadott parancsokkal is kezelhető melyeket szintén a  segéd program értelmez majd.

***7.6 Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.03.23. 12:00 | 3 óra | Czetter  Frontó  Tomori  Verasztó | A specifikáció módosításainak kiértékelése. Bemeneti nyelv tervezése. |
| 2018.03.24. 15:00 | 4 óra | Czetter  Frontó  Tomori  Verasztó | Változtatások átbeszélése, és szequencia, osztálydiagrammok módosítása. Bemeneti nyelv megalkotása. |
| 2018.03.25. 19:00 | 3 óra | Czetter  Frontó  Tomori  Verasztó | Use-case és teszt-esetek összeírása. Dokumentum átnézése, javítása formázása. |

**08. Részletes tervek**

12 - ch33kybr33ky\_labz

Konzulens:

Bodó Zsófia

Csapattagok

|  |  |  |
| --- | --- | --- |
| **Frontó András** | **G392CQ** | andrasfronto@gmail.com |
| **Czetter Gábor** | **CP6OG3** | cz3tt1@gmail.com |
| **Tomori Dániel** | **TDWSF5** | tomori.daniel@gmail.com |
| **Verasztó Bálint** | **VSDRBZ** | balintveraszto@gmail.com |

1. **Részletes tervek**
   1. ***Osztályok és metódusok tervei.***
      1. **Box**
         * **Felelősség**

Tárolja a Field-et amin jelenleg tartózkodik, továbbá felelős azért, hogy éppen be van-e szorulva.

* + - * **Ősosztályok**

Moveable → Box

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **-boolean stuck:** Azért felelős, hogy a Box be van-e szorulva vagy sem.
  + - * **Metódusok**
* **boolean GetStuck():** Visszaadja, hogy be van-e szorulva a Box.
* **void SetStuck():** Beállítja a Box szorulását.
* **boolean Control(m: Player, d: Direction, f: int):** Elindít egy lépési folyamatot a megadott Player-től, a megadott irányba, a megadott erővel.
* **boolean Kill():** A doboznál nem történik semmi és false-val jön vissza mivel a doboz nem össze nyomható.
* **void Press():** Button nyomáskor hívódik meg. Aktiválja a Button-t ami a hozzá tartozó TrapDoor-t.
* **Player GetLastTouchedMe():** Visszaadja azt a Player-t aki legutoljára beletolt.
* **void Die():** Meghívódik, ha a Box elpusztul. Target-be ér vagy Hole-ba zuhan.
  + 1. **Button**
       - **Felelősség**

Felelős minden Field feladatért, továbbá a hozzá beállított TrapDoor nyitás és csukásáért Box hatására.

* + - * **Ősosztályok**

Field → Button

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **-myTrap**: A hozzá tartozó TrapDoor, amit ki- illetve bezár.
  + - * **Metódusok**
* **void Activate(b: boolean):** Activálja a Button-hoz rendelt TrapDoor-t.
* **void FieldAction():**  Akkor fut le mikor rá lépnek a gombra. Megindítja a megnyomódási folyamatot illetve a sima mezőhöz hasonlóan ellenőrzi a beszorult dobozokat.
  + 1. **Directions**

Az irányokat tartalmazó enumeráció. (UP, DOWN, RIGHT, LEFT).

* + 1. **Field**
       - **Felelősség**

Felelős a rajta lévő Moveableért, továbbá minden szomszédját ismeri és számontartja őket a megadott irányba. Rendelkezik saját súrlódással amit egy rajta álló Player tud kedve szerint változtatni. Ő hajtja végre a kezdeményezést és a léptetést a Moveable-ök között.

* + - * **Ősosztályok**

-

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **-Field[Directions] neighbours**: Tárolja a szomszédokat minden irányban.
* **-Moveable myMoveable:** A Field lévő Moveable-t tárolja.
* **-int friction:** A Field pillanatnyi súrlódását tárolja amit a rajta álló Player tud változtatni.
  + - * **Metódusok**
* **Field GetNeighbour(d: Direction):** A megadott irányba lévő Field-et adja vissza.
* **void SetNeighbour(d: Direction, f: Field):** Beállítja az adott irányba a paraméterül kapott Field-et.
* **Moveable GetmyMovable():** Visszaadja a rajta lévő Moveable objektumot. Ha nincs rajta, a visszatérési értéke null.
* **int Getfriction():** Visszaadja az adott Field súrlódását.
* **void Setfriction():** Növeli az adott Field súrlódását, maximum érték esetén visszaáll nullára.
* **void Register(m: Movable):** Beállít az adott Field-re egy egy megadott Moveable objektumot.
* **boolean Step(p: Player, d: Direction, f: int):**  Tovább lépési szándékot jelez a megadott Player-től az adott irányba. Ha a visszatérési értéke igaz, akkor szabadon léphet tovább, egyébként a lépés meghiúsult hibás tolás, vagy a kevés erő miatt.
* **void Remove():** Az épp rajta álló Moveablet eltünteti.
* **void virtual FieldAction():** Minden Field-hez az egyedi tulajdonságot hajtja végre ami kötelezően meghívódik, ha valaki rálép vagy rátolódik. Figyeli, hogy beszorul e a láda.
  + 1. **Game**
       - **Felelősség**

Egy singleton osztály. Feladata a játék elkezdése és lezárása, továbbá ezeknek a feltételeknek az ellenőrzése. Folyamatosan számon tartja hány Box mozgatható még, így ha ez a szám nullára csökken a játék autómatikusan véget ér, valamint a Player-ek pontjait és számát is ő tartja nyilván.

* + - * **Ősosztályok**

-

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **- int boxes**
* **- int players**
* **- int[] overallScore**
* **- Map active\_map**
  + - * **Metódusok**
* **+ void UpdateScore(Player p):** Frissíti az adott Player pontszámát. Vagy növeli, vagy nulláza abban az esetben ha meghalt. Ha kiesik 3 játékos befejezi a játékot.
* **+ void CountBoxes(int add):** Számlálja a mozgatható Box-okat a Map-en. Amikor elfogynak befejezi a játékot.
* **+ void NewGame(int playerCount):** Új játékot indít a megadott számú játékossal.
* **+ void EndGame():** Véget ér a játék, ha vagy mozgatható Box, vagy 1 nél több irányítható Player nincs már a Map-en.
* **+ void CheckForInvalidShape():** Át nézi az egész térképet és olyan alakzat után kutat amiből a legerősebb játékos se tud kitolni dobozt ha talál ilyet akkor az alakzat összes dobozát beszorultként könyveli el.
  + 1. **Hole**
       - **Felelősség**

A Fieldből öröklődik, ha bármilyen Moveable rákerül, az törlődik. A Player elveszíti az összes pontjait a Box pedig meghal, anélkül hogy bárkinek is pontot adna.

* + - * **Ősosztályok**

Field → Hole

* + - * **Interfészek**

-

* + - * **Attribútumok**

**-**

* + - * **Metódusok**
* **+ void FieldAction():** Ebben az esetben, bármi ami rákerül, leesik, tehát törlődik a játékból.
  + 1. **Map**
       - **Felelősség**

A pálya szabályos kialakításáért felelős, ezzel együtt raktározza a Field-eket.

* + - * **Ősosztályok**

*-*

* + - * **Interfészek**

*-*

* + - * **Attribútumok**
* **-Field[] fields**
  + - * **Metódusok**
* **+ void CreateMap()**: Megadott algoritmusok szerint alkottja meg random a pályát az egyszerű Field-ek és a speciális Field-ek használatával. Lerakja a Box-okat, Wall-okat és a Player-eket is.
  + 1. **Moveable**
       - **Felelősség**

A Moveable egy olyan abstract osztály melyből a Box és a Player is egyaránt származik. Egy Field csak egy Moveable-t taralmazhat és a Moveable is eltárolja a Field-et amin épp elhelyezkedik.

* + - * **Ősosztályok**

-

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **-Field myField**
  + - * **Metódusok**

(Az abstract függvényeket nem vettük fel a jobb átláthatóság miatt)

* **+ virtual boolean Control(m: Player, d: Direction, f: int):** Mozdulást intéz egy Field felé, és visszaadja, hogy sikerült-e a mozdulás. Első paramétere megmondja, hogy ki tolta, a második pedig, hogy milyen irányba, a harmadik pedig hogy milyen erővel.
* **+ virtual boolean IsStuck():** Alapvetően false-val tér vissza.
  + 1. **Player**
       - **Felelősség**

A játékos “bábujának” megvalósítása. Tárolja az adott játékos pontjait, illetve az erejét - azaz hogy hány egymás utáni dobozt tud eltolni. Megvalósítja a “lépést” a térképen a játékos(ok) számára. Moveable-ből származtatott osztály.

* + - * **Ősosztályok**

Moveable→ Player

* + - * **Attribútumok**
* **- int score:** tárolja, hogy hány pontja van az adott játékosnak.
* **- int strength:** tárolja, hogy hány egymás utáni dobozt tud eltolni az adott játékos sima felületen.
  + - * **Metódusok**
* **+ boolean Control(m: Player, d: Direction, f: int)**: Egy lépést indít *m* játékostól *d* irányba, *f*  “erővel”. Visszatérési értéke megadja, hogy az adott lépés sikeres volt-e az átadott paraméterek mellett.
* **+ boolean Kill()**: Meghívja a saját Fieldjének Remove() metódusát, ezzel “megölve” magát hiszen a játékos összenyomható.
* **+ void Press():** Ez a függvény hívja meg az adott Button Activate() függvényét - false paraméterrel, hiszen egy játékos nem tud megnyomni egy gombot.
* **+ Player GetLastTouchedMe():** Önmagát adja vissza.
* **+ void AddScore():** Egyel növeli az adott Player score változóját, majd a Game UpdateScore() függvényét meghívja.
* **+ int GetScore():** Vissza tér a Player score változójának értékével.
* **+ void DeadScore():** -1 -re állítja a Player pontszámát, majd a Game UpdateScore() függvénéyt meghívva kiveteti a játékból a Playert.
* **+ void Die():** Meghívja a saját DeadScore() függvényét ezzel elindítva a meghalásakor lejátszandó események sorozatát.
  + 1. **Target**
       - **Felelősség**

Felelőssége, hogy csak a Box-okra aktiválódjon, a Playereket figyelmen kívül hagyja, továbbá, hogy az utolsó lökést adó Player-nek pont jóváírást kezdeményezzen.

* + - * **Ősosztályok**

Field → Target

* + - * **Interfészek**

-

* + - * **Attribútumok**
      * **Metódusok**
* **+ void FieldAction():** Rátoláskor a Box eltűnik és az utolsó Playernek aki hozzá ért pont írást kezdeményez.
  + 1. **TrapDoor**
       - **Felelősség**

Felelős a nyitásért illetve csukódásért. Ha zárva van egyszerű Field-ként viselkedik. Ha nyitva, akkor pedig Hole-ként.

* + - * **Ősosztályok**

Field → Hole → TrapDoor

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **-boolean isActive**
  + - * **Metódusok**
* **+ void SetActive(boolean: b):** Átállítja a TrapDoor állását. Nyitottból csukott, csukottból nyitott.
* **+ void FieldAction():** Aktiválódáskor kinyílik, és ha éppen egy Player vagy egy Box volt rajta, akkor azt törli. Ha a hozzá tartozó Button-ról letolják a Box-ot akkor bezárul és sima Field lesz amire midnen következmény nélkül rá lehet lépni vagy tolni.
  + 1. **Wall**
       - **Felelősség**

Egy elmozdíthatatlan Box. Tulajdonképpen mindig stuckolva van. Felelős azért, hogy ne lehessen se eltolni, se átmenni rajta.

* + - * **Ősosztályok**

Moveable → Box → Wall

* + - * **Interfészek**

-

* + - * **Attribútumok**
* **-boolean stuck**
  + - * **Metódusok**
* **+ boolean Control(Player m, Direction d):** Mindig biztosan false-val tér vissza hiszen nem lehet rajta át menni és eltolhatatlan is.
* **+ boolean IsStuck():** Mindig biztosan true-val tér vissza hiszen folyton mozdíthatatlan.
  1. ***A tesztek részletes tervei, leírásuk a teszt nyelvén***
     1. **Ember tol dobozt - sikerül**
* **Leírás**

Teszteli, hogy a tolás sikeres lesz-e, olyan feltételek mellett, amelyek biztosítani igyekeznek a sikerességet.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a doboz és az ember áthelyeződik-e ezáltal a tolás sikeres lesz-e.

* **Bemenet**

create Field F1;

create Field F2;

create Field F3;

create Player P1;

create Box B1;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “F3” “RIGHT”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “B1”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push box(success) test: Successfull.

Field F1 has been created.

Field F2 has been created.

Field F3 has been created.

Player P1 has been created.

Box B1 has been created.

F2 connected to the Right side of F1.

F3 connected to the Right side of F2.

P1 has been moved to F1.

B1 has been moved to F2.

P1 has been selected.

Moving P1 Right

Moving B1 Right

B1 has moved to F3.

P1 has moved to F2.

* + 1. **Ember tol dobozt - nem sikerül**
* **Leírás**

Teszteli, hogy a tolás sikeres lesz-e, olyan feltételek mellett, amelyek biztosítani igyekeznek a bukást.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a doboz és az ember a helyén marad vagy sem. Amennyiben igen a teszt sikeres.

* **Bemenet**

create Field F1;

create Field F2;

create Field F3;

create Player P1;

create Box B1;

create Wall W1;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “F3” “RIGHT”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “B1”;

connect Moveable “F3” “W1”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push box(fail) test: Successful.

Field F1 has been created.

Field F2 has been created.

Field F3 has been created.

Player P1 has been created.

Box B1 has been created.

Wall W1 has been created.

F2 connected to the Right side of F1.

F3 connected to the Right side of F2.

P1 has been moved to F1.

B1 has been moved to F2.

W1 has been moved to F3.

P1 has been selected.

Moving P1 Right

Moving B1 Right

B1 has not been moved to F3.

P1 has not been moved to F2.

* + 1. **Ember tolni próbál embert - nem sikerül**
* **Leírás**

Teszteli, hogy a tolás sikeres lesz-e, a korábban megfogalmazott döntéseink alapján, nem lesz sikeres.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a emberek a helyükön maradtak-e és hogy egyik sem semmisült meg.

* **Bemenet**

create Field F1;

create Field F2;

create Field F3;

create Player P1;

create Player P2;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “F3” “RIGHT”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “P2”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push player test: Successful.

Field F1 has been created.

Field F2 has been created.

Field F3 has been created.

Player P1 has been created.

Player P2 has been created.

F2 connected to the Right side of F1.

F3 connected to the Right side of F2.

P1 has been moved to F1.

P2 has been moved to F2.

P1 has been selected.

Moving P1 Right

P1 has not been moved to F2.

* + 1. **Ember tol dobozt emberre - ember eltolódik**
* **Leírás**

Amennyiben nincs az ember mögött akadály, vagy bármi ami akadályozni tudja az eltolás, az ember nem hal meg, hanem arréb lép.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy az ember eltolódik és nem semmisül meg.

* **Bemenet**

create Field F1;

create Field F2;

create Field F3;

create Field F4;

create Player P1;

create Box B1;

create Player P2;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “F3” “RIGHT”;

connect Neighbour “F3” “F4” “RIGHT”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “B1”;

connect Moveable “F3” “P2”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push box to another player(player survive) test: Successful.

Field F1 has been created.

Field F2 has been created.

Field F3 has been created.

Field F4 has been created.

Player P1 has been created.

Box B1 has been created.

Player P2 has been created.

F2 connected to the Right side of F1.

F3 connected to the Right side of F2.

F4 connected to the Right side of F3.

P1 has been moved to F1.

B1 has been moved to F2.

P2 has been moved to F3.

P1 has been selected.

Moving P1 Right

Moving B1 Right

Moving P2 Right

P2 has been moved to F4.

B1 has been moved to F3.

P1 has been moved to F2.

* + 1. **Ember tol dobozt emberre - ember meghal**
* **Leírás**

Amennyiben a léptetés nem lehetséges, a hátrébb lévő munkás meghal.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrízzük, hogy a toló ember és a doboz eltolódik a másik ember pedig meghal.

* **Bemenet**

create Field F1;

create Field F2;

create Field F3;

create Field F4;

create Player P1;

create Box B1;

create Player P2;

create Wall W1;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “F3” “RIGHT”;

connect Neighbour “F3” “F4” “RIGHT”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “B1”;

connect Moveable “F3” “P2”;

connect Moveable “F4” “W1”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push box to another player(player die) test: Successful.

Field F1 has been created.

Field F2 has been created.

Field F3 has been created.

Field F4 has been created.

Player P1 has been created.

Box B1 has been created.

Player P2 has been created.

Wall W1 has been created.

F2 connected to the Right side of F1.

F3 connected to the Right side of F2.

F4 connected to the Right side of F3.

P1 has been moved to F1.

B1 has been moved to F2.

P2 has been moved to F3.

W1 has been moved to F4.

P1 has been selected.

Moving P1 Right

Moving B1 Right

Moving P2 Right

P2 has not been moved to F4.

P2 died.

P2’s score has been updated.

B1 has been moved to F3.

P1 has been moved to F2.

* + 1. **Ember tol dobozt a célba**
* **Leírás**

Amennyiben egy doboz bekerül a célba annak az embernek pont jár aki utoljára hozzáért. Teszteljük a cél működését valamint a pontozást is.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a doboz eltolódik-e valamint hogy a toló ember megkapja-e a pontot és hogy a doboz eltűnik-e.

* **Bemenet**

create Field F1;

create Field F2;

create Target T1;

create Player P1;

create Box B1;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “T1” “RIGHT”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “B1”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push box to target test: Successful.

Field F1 has been created.

Field F2 has been created.

Target T1 has been created.

Player P1 has been created.

Box B1 has been created.

F2 connected to the Right side of F1.

T1 connected to the Right side of F2.

P1 has been moved to F1.

B1 has been moved to F2.

P1 has been selected.

Moving P1 Right

Moving B1 Right

B1 has been moved to T1.

B1 has been destroyed.

P1’s score has been updated.

P1 has been moved to F2.

* + 1. **Ember célra lép**
* **Leírás**

Cél mező tesztelése, ember hatására. Amennyiben ember lép a célra, a célnak azt figyelmen kívül kell hagyni, ellenben a dobozzal.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy ha ember lép a célra nem történik sem pontbeli változás sem megsemmisítés.

* **Bemenet**

create Field F1;

create Target T1;

create Player P1;

connect Neighbour “F1” “T1” “RIGHT”;

connect Moveable “F1” “P1”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player step to target test: Successful

Field F1 has been created.

Target T1 has been created.

Player P1 has been created.

T1 connected to the Right side of F1.

P1 has been moved to F1.

P1 has been selected.

Moving P1 Right

P1 has been moved to T2.

* + 1. **Ember lyukra lép**
* **Leírás**

Lyuk mező tesztelése, ember hatására. A lyuk mező megöli az embert.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy lyukra lépéskor megsemmisül az ember és a ponjait töröljük.

* **Bemenet**

create Field F1;

create Hole H1;

create Player P1;

connect Neighbour “F1” “H1” “RIGHT”;

connect Moveable “F1” “P1”;

choose “P1”;

step “RIGHT”; csocsi

* **Elvárt kimenet**

Player step to hole test: Successful

Field F1 has been created.

Hole H1 has been created.

Player P1 has been created.

H1 connected to the Right side of F1.

P1 has been moved to F1.

P1 has been selected.

Moving P1 Right

P1 has been moved to H1.

P1 died.

P1’s score has been updated.

* + 1. **Ember gombra lép**
* **Leírás**

Gomb mező tesztelése, ember hatására. A gomb menő figyelmen kívül hagyja az embert, csak doboz hatására aktiválódik.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a gombra lépéskor nem történik érdemi változás.

* **Bemenet**

create Field F1;

create Button B1;

create TrapDoor TD1;

create Player P1;

connect Neighbour “F1” “B1” “RIGHT”;

connect Moveable “F1” “P1”;

connect Button-TrapDoor “B1” “TD1”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player step to button test: Successful

Field F1 has been created.

Button B1 has been created.

TrapDoor TD1 has been created.

Player P1 has been created.

B1 connected to the Right side of F1.

P1 has been moved to F1.

B1 button connected to TD1 trapdoor.

P1 has been selected.

Moving P1 Right

P1 has been moved to B1.

TD1 is not activated.

* + 1. **Ember inaktív csapdára lép**
* **Leírás**

Csapda / gomb mező tesztelése, ember hatására. Inaktív csapda esetén a csapda mező nem csinál semmit a rajta álló emberel vagy dobozzal.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük a lépés sikerességét, továbbá, hogy a Player a TrapDooron nem esik le, hiszen a hozzá rendelt Button alapjáraton nulla.

* **Bemenet**

create Field F1;

create Button B1;

create TrapDoor TD1;

create Player P1;

connect Neighbour “F1” “TD1” “RIGHT”;

connect Moveable “F1” “P1”;

connect Button-TrapDoor “B1” “TD1”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player step to trapdoor(inactive) test: Successful

Field F1 has been created.

Button B1 has been created.

TrapDoor TD1 has been created.

Player P1 has been created.

TD1 connected to the Right side of F1.

P1 has been moved to F1.

B1 button connected to TD1 trapdoor.

P1 has been selected.

Moving P1 Right

P1 has been moved to TD1.

* + 1. **Ember tol dobozt lyukba**
* **Leírás**

Amennyiben doboz lyukra tolódik, megsemmisül. A lyuk működését és a doboz eltüntetését teszteljük.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy az egységek eltolódtak, továbbá, hogy a Box a Hole-on eltűnik és a Player nem kap érte pontot.

* **Bemenet**

create Field F1;

create Field F2;

create Hole H1;

create Player P1;

create Box B1;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “H1” “RIGHT”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “B1”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push box to hole test: Successful

Field F1 has been created.

Field F2 has been created.

Hole H1 has been created.

Player P1 has been created.

Box B1 has been created.

F2 connected to the Right side of F1.

H1 connected to the Right side of F2.

P1 has been moved to F1.

B1 has been moved to F2.

P1 has been selected.

Moving P1 Right

Moving B1 Right

B1 has been moved to H1.

B1 has been destroyed.

P1 has been moved to F2.

* + 1. **Ember tol dobozt csapdára**
* **Leírás**

Amennyiben doboz csapda mezőre tolódik és a csapda aktív, megsemmisül. A csapda működését és a doboz eltüntetését teszteljük.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy az egységek eltolódtak, továbbá, hogy a Box a TrapDoor-on nem tűnik el, mivel a hozzárendelt Button alapjáraton inaktív.

* **Bemenet**

create Field F1;

create Field F2;

create TrapDoor TD1;

create Button B1;

create Player P1;

create Box Bx1;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “TD1” “RIGHT”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “Bx1”;

connect Button-TrapDoor “B1” “TD1”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push box to trapdoor test: Successful

Field F1 has been created.

Field F2 has been created.

TrapDoor TD1 has been created.

Button B1 has been created.

Player P1 has been created.

Box Bx1 has been created.

F2 connected to the Right side of F1.

TD1 connected to the Right side of F2.

P1 has been moved to F1.

Bx1 has been moved to F2.

B1 button connected to TD1 trapdoor.

P1 has been selected.

Moving P1 Right

Moving B1 Right

B1 has been moved to TD1.

P1 has been moved to F2.

* + 1. **Ember - doboz - ember - doboz sorozat utolsó elemének célba tolása**
* **Leírás**

Ebben a tesztesetben a dokumentáltak szerint az kapja a pontot aki utoljára ért a dobozhoz, akkor is ha kényszerítve volt a mozgásra.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy minden egység eltolódott rendesen, továbbá az utolsó Box eltűnik-e a Targetben, valamint az utolsó lökést adó Player kapja-e a pontot a folyamat végén.

* **Bemenet**

create Field F1;

create Field F2;

create Field F3;

create Field F4;

create Target T1;

create Player P1;

create Box B1;

create Player P2;

create Box B2;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “F3” “RIGHT”;

connect Neighbour “F3” “F4” “RIGHT”;

connect Neighbour “F4” “T1” “RIGHT”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “B1”;

connect Moveable “F3” “P2”;

connect Moveable “F4” “B2”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push box, box push player, player push box to target test: Successful.

Field F1 has been created.

Field F2 has been created.

Field F3 has been created.

Field F4 has been created.

Target T1 has been created.

Player P1 has been created.

Box B1 has been created.

Player P2 has been created.

Box B2 has been created.

F2 connected to the Right side of F1.

F3 connected to the Right side of F2.

F4 connected to the Right side of F3.

T1 connected to the Right side of F4.

P1 has been moved to F1.

B1 has been moved to F2.

P2 has been moved to F3.

B2 has been moved to F4.

P1 has been selected.

Moving P1 Right

Moving B1 Right

Moving P2 Right

Moving B2 Right

B2 has been moved to T1

B2 has been destroyed.

P2’s score has been updated.

P2 has been moved to F4.

B1 has been moved to F3.

P1 has been moved to F2.

* + 1. **Ember tol dobozt kapcsolóra.**
* **Leírás**

Amennyiben doboz gomb mezőre tolódik, a gombhoz tartozó csapda aktív lesz. A gomb működését és a csapda aktiválását teszteljük.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a tolás megtörtént és a Box aktiválta a Button-hoz csatolt TrapDoor-t.

* **Bemenet**

create Field F1;

create Field F2;

create Button B1;

create TrapDoor TD1;

create Player P1;

create Box Bx1;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “B1” “RIGHT”;

connect Button-TrapDoor “B1” “TD1”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “Bx1”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push box to button(trapdoor open) test: Successful

Field F1 has been created.

Field F2 has been created.

Button B1 has been created.

TrapdDoor TD1 has been created.

Player P1 has been created.

Box Bx1 has been created.

F2 connected to the Right side of F1.

B1 connected to the Right side of F2.

B1 button connected to TD1 trapdoor.

P1 has been moved to F1.

Bx1 has been moved to F2.

P1 has been selected.

Moving P1 Right

Moving B1 Right

Bx1 has been moved to B1.

B1 actived.

TD1 is set to active.

P1 has been moved to F2.

* + 1. **Ember tol dobozt kapcsolóra, a kapcsolóhoz tartozó csapdán ember van.**
* **Leírás**

Egy összetettebb teszteset. Amennyiben doboz gomb mezőre tolódik, a gombhoz tartozó csapda aktív lesz. A csapda működéséből adódóan az ember meghal.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy sikerült-e a tolás, továbbá a Button benyomódott-e és a Button-hoz csatolt TrapDoor kinyílt és ami rajta volt az beleesett aztán meghalt.

* **Bemenet**

create Field F1;

create Field F2;

create Button B1;

create TrapDoor TD1;

create Player P1;

create Player P2;

create Box Bx1;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “B1” “RIGHT”;

connect Neighbour “B1” “TD1” “RIGHT”;

connect Button-TrapDoor “B1” “TD1”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “Bx1”;

connect Moveable “TD1” “P2”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push box to button(trapdoor open and kill a player) test: Successful

Field F1 has been created.

Field F2 has been created.

Button B1 has been created.

TrapdDoor TD1 has been created.

Player P1 has been created.

Player P2 has been created.

Box Bx1 has been created.

F2 connected to the Right side of F1.

B1 connected to the Right side of F2.

TD1 connected to the Right side of B1.

B1 button connected to TD1 trapdoor.

P1 has been moved to F1.

Bx1 has been moved to F2.

P2 has been moved to TP1.

P1 has been selected.

Moving P1 Right

Moving B1 Right

Bx1 has been moved to B1.

TD1 is set to active.

P2 died.

P2’s score has been updated.

P1 has been moved to F2.

* + 1. **Ember tol dobozt kapcsolóra, a kapcsolóhoz tartozó csapdán doboz van.**
* **Leírás**

Egy összetettebb teszteset. Amennyiben doboz gomb mezőre tolódik, a gombhoz tartozó csapda aktív lesz. A csapda működéséből adódóan a doboz eltűnik.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy sikerült-e a tolás, továbbá a Button benyomódott-e és a Button-hoz csatolt TrapDoor kinyílt és ami rajta volt az beleesett aztán meghalt.

* **Bemenet**

create Field F1;

create Field F2;

create Button B1;

create TrapDoor TD1;

create Player P1;

create Box Bx1;

create Box Bx2;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “B1” “RIGHT”;

connect Neighbour “B1” “TD1” “RIGHT”;

connect Button-TrapDoor “B1” “TD1”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “Bx1”;

connect Moveable “TD1” “Bx2”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push box to button(trapdoor open and vanishe a box) test: Successful

Field F1 has been created.

Field F2 has been created.

Button B1 has been created.

TrapdDoor TD1 has been created.

Player P1 has been created.

Box Bx1 has been created.

Box Bx2 has been created.

F2 connected to the Right side of F1.

B1 connected to the Right side of F2.

TD1 connected to the Right side of B1.

B1 button connected to TD1 trapdoor.

P1 has been moved to F1.

Bx1 has been moved to F2.

Bx2 has been moved to TP1.

P1 has been selected.

Moving P1 Right

Moving B1 Right

Bx1 has been moved to B1.

TD1 is set to active.

Bx2 has been destroyed.

P1 has been moved to F2.

* + 1. **Ember tol dobozt ami beltolja a mellette lévő embert egy lyukba**
* **Leírás**

Az ember lyuk általi megölését teszteljük.

* **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy minden Moveable lépet egyett és a sor végén álló Player beletolódott-e a Hole-ba és megsemmisült-e.

* **Bemenet**

create Field F1;

create Field F2;

create Field F3;

create Hole H4;

create Player P1;

create Player P2;

create Box B1;

connect Neighbour “F1” “F2” “RIGHT”;

connect Neighbour “F2” “F3” “RIGHT”;

connect Neighbour “F3” “H1” “RIGHT”;

connect Moveable “F1” “P1”;

connect Moveable “F2” “B1”;

connect Moveable “F3” “P2”;

choose “P1”;

step “RIGHT”;

* **Elvárt kimenet**

Player push box, box push another player to hole test: Successful

Field F1 has been created.

Field F2 has been created.

Field F3 has been created.

Hole H1 has been created.

Player P1 has been created.

Player P2 has been created.

Box B1 has been created.

F2 connected to the Right side of F1.

F3 connected to the Right side of F2.

H1 connected to the Right side of F3.

P1 has been moved to F1.

B1 has been moved to F2.

P2 has been moved to F3.

P1 has been selected.

Moving P1 Right

Moving B1 Right

Moving P2 Right

P2 has been moved to H1

P2 died.

P2’s score has been updated.

B1 has been moved to F3.

P1 has been moved to F2.

* + 1. **Ember tol dobozt ami beletolja a mellette lévő dobozt egy lyukba**

1. **Leírás**

Az doboz lyuk általi megsemmisítését teszteljük.

1. **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy minden Moveable lépett egyet és a sor végén álló Box megsemmisült a Hole-ban.

1. **Bemenet**

create Field F1

create Field F2

create Field F3

create Hole H1

create Player P1

create Box B1

create Box B2

connect Neighbour F1 F2 RIGHT

connect Neighbour F2 F3 RIGHT

connect Neighbour F3 H1 RIGHT

connect Moveable F1 P1

connect Moveable F2 B1

connect Moveable F3 B2

choose P1

step RIGHT

1. **Elvárt kimenet**

Player push box, box push another box to hole test: Successful

Field F1 has been created.

Field F2 has been created.

Field F3 has been created.

Hole H1 has been created.

Player P1 has been created.

Box B1 has been created.

Box B2 has been created.

F2 has been connected to the RIGHT side of F1.

F3 has been connected to the RIGHT side of F2.

H1 has been connected to the RIGHT side of F3.

P1 has been placed to F1

B1 has been placed to F2

B2 has been placed to F3

P1 has been selected.

Moving P1 RIGHT

Moving B1 RIGHT

Moving B2 RIGHT

Hole has killed a Moveable.

B2 has been moved to H1.

B1 has been moved to F3.

P1 has been moved to F2.

* 1. ***A tesztelést támogató programok tervei***

A tesztelő program fog a felhasználótól inputot be venni vagy előre megírt script formában vagy mint egy interpreter amivel az általunk definiált bemeneti nyelv parancsaival lehet majd kommunikálni. Ezeket továbbítja majd a Programunk felé az pedig a lezajlódó folyamatok alapján válaszol majd és a válasz alapján a segéd program a kimeneti nyelven tesztek lefuttatása után kimenetet generál majd.

* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 20118.04.07. 15:00 | 6 óra | Czetter  Frontó  Tomori  Verasztó | Az osztályok dokumentálása, véglegesítése.  Tesztesetek kidolgozása. |
| 20118.04.07. 15:00 | 4 óra | Czetter  Frontó  Tomori  Verasztó | Kimeneti nyelv fixálása, kimenetek kiírása. Végső simítások. |

**10. Prototípus beadása**

12 - ch33kybr33ky\_labz

Konzulens:

Bodó Zsófia

Csapattagok

|  |  |  |
| --- | --- | --- |
| **Frontó András** | **G392CQ** | andrasfronto@gmail.com |
| **Czetter Gábor** | **CP6OG3** | cz3tt1@gmail.com |
| **Tomori Dániel** | **TDWSF5** | tomori.daniel@gmail.com |
| **Verasztó Bálint** | **VSDRBZ** | balintveraszto@gmail.com |

1. **Prototípus beadása**
   1. ***Fordítási és futtatási útmutató***
      1. **Fájllista**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fájl neve** | **Méret** | **Keletkezés ideje** | **Tartalom** |
| Box.java | 4 KB | 2018. ‎április ‎22. | Box-ot megvalósító osztály. |
| Button.java | 2 KB | 2018. ‎április ‎22. | Button-tmegvalósító osztály. |
| Direction.java | 1 KB | 2018. ‎április ‎15. | Direction enumeráció. |
| Field.java | 5 KB | 2018. ‎április ‎22. | Field-et megvalósító osztály. |
| Game.java | 7 KB | 2018. ‎április ‎22. | Game-et megvalósító osztály. |
| Hole.java | 1 KB | 2018. ‎április ‎22. | Hole-t megvalósító osztály. |
| Interpreter.java | 12 KB | 2018. ‎április ‎22. | Interpreter-t megvalósító osztály. |
| Map.java | 17 KB | 2018. ‎április ‎22. | Map-et megvalósító osztály. |
| Moveable.java | 2 KB | 2018. ‎április ‎22. | Moveable-t megvalósító osztály. |
| Player.java | 4 KB | 2018. ‎április ‎22. | Player-t megvalósító osztály. |
| Target.java | 1 KB | 2018. ‎április ‎22. | Target-et megvalósító osztály. |
| TrapDoor.java | 1 KB | 2018. ‎április ‎22. | TrapDoor-t megvalósító osztály. |
| Wall.java | 1 KB | 2018. ‎április ‎22. | Wall-t megvalósító osztály. |
| MANIFEST.MF | 1 KB | 2018. április 22. | .jar fájl generálásához |
| Prototipus.jar | 20 KB | 2018. április 22. | A lefordított projekt ebbe van becsomagolva |

* + 1. **Tesztlista**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fájl neve** | **Méret** | **Keletkezés ideje** | **Tartalom** |
| Player\_box\_box\_to\_hole | 1 KB | 2018. ‎április ‎22. | Player tol dupla Box-ot és a végén egy Hole van |
| Player\_box\_player\_box\_to\_target | 1 KB | 2018. ‎április ‎22. | Player tol Box-ot ami megtol egy másik Player-t aki betolja az előtte lévő Box-ot a Target-be |
| Player\_box\_player\_to\_hole | 1 KB | 2018. ‎április ‎22. | Player tol Box-ot ami megtol egy másik Player-t aki beleesik egy lyukba. |
| Player\_can\_push\_with\_oil | 1 KB | 2018. ‎április ‎22. | Player tol dobozokat olajos Field-en így nem veszít az erejéből. |
| Player\_cant\_push\_with\_honey | 1 KB | 2018. ‎április ‎22. | Player nem tudja eltolni a Box-okat mert a méz elveszi az erejét |
| Player\_push\_box\_(fail) | 1 KB | 2018. ‎április ‎22. | Player próbál Box-ot tolni, de nem sikerül |
| Player\_push\_box\_(success) | 1 KB | 2018. ‎április ‎22. | Player próbál Box-ot tolni és sikerül |
| Player\_push\_box\_to\_button | 1 KB | 2018. ‎április ‎22. | Player tol Box-ot Button-ra |
| Player\_push\_box\_to\_button\_(box\_on\_trapdoor) | 1 KB | 2018. ‎április ‎22. | Player tol Box-ot Button-ra, ami miatt kinyílek a TrapDoor és a rajta lévő Box leesik. |
| Player\_push\_box\_to\_button\_(player\_on\_trapdoor) | 1 KB | 2018. ‎április ‎22. | Player tol Box-ot Button-ra, ami miatt kinyílek a TrapDoor és a rajta lévő Player leesik. |
| Player\_push\_box\_to\_hole | 1 KB | 2018. ‎április ‎22. | Player tol Box-ot Hole-ba |
| Player\_push\_box\_to\_player\_(die) | 1 KB | 2018. ‎április ‎22. | Player tol Box-ot másik Player-re aki meghal |
| Player\_push\_box\_to\_player\_(survive) | 1 KB | 2018. ‎április ‎22. | Player tol Box-ot másik Player-re de túléli |
| Player\_push\_box\_to\_target | 1 KB | 2018. ‎április ‎22. | Player betolja a Box-ot a célba |
| Player\_push\_box\_to\_trapdoor | 1 KB | 2018. ‎április ‎22. | Player TrapDoor-ra tol egy Box-ot |
| Player\_push\_box\_with\_honey | 1 KB | 2018. ‎április ‎22. | Player mézes Field-en tol meg egy Box-ot |
| Player\_push\_player | 1 KB | 2018. ‎április ‎22. | Player tolna másik Playert de nem sikerül neki |
| Player\_step\_to\_button | 1 KB | 2018. ‎április ‎22. | Player gombra lép |
| Player\_step\_to\_hole | 1 KB | 2018. ‎április ‎22. | Player Hola-ra lép |
| Player\_step\_to\_target | 1 KB | 2018. ‎április ‎22. | Player Target-re lép |
| Player\_step\_trapdoor\_(inactive) | 1 KB | 2018. ‎április ‎22. | Player inaktív TrapDoor-ra lép |

* + 1. **Fordítás**

Ezt a megfelelo mappabol kell futatni!!!

|  |
| --- |
| javac killer\_sokoban/\*.java |

* + 1. **Futtatás**

Program indítésa standard bemenettel. (Ilyenkor mi írjuk a script kódot)

|  |
| --- |
| java killer\_sokoban.Game  **VAGY**java -jar Prototipus.jar |

Program indítása előre megírt teszteset betöltésével.

|  |
| --- |
| java killer\_sokoban.Game <eleresi ut a scripthez>  **VAGY**  java -jar Prototipus.jar <eleresi ut a scripthez> |

Illetve 2 kattintás a jar-ra is elvileg elindítja ha minden adott hozza.

* + 1. **Script nyelv használata**

Két lehetőség áll a felhasználók rendelkezésére. Vagy betöltenek egy előre megírt tesztet és így csak a kimenetet kell ellenőrízniük, vagy a script nyelv kihasználásával nekiállnak egy saját teszteset megalkotásának.

Ha az utóbbi mellett döntöttek akkor a bemenetnek több mindennek meg kell felelnie:

* A parancsokat enter kell, hogy elválassza egymástól. (A legutolsó parancs után is szükséges egy enter)
* A parancson belül a részek között szóközt kell írni, más karakter nem elfogadott.
* A connect parancsban csak létrehozott Field-ekkel és Moveable-ekkel lehet dolgozni.
* choose parancs rész után csak Player neve állhat
* step parancsot csak a choose parancs használata után lehet használni
* Direction-t kötelezően csupa nagybetűvel kell írni.
* create Map parancs egy kész map-et generál és kis is rajzolja, viszont nem használható rajta a többi parancs.

Felhasználható parancsok:

(Az idéző jelek a parancsban nem kellenek, itt csak a jobb elszeparálást szolgálják.)

* ***create Map***

***(Random generál egy Map-et)***

* ***create Field “Field neve”***

***(Létrehoz egy Field-et)***

* ***create Wall “Wall neve”***

***(Létrehoz egy Wall-t)***

* ***create Hole “Hole neve”***

***(Létrehoz egy Hole-t)***

* ***create Button “Button neve”***

***(Létrehoz egy Button-t)***

* ***create TrapDoor “TrapDoor neve”***

***(Létrehoz egy TrapDoor-t)***

* ***create Target “Target neve”***

***(Létrehoz egy Target-et)***

* ***create Player “Player neve”***

***(Létrehoz egy Player-t)***

* ***create Box “Box neve”***

***(Létrehoz egy Box-ot)***

* ***connect Neighbour “Field1 neve” “Field2 neve” “Direction”***

***(Szomszédol két mezőt a megadott irányba)***

* ***connect Moveable “Field neve” “Moveable neve”***

***(Ráhelyez egy Moveable-t a megadott Field-re)***

* ***connect Button-TrapDoor “Button neve” “TrapDoor neve”***

***(Button-t és a TrapDoor-ját köti össze)***

* ***choose “Player neve”***

***(Kiválaszt egy Player-t)***

* ***step “Direction”***

***(Lép a megadott irányba a kiválasztott Player)***

* ***putSomeThing “Field neve”***

***(A megadott Fieldet először olajossá, majd másodszorra beütve mézessé teszi)***

* ***listNeighbour***

***(Kilistázza a kiválaszott Player Field-jének szomszédait)***

* 1. ***Tesztek jegyzőkönyvei***
     1. **Player\_box\_box\_to\_hole**

|  |  |
| --- | --- |
| **Tesztelő neve** | Czetter Gábor |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_box\_player\_box\_to\_target**

|  |  |
| --- | --- |
| **Tesztelő neve** | Frontó András |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_box\_player\_to\_hole**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tomori Dániel |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_can\_push\_with\_oil**

|  |  |
| --- | --- |
| **Tesztelő neve** | Verasztó Bálint |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_cant\_push\_with\_honey**

|  |  |
| --- | --- |
| **Tesztelő neve** | Czetter Gábor |
| **Teszt időpontja** | 2018. ‎április ‎22. |

|  |  |
| --- | --- |
| **Tesztelő neve** | Czetter Gábor |
| **Teszt időpontja** | 2018. április 22. |
| **Teszt eredménye** | Player eltudta tolni a dobozokat a surlódás ellenére |
| **Lehetséges hibaok** | Hibás Step() metódus |
| **Változtatások** | Step() javítása |

* + 1. **Player\_push\_box\_(fail)**

|  |  |
| --- | --- |
| **Tesztelő neve** | Frontó András |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_push\_box\_(success)**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tomori Dániel |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_push\_box\_to\_button**

|  |  |
| --- | --- |
| **Tesztelő neve** | Verasztó Bálint |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_push\_box\_to\_button\_(box\_on\_trapdoor)**

|  |  |
| --- | --- |
| **Tesztelő neve** | Czetter Gábor |
| **Teszt időpontja** | 2018. ‎április ‎22. |

|  |  |
| --- | --- |
| **Tesztelő neve** | Czetter Gábor |
| **Teszt időpontja** | 2018. ‎április ‎22. |
| **Teszt eredménye** | A Box nem esik le az éppen nyíló TrapDoor-on |
| **Lehetséges hibaok** | Hibás a TrapDoor FieldAction() metódusa. |
| **Változtatások** | FieldAction() javítva |

* + 1. **Player\_push\_box\_to\_button\_(player\_on\_trapdoor)**

|  |  |
| --- | --- |
| **Tesztelő neve** | Frontó András |
| **Teszt időpontja** | 2018. ‎április ‎22. |

|  |  |
| --- | --- |
| **Tesztelő neve** | Frontó András |
| **Teszt időpontja** | 2018. ‎április ‎22. |
| **Teszt eredménye** | A Player nem esik le az éppen nyíló TrapDoor-on |
| **Lehetséges hibaok** | Hibás a TrapDoor FieldAction() metódusa. |
| **Változtatások** | FieldAction() javítva |

* + 1. **Player\_push\_box\_to\_hole**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tomori Dániel |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_push\_box\_to\_player\_(die)**

|  |  |
| --- | --- |
| **Tesztelő neve** | Verasztó Bálint |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_push\_box\_to\_player\_(survive)**

|  |  |
| --- | --- |
| **Tesztelő neve** | Czetter Gábor |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_push\_box\_to\_target**

|  |  |
| --- | --- |
| **Tesztelő neve** | Frontó András |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_push\_box\_to\_trapdoor**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tomori Dániel |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_push\_box\_with\_honey**

|  |  |
| --- | --- |
| **Tesztelő neve** | Verasztó Bálint |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_push\_player**

|  |  |
| --- | --- |
| **Tesztelő neve** | Czetter Gábor |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_step\_to\_button**

|  |  |
| --- | --- |
| **Tesztelő neve** | Frontó András |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_step\_to\_hole**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tomori Dániel |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_step\_to\_target**

|  |  |
| --- | --- |
| **Tesztelő neve** | Verasztó Bálint |
| **Teszt időpontja** | 2018. ‎április ‎22. |

* + 1. **Player\_step\_trapdoor\_(inactive)**

|  |  |
| --- | --- |
| **Tesztelő neve** | Czetter Gábor |
| **Teszt időpontja** | 2018. ‎április ‎22. |

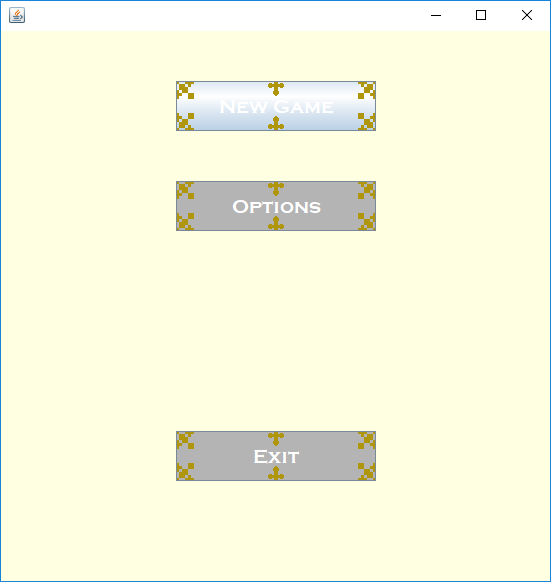
* 1. ***Értékelés***

|  |  |
| --- | --- |
| **Tag neve** | **Munka százalékban** |
| Czetter Gábor | 25% |
| Frontó András | 25% |
| Tomori Dániel | 25% |
| Verasztó Bálint | 25% |

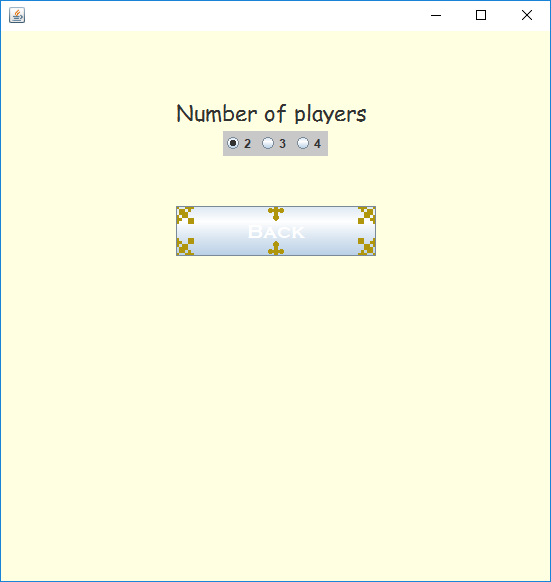
* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.04.16. 13:00 | 6 óra | Czetter  Frontó  Tomori  Verasztó | Megbeszélés, feladatok kiosztása, tesztekesetek megírása programozás elkezdése. |
| 2018.04.17. 13:00 | 4 óra | Czetter  Frontó  Tomori  Verasztó | Tesztesetek megírása, programozás. |
| 2018.04.20  16:00 | 2 óra | Czetter  Frontó  Tomori  Verasztó | Programozás. |
| 2018.04.21. 13:00 | 5 óra | Czetter  Frontó  Tomori  Verasztó | Programozás. |
| 2018.04.22. 16:00 | 7 óra | Czetter  Frontó  Tomori  Verasztó | Tesztek végső futtatása, dokumentáció megírása. |

1. **Grafikus felület specifikációja**
   1. ***A grafikus interfész***

**

Itt a játék menüje látható. Ez az a felület amit a felhasználók először pillantanak meg a játék indítása során. Innen lehet új játékot indítani a “New Game” gombra kattintva, továbbá ki lehet választani a játékosok számát egy almenüben amit pedig az “Options” gombbal lehet megnyitni, Ha nem szeretnénk tovább játszani akkor pedig az “Exit” feliratú gombra kattintva leállíthatjuk a program futását.

**

Itt látható az Options almenü képe. Itt lehet kiválasztani, hogy mennyi játékossal szeretnénk játszani.



Itt láthathó pedig maga a játék. 11x11-es négyzetnyi területen terül el a random generált térkép. Minden alkalommal a pálya szélét falak határolják így nem lehet kimenni róla. A többi  Alul pedig folyamatosan nyomonkövethető a Playerek aktuális pontszáma.

Felhasznált elemek:

Player:

https://lh5.googleusercontent.com/c3PAVWMCQ0-8TioAOgLN_u_vy07mtKkpKse79_vPwHQvowgTfdOtz4BJeCpFNk4qk8AwudDkI2qqcPKvnWCLlBh3ojlDppxxxoLrY4cTVVSkGQUKn2KPloVjhMWLoKWX8ICGtnzntvy6a7pDpwhttps://lh5.googleusercontent.com/KzWTgO0ng_s6BKmbPOUzmPIUD9PiX0Q-Mdn9yBR3T5IEzhtvHeq_erLBWJSJhVEmN_yHhNujZtK6AaXncESQPZesmRrbd561EW5qJAiBjVwyVWz-91n4WDWbr5F7SruqGVUrjjS19L-jhIzLQghttps://lh6.googleusercontent.com/q8oon6SmeuD02xGPF3r-KEpCmnqkhIyf2nvxwX1HM9_sf2RA7nx3zT70mwMcZ4LIevVi0_uanxdOqOC-e7IuX3x5v4iUYXOxQGQpwJIiuTq2Dm-Qx_qkutXyHK2jYLqZTxO9iD69Jdq9SHNPvwhttps://lh6.googleusercontent.com/cKOzE5vattQWc55CkUwDWJmypuGr2JTYPz9UWilwEieJ5q_3e3YQTujYR1J3of1lTHp54nUhY_BGKUMbKwntSKLJY632Z6UYht0OEDD5E8_bkEmPoYU4WctRNtrfh9oCsAwSPzdncQX9xmyLjQ

A Playereknek felülnézet miatt csak a sapkájuk látszik. A különböző színnel vannak megkülönböztetve, így nem lesz kavarodás a játék során.

Box:

https://lh3.googleusercontent.com/096HsGapl4GdcN7lEqjOSEyiBnFye8gP8NaElkGZVnIc7nQf5IdJqm0ao_IffY_sRmUjAZBjVq5HTLEnHw30ebNcWvlCWiNbO6nrCSAl0eU6IEzbQvWtFiuUK56XBI8xza3xQnd1BHN4zUle0Q

Ezeket tologatják a Playerek a játék során.

Wall:

https://lh3.googleusercontent.com/cy4gNes1Bx3ydoS-i2_BRasF3lxMrDOzXje3M76yUjxbokQbf91NuLNZOwUIaC4WjGGW29CmXqToEobfTu9GeIDJRG1mT8fh4bgmKTniRT_fQtRp1TXRIjbxifADMIiQ2_NaL6hn60Y83c9hCw

A programban öröklődik a Box-ból. A cél az volt, hogy ezt a grafikus felületen is érzékeltessük így egy robosztusabb doboz lett a végeredmény.

Field:

https://lh6.googleusercontent.com/9TqS81vxQpEwI03ECI4BfjdRZNeEEaHenKcNSAlWJU2KBLvnwr9PKcToWk8ab33GJ_jARPnmyjZNSfglOiucnXrFx-dkRb-6fBeqymv9q1Wkm8JLPv0roTzE3GC7m7kASJwsEXokxQIcdMZkbQ

Kis kerettel ellátott egyszínű háterű mező.

Hole:

https://lh4.googleusercontent.com/Kv2Yy91lGGuCRJc0G5NoosDIKEUhPsMqOwxQy37Qw51na60CG3JaQ3Qz-iZutLP3ASOpjOxZ63sgBi3Rv7KA1iHKF0biZ_4Y4fxRp4v0ohpKg-F6t3FdDqBTO6tW8RvWycGDOacomPDl-9aa6A

A Field sötét színe miatt, nem lett volna célszerű egy egyszerű fekete repedést rájzolni, mivel a játékosok könnyen figyelmen kívül hagyták volna és minden ok nélkül követtek volna el öngyilkosságot. Így egy világosabb szürke fémes keretet kapott, hogy jobban felhívja magára a figyelmet.

Button:

https://lh4.googleusercontent.com/x_W_vPHYTwkIhllXebbSKJ9j-a5qEJBxFRmAqKkwNTP3UlOEn9SbPbZBYhokCqM3RdFOS_mom323zSsHBsU-8bdRtsX7lNiU8Rof_6IZxjjsmHcNPxeYK0_i266ELiDVpbx8-0HAuAwdfM9f8g

Egy piros gomb ami hasonló mintát kapott mint ami a Trapdoor-on van. Így az új játékosok könnyebben rájönnek, hogy a kettő között valami féle kapcsolat van.

Trapdoor:

https://lh4.googleusercontent.com/a9h86bO5pwlGujnhDEQuXz2Js2afx2LZx3E1bIU7mRv_BLVnfggqbkabD8eotPQW7r7Wo3ebeX4LyaE7ZXGbvpA1cF7DQ0mkOroV2n0omThgBmpSPZMGTyeUShYUs_N_Yi4pjIzNCoe56dWx9ghttps://lh4.googleusercontent.com/n6dm7uCM43x-cXq1E78r9-y0EOCKA84yEnYQyslyS2lSY3w3WXIbwTOcYhv4wohjW0GdzImNVZn70XMWqYH93VCVQdV20Y9l43LHj-iCB8se0h5dk2yXnjX7Op_f9IuyyselRt6fMnMUmrjXQw

Button megnyomására kinyílik, ha a Button inaktívvá válik akkor pedig visszacsukódik.

Target:

https://lh6.googleusercontent.com/INp5_9VaBBFqGPqRqgmvJ5VxW0lwemvrgozn8w5sQuBWOgEK0vJNiJgYPN_SoNQ3qDu0Shs_bJe1_QU_bN5UYyebb5z-PLJy2Xzgru0YDr5NqVik_kuVxKZmAAuq_WPGz4ySjS0E6lOSc1iQgw

Mivel ez a játék központi része, ezért fontos volt, hogy ez legyen az egyik legdíszesebb Field. Így első pillantásra a játékos látja, hogy hol helyezkedik el, és elkezdhet gondolkodni, hogy melyik a legrövidebb út felé.

Honey/Oil

https://lh4.googleusercontent.com/vOnOoirpy6vjhu5IR5RH4QezCos7iHNrkBnbsuA8g0ww7rA-hMFo3yZ5IkypjS_2uJlVbXQzrfD-Y7rWO-4XqIvzQ1WAjE02sHtAEU9d6ELIEAOXWsm8PkxcHnaFTQi1C7Rs5BrZDZw-fKog9Ahttps://lh4.googleusercontent.com/R8-pU3wNYMbWY3vWNfVlg9hTvqzF36DAPPzVYVU1DKtP4WffzlHkB55bZ-gKu0_3d0lELY5J7Cgdsv41vzLrNJjbBwinVGyYfHhVYaDwIlJeemFSl6cEXKymjDjRfqO02VlCcBETcmp0TAaHzghttps://lh6.googleusercontent.com/LtW3s3I5wOgLQR9mWTJEGkNGv5nmGG0lkE0HNgSO9qpSGRSC_M4cmg28VqLWIlcGzltN9uICtuTbkkF1JwEMhA0HJP9xQ0AOSJDYppQ4Zjc7JRz1-bo2xmzx-DE41DnEj2KuwXGQTHv3Cyg3NAhttps://lh6.googleusercontent.com/dOGgdB623X8u-9D_DmstMv5b_dyWDQLb22-_evhDML02HsUXKNk6fAkotU-LmENSXr48Y5DZPkxxPzSJPt5D1OUFkYbvr_zbPdOEtfR7uw1tv52rRTU3Xa4ZhhE3lqgnglYIWj_EYUSHmBBJHQhttps://lh5.googleusercontent.com/uKaEmuGB3oDlJX0ZkJ_WBA0AVA01j8B-OcGcwAXtquNtYywPx_RRdjHp1QF3MYd4hNYvhxKweXE3n78WniZPvOeq2h7BDg1bhWJsLDSxAMZR689nxakpHgvlyVS-xje3DpEzWkKVKOy3FDT8cQ

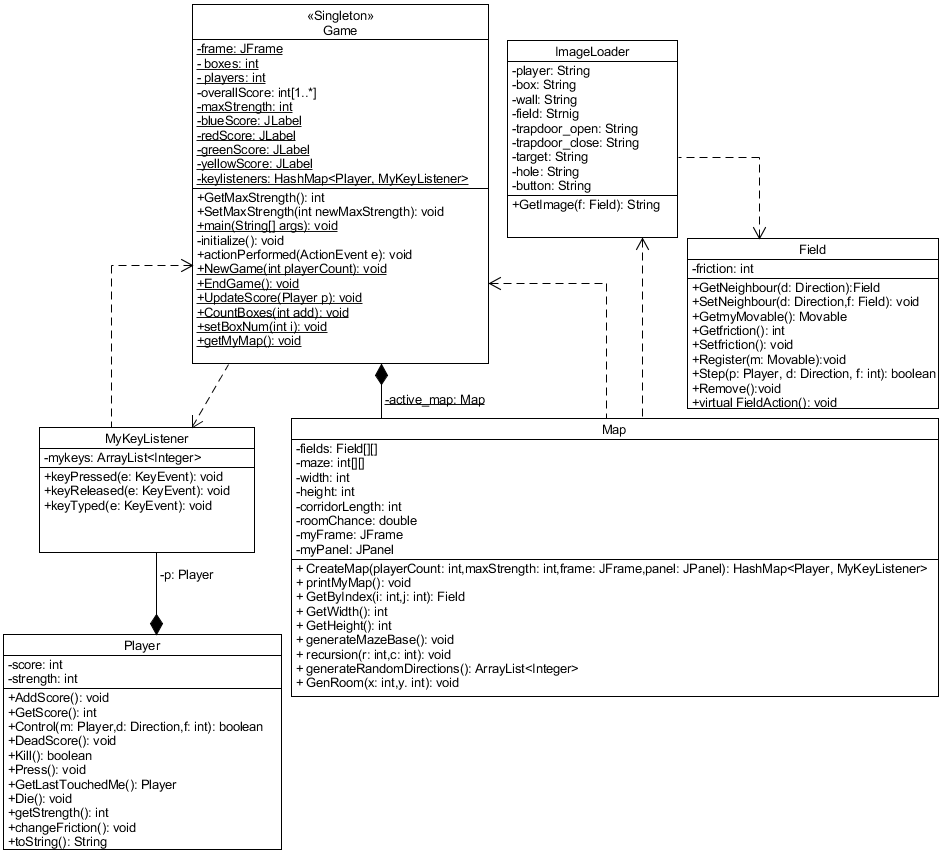
https://lh5.googleusercontent.com/OFkAh9oPEdoosLJhSSGwSmzZHdZ121aXlcK672oYoEACVbEabtEPmxu47jC3ac0caiUz9hlxIB_I9rFukpyZrgPXYrwwkOqo51AGfWY1uB-eqqIJgHp9cYKl-ElN4IuZ1z8HzROQo14aeFK1YAhttps://lh3.googleusercontent.com/3SIzk7jLwhGgPKRz9YDcbHvpdii4gwQKXvm9BEQ4kTIKoDylyb2EbzzyR9cDvGiydqEV28HjyWB2iIXDECuJyxJikGBCuBYV0QOiGQJ2I1GTBzy3nEayrSetj51JXnzTvqjK7gLiLYj5nTONFwhttps://lh4.googleusercontent.com/6l5Qipoq0bTjrXxbHnYwoFxxAP2CfVLJ0V7o9jrkQYl3cvNDaZfyXyYx5A65k8l82fm25lTB4n3s7Fy-vJOVRW7NL5qdQloXUKVZetkkDtiESRsL1fl7B4J7Te1KD5essx9xVN8nHNyHwhK5qQhttps://lh4.googleusercontent.com/BBXV82Qav-e6ScE3rMSGgpqZ8REL72v9DRvsQsQhZnpp_RcAx_DaEHRT4xq-lmHYMsffU5hqtXrbGcEjCJ39wMGuYtACxoGGz5azcIX8ohkWpIXzw1kSLQviQtqIKhvAz3XeMVbVeA_KdSQCEQhttps://lh3.googleusercontent.com/Ca2ZyIxa_TZhVuVwOzefGFwlH0fcs_Auebd-G6FO-jzuQdVL1BRo_5tB4zMPSmC-fDsBlXsS4WiNRA3MnRWr38_-SqSgDntC9W0ce9NObMC0zov4t1MAhZX6XBUviYVNIAWNGWLkVIkW_j4obA

Minden mezőre ahova lehet Honey-t vagy Oil-t lerakni, külön meglett alkotva a bemocskolt verziójuk.

* 1. ***A grafikus rendszer architektúrája***
     1. **A felület működési elve**

A Grafikus felület megvalósításához a lehető legkevesebb változtatásra törekedtünk. Minden Field vagy Moveable leszármazott osztály tárolja mostantól az őt reprezentáló kép elérési útját. Ezt pedig az ősosztályokban definiált  GetImage függvénnyel lehet elérni. A grafikus felület kirajzolásáért a Game és a Map osztályok felelősek. A Game felrajzolja az ablakot és a menü is innen érhető el. Magát a játékot a Mapben tároltak alapján a Map új függvénye a printMyMap() fogja megjeleníteni. Ezen felül a Map elkészítésekor minden playerhez kell rendelni egy MyKeyListenert ami egy új osztály. Ez az osztály felelős a Playerek mozgatásáért. A kép mindig akkor frissül amikor valaki lép. Ezért a MyKeyListener felel ő hívja meg a lépés után a Game Singleton osztály getMyMap függvényét ami továbbítja ezt a frissitési szándékot a Map felé, majd az újra rajzolja magát. Tehát push alapelvet alkalmaztunk a felület megvalósítása során.

* + 1. **A felület osztály-struktúrája**

****

* 1. ***A grafikus objektumok felsorolása***
     1. **Game**
        + **Felelősség**

Az eddigieken túl, a grafikus megjelenítésért is  felelős lett.

* + - * **Ősosztályok**

**-**

* + - * **Interfészek**

**-**

* + - * **Attribútumok**
* - JFrame frame
* - JPanel panel
* - JLabel yellowScore
* - JLabel greenScore
* - JLabel redScore
* - JLabel blueScore
  + - * **Metódusok**
* **private void initialize()**: Inicializálja a játék ablakát, létrehozza a gombokat, stb.
* **public void actionPerformed(ActionEvent e):**  RadioButton választásakor fut le, playerek számát változtatja
* **public static void UpdateScore(Player p):** Az eddigi funkciók mellett változtatja a Playerek pontjait kiíró JLabel szövegét. Illetve figyel hogy egy halott játékos már ne mozoghasson többet ezért törli a hozzá tartozó MyKeyListener-t.
  + 1. **Map**
       - **Felelősség**

Az eddigieken túl, ki is rajzolja a mapot.

* + - * **Ősosztályok**

**-**

* + - * **Interfészek**

**-**

* + - * **Attribútumok**
* **- JFrame myFrame**
* **- JPanel myPanel**
  + - * **Metódusok**
* **public HashMap<Player, MyKeyListener> CreateMap(int playerCount, int maxStrength, JFrame frame,JPanel panel):** Az eddigi feladatai mellett MyKeyListenereket rendel a playerekhez. Illetve megjegyzi a parameterkent kapott JFramet és JPanelt a késöbbiekre.
* **public void printMyMap() :**  A korábban eltárolt JPanelre kirajzolja a  mezőket.
  + 1. **MyKeyListener**
       - **Felelősség**

Ez egy teljesen új osztály aminek a feladata, hogy billentyű lenyomáskor léptesse a megfelelő Playereket.

* + - * **Ősosztályok**

-

* + - * **Interfészek**

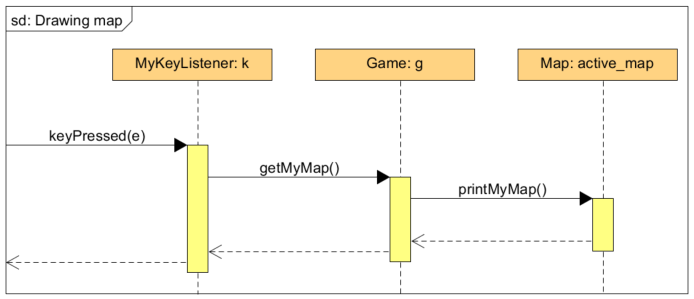
KeyListener

* + - * **Attribútumok**
* -Player p: A hozzá tartozó Playert tárolja akit mozgat
* - ArrayList<Integer> mykeys: A gombok kódjai amikre reagál
  + - * **Metódusok**
* **-void keyPressed(KeyEvent e):** Örökli a KeyListener Interfésztől. Ebben a függvényben van definiálva hogy melyik gomb lenyomásakor milyen akciót kezdeményez.

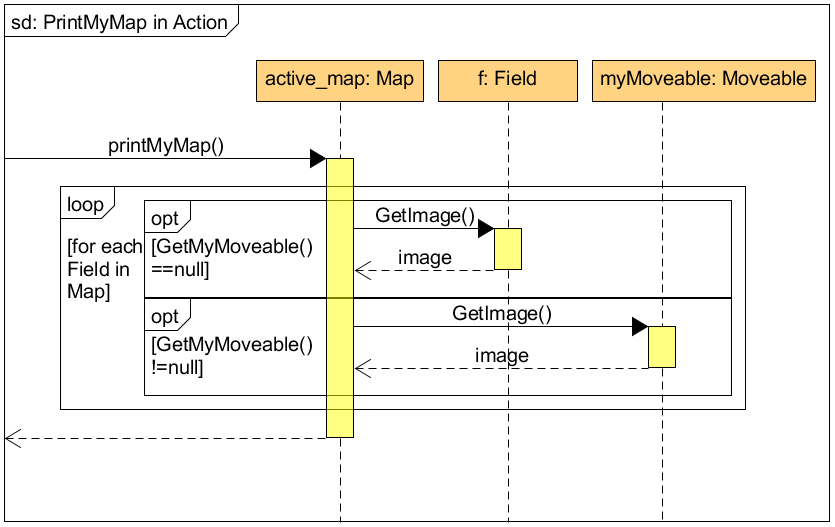
*A többi interfésztől örökölt függvényt bár megvalósítom de üresen hagyom mert azokra nincs szükség.*

* 1. ***Kapcsolat az alkalmazói rendszerrel***

**11.4.1 Map kirajzolása**

**

**11.4.2 Map kirajzolása „trigger” hatására**



* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.04.29. 15:00 | 4 óra | Czetter  Frontó  Tomori  Verasztó | Grafikus felület rajzolása. |
| 2018.04.30. 15:00 | 4 óra | Czetter  Frontó  Tomori  Verasztó | Diagrammok készítése, tervezés. |
| 2018.05.01. 17:00 | 4 óra | Czetter  Frontó  Tomori  Verasztó | Munkafolyamatok folytatása, hibák keresése és javítása. |

**13. Grafikus változat beadása**

12 - ch33kybr33ky\_labz

Konzulens:

Bodó Zsófia

Csapattagok

|  |  |  |
| --- | --- | --- |
| **Frontó András** | **G392CQ** | andrasfronto@gmail.com |
| **Czetter Gábor** | **CP6OG3** | cz3tt1@gmail.com |
| **Tomori Dániel** | **TDWSF5** | tomori.daniel@gmail.com |
| **Verasztó Bálint** | **VSDRBZ** | balintveraszto@gmail.com |

1. **Grafikus változat beadása**
   1. ***Fordítási és futtatási útmutató***
      1. **Fájllista**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fájl neve** | **Méret** | **Keletkezés ideje** | **Tartalom** |
| Box.java | 4 KB | 2018.05.13. | Box-ot megvalósító osztály |
| Button.java | 2 KB | 2018.05.13. | Button-t megvalósító osztály |
| Direction.java | 1 KB | 2018.05.13. | Dirction enumeráció |
| Field.java | 5 KB | 2018.05.13. | Field-et megvalósító osztál |
| Game.java | 10 KB | 2018.05.13. | Game-et megvalósító osztály |
| Hole.java | 1 KB | 2018.05.13. | Hole-t megvalósító osztály |
| ImageLoader.java | 2 KB | 2018.05.13. | ImageLoader-t megvalósító osztály |
| Map.java | 15 KB | 2018.05.13. | Map-ot megvalósító osztály |
| Moveable.java | 2 KB | 2018.05.13. | Moveable-t megvalósító osztály |
| MyKeyListener.java | 2 KB | 2018.05.13. | MyKeyListener-t megvalósító osztály |
| Player.java | 4 KB | 2018.05.13. | Player-t megvalósító osztály |
| Target.java | 1 KB | 2018.05.13. | Target-et megvalósító osztály |
| TrapDoor.java | 1 KB | 2018.05.13. | TrapDoor-t megvalósító osztály |
| Wall.java | 1 KB | 2018.05.13. | Wall-t megvalósító osztály |
| blueplayer.png | 1 KB | 2018.05.13. | Kék játékos képe |
| box.png | 1 KB | 2018.05.13. | Box képe |
| button.png | 1 KB | 2018.05.13. | Button képe |
| button\_honey.png | 4 KB | 2018.05.13. | Mézezett Button képe |
| button\_oil.png | 4 KB | 2018.05.13. | Olajozott Button képe |
| buttonbg.png | 1 KB | 2018.05.13. | Menü gombjainak háttere |
| field.png | 1 KB | 2018.05.13. | Field képe |
| field\_honey.png | 1 KB | 2018.05.13. | Mézett Field képe |
| field\_oil.png | 1 KB | 2018.05.13. | Olajozott Field képe |
| greenplayer.png | 1 KB | 2018.05.13. | Zöld játékos képe |
| hole.png | 1 KB | 2018.05.13. | Lyuk képe |
| redplayer.png | 1 KB | 2018.05.13. | Piros játékos képe |
| target.png | 1 KB | 2018.05.13. | Target képe |
| target\_honey.png | 4 KB | 2018.05.13. | Mézezett Target képe |
| target\_oil.png | 4 KB | 2018.05.13. | Olajozott Target képe |
| trapdoor\_close.png | 1 KB | 2018.05.13. | Csukott TrapDoor képe |
| trapdoor\_close\_honey.png | 1 KB | 2018.05.13. | Mézezett csukott TrapDoor képe |
| trapdoor\_close\_oil.png | 1 KB | 2018.05.13. | Olajozott csukott TrapDoor képe |
| trapdoor\_open.png | 1 KB | 2018.05.13. | Nyitott TrapDoor képe |
| trapdoor\_open\_honey.png | 1 KB | 2018.05.13. | Mézezett nyitott TrapDoor képe |
| trapdoor\_open\_oil.png | 1 KB | 2018.05.13. | Olajozott nyitott TrapDoor képe |
| wall.png | 1 KB | 2018.05.13. | Wall képe |
| yellowplayer.png | 1 KB | 2018.05.13. | Sárga játékos képe |
| MANIFEST.mf | 32 B | 2018.05.13. | Manifest file a .jar fájlhoz |
| Grafikus.jar | 24 KB | 2018.05.13. | Az előre elkészített .jar fájl |

* + 1. **Fordítás és telepítés**

A megfelelő mappában a szokásos javac-os megoldással lehet fordítani a következő paranccsal:

*javac killer\_sokoban/\*.java*

* + 1. **Futtatás**

A játék parancssorból a megfelelő mappába navigálva a következő paranccsal indítható:

*java killer\_sokoban.Game*

Vagy az általunk előre elkészített .jar filelal.

java -jar Grafikus.jar

* 1. ***Értékelés***

|  |  |
| --- | --- |
| **Tag neve** | **Munka százalékban** |
| Czetter Gábor | 25 |
| Frontó András | 25 |
| Tomori Dániel | 25 |
| Verasztó Bálint | 25 |

* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.05.07 13.00 | 5 óra | Czetter Gábor Frontó András Tomori Dániel Verasztó Bálint | Koncepció felvázolása, kódolás |
| 2018.05.13 17.00 | 5 óra | Czetter Gábor Frontó András Tomori Dániel Verasztó Bálint | Kódolás, dokumentáció elkészítése |

**14. Összefoglalás**

12 - ch33kybr33ky\_labz

Konzulens:

Bodó Zsófia

Csapattagok

|  |  |  |
| --- | --- | --- |
| **Frontó András** | **G392CQ** | andrasfronto@gmail.com |
| **Czetter Gábor** | **CP6OG3** | cz3tt1@gmail.com |
| **Tomori Dániel** | **TDWSF5** | tomori.daniel@gmail.com |
| **Verasztó Bálint** | **VSDRBZ** | balintveraszto@gmail.com |

1. **Összefoglalás**
   1. ***A projektre fordított összes munkaidő***

|  |  |
| --- | --- |
| **Tag neve** | **Munkaidő (óra)** |
| Frontó András | 120 |
| Czetter Gábor | 120 |
| Tomori Dániel | 120 |
| Verasztó Bálint | 120 |
| **Összesen** | 480 |

***• A feltöltött programok forrássorainak száma***

|  |  |
| --- | --- |
| **Fázis** | **Kódsorok száma** |
| Szkeleton | 698 |
| Prototípus | 1069 |
| Grafikus változat | 1169 |
| **Összesen** | 2936 |

* 1. ***• Projekt összegzés***
     1. **Mit tanultak a projektből konkrétan és általában?**

Csapatként dolgozni sokkal hatékonyabb mint egyedül, a megbeszélések nem okoztak különösebb problémát, jó összhang volt a csapatban. A jól szervezett közös megbeszélésekkel és rendszeres összedolgozással tartani tudtuk a határidőket.

* + 1. **Mi volt a legnehezebb és a legkönnyebb?**

A legkönyebb a kód elkészítése volt. Kiemelkedő nehézségek nem voltak, a tervezős, diagramm rajzolgatós részeknél pepecseltünk néha.

* + 1. **Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?**

Igen.

* + 1. **Ha nem, akkor hol okozott ez nehézséget?**

Nem okozott nehézséget.

* + 1. **Milyen változtatási javaslatuk van?**

A tárgy koncepciója nagyon érdekes volt és alapvetően kellemes kihívást nyújtott, bár kicsit overkillre sikeredett.