

1.

Sivatagi vízhálózat

Egyesített dokumentáció 36 – codex

Konzulens:
Kovács Boldizsár

Csapattagok

Andai Roland
Bodnár Márk
Gutási Ádám
Sziksza Levente
Vajda András

H3G1PQ
AAL25F
XW53QZ
AENWDT
M7LWCB

andairoland@gmail.com
mr.pc.mark@gmail.com
gutasiadm@gmail.com
szikilevi01@gmail.com
andrisvajda71@gmail.com

2023.06.02.

2. Követelmény, projekt, funkcionalitás

2.1 Bevezetés

2.1.1 Cél

A dokumentum célja a projekt bemutatása, továbbá a követelmények és a funkcionalitások specifikálása.

2.1.2 Szakterület

Játékszoftver

2.1.3 Definíciók, rövidítések

White Star UML - UML diagramok készítéséhez használt szoftver.

UML - Unified Modeling Language, általános célú modellezési nyelv.

HSZK - Hallgatói Számítógép Központ, a BME R épületében.

BME - Budapesti Műszaki és Gazdaságtudományi Egyetem.

JDK - Java Development Kit, a java alkalmazások fejlesztéséhez szükséges szoftver.

Java - Egy programozási nyelv, mely alkalmazásával valósítjuk meg a projektet.

Java Runtime - Java Runtime Environment(JRE), a java alkalmazások futtatásához szükséges szoftver.

GitHub - A programkód verziókezeléséhez használt szolgáltatás.

Google Drive - A dokumentáció verziókezeléséhez használt szolgáltatás.

Discord - Hang és üzenet alapú kommunikációs szolgáltatás, a csapat meetingek lebonyolításához használjuk.

Messenger - Üzenetküldő szolgáltatás, a kapcsolattartás egyik fő eszköze a csapattagok között.

2.1.4 Hivatkozások

www.iit.bme.hu/oktatas/tanszeki_targyak/BMEVIIIAB02 - Szoftver projekt laboratórium tárgy honlapja

www.iit.bme.hu/oktatas/tanszeki_targyak/BMEVIIIAB01 - Tavalyi szoftvertechnológia tárgy honlapja

www.google.com/drive/ - Google drive

github.com - GitHub

Előző évi szoftvertechnológia házi feladatok

2.1.5 Összefoglalás

A dokumentum további részében bemutatjuk a szoftver alapvető funkcióit, követelményeit, valamint felhasználókra vagy erőforrásokra vonatkozó korlátozásait.

2.2 Áttekintés

2.2.1 Általános áttekintés

A létrejövő szoftver a legmagasabb szinten három különböző architekturális rétegből áll:

- **Adatelérési réteg:** Ezen réteg felelős minden olyan funkcióért, mely adatok elérését, módosítását, vagy tárolását valósítja meg. Az ezen réteghez kialakítandó interfész lényege, hogy könnyed adatelérést és manipulálást biztosítson a programozónak.

- **Grafikus réteg:** Ezen réteg felelős a program megjelenéséért. A grafikus réteg tartalmazza a szoftver felhasználói felületét, beleértve a grafikus elemeket, illetve az ezen elemek megjelenítését biztosító funkciókat. A grafikus réteg megában foglalja továbbá a szoftver és a felhasználó közti működést is, aminek segítségével a felhasználó interakcióba léphet a szoftverrel.
- **Üzleti logikai réteg:** Ezen réteg tartalmazza a szoftver üzleti logikát. Ide tartoznak még azok a főbb funkciók és szabályok, amik meghatározzák, hogy a többi réteg egyes funkciói milyen események hatására és mikor hajtódhatnak végre. A logikai réteg tartalmazza továbbá az adatok struktúráját és az egymással történő viselkedésüket.
A szoftvernek nincs semmilyen hálózattal kapcsolatos elvárása, az nem szükséges a játékkal való játékhöz.

2.2.2 Funkciók

A drukmákor 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 forráshoz és a ciszternához tetszőleges véges számú cső csatlakozhat, viszont nem lehet csövet bekötni direkt a forrás és ciszterna közé. Forrásból forrásba vezető cső sem létezhet. Egy pumpa több (de a pumpára jellemző véges számú) csövet is összeköthet, és minden pumpán külön-külön állítható, hogy éppen melyik belekötött csőből melyik másik csőbe pumpáljon mind a szerelők és mind a szabotőrök által, azonban egyszerre csak egy bemenete és egy kimenete lehet. A többi rákötött cső eközben el van zárvva. A pumpák véletlenszerű körönként el tudnak romlani, ilyenkor megszűnik az adott pumpánál a vízáramlás. A pumpák mindegyike rendelkezik egy víztartályval, amit a víz átemelése közben használ átmeneti tárolóként. A pumpa nem tárol önmagában vizet. A pumpa csak akkor tud vizet pumpálni egy csőbe, ha a cső szabad kapacitása ezt lehetővé teszi.

A csőhálózat bővíthető, változtatható. A csövek kellően rugalmasak ahhoz, hogy a szerelők által az egyik végüket lecsatlakoztatva egy másik aktív elemhez elvihetők és ott felcsatlakoztathatók legyenek. A ciszternáknál folyamatosan készülnek az új csövek, amiket a szerelők kihúzhatnak a ciszternákból, amelyek egyik vége a ciszternához kapcsolódik, a másik azonban szabad. A szabad végű csövekből a csőbe betáplált víz a homokba folyik.

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 megfoltozása 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 ciszternáknál magukhoz tudnak venni új pumpát is, amit egy cső közepén tudnak elhelyezni. A csövet ehhez ketté kell vágni, és a két végét a pumpához kell csatlakoztatni. A szerelőknél egyszerre csak egy darab tárgy lehet. (legyen az akár cső vagy akár pumpa). Ezeket csak szabályosan helyezhetik el, tárgyakat eldobni nem lehet.

A hálózaton élnek a nomád szabotőrök is, akik a pumpákat tudják átállítani és a csöveket szokták kilyukasztani. A lyukas csövekből a víz a sivatagba elfolyik. Egy csövön egyszerre csak 1 lyuk lehet. A szabotőrök nem vihetnek magukkal tárgyakat.

Mivel a sivatag veszélyes hely, a szerelők és a szabotőrök csak a csőhálózaton haladhatnak. A pumpáknál, a forrásokon és a ciszternáknál kikerülhetik egymást, de a csöveken már nem tudnak elmenni egymás mellett, egy csövön egyszerre csak egy ember állhat.

A játék körökből áll, amelyet két csapat legalább 2-2 játékossal játszsa. A játékosok száma állítható a játék kezdetén. A szabotőrök dolga, hogy minél több víz folyjon el a lyukakon, a szerelők pedig azon dolgoznak, hogy minél több víz jusson a ciszternákba. Az a csapat nyer, amelyik a játék végére több vizet szerez. Állítható, hogy hány kör után ér véget a játék. Egy körben az egyik csapat minden játékosa mozoghat egyszer, a szerelés, csőlyukasztás vagy pumpa elhelyezés is mozgásnak számít, a másik csapatból azonban senki, ha vége az adott körnek akkor következik a másik csapat.

2.2.3 Felhasználók

A felhasználók száma legalább 4 fő a játék rendeltetésszerű használatához.

Az egyes felhasználók legyenek képesek kezelni az alapvető perifériákat (billentyűzet, egér).

2.2.4 Korlátozások

A szoftver játékprogram mivoltából fakadóan nem létszükségletű, hogy minden elérhető legyen.

Játék befejeztével ajánlott bezárni, leállítani a programot.

Egy játékmenet csak egyetlen gépen játszható, nem lehet hálózatban is használni.

2.2.5 Feltételezések, kapcsolatok

Szoftver projekt laboratórium tárgy honlapja: Az adott feladat követelményeinek és részleteinek a leírása.

Tavalyi szoftvertechnológia tárgy honlapja: Az előadáson kivetített diákok tartalma segített a use case-ek elkészítésében

Google drive: Fájlok megosztása valamint mindenki számára szerkeszthető fájlok használata.

GitHub: A programkód közös megírásához és megosztásához használt oldal.

Előző évi szoftvertechnológia házi feladatok: Néhány követelmény illetve use case átvitele az idei feladatba.

2.3 Követelmények

2.3.1 Funkcionális követelmények

A továbbiakban négy különböző betűvel jelöljük a projekt során előforduló funkcionális követelményeket. Ezek a következők:

- **F - Funkcionális követelmény**
- **E - Erőforrásokkal kapcsolatos követelmények**
- **A - Átadással kapcsolatos követelmények**
- **N - Nemfunkcionális követelmény**

A betűket egy kétjegyű szám követi, mely azonosítja a különböző kategóriákon belüli követelményeket.

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
F01	Csőrendszer szállítja a vizet.	Strukturális áttekintés	alapvető	Megrendelő	Simulate water flow	

F02	Egy forráshoz tetszőleges véges számú cső csatlakozhat.	Kellően nagy számú csövet csatlakoztatunk egy tetszőleges forráshoz.	alapvető	csapat	Place tube-end	
F03	Egy ciszternához tetszőleges véges számú cső csatlakoztatható.	Kellően nagy számú csövet csatlakoztatunk egy tetszőleges ciszternához	alapvető	megrendelő	Place tube-end	
F04	Nem lehet csövet bekötni, forrás és ciszterna közt.	Megpróbálunk bekötni, és ellentmondásra kell jutnunk.	alapvető	csapat	Place tube-end	
F05	A csövek elágazás mentesek	Strukturális áttekintés	alapvető	Megrendelő	View map	
F06	A csőrendszer csövekből, ciszternákból, pumpákból és forrásokból áll.	Strukturális áttekintés	alapvető	Megrendelő	View map	
F07	Egy pumpa több, a pumpára jellemző véges számú csövet is összeköthet	Ha a pumpa csőkapacitásán belüli számú csővel kötjük össze, akkor sikerülnie kell. Ha nagyobbal, akkor nem.	alapvető	Megrendelő	View map	
F08	Minden pumpán	A pumpa beállítása valamilyen	alapvető	Megrendelő	Change pump	

	külön-külön állítható, hogy melyik csőből melyikbe pumpálja a vizet.	állásba, majd megfigyeljük a két résztvevő csövet.			flow direction	
F09	Egy pumpána k egyszerre maximu m egy aktív be - és kimenete lehet Mindekö zben a többi rákötött cső eközben el van zárva.	Pumpa átállítása után meg kell vizsgálni hogy a korábbi be és kimenetek el legyenek zárva az újakban pedig folyjon a víz	alapvető	Megrendelő	Change pump flow direction	
F10	Csak akkor pumpálha t csőbe pumpa vizet, ha azt a cső szabad kapacitás a lehetővé teszi.	Pumpa átállításakor megvizsgálni az a csövet amelyikbe pumpálni szeretnénk. Ha ez a cső vízzel van tele akkor nem választható.	alapvető	Megrendelő	Simulate water flow	A cső csak üres vagy vízzel teli állapotba n lehet.
F11	A csöveget a szerelők áthelyezhetik. A csövek végpontja ik átköthető k más	Nem lehetséges áthelyezések/ illegális összeköttetések /kombinációk: forrás-forrás, ciszterna-ciszterna, ciszterna-forrás, forrás-ciszterna. Ha van legalább	alapvető	Megrendelő	Move repairma n	

	elemekbe .	egy végpontja és ezen nem illegális összeköttetés akkor áthelyezhető.				
F12	Új csövek a ciszternáknál alakulnak , melyek egyik vége szabadon bekötethető valamely elembe. Az újonnan létrejött csövek másik vége a ciszternába van kötve.	Amikor generáltan új cső keletkezik akkor végpontokat ellenőrizzük. Egyik végpont: ciszterna másik sivatag/semmi.	alapvető	Megrendelő	Lift tube-end, Place tube-end	
F13	A szabad végű csövekből a csőbe betáplált víz elfolyik a homokba.	A cső víztartalmának ellenőrzése a szimuláció után.	alapvető	Megrendelő	Simulate water flow	
F14	A lyukas csövekből a víz elfolyik a homokba.	A cső víztartalmának ellenőrzése a szimuláció után.	alapvető	Megrendelő	Simulate water flow	
F15	A pumpa nem tárol önmagában vizet.	A víz pumpálásakor az a bemeneti csőről azonnal a kimenetibe kerüljön	alapvető	Csapat	Simulate water flow	
F16	Véletlenszerű körönkén	Néhány kör szimulálása a játékban és	alapvető	Megrendelő	Damage pump	

	t a pumpák el tudnak romlani.	megfigyelni hogy elromlik-e valamelyik pumpa				
F17	A játék körökből áll.	A játék menetének áttekintése.	fontos	Csapat	Manage game	
F18	A szerelők tudnak pumpát javítani	Egy szerelőnek meg kell javítani egy véletlen elromlott pumpát	alapvető	Megrendelő	Repair elements	
F19	A szerelők át tudják állítani a pumpák aktív ki- és bemeneteit	A szerelőknek meg kell próbálnia állítani a pumpák be és kimenetét és sikkerrel kell járniuk.	alapvető	Megrendelő	Change pump flow direction	
F20	A szerelők tudnak csöveget foltozni.	Ha egy szerelő megpróbál befelozni egy lyukas csövet a művelet végére a csőnek épnek kell lennie.	alapvető	Megrendelő	Repair elements	
F21	A szerelők módosítják a csövek végpontja it.	A művelet végrehajtása után az adott cső végpontja egy másik elembe van kötve.	alapvető	Megrendelő	Lift tube-end, Place tube-end	
F22	A szerelők vihetnek új pumpákat magukkal a ciszternáktól.	A művelet (pumpa felvétele) után a játékos rendelkezik egy pumpával.	alapvető	Megrendelő	Move repairman	
F23	A szerelők	Ha egy szerelő megpróbál	alapvető	Megrendelő	Lift tube-end	

		kihúzhatn ak új csövet a ciszterná któl.	felvenni egy csövet az egyik ciszternánál akkor sikerrel kell járnia.				
F24	A szerelőknél egyszerre csak egy darab tárgy lehet.	Ellenőrizni kell, hogy szerelők vagy csak egy csövet vagy csak egy pumpát tudnak-e szállítani egy adott időpillanatban. minden további tárgyfelvételnek sikertelennek kell lennie.	fontos	Csapat	Lift tube-end, Lift pump		
F25	A szabotörök nem vihetnek magukkal tárgyakat.	Egy szabotör megpróbál felvenni egy csövet vagy pumpát. Ennek sikertelennek kell lennie	fontos	Csapat	Move saboteur		
F26	Ha a szerelők új pumpát helyezne ki el, akkor kettévágják azt a csövet, amelyre a pumpát helyezik.	A művelet befejeztével az eredeti cső helyett két új jön létre. Mindkettő csőnek az egyik vége a pumpába van kötve.	alapvető	Megrendelő	Place pump		
F27	A szabotörök kilyukaszthatnak csöveget.	Ha a szabotör megpróbál kilyukasztani egy ép csövet a művelet végére annak lyukasnak kell lennie.	alapvető	Megrendelő	Damage tube		
F28	A szabotörök át tudják állítani a pumpák	A művelet befejeztével a pumpa a kiválasztott csőbe pumpálja át a vizet.	alapvető	Megrendelő	Change pump flow direction		

	aktív csöveit.					
F29	A játékosok által irányított karakterek csak a csőhálózaton közlekedhetnek.	Ha a játékos megpróbál lelépni a csövekről akkor a próbálkozásnak sikertelennek kell lennie.	alapvető	Megrendelő	Move repairman, Move saboteur	
F30	Mindkét csapatban legalább 2-2 játékos van.	A játék kezdete előtt a körülmények áttekintése	alapvető	Megrendelő	összes	
F31	A szerelők célja, hogy minél több víz jusson el a forrásokból a ciszternába.	A szerelő csapat olyan stratégiát igyekszik alkalmazni, hogy minél több víz érjen a ciszternákba.	alapvető	Megrendelő	Simulate water flow, Calculate points	
F32	A szabotörök célja, hogy minél több víz folyjon el a sivatagban.	A szabotőr csapat olyan stratégiát igyekszik alkalmazni, hogy minél több víz folyjon el.	alapvető	Megrendelő	Simulate water flow, Calculate points	
F33	A csövön egyszerre csak egy karakter tartózkodhat.	Ha egy játékos megkísérrel egy olyan csőre lépni ahol már állnak, akkor a műveletnek sikertelennek kell lennie.	alapvető	Megrendelő	Move repairman, Move saboteur	

F34	A pumpán kikerülhetik egymást a karakterek.	Több játékos rááll a pumpára.	alapvető	Megrendelő	Move repairman, Move saboteur	
F35	Állítható, hogy hány kör után ér véget a játék.	Megfigyeljük, hogy a játék valóban a kezdéskör beállított körszámú-e.	opcionális	Csapat	Manage game	
F36	Olyan cső, mely forrásba vezet, nem létezhet.	Strukturális áttekintés	fontos	Csapat	View map, Simulate water flow	
F37	Forráson és ciszternán is ki tudják kerülni a játékosok egymást.	Több játékos rááll ezen elemekre.	fontos	Csapat	Move repairman, Move saboteur	
F38	Egy kör során csak az egyik csapat tevékenykedhet. Egy csapat minden második körben kerül sorra.	A játék menetének áttekintése.	fontos	Csapat	összes	

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
E01	A futtató számítógép erőforrásainak minimum meg kell felelnie egy HSZK-s számítógép képességei nek.	A készülő szoftvert teszteljük a HSZK által biztosított számítógépen.	alapvető	Megrendelő	
E02	A fejlesztéshoz szükséges 17-es vagy újabb verziójú JDK.	Megpróbáljuk régebben játszani, és ellentmondásra jutunk.	alapvető	Csapat	

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
A01	Billentyűzet szükséges a szoftver rendeltetés szerű használatához.	Megpróbáljuk billentyűzet nélkül játszani, és ellentmondásra jutunk.	fontos	Csapat	
A02	Egér szükséges a szoftver rendeltetés szerű használatához.	Megpróbáljuk egér nélkül játszani, és ellentmondásra jutunk.	fontos	Csapat	
A04	Megjelenítő eszköz szükséges a szoftver rendeltetés	Megpróbáljuk megjelenítő eszköz nélkül	fontos	Csapat	

	szerű használatához.	játszani, és ellentmondásra jutunk.			
A03	Java Runtime szükséges a szoftver futtatásához	Megpróbáljuk nélkülle játszani, és ellentmondásra jutunk.	alapvető	Csapat	

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	Move repairman
Rövid leírás	A játékos egy szerelőt irányít a csőhálózaton keresztül
Aktorok	Repairman Player
Főforgatókönyv	A játékos a szerelőt egy elemről egy másik szomszédosra mozgatja.
	1.A.1 Ha a cél mezőre nem léphet, akkor a játékos marad az eredeti pozícióján.

Use-case neve	Repair elements
Rövid leírás	A játékos megjavít elromlott elemeket.
Aktorok	Repairman Player
Főforgatókönyv	Ha a játékos egy elromlott elemen áll, akkor azt megjavítja.
	1.A.1. Ha az adott elem egy lyukas cső, akkor azt befoltozza.
	1.B.1 Ha az adott elem egy elromlott pumpa, akkor azt megjavítja.

Use-case neve	Move saboteur
Rövid leírás	A játékos egy szabotőrt irányít a csőhálózaton keresztül.
Aktorok	Saboteur Player
Főforgatókönyv	A játékos a szabotőrt egy elemről egy másik szomszédosra mozgatja.
	1.A.1 Ha a cél mezőre nem léphet, akkor a játékos marad az eredeti pozícióján.

Use-case neve	Damage tube
Rövid leírás	A szabotőr megrongál egy csővezetéket
Aktorok	Saboteur Player
Főforgatókönyv	A játékos által irányított szabotőr kilyukasztja azt a csövet, amin áll.

Use-case neve	Change pump flow direction
Rövid leírás	A játékos módosíthatja, hogy a pumpa melyik belekötött csőből melyik másikba pumpáljon.
Aktorok	Repairman Player, Saboteur Player
Főforgatókönyv	A játékos aktuális pozícióján egy pumpa található. 1.1 A játékos kiválasztja, melyik cső legyen a bemenet, és melyik a kimenet.

Use-case neve	View map
Rövid leírás	A játékos megtekinti a csőrendszert..

Aktorok	Repairman Player, Saboteur Player, Controller
Főforgatókönyv	A rendszer kirajzolja a pálya aktuális állapotát. A játékos megtekinti a pálya aktuális állapotát

Use-case neve	Simulate water flow
Rövid leírás	A játék szimulálja az aktuális játékelemek állapotai alapján a vízfolyást.
Aktorok	Controller
Főforgatókönyv	1.A Controller kifolyatja az összes lyukas csőből a vizet a sivatagba.
Alternatív forgatókönyv	2.A Controller végiglépked az összes nem cső elemen. 2.A.1. Az összes forrás vizet pakol a rá csatlakoztatott csövekbe, ha a kapacitás engedi. 2.B.1. Az összes ciszterna vizet vesz ki az összes rácsatlakoztatott csőből, ha lehetséges. 2.C.1. Az összes pumpa vizet emel át a bemeneti csőből a kimenetibe, ha ezt a cső kapacitása és a pumpa állapota engedi.

Use-case neve	Calculate points
Rövid leírás	A játék a lefutott vízsímsimuláció után megszámolja csapatok pontjait.
Aktorok	Controller
Főforgatókönyv	A Controller megszámolja, hogy mennyi víz érkezett a ciszternákba, és kiüríti azokat.
Alternatív forgatókönyv	1.A.1 A Controller megszámolja, hogy mennyi víz folyt el a csöveken.

Use-case neve	Manage game
Rövid leírás	A rendszer számolja a köröket.
Aktorok	Controller
Főforgatókönyv	A Controller minden csapat lépése után növeli a számlálót.
Alternatív forgatókönyv	Ha a számláló eléri a max. körök számát, a játék véget ér. A Controller összehasonlítja az összeszámol pontokat.

Use-case neve	Damage pump
Rövid leírás	A Controller elrontja a pumpát.
Aktorok	Controller
Főforgatókönyv	A Controller véletlen időközönként elrontja valamelyik pumpát.

Use-case neve	Lift pump
Rövid leírás	A szerelő magához vesz egy pumpát.
Aktorok	Repairman Player
Főforgatókönyv	A szerelő magához vesz egy pumpát a ciszternánál, ha még nincs nála semmi.

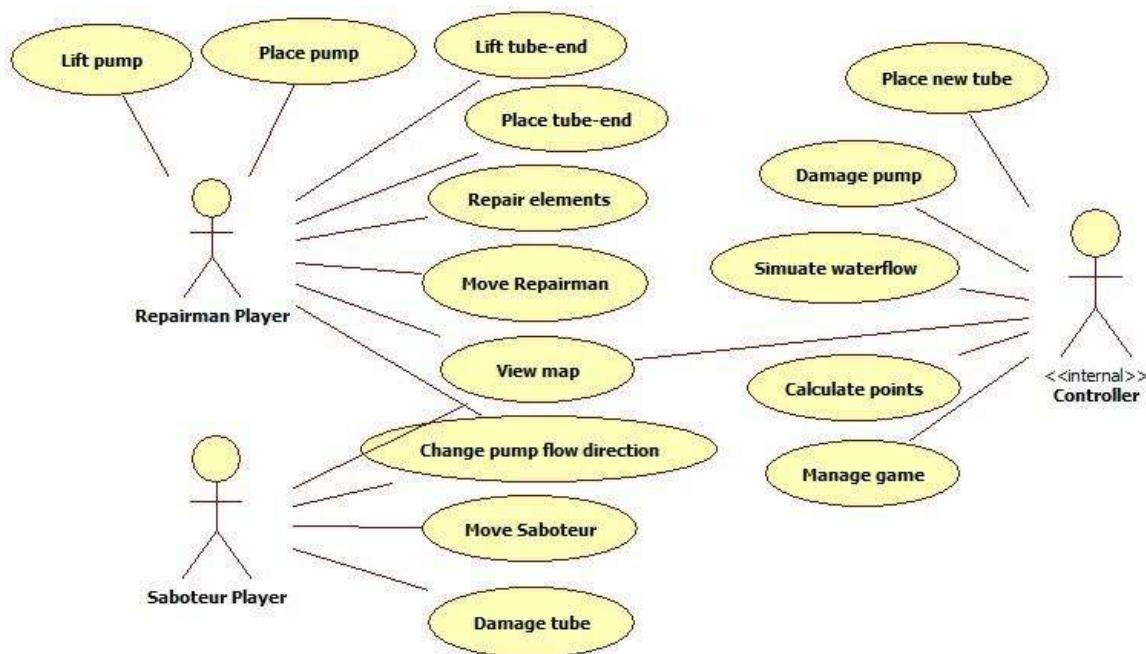
Use-case neve	Place pump
Rövid leírás	A szerelő lerak egy pumpát.
Aktorok	Repairman Player
Főforgatókönyv	A szerelő magához vett pumpát lehelyezi úgy, hogy ketté vág vele egy csövet.

Use-case neve	Lift tube-end
Rövid leírás	A szerelő magához vesz egy csővéget.
Aktorok	Repairman Player
Főforgatókönyv	A szerelő magához veszi az egyik cső egyik végét, ha még nincs nála semmi.

Use-case neve	Place tube-end
Rövid leírás	A szerelő lerakja a nála lévő cső végét.
Aktorok	Repairman Player
Főforgatókönyv	A szerelő beköti a nála lévő csővéget egy pumpába, forrásba vagy ciszternába.

Use-case neve	Place new tube
Rövid leírás	A Controller készít egy új csövet.
Aktorok	Controller
Főforgatókönyv	A Controller egy olyan csövet készít, melynek egyik vége az egyik ciszternához csatlakozik, a másik vége pedig szabad.

2.4.2 Use-case diagram



2.5 Szótár

Ciszterna:

Városok vízgyűjtői, amelyekbe a csöveken keresztül víz folyik.

Cső:

Alapvető elem melynek maximum két végpontja lehet: pumpa, forrás és ciszterna.

Kivételt képeznek: forrás-forrás, ciszterna-ciszterna, ciszterna-forrás, tehát ezek között nem lehet cső.

Minden csőnek van legalább egy végpontja, tehát nem találni csak úgy a sivatagban elhagyott csöveget.

Cső kilyukasztása:

Szabotör képesség, amelyben a cső amiben éppen állnak (és még nem lyukas), lehetőségük van kilyukasztani.

Cső megjavítása:

Szerelő képesség, amelyben egy lyukas cső amiben állnak megjavítható, így képes lesz ismét átjuttatni a végpontig a vizet(nem folyik el a sivatagba).

Cső szabad kapacitás:

Amikor a csőben éppen nincsen víz.

Ekkor lehet bele vizet pumpálni.

Csőrendszer:

Karakterek által bejárható terület, amely a pumpák és csövek teljes hálózatát tükrözi.

Csőrendszer bővítése:

Új pumpák elhelyezése.

Csőrendszer vizet szállít:

Azon folyamat amelyben a forrásokból kiindulva a víz áramlik, csöveken és pumpákon keresztül a ciszternák felé.

Csőrendszer változtatása:

Csövek áthelyezése, végpontjainak megváltoztatása.

Drukmákor sivatag:

Játék helyszíne.

Elromlott pumpa:

Olyan állapota a pumpának, amikor véletlenül meghibásodik.

Ilyenkor nem pumpálja tovább a vizet.

Szerelőre van szükség a helyreállításhoz.

Forrás:

A víz eredési pontja.

Innen folyik a víz a csőrendszeren keresztül a ciszternákba.

Hálózat kapacitása:

Csőrendszer maximum mennyi víz átvitelére képes a ciszternákba.

Játékos:

Valós személy, aki éppen játszik.

Karakter:

Játékos által választhatók és irányíthatók.

Kétféle létezik szabotőr és szerelő.

Karakterek mozgása a csőrendszeren:

Egy soron következő szabotőr vagy szerelő a jelenlegi helyzetétől elérhető(hozzá kapcsolódó) forrásra, ciszternára, csőre, pumpára való átlépése.

Kapcsolódó csónál azonban van még egy olyan feltétel, hogy csak abban az esetben mozoghat oda, ha más karakter jelenleg nem áll ott.

Kilyukadt cső:

A cső egy olyan állapota amikor elfolyik belőle a víz és nem tud teljesen átjutni rajta.

Tehát nem jut el a kimeneti pumpáig, ciszternáig hanem a sivatagba folyik.

Szabotőr tudja kilyukasztani a csöveget és ameddig egy szerelő meg nem javítja addig lyukas marad.

Pumpa:

Csövek összekötésére szolgáló alapelem, a csőrendszer része.

Egy pumpa számos csövet köthet össze, de egyszerre csak egy csőből kaphat vizet és ezt csak egy másik csőbe tudja tovább pumpálni.

Szabotőr:

Karakter amely képes mozogni a csőrendszeren, pumpákat átállítani(irányukat) valamint csöveget kilyukasztani.

Szerelő:

Karakter amely képes mozogni a csőrendszeren, pumpákat átállítani(irányukat), csöveget megjavítani valamit felvenni új pumpát a ciszternáknál és ezt elhelyezni csövek között.

Új pumpa elhelyezésekor először kettévág egy csövet és ezt követően csatlakoztatja őket a pumpához.

2.6 Projekt terv

A projekt végrehajtási lépései, azok határidejei¹:

Projekt végrehajtási lépés	Határidő	Heti felelős
Követelmény, projekt, funkcionalitás	március. 13. 14:15	Gutási
Analízis modell (I. változat)	márc. 20. 14:15	Andai
Analízis modell (II. változat)	márc. 27. 14:15	Bodnár
Szkeleton tervezése	ápr. 3. 14:15	Szikszai
Szkeleton elkészítése	ápr. 17. 14:15	Vajda
Prototípus koncepciója	ápr. 24. 14:15	Gutási
Részletes tervezek	máj. 3. labor	Andai
Prototípus elkészítése	máj. 8. 14:15	Bodnár
Grafikus változat tervezek	máj. 15. 14:15	Szikszai
Grafikus változat elkészítése	máj 31. labor	Vajda Gutási
Egyesített dokumentáció	jún. 2. 14.00	Andai

A dokumentumok közös szerkesztését a **Google Drive** teszi lehetővé, melyen keresztül a csapat tagjai minden, a projekthez tartozó dokumentumot közösen látnak és szerkeszthetnek. A forráskód megosztása a **GitHub** platformon történik, ami lehetővé teszi a kód verziókezelését, továbbá lehetővé teszi a közös fejlesztést.

A csapat minden, a projekthez kapcsolódó online meetinget a **Discord**on tart meg. Ezen kívül az általánosabb jellegű kommunikáció a **Messenger** platformon történik.

¹ Forrás <https://www.iit.bme.hu/targyak/BMEVIIIAB02/%C3%BCtemterv-hat%C3%A1rid%C5%91k> Letöltés ideje: 2023.03.11. 8:51

2.7 Napló

Kezdet	Időtartam	Résznevők	Leírás
2023.03.05 14:30	1,5 óra	Andai, Bodnár, Gutási, Sziksai, Vajda	Értekezlet. Általános ötletelés a projekt megvalósításával kapcsolatban. Döntés: Értekezletek időpontjai egyeztetve, Funkcionális követelmény azonosító rendszer kitalálása. Heti projekt felelősök kiosztása a projekt tervben (2.6. pont).
2023.03.08 16:00	1 óra	Andai, Bodnár, Gutási, Sziksai, Vajda	Értekezlet. Dokumentum felosztása. Döntések: <ul style="list-style-type: none"> - Andai lesz a projekt kapcsolattartója, továbbá elkészíti a 2.4 bekezdést, vele együtt a use case diagramot. - Bodnár elkészíti a 2.2.3, 2.2.4, 2.3.2, 2.5 bekezdéseket. - Gutási elkészíti a 2.1.5, 2.2.1, 2.6 bekezdéseket és felügyeli a naplázást. - Sziksai elkészíti a 2.1.3, 2.3.3 bekezdéseket. - Vajda elkészíti a 2.1.4, 2.2.5 bekezdéseket. - Sziksai és Vajda közösen elkészítik a 2.2.2 bekezdést. - A fenti feladatok határideje 2023.03.12.
2023.03.08 17:00	1,5 óra	Andai, Bodnár, Gutási, Sziksai, Vajda	Értekezlet. Funkcionális követelmények, use case és Funkciók átbeszélése, közös ötletelés. Döntés: <ul style="list-style-type: none"> - mindenki igyekszik kiegészíteni a 2.3.1 pontot
2023.03.09 8:00	1 óra	Andai	Use-case leírások, use-case diagram
2023.03.09 9:00	1 óra	Andai	Use-case leírások, use-case diagram
2023.03.10 10:00	1 óra	Andai, Bodnár, Gutási, Sziksai, Vajda	Értekezlet. A 2.3.1 szakaszhoz kitalált ötletek bemutatása, ezek közös pontosítása, értékelése. Döntés: <ul style="list-style-type: none"> - A 2.3.1 szakaszt közösen szerkesztjük meg, javítjuk az esetlegesen felmerülő problémákat. mindenki megközelítőleg 7-7db funkcionális követelményt ír meg a táblázatba (annak minden kitöltendő sorával.)

2023.03.10 11:00	1 óra	Bodnár	2.2.4, 2.5 szakaszok megírása
2023.03.10 21:30	1 óra	Szikszai	Rövidítés és definíciók bővítése, Funkcionális követelmények bővítése és Use Case-ekkel való összekapcsolása
2023.03.11 8:00	1 óra	Gutási	2.1.5, 2.2.1, 2.6 szakaszok megírása.
2023.03.11 9:30	1 óra	Szikszai	Funkciók szerkesztése, Definíciók bővítése
2023.03.11 11:00	1 óra	Vajda	Hivatkozások és kapcsolataik megírása, funkciók bővítése, funkcionális követelmények ellenőrzésének bővítése
2023.03.12 11:00	1 óra	Gutási	Dokumentum formázása, előkészítés nyomtatásra.
2023.03.12 12:00	1 óra	Gutási	Napló pontosítása.
2023.03.12 13:00	1 óra	Andai	Funkcionális követelmények kiegészítése.
2023.03.12 13:30	1 óra	Szikszai	Funkcionális követelmények pontosítása és Funkciók bővítése, formázása.
2023.03.12 14:00	1 óra	Bodnár	Szótár pontosítása és bővítése
2023.03.12 15:00	1 óra	Vajda	funkciók és funkcionális követelmények kiegészítése

2023.03.20.

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Ciszterna:

Szammon tartja mennyi víz folyik a ciszternába, valamint új pumpákat és a ciszternához kapcsolódó csöveket generál.

3.1.2 Cső:

Szammon tartja, hogy az adott cső éppen lyukas-e, mennyi víz folyik ki belőle és azt is, hogy a két vége mihez kapcsolódik.

3.1.3 Forrás:

Minden belekötött csőbe vizet küld, (amint egy csapat befejezte a lépést).

3.1.4 Pumpa:

A pumpa aktuális bemeneti és kimeneti csöveit karbantartja, valamint azt, hogy éppen hibás-e a pumpa.

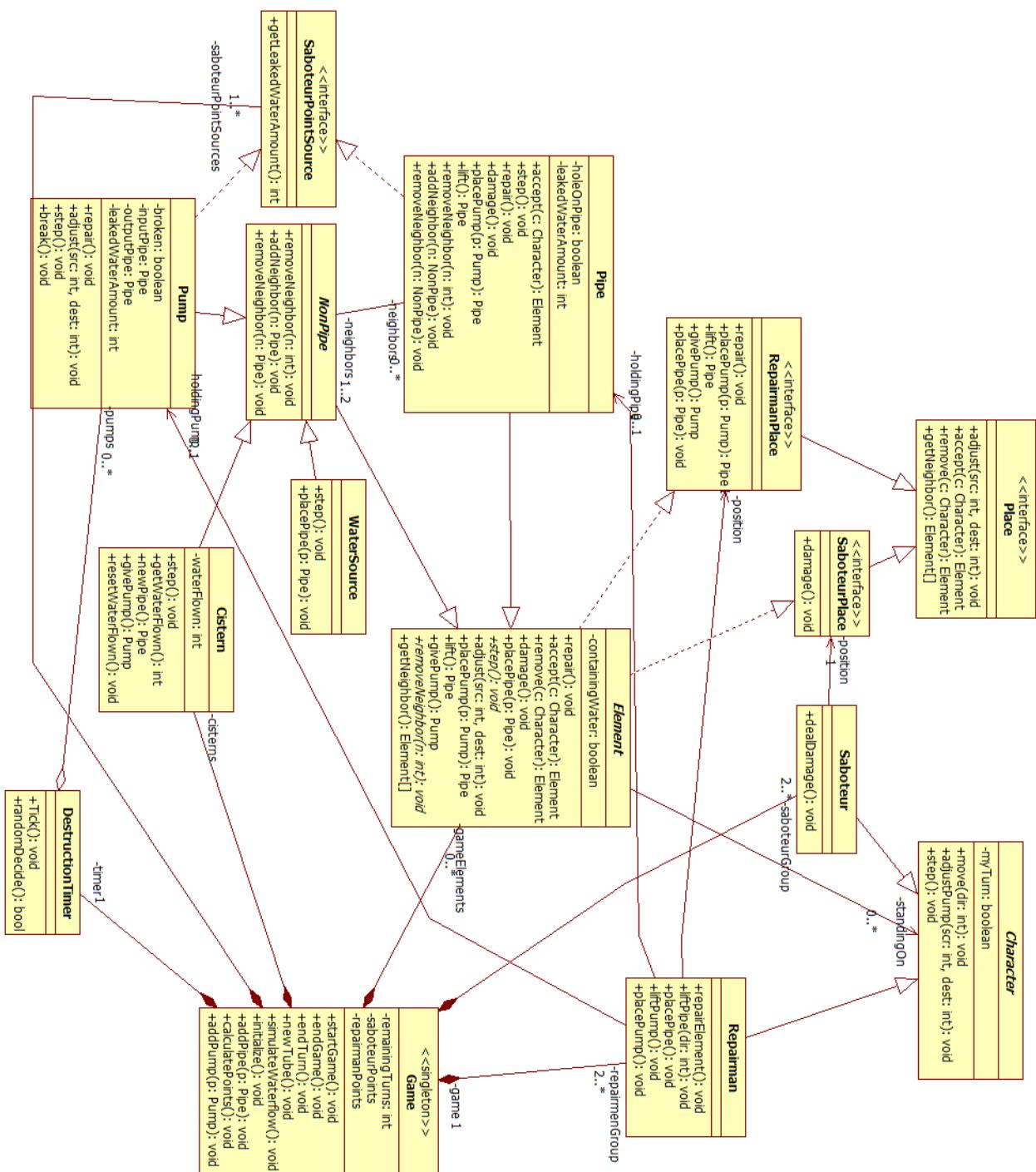
3.1.5 Szabotőr:

A szabotőr karakter speciális képességeit zárja egységbe.
Képes kilyukasztani csöveket.

3.1.6 Szerelő:

A szerelő karakter speciális képességeit zárja egységbe.
Tud javítani csövet és pumpát is, ciszternánál felvenni pumpát és egy cső kettévágását követően elhelyezni.

3.2 Statikus struktúra diagram



3.3 Osztályok leírása

3.3.1 Character

Felelősség

Ebből az osztályból származnak le a játékosok által irányítható karakterek (szerelők, és szabotőrök).

Attribútumok

- **bool myTurn:** A step() állítja be igazra, ha az adott karakter van soron (ő léphet). Miután lépett, hamisra állítódik.

Metódusok

- **void move(int dir):** Karakter átléptetése egy irányba. Az irány egy szám, ami egyértelműen azonosítja egy elem szomszédait.
- **void adjustPump(int src, int dest):** Pumba aktív ki- és bemenő élének beállítása (meghívja a Place interfész adjust() metódusát). A két szám egyértelműen azonosít egy-egy csövet, ami a pumpába van kötve.
- **void step():** A Game osztály hívja meg, ezzel kiválasztva a karaktert, hogy ő léphet. Ezen metódus elején állítódik be a myTurn attribútum igazra, ezzel jelezve, hogy a bemenet erre a karakterre irányul. Miután a karakter lépett, a myTurn értéke ismét hamis lesz, és a step() metódus visszatér.

3.3.2 Cistern

Felelősség

A vízhálózat egyik végpontjaként szolgáló víznyelő. Tárolja a belefolyt víz mennyiségét lekérdezésig. Új, belőle félíg szabadba kivezető csövet hoz létre, ha nincs ilyen. A szerelők fel tudnak róla venni pumpát.

Ősosztályok

Element→NonPipe→Cistern

Interfészek

Megvalósítja a **Place** -, ezáltal a **RepairmanPlace** és a **SaboteurPlace** interfészt is.

Attribútumok

- **int waterFlown:** A lépés alatt a ciszternába folyt víz mennyisége.

Metódusok

- **void newPipe():** Létrehoz egy új csövet ami félíg szabad és félíg a ciszternába van kötve, ha nem létezik másik ilyen cső.
- **void Step():** Kiveszi a vizet a belé csatlakozott csövekből. Ezután meghívja az adott cső step() metódusát is, ami rekurzívan továbbhívja a step()-et a NonPipe szomszédján. Ha a NonPipe-nak van vize, akkor azt továbbadja annak a csőnek, aki hívta rajta a step() metódust. Ez rekurzívan folytatódik addig, ameddig el nem érünk egy forrást így, vagy el nem érkezünk egy olyan pumpához, amiben hurokél van. Ez lejátszódik a ciszterna összes belecsatlakoztatott csövén. Végül a getWaterflown érték annyival növekszik, amennyi víz a becsatlakoztatott csöveken keresztül beérkezett.

- **int getWaterFlown()**: Visszaadja a ciszternába belefolyt víz mennyiségét.
- **void resetWaterFlown()**: Visszaállítja a ciszternába folyt értékét nullára. Ezután vissza is állítja ezt az értéket nullára.
- **Pump givePump()**: Létrehoz egy pumpát és visszaadja azt. A karakter kérhet új pumpát a ciszternától.

3.3.3 DestructionTimer

Felelősség

Véletlen eséllyel elrontja a pumpákat.

Attribútumok

- **Pump[] pumps**: tárolja a játékban szereplő pumpákat

Metódusok

- **void tick()**: Végigitáról az összes pumpán és adott eséllyel elrontja azokat.
- **boolean randomDecide()**: eldönti hogy az adott pumpa elromoljon-e

3.3.4 Element

Felelősség

Közös absztrakt őse a pályaelemeknek. Megvalósítja a karakterek számára szükséges függvényeket, azok alap implementációit megvalósítja.

Interfészek

Megvalósítja a **SaboteurPlace** és **RepairmanPlace** interfészeket, ezáltal a **Place** interfészt is.

Attribútumok

- **bool containingWater**: Ez tárolja, hogy az adott elemben van-e éppen víz.
- **Character[0..*] standingOn**: Lista a pályaelemen álló játékosokról.

Metódusok

- **void repair()**: Ezen a metóduson keresztül lehet őt megjavítani.
- **void adjust(int src, int dest)**: Pumpa aktuális bemenetét(src) és kimenetét(dest) állítja át sorszámaik alapján.
- **Character[0..*] getstandingOn()**: Az elemen álló karakterek lekérdezése.
- **bool accept(Character c)**: Őt kell meghívni ha rá szeretnénk lépni a mezőre. A karakterünket kell átadni paraméterként. Azzal tér vissza, hogy sikeres volt-e a rálépés.
- **Element remove(Character c)**: Őt kell meghívni ha le szeretnénk lépni a mezőről. A karakterünket kell átadni paraméterként. Azzal tér vissza, hogy sikeres volt-e a lelépés.
- **void damage()**: Rajta keresztül lehet megrongálni az elemet.
- **NonPipe givePump()**: Megpróbálunk kérni tőle egy pumpát. A különböző leszármazottak máshogyan implementálják majd. Ha tud pumpát adni, akkor NonPipe típusként adja vissza, mert így kompatibilis a szomszédok menedzselsével.
- **void placePump(Pump p)**: Megpróbálunk rajta keresztül lerakni egy pumpát. A lerakandó pumpát paraméterben adjuk át neki.
- **Pipe lift(int dir)**: Ezen keresztül lehet majd megpróbálni felemelni a kívánt elemet. A paramétere egy irány, ami a meghívott objektum adott irányú szomszédjára utal. Visszatér az adott irányban lévő szomszédos csővel, és lecsatlakoztatja magáról azt.
- **void placePipe(Pipe p)**: Ezen keresztül lehet a csövek végét lehelyezni. Paraméterben kell megadni neki a csövet.
- **bool getContainingWater()**: containingWater attribútum lekérdezése
- **bool setContainingWater()**: containingWater attribútum beállítása.
- **Element[] getNeighbor()**: Visszaadja az adott Element szomszédait.
- **void step()**: Absztrakt függvény, amely a vízfolyást teszi lehetővé.

3.3.5 Game

Felelősség

Kezeli a játék indítását, befejezését és a vízfolyás szimulációját. Továbbá tárolja a csapatokat, benne a játékban szereplő karakterekkel, ezen csapatok pontjait és a játék egyéb elemeit is. Kezeli a víz szimulációját, és a körök felosztását.

Attribútumok

- **Repairman[2..*] repairmanGroup:** A szerelő csapatot alkotó karaktereket tárolja.
- **Saboteur[2..*] saboteurGroup:** A szabotőr csapatot alkotó karaktereket tárolja.
- **int repairmanPoints:** A szerelő csapat pontjait tárolja.
- **int saboteurPoints:** A szabotőr csapat pontjait tárolja.
- **Element[0..*] gameElements:** minden a játékban található elementet tárol
- **Cistern[1..*] cisterns:** Tárolja a játékban szereplő ciszternákat.
- **saboteurPointSource[1..*] saboteurPointSources:** Olyan elemeket tárol, melyekből kaphatnak pontot a szabotőrök.
- **int remainingRounds:** A játékból hátramaradó körök száma. Egy kör több lépésből áll. Először a Game meghívja a DestructionTimer Tick() metódusát. Ezután először az egyik csapat összes karaktere, majd a másik csapat összes karaktere lép. minden karakter lépése után szimulálódik a vízfolyás, és hozzáadódnak a pontok a csapatok számára.
- **DestructionTimer timer:** időzítő amely a pumpák véletlen elromlásáért felelős

Metódusok

- **void initialize():** Játék inicializálása. Betölti az előre megtervezett kiinduló pályát. Létrehozza az ezt a pályát reprezentáló objektumokat.
- **void playGame():** A játék fő metódusa. Addig tart, ameddig van következő kör.
- **void endGame():** Játék befejezése, kiírja a győztes csapatot.
- **void SimulateWaterflow():** Vízfolyás szimulálása a kör végén.
- **void addElement(Element e):** Objektum hozzáadása a pályához.
- **Element[] getGameElements():** A játékban szereplő elemek lekérdezése.
- **void addSaboteur():** Létrehoz és hozzáad egy Szabotőrt a játékhoz.
- **void addRepairman():** Létrehoz és hozzáad egy Szerelőt a játékhoz.
- **void addPipe(p: Pipe):** Hozzáad egy csövet a játékhoz.
- **void calculatePoints():** A két csapat pontszámainak növelése.
- **int getSaboteurPoints():** A szabotőr csapat pontjainak visszaadása.
- **int getRepairmanPoints():** A szerelő csapat pontjainak visszaadása.
- **void addPump(p:Pump):** Hozzáad egy pumpát a játékhoz.

3.3.6 NonPipe

Felelősség

Absztrakt osztály, amely magában foglalja minden csőtől eltérő helyeket ahova léphet a játékos.

Fő feladata karbantartani/tárolni az aktuális szomszédokat, tehát azon csöveket amelyek kapcsolódnak az adott nemesőhöz.

A program későbbi bővíthetőségét könnyíti.

Ősosztályok

Element → NonPipe

Interfészek

Megvalósítja a **SaboteurPlace** és **RepairmanPlace** interfészeket, ezáltal a **Place** interfészt is.

Attribútumok

- **Pipe[] neighbors** A hozzá csatlakozó csöveket tárolja.

Metódusok

- **void addNeighbor(Pipe _pipe):**

Cső csatlakoztatásakor bővíti a neighbors attribútumot az új csővel.

- **void removeNeighbor(Pipe _pipe):**

Cső lecsatlakoztatásakor törli az adott csövet a neighbors-ból.

- **Pipe[] getNeighbors():**

Visszaadja az összes szomszédos csövet.

- **Pipe[] getNeighbor(int dir):**

Visszaadja a paraméterként megadott szomszédos csövet.

3.3.7 Pipe

Felelősség

Csőként funkcionál, lyukasztható, foltozható, összecsatlakoztatható más nem csövekkel. Nyilvántartja a nem cső szomszédjait. Egyszerre egy játékos állhat rajta ezt számon tartja. Számolja a rajta kifolyt víz mennyiségét. Ezt le lehet tőle kérdezni. Engedi, hogy rakjanak bele vizet, illetve hogy kivegyenek belőle.

Ősosztályok

Element→Pipe

Interfészek

Megvalósítja a **SaboteurPlace** és **RepairmanPlace** interfészket, ezáltal a **Place** interfészt is. Megvalósítja a **SaboteurPointSource** interfészt.

Attribútumok

- **NonPipe[] neighbors:** a szomszédait tároló lista
- **bool holeOnPipe:** van-e rajta lyuk
- **int leakedWaterAmount:** számolja a rajta kifolyó víz mennyiségét

Metódusok

- **int getLeakedWaterAmount():** Visszaadja, a leakedWaterAmountot, és lekérdezés után visszaállítja azt nullára.
- **bool accept(Character c):** Felüldefiniálja az Element Accept metódusát.
- **void step():** Ha lyukas, kifolyatja magából a vizet és növeli a LeakedWaterAmountot
- **void repair():** holeOnPipe-ot false-ra állítja.
- **void damage():** holeOnPipe-ot true-ra állítja.
- **void removeNeighbor(int n):** A paraméterül kapott irányú szomszédját eltávolítja.
- **addNeighbor(NonPipe n):** Cső szomszédjának beállítása
- **Pipe lift():** Olyan cső felemelésénél használatos, amelyiknek az egyik fele nincs sehol sem beköve. Ekkor a cső egyik végének a felemeléséhez rajta kell állni ezen a csövön.
- **Pipe placePump(Pump p):** Létrehoz egy új csövet, majd közé és a meglévő cső közé a nála lévő pumpát helyezi el. Visszaadja a új csövet.

3.3.8 Place

Felelősség

Interfész, ami azokat a metódusokat tartalmazza, amelyeket minden karakter elérhet a játék összes elemén.

Metódusok

- **void adjust(src: int, dest: int):** Pumba aktuális bemenetét(src) és kimenetét(dest) állítja át sorszámaik alapján.
- **bool accept(c: Character):** Karakter felhelyezése az adott elemre. Ha az elem elfogadja a karaktert, igaz értékkel tér vissza. Ellenkező esetben hamis értékkel tér vissza.
- **Element remove(c: Character):** Karakter levétele az adott elemről. Ha az elem sikeresne végrehajtja a karakter levételét, igaz értékkel tér vissza. Ellenkező esetben hamis értékkel tér vissza.
- **Element[] getNeighbor():** Visszaadja az adott elem szomszédjait.

3.3.9 Pump

Felelősség

A pumpának mint aktív elem, karbantartja a bemenetét és kimenetét.
Azaz melyik csőből kap vizet és melyikbe pumpálja tovább.

Ősosztályok

Element □ Nonpipe

Interfészek

Megvalósítja a **SaboteurPlace** és **RepairmanPlace** interfészket, ezáltal a **Place** interfészt is.
Megvalósítja a **SaboteurPointSource** interfészt.

Attribútumok

- **bool broken** Tud-e vizet pumpálni az adott pumpa
- **Pipe inputPipe** Az a cső ahonnan szívja a vizet.
- **Pipe outputPipe** Az a cső ahova pumpálja a vizet.

Metódusok

- **Pipe getInputPipe():**

Visszaadja az inputPipe attribútumot, melyik csőből kapja aktuálisan a vizet.

- **void setInputPipe(Pipe _pipe):**

Beállítja inputPipe attribútumot a paraméterként megadott Pipe-ra.

- **Pipe getOutputPipe(Pipe _pipe):**

Visszaadja az outputPipe attribútumot, melyik csőbe pumpálja éppen a vizet.

- **void setOutputPipe(Pipe _pipe):**

Beállítja outputPipe attribútumot a paraméterként megadott Pipe-ra.

- **void adjust(int src, int dest):**

Pumpa aktuális bemenetét(src) és kimenetét(dest) állítja át sorszámaik alapján.

- **void repair():** A broken attribútumot true-ra állítja. Ez akkor hívódik amikor egy elromlott pumpát megjavít egy szerelő.

3.3.10 Repairman

Felelősség

Játékos által irányított szerelő karakter. Célja, hogy minél több víz jusson el a forrásoktól a ciszternákba.

Ősosztályok

Character→Repairman

Attribútumok

- **RepairmanPlace position:** Ahol éppen a Szerelő áll.
- **Pipe holdingPipe:** A felvehető csővég tárolására szolgál.
- **Pump holdingPump:** A felvehető Pumpa tárolására szolgál.
- **Game game:** Tárolja a játék egy referenciaját, mely az új Csövek és Pumpák eltárolásához szükséges.

Metódusok

- **void RepairElement():** Megjavítja azt az Element játékelemet, amin áll. Megtörténhet, hogy ezt egy olyan elemen állva végzi, ami nem tud elromlani/megjavulni. Ekkor nem történik semmi.
- **void LiftPipe(int dir):** Ha a játékos nem csövön áll, akkor az adott elementnek amin áll a paraméterben átadott számú szomszédját magához veszi
- **void LiftPump():** Ha a játékos egy ciszternán áll, akkor magához vehet egy új pumpát.
- **void PlacePump():** Ha a játékos egy csövön áll, lehelyezheti a nála levő pumpát.
- **void PlacePipe():** Ha a játékos épp nem egy csövön áll, akkor lehelyezheti a nála lévő csövet.
- **Pump getHoldingPump():** Visszaadja a karakternél levő pumpát.
- **void setHoldingPump(Pump p):** Beállítja a karakternél levő pumpát.
- **Pipe getHoldingPipe():** Visszaadja a karakternél levő csövet.
- **void setHoldingPipe(Pipe p):** Beállítja a karakternél levő csövet.

3.3.11 RepairmanPlace

Felelősség

Interfész, ami azokat a metódusokat tartalmazza, amelyeket a szerelők elérhetnek a játék elemein.

Ősosztályok

Place interfész

- **Metódusok**

- **void repair():** A karakter hívja ezt a metódust. Hatására a pályaelem megjavul.
- **void placePump(Pump p):** Elhelyezi az adott pumpát egy csövön, lényegében kettévágva azt.
- **Pipe lift(int dir):** Az elem egy adott számmal azonosított csövének lecsatlakoztatása. Visszaadja a játékosnak a csövet, amit lecsatlakoztatott. A cső ezen elembe mutató végét null-ra állítja
- **Pump givePump():** Létrehoz egy új pumpát, amit utána odaad a játékosnak. Ez ciszternákon működik.
- **void PlacePipe(Pipe p):** Beköti a paraméterként kapott csövet az interfészt megvalósító elembe.

3.3.12 Saboteur

Felelősség

A játékosok által irányított szabotőrök, céljuk hogy minél több víz folyjon el a sivatagba a játék során.

Ősosztályok

Character → Saboteur

Attribútumok

- **SaboteurPlace position:** Elem, amin a szabotőr éppen áll.

Metódusok

- **void dealDamage():** Kilyukasztja az adott csövet amin áll amennyiben az még nem volt lyukas

3.3.13 SaboteurPlace

Felelősség

Interfész, ami azokat a metódusokat tartalmazza, amelyeket a szabotőrök elérhetnek a játék elemein.

Ősosztályok

Place interfész

- **Metódusok**

- **void damage():** Megrongálja az adott elemet. Cső esetén kilyukasztja azt, egyéb elemeken nem hajt végre semmit.

3.3.14 SaboteurPointSource

Felelősség

Interfész, ami azokat a metódusokat tartalmazza, amelyek a szabotőrök pontjainak meghatározásához szükséges..

- **Metódusok**

- **int getLeakedWaterAmount():** Visszaadja, mennyi víz folyt el az adott szimulációs körben és lekérdezés után visszaállítja azt nullára.

3.3.15 WaterSource

Felelősség

Korlátlan vízforrás, tőle indul a víz áramlása.

Ősosztályok

Element→NonPipe

Interfészek

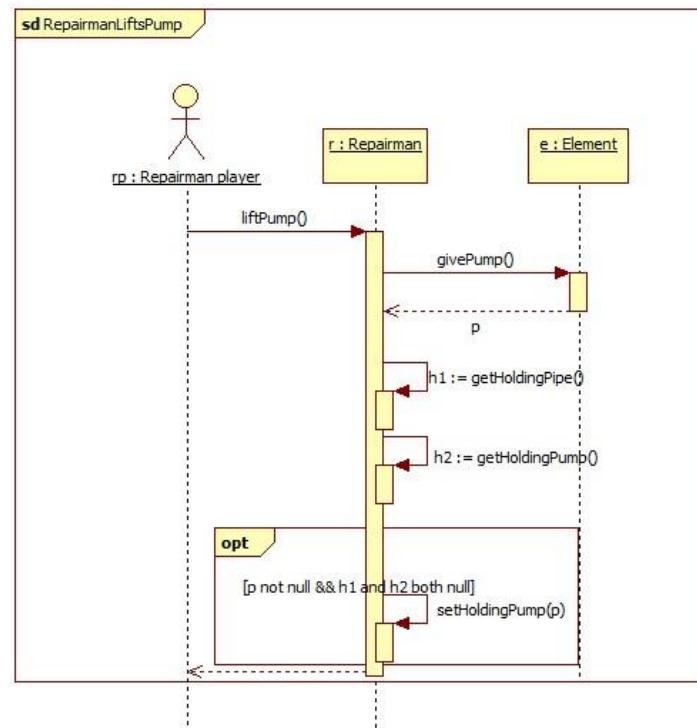
Megvalósítja a **SaboteurPlace** és **RepairmanPlace** interfészket, ezáltal a **Place** interfészt is.

Metódusok

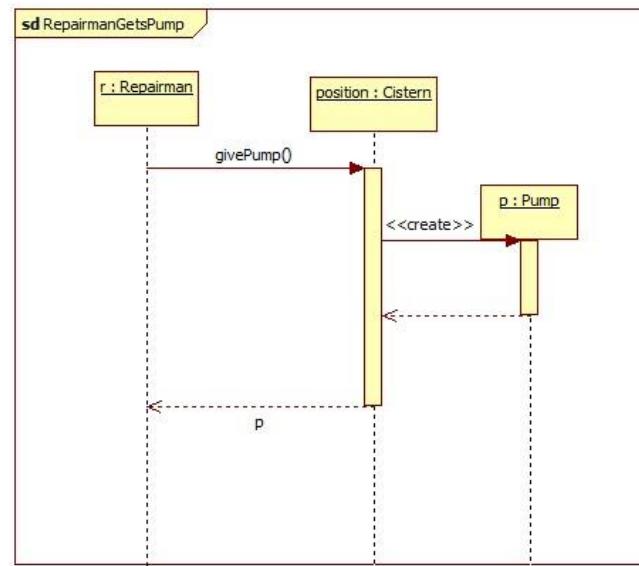
- **void step()**: Elindítja a víz áramlását a Ciszternából, minden bele csatlakozó csövet megtölt vízzel, ha van szabad kapacitása.
- **void placePipe(p: Pipe)**: Amennyiben a paraméterben kapott cső egy másik forrásból származik akkor nem fogadja el, egyéb esetben igen.

3.4 Szekvencia diagramok

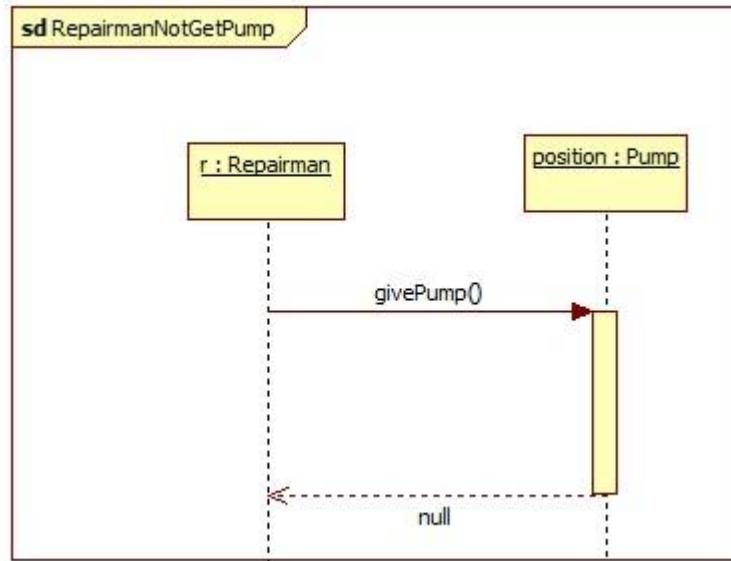
3.4.1 RepairmanLiftPump



3.4.2 Repairman receives pump

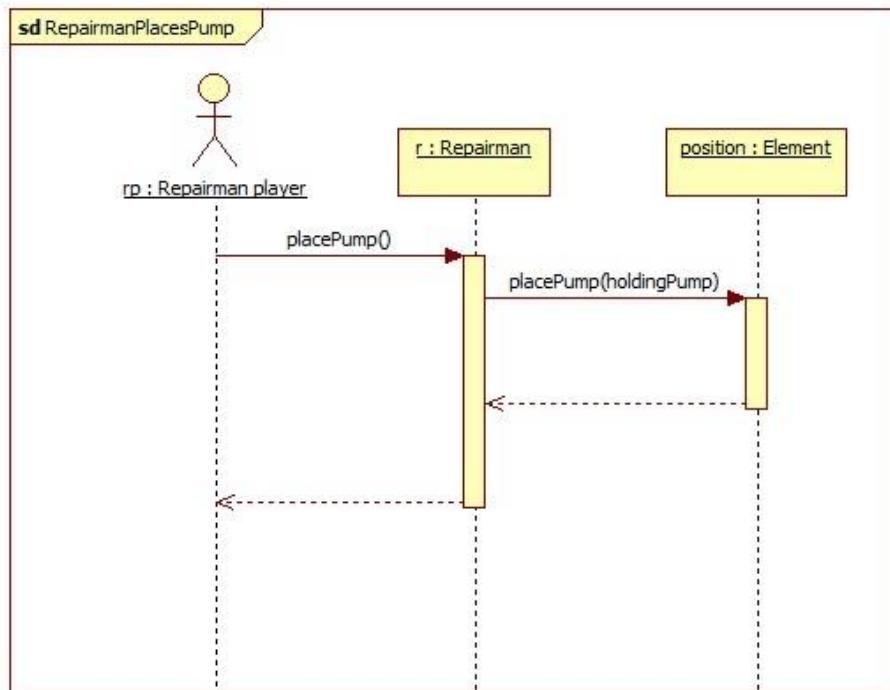


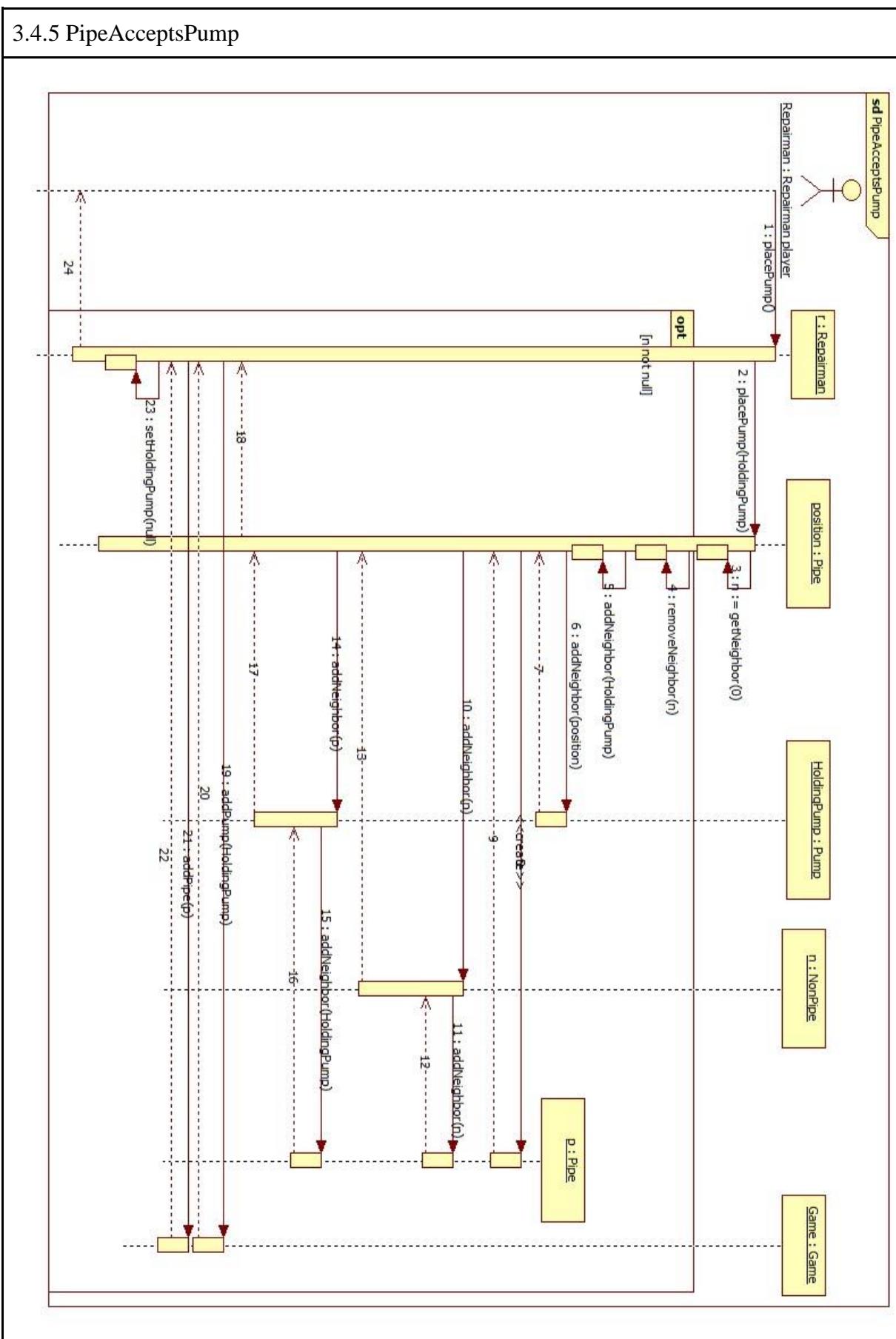
3.4.3 Repairman not receiving pump



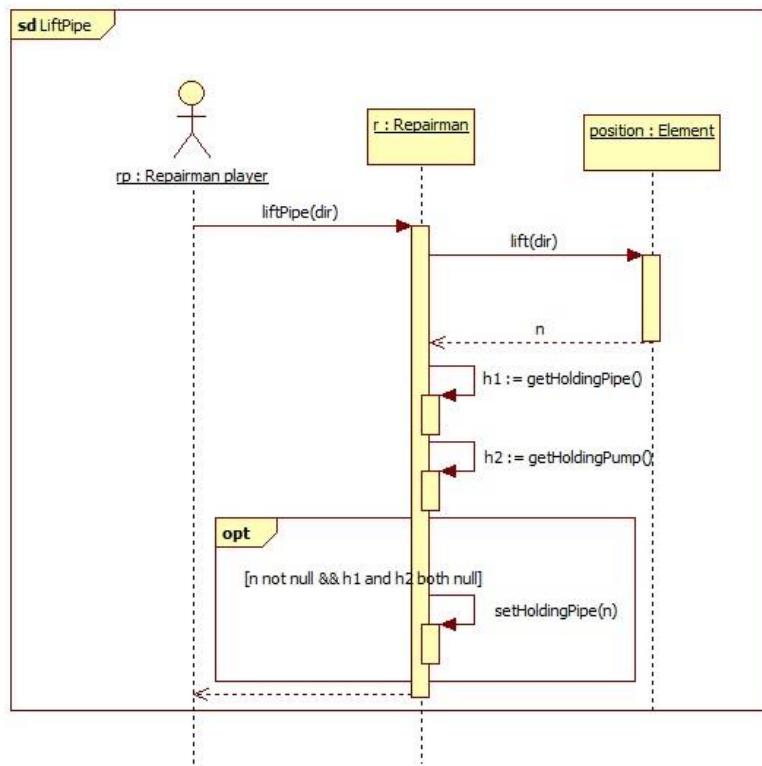
Ugyanígy viselkedik a forrás és a cső is.

3.4.4 RepairmanPlacesPump

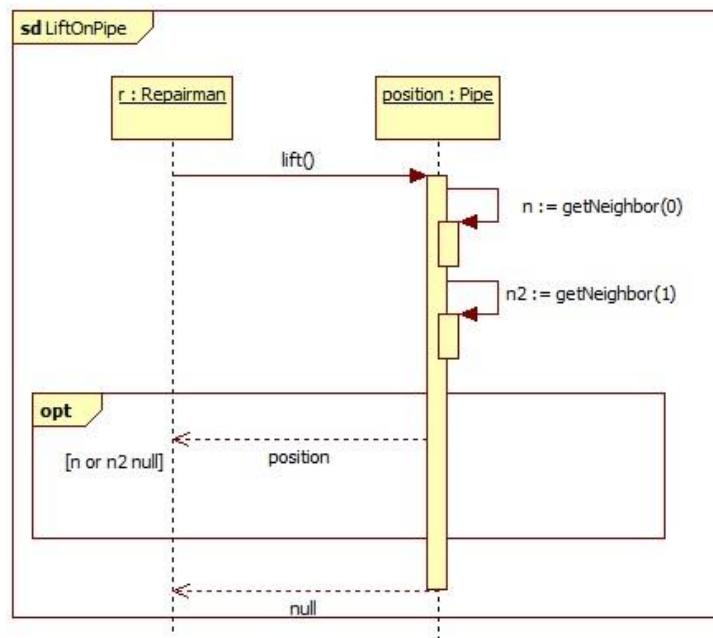




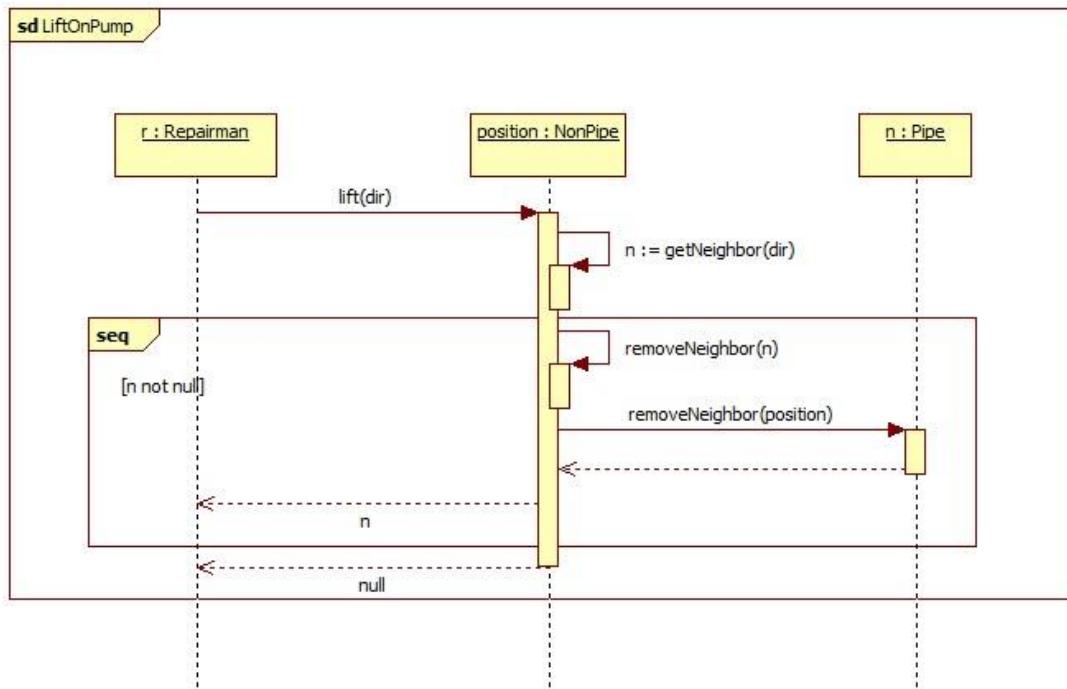
3.4.6 RepairmanLiftsPipe



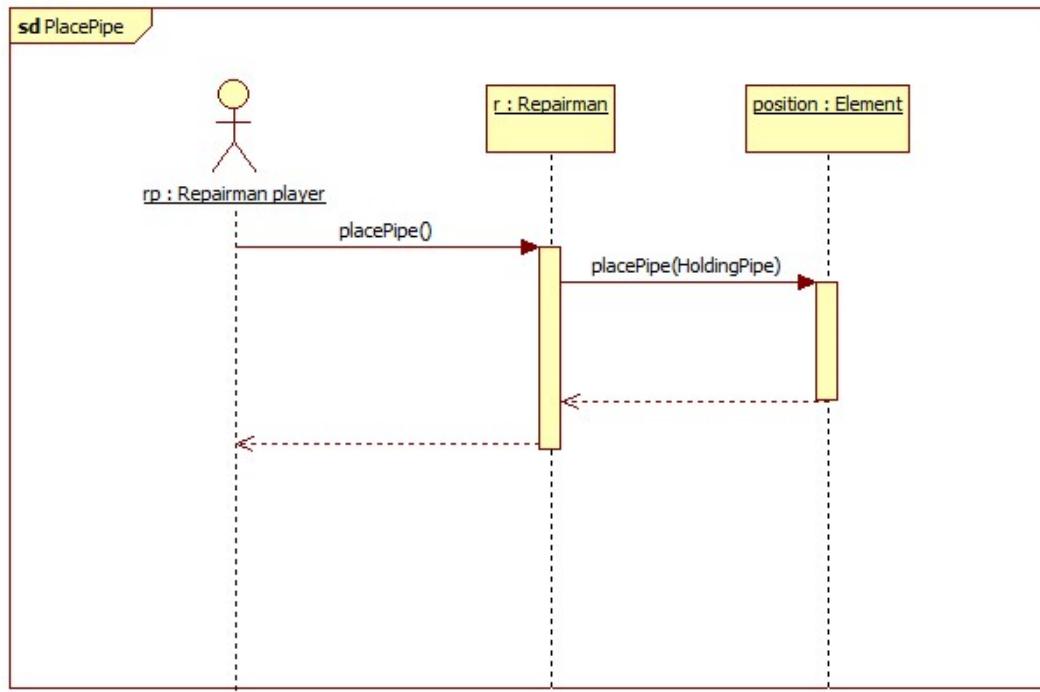
3.4.7 Lift Pipe, character standing on pipe



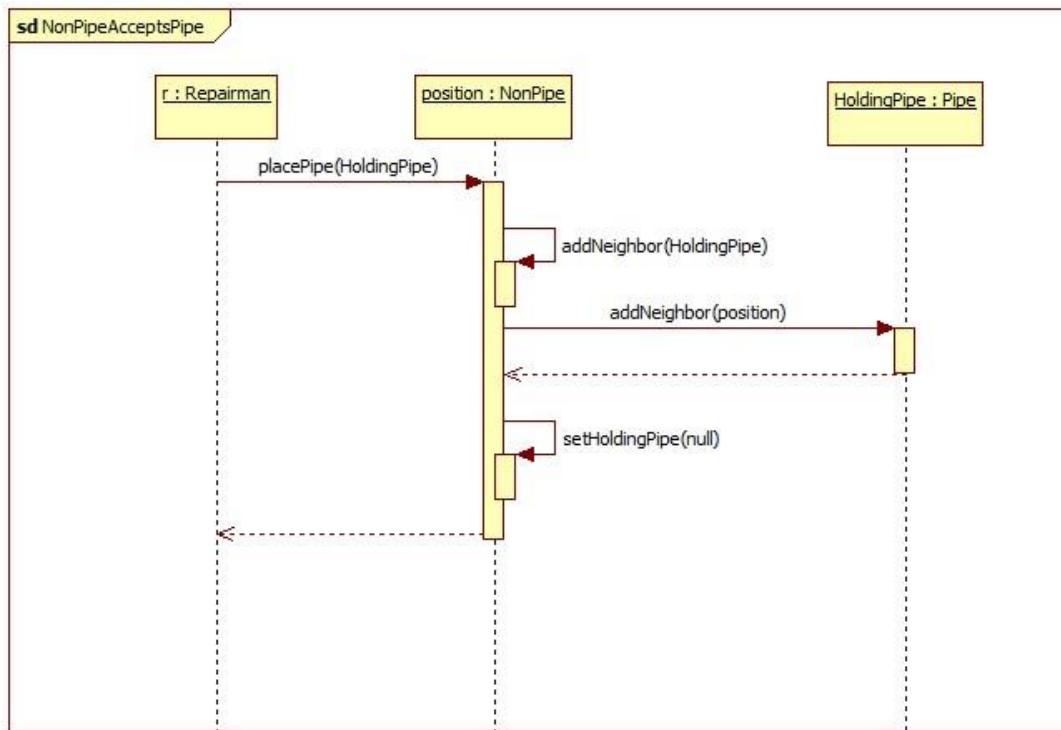
3.4.8 Lift Pipe, character standing on pump



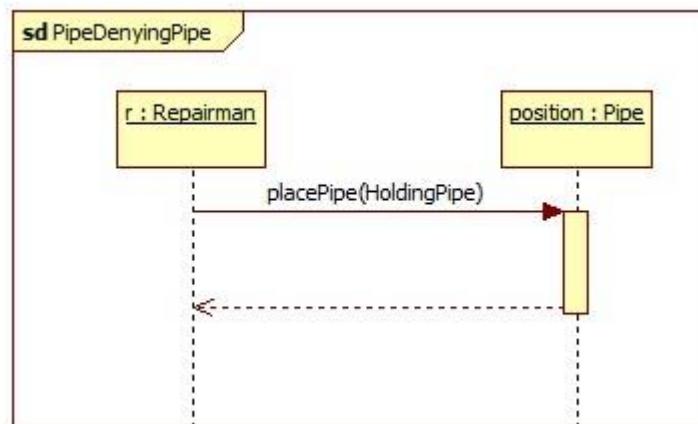
3.4.9 Repairmaqn placing pipe.



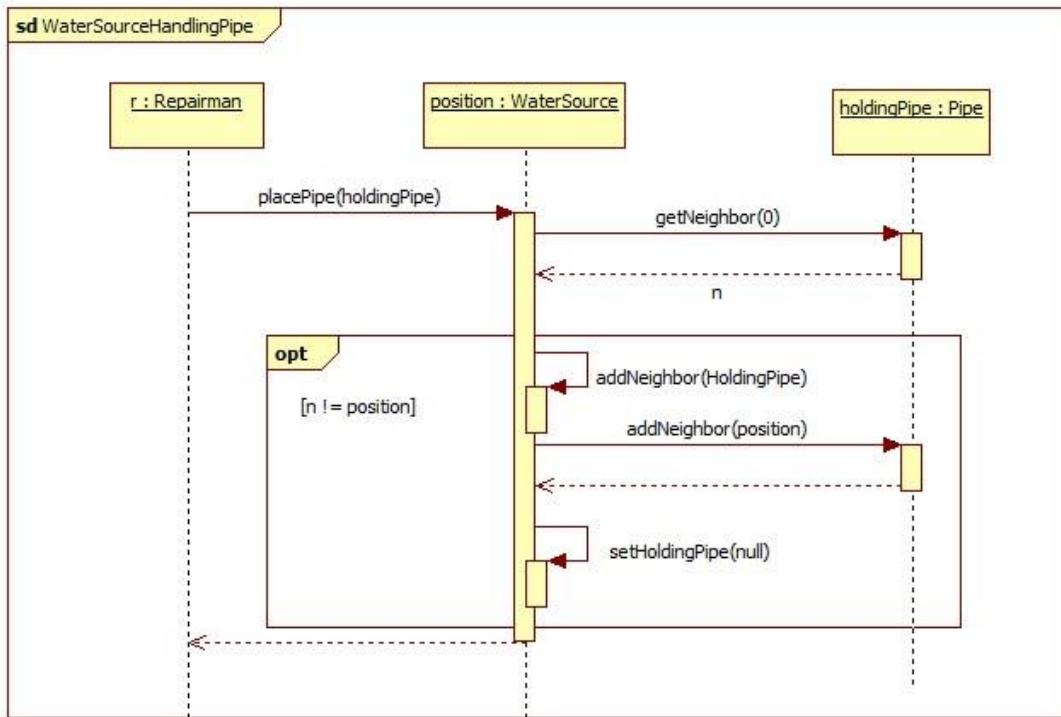
3.4.10. Nonpipe accepting a pipe.



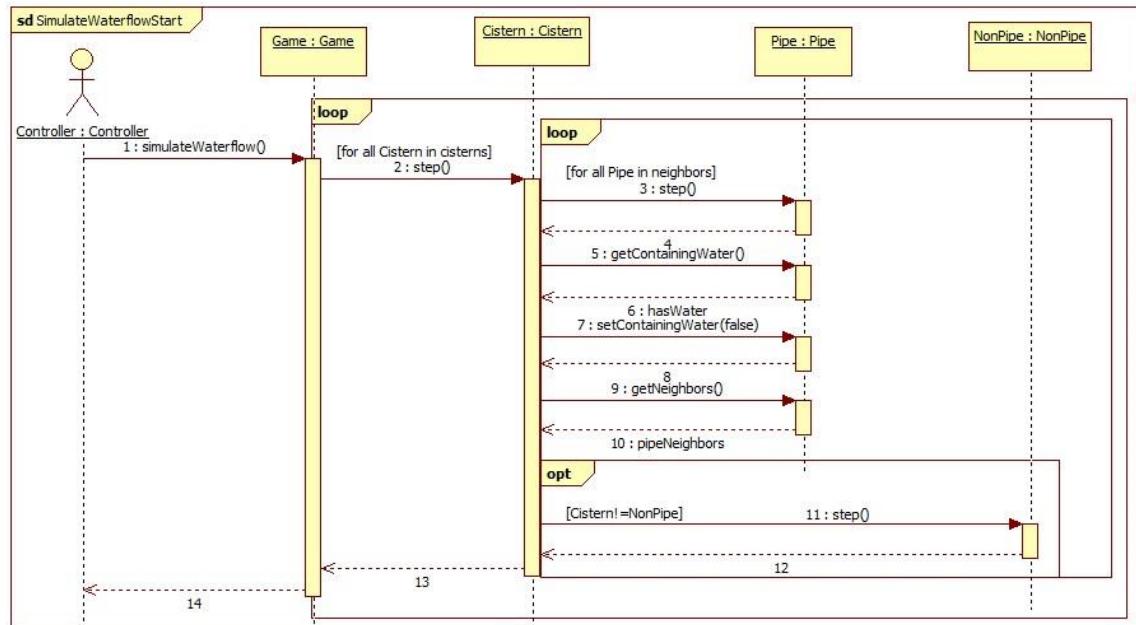
3.4.11 Pipe denying a pipe



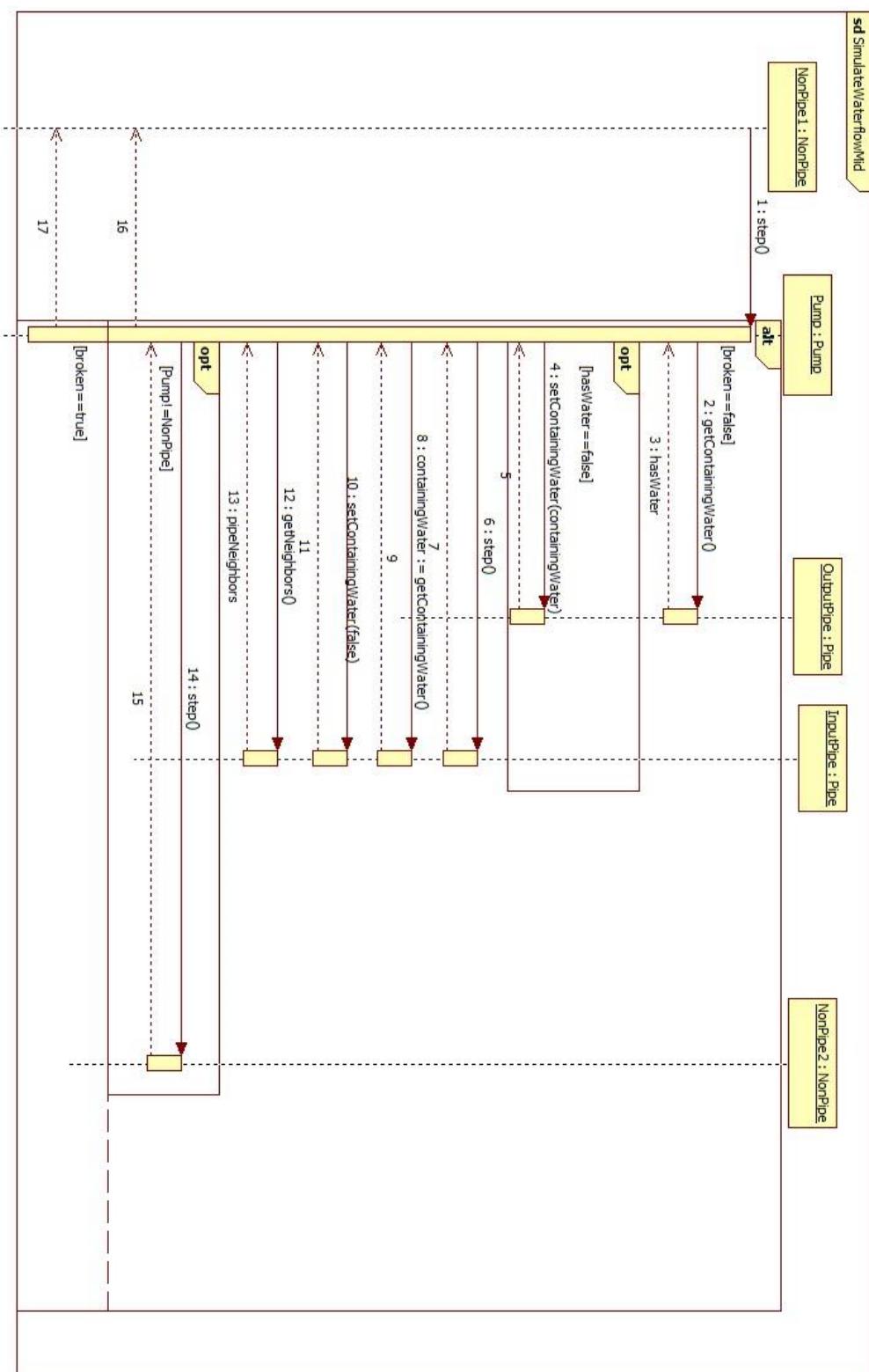
3.4.12. Watersource handling a pipe



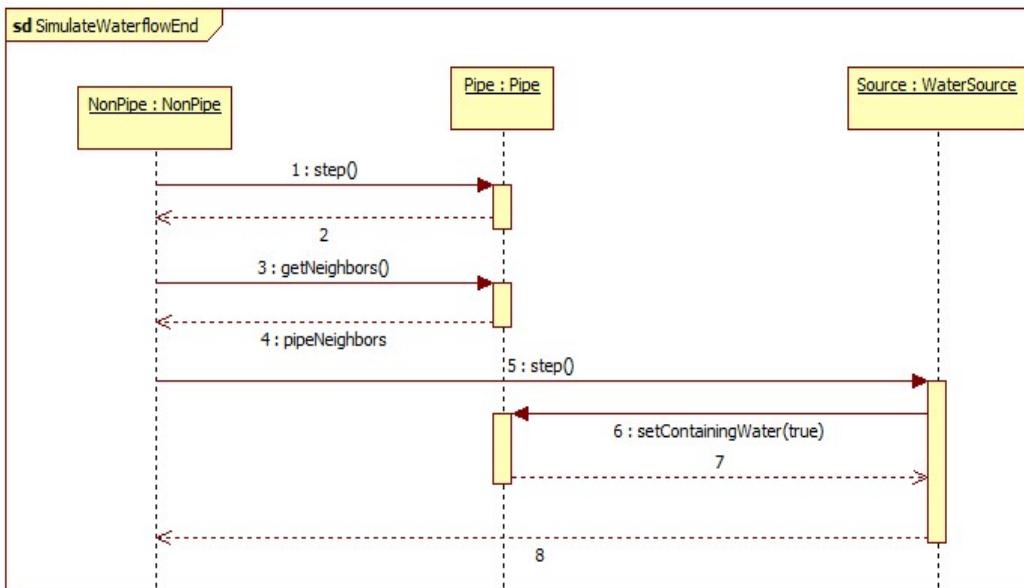
3.4.13. SimulateWaterflowStart



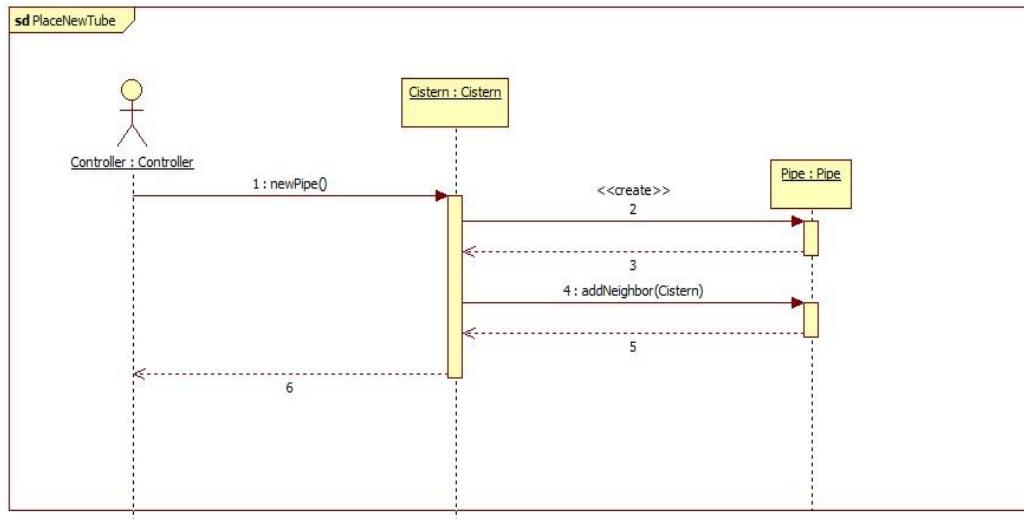
3.4.14 SimulateWaterflowMid



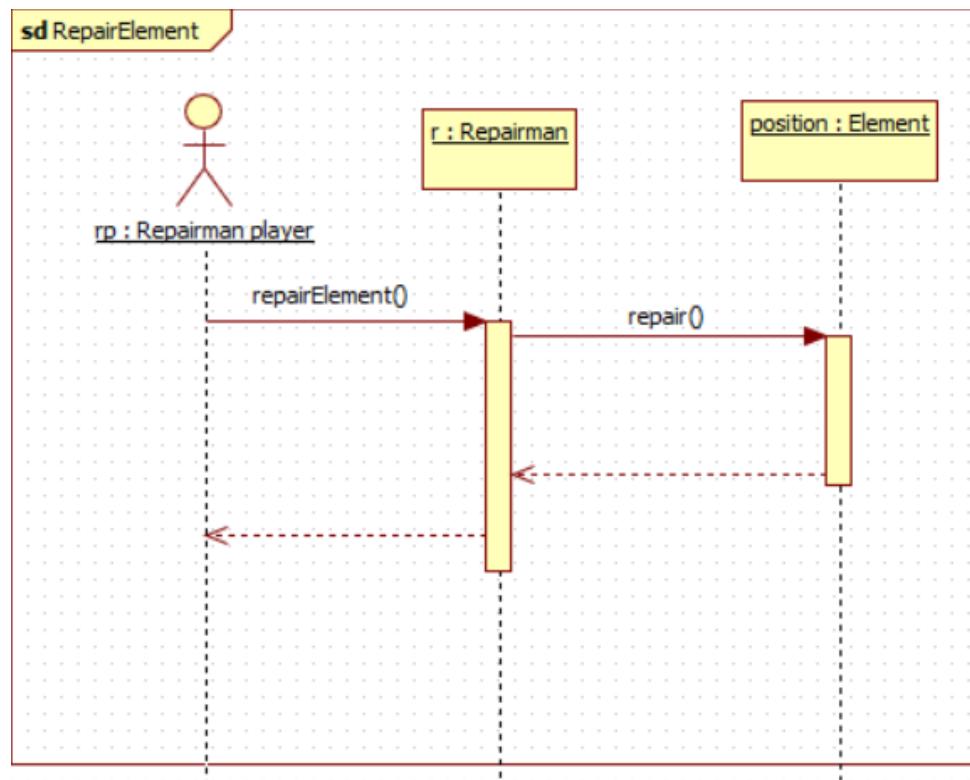
3.4.15 SimulateWaterflowEnd



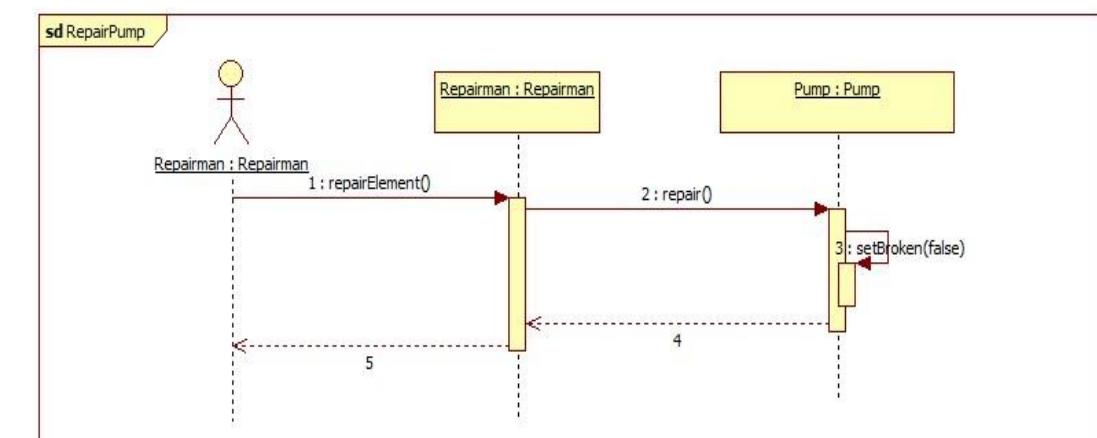
3.4.16 PlaceNewTube



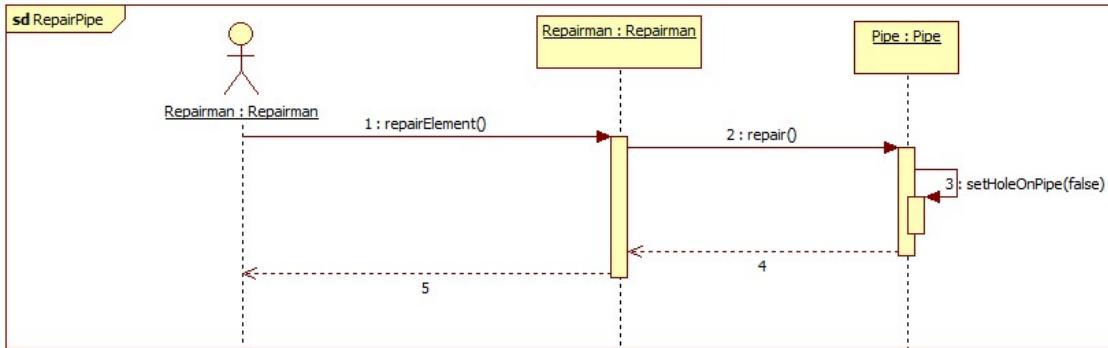
3.4.17 Repair Elements



3.4.18 Repair Pump

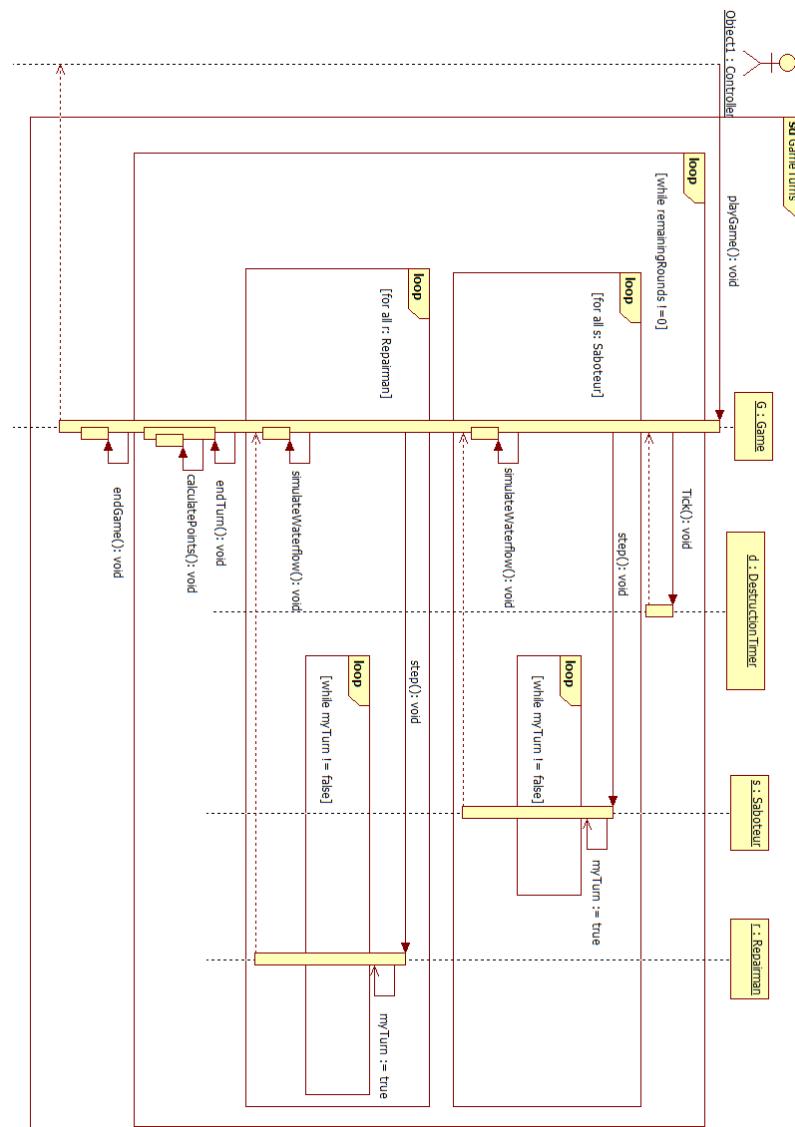


3.4.19. Repair Pipe



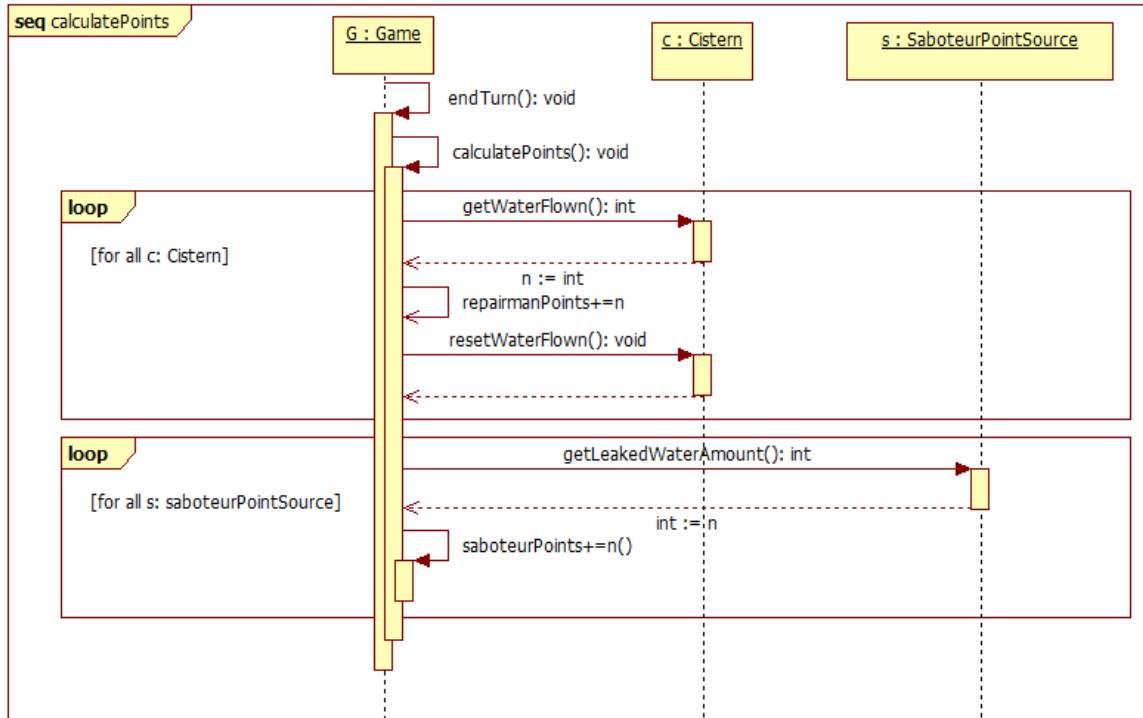
3.4.20. Game Turns

Megjegyzés: A Saboteur és Repairman lépések a 3.4.23 szekvenciadiagramon vannak részletesen bemutatva.

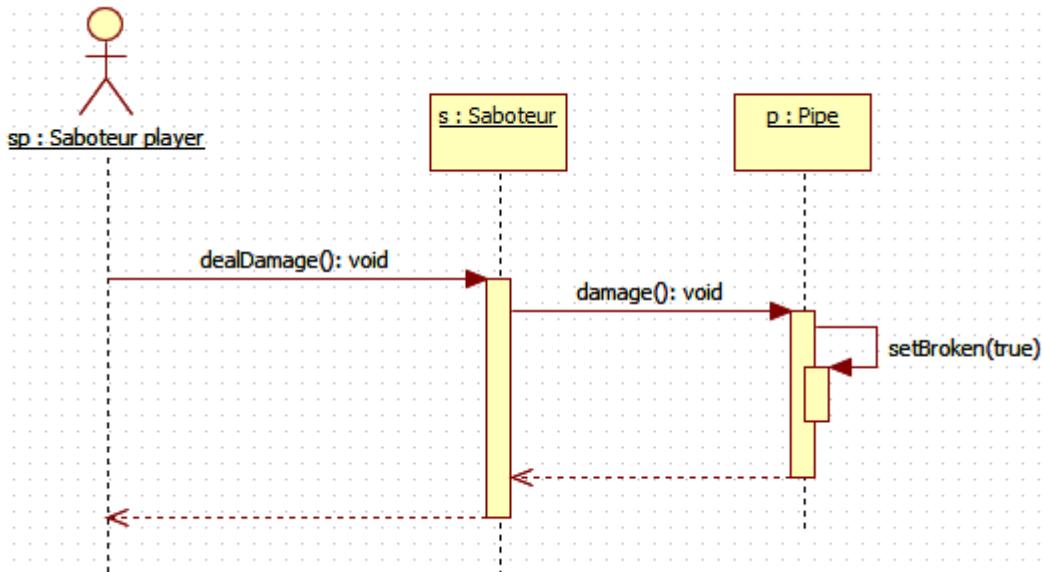


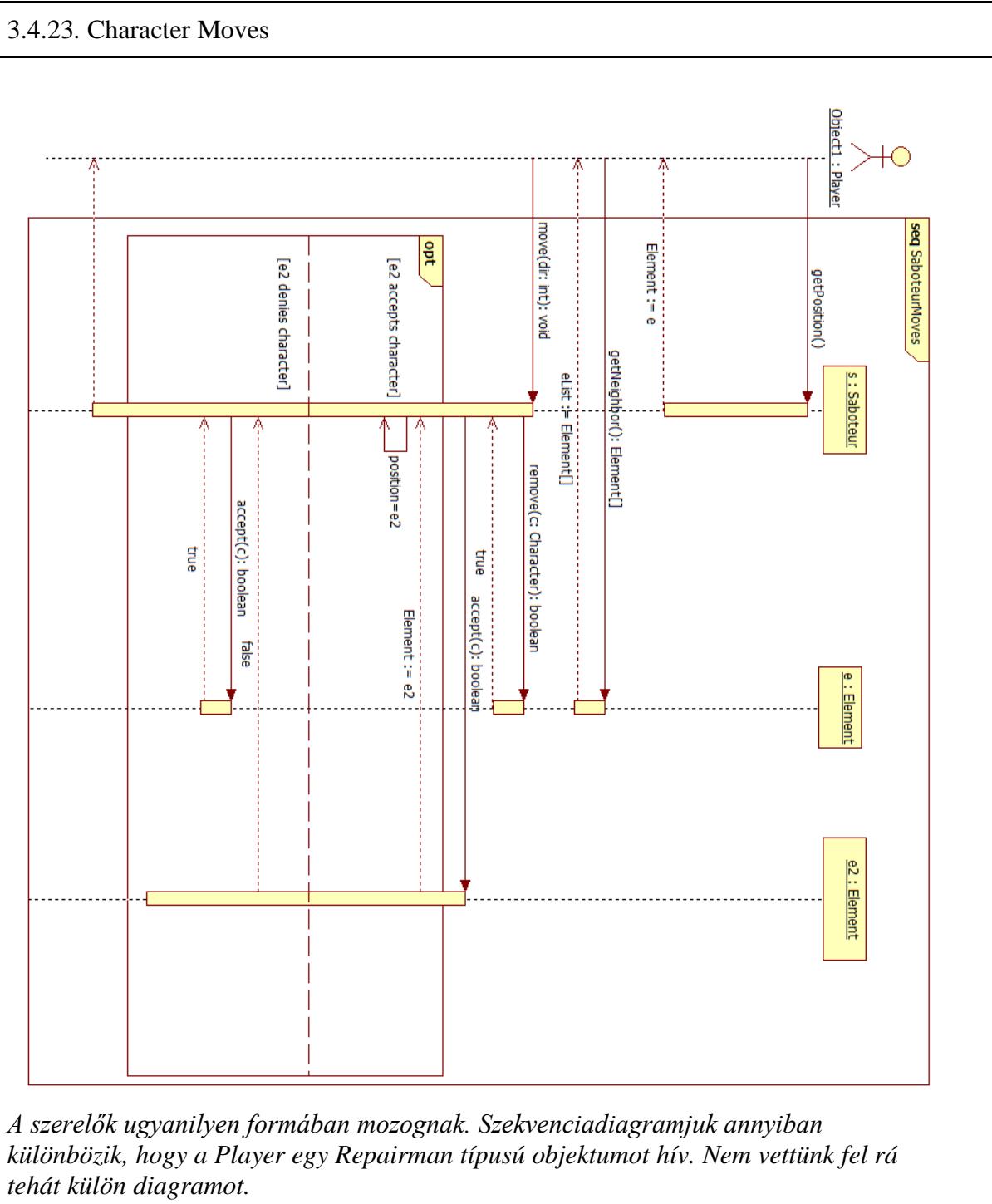
3.4.21 Calculate points at the end of turn

A 3.4.12-es szekvenciadiagram végét, a calculatePoints() metódust mutatja be pontosabban.



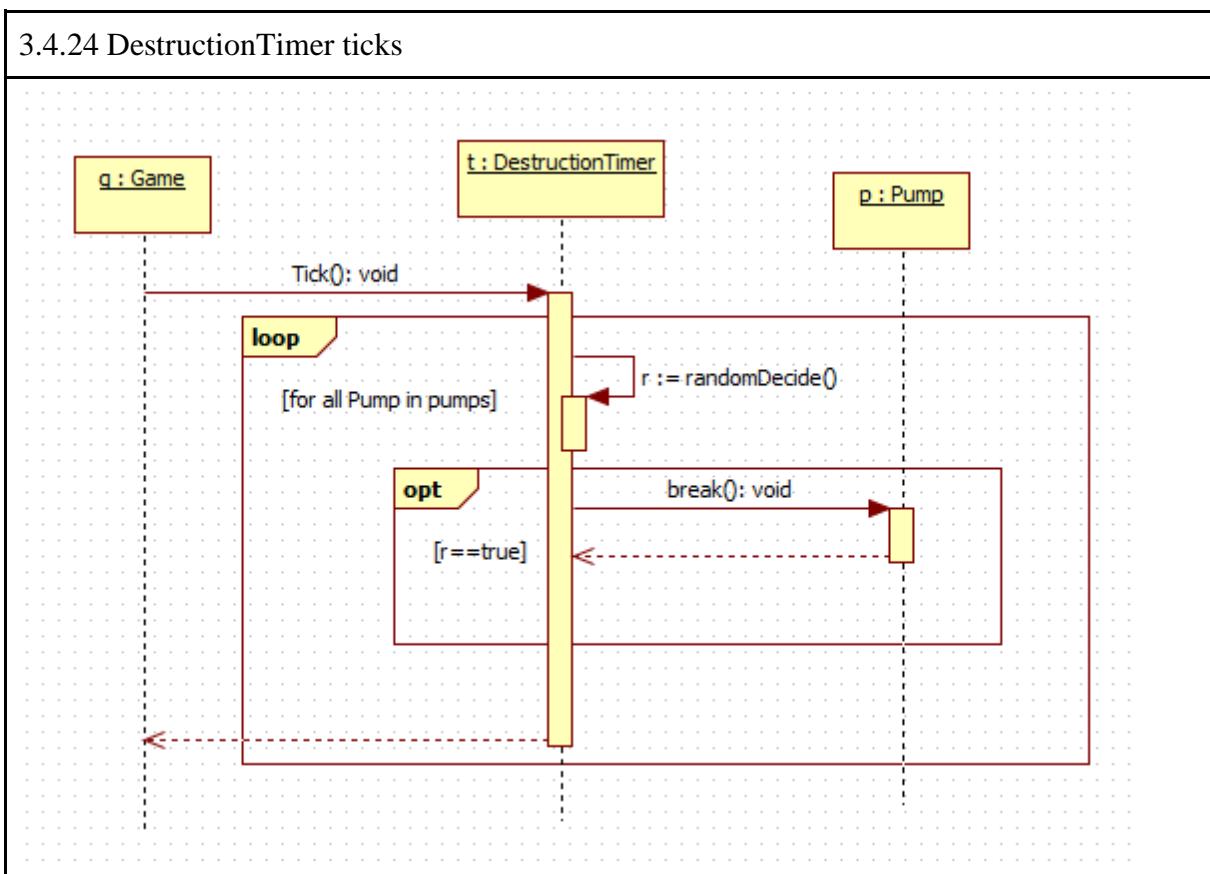
3.4.22. Saboteur Deals Damage





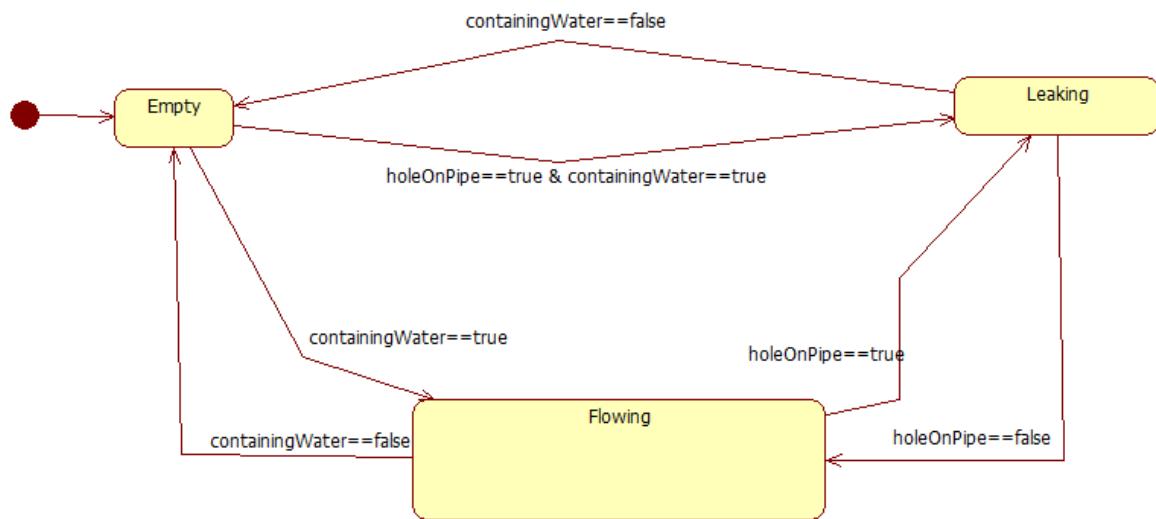
A szerelők ugyanilyen formában mozognak. Szekvenciadiagramjuk annyiban különbözik, hogy a Player egy Repairman típusú objektumot hív. Nem vettünk fel rá tehát külön diagramot.

3.4.24 DestructionTimer ticks

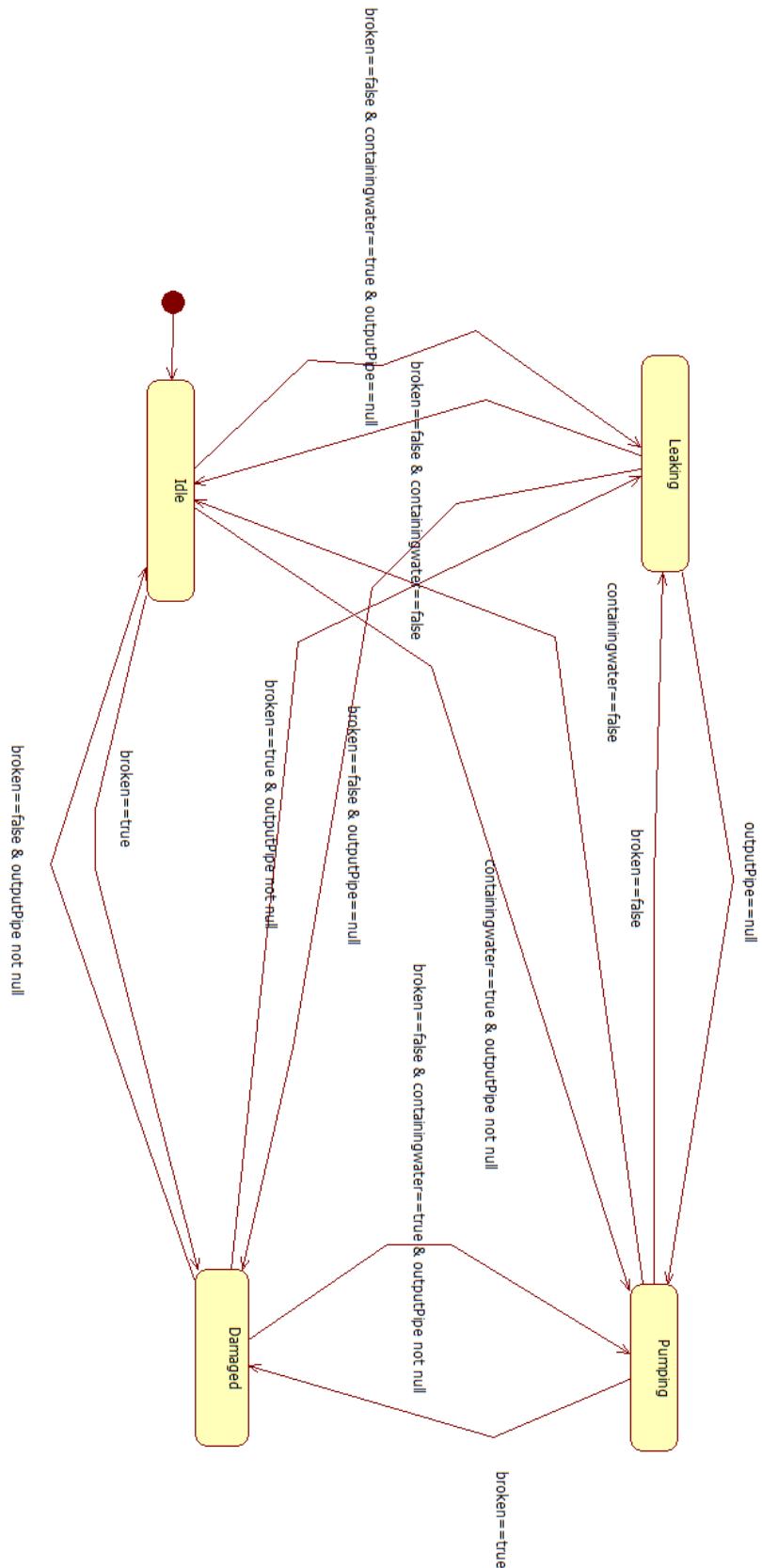


State-chartok

3.4.1 Pipe state-chart



3.4.2 Pump state-chart



3.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2023.03.15 10:00	3 óra	Andai, Bodnár, Gutási, Sziksai, Vajda	<p>Értekezlet. Ötletek az osztályok és azok metodusainak kivitelezéséről. Döntés: Vajda készíti el az osztálydiagramot (3.2).</p> <p>Andai:</p> <ul style="list-style-type: none"> - elkészíti a következő osztályleírásokat: <u>Element, Pipe</u> - elkészíti a következő szekvenciadiagramokat: <u>Lift pump</u> <u>3.4.1, Place pump</u> <u>3.4.2, Lift tube-end</u> <u>3.4.3, Place tube-end</u> <u>3.4.4</u> <p>Bodnár</p> <ul style="list-style-type: none"> - elkészíti a következő osztályleírásokat: <u>Nonpipe, Pump</u> - elkészíti a következő szekvenciadiagramokat: Repair elements, Move repairman, Move saboteur <p>Gutási</p> <ul style="list-style-type: none"> - elkészíti a következő osztályleírásokat: <u>Repairman, Game</u> <p>Sziksai</p> <ul style="list-style-type: none"> - elkészíti a következő osztályleírásokat: <u>Cistern, Watersource</u> - elkészíti a következő szekvenciadiagramokat: Simulate water flow(3.4.5; 3.4.6; 3.4.7), Place new tube(3.4.8) <p>Vajda</p> <ul style="list-style-type: none"> - elkészíti a következő osztályleírásokat: <u>Character, Saboteur</u>
2022.03.15	2 óra	Vajda	Osztálydiagram szerkesztése
2022.03.17 8:00	2 óra	Andai	3.4.1 és 3.4.2 szekvenciadiagramok elkészítése
2023.03.17 10:30	1 óra	Andai, Bodnár, Gutási, Sziksai, Vajda	<p>Értekezlet. Döntés: Sziksai kitalálja a holnapi értekezletig és bemutatja, hogyan oldódjon meg az osztályok tárolása. Vajda kitalálja a holnapi értekezletig és bemutatja, hogyan lehessen szimulálni a vízfolyást.</p> <p>Gutási kitalálja a holnapi értekezletig és bemutatja, hogyan történjen a körökre osztása a játéknak.</p>

2023.03.17 11:30	1 óra	Bodnár	3.3.5 és 3.3.6 osztályleírások elkészítése
2023.03.17. 12:00	2 óra	Andai	3.4.3 és a 3.4.4 szekvenciadiagramok elkészítése, 3.3.1 és 3.3.4 osztályleírások elkészítése
2023.03.17 12:30	2 óra	Szikszai	3.4.5, 3.4.6, 3.4.7, 3.4.8 szekvenciadiagramok elkészítése A 3.3.7 és a 3.3.8 osztályleírások elkészítése
2023.03.17 19:00	2 óra	Vajda	Osztálydiagram javítása, osztályleírások bővítése
2023.03.17 20:30	1 óra	Gutási	State-chart diagrammok elkészítése
2023.03.18 11:30	1 óra	Bodnár	3.1 objektum katalógus elkészítése
2023.03.18 15:00	2 óra	Andai, Bodnár, Gutási, Szikszai, Vajda	Értekezlet. Vajda, Szikszai és Gutási bemutatta a tegnapi értekezleten kiosztott részfeladatok egy lehetséges implementációját. További közös ötletek szekvenciadiagramokkal kapcsolatban. Döntés: Andai, Gutási, Szikszai és Vajda egymás közt felosztva szekvenciadiagramokat készít.
2023.03.18 17:00	3 óra	Szikszai, Gutási	Értekezlet. Döntés: Szikszai további interfészket vesz fel az osztálydiagramban (Place , RepairmanPlace , SaboteurPlace). Továbbá egyéb, az értekezleten megbeszélt metodusokat vesz fel. Gutási dokumentálja az újonnan felvett metodusokat.
2023.03.18 23:00	1 óra	Szikszai	Szekvencia diagramok javítása: 3.4.5, 3.4.6, 3.4.7
2023.03.19 9:30	1 óra	Andai, Gutási, Szikszai, Vajda	Értekezlet. Osztálydiagrammok átszervezése, tervezési döntések megvitatása. Döntés: - Szikszai további interfészket vesz fel az osztálydiagramban (SaboteurPointSource) - Andai, Vajda és Gutási ellenőrzi az osztályleírásokat, és szükség esetén kiegészíti őket.
2023.03.19 10:30	6 óra	Gutási	Place, RepairmanPlace, SaboteurPlace, SaboteurPointSource, Character, Cistern osztályok dokumentációjának bővítése, kiegészítése az előző értekezleteken megbeszélt módosításokkal. 3.4.20, 3.4.21 és 3.4.23 szekvenciadiagramok létrehozása. Dokumentum rendezése, előkészítése nyomtatásra.

2023.03.19 10:30	5 óra	Szikszai	Osztálydiagram bővítése, változások dokumentálása, Szekvencia diagramok javítása (3.4.13, 1.4.14, 3.4.15, 3.4.16)
2023.03.19 11:00	6 óra	Vajda	Osztálydiagram bővítése új osztályok felvétele, osztályleírások kiegészítése, Szekvenciadiagrammok készítése 3.4.22, 3.4.24, Osztályok abc sorrendbe rendezése
2023.03.19 18:00	1 óra	Szikszai	Szekvencia diagramok javítása(3.4.18, 3.4.19, 3.4.5) és dokumentum formázása

4. Analízis modell kidolgozásas

4.1 Objektum katalógus

4.1.1 Ciszterna:

Számoltartja mennyi víz folyik a ciszternába, valamint új pumpákat és a ciszternához kapcsolódó csöveket generál.

4.1.2 Cső:

Számoltartja, hogy az adott cső éppen lyukas-e, mennyi víz folyik ki belőle és azt is, hogy a két vége mihez kapcsolódik.

4.1.3 Forrás:

Minden belekötött csőbe vizet küld, (amint egy csapat befejezte a lépést).

4.1.4 Pumpa:

A pumpa aktuális bemeneti és kimeneti csöveit karbantartja, valamint azt, hogy éppen hibás-e a pumpa.

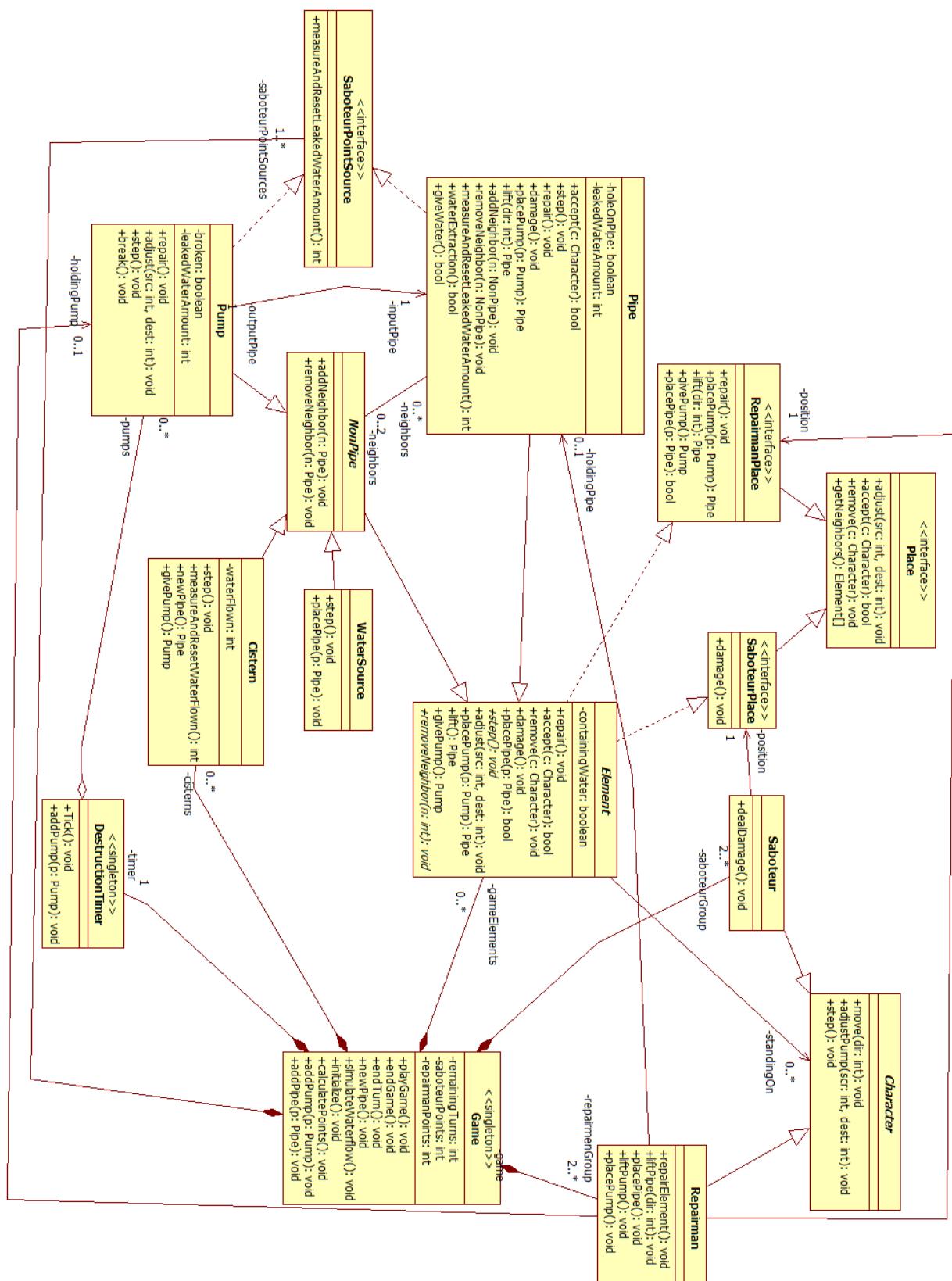
4.1.5 Szabotőr:

A szabotőr karakter speciális képességeit zárja egységbe.
Képes kilyukasztani csöveket.

4.1.6 Szerelő:

A szerelő karakter speciális képességeit zárja egységbe.
Tud javítani csövet és pumpát is, ciszternánál felvenni pumpát és egy cső kettévágását követően elhelyezni.

4.2 Statikus struktúra diagram



4.3 Osztályok leírása

4.3.1 Character

- **Felelősség**

Ebből az osztályból származnak le a játékosok által irányítható karakterek (szerelők, és szabotőrök).

- **Attribútumok**

- **Metódusok**

- **void move(int dir):** Karakter átléptetése egy irányba. Az irány egy szám, ami egyértelműen azonosítja egy elem szomszédait.
- **void adjustPump(int src, int dest):** Pumpa aktív ki- és bemenő élének beállítása (meghívja a Place interfész adjust() metódusát). A két szám egyértelműen azonosít egy-egy csövet, ami a pumpába van kötve.

- **void step():** A Game osztály hívja meg, ezzel kiválasztva a karaktert, hogy ő léphet. Ezen metódus elején állítódik be a myTurn attribútum igazra, ezzel jelezve, hogy a bemenet erre a karakterre irányul. Miután a karakter lépett, a myTurn értéke ismét hamis lesz, és a step() metódus visszatér.

4.3.2 Cistern

- **Felelősség**

A vízhálózat egyik végpontjaként szolgáló víznyelő. Tárolja a belefolyt víz mennyiségét lekérdezésig. Új, belőle félíg szabadba kivezető csövet hoz létre, ha nincs ilyen. A szerelők fel tudnak róla venni pumpát.

- **Ősosztályok**

Element→NonPipe

- **Interfészek**

Megvalósítja a **SaboteurPlace** és **RepairmanPlace** interfészket, ezáltal a **Place** interfész is.

- **Attribútumok**

- **int waterFlown:** A lépés alatt a ciszternába folyt víz mennyisége.

- **Metódusok**

- **Pipe newPipe():** Létrehoz egy új csövet ami félíg szabad és félíg a ciszternába van kötve, ha nem létezik másik ilyen cső.
- **void Step():** Szimulálja a víz folyását.
- **int measureAndResetWaterFlown():** Visszaadja a ciszternába belefolyt víz mennyiségét, és lenullázza a waterFlown-t.
- **Pump givePump():** Létrehoz egy pumpát és visszaadja azt. A karakter kérhet új pumpát a ciszternától.

4.3.3 DestructionTimer

- **Felelősség**

Véletlen eséllyel elrontja a pumpákat.

- **Attribútumok**
- **Pump[] pumps:** Tárolja a játékban szereplő pumpákat.
- **Metódusok**
- **void tick():** Végigitárál az összes pumpán és adott eséllyel elrontja azokat.
- **void addPump(Pump p):** új pumpa hozzáadása a tároltakhoz

4.3.4 Element

- **Felelősség**

Közös absztrakt ōse a pályaelemeknek. Megvalósítja a karakterek számára szükséges függvényeket, azok alap implementációit megvalósítja.

- **Interfészek**

Megvalósítja a **SaboteurPlace** és **RepairmanPlace** interfészket, ezáltal a **Place** interfészt is.

- **Attribútumok**
- **bool containingWater:** Ez tárolja, hogy az adott elemben van-e éppen víz.
- **Character[0..*] standingOn:** Lista a pályaelemen álló játékosokról.
- **Metódusok**
- **void repair():** Ezen a metóduson keresztül lehet őt megjavítani.
- **void adjust(int src, int dest):** Pumba aktuális bemenetét(src) és kimenetét(dest) állítja át sorszámaik alapján.
- **Character[0..*] getstandingOn():** Az elemen álló karakterek lekérdezése.
- **Bool accept(Character c):** Őt kell meghívni ha rá szeretnénk lépni a mezőre. A karakterünket kell átadni paraméterként. Azzal tér vissza, hogy sikeres volt-e a rálépés.
- **void remove(Character c):** Őt kell meghívni ha le szeretnénk lépni a mezőről. A karakterünket kell átadni paraméterként. Azzal tér vissza, hogy sikeres volt-e a lelépés.
- **void damage():** Rajta keresztül lehet megrongálni az elemet.
- **Pump givePump():** Megpróbálunk kérni tőle egy pumpát. A különböző leszármazottak máshogy implementálják majd. Ha tud pumpát adni, akkor NonPipe típusként adja vissza, mert így kompatibilis a szomszédok menedzselésével.
- **Pipe placePump(Pump p):** Megpróbálunk rajta keresztül lerakni egy pumpát. A lerakandó pumpát paraméterben adjuk át neki.
- **Pipe lift(int dir):** Ezen keresztül lehet majd megpróbálni felemelni a kívánt elemet. A paramétere egy irány, ami a meghívott objektum adott irányú szomszédjára utal. Visszatér az adott irányban lévő szomszédos csővel, és lecsatlakoztatja magáról azt.
- **bool placePipe(Pipe p):** Ezen keresztül lehet a csövek végét lehelyezni. Paraméterben kell megadni neki a csövet. Visszatér azzal, hogy sikeres-e.
- **Element[] getNeighbors():** Visszaadja az adott Element szomszédait.
- **void step():** Absztrakt függvény, amely a vízfolyást teszi lehetővé.
- **void removeNeighbor(int n):** Eltávolítja az egyik szomszédot

4.3.5 Game

- **Felelősség**

Kezeli a játék indítását, befejezését és a vízfolyás szimulációját. Továbbá tárolja a csapatokat, benne a játékban szereplő karakterekkel, ezen csapatok pontjait és a játék egyéb elemeit is. Kezeli a víz szimulációját, és a körök felosztását.

- **Attribútumok**
- **Repairman[2..*] repairmanGroup:** A szerelő csapatot alkotó karaktereket tárolja.
- **Saboteur[2..*] saboteurGroup:** A szabotőr csapatot alkotó karaktereket tárolja.
- **int repairmanPoints:** A szerelő csapat pontjait tárolja.
- **int saboteurPoints:** A szabotőr csapat pontjait tárolja.
- **Element[0..*] gameElements:** minden a játékban található elementet tárol
- **Cistern[0..*] cisterns:** Tárolja a játékban szereplő ciszternákat.
- **saboteurPointSource[0..*] saboteurPointSources:** Olyan elemeket tárol, melyekből kaphatnak pontot a szabotőrök.
- **int remainingRounds:** A játékból hátramaradó körök száma. Egy kör több lépésből áll. Először a Game meghívja a DestructionTimer Tick() metódusát. Ezután először az egyik csapat összes karaktere, majd a másik csapat összes karaktere lép. minden karakter lépése után szimulálódik a vízfolyás, és hozzáadódnak a pontok a csapatok számára.
- **DestructionTimer timer:** időzítő amely a pumpák véletlen elromlásáért felelős

- **Metódusok**

• **void initialize():** Játék inicializálása. Betölti az előre megtervezett kiinduló pályát. Létrehozza az ezt a pályát reprezentáló objektumokat.

- **void playGame():** A játék fő metódusa. Addig tart, ameddig van következő kör.
- **void endGame():** Játék befejezése, kiírja a győztes csapatot.
- **void endTurn():** Adott kör vége, vízfolyás következik
- **void SimulateWaterflow():** Vízfolyás szimulálása a kör végén.
- **void addElement(Element e):** Objektum hozzáadása a pályához.
- **Element[] getGameElements():** A játékban szereplő elemek lekérdezése.
- **void addSaboteur():** Létrehoz és hozzáad egy Szabotőrt a játékhoz.
- **void addRepairman():** Létrehoz és hozzáad egy Szerelőt a játékhoz.
- **void addPipe(p: Pipe):** Hozzáad egy csövet a játékhoz.
- **void calculatePoints():** A két csapat pontszámainak növelése.
- **void addPump(p:Pump):** Hozzáad egy pumpát a játékhoz.
- **void newPipe():** Létrehoz egy új csövet a ciszternánál.

4.3.6 NonPipe

- **Felelősség**

Absztrakt osztály, amely magában foglalja minden csőtől eltérő helyeket ahova léphet a játékos.

Fő feladata karbantartani/tárolni az aktuális szomszédokat, tehát azon csöveket amelyek kapcsolódnak az adott nemesőhöz.

A program későbbi bővíthetőségét könnyíti.

- **Ősosztályok**

Element

- **Interfészek**

Megvalósítja a **SaboteurPlace** és **RepairmanPlace** interfészeket, ezáltal a **Place** interfészt is.

- **Attribútumok**

- **Pipe[] neighbors** A hozzá csatlakozó csöveket tárolja.addnei

- **Metódusok**

- **void addNeighbor(Pipe _pipe):**

Cső csatlakoztatásakor bővíti a neighbors attribútumot az új csővel.

- **void removeNeighbor(Pipe _pipe):**

Cső lecsatlakoztatásakor törli az adott csövet a neighbors-ból.

- **Pipe[] getNeighbors():**

Visszaadja az összes szomszédos csövet.

- **Pipe getNeighbor(itn dir):** Visszaad egy adott irányban levő szomszédos csövet.

4.3.7 Pipe

- **Felelősség**

Csőként funkcionál, lyukaszható, foltozható, összecsatlakoztatható más nem csövekkel.

Nyilvántartja a nem cső szomszédjait. Egyszerre egy játékos állhat rajta ezt számon tartja.

Számolja a rajta kifolyó víz mennyiségét. Ezt le lehet tőle kérdezni. Engedi, hogy rakjanak bele vizet, illetve hogy kivegyenek belőle.

- **Ősosztályok**

Element

- **Interfészek**

Megvalósítja a **SaboteurPlace** és **RepairmanPlace** interfészeket, ezáltal a **Place** interfészt is.

Megvalósítja a **SaboteurPointSource** interfészt.

- **Attribútumok**

- **NonPipe[] neighbors:** a szomszédait tároló lista

- **bool holeOnPipe:** van-e rajta lyuk

- **int leakedWaterAmount:** számolja a rajta kifolyó víz mennyiségét

- **Metódusok**
- **int measureAndResetLeakedWaterAmount():** Visszaadja, a leakedWaterAmountot, és lekérdezés után visszaállítja azt nullára.
- **bool waterExtraction():** Kiszívja az adott csőről a vizet és visszatér, hogy volt-e benne.
- **bool giveWater():** Vizet próbál tenni a csőbe, visszaadja, hogy sikerült-e.
- **bool accept(Character c):** Felüldefiniálja az Element Accept metódusát.
- **void step():** Ha lyukas, kifolyatja magából a vizet és növeli a LeakedWaterAmountot
- **void repair():** holeOnPipe-ot false-ra állítja.
- **void damage():** holeOnPipe-ot true-ra állítja.
- **void removeNeighbor(NonPipe n):** A paraméterül kapott irányú szomszédját eltávolítja.
- **void addNeighbor(NonPipe n):** Cső szomszédjának beállítása.
- **Pipe getNeighbor(int dir):** Visszaad egy adott irányban levő szomszédos csövet.
- **Pipe lift(int dir):** Olyan cső felemelésénél használatos, amelyiknek az egyik fele nincs sehol sem beköve. Ekkor a cső egyik végének a felemeléséhez rajta kell állni ezen a csövön. A paramétert nem használjuk semmire, csak egyformának kell lennie azzal, amit felülír.
- **Pipe placePump(Pump p):** Létrehoz egy új csövet, majd közé és a meglévő cső közé a nála lévő pumpát helyezi el. Visszaadja a új csöv et.

4.3.8 Place

- **Felelősség**

Interfész, ami azokat a metódusokat tartalmazza, amelyeket minden karakter elérhet a játék összes elemén.

- **Metódusok**
- **void adjust(src: int, dest: int):** Pumpa aktuális bemenetét(src) és kimenetét(dest) állítja át sorszámaik alapján.
- **bool accept(c: Character):** Karakter felhelyezése az adott elemre. Ha az elem elfogadja a karaktert, igaz értékkal tér vissza. Ellenkező esetben hamis értékkel tér vissza.
- **void remove(c: Character):** Karakter levétele az adott elemről. Ha az elem sikeresen végrehajtja a karakter levételét, igaz értékkal tér vissza. Ellenkező esetben hamis értékkel tér vissza.
- **Element[] getNeighbors():** Visszaadja az adott elem szomszédjait.

4.3.9 Pump

- **Felelősség**

A pumpának mint aktív elem, karbantartja a bemenetét és kimenetét.
Azaz melyik csőből kap vizet és melyikbe pumpálja tovább.

- **Ősosztályok**

Element → Nonpipe

- **Interfészek**

Megvalósítja a **SaboteurPlace** és **RepairmanPlace** interfészket, ezáltal a **Place** interfészt is.
Megvalósítja a **SaboteurPointSource** interfészt.

- **Attribútumok**
- **bool broken** Tud-e vizet pumpálni az adott pumpa
- **Pipe inputPipe** Az a cső ahonnan szívja a vizet.
- **Pipe outputPipe** Az a cső ahova pumpálja a vizet.

- **Metódusok**
- **Pipe getInputPipe():**
Visszaadja az inputPipe attribútumot, melyik csőből kapja aktuálisan a vizet.
- **void setInputPipe(Pipe _pipe):**
Beállítja inputPipe attribútumot a paraméterként megadott Pipe-ra.
- **Pipe getOutputPipe(Pipe _pipe):**
Visszaadja az outputPipe attribútumot, melyik csőbe pumpálja éppen a vizet.
- **void setOutputPipe(Pipe _pipe):**
Beállítja outputPipe attribútumot a paraméterként megadott Pipe-ra.
- **void adjust(int src, int dest):**
Pumpa aktuális bemenetét(src) és kimenetét(dest) állítja át sorszámaik alapján.
- **void repair():** A broken attribútumot false-ra állítja. Ez akkor hívódik amikor egy elromlott pumpát megjavít egy szerelő.
- **void break():** A broken attribútumot true-ra állítja.
- **void step():** A vízfolyást szimulálja

4.3.10 Repairman

- **Felelősség**
Játékos által irányított szerelő karakter. Célja, hogy minél több víz jusson el a forrásuktól a ciszternákba.

- **Ősosztályok**
Character

- **Attribútumok**
- **RepairmanPlace position:** Ahol éppen a Szerelő áll.
- **Pipe holdingPipe:** A felvehető csővég tárolására szolgál.
- **Pump holdingPump:** A felvehető Pumpa tárolására szolgál.
- **Game game:** Tárolja a játék egy referenciáját, mely az új Csövek és Pumpák eltárolásához szükséges.

- **Metódusok**
- **void RepairElement():** Megjavítja azt az Element játékelementet, amin áll.
Megtörténhet, hogy ezt egy olyan elemen állva végzi, ami nem tud elromlani/megjavulni. Ekkor nem történik semmi.
- **void LiftPipe(int dir):** Ha a játékos nem csövön áll, akkor az adott elementnek amin áll a paraméterben átadott számú szomszédját magához veszi
- **void LiftPump():** Ha a játékos egy ciszternán áll, akkor magához vehet egy új pumpát.
- **void PlacePump():** Ha a játékos egy csövön áll, lehelyezheti a nála levő pumpát.
- **void PlacePipe():** Ha a játékos épp nem egy csövön áll, akkor lehelyezheti a nála lévő csövet.
- **Pump getHoldingPump():** Visszaadja a karakternél levő pumpát.

- **void setHoldingPump(Pump p):** Beállítja a karakternél levő pumpát.
- **Pipe getHoldingPipe():** Visszaadja a karakternél levő csövet.
- **void setHoldingPipe(Pipe p):** Beállítja a karakternél levő csövet.

4.3.11 RepairmanPlace

- **Felelősség**

Interfész, ami azokat a metódusokat tartalmazza, amelyeket a szerelők elérhetnek a játék elemein.

- **Interfészek**

- Megvalósítja a **Place** interfészt.

- **Ősosztályok**

Place interfész

- **Metódusok**

- **void repair():** A karakter hívja ezt a metódust. Hatására a pályaelem megjavul.
- **Pipe placePump(Pump p):** Elhelyezi az adott pumpát egy csövön, lényegében kettévágva azt.
- **Pipe lift(int dir):** Az elem egy adott számmal azonosított csövének lecsatlakoztatása. Visszaadja a játékosnak a csövet, amit lecsatlakoztatott. A cső ezen elembe mutató végét nullára állítja

• **Pump givePump():** Létrehoz egy új pumpát, amit utána odaad a játékosnak. Ez ciszternákon működik.

• **bool PlacePipe(Pipe p):** Beköti a paraméterként kapott csövet az interfészt megvalósító elembe. Visszatér azzal, hogy sikeres-e.

4.3.12 Saboteur

- **Felelősség**

A játékosok által irányított szabotőrök, céljuk hogy minél több víz folyjon el a sivatagba a játék során.

- **Ősosztályok**

Character

- **Attribútumok**

• **SaboteurPlace position:** Elem, amin a szabotör éppen áll.

- **Metódusok**

• **void dealDamage():** Kilyukasztja az adott csövet amin áll amennyiben az még nem volt lyukas

4.3.13 SaboteurPlace

- **Felelősség**

Interfész, ami azokat a metódusokat tartalmazza, amelyeket a szabotőrök elérhetnek a játék elemein.

- **Ősosztályok**

Place interfész

- **Metódusok**

• **void damage():** Megrongálja az adott elemet. Cső esetén kilyukasztja azt, egyéb elemeken nem hajt végre semmit.

4.3.14 SaboteurPointSource

- **Felelősség**

Interfész, ami azokat a metódusokat tartalmazza, amelyek a szabotőrök pontjainak meghatározásához szükséges..

- **Metódusok**

• **int measureAndResetLeakedWaterAmount():** Visszaadja, mennyi víz folyt el az adott szimulációs körben és lekérdezés után visszaállítja azt nullára.

4.3.15 WaterSource

- **Felelősség**

Korlátlan vízforrás, tőle indul a víz áramlása.

- **Ősosztályok**

Element→NonPipe

- **Interfészek**

Megvalósítja a **SaboteurPlace** és **RepairmanPlace** interfészket, ezáltal a **Place** interfészt is.

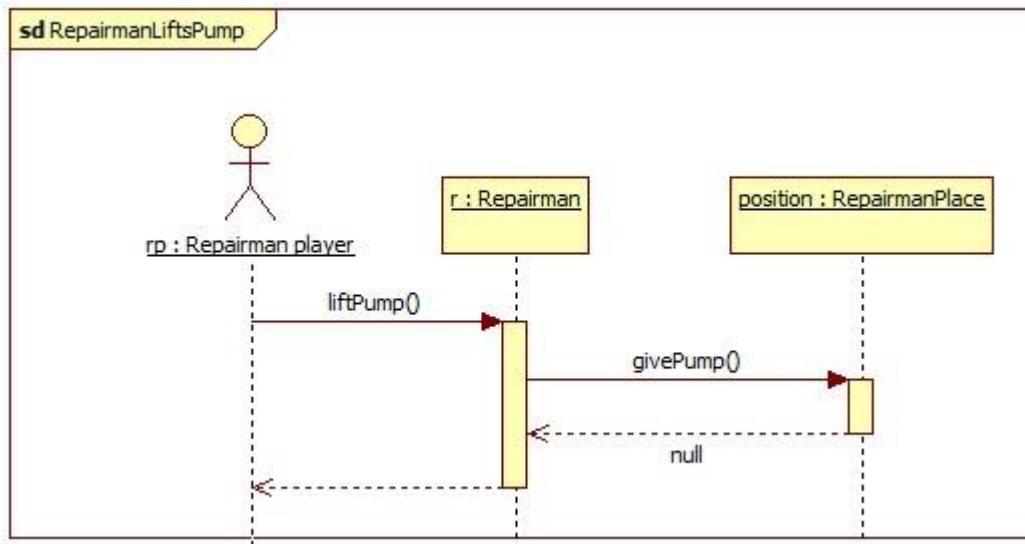
- **Metódusok**

- **void step()**: Elindítja a víz áramlását a Ciszternából, minden bele csatlakozó csövet megtölt vízzel, ha van szabad kapacitása.

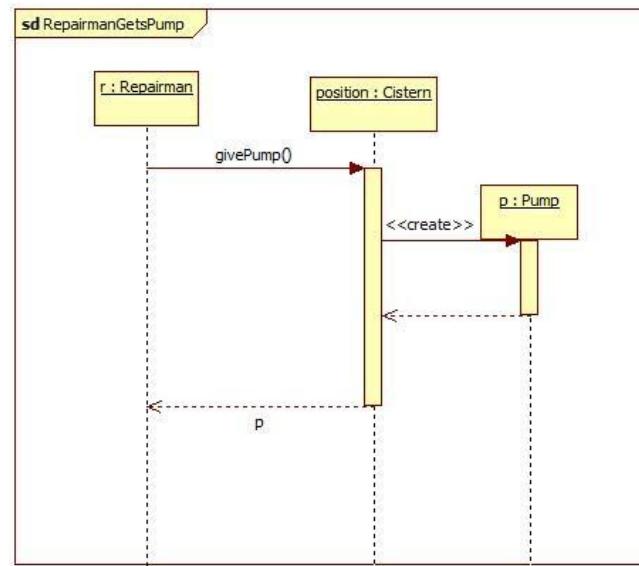
- **bool placePipe(p: Pipe)**: Amennyiben a paraméterben kapott cső egy másik forrásból származik akkor nem fogadja el, egyéb esetben igen. Visszatér azzal, hogy sikeres-e.

4.4 Szekvencia diagramok

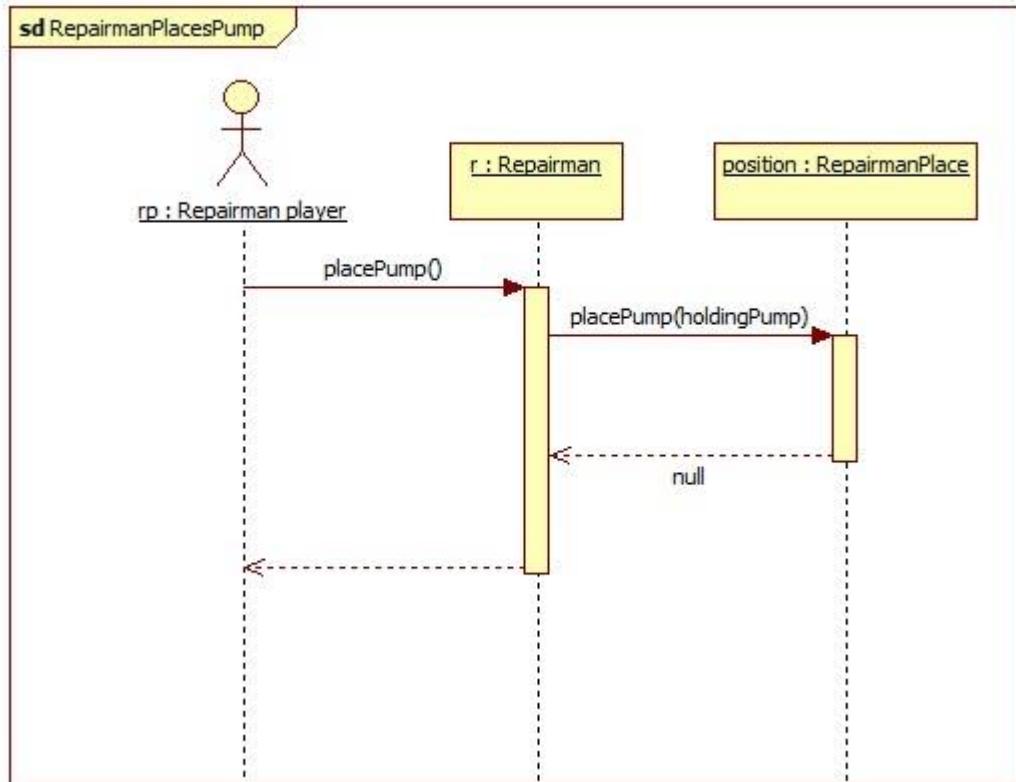
4.4.1 RepairmanLiftPump

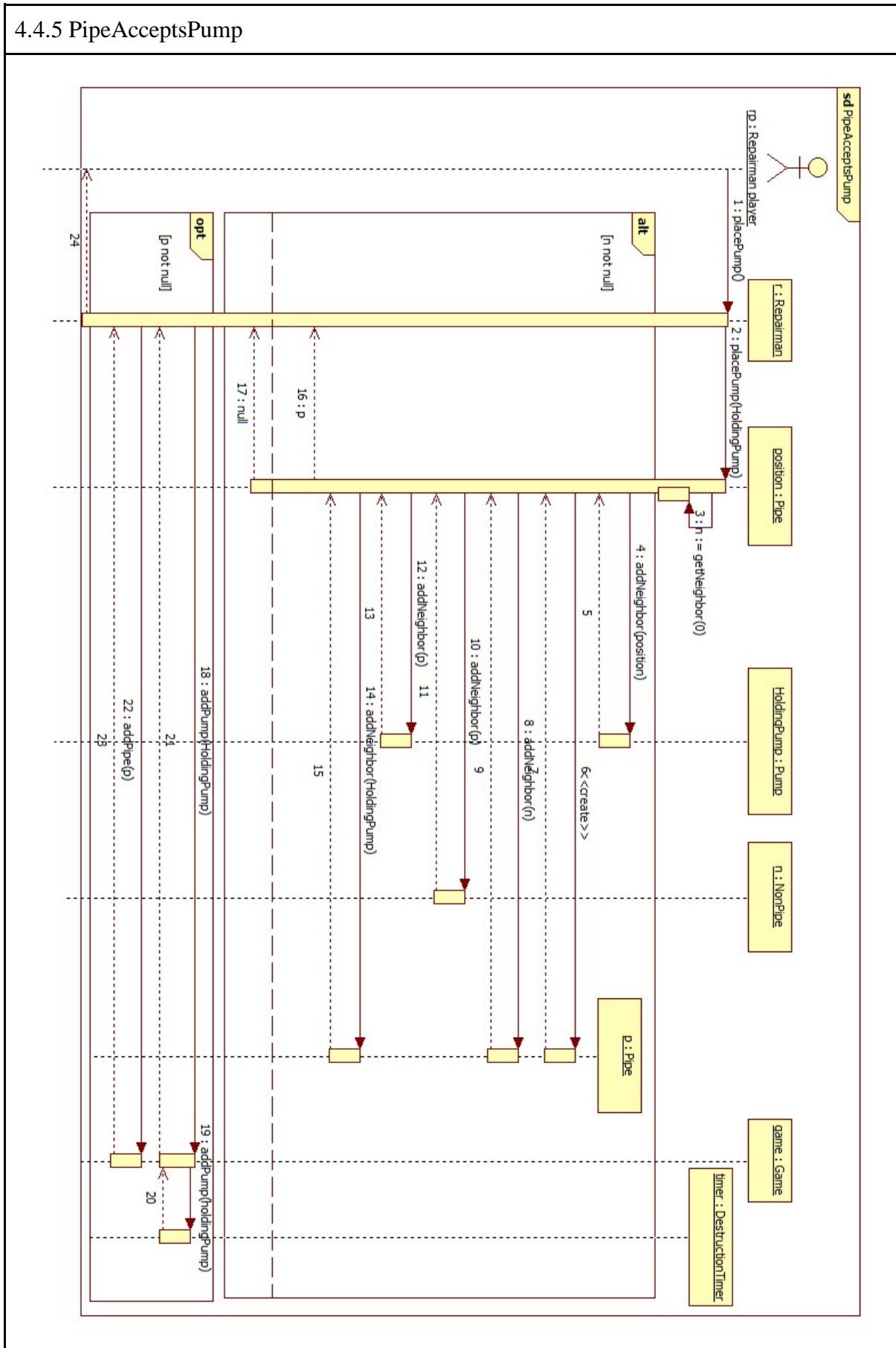


4.4.2 Repairman recieves pump

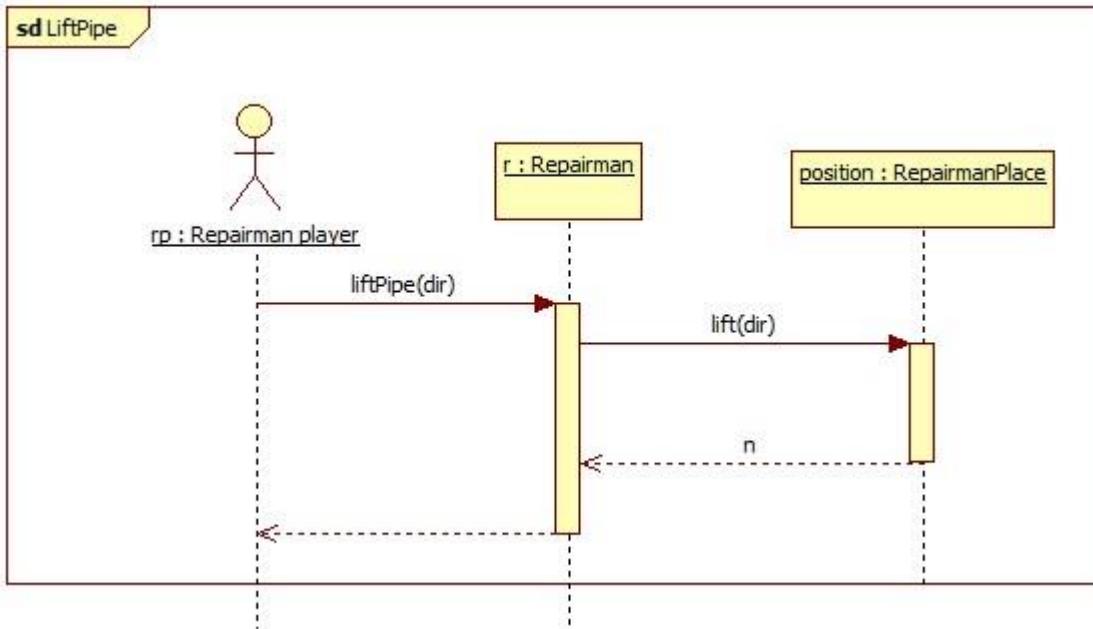


4.4.4 RepairmanPlacesPump

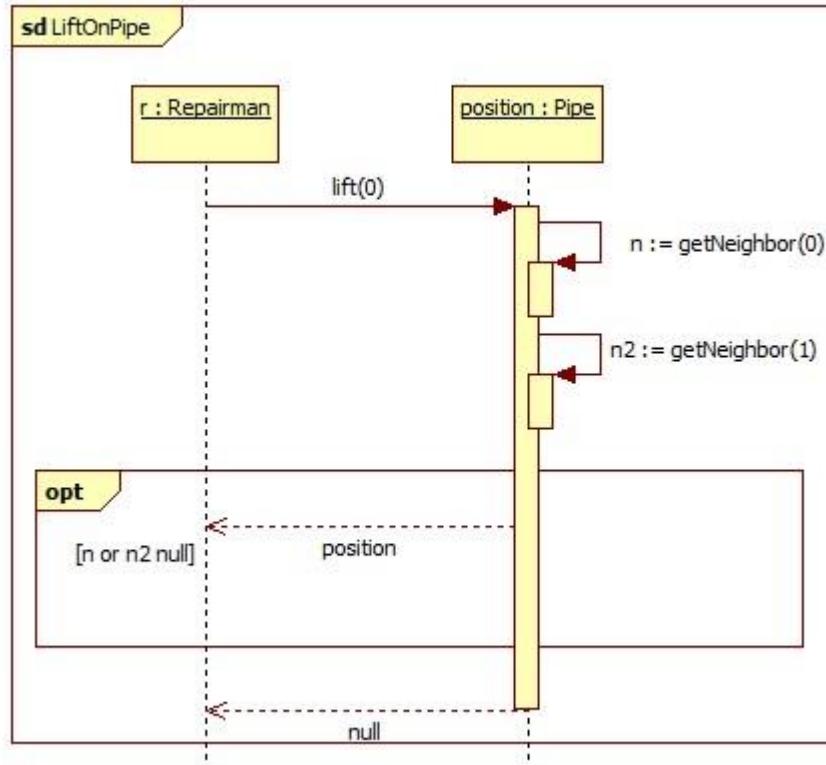




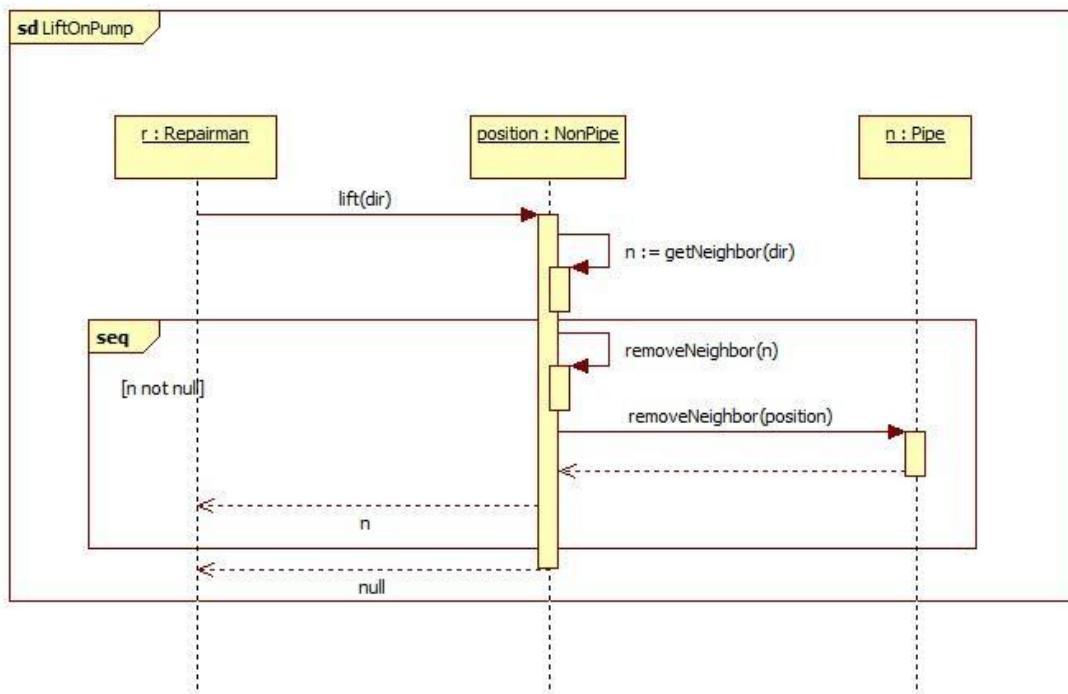
4.4.6 RepairmanLiftsPipe



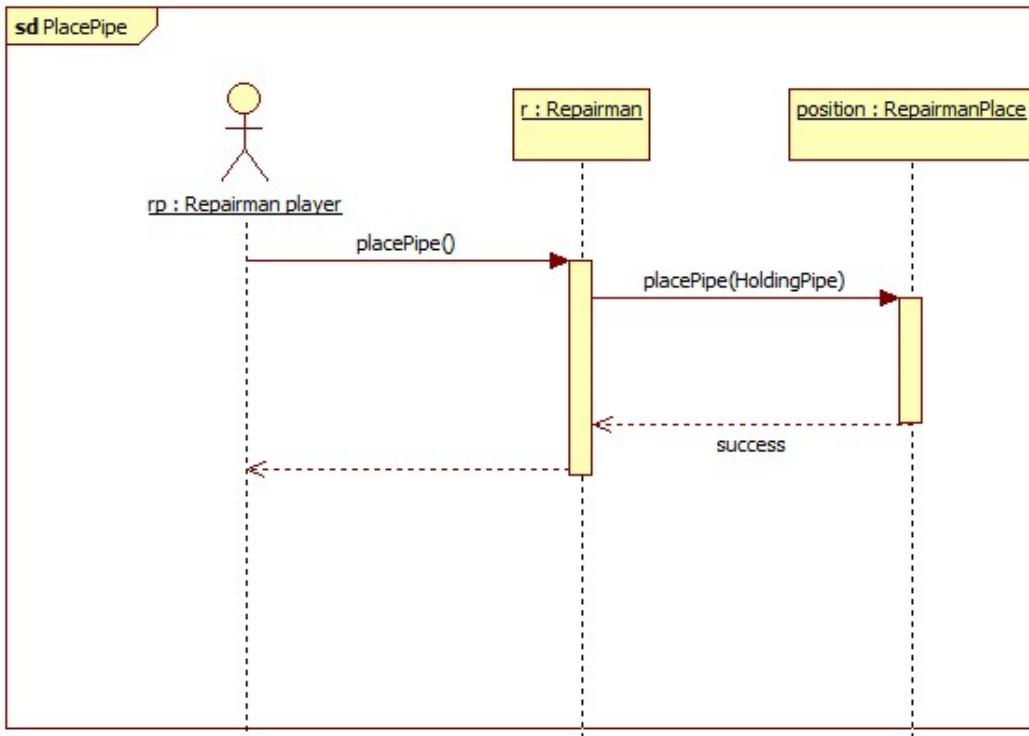
4.4.7 Lift Pipe, character standing on pipe



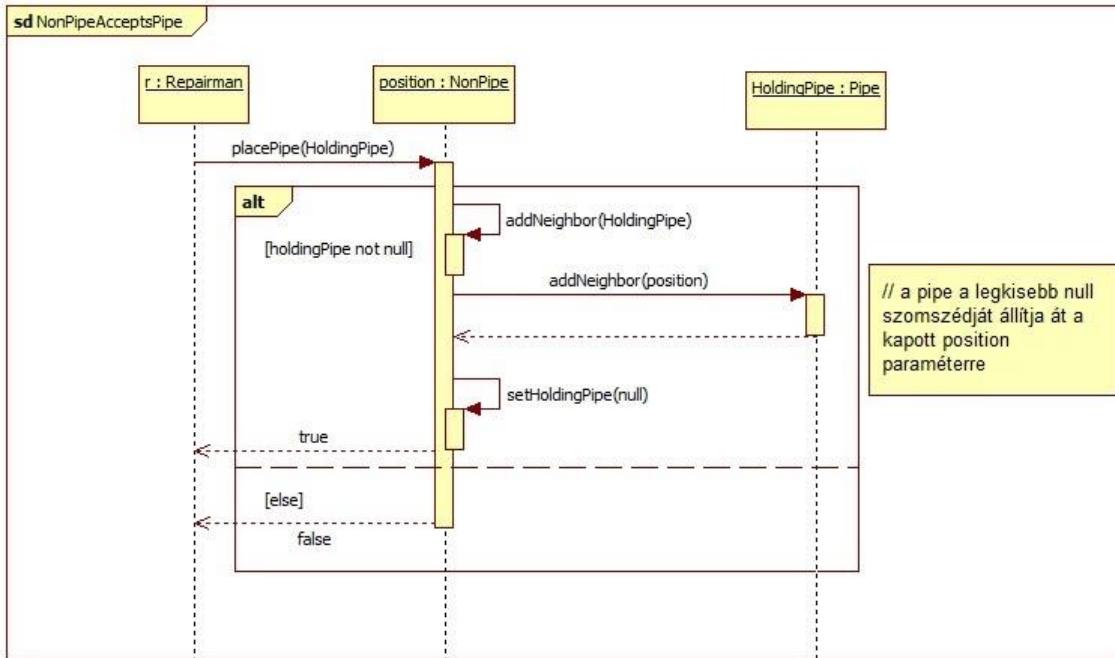
4.4.8 Lift Pipe, character standing on pump



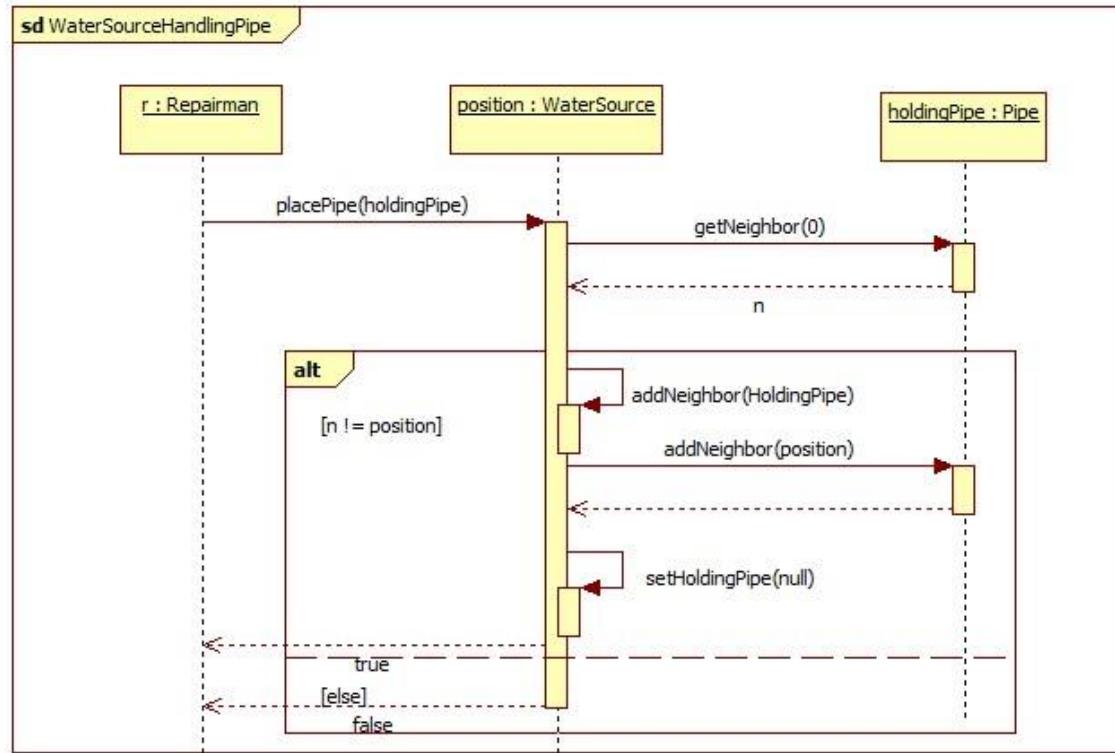
4.4.9 Repairman placing pipe.



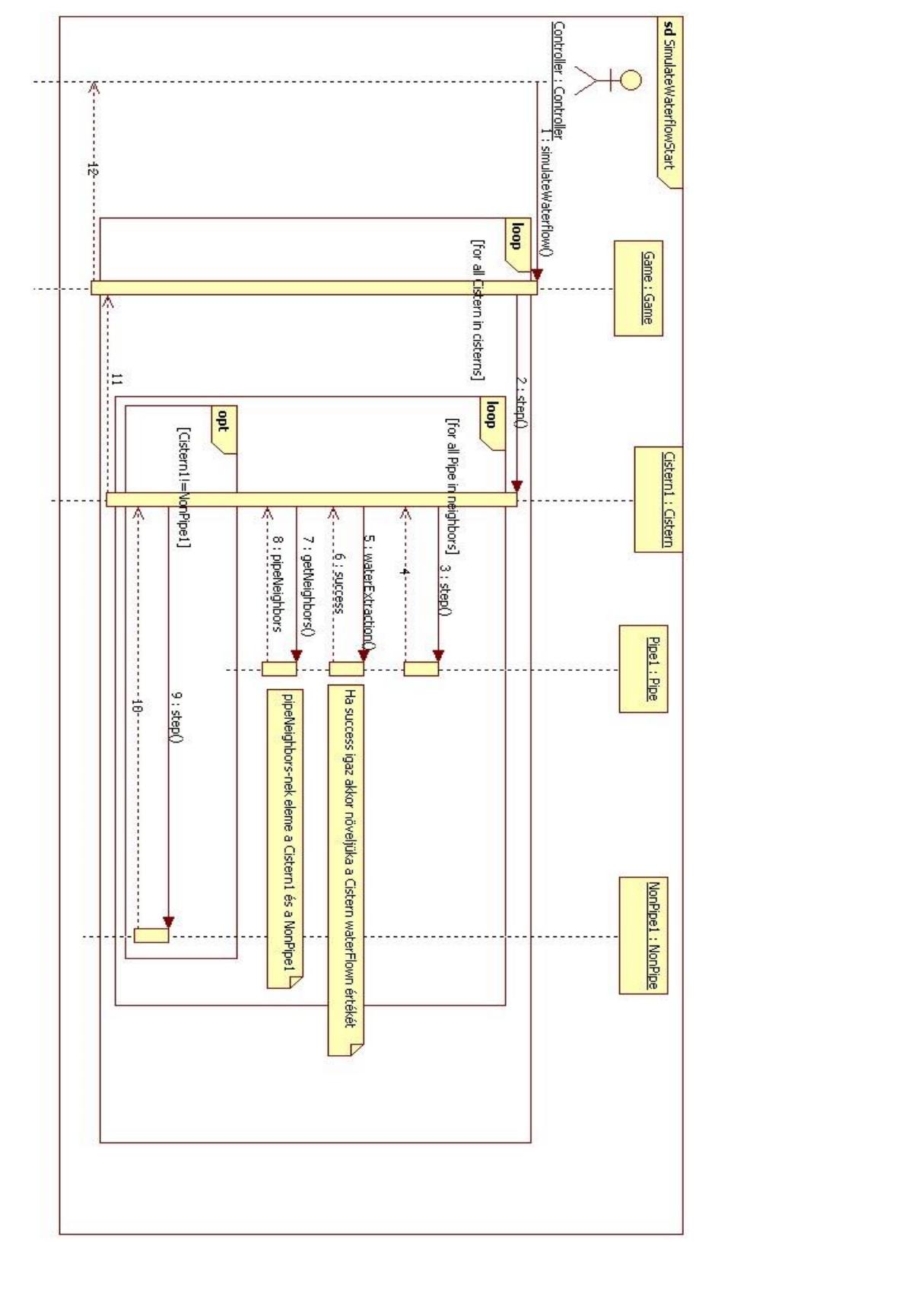
4.4.10. Nonpipe accepting a pipe.



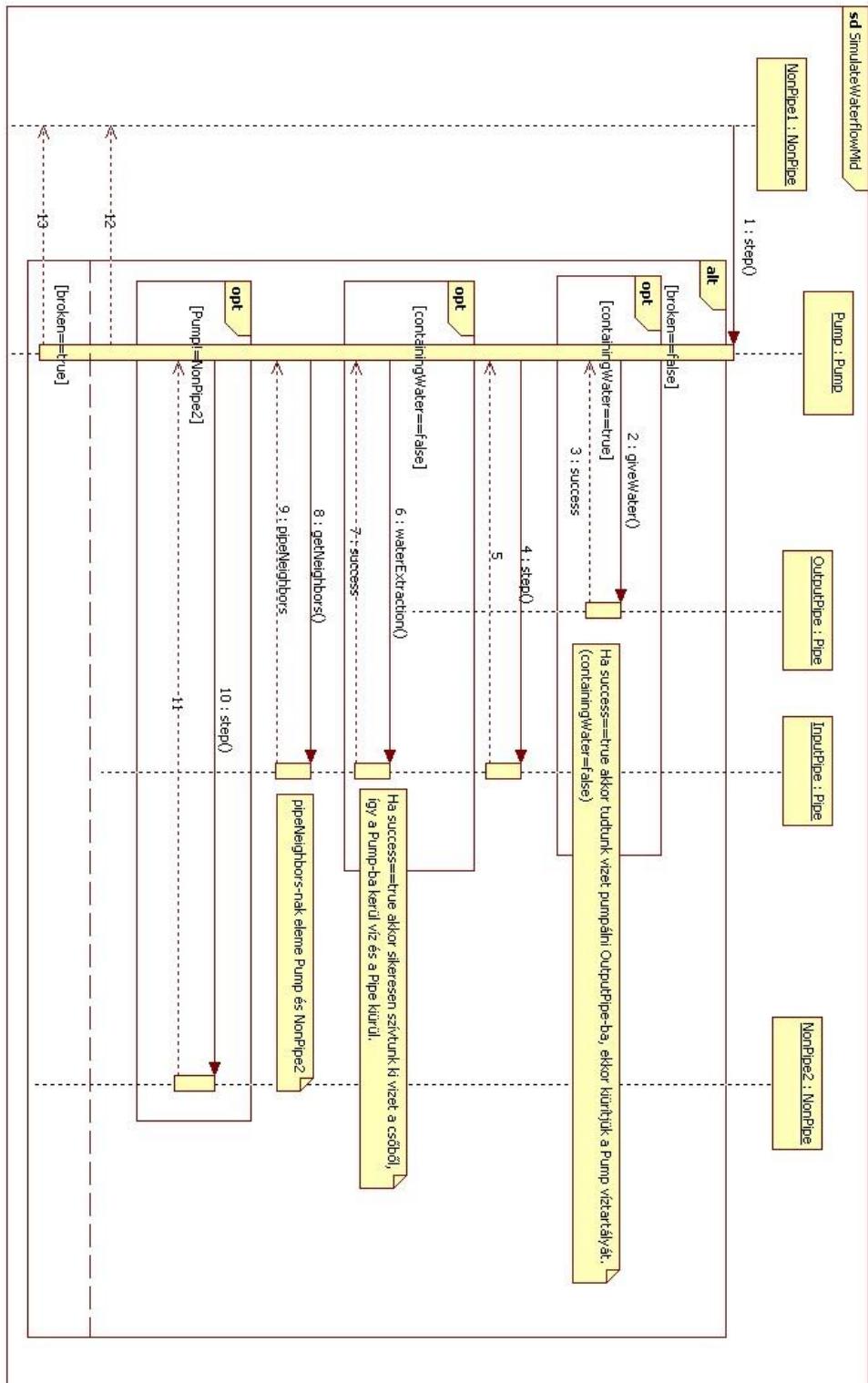
4.4.12. Watersource handling a pipe



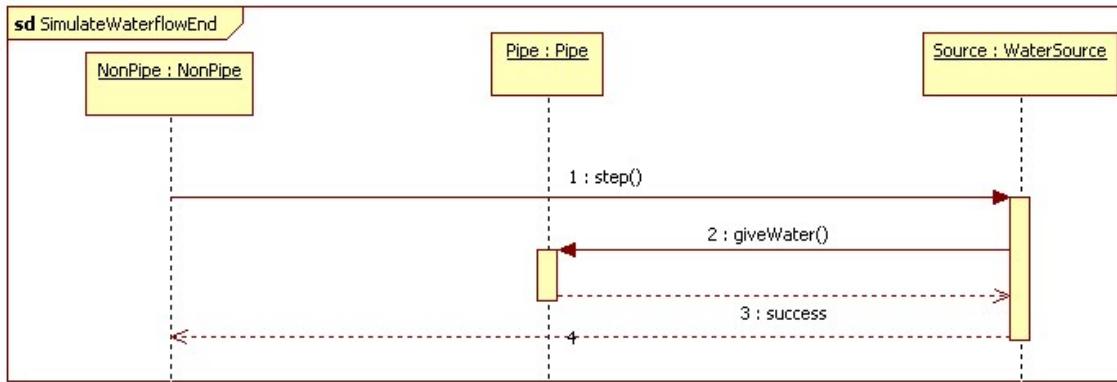
4.4.13. SimulateWaterflowStart



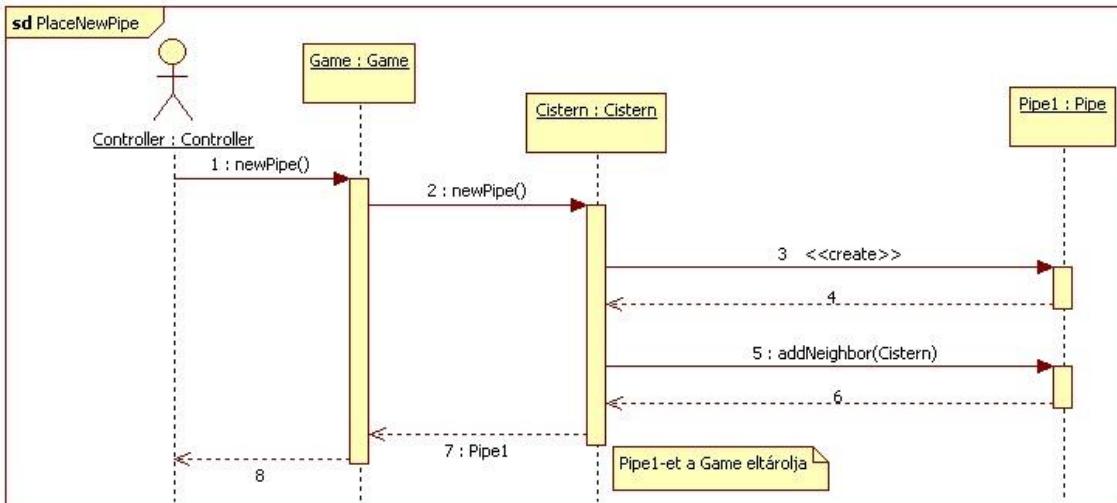
4.4.14 SimulateWaterflowMid



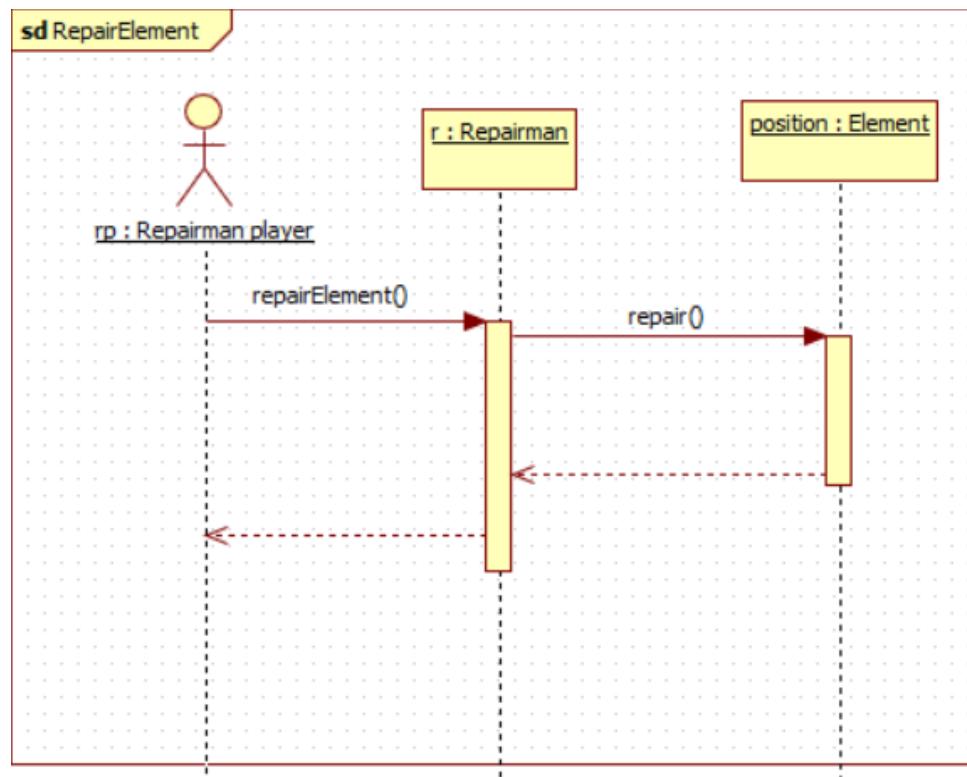
4.4.15 SimulateWaterflowEnd



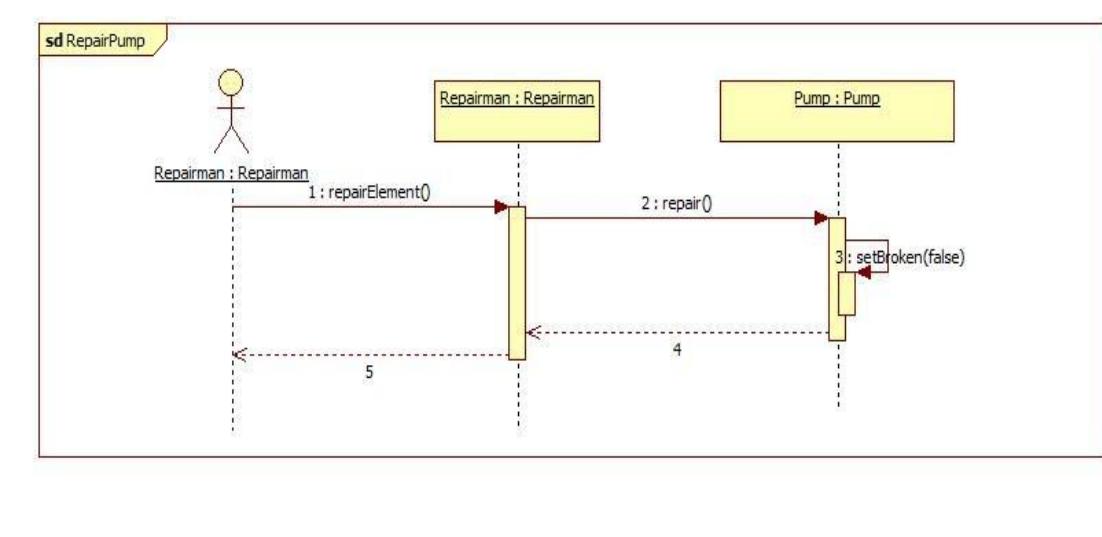
4.4.16 PlaceNewPipe



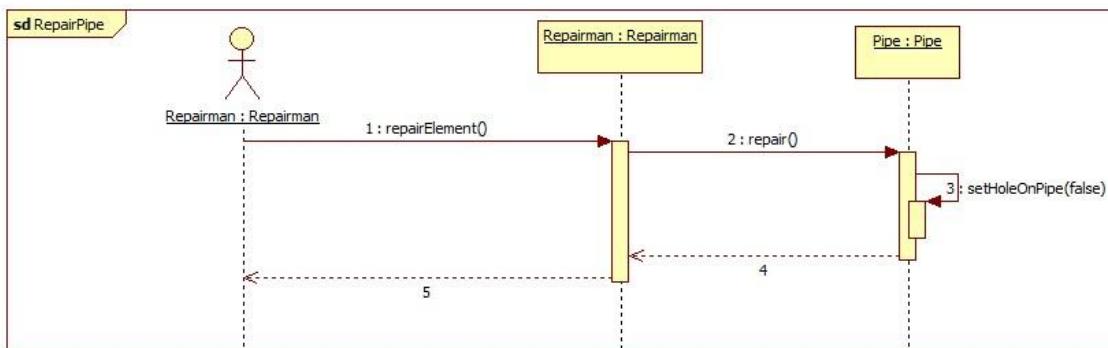
4.4.17 Repair Elements



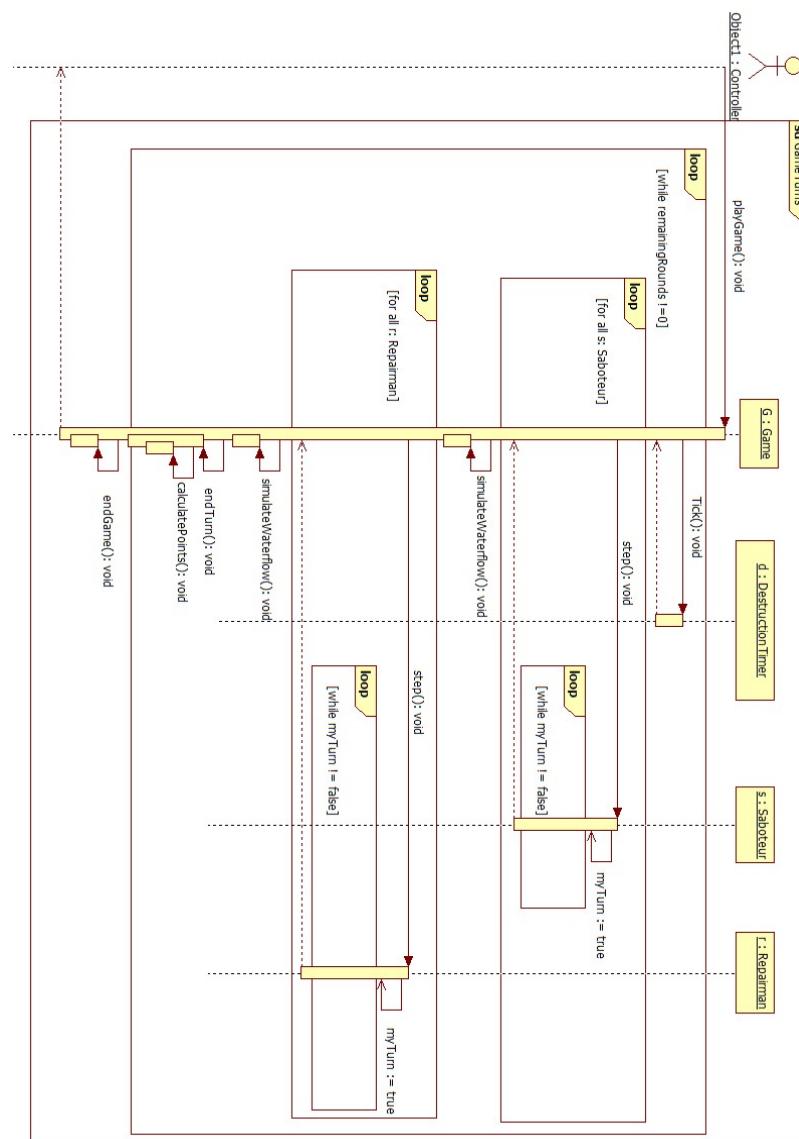
4.4.18 Repair Pump



4.4.19. Repair Pipe

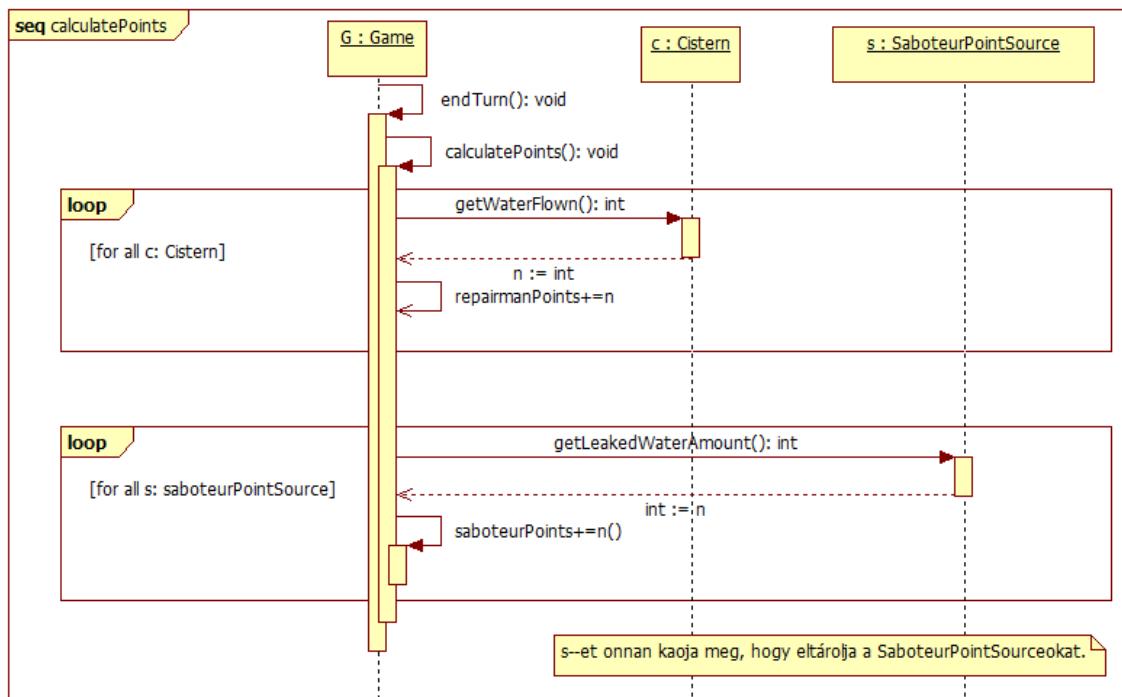


4.4.20. Game Turns



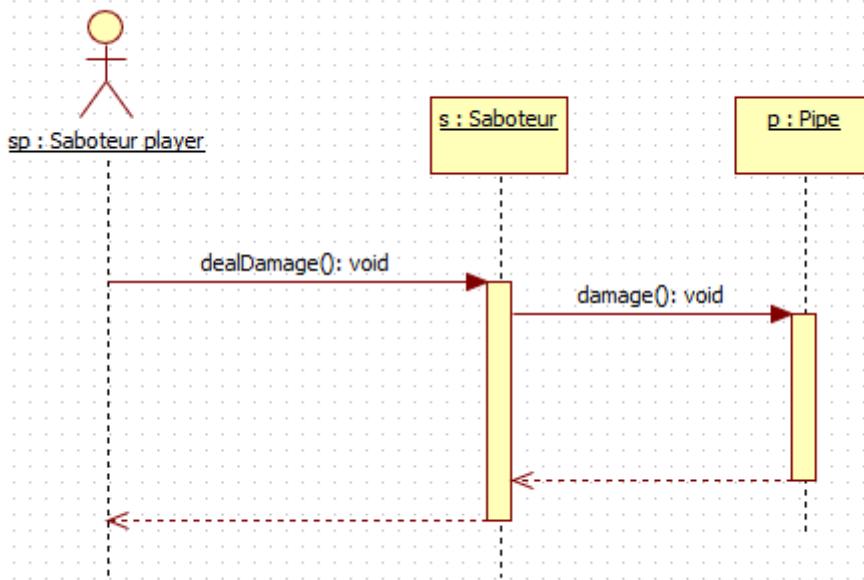
Megjegyzés: A karakterek lépése a 4.4.23 szekvenciadiagramon vannak részletesen bemutatva

4.4.21 Calculate points at the end of turn

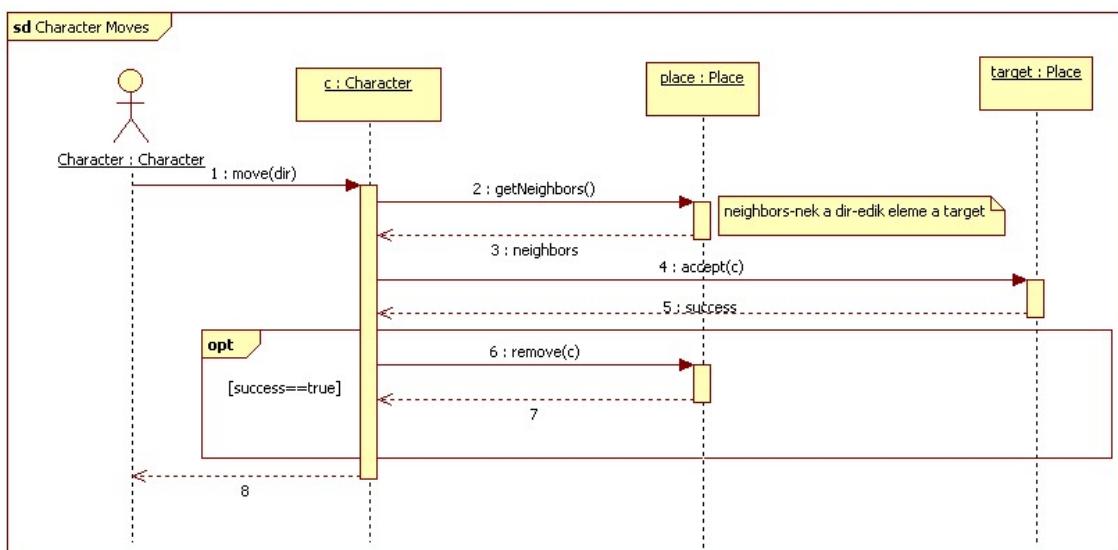


A 4.4.12-es szekvenciadiagramjának végét, a `calculatePoints()` metódust mutatja be pontosabban.

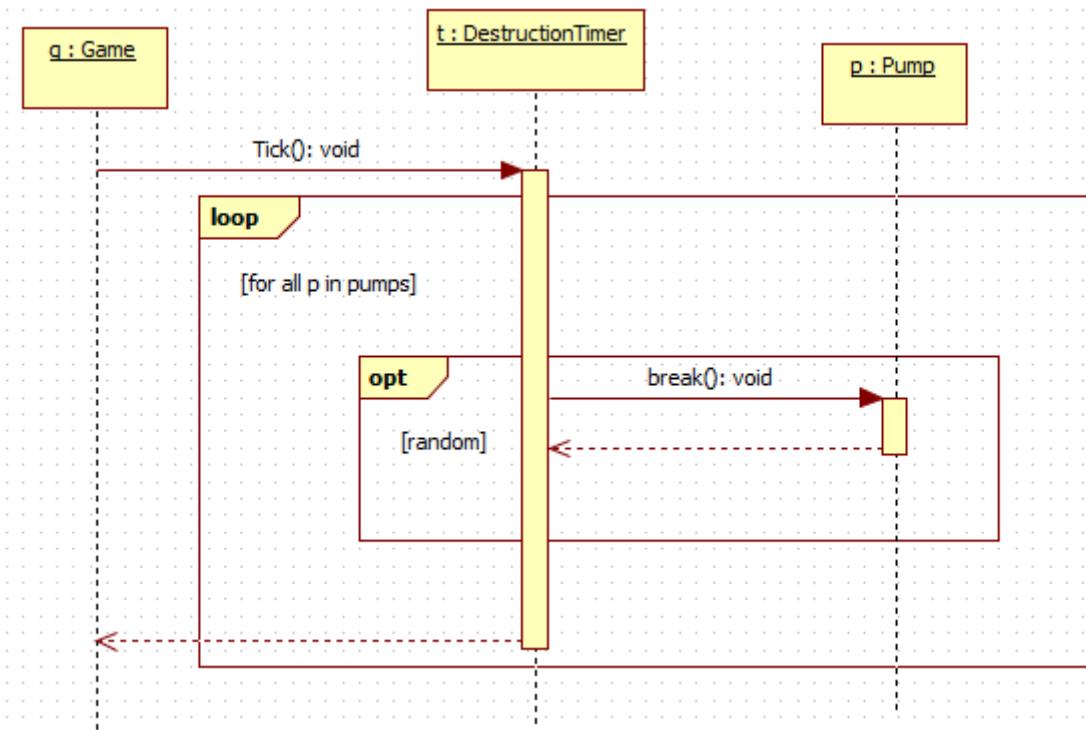
4.4.22. Saboteur Deals Damage



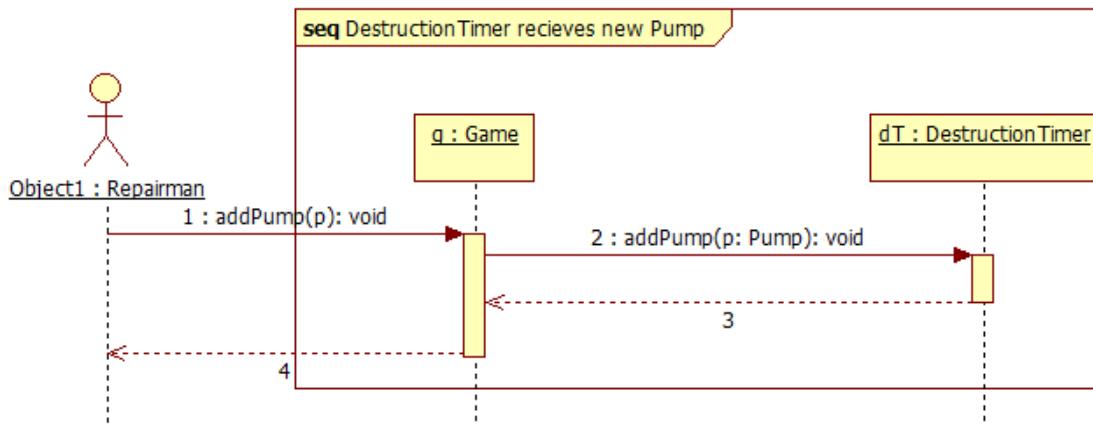
4.4.23. Character Moves



4.4.24 DestructionTimer ticks



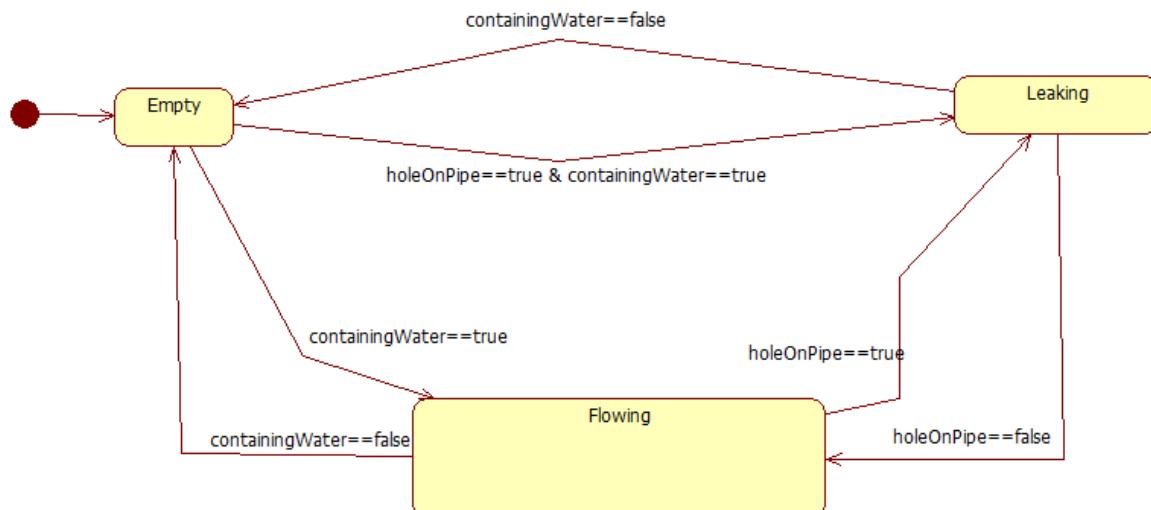
4.4.25 DestructionTimer receives new Pump



Ez a lépés a 4.4.5. szekvenciadiagram 23. és 24. lépése közt történik.

State-chartok

4.4.1 Pipe state-chart



4.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2023.03.24 9:00	1 óra	Andai, Bodnár, Gutási, Szikszai, Vajda	Értekezlet. Döntés: - Vajda javítja az átírandó osztályleírást és osztálydiagramokat. - A többiek szekvenciadiagramokat javítanak, melyeket felosztanak maguk között.
2023.03.24 10:00	1 óra	Andai	3.4.1-3.4.12 Szekvenciadiagramok javítása
2023.03.24	1 óra	Vadja	Osztálydiagram javítása az előző heti értékelés alapján
2023.03.25 11:00	1 óra	Bodnár	Osztályleírások(3.2.2, 3.2.3, 3.2.4, 3.2.5) és osztálydiagram közötti eltérések, hibák keresése/javítása.
2023.03.25 13:00	1 óra	Bodnár	Osztályleírások (3.2.7, 3.2.8, 3.2.9, 3.2.10, 3.2.11) és osztálydiagram közötti eltérések, hibák keresése/javítása.
2023.03.25 14:00	0.5 óra	Andai	3.4.1-3.4.12 szekvenciadiagramok egyeztetése az osztályleírásokkal
2023.03.26 10:00	2.5 óra	Andai	Analízis modell értékelésében szereplő hibák javítása az osztályleírásokban. (3.2)
2023.03.26 10:00	3 óra	Szikszai	Osztályigram és osztályleírások egyeztetése. Szekvencia diagramok javítása.(3.4.13, 3.4.14, 3.4.15, 3.4.16)
2023.03.26 10:00	3 óra	Vajda	Osztálydiagram és osztályleírások egyeztetése (3.3.4, 3.3.5, 3.3.7)
2023.03.26 19:00	1.5 óra	Gutási	3.4.25. Szekvenciadiagram létrehozása, inkonziszencia megszüntetése az osztálydiagram és a dokumentumban levő osztályleírások között.
2023.03.26 19:00	1 óra	Bodnár	Szekvenciadiagramok és osztályleírások közötti inkonziszencia megszüntetése.

5. Szkeleton tervezése

5.1 A szkeleton modell valóságos use-case-ai

5.1.1 Use-case diagram



5.1.2 Use-case leírások

5.1.2.1

Use-case neve	Repairman standing on Cistern and lifts pump
Rövid leírás	A szerelő felvesz egy pumpát.
Aktorok	Repairman
Forgatókönyv	1.1 A szerelő kér a ciszternától egy pumpát. 1.2 A ciszterna létrehoz egy új pumpát. 1.3 A szerelő sikeresen megkapja a pumpát a ciszternától.

5.1.2.2

Use-case neve	Repairman cannot lift pump
Rövid leírás	A szerelő nem kap pumpát.
Aktorok	Repairman
Forgatókönyv	1.1 A szerelő egy olyan objektumtól kér pumpát, ami nem tud neki azt adni. 1.2 A szerelő nem kap pumpát.

5.1.2.3

Use-case neve	Repairman cannot place pump
Rövid leírás	A szerelő nem tudja elhelyezni a pumpát.
Aktorok	Repairman
Forgatókönyv	1.1 A szerelő egy olyan objektumon szeretné elhelyezni a pumpát, ahová nem tudja. 1.2 Sikertelen a letétel.

5.1.2.4

Use-case neve	Repairman places pump
Rövid leírás	A szerelő sikeresen lehelyezi a pumpát.
Aktorok	Repairman
Forgatókönyv	1.1 A szerelő lecsatlakoztatja a cső egyik végén lévő pumpát a csőről. 1.2 A cső ezen végére beköti a szerelő a nála lévő pumpát. 1.3 A szerelő létrehoz egy új csövet, majd ezt beköti a letett pumpába. 1.4 A másik felét a csőnek beköti a lecsatlakoztatott pumpába. 1.5 A szerelő regisztrálja a játékba az új csövet és a pumpát.

5.1.2.5

Use-case neve	Repairman cannot place pump on a zero-end-pipe
Rövid leírás	A szerelő nem tudja lehelyezi a pumpát egy olyan csövön, amely sehol sem van bekötve.
Aktorok	Repairman
Forgatókönyv	1.1 A szerelő megpróbálja lehelyezni a nála lévő pumpát a csövön. 1.2 Nem sikerül a letétel.

5.1.2.6

Use-case neve	PlaceNewPipe
Rövid leírás	A Controller új csövet hoz létre egy ciszternán.
Aktorok	Controller
Forgatókönyv	1.1 A Controller szól a Game-nek, hogy hozzon létre új csövet 1.2 A Game szól a ciszternának, hogy hozzon létre egy új csövet. 1.3.1 A ciszterna létrehoz egy új csövet 1.3.2 Az új csőhöz hozzáadja megát a ciszterna, mint szomszéd. 1.4.1 A ciszterna visszaadja a cső hivatkozását a Game-nek. 1.4.2 A Game eltárolja a cső referenciáját

5.1.2.7

Use-case neve	Repair Pump
Rövid leírás	A szerelő karakter megpróbál megszerelni egy pumpát.
Aktorok	Repairman
Forgatókönyv	<p>1.1 A játékos szól a karakternek, hogy kísérelje meg megszerelni azt a pumpát, amin áll.</p> <p>1.2 A játékos által irányított karakter megkísérli megjavítani a pumpát, amin áll.</p> <p>1.2.1 A tesztelő eldönti, hogy el volt-e romolva a pumpa.</p> <p>1.3.A Ha a pumpa, amin áll, el volt romolva, akkor megjavul.</p> <p>1.3.B Ha a pumpa, amin áll, nem volt elromolva, akkor semmi sem történik.</p>

5.1.2.8

Use-case neve	Repair Pipe
Rövid leírás	A szerelő karakter megpróbál megszerelni egy csövet.
Aktorok	Repairman
Forgatókönyv	<p>1.1 A játékos szól a karakternek, hogy kísérelje meg megszerelni azt a csövet, amin áll.</p> <p>1.2 A játékos által irányított karakter megkísérli megjavítani a csövet, amin áll.</p> <p>1.3.A Ha a cső, amin áll, el volt romolva, akkor megjavul.</p> <p>1.3.B Ha a cső, amin áll, nem volt elromolva, akkor semmi sem történik.</p>

5.1.2.9

Use-case neve	Game Turns
Rövid leírás	A játék elindulását, destructionTimer működését, karakterek lépésének sorrendjét és a játék befejeződését ellenőrző teszt.
Aktorok	Controller, Player
Forgatókönyv	<p>1.0 A tesztelő megadja, hány körös legyen a játék.</p> <p>1.1 A Controller szól a Game osztálynak, hogy a játék elindul.</p> <p>1.2.A Amíg a hátramaradt köröket számláló értéke nem éri el a nullát, addig minden kör elején a Game lépteti a destructionTimert.</p> <p>1.2.3 destructionTimer minden pumpára eldönti, hogy elromoljon-e az adott körben.</p> <p>1.2.3.A Ha a destructionTimer egy adott pumpát tönkretesz, jelzi azt a tesztelő számára.</p> <p>1.2.4 Továbbá minden körben minden játékos léphet:</p> <p>1.2.4.1 Először a szabotőr csapat karakterei léphetnek:</p> <p>1.2.4.1.1 Kiválasztjuk a tesztnél, hogy melyik karakter léphet.</p> <p>1.2.4.1.2.A Game szól az adott játékos által irányított karakternek, hogy léphet.</p> <p>1.2.4.1.3.A Amíg a tesztelő nem nyomja meg az ENTER billentyűt, az adott karakter van soron.</p> <p>1.2.4.1.4 Ha a karakter ENTER-t nyomott, akkor a következő játékos jöhét.</p> <p>1.2.4.2 Ezután a szerelő csapat karakterei léphetnek, egymás után.</p> <p>1.2.4.2.1 Először a szabotőr csapat karakterei léphetnek:</p> <p>1.2.4.2.1.1 Kiválasztjuk a tesztnél, hogy melyik karakter léphet.</p> <p>1.2.4.2.2.A Game szól az adott játékos által irányított karakternek, hogy léphet.</p> <p>1.2.4.2.3.A Amíg a tesztelő nem nyomja meg az ENTER billentyűt, az adott karakter van soron.</p> <p>1.2.4.2.4 Ha a karakter ENTER-t nyomott, akkor a következő játékos jöhét.</p> <p>1.3.A Ha a hátramaradt köröket számláló értéke eléri a nullát:</p> <p>1.3.A.1 Game meghívja önmagán a játékot befejező metódust.</p> <p>1.3.A.2 Megjelenik a két csapat által elért pontszám, ezek alapján eldöntődik, hogy ki nyert.</p> <p>1.3.A.3 Játék véget ér.</p>

5.1.2.10

Use-case neve	SimulateWaterflow(Start, Mid, End)
Rövid leírás	A víz folyik a Forrástól a Ciszternáig a vízhálózaton keresztül.
Aktorok	Controller
Forgatókönyv	<p>1.1 A Game értesíti a Ciszternát, hogy szimulálják a víz folyását.</p> <p>1.2 A Ciszternák értesítik a belé csatlakozó csövet, hogy szimulálja a víz folyását.</p> <p>1.3 A Ciszterá magába szívja a vizet a hozzá csatlakoztatott csőből.</p> <p>1.4 A Ciszterna értesítik az előtte levő aktív elemet a csőhálózatban, ami egy Pumpa.</p> <p>1.5 Ha van víz a Pumpa víztartályában és a Pumpa működik akkor próbál vizet pumpálni a kimeneti csövébe.</p> <p>1.6 A Pumpa értesíti a bemeneti csövét, hogy szimulálja a víz folyását.</p> <p>1.7 Ha a Pumpa kiürült és működik, akkor próbál vizet szívni a tartályába a bemeneti csövéből.</p> <p>1.8 A Pumpa értesíti az előtte lévő aktív elemet, ha az nem önmaga, hogy szimulálja a víz folyását, ami a Forrás.</p> <p>1.9 A Forrás megpróbál vizet pumpálni a kimeneti csöveibe.</p>

5.1.2.11

Use-case neve	Watersource handling a pipe
Rövid leírás	Az egyik Szerelő megpróbál bekötni egy csövet a Forrásba.
Aktorok	Repairman
Forgatókönyv	<p>1.1 A Szerelő megpróbálja bekötni a kezében lévő Cső egyik végén a Forrásba</p> <p>1.2.A Ha a Cső másik vége is a Forrásba vezet akkor a művelet sikertelen.</p> <p>1.2.B Ha a Cső másik vége nem Forrásba van kötve akkor sikeresen beköti.</p>

5.1.2.12

Use-case neve	Repairman lifts pipe
Rövid leírás	A szerelő megpróbálja felemeli az egyik cső végét
Aktorok	Repairman
Forgatókönyv	<p>1.1 A szerelő meghívja lift(dir) metódust azon az objektumon amin áll</p> <p>1.2.A Ha pumpán,cisterán,forráson áll, akkor a paraméterben megadott számú cső egyik végét lecsatlakoztatja az adott objektumról és magához veszi.</p> <p>1.2.B Ha minden végén bekötött csövön áll akkor a felemelés sikertelen lesz</p> <p>1.2.C Ha egyik végén szabad csövön áll akkor a szabad véget magához veszi.</p>

5.1.2.13

Use-case neve	Repairman placing pipe
Rövid leírás	Szerelő lehelyezi a nála lévő csövet
Aktorok	Repairman
Forgatókönyv	<p>1.1 A szerelő meghívja a placePipe(HoldingPipe) metódust az objektumon amin áll</p> <p>1.2.A Ha ez nem egy cső akkor a szerelő sikeresen lehelyezi a nála lévő csövet.</p> <p>1.2.B ha egy csövön hívja meg akkor nem sikerül a lehelyezni a csövet.</p>

5.1.2.14

Use-case neve	Pump or cistern handling pipe
Rövid leírás	Szerelő lehelyez egy nála lévő csöve egy pumpára vagy cisternára
Aktorok	Repairman
Forgatókönyv	<p>1.1 A Szerelő megpróbálja bekötni a kezében lévő Cső véget egy pumpába vagy cisternába</p> <p>1.2. Amennyiben van a szerelőnél cső a lehelyezés minden sikeres</p>

5.1.2.15

Use-case neve	Calculate points at the end of turn
Rövid leírás	Pontok összeszámolása a kör végén
Aktorok	Controller
Forgatókönyv	<p>1.1 A kör véget ér és a game osztályban tárolt pontokat szeretnénk frissíteni.</p> <p>1.2 A Game objektum az összes benne tárolt ciszternától lekérdezi, hogy mennyi víz folyt bele.</p> <p>1.3 Ezt a lekérdezett víz mennyiséget hozzáadja a szerelő csapat pontjaihoz.</p> <p>2.1. Lekérdezi, az összes olyan elemet amiből elfolyhat víz, hogy mennyi víz folyt el belőlük.</p> <p>2.2. Ezt a lekérdezett elfolyt víz mennyiséget hozzáadja a szabotőr csapat pontjaihoz.</p>

5.1.2.16

Use-case neve	Saboteur Deals Damage
Rövid leírás	Egy szabotőr kilyukaszt egy objektumot.
Aktorok	Saboteur
Forgatókönyv	<p>1.1 Egy szabotőr megpróbálja kilyukasztani az objektumot amin áll.</p> <p>1.2.A Ha az objektum már lyukas volt, akkor nem történik semmi. Ellenkező esetben amennyiben az objektum lyukasztható, az kilyukad</p>

5.1.2.17

Use-case neve	Character Moves
Rövid leírás	Egy karakter mozgása a csőrendszeren.
Aktorok	Saboteur, Repairman
Forgatókönyv	<p>1.1 A karakter kiválasztja melyik objektumra szeretne átlépni.</p> <p>1.2 Az aktuális objektum amin éppen áll, visszaadja a választott objektumot a szomszédai közül.</p> <p>1.3 A cél objektumnak átadódik a karakter, tehát megpróbál átlépni oda.</p> <p>1.4.A Ha az előbbi átlépés sikeres volt akkor a korábbi pozícióról eltávolítjuk a karaktert.</p> <p>1.4.B Ha az előbbi átlépés sikertelen volt akkor marad a korábbi pozíción a karakter.</p>

5.1.2.18

Use-case neve	DestructionTimer ticks
Rövid leírás	Pumpák véletlenszerű elromlása.
Aktorok	Controller
Forgatókönyv	<p>1.1 A kör elején a Game objektum jelez a DestructionTimer objektumnak.</p> <p>1.2 A DestructionTimer végigiterál az összes pumpán.</p> <p>1.3 Iteráció közben valamilyen valószínűsséggel elrontja az adott pumpát.</p>

5.1.2.19

Use-case neve	DestructionTimer receives new Pump
Rövid leírás	Új pumpa elhelyezésekor a DestructionTimer-ben tárolt pumpák listájának bővítése.
Aktorok	Repairman
Forgatókönyv	<p>1.1 A szerelő átadja a game objektumnak a pumpát amit elhelyez a csőrendszerben.</p> <p>1.2 A kapott pumpát a game objektum továbbadja a DestructionTimer objektumnak, amit majd hozzáad a listához.</p>

5.2 A szkeleton kezelői felületének terve, dialógusok

A szkeleton konzolos alkalmazás, mely a standard bemenetről fogad bemeneteket. A program indításakor a szkeleton kilistázza az elérhető tesztek nevét, számokkal megjelölve azokat. Ezek alapján a lista egy sora például a következőképpen nézhet ki:

```
3 - Repairman cannot lift pump
```

A felhasználó ezután a megfelelő számot beírva, majd ENTER gombotütve kiválasztja a kívánt tesztet, és futtatja azt.

A tesztek során a Skeleton, minden a teszthez tartozó objektum létrehozást, metódushívást, értékállítást és visszatérési értékeket is egyértelműen jelöl. Az inicializáló metódushívásokat a Skeleton nem írja ki. Nyomon követhető továbbá az is, hogy a teszthez tartozó use case esetén melyik pontnál tartunk. Ezt az adott teszt száma után írja ki a program. A teszt eredményét pedig az utolsó sor tartalmazza (siker, vagy sikertelen teszt szöveggel).

A kimenetben **a→b.Foo()** típusú jelölések jelentik azt, hogy az **a** objektum meghívja **b Foo** metódusát. A teszt legelső metódushívásának jelölése **BEGIN→b.Foo()** -, míg a legutolsó **END←b.Foo()** alakú. A metódus után kettősponttal elválasztva lehet visszatérési érték. Ha egy objektum önmagán hív meg metódust, vagy állít attribútumokat, akkor azt egyszerűen kiírja a skeleton a következő formában: **b.bar=true**, ahol **b** objektum **bar** attribútumát true értékre állította.

A teszt további pontjai egyértelműen követhetők: az alkalmazás kiírja, a teszt forgatókönyvének hányadik számú pontjánál tart éppen. minden olyan pontban, ahol a felhasználó által beírt érték alapján folytatódik az adott teszt, a standard kimeneten a forgatókönyv pontjának száma helyén **DECISION** szó olvasható. Ezt követi az eldöntendő kérdés, utána pedig zárójelek között felsorova a lehetséges válaszok.

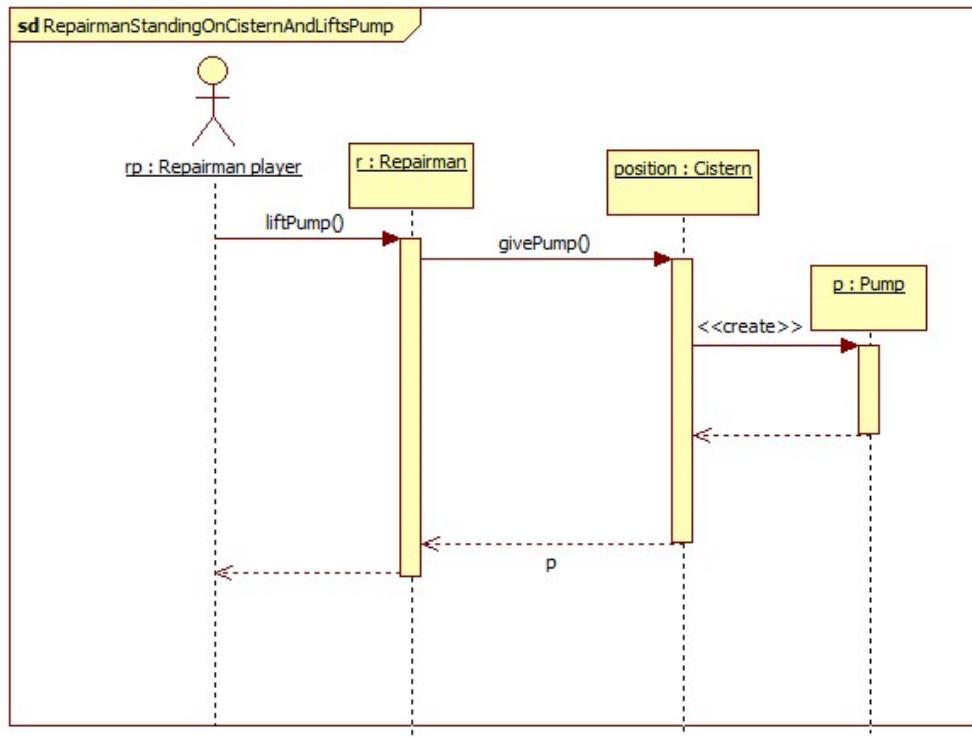
Egy teszt lefutása például a következőképpen nézhet ki:

```
8. Repair Pump
1.1 BEGIN->r1.repairElement()
1.2 r1->p1.repair()
DECISION - A pumpa el volt romolva ? (I,N)> I
1.3.A          p1.broken=false
r1<-p1.repair():void;
END←r1.repairElement();
```

5.3 Szekvencia diagramok a belső működésre

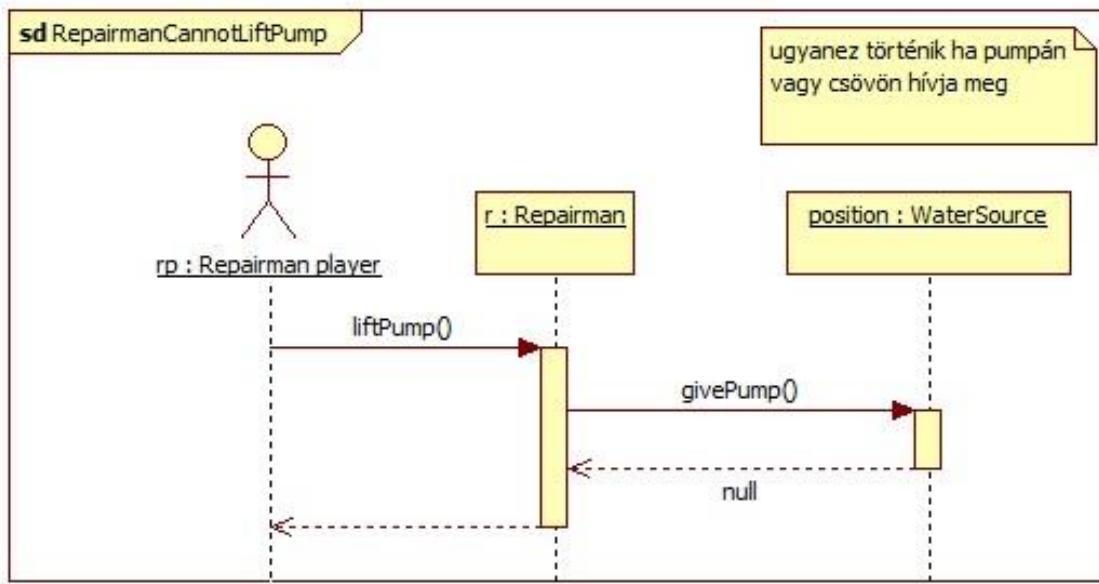
5.3.1 Repairman standing on Cistern and lifts pump

Inicializálja: 5.4.1

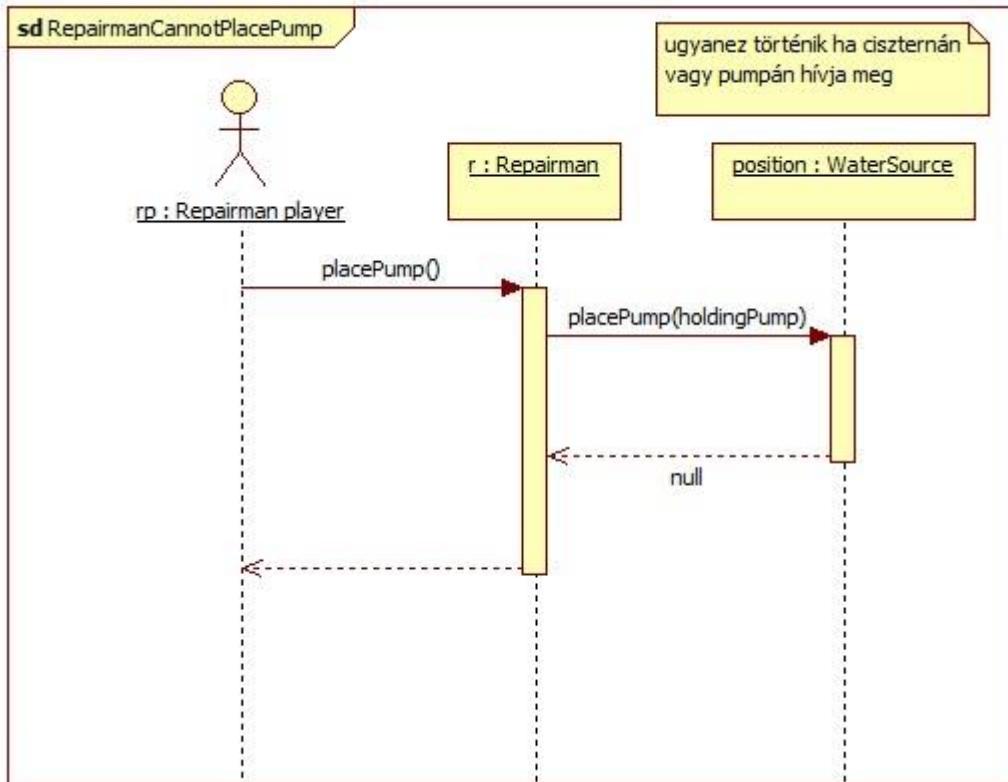


5.3.2 Repairman cannot lift pump

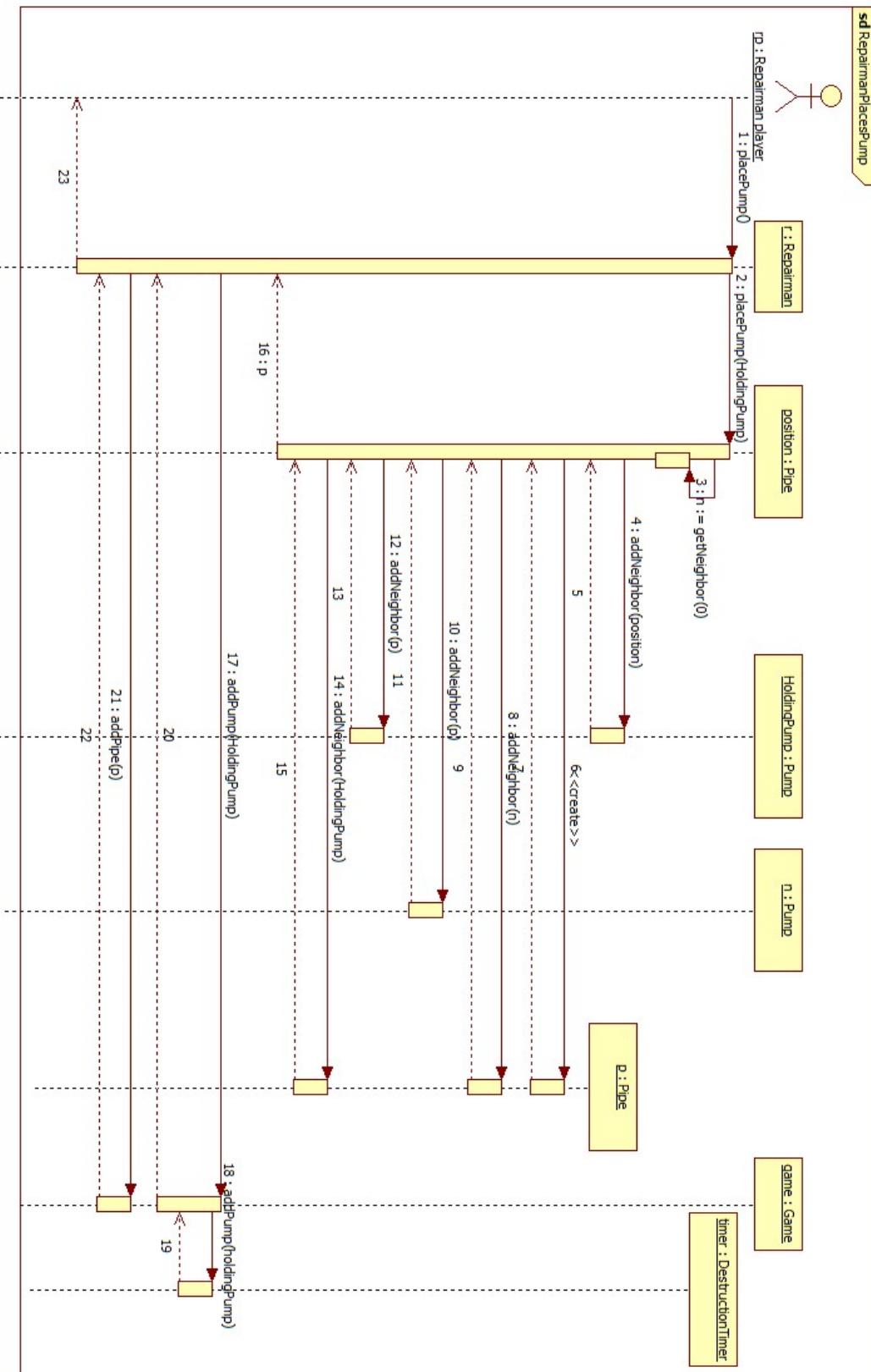
Inicializálja: 5.4.2



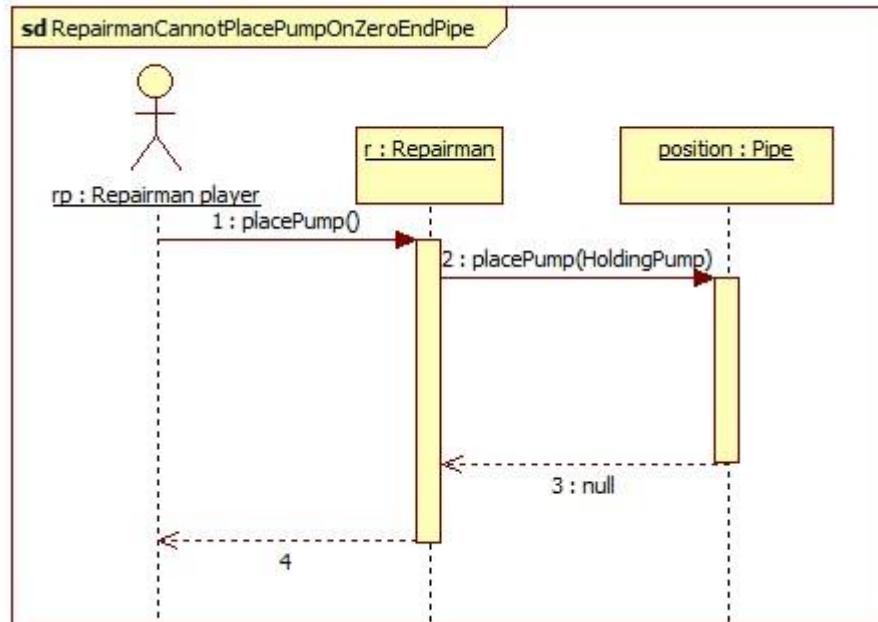
5.3.3 Repairman cannot place pump
Inicializálja: 5.4.3



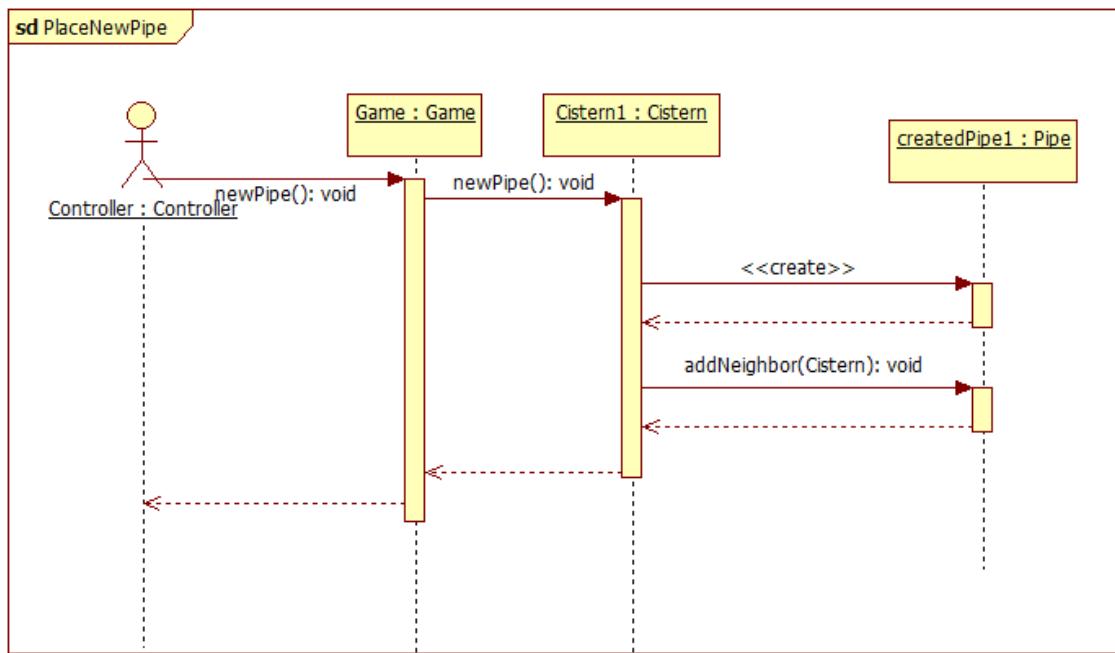
5.3.4 Repairman places pump
Inicializálja: 5.4.4



5.3.5 Repairman cannot place pump on a zero-end-pipe
 Inicializálja: 5.4.5

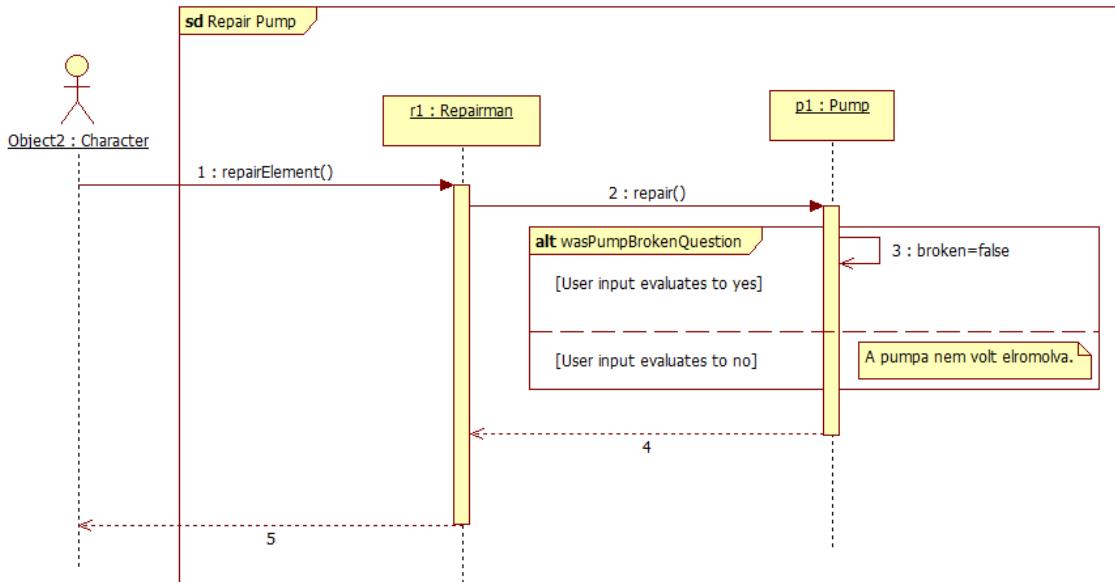


5.3.6 PlaceNewPipe
 Inicializálja: 5.4.6



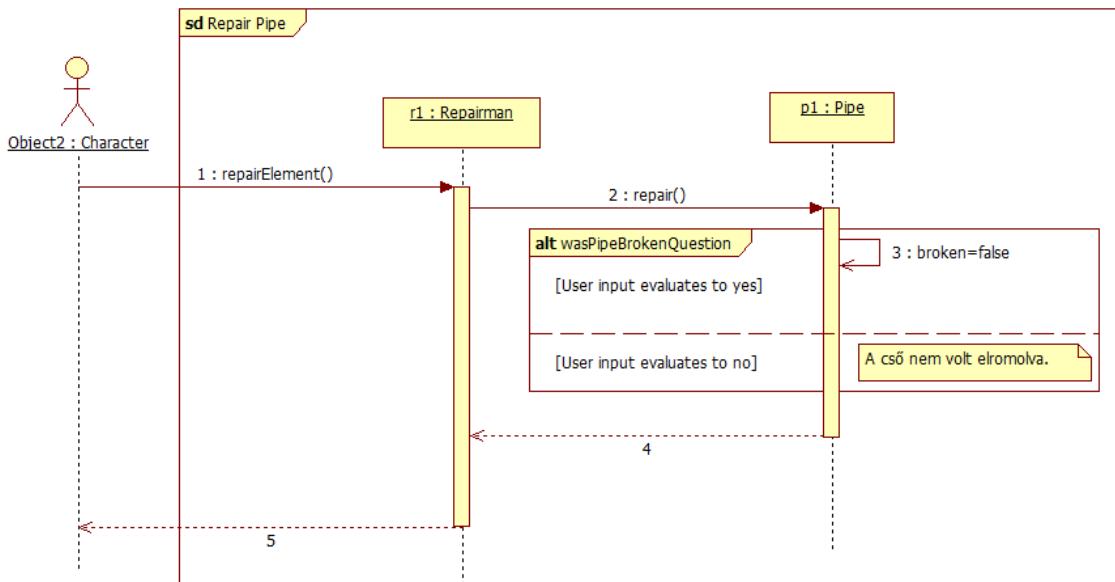
5.3.7 Repair Pump

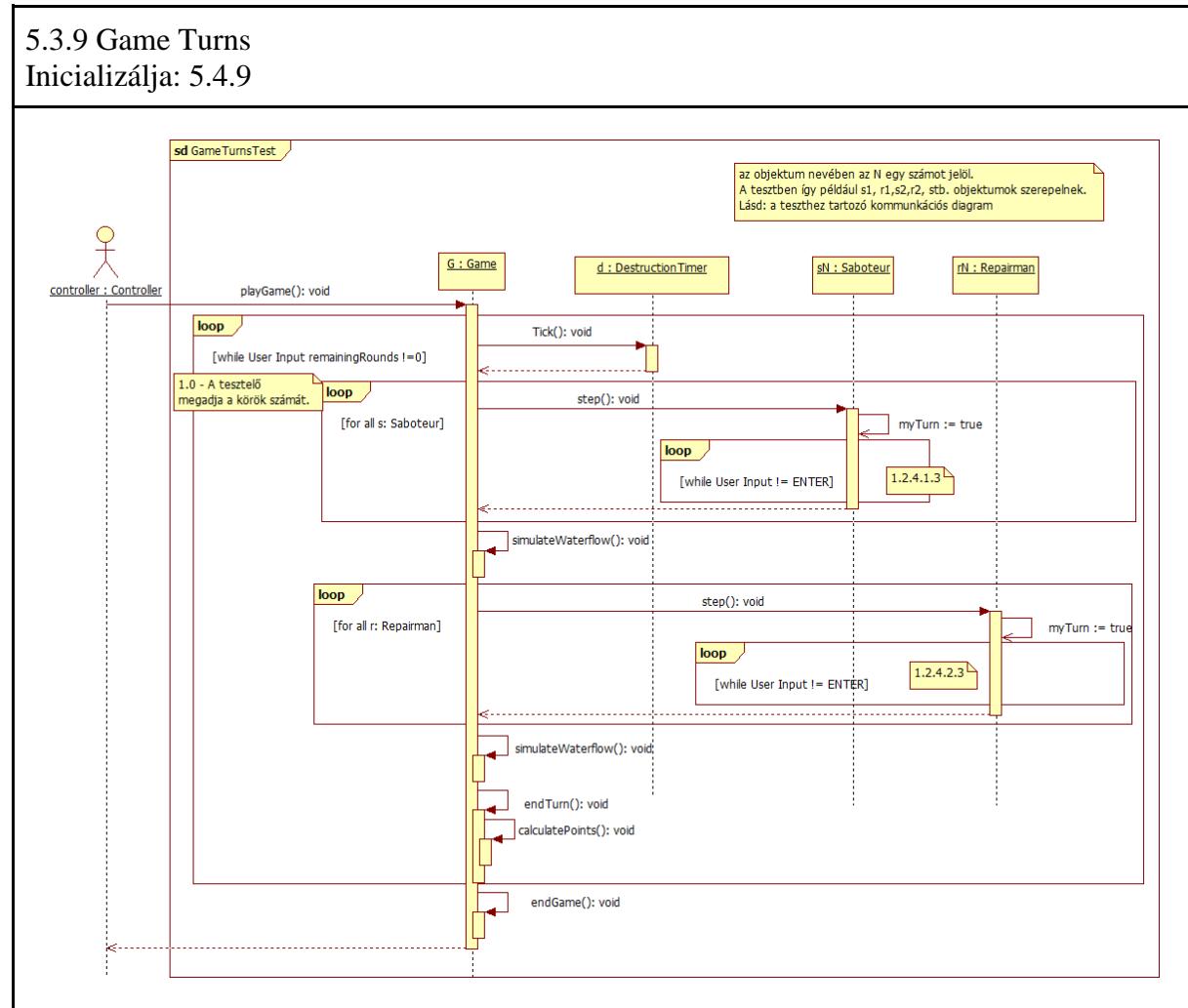
Incializálja: 5.4.7

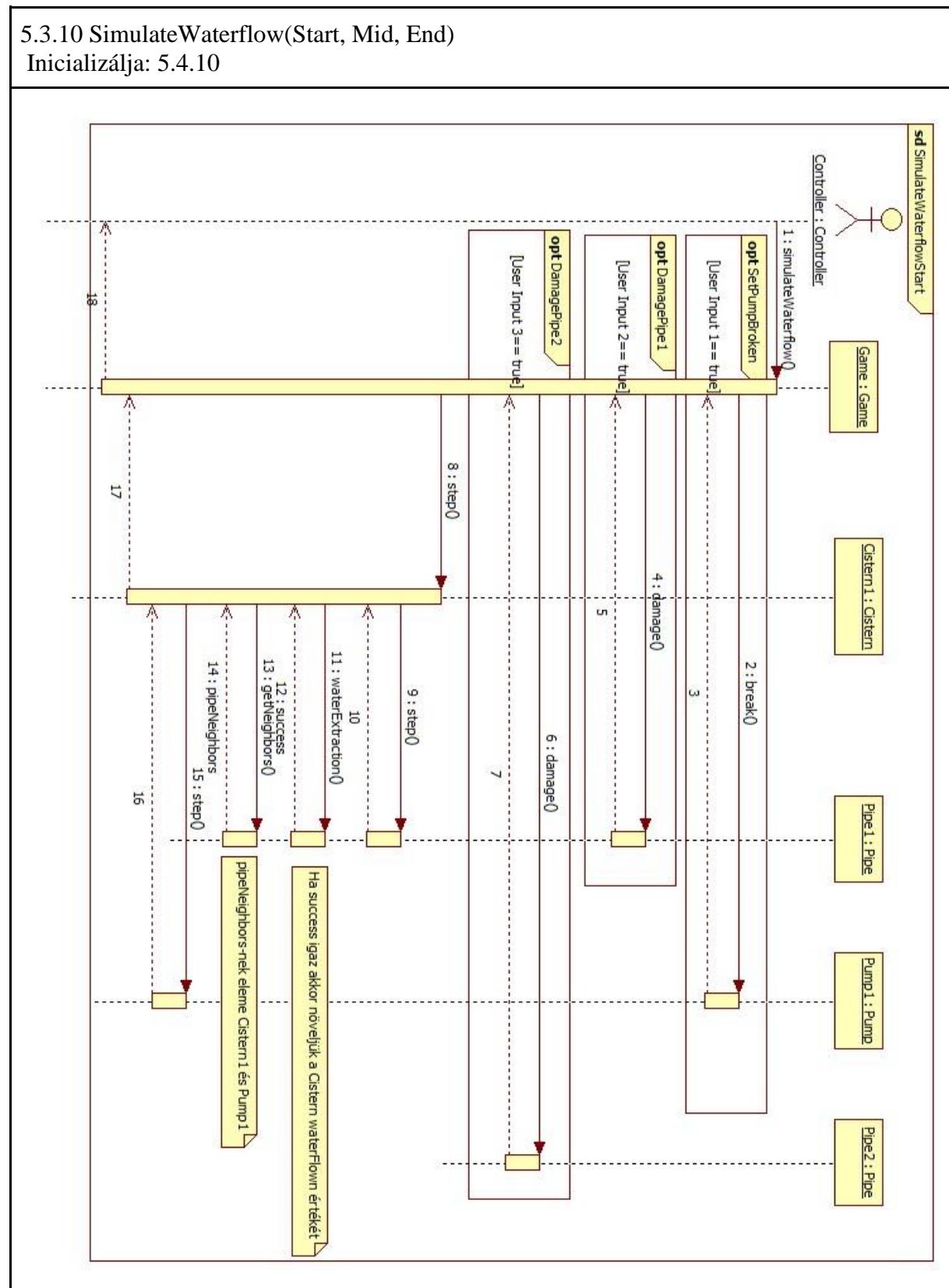


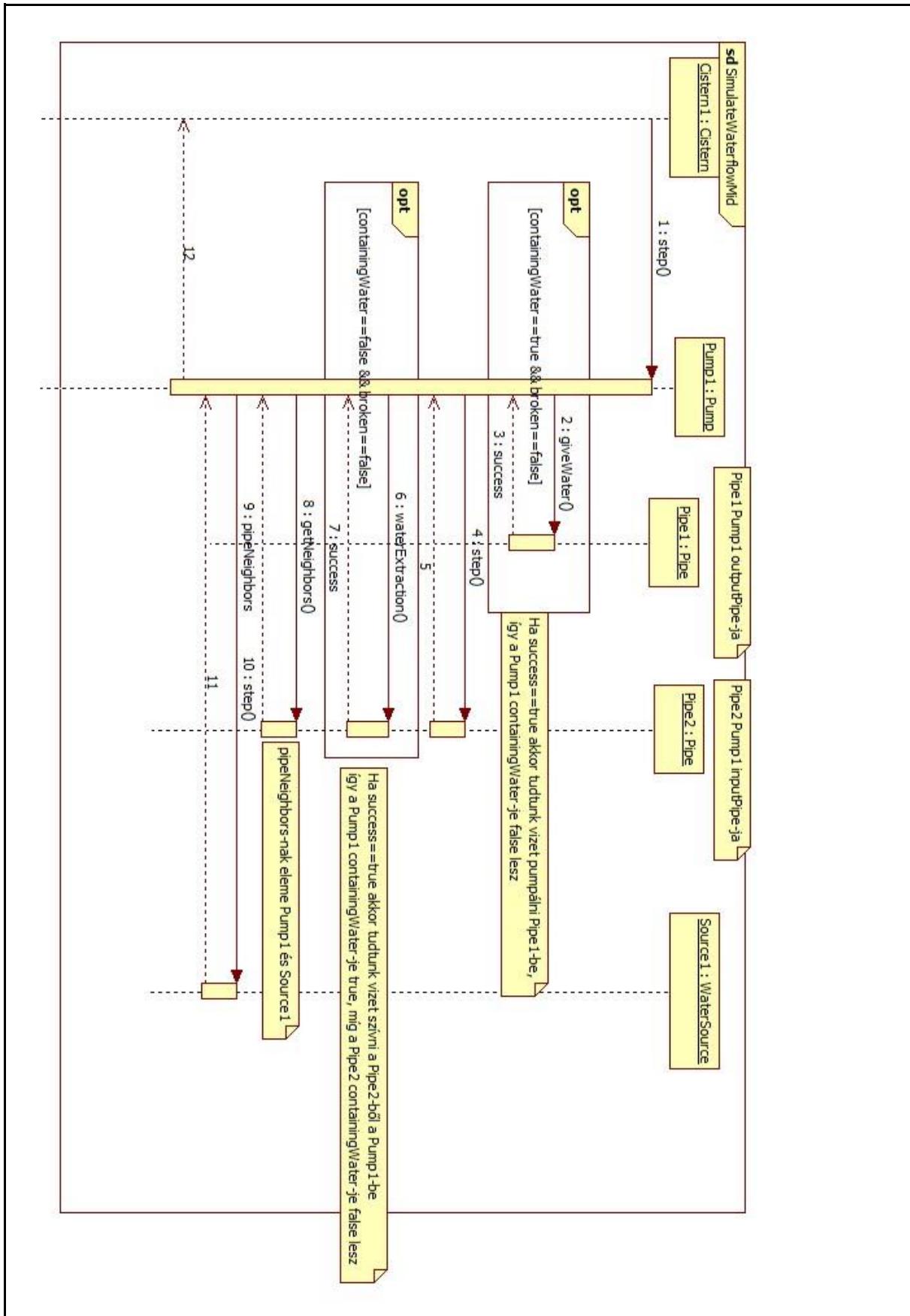
5.3.8 Repair Pipe

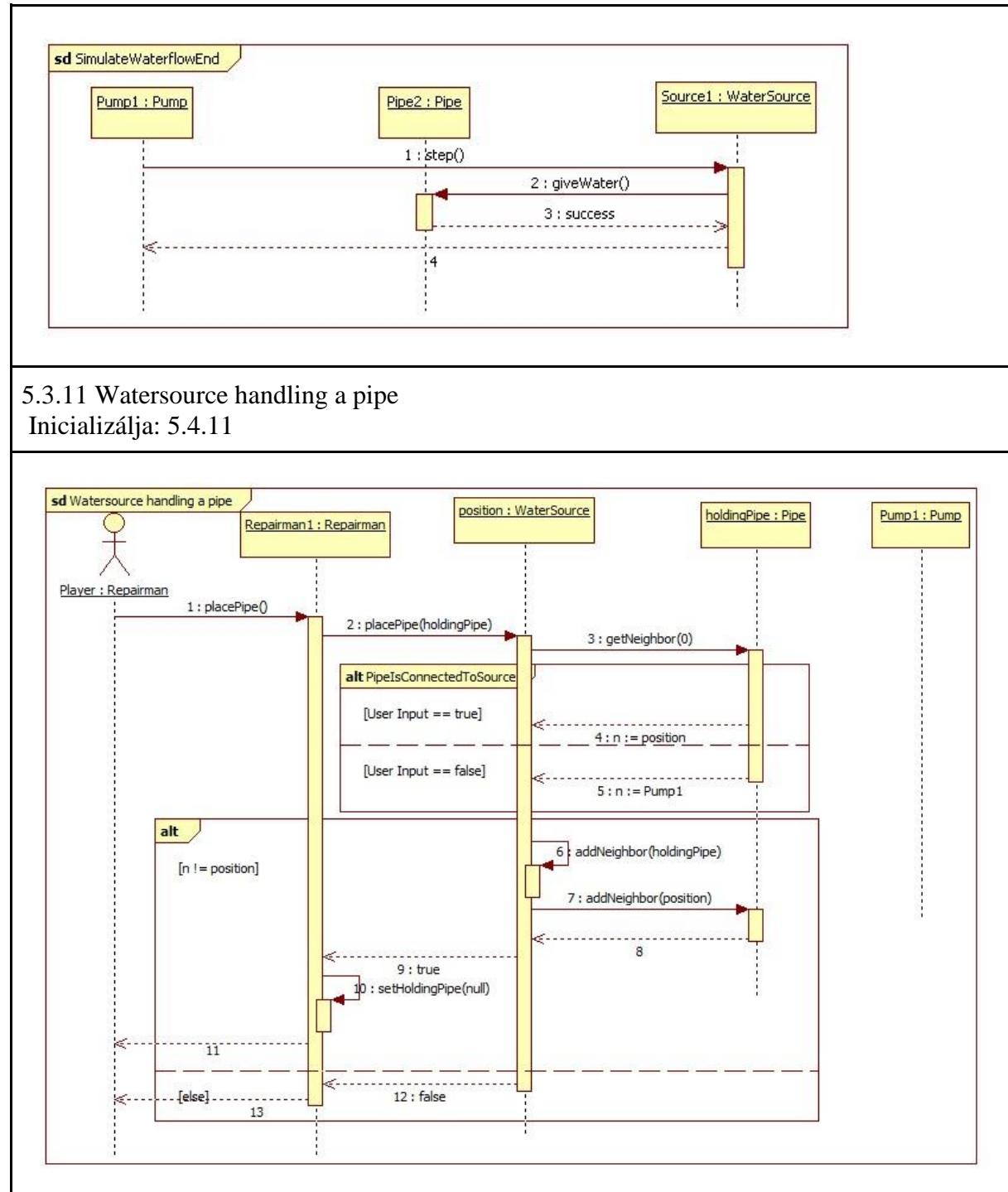
Incializálja: 5.4.8





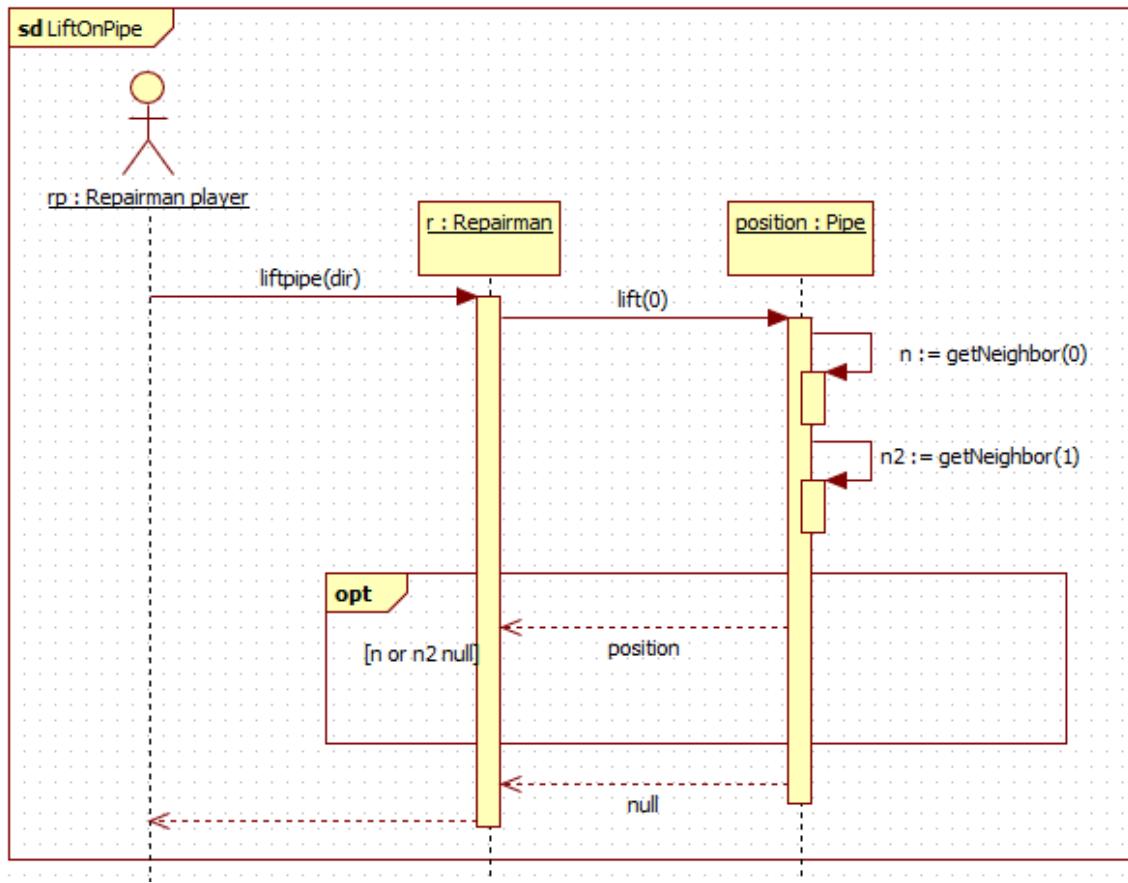






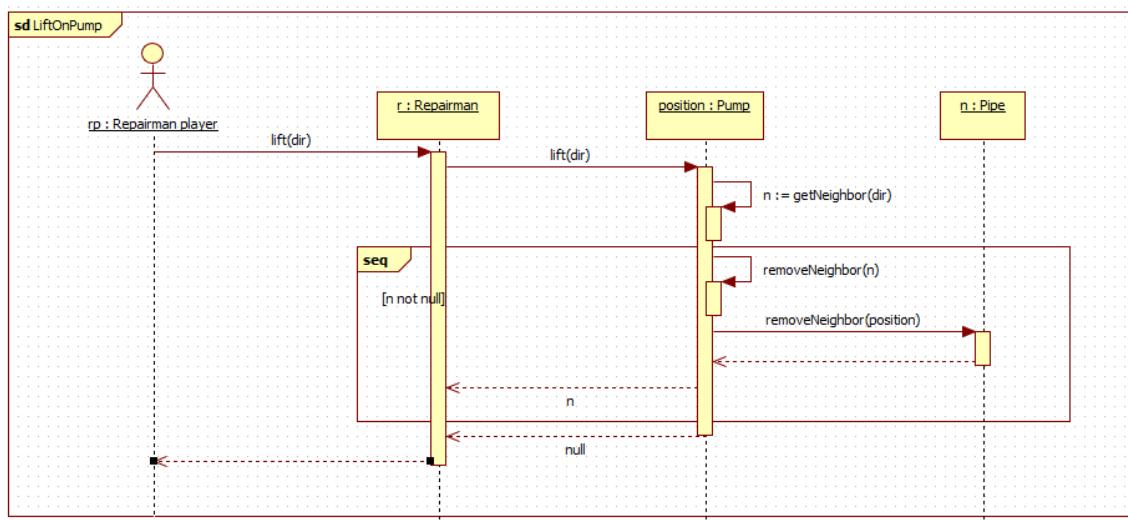
5.3.12 Repairman lifts pipe

Inicializálja: 5.4.12.1



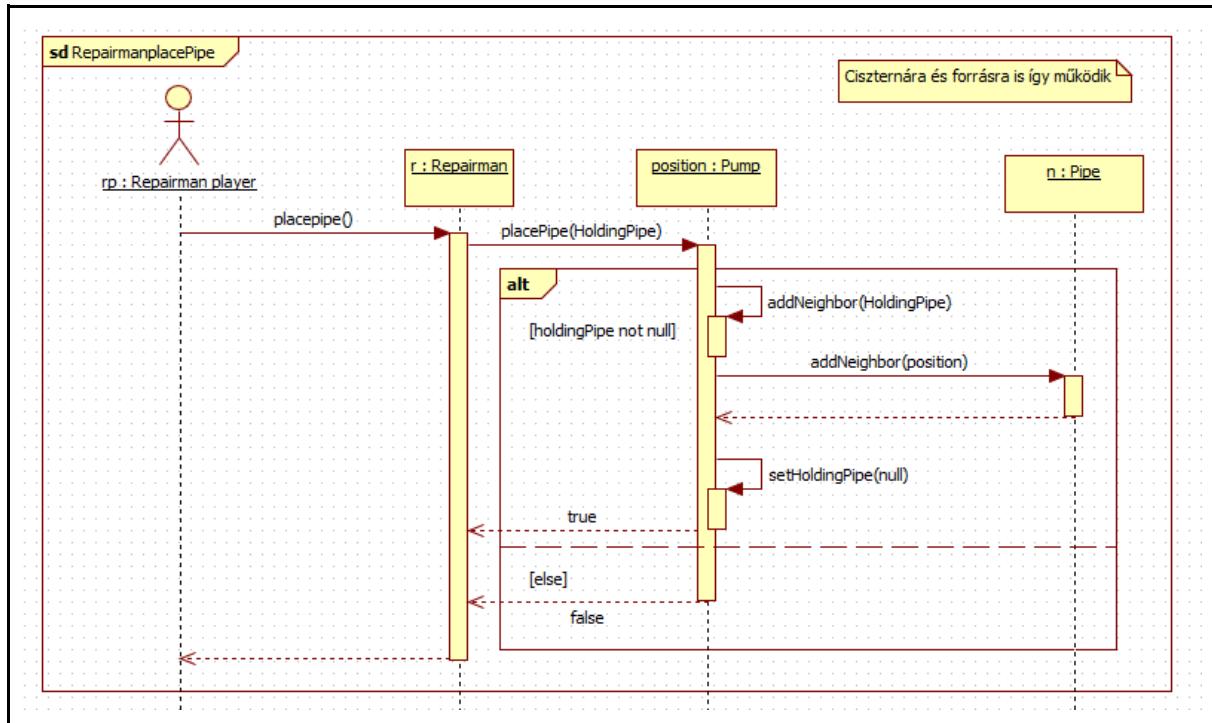
Repairman Lift on pump

Inicializálja: 5.4.12.2



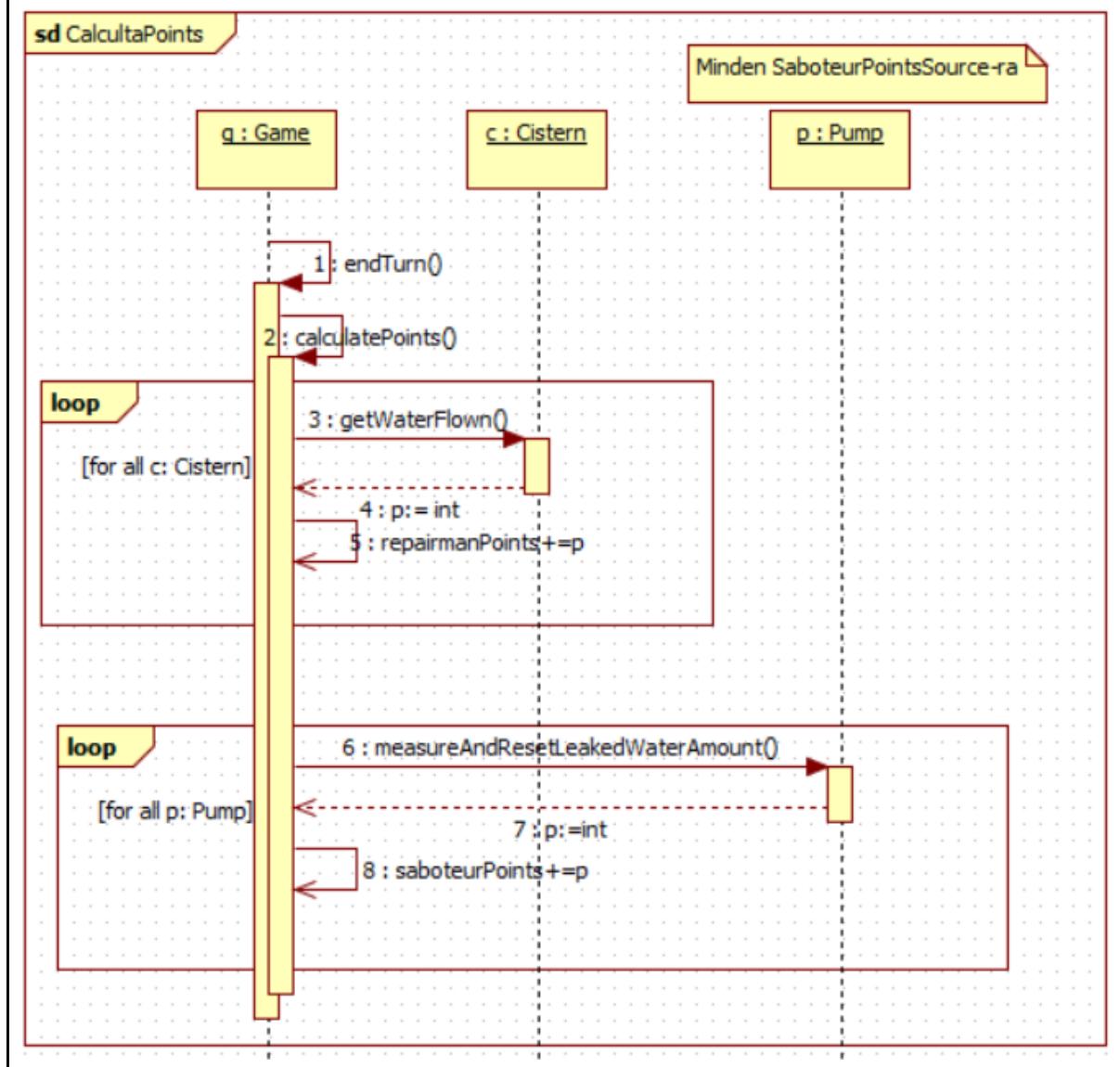
5.3.13 Repairman place pipe

Inicializálja: 5.4.13



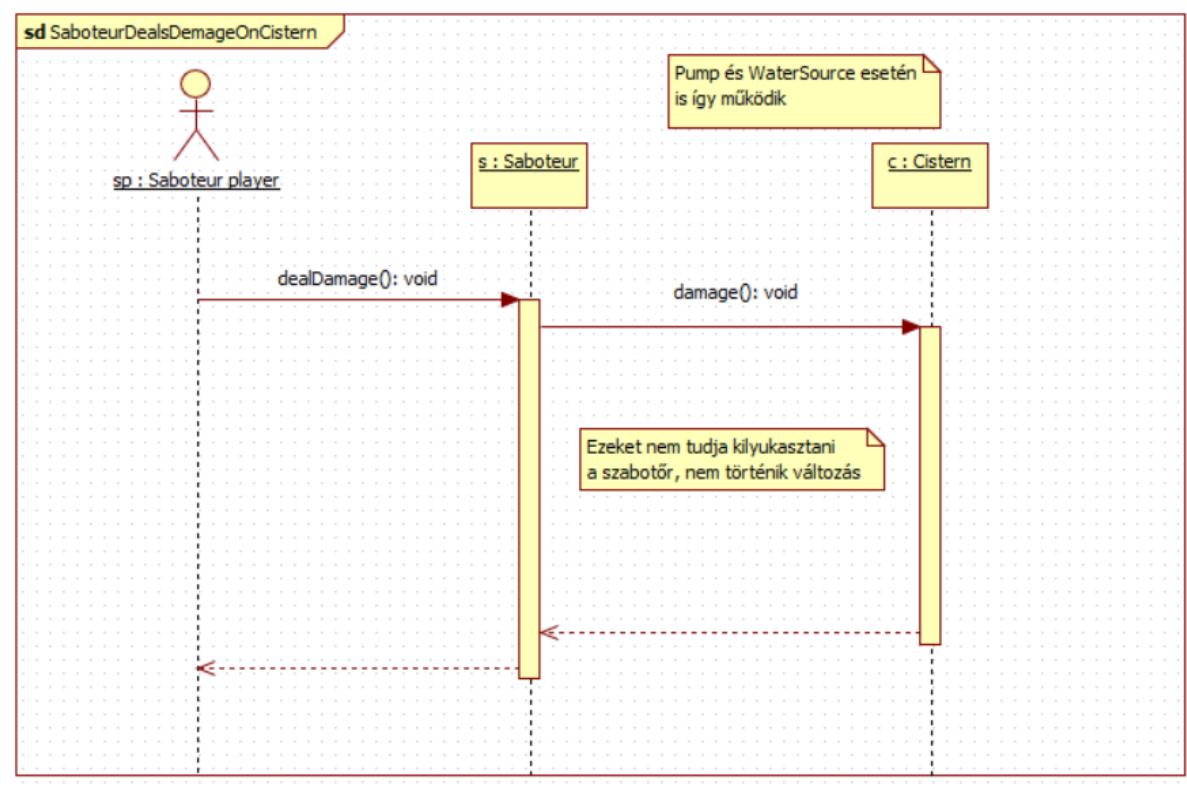
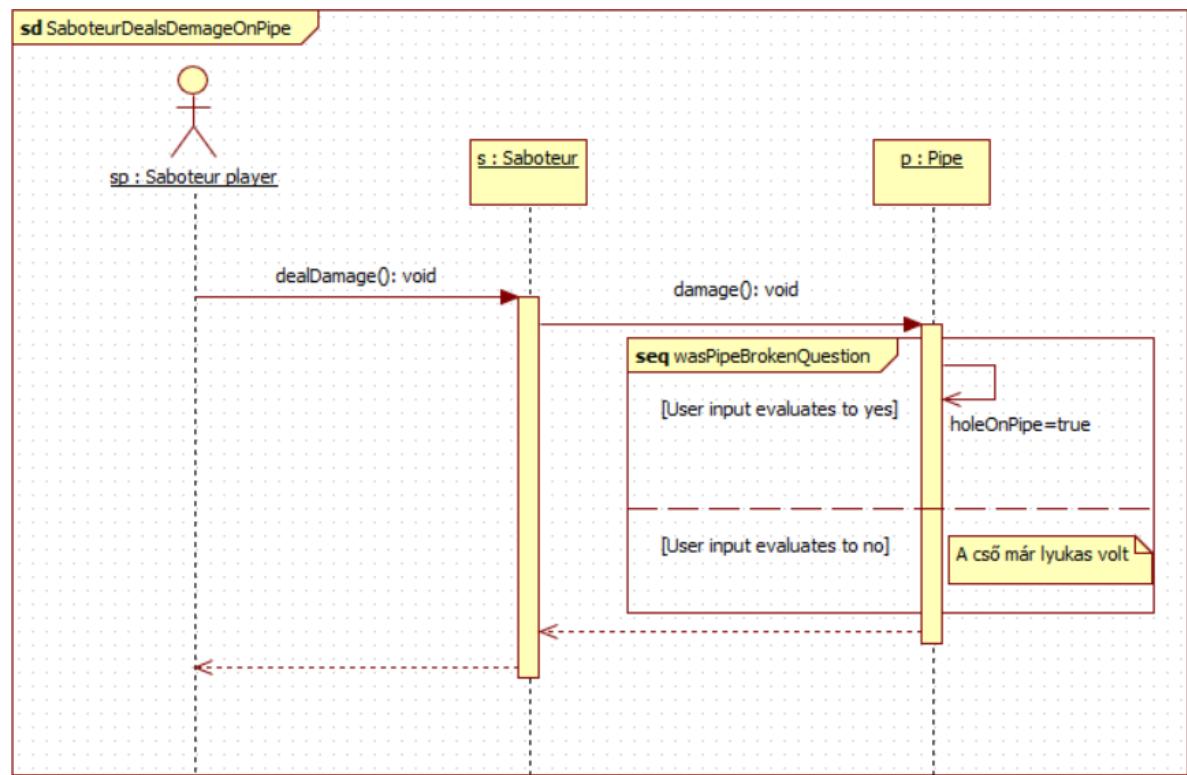
5.3.15 Calculate Points

Incializálja: 5.4.15



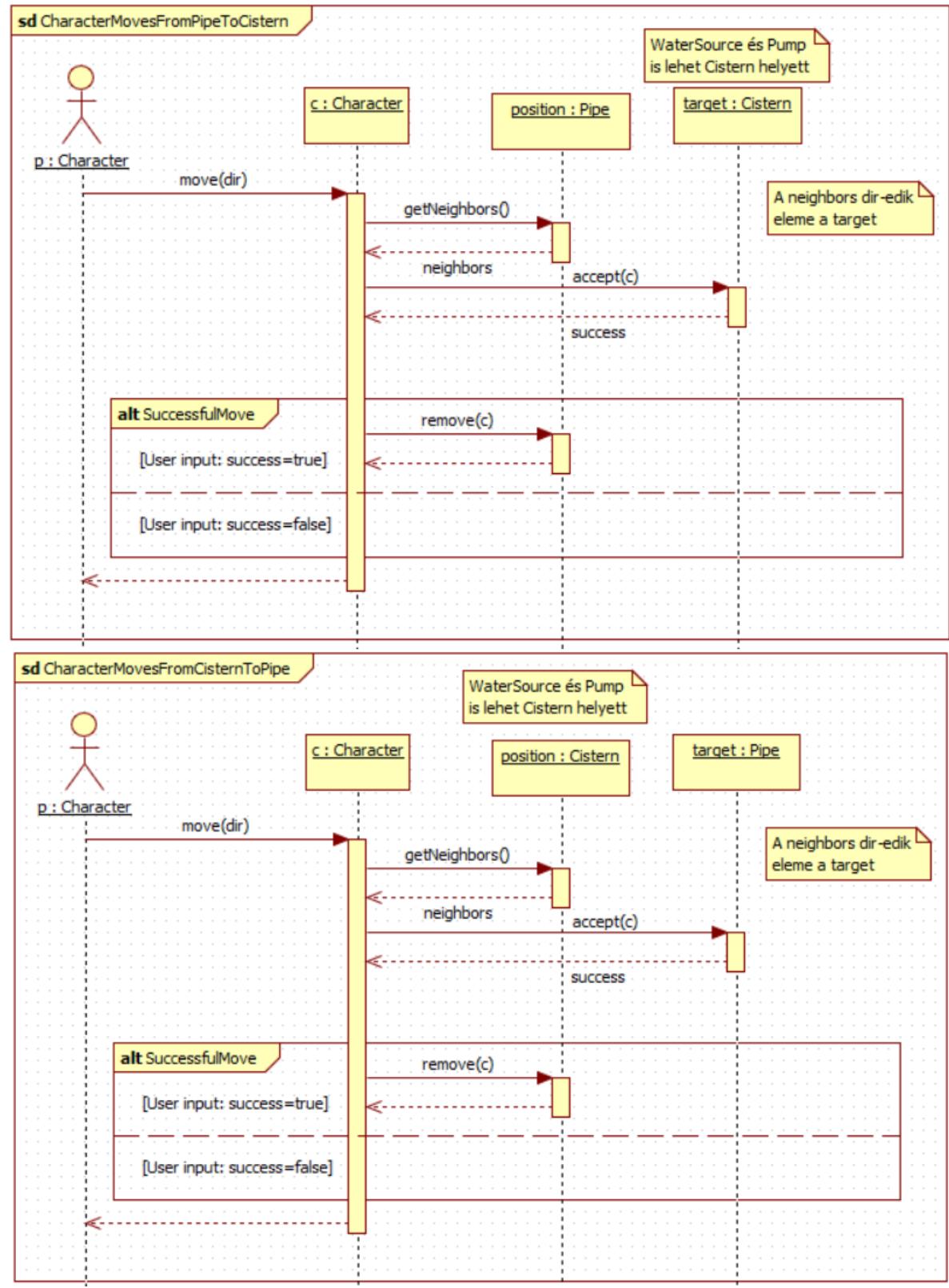
5.3.16 SaboteurDealsDamageOnPipe

Incializálja: 5.4.16



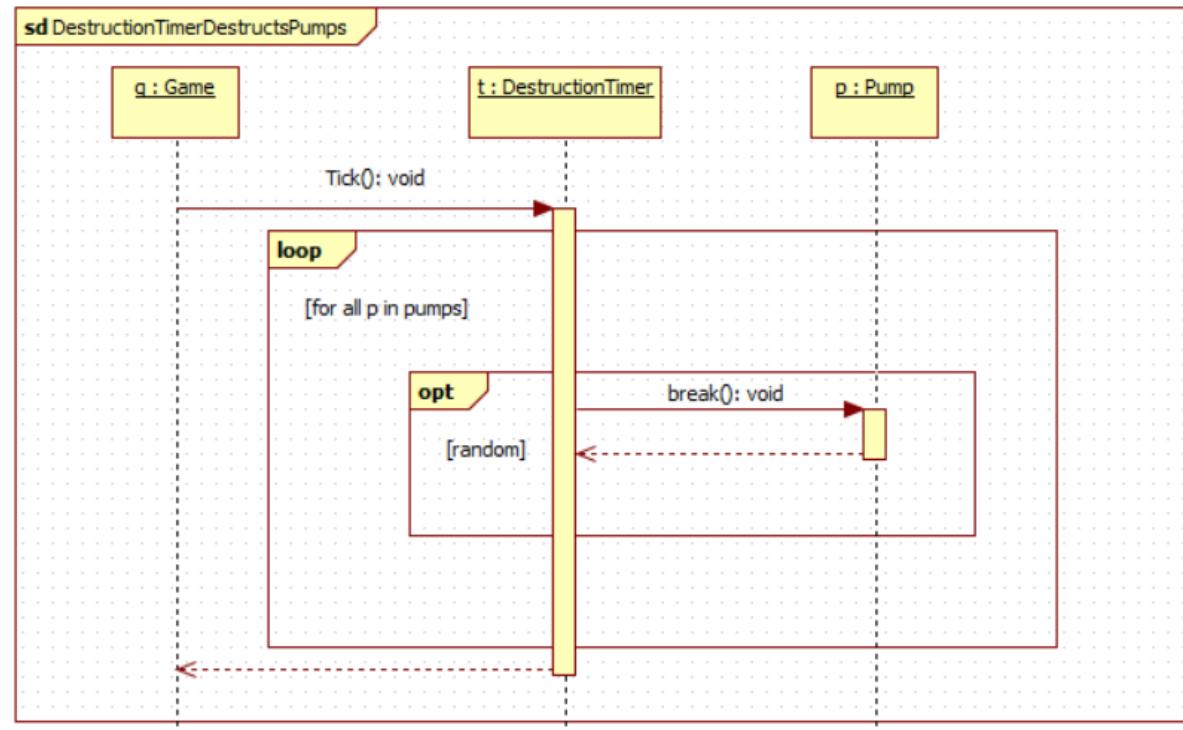
5.3.17 Character Moves

Incializálja: 5.4.17



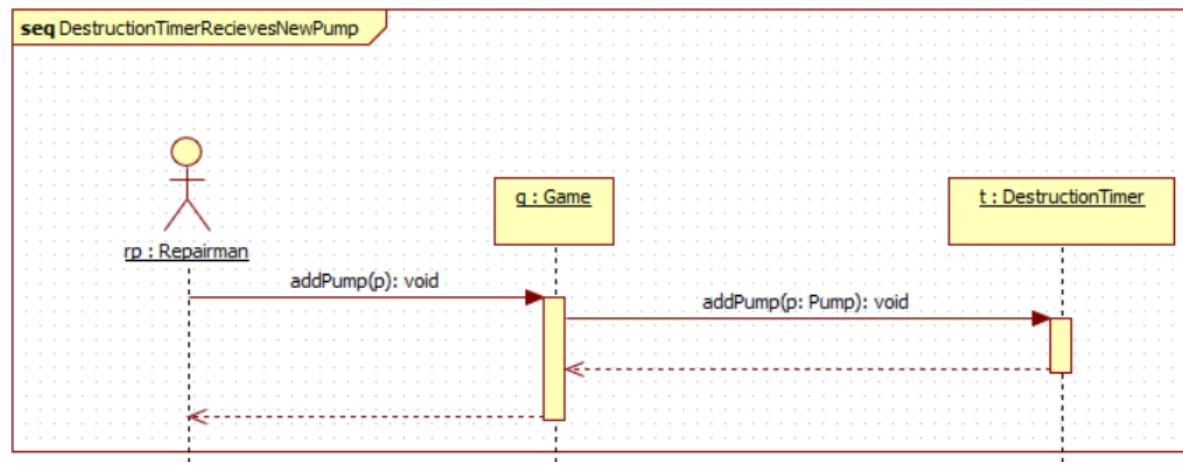
5.3.18 DestructionTimer ticks

Incializálja: 5.4.18



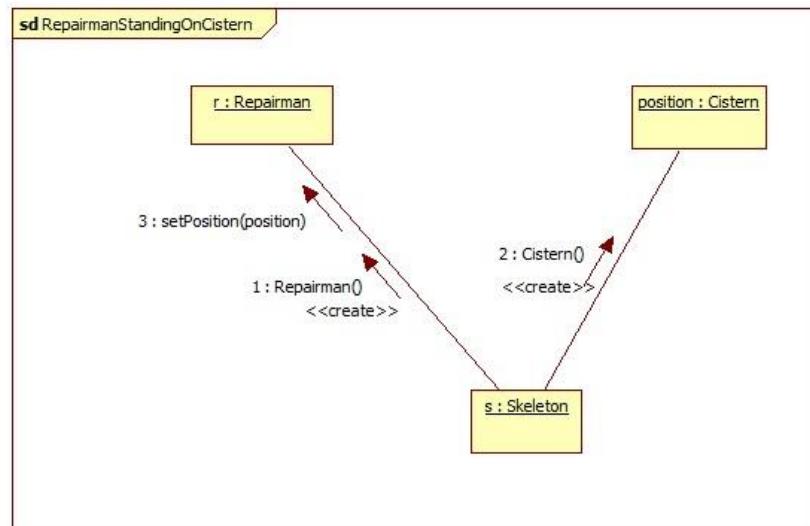
5.3.19 DestructionTimerRecievesNewPump

Incializálja: 5.4.19

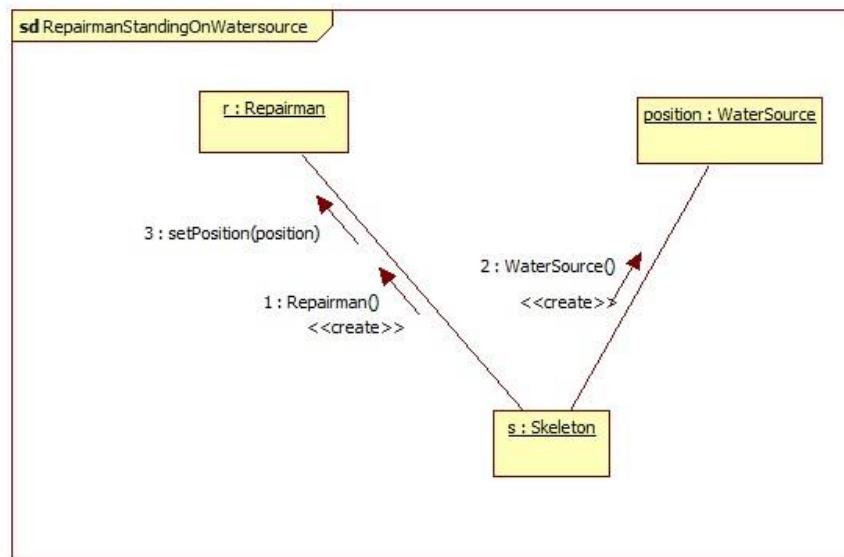


5.4 Kommunikációs diagramok

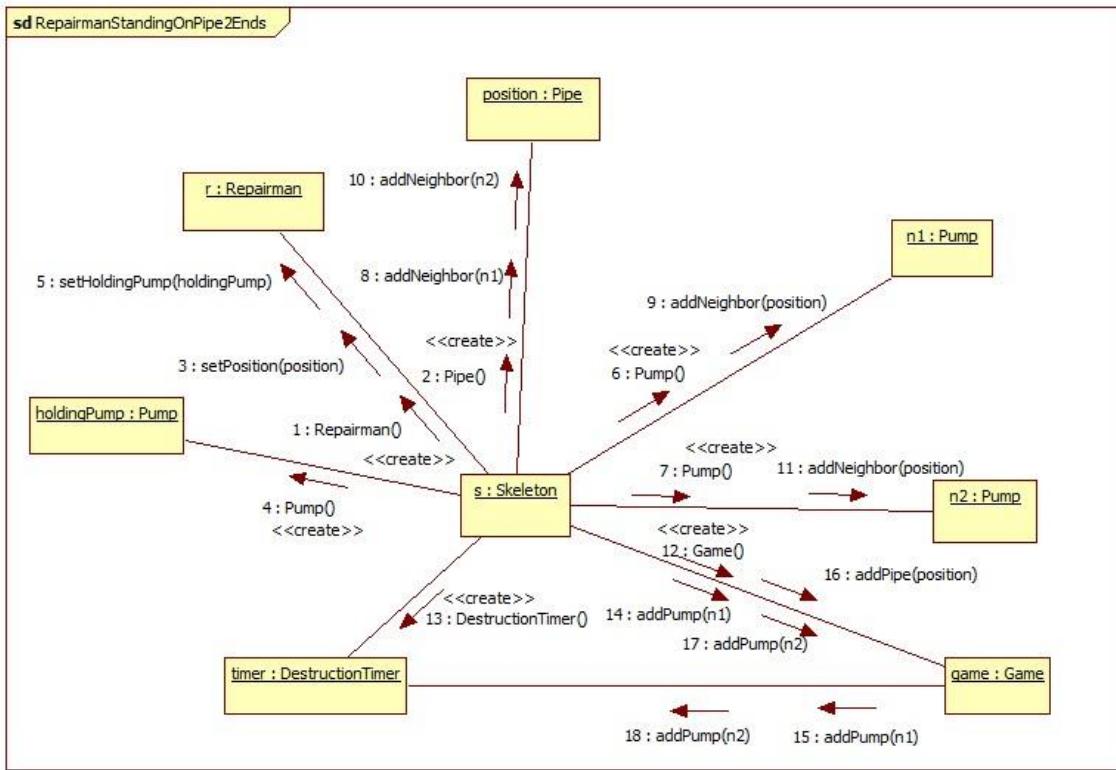
5.4.1 RepairmanStandingOnCistern



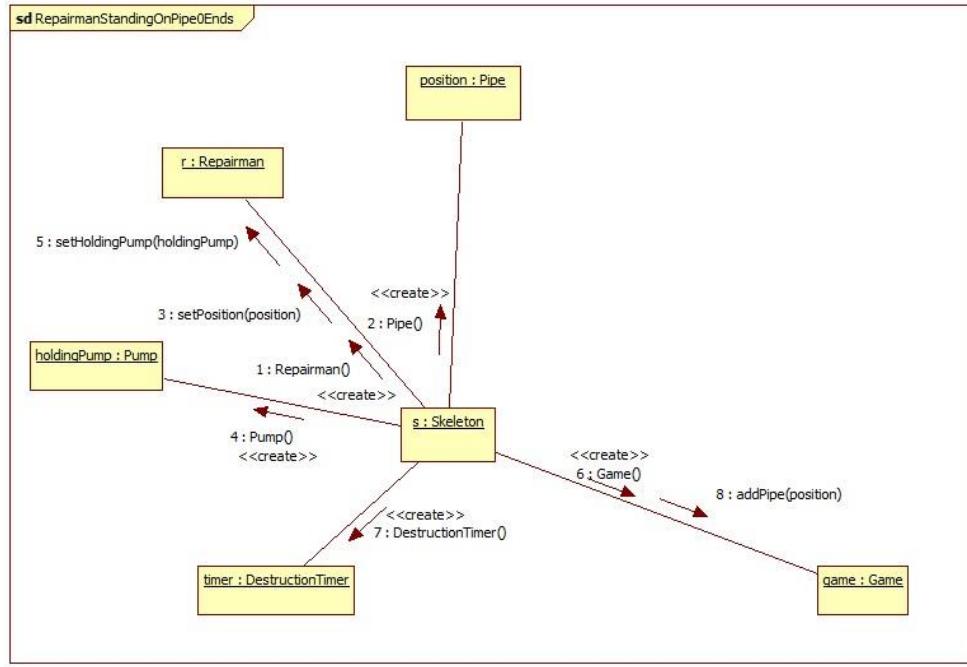
5.4.2 RepairmanStandingOnWaterSoruce



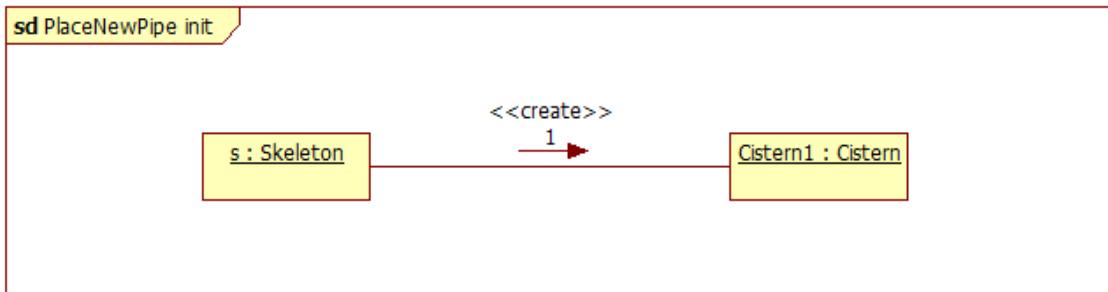
5.4.4 RepairmanStandingOnPipe2Ends



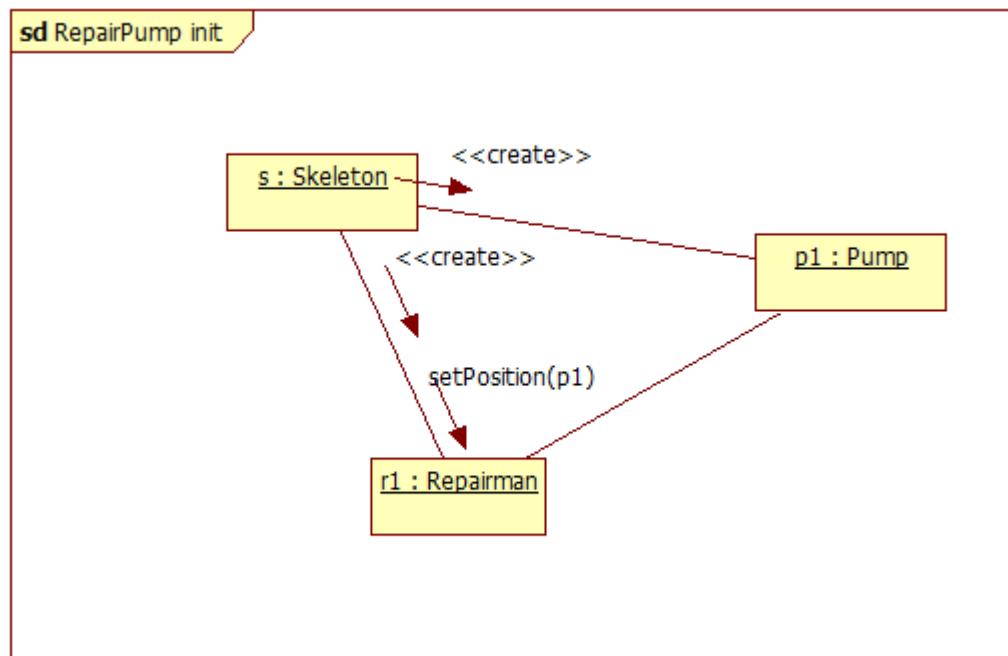
5.4.5 RepairmanStandingOnPipe0Ends



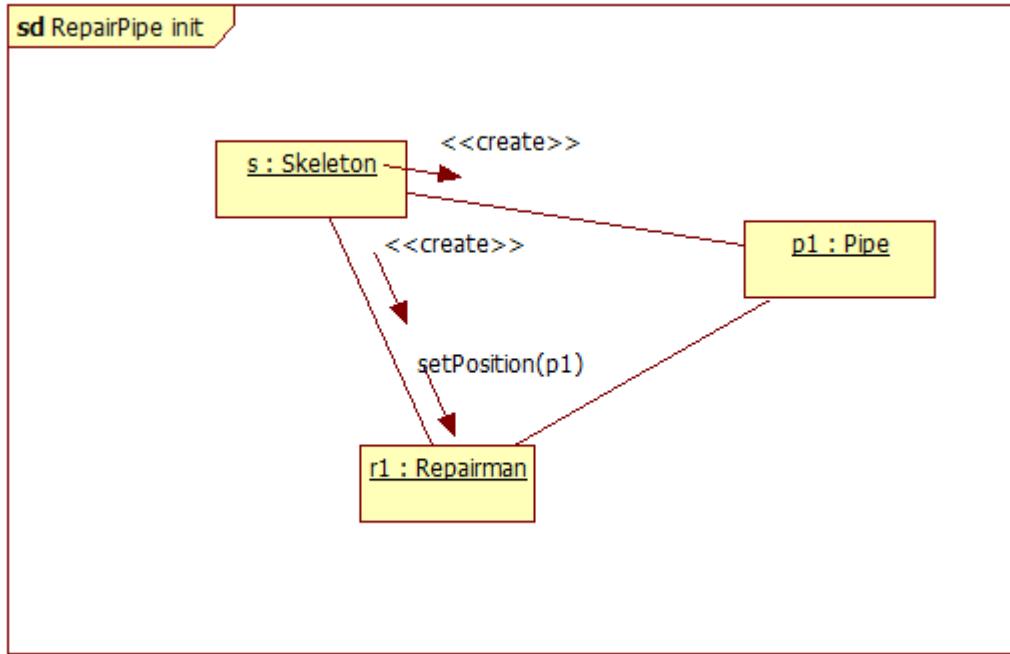
5.4.6 PlaceNewPipe init



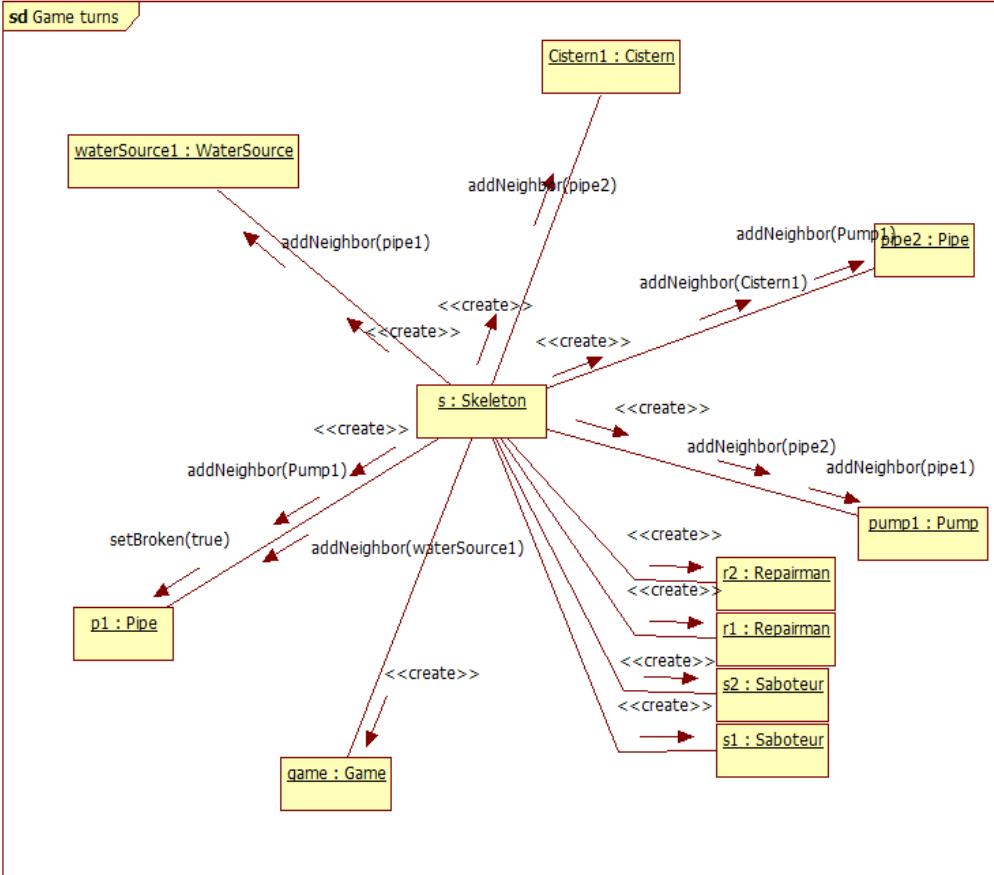
5.4.7 Repair Pump init



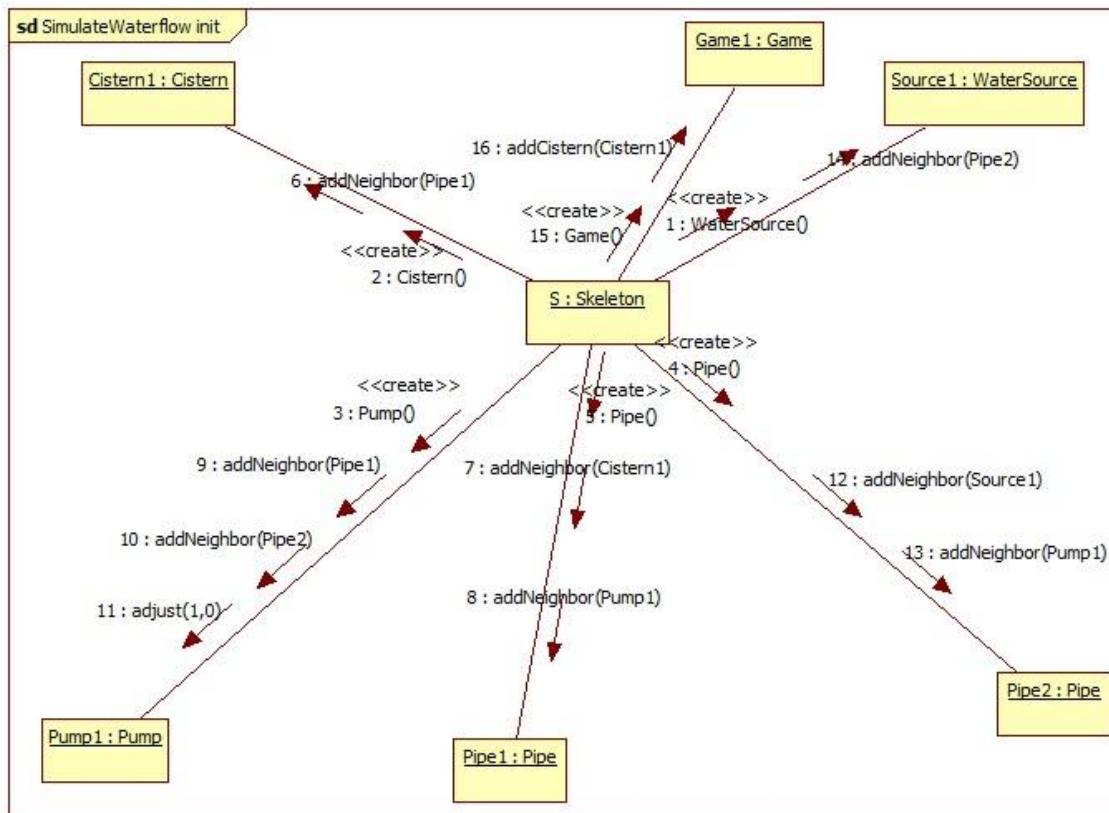
5.4.8 Repair Pipe init



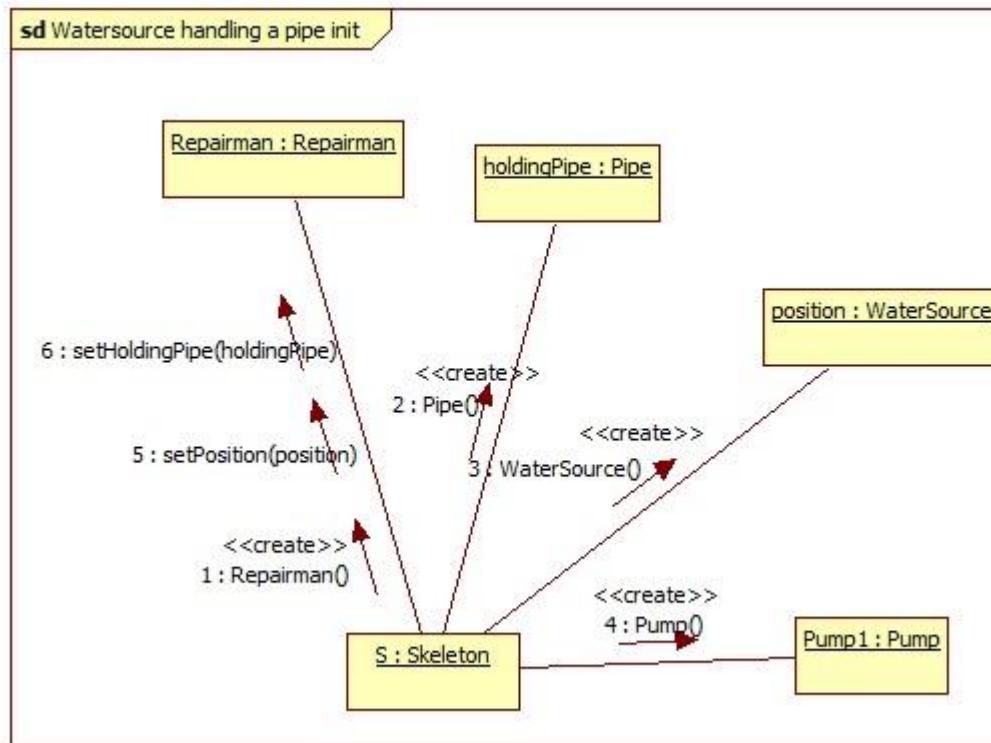
5.4.9 Game Turns



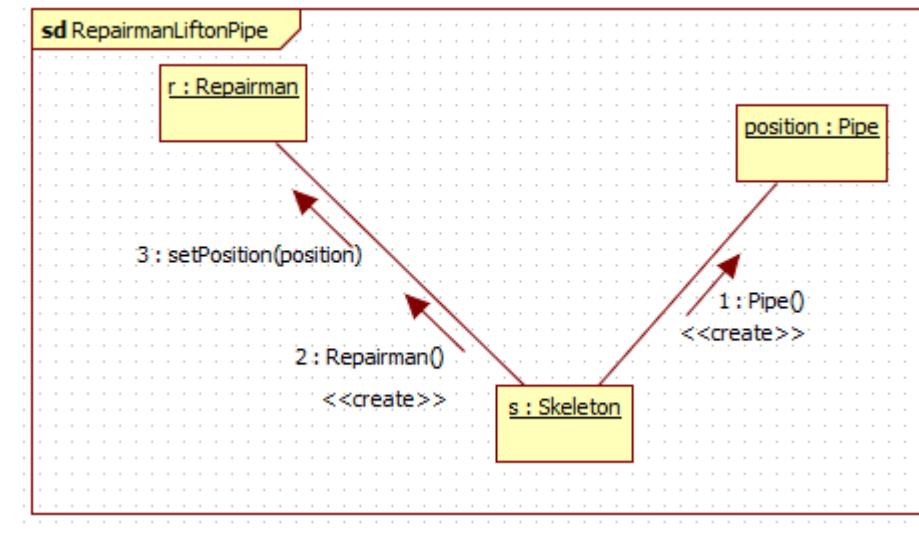
5.4.10 SimulateWaterflow init



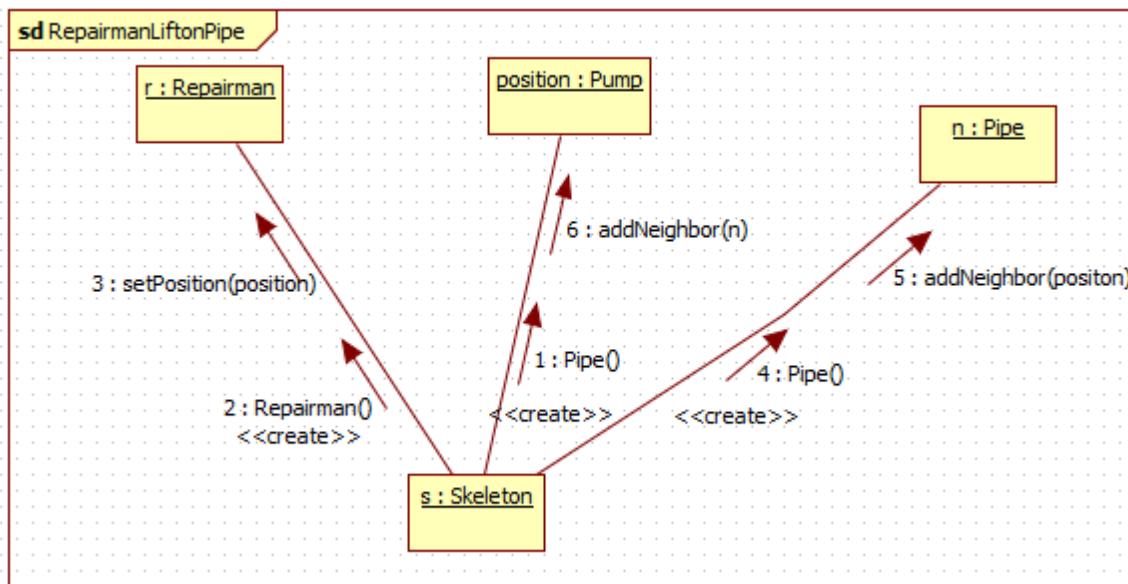
5.4.11 Watersource handling a pipe init



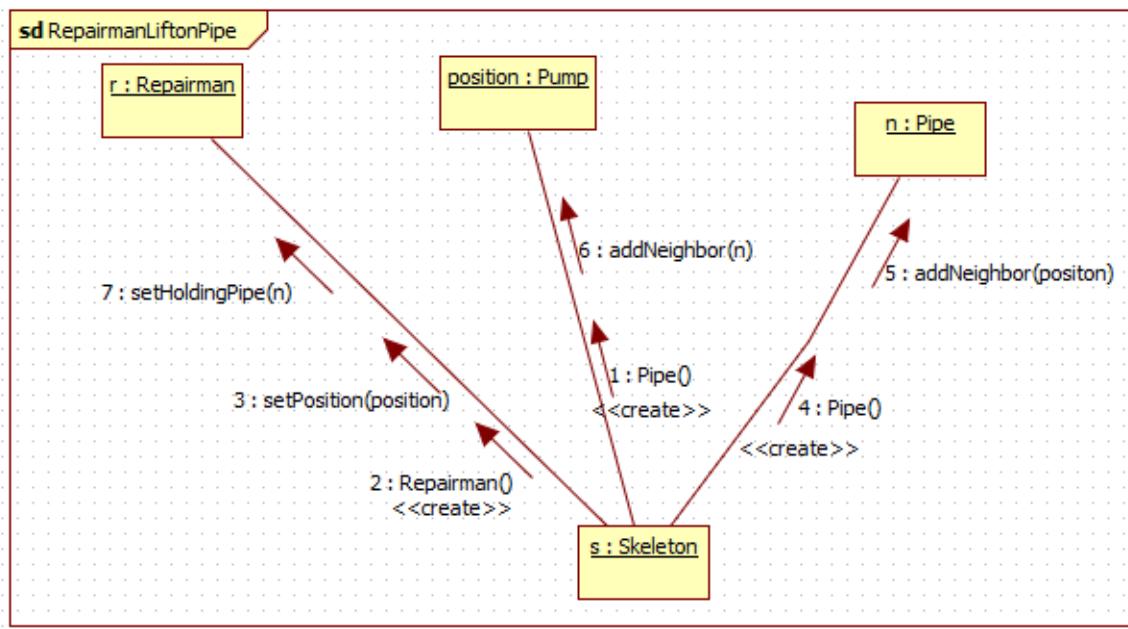
5.4.12.1 Repairman Lift on Pipe



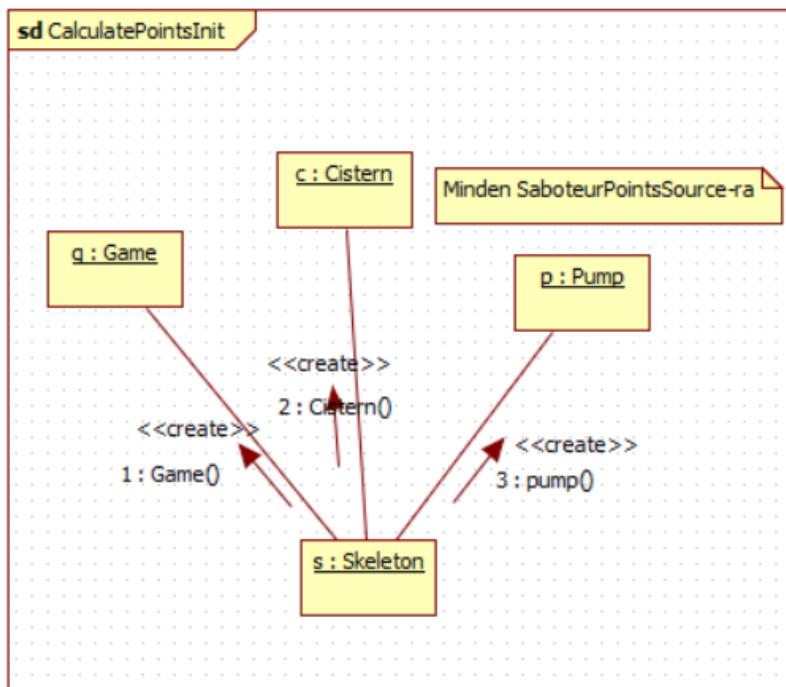
5.4.12.2 Repairman Lift on pump



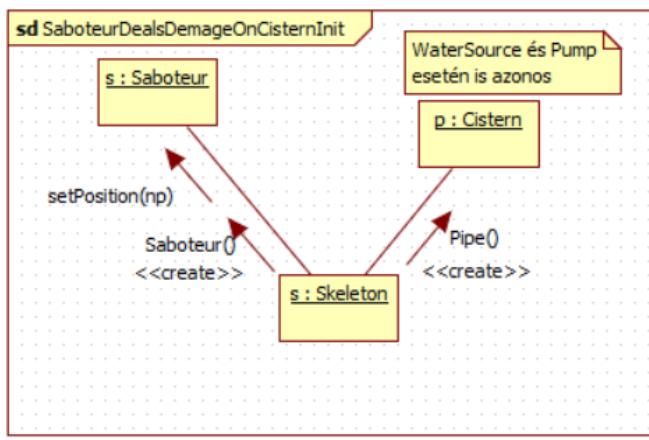
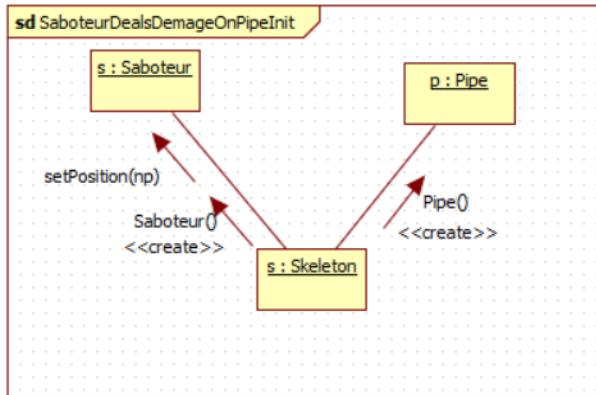
5.4.13 Repairman placing pipe



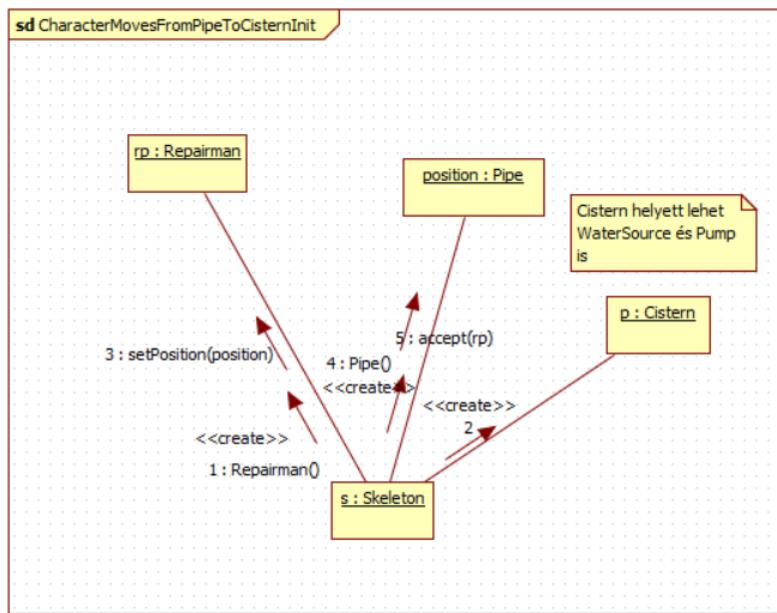
5.4.15 Calculate Points init



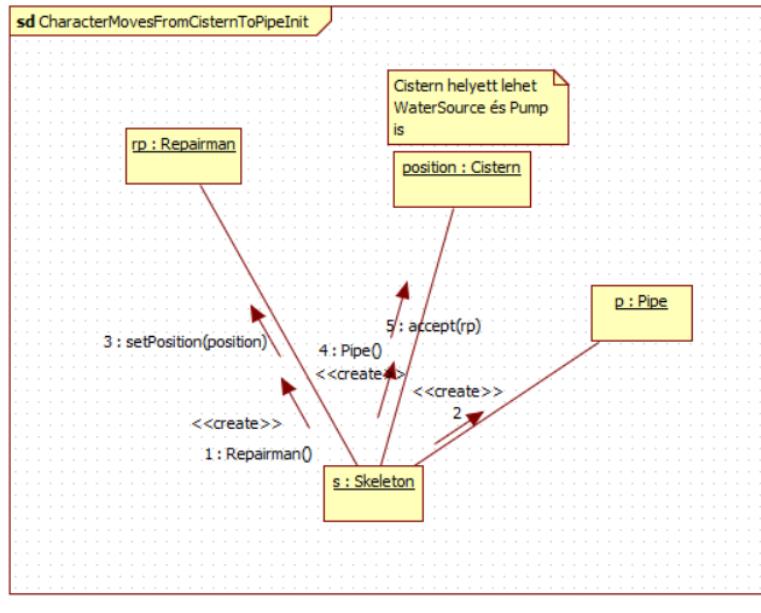
5.4.16 SaboteurDealsDamageOnPipeInit



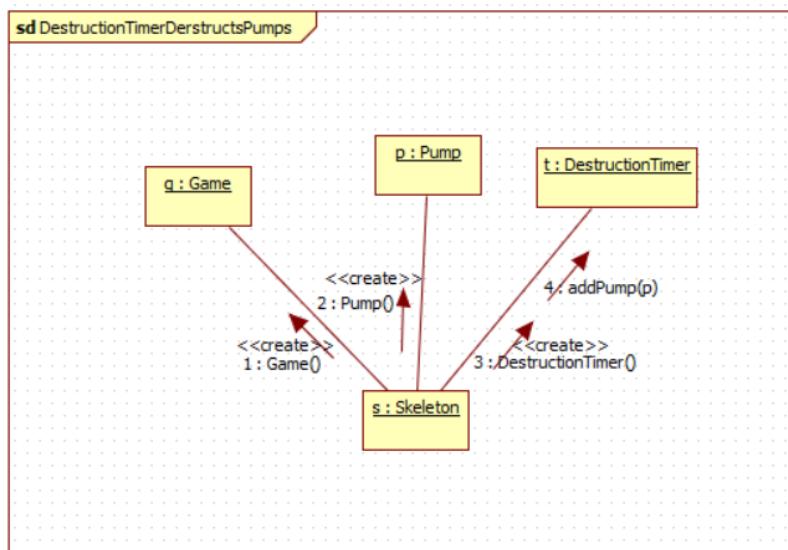
5.4.17.1 CharacterMovesFromPipeToCisternInit



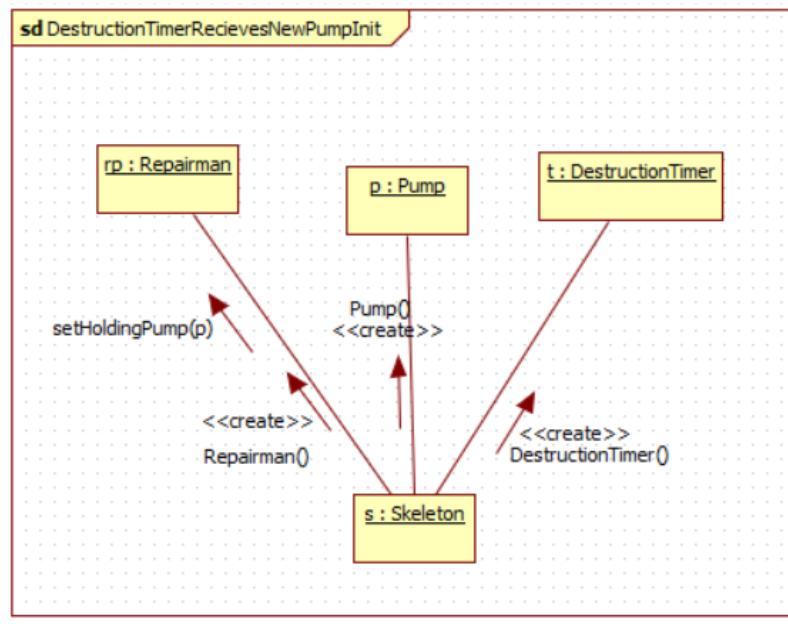
5.4.17.2 CharacterMovesFromCisternToPipeInit



5.4.18 DestructionTimerDestructsPumpsInit



5.4.19 DestructionTimerRecievesNewPumpInit



5.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2023.03.31 9:00	1 óra	Andai, Bodnár Gutási, Sziksai, Vajda	Értekezlet. Döntés: - Andai feldolgozza az előző heti szekvencia diagramból a 3.4.1-3.4.5 diagramokat - Vajda: 3.4.6-3.4.10 - Sziksai: 3.4.12-3.4.15 - Gutási: 3.4.16-3.4.20 - Bodnár: 3.4.21-3.4.25 - Gutási elkészíti az 5.2 szakaszt.
2023.03.31 10:00	1 óra	Vajda	A korábbi 3.4.6-3.4.10 szekvencia diagrammokhoz a use-casek elkészítése
2023.03.31 19:30	1 óra	Gutási	3.4.16-3.4.20 szekvenciadiagram feldolgozása, azokhoz tartozó use case-ek megírása.
2023.03.31 22:00	1 óra	Sziksai	A 3.2.12, 3.4.13, 3.4.14, és a 3.4.15-ös szekvencia diagramokhoz tartozó use case-ek létrehozása.
2023.03.31 22:00	1 óra	Andai	A 3.4.1-3.4.5 korábbi szekvenciadiagramhoz kapcsolódó use-casek elkészítése
2023.04.01 10:00	1 óra	Andai, Bodnár Gutási, Sziksai, Vajda	Értekezlet. Döntés: - Andai elkészíti a use-case diagramot

			- A többiek elkészítik az általuk megírt use case-ekhez tartozó kommunikációs és szekvencia diagramokat.
2023.04.01 11:00	1 óra	Andai	5.3.1-5.3.5 szekvencia diagramok elkészítése
2023.04.01 11:00	1 óra	Bodnár	A korábbi 3.4.21-3.4.24 szekvenciadiagramokhoz tartozó use-casek elkészítése
2023.04.01 16:30	1 óra	Gutási	5.3.7-5.3.9 és 5.4.7-5.4.9 diagramok elkészítése
2023.04.01 17:30	1.5 óra	Andai	5.4.1-5.4.5 kommunikációs diagramok elkészítése, 5.1.1 use-case diagram elkészítése
2023.04.01 23:00	1,5 óra	Szikszai	5.3.10-5.3.11 szekvencia diagramok elkészítése
2023.04.02 13:00	1,5 óra	Bodnár	5.3.13-5.3.29 szekvencia diagramok elkészítése
2023.04.02 13:00	3 óra	Vajda	5.4.12-5.4.14 Szekvenciadiagramok és ehhez tartozó kommunikációs diagrammok elkészítése
2023.04.02. 17:30	3 óra	Szikszai	5.4.16 kommunikációs diagram elkészítése. 5.3.10-11 szekvencia diagramok javítása.
2023.04.02 21:00	1 óra	Andai	5.1.2.1-5.1.2.5 újragondolása, 5.1.1 módosítása a use-caseknek

			megfelelően
2023.04.02. 19:30	2 óra	Bodnár	5.4.13-5.4.19 kommunikációs diagramok elkészítése. Szekvencia diagramok javítása.
2023.04.02 20:00	1 óra	Gutási	5.2 szakasz megírása, 5.4.6-5.4.9 diagramok javítása
2023.04.03 7:30	1,5 óra	Gutási	Dokumentum számozásának és formázásának javítása, előkészítés a nyomtatásra.

2023.04.17

6. Szkeleton beadás

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
App.java	164 B	2023.04.16. 10:22	Az alkalmazás indítása, leállítása
Character.java	353 B	2023.04.17 7:12	Character osztály
Cistern.java	3845 B	2023.04.17 7:12	Cistern osztály
DestructionTimer.java	1243 B	2023.04.17 7:12	DsetructionTimer osztály
Element.java	2664 B	2023.04.16 15:17	Element osztály
Game.java	6 969 B	2023.04.17 7:12	Game osztály
NonPipe.java	1912 B	2023.04.16 21:19	NonPipe osztály
Pair.java	314 B	2023.04.16 10:22	Pair osztály. Különböző típusú objektumok párokban történő tárolására szolgál.
Pipe.java	9099 B	2023.04.16 21:19	Pipe osztály
Place.java	511 B	2023.04.17 7:12	Place osztály
Pump.java	4363 B	2023.04.17 7:12	Pump osztály
Repairman.java	5984 B	2023.04.17 7:12	Repairman osztály
RepairmanPlace.java	493 B	2023.04.16 12:38	RepairmanPlace osztály
Saboteur.java	2136 B	2023.04.17 7:12	Saboteur osztály
SaboteurPlace.java	123 B	2023.04.16 10:22	SaboteurPlace osztály
SabouteurPointSource.java	191 B	2023.04.16 10:22	SaboteurPointSource osztály
Skeleton.java	18983 B	2023.04.17 7:50	Skeleton osztály, mely tartalmazza a teszteket, és a tesztek kiválasztását szolgáló menüt
Tabulator.java	333 B	2023.04.16 16:03	Tabulator osztály, mely a tesztek kiírását segíti.
WaterSource.java	2320 B	2023.04.17 7:12	WaterSource osztály

6.1.2 Fordítás

A forrásfájlok fordításához a forrásfájlokat tartalmazó mappában, a `javac App.java` parancsot kell kiadni. Ehhez szükséges, hogy a fordító gépen telepítve legyen Java 8-hoz tartozó fejlesztői csomag (JDK).

6.1.3 Futtatás

A program futtatásához a lefordított .class fájlokat tartalmazó mappában, használjuk a `java App` parancsot. Ennek hatására a program elindul.

6.2 Értékelés

A csapat tagjai aláírásukkal igazolják, hogy az alábbi táblázatban szereplő arányokat ellenőrizték és elfogadták.

Tag neve	Tag neptun	Munka százalékban	Aláírás
Andai Roland	H3G1PQ	19.45%	
Bodnár Márk	AAL25F	17.21%	
Gutási Ádám	XW53QZ	20.2%	
Szikszai Levente	AENWDT	24.19%	
Vajda András	M7LWCB	18.95%	

6.3 Napló

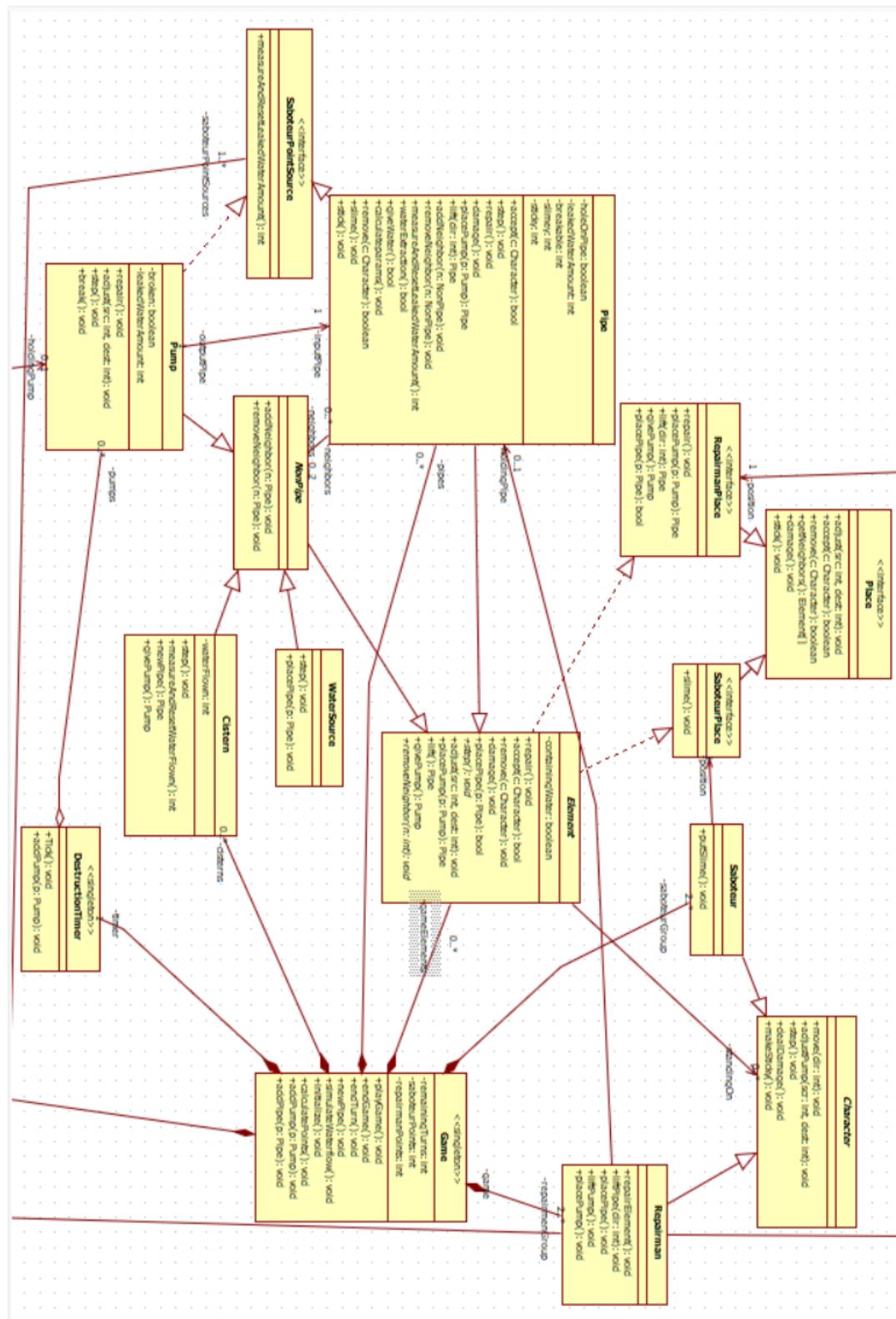
Kezdet	Időtartam	Résztvevők	Leírás
2023.04.12 10:00	1 óra	Andai, Bodnár Gutási, Sziksai	Értekezlet. Döntés: Andai: Element osztály és minden további osztály létrehozása, ami ezen osztályhoz kapcsolódik. Bodnár: Character osztály és minden további osztály létrehozása, ami ezen osztályhoz kapcsolódik. Gutási: Skeleton osztály létrehozása, tesztek menüpárok létrehozása Sziksai: Game, DestructionTimer és minden további osztály létrehozása
2023.04.14 7:30	3 óra	Andai	Element és a hozzá kapcsolódó további 9 osztály deklarálása és implementálása
2023.04.14 10:00	1,5 óra	Bodnár	Character, Repairman, Saboteur osztályok deklarálása és megvalósítása.
2023.04.14. 13:30	1,5 óra	Sziksai	A Game osztály megvalósítása, a DestructionTimer deklarálása és megvalósítása
2023.04.14 18:00	1.5 óra	Andai	Elementhez kapcsolódó osztályokhoz Javadoc kommentek készítése
2023.04.15 10:00	2 óra	Bodnár	Character, Repairman, Saboteur osztályokon módosítások. Element, Pipe osztály bővítése.
2023.04.15. 14:00	1 óra	Sziksai	SimulateWaterflow és WaterSourceHandling API tesztek elkészítése
2023.04.16	1.5 óra	Andai, Bodnár Gutási, Sziksai	Értekezlet. Elkészül osztályok közös átnézése, tesztek átbeszélése. Döntés: Andai: elkészíti az 5.3.1-5.3.5 teszteket Bodnár: Elkészíti az 5.3.16-5.3.20 teszteket Gutási: Elkészíti az 5.3.6-5.3.8 és 5.3.13 teszteket Sziksai: elkészíti az 5.3.9-5.3.11 teszteket. Vajda: elkészíti az 5.3.12-5.3.15 teszteket
2023.04.16 12:30	2 óra	Gutási	5.3.6-5.3.8, és 5.3.13 tesztek megírása
2023.04.16 12:30	1.5 óra	Andai	5.3.1-5.3.5 tesztek megírása

2023.04.16 12:30	5 óra	Szikszai	SimulateWaterflow és WaterSourceHandlingAPipe tesztek véglegesítése és hozzájuk tartozó osztályok módosítása
2023.04.16 15:00	3 óra	Vajda	5.3.12-5.3.15 tesztek megírása
2023.04.16 22:00	3 óra	Bodnár	5.3.16-5.3.20 tesztek megírása, javítása, hozzá kapcsolódó függvények módosítása.
2023.04.17. 7:00	2 óra	Gutási	Kód javítása, elírások kijavítása, projekt összerendezése beadása, adminisztráció

7. Prototípus koncepciója

7.0 Változás hatása a modellre

7.0.1 Módosult osztálydiagram



7.0.2 Új vagy megváltozó metódusok

void putSlime(): A Saboteur osztály része, csúszóssá teszi az adott csövet amin éppen áll

boolean accept(Character c): A változás, hogy ha a cső csúszós akkor tovább- vagy visszadobja a rálépő játékosokat

void damage(): A SaboteurPlace interfészről átkerült a Place interfészbe

dealDamage(): Változás, hogy a Repairman is meghívhatja, illetve csak akkor lyukasztja ki az adott ha ki lehet

void Slime(): Beállítja a slimey változót egy értékre

void Sticky(): Beállítja a sticky változót egy értékre

void setBreakable(): Beállítja a breakable változót egy értékre

int getSlimey(): Visszaadja a slimey változó értékét

int getSticky(): Visszaadja a sticky változó értékét

int getBreakable(): Visszaadja a breakable változó értékét

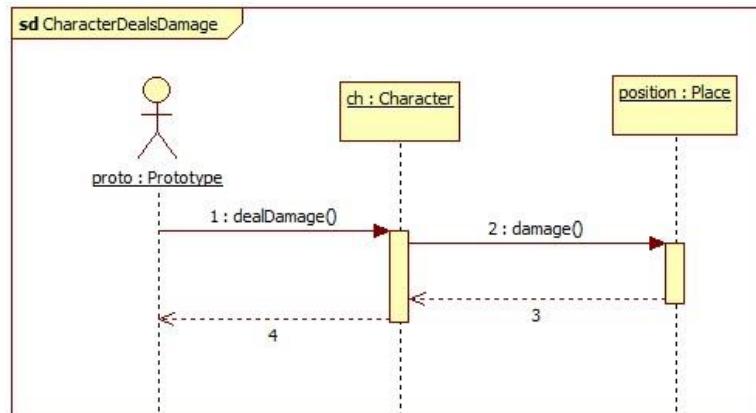
void calculateParams(): Amennyiben nem 0 akkor csökkenti a slimey és breakable attribútumokat. Ha valaki áll egy sticky csövön és nem 0, akkor a sticky értékét is csökkenti eggyel.

boolean remove(Character c): Változás, hogy csak akkor engedi lelépni a karaktert amennyiben a sticky attribútum 0, és visszatér a lelépés sikerességével.

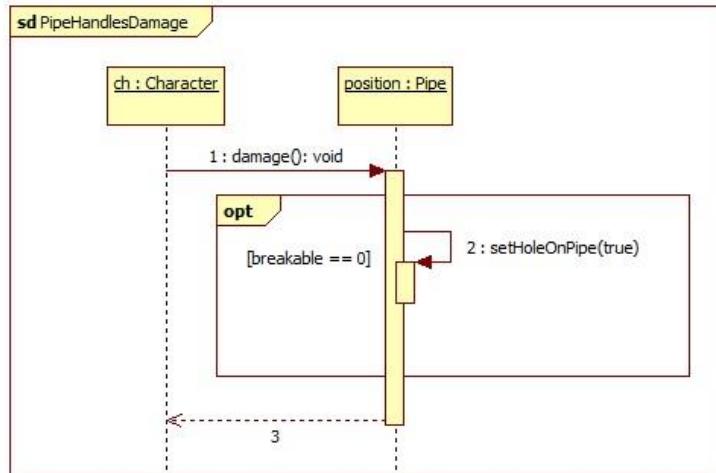
void makeSticky(): Ragadóssá teszi az adott csövet amin áll

7.0.3 Szekvencia-diagramok

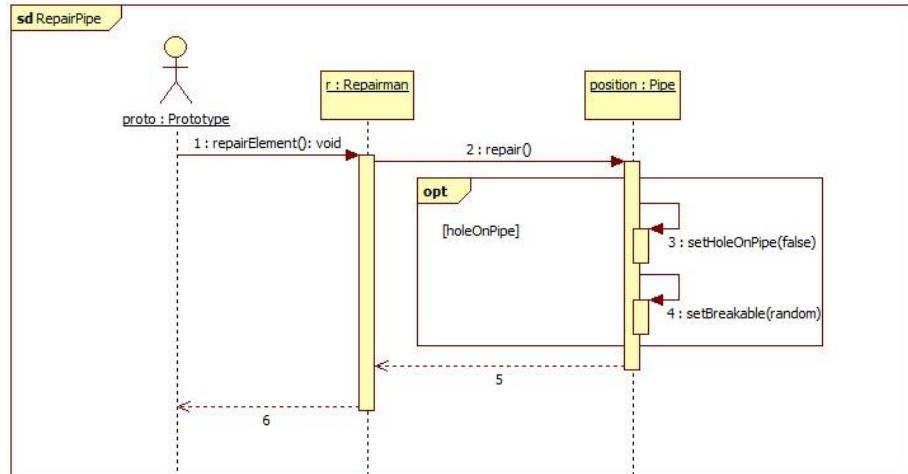
7.0.3.1 Character deals damage



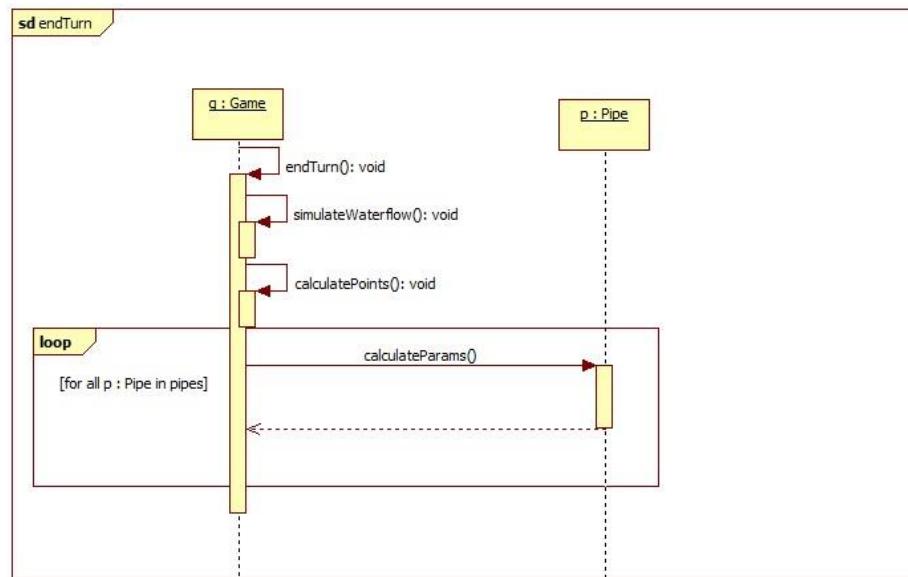
7.0.3.2 Pipe handles damage



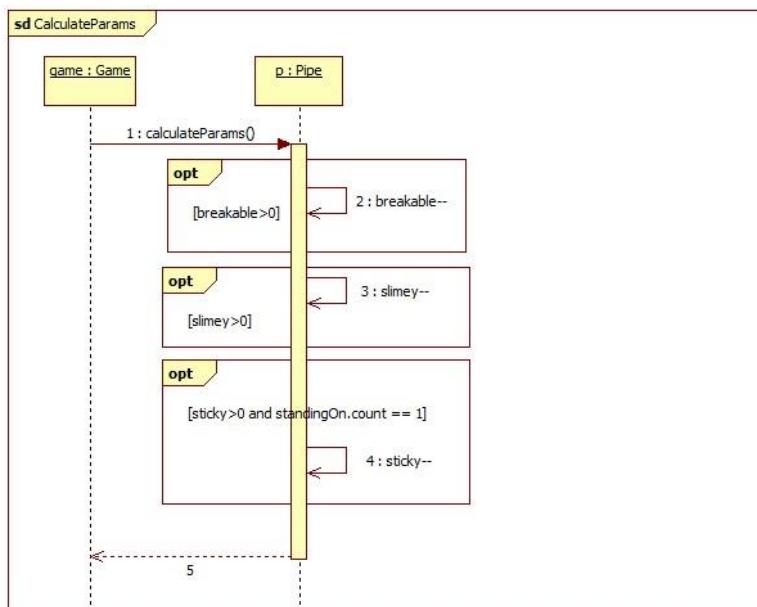
7.0.3.3 Repair pipe



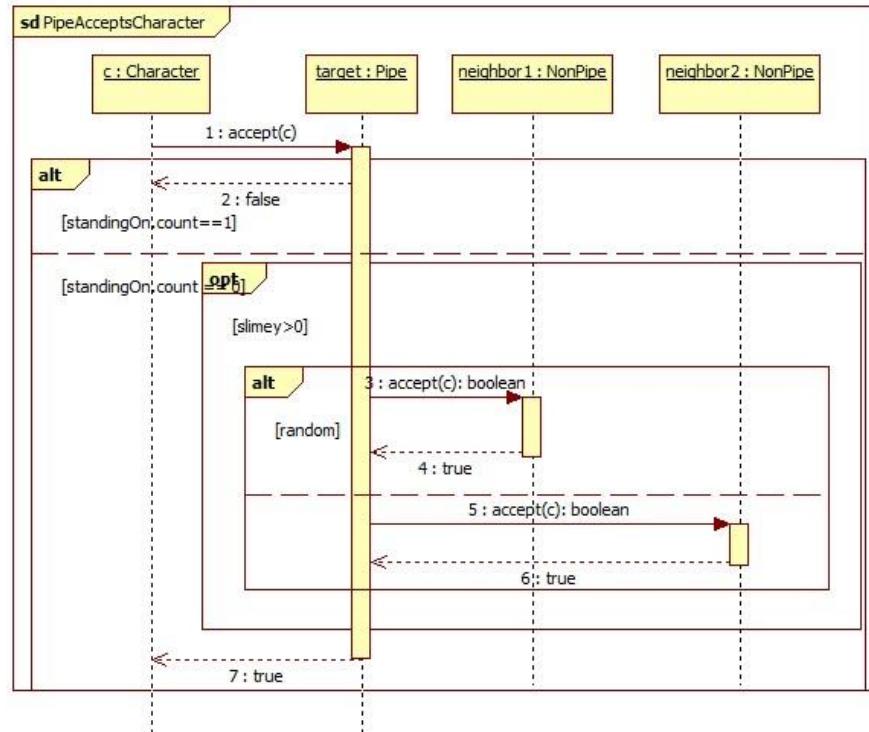
7.0.3.4 End turn



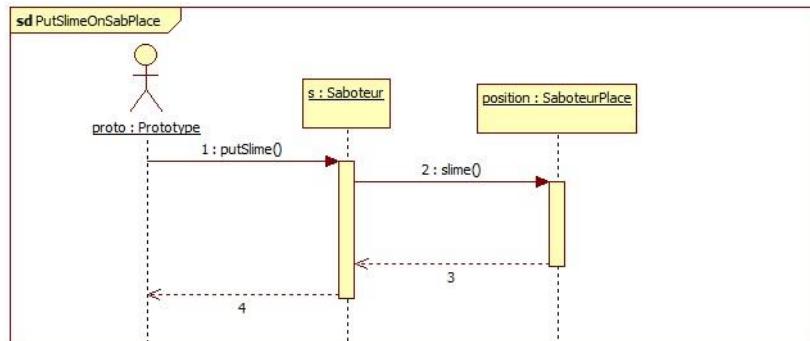
7.0.3.5 Calculate params



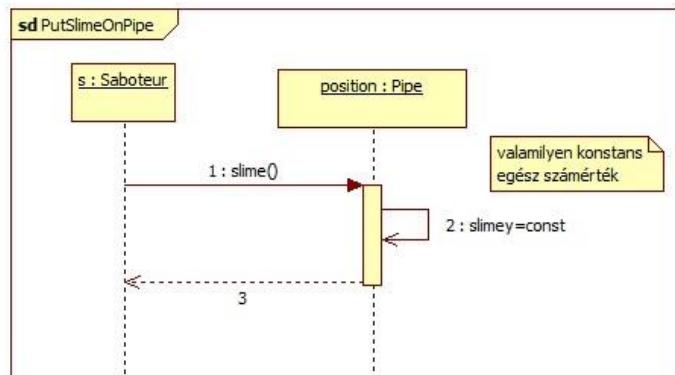
7.0.3.6 Pipe accepts character



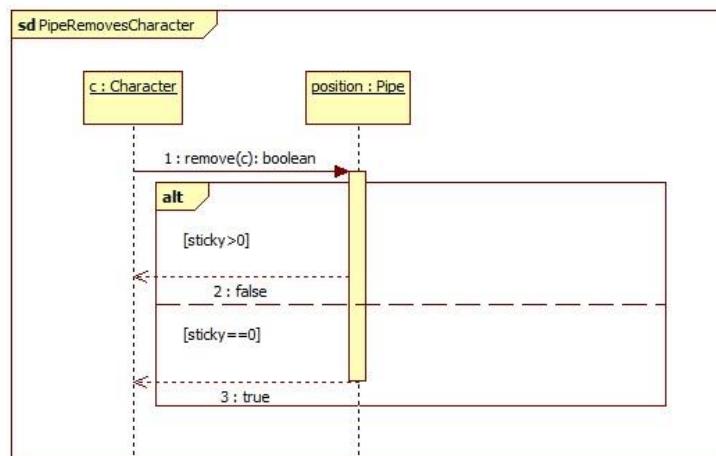
7.0.3.7 Put slime on saboteur place



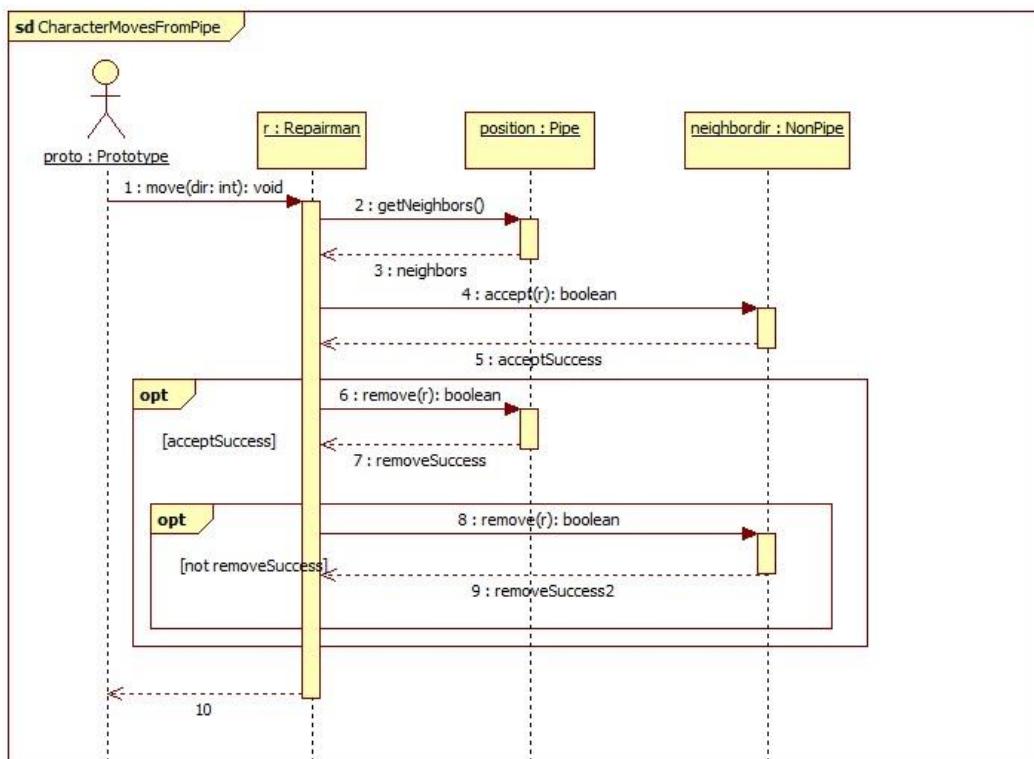
7.0.3.8 Put slime on pipe



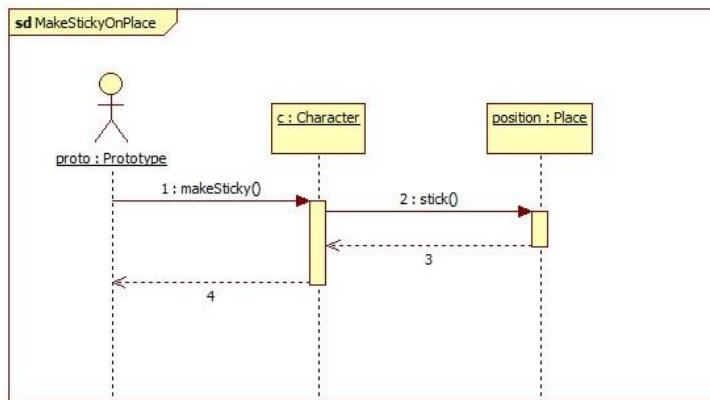
7.0.3.9 Pipe removes character



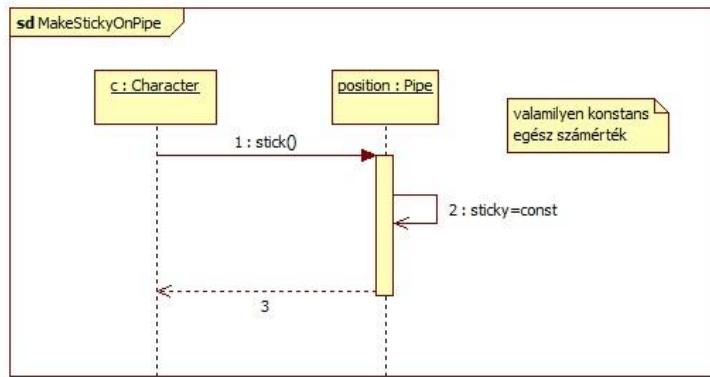
7.0.3.10 Character moves



7.0.3.11 Make sticky on place



7.0.3.12 Make sticky on pipe



7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A parancsokat a szabványos bemenetről tudja fogadni az alkalmazás, a kimenet a szabványos kimeneten jelenik meg. A játék állapota fájlba menthető és onnan visszatölthető parancsok segítségével.

7.1.2 Bemeneti nyelv

Adjust

Leírás: Beállítja a Pumba ki és bemeneti csövét amin áll.

Opciók: *src dest* - ahol *src* a bemeneti Pumba száma és *dest* a kimeneti Pumba száma.

Move

Leírás: A játékos átlép egy elemről egy másikra.

Opciók: *dir* - Melyik objektumra történjen a mozgás, egy pozitív egész szám.

Stab

Leírás: Kilyukasztja a csövet amin áll.

Opciók: Nincs***Repair***

Leírás: Megjavítja a Csövet vagy Pumpát amin áll.

Opciók: Nincs

PickUpPump

Leírás: A Ciszternáról felvesz egy Pumpát.

Opciók: Nincs

PickUpPipe

Leírás: Felemeli a cső egyik végét az adott elemről amin áll.

Opciók: *pipe* - egész szám, ami a Csövet azonosítja amit fel akarunk emelni.

PlacePump

Leírás: Lerakja a Repairman játékosnál lévő Pumpát.

Opciók: Nincs

PlacePipe

Leírás: Lerakja a Repairman játékosnál lévő Csövet.

Opciók: Nincs

Stick

Leírás: Ragadóssá teszi a csövet amin áll.

Opciók: Nincs

Slime

Leírás: Csúszóssá teszi a csövet amin áll.

Opciók: Nincs

EndTurn

Leírás: Befejezi az adott játékos körét, a következő játékos fog következni.

Opciók: Nincs

Save

Leírás: Elmenti az adott játék állapotát.

Opciók: *file* - A mentett *file*.txt fájl neve, mely a pálya és a játékosok állapotát tárolja. A *file*Points.txt pedig a csapatok pontszámát és a hátralévő köröket tárolja.

Load

Leírás: Visszatölti a játék egy mentett állapotát.

Opciók: *file* - Az adott mentéshez tartozó *file*.txt és *file*Points.txt fájlok neve.

Help

Leírás: Kilistázza az adható parancsokat.

Opciók: Nincs

ToggleRandom

Leírás: Be és kikapcsolja a Random eseményeket.

Opciók: **I/N** - I - igen,bekapcsol ; N - nem, kikapcsol

Pályakép:

Type:

C-Cistern

Pu-Pump

Pi-Pipe

W-WaterSource

S-Saboteur

R-Repairman

Params:

WaterSource:

Cistern:

Pump: isBroken containingWater

Pipe: hasHole containingWater slimy sticky

Saboteur: Place

Repairman: Place holdingPipe HoldingPump

Syntax:

Type Name Param1..N //Deklarálás

Name Neighbor1..N //Szomszédok beállítása

Példa:

```
###Declaration###

```

C Cistern

Pu Pump false true

W Source

Pi Pipe1 false true 0 0

Pi Pipe2 false true 1 4

```
###Players###

```

R Repairman Source null null

```
###Neighbors###

```

Cistern Pipe1

Pipe1 Cistern Pump

Pump Pipe1 Pipe2

Pipe2 Pump Source

Source Pipe2

7.1.3 Kimeneti nyelv

Adjust

- Input set **Object1**, Output set **Object2**
- Invalid parameters, nothing changed
- Object** is not Adjustable
- Out of Action

Move

- Successfully moved to **Object**
- Failed to move to **Object**
- Sticked to **Object**, can't move until: **Turns**
- Slipped to **Object**
- Out of Action

Stab

- Successfully stabbed **Object**
- Failed to stab **Object**, unbreakable until: **Turns**
- Failed to stab **Object**, already has hole
- Failed to stab, **Object** is unbreakable
- Out of Action

Repair

- Successfully repaired **Object**
- Object** unrepairable
- Object** not broken
- Out of Action
- Not valid action for Saboteur

PickUpPump

- Can't pick up Pump here
- Successfully picked up **Object**
- Out of Action
- Not valid action for Saboteur

PickUpPipe

- Invalid Pipe to pick up
- Successfully picked up **Object**
- Out of Action
- Not valid action for Saboteur

PlacePump

- No placeable Pump
- Successfully placed **Object**
- Can't place **Object1** on **Object2**
- Out of Action
- Not valid action for Saboteur

PlacePipe

- No placeable Pipe
- Successfully placed **Object**
- Can't place **Object1** on **Object2**
- Out of Action
- Not valid action for Saboteur

Stick

- Object** is now sticky
- Object** can't be sticky
- Out of Action

Slime

- Object** is now slimey
- Object** can't be slimey
- Out of Action
- Not valid action for Repairman

EndTurn

- Player1 turn ended Player2's turn next

Load

- Load failed
- Load successful

Help

-Move dir, Adjust src dest, Repair, Stab, PickUpPump, PickUpPipe pipe, PlacePump, PlacePipe, Stick, EndTurn, Save file, Load file, Help, ToggleRandom I/N

-Move dir, Adjust src dest, Stab, Stick, Slime, EndTurn, Save file, Load file, Help, ToggleRandom I/N

ToggleRandom

- Random events Enabled
- Random events Disabled
- Invalid input

Az **Object**, **Player** neveket, a **Turns** pedig számot jelöl.

7.2 Összes részletes use-case

Az alábbi use case-ek esetében **vastag** betűvel vannak szedve a felhasználói felületen kiadott parancsok, és “*idézőjelben, döntött betűvel*” szedve a felület kimenetei.

7.2.1

Use-case neve	Repairman standing on Cistern and lifts pump
Rövid leírás	A szerelő felvesz egy pumpát.
Aktorok	Prototype
Forgatókönyv	1.1 A szerelő kér a ciszternától egy pumpát. (PickUpPipe) 1.1.2 A ciszterna létrehoz egy új pumpát.

	1.1.3 A szerelő sikeresen megkapja a pumpát a ciszternától, a kimenet szövege: “Successfully picked up <i>Object</i> ”, ahol Object a pumpa.
--	---

7.2.2

Use-case neve	Repairman cannot lift pump
Rövid leírás	A szerelő nem kap pumpát.
Aktorok	Prototype
Forgatókönyv	1.1 A szerelő egy olyan objektumtól kér pumpát, ami nem tud neki azt adni. (PickUpPump) 1.2 A szerelő nem kap pumpát, a kimenet szövege: “Can’t pick up Pump here”

7.2.3

Use-case neve	Repairman cannot place pump
Rövid leírás	A szerelő nem tudja elhelyezni a pumpát.
Aktorok	Prototype
Forgatókönyv	1.1 A szerelő egy olyan objektumon szeretné elhelyezni a pumpát (PlacePump), ahová nem tudja. 1.2 Sikertelen a letétel, a felület a következő szöveget írja ki: “Can’t place <i>Object1</i> on <i>Object2</i> “ ahol <i>Object1</i> a pumpa, <i>Object2</i> az objektum, amin nem lehet elhelyezni.

7.2.4

Use-case neve	Repairman places pump
Rövid leírás	A szerelő sikeresen lehelyezi a pumpát.
Aktorok	Prototype
Forgatókönyv	1.1 A játékos kiadja a PlacePump parancsot. 1.2.A ha a szerelőnél nincs pumpa akkor a művelet sikertelen (“No placeable Pump” kiírása) 1.2.B A szerelő lecsatlakoztatja a cső egyik végén lévő pumpát a csőről. 1.3.B A cső ezen végére beköti a szerelő a nála lévő pumpát. 1.4.B A szerelő létrehoz egy új csövet, majd ezt beköti a letett pumpába. 1.5.B A másik felét a csőnek beköti a lecsatlakoztatott pumpába. 1.6.B A szerelő regisztrálja a játékba az új csövet és a pumpát. 1.7.B “Successfully placed <i>Object</i> ” kiírása.

7.2.5

Use-case neve	Repairman cannot place pump on a zero-end-pipe
Rövid leírás	A szerelő nem tudja lehelyezni a pumpát egy olyan csövön, amely sehol sem nincs bekötve.
Aktorok	Prototype
Forgatókönyv	1.1 A szerelő megpróbálja lehelyezni a nála lévő pumpát a csövön. (PlacePump)

	1.2 Nem sikerül a letétel. (“Can’t place <i>Object1</i> on <i>Object2</i> ”)
--	--

7.2.6

Use-case neve	PlaceNewPipe
Rövid leírás	A Controller új csövet hoz létre egy ciszternán.
Aktorok	Prototype
Forgatókönyv	<p>1.1 A Controller szól a Game-nek, hogy hozzon létre új csövet</p> <p>1.2 A Game szól a ciszternának, hogy hozzon létre egy új csövet.</p> <p>1.3.1 A ciszterna létrehoz egy új csövet</p> <p>1.3.2 Az új csőhöz hozzáadja megát a ciszterna, mint szomszéd.</p> <p>1.4.1 A ciszterna visszaadja a cső hivatkozását a Game-nek.</p> <p>1.4.2 A Game eltárolja a cső referenciáját</p>

7.2.7

Use-case neve	Repair Pump
Rövid leírás	A szerelő karakter megpróbál megszerelni egy pumpát.
Aktorok	Prototype
Forgatókönyv	<p>1.1 A játékos szól a karakternek, hogy kísérélje meg megszerelni azt a pumpát, amin áll. (Repair parancs kiadása)</p> <p>1.2 A játékos által irányított karakter megkísérli megjavítani a pumpát, amin áll.</p> <p>1.2.1 A tesztelő eldönti, hogy el volt-e romolva a pumpa.</p> <p>1.3.A Ha a pumpa, amin áll, el volt romolva, akkor megjavul. (“Successfully repaired <i>Object</i>” kiírása)</p> <p>1.3.B Ha a pumpa, amin áll, nem volt elromolva, akkor semmi sem történik. (“<i>Object</i> not broken” kiírása)</p>

7.2.8

Use-case neve	Repair Pipe
Rövid leírás	A szerelő karakter megpróbál megszerelni egy csövet.
Aktorok	Prototype
Forgatókönyv	<p>1.1 A játékos szól a karakternek, hogy kísérélje meg megszerelni azt a csövet, amin áll.</p> <p>1.2 A játékos által irányított karakter megkísérli megjavítani a csövet, amin áll.</p> <p>1.3.A Ha a cső, amin áll, el volt romolva, akkor megjavul. (“Successfully repaired <i>Object</i>” kiírása)</p> <p>1.3.B Ha a cső, amin áll, nem volt elromolva, akkor semmi sem történik. (“<i>Object</i> not broken” kiírása)</p>

7.2.9

Use-case neve	Game Turns
Rövid leírás	A játék elindulását, destructionTimer működését, karakterek lépéssének sorrendjét és a játék befejeződését ellenőrző teszt.
Aktorok	Prototype
Forgatókönyv	1.0 A tesztelő megadja, hány körös legyen a játék.

	<p>1.1 A Controller szól a Game osztálynak, hogy a játék elindul.</p> <p>1.2.A Amíg a hátramaradt köröket számláló értéke nem éri el a nullát, addig minden kör elején a Game lépteti a destructionTimert.</p> <p>1.2.3 destructionTimer minden pumpára eldönti, hogy elromoljon-e az adott körben.</p> <p>1.2.3.A Ha a destructionTimer egy adott pumpát tönkretesz, jelzi azt a tesztelő számára.</p> <p>1.2.4 Továbbá minden körben minden játékos léphet:</p> <p>1.2.4.1 Először a szabotőr csapat karakterei léphetnek:</p> <p>1.2.4.1.1 Kiválasztjuk a tesztnél, hogy melyik karakter léphet.</p> <p>1.2.4.1.2.A Game szól az adott játékos által irányított karaktereknek, hogy léphet.</p> <p>1.2.4.1.3.A Amíg a tesztelő nem írja be az EndTurn parancsot, addig az adott karakter van soron.</p> <p>1.2.4.1.4 Ha a karakter beírta az EndTurn parancsot, akkor a következő játékos jöhét. (“<i>Player1</i> turn ended <i>Player2</i>’s turn next” kiírása)</p> <p>1.2.4.2 Ezután a szerelő csapat karakterei léphetnek, egymás után.</p> <p>1.2.4.2.1 Először a szabotőr csapat karakterei léphetnek:</p> <p>1.2.4.2.1 Kiválasztjuk a tesztnél, hogy melyik karakter léphet.</p> <p>1.2.4.2.2.A Game szól az adott játékos által irányított karaktereknek, hogy léphet.</p> <p>1.2.4.2.3.A Amíg a tesztelő nem írja be az EndTurn parancsot, addig az adott karakter van soron.</p> <p>1.2.4.2.4 Ha a karakter beírta az EndTurn parancsot, akkor a következő játékos jöhét. (“<i>Player1</i> turn ended <i>Player2</i>’s turn next” kiírása)</p> <p>1.3.A Ha a hátramaradt köröket számláló értéke eléri a nullát:</p> <p>1.3.A.1 Game meghívja önmagán a játékot befejező metódust.</p> <p>1.3.A.2 Megjelenik a két csapat által elért pontszám, ezek alapján eldöntődik, hogy ki nyert.</p> <p>1.3.A.3 játék véget ér.</p>

7.2.10

Use-case neve	SimulateWaterflow(Start, Mid, End)
Rövid leírás	A víz folyik a Forrástól a Ciszternáig a vízhálózaton keresztül.
Aktorok	Prototype
Forgatókönyv	<p>1.1 A Game értesíti a Ciszternát, hogy szimulálják a víz folyását.</p> <p>1.2 A Ciszternák értesítik a belé csatlakozó csövet, hogy szimulálja a víz folyását.</p> <p>1.3 A Ciszterna magába szívja a vizet a hozzá csatlakoztatott csőből.</p> <p>1.4 A Ciszterna értesítik az előtte levő aktív elemet a csőhálózatban, ami egy Pumpa.</p> <p>1.5 Ha van víz a Pumpa víztartályában és a Pumpa működik akkor próbál vizet pumpálni a kimeneti csövébe.</p> <p>1.6 A Pumpa értesíti a bemeneti csövét, hogy szimulálja a víz folyását.</p> <p>1.7 Ha a Pumpa kiürült és működik, akkor próbál vizet szívni a tartályába a bemeneti csövéből.</p> <p>1.8 A Pumpa értesíti az előtte lévő aktív elemet, ha az nem önmaga, hogy szimulálja a víz folyását, ami a Forrás.</p> <p>1.9 A Forrás megpróbál vizet pumpálni a kimeneti csöveibe.</p>

7.2.11

Use-case neve	Watersource handling a pipe
Rövid leírás	Az egyik Szerelő megpróbál bekötni egy csövet a Forrásba.
Aktorok	Prototype
Forgatókönyv	<p>1.1 A Szerelő megpróbálja bekötni a kezében lévő Cső egyik végén a Forrásba (PlacePipe) parancs</p> <p>1.2.A Ha a cső másik vége is a Forrásba vezet akkor a művelet sikertelen. (“<i>Can’t place Object1 on Object2</i>” kiírása)</p> <p>1.2.B Ha a Cső másik vége nem Forrásba van kötve akkor sikeresen beköt. (“<i>Successfully placed Object</i>” kiírása)</p>

7.2.12

Use-case neve	Repairman lifts pipe
Rövid leírás	A szerelő megpróbálja felemeli az egyik cső végét
Aktorok	Prototype
Forgatókönyv	<p>1.1 A szerelőt irányító játékos beírja a PickUpPipe parancsot.</p> <p>1.2.A Ha pumpán,cisterán,forráson áll, akkor a paraméterben megadott számú cső egyik végét lecsatlakoztatja az adott objektumról és magához veszi. (“<i>Successfully picked up Object</i>” kiírása)</p> <p>1.2.B Ha minden végén bekötött csövön áll akkor a felemelés sikertelen lesz (“<i>Invalid Pipe to pick up</i>” kiírása)</p> <p>1.2.C Ha egyik végén szabad csövön áll akkor a szabad véget magához veszi. (“<i>Successfully picked up Object</i>” kiírása)</p>

7.2.13

Use-case neve	Repairman placing pipe
Rövid leírás	Szerelő lehelyezi a nála lévő csövet
Aktorok	Prototype
Forgatókönyv	<p>1.1 A szerelőt irányító játékos beírja a PlacePipe parancsot.</p> <p>1.2.A Ha ez nem egy cső akkor a szerelő sikeresen lehelyezi a nála lévő csövet. (“<i>Successfully placed Object</i>” kiírása)</p> <p>1.2.B ha egy csövön hívja meg akkor nem sikerül a lehelyezni a csövet. (“<i>Can’t place Object1 on Object2</i>” kiírása)</p> <p>1.2.C ha nincs nála cső akkor a művelet sikertelen. (“<i>No placeable Pipe</i>”)</p>

7.2.14

Use-case neve	Pump or cistern handling pipe
Rövid leírás	Szerelő lehelyez egy nála lévő csöve egy pumpára vagy cisternára
Aktorok	Prototype
Forgatókönyv	1.1 A Szerelő megpróbálja bekötni a kezében lévő Cső véget egy pumpába vagy ciszternába (PlacePipe parancs)

	1.2. Amennyiben van a szerelőnél cső a lehelyezés minden sikeres (“Successfully placed <i>Object</i> ” kiírása)
--	---

7.2.15

Use-case neve	Calculate points at the end of turn
Rövid leírás	Pontok összeszámolása a kör végén
Aktorok	Prototype
Forgatókönyv	<p>1.1 A kör véget ér és a game osztályban tárolt pontokat szeretnénk frissíteni.</p> <p>1.2 A Game objektum az összes benne tárolt ciszternától lekérdezi, hogy mennyi víz folyt bele.</p> <p>1.3 Ezt a lekérdezett víz mennyiséget hozzáadja a szerelő csapat pontjaihoz.</p> <p>2.1. Lekérdezi, az összes olyan elemet amiből elfolyhat víz, hogy mennyi víz folyt el belőlük.</p> <p>2.2. Ezt a lekérdezett elfolyt víz mennyiséget hozzáadja a szabotőr csapat pontjaihoz.</p>

7.2.16

Use-case neve	Player Deals Damage
Rövid leírás	Egy játékos kilyukaszt egy objektumot.
Aktorok	Prototype
Forgatókönyv	<p>1.1 Egy játékos megpróbálja kilyukasztani az objektumot amin áll. (Stab parancs)</p> <p>1.2.A Ha az objektum már lyukas volt, akkor nem történik semmi. (“Failed to stab <i>Object</i>, already has hole” kiírása)</p> <p>1.2.B Ellenkező esetben amennyiben az objektum lyukasztható és nem volt javítva a közelmúltban, az kilyukadt (“Successfully stabbed <i>Object</i>” kiírása)</p> <p>1.2 C Ha az objektum javítva volt a közelmúltban a lyukasztás sikertelen (“Failed to stab <i>Object</i>, unbreakable until: <i>Turns</i>”)</p> <p>1.2 D Az objektum nem lyukasztható (“Failed to stab, <i>Object</i> is unbreakable”)</p>

7.2.17

Use-case neve	Character Moves
Rövid leírás	Egy karakter mozgása a csőrendszeren.
Aktorok	Prototype
Forgatókönyv	<p>1.1 A karakter kiválasztja melyik objektumra szeretné átlépni. (Move dir parancs kiadása)</p> <p>1.2 Az aktuális objektum amin éppen áll, visszaadja a választott objektumot a szomszédai közül.</p> <p>1.3 A cél objektumnak átadódik a karakter, tehát megpróbál átlépni oda.</p> <p>1.4.A Ha az előbbi átlépés sikeres volt akkor a korábbi pozícióról eltávolítjuk a karaktert. (“Successfully moved to <i>Object</i>” kiírása)</p>

	<p>1.4.B Ha az előbbi átlépés sikertelen volt akkor marad a korábbi pozíció a karakter. (“Failed to move to <i>Object</i>” kiírása)</p> <p>1.4.C Ha a karakter odaragadt az előző pozíójához, akkor az átlépés sikertelen volt. (“Sticked to <i>Object</i>, can ’t move until: <i>Turns</i>” kiírása)</p>
--	---

7.2.18

Use-case neve	DestructionTimer ticks
Rövid leírás	Pumpák véletlenszerű elrolásá.
Aktorok	Prototype
Forgatókönyv	<p>1.0 Szükséges, hogy a véletlenszerű események be legyen kapcsolva (ToggleRandom parancs)</p> <p>1.1 A kör elején a Game objektum jelez a DestructionTimer objektumnak.</p> <p>1.2 A DestructionTimer végigiterál az összes pumpán.</p> <p>1.3 Iteráció közben valamilyen valószínűsséggel elrontja az adott pumpát.</p>

7.2.19

Use-case neve	DestructionTimer receives new Pump
Rövid leírás	Új pumpa elhelyezésekor a DestructionTimer-ben tárolt pumpák listájának bővítése.
Aktorok	Prototype
Forgatókönyv	<p>1.1 A szerelő átadja a game objektumnak a pumpát amit elhelyez a csőrendszerben.</p> <p>1.2 A kapott pumpát a game objektum továbbadja a DestructionTimer objektumnak, amit majd hozzáad a listához.</p>

7.2.20

Use-case neve	Character slips thru pipe
Rövid leírás	A karakter átcusszan egy csöön a cső másik végére.
Aktorok	Prototype
Forgatókönyv	<p>1.1 Ha a cső, amire a karakter rálép épp csúszós, akkor a karakter egyből átcúszik a cső másik felén levő objektumra.</p> <p>1.2 (“Slipped to <i>Object</i>” kiírása)</p>

7.2.21

Use-case neve	Put slime on NonPipe
Rövid leírás	NonPipe objektum csúszóssá tétele
Aktorok	Prototype
Forgatókönyv	<p>1.1 A játékos megpróbál csúszóssá tenni egy nem csövet. (Slime)</p> <p>1.2 Sikertelen a művelet. (“<i>Object</i> can ’t be slimey”)</p>

7.2.22

Use-case neve	Put slime on pipe
Rövid leírás	Pipe objektum csúszóssá tétele

Aktorok	Prototype
Forgatókönyv	1.1 A játékos megpróbál csúszóssá tenni egy csövet. (Slime) 1.2 Sikeresen beállítódik a slimey attribútuma. (“ <i>Object</i> is now <i>slimey</i> ”)

7.2.23

Use-case neve	Make sticky on NonPipe
Rövid leírás	NonPipe objektum ragadóssá tétele.
Aktorok	Prototype
Forgatókönyv	1.1 A játékos megpróbál ragacsot lerakni egy nem csőre. (Stick) 1.2 Sikertelen művelet. (“ <i>Object</i> can’t be sticky”)

7.2.24

Use-case neve	Make Sticky on Pipe
Rövid leírás	Cső ragadóssá tétele.
Aktorok	Prototype
Forgatókönyv	1.1 A játékos megpróbál ragacsot lerakni egy csőre. (Stick) 1.2 Sikeresen beállítódik a sticky attribútuma. (“ <i>Object</i> is now <i>sticky</i> ”)

7.2.25

Use-case neve	Save game
Rövid leírás	A játék elmentése
Aktorok	Prototype
Forgatókönyv	1.1 A játékos kiadja a Save file parancsot 1.2.A A mentés sikeres (“Save successful” kiírása) 1.2.B A mentés sikertelen (“Save failed” kiírása)

7.2.26

Use-case neve	Load game
Rövid leírás	A játék betöltése
Aktorok	Prototype
Forgatókönyv	1.1 A játékos kiadja a Load file parancsot 1.2.A A betöltés sikeres (“Load successful” kiírása) 1.2.B A betöltés sikertelen (“Load failed” kiírása)

7.2.27

Use-case neve	Help
Rövid leírás	A játékos számára elérhető parancsok megjelenítése
Aktorok	Prototype
Forgatókönyv	1.1.A A Repairman játékos kiadja a Help parancsot (“Move dir, Adjust src dest, Repair, Stab, PickUpPump, PickUpPipe pipe, PlacePump, PlacePipe, Stick, EndTurn, Save file, Load file, Help, ToggleRandom I/N” kiírása) 1.1.B A Saboteur játékos kiadja a Help parancsot (“Move dir, Adjust src dest, Stab, Stick, Slime, EndTurn, Save file, Load file, Help, ToggleRandom I/N” kiírása)

7.2.27

Use-case neve	ToggleRandom
Rövid leírás	A véletlenm történő események ki és be kapcsolása
Aktorok	Prototype
Forgatókönyv	<p>1.1 A játékos kiadja a ToggleRandom I/N parancsot</p> <p>1.2.A Ha a paraméter I akkor engedélyezzük a véletlen eseményeket ("Random events Enabled" kiírása)</p> <p>1.2.B Ha a paraméter H akkor kikapcsoljuk a véletlen eseményeket ("Random events Disabled" kiírása)</p> <p>1.2.C Ha a paraméter nem I és nem N akkor sikertelen a parancs ("Invalid input" kiírása)</p>

7.2.28

Use-case neve	Adjust Pump
Rövid leírás	A játékosok állítják a pumpa be és kimenetét amin állnak.
Aktorok	Prototype
Forgatókönyv	<p>1.1 A játékos kiadja az Adjust src dest parancsot</p> <p>1.2.A Ha a játékos pumpán áll és érvényesek a paraméterek az állítás sikeres ("Input set <i>Object1</i>, Output set <i>Object2</i>" kiírása)</p> <p>1.2.B A játékos pumpán áll de a paraméterek helytelenek, az állítás sikertelen ("Invalid parameters, nothing changed" kiírása)</p> <p>1.2.C A játékos nem pumpán áll, így az állítás sikertelen ("<i>Object</i> is not Adjustable" kiírása)</p>

7.2.29

Use-case neve	Out of Action
Rövid leírás	A játékos kifogyott a végezhető tevékenységekből.
Aktorok	Prototype
Forgatókönyv	<p>1.1.A A Repairman kiadja a Move, Adjust, PickUpPipe, PickUpPump, PlacePipe, PlacePump, Stab, Repair, Stick parancsok egyikét, úgy hogy kifogyott a végezhető tevékenységekből a körében.</p> <p>1.1.B A Saboteur kiadja a Move, Adjust, Stab, Slime, Stick parancsok egyikét, úgy hogy kifogyott a végezhető tevékenységekből a körében.</p> <p>1.2 A parancs nem hajtódik végre ("Out of Action" kiírása)</p>

7.2.30

Use-case neve	Invalid Action
Rövid leírás	A játékos olyan parancsot ad ki melyet nem hajthat végre
Aktorok	Prototype
Forgatókönyv	<p>1.1.A A Repairman kiadja a Slime parancsot</p> <p>1.1.B A Saboteur kiadja PickUpPipe, PickUpPump, PlacePipe, PlacePump, Repair parancsot</p> <p>1.2.A A tevékenység sikertelen ("Not valid action for Repairman")</p> <p>1.2.B A tevékenység sikertelen ("Not valid action for Saboteur" kiírása)</p>

7.2.31

Use-case neve	Characters repairs unrepairable object
Rövid leírás	A játékos olyan játékelemet próbál megszerelni, ami nem szerelhető
Aktorok	Prototype
Forgatókönyv	<p>1.1 A játékos szól a karakterek, hogy kísérélje meg megszerelni azt az objektumot. (Repair parancs kiadása)</p> <p>1.2 A játékos által irányított karakter megkísérli megjavítani azt az objektumot, amin áll.</p> <p>1.2.1 ("<i>Object unrepairable</i>" kiírása)</p>

7.3 Tesztelési terv

7.3.1

Teszt-eset neve	Create map test
Rövid leírás	<p>1. Létrehozunk egy szövegszerkesztő segítségével egy tetszőleges kezdő pályát, amit kiírunk utána egy fájlba.</p> <p>2. A létrehozott szövegfájlt betöljük a prototípusban.</p> <p>3. A prototípusban kiadjuk a Save parancsot, melynek hatására elmentük a pályát.</p> <p>4. Összehasonlítjuk az eredeti bemeneti fájlt és a prototípusból mentett fájlt (például diff parancssori parancssal). Ekkor a különbségek csak annak kell lennie, hogy ebben mivel ebben a fájlban tárolódnak a játékosok is, ezért azokat is látnunk kell a mentett fájlban.</p>
Teszt célja	A szövegszerkesztővel létrehozott pálya és a prototípus közötti adatátvitel helyességének ellenőrzése.

7.3.2

Teszt-eset neve	Points test
Rövid leírás	<p>1. Elindítunk egy játékot.</p> <p>2. Tetszőleges pillanatban kimentjük a teljes játékállást, bezárjuk a prototípust.</p> <p>3. Ismét elindítjuk a játékot, majd betöljük az előzőleg kimentett játékállást, megvizsgáljuk a csapatok pontszámainak számát.</p>
Teszt célja	Pontszámok betöltésének, tárolásának és mentésének ellenőrzése

7.3.3

Teszt-eset neve	Remaining turns test
Rövid leírás	1. Elindítunk egy játékot.

	<p>2. Tetszőleges pillanatban kimentjük a teljes játékállást, bezárjuk a prototípust.</p> <p>3. Ismét elindítjuk a játéket, majd betöljtük az előzőleg kimentett játékállást, megvizsgáljuk a hátramaradt körök számát.</p>
Teszt célja	A hátramaradt körök betöltésének, tárolásának és mentésének ellenőrzése

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

szövegszerkesztő

Egy olyan program, melynek segítségével létre tudunk hozni egy szövegfájlt, ami tartalmazza a kezdő pálya adatait.

diff

Egy parancssori program, melynek segítségével két fájl közti eltéréseket vizsgálhatjuk meg. Ezt használjuk az olyan fájlok vizsgálatákor, ahol fontosak a be- és kimeneti fájlok közti különbségek

javac

A Java fejlesztési csomaghoz tartozó program, melynek segítségével futtatható binárisá fordíthatóak az állományok.

7.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2023.04.20 10:00	1,5 óra	Andai, Bodnár, Gutási, Sziksai, Vajda	Értekezlet. Alfeladatok kiosztása egymás között. Döntés: Vajda ekészíti a 7.0.1 és 7.0.2 szakaszt, Andai elkészíti a 7.0.3 szakaszt, Sziksai elkészíti a 7.1.2 és 7.1.3 szakaszt., Gutási elkészíti a 7.3 és 7.4 szakaszt.
2023.04.21 9:00	3,5 óra	Andai	Reagálás a feladatkiírás módosítására. Diagramok módosításainak tervezése. 7.0.3.1- 7.0.3.9 diagramok szerkesztése.
2023.04.21 19:00	1,5 óra	Vajda	Feladat kiírás módosítások felvezetése az osztálydiagramra
2023.04.21 22:45	1 óra	Andai	7.0.3.10-7.0.3.12 szekvencia diagramok szerkesztése.
2023.04.22 13:00	1,5 óra	Sziksai	Bemeneti nyelv és Kimeneti nyelv elkészítése
2023.04.23 15:00	1,5 óra	Sziksai	Kimeneti nyelv beültetése a use case-ekbe, kimeneti nyelv bővítése
2023.04.23 15:00	2,5 óra	Gutási	Use case-ek átemelése az 5. dokumentumból, azok ki- és bemeneti nyelvekkel való kiegészítése , 7.3 és 7.4 pontok megírása

2023.04.23 19:00	2,5 óra	Vajda	Osztálydiagram javítása, új és változott metódusok leírása a 7.0.2 pontban
2023.04.23 20:00	1 óra	Andai	7.2.21-7.2.24 use- case leírások kifejtése, apróbb javítások a dokumentumban

8. Részletes tervez

8.1 Osztályok és metódusok tervez.

8.1.1 Character

- **Felelősség**

Ebből az osztályból származnak le a játékosok által irányítható karakterek (szerelők, és szabotőrök).

- **Attribútumok**

- **Metódusok**

- **void move(int dir):** Karakter átléptetése egy irányba. Az irány egy szám, ami egyértelműen azonosítja egy elem szomszédait.

```
neighbors:=position.getNeighbors()
lastPosition=position
removeSuccess:=position.remove(r)
Ha(removeSuccess):
acceptSuccess:=neighbordir.accept(r)
Ha (!acceptSuccess):
    lastPosition.accept(c)
Különben:
return;
```

- **void adjustPump(int src, int dest):** Pumpa aktív ki- és bemenő élének beállítása (meghívja a Place interfész adjust() metódusát). A két szám egyértelműen azonosít egy-egy csövet, ami a pumpába van kötve.

```
position.setInputPipe(src)
position.setOutputPipe(dest)
```

- **void step():** A Game osztály hívja meg, ezzel kiválasztva a karaktert, hogy ő léphet. Ezen metódus elején állítódik be a myTurn attribútum igazra, ezzel jelezve, hogy a bemenet erre a karakterre irányul. Miután a karakter lépett, a myTurn értéke ismét hamis lesz, és a step() metódus visszatér.

```
myTurn=true
Amíg myTurn
lastInput=input
lastInputSuccess=input()
Ha(lastInputSuccess && lastInput!=move)
    myTurn=false
return
```

- **void makeSticky():** Ragadóssé teszi az adott csövet amin áll

```
position.stick()
```

- **void dealDamage():** Kilyukasztja az adott csövet amin áll amennyiben az még nem volt lyukas)

```
position.damage()
```

8.1.2 Cistern

- **Felelősség**

A víz szimulációja innen indul. A belefolyt víz mennyiségét tárolja. A játékosok véges számban rá tudnak lépni. Új csövek jönnek létre rajta, melyek egyik vége bele, másik vége pedig a sivatagba vannak kötve. A Szerelő játékosok felvehetnek innen új Pumpát.

- **Ösosztályok**

- ***Element → NonPipe***

- **Interfészek**

- ***SaboteurPlace, RepairmanPlace***

- **Attribútumok**

- **- int waterFlown:** A belé folyt víz mennyisége.

- **Metódusok**

- **+ Cistern():** Meghívja az ős konstruktort és a waterFlown-t 0-ra állítja.

- **+ void step():** Magába szívja a bemeneti csövekből a vizet, elindítja a víz szimulációt ezen csövek mentén.

```
Minden p csőre a Neighbors-ban
p.step()
HA p.extractWater()
    waterFlown++
Minden np NonPipe-ra p.getNeighbors() -ból
    HA np nem = this
        np.step()
```

- **+ int measureAndResetWaterFlown():** Visszaadja majd 0-ra állítja a Ciszternába folyt víz mennyiségét.
- **+ Pipe newPipe():** Ha nincs sivatagban végződő cső a szomszédjai közöttakkor létrehoz egyet.

```
boolean létezik = false
Minden p Pipe-ra a neighbors-ból
HA p.getNeighbors().get(1) == null
    létezik = true
HA nem létezik
Pipe createdPipe = új Pipe()
createdPipe.setName("createdPipe1");
createdPipe.addNeighbor(this)
neighbors.add(createdPipe)
Visszatér createdPipe
Visszatér null
```

- **+ Pump givePump():** Visszaad egy új pumpát és kiírja, hogy sikeres.

```
Pump p = új Pump()
```

```

    p.setName("p");
Kiírni "Successfully picked up"+p.getName()
Visszatér p

```

8.1.3 DestructionTimer

- **Felelősség**

Időzítő, mely minden játékos lépése előtt, véletlen eséllyel elrontja az egyes pumpákat.

- **Atribútumok**

- **-Pump[0..*] pumps:** A tárolt pumpák, melyeket elronthat az időzítő.
- **-String Name:** Az időzítő neve a játékban (pl.: Timer1)

- **Metódusok**

- **+String getName():** Pumba nevének lekérdezése. String típussal tér vissza.
- **+void setName(String name):** Pumba nevének beállítása. String típusú paraméterként várja a pumpa nevét.
- **+void tick():** Egy bizonyos eséllyel elrontja a pumpákat egyesével. Amikor lefut a metódus, minden egyes tárolt pumpára kiszámol egy véletlenszerű számot, mely 0 és 1 közé esik. Ha például 10% eséllyel ront el egy pumpát a tick() metódus, akkor minden olyan esetben, amikor a szám <=0.1, akkor meghívódik az adott pumpa **breakPump()** metódusa.
- Pszeudokód 10% esélyű elromlásra:

Minden tárolt p pumpára:

```
ran:= véletlenszerű szám 0 és 1 között;
```

```
HA (ran <= 0.1) :
```

```
    (p elrontása)
```

- **+void addPump(Pump p):** Hozzáadja p pumpát a tárolt pumpák listájához.

8.1.4 Element

- **Felelősség**

- Közös absztrakt őse a pályaelemeknek. Megvalósítja a karakterek számára szükséges függvényeket, azok alap implementációit megvalósítja.

- **Interfészek**

- RepairmanPlace, SaboteurPlace

- **Atribútumok**

- # boolean containingWater: eltárolja, hogy van-e benne víz
- # List<Character> standingOn: eltárolja, hogy kik állnak rajta
- – String name: az objektum neve

- **Metódusok**

- **+ Element():** konstruktor, inicializálja a vizet, és létrehoz egy listát a rajta álló karaktereknek

```

containingWater := false
standingOn := új Lista
○ + void repair(): alapértelmezetten nem csinál semmit

```

```
kiír name + „ unrepairable”
```

- + boolean accept(Character c): alapértelmezetten elfogadja a rálépő karaktert

```
standingOn hozzáad c
```

```
kiír „Successfully moved to „ + name
```

```
visszatér true
```

- + boolean remove(Character c): alapértelmezetten törli a karaktert magáról

```
standingOn eltávolít c
```

```
visszatér true
```

- + boolean placePipe(Pipe p): alapértelmezetten nem csinál semmit

```
kiír „Can't place „ + p.getName + „ on „ + name
```

```
visszatér false
```

- + void step(): Absztrakt függvény, amely a vízfolyást teszi lehetővé.
- + void adjust(int src, int dest): alapértelmezetten nem csinál semmit

```
kiír name + „ is not adjustable”
```

```
visszatér false
```

- + Pipe placePump(Pump p): alapértelmezetten nem csinál semmit

```
kiír „Can't place „ + p.getName + „ on „ + name
```

```
visszatér null
```

- + Pipe lift(int dir): alapértelmezetten az adott irányban belekötött csövet adja vissza, és lecsatlakoztatja magáról

```
n := neighbors[dir]
```

```
ha n nem null:
```

```
    removeNeighbor(n)
```

```
    n.removeNeighbor(önmaga)
```

```
kiír „Successfully picked up ” + n.getName
```

```
    visszatér n
```

```
egyébként:
```

```
kiír „Invalid object to pick up”
```

```
visszatér null
```

- + Pump givePump(): alapértelmezetten nem csinál semmit

```
kiír „Can't pick up Pump here”
```

```
visszatér null
```

- + void stick(): alapértelmezetten nem csinál semmit

```
kiír name + „ can't be sticky”
```

- + void slime(): alapértelmezetten nem csinál semmit

```
kiír name + „ can't be slimy”
```

- + String getName(): visszaadja a nevét

```
visszatér name
```

- + void setName(String n): beállítja a nevét a paraméterben kapotttra

```
name := n
```

8.1.5 Game

- **Felelősség**

Kezeli a játék indítását, befejezését és a vízfolyás szimulációját. Továbbá tárolja a csapatokat, benne a játékban szereplő karakterekkel, ezen csapatok pontjait és a játék egyéb elemeit is. Kezeli a víz szimulációját, és a körök felosztását. .

- **Attribútumok**
- **-Repairman[2..*] repairmanGroup:** A szerelő csapatot alkotó karaktereket tárolja.
- **-Saboteur[2..*] saboteurGroup:** A szabotőr csapatot alkotó karaktereket tárolja.
- **-int repairmanPoints:** A szerelő csapat pontjait tárolja.
- **-int saboteurPoints:** A szabotőr csapat pontjait tárolja.
- **-Element[0..*] gameElements:** minden a játékban található elementet tárol
- **-Cistern[0..*] cisterns:** Tárolja a játékban szereplő ciszternákat.
- **-SaboteurPointSource[0..*] saboteurPointSources:** Olyan elemeket tárol, melyekből kaphatnak pontot a szabotőrök.
- **-int remainingRounds:** A játékból hátramaradó körök száma. Egy kör több lépésből áll. Először a Game meghívja a DestructionTimer Tick() metódusát. Ezután először az egyik csapat összes karaktere, majd a másik csapat összes karaktere lép. minden karakter lépése után szimulálódik a vízfolyás, és hozzáadódnak a pontok a csapatok számára.
- **-DestructionTimer timer:** időzítő amely a pumpák véletlen elromlásáért felelős

- **Metódusok**

- **+Game():** Az osztály konstruktora.

```
(Listák inicializálása) ;
repairmanPoints:=0;
saboteurPoints:=0;
remainingRounds:=1;
```

Ezeket az értékeket az initialize később felülírhatja.

- **+void initialize():** Játék inicializálása. A load bemeneti nyelvi parancs a fájl betöltése után ezt a függvényt is meghívja, melyen keresztül beállítódnak a játékkal kapcsolatos különböző attribútumok.
- **+void playGame():** Elindítja a játékot, és menedzseli, melyik karakter léphet éppen.

Amíg remainingRounds>0:

Minden egyes s szabotőrre a saboteurGroupban:

```
s.step();
endTurn();
```

Minden egyes r szerelőre a repairmanGroupban:

```
r.step();
endTurn();
```

remainingRounds-=1;

endGame();

- **+void endGame():** Játék befejezése, a győztes csapat kihirdetése.
Ha saboteurPoints>repairmanPoints:
Kír: "Saboteur Team Won!";
Ha repairmanPoints>saboteurPoints:
Kír: "Repairman Team Won!";
Egyébként:
Kír: "It's a Draw!";
- **+void endTurn():** Egy adott kör véget ér, elindul a vízszimuláció
SimulateWaterflow();
CalculatePoints();
- **+void SimulateWaterflow():** Elindítja a víz szimulációját a ciszternáktól.
Minden c ciszternára:
c.step()
 - **+void addElement(Element e):** Objektum hozzáadása a pályához.
 - **+Element[] getGameElements():** A játékban szereplő elemek lekérdezése.
 - **+void addSaboteur():** Létrehoz és hozzáad egy Szabotőt a játékhoz.
 - **+void addRepairman():** Létrehoz és hozzáad egy Szerelőt a játékhoz.
 - **+void addPipe(p: Pipe):** Hozzáad egy csövet a játékhoz.
 - **+void addCistern(c: Cistern):** Hozzáad egy ciszternát a pályához.
 - **+void calculatePoints():** Növeli a csapatok pontszámait a ciszternákba befolyt és a csövekből, illetve pumpákból elfolyt víz alapján.

```

repairmanWater:=0;
saboteurWater:=0;
minden c Ciszternára:
ciWater:=c.measureAndResetWaterFlow();
repairmanWater+=ciWater;
minden SP saboteurPointSource-ra a saboteurPointSources-ben:
sWater=SP..measureAndResetLeakedWaterAmount();
saboteurWater+=sWater
repairmanPoints+=repairmanWater;
saboteurPoints+=saboteurWater;

```

- **+void addPump(p:Pump):** Hozzáad egy pumpát a játékhoz.
- **+void newPipe():** Létrehoz egy új csövet a ciszternánál.
- **+setTimer(dt: DestructionTimer):** Hozzáad egy DestructionTimer időzítőt a játékhoz.

8.1.6 NonPipe

- **Felelősség**
 - Az absztrakt ōsosztálya a Cistern, WaterSource és a Pump osztályoknak, a közös funkciókat valósítja meg.
- **Ósosztályok**
 - **Element**
- **Interfészek**
 - **SaboteurPlace, RepairmanPlace**
- **Attribútumok**
 - **# List<Pipe> Neighbors:** Tárolja a szomszédos csöveket.
- **Metódusok**
 - **+ NonPipe():** Meghívja az ōs konstruktort és inicializálja neighbors-t.
 - **+ void addNeighbor(Pipe n):** Hozzáadja n-t a neighbors-hoz.
 - **+ void removeNeighbor(Pipe n):** Eltávolítja n-t a neighbors-ból.
 - **+ List<Pipe> getNeighbors():** Visszaadja a szomszédos csöveket.
 - **+ Pipe lift(int dir):** Az egyik hozzá csatlakozó cső lecsatolása, visszaadja a lecsatolt csövet és a dir a lecsatolandó csövet határozza meg.

```
Pipe p = neighbors.get(dir)
Ha p nem null
removeNeighbors(p)
P.removeNeighbors(this)
Visszatér p
Különben
Visszatér null
```

- **+ boolean placePipe(Pipe holdingPipe):** Egy cső bekötése NonPipe-ba, ahol holdingPipe a bekötendő cső.

```
HA holdingPipe nem null
holdingPipe.addNeighbor(this)
addNeighbor(holdingPipe)
Kiírni: "Successfully placed"+holdingPipe.getName()
Visszatér tere
Különben
Kiírni "No placeable Pipe"
Visszatér false
```

8.1.7 Pipe

- **Felelősség**

Csőként funkciomál. Egy játékos állhat rajta ezt számon tartja. Számolja a rajta kifolyt víz mennyiségét. Ezt le lehet tőle kérdezni. Engedi, hogy rakjanak bele vizet, illetve hogy kivegyenek belőle. Foltozható, lyukasztható ragacsoszsa, csúszóssá tehető.

- **Ősosztályok**

- Element

- **Interfészek**

- SaboteurPointSource

- **Attribútumok**

- – boolean **holeOnPipe**: van-e lyuk a csövön
- – int **leakedWaterAmount**: mennyi víz szivárgott ki rajta az előző lekérdezés óta
- – List<NonPipe> **neighbors**: a cső szomszédjait tároló lista
- – int **breakable**: hány körig nem még lehet újra kilyukasztani
- – int **slimey**: hány körig csúszós még a cső
- – int **sticky**: hány körig ragacsos még a cső

- **Metódusok**

- + **Pipe()**: konstruktor, amelyben a tagváltozókat inicializáljuk

```
holeOnPipe := false
```

```
neighbors := új Lista
```

```
leakedWaterAmount := false
```

- + boolean **accept(Character c)**: Engedi magára lépni a paraméterben kapott karaktert, ha ez lehetséges.

```
ha standingOn.count == 1:
```

```
kiír "Failed to move to " + name
```

```
visszatér false
```

```
egyébként:
```

```
ha slimey > 0:
```

```
success = false
```

```
ha neighbors.count > 1:
```

```
ha random igaz:
```

```
success = neighbors[0].accept(c)
```

```
kiír "Slipped to " + neighbors[0].name
```

```
egyébként:
```

```
    success = neighbors[1].accept(c)
```

```
kiír "Slipped to " + neighbors[1].name
```

```
egyébként:
```

```
    success = neighbors[0].accept(c)
```

```
kiír "Slipped to " + neighbors[0].name
```

```
visszatér success
```

```
egyébként:
```

```
neighbors hozzáad c
kiír "Successfully moved to " + name
visszatér true
```

- + boolean remove(Character c): Eltávolítja magáról a paraméterben kapott karaktert, ha ez lehetséges. Ha a sticky nagyobb 0-nál, akkor bele van ragadva a karakter.

```
ha sticky > 0:
    sticky csökkent
kiír "Sticked to " + name + "can't move until: " + sticky
    visszatér false
egyébként:
    neighbors töröl c
ha sticky < 0:
    sticky := -sticky
    visszatér true
```

Ha 0, nincs beleragadva. Ha kisebb mint 0, akkor az azt jelenti, hogy az a játékos ragasztózta be, aki most el akar lépni. Őt még elengedi, viszont átírja pozitívra a sticky értékét, hogy a következő játékos ragadjon bele.

- + void step(): Ha lyukas, kifolyatja magából a vizet és növeli a LeakedWaterAmountot

```
ha holeOnPipe:
    containingWater := false
    leakedWaterAmount növel
○ + void repair(): holeOnPipe-ot false-ra állítja, ha lyukas volt
```

```
ha holeOnPipe:
    holeOnPipe := false
    breakable := random
kiír "Successfully repaired " + name
egyébként:
kiír name + " not broken"
○ + void damage(): holeOnPipe-ot true-ra állítja, ha lyukasztható
```

```
ha breakable 0:
ha holeOnPipe:
kiír "Failed to stab " + name + "already has a hole"
egyébként:
holeOnPipe := true
kiír "Successfully stabbed " + name
egyébként:
kiír "Failed to stab " + name + "unbreakable until: " +
breakable
```

- + Pipe placePump(Pump holdingPump): Létrehoz egy új csövet, majd közé és a meglévő cső közé a paraméterben kapott pumpát helyezi el. Visszaadja az új csövet.

```

ha holdingPump null:
kiír "No placeable pump"
n := neighbors[0]
ha n nem null:
    removeNeighbor(n)
    n.removeNeighbor(önmaga)
    holdingPump.addNeighbor(önmaga)
    addNeighbor(holdingPump)
    p := új Pipe
    p.addNeighbor(n)
    n.addNeighbor(p)
    holdingPump.addNeighbor(p)
    p.addNeighbor(holdingPump)
kiír "Successfully placed " + holdingPump.name
visszatér p
egyébként:
kiír "Can't place " + holdingPump.name + " on " + name
visszatér null

```

- + **Pipe lift(dir int)**: Olyan cső felemelésénél használatos, amelyiknek az egyik fele nincs sehova sem beköve. Ekkor a cső egyik végének a felemeléséhez rajta kell állni ezen a csövön.

```

ha neighbors[0] null xor neighbors[1] null:
    kiír "Successfully picked up " + name
visszatér önmagával
egyébként:
kiír "Invalid pipe to pick up"
visszatér null

```

A paramétert nem használjuk semmire, csak egyformának kell lennie azzal, amit felülír.

- + **void addNeighbor(NonPipe n)**: Cső szomszédjának beállítása.

```

ha neighbors.count < 2:
    neighbors hozzáad n

```

- + **void removeNeighbor(NonPipe n)**: A paraméterül kapott szomszédot eltávolítja.

```
neighbors töröl n
```

- + **int measureAndResetLeakedWaterAmount()**: Visszaadja, a leakedWaterAmountot, és lekérdezés után visszaállítja azt nullára
- ```

l := leakedWaterAmount
leakedWaterAmount := 0
visszatér l

```
- + **boolean WaterExtraction()**: Kiszívja az adott csővől a vizet és visszatér, hogy volt-e benne.

```
ha containingWater:
```

```

containingWater := false
visszatér true
egyébként:
visszatér false
○ + boolean giveWater(): Vizet próbál tenni a csőbe, visszaadja, hogy sikerült-e.

ha containingWater:
visszatér false
egyébként:
containingWater := true
visszatér true
○ + void calculateParams(): Körönként hívódik meg, és csökkenti a stick, slimey és a
breakable paramétert, ha kell.

ha breakble > 0:
breakable csökkent
ha slimey > 0:
slimey csökkent
○ + void slime(): Csúszóssá teszi a csövet. Beállítja a slimey értékét egy konstansra.

slimey := konstans
kiír name + " now is slimey"
○ + void stick(): Ragadóssá teszi a csövet. Beállítja a sticky értékét egy konstansra. Ez
a konstans egy negatív érték lesz, ez jelzi, hogy még aki letette, el tud menekülni, és az utána
következő játékosra lesz érvényes. Majd az első remove-nál állítódik pozitívra.

sticky := konstans
kiír name + " now is sticky"

```

### 8.1.8 Place

- **Felelősség**
  - Közös interfésze azoknak az osztályoknak, amelyekre lehet lépni.
- **Metódusok**
  - + **void adjust(int src, int dest)**: A pumpa pumpálási irányának állítására szolgáló függvény, amit majd a pumpa valósít csak meg érdemben.
  - + **boolean accept(Character c)**: A karakterek egy helyre való lépésének kezelésére szolgáló függvény, amelyet a különböző elemek, különbözőképpen valósítanak meg.
  - + **boolean remove(Character c)**: A karakterek egy adott helyről való leléptetésére szolgál, amit a különböző elemek különböző módon valósítanak meg.
  - + **Element[] getNeighbors()**: A szomszédok listájának visszaadását csinálja.
  - + **void damage()**: Ezen keresztül lehet rongálni a különböző elemeket. Azok, akiket lehet rongálni, a saját karakterisztikájuknak megfelelően implementálják.
  - + **void stick()**: Ezen keresztül lehet a csövet ragadóssá tenni. Csak a cső implementálja érdemben.

### 8.1.9 Pump

- **Felelősség**
- Pumpaként funkcionál, el tud romlani és a bemenetéről vizet pumpál a kimenetére ha jól működik. Ha nincs kimeneti csöve akkor számolja az ott elfolyt víz mennyiségét.
- Szomszédait tárolja melyek csak csövek lehetnek. Játékosok rá tudnak lépni véges számban.
- **Ösosztályok**
- ***Element → NonPipe***
- **Interfészek**
- ***SaboteurPointSource, SaboteurPlace, RepairmanPlace***
- **Attribútumok**
- - **Pipe inputPipe:** A bemenetként szolgáló cső melyről vizet szív.
- - **Pipe outputPipe:** A kimenetként szolgáló cső ahova vizet pumpál.
- - **boolean broken:** az tárolja, hogy működik-e a pumpa.
- - **int leakedWaterAmount:** A belőle a sivatagba elfolyt vizet tárolja.
- **Metódusok**
- + **Pump():** Meghívja az ős konstruktörét, a broken-t false-ra, míg a leakedWaterAmount-ot 0-ra állítja.
- + **void repair():** Megjavítja a pumpát ha el volt romolva.

```

HA broken
broken=false
Kiírni: "Successfully repaired"+name
Különben
Kiírni: name+"not broken"

```

- + **void adjust(int src, int dest):** Beállítja a bemeneti és kimeneti csöveget az src-edik és a dest-edik szomszédjára.

```

HA src < neighbours.size() és dest < neighbors.size()
inputPipe = neighbors.get(src) ;
outputPipe = neighbors.get(dest) ;
Kiírni: "Input set "+neighbors.get(src).getName()+""
Output set "+neighbors.get(dest).getName()+"
Különben
Kiírni: "Invalid parameters, nothing changed"

```

- + **void step():** Ha van a pumpában víz azt átpumpálja a kimeneti csövébe, ha kiürült a pumpa szív vizet a bemeneti csövéről.

```

HA containingWater és nem broken
HA outputPipe.giveWater()
 containingWater=false
inputPipe.step()
HA nem containingWater és nem broken

```

```
Ha inputPipe.waterExtraction()
 containingWater=true
```

- **+ void breakPump():** Elrontja a pumpát. Beállítja a broken értékét true-ra.
- **+ int measureAndResetLeakedWaterAmount():** Visszaadja, majd 0-ra állítja a Pumpán keresztül a sivatagba elfolyt víz mennyiségét.

### 8.1.10 Repairman

- **Felelősség**

Játékos által irányított szerelő karakter. Célja, hogy minél több víz jusson el a forrásoktól a ciszternákba.

- **Ősosztályok**

Character

- **Attribútumok**

- **RepairmanPlace position:** Ahol éppen a Szerelő áll.

- **Pipe holdingPipe:** A felvehető csővég tárolására szolgál.

- **Pump holdingPump:** A felvehető Pumpa tárolására szolgál.

- **Game game:** Tárolja a játék egy referenciáját, mely az új Csövek és Pumpák eltárolásához szükséges.

- **Metódusok**

- **void RepairElement():** Megjavítja azt az Element játékelementet, amin áll.

Megtörténhet, hogy ezt egy olyan elemen állva végzi, ami nem tud elromlani/megjavulni. Ekkor nem történik semmi.

**position.repair()**

- **void LiftPipe(int dir):** Ha a játékos nem csövön áll, akkor az adott elementnek amin áll a paraméterben átadott számú szomszédját magához veszi

**position.lift(dir)**

- **void LiftPump():** Ha a játékos egy ciszternán áll, akkor magához vehet egy új pumpát.

**position.givePump();**

- **void PlacePump():** Ha a játékos egy csövön áll, lehelyezheti a nála levő pumpát.

**position.placePump(holdingPump)**

- **void PlacePipe():** Ha a játékos épp nem egy csövön áll, akkor lehelyezheti a nála lévő csövet.

**position.placePipe(holdingPipe)**

- **Pump getHoldingPump():** Visszaadja a karakternél levő pumpát.

**return holdingPump**

- **void setHoldingPump(Pump p):** Beállítja a karakternél levő pumpát.

**holdingPump=p**

- **Pipe getHoldingPipe():** Visszaadja a karakternél levő csövet.

**return holdingPipe**

- **void setHoldingPipe(Pipe p):** Beállítja a karakternél levő csövet.

**holdingPipe=p;**

### 8.1.11 RepairmanPlace

- **Felelősség**

- Közös interfésze azoknak az osztályoknak, amelyekre a szerelő tud lépni.

- **Ősosztályok**

- Place

- **Metódusok**
- + **void repair()**: Ezen keresztül lehet a különböző elemeket megjavítani. Akik javíthatók, a saját módjaiknak megfelelően implementálják.
- + **Pipe placePump(Pump p)**: Ezen keresztül lehet pumpát lehelyezni. Érdemben csak a cső implementálja.
- + **Pipe lift(int dir)**: Ő felelős a csövek felemeléséért. Rajta keresztül lehet felemelni egy csak ciszternába kötött csövet, ha a csövön állunk. Illetve ha nem csövön állunk, akkor a megadott irányba lévő csövet tudjuk felemelni.
- + **Pump givePump()**: Rajta keresztül lehet új pumpát igényelni. Érdemben csak a ciszterna valósítja meg.
- + **boolean placePipe(Pipe p)**: Ő felelős a csövek lehelyezéséért. A nem csövek valósítják meg érdemben. Ők bekötik magukba a paraméterben kapott csövet.

### 8.1.12 Saboteur

- **Felelősség**

A játékosok által irányított szabotőrök, céljuk hogy minél több víz folyjon el a sivatagba a játék során.

- **Ősosztályok**

Character

- **Attribútumok**
  - **SaboteurPlace position**: Elem, amin a szabotőr éppen áll.
  - **Metódusok**
  - **void putSlime()**: A Saboteur osztály része, csúszóssá teszi az adott csövet amin éppen áll
- ```
position.slime()
```

8.1.13 SaboteurPlace

- **Felelősség**

- Közös interfésze azoknak az osztályoknak, amelyekre a szabotőr tud lépni.

- **Ősosztályok**

- Place

- **Metódusok**

- + **void slime()**: Rajta keresztül lehet csúszóssá tenni egy csövet. Csak a cső valósítja meg érdemben.

8.1.14 WaterSource

- **Felelősség**
 - Végtelen vízforrás. Eltárolja a szomszédos csöveket és beléjük vizet pumpál. Véges számú Játékos rá tud lépni.
- **Ősosztályok**
 - *Element → NonPipe*
- **Interfészek**
 - *SaboteurPlace, RepairmanPlace*
- **Metódusok**
 - **+ WaterSource():** Meghívja az ōs konstrukturát.
 - **+ void step():** Vizet pumpál minden belekötött csőbe.
 - **+ boolean placePipe(Pipe holdingPipe):** Bekötjük a holdingPipe csövet a Forráshoz, visszaadja hogy sikerrel jártunk-e. Nem lehet olyan cső aminek minden két vége ugyanabba a Forrásba van kötve.

```
boolean out
NonPipe n = holdingPipe.getNeighbors().get(0)
HA n nem = this
addNeighbor(holdingPipe)
holdingPipe.addNeighbor(this)
out = true
Kiírni "Successfully placed"+holdingPipe.getName()
Különben
out = false
Visszatér out
```

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

Az elvárt kimenethez azokat írjuk kiemeltre, amik a kimeneti nyelv részei!

Bővítesek a tesztekhez:

- Egy játékosnak maximum 3 action-je lehet, utána fogy csak ki az lépések számából.
- Bevezetünk egy *info* parancsot, amely a paraméterben megadott objektum állapotát írja ki:
 - példa: **info pipe1**
 - **pipe1 false true 5 0** -> ez a következőket jelöli:
hasHole, continingWater, slimy 5 körig, sticky 0 körig
 - **példa2: info pump1**
 - **pump1 true false** -> isBroken continingWater mezőket jelöli
- A **filePoints.txt** tartalma bővül azzal az információval, hogy a Pumpák mekkora eséllyel romlanak el körönként, egy cső foltozás után hány körig nem lyukasztható.

8.2.3.1 Pályadeklarációk a tesztekre

8.2.3.1.1 1.txt

```
###Declaration###
Pu p1 false false
C c
Pi pil true true 0 5
Pi pi2 false true 0 0
###Players###
R repairman c null null
###Neighbors###
c pil
pil c p1
p1 pil pi2
```

8.2.3.1.2 2.txt

```
###Declaration###
Pu p1 false false
C c
Pi pil false false 0 0
###Players###
R repairman c null null
###Neighbors###
c pil
pil c p1
p1 pil
```

8.2.3.1.3 3.txt

```
###Declaration###
Pu p1 false false
C c
Pi pil false false 5 5
###Players###
R repairman c null null
###Neighbors###
c pil
pil c p1
p1 pil
```

8.2.3.1.4 4.txt

```
###Declaration###
Pu p1 false false
Pu p2 false false
C c
W w
```

```

Pi pi1 false false 0 0
Pi pi2 false false 0 0
Pi pi3 false false 0 0
Pi pi4 false false 0 0
###Players###
R repairman1 c null null
R repairman2 c null null
S saboteur1 c null null
S saboteur2 c null null
###Neighbors###
c pi1
c pi2
pi1 c
pi2 c p1
pi3 w p1
w pi3
pi4 p1 p2
p1 pi1
p1 pi3
p2 pi4

```

8.2.3.1.5 5.txt

```

###Declaration###
Pu p1 false false
Pu p2 false false
C c
W w
Pi pi1 false false 0 0
Pi pi2 false false 0 0
Pi pi3 false false 0 0
Pi pi4 false false 0 0
###Players###
R repairman1 w null null
R repairman2 w null null
S saboteur1 c null null
S saboteur2 c null null
###Neighbors###
c pi1
c pi2
pi1 c
pi2 c p1
pi3 w p1
w pi3
pi4 p1 p2
p1 pi1

```

```
p1 pi3
p2 pi4
```

8.2.3.1.6 6.txt

```
###Declaration###
C Cistern
Pu Pump1 false true
Pu Pump2 false true
Pu Pump3 false true
Pu Pump4 false true
Pu Pump5 false true
Pu Pump6 false true
Pu Pump7 false true
Pu Pump8 false true
Pu Pump9 false true
Pu Pump10 false true
W Source
Pi Pipe1 false true 0 0
Pi Pipe2 false true 1 4
###Players###
S Saboteur1 Pipe1 null null
###Neighbors###
Cistern Pipe1
Pipe1 Cistern Pump1
Pump1 Pipe1 Pipe2
Pipe2 Pump1 Source
Source Pipe2
```

8.2.3.1.7 7.txt

```
###Declaration###
C Cistern
Pu Pump false true
W Source
Pi Pipe1 false true 0 0
Pi Pipe2 false true 1 4
###Players###
R Repairman1 Pump null null
S Saboteur1 Pump null null
###Neighbors###
Cistern Pipe1
Pipe1 Cistern Pump
Pump Pipe1 Pipe2
Pipe2 Pump Source
Source Pipe2
```

8.2.3.1.8 8.txt

```
###Declaration###
C c
Pi pipeOnGround false false 0 0
Pi holdingPipe false false 5 0
Pu pump false false
###Players###
R repairman c holdingPipe null
S sab c
###Neighbors###
c pipeOnGround holdingPipe
holdingPipe c
pipeOnGround c pump
pump pipeOnGround
```

8.2.3.1.9 9.txt

```
###Declaration###
Pu p1 false false
Pu p2 false false
Pu p3 false false
Pi pi1 false false 0 0
Pi pi2 false false 0 0
###Players###
R repairman p2 null null
###Neighbors###
p1 pi1
p2 pi1
p2 pi2
p3 pi2
pi1 p1
pi1 p2
pi2 p2
pi2 p3
```

8.2.3.1.10 10.txt

```
###Declaration###
C c
Pi pipeOnGround1 false false 0 0
Pi pipeOnGround2 false false 0 0
Pi holdingPipe false false 0 0
Pu pump1 false false
Pu pump2 false false
###Players###
R r1 holdingPipe null null
R r2 c null null
S sab c
```

```
###Neighbors###
c pipeOnGround1 holdingPipe
holdingPipe c
pipeOnGround1 c pump1
pump1 pipeOnGround1 pipeOnGround2
pipeOnGround2 pump1 pump2
pump2 pipeOnGround2
```

8.2.3.1.11 11.txt

```
###Declaration###
Pu p1 false false
C c
W w
Pi pi1 false false 0 0
Pi pi2 false false 0 0
Pi pi3 false false 0 0
###Players###
R repairman p1 pi3 null
###Neighbors###
c pi1
pi1 pi1
pi1 pi2
w pi2
pi1 p1
pi1 c
pi2 w
pi2 p1
pi3 p1
```

8.2.3.1.12 12.txt

```
###Declaration###
C c
Pu Pump false true
Pi Pipe1 false false 0 0
###Players###
R Repairman1 Pump null null
S Saboteur1 Pump null null
###Neighbors###
c Pipe1
Pump Pipe1
Pipe1 c
Pipe1 Pump
```

8.2.3.1.13 13.txt

```
###Declaration###
Pu p1 false false
C c
W w
Pi pi1 false false 0 0
Pi pi2 false false 0 0
###Players###
S saboteur w null null
###Neighbors###
c pi1
p1 pi1
p1 pi2
w pi2
pi1 p1
pi1 c
pi2 w
pi2 p1
```

8.2.4 Place pump on sticky pipe

- **Leírás**

A szerelő a ciszternán felvesz egy pumpát, átsétál a bekötött csőre, majd lehelyezi ott a nála lévő pumpát. Kapott 1. teszt.

- **Ellenőrzött funkcionális, várható hibahelyek**

A cső vágása esetén az állapotai nem duplikálódnak (ez lenne az elvárt viselkedés), csak az egyik csőnek maradnak meg a korábbi tulajdonságai, a másik cső minden paramétere alapértelmezett lesz.

- **Bemenet**

```
load 1
PickUpPump
move 0
PlacePump
info pi1
info pi3 //az újonnan létrehozott cső
• Elvárt kimenet
Load successful
Successfully picked up holdingPump1
Successfully moved to pi1
Successfully placed holdingPump1
pi1 true true 0 5
pi3 false false 0 0
```

8.2.5 Pick up pump from pipe, pump and cistern

- **Leírás**

Megpróbálunk felvenni pumpát egy csőről, egy pumpáról és egy ciszternáról is. Kapott 2. teszt.

- **Ellenőrzött funkcionális, várható hibahelyek**

Az elvárt viselkedés, hogy ne tudjunk a csőről és a pumpáról pumpát felvenni. A csiszternáról viszont tudjunk.

- **Bemenet**

```
load 2
move 0
PickUpPump
info repairman
move 1
EndTurn //mert három action lehet egy körben
PickUpPump
info repairman
move 0
move 0
EndTurn
PickUpPump
info repairman
```

- **Elvárt kimenet**

```
Load successful
Successfully moved to p1
Can't pick up Pump here
repairman p1 null null
Successfully moved to p1
repairman turn ended repairman's turn next
Can't pick up Pump here
repairman p1 null null
Successfully moved to p1
Successfully moved to c
repairman turn ended repairman's turn next
Successfully picked up holdingPump1
repairman c null holdingPump1
```

8.2.6 Move to a slimey moving pipe

- **Leírás**

Olyan csőre lépünk rá ami csúszós, és a túlsó végét úgy viszik, hogy a játékos aki viszi, az egy csövön áll. Kapott 3. teszt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A mi modellünk szerint ilyenkor, ha az egyik vége a csőnek null-ba van kötve és csúszós, akkor garantáltan a másik irányba csúszik. Ezt szeretnénk tesztelni.

- **Bemenet**

```
load 8
move 0 // a szerelő lép
EndTurn
move 1 // a szabotőr lép rá a cipelt csőre
```

- **Elvárt kimenet**

```
Load successful
Successfully moved to pipeOnGround
repairman turn ended sab's turn next
Slipped to c
```

8.2.7 Lifting both end of a pipe

- **Leírás**

A cső két végének egyszerre cipelése, lehetőleg egy rajta álló játékossal. Az egyik szerelő felveszi a ciszternába egy oldalról kötött csövet, amin áll. Elmozog, majd a szabotőr rálép a csőre. Ezután a másik szerelő felveszi a másik oldalát a csőnek. A szabotőr megpróbál lelépni. Majd a csövet bekötik a 2 pumpába a szerelők. Kapott 4. teszt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A szerelők tudják vinni ide-oda a csövet, ne legyen vele probléma. A szabotőr ne tudjon lelépni a cipelt csőről.

- **Bemenet**

```
load 10
PickUpPipe 0
move 0
move 0
EndTurn
EndTurn
move 1
EndTurn
move 1
move 1
move 1
EndTurn
PickUpPipe 1
move 0
move 1
EndTurn
move 0
EndTurn
PlacePipe
EndTurn
PlacePipe
```

- **Elvárt kimenet**

```

Load successful
Successfully picked up holdingPipe
Successfully moved to c
Successfully moved to pipeOnGround1
r1 turn ended r2's turn next
r2 turn ended sab's turn next
Successfully moved to holdingPipe
sab turn ended r1's turn next
Successfully moved to pump1
Successfully moved to pipeOnGround2
Successfully moved to pump2
r1 turn ended r2's turn next
Successfully picked up holdingPipe
Successfully moved to pipeOnGround1
Successfully moved to pump1
r2 turn ended sab's turn next
Failed to move to null
sab turn ended r1's turn next
Successfully placed holdingPipe
r1 turn ended r2's turn next
Successfully placed holdingPipe

```

8.2.8 Make pipe sticky and trying to move

- **Leírás**

Rálépünk a pi1-es csőre. Bekenjük a csövet ragasztóval. Addig próbálgatunk ellépni onnan, amíg nem sikerül. Majd ezt követően visszalépünk, és újra ellépünk onnan. Kapott 5. teszt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Elvárt viselkedés, hogy ha bekentük a csövet, utána ne tudjunk azonnal ellépni. Kelljen pár próbálkozás mire sikerül lelépni a csőről. Amikor visszalépünk rá, azt várjuk, hogy akkor már simán le tudjunk róla lépni

- **Bemenet**

```

load 2
move 0
stick
info pi1
move 0
EndTurn
move 0
move 0 // ha háromra állítja be sticky-t
info pi1
move 0
EndTurn
move 0

```

- **Elvárt kimenet**

```

Load successful
Successfully moved to p1
p1 is now sticky
p1 false false 0 3
sticked to p1, can't move until: 2
repairman turn ended repairman's turn next
sticked to p1, can't move until: 1
Successfully moved to c
p1 false false 0 0
Successfully moved to p1
repairman turn ended repairman's turn next
Successfully moved to c

```

8.2.9 Move to a sticky and slimy pipe

- **Leírás**

Rálépünk egy egyszerre ragadós és csúszós csőre. Kapott 6. teszt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt várjuk el, hogy a karakterünk meghívja a remove függvényt a NonPipeon, amiről próbál ellépni. Onnan sikeresen törli magát, ezért meghívja az accept függvényt a csövön, ami mivel csúszós, ezért tovább hívja vagy a túlsó vagy az innenső végén lévő elemen az acceptet. Majd ez elfogadja, igazzal tér vissza, ezért már az utolsó remove nem hívódik meg. Technikailag nálunk az történik, hogy ilyen esetben úgy teszünk mintha nem is lépne rá a csőre, mert még mielőtt rálépne, lecsúszik róla. Ezért nincs is mibe beleragadnia a modellünk szerint.

- **Bemenet**

```

load 3
move 0

```

- **Elvárt kimenet**

```

Load successful
Slipped to p1 // vagy Slipped to c

```

8.2.10 Connect new one-end cistern pipe to pump

- **Leírás**

A legalapvetőbb cső átkötés. Egy ciszternából kiinduló, semmibe vezető csövet kötünk egy pumpába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt ellenőrizzük, hogy az újonnan létrejövő csövek a ciszternán megfelelően beköthetőek-e a hálózatba. Azt várjuk el, hogy igen.

- **Bemenet**

```

Load 4
Move 1
Move 0
Move 0
EndTurn
EndTurn
EndTurn
EndTurn
Move 0
EndTurn
EndTurn
EndTurn
EndTurn
EndTurn
PlacePipe
Info p1

- Elvárt kimenet

Load successful
Successfully moved to p1
Successfully moved to c
Successfully moved to p1
s1 turn ended s2 turn next
s2 turn ended r1 turn next
r1 turn ended r2 turn next
r2 turn ended s1 turn next
Successfully moved to p2
s1 turn ended s2 turn next
s2 turn ended r1 turn next
r1 turn ended r2 turn next
r2 turn ended s1 turn next
Successfully placed p1
p1 c p2 false false

```

8.2.11 Game turns, test actions

- Leírás

A játék körre osztása, ütemezése.

- Ellenőrzött funkcionalitás, várható hibahelyek

A játék köreit, annak ütemezését és a játékosok akcióinak sorrendiségét ellenőrizzük. Ellenőrizzük többek között azt, hogy egy szerelő meg tud-e szerezni egy elromlott objektumot, illetve egy szabotőr el tud-e rontani egyet. A 4db endTurn hatására egy teljes kör is véget ér, így a vízszimuláció is lefut. Fontos azonban észrevenni azt, hogy nem állította be senki a tesz során az aktív ki/bemeneti csöveget, így nem jelenik meg víz a rendszerben.

- Bemenet

```

Load 4
Move 0
Stab
Move 0
EndTurn
Move 0
Move 0
Stab
EndTurn
Move 0
Repair
Move 0
EndTurn
Move 0
Move 0
Repair
EndTurn
Info pi2
Info p1

```

- Elvárt kimenet

```

Load successful
Successfully moved to pi2
Successfully stabbed pi2
Successfully moved to p1
s1 turn ended s2 turn next
Successfully moved to pi2
Successfully moved to p1
Successfully stabbed p1
s2 turn ended r1 turn next
Successfully moved to pi2
Successfully repaired pi2
Successfully moved to p1
r1 turn ended r2 turn next
Successfully moved to pi2
Successfully moved to p1
Successfully repaired p1
r2 turn ended s1 turn next
p1 true false
pi1 c p2 false false false false

```

8.2.12 Place Pump on Pipe

- Leírás

Pumpa elhelyezése egy csőre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ebben a tesztesetben azt ellenőrizzük, hogy lehelyezhető-e egy pumpa egy csőre, annak hatására kettévágódik-e egy cső, azaz létrejön-e még egy cső.

- **Bemenet**

```
Load 5
EndTurn
EndTurn
PickUpPump
Move 0
PlacePump
Info pi3
Info pi4
```

- **Elvárt kimenet**

```
Load successful
s1 turn ended s2 turn next
s2 turn ended r1 turn next
Successfully picked up p3 //p3 automatikusan létrejött a
WaterSourceon
Successfully moved to pi3
Successfully placed p3
pi3 p1 p3 false false 0 0
pi4 p3 w false false 0 0
```

8.2.13 Place Pump on Pump

- **Leírás**

Pumpa elhelyezése egy pumpára.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt ellenőrizzük, hogy a játék megtagadja-e pumpa elhelyezését pumpára, ami egy nem megengedett művelet.

- **Bemenet**

```
Load 5
EndTurn
EndTurn
PickUpPump
Move 0
Move 0
EndTurn
EndTurn
EndTurn
EndTurn
PlacePump
```

- **Elvárt kimenet**

```

Load successful
s1 turn ended s2 turn next
s2 turn ended r1 turn next
Successfully picked up p3
Successfully moved to pi3
Successfully moved to p1
r1 turn ended r2 turn next
r2 turn ended s1 turn next
s1 turn ended s2 turn next
s2 turn ended r1 turn next
Can't place p3 on p1

```

8.2.14 Place Pump on Pipe without Pump in hand

- **Leírás**

Pumpa elhelyezése egy csőre anélkül, hogy a karakter kezében lenne egy pumpa.

- **Ellenőrzött funkcionális, várható hibahelyek**

Azt ellenőrizzük, hogy a játék megtagadja-e egy pumpa elhelyezését egy csőre, amennyiben a játékosnál semmilyen pumpa sincs. Azt várjuk, hogy a játék megtagadja ezt a műveletet.

- **Bemenet**

```

Load 5
EndTurn
EndTurn
Move 0
PlacePump
Info pi3

```

- **Elvárt kimenet**

```

Load successful
s1 turn ended s2 turn next
s2 turn ended r1 turn next
Successfully moved to pi3
No placeable Pump
pi3 p1 w false false false false

```

8.2.15 Saboteur Invalid Action and Out of Action

- **Leírás**

Szabotőr megpróbál megfoltozni egy csövet, majd egy körben megpróbál túl sok action-t csinálni.

- **Ellenőrzött funkcionális, várható hibahelyek**

Azt ellenőrizzük, hogy a szabotőr képes-e javítani, továbbá ki tud-e fogyni action-ökből.

- **Bemenet**

```

Load 6
Repair
Stab
Stick
Slime
Move 1

- Elvárt kimenet

Load successful
Not valid action for Saboteur
Successfully stabbed Pipe1
Pipe1 is now sticky
Pipe1 is now slimey
Out of Action

```

8.2.16 Repairman moves pipe

- **Leírás**

Szerelő sikeresen áthelyez egy csövet egyik pumpáról egy másikra

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt ellenőrizzük, hogy a szabotőr át tud-e helyezni egy csövet az egyik pumpáról a másikra.

Azt várjuk, hogy sikeresen át tud helyezni.

- **Bemenet**

```

Load 9
PickUpPipe 0
Move 0
PlacePipe
Info p1
Info p2

- Elvárt kimenet

Load successful
Successfully picked up p1
Successfully moved to p3
Successfully placed p1
p1 p1 p3 false false
pi2 p2 p3 false false

```

8.2.17 Connect loop to pump,cistern,watersource

- **Leírás**

Hurokél bekötése pumpába cisternába és forrásba

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Eredményként azt várjuk, hogy csak a forráson nem sikerül a hurokél bekötése

- **Bemenet**

```

load 11
PlacePipe
info pi3
PickUpPipe 0
Move 0
PlacePipe
Move 0
PickUpPipe 0
Move 0
PlacePipe
Move 0
Move 1
PlacePipe
Move 0
Move 0
PickUpPipe 0
Move 0
Move 1
PlacePipe

```

- Elvárt kimenet

```

Load successful
Successfully placed pi3
pi3 p1 p1 false false
Successfully picked up pi3
Successfully moved to w
Successfully placed pi3
Successfully moved to p1
Successfully picked up pi3
Successfully moved to w
Can't place pi3 on w
Successfully moved to p1
Successfully moved to c
Successfully placed pi3
Successfully moved to p1
Successfully moved to w
Successfully picked up pi3
Successfully moved to p1
Successfully moved to c
Successfully placed pi3

```

8.2.18 Saboteur tries to break unbreakable pipe

- Leírás

Szabotőr kilúkaszt egy csövet, azt megjavítják majd megpróbálja újra kilúkasztani.

- Ellenőrzött funkcionalitás, várható hibahelyek

Eredményként azt várjuk hogy a második lúkasztás már ne legyen sikeres.

- **Bemenet**

```
Load 12
```

```
EndTurn
```

```
Move 0
```

```
Stab
```

```
Move 0
```

```
EndTurn
```

```
Move 0
```

```
Repair
```

```
Move 0
```

```
EndTurn
```

```
Move 0
```

```
Stab
```

- **Elvárt kimenet**

```
Load successful
```

```
Repairman1 turn ended Saboteur1's turn next
```

```
Successfully moved to Pipe1
```

```
Successfully stabbed Pipe1
```

```
Successfully moved to Pump
```

```
Saboteur1 turn ended Repairman1's turn next
```

```
Successfully moved to Pipe1
```

```
Successfully repaired Pipe1
```

```
Successfully moved to Pump
```

```
Repairman1 turn ended Saboteur1's turn next
```

```
Successfully moved to Pipe1
```

```
Failed to stab Pipe1, unbreakable until: 4
```

8.2.19 Repairman slipped to an object

- **Leírás**

Szabotőr csúszóssá tesz egy csövet majd egy szerelő megpróbál rálépni

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

```
Load 12
```

```
EndTurn
```

```
Move 0
```

```
Slime
```

```
Move 0
```

```
Endturn
```

```
Move 0
```

- **Elvárt kimenet**

```

Load successful
Repairman1 turn ended Saboteur1's turn next
Successfully moved to Pipe1
Pipe1 is now slimy
Successfully moved to Pump
Saboteur1 turn ended Repairman1's turn next
Slipped to Pump

```

8.2.20 Sticky and slimy pump,cistern, watersource

- Leírás

Szabotőr megpróbál ragacsossá illetve csúszóssá tenni egy pumpát, ciszternát, forrást

- Ellenőrzött funkcionalitás, várható hibahelyek

Eredményként azt várjuk, hogy egyik művelet se legyen sikeres

Bemenet

```
Load 13
```

```
Slime
```

```
Stick
```

```
Move 0
```

```
Slime
```

```
Stick
```

```
Move 0
```

```
Slime
```

```
Stick
```

- Elvárt kimenet

```
Load successful
```

```
w can't be slimey
```

```
w can't be sticky
```

```
Successfully moved to p1
```

```
p1 can't be slimey
```

```
p1 can't be sticky
```

```
Successfully moved to c
```

```
c can't be slimey
```

```
c can't be sticky
```

8.2.21 Same source and destination of water

- **Leírás**

Egy pumpa ki- és bemenetének próbáljuk meg ugyanazt a csövet megadni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy lehet-e egy pumpán állva beállítani aktív ki- és bemenetnek ugyanazon csövet. Eredményül azt várjuk, hogy ez a művelet sikeresen megtörténik.

Egy kivétel van amikor forrásba próbálunk vizet pumpálni (következő teszt).

- **Bemenet**

```
Load 2
```

```
Move 0
```

```
Move 1
```

```
Src 0 0
```

- **Elvárt kimenet**

```
Load successful
```

```
Successfully moved to p1
```

```
Successfully moved to p1
```

```
Input set p1, Output set p1
```

8.2.22 Pump water to the WaterSource

- **Leírás**

Megpróbálunk vizet pumpálni a forrásba.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy lehet-e egy a forrásba vizet pumpálni.

Azt várjuk, hogy egy olyan pumpán állva amit egy cső köt össze a forrással, nem fogjuk tudni beállítani aktív kimenetnek a forrás felé vezető csövet. Ez azt is jelenti, hogy forrásnál nincs hurokél sem.

- **Bemenet**

```
Load 6
```

```
Move 0
```

```
Move 1
```

```
Src 1 1 //ez a Pipe2
```

- **Elvárt kimenet**

```
Load successful
```

```
Successfully moved to Pipe1
```

```
Successfully moved to Pump1
```

```
Invalid parameters, nothing changed
```

8.2.23 Saboteur tries to pick up Pump from Cistern

- **Leírás**

Szabotőr megpróbál ciszternából pumpát felvenni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy lehet-e szabotőrnak felvennie pumpát egy ciszternán állva.

Az várjuk, hogy ez ne történhessen meg és jelezze is a program.

- **Bemenet**

```

Load 12
EndTurn
PickUpPump

- Elvárt kimenet

Load successful
Repairman1's turn ended Saboteur1's turn next
Saboteur can't pick up a Pump

```

8.2.24 Repairman tries to place pipe without having a pipe

- **Leírás**

Szerelő megpróbál letenni csövet, úgy hogy nincs nála.

- **Ellenőrzött funkcionális, várható hibahelyek**

Ellenőrizzük, hogy lehet-e szerelőnek úgy csövet lerakni, hogy nincs nála.

Azt várjuk, hogy nem lehessen.

- **Bemenet**

```
Load 2
```

```
PlacePipe
```

- **Elvárt kimenet**

```
Load successful
```

```
No Pipe to place
```

8.2.25 Repairman tries to make Pipe slimy

- **Leírás**

Szerelő megpróbál csúszóssá tenni egy csövet.

- **Ellenőrzött funkcionális, várható hibahelyek**

Ellenőrizzük, hogy lehet-e szerelőnek csúszóssá tenni egy csövet.

Azt várjuk, hogy ez nem fog sikerülni.

- **Bemenet**

```
Load 2
```

```
Move 0
```

```
Slime
```

- **Elvárt kimenet**

```
Load successful
```

```
Successfully moved to pil
```

```
Repairman can't make it slimy
```

8.2.26 Adjust same pipe

- **Leírás**

Az egyik csapat játékosai megpróbálják felvenni egy csőnek ugyanazon végét.

- **Ellenőrzött funkcionális, várható hibahelyek**

Ellenőrizzük, hogy ha egy karakter felemeli egy cső egyik végét és ugyanezzel a számmal megpróbálja az öt követő karakter is akkor ez nem fog sikerülni.

- **Bemenet**

```

Load 4
PickUpPipe 0
EndTurn
PickUpPipe 0
• Elvárt kimenet
Load successful
Successfully picked up pil
s1's turn ended s2's turn next
Failed to pick up Pipe

```

8.2.27 Repairman repairs WaterSource

- **Leírás**
Szerelő megpróbál megjavítani egy forrást.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A Forrás nem megjavítható.
- **Bemenet**

```

Load 5
Repair

```

- **Elvárt kimenet**

```

Load successful
Source1 unrepairable

```

8.2.28 Adjust same and invalid input

- **Leírás**
A Szerelő megpróbálja a ki és bemenetet ugyan arra a csőre állítani. A Szabotőr pedig érvénytelen számot ad meg paraméterül.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Az első Adjust sikeres, de a második nem.
- **Bemenet**

```

Load 7
Adjust 0 1
Adjust 0 0
EndTurn
Adjust 1 0
Adjust -10 2

```

- **Elvárt kimenet**

```

Load successful
Input set Pipe1, Output set Pipe2
Input set Pipe1, Output set Pipe1
Repairman1 turn ended Saboteur1's turn next
Input set Pipe2, Output set Pipe1
Invalid parameters, nothing changed

```

8.2.29 Load and Save

- **Leírás**

Betöltünk egy mentést, elmentjük más néven és összehasonlítjuk a tartalmukat.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A fájl tartalma nem változhat.

- **Bemenet**

Load 1

Save 10

- **Elvárt kimenet**

Load successful

8.2.30 Help

- **Leírás**

Kiadjuk a Help parancsot Szerelővel, majd Szabotőrrel is.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A megfelelő kimenetnek kell megjelennie.

- **Bemenet**

Load 7

Help

EndTurn

Help

- Elvárt kimenet

```
Load successful
Move dir
Adjust src dest
Repair
Stab
PickUpPump
PickUpPipe pipe
PlacePump
PlacePipe
Stick
EndTurn
Save file
Load file
Help
ToggleRandom I/N
Repairman1 turn ended Saboteur1's turn next
Move dir
Adjust src dest
Stab
Stick
Slime
EndTurn
Save file
Load file
Help
ToggleRandom I/N
```

8.2.31 ToggleRandom

- **Leírás**

Betöltünk egy mentést, invalid inputot adunk, majd bekapcsoljuk a random eseményeket. Befejezzük a kört majd az aktuális állapotot elmentjük. Újratöltsük a mentést, kikapcsoljuk a random eseményeket, befejezzük a körünket, majd elmentjük ezt az állapotot is.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az első mentésnél a Pumpák valamennyi része fog elromlani a random esemény miatt, míg a második mentésnél az összes el lesz romolva. Ezeket a mentési fájlból ellenőrizhetjük.

- **Bemenet**

```
Lood 6
ToggleRandom 2000
ToggleRandom I
EndTurn
Save RandomEnabled
Lood 6
ToggleRandom N
EndTurn
Save RandomDisabled
```

- **Elvárt kimenet**

```
Load successful
Invalid input
Random events Enabled
Repairman1 turn ended Saboteur1's turn next
Load successful
Random events Disabled
Saboteur1 turn ended Repairman1's turn next
```

8.2.32 Invalid Move

- **Leírás**

Olyan helyekre akarunk mozogni ami nem létezik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A mozgás sikertelen ezekre a bemenetekre.

- **Bemenet**

```
Load 1
Move 100
Move -99
```

- **Elvárt kimenet**

```
Load successful
Failed to move to Object
Failed to move to Object
```

8.3 A tesztelést támogató programok tervezése

- A tesztadatokat tartalmazó fájlokat egy egyszerű szövegszerkesztő program segítségével előállíthatjuk. További külön programot nem igényel a tesztek létrehozása.
- Az elvárt kimenetet a prototípuson belül összehasonlíthatjuk a dokumentumban megadott elvárt kimenetekkel manuálisan, továbbá az olyan teszteknél, ahol ki is mentünk állapotokat, a windows rendszerek parancssorában elérhető **fc** (file compare) parancsot alkalmazva elvégezhetjük a régebbi és újabb fájlok módosítását.

8.4 Napló

Kezdet	Időtartam	Résznevők	Leírás
2023.04.28 11:00	1 óra	Andai, Bodnár, Gutási, Sziksai	Értekezlet. Döntés: Használhatatlan a Google Docs
2023.04.29 9:00	1 óra	Andai	Komplexebb tesztesetek kitalálása, leírása az ötletek közé
2023.04.29 9:30	2 óra	Gutási	Game és DestuctionTimer osztály leírásának és pszeudokódjainak elkészítése
2023.04.29 10:30	2 óra	Bodnár	Character, Repairman és Saboteur osztályok leírásainak és pszeudokódjainak elkészítése
2023.04.29 13:00	3.5 óra	Andai	Place, RepairmanPlace, SaboteurPlace, Element, Pipe osztályok és interfészek leírásainak elkészítése, és az algoritmusok pszeudokódjainak megírása
2023.04.29 10:00	2 óra	Bodnár	Tesztesetek kitalálása, Character osztály javítása, bővítése
2023.04.30 16:00	2 óra	Sziksai	NonPipe, Cistern, WaterSource és Pump osztályok leírásának és pszeudokódjának elkészítése
2023.05.01 12:00	1,5 óra	Andai	8.2.2,-3,-6,-7 tesztek elkészítése
2023.04.29 9:30	2 óra	Gutási	8.2.8-8.2.12 tesztek elkészítése

2023.05.01. 17:30	1,5 óra	Bodnár	8.2.19,8.2.20-8.2.24 tesztek elkészítése
2023.05.01. 17:30	1,5 óra	Szikszai	8.2.25-30 és 8.2.13-as tesztek elkészítése
2023.05.01 20:30	1 óra	Andai	8.2.4 és 8.2.5 tesztek elkészítése
2023.05.01 21:00	2 óra	Vajda	8.2.14, és 8.2.15 tesztek elkészítése, betölthető pályák hozzáadása
2023.05.02. 10:00	1,5 óra	Bodnár	8.2.20-as teszt elkészítése, osztályok javítása, tesztesetek javítása, bővítése
2023.05.02 18:45	1 óra	Andai	Pipe remove függvényének újragondolása a stick miatt, kiíratások beírása a Pipe osztály függvényeibe, 8.2.7 újragondolása a megváltozott move függvény miatt
2023.05.02 19:00	2 óra	Vajda	8.2.16, 8.2.17 és 8.2.18 tesztek elkészítése, osztályok, tesztesetek javítása
2023.05.03 5:30	1 óra	Gutási	8.3 megírása, dokumentum rendezése, előkészítése nyomtatásra,

9. Prototípus beadása

9.1 Fordítási és futtatási útmutató

9.1.1 Fájllista

Fájl neve	Méret (B)	Keletkezés ideje	Tartalom
App.java	158	2023.05.12 22:56	Az alkalmazás indítása, leállítása
Character.java	466	2023.05.12 22:56	Character osztály
Cistern.java	3051	2023. 05. 14. 22:27	Cistern osztály
DestructionTimer.java	1762	2023. 05. 14. 23:05	DestructionTimer osztály
Element.java	4317	2023. 05. 14. 22:27	Element osztály
Game.java	8368	2023. 05. 14. 23:05	Game osztály
NonPipe.java	1861	2023. 05. 14. 22:27	NonPipe osztály
Pair.java	314	2023. 04. 16. 10:22	Pair osztály
Pipe.java	8460	2023. 05. 14. 22:27	Pipe osztály
Place.java	581	2023. 05. 12. 22:56	Place osztály
Prototype.java	19946	2023. 05. 14. 23:05	Prototype osztály
Pump.java	4175	2023. 05. 14. 22:27	Pump osztály
Repairman.java	4424	2023. 05. 14. 22:27	Repairman osztály
RepairmanPlace.java	565	2023. 05. 14. 22:56	RepairmanPlace osztály
Saboteur.java	2933	2023. 05. 14. 22:27	Saboteur osztály
SaboteurPlace.java	183	2023. 05. 14. 22:56	SaboteurPlace osztály
SaboteurPointSource.java	191	2023. 04. 16. 15:43	SaboteurPointSource osztály
Skeleton.java	18989	2023. 04. 17. 7:50	Skeleton osztály
Tabulator.java	333	2023. 04. 16. 16:03	Tabulator osztály
Watersource.java	1566	2023. 05. 14. 22:27	Watersource osztály
1.txt	182	2023. 05. 12. 22:56	A load 1 parancs során betöltendő adathalmazokat tartalmazza

2.txt	155	2023. 05. 13. 11:27	A load 2 parancs során betöltendő adathalmazokat tartalmazza
3.txt	155	2023. 05. 14. 22:27	A load 3 parancs során betöltendő adathalmazokat tartalmazza
4.txt	394	2023. 05. 15. 00:56	A load 4 parancs során betöltendő adathalmazokat tartalmazza
5.txt	394	2023. 05. 13. 11:39	A load 5 parancs során betöltendő adathalmazokat tartalmazza
6.txt	478	2023. 05. 13. 14:12	A load 6 parancs során betöltendő adathalmazokat tartalmazza
7.txt	294	2023. 05. 13. 14:12	A load 7 parancs során betöltendő adathalmazokat tartalmazza
8.txt	275	2023. 05. 14. 22:27	A load 8 parancs során betöltendő adathalmazokat tartalmazza
9.txt	242	2023. 05. 14. 23:55	A load 9 parancs során betöltendő adathalmazokat tartalmazza
10.txt	417	2023. 05. 13. 11:41	A load 10 parancs során betöltendő adathalmazokat tartalmazza
11.txt	241	2023. 05. 15. 00:42	A load 11 parancs során betöltendő adathalmazokat tartalmazza
12.txt	207	2023. 05. 13. 11:42	A load 12 parancs során betöltendő adathalmazokat tartalmazza
13.txt	220	2023. 05. 13. 11:42	A load 13 parancs során betöltendő adathalmazokat tartalmazza
14.txt	394	2023. 05. 15. 00:57	A load 14 parancs során betöltendő adathalmazokat tartalmazza

9.1.2 Fordítás

A forrásfájlok fordításához a forrásfájlokat tartalmazó mappában, a javac App.java parancsot kell kiadni. Ehhez szükséges, hogy a fordító gépen telepítve legyen Java 8-hoz tartozó fejlesztői csomag (JDK).

9.1.3 Futtatás

A program futtatásához a lefordított .class fájlokat tartalmazó mappában, használjuk a java App parancsot. Ennek hatására a program elindul. A felhasznált képfájlok helyezkedjenek el a fordított .class fájlokat tartalmazó mappán belül az **img** mappában.

9.2 Tesztek jegyzőkönyvei

9.2.1 Place pump on sticky pipe

Tesztelő neve	Andai
Teszt időpontja	2023.05.14. 19:01

9.2.2 Pick up pump from pipe, pump and cistern

Tesztelő neve	Andai
Teszt időpontja	2023.05.14. 19:03

9.2.3 Move to a slimy moving pipe

Tesztelő neve	Andai
Teszt időpontja	2023.05.14. 19:04

9.2.4 Lifting both end of a pipe

Tesztelő neve	Andai
Teszt időpontja	2023.05.14. 19:08

9.2.5 Make pipe sticky and trying to move

Tesztelő neve	Andai
Teszt időpontja	2023.05.14. 18:58

Tesztelő neve	Andai
Teszt időpontja	2023.05.14. 18:53
Teszt eredménye	A szerelő miután beragacsozza a csövet, azonnal beleragad.
Lehetséges hibaok	A stick függvény hibás implementációja.
Változtatások	A stick függvény a sticky-t -3-ra állítja, és a remove állítja 3-ra ha -3.

9.2.6 Move to a sticky and slimy pipe

Tesztelő neve	Andai
Teszt időpontja	2023.05.14. 18:59

9.2.7 Connect new one-end cistern pipe to pump

Tesztelő neve	Gutási
---------------	--------

Teszt időpontja	2023.05.15 1:00
------------------------	-----------------

Tesztelő neve	Gutási
Teszt időpontja	2023.05.15 0:57
Teszt eredménye	A javított parancssorozattal sikeresen át tudunk helyezni egy új csövet egy pumpára.
Lehetséges hibaok	Atesztelest inicializáló fájlban történt hibás pályaleírás, továbbá a pályaleírásban változott sorrendek nem megfelelő eredményt okoztak.
Változtatások	<p>A teszt során a következő parancssorozatot adjuk ki a prototypéra:</p> <pre>load 14 move 0 move 0 pickupipe 0 endturn endturn endturn endturn move 0 move 1 move 0 endturn endturn endturn endturn move 1 placepipe save output</pre>

9.2.8 Game turns, test actions

Tesztelő neve	Gutási
Teszt időpontja	2023.05.14 23:10

Tesztelő neve	Gutási
Teszt időpontja	2023.05.14 23:08
Teszt eredménye	A körkre osztály működik, a karakterek tudnak mozogni, tönkretenni és megfogozni objektumokat
Lehetséges hibaok	Az eredeti teszt hibás implementációja.
Változtatások	<p>A teszt során a következő parancssorozatot adjuk ki a prototypéra:</p> <pre>load 4 endturn endturn move 1 stab move 1</pre>

	<pre>endturn move 1 move 1 stab endturn move 1 repair move 1 save output</pre> <p>Az így módosul elvárt eredmény ez, hogy a szomszédok nem változnak, viszont a karakterek pozíciói igen (repairman1: p1, repairman2: c, saboteur1: p1, saboteur2: p1) (Ez ellenőrizhető a megfelelő mappában a fc 4.txt output.txt parancs kiadásával).</p>
--	--

9.2.9 Place Pump on Pipe

Tesztelő neve	Gutási
Teszt időpontja	2023.05.14. 23:28

Tesztelő neve	Gutási
Teszt időpontja	2023.05.14. 23:26
Teszt eredménye	A pumpát sikeresen lehelyeztük a kiválasztott csőre, az új köztes cső is sikeresen létrejött.
Lehetséges hibaok	Az eredeti teszt hibás implementációja.
Változtatások	A parancssorozat, amivel tesztelünk a következő: <pre>load 5 move 0 move 1 move 0 move 0 endturn endturn endturn endturn move 0 pickuppump move 1 endturn endturn endturn endturn placepump save output</pre> <p>Ezekben az fc 5.txt output.txt parancssal összehasonlíthatjuk a két fájlt. Megfigyelhetjük az újonnan létrejött pumpát és csövet.</p>

9.2.10 Place Pump on Pump

Tesztelő neve	Gutási
Teszt időpontja	2023.05.14. 23:42
Teszt eredménye	A pumpát nem sikerült letenni egy másik pumpán.
Lehetséges hibaok	Az eredeti teszt hibás implementációja.
Változtatások	<p>A parancssorozat, amivel tesztelünk, a következő:</p> <pre>load 5 move 0 move 1 move 1 endturn endturn endturn endturn move 0 move 0 pickuppump endturn endturn endturn endturn move 1 move 1 placepump save output</pre> <p>A fc 5.txt output.txt hatására megfigyelhető, hogy az outputban a létrehozott pumpa repairman1 kezében van, de senkinek sem szomszédja, így nem jelenik meg a szomszédok listájában.</p>

9.2.11 Place Pump on Pipe without Pump in hand

Tesztelő neve	Gutási
Teszt időpontja	2023.05.14 23:47

Tesztelő neve	Gutási
Teszt időpontja	2023.05.13 23:46
Teszt eredménye	Nem lehetett úgy pumpát lerakni, hogy az nincs a kezünkben.
Lehetséges hibaok	Az eredeti teszeset hibás implementációja
Változtatások	<p>A módosított parancssorozat:</p> <pre>load 5 move 1 move 0 placepump save output</pre>

9.2.12 Saboteur Invalid Action and Out of Action

Tesztelő neve	Gutási
Teszt időpontja	2023.05.14 23:52

Tesztelő neve	Gutási
Teszt időpontja	2023.05.14 23:49
Teszt eredménye	A fc 6.txt output.txt hatására látható, hogy a szabotőr sikeresen módosította a cső paramétereit, és átlépett egy pumpára
Lehetséges hibaok	Az eredeti teszt 4 lépés/körrel számolt. Ez 3-ra változott, így ezalapján át kellett minimálisan írni a parancssorozatot:
Változtatások	A javított parancssorozat: load 6 repair stab stick endturn slime move 1 save output

9.2.13 Repairman moves pipe

Tesztelő neve	Gutási
Teszt időpontja	2023.05.14 23:56

Tesztelő neve	Gutási
Teszt időpontja	2023.05.14 23:54
Teszt eredménye	pi2 cső egyik vége átkerült p2-ről p1-re
Lehetséges hibaok	A tesztesetet inicializáló fájl módosult, így módosítani kellett a tesztelésen is.
Változtatások	Az eredeti teszttel ellentétben itt pi2 p2-beli végét vesszük fel, p1-re lépünk, majd bekötjük azt. Ennek hatására módosul pi2 két végének helye. A módosított parancssorozat: load 9 pickupipe 1 move 0 move 0 endturn placepipe save output

9.2.14 Connect loop to pump,cistern,watersource

Tesztelő neve	Vajda
Teszt időpontja	2023.05.15 0:40

Tesztelő neve	Vajda
Teszt időpontja	2023.05.15 0:32
Teszt eredménye	Az összeköttetések egyes helyeken más sorrendben vannak, illetve nem veszi figyelembe a teszt a maximum lépések számát
Lehetséges hibaok	A teszt létrehozása óta változott a 11.txt fájl tartalma, így kicsit eltérő a tesztelt pálya.
Változtatások	<p>A javított teszthez tartozó parancssorrend:</p> <pre>load 11 placepipe pickuppipe 2 move 1 endturn move 0 placepipe move 0 endturn move 1 pickuppipe 2 move 1 endturn move 0 placepipe // nem ir ki itt semmit de amugy nem negedi a lerakast move 0 move 1 endturn move 0 move 1 placepipe endturn move 0 move 0 mvoe 1 endturn move 0 pickuppipe 2 move 0 endturn move 1 move 0 move 1 endturn placepipe</pre>

9.2.15 Saboteur tries to break unbreakable pipe

Tesztelő neve	Vajda
Teszt időpontja	2023.05.15 0:03

Tesztelő neve	Vajda
Teszt időpontja	2023.05.15 0:00

Teszt eredménye	Pumpal nem találkozunk a teszt során, Pipe1-ről rögtön c-re lépnek a karakterek
Lehetséges hibaok	A teszt létrehozása óta változott a 12.txt fájl tartalma, így kicsit eltérő a tesztelt pálya.
Változtatások	A javított bemeneti parancsok sorrendje: load 12 endturn move 0 stab move 0 endtirn move 0 repair move 0 endturn move 0 stab save output

9.2.16 Repairman slipped to an object

Tesztelő neve	Vajda
Teszt időpontja	2023.05.15 0:12

Tesztelő neve	Vajda
Teszt időpontja	2023.05.15 0:10
Teszt eredménye	Eltérő az eredmény a 8. dokumentumhoz képest, a szabotőr nem lép vissza a Pumpára
Lehetséges hibaok	A teszt létrehozása óta változott a 8.txt fájl tartalma, így kicsit eltérő a tesztelt pálya.
Változtatások	A javított bemeneti parancsok sorrendje: load 12 enturn move 0 slime move 1 endturn move 0 save output

9.2.17 Sticky and slimy pump,cistern, watersource

Tesztelő neve	Vajda
Teszt időpontja	2023.05.15 0:20

Tesztelő neve	Vajda
Teszt időpontja	2023.05.15 0:16

Teszt eredménye	Az eredeti teszthez képest apróbb módosításokkal sikeresen végigléptettünk a pályán egy szabotőrt, aki megpróbált minden elemet ragacsossá és csúszóssá tenni.
Lehetséges hibaok	A teszt létrehozása óta változott a 13.txt fájl tartalma, így kicsit eltérő a tesztelt pálya.
Változtatások	A javított bemeneti parancsok sorrendje: load 13 slime stick move 0 endturn move 1 slime stick endturn move 0 move 1 endturn slime stick save output

9.2.18 Same source and destination of water

Tesztelő neve	Bodnár
Teszt időpontja	2023.05.14. 20:57

Tesztelő neve	Bodnár
Teszt időpontja	2023.05.14. 20:57
Teszt eredménye	load 2 2.txt Load successful 2Points.txt Load Successful move 0 Successfully moved to p1 move 1 Successfully moved to p1 adjust 0 0 Input set p1 Output set p1

9.2.19 Pump water to the WaterSource

Tesztelő neve	Bodnár
Teszt időpontja	2023.05.14. 21:00

Tesztelő neve	Bodnár
Teszt időpontja	2023.05.14. 21:00

Teszt eredménye	Sikeresen beállítható kimenetnek is az a cső, mert a forrás attól függetlenül nem fogad be vizet. load 6 6.txt Load successful 6Points.txt Load Successful move 1 Successfully moved to Pump1 adjust 1 1 Input set Pipe2 Output set Pipe2
------------------------	--

9.2.20 Saboteur tries to pick up Pump from Cistern

Tesztelő neve	Bodnár
Teszt időpontja	2023.05.14. 21:08

Tesztelő neve	Bodnár
Teszt időpontja	2023.05.14. 21:08
Teszt eredménye	load 12 12.txt Load successful 12Points.txt Load Successful Endturn Repairman1's turn ended, Saboteur1's turn next pickuppump Ilyen muveletre nem képes a jatekos

9.2.21 Repairman tries to place pipe without having a pipe

Tesztelő neve	Bodnár
Teszt időpontja	2023.05.14. 21:10

Tesztelő neve	Bodnár
Teszt időpontja	2023.05.14. 21:10
Teszt eredménye	load 2 2.txt Load successful 2Points.txt Load Successful placepipe Nullptr: java.lang.NullPointerException (javítás előtt)
Lehetséges hibaok	Nem volt kezelve, ha nullpointert ad át lerakni való csőnek.
Változtatások	Repairman PlacePipe metódus javítása.

9.2.22 Repairman tries to make Pipe slimy

Tesztelő neve	Bodnár
Teszt időpontja	2023.05.14. 21:14

Tesztelő neve	Bodnár
Teszt időpontja	2023.05.14. 21:08
Teszt eredménye	load 2 2.txt Load successful 2Points.txt Load Successful move 0 Successfully moved to pi1 slime Ilyen muveletre nem képes a jatekos

9.2.23 Adjust same pipe

Tesztelő neve	Bodnár
Teszt időpontja	2023.05.14. 21:20

Tesztelő neve	Bodnár
Teszt időpontja	2023.05.14. 21:25
Teszt eredménye	IndexOutOfBoundsException load 4 4.txt Load successful 4Points.txt Load Successful pickuppipe 1 Successfully picked up pi2 endturn repairman1's turn ended, repairman2's turn next pickuppipe 1 Invalid object to pick up (ez kivételkezelés után)
Lehetséges hibaok	Nincs kezelve az amikor egy nem létező szomszéd számát adjuk meg, létező listaelemet akarunk elérni.
Változtatások	A lift metódusok implementációinál kivételkezelés.

9.2.24 Repairman repairs WaterSource

Tesztelő neve	Szikszai
Teszt időpontja	2023.05.14. 23:20

Tesztelő neve	Szikszai
Teszt időpontja	2023.05.14. 23:20
Teszt eredménye	5.txt Load successful w unrepairable
Változtatások	Source1 helyett a WaterSource neve w

9.2.25 Adjust same and invalid input

Tesztelő neve	Szikszai
----------------------	----------

Teszt időpontja	2023.05.14. 23:15
------------------------	-------------------

Tesztelő neve	Szikszai
Teszt időpontja	2023.05.14. 23:15
Teszt eredménye	A cső bemenetének állítása sikerül amíg nem adunk meg paraméterek negatív számot (vagy túl nagy számot)

9.2.26 Load and Save

Tesztelő neve	Szikszai
Teszt időpontja	2023.05.14. 23:10

Tesztelő neve	Szikszai
Teszt időpontja	2023.05.14. 23:10
Teszt eredménye	A két fájl lényegi tartalma megegyezik

9.2.27 Help

Tesztelő neve	Szikszai
Teszt időpontja	2023.05.14. 23:00

Tesztelő neve	Szikszai
Teszt időpontja	2023.05.14. 23:00
Teszt eredménye	7.txt Load successful adjust int int move int stab repair pickuppump pickuppipe int placepump placepipe stick endturn save fileName load fileName help info string togglerandom I/N q Repairman1's turn ended, Saboteur1's turn next adjust int int move int stab stick slime endturn save fileName

	load fileName help info string togglerandom I/N q
Változtatások	A parancsokat kis és nagy betűkkel is elfogadjuk

9.2.28 ToggleRandom

Tesztelő neve	Szikszai
Teszt időpontja	2023.05.14. 22:50

Tesztelő neve	Szikszai
Teszt időpontja	2023.05.14. 22:50
Teszt eredménye	6.txt Load successful Random events Enabled Saboteur1's turn ended, Saboteur1's turn next Save Successful 6.txt Load successful Random events Disabled Saboteur1's turn ended, Saboteur1's turn next Save Successful
Lehetséges hibaok	A random generátor elrontja az összes pumpát így nem lesz különbség a 2 fájl tartalma között.

9.2.29 Invalid Move

Tesztelő neve	Szikszai
Teszt időpontja	2023.05.14. 23:25

Tesztelő neve	Szikszai
Teszt időpontja	2023.05.14. 23:25
Teszt eredménye	1.txt Load successful Failed to move, invalid index. Failed to move, invalid index.
Változtatások	A kimenet változott: Failed to move to Object-ról, Failed to move, invalid index.-re

9.3 Értékelés

A projekt tagjai aláírásukkal igazolják, hogy az alább szereplő százalékarányok valósak és elfogadják azokat.

Tag neve	Tag neptun	Munka százalékban	Aláírás
Andai Roland	H3G1PQ	21.44%	

Bodnár Márk	AAL25F	18.10%	
Gutási Ádám	XW53QZ	19.33%	
Szikszai Levente	AENWDT	22.85%	
Vajda András	M7LWCB	18.28%	

9.4 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2023.05.11. 12:00	1 óra	Andai, Bodnár, Gutási, Sziksai, Vajda	Értekezlet. Döntés: Andai és Sziksai elkészítik a Prototype osztályt, a többiek átvezetik az osztályokba a módosításokat, hogy megfeleljenek a Prototype teszteknek.
2023.05.11. 18:00	2 óra	Bodnár	Character, Repairman, Saboteur, osztályok új metódusainak megvalósítása, ezekből fakadó egyéb változtatások.
2023.05.11. 18:00	4 óra	Sziksai	Prototype osztály létrehozása, load és save függvények megvalósítása, pályaelemek, játékosok és game osztályok toString metódusainak és paraméteres konstruktorainak megvalóstása.
2023.05.12. 14:00	2 óra	Andai	A prototype command felületének megvalósítása.
2023.05.13 10:00	2 óra	Bodnár	Place, SaboteurPlace osztályok módosítások, pálya deklarációk javítása
2023.05.13 12:00	1 óra	Gutási	DestructionTimer, Cistern, Element, Game, WaterSource osztályok módosítása
2023.05.14. 11:00	2.5 óra	Andai	ActionCounter implementálása a prototype-ban, info függvény javítása. Játékosok mozgásának javítása. Cső felemelésének, letevésének hibáinak javítása.
2023.05.14. 13:00	2 óra	Bodnár	Negatív tesztknél jelentkező kivételek kezelése, több helyen NullPointerException,IndexOutOfBoundsException
2023.05.14. 17:00	1,5 óra	Sziksai	Save és Repair javítása, kommenek írása, Pump addNeighbor felüldefiniálása, togglerandom megvalósítása, help javítása, tesztesetek ellenőrzése
2023.05.14. 18:45	0.5 óra	Andai	10.2.1-10.2.6 tesztek végrehajtása
2023.05.14. 21:00	0.5 óra	Bodnár	10.2.19-10.2.24 tesztek végrehajtása
2023.05.14 22:30	1 óra	Sziksai	togglerandom javítása és tesztek ellenőrzése, dokumentálása 10.2.25-10.2.30
2023.05.14 23:45	2.5 óra	Vajda	10.2.15-10.2.18 tesztek dokumentálása

2023.05.14 23:00	2.5 óra	Gutási	10.2.8-10.2.14 tesztek dokumentálása, dokumentum formázása, adminisztráció
------------------	---------	--------	--

10. Grafikus felület specifikációja

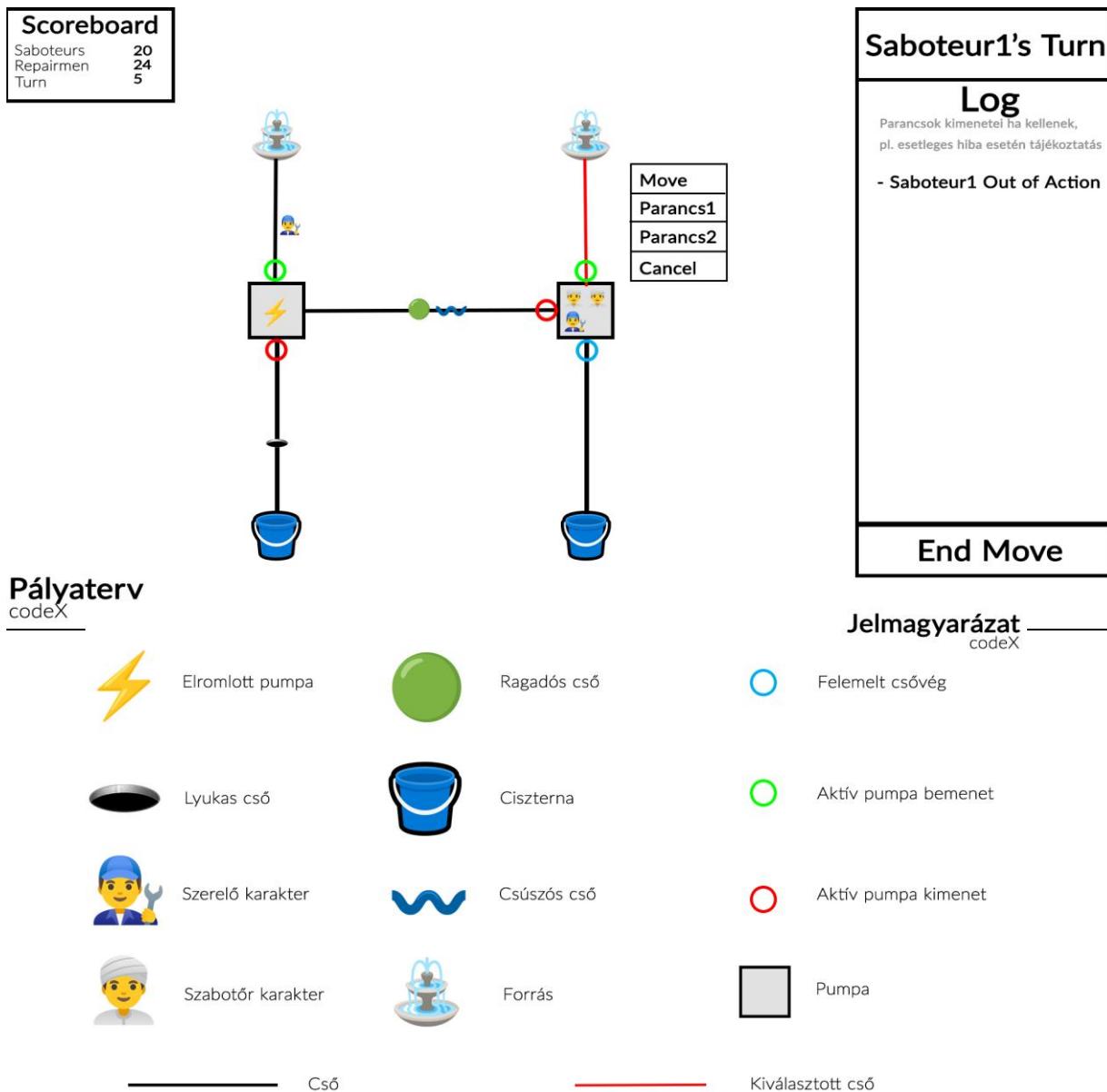
10.1 A grafikus interfész

Menü
codeX

Játékosok száma **2** 

Körök száma **4** 

Start

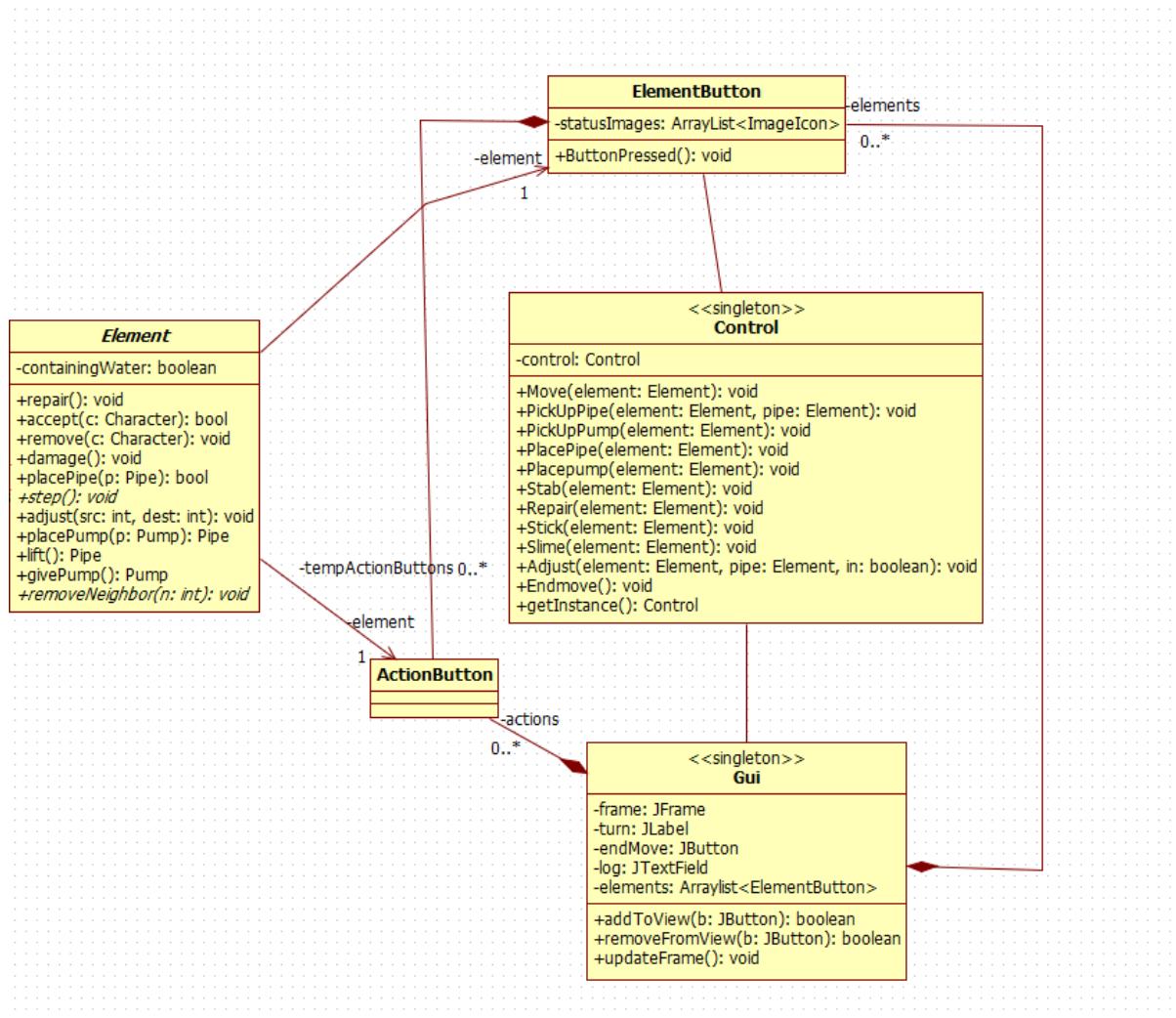


10.2 A grafikus rendszer architektúrája

10.2.1 A felület működési elve

Grafikus megvalósításunk során az MVC modellt követjük, ahol a View a GUI -, a Controller a Control osztály. A Model pedig a már a prototípusban elkészített osztályok halmaza. Továbbá pull alapelveket alkalmazunk, azaz a felület kérdezi le a modellt a változásokról (például játékosok lépéseinek és játékmenetek köreinek végén).

10.2.2 A felület osztály-struktúrája



10.3 A grafikus objektumok felsorolása

10.3.1 Control

- **Felelősség**

Singleton osztály, melynek a feladata a GUI és a Game összekötése, a megjelenítés és a játéklogika szétválasztása.

- **Attribútumok**

- **-control:Control:** Az egyetlen Control objektumot tartalmazó statikus változó

- **Metódusok**

- **+Control getInstance():** Visszaadja az egyetlen Control objektumot

- **+ void Move(Element element):** Átmozgatjuk az aktuális játékos karakterét a paraméterként kapott element-re.

- **+ void PickUpPipe(Element element, Element pipe):** Felveszünk egy csövet a paraméterül kapott element-ról, ahol pipe a felvett cső.

- **+ void PickUpPump(Element element):** Felveszünk egy Pumpát a paraméterként kapott element-ról

- **+ void PlacePipe(Element element):** Lehelyezünk egy csövet a paraméterül kapott element-re

- **+ void PlacePump(Element element):** Lehelyezünk egy Pumpát a paraméterül kapott element-re

- **+ void Stab(Element element):** Kilyukasztjuk a paraméterül kapott element-et

- **+ void Repair(Element element):** Megjavítjuk a paraméterül kapott element-et

- **+ void Stick(Element element):** Ragadóssá tesszük a paraméterül kapott element-et

- **+ void Slime(Element element):** Csúszóssá tesszük a paraméterül kapott element-et

- **+ void Adjust(Element element, Element pipe, boolean in):** Beállítja a paraméterül kapott element be és kimeneti csövét. Pipe a cső amit be szeretnénk állítani és in azt mondja meg hogy be vagy kimenet legyen.

- **+ void EndMove():** Befejezi az aktuális játékos körét.

- **+ void LoadPlayerCount(int playerCount):** Átadja a létrehozandó játékosok számát a Game-nek

- **+void LoadTurnCount(int turnCount):** Átadja a körök számát a Game-nek

10.3.2 GUI

- **Felelősség**

Összefogja az összes kijelzőn megjelenítendő komponenst. Ő tárolja az elemeket, illetve a játék menetéről tájékoztató komponenseket.

- **Attribútumokn**
 - **-JFrame frame:** Az összes további komponens megjelenítéséhez használt frame.
 - **-JLabel turn:** Aktuális játékos nevének kiíratására szolgál.
 - **-JButton endMove:** Ezzel a gombbal lehet befejezni a kört.
 - **-JTextField log:** Szöveges kimenet megjelenítésére szolgál. Például hiba esetén itt kap a felhasználó bővebb magyarázatot.
 - **-ArrayList<ElementButton> elements:** A megjelenített elemek gombjai.
- **Metódusok**
 - **+bool addToView(JButton b):** Ezen keresztül lehet gombot hozzáadni a Frame-hez.
 - **+bool removeFromView(JButton b):** Eltávolítja a paraméterben megadott gombot a Frame-ból.
 - **+void updateFrame():** Újrarendezze a Frame-et.

10.3.3 ElementButton

- **Felelősség**

Feladata a pályát felépítő elemek reprezentálása a grafikus felületen, azok kattintásának lekezelése. További feladata az elvégezhető interakciókat reprezentáló ActionButton-ök megjelenítése.

- **Ósosztályok**

JButton

- **Attribútumok**
 - **-Element element:** A gomb által reprezentált Element pályaelem.
 - **-ArrayList<ImageIcon> statusImages:** az objektum státuszának kirajzolásához használt többi képelemek (pl lyukas, csúszós, ragadós csövekhez a megfelelő ikonok)
 - **-ArrayList<ActionButton> tempActionButtons:** azon gombok, amelyeket a kattintás hatására meg kell jelenítenie az ElementButton-nek. Amint egy ActionButton interakciója lefutott, törlődnek ezek a gombok
 - **-Control controller:** A Controller objektum referenciája.
- **Metódusok**
 - **+void ButtonPressed():** Meghívásának hatására megjeleníti az elvégezhető interakciókat ActionButton-ök formájában.

10.3.4 ActionButton

- **Felelősség**

ElementButtonokhoz tartozó választható interakciók reprezentációja.

Feladata ezen kattintható gombokhoz tartozó megfelelő függvény hozzárendelése.

- **Ősosztályok**

JButton

- **Attribútumok**

- **-ArrayList<Element> elements:** A gomb ezen Element pályaelemeken végzi a műveletet.

- **-Control controller:** A Controller objektum referenciaja.

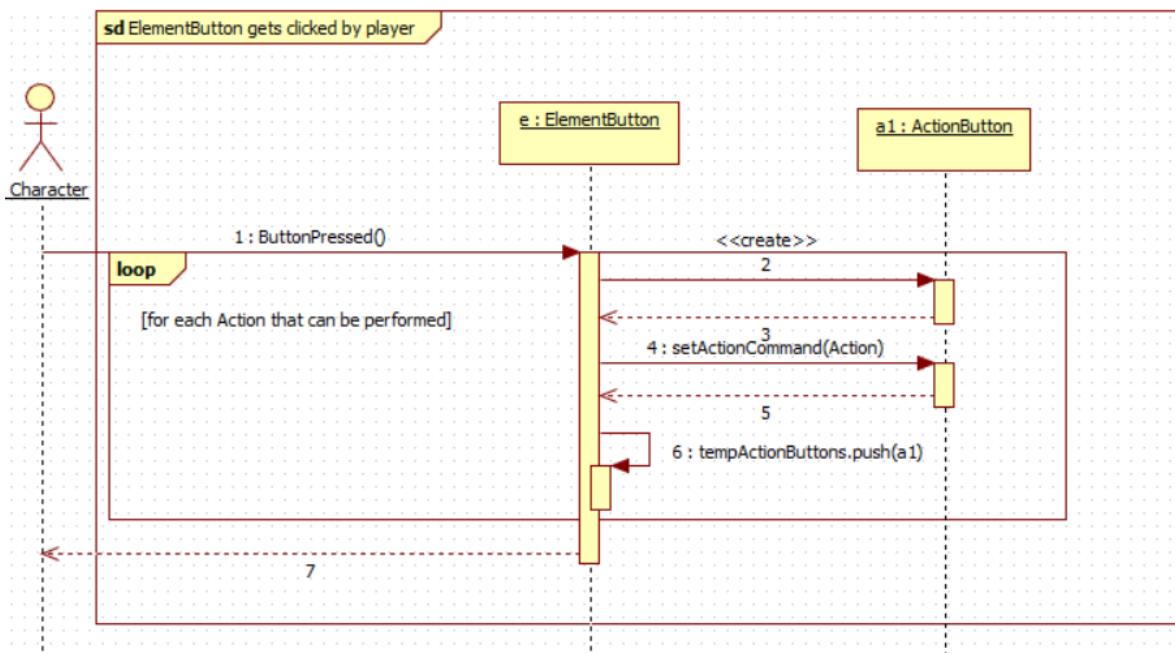
- **Metódusok**

- **+void ButtonPressed():** Meghívja a Control osztályban a hozzá tartozó megfelelő függvényt.

10.4 Kapcsolat az alkalmazói rendszerrel

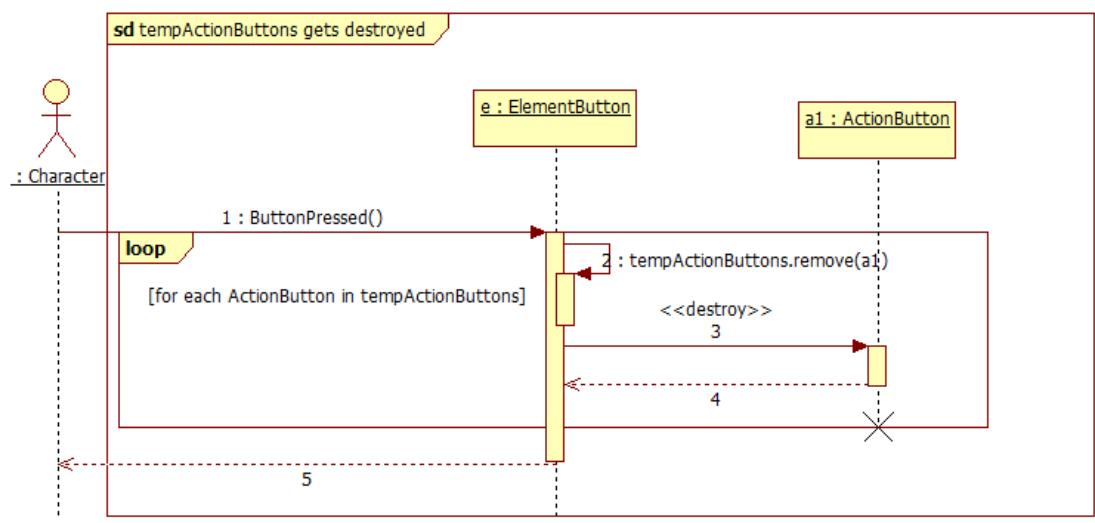
10.4.1 ElementButton gets clicked by a player

Egy Játékos rákattint egy ElementButton-re, melynek hatására létrejönnek (megjelennek) az ActionButtonök



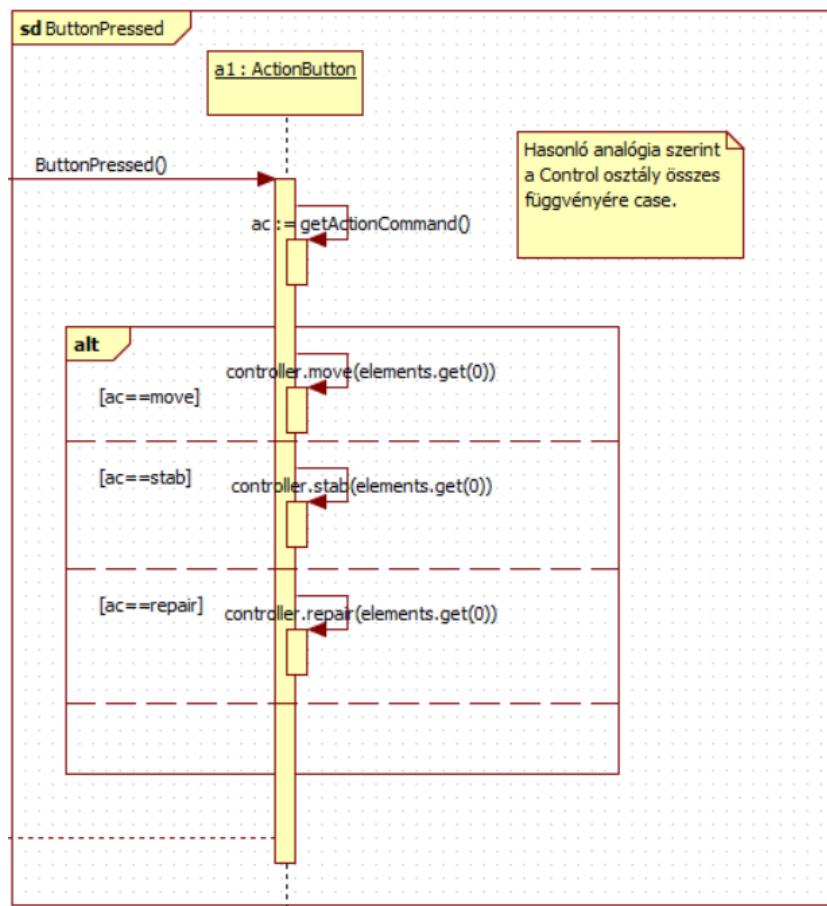
10.4.2 ElementButton deletes temporary ActionButtons

Befejeződik a kattintott ActionButton által elindított interakció, így megszűnnek a gombok.

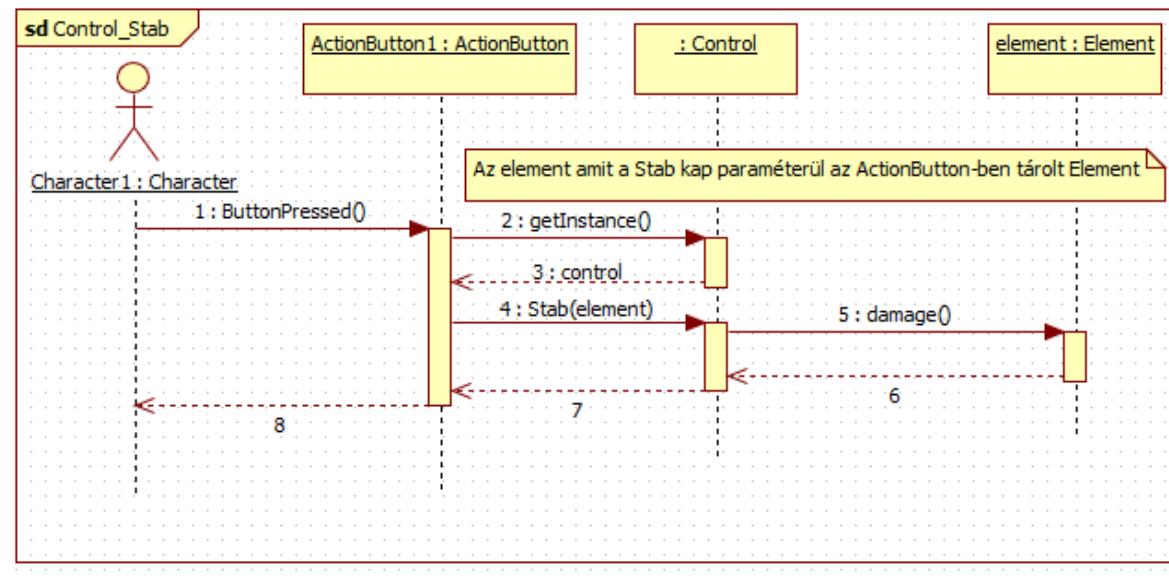


10.4.3 ActionButton ButtonPressed

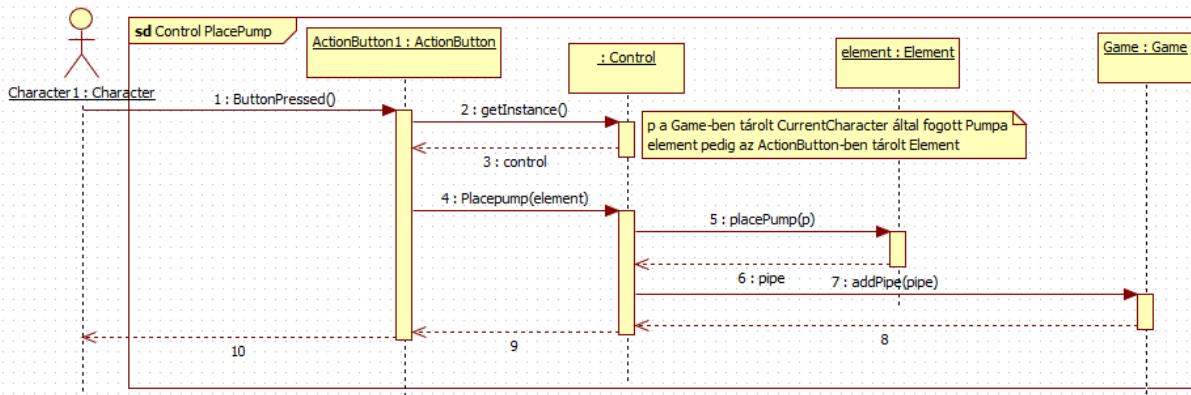
Egy valamelyen ActionButton kattintásakor.



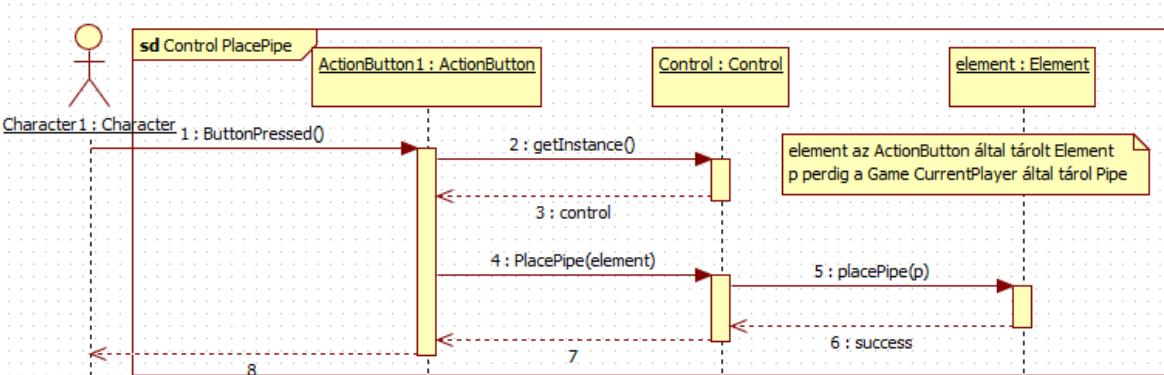
10.4.4 Control Stab



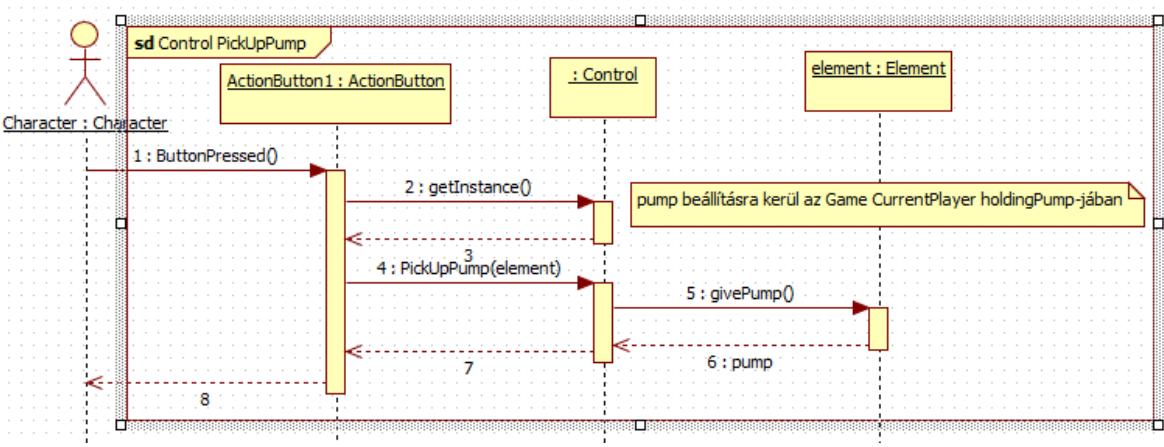
10.4.5 Control PlacePump



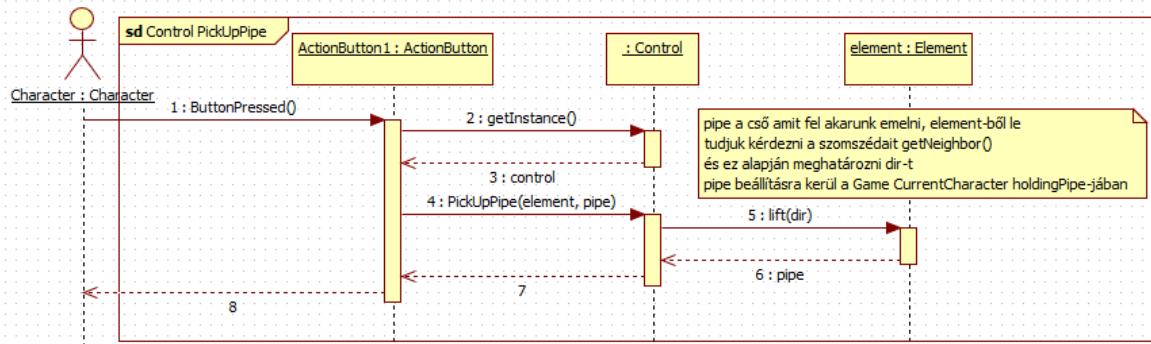
10.4.6 Control PlacePipe



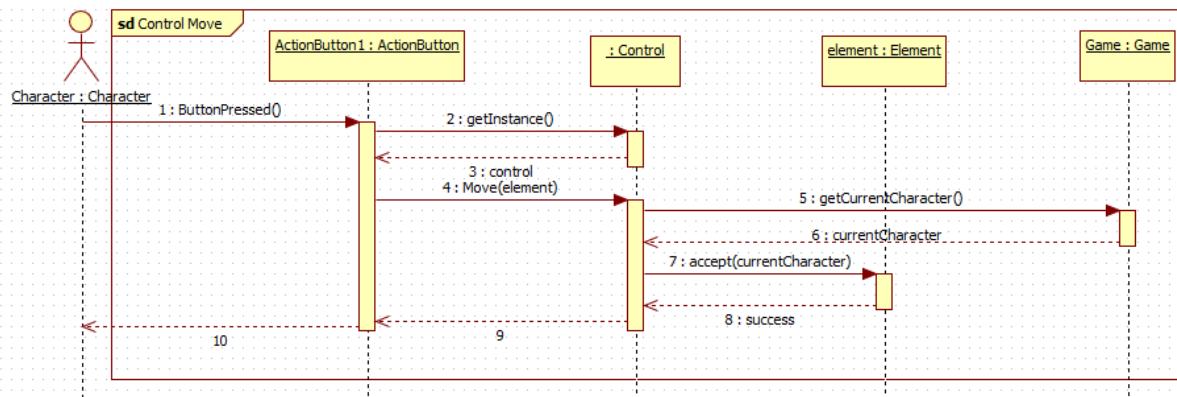
10.4.7 Control PickUpPump



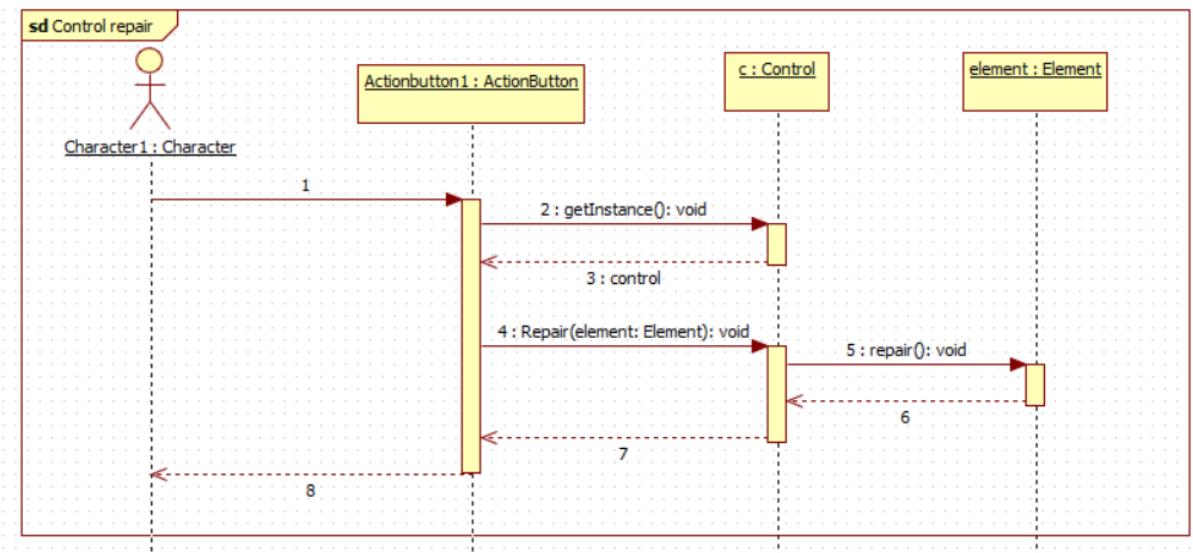
10.4.8 Control PickUpPipe



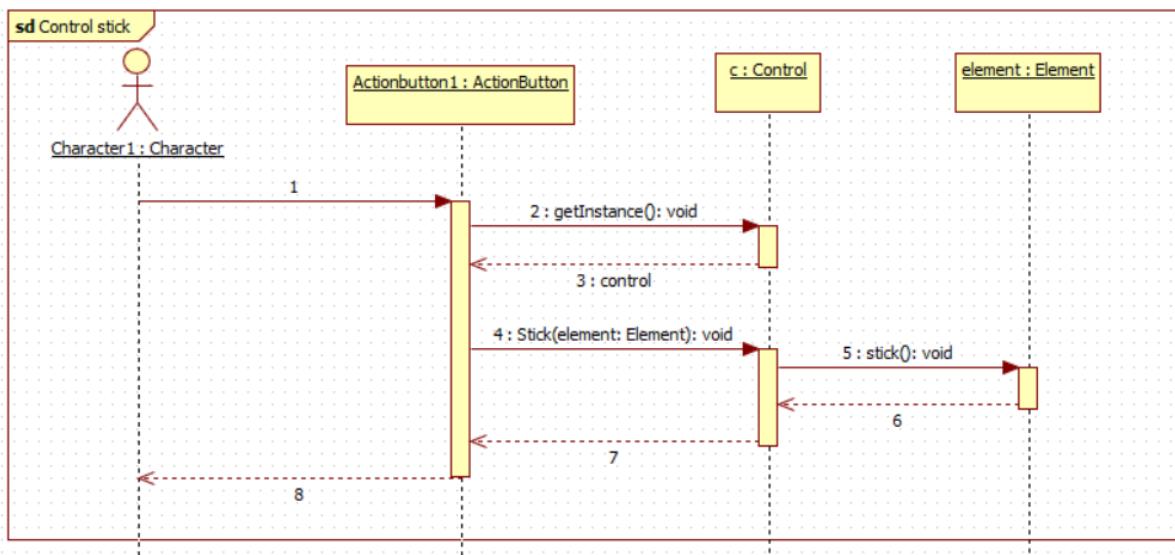
10.4.9 Control Move



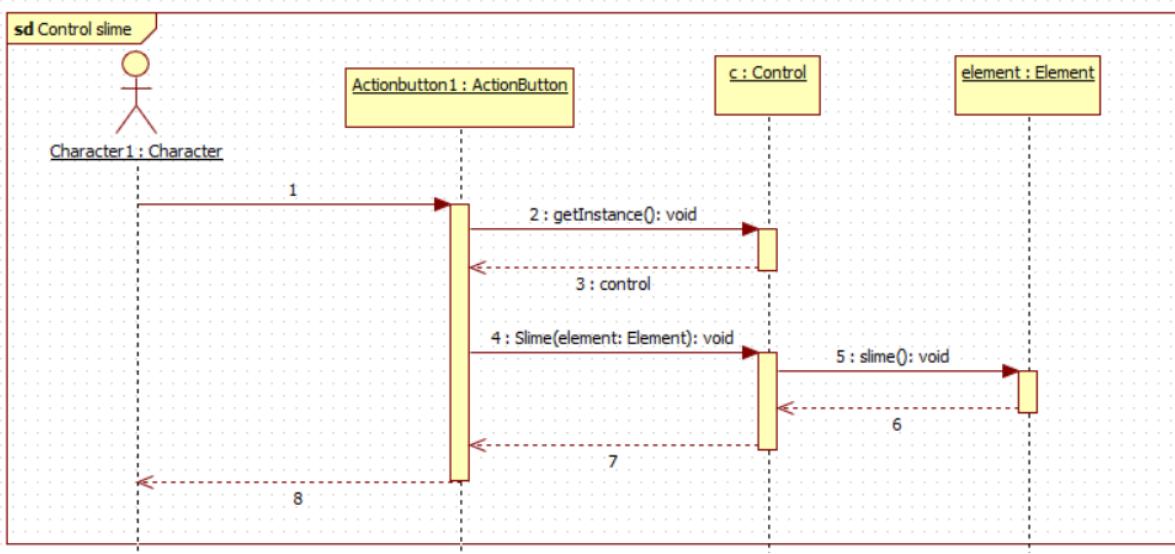
10.4.10 Control Repair



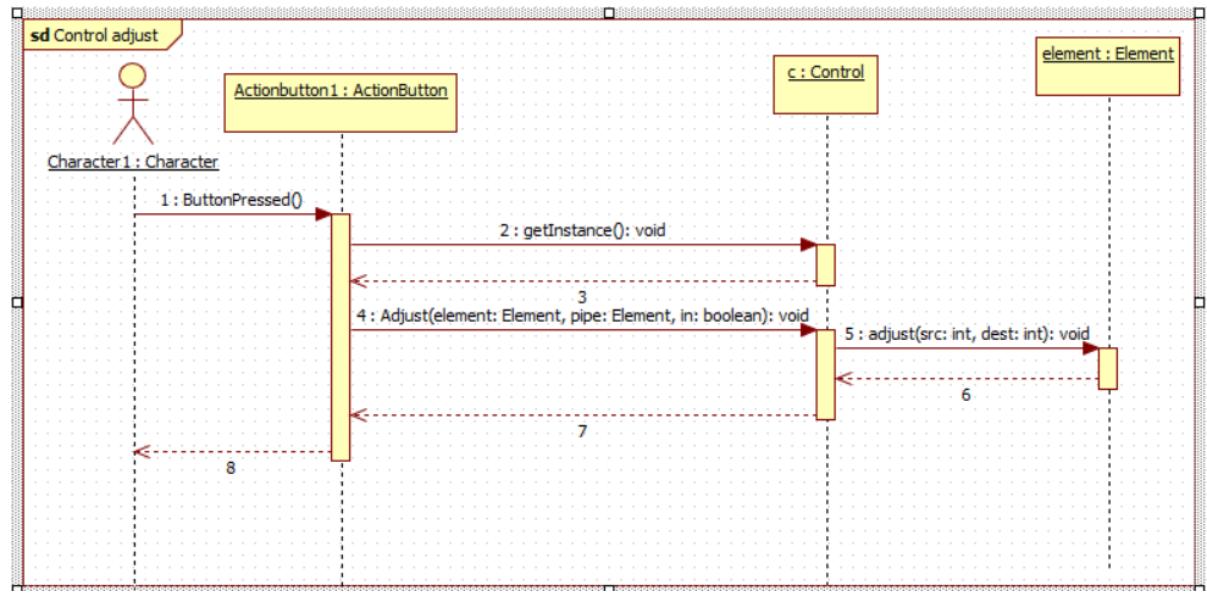
10.4.11 Control Stick



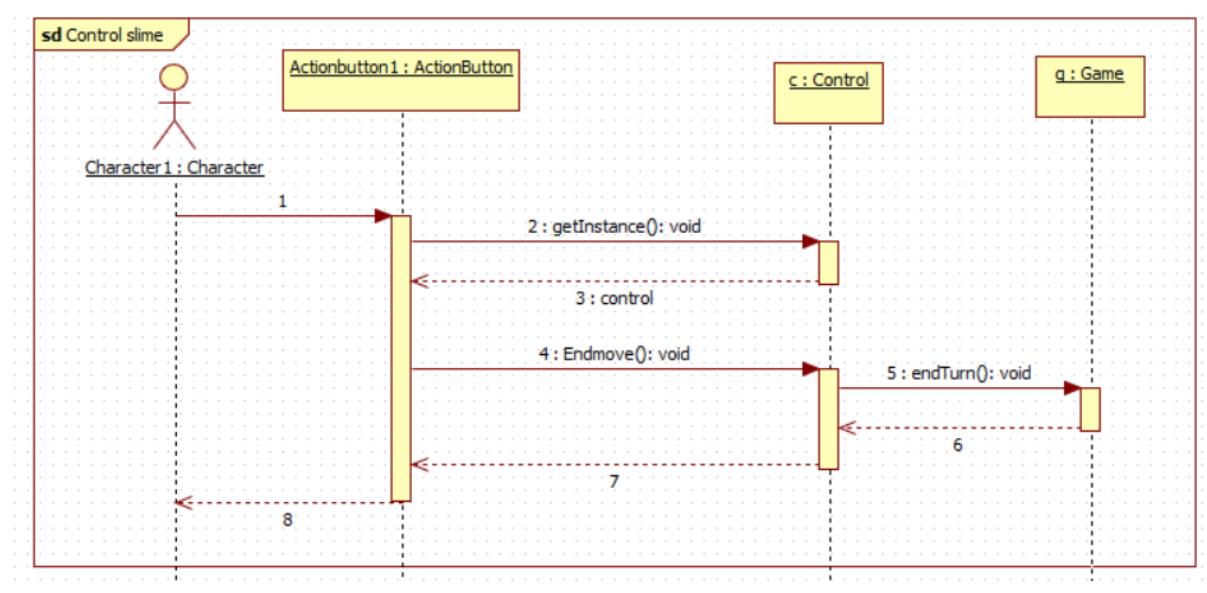
10.4.12 Control Slime

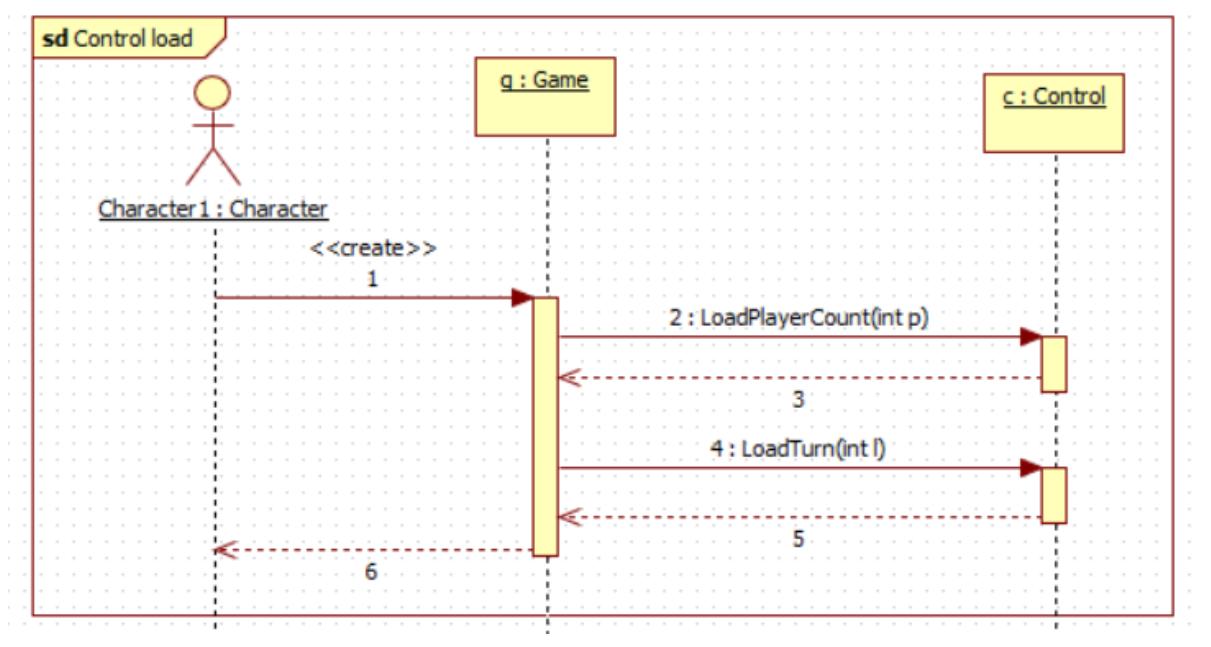


10.4.13 Control adjust



10.4.14 Control Endmove



10.4.15 Control Load

10.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2023.05.20. 18:45	0.5 óra	Bodnár, Gutási, Sziksai, Vajda	Értekezlet. Döntés: Sziksai kidolgozza a Control osztályt. Andai kidolgozza a GUI osztályt. Bodnár kidolgozza az ActionEvent osztályt. Gutási elkészíti az ElementButton osztályt, majd Vajda elkészíti az ezekhez tartozó osztálydiagrammot.
2023.05.20 20:00	2 óra	Sziksai	Osztályok és metódusaik, valamint a kezelőfelület tervének vázlatos megvalósítása, a Control osztály kidolgozása
2023.05.21. 9:30	0.5 óra	Andai	A GUI osztály dokumentálása.
2023.05.21 10:00	0.5 óra	Gutási	Az ElementButton osztály dokumentálása.
2023.05.21. 11:30	0.5 óra	Bodnár	Az ActionEvent osztály dokumentálása.
2023.05.21. 13:00	0.5 óra	Bodnár, Gutási, Sziksai	Értekezlet. Általános ötletelés az ActionButton-ök megjelenítésével kapcsolatban Döntés: mindenki azokat a szekvenciadiagramokat készíti el, ami az általa elkészített osztályhoz tartozik. Sziksai és Vajda felosztják egymást közt a Control osztály metódusaihoz tartozó szekvenciadiagramot elkészítését.
2023.05.21 10:00	1 óra	Gutási	A grafikus interfész bemutató képek megszerkesztése
2023.05.21 15:0	3 óra	Vajda	Új osztályok és metódusaik javítása, osztálydiagram elkészítése
2023.05.21 20:00	1 óra	Gutási	11.4.1 és 11.4.2 szekvenciadiagramok megszerkesztése.
2023.05.21 23:00	1 óra	Bodnár	11.4.3 szekvenciadiagram elkészítése, javítások.
2023.05.22 0:00	1,5 óra	Sziksai	11.4.4-9 szekvenciadiagramok elkészítése
2023.05.22 6:30	15 perc	Gutási	11.2.1 megírása, dokumentum rendezése
2023.05.22 9:00	1 óra	Vajda	11.4.10-11.4.15 szekvenciadiagramok elkészítése

11. Grafikus változat beadása

11.1 Fordítási és futtatási útmutató

11.1.1 Fájllista

Fájl neve	Méret (Byte)	Keletkezés ideje	Tartalom
ActionButton.java	3029	2023.05.30 18:32	ActionButton osztály
App.java	178	2023.05.30 18:33	App osztály: az alkalmazás indítása, leállítása
Character.java	1094	2023.05.30 18:33	Character osztály
Cistern.java	3289	2023.05.30 18:33	Cistern osztály
Control.java	4785	2023.05.30 18:35	Control osztály
DestructionTimer.java	1718	2023.05.30 18:35	DestructionTimer osztály
Element.java	7743	2023.05.30 18:36	Element osztály
ElementButton.java	23271	2023.05.30 18:42	ElementButton osztály
Game.java	9682	2023.05.30 18:42	Game osztály
Gui.java	22095	2023.05.30 20:27	Gui osztály
NonPipe	2074	2023.05.30 18:45	NonPipe osztály
Pair.java	395	2023.05.30 18:45	Pair osztály
Pipe.java	11012	2023.05.30 18:46	Pipe osztály
Place.java	653	2023.05.30 18:46	Place osztály
Prototype.java	20001	2023.05.30 18:46	Prototype osztály
Pump.java	5941	2023.05.30 20:27	Pump osztály
Repairman.java	5532	2023.05.30 18:48	Repairman osztály
RepairmanPlace.java	671	2023.05.30 18:49	RepairmanPlace osztály
Saboteur.java	3047	2023.05.30 18:54	Saboteur osztály
SaboteurPlace.java	296	2023.05.30 18:49	SaboteurPlace osztály
SaboteurPointSource.java	277	2023.05.30 18:52	SaboteurPointSource osztály
Skeleton.java	19091	2023.05.30 18:54	Skeleton osztály
Tabulator.java	432	2023.05.30 18:52	Tabulator osztály
WaterSource.java	1707	2023.05.30 18:53	WaterSource osztály
img ¹ /blue-pipe.png	23030	2023.05.29 13:02	Kék csövet ábrázoló kép
img/bucket.png	11795	2023.05.21 09:43	Vödröt ábrázoló kép
img/droplets.png	5400	2023.05.21 09:44	Cseppek ábrázoló kép
img/fountain.png	13641	2023.05.21 09:44	Szökőkutat ábrázoló kép
img/green-pipe.png	22623	2023.05.29 13:02	Zöld csövet ábrázoló kép
img/high-voltage.png	4842	2023.05.21 13:32	Villámot (elromlott pumpát) ábrázoló kép
img/hole.png	7941	2023.05.21 09:45	Lyukat ábrázoló kép
img/man-mechanic.png	14058	2023.05.21 09:47	Szerelő karaktert ábrázoló kép

¹ A képek forrása: <https://emojipedia.org>. Letöltéseik idejét lásd a keletkezés ideje oszlopban.

img/man-wearing-turban.png	9060	2023.05.21 09:50	Szabotőr karaktert ábrázoló kép
img/man-with-blue-pipe.png	125772	2023.05.29 13:02	Szerelő karaktert kék csővel ábrázoló kép
img/man-with-green-pipe.png	65724	2023.05.29 13:02	Szerelő karaktert zöld csővel ábrázoló kép
img/man-with-pump.png	179172	2023.05.29 13:02	Szerelő karaktert pumpával ábrázoló kép
img/man-with-red-pipe.png	64686	2023.05.29 13:02	Szerelő karaktert piros csővel ábrázoló kép
img/man-with-yellow-pipe.png	65447	2023.05.29 13:02	Szerelő karaktert sárga csővel ábrázoló kép
img/pump.png	4228	2023.05.29 13:02	Pumpát ábrázoló kép
img/red-pipe.png	19859	2023.05.29 13:02	Piros csövet ábrázoló kép
img/sticky.png	5269	2023.05.29 13:02	Ragadós csövet jelöli ikont ábrázoló kép
img/wavy-dash.png	5022	2023.05.21 09:55	Csúszós csövet jelző ikont ábrázoló kép
img/yellow-pipe.png	19290	2023.05.29 13:02	Sárga csövet ábrázoló kép

11.1.2 Fordítás és telepítés

A forrásfájlok fordításához a forrásfájlokat tartalmazó mappában, a **javac App.java -Xlint:unchecked** parancsot kell kiadni. Ehhez szükséges, hogy a fordító gépen telepítve legyen Java 8-hoz tartozó fejlesztői csomag (JDK).

11.1.3 Futtatás

A program futtatásához a lefordított .class fájlokat tartalmazó mappában, használjuk a **java App** parancsot. Ennek hatására a program elindul. A felhasznált képfájlok helyezkedjenek el a fordított .class fájlokat tartalmazó mappán belül az **img** mappában.

11.2 Értékelés

A projekt tagjai aláírásukkal igazolják, hogy az alább szereplő százalékarányok valósak és elfogadják azokat.

Tag neve	Tag neptun	Munka százalékban	Aláírás
Andai Roland	H3G1PQ	21,44%	
Bodnár Márk	AAL25F	18,78%	
Gutási Ádám	XW53QZ	19,13%	
Szikszai Levente	AENWDT	21,58%	
Vajda András	M7LWCB	19,06%	

11.3 Napló

Kezdet	Időtartam	Résznevők	Leírás
2023.05.25 12:00	2.5 óra	Andai, Bodnár, Gutási, Sziksai, Vajda	Értekezlet. UI tervezése, ötletelés
2023.05.26 8:00	6 óra	Vajda	Control, Gui, ElementButton, ActionButton osztályok vázának elkészítése. Control osztály függvényeinek megvalósítása. Game osztály bővítése.
2023.05.26. 23:00	1 óra	Sziksai	Control, Game, Element kommentezése, canPerformAction megvalósítása az Element-ben ActionButton és ElementButton: ActionPerformed vázának megvalósítása
2023.05.27. 8:30	1.5 óra	Andai	NonPipe típusú ElementButtonök megjelenítésének implementálása
2023.05.27. 10:00	3 óra	Gutási	Gui osztály implementálása, összekötése a Control-lal.
2023.05.27. 11:00	4 óra	Bodnár	ActionButton implementálása, ElementButton bővítése.
2023.05.27. 14:00	2 óra	Andai	Csövek megjelenítésének implementálása
2023.05.27. 21:30	1 óra	Andai	Pumpák, csövek állapotainak megjelenítése
2023.05.28. 11:30	1 óra	Andai	Pumpák ki- és bemenetének jelzése
2023.05.28 15:00	1 óra	Sziksai	Többszálú működés megvalósítása, Character, Repairman, Saboteur, Game, Gui, Control módosítása
2023.05.28 14:30	2 óra	Andai	Felemelt cső megjelenítése.
2023.05.28 21:00	0.5 óra	Andai	Hurokcső megjelenítése.
2023.05.29 14:00	4 óra	Bodnár	showActionButtonWindow függvény, választható interakciók megjelenítésének tervezése, implementálása, tesztelése
2023.05.29 15:00	1 óra	Andai	EndGamePanel elkészítése, updateFrame javítása
2023.05.29 17:00	3 óra	Vajda	Karakterek megjelenítése az elemeken, új pumpa bekötésének és a View frissítésének problémáinak feloldása
2023.05.30 10:00	2.5 óra	Andai	ActionButton elrendezésének javítása, ScoreBoard fix,

			ActionButton gombjainak láthatóságának kezelése,
2023.05.30 15:00	1 óra	Andai	Csövek javításának kezelése, csövek/pumpák felemelésének/letevésének feltételeinek kezelés. 13.1.3 Kitöltése
2023.05.30 15:30	3 óra	Szikszai	Vízfolyás javítása, PlacePump, Move, PickUpPipe, PlacePipe action-ök megvalósítása (ElementButton)
2023.05.30 16:00	4 óra	Gutási	Gui, ElementButton, ActionButton osztály módosítása. 13.1.2 Kitöltése.
2023.05.30. 22:00	2.5 óra	Bodnár	Előző laboralkalmon megbeszélt követelmények tesztelése, apróbb módosítások a modellben. 13.1.1 Kitöltése

12. Összefoglalás

12.1 A projektre fordított összes munkaidő

Tag neve	Munkaidő (óra)
Bodnár Márk	67
Gutási Ádám	68.25
Szikszai Levente	77
Vajda András	68
Andai Roland	76.5
Összesen	356.75

- A feltöltött programok forrássorainak száma**

Fázis	KódSOROK száma
Szkeleton	2086
Prototípus	2787
Grafikus változat	4458
Összesen	9331

12.2 • Projekt összegzés

12.2.1 Mit tanultak a projektből konkrétan és általában?

Hogyan kell egy kicsit komplexebb feladatnak nekiállni, megtervezni és elkészíteni. Illetve hogyan dolgozzunk csapatban, hogyan tudjuk szétosztani részekre a feladatokat.

12.2.2 Mi volt a legnehezebb és a legkönnyebb?

A legkönnyebb az volt, amikor 2 hetes leadási határidőt kaptunk egy feladatrészre.. A legidegőrlőbbek a pushok miatt kialakult merge conflictok megoldásai voltak.

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

A TAD szerint kb. heti 4.5 órával per fő el lehet készíteni a feladatokat. Ennél kicsit több időt kellett rászánni, hogy megmaradjon a beadandók magas színvonala.

12.2.4 Ha nem, akkor hol okozott ez nehézséget?

Az Analízis modell I.-gyel volt a vártnál sokkal több feladat, megbeszélni és javítanivaló.

12.2.5 Milyen változtatási javaslatuk van?

Semmilyen, a tárgy ilyen formában tökéletes.

12.2.6 Milyen feladatot ajánlanának a projektre?

Ki lehetne indulni valamilyen társasjátékból (pl. Monopoly, majd akár további szabályokat kitalálni hozzá).

12.2.7 Egyéb kritika és javaslat

Egyes heteken a beadási határidő rövidnek bizonyult, felgyülemlettek a beadandók a többi tárggyal együtt. Érdemes lehet néhány héten akár csak egy nappal is eltolni a beadási határidőt.