**Összesített dokumentáció**

30 – ez\_mode

Konzulens:

Koppa Dániel

Csapattagok

|  |  |  |
| --- | --- | --- |
| Nyilas Péter | Y8QBCS | [nyilaspeter001@gmail.com](mailto:nyilaspeter001@gmail.com) |
| Farkas Bence | D6AE9F | [bencefarkas27@gmail.com](mailto:bencefarkas27@gmail.com) |
| Zsigmond Botond | MV7M6D | [zsigmond.botond.9@gmail.com](mailto:zsigmond.botond.9@gmail.com) |
| Sőregi Larina | M4B9MC | [soregi.larina2001@gmail.com](mailto:soregi.larina2001@gmail.com) |
| Kis Benedek M. | JOYAXJ | [kisbeni2003@gmail.com](mailto:kisbeni2003@gmail.com) |

#### Követelmény, projekt, funkcionalitás

###### Bevezetés

* + 1. Cél

Ez a dokumentum bemutatja a feladatot, a projekt követelményeit, és hogy mi hogyan fog működni.

* + 1. Szakterület

Ez egy többszemélyes számítógépes játék megvalósítása, melyet két csapat játszik egymás ellen.

* + 1. Definíciók, rövidítések

**ill.** = illetve

**stb.**: satöbbi

**JDK**: Java Development Kit

**min.**: minimum

**random**: véletlenszerű

* + 1. Hivatkozások

Szoftver projekt laboratórium tárgyhonlap Szoftvertechnológia órai jegyzet és előadásdiák Programozás alapjai 3 órai jegyzet és előadásdiák https://[www.iit.bme.hu/targyak/BMEVIIIAB02](http://www.iit.bme.hu/targyak/BMEVIIIAB02)

* + 1. Összefoglalás

Továbbiakban bővebben ismertetésre kerülnek a készítendő szoftver sajátosságai, mind a leendő felhasználók szempontjából, mind a mi, fejlesztők szempontjából fontosnak tartott aspektusok. Továbbá ismertetésre kerülnek a szoftver megvalósításával kapcsolatos tervek.

###### Áttekintés

* + 1. Általános áttekintés

A játék szerkezete, alapvetően két nagy részre bontható: pálya elemei és a játékosok. A pálya elemei lehetnek rugalmas csövek, illetve valamiféle aktív elemek.

* napelemes vízemelő pumpa
* forrás
* ciszterna

Legalább 4 játékos lesz, 2 csapatba osztva. A két csapat a szerelők és a nomádok. Mindkét csapat különböző célokat tűz ki a játék során.

* + 1. Funkciók

A drukmákori sivatagon át bonyolult csőrendszer szállítja a vizet a hegyi forrásokból a sivatagon túl elterülő városok ciszternáiba. A víz a forrásoknál ered, ideális esetben áthalad valamilyen útvonalon a csőrendszeren és a ciszternákba töltődik. A csőrendszer egyszerű, elágazás nélküli csövekből és a csövekhez csatlakozó aktív elemekből. Ezek az elemek a forrás, a ciszterna, a napelemmel működő vízátemelő pumpa. A rendszer változtatható és bővíthető csövek mozgatásával, pumpák állításával, vagy ezen elemek újonnan hozzáadásával. Ez a csöveknél és a pumpánál más-más módon történhet.

Egy pumpa több, de véges számú csövet köt össze. Minden pumpán külön-külön állítható, hogy éppen melyik belekötött csőből melyik másik rákötött csőbe továbbítsa a vizet. Ilyenkor minden más rákötött cső el van zárva, oda nem folyik víz.

A pumpák véletlen időközönként el tudnak romlani, ez azt jelenti, hogy az adott pumpa nem tudja a vizet továbbítani, megszűnik a vízáramlás. Mindegyik pumpa rendelkezik egy víztartállyal, amit a víz átemelése közben használ átmeneti tárolóként. Az elromlott pumpáknál a víz ebben a tartályban marad mindaddig, amíg azt meg nem javítják. A pumpa csak akkor tud vizet pumpálni egy csőbe, ha a cső szabad kapacitása ezt lehetővé teszi. Ellenkező esetben egy másik csövet kell választani a feladatra.

A csőhálózat bővíthető, változtatható. A csövek rugalmasságának köszönhetően azokat szabadon le lehet csatlakoztatni a pumpákról, a hálózat egy másik részére átvinni, és ott rácsatlakoztatni egy másik pumpára. Így átalakítva, optimalizálva a hálózatot, vagyis a víz útját a lehető legrövidebbre szűkíteni. A ciszternáknál folyamatosan készülnek az új csövek, amelyek egyik vége a ciszternához kapcsolódik, a másik azonban szabad. A szabad végű csövekből a csőbe betáplált víz kifolyik a hálózatból, a homokba, így nem jut el a városi ciszternákhoz.

A csőhálózatot a szerelők tartják karban. Ők javítják meg az elromlott pumpákat, valamint ők állítják át a pumpákat, hogy mindig a lehető legtöbb víz tudjon áthaladni a hálózaton és jusson el a városi ciszternákig. Ha egy cső kilyukad, az adott lyukas cső megfoltozása is a szerelők feladata. A kilyukadt csövekből a víz kifolyik a homokba, a csövek másik végén lévő pumpához már nem jut belőle. A ciszternáknál lévő szabad csövekkel lehetséges a hálózat kapacitásának növelése a szerelők számára. A ciszternáknál magukhoz tudnak venni új pumpát is, amit egy cső közepén tudnak elhelyezni. A csövet ehhez ketté kell vágni, és a két újonnan keletkezett végét a pumpához kell csatlakoztatni.

A hálózaton élnek a nomád szabotőrök is, akik feladata a vízátvitel szabotálása, megakadályozása. A pumpákat tudják átállítani, hogy azok ne a legoptimálisabb irányba, ne a megfelelő csőbe továbbítsák a vizet. A nomádok másik tevékenysége a csövek kilyukasztása. Ilyenkor a lyukas csöveken a víz elfolyik a homokba.

Mivel a sivatag veszélyes hely, ezért a szerelők és a szabotőrök csak a csőhálózaton haladhatnak, nem léphetnek le a hálózat elemeiről. Ha egy pumpánál találkoznak, akkor ott kikerülhetik egymást, egy pumpán egyszerre többen is állhatnak. Azonban a csöveken már nem tudnak elmenni egymás mellett, egy csövön egyszerre csak egy ember állhat.

A játékot két csapat játssza: a szerelők csapata és a nomádok csapata. Legalább 2 tagnak kell lennie minden csapatban, vagyis összesen legalább 4 fő kell a program használatához. A nomádok (vagy másnéven szabotőrök) dolga, hogy minél több víz folyjon el a lyukakon a homokba, vagyis, hogy ne jusson el a ciszternákig. A szerelők pedig azon dolgoznak, hogy minél több vizet tudjanak átfolyatni a rendszeren, vagyis, hogy minél több jusson el a ciszternákba. Az a csapat nyer, amelyik a játék végére több vizet szerez.

* + 1. Felhasználók

A játékot egyszerre több felhasználó használja. A felhasználóknak a program használatához nincs szükségük különösebb előismeretre.

* + 1. Korlátozások

A játék fejlesztéséhez a Java könyvtárat kell alkalmazni és a felhasználói felületet a swing csomaggal kell megvalósítani.

* + 1. **Feltételezések, kapcsolatok** Szoftver projekt laboratórium feladatkiírás: https://[www.iit.bme.hu/file/11582/feladat](http://www.iit.bme.hu/file/11582/feladat)

###### Követelmények

* + 1. Funkcionális követelmények

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** | **Use-**  **case** |
| A-01 | A rendszerben vannak csövek, és a hozzájuk csatlakozó aktív elemek. | Megvizsgáljuk a hálózatot. | Alapvető | Feladatleírás | Mind |
| A-02 | A játékosok a  „Szerelő” és  „Nomád” karaktereket irányítják a vízhálózaton. | Másfajta karaktert nem tudunk irányítani. | Alapvető | Feladatleírás | Mind |
| A-03 | Egy pumpa több csövet is összeköthet, de egyszerre csak egy-egy aktív be- és kimenete lehet. | Megvizsgáljuk, hogy tényleg csak egy irányba folyik-e víz a pumpán át. | Alapvető | Feladatleírás | Mind |
| A-04 | Minden karakter a hálózaton halad, de egy csövön egyszerre csak egy karakter lehet. | Megvizsgáljuk, hogy van-e olyan cső ahol több karakter áll. | Alapvető | Feladatleírás | Lép |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A-05 | Minden karakter átállíthatja a pumpákat. | Megpróbálunk pumpát átállítani mindkét féle karakterrel. | Alapvető | Feladatleírás | Pumpa állít |
| A-06 | A „Szerelő” karakterek át tudják állítani a csöveket. | Átállítunk egy csövet, hogy máshova irányítsuk a vizet. | Alapvető | Feladatleírás | Csőállít ás |
| A-07 | A „Szerelő” karakterek meg tudnak foltozni csöveket. | Megpróbálunk megfoltozni egy lyukas csövet. | Alapvető | Feladatleírás | Csövet foltoz |
| A-08 | A „Nomád” karakterek ki tudják lyukasztani a csöveket. | Megpróbálunk kilyukasztani egy működő csövet. | Alapvető | Feladatleírás | Csövet lyukaszt |
| A-09 | A „Nomád” csapat akkor nyer, ha a játék végén a homokba ment több víz. | Ellenőrizzük a vízállást a játék végén. | Alapvető | Feladatleírás | Mind |
| A-10 | A „Szerelő” csapat akkor nyer, ha a játék végén a ciszternákba ment több víz. | Ellenőrizzük a vízállást a játék végén. | Alapvető | Feladatleírás | Mind |
| A-11 | A lyukas csövekből a víz a homokra folyik, nem jut el a cső végi pumpáig. | Ellenőrizzük, hogy egy lyukas cső utáni pumpában van-e víz. | Alapvető | Feladatleírás | Kör |
| F-01 | A pumpák valamilyen időközönként elromlanak. | Adott időközönként megnézzük, van- e rossz pumpa. | Fontos | Feladatleírás | Pumpa elromlik |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F-02 | A pumpák a rajtuk lévő víztartályban tárolják ideiglenesen a vizet, ha adott cső már tele van. | Megpróbálunk egy teli csőbe vizet irányítani. | Fontos | Feladatleírás | Pumpa pumpál |
| F-03 | A ciszternáknál folyamatosan készülnek új csövek, melyek egyik vége a ciszternához csatlakozik, másik  vége szabad. | Vizet küldünk át egy ilyen szabad végű csövön, ellenőrizve, hogy a homokba folyik-e. | Fontos | Feladatleírás | Cső létrehoz |
| F-04 | A „Szerelő” karakterek a ciszternákra lépve vehetnek föl új pumpákat is, melyeket csövek közepére rakhatnak  le. | Fölveszünk egy új pumpát egy ciszternánál. | Fontos | Feladatleírás | Pumpa létrehoz |
| F-05 | Pumpákon kikerülhetik egymást a karakterek, azaz egyszerre több is tartózkodhat ott. | Több karakterrel egy pumpára lépünk. | Fontos | Feladatleírás | Lép |
| F-06 | Legalább 2-2 játékossal kell játszani. | Megpróbálunk 1-  1 játékossal elindítani egy játékot. | Fontos | Feladatleírás | Kör |
| O-1 | A vízhálózat bővíthető. | Bármelyik elemből plusz darabokat rakunk a hálózatba. | Opcionális | Feladatleírás | Mind |

* + 1. **Erőforrásokkal kapcsolatos követelmények**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** |
| R-101 | A program java nyelven készül | bemutatás | Alapvető | Feladatki-írás |
| R-102 | Az elkészült program futásához JDK szükséges | bemutatás | Fontos | Feladatki-írás |
| R-103 | A program futtatásához ajánlott operációs rendszer  Windows 10 | bemutatás | Opcionális |  |
| R-104 | A program fő beviteli eszközei az egér és a billentyűzet. | bemutatás | Fontos | Csapat |

* + 1. Átadással kapcsolatos követelmények

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Azonosító** | **Leírás** | **Ellenőrzés** | **Prioritás** | **Forrás** |
| R-111 | Dokumentáció  átadása | bemutatás | alapvető | Feladatki-írás |
| R-112 | Szkeleton átadás | bemutatás | alapvető | Feladatki-írás |
| R-113 | Prototípus átadás | bemutatás | alapvető | Feladatki-írás |
| R-114 | Teljes program átadása | bemutatás | alapvető | Feladatki-írás |

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

Nincs ilyen fajta követelmény.

###### Lényeges use-case-ek

* + 1. Use-case leírások

|  |  |
| --- | --- |
| **Use-case neve** | Lép |
| **Rövid leírás** | Egy karakter lép egyet a hálózaton. |
| **Aktorok** | Játékos |
| **Forgatókönyv** | Egy karakter lép egy csőről egy másikra, egy csőről egy  pumpára, vagy fordítva. |

|  |  |
| --- | --- |
| **Use-case neve** | Csövet foltoz |
| **Rövid leírás** | Egy karakter megjavítja a csövet. |
| **Aktorok** | Szerelő |
| **Forgatókönyv** | Egy lyukas csőre lépve a szerelő megfoltozza azt. |

|  |  |
| --- | --- |
| **Use-case neve** | Pumpát javít |
| **Rövid leírás** | Egy karakter kijavít egy pumpát. |
| **Aktorok** | Szerelő |
| **Forgatókönyv** | Egy rossz pumpára lépve a szerelő kijavítja azt. |

|  |  |
| --- | --- |
| **Use-case neve** | Csövet lyukaszt |
| **Rövid leírás** | Egy karakter kilyukaszt egy csövet. |
| **Aktorok** | Nomád |
| **Forgatókönyv** | Egy nomád egy jó csőre lépve kilyukasztja azt. |

|  |  |
| --- | --- |
| **Use-case neve** | Pumpa elromlik |
| **Rövid leírás** | Egy pumpa elromlik. |
| **Aktorok** | Rendszer |
| **Forgatókönyv** | Adott időközönként véletlenszerűen elromlik egy pumpa. |

|  |  |
| --- | --- |
| **Use-case neve** | Kör |
| **Rövid leírás** | Lejátszódik egy kör a játékból. |
| **Aktorok** | Rendszer |
| **Forgatókönyv** | Véletlenszerűen elronthat egy pumpát, levon egy kört a játékidőből, engedi lépni a játékosokat, létrehoz a ciszternáknál új csöveket és pumpákat, figyeli, hogy véget  ért-e a játék. |

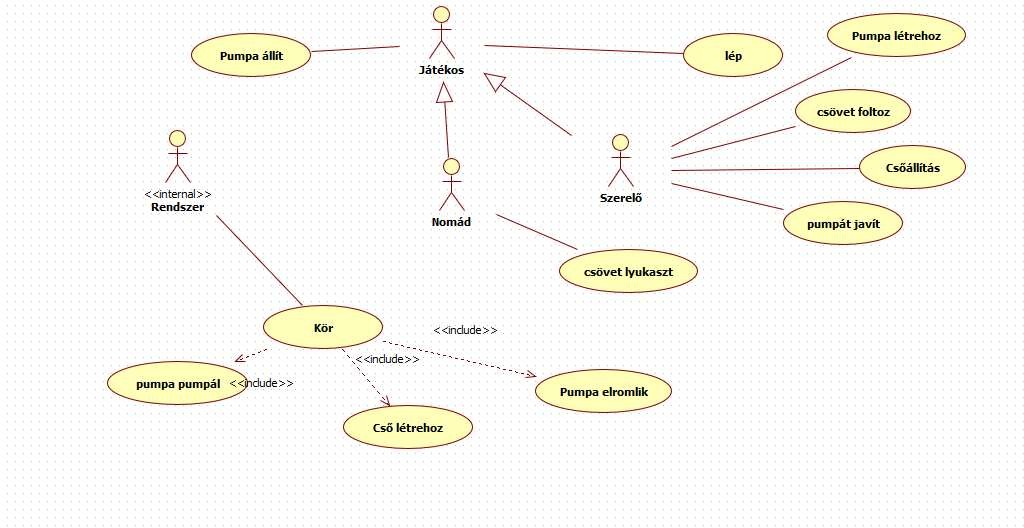
|  |  |
| --- | --- |
| **Use-case neve** | Pumpa pumpál |
| **Rövid leírás** | A csövekbe tesz vizet. |
| **Aktorok** | Rendszer |
| **Forgatókönyv** | A pumpa egyik csőből a másikba, vagy a rajta lévő  víztartályba tesz vizet, avagy fordítva. |

|  |  |
| --- | --- |
| **Use-case neve** | Pumpa állít |
| **Rövid leírás** | Átállít egy pumpát |
| **Aktorok** | Játékos |
| **Forgatókönyv** | Átállítja, hogy a pumpához kötött csövek közül melyiken  folyjon a víz. |

|  |  |
| --- | --- |
| **Use-case neve** | Pumpa létrehoz |
| **Rövid leírás** | Új pumpát hoz létre. |
| **Aktorok** | Szerelő |
| **Forgatókönyv** | A szerelő adott mezőkre lépve kap egy új pumpát. |

|  |  |
| --- | --- |
| **Use-case neve** | Csőállítás |
| **Rövid leírás** | Átállít egy csövet. |
| **Aktorok** | Szerelő |
| **Forgatókönyv** | A szerelő egy csőre lépve átállíthatja, hogy az a cső honnan  hová vezet. |

|  |  |
| --- | --- |
| **Use-case neve** | Cső létrehoz |
| **Rövid leírás** | Új csövet hoz létre. |
| **Aktorok** | Rendszer |
| **Forgatókönyv** | A ciszternáknál a rendszer új csövet hoz létre, amelyeknek  egyik vége szabad. |

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

###### Szótár

**beállítás**: a szerelők egyik tevékenysége, a pumpa beállított állapotba kerül

**beállított pumpa**: a pumpa a megfelelő csőbe továbbítja a vizet

**ciszterna**: az átfolyatott víz végső állomása, a városok vízgyűjtője

**csapat**: a játékot két csapat játszhatja: a szerelők és a nomádok csapata. Mindkét csapat min. 2 főből állhat

**csatlakoztatás**: a cső egyik végének rácsatlakoztatása vagy lecsatlakoztatása egy pumpáról

**cső**: a játékmező egyik típusa, interaktálható elem

**csőrendszer**: másnéven csőhálózat, vagy hálózat. A játékpálya, ezen belül mozognak a játékosok. Két típusú elemből áll: cső és pumpa

**elállítás**: a nomádok egyik tevékenysége, a pumpák elállítódnak

**elállított puma**: a nomádok pumpa elállításának következménye, az adott pumpa nem tudja a vizet továbbítani

**elromlott pumpa**: random időközönként, random pumpánál következik be, ha elromlott, akkor nem tudja a vizet továbbítani

**elzárt cső**: a pumpáknál csak egy csövön érkezhet a víz és egy csövön távozhat, a többi rákötött cső el van zárva

**hegyi forrás**: innen érkezik a víz a csőrendszerbe

**homok**: a nomádok által kifolyatott víz ide kerül

**játékos**: a program felhasználója, valamelyik csapat tagja: vagy szerelő, vagy nomád

**kapacitás növelés**: a szerelők egyik tevékenysége, a szabad csövekkel és pumpákkal növelni tudják a hálózatot és annak a víz átfolyató képességét

**karbantartás**: a szerelők tevékenységeinek összessége

**kikerülés**: a játékosok mozgás közben a pumpákon el tudnak menni egymás mellett

**lyukas cső**: a nomádok lyukasztásának eredménye, a cső nem tudja továbbítani a vizet, kifolyik a rendszerből

**lyukasztás**: a nomádok egyik tevékenysége, a csövek kilyukasztása **megfoltozás**: a szerelők egyik tevékenysége, a csövek megfoltozása **megfoltozott cső**: a csövön lévő lyuk eltűntetése

**megjavítás**: a szerelők egyik tevékenysége, az elromlott pumpák megjavítása

**mozgás**: a játékosok mozgása, ami csak a csőrendszeren belül történhet. A pumpákon kikerülhetik egymást, a csöveken nem

**nomád**: másnéven szabotőr. Az egyik csapat megnevezése, minimum 2 főből áll. A nomádok a csöveket kilyukasztják és a pumpákat elállítják. Céljuk, hogy minél több víz folyjon el a lyukakon.

**nyertes**: az a csapat nyer, amelyik több vizet tudott megszerezni

**puma**: a játékmező egyik típusa, interaktálható elem

**sivatag**: játéktér, ahol a két csapat versenyzik

**szabad cső**: olyan cső, amivel a hálózat szabadon bővíthető, egyik vége a cisztárnához csatlakozni, a másikat a szerelők tudják bekötni

**szabotálás**: a nomádok tevékenységeinek összessége

**szerelő**: az egyik csapat megnevezése, minimum 2 főből áll. A szerelők feladata az elromlott pumpákat megjavítása és beállítása, valamint a kilyukadt csöveket megfoltozása. Céljuk, hogy minél több vizet juttassanak el a városok ciszternáiba a forrásokból.

**új pumpa**: olyan pumpa, amivel a hálózat szabadon bővíthető, a szerelők kettévágnak egy csövet és a vágásnál keletkezett két végét rácsatlakoztatják

**víz**: a forrásoktól, a csőrendszeren át, a ciszternákig folyik a H2O. A játékosok célja minél nagyobb mennyiségű H2O megszerzése

**víztartály**: a pumpák belsejében található, az átfolyatás közben átmeneti tárolóként szolgál

###### Projekt terv

|  |  |  |
| --- | --- | --- |
| **Hét** | **Feladat** | **Határidő** |
| 2 | Követelmények, projekt,  funkcionalistás | márc.13. |
| 3 | Analízis modell (1.változat) | márc.20. |
| 4 | Analízis modell (2.változat) | márc.27. |
| 5 | Szkeleton tervezése | ápr.3. |
| 6 | Szkeleton elkészítése | ápr.17. |
| 7 |
| 8 | Prototípus koncepciója  Szkeleton bemutatása | ápr.24. |
| 9 | Részletes tervek | máj.03. |
| 10 | Prototípus elkészítése | máj.08. |
| 11 | Grafikus változat tervei  Prototípus bemutatása | máj.15. |
| 12 | Grafikus változat elkészítése | máj.31. |
| 13 |
| 14 | Egyesített dokumentáció  Grafikus változat bemutatása | jún.2. |

|  |  |
| --- | --- |
| **Név** | **feladatkör** |
| Kis | software developer, full stack |
| Nyilas | scrum master, software developer, back end |
| Zsigmond | software developer, back end |
| Farkas | software developer, back end |
| Sőregi | software developer, front end |

Felhasznált eszközök:

* GitHub (Git)
* IntelliJ (IDEA)
* Microsoft Word

**Fejlesztési technika:** agilis megközelítés

Issuekat ír ki a scrum master, és ha lehet akkor az egyeztetett feladatkörök alapján kiosztja a feladatokat. Amennyiben ez nem lehetséges munkaterhelés alapján ossza ki a feladatokat, és a scrum elején ezt megbeszéljük és kiosztjuk. A meghatározott sprint időtartamának végén a feladatok bezárásra kerülnek és a main branchre tesszük a változtatásokat.

###### Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2023. 03. 07. 12:30 | 30 perc | Farkas  Kis  Nyilas  Sőregi  Zsigmond | Konzultáció: Kick-off meeting |
| 2023. 03. 07. 13:00 | 5 óra | Kis | Tevékenység:  Kis elkészíti a skeletonokat, a repository-t, és beállítja a verziókövetést. |
| 2023. 03. 08. 12:00 | 1 óra | Nyilas | Tevékenység:  Issue-k összeírása és kiosztása |
| 2023. 03. 08. 13:00 | 4 óra | Zsigmond | Tevékenység:  Zsigmond leírja a funkcionális követelményeket. |
| 2023. 03. 08. 17:00 | 1 óra | Nyilas  Zsigmond | Konzultáció: Use-case diagram tervezés |
| 2023. 03. 08. 17:00 | 15 perc | Kis | Tevékenység: Fedlap elkészítése |
| 2023. 03. 08. 18:00 | 1 óra | Nyilas | Tevékenység:  Use-case diagram elkészítése |
| 2023.03.09. 17:00 | 2 óra | Farkas | Tevékenység:  Farkas leírja az erőforrással és az átadással kapcsolatos követelményeket, ill a hivatkozások, |
| 2023. 03. 10. 17:00 | 1 óra | Farkas Sőregi | Konzultáció:  Megmaradt issue-k kiosztása |
| 2023. 03. 10. 18:00 | 2 óra | Sőregi | Tevékenység:  Szótár elkészítése, Hivatkozások összeírása |
| 2023. 03. 10. 18:30 | 30 perc | Farkas Nyilas | Konzultáció:  Projektterv megbeszélése |
| 2023. 03. 10. 19:00 | 1 óra | Farkas | Tevékenység:  Projektterv dokumentálása |
| 2023. 03. 10. 20:00 | 1 óra | Farkas  Nyilas  Sőregi | Konzultáció: Sprint zárás, naplózás |
| 2023. 03. 11. 07:00 | 2 óra | Sőregi | Tevékenység:  Funkció fejezet megírása |

1. **Analízis modell kidolgozása**
   1. Objektum katalógus
      1. Csőrendszer (Map)

A csőrendszer a sivatagon át szállítja a vizet a hegyi forrástól a városi ciszternákig. Elágazásmentes csövekből és napelemmel működő pumpákból áll össze a hálózat. A rendszer változtatható és bővíthető csövek mozgatásával, pumpák állításával, vagy ezen elemek hozzáadásával.

* + 1. Ciszterna (Cistern)

A ciszternák gyűjtik össze a forrástól a csőhálózaton keresztül szállított vizet. Itt folyamatosan új csövek készülnek, ezek egyik vége egyből a ciszternákhoz csatlakozik, a másik végük szabad. Új pumpák is találhatók itt, ezeket azonban csak a szerelők tudják beépíteni a hálózatba.

* + 1. Cső (Pipe)

A cső a csőrendszer egyik fajta alkotóeleme. Egyszerű, elágazásmentes. A csöveket szabadon le lehet csatlakoztatni a pumpákról, a hálózat egy másik részére átvinni, és ott rácsatlakoztatni egy pumpára. Ha egy cső egyik vége szabad, akkor a rajta átmenő víz a homokba folyik. Ha egy cső kilyukad (vagy kilyukasztják), akkor a víz kifolyik, és a másik végén lévő pumpához már nem tud eljutni.

* + 1. Forrás (WaterSpring)

A víz a forrásból ered. A forráshoz lehet csatlakoztatni a csőrendszer elemeit. A csőrendszeren foglalatoskodó emberek innen indulnak dolgukra.

* + 1. Pumpa (Pump)

A pumpa a csőrendszer másik típusú alkotóeleme. Egy pumpa több csövet is összeköthet, de egyszerre csak egy bemenete és egy kimenete lehet. Egyenként állítható, hogy a rá csatlakoztatott csövek közül melyeken folyjon tovább a víz. A többi bekötött cső ilyenkor elzárt állapotban van. Véletlen időközönként, véletlenszerűen el tudnak romlani a pumpák.

Ha egy pumpa elromlik, akkor megszűnik a rajta áthaladó vízfolyás.

Minden pumpának van egy belső víztartálya, amely az átfolyatott víz átmeneti tárolójaként funkcionál. Csak akkor tudja egy csőbe továbbítani a vizet, ha annak a csőnek a kapacitása ezt engedi.

* + 1. Szerelő (Plumber)

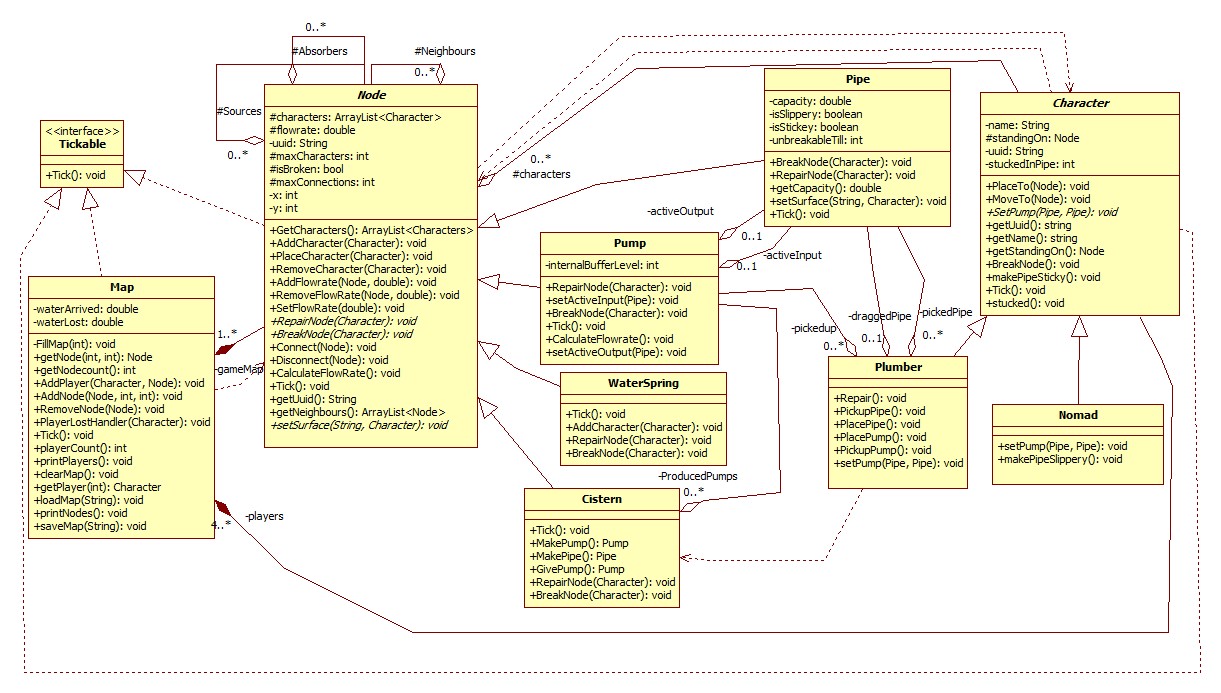
A szerelők feladata a csőhálózat karbantartása. Az elromlott pumpákat megjavítják és a kilyukadt csöveket megfoltozzák. Az ő feladatuk a pumpák beállítása is, amiket úgy állítanak be, hogy mindig a lehető legnagyobb vízmennyiség tudjon átfolyni a rendszeren.

A szabadon lévő csövek segítségével növelni tudják a vízhálózat kapacitását. A ciszternáknál új pumpákhoz juthatnak, amit egy cső közepénél, azt kettévágva tudnak beépíteni a rendszerbe. Csak a csőhálózaton tudnak haladni. A pumpáknál ki tudják kerülni egymást, vagy a nomádokat, de a csöveken már nem. Céljuk, hogy minél több vizet juttassanak el a városok ciszternáiba. Legalább két szerelő tevékenykedik a csőrendszeren.

* + 1. Nomád szabotőr (Nomad)

A nomádok a csöveket kilyukasztják és a pumpákat elállítják. Csak a csőhálózaton tudnak haladni. A pumpáknál ki tudják kerülni egymást, vagy a szerelőket, de a csöveken már nem. Céljuk, hogy minél több víz folyjon el a lyukakon. Legalább két nomád szabotálja a vízhálózatot.

* 1. Statikus struktúra diagramok

******

*Megjegyzés: ez a frissített osztálydiagram már tartalmazza a prototípus koncepcióban megfogalamazott újfajta követelményeket*

* 1. Osztályok leírása
     1. Character
* Felelősség

A játékban szereplő karaktereket megvalósító osztályok ősosztálya. Tárolja, hogy a pálya mely elemén (Node) tartózkodik, és az összes karakter közös viselkedését megvalósítja.

* Ősosztályok

Nincs ősosztálya.

* Interfészek

Nem implementál interfészt.

* Attribútumok
  + **name**: karakter neve
  + **uuid**: karakter egyedi azonosítója
  + **standingOn**: eltárolja, hogy a karakter a Map mely elemén, melyik Node-on áll.
* Metódusok
  + **void placeTo(Node startPoint)**: A karaktert lehelyezi a pálya egy elemére, a játék kezdetekor.
  + **void moveTo(Node moveTo)**: A karakter mozgását valósítja meg. Áthelyezi a paraméterként megadott Nodera, és leveszi az előző Noderól.
  + ***void setPump(Pipe in, Pipe out)****:* A karakter beállítja a Pumpán, amin áll. A Pumpához csatlakoztatott csövek közül melyiket nyitja meg, hogy a pumpába ki és be- folyjon a víz. Absztrakt metódus, leszármazottak implementálják.
  + **string getUuid():** Visszaadja a karakter uuidjét.
  + **string getName():** Visszaadja a karakter nevét.
  + **Node getStaindingOn():** Visszaadja azt a Nodeot, amin a karakter áll.
  + **void breakNode():** Megpróbálja megrongálni azt a Nodeot amin éppen áll. Ha az a Node nem pipe akkor kivételt dob.
    1. Cistern
* Felelősség

A játékban szereplő ciszternákat megvalósító osztály. Ide gyűjtik a vizet a szerelők, és készülnek itt új csövek és pumpák is.

* Ősosztályok

Node →Cistern

* Interfészek

Tickable interfészt implementálja.

* **Attribútumok** *A felsoroltokon kívül rendelkezik még a Node absztrakt ősosztály attribútumaival.*
  + **prodicedPumps:** A ciszterna maga által készített pumpákat tárolja.
* Metódusok
  + **void tick()**: időközönként számolja a ciszternába beérkezett vizet, a hozzá kapcsolt csövekből.
  + **Pipe makePipe()**: Új csöveket készít, amit lehelyez a Map-re és magához csatolja.
  + **Pump makePump()**: Új pumpákat készít, amit elraktároz a producedPump-ba.
  + **Pump givePump():** A rajta álló szerelőnek ad egy pumpát ha az akar felvenni.
  + **void repairNode(Character character):** Próbálja megjavítja az objektumot, de csak kivételt dob.
  + **void breakNode(Character character):** Próbálja elrontja az objektumot, de csak kivételt dob.
    1. Map
* Felelősség

A játékteret megvalósító osztály, pálya. Játéktért elemeit (Node-okat), és a játékos karaktereket csoportosítja.

* Ősosztályok

Nincs ősosztály

* Interfészek

Tickable interfészt implementálja.

* Attribútumok
  + **waterArrived:** Nyilvántartja, hogy mennyi víz folyt be a ciszternába.
  + **waterLost:** Nyilvántartja, hogy mennyi víz folyt el a lyukakon a sivatagi homokra.
  + **players:** A pályán levő játékos Character-ek.
  + **gameMap:** A pálya elemeit tartalmazó lista.
* Metódusok
  + **void FillMap ()**:Map feltöltése elemekkel és játékososkkal.
  + **Node getNode(int x, int y):** Visszaadja a megadott koordinátán lévő Nodeot, ha van ilyen.
  + **int getNodeCount():** Megszámolja és visszaadja a pályán lévő Nodeok számát.
  + **void addNode (Node node, int x, int y)**: Új pályaelem (Node) hozzáadása a pályához a megadott koordinátákra
  + **void removeNode(Node n)**: Kiveszi a megadott Nodeot a pályából.
  + **void addPlayer(Character c, Node n):** A paraméterként megadott játékos hozzáadása a játékhoz, és lehelyezése a megadott Nodera.
  + **void Tick():** Időközönként (tick) számolja a ciszternába folyt, és a homokba elfolyt víz mennyiséségt.
  + **void PlayerLostHandler(Character c):** Ha egy player alatt eltűnik a node akkor ez a metódus fogja visszarakni a megfelelő Nodera.
  + **int playerCount():** Visszaadja a játékosok számát.
  + **void printPlayers():** Kiírja a pályán lévő játékos nevét és hogy melyik Nodeon áll.
  + **void clearMap():** Kiüríti a pályát.
  + **Character getPlayer(int idx):** Visszaadja az adott indexű játékost.
  + **void loadMap(string fileName):** Betölt a megadott nevű fájlból egy pályát.
  + **void saveMap(string fileName):** Kimenti a megadott nevű fájlba a pályát.
  + **void printNodes():** Kiírja a pálya elemeinek, azaz a Nodeokank a koordinátáit és a uuidket.
    1. Node
* Felelősség

A Node osztály egy absztrakt osztály, ezt implementálja minden osztály, amik a elemei lesznek, és játéktérre tehetjük.

* Ősosztályok

Nincs ősosztály.

* Interfészek

Tickable interfészt implementálja.

* Attribútumok
  + **logger:** egy logger ehhez az osztályhoz.
  + **uuid:** a node egyedi azonosítója.
  + **characters:** a jelenleg ezen az nodeon álló karakterek listája.
  + **neighbours:** a jelenleg ezzel a node-dal szomszédos és a node-ba csatlakoztatott más nodeok listája.
  + **sources:** azon nodeok, ahonnan folyik víz ebbe a nodeba.
  + **absorbers:** azon nodeok, ahová folyik víz ebből a nodeba.
  + **maxCharacters:** az ezen a nodeon a maximálisan elfogadott karakterek száma.
  + **maxConnections:** az ehhez a nodehoz maximálisan csatlakoztatható elemek száma.
  + **isBroken:** A node állapota, műkódik e vagy sem.
  + **flowRate:** hogy éppen mennyi víz folyik keresztül a nodeon.
  + **x:** Node x koordináltja a pályán.
  + **y:** Node y koordináltja a pályán.
* Metódusok
  + **String getUuid()**: visszaadja az uuid attribútum értékét.
  + **ArrayList<Character> getCharactersOn():** Visszaadja a nodeon álló karakterke listáját.
  + **ArrayList<Node> getNeighbours():** Visszaadja a node-hoz kötött nodeok listáját.
  + **void placeCharacter(Character character):** elhelyez egy karaktert a nodeon. Játék inicializáláakor használatos, vagy ha a játékos alól eltűnt elem miatt vissza kell helyezni a kezdő nodera.
  + **void addCharacter(Character character):** Hozzáad egy karaktert a characters listához, ha még fér karakter a nodera.
  + **void removeCharacter(Character character):** kivesz egy karaktert a characters listából.
  + ***void repairNode(Character character):*** Megjavítja az objektumot. Absztrakt, leszármazottak implementálják.
  + ***void breakNode(Character character):*** Elrontja az objektumot. Absztrakt, leszármazottak implementálják.
  + **void addFlowRate(Node source, double excededFlow):** Növeli a flowRate-et a megadott értékkel.
  + **void removeFlowRate(Node source, double flowRate):** Csökkenti a flowRate-et a megadott étékkel.
  + **void setFlowRate(double flowRate):** Beállítja a flowRate-et a megadott értékre.
  + **void tick():** Időközönként, azaz körönként számolja a flowRate-et.
  + **void calculateFlowRate():** Számolja a Nodeon átfolyó vít mennyiségét.
  + **void connect(Node node2):** Hozzácsatlakoztatja a megadott Nodeot.
  + **void disconnect(Node node):** Lecsatlakoztatja a megadott Nodeot.
    1. Nomad
* Felelősség

A szabotőr nomád-karaktereket megvalósító osztály. A nomád ki tudja lyukasztani a csöveket (Pipe), el tudja állítani a pumpákat.

* Ősosztályok

Character →Nomad .

* Interfészek

Nem implementál interfészt.

* Attribútumok
  + **name**: karakter neve
  + **uuid**: karakter egyedi azonosítója
  + **standingOn**: eltárolja, hogy a karakter a Map mely elemén, melyik Node-on áll.
* Metódusok
  + **void setPump(Pipe in, Pipe out)**: A nomád beállítja a Pumpán, amin áll. A Pumpához csatlakoztatott csövek közül melyiket nyitja meg, hogy a pumpába ki és be- folyjon a víz.
    1. Pipe
* Felelősség

Ez az osztály implementálja a játékban a csöveket. A csöveket ki lehet lyukasztani és meg lehet javítani, továbbá csatlakoztatni lehet őket más objektumokhoz (összesen maximum kettőhöz), és összesen 1 játékos állhat rajtuk. Ősosztályok

Node →Pipe

* Interfészek

Tickable interfészt implementálja.

* **Attribútumok:** *A felsoroltokon kívül rendelkezik még a Node absztrakt ősosztály attribútumaival.*
  + **capacity**: Azt mutatja mennyi a kapacitása ennek a csőnek (vagyis azt, mennyi víz fér bele).
* Metódusok
  + **double getCapacity()**: Visszaadja a capacity értékét, vagyis azt, mennyi víz fér még bele.
  + **void repairNode(Character character)**: Megjavítja/megfoltozza a csövet.
  + **void breakNode(Character character)**: Kilyukasztja, elrontja a csövet akkor, ha az még nem lyukas
  + **void tick()**: Körönként számolja a flowRatet,
    1. Plumber
* Felelősség

Ez az osztály a játékban lévő vízszerelők osztálya. A vízszerelők meg tudnak javítani csöveket és pumpákat, le tudnak rakni új csöveket és pumpákat, illetve beállíthatják a pumpákat, hogy a ciszternákba tereljék a vizet. A már lerakott csövek átmozgatásával is tudják a vizet a ciszternába terelni.

* Ősosztályok

Character →Plumber

* Interfészek

Nem implementál interfészt

* Attribútumok
  + **name**: karakter neve
  + **uuid**: karakter egyedi azonosítója
  + **standingOn**: eltárolja, hogy a karakter a Map mely elemén, melyik Node-on áll.
  + **pickedUpPump:** A plumber inventoryában a ciszternán felvett pumpát tárolja.
  + **pickedUpPipe:** A plumber inventoryában a mindkét végén lecsatlakoztatott majd felvett csöveket tárolja.
  + **draggedPipe:** Az egyik végén lecsatlakoztatott, és átmozgatott cső.
* Metódusok
  + **void repair()**: megjavítja a Node-ot, amin áll.
  + **void placePump()**: elhelyez egy pumpát oda, ahol áll. Ha csövön áll sikeresen lehelyezi, ha més nodeon akkor kivételt dob.
  + **void placePipe()**: Lehelyezi egy csövet a pályára, és hozzáköti ahhoz a nodehoz amin áll.
  + **void pickupPump():** Megpróbál felvenni egy pumpát. Ha nem ciszternán áll akkor kivételt dob.
  + **void pickupPipe():** Lecsatlakoztatja a cső egyik végét és elhúzza egy másik elemhez. Ha a cső mindkét vége disconnected akkor felveszi az egészet az inventoryába.
  + **void setPump(Pipe in, Pipe out)**: A plumber beállítja a Pumpán, amin áll. A Pumpához csatlakoztatott csövek közül melyiket nyitja meg, hogy a pumpába ki és be- folyjon a víz.
    1. Pump
* Felelősség

Ez a játékban szereplő pumpák osztálya. Egy pumpa állítható, el tud romlani, és meg lehet javítani. Egy pumpán több játékos is állhat egyszerre.

* Ősosztályok

Node →Pump

* Interfészek

Tickable interfészt implementálja.

* **Attribútumok:** *A felsoroltokon kívül rendelkezik még a Node absztrakt ősosztály attribútumaival.*
  + **internalBufferLevel**: ezt arra használjuk, hogy a flowRate-et számoljuk.
  + **activeInput**: ez a pumpa bemenetei közül tárolja azt a csövet, amelyik aktív.
  + **activeOutput**: a pumpa kimenetei közül tárolja azt a csövet, amelyik aktív.
* Metódusok
  + **void repairNode(Character character)**: Megjavítja a pumpát.
  + **void breakNode(Character character)**: Elrontja a pumpát.
  + **void setActiveInput(Pipe p)**: Beállítja az activeInput-ot, azaz hogy, a szomszédjai közül a megadott csőből folyjon be víz a pumpába.
  + **void setActiveOutput(Pipe p)**: Beállítja az activeOutput-ot azaz hogy, a szomszédjai közül a megadott csőbe folyjon ki víz a pumpából.
  + **void calculateFlowRate()**: Számolja a flowRatet
  + **void tick()**: Időközönként (random) elrontja a pupmpát, és számolja a flowRate-et.
    1. Tickable
* **Felelősség**

Ez az interfész minden objektumnak, aminek időközönként kell csinálnia valamit.

* **Ősosztályok**

Nincsenek ősosztályok.

* **Metódusok**
  + ***void tick()****:* az adott időközönként végzett tevékenységre való.
    1. WaterSpring
* **Felelősség**

A WaterSprring egy olyan speciális elem, amivel a ciszternát tudjuk feltölteni. Nem lehet sem elrontani, sem megjavítani, illetve nincs maximum kapacitása.

* **Ősosztályok**

Node →WaterSpring

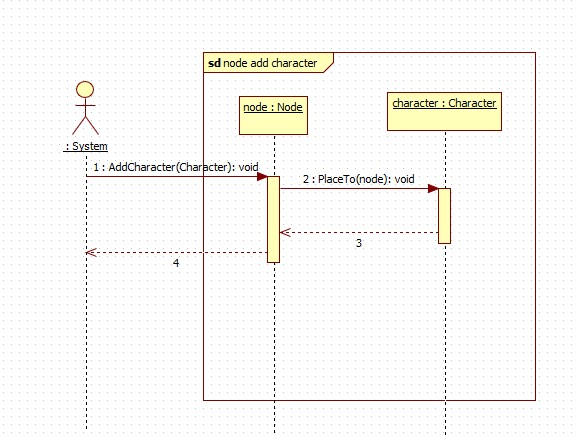
* **Interfészek**

Tickable interfészt implementálja.

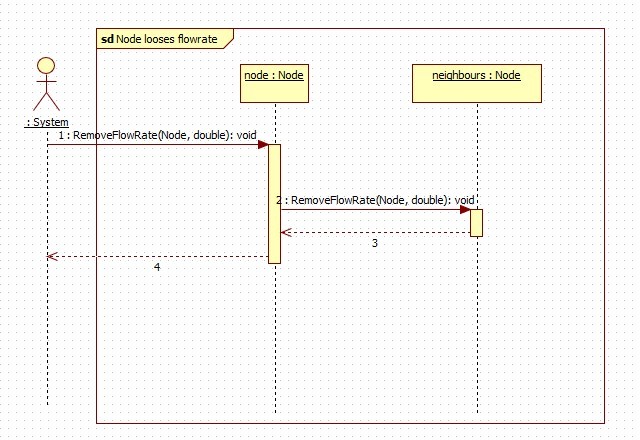
* **Attribútumok**

Nincsnek attribútumai.

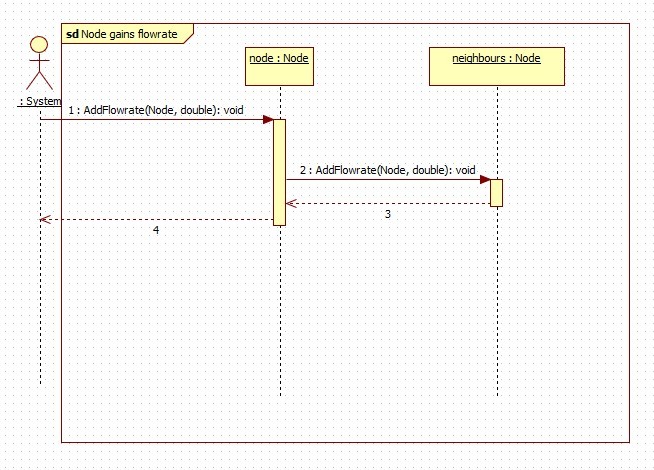
* **Metódusok**
  + **void addCharacter(Character character)**: Hozzáad egy karaktert a charactersOn listához.
  + **void repairNode(Character character)**: Beállítjuk, hogy ez nem lehetséges, kivételt fog dobni.
  + **void breakNode(Character character)**: Beállítjuk, hogy ez nem lehetséges, kivételt fog dobni.
  + **void tick()**: Időközönként számolja a flowRate-et.
  1. Szekvencia diagramok
     1. Add character to Node



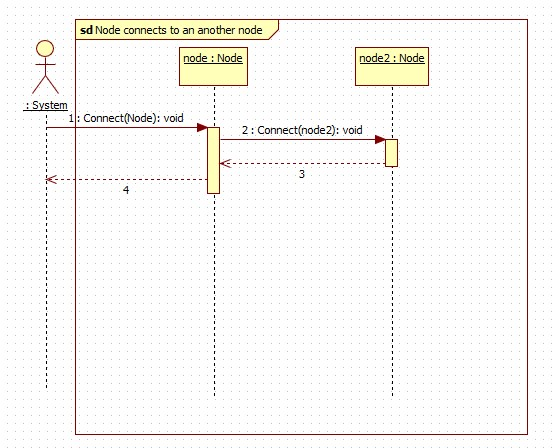
* + 1. **Node loses flowrate:**



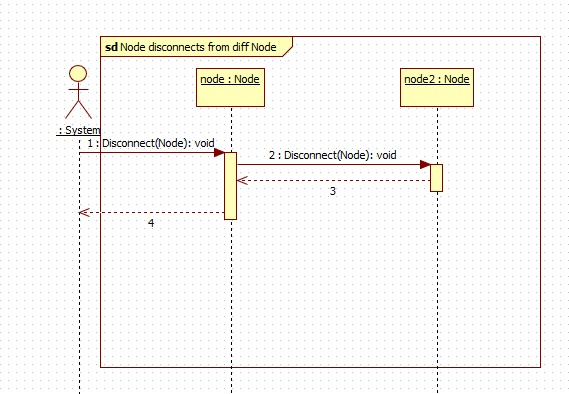
* + 1. Node gains flowrate



* + 1. **Node connects to diff Node**



* + 1. Node disconnects from diff Node

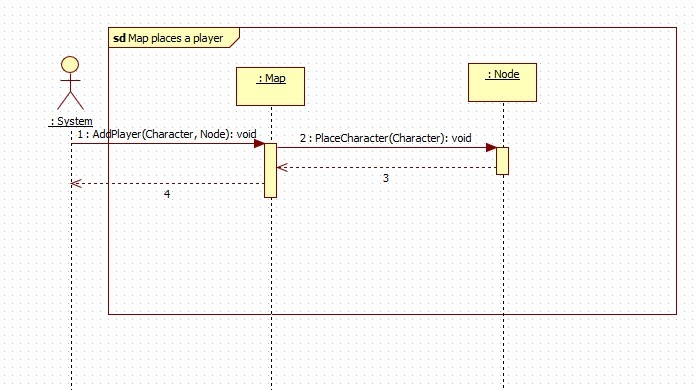


* + 1. **Map initializes**

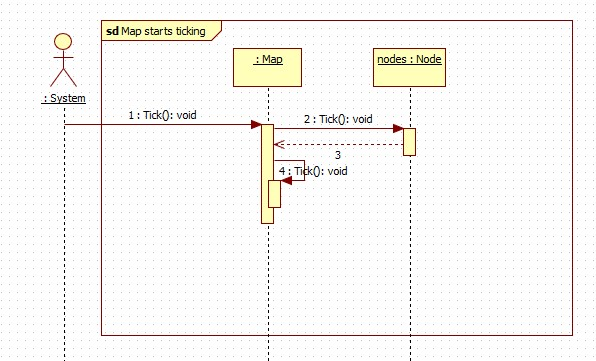
A képen szöveg, diagram, sor, Párhuzamos látható

Automatikusan generált leírás

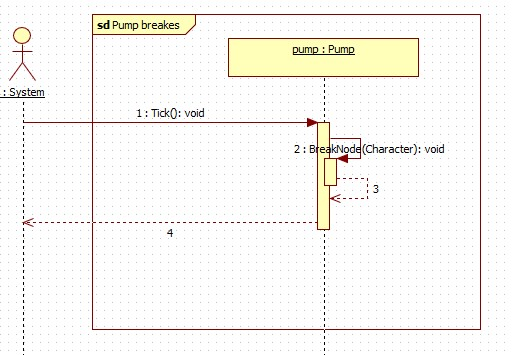
* + 1. Map places a character



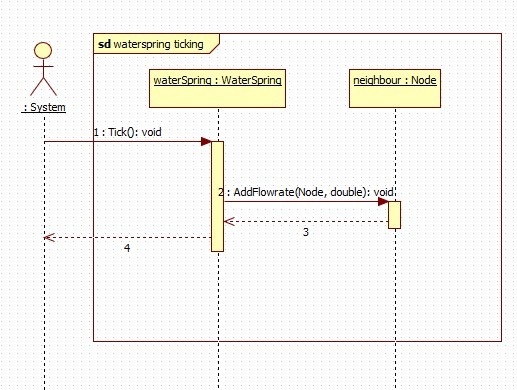
* + 1. **Map starts ticking**



* + 1. Pump breaks



* + 1. **WaterSpring ticking**



* + 1. Cistern makes Pump

A képen szöveg, diagram, sor, képernyőkép látható

Automatikusan generált leírás

* + 1. **Cistern makes Pipe**

A képen szöveg, képernyőkép, diagram, Párhuzamos látható

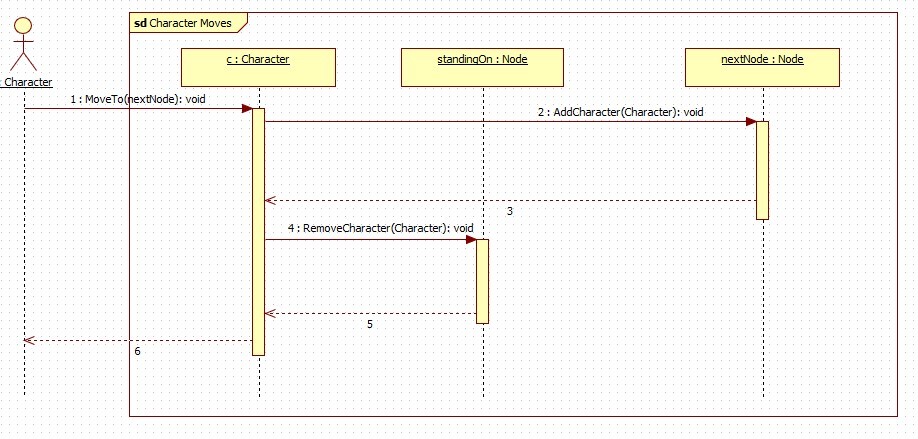
Automatikusan generált leírás

* + 1. Character set Pump

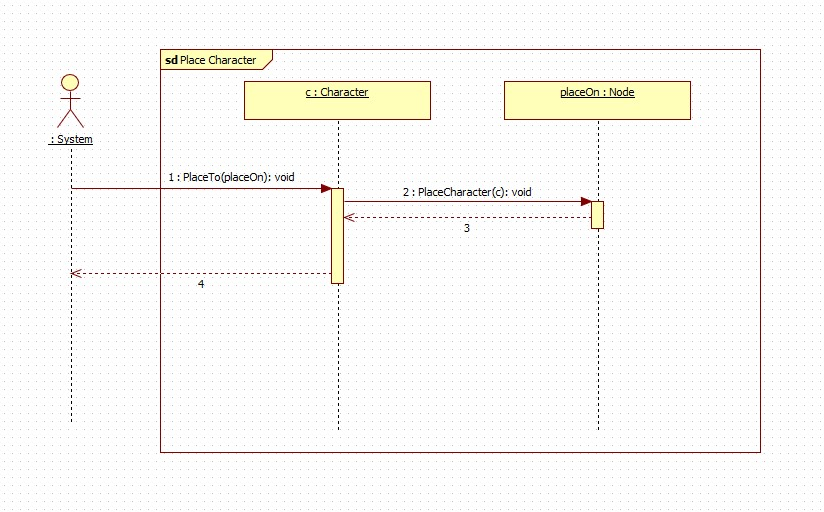
A képen szöveg, diagram, képernyőkép, szám látható

Automatikusan generált leírás

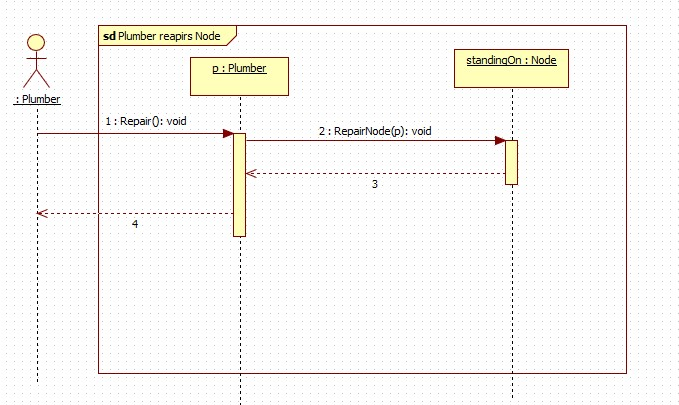
* + 1. **Character Moves**



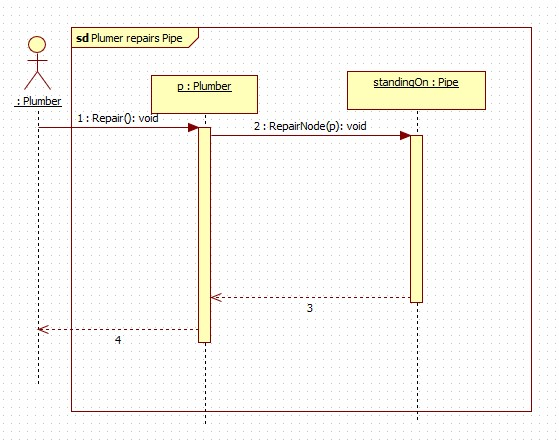
* + 1. Place Character



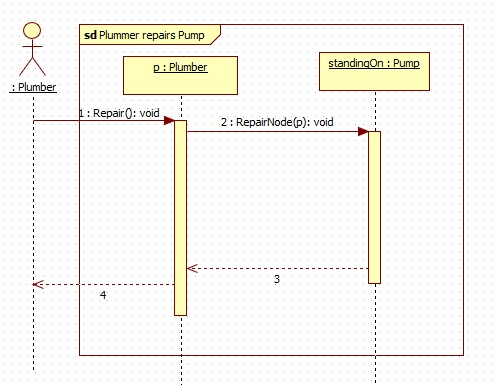
* + 1. **Plumber repairs node**



* + 1. Plumber repairs Pipe



* + 1. **Plumber repairs Pump**



* + 1. Plumber places Pipe

A képen szöveg, képernyőkép, diagram, Párhuzamos látható

Automatikusan generált leírás

* + 1. **Plumber places Pump**

A képen szöveg, diagram, szám, Párhuzamos látható

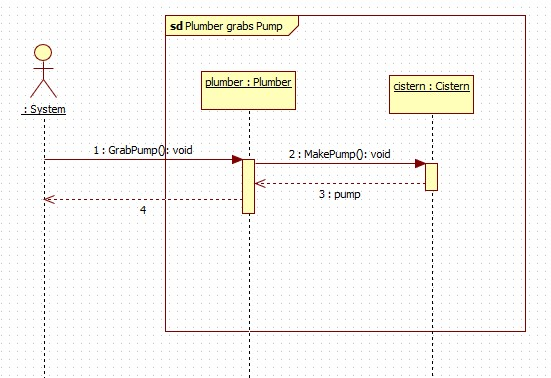
Automatikusan generált leírás

* + 1. Plumber grabs Pipe

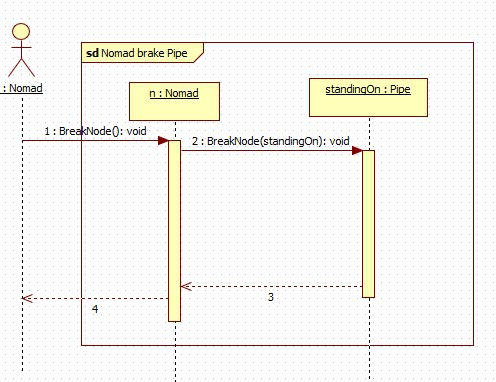
A képen szöveg, diagram, szám, Párhuzamos látható

Automatikusan generált leírás

* + 1. **Plumber grabs Pump**



* + 1. **Nomad brake Pipe**



* 1. Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2023.03.08. 21:00 | 3 óra | Sőregi | Tevékenység: Objektumok listázása és a leírásuk elkészítése |
| 2023.03.17 14:00 | 3 óra | Nyilas | Tevékenység: Osztálydiagram megtervezése és megvalósítás elkezdése |
| 2023.03.17 19:00 | 3 óra | Sőregi | Tevékenység: Osztálydiagram befejezése és végső finomítások |
| 2023. 03. 17. 11:30 | 3 óra | Farkas | Tevékenység: Szekvenicadiagramok átgondolása, Character viselkedést leíró diagramok elkészítése |
| 2023. 03. 18. 10:30 | 1 óra | Farkas | Tevékenység: Nomad, Map és Character osztályok dokumentálása. |
| 2023. 03. 18. 14:15 | 2,5 óra | Zsigmond | Tevékenység: Osztályok leírásának befejezése |
| 2023. 03. 18. 10:0 | 4 óra | Kis | Tevékenység: Szekvencia diagramok |
| 2023. 03. 23 18:00 | 2,5 óra | Kis | Osztály diagram és internal design rework |
| 2023. 03. 25 17:00 | 30 p | Zsigmond, Farkas | Konzultáció, szekvenciák átgondolása |
| 2023. 03. 25 17:30 | 30 p | Farkas | Szekvenciák átszerkesztése |
| 2023. 03. 25 18:00 | 30 p | Zsigmond | Diagramok dokumentálása |
| 2023. 03. 26 19:00 | 1 óra | Mindenki | Konzultáció, dokumentumok átfutás |

1. **Szkeleton tervezése**

###### A szkeleton modell valóságos use-case-ei

* + 1. Use-case diagram (Test-case)

**A képen szöveg, diagram, sor, kör látható

Automatikusan generált leírás**

* + 1. **Use-case leírások**

|  |  |
| --- | --- |
| **Use-case neve** | **Start the game** |
| **Rövid leírás** | A játékos elindítja a játékot |
| **Aktorok** | Player |
| **Forgatókönyv** | A játék betöltődik |

|  |  |
| --- | --- |
| **Use-case neve** | **Move Character** |
| **Rövid leírás** | A játékos mozgatja a karakterét |
| **Aktorok** | Player |
| **Forgatókönyv** | A játékos egy Plumbert (szerelőt) egy csőre helyez |
| **Forgatókönyv** | A játékos egy Plumbert (szerelőt) egy pumpára helyez |
| **Forgatókönyv** | A játékos egy Nomádot (szabotőrt) egy csőre helyez |
| **Forgatókönyv** | A játékos egy Nomádot (szabotőrt) egy pumpára helyez |

|  |  |
| --- | --- |
| **Use-case neve** | **Nomad stands on a pipe** |
| **Rövid leírás** | A játékos által irányított nomád egy cső típusú elemre lépett |
| **Aktorok** | Player |
| **Forgatókönyv** | A nomád kilyukasztja a csövet (break pipe) |

|  |  |
| --- | --- |
| **Use-case neve** | **Character stands on a pump** |
| **Rövid leírás** | A játékos által irányított karakter egy pumpa típusú elemre |

|  |  |
| --- | --- |
|  | lépett |
| **Aktorok** | Player |
| **Forgatókönyv** | A karakter állít a pumpán |

|  |  |
| --- | --- |
| **Use-case neve** | **Plumber stands on a pump** |
| **Rövid leírás** | A játékos által irányított szerelő egy pumpa típusú elemre lépett |
| **Aktorok** | Player |
| **Forgatókönyv** | Ha a pumpa, amire a szerelő lépett rossz, akkor megjavítja |

|  |  |
| --- | --- |
| **Use-case neve** | **Plumber stands on a pipe** |
| **Rövid leírás** | A játékos által irányított szerelő egy cső típusú elemre lépett |
| **Aktorok** | Player |
| **Forgatókönyv** | Ha a cső, amire a szerelő lépett lyukas, akkor megjavítja |
| **Forgatókönyv** | Ha van nála új pumpa, akkor azt telepítheti az adott csőre |

|  |  |
| --- | --- |
| **Use-case neve** | **Plumber stands on cistern** |
| **Rövid leírás** | A játékos által irányított szerelő egy ciszterna elemre lépett |
| **Aktorok** | Player |
| **Forgatókönyv** | A szerelő új pumpát vesz magához |
| **Forgatókönyv** | A szerelő egy új cső végét veszi magához |

|  |  |
| --- | --- |
| **Use-case neve** | **Break pipe** |
| **Rövid leírás** | A cső elrontása, kilyukasztása |
| **Aktorok** | Player |
| **Forgatókönyv** | Egy nomád tönkreteszi a csövet, amin éppen áll |

|  |  |
| --- | --- |
| **Use-case neve** | **Set pump** |
| **Rövid leírás** | A karakter állít a pumpán, amin éppen áll |
| **Aktorok** | Player |
| **Forgatókönyv** | Egy szerelő beállítja az adott pumpát |
| **Forgatókönyv** | Egy nomád elállítja az adott pumpát |

|  |  |
| --- | --- |
| **Use-case neve** | **Repair pump** |
| **Rövid leírás** | A tönkrement pumpa megjavítása |
| **Aktorok** | Player |
| **Forgatókönyv** | Egy szerelő megjavítja az elromlott pumpát, amin éppen áll |

|  |  |
| --- | --- |
| **Use-case neve** | **Picks up pump** |
| **Rövid leírás** | Új pumpa felvétele |
| **Aktorok** | Player |
| **Forgatókönyv** | Egy szerelő új pumpát vesz magához, ha egy ciszternán tartózkodik |

|  |  |
| --- | --- |
| **Use-case neve** | **Picks up the end of pipe** |
| **Rövid leírás** | Egy cső egyik végének felvétele, hogy áthelyezhető legyen |
| **Aktorok** | Player |

|  |  |
| --- | --- |
| **Forgatókönyv** | Egy szerelő felveszi egy cső egyik végét |

|  |  |
| --- | --- |
| **Use-case neve** | **Deploys pump** |
| **Rövid leírás** | Az újonnan felvett pumpa telepítése a rendszerbe |
| **Aktorok** | Player |
| **Forgatókönyv** | Egy szerelő egy cső közepére telepíti az új pumpát |

|  |  |
| --- | --- |
| **Use-case neve** | **Repair pipe** |
| **Rövid leírás** | Egy kilyukasztott cső megfoltozása, megjavítása |
| **Aktorok** | Player |
| **Forgatókönyv** | Egy szerelő megjavítja a lyukas csövet, amin éppen áll |

###### A szkeleton kezelői felületének terve, dialógusok

A szkeleton vezérlésére a konzolon keresztül lesz lehetőség. Minden akció előtt vagy után megjelenik majd egy kérdés, hogy mi legyen a következő interakció. Egyelőre menü alapú az interakció ezért lesz egy segítség menü is ami ezt elmagyarázza.

Folyamat:

A szkeleton elindul és kér egy inputot

* Pálya módosítása
  + Pálya elem lerakása
    - Cső
    - Pumpa
    - Ciszterna
    - Forrás
  + Elemek kitörlése
  + Elemek áthelyezése
* Karakter lehelyezése
  + Nomád
  + Szerelő
* Karakter interakció végrehajtása
  + Mozgás
  + Javítás
  + Elrontás
  + Állítás
* A pálya tickelése (az egész pálya informálása egy kör elmúlásáról)

A válaszok minden esetben egy logger által írt INFO típusú válaszok lesznek melyek a következő formájúak:

HH-pp-mpmp.ms [INFO]: (interakcióba lépő osztály uuid-ja) az akció részletei. Ezután újra felteszi a program a kérdést, hogy mi legyen a következő akció.

###### Szekvencia diagramok a belső működésre

A felhasznált use-case-ek az adott diagram alatt vannak feltüntetve.

A képen szöveg, diagram, Párhuzamos, Tervrajz látható

Automatikusan generált leírás

* Move Character
* Character stands on a pump
* Set pump

A képen szöveg, diagram, képernyőkép, sor látható

Automatikusan generált leírás

* Move Character
* Plumber stands on a pump

A képen szöveg, képernyőkép, diagram, szám látható

Automatikusan generált leírás

* Move Character
* Plumber stands on a pipe

A képen szöveg, képernyőkép, diagram, Párhuzamos látható

Automatikusan generált leírás

* Move Character
* Nomad stands on a pipe
* Break pipe

A képen szöveg, képernyőkép, szám, diagram látható

Automatikusan generált leírás

* Move CharacterA képen szöveg, képernyőkép, diagram, sor látható

  Automatikusan generált leírás
* Move Character
* Nomad stands on a pipe

A képen szöveg, diagram, képernyőkép, sor látható

Automatikusan generált leírás

* Move Character
* Plumber stands on a pipe
* Repair Pipe

A képen szöveg, képernyőkép, diagram, sor látható

Automatikusan generált leírás

* Move Character
* Plumber stands on a cistern
* Picks up a pipe

A képen szöveg, diagram, képernyőkép, sor látható

Automatikusan generált leírás

* Move Character
* Plumber stands on a pipe
* Deploys PumpA képen szöveg, diagram, képernyőkép, szám látható

  Automatikusan generált leírás
* Move Character
* Plumber stands on a pump
* Repair PumpA képen szöveg, képernyőkép, diagram, sor látható

  Automatikusan generált leírás
* Move Character
* Plumber stands on a cisternA képen szöveg, diagram, Párhuzamos, szám látható

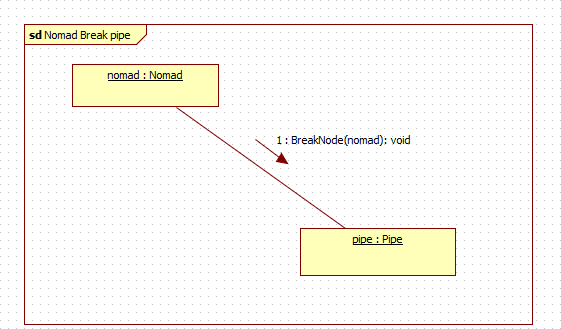
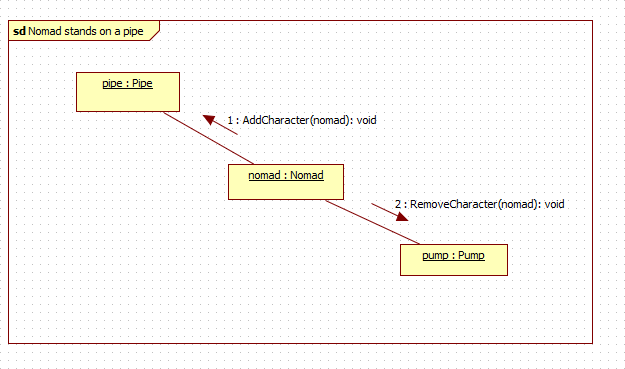
  Automatikusan generált leírás
* Move Character
* Plumber stands on a cistern
* Picks up a pump

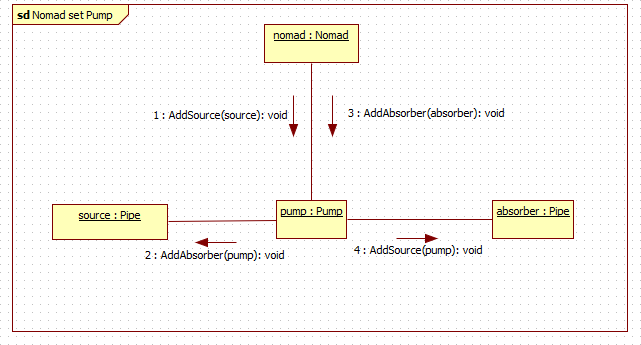
A képen szöveg, képernyőkép, diagram, szám látható

Automatikusan generált leírás

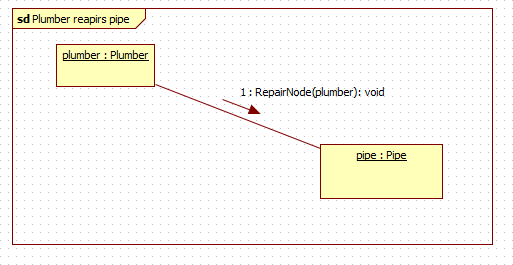
* Move Character
* Nomad stands on a pipe
* Plumber stands on a pipe

###### Kommunikációs diagramok

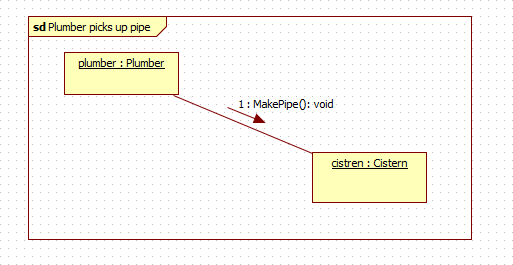
* + 1. Nomad breaks pipe
    2. **Nomad stands on a pipe**
    3. Nomad stands on a pipe



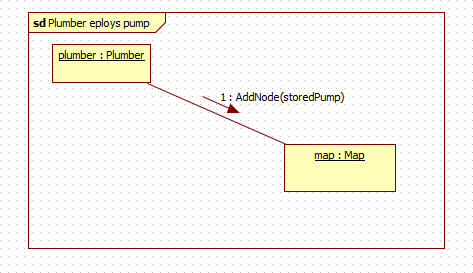
* + 1. **Plumber repairs pipe**



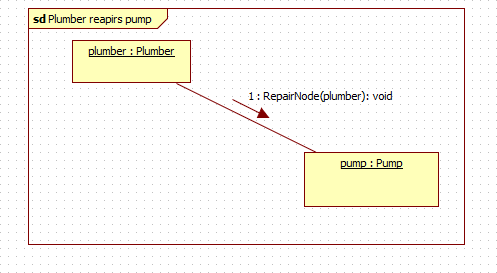
* + 1. Plumber picks up pipe



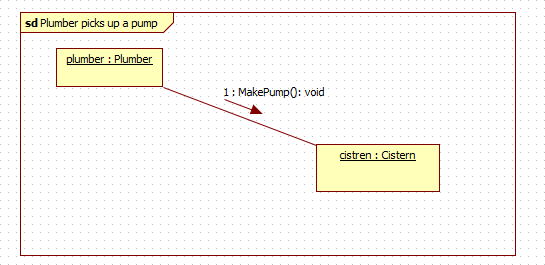
* + 1. **Plumber deploys pump**

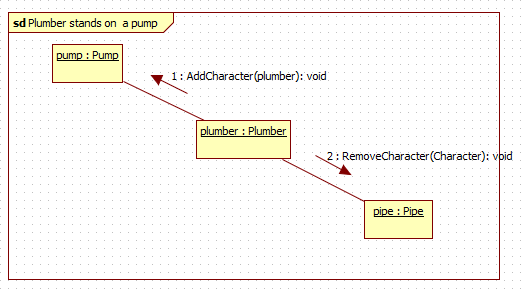
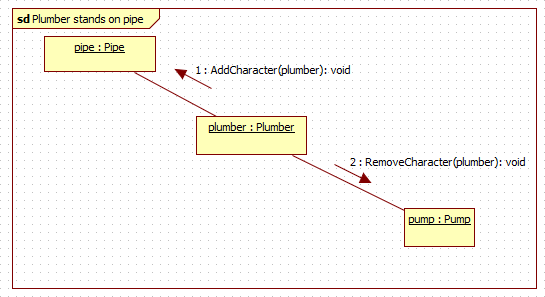
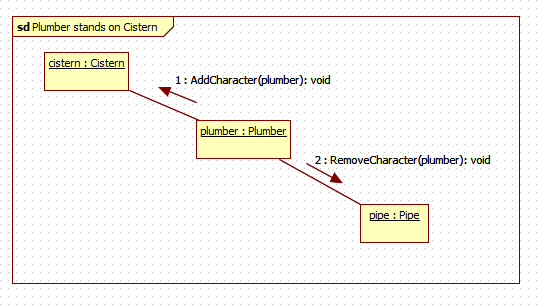
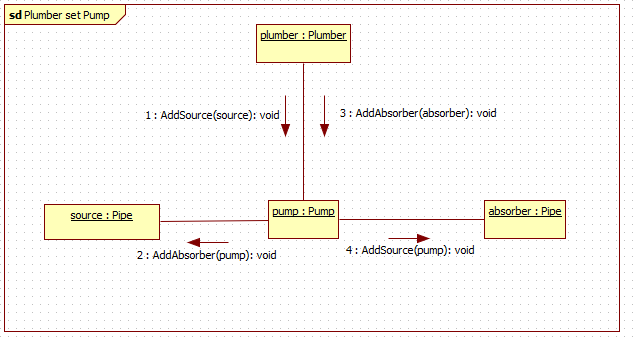


* + 1. Plumber repairs pump



* + 1. **Plumber picks up a pump**



* + 1. Plumber stands on a pump
    2. **Plumber stands on a pipe**
    3. Plumber stands on a cistern
    4. **Plumber sets pump**

###### Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2023.03.30 10:00 | 2 óra | Nyilas | Tevékenység:  Test-case diagram elkészítése |
| 2023.04.01 17:00 | 1,5 óra | Sőregi | Tevékenység:  Test-case diagram dokumentálása |
| 2023.03.31 11:00 | 2,5 óra | Farkas | Tevékenység:  Test-case szekvenciák tervezése, rajzolása |
| 2023.04.02 12:00 | 1 óra | Farkas | Tevékenység:  Kommunikációs diagramok rajzolása |
| 2023.04.02 10:00 | 1 óra | Zsigmond | Tevékenység:  szekvenciák és kommunikációs diagramok dokumentálása |
| 2023.04.02 17:00 | 30 perc | Kis | Tevékenység:  A szkeleton kezelői felületének terve, dialógusok |

**6. Szkeleton beadás**

***6.1 Fordítási és futtatási útmutató***

**6.1.1 Fájllista**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fájl neve** | **Méret** | **Keletkezés ideje** | **Tartalom** |
| Main.java | 5 664 | ‎2023. február 23. | Main fájl |
| Map.java | 3 649 | 2023. március 1. | Ez az osztály felelős a játék pályájáért. |
| SkeletonTest.java | 5 683 | 2023. április 15. | A skeleton tesztelését valósítja meg. |
| Tickable.java | 122 | 2023. március 1. | Interfész a tick-elhető osztályoknak. |
| Character.java | 2 544 | 2023. március 1. | A játékban szereplő karakterekért felelős osztály. Absztrakt osztály az összes karakternek. |
| Nomad.java | 621 | 2023. március 1. | A nomád karaktert reprezentáló osztály. |
| Plumber.java | 818 | 2023. március 1. | A szerelő karaktert reprezentáló osztály. |
| InvalidPlayerActionException.java | 262 | 2023. március 1. | Kivételt dob, ha egy játékos érvénytelen műveletet próbál végrehajtani. |
| InvalidPlayerMovementException.java | 268 | 2023. március 1. | Kivételt dob, ha egy játékos érvénytelen pozícióba próbál mozogni. |
| NotFoundExeption.java | 255 | 2023. március 1. | Kivételt dob, ha egy játékos nem található a szomszédos elemen, ahova mozogni akar. |
| ObjectFullException.java | 227 | 2023. március 1. | Kivételt dob, ha egy objektum elérte maximum kapacitását. |
| Cistern.java | 1 503 | 2023. március 17. | Ciszterna node-ot reprezentáló osztály. |
| Node.java | 5 960 | 2023. március 17. | A Node osztály a pályára helyezhető összes objektum alaposztálya. |
| Pipe.java | 1 207 | 2023. március 17. | Cső node-ot reprezentáló osztály. |
| Pump.java | 2 227 | 2023. március 17. | Pumpa node-ot reprezentáló osztály. |
| WaterSpring.java | 1 323 | 2023. március 17. | Egy speciális node, amit a ciszterna feltöltéséhez használható. |

**6.1.2 Fordítás**

Windows command line-ból lehet fordítani.   
A használandó commandok:   
  
mkdir .\out; cp .\lib\\*.jar .\out; javac -d .\out\ .\src\main\java\com\ez\_mode\Main.java .\src\main\java\com\ez\_mode\\*.java .\src\main\java\com\ez\_mode\characters\\* .\src\main\java\com\ez\_mode\exceptions\\*.\src\main\java\com\ez\_mode\objects\\* -cp .\lib\log4j-api-2.20.0.jar;.\lib\log4j-core-2.20.0.jar

**6.1.3 Futtatás**

A használandó commandok:

cd ./out; java -cp ";%CLASSPATH%;.\log4j-api-2.20.0.jar;.\log4j-core-2.20.0.jar" com.ez\_mode.Main

***6.2 Értékelés***

|  |  |  |
| --- | --- | --- |
| **Tag neve** | **Tag neptun** | **Munka százalékban** |
| Farkas Bence | D6AE9F | 21% |
| Kis Benedek | JOYAXJ | 23% |
| Nyilas Péter | Y8QBCS | 21% |
| Sőregi Larina | M4B9MC | 21% |
| Zsigmond Botond | MV7M6D | 14% |

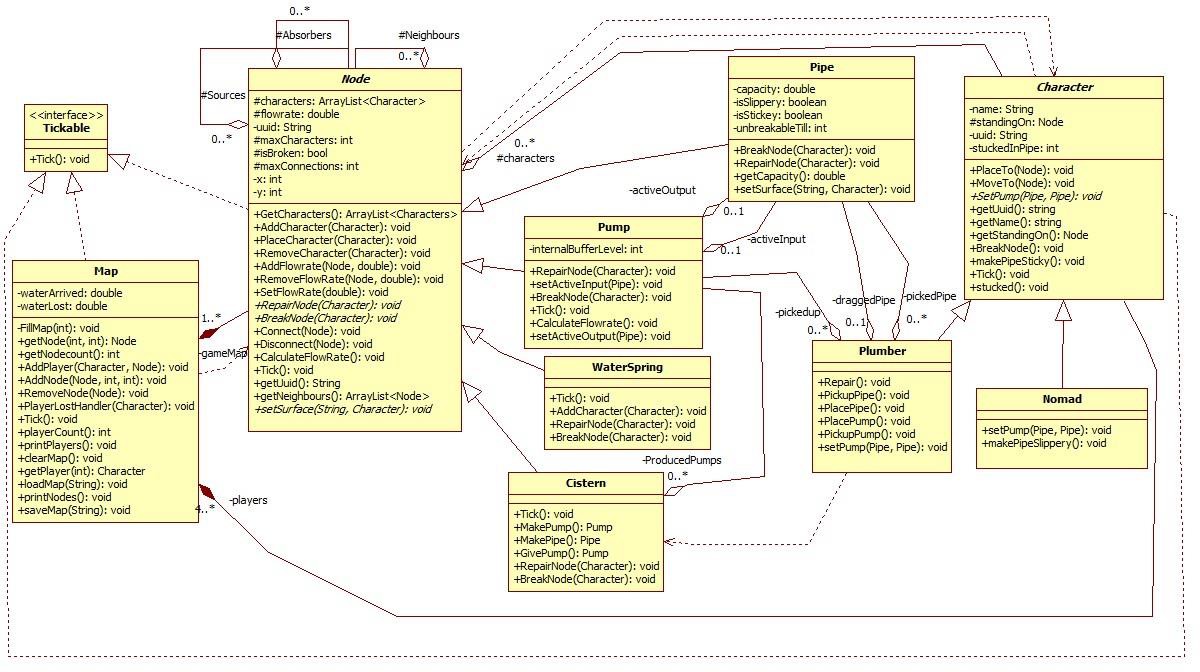
***6.3 Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2023. 03. 01. 12:00 | 10 óra | Kis | Tevékenység:  Skeleton megírása |
| 2023.04.15 11:30 | 2,5 óra | Farkas | Tevékenység:  Skeleton tesztek első felének implementálása |
| 2023.04.15 11:00 | 3 óra | Nyilas | Tevékenység:  Skeleton tesztek második felének implementálása |
| 2023.04.14 13:00 | 2 óra | Farkas | Tevékenység:  Szekvenciák kijavítása |
| 2023. 04. 16. 10:00 | 2 óra | Sőregi | Tevékenység:  Dokumentáció |
| 2023. 04. 16. 10:00 | 2 óra | Kis | Tevékenység:  Main osztály szerkesztése a tesztekhez |
| 2023. 04. 16. 14:00 | 2 óra | Zsigmond | Tevékenység:  Teszt dialógusok |
| 2023. 04. 16. 19:00 | 2 óra | Farkas, Nyilas, Sőregi, Kis | Értekezlet |

1. **Prototípus koncepciója**

# *Változás hatása a modellre*

## Módosult osztálydiagram



*Utolsó oldalon elforgatva nagyobb felbontásban megtalálható.*

## **Új vagy megváltozó metódusok**

* + - 1. Map

+getNode(int, int): Visszaadja az adott 2 dimenziós indexen lévő Node-ot a Map-ból.

+getNodeCount(): Visszaadja a Map-ban lévő Node-ok számát.

+removeNode(Node): Leveszi a Mapről a paraméterként megadott Node-ot.

+playerCount(): Visszaadja a pályán lévő játékosok számát.

+printPlayers(): Kiírja a pályán lévő játékosokat.

+clearMap(): Pálya kiürítése.

+getPlayer(int): Visszaadja az adott indexű játékost.

+loadMap(String): Betölt egy pályát.

+saveMap(String): Elment egy pályát.

+printNodes(): Kiírja a pálya Nodejait.

* + - 1. Node

+getUuid(): Visszaadja az adott Node egyedi azonosítóját.

+getNeighbours(): Visszaad egy az adott Node-hoz tartozó listát annak szomszédairól.

+getCharacters(): Visszaadja az adott Nodeon álló játékosokat.

* + - 1. Cistern

+RepairNode(Character): Nem hagyja, hogy a ciszternát megjavítsák.

+BreakNode(Character): Nem hagyja, hogy a ciszterna elromoljon.

+givePump(): Odaad egy már elkészített pumpát a rajta álló szerelőnek, ha van ilyen.

* + - 1. Pump

+setActiveInput(Pipe): Beállítja, hogy honnan folyik a víz a pumpába.

+setActiveOutput(Pipe): Beállítja, hogy hová folyik a víz a pumpából.

* + - 1. Pipe

+getCapacity(): Visszaadja, hogy mekkora az adott cső kapacitása.

+Tick(): A Node-tól örökölt flowrate számolásán kívül iterációnként csökkenti az utolsó javításnál generált véletlen számot, ameddig nem lehet kilyukasztani.

+RepairNode(): A cső megjavításán kívül generál egy random int-et ameddig nem lehet újra kilyukasztani a csövet.

* + - 1. Plumber

+setPump(Pipe, Pipe): Beállítja a pumpát, amin a szerelő áll.

\*GrabPipe()  PickupPipe(): Fölvesz egy pumpát a ciszternáról.

\*AddPump()  PlacePump(): Leteszi a pumpát egy csőre.

\*GrabPump()  PickupPump(): Fölvesz egy pumpát a ciszternáról.

* + - 1. Character

+getUuid(): Visszaadja az adott karakter egyedi azonosítóját.

+getStandingOn(): Visszaadja hogy hol áll az adott karakter.

+getName(): Visszaadja az adott karakter nevét.

+BreakNode(): Játékos megrongálja azt a Nodeot amin áll.

+Tick(): Amennyiben a karakter ragacsos csőre lép, akkor a véletlen generált időt ameddig nem lépet el, csökkentjük minden meghíváskor.

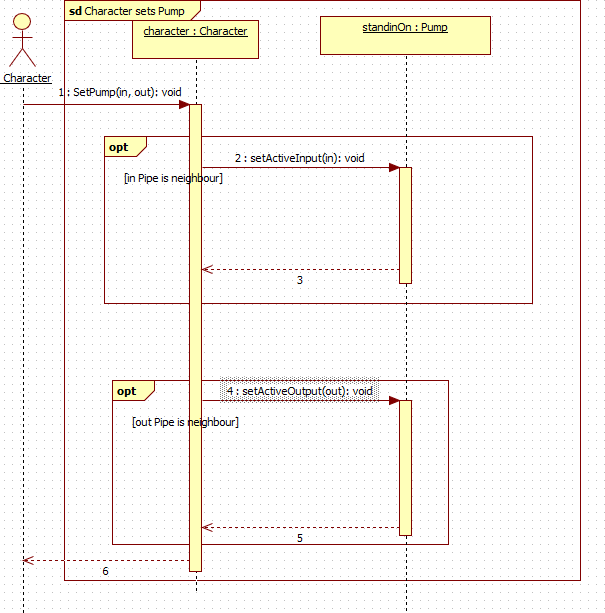
+stucked(): A függvény akkor hívódik ha a karakter egy ragacsos csőre lép, és generál egy random számot ameddig nem léphet tovább.

* + - 1. Nomad

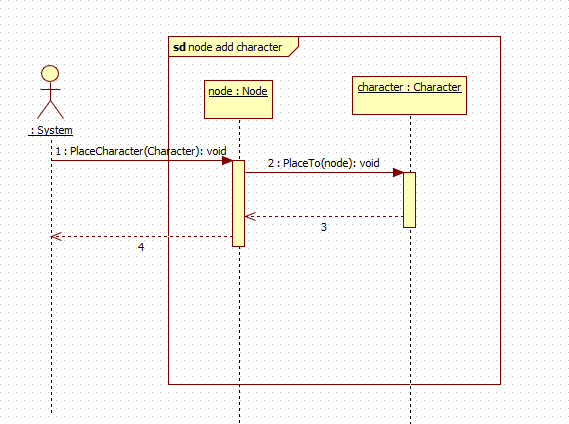
+setPump(Pipe, Pipe): Beállítja a pumpát, amin a nomád áll.

## **Szekvencia-diagramok**

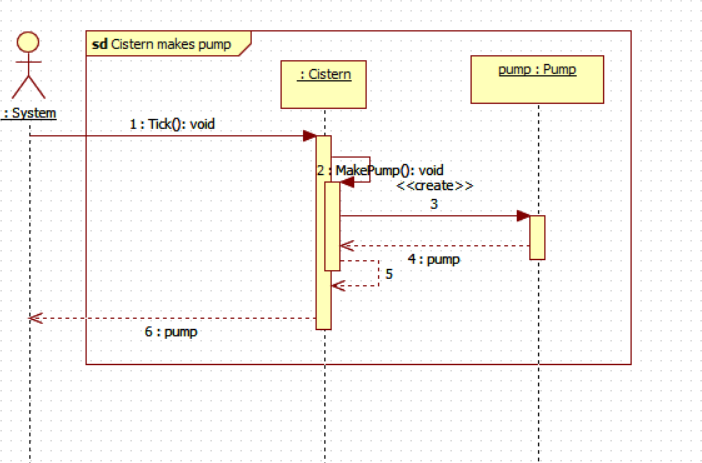
* + - 1. Character sets pump



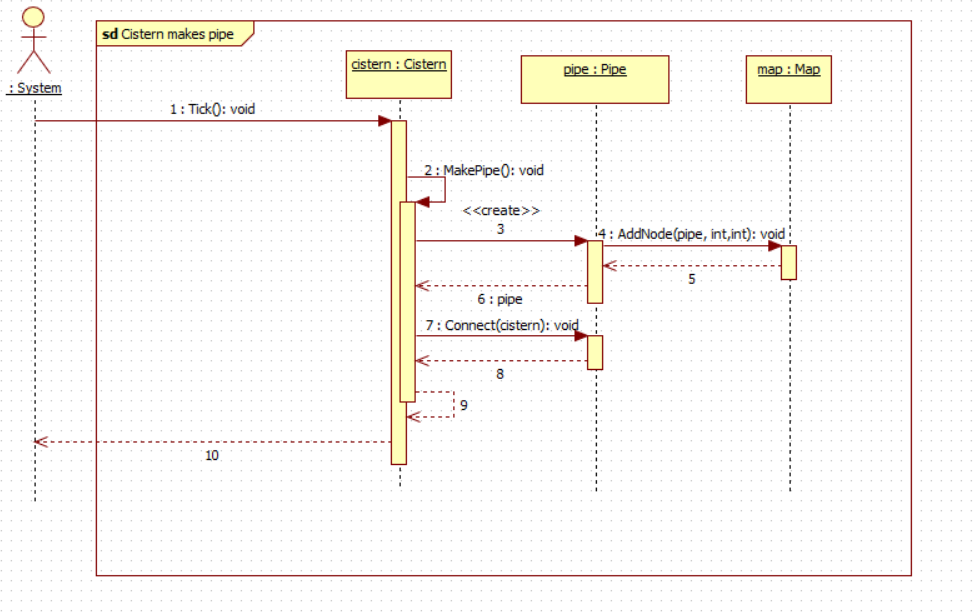
* + - 1. Add character to Node



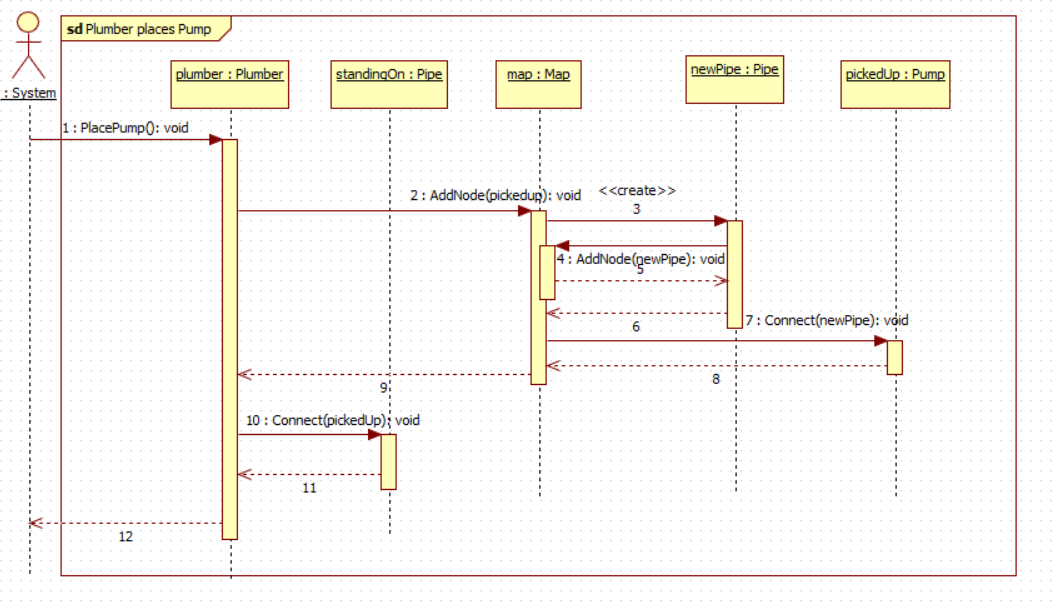
* + - 1. Cistern makes pump



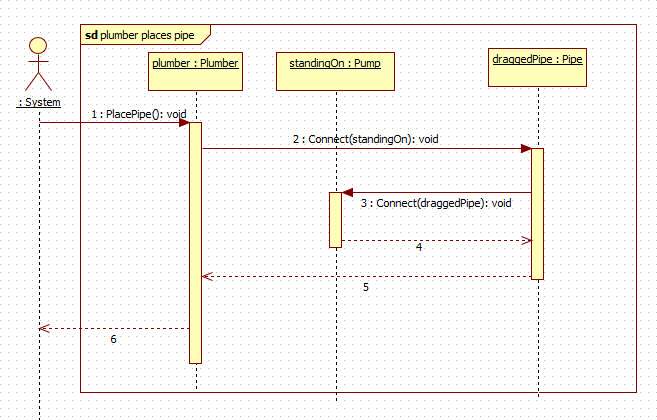
* + - 1. Cistern makes pipe



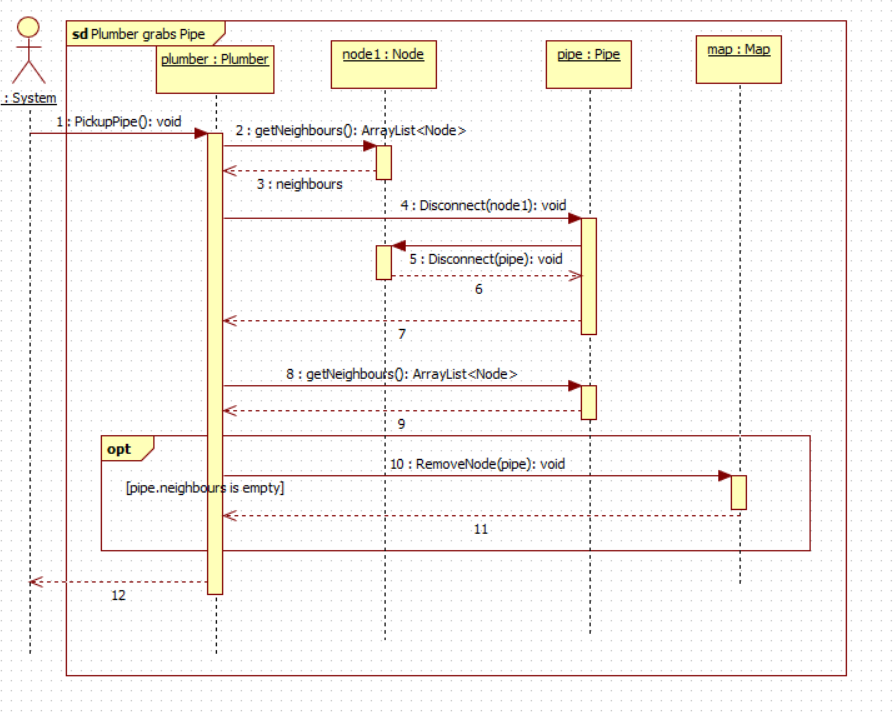
* + - 1. Plumber places pump



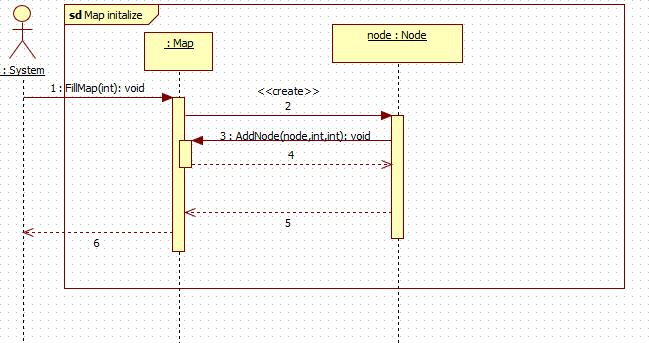
* + - 1. Plumber places pipe

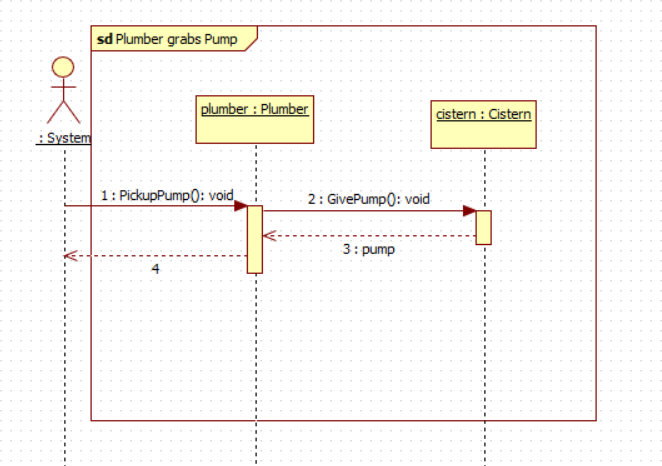


* + - 1. Plumber grabs pipe

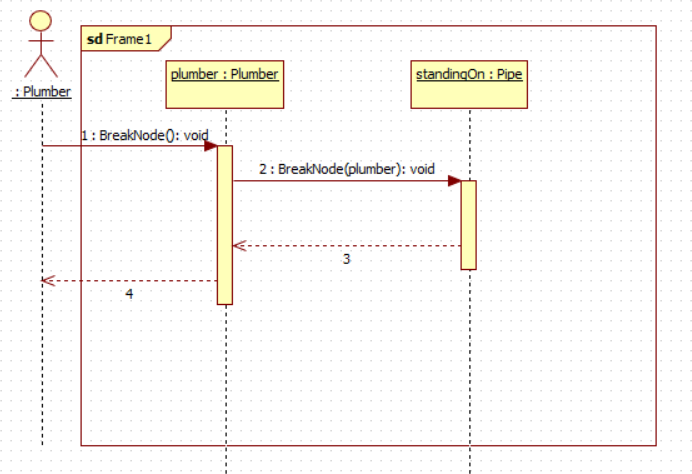


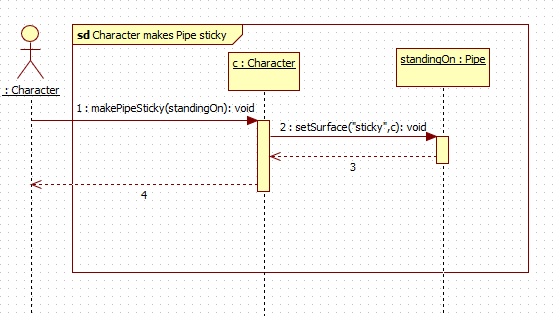
* + - 1. Map initalize

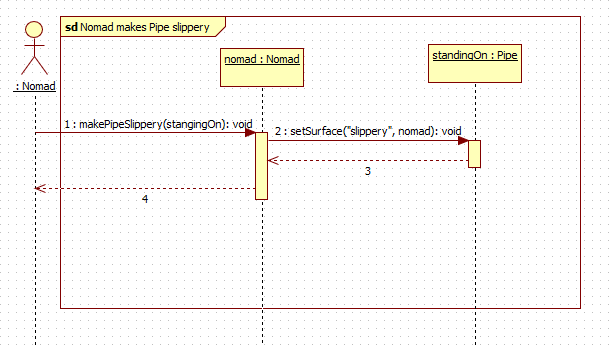


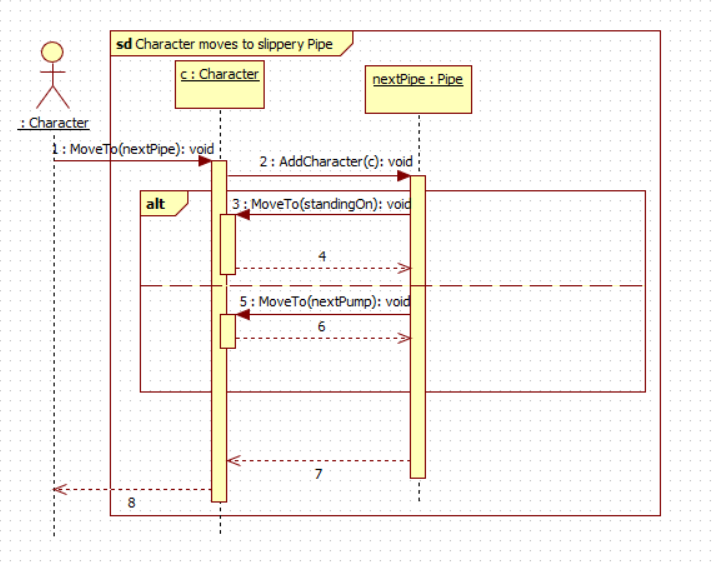
* + - 1. Plumber grabs pump

## **Új funkciók szekvenciái**

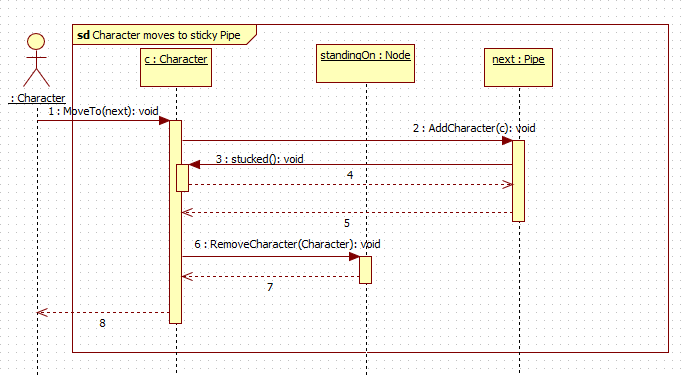
* + - 1. Plumber breaks Pipe
      2. Character makes Pipe sticky



* + - 1. Nomad makes Pipe slippery
      2. Character moves to slippery pipe



* + - 1. Character moves to sticky Pipe



# *Prototípus interface-definíciója*

A különböző teszt esetekhez egy-egy parancsszó tartozik, melyek kiadásával tesztelhető a prototípus, illetve azok láncolásával komplexebb tesztesetek készíthetőek. Egy teszteset sikeres, ha a várt eredmény megegyezik a teszteset által kiírt adatokkal.

## Az interfész általános leírása

A parancsok szabványos bemenetre érkeznek, ami történhet terminálból, illetve előre megírt tesztesetek esetén fájlból, a kimenete a tesztesteknek szintén fájlba vagy terminálba is kiíródhat.

## Bemeneti nyelv

Parancsok:

* *fill*

Leírás: Feltölti a map-ot random elemekkel, ha nem akarjuk manuálisan felvenni az elemeket és játékosokat

Opciók: player count: hány darab játékost akarunk lerakatni

* *load*

Leírás: Beolvas egy megadott konfigurációs file-ból egy játék állapotot Opciók: file path: melyik file-t akarjuk beolvasni

* *save*

Leírás: Kiírja a játék állapotát egy általunk megadott file-ba Opciók: file path: hova mentse el a játékot

* *character*

Leírás: a karakterekhez tartozó akciókat tudjuk innen indítani Opciók:

* + *move: mozgás a karakterrel*
  + *break: a karakter alatti node elrontása*
  + *repair: a karakter alatti node megjavítása*
  + *set: a node átállítása*
  + *pick up: a node felvétele*
  + *place: a karakter hátizsákjából egy elem lehelyezése*
* *node*

Leírás: a node-nak a kezelési opciói Opciók:

* + *break: kilyukad a cső vagy pumpa*
  + *repair: megjavul*
  + *setrandombreak: pumpa magától elromolhat e.*
  + *setslippery: pipe csúszóssá tétele*
  + *setsticky: pipe ragacsossá tétele*
* *add*

*Leírás: új elemet vagy karaktert tudunk felvenni itt Opciók:*

* + *character: karakter veszünk fel Opciók:*
    - *name: a karakter neve*
    - *type: a karakter típusa (nomad vagy plumber)*
    - *node: ahova le szeretnénk helyezni*
  + *node: Opciók:*
    - *type: a node fajtálya (cistern, waterSpring, pipe, pump)*
    - *position: egy x és y koordináta, ahova legyen lehelyezve*
* *map*

Leírás: a térkép akciói Opciók:

* + *tick: az egész map minden elemének 1 körrel való léptetése*
  + *print: a térkép elemeinek kiírtatása átlátható módon*
* *exit*

Leírás: a prototípus elhagyása

Konfigurációs file nyelve:

A konfigurációs file egy json alapú file amiben a következő sorrendben lehet definiálni a játék állapotát:

{

size: a játék térképének mérete,

players: egy lista amiben vannak a játékosok információi

map: egy lista, amiben a node objektumok szerepelnek nem feltétlenül sorrendben

}

A players objektum elemei a következő képpen néznek ki:

{

type: a player típusa name: a player neve x: az x koordinátája y: az y koordinátája

inventory: a hátizsákja ami egy node objektum lista de ez csak Plumber típusú playereknek lehet

}

A map objektum elemei a következő képpen néznek ki:

{

type: a node típusa

x: a node x koordinátája y: a node y koordinátája

connections: ez a tulajdonság nem feltétlenül kell hogy szerepeljen csak ha a típus Pipe vagy Pump, ez egy lista a szomszédok koordinátájával, amikkel a node össze van kötve

}

Pl:

{

"size": 10,

"players": [

{

"type": "plumber",

"name": "Mario", "x": 0,

"y": 0,

"inventory": []

},

{

"type": "nomad",

"name": "Wario", "x": 0,

"y": 0

},

],

"map": [

{

"type": "cistern", "x": 0,

"y": 0

},

{

"type": "pipe", "x": 0,

"y": 1,

"connections": [

{

"x": 0,

"y": 0

}

]

}

]

}

## Kimeneti nyelv

Mikor belépünk az alkalmazásbak, egy menü fogad minked mely kiírja a bemeneti nyelv alap opcióit, mint a *fill* vagy a *load*.

A menük meghívásakor a következő kimeneteket kaphatjuk:

* fill
  + „How many players are there?”
  + “The map is filled up with {x} players and nodes.”
  + “There was an error. “
  + Ilyenkor kiírja a mehívott függvényeket.
* load
  + „What is the path to the configuration file?”
  + „Map loaded succefully”
  + „There was an error loading the map!”
  + Ha sikeresen betöltött akkor látjuk a függvény hívásokat.
* save
  + „Where do you want to save the map?”
  + „Map saved succesfully!”
  + „There was an error saving the map!”
* character
  + „Select a character.” kilistázza az elérhető karaktereket
  + „There are no characters on the map!”
  + „Do you want to add a new character?” átugrik a karakter hozzáadás menübe
  + Amikor meghívunk egy player akciót akkor kiírja a függvény hívásokat, illetve az akció sikerességét.
    - move: “Where do you want to move?” kilistázza a lehetséges helyeket vagy kiírja hogy nem tud mozogni
    - break: függvényhívás és sikeresség
    - repair: függvényhívás és sikeresség
    - set: „Which pipe is the input and the output?”
    - pickup: függvényhívás és sikeresség
    - place: függvényhívás és sikeresség
* node
  + „Select a node” kiírja a nodeokat koordinátákkal
    - break: függvényhívás és sikeresség
    - repair: függvényhívás és sikeresség
* add
* „What do you want to add?”
  + character: „What is the type of the cahracter?”

„What is the name?”

„Where do you want to put this character?” felsorolja a nodeokat

„Character creation aborted”

* + node: „What type of node?”

„What is the position of the node?”

„Node creation aborted”

* map
  + kiírja a térképet olvasható formában négyzetrácsosan
* exit
  + „Exiting game...”

# *Összes részletes use-case*

|  |  |
| --- | --- |
| **Use-case neve** | fill |
| **Rövid leírás** | Feltölti a map-ot random elemekkel, ha nem akarjuk manuálisan felvenni az elemeket és játékosokat |
| **Aktorok** | Map |
| **Forgatókönyv** | Feltöltődik a map elemekkel. |

|  |  |
| --- | --- |
| **Use-case neve** | load |
| **Rövid leírás** | Betölt egy előre megadott játékkonfigurációs fájlt. |
| **Aktorok** | Map |

|  |  |
| --- | --- |
| **Forgatókönyv** | 1. Feltöltődik a map a fájlból a megfelelő elemekkel. 2. Nem létezik a megadott fájl. |

|  |  |
| --- | --- |
| **Use-case neve** | save |
| **Rövid leírás** | Kiírja a játék állapotát a megadott fájlba. |
| **Aktorok** | Map |
| **Forgatókönyv** | 1. Lementjük a játékot. 2. Mentés során security exception keletkezik. |

|  |  |
| --- | --- |
| **Use-case neve** | character |
| **Rövid leírás** | Karakterhez tartozó akciókat innen lehet indítani |
| **Aktorok** | Character |
| **Forgatókönyv** | 1. Kiválasztom valamely karakter almenüt 2. Nincsenek karakterek, és felajánlja a program egy karakter elkészítését. |

|  |  |
| --- | --- |
| **Use-case neve** | move |
| **Rövid leírás** | Mozgás a karakterrel |
| **Aktorok** | Character |
| **Forgatókönyv** | 1. Ha nomad akar ciszternára lépni akkor hiba 2. Ha a character olyan pipera akar lépni amin már állnak akkor hiba. 3. Hibás inputot adtunk meg, ezért nem tudunk mozogni. 4. Odaragadt egy csőhöz így nem tud lépni 5. Character sikeresen lép két elem között. |

|  |  |
| --- | --- |
| **Use-case neve** | break |
| **Rövid leírás** | Megpróbálja megrongálni az alatta lévő Nodeot |
| **Aktorok** | Character |
| **Forgatókönyv** | 1. Sikerül megrongáli 2. Nem sikerül megrongálni, mert nem csövön állt a karakter. 3. Nem sikerült megrongálni, mert már eleve megrongált az állapota. |

|  |  |
| --- | --- |
| **Use-case neve** | repair |
| **Rövid leírás** | Megpróbálja megjavítani az alatta lévő Nodeot |
| **Aktorok** | Plumber |
| **Forgatókönyv** | 1. Sikerült a javítás 2. Nem sikerül megrongálni, mert nem csövön vagy pumpán állt a karakter |

|  |  |
| --- | --- |
| **Use-case neve** | set |
| **Rövid leírás** | Megpróbálja átállítani a pumpát |
| **Aktorok** | Character |
| **Forgatókönyv** | 1. Sikerült az átállítás 2. Nem sikerül az átállítás, mert nem pumpán állt a karakter |

|  |  |
| --- | --- |
| **Use-case neve** | pickup |
| **Rövid leírás** | Cső vagy pumpa felvétele |
| **Aktorok** | Plumber |
| **Forgatókönyv** | 1. Felveszi a cső egyik végét 2. Felveszi az új pumpát a ciszternán. 3. Felveszi a teljes csövet ha mindkét vége szabad. |

|  |  |
| --- | --- |
| **Use-case neve** | place |
| **Rövid leírás** | Plumber lehelyez egy elemet |
| **Aktorok** | Plumber |
| **Forgatókönyv** | 1. Lehelyezi a húzott cső végét 2. Lehelyezi a pumpát egy csőre. 3. Lehelyez egy teljes csövet. |

|  |  |
| --- | --- |
| **Use-case neve** | node |
| **Rövid leírás** | Node menüje/ kezelési opciói |
| **Aktorok** | Node |
| **Forgatókönyv** | 1.Választunk egy opciót. 2.Érvénytelen opciót választunk. |

|  |  |
| --- | --- |
| **Use-case neve** | break |
| **Rövid leírás** | Elrontja a Nodeot |
| **Aktorok** | Node |
| **Forgatókönyv** | 1.Elrontja a pumpát. 2.Kilyukasztja a csövet.  3.Hiba ha nem csövet/pumpát akarunk elrontani |

|  |  |
| --- | --- |
| **Use-case neve** | repair |
| **Rövid leírás** | Megjavítja a nodeot |
| **Aktorok** | Node |
| **Forgatókönyv** | 1. Megjavítja a Nodeot. 2. Hiba, ha nem csövet/pumpát akarunk javítani |

|  |  |
| --- | --- |
| **Use-case neve** | setrandombreak |
| **Rövid leírás** | Beállítja a véletlenszerű meghibásodást |
| **Aktorok** | Pump |
| **Forgatókönyv** | 1. Pumpa véletlenül meghibásodhat. 2. Nem hibásodhat meg véletlenül a pumpa. |

|  |  |
| --- | --- |
| **Use-case neve** | setslippery |
| **Rövid leírás** | Beállítja hogy a cső csúszós e |
| **Aktorok** | Pipe |
| **Forgatókönyv** | 1.Beállítódik a pipeon a csúszós felület. |

|  |  |
| --- | --- |
| **Use-case neve** | setsticky |
| **Rövid leírás** | Beállítja hogy a cső ragadós e |
| **Aktorok** | Pipe |
| **Forgatókönyv** | 1.Beállítódik a pipeon a ragadós felület. |

|  |  |
| --- | --- |
| **Use-case neve** | add |
| **Rövid leírás** | Új elemek / karakterek fölvétle |
| **Aktorok** | internal Controller |
| **Forgatókönyv** | 1.Választunk egy opciót. 2.Érvénytelen opciót választunk. |

|  |  |
| --- | --- |
| **Use-case neve** | add character |
| **Rövid leírás** | Karakter fölvétele |
| **Aktorok** | internal Controller |
| **Forgatókönyv** | 1. Helyesen beállítja a karakter paramétereit 2. Hibásan állítja be a paraméterket, nem sikerül a felvétel |

|  |  |
| --- | --- |
| **Use-case neve** | add node |
| **Rövid leírás** | Pályaelem fölvétele |
| **Aktorok** | internal Controller |
| **Forgatókönyv** | 1. Helyesen beállítja a node paramétereit 2. Hibásan állítja be a paraméterket, nem sikerül a felvétel |

|  |  |
| --- | --- |
| **Use-case neve** | map |
| **Rövid leírás** | map menüje |
| **Aktorok** | map |
| **Forgatókönyv** | 1.Választunk egy opciót. 2.Érvénytelen opciót választunk. |

|  |  |
| --- | --- |
| **Use-case neve** | tick |
| **Rövid leírás** | Map minden elemének egy körrel való léptetése |
| **Aktorok** | map |
| **Forgatókönyv** | 1. Sikeres léptetés 2. Valamely elemnél hibába ütköztünk |

|  |  |
| --- | --- |
| **Use-case neve** | print |
| **Rövid leírás** | Map minden elemének kiírás |
| **Aktorok** | map |
| **Forgatókönyv** | 1.Elemek kiírása |

|  |  |
| --- | --- |
| **Use-case neve** | exit |
| **Rövid leírás** | Programból való kilépés |
| **Aktorok** | internal Controller |
| **Forgatókönyv** | 1.Kilpés a programból |

* 1. ***Tesztelési terv***

|  |  |
| --- | --- |
| **Teszt-eset neve** | Pálya kiírása |
| **Rövid leírás** | A program a betölti és kirajzolja a konzolra a játékpályát |
| **Teszt célja** | A pálya helyes betöltését, az egyes elemek megfelelő elhelyezkedését ellenőrzi. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Játékos hozzáadása |
| **Rövid leírás** | A program felvesz egy új játékost a pályára |
| **Teszt célja** | A játékos helyes felvétele, megfelelő helyre való elhelyezése. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Új pumpa elem felvétele |
| **Rövid leírás** | A szerelő magához vesz egy új pumpát a ciszternáknál |
| **Teszt célja** | Megvizsgálja, hogy a szerelő milyen elemen áll. Új pumpát csak ciszternánál vehet fel. Ha nem ciszternán  áll, akkor ezt üzenet jelzi a játékosnak. Ha ciszternán áll, akkor felveszi az új pumpát. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Cső elem felvétele |
| **Rövid leírás** | A szerelő felveszi a cső végét |
| **Teszt célja** | Megvizsgálja, hogy milyen elemen áll. Lecsatlakoztatja a cső egyik végét. Ha a cső mindkét vége lecsatlakoztatott, akkor felveszi a csövet az inventory- ba. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Pumpa elhelyezése |
| **Rövid leírás** | A szerelő egy új pumpát telepít a hálózatra |
| **Teszt célja** | Megvizsgálja, hogy a szerelő milyen elemen áll. Ha nem csövön áll, akkor ezt üzenet jelzi a játékosnak. Ezen  felül ellenőri a szerelőnél lévő elemeket. Ha csövön áll és az inventory-ja tartalmaz egy pumpát, akkor telepíti  azt az adott csőre. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Cső elhelyezése |
| **Rövid leírás** | A szerelő egy új pumpát telepít a hálózatra |
| **Teszt célja** | Megvizsgálja a szerelőnél található elemeket. Ha nincs nála cső, akkor ezt üzenet jelzi a játékosnak. Ha az inventory-ja tartalmaz csövet, akkor hozzáadja a rendszerhez. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Karakter mozgatása |
| **Rövid leírás** | A karakterek mozgatása egyik elemről a másikra |
| **Teszt célja** | Az adott karakter áthelyezését vizsgálja, ami történhet csőre, pumpára, valamint a szerelők esetében ciszterna elemre. Ehhez lekéri a szomszédos elemeket. Ha csőre szeretne lépni, akkor megvizsgálja azt is, hogy azon  tartózkodik-e másik játékos. Ha igen, ezt üzenet jelzi. Egyéb esetben áthelyezi a játékost. Ha a lépés egy cső elemre történik, megnézi annak csúszósságát. Ha a cső csúszik, akkor a játékos annak egyik végéhez kapcsolódó elemre kerül. Ha a cső ragad, akkor a  továbblépési idő számlálója elindul. |
|  | |
| **Teszt-eset neve** | Cső lyukasztása |
| **Rövid leírás** | Az adott karakter eltöri a csövet, amin éppen áll |
| **Teszt célja** | Ellenőrzi, hogy cső típusú elemen áll-e és hogy ez a cső még nincs eltörve, vagy az eltörés óta eltelt adott idő, majd eltöri azt. Ha már törött a cső, akkor ezt üzenet  jelzi és a művelet nem hajtható végre. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Pumpa elállítása |
| **Rövid leírás** | A nomád karakter elállítja a pumpát, amin éppen áll |
| **Teszt célja** | Ellenőrzi, hogy pumpa típusú elemen áll-e a nomád és hogy a pumpa még nincs elállítva, majd elállítja azt. Ha ez nem teljesül, akkor azt a program jelzi. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Pumpa javítása |
| **Rövid leírás** | A szerelő megjavít egy pumpát |
| **Teszt célja** | Megnézi, hogy a szerelő pumpán áll-e és hogy az a pumpa törött-e. Ha igen, végrehajtja a műveletet. Egyéb esetben jelzi a problémát. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Cső javítása |
| **Rövid leírás** | A szerelő megfoltoz egy csövet |
| **Teszt célja** | Megnézi, hogy a szerelő csövön áll-e és hogy az a cső lyukas-e. Ha igen, végrehajtja a műveletet. Egyéb esetben jelzi a problémát. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Cső csúszóssá tétele |
| **Rövid leírás** | A nomád karakter csúszóssá teszi a csövet, amin éppen áll |
| **Teszt célja** | Ellenőrzi, hogy cső típusú elemen áll-e, majd csúszóssá teszi. |

|  |  |
| --- | --- |
| **Teszt-eset neve** | Cső ragadóssá tétele |
| **Rövid leírás** | A karakter ragadóssá teszi a csövet, amin éppen áll |
| **Teszt célja** | Ellenőrzi, hogy cső típusú elemen áll-e, majd ragadóssá teszi. |

* 1. ***Tesztelést támogató segéd- és fordítóprogramok specifikálása*** A teszteseteket parancssorból lehet indítani és a tesztek eredménye is itt lesz látható. A konzolra kiírt eredmény összevethető az elvárt eredménnyel, így megállapítható a művelet

helyessége.

* 1. ***Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2023. 04. 19. 13:00 | 3,5 óra | Zsigmond | A feladat előkészítése, változások leírása. |
| 2023. 04. 19. 14:30 | 15 perc | Nyilas, Zsigmond | Értekezlet a változásokról. |
| 2023. 04. 19. 14:45 | 2 óra | Nyilas | A változások beírása a diagramokba. |
| 2023. 04. 19 17:00 | 2 óra | Farkas | Változás által érintett szekvenciák módosítása |
| 2023. 04. 20 19:00 | 1,5 óra | Farkas | Csúszós/ragacsos csövek koncepciójának átgondolása, majd a megfelelőnek ítélt változtatások a class diagramban + kezdeti szekvenciák |
| 2023. 04.21 10:50 | 1 óra | Farkas | Csúszós/ragacsos csövek szekvenciáinak kiegészítése, további szükséges szekvenciák elkészítése |
| 2023. 04. 21. 19:00 | 1,5 óra | Sőregi | Tesztek kidolgozása |
| 2023. 04. 23. 12:15 | 1,5 óra | Farkas | Class diagram és szekvenciák apróbb javításai és módosítása, a csapat tagjainak javaslatai szerint. |
| 2023. 04. 23. 17:30 | 0,5 óra | Sőregi | Tesztek változtatása a új módosítások szerint |

1. **Részletes tervek**

###### Osztályok és metódusok tervei.

* + 1. Character
* Felelősség

A játékban szereplő karaktereket megvalósító osztályok ősosztálya. Tárolja, hogy a pálya mely elemén (Node) tartózkodik, és az összes karakter közös viselkedését megvalósítja.

* Ősosztályok

Nincs ősosztálya.

* Interfészek

Tickable interfészt implementálja.

* Attribútumok
  + **name**: karakter neve
  + **uuid**: karakter egyedi azonosítója
  + **standingOn**: eltárolja, hogy a karakter a Map mely elemén, melyik Node-on áll.
  + **stuckedInPipe**: random generált idő, ameddig a karakter nem léphet el a csőről ha az ragacsos.
* Metódusok
  + **void placeTo(Node startPoint)**: A karaktert lehelyezi a pálya egy elemére, a játék kezdetekor.
  + **void moveTo(Node moveTo)**: A karakter mozgását valósítja meg. Áthelyezi a paraméterként megadott Nodera, és leveszi az előző Noderól.
    - if(stuckedInPipe == 0)
      * than moveTo addCharacter(character);
      * standingOn removeCharacter(character);
      * character standingOn = moveTo;
      * endif
  + **void setPump(Pipe in, Pipe out)***:* A karakter beállítja a Pumpán, amin áll. A Pumpához csatlakoztatott csövek közül melyiket nyitja meg, hogy a pumpába ki és be- folyjon a víz.
  + **string getUuid():** Visszaadja a karakter uuidjét.
  + **string getName():** Visszaadja a karakter nevét.
  + **Node getStaindingOn():** Visszaadja azt a Nodeot, amin a karakter áll.
  + **void breakNode():** Megpróbálja megrongálni azt a Nodeot amin éppen áll. Ha az a Node nem pipe akkor kivételt dob.
    - standingOn breakNode;
  + **void makePipeSticky():** Megpróbálja ragacsossá tenni azt a Nodeot amin áll. Ha az a Node nem pipe akkor kivételt dob.
    - stadningON setSrface(„sticky”, character);
  + **void tick():** Tickable interfészt implementált metódusa. Körönként csökkenti a stuckedInPipe időt, miután ragacsos csőre lépett.
    - if (stuckedInPipe > 0) than stuckedInPipe -= 1;
  + **void stuck():** Ha ragacsos csőre lépünk generálunk egy véletlen számot, és beállítjuk ezt a stuckedInPipe időnek.
    - stuckedInPipe = random int;
    1. Cistern
* Felelősség

A játékban szereplő ciszternákat megvalósító osztály. Ide gyűjtik a vizet a szerelők, és készülnek itt új csövek és pumpák is.

* Ősosztályok

Node →Cistern

* Interfészek

Tickable interfészt implementálja.

* **Attribútumok** *A felsoroltokon kívül rendelkezik még a Node absztrakt ősosztály attribútumaival.*
  + **prodicedPumps:** A ciszterna maga által készített pumpákat tárolja.
* Metódusok
  + **void tick()**: időközönként számolja a ciszternába beérkezett vizet, a hozzá kapcsolt csövekből.
    - if (size of sources < maxConnections)
      * than newPipe = Cistern makePipe();
      * add newPipe to sources;
    - endif
    - while(iterate throught sources)
      * Map waterArrives += Node flowRate;
    - end
  + **Pipe makePipe()**: Új csöveket készít, amit lehelyez a Map-re és magához csatolja.
    - create Pipe temp;
    - temp connect(Cistern);
    - Map addNode(temp, x, y);
    - return temp;
  + **Pump makePump()**: Új pumpákat készít, amit elraktároz a producedPump-ba.
    - create Pump temp;
    - add temp to producedPumps;
  + **Pump givePump():** A rajta álló szerelőnek ad egy pumpát ha az akar felvenni.
    - if (producedPump is not empty)
      * than remove and return Pump from producedPump;
    - else return null;
  + **void repairNode(Character character):** Próbálja megjavítja az objektumot, de csak kivételt dob.
  + **void breakNode(Character character):** Próbálja elrontja az objektumot, de csak kivételt dob.
  + **void setSurface(string surface, Character c):** Próbálja csúsuóssá/ragacsossá az objektumot, de csak kivételt dob.
    1. Map
* Felelősség

A játékteret megvalósító osztály, pálya. Játéktért elemeit (Node-okat), és a játékos karaktereket csoportosítja.

* Ősosztályok

Nincs ősosztály

* Interfészek

Tickable interfészt implementálja.

* Attribútumok
  + **waterArrived:** Nyilvántartja, hogy mennyi víz folyt be a ciszternába.
  + **waterLost:** Nyilvántartja, hogy mennyi víz folyt el a lyukakon a sivatagi homokra.
  + **players:** A pályán levő játékos Character-ek.
  + **gameMap:** A pálya elemeit tartalmazó lista.
* Metódusok
  + **void FillMap ()**:Map feltöltése elemekkel és játékososkkal.
  + **Node getNode(int x, int y):** Visszaadja a megadott koordinátán lévő Nodeot, ha van ilyen.
  + **int getNodeCount():** Megszámolja és visszaadja a pályán lévő Nodeok számát.
    - count = 0;
    - for each column in gameMap
      * for each node in column
        + if node is not null then

count = count+1;

* + - * + end if
      * end for
    - end for
    - return count;
  + **void addNode (Node node, int x, int y)**: Új pályaelem (Node) hozzáadása a pályához a megadott koordinátákra
  + **void removeNode(Node n)**: Kiveszi a megadott Nodeot a pályából.
  + **void addPlayer(Character c, Node n):** A paraméterként megadott játékos hozzáadása a játékhoz, és lehelyezése a megadott Nodera.
  + **void Tick():** Időközönként (tick) számolja a ciszternába folyt, és a homokba elfolyt víz mennyiséségt.
    - for each nodes array in gameMap
      * for each node in nodes
        + node.tick();
      * end for
    - end for
  + **int playerCount():** Visszaadja a játékosok számát.
  + **void printPlayers():** Kiírja a pályán lévő játékos nevét és hogy melyik Nodeon áll.
    - for i from 0 to players size()-1
      * player = players.[i];
      * print player’s name and the node’s uuid;
    - end for
  + **void clearMap():** Kiüríti a pályát.
  + **Character getPlayer(int idx):** Visszaadja az adott indexű játékost.
  + **void loadMap(string fileName):** Betölt a megadott nevű fájlból egy pályát.
    - if file does not end with ”.json” then
      * print ”the file must be a .json configuration file!”
    - else
      * load the map
    - end
  + **void saveMap(string fileName):** Kimenti a megadott nevű fájlba a pályát.
    - if file does not end with ”.json” then
      * print ”the file must be a .json configuration file!”
    - else
      * print the map
    - end
  + **void printNodes():** Kiírja a pálya elemeinek, azaz a Nodeokank a koordinátáit és a uuidket.
    - for each nodes array in gameMap
      * for each node in nodes
        + if node is not null then

print x and y coordinates and uuid of the node;

* + - * + end if
      * end for
    - end for
    1. Node
* Felelősség

A Node osztály egy absztrakt osztály, ezt implementálja minden osztály, amik a elemei lesznek, és játéktérre tehetjük.

* Ősosztályok

Nincs ősosztály.

* Interfészek

Tickable interfészt implementálja.

* Attribútumok
  + **logger:** egy logger ehhez az osztályhoz.
  + **uuid:** a node egyedi azonosítója.
  + **characters:** a jelenleg ezen az nodeon álló karakterek listája.
  + **neighbours:** a jelenleg ezzel a node-dal szomszédos és a node-ba csatlakoztatott más nodeok listája.
  + **sources:** azon nodeok, ahonnan folyik víz ebbe a nodeba.
  + **absorbers:** azon nodeok, ahová folyik víz ebből a nodeba.
  + **maxCharacters:** az ezen a nodeon a maximálisan elfogadott karakterek száma.
  + **maxConnections:** az ehhez a nodehoz maximálisan csatlakoztatható elemek száma.
  + **isBroken:** A node állapota, műkódik e vagy sem.
  + **flowRate:** hogy éppen mennyi víz folyik keresztül a nodeon.
  + **x:** Node x koordináltja a pályán.
  + **y:** Node y koordináltja a pályán.
* Metódusok
  + **String getUuid()**: visszaadja az uuid attribútum értékét.
  + **ArrayList<Character> getCharactersOn():** Visszaadja a nodeon álló karakterke listáját.
  + **ArrayList<Node> getNeighbours():** Visszaadja a node-hoz kötött nodeok listáját.
  + **void placeCharacter(Character character):** elhelyez egy karaktert a nodeon. Játék inicializáláakor használatos, vagy ha a játékos alól eltűnt elem miatt vissza kell helyezni a kezdő nodera.
  + **void addCharacter(Character character):** Hozzáad egy karaktert a characters listához, ha még fér karakter a nodera.
    - if (size of characters >= macCharacters)
      * than throw exception;
    - else while(iterate throught neighbours)
      * if (a neighbour contains character) than add character to characters
    - end
  + **void removeCharacter(Character character):** kivesz egy karaktert a characters listából.
    - if(characters contains character)
      * than remove character from characters
  + ***void repairNode(Character character):*** Megjavítja az objektumot. Absztrakt, leszármazottak implementálják.
  + ***void breakNode(Character character):*** Elrontja az objektumot. Absztrakt, leszármazottak implementálják.
  + **void addFlowRate(Node source, double excededFlow):** Növeli a flowRate-et a megadott értékkel.
    - if(sources contains source)
      * than flowRate += excededFlow;
      * while(iterate throught absorbers)
        + addFlowRate();
        + end
    - endif
  + **void removeFlowRate(Node source, double flowRate):** Csökkenti a flowRate-et a megadott étékkel.
    - if(sources contains source)
      * than flowRate -= flowRate;
      * while(iterate throught absorbers)
        + removeFlowRate();
        + end
    - endif
  + **void setFlowRate(double flowRate):** Beállítja a flowRate-et a megadott értékre.
  + **void tick():** Időközönként, azaz körönként számolja a flowRate-et.
    - calculateFlowRate();
  + **void calculateFlowRate():** Számolja a Nodeon átfolyó vít mennyiségét.
    - if(Node is broken)
      * than Map waterLost += Node flowRate;
      * while(iterate througth absorber nodes)
        + absotber node removeFlowRate(Node, Node flowRate);
        + end
    - else
      * while(iterate througth absorber nodes)
        + absotber node addFlowRate(Node, Node flowRate);
        + end
  + **void connect(Node node2):** Hozzácsatlakoztatja a megadott Nodeot.
    - node1 neighbours add node2;
    - if(node2 neigbours not contains node1)
      * than node2 connect(node1);
  + **void disconnect(Node node):** Lecsatlakoztatja a megadott Nodeot.
    - node1 neighbours remove node2;
    - if(node2 neigbours contains node1)
      * than node2 disconnect(node1);
  + ***void setSurface(string surface, Character c):*** Absztrakt metódus amit a leszármazottak implementálnak. Célja, hogy a karakterek tudják ragacsossá/csúszóssá teni a pályaelemek egy részét (pipe).
* **boolean fullOfConn():** Visszaadja lehet e még a nodre elemeket csatlakoztatni.
  + 1. Nomad
* Felelősség

A szabotőr nomád-karaktereket megvalósító osztály. A nomád ki tudja lyukasztani a csöveket (Pipe), el tudja állítani a pumpákat és ragacsossá illetve csúszóssá tudja tenni a csövetket.

* Ősosztályok

Character →Nomad .

* Interfészek

Tickable interfészt implementálja.

* Attribútumok
  + **name**: karakter neve
  + **uuid**: karakter egyedi azonosítója
  + **standingOn**: eltárolja, hogy a karakter a Map mely elemén, melyik Node-on áll.
  + **stuckedInPipe**: random generált idő, ameddig a karakter nem léphet el a csőről ha az ragacsos.
* Metódusok
  + **void setPump(Pipe in, Pipe out)**: A nomád beállítja a Pumpán, amin áll. A Pumpához csatlakoztatott csövek közül melyiket nyitja meg, hogy a pumpába ki és be- folyjon a víz.
    - if(in is neighbour of standingOn)
      * than setActiveInput(in);
    - if(out is neighbour of standingOn)
      * setActiveOutput(out);
  + **void makePipeSlippery()**:A nomád csúszóssá teszi a cső felszínét.
    - standingOn setSurface(„slippery”, Nomad);
    1. Plumber
* Felelősség

Ez az osztály a játékban lévő vízszerelők osztálya. A vízszerelők meg tudnak javítani csöveket és pumpákat, le tudnak rakni új csöveket és pumpákat, illetve beállíthatják a pumpákat, hogy a ciszternákba tereljék a vizet. A már lerakott csövek átmozgatásával is tudják a vizet a ciszternába terelni, továbbá ők is tudnak csövet lyukasztani

* Ősosztályok

Character →Plumber

* Interfészek

Tickable interfészt implementálja.

* Attribútumok
  + **name**: karakter neve
  + **uuid**: karakter egyedi azonosítója
  + **standingOn**: eltárolja, hogy a karakter a Map mely elemén, melyik Node-on áll.
  + **stuckedInPipe**: random generált idő, ameddig a karakter nem léphet el a csőről ha az ragacsos.
  + **pickedUpPump:** A plumber inventoryában a ciszternán felvett pumpát tárolja.
  + **pickedUpPipe:** A plumber inventoryában a mindkét végén lecsatlakoztatott majd felvett csöveket tárolja.
  + **draggedPipe:** Az egyik végén lecsatlakoztatott, és átmozgatott cső.
* Metódusok
  + **void repair()**: megjavítja a Node-ot, amin áll.
    - standingOn repairNode(Plumber);
  + **void placePump()**: elhelyez egy pumpát oda, ahol áll. Ha csövön áll sikeresen lehelyezi, ha més nodeon akkor kivételt dob.
    - if(has a pickedPump)
      * than Map addNode(pickedPump);
      * create Pipe newPipe;
      * Map addNode(newPipe);
      * newPipe connect(pickedUp);
      * standingOn connect(pickedUp); endif
  + **void placePipe()**: Lehelyezi egy csövet a pályára, és hozzáköti ahhoz a nodehoz amin áll.
    - if(has draggedPipe)
      * than standingOn connect(draggedPipe);
      * draggedPipe = null;
      * return;
    - else if(has pickedUpPipe)
      * than standingOn connect(pickedUpPipe);
      * Map addNode(pickedUpPipe)
      * pickedUpPipe = null;
      * return;
  + **void pickupPump():** Megpróbál felvenni egy pumpát. Ha nem ciszternán áll akkor kivételt dob.
    - Pump P = standingOn givePump();
    - if(P is not null)
      * than pickedUpPump = P;
    - endif
  + **void pickupPipe():** Lecsatlakoztatja a cső egyik végét és elhúzza egy másik elemhez. Ha a cső mindkét vége disconnected akkor felveszi az egészet az inventoryába.
    - while(iterate throught neighbours of standingOn)
      * if(standingOn has at least2 neighbours)
        + than draggedPipe = iterated Node;
        + break;
        + endif
    - end
    - if(draggedPipe neighbours is empty)
      * than pickedUpPipe = draggedPipe;
      * draggedPipe = null;
    - endif
  + **void setPump(Pipe in, Pipe out)**: A plumber beállítja a Pumpán, amin áll. A Pumpához csatlakoztatott csövek közül melyiket nyitja meg, hogy a pumpába ki és be- folyjon a víz.
    - if(in is neighbour of standingOn)
      * than setActiveInput(in);
    - if(out is neighbour of standingOn)
      * setActiveOutput(out);
* **void getEmptyPlace(Node node1, Node node2)**: Keres egy üres helyet ahova tud elemeket lehelyezni.
  + 1. Pipe
* Felelősség

Ez az osztály implementálja a játékban a csöveket. A csöveket ki lehet lyukasztani és meg lehet javítani, továbbá csatlakoztatni lehet őket más objektumokhoz (összesen maximum kettőhöz), és összesen 1 játékos állhat rajtuk. A csöveket a játékosok ragacsossá tudják tenni, továbbá a nomad típusú játékosok csúszóssá is.

* Ősosztályok

Node →Pipe

* Interfészek

Tickable interfészt implementálja.

* **Attribútumok:** *A felsoroltokon kívül rendelkezik még a Node absztrakt ősosztály attribútumaival.*
  + **capacity**: Azt mutatja mennyi a kapacitása ennek a csőnek (vagyis azt, mennyi víz fér bele).
  + **isSlippery:** Csúszós-e a cső felszíne.
  + **isStickey:** Ragacsos-e a cső felszíne.
  + **unbreakableTill:** Cső megfoltozásakor generált véletlen szám, ameddig nem lehet újra kilyukasztani.
* Metódusok
  + **double getCapacity()**: Visszaadja a capacity értékét, vagyis azt, mennyi víz fér még bele.
  + **void repairNode(Character character)**: Megjavítja/megfoltozza a csövet és generál egy véletlen számot, ameddig nem lehet újra kilyukasztani, ezt a számot a unbreakableTill-be menti.
    - if(Pipe is broken)
      * than isBroken = false;
      * unbreakableTill = random int;
  + **void breakNode(Character character)**: Kilyukasztja, elrontja a csövet akkor, ha az unbreakableTill 0, egyéb különben nem lehet lyukasztani
    - if(Pipe is not Broken and unbreakableTill == 0)
      * than isBroken = true;
  + **void tick()**: Körönként számolja a flowRatet, és csökkenti a unbreakableTill-t ha az nem érte el a 0-t.
    - calculateFlowRate();
    - if(unbreakableTill > 0) than unbreakableTill -= 1;
  + **void setSurface(string type, Character c):** A cső felszínét állítja be a paraméterként beadott típusúra. Ha a type paraméter sticky akkor az isStickey lesz igaz, ha slippery akkor az isSlippery, ha pedig none akkor mind a kettő false lesz.
    - if(type == sticky) than isSticky=true;
    - else if(type == slippery) than isSlippery = true;
    1. Pump
* Felelősség

Ez a játékban szereplő pumpák osztálya. Egy pumpa állítható, el tud romlani, és meg lehet javítani. Egy pumpán több játékos is állhat egyszerre.

* Ősosztályok

Node →Pump

* Interfészek

Tickable interfészt implementálja.

* **Attribútumok:** *A felsoroltokon kívül rendelkezik még a Node absztrakt ősosztály attribútumaival.*
  + **internalBufferLevel**: ezt arra használjuk, hogy a flowRate-et számoljuk.
  + **activeInput**: ez a pumpa bemenetei közül tárolja azt a csövet, amelyik aktív.
  + **activeOutput**: a pumpa kimenetei közül tárolja azt a csövet, amelyik aktív.
* Metódusok
  + **void repairNode(Character character)**: Megjavítja a pumpát.
    - if(Pump is broken)
      * than isBroken = false;
  + **void breakNode(Character character)**: Elrontja a pumpát.
    - if(Pump is not Broken)
      * than isBroken = true;
  + **void setActiveInput(Pipe p)**: Beállítja az activeInput-ot, azaz hogy, a szomszédjai közül a megadott csőből folyjon be víz a pumpába.
    - activeInput = p;
  + **void setActiveOutput(Pipe p)**: Beállítja az activeOutput-ot azaz hogy, a szomszédjai közül a megadott csőbe folyjon ki víz a pumpából.
    - activeOutput = p;
  + **void calculateFlowRate()**: Számolja a flowRatet
    - if(Pump is not broken)
      * than if(absorbers conatins activeInput)
        + than if(absorbers conatins activeOutput)

setFlowRate(activeInput flowRate);

activeOutput += flowRate;

* + - * + else internalBufferLevel += activeInput flowRate;
      * else if(internalBufferLevel > 0)
        + than setFlowRate(min{internalBufferLevel, activeInput capacity});
        + activeOutput flowRate += flowRate;
    - else
      * setFlowRate(0);
      * internalBufferLevel += activeInput flowRate;
  + **void tick()**: Időközönként (random) elrontja a pupmpát, és számolja a flowRate-et.
    - calculateFlowRate();
    - if(random int 1) breakNode();
    1. Tickable
* Felelősség

Ez az interfész minden objektumnak, aminek időközönként kell csinálnia valamit.

* Ősosztályok

Nincsenek ősosztályok.

* Metódusok
  + ***void tick()***: Az adott időközönként végzett tevékenységre való megvalósítandó metódus.
    1. WaterSpring
* Felelősség

A WaterSprring egy olyan speciális elem, amivel a ciszternát tudjuk feltölteni. Nem lehet sem elrontani, sem megjavítani, illetve nincs maximum kapacitása.

* Ősosztályok

Node →WaterSpring

* Interfészek

Tickable interfészt implementálja.

* **Attribútumok:** *A Node absztrakt ősosztály attribútumaival rendelkezik.*
* Metódusok
  + **void addCharacter(Character character)**: Hozzáad egy karaktert a charactersOn listához.
  + **void repairNode(Character character)**: Beállítjuk, hogy ez nem lehetséges, kivételt fog dobni.
  + **void breakNode(Character character)**: Beállítjuk, hogy ez nem lehetséges, kivételt fog dobni.
  + **void tick()**: Időközönként számolja a flowRate-et.
    - while(iterate throught neighbours)
      * if(neighbour is not borken)
        + than neighbour addFlowRate(waterSpring, 1);
        + endif

###### A tesztek részletes tervei, leírásuk a teszt nyelvén

* + 1. Pálya kiírása
       - Leírás

A program betölti és kirajzolja a konzolra a játékpályát.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek

Ellenőrizzük, hogy a program helyesen tölti-e be a pályát. Hiba lehet, ha nem úgy néz ki a pálya, mint ahogy elvárnánk; vagy ha nem található a fájl.

* + - * Bemenet

*print map*

* + - * Elvárt kimenet

*Successfully loaded map.*

(megjegyzés: ezután minden tesztesetnél feltételezzük, hogy a pálya létezik a teszt lefutásakor)

* + 1. Játékos hozzáadása
       - Leírás

A program fölvesz egy új játékost a pályára.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek

Ellenőrizzük, hogy a játékost jó helyre vettük-e föl, és jó játékost. Hiba lehet, ha nem vesz fel játékost, vagy rossz helyre, rossz fajtát.

* + - * Bemenet
      * *add character <kataker neve> <x koordináta > <y koordináta >*
      * Elvárt kimenet

*Successfully added character.*

* + 1. Új pumpaelem felvétele
       - Leírás

A szerelő magához vesz egy új pumpát a ciszternáktól.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek

Ellenőrizzük, hogy a szerelő jó helyen áll-e, és hogy felvett-e egy pumpát (ezt üzeneten jelzi a játékosnak, ha tényleg ciszternán áll). Hiba lehet, ha nem értesíti a játékost hogy nem ciszternán áll, vagy nem vette föl a pumpát.

* + - * Bemenet

*add* *node* *pump*  *<x koordináta>*  *<y koordináta>*

* + - * Elvárt kimenet

*Successfully added pump.*

* + 1. Új csőelem felvétele
       - Leírás

A szerelő magához vesz egy új csövet a ciszternáktól.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek

Ellenőrizzük, hogy a szerelő jó helyen áll-e, és hogy felvett-e egy csövet (ezt üzeneten jelzi a játékosnak, ha tényleg ciszternán áll). Hiba lehet, ha nem értesíti a játékost hogy nem ciszternán áll, vagy nem vette föl a csövet.

* + - * Bemenet

*add* *node* *pipe*  *<x koordináta>*  *<y koordináta>*

* + - * Elvárt kimenet

*Successfully added pipe.*

* + 1. Pumpa elhelyezése
       - Leírás

A szerelő új pumpát telepít a hálózatra.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek

Ellenőrizzük, hogy a szerelő jó helyen áll-e, (üzeneten jelzi a játékosnak, ha nem jó helyen áll), és hogy van-e pumpa nála (ha nincs, akkor is szól a játékosnak). Hiba lehet, ha nem telepíti, pedig van nála és jó helyen áll; ha telepít úgy is, ha nincs nála; vagy ha rossz helyre telepíti.

* + - * Bemenet
      * *character place pump <karakter>*
      * Elvárt kimenet

*Successfully deployed pump.*

* + 1. Cső áthelyezése
       - Leírás

Adott cső bekötése a hálózatba.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek

Ellenőrizzük, hogy a szerelő jó helyen áll-e, és hogy van-e cső nála (ha bármelyik nem teljesül, azt jelzi a játékosnak). Hiba lehet, ha nem jó helyen teszi le a csövet, ha letesz egy csövet, pedig nincs nála, vagy ha nem tesz le csövet, pedig lehetne.

* + - * Bemenet
      * *character place pipe <karakter >*
      * Elvárt kimenet

*Successfully placed pipe.*

* + 1. Karakter mozgatás
       - Leírás

A karakterek mozgatása egyik node-ról a másikra.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek
      * Az adott karakter áthelyezését vizsgálja, ami történhet csőre, pumpára, valamint a szerelők esetében ciszterna elemre. Ehhez lekéri a szomszédos elemeket. Ha csőre szeretne lépni, akkor megvizsgálja azt is, hogy azon tartózkodik-e másik játékos. Ha igen, ezt üzenet jelzi. Egyéb esetben áthelyezi a játékost. Ha a lépés egy cső elemre történik, megnézi annak csúszósságát. Ha a cső csúszik, akkor a játékos annak egyik végéhez kapcsolódó elemre kerül. Ha a cső ragad, akkor a továbblépési idő számlálója elindul. Hiba lehet, ha mozgatja a játékost, pedig nem szabadna, vagy ha nem mozgatja a játékost, pedig az tudna a célra lépni.
      * Bemenet
      * *character move <karakter> <x koordináta> <y koordináta >*
      * Elvárt kimenet

*Successfully moved character.*

* + 1. Cső lyukasztása
       - Leírás

Adott karakter kilyukasztja a csövet, amin éppen áll.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek

Ellenőrizzük, hogy jó helyen áll-e, nincs-e már eltörve a cső, vagy a legutóbbi megjavítás óta eltelt idő még nem érte el a megadottat. Bármelyik feltétel nem teljesül, azt jelzi a játékosnak, egyéb esetben kilyukasztja a csövet. Hiba lehet, ha kilyukasztja a csövet, amit nem szabadna, vagy ami már lyukas, illetve ha nem lyukasztja ki, pedig ki szabadna.

* + - * Bemenet
      * *character break <karakter>*
      * Elvárt kimenet

*Successfully broken pipe.*

* + 1. Pumpa elállítása
       - Leírás

Adott karakter elállítja a pumpát, amin éppen áll.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek

Ellenőrizzük, hogy jó helyen áll-e, ha nem, azt jelezzük a játékosnak. Hiba lehet, ha nem értesíti a játékost hogy rossz helyen áll, vagy értesíti, pedig jó helyen állt, illetve ha nem állítja át a pumpát, pedig lehetne.

* + - * Bemenet
      * *character* *set <karakter>*
      * Elvárt kimenet

*Successfully set pump.*

* + 1. Pumpa javítása
       - Leírás

A szerelő megjavítja a pumpát, amin áll.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek

A szerelőt egy Pumpára helyezi és ellenőrzi, hogy a pumpa törött-e. Ha igen, végrehajtja a műveletet. Egyéb esetben jelzi a problémát. Hiba lehet, ha nem értesíti a játékost hogy nem pumpán áll, vagy ha nem javítja meg, pedig meg kellene.

* + - * Bemenet

*character*  *repair pump<karakter>*

* + - * Elvárt kimenet

*Successfully repaired pump.*

* + 1. Cső javítása
       - Leírás

A szerelő megjavítja a csövet, amin áll.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek

A szerelőt egy csőre helyezi és ellenőrzi, hogy a cső lyukas-e. Ha igen, végrehajtja a műveletet. Egyéb esetben jelzi a problémát. Hiba lehet, ha nem értesíti a játékost hogy nem pumpán áll, vagy ha nem javítja meg, pedig meg kellene.

* + - * Bemenet

*character*  *repair pipe <karakter>*

* + - * Elvárt kimenet

*Successfully repaired pipe.*

* + 1. Cső csúszóssá tétele
       - Leírás

A nomád szabotőr csúszóssá teszi a csövet, amin éppen áll.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek

A nomádot egy csőre helyezi és ellenőrzi, hogy az a cső csúszós-e. Ha nem csúszós csövön áll, végrehajtja a műveletet. Egyéb esetben jelzi a problémát. Hiba lehet, ha nem értesíti a játékost hogy nem csövön áll, vagy ha nem teszi csúszóssá, pedig kellene.

* + - * Bemenet

*character*  *slippery<karakter>*

* + - * Elvárt kimenet

*Successfully made pipe slippery. Slippery when wet.*

* + 1. Cső ragadóssá tétele
       - Leírás

A nomád szabotőr ragadóssá teszi a csövet, amin éppen áll.

* + - * Ellenőrzött funkcionalitás, várható hibahelyek

A nomádot egy csőre helyezi és ellenőrzi, hogy az a cső ragadós-e. Ha nem ragadós csövön áll, végrehajtja a műveletet. Egyéb esetben jelzi a problémát. Hiba lehet, ha nem értesíti a játékost hogy nem csövön áll, vagy ha nem teszi ragadóssá, pedig kellene.

* + - * Bemenet

*character*  *sticky<karakter>*

* + - * Elvárt kimenet

*Successfully made pipe sticky.*

###### A tesztelést támogató programok tervei

A tesztelést végző segéd program teszt eseteket olvas be majd futtatja az alkalmazást és végrehajtja a megadott teszt esetet. A teszt esetek az előbbiekben specifikált bemeneti nyelven vannak írva, illetve minden teszt esethez tartozik egy elvárt kimenet. A futtatást követően a tesztelő program összehasonlítja az elvárt kimenetet a tényleges kimenettel, ha ez nem egyezik, akkor a teszteset során valamilyen hibába botlottunk, ekkor a tesztelő jelzi hogy a teszt futása hibás volt. Abban az esetben ha meg egyezik az elvárt kimenet az aktuális kimenettel a tesztet sikeresnek titulálja a tesztelő program.

###### Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2023.04.26. 12:00 | 10 perc | Farkas  Kis  Sőregi  Nyilas | Értekezlet.  Heti munka felosztása |
| 2023.04.26. 12:15 | 2,5 óra | Farkas | Osztályok, metódusok dokumentálása. |
| 2023. 04. 26. 13:30 | 30 perc | Nyilas, Zsigmond | Tesztesetekről szóló értekezlet. |
| 2023. 04. 26. 14:00 | 3 óra | Nyilas | Tesztesetek kidolgozása. |
| 2023. 04. 26. 14:00 | 3 óra | Zsigmond | Tesztesetek dokumentálása. |
| 2023 04.29 11:00 | 1,5 óra | Farkas | Bonyolultabb, hosszabb metódusok pszeudo-kódos leírása |
| 2023. 04. 29. 15:30 | 1,5 óra | Zsigmond | Pszeudo-kódos leírások befejezése, néhány előzőleg megírt javítása |
| 2023. 05. 02 19:00 | 1 óra | Kis  Sőregi  Nyilas  Farkas | Záró értekezlet |

#### Prototípus beadása

###### Fordítási és futtatási útmutató

* + 1. Fájllista

|  |  |  |  |
| --- | --- | --- | --- |
| **Fájl neve** | **Méret** | **Keletkezés ideje** | **Tartalom** |
| Main.java | 17 334 | 2023. február 23. | Main fájl |
| Map.java | 12 617 | 2023. március 1. | Ez az osztály felelős a játék pályájáért. |
| ProtoTest.java | 10 551 | 2023. május 12. | A prototípus tesztelését valósítja meg. |
| Tickable.java | 122 | 2023. március 1. | Interfész a tick-elhető osztályoknak. |
| Character.java | 4 496 | 2023. március 1. | A játékban szereplő karakterekért felelős osztály. Absztrakt osztály az összes karakternek. |
| Nomad.java | 552 | 2023. március 1. | A nomád karaktert reprezentáló osztály. |
| Plumber.java | 4 073 | 2023. március 1. | A szerelő karaktert reprezentáló osztály. |
| InvalidPlayerActionException.java | 262 | 2023. március 1. | Kivételt dob, ha egy játékos érvénytelen műveletet próbál végrehajtani. |
| InvalidPlayerMovementException.java | 268 | 2023. március 1. | Kivételt dob, ha egy játékos érvénytelen pozícióba próbál mozogni. |
| NotFoundExeption.java | 255 | 2023. március 1. | Kivételt dob, ha egy játékos nem található a szomszédos elemen, ahova mozogni akar. |
| ObjectFullException.java | 227 | 2023. március 1. | Kivételt dob, ha egy objektum elérte maximum kapacitását. |
| NotJSONArray.java | 49 387 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONException.java | 1 067 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONObject.java | 86 517 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONPointer.java | 9 070 | 2023. május 2. | A map kiirasahoz tartozo |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | file beolvasasahoz sugseges osztaly |
| NotJSONPointerException.java | 556 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONPropertyIgnore.java | 664 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONPropertyName.java | 747 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONString.java | 745 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONTokener.java | 14 980 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONWriter.java | 12 869 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| Cistern.java | 1 503 | 2023. március  17. | Ciszterna node-ot reprezentáló osztály. |
| Node.java | 5 960 | 2023. március  17. | A Node osztály a pályára helyezhető összes objektum alaposztálya. |
| Pipe.java | 1 207 | 2023. március  17. | Cső node-ot reprezentáló osztály. |
| Pump.java | 2 227 | 2023. március  17. | Pumpa node-ot reprezentáló osztály. |
| WaterSpring.java | 1 323 | 2023. március  17. | Egy speciális node, amit a ciszterna feltöltéséhez használható. |
| tesztek.txt | 435 | 2023. május 14 | Teszt inputok |
| forditas.txt | 567 | 2023. 05. 14. | A fordításhoz a segítség |

* + 1. **Fordítás**

forditas terminalban:

mkdir .\out

cp .\lib\\*.jar .\out

javac -d .\out\ .\src\main\java\com\ez\_mode\Main.java .\src\main\java\com\ez\_mode\\*.java

.\src\main\java\com\ez\_mode\characters\\* .\src\main\java\com\ez\_mode\exceptions\\*

.\src\main\java\com\ez\_mode\objects\\* .\src\main\java\com\ez\_mode\notJson\\* -cp ".\lib\log4j-api-2.20.0.jar;.\lib\log4j-core-2.20.0.jar"

* + 1. Futtatás

futtatas terminalbol:

cd ./out

java -cp ";%CLASSPATH%;.\log4j-api-2.20.0.jar;.\log4j-core-2.20.0.jar" com.ez\_mode.Main

cmdből a cp-t át kell írni copyra, és a ;-t át kell írni :ra

###### Tesztek jegyzőkönyvei

* + 1. MapPrintTest

|  |  |
| --- | --- |
| **Tesztelő neve** | Sőregi Larina |
| **Teszt időpontja** | 2023.05.15.9:30 |

* + 1. **CharacterAddTest**

|  |  |
| --- | --- |
| **Tesztelő neve** | Farkas Bence |
| **Teszt időpontja** | 2023.05.15.10:01 |

* + 1. AddNewPumpTest

|  |  |
| --- | --- |
| **Tesztelő neve** | Sőregi Larina |
| **Teszt időpontja** | 2023.05.15.10:02 |

* + 1. **AddNewPipeTest**

|  |  |
| --- | --- |
| **Tesztelő neve** | Farkas Bence |
| **Teszt időpontja** | 2023.05.15.10:00 |

* + 1. PlacePumpTest

|  |  |
| --- | --- |
| **Tesztelő neve** | Sőregi Larina |
| **Teszt időpontja** | 2023.05.15.9:40 |

* + 1. **PlacePipeTest**

|  |  |
| --- | --- |
| **Tesztelő neve** | Farkas Bence |
| **Teszt időpontja** | 2023.05.15.10:06 |

* + 1. MakePipeStickyTest

|  |  |
| --- | --- |
| **Tesztelő neve** | Sőregi Larina |
| **Teszt időpontja** | 2023.05.15.9:45 |

* + 1. **MoveCharacterTest**

|  |  |
| --- | --- |
| **Tesztelő neve** | Farkas Bence |
| **Teszt időpontja** | 2023.05.15.10:08 |

* + 1. BreakPipeTest

|  |  |
| --- | --- |
| **Tesztelő neve** | Sőregi Larina |
| **Teszt időpontja** | 2023.05.15.9:50 |

* + 1. **SetPumpTest**

|  |  |
| --- | --- |
| **Tesztelő neve** | Farkas Bence |
| **Teszt időpontja** | 2023.05.15.10:10 |

* + 1. RepairPumpTest

|  |  |
| --- | --- |
| **Tesztelő neve** | Sőregi Larina |
| **Teszt időpontja** | 2023.05.15.9:55 |

* + 1. **RepairPipeTest**

|  |  |
| --- | --- |
| **Tesztelő neve** | Farkas Bence |
| **Teszt időpontja** | 2023.05.15.10:11 |

* + 1. **MakePipeSlipperyTest**

|  |  |
| --- | --- |
| **Tesztelő neve** | Sőregi Larina |
| **Teszt időpontja** | 2023.05.15.10:00 |

###### Értékelés

|  |  |  |
| --- | --- | --- |
| **Tag neve** | **Tag neptun** | **Munka százalékban** |
| Kis Benecek | JOYAXJ | 22% |
| Farkas Bence | D6AE9F | 21% |
| Sőregi Larina | M4B9MC | 19% |
| Nyilas Péter | Y8QBCS | 22% |
| Zsigmond Botond | MV7M6D | 16% |

###### Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2023.05.05. 12:00 | 2,5 óra | Farkas | Metódus implementációk |
| 2023. 05. 09. 17:00 | 3 óra | Nyilas | Metódus implementálások / javítások |
| 2023. 05. 10. 18:00 | 2 óra | Farkas | Javítások a kódban |
| 2023. 05. 10. 11:00 | 3,5 óra | Farkas | Teszt parancsok átgondolása, teszt beolvasó implementálása |
| 2023. 05. 10. 12:00 | 1 óra | Kis | A menü átírása |
| 2023. 05. 12 13:00 | 4 óra | Nyilas | Tesztek implementációja |
| 2023. 05. 13. 18:00 | 5 óra | Zsigmond | Metódus implementációk |
| 2023. 05. 13. 17:00 | 1 óra | Kis | A kimenet fileba írhatósága |
| 2023. 05. 14. 12:00 | 1 óra | Zsigmond | Tesztek javítása |
| 2023. 05. 14. 12:00 | 7 óra | Nyilas | metódusimplementáció, debug, tesztek átnézése, tesztek kipróbálása |
| 2023. 05. 14. 13:00 | 4,5 óra | Zsigmond | Tesztbemenetek leírása, tesztek javítása, metódus implementációk |
| 2023. 05. 14. 19:00 | 3 óra | Kis | A program rendberakása |
| 2023. 05. 14. 19:00 | 3 óra | Nyilas | A program rendberakása |
| 2023.05.14. 21:00 | 2 óra | Farkas Sőregi | Finalize+ tesztelés |

1. **Grafikus felület specifikációja**

###### A grafikus interfész

A szoftver elindításakor megjelenik a főmenü, ahol el tudjuk indítani a játékot:

Itt további

gombok is szerepelni fognak.

Ezen kívül, a különböző elemek a pályán:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| *Nomád szabotőr* | *Szerelő* | *Homok* | *Üres cső* |
|  |  |  |  |
| *Vízzel teli cső* | *Lyukas cső* | *Eltörés* | *Javítás* |
|  |  |  |  |
| *Vízzel teli pumpa* | *Üres pumpa* | *Rossz pumpa* | *Csúszós cső* |
|  |  |  |  |
| *Ragadós cső* | *Ciszterna* | *Vízforrás* | *Pumpa beállítása* |
|  |  |  |  |
| *Fel- és lefelé*  *mozgás* | *Jobbra és balra*  *mozgás* | *Cső fölvétele* | *Pumpa fölvétele* |

Egy minta a pálya kinézetéről:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **GAME End game** | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

###### A grafikus rendszer architektúrája

* + 1. A felület működési elve

A modell *push* alapú: bármilyen változás vagy esemény bekövetkezte (esetünkben egérrel kattintás) értesíti az összes osztályt a változásról, ami az adott osztályt érinti és a megfelelő metódust hívja meg.

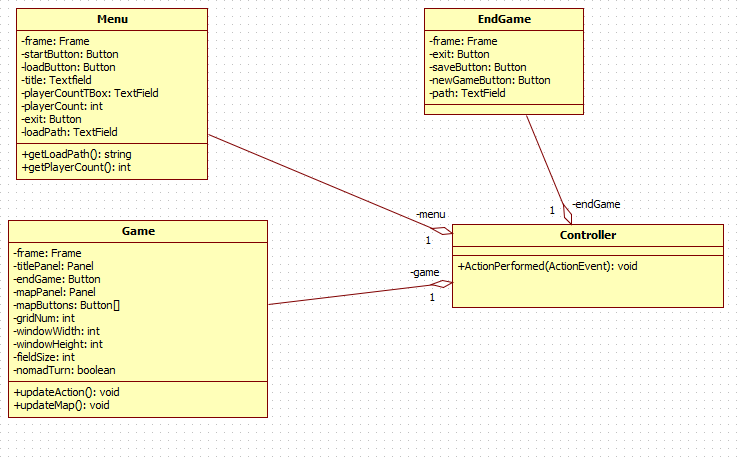
A program elindításakor egy menü indul el, ezért a *Menu* osztály felelős. Ennek a feladata a

játék elindítása, kilépés a játékból, játékosok hozzáadása, valamit egy meglévő játék betöltése. A *Controller* osztály az egyes gombok lenyomását figyelni. A menüt és a játékot is teljes egészében egérrel irányítjuk, ha rákattintunk egy gombra, meghívja az actionPerformed

metódust. A játék futása közben csak a legalsó sorban lévő ,,action bar”-on lévő

tevékenységek végezhetők el az adott karakterrel. Ez a nomád és a szerelő esetében változhat, például a szerelőnél repair ikon, a nomádnál a break ikon jelenik meg. Ez attól függően

változik, hogy melyik csapat következik, melyik csapat köre van éppen.

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

###### A grafikus objektumok felsorolása

* + 1. Controller
* Felelősség

Az osztály, amely figyeli a gombok eseményeit. Több metódus, de működésük elve hasonló.

* + 1. Game
* Felelősség

Magát a pályát kirajzoló és azt, valamint a játékos tevékenységét kezelő osztály.

* Attribútumok
  + **…ImagePath (pl. cisternImagePath)**: a képek helyei, amiket használunk a pálya elemeihez
  + **gridNum**: hogy mekkora a pálya
  + **windowWidth, windowHeight**: az ablak szélessége és magassága
  + **fieldSize**: hogy mekkora egy-egy mező a pályán
  + **frame**: maga az ablak
  + titlePanel
  + **textField**
  + mapPanel
  + **mapButtons**
  + endGame

ezek a különböző megjelenített dolgok, a gombok amikkel kezeljük a játékot, a panelek amiken láthatjuk az információkat, stb.

* + **…Icon (pl. cisternIcon)**: az ikonok, vagyis a pálya elemeinek kinézete
  + **nomadTurn**: egy változó, ami megmondja, melyik csapat köre van éppen
* Metódusok
  + **public void actionMap():** update-eli a pályán látható elemeket
  + **public void updateAction()**: update-eli az action bar elemeit
    1. Menu
* Felelősség

A program elindításakor megjelenő menüért felel.

* Attribútumok
  + **frame**: maga az ablak
  + **startButton**: a játék elindítására szolgáló gomb
  + **title:** a panel címe (vagyis, hogy „Menu”), majd, hogy melyik csapat köre
  + **loadButton**: előzőleg mentett pálya betöltése
  + **title**: kiírja azt, hogy Menu
  + **playerCountTBox**: Mennyi játékos lesz beírható ide
  + **playerCount**: Játékosszám
  + **exit**: kilépés gomb
  + **loadPath**: előzőleg mentett pálya elérési útjának megadása
* Metódusok
  + **public void getLoadPath():** elérési útja a pályának
  + **public void getPlayerCount():** játékosok (karakszerek/csapat) száma
    1. EndGame
* Felelősség

A játék végén megjelenő ablak. Itt lehet kilépni, új játékot indítani, menteni a pályát és ennek elérési útját megadni.

* Attribútumok
  + **frame**: maga az ablak
  + **exit**: gomb a játékból való kilépéshez
  + **saveButton:** pálya kimentése fájlba
  + **newGameButton:** új játékot indít, vagyis visszadob a start menübe
  + **path:** a mentendő fájl elérési útját ide kell megadni

###### Kapcsolat az alkalmazói rendszerrel

###### Napló

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2023. 05. 17. 18:00 | 2 óra | Sőregi | Pálya elrendezése megtervezése, pár ikon elkészítése |
| 2023. 05. 17. 16:45 | 1 óra | Zsigmond | A grafikus felülethez tartozó specifikáció elkezdése. |
| 2023. 05. 18. 20:00 | 0,5 óra | Sőregi | További ikonok megrajzolása |
| 2023. 05. 19. 16:00 | 0,5 óra | Zsigmond | A specifikáció frissítése. |
| 2023. 05. 20. 15:00 | 1,5 óra | Sőregi | Karakterek megrajzolása |
| 2023. 05. 20. 20:00 | 2 óra | Sőregi | Ábrák befejezése, action bar ikonok megtervezése és elkészítése |
| 2023. 05. 21. 12:00 | 2 óra | Zsigmond | Dokumentáció folytatása |
| 2023. 05. 21. 15:00 | 1 óra | Sőregi | Metódusok és attribútumok kigondolása és gyűjtése |
| 2023. 05. 21. 16:00 | 10 perc | Farkas  Sőregi | Osztály és szekvencia diagramok megbeszélése, ötletelés |
| 2023. 05. 21. 16:15 | 2 óra | Farkas | Diagramok rajzolása |
| 2023. 05. 21. 18: 00 | 10 perc | Nyilas  Sőregi | Értekezlet Minta pálya tervezése és dokumentáció javítások |
| 2023. 05. 21. 18:30 | 30 perc | Sőregi | Minta pálya elkészítése |
| 2023. 05. 21. 19:00 | 1,5 óra | Kis  Nyilas  Sőregi | Értekezlet Végső simítások a dokumentáción |

**13 Grafikus változat beadása**

***13.1 Fordítási és futtatási útmutató***

**13.1.1 Fájllista**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fájl neve** | **Méret** | **Keletkezés ideje** | **Tartalom** |
| Main.java | 17,8 KB (18 280 bytes) | ‎2023. február 23. | Main fájl |
| Map.java | 14,1 KB (14 495 bytes) | 2023. március 1. | Ez az osztály felelős a játék pályájáért. |
| Tickable.java | 122 bytes (122 bytes) | 2023. március 1. | Interfész a tick-elhető osztályoknak. |
| Character.java | 4,47 KB (4 584 bytes) | 2023. március 1. | A játékban szereplő karakterekért felelős osztály. Absztrakt osztály az összes karakternek. |
| Nomad.java | 597 bytes (597 bytes) | 2023. március 1. | A nomád karaktert reprezentáló osztály. |
| Plumber.java | 8,45 KB (8 660 bytes) | 2023. március 1. | A szerelő karaktert reprezentáló osztály. |
| InvalidPlayerActionException.java | 262 bytes (262 bytes) | 2023. március 1. | Kivételt dob, ha egy játékos érvénytelen műveletet próbál végrehajtani. |
| InvalidPlayerMovementException.java | 268 bytes (268 bytes) | 2023. március 1. | Kivételt dob, ha egy játékos érvénytelen pozícióba próbál mozogni. |
| NotFoundExeption.java | 255 bytes (255 bytes) | 2023. március 1. | Kivételt dob, ha egy játékos nem található a szomszédos elemen, ahova mozogni akar. |
| ObjectFullException.java | 227 bytes (227 bytes) | 2023. március 1. | Kivételt dob, ha egy objektum elérte maximum kapacitását. |
| NotJSONArray.java | 49 387 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONException.java | 1 067 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONObject.java | 86 517 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONPointer.java | 9 070 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
|  |  |  |  |
| NotJSONPointerException.java | 556 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONPropertyIgnore.java | 664 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONPropertyName.java | 747 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONString.java | 745 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONTokener.java | 14 980 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| NotJSONWriter.java | 12 869 | 2023. május 2. | A map kiirasahoz tartozo file beolvasasahoz sugseges osztaly |
| Cistern.java | 2,91 KB (2 980 bytes) | 2023. március 17. | Ciszterna node-ot reprezentáló osztály. |
| Node.java | 7,27 KB (7 452 bytes) | 2023. március 17. | A Node osztály a pályára helyezhető összes objektum alaposztálya. |
| Pipe.java | 4,43 KB (4 543 bytes) | 2023. március 17. | Cső node-ot reprezentáló osztály. |
| Pump.java | 4,35 KB (4 459 bytes) | 2023. március 17. | Pumpa node-ot reprezentáló osztály. |
| WaterSpring.java | 2,22 KB (2 279 bytes) | 2023. március 17. | Egy speciális node, amit a ciszterna feltöltéséhez használható. |
| Controller.java | 12,7 KB (13 047 bytes) | 2023. május 25. | A kezelőfelület eseményeit/akcióit kezelő osztály |
| Game.java | 2,85 KB (2 923 bytes) | 2023. május 25. | Játék grafikus megjelenítéséért felelős osztály. |
| EndGame.java | 19,6 KB (20 082 bytes) | 2023. május 26 | Játék végét jelző, grafikus felület. |
| Menu.java | 4,96 KB (5 087 bytes) | 2023 május 26 | Játék elején megjelenő grafikus menü |
| PopUp.java | 2,44 KB (2 501 bytes) | 2023. május 27 | Játék közben olyan akciókhoz használt grafikus elem, amikor egy adott mezőn a játékosnak döntenie kell (pl. merre állítja a pumpát) |
| forditas.txt | 549 bytes (549 bytes) | 2023. 05. 14. | A fordításhoz a segítség |

**13.1.2 Fordítás és telepítés**

forditas terminalban:

mkdir .\out\src\main\resources\

cp .\src\main\resources\\* .\out\src\main\resources\

javac -d .\out\ .\src\main\java\com\ez\_mode\Main.java .\src\main\java\com\ez\_mode\\*.java .\src\main\java\com\ez\_mode\characters\\* .\src\main\java\com\ez\_mode\exceptions\\* .\src\main\java\com\ez\_mode\objects\\* .\src\main\java\com\ez\_mode\notJson\\* .\src\main\java\com\ez\_mode\gui\\*.java

**13.1.3 Futtatás**

futtatas terminalbol

cd ./out

java -cp ";%CLASSPATH%" com.ez\_mode.Main

cmdből a cp-t át kell írni copyra, és a ;-t át kell írni :ra

***13.2 Értékelés***

|  |  |  |
| --- | --- | --- |
| **Tag neve** | **Tag neptun** | **Munka százalékban** |
| Kis Benedek | JOYAXJ | 22% |
| Farkas Bence | D6AE9F | 21% |
| Sőregi Larina | M4B9MC | 22% |
| Nyilas Péter | Y8QBCS | 21% |
| Zsigmond Botond | MV7M6D | 14% |

***13.3 Napló***

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2023. 05. 24 14:00 | 4 óra | Sőregi | Menü, grid és pályaméretezés |
| 2023. 05. 25 10:00 | 2 óra | Sőregi | Játékhoz használt képek programhoz adása. |
| 2023. 05. 24 12:00 | 1,5 óra | Zsigmond | Random pályagenerálás |
| 2023. 05. 25 15:00 | 4 óra | Sőregi | Grid méretezés, gombok és actionbar implementálás |
| 2023. 05. 25 10:00 | 2 óra | Farkas | Véletlenszerű pályagenerálás felülvizsgálata |
| 2023. 05. 25 8:00 | 6 óra | Nyilas | Frotend-backend integráció |
| 2023. 05. 26 8:00 | 4 óra | Sőregi | Menü, EndGame és Controller osztályok kommentezése |
| 2023. 05. 26 10:00 | 4 óra | Kis | Move és Break actionok implementálása, javítása |
| 2023. 05. 26 10:00 | 4 óra | Nyilas | Tesztelés, debug és integráció folytatása |
| 2023. 05. 26 16:00 | 5 óra | Sőregi | Menüben játékos nevek és számok beolvasása, debugolás. |
| 2023. 05. 27. 11:00 | 5 óra | Farkas | Karakter specifikus actionok implementálása |
| 2023. 05. 27 13:00 | 5 óra | Sőregi | Actionok utáni frontend update implementálása |
| 2023. 05. 28 10:00 | 2 óra | Farkas | Pump vagy Pipe pickup vagy place lepések egy action alá vonása, inventory alapján |
| 2023. 05. 28 10:00 | 2,5 óra | Sőregi | Flow update, popup ablakok és SetPump action implementálás. |
| 2023. 05. 28 14:00 | 1,5 óra | Zsigmond | Összes dokumentáció összefűzése. |
| 2023. 05. 28 14:00 | 5 óra | Kis | Map grafikus kirajzolásának javítása. |
| 2023. 05. 29 17:00 | 1,5 óra | Kis | Járékos akciók kezelése a GUIn keresztül és elemek megjelenítése. |
| 2023. 05. 29 20:00 | 2,5 óra | Zsigmond | Map load/save menüben |
| 2023. 05. 30. 12:30 | 3,5 óra | Farkas | Pump és Pipe pickup és place actionok implementálása |
| 2023. 05. 30 17:00 | 3,5 óra | Kis | Map load/save javítás és debugolás |
| 2023. 05. 30 18:00 | 3 óra | Nyilas | Waterflow calculation rework |
| 2023. 05. 30 21:00 | 3,5 óra | Nyilas | Debuggolás |
| 2023. 05. 31 00:30 | 3,5 óra | Sőregi | Debuggolás |
| 2023. 05. 31 04:00 | 3,5 óra | Kis | Debuggolás |

**14. Összefoglalás**

***14.1 A projektre fordított összes munkaidő***

|  |  |
| --- | --- |
| **Tag neve** | **Munkaidő (óra)** |
| *Sőregi Larina* | 68 |
| *Nyilas Péter* | 58 |
| *Kis Benedek* | 60 |
| *Farkas Bence* | 56 |
| *Zsigmond Botond* | 38 |
| **Összesen** | **280** |

***A feltöltött programok forrássorainak száma***

|  |  |
| --- | --- |
| **Fázis** | **Kódsorok száma** |
| Szkeleton | 1141 |
| Prototípus | 1690 + 5245(NotJSON) |
| Grafikus változat | 3250 + 5245(NotJSON) |
| **Összesen** | **16571** |

***14.2 Projekt összegzés***

**14.2.1 Mit tanultak a projektből konkrétan és általában?**

Sokat kell még fejlődnünk a csapatmunka terén.

**14.2.2 Mi volt a legnehezebb és a legkönnyebb?**

Legnehezebb: rossz volt az időbeosztásunk így sokszor maradt túl sok dolog az utolsó pillanatra  
Legkönnyebb: dokumantáció elkészítése.

**14.2.3 Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?**

Nem, az elején hosszabb időintervallum alatt kaptunk egy bizonyos mennyiségét, de a végén már ugyamazt az extra pontmennyiséget harmada annyi idő alatt katuk meg.

**14.2.4 Ha nem, akkor hol okozott ez nehézséget?**

Az utolsó két nagybeadás előtt.

**14.2.5 Milyen változtatási javaslatuk van?**

Több segédanyagot, és konkrétabban megfogalmazott heti feladatkiírásokat.

**14.2.6 Milyen feladatot ajánlanának a projektre?**

-

**14.2.7 Egyéb kritika és javaslat**Lehetnének részenkénti konkrét eligazítások amivel ellenőrizni tudjuk hogy jó irányba haladunk, és nem rontottuk el teljesen az addigi munkát.  
Verziókezelésről, csapatmunkáról lehetne több szó/segédanyag.