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

A dokumentum célja ismertetni a csapat által elkészítendő szoftver követelményeit, alapvető tulajdonságait, valamint a csapat által alkalmazott segédeszközöket, folyamatokat.

* + 1. **Szakterület**

A szoftver egy játék, nem kifejezett szakterületre készül, hanem általános felhasználára, szórakoztatásra. A játék a sokoban nevű játék egy variációja.

* + 1. **Definíciók, rövidítések**

Eclipse - fejlesztőkörnyezet

funkció - A program egy elvárt, egységként kezelhető működési része

Git - verziókezelő rendszer

GitHub - Git-et használó szolgáltatás, elsősorban forráskódok kezelésére

Google Docs - webalapú szövegszerkesztő

HSZK - Hallgatói Számítógép Központ

instance - adott szoftver futó példánya

integráció - egységesítés, beolvasztás

interfész - a számítógép és a felhasználó érintkezési felülete

Java9 - a Java programozási nyelv 9-es verziója

JDK9 - Java Development Kit, Java programozási nyelvhez használt fejlesztőeszköz

Jenkins - folyamatos integrációs eszköz

JRE9 - Java Runtime Environment, Java nyelven írt alkalmazások futtatásához szükséges program

proto - A program azon állapota amikor már a belső működés megvan, de még nem rendelkezik grafikus megjelenítéssel

standard - szabvány

szkeleton - a program azon állapota amikor a belső felépítése már megvan de még nem működik

szoftver - számítógépre írt program és a hozzá tartozó dokumentáció

use-case - használati eset

WhiteStarUML - modellező eszköz melynek segítségével UML diagramokat lehet készíteni

* + 1. **Hivatkozások**

Feladatkiírás - https://www.iit.bme.hu/targyak/BMEVIIIAB02/feladat

* + 1. **Összefoglalás**

A dokumentum későbbi részeiben a következők találhatók:

* Áttekintés, ami nagyvonalakban bemutatja a szoftvert
* A szoftverrel kapcsolatos követelmények
* Lényegesebb use-case-ek felsorolása, valamint use-case diagram
* Szótár, ami a szoftverrel kapcsolatos nem hétköznapi szavak definícióit tartalmazza
* A projekt kivitelezésének terve
* Projekt napló
  1. **Áttekintés**
     1. **Általános áttekintés**

A szoftver két alapvető részből áll:

* játék logika
* megjelenítés

A megjelenítő feladata egy interfészt adni a felhasználó felé, amivel a játék logikát manipulálni lehet.

A játék logika felel a játék szabályainak betartásáért, a játék vége feltételek figyeléséért, a játék állapotának nyomon követéséért.

A Pálya alrendszer felel a pályák betöltéséért és a játéklogika számára rendelkezésre állítja azokat.

* + 1. **Funkciók**

A program egyfajta logikai játék. A célja, hogy a raktárban lévő ládákat a megfelelő helyre juttassuk, ezzel pontokat gyűjtve. A játékosok a munkások mozgatásával képesek a ládák helyzetét változtatni. A raktári rakodómunkások között folyik verseny, a kékpolós munkás és a piros polós munkás harca ez. A megfelelő ládák, megfelelő sorrendben való mozgatásával kell eljuttatni a lehető legtöbb ládát a célba. A gyors, stratégikus gondolkodás elengedhetetlen. A ládák célba juttatását lehet akadályozni is kapcsolókkal aktivált lyukakkal. A lyukakban a ládák és a játékosok is eltűnnek. Továbbá a pályán találhatóak nem vezérelt lyukak is, ezek mindig ott vannak. Így színes variációt biztosít a játék a stratégiáknak.  
A játék időre megy. Az idő lejártával a játék véget ér, ekkor a magasabb pontszámú játékos nyer. A játék idő előtt is véget érhet, amennyiben az egyik játékos beleesik egy lyukba, vagy az egyik játékos a falnak préselődik és persze fel is adhatja a játékot.

A munkások a ládákat mozgatva képesek egymást eltolni. Amennyiben nincs hely, hogy a megfelelő irányba elmozduljon a játékos, tehát falba vagy ládába ütközik, akkor meghal és a másik játékos nyer. A ládákat azonban bölcsen kell rendezni mivel egymásba nem tolhatóak és így elzárhatják a munkások útját.  
A pontokat a játék során a megfelelő mezőkre tolt ládák számával lehet növelni, így ezek maximalizálva vannak a játék kezdetekor, a raktárban található ládák számával. Így teret adva olyan taktikai meggondolásoknak mint a másik játékos ládáinak megsemmisítése a csapdákkal. A pályán a rakodóterületek, ahova a ládákat el kell juttatni, változatosak. Lehetnek több részletben elszórva, vagy egybefüggően.

A csapdák olyan kapcsolók amik nyomás, azaz súly hatására aktiválódnak és kinyitnak egy lyukat a raktár padlóján. Fontos megjegyezni, hogy a csapdákat a munkások súlya nem tudja aktiválni, ehhez egy láda súlya szükséges. Amennyiben elmozgatjuk a csapdáról a rajta található ládát, akkor a kapcsoló visszaáll és a lyuk bezárul. Érdemes tehát megfontoltan mozogni és megjegyezni a lyukak helyét. A játékban az ilyen kapcsolható csapdákon kívül vannak a pályán permanens lyukak is.

A pályák szerkezete megköveteli az előre gondolkodást és a tervezést, a munkások a pálya egy megszabott területéről indulnak. A játék indulás előtt hagy néhány másodpercet a pálya szemrevételezésére és a munkaterv megalkotására. A játék időtartama alatt a ládák helyzete jelentősen változhat. Ehhez a jó munkásnak alkalmazkodnia kell tudni, a legjobb munkásnak pedig előre látni ezeket a változásokat.

A játékosok a játék indítása utána kiválaszthatják, hogy mely adott sorszámú pályán akarják esetleg összemérni erejüket. A játék végeztével választhatunk másik pályát de akár visszavágót is kérhetünk. A játék időtartama változtatható, így adva még több változatosságot a kihíváshoz. Például a gyakorlott munkások versenyéhez egy rövidebb, izgalmasabb időkorlát adható, a kezdő rakodók pedig állíthatnak több idő a játék kitanulásához illetve hogy hozzászokjanak a raktár tempójához.

* + 1. **Felhasználók**

A felhasználók a játék természetét, a lehetőségeket és az eszközök használatát könnyen elsajátíthatják. A megfelelő rövid leírások elolvasásával, speciális képzettség nélkül játszható, korra való tekintet nélkül.

* + 1. **Korlátozások**

A szoftver elkészítéséhez csak a Java9 Standard könyvtárait lehet felhasználni.

* + 1. **Feltételezések, kapcsolatok**

Feladatleírás - https://www.iit.bme.hu/targyak/BMEVIIIAB02/feladat

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Use-case** | **Komment** |
| 1.1 | A pálya mezőkből áll | bemutatás | Alapvető | Megrendelő | View Map |  |
| 1.2 | A mezőn lehet munkás | bemutatás | Alapvető | Megrendelő | View Map |  |
| 1.3 | A munkások mozgathatók | bemutatás | Alapvető | Megrendelő | Move Worker |  |
| 1.4 | A munkások meghalhatnak | bemutatás | Alapvető | Megrendelő | Move Worker |  |
| 1.5 | A mezőn lehet lyuk | bemutatás | Alapvető | Megrendelő | View Map |  |
| 1.6 | A lyukba bele eshet a játékos | bemutatás | Alapvető | Megrendelő | Control Traps |  |
| 1.7 | A lyukba bele eshet a láda | bemutatás | Alapvető | Megrendelő | Control Traps |  |
| 1.8 | Vannak dobozok a pályán | bemutatás | Alapvető | Megrendelő | View Map |  |
| 1.9 | A dobozokat mozgathatja a játékos | bemutatás | Alapvető | Megrendelő | Move Worker |  |
| 1.10 | Doboz tolhat dobozt | bemutatás | Alapvető | Megrendelő | Move Worker |  |
| 1.11 | Doboz tolhat játékost | bemutatás | Alapvető | Megrendelő | Move Worker |  |
| 1.12 | A doboz ha célba ér eltűnik | bemutatás | Alapvető | Csapat | View Map |  |
| 1.13 | A mezőn lehet fal | bemutatás | Alapvető | Megrendelő | View Map |  |
| 1.14 | A pálya széle fal | bemutatás | Fontos | Csapat | View Map |  |
| 1.15 | A mezőn lehet kapcsoló | bemutatás | Alapvető | Megrendelő | View Map |  |
| 1.16 | A kapcsoló nyitja a lyukat ha aktív | bemutatás | Alapvető | Megrendelő | Control Traps |  |
| 1.17 | A kapcsoló zárja a lyukat ha nem aktív | bemutatás | Alapvető | Megrendelő | Control Traps |  |
| 1.18 | A kapcsolót a doboz aktiválja | bemutatás | Alapvető | Megrendelő | Control Traps |  |
| 1.19 | A kapcsolót a játékos nem aktiválja | bemutatás | Alapvető | Megrendelő | Control Traps |  |
| 1.20 | A kapcsoló visszaáll ha nincs rajta doboz | bemutatás | Alapvető | Csapat | Control Traps |  |
| 1.21 | A játék kezdete előtt a pálya megtekinthető | bemutatás | Fontos | Csapat | View Map |  |

* + 1. **Erőforrásokkal kapcsolatos követelmények**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Komment** |
| 2.1 | Git | nincs | alapvető | csapat | elolsztott verziókezelő |
| 2.2 | GitHub | nincs | alapvető | csapat | Hosztolt Git repo |
| 2.3 | Eclipse | nincs | alapvető | nincs | Integrált fejlesztőkörnyezet |
| 2.4 | Billentyűzet | nincs | alapvető | csapat |  |
| 2.5 | Egér | nincs | alapvető | csapat |  |
| 2.6 | Monitor | nincs | alapvető | csapat |  |
| 2.7 | BME HSZK-ban található PC | bemutatás | alapvető | megrendelő |  |
| 2.8 | WhiteStarUML | nincs | alapvető | csapat | UML modellező program |
| 2.9 | Google Docs | nincs | alapvető | csapat | Szövegszerkesztő |
| 2.10 | JRE9 & JDK9 | nincs | alapvető | csapat | Futtatókörnyezet |
| 2.11 | Jenkins | nincs | opcionális | csapat | Continous Integration |

* + 1. **Átadással kapcsolatos követelmények**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Komment** |
| 3.1 | Szkeleton beadás | bemutatás | alapvető | megrendelő | március 19. |
| 3.2 | Proto beadás | bemutatás | alapvető | megrendelő | április 23. |
| 3.3 | Grafikus beadás | bemutatás | alapvető | megrendelő | május 1 |
| 3.4 | Segítség nélkül, útmutató alapján telepíthető | bemutatás | alapvető | megrendelő |  |
| 3.5 | Bemutatás a HSZK-ban | bemutatás | alapvető | megrendelő |  |

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

Nincs ilyen.

* 1. **Lényeges use-case-ek**

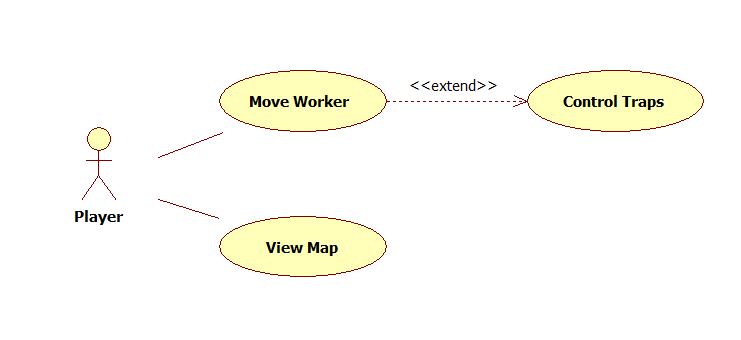
**Use-case leírások**

|  |  |
| --- | --- |
| **Use-case neve** | Move Worker |
| **Rövid leírás** | A játékos irányítja a munkást. |
| **Aktorok** | Player |
| **Forgatókönyv** | 1.A játékos felfelé ,lefelé balra,jobbra mozgathatja a munkást. |
| **Alternatív Forgatókönyv** | 1.A.1.Ha a munkás lyukra lép,akkor meghal,ekkor a játék véget ér. |
| **Alternatív Forgatókönyv** | 1.B.1.Ha a munkásra ládát tolnak akkor egy mezőt arrébb lép |
| **Alternatív Forgatókönyv** | 1.B.2.Ha a munkásra ládát tolnak,de az nem tud arrébb mozdulni akkor meghal. |

|  |  |
| --- | --- |
| **Use-case neve** | View Map |
| **Rövid leírás** | A játékos megtekinti a pályát. |
| **Aktorok** | Player |
| **Forgatókönyv** | 1.A rendszer kirajzolja a pálya aktuális állapotát.  2.A játékos megtekinti a pálya aktuális állapotát. |

|  |  |
| --- | --- |
| **Use-case neve** | Control Traps |
| **Rövid leírás** | A rendszer a padlókon lévő csapóajtókat nyitja vagy zárja. |
| **Aktorok** | Player |
| **Forgatókönyv** | 1.A csapóajtóra láda kerül és aktiválja.  2.A csapóajtóról lekerül a láda és deaktiválódik. |

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



* 1. **Szótár**

csapda, lyuk – egy mezőnyi hely a pályán melyet egy kapcsoló irányít. Két állapota van: ha zárva van padlóként azaz semlegesen viselkedik, ha nyitva akkor eltünteti a rajta tartózkodó ládát vagy munkást

fal – a pálya szélét reprezentáló objektum

kapcsoló – a munkások által állítható kétállapotú rendszer mely egy hozzá kapcsolt csapdát irányít

kék – a második játékos által irányított munkás megkülönböztető színe

játék vége – akkor következik be, ha valamelyik munkás meghal vagy ha lejár az idő, az a munkás nyer, aki életben maradt, vagy ha mindketten élnek akkor az, aki több pontot szerzett

láda – munkások által tolással mozgatható objektum, a cél, hogy a ládát a rakodóterületre juttassuk

meghal – egy munkás akkor hal meg ha nyitott csapdán tartózkodik vagy ha egy másik munkás egy ládával falnak vagy oszlopnak nyomja

oszlop – falként viselkedő egységnyi terület azonban a pályán bárhol lehet, ahol nincs csapda vagy rakodóterület

piros – az első játékos által irányított munkás megkülönböztető színe

pont – egy adott munkás teljesítményét jellemző szám, ha egy munkás egy ládát egy rakodóterületre tol akkor eggyel növeli ezt a számot

(rakodó)munkás – a játékosok által irányított szereplők

rakodóterület – az az egységnyi hely, ahova egy ládát eljuttatva egy munkás pontot kap

raktár – a játék pályája

## Projekt terv

### 2.6.1 Csapat

Csapatunk 5 főből áll, a felelősségeket a következőképp osztottuk ki a tagok közt, annak érdekében, hogy mindenki nagyjából egyformán legyen terhelve:

|  |  |
| --- | --- |
| **Név** | **Felelősségek** |
| Pünkösd Marcell | Kód és dokumentáció |
| Torma Kristóf | Menedzsment és kód |
| Tóth Dániel Péter | Kód és dokumentáció |
| Tóth Vince | Kód és dokumentáció |
| Veres Csaba | UML és dokumentáció |

### 2.6.2 Kommunikáció

**Telegram** A csapatnak létrehoztunk egy supergroup-ot telegrammon, ahol sürgős csapatot érintő dolgokat azonnal el tudjuk intézni. Továbbra ezt használjuk a kérdéses ügyek megbeszélésére is, amely nem igényel személyes találkozót.

**Személyes megbeszélések** Eseti alkalmakkor, ha úgy érezzük, szükséges valamit közösen megbeszélni, akkor előre egyeztetünk időpontot a találkozásra. Ezen kívül minden szerdán 8:15-kor találkozunk a konzultáción.

### 2.6.4 Csapatmunkát támogató eszközök

**Verziókezelés** A verziókezeléshez a Git nevű verziókezelő rendszert használjuk. Ezzel könnyebbé téve a csapatnak, hogy ugyanazon a kódbázison dolgozzunk mindannyian. A kódunkat a GitHub központi tárhelyén tároljuk. Ennek segítségével könnyen követhetjük, hogy ki mivel foglalkozott, és hogy kinek a hibája, ha valami nem működik. :)

**Időkövetés** Írtunk python-ban egy telegram botot, amellyel könnyen követni lehet, hogy ki mikor mennyit dolgozott a projekten. Illetve erről kiértékelést is tudunk készíteni. Ez többek közt segíti a napló megírását.

**Tesztelés** A tesztelés automatizálására a Jenkins nevű szoftvert használjuk, ami automatikusan lefuttatja a tesztelést, és erről részletes jelentést készít. Onnan tudjuk, hogy rendben van a szoftver, hogy minden “zöld”.

### 

### 2.6.3. Használt programok

**Verziókezelés** A verziókezeléshez a fent említett okokból a Git programot használjuk.

**Dokumentáció** A Dokumentációt a Google Docs szöveg szerkesztőjével írjuk, a benne megtalálható UML diagrammok pedig WhiteStarUML-ben készülnek. Azért a Google Docs-ra esett a választás, mert könnyű benne több embernek ugyanazon a dokumentumon dolgozni, és egyből látjuk a mások változtatásait.

**Fejlesztő környezet** A forráskódotEclipse IDE-ben készítjük.

### 2.6.4. Mérföldkövek és határidők

|  |  |  |
| --- | --- | --- |
| **Dátum** | **Leírás** | **Ellenőrzés** |
| 2018. 02. 19. | Követelmény, projekt, funkcionalitás | beadás |
| 2018. 02. 26. | Analízis modell kidolgozása 1. | beadás |
| 2018. 03. 05. | Analízis modell kidolgozása 2. | beadás |
| 2018. 03. 12. | Szkeleton tervezése | beadás |
| 2018. 03. 19. | Szkeleton | beadás |
| 2018. 03. 21. | **Szkeleton** | bemutatás |
| 2018. 03. 26. | Prototípus koncepciója | beadás |
| 2018. 04. 09. | Részletes tervek | beadás |
| 2018. 04. 23. | Prototípus | beadás |
| 2018. 04. 25. | **Prototípus** | bemutatás |
| 2018. 05. 02. | Grafikus felület specifikációja | beadás |
| 2018. 05. 14. | Grafikus változat | beadás |
| 2018. 05. 16. | **Grafikus változat** | bemutatás |
| 2018. 05. 18. | Összefoglalás | beadás |

A feladatot 3 lépcsőben oldjuk meg, a lépcsőket vastagon szedve jelöltük. Ezek bővebben kifejtve:

A **szkeleton** változat célja annak bizonyítása, hogy az objektum és dinamikus modellek a definiált feladat egy modelljét alkotják. A szkeleton egy program, amelyben már valamennyi, a végső rendszerben is szereplő business objektum szerepel. Az objektumoknak csak az interfésze definiált. Valamennyi metódus az indulás pillanatában az ernyőre szöveges változatban kiírja a saját nevét, majd meghívja azon metódusokat, amelyeket a szolgáltatás végrehajtása érdekében meg kell hívnia. Amennyiben a metódusból valamely feltétel fennállása esetén hívunk meg más metódusokat, akkor a feltételre vonatkozó kérdést interaktívan az ernyőn fel kell tenni és a kapott válasz alapján kell a továbbiakban eljárni. A szkeletonnak alkalmasnak kell lenni arra, hogy a különböző forgatókönyvek és szekvencia diagramok ellenőrizhetők legyenek. Csak karakteres ernyőkezelés fogadható el, mert ez biztosítja a rendszer egyszerűségét.  
  
A **prototípus** program célja annak demonstrálása, hogy a program elkészült, helyesen működik, valamennyi feladatát teljesíti. A prototípus változat egy elkészült program kivéve a kifejlett grafikus interfészt. Ez a program is parancssorból futtatható és karakteres ernyőkezelést alkalmaz. A változat tervezési szempontból elkészült, az ütemezés, az aktív objektumok kezelése megoldott. A business objektumok - a megjelenítésre vonatkozó részeket kivéve - valamennyi metódusa a végleges algoritmusokat tartalmazza. A megjelenítés és működtetés egy alfanumerikus ernyőn vezérelhető és követhető, ugyanakkor a vezérlés fájlból is történhet és a megjelenítés fájlba is logolható, ezzel megteremtve a rendszer tesztelésének lehetőségét. Különös figyelmet kell fordítani a parancssori interfész logikájára, felépítésére, valamint arra, hogy az mennyiben tükrözi és teszi láthatóvá a program működését, a beavatkozások hatásait.  
  
A teljes ( **grafikus** ) változat a prototípustól elvileg csak a kezelői felület minőségében különbözhet. Ennek változatnak az értékelésekor a hangsúlyt sokkal inkább a megvalósítás belső szerkezetére, semmint a külalakra kell helyezni.

## Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.02.12 19:00 | 0,5 óra | Pünkösd  Tóth V  Tóth D  Veres  Torma | Értekezlet.  Döntés: Pünkösd elkészíti a 2.6-os részt, Tóth V elkészíti a 2.5-ös részt, Tóth D elkészíti a 2.4es részt, Veres elkészíti a 2.3-as részt, Torma elkészíti a 2.1 és 2.2-es részeket |
| 2018.02.12. 20:00 | 2 óra | Torma | Tevékenység: Torma elkészítette a 2.1-es és 2.2-es részeket |
| 2018.02.13. 17:00 | 2 óra | Veres | Tevékenység: Veres elkészítette a 2.3-as részt. |
| 2018.02.14. 21:00 | 2 óra | Pünkösd | Tevékenység: Pünkösd elkészítette a 2.6-os részt. |
| 2018.02.15. 21:00 | 2 óra | Tóth V | Tevékenység: Tóth V elkészítette a 2.5-ös részt. |
| 2018.02.16. 21:00 | 2 óra | Tóth D | Tevékenység: Tóth D elkészítette a 2.4-es részt. |

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

Doboz, amit a munkás tol. Ha kapcsolón van, akkor aktiválja azt. Ha a helyére kerül, akkor eltűnik. Eltolhat más dobozokat és munkást.

* + 1. **Direction**

A játékban lévő irányokat reprezentáló enumeráció.

* + 1. **EndTile**

Rakódóterület. Az a típusú Tile ahova egy ládát eljuttatva a játékos pontot kap.

* + 1. **Game**

Az játékot mint egységet megvalósító osztály, tartalmazza a játszható pályákat.

* + 1. **Hole**

A lyuk egy olyan pálya elem aminek 2 állapota van: nyitott és zárt, amennyiben nyitott és játékos vagy doboz kerül rá akkor azt eltünteti. Lyukak állapotát kapcsolók váltják, de vannak olyan eset mikor a lyuk nincs kapcsolóhoz kötve ezáltal állandóan nyitott állapotban van.

* + 1. **Map**

A játék egy pályáját valósítja meg. Tartalmazza a játék összes mezőjét, valamint felel a játékosok pontszámainak és a hátralévő idő nyilván tartásáért.

* + 1. **Switch**

Kapcsoló, amely irányítja a hozzá tartozó lyukat. Ha van rajt doboz, akkor aktív, ha nincs, akkor inaktív.

* + 1. **Thing**

Absztrakt osztály, amely az általános a pályán lévő nem statikus elemet reprezentálja.

* + 1. **Tile**

Ez az objektum reprezentálja a pálya egy mezőjét, amin lehet láda, munkás.

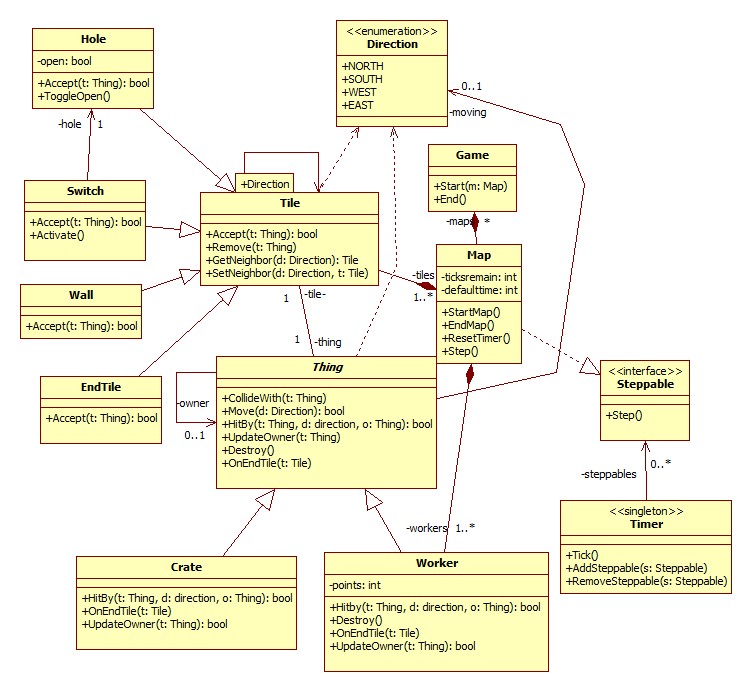
* + 1. **Wall**

Egy olyan pálya elem ami elérthetetlenné teszi azt a mezőt amin elhelyezkedik, kötelezően megjelenik a pálya szélét képző mezőkön, de előfordulhat belső mezőkön is.

* + 1. **Worker**

A munkás a játékos által irányított ‘karakter’, képes lépni, tolni ládákat, beleesni egy lyukba (ekkor meghal).

* 1. **Statikus struktúra diagramok**



* 1. **Osztályok leírása**
     1. **Crate**
        + **Felelősség**

A pályán lévő ládákat reprezentáló osztály.

* + - * **Ősosztályok**

Thing

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**

Nincs.

* + - * **Metódusok**
* **bool HitBy(t: Thing,d: Direction,o: Thing)**: közli egy másik Thinggel hogy nekiment egy másik Thing, és, hogy mozduljon el ha tud.
* **void OnEndTile(t: EndTile)**: Pontot ad a helyretoló munkásnak és eltűnik a pályáról.
* **bool UpdateOwner(newvalue:Thing)**: Beállítja az owner attribútum értékét a newrvalue-ra.
  + 1. **Direction**
       - **Felelősség**

Enumeráció, amelyben a játékban lévő irányok vannak tárolva.

* + - * **Ősosztályok**

Nincs.

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**
* **+NORTH:** Észak
* +**SOUTH**: Dél
* **+WEST:** Nyugat
* +**EAST**: Kelet
  + - * **Metódusok**

Nincs.

* + 1. **EndTile**
       - **Felelősség**

Azt a Tilet reprezentálja, amelyre rá kell tolni a dobozokat.

* + - * **Ősosztályok**

Tile

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**

Nincs.

* + - * **Metódusok**
* **bool Accept(t:Thing):** Ha doboz kerül rá, akkor eltűnik az adott doboz és a munkás, aki rátolta pontot kap. Ha nem doboz kerül rá, akkor sima tileként viselkedik.
  + 1. **Game**
       - **Felelősség**

A játék egészét reprezentáló osztály.

* + - * **Ősosztályok**

Nincs.

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**
* **-maps: Map[\*]:** A játékot alkotó pályák összessége.
  + - * **Metódusok**
* **void Start():** Elindítja a játékot.
* **void End():** Leállítja a játékot.
* **Map[] GetMaps():** Visszaadja a maps attribútum értékét.
* **void SetMaps(newvalue:Map[]):** Beállítja a maps attribútum értékét a newrvalue-ra.
  + 1. **Hole**
       - **Felelősség**

A pályán lévő lyukakat reprezentáló osztály.

* + - * **Ősosztályok**

Tile

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**
* **-open:bool:** Nyitott-e a lyuk. (Ha nem nyitott, akkor sima tileként viselkedik.)
  + - * **Metódusok**
* **bool Accept(t: Thing):** Ha a lyuk nem nyitott, akkor sima tileként viselkedik, ha nyitott, akkor elnyeli a rálépő Thinget.
* **bool GetOpen():** Visszaadja az open attribútum értékét.
* **void ToggleOpen():** Negálja az open attribútum értékét és biztosítja a láda vagy munkás leesését ha éppen kinyitották és rajta állnak.
  + 1. **Map**
       - **Felelősség**

Egy pályát reprezentáló osztály.

* + - * **Ősosztályok**

Nincs.

* + - * **Interfészek**

Steppable

* + - * **Attribútumok**
* **-tiles: Tile [1..\*]:** A pályát alkotó tileok.
* **-workers: Worker [1..\*]:** A pályán lévő munkások.
* **-ticksremain: int**: Visszamaradó idő, tickekben.
* **-defaulttime: int**: Alapértelmezett visszamaradó idő, tickekben.
  + - * **Metódusok**
* **void StartMap():** Inicializálja a pályát.
* **void EndMap():** Miután eltűnt minden Crate, ez a függvény hívódik meg. Elindítja a következő pályát (ha van).
* **Tile[] GetTiles()**: Visszaadja a tiles attribútum értékét.
* **void SetTiles(newvalue:Tile[]):** Beállítja a tiles attribútum értékét a newrvalue-ra.
* **Worker[] GetWorkers():** Visszaadja a workers attribútum értékét.
* **void SetWorkers(newvalue:Worker[]):** Beállítja a workers attribútum értékét a newrvalue-ra.
* **void SetTicksRemain (i: int):** Beállítja a visszamaradó időt.
* **int GetTicksRemain():** Lekérdezi, mennyi idő van hátra.
* **void ResetTimer()**: Visszaállítja a visszamaradó időt az alapértlmezettre.
* **int getDefaultTime()**: Lekérdezi az alapértelmezett visszamaradó időt.
* **void setDefaultTime(t: int)**: Beállítja az alapértelmezett visszamaradó időt.
  + 1. **Steppable**
       - **Felelősség**

Interfész, ami a lépni képes dolgokat reprezentálja a játékban.

* + - * **Ősosztályok**

Nincs

* + - * **Metódusok**
* **void Step():** E függvény meghívásakor lép.
  + 1. **Switch**
       - **Felelősség**

A pályán lévő Hole-okat kapcsoló kapcsolókat reprezentáló osztály.

* + - * **Ősosztályok**

Tile

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**
* **-hole: Hole:** Az a Hole amelyet az adott Switch kapcsol.
  + - * **Metódusok**
* **bool Accept(t:Thing):** Befogadja a rálépő Thinget, mint a rendes Tile, de, ha Crate lépett rá, akkor aktiválja a hozzá tartozó Holet.
* **Hole GetHole():** Visszaadja a Hole attribútum értékét.
* **void SetHole(newvalue:Hole):** Beállítja a Hole attribútum értékét a newrvalue-ra.
* **void Activate():** Megnyomja a kapcsolót.
  + 1. ***Thing***
       - **Felelősség**

Absztrakt osztály, amely az általános a pályán lévő nem statikus elemet reprezentálja.

* + - * **Ősosztályok**

Nincs.

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**
* **-tile:Tile:** Ezen a tileon helyezkedik el a Thing.
* **-owner:Thing:** Ez a Thing mozgatja éppen.
* **-moving: Direction**: Ebbe az irányba mozog a Thing.
  + - * **Metódusok**
* **bool CollideWith(t:Thing)**: Ütközik t-vel és visszatér igaz/hamis értékkel, ami azt jelenti, hogy sikerült-e ütközni.
* **void Move(d: Direction):** ‘d’ irányba mozgatja a Thing-et.
* **bool HitBy(t: Thing,d: Direction,o: Thing)**: közli egy másik Thinggel hogy nekiment egy másik Thing, és, hogy mozduljon el ha tud.
* **void OnEndTile(t: EndTile)**: Közli az adott Thinggel, hogy EndTile-on van.
* **Thing GetOwner()**: Visszaadja az owner attribútumot.
* **bool UpdateOwner(newvalue:Thing)**: Beállítja az owner attribútum értékét a newrvalue-ra.
* **Direction GetMoving()**: Visszaadja a moving attribútum értékét.
* **void SetMoving(newvaluevalue:Direction)**: Beállítja a moving attribútum értékét a newvalue-ra.
* **Tile GetTile()**: Visszaadja a Tile attribútum értékét.
* **void SetTile(newvaluevalue:Tile)**: Beállítja a Tile attribútum értékét a newvalua-ra.
* **void Destroy():** Leveszi magát a pályáról és meghívja a saját destruktorát.
  + 1. **Timer**
       - **Felelősség**

Idózítő osztály, singleton.

* + - * **Ősosztályok**

Nincs.

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**

**-steppables: Steppable [0..\*]:** Timer által léptetendő léptethető objektumok.

* + - * **Metódusok**
* **void Tick():** Egyet “kattan” az óra, léptet minden lépni képes objektumot.
* **void AddSteppable(s: Steppable)**: Hozzáad egy Steppable objektumot a léptetendő objektumok listájára.
* **void RemoveSteppable(s: Steppable)**: Eltávolítja ‘s’-t a léptetendő objektumok listájáról, ha a listában van.
  + 1. **Tile**
       - **Felelősség**

Az általános pályaelemet reprezentálja, amin lehet Thing.

* + - * **Ősosztályok**

Nincs.

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**
* **-nexttiles[Direction]:Tile**: A tileból induló tileok.
* **-thing:Thing**: A tileon lévő thing.
  + - * **Metódusok**
* **bool Accept(m: Thing)** Befogadja a bejövő Thing objektumot. Ha nem üres, akkor ütközteti a rajtalévővel.
* **void Remove(m: Thing)**: eltávolítja az éppen rajta lévő Thinget.
* **Tile GetNeighbor(d: Direction)**: d irányban lévő szomszédos Tile-lal tér vissza, vagy NULL-lal, ha nincs ebben az irányban tile.
* **void SetNeighbor(d: Direction, t: Tile)**: Beállítja t-re d irányban a szomszédját.
  + 1. **Wall**
       - **Felelősség**

A pálya szélén és a pályán lévő falakat reprezentáló osztály.

* + - * **Ősosztályok**

Tile

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**

Nincs.

* + - * **Metódusok**
* **bool Accept(t: Thing):** Mindig false-sal tér vissza, mivel falra menni nem lehet.
  + 1. **Worker**
       - **Felelősség**

A játékos által irányított munkást reprezentáló osztály.

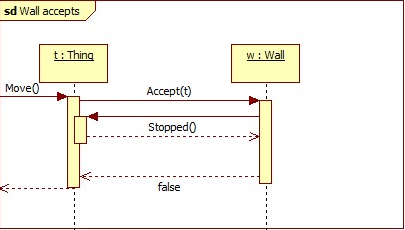
* + - * **Ősosztályok**

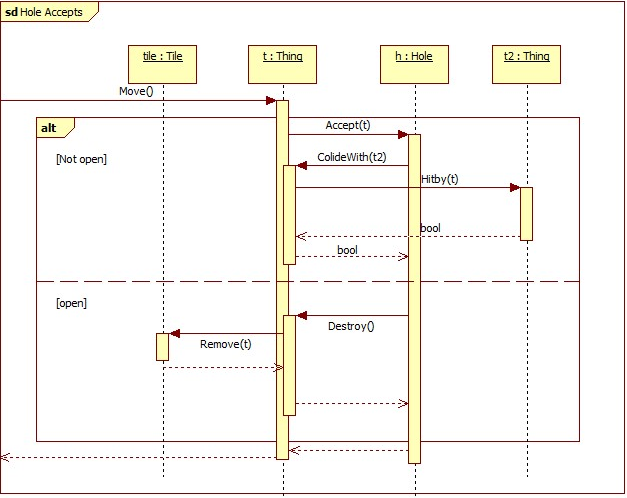
Thing

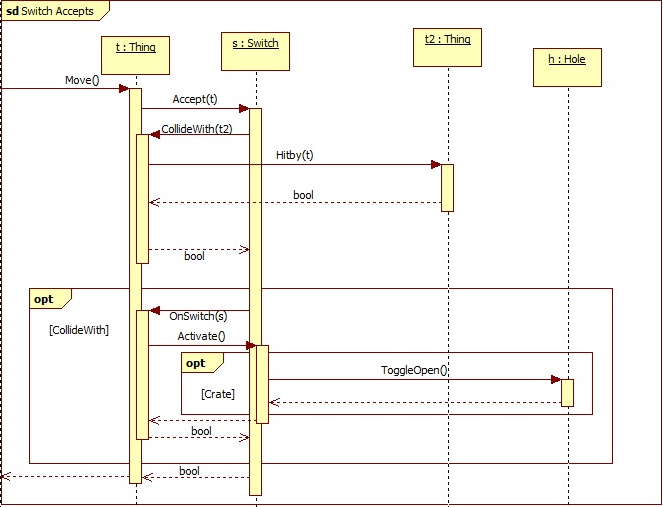
* + - * **Interfészek**

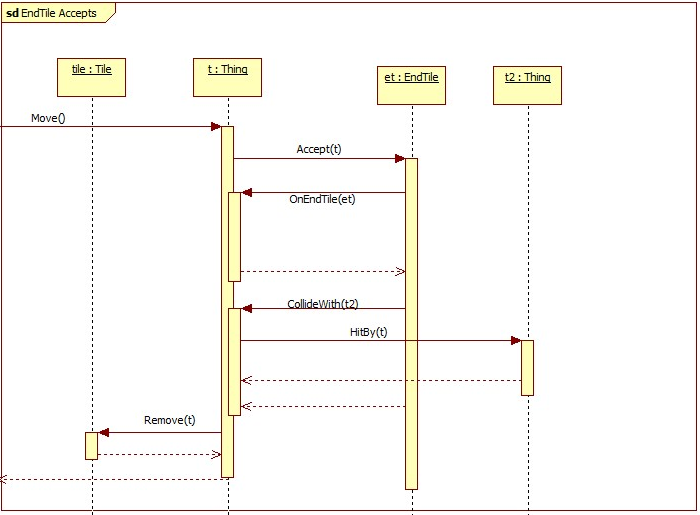
Nincs.

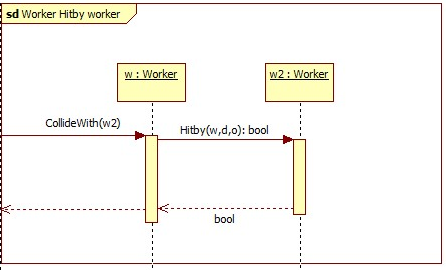
* + - * **Attribútumok**
* **-points: int:** A munkás pontjainak száma.
  + - * **Metódusok**
* **bool HitBy(t: Thing)**: metódus leírása
* **void OnEndTile(t: EndTile)**: Nem csinál semmit.
* **int GetPoints()**: Visszaadja a points attribútum értékét.
* **void SetPoints(newvalue:int)**: Beállítja a points attribútum értékét a newvalue-ra.
* **bool UpdateOwner(newvalue:Thing)**: Beállítja az owner attribútum értékét a newrvalue-ra.
* **void Destroy():** Leveszi magát a pályáról és meghívja a saját destruktorát.
  1. **Szekvencia diagramok**

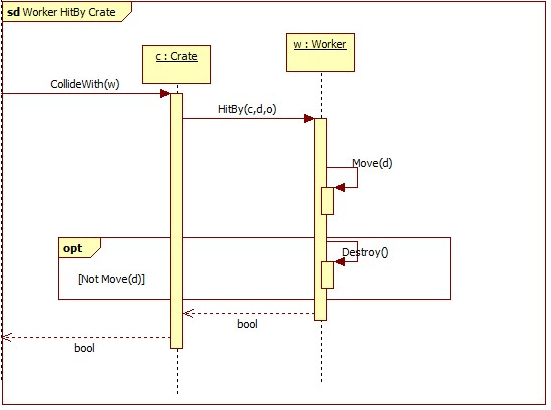
****

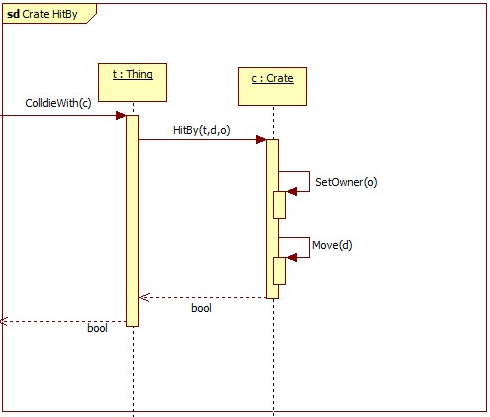
****

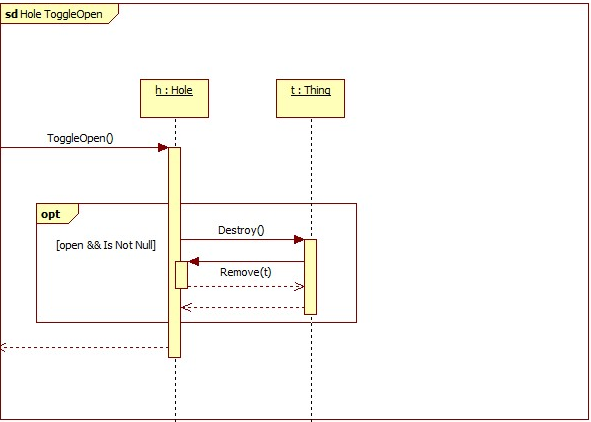
****

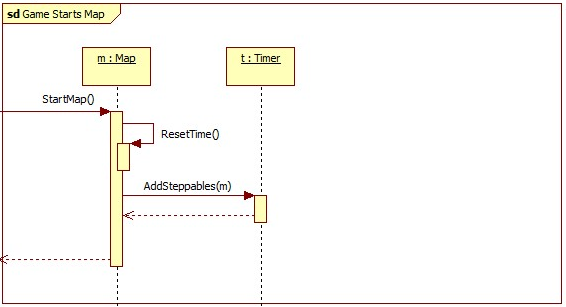
****

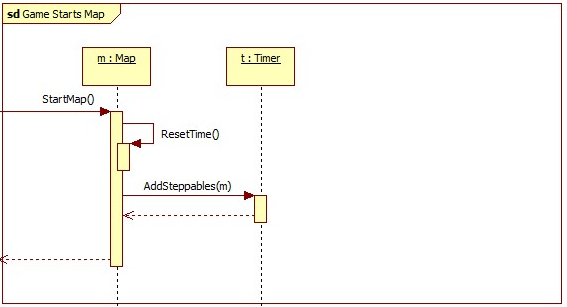
****

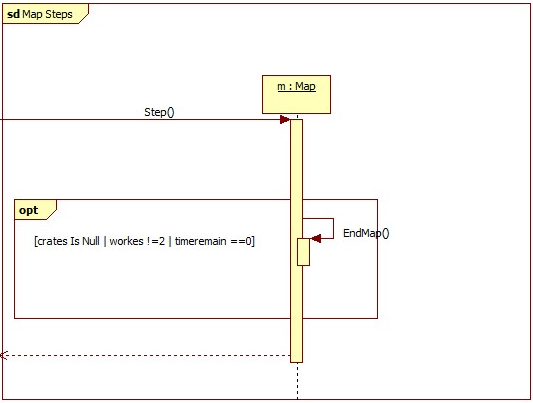
****

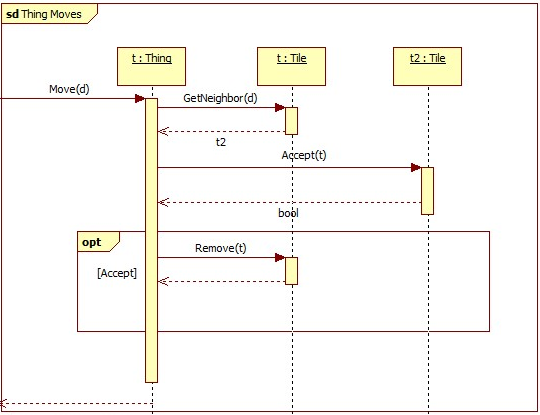
****

****

****

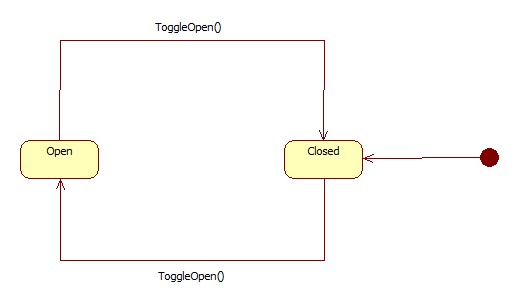
****

****

****

* 1. **State-chartok**

**Hole:**

****

**Napló**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2017.02.19. 18:00 | 2 óra | Torma | Tevékenység: Dokumentum előkészítése, osztály diagram skeletonjának elkészítése. |
| 2017.02.20. 20:00 | 2 óra | Pünkösd | Tevékenység: Osztály diagram véglegesítése. |
| 2017.02.21. 22:00 | 2 óra | Tóth Vince | Tevékenység: Szekvencia diagrammok elkészítése, osztály diagram adjusztálása. |
| 2017.02.22. 15:00 | 2 óra | Tóth D | Tevékenység: Szekvencia diagrammok adjusztálása. |
| 2017.02.23. 19:00 | 2 óra | Veres | Tevékenység: Dokumentum elkészítése, osztály és szekvencia diagrammok adjusztálása. |

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

Doboz, amit a munkás tol. Ha kapcsolón van, akkor aktiválja azt. Ha a helyére kerül, akkor eltűnik. Eltolhat más dobozokat és munkást.

* + 1. **EndTile**

Rakódóterület. Az a típusú Tile ahova egy ládát eljuttatva a játékos pontot kap.

* + 1. **Hole**

A lyuk egy olyan pálya elem aminek 2 állapota van: nyitott és zárt, amennyiben nyitott és játékos vagy doboz kerül rá akkor azt eltünteti. Lyukak állapotát kapcsolók váltják, de vannak olyan eset mikor a lyuk nincs kapcsolóhoz kötve ezáltal állandóan nyitott állapotban van.

* + 1. **Map**

A játék egy pályáját valósítja meg. Tartalmazza a játék összes mezőjét, valamint felel a játékosok pontszámainak és a hátralévő idő nyilván tartásáért.

* + 1. **Switch**

Kapcsoló, amely irányítja a hozzá tartozó lyukat. Ha van rajt doboz, akkor aktív, ha nincs, akkor inaktív.

* + 1. **Thing**

Absztrakt osztály, amely az általános a pályán lévő nem statikus elemet reprezentálja.

* + 1. **Tile**

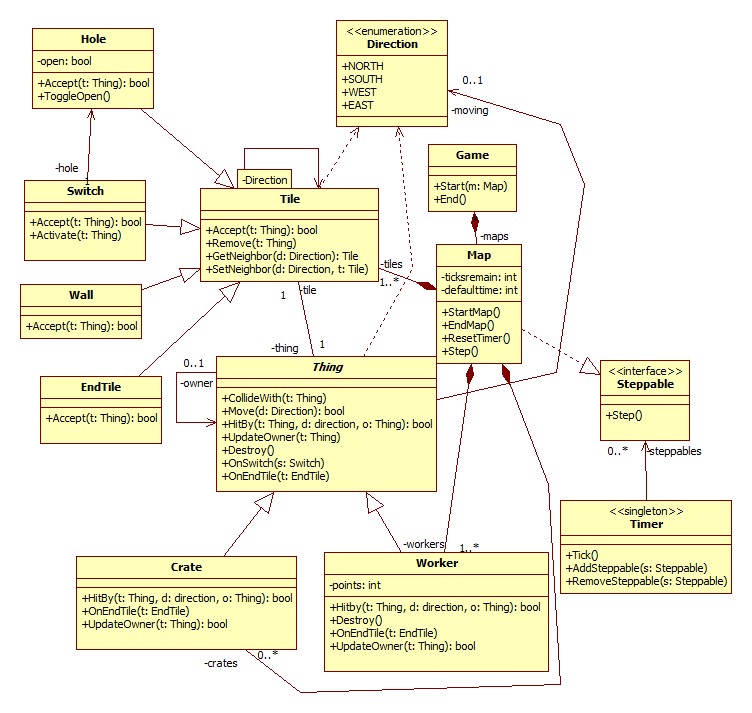
Ez az objektum reprezentálja a pálya egy mezőjét, amin lehet láda, munkás.

* + 1. **Wall**

Egy olyan pálya elem ami elérthetetlenné teszi azt a mezőt amin elhelyezkedik, kötelezően megjelenik a pálya szélét képző mezőkön, de előfordulhat belső mezőkön is.

* + 1. **Worker**

A munkás a játékos által irányított ‘karakter’, képes lépni, tolni ládákat, beleesni egy lyukba (ekkor meghal).

* 1. **Statikus struktúra diagramok**
  2. **Osztályok leírása**
     1. **Crate**
        + **Felelősség**

A pályán lévő ládákat reprezentáló osztály.

* + - * **Ősosztályok**

Thing

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**

Nincs.

* + - * **Metódusok**
* **bool HitBy(t: Thing,d: Direction,o: Thing)**: közli egy másik Thinggel hogy nekiment egy másik Thing, és, hogy mozduljon el ha tud.
* **void OnEndTile(t: EndTile)**: Pontot ad a helyretoló munkásnak és eltűnik a pályáról.
* **bool UpdateOwner(newvalue:Thing)**: Beállítja az owner attribútum értékét a newrvalue-ra.
  + 1. **Direction**
       - **Felelősség**

Enumeráció, amelyben a játékban lévő irányok vannak tárolva.

* + - * **Ősosztályok**

Nincs.

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**
* **+NORTH:** Észak
* +**SOUTH**: Dél
* **+WEST:** Nyugat
* +**EAST**: Kelet
  + - * **Metódusok**

Nincs.

* + 1. **EndTile**
       - **Felelősség**

Azt a Tilet reprezentálja, amelyre rá kell tolni a dobozokat.

* + - * **Ősosztályok**

Tile

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**

Nincs.

* + - * **Metódusok**
* **bool Accept(t:Thing):** Ha doboz kerül rá, akkor eltűnik az adott doboz és a munkás, aki rátolta pontot kap. Ha nem doboz kerül rá, akkor sima tileként viselkedik.
  + 1. **Game**
       - **Felelősség**

A játék egészét reprezentáló osztály.

* + - * **Ősosztályok**

Nincs.

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**
* **-maps: Map[\*]:** A játékot alkotó pályák összessége.
  + - * **Metódusok**
* **void Start():** Elindítja a játékot.
* **void End():** Leállítja a játékot.
* **Map[] GetMaps():** Visszaadja a maps attribútum értékét.
* **void SetMaps(newvalue:Map[]):** Beállítja a maps attribútum értékét a newrvalue-ra.
  + 1. **Hole**
       - **Felelősség**

A pályán lévő lyukakat reprezentáló osztály.

* + - * **Ősosztályok**

Tile

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**
* **-open:bool:** Nyitott-e a lyuk. (Ha nem nyitott, akkor sima tileként viselkedik.)
  + - * **Metódusok**
* **bool Accept(t: Thing):** Ha a lyuk nem nyitott, akkor sima tileként viselkedik, ha nyitott, akkor elnyeli a rálépő Thinget.
* **bool GetOpen():** Visszaadja az open attribútum értékét.
* **void ToggleOpen():** Negálja az open attribútum értékét és biztosítja a láda vagy munkás leesését ha éppen kinyitották és rajta állnak.
  + 1. **Map**
       - **Felelősség**

Egy pályát reprezentáló osztály.

* + - * **Ősosztályok**

Nincs.

* + - * **Interfészek**

Steppable

* + - * **Attribútumok**
* **-tiles: Tile [1..\*]:** A pályát alkotó tileok.
* **-workers: Worker [1..\*]:** A pályán lévő munkások.
* **-ticksremain: int**: Visszamaradó idő, tickekben.
* **-defaulttime: int**: Alapértelmezett visszamaradó idő, tickekben.
  + - * **Metódusok**
* **void StartMap():** Inicializálja a pályát.
* **void EndMap():** Miután eltűnt minden Crate, ez a függvény hívódik meg. Elindítja a következő pályát (ha van).
* **Tile[] GetTiles()**: Visszaadja a tiles attribútum értékét.
* **void SetTiles(newvalue:Tile[]):** Beállítja a tiles attribútum értékét a newrvalue-ra.
* **Worker[] GetWorkers():** Visszaadja a workers attribútum értékét.
* **void SetWorkers(newvalue:Worker[]):** Beállítja a workers attribútum értékét a newrvalue-ra.
* **void SetTicksRemain (i: int):** Beállítja a visszamaradó időt.
* **int GetTicksRemain():** Lekérdezi, mennyi idő van hátra.
* **void ResetTimer()**: Visszaállítja a visszamaradó időt az alapértlmezettre.
* **int getDefaultTime()**: Lekérdezi az alapértelmezett visszamaradó időt.
* **void setDefaultTime(t: int)**: Beállítja az alapértelmezett visszamaradó időt.
  + 1. **Steppable**
       - **Felelősség**

Interfész, ami a lépni képes dolgokat reprezentálja a játékban.

* + - * **Ősosztályok**

Nincs

* + - * **Metódusok**
* **void Step():** E függvény meghívásakor lép.
  + 1. **Switch**
       - **Felelősség**

A pályán lévő Hole-okat kapcsoló kapcsolókat reprezentáló osztály.

* + - * **Ősosztályok**

Tile

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**
* **-hole: Hole:** Az a Hole amelyet az adott Switch kapcsol.
  + - * **Metódusok**
* **bool Accept(t:Thing):** Befogadja a rálépő Thinget, mint a rendes Tile.
* **Hole GetHole():** Visszaadja a Hole attribútum értékét.
* **void SetHole(newvalue:Hole):** Beállítja a Hole attribútum értékét a newrvalue-ra.
* **void Activate(t: Thing):** Megnyomja a kapcsolót ha Crate érkezik rá.
  + 1. ***Thing***
       - **Felelősség**

Absztrakt osztály, amely az általános a pályán lévő nem statikus elemet reprezentálja.

* + - * **Ősosztályok**

Nincs.

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**
* **-tile:Tile:** Ezen a tileon helyezkedik el a Thing.
* **-owner:Thing:** Ez a Thing mozgatja éppen.
* **-moving: Direction**: Ebbe az irányba mozog a Thing.
  + - * **Metódusok**
* **bool CollideWith(t:Thing)**: Ütközik t-vel és visszatér igaz/hamis értékkel, ami azt jelenti, hogy sikerült-e ütközni.
* **void Move(d: Direction):** ‘d’ irányba mozgatja a Thing-et.
* **bool HitBy(t: Thing,d: Direction,o: Thing)**: közli egy másik Thinggel hogy nekiment egy másik Thing, és, hogy mozduljon el ha tud.
* **void OnEndTile(t: EndTile)**: Közli az adott Thinggel, hogy EndTile-on van.
* void **OnSwitch(s: Switch)**:Közli az adott Thinggel hogy Switch-en van.
* **Thing GetOwner()**: Visszaadja az owner attribútumot.
* **bool UpdateOwner(newvalue:Thing)**: Beállítja az owner attribútum értékét a newrvalue-ra.
* **Direction GetMoving()**: Visszaadja a moving attribútum értékét.
* **void SetMoving(newvaluevalue:Direction)**: Beállítja a moving attribútum értékét a newvalue-ra.
* **Tile GetTile()**: Visszaadja a Tile attribútum értékét.
* **void SetTile(newvaluevalue:Tile)**: Beállítja a Tile attribútum értékét a newvalua-ra.
* **void Destroy():** Leveszi magát a pályáról és meghívja a saját destruktorát.
  + 1. **Timer**
       - **Felelősség**

Idózítő osztály, singleton.

* + - * **Ősosztályok**

Nincs.

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**

**-steppables: Steppable [0..\*]:** Timer által léptetendő léptethető objektumok.

* + - * **Metódusok**
* **void Tick():** Egyet “kattan” az óra, léptet minden lépni képes objektumot.
* **void AddSteppable(s: Steppable)**: Hozzáad egy Steppable objektumot a léptetendő objektumok listájára.
* **void RemoveSteppable(s: Steppable)**: Eltávolítja ‘s’-t a léptetendő objektumok listájáról, ha a listában van.
  + 1. **Tile**
       - **Felelősség**

Az általános pályaelemet reprezentálja, amin lehet Thing.

* + - * **Ősosztályok**

Nincs.

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**
* **-nexttiles[Direction]:Tile**: A tileból induló tileok.
* **-thing:Thing**: A tileon lévő thing.
  + - * **Metódusok**
* **bool Accept(m: Thing)** Befogadja a bejövő Thing objektumot. Ha nem üres, akkor ütközteti a rajtalévővel.
* **void Remove(m: Thing)**: eltávolítja az éppen rajta lévő Thinget.
* **Tile GetNeighbor(d: Direction)**: d irányban lévő szomszédos Tile-lal tér vissza, vagy NULL-lal, ha nincs ebben az irányban tile.
* **void SetNeighbor(d: Direction, t: Tile)**: Beállítja t-re d irányban a szomszédját.
  + 1. **Wall**
       - **Felelősség**

A pálya szélén és a pályán lévő falakat reprezentáló osztály.

* + - * **Ősosztályok**

Tile

* + - * **Interfészek**

Nincs.

* + - * **Attribútumok**

Nincs.

* + - * **Metódusok**
* **bool Accept(t: Thing):** Mindig false-sal tér vissza, mivel falra menni nem lehet.
  + 1. **Worker**
       - **Felelősség**

A játékos által irányított munkást reprezentáló osztály.

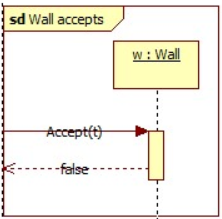
* + - * **Ősosztályok**

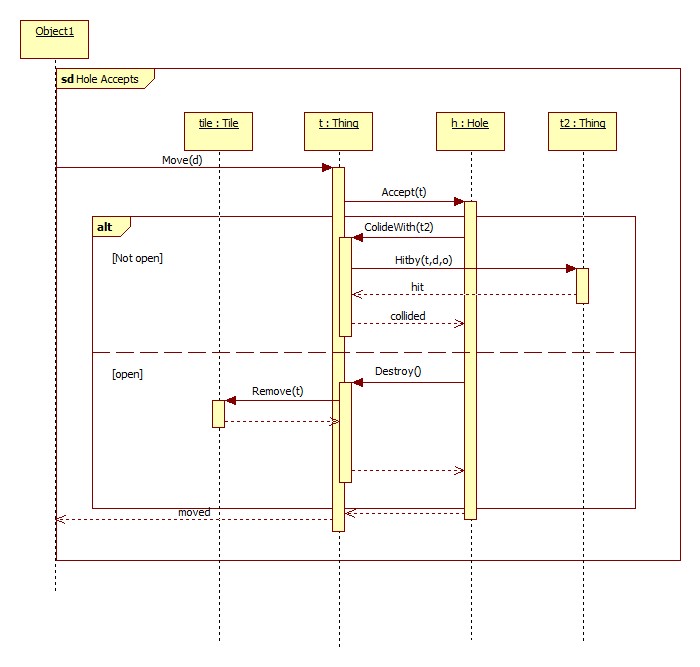
Thing

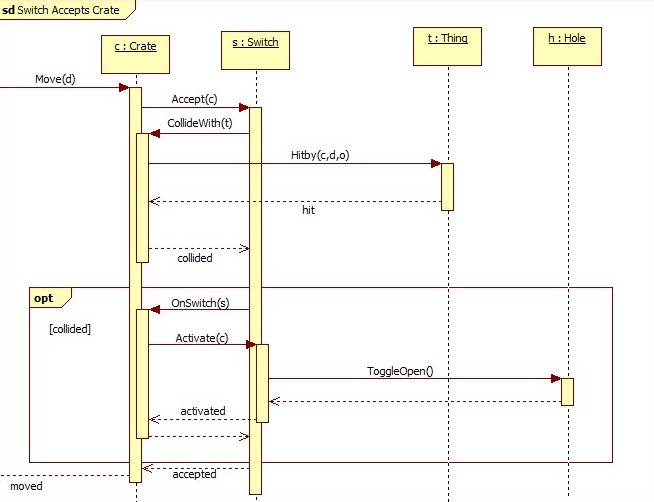
* + - * **Interfészek**

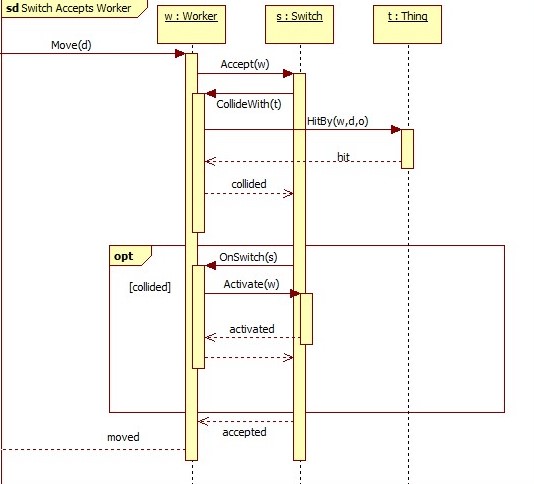
Nincs.

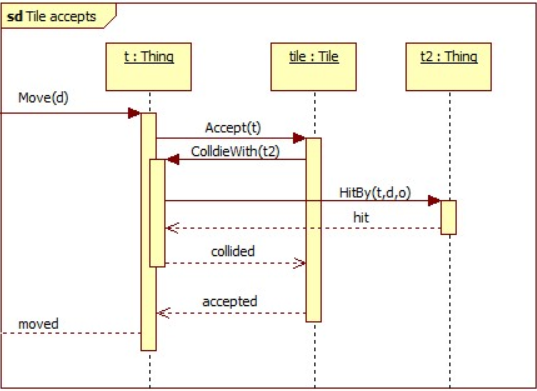
* + - * **Attribútumok**
* **-points: int:** A munkás pontjainak száma.
  + - * **Metódusok**
* **bool HitBy(t: Thing)**: metódus leírása
* **void OnEndTile(t: EndTile)**: Nem csinál semmit.
* **int GetPoints()**: Visszaadja a points attribútum értékét.
* **void SetPoints(newvalue:int)**: Beállítja a points attribútum értékét a newvalue-ra.
* **bool UpdateOwner(newvalue:Thing)**: Beállítja az owner attribútum értékét a newrvalue-ra.
* **void Destroy():** Leveszi magát a pályáról és meghívja a saját destruktorát.
  1. **Szekvencia diagramok**

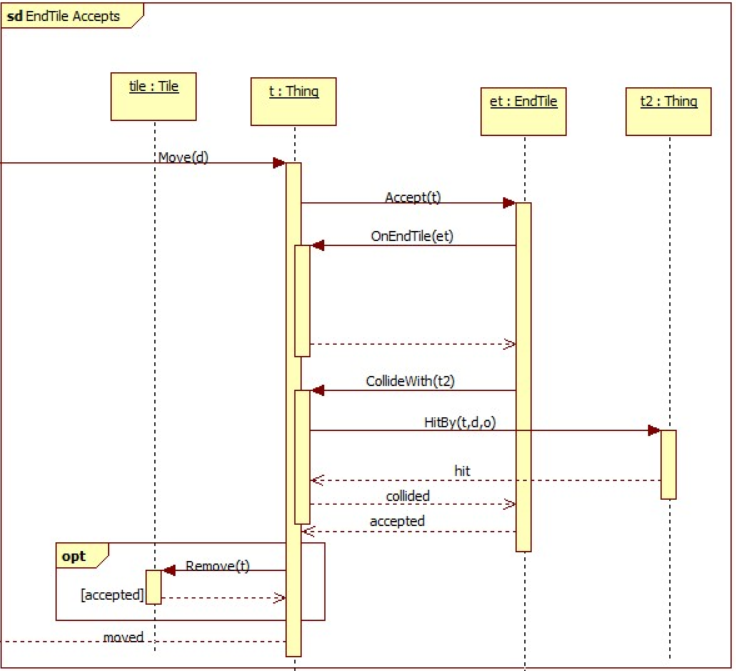
****

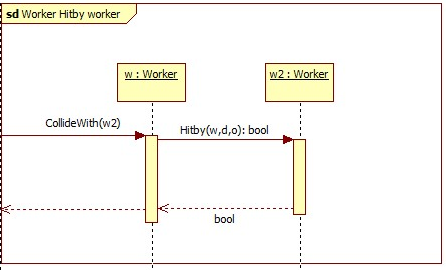
****

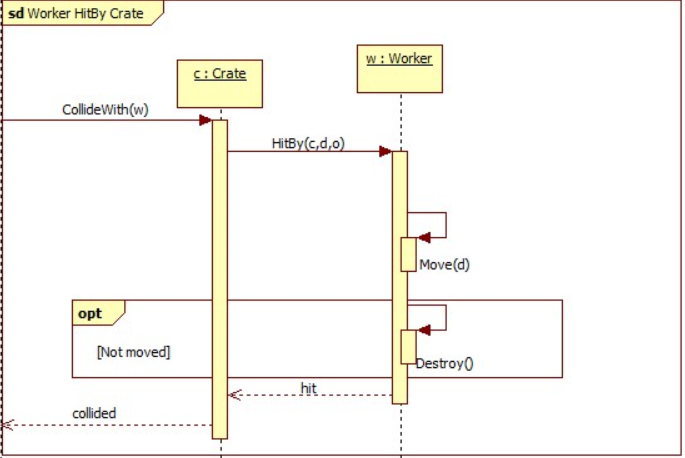
****

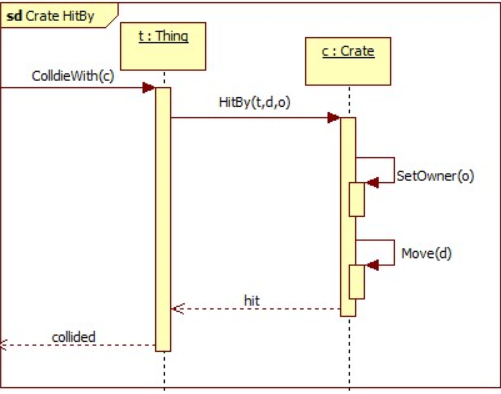
****

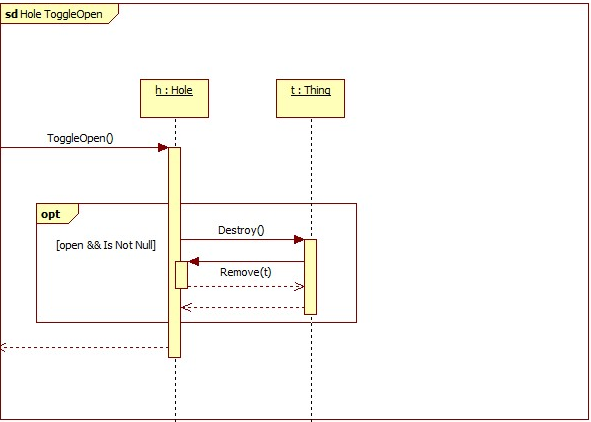
****

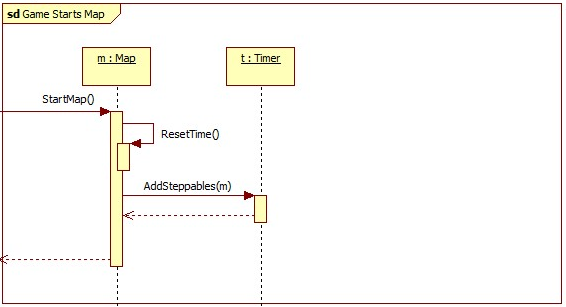
****

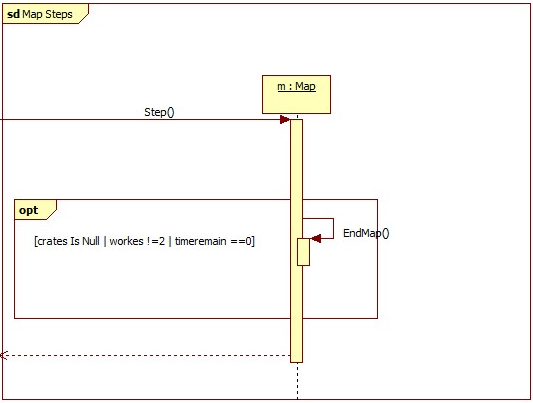
****

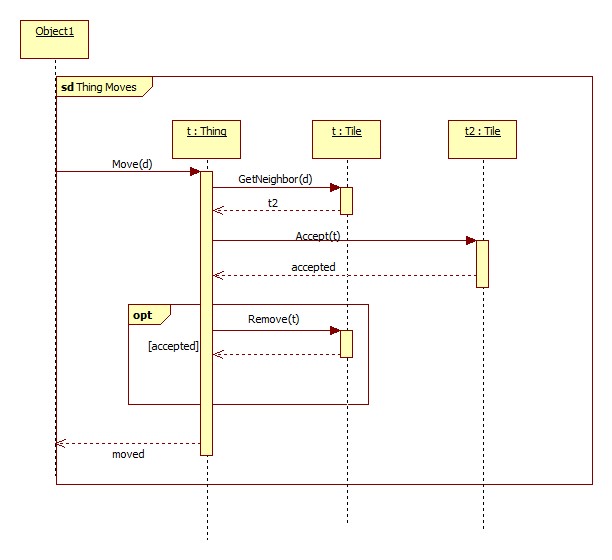
****

****

****

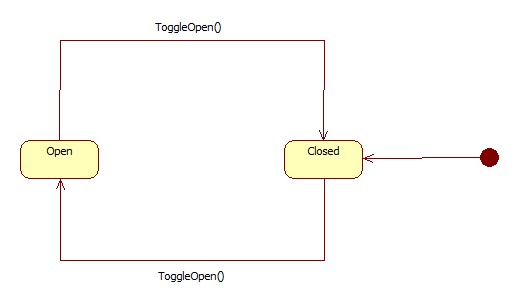
****

****

****

* 1. **State-chartok**

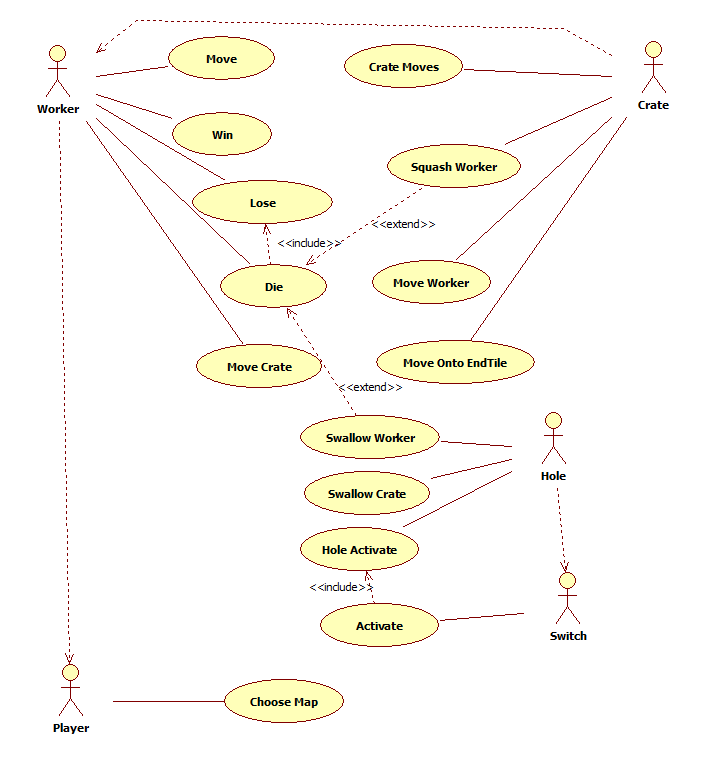
**Hole:**

****

**Napló**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2017.03.01. 17:00 | 0,5 óra | Torma  Pünkösd  Tóth Vince  Tóth D  Veres | Gyűlés: Változások megbeszélése. Döntés: Szekvencia diagrammok esztétikai javítása, felmerült funkcionális javítások elkészítése. |
| 2017.03.4. 20:00 | 0.5 óra | Pünkösd | Tevékenység: Osztály diagram javítása. |
| 2017.03.2. 22:00 | 0.5 óra | Tóth Vince | Tevékenység: Szekvencia diagrammok logikai javítása. |
| 2017.03.22. 15:00 | 0.5 óra | Tóth D | Tevékenység: Szekvencia diagrammok esztétikai javítása. |
| 2017.03.23. 19:00 | 0.5 óra | Veres | Tevékenység: Dokumentum előkészítése, apró szekvencia diagram hibák javítása. |
| 2017.03.5. 00:00 | 0.5 óra | Torma | Tevékenység: Dokumentum véglegesítése, lektorálás, nyomtatás. |

1. **Szkeleton tervezése**
   1. ***A szkeleton modell valóságos use-case-ei***
2. *Worker üres mezőre mozog*
3. *Crate eltol másik Cratet*
4. *Crate eltol egy Worker-t*
5. *Worker falra / oszlopra próbál lépni*
6. *Worker egy másik Workernek ütközik*
7. *Worker áttol egy Cratet üres mezőre*
8. *Worker Crate-t falnak / oszlopnak tol*
9. *Worker nyitott Hole-ra lép*
10. *Crate Cratet tol egy Workerre*
11. *Crate nyitott Hole-ra kerül*
12. *Crate aktiválja a kapcsolót*
13. *Worker Cratet tol egy másik Workerre*
14. *Crate EndTile-ra mozog*
15. *Switch aktivál egy Holet*
16. *Switch deaktivál egy Holet*
    * 1. **Use-case diagram**



* + 1. **Use-case leírások**

|  |  |
| --- | --- |
| **Use-case neve** | Worker Wins |
| **Rövid leírás** | Egy Worker nyer. |
| **Aktorok** | Worker |
| **Forgatókönyv** | Egy Worker nyer, mert a pálya végén neki van a legtöbb pontja, vagy Ő az egyetlen életben maradt Worker. |

|  |  |
| --- | --- |
| **Use-case neve** | Worker Moves |
| **Rövid leírás** | Egy worker mozog a pályán. |
| **Aktorok** | Worker |
| **Forgatókönyv** | Egy Worker lép egy adott irányba, mert a felhasználó irányítja, vagy egy Crate tolja, vagy nem tud mozogni, mert nincs helye az adott irányban. (Vagy Ő, vagy, amit tol falnak ütközik.) |

|  |  |
| --- | --- |
| **Use-case neve** | Worker Loses |
| **Rövid leírás** | Egy Worker veszít. |
| **Aktorok** | Worker |
| **Forgatókönyv** | Egy Worker veszít, mert meghal, vagy a pálya végén nem neki van a legtöbb pontja. |

|  |  |
| --- | --- |
| **Use-case neve** | Worker Moves Crate |
| **Rövid leírás** | Worker mozgatja a Cratet. |
| **Aktorok** | Worker |
| **Forgatókönyv** | Egy Worker mozgat egy Cratet valamilyen irányba, mert a játékos irányítja, vagy Őt is tolja valaki, vagy nem tudja mozgatni, mert nincs hely az adott irányba. (Vagy Ő, vagy, amit tol falnak ütközik.) |

|  |  |
| --- | --- |
| **Use-case neve** | Player Choses Map |
| **Rövid leírás** | A Játékos kiválasztja a pályát, amin játszani szeretne. |
| **Aktorok** | Player |
| **Forgatókönyv** | A program felkínál egy listát a választható pályákról, amik közül a játékos választ. |

|  |  |
| --- | --- |
| **Use-case neve** | Crate Moves |
| **Rövid leírás** | Crate mozgatódik a pályán. |
| **Aktorok** | Crate |
| **Forgatókönyv** | Egy Crate mozgatva van a pályán, esetlegesen Ő is tolhat valakit, ha van valami az útjában, vagy nem mozdul, mert az adott irányba nincs hely. (Vagy Ő, vagy, amit tol falnak ütközik.) |

|  |  |
| --- | --- |
| **Use-case neve** | Crate Ativates Switch |
| **Rövid leírás** | Crate rámegy egy Switch-re és aktiválja. |
| **Aktorok** | Crate |
| **Forgatókönyv** | Egy Crate rámegy egy Switchre, ami ezáltal aktivál egy Hole-t. |

|  |  |
| --- | --- |
| **Use-case neve** | Crate Moves Worker |
| **Rövid leírás** | Crate ellök egy Worker-t. |
| **Aktorok** | Crate |
| **Forgatókönyv** | Egy Crate eltol egy Workert az adott irányba. |

|  |  |
| --- | --- |
| **Use-case neve** | Crate Moves Crate |
| **Rövid leírás** | Crate ellök egy másik Crate-et. |
| **Aktorok** | Crate |
| **Forgatókönyv** | Egy Crate eltol egy másik Crate-et az adott irányba. |

|  |  |
| --- | --- |
| **Use-case neve** | Switch Toggles Hole |
| **Rövid leírás** | Switch kapcsol egy Hple-t. |
| **Aktorok** | Switch |
| **Forgatókönyv** | A Switch állapota megváltozott, ezért a Hole állapotát is változtatja. (Lenyomott állapotba lépéskor nyitja, csukott állapotba lépéskor csukja.) |

|  |  |
| --- | --- |
| **Use-case neve** | Hole Swallows Worker |
| **Rövid leírás** | Hole elnyel egy Worker-t. |
| **Aktorok** | Hole |
| **Forgatókönyv** | Hole-ba, ami nyitva van beleesik egy Worker, emiatt meghal. |

|  |  |
| --- | --- |
| **Use-case neve** | Hole Activates |
| **Rövid leírás** | Hole aktiválódik. |
| **Aktorok** | Hole |
| **Forgatókönyv** | A hozzá tartozó Switch aktiválja a Hole-t és belesik ami rajta van. |

|  |  |
| --- | --- |
| **Use-case neve** | Crate Moves Onto EndTile |
| **Rövid leírás** | Crate rámegy egy EndTile-ra. |
| **Aktorok** | Crate |
| **Forgatókönyv** | Crate rámegy egy EndTile-ra, majd eltűnik és az Őt aktívan toló Worker kap pontot. Ha nincs több Crate a pályán, akkor a pályának vége. |

|  |  |
| --- | --- |
| **Use-case neve** | Crate Squashes Worker |
| **Rövid leírás** | Crate összenyom egy Worker-t. |
| **Aktorok** | Crate |
| **Forgatókönyv** | Crate mozdulni próbál, maga előtt tolna egy Workert, de az nem tud mozdulni, emiatt a Crate rámegy a Workerre és megöli Őt. |

|  |  |
| --- | --- |
| **Use-case neve** | Hole Swallows Crate |
| **Rövid leírás** | Hole elnyel egy Crate-t. |
| **Aktorok** | Hole |
| **Forgatókönyv** | Hole-ra, ami nyitva van beleesik egy Crate, emiatt eltűnik. Amennyiben ez volt az utolsó Crate, akkor a pálya véget ér. |

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

A szekeleton menüvezérelt módon fog működni. A felhasználónak meg kell adnia a kívánt parancsnak a kódját, majd a program lefuttatja azt. A program indítás után ki kell választani a pályát. A menü tervezett felépítése itt látható:

1. Worker moves

\*1.1 Szabad az út? Y/N

\*1.2 Eltolt valamit? Y/N

2. Tile-re Worker érkezne

\*2.1 Üres a Tile? Y/N

\*2.1.1 Doboz van a Tile-n? Y/N  
 \*2.1.2 Másik worker van a Tile-n? Y/N  
3. A Tile-re dobozt tolnának  
 \*3.1 Üres a Tile? Y/N

\*3.1.1 Doboz van a Tile-n? Y/N  
 \*3.1.2 Másik Worker van a Tile-n? Y/N  
4. Egy Játékossal doboz ütközik  
 \*4.1 Van hova csúsznia a játékosnak? Y/N  
5. Egy lyukra dobozt tolnak

\*5.1 Nyitva van a lyuk? Y/N

6. Egy lyukra játékos akar lépni

\*6.1 Nyitva van a lyuk? Y/N

7. Egy kapcsolóra dobozt tolnak

\*7.1 Van valami a kapcsolóhoz tartozó lyukon? Y/N

\*7.1.1 Doboz van a lyukon? Y/N

9. Kilépés a játékból

\*9.1 A játék valamelyik játékos győzelmével, vagy felhasználói megszakítással zárult. 1/2/q

A \*-gal jelölt menü pontokat nem lehet kiválasztani, ezt a program automatikusan megteszi, ha a felhasználó a parancs szülőjét meghívta. Ezek (1 kivétellel) igen/nem típusú kérdések. A kérdések végén láthatóak a lehetséges válaszok. Az utolsó pont lefuttatásnál az ‘1’ és ‘2’ lehetőségek lefuttatása után a játék újra visszalépteti a játékost a pálya választási fázisba, felhasználói megszakítás esetén pedig kilép a program.

Egy általános parancs futása a következő képen néz ki:

! Parancs kódja és neve

> [ClassName].MethodName(VariableName) (Metódus hívás)

? A metódus futása során előforduló elágazások kiválasztása végett kérdések.

# ValueName=NewValue

< [ClassName].MethodName(VariableName):ReturnValue (Metódus visszatérése, annak értékével együtt.)

A parancs futása úgy kezdődik, hogy kommentben (!-lel induló sor) kiírja a szekeleton, hogy melyik parancs fog lefutni, majd a futás közben előforduló metódus hívások a elején egy ‘>’ karakter jelzi, hogy metódus hívásról van szó, majd szögletes zárójelek között következik a metódus osztályának a neve, amit pont után követ e metódus neve, valamint paraméter(ek) értéke(i).

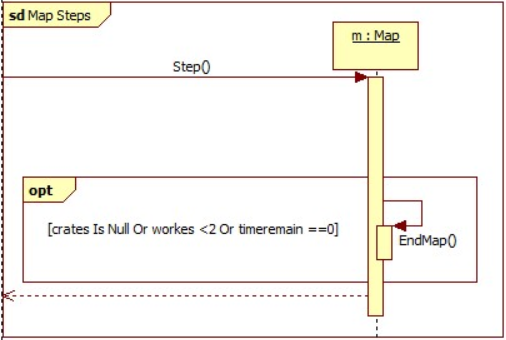
Egy metódus futása közben előforduló elágazások közül az ág kiválasztására a felhasználónak kérdést (?-lel kezdődő sor) tesz fel a program, majd a kiválasztott ágban folytatja tovább.

Amikor egy metódus visszatér, akkor a sor elején ‘<’ karakter jelzi ezt. A metódus híváshoz hasonlóan kiírja a program, mely metódus tér vissza, majd kettőspont után kiírja a visszatérési értéket. (Void esetén “void”-ot.)

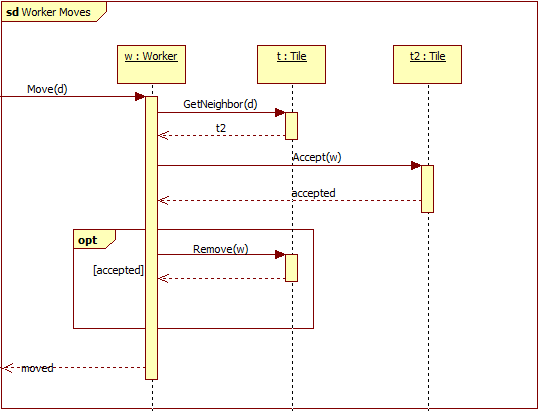
Metódus futása közben előfordulhatnak kommentek. (!-lel kezdődő sorok)

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

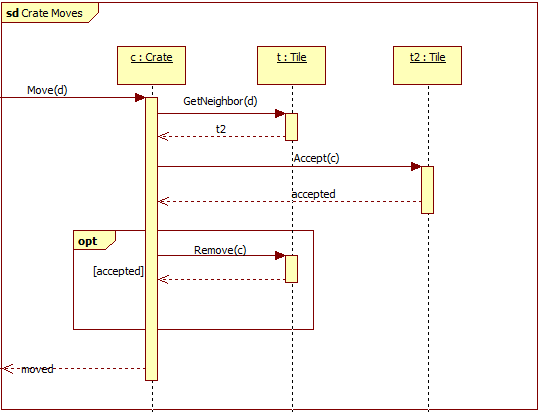
*Az ábrák alatt feltüntetésre kerül, mely Use-Case-hez tartozik.*



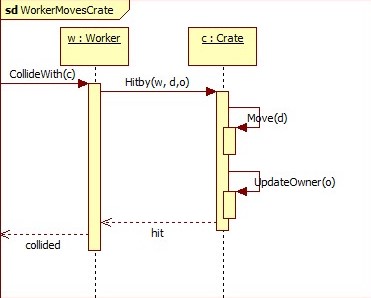
5.3.1. ábra: Worker wins, Worker loses



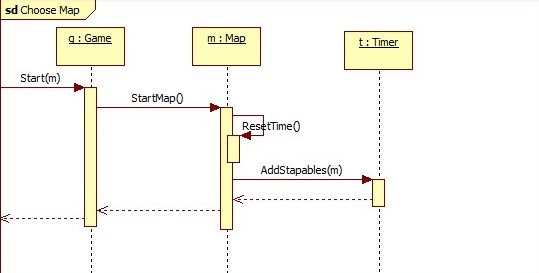
5.3.2. ábra: Worker Moves



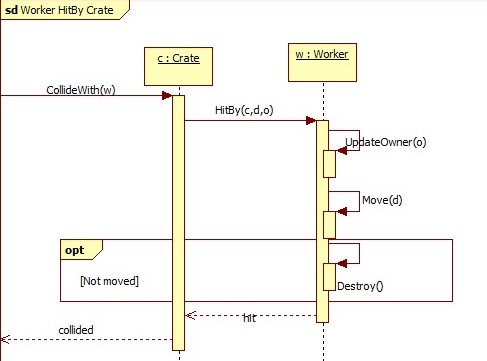
5.3.3. ábra: Crate Moves



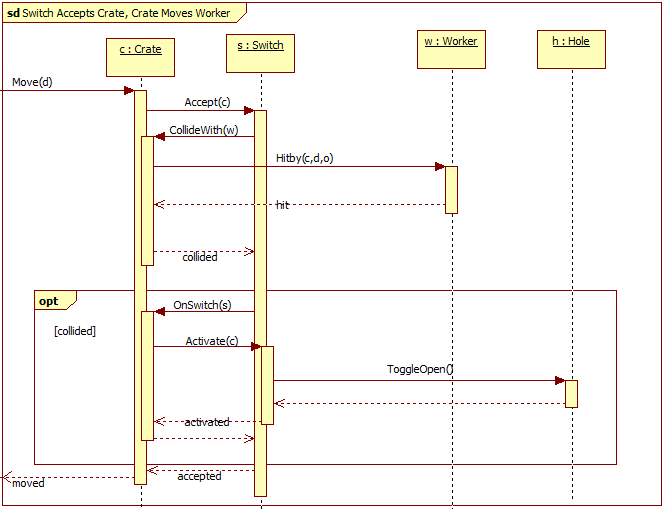
5.3.4. ábra: Worker Moves Crate



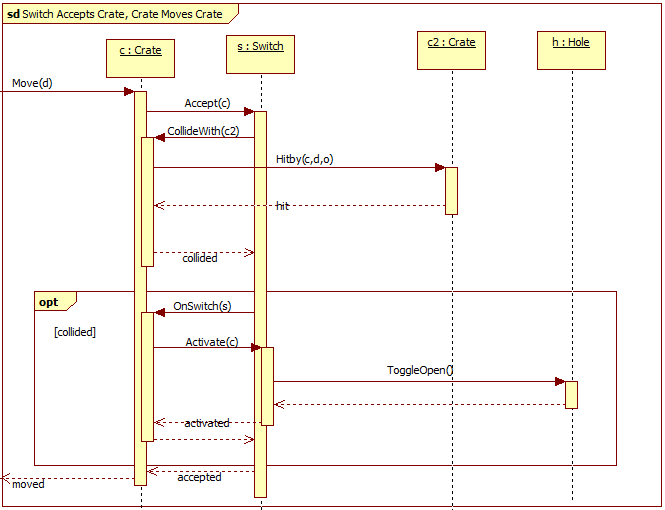
5.3.5. ábra: Player Chooses Map



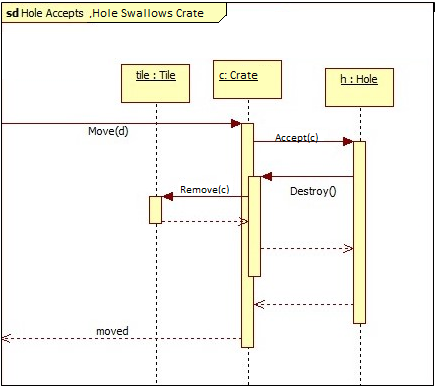
5.3.6. ábra: Crate Moves Worker, Crate Squashes Worker



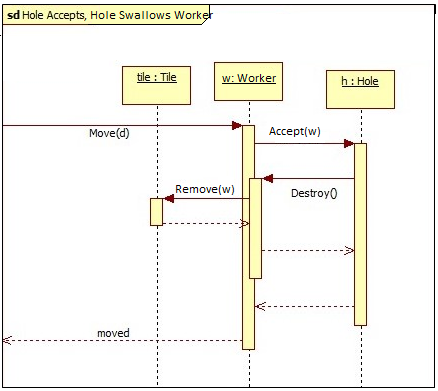
5.3.7. ábra: Crate Activates Switch, Switch Toggles Hole, Hole Activates, Crate Moves Worker



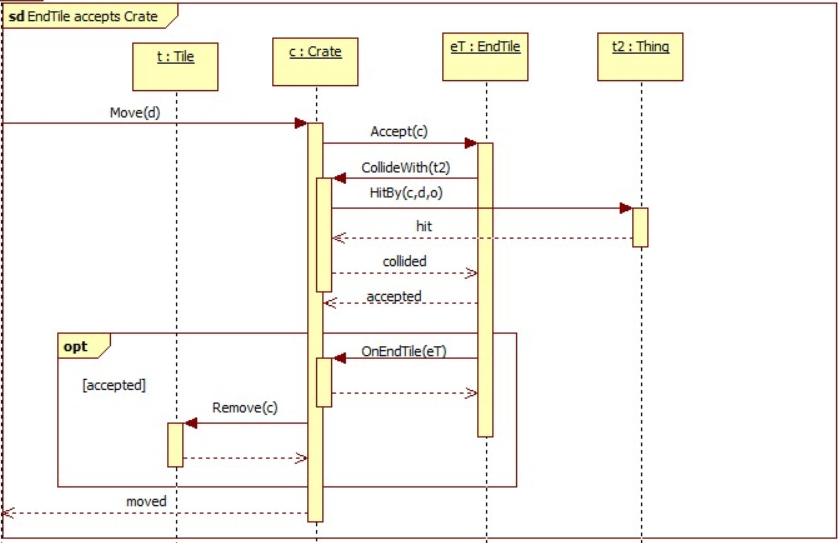
5.3.8 ábra: Crate Activates Switch, Switch Toggles Hole, Hole Activates, Crate Moves Crate



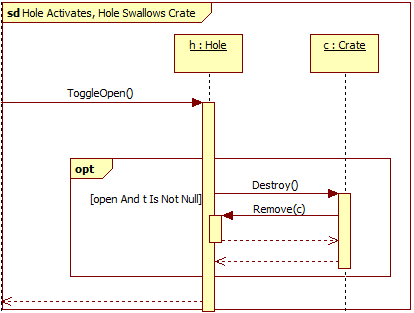
5.3.9. ábra: Hole Swallows Crate



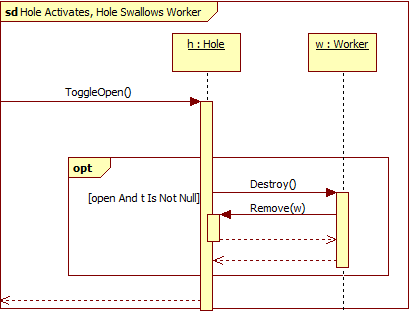
5.3.10. ábra: Hole Swallows Worker



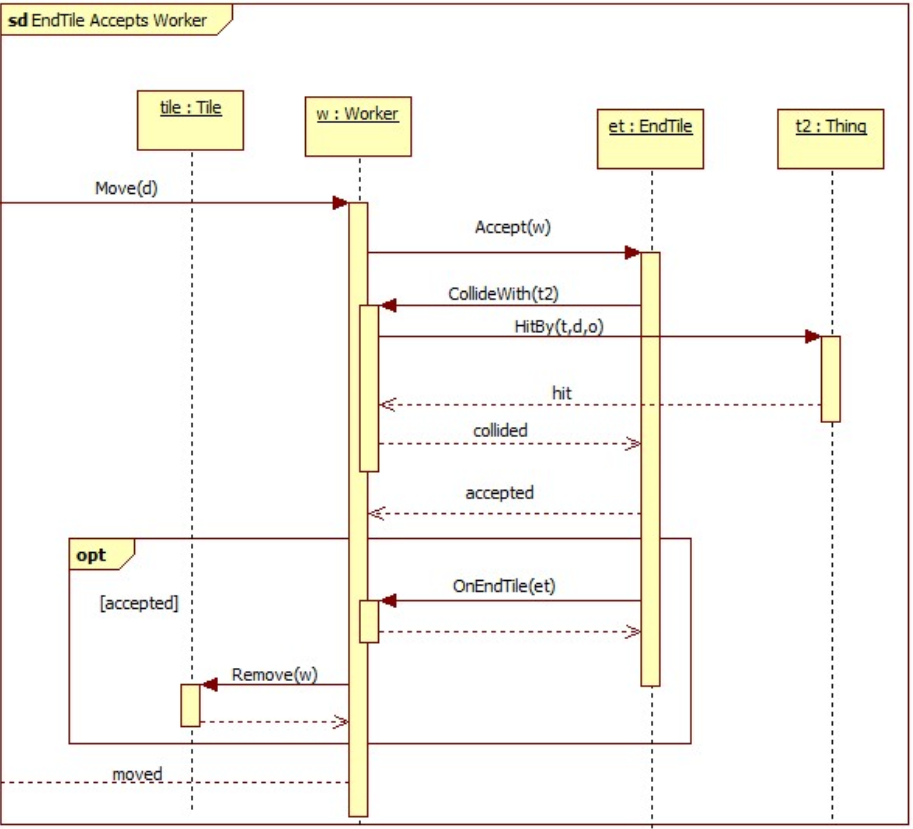
5.3.11. ábra: Crate Moves Onto Endtile



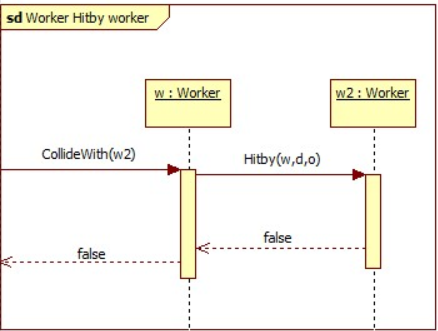
5.3.12. ábra: Hole Activates, Hole Swallows Crate



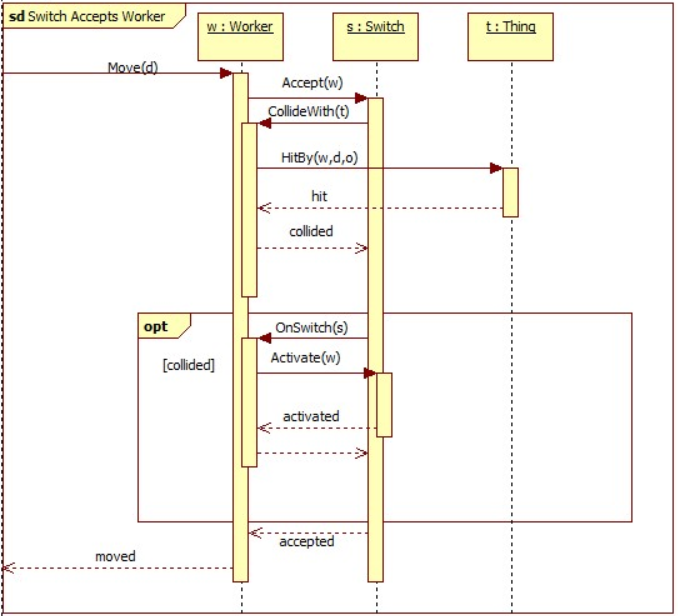
5.3.13. ábra: Hole Activates, Hole Swallows Worker



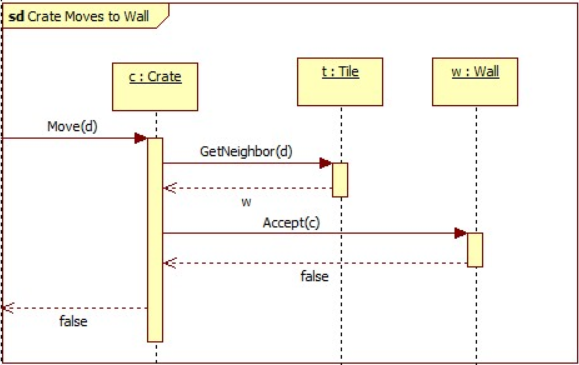
5.3.14. ábra: Worker Moves



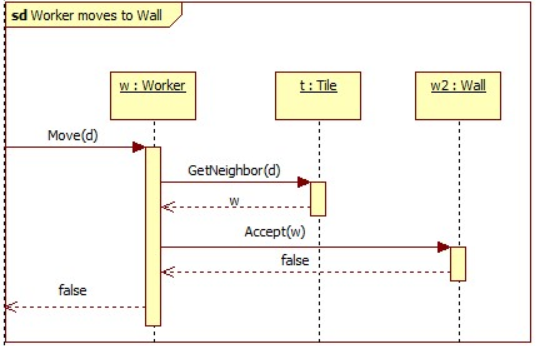
5.3.15. ábra: Worker Moves



5.3.16. ábra: Worker Moves

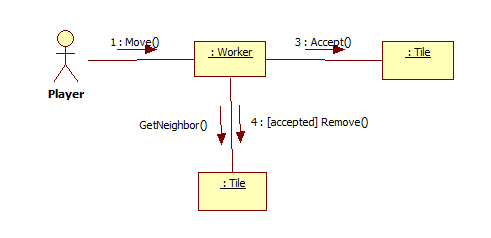


5.3.17. ábra: Crate Moves

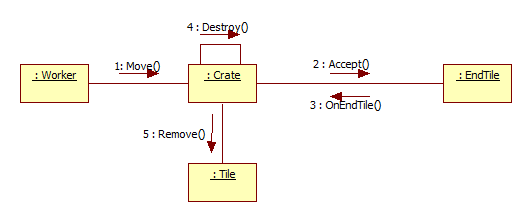


5.3.18. ábra: Worker Moves

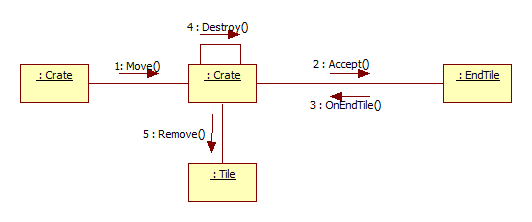
* 1. ***Kommunikációs diagramok***

**

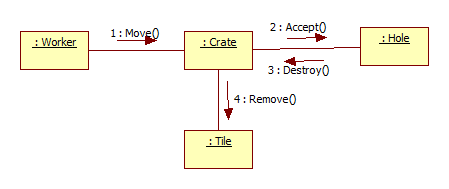
*5.4.1. ábra :Worker Moves*

**

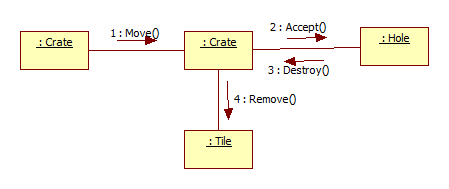
*5.4.2. ábra:* Crate Moves Onto EndTile, Worker Moves Crate



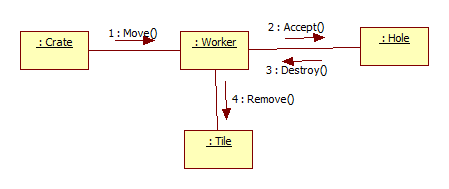
*5.4.3. ábra:*Crate Moves Onto EndTile, Crate Moves Crate



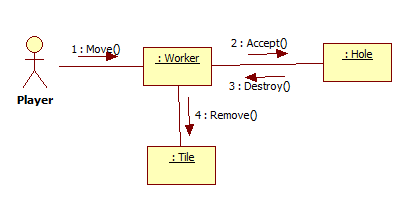
*5.4.4. ábra:* Hole Swallows Crate, Worker Moves Crate



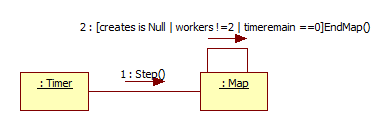
*5.4.5. ábra:* Hole Swallows Crate, Crate Moves Crate



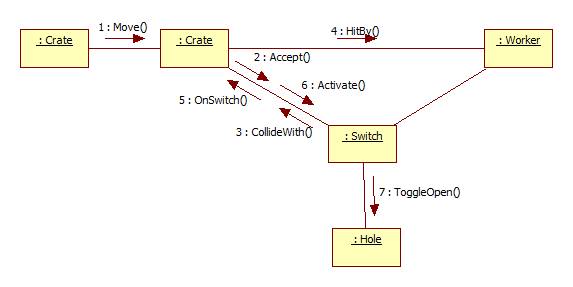
*5.4.6. ábra:* Hole Swallows Worker, Crate Moves Worker, Worker Loses



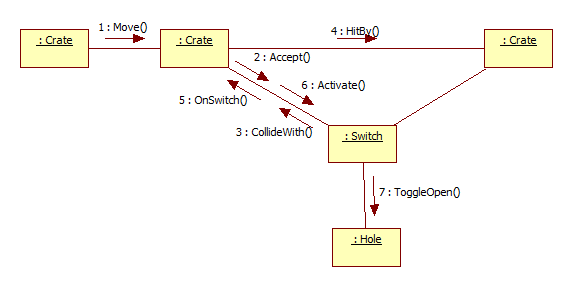
*5.4.7. ábra:* Hole Swallows Worker , Worker Moves , Worker Loses



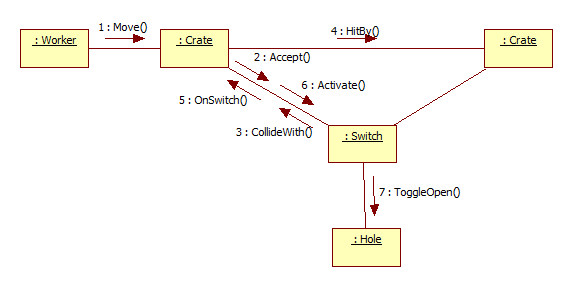
*5.4.8. ábra:* Worker Wins, Worker Loses



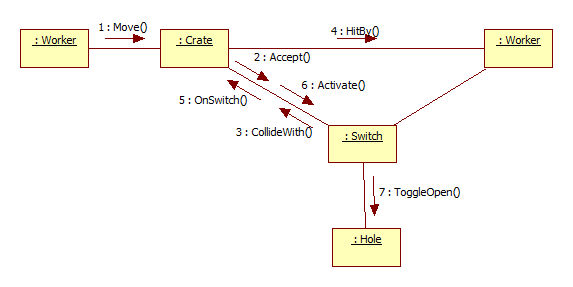
*5.4.9. ábra:* Crate Activates Switch, Switch Toggles Hole, Hole Activates, Crate Moves Crate,Crate Moves Worker



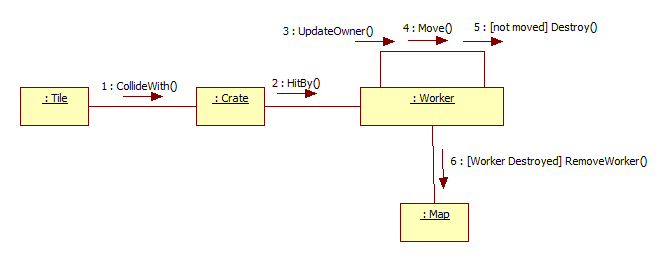
*5.4.10. ábra:* Crate Activates Switch, Switch Toggles Hole, Hole Activates, Crate Moves Crate



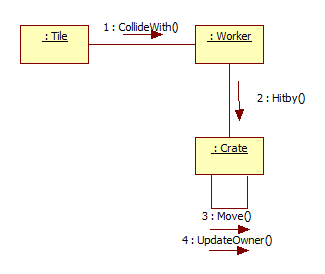
*5.4.11. ábra:* Crate Activates Switch, Switch Toggles Hole, Hole Activates, Crate Moves Crate, Worker Moves Crate



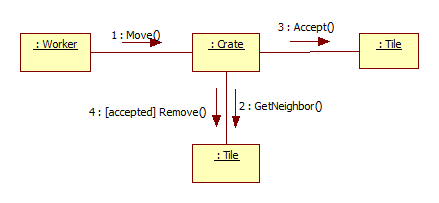
*5.4.12. ábra:* Crate Activates Switch, Switch Toggles Hole, Hole Activates, Crate Moves Worker, Worker Moves Crate



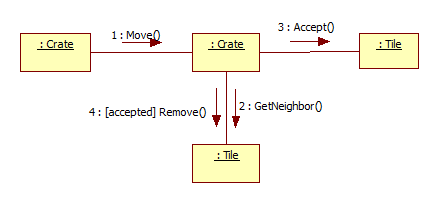
*5.4.13. ábra:* Crate Squashes Worker



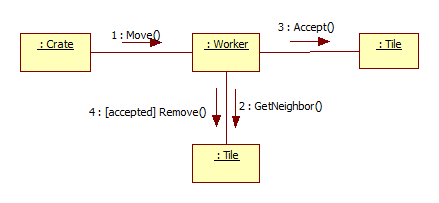
*5.4.14. ábra:* Worker Moves Crate



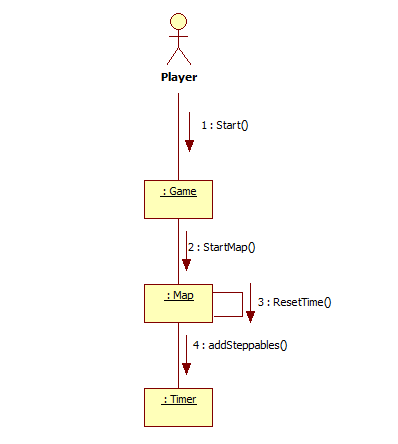
*5.4.15. ábra:* Crate Moves, Worker Moves Crate

**

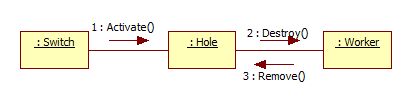
*5.4.16. ábra:* Crate Moves, Crate Moves Crate

**

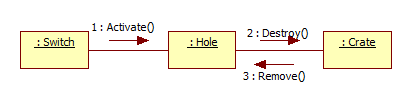
*5.4.17. ábra:* Crate Moves, Crate Moves Worker

**

*5.4.18. ábra:* Player Chooses Map



*5.4.19. ábra:* Hole Activates, Hole Swallows Worker



*5.4.20. ábra:* Hole Activates, Hole Swallows Crate

* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.03.05. 19:00 | 2,5 óra | Torma | Tevékenység: Use-Casek elkészítése és felvitele a dokumentumba. |
| 2018.03.06. 16:00 | 2,5 óra | Tóth D. | Tevékenység: Szekvencia diagramok elkészítése és felvitele a dokumentumba. |
| 2018.03.07. 20:00 | 2,5 óra | Tóth V. | Tevékenység: Kommunikációs diagramok elkészítése és felvitele a dokumentumba. |
| 2018.03.08. 21:00 | 2,5 óra | Veres | Tevékenység: Felhasználói felület megtervezése. |
| 2018.03.09. 20:00 | 2,5 óra | Pünkösd | Tevékenység: Dokumentum lektorálása, hibák javítása. |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Fájl neve** | **Méret** | **Keletkezés ideje** | **Tartalom** |
| Crate.java | 2176 btye | 2018.03.12. 20:20 | A játékbeli dobozt reprezentáló osztály és implementációja. |
| Direction.java | 378 byte | 2018.03.12. 20:20 | A Direction enumeráció, a játékbeli irányok enumerációja. |
| EndTile.java | 879 byte | 2018.03.12. 20:20 | A játékbeli dobozok célját reprezentáló osztály és implementációja. |
| Game.java | 2881 byte | 2018.03.12. 20:20 | A játékot reprezentáló osztály és implementációja. |
| Hole.java | 2323 byte | 2018.03.12. 20:20 | A játékbeli lyukakat és csapdákat reprezentáló osztály és implementációja. |
| Main.java | 737 byte | 2018.03.12. 20:20 | A program indulási pontja, a Main függvényt tartalmazó osztály. |
| Map.java | 5729 byte | 2018.03.12. 20:20 | A játékban lévő pályákat reprezentáló osztály és implementációja. |
| Steppable.java | 214 byte | 2018.03.12. 20:20 | Interfész, amit minden olyan osztály valósít meg, ami időközönként “lép”, vagyis periodikusan végrehajt valamit. |
| Switch.java | 1906 byte | 2018.03.12. 20:20 | A csapdát kapcsoló “switch”-eket reprezentáló osztály és implementációja. |
| Thing.java | 4496 byte | 2018.03.12. 20:20 | A munkás és doboz közös őse, absztrakt osztály és bizonyos függvények implementációja. |
| Tile.java | 3308 byte | 2018.03.12. 20:20 | A pálya ezekből az elemekből (és speciális változataiból) áll. |
| Timer.java | 2872 byte | 2018.03.12. 20:20 | Időzítő, amely léptet minden a Steppable interfészt megvalósító és a Timernél regisztrált objektumot. |
| Wall.java | 495 byte | 2018.03.12. 20:20 | A pályán lévő falakat és oszlopokat realizáló osztály és implementációja. |
| Worker.java | 2702 byte | 2018.03.12. 20:20 | A játékos által irányított munkásokat reprezentáló osztály és implementációja. |
| Menu.java | 16056 byte | 2018.03.12. 20:20 | A szekeleton menüjét megvalósító osztály. |
| MANIFEST.mf | 47 Bytes | 2018.03.18. 23:00 | Java manifest file, ami többek közt megmondja, hogy a JAR fileban hol van a main class. |
| build.sh | 362 Bytes | 2018.03.18. 0:45 | A JAR file buildelés automatizálására szolgáló script. |
| build.bat | 355 Bytes | 2018.03.18. 0:45 | A JAR file buildelés automatizálására szolgáló script. |
| options.txt | 80 bytes | 2018.03.18. 0:45 | A fordításhoz szükséges opciók. |
| files.txt | 334 bytes | 2018.03.18. 0:45 | Fordítandó fileok listája. |

* + 1. **Fordítás**

A fordításhoz szükséges környezet: Microsoft Windows vagy GNU/Linux operációs rendszer illetve Java JDK 9-es verzió.

A fordítás, és JAR fájl buildelés megkönnyítésére mellékeltünk egy buildelő scriptet mind Windowsra, mind Linuxra.

**Buildelés Windows alatt:**

Kattintsunk duplán a build.bat állományra.

**Buildelés Linux alatt:**

Tegyük futtathatóvá a build.sh állományt, majd futtassuk a ./build.sh paranccsal abból a mappából amiben van.

**Troubleshooting:**

Ha a script lefutott, de nem jelent meg a JAR file, akkor ellenőrízzük, hogy fennvan-e a javac 1.9-es verziója (JDK9), illetve, hogy a megfelelő mappából futtatuk-e. Illetve, hogy a fordításhoz szükséges binárisok mappája benne van-e a PATH környezeti változónkban.

**Fordítás Java 8-cal**

Ha nem elérhető a Java 9 a számítógépünkön, akkor lefordíthatjuk/futtathatjuk Java 8-cal is. Ám ebben az esetben nem tudjuk biztosítani, hogy a program pontosan úgy fut, mint ahogy terveztük. Ha Java 8-cal szeretnénk fordítani, akkor az ***options.txt*** állományban módosítsuk a **-source és -target** argumentumok után az *1.9-et 1.8-ra*, mentünk és próbáljuk újra a fordítást.

* + 1. **Futtatás**

A lefordított JAR fájlt a

|  |
| --- |
| java -jar KillerSokoban.jar |

paranccsal futtathatjuk.

* 1. ***Értékelés***

|  |  |
| --- | --- |
| **Tag neve** | **Munka százalékban** |
| Tóth Dániel Péter | 20% |
| Pünkösd Marcell | 20% |
| Tóth Vince | 20% |
| Torma Kristóf | 20% |
| Veres Csaba Miklós | 20% |

* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.03.12. 21:00 | 4 óra | Pünkösd | Tevékenység: Analízis modell lekódolása. |
| 2018.03.13. 10:00 | 4 óra | Tóth D. | Tevékenység: Menü elkészítése. |
| 2018.03.14. 16:00 | 4 óra | Veres | Tevékenység: Analízis modellen nem szereplő logika elkészítése. |
| 2018.03.15. 22:00 | 4 óra | Tóth V. | Tevékenység: Szkeleton tesztelése, hibák javítása |
| 2018.03.15. 21:00 | 4 óra | Torma | Tevékenység: Kód véglegesítése, dokumentum összeállítása, beadás. |

1. **Prototípus koncepciója**
   1. ***Változás hatása a modellre***

Worker

**A munkások bár több ládát is eltolhatnak egyszerre, minden munkás rá jellemző erővel tol. Ha a ládák együttes tapadási súrlódási ereje ennél nagyobb, akkor a tolás nem sikerül.**

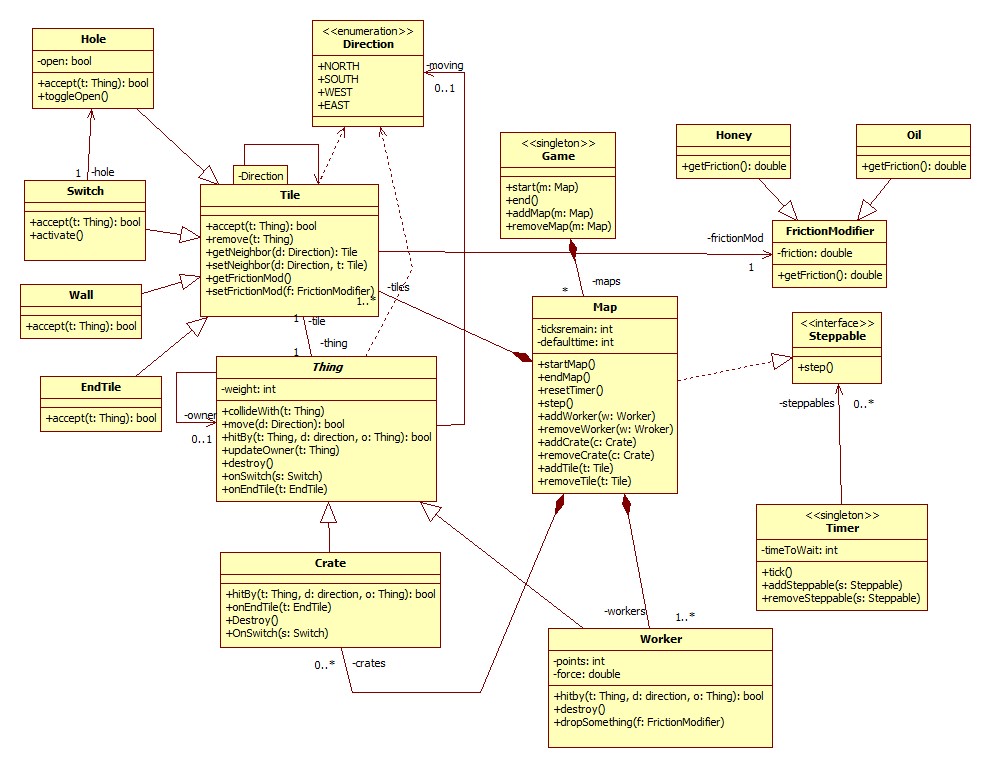
Ehhez felvettünk egy force változót, aminek segítségével meg lehet állapítani, hogy az adott munkás el tudja-e tolni az előtte lévő objektumokat.

Tile

**A padlóra különböző kenőanyagokat tehetnek a munkások: olajat, amitől csúszósabb lesz (csökken a tapadása) és mézet, amitől ragacsos (nő a tapadása).**

Ehhez felvettünk egy új osztályt és ennek két leszármazottját (FrictionModifier), ami, ha rákerül a tile-ra, akkor a rárakott objektum segítségével módosul a tile tapadása.

* + 1. **Módosult osztálydiagram**

**

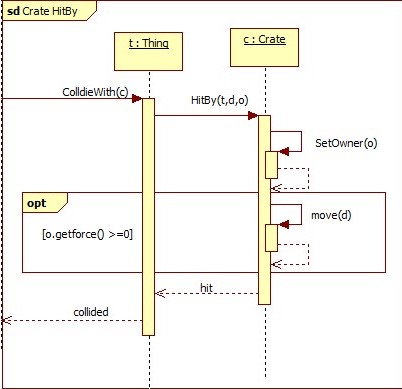
* + 1. **Új vagy megváltozó metódusok**

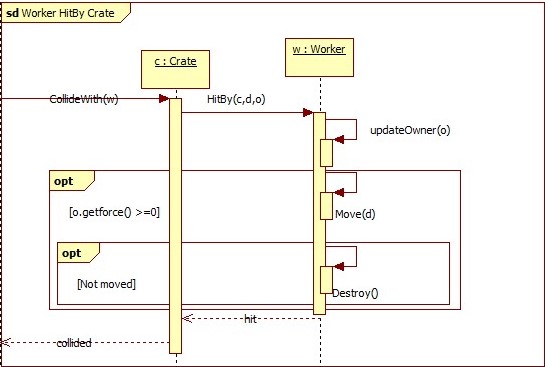
**Crate::bool HitBy(t: Thing,d: Direction,o: Thing)**: közli egy másik Thinggel hogy nekiment egy másik Thing, és, hogy mozduljon el, ha az ownerének van ereje mozgatni őket.

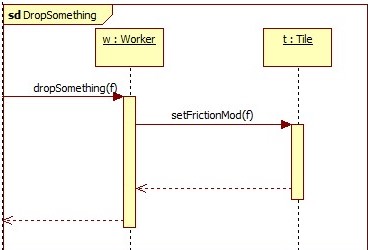
**Worker::bool HitBy(c: Crate,d: Direction,o: Thing)**: közli egy másik Thinggel hogy nekiment egy másik Thing, és, hogy mozduljon el, ha az ownerének van ereje mozgatni őket.

**Worker::void dropSomething(f: FrictionModifier)**: Lerak a tilera (ahol van) egy FrictionModifier-t.

* + 1. **Szekvencia-diagramok**

**

**

**

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

Az interfész csak a szabványos bemenetről fogad parancsokat, és a szabványos kimenetre írja ki az esetleges kimenetet. Ezáltal pipeok segítségével akár segédprogram segítségével is tesztelhető. A testesetek a prototípusnak adandó parancsok sorozatából, valamint az adott sorozatra adandó helyes kimenet található. A tesztek sikeresek, ha a valós, és a leírt elvárt kimenet megegyezik. A program bemeneti nyelve nem case sensitive.

* + 1. **Bemeneti nyelv**

***newMap***

**Leírás:** Új Map elkészítése, ez előzőleg készített mapot ez után nem lehet módosítani.

**Opciók: -**

***newTile***

**Leírás:** Új Tile lerakása, opcionálisan megadható típus. Lerakás után kiírja a Tile ID-jét, amelyet további munkához lehet használni.

**Opciók:** TileType (Ha nem felismert típus érkezik, akkor alapértelmezett Tile-t rak le.)

***connectTiles***

**Leírás:** Két Tile csatlakoztatása. A Direction az első Tile szempontjából értelmezendő.

**Opciók:** TileID1 Direction TileID2

***newThing***

**Leírás:** Új Thing lerakása az adott ID-jű Tilera. Kiírja az adott Thing ID-jét.

**Opciók:** ThingType (Ha nem sikerül felismerni, akkor Crate-et rak le.) TileID [Opcionális, ha a ThingType==Worker: PushForce]

***toggleTimer***

**Leírás:** Timer lépegetését ki/bekapcsolja, valamint kiírja a timer aktuális állapotát.

**Opciók:** -

***getTimerState***

**Leírás:** Kiírja a Timer aktuális állapotát.

**Opciók:** -

***putFrictionModifieronTile***

**Leírás:** Lerak egy adott típusú FrictionModifiert a kapott ID-jű Tilera.

**Opciók:** Type (Ha nem ismeri fel, akkor Oil-t rak le.) TileID

***getTileState***

**Leírás:** Kiírja, hogy van-e Thing és/vagy FrictionModifier az adott ID-jű Tileon.

**Opciók:** TileID

***moveWorker***

**Leírás:** Elmozgatja az adott ID-jű Wörker-t egy adott irányba. Kiírja, sikeres volt-e a mozgás.

**Opciók:** WorkerID Direction

***step***

**Leírás:** Steppable-k manuális léptetése.

**Opciók: -**

***getGameState***

**Leírás:** Kiírja a játék jelenlegi állását.

**Opciók: -**

***XMLprepare***

**Leírás:** Ez után a parancs után XML következik.

**Opciók: -**

***XMLover***

**Leírás:** Eddig a parancsig tartott az XML bemenet.

**Opciók: -**

***VerboseMode***

**Leírás:** Nem csak a parancsoknak van kimenete, hanem az eseményeknek is.

**Opciók: -**

***SetHoleState***

**Leírás:** Beállítja a luk állapotát.

**Opciók:** holeTileID open: nyitva van-e

***ConnectSwitchTo***

**Leírás:** Összeköti a kapcsolót a csapdával.

**Opciók:** switchTileID holeTileID

A pályákat leíró fájlok XML formátumban lesznek, a következő DTD-k szerint:

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (#PCDATA)

<!ELEMENT type (#PCDATA)

<!ELEMENT connectsto (#PCDATA)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (#PCDATA)

<!ELEMENT tileid (#PCDATA)

<!ELEMENT force (#PCDATA)

<!ELEMENT frictionmoifier (tileid, type)>

<!ELEMENT tileid (#PCDATA)

<!ELEMENT type (#PCDATA)

]>

* + 1. **Kimeneti nyelv**

A következő parancsokanak van kimenete, ami a következő formátumban történik:

***newTile***

Kimeneti formátum:TileID (integer)

***newThing***

Kimeneti formátum:ThingID (integer)

***getTimerState***

Kimeneti formátum:isEnabled (bool)

***getTileState***

Kimeneti formátum: ThingID (integer) ThingType (String) FrictionModifierType (String)

***moveWorker***

Kimeneti formátum: moved (bool)

***getGameState***

Kimeneti formátum:

*Workers (literal)*

*WorkerID (integer) force (double)*

*Crates (literal)*

*CrateID (integer)*

*Maps (literal)*

*numofMaps (integer)*

* 1. ***Összes részletes use-case***

|  |  |
| --- | --- |
| **Use-case neve** | **connectTiles** |
| **Rövid leírás** | Összecsatlakoztat két Tilet egy bizonyos irányból. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A két, paraméterben megadott Tile-t összecsatlakoztatja egy szintén megadott irányból amely az elsőnek megadott Tile szempontjából értelmezendő. |

|  |  |
| --- | --- |
| **Use-case neve** | **getGameState** |
| **Rövid leírás** | Kiírja a játék aktuális állapotát. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A parancs hatására a program kilistázza a játékban lévő objektumokat(Workers, Crates, Maps), valamint azok ID-jét. |

|  |  |
| --- | --- |
| **Use-case neve** | **getTileState** |
| **Rövid leírás** | Megmutatja az adott Tile jelenlegi állapotát. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A paraméterben megadott ID-jű Tile-ról megmondja, hogy az adott pillanatban van-e rajta Thing vagy súrlódást módosító anyag illetve, hogy milyen típusú. |

|  |  |
| --- | --- |
| **Use-case neve** | **getTimerState** |
| **Rövid leírás** | Kiírja a Timer állapotát. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A parancs hatására a program kiírja, hogy a Timer az adott pillanatban aktív-e. |

|  |  |
| --- | --- |
| **Use-case neve** | **moveWorker** |
| **Rövid leírás** | Egy Worker mozgatása. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Az adott ID-jű Workert a paraméterben megadott irányban lévő Tile-ra próbálja mozgatni, ütközteti az ott lévő dolgokkal majd kiírja, hogy sikeres volt-e a mozgatás. |

|  |  |
| --- | --- |
| **Use-case neve** | **newMap** |
| **Rövid leírás** | Új Map létrehozása. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A parancs hatására egy új Map-ot hoz létre. |

|  |  |
| --- | --- |
| **Use-case neve** | **newThing** |
| **Rövid leírás** | Új Thing létrehozása. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A paraméterben megadott ID-jű Tile-ra lerak egy Thinget ami alapértelmezetten Crate azonban egy paraméterrel módosítható Worker-é. Ebben az esetben megadható a Worker tolóereje. |

|  |  |
| --- | --- |
| **Use-case neve** | **newTile** |
| **Rövid leírás** | Új Tile létrehozása |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Létrehoz egy új a paraméterben megadott típusú Tile-t majd kiírja az ID-jét, hogy az új Tile-t tesztekre lehessen használni. |

|  |  |
| --- | --- |
| **Use-case neve** | **putFrictionModifieronTile** |
| **Rövid leírás** | FrictionModifier lehelyezése. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Az adott ID-jű Tile-ra lerak egy FrictionModifiert ami alapértelmezetten Oil azonban állítható hogy Honey legyen. |

|  |  |
| --- | --- |
| **Use-case neve** | **step** |
| **Rövid leírás** | Léptetés |
| **Aktorok** | A Timer osztály. |
| **Forgatókönyv** | Minden léptethető objektumot eggyel léptet. |

|  |  |
| --- | --- |
| **Use-case neve** | **toggleTimer** |
| **Rövid leírás** | A Timer osztály ki-be kakapcsolása |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | Hatására a Timer bekapcsolt állatból kikapcsoltba, kikapcsoltból pedig bekapcsoltba megy át. Kiírja az aktuális pillanatban hátralévő időt. |

|  |  |
| --- | --- |
| **Use-case neve** | **XMLprepare** |
| **Rövid leírás** | XML előkészítése. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A parancs hatására a program XML-ként értelmezi a parancs kiadása utáni bemenetet. |

|  |  |
| --- | --- |
| **Use-case neve** | **ConnectSwitchTo** |
| **Rövid leírás** | Kapcsoló csatolása egy csapdához. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A parancs beállítja, az adott kapcsoló melyik csapdához tartozik. |

|  |  |
| --- | --- |
| **Use-case neve** | **SetHoleState** |
| **Rövid leírás** | Beszédes mód bekapcsolása |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A parancs beállítja egy adott lyuk állapotát. |

|  |  |
| --- | --- |
| **Use-case neve** | **VerboseMode** |
| **Rövid leírás** | Beszédes mód bekapcsolása |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A parancs hatására a program nem csak a parancsok kimenetét jegyzi, hanem a program eseményeit is. |
| **Use-case neve** | **XMLover** |
| **Rövid leírás** | XML lezárása. |
| **Aktorok** | Tesztelő |
| **Forgatókönyv** | A parancs hatására a program már nem fog XML-t várni a bemeneten. |

* 1. ***Tesztelési terv***

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Workerek mozgása** |
| **Rövid leírás** | A Workerek mozgásának tesztelése. |
| **Teszt célja** | Teszteli, hogy a workerek képesek mozogni a pályán, ha semmi nem állja útjukat. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Akadályok** |
| **Rövid leírás** | A Workerek mozgásának tesztelése. |
| **Teszt célja** | Teszteli, hogy a workerek nem tudnak áthaladni az akadályokon (falak, nem tolható láda). |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Szimpla doboz tolás** |
| **Rövid leírás** | A Workerek mozgásának tesztelése |
| **Teszt célja** | Teszteli, hogy a workerek képesek egy doboz eltolására. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Több doboz tolás** |
| **Rövid leírás** | A Workerek mozgásának tesztelése |
| **Teszt célja** | Teszteli, hogy a worker képes az erejét nem meghaladó mennyiségő dobozt eltolni. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Túl sok doboz tolás** |
| **Rövid leírás** | Workerek mozgásának tesztelése |
| **Teszt célja** | Teszteli, hogy a worker nem képes az erejét meghaladó mennyiségő dobozt eltolni. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Workerek dobozzal tolása** |
| **Rövid leírás** | A Workerek mozgásának tesztelése |
| **Teszt célja** | Teszteli, hogy a workerek képesek eltolni egymást a dobozok segítségével |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Doboz falba tolás** |
| **Rövid leírás** | A Workerek mozgásának tesztelése |
| **Teszt célja** | Teszteli, hogy a dobozt nem lehet falba tolni. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Lyuk vs. Worker** |
| **Rövid leírás** | A Pályán található lyukak tesztelése |
| **Teszt célja** | Teszteli, hogy a pályán található nyitott lyukakra lépve, meghal a játékos. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Lyuk vs. Láda** |
| **Rövid leírás** | A Pályán található lyukak tesztelése |
| **Teszt célja** | Teszteli, hogy a pályán található nyitott lyukra ládát tolva eltünteti azt. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Zárt lyuk teszt** |
| **Rövid leírás** | A Pályán található lyukak tesztelése |
| **Teszt célja** | Teszteli, hogy a pályán található lyukak zárt állapotban sima tile-ként viselkednek. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Trap switch** |
| **Rövid leírás** | A Pályán található lyukak tesztelése |
| **Teszt célja** | Teszteli, hogy a pályán található lyukakat a hozzájuk tartozó kapcsolóra ládát tolva lehet kapcsolni. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Friction modification test** |
| **Rövid leírás** | A surlódást befolyásoló tényezők tesztje |
| **Teszt célja** | Teszteli, hogy a worker képes lehelyezni a súrlódást módosító anyagokat. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Oil teszt** |
| **Rövid leírás** | A surlódást befolyásoló tényezők tesztje |
| **Teszt célja** | Teszteli, hogy egy játékos, aki az erejét éppen meghaladó összsúrlódású dobozt nem tud eltolni, olajos felülen képes erre. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Honey teszt** |
| **Rövid leírás** | A surlódást befolyásoló tényezők tesztje |
| **Teszt célja** | Teszteli, hogy egy játékos az erejét éppen nem meghaladó összsúrlódású dobozt képes eltolni, ragacsos felületen nem képes erre. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Timeout** |
| **Rövid leírás** | Játék záró feltételek |
| **Teszt célja** | Teszteli, hogy az idő lejártával véget ér a játék. |

|  |  |  |
| --- | --- | --- |
| **Teszt-eset neve** | **Gyilkosság** | |
| **Rövid leírás** | Játék záró feltételek | |
| **Teszt célja** | Teszteli, hogy a játékosok megölik egymást láda tolással, és ezzel véget ér a játék. | |
| **Teszt-eset neve** | | **Rakodó területre tolás** | |
| **Rövid leírás** | | Játék záró feltételek | |
| **Teszt célja** | | Teszteli, hogy a dobozok rakodó területre tolásával véget ér a játék. | |

|  |  |
| --- | --- |
| **Teszt-eset neve** | **Pálya betöltése** |
| **Rövid leírás** | A pálya betöltés tesztelése. |
| **Teszt célja** | Teszteljük, hogy egy adott pálya fájl beolvasása után megfelelő a memóriába betöltött objektumok struktúrája. |

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

Mivel minden tesztre kézzel beírni majd szemre összehasonlítani az eredményeket fárasztó lenne, de szerencsére a program nem bíz semmit a véletlenre, így működése determinisztikusnak tekinthető. Ezt kihasználva egyszerűen az standard bementet, és kimenetet használva könnyen kezelhető teszteseteket írhatunk.   
Egy teszt két fájlból áll: a bemenetből, és az elvárt kimenetből. A teszt során egy egyszerű scripttel elindíjuk a programot, majd annak a standard bemenetén beadjuk a fájlt. A kimenetét rögzítjük, majd a futás végeztével összehasonlíjuk az elvárt kimenettel. Ha a kettő egyezik, akkor sikeres tesztről számolhatunk be. Ha valami eltérés van, akkor a script kiírja, hogy mi az.

* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.03.21 12:00 | 2 óra | Torma | Tevékenység: Koncepció kialakítása. |
| 2018.03.22 20:00 | 2 óra | Tóth D | Tevékenység: Diagrammok elkészítése. |
| 2018.03.23 15:00 | 2 óra | Tóth V | Tevékenység: Use-Casek elkészítése. |
| 2018.03.24 21:00 | 2 óra | Veres | Tevékenység: Specifikációk elkészítése. |
| 2018.03.25 23:00 | 2 óra | Pünkösd | Tevékenység: Maradék munka elvégzése. |

**8. Részletes tervek**

***8.1 Osztályok és metódusok tervei.***

**8.1.1 Thing**

● **Felelősség**

Absztrakt osztály, amely az általános a pályán lévő nem statikus elemet reprezentálja.

● **Attribútumok**

* **- owner : Thing:** Ez a Thing mozgatja éppen.
* **- moving: Direction:** Az irány amerre a Thing mozog.
* **- tile: Tile:** Az a Tile amin a Thing éppen tartózkodik.
* **- weight: Integer:** A Thing súlya.

● **Metódusok**

* **+move (d: Direction): boolean:** Mozgatja a Thing-et az adott irányba, ehhez meghívja az adott irányba eső Tile Accept függvényét, ami ha igazzal tér vissza végre hajtódik a mozgás és a move is igazzal tér majd vissza.
* **+collideWith (t: Thing):** **boolean:** Ütközteti a Thing-et a paraméterként kapott Thing-el, úgy hogy meghívja annak hitBy függvényét és annak visszatérési értékével tér majd vissza.
* **+hitBy(t: Thing, d: Direction):, o: Thing):** **boolean:** Ütközéskor hívódik meg. Beállítja a Thing új owner-jét. Ezenfelül itt végzi el az erőhöz kapcsolódó számításokat, amennyiben elég ereje van a munkásnak a Thing eltolásához a mozgás folytatódik ellenkező esetben false-al tér vissza.
* **+onSwitch(s: Switch):** Akkor hívódik meg, ha a Thing kapcsolóra lép. Itt nem csinál semmit, viszont a láda esetében felüldefiniálásra kerül.
* **+onEndTile(t: EndTile):** Akkor hívódik meg ha a Thing EndTile-ra lép. Itt nem csinál semmit, viszont a láda esetében felüldefiniálásra kerül.
* **+Destroy():** Elpusztítja a Thing-et. Meghívódhat például lyukra vagy láda esetében célmezőre lépéskor. Alapvetően nem csinál semmit majd felüldefiniálásra kerül a Crate és Worker osztályokban.
* **+getTile():** Visszaadja melyik Tile-on áll éppen a Thing.
* **+setTile():** Beállítja, hogy melyik Tile-on áll a Thing.
* **+getOwner (): Integer:** Visszaadja a Thing owner-jét.
* + **updateOwner (o: Thing):** Beállítja a Thing owner-jét.
* **+getWeight (): Integer:** Visszaadja a Thing súlyát.
* **+setWeight (w: Integer):** Beállítja a Thing súlyát.

**8.1.2 Crate**

● **Felelősség**

A pályán lévő ládákat reprezentáló osztály.

● **Ősosztályok**

**Thing**

● **Metódusok**

* **+onEndTile(t: EndTile):** Akkor hívódik meg, ha a Crate EndTile-ra lép. Ekkor növekszik a Crate owner-jének a pontszáma, majd a Crate megsemmisül.
* **+onSwitch(s: Switch):** Akkor hívódik meg, ha a Crate kapcsolóra lép, ekkor aktiválja a kapcsolót(azzal hogy meghívja a kapcsoló activate függvényét).
* **+Destroy():** Elpusztítja a Crate-et.

**8.1.3 Worker**

● **Felelősség**

A játékos által irányított munkást reprezentáló osztály.

● **Ősosztályok**

**Thing**

● **Attribútumok**

**Az ősosztály attribútumai mellett ezekkel rendelkezik:**

● **- points: Integer:** A munkás pontjainak a száma.

● **- force: Double:** A munkás ereje, amivel tolni tud.

● **Metódusok**

* **+hitBy(t: Thing, d: Direction):, o: Thing):** **boolean:** Hasonlóan működik, mint az ős függvénye azzal a különbséggel, hogy munkás nem tolhat másik munkást közvetlenül és ez a függvény kezeli azt is mikor láda tolódik egy munkásra.
* **+dropOil():** A munkás olajjal keni be a Tile-t amin éppen áll.
* **+dropHoney():** A munkás mézzel keni be a Tile-t amin áll.
* **+getPoints ():Integer** Visszaadja a Worker pontjait.
* **+setPoints (p: Integer):** Beállítja a Worker pontjait.
* **+getForce ():Double** Visszaadja a Worker erejét.
* **+setForce (f: Double):** Beállítja a Worker erejét.
* **+Destroy():** Elpusztítja a Worker-t.

**8.1.4 Tile**

● **Felelősség**

Az általános pályaelemet reprezentálja, amin lehet Thing.

● **Attribútumok**

* **-nexttiles[Direction]:Tile**: A Tile-ból induló Tile-ok.
* **-thing:Thing**: A Tile-on lévő thing.
* -**frictionMod:FrictionModifier:** A Tile tapadási módosítója.

**Metódusok**

* **+accept(m: Thing):boolean**Befogadja a bejövő Thing objektumot. Ha nem üres, akkor ütközteti a rajtalévővel.
* **+remove(m: Thing)**: eltávolítja az éppen rajta lévő Thinget.
* **+getNeighbor(d: Direction):Tile** d irányban lévő szomszédos Tile-lal tér vissza, vagy NULL-lal, ha nincs ebben az irányban tile.
* **+setNeighbor(d: Direction, t: Tile)**: Beállítja t-re d irányban a szomszédját.
* +**getThing():Thing:** Visszatér a Tile-on lévő Thing-el
* **+setThing(t: Thing):** Beállítja a Thing-et a paraméterül kapott t-re.
* **+getFrictionMod():FrictionModifier:** Visszaadja, hogy milyen tapadás módosító van a Tile-on éppen.
* **+setFrictionMod(f: FrictionModifier):** Beállítja a frictionMod értékét.

**8.1.5 Direction**

● **Felelősség**

Enumeráció, amelyben a játékban lévő irányok vannak tárolva.

● **Attribútumok**

**+NORTH:** Észak

**+SOUTH:** Dél

**+WEST:** Nyugat

**+East:** Kelet

**8.1.6 EndTile**

● **Felelősség**

Azt a Tile-t reprezentálja, amelyre rá kell tolni a dobozokat.

● **Ősosztályok**

**Thing**

● **Metódusok**

* **+accept(m: Thing):boolean**: Ha dobozt fogad be akkor a doboz eltűnik és a doboz owner-je pontokat kap. Ha munkás lép rá akkor ugyanúgy működik mint a Tile accept-je.

.

**8.1.7 Game**

● **Felelősség**

A játék egészét reprezentáló osztály.

● **Attribútumok**

**–maps: Map [\*]:** A játékot álló pályák összessége.

**–currentMap: Map:** Az a pálya, amin éppen játszunk.

● **Metódusok**

* **+start (m: Map):** Elindítja a paraméterül kapott pályát.
* **+end ():** Leállítja a játékot.
* **+getMaps (): Map []:** Visszaadja a maps attribútum értékét.
* **+setMaps(newvalue: Map []):** Beállítja a maps attribútum értékét a newrvalue-ra.
* **+addMap (m: Map):** Hozzá ad egy pályát a maps-hez.
* **+removeMap (m: Map):** Töröl egy pályát a maps-ből.
* **+getCurrentMap (): Map:** Visszaadja azt a pályát, amin éppen játszunk.
* **+setCurrentMap (m: Map):** Beállítja a currentMap értéket.

**8.1.8 Hole**

● **Felelősség**

A pályán lévő lyukakat reprezentáló osztály.

● **Ősosztályok**

**Tile**

● **Attribútumok**

* **-open:boolean:** Nyitott-e a lyuk.

● **Metódusok**

* **+accept (t: Thing):** Ha a lyuk nem nyitott, akkor a sima Tile accept függvényéhez hasonlóan viselkedik, ha nyitott, akkor elnyeli a rálépő Thing-et.
* **+getOpen (): boolean:** Visszaadja az open attribútum értékét.
* **+toggleOpen ():** Negálja az open attribútum értékét és biztosítja a láda vagy munkás leesését, ha éppen kinyitották és rajta állnak.

**8.1.9 Map**

● **Felelősség**

Egy pályát reprezentáló osztály.

● **Interfészek**

**Steppable**

● **Attribútumok**

* **- tiles: Tile [1..\*]:** A pályát alkotó Tile-ok.
* **-workers: Worker [1..\*]:** A pályán lévő munkások.
* **–crates : Crate[1..\*]:** A pályán lévő dobozok.
* **-ticksremain: int**: Visszamaradó idő, tickekben.
* **-defaulttime: int**: Alapértelmezett visszamaradó idő, tickekben.

● **Metódusok**

* **+ startMap ():** Inicializálja a pályát.
* **+ endMap ():** Miután eltűnt minden Crate, ez a függvény hívódik meg. Elindítja a következő pályát (ha van).
* **+step():** A pálya lép( Csökkenti a ticksremain értékét és ellenőrzi hogy vége van-e a pályának).
* **+getTiles ()**: **Tile[] :** Visszaadja a tiles attribútum értékét.
* **+ SetTiles (newvalue: Tile[]):** Beállítja a tiles attribútum értékét a newrvalue-ra.
* **+getWorkers (): Worker[] :** Visszaadja a workers attribútum értékét.
* **+setWorkers(newvalue:Worker[]):** Beállítja a workers attribútum értékét a newrvalue-ra.
* **+addWorker(w: Worker):** Egy munkást ad hozzá a pályához.
* **+removeWorker(w: Worker):** Töröl egy munkást a pályáról.
* **+addCrate(c: Crate):** Hozzáad egy ládát a pályához.
* **+removeCrate(c : Crate):** Töröl egy ládát a pályáról.
* **+addTile(t: Tile):** Egy Tile-al bővíti a pályát.
* **+removeTile(t: Tile):** Töröl egy Tile-t a pályáról.
* **+getTicksRemain():Integer:** Lekérdezi, mennyi idő van hátra.
* **+setTicksRemain (i: int):** Beállítja a visszamaradó időt.
* **+resetTimer()**: Visszaállítja a visszamaradó időt az alapértelmezettre.
* **+getDefaultTime()**:**Integer:** Lekérdezi az alapértelmezett visszamaradó időt.
* **+setDefaultTime(t: int)**: Beállítja az alapértelmezett visszamaradó időt.

**8.1.10 Steppable**

● **Felelősség**

Az általános pályaelemet reprezentálja, amin lehet Thing.

● **Metódusok**

* **+ Step():** E függvény meghívásakor lép.

**8.1.11 Switch**

● **Felelősség**

A pályán lévő Hole-okat kapcsoló kapcsolókat reprezentáló osztály.

● **Ősosztályok**

**Tile**

● **Attribútumok**

* **-hole: Hole:** Az a Hole amelyet az adott Switch kapcsol.

● **Metódusok**

* **+ accept(t:Thing):boolean** Befogadja a rálépő Thing-et, mint a rendes Tile és meghívja a Thing onSwitch függvényét.
* **+ setHole():Hole** Visszaadja a hole attribútum értékét.
* **+ getHole():Hole:** Visszaadja a hole attribútum értékét.
* **+activate():** Meghívódik ha Crate érkezik a kapcsolóra és nyitja a lyukat.

**8.1.12 Timer**

● **Felelősség**

Idózítő osztály, singleton.

● **Attribútumok**

* **-steppables: Steppable [0..\*]:** Timer által léptetendő léptethető objektumok.

● **Metódusok**

* **+ Tick():** Egyet “kattan” az óra, léptet minden lépni képes objektumot.
* **+ AddSteppable(s: Steppable)**: Hozzáad egy Steppable objektumot a léptetendő objektumok listájára.
* **+RemoveSteppable(s: Steppable)**: Eltávolítja ‘s’-t a léptetendő objektumok listájáról, ha a listában van.

**8.1.13 Wall**

● **Felelősség**

A pálya szélén és a pályán lévő falakat reprezentáló osztály.

● **Ősosztályok**

**Tile**

● **Metódusok**

* **+ accept(t:Thing):boolean:** Mindig false-sal tér vissza, mivel falra menni nem lehet.

**8.1.14 FrictionModifier**

● **Felelősség**

Ez az osztály felelős a Tile-ok tapadási együtthatójának beállítására.Ebből származik le a Honey és Oil osztály.

● **Attribútumok**

* **–friction :Double :** a tapadási együttható ,aminek az értéke ennek az osztálynak az esetében 1.

● **Metódusok**

* **+ getFriction():Double:** Visszaadja a friction értékét.

**8.1.15 Honey**

● **Felelősség**

A mézet reprezentáló osztály.

● **Ősosztályok**

**FrictionModifier**

● **Metódusok**

* **+ getFriction():Double:** Az ősosztály friction értékének másfélszeresével tér vissza

**8.1.16 Oil**

● **Felelősség**

Az olajat reprezentáló osztály

● **Ősosztályok**

**FrictionModifier**

● **Metódusok**

* **+ getFriction():Double:** Az ősosztály friction értékének felével tér vissza.
  1. ***A tesztek részletes tervei, leírásuk a teszt nyelvén***
     1. **Workerek mozgása**

**Leírás**

Teszteli, hogy a workerek képesek mozogni a pályán, ha semmi nem állja útjukat.

**Ellenőrzött funkcionalitás:**A munkás adott irányba (jelen esetben északra) mozgatása.

**Hibák amik előfordulhatnak:**A mozgás nem megy végbe, rossz irányba mozdul a munkás.  
**Bemenet**

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

]>

XMLOver

moveWorker 0 NORTH

**Elvárt kimenet**

true

* + 1. **Akadályok**

**Leírás**

Teszteli, hogy a workerek nem tudnak áthaladni az akadályokon.

**Ellenőrzött funkcionalitás:**munkás mozgatása nem tolható doboznak, doboz/munkás mozgatása falnak

**Hibák amik előfordulhatnak:**A fal accept-olja a ládát/workert, a munkás sikeresen mozog a láda helyére.

**Bemenet**

VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (wall)

<!ELEMENT connectsto (1)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (3)

<!ELEMENT type (0)

<!ELEMENT connectsto (4)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (4)

<!ELEMENT type (0)

<!ELEMENT connectsto (3)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (5)

<!ELEMENT type (wall)

<!ELEMENT connectsto (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (3)

<!ELEMENT force (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (4)

<!ELEMENT force (0)

]>

XMLOver

moveWorker 0 NORTH

moveWorker 1 NORTH

**Elvárt kimenet**

false

false

* + 1. **Szimpla doboz tolás**

**Leírás**

Teszteli, hogy a worker-ek képesek egy doboz eltolására.

**Ellenőrzött funkcionalitás:**A worker ládát tol, üres Tile ládát accept-el.

**Hibák amik előfordulhatnak:**Nem sikerül eltolni a ládát.

**Bemenet**

VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (3)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (2)

<!ELEMENT force (0)

]>

XMLOver

moveWorker 0 NORTH

**Elvárt kimenet**

Crate 1 : moved

true

* + 1. **Több doboz tolás**

**Leírás**

Teszteli, hogy a worker képes az erejét nem meghaladó mennyiségű dobozt eltolni.

**Ellenőrzött funkcionalitás:**Munkás ládát tol, láda ládát tol, üres Tile accept-el.

**Hibák amik előfordulhatnak:**A munkás nem tolja el a ládát függetlenül attól ,hogy elég ereje van, a láda nem tolja el a többi ládát.

**Bemenet**

VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (3)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (4)

<!ELEMENT type (0)

<!ELEMENT connectsto (3)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (5)

<!ELEMENT type (0)

<!ELEMENT connectsto (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (2)

<!ELEMENT force (0)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (3)

<!ELEMENT force (0)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (4)

<!ELEMENT force (0)

]>

XMLOver

moveWorker 0 NORTH

**Elvárt kimenet:**

Crate 3 : moved

Crate 2 : moved

Crate 1 : moved

true

* + 1. **Túl sok doboz tolás**

**Leírás**

Teszteli, hogy a worker nem képes az erejét meghaladó mennyiségő dobozt eltolni.

**Ellenőrzött funkcionalitás:**Munkás ládát tol, láda ládát tol

**Hibák amik előfordulhatnak:**A munkásnak sikerül ládát tolnia úgy is hogy nincs elég ereje.

**Bemenet**

VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (3)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (4)

<!ELEMENT type (0)

<!ELEMENT connectsto (3)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (5)

<!ELEMENT type (0)

<!ELEMENT connectsto (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (1)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (2)

<!ELEMENT force (0)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (3)

<!ELEMENT force (0)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (4)

<!ELEMENT force (0)

]>

XMLOver

moveWorker 0 NORTH

**Elvárt kimenet:**

false

* + 1. **Workerek dobozzal tolása**

**Leírás**

Teszteli, hogy a workerek képesek eltolni egymást a dobozok segítségével

**Ellenőrzött funkcionalitás:**Munkás ládát tol, láda munkást tol, Tile munkást accept-el.

**Hibák amik előfordulhatnak:**A munkás nem tudja eltolni a 2 Thing-et, a láda nem tolja a munkást.

**Bemenet**

VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (3)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (4)

<!ELEMENT type (0)

<!ELEMENT connectsto (3)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (2)

<!ELEMENT force (0)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (3)

<!ELEMENT force (0)

]>

XMLOver

moveWorker 0 NORTH

**Elvárt kimenet**

Crate 1 : moved

true

* + 1. **Doboz falba tolás**

**Leírás**

Teszteli, hogy a dobozt nem lehet falba tolni.

**Ellenőrzött funkcionalitás:**Worker ládát tol, Fal accept-el.

**Hibák amik előfordulhatnak:**A falra lép a doboz, így a worker is sikeresen lép.

**Bemenet**

VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (3)

<!ELEMENT type (wall)

<!ELEMENT connectsto (2)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (2)

<!ELEMENT force (0)

]>

XMLOver

moveWorker 0 NORTH

**Elvárt kimenet**

false

* + 1. **Lyuk vs. Worker**

**Leírás**

Teszteli, hogy a pályán található nyitott lyukakra lépve, meghal a játékos.

**Ellenőrzött funkcionalitás:**Lyuk munkást accept-el, munkás megsemmisül.

**Hibák amik előfordulhatnak:**A munkás nem semmisül meg mikor a nyitott lyukra lép.

**Bemenet**

VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (hole)

<!ELEMENT connectsto (1)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

]>

XMLOver

SetHoleState 2 open

moveWorker 0 NORTH

getGameState

**Elvárt kimenet**

Hole 2 : open

Worker 0 : died

true

Workers

Crates

Maps

1

* + 1. **Lyuk vs. Láda**

**Leírás**

Teszteli, hogy a pályán található nyitott lyukra ládát tolva eltünteti azt.

**Ellenőrzött funkcionalitás:**Munkás ládát tol, lyuk ládát accept-el.

**Hibák amik előfordulhatnak:**Nem sikerül a mozgás, nem pusztul el a láda.

**Bemenet**

VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (3)

<!ELEMENT type (hole)

<!ELEMENT connectsto (2)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (2)

<!ELEMENT force (0)

]>

XMLOver

SetHoleState 3 open

moveWorker 0 NORTH

**Elvárt kimenet**

Hole 3 : open

Crate 1 : died

Crate 1 : moved

true

* + 1. **Zárt lyuk teszt**

**Leírás**

Teszteli, hogy a pályán található lyukak zárt állapotban sima tile-ként viselkednek.

**Ellenőrzött funkcionalitás:**Zárt lyuk munkást accept-el.

**Hibák amik előfordulhatnak:**A munkás meghal.

**Bemenet**

VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (hole)

<!ELEMENT connectsto (1)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

]>

XMLOver

moveWorker 0 NORTH

getGameState

**Elvárt kimenet**

true

Workers

0 4

Crates

Maps

1

* + 1. **Trap switch**

**Leírás**

Teszteli, hogy a pályán található lyukakat a hozzájuk tartozó kapcsolóra ládát tolva lehet kapcsolni.

**Ellenőrzött funkcionalitás:**Munkás ládát tol, kapcsoló ládát accept-el, lyuk kinyílik a kapcsoló miatt.

**Hibák amik előfordulhatnak:**Nem nyílik ki a lyuk

**Bemenet**VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (3)

<!ELEMENT type (switch)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (4)

<!ELEMENT type (hole)

<!ELEMENT connectsto (3)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (2)

<!ELEMENT force (0)

]>

XMLOver

ConnectSwitchTo 3 4

moveWorker 0 NORTH

getGameState

**Elvárt kimenet**Hole 4 : open

Crate 1 : moved

true

Workers

0 3

Crates  
1

Maps

1

* + 1. **Friction modification test**

**Leírás**

Teszteli, hogy a worker képes lehelyezni a súrlódást módosító anyagokat.

**Ellenőrzött funkcionalitás:**FrictionModifier elhelyezése.

**Hibák amik előfordulhatnak:**Nem sikerül elhelyezni a FrictionModifiert.

**Bemenet**VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (0)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (1)

]>

XMLOver

putFrictionModifieronTile Oil 1

**Elvárt kimenet**Worker 0 : fmput

* + 1. **Oil teszt**

**Leírás**

Teszteli, hogy egy játékos, aki az erejét éppen meghaladó összsúrlódású dobozt nem tud eltolni, olajos felületen képes erre.

**Ellenőrzött funkcionalitás:**Olaj elhelyezése, munkás ládát tol.

**Hibák amik előfordulhatnak:**Nem sikerül eltolni a ládát még olajos felületen sem.

**Bemenet**VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (3)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (4)

<!ELEMENT type (0)

<!ELEMENT connectsto (3)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (5)

<!ELEMENT type (0)

<!ELEMENT connectsto (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (2)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (2)

<!ELEMENT force (0)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (3)

<!ELEMENT force (0)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (4)

<!ELEMENT force (0)

<!ELEMENT frictionmoifier (tileid, type)>

<!ELEMENT tileid (1)

<!ELEMENT type (oil)

<!ELEMENT frictionmoifier (tileid, type)>

<!ELEMENT tileid (2)

<!ELEMENT type (oil)

<!ELEMENT frictionmoifier (tileid, type)>

<!ELEMENT tileid (3)

<!ELEMENT type (oil)

<!ELEMENT frictionmoifier (tileid, type)>

<!ELEMENT tileid (4)

<!ELEMENT type (oil)

<!ELEMENT frictionmoifier (tileid, type)>

<!ELEMENT tileid (5)

<!ELEMENT type (oil)

]>

XMLOver

moveWorker 0 NORTH

**Elvárt kimenet**Worker 0 : fmput

Crate 3 : moved

Crate 2 : moved

Crate 1 : moved

true

* + 1. **Honey teszt**

**Leírás**

Teszteli, hogy egy játékos az erejét éppen nem meghaladó összsúrlódású dobozt képes eltolni, ragacsos felületen nem képes erre.

**Ellenőrzött funkcionalitás:**Méz elhelyezése, munkás ládát tol.

**Hibák amik előfordulhatnak:**A munkásnak sikerül eltolni a ládákat.

**Bemenet**VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (3)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (4)

<!ELEMENT type (0)

<!ELEMENT connectsto (3)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (5)

<!ELEMENT type (0)

<!ELEMENT connectsto (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (3)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (2)

<!ELEMENT force (0)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (3)

<!ELEMENT force (0)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (4)

<!ELEMENT force (0)

<!ELEMENT frictionmoifier (tileid, type)>

<!ELEMENT tileid (1)

<!ELEMENT type (honey)

<!ELEMENT frictionmoifier (tileid, type)>

<!ELEMENT tileid (2)

<!ELEMENT type (honey)

<!ELEMENT frictionmoifier (tileid, type)>

<!ELEMENT tileid (3)

<!ELEMENT type (honey)

<!ELEMENT frictionmoifier (tileid, type)>

<!ELEMENT tileid (4)

<!ELEMENT type (honey)

<!ELEMENT frictionmoifier (tileid, type)>

<!ELEMENT tileid (5)

<!ELEMENT type (honey)

]>

XMLOver

moveWorker 0 NORTH

**Elvárt kimenet**Worker 0 : fmput

false

* + 1. **Timeout**

**Leírás**

Teszteli, hogy az idő lejártával véget ér a játék.

**Ellenőrzött funkcionalitás:**A pálya lép.

**Hibák amik előfordulhatnak:**Nem csökken a hátralévő idő, nem ér véget a pálya mikor lejár az idő.

**Bemenet**VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (0)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (1)

]>

XMLOver

step

step

step

step

step

step

step

step

step

step

**Elvárt kimenet**

GameOver

* + 1. **Gyilkosság**

**Leírás**

Teszteli, hogy a játékosok megölik egymást láda tolással, és ezzel véget ér a játék.

**Ellenőrzött funkcionalitás:**Munkás ládát tol, láda munkást tol, fal munkást accept-el,munkás meghal.

**Hibák amik előfordulhatnak:**Nem sikerül a mozgás, nem hal meg a munkás akire ládát tolnak.

**Bemenet**VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (3)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (4)

<!ELEMENT type (wall)

<!ELEMENT connectsto (3)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (2)

<!ELEMENT force (0)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (3)

<!ELEMENT force (0)

]>

XMLOver

moveWorker 0 NORTH

getGameState

**Elvárt kimenet**Worker 2 : died

Crate 1 : moved

true

Workers

0 3

Crates

1

Maps

1

* + 1. **Rakodó területre tolás**

**Leírás**

Teszteli, hogy a dobozok rakodó területre tolásával véget ér a játék.

**Ellenőrzött funkcionalitás:**Munkás ládát tol, láda Endtile-ra lép, láda elpusztul, játékos pontokat kap.

**Hibák amik előfordulhatnak:**A láda nem tűnik el mikor a cél mezőre érkezik, a játékos nem kap pontot a ládáért.

**Bemenet**VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (3)

<!ELEMENT type (endtile)

<!ELEMENT connectsto (2)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (2)

<!ELEMENT force (0)

]>

XMLOver

moveWorker 0 NORTH

**Elvárt kimenet**

Crate 1 : onendtile

Crate 1 : died

Crate 1 : moved

true

* + 1. **Pálya betöltése**

**Leírás**

Teszteljük, hogy egy adott pálya fájl beolvasása után megfelelő a memóriába betöltött objektumok struktúrája.

**Ellenőrzött funkcionalitás:**Map/Worker/Crate létrehozása.

**Hibák amik előfordulhatnak:**Map/Worker/Crate nem jön létre.

**Bemenet**VerboseMode

XMLPrepare

<!DOCTYPE map

[

<!ELEMENT map>

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (1)

<!ELEMENT type (0)

<!ELEMENT connectsto (2)

<!ELEMENT tile(id, type, connectsto)>

<!ELEMENT id (2)

<!ELEMENT type (0)

<!ELEMENT connectsto (1)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (worker)

<!ELEMENT tileid (1)

<!ELEMENT force (4)

<!ELEMENT thing(type, tileid, force)>

<!ELEMENT type (crate)

<!ELEMENT tileid (2)

<!ELEMENT force (0)

]>

XMLOver

getGameState

**Elvárt kimenet**Workers

0 4

Crate

1

Maps

1

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

A tesztelést végző segéd- program a specifikált formátumban megírt teszt- fájlokat olvassa be, elindítja az alkalmazást, megadja neki az előírt bemenetet, majd a kapott kimenetet összehasonlítja a megadott elvárt kimenettel, ˝ és amelyik sorban eltérést észlel, ott jelzi a különbséget. Ha nincs eltérés és az stderr-en nem érkezett semmi, a teszt sikeresen lefutott.

* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.04.02. 8:00 | 2 óra | Torma | Tevékenység: Teszt esetek felének elkészítése. |
| 2018.04.03. 10:00 | 2 óra | Pünkösd | Tevékenység:Teszt esetek maradék felének elkészítése. |
| 2018.04.04. 15:00 | 2 óra | Tóth D | Tevékenység: Osztályok leírásainak felének elkészítése. |
| 2018.04.05. 17:00 | 2 óra | Tóth V | Tevékenység: Osztály leírások befejezése. |
| 2018.04.06. 12:00 | 2 óra | Veres | Tevékenység: Dokumentum véglegesítése, lektorálás, nyomtatásra előkészítés. |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Fájl neve** | **Méret** | **Keletkezés ideje** | **Tartalom** |
| src/logic/Commander.java | 9.5 Kbyte | 2018.04.23. 3:33 |  |
| src/logic/Crate.java | 1.1 Kbyte | 2018.03.12. 20:20 | A játékbeli dobozt reprezentáló osztály és implementációja. |
| src/logic/Direction.java | 644 byte | 2018.03.12. 20:20 | A Direction enumeráció, a játékbeli irányok enumerációja. |
| src/logic/EndTile.java | 728 byte | 2018.03.12. 20:20 | A játékbeli dobozok célját reprezentáló osztály és implementációja. |
| src/logic/FrictionModifier.java | 179 byte | 2018.04.23. 3:33 |  |
| src/logic/Game.java | 2.3 Kbyte | 2018.03.12. 20:20 | A játékot reprezentáló osztály és implementációja. |
| src/logic/Hole.java | 1.7 Kbyte | 2018.03.12. 20:20 | A játékbeli lyukakat és csapdákat reprezentáló osztály és implementációja. |
| src/logic/Honey.java | 137 byte | 2018.04.23. 3:33 |  |
| src/logic/Main.java | 373 byte | 2018.03.12. 20:20 | A program indulási pontja, a Main függvényt tartalmazó osztály. |
| src/logic/Map.java | 3.9 Kbyte | 2018.03.12. 20:20 | A játékban lévő pályákat reprezentáló osztály és implementációja. |
| src/logic/Oil.java | 134 byte | 2018.04.23. 3:33 |  |
| src/logic/Steppable.java | 214 byte | 2018.03.12. 20:20 | Interfész, amit minden olyan osztály valósít meg, ami időközönként “lép”, vagyis periodikusan végrehajt valamit. |
| src/logic/Switch.java | 1.5 Kbyte | 2018.03.12. 20:20 | A csapdát kapcsoló “switch”-eket reprezentáló osztály és implementációja. |
| src/logic/Thing.java | 3.5 Kbyte | 2018.03.12. 20:20 | A munkás és doboz közös őse, absztrakt osztály és bizonyos függvények implementációja. |
| src/logic/Tile.java | 2.8 Kbyte | 2018.03.12. 20:20 | A pálya ezekből az elemekből (és speciális változataiból) áll. |
| src/logic/Timer.java | 2.8 Kbyte | 2018.03.12. 20:20 | Időzítő, amely léptet minden a Steppable interfészt megvalósító és a Timernél regisztrált objektumot. |
| src/logic/Wall.java | 443 byte | 2018.03.12. 20:20 | A pályán lévő falakat és oszlopokat realizáló osztály és implementációja. |
| src/logic/Worker.java | 2.2 Kbyte | 2018.03.12. 20:20 | A játékos által irányított munkásokat reprezentáló osztály és implementációja. |
| src/MANIFEST.MF | 47 byte | 2018.03.18. 23:00 | Java manifest file, ami többek közt megmondja, hogy a JAR fileban hol van a main class. |
| testdata/akadalyok.testcase | 912 byte | 2018.04.22. 19:43 | Az "Akadályok" Teszteset bemenete |
| testdata/akadalyok.testcase.expected | 12 byte | 2018.04.22. 19:43 | Az "Akadályok" Teszteset elvárt kimenete |
| testdata/doboz\_falba\_tolas.testcase | 589 byte | 2018.04.22. 19:43 | A "Doboz falba tolás" Teszteset bemenete |
| testdata/doboz\_falba\_tolas.testcase.expected | 6 byte | 2018.04.22. 19:43 | A "Doboz falba tolás" Teszteset elvárt kimenete |
| testdata/friction\_modification\_test.testcase | 299 byte | 2018.04.22. 19:43 | A "Friction modification test" Teszteset bemenete |
| testdata/friction\_modification\_test.testcase.expected | 17 byte | 2018.04.22. 19:43 | A "Friction modification test" Teszteset elvárt kimenete |
| testdata/gyilkossag.testcase | 804 byte | 2018.04.22. 19:43 | A "Gyilkosság" Teszteset bemenete |
| testdata/gyilkossag.testcase.expected | 65 byte | 2018.04.22. 19:43 | A "Gyilkosság" Teszteset elvárt kimenete |
| testdata/honey\_test.testcase | 1.4K byte | 2018.04.22. 19:43 | A "Honey Test" Teszteset bemenete |
| testdata/honey\_test.testcase.expected | 6 byte | 2018.04.22. 19:43 | A "Honey Test"Teszteset elvárt kimenete |
| testdata/lyuk\_vs\_lada.testcase | 609 byte | 2018.04.22. 19:43 | A "Lyuk vs. Láda" Teszteset bemenete |
| testdata/lyuk\_vs\_lada.testcase.expected | 36 byte | 2018.04.22. 19:43 | A "Lyuk vs. Láda" Teszteset elvárt kimenete |
| testdata/lyuk\_vs\_worker.testcase | 421 byte | 2018.04.22. 19:43 | A "Lyuk vs. Worker" Teszteset bemenete |
| testdata/lyuk\_vs\_worker.testcase.expected | 43 byte | 2018.04.22. 19:43 | A "Lyuk vs. Worker" Teszteset elvárt kimenete |
| testdata/oil\_teszt.testcase | 1.4 Kbyte | 2018.04.22. 19:43 | Az "Oil teszt" Teszteset bemenete |
| testdata/oil\_teszt.testcase.expected | 53 byte | 2018.04.22. 19:43 | Az "Oil teszt" Teszteset elvárt kimenete |
| testdata/palya\_betoltese.testcase | 481 byte | 2018.04.22. 19:43 | A "Pálya betöltése" Teszteset bemenete |
| testdata/palya\_betoltese.testcase.expected | 27 byte | 2018.04.22. 19:43 | A "Pálya betöltése" Teszteset elvárt kimenete |
| testdata/rakodo\_teruletre\_tolas.testcase | 592 byte | 2018.04.22. 19:43 | A "Rakodó területre tolás" Teszteset bemenete |
| testdata/rakodo\_teruletre\_tolas.testcase.expected | 41 byte | 2018.04.22. 19:43 | A "Rakodó területre tolás" Teszteset elvárt kimenete |
| testdata/szimpla\_doboz\_tolas.testcase | 586 byte | 2018.04.22. 19:43 | A "Szimpla doboz tolás" Teszteset bemenete |
| testdata/szimpla\_doboz\_tolas.testcase.expected | 21 byte | 2018.04.22. 19:43 | A "Szimpla doboz tolás" Teszteset elvárt kimenete |
| testdata/timeout.testcase | 267 byte | 2018.04.22. 19:43 | A "Timeout" Teszteset bemenete |
| testdata/timeout.testcase.expected | 9 byte | 2018.04.22. 19:43 | A "Timeout" Teszteset elvárt kimenete |
| testdata/tobb\_doboz\_tolas.testcase | 988 byte | 2018.04.22. 19:43 | A "Több doboz tolás" Teszteset bemenete |
| testdata/tobb\_doboz\_tolas.testcase.expected | 53 byte | 2018.04.22. 19:43 | A "Több doboz tolás" Teszteset elvárt kimenete |
| testdata/trap\_switch.testcase | 725 byte | 2018.04.22. 19:43 | A "Trap Switch" Teszteset bemenete |
| testdata/trap\_switch.testcase.expected | 47 byte | 2018.04.22. 19:43 | A "Trap Switch" Teszteset elvárt kimenete |
| testdata/tul\_sok\_doboz\_tolas.testcase | 988 byte | 2018.04.22. 19:43 | A "Túl sok doboz tolás" Teszteset bemenete |
| testdata/tul\_sok\_doboz\_tolas.testcase.expected | 6 byte | 2018.04.22. 19:43 | A "Túl sok doboz tolás" Teszteset elvárt kimenete |
| testdata/workerek\_dobozzal\_tolasa.testcase | 788 byte | 2018.04.22. 19:43 | A "Workerek dobozza tolása" Teszteset bemenete |
| testdata/workerek\_dobozzal\_tolasa.testcase.expected | 21 byte | 2018.04.22. 19:43 | A "Workerek dobozza tolása" Teszteset elvárt kimenete |
| testdata/workerek\_mozgasa.testcase | 374 byte | 2018.04.22. 19:43 | A "Workerek mozgása" Teszteset bemenete |
| testdata/workerek\_mozgasa.testcase.expected | 5 byte | 2018.04.22. 19:43 | A "Workerek mozgása" Teszteset elvárt kimenete |
| testdata/zart\_lyuk\_teszt.testcase | 401 byte | 2018.04.22. 19:43 | A "Zárt lyuk teszt" Teszteset bemenete |
| testdata/zart\_lyuk\_teszt.testcase.expected | 31 byte | 2018.04.22. 19:43 | Az "Zárt lyuk teszt" Teszteset elvárt kimenete |
| test.bat | 663 byte | 2018.04.22. 19:43 | A tesztelés automatizálására szolgáló script. (Windows) |
| test.sh | 2.3 Kbyte | 2018.04.22. 19:43 | A tesztelés automatizálására szolgáló script. (Linux) |
| files.txt | 396 byte | 2018.04.22. 20:13 | A fordításhoz szükséges fájlok. |
| options.txt | 75 byte | 2018.03.18. 0:45 | A fordításhoz szükséges opciók. |
| build.bat | 381 byte | 2018.03.18. 0:45 | A JAR file buildelés automatizálására szolgáló script. (Windows) |
| build.sh | 382 byte | 2018.03.18. 0:45 | A JAR file buildelés automatizálására szolgáló script. (Linux) |

* + 1. **Fordítás**

A fordításhoz szükséges környezet: Microsoft Windows vagy GNU/Linux operációs rendszer illetve Java JDK 9-es verzió.

A fordítás, és JAR fájl buildelés megkönnyítésére mellékeltünk egy buildelő scriptet mind Windowsra, mind Linuxra.

**Buildelés Windows alatt:**

Kattintsunk duplán a build.bat állományra.

**Buildelés Linux alatt:**

Tegyük futtathatóvá a build.sh állományt, majd futtassuk a ./build.sh paranccsal abból a mappából amiben van.

**Troubleshooting:**

Ha a script lefutott, de nem jelent meg a JAR file, akkor ellenőrízzük, hogy fennvan-e a javac 1.9-es verziója (JDK9), illetve, hogy a megfelelő mappából futtatuk-e. Illetve, hogy a fordításhoz szükséges binárisok mappája benne van-e a PATH környezeti változónkban.

**Fordítás Java 8-cal**

Ha nem elérhető a Java 9 a számítógépünkön, akkor lefordíthatjuk/futtathatjuk Java 8-cal is. Ám ebben az esetben nem tudjuk biztosítani, hogy a program pontosan úgy fut, mint ahogy terveztük. Ha Java 8-cal szeretnénk fordítani, akkor az ***options.txt*** állományban módosítsuk a **-source és -target** argumentumok után az *1.9-et 1.8-ra*, mentünk és próbáljuk újra a fordítást.

* + 1. **Futtatás**

A lefordított JAR fájlt a test.bat parancssorból történő indításával futtathatjuk.

* 1. ***Tesztek jegyzőkönyvei***
     1. **Workerek mozgása**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Dániel |
| **Teszt időpontja** | 2014.04.20 |

* + 1. **Akadályok**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Dániel |
| **Teszt időpontja** | 2014.04.20 |

* + 1. **Szimpla doboz tolás**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Dániel |
| **Teszt időpontja** | 2014.04.20 |

* + 1. **Több doboz tolás**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Dániel |
| **Teszt időpontja** | 2014.04.20 |

* + 1. **Túl sok doboz tolás**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Dániel |
| **Teszt időpontja** | 2014.04.20 |

* + 1. **Workerek dobozzal tolása**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Dániel |
| **Teszt időpontja** | 2014.04.21 |

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Dániel |
| **Teszt időpontja** | 2014.04.21 |
| **Teszt eredménye** | Nem tudta eltolni. |
| **Lehetséges hibaok** | Túl nehéz volt a munkás. |
| **Változtatások** | Súlyértékek újraszámolása. |

* + 1. **Doboz falba tolás**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Dániel |
| **Teszt időpontja** | 2014.04.21 |

* + 1. **Lyuk vs. Worker**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Dániel |
| **Teszt időpontja** | 2014.04.21 |

* + 1. **Lyuk vs. Láda**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Dániel |
| **Teszt időpontja** | 2014.04.21 |

* + 1. **Zárt lyuk teszt**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Vince |
| **Teszt időpontja** | 2014.04.21 |

* + 1. **Trap switch**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Vince |
| **Teszt időpontja** | 2014.04.21 |

* + 1. **Friction modification test**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Vince |
| **Teszt időpontja** | 2014.04.21 |

* + 1. **Oil teszt**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Vince |
| **Teszt időpontja** | 2014.04.21 |

* + 1. **Honey teszt**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Vince |
| **Teszt időpontja** | 2014.04.21 |

* + 1. **Timeout**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Vince |
| **Teszt időpontja** | 2014.04.22 |

* + 1. **Gyilkosság**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Vince |
| **Teszt időpontja** | 2014.04.22 |

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Vince |
| **Teszt időpontja** | 2014.04.21 |
| **Teszt eredménye** | Nem halt meg a worker. |
| **Lehetséges hibaok** | Túl nehéz volt a láda, nem tudta megmozdítani. |
| **Változtatások** | Fizikai jellemzők átváltoztatása. |

* + 1. **Rakodó területre tolás**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Vince |
| **Teszt időpontja** | 2014.04.22 |

* + 1. **Pálya betöltése**

|  |  |
| --- | --- |
| **Tesztelő neve** | Tóth Vince |
| **Teszt időpontja** | 2014.04.22 |

* 1. ***Értékelés***

|  |  |
| --- | --- |
| **Tag neve** | **Munka százalékban** |
| Torma Kristóf | 20% |
| Veres Csaba Miklós | 20% |
| Pünkösd Marcell | 20% |
| Tóth Dániel | 20% |
| Tóth Vince | 20% |

* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.04.13. 18:00 | 5 óra | Torma | Tevékenység: Teszt olvasó véglegesítése. |
| 2018.04.14. 20:00 | 5 óra | Veres | Tevékenység: Dokumentálás |
| 2018.04.15. 15:00 | 5 óra | Pünkösd | Tevékenység: Teszt bemenetek szerkesztése |
| 2018.04.16. 21:00 | 5 óra | Tóth D | Tevékenység: Tesztelés és javítás |
| 2018.04.17. 22:00 | 5 óra | Tóth V | Tevékenység: Tesztelés és javítás |

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

Mint látható, a játék

***11.1.1 Látványterv***

|  |  |
| --- | --- |
|  | Standby állapot  Megjegyzés: A szürke/fehér négyzetrács csak négyzetek méretének szemléltetése végett vannak |
|  | Legördülő menü |
|  | Játék közben |

***11.1.2 Grafikus elemek***

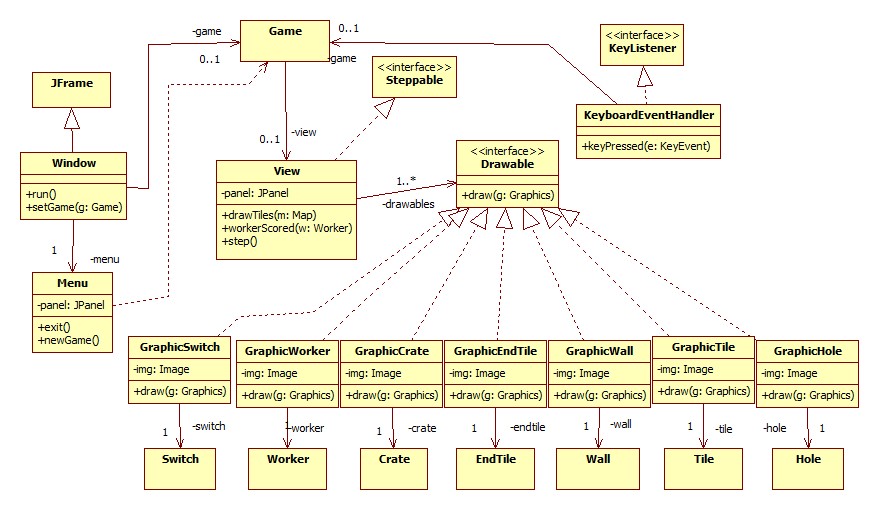
|  |  |
| --- | --- |
|  | Padló elem |
|  | Fal elem |
|  | Cél elem |
|  | Kapcsoló |
|  | Tolható doboz |
|  | Lyuk Megjegyzés: Zárt állapotban úgy néz ki mint a Padló elem |
|  | Kék és piros worker |

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

A grafikus felület megvalósításánál törekedtünk az MVC architektúra megvalósítására. A megjelenítés push elvre épül. Minden modellben létező objektumnak van egy grafikus párja ami megvalósítja a **Drawable** interface-t. A megjelenítésért a **View** osztály felel. Ez tárolja a **Drawable** osztályokat. Ha a modellben valami változás történik akkor a modellbeli objektum erről értesíti a **View** osztályt ami újrarajzoltatja a megváltozott objektumhoz tartozó **Drawable** grafikus objektumot. Ezek mind eltárolják a **Game**-ben lévő párjukat így lekérdezik attól a változást így könnyedén kirajzolhatják magukat a megváltozott helyzetben. A pályán lévő **Tile**-ok kirajzolása azonban nem push alapon működik. A **Map** létrejötténél a **View** a **Map** osztály alapján kirajzolja az összes **Tile**-t. Ezután azonban a **Worker**ek és **Crate**k kirajzolása, eltüntetése, mozgása már végig esemény vezérelt módon működik. A bemenetet a **KeyboardEventHandler** osztály kezeli ez megvalósítja a **KeyListener** interface szükséges metódusait. A lenyomott gomb alapján kezeli a hozzá tartozó **Game** objektumot.

A játék indulását a **Menu** osztály kezeli, ezt lehet használni új játék indítására valamint kilépésre.

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

**

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

A játék egészét reprezentáló osztály.

* + - * **Ősosztályok**

*-*

* + - * **Interfészek**

*-*

* + - * **Attribútumok**
* **-view :View:**Egy referencia a grafikus megjelenítést végző osztályra
  + - * **Metódusok**

**-**

* + 1. **KeyBoardEventHandler**
       - **Felelősség**

*Ez az osztály felelős a bemenet kezelésért.*

* + - * **Ősosztályok**

*-*

* + - * **Interfészek**

*KeyListener*

* + - * **Attribútumok**

**-**

* + - * **Metódusok**
* **+ KeyPressed(e: KeyEvent):** Kezeli a lenyomott billentyűket.
  + 1. **Window**
       - **Felelősség**

*Egy ablak ami azért felelős hogy megjelenítse a játék menüjét és a játékot..*

* + - * **Ősosztályok**

*JFrame*

* + - * **Interfészek**

*-*

* + - * **Attribútumok**
* **-game:Game:** Ezt a játékot indítja el.
* **-menu :Menu:**Ezt a menüt jeleníti meg.
  + - * **Metódusok**
* **+ run():** Megjeleníti a menüt.
* +**setGame(g: Game):** Beállítja a game attribútum értékét.
  + 1. **Menu**
       - **Felelősség**

*A játék főmenüjét megvalósító osztály.*

* + - * **Ősosztályok**

*-*

* + - * **Interfészek**

*-*

* + - * **Attribútumok**
* **# panel:Jpanel :** A menü megjelenítéséhez használt JPanel.
  + - * **Metódusok**
* **+ exit():** Kilép a játékból és bezárja az ablakokat.
* +**newGame():** Elindít egy új játékot.
  + 1. **View**
       - **Felelősség**

*A kapcsoló kirajzolása.*

* + - * **Ősosztályok**

*-*

* + - * **Interfészek**

*Steppable*

* + - * **Attribútumok**
* **- panel:Jpanel :** Egy JPanel amire kirajzoljuk a játékot.
* -drawables:Drawable[]: Az összes kirajzolható objektumot tartalmazó lista.
  + - * **Metódusok**
* **+ DrawTiles(m: Map):** Kirajzolja a paraméterül kapott Map mezőit és a tartalmuk.
* **+workerScored(w: Worker):** Frissíti a játékos pontszámát.
* **+step():** A view lép.
  + 1. **Drawable**
       - **Felelősség**

*Egy interface a kirajzolható objektumok számára.*

* + - * **Ősosztályok**

*-*

* + - * **Interfészek**

*-*

* + - * **Attribútumok**

**-**

* + - * **Metódusok**
* **+ Draw(Graphics):** Egy objektum kirajzolásáért felelős függvény.
  + 1. **GraphicHole**
       - **Felelősség**

*Egy lyuk kirajzolása.*

* + - * **Ősosztályok**

*-*

* + - * **Interfészek**

*Drawable*

* + - * **Attribútumok**
* **# img:Image :** A kirajzolandó lyuk képe.
* **-hole: Hole:** A kirajzolandó lyuk.
  + - * **Metódusok**
* **+ Draw(Graphics):** Kirajzolja a lyukat.
  + 1. **GraphicTile**
       - **Felelősség**

*Egy egyszerű mező kirajzolása.*

* + - * **Ősosztályok**

*-*

* + - * **Interfészek**

*Drawable*

* + - * **Attribútumok**
* **- img:Image :** A kirajzolandó mező képe.
* **-tile: Tile:** A kirajzolandó mező.
  + - * **Metódusok**
* **+ Draw(Graphics):** Kirajzolja a mezőt.
  + 1. **GraphicWall**
       - **Felelősség**

*A fal kirajzolása.*

* + - * **Ősosztályok**

*-*

* + - * **Interfészek**

*Drawable*

* + - * **Attribútumok**
* **- img:Image :** A kirajzolandó fal képe.
* **-wall: Wall:** A kirajzolandó fal.
  + - * **Metódusok**
* **+ Draw(Graphics):** Kirajzolja a falat.
  + 1. **GraphicEndTile**
       - **Felelősség**

*A cél mező kirajzolása.*

* + - * **Ősosztályok**

*-*

* + - * **Interfészek**

*Drawable*

* + - * **Attribútumok**
* **- img:Image :** A kirajzolandó cél mező képe.
* **-endtile: EndTile:** A kirajzolandó cél mező.
  + - * **Metódusok**
* **+ Draw(Graphics):** Kirajzolja a cél mezőt.
  + 1. **GraphicCrate**
       - **Felelősség**

*A doboz kirajzolása.*

* + - * **Ősosztályok**

*-*

* + - * **Interfészek**

*Drawable*

* + - * **Attribútumok**
* **- img:Image :** A kirajzolandó doboz képe.
* **crate: Crate:** A kirajzolandó doboz.
  + - * **Metódusok**
* **+ Draw(Graphics):** Kirajzolja a dobozt.
  + 1. **GraphicWorker**
       - **Felelősség**

*A munkás kirajzolása.*

* + - * **Ősosztályok**

*-*

* + - * **Interfészek**

*Drawable*

* + - * **Attribútumok**
* **- img:Image :** A kirajzolandó munkás képe.
* **-worker: Worker:** A kirajzolandó munkás.
  + - * **Metódusok**
* **+ Draw(Graphics):** Kirajzolja a munkást.
  + 1. **GraphicSwitch**
       - **Felelősség**

*A kapcsoló kirajzolása.*

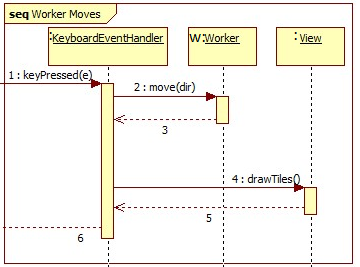
* + - * **Ősosztályok**

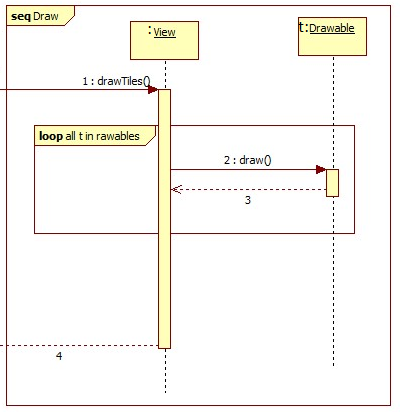
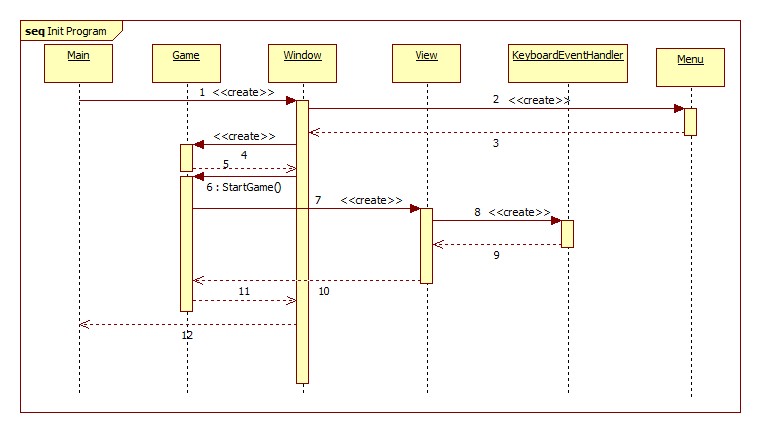
*-*

* + - * **Interfészek**

*Drawable*

* + - * **Attribútumok**
* **- img:Image :** A kirajzolandó kapcsoló képe.
* **-switch: Switch:** A kirajzolandó kapcsoló.
  + - * **Metódusok**
* **+ Draw(Graphics):** Kirajzolja a kapcsolót.
  1. ***Kapcsolat az alkalmazói rendszerrel***





* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.04.29. 10:00 | 2 óra | Torma | Tevékenység: Grafikus Interfész rész elkészítése |
| 2018.04.30. 10:00 | 2 óra | Pünkösd | Tevékenység: Grafikus interfész architektúrájának elkészítése |
| 2018.05.01. 11:00 | 2 óra | Tóth D | Tevékenység: Grafikus objektumok felsorolása |
| 2018.05.01. 12:00 | 2 óra | Tóth V | Tevékenység: Kapcsolat az alkalmazói rendszerrel elkészítése |
| 2018.05.01. 12:00 | 2 óra | Veres | Tevékenység: Hibák javítása, pontosítás, dokumentum elkészítése |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Fájl neve** | **Méret** | **Keletkezés ideje** | **Tartalom** |
| Crate.java | 1640 btye | 2018.03.12. 20:20 | A játékbeli dobozt reprezentáló osztály és implementációja. |
| Direction.java | 644 byte | 2018.03.12. 20:20 | A Direction enumeráció, a játékbeli irányok enumerációja. |
| EndTile.java | 1366 byte | 2018.03.12. 20:20 | A játékbeli dobozok célját reprezentáló osztály és implementációja. |
| FrictionModifier.java | 482 byte | 2018. 04. 07. | A súrlódási együtthatót jelképező ozstály. |
| Game.java | 3349byte | 2018.03.12. 20:20 | A játékot reprezentáló osztály és implementációja. |
| Hole.java | 2323 byte | 2018.03.12. 20:20 | A játékbeli lyukakat és csapdákat reprezentáló osztály és implementációja. |
| Honey.java | 396 byte | 2018. 04. 07. | A mézet reprezentáló osztály. |
| Main.java | 967 byte | 2018.03.12. 20:20 | A program indulási pontja, a Main függvényt tartalmazó osztály. |
| Map.java | 3992 byte | 2018.03.12. 20:20 | A játékban lévő pályákat reprezentáló osztály és implementációja. |
| Oil.java | 389 byte | 2018. 04. 07. | Az olajat reprezentáló osztály. |
| Steppable.java | 266 byte | 2018.03.12. 20:20 | Interfész, amit minden olyan osztály valósít meg, ami időközönként “lép”, vagyis periodikusan végrehajt valamit. |
| Switch.java | 2691 byte | 2018.03.12. 20:20 | A csapdát kapcsoló “switch”-eket reprezentáló osztály és implementációja. |
| Thing.java | 5840 byte | 2018.03.12. 20:20 | A munkás és doboz közös őse, absztrakt osztály és bizonyos függvények implementációja. |
| Tile.java | 3850 byte | 2018.03.12. 20:20 | A pálya ezekből az elemekből (és speciális változataiból) áll. |
| Timer.java | 4085 byte | 2018.03.12. 20:20 | Időzítő, amely léptet minden a Steppable interfészt megvalósító és a Timernél regisztrált objektumot. |
| Wall.java | 1055 byte | 2018.03.12. 20:20 | A pályán lévő falakat és oszlopokat realizáló osztály és implementációja. |
| Worker.java | 4257 byte | 2018.03.12. 20:20 | A játékos által irányított munkásokat reprezentáló osztály. |
| MANIFEST.mf | 47 byte | 2018.03.18. 23:00 | Java manifest file, ami többek közt megmondja, hogy a JAR fileban hol van a main class. |
| LevelElements.java | 396 byte | 2018.05.13. | Enumeráció ami a pályák beolvasásánál használható. |
| LevelLoader,java | 3657 byte | 2018.05.13. | A pálya betöltéséért felel. |
| LevelStorage.java | 938 byte | 2018.05.13. | Pályákat tárol. |
| AbstractGraphic.java | 612 byte | 2018.05.13. | A grafikai elemek implementációját segíti. |
| Disappearable.java | 129 byte | 2018.05.13. | Egy interfész ami biztosítja a dissapear metódust. |
| Drawable.java | 198 byte | 2018.05.13. | Interfész a kirajzolható objektumoknak. |
| GraphicCrate.java | 1468 byte | 2018.05.13. | A doboz kirajzolásáért felelős osztály. |
| GraphicEndTile.java | 2361 byte | 2018.05.13. | A cél mező kirajzolásáért felelős osztály. |
| GraphicHole.java | 2422 byte | 2018.05.13. | A lyuk kirajzolásáért felelős osztály. |
| GraphicSwitch.java | 2524 byte | 2018.05.13. | A kapcsoló kirajzolásáért felelős osztály. |
| GraphicTile.java | 2309 byte | 2018.05.13. | A mező kirajzolásáért felelős osztály. |
| GraphicWall.java | 1494 byte | 2018.05.13. | A fal kirajzolásáért felelős osztály. |
| GraphicWorker.java | 2309 byte | 2018.05.13. | A munkás kirajzolásáért felelős osztály. |
| IconCollection.java | 4124 byte | 2018.05.13. | A játékban használt képeket tároló osztály. |
| SokobanGui.java | 11648 byte | 2018.05.13. | A Gui megjelenítésért felelős. |
| build.sh | 362 Bytes | 2018.03.18. 0:45 | A JAR file buildelés automatizálására szolgáló script. |
| build.bat | 355 Bytes | 2018.03.18. 0:45 | A JAR file buildelés automatizálására szolgáló script. |
| options.txt | 80 bytes | 2018.03.18. 0:45 | A fordításhoz szükséges opciók. |
| files.txt | 835 byte | 2018. 03. 18. | A fordítandó fájlok listája. |
| funlevel.txt | 144 byte | 2018. 05. 13 | Egy demo pályát tartalmazó fájl a betöltés tesztelésére. |
| doboz.png | 1 281 bytes | 2018 05.01. | A dobozokat megjelenítő ikon. |
| floor.png | 5 076 bytes | 2018 05.01. | A mezőt ábrázoló kép. |
| Wall.png | 19 934 bytes | 2018.05.01. | A fal ikonja. |
| EasterEggBlueLeft.png | 22636 byte | 2018. 05. 13. | A kék munkás balra fordulva. |
| EasterEggBlueRight.png | 22642 byte | 2018. 05. 13. | A kék munkás jobbra fordulva. |
| EasterEggRedRight.png | 22998 byte | 2018. 05. 13. | A piros munkás balra fordulva. |
| EasterEggRedRight.png | 22998 byte | 2018. 05. 13. | A piros munkás jobbra fordulva. |
| Floor\_with\_ Button.png | 18 308 bytes | 2018.05.13. | A mezőn lévő kapcsoló képe. |
| Floor\_with\_ HoleOpen.png | 17 890 bytes | 2018.05.13. | A lyukat ábrázoló kép. |
| Floor\_with\_ Target.png | 21 418 bytes | 2018.05.13. | A cél mezőt ábrázoló ikon. |
| Floor\_with\_ Honey.png | 22 545 bytes | 2018.05.13. | Mézes mező képe. |
| Floor\_with\_ Oil.png | 22 016 bytes | 2018.05.13. | Olajos mező képe. |
| Floor\_with\_ Target\_with\_ Honey.png | 22 879 bytes | 2018.05.13. | Mézes cél mező képe. |
| Floor\_with\_ Target\_with\_ Oil.png | 22 822 bytes | 2018.05.13. | Olajos cél mező képe. |
| Floor\_with\_ Button\_with\_ Honey.png | 22 545 bytes | 2018.05.13. | Mézes kapcsoló képe. |
| Floor\_with\_  Button\_with\_  Oil.png | 22 467 bytes | 2018.05.13. | Olajos kapcsoló képe. |
| HoneyJar.png | 19 271 bytes | 2018.05.13. | A játék ikonja. |

* + 1. **Fordítás és telepítés**

A fordításhoz szükséges környezet: Microsoft Windows vagy GNU/Linux operációs rendszer illetve Java JDK 9-es verzió.

A fordítás, és JAR fájl buildelés megkönnyítésére mellékeltünk egy buildelő scriptet mind Windowsra, mind Linuxra.

**Buildelés Windows alatt:**

Kattintsunk duplán a build.bat állományra.

**Buildelés Linux alatt:**

Tegyük futtathatóvá a build.sh állományt, majd futtassuk a ./build.sh paranccsal abból a mappából amiben van.

**Troubleshooting:**

Ha a script lefutott, de nem jelent meg a JAR file, akkor ellenőrízzük, hogy fennvan-e a javac 1.9-es verziója (JDK9), illetve, hogy a megfelelő mappából futtatuk-e. Illetve, hogy a fordításhoz szükséges binárisok mappája benne van-e a PATH környezeti változónkban. Amennyiben ez sem megy, szükség van egy új Eclipse projekt elkészítésébe, majd az ansiliary, gui és logic csomagok létrehozására és az oda tartozó forrásfileok importálásába. Figyelni kell, hogy az assets mappa a projekt főkönyvátárban legyen.

**Fordítás Java 8-cal**

Ha nem elérhető a Java 9 a számítógépünkön, akkor lefordíthatjuk/futtathatjuk Java 8-cal is. Ám ebben az esetben nem tudjuk biztosítani, hogy a program pontosan úgy fut, mint ahogy terveztük. Ha Java 8-cal szeretnénk fordítani, akkor az ***options.txt*** állományban módosítsuk a **-source és -target** argumentumok után az *1.9-et 1.8-ra*, mentünk és próbáljuk újra a fordítást.

* + 1. **Futtatás**

A lefordított JAR fájlt a

|  |
| --- |
| java -jar KillerSokoban.jar |

paranccsal futtathatjuk. Figyelni kell, hogy amennyiben nem IDE-vel történt a fordítás, az assets mappa és a .jar file egy könyvtárban legyenek.

* 1. ***Értékelés***

|  |  |
| --- | --- |
| **Tag neve** | **Munka százalékban** |
| Tóth Dániel Péter | 20% |
| Pünkösd Marcell | 20% |
| Tóth Vince | 20% |
| Torma Kristóf | 20% |
| Veres Csaba Miklós | 20% |

* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2018.03.07. 21:00 | 4 óra | Pünkösd | Tevékenység: Logika előkészítése GUI-val interfészelésre. |
| 2018.03.08. 10:00 | 4 óra | Tóth D. | Tevékenység: GUI elkészítése. |
| 2018.03.09. 16:00 | 4 óra | Veres | Tevékenység: GUI logika leprogramozása. |
| 2018.03.10. 22:00 | 4 óra | Tóth V. | Tevékenység: Program tesztelése, hibák javítása |
| 2018.03.11. 21:00 | 4 óra | Torma | Tevékenység: Kód véglegesítése, dokumentum összeállítása, beadás. |

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

|  |  |
| --- | --- |
| **Tag neve** | **Munkaidő (óra)** |
| *Tóth Dániel Péter* | 26 |
| *Pünkösd Marcell* | 26 |
| *Tóth Vince* | 26 |
| *Torma Kristóf* | 26 |
| *Veres Csaba Miklós* | 26 |
| **Összesen** | 130 |

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

|  |  |
| --- | --- |
| **Fázis** | **Kódsorok száma** |
| Szkeleton | 1616 |
| Prototípus | 1428 |
| Grafikus változat | 3029 |
| **Összesen** | 6073 |

* 1. ***• Projekt összegzés***

**Tóth Dániel Péter**

* + 1. **Mit tanultak a projektből konkrétan és általában?**

Azt, hogy a pontos és jól formázott munka fontosabb mint gondolnánk. A bugok és furcsa jelenségek elsődleges forrása. Így a csapatmunka hátrányait és előnyeit is realizáltam.

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

A legnehezebb a munka idejét megbecsülni, sokszor későn vettük észre a hátralévő tennivalókat és éjszakázás lett belőle.

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

Általában igen.

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

A főleg ismétlődő dokumentációs munkáknál amikkel sok munka volt.

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

Több kódolós részt belevinni a munkatervbe már az elején, hogy ne csak a szorgalmi időszak vége felé maradjon a kódolós és a modellezős munka nagyrésze mikor a többi tárgyból is terhelés van rajtunk.

* + 1. **Milyen feladatot ajánlanának a projektre?**

Forgalom irányitást szimulátort leegyszerűsítve (autókra, de vonatokra is kivitelezhető vágányokkal).

* + 1. **Egyéb kritika és javaslat**

Azt a megkötést, hogy csak a bentlévő gépeken és az azon elérhető dolgokkal lehessen futtatni, egy kicsit lehetne megengedőbbé tenni, hogy az esetlegesen ott felmerülő gondok ne legyen drasztikusak, előfordult az, hogy alapvető alapokat igénylő dolgok nem működtek.

**Pünkösd Marcell**

* + 1. **Mit tanultak a projektből konkrétan és általában?**

Megtanultam, hogy egy jól kigondolt modell és ahhoz társított dokumentáció nagyban segíti a programozói munkát. Illetve ha csapatban dolgozunk rajta, akkor az összetettebb feladatok sem olyan ijesztőek.

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

A legnehezebb talán az volt, hogy hétközben is rávegyük magunkat a munkára, hogy ne az utolsó este kelljen összerakni mindent. Emellett a könnyebb volt leprogramozni az elkészült modellt, viszont kidolgozni azt úgy, hogy tökéletes legyen, szintén nehezebb.

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

Nagyjából igen. A GUI elkészítése például nehézséget okozott, arra kellett volna még idő.

* + 1. **Ha nem, akkor hol okozott ez nehézséget?**
    2. **Milyen változtatási javaslatuk van?**

Jó lenne, ha amennyiben a programozás közben bukik ki, hogy rossz a modell agilisabban lehetne változtatni a modellen.

* + 1. **Milyen feladatot ajánlanának a projektre?**

Egy Worms-féle játék szerintem teljes mértékben passzol a tárgyhoz. Esetleg valami (nem feltétlen játék) ami hálózaton kommunikál (lehetőség szerint egy már bevált protokollon, mivel ennek kifejlesztése már bonyolultabb feladat). Mindig izgalmas amikor két gép kommunikálni kezd.

* + 1. **Egyéb kritika és javaslat**

Érdemes lenne átgondolni azt a kikötést, hogy nem lehet módosítani, csak az előző heti dokumentumon.

Illetve a dokumentáció leadásnak legalább egy része (ha olyan van, hogy kisebb jelentősségű a dokumentum) mehetne elektronikusan, sajnos láttam csapatokat akik utolsó pillanatban még nyomtatni rohangáltak, és egyszer mi is majdnem beleestünk ebbe.

Továbbá azon a kikötésen mindenképpen finomítanék, hogy csak a HSZK gépeken fordított és ott futtatott programot lehet demozni, Sajnos előfordult, hogy az ottani gépek nem megfelelően voltak konfigurálva (rossz JDK verzió, hijányos vagy rosszul beállított PATH) így nem tudtuk rendesen lefordítani a programunkat, vagy éppen a teszt nem futott le, miközben nálunk, mindanyiunk gépen sikeresen lefutott minden. Így elég sok kellemetlen helyzetet meg tudnánk úszni.

**Tóth Vince**

* + 1. **Mit tanultak a projektből konkrétan és általában?**

Konkrétan és általában is, a kommunikáció fontosságát, hogy egy 5 fős csapat ne 5 felé dolgozzon.

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

A legkönnyebb a feladatok kiosztása illetve azok megértése volt, leginkább a felelősségteljes és naprakész csapatkapitánynak köszönhetően. A legnehezebb pedig átlátni és hozzászokni mások munkájához, a különbségeket áthidalni.

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

A pontszámokkal igen a feladat nehézségével nem mindíg állt összhangban. Megesett, hogy nemtervezett éjszakázás lett belőle.

* + 1. **Ha nem, akkor hol okozott ez nehézséget?**
    2. **Milyen változtatási javaslatuk van?**

Esetleg változó időt adni különböző feladatokra illetve a nagyon repetatív részek elhagyásával megspórolni a papírt.

* + 1. **Milyen feladatot ajánlanának a projektre?**

Galaga, vagy esetleg valamilyen RPG jellegű egyszerűbb játék mint a Pokemon.

* + 1. **Egyéb kritika és javaslat**

Lehetne máskor a beadás ideje, a vidékieknek akiknek nincs mobilis gépe, nehézséget okoz néha amikor haza utaznának.

**Torma Kristóf**

* + 1. **Mit tanultak a projektből konkrétan és általában?**

Leginkább a soft skilljeim fejlődtek a tárgy kapcsán. Megtanultam, hogy kell kézben tartani egy csapatot úgy, hogy a munka is dübörögjön, de a hangulat is jó legyen. A dokumentáció fáj, de hasznos.

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

A legnehezebb az volt, hogy úgy kellett kitalálni a modellt (és a GUI modelljét), hogy az majd működjön leprogramozva. Miután ez megvolt a programozás része nyilván triviális volt.

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

Összességében igen, de hetekre lebontva nem mindig. Vannak olyan hetek, amikor alig kell foglalkozni a projekttel, de olyanok is, amikor annyi a munka, hogy nem látunk ki belőle.

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

A modellezős részek kicsit húzósak voltak, valamint a GUI-t elkészíteni a modell simítgatása után olyan volt, mintha a Mount Everestre akarnánk felbiciklizni. Verseny biciklivel.

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

Érdemes lenne átállni RUP-ról valamilyen agilisebb módszerre, hogy ne kösse a modell ennyire keményen a kezét a csapatnak, ugyanis megmenteni egy elrontott modellt lehetetlen. Ez kifejezetten akkor probléma, ha a modell úgy hibás, hogy a labvez sem veszi észre. Szerencsére velünk ilyen nem fordult elő, mert a labvezünk kifejezetten igényes volt, de voltak olyan csoportok, ahol ez prevalens volt, sajnos.

* + 1. **Milyen feladatot ajánlanának a projektre?**

Szerintem nem lenne rossz ötlet egy word processor elkészítése. Nem feltétlen ragaszkodnék a játékok témaköréhez.

* + 1. **Egyéb kritika és javaslat**

Szerintem a legfontosabb az lenne, ha a tárgy át lenne hangolva egy agilis megközelítésre.

**Veres Csaba Miklós**

* + 1. **Mit tanultak a projektből konkrétan és általában?**

A modellezés időigényességét és a pontos modellalkotás nehézségét, a megfelelő modellező programok használata is sok gyakorlatot adott. Továbbá hasznos volt csapatban dolgozni és megtanulni alkalmazkodni a többiekhez.

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

A legnehezebb a modellt és a kódolást összeigazítani a néha hanyagul elkészített modell miatt. A legkönnyebb része a csapat munkára fogása mivel szerencsésen megbízható csapattársakat találtam.

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

Igen.

* + 1. **Ha nem, akkor hol okozott ez nehézséget?**
    2. **Milyen változtatási javaslatuk van?**

Néha nehéz megállapítani mit kér a feladatkiírás egy hétre, azokat lehetne pontosítani, hogy ne az előző évek anyagaira kelljen hagyatkozni mikor valami nem tiszta.

* + 1. **Milyen feladatot ajánlanának a projektre?**

Kicsit AI-os akvárium app például, halakkal interakcióval, hogy ne csak modellezésnél kelljen gondolkodni hanem a feladat (program) megoldásán is.

* + 1. **Egyéb kritika és javaslat**

Sajnos a tárgy egy félévben van a grafikával, ezt figyelembe véve szerintem a terhelés-elosztáson kicsit változtatni kéne.