

Egyesített dokumentáció

55 – macaroni

Konzulens:
Simon Balázs

Csapattagok

Kereszthury Péter
Szilczer Ármin
Tóth Gábor Tamás
Galgóczy Gábor
Harsányi Levente

PUKAU0
THGPZB
HAFLCT
QS2WJR
FD7NQC

kereszthurypeter@gmail.com
arminszi1czer@gmail.com
gabor7d2@gmail.com
ggabor2002@gmail.com
kockalakoo@gmail.com

2023.06.02.

2. Követelmény, projekt, funkcionalitás

2.1 Bevezetés

2.1.1 Cél

A megrendelő és a fejlesztőcsapat programmal kapcsolatos követelményeinek rögzítése, a program alapvető funkcionalitásainak bemutatása.

2.1.2 Szakterület

A szoftver saját, otthoni felhasználásra tervezett játékprogram, amelynek célja a szórakoztatás.

2.1.3 Definíciók, rövidítések

- **Discord:** Szoftver, amivel lehet üzenetváltást végezni célszeméllyel, vagy akár egy csoporttal, ezenkívül lehet hanghívást és videohívást is végezni.
- **felhő:** interneten keresztül elérhető számítógép(ek), ami(ke)n szoftvert lehet futtatni.
- **Github:** Online szolgáltatás, amivel fájlokat, legfőképpen forrásfájlokat lehet szinkronban tartani emberek számítógépei között.
- **Google Dokumentumok:** Online dokumentumszerkesztő- és kezelő szolgáltatás, ami Google Drive-ra menti a dokumentumokat.
- **Google Drive:** online tárhelyet biztosító szolgáltatás.
- **IL408:** Terem a BME I épületében, ahol hetente a megrendelővel személyesen konzultál a fejlesztőcsapat.
- **IntelliJ IDEA:** Fejlesztőkörnyezet, ami többek között Java nyelven tesz lehetővé szoftverfejlesztést.
- **Java:** Programozási nyelv.
- **JDK:** Java Development Kit, programok és könyvtárak gyűjteménye, amik lehetővé teszik a Java nyelven történő szoftverfejlesztést és Java programok fordítását, futtatórendszerrel (JRE-t is tartalmaz).
- **JRE:** Java Runtime Environment, programok és könyvtárak gyűjteménye, amik lehetővé teszik a Java nyelven készített programok futtatását a számítógépen.
- **StarUML:** UML modellezéshez használt szoftver.
- **Trello:** Online szolgáltatás (weboldal) teendők csapaton belüli kiosztására, nyomon követésére.

2.1.4 Hivatkozások

- **Discord:** <https://discord.com>
- **Github:** <https://github.com>
- **Google Dokumentumok:** <https://docs.google.com>
- **Google Drive:** <https://drive.google.com>
- **IntelliJ IDEA:** <https://www.jetbrains.com/idea>
- **Kari felhő:** <https://niif.cloud.bme.hu>
- **StarUML:** <https://staruml.io>
- **Tárgyhonlapbeli feladatkiírás:** <https://www.iit.bme.hu/file/11582/feladat>
- **Trello:** <https://trello.com>

2.1.5 Összefoglalás

Jelen dokumentum a továbbiakban az alábbiakat tartalmazza:

- A 2. fejezetben ismerteti a program alapvető működését, a játékmenet módját és szabályait.
- A 3. fejezet kitér a funkcionális, erőforrásokkal, átadással kapcsolatos és egyéb kategóriákba sorolható követelményekre.
- A 4. fejezet a use-case-eket ismerteti.
- Az 5. fejezetben, a szótárban szerepel a magyarázata a nem köznyelvi kifejezéseknek és bármi másnak, ami a dokumentumban nem lenne egyértelmű.
- A 6. fejezet a projekt elkészítésének tervezett módját fejti ki. Összegyűjti a szoftvereket és rendszereket, amiket dokumentum- és verziókezeléshez, csoportmunkához, szoftverfejlesztéshez a fejlesztő csapat használni fog.
- A 7. fejezetben a projekttel kapcsolatos eddigi tevékenységek és értekezletek naplózása található.

2.2 Áttekintés

2.2.1 Általános áttekintés

- A szoftver az operációs rendszerre támaszkodva képes ablakot létrehozni, és azon grafikus elemeket megjeleníteni. (*grafikus alrendszer*)
- A szoftverrel együtt járhatnak grafikus és nem grafikus forrásfájlok is. Ezeket a program futása során megpróbálhatja írni/olvasni. A szoftverhez tartozó fájlok és almappák külső szerkesztése a program sikertelen működését okozhatja. (*fájlkezelő alrendszer*)
- A program az általa létrehozott (grafikus) ablakon keresztül képes ezen ablakhoz tartozó események feldolgozására, ezzel meghatározva a felhasználó és a szoftver közötti kapcsolatot. (*eseménykezelő alrendszer*)
- A kész szoftver internetkapcsolat nélkül is igénybe vehető.
- A felhasználó személyes adatait semmilyen formában nem tárolja.

2.2.2 Funkciók

A játék a fiktív közel-keleti Drukmákor térségben játszódik, aminek meghatározó része a közepén elterülő hatalmas sivatag. A térséget szárazság sújtja már évtizedek óta, víz pedig csak a magas hegyek forrásaiból csordogál. A városok lakói csak a veszélyes sivatagon keresztül juthatnak vízhez, ám vannak, akiknek ez nem érdekük...

A drukmákorai sivatagon át bonyolult csőrendszer szállítja a vizet a hegyi forrásokból a sivatagon túl elterülő városok ciszternáiba. A csőrendszer egyszerű, elágazás nélküli csövekből és a csövekhez csatlakozó aktív elemekből (forrás, ciszterna, napelemmel működő vízátemelő pumpa stb.) áll. Egy pumpa több (de a pumpára jellemző véges számú) csövet is összecsatlakoztat, és minden pumpán külön-külön állítható, hogy éppen melyik bele csatlakoztatott csőből melyik másik csőbe pumpáljon, azonban egyszerre csak egy bemenete és egy kimenete lehet. A többi rácsatlakoztatott cső eközben el van zárva. A pumpák véletlen időközönként el tudnak romlani, ilyenkor megszűnik az adott pumpánál a vízáramlás. A pumpák mindegyike rendelkezik egy víztartályal, amit a víz átemelése közben használ átmeneti tárolóként. A pumpa csak akkor tud vizet pumpálni egy csőbe, ha a cső szabad kapacitása ezt lehetővé teszi. A pumpa csak akkor tud vizet pumpálni a bemeneti csövből a víztartályába, ha a víztartályának kapacitása ezt lehetővé teszi, és van bemeneti csöve.

A csőhálózat bővíthető, változtatható. A csövek kellően rugalmasak ahhoz, hogy az egyik végüket lecsatlakoztatva egy másik aktív elemhez elvihetők és ott felcsatlakoztathatók legyenek. A ciszternáknál folyamatosan készülnek az új csövek, amelyek egyik vége a ciszternához kapcsolódik, a másik azonban szabad. A csövek szabad végéből a csőbe táplált víz a földre folyik. A nagy szárazság miatt az elfolyt vizet azonnal felissza a talaj, a városlakók szerencséjére viszont a ciszternákból már nem folyik ki a víz és a hegyi források sem apadnak el soha.

A csőhálózatot a szerelők tartják karban. Ők javítják meg az elromlott pumpákat, ők állítják át a pumpákat, hogy minden a lehető legtöbb víz tudjon áthaladni a hálózaton, és ha egy cső kilyukad, az ő dolguk a cső megfoltzása is. A kilyukadt csövekből a víz kifolyik, a csövek végén lévő pumpához már nem jut belőle. A szerelők dolga a ciszternáknál lévő szabad csövekkel a hálózat kapacitásának növelése. A szerelők a ciszternára állva felvehetik a szabad végét a csőnek, minden más esetben csőfelvételnél azt a végét veszik fel a csőnek, ami ahhoz az aktív elemhez csatlakozik, amin állnak. Egy szerelő egyszerre csak egy szabad csővéget tud cipelni és máshova felcsatolni. A szerelők a ciszternáknál magukhoz tudnak venni új pumpát is, amit egy cső közepén tudnak elhelyezni, ha arra ráállnak. A csövet ehhez ketté kell vágni, és a két végét a pumpához kell csatlakoztatni. Ez automatikusan megtörténik, viszont a pumpa bemeneti és kimeneti csövét ettől még be kell állítani, alapértelmezetten minden csatlakoztatott cső zárva van.

A hálózaton élnek a nomád szabotőrök is, akik a pumpákat tudják átállítani és a csöveket szokták kilyukasztani. A szabotőrök a földjeiken áthúzódó vízhálózatot támadják, a szerelőkkel szemben nem hosztílisak.

Mivel a sivatag veszélyes hely, a szerelők és a szabotőrök csak a csőhálózaton haladhatnak. A pumpáknál kikerülhetik egymást, de a csöveken már nem tudnak elmenni egymás mellett, egy csövön egyszerre csak egy ember állhat. Bár a csövek rugalmasak, lecsatlakoztatott végű állapotban kényesek a terhelésre, ezért ekkor nem állhat rajtuk senki.

A játékot a két csapat legalább 2-2 játékossal játszza. A szabotőrök dolga, hogy minél több víz folyjon ki a földre, a szerelők pedig azon dolgoznak, hogy minél több víz jusson a ciszternákba. Az a csapat nyer, amelyik a játék végére több vizet szerez.

2.2.3 Felhasználók

A játék korhatárra való tekintet nélkül játszható, a betanulási ideje rövid. Legalább 4 játékos szükséges a játékhoz.

2.2.4 Korlátozások

A szoftvernek Java nyelven kell íródnia, és el kell tudnia futni egy mai gépen, amin rendelkezésre áll egy JRE. Az elkészült játékprogram csak és kizárolag helyi-többjátékos módú, ezért a játékosoknak egy gépen kell játszaniuk a játékot.

2.2.5 Feltételezések, kapcsolatok

- **Discord:** A csapattagok közötti kommunikációra és az értekezletek megtartására használjuk.
- **Github:** A program forráskódjának megosztására használjuk a csapat tagjai között.
- **Google Dokumentumok:** Dokumentumok írására használjuk.
- **Google Drive:** Dokumentumok és egyéb fájlok tárolására és megosztására használjuk.
- **IntelliJ IDEA:** A program Java nyelven történő fejlesztésére használjuk.

- **Kari felhő:** Lásd: EEK1 követelmény.
- **StarUML:** UML modellezésre használjuk.
- **Tárgyhonlapbeli feladatkiírás:** A játék működésének alapjául vett leírás.
- **Trello:** A projekttel kapcsolatos teendők kezelésére használjuk.

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Koment
FLOW01	Forrásokból csöveken és pumpákon keresztül ciszternákba és a földre víz folyik.	Nem jelenik meg víz az említett elemeken kívül.	Alapvető	Megrendelő	Control waterflow	-
FLOW02	A forrásokból végtelen mennyiségű víz képes kifolyni.	A forrásokban mindenkorban víz keletkezni.	Alapvető	Fejlesztőcsapat	Control waterflow	-
FLOW03	A ciszternák és a föld végtelen mennyiségű vizet képes befogadni.	Nem telítődik sem a föld, sem a cisterna.	Fontos	Fejlesztőcsapat	Control waterflow	-
FLOW04	A ciszternákból és a földből nem folyhat ki víz.	A bennük lévő víz száma nem csökken.	Alapvető	Fejlesztőcsapat	Control waterflow	-
FLOW05	A csövek aktív elemhez csatlakozhatnak.	Létezhet cső, ami csatlakozik aktív elemhez.	Alapvető	Megrendelő	Move pipe end, Place pump	-
FLOW06	Egy aktív elemhez bármennyi cső csatlakozhat.	Nincs korlátja egy aktív elemnél a hozzá kapcsolt csöveknek.	Fontos	Megrendelő	View Map	-
FLOW07	A forrásra csatlakoztatott csőbe a forrásból a víz folyik, amíg a cső nincs tele.	Nem fordul elő olyan eset amikor egy csőbe folyhatna víz, de nem folyik.	Fontos	Megrendelő	Control waterflow	-

FLOW08	Egy pumpa egy bemeneti csőből a víztartályán keresztül egy kimeneti csőbe pumpál vizet, a többi cső le van zárva.	Csak a bemeneti csőből vesz ki és csak a kimeneti csőbe tesz be.	Alapvető	Megrendelő	Control waterflow, Configure pump	-
FLOW09	A pumpában a kimeneti és bemeneti cső nem lehet azonos.	A pumpa átállításakor nem fordul elő olyan, hogy a kettő azonos	Opcionális	Fejlesztőcsapat	Control waterflow, Configure pump	-
FLOW10	A pumpa csak akkor tud vizet pumpálni a bemeneti csőből a víztartályába, ha a víztartályának kapacitása ezt lehetővé teszi, és van bemeneti csöve.	A pumpa nem pumpál magába vizet, ha a kapacitása ezt nem engedi meg.	Alapvető	Megrendelő	Control waterflow	-
FLOW11	A pumpa csak akkor tud vizet pumpálni a kimeneti csőbe, ha van kimeneti csöve, a cső szabad kapacitása ezt lehetővé teszi, és van a pumpa víztartályában víz.	A pumpa nem pumpál ki vizet, ha a kimeneti cső kapacitása ezt nem engedi meg.	Alapvető	Megrendelő	Control waterflow	-
MOVE01	A karakterek a csöveken és az aktív elemeken mozoghatnak.	Mozgáskor nem kerülnek más hova.	Alapvető	Megrendelő	Move	-
MOVE02	Egy karakter csak olyan csőre vagy aktív elemre léphet, amely szomszédos jelenlegi pozíójával.	Mozgáskor nem lépnek nem szomszédos elemre.	Fontos	Fejlesztőcsapat	Move	-

MOVE03	Egy csövön egyszerre egy karakter állhat.	Amikor egy karakter megpróbál átmozogni egy csőre, amin már áll valaki, nem engedi a rálépést.	Fontos	Megrendelő	Move	-
MOVE04	Egy aktív elemen bármennyi karakter állhat.	Aktív elemre lépő karaktert nem utasítanak vissza.	Fontos	Megrendelő	Move	-
PUBR01	A pumpák bizonyos időközönként el tudnak romlani.	A megrendelő ellenőri leadáskor.	Fontos	Megrendelő	Break pump	-
PUBR02	Az elromlott pumpák nem pumpálnak vizet.	Egy elromlott pumpán nem megy át víz.	Alapvető	Megrendelő	Control waterflow	-
PUBR03	Az elromlott pumpákat a szerelők meg tudják javítani.	Amikor egy szerelő egy elromlott pumpán áll, képes azt megjavítani, ez után képes pumpálni.	Alapvető	Megrendelő	Fix network	-
PUBR04	A szerelők csak azt a pumpát javíthatják meg, amin éppen állnak.	Amikor egy szerelő megpróbál javítani egy pumpát nem javul meg másik pumpa, csak az amelyiken áll.	Opcionális	Fejlesztőcsapat	Fix network	-
PUCG01	A karakterek beállíthatják a pumpák kimeneti és bemeneti csövét.	A karakter egy adott pumpán kiválaszthat két csövet, ekkor ez a két cső lesz a bemeneti és kimeneti cső.	Alapvető	Megrendelő	Configure pump	-

PUCG02	A karakterek csak annak a pumpának állíthatják a kimeneti és bemeneti csövét, amin éppen állnak.	Amikor a karakter pumpát átállít az és csak az a pumpa állítódik át, amin áll.	Opcionális	Fejlesztőcsapat	Configure pump	-
PIBR01	A szabotörök kilyukaszthatják a csöveket.	Amikor egy szabotör megpróbálja kilyukasztani azt a csövet amin áll, az a cső lyukas lesz.	Alapvető	Megrendelő	Break pipe	-
PIBR02	A szabotörök csak azt a csövet lyukaszthatják ki, amin éppen állnak.	Amikor egy szabotör kilyukaszt egy csövet az és csak az a cső lyukad ki, amin áll.	Opcionális	Fejlesztőcsapat	Break pipe	-
PIBR03	A kilyukasztott csőben lévő víz a földre folyik.	Amikor egy cső kilyukad, a csőből a víz eltűnik, és a földön megjelenik, a földre folyt víz számlálóját növelte.	Alapvető	Megrendelő	Control waterflow	-
PIBR04	A kilyukasztott csőből nem lehet vizet kipumpálni.	Amikor egy lyukas cső egy pumpa bemenete, akkor abból a pumpa nem képes kipumpálni vizet.	Alapvető	Megrendelő	Control waterflow	-
PIBR05	A szerelők megjavíthatják a kilyukasztott csöveket.	Amikor egy szerelő egy lyukas csövön állva megpróbálja azt megjavítani, az többé nem lesz lyukas.	Alapvető	Megrendelő	Fix network	-

PIBR06	A szerelők csak azt a csövet javíthatják meg, amin éppen állnak.	Amikor egy szerelő megpróbál javítani egy csövet nem javul meg másik cső, csak az amelyiken áll.	Opcionális	Fejlesztőcsapat	Fix network	-
PICR01	A cisternáknál bizonyos időközönként új csövek jönnek létre.	A megrendelő ellenőrzi leadáskor.	Fontos	Megrendelő	Spawn pipes	-
PICR02	A cisternánál létrejött új cső egyik vége a cisternához csatlakozik, másik vége szabad.	A megrendelő ellenőrzi leadáskor.	Fontos	Megrendelő	Spawn pipes	-
PIMO01	A csövek végét a szerelők le tudják csatlakoztatni, ha nem áll rajta egy karakter se, ez a cső vég a szerelő kezébe kerül.	Ha áll rajta valaki, nem tudja lecsatlakoztatni. Lecsatlakozás után a csővég a szerelő kezében van.	Alapvető	Fejlesztőcsapat	Move pipe end	-
PIMO02	A csőbe betáplált víz a cső szabad végéből a földre folyik.	Ha egy szabad végű csőbe víz folyik, akkor az onnan eltűnik, és a víz a földre kerül. Ezzel növelve a szabotőrök pontszámát.	Alapvető	Megrendelő	Control waterflow	-
PIMO03	A lecsatlakoztatott végű csövön nem állhat karakter.	Ha karakter áll rajta, akkor nem lehet lecsatlakoztatni.	Opcionális	Fejlesztőcsapat	Move, Move pipe end	-

PIMO04	A csőnek csak az a vége csatlakoztatható le, amely vége arra az aktív elemre csatlakozik, amelyen a cső végét lecsatlakoztató szerelő áll, és a cső nem új cső.	Amikor a szerelő megpróbálja lecsatlakoztatni a csövet, akkor az és csak az a cső kerül lecsatlakoztatásra.	Opcionális	Fejlesztőcsapat	Move pipe end	-
PIMO05	A ciszternához csatlakozó új cső lecsatlakoztatása kor a cső szabad vége kerül a szerelő kezébe, és az új csőből rendes cső lesz.	Amikor a szerelő megpróbálja lecsatlakoztatni a csövet, akkor annak csőnek a szabad vége kerül a szerelő kezébe.	Fontos	Fejlesztőcsapat	Move pipe end	-
PIMO06	A cső lecsatlakoztatott végét a szerelő magával viszi mozgás közben.	A felvett cső a lecsatlakoztatást végző szerelő kezében marad.	Alapvető	Megrendelő	Move pipe end	-
PIMO07	A lecsatlakoztatott végű csőből nem lehet vizet kipumpálni.	Amikor egy lecsatlakoztatott végű cső egy pumpa bemenete, akkor abból a pumpa nem képes kipumpálni vizet.	Alapvető	Megrendelő	Move pipe end, Control waterflow	-
PIMO08	Egy szerelő egyszerre egy lecsatlakoztatott csővéget vihet magával.	Amikor egy szerelőnél már van lecsatlakoztatott cső vég nem áll rendelkezésére egy másik cső lecsatlakoztatása.	Opcionális	Fejlesztőcsapat	Move pipe end	-

PIMO09	A szerelő a magával vitt lecsatlakoztatott csővéget csatlakoztathatja arra az aktív elemre, amelyen éppen áll.	Amikor egy szerelő kezében van csővég, és aktív elemen áll, választhatja a cső felcsatlakoztatása opciót. Ekkor az csatlakoztatva lesz azzal az aktív elemmel, ahol a szerelő áll.	Alapvető	Fejlesztőcsapat	Move pipe end	-
PIMO10	A lecsatlakoztatott cső vége csak akkor csatlakoztatható az adott aktív elemre, ha a cső másik vége nincsen ugyanoda csatlakoztatva.	Amikor ezek a feltételek fennállnak, a szerelő nem kapja meg a lecsatlakoztatás opciót.	Opcionális	Fejlesztőcsapat	Move pipe end	-
PUCR01	A szerelők a ciszternáknál magukhoz vehetnek egy pumpát.	Amikor a szerelő egy ciszternán áll, lehetőségére áll ez az opció, amit ha választ a kezében megjelenik egy pumpa	Alapvető	Megrendelő	Acquire pump	-
PUPL01	A szerelők lehelyezhetnek pumpákat.	Ha van egy szerelőnél pumpa és egy csövön áll, akkor lehetőségére áll a pumpa lehelyezése opció.	Alapvető	Megrendelő	Place pump	-
PUPL02	A szerelők csak arra a csőre helyezhetnek le pumpát, amin éppen állnak.	Arra és csak arra a csőre kerül pumpa.	Opcionális	Fejlesztőcsapat	Place pump	-

PUPL03	A szerelők csak akkor helyezhetnek le pumpát, ha van náluk pumpa.	Egy szerelő nem helyezhet le pumpát, ha nincs a kezében.	Alapvető	Megrendelő	Place pump	-
PUPL04	A szerelők a pumpa lehelyezése után az új pumpára állnak.	A pumpa lehelyezése után a pumpát lehelyező szerelő helye csak és kizárolag az új pumpa lesz.	Opcionális	Fejlesztőcsapat	Place pump	-
PUPL05	Az újonnan lehelyezett pumpa a meglévő csövet kettévágja, az új csővégek az új pumpához csatlakoznak.	Az új pumpához két cső van csatlakoztatva . A csövek másik vége az eredeti helyre csatlakozik továbbra is.	Fontos	Megrendelő	Place pump	-
PUPL06	A cső kettévágása után az újonnan lehelyezett pumpában a csövek le vannak zárva.	A pumpa elhelyezésekor egyik cső sem funkcionál se kimeneti, se bemeneti csóként.	Fontos	Fejlesztőcsapat	Place pump	-
CTGM01	A játékot két csapat játsza: szerelők és szabotörök.	A játék indításakor ez a két csapat van jelen.	Alapvető	Megrendelő	Control teams	-
CTGM02	Mindkét csapatban legalább két játékosnak kell lennie.	Csak akkor indítható a játék, ha minden csapatban legalább 2 játékos van..	Opcionális	Megrendelő	Control teams	-
CTGM03	A szerelők a ciszternákba gyűjtik a vizet.	Nem szereznek máshonnan pontot. Amikor egy ciszternába kerül víz, a szerelők víz számlálója megnő.	Alapvető	Megrendelő	Control teams	-

CTGM04	A szabotörök a földön gyűjtik a vizet.	Nem szereznek máshonnan pontot. Amikor a földre kerül víz, a szabotörök víz számlálója megnő.	Alapvető	Megrendelő	Control teams	-
CTGM05	A csapatok az összegyűjtött víz mennyisége alapján pontokat kapnak.	Nem szereznek máshonnan pontot.	Alapvető	Megrendelő	Control teams	-
CTGM06	Az a csapat nyer, akinek a játék végén legtöbb pontja lesz.	A játék végén a játék kiírja a nyertest.	Alapvető	Megrendelő	Control teams	-

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
EKK1	A forrásprogram a kari felhőben biztosított környezetben, az ott megtalálható JDK alatt (parancssorból) lefordítható és futtatható legyen.	A megrendelő fogja ellenőrizni.	Alapvető	Megrendelő	-
EKK2	A szöveges dokumentumok szerkesztéséhez Google Dokumentumokat használunk.	A fejlesztőcsapat ellenőrzi.	Fontos	Fejlesztőcsapat	-
EKK3	A dokumentumok tárolására egy megosztott Google Drive-ot használunk.	A fejlesztőcsapat ellenőrzi.	Fontos	Fejlesztőcsapat	-
EKK4	A programkódok tárolására és verziókezelésére Githubot használunk.	A fejlesztőcsapat ellenőrzi.	Fontos	Fejlesztőcsapat	-
EKK5	Az UML diagramok elkészítéséhez StarUML-t használunk.	A fejlesztőcsapat ellenőrzi.	Fontos	Fejlesztőcsapat	-

EKK6	A tényleges program fejlesztéséhez IntelliJ IDEA-t fogunk használni.	A fejlesztőcsapat ellenőrzi.	Fontos	Fejlesztőcsapat	-
EKK7	A feladatok szétosztásához és követéséhez Trellot használunk.	A fejlesztőcsapat ellenőrzi.	Fontos	Fejlesztőcsapat	-
EKK8	Az értekezletek és a csapattagok közötti kommunikációhoz Discordot használunk.	A fejlesztőcsapat ellenőrzi.	Fontos	Fejlesztőcsapat	-
EKK9	A program helyes működéséhez kell egy billentyűzet, egér és monitor.	Bemutatás	Alapvető	Fejlesztőcsapat	-

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
VEGLE1	Az egyetemi félév végén (14. oktatási héten pénteken 13:00 óráig) leadjuk a megfelelő helyen a szoftver projekt laboratórium konzulensünknek.	A megrendelő (jelen esetben a konzulens) fogja ellenőrizni a végterméket.	Alapvető	Megrendelő	-
SZKEL1	ápr. 17. 14:15 -ig el kell készülni a feladat minimális szkeleton verziójával.	Megrendelő	Alapvető	Megrendelő	-
PROT1	máj. 8. 14:15 -ig el kell készülni a játék prototípusával.	Megrendelő	Alapvető	Megrendelő	-
GRAF1	jún. 2. 14.00 -ig el kell készülni a végleges grafikus verzióval.	Megrendelő	Alapvető	Megrendelő	-

2.3.4 Egyéb nem funkcionális követelmények

Nincs egyéb nem funkcionális követelmény.

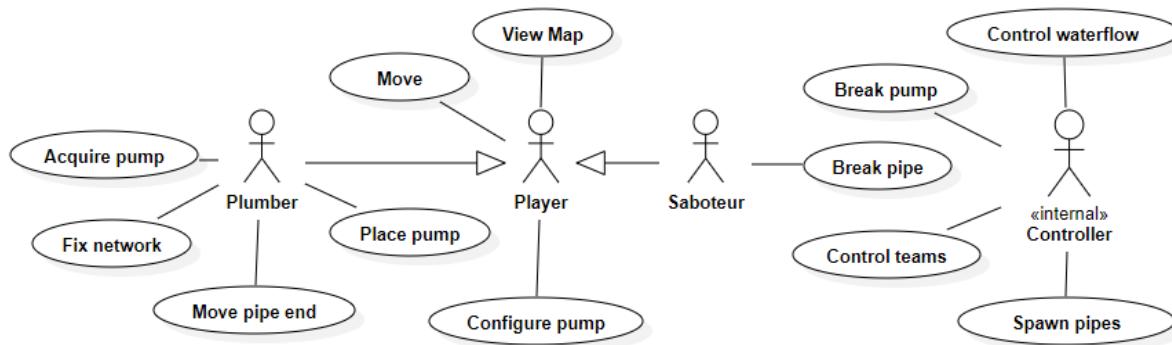
2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	Move
Rövid leírás	A játékosok a karaktereket mozgathatják a csőhálózaton keresztül.
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> A játékos karaktere jelenlegi pozíójával szomszédos helyre lép, ha van rajta szabad hely.
Use-case neve	Configure pump
Rövid leírás	A játékosok beállíthatják a kimeneti és bemeneti csövet azon a pumpán, amin éppen állnak.
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> A játékos beállítja a pumpa kimeneti és bemeneti csövét. A pumpára csatlakoztatott többi cső el lesz zárva.
Use-case neve	View Map
Rövid leírás	A játékos megtekintheti a pályát.
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> A rendszer kirajzolja a pálya aktuális állapotát. A játékos megtékinti a pálya aktuális állapotát.
Use-case neve	Place pump
Rövid leírás	A szerelők lehelyezhetnek egy pumpát arra a csőre, amin éppen állnak.
Aktorok	Plumber
Forgatókönyv	<ol style="list-style-type: none"> Ha a szerelőnél van pumpa, lehelyezi a csőre. Ekkor a cső kettéválik, és a pumpába csatlakozik.
Use-case neve	Acquire pump
Rövid leírás	A szerelő a cisternán állva magához vehet egy pumpát.
Aktorok	Plumber
Forgatókönyv	<ol style="list-style-type: none"> A szerelő a cisternán magához vesz egy pumpát.
Use-case neve	Fix network
Rövid leírás	A szerelők megjavíthatják az elromlott pumpákat és a kilyukadt csöveget.
Aktorok	Plumber
Forgatókönyv	<ol style="list-style-type: none"> A szerelő megjavítja azt a kilyukadt csövet, amin éppen áll. A szerelő megjavítja azt az elromlott pumpát, amin éppen áll.
Use-case neve	Break pipe
Rövid leírás	A szabotörök kilyukaszthatják az ép csöveget.
Aktorok	Saboteur
Forgatókönyv	<ol style="list-style-type: none"> A szabotör kilyukasztja azt a csövet, amin éppen áll
Use-case neve	Move pipe end
Rövid leírás	A szerelők lecsatlakoztathatják a pumpákról a rá csatlakozó csövek végét, vagy felvehetnek egy cisternánál egy szabad csővéget, amit magukkal vihetnek, és lerakhatnak máshova.

Aktorok	Plumber
Forgatókönyv	<ol style="list-style-type: none"> 1. A szerelő lecsatlakoztatja egy csönek a végét, amin nem áll más játékos, és oda van csatlakoztatva, ahol éppen áll. 2. A szerelő lehelyezi a nála lévő lecsatlakoztatott csővéget oda, ahol áll.
Alternatív forgatókönyv	1.A.1 Ha a szerelő ciszternán áll, akkor lecsatlakoztatás helyett felvehet egy ott lévő szabad csővéget is.
Use-case neve	Spawn pipes
Rövid leírás	A ciszternáknál új csövek és új pumpák lesznek felvehetőek.
Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> 1. A ciszternáknál bizonyos időközönként szabad végű csövek jelennek meg.
Use-case neve	Break pump
Rövid leírás	Bizonyos időközönként a pumpák elromlanak.
Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> 1. Egy pumpa elromlik. Amíg nem lesz megjavítva, nem pumpál vizet.
Use-case neve	Control waterflow
Rövid leírás	A Controller irányítja a hálózatban a víz folyását, és üzemelteti a pumpákat.
Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> 1. A forrásokból a rájuk csatlakoztatott csövekbe folyik a víz. 2. A kilyukadt vagy lecsatlakoztatott végű csövekből kifolyik a földre a víz. 3. A működő pumpák átpumpálják a vizet a víztartályukból a kimeneti csövükbe. 4. A működő pumpák a bemeneti csövükből átpumpálják a vizet a víztartályukba. 5. A ciszternára csatlakoztatott csövekből a ciszternába folyik a víz.
Use-case neve	Control teams
Rövid leírás	A csapatokat kezeli, számon tartja a csapatok által szerzett pontokat, a játék végén győztest hirdet.
Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> 1. Számolja a játék közben összegyűjtött vízmennyiséget. 2. Az összegyűjtött vízmennyiség alapján pontokat számít a csapatoknak. 3. A játék végén a legtöbb ponttal rendelkező csapat nyer.

2.4.2 Use-case diagram



2.5 Szótár

- **Aktív elem:** Elem (ciszterna, forrás, vagy pumpa), amelyhez cső csatlakoztatható.
- **Áll:** Egy karakter a csőhálózat valamelyik elemén tartózkodik.
- **Bemeneti cső:** Egy pumpánál az a cső, ahonnan a pumpa pumpál.
- **Ciszterna:** Elem, ahonnan nem folyhat ki víz, csak befolyhat. A szerelők ide gyűjtik a vizet.
- **Csapat:** A játékban azonos karakter fajtával rendelkező játékosok halmaza (szerelők, szabotőrök). Az egy csapatban lévő játékosok közösen gyűjlik maguknak a vizet.
- **Csatlakozás:** Fizikai kapcsolat két elem között (a játék világában értelmezve). Karakterek mozoghatnak és víz folyhat két csatlakozott elem között.
- **Csatlakoztat:** A karakter csatlakozást létesít az előzőleg általa lecsatlakoztatott csővég azon aktív elem között, amin a karakter éppen áll.
- **Cső:** Korlátos kapacitással rendelkező elem, amelynek két végpontja van (lásd. cső vég) és minden végén csatlakozhat valamilyen aktív elemhez.
- **Csőhálózat:** A játék karaktereinek tartózkodási helye. Ezen keresztül folyik a víz, amit a csapatok később összegyűjtenek. Elemekből áll.
- **Cső vég:** Egy adott csőnek az a része, ahol egy aktív elemhez tud csatlakozni. minden cső két ilyennel rendelkezik.
- **Csövet kettévág:** A csőből két cső lesz. Az eredeti csővégek megmaradnak, az újonnan keletkezett csővégek szabadok.
- **Elem:** Objektum, ami a csőhálózat része.
- **Elromlik:** Egy pumpa elromlott pumpa állapotba kerül.
- **Elromlott Pumpa:** Pumpa, ami nem képes pumpálni.
- **Folyik:** Adott mennyiségű víz képes helyet változtatni egyik elemből másik elemben vagy a földre.
- **Forrás:** Elem, ahonnan víz folyik ki.
- **Föld:** A lyukas vagy szabad végű csőből ide folyik a víz. A szabotőrök ide gyűjtik a vizet.
- **Játék:** A végső szoftver termék játszható része.
- **Játékos:** A szoftver egy felhasználójának a játékban betöltött szerepe. minden játékos rendelkezik egy karakterrel, amit a játék folyamán irányíthat. A játékos karaktere irányításával szerezhet vizet
- **Kapacitás:** A maximálisan befogadható víz mértéke.
- **Karakter:** Képességekkel rendelkező entitás, a játék része. minden karakter mozoghat és pumpák ki- és bemenetét állíthatja. A karakter fajtája lehet szerelő vagy szabotőr, ezeknek vannak további képességeik.

- **Kéz:** Ahol egy szerelő képes lecsatlakoztatott cső véget és/vagy pumpát tartani.
- **Kilyukaszt:** Egy csövet megváltoztat, hogy lyukas cső legyen.
- **Kimeneti cső:** Egy pumpánál az a cső, ahova a pumpa pumpál.
- **Lecsatlakoztat:** A karakter megszüntet egy cső és aktív elem közötti csatlakozást, a cső ezen vége ettől kezdve szabad vég lesz.
- **Lecsatlakoztatott vég:** Lásd: szabad vég
- **Lyukas cső:** Cső, amiben tárolt víz a földre folyik.
- **Mozog:** Egy karakter megváltoztatja, hogy jelenleg melyik elemen áll. Csak szomszédos elemek között lehet mozogni.
- **Működő:** Képes rajta víz átfolyni.
- **Nyer:** Az adott csapat nyertes állapotba kerül.
- **Nyertes:** Az a csapat, amelyik több vizet szerez a neki kijelölt helyre.
- **Pumpa:** Víztartályal rendelkező aktív elem. Bemeneti és kimeneti cső között képes pumpálni, a többi cső ekkor zárva van.
- **Pumpál:** Vizet folyat egy elem és egy vele szomszédos pumpa tartálya között.
- **Szabad vég:** Egy cső olyan vége, ami nem csatlakozik semmilyen aktív elemhez sem.
- **Szabotőr:** Karakterfaj, aki képes csöveget kilyukasztani.
- **Szerelő:** Karakterfajta, aki képes pumpákat és lyukas csöveget megjavítani, új pumpát a kezébe venni ciszternánál, valamint aktív elemen állva lecsatlakoztatni egy csővéget és azt a kezébe venni. A kezében lévő pumpát le tudja rakni csövön állva, ami következetében a csövet kettévágja és a pumpát a 2 újonnan keletkezett szabad végű csőbe köti be, és a kezében lévő lecsatlakoztatott csővéget csatlakoztatni tudja.
- **Szomszéd:** Egy adott elemnek azon elemek a szomszédi, akik hozzá vannak csatlakoztatva.
- **Új cső:** Az a cső, ami újonnan a ciszternán keletkezett, aminek az egyik vége szabad, a másik vége a ciszternába van csatlakoztatva.
- **Vizet Gyűjt:** Olyan tevékenység végzése, ami az adott csapat nyerését elősegíti.
- **Víz:** A játék fő nyersanyaga. Forrásban generált egység, ami átmenetileg tárolódik és átfolyik (lásd. folyik) a csőhálózaton.
- **Víztartály:** Korláatos kapacitással rendelkező vizet tároló objektum.
- **Zárt cső:** Egy pumpánál lévő olyan cső, ami nem kimeneti vagy bemeneti csöve a pumpának. Ilyenkor a pumpa ezzel a csővel nem végez semmilyen interakciót.

2.6 Projekt terv

A projekt ütemterve

A projekt produktumait heti rendszerességgel állítjuk elő, és adjuk be a feltüntetett határidőre az alábbi ütemterv alapján:

Hét	Feladat	Határidő
2	Követelmény, projekt, funkcionalitás	március. 13. 14:15
3	Analízis modell (I. változat)	márc. 20. 14:15
4	Analízis modell (II. változat)	márc. 27. 14:15

5	Szkeleton tervezése	ápr. 3. 14:15
6-7	Szkeleton elkészítése	ápr. 17. 14:15
8	Prototípus koncepciója	ápr. 24. 14:15
9	Részletes tervezek	máj. 3. labor
10	Prototípus elkészítése	máj. 8. 14:15
11	Grafikus változat tervezek	máj. 15. 14:15
12-13	Grafikus változat elkészítése	máj 31. labor
14	Egyesített dokumentáció	jún. 2. 14.00

Az ütemtervben szereplő feladatokat a csapat tagjai közösen készítik el, az összes csapattag minden feladatért közösen vállal felelősséget. A feladatokon belüli pontos beosztás az adott feladat kezdésekor lesz kijelölve.

Kapcsolattartó: Kereszthury Péter

A projekt készítéséhez használandó eszközök, erőforrások

A csapat tagjai saját számítógépükön (mint erőforrás) használhatják a feladatok elvégzésére.

A készülő dokumentumokat, UML diagramokat és egyéb fájlokat **Google Drive-ban tároljuk**, dokumentumokat **Google Dokumentumokkal szerkesztünk**, és azzal exportáljuk ki nyomtatható formátumba (pdf fájlba). UML diagramokat **StarUML-el** szerkesztünk.

Discord használatával bonyolítjuk le a csapattagok közötti értekezleteket és üzenetváltásokat.

Trello segítségével osztjuk ki a csapattagok közötti elvégzendő **feladatokat**, és minden csapattag számon tartja a kiosztott feladataira, hogy **éppen dolgozik rajta**, hogy készen van és **más csapattagok átnézésére vár**, vagy már át lett nézve és teljesen **készen van** az adott feladat.

A programot **Java** nyelven fejlesztjük, **IntelliJ IDEA** használatával. A programhoz kapcsolódó fájlokat (forrásfájlok és egyéb projektfájlok) **Github-on** fogjuk tárolni, és az IntelliJ IDEA beépített Github interakciójával fogjuk szinkronban tartani az online Github projekttel. Ezzel azt is elérjük, hogy a csapattagok egymás kódját könnyedén meg tudják osztani és tudunk közösen dolgozni a programon.

2.7 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2023.03.08. 12:15	2 óra	Harsányi Tóth	Értekezlet: Konzultáltunk az IL408-ban a laborvezetővel. Döntés: értesíteni a csapat többi tagját a következő értekezlet idejéről
2023.03.08. 16:00	1 óra	Harsányi Kereszthury Tóth	Értekezlet: Megvitattuk a feladat szövegét. Döntés: a csapat többi tagja is értelmezi a feladat szövegét
2023.03.09. 17:30	1 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: Feladatak kiosztása. Döntés: Az alább felsorolt feladatakat elkészítjük a következő értekezletig. Harsányi 2.3.3 Kereszthury 2.3.1, 2.4.1, 2.4.2 Szilczer 2.1.2, 2.2.2, 2.2.3, 2.3.2 Tóth 2.1.1
2023.03.09. 18:30	1,5 óra	Kereszthury	Tevékenység: 2.3.1 Funkcionális követelmények kidolgozása.
2023.03.09. 23:30	2 óra	Szilczer	Tevékenység: 2.1.2 Szakterület 2.2.2 Funkciók kiegészítése 2.2.3 Felhasználók 2.3.2 Erőforrás követelmények
2023.03.09. 17:10	2 óra	Kereszthury	Tevékenység: 2.4.1 Use-casek 2.4.2 Use-case diagram kifejtése, elkészítése.
2023.03.09. 18:30	0,25 óra	Harsányi	Tevékenység: 2.3.2 Erőforrás követelmények EKK1 2.3.3 Átadási követelmények
2023.03.10. 20:00	2 óra	Galgóczy	Tevékenység: Eddig elkészült dokumentumok ellenőrzése (kivéve 2.4.1).
2023.03.11. 12:00	1 óra	Galgóczy	Tevékenység: 2.4.1 Use-casek ellenőrzése
2023.03.11. 14:00	0,25 óra	Tóth	2.1.1 Cél

2023.03.11. 21:00	1,5 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: Feladatak kiosztása Döntés: Mindenki befejezi az összes feladatát hétfő 11-ig. A feladatok: Galgóczy 2.2.1 Harsányi 2.3.1, 2.3.4, 2.5 Szilczer 2.2.4 Tóth 2.1.3, 2.1.4, 2.1.5, 2.2.5, 2.6
2023.03.11. 23:00	1 óra	Harsányi	Tevékenység: 2.5 Szótár 2.3.4 Egyéb nem funkcionális követelmények
2023.03.11. 23:00	2 óra	Tóth	Tevékenység: 2.1.5 Összefoglalás 2.1.4 Hivatkozások elkezdése 2.2.5 Feltételezések, kapcsolatok elkezdése fedlap elkészítése dokumentum átnézése
2023.03.12. 01:00	0,5 óra	Galgóczy	Tevékenység: 2.2.1 Általános áttekintés
2023.03.12. 13:30	1 óra	Tóth	Tevékenység: 2.1.3 Definíciók, rövidítések 2.1.4 Hivatkozások 2.1.5 Összefoglalás javítása 2.2.5 Feltételezések, kapcsolatok
2023.03.12. 14:00	2 óra	Szilczer	Tevékenység: 2.2.2 Funkciók kiegészítés 2.2.3 Felhasználók 2.2.4 Korlátozások 2.3.2 Erőforrásokkal kapcsolatos követelmények kiegészítése Elkészült feladatak átnézése, helyesírási hibák javítása
2023.03.12. 17:30	0,5 óra	Galgóczy	Tevékenység: 2.2.1 Általános áttekintés
2023.03.12. 18:30	2 óra	Tóth	Tevékenység: 2.6 Projekt terv Elkészült fejezetek ellenőrzése
2023.03.12. 20:30	4 óra	Tóth	Tevékenység: 2.3.1 Funkcionális követelmények átnézése, rendezése, javítása, azonosítóval ellátása
2023.03.12. 21:00	3 óra	Harsányi	Tevékenység: 2.5 Szótár
2023.03.12. 23:00	1 óra	Szilczer	Tevékenység: 2.2.2 Funkciók követelményekkel összhangba hozása

2023.03.13. 00:00	2 óra	Galgóczy	Tevékenység: 2.5 Szótár javítása
2023.03.13. 00:15	3 óra	Harsányi	Tevékenység: 2.3.1 Funkcionális követelmények ellenőrzés, prioritás, forrás, use-case
2023.03.13. 02:00	1 óra	Tóth	Tevékenység: 2.5 Szótár ellenőrzése, javítása
2023.03.13. 10:00	0,5 óra	Tóth	Tevékenység: 2.3.1 Funkcionális követelmények ellenőrzések átnézése.
2023.03.13. 11:00	1 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: A teljes dokumentum átnézése közösen. Döntés: A dokumentum beadásra készen áll.

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Ciszterna

Csövek vannak hozzá kapcsolva, amikben ha van víz, az a ciszternába folyik, megnövelve a ciszternába gyűjtött víz mennyiséget. Véletlen időközönként új cső jelenik meg rajta.

3.1.2 Cső

Vizet lehet behelyezni és kivenni belőle. Korlátos kapacitása van, aminél több víz nem helyezhető bele. Ha lyukas vagy szabad végű, akkor a benne lévő és bele helyezett víz a földre folyik, ekkor nem lehet belőle vizet pumpálni. Ha egy karakter akar rajta közlekedni, de már áll rajta valaki, vagy valaki kezében tartja valamely végét, akkor azt visszadobja. Amikor egy szerelő megpróbálja felvenni, ha újonnan generált akkor a szabad végét adja a szerelő kezébe, egyéb esetben, pedig a szerelő felőli végét (a szerelő alatti aktív elemhez csatlakoztatottat). Ha ráhelyeznek egy új pumpát, akkor kettévágódik; két új cső és egy pumpa keletkezik a helyén.

3.1.3 Forrás

Csövek vannak hozzá kapcsolva, amikben ha van hely, a forrásból víz folyik beléjük.

3.1.4 Föld

A csőhálózatot körülvevő terület. Ha egy lyukas vagy szabad végű csőbe víz kerül, akkor az onnan átfolyik a földre, ahol a földre gyűjtött víz mennyisége megnő.

3.1.5 Pumpa

Aktív elem, ami a bemeneti csövéből a víztartályába és víztartályából a kimeneti csőbe pumpál vizet, a többi csövét eközben zárva tarja. El tud romlani és meg lehet javítani. Valamint át lehet állítani a kimeneti és bemeneti csövét.

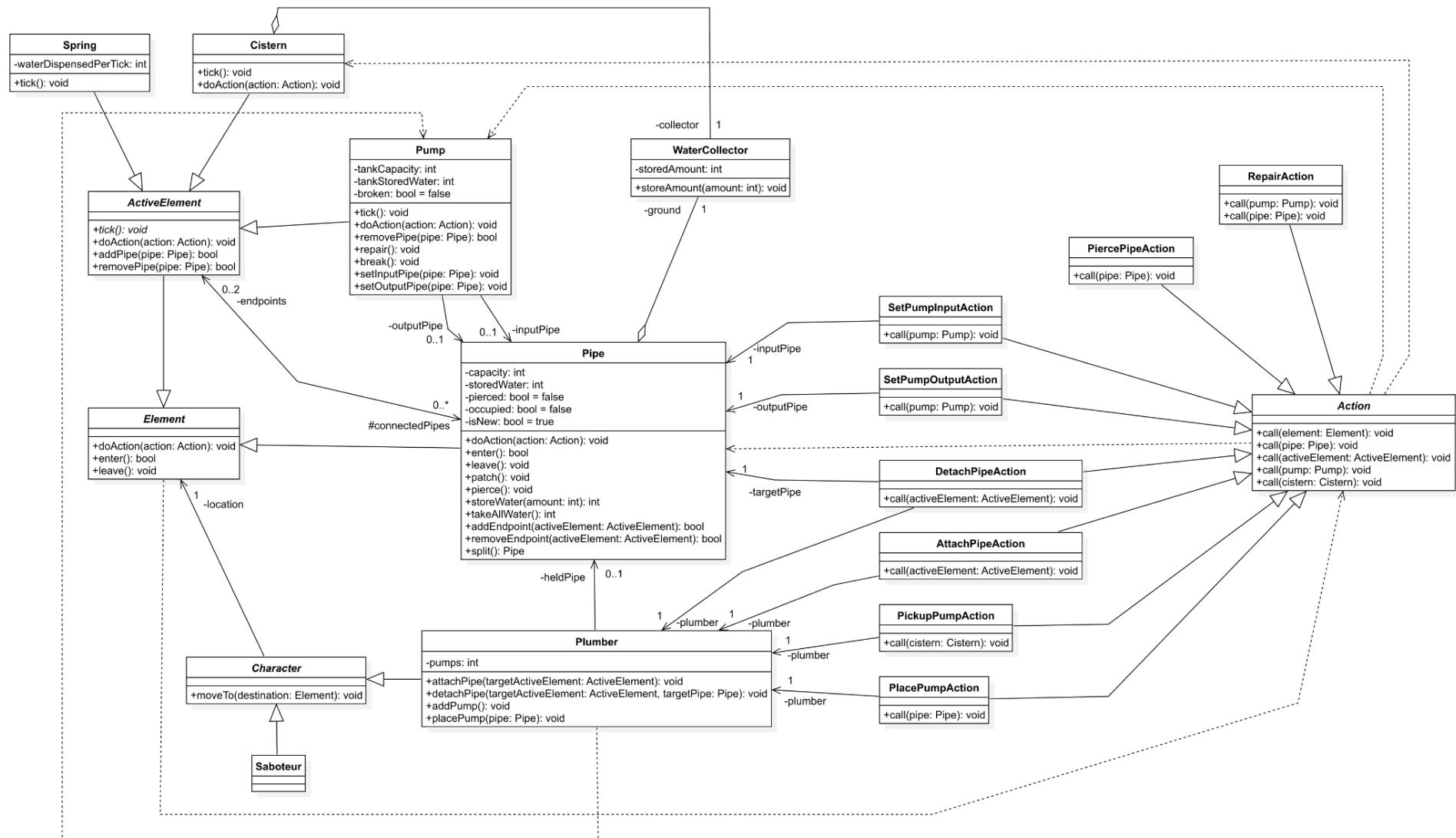
3.1.6 Szabotőr

Mozog a csőhálózaton, pumpákon kimeneti és bemeneti csöveket állít át és kilyukaszt csöveket.

3.1.7 Szerelő

Mozog a csőhálózaton. A pumpákon beállíthatja a kimeneti és bemeneti csövet. Meg tudja javítani a pumpát. Be tudja foltozni a lyukas csövet. Le tud csatlakoztatni cső végeket, azokat a kezébe venni, és elvinni őket más aktív elemekhez, majd ott csatlakoztatni. A ciszternánál képes új pumpát felvenni. Ha egy csövön áll és van nála pumpa akkor azt lehelyezheti, ezzel a csövet kettévágva.

3.2 Statikus struktúra diagramok



3.3 Osztályok leírása

3.3.1 Action

- **Felelősség**

Az *Action* a játékosok (szerelő, szabotőr, vagy bármely játékos) által elemeken végezhető műveletek elvégzéséért felelős absztrakt űrosztály. Az *Action* leszármazottak a megfelelő *call* függvények felülírásával definiálják, hogy milyen *Element* típus esetén mit kell csinálni az adott művelet véghezviteléért.

Alapértelmezetten a specifikusabb *Element* típusú *call* függvények az ōs típusú *call* függvényt hívják, ezzel elérve azt, ha például egy művelet bármilyen *ActiveElement*-en végrehajtható, akkor abban az *Action* leszármazottban elég a *call(ActiveElement)*-et definálni. Például egy *Pump* a *call(Pump)*-ot fogja meghívni, ami ez esetben felterjeszti a hívást a *call(ActiveElement)*-hez.

- **Ősosztályok**

Nincs ősosztálya.

- **Metódusok**

- **void call(Element element)**: Meghívásakor elvégzi a leszármazott osztály által meghatározott műveletet a kapott *Element* példányon, ha az elvégezhető *Element*-en.
- **void call(Pipe pipe)**: Alapértelmezetten a *call(Element)* függvényt hívja. A leszármazottak felüldefiniálhatják saját működésük szerint.
- **void call(ActiveElement activeElement)**: Alapértelmezetten a *call(Element)* függvényt hívja. A leszármazottak felüldefiniálhatják saját működésük szerint.
- **void call(Pump pump)**: Alapértelmezzen a *call(ActiveElement)* függvényt hívja. A leszármazottak felüldefiniálhatják saját működésük szerint.
- **void call(Cistern cistern)**: Alapértelmezzen a *call(ActiveElement)* függvényt hívja. A leszármazottak felüldefiniálhatják saját működésük szerint.

3.3.2 ActiveElement

- **Felelősség**

Aktív elem. Összeköti a csöveget, folyatja a hálózatban a vizet, csövet lehet felcsatlakoztatni rá és lecsatlakoztatni róla. Absztrakt osztály.

- **Ősosztályok**

Element → ActiveElement

- **Attribútumok**

- **Pipe[0..*] connectedPipes**: Hozzá csatlakoztatott csövek.

- **Metódusok**

- **void tick()**: Absztrakt metódus. Felüldefiniálásával a hozzácsatlakoztatott csövek közti vízfolyás menetét lehet megadni.
- **void doAction(Action action)**: A kapott *action* objektumon meghívja a *call* függvényt, paraméterként saját magát adva.
- **bool addPipe(Pipe pipe)**: Hozzáad egy csövet. Ha már hozzá van csatlakoztatva az adott cső, akkor nem képes végrehajtani a kívánt műveletet, és *false*-szal tér vissza

(egyéb esetben *true*-val). A kapott csőre az *addEndpoint(ActiveElement)* függvényt is meghívja, ezzel a csőhöz is hozzá lesz csatlakoztatva a jelen *ActiveElement* példány.

- **bool removePipe(Pipe pipe):** Megpróbálja eltávolítani magáról a paraméterként kapott csövet. Ha nincs hozzácsatlakoztatva a megadott cső, *false*-szal tér vissza. Sikeres esetben *true*-val tér vissza. A kapott csőre a *removeEndpoint(ActiveElement)* függvényt is meghívja, ezzel a csőről is le lesz csatlakoztatva a jelen *ActiveElement* példány.

3.3.3 AttachPipeAction

- **Felelősség**

Szerelők által végezhető művelet. A szerelő kezében lévő csővég csatlakoztatását valósítja meg azon az aktív elemen, amin végül elvégződik.

- **Ősosztályok**

Action → AttachPipeAction

- **Attribútumok**

- **Plumber plumber:** A szerelő, aki végzi ezt a műveletet.

- **Metódusok**

- **void call(ActiveElement activeElement):** Elvégzi a paraméterként kapott *ActiveElement* példányra a szerelő kezében lévő csővég felcsatlakoztatását, és a szerelőtől elveszi a csővéget. Ha a cső másik vége ugyanebbe az *ActiveElement* példányba csatlakozik, nem történik meg a felcsatlakoztatás és a szerelő kezében marad a csővég.

3.3.4 Cistern

- **Felelősség**

Vizet vesz el a szomszédos csövekből, és eltárolja azt.

- **Ősosztályok**

Element → ActiveElement → Cistern

- **Attribútumok**

- **WaterCollector collector:** A vízgyűjtő, amibe a ciszternák a vizet gyűjtik.

- **Metódusok**

- **void tick():** A hozzá csatlakoztatott csövekből átfolyatja a vizet és odaadja a *collector*-nak eltárolásra.
- **void doAction(Action action):** A kapott *action* objektumon meghívja a *call* függvényt, paraméterként saját magát adva.

3.3.5 Character

- **Felelősség**

Az elemeken mozog, számon tartja jelenlegi tartózkodási helyét. Állítgatja a pumpák kimeneti és bemeneti csövét. Absztrakt osztály.

- **Ősosztályok**

Nincs ősosztálya.

- **Attribútumok**

- **Element location:** A hely ahol a karakter áll.

- **Metódusok**

- **void moveTo(Element destination):** Átvándorol a kívánt mezőre.

3.3.6 DetachPipeAction

- **Felelősség**

Szerelők által végezhető művelet. Az attribútumként megadott cső egyik végének lecsatlakoztatását végzi arról az aktív elemről, aminek ez az *Action* példány oda lesz adva. A lecsatlakoztatott csővéget végül megkapja a szerelő.

- **Ősosztályok**

Action → DetachPipeAction

- **Attribútumok**

- **Plumber plumber:** A szerelő, aki végzi ezt a műveletet.
- **Pipe targetPipe:** A lecsatlakoztatni kívánt cső.

- **Metódusok**

- **void call(ActiveElement activeElement):** Elvégzi a paraméterként kapott *ActiveElement* példányon a *targetPipe* cső lecsatlakoztatását. Ha a csövön áll karakter, akkor nem történik meg a lecsatlakoztatás.

3.3.7 Element

- **Felelősség**

A csőrendszer eleme, a karakterek ezen képesek mozogni. *Action*-t lehet rajta végrehajtani.

- **Ősosztályok**

Nincs ősosztálya.

- **Metódusok**

- **void doAction(Action action):** A kapott *action* objektumon meghívja a *call* függvényt, paraméterként saját magát adva.
- **bool enter():** Az elem fogadja a rálépő karaktert, *true* értékkel tér vissza.
- **void leave():** Valaki lelép az elemről.

3.3.8 PickupPumpAction

- **Felelősség**

Szerelők által, ciszternán állva végezhető művelet. A szerelő a kezébe vesz egy pumpát.

- **Ősosztályok**

Action → PickupPumpAction

- **Attribútumok**

- **Plumber plumber:** A szerelő, aki végzi ezt a műveletet.

- **Metódusok**

- **void call(Cistern cistern):** A szerelő kezében lévő pumpák számának növelése eggyel.

3.3.9 PiercePipeAction

- **Felelősség**

Szabotörök által végezhető művelet. Annak a csőnek a kilyukasztása, aminek ez az *Action* példány oda lesz adva.

- **Ősosztályok**

Action → PiercePipeAction

- **Metódusok**

- **void call(Pipe pipe):** A kapott *Pipe* példány kilyukasztása.

3.3.10 Pipe

- **Felelősség**

Játékbeli csövet leíró osztály. Két végponttal rendelkezik, amik egy-egy különböző aktív elemhez vannak csatlakoztatva.

Egyes aktív elemek vizet helyezhetnek bele és szedhetnek ki belölle.

Ha kilyukasztják, a benne lévő víz a földre folyik, és minden további bele folyó víz is a földre kerül, egészen addig, amíg meg nem szerelik.

Egy szerelő képes felvenni az egyik végét, és átvinni más aktív elemhez, ahol azt felcsatlakoztathatja. Ha a cső egy új (a ciszternánál létrehozott) cső, akkor felvételkor a cső szabad vége kerül a felvező kezébe. Felvételkor a csőben lévő víz a földre folyik, és szabad végű csőbe folyó víz is a földre kerül.

A karakterek képesek közlekedni a csövön. Viszont egyszerre csak egy valaki állhat rajta. Ha valaki már áll rajta, akkor más nem képes fogadni. Ha valamelyik végpontja szabad (nem csatlakozik semmilyen aktív elemhez sem), ugyancsak nem engedi újabb karakterek rálépését. Ha már valaki áll rajta, nem szabad a végeit lecsatlakoztatni.

Elhelyezhető rajta új pumpa. Ekkor középen kettévágódik, és a két új csővég pumpára való felcsatlakoztatásáért a műveletet végző karakter felelős.

- **Ősosztályok**

Element → Pipe

- **Attribútumok**

- **bool pierced:** Ez az érték igaz, ha a cső lyukas.
- **bool occupied:** Ez az érték mondja meg, hogy áll-e már valaki a csövön.
- **bool isNew:** Ez az érték igaz, ha a cső egy új cső, (ami a ciszternánál keletkezett,) ellenkező esetben hamis. (Azaz ha már csatlakoztatva van a csőrendszer többi részéhez, vagy egy szerelő már felvette a kezébe).
- **int capacity:** A csőben tárolt víz kapacitása. (A maximálisan befogadható víz mértéke.)

- **int storedWater:** Adott pillanatban a csőben lévő víz mennyisége. Nem haladhatja meg a *capacity*-t.
- **WaterCollector ground:** A csatornahálózat alatt lévő “föld”. Ez tárolja a kifolyt víz mennyiségét.
- **ActiveElement[0..2] endpoints:** A cső végeihez csatlakoztatott aktív elemek halmaza.
- **Metódusok**
 - **void doAction(Action action):** A kapott *action* objektumon meghívja a *call* függvényt, paraméterként saját magát adva.
 - **bool enter():** Megpróbálja fogadni a rálépő karaktert. Ha a cső *endpoints* halmaza nem rendelkezik két értékkel, vagy az *occupied* értéke igaz, akkor nem lehetséges rálépní és *false* értékkel tér vissza. Ellenkező esetben átállítja az *occupied* értékét igazra, és *true*-val tér vissza.
 - **void leave():** Valaki elhagyja a csövet, a cső *occupied* értéke *false*-ra változik.
 - **void patch():** A cső befoltozása, azaz a *pierced* állapot *false*-lesz.
 - **void pierce():** A cső kilyukasztása, azaz a *pierced* állapot *true*-ra állítása. A *storedWater*ben lévő vízmennyiséget áthelyezi a *ground*-ba.
 - **int storeWater(amount int):** Eltárolja az adott mennyiségi vízet a *storedWater*ben, majd visszatér az el nem tárolt vízmennyiséggel. Ha lyukas a cső (a *pierced* értéke *true*), akkor a víz a *storedWater* helyett a *ground*ba kerül.
 - **int takeAllWater():** Kiveszi az összes vizet a csőből.
 - **bool addEndpoint(ActiveElement activeElement):** Csatlakoztatja hozzá a kapott *activeElementet*, ha az már nincs benne az *endpoints*ban. Majd hozzáadja az *endpoints*-hoz, és visszatér *true*-val tér vissza. Ha az *endpoints* már tartalmazza a kapott *activeElementet*, akkor az nem adódik hozzá az *endpoints*-hoz és a függvény visszatérése *false* lesz.
 - **bool removeEndpoint(ActiveElement activeElement):** Eltávolítja a kapott *activeElementet* az *endpoints* listából, ha az *occupied* és az *isNew* értéke *false* (akkor *true* értéket ad vissza). Ha az *isNew* értéke *true*, akkor *occupied* értékétől függetlenül *true* értékkel fog visszatérni, és nem távolítja el a kapott *activeElementet*, hanem az *isNew* értékét átállítja *false*-ra. Ha az *occupied* értéke *true* és *isNew* értéke *false*, nem távolítódik el az adott végpont, és a visszatérési érték *false* lesz.
 - **Pipe split():** A csövet kettévágja, vagyis a jelenlegi cső egyik endpoint-ját lecsatlakoztatja. A folyamat alatt új cső objektum jön létre, és ez összecsatlakoztatásra kerül az előzőleg lecsatlakoztatott endpoint-tal. Az újonnan létrehozott csövet visszatéríti.

3.3.11 PlacePumpAction

- **Felelősség**

Szerelők által, csövön állva végezhető művelet. A szerelő egy kezében lévő pumpát helyez a csőre, amin áll.

- **Ősosztályok**

Action → PlacePumpAction

- **Attribútumok**

- **Plumber plumber:** A szerelő, aki végzi ezt a műveletet.

- **Metódusok**

- **void call(Pipe pipe):** A szerelő kezében lévő pumpák közül egyet lehelyez a csőre, amin a szerelő áll. A cső ennek következtében kettévágódik, és a lehelyezett pumpa a két új csővéghez kötődik be. A szerelő átmozog a lehelyezett pumpára, és eggyel csökkenti a kezében lévő pumpák számát.

3.3.12 Plumber

- **Felelősség**

Képes pumpákat és csövet felvenni/lecsatlakoztatni, illetve lehelyezni/felcsatlakoztatni, pumpákat javítani, csöveket befoltozni.

- **Ősosztályok**

Character → Plumber

- **Attribútumok**

- **int pumps:** A szerelőnél lévő pumpák száma.
- **Pipe heldPipe:** A cső aminek a vége a kezében van. Ha nincs nála cső, akkor ez *null*.

- **Metódusok**

- **void attachPipe(ActiveElement targetActiveElement):** Ha a *heldPipe* értéke nem *null*, (azaz tart csövet a kezében,) megpróbálja felcsatlakoztatni a *heldPipe* csövet a kapott *activeElementre*. Ha ez sikeres, akkor a *heldPipe* értéke *null* lesz.
- **void detachPipe(ActiveElement targetActiveElement, targetPipe: Pipe):** Ha a *heldPipe* értéke *null* (azaz nem tart csövet a kezében), megpróbálja lecsatlakoztatni a paraméterként kapott csövet a kapott aktív elemről. Ha ez sikeres, a lecsatlakoztatott cső a *heldPipe*-ban tárolódik el.
- **void addPump():** A szerelő magához vesz egy pumpát.
- **void placePump(Pipe pipe):** Ha a *pumps* értéke nagyobb mint nulla, akkor csökkenti a *pumps* értékét eggyel és lehelyezi a pumpát az adott csőre.

3.3.13 Pump

- **Felelősség**

El tud romlani. Amikor nincs elromolva, akkor tickenként kimeneti és bemeneti csövei közt a víztartályán keresztül folyatja a vizet. Kimeneti és bemeneti csöve állítható.

- **Ősosztályok**

Element → ActiveElement → Pump

- **Attribútumok**

- **bool broken:** *True* ha a pumpa el van romolva, *false*, ha helyesen működik.
- **int tankCapacity:** A pumpa víztartályának víz tároló kapacitása. (A maximálisan befogadható víz mértéke.)
- **int tankStoredWater:** A pumpa víztartályában tárolt víz mennyisége. Nem haladhatja meg a *tankCapacity*-t.
- **Pipe inputPipe:** A pumpa bemeneti csöve.
- **Pipe outputPipe:** A pumpa kimeneti csöve.

- **Metódusok**

- **void tick()**: Bemeneti csövéből próbál vizet pumpálni víztárolóján keresztül a kimeneti csövébe ha a pumpa működik. Elsőként kimeneti csövébe próbál pumpálni, majd a bemeneti csőből a saját víztárolójába.
- **void doAction(Action action)**: A kapott *action* objektumon meghívja a *call* függvényt, paraméterként saját magát adva.
- **bool removePipe(Pipe pipe)**: Megpróbál lecsatlakoztatni magáról egy csövet. Ha ez sikerül akkor *true*-val tér vissza, más esetben *false*-szal. Ha a lecsatlakoztatott cső az *inputPipe* vagy az *outputPipe* volt, akkor az adott érték is nullázódik.
- **void repair()**: Átállítja a *broken* változó értékét *false*-ra.
- **void break()**: Átállítja a *broken* változó értékét *true*-ra.
- **void setInputPipe(Pipe pipe)**: Átállítja az *inputPipe* értékét a kapott csőre.
- **void setOutputPipe(Pipe pipe)**: Átállítja az *outputPipe* értékét a kapott csőre.

3.3.14 RepairAction

- **Felelősség**

Szerelők által végezhető művelet. Annak a csőnek vagy pumpának a megjavítása, aminek ez az *Action* példány oda lesz adva.

- **Ősosztályok**

Action → RepairAction

- **Metódusok**

- **void call(Pump pump)**: A kapott *Pump* példány megjavítása.
- **void call(Pipe pipe)**: A kapott *Pipe* példány befoltozása.

3.3.15 Saboteur

- **Felelősség**

A csőhálzon mozognak és csöveket lyukasztgatnak. A földre gyűjtik a vizet.

- **Ősosztályok**

Character → Saboteur

3.3.16 SetPumpInputAction

- **Felelősség**

Karakterek által végezhető művelet. Annak a pumpának a bemeneti csövének átállítása az attribútumként megadott csőre, aminek ez az *Action* példány oda lesz adva.

- **Ősosztályok**

Action → SetPumpInputAction

- **Attribútumok**

- **Pipe inputPipe**: A cső, amit a pumpa bemeneti csövének szeretnénk beállítani.

- **Metódusok**

- **void call(Pump pump):** A kapott *Pump* példány bemeneti csövének *inputPipe* csőre állítása.

3.3.17 SetPumpOutputAction

- **Felelősség**

Karakterek által végezhető művelet. Annak a pumpának a kimeneti csövének átállítása az attribútumként megadott csőre, aminek ez az *Action* példány oda lesz adva.

- **Ősosztályok**

Action → SetPumpOutputAction

- **Attribútumok**

- **Pipe outputPipe:** A cső, amit a pumpa kimeneti csövének szeretnénk beállítani.

- **Metódusok**

- **void call(Pump pump):** A kapott *Pump* példány kimeneti csövének *outputPipe* csőre átállítása.

3.3.18 Spring

- **Felelősség**

Tickenként vizet folyat a rácsatlakoztatott csövekbe.

- **Ősosztályok**

Element → ActiveElement → Spring

- **Attribútumok**

- **int waterDispensedPerTick:** A tickenként kibocsátott víz mennyisége.

- **Metódusok**

- **void tick():** A *connectedPipes*-okba a *waterDispensedPerTick*-nek megfelelő mennyiségű vizet tölt (a cső által visszadobott vízzel nem foglalkozik).

3.3.19 WaterCollector

- **Felelősség**

Korlátlan kapacitással vizet tárol. Számon tartja a belefolyt víz mennyiségét.

- **Ősosztályok**

Nincs ősosztálya.

- **Attribútumok**

- **storedAmount:** Az összegyűjtött víz mennyisége.

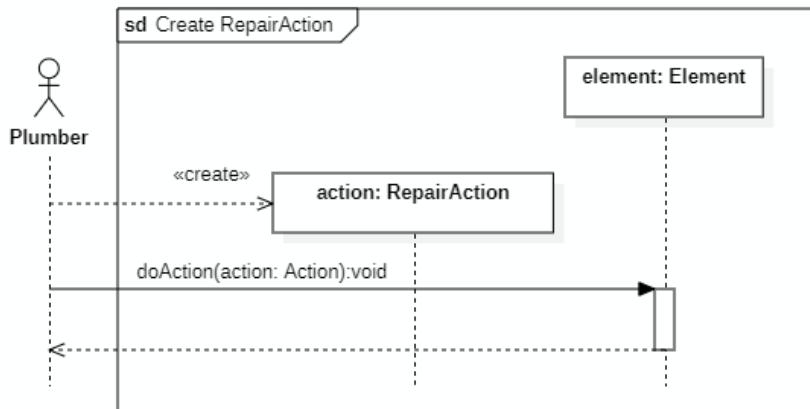
- **Metódusok**

- **void storeAmount(int amount):** Vizet fogad be.

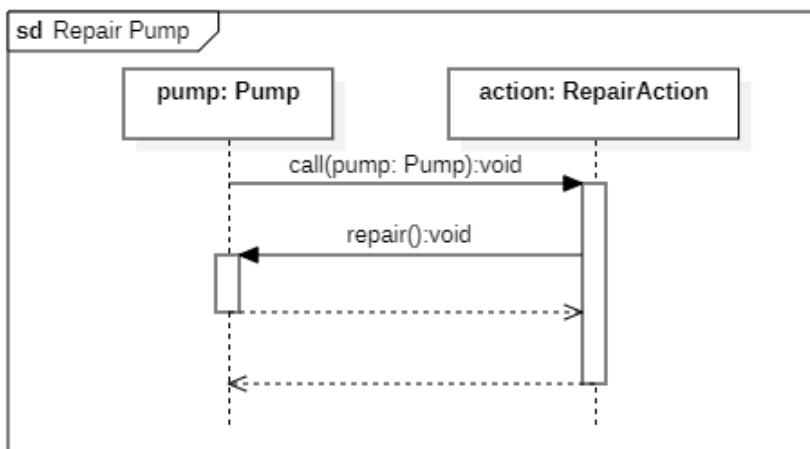
3.4 Szekvencia diagramok

Megjegyzés: Az egyes *Elementek doAction(Action)* függvényei lentebb lévő diagrammokon vannak külön feltüntetve.

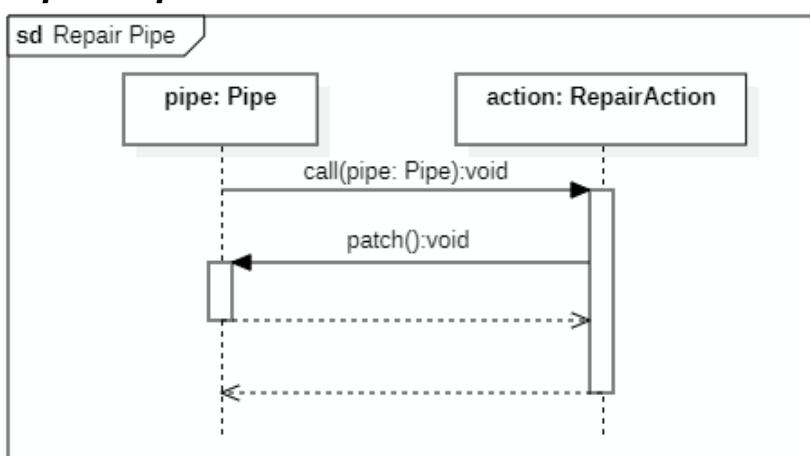
3.4.1 Create RepairAction



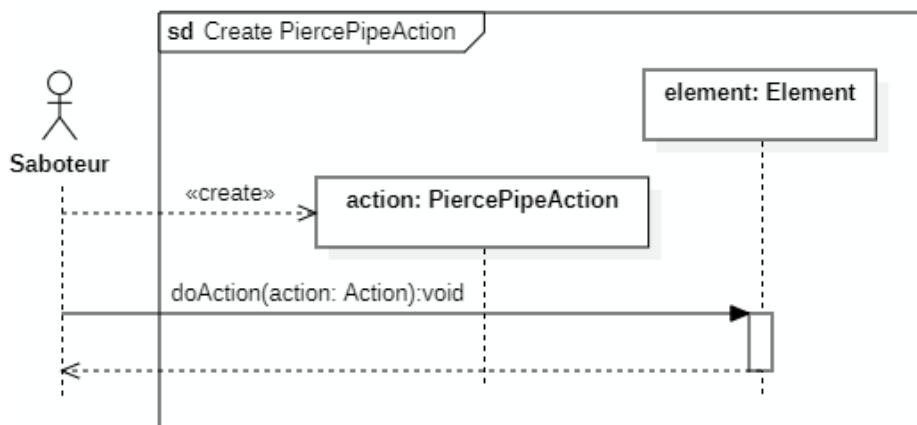
3.4.2 Repair Pump



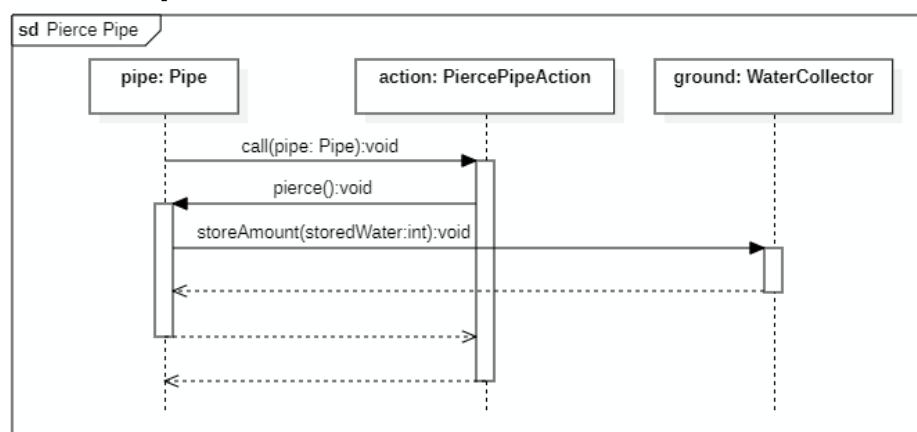
3.4.3 Repair Pipe



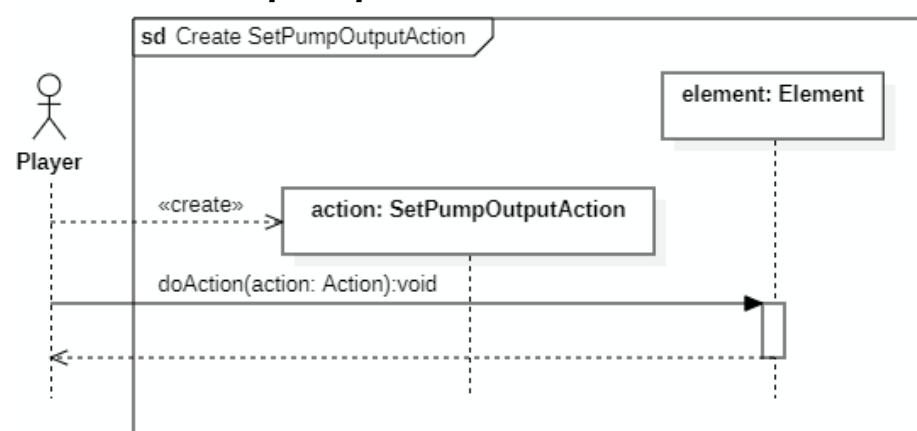
3.4.4 Create PiercePipeAction



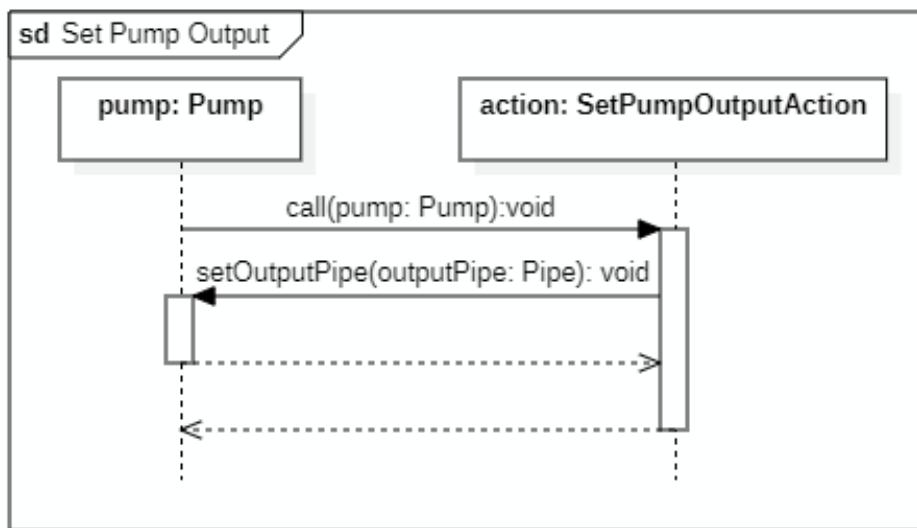
3.4.5 Pierce Pipe



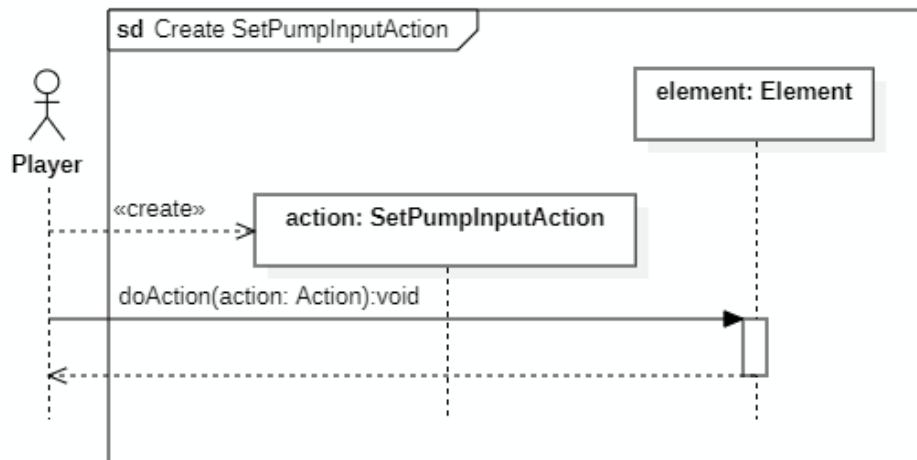
3.4.6 Create SetPumpOutputAction



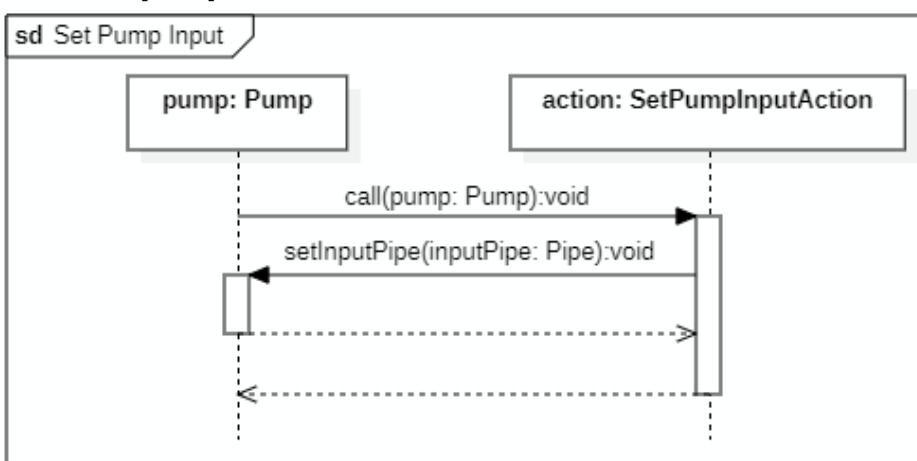
3.4.7 Set Pump Output



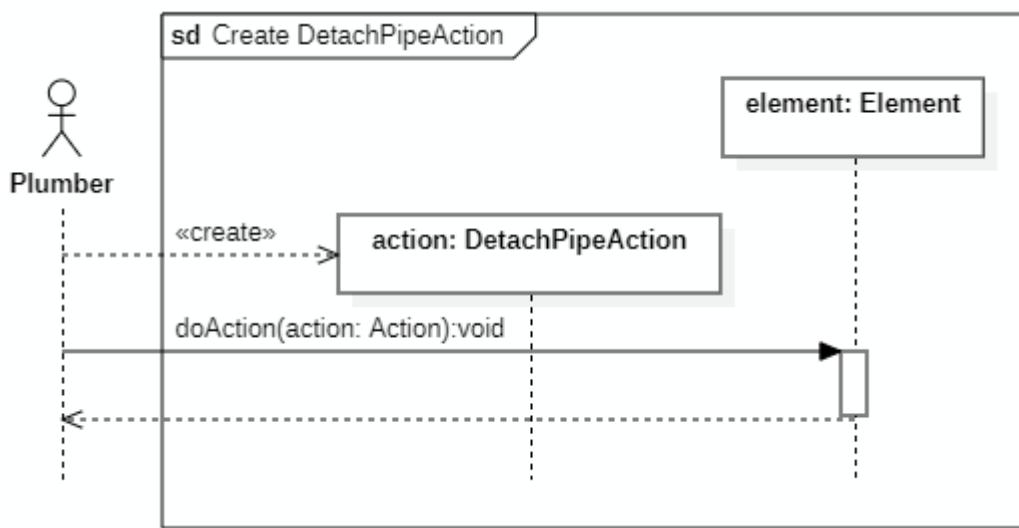
3.4.8 Create SetPumpInputAction



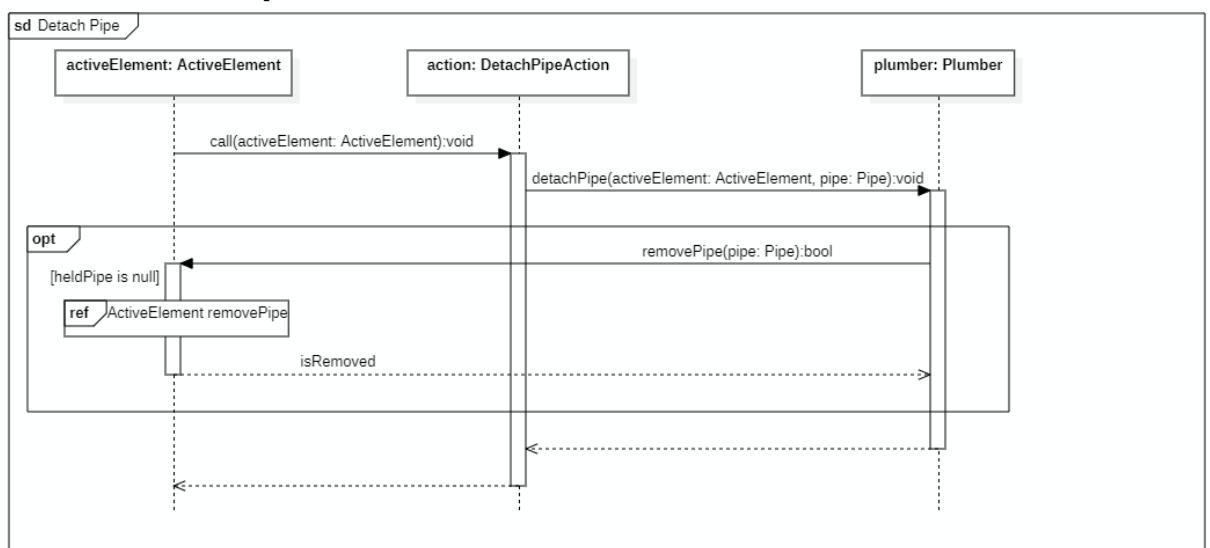
3.4.9 Set Pump Input



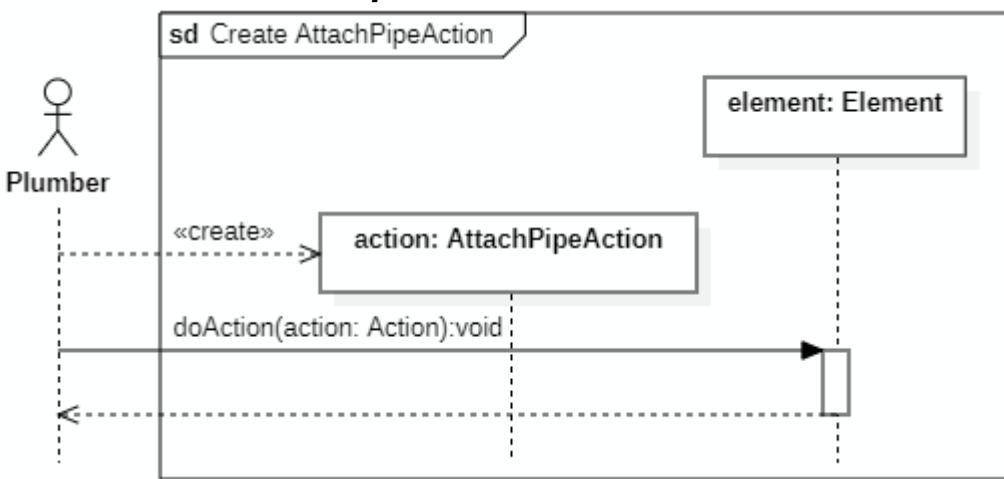
3.4.10 Create DetachPipeAction



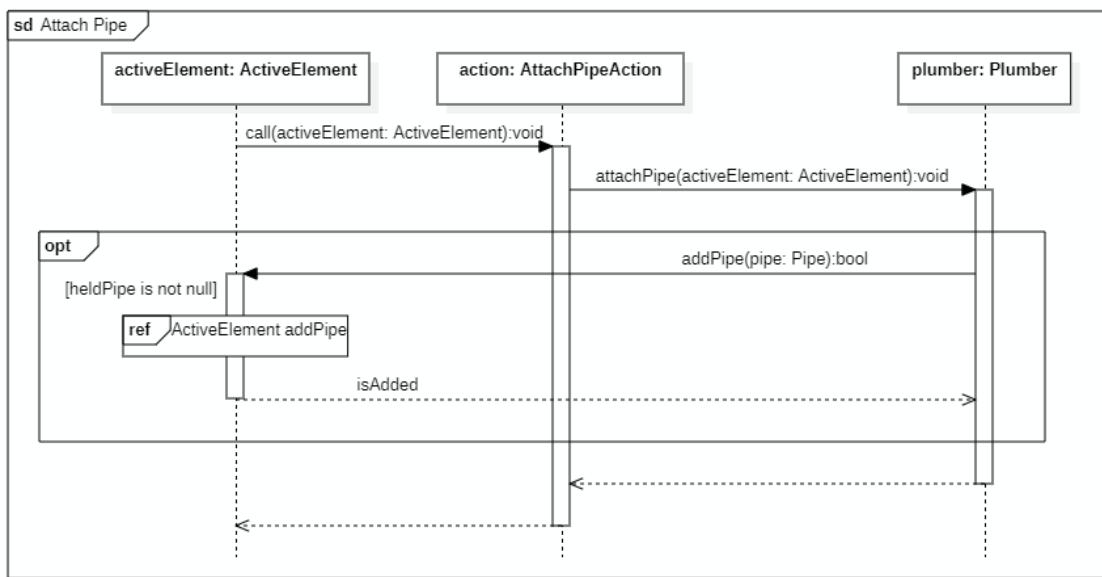
3.4.11 Detach Pipe



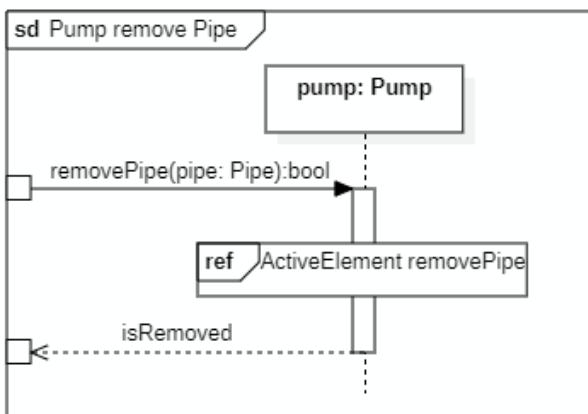
3.4.12 Create AttachPipeAction



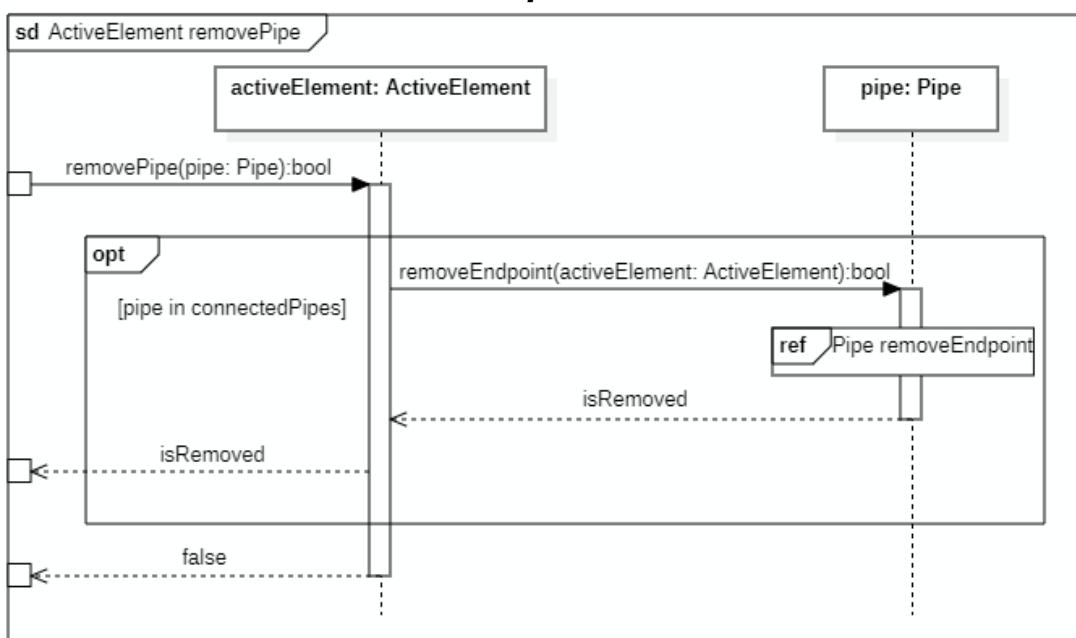
3.4.13 Attach Pipe



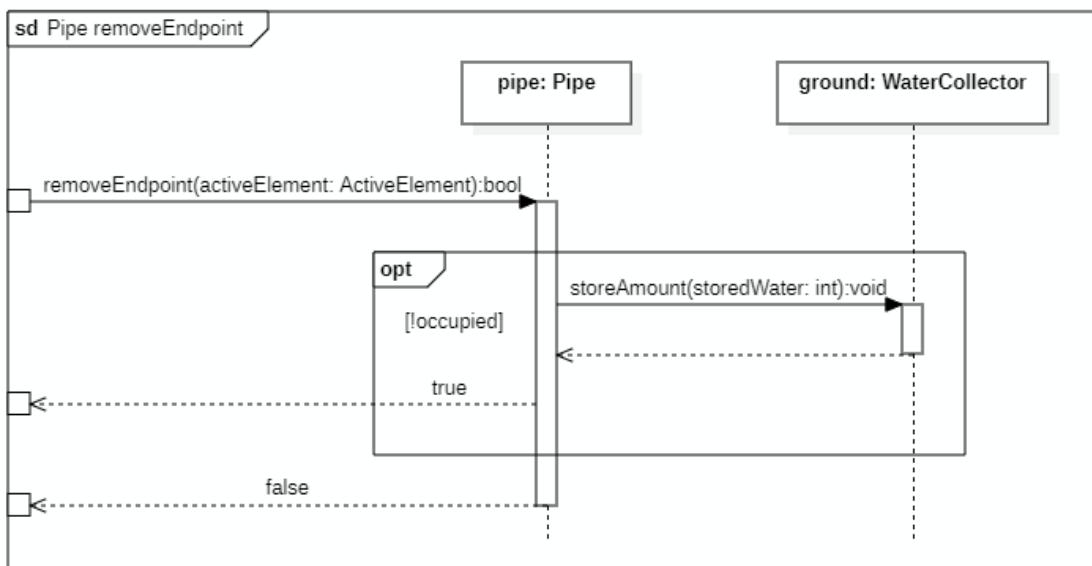
3.4.14 Pump remove Pipe



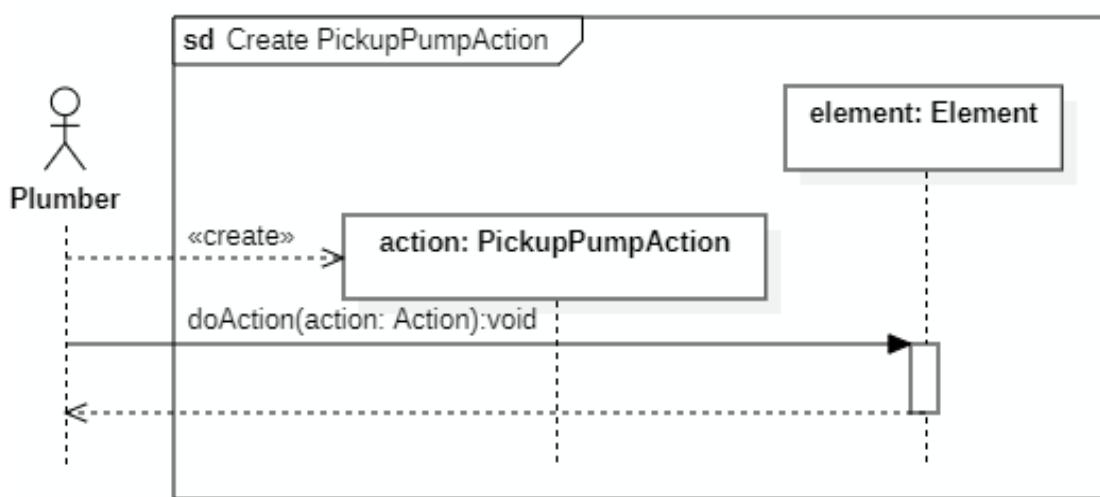
3.4.15 ActiveElement removePipe



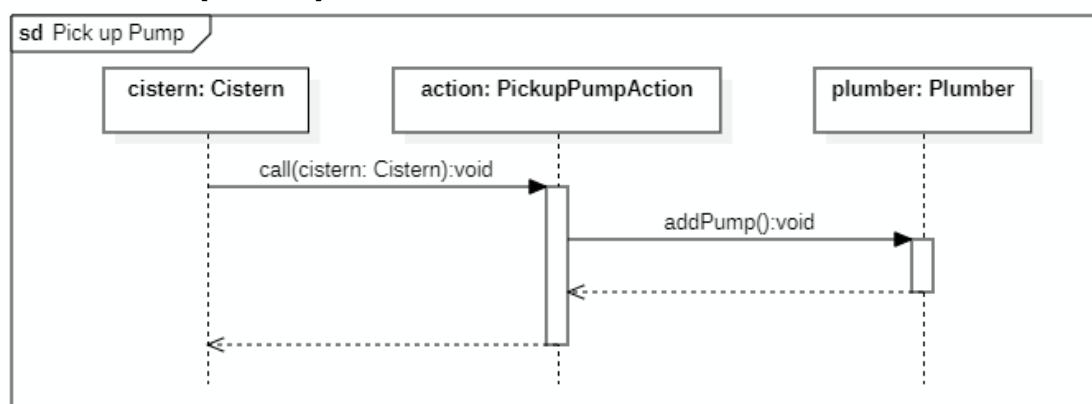
3.4.16 Pipe removeEndpoint



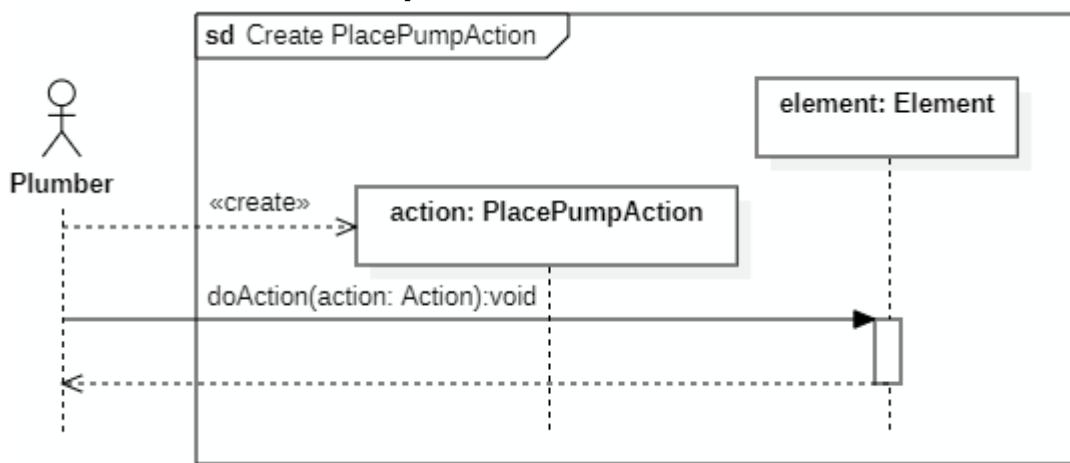
3.4.17 Create PickupPumpAction



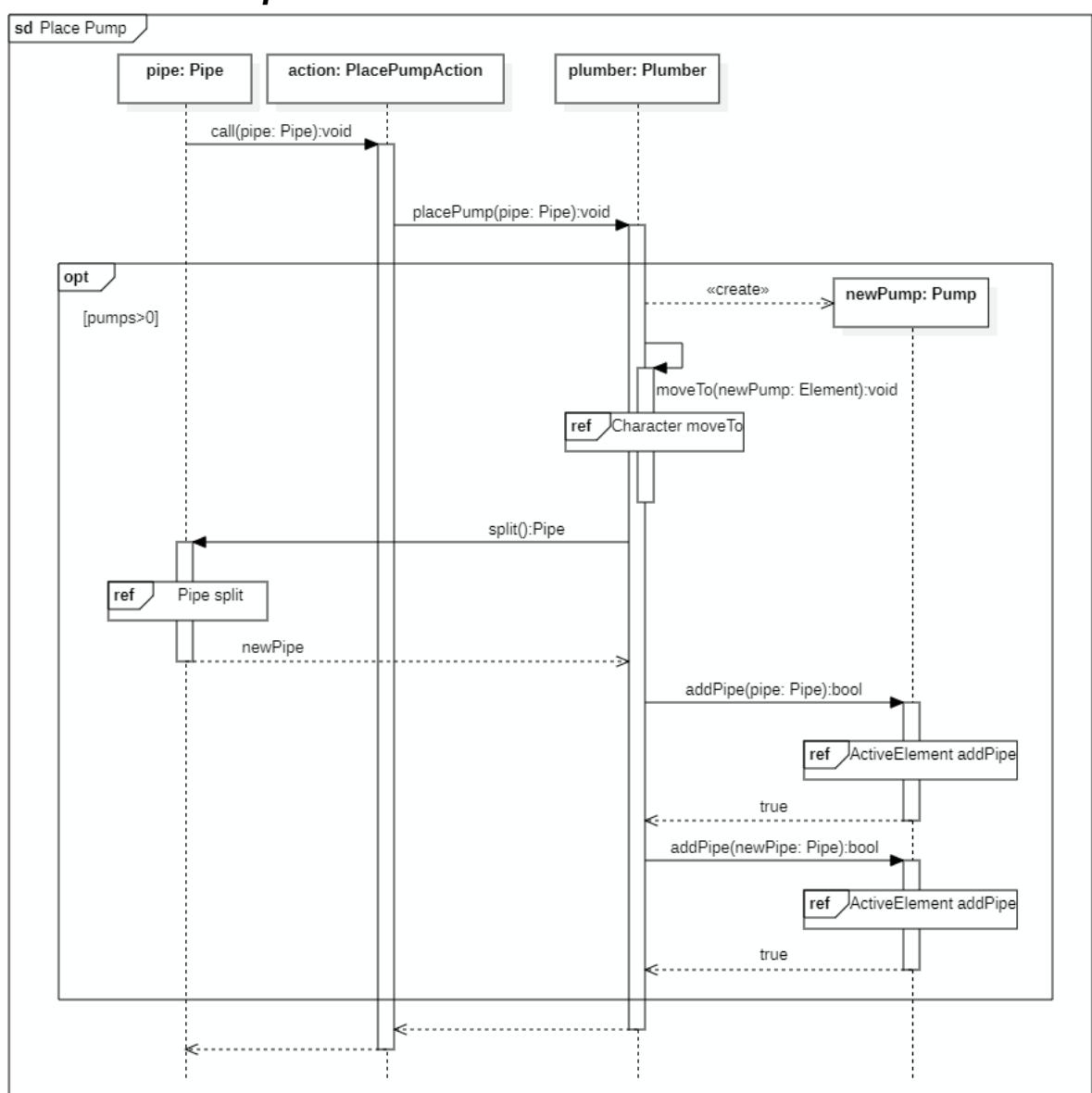
3.4.18 Pick up Pump



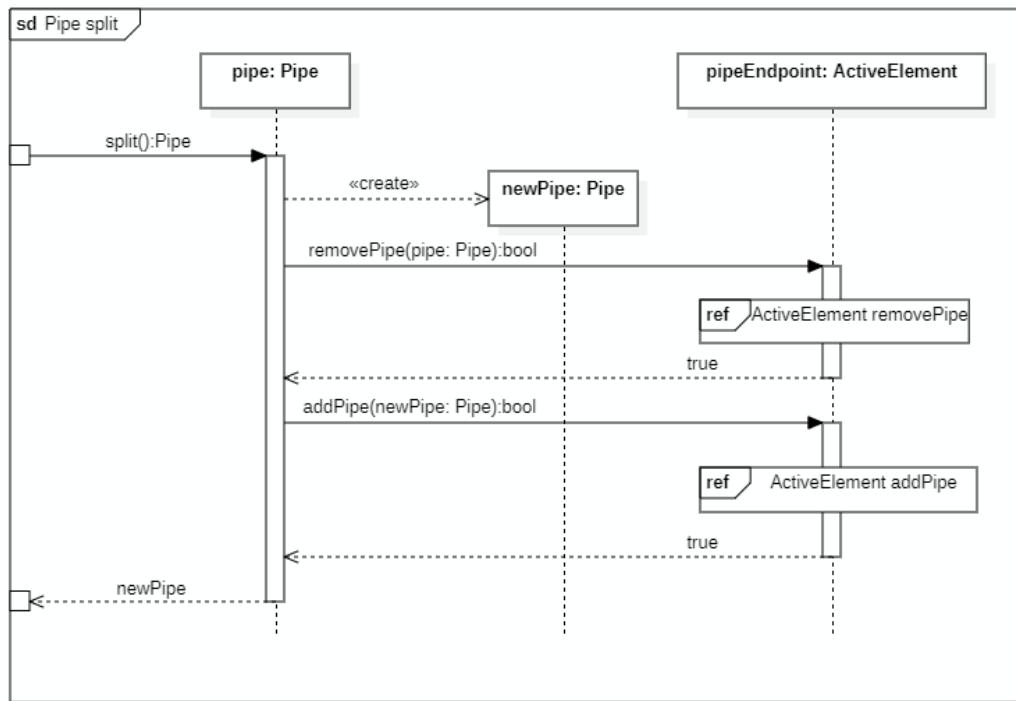
3.4.19 Create PlacePumpAction



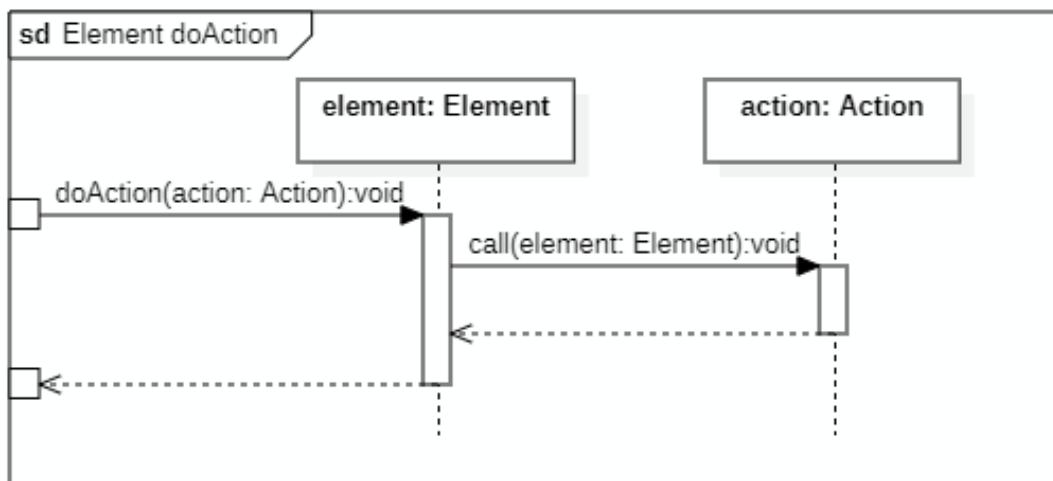
3.4.20 Place Pump



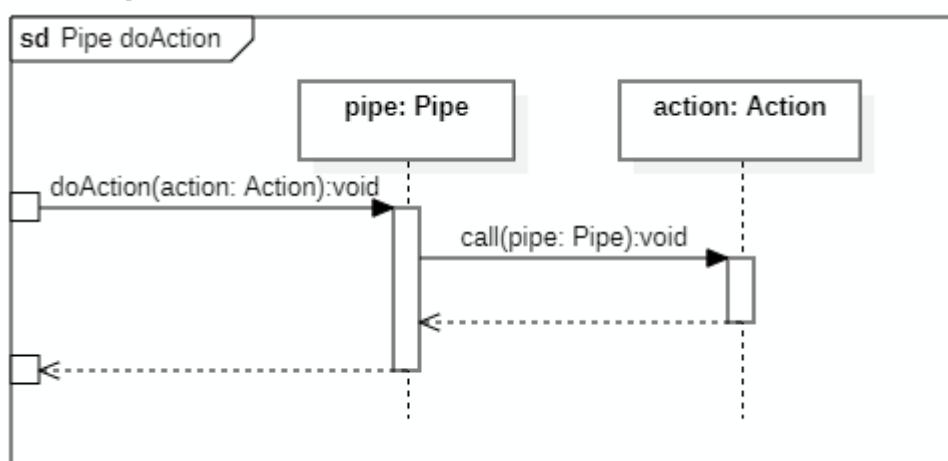
3.4.21 Pipe split

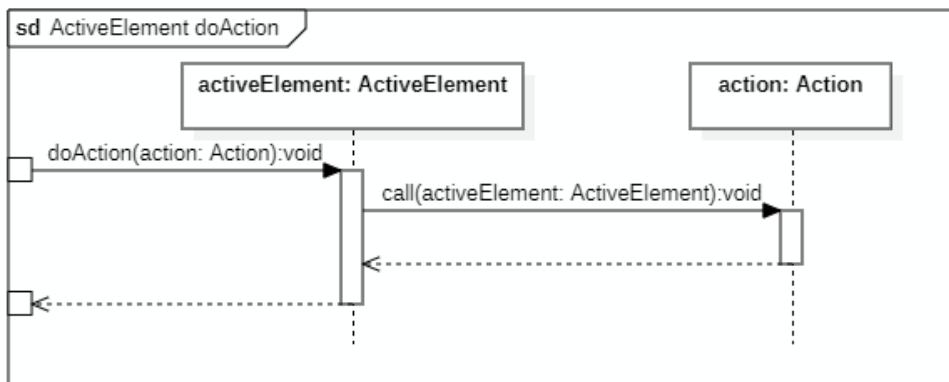
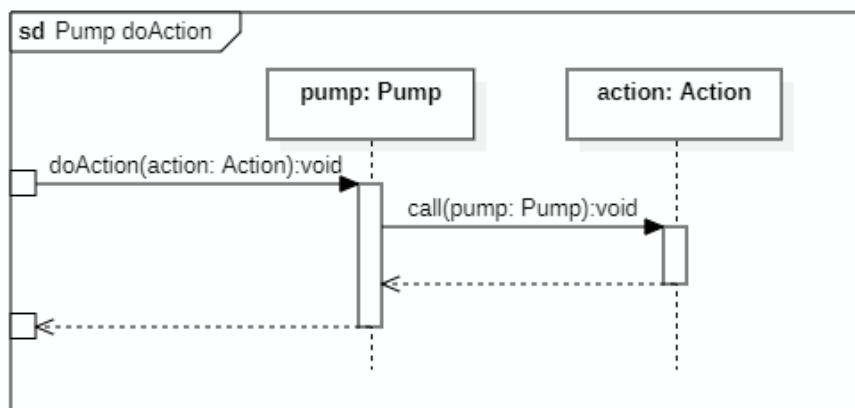
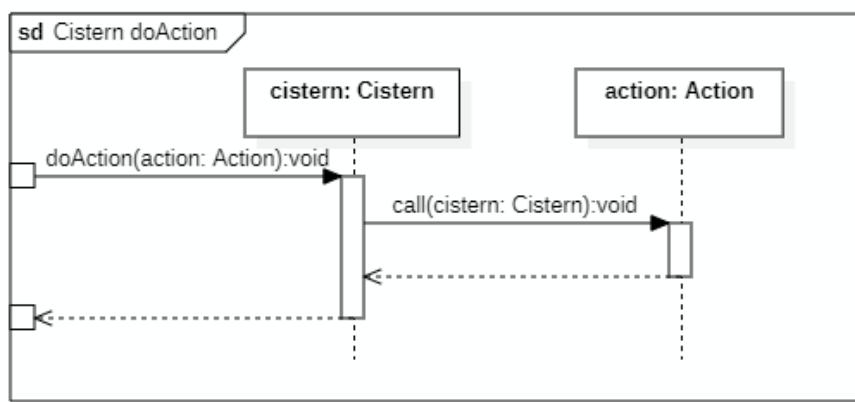


3.4.22 Element doAction

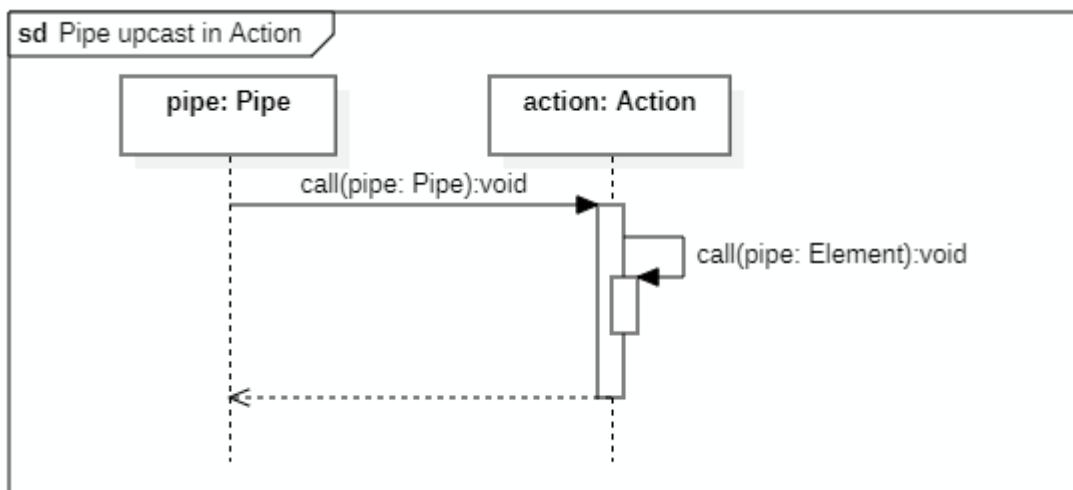


3.4.23 Pipe doAction

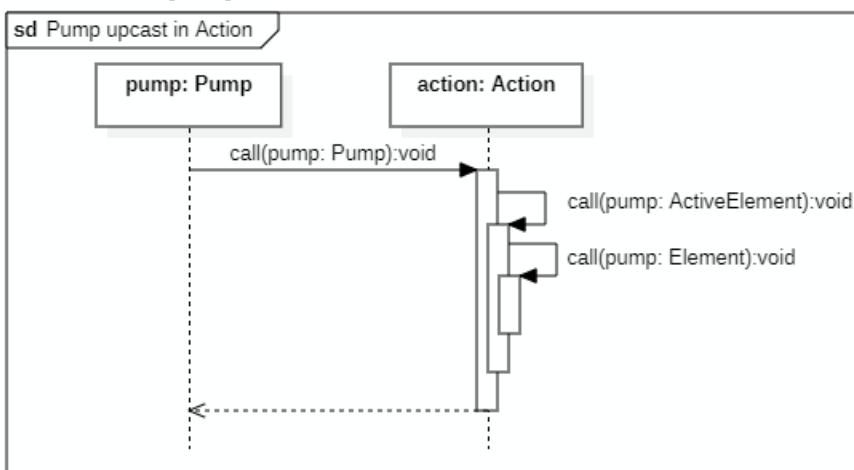


3.4.24 ActiveElement doAction**3.4.25 Pump doAction****3.4.26 Cistern doAction**

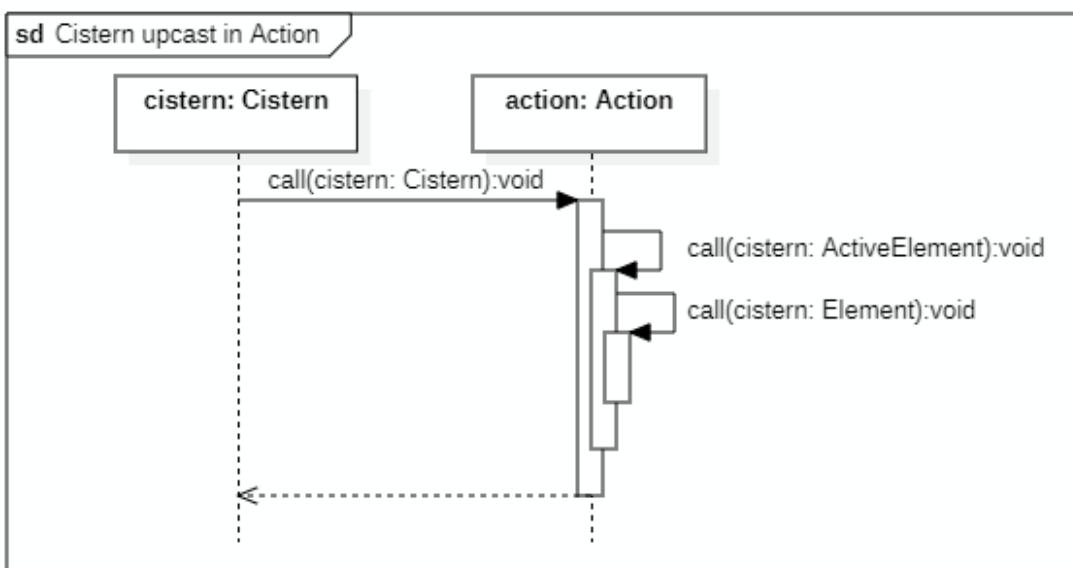
3.4.27 Pipe upcast in Action



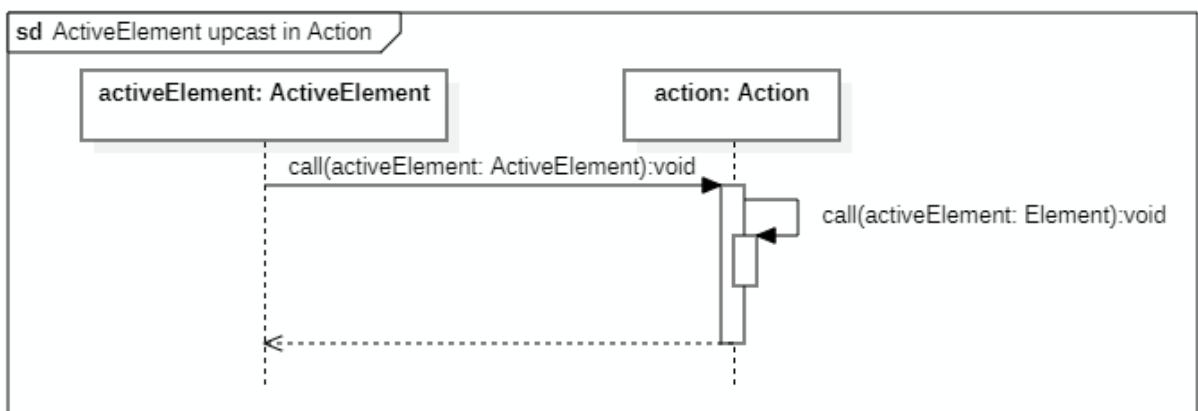
3.4.28 Pump upcast in Action



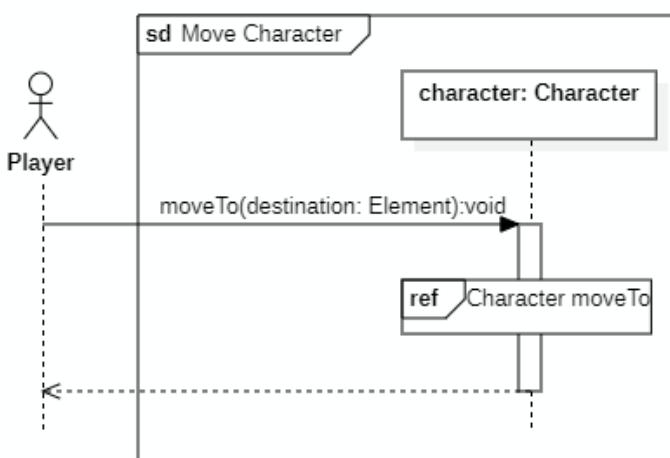
3.4.29 Cistern upcast in Action



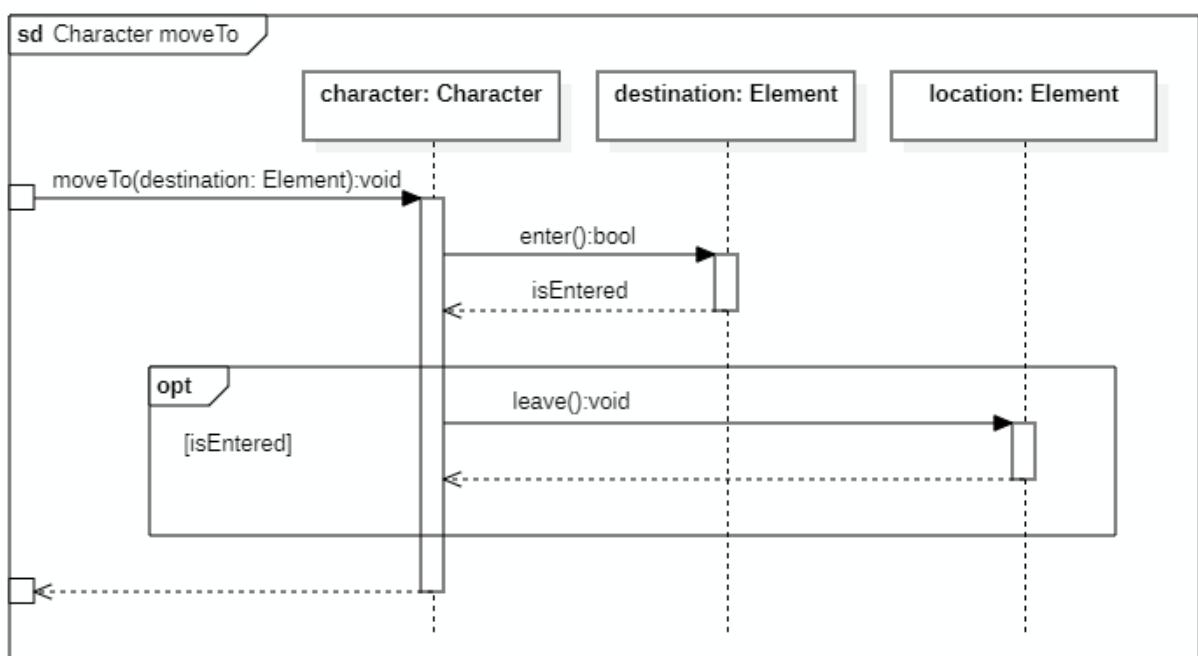
3.4.30 ActiveElement upcast in Action



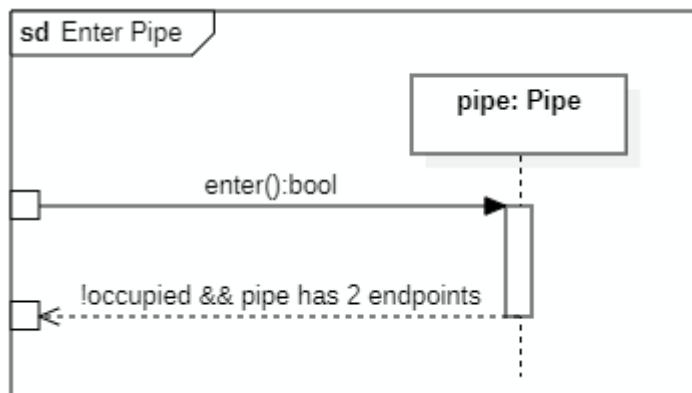
3.4.31 Move Character



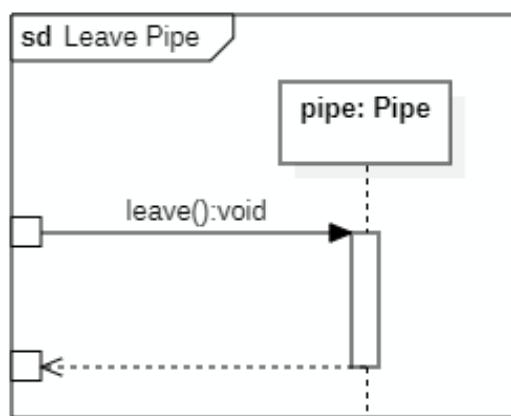
3.4.32 Character moveTo



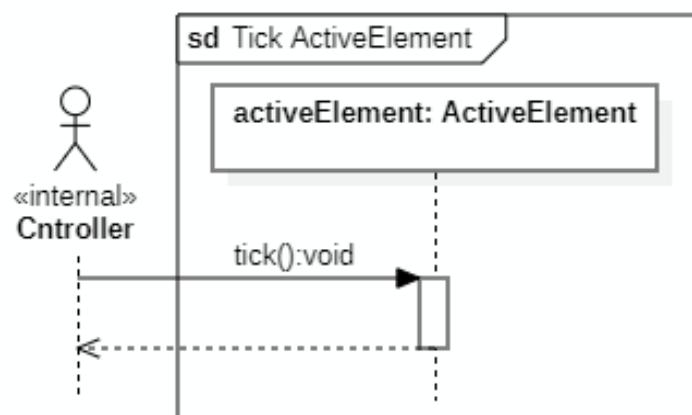
3.4.33 *Enter Pipe*



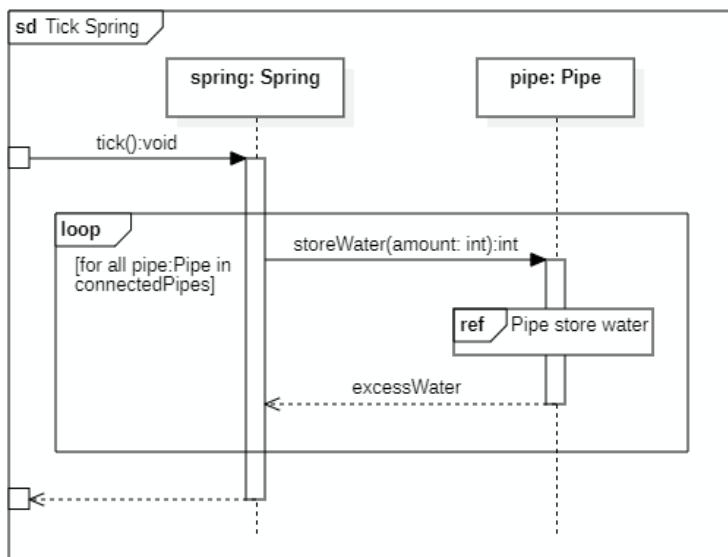
3.4.34 *Leave Pipe*



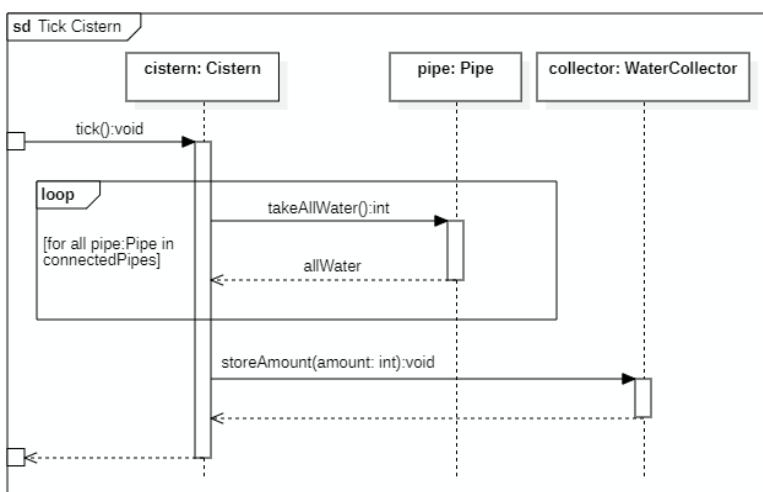
3.4.35 *Tick ActiveElement*



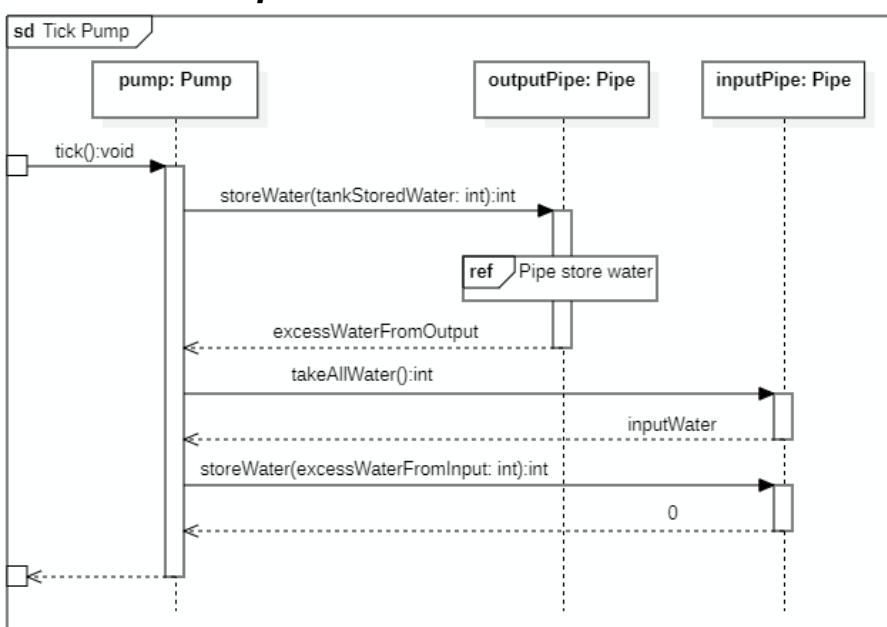
3.4.36 Tick Spring



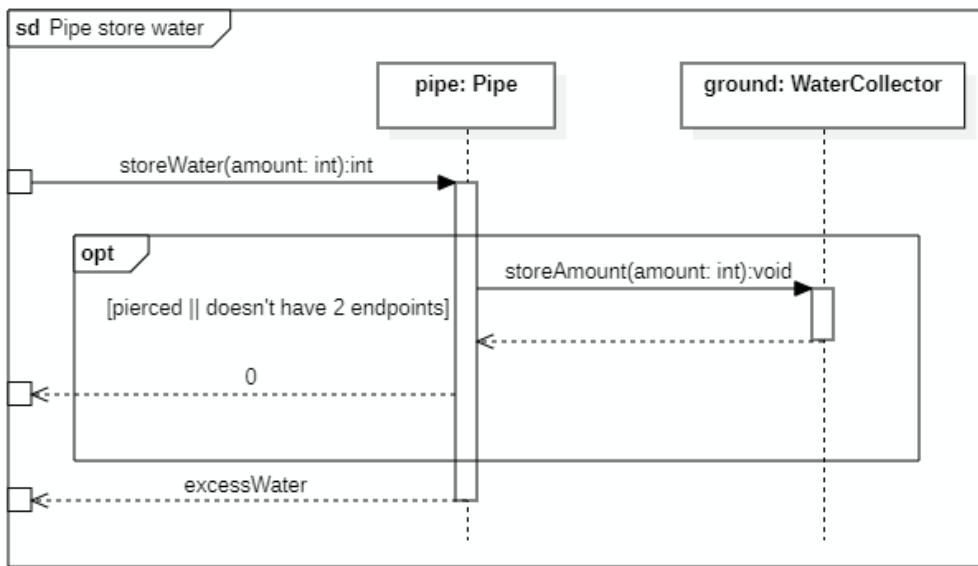
3.4.37 Tick Cistern



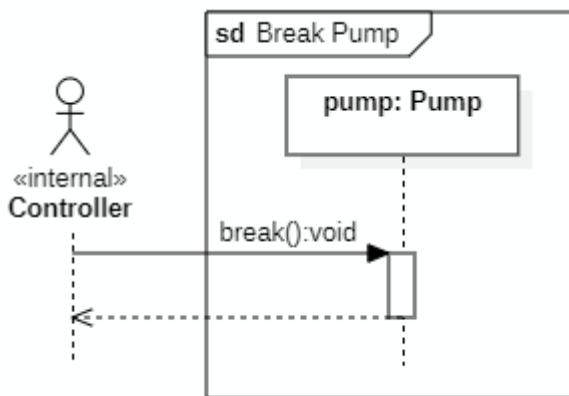
3.4.38 Tick Pump



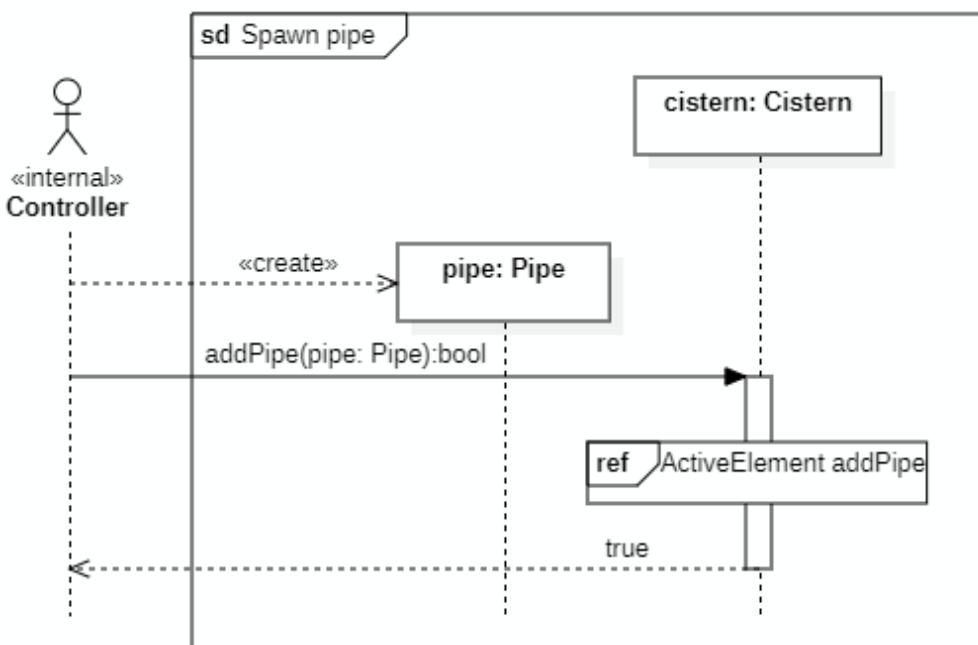
3.4.39 Pipe store water



3.4.40 Break Pump

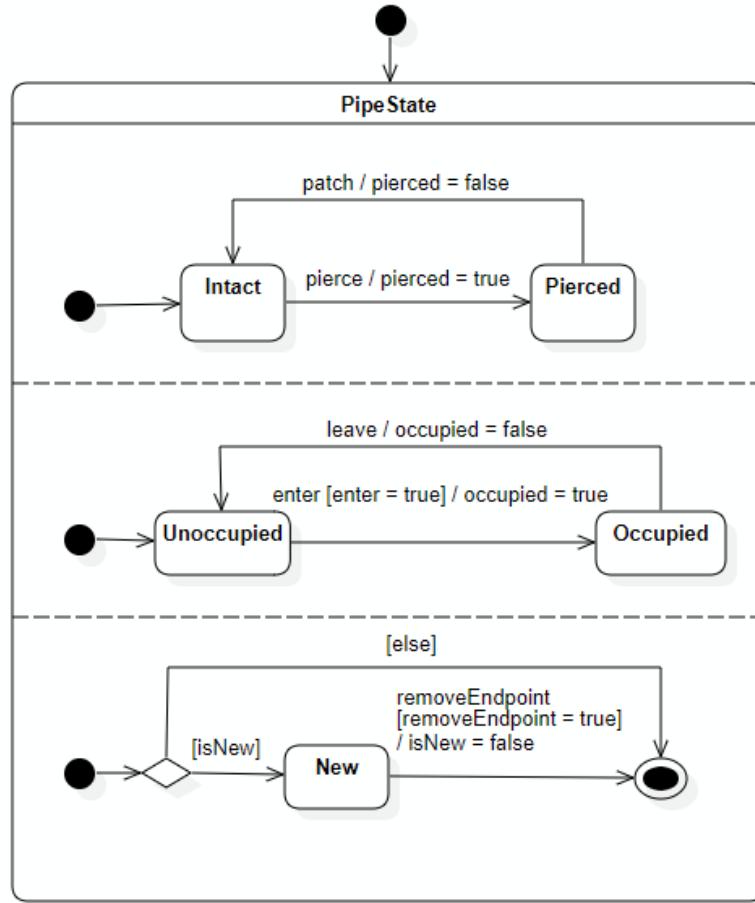


3.4.41 Spawn pipe



3.5 State-chartok

3.5.1. A Pipe osztály diagramja



3.5.2 A Pump osztály diagramja



3.6 Napló

Kezdet	Időtartam	Résznevők	Leírás
2023.03.14. 20:00	2 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet. Döntés: Harsányi a 3.1-es Galgóczy, Kereszthury, Tóth a 3.2-es feladaton dolgozik.
2023.03.14 22:00	4 óra	Kereszthury	Tevékenység: 3.2 Osztálydiagram készítése
2023.03.15. 14:00	2 óra	Galgóczy	Tevékenység: 3.2 Osztálydiagram módosítása
2023.03.15. 15:00	2 óra	Tóth	Tevékenység: 3.2 Osztálydiagram bővítése
2023.03.15. 17:00	1 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet. 3.2 Osztálydiagram átnézése Döntés: Az osztálydiagram még problémás részeit Galgóczy és Tóth javítja.
2023.03.16. 13:00	3 óra	Tóth	Tevékenység: 3.2 Osztálydiagram javítása
2023.03.16. 16:00	3 óra	Galgóczy	Tevékenység: 3.2 Osztálydiagram pontosítása
2023.03.16. 20:00	2 óra	Tóth	Tevékenység: 3.2 Osztálydiagram asszociációinak elkészítése, kinézet javítása.
2023.03.17. 18:00	2 óra	Harsányi	Tevékenység: 3.1 Objektum katalógus elkészítése
2023.03.17. 20:00	2,5 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet. Döntés: Harsányi a 3.3-as Kereszthury a 3.5-ös Szilczer a 3.4-es feladatot készíti el.
2023.03.18. 19:00	2 óra	Szilczer	Tevékenység: 3.4 Szekvencia diagramok készítése
2023.03.19. 12:00	2 óra	Szilczer	Tevékenység: 3.4 Szekvencia diagramok készítése
2023.03.19. 17:00	1 óra	Kereszthury	Tevékenység: 3.5 state-chartok elkészítése
2023.03.19. 17:00	3 óra	Szilczer	Tevékenység: 3.4 Szekvencia diagramok készítése és a meglévők hibáinak javítása
2023.03.19. 20:00	7 óra	Harsányi	Tevékenység: A már kitalált osztályok dokumentálása a 3.3 (Osztályok leírása) fejezetbe
2023.03.19. 23:00	3 óra	Tóth	Tevékenység: 3.3 Osztályok leírása fejezetben az Action-ök leírásának elkészítése.

2023.03.20. 00:00	4 óra	Szilczer	Tevékenység: 3.4 Szekvencia diagramok készítése és a meglévők hibáinak javítása
2023.03.20. 2:00	2 óra	Tóth	Tevékenység: 3.4 Szekvencia diagramok átnézése, összevetése osztálydiagrammal.
2023.03.20. 3:00	1 óra	Galgóczy	Tevékenység: 3.3 Osztályok leírásának ellenőrzése
2023.03.20. 10:00	2 óra	Szilczer	Tevékenység: 3.4 Szekvencia diagramok hibáinak javítása
2023.03.20. 11:00	1 óra	Galgóczy	Tevékenység: 3.3 Osztályok leírásának ellenőrzése
2023.03.20. 11:00	1 óra	Kereszthury	Tevékenység: 3.4 Szekvencia diagramok dokumentumba helyezése
2023.03.20. 12:00	1 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: A teljes dokumentum átnézése. Döntés: A dokumentum beadásra készen áll.

4. Analízis modell kidolgozása II.

4.1 Objektum katalógus

4.1.1 Ciszterna

Csövek vannak hozzá kapcsolva, amikben ha van víz, az a ciszternába folyik, megnövelve a ciszternába gyűjtött víz mennyiséget. Véletlen időközönként új cső jelenik meg rajta.

4.1.2 Cső

Vizet lehet behelyezni és kivenni belőle. Korlátos kapacitása van, aminél több víz nem helyezhető bele. Ha lyukas vagy szabad végű, akkor a benne lévő és bele helyezett víz a földre folyik, ekkor nem lehet belőle vizet pumpálni. Ha egy karakter akar rajta közlekedni, de már áll rajta valaki, vagy valaki kezében tartja valamely végét, akkor azt visszadobja. Amikor egy szerelő megpróbálja felvenni, ha újonnan generált akkor a szabad végét adja a szerelő kezébe, egyéb esetben, pedig a szerelő felőli végét (a szerelő alatti aktív elemhez csatlakoztatottat). Ha ráhelyeznek egy új pumpát, akkor kettévágódik; két új cső és egy pumpa keletkezik a helyén.

4.1.3 Forrás

Csövek vannak hozzá kapcsolva, amikben ha van hely, a forrásból víz folyik beléjük.

4.1.4 Föld

A csőhálózatot körülvevő terület. Ha egy lyukas vagy szabad végű csőbe víz kerül, akkor az onnan átfolyik a földre, ahol a földre gyűjtött víz mennyisége megnő.

4.1.5 Pumpa

Aktív elem, ami a bemeneti csövéből a víztartályába és víztartályából a kimeneti csőbe pumpál vizet, a többi csövét eközben zárva tarja. El tud romlani és meg lehet javítani. Valamint át lehet állítani a kimeneti és bemeneti csövét.

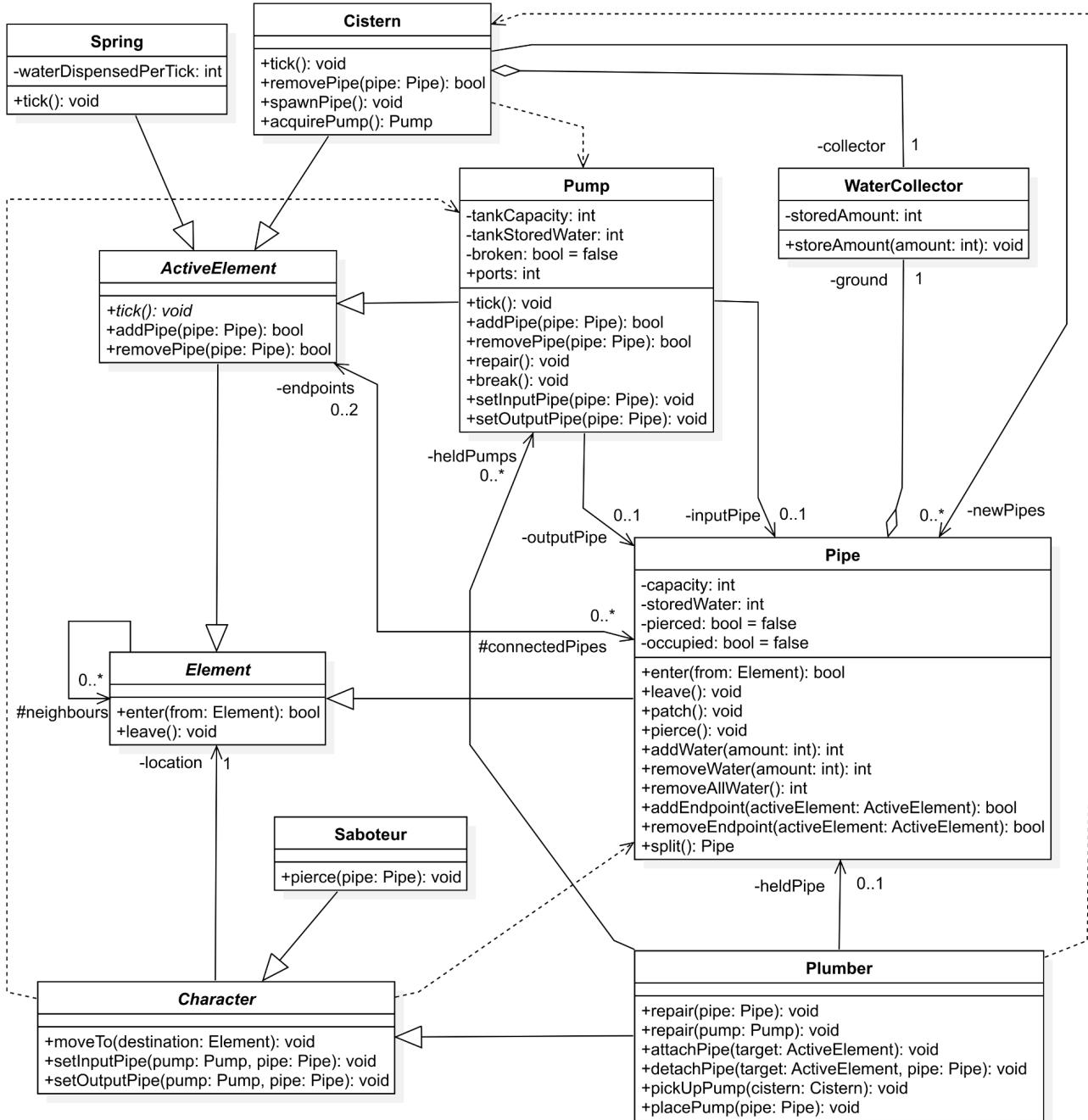
4.1.6 Szabotőr

Mozog a csőhálózaton, pumpákon kimeneti és bemeneti csöveket állít át és kilyukaszt csöveket.

4.1.7 Szerelő

Mozog a csőhálózaton. A pumpákon beállíthatja a kimeneti és bemeneti csövet. Meg tudja javítani a pumpát. Be tudja foltozni a lyukas csövet. Le tud csatlakoztatni cső végeket, azokat a kezébe venni, és elvinni őket más aktív elemekhez, majd ott csatlakoztatni. A ciszternánál képes új pumpát felvenni. Ha egy csövön áll és van nála pumpa akkor azt lehelyezheti, ezzel a csövet kettévágva.

4.2 Statikus struktúra diagramok



4.3 Osztályok leírása

4.3.1 ActiveElement

- **Felelősség**

Aktív elem. Összeköti a csöveket, folyatja a hálózatban a vizet, csövet lehet felcsatlakoztatni rá és lecsatlakoztatni róla. Absztrakt osztály.

- **Ősosztályok**

Element → ActiveElement

- **Attribútumok**

- **Pipe[0..*] connectedPipes:** A hozzá csatlakoztatott csövek halmaza.

- **Metódusok**

- **void tick():** Absztrakt metódus. Felüldefiniálásával a hozzácsatlakoztatott csövek közti vízfolyás menetét lehet megadni.
- **bool addPipe(Pipe pipe):** Hozzáad egy csövet. Ha már hozzá van csatlakoztatva a kapott cső, vagy a folyamat közben ezen csövön meghívott *addEndpoint* függvény *false*-szal tér vissza, akkor nem képes végrehajtani a kívánt műveletet, és *false*-szal tér vissza (egyéb esetben *true*-val). Ha sikerült a csatlakoztatás, akkor a *neighbours* listát is frissíti.
- **bool removePipe(Pipe pipe):** Megpróbálja eltávolítani magáról a paraméterként kapott csövet. Ha nincs hozzácsatlakoztatva a megadott cső, vagy a folyamat közben a csövön meghívott *removeEndpoint* függvény *false*-szal tér vissza, *false*-szal tér vissza. Sikeres esetben *true*-val tér vissza. Ha sikerült a lecsatlakoztatás, akkor a *neighbours* listát is frissíti.

4.3.2 Cistern

- **Felelősség**

Vizet vesz el a rácsatlakoztatott (szomszédos) csövekből, és eltárolja azt. Kezeli a nála létrejött új csöveket. Pumpákat lehet felvenni tőle.

- **Ősosztályok**

Element → ActiveElement → Cistern

- **Attribútumok**

- **WaterCollector collector:** A vízgyűjtő, amiben a ciszterna a vizet gyűjti.
- **Pipe[0..*] newPipes:** Az új csövek.

- **Metódusok**

- **void tick():** A hozzá csatlakoztatott csövekből kiveszi az összes vizet, és eltárolja a *collector*-ban.
- **bool removePipe(Pipe pipe):** Ha a kapott cső új cső, eltávolítja a csövet a *newPipes* listából, és *true*-val tér vissza. Egyéb esetben megpróbálja lecsatlakoztatni magáról a kapott csövet, ha ez sikerül, akkor *true*-val tér vissza, más esetben *false*-szal.

- **void spawnPipe():** Létrehoz egy új csövet, aminek egyik végét magához csatlakoztatja, és eltárolja a *newPipes* listában.
- **Pump acquirePump():** Ad egy új pumpát.

4.3.3 Character

- **Felelősség**

Az elemeken mozog, számon tartja jelenlegi tartózkodási helyét. Állítgatja a pumpák kimeneti és bemeneti csövét. Absztrakt osztály.

- **Ősosztályok**

Nincs ősosztálya.

- **Attribútumok**

- **Element location:** A hely, ahol a karakter áll.

- **Metódusok**

- **void moveTo(Element destination):** Meghívja a paraméterként kapott *Element enter* függvényét, ha ez *true*-val tér vissza, akkor meghívja a jelenlegi *location leave* függvényét és átállítja a *location* attribútumát a paraméterként kapott *Element*-re.
- **void setInputPipe(Pump pump, Pipe pipe):** Meghívja a paraméterként kapott pumpa *setInputPipe* függvényét a paraméterként kapott csövel, ha a pumpán áll.
- **void setOutputPipe(Pump pump, Pipe pipe):** Meghívja a paraméterként kapott pumpa *setOutputPipe* függvényét a paraméterként kapott csövel, ha a pumpán áll.

4.3.4 Element

- **Felelősség**

A csőrendszer eleme, a karakterek ezen képesek mozogni.

- **Ősosztályok**

Nincs ősosztálya.

- **Attribútumok**

- **Element[0..*] neighbours:** Az elem szomszédjait tároló attribútum.

- **Metódusok**

- **bool enter(Element from):** *True*-val tér vissza, ha a paraméterként kapott *Element* példány a szomszédja (tehát rá lehet lépni a paraméterként kapott *Element* példányról.)
- **void leave():** Értesíti az elemet arról, hogy lelépett róla egy játékos.

4.3.5 Pipe

- **Felelősség**

Játékbeli csövet leíró osztály. Egyes aktív elemek vizet helyezhetnek bele és szedhetnek ki belőle.

Egyszerre egy karakter állhat rajta. A szabotőrök kilyukaszthatják. A szerelők befoltozhatják, átcsatlakoztathatják a végeit más aktív elemre, helyezhetnek el rajta pumpát.

- **Ősosztályok**

Element → Pipe

- **Attribútumok**

- **int capacity:** A csőben tárolt víz kapacitása. (A maximálisan befogadható víz mértéke.)
- **int storedWater:** Adott pillanatban a csőben lévő víz mennyisége. Nem haladhatja meg a *capacity*-t.
- **bool pierced:** Ez az érték igaz, ha a cső lyukas.
- **bool occupied:** Ez az érték mondja meg, hogy éppen áll-e valaki a csövön.
- **WaterCollector ground:** A csatornahálózat alatt lévő “föld”. Ez tárolja a kifolyt víz mennyiségét.
- **ActiveElement[0..2] endpoints:** A cső végeihez csatlakoztatott aktív elemek halmaza.

- **Metódusok**

- **bool enter(Element from):** Ha a paraméterként kapott *Element* nem a cső szomszédja, vagy a csőnek van lecsatlakoztatott vége, vagy az *occupied* értéke *true*, akkor nem lehetséges rálépní, és *false* értékkel tér vissza. Ellenkező esetben átállítja az *occupied* értékét igazra, és *true*-val tér vissza.
- **void leave():** Valaki elhagyja a csövet, a cső *occupied* értéke *false*-ra változik.
- **void patch():** A cső befoltozása, azaz a *pierced* állapot *false* lesz.
- **void pierce():** A cső kilyukasztása, azaz a *pierced* változó *true*-ra állítása. A *storedWater*-ben lévő vízmennyiséget áthelyezi a *ground*-ba.
- **int addWater(int amount):** Megpróbálja eltárolni az adott mennyiségű vizet a *storedWater* attribútumban. Ha a *storedWater* meghaladná a *capacity* értékét, akkor a *capacity* és a *storedWater* különbségét tárolja el, majd visszatéríti az el nem tárolt vízmennyiséget. Ha lyukas a cső (a *pierced* értéke *true*) vagy van lecsatlakoztatott vége, akkor a víz a *storedWater* helyett a *ground*-ba kerül.
- **int removeWater(int amount):** Megpróbálja kivenni az adott mennyiségű vizet a *storedWater*-ból, és a visszatérési érték a sikeresen kivett mennyiség lesz. Ha a kérő mennyiség nagyobb, mint a *storedWater* értéke, akkor a visszatérési érték a *storedWater* értéke lesz, és a *storedWater* értéke 0 lesz.
- **int removeAllWater():** Kiveszi az összes vizet a csőből és visszatér a kivett mennyiséggel.
- **bool addEndpoint(ActiveElement activeElement):** Csatlakoztatja magához a kapott *activeElement*-et (azaz hozzáadja az *endpoints*-hoz), ha az még nincs benne az *endpoints*-ban és az *endpoints*-nak még nincs 2 eleme, majd *true*-val tér vissza. Egyéb esetben a csatlakoztatás sikertelen és a függvény visszatérése *false* lesz. Ha sikerült a csatlakoztatás, akkor a *neighbours* listát is frissíti.
- **bool removeEndpoint(ActiveElement activeElement):** Ha az *endpoints* listában benne van a kapott *activeElement*, és az *occupied* értéke *false*, akkor eltávolítja a kapott *activeElement*-et az *endpoints* listából, kifolyatja a csőben lévő vizet a földre, és *true* értéket ad vissza. Más esetben a visszatérési érték *false* lesz. Ha sikerült a lecsatlakoztatás, akkor a *neighbours* listát is frissíti.
- **Pipe split():** A csövet kettévágja, vagyis a jelenlegi cső egyik endpoint-ját lecsatlakoztatja. A folyamat alatt új cső objektum jön létre, és ez összecsatlakoztatásra kerül az előzőleg lecsatlakoztatott endpoint-tal. Az újonnan létrehozott csövet visszatéríti.

4.3.6 Plumber

- **Felelősség**

Képes pumpákat felvenni/lehelyezni, csöveket le- és felcsatlakoztatni, illetve pumpákat és csöveket javítani. A ciszternákba gyűjtik a vizet.

- **Ősosztályok**

Character → Plumber

- **Attribútumok**

- **Pump[0..*] heldPumps:** A szerelőnél lévő pumpák.
- **Pipe heldPipe:** A cső aminek a vége a kezében van. Ha nincs nála cső, akkor ez *null*.

- **Metódusok**

- **void repair(Pipe pipe):** Ha a paraméterként kapott csövön áll, akkor megjavítja.
- **void repair(Pump pump):** Ha a paraméterként kapott pumpán áll, akkor megjavítja.
- **void attachPipe(ActiveElement target):** Ha a paraméterként kapott *ActiveElement*-en áll, és a *heldPipe* értéke nem *null* (azaz tart csövet a kezében), megpróbálja felcsatlakoztatni ezt a csövet a kapott *ActiveElement*-re. Ha ez sikeres, akkor a *heldPipe* értéke *null* lesz.
- **void detachPipe(ActiveElement target, Pipe pipe):** Ha a paraméterként kapott *ActiveElement*-en áll, és a *heldPipe* értéke *null* (azaz nem tart csövet a kezében), megpróbálja lecsatlakoztatni a paraméterként kapott csövet a kapott aktív elemről. Ha ez sikeres, a lecsatlakoztatott cső a *heldPipe*-ban tárolódik el.
- **void pickUpPump(Cistern cistern):** Ha a paraméterként kapott ciszternán áll, akkor a ciszternától kér egy pumpát.
- **void placePump(Pipe pipe):** Ha a paraméterként kapott csövön áll, és van nála pumpa, akkor azt a pumpát lehelyezi. Ehhez a kapott csövön meghívja a *split* függvényt, és a pumpába beköti a *pipe*-ot és a *split* visszatérési értékeként kapott másik csövet. Ezután a lehelyezett pumpára áll.

4.3.7 Pump

- **Felelősség**

El tud romlani. Amikor nincs elromolva, akkor kimeneti és bemeneti csövei között a víztartályán keresztül folyatja a vizet. Kimeneti és bemeneti csövei állíthatók.

- **Ősosztályok**

Element → ActiveElement → Pump

- **Attribútumok**

- **int tankCapacity:** A pumpa víztartályának víz tároló kapacitása. (A maximálisan befogadható víz mértéke.)
- **int tankStoredWater:** A pumpa víztartályában tárolt víz mennyisége. Nem haladhatja meg a *tankCapacity*-t.
- **bool broken:** *True* ha a pumpa el van romolva. *False*, ha helyesen működik.
- **int ports:** A pumpa portjainak száma, vagyis a maximum hozzácsatlakoztatható csövek száma.
- **Pipe inputPipe:** A pumpa bemeneti csöve.

- **Pipe outputPipe:** A pumpa kimeneti csöve.
- **Metódusok**
 - **void tick():** Bemeneti csövéről próbál vizet pumpálni víztárolóján keresztül a kimeneti csövébe, ha a pumpa működik. Elsőként kimeneti csövéről próbál pumpálni, majd a bemeneti csőről a saját víztárolójába.
 - **bool addPipe(Pipe pipe):** Megpróbálja felcsatlakoztatni magára a kapott csövet. Ha már az összes portja foglalt, nem sikerül a felcsatlakoztatás és *false*-al tér vissza. Egyéb esetben, ha nincs még rácsatlakoztatva az adott cső, *true*-val tér vissza.
 - **bool removePipe(Pipe pipe):** Megpróbálja lecsatlakoztatni magáról a kapott csövet. Ha ez sikerül, akkor *true*-val tér vissza, más esetben *false*-szal. Ha a lecsatlakoztatott cső az *inputPipe* vagy az *outputPipe* volt, akkor az adott érték is nullázódik.
 - **void repair():** Átállítja a *broken* változó értékét *false*-ra.
 - **void break():** Átállítja a *broken* változó értékét *true*-ra.
 - **void setInputPipe(Pipe pipe):** Átállítja az *inputPipe* értékét a kapott csőre, ha az a cső csatlakozik a pumpához és nincs beállítva kimeneti csőnek.
 - **void setOutputPipe(Pipe pipe):** Átállítja az *outputPipe* értékét a kapott csőre, ha az a cső csatlakozik a pumpához és nincs beállítva bemeneti csőnek.

4.3.8 Saboteur

- **Felelősség**
A csőhálózaton mozognak és csöveket lyukasztgatnak. A földre gyűjtik a vizet.
- **Ősosztályok**
Character → Saboteur
- **Metódusok**
 - **void pierce(Pipe pipe):** Kilyukasztja a paraméterként kapott csövet, ha rajta áll.

4.3.9 Spring

- **Felelősség**
Vizet folyat a rácsatlakoztatott csövekbe.
- **Ősosztályok**
Element → ActiveElement → Spring
- **Attribútumok**
 - **int waterDispensedPerTick:** A tickenként kibocsátott víz mennyisége.
- **Metódusok**
 - **void tick():** A *connectedPipes*-okba a *waterDispensedPerTick*-nek megfelelő mennyiségű vizet tölt (a cső által visszadobott vízzel nem foglalkozik).

4.3.10 WaterCollector

- **Felelősség**
Korlátlan kapacitással vizet tárol. Számon tartja a belefolyt víz mennyiségét.

- **Ősosztályok**

Nincs ősosztálya.

- **Attribútumok**

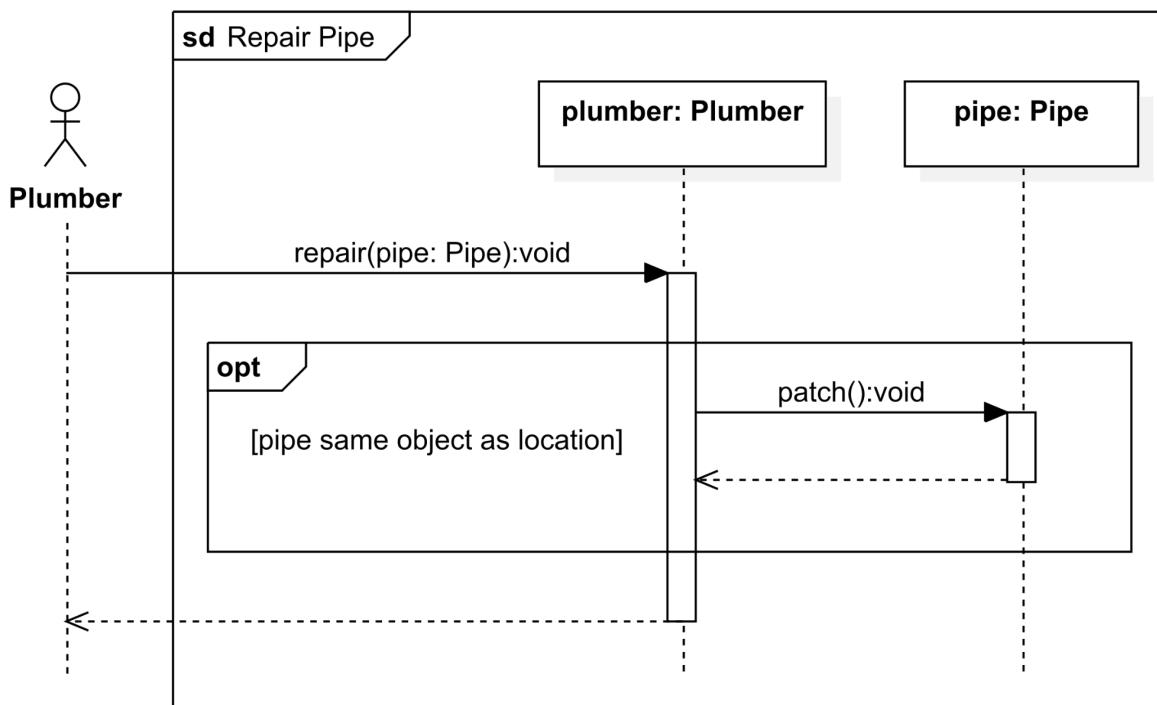
- **storedAmount:** Az összegyűjtött víz mennyisége.

- **Metódusok**

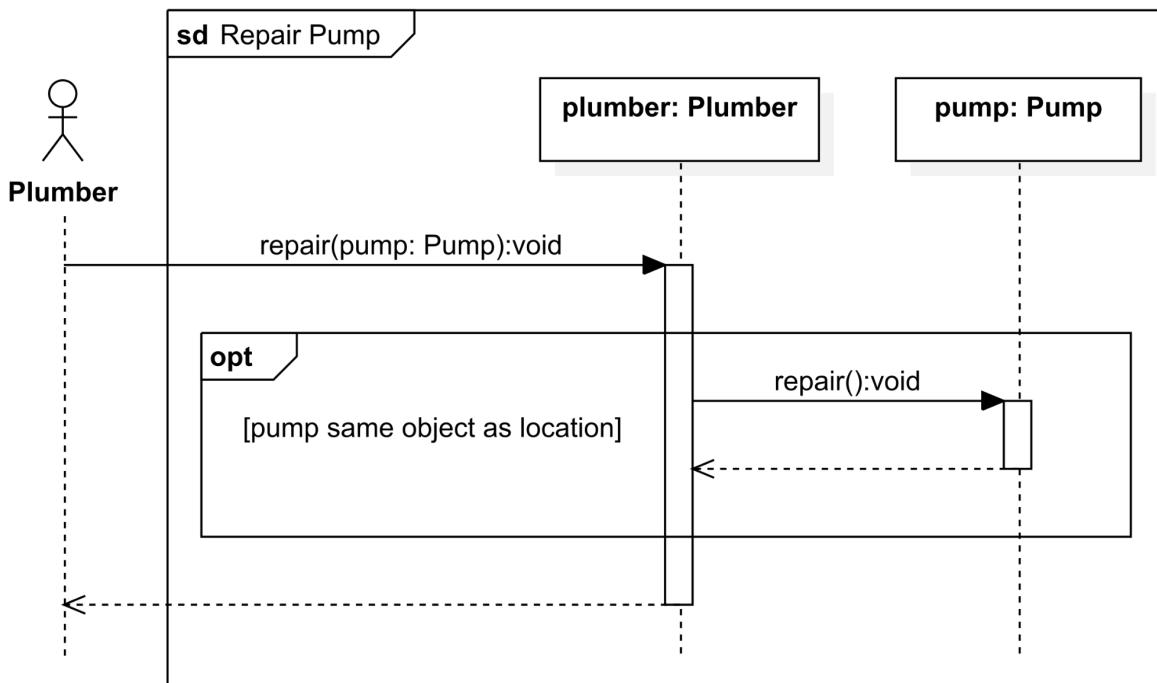
- **void storeAmount(int amount):** A kapott vízmennyiséget befogadja.

4.4 Szekvencia diagramok

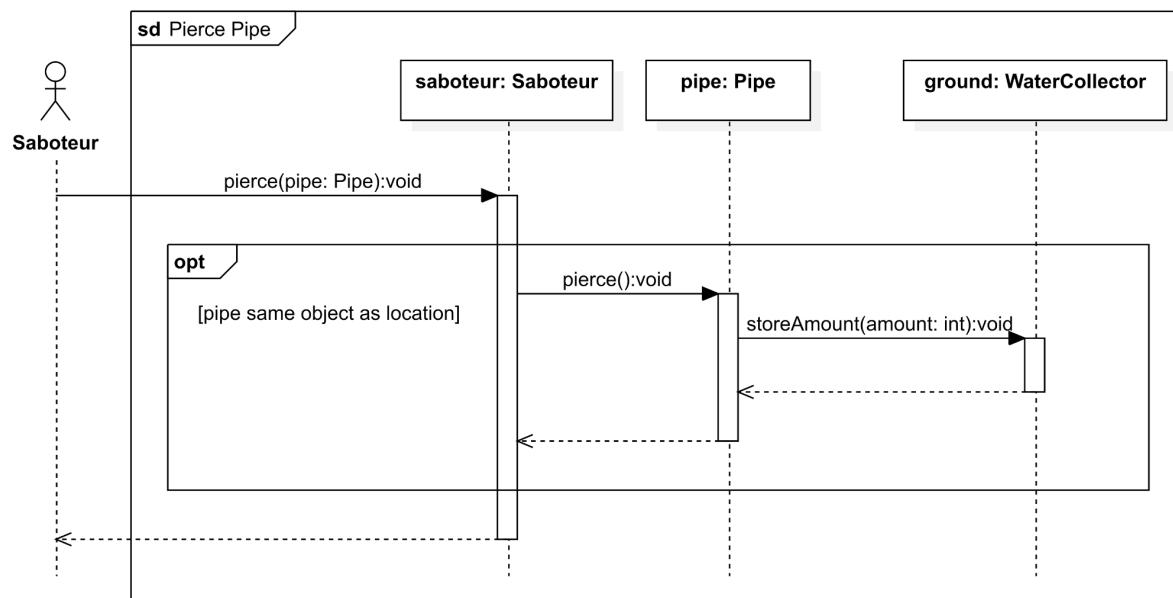
4.4.1 Repair Pipe



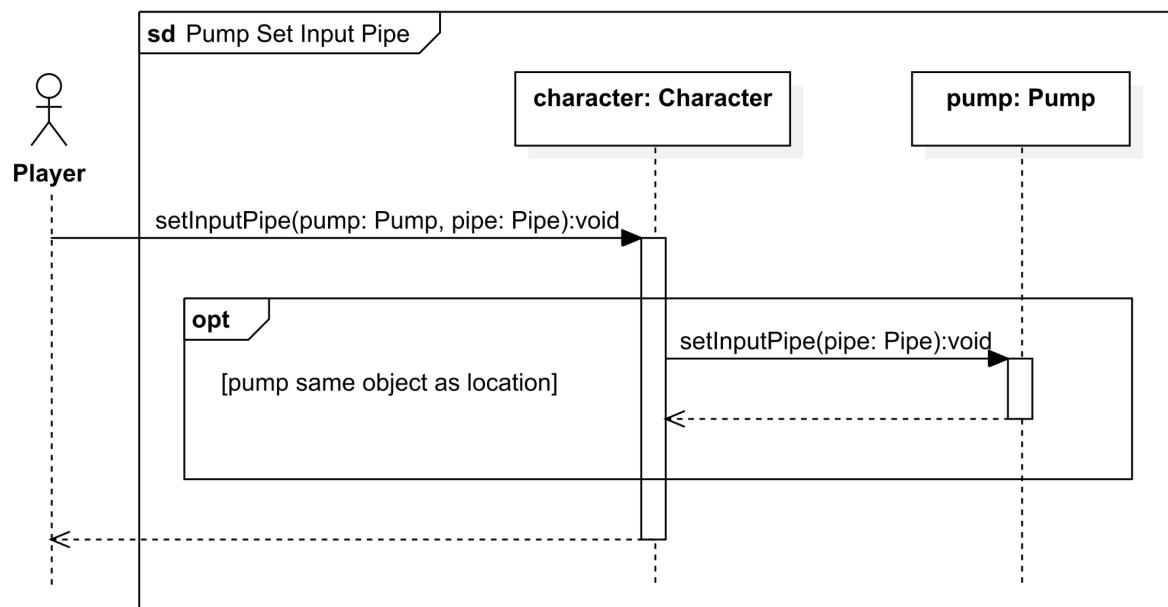
4.4.2 Repair Pump



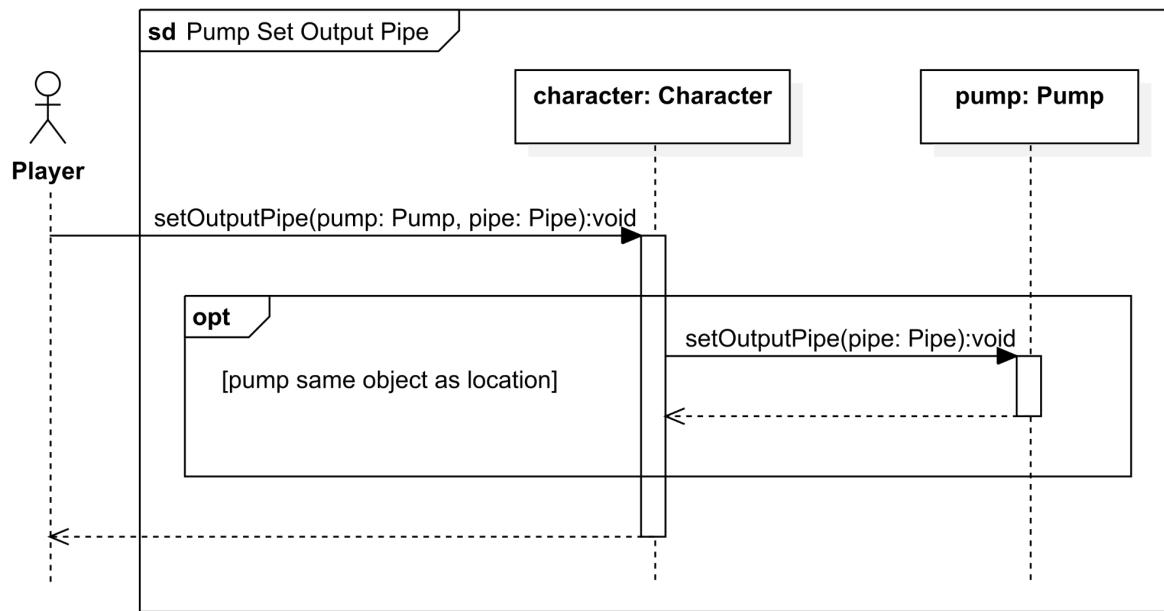
4.4.3 Pierce Pipe



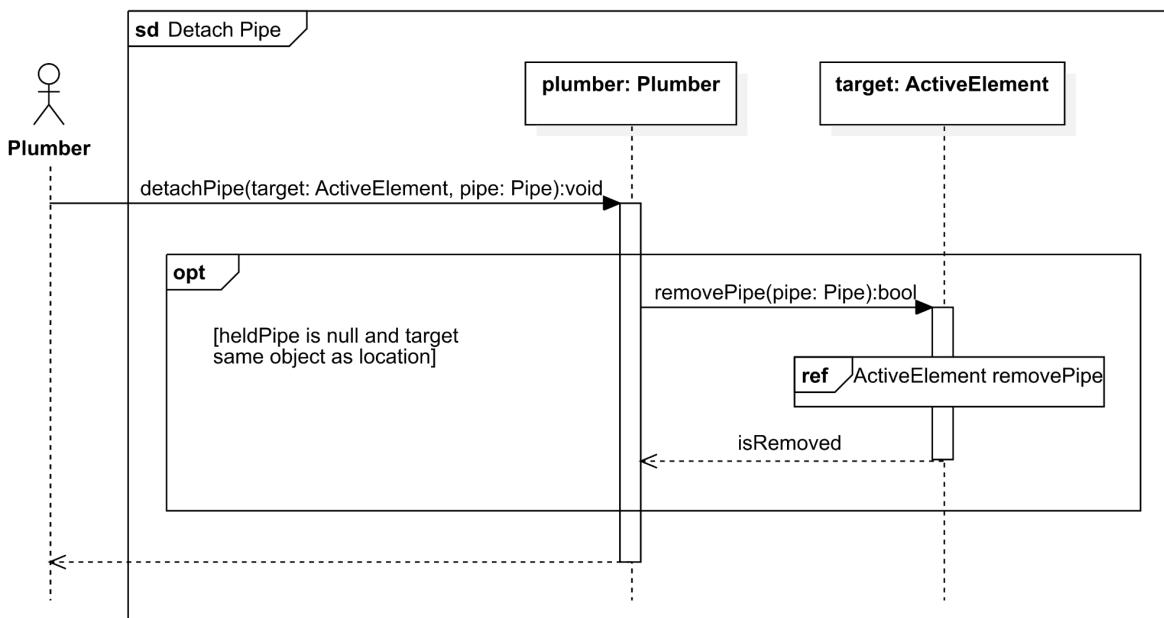
4.4.4 Pump Set Input Pipe



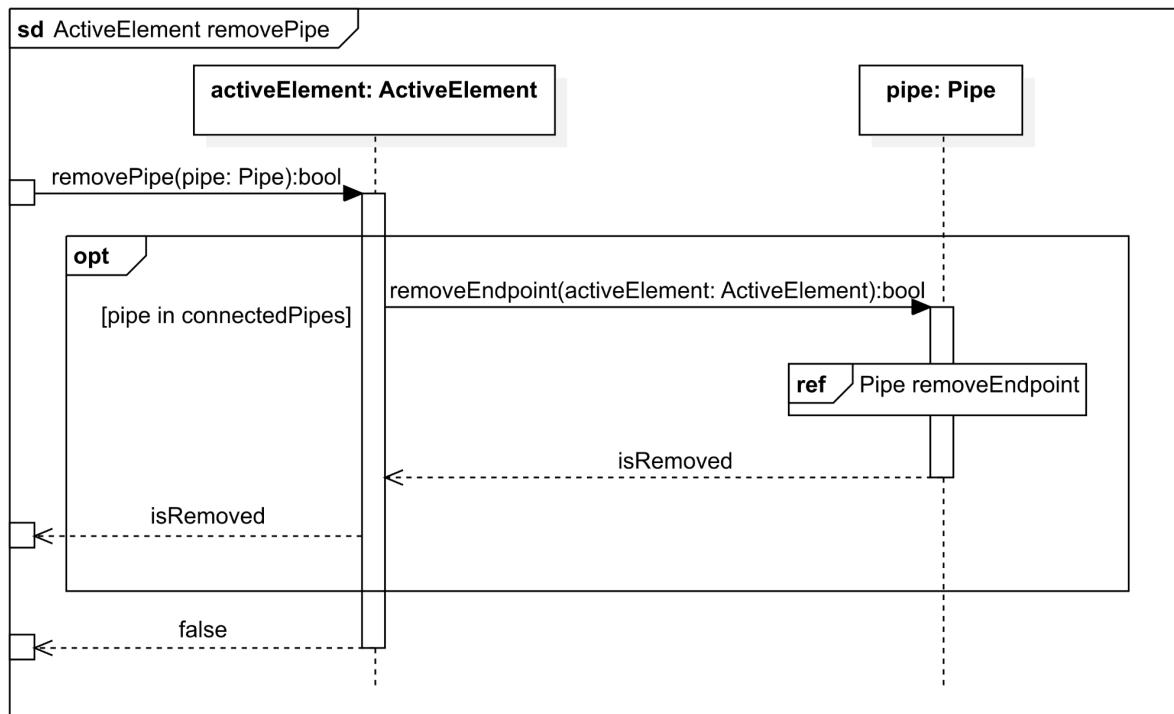
4.4.5 Pump Set Output Pipe



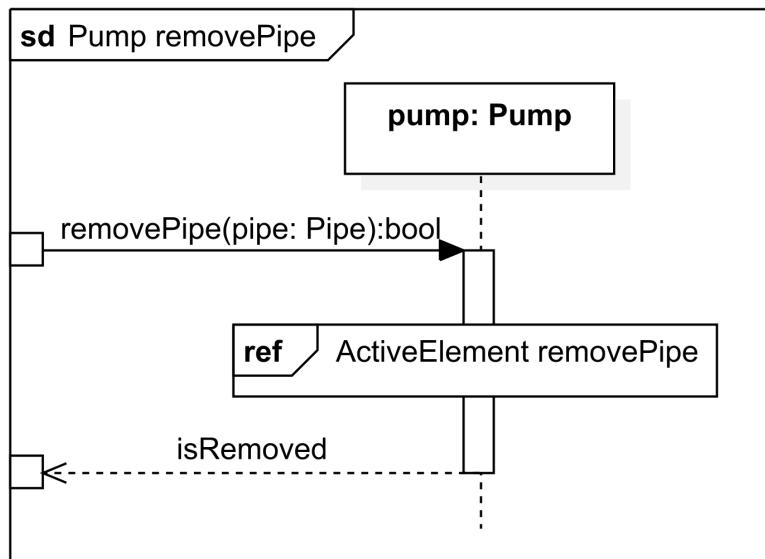
4.4.6 Detach Pipe



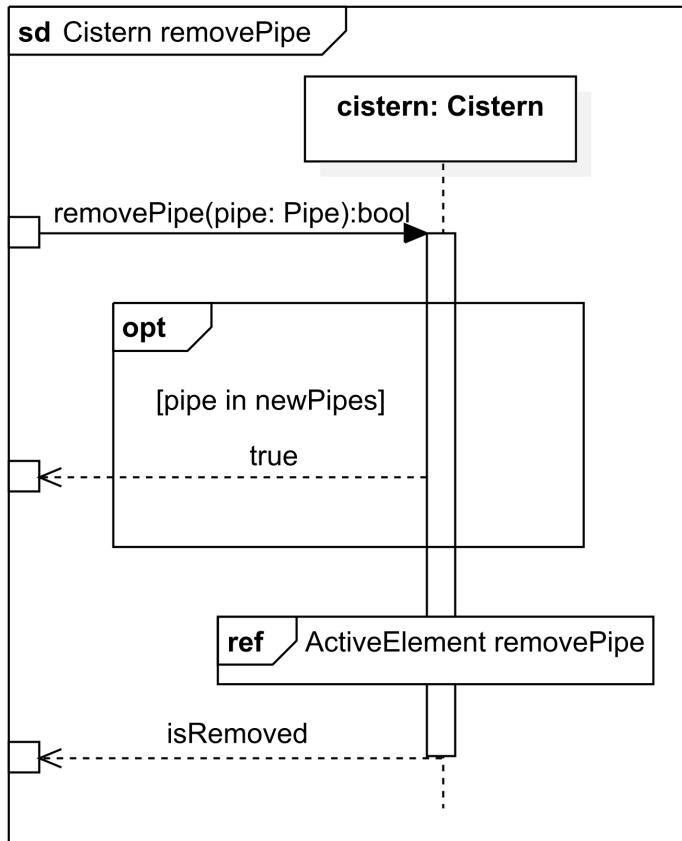
4.4.7 ActiveElement removePipe



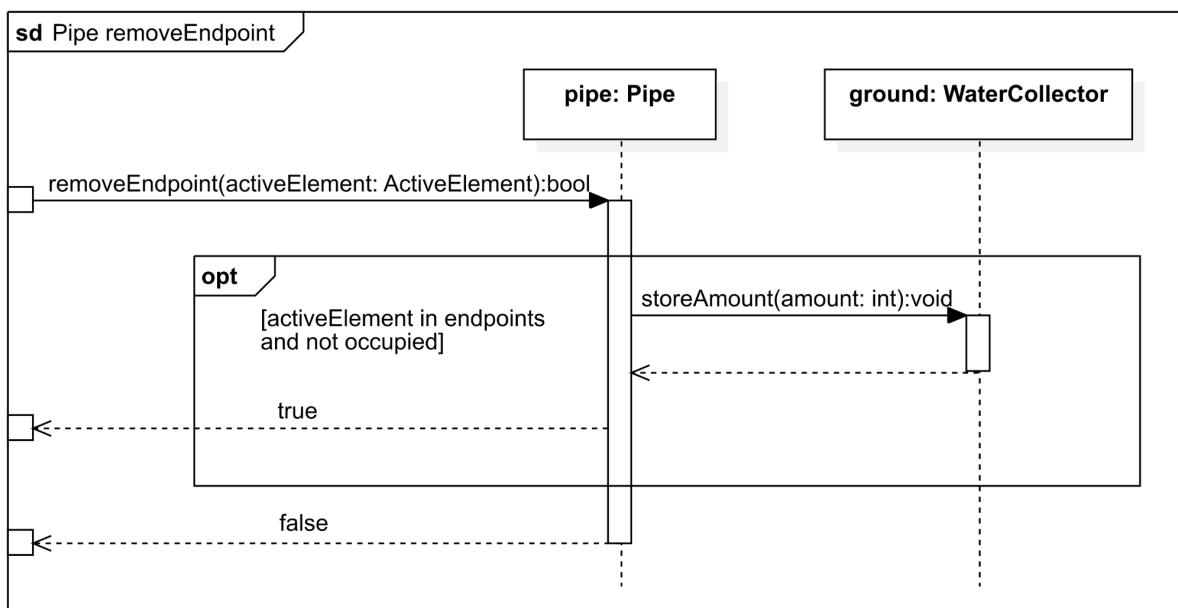
4.4.8 Pump removePipe



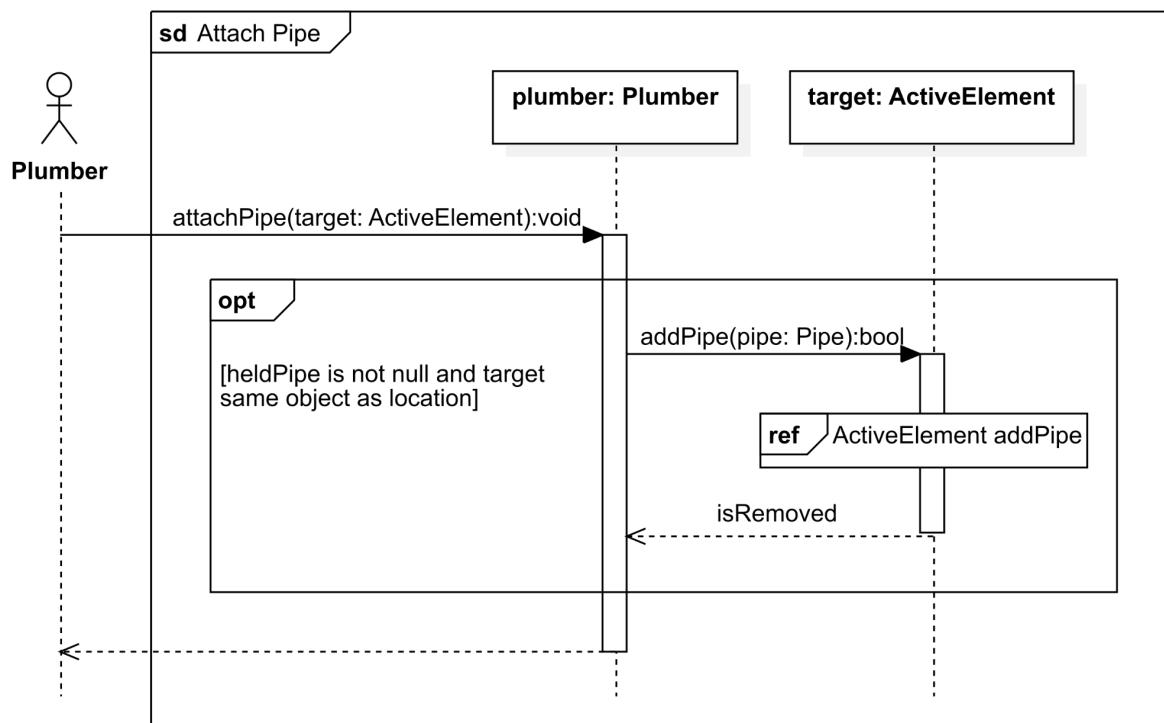
4.4.9 Cistern removePipe



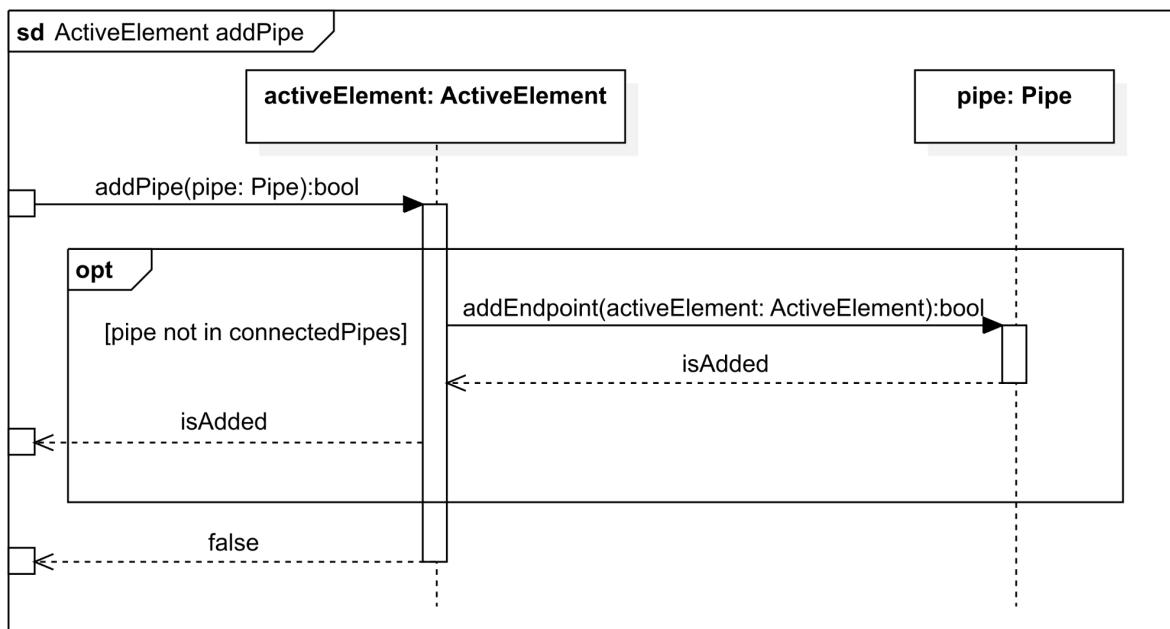
4.4.10 Pipe removeEndpoint



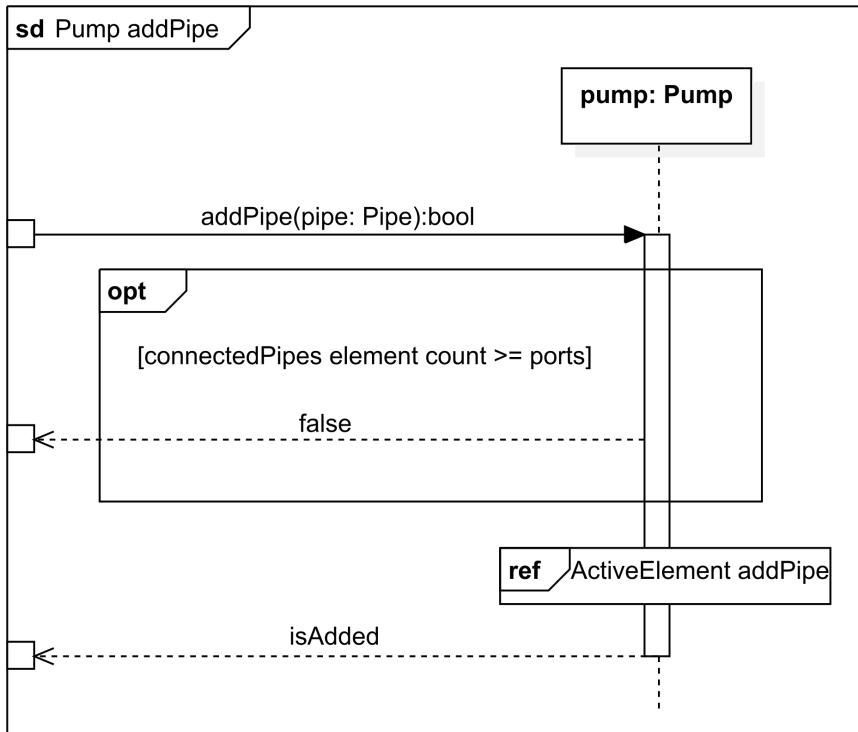
4.4.11 Attach Pipe



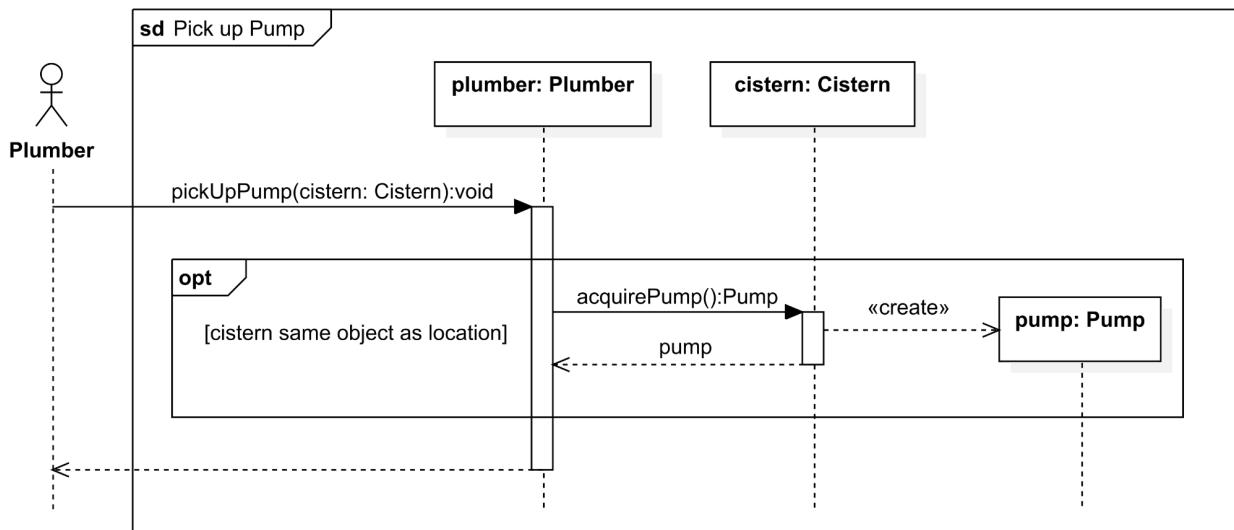
4.4.12 ActiveElement addPipe



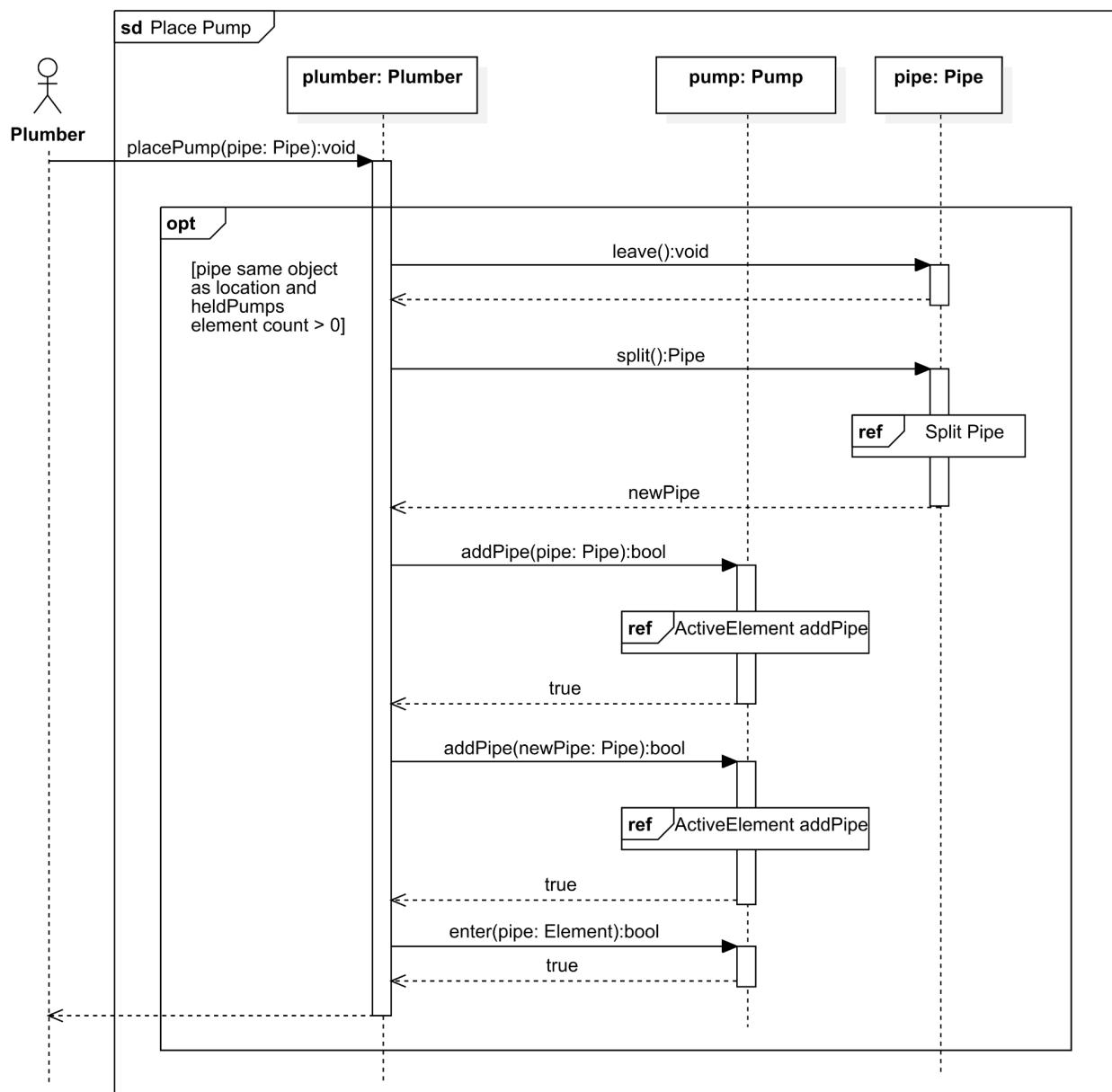
4.4.13 Pump addPipe



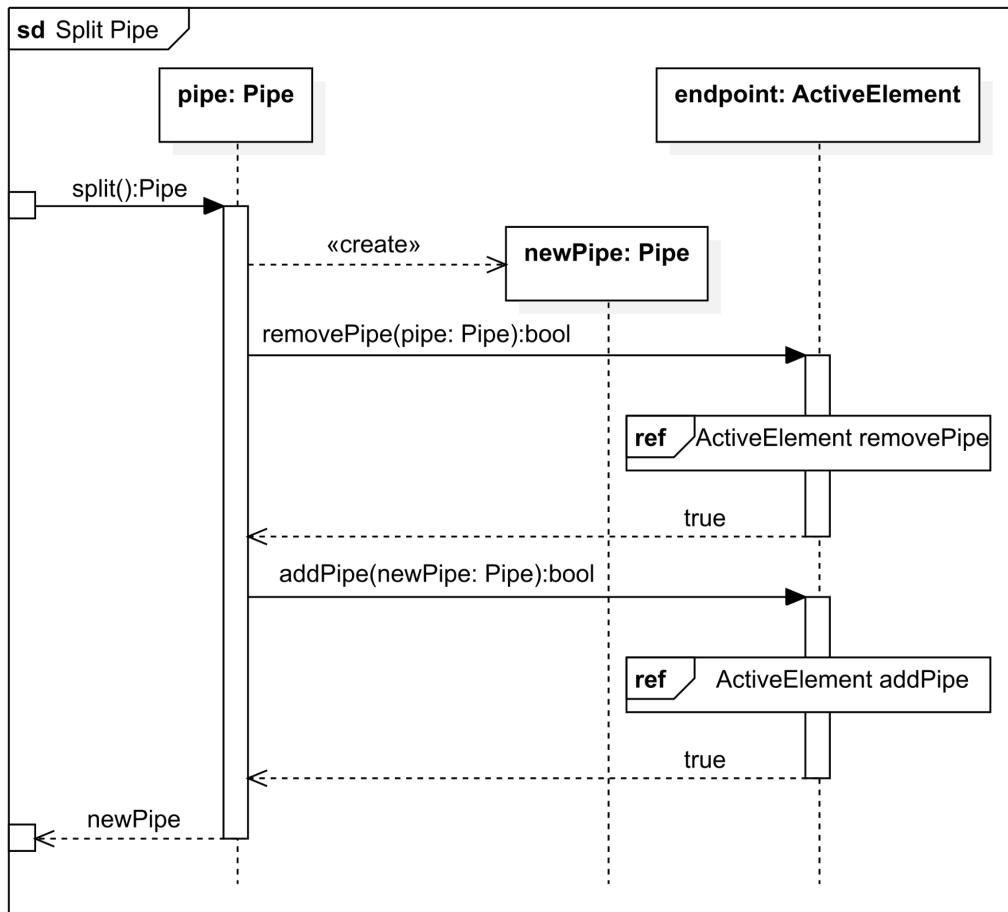
4.4.14 Pick up Pump



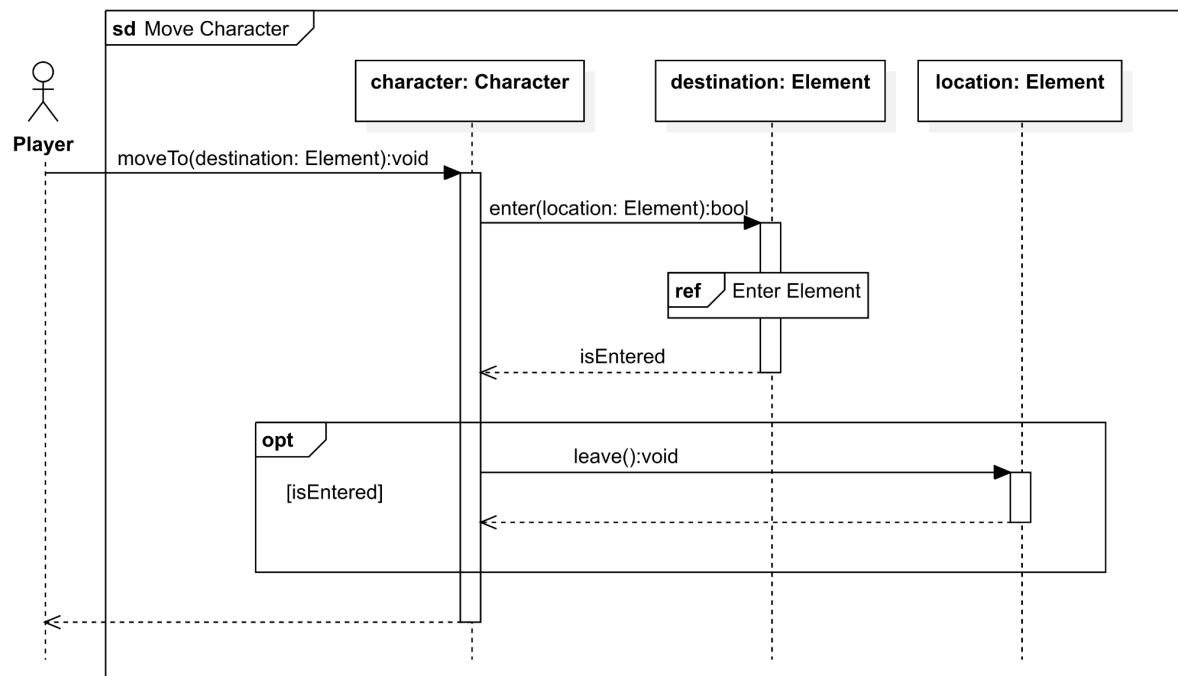
4.4.15 Place Pump



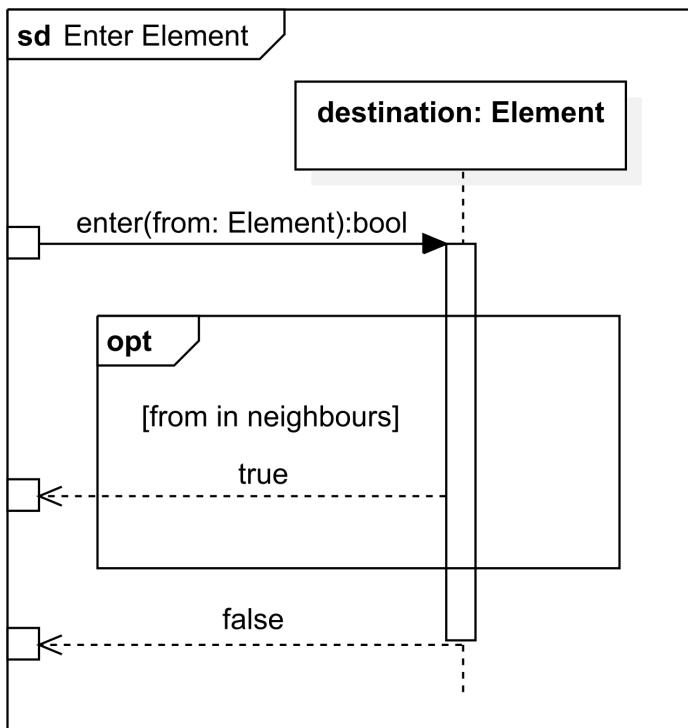
4.4.16 Split Pipe



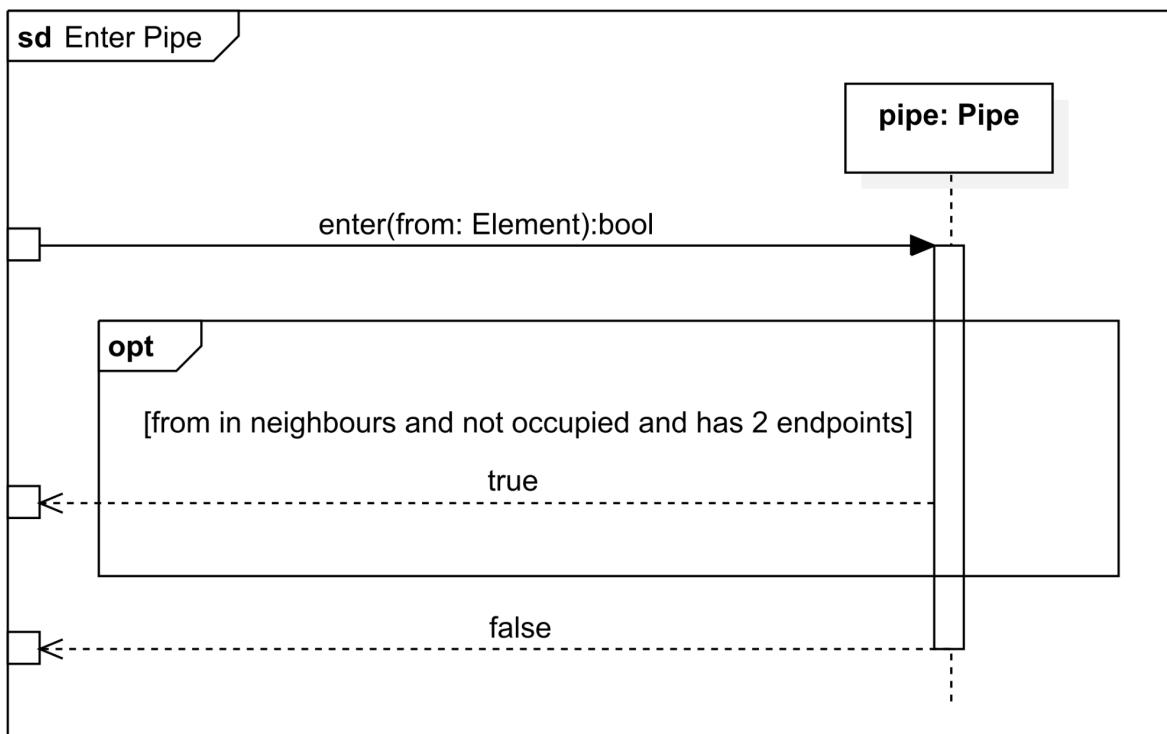
4.4.17 Move Character

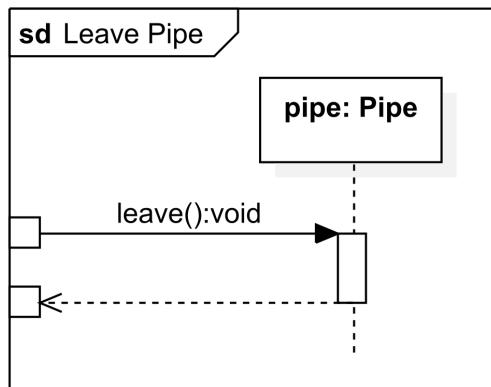
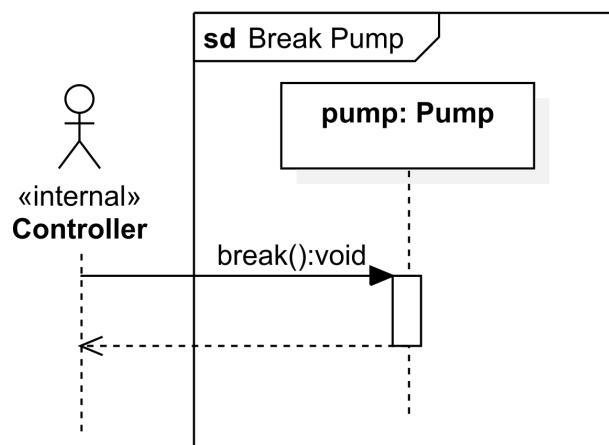


4.4.18 *Enter Element*

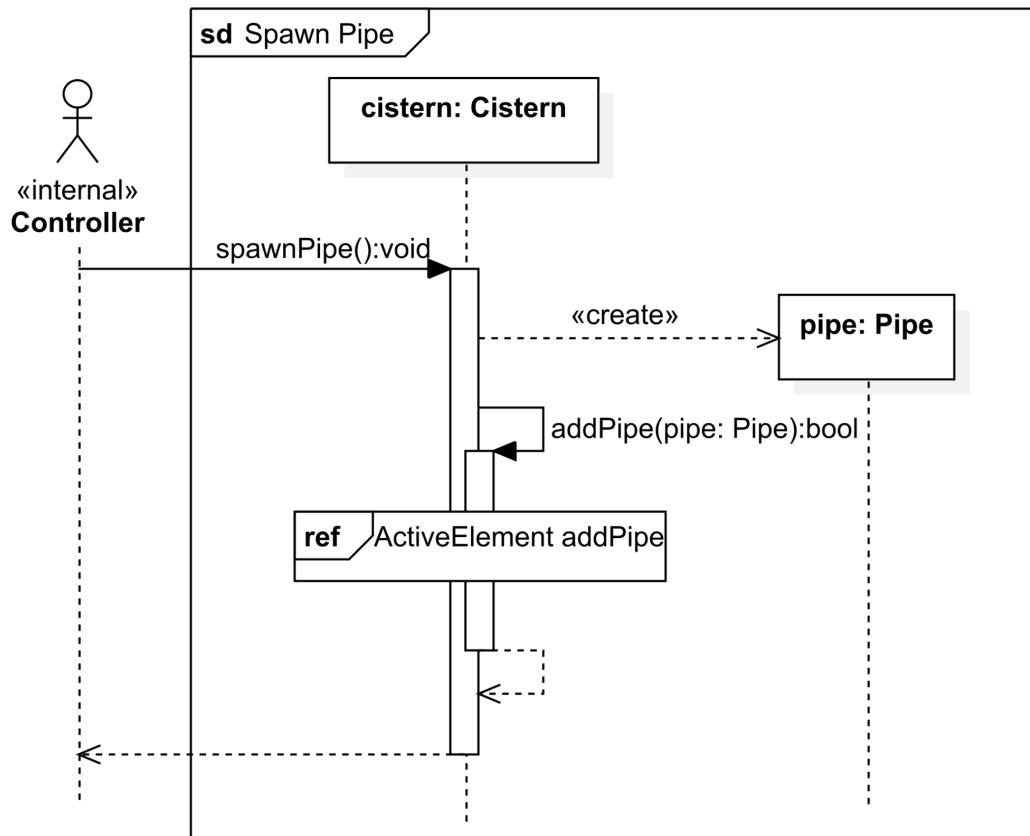


4.4.19 *Enter Pipe*

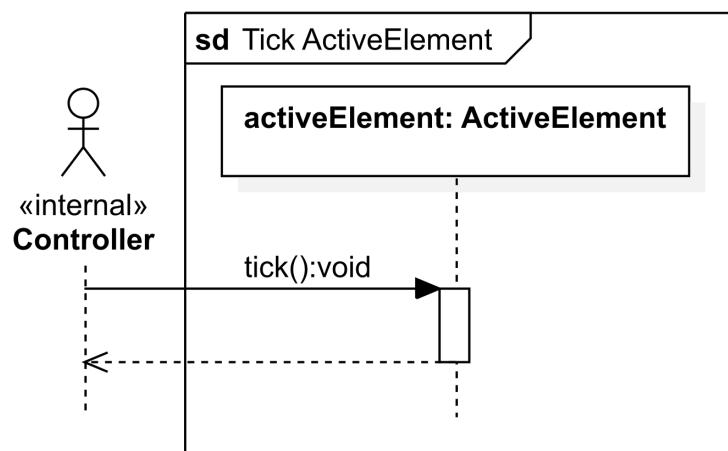


4.4.20 *Leave Pipe***4.4.21 *Break Pump***

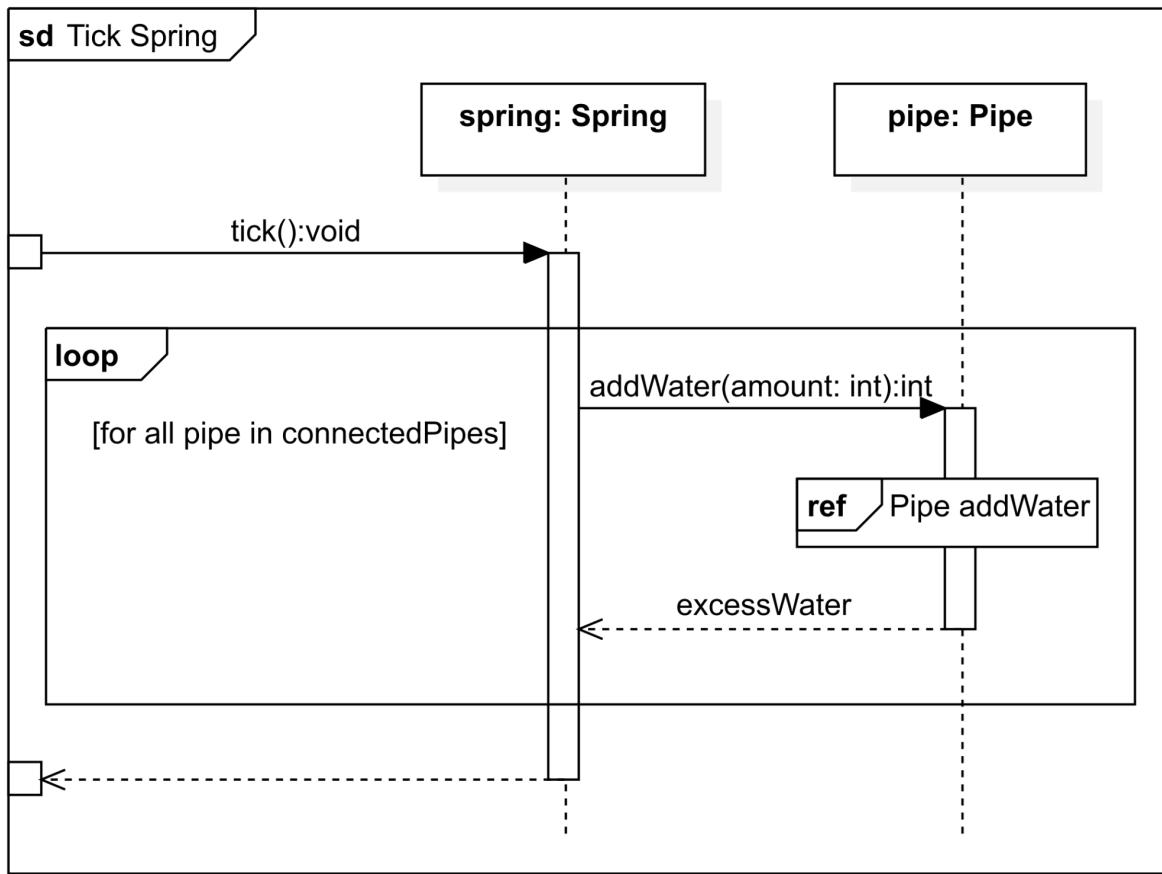
4.4.22 *Spawn Pipe*



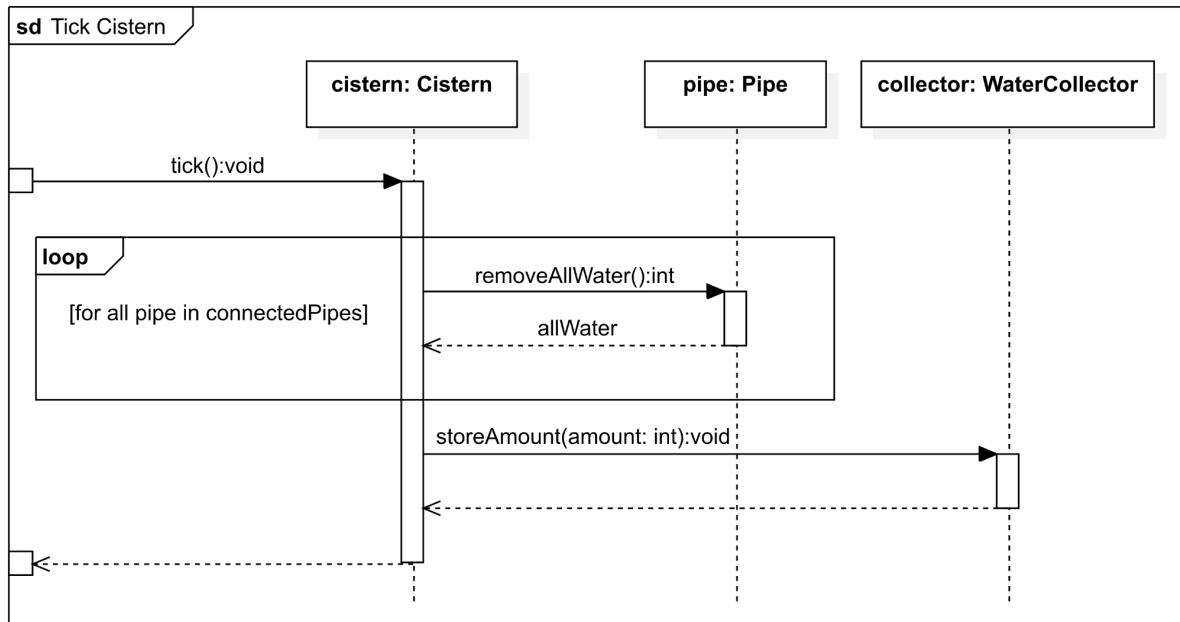
4.4.23 *Tick ActiveElement*



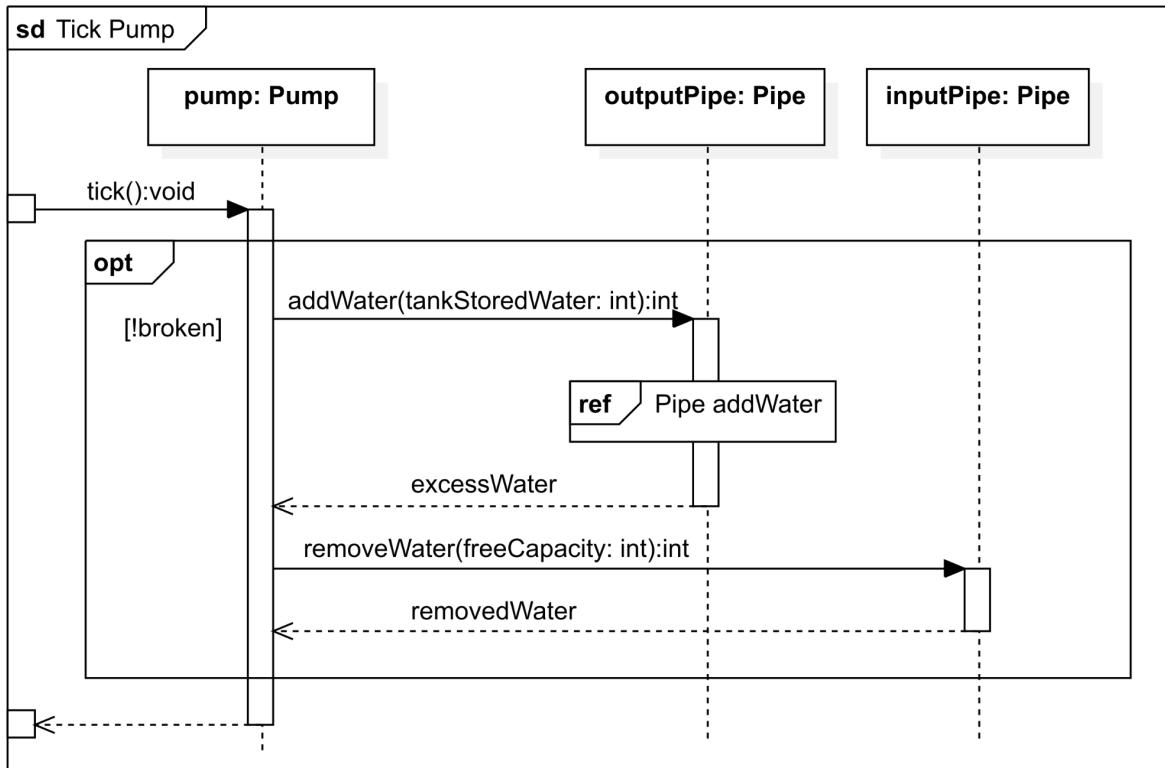
4.4.24 *Tick Spring*



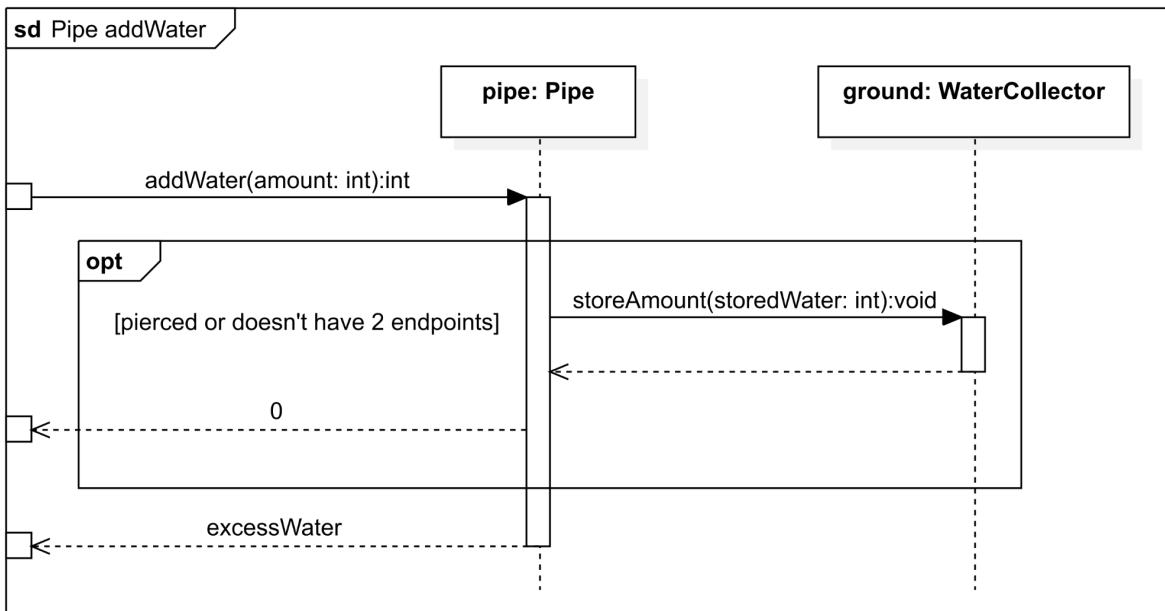
4.4.25 *Tick Cistern*



4.4.26 Tick Pump

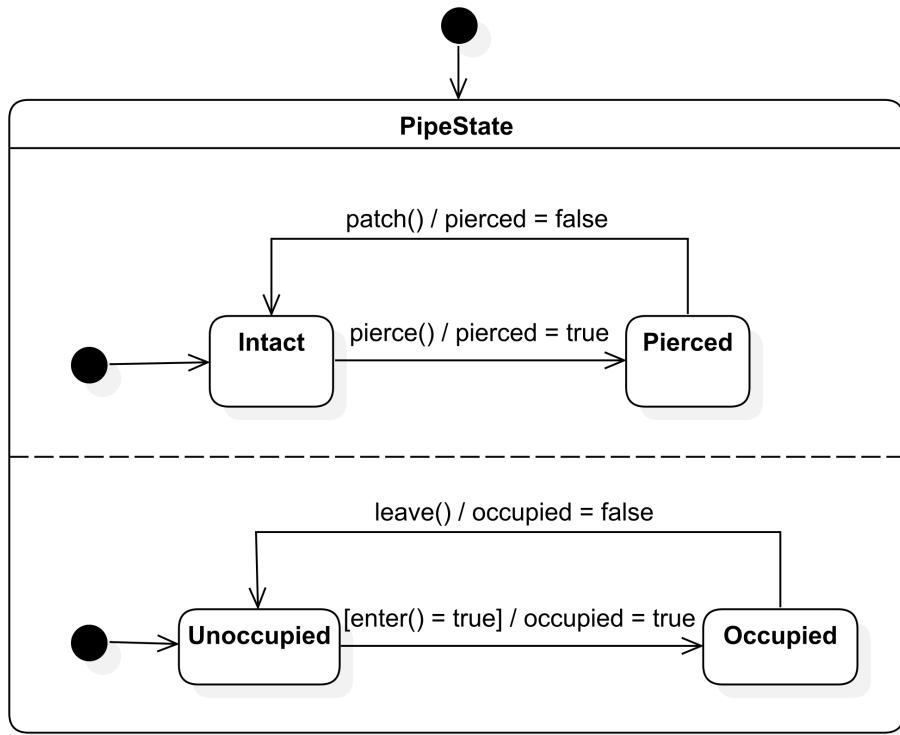


4.4.27 Pipe addWater

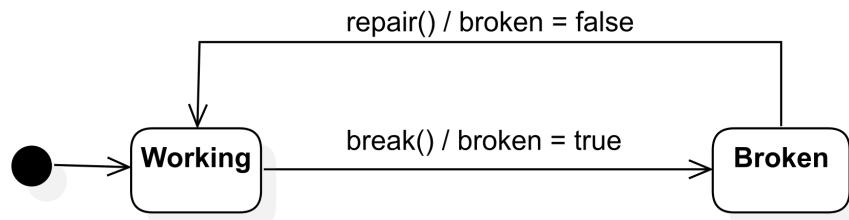


4.5 State-chartok

4.5.1 A Pipe osztály diagramja



4.5.2 A Pump osztály diagramja



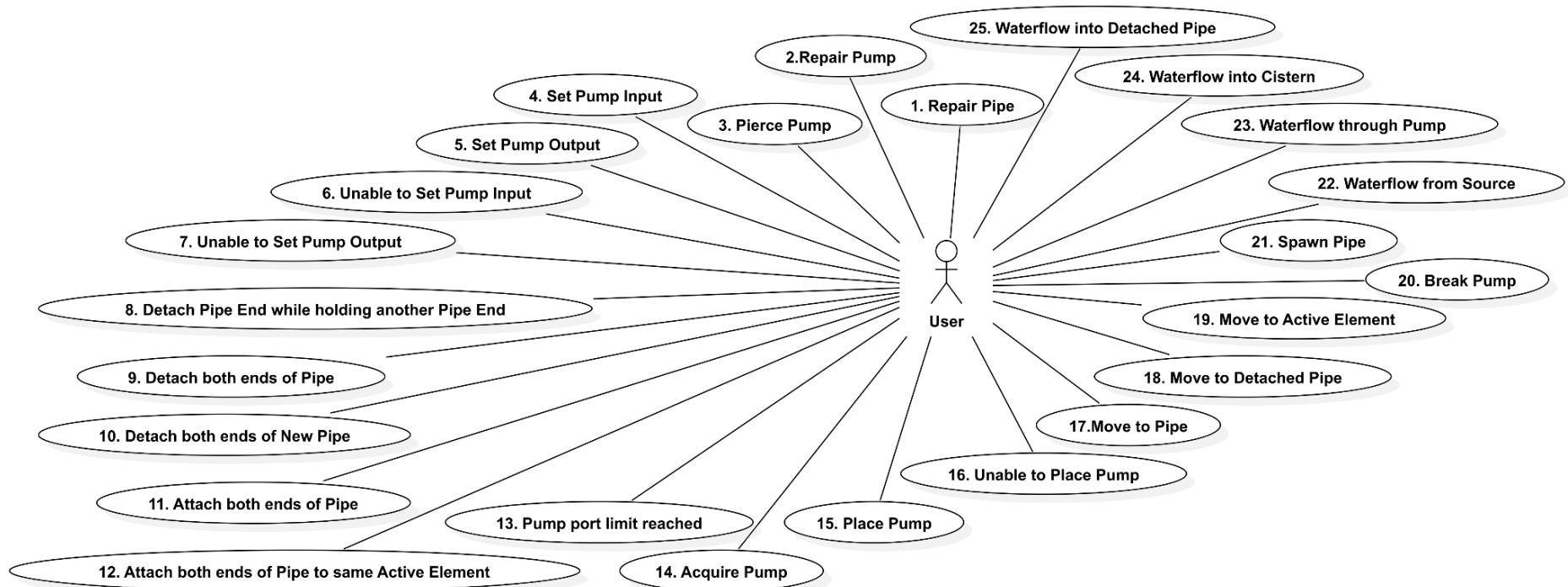
4.6 Napló

Kezdet	Időtartam	Résznevők	Leírás
2023.03.22. 12:15	1,5 óra	Galgóczy Kereszthury Szilczer Tóth	Értekezlet: Konzultáltunk az IL408-ban a laborvezetővel. Döntés: Az osztálydiagramot és a szekvenciadiagramokat át kell alakítani, az Action-ös megvalósítást lecserélni valami másra. A következő értekezleten kiosztjuk a feladatokat.
2023.03.24. 18:00	1 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: Elvégzendő feladatok kiosztása. Döntés: Tóth módosítja a 4.2-ben szereplő osztálydiagramot. Kereszthury módosítja a 4.3-ban szereplő osztályleírásokat. Szilczer módosítja a 4.4-ben szereplő szekvenciadiagramokat.
2023.03.24. 19:00	3 óra	Tóth	Tevékenység: 4.2 osztálydiagram módosítása
2023.03.24. 19:00	3 óra	Kereszthury	Tevékenység: 4.3 osztályleírások módosítása 4.5 State-chartok módosítása.
2023.03.24. 22:00	3 óra	Galgóczy	Tevékenység: 4.3 osztályleírások ellenőrzése, összevetése az osztálydiagrammal.
2023.03.24. 20:00	2 óra	Szilczer	Tevékenység: 4.4 Szekvenciadiagramok újrarendezése
2023.03.25. 12:00	2 óra	Szilczer	Tevékenység: 4.4 Szekvenciadiagramok újrarendezése
2023.03.25. 16:00	6 óra	Szilczer	Tevékenység: 4.4 Szekvenciadiagramok újrarendezése
2023.03.26. 20:00	2,5 óra	Galgóczy	Tevékenység: 4.2, 4.3, 4.4 pontok ellenőrzése
2023.03.26. 22:30	2,5 óra	Tóth	Tevékenység: 4.2, 4.3, 4.4 pontok javítása
2023.03.27. 11:00	0,5 óra	Tóth	4.4 Szekvenciadiagramok dokumentumba illesztése, formázása
2023.03.27. 11:30	1 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: A teljes dokumentum átnézése. Döntés: A dokumentum beadásra készen áll.

5. Szkeleton tervezése

5.1 A szkeleton modell valóságos use-case-ai

5.1.1 Use-case diagram



5.1.2 Use-case leírások

Use-case neve	1. Repair Pipe
Rövid leírás	A felhasználó szerelőt irányít, aki megjavít egy lyukas csövet.
Aktorok	User
Forgatókönyv	1. A szerelő megjavítja a kilyukasztott csövet, amin áll.

Use-case neve	2. Repair Pump
Rövid leírás	A felhasználó szerelőt irányít, aki megjavít egy elromlott pumpát.
Aktorok	User
Forgatókönyv	1. A szerelő megjavítja a pumpát, amin áll.

Use-case neve	3. Pierce Pipe
Rövid leírás	A felhasználó szabotört irányít, aki kilyukaszt egy csövet.
Aktorok	User
Forgatókönyv	1. A szabotör kilyukasztja a csövet, amin áll. 2. A csőből kifolyik a benne lévő víz a földre.

Use-case neve	4. Set Pump Input
Rövid leírás	A felhasználó szerelőt irányít, aki átállítja egy pumpa bemenetét egy kiválasztott csőre. (ugyanez működik szabotörrel is)
Aktorok	User
Forgatókönyv	1. A szerelő meghívja a bemenet átállító függvényt, benne a kiszemelt csővel. 2. A pumpa ellenőrzi, hogy az adott cső nincs már csatlakoztatva, mint kimeneti cső. 3. A pumpa átállítja a bemeneti csövet.

Use-case neve	5. Set Pump Output
Rövid leírás	A felhasználó szerelőt irányít, aki átállítja egy pumpa kimenetét egy kiválasztott csőre. (ugyanez működik szabotörrel is)
Aktorok	User
Forgatókönyv	1. A szerelő meghívja a kimenet átállító függvényt, benne a kiszemelt csővel. 2. A pumpa ellenőrzi, hogy az adott cső nincs már csatlakoztatva, mint bemeneti cső. 3. A pumpa átállítja a kimeneti csövet.

Use-case neve	6. Unable to Set Pump Input
Rövid leírás	A felhasználó szerelőt irányít, aki átállítja egy pumpa bemenetét egy kiválasztott csőre, de a cső már be van állítva kimenetinek, ezért ez nem sikerül. (ugyanez működik szabotörrel is)
Aktorok	User
Forgatókönyv	1. A szerelő meghívja a bemenet átállító függvényt, benne a kiszemelt csővel.

	<ol style="list-style-type: none"> 2. A pumpa ellenőrzi, hogy az adott cső nincs már csatlakoztatva, mint kimeneti cső. 3. Mivel ez a cső a kimeneti cső is, pumpa nem állítja át a bemeneti csövet.
--	--

Use-case neve	7. Unable to Set Pump Output
Rövid leírás	A felhasználó szerelőt irányít, aki át állítja egy pumpa kimenetét egy kiválasztott csőre, de a cső már be van állítva bemenetinek, ezért ez nem sikerül. (ugyanez működik szabotőrel is)
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A szerelő meghívja a kimenet átállító függvényt, benne a kiszemelt csővel. 2. A pumpa ellenőrzi, hogy az adott cső nincs már csatlakoztatva, mint bemeneti cső. 3. Mivel ez a cső a bemeneti cső is, pumpa nem állítja át a kimeneti csövet.

Use-case neve	8. Detach Pipe End while holding another Pipe End
Rövid leírás	A felhasználó szerelőt irányít, aki megpróbál lecsatlakoztatni egy csövet úgy, hogy egy csővég már a kezében van, ezért a lecsatlakoztatás nem sikerül.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A szerelő lecsatlakoztatja a kiszemelt csövet, de ez nem sikerül, mert már van a kezében csővég.

Use-case neve	9. Detach both ends of Pipe
Rövid leírás	A felhasználó két szerelőt irányít, mindenketen különböző aktív elemen állnak, és egy, a két aktív elemet összekötő csövet csatlakoztatnak le mindenketten.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. Az egyik szerelő lecsatlakoztat egy csövet az aktív elemről, amin éppen áll, a szerelő "kezébe" kerül a lecsatlakoztatott csővég. 2. A csőben lévő víz a földre folyik. 3. A másik szerelő lecsatlakoztatja ugyanannak a csőnek a másik végét arról az aktív elemről, amin éppen áll, ez a szerelő kezébe kerül. 4. A csőben lévő víz a földre folyik.
Alternatív forgatókönyv	<ol style="list-style-type: none"> 1.A. A cső foglalt, ezért a lecsatlakoztatás nem sikerül, a szerelő keze üres marad. 2.A. Az 1.A. miatt a cső nem kerül lecsatlakoztatásra, ezért nem folyik ki belőle a víz. 3.A. A cső foglalt, ezért a lecsatlakoztatás nem sikerül, a szerelő keze üres marad. 4.A. A 3.A. miatt a cső nem kerül lecsatlakoztatásra, ezért nem folyik ki belőle a víz.

Use-case neve	10. Detach both ends of New Pipe
----------------------	----------------------------------

Rövid leírás	A felhasználó két szerelőt irányít, akik egy ciszternáról egy új cső minden két végét lecsatlakoztatják.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> Az egyik szerelő lecsatlakoztatja az új cső távoli végét. A cső megszűnik új csőnek lenni. A másik szerelő lecsatlakoztatja a cső ciszternához közeli végét.
Alternatív forgatókönyv	1.A. A cső foglalt, ezért a felvétel nem sikerült. Normál működés mellett ez nem fordulhat elő, mivel a cső új cső.

Use-case neve	11. Attach both ends of Pipe
Rövid leírás	A felhasználó 2 szerelőt irányít, minden két szerelő kezében ugyanazon cső egy-egy vége van. A szerelők két különböző aktív elemen állva csatlakoztatják a kezükben lévő csővéget az aktív elemhez.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> Az egyik szerelő felcsatlakoztatja a nála lévő csővéget arra az aktív elemre, amin áll. A másik szerelő is felcsatlakoztatja a nála lévő csővéget arra az aktív elemre, amin áll.

Use-case neve	12. Attach both ends of Pipe to same Active Element
Rövid leírás	A felhasználó egy szerelőt irányít, aki megpróbálja felcsatlakoztatni a kezében lévő csövet arra az aktív elemre, amire a cső másik vége van felcsatlakoztatva.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> A szerelő felcsatlakoztatja a nála lévő csővéget az aktív elemre, amin éppen áll, de ez nem sikerül, mert a cső másik vége is erre van csatlakoztatva.

Use-case neve	13. Pump port limit reached
Rövid leírás	A felhasználó egy szerelőt irányít, aki megpróbál felcsatlakoztatni egy kezében lévő csövet egy olyan pumpára, amin már nincs hely.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> A szerelő felcsatlakoztatja a nála lévő csővéget a pumpára, amin éppen áll, de ez nem sikerül, mert a pumpára már nem lehet több csövet csatlakoztatni.

Use-case neve	14. Acquire Pump
Rövid leírás	A felhasználó szerelőt irányít, aki egy ciszternán állva felvehet egy pumpát.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> A szerelő kér egy pumpát a ciszternától, amin áll. A ciszterna létrehoz egy új pumpát. A szerelő megkapja a létrehozott pumpát.

Use-case neve	15. Place Pump
Rövid leírás	A felhasználó egy szerelőt irányít, aki lehelyezi a nála lévő pumpát az adott csőre, amin áll.
Aktorok	User
Forgatókönyv	1. A csövön álló szerelő lehelyez egy pumpát.

Use-case neve	16. Unable to Place Pump
Rövid leírás	A felhasználó egy szerelőt irányít, aki megpróbál lehelyezni egy pumpát a csőre, amin áll. De ez nem sikerül mert nincsen nála pumpa.
Aktorok	User
Forgatókönyv	1. A csövön álló szerelő megpróbál lehelyezni egy pumpát. 2. Ez nem sikerül, mert nincs nála pumpa.

Use-case neve	17. Move to Pipe
Rövid leírás	A felhasználó egy tetszőleges fajta karaktert (jelen esetben szabotőrt) irányít, aki rálépne egy csőre egy tetszőleges fajta ActiveElementről (jelen esetben pumpáról).
Aktorok	User
Forgatókönyv	1. A szabotőr megpróbál rálépni a csőre. 2. A szabotőr lelép a pumpáról.
Alternatív forgatókönyv	1.A. A cső foglalt, ezért nem sikerült rálépnie. 2.A. Az 1.A. miatt nem lép le a pumpáról.

Use-case neve	18. Move to Detached Pipe
Rövid leírás	A felhasználó egy tetszőleges fajta karaktert (jelen esetben szabotőrt) irányít, aki rálépne egy lecsatlakoztatott végű csőre egy tetszőleges fajta ActiveElementről (jelen esetben pumpáról), de ez nem sikerül, mert a cső egyik vége le van csatlakoztatva.
Aktorok	User
Forgatókönyv	1. A szabotőr megpróbál rálépni a csőre. 2. A cső visszautasítja a kérést. 3. A szabotőr nem lép le a pumpáról.

Use-case neve	19. Move to Active Element
Rövid leírás	Az egyik felhasználó által irányított karakter átlép egy csőről egy aktív elemre.
Aktorok	User
Forgatókönyv	1. A karakter csőről aktív elemre lép.

Use-case neve	20. Break Pump
Rövid leírás	A felhasználó kérésére a controller el tud rontani pumpákat.
Aktorok	User
Forgatókönyv	1. A controller elront egy pumpát.

Use-case neve	21. Spawn Pipe
----------------------	----------------

Rövid leírás	A felhasználó kérésére a controller új csöveget tud a ciszternára létrehozni.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A ciszterna létrehoz egy új csövet. 2. A ciszterna csatlakoztatja magához a csövet.

Use-case neve	22. Waterflow from Spring
Rövid leírás	A felhasználó lépteti a forrást, ami a rácsatlakoztatott csőbe vizet folyat.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A forrás vizet folyat a rácsatlakoztatott csőbe.
Alternatív forgatókönyv	<ol style="list-style-type: none"> 1.A. A cső lyukas, ezért a bele folyatott víz a földre folyik.

Use-case neve	23. Waterflow through Pump
Rövid leírás	A felhasználó lépteti a pumpát, ami a kimenetként állított csőbe a tartályából vizet folyat, majd a bemenetként állított csőből a tartályába pumpál vizet.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A pumpa a kimeneti csövébe kifolyat a tartályából annyi vizet, amennyi a csőbe fér, vagy amennyi a tartályában van, amelyik kevesebb. 2. A pumpa a bemeneti csövéből a tartályába pumpál annyi vizet, amennyi belefér, vagy amennyi a csőben van, amelyik kevesebb.
Alternatív forgatókönyv	<ol style="list-style-type: none"> 1.A. A pumpa el van romolva, ezért nem mozgat vizet. 1.B. A kimenetként beállított cső lyukas, ezért a csőbe pumpált víz a földre folyik. 2.A. A pumpa el van romolva, ezért nem mozgat vizet.

Use-case neve	24. Waterflow into Cistern
Rövid leírás	A felhasználó lépteti a ciszternát, ami a rácsatlakoztatott csőből kiszedi az összes vizet.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A ciszterna kiszedi az összes vizet a csőből. 2. A ciszterna eltárolja a kiszedett vízmennyiséget a vízgyűjtőjében.

Use-case neve	25. Waterflow into Detached Pipe
Rövid leírás	A felhasználó lépett egy forrást, ami vizet rak egy lecsatlakoztatott végű csőbe. (A use case hasonlóan működne, ha egy pumpa raktára a vizet a csőbe).
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A forrás valamennyi vizet rak a csőbe. 2. A cső az összes vizet belerakja a föld vízgyűjtőbe. 3. A csőből minden víz kifolyt, ezért a visszaadott túlfolyási vízmennyiség 0.

5.2 A szkeleton kezelői felületének terve, dialógusok

A program megjelenítése konzol ablakban történik az egyszerűség és átláthatóság érdekében.

5.2.1 Főmenü

A program indításakor kiír egy kis köszöntő szöveget, majd felsorolja a lefuttatható teszteseteket. (Ez a lista megegyezik a Use-case-ek listájával!) (Minden teszteset elő egy space jel kerül.)

Példa:

1. Repair Pipe
2. Repair Pump

...

A felhasználónak ki kell választania melyik tesztet akarja futtatni, a konzol ablakba “>” jel mögé beírva, majd a kiválasztott teszteset lefuttatásra kerül.

5.2.2 Teszteset futása

Az első sorba a teszt száma, majd a neve kerül.

Ezt követően az adott teszt inicializáló szekvenciája kerül kiírásra (5.4), majd a teszt futási szekvenciája.

Minden alkalommal amikor egy függvény meghívásra kerül, a program kiír egy “->” jelet, az objektum nevét, amin a függvény hívódott, a meghívott függvény nevét és zárójelek között az átadott értékek típusát és értékét, és a függvény visszatérési típusát. Visszatéréskor pedig egy “<-” jelet ír ki, ha van visszatérési érték, akkor azt mögé helyezi. Ha egy objektum keletkezik azt “Create *objektum név*: *objektum típus*” mintára adjuk meg.

Minden sorban a hívás mélységenek megfelelően 4 darab space kerül lehelyezésre, minden mélységi szint után.

Példa:

3. Pierce Pipe:

Initializing test:

```
-> Create pump1: Pump
<-
-> Create ground: WaterCollector
<-
-> Create pipe: Pipe
<-
-> Create pump2: Pump
<-
-> pump1.addPipe(pipe)
-> pipe.addEndpoint(pump1)
<-
<-
-> pump2.addPipe(pipe)
-> pipe.addEndpoint(pump2)
<-
<-
```

```
-> Create saboteur: Saboteur
<-
-> saboteur.moveTo(pipe)
    -> pipe.enter(null)
    <-
<-
Sequence of test:
-> saboteur.pierce()
    -> pipe.pierce()
        -> ground.storeAmount(amount: int)
        <-
    <-
<-
példa vége
```

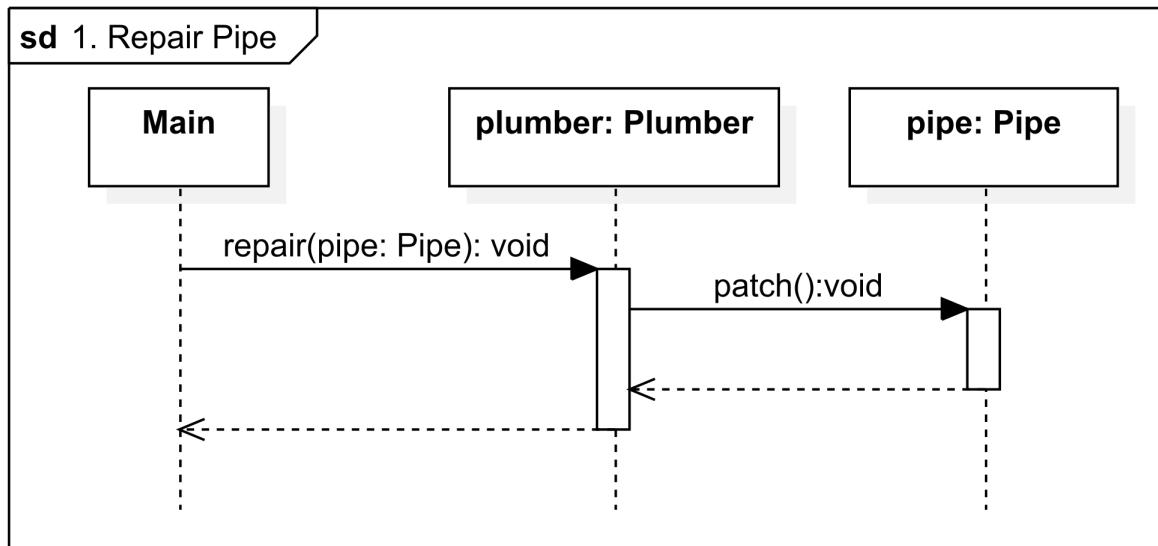
Amikor egy adott funkció lefutásának egy elem belső állapotára lesz szüksége, a kód azt a felhasználótól kéri be a szekvencia futása közben a konzol ablak által, majd a program a felhasználó által felvett érték szerint folytatódik.

Helyes működés esetén a lefutás közben megjelenő szöveg összevethető a dokumentumban található kommunikációs- és szekvenciadiagrammokkal. A működés helyességét e kettőnek megegyezése mutatja.

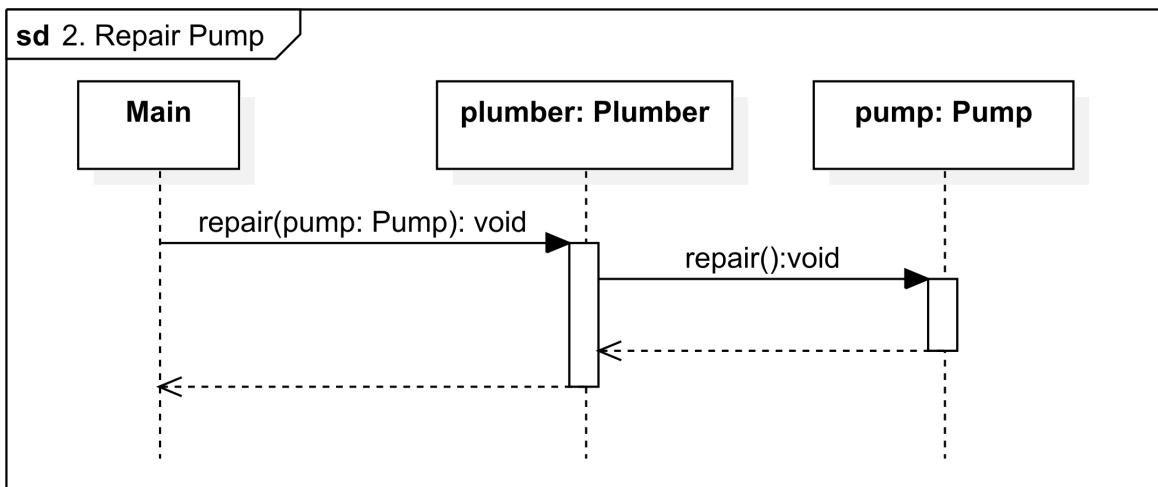
A teszt lefutását követően a program újra felajánlja a teszteset listát és az újabb teszteset indításának lehetőségét.

5.3 Szekvencia diagramok a belső működésre

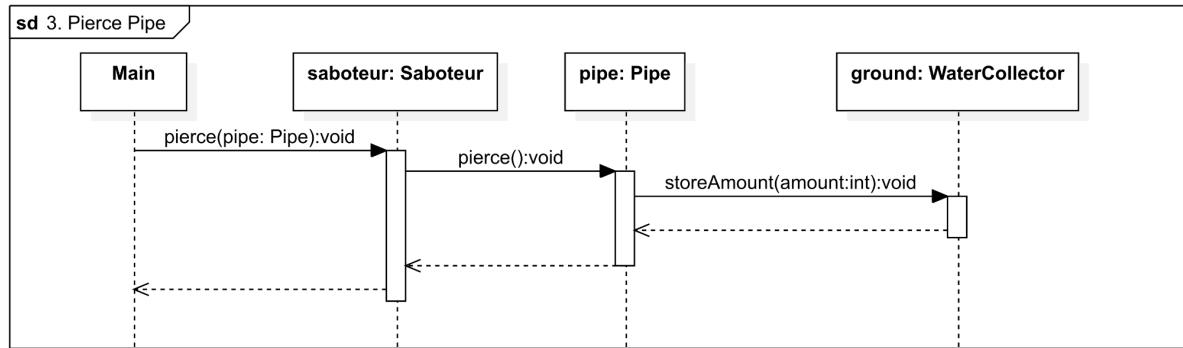
5.3.1 Repair Pipe



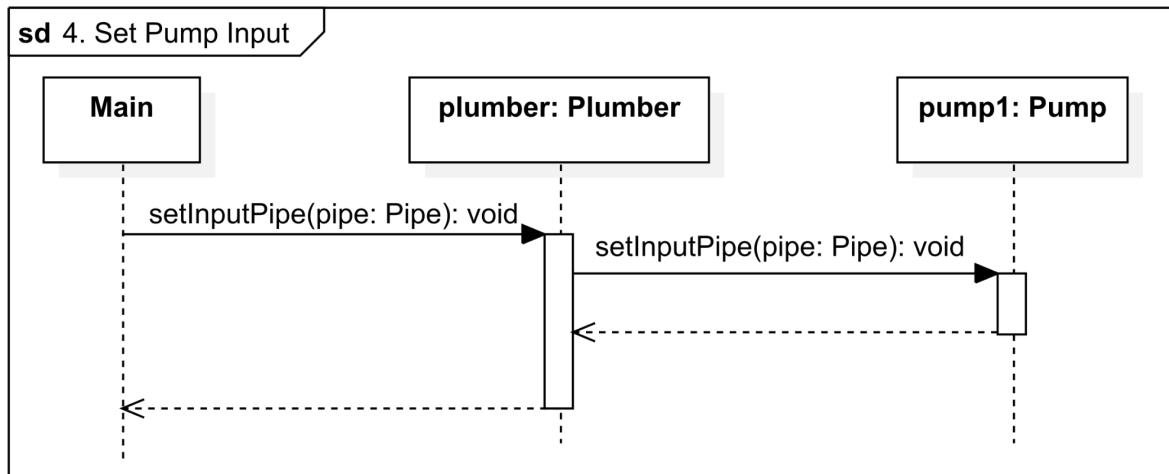
5.3.2 Repair Pump



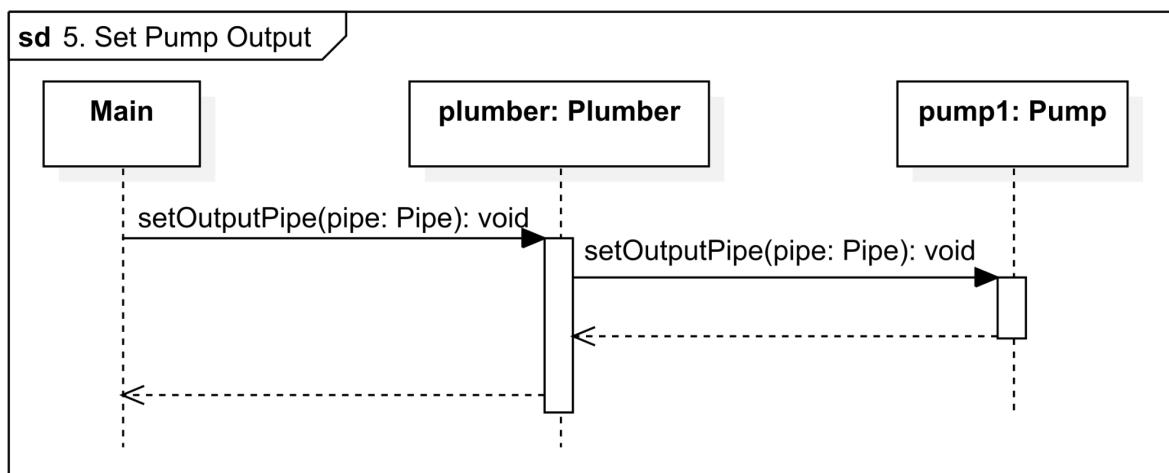
5.3.3 Pierce Pipe



5.3.4 Set Pump Input



5.3.5 Set Pump Output



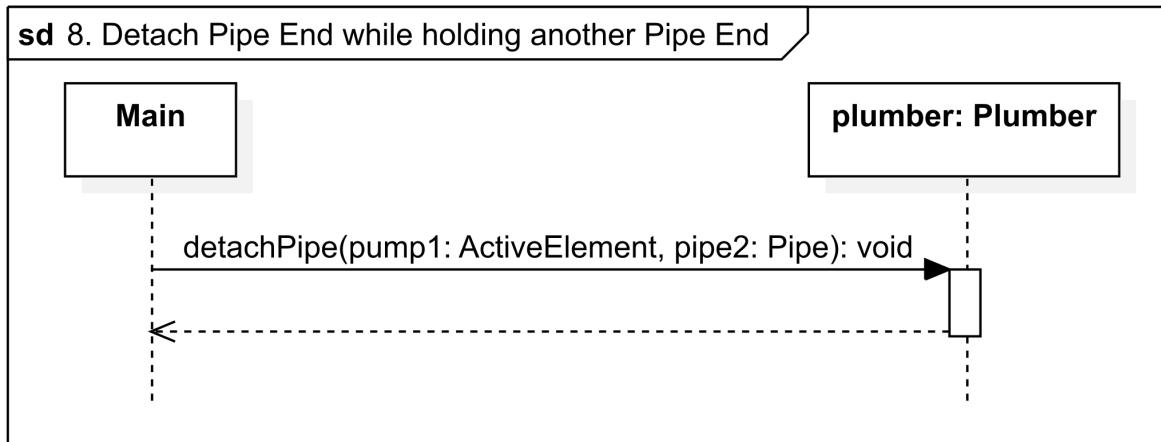
5.3.6 Unable to Set Pump Input

lásd: 5.3.4 Set Pump Input

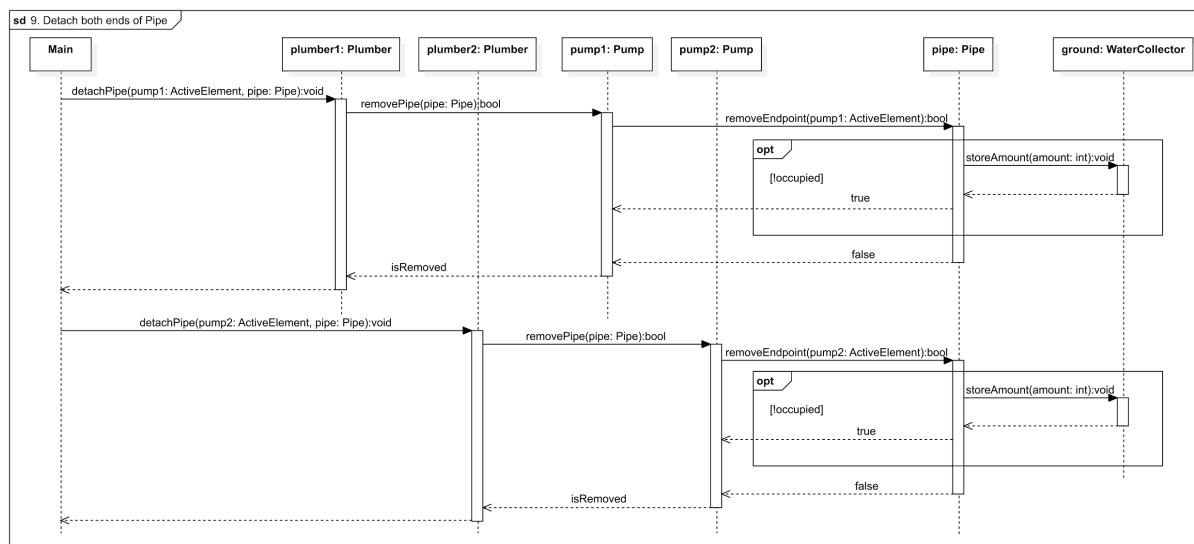
5.3.7 Unable to Set Pump Output

lásd: 5.3.5 Set Pump Output

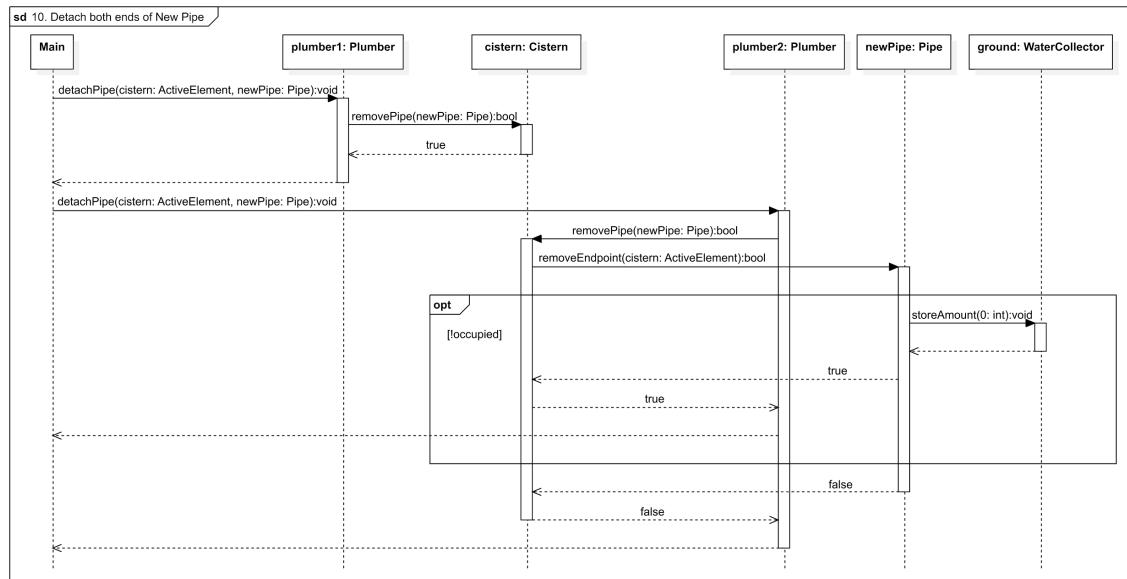
5.3.8 Detach Pipe End while holding another Pipe End



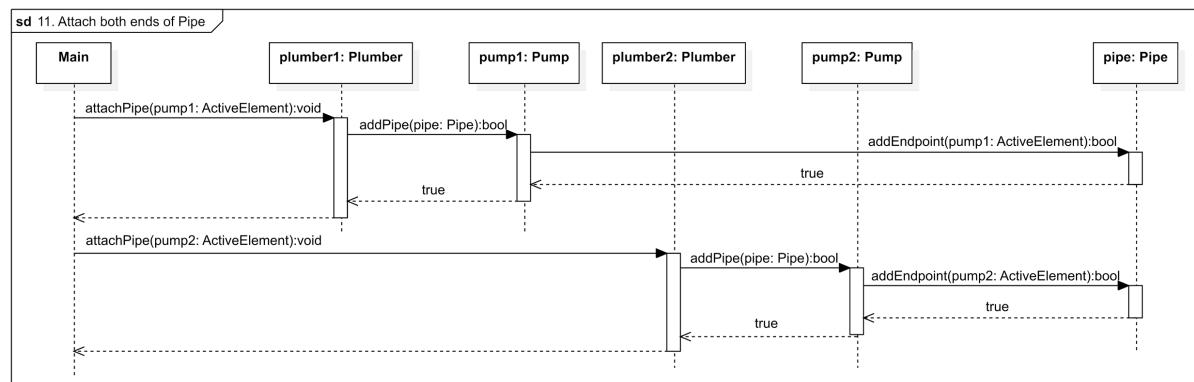
5.3.9 Detach Both ends of Pipe



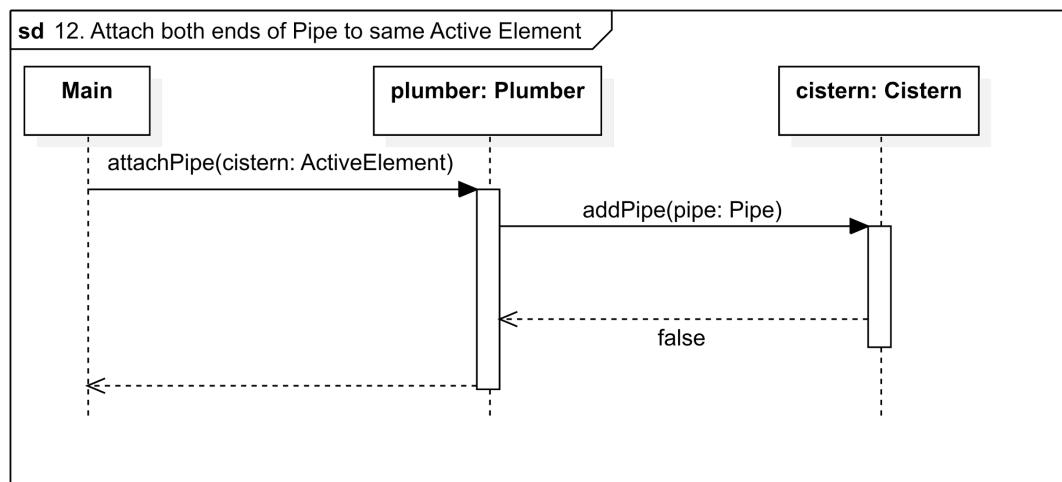
5.3.10 Detach both ends of New Pipe



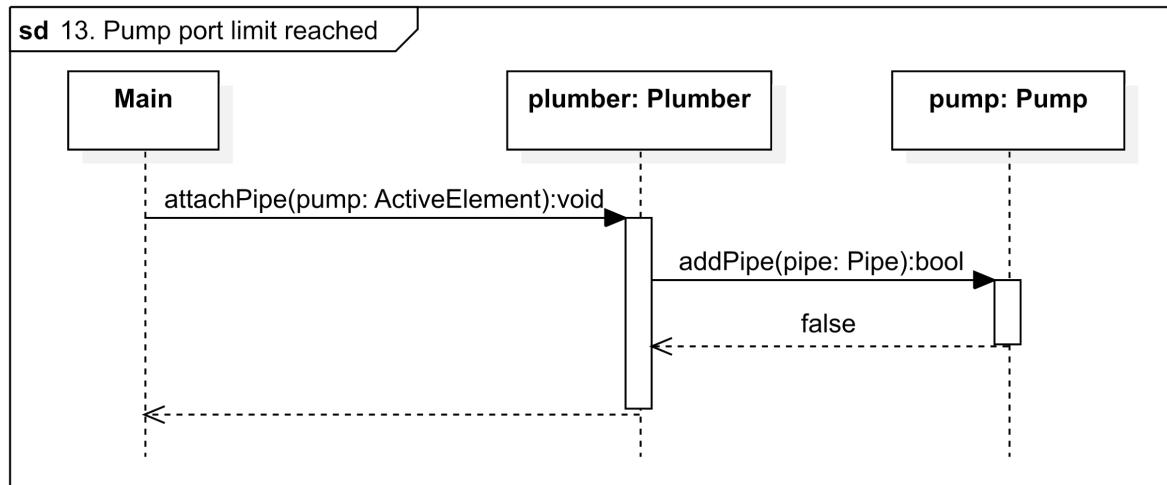
5.3.11 Attach both ends of Pipe



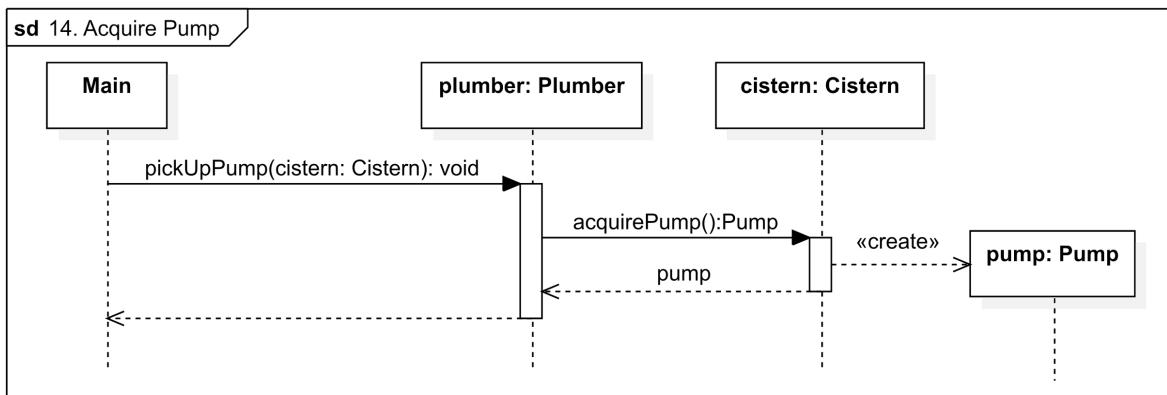
5.3.12 Attach both ends of Pipe to same Active Element



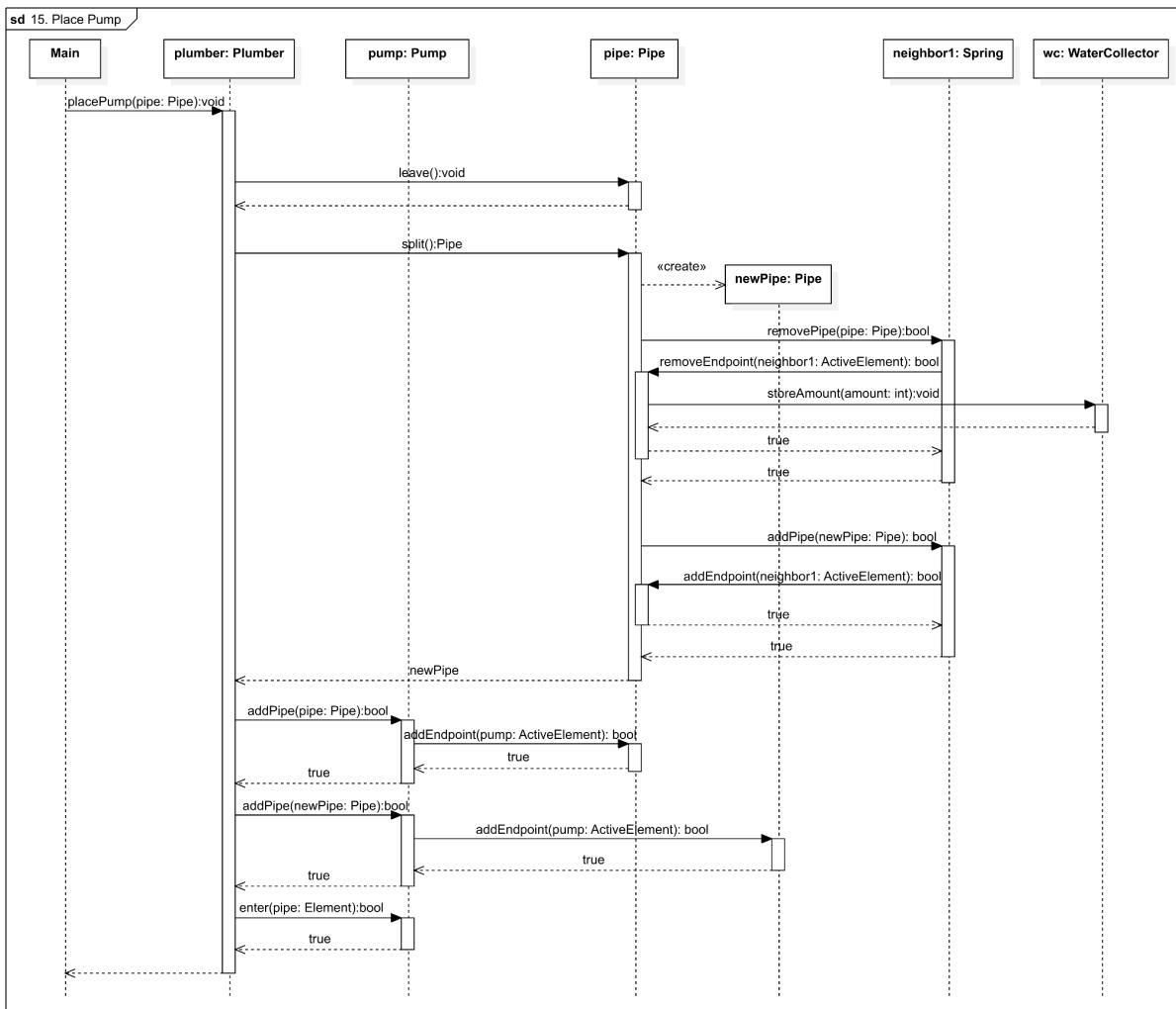
5.3.13 Pump port limit reached



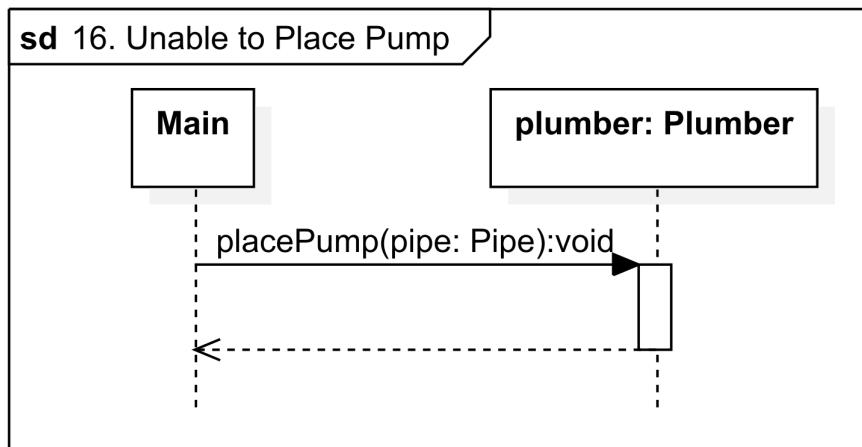
5.3.14 Acquire Pump



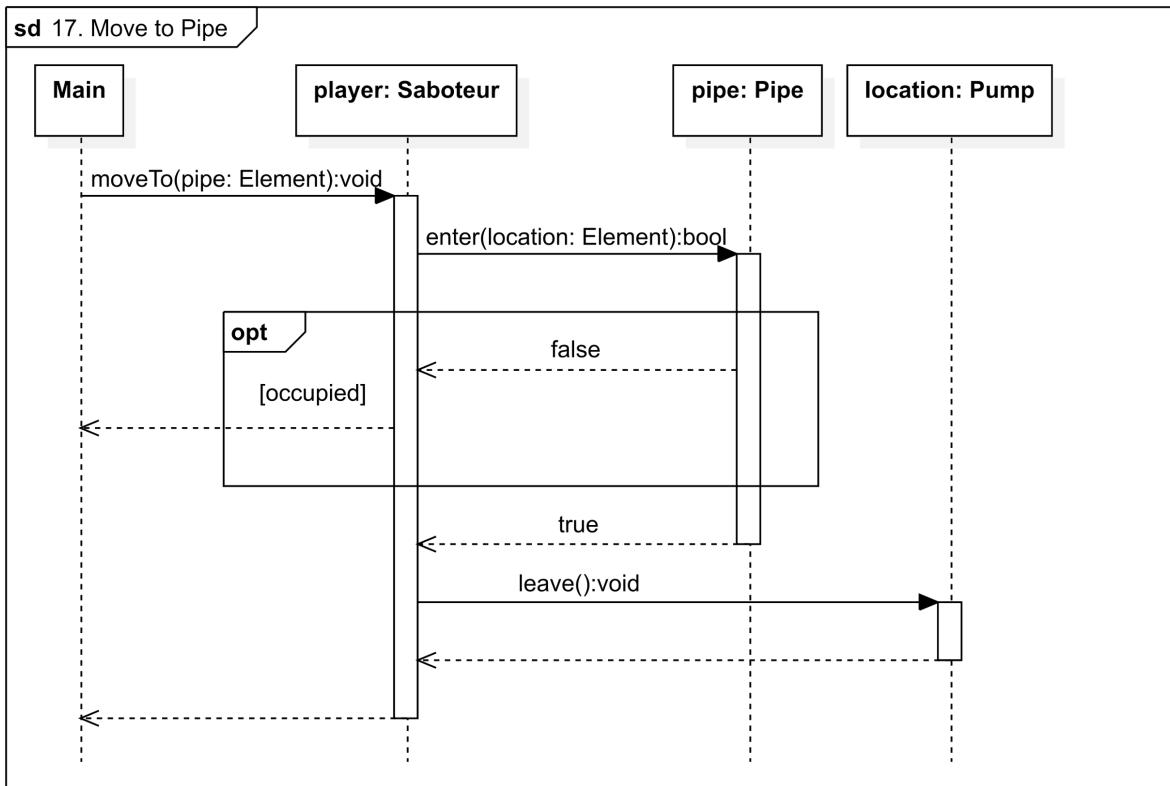
5.3.15 Place Pump



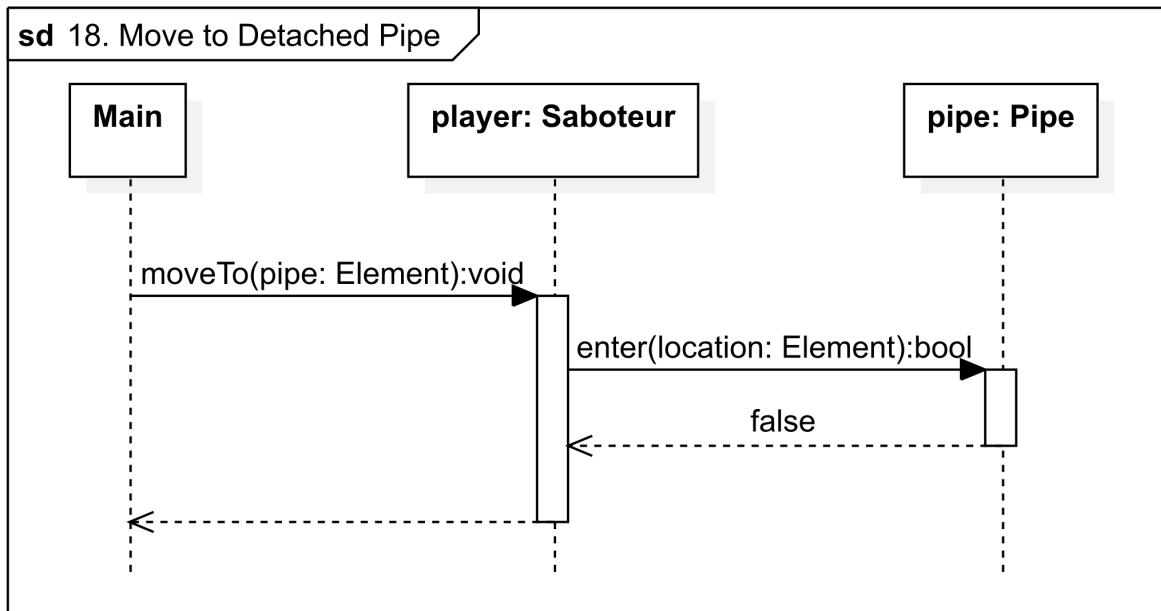
5.3.16 Unable to Place Pump



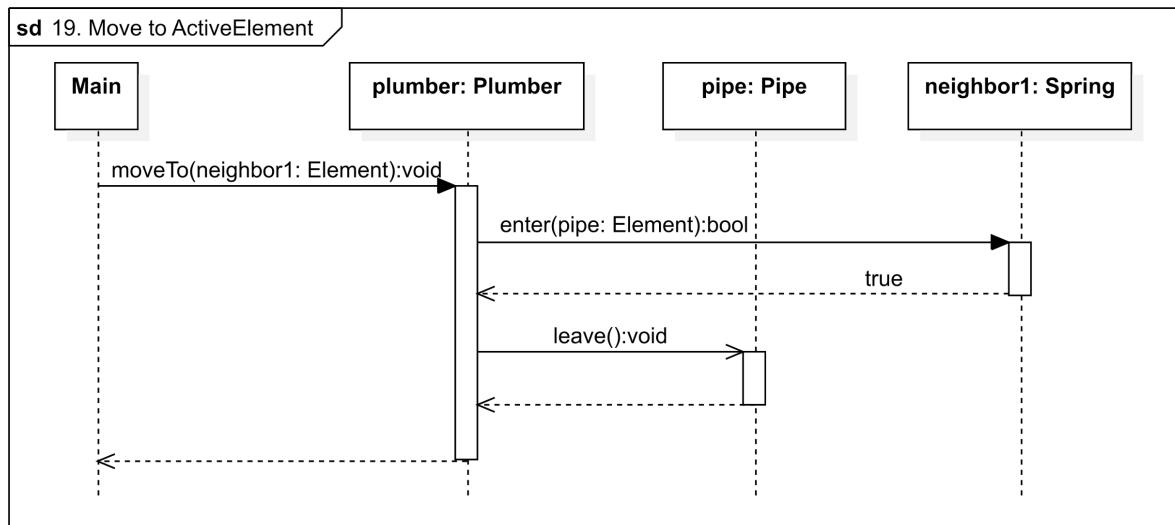
5.3.17 Move to Pipe



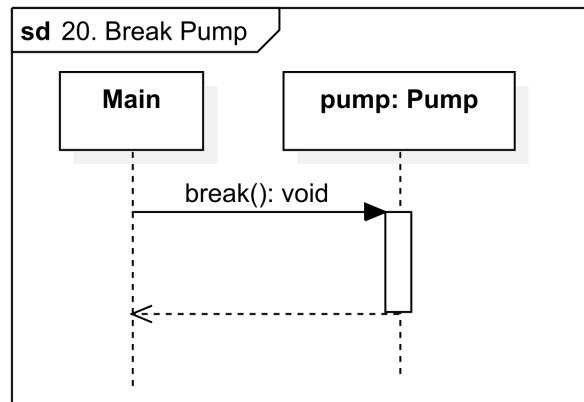
5.3.18 Move to Detached Pipe



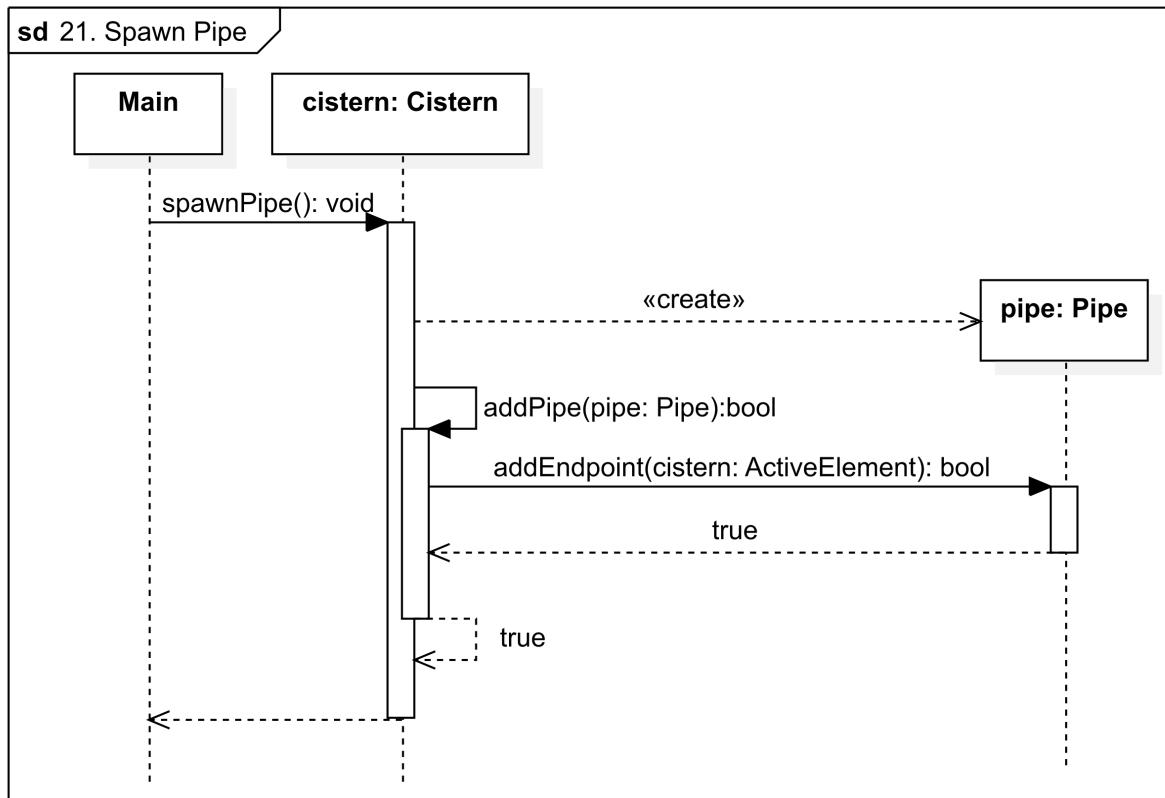
5.3.19 Move to Active Element



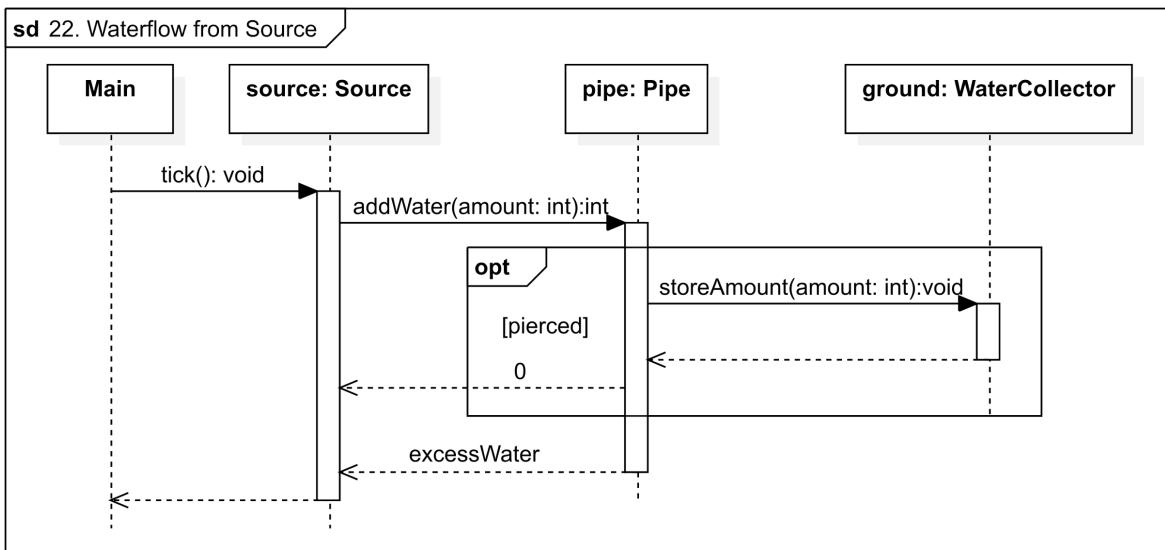
5.3.20 Break Pump



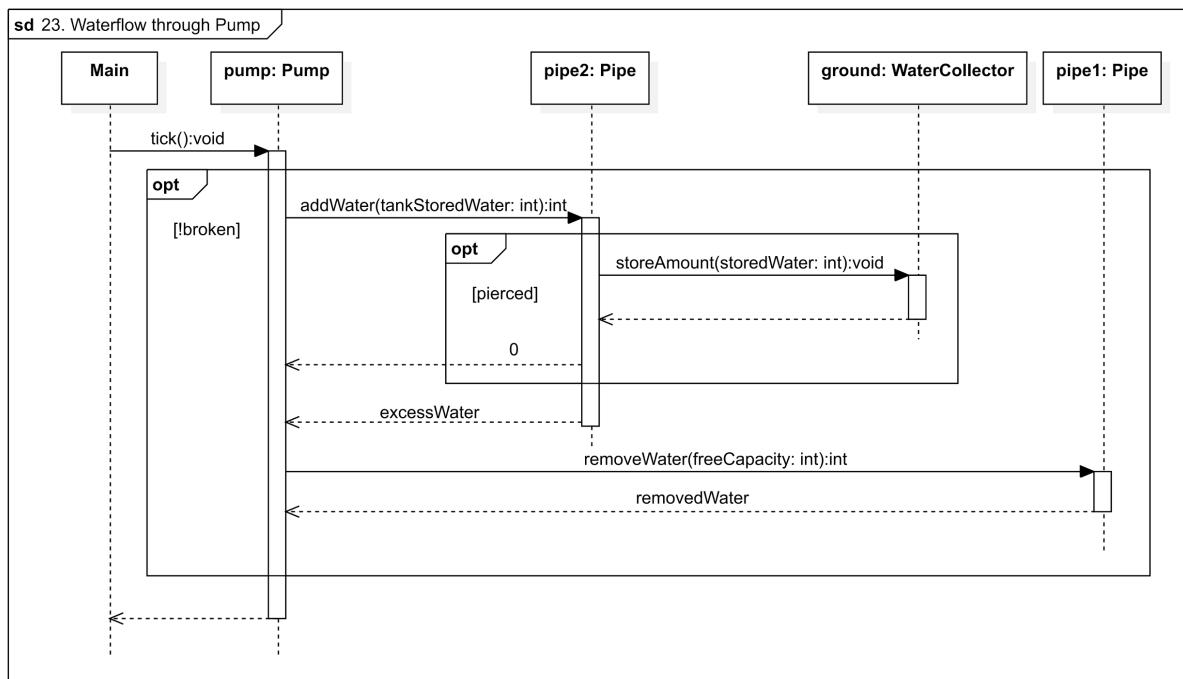
5.3.21 *Spawn Pipe*



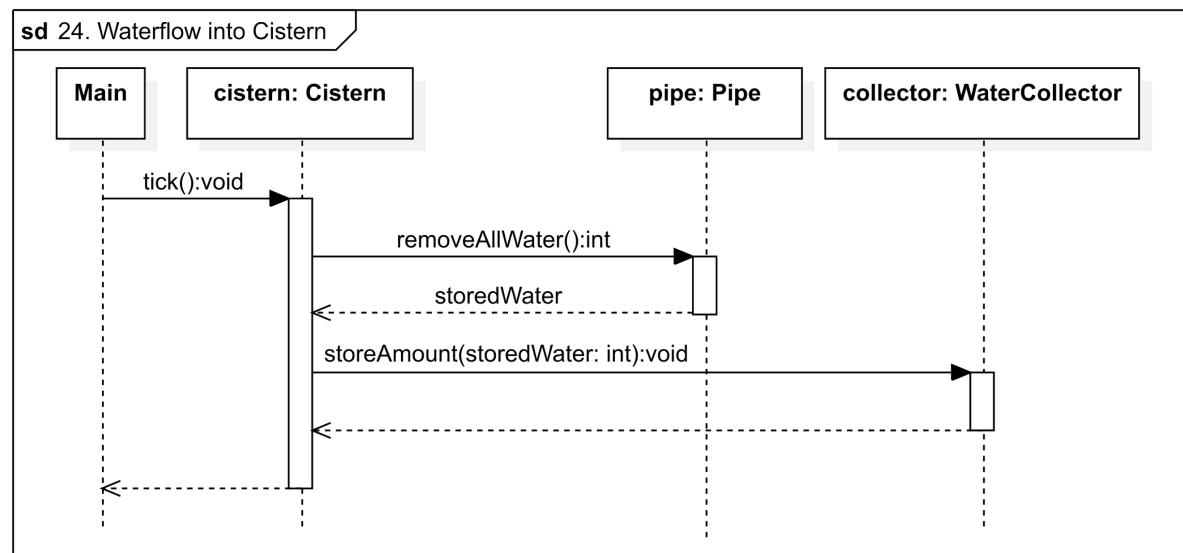
5.3.22 *Waterflow from Spring*

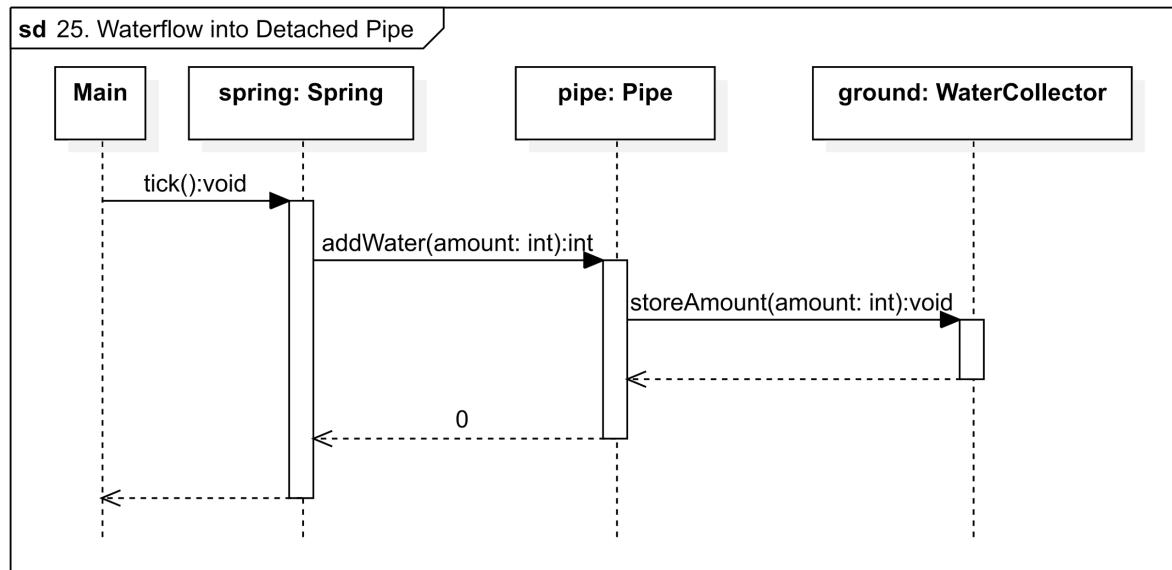


5.3.23 Waterflow through Pump



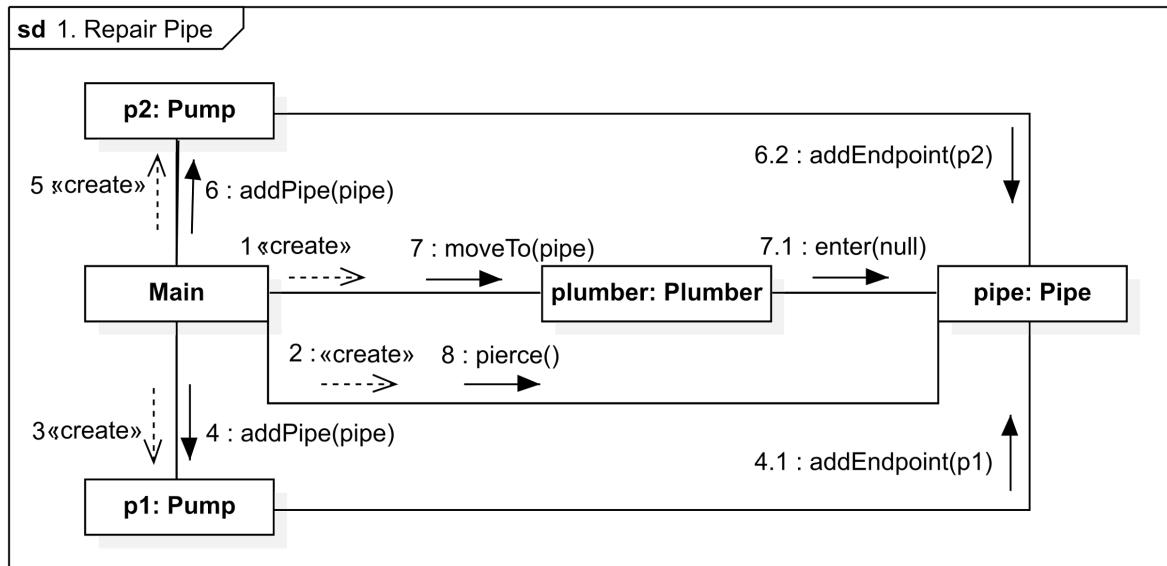
5.3.24 Waterflow into Cistern



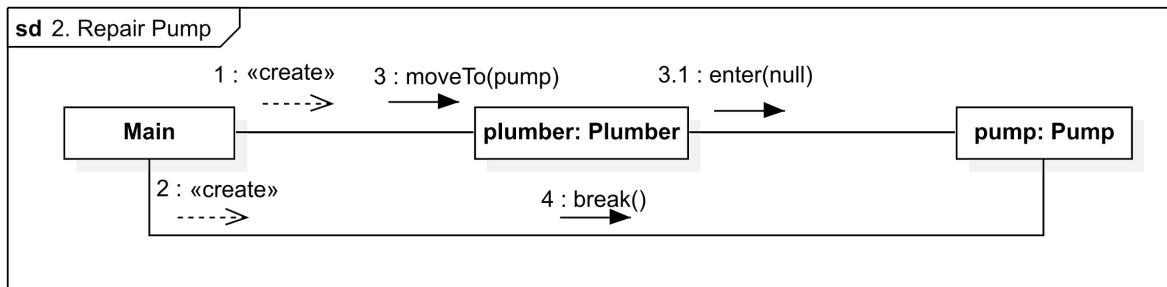
5.3.25 Waterflow into Detached Pipe

5.4 Kommunikációs diagramok

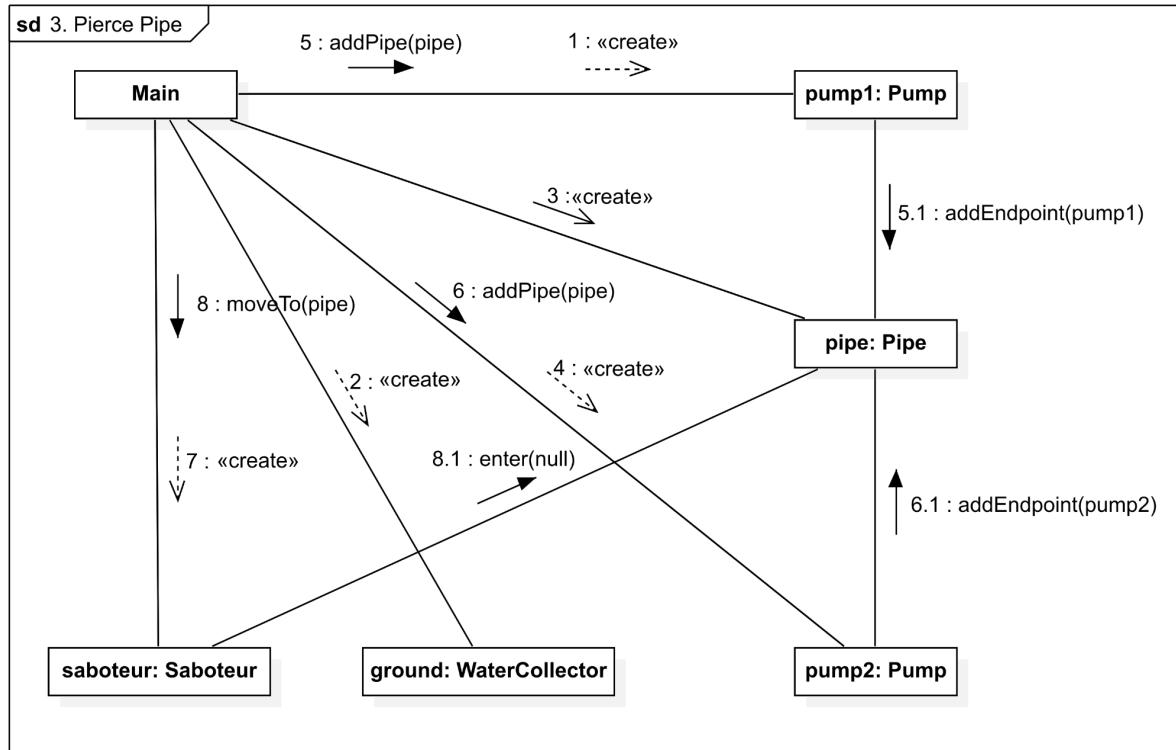
5.4.1 Repair Pipe



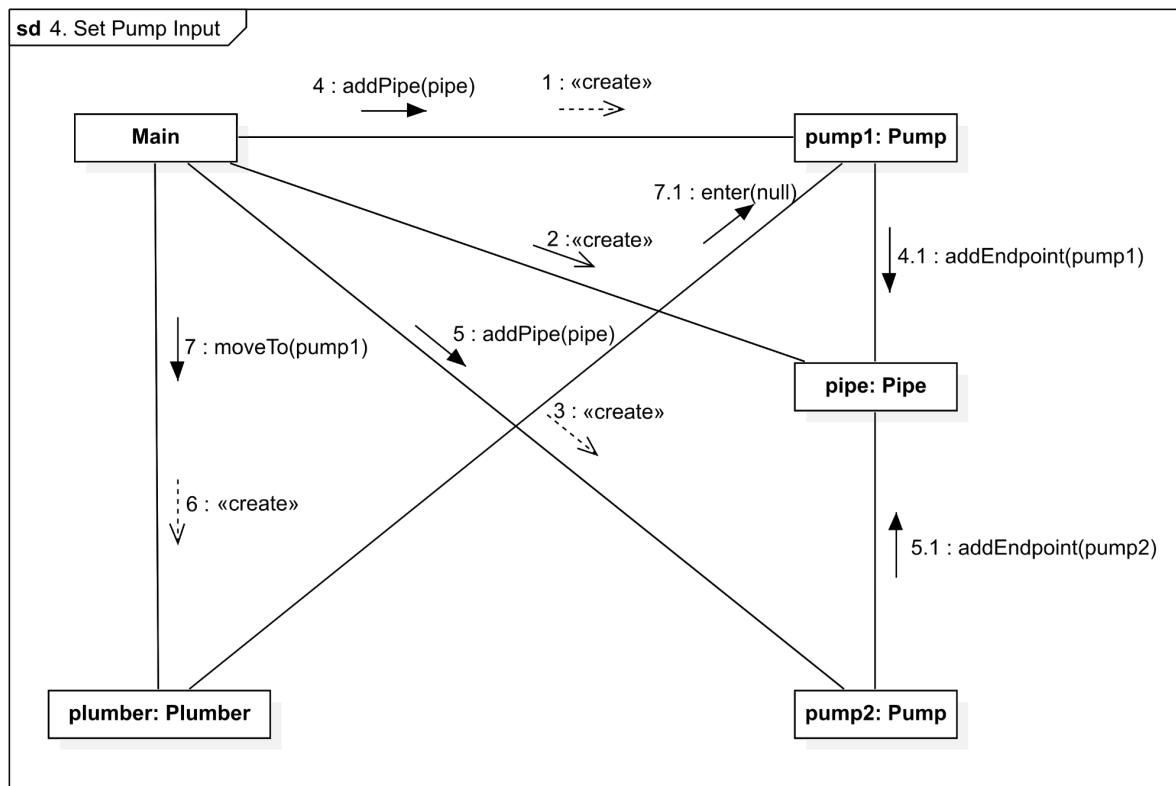
5.4.2 Repair Pump



5.4.3 Pierce Pipe



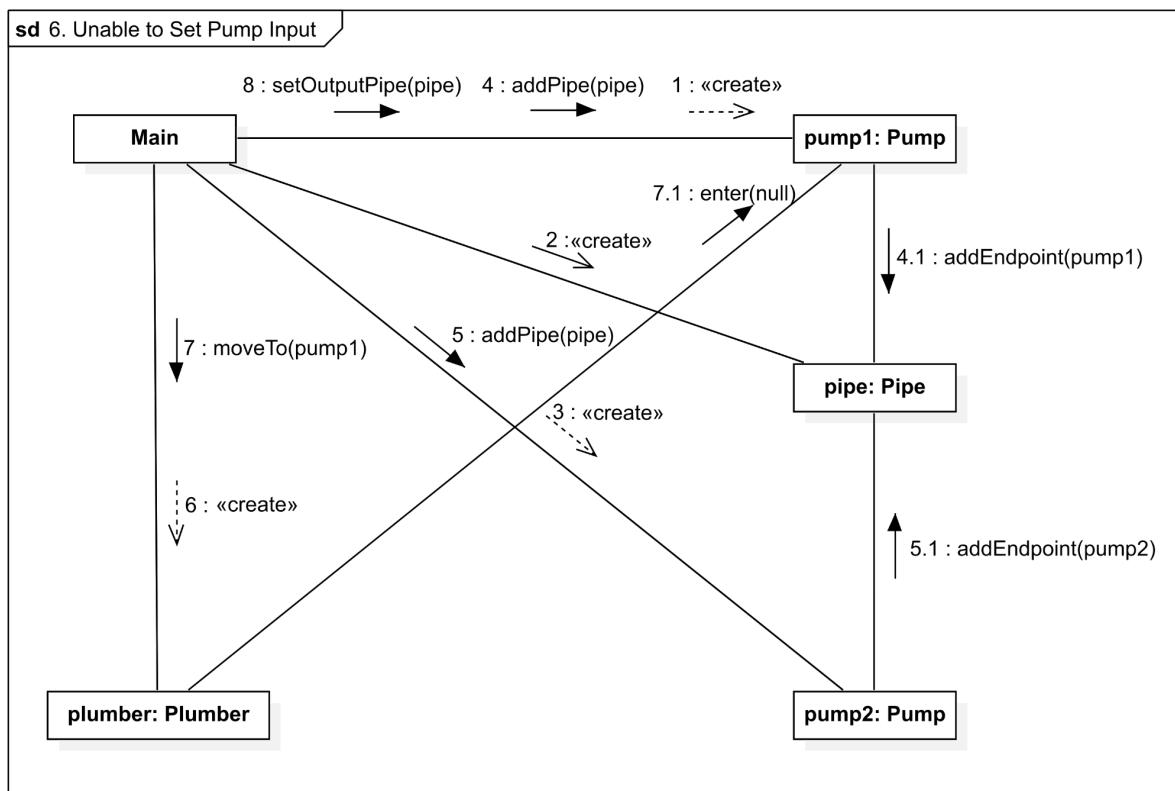
5.4.4 Set Pump Input



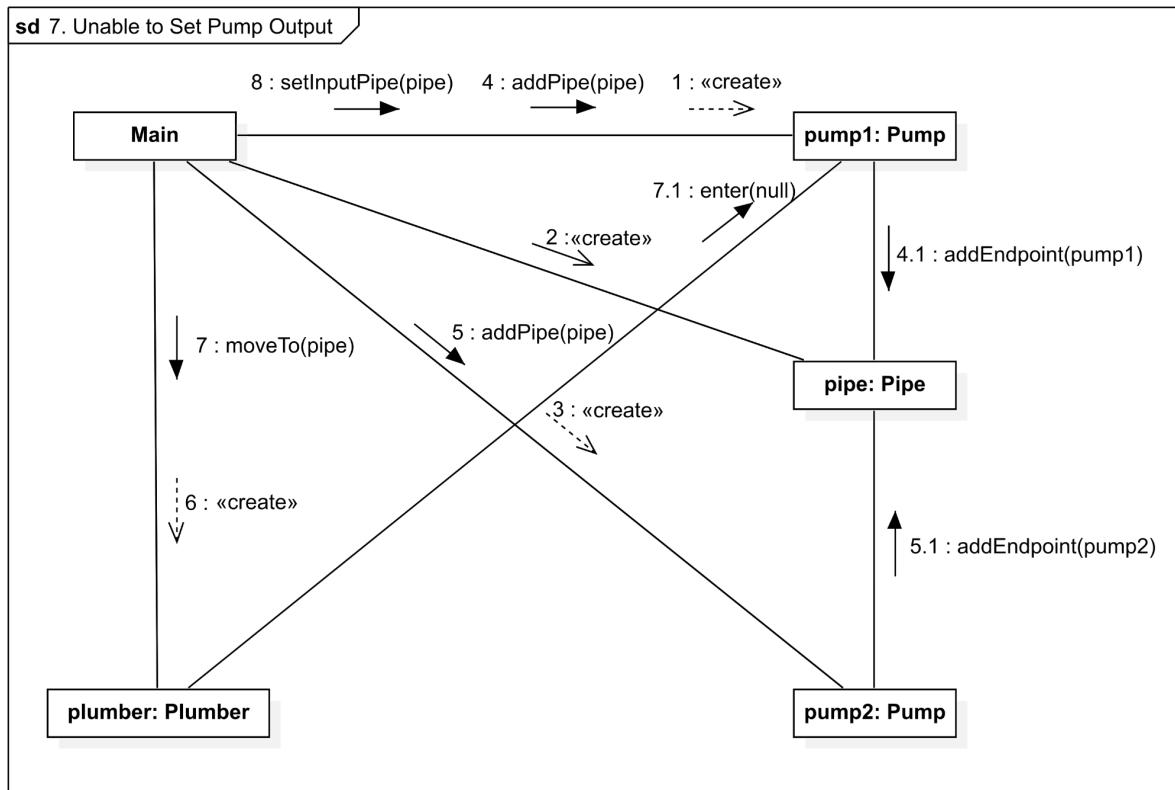
5.4.5 Set Pump Output

lásd: 5.4.4. Set Pump Input

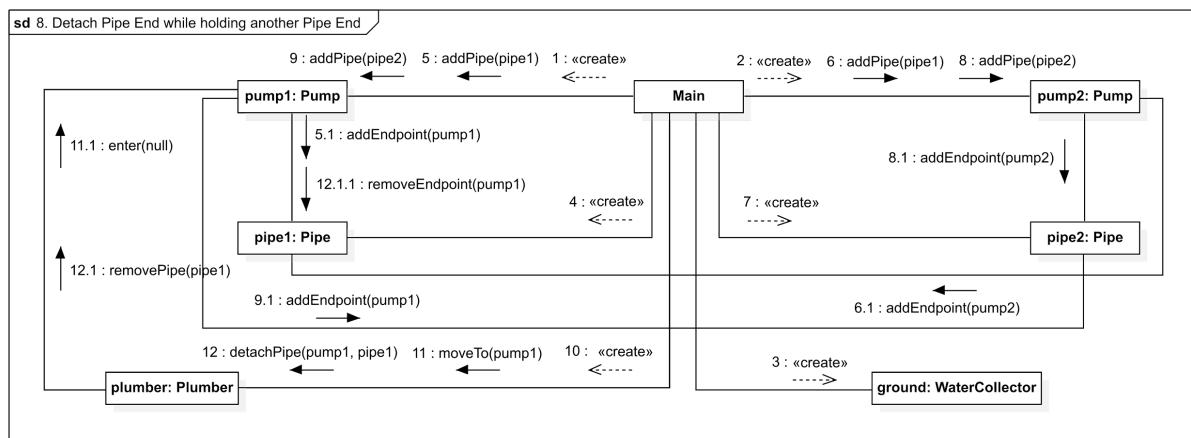
5.4.6 Unable to Set Pump Input



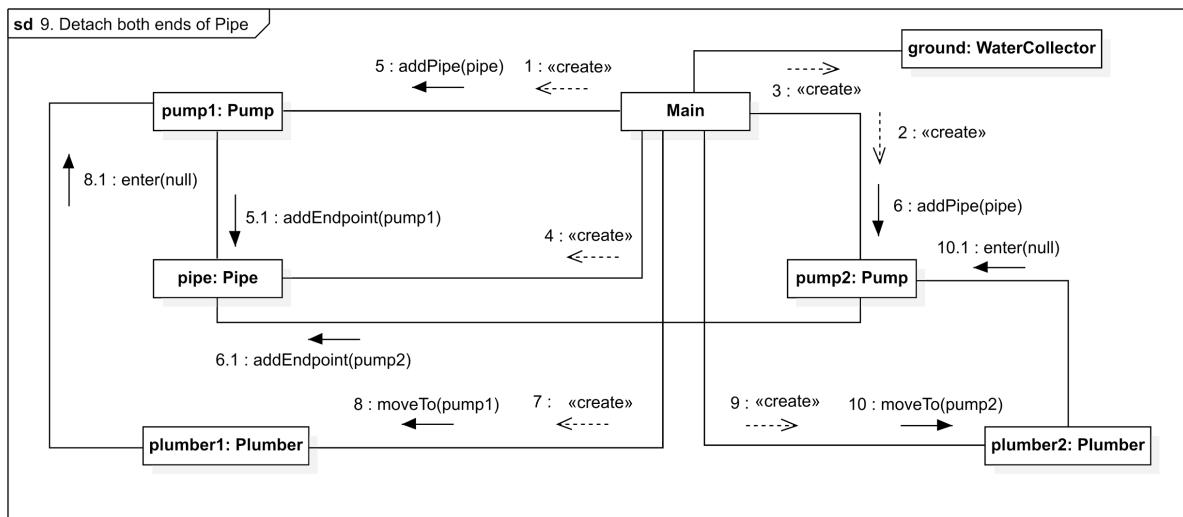
5.4.7 Unable to Set Pump Output



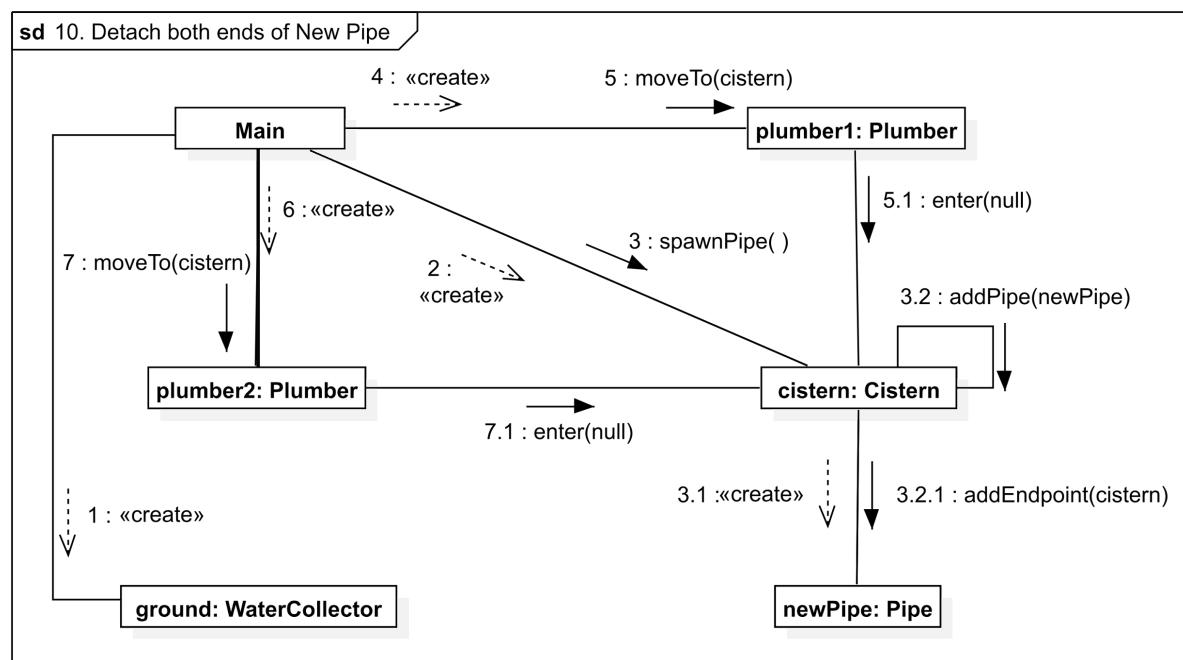
5.4.8 Detach Pipe End while holding another Pipe End



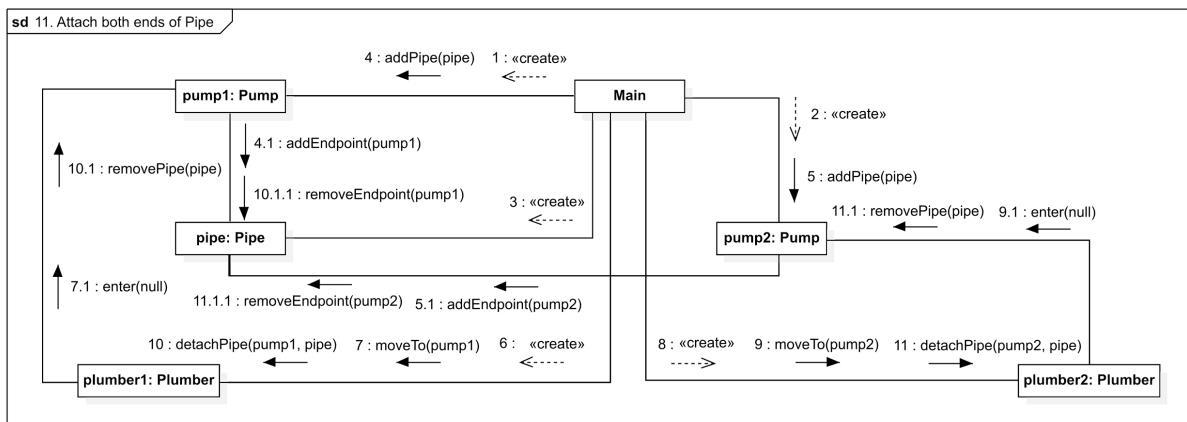
5.4.9 Detach both ends of Pipe



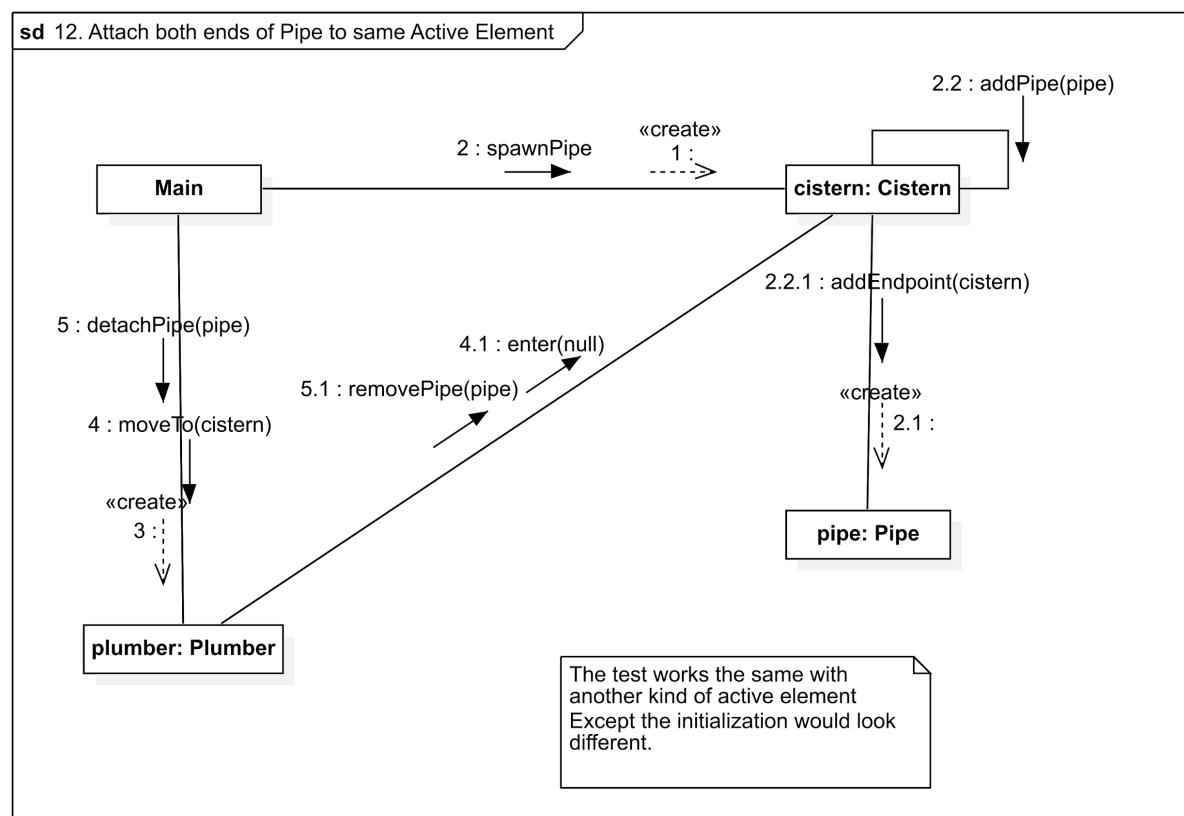
5.4.10 Detach both ends of New Pipe



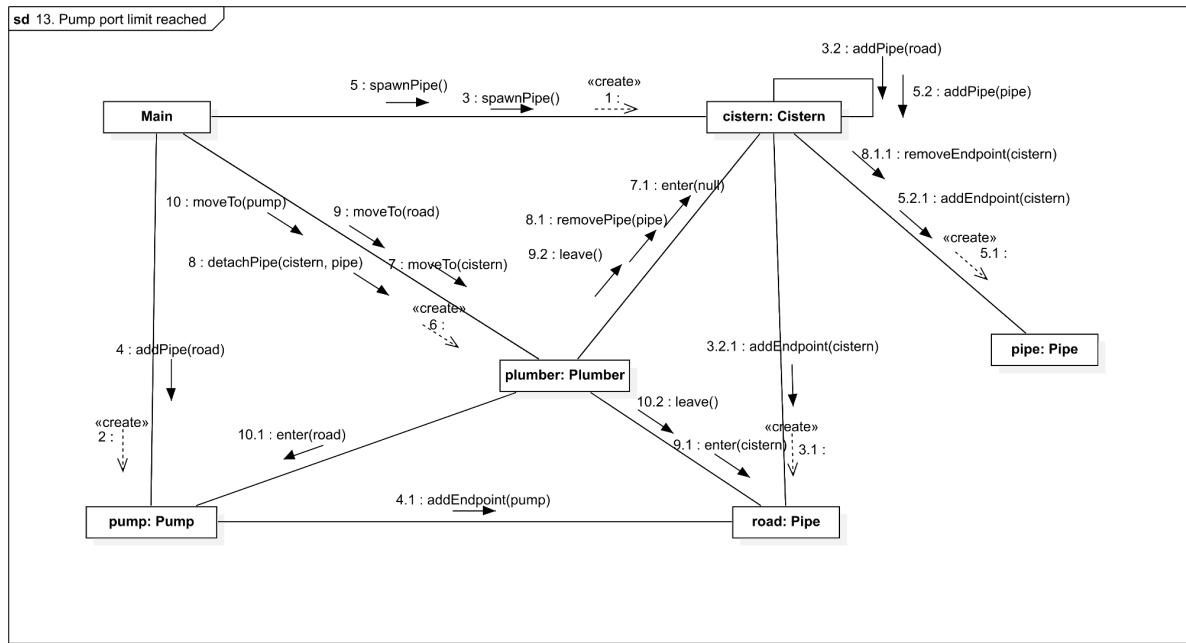
5.4.11 Attach both ends of Pipe



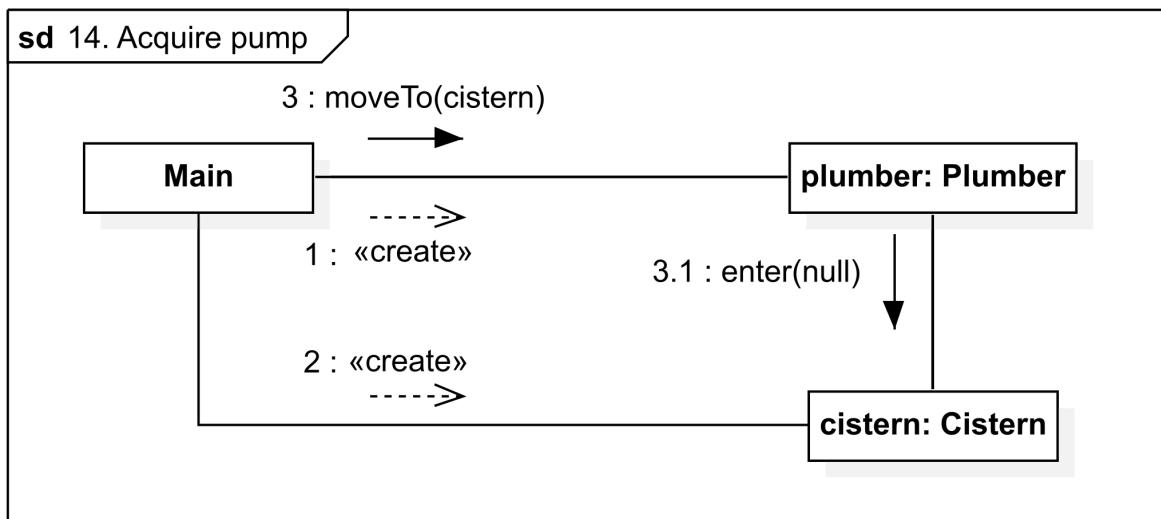
5.4.12 Attach both ends of pipe to same Active Element



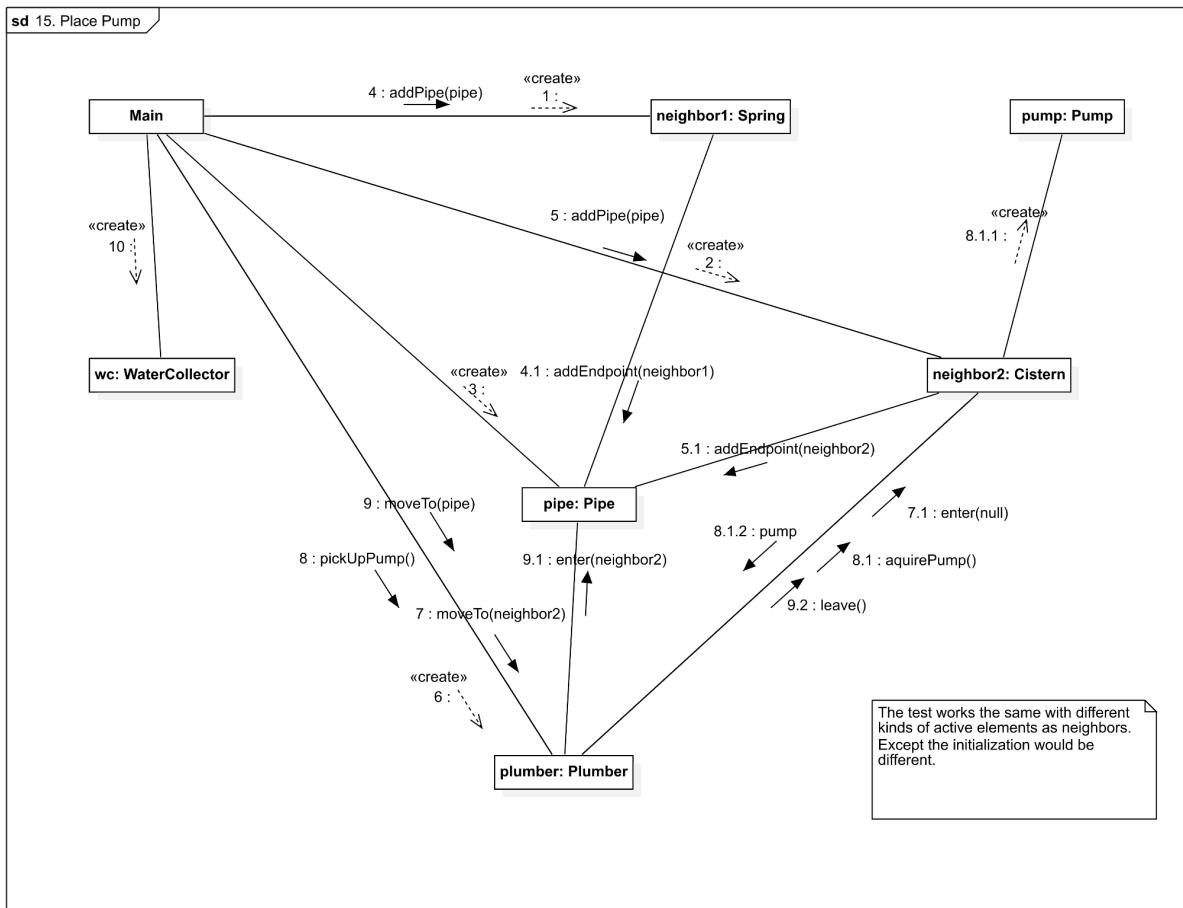
5.4.13 Pump port limit reached



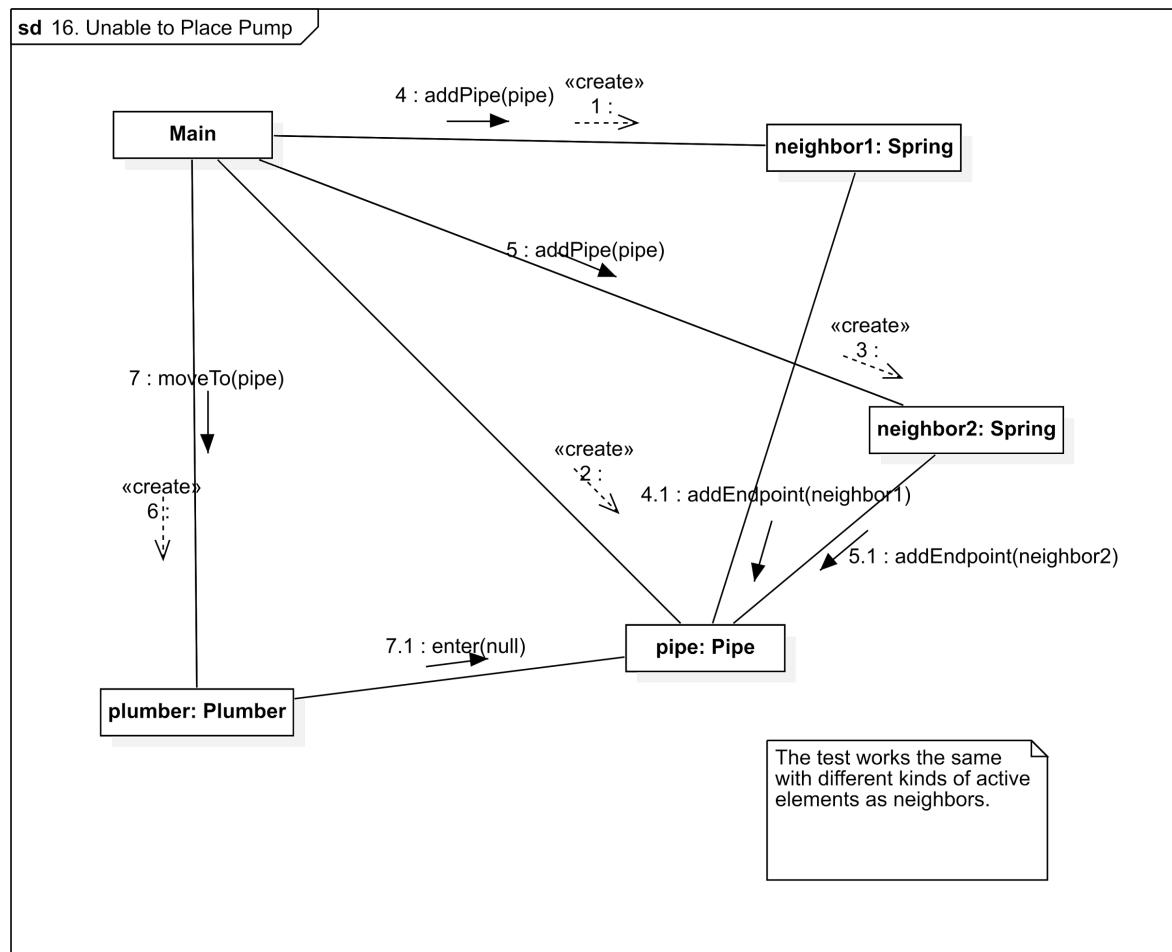
5.4.14 Acquire pump



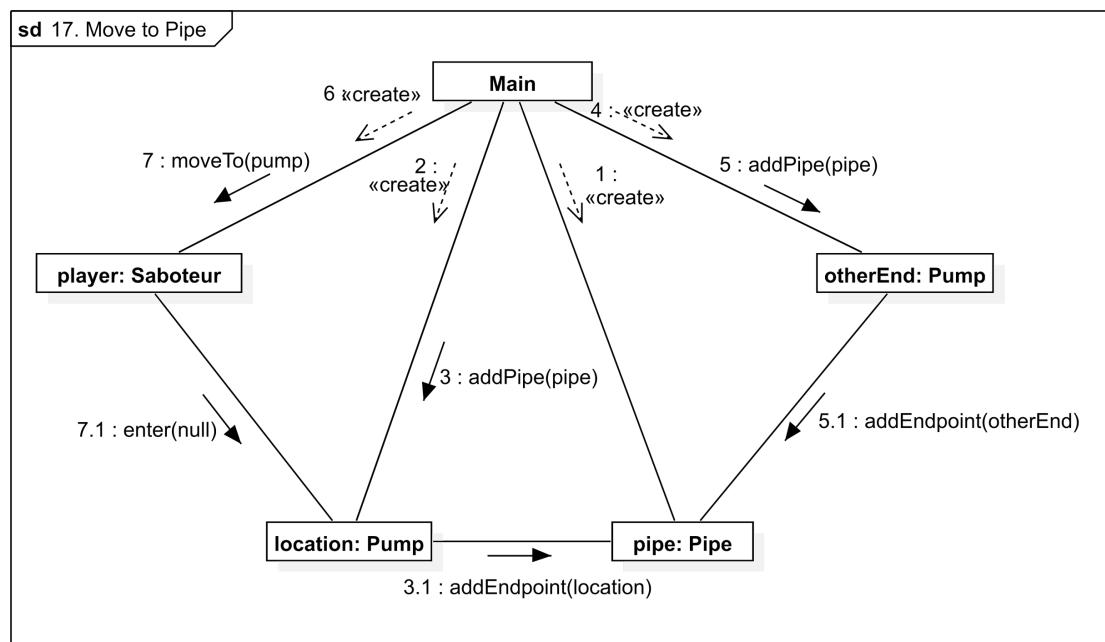
5.4.15 Place Pump



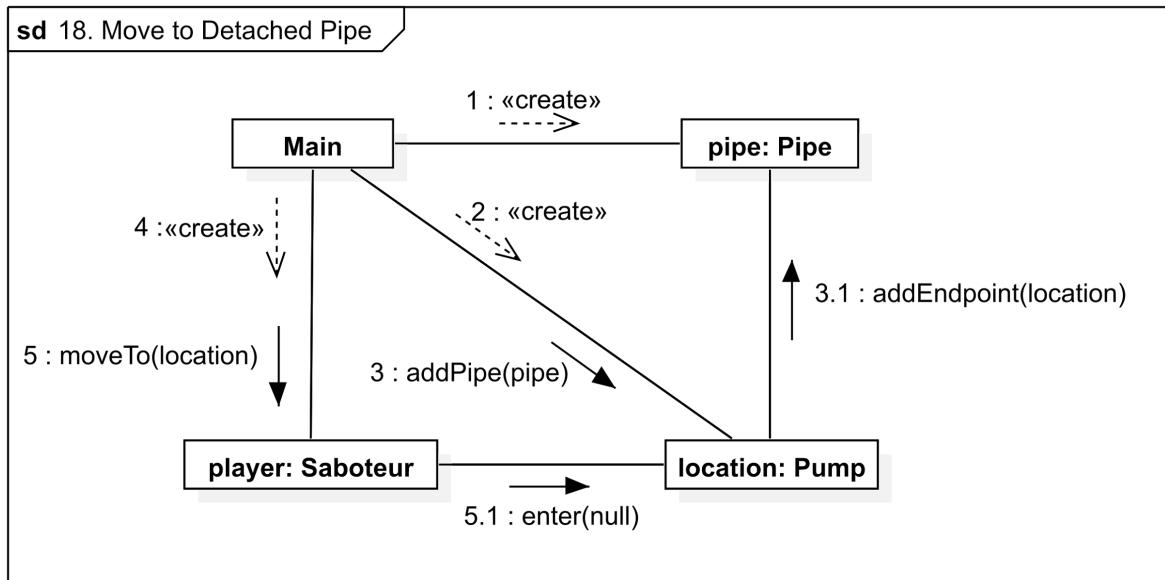
5.4.16 Unable to Place Pump



5.4.17 Move to Pipe



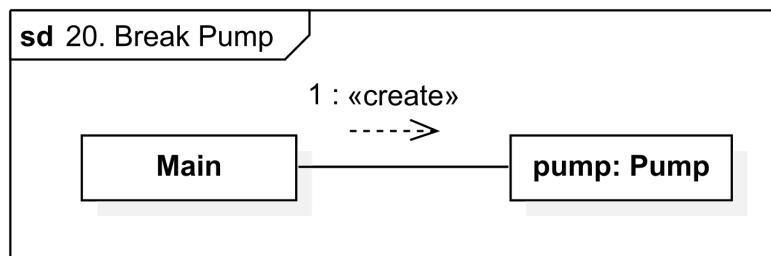
5.4.18 *Move to Detached Pipe*



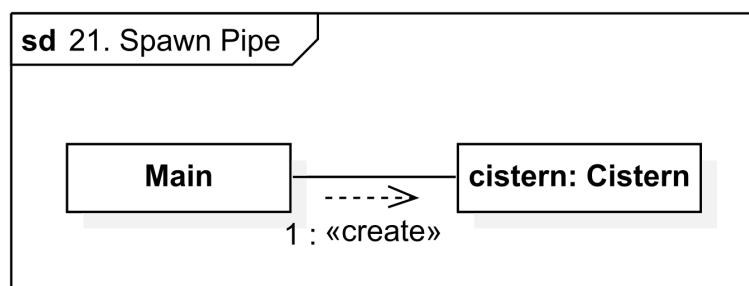
5.4.19 *Move to Active Element*

lásd: 5.4.16. Unable to Place Pump

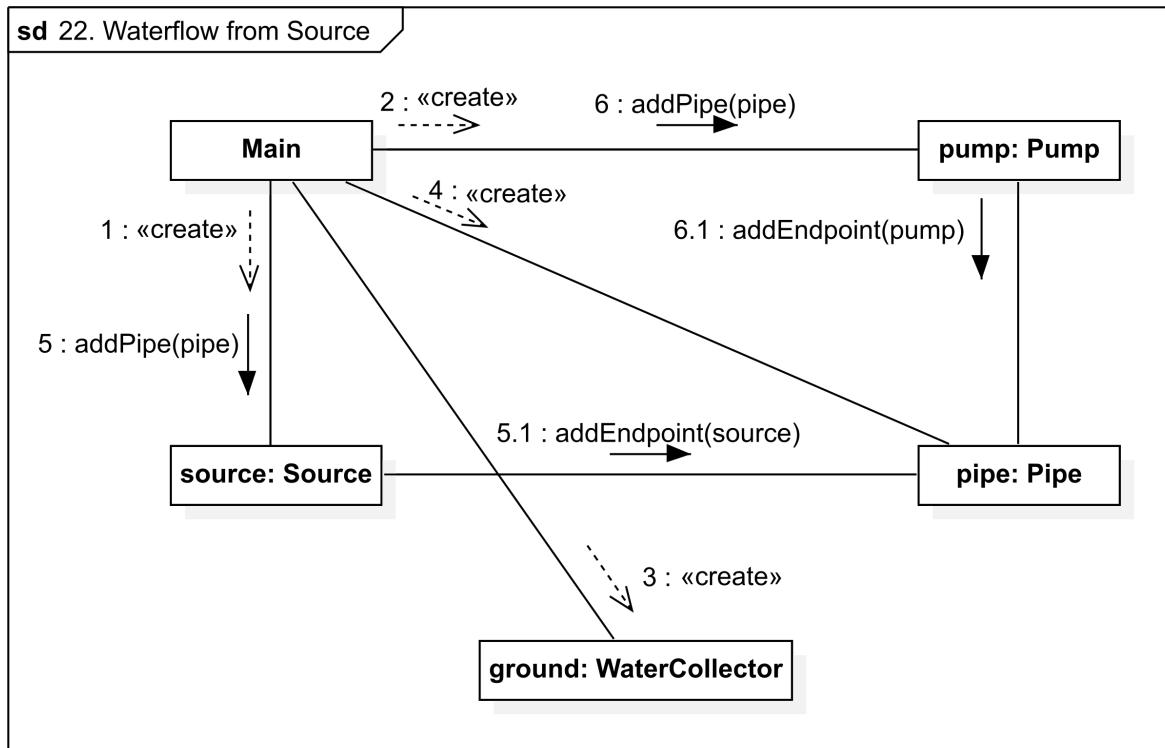
5.4.20 *Break Pump*



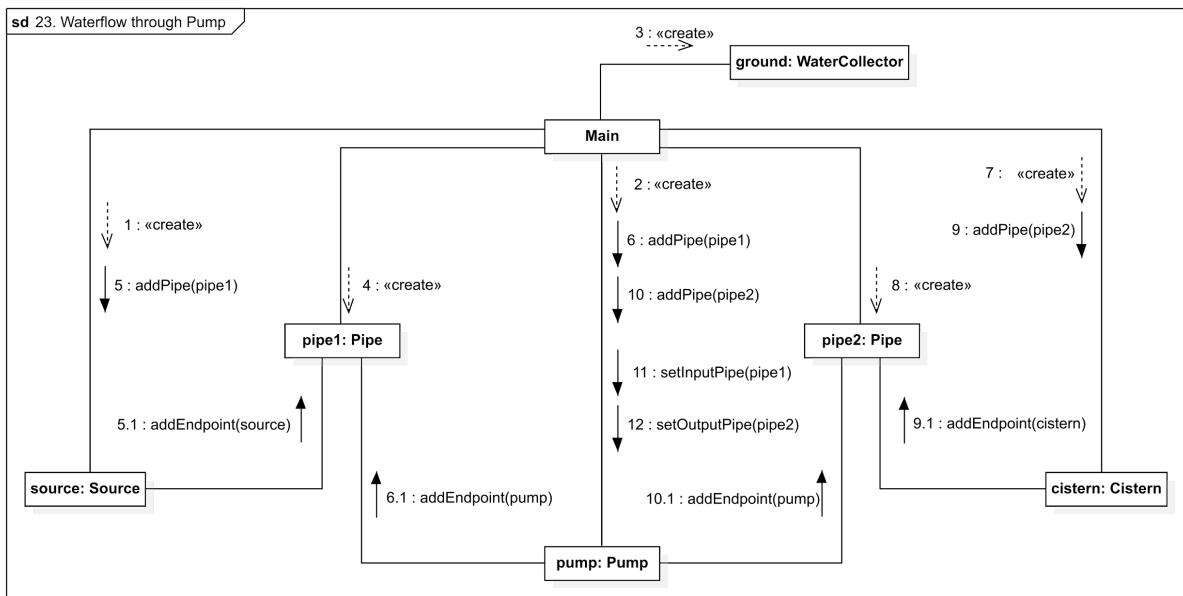
5.4.21 *Spawn Pipe*



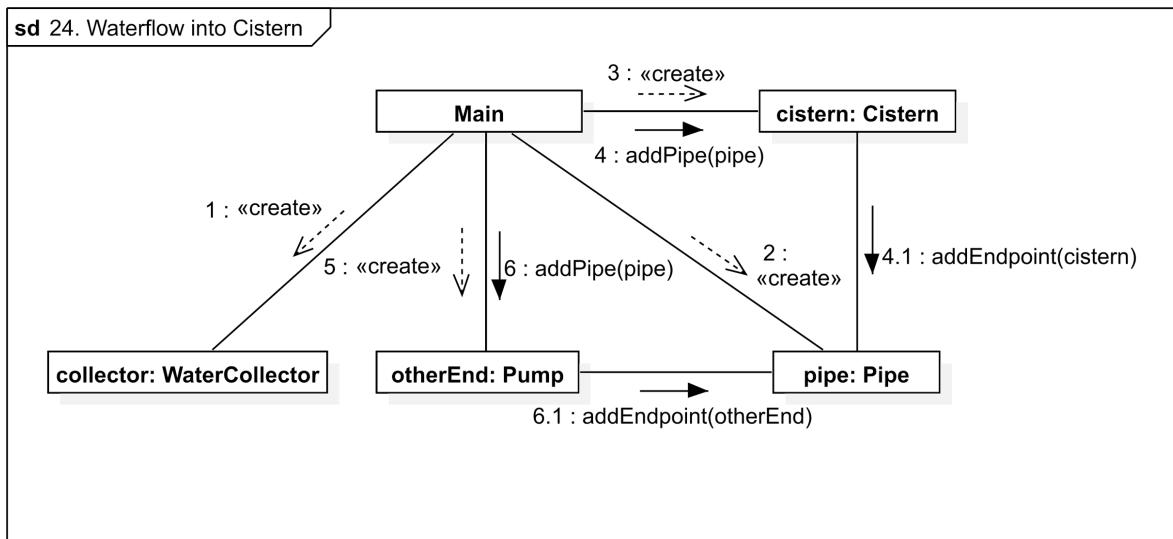
5.4.22 Waterflow from Spring



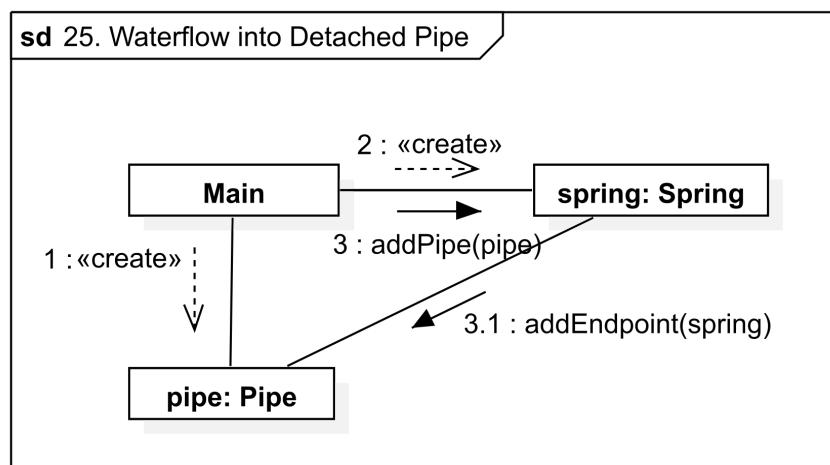
5.4.23 Waterflow through Pump



5.4.24 Waterflow into Cistern



5.4.25 Waterflow into Detached Pipe



Napló

Kezdet	Időtartam	Résztvevők	Leírás
2023.03.29. 12:15	1,5 óra	Harsányi Szilczer Tóth	Értekezlet: Konzultáltunk az IL408-ban a laborvezetővel. Döntés: 03.29. délutáni értekezleten fogjuk átbeszélni és kiosztani a feladatokat.
2023.03.29. 17:00	1 óra	Harsányi Kereszthury Szilczer Tóth	Értekezlet: Feladatok kiosztása Döntés: Harsányi elkészíti az 5.1.1-et, Tóth kijavítja a 4-es dokumentum konzultación megbeszélt hibáit.
2023.03.29. 18:00	0,5 óra	Harsányi	5.1.1 Use-case diagram use-case-k összegyűjtése
2023.03.30. 15:00	1 óra	Tóth	4-es dokumentum javítása.
2023.03.31. 20:00	2 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: Kiosztottuk az elvégzendő feladatokat. Döntés: Harsányi a use-case-eket kiegészíti. Tóth ellenőrzi a use-case-eket és a use-case diagramot. A use-casek kiosztásra kerültek.
2023.04.01. 21:00	1,5 óra	Harsányi	5.1.1 Use-case diagram rajzolás.
2023.04.02. 9:00	4 óra	Kereszthury	1., 2., 14., 20., 21. use-casek elkészítése.
2023.04.02. 12:00	2 óra	Szilczer	17., 18. use case elkészítése.
2023.04.02. 19:00	4 óra	Szilczer	10., 24., 25., use case elkészítése.
2023.04.02. 19:30	6 óra	Galgóczy	12, 13, 15, 16, 19 -es pontokhoz kommunikációs & szekvencia diagrammok, use-case leírások készítése
2023.04.02 21:00	4 óra	Harsányi	3., 4., 5., 6. use-case elkészítése (kommunikációs és szekvenciadiagrammokkal együtt)
2023.04.02 22:30	5 óra	Tóth	8, 9, 11, 22 use case elkészítése, Kereszthury és Szilczer munkájának ellenőrzése.
2023.04.03 2:00	1 óra	Harsányi	5.2 A szkeleton kezelői felületének terve, dialógusok
2023.04.03. 10:00	3 óra	Szilczer	10., 13., 16., 17., 18., 19., 24., 25. use case-ek javítása, diagrammok dokumentumba helyezése.
2023.04.03. 11:00	1 óra	Tóth	23-as use case elkészítése. Harsányi munkájának ellenőrzése.
2023.04.03. 12:00	1 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: A teljes dokumentum átnézése. Döntés: A dokumentum beadásra készen áll.

6. Szkeleton beadás

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Main.java	174	2023.04.06. 22:10	A kód innen indul, tartalmazza a main függvényt.
ObjectFactory.java	7465	2023.04.10. 20:52	A tesztek inicializálása és futása során példányokat készít osztályokból.
Skeleton.java	4942	2023.04.06. 23:10	A szkeleton szoftver futási központja. Meghívja a teszesetek inicializáló és futató függvényeit.
UserIO.java	7631	2023.04.06. 23:10	A tesztek kiírását végzi. Bekér a felhasználótól értékeket.
Utils.java	640	2023.04.10. 23:51	Utility függvény(eket) tartalmazó statikus osztály.

Fájl neve	Méret	Keletkezés ideje	Tartalom
ActiveElement.java	1697	2023.04.06. 23:10	Az Aktív elemek osztálya.
Character.java	1706	2023.04.06. 23:10	A Karakterek osztálya.
Cistern.java	2462	2023.04.06. 23:10	A Ciszternák osztálya.
Element.java	1169	2023.04.06. 23:10	Az Elemek osztálya.
Pipe.java	5960	2023.04.06. 23:10	A Csövek osztálya.
Plumber.java	3283	2023.04.06. 23:10	A Szerelők osztálya.
Pump.java	3525	2023.04.06. 23:10	A Pumpák osztálya.
Saboteur.java	449	2023.04.06. 23:10	A Szabotörök osztálya.
Spring.java	458	2023.04.06. 23:10	A Források osztálya.
WaterCollector.java	453	2023.04.06. 23:10	A Víztartályok osztálya.

Fájl neve	Méret	Keletkezés ideje	Tartalom
AcquirePumpTest.java	692	2023.04.07. 2:28	Az <i>Acquire Pump</i> tesztesetet megvalósító osztály.
AttachBothEndsOfPipeTest.java	1291	2023.04.07. 2:28	Az <i>Attach both ends of Pipe</i> tesztesetet megvalósító osztály.
AttachBothEndsOfPipeToSameActiveElementTest.java	1011	2023.04.07. 2:28	Az <i>Attach both ends of Pipe to same Active Element</i> tesztesetet megvalósító osztály.
BreakPumpTest.java	435	2023.04.07. 2:28	A <i>Break Pump</i> tesztesetet megvalósító osztály.
DetachBothEndsOfNewPipeTest.java	1234	2023.04.07. 2:28	A <i>Detach both ends of New Pipe</i> tesztesetet megvalósító osztály.
DetachBothEndsOfPipeTest.java	1261	2023.04.07. 2:28	A <i>Detach both ends of Pipe</i> tesztesetet megvalósító osztály.
DetachPipeEndWhileHoldingAnotherPipeEndTest.java	1365	2023.04.07. 2:28	A <i>Detach Pipe End while holding another Pipe End</i> tesztesetet megvalósító osztály.
MoveToActiveElementTest.java	965	2023.04.07. 2:28	A <i>Move to Active Element</i> tesztesetet megvalósító osztály.
MoveToDetachedPipeTest.java	836	2023.04.07. 2:28	A <i>Move to Detached Pipe</i> tesztesetet megvalósító osztály.
MoveToPipeTest.java	961	2023.04.07. 2:28	A <i>Move to Pipe</i> tesztesetet megvalósító osztály.
PiercePipeTest.java	1132	2023.04.07. 2:28	A <i>Pierce Pipe</i> tesztesetet megvalósító osztály.
PlacePumpTest.java	1125	2023.04.07. 2:28	A <i>Place Pump</i> tesztesetet megvalósító osztály.
PumpPortLimitReachedTest.java	1400	2023.04.07. 2:28	A <i>Pump port limit reached</i> tesztesetet megvalósító osztály.
RepairPipeTest.java	989	2023.04.07. 2:28	A <i>Repair Pipe</i> tesztesetet megvalósító osztály.
RepairPumpTest.java	616	2023.04.07. 2:28	A <i>Repair Pump</i> tesztesetet megvalósító osztály.
SetPumpInputTest.java	952	2023.04.07. 2:28	A <i>Set Pump Input</i> tesztesetet megvalósító osztály.
SetPumpOutputTest.java	955	2023.04.07. 2:28	A <i>Set Pump Output</i> tesztesetet megvalósító osztály.
SpawnPipeTest.java	550	2023.04.07. 2:28	A <i>Spawn Pipe</i> tesztesetet megvalósító osztály.
TestCase.java	337	2023.04.07. 2:28	A tesztek interfészze, amiből minden más teszt öröklődik.

UnableToPlacePumpTest.java	961	2023.04.07. 2:28	A <i>Unable to Place Pump</i> teszteset megvalósító osztály.
UnableToSetPumpInputTest.java	1004	2023.04.07. 2:28	A <i>Unable to Set Pump Input</i> teszteset megvalósító osztály.
UnableToSetPumpOutputTest.java	1006	2023.04.07. 2:28	A <i>Unable to Set Pump Output</i> teszteset megvalósító osztály.
WaterflowFromSpringTest.java	947	2023.04.07. 2:28	A <i>Waterflow from Spring</i> teszteset megvalósító osztály.
WaterflowIntoCisternTest.java	991	2023.04.07. 2:28	A <i>Waterflow into Cistern</i> teszteset megvalósító osztály.
WaterflowIntoDetachedPipeTest.java	762	2023.04.07. 2:28	A <i>Waterflow into Detached Pipe</i> teszteset megvalósító osztály.
WaterflowThroughPumpTest.java	1195	2023.04.07. 2:28	A <i>Waterflow through Pump</i> teszteset megvalósító osztály.

6.1.2 Fordítás

A fordítás elvégezhető a kari felhő gépen, illetve saját gépen is, amennyiben van telepítve a számítógépre legalább 18-as verziójú Java JDK.

Fordítás lépései:

- A forrásfájlokat tartalmazó ZIP fájlt kicsomagolni tetszőleges helyre.
- Parancssorban elnavigálni arra a mappába, amiben az "src" mappa található. (cd használatával)
- Futtatni az alábbi kódot, ami lefordítja a programot:
`javac -cp src src/macaroni/*.java src/macaroni/model/*.java src/macaroni/tests/*.java -d out`

6.1.3 Futtatás

A futtatás végezhető a kari felhő gépen, illetve saját gépen is, amennyiben van telepítve a számítógépre legalább 18-as verziójú Java JDK.

Futtatás lépései:

- Fordítás elvégzése
- Parancssorban elnavigálni arra a mappába, amiben az "src" mappa található. (cd használatával)
- Futtatni az alábbi kódot, ami elindítja a programot:
`java -cp out macaroni.Main`

6.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Kereszthury Péter	PUKAU0	20%
Szilczer Ármin	THGPZB	20%
Tóth Gábor Tamás	HAFLCT	20%
Galgóczy Gábor	QS2WJR	20%
Harsányi Levente	FD7NQC	20%

6.3 Napló

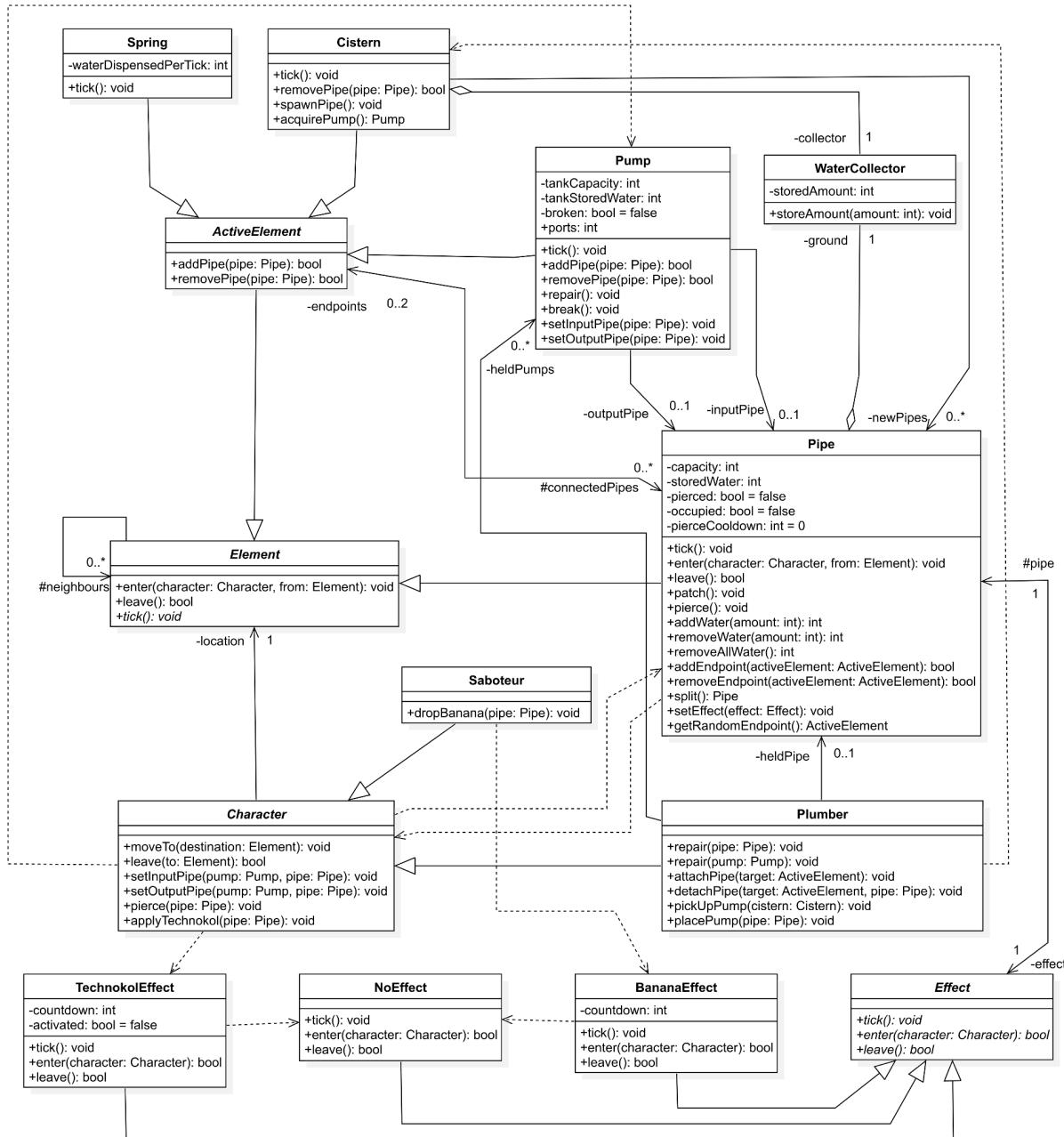
Kezdet	Időtartam	Résztvevők	Leírás
2023.04.05. 12:15	1,5 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: Konzultáltunk az IL408-ban a laborvezetővel. Döntés: A következő értekezleten kiosztjuk az elvégzendő feladatokat.
2023.04.06. 20:00	1 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: Feladatak kiosztása Döntés: Kiosztottuk, hogy a modell és a tesztelőkörnyezet osztályai közül melyiknek ki készítse el a Java-beli implementációját: Galgóczy: Character, Skeleton Harsányi: Plumber, Saboteur Kereszthury: Element, Pump, WaterCollector Szilczer: ActiveElement, Cistern, Spring Tóth: Pipe, UserIO, ObjectFactory
2023.04.06. 21:00	2 óra	Szilczer	ActiveElement, Cistern, Spring osztályok elkészítése.
2023.04.06. 21:00	2 óra	Tóth	Pipe osztály elkészítése.
2023.04.06. 21:00	2 óra	Harsányi	Plumber, Saboteur osztályok elkészítése.
2023.04.06. 21:00	2 óra	Kereszthury	Element, Pump, WaterCollector osztályok elkészítése.
2023.04.06. 21:00	0,5 óra	Galgóczy	Character osztály elkészítése
2023.04.06. 21:30	0,5 óra	Galgóczy	Test interface és konkrét test case-ek vázainak megírása.
2023.04.06. 22:00	1 óra	Galgóczy	MVP Skeleton megírása.
2023.04.10. 19:00	1 óra	Tóth	Element, ActiveElement, Cistern, Spring, Pump ellenőrzése.
2023.04.10. 19:00	1 óra	Szilczer	Pipe, Character, Plumber, Saboteur, WaterCollector ellenőrzése.
2023.04.10. 20:00	1 óra	Tóth	UserIO, ObjectFactory osztályok vázának megírása.
2023.04.12. 22:00	3 óra	Tóth	UserIO, ObjectFactory befejezése, dokumentálása.

2023.04.12. 22:00	2 óra	Galgóczy	Skeleton osztály megírása.
2023.04.13. 16:00	1 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: Elkészítendő tesztek kiosztása. Döntés: Kiosztottuk, ki melyik tesztet írja meg: Harsányi: 2., 3., 4., 5., 6., 7. Kereszthury: 9., 10., 11., 12., 13., 14., 15., 16., 17., 19., 20., 21., 22., 24., 25. Szilczer: 18 Tóth: 1, 8, 23
2023.04.14. 19:00	3 óra	Harsányi	2., 3., 4., 5., 6., 7. test-case-ek megírása
2023.04.14. 22:00	2 óra	Kereszthury	10., 11., 14., 16., 17., 19., 20., 21. test-case-ek megírása
2023.04.15. 00:00	0,5 óra	Szilczer	18-as teszt megírása, ActiveElement osztály javítása.
2023.04.15. 01:00	1 óra	Tóth	UserIO osztály javítása.
2023.04.15. 9:30	1,5 óra	Kereszthury	9., 12., 13., 15., 22., 24., 25. test-case-ek megírása
2023.04.15. 11:00	1 óra	Kereszthury	2., 3., 4., 8. test-case-ek ellenőrzése
2023.04.15. 15:00	1 óra	Tóth	1., 8., 23. tesztek megírása, Pipe osztály javítása.
2023.04.15. 18:00	1 óra	Kereszthury	Test-case-ek ellenőrzése, javítása
2023.04.15. 19:00	5 óra	Szilczer	Mind a 25 teszt ellenőrzése és javítása.
2023.04.16. 23:30	2 óra	Harsányi	6.1.1 Fájllista
2023.04.16. 01:30	0,5 óra	Galgóczy	6.1.1 Fájllista ellenőrzése és javítása
2023.04.17. 10:00	0,5 óra	Tóth	6.1.2 Fordítás, 6.1.3 Futtatás megírása
2023.04.17. 11:00	0,5 óra	Galgóczy Harsányi Szilczer Tóth	Értekezlet: A teljes dokumentum átnézése, kis javítások. Döntés: A dokumentum beadásra készen áll.

7. Prototípus koncepciója

7.0 Változás hatása a modellre

7.0.1 Módosult osztálydiagram



7.0.2 Új vagy megváltozó metódusok

7.0.2.1 ActiveElement

- **Metódusok**
- **void tick()**: Át lett helyezve az Element osztályba.

7.0.2.2 Character

- **Metódusok**
- **void moveTo(Element destination)**: Meghívja a paraméterként kapott *Element enter* függvényét.
- **bool leave(to: Element)**: Meghívja a *location leave* függvényét, ha ez *true*-val tér vissza, akkor beállítja a *location*-t *to*-ra.
- **void pierce(Pipe pipe)**: Megpróbálja kilyukasztani a paraméterként kapott csövet, ha rajta áll.
- **void applyTechnokol(Pipe pipe)**: Összetehetők a paraméterként kapott csövet a cső *setEffect* függvényének meghívásával.

7.0.2.3 Element

- **Metódusok**
- **void tick()**: Absztrakt metódus. Az idő műlását szimulálva lépteti az objektumot.
- **void enter(Character character, Element from)**: Ellenőrzi, hogy a *from* benne van-e a *neighbours* listában, ha igen, meghívja a paraméterül kapott karakter *leave* függvényét.
- **bool leave()**: Értesíti az elemet arról, hogy lelépett róla egy játékos, minden *true*-val tér vissza.

7.0.2.4 Pipe

- **Attribútumok**
- **int pierceCooldown**: A cső (újból) kilyukaszthatóságáig hátralévő idő.
- **Effect effect**: A csövön lévő mozgást befolyásoló effekt.
- **Metódusok**
- **void tick()**: Lépteti a csövet. Csökkenti a *pierceCooldown*-t, és meghívja az *effect tick*-jét.
- **void enter(Character character, Element from)**: Ha a paraméterként kapott *Element* nem a cső szomszédja, vagy a csőnek van lecsatlakoztatott vége, vagy az *occupied* értéke *true*, akkor nem lehetséges rálépni, és *false* értékkel tér vissza. Ellenkező esetben meghívja az *effect enter*-jét, ha ez *true*-t ad vissza, átállítja az *occupied* értékét igazra, és *true*-val tér vissza, ha *false*-ot ad vissza, akkor *false*-szal tér vissza.
- **bool leave()**: Meghívja az *effect leave* függvényét. Ha ez *true*-t ad vissza, a cső *occupied* értéke *false*-ra változik. Visszaadja a meghívott függvény értékét.
- **void patch()**: A cső befoltozása, azaz a *pierced* állapot *false* lesz. Beállítja *pierceCooldown*-t egy véletlen értékre.
- **void pierce()**: Kilyukasztja a csövet, azaz a *pierced* változtat *true*-ra állítja, ha a *pierceCooldown* lejárt (nullával egyenlő). Ilyenkor a *storedWater*-ben lévő vízmennyiséget áthelyezi a *ground*-ba.
- **void setEffect(Effect effect)**: Eltárolja a paraméterül kapott *effectet*.
- **ActiveElement getRandomEndpoint()**: Visszaadja az egyik végpontját véletlenszerűen választva.

7.0.2.5 Plumber

- **Attribútumok**
- **void placePump(Pipe pipe)**: Ha a paraméterként kapott csövön áll, van nála pumpa, és le tud lépni a *location*-jéről (meghívja a *location leave* függvényét), akkor azt a pumpát lehelyezi. Ehhez a kapott csövön meghívja a *split* függvényt, és a pumpába beköti a *pipe*-ot és a *split* visszatérési értékeként kapott másik csövet. Ezután a lehelyezett pumpára áll.

7.0.2.6 Saboteur

- **Attribútumok**
- **void pierce(Pipe pipe)**: Átkerült a *Character* osztályba.
- **void dropBanana(Pipe pipe)**: Ráejt egy banánt a paraméterként kapott csőre a cső *setEffect* függvényének meghívásával.

7.0.2.7 Effect

Csőről, és csőre való mozgást módosító absztrakt osztály.

- **Attribútumok**
- **Pipe pipe**: A cső amin hat az effekt.
- **Attribútumok**
- **void tick()**: Absztrakt metódus. Az idő múlását szimulálva lépteti az objektumot.
- **bool enter(Character character)**: Absztrakt metódus. Meghívódik, ha valaki rálép a csőre. Visszaadja, hogy sikeres volt-e a csőrelépés.
- **bool leave()**: Absztrakt metódus. Meghívódik, ha valaki lelép a csőről, visszaadja, hogy sikeres volt-e a csőről lelépés.

7.0.2.8 NoEffect

Olyan Effect, ami nem módosítja az alap csőre/ről lépés mechanikát.

- **Attribútumok**
- **void tick()**: Nem csinál semmit.
- **bool enter(Character character)**: Meghívja a paraméterül kapott karakter *leave* függvényét a *pipe*-pal. Visszaadja, hogy sikeres volt-e a csőre lépés.
- **bool leave()**: Mindig igazat ad vissza, mert minden le lépni a csőről ennél az effektnél.

7.0.2.9 BananaEffect

A banánozottságot megvalósító effekt. Ha valaki a csőre lép amikor ez az effekt aktív, a cső egyik végére csúszik.

- **Attribútumok**
- **int countdown**: Az effekt hátralévő ideje. Ha nagyobb mint 0, az effekt aktív.
- **Attribútumok**
- **void tick()**: Lépteti az effektet azzal, hogy csökkenti a *countdown*-t eggyel. Ha a *countdown* 0 lesz, az effekt lejár, és beállítja a cső effektjét egy új NoEffect-re.

- **bool enter(Character character)**: Rálépteti a kapott karaktert a *pipe* egyik végpontjára a *leave* függvénnyel. Az egyik végpontot a *pipe getRandomEndpoint* függvényének meghívásával kapja meg.
- **bool leave()**: Mindig igazat ad vissza, mert minden le lépni a csőről ennél az effektnél.

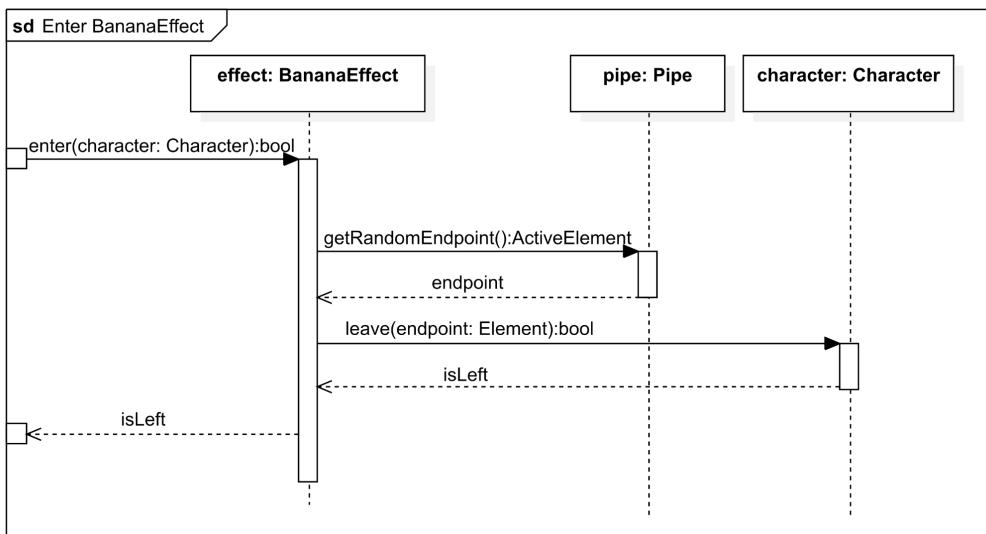
7.0.2.10 TechnokolEffect

A technokolozottság megvalósító effekt. Ha valaki a csőről lelépne amikor ez az effekt aktív, nem sikerül neki, beragad.

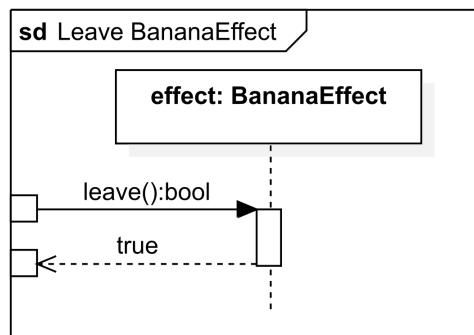
- **Attribútumok**
- **int countdown**: Az effekt hátralévő ideje. Ha nulla lesz, az effekt lejár.
- **bool activated**: Aktív-e az effekt. Ha igen, a lelépés nem fog sikerülni.
- **Metódusok**
- **void tick()**: Lépteti az effektet azzal, hogy csökkenti a *countdown*-t eggyel. Ha a *countdown* 0 lesz, az effekt lejár, és beállítja a cső effektjét egy új NoEffect-re.
- **bool enter(Character character)**: Meghívja a paraméterül kapott karakter *leave* függvényét a *pipe*-pal. Visszaadja, hogy sikeres volt-e a csőre lépés.
- **bool leave()**: Aktiválja az effektet, ha még nem aktív. Ilyenkor még engedi a csőről való lelépést, azzal, hogy *true*-t ad vissza, hagyva, hogy a technokolt lerakó karakter elmeneküljön. Ha az effekt aktív, *false*-szal tér vissza, beragasztva a lelépni próbáló karaktert.

7.0.3 Szekvencia-diagramok

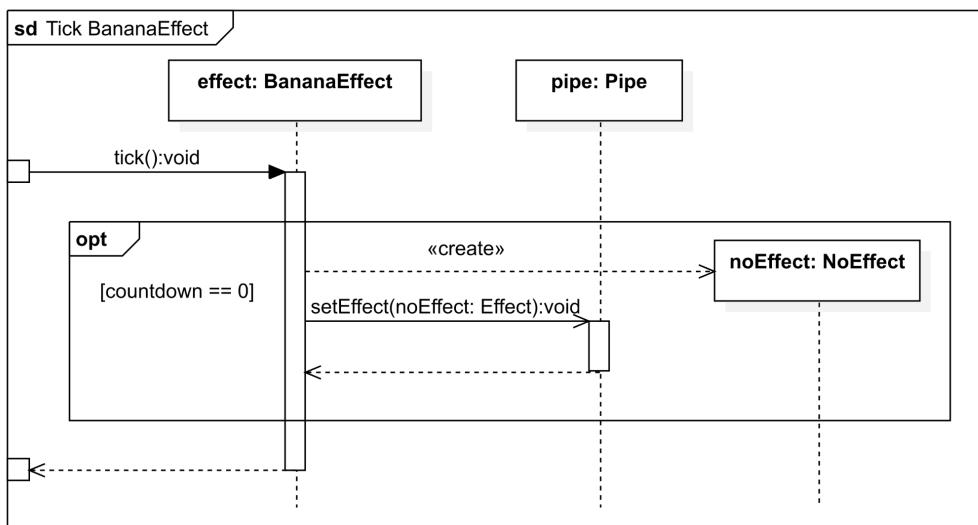
7.0.3.1 Enter BananaEffect



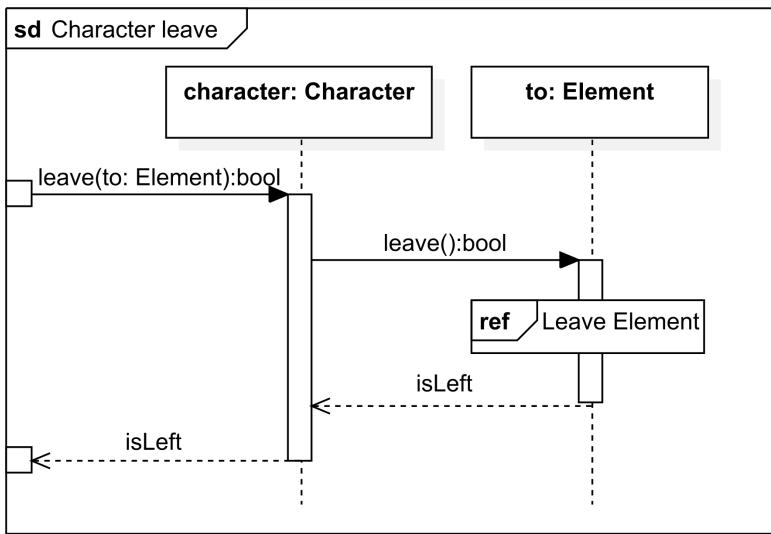
7.0.3.2 LeaveBananaEffect



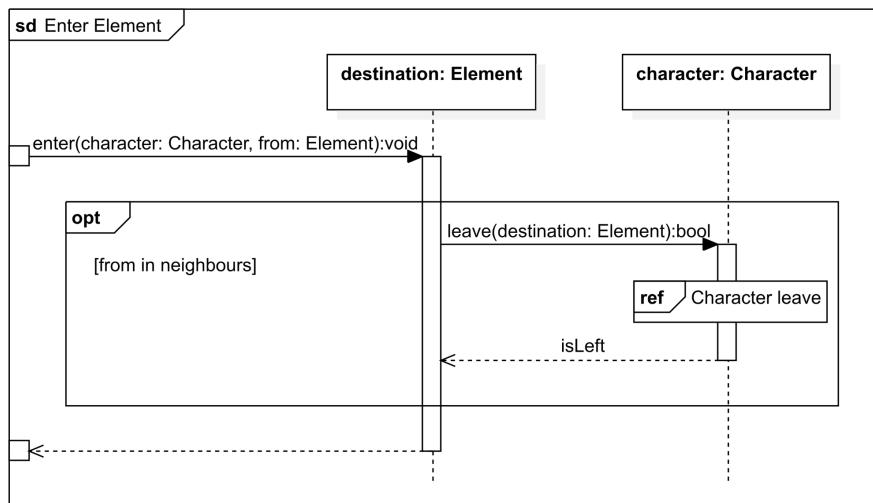
7.0.3.3 Tick BananaEffect



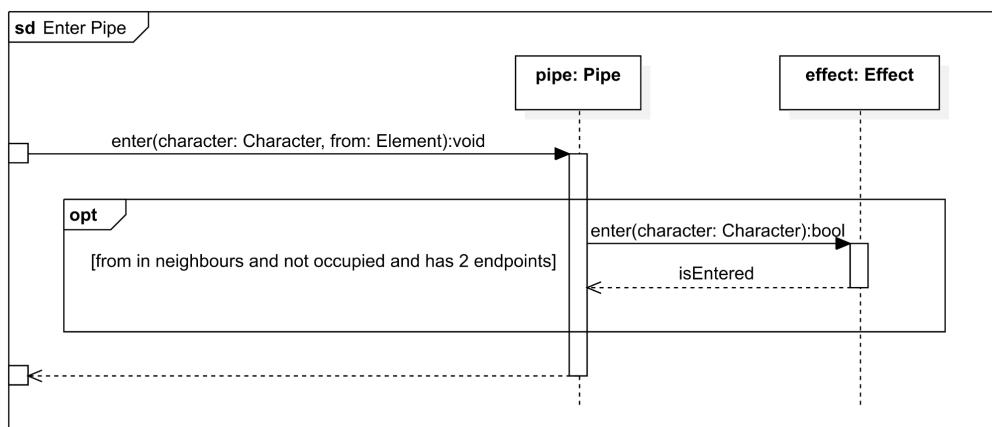
7.0.3.4 Character leave



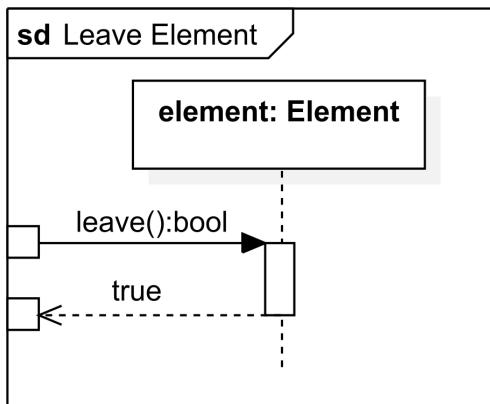
7.0.3.5 Enter Element



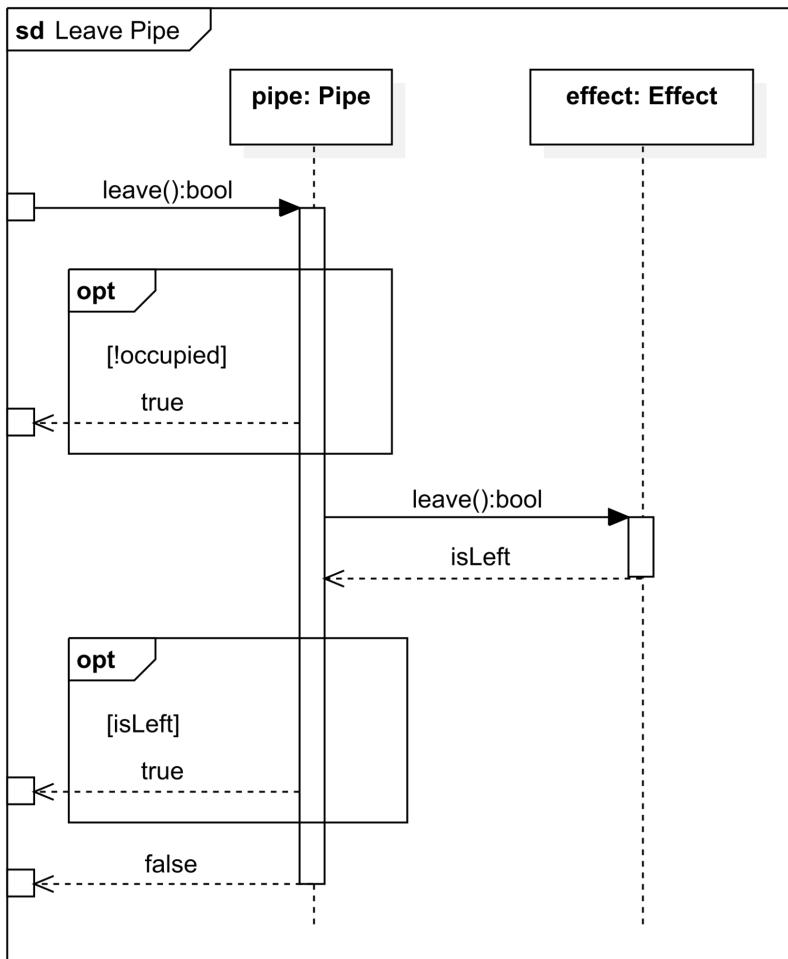
7.0.3.6 Enter Pipe



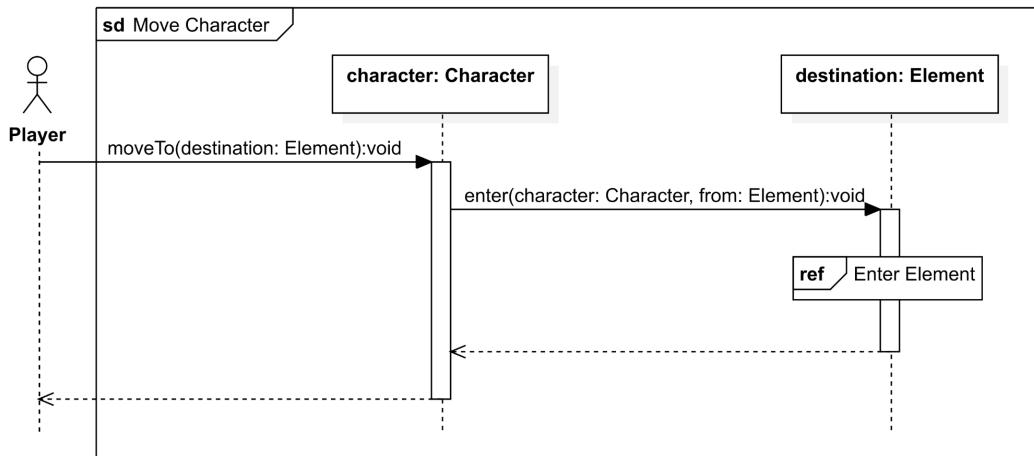
7.0.3.7 Leave Element



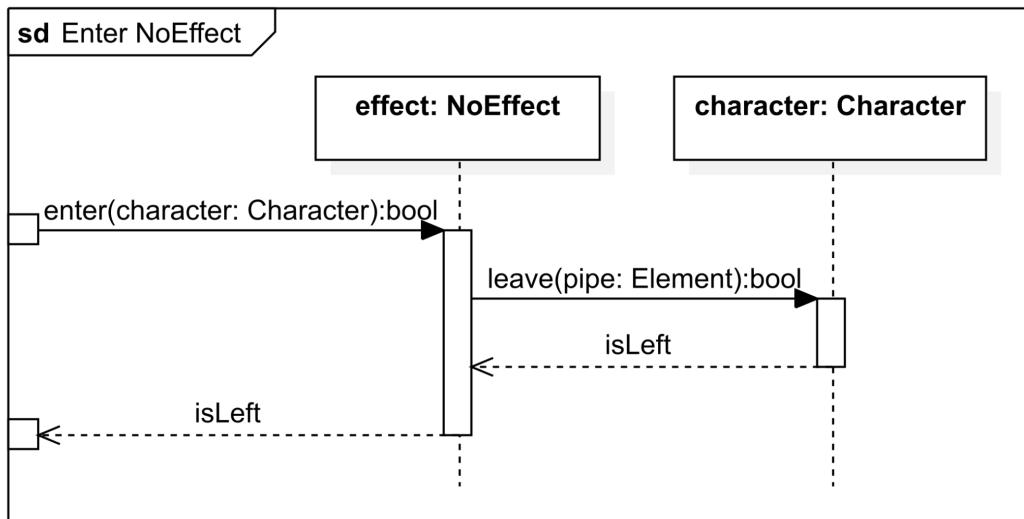
7.0.3.8 Leave Pipe



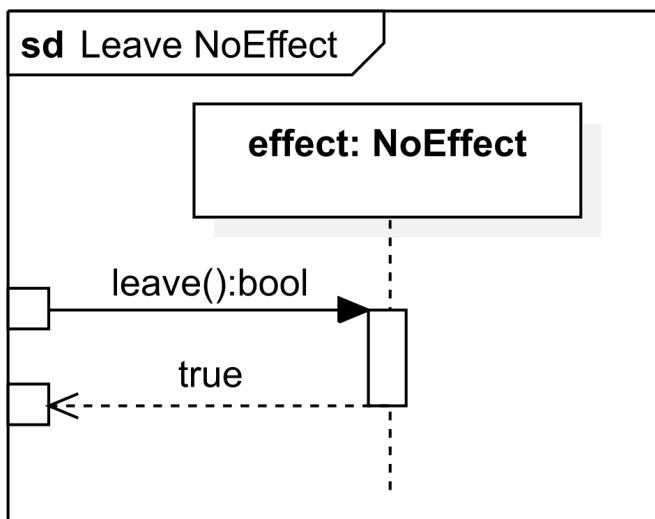
7.0.3.9 Move Character



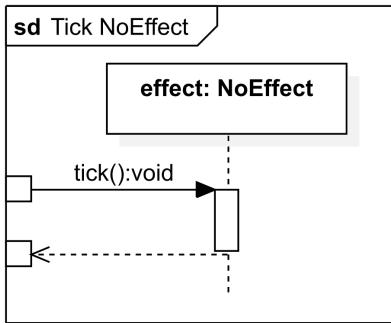
7.0.3.10 Enter NoEffect



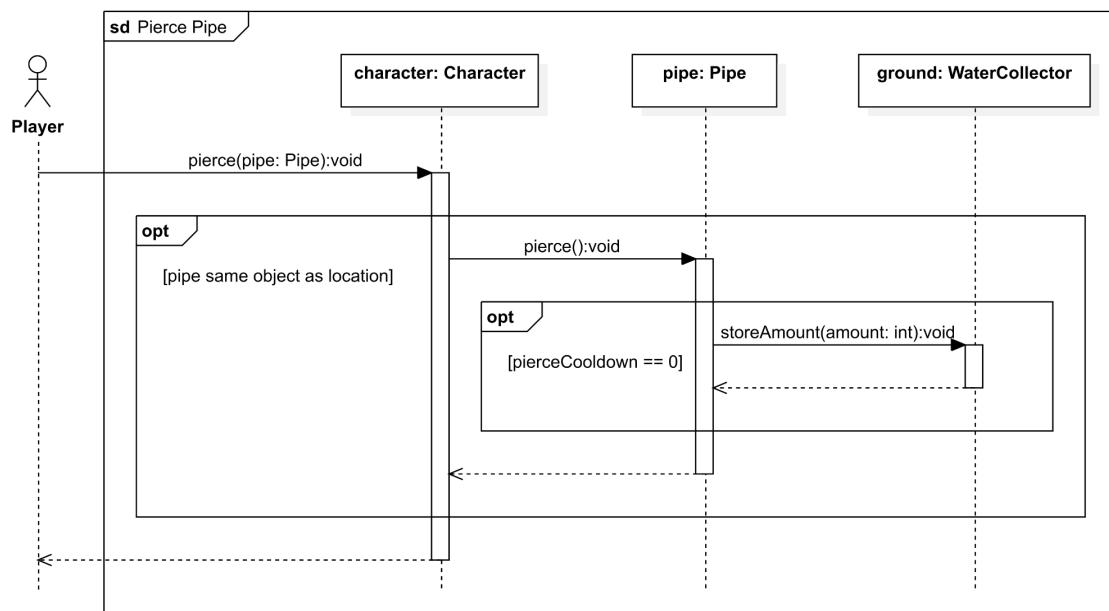
7.0.3.11 Leave NoEffect



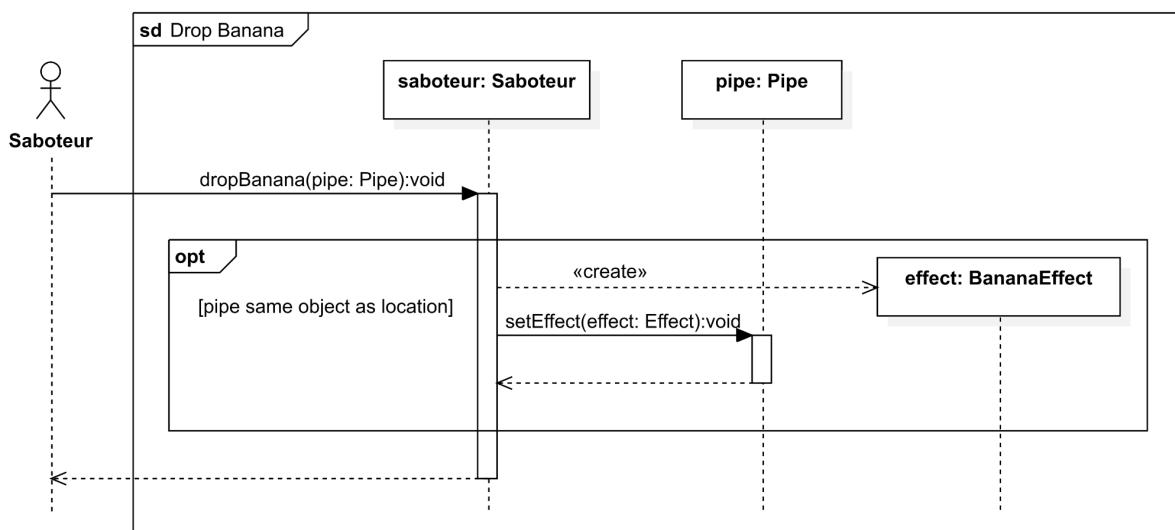
7.0.3.12 Tick NoEffect



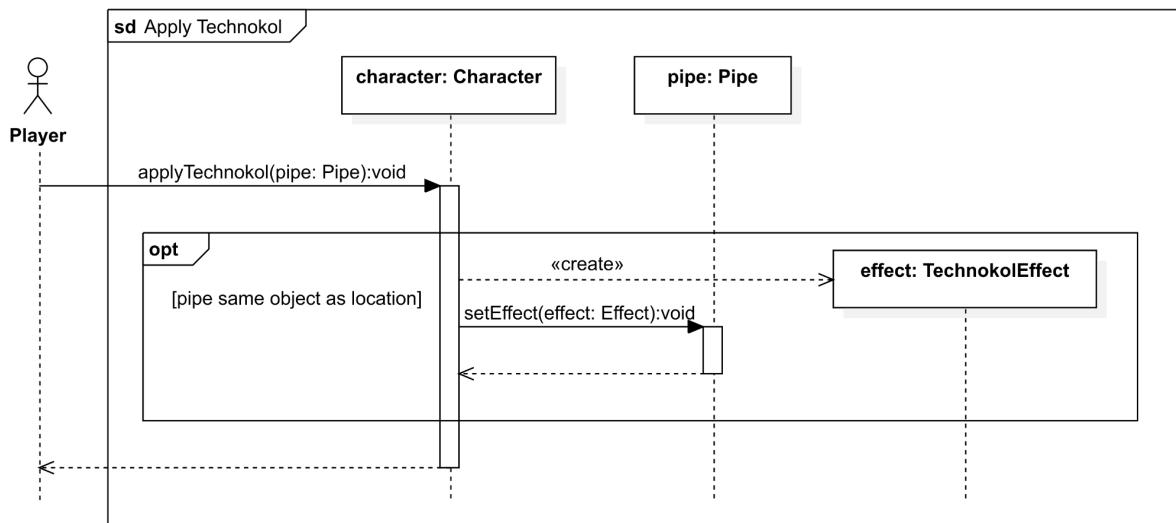
7.0.3.13 Pierce Pipe



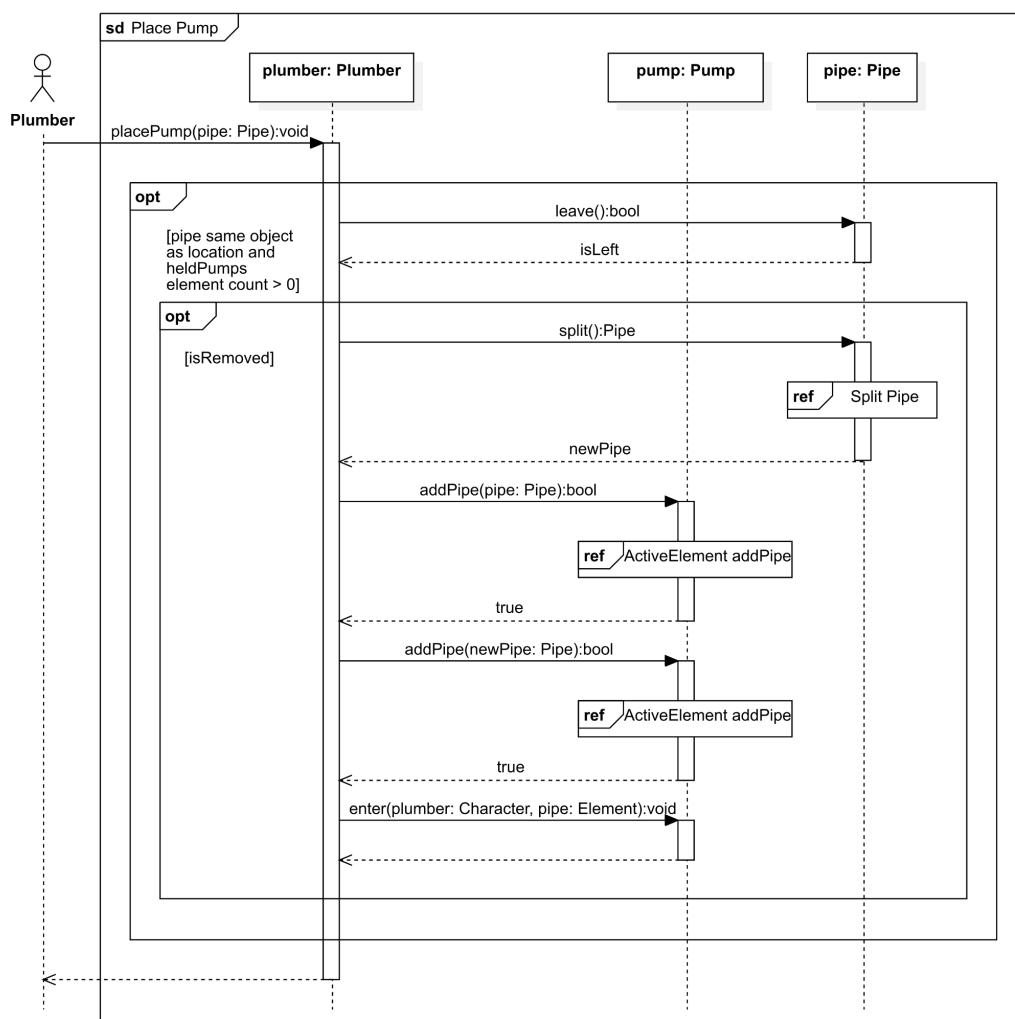
7.0.3.14 Drop Banana



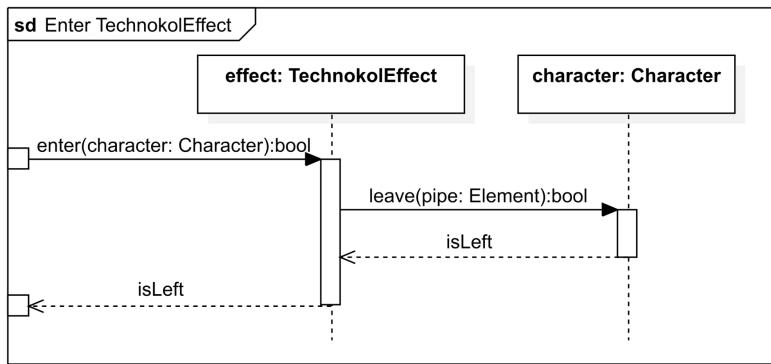
7.0.3.15 Apply Technokol



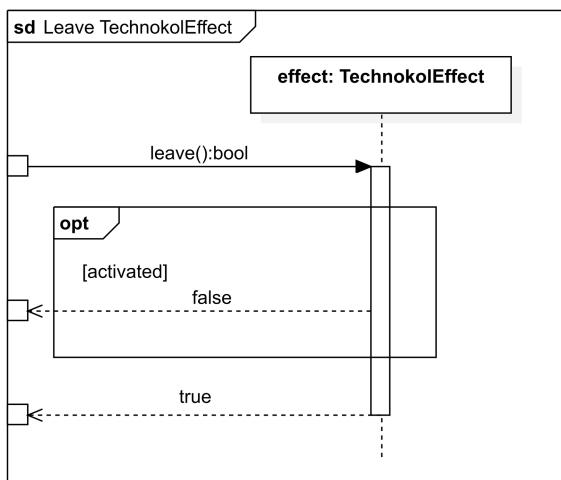
7.0.3.16 Place Pump



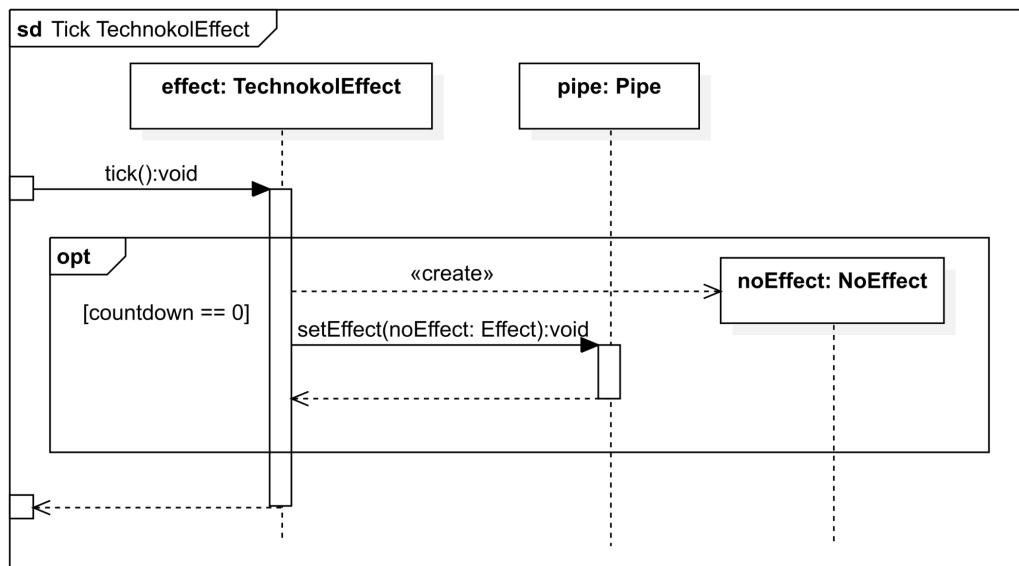
7.0.3.17 Enter TechnokolEffect



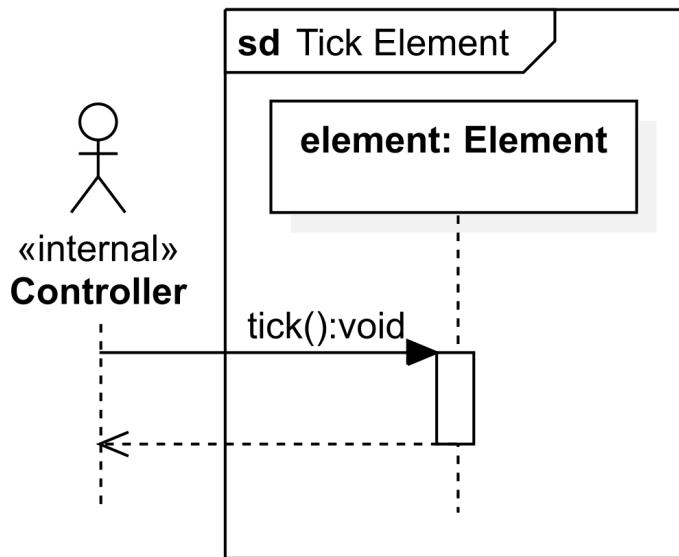
7.0.3.18 Leave TechnokolEffect



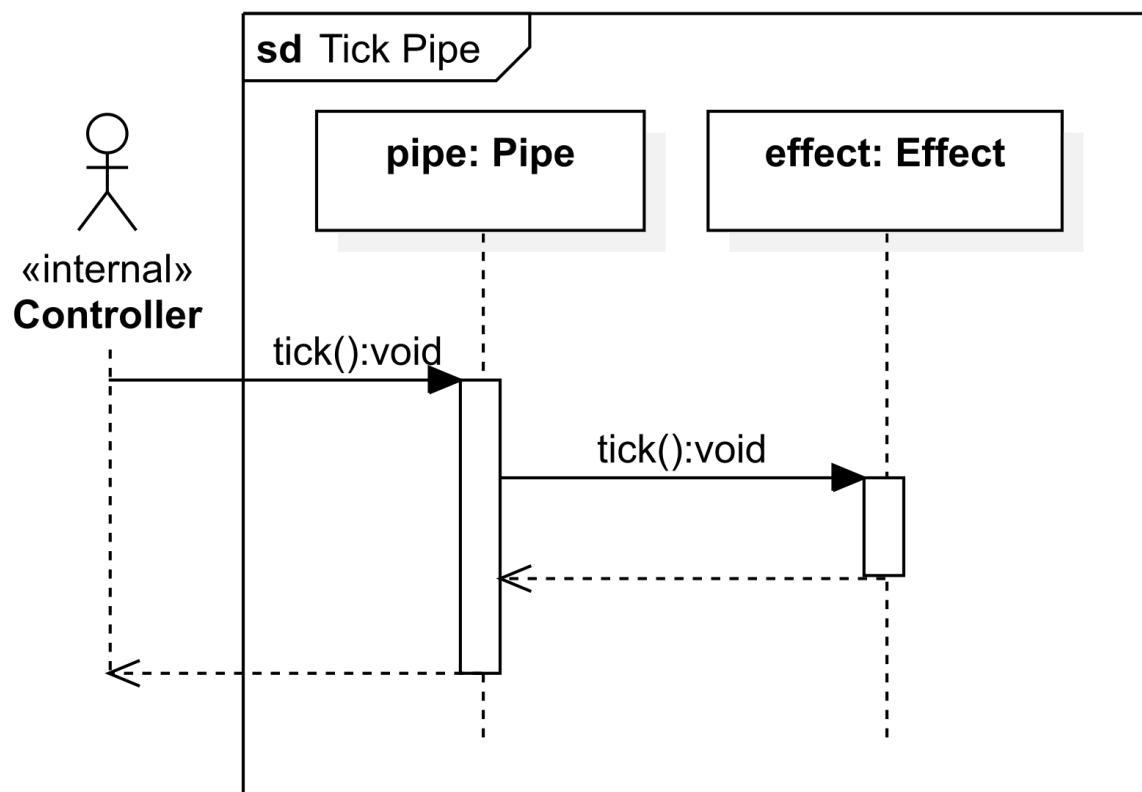
7.0.3.19 Tick TechnokolEffect



7.0.3.20 Tick Element



7.0.3.21 Tick Pipe



7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A parancsok a szabványos bemenetre érkeznek, ami történhet terminálból, illetve előre megírt teszesetek esetén fájlból. A program kimenete a szabványos kimenetre íródik ki.

A proto képes adott konfigurációs fájlokat is beolvasni valamint előállítani.

(A konfigurációs fájl pontos formátuma a 7.1.2 fejezet végén van kifejtve)

7.1.2 Bemeneti nyelv

A program szabványos bemenetén lehet parancsokat megadni. A parancsok után következő paraméterek space-ekkel vannak elválasztva, a parancsfeldolgozás case insensitive.

A programban létrehozott objektumoknak nevet kell adni, a nevek azonban csak az angol abc betűit és számjegyeket tartalmazhatnak.

A program által elfogadott parancsok:

A relációs jelek között szereplő paraméterek kötelezőek, a szögeletes zárójelben lévők opcionálisak, a paraméterek sorrendje kötött. A kettőspontok után az értelmezett paraméterek típusa szerepel.

‘Create’ parancsok esetén különböző pályaelemeket és karaktereket lehet létrehozni, melyeknek kötelező egyedi nevet adni.

Create pump <name: string> [ports: int] [tankCapacity: int]

Leírás:

Létrehoz a pályán egy pumpát.

Paraméterek:

- name: a pumpa neve
- ports: a pumpa csatlakozóinak száma (maximum ennyi cső köthető rá), alapértelmezett értéke **4**
- tankCapacity: a pumpa tartályának kapacitása, alapértelmezett értéke **5**

Create collector <name: string>

Leírás:

Létrehoz egy vízgyűjtőt (WaterCollector-t).

Paraméterek:

- name: a vízgyűjtő neve

Create pipe <name: string> [capacity: int] [ground: WaterCollector]

Leírás:

Létrehoz a pályán egy csövet.

Paraméterek:

- name: a cső neve
- capacity: a cső kapacitása, alapértelmezett értéke **10**
- ground: a vízgyűjtő neve, amit a cső földként fog kezelni. Megadáskor már léteznie kell. Amennyiben nincs megadva, nem lesz definiálva (null).

Create cistern <name: string> [collector: WaterCollector]

Leírás:

Létrehoz a pályán egy ciszternát.

Paraméterek:

- name: a ciszterna neve
- collector: a vízgyűjtő neve, amit a ciszterna használni fog. Megadáskor már léteznie kell. Amennyiben nincs megadva, nem lesz definiálva (null).

Create spring <name: string> [waterDispensedPerTick: int]**Leírás:**

Létrehoz a pályán egy forrást.

Paraméterek:

- name: a forrás neve
- waterDispensedPerTick: a forrásból a csövekbe egy időegység alatt kifolyó vízmennyiség, alapértelmezett értéke 4

Create plumber <name:string> <startLocation: Element>**Leírás:**

Létrehoz egy szerelőt.

Paraméterek:

- name: a szerelő neve
- startLocation: a szerelő kiinduló pozíciójának (pályaelem) neve. Ennek a pályaelemnek már léteznie kell.

Create saboteur <name:string> <startLocation: Element>**Leírás:**

Létrehoz egy szabotőrt.

Paraméterek:

- name: a szabotőr neve
- startLocation: a szabotőr kiinduló pozíciójának (pályaelem) neve. Ennek a pályaelemnek már léteznie kell.

Attach <pipe: Pipe> <element: ActiveElement>**Leírás:**

Hozzáköti a paraméterként kapott csövet a paraméterként kapott aktív elemhez.

Paraméterek:

- pipe: a cső neve
- element: az aktív elem neve

Detach <pipe: Pipe> <element: ActiveElement>**Leírás:**

Megszünteti a paraméterként kapott cső és aktív elem közötti csatlakozást.

Paraméterek:

- pipe: a cső neve
- element: az aktív elem neve

List**Leírás:**

Kilistázza az összes létrehozott pályaelem és karakter nevét.

View <pump: Pump>**Leírás:**

Kiírja a paraméterként kapott pumpa attribútumait.

Paraméterek:

- pump: a pumpa neve

View <collector: WaterCollector>**Leírás:**

Kírja a paraméterként kapott vízgyűjtő attribútumát.

Paraméterek:

- collector: a vízgyűjtő neve

View <pipe: Pipe>

Leírás:

Kírja a paraméterként kapott cső attribútumait.

Paraméterek:

- pipe: a cső neve

View <cistern: Cistern>

Leírás:

Kírja a paraméterként kapott ciszterna attribútumait.

Paraméterek:

- pump: a pumpa neve

View <spring: Spring>

Leírás:

Kírja a paraméterként kapott forrás attribútumait.

Paraméterek:

- pump: a forrás neve

View <plumber: Plumber>

Leírás:

Kírja a paraméterként kapott szerelő attribútumait.

Paraméterek:

- plumber: a szerelő neve

View <saboteur: Saboteur>

Leírás:

Kírja a paraméterként kapott szabotőr attribútumát.

Paraméterek:

- saboteur: a szabotőr neve

Load <path: string>

Leírás:

Betölti az útvonalon megadott konfigurációs (pályát és karaktereket definiáló) fájlt.

A betöltés előtt létrehozott objektumok (elemek és karakterek) megszűnnek létezni.

Paraméterek:

- path: a fájl elérési útja

Save <path: string>

Leírás:

Elmenti az útvonalon megadott konfigurációs (pályát és karaktereket definiáló) fájlba a létrehozott objektumokat.

Paraméterek:

- path: a fájl elérési útja

Help [command: string]

Leírás:

A command paraméter nélkül kiírja az elérhető parancsok neveit. Amennyiben a command paraméter meg van adva, az adott parancs paramétere zését írja ki.

Paraméterek:

- command: a parancs neve, amiről információ kéne

Tick [element: Element]

Leírás:

Meghívja a paraméternek adott elem tick() függvényét. Amennyiben nincs ez a paraméter megadva, az összes létrehozott elemen meghívja.

Paraméterek:

- element: a pályaelem neve, amin külön kell meghívni a tick()-et

Break <pump: Pump>

Leírás:

Eltöri a paraméterként kapott pumpát.

Paraméterek:

- pump: a pumpa neve, amit el kell törni

SpawnPipe <cistern: Cistern>

Leírás:

Létrehoz egy új csövet a paraméterként kapott ciszternán.

Paraméterek:

- cistern: a ciszterna neve, amin csövet kell létrehozni

Move <character: Character> <element: Element>

Leírás:

A paraméterként kapott karaktert a paraméterként kapott pályaelemre mozgatja, amennyiben az lehetséges.

Paraméterek:

- character: a karakter neve, akit mozgatni kell
- element: a pályaelem neve, amire a karaktert mozgatni kell

SetInput <character: Character> <pump: Pump> <pipe: Pipe>

Leírás:

A paraméterként kapott pumpa bemeneti csövénék állítja a kapott csövet a kapott karakter (amennyiben lehetséges).

Paraméterek:

- character: a karakter neve, aki a pumpát állítja
- pump: a pumpa neve, amit állítani kell
- pipe: a beállítandó bemeneti cső

SetOutput <character: Character> <pump: Pump> <pipe: Pipe>

Leírás:

A paraméterként kapott pumpa kimeneti csövénék állítja a kapott csövet a kapott karakter (amennyiben lehetséges).

Paraméterek:

- character: a karakter neve, aki a pumpát állítja
- pump: a pumpa neve, amit állítani kell
- pipe: a beállítandó kimeneti cső

Pierce <character: Character> <pipe: Pipe>**Leírás:**

A paraméterként kapott karakter kilyukasztja a paraméterként kapott csövet (ha lehetséges).

Paraméterek:

- character: a karakter neve, aki a csövet töri
- pipe: a cső, amit el kell törni

Technokol <character: Character> <pipe: Pipe> [technokolTime: int]**Leírás:**

A paraméterként kapott karakter ragadóssá tudja tenni a paraméterként kapott csövet (ha lehetséges).

Paraméterek:

- character: a karakter neve, aki ragaszt
- pipe: a cső, amit össze kell ragasztóznai
- technokolTime: az idő, amíg a cső ragasztós marad, alapértelmezett értéke véletlenszerű

Banana <saboteur: Saboteur> <pipe: Pipe> [bananaTime: int] [slideBack: boolean]**Leírás:**

A paraméterként kapott szabotőr csúszóssá tudja tenni a paraméterként kapott csövet (ha lehetséges).

Paraméterek:

- saboteur: a szabotőr neve, aki síkosít
- pipe: a cső, amit síkosítani kell
- bananaTime: az idő, amíg a cső csúszós marad, alapértelmezett értéke véletlenszerű
- slideBack: igaz, ha a csúszós csőre rálépő karakter kiindulási pontjára csússzon vissza, hamis ha a másik irányba, alapértelmezett értéke véletlenszerű

Repair <plumber: Plumber> <pipe: Pipe> [invulnerability: int]**Leírás:**

A paraméterként kapott szerelő meg tudja javítani a paraméterként kapott csövet (ha lehetséges).

Paraméterek:

- plumber: a szerelő neve, aki javít
- pipe: a cső, amit javítani kell
- invulnerability: az idő, amíg a cső törhetetlen marad, alapértelmezett értéke véletlenszerű

Repair <plumber: Plumber> <pump: Pump>**Leírás:**

A paraméterként kapott szerelő meg tudja javítani a paraméterként kapott pumpát (ha lehetséges).

Paraméterek:

- plumber: a szerelő neve, aki javít
- pump: a pumpa, amit javítani kell

Attach <plumber: Plumber> <element: ActiveElement>**Leírás:**

A paraméterként kapott szerelő rácsatlakoztatja a paraméterként kapott aktív elemre a kezében tartott csövet (ha lehetséges).

Paraméterek:

- plumber: a szerelő neve
- element: az aktív elem, amire csatlakoztat

Detach <plumber: Plumber> <element: ActiveElement> <pipe: Pipe>**Leírás:**

A paraméterként kapott szerelő lecsatlakoztatja a paraméterként kapott aktív elemről a paraméterként kapott csövet (ha lehetséges).

Paraméterek:

- plumber: a szerelő neve
- element: az aktív elem, amire csatlakoztat
- pipe: a lecsatlakoztatni kívánt cső

PickUpPump <plumber: Plumber> <cistern: Cistern>**Leírás:**

A paraméterként kapott szerelő felvesz egy pumpát a paraméterként kapott ciszternáról (ha rajta áll).

Paraméterek:

- plumber: a szerelő neve
- cistern: a ciszterna, amin a szerelő áll

PlacePump <plumber: Plumber> <pipe: Pipe>**Leírás:**

A paraméterként kapott szerelő lehelyez egy pumpát a paraméterként kapott csőre (ha rajta áll).

Paraméterek:

- plumber: a szerelő neve
- pipe: a cső, amin a szerelő áll

A konfigurációs (kimeneti és bemeneti fájlok) nyelvtana:

A fájlok első sorában szerepelnek a létező objektumok nevei és típusai space-ekkel elválasztva az alábbi formátumban [név1]:[típus1] [név2]:[típus2] ...

Az objektumok abc sorrendben szerepelnek a fájlban. (Az első sorban, és a felsorolásban is)

A következő sorokban szerepelnek a különböző objektumok (karakterek és pályaelemek) részletesen kifejtve (névvel és attribútumértékekkel) a következő módon:

[pályaelem vagy karakter neve]

[attribútumérték 1]

[attribútumérték 2]

[...]

[második pályaelem vagy karakter neve]

...

Az attribútumok előtt tabulátor szerepel, a lista vagy tömb attribútumok elemei egy sorban, space-el elválasztva szerepelnek. Az objektumokra mutató referenciákat az objektumok nevei reprezentálják.

Az attribútumok sorrendje megegyezik a ‘View’ parancsokban található attribútumsorrenddel. (Ld.: 7.1.3 Kimeneti nyelv, ‘View’ <...>)

7.1.3 Kimeneti nyelv

Amikor a program parancsok befogadására képes, akkor a felhasználó egy ‘>’ karakterrel kezdődő sorba írhatja az utasításokat.

Érvénytelen vagy hibás formátumú parancs esetén a program az alábbi hibaüzenettel jelez vissza a felhasználónak: “! invalid command, use ‘help’ to see the full list of commands”.

Hibás paraméterezésű parancs esetén a program hibaüzenetet ír ki a felhasználó számára: “! unknown parameter found in command”

Parancsok kimenetei:

Create: Amennyiben a létrehozott pályaelem neve foglalt, a program a következőt írja ki: “! object with name already exists”, különben nem ír ki semmit.

Attach: Ha az összekötést nem a szerelő végzi és sikertelen, a program a következőt írja ki: “! unable to make connection”, különben nem ír ki semmit.

List: Kilistázza az összes objektum nevét (pályaelemet és karajtart egyaránt) Az azonos típusú objektumok egy sorban, a sor elején típusuk jelenik meg. (pl.: “Pipe: pipe1, pipe2, …”)

View: Kiíratja egy adott pályaelem vagy karakter adatait (belőli attribútumait) külön sorokba a következő formátumban: “- <attribútum neve>: <attribútum értéke>” (lista esetén az attribútumértékek vesszővel és space-ekkel vannak elválasztva). Asszociációk esetén az adott objektum(ok) neve(i) van(nak) kiírva.

Az egyes osztályok attribútumai (ebben a sorrendben jelennek meg):

- Pump: tankCapacity, tankStoredWater, broken, ports, outputPipe, inputPipe, connectedPipes
- WaterCollector: storedAmount
- Pipe: capacity, storedWater, pierced, occupied, pierceCooldown, ground, endpoints, effectType (Banana, Technokol or None)
- Cistern: newPipes, collector, connectedPipes
- Spring: waterDispensedTick, connectedPipes
- Plumber: heldPipe, heldPumps, location
- Saboteur: location

Ha nem létezik a megadott nevű objektum, a program a következőt írja ki: “! object not found”.

Load: Amennyiben a betöltés sikertelen, értesíti a felhasználót: “! cannot load file”

Ha a fájl nem létezik a “! file not found” üzenet jelenik meg.

Sikeress betöltés esetén a “load successful” üzenet jelenik meg.

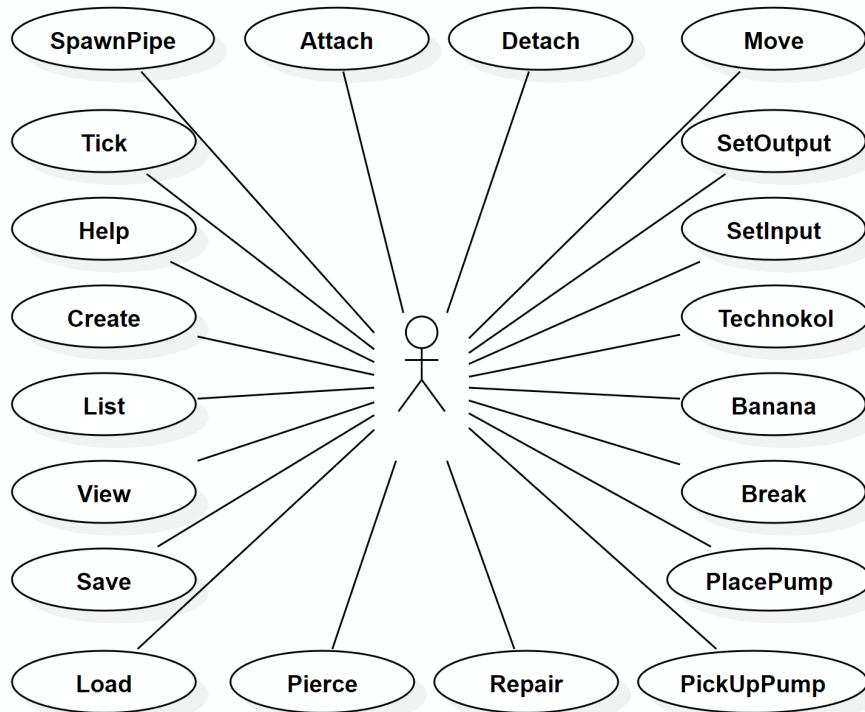
Save: Amennyiben a kiírás sikertelen, értesíti a felhasználót: “! cannot save file”

Sikeress mentés esetén a “save successful” üzenet jelenik meg. A program létrehoz egy új fájlt a paraméterként kapott helyen.

Help: A command paraméter nélkül kiírja az elérhető parancsok neveit egy sorba, vesszővel és space-el elválasztva. Amennyiben a command paraméter meg van adva, az adott parancs paraméterezését írja ki. (A 7.1.2 fejezetben szereplő formában.)

Ha nincsen a command paraméternek megfelelő parancs, a paramétert nem veszi figyelembe.

7.2 Összes részletes use-case



Use-case neve	Help
Rövid leírás	A program segítséget nyújt a felhasználónak.
Aktorok	User
Forgatókönyv	A program kiírja a parancsok listáját
Alternatív forgatókönyv	A program kiírja az adott parancs paramétereit

Use-case neve	Create
Rövid leírás	A felhasználó létrehozhat objektumokat.
Aktorok	User
Forgatókönyv	A felhasználó létrehoz egy pumpát
Alternatív forgatókönyv	A felhasználó létrehoz egy vízgyűjtőt
Alternatív forgatókönyv	A felhasználó létrehoz egy csövet
Alternatív forgatókönyv	A felhasználó létrehoz egy ciszternát
Alternatív forgatókönyv	A felhasználó létrehoz egy forrást
Alternatív forgatókönyv	A felhasználó létrehoz egy szerelőt
Alternatív forgatókönyv	A felhasználó létrehoz egy szabotort
Alternatív forgatókönyv	A felhasználó nem tud létrehozni semmit, mivel a létrehozni kívánt objektum neve foglalt.

Use-case neve	List
Rövid leírás	A program kiírja típusonként a létező objektumok neveit.
Aktorok	User
Forgatókönyv	A program kiírja típusonként a létrehozott objektumok neveit.

Use-case neve	View
Rövid leírás	A felhasználó megnézheti egy pályaelem vagy egy szerelő belső attribútumainak értékét.
Aktorok	User
Forgatókönyv	A felhasználó megnéz egy pumpát
Alternatív forgatókönyv	A felhasználó megnéz egy vízgyűjtőt
Alternatív forgatókönyv	A felhasználó megnéz egy csövet
Alternatív forgatókönyv	A felhasználó megnéz egy ciszternát
Alternatív forgatókönyv	A felhasználó megnéz egy forrást
Alternatív forgatókönyv	A felhasználó megnéz egy szerelőt
Alternatív forgatókönyv	A felhasználó megnéz egy szabotórt
Alternatív forgatókönyv	A felhasználó nem tudja megnézni az objektumot, mert nem létezik.

Use-case neve	Save
Rövid leírás	A program egy külső fájlba menti az objektumok jelenlegi állapotát.
Aktorok	User
Forgatókönyv	1. A program kiírja az adatait egy fájlba 2. A program értesíti a felhasználót a sikeres kiírásról.
Alternatív forgatókönyv	1. A fájl mentése sikertelen. 2. A program értesíti a felhasználót.

Use-case neve	Load
Rövid leírás	A program egy külső fájlból beolvassa az objektumok állapotát és létrehozza azokat.
Aktorok	User
Forgatókönyv	1. A program törli az éppen létező objektumokat. 2. A program beolvassa az objektumok adatait egy fájlból, és létrehozza azokat. 3. A program értesíti a felhasználót a sikeres betöltésről.
Alternatív forgatókönyv	2. A fájl beolvasása sikertelen. 3. A program értesíti a felhasználót.

Use-case neve	Pierce
Rövid leírás	Egy karakter kilyukaszt egy csövet.
Aktorok	User
Forgatókönyv	1. A felhasználó utasít egy karaktert, hogy lukasszon ki egy csövet. 2. A karakter kilyukasztja a csövet.
Alternatív forgatókönyv	1. A felhasználó utasít egy karaktert, hogy lukasszon ki egy csövet. 2. A karakter nem tudja kilyukasztani a csövet.

Use-case neve	Repair
Rövid leírás	Egy szerelő megjavít egy csövet vagy egy pumpát.
Aktorok	User

Forgatókönyv	1. A felhasználó utasít egy szerelőt, hogy javítson meg egy pumpát vagy csövet. 2. A szerelő megjavítja a pumpát vagy csövet.
Alternatív forgatókönyv	1. A felhasználó utasít egy szerelőt, hogy javítson meg egy pumpát vagy csövet. 2. A szerelő nem tudja megjavítani a pumpát vagy csövet.

Use-case neve	PickUpPump
Rövid leírás	Egy szerelő felvesz egy pumpát.
Aktorok	User
Forgatókönyv	1. A felhasználó utasít egy szerelőt, hogy vegyen fel egy pumpát. 2. A szerelő felvesz egy pumpát.
Alternatív forgatókönyv	1. A felhasználó utasít egy szerelőt, hogy vegyen fel egy pumpát. 2. A szerelő nem tudja felvenni a pumpát.

Use-case neve	PlacePump
Rövid leírás	Egy szerelő lehelyez egy pumpát.
Aktorok	User
Forgatókönyv	1. A felhasználó utasít egy szerelőt, hogy helyezzen le egy pumpát. 2. A szerelő lehelyezi a pumpát.
Alternatív forgatókönyv	1. A felhasználó utasít egy szerelőt, hogy helyezzen le egy pumpát. 2. A szerelő nem tudja lehelyezni a pumpát.

Use-case neve	Break
Rövid leírás	A program elront egy pumpát.
Aktorok	User
Forgatókönyv	1. A felhasználó utasítja a programot, hogy rontson el egy pumpát. 2. A program elront egy pumpát.
Alternatív forgatókönyv	1. A felhasználó utasítja a programot, hogy rontson el egy pumpát. 2. A program nem tudja elrontani a pumpát.

Use-case neve	Banana
Rövid leírás	Egy szabotőr csúszóssá tesz egy csövet.
Aktorok	User
Forgatókönyv	1. A felhasználó utasít egy szabotőrt, hogy tegyen csúszóssá egy csövet. 2. A szabotőr csúszóssá tesz egy csövet.
Alternatív forgatókönyv	1. A felhasználó utasít egy szabotőrt, hogy tegyen csúszóssá egy csövet. 2. A szabotőr nem tudja csúszóssá tenni a csövet.

Use-case neve	Technokol
Rövid leírás	Egy karakter ragadóssá tesz egy csövet.

Aktorok	User
Forgatókönyv	<p>1. A felhasználó utasít egy karaktert, hogy tegyen ragadóssá egy csövet.</p> <p>2. A karakter ragadóssá tesz egy csövet.</p>
Alternatív forgatókönyv	<p>1. A felhasználó utasít egy karaktert, hogy tegyen ragadóssá egy csövet.</p> <p>2. A karakter nem tudja ragadóssá tenni a csövet.</p>

Use-case neve	SetInput
Rövid leírás	Egy karakter beállít egy csövet egy pumpa bemeneti csövének.
Aktorok	User
Forgatókönyv	<p>1. A felhasználó utasít egy karaktert, hogy tegyen egy csövet bemeneti csővé egy pumpában.</p> <p>2. A karakter beállítja a bemeneti csövet.</p>
Alternatív forgatókönyv	<p>1. A felhasználó utasít egy karaktert, hogy tegyen egy csövet bemeneti csővé egy pumpában.</p> <p>2. A karakter nem tudja beállítani a bemeneti csövet.</p>

Use-case neve	SetOutput
Rövid leírás	Egy karakter beállít egy csövet egy pumpa kimeneti csövének.
Aktorok	User
Forgatókönyv	<p>1. A felhasználó utasít egy karaktert, hogy tegyen egy csövet kimeneti csővé egy pumpában.</p> <p>2. A karakter beállítja a kimeneti csövet.</p>
Alternatív forgatókönyv	<p>1. A felhasználó utasít egy karaktert, hogy tegyen egy csövet kimeneti csővé egy pumpában.</p> <p>2. A karakter nem tudja beállítani a kimeneti csövet.</p>

Use-case neve	Move
Rövid leírás	Egy karakter mozog.
Aktorok	User
Forgatókönyv	<p>1. A felhasználó utasít egy karaktert, hogy lépjen egy megadott elemre.</p> <p>2. A karakter rálép a megadott elemre.</p>
Alternatív forgatókönyv	<p>1. A felhasználó utasít egy karaktert, hogy lépjen egy megadott elemre.</p> <p>2. A karakter nem tud rálépní a megadott elemre.</p>

Use-case neve	Detach
Rövid leírás	A program vagy egy karakter lecsatlakoztat egy csövet.
Aktorok	User
Forgatókönyv	<p>1. A felhasználó utasítja a programot, hogy csatlakoztasson le egy adott elemről egy csövet.</p> <p>2. A program lecsatlakoztatja a csövet.</p>
Alternatív forgatókönyv	<p>1. A felhasználó utasít egy szerelőt, hogy csatlakoztasson le egy adott elemről egy csövet.</p> <p>2. A szerelő lecsatlakoztatja a csövet.</p>

Alternatív forgatókönyv	1. A felhasználó utasít egy szerelőt, hogy csatlakoztasson le egy adott elemről egy csövet. 2. A szerelő nem tudja lecsatlakoztatni a csövet.
--------------------------------	--

Use-case neve	Attach
Rövid leírás	A program vagy egy karakter felcsatlakoztat egy csövet.
Aktorok	User
Forgatókönyv	1. A felhasználó utasítja a programot, hogy csatlakoztasson fel egy adott elemre egy csövet. 2. A program felcsatlakoztatja a csövet.
Alternatív forgatókönyv	1. A felhasználó utasítja a programot, hogy csatlakoztasson fel egy adott elemre egy csövet. 2. A program nem tudja felcsatlakoztatni a csövet.
Alternatív forgatókönyv	1. A felhasználó utasít egy szerelőt, hogy csatlakoztasson fel egy adott elemre egy csövet. 2. A szerelő felcsatlakoztatja a csövet.
Alternatív forgatókönyv	1. A felhasználó utasít egy szerelőt, hogy csatlakoztasson fel egy adott elemre egy csövet. 2. A szerelő nem tudja felcsatlakoztatni a csövet.

Use-case neve	SpawnPipe
Rövid leírás	A program létrehoz egy ciszternán egy új csövet.
Aktorok	User
Forgatókönyv	1. A felhasználó utasítja a programot, hogy hozzon létre egy ciszternán egy új csövet. 2. A program létrehozza az új csövet.

Use-case neve	Tick
Rövid leírás	A program lépteti az elemeket.
Aktorok	User
Forgatókönyv	1. A felhasználó utasítja a programot, hogy lépesse az elemeket. 2. A program lépteti az elemeket.
Alternatív forgatókönyv	1. A felhasználó utasítja a programot, hogy léptessen egy adott elemet. 2. A program lépteti az adott elemet.

7.3 Tesztelési terv

[A tesztelési tervben definiálni kell, hogy a be- és kimeneti fájlok egybevetésével miként végezhető el a program tesztelése. Meg kell adni magas szintű teszt forgatókönyveket. Az egyes teszteket elég informálisan, szabad szövegként leírni, tesztesetenként egy-öt mondatban. minden teszthez meg kell adni, hogy mi a célja, a proto mely funkcionálitását, osztályait stb. teszteli. Az alábbi táblázat minden teszt-esethez külön-külön elkészítendő.]

Teszt-eset neve	1. Repair Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa és egy hozzájuk csatlakoztatott kilyukasztott cső szerepel. A csövön egy szerelő áll. <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő megjavítja a csövet <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> 2 pumpa, a köztük lévő megjavított cső, amin egy szerelő áll, és a csőben tárolva van, hogy egy ideig nem lehet kilyukasztani.
Teszt célja	<p>Megbizonyosodni arról, hogy egy szerelő meg tud javítani egy kilyukasztott csövet.</p> <p>Teszteli a Plumber és Pipe osztályok <i>repair</i> metódusát.</p>

Teszt-eset neve	2. Repair Pump
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, amin egy elromlott pumpa szerepel, rajta egy szerelővel <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő megjavítja a pumpát <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> Egy megjavított pumpa, és a rajta levő szerelő
Teszt célja	<p>Megbizonyosodni arról, hogy a szerelő meg tud javítani egy elromlott pumpát.</p> <p>Teszteli a Plumber és Pump osztályok <i>repair</i> metódusát.</p>

Teszt-eset neve	3. Pierce Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa és egy hozzájuk csatlakoztatott cső szerepel. A csövön egy szerelő áll. <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő kilyukasztja a csövet <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> 2 pumpa, a köztük lévő kilyukasztott cső, amin egy szerelő áll
Teszt célja	<p>Megbizonyosodni arról, hogy egy karakter képes lyukasztani egy csövet.</p> <p>Teszteli a Character és Pipe osztályok <i>pierce</i> metódusát.</p>

Teszt-eset neve	4. Pierce unpierceable Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa és egy hozzájuk csatlakoztatott cső szerepel. A csövön egy szerelő áll. A csőben el van tárolva, hogy nemrég volt javítva. <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő megkísérli kilyukasztani a csövet <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> 2 pumpa köztük egy kilyukasztatlan cső rajta egy szerelő
Teszt célja	Megbizonyosodni arról, hogy nem lehet azonnal kilyukasztani egy nemrégiben megjavított csövet. Teszteli a Plumber osztály <i>piercePipe</i> és a Pipe osztály <i>pierce</i> metódusát.

Teszt-eset neve	5. Technokol Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa és egy hozzájuk csatlakoztatott cső szerepel. A csövön egy szerelő áll. <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő ragadóssá teszi a csövet <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> 2 pumpa, köztük egy ragadós cső, ezen egy szerelő
Teszt célja	Megbizonyosodni arról, hogy egy karakter képes ragadóssá tenni egy csövet. Teszteli a Character és Pipe osztályok <i>technokol</i> metódusát.

Teszt-eset neve	6. Banana Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa és egy hozzájuk csatlakoztatott cső szerepel. A csövön egy szabotőr áll. <p>Teszt futása:</p> <ol style="list-style-type: none"> A szabotőr csúszóssá teszi a csövet <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> 2 pumpa, köztük egy csúszós cső, ezen egy szerelő
Teszt célja	Megbizonyosodni arról, hogy egy szabotőr képes csúszóssá tenni egy csövet. Teszteli a Saboteur és Pipe osztályok <i>banana</i> metódusát.

Teszt-eset neve	7. Set Pump input
Rövid leírás	Inicializálás:

	<ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa és egy hozzájuk csatlakoztatott cső szerepel. Egy szerelő áll az első pumpán. <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő átállítja a pumpa bemenetét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> 2 pumpa, köztük egy cső, az 1. pumpa bemenete a cső és a szerelő az 1. pumpán áll
Teszt célja	Megbizonyosodni arról, hogy egy karakter képes átállítani egy pumpa bemenetét. Teszteli a Character és Pump osztályok <i>setPumpInput</i> metódusát.

Teszt-eset neve	8. Set Pump output
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa és egy hozzájuk csatlakoztatott cső szerepel. Egy szerelő áll az első pumpán. <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő átállítja a pumpa kimenetét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> 2 pumpa, köztük egy cső, az 1. pumpa kimenete a cső és a szerelő az 1. pumpán áll
Teszt célja	Megbizonyosodni arról, hogy egy karakter képes átállítani egy cső kimenetét. Teszteli a Character és Pump osztályok <i>setPumpOutput</i> metódusát.

Teszt-eset neve	9. Unable to Set Pump input
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa és egy hozzájuk csatlakoztatott cső szerepel. A cső be van állítva a pumpa kimenetének. Egy szerelő áll az első pumpán. <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő megpróbálja átállítani a pumpa bemenetét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> 2 pumpa, köztük egy cső, az 1. pumpa kimenete a cső, bemenettel nem rendelkezik és a szerelő az 1. pumpán áll
Teszt célja	Megbizonyosodni arról, hogy egy karakter nem képes átállítani egy cső bemenetét, ha az adott cső már be van állítva a pumpán kimenetként. Teszteli a Character és Pump osztályok <i>setPumpInput</i> metódusát.

Teszt-eset neve	10. Unable to Set Pump output
Rövid leírás	Inicializálás:

	<ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa és egy hozzájuk csatlakoztatott cső szerepel. A cső be van állítva a pumpa bemenetének. Egy szerelő áll az első pumpán. <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő megpróbálja átállítani a pumpa kimenetét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> 2 pumpa, köztük egy cső, az 1. pumpa bemenete a cső, kimenettel nem rendelkezik, és a szerelő az 1. pumpán áll
Teszt célja	Megbizonyosodni arról, hogy egy karakter nem képes átállítani egy cső kimenetét, ha az adott cső már be van állítva a pumpán bemenetként. Teszteli a Character és Pump osztályok <i>setPumpOutput</i> metódusát.

Teszt-eset neve	11. Detach both ends of unoccupied Pipe from Pump
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa és egy hozzájuk csatlakoztatott cső szerepel. A pumpákon egy-egy szerelő áll. <p>Teszt futása:</p> <ol style="list-style-type: none"> Az 1. szerelő lecsatlakoztatja a cső felé eső végét A 2. szerelő lecsatlakoztatja a cső felé eső végét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> 2 pumpán 1-1 szerelő kezükben ugyanannak a csőnek két vége A cső nincs hozzá csatlakoztatva egy pumpához sem
Teszt célja	Megbizonyosodni arról, hogy szerelők képesek felemelni minden minden két végét egy csőnek, ha azon senki sem áll. Teszteli a Plumber osztály <i>detachPipe</i> , Pump <i>removePipe</i> és Pipe <i>removeEndpoint</i> metódusokat.

Teszt-eset neve	12. Detach Pipe from Cistern
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol egy ciszterna és egy pumpa van, hozzájuk csatlakoztatva egy cső, a ciszternán egy szerelő áll. <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő lecsatlakoztatja a cső felé eső végét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> A pumpához hozzá van csatlakoztatva egy cső vég, de a ciszternához már nem A szerelő a ciszternán áll és nála van a cső másik vége

Teszt célja	Megbizonyosodni, arról hogy egy szerelő képes felemelni csővéget egy cisternáról. Teszteli a Plumber osztály <i>detachPipe</i> , Cistern <i>removePipe</i> és Pipe <i>removeEndpoint</i> metódusokat.
--------------------	--

Teszt-eset neve	13. Detach Pipe from Spring
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol egy forrás és egy pumpa van, hozzájuk csatlakoztatva egy cső, a forráson egy szerelő áll. <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő lecsatlakoztatja a cső felé eső végét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> A pumpához hozzá van csatlakoztatva egy cső vég, de a forráshoz már nem A szerelő a forráson áll és nála van a cső másik vége
Teszt célja	Megbizonyosodni, arról hogy egy szerelő képes felemelni csővéget egy forrásról. Teszteli a Plumber osztály <i>detachPipe</i> , ActiveElement <i>removePipe</i> és Pipe <i>removeEndpoint</i> metódusokat.

Teszt-eset neve	14. Detach occupied Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa hozzájuk csatlakoztatva egy cső és 2 szerelő Az egyik szerelő az egyik pumpán áll, a másik a csövön <p>Teszt futása:</p> <ol style="list-style-type: none"> A pumpán álló szerelő megpróbálja lecsatlakoztatni a cső felé eső végét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> A kimenet megegyezik a bemenettel
Teszt célja	Megbizonyosodni, arról hogy egy szerelő nem képes felemelni egy olyan csőnek a végét, amin valaki áll. Teszteli a Plumber osztály <i>detachPipe</i> , ActiveElement <i>removePipe</i> és Pipe <i>removeEndpoint</i> metódusokat.

Teszt-eset neve	15. Detach Pipe end while holding another Pipe end
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa van, hozzájuk csatlakoztatva egy cső, és az első pumpán egy szerelő áll A szerelő kezében egy másik, a második pumpához hozzácsatlakoztatott cső vége van <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő megpróbálja lecsatlakoztatni a cső felé eső végét <p>Elvárt kimenet:</p>

	<ul style="list-style-type: none"> • A kimenet megegyezik a bemenettel • *tehát nem történt semmi
Teszt célja	Megbizonyosodni, arról hogy egy szerelő nem képes felvenni új csövet, ha már van egy cső a kezében. Teszteli a Plumber osztály <i>detachPipe</i> , ActiveElement <i>removePipe</i> és Pipe <i>removeEndpoint</i> metódusokat.

Teszt-eset neve	16. Detach both ends of new Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol egy ciszterna szerepel, rajta 2 szerelővel • Ciszternához van csatlakoztatva egy új cső <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. Az 1. szerelő lecsatlakoztatja a cső tőle messzebb eső végét. 2. A 2. szerelő lecsatlakoztatja a cső másik végét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • A ciszternán 2 szerelő áll kezükben a cső egy-egy vége
Teszt célja	Megbizonyosodni, arról hogy szerelők képesek felemelni minden két végét egy újonnan keletkezett csőnek. Teszteli a Plumber osztály <i>detachPipe</i> , Cistern <i>removePipe</i> és Pipe <i>removeEndpoint</i> metódusokat.

Teszt-eset neve	17. Attach both ends of Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa, rajtuk egy-egy szerelő kezükben egy cső két vége. <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. Az 1. szerelő felcsatlakoztatja a cső nála lévő végét 2. A 2. szerelő felcsatlakoztatja a cső másik végét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • 2 pumpa hozzájuk csatlakoztatva egy cső, minden két pumpán egy-egy szerelő, üres kézzel
Teszt célja	Megbizonyosodni, arról hogy szerelők képesek felcsatlakoztatni minden két végét egy teljesen lecsatlakoztatott csőnek. Teszteli a Plumber osztály <i>attach</i> , ActiveElement <i>addPipe</i> és Pipe <i>addEndpoints</i> metódusokat.

Teszt-eset neve	18. Attach both ends of Pipe to same Pump
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol 1 pumpa, hozzá csatlakoztatva egy cső egyik vége • A pumpán egy szerelő kezében a cső másik vége <p>Teszt futása:</p>

	<p>1. A szerelő megpróbálja felcsatlakoztatni a cső másik végét Elvárt kimenet:</p> <ul style="list-style-type: none"> • A kimenet megegyezik a bemenettel
Teszt célja	<p>Megbizonyosodni, arról hogy szerelők nem képesek felesatlakoztatni minden két végét egy féllel lecsatlakoztatott csőnek ugyanahhoz a pumpához ahova már az egyik vége csatlakoztatva van. Teszteli a Plumber osztály <i>attach</i>, Pump <i>addPipe</i> és Pipe <i>addEndpoint</i> metódusokat.</p>

Teszt-eset neve	19. Attach both ends of Pipe to same Cistern
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol 1 ciszterna, hozzá csatlakoztatva egy cső egyik vége • A ciszternán egy szerelő kezében a cső másik vége <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A szerelő megpróbálja felcsatlakoztatni a cső másik végét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • A kimenet megegyezik a bemenettel
Teszt célja	<p>Megbizonyosodni, arról hogy szerelők nem képesek felesatlakoztatni minden két végét egy féllel lecsatlakoztatott csőnek ugyanahhoz a Ciszternához ahova már az egyik vége csatlakoztatva van. Teszteli a Plumber osztály <i>attach</i>, Cistern <i>addPipe</i> és Pipe <i>addEndpoint</i> metódusokat.</p>

Teszt-eset neve	20. Attach both ends of Pipe to same Spring
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol 1 forrás, hozzá csatlakoztatva egy cső egyik vége • A forráson egy szerelő kezében a cső másik vége <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A szerelő megpróbálja felcsatlakoztatni a cső másik végét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • A kimenet megegyezik a bemenettel
Teszt célja	<p>Megbizonyosodni, arról hogy szerelők nem képesek felesatlakoztatni minden két végét egy féllel lecsatlakoztatott csőnek ugyanahhoz a forráshoz ahova már az egyik vége csatlakoztatva van. Teszteli a Plumber osztály <i>attach</i>, ActiveElement <i>addPipe</i> és Pipe <i>addEndpoint</i> metódusokat.</p>

Teszt-eset neve	21. Attach Pipe to Pump with no free ports
Rövid leírás	Inicializálás:

	<ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol 1 pumpa, hozzá csatlakoztatva 2 csővég • A pumpa portszáma 2 (tehát maximum 2 csővég csatlakoztatható hozzá) • A pumpán egy szerelő kezében egy másik cső vége <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A szerelő megpróbálja felcsatlakoztatni a nála lévő cső végét <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • A kimenet megegyezik a bemenettel
Teszt célja	Megbizonyosodni, arról hogy egy szerelő nem képes felcsatlakoztatni egy csövet egy olyan pumpához, amihez már maximális számú cső van csatlakoztatva. Teszteli a Plumber osztály <i>attach</i> és Pump <i>addPipe</i> metódusokat.

Teszt-eset neve	22. Acquire Pump
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol egy ciszterna van és rajta egy szerelő <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A szerelő felvesz egy új pumpát <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • Egy ciszterna rajta egy szerelővel, a szerelőnél egy puma
Teszt célja	Megbizonyosodni, arról hogy egy szerelő képes új pumpát felvenni egy ciszternánál. Teszteli a Plumber <i>pickupPump</i> és a Cistern <i>acquirePump</i> metódusokat.

Teszt-eset neve	23. Place Pump
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa hozzájuk csatlakoztatva egy cső szerepel és egy szerelő áll a csövön, és van a kezében egy pumpa <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A szerelő lerakja a nála lévő pumpát <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • A két eredeti pumpához hozzácsatlakoztatva egy-egy, de különböző, csővég • Az új pumpa hozzácsatlakoztatva a két csőhöz • A szerelő az új pumpán áll
Teszt célja	Megbizonyosodni, hogy egy szerelő képes lerakni a kezéből a soron következő pumpát a csőre, amin áll. Teszteli a Plumber <i>placePump</i> , Pipe <i>split</i> metódusokat.

Teszt-eset neve	24. Unable to Place Pump
------------------------	--------------------------

Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa hozzájuk csatlakoztatva egy cső rajta egy szerelő szerepel, üres kézzel <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő megpróbálja lehelyezni a pumpát a csőre <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> A kimenet megegyezik a bemenettel
Teszt célja	Megbizonyosodni, arról hogy egy szerelő nem képes lehelyezni pumpát, ha nincsen nála pumpa. Teszteli a Plumber <i>placePump</i> metódust.

Teszt-eset neve	25. Move to unoccupied Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa hozzájuk csatlakoztatva egy cső szerepel és egy szerelő áll az első pumpán <p>Teszt futása:</p> <ol style="list-style-type: none"> A szerelő átlép a csőre <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> 2 pumpa hozzájuk csatlakoztatva egy cső rajta egy szerelő szerepel
Teszt célja	Megbizonyosodni arról, hogy egy karakter képes átlépni egy olyan csőre, amin nem áll senki. Teszteli a Character <i>moveTo</i> , Element <i>leave</i> , Pipe <i>enter</i> metódusokat.

Teszt-eset neve	26. Move to occupied Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa hozzájuk csatlakoztatva egy cső és 2 szerelő Az egyik szerelő az egyik pumpán áll, a másik a csövön <p>Teszt futása:</p> <ol style="list-style-type: none"> Az 1. szerelő megpróbál átlépni a csőre <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> A kimenet megegyezik a bemenettel
Teszt célja	Megbizonyosodni arról, hogy egy karakter nem képes átlépni egy olyan csőre, amin már állnak. Teszteli a Character <i>moveTo</i> , Pipe <i>enter</i> metódusokat.

Teszt-eset neve	27. Move to detached Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol egy pumpáhpz van csatlakoztatva egy csőnek az egyik vége, a cső másik vége nincs hozzá csatlakoztatva semmihez sem.

	<ul style="list-style-type: none"> • A pumpán egy szerelő áll. <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A szerelő megpróbál átlépni a csőre <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • A kimenet megegyezik a bemenettel
Teszt célja	Megbizonyosodni arról, hogy egy karakter nem képes átlépni egy olyan csőre, aminek legalább az egyik vége le van csatlakoztatva. Teszteli a Character <i>moveTo</i> , Pipe <i>enter</i> metódusokat.

Teszt-eset neve	28. Move to Pump
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa hozzájuk csatlakoztatva egy cső rajta egy szerelő szerepel <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A szerelő átlép az 1. pumpára <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • 2 pumpa köztük egy cső, az 1. pumpán áll a szerelő
Teszt célja	Megbizonyosodni arról, hogy egy karakter képes rálépni egy pumpára egy csőről. Teszteli a Character <i>moveTo</i> , Pipe <i>leave</i> , Pump <i>enter</i> metódusokat.

Teszt-eset neve	29. Move to Cistern
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol egy pumpa és egy ciszterna hozzájuk csatlakoztatva egy cső rajta egy szerelő szerepel <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A szerelő átlép a ciszternára <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • egy pumpa és egy ciszterna hozzájuk csatlakoztatva egy cső, a szerelő a ciszternán áll
Teszt célja	Megbizonyosodni arról, hogy egy karakter képes rálépni egy ciszternára egy csőről. Teszteli a Character <i>moveTo</i> , Pipe <i>leave</i> , Cistern <i>enter</i> metódusokat.

Teszt-eset neve	30. Move to Spring
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol egy pumpa és egy forrás hozzájuk csatlakoztatva egy cső rajta egy szerelő szerepel <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A szerelő átlép a forrásra <p>Elvárt kimenet:</p>

	<ul style="list-style-type: none"> • egy pumpa és egy forrás hozzájuk csatlakoztatva egy cső, a szerelő a forráson áll
Teszt célja	Megbizonyosodni arról, hogy egy karakter képes rálépní egy forrásra egy csőről. Teszteli a Character <i>moveTo</i> , Pipe <i>leave</i> , Spring <i>enter</i> metódusokat.

Teszt-eset neve	31. Move to banana Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa hozzájuk csatlakoztatva egy cső szerepel és egy szerelő áll az első pumpán • A cső csúszóssá van téve • A csúsztató értékét beállítjuk hogy a 2. pumpára csússzon <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A szerelő átlép a csőre 2. A szerelő tovább csúszik az egyik, véletlenszerűen kiválasztott pumpára (ezesetben a 2. pumpára) <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • 2 pumpa hozzájuk csatlakoztatva egy cső, a 2. pumpán egy szerelő áll
Teszt célja	Megbizonyosodni arról, hogy amikor valaki egy csúszós csőre lép, akkor egy véletlenszerűen kiválasztott, a csőhöz csatlakoztatott pumpára lép tovább. Teszteli a Character <i>moveTo</i> , Pipe <i>enter</i> , Pump <i>enter</i> metódusokat.

Teszt-eset neve	32. Move through technokoled Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol 2 pumpa hozzájuk csatlakoztatva egy cső szerepel és egy szerelő áll az első pumpán • A cső ragadóssá van téve <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A szerelő átlép a csőre 2. A szerelő megpróbál továbblépni a 2. pumpára <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • 2 pumpa hozzájuk csatlakoztatva egy cső rajta egy szerelő szerepel
Teszt célja	Megbizonyosodni arról, hogy egy karakter amikor rálép egy ragadós csőre, addig nem tud lelépni róla vagy más akciót végrehajtani, amíg a ragadás el nem műlik. Teszteli a Pipe <i>enter</i> metódust.

Teszt-eset neve	33. Break Pump
Rövid leírás	Inicializálás:

	<ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol egy nem törött pumpa szerepel <p>Teszt futása:</p> <ol style="list-style-type: none"> A Controller eltöri a pumpát <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> Egy törött pumpa
Teszt célja	Megbizonyosodni arról, hogy a Controller hívására a pumpa képes elromlani. Teszteli a Pump <i>break</i> metódust.

Teszt-eset neve	34. Spawn Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol egy ciszterna szerepel <p>Teszt futása:</p> <ol style="list-style-type: none"> A Controller a ciszternát megkéri, hogy csináljon egy új csövet magára <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> 2 pumpa hozzájuk csatlakoztatva egy cső rajta egy szerelő szerepel
Teszt célja	Megbizonyosodni arról, hogy a Controller hívására új cső keletkezik a ciszternánál. Teszteli a Cistern <i>spawnPipe</i> metódust.

Teszt-eset neve	35. Waterflow from Spring
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol egy forrás és egy pumpa van hozzájuk csatlakoztatva egy üres cső <p>Teszt futása:</p> <ol style="list-style-type: none"> A Controller szól a forrásnak, hogy follyon <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> egy forrás és egy pumpa van hozzájuk csatlakoztatva egy vízzel töltött cső
Teszt célja	Megbizonyosodni arról, hogy a forrásból képes víz folyni a szomszédos csövekbe. Teszteli a Spring <i>tick</i> metódust.

Teszt-eset neve	36. Waterflow through Pump
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> Betöltünk a tesztprogramba egy pályát, ahol 3 pumpa és 2 cső van. Ebből kettőhöz egy-egy, de különböző, csővég, a 3.-hoz a két cső másik vége van csatlakoztatva. Az 1. csőben van víz és a 3. pumpa bemeneti csövének van beállítva A 2. csőben nincs víz és a 3. pumpa kimeneti csövének van beállítva <p>Teszt futása:</p>

	<p>1. A Controller szól a 3. pumpának, hogy folyjon át rajta a víz</p> <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • 3 pumpa 2 cső • Az 1. csőben nincs víz és a 3. pumpa bemeneti csövének van beállítva • A 2. csőben van víz és a 3. pumpa kimeneti csövének van beállítva
Teszt célja	Megbizonyosodni arról, hogy egy pumpa képes vizet kipumpálni bemeneti csőből egy kimeneti csőbe. Teszteli a Pump <i>tick</i> metódust.

Teszt-eset neve	37. Waterflow into Cistern
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol van egy ciszterna, és egy cső bele van kötve <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A Controller szól a ciszternának, hogy folyjon <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • egy ciszterna és egy hozzá csatlakoztatott, üres cső • A ciszternában van a csőből kifolyt vízmennyiség
Teszt célja	Megbizonyosodni arról, hogy a víz képes befolyni a csövekből egy szomszédos ciszternánál. Teszteli a Cistern <i>tick</i> metódust.

Teszt-eset neve	38. Waterflow into full Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol egy forrás és egy pumpa van, és hozzájuk csatlakoztatva egy vízzel teli cső. <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A Controller szól a forrásnak, hogy folyasson vizet. <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • A kimenet megegyezik a bemenettel
Teszt célja	Megbizonyosodni arról, hogy nem képes több víz befolyni egy teli csőbe. Teszteli a Pipe <i>addWater</i> , Spring <i>tick</i> metódusokat.

Teszt-eset neve	39. Waterflow into detached Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol egy forrás van, és hozzá csatlakoztatva egy üres cső, aminek be van állítva a földje. <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A Controller szól a forrásnak, hogy folyasson vizet.

	<p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • A földön van a csőbe belefolyatott víz.
Teszt célja	Megbizonyosodni arról, hogy a víz amikor befolyik egy lecsatlakoztatott végű csőbe, akkor az kifolyik a földre. Teszteli a Pipe <i>addWater</i> és a WaterCollector <i>storeAmount</i> metódust.

Teszt-eset neve	40. Waterflow into pierced Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol egy forrás és egy pumpa van, és hozzájuk csatlakoztatva egy üres, lyukas cső, aminek be van állítva a földje. <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A Controller szól a forrásnak, hogy folyasson vizet. <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • A földön van a csőbe belefolyatott víz.
Teszt célja	Megbizonyosodni arról, hogy a víz amikor befolyik egy kilyukasztott csőbe, akkor az kifolyik a vízre. Teszteli a Pipe <i>addWater</i> és a WaterCollector <i>storeAmount</i> metódust.

Teszt-eset neve	41. Tick Banana Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol egy csúszós cső van • A csúszósság értékét beállítjuk, hogy 1 idő műlás után eltűnjön <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A Controller jelzi a csőnek, hogy múlt az idő <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • Egy nem csúszós (normális) cső
Teszt célja	Megbizonyosodni arról, hogy egy cső csúszós állapota az idő műlásával megszűnik. Teszteli a Pipe <i>tick</i> és a BananaEffect <i>tick</i> metódust.

Teszt-eset neve	42. Tick Technokoled Pipe
Rövid leírás	<p>Inicializálás:</p> <ul style="list-style-type: none"> • Betöltünk a tesztprogramba egy pályát, ahol egy ragadós cső van • A ragadósság értékét beállítjuk, hogy 1 idő műlás után eltűnjön <p>Teszt futása:</p> <ol style="list-style-type: none"> 1. A Controller jelzi a csőnek, hogy múlt az idő <p>Elvárt kimenet:</p> <ul style="list-style-type: none"> • Egy nem ragadós (normális) cső
Teszt célja	Megbizonyosodni arról, hogy egy cső ragadós állapota az idő műlásával megszűnik.

	Teszteli a Pipe <i>tick</i> és a TechnokolEffect <i>tick</i> metódust.
--	--

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A teszteléshez saját segéd-programot és tesztelést leíró fájlokat használunk. (lásd: 7.4.1)

7.4.1 Tesztelést támogató segéd-program:

- A program egy fájl-elérési-útvonalat vár. Ezt megadva lefuttatja a kapott tesztelési fájlból található parancsokat. Amennyiben a megadott elérési út egy mappához tartozik, úgy lefuttatja az összes mappában lévő ‘*pasta*’ kiterjesztéssel rendelkező fájl tesztjét.
- A tesztelési fájl első sorában egy pályaleíró fájl elérési útvonalát, második sorában a teszt elvárt kimenetét leíró fájl elérési útvonalát tartalmazza. Ezután következnek a tényleges tesztelési utasítások.
- A program az adott teszt futtatása során betölti a fájlból leírt kezdeti pályát, majd a tesztelt program bemenetére irányítja a tesztfájlból lévő parancsokat.
- A program az adott teszt futása végén fájlba írja (az eredeti tesztfájl mappájába) a program állapotát. A kiírt fájl neve megegyezik az eredeti tesztleíró fájl nevével, kiterjesztése ‘*.out.pasta*’.
- A teszt(ek) futtatása után a program összehasonlítja a tesztek kimenetét az elvárt kimenetükkel. Ezután kiírja a standard kimenetére a teszt(ek) eredményét. Sikertelen teszt esetén a teszthez tartozó kimeneti fájl első elvárttól eltérő sorát és sorszámát is kiírja.

7.5 Napló

Kezdet	Időtartam	Résznevők	Leírás
2023.04.19. 21:00	3 óra	Harsányi Kereszthury Tóth	Értekezlet: Értelmeztük a heti feladatokat, megoldásokat kerestünk a módosításokra. Döntés: Harsányi a 7.3, 7.4 Kereszthury a 7.1, 7.2 Szilczer a 7.0 feladatokat végzi el.
2023.04.20. 18:00	2 óra	Kereszthury	7.1 parancsok írása
2023.04.21. 17:00	3 óra	Kereszthury	7.1 befejezése
2023.04.20. 19:00	4.5 óra	Harsányi	7.3 1-21.ig
2023.04.21. 20:00	3.5 óra	Kereszthury	7.2 elkészítése
2023.04.22. 19:00	5 óra	Szilczer	7.0.1 elkészítése
2023.04.23. 02:00	2 óra	Galgóczy	Csúszás és ragasztás logikájának kitalálása
2023.04.23. 16:00	2 óra	Szilczer	7.0.3 készítése
2023.04.23. 18:00	4 óra	Galgóczy	Csúszás és ragasztás logikai beillesztése a modellbe
2023.04.23. 21:00	4.5 óra	Harsányi	7.3 22-42. ig
2023.04.23. 22:00	6 óra	Szilczer	7.0.3 elkészítése
2023.04.24. 2:00	1 óra	Harsányi	7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása
2023.04.24. 2:00	2 óra	Tóth	7.3. 1-22-ig átnézése
2023.04.24. 10:00	3 óra	Szilczer	7.0.2 elkészítése
2023.04.24. 11:00	1 óra	Tóth	7.1, 7.2 átnézése, javítása
2023.04.24. 12:00	1 óra	Galgóczy	7.4 megírása
2023.04.24. 12:00	1 óra	Tóth	7.3. 22-42-ig átnézése, javítása

8. Részletes tervez

8.1 Módosított prototípus interface-definíció

8.1.1 Az interfész általános leírása

A parancsok a szabványos bemenetre érkeznek, ami történhet terminálból, illetve előre megírt teszesetek esetén fájlból. A program kimenete a szabványos kimenetre íródik ki.

A proto képes adott konfigurációs (pályaleíró) fájlokat is beolvasni valamint előállítani.
(A konfigurációs fájl pontos formátuma a 8.1.2 fejezet végén van kifejtve)

8.1.2 Bemeneti nyelv

A program szabványos bemenetén lehet parancsokat megadni. A parancsok után következő paraméterek space-ekkel vannak elválasztva, a parancsfeldolgozás első szava case insensitive.

A programban létrehozott objektumoknak nevet kell adni, a nevek azonban csak az angol abc betűit és számjegyeket tartalmazhatnak.

A program által elfogadott parancsok:

A relációs jelek között szereplő paraméterek kötelezők, a szögletes zárójelben lévők opcionálisak, a paraméterek sorrendje kötött. A kettőspontok után az értelmezett paraméterek típusa szerepel.

‘Create’ parancsok esetén különböző pályaelemeket és karaktereket lehet létrehozni, melyeknek kötelező egyedi nevet adni.

Create pump <name: string> [ports: int] [tankCapacity: int]

Leírás:

Létrehoz a pályán egy pumpát.

Paraméterek:

- name: a pumpa neve
- ports: a pumpa csatlakozóinak száma (maximum ennyi cső köthető rá), alapértelmezett értéke 4
- tankCapacity: a pumpa tartályának kapacitása, alapértelmezett értéke 5

Create collector <name: string>

Leírás:

Létrehoz egy vízgyűjtőt (WaterCollector-t).

Paraméterek:

- name: a vízgyűjtő neve

Create pipe <name: string> <ground: WaterCollector> [capacity: int]

Leírás:

Létrehoz a pályán egy csövet.

Paraméterek:

- name: a cső neve
- ground: a vízgyűjtő neve, amit a cső földként fog kezelni. Megadáskor már léteznie kell.
- capacity: a cső kapacitása, alapértelmezett értéke 10

Create cistern <name: string> <collector: WaterCollector>

Leírás:

Létrehoz a pályán egy ciszternát.

Paraméterek:

- name: a ciszterna neve
- collector: a vízgyűjtő neve, amit a ciszterna használni fog. Megadáskor már léteznie kell.

Create spring <name: string> [waterDispensedPerTick: int]**Leírás:**

Létrehoz a pályán egy forrást.

Paraméterek:

- name: a forrás neve
- waterDispensedPerTick: a forrásból a csövekbe egy időegység alatt kifolyó vízmennyiség, alapértelmezett értéke 4

Create plumber <name:string> <startLocation: Element>**Leírás:**

Létrehoz egy szerelőt.

Paraméterek:

- name: a szerelő neve
- startLocation: a szerelő kiinduló pozíciójának (pályaelem) neve. Ennek a pályaelemnek már léteznie kell.

Create saboteur <name:string> <startLocation: Element>**Leírás:**

Létrehoz egy szabotőrt.

Paraméterek:

- name: a szabotőr neve
- startLocation: a szabotőr kiinduló pozíciójának (pályaelem) neve. Ennek a pályaelemnek már léteznie kell.

Attach <pipe: Pipe> <element: ActiveElement>**Leírás:**

Hozzáköti a paraméterként kapott csövet a paraméterként kapott aktív elemhez.

Paraméterek:

- pipe: a cső neve
- element: az aktív elem neve

Detach <pipe: Pipe> <element: ActiveElement>**Leírás:**

Megszünteti a paraméterként kapott cső és aktív elem közötti csatlakozást.

Paraméterek:

- pipe: a cső neve
- element: az aktív elem neve

List**Leírás:**

Kilistázza az összes létrehozott pályaelem és karakter nevét és típusát.

View <pump: Pump> [attributeName: string]**Leírás:**

Kiírja a paraméterként kapott pumpa attribútumait.

Paraméterek:

- pump: a pumpa neve
- attributeName: amennyiben meg van adva, csak a megadott nevű attribútum az összes attribútum helyett, az objektum ‘neve: típusa’ a szokásos módon íródik ki

View <collector: WaterCollector> [attributeName: string]**Leírás:**

Kírja a paraméterként kapott vízgyűjtő attribútumát.

Paraméterek:

- collector: a vízgyűjtő neve neve
- attributeName: amennyiben meg van adva, csak a megadott nevű attribútum az összes attribútum helyett, az objektum ‘neve: típusa’ a szokásos módon íródik ki

View <pipe: Pipe> [attributeName: string]**Leírás:**

Kírja a paraméterként kapott cső attribútumait.

Paraméterek:

- pipe: a cső neve
- attributeName: amennyiben meg van adva, csak a megadott nevű attribútum az összes attribútum helyett, az objektum ‘neve: típusa’ a szokásos módon íródik ki

View <cistern: Cistern> [attributeName: string]**Leírás:**

Kírja a paraméterként kapott ciszterna attribútumait.

Paraméterek:

- pump: a pumpa neve
- attributeName: amennyiben meg van adva, csak a megadott nevű attribútum az összes attribútum helyett, az objektum ‘neve: típusa’ a szokásos módon íródik ki

View <spring: Spring> [attributeName: string]**Leírás:**

Kírja a paraméterként kapott forrás attribútumait.

Paraméterek:

- pump: a forrás neve
- attributeName: amennyiben meg van adva, csak a megadott nevű attribútum az összes attribútum helyett, az objektum ‘neve: típusa’ a szokásos módon íródik ki

View <plumber: Plumber> [attributeName: string]**Leírás:**

Kírja a paraméterként kapott szerelő attribútumait.

Paraméterek:

- plumber: a szerelő neve
- attributeName: amennyiben meg van adva, csak a megadott nevű attribútum az összes attribútum helyett, az objektum ‘neve: típusa’ a szokásos módon íródik ki

View <saboteur: Saboteur> [attributeName: string]**Leírás:**

Kírja a paraméterként kapott szabotőr attribútumát.

Paraméterek:

- saboteur: a szabotőr neve
- attributeName: amennyiben meg van adva, csak a megadott nevű attribútum az összes attribútum helyett, az objektum ‘neve: típusa’ a szokásos módon íródik ki

Load <path: string>**Leírás:**

Betölti az útvonalon megadott konfigurációs (pályát és karaktereket definiáló) fájlt.
A betöltés előtt létrehozott objektumok (elemek és karakterek) megszűnnek létezni.

Paraméterek:

- path: a fájl elérési útja

Save <path: string>**Leírás:**

Elmenti az útvonalon megadott konfigurációs (pályát és karaktereket definiáló) fájlba a létrehozott objektumokat.

Paraméterek:

- path: a fájl elérési útja

Help [command: string]**Leírás:**

A command paraméter nélkül kiírja az elérhető parancsok neveit. Amennyiben a command paraméter meg van adva, az adott parancs paramétereit írja ki.

Paraméterek:

- command: a parancs neve, amiről információ kéne

Tick [element: Element]**Leírás:**

Meghívja a paraméterek adott elem tick() függvényét. Amennyiben nincs ez a paraméter megadva, az összes létrehozott elemen meghívja.

Paraméterek:

- element: a pályaelem neve, amin külön kell meghívni a tick()-et

Break <pump: Pump>**Leírás:**

Eltörí a paraméterként kapott pumpát.

Paraméterek:

- pump: a pumpa neve, amit el kell törni

SpawnPipe <cistern: Cistern> <ground: WaterCollector> <pipeName: string>**Leírás:**

Létrehoz egy új csövet a paraméterként kapott ciszternán.

Paraméterek:

- cistern: a ciszterna neve, amin csövet kell létrehozni
- ground: az a vízgyűjtő, ami a létrehozott csőben a föld lesz
- pipeName: a létrejövő cső neve (a create parancs név követelményeinek megfelelően)

Move <character: Character> <element: Element>**Leírás:**

A paraméterként kapott karaktert a paraméterként kapott pályaelemre mozgatja, amennyiben az lehetséges.

Paraméterek:

- character: a karakter neve, akit mozgatni kell

- element: a pályaelem neve, amire a karaktert mozgatni kell

Set input <character: Character> <pump: Pump> <pipe: Pipe>

Leírás:

A paraméterként kapott pumpa bemeneti csövének állítja a kapott csövet a kapott karakter (amennyiben lehetséges).

Paraméterek:

- character: a karakter neve, aki a pumpát állítja
- pump: a pumpa neve, amit állítani kell
- pipe: a beállítandó bemeneti cső

Set output <character: Character> <pump: Pump> <pipe: Pipe>

Leírás:

A paraméterként kapott pumpa kimeneti csövének állítja a kapott csövet a kapott karakter (amennyiben lehetséges).

Paraméterek:

- character: a karakter neve, aki a pumpát állítja
- pump: a pumpa neve, amit állítani kell
- pipe: a beállítandó kimeneti cső

Pierce <character: Character> <pipe: Pipe>

Leírás:

A paraméterként kapott karakter kilyukasztja a paraméterként kapott csövet (ha lehetséges).

Paraméterek:

- character: a karakter neve, aki a csövet töri
- pipe: a cső, amit el kell törni

Technokol <character: Character> <pipe: Pipe> [technokolTime: int]

Leírás:

A paraméterként kapott karakter ragadóssá tudja tenni a paraméterként kapott csövet (ha lehetséges).

Paraméterek:

- character: a karakter neve, aki ragaszt
- pipe: a cső, amit össze kell ragasztózní
- technokolTime: az idő, amíg a cső ragasztós marad, alapértelmezett értéke véletlenszerű

Banana <saboteur: Saboteur> <pipe: Pipe> [bananaTime: int] [slideBack: boolean]

Leírás:

A paraméterként kapott szabotőr csúszóssá tudja tenni a paraméterként kapott csövet (ha lehetséges).

Paraméterek:

- saboteur: a szabotőr neve, aki síkosít
- pipe: a cső, amit síkosítani kell
- bananaTime: az idő, amíg a cső csúszós marad, alapértelmezett értéke véletlenszerű
- slideBack: igaz, ha a csúszós csőre rálépő karakter kiindulási pontjára csússzon vissza, hamis ha a másik irányba, alapértelmezett értéke véletlenszerű

Repair <plumber: Plumber> <pipe: Pipe> [invulnerability: int]**Leírás:**

A paraméterként kapott szerelő meg tudja javítani a paraméterként kapott csövet (ha lehetséges).

Paraméterek:

- plumber: a szerelő neve, aki javít
- pipe: a cső, amit javítani kell
- invulnerability: az idő, amíg a cső törhetetlen marad, alapértelmezett értéke véletlenszerű

Repair <plumber: Plumber> <pump: Pump>**Leírás:**

A paraméterként kapott szerelő meg tudja javítani a paraméterként kapott pumpát (ha lehetséges).

Paraméterek:

- plumber: a szerelő neve, aki javít
- pump: a pumpa, amit javítani kell

Attach <plumber: Plumber> <element: ActiveElement>**Leírás:**

A paraméterként kapott szerelő rácslakoztatja a paraméterként kapott aktív elemre a kezében tartott csövet (ha lehetséges).

Paraméterek:

- plumber: a szerelő neve
- element: az aktív elem, amire csatlakoztat

Detach <plumber: Plumber> <element: ActiveElement> <pipe: Pipe>**Leírás:**

A paraméterként kapott szerelő lecsatlakoztatja a paraméterként kapott aktív elemről a paraméterként kapott csövet (ha lehetséges).

Paraméterek:

- plumber: a szerelő neve
- element: az aktív elem, amire csatlakoztat
- pipe: a lecsatlakoztatni kívánt cső

GetPump <plumber: Plumber> <cistern: Cistern> <pumpName: string>**Leírás:**

A paraméterként kapott szerelő felvesz egy pumpát a paraméterként kapott ciszternáról (ha rajta áll).

Paraméterek:

- plumber: a szerelő neve
- cistern: a ciszterna, amin a szerelő áll
- pumpName: a létrejött pumpa neve (a create parancs név követelményeinek megfelelően)

PlacePump <plumber: Plumber> <pipe: Pipe> <newPipeName: string>**Leírás:**

A paraméterként kapott szerelő lehelyez egy pumpát a paraméterként kapott csőre (ha rajta áll).

Paraméterek:

- plumber: a szerelő neve

- pipe: a cső, amin a szerelő áll
- newPipeName: az újonnan létrejövő cső neve (a create parancs név követelményeinek megfelelően)

A konfigurációs/pálya fájlok (kimeneti és bemeneti) nyelvtana:

Egymás után felsorolva szerepelnek a különböző objektumok (karakterek és pályaelemek) részletesen kifejtve (objektum neve, típusa és attribútumai név-érték hozzárendeléssel) a következő módon:

[pályaelem vagy karakter neve]: [pályaelem vagy karakter típusa]

[attribútum neve] = [attribútum értéke]

[attribútum neve] = [attribútum értéke]

[...] = [...]

[második pályaelem vagy karakter neve]

...

Az attribútum bejegyzések tetszőleges sorrendben szerepelhetnek a fájlban, előttük tabulátor szerepel. Az asszociációkat az objektumok nevei reprezentálják, ha nincs beállítva egy asszociáció, akkor az egyenlőségjel után "null" szöveg szerepel. Többes asszociáció esetén (lista/tömb) az attribútum érték helyére az asszociált objektumok nevei kerülnek vesszővel elválasztva [] karakterek közé.

Ha egy attribútum nem jelenik meg a bemeneti konfigurációs fájlban, az alapértelmezett értékkel lesz inicializálva.

8.1.3 Kimeneti nyelv

Érvénytelen vagy hibás formátumú parancs esetén a program az alábbi hibaüzenettel jelez vissza a felhasználónak: "! unknown command, use 'help' to see the full list of commands".

Hibás paraméterezésű parancs esetén a program hibaüzenetet ír ki a felhasználó számára:

"! invalid parameter found in command"

Hiányzó paraméter esetén: "! missing parameter" üzenet jelenik meg.

Parancsok kimenetei:

Create: Amennyiben a létrehozott pályaelem neve foglalt, a program a következőt írja ki: "! object with name already exists", különben nem ír ki semmit.

Attach: Ha az összekötést nem a szerelő végzi és sikertelen, a program a következőt írja ki: "! unable to make connection", különben nem ír ki semmit.

List: Kilistázza az összes objektum nevét (pályaelemet és karaktert egyaránt) Az azonos típusú objektumok egy sorban, a sor elején típusuk jelenik meg. (pl.: "Pipe: pipe1, pipe2, ...")

View: Kiíratja egy adott pályaelem vagy karakter adatait (belso attribútumait) külön sorokba a következő formátumban (külön választott attribútum esetén csak azt):

Az első sorban a pályaelem / karakter neve és típusa szerepel kettősponttal elválasztva

A következő sorokban a pályaelem / karakter attribútumai jelennek meg:

"<attribútum neve> = <attribútum értéke>"

Az attribútumok sorainak elején egy tabulátor karakter szerepel, lista esetén a lista elemei szögeletes zárójelek között szerepelnek, egymástól vesszővel vannak elválasztva. Üres lista esetén csak a szögeletes zárójelek szerepelnek.

Asszociációk esetén az adott objektum(ok) neve(i) van(nak) kiírva.

Ha nincs asszociáció (null), a kimenetben az egyenlőségjel után "null" szöveg szerepel.

Az egyes osztályok attribútumai az alábbi sorrendben jelennek meg:

- Pump: ports, tankCapacity, tankStoredWater, broken, inputPipe, outputPipe, connectedPipes, neighbours
- WaterCollector: storedAmount
- Pipe: effect (Banana, Technokol or None), endpoints, occupied, pierced, pierceCooldown, ground, capacity, storedWater, neighbours
- Cistern: newPipes, collector, connectedPipes, neighbours
- Spring: waterDispensedPerTick, connectedPipes, neighbours
- Plumber: heldPipe, heldPumps, location
- Saboteur: location

Ha nem létezik a megadott nevű objektum, a program a következőt írja ki: “! object not found”.

Ha a parancsban opcionálisan megadott attribútumnév a kiíratott osztályban nem szerepel, a program a következőt írja ki: “! attribute not found”

Load: Amennyiben a betöltés sikertelen, értesíti a felhasználót: “! cannot load file”

Ha a fájl nem létezik a “! file not found” üzenet jelenik meg.

Sikerességtől esetén a “load successful” üzenet jelenik meg.

Save: Amennyiben a kiírás sikertelen, értesíti a felhasználót: “! cannot save file”

Sikerességtől esetén a “save successful” üzenet jelenik meg. A program létrehoz egy új fájlt a paraméterként kapott helyen.

Help: A command paraméter nélkül kiírja az elérhető parancsok neveit egy sorba, vesszővel és space-el elválasztva. Amennyiben a command paraméter meg van adva, az adott parancs paramétereinek kiírását írja ki. (A 8.1.2 fejezetben szereplő formában.)

Ha nincsen a command paraméternek megfelelő parancs, a paramétert nem veszi figyelembe.

8.2 Osztályok és metódusok tervez

8.2.1 ActiveElement

• Felelősség

Aktív elem. Összeköti a csöveget, folyatja a hálózatban a vizet, csövet lehet felcsatlakoztatni rá és lecsatlakoztatni rólá. Absztrakt osztály.

• Ősosztályok

Element → ActiveElement

• Attribútumok

- #Pipe[0..*] **connectedPipes**: A hozzá csatlakoztatott csövek halmaza.

• Metódusok

- +bool **addPipe(Pipe pipe)**: Hozzáad egy csövet. Ha már hozzá van csatlakoztatva a kapott cső, vagy a folyamat közben ezen csövön meghívott *addEndpoint* függvény *false*-szal tér vissza, akkor nem képes végrehajtani a kívánt műveletet, és *false*-szal tér vissza (egyéb esetben *true*-val). Ha sikerült a csatlakoztatás, akkor a *neighbours* listát is frissíti.

- Pszeudokód:

függvény addPipe(Pipe pipe): bool

ha még nincs hozzá csatlakoztatva **és** hozzáadható a csőhöz egy új végpont

pipe hozzáadása a csatlakoztatott csövekhez

pipe hozzáadása a szomszédokhoz

térj vissza igazzal

különben:

térj vissza hamissal

- **+bool removePipe(Pipe pipe):** Megpróbálja eltávolítani magáról a paraméterként kapott csövet. Ha nincs hozzácsatlakoztatva a megadott cső, vagy a folyamat közben a csövön meghívott *removeEndpoint* függvény *false*-szal tér vissza, *false*-szal tér vissza. Sikeres esetben *true*-val tér vissza. Ha sikerült a lecsatlakoztatás, akkor a *neighbours* listát is frissíti.

○ Pszeudokód:

függvény removePipe(Pipe pipe): bool

ha hozzá van csatlakoztatva **és** a csőnek eltávolítható az adott végpontja:

pipe eltávolítása a csatlakoztatott csövekből

pipe eltávolítása a szomszédokból

térj vissza igazzal

különben:

térj vissza hamissal

8.2.2 BananaEffect

- **Felelősség**

A banánozottságot megvalósító effekt. Ha valaki a csőre lép amikor ez az effekt aktív, a cső egyik végére csúszik.

- **Ősosztályok**

Effect → BananaEffect

- **Attribútumok**

● **- int countdown:** Az effekt hátralévő ideje. Ha nagyobb mint 0, az effekt aktív.

- **Metódusok**

● **+ boolean enter(Character character):** A karaktert a cső egy véletlenszerű végpontjára helyezi át.

● **+ boolean leave():** Mindig *true*-val tér vissza, csúszós csőről bármikor le lehet lépni.

● **+ void tick():** Csökkenti a *countdown* attribútum értékét, ha az eléri a 0-t, kicseréli a csövön lévő effektet *NoEffect*-re.

8.2.3 Character

- **Felelősség**

Az elemeken mozog, számon tartja jelenlegi tartózkodási helyét. Állítgathatja a pumpák kimeneti és bemeneti csövét. Kilyukaszthatja a csöveket, és össze is Technokolozhatja őket. Absztrakt osztály.

- **Ősosztályok**

Nincs ősosztálya.

- **Attribútumok**

- **#Element location:** A hely, ahol a karakter áll.

- **Metódusok**

- **+void moveTo(Element destination):** Megpróbál rálépni a kapott célhelyre.
- **+bool leave(to: Element):** Megkísérle lelépni a helyéről, ha ez sikeres volt, átlép a kapott célhelyre. Visszaadja, hogy sikeres volt-e a lelépés.
- **+void setInputPipe(Pump pump, Pipe pipe):** Megpróbálja beállítani a paraméterként kapott pumpa bemeneti csövét a paraméterként kapott csőre, ha a pumpán áll.
- **+void setOutputPipe(Pump pump, Pipe pipe):** Megpróbálja beállítani a paraméterként kapott pumpa kimeneti csövét a paraméterként kapott csőre, ha a pumpán áll.
- **+void pierce(Pipe pipe):** Megpróbálja kilyukasztani a paraméterként kapott csövet, ha rajta áll.
- **+void applyTechnokol(Pipe pipe):** A paraméterként kapott cső effektjét beállítja *TechnokolEffect*-re.

8.2.4 Cistern

- **Felelősség**

Vizet vesz el a rácsatlakoztatott (szomszédos) csövekből, és eltárolja azt. Kezeli a nála létrejött új csöveket. Pumpákat lehet felvenni nála.

- **Ősosztályok**

Element → ActiveElement → Cistern

- **Attribútumok**

- **-WaterCollector collector:** A vízgyűjtő, amiben a ciszterna a vizet gyűjti.
- **-Pipe[0..*] newPipes:** Az új csövek.

- **Metódusok**

- **+void tick():** A hozzácsatlakoztatott csövekből kiveszi az összes vizet, és eltárolja a *collector*-ban.
- **+bool removePipe(Pipe pipe):** Ha a kapott cső új cső, eltávolítja a csövet a *newPipes* listából, és true-val tér vissza. Egyéb esetben megpróbálja lecsatlakoztatni magáról a kapott csövet, és visszatér ennek a sikereségével.
- **+void spawnPipe():** Létrehoz egy új csövet, aminek egyik végét magához csatlakoztatja, és eltárolja a *newPipes* listában.
- **+Pump acquirePump():** Létrehoz egy új pumpát és visszaadja azt.

8.2.5 Effect

- **Felelősség**

Csőre való rálépései és/vagy lelépései viselkedést módosító osztályok absztrakt ősosztálya.

- **Ősosztályok**

Nincs ősosztálya.

- **Attribútumok**

- **# Pipe pipe:** A cső, amire az effekt hatással van.

- **Metódusok**

- + **boolean enter(Character character):** Absztrakt metódus, a karakter csőre való rálépését kezeli, *true* értéket ad vissza, ha a karakter ráléphet a csőre, *false*-t ha nem.
- + **boolean leave():** Absztrakt metódus, a karakter csőről való lelépését kezeli, *true* értéket ad vissza, ha a karakter leléphet a csőről, *false*-t ha nem.
- + **void tick():** Absztrakt metódus, az idő műlását szimulálva frissíti a csövet.

8.2.6 Element

- **Felelősség**

A csőrendszer eleme, a karakterek ezen képesek mozogni. Absztrakt osztály.

- **Ősosztályok**

Nincs ősosztálya.

- **Attribútumok**

- **#Element[0..*] neighbours:** Az elem szomszédjait tároló attribútum.

- **Metódusok**

- +**void enter(Element from):** Ellenőrzi, hogy a *from* benne van-e a *neighbours* listában, ha igen, a karaktert magára kíséri léptetni.
- +**bool leave():** Értesíti az elemet arról, hogy lelépett róla egy játékos és visszaadja azt, hogy sikeres volt-e a lelépés. Mindig igazat ad vissza.
- +**void tick():** Absztrakt metódus. Az idő műlását szimulálva lépteti az objektumot.

8.2.7 NoEffect

- **Felelősség**

Csövön lévő semmilyen effektet megvalósító osztály. Alapértelmezetten a csöveken ilyen effekt példány van, ez nem módosítja egyáltalán a karakter rálépést/lelépést.

- **Ősosztályok**

Effect → NoEffect

- **Metódusok**

- + **boolean enter(Character character):** *True* értéket ad vissza, ha a karakter jelenlegi helyéről való lelépés sikeres, egyébként *false*-t.
- + **boolean leave():** minden esetben *true*-t ad vissza.
- + **void tick():** Nem csinál semmit.

8.2.8 Pipe

- **Felelősség**

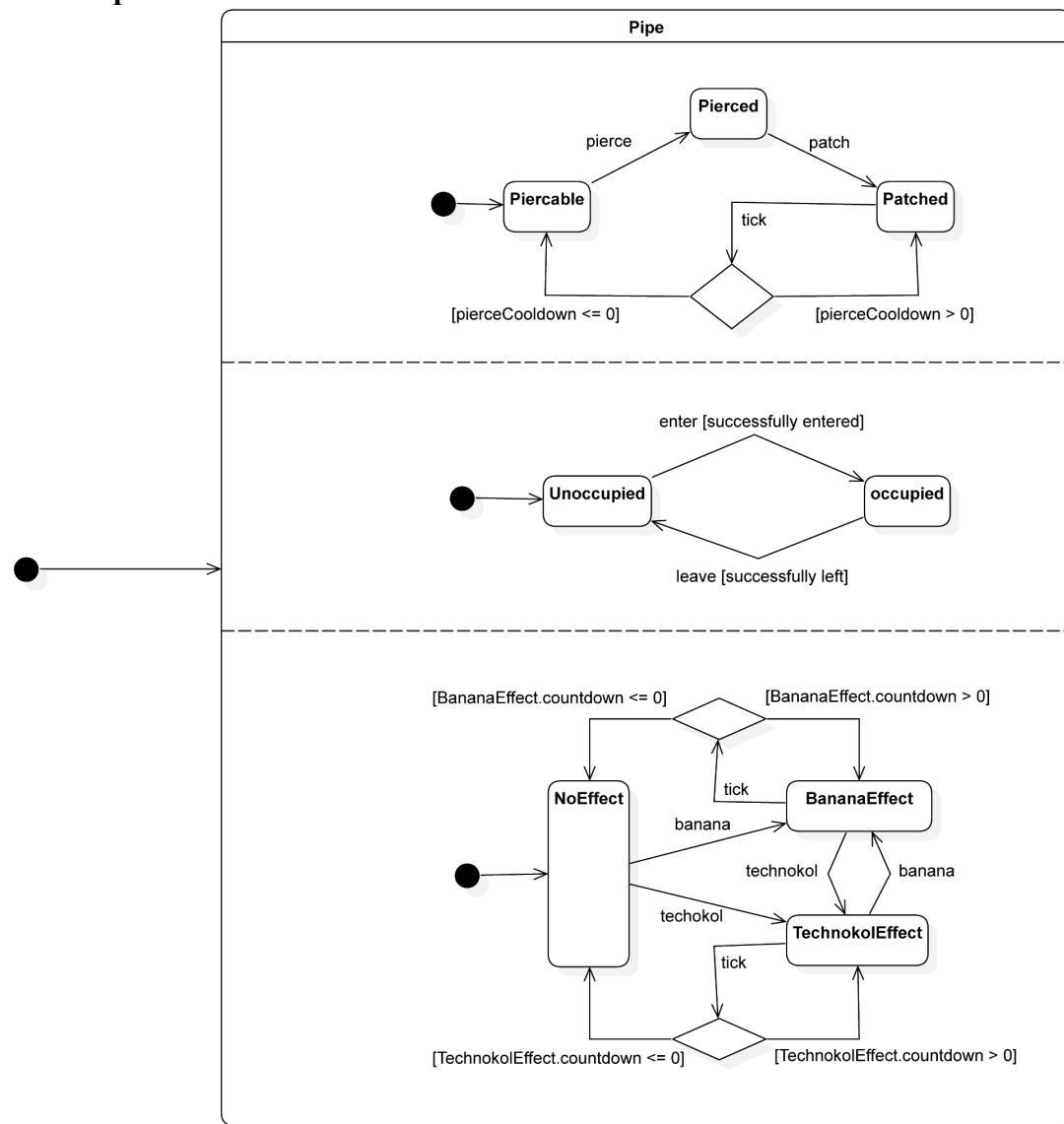
Játékbeli csövet leíró osztály. Egyes aktív elemek vizet helyezhetnek bele és szedhetnek ki belőle.

Egyszerre egy karakter állhat rajta. A szabotőrök kilyukaszthatják. A szerelők befolozhatják, átcsatlakoztathatják a végeit más aktív elemre, helyezhetnek el rajta pumpát.

- **Ősosztályok**

Element → Pipe

- **Állapotok**



A csövet csak a *Pierceable* állapotában lehet kilyukasztani, ha ez megtörténik átkerül a *Pierced* állapotba, és minden benne tárolt és a jövőben belekerülő víz a földre folyik, egész addig, amíg valaki meg nem javítja. Ekkor átkerül *Patched* állapotba, ahol egy meghatározott ideig marad, majd visszatér a *Pierceable*, kilyukasztható állapotba.

A csöre csak *Unoccupied* állapotban lehet rálépni, ha valaki sikeresen rálép, átkerül *Occupied* állapotba, ahol már nem enged senkit belépni. Ha a valaki sikeresen lelép a csőről, akkor visszaáll *Unoccupied* állapotba, és újra rá lehet lépni.

A cső alapvetően *NoEffect* állapotban van, ami azt jelenti, hogy se nem ragadós, se nem csúszós. Ha valaki valamikor ragadóssá vagy csúszóssé teszi, átkerül a neki megfelelő állapotba, és újraindítja az időzítőjét. Az időzítő lejártával visszakerül *NoEffect* állapotba. Ragadós állapotban (*TechnokolEffect*) ha belép egy karakter, addig nem tud semmit se tenni, amíg ki nem lép belőle. Csúszós állapotban (*BananaEffect*) ha belép egy karakter, akkor tovább csúszik egy véletlenszerű szomszédos aktív elemre.

- **Attribútumok**

- **-Effect effect:** A csövön lévő mozgást befolyásoló effekt.
- **-int capacity:** A csőben tárolt víz kapacitása. (A maximálisan befogadható víz mértéke.)
- **-int storedWater:** Adott pillanatban a csőben lévő víz mennyisége. Nem haladhatja meg a *capacity*-t.
- **-bool pierced:** Ez az érték igaz, ha a cső lyukas.
- **-int pierceCooldown:** A cső (újból) kilyukaszthatóságáig hátralévő idő.
- **-bool occupied:** Ez az érték mondja meg, hogy éppen áll-e valaki a csövön.
- **-WaterCollector ground:** A csatornahálózat alatt lévő “föld”. Ez tárolja a kifolyt víz mennyiségét.0xx
- **-ActiveElement[0..2] endpoints:** A cső végeihez csatlakoztatott aktív elemek halmaza.

- **Metódusok**

- **+void tick():** Lépteti a csövet. Csökkenti a *pierceCooldown*-t, és meghívja az *effect tick*-jét.
- **+void enter(Character character, Element from):** Ha a paraméterként kapott *Element* nem a cső szomszédja, vagy a csőnek van lecsatlakoztatott vége, vagy az *occupied* értéke *true*, akkor nem lehetséges rálépni, és *false* értékkel tér vissza. Ellenkező esetben meghívja az *effect enter*-jét, ha ez *true*-t ad vissza, átállítja az *occupied* értékét igazra, és *true*-val tér vissza, ha *false*-ot ad vissza, akkor *false*-szal tér vissza.
 - Pszeudokód:

függvény enter(Character character, Element from): void

ha a from a szomszédod és nem vagy foglalt és mindkét véged csatlakoztatva van

ha a csövön lévő effekt beengedi a karaktert

character belép az effekt-be

occupied érték átállítása igazra
- **+void leave():** Meghívja az *effect leave* függvényét. Ha ez *true*-t ad vissza, a cső *occupied* értéke *false*-ra változik. Visszaadja a meghívott függvény értékét.
- **+void patch():** A cső befoltozása, azaz a *pierced* állapot *false* lesz. Beállítja *pierceCooldown*-t egy véletlen értékre.
- **+void pierce():** Kilyukaszta a csövet, azaz a *pierced* változót *true*-ra állítja, ha a *pierceCooldown* lejárt (nullával egyenlő). Ilyenkor a *storedWater*-ben lévő vízmennyiséget áthelyezi a *ground*-ba.
- **+int addWater(int amount):** Megpróbálja eltárolni az adott mennyiségű vizet a *storedWater* attribútumban. Ha a *storedWater* meghaladná a *capacity* értékét, akkor a *capacity* és a *storedWater* különbségét tárolja el, majd visszatéríti az el nem tárolt vízmennyiséget. Ha lyukas a cső (a *pierced* értéke *true*) vagy van lecsatlakoztatott vége, akkor a víz a *storedWater* helyett a *ground*-ba kerül.
 - Pszeudokód:

függvény *addWater(int amount)*: int

ha nincs csatlakoztatva az egyik végpont **vagy** lyukas a cső:

add át a beérkező vizet a földnek

térj vissza 0-val

különben:

ha a beérkező vízzel több víz lenne eltárolva, mint amennyit a kapacitás megenged:

tárold el a kapacitásnak megfelelő mennyiségű vizet

térj vissza a maradék vízzel, ami nem került eltárolásra

különben:

tárold el a beérkező vizet

térj vissza 0-val

- **+int removeWater(int amount)**: Megpróbálja kivenni az adott mennyiséggű vizet a *storedWater*-ból, és a visszatérési érték a sikeresen kivett mennyiség lesz. Ha a kérт mennyiség nagyobb, mint a *storedWater* értéke, akkor a visszatérési érték a *storedWater* értéke lesz, és a *storedWater* értéke 0 lesz.

○ Pszeudokód:

függvény *removeWater(int amount)*: int

ha az *amount* nagyobb mint a *storedWater*:

nullázd a tárolt vizet

térj vissza az eredetileg benne lévő vízzel

különben:

csökkentsd a tárolt vizet az *amount*tal

térj vissza az *amount*tal

- **+int removeAllWater()**: Kiveszi az összes vizet a csőből és visszatér a kivett mennyiséggel.

- **+bool addEndpoint(ActiveElement activeElement)**: Csatlakoztatja magához a kapott *activeElement*-et (azaz hozzáadja az *endpoints*-hoz), ha az még nincs benne az *endpoints*-ban és az *endpoints*-nak még nincs 2 eleme, majd *true*-val tér vissza. Egyéb esetben a csatlakoztatás sikertelen és a függvény visszatérése *false* lesz. Ha sikerült a csatlakoztatás, akkor a *neighbours* listát is frissíti.

○ Pszeudokód:

függvény *addEndpoint(ActiveElement activeElement)*: bool

ha az *activeElement* még nincs hozzá csatolva:

ha van szabad végpont:

csatlakoztasd az *activeElement*-et

térj vissza igazzal

különben:

térj vissza hamissal

különben:

térj vissza hamissal

- **+bool removeEndpoint(ActiveElement activeElement)**: Ha az *endpoints* listában benne van a kapott *activeElement*, és az *occupied* értéke *false*, akkor eltávolítja a kapott *activeElement*-et az *endpoints* listából, kifolyatja a csőben lévő vizet a földre, és *true* értéket ad vissza. Más esetben a visszatérési érték *false* lesz. Ha sikerült a lecsatlakoztatás, akkor a *neighbours* listát is frissíti.

○ Pszeudokód:

függvény *removeEndpoint(ActiveElement activeElement)*: bool

ha foglalt **vagy** az *activeElement* nem végpontod

térj vissza hamissal

különben:

activeElement eltávolítása a végpontok közül
activeElement eltávolítása a szomszédok közül
 a tárolt víz átadása a földnek
terj vissza igazzal

- **+Pipe split():** A csövet kettévágja, vagyis a jelenlegi cső egyik endpoint-ját lecsatlakoztatja. A folyamat alatt új cső objektum jön létre, és ez összecsatlakoztatásra kerül az előzőleg lecsatlakoztatott endpoint-tal. Az újonnan létrehozott csövet visszatéríti.
 - Pszeudokód:

függvény split(): Pipe

létrehoz egy új csövet
 eltávolítja az egyik végpontját
 hozzáadódik az új csőhöz a végpont
terj vissza az új csővel
- **+void setEffect(Effect effect):** Beállítja a paraméterül kapott effectet.
- **+ActiveElement getRandomEndpoint():** Visszaadja az egyik végpontját véletlenszerűen választva.

8.2.9 Plumber

- **Felelősség**

Képes pumpákat felvenni/lehelyezni, csöveket le- és felcsatlakoztatni, illetve pumpákat és csöveket javítani. A ciszternákba gyűjt a vizet.

- **Ősosztályok**

Character → Plumber

- **Attribútumok**

- **-Pump[0..*] heldPumps:** A szerelőnél lévő pumpák.
- **-Pipe heldPipe:** A cső, aminek a vége a kezében van. Ha nincs nála cső, akkor ez *null*.

- **Metódusok**

- **+void repair(Pipe pipe):** Ha a paraméterként kapott csövön áll, akkor megjavítja.
- **+void repair(Pump pump):** Ha a paraméterként kapott pumpán áll, akkor megjavítja.
- **+void attachPipe(ActiveElement target):** Ha a paraméterként kapott *ActiveElement*-en áll, és a *heldPipe* értéke nem *null* (azaz tart csövet a kezében), megpróbálja felcsatlakoztatni ezt a csövet a kapott *ActiveElement*-re. Ha ez sikeres, akkor a *heldPipe* értéke *null* lesz.

○ Pszeudokód:

függvény attachPipe(ActiveElement target): void

ha a target-en állsz és van a kezedben cső:

ha hozzá lehet csatlakoztatni a kezedben lévő csövet a targethez

heldPipe csatlakoztatása *target*hez

heldPipe nullázása

- **+void detachPipe(ActiveElement target, Pipe pipe):** Ha a paraméterként kapott *ActiveElement*-en áll, és a *heldPipe* értéke *null* (azaz nem tart csövet a kezében), megpróbálja lecsatlakoztatni a paraméterként kapott csövet a kapott aktív elemről. Ha ez sikeres, a lecsatlakoztatott cső a *heldPipe*-ban tárolódik el.

○ Pszeudokód:

függvény `detachPipe(ActiveElement target, Pipe pipe): void`

ha a csövön állsz és nincs a kezedenben cső:

ha le lehet csatlakoztatni a *pipe*-ot

pipe lecsatlakoztatása a *targetról*

pipe felvétele a *heldPipe*-ba

- **+void pickUpPump(Cistern cistern):** Ha a paraméterként kapott ciszternán áll, akkor a ciszternától kér egy pumpát.
- **+void placePump(Pipe pipe):** Ha a paraméterként kapott csövön áll, van nála pumpa, és le tud lépni a *location*-jéről (meghívja a *location leave* függvényét), akkor a pumpát lehelyezi. Ehhez a kapott csövön meghívja a *split* függvényt, a pumpába beköti a *pipe*-ot, és a *split* visszatérési értékeként kapott másik csövet. Ezután a lehelyezett pumpára áll.

○ Pszeudokód:

függvény `placePump(Pipe pipe): void`

ha a *pipe-on* állsz és van a kezedenben legalább egy pumpa:

ha el tudod hagyni a *pipe*-ot:

vágd ketté a *pipe*-ot (hívд meg a *split()* függvényét)

a *heldPump*-ból vegyél ki egy pumpát

a kivett pumpához csatlakoztasd hozzá a *pipe*-ot

a kivett pumpához csatlakoztasd hozzá a levágott csövet
lépj rá az új pumpára

8.2.10 Pump

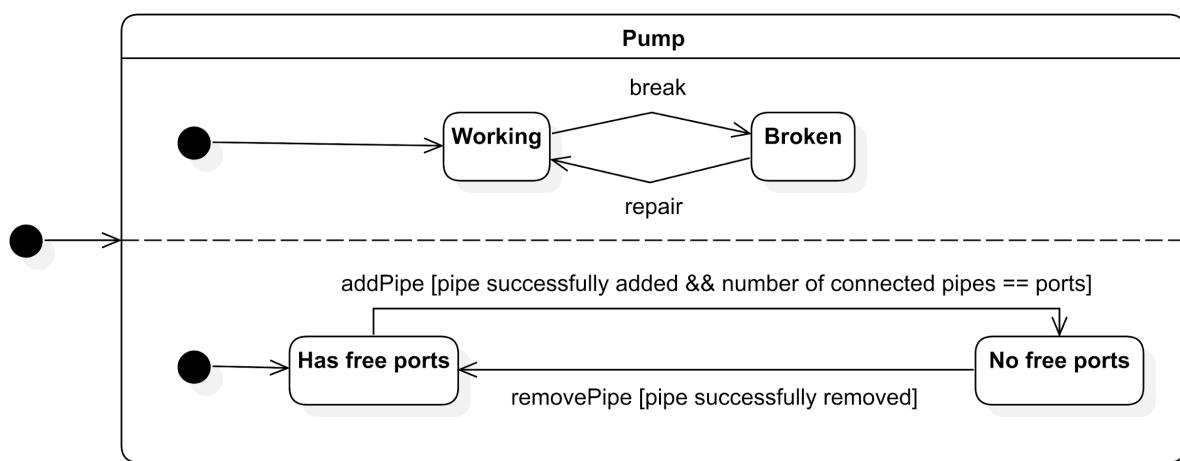
• Felelősség

Kimeneti és bemeneti csövei között a víztartályán keresztül folyatja a vizet. El tud romlani, ebben az esetben a csövei között a vízfolyás megszűnik. Kimeneti és bemeneti csövei állíthatók.

• Ősosztályok

Element → ActiveElement → Pump

• Állapotok



A pumpa *Working* állapotban mozgatja a vizet a tartályából a kimeneti csövébe, és a bemeneti csövből a tartályába amikor léptetik. *Broken* állapotban ez nem történik meg. A *Working*

állapotot a *broken* attribútum *false* értéke, míg a *Broken* állapotot az attribútum *true* értéke reprezentálja.

A *Has free ports* állapotban a pumpához lehet még csöveget csatlakoztatni, ellenben a *No free ports* állapottal, ahol a pumpára már a maximális mennyiséggű cső van csatlakoztatva, így minden sikertelen lesz a csatlakoztatási kísérlet. Az állapotokat a *ports* számattribútum és a *connectedPipes* asszociáció elemei számának a viszonya reprezentálja.

- **Attribútumok**

- **-int tankCapacity:** A pumpa víztartályának víztároló kapacitása. (A maximálisan befogadható víz mértéke.)
- **-int tankStoredWater:** A pumpa víztartályában tárolt víz mennyisége. Nem haladhatja meg a *tankCapacity*-t.
- **-bool broken:** *True* ha a pumpa el van romolva. *False*, ha helyesen működik.
- **-int ports:** A pumpa portjainak száma, vagyis a maximum hozzácsatlakoztatható csövek száma.
- **-Pipe inputPipe:** A pumpa bemeneti csöve.
- **-Pipe outputPipe:** A pumpa kimeneti csöve.

- **Metódusok**

- **+void tick():** Ha a pumpa működik, a kimeneti csövébe pumpál annyi vizet, amennyit a cső kapacitása, illetve a tárolt víz mennyisége enged, majd megtölți a tartályát a lehető legtöbb vízzel a bemeneti csőből.
- **+bool addPipe(Pipe pipe):** Megpróbálja felcsatlakoztatni magára a kapott csövet. Ha már az összes portja foglalt, nem sikerül a felcsatlakoztatás és *false*-al tér vissza. Egyéb esetben azt csinálja, mint az őse.
- **+bool removePipe(Pipe pipe):** Megpróbálja lecsatlakoztatni magáról a kapott csövet. Ugyanazt csinálja mint az őse, de ha a sikeresen lecsatlakoztatott cső az *inputPipe* vagy az *outputPipe* volt, akkor az adott érték nullázódik.
- **+void repair():** Átállítja a *broken* változó értékét *false*-ra.
- **+void break():** Átállítja a *broken* változó értékét *true*-ra.
- **+void setInputPipe(Pipe pipe):** Átállítja az *inputPipe* értékét a kapott csőre, ha az a cső csatlakozik a pumpához és nincs beállítva kimeneti csőnek.
- **+void setOutputPipe(Pipe pipe):** Átállítja az *outputPipe* értékét a kapott csőre, ha az a cső csatlakozik a pumpához és nincs beállítva bemeneti csőnek.

8.2.11 Saboteur

- **Felelősség**

A csőhálózaton mozognak és banánokat dobálnak. A földre gyűjtik a vizet.

- **Ősosztályok**

Character → Saboteur

- **Metódusok**

- **+void dropBanana(Pipe pipe):** Beállítja a cső effektjét *BananaEffect*-re.

8.2.12 Spring

- **Felelősség**

Vizet folyat a rácsatlakoztatott csövekbe.

- **Ősosztályok**

Element → ActiveElement → Spring

- **Attribútumok**

- **-int waterDispensedPerTick:** A tickenként kibocsátott víz mennyisége.

- **Metódusok**

- **+void tick():** A *connectedPipes*-okba a *waterDispensedPerTick*-nek megfelelő mennyiségű vizet tölt (a cső által visszadobott vízzel nem foglalkozik).

8.2.13 TechnokolEffect

- **Felelősség**

A cső ragasztós (technokolos) effektjét implementáló osztály. A karakter rálépés után nem tud lelépni a csőről, amíg a ragasztó effekt ideje le nem jár.

- **Ősosztályok**

Effect → TechnokolEffect

- **Attribútumokcsere**

- **- int countdown:** Az effekt hátralévő ideje. Ha nulla lesz, az effekt lejár.
- **- boolean activated:** Élesítve van-e a ragasztó. Az első lelépésnél lesz *true*-ra állítva, így a karakter aki ragasztotta a csövet még le tud lépni.

- **Metódusok**

- **+ boolean enter(Character character):** *True* értéket ad vissza, ha a karakter jelenlegi helyéről való lelépés sikeres, egyébként *false*-t.
- **+ boolean leave:** Ha az activated értéke *false*, átállítja *true*-ra, és *true*-val tér vissza (le lehet lépni a csőről), egyébként *false*-al tér vissza (nem lehet lelépni a csőről).
- **+ void tick():** Csökkenti a *countdown* attribútum értékét, ha az eléri a 0-t, kicseréli a csövön lévő effektet *NoEffect*-re.

8.2.14 WaterCollector

- **Felelősség**

Vizet tárol végtelen kapacitással.

- **Ősosztályok**

Nincs ősosztálya

- **Attribútumok**

- **-int storedAmount:** A tárolt víz mennyisége.

- **Metódusok**
 - **+void storeAmount(int amount)**: Hozzáadja a *storedAmount* változóhoz a kapott mennyiséggű vizet.

8.3 A tesztek részletes tervei, leírásuk a teszt nyelvén

8.3.1 Repair Pipe

- **Leírás**

Teszteli, hogy egy szerelő meg tudja javítani a kilyukasztott csövet, amin áll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a Plumber osztály *repair(Pipe)* és a Pipe osztály *patch* metódusát. Lehetséges hiba, hogy a cső lyukas állapotban marad (*Pipe.repair()* hibája).

- **Bemenet**

```
load 1.map.pasta
repair plumber pipe
view pipe pierced
```

- **Pálya fájl (1.map.pasta)**

```
pump1: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pump2: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pipe: Pipe
    endpoints = [pump1, pump2]
    occupied = true
    pierced = true
    ground = collector
    neighbours = [pump1, pump2]
collector: WaterCollector
plumber: Plumber
    location = pipe
```

- **Elvárt kimenet**

```
load successful
pipe: Pipe
    pierced = false
```

8.3.2 Repair Pump

- **Leírás**

Teszteli, hogy egy szerelő meg tudja javítani az elromlott pumpát, amin áll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a Plumber osztály *repair(Pump)* és a Pump osztály *repair* metódusát. Lehetséges hiba, hogy a cső elromlott állapotban marad (*Pump.repair()* hibája).

- **Bemenet**

```
load 2.map.pasta
repair plumber pump
view pump broken
```

- **Pálya fájl (2.map.pasta)**

```
pump: Pump
    broken = true
plumber: Plumber
```

- **Elvárt kimenet**

```
load successful
pump: Pump
    broken = false
```

location = pump

8.3.3 Pierce Pipe

- **Leírás**

Teszeli, hogy egy szerelő (karakter) ki tudja lyukasztani azt a csövet, amin áll, ha a cső pierceCooldown-ja már lejárt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a Character és Pipe osztályok *pierce* metódusát. Lehetséges hiba, hogy a cső nem kerül át kilyukasztott állapotba (Pipe.*pierce()* hibája).

- **Bemenet**

load 3.map.pasta
pierce plumber pipe
view pipe pierced

- **Pálya fájl (3.map.pasta)**

pump1: Pump
connectedPipes = [pipe]
neighbours = [pipe]
pump2: Pump
connectedPipes = [pipe]
neighbours = [pipe]
pipe: Pipe
endpoints = [pump1, pump2]
occupied = true
pierced = false
pierceCooldown = 0
ground = collector
neighbours = [pump1, pump2]
collector: WaterCollector
plumber: Plumber
location = pipe

- **Elvárt kimenet**

load successful
pipe: Pipe
pierced = true

8.3.4 Pierce unpierceable Pipe

- **Leírás**

Teszeli, hogy egy szerelő (karakter) nem tudja kilyukasztani azt a csövet, amin áll, ha a cső pierceCooldown-ja nem járt még le (nemrégiben lett megjavítva).

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a Character és Pipe osztályok *pierce* metódusát. Lehetséges hiba, hogy a cső átkerül kilyukasztott állapotba, a pierceCooldown-t figyelmen kívül hagyva (Pipe.*pierce()* hibája).

- **Bemenet**

load 4.map.pasta
pierce plumber pipe
view pipe pierced

- **Pálya fájl (4.map.pasta)**

pump1: Pump
connectedPipes = [pipe]

- **Elvárt kimenet**

load successful
pipe: Pipe

```

neighbours = [pipe]           pierced = false
pump2: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pipe: Pipe
    endpoints = [pump1, pump2]
    occupied = true
    pierced = false
    pierceCooldown = 10
    ground = collector
    neighbours = [pump1, pump2]
collector: WaterCollector
plumber: Plumber
    location = pipe

```

8.3.5 Technokol Pipe

- Leírás**
Egy szerelő (karakter) ragadóssá tesz egy csövet.
- Ellenőrzött funkcionalitás, várható hibahelyek**
Teszteli a Character osztály *applyTechnokol* és a Pipe osztály *setEffect* függvényét.
Lehetséges hiba, hogy a csőre nem kerül ragadós effekt.
- Bemenet**
load 5.map.pasta
technokol plumber pipe
view pipe effect
- Pálya fájl (5.map.pasta)**
Lásd 8.3.3 pálya fájl.
- Elvárt kimenet**
load successful
pipe: Pipe
effect = Technokol

8.3.6 Banana Pipe

- Leírás**
Egy szabotőr csúszóssá tesz egy csövet.
- Ellenőrzött funkcionalitás, várható hibahelyek**
Teszteli a Saboteur osztály *dropBanana* és a Pipe osztály *setEffect* függvényét.
Lehetséges hiba, hogy a csőre nem kerül csúszós effekt.
- Bemenet**
load 6.map.pasta
banana saboteur pipe
view pipe effect
- Pálya fájl (6.map.pasta)**
pump1: Pump
 connectedPipes = [pipe]
 neighbours = [pipe]
pump2: Pump
 connectedPipes = [pipe]
- Elvárt kimenet**
load successful
pipe: Pipe
effect = Banana

```

neighbours = [pipe]
pipe: Pipe
  effect = None
  endpoints = [pump1, pump2]
  neighbours = [pump1, pump2]
saboteur: Saboteur
  location = pump

```

8.3.7 Set Pump input

- Leírás**
Egy szerelő (karakter) átállítja egy pumpa bemenetét.
- Ellenőrzött funkcionalitás, várható hibahelyek**
Teszteli a Character és a Pump osztály *setInputPipe* függvényét. Lehetséges hiba, hogy a cső nem kerül beállításra a pumpa bemeneti csövének.

- Bemenet**
load 7-8.map.pasta
set input plumber pump1 pipe
view pump1 inputPipe

- Pálya fájl (7-8.map.pasta)**

```

pump1: Pump
  inputPipe = null
  outputPipe = null
  connectedPipes = [pipe]
  neighbours = [pipe]
pump2: Pump
  connectedPipes = [pipe]
  neighbours = [pipe]
pipe: Pipe
  endpoints = [pump1, pump2]
  ground = ground
  neighbours = [pump1, pump2]
plumber: Plumber
  location = pump1
ground: WaterCollector

```

- Elvárt kimenet**

```

load successful
pump1: Pump
  inputPipe = pipe

```

8.3.8 Set Pump output

- Leírás**
Egy szerelő átállítja egy pumpa kimenetét.
- Ellenőrzött funkcionalitás, várható hibahelyek**
Teszteli a Character és a Pump osztály *setOutputPipe* függvényét. Lehetséges hiba, hogy a cső nem kerül beállításra a pumpa kimeneti csövének.
- Bemenet**
load 7-8.map.pasta
set output plumber pump1 pipe
view pump1 outputPipe
- Pálya fájl (7-8.map.pasta)**
- Elvárt kimenet**

Lásd 8.3.7 pálya fájl.

```
load successful
pump1: Pump
    outputPipe = pipe
```

8.3.9 Unable to Set Pump input

- **Leírás**

Egy szerelő megpróbálja átállítani egy pumpa bemenetét egy olyan csőre, ami már meg van neki adva kimenetként, úgyhogy sajnos nem sikerül neki.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a Character és a Pump osztály `setInputPipe` függvényét. Lehetséges hiba, hogy a cső beállításra kerül a pumpa bemeneti csövének annak ellenére, hogy már kimeneti csőnek be van állítva.

- **Bemenet**

```
load 9.map.pasta
set input plumber pump1 pipe
view pump1 inputPipe
view pump1 outputPipe
```

- **Pálya fájl (9.map.pasta)**

```
pump1: Pump
    outputPipe = pipe
    connectedPipes = [pipe]
    neighbours = [pipe]
pump2: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pipe: Pipe
    endpoints = [pump1, pump2]
    ground = ground
    neighbours = [pump1, pump2]
plumber: Plumber
    location = pump1
ground: WaterCollector
```

- **Elvárt kimenet**

```
load successful
pump1: Pump
    inputPipe = null
pump1: Pump
    outputPipe = pipe
```

8.3.10 Unable to Set Pump output

- **Leírás**

Egy szerelő megpróbálja átállítani egy pumpa kimenetét egy olyan csőre, ami már meg van neki adva bemenetként, úgyhogy sajnos nem sikerül neki.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a Character és a Pump osztály `setOutputPipe` függvényét. Lehetséges hiba, hogy a cső beállításra kerül a pumpa kimeneti csövének annak ellenére, hogy már bemeneti csőnek be van állítva.

- **Bemenet**

```
load 10.map.pasta
set output plumber pump1 pipe
view pump1 inputPipe
view pump1 outputPipe
```

- **Pálya fájl (10.map.pasta)**

```

pump1: Pump
    inputPipe = pipe
    connectedPipes = [pipe]
    neighbours = [pipe]
pump2: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pipe: Pipe
    endpoints = [pump1, pump2]
    ground = ground
    neighbours = [pump1, pump2]
plumber: Plumber
    location = pump1
ground: WaterCollector

```

- **Elvárt kimenet**

```

load successful
pump1: Pump
    inputPipe = pipe
pump1: Pump
    outputPipe = null

```

8.3.11 Detach both ends of unoccupied Pipe from Pump

- **Leírás**

Teszteli, hogy 2 különböző pumpán állva két szerelő a 2 pumpát összekötő cső 2 végét le tudja csatlakoztatni, amennyiben a cső nem foglalt (nem állnak rajta).

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a Character *detachPipe*, a Pump és ActiveElement *removePipe*, és a Pipe *removeEndpoint* függvényeket. Lehetséges hiba, hogy a cső csatlakozva marad az egyik/másik/mindkettő pumpához, vagy hogy nem kerül minden két szerelő kezébe a lecsatlakoztatott cső.

- **Bemenet**

```

load 11.map.pasta
detach plumber1 pump1 pipe
detach plumber2 pump2 pipe
view pipe endpoints
view pipe neighbours
view plumber1 heldPipe
view plumber1 location
view plumber2 heldPipe
view pump1 connectedPipes
view pump1 neighbours
view pump2 connectedPipes
view pump2 neighbours

```

- **Pálya fájl (11.map.pasta)**

```

ground: WaterCollector
pipe: Pipe
    endpoints = [pump1, pump2]
    occupied = false
    ground = ground
    neighbours = [pump1, pump2]
plumber1: Plumber
    heldPipe = null

```

- **Elvárt kimenet**

```

load successful
pipe: Pipe
    endpoints = []
pipe: Pipe
    neighbours = []
plumber1: Plumber
    heldPipe = pipe
plumber1: Plumber

```

```

location = pump1
plumber2: Plumber
    heldPipe = null
    location = pump2
pump1: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pump2: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
location = pump1
plumber2: Plumber
    heldPipe = pipe
pump1: Pump
    connectedPipes = []
pump1: Pump
    neighbours = []
pump2: Pump
    connectedPipes = []
pump2: Pump
    neighbours = []

```

8.3.12 Detach Pipe from Cistern

- **Leírás**

Teszteli, hogy egy szerelő le tudja csatlakoztatni a ciszternáról egy cső végét, amennyiben a ciszternán áll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a Character *detachPipe*, a Cistern és ActiveElement *removePipe*, és a Pipe *removeEndpoint* függvényeket. Lehetséges hiba, hogy a cső csatlakozva marad a ciszternához, vagy hogy nem kerül a szerelő kezébe a lecsatlakoztatott cső.

- **Bemenet**

```

load 12.map.pasta
detach plumber cistern pipe
view pipe endpoints
view pipe neighbours
view plumber heldPipe
view plumber location
view pump connectedPipes
view pump neighbours
view cistern connectedPipes
view cistern neighbours

```

- **Pálya fájl (12.map.pasta)**

```

ground: WaterCollector
cisternCollector: WaterCollector
pipe: Pipe
    endpoints = [pump, cistern]
    occupied = false
    ground = ground
    neighbours = [pump, cistern]
plumber: Plumber
    heldPipe = null
    location = cistern
pump: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
cistern: Cistern
    connectedPipes = [pipe]
    neighbours = [pipe]

```

- **Elvárt kimenet**

```

load successful
pipe: Pipe
    endpoints = [pump]
pipe: Pipe
    neighbours = [pump]
plumber: Plumber
    heldPipe = pipe
plumber: Plumber
    location = cistern
pump: Pump
    connectedPipes = [pipe]
pump: Pump
    neighbours = [pipe]
cistern: Cistern
    connectedPipes = []
cistern: Cistern

```

```
collector = cisternCollector
neighbours = []
```

8.3.13 Detach Pipe from Spring

- **Leírás**

Teszeli, hogy egy szerelő le tudja csatlakoztatni a forrásról egy cső végét, amennyiben a forráson áll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a Character *detachPipe*, a Spring és ActiveElement *removePipe*, és a Pipe *removeEndpoint* függvényeket. Lehetséges hiba, hogy a cső csatlakozva marad a forráshoz, vagy hogy nem kerül a szerelő kezébe a lecsatlakoztatott cső.

- **Bemenet**

```
load 13.map.pasta
detach plumber spring pipe
view pipe endpoints
view pipe neighbours
view plumber heldPipe
view plumber location
view pump connectedPipes
view pump neighbours
view spring connectedPipes
view spring neighbours
```

- **Pálya fájl (13.map.pasta)**

```
ground: WaterCollector
pipe: Pipe
    endpoints = [pump, spring]
    occupied = false
    ground = ground
    neighbours = [pump, spring]
plumber: Plumber
    heldPipe = null
    location = spring
pump: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
spring: Spring
    connectedPipes = [pipe]
    neighbours = [pipe]
```

- **Elvárt kimenet**

```
load successful
pipe: Pipe
    endpoints = [pump]
    pipe: Pipe
        neighbours = [pump]
    plumber: Plumber
        heldPipe = pipe
    plumber: Plumber
        location = spring
    pump: Pump
        connectedPipes = [pipe]
    pump: Pump
        neighbours = [pipe]
    spring: Spring
        connectedPipes = []
    spring: Spring
        neighbours = []
```

8.3.14 Detach occupied Pipe

- **Leírás**

Teszeli, hogy egy szerelő egy foglalt csövet nem tud lecsatlakoztatni arról pumpáról (aktív elemről), amin áll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a Character *detachPipe*, a Pump és ActiveElement *removePipe*, és a Pipe *removeEndpoint* függvényeket. Lehetséges hiba, hogy a cső mégis lecsatlakoztatásra

kerül, figyelmen kívül hagyva, hogy foglalt, vagy nem kerül lecsatlakoztatásra, de a szerelő kezébe kapja a csövet.

- **Bemenet**

```
load 14.map.pasta
detach plumber1 pump1 pipe
view pipe endpoints
view pipe neighbours
view plumber1 heldPipe
view plumber1 location
view plumber2 location
view pump1 connectedPipes
view pump1 neighbours
view pump2 connectedPipes
view pump2 neighbours
```

- **Pálya fájl (14.map.pasta)**

```
ground: WaterCollector
pipe: Pipe
    endpoints = [pump1, pump2]
    occupied = true
    ground = ground
    neighbours = [pump1, pump2]
plumber1: Plumber
    heldPipe = null
    location = pump1
plumber2: Plumber
    location = pipe
pump1: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pump2: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
```

- **Elvárt kimenet**

```
load successful
pipe: Pipe
    endpoints = [pump1, pump2]
pipe: Pipe
    neighbours = [pump1, pump2]
plumber1: Plumber
    heldPipe = null
plumber1: Plumber
    location = pump1
plumber2: Plumber
    location = pipe
pump1: Pump
    connectedPipes = [pipe]
pump1: Pump
    neighbours = [pipe]
pump2: Pump
    connectedPipes = [pipe]
pump2: Pump
    neighbours = [pipe]
```

8.3.15 Detach Pipe end while holding another Pipe end

- **Leírás**

Ellenőrzi, hogy egy szerelő nem tud másik csövet lecsatlakoztatni úgy, hogy már van a kezében egy lecsatlakoztatott csővég.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a Character *detachPipe* függvényét. Lehetséges hiba, hogy a cső mégis lecsatlakoztatásra kerül, figyelmen kívül hagyva, hogy már van a szerelő kezében cső.

- **Bemenet**

```
load 15.map.pasta
detach plumber pump1 pipe1
view pipe1 endpoints
view pipe1 neighbours
```

```

view pipe2 endpoints
view pipe2 neighbours
view plumber heldPipe
view plumber location
view pump1 connectedPipes
view pump1 neighbours
view pump2 connectedPipes
view pump2 neighbours

```

- **Pálya fájl (15.map.pasta)**

```

ground: WaterCollector
pipe1: Pipe
    endpoints = [pump1, pump2]
    ground = ground
    neighbours = [pump1, pump2]
pipe2: Pipe
    endpoints = [pump2]
    ground = ground
    neighbours = [pump2]
plumber: Plumber
    heldPipe = pipe2
    location = pump1
pump1: Pump
    connectedPipes = [pipe1]
    neighbours = [pipe1]
pump2: Pump
    connectedPipes = [pipe1]
    neighbours = [pipe1]

```

- **Elvárt kimenet**

```

load successful
pipe1: Pipe
    endpoints = [pump1, pump2]
pipe1: Pipe
    neighbours = [pump1, pump2]
pipe2: Pipe
    endpoints = [pump2]
pipe2: Pipe
    neighbours = [pump2]
plumber: Plumber
    heldPipe = pipe2
plumber: Plumber
    location = pump1
pump1: Pump
    connectedPipes = [pipe1]
pump1: Pump
    neighbours = [pipe1]
pump2: Pump
    connectedPipes = [pipe1]
pump2: Pump
    neighbours = [pipe1]

```

8.3.16 Detach both ends of new Pipe

- **Leírás**

Ellenőrzi, hogy két szerelő azonos ciszternán állva, le tudja csatlakoztatni minden két végét az ott keletkezett új csőnek. A lecsatlakoztatáskor az első szerelő a cső lecsatlakoztatott végét kapja meg, a második szerelő a cső ciszternába csatlakozó végét kapja meg.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a Character *detachPipe*, a Pump, Cistern és ActiveElement *removePipe*, és a Pipe *removeEndpoint* függvényét. Lehetséges hiba, hogy a cső ciszternához eső végét kapja meg az első szerelő, és ilyenkor a második szerelő nem tudja lecsatlakoztatni a másik végét, vagy egyik szerelő sem tudja lecsatlakoztatni az új csövet, vagy sikerül a lecsatlakoztatás, de nem kerül a szerelők kezébe a cső.

- **Bemenet**

```

load 16.map.pasta
detach plumber1 cistern pipe
view pipe endpoints
view pipe neighbours

```

```

view plumber1 heldPipe
view cistern connectedPipes
view cistern neighbours
detach plumber2 cistern pipe
view pipe endpoints
view pipe neighbours
view plumber2 heldPipe
view cistern connectedPipes
view cistern neighbours

```

- **Pálya fájl (16.map.pasta)**

```

cisternCollector: WaterCollector
cistern: Cistern
    newPipes = [pipe]
    collector = cisternCollector
    connectedPipes = [pipe]
    neighbours = [pipe]
ground: WaterCollector
pipe: Pipe
    endpoints = [cistern]
    ground = ground
    neighbours = [cistern]
plumber1: Plumber
    heldPipe = null
    location = cistern
plumber2: Plumber
    heldPipe = null
    location = cistern

```

- **Elvárt kimenet**

```

load successful
pipe: Pipe
    endpoints = [cistern]
pipe: Pipe
    neighbours = [cistern]
plumber1: Plumber
    heldPipe = pipe
cistern: Cistern
    connectedPipes = [pipe]
cistern: Cistern
    neighbours = [pipe]
pipe: Pipe
    endpoints = []
pipe: Pipe
    neighbours = []
plumber2: Plumber
    heldPipe = pipe
cistern: Cistern
    connectedPipes = []
cistern: Cistern
    neighbours = []

```

8.3.17 Attach both ends of Pipe

- **Leírás**

Teszteli, hogy két szerelő két különböző pumpán (aktív elemen) állva, kezükben ugyanannak a csőnek egy-egy végével, fel tudja csatlakoztatni a cső nála levő végét arra a pumpára (aktív elemre) amin áll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a Character *attachPipe*, a Pump és ActiveElement *addPipe*, és a Pipe *addEndpoint* függvényét. Lehetséges hiba, hogy a cső egyik/másik/mindkettő vége nem kerül felcsatlakoztatásra, és/vagy a szerelők kezében marad a csővég, annak ellenére, hogy sikeres volt a felcsatlakoztatás.

- **Bemenet**

```

load 17.map.pasta
attach plumber1 pump1
view plumber1 heldPipe
view pump1 connectedPipes

```

```

view pump1 neighbours
attach plumber2 pump2
view plumber2 heldPipe
view pump2 connectedPipes
view pump2 neighbours
view pipe endpoints
view pipe neighbours

```

- **Pálya fájl (17.map.pasta)**

collector: WaterCollector

pipe: Pipe

```

endpoints = []
ground = collector
neighbours = []

```

pump1: Pump

```

connectedPipes = []
neighbours = []

```

pump2: Pump

```

connectedPipes = []
neighbours = []

```

plumber1: Plumber

```

heldPipe = pipe
heldPumps = []
location = pump1

```

plumber2: Plumber

```

heldPipe = pipe
heldPumps = []
location = pump2

```

- **Elvárt kimenet**

load successful

plumber1: Plumber

```

heldPipe = null

```

pump1: Pump

```

connectedPipes = [pipe]

```

pump1: Pump

```

neighbours = [pipe]

```

plumber2: Plumber

```

heldPipe = null

```

pump2: Pump

```

connectedPipes = [pipe]

```

pump2: Pump

```

neighbours = [pipe]

```

pipe: Pipe

```

endpoints = [pump1, pump2]

```

pipe: Pipe

```

neighbours = [pump1, pump2]

```

8.3.18 Attach both ends of Pipe to same Pump

- **Leírás**

Teszteli, hogy egy szerelő egy pumpán állva, kezében egy csőnek az egyik végével, aminek a másik vége a pumpára van csatlakoztatva, amin áll, nem tudja felcsatlakoztatni a kezében levő cső végét, mivel a másik vége már hozzá van csatlakoztatva a pumpához.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a Plumber *attachPipe* és a Pump, ActiveElement *addPipe* függvényeket. Lehetséges hiba, hogy mégis felcsatlakoztatásra kerül a cső vége, annak ellenére, hogy a másik vége már csatlakoztatva van ugyanehhez a pumpához.

- **Bemenet**

```

load 18.map.pasta
attach plumber pump
view plumber heldPipe
view pump connectedPipes
view pump neighbours
view pipe endpoints
view pipe neighbours

```

- **Pálya fájl (18.map.pasta)**

- **Elvárt kimenet**

```

pipe: Pipe
    endpoints = [pump]
    ground = collector
    neighbours = [pump]
collector: WaterCollector
plumber: Plumber
    heldPipe = pipe
    location = pipe
pump: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]

load successful
plumber: Plumber
    heldPipe = pipe
pump: Pump
    connectedPipes = [pipe]
pump: Pump
    neighbours = [pipe]
pipe: Pipe
    endpoints = [pump]
pipe: Pipe
    neighbours = [pump]

```

8.3.19 Attach both ends of Pipe to same Cistern

- **Leírás**

Teszteri, hogy egy szerelő egy ciszternán állva, kezében egy csőnek az egyik végével, aminek a másik vége a ciszternára van csatlakoztatva, amin áll, nem tudja felcsatlakoztatni a kezében levő cső végét, mivel a másik vége már hozzá van csatlakoztatva a ciszternához.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteri a Plumber *attachPipe* és az ActiveElement *addPipe* függvényeket. Lehetséges hiba, hogy mégis felcsatlakoztatásra kerül a cső vége, annak ellenére, hogy a másik vége már csatlakoztatva van ugyanehhez a ciszternához.

- **Bemenet**

```

load 19.map.pasta
attach plumber cistern
view plumber heldPipe
view cistern connectedPipes
view cistern neighbours
view pipe endpoints
view pipe neighbours

```

- **Pálya fájl (19.map.pasta)**

```

pipe: Pipe
    endpoints = [cistern]
    ground = collector1
    neighbours = [cistern]
collector1: WaterCollector
collector2: WaterCollector
plumber: Plumber
    heldPipe = pipe
    location = pipe
cistern: Cistern
    collector = collector2
    connectedPipes = [pipe]
    neighbours = [pipe]

```

- **Elvárt kimenet**

```

load successful
plumber: Plumber
    heldPipe = pipe
cistern: Cistern
    connectedPipes = [pipe]
cistern: Cistern
    neighbours = [pipe]
pipe: Pipe
    endpoints = [cistern]
pipe: Pipe
    neighbours = [cistern]

```

8.3.20 Attach both ends of Pipe to same Spring

- **Leírás**

Teszeli, hogy egy szerelő egy forráson állva, kezében egy csőnek az egyik végével, aminek a másik vége a forrásra van csatlakoztatva, amin áll, nem tudja felcsatlakoztatni a kezében levő cső végét, mivel a másik vége már hozzá van csatlakoztatva a forráshoz.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a Plumber *attachPipe* és az ActiveElement *addPipe* függvényeket. Lehetséges hiba, hogy mégis felcsatlakoztatásra kerül a cső vége, annak ellenére, hogy a másik vége már csatlakoztatva van ugyanehhez a forráshoz.

- **Bemenet**

```
load 20.map.pasta
attach plumber spring
view plumber heldPipe
view spring connectedPipes
view spring neighbours
view pipe endpoints
view pipe neighbours
```

- **Pálya fájl (20.map.pasta)**

```
pipe: Pipe
    endpoints = [spring]
    ground = collector
    neighbours = [spring]
collector: WaterCollector
plumber: Plumber
    heldPipe = pipe
    location = pipe
spring: Spring
    connectedPipes = [pipe]
    neighbours = [pipe]
```

- **Elvárt kimenet**

```
load successful
plumber: Plumber
    heldPipe = pipe
spring: Spring
    connectedPipes = [pipe]
spring: Spring
    neighbours = [pipe]
pipe: Pipe
    endpoints = [spring]
pipe: Pipe
    neighbours = [spring]
```

8.3.21 Attach Pipe to Pump with no free ports

- **Leírás**

Ellenőrzi, hogy egy szerelő nem tudja a nála levő csövet felcsatlakoztatni egy olyan pumpán állva, aminek már nincs szabad portja.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a Plumber *attachPipe* és a Pump *addPipe* függvényeket. Lehetséges hiba, hogy a cső felcsatlakoztatásra kerül, figyelmen kívül hagyva a portszámát.

- **Bemenet**

```
load 21.map.pasta
attach plumber pump
view plumber heldPipe
view pump connectedPipes
view pump neighbours
view pipe3 endpoints
view pipe3 neighbours
```

- **Pálya fájl (21.map.pasta)**

```
pump: Pump
```

- **Elvárt kimenet**

```
load successful
```

```

ports = 2
connectedPipes = [pipe1, pipe2]
neighbours = [pipe1, pipe2]
collector: WaterCollector
pipe1: Pipe
    ground = collector
    neighbours = [pump]
pipe2: Pipe
    ground = collector
    neighbours = [pump]
pipe3: Pipe
    ground = collector
plumber: Plumber
heldPipe = pipe3
location = pump

```

8.3.22 Acquire Pump

- Leírás**
Teszteli, hogy egy szerelő fel tud-e venni egy pumpát egy ciszternán állva.
- Ellenőrzött funkcionalitás, várható hibahelyek**
Teszteli a *Plumber* osztály *pickUpPump* és a *Cistern* osztály *acquirePump* függvényét.
Várható hiba, hogy a szerelő nem kap egy új pumpát a kezébe.

- Bemenet**

```

load 22.map.pasta
getpump plumber cistern pump
view plumber heldPumps

```

- Pálya fájl (22.map.pasta)**

```

collector: WaterCollector
cistern: Cistern
    collector = collector
plumber: Plumber
    heldPumps = []
    location = cistern

```

```

plumber: Plumber
heldPipe = pipe3
pump: Pump
connectedPipes = [pipe1, pipe2]
pump: Pump
neighbours = [pipe1, pipe2]
pipe3: Pipe
endpoints = []
pipe3: Pipe
neighbours = []

```

- Elvárt kimenet**

```

load successful
plumber: Plumber
heldPumps = [pump]

```

8.3.23 Place Pump

- Leírás**
Teszteli, hogy egy szerelő le tud-e rakni egy pumpát egy csőre.
- Ellenőrzött funkcionalitás, várható hibahelyek**
Teszteli a *Plumber* osztály *placePump*, a *Pipe* osztály *split*, *addEndpoint*, *removeEndpoint*, és az *ActiveElement* osztály *addPipe*, illetve *removePipe* metódusát.
Várható hiba, hogy nem jön létre új cső és nem kerül becsatlakoztatásra a lerakott pumpa az eredeti, és az új cső közé.
- Bemenet**

```

load 23.map.pasta
placepump plumber pipe newPipe
view plumber location

```

view pipe endpoints
 view pipe neighbours
 view pump3 connectedPipes
 view pump3 neighbours
 view newPipe endpoints
 view newPipe neighbours

- **Pálya fájl (23.map.pasta)**

pump1: Pump
 connectedPipes = [pipe]
 neighbours = [pipe]
 pump2: Pump
 connectedPipes = [pipe]
 neighbours = [pipe]
 pump3: Pump
 pipe: Pipe
 endpoints = [pump1, pump2]
 occupied = true
 pierced = true
 ground = collector
 neighbours = [pump1, pump2]
 collector: WaterCollector
 plumber: Plumber
 heldPumps = [pump3]
 location = pipe

- **Elvárt kimenet**

load successful
 plumber: Plumber
 location = pump3
 pipe: Pipe
 endpoints = [pump2, pump3]
 pipe: Pipe
 neighbours = [pump2, pump3]
 pump3: Pump
 connectedPipes = [pipe, newPipe]
 pump3: Pump
 neighbours = [pipe, newPipe]
 newPipe: Pipe
 endpoints = [pump1, pump3]
 newPipe: Pipe
 neighbours = [pump1, pump3]

8.3.24 Unable to Place Pump

- **Leírás**

Teszteli, hogy nem lehet lerakni egy pumpát egy csőre, ha nincs a szerelő kezében egy pumpa se.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a *Plumber* osztály *placePump* függvényét. Várható hiba, hogy a cső kapcsolódásai megváltoznak és új cső keletkezik.

- **Bemenet**

load 24.map.pasta
 placepump plumber pipe newPipe
 view plumber location
 view pipe endpoints
 view pipe neighbours
 view newPipe

- **Pálya fájl (24.map.pasta)**

pump1: Pump
 connectedPipes = [pipe]
 neighbours = [pipe]
 pump2: Pump
 connectedPipes = [pipe]
 neighbours = [pipe]
 pipe: Pipe

- **Elvárt kimenet**

load successful
 plumber: Plumber
 location = pipe
 pipe: Pipe
 endpoints = [pump1, pump2]
 pipe: Pipe
 neighbours = [pump1, pump2]

```

endpoints = [pump1, pump2]           ! object not found
occupied = true
pierced = true
ground = collector
neighbours = [pump1, pump2]
collector: WaterCollector
plumber: Plumber
heldPumps = []
location = pipe

```

8.3.25 Move to unoccupied Pipe

- **Leírás**

Egy szerelő rálép egy szabadon lévő (nem foglalt) csőre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a *Character moveTo*, *Pipe enter* és *ActiveElement leave* metódusokat.
Lehetséges hibák: *Character* lelép, de nem lép át, a *Pipe* nem engedi be.

- **Bemenet**

```

load 25.map.pasta
move plumber pipe
view plumber location
view pipe occupied

```

- **Pálya fájl (25.map.pasta)**

```

pump1: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pump2: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pipe: Pipe
    endpoints = [pump1, pump2]
    occupied = false
    ground = collector
    neighbours = [pump1, pump2]
collector: WaterCollector
plumber: Plumber
location = pump1

```

- **Elvárt kimenet**

```

load successful
plumber: Plumber
location = pipe
pipe: Pipe
occupied = true

```

8.3.26 Move to occupied Pipe

- **Leírás**

Egy szerelő megpróbál rálépní egy foglalt csőre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a *Character moveTo* és *Pipe enter* metódusokat.
Lehetséges hibák: *Character* lelép mielőtt a pipe beengedné, a *Pipe* beengedi.

- **Bemenet**

```
load 26.map.pasta
```

```
move plumber1 pipe
view plumber1 location
view pipe occupied
```

- **Pálya fájl (26.map.pasta)**

pump1: Pump
 connectedPipes = [pipe]
 neighbours = [pipe]

pump2: Pump
 connectedPipes = [pipe]
 neighbours = [pipe]

pipe: Pipe
 endpoints = [pump1, pump2]
 occupied = true
 ground = collector
 neighbours = [pump1, pump2]

collector: WaterCollector

plumber1: Plumber
 location = pump1

plumber2: Plumber
 location = pipe
- **Elvárt kimenet**

load successful

plumber1: Plumber
 location = pump1

pipe: Pipe
 occupied = true

8.3.27 Move to detached Pipe

- **Leírás**
 Egy szerelő megpróbál rálépni egy szabad végű csőre.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
 Teszteli a *Character moveTo* és *Pipe enter* metódusokat.
 Lehetséges hibák: *Character* lelép mielőtt a pipe beengedné, a *Pipe* beengedi.
- **Bemenet**

load 27.map.pasta

move plumber pipe

view plumber location

view pipe occupied
- **Pálya fájl (27.map.pasta)**

pump: Pump
 connectedPipes = [pipe]
 neighbours = [pipe]

pipe: Pipe
 endpoints = [pump]
 occupied = false
 ground = collector
 neighbours = [pump]

collector: WaterCollector

plumber: Plumber
 location = pump
- **Elvárt kimenet**

load successful

plumber: Plumber
 location = pump

pipe: Pipe
 occupied = false

8.3.28 Move to Pump

- **Leírás**

Egy szerelő rálép egy pumpára.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a *Character moveTo*, *Pump enter* és *Pipe leave* metódusokat.

Lehetséges hibák: *Character* lelép, de nem lép át, a *Pump* nem engedi be.

- **Bemenet**

```
load 28.map.pasta
move plumber pump1
view plumber location
view pipe occupied
```

- **Pálya fájl (28.map.pasta)**

```
pump1: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pump2: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pipe: Pipe
    endpoints = [pump1, pump2]
    occupied = true
    ground = collector
    neighbours = [pump1, pump2]
collector: WaterCollector
plumber: Plumber
location = pipe
```

- **Elvárt kimenet**

```
load successful
plumber: Plumber
location = pump1
pipe: Pipe
occupied = false
```

8.3.29 Move to Cistern

- **Leírás**

Egy szerelő rálép egy ciszternára.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a *Character moveTo*, *Cister enter* és *Pipe leave* metódusokat.

Lehetséges hibák: *Character* lelép, de nem lép át, a *Cistern* nem engedi be.

- **Bemenet**

```
load 29.map.pasta
move plumber cistern
view plumber location
view pipe occupied
```

- **Pálya fájl (29.map.pasta)**

```
pump: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pipe: Pipe
    endpoints = [pump]
    occupied = true
```

- **Elvárt kimenet**

```
load successful
plumber: Plumber
location = cistern
pipe: Pipe
occupied = false
```

```

ground = collector1
neighbours = [pump]
collector1: WaterCollector
collector2: WaterCollector
plumber: Plumber
location = pipe
cistern: Cistern
collector = collector2
connectedPipes = [pipe]
neighbours = [pipe]

```

8.3.30 Move to Spring

- **Leírás**

Egy szerelő rálép egy forrásra.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a *Character moveTo*, *Spring enter* és *Pipe leave* metódusokat.

Lehetséges hibák: *Character* lelép, de nem lép át, a *Spring* nem engedi be.

- **Bemenet**

```

load 30.map.pasta
move plumber spring
view plumber location
view pipe occupied

```

- **Pálya fájl (30.map.pasta)**

```

pump: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pipe: Pipe
    endpoints = [pump]
    occupied = true
    ground = collector
    neighbours = [pump]
collector: WaterCollector
plumber: Plumber
location = pipe
spring: Spring
    connectedPipes = [pipe]
    neighbours = [pipe]

```

- **Elvárt kimenet**

```

load successful
plumber: Plumber
location = spring
pipe: Pipe
occupied = false

```

8.3.31 Move to banana Pipe

- **Leírás**

Egy szerelő rálép egy csúszós csöre és tovább csúszik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a *Character moveTo*, *Pipe enter*, *BananaEffect enter* és *Pipe leave* metódusokat.

Lehetséges hibák: *Character* nem csúszik tovább.

- **Bemenet**

```

load 31.map.pasta
banana saboteur pipe 1 false
move saboteur pump1
move saboteur pipe
view saboteur location
view pipe occupied

```

- **Pálya fájl (31.map.pasta)**

```

pump1: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pump2: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pipe: Pipe
    endpoints = [pump1, pump2]
    occupied = true
    ground = collector
    neighbours = [pump1, pump2]
collector: WaterCollector
saboteur: Saboteur
location = pipe

```

- **Elvárt kimenet**

```

load successful
saboteur: Saboteur
location = pipe
pipe: Pipe
occupied = false

```

8.3.32 Move through technokoled Pipe

- **Leírás**

Egy szerelő rálép egy ragadós csöre. Majd megpróbál továbblépni, de beragad.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a *Character moveTo*, *Pipe enter*, *TechnokolEffect enter*, *Pipe leave* és *TechnokolEffect leave* metódusokat.

Lehetséges hibák: *Character* egyből tovább tud lépni, a ragadóság elmulásával se tud a *Character* továbblépni.

- **Bemenet**

```

load 32.map.pasta
technokol plumber pipe 1
move plumber pump1
move plumber pipe
move plumber pump2
view plumber location
tick pipe
move plumber pump2
view plumber location

```

- **Pálya fájl (32.map.pasta)**

```

pump1: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
pump2: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]

```

- **Elvárt kimenet**

```

load successful
plumber: Plumber
location = pipe
plumber: Plumber
location = pump2

```

```

pipe: Pipe
    endpoints = [pump1, pump2]
    occupied = true
    ground = collector
    neighbours = [pump1, pump2]
collector: WaterCollector
plumber: Plumber
location = pipe

```

8.3.33 Break Pump

- **Leírás**

Teszteli, hogy el lehet-e törni egy pumpát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a *Pump* osztály *break* függvényét. Várható hiba, hogy a pumpa *broken* változója nem áll át *true*-ra.

- **Bemenet**

```

load 33.map.pasta
break pump
view pump broken

```

- **Pálya fájl (33.map.pasta)**

```

pump: Pump
broken = false

```

- **Elvárt kimenet**

```

load successful
pump: Pump
broken = true

```

8.3.34 Spawn Pipe

- **Leírás**

Teszteli, hogy tud-e a ciszterna új csövet létrehozni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a *Cistern* osztály *spawnPipe* függvényét. Várható hiba, hogy az új cső nem jön létre, vagy nincs a ciszternához kötve.

- **Bemenet**

```

load 34.map.pasta
spawnpipe cistern collector2 pipe
view cistern
view pipe ground
view pipe endpoints
view pipe neighbours

```

- **Pálya fájl (34.map.pasta)**

```

collector1: WaterCollector
cistern: Cistern
    collector = collector1
collector2: WaterCollector

```

- **Elvárt kimenet**

```

load successful
cistern: Cistern
    newPipes = [pipe]
    collector = collector1
    connectedPipes = [pipe]
    neighbours = [pipe]
pipe: Pipe
    ground = collector2

```

```

pipe: Pipe
    endpoints = [cistern]
pipe: Pipe
    neighbours = [cistern]

```

8.3.35 Waterflow from Spring

- **Leírás**

Teszeli, hogy a forrásból tud-e folyni a víz a csatlakoztatott csőbe.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a *Spring* osztály *tick* metódusát. Várható hiba, hogy a csőbe nem kerül víz (pipe.storedWater == 0).

- **Bemenet**

```

load 35.map.pasta
view pipe storedWater
tick spring
view pipe storedWater

```

- **Pálya fájl (35.map.pasta)**

```

ground: WaterCollector
pump: Pump
    connectedPipes = [pipe]
    neighbours = [pipe]
spring: Spring
    connectedPipes = [pipe]
    neighbours = [pipe]
    waterDispensedPerTick = 1
pipe: Pipe
    endpoints = [pump, spring]
    neighbours = [pump, spring]
    pierced = false
    capacity = 5
    storedWater = 0
    ground = ground

```

- **Elvárt kimenet**

```

load successful
pipe: Pipe
    storedWater = 0
pipe: Pipe
    storedWater = 1

```

8.3.36 Waterflow through Pump

- **Leírás**

Teszeli a pumpa (*pump2*) vízmozgatását. Először a tartályába, majd a kimeneti csövébe rakja a vizet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a *Pump* osztály *tick*, *Pipe* osztály *addWater* és *removeWater* függvényét. Várható hiba, hogy nem kerül át a bemeneti csőből (*pipe1*) a tartályba, illetve a következő tick hatására a tartályból a kimeneti csőbe (*pipe2*).

- **Bemenet**

```

load 36.map.pasta
view pipe1 storedWater
view pump2 tankStoredWater
view pipe2 storedWater

```

```

tick pump2
view pipe1 storedWater
view pump2 tankStoredWater
view pipe2 storedWater
tick pump2
view pipe1 storedWater
view pump2 tankStoredWater
view pipe2 storedWater

```

- **Pálya fájl (36.map.pasta)**

```

ground: WaterCollector
pump1: Pump
    connectedPipes = [pipe1]
    neighbours = [pipe1]
pump2: Pump
    connectedPipes = [pipe1, pipe2]
    neighbours = [pipe1, pipe2]
    inputPipe = pipe1
    outputPipe = pipe2
pump3: Pump
    connectedPipes = [pipe2]
    neighbours = [pipe2]
pipe1: Pipe
    endpoints = [pump1, pump2]
    neighbours = [pump1, pump2]
    pierced = false
    capacity = 5
    storedWater = 1
    ground = ground
pipe2: Pipe
    endpoints = [pump2, pump3]
    neighbours = [pump2, pump3]
    pierced = false
    capacity = 5
    storedWater = 0
    ground = ground

```

- **Elvárt kimenet**

```

load successful
pipe1: Pipe
    storedWater = 1
pump2: Pump
    tankStoredWater = 0
pipe2: Pipe
    storedWater = 0
pipe1: Pipe
    storedWater = 0
pump2: Pump
    tankStoredWater = 1
pipe2: Pipe
    storedWater = 0
pipe1: Pipe
    storedWater = 0
pump2: Pump
    tankStoredWater = 0
pipe2: Pipe
    storedWater = 1

```

8.3.37 Waterflow into Cistern

- **Leírás**

Teszeli, hogy a ciszterna ki tudja-e szedni a rácsatlakoztatott csőből a vizet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a *Cistern* osztály *tick*, *Pipe* osztály *removeAllWater* függvényét. Várható hiba, hogy nem szedi ki a vizet a csőből, és nem rakja át a vízgyűjtőjébe.

- **Bemenet**

```

load 37.map.pasta
view pipe storedWater
view collector storedAmount
tick cistern
view pipe storedWater

```

view collector storedAmount

- **Pálya fájl (37.map.pasta)**

ground: WaterCollector
 collector: WaterCollector
 pump: Pump
 connectedPipes = [pipe]
 neighbours = [pipe]
 pipe: Pipe
 endpoints = [pump, cistern]
 neighbours = [pump, cistern]
 pierced = false
 capacity = 5
 storedWater = 1
 ground = ground
 cistern: Cistern
 connectedPipes = [pipe]
 neighbours = [pipe]
 collector = collector

- **Elvárt kimenet**

load successful
 pipe: Pipe
 storedWater = 1
 collector: WaterCollector
 storedAmount = 0
 pipe: Pipe
 storedWater = 0
 collector: WaterCollector
 storedAmount = 1

8.3.38 Waterflow into full Pipe

- **Leírás**

Teszteli, hogy nem tud-e belerakni vizet (pl. egy forrás) egy olyan csőbe, ami elérte a kapacitását.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteli a *Pipe* osztály *addWater* függvényét. Várható hiba, hogy a csőben több víz lesz, mint a kapacitása.

- **Bemenet**

load 38.map.pasta
 view pipe capacity
 view pipe storedWater
 tick spring
 view pipe capacity
 view pipe storedWater

- **Pálya fájl (38.map.pasta)**

ground: WaterCollector
 pump: Pump
 connectedPipes = [pipe]
 neighbours = [pipe]
 spring: Spring
 connectedPipes = [pipe]
 neighbours = [pipe]
 waterDispensedPerTick = 1
 pipe: Pipe
 endpoints = [pump, spring]
 neighbours = [pump, spring]
 pierced = false
 capacity = 5

- **Elvárt kimenet**

load successful
 pipe: Pipe
 capacity = 5
 pipe: Pipe
 storedWater = 5
 pipe: Pipe
 capacity = 5
 pipe: Pipe
 storedWater = 5

```
storedWater = 5
ground = ground
```

8.3.39 Waterflow into detached Pipe

- **Leírás**

Teszeli, hogy a lecsatlakoztatott végű csőbe rakott víz elfolyik-e a földre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a *Pipe* osztály *addWater*, *WaterCollector* osztály *storeAmount* függvényét.
Várható hiba, hogy nem kerül a csőből a vízgyűjtőbe a víz.

- **Bemenet**

```
load 39.map.pasta
view pipe storedWater
view ground storedAmount
tick spring
view pipe storedWater
view ground storedAmount
```

- **Pálya fájl (39.map.pasta)**

```
ground: WaterCollector
spring: Spring
    connectedPipes = [pipe]
    neighbours = [pipe]
    waterDispensedPerTick = 1
pipe: Pipe
    endpoints = [spring]
    neighbours = [spring]
    pierced = false
    capacity = 5
    storedWater = 0
    ground = ground
```

- **Elvárt kimenet**

```
load successful
pipe: Pipe
    storedWater = 0
ground: WaterCollector
    storedAmount = 0
pipe: Pipe
    storedWater = 0
ground: WaterCollector
    storedAmount = 1
```

8.3.40 Waterflow into pierced Pipe

- **Leírás**

Teszeli, hogy a lyukas csőbe rakott víz elfolyik-e a földre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszeli a *Pipe* osztály *addWater*, *WaterCollector* osztály *storeAmount* függvényét.
Várható hiba, hogy nem kerül a csőből a vízgyűjtőbe a víz.

- **Bemenet**

```
load 40.map.pasta
view pipe storedWater
view ground storedAmount
tick spring
view pipe storedWater
view ground storedAmount
```

- **Pálya fájl (40.map.pasta)**

```
ground: WaterCollector
pump: Pump
```

- **Elvárt kimenet**

```
load successful
pipe: Pipe
```

```

connectedPipes = [pipe]
neighbours = [pipe]
spring: Spring
    connectedPipes = [pipe]
    neighbours = [pipe]
    waterDispensedPerTick = 1
pipe: Pipe
    endpoints = [pump, spring]
    neighbours = [pump, spring]
    pierced = true
    capacity = 5
    storedWater = 0
    ground = ground

```

```

storedWater = 0
ground: WaterCollector
    storedAmount = 0
pipe: Pipe
    storedWater = 0
ground: WaterCollector
    storedAmount = 1

```

8.3.41 Tick Banana Pipe

- **Leírás**

Teszteri, hogy lejár-e egy csövön az 1 élettartamú banán effekt, és egy *NoEffect* kerül-e a helyére.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteri a *Pipe* osztály *tick*, *setEffect*, *BananaEffect* osztály *tick* függvényét. Várható hiba, hogy a tickelés hatására nem szűnik meg az effekt (lesz helyette *NoEffect*).

- **Bemenet**

```

load 41-42.map.pasta
banana saboteur pipe 1
view pipe effect
tick pipe
view pipe effect

```

- **Pálya fájl (41-42.map.pasta)**

```

ground: WaterCollector
saboteur: Saboteur
    location = pipe
pipe: Pipe
    occupied = true
    ground = ground

```

- **Elvárt kimenet**

```

load successful
pipe: Pipe
    effect = Banana
pipe: Pipe
    effect = None

```

8.3.42 Tick Technokoled Pipe

- **Leírás**

Teszteri, hogy lejár-e egy csövön az 1 élettartamú technokol effekt, és egy *NoEffect* kerül-e a helyére.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Teszteri a *Pipe* osztály *tick*, *setEffect*, *TechnokolEffect* osztály *tick* függvényét. Várható hiba, hogy a tickelés hatására nem szűnik meg az effekt (lesz helyette *NoEffect*).

- **Bemenet**

```

load 41-42.map.pasta
technokol saboteur pipe 1

```

view pipe effect
tick pipe
view pipe effect

- **Pálya fájl (41-42.map.pasta)**
Lásd 8.3.41 Pálya fájl.

- **Elvárt kimenet**
load successful
pipe: Pipe
 effect = Technokol
pipe: Pipe
 effect = None

8.4 A tesztelést támogató programok tervezése

A tesztek ellenőrzésére saját segédprogramot készítettünk, amely képes egy tesztfájl tartalmát átírányítani a fő program standard inputjára, és az így (standard outputon) kapott kimenetet összehasonlítani egy megadott elvárt kimenetet tartalmazó fájl tartalmával.

Teszt(ek) futtatásakor a segédprogramot a fő program mappájából (ahonnan a `java macaroni.Main` parancsot lehet sikeresen futtatni) kell indítani.

Tesztfájl tartalma: - Csak proto által értelmezhető parancsokat tartalmazhat (külön sorokban)

Összesített tesztfájl / konfig fájl: - Csak tesztfájlok elérési útvonalát tartalmazhatja (külön sorokban). Kiterjesztése: “.config.pasta”! Más kiterjesztésű fájlt a rendszer teszfájlként próbál meg feldolgozni.

Elvárt kimeneti fájl: A program úgy dolgozik, mintha egy adott tesztfájllal egy mappában, egy “expected” nevű mappán belül “.expected.pasta” kiterjesztéssel az adott tesztfájllal megegyező nevű elvárt kimenetet leíró fájl lenne. Ha ilyet nem talál, akkor a teszt futása sikertelen lesz.

A segédprogram indításakor argumentumként adjuk meg egy vagy több tesztet leíró fájl elérési útját. (Ezek lehetnek tesztfájlok, vagy konfig fájlok is.) Ha egy fájl többször is szerepel, csak egyszer fogja a program futtatni.

A megadott tesztek párhuzamosan kerülnek kiértékelésre. A segédprogram minden esetben külön fő program egységet indít, és ennek a standard inputjára irányítja a tesztfájl tartalmát. A fő program standard inputján megjelenő kimenetet ezután összehasonlítja a teszthez tartozó elvárt kimeneti fájllal, amennyiben az létezik.

Végül a segédprogram saját standard outputjára írja a tesztek eredményét. Elsőnek felsorolja a sikeresen lefutott teszteket, majd kiírja a sikertelen tesztek első elvárattól eltérő kimeneti sorát. Elvárt kimeneti fájl hiányában eltérő sor helyett a hiány tényét dokumentálja.

Példa kimenet (segédprogram kimenetével megegyező formátumban):

==== PASSED ===

|> test3.pasta
|> test4.pasta

==== FAILED ===

|> test1.pasta
(line_6) Expected: “ pierced = true”, Got: “ pierced = false”

|> test2.pasta
(line_4) Missing expected output file!

8.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2023.04.26. 12:15	1,5 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: Konzultáltunk az IL408-ban a laborvezetővel. Döntés: A következő értekezleten kiosztjuk az elvégzendő feladatokat.
2023.04.27. 10:00	4 óra	Kereszthury	7. parancsok , kimeneti és bemeneti nyelv javítása, módosítása, kibővítése
2023.05.01. 22:00	2 óra	Szilczer	8.2.3, 8.2.4, 8.2.6, 8.2.10, 8.2.11, 8.2.12, 8.2.14 elkészítése.
2023.05.02. 13:00	1 óra	Tóth	8.2.2, 8.2.5, 8.2.7, 8.2.13 elkészítése.
2023.05.02. 15:00	6 óra	Harsányi	8.2 befejezése
2023.05.02. 18:30	1,5 óra	Galgóczy	8.4 megtervezése és leírása
2023.05.02. 20:00	0,5 óra	Galgóczy	8.2 Spring, Character és Cistern átnézése
2023.05.02. 20:30	2 óra	Galgóczy	8.1 átnézése
2023.05.02. 22:00	3 óra	Kereszthury	8.2 átnézése, ellenőrzése
2023.05.02. 22:30	6,5 óra	Galgóczy	8.3 1-4, 17-34 tesztek kimeneti és bemeneti nyelvének leírása
2023.05.03. 00:00	1,5 óra	Harsányi	8.3.5-10 elkészítése
2023.05.03. 00:00	2 óra	Tóth	8.3.11-16 tesztesetek elkészítése
2023.05.03. 00:00	4,5 óra	Szilczer	8.2.10 statechart, 8.3.35-42 elkészítése.
2023.05.03. 10:00	1 óra	Harsányi	8.3.25-32 elkészítése
2023.05.03. 10:00	1 óra	Szilczer	8.3.22-24, 8.3.33-34 tesztesetek leírásának elkészítése.
2023.05.03. 10:00	1 óra	Tóth	8.3.17-21 tesztesetek leírásának elkészítése.

10. Prototípus beadása

10.0 Módosítások

10.0.1 Prototípus interface-definíció

A prototípus kimeneti sorának kezdetén “>” jellet jelzi, ha új parancs fogadására készen áll.

10.0.2 A tesztelést támogató programok tervezése

Az elkészült program helyességének ellenőrzését saját tesztelőprogram segítségével végezzük. A továbbiakban ennek a szoftvernek a működését részletezem.

Tesztfájl:

Egy specifikus tesztesetet leíró fájl. Soronként tartalmazhat a prototípus program által értelmezhető parancsot, üres sort, kommentet, vagy assertion-t.

Load és save parancs esetén elég a tesztfájl mappájához relatívan megadni az elérési útvonalat. A “#”-el kezdődő sorokat a tesztelőprogram nem veszi figyelembe.

A program képes ellenőrzéseket futtatni egy játékbeli objektum attribútumára. Ehhez a sornak az alábbi módon kell kinéznie: @<objectName>.<attributeName> = <expectedValue>, ahol a relációs jelzett értékek helyére valódi programnak megfelelő szöveget kell behelyettesíteni. Egy objektum létezésére az alábbi szintaxissal lehet rákérdezni: @<objectName>. Illetve nem létezésére: @!<objectName>.

A tesztelőprogram minden nem ‘@’-el vagy ‘#’-el kezdődő nem üres sort prototípus parancsként értelmez.

Példa:

load maps/24.map.pasta

placepump plumber pipe newPipe

```
@plumber:location = pipe
# assert that newPipe does not exist
@!newPipe
```

Konfigfájl:

Több fájlt leíró fájl. Soronként tesztfájlok vagy konfigfájlok elérési útvonalát, üres sort, és kommentet tartalmaz. A fájlok leírásához elég a konfigfájl mappájához relatívan megadni az elérési útvonalat. A “#”-el kezdődő sorokat a tesztelőprogram nem veszi figyelembe.

A tesztelőprogram minden nem ‘#’-el kezdődő nem üres sort tesztfájlként vagy konfigfájlként értelmez, kiterjesztéstől függően. A “.config.pasta” kiterjesztésű fájlokat konfigfájlként kezeli, minden más kiterjesztésű tesztfájlként. Konfigfájlok esetén a fájl mappáján belül egy másik mappában kell, hogy létezzen a hivatkozott config; különben nem lesz feldolgozva.

Példa:

config

test1.pasta

test2.pasta

```
# nested config:  
nested/other_tests.config.pasta
```

Futtatás:

A tesztprogramot a főprogram mappájából kell futtatni (ahol sikeresen kiadható a 'java macaroni.Main' parancs). A program az argumentumként megadott (egyszerre akár több) teszt- és konfigfájlokat fogja kiértékelni. Amelyik tesztfájl többször szerepel, az is csak egyszer fog lefutni.
(lásd.: 10.1.3)

Naplózás:

A program a kapott tesztekhez kimeneti mappát készít, elsősorban naplózási szempontból. Ezt az első argumentumként kapott fájl mappáján belül egy "logs" nevű mappába teszi, dátum és idő szerint "yyyy.MM.dd-HH_mm_ss" formátumú névvel ellátva. Ezután a program az alapvetően generált fájait ezen mappán belülre helyezi.

Működés:

A kapott tesztek párhuzamosan kerülnek futtatásra, minden teszthez külön fő programot indítva. A tesztfájlból lévő parancsok és ellenőrzések a kapott sorrendben kerülnek feldolgozásra, ahol az egyes ellenőrzések helyett egy *view* parancs adódik ki a fő program szabványos kimenetére.

Mentés:

A fő program által kapott kimeneteket és oda küldött bemeneteket egy teszthez tartozó kimeneti fájlban együtt tárolja el a program. A fájl helye az idő szerint generált mappában egy új "out" nevű mappa lesz. Neve megegyezik a teszt(fájl) nevével, kiterjesztése ".out.pasta". A program a futtatott tesztekről is másolatot készít az új idővel bélyegzett mappába.

Eredmények:

A program a tesztek futása előtt kiírja, hogy elkezdte a futtatást. Egy teszt futtatása után kiírja az adott teszt nevét. Az összes teszt lefutása után az eredményeket a szabályos kimenetére és egy "_SUM.pasta" nevű fájlba összegezve az alábbi formátumba írja ki a program:

```
==== PASSED ====  
|> test3.pasta  
|> test4.pasta
```

```
==== FAILED !===  
|> test1.pasta  
(line_7) @pipe.pierced => Expected: true, Got: false  
(line_8) @pipe.patched => Invalid field name  
  
|> test2.pasta  
(line_4) @pipe.pierced => Object not found  
(line_5) @notPipe => Object does not exist  
(line_6) @!existingPipe => Object exists
```

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
src/macaroni/commandHandler/commands/ AttachCommand .java	1697	2023.04.27. 0:57	A csatlakoztató parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ BananaCommand .java	1715	2023.04.28. 1:01	A csúszóssá tevő parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ BreakCommand .java	1095	2023.04.27. 23:09	A tönkretevő parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ Command .java	1745	2023.04.26. 21:38	A parancsok interfésze.
src/macaroni/commandHandler/commands/ CreateCommand .java	7780	2023.04.26. 21:38	Az objektumok létrehozásának parancsát megvalósító osztály.
src/macaroni/commandHandler/commands/ DetachCommand .java	2120	2023.04.26. 0:22	A lecsatlakoztató parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ GetPumpCommand .java	1352	2023.04.28. 1:01	A pumpa felvétel parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ HelpCommand .java	1872	2023.04.26. 21:38	A segítség parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ ListCommand .java	2388	2023.04.26. 22:11	A listázó parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ LoadCommand .java	1550	2023.04.27. 23:09	A pályabetöltő parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ MoveCommand .java	1157	2023.04.28. 1:01	A játékos mozgás parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ PierceCommand .java	1151	2023.04.28. 1:01	A kilyukasztó parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ PlacePumpCommand .java	1342	2023.04.28. 1:01	A pumpa lehelyező parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ RepairCommand .java	1893	2023.04.28. 1:01	A javítás parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ SaveCommand .java	1817	2023.04.27 23:09	A fájlba mentés parancsot megvalósító osztály.

src/macaroni/commandHandler/commands/ SetCommand.java	1478	2023.04.28. 1:01	A pumpa be- vagy kimenet állító parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ SpawnPipeCommand.java	1448	2023.04.28. 1:01	A ciszternára új csövet helyező parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ TechnokolCommand.java	1519	2023.04.28. 1:01	A ragadóssá tevő parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ TickCommand.java	1792	2023.04.27 23:09	Az idő műlás parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ ViewCommand.java	1754	2023.04.27. 0:15	A játékbeli objektumok állapotának megtekintése parancsot megvalósító osztály.
src/macaroni/commandHandler/ CommandInterpreter.java	3663	2023.04.26 21:38	A felhasználó által kiadott parancsokat feldolgozó osztály..

Fájl neve	Méret	Keletkezés ideje	Tartalom
src/macaroni/model/element/ ActiveElement.java	1428	2023.04.06. 23:10	Az Aktív elemek osztálya.
src/macaroni/model/effect/ BananaEffect.java	1624	2023.04.23. 18:17	A Csúszósság effektjét megvalósító osztály.
src/macaroni/model/character/ Character.java	3001	2023.04.06. 23:10	A Karakterek osztálya.
src/macaroni/model/element/ Cistern.java	1961	2023.04.06. 23:10	A Ciszternák osztálya.
src/macaroni/model/effect/ Effect.java	1078	2023.04.23. 18:17	Az effektek interfésze.
src/macaroni/model/element/ Element.java	1029	2023.04.06. 23:10	Az Elemek osztálya.
src/macaroni/model/effect/ NoEffect.java	1043	2023.04.23. 18:17	A semmilyen effektet megvalósító osztály.
src/macaroni/model/element/ Pipe.java	7255	2023.04.06. 23:10	A Csövek osztálya.
src/macaroni/model/character/ Plumber.java	3779	2023.04.06. 23:10	A Szerelők osztálya.
src/macaroni/model/element/ Pump.java	3865	2023.04.06. 23:10	A Pumpák osztálya.
src/macaroni/model/character/ Saboteur.java	858	2023.04.06. 23:10	A Szabotőrök osztálya.
src/macaroni/model/element/ Spring.java	936	2023.04.06. 23:10	A Források osztálya.
src/macaroni/model/effect/ TechnokolEffect.java	1867	2023.04.23. 18:17	A Ragadósság effektjét megvalósító osztály.

src/macaroni/model/misc/ WaterCollector.java	450	2023.04.06. 23:10	A Víztartályok osztálya.
--	-----	-------------------	--------------------------

Fájl neve	Méret	Keletkezés ideje	Tartalom
src/macaroni/ Main.java	263	2023.04.26 21:38	A proto kódja innen indul. Tartalmazza a main függvényt.
src/macaroni/utils/ ModelObjectFactory.java	10238	2023.04.28. 16:11	Osztály amely modell osztályokból hoz létre példányokat, és nyilvántartja azokat név szerint.
src/macaroni/utils/ ModelObjectSerialize.java	10907	2023.05.02. 14:43	Osztály, ami a modell osztályok és szöveges reprezentációjuk közti konverziót végzi.
src/macaroni/utils/ Random.java	5629	2023.05.10. 1:31	Véletlen szám generáló osztály, ami segítségével lehet determinisztikus értékeket is beállítani a véletlen helyett.

Fájl neve	Méret	Keletkezés ideje	Tartalom
src/pasta/asserts/ Assertion.java	517	2023.05.01. 3:37	Funckionális interface tesztelés közbeni ellenőrzés elvégzéséhez.
src/pasta/asserts/ Assertions.java	1719	2023.05.10. 23:00	Segédosztály tesztelési ellenőrzéshez.
src/pasta/asserts/ AttributeAssertion.java	2975	2023.05.10. 23:00	Tagváltozói ellenőrzést leíró osztály.
src/pasta/ Main.java	11738	2023.04.30. 1:04	A tesztelést támogató program kódja innen indul. Tartalmazza a main függvényt.
src/pasta/asserts/ ObjectAssertion.java	1887	2023.05.11 1:56	Objektum létezésének ellenőrzését leíró osztály.
src/pasta/asserts/ Result.java	838	2023.04.30. 17:45	Ellenőrzés végeredményét leíró osztály.
src/pasta/ Test.java	10376	2023.04.30. 15:21	Teszeset kereső osztály.
tests/all.config.pasta	170	2023.05.11. 2:37	Az összes tesztet közösen elindító konfigfájl.

Fájl neve	Méret	Keletkezés ideje	Tartalom
tests/Basic Actions/.config.pasta	361	2023.05.10. 22:33	Az 1-10., 22-24. teszteket közösen elindító konfigfájl.
tests/Basic Actions/1. Repair Pipe.pasta	71	2023.05.03. 2:46	Az 1. teszt tesztfájlja.
tests/Basic Actions/maps/1.map.pasta	342	2023.05.03. 2:46	Az 1. teszthez tartozó pályaleíró fájl.

tests/Basic Actions/2. Repair Pump.pasta	70	2023.05.03. 2:46	A 2. teszt tesztfájlja.
tests/Basic Actions/maps/2.map.pasta	64	2023.05.03. 2:46	A 2. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/3. Pierce Pipe.pasta	70	2023.05.03. 2:46	A 3. teszt tesztfájlja.
tests/Basic Actions/maps/3.map.pasta	367	2023.05.03. 2:46	A 3. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/4. Pierce unpierceable Pipe.pasta	71	2023.05.03. 2:46	A 4. teszt tesztfájlja.
tests/Basic Actions/maps/4.map.pasta	356	2023.05.03. 2:46	A 4. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/5. Technokol Pipe.pasta	77	2023.05.10. 22:33	Az 5. teszt tesztfájlja.
tests/Basic Actions/maps/5.map.pasta	367	2023.05.10. 22:33	Az 5. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/6. Banana Pipe	72	2023.05.10. 22:33	A 6. teszt tesztfájlja.
tests/Basic Actions/maps/6.map.pasta	271	2023.05.10. 22:33	A 6. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/7. Set Pump input.pasta	82	2023.05.10. 22:33	A 7. teszt tesztfájlja.
tests/Basic Actions/maps/7.map.pasta	341	2023.05.10. 22:33	A 7. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/8. Set Pump output.pasta	84	2023.05.10. 22:33	A 8. teszt tesztfájlja.
tests/Basic Actions/maps/8.map.pasta	341	2023.05.10. 22:33	A 8. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/9. Unable to Set Pump input.pasta	108	2023.05.10. 22:33	A 9. teszt tesztfájlja.
tests/Basic Actions/maps/9.map.pasta	319	2023.05.10. 22:33	A 9. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/10. Unable to Set pump output.pasta	110	2023.05.10. 22:33	A 10. teszt tesztfájlja.
tests/Basic Actions/maps/10.map.past a	318	2023.05.10. 22:33	A 10. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/22. Acquire Pump.pasta	87	2023.05.03. 4:41	A 22. teszt tesztfájlja.
tests/Basic Actions/maps/22.map.past a	134	2023.05.03. 4:41	A 22. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/23. Place Pump.pasta	309	2023.05.03. 4:41	A 23. teszt tesztfájlja.

tests/Basic Actions/maps/ 23.map.pasta	380	2023.05.03. 4:41	A 23. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/ 24. Unable to Place Pump.pasta	166	2023.05.03. 4:41	A 24. teszt tesztfájla.
tests/Basic Actions/maps/ 24.map.pasta	362	2023.05.03. 4:41	A 24. teszthez tartozó pályaleíró fájl.

Fájl neve	Méret	Keletkezés ideje	Tartalom
tests/Attach-Detach Actions/.config.pasta	508	2023.05.10. 22:33	A 11-21. teszteket közösen elindító konfigfájl.
tests/Attach-Detach Actions/ 11. Detach both ends of unoccupied Pipe from Pump.pasta	315	2023.05.02. 22:00	A 11. teszt tesztfájla.
tests/Attach-Detach Actions/maps/ 11.map.pasta	402	2023.05.02. 22:00	A 11. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 12. Detach Pipe from Cistern.pasta	279	2023.05.03. 1:52	A 12. teszt tesztfájla.
tests/Attach-Detach Actions/maps/ 12.map.pasta	415	2023.05.03. 1:52	A 12. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 13. Detach Pipe from Spring.pasta	275	2023.05.03. 1:52	A 13. teszt tesztfájla.
tests/Attach-Detach Actions/maps/ 13.map.pasta	342	2023.05.03. 1:52	A 13. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 14. Detach occupied Pipe.pasta	327	2023.05.03. 1:52	A 14. teszt tesztfájla.
tests/Attach-Detach Actions/maps/ 14.map.pasta	379	2023.05.03. 1:52	A 14. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 15. Detach Pipe end while holding another Pipe end.pasta	362	2023.05.03. 1:52	A 15. teszt tesztfájla.
tests/Attach-Detach Actions/maps/ 15.map.pasta	408	2023.05.03. 1:52	A 15. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 16. Detach both ends of new Pipe.pasta	370	2023.05.03. 3:17	A 16. teszt tesztfájla.
tests/Attach-Detach Actions/maps/ 16.map.pasta	403	2023.05.03. 3:17	A 16. teszthez tartozó pályaleíró fájl.

tests/Attach-Detach Actions/ 17. Attach both ends of Pipe.pasta	386	2023.05.03. 4:10	A 17. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/ 17.map.pasta	321	2023.05.03. 4:10	A 17. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 18. Attach both ends of Pipe to same Pump.pasta	238	2023.05.03. 4:10	A 18. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/ 18.map.pasta	186	2023.05.03. 4:10	A 18. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 19. Attach both ends of Pipe to same Cistern.pasta	308	2023.05.03. 4:10	A 19. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/ 19.map.pasta	201	2023.05.03. 4:10	A 19. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 20. Attach both ends of Pipe to same Spring.pasta	246	2023.05.03. 4:10	A 20. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/ 20.map.pasta	196	2023.05.03. 4:10	A 20. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 21. Attach Pipe to Pump with no free ports.pasta	343	2023.05.03. 4:10	A 21. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/ 21.map.pasta	197	2023.05.03. 4:10	A 21. teszthez tartozó pályaleíró fájl.

Fájl neve	Méret	Keletkezés ideje	Tartalom
tests/Character Move/.config.pasta	262	2023.05.10. 22:33	A 25-32. teszteket közösen elindító konfigfájl.
tests/Character Move/ 25. Move to unoccupied Pipe.pasta	96	2023.05.03. 3:13	A 25. teszt tesztfájlja.
tests/Character Move/maps/ 25.map.pasta	324	2023.05.03. 3:13	A 25. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 26. Move to occupied Pipe.pasta	99	2023.05.03. 3:13	A 26. teszt tesztfájlja.
tests/Character Move/maps/ 26.map.pasta	364	2023.05.03. 3:13	A 26. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 27. Move to detached Pipe.pasta	97	2023.05.03. 3:13	A 27. teszt tesztfájlja.

tests/Character Move/maps/ 27.map.pasta	239	2023.05.03. 3:13	A 27. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 28. Move to Pump.pasta	99	2023.05.03. 3:13	A 28. teszt tesztfájlja.
tests/Character Move/maps/ 28.map.pasta	322	2023.05.03. 3:13	A 28. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 29. Move to Cistern.pasta	103	2023.05.03. 3:13	A 29. teszt tesztfájlja
tests/Character Move/maps/ 29.map.pasta	368	2023.05.03. 3:13	A 29. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 30. Move to Spring.pasta	101	2023.05.03. 3:13	A 30. teszt tesztfájlja.
tests/Character Move/maps/ 30.map.pasta	308	2023.05.03. 3:13	A 30. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 31. Move to banana Pipe.pasta	151	2023.05.03. 3:13	A 31. teszt tesztfájlja.
tests/Character Move/maps/ 31.map.pasta	324	2023.05.03. 3:13	A 31. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 32. Move through technokoled Pipe.pasta	201	2023.05.03. 3:13	A 32. teszt tesztfájlja.
tests/Character Move/maps/ 32.map.pasta	322	2023.05.03. 3:13	A 32. teszthez tartozó pályaleíró fájl.

Fájl neve	Méret	Keletkezés ideje	Tartalom
tests/Controller Actions/.config.pasta	56	2023.05.10. 22:33	A 33., 34. teszteket közösen elindító konfigfájl.
tests/Controller Actions/ 33. Break Pump.pasta	61	2023.05.03. 4:51	A 33. teszt tesztfájlja.
tests/Controller Actions/maps/ 33.map.pasta	32	2023.05.03. 4:51	A 33. teszthez tartozó pályaleíró fájl.
tests/Controller Actions/ 34. Spawn Pipe.pasta	241	2023.05.03. 4:51	A 34. teszt tesztfájlja.
tests/Controller Actions/maps/ 34.map.pasta	102	2023.05.03. 4:51	A 34. teszthez tartozó pályaleíró fájl.

Fájl neve	Méret	Keletkezés ideje	Tartalom
tests/Element Tick/.config.pasta	289	2023.05.10. 22:33	A 35-42. teszteket közösen elindító konfigfájl.
tests/Element Tick/ 35. Waterflow from Spring.pasta	89	2023.05.03. 1:01	A 35. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/ 35.map.pasta	349	2023.05.03. 1:01	A 35. teszthez tartozó pályaleíró fájl.

tests/Element Tick/ 36. Waterflow through Pump.pasta	286	2023.05.03. 1:43	A 36. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/ 36.map.pasta	610	2023.05.03. 1:43	A 36. teszthez tartozó pályaleíró fájl.
tests/Element Tick/ 37. Waterflow into Cistern.pasta	148	2023.05.03. 2:02	A 37. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/ 37.map.pasta	376	2023.05.03. 2:02	A 37. teszthez tartozó pályaleíró fájl.
tests/Element Tick/ 38. Waterflow into full Pipe.pasta	129	2023.05.03. 2:07	A 38. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/ 38.map.pasta	349	2023.05.03. 2:07	A 38. teszthez tartozó pályaleíró fájl.
tests/Element Tick/ 39. Waterflow into detached Pipe.pasta	141	2023.05.03. 2:12	A 39. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/ 39.map.pasta	271	2023.05.03. 2:12	A 39. teszthez tartozó pályaleíró fájl.
tests/Element Tick/ 40. Waterflow into pierced Pipe.pasta	141	2023.05.03. 2:16	A 40. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/ 40.map.pasta	348	2023.05.03. 2:16	A 40. teszthez tartozó pályaleíró fájl.
tests/Element Tick/ 41. Tick Banana Pipe.pasta	111	2023.05.03. 2:37	A 41. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/ 41.map.pasta	116	2023.05.03. 2:37	A 41. teszthez tartozó pályaleíró fájl.
tests/Element Tick/ 42. Tick Technokoled Pipe.pasta	117	2023.05.03. 2:46	A 42. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/ 42.map.pasta	116	2023.05.03. 2:46	A 42. teszthez tartozó pályaleíró fájl.

10.1.2 Fordítás

A fordítás elvégezhető a kari felhő gépén, illetve saját gépen is, amennyiben van telepítve a számítógépre legalább 18-as verziójú Java JDK.

Fordítás lépései:

- A forrásfájlokat tartalmazó **ZIP fájlt kicsomagolni** tetszőleges helyre.
- **Parancessorban elnavigálni** arra a mappába, amiben az “src” mappa található. (cd használatával)
- **Futtatni az alábbi kódot**, ami egyszerre lefordítja a prototípus és a tesztelő programot is:
`javac -cp src src/macaroni/Main.java src/pasta/Main.java -d out`

10.1.3 Prototípus futtatása

A futtatás végezhető a kari felhő gépen, illetve saját gépen is, amennyiben van telepítve a számítógépre legalább 18-as verziójú Java JDK.

Futtatás lépései:

- **Fordítás elvégzése**
- **Parancssorban elnavigálni** abba a mappába, amiben az “src” mappa található. (cd használatával)
- **Futtatni az alábbi kódot**, ami elindítja a prototípus programot:
`java -cp out macaroni.Main`

10.1.4 Tesztelő futtatása

A futtatás végezhető a kari felhő gépen, illetve saját gépen is, amennyiben van telepítve a számítógépre legalább 18-as verziójú Java JDK.

Futtatás lépései:

- **Fordítás elvégzése**
- **Parancssorban elnavigálni** abba a mappába, amiben az “src” mappa található. (cd használatával), és azon belül az “out” mappába navigálni
- **Tesztelő program elindítása** (<fájl> helyére egy tesztfájl vagy konfigfájl abszolút vagy relatív elérési útját kell megadni, **ha space-t tartalmaz akkor idézőjelek közé**):
`java pasta.Main <fájl>`
- Példaképpen az összes teszt lefuttatása párhuzamosan:
`java pasta.Main ../../tests/all.config.pasta`

10.2 Tesztek jegyzőkönyvei

10.2.1 Repair Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.2 Repair Pump

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.3 Pierce Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.4 Pierce unpierceable Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.5 Technokol Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.6 Banana Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:53
Teszt eredménye	(line 5) @pipe.effect => Object not found
Lehetséges hibaok	6.map.pasta fájlban elírás
Változtatások	6.map.pasta fájlban a szabotör <i>location</i> -je <i>pipe</i> -ra cserélve <i>pump</i> helyett.

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.14. 0:28

10.2.7 Set Pump input

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.8 Set Pump output

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.9 Unable to Set Pump input

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.10 Unable to Set Pump output

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.11 Detach both ends of unoccupied Pipe from Pump

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.12 Detach Pipe from Cistern

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.13 Detach Pipe from Spring

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.14 Detach occupied Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.15 Detach Pipe end while holding another Pipe end

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:50

10.2.16 Detach both ends of new Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.17 Attach both ends of Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.18 Attach both ends of Pipe to same Pump

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.19 Attach both ends of Pipe to same Cistern

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.20 Attach both ends of Pipe to same Spring

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.21 Attach Pipe to Pump with no free ports

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.22 Acquire Pump

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.23 Place Pump

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.24 Unable to Place Pump

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.25 Move to unoccupied Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.26 Move to occupied Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.27 Move to detached Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.28 Move to Pump

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.29 Move to Cistern

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.30 Move to Spring

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.31 Move to banana Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:54
Teszt eredménye	(line 7) @saboteur.location => Expected: pipe, Got: pump1
Lehetséges hibaok	Az engedélyezhető determinisztikusság nem lett implementálva.
Változtatások	Engedélyezhető determinisztikusság implementálása.

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.14. 2:03
Teszt eredménye	(line 7) @saboteur.location => Expected: pipe, Got: pump1
Lehetséges hibaok	Elírás a 31. Move to banana Pipe.pasta fájlban.
Változtatások	A szabotőr elvárt helyének pipe-ról pump2-re cserélése.

Tesztelő neve	Tóth Gábor Tamás
Teszt időpontja	2023.05.14. 15:43
Teszt eredménye	(line 7) @saboteur.location => Expected: pump2, Got: pump1
Lehetséges hibaok	Az engedélyezhető determinisztikusság hibásan működik, véletlenszerűen választja azt, hogy a cső melyik végére csúszik a játékos.
Változtatások	Az engedélyezhető determinisztikusság javítása.

Tesztelő neve	Tóth Gábor Tamás
Teszt időpontja	2023.05.14. 15:49

10.2.32 Move through technokoled Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:55
Teszt eredménye	(line 13) @plumber.location => Expected: pump2, Got: pipe
Lehetséges hibaok	Az engedélyezhető determinisztikusság nem lett implementálva.
Változtatások	Engedélyezhető determinisztikusság implementálása.

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.14. 1:58

10.2.33 Break Pump

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:51

10.2.34 Spawn Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:52

10.2.35 Waterflow from Spring

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:52

10.2.36 Waterflow through Pump

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:52

10.2.37 Waterflow into Cistern

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:52

10.2.38 Waterflow into full Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:52

10.2.39 Waterflow into detached Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:52

10.2.40 Waterflow into pierced Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:52

10.2.41 Tick Banana Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:56
Teszt eredménye	(line 9) @pipe.effect => Expected: None, Got: Banana
Lehetséges hibaok	Az engedélyezhető determinisztikusság nem lett implementálva.
Változtatások	Engedélyezhető determinisztikusság implementálása.

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.14. 1:58

10.2.42 Tick Technokoled Pipe

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.13. 22:56
Teszt eredménye	(line 9) @pipe.effect => Expected: None, Got: Technokol

Lehetséges hibaok	Az engedélyezhető determinisztikusság nem lett implementálva.
Változtatások	Engedélyezhető determinisztikusság implementálása.

Tesztelő neve	Szilczer Ármin
Teszt időpontja	2023.05.14. 1:58

10.3 Értékelés

Tag neve	Tag neptun	Munka százalékban
Galgóczy Gábor	QS2WJR	20
Harsányi Levente	FD7NQC	20
Kereszthury Péter	PUKAU0	20
Szilczer Ármin	THGPZB	20
Tóth Gábor Tamás	HAFLCT	20

10.4 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2023.04.23. 16:30	2 óra	Galgóczy	Mozgás átdolgozása és effektek implementálása
2023.04.26. 16:00	4 óra	Kereszthury	Parancsfeldolgozó alapjainak elkészítése. Create, list, help, attach, detach parancsok implementálása
2023.04.27. 19:00	4 óra	Kereszthury	Maradék parancsok implementálása, a kivéve load és save.
2023.05.01. 22:00	2 óra	Szilczer	Javadoc átnézése, javítása.
2023.05.02. 0:00	1 óra	Tóth	ObjectFactory osztály átdolgozása
2023.05.02. 13:00	3 óra	Tóth	Load és save parancsok implementálása.
2023.05.03. 5:00	1 óra	Galgóczy	Tesztelést támogató szoftver tervének dokumentálása
2023.05.06. 16:00	4 óra	Galgóczy	Tesztelő szoftver kódolása
2023.05.06. 21:30	2 óra	Galgóczy	Tesztelő szoftver kódolása
2023.05.08. 14:00	5 óra	Galgóczy	Tesztelő szoftver kódolása
2023.05.10. 21:00	1 óra	Galgóczy	Tesztetek migrálása doksból gitre
2023.05.10. 21:00	1 óra	Tóth	Random osztály implementálása
2023.05.10. 22:00	1 óra	Galgóczy	Tesztelő szoftver kódolása
2023.05.11. 23:00	1 óra	Galgóczy	README írás a teszteléshez
2023.05.11. 21:00	0,5 óra	Galgóczy	Tesztelést támogató szoftver tervének dokumentálásának javítása
2023.05.12. 16:00	1,5 óra	Harsányi	10.1.1 Fájllista
2023.05.13. 0:30	1 óra	Galgóczy	Tesztelést támogató szoftver kódjának kommentezése
2023.05.13. 22:30	1 óra	Szilczer	Összes teszt egyszeri futtatása, dokumentálása (10.2)
2023.05.13. 23:00	2 óra	Tóth	Random osztály determinisztikus módjának javítása.
2023.05.14. 0:00	0,5 óra	Szilczer	6-os teszt javítása
2023.05.14. 15:30	0,5 óra	Tóth	31-es teszt újrafuttatása, probléma javítása.
2023.05.14. 16:00	2 óra	Tóth	Teljes projekt javadoc-jának átnézése, formai javítása.
2023.05.14. 22:00	1 óra	Tóth	10.1.2, 10.1.3, 10.1.4 megírása, fordítás és futtatás működésének ellenőrzése
2023.05.14. 23:00	0,5 óra	Tóth	10.1.1 méretek kitöltése, 10.1 és 10.2 átnézése

11. Grafikus felület specifikációja

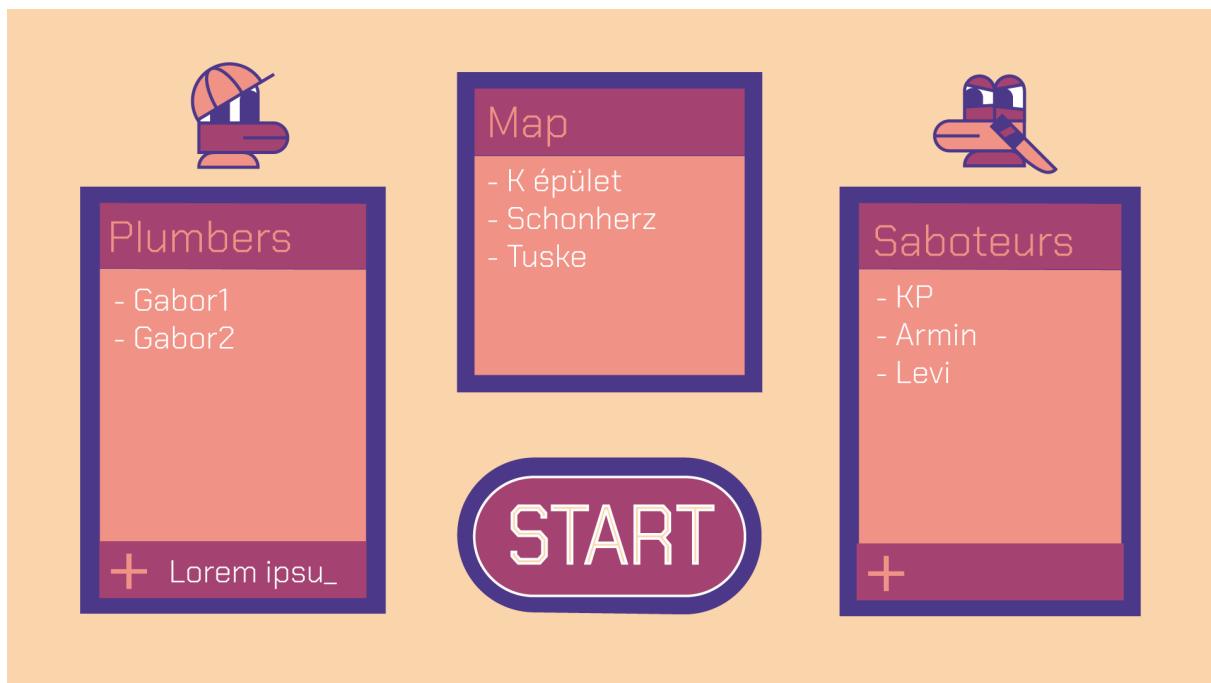
11.1 A grafikus interfész

11.1.1 Főmenü nézet

A program elindításakor a főmenü jelenik meg (lásd alább).

Ebben a menüben kell kiválasztani, milyen pályán szeretnénk játszani, és játékosokat felvenni a 2 csapatba (szerelők és szabotörök). Alul elsőnek az új játékos nevét kell leírni, majd a plusz jelre kattintani a felvételhez.

Végül a Start gombbal indítható a játék, ha már ki van választva a pálya, és minden csapatba legalább két karakter fel van véve. Ennek hatására megjelenik a Játék nézete.

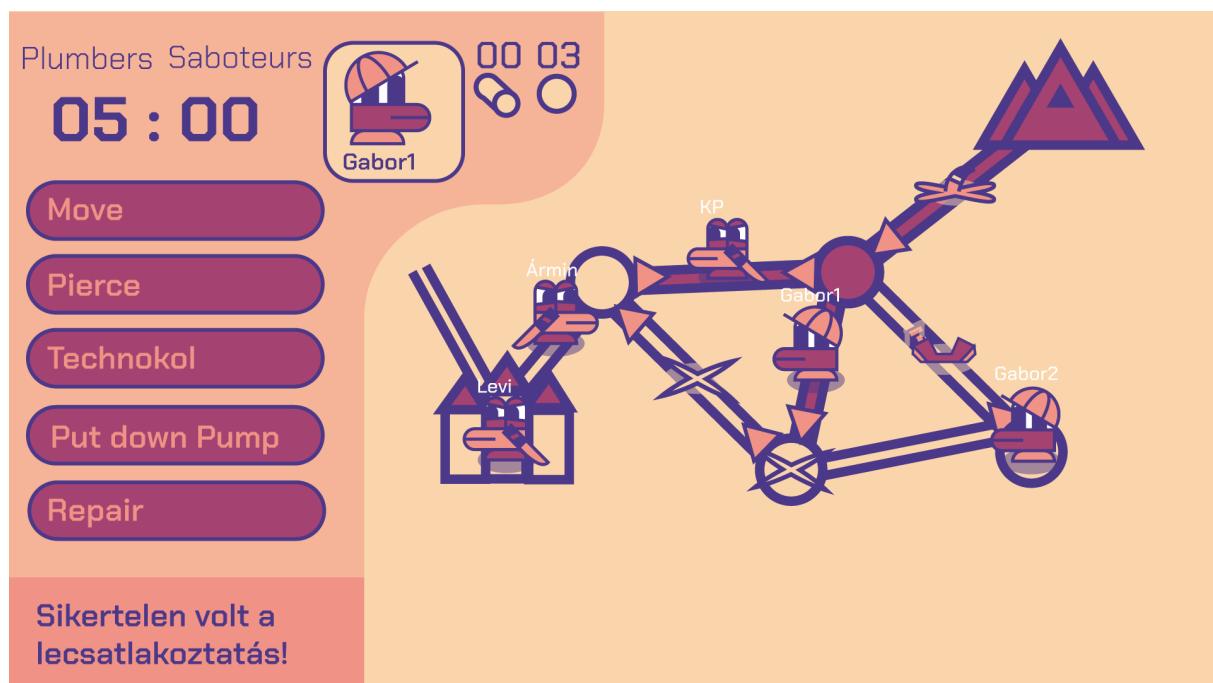


11.1.2 Játék nézet

A játék indításakor megjelenik a játék nézet (lásd alább), amin látható a két csapat jelenlegi pontszáma (gyűjtött vize), az éppen soron lévő játékos, az általa végezhető műveletek, és egy státusz szöveg.

Jobb oldalt magát a pályát láthatjuk. A források jelölésére hegységet, a pumpák jelölésére karikát, a ciszternák jelölésére várat használunk. Az a cső vagy pumpa, amiben van víz, lila színnel van jelölve, amiben nincsen az pedig bézs színnel. A lyukas/ragasztós/csúszós cső és az elromlott pumpa ikonokkal vannak jelölve. A pumpák ki- és bemeneti csövét kis narancssárga háromszög jelöli.

A játékosok karaktereit szerephez illő bábu ikonokkal jelöljük a pályán. Az egyes karakterek mellett megjelenik a játékos neve is.



11.1.3 Játék vége

Fix mennyiségű játék kör után a játék eredménye lesz megjelenítve, hogy melyik csapat nyert.

Ha a szerelők nyertek:



Ha a szabotőrök nyertek:



11.2 A grafikus rendszer architektúrája

11.2.1 A felület működési elve

Megjelenítés

A megjelenítendő modellbeli osztályokhoz tartoznak View osztályok, amelyek egy modellbeli osztály egy példányát és annak pozíóját tárolják, és a *draw()* függvény hívására kirajzolják a program ablakára az adott példányt (pull modell). A kirajzoláshoz felvételre került a modellbe pár getter függvény is, amiket az adott modell osztály View osztálya információ lekérés céljából meghív.

A jelenleg soron lévő játékos és annak helye alapján (hogy min áll) vannak megjelenítve az elvégezhető műveletek az ablakon.

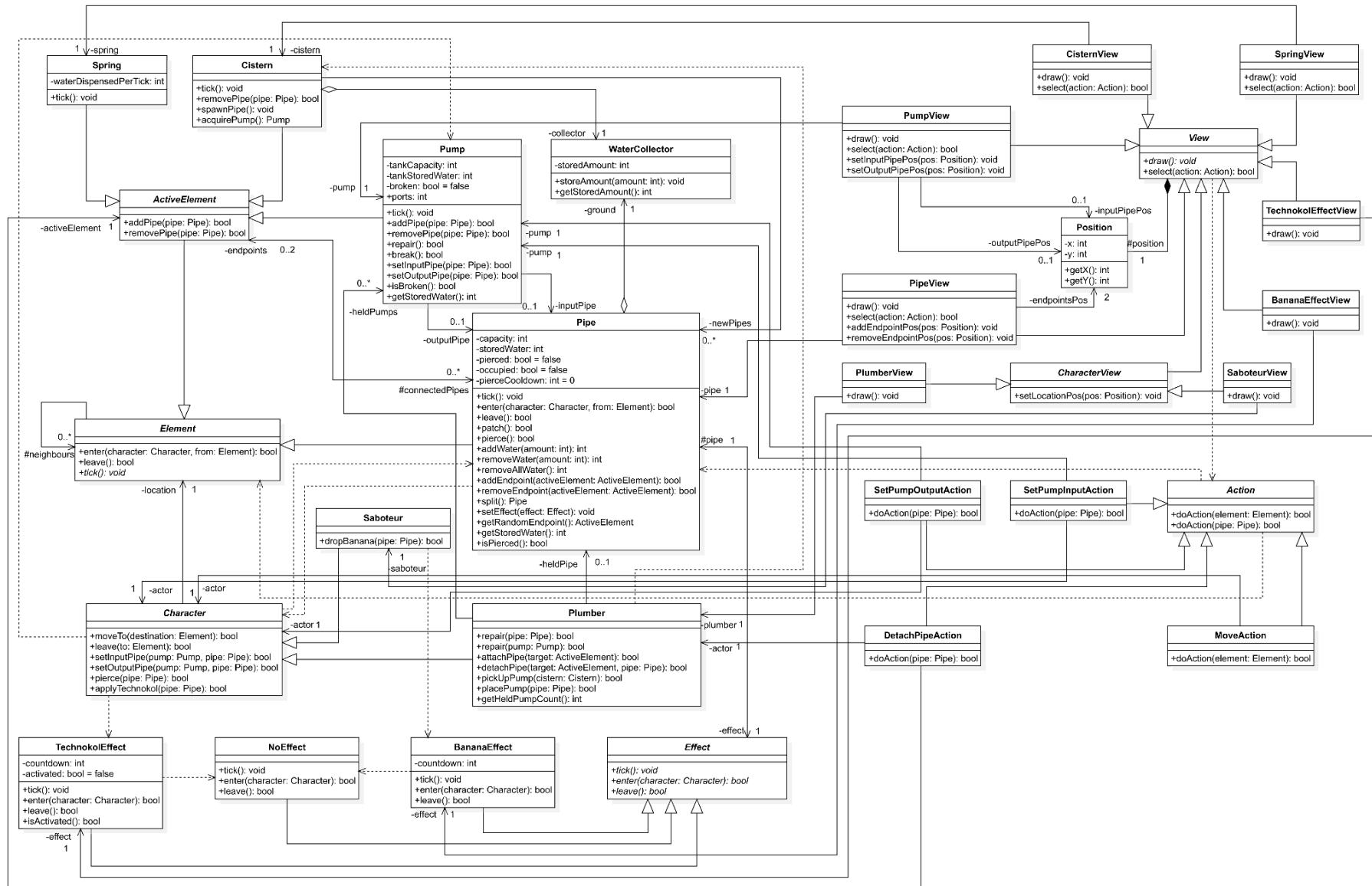
Műveletek végrehajtása

Amelyik művelethez nem szükséges pályaelement kiválasztani (*Repair Pipe*, *Repair Pump*, *Pierce Pipe*, *Attach Pipe*, *Pick up Pump*, *Place Pump*, *Banana Pipe*, *Technokol Pipe*), ahhoz a jelenleg soron lévő játékos és annak helye van felhasználva a művelet végrehajtásához, és a művelet gombnyomásra megtörténik.

Amelyik művelethez szükséges pályaelement kiválasztani (*Move*, *Detach Pipe*, *Set Pump input*, *Set Pump output*), annál a művelet gombnyomására nem történik meg a művelet rögtön, hanem ki kell választani egy megfelelő típusú elemet a pályán kattintással és ekkor megy végbe a művelet. Egyes View osztályokba felvételre került egy *select(Action)* függvény, ami ezen műveletek végrehajtására szolgál.

Sikertelen művelet esetén a státusz mezőbe egy sikertelen üzenet íródik. Ekkor újra lehet próbálkozni a megengedett műveletekkel. Ha egy művelet sikeres volt, akkor a következő játékos kerül sorra, törlődik a státusz mező szövege, majd a játék léptetésre kerül. A játék léptetése a következőkből áll: véletlenszerűen el lesz rontva egy pumpa, véletlenszerűen létre lesz hozva egy ciszternán egy cső, majd az összes pályaelem léptetésre és kirajzolásra kerül.

11.2.2 A felület osztály-struktúrája



11.3 A grafikus objektumok felsorolása

A modell osztályainak változása

11.3.1 Character

- **Metódusok**

- **+bool moveTo(Element destination)**: Igazzal tér vissza, ha sikeres volt.
- **+bool setInputPipe(Pump pump, Pipe pipe)**: Igazzal tér vissza, ha sikeres volt.
- **+bool setOutputPipe(Pump pump, Pipe pipe)**: Igazzal tér vissza, ha sikeres volt.
- **+bool pierce(Pipe pipe)**: Igazzal tér vissza, ha sikeres volt.
- **+bool applyTechnokol(Pipe pipe)**: Igazzal tér vissza, ha sikeres volt.

11.3.2 Element

- **Metódusok**

- **+bool enter(Element from)**: Igazzal tér vissza , ha sikeres volt.

11.3.3 Pipe

- **Metódusok**

- **+bool enter(Character character, Element from)**: Igazzal tér vissza, ha sikeres volt.
- **+bool patch()**: Igazzal tér vissza, ha sikeres volt.
- **+bool pierce()**: Igazzal tér vissza, ha sikeres volt.
- **+bool isPierced()**: Igazzal tér vissza, ha a cső ki van lyukasztva.
- **+int getStoredWater()**: Visszatér a csőben lévő víz mennyiségével.

11.3.4 Plumber

- **Metódusok**

- **+bool repair(Pipe pipe)**: Igazzal tér vissza, ha sikeres volt.
- **+bool repair(Pump pump)**: Igazzal tér vissza, ha sikeres volt.
- **+bool attachPipe(ActiveElement target)**: Igazzal tér vissza, ha sikeres volt.
- **+bool detachPipe(ActiveElement target, Pipe pipe)**: Igazzal tér vissza, ha sikeres.
- **+bool pickUpPump(Cistern cistern)**: Igazzal tér vissza, ha sikeres volt.
- **+bool placePump(Pipe pipe)**: Igazzal tér vissza, ha sikeres volt.
- **+int getHeldPumpCount()**: Visszatér a szerelő kézben tartott pumpáinak számával.

11.3.5 Pump

- **Metódusok**

- **+bool repair()**: Igazzal tér vissza, ha sikeres volt.
- **+bool break()**: Igazzal tér vissza, ha sikeres volt.
- **+bool setInputPipe(Pipe pipe)**: Igazzal tér vissza, ha sikeres volt.
- **+bool setOutputPipe(Pipe pipe)**: Igazzal tér vissza, ha sikeres volt.
- **+bool isBroken()**: Igazzal tér vissza, ha a pumpa el van törve.

11.3.6 Saboteur

- **Metódusok**
 - `+bool dropBanana(Pipe pipe)`: Igazzal tér vissza, ha sikeres volt.

11.3.7 TechnokolEffect

- **Metódusok**
 - `+bool isActivated()`: Igazzal tér vissza, ha az effekt aktiválva van.

11.3.8 WaterCollector

- **Metódusok**
 - `+int getStoredAmount()`: Visszatér a benne tárolt víz mennyiségével.

Az új osztályok felsorolása

11.3.9 Action

- **Felelősség**
Felhasználói interakciók kezelésére használt osztály.
- **Metódusok**
 - `+bool doAction(element: Element)`: minden esetben hamissal tér vissza.
 - `+bool doAction(pipe: Pipe)`: Meghívja önmagán a `doAction(Element)`-et, és annak a visszatérési értékével tér vissza.

11.3.10 MoveAction

- **Felelősség**
Mozgási interakciót kezelő osztály.
- **Ősosztályok**
Action
- **Attribútumok**
 - **-Character actor**: A karakter, ami a mozgást végzi.
- **Metódusok**
 - `+bool doAction(Element element)`: Ha tudja, elvégzi a játékos paraméterként kapott elemre való mozgatását. Igazzal tér vissza ha sikeres.

11.3.11 DetachPipeAction

- **Felelősség**
Cső aktív elemről való lekötési interakciót kezelő osztály.
- **Ősosztályok**
Action

- **Attribútumok**
 - **-Plumber actor:** A szerelő, aki az akciót végzi
 - **-ActiveElement activeElement:** Az aktív elem, amiről lecsatlakoztatja a csövet
- **Metódusok**
 - **+bool doAction(Pipe pipe):** Ha tudja, elvégzi a paraméterként kapott cső lecsatlakoztatását. Igazzal tér vissza ha sikeres.

11.3.12 SetPumpInputAction

- **Felelősség**
Pumpa bemeneti csőállítási interakciót kezelő osztály.
- **Ősosztályok**
Action
- **Attribútumok**
 - **-Character actor:** A karakter, aki az akciót végzi
 - **-Pump pump:** A pumpa, amin az akciót végzi
- **Metódusok**
 - **+bool doAction(Pipe pipe):** Ha tudja, elvégzi a paraméterként kapott cső bemeneti csőnek való beállítását. Igazzal tér vissza ha sikeres.

11.3.13 SetPumpOutputAction

- **Felelősség**
Pumpa kimeneti csőállítási interakciót kezelő osztály.
- **Ősosztályok**
Action
- **Attribútumok**
 - **-Character actor:** A karakter, aki az akciót végzi
 - **-Pump pump:** A pumpa, amin az akciót végzi
- **Metódusok**
 - **+bool doAction(Pipe pipe):** Ha tudja, elvégzi a paraméterként kapott cső kimeneti csőnek való beállítását. Igazzal tér vissza ha sikeres.

11.3.14 Position

- **Felelősség**
A kirajzolt objektumok helyét tartalmazó osztály (x és y koordinátákkal).
- **Attribútumok**
 - **-int x:** Az x koordináta
 - **-int y:** Az y koordináta

- **Metódusok**

- `+int getX()`: Visszatér az x koordináta értékével.
- `+int getY()`: Visszatér az y koordináta értékével.

11.3.15 View

- **Felelősség**

Objektumok grafikus megjelenítéséért és a rajtuk kijelölt akciók elvégzéséért felelős absztrakt osztály.

- **Attribútumok**

- **#Position position:** A pozíció, ahova ki kell rajzolni az objektumot.

- **Metódusok**

- `+void draw()`: A kirajzolásért felelős függvény, absztrakt.
- `+bool select(Action action)`: Amennyiben a felhasználó erre a példányra kattintott, elvégzi a paraméterként kapott action-t. Igazzal tér vissza amennyiben sikeres.

11.3.16 CharacterView

- **Felelősség**

A karakterek megjelenítéséért felelős absztrakt osztály.

- **Ősosztályok**

View → CharacterView

- **Metódusok**

- `+void setLocationPos(Position pos)`: Beállítja a pozícióját a paraméterként kapott pozícióra.

11.3.17 PlumberView

- **Felelősség**

A szerelők megjelenítéséért felelős osztály.

- **Ősosztályok**

View → CharacterView → PlumberView

- **Attribútumok**

- **-Plumber plumber:** A szerelő, akit meg kell jeleníteni

- **Metódusok**

- `+void draw()`: A szerelő megjelenítését végző függvény.

11.3.18 SaboteurView

- **Felelősség**

A szabotőrök megjelenítéséért felelős osztály.

- **Ősosztályok**

View → CharacterView → SaboteurView

- **Attribútumok**

- **-Saboteur saboteur:** A szabotőr, akit meg kell jeleníteni

- **Metódusok**

- **+void draw():** A szabotőr megjelenítését végző függvény.

11.3.19 PipeView

- **Felelősség**

A csövek megjelenítéséért felelős osztály.

- **Ősosztályok**

View → PipeView

- **Attribútumok**

- **-Pipe pipe:** A cső, amit meg kell jeleníteni
- **-Position[2] endpointsPos:** A cső végpontjainak pozíciója. Mindig 2 van belőle, attól függetlenül, hogy a cső valójában csatlakoztatva van-e bármihöz.

- **Metódusok**

- **+void draw():** A cső megjelenítését végző függvény.

11.3.20 PumpView

- **Felelősség**

A pumpák megjelenítéséért felelős osztály.

- **Ősosztályok**

View → PumpView

- **Attribútumok**

- **-Pump pump:** A pumpa, amit meg kell jeleníteni
- **-Position inputPipePos:** A pumpa bemeneti csövének pozíció referenciaja.
- **-Position outputPipePos:** A pumpa kimeneti csövének pozíció referenciaja.

- **Metódusok**

- **+void draw():** A pumpa megjelenítését végző függvény.

11.3.21 CisternView

- **Felelősség**

A ciszternák megjelenítéséért felelős osztály.

- **Ősosztályok**

View → CisternView

- **Attribútumok**
 - **-Cistern cistern:** A ciszterna, amit meg kell jeleníteni
- **Metódusok**
 - **+void draw():** A ciszterna megjelenítését végző függvény.

11.3.22 SpringView

- **Felelősség**

A források megjelenítéséért felelős osztály.

- **Ősosztályok**

View → SpringView

- **Attribútumok**

- **-Spring spring:** A forrás, amit meg kell jeleníteni

- **Metódusok**

- **+void draw():** A forrás megjelenítését végző függvény.

11.3.23 TechnokolEffectView

- **Felelősség**

A technokol effekt megjelenítéséért felelős osztály.

- **Ősosztályok**

View → TechnokolEffectView

- **Attribútumok**

- **-TechnokolEffect effect:** A technokol effekt, amit meg kell jeleníteni

- **Metódusok**

- **+void draw():** A technokol effekt megjelenítését végző függvény.

11.3.24 BananaEffectView

- **Felelősség**

A banán effekt megjelenítéséért felelős osztály.

- **Ősosztályok**

View → BananaEffectView

- **Attribútumok**

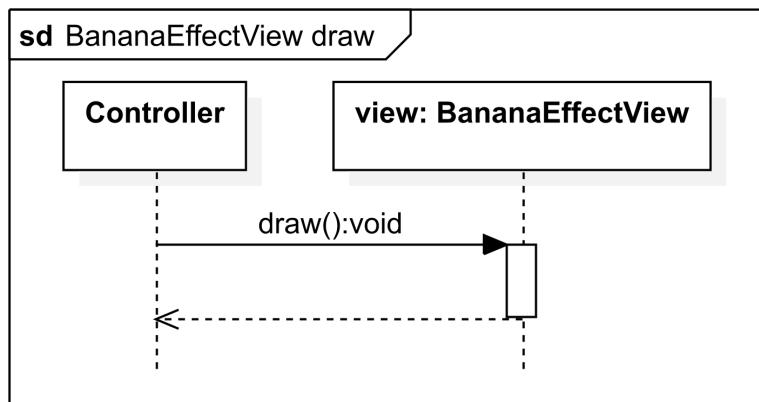
- **-BananaEffect effect:** A banán effekt, amit meg kell jeleníteni

- **Metódusok**

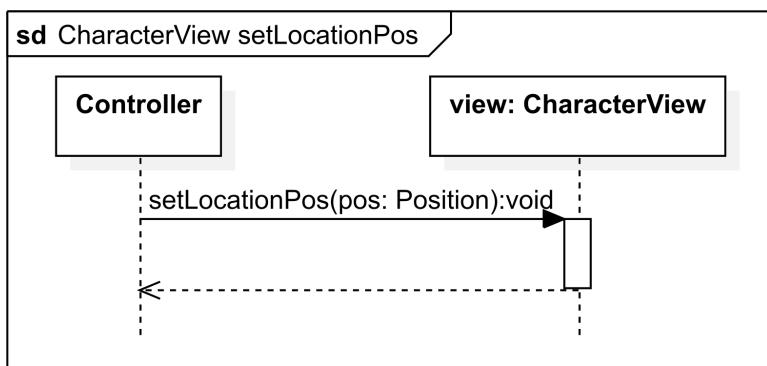
- **+void draw():** A banán effekt megjelenítését végző függvény.

11.4 Kapcsolat az alkalmazói rendszerrel

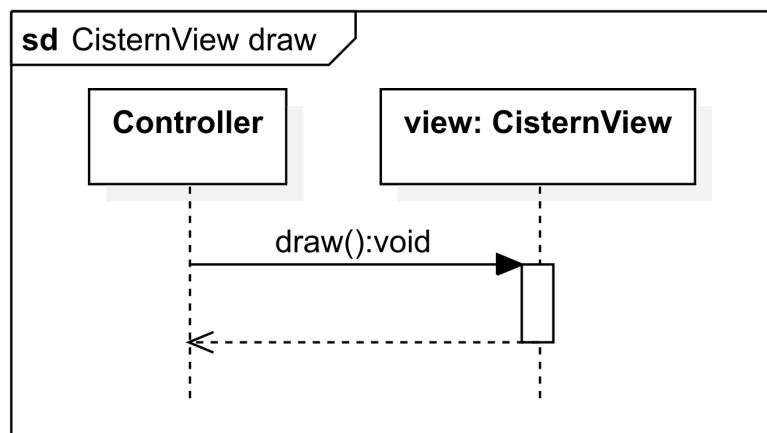
11.4.1 BananaEffectView draw



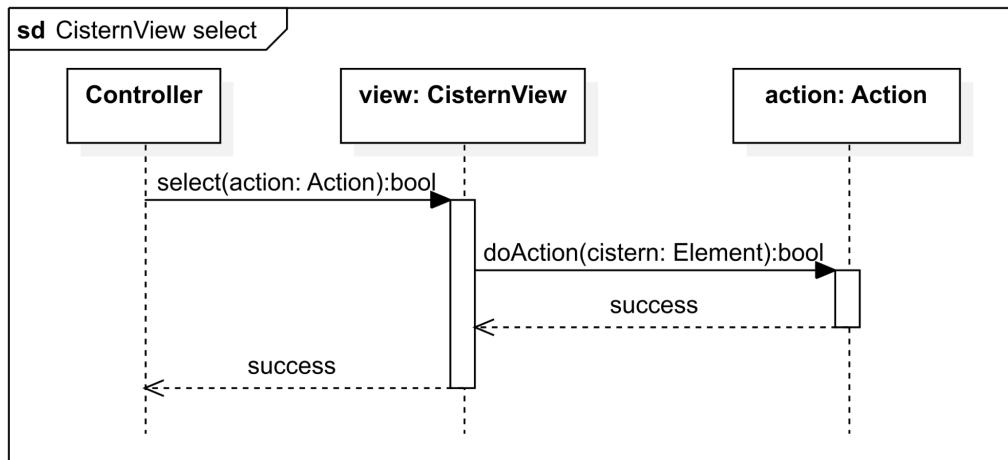
11.4.2 CharacterView setLocationPos



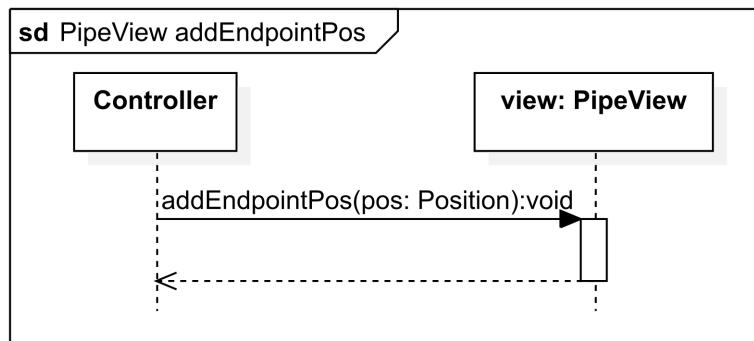
11.4.3 CisternView draw



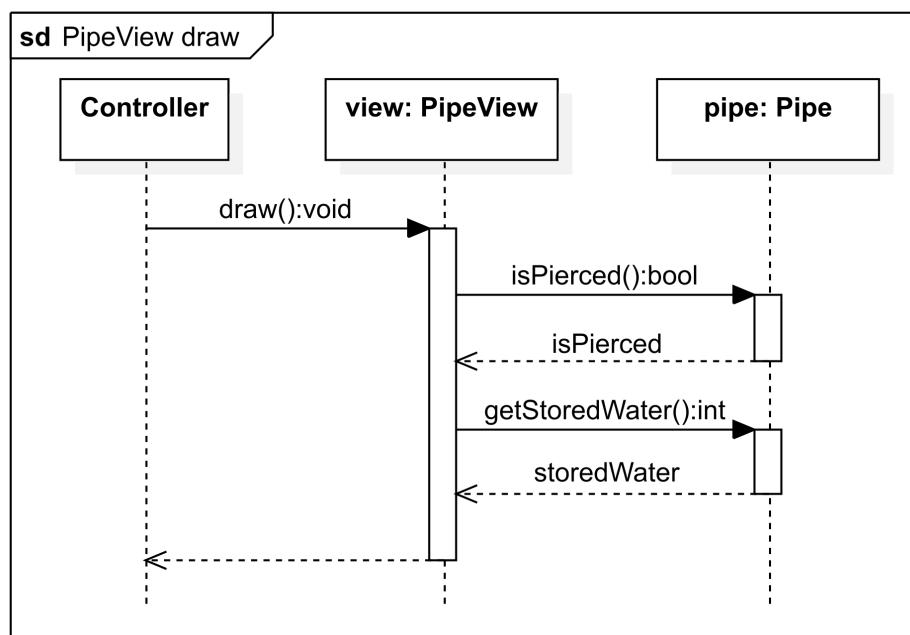
11.4.4 *CisternView select*



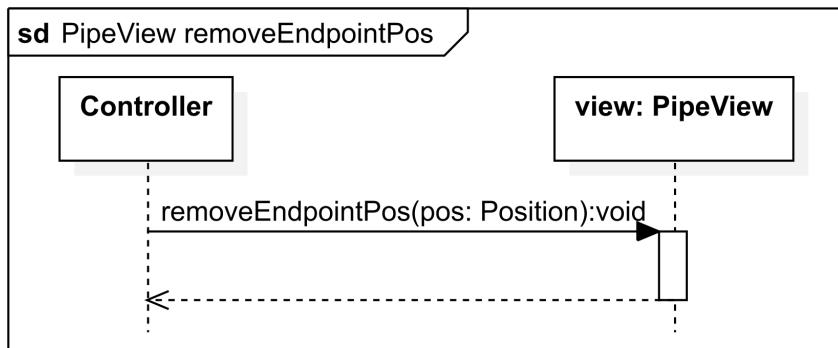
11.4.5 *PipeView addEndpointPos*



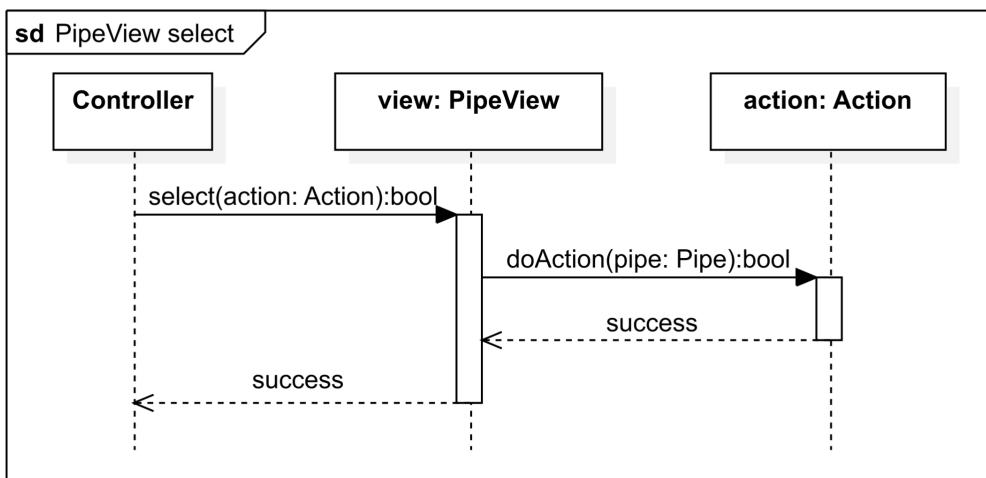
11.4.6 *PipeView draw*



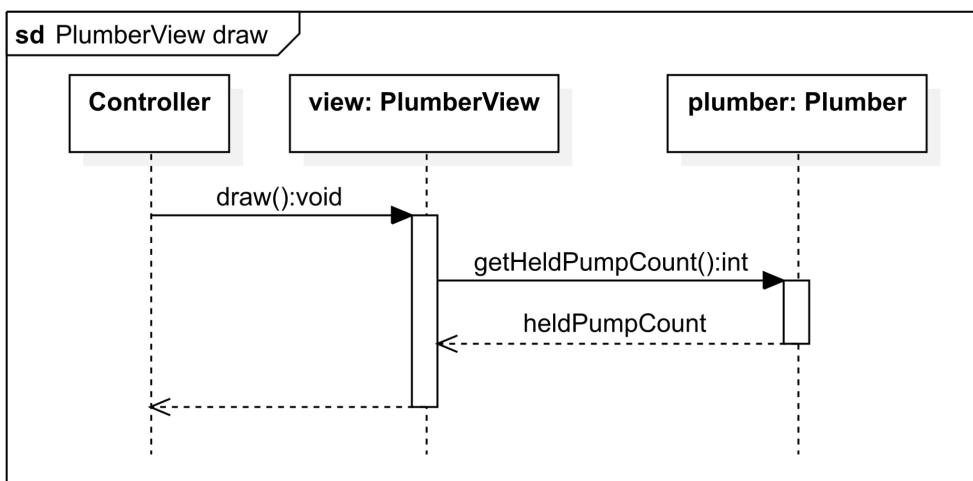
11.4.7 PipeView removeEndpointPos



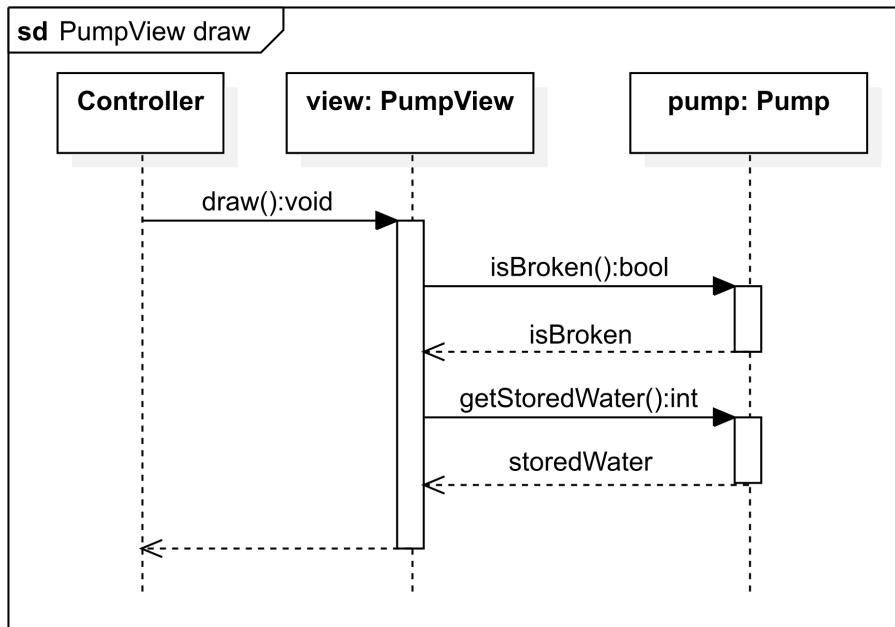
11.4.8 PipeView select



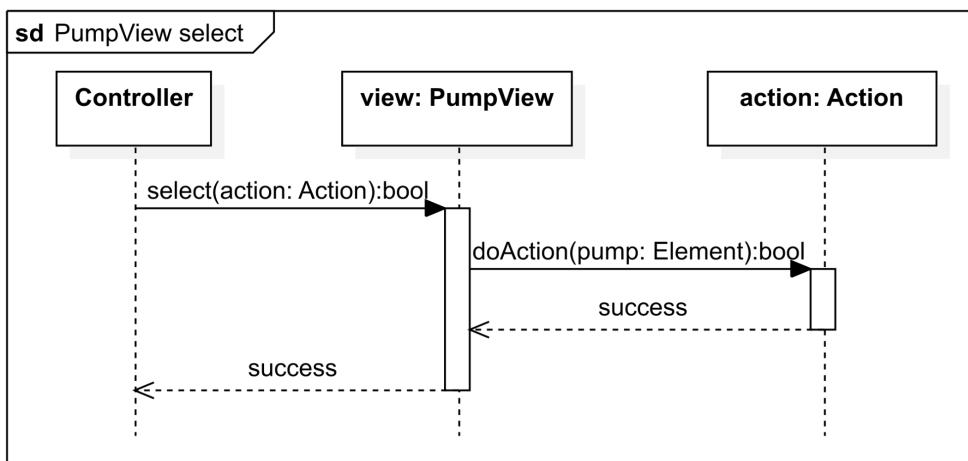
11.4.9 PlumberView draw



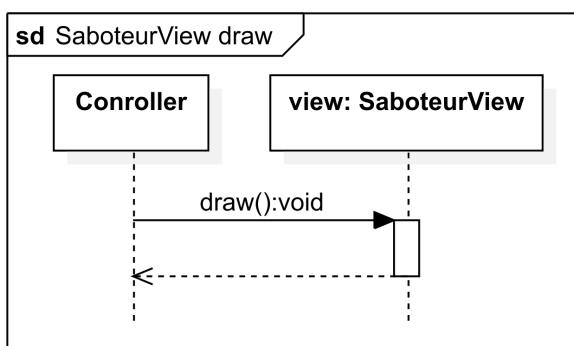
11.4.10 PumpView draw



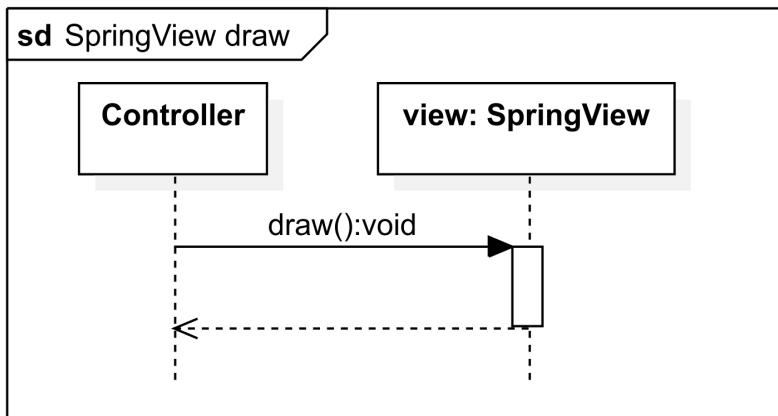
11.4.11 PumpView select



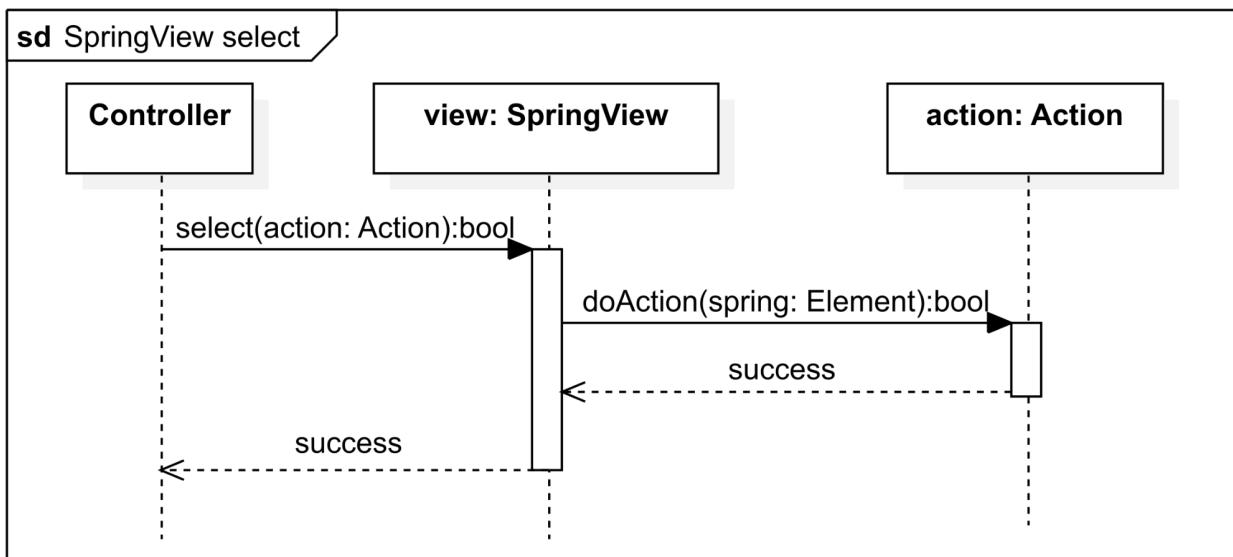
11.4.12 SaboteurView draw



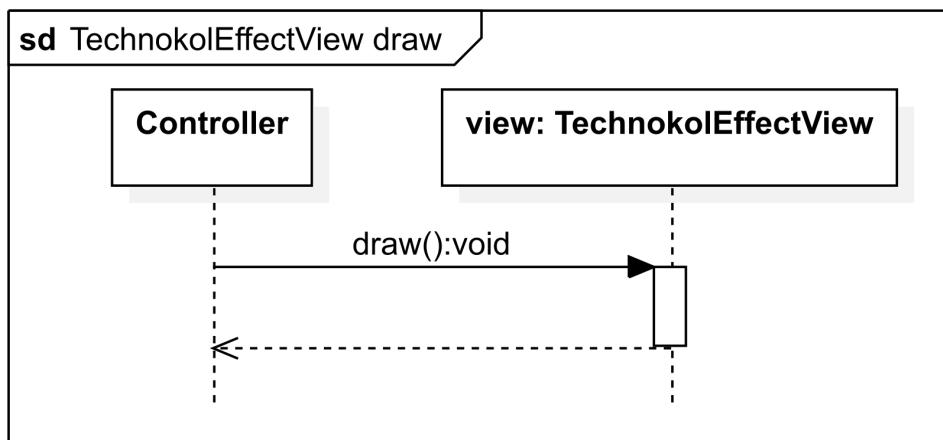
11.4.13 SpringView draw



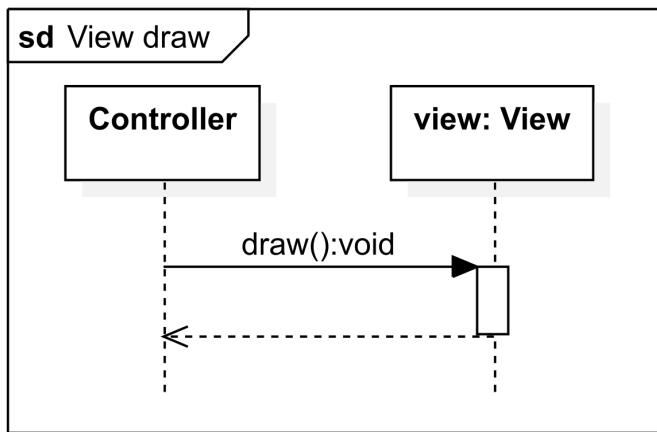
11.4.14 SpringView select



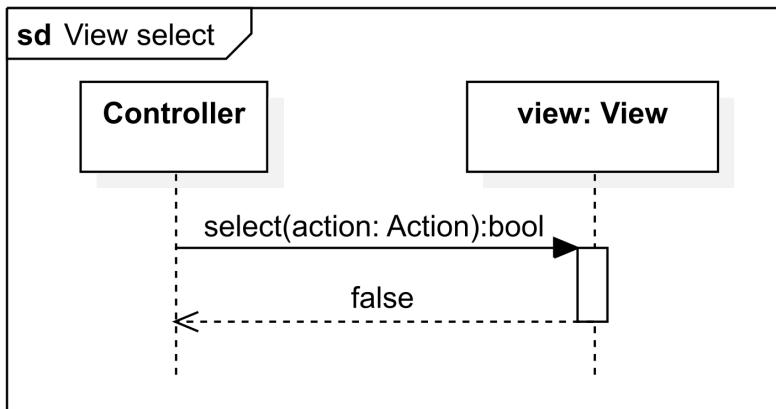
11.4.15 TechnokolEffectView draw



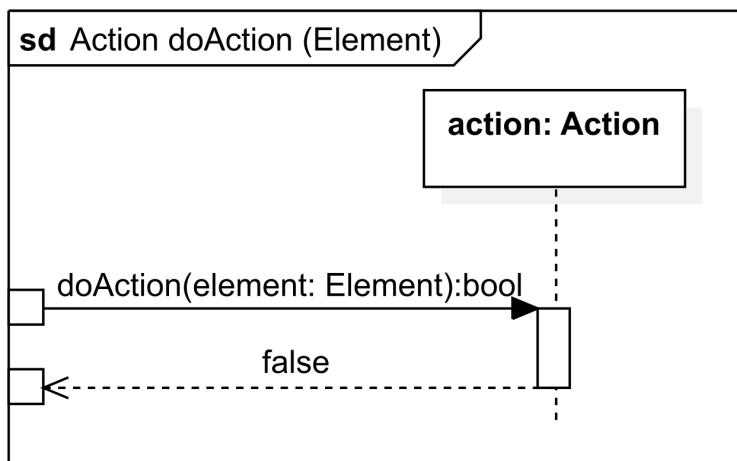
11.4.16 View draw



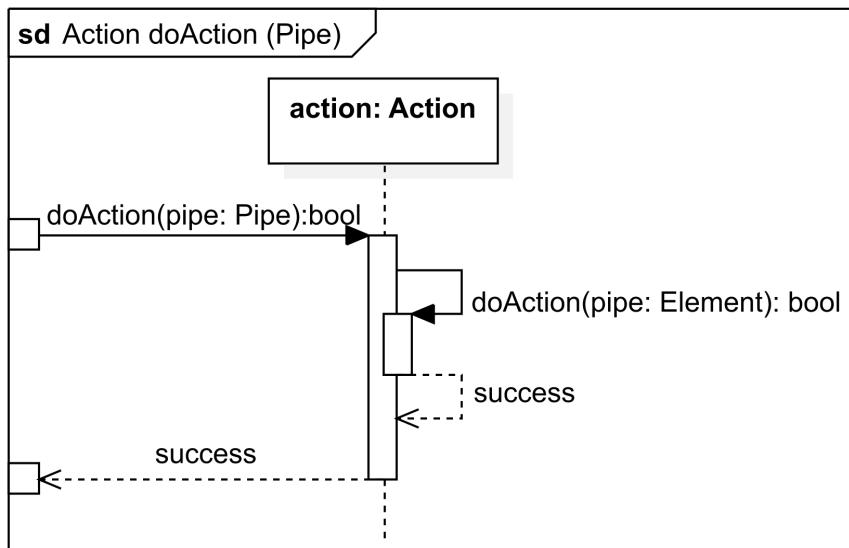
11.4.17 View select



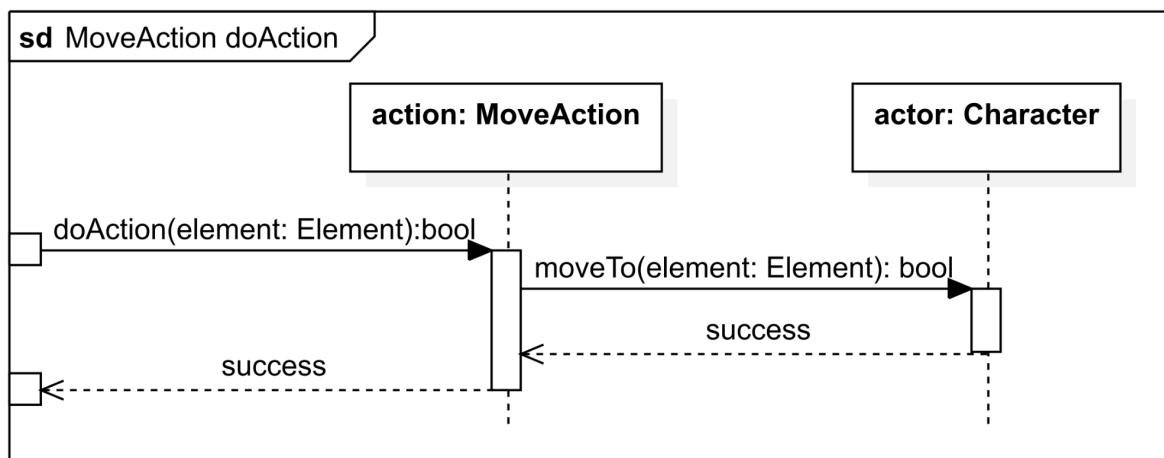
11.4.18 Action doAction (Element)



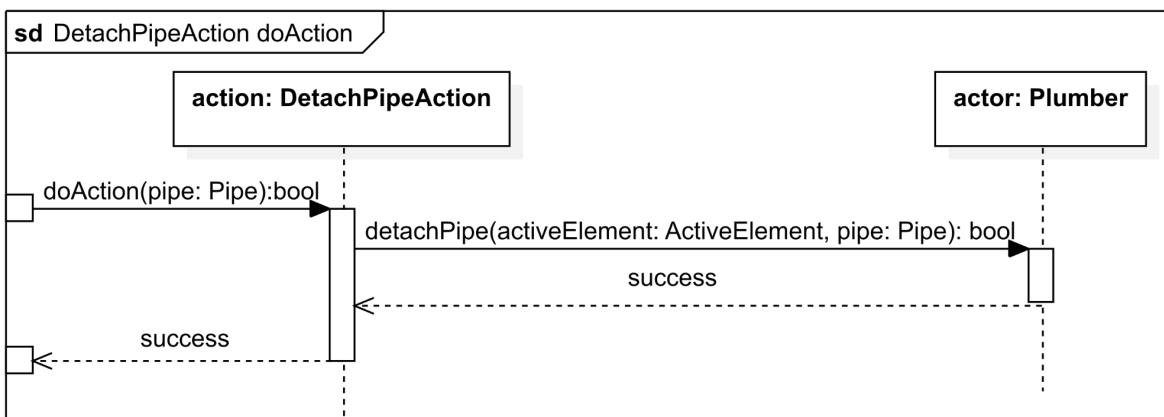
11.4.19 Action doAction (Pipe)



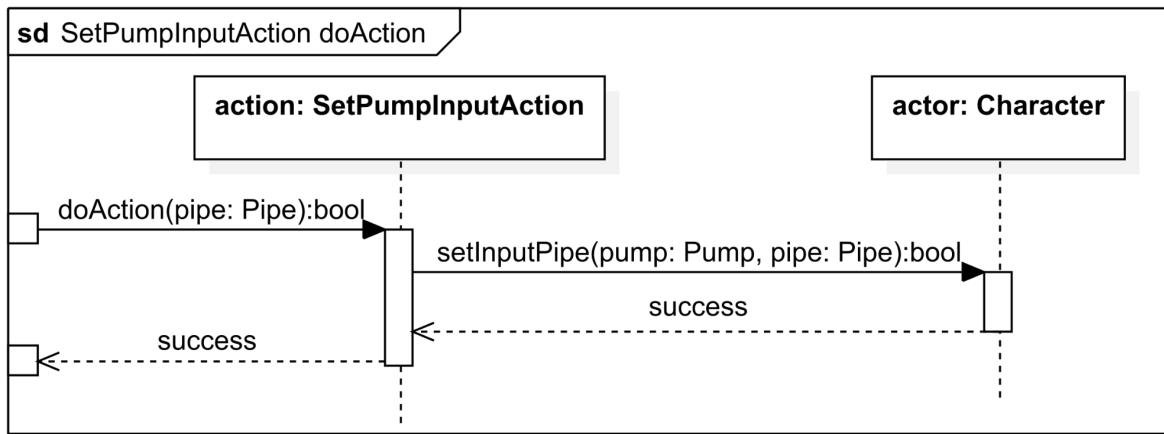
11.4.20 MoveAction doAction



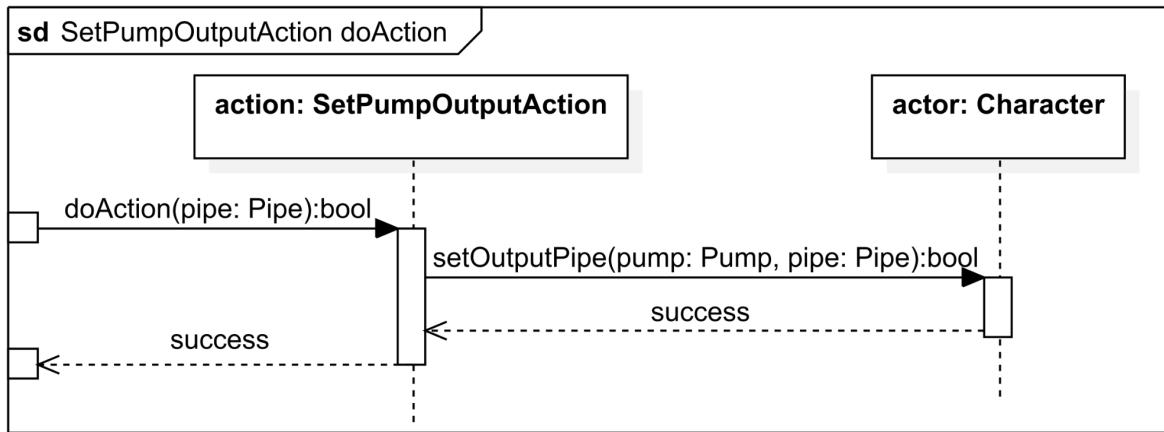
11.4.21 DetachPipeAction doAction



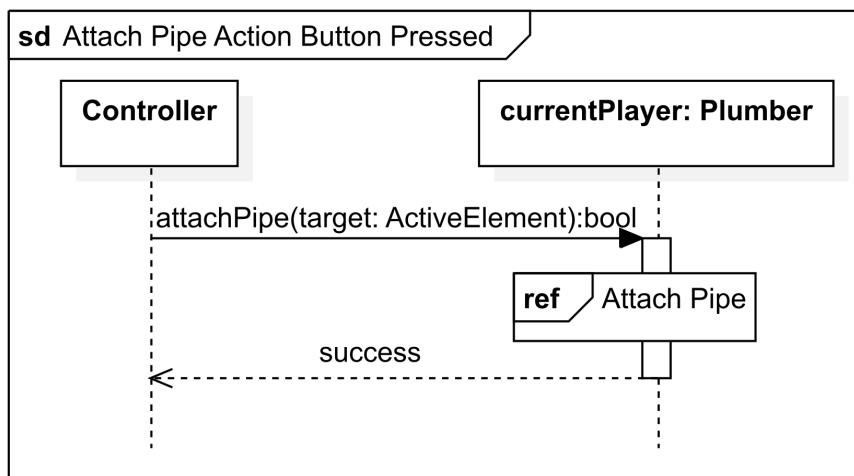
11.4.22 SetPumpInputAction doAction



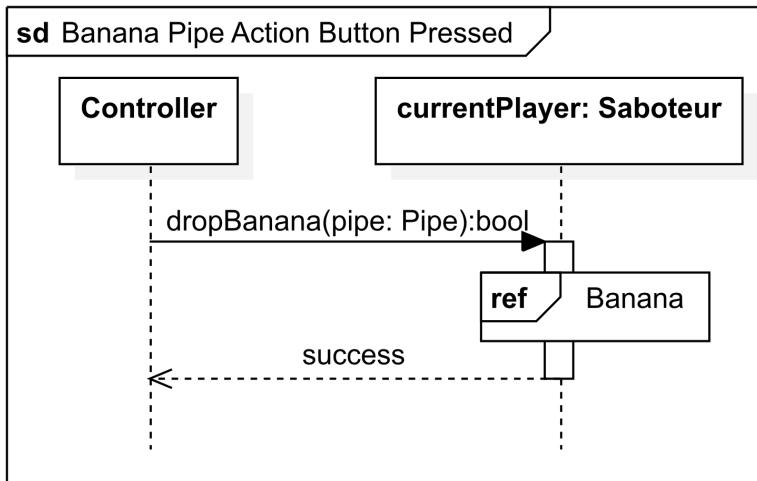
11.4.23 SetPumpOutputAction doAction



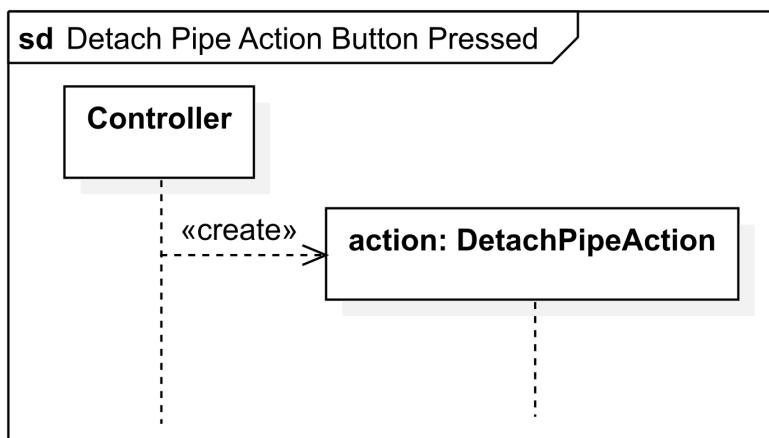
11.4.24 Attach Pipe Action Button Pressed



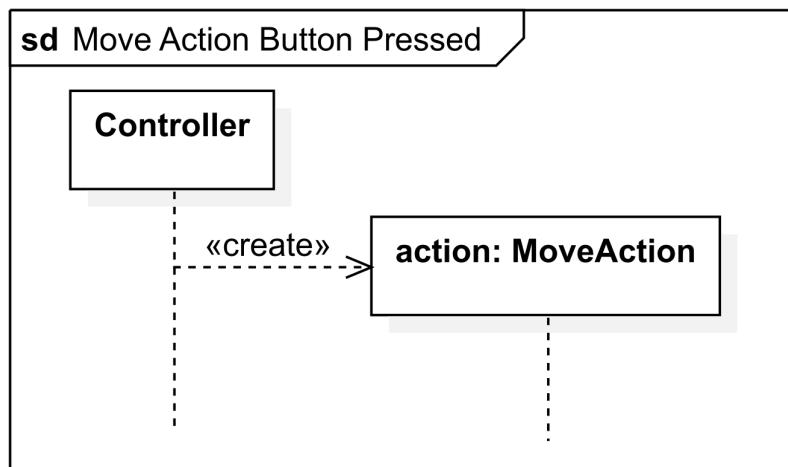
11.4.25 Banana Pipe Action Button Pressed



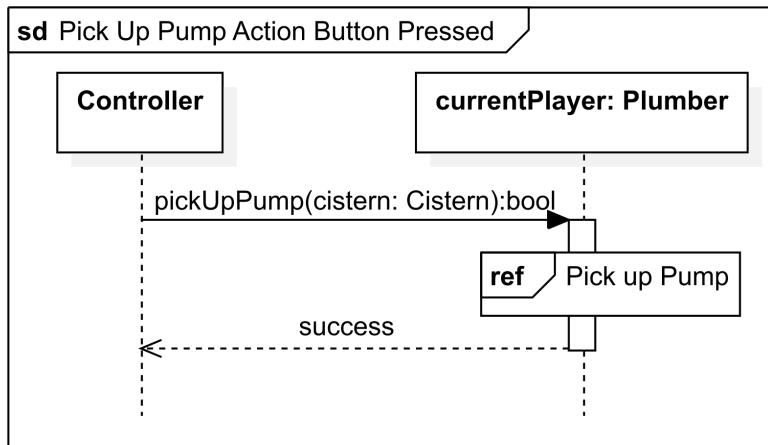
11.4.26 Detach Pipe Action Button Pressed



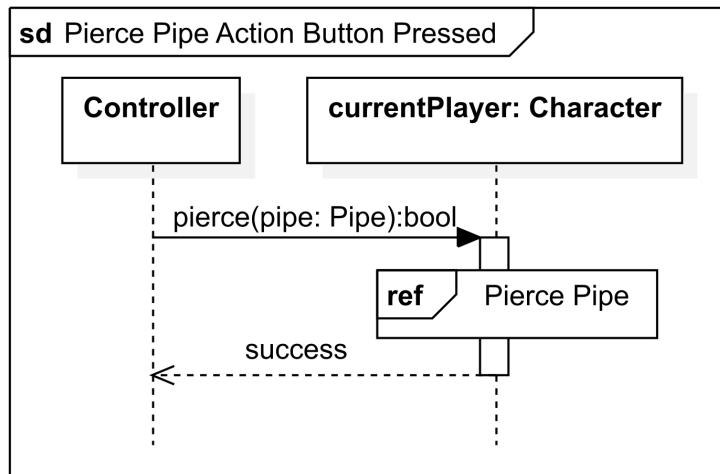
11.4.27 Move Action Button Pressed



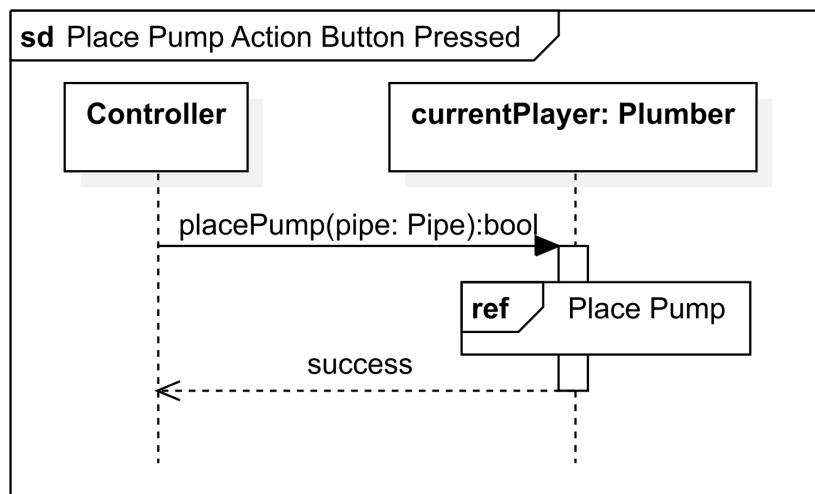
11.4.28 Pick Up Pump Action Button Pressed



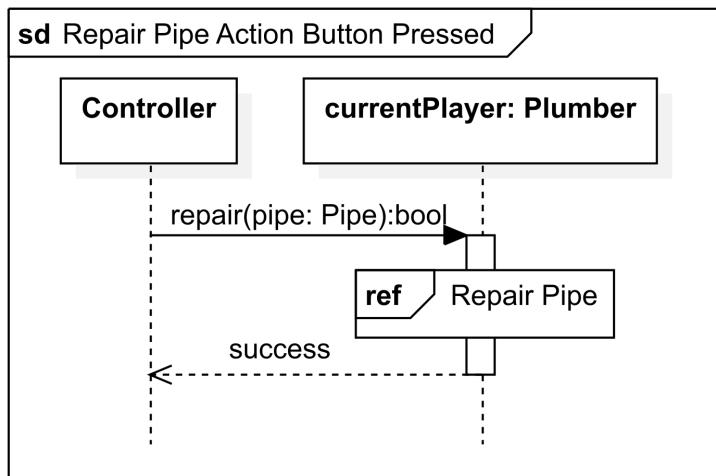
11.4.29 Pierce Pipe Action Button Pressed



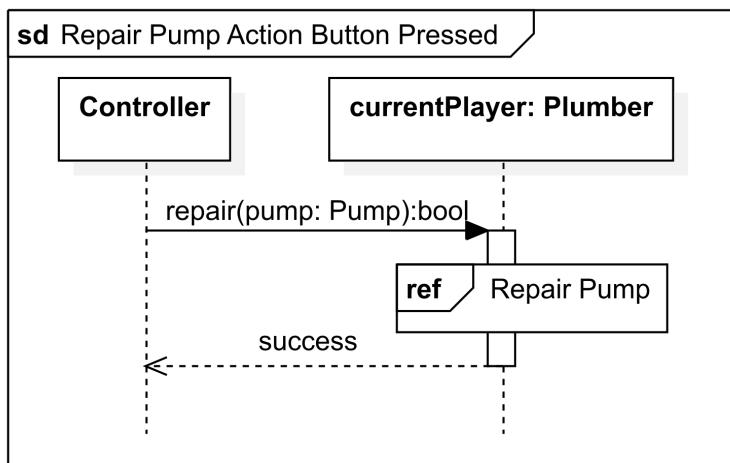
11.4.30 Place Pump Action Button Pressed



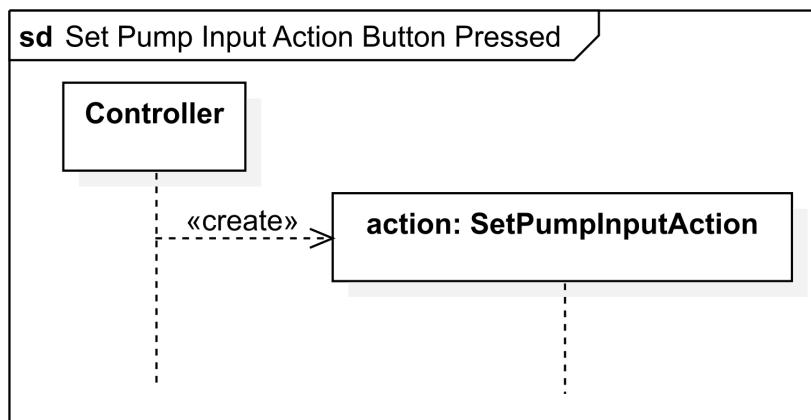
11.4.31 Repair Pipe Action Button Pressed



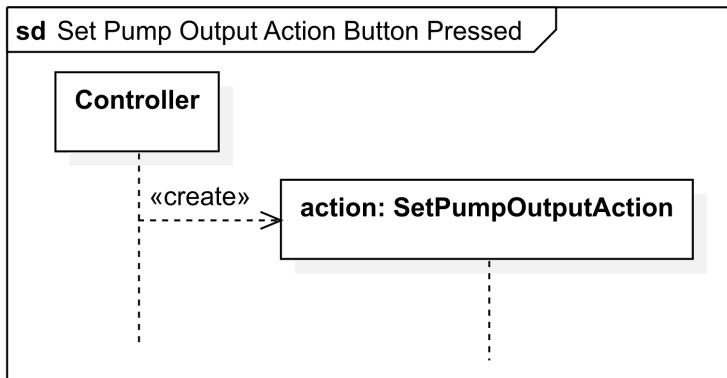
11.4.32 Repair Pump Action Button Pressed



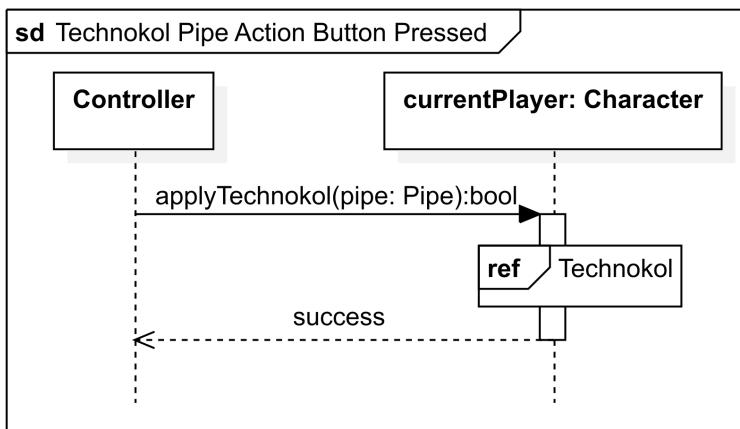
11.4.33 Set Pump Input Action Button Pressed



11.4.34 Set Pump Output Action Button Pressed

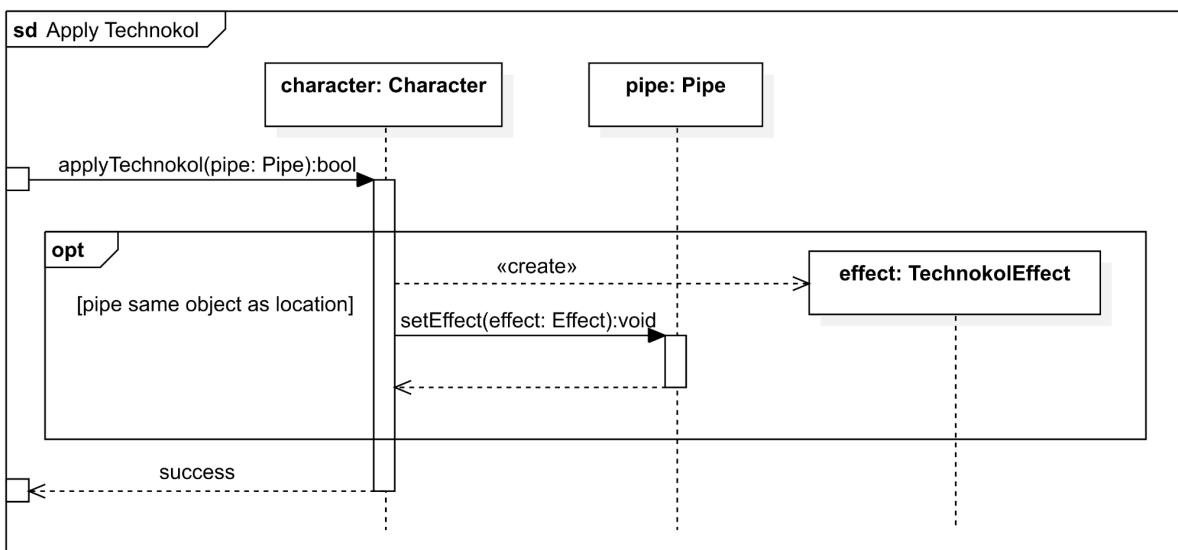


11.4.35 Technokol Pipe Action Button Pressed

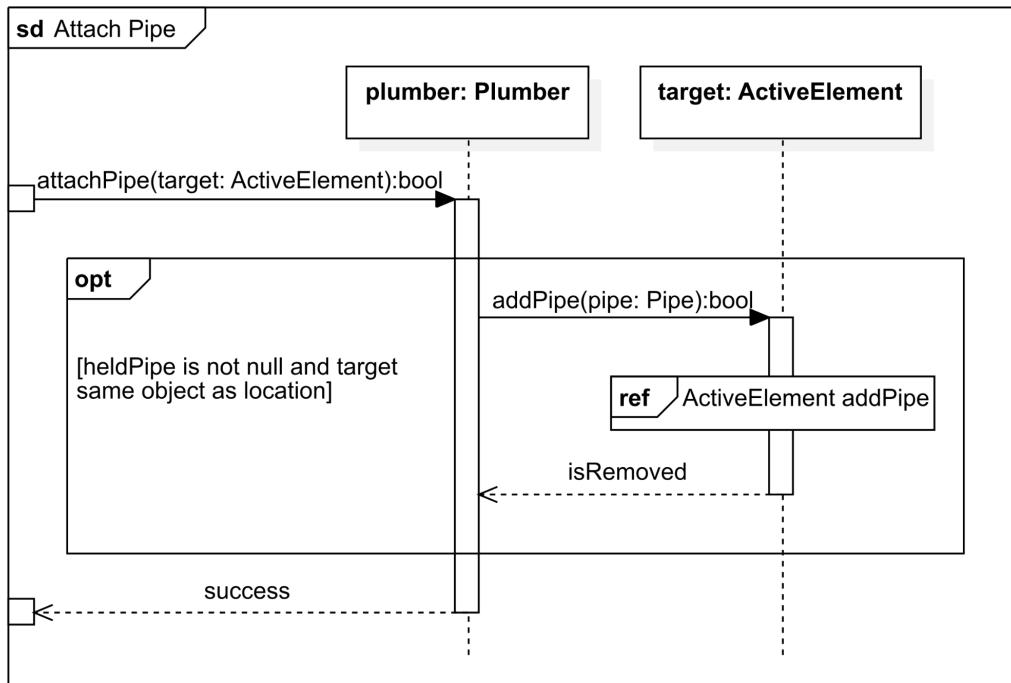


Innentől a modell osztályainak módosításai következnek.

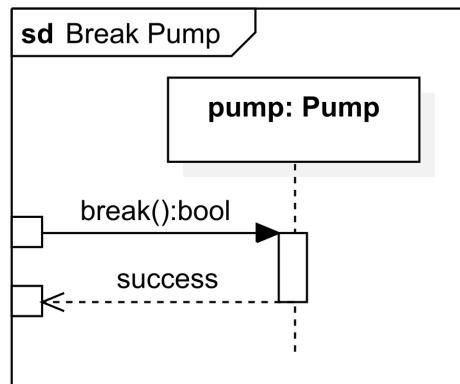
11.4.36 Apply Technokol



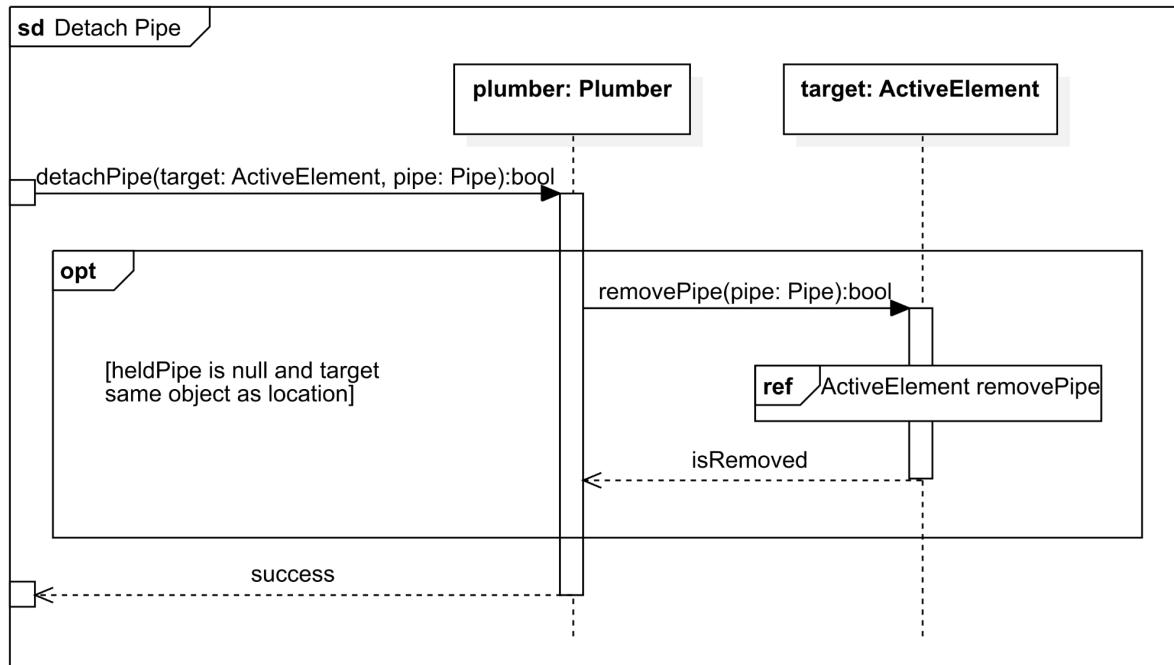
11.4.37 Attach Pipe



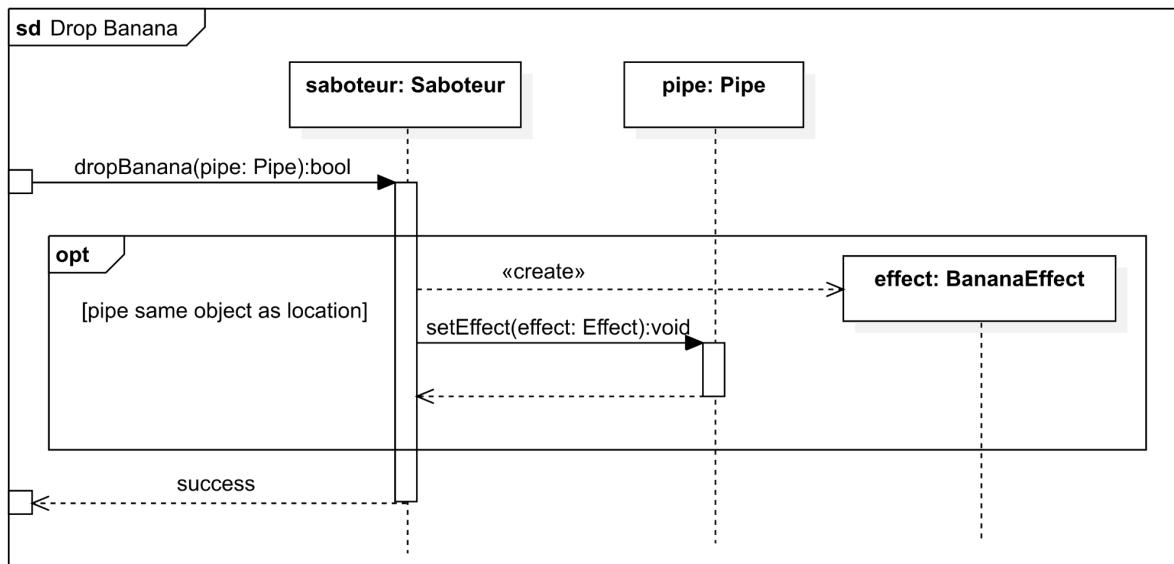
11.4.38 Break Pump



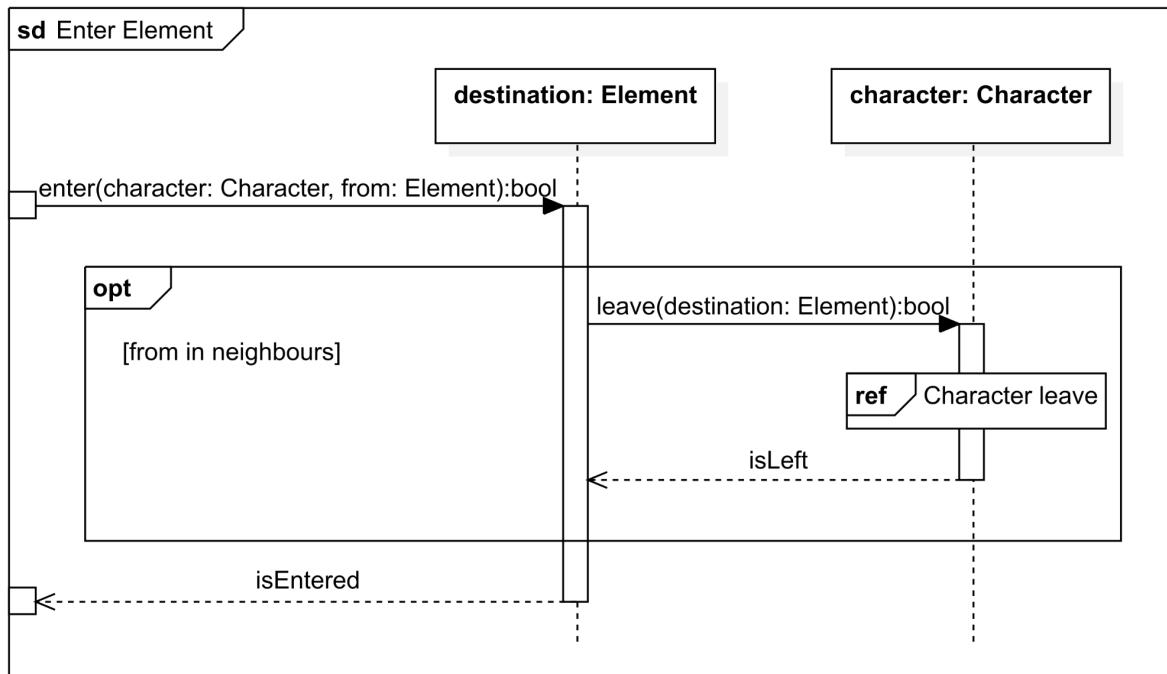
11.4.39 Detach Pipe



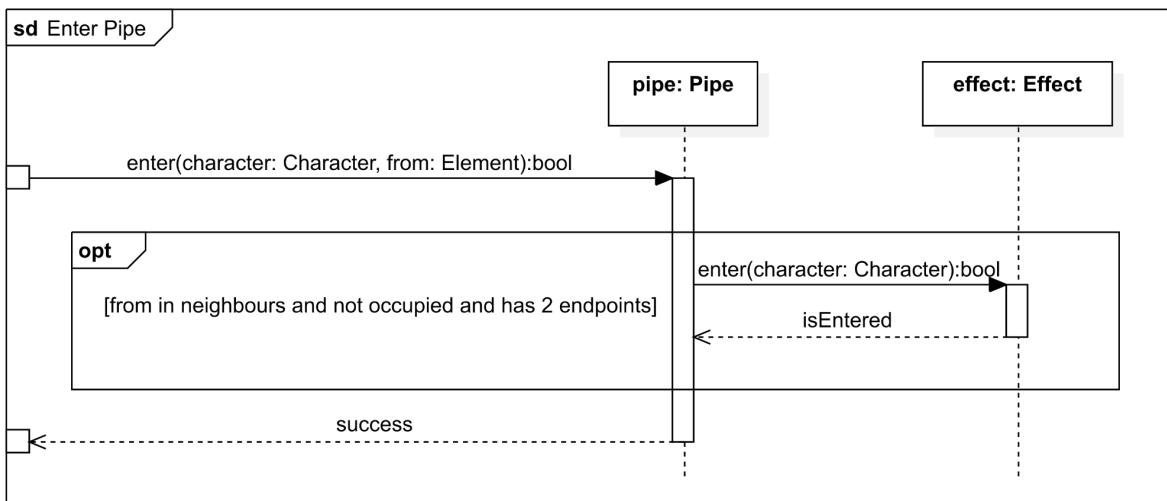
11.4.40 Drop Banana



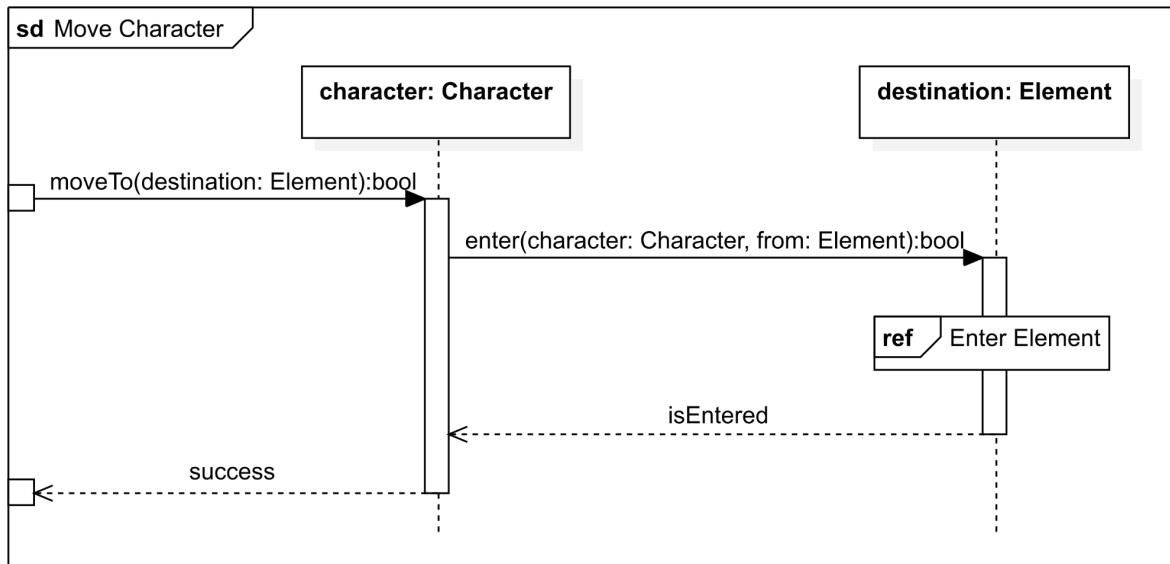
11.4.41 Enter Element



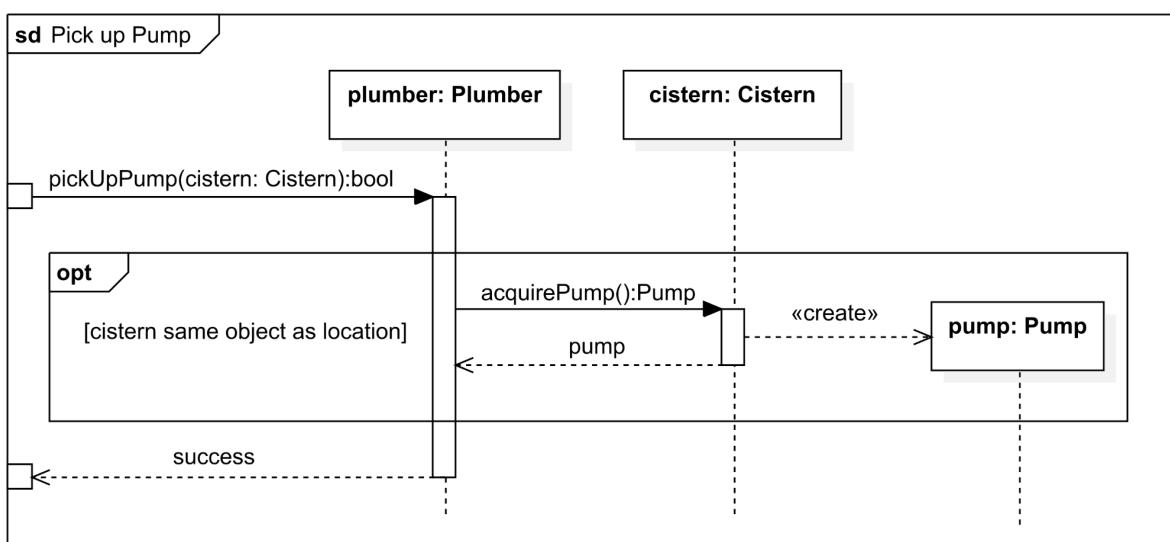
11.4.42 Enter Pipe



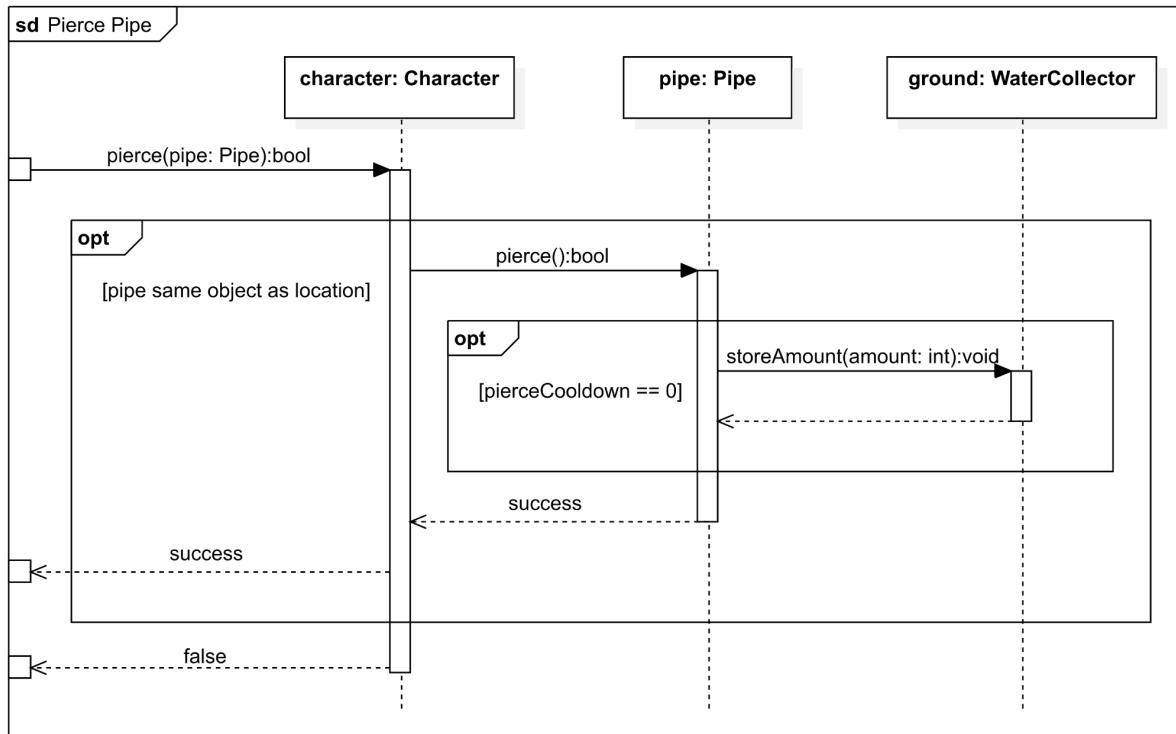
11.4.43 Move Character



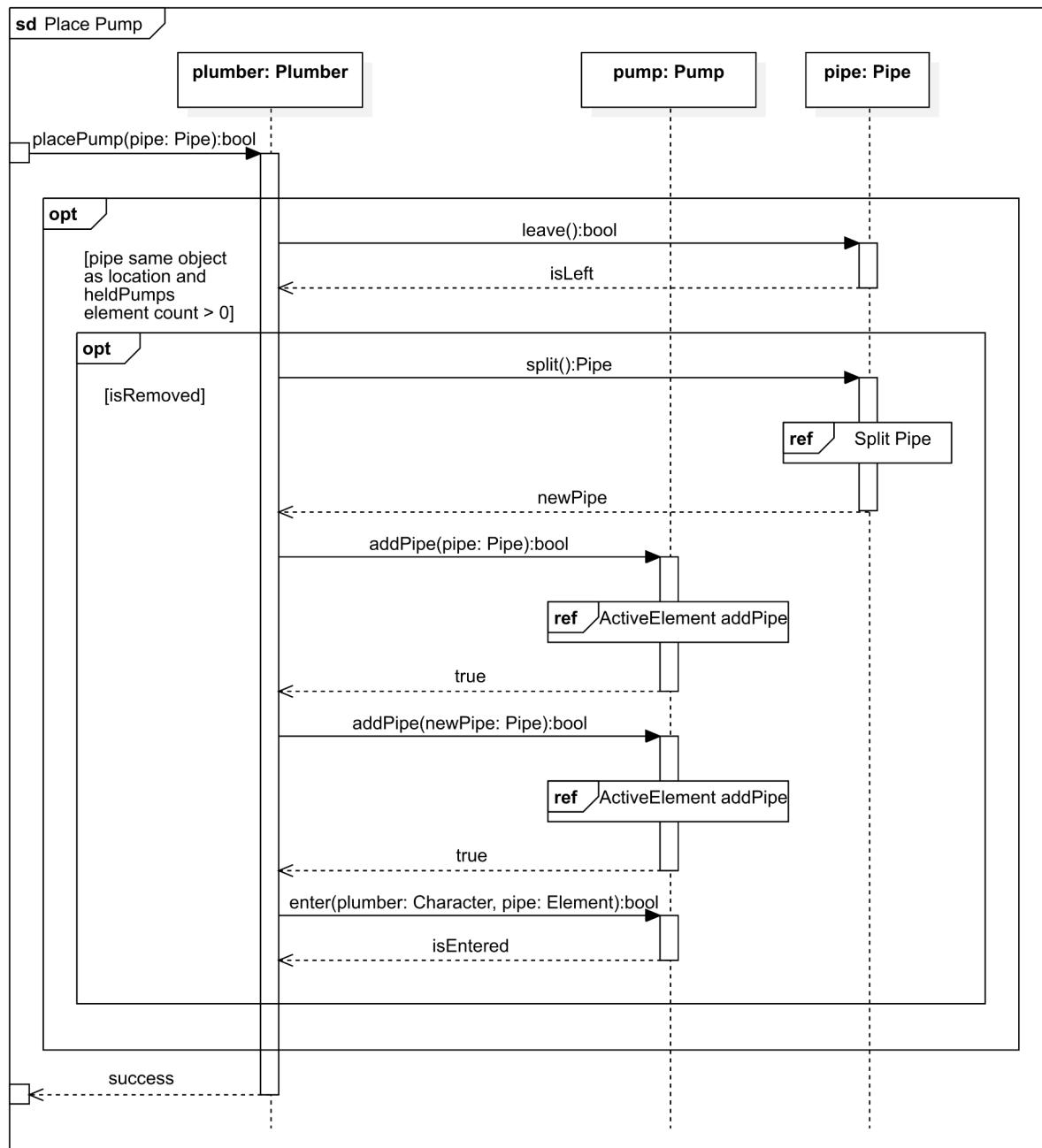
11.4.44 Pick up Pump



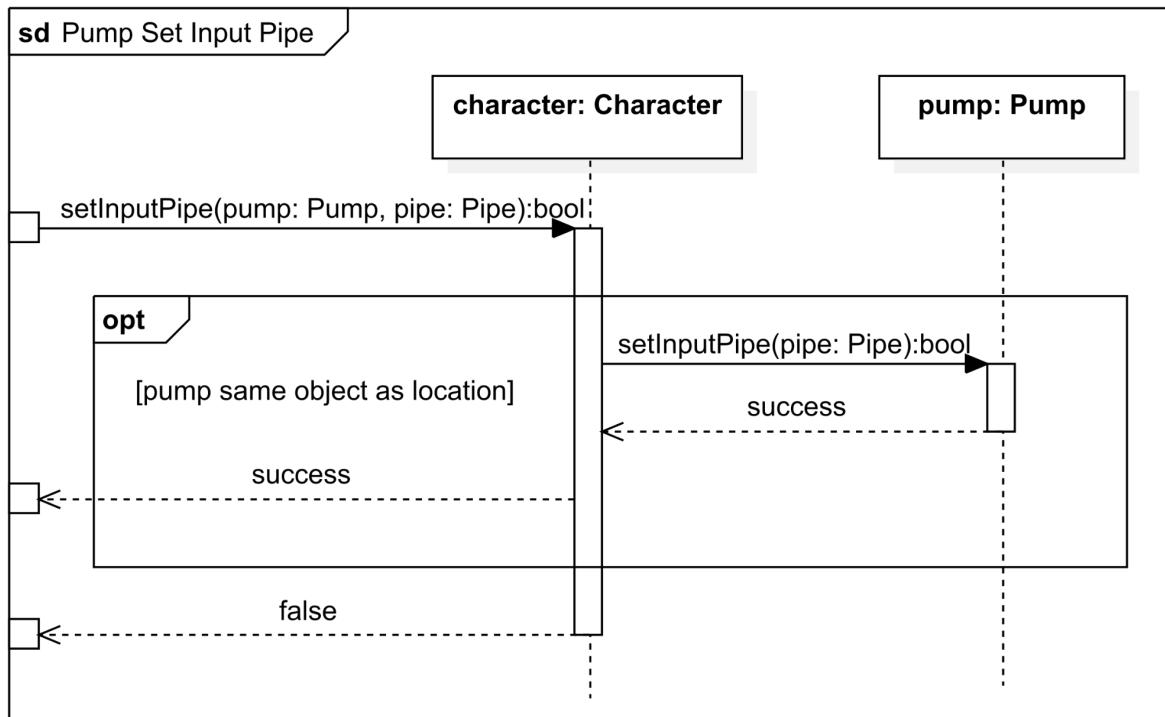
11.4.45 Pierce Pipe



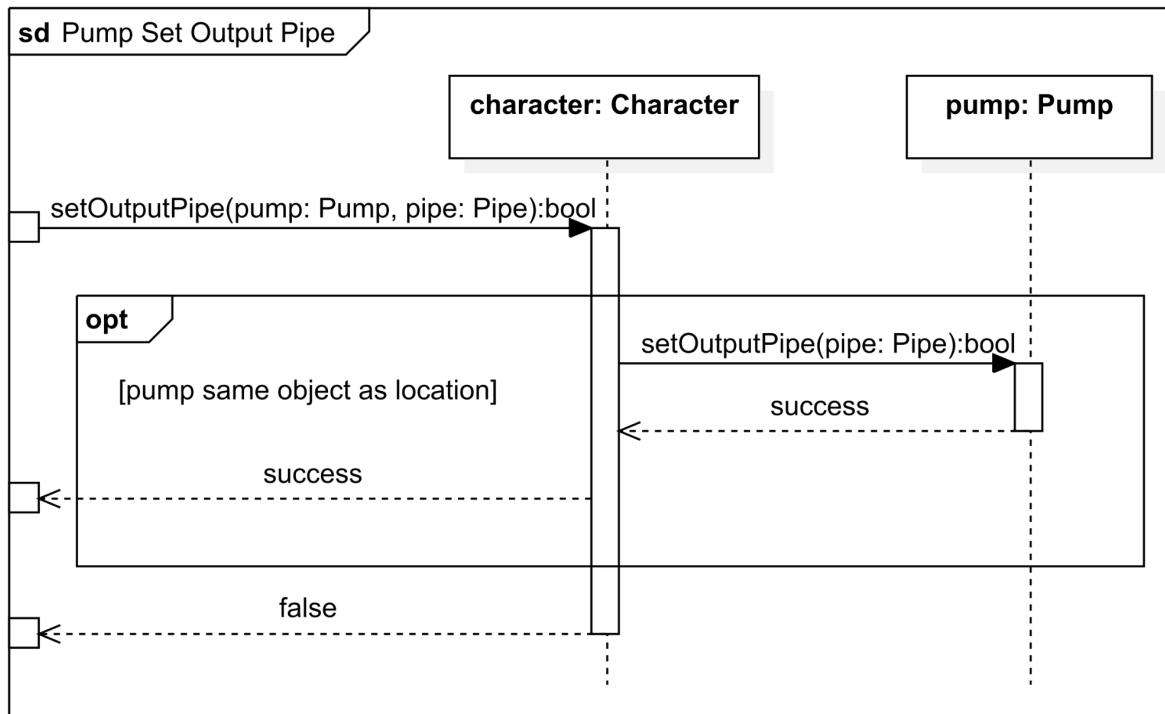
11.4.46 Place Pump



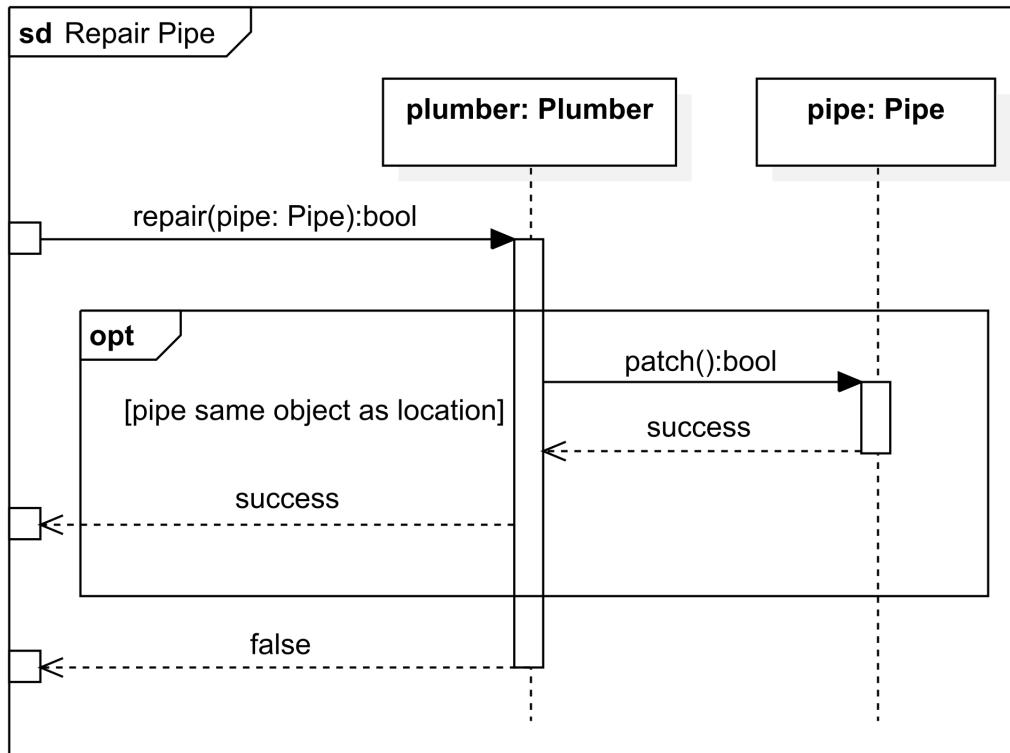
11.4.47 Pump Set Input Pipe



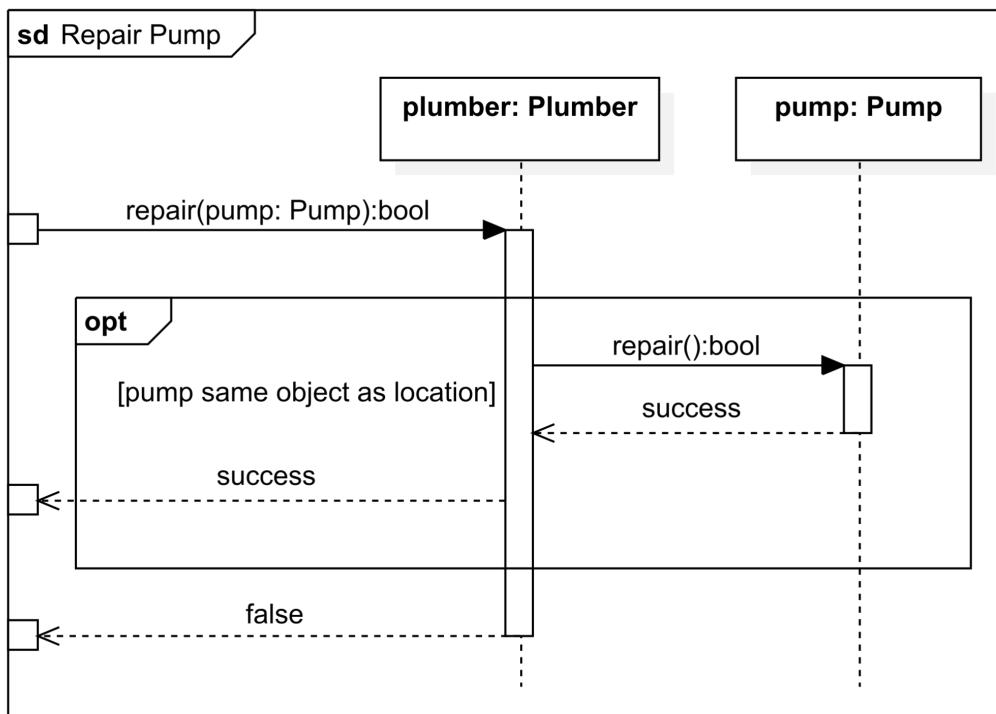
11.4.48 Pump Set Output Pipe



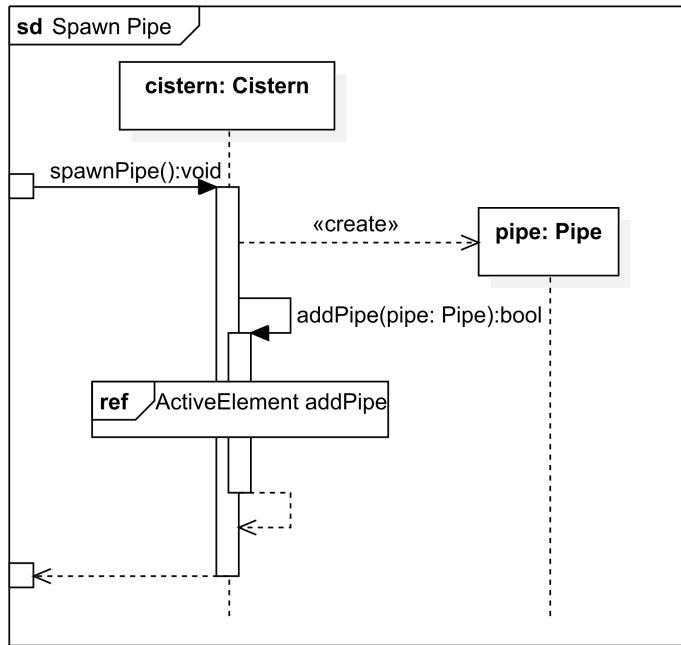
11.4.49 Repair Pipe



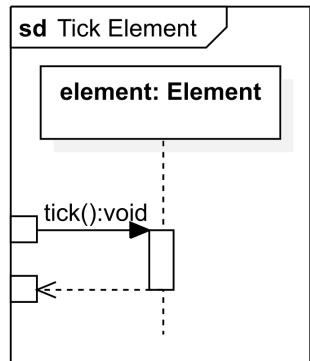
11.4.50 Repair Pump



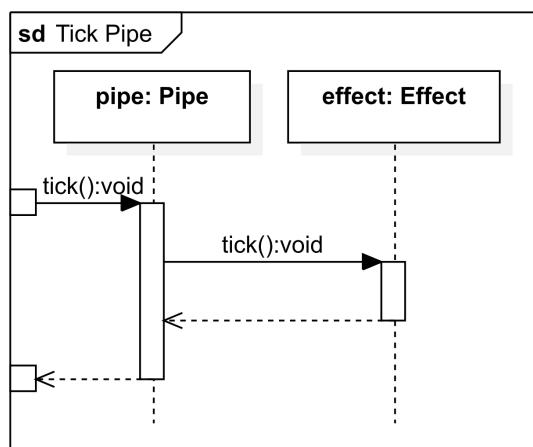
11.4.51 *Spawn Pipe*



11.4.52 *Tick Element*



11.4.53 *Tick Pipe*



11.5 Napló

Kezdet	Időtartam	Résznevők	Leírás
2023.05.17. 12:00	2 óra	Galgóczy Harsányi Szilczer Tóth	Értekezlet: Konzultáltunk az IL408-ban a laborvezetővel. Döntés: A következő értekezleten kiosztjuk az elvégzendő feladatokat.
2023.05.19. 16:00	1 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: Kiosztottuk az elvégzendő feladatokat. Döntés: Harsányi a 11.1, Kereszthury a 11.3, Szilczer és Tóth a 11.2 és 11.4 feladatokat készít el.
2023.05.20. 13:00	8 óra	Galgóczy Szilczer Tóth	Értekezlet: Megterveztük a View és Action osztályokat. Kiosztottuk az elkészítendő szekvenciadiagrammokat. Döntés: Szilczer a view osztályok és a modell változtatott szekvenciáit, Tóth a gombnyomás szekvenciáit készít el.
2023.05.21. 14:00	6 óra	Szilczer	Tevékenység: 11.2.2 kidolgozása
2023.05.21. 16:00	2 óra	Tóth	Tevékenység: 11.2.2 Action-ök elkészítése.
2023.05.21. 19:00	3 óra	Kereszthury	Tevékenység: 11.3 elkészítése
2023.05.21. 20:00	4 óra	Harsányi	Tevékenység: 11.1 illusztrációk elkészítése
2023.05.21. 21:00	5 óra	Szilczer	Tevékenység: 11.4.1-17 és 36-53 elkészítése
2023.05.21. 21:00	5 óra	Tóth	Tevékenység: 11.4.18-35 elkészítése.
2023.05.21. 02:00	1 óra	Szilczer	Tevékenység: 11.4 diagramjainak dokumentumba illesztése
2023.05.21. 02:00	1 óra	Tóth	Tevékenység: 11.1 szövegek megírása, 11.2.1 megírása.
2023.05.22. 10:45	1 óra	Galgóczy	Tevékenység: 11.1 és 11.2 javítása
2023.05.22. 12:00	1 óra	Galgóczy Harsányi Kereszthury Szilczer Tóth	Értekezlet: A dokumentum közös átnézése. Döntés: A dokumentum beadásra készen áll.

13. Grafikus változat beadása

13.1 Fordítási és futtatási útmutató

13.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
src/macaroni/commandHandler/commands/ AttachCommand .java	1697	2023.04.27. 0:57	A csatlakoztató parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ BananaCommand .java	1715	2023.04.28. 1:01	A csúszóssá tevő parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ BreakCommand .java	1095	2023.04.27. 23:09	A tönkretevő parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ Command .java	1745	2023.04.26. 21:38	A parancsok interfésze.
src/macaroni/commandHandler/commands/ CreateCommand .java	7780	2023.04.26. 21:38	Az objektumok létrehozásának parancsát megvalósító osztály.
src/macaroni/commandHandler/commands/ DetachCommand .java	2120	2023.04.26. 0:22	A lecsatlakoztató parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ GetPumpCommand .java	1352	2023.04.28. 1:01	A pumpa felvétel parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ HelpCommand .java	1872	2023.04.26. 21:38	A segítség parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ ListCommand .java	2388	2023.04.26. 22:11	A listázó parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ LoadCommand .java	1550	2023.04.27. 23:09	A pályabetöltő parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ MoveCommand .java	1157	2023.04.28. 1:01	A játékos mozgás parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ PierceCommand .java	1151	2023.04.28. 1:01	A kilyukasztó parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ PlacePumpCommand .java	1342	2023.04.28. 1:01	A pumpa lehelyező parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ RepairCommand .java	1893	2023.04.28. 1:01	A javítás parancsot megvalósító osztály.

src/macaroni/commandHandler/commands/ SaveCommand.java	1817	2023.04.27 23:09	A fájlba mentés parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ SetCommand.java	1478	2023.04.28. 1:01	A pumpa be- vagy kimenet állító parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ SpawnPipeCommand.java	1448	2023.04.28. 1:01	A ciszternára új csövet helyező parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ TechnokolCommand.java	1519	2023.04.28. 1:01	A ragadóssá tevő parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ TickCommand.java	1792	2023.04.27 23:09	Az idő műlás parancsot megvalósító osztály.
src/macaroni/commandHandler/commands/ ViewCommand.java	1754	2023.04.27. 0:15	A játékbeli objektumok állapotának megtekintése parancsot megvalósító osztály.
src/macaroni/commandHandler/ CommandInterpreter.java	3663	2023.04.26 21:38	A felhasználó által kiadott parancsokat feldolgozó osztály..

Fájl neve	Méret	Keletkezés ideje	Tartalom
src/macaroni/model/element/ ActiveElement.java	1428	2023.04.06. 23:10	Az Aktív elemek osztálya.
src/macaroni/model/effect/ BananaEffect.java	1624	2023.04.23. 18:17	A Csúszósság effektjét megvalósító osztály.
src/macaroni/model/character/ Character.java	3001	2023.04.06. 23:10	A Karakterek osztálya.
src/macaroni/model/element/ Cistern.java	1961	2023.04.06. 23:10	A Ciszternák osztálya.
src/macaroni/model/effect/ Effect.java	1078	2023.04.23. 18:17	Az effektek interfésze.
src/macaroni/model/element/ Element.java	1029	2023.04.06. 23:10	Az Elemek osztálya.
src/macaroni/model/effect/ NoEffect.java	1043	2023.04.23. 18:17	A semmilyen effektet megvalósító osztály.
src/macaroni/model/element/ Pipe.java	7255	2023.04.06. 23:10	A Csövek osztálya.
src/macaroni/model/character/ Plumber.java	3779	2023.04.06. 23:10	A Szerelők osztálya.
src/macaroni/model/element/ Pump.java	3865	2023.04.06. 23:10	A Pumpák osztálya.
src/macaroni/model/character/ Saboteur.java	858	2023.04.06. 23:10	A Szabotörök osztálya.
src/macaroni/model/element/ Spring.java	936	2023.04.06. 23:10	A Források osztálya.

src/macaroni/model/effect/ TechnokoleEffect.java	1867	2023.04.23. 18:17	A Ragadósság effektjét megvalósító osztály.
src/macaroni/model/misc/ WaterCollector.java	450	2023.04.06. 23:10	A Víztartályok osztálya.

Fájl neve	Méret	Keletkezés ideje	Tartalom
src/macaroni/ Main.java	263	2023.04.26 21:38	A proto kódja innen indul. Tartalmazza a main függvényt.
src/macaroni/utils/ ModelObjectFactory.java	10238	2023.04.28. 16:11	Osztály amely modell osztályokból hoz létre példányokat, és nyilvántartja azokat név szerint.
src/macaroni/utils/ ModelObjectSerializer.java	10907	2023.05.02. 14:43	Osztály, ami a modell osztályok és szöveges reprezentációjuk közti konverziót végzi.
src/macaroni/utils/ Randomdom.java	5629	2023.05.10. 1:31	Véletlen szám generáló osztály, ami segítségével lehet determinisztikus értékeket is beállítani a véletlen helyett.

Fájl neve	Méret	Keletkezés ideje	Tartalom
src/pasta/asserts/ Assertion.java	517	2023.05.01. 3:37	Funckionális interface tesztelés közbeni ellenőrzés elvégzéséhez.
src/pasta/asserts/ Assertions.java	1719	2023.05.10. 23:00	Segédosztály tesztelési ellenőrzéshez.
src/pasta/asserts/ AttributeAssertion.java	2975	2023.05.10. 23:00	Tagváltozói ellenőrzést leíró osztály.
src/pasta/ Main.java	11738	2023.04.30. 1:04	A tesztelést támogató program kódja innen indul. Tartalmazza a main függvényt.
src/pasta/asserts/ ObjectAssertion.java	1887	2023.05.11 1:56	Objektum létezésének ellenőrzését leíró osztály.
src/pasta/asserts/ Result.java	838	2023.04.30. 17:45	Ellenőrzés végeredményét leíró osztály.
src/pasta/ Test.java	10376	2023.04.30. 15:21	Teszteset kereső osztály.
tests/all.config.pasta	170	2023.05.11. 2:37	Az összes tesztet közösen elindító konfigfájl.

Fájl neve	Méret	Keletkezés ideje	Tartalom
tests/Basic Actions/.config.pasta	361	2023.05.10. 22:33	Az 1-10., 22-24. teszteket közösen elindító konfigfájl.
tests/Basic Actions/1. Repair Pipe.pasta	71	2023.05.03. 2:46	Az 1. teszt tesztfájlja.

tests/Basic Actions/maps/ 1.map.pasta	342	2023.05.03. 2:46	Az 1. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/2. Repair Pump.pasta	70	2023.05.03. 2:46	A 2. teszt tesztfájlja.
tests/Basic Actions/maps/ 2.map.pasta	64	2023.05.03. 2:46	A 2. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/3. Pierce Pipe.pasta	70	2023.05.03. 2:46	A 3. teszt tesztfájlja.
tests/Basic Actions/maps/ 3.map.pasta	367	2023.05.03. 2:46	A 3. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/4. Pierce unpierceable Pipe.pasta	71	2023.05.03. 2:46	A 4. teszt tesztfájlja.
tests/Basic Actions/maps/ 4.map.pasta	356	2023.05.03. 2:46	A 4. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/5. Technokol Pipe.pasta	77	2023.05.10. 22:33	Az 5. teszt tesztfájlja.
tests/Basic Actions/maps/ 5.map.pasta	367	2023.05.10. 22:33	Az 5. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/6. Banana Pipe	72	2023.05.10. 22:33	A 6. teszt tesztfájlja.
tests/Basic Actions/maps/ 6.map.pasta	271	2023.05.10. 22:33	A 6. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/7. Set Pump input.pasta	82	2023.05.10. 22:33	A 7. teszt tesztfájlja.
tests/Basic Actions/maps/ 7.map.pasta	341	2023.05.10. 22:33	A 7. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/8. Set Pump output.pasta	84	2023.05.10. 22:33	A 8. teszt tesztfájlja.
tests/Basic Actions/maps/ 8.map.pasta	341	2023.05.10. 22:33	A 8. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/9. Unable to Set Pump input.pasta	108	2023.05.10. 22:33	A 9. teszt tesztfájlja.
tests/Basic Actions/maps/ 9.map.pasta	319	2023.05.10. 22:33	A 9. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/10. Unable to Set pump output.pasta	110	2023.05.10. 22:33	A 10. teszt tesztfájlja.
tests/Basic Actions/maps/ 10.map.past a	318	2023.05.10. 22:33	A 10. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/22. Acquire Pump.pasta	87	2023.05.03. 4:41	A 22. teszt tesztfájlja.
tests/Basic Actions/maps/ 22.map.past a	134	2023.05.03. 4:41	A 22. teszthez tartozó pályaleíró fájl.

tests/Basic Actions/23. Place Pump.pasta	309	2023.05.03. 4:41	A 23. teszt tesztfájlja.
tests/Basic Actions/maps/ 23.map.pasta	380	2023.05.03. 4:41	A 23. teszthez tartozó pályaleíró fájl.
tests/Basic Actions/24. Unable to Place Pump.pasta	166	2023.05.03. 4:41	A 24. teszt tesztfájlja.
tests/Basic Actions/maps/ 24.map.pasta	362	2023.05.03. 4:41	A 24. teszthez tartozó pályaleíró fájl.

Fájl neve	Méret	Keletkezés ideje	Tartalom
tests/Attach-Detach Actions/.config.pasta	508	2023.05.10. 22:33	A 11-21. teszteket közösen elindító konfigfájl.
tests/Attach-Detach Actions/ 11. Detach both ends of unoccupied Pipe from Pump.pasta	315	2023.05.02. 22:00	A 11. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/ 11.map.pasta	402	2023.05.02. 22:00	A 11. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 12. Detach Pipe from Cistern.pasta	279	2023.05.03. 1:52	A 12. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/ 12.map.pasta	415	2023.05.03. 1:52	A 12. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 13. Detach Pipe from Spring.pasta	275	2023.05.03. 1:52	A 13. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/ 13.map.pasta	342	2023.05.03. 1:52	A 13. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 14. Detach occupied Pipe.pasta	327	2023.05.03. 1:52	A 14. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/ 14.map.pasta	379	2023.05.03. 1:52	A 14. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 15. Detach Pipe end while holding another Pipe end.pasta	362	2023.05.03. 1:52	A 15. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/ 15.map.pasta	408	2023.05.03. 1:52	A 15. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/ 16. Detach both ends of new Pipe.pasta	370	2023.05.03. 3:17	A 16. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/ 16.map.pasta	403	2023.05.03. 3:17	A 16. teszthez tartozó pályaleíró fájl.

tests/Attach-Detach Actions/17. Attach both ends of Pipe.pasta	386	2023.05.03. 4:10	A 17. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/17.map.pasta	321	2023.05.03. 4:10	A 17. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/18. Attach both ends of Pipe to same Pump.pasta	238	2023.05.03. 4:10	A 18. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/18.map.pasta	186	2023.05.03. 4:10	A 18. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/19. Attach both ends of Pipe to same Cistern.pasta	308	2023.05.03. 4:10	A 19. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/19.map.pasta	201	2023.05.03. 4:10	A 19. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/20. Attach both ends of Pipe to same Spring.pasta	246	2023.05.03. 4:10	A 20. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/20.map.pasta	196	2023.05.03. 4:10	A 20. teszthez tartozó pályaleíró fájl.
tests/Attach-Detach Actions/21. Attach Pipe to Pump with no free ports.pasta	343	2023.05.03. 4:10	A 21. teszt tesztfájlja.
tests/Attach-Detach Actions/maps/21.map.pasta	197	2023.05.03. 4:10	A 21. teszthez tartozó pályaleíró fájl.

Fájl neve	Méret	Keletkezés ideje	Tartalom
tests/Character Move/.config.pasta	262	2023.05.10. 22:33	A 25-32. teszteket közösen elindító konfigfájl.
tests/Character Move/ 25. Move to unoccupied Pipe.pasta	96	2023.05.03. 3:13	A 25. teszt tesztfájlja.
tests/Character Move/maps/ 25.map.pasta	324	2023.05.03. 3:13	A 25. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 26. Move to occupied Pipe.pasta	99	2023.05.03. 3:13	A 26. teszt tesztfájlja.
tests/Character Move/maps/ 26.map.pasta	364	2023.05.03. 3:13	A 26. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 27. Move to detached Pipe.pasta	97	2023.05.03. 3:13	A 27. teszt tesztfájlja.

tests/Character Move/maps/ 27.map.pasta	239	2023.05.03. 3:13	A 27. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 28. Move to Pump.pasta	99	2023.05.03. 3:13	A 28. teszt tesztfájlja.
tests/Character Move/maps/ 28.map.pasta	322	2023.05.03. 3:13	A 28. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 29. Move to Cistern.pasta	103	2023.05.03. 3:13	A 29. teszt tesztfájlja
tests/Character Move/maps/ 29.map.pasta	368	2023.05.03. 3:13	A 29. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 30. Move to Spring.pasta	101	2023.05.03. 3:13	A 30. teszt tesztfájlja.
tests/Character Move/maps/ 30.map.pasta	308	2023.05.03. 3:13	A 30. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 31. Move to banana Pipe.pasta	151	2023.05.03. 3:13	A 31. teszt tesztfájlja.
tests/Character Move/maps/ 31.map.pasta	324	2023.05.03. 3:13	A 31. teszthez tartozó pályaleíró fájl.
tests/Character Move/ 32. Move through technokoled Pipe.pasta	201	2023.05.03. 3:13	A 32. teszt tesztfájlja.
tests/Character Move/maps/ 32.map.pasta	322	2023.05.03. 3:13	A 32. teszthez tartozó pályaleíró fájl.

Fájl neve	Méret	Keletkezés ideje	Tartalom
tests/Controller Actions/.config.pasta	56	2023.05.10. 22:33	A 33., 34. teszteket közösen elindító konfigfájl.
tests/Controller Actions/ 33. Break Pump.pasta	61	2023.05.03. 4:51	A 33. teszt tesztfájlja.
tests/Controller Actions/maps/ 33.map.pasta	32	2023.05.03. 4:51	A 33. teszthez tartozó pályaleíró fájl.
tests/Controller Actions/ 34. Spawn Pipe.pasta	241	2023.05.03. 4:51	A 34. teszt tesztfájlja.
tests/Controller Actions/maps/ 34.map.pasta	102	2023.05.03. 4:51	A 34. teszthez tartozó pályaleíró fájl.

Fájl neve	Méret	Keletkezés ideje	Tartalom
tests/Element Tick/.config.pasta	289	2023.05.10. 22:33	A 35-42. teszteket közösen elindító konfigfájl.
tests/Element Tick/ 35. Waterflow from Spring.pasta	89	2023.05.03. 1:01	A 35. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/ 35.map.pasta	349	2023.05.03. 1:01	A 35. teszthez tartozó pályaleíró fájl.

tests/Element Tick/36. Waterflow through Pump.pasta	286	2023.05.03. 1:43	A 36. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/36.map.pasta	610	2023.05.03. 1:43	A 36. teszthez tartozó pályaleíró fájl.
tests/Element Tick/37. Waterflow into Cistern.pasta	148	2023.05.03. 2:02	A 37. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/37.map.pasta	376	2023.05.03. 2:02	A 37. teszthez tartozó pályaleíró fájl.
tests/Element Tick/38. Waterflow into full Pipe.pasta	129	2023.05.03. 2:07	A 38. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/38.map.pasta	349	2023.05.03. 2:07	A 38. teszthez tartozó pályaleíró fájl.
tests/Element Tick/39. Waterflow into detached Pipe.pasta	141	2023.05.03. 2:12	A 39. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/39.map.pasta	271	2023.05.03. 2:12	A 39. teszthez tartozó pályaleíró fájl.
tests/Element Tick/40. Waterflow into pierced Pipe.pasta	141	2023.05.03. 2:16	A 40. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/40.map.pasta	348	2023.05.03. 2:16	A 40. teszthez tartozó pályaleíró fájl.
tests/Element Tick/41. Tick Banana Pipe.pasta	111	2023.05.03. 2:37	A 41. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/41.map.pasta	116	2023.05.03. 2:37	A 41. teszthez tartozó pályaleíró fájl.
tests/Element Tick/42. Tick Technokoled Pipe.pasta	117	2023.05.03. 2:46	A 42. teszt parancsait tartalmazó fájl.
tests/Element Tick/maps/42.map.pasta	116	2023.05.03. 2:46	A 42. teszthez tartozó pályaleíró fájl.

Fájl neve	Méret	Keletkezés ideje	Tartalom
src/macaroni/actions/Action.java	316	2022.05.22. 17:03	Az akciók absztrakt ōosztálya.
src/macaroni/actions/DetachPipeAction.java	1 078	2022.05.22. 17:03	A cső lecsatlakoztató akció osztálya.
src/macaroni/actions/MoveAction.java	1 425	2022.05.22. 17:03	A karakter mozgató akció osztálya.
src/macaroni/actions/SetPumpInputAction.java	824	2022.05.22. 17:03	A pumpa bemenet állító akció osztálya.
src/macaroni/actions/SetPumpOutputAction.java	828	2022.05.22. 17:03	A pumpa kimenet állító akció osztálya.

Fájl neve	Méret	Keletkezés ideje	Tartalom
src/macaroni/math/Vector2D.java	2 373	2022.05.29. 23:59	2D vektorokat reprezentáló osztály.

Fájl neve	Méret	Keletkezés ideje	Tartalom
src/macaroni/app/App.java	1 183	2022.05.27. 13:35	Elindítja, frissíti és kirajzolja az ablakot (Window).
src/macaroni/app/AssetManager.java	2 563	2022.05.29. 20:17	A játékban használt különböző ábrák vagy betűtípusok betöltésére szolgál.
src/macaroni/app/Controller.java	2 099	2022.05.27. 14:52	
src/macaroni/app/GameColors.java	437	2022.05.28. 13:35	A játékban használt színek.
src/macaroni/app/Window.java	1 487	2022.05.27. 13:35	Az ablak, ahol a játék fut.
src/macaroni/app/gameView/ActionButton.java	2 330	2022.05.30. 12:55	Az adott karakter által végrehajtható egyik akciót végrehajtó gomb.
src/macaroni/app/gameView/Background.java	605	2022.05.30. 20:58	A háttérbeli fordított L osztálya.
src/macaroni/app/gameView/CharacterDisplayPanel.java	2 290	2022.05.30. 12:55	A profilkép osztálya.
src/macaroni/app/gameView/Dragger.java	1 486	2022.05.30. 22:50	A pálya képen történő mozgatását végző osztály.
src/macaroni/app/gameView/GameMenu.java	16 880	2022.05.27. 16:32	A játék ablak különböző elemeit összefogó osztály.
src/macaroni/app/gameView/GameOverPanel.java	1 025	2022.05.31. 3:21	A játék befejezésekor megjelenő panel.
src/macaroni/app/gameView/GamePanel.java	5 970	2022.05.30. 12:55	A játék ablak bal oldali elemeit összefogó osztály.
src/macaroni/app/gameView/HandView.java	3 446	2022.05.31. 2:45	A játékos kezében lévő pumpák és csövek számának megjelenítéséért felelős.
src/macaroni/app/gameView/MapPanel.java	5 308	2022.05.27. 16:32	A pálya megjelenítéséhez használt panel.
src/macaroni/app/gameView/NotificationLabel.java	1 048	2022.05.30. 12:55	A bal alsó jelző ablak osztálya.
src/macaroni/app/gameView/ScorePanel.java	2 268	2022.05.30. 12:55	A pontszámot kijelző osztály.
src/macaroni/app/gameView/ViewRepository.java	969	2022.05.31. 2:57	A játék ablak kirajzolására használt objektumokat tárolja.

src/macaroni/app/logic/Game.java	10 387	2022.05.27. 16:32	A program játék részének központi futtatása.
src/macaroni/app/logic/GameInfo.java	370	2022.05.27. 16:32	A játék központi információit tartalmazó rekord.
src/macaroni/app/menuView/MainMenu.java	6 072	2022.05.28. 13:35	A menü ablak különböző elemeit összefogó osztály.
src/macaroni/app/menuView/MenuLabel.java	992	2022.05.28. 13:35	A menü ablak címkeinek osztálya.
src/macaroni/app/menuView/MenuList.java	1 534	2022.05.28. 13:35	A menü ablakon lévő listázó elem osztálya.
src/macaroni/app/menuView/MenuListCellRenderer.java	1 839	2022.05.28. 13:35	A menü ablakon lévő listázó elem kirajzoló osztálya.
src/macaroni/app/menuView/MenuPanel.java	2 323	2022.05.28. 13:35	A menü ablak paneljainak osztálya.
src/macaroni/app/menuView/MenuStartButton.java	2 825	2022.05.28. 13:35	A start gomb osztálya.
src/macaroni/app/menuView/MenuTeamButton.java	1 920	2022.05.28. 13:35	A csapat kezelő új tag felvevő gombjának osztálya.
src/macaroni/app/menuView/TeamPanel.java	1 785	2022.05.28. 13:35	A csapat kezelő osztálya.
src/macaroni/app/menuView/TeamTextBox.java	2 537	2022.05.28. 13:35	A csapat kezelő új tag felvevő textbox-jának osztálya.

Fájl neve	Méret	Keletkezés ideje	Tartalom
src/macaroni/views/BananaEffectView.java	1 205	2022.05.22. 17:03	A csúszós effektus megjelenítéséért felelős.
src/macaroni/views/CharacterView.java	665	2022.05.22. 17:03	A karakter megjelenítéséért felelős.
src/macaroni/views/CisternView.java	1 195	2022.05.22. 17:03	A ciszterna megjelenítéséért felelős.
src/macaroni/views/PipeView.java	3 165	2022.05.22. 17:03	A cső megjelenítéséért felelős.
src/macaroni/views/PlumberView.java	833	2022.05.22. 17:03	A szerelő megjelenítéséért felelős.
src/macaroni/views/Position.java	423	2022.05.22. 17:03	Egy adott elem képernyőn elfoglalt helyét jelző osztály.

src/macaroni/views/PumpView.java	4 538	2022.05.22. 17:03	A pumpa megjelenítéséért felelős.
src/macaroni/views/SaboteurView.java	810	2022.05.22. 17:03	A szabotör megjelenítéséért felelős.
src/macaroni/views/SpringView.java	1 184	2022.05.22. 17:03	A forrás megjelenítéséért felelős.
src/macaroni/views/TechnokolEffectView.java	1 195	2022.05.22. 17:03	A ragadós effektus megjelenítéséért felelős.
src/macaroni/views/View.java	529	2022.05.22. 17:03	A megjelenítésért felelős osztályok abstract őse.

Fájl neve	Méret	Keletkezés ideje	Tartalom
assets/fonts/chakraPatch/ChakraPatch-Bold.ttf	71 436	2022.05.28. 13:35	A Játékban használt betűtípus.
assets/fonts/chakraPatch/ChakraPatch-BoldItalic.ttf	73 772	2022.05.28. 13:35	
assets/fonts/chakraPatch/ChakraPatch-Italic.ttf	73 808	2022.05.28. 13:35	
assets/fonts/chakraPatch/ChakraPatch-Light.ttf	71 912	2022.05.28. 13:35	
assets/fonts/chakraPatch/ChakraPatch-LightItalic.ttf	74 192	2022.05.28. 13:35	
assets/fonts/chakraPatch/ChakraPatch-Medium.ttf	71 608	2022.05.28. 13:35	
assets/fonts/chakraPatch/ChakraPatch-MediumItalic.ttf	73 972	2022.05.28. 13:35	
assets/fonts/chakraPatch/ChakraPatch-Regular.ttf	71 556	2022.05.28. 13:35	
assets/fonts/chakraPatch/ChakraPatch-SemiBold.ttf	71 600	2022.05.28. 13:35	
assets/fonts/chakraPatch/ChakraPatch-SemiBoldItalic.ttf	73 980	2022.05.28. 13:35	
assets/fonts/chakraPatch/OFL.txt	4 489	2022.05.28. 13:35	A Játékban használt betűtípus licensze.

Fájl neve	Méret	Keletkezés ideje	Tartalom
assets/images/map/Banana.png	6 761	2022.05.29. 21:11	A csúszós effektus ábrája.
assets/images/map/Broken.png	6 807	2022.05.29. 21:11	A töröttséget mutató ábra.
assets/images/map/Cistern.png	17 649	2022.05.28. 13:35	A ciszterna ábrája.
assets/images/map/EffectShadow.png	1 905	2022.05.29. 21:11	Az effektek jelenlétét jelző ábra. Célja, hogy

			az effekt és a cső ne mosódjon össze.
assets/images/map/Plumber.png	10 575	2022.05.28. 13:35	A szerelő ábrája.
assets/images/map/PumpEmpty.png	7 438	2022.05.29. 21:11	Az üres pumpa ábrája.
assets/images/map/PumpFull.png	7 113	2022.05.29. 21:11	A teli pumpa ábrája.
assets/images/map/Saboteur.png	10 276	2022.05.28. 13:35	A szabotőr ábrája.
assets/images/map/Spring.png	18 926	2022.05.29. 21:11	A forrás ábrája.
assets/images/map/Technokol.png	6 038	2022.05.29. 21:11	A ragadós effektus ábrája.
assets/images/map/triangle.png	2 664	2022.05.29. 21:11	A Pumpán a bemenetet vagy kimenetet jelző kis ábra.
assets/images/menu/Background.png	62 388	2022.05.30. 20:58	A játék hátterében lévő fordított L.
assets/images/menu/MenuCistern.png	17 649	2022.05.28. 16:00	A főmenün megjelenő ciszterna.
assets/images/menu/MenuSaboteur.png	22 441	2022.05.28. 16:00	A főmenün megjelenő szerelő.
assets/images/menu/MenuPlumber.png	21 027	2022.05.28. 16:00	A főmenün megjelenő szabotőr.
assets/images/menu/MiniPipe.png	5 536	2022.05.30. 21:18	A csövek kis ikonja.
assets/images/menu/MiniPump.png	4 202	2022.05.30. 21:18	A pumpák kis ikonja.
assets/images/menu/PlumberProfile.png	20 422	2022.05.30. 21:18	A szerelő profilképe.
assets/images/menu/PlumberWin.png	139 925	2022.05.31. 3:04	A szerelők győzelmét kijelző ábra.
assets/images/menu/SabouterProfile.png	18 042	2022.05.30. 21:18	A szabotőr profilképe.
assets/images/menu/SabouterWin.png	149 612	2022.05.31. 3:04	A szabotörök győzelmét kijelző ábra.

Fájl neve	Méret	Keletkezés ideje	Tartalom
assets/maps/K Épület.map	2 004	2023.05.29. 23:12	A játékban használható térképek.
assets/maps/Schönherz.map	4 945	2023.05.29. 23:12	
assets/maps/Tüske.map	1 283	2023.05.29. 23:12	

13.1.2 Fordítás és telepítés

A fordítás elvégezhető a kari felhő gépén, illetve saját gépen is, amennyiben van telepítve a számítógépre legalább 18-as verziójú Java JDK.

Fordítás lépései:

- A forrásfájlokat tartalmazó **ZIP fájlt kicsomagolni** tetszőleges helyre.
- **Parancssorban elnavigálni** arra a mappába, amiben az “src” mappa található. (cd használatával)
- **Futtatni az alábbi kódot**, ami egyszerre lefordítja a prototípus és a tesztelő programot is:
javac -cp src src/macaroni/Main.java src/pasta/Main.java -d out

13.1.3 Futtatás

A futtatás végezhető a kari felhő gépén, illetve saját gépen is, amennyiben van telepítve a számítógépre legalább 18-as verziójú Java JDK.

Futtatás lépései:

- **Fordítás elvégzése**
- **Parancssorban elnavigálni** abba a mappába, amiben az “src” mappa található. (cd használatával)
- **Futtatni az alábbi kódot**, ami elindítja a prototípus programot:
java -cp out macaroni.Main

13.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Galgóczy Gábor	QS2WJR	20
Harsányi Levente	FD7NQC	20
Kereszthury Péter	PUKAU0	20
Szilczer Ármin	THGPZB	20
Tóth Gábor Tamás	HAFLCT	20

13.3 Napló

Kezdet	Időtartam	Résznevők	Leírás
2023.05.24. 12:00	1,5 óra	Galgóczy Harsányi Kereszthury Tóth Szilczer	Értekezlet: Konzultáltunk az IL408-ban a laborvezetővel. Döntés: Galgóczy elkészíti a program alapját. Harsányi felelős a dokumentumért és az art-ért. Kereszthury elkészíti a főmenüt. Szilczer elkészíti a játéklogikát. Tóth elkészíti a játékmenüt.
2023.05.27. 13:00	3 óra	Galgóczy	A program alapjának felépítése
2023.05.27. 18:00	1 óra	Galgóczy	A program alapjának felépítése
2023.05.27. 19:00	1.5 óra	Harsányi	Assetek kirendelése
2023.05.27. 22:00	5 óra	Kereszthury	Főmenü készítése
2023.05.27. 11:00	2 óra	Kereszthury	Főmenü készítése
2023.05.28. 00:00	1 óra	Galgóczy	Főmenü felhasználóbarátabbá tétele
2023.05.28. 15:00	6 óra	Tóth	Action és View osztályok megírása, játékvezérlési logika elkezdése.
2023.05.29. 18:00	5 óra	Szilczer	View osztályok rajzolásának elkészítése
2023.05.30. 22:00	2 óra	Kereszthury	Játéknézet menüpontjainak készítése
2023.05.30. 09:45	3 óra	Kereszthury	Játéknézet menüpontjainak készítése
2023.05.30. 20:30	10 óra	Szilczer	Játéknézetek és pályák készítése
2023.05.30 21:00	9 óra	Tóth	Játékvezérlési logika, műveletek végzésének implementálása.
2023.05.31. 9:00	2 óra	Harsányi	13.1.1 Fájllista
2023.05.31. 10:00	1 óra	Szilczer	Játéknézet hibajavításai
2023.05.31. 11:00	1 óra	Galgóczy Harsányi Kereszthury Tóth Szilczer	Értekezlet: A dokumentum átnézése. Döntés: A dokumentum beadásra kész.

14. Összefoglalás

55 – macaroni

Konzulens:
Simon Balázs

Csapattagok

Kereszthury Péter
Szilczer Ármin
Tóth Gábor Tamás
Galgóczy Gábor
Harsányi Levente

PUKAU0 kereszthurypeter@gmail.com
THGPZB arminszi1czer@gmail.com
HAFLCT gabor7d2@gmail.com
QS2WJR ggabor2002@gmail.com
FD7NQC kockalakoo@gmail.com

2023.06.02.

14. Összefoglalás

14.1 A projektre fordított összes munkaidő

Tag neve	Munkaidő (óra)
Galgóczy Gábor	107,5
Harsányi Levente	89,25
Kereszthury Péter	92,5
Szilczer Ármin	138,5
Tóth Gábor Tamás	131,25
Összesen	559

14.2 A feltöltött programok forrássorainak száma

Fázis	Kódsorok száma
Szkeleton	1515
Prototípus	2163
Grafikus változat	4359
Összesen	8037

14.3 Projekt összegzés

14.3.1 Mit tanultak a projektből konkrétan és általában?

A csapatmunka és az csapattagok összehangolt működésének megszervezése volt a legnagyobb kihívás, ezt ahogyan haladtunk előre, egyre jobban felismertük és javítottunk rajta.

Fontos, minden beadandóhoz/projektrészhez legyen szervező (“főnök”) aki koordinálja a csapattagokat, és a meeting-eket, feladatok kiosztását, mert anélkül könnyen szébomlik a csapat.

Minél hamarabb meg kell tervezni (akár közösen) az alapvető dolgokat és kiosztani a párhuzamosan végezhető feladatokat.

14.3.2 Mi volt a legnehezebb és a legkönnyebb?

A legkönnyebb ennek az összefoglalás dokumentumnak a megírása volt, illetve a kódolás volt a 2. legkönnyebb dolog, mivel a tervezet megléte esetén már nem valami sok munka a kódolás megírni hozzá. A legnehezebb a hosszú dokumentumok megszerkesztése volt. Ezenkívül a szerkesztőprogrammal néha elég sokat bajlódtunk (Google Docs).

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

Nagyrészt igen, viszont itt a 3. szekcióban ez az összefoglalás max 10 pontot kellene, hogy érjen, és a maradék pontszámot pedig egyenletesen elosztani a grafikus tervezés és megvalósítás között.

14.3.4 Ha nem, akkor hol okozott ez nehézséget?

Volt időnk minden feladatot megcsinálni, tehát a pontszámok nehézséget nem okoztak.

14.3.5 Milyen változtatási javaslatuk van?

A szkeleton és a prototípus fázist össze kéne vonni, sokkal több munka ment a szekvenciakiírásba és az adatbekérésre mint a modell létrehozására és ellenőrzésére. A szekvenciakiírásokat meg kéne szüntetni és a tesztelést rendes unit testing library-vel végezni, ez jobban tükrözne egy valós szoftverfejlesztési projektet. Kevesebb dolgot kéne a dokumentumba leírni, pl. osztályok leírása, tesztfájlok leírása, és több hangsúlyt kellene fektetni a fejlesztést támogató rendszerekre (verziókezelés, TODO board használata, feladatok kiosztása, szervezés)

felesleges dolgok:

- osztályok leírása
- szkeleton tesztelés (szekvencia diagramok kiírása) - lehetett volna valami könyvtárral ami automatikusan logolja az annotated függvények hívását
- prototípus tesztelés tesztesetek dokumentumba leírása

szkeleton/proto tesztelésnél és grafikus változat jegyzőkönyvénél is túl kevés dolgot kellett ellenőrizni

egyértelműen el kellett volna az elején mondani, hogy lehet javítani előző doksis a következőben, és hogy ez jár-e pontlevonással

a grafikus résznél a grafikus megjelenítéshez szükséges osztályokat kellett felvenni az osztálydiagramra és szekvenciadiagramokra, de nem volt egyértelműen leírva, hogy a "grafikus megjelenítés" alatt mit kell érteni, ténylegesen csak a megjelenést, vagy a műveletek végzését is (vagyis hogy milyen módon lehet a grafikus interfészen modellmódosító műveleteket végezni)

a játékvezérlési logika nem volt egyértelműen leírva, hogy kell-e vagy nem, illetve hova kell túl sok, és néha feleslegesnek érződő dokumentációt kellett írni, főleg amikor ez a kódból is kiolvasható

csak mint az előző éves szofttech-en, itt is sokszor arra ment ki a feladat, hogy ki tud jobban belelátni Goldinak a fejébe, mert nincs egyértelműen leírva, hogy mi a számára megfelelő design, és ő is gyakran ellentmond a saját szavainak

14.3.6 Milyen feladatot ajánlanának a projektre?

14.3.7 Egyéb kritika és javaslat