

2. Követelmény, projekt, funkcionalitás

2.1 Bevezetés

2.1.1 Cél

A projekt célja olyan játékprogram készítése, ami egy sivatagi csőrendszert szimulál és két csapat játszhatja egymás ellen. Az elkészült szoftvernek meg kell felelnie az alábbi specifikációnak.

2.1.2 Szakterület

A kialakítandó szoftver egy játékprogram, amely szórakozásra, valamint stratégiai gondolkodás fejlesztésére használható.

2.1.3 Definíciók, rövidítések

JRE: Java Runtime Environment

JDK: Java Development Kit

MVC (-modell): Model-View-Controller

2.1.4 Hivatkozások

<https://www.iit.bme.hu/file/11582/feladat>

<https://devil.iit.bme.hu:9181/hercules/>

https://vik.wiki/images/e/e4/Prog3_7_eloadas_2020.pdf

<https://app.diagrams.net/>

2.1.5 Összefoglalás

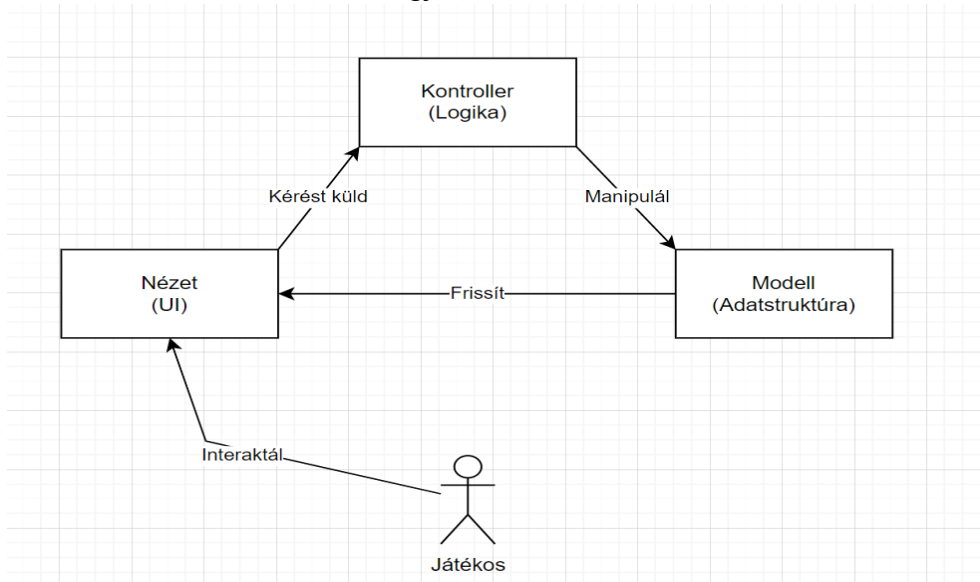
A dokumentum a továbbiakban részletes követelményeket fogalmaz meg a szoftverrel, annak funkcióival, tulajdonságaival és az azt futtató hardverrel szemben valamint leírja a projekt elkészülési folyamatának tervét.

2.2 Áttekintés

2.2.1 Általános áttekintés

A program architektúrája alapvetően az MVC mintát használja.

A modell a program legbelső komponense, tárolja és kezeli az adatokat. A kontroller eseményvezérelt módon kommunikál a felhasználóval és irányítja a modellt. A nézet közvetlenül a felhasználói felület megjelenítéséért felel.



2.2.2 Funkciók

A drukmákorai sivatagban mindig is komoly probléma volt a vízellátás megoldása. Ennek érdekében egy bonyolult, összefüggő csőrendszert építettek ki. A csőrendszer a ciszternák és a sivatagon túli városok között végzi a vízellátást, hogy minden lakosnak elegendő víz jusson. A csőrendszer két fő alkotóeleme a csövek, illetve az aktív elemek.

A források a pálya bal oldalán, a ciszternák a jobb oldalán helyezkednek el. A pumpák bárhol lehetnek a kettő között. A csövek egyenes szakaszok az aktív elemek közt. A csövek nem keresztezik egymást, és a játék elindításakor minden esetben egy alapértelmezett kezdőpálya fogadja a játékosokat.

A csövek elágazás nélküliek, az aktív elemek pedig a csövekhez kapcsolódnak, ide sorolható többek között a forrás, a ciszterna, vagy a vízátemelő pumpa. A csőrendszer rugalmasan bővíthető, változtatható, módosítható. Ezáltal a víz városokba való eljuttatása optimálisabbá válhat. A csövek maguk is rugalmasak, egyik végük lecsatlakoztatható, és másik elemhez kapcsolható (de közben a másik vég rögzített marad). Időnként a ciszternánál új cső keletkezhet, ha még van hely neki. Ezeknek a csöveknek az egyik vége a ciszternához kapcsolódik, másik végük azonban szabad. A szabad végű csövekbe betáplált víz a homokba folyik (kárba vész).

Egy adott pumpa több (de az adott pumpára jellemző véges számú) csövet is összeköthet. Minden pumpán szabadon állítható, hogy melyik csőből melyikbe

pumpáljon. Egyszerre csak egy bemenete és kimenete állítható be és eközben a többi rákötött cső zárt állapotban van. A pumpák véletlenszerű módon elromlanak, ilyenkor a vízáramlás megszűnik, nem jut át rajta víz. Minden pumpa rendelkezik egy víztartállyal. A víztartály egy átmeneti tárolóként funkcionál a bemeneti és kimeneti csövek között. A csövek is kapacitással rendelkeznek, egy csőbe egy pumpa csak akkor tud vizet pumpálni, ha a csőkapacitás ezt lehetővé teszi.

Fontos, hogy minél nagyobb kapacitással, és minél gyorsabban érjen el a városba a víz. Azonban egyes szereplők ezt meg szeretnék akadályozni. A csőrendszerben kétféle játékos mozog, szerelők és szabotőrök. Előbbiek tartják karban a csőrendszert. Megjavítják az elromlott pumpákat; ha egy cső kilyukad, megfoltozzák; átállítják a pumpákat, hogy minél több víz áramoljon át. A céljuk, hogy a ciszternákba minél több víz jusson el. Ezért pontokat kapnak.

A szerelők fontos feladata a (ciszternáknál lévő) szabad csövekkel a kapacitásnövelés. Ebben segít még az, hogy a ciszternáknál lehetőség van pumpák felvételére is, ami egy tetszőleges cső közepére helyezhető, ha nem sérti a pumpák minimális távolságát. Ehhez a csövet ketté kell vágni, és a pumpát a két véghez hozzá kell csatlakoztatni.

A szabotőrök éppen ellentétes feladatot látnak el, gátolják, akadályozzák a vízmozgást. A szabotőrök a pumpákat állítják, és a csöveket lyukasztják ki. A kilyukadt csövekből a víz kifolyik, és így a pumpához nem jut víz. Az ő céljuk, hogy minél több víz folyjon ki a lyukakon, amiért pontokat kapnak.

A szerelők és a szabotőrök is egyaránt csakis a hálózaton mozoghatnak, sehol másutt. Egyszerre egy csövön egy játékos állhat, függetlenül a csapatától. Egy pumpán viszont többen is állhatnak, ezért itt a játékosok kikerülhetnek egymást.

Mindkét csapatban legalább 2 fő van. A játékot egy játékos indíthatja el, és akkor ér véget, amikor az idő lejár. A mozgás kattintással, minden más a billentyűzet megfelelő gombjainak segítségével irányítható.

Minden körben minden játékos lép egyesével. Egy adott körben egy adott játékos kap valamennyi időt a lépésre, de hamarabb is befejezheti azt. Ezen kívül a csapatok össz-idővel is rendelkeznek. Az össz-idő a csapat teljes játék alatti lépéseire vonatkozik. Amikor egy csapat össz-ideje lejár, a játék véget ér és a másik csapatnak (a megmaradt idővel arányosan) extra pont jár.

2.2.3 Felhasználók

A játékot bárki játszhatja, akinek a futtatási környezete a 2.3.2 pontban megfogalmazott követelményeket teljesíti és a követelmények között szereplő perifériákat használni tudja. Több játékos játszhat egyszerre egy gépen.

2.2.4 Korlátozások

Elvárás, hogy az elkészült szoftver jól fusson a meghatározott minimális hardveren. Modulárisnak és könnyen bővíthetőnek kell lennie. A cél-felhasználók számára könnyen kezelhető felülettel kell rendelkeznie.

2.2.5 Feltételezések, kapcsolatok

A linkeken található anyagokat a projekt szakszerűbb megírásához és a dokumentáció különböző elemeinek elkészítéséhez használtuk (pl.: draw.io a use-case diagram elkészítéséhez).

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azo nosít ó	Leírás	Ellenőrzé s	Prioritá s	Forrás	Use-case	Ko m me nt
SV1	A játékot el lehet indítani, ilyenkor a víz elkezd folyni a nyitott csöveken és pumpákon.	Bemutatas	Alapvet ő	Csapat	Játék indítása	
SV2	A játékot le lehet állítani, ilyenkor a csöveken és a pumpákon nem folyik több víz.	Bemutatas	Alapvet ő	Csapat	Játék leállítása	
SV3	A csőrendszer a források és a városok ciszternái között fut.	Bemutatas	Alapvet ő	Feladatkiír ás	Játék léptetése, Mozgatás	
SV4	A csőrendszer részei a csövek és az aktív elemek (pl. ciszterna, forrás, pumpa, stb.).	Bemutatas	Alapvet ő	Feladatkiír ás	Játék léptetése, Mozgatás	
SV5	A csövek le- és felcsatlakoztathatóak, és nem keresztezhetik egymást.	Bemutatas	Alapvet ő	Feladatkiír ás	Cső végének mozgatása	
SV6	Az új csövek folyamatosan készülnek a ciszternáknál egy határig. Az új csövek egyik vége kapcsolódik a ciszternákhoz, a másik szabad.	Bemutatas	Fontos	Feladatkiír ás	Cső végének mozgatása Játék léptetése	
SV7	Ha egy cső ép, azaz nem lyukas, és mindkét vége csatlakoztatott, akkor a víz áthalad rajta.	Bemutatas	Alapvet ő	Feladatkiír ás	Játék léptetése, Cső megjavítása, Cső kilyukasztása, Cső végének mozgatása	
SV8	Ha egy cső lyukas, akkor a víz kifolyik belőle.	Bemutatas	Alapvet ő	Feladatkiír ás	Játék léptetése, Cső megjavítása, Cső kilyukasztása	
SV9	A szabad végű csőből a víz a homokba folyik.	Bemutatas	Fontos	Feladatkiír ás	Játék léptetése, Cső végének mozgatása	
SV10	Minden pumpa egy rá jellemző számú csövet köthet össze legfeljebb.	Bemutatas	Alapvet ő	Feladatkiír ás	Pumpa lerakása, Pumpa állítása	
SV11	Minden pumpán külön állítható egy bemenet és egy kimenet.	Bemutatas	Fontos	Feladatkiír ás	Pumpa állítása	
SV12	Az a cső, amelyik nem bemenete az adott pumpának, az el van zárva.	Bemutatas	Fontos	Feladatkiír ás	Pumpa állítása	

SV1 3	A pumpák véletlenszerű módon el tudnak romlani.	Bemutató	Fontos	Feladatkiírás	Pumpát meghibásít	
SV1 4	Minden pumpának van egy víztartálya, ami a víz átemelése során egy átmeneti tárolóként funkcionál.	Bemutató	Fontos	Feladatkiírás	Játék léptetése	
SV1 5	A csövek kapacitással rendelkeznek.	Bemutató	Alapvető	Feladatkiírás	Játék léptetése	
SV1 6	Egy csőbe akkor pumpálható víz, ha a cső kapacitása ezt megengedi.	Bemutató , Kiértékelés	Fontos	Feladatkiírás	Játék léptetése	
SV1 7	A hálózatban szerelők és szabotőrök vannak	Bemutató	Alapvető	Feladatkiírás	Mozgatás	
SV1 8	A szerelők pumpákat állíthatnak.	Bemutató	Alapvető	Feladatkiírás	Pumpa állítása	
SV1 9	A szerelők csöveket és pumpákat javíthatnak.	Bemutató	Alapvető	Feladatkiírás	Cső megjavítása, Pumpa megjavítása	
SV2 0	A szerelők a ciszternáknál pumpát fel is vehetnek.	Bemutató	Fontos	Feladatkiírás	Pumpa felvétele	
SV2 1	A ciszternánál felvett pumpákat egy cső közepén a szerelők elhelyezhetik, ha lehetséges.	Bemutató	Fontos	Feladatkiírás	Pumpa lerakása	
SV2 2	A pumpa elhelyezéséhez az adott csövet középen ketté kell vágni.	Bemutató	Fontos	Feladatkiírás	Pumpa lerakása	
SV2 3	A szabotőrök pumpákat állíthatnak.	Bemutató	Alapvető	Feladatkiírás	Pumpa állítása	
SV2 4	A szabotőrök csöveket lyukaszthatnak ki.	Bemutató	Alapvető	Feladatkiírás	Cső kilyukasztása	
SV2 5	A játékosok csak a csőrendszeren mozoghatnak.	Bemutató	Alapvető	Feladatkiírás	Mozgatás, Játék léptetése	
SV2 6	A szabotőr a szabad végén kifolyt vízért nem kap pontot.	Bemutató , Kiértékelés	Fontos	Csapat	Játék léptetése	
SV2 7	A játékosok a pumpáknál kerülhetnek ki egymást.	Bemutató	Alapvető	Feladatkiírás	Mozgatás, Játék léptetése	
SV2 8	Egyszerre egy csövön egy játékos állhat, ott nem kerülhetnek ki egymást.	Bemutató	Alapvető	Feladatkiírás	Mozgatás, Játék léptetése	
SV2 9	A játékosok szabotőr-, illetve szerelő csapatokba szerveződnek.	Bemutató	Fontos	Feladatkiírás	Játék indítása, Játék léptetése	
SV3 0	Mindkét csapatban legalább 2 fő van.	Bemutató	Alapvető	Feladatkiírás	Játék indítása, Játék léptetése	
SV3 1	Az a csapat nyer, amelyik több pontot szerez.	Kiértékelés	Fontos	Feladatkiírás	Játék leállítása	
SV3 2	A kör játékosonként megy.	Bemutató	Opcionális	Csapat	Játék léptetése, Mozgatás	
SV3 3	Minden játékos lépésére egy meghatározott idő áll rendelkezésre. Minden csapat egy összesített csapatidővel is rendelkezik.	Bemutató	Opcionális	Csapat	Játék léptetése, Játék leállítása	
SV3 4	Az össz csapatidő lejártakor a játéknak vége	Bemutató	Opcionális	Csapat	Játék leállítása	

SV3 5	A szabotőrök csapata minden lyukas csövön kifolyt vízmennyiségért pontot kapnak.	Bemutatós ,Kiértékelés	Fontos	Feladatkiírás	Játék leállítása	
SV3 6	A szerelők csapata minden a csövekből a ciszternákba befolyt vízmennyiségért pontot kapnak.	Bemutatós ,Kiértékelés	Fontos	Feladatkiírás	Játék leállítása	

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
RR1	Egér	Az egér mozgatására mozog-e a kurzor.	Alapvető	Felhasználó	
RR2	Billentyűzet	Írható mezőbe gépelés hatására megjelennek-e a betűk.	Alapvető	Felhasználó	
RR3	Monitor	Látszik-e a játék.	Alapvető	Felhasználó	
RR4	Java 8+	Parancssorból, a "java -version" parancs beírása.	Alapvető	Felhasználó	

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
DR1	A forráskód lefordítható standard JDK alatt parancssorból.	Bemutatós, Kiértékelés	Alapvető	Csapat	
DR2	A szoftver futtatható állományként indítható.	Bemutatós, Kiértékelés	Alapvető	Csapat	
DR3	Windows 10 vagy újabb, vagy Grafikus megjelenítővel rendelkező Linux operációs rendszer.	Bemutatós	Alapvető	Felhasználó	

2.3.4 Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
NFR1	A szoftver megbízhatóan fut, a játék nem áll le váratlanul.	Bemutatós	Fontos	Csapat	
NFR2	Hordozható, csak a JRE szükséges a futtatáshoz	Bemutatós, Kiértékelés	Alapvető	Csapat	
NFR3	A szoftver futtatása nem jelent semmilyen jellegű biztonsági kockázatot.	Bemutatós.	Fontos	Csapat	

NFR4	A célfelhasználók számára könnyen kezelhető.	Bemutató	Fontos	Csapat	
------	--	----------	--------	--------	--

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	Játék leállítása
Rövid leírás	A játék végén a játékot a rendszer megállítja, majd kiválasztja a nyertes csapatot.
Aktorok	Kontroller
Forgatókönyv	A játék megáll, a játékosok nem léphetnek tovább. A lyukakon kifolyt, valamint a ciszternákba jutott vízmennyiséget a kontroller összeszámolja. Amelyik csapatnak még nem járt le az ideje, annak extra pontokat számol el. A pontok alapján eldől, hogy melyik csapat nyert.

Use-case neve	Játék léptetése
Rövid leírás	A Kontroller vezérli a játék előrehaladtát
Aktorok	Kontroller
Forgatókönyv	A Kontroller frissíti a játék elemeinek állapotát és egyesével felkéri a játékosokat, hogy lépjenek.

Use-case neve	Pumpát meghibásít
Rövid leírás	Egy kör során a pumpák véletlenszerűen, egymástól függetlenül meghibásodhatnak. Ezt a Kontroller dönti el.
Aktorok	Kontroller
Forgatókönyv	A Kontroller kiválaszt egy működő pumpát és adott valószínűséggel meghibásítja azt.

Use-case neve	Játék indítása
Rövid leírás	A csapatok megszervezése után bármelyik játékos elindíthatja a játékot a "Játék indítása" gombra kattintva a főmenüben
Aktorok	Játékos
Forgatókönyv	A játékos rányom a "Játék indítása" gombra, ekkor a játék elindul

Use-case neve	Játékos mozgatása
Rövid leírás	A játékosok mozgathatják a játékbeli karaktereiket
Aktorok	Játékos
Forgatókönyv	A játékos a szomszédos elemek közül valamelyikre léphet, ha az nem foglalt. (A pumpa soha nem lehet foglalt, mert azon akárhányan állhatnak.)

Use-case neve	Cső kilyukasztása
Rövid leírás	A szabotőr kilyukasztja a csövet amin áll
Aktorok	Szabotőr
Forgatókönyv	Ha a szabotőr olyan csövön áll, ami még nem lyukas, akkor kilyukasztja. Különben nem történik semmi.

Use-case neve	Pumpa állítása
Rövid leírás	A játékos beállítja, hogy a pumpa melyik csőből melyikbe pumpáljon.

Aktorok	Játékos
Forgatókönyv	A pumpába csatlakoztatott csövek közül kiválasztja a forrás és a cél csövet.

Use-case neve	Cső megjavítása
Rövid leírás	A szerelő megjavítja a csövet amin áll
Aktorok	Szerelő
Forgatókönyv	Ha a szerelő lyukas csövön áll, akkor megjavítja. Különben nem történik semmi.

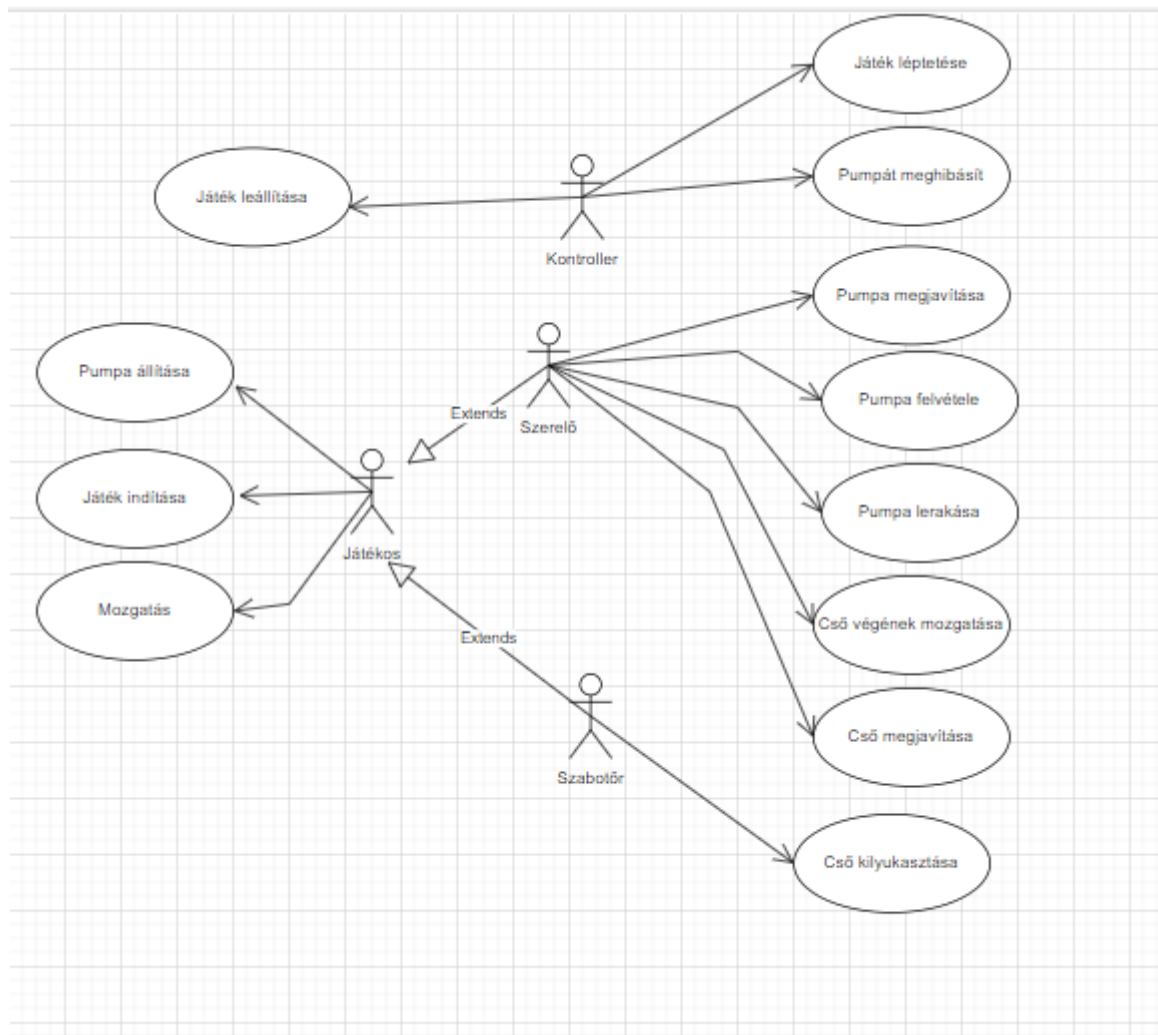
Use-case neve	Pumpa megjavítása
Rövid leírás	A szerelő megjavítja a pumpát amin áll
Aktorok	Szerelő
Forgatókönyv	Ha a szerelő olyan pumpán áll, ami elromlott, akkor megjavítja. Különben nem történik semmi.

Use-case neve	Pumpa felvétele
Rövid leírás	A szerelő felvesz egy pumpát
Aktorok	Szerelő
Forgatókönyv	A szerelő magához vesz egy pumpát. Magával viszi, később lerakhatja.

Use-case neve	Pumpa lerakása
Rövid leírás	A szerelő lerak egy pumpát
Aktorok	Szerelő
Forgatókönyv	Ha a szerelő csövön áll, ekkor a csövet kettévágja és a két keletkezett rész közé illeszti a pumpát. Különben nem történik semmi.

Use-case neve	Cső végének mozgatása
Rövid leírás	A szerelő felveheti a cső végét és azzal mozoghat. Később lerakhatja azt. Ekkor a cső csatlakozhat egy aktív elemhez.
Aktorok	Szerelő
Forgatókönyv	A játékos magához veszi a szomszédos cső végét. Ekkor az nála marad, ameddig le nem rakja.
Alternatív forgatókönyv	A játékos csatlakoztatja a cső végét egy szomszédos aktív elemhez.

2.4.2 Use-case diagram



2.5 Szótár

Aktív elem: Olyan elem, ami vizet tárolhat, vagy mozgathat, és a csőhöz kapcsolódik. Például: forrás, ciszterna, pumpa. A cső nem számít aktív elemnek. Az aktív elemeket csövek kötik össze, egymáshoz közvetlenül nem csatlakozhatnak.

Ciszterna: A csőrendszer része, ide érkezik a víz.

Cső ép: Ekkor a víz nem folyik a földre, a csőben szabadon áramolhat.

Cső lyukas: Ekkor a csőben lévő víz azonnal a földre folyik, nem juthat tovább a csövön.

Cső: A cső pumpához, forráshoz, vagy ciszternához csatlakozó elágazásmentes elem. Vizet tárolhat, de maga a cső nem mozgatja a vizet.

Csőrendszer/Rendszer: A játék a csőrendszeren folyik. A csőrendszer szállítja a vizet és a játékosok is azon mozognak.

Elem: A csőrendszer eleme. Például: cső.

Extra pont: A játék végén megmaradt idővel arányos mennyiségben elszámolt bónuszpont.

Forrás: A forrás a csőrendszer része, itt lép be a víz a rendszerbe.

Homok/Föld: A rendszernek nem része, de ide is érkezik víz.

Játékos: A játék szereplője, aki a csőrendszeren mozoghat és a szerepétől függő módon kezelheti a csőrendszer elemeit. Hivatkozhat a felhasználóra is, aki a szereplőt irányítja.

Játékos helye: A játékosok a csőrendszer elemein állnak. Az az elem, amin egy játékos áll, a játékos helye.

Játékosok kikerülnek egymást: A soron következő játékos elmegy a másik mellett egy pumpán.

Játék vége: Az a pillanat, amikor az egyik csapat összideje lejár. Nem történik több lépés és összesítjük a pontokat.

Kör: Egy teljes körnek az számít, amikor minden játékos léphetett.

Lépés: Egy lépés során az adott játékos kap valamennyi időt, hogy végezze a tevékenységét, azaz mozoghat, pumpákat állíthat, stb.

Lépés-idő: Az az időtartam, ami a játékosnak rendelkezésre áll egy lépés megtételéhez.

Össz-idő: Az az időtartam, ami egy csapatnak rendelkezésre áll összesen a játékra.

Pontok: A csapatok pontokért játszanak. A szerelők a ciszternákba jutott vízért a szabotőrök pedig a lyukakon kifolyt vízért kapnak pontot. Továbbá a játék végén extra pont jár annak a csapatnak, amelyiknek még van hátralévő ideje.

Pumpa elomlott: Ekkor a pumpa nem mozgatja a vizet.

Pumpa működik: Ekkor a pumpa a forrás- és célcső között mozgatja a vizet.

Pumpa: Több, de véges számú csövet összekötő elem. Van egy forrás- és egy célcsőve. Bemeneti és kimeneti csövekként is hivatkozhatunk ezekre.

Soron következő játékos: Az a játékos, aki éppen léphet.

Szabotőr csapat: A csapat célja, hogy minél több víz folyjon ki.

Szabotőr: A szabotőr csapatba tartozó játékos.

Szerelő csapat: A csapat célja, hogy a ciszternákba minél több víz jusson.

Szerelő: A szerelő csapatba tartozó játékos.

Szomszédos elem: A rendszer elemei hálózatot alkotnak. Két elem szomszédos, ha egymáshoz csatlakoznak. Egy elem szomszédos egy játékosal, ha szomszédos azzal az elemmel, amin a játékos áll.

Térkép: A csőrendszert jelző grafikai felület.

Víz: A vizet csőrendszer szállítja és ez alapján történik a csapatok pontozása.

Víztartály: A víz átemelése során átmeneti tárolóként funkcionál.

2.6 Projekt terv

A projekt lépései:

A projekt három fő mérföldkő teljesítését foglalja magába.

1. Szkeleton: Ebben a szakaszban először a csapat a kidolgozza a feladat leírása alapján a specifikációt. Ezt a csapat több megbeszélés során aprólékosan és nagy odafigyeléssel dolgozza ki. A feladatterv alapján a csapat megkezdheti a különböző modellek és azok dokumentációinak elkészítését. Ennek a szakasznak a végén a csapat leadja a programkód vázát az úgynevezett szkeletont és az ehhez kapcsolódó dokumentációkat.
2. Prototípus: Ebben a szakaszban a csapatnak a szkeleton és a megtervezett modellek alapján elő kell állítania a program egy már működőképes, de grafikus felületet nem tartalmazó verzióját. Ez azt jelenti, hogy a szkeleton fázisban megtervezett algoritmusokat implementálni kell és természetesen kezelni kell az ezzel felmerülő problémákat. A programnak ezen verzióját már alá kell vetni teszteknek, hogy biztosítsuk a hibamentes működést. Ennek a

szakasznak a végén a csapat leadja a már működőképes programkódját és az ehhez a szakaszhoz kapcsolódó dokumentációkat.

3. Grafikus felület: Az utolsó szakaszban a csapat feladata a program grafikus felületének elkészítése. A szakasz végén a csapat leadja a végső programot, amely már teljesen működőképes és az eddigi dokumentációk egyesített változatát.

Határidők:

Feladat neve	Határidő
Csapatalkítás	2023.03.03
Követelmény, projekt, funkcionalitás	2023.03.13
Analízis modell I.	2023.03.20
Analízis modell II.	2023.03.27
Szkeleton tervezése	2023.04.03
Szkeleton elkészítése	2023.04.17
Prototípus koncepciója	2023.04.24
Részletes tervek	2023.05.03
Prototípus elkészítése	2023.05.08
Grafikus változat tervei	2023.05.15
Grafikus változat elkészítése	2023.05.31
Egyesített dokumentáció	2023.06.02

Kommunikációs modell és eszközök:

A csapat kommunikációja egyszerre több csatornán zajlik. Ennek célja, hogy mindig a lehető legmegfelelőbb eszközt használjuk.

A felhasznált eszközök kommunikációs célokra:

- Discord: A Discord kommunikációs applikáció lehetővé teszi a csapattagok számára, hogy könnyedén tudjanak együtt dolgozni, ha gépközelben vannak. A program triviálissá teszi az értekezletek szervezését, hiszen a szerver folyamatosan létezik, bármikor be lehet csatlakozni 1-1 szobába és már lehet is beszélni. Másik előnye, hogy támogatja a képernyő megosztását, amellyel a csapattagok nagyon könnyedén magyarázhatják el egymásnak kérdéseiket / problémáikat. Nem utolsó sorban pedig a szöveges csatornák lehetővé teszik a dokumentumok "mentését", így azok nem vesznek el, vissza lehet őket bármikor nézni, ki, mikor, mit csinált / küldött.
- Facebook Messenger: Ha a csapattagok nincsenek gépközelben a Facebook Messengeren mégis mindenki a nap legnagyobb részében elérhető. Itt lehet megbeszélni az értekezletek helyét és idejét melyet élőben vagy Discordon végzünk.
- Google Drive/Docs és az ott lévő kommentek: A Google Docs-ban nagyon könnyen lehet a dokumentumok szélén kommenteket hagyni, amelyek nagy segítséget nyújtanak a többi csapattagnak a már elvégzett feladatok / feladatrészek megértésében és a még el nem készült felismerésében
- Fizikai megbeszélések: A csapat nagyrészt az élőben tart értekezleteket.

Verziókezelési modell és eszközök:

- Dokumentumok verziókezelése: A szöveges dokumentumokat és dokumentációkat a Google Drive és Google Docs segítségével verziókezeljük. A különböző sablonok egy központi, minden csapattal megosztott mappába kerülnek, ahol minden csapattag, idejét figyelembe véve, egyenlő megosztásban dolgozik rajta.
- Kód verziókezelése: A kód verziókezelésére a Git-et használjuk. Itt egy mindenkivel megosztott fő repositoryba fogjuk a kódot, hasonlóan a dokumentumokhoz, egyenlő megosztásban írni, fejleszteni. Minden változtatás a kódon ide fog kerülni, innen fogjuk a minden beadás előtt letölteni majd beadni.

A kódoláshoz használt IDE-k: IntelliJ, VSCode

A csapat és beosztásaik:

Név	Beosztás, feladatok
Szautner Béla László	dokumentáció, tesztelés, kódolás, projekt vezetése
Tóth Gábor	dokumentáció, tesztelés, kódolás
Koleszár Kristóf	dokumentáció, tesztelés, kódolás
Kovács Zoltán	dokumentáció, tesztelés, kódolás
Gegő Levente	dokumentáció, tesztelés, kódolás

2.7 Napló

Kezdet	Időtartam	Résztevők	Leírás
2023.03.08. 11:00	3 óra	Szautner Kovács Gegő Tóth	Értekezlet: dokumentáció elkezdése
2023.03.08. 16:30	1 óra	Koleszár Gegő	2.1.3. Definíciók, rövidítések
2023.03.08. 16:30	1 óra	Koleszár	2.2.2 Funkciók elkezdése
2023.03.08. 20:00	2 óra	Koleszár	2.2.2 Funkciók folytatása
2023.03.09 11:30	1 óra	Kovács	2.4.1 A use casek leírásainak megírása
2023.03.10 14:00	1 óra	Kovács	2.6 A projekt tervének megírása
2023. 03. 10. 15:00	2 óra	Koleszár Szautner	2.3.1 Funkcionális követelmények elkezdése
2023. 03. 11. 14:30	2 óra	Szautner Gegő Tóth	Értekezlet: játékszabályok specifikációja.
2023. 03. 11 16:30	2 óra	Tóth	2.3 Követelmények megfogalmazása.
2023.03.12. 11:00	2,5 óra	Gegő	dokumentáció revízió
2023.03.12. 11:30	2 óra	Koleszár	2.2.2. Funkciók megbeszélése, véglegesítése 2.3. Funkcionális követelmények véglegesítése 2.5 Szótár befejezése Dokumentum újraolvasása, kisebb hibák javítása
2023.03.12 10:00	1.5 óra	Tóth	2.2.2. Funkciók folytatása. 2.3 Funkcionális és egyéb követelmények revíziója, javítás és bővítése.
2023.03.12 12:00	2 óra	Kovács	A teljes dokumentációban javítások
2023.03.12 19:30	1.5 óra	Kovács Gegő Szautner Koleszár Tóth	Értekezlet: A teljes dokumentáció ellenőrzése, javítások

3. Analízis modell kidolgozása

3.1 Objektum katalógus

Csapat: 2 vagy több játékost összefoglaló, a játékosok összpontszámáért és összidejéért, valamint a soron következő (csapaton belüli) játékos kiválasztásért felel. A szerelők, illetve szabotőrök is külön-külön alkotnak egy-egy csapatot.

Szerelő: Olyan játékbeli karakter, azaz belépési pont a játékosok számára a játék (sivatagi vízhálózat) világába, amely azért felel, hogy a csőrendszeren keresztül minél több víz folyjon be a ciszternákba. Feladatai közé tartozik a lyukas csövek megjavítása, új cső lerakása, pumpa lerakása, pumpa átállítása.

Szabotőr: Olyan játékbeli karakter, azaz belépési pont a játékosok számára a játék (sivatagi vízhálózat) világába, amely azért felel, hogy minél több víz folyjon ki a csőhálózatból, valamint minél kevesebb víz folyjon be a ciszternákba. Feladatai közé tartozik a csövek kilyukasztása, pumpa átállítása.

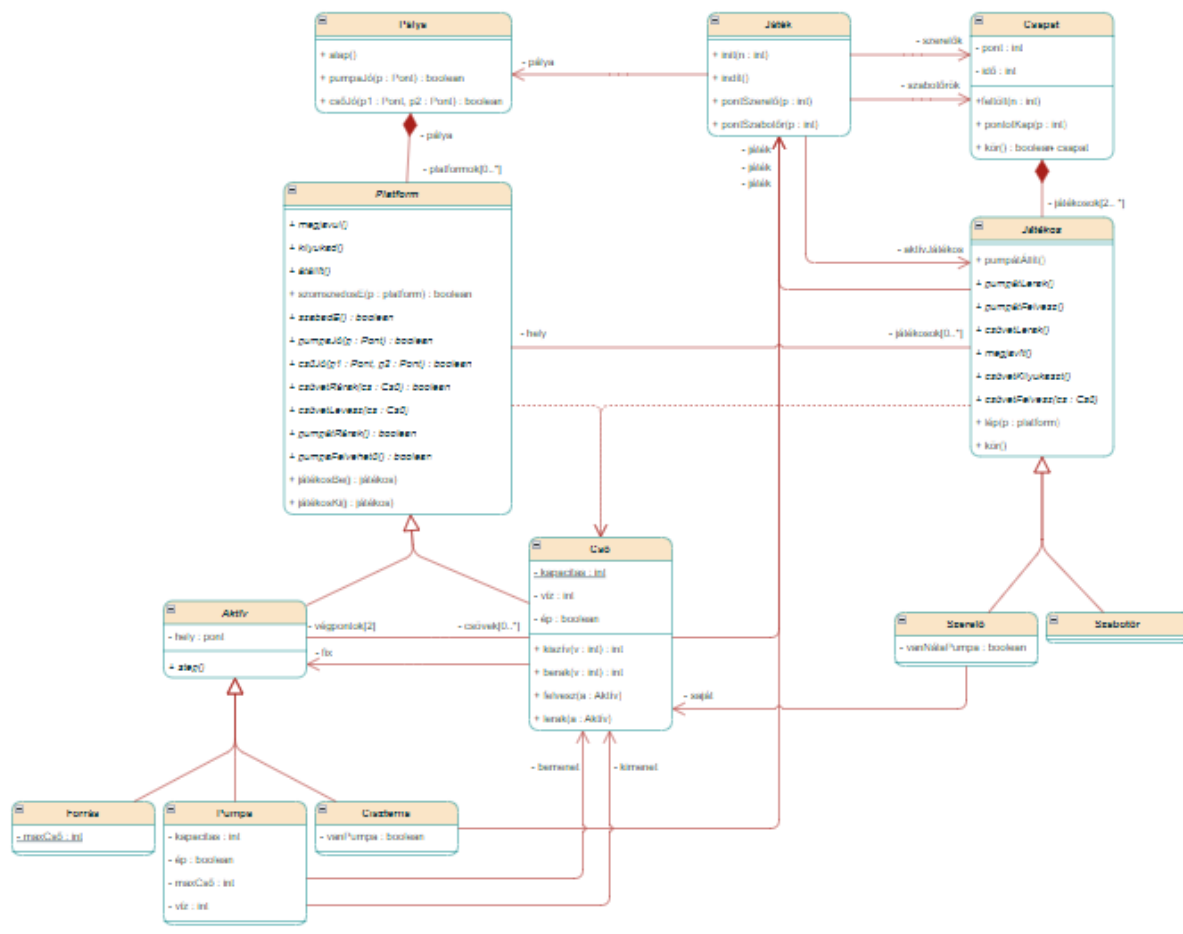
Cső: Olyan objektum a játékban, amelynek a felelősségei közé tartozik, hogy két pumpa, forrás-pumpa, pumpa-ciszterna között vizet tudjon szállítani, amennyiben ép, és vizet engedjen ki a sivatagba, amennyiben lyukas. A csövekhez kapcsolódnak a játékbeli aktív elemek.

Forrás: Olyan objektum, amely azért felel, hogy játékba víz kerülhessen. Ez teszi lehetővé, hogy a rá csatlakoztatott csőben víz legyen. A Forrást nem lehet "tönkretenni", így mindig áramlik be víz a játékba.

Pumpa: Olyan objektum, amely lehetővé teszi több megadott hosszúságú cső összekapcsolását, valamint felelőssége kiterjed arra is, hogy meg lehet határozni melyik két csövet kapcsoljon aktívan össze, azaz melyik csőből melyik másik csőbe kerüljön az egységnyi víz, amennyiben a pumpa üzemel. A pumpa csak 1 bemeneti és 1 kimeneti csövet tud összekötni, azaz több csőből több csőbe nem tud pumpálni.

Ciszterna: Olyan objektum, amely azért felel, hogy játékban lévő víz pontként jóváírható legyen. Hasonlóan működik mint a Forrás objektum, azaz ezt sem lehet "tönkretenni", tehát amennyiben a hozzá kapcsolódó csőben van x egységnyi víz, abban az esetben az x egységnyi víz befolyik a Ciszternába, és pontként jóváíródik a Szerelő csapatnak.

3.2 Statikus struktúra diagramok



3.3 Osztályok leírása

3.3.1 Aktív

- **Felelősség**

A pálya aktív elemeinek az ősosztálya. Biztosítja a léptethetőséget. Absztrakt osztály.

- **Ősosztályok**

Platform

- **Interfészek**

—

- **Attribútumok**

- **Cső[] csövek:** Az aktív elembe csatlakoztatott csövek.
- **Pont hely:** Az aktív elem helye a pályán (grafikus megjelenítéshez).

- **Metódusok**

- **void step():** Az aktív elem minden gametick-ben frissíti az állapotát. **Absztrakt**
- **csövetLevesz(Cső cs):** A paraméterként kapott cs csövet leveszi magáról.
- **boolean csövetRáak(Cső cs):** A paraméterként kapott cs csövet ráakja magára. Visszatérési értéke, hogy ez megtehető volt-e.

3.3.2 Ciszterna

- **Felelősség**

A játékban egy ciszternát reprezentál. A játék során a ciszterna minden csőből, amely hozzá van kötve kivesz egy egység vizet, ez egyenlő azzal, hogy a ciszternába befolyt az a vízmennyiség. Minden befolyt vízmennyiség után ad a szerelők csapatának egy pontot.

- **Ősosztályok**

Platform → Aktív

- **Interfészek**

-

- **Attribútumok**

- **boolean vanPumpa:** Megadja, hogy a ciszternán van-e pompa.
- **Játék játék:** A játék példánya.

- **Metódusok**

- **void step():** Minden gametick-ben a ciszterna minden csőből, amely hozzá kapcsolódik kivesz egy egység vizet, majd a befolyt vízmennyiséget pontokban a szerelők csapatának adja.
- **boolean pompaFelvehető():** Megadja, hogy a ciszternáról fel lehet-e venni pumpát.

3.3.3 Csapat

- **Felelősség**

A játékot játszó egy csapatba tartozó játékosokat tartalmazza. Számon tartja a csapat pontjait és hátralévő összidejét, illetve lépteti, hogy ki a soron következő játékos.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **Játékos[] játékosok:** A csapatba tartozó játékosok.
- **int pont:** A csapat pontjai.
- **int idő:** A csapat összideje mp-ben.

- **Metódusok**

- **void pontotKap(int p):** A csapat a p paraméterben megadott pontot kap, így növelve annyival a pontjainak számát.
- **boolean kör():** Minden a csapatba tartozó játékost felkéri a körére. Folyamatosan csökkenti a csapat összidejét addig, amíg minden, a csapatba tartozó játékos be nem fejezte a körét.
- **void feltölt(int n):** Létrehoz a csapatba n db játékost. Ezeket hozzá is adja a csapathoz.

3.3.4 Cső

- **Felelősség**

A cső a játékban két aktív elem összekötésére és víz tárolására szolgál.

- **Ősosztályok**

Platform

- **Interfészek**

-

- **Attribútumok**

- **static int kapacitás:** A cső kapacitása víz egységben mérve. Ennél több víz nem kerülhet bele.
- **int víz:** Tárolja, hogy mennyi víz van éppen a csőben.
- **boolean ép:** A cső állapotát tárolja, ha a cső nem ép, a benne lévő víz elfolyik.
- **Játék játék:** A játék példánya.
- **Aktív fix:** Az az oldal, amelyet éppen nem mozgatunk.
- **Aktív[] végpontok:** A cső végpontjai.

- **Metódusok**

- **int kiszív(int v):** A csőből v számú vizet próbál kiszívni. Visszatérési értéke a vízmennyiség, amennyit ki lehet szívni (pl.: víz = 3, v = 4 → return = 3), ennyivel kevesebb lesz a csőben a víz.
- **int berak(int v):** A csőbe v számú vizet próbál berakni. Ha a cső túllépné így a kapacitását, akkor kapacitás mennyiségű víz lesz benne. Visszatérési értéke (v - az a vízmennyiség amennyi a csőbe bele tudott menni).
- **void felvesz (Aktív a):** Egy cső felvétele. Az argumentum a cső oldalát adja meg.
- **void lerak (Aktív a):** Egy cső lerakása. Az argumentum azt adja meg, hogy melyik aktív elemre rakjuk rá a csövet.
- **boolean pumpátRarak():** Egy cső közepére egy pumpát megpróbál elhelyezni. Visszaadja, hogy ez sikeres volt-e.
- **void kilyukad():** Cső kilyukasztása.
- **void megjavul():** Cső megjavítása.

3.3.5 Forrás

- **Felelősség**

A forrás szolgáltatja a vizet a hozzá csatlakoztatott csövekbe.

- **Ősosztályok**

Platform → Aktív

- **Interfészek**

-

- **Attribútumok**

- **static int maxCső:** A forrásba kapcsolódó csövek maximális száma.

- **Metódusok**

- **void step():** Minden gametick-ben a forrás a hozzá csatlakoztatott csövekbe egy egység vizet rak. Bizonyos időközönként a forráson cső keletkezik. Ezeknek a csöveknek az egyik vége a forrásba, a másik végük a semmibe csatlakozik.
- **boolean csövetRáarak(Cső cs):** A paraméterként kapott cs csövet ráarakja magára. Visszatérési értéke, hogy ez megtehető volt-e.

3.3.6 Játék

- **Felelősség**

A játék lebonyolításáért felelős osztály.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **Játékos aktívJátékos:** Az éppen soron következő játékos.
- **Csapat szabotőrök:** A szabotőrök alkotta csapat.
- **Csapat szerelők:** A szerelők alkotta csapat.
- **Pálya pálya:** A jelenlegi pálya. Ez alapesetben az alapértelmezett pálya.

- **Metódusok**

- **void indít():** Ez a metódus indítja el a játékot, illetve tölti be a pályát és kezdi meg a játékot.
- **void pontSzerelo(int p):** A szerelő csapat p pontot kap.
- **void pontSzabotor(int p):** A szabotőr csapat p pontot kap.
- **void init(int n):** A játék kezdeti állapotát inicializáló függvény. Az n paraméter megadja a játékosok számát.

3.3.7 Játékos

- **Felelősség**

A játékosok karaktereit reprezentáló absztrakt őszosztály. Biztosítja a karakterek alapvető akcióit.

- **Őszosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **Platform hely:** Az a Platform, amin a játékos áll.
- **Játék játék:** A játék példánya.

- **Metódusok**

- **void pumpátÁllít()** : A játékos beállítja a pumpát, amin éppen áll. Ha nem azon áll, akkor nem történik semmi.
- **void pumpátLerak()**: A játékos megpróbálja a nála lévő pumpát lerakni. **Absztrakt**
- **void pumpátFelvesz()**: A játékos megpróbálja a platformról amin áll felvenni a pumpát. **Absztrakt**
- **void csövetLerak()**: A játékos megpróbálja lerakni a nála lévő csövet. **Absztrakt**
- **void csövetFelvesz(Cső cs)**: A játékos megpróbálja felvenni a cs csövet. **Absztrakt**
- **void lép(platform p)**: A játékos megpróbál a p platformra lépni. Ha ez nem sikerül, akkor a saját platformján marad.
- **void kör()**: A játékost felkéri a csapata a körre, ilyenkor kap egy előre meghatározott időt a játékos, ami alatt cselekedhet akármennyit. Az idő lejártával a játék automatikusan elveszi az irányítást a játékostól. A játékos az idő lejártá előtt is befejezheti a körét.
- **void megjavít()**: A játékos megpróbálja megjavítani azt a Platformot, amin éppen tartózkodik. **Absztrakt**
- **void csövetKilyukaszt()**: A játékos megpróbálja kilyukasztani azt a Platformot, amin éppen áll.

3.3.8 Pálya

- **Felelősség**

A játék pályáját reprezentálja. Ez az osztály tárolja a pálya elemeit és ellenőrzi, hogy a pályán kért módosítás (pl. pumpa lerakása) legális-e.

- **Őszosztályok**

-

- **Interfészek**

-

- **Attribútumok**
 - **platform[] platformok:** A pálya platformjai.
- **Metódusok**
 - **boolean pumpaJó(Pont p):** Visszaadja, hogy egy pumpát az adott p pontra el lehet-e helyezni.
 - **boolean csőJó(Pont p1, Pont p2):** Visszaadja, hogy egy, a p1 és p2 pontokat összekötő csövet le lehet-e rakni.
 - **void alap():** Beállítja az alapértelmezett pályát (ez magába foglalja a pálya elemeinek legenerálását/létrehozását).

3.3.9 Platform

- **Felelősség**

A játék mezőit reprezentáló absztrakt őosztály. Biztosítja a mezők számára az alapvető viselkedéseket.
- **Őosztályok**

-
- **Interfészek**

-
- **Attribútumok**
 - **Játékos[] játékosok:** A platformon tartózkodó játékosok listája.

- **Metódusok**

- **void megjavul():** A platform újra rendesen működik. **Absztrakt**
- **void kilyukad():** A platformot tönkreteszik. **Absztrakt**
- **void átállít():** A platformot átállítják. **Absztrakt**
- **boolean szomszedosE(Platform p):** Visszaadja, hogy két platform szomszédos-e.
- **boolean szabadE():** Visszaadja, hogy az adott platform szabad-e.
- **boolean pumpaJó(Pont p):** Visszaadja, hogy a platform és a p pontra helyezett pumpa közt nincs-e konfliktus.
- **boolean csőJó(Pont p1, Pont p2):** Visszaadja, hogy a platform és a p1 és p2 pontokat összekötő cső között nincs-e konfliktus.
- **boolean pumpaFelvehető() :** Visszaadja, hogy a pumpát felveheti-e egy játékos. **Absztrakt**
- **void pumpátRáarak():** A pumpát elhelyezi egy cső közepén. Ehhez a csövet középen ketté kell vágni. **Absztrakt**
- **void játékosBe(Játékos j):** A j játékost hozzáadja a platform saját magához.
- **void játékosKi(Játékos j):** A j játékost a platform eltávolítja saját magáról.
- **boolean csövetRáarak(Cső cs):** A platformra megpróbálunk felcsatlakoztatni egy csövet. **Absztrakt**
- **void csövetLevesz(Cső cs):** A cs csövet levesszük a platformról. **Absztrakt**

3.3.10 Pumpa

- **Felelősség**

A pumpa egy olyan aktív elem, amihez a hozzá csatlakoztatott csövek közül - ha a pumpa ép - az egyik kiválasztott csőből pumpálja pontosan egy másik kiválasztott csőbe a vizet. Ezt úgy teszi, hogy a bejövő csőből kiveszi, a kimenő csőbe pedig berak egy egység vizet gametick-enként. Emellett minden gametick-ben megvan az esélye, hogy a pumpa elromlik.

- **Ősosztályok**

Platform → Aktív

- **Interfészek**

-

- **Attribútumok**

- **boolean ép:** A pumpa állapotát tárolja, ha a pumpa nem ép, tehát elromlott, akkor nem pumpálja a vizet.

- **int víz:** A pumpa belső tartályában lévő víz mennyisége.
- **int kapacitás:** A pumpa tartályának kapacitása.
- **Cső kimenet:** A pumpa ebbe a csőbe pumpálja a vizet.
- **Cső bemenet:** A pumpa ebből a csőből pumpálja a vizet.
- **int maxCső:** A pumpára maximálisan csatlakoztatható csövek száma.
- **Metódusok**
 - **void step():** A pumpa minden gametick-ben a bejövő csőből egy egység vizet megpróbál kivenni, majd a kivett vízmennyiséget berakja a kimenő csőbe. Ebben a metódusban történik a pumpa véletlenszerű elromlása is.
 - **void átállít():** A függvény meghívásakor a felhasználó ki tud választani két csövet, amelyek közül az egyik az új bemeneti, a másik az új kimeneti csőve lesz a pumpának.
 - **void megjavul():** A pumpán amikor meghívják ezt a függvényt, akkor ezek után ismét működni fog, mindaddig, amíg ismét el nem romlik.
 - **boolean csövetRáak(Cső cs):** A paraméterként kapott cs csövet ráakja magára. Visszatérési értéke, hogy ez megtehető volt-e.

3.3.11 Szerelő

- **Felelősség**

A játékban egy szerelő karakter. Ő tudja a csöveket és pumpákat javítani, felvenni és lerakni.

- **Őszosztályok**

Játékos

- **Interfészek**

-

- **Attribútumok**

- **boolean vanNálaPumpa():** Ha a szerelőnél van pumpa igaz, ha nincs hamis.
- **Cső saját:** A szerelőnél lévő cső.

- **Metódusok**

- **void pumpátLerak():** A szerelő megpróbálja lerakni a nála lévő pumpát a platformra, amin áll.
- **void pumpátFelvesz():** A szerelő megpróbál felvenni egy pumpát arról a platformról, amin áll.
- **void csövetFelvesz(Cső cs):** A szerelő megpróbálja felvenni a cs csövet.
- **void csövetLerak():** A szerelő megpróbálja a csövet lerakni a platformra, amin áll.
- **void megjavít():** A szerelő megpróbálja megjavítani a platformot, amin áll.

3.3.12 Szabotőr

- **Felelősség**

A játékban egy szabotőr karakter. Ő tudja a csöveket kilyukasztani és a pumpákat átállítani.

- **Őszosztályok**

Játékos

- **Interfészek**

-

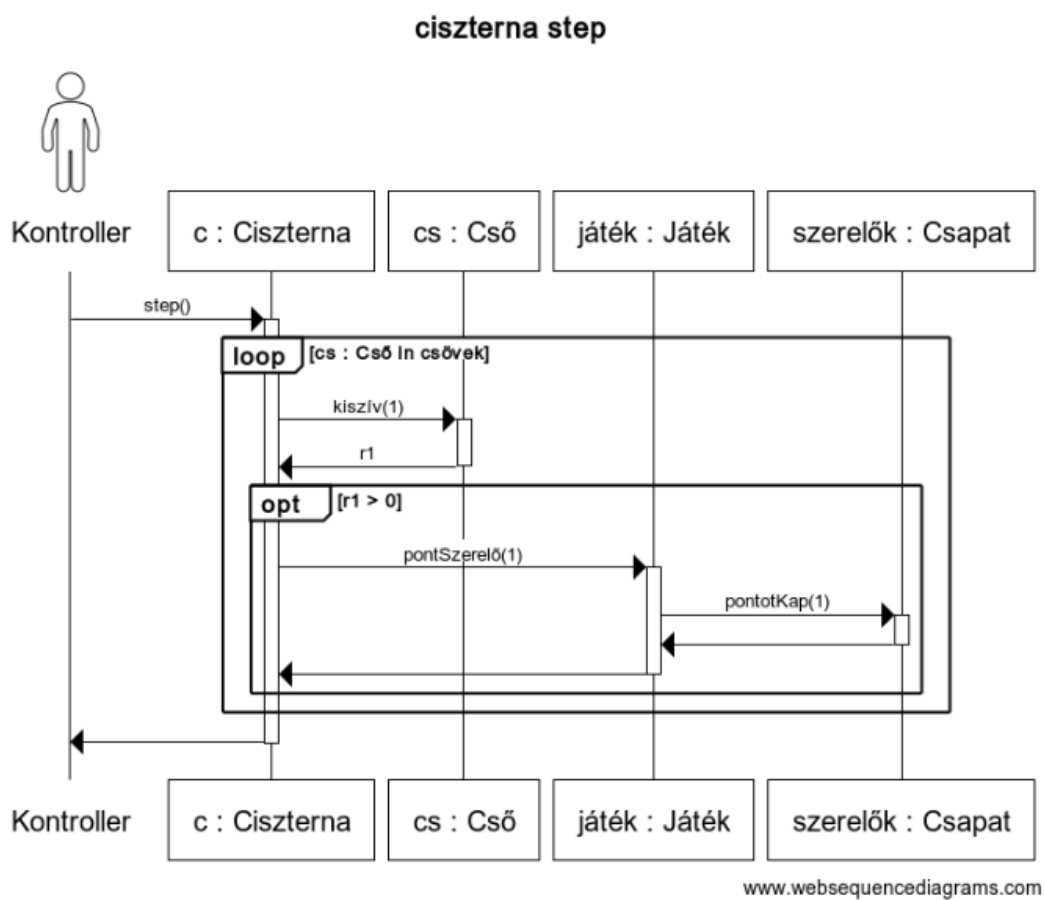
- **Attribútumok**

-

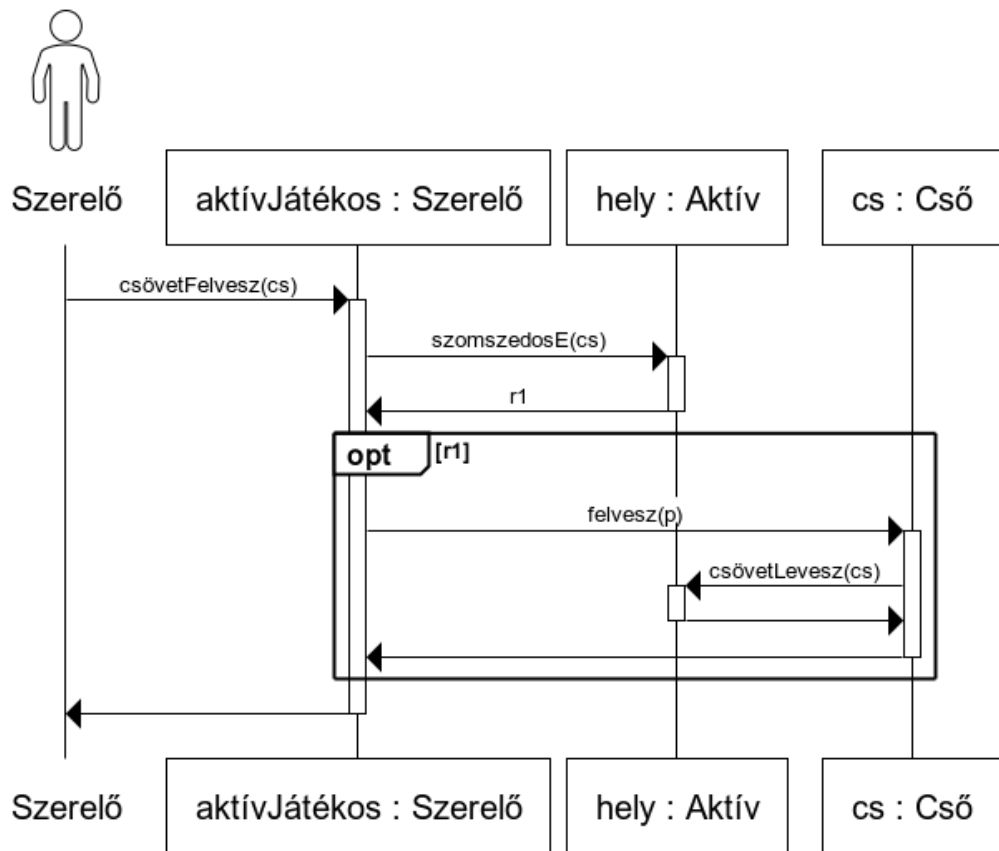
- **Metódusok**

- **void csövetKilyukaszt():** A szabotőr megpróbálja kilyukasztani a csövet amin áll.

3.4 Szekvencia diagramok

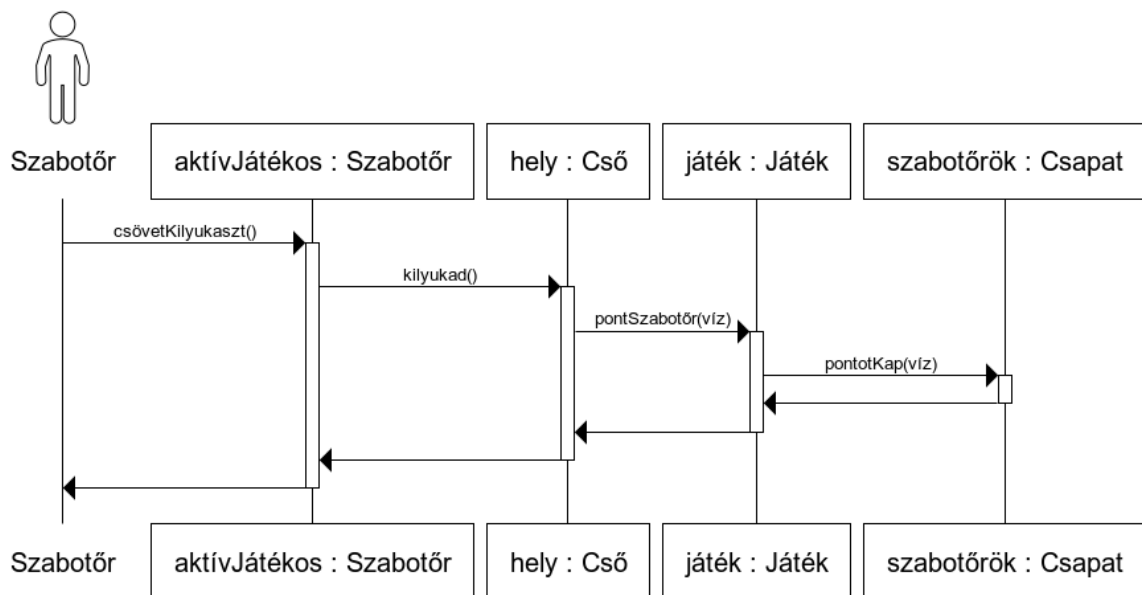


csövet felvesz

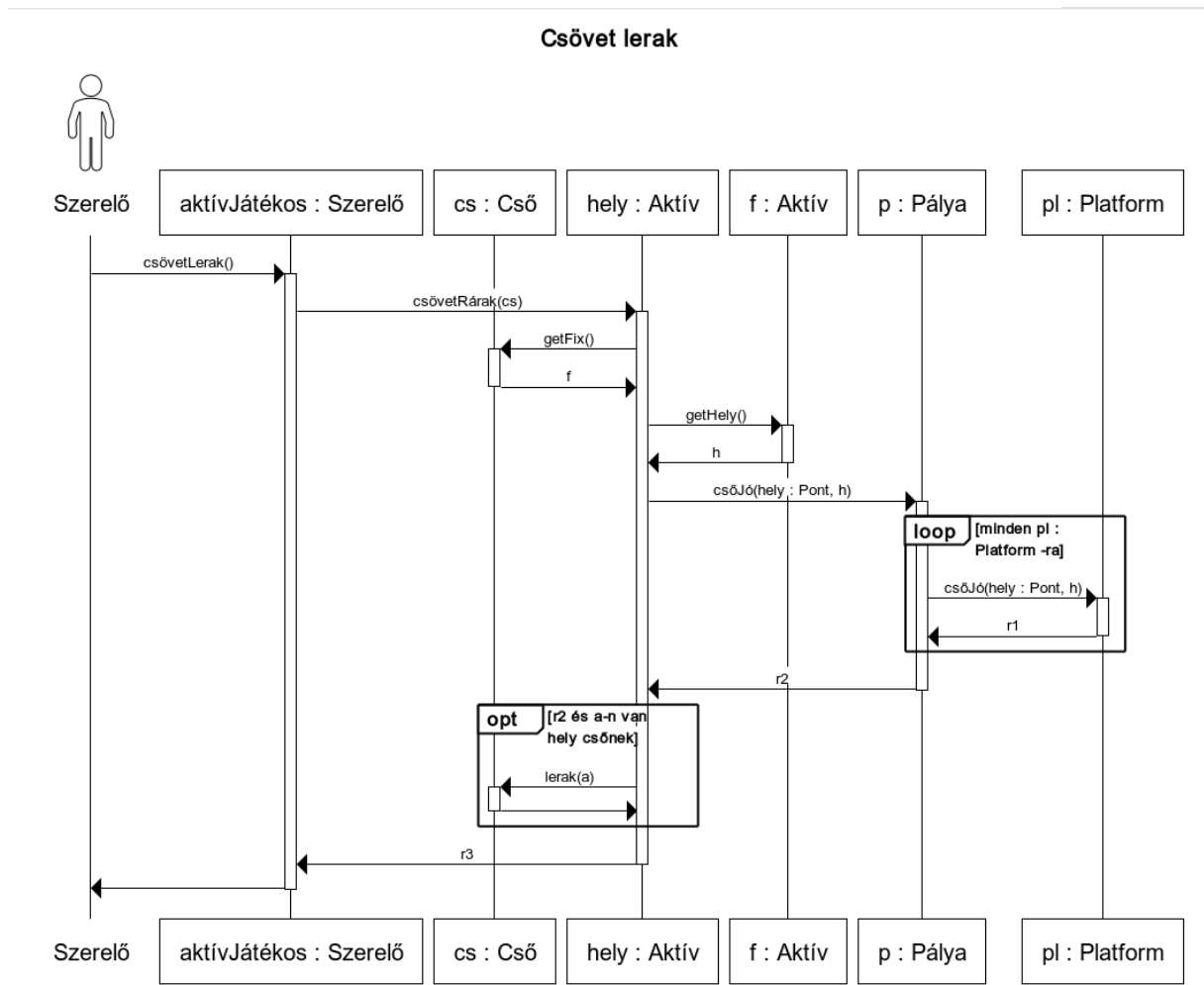


www.websequencediagrams.com

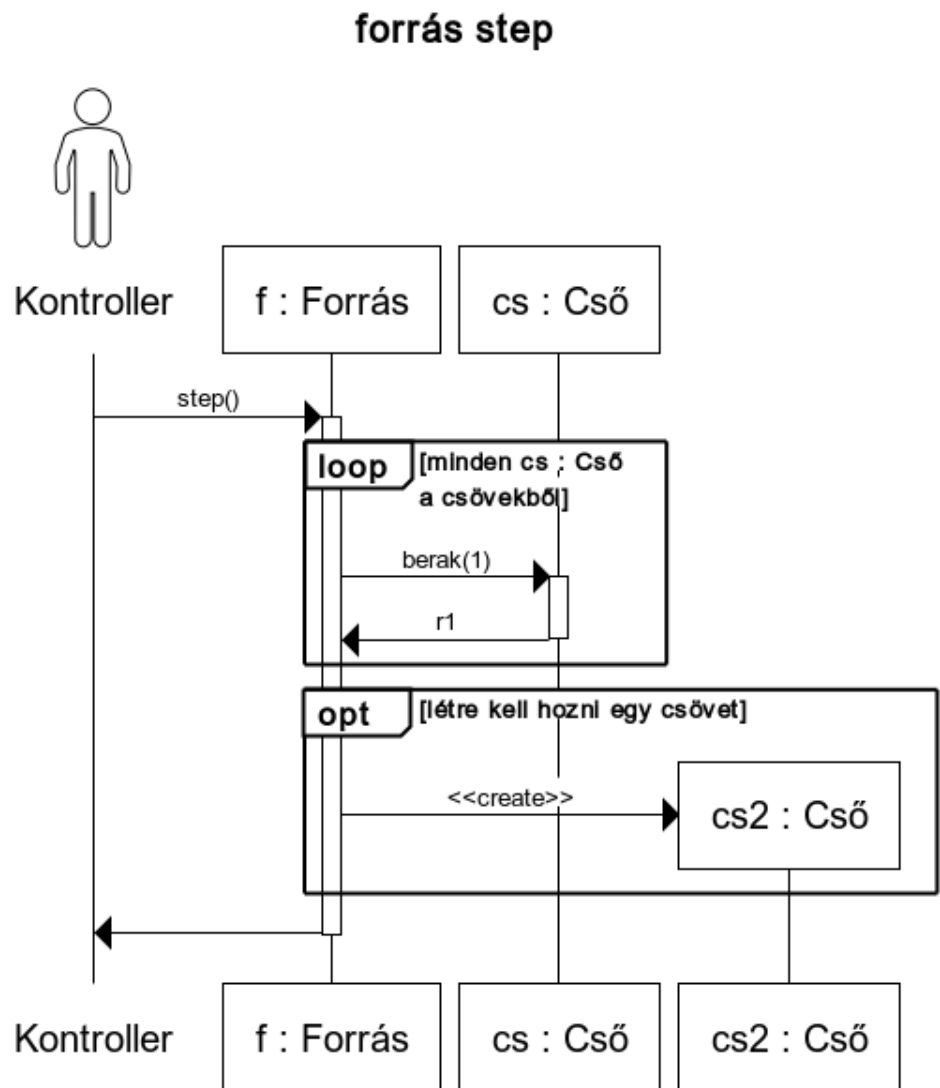
csövet kilyukaszt



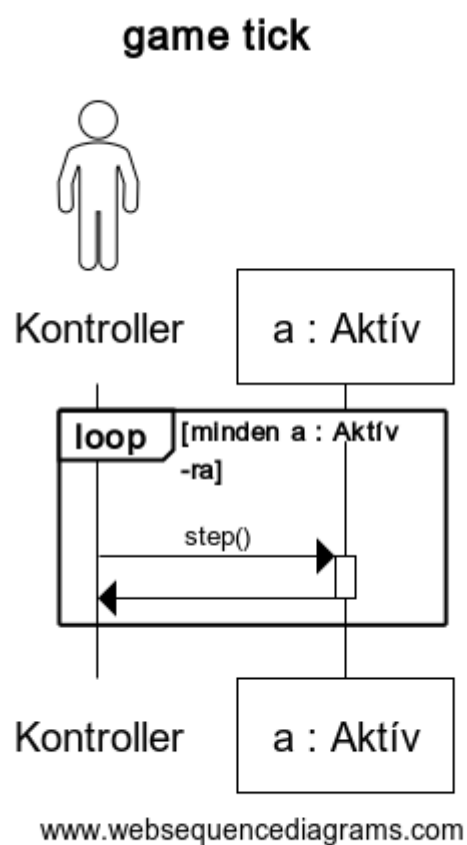
www.websequencediagrams.com

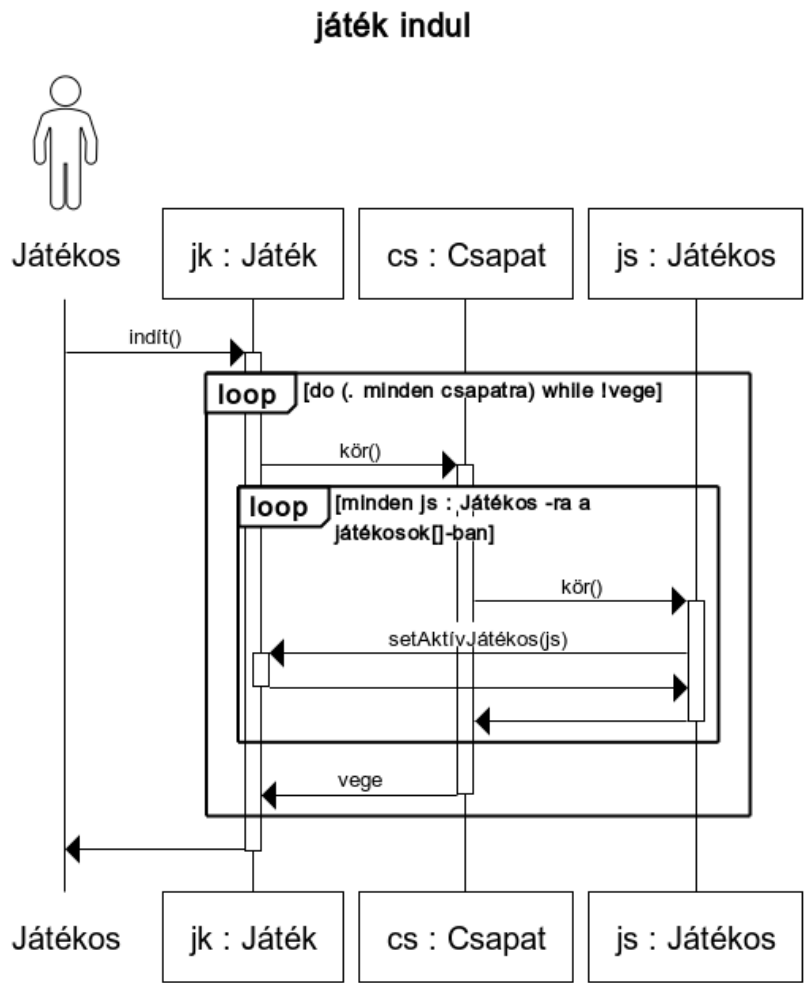


A csövet lerak szekvencia diagramban a hely : Aktív azt az aktív elemet reprezentálja amelyen a szerelő áll és ezáltal ismeri. A hely: Pont viszont a pálya egy megadott pontját jelzi (ez a grafikus megjelenítéshez kell).

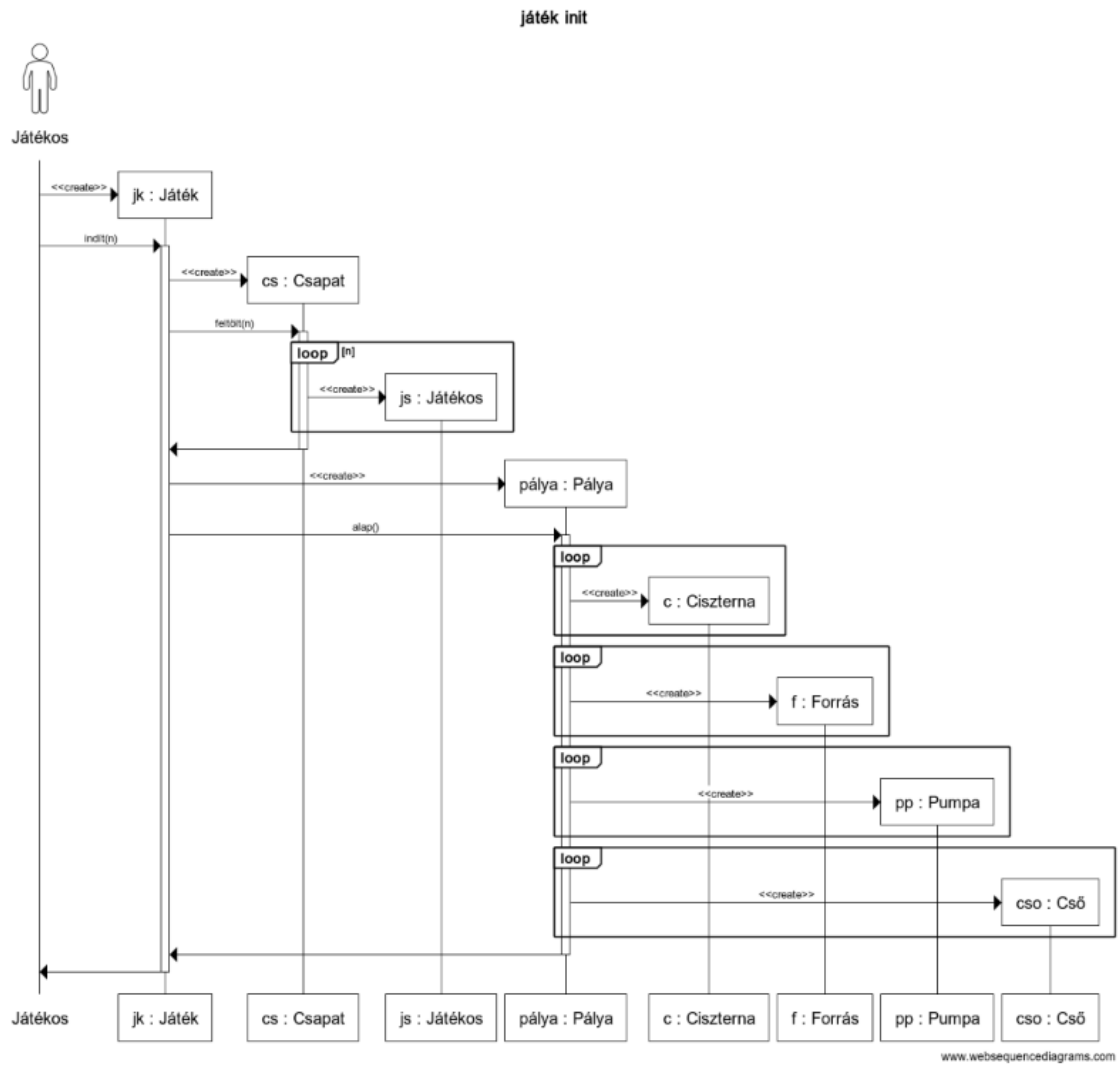


www.websequencediagrams.com

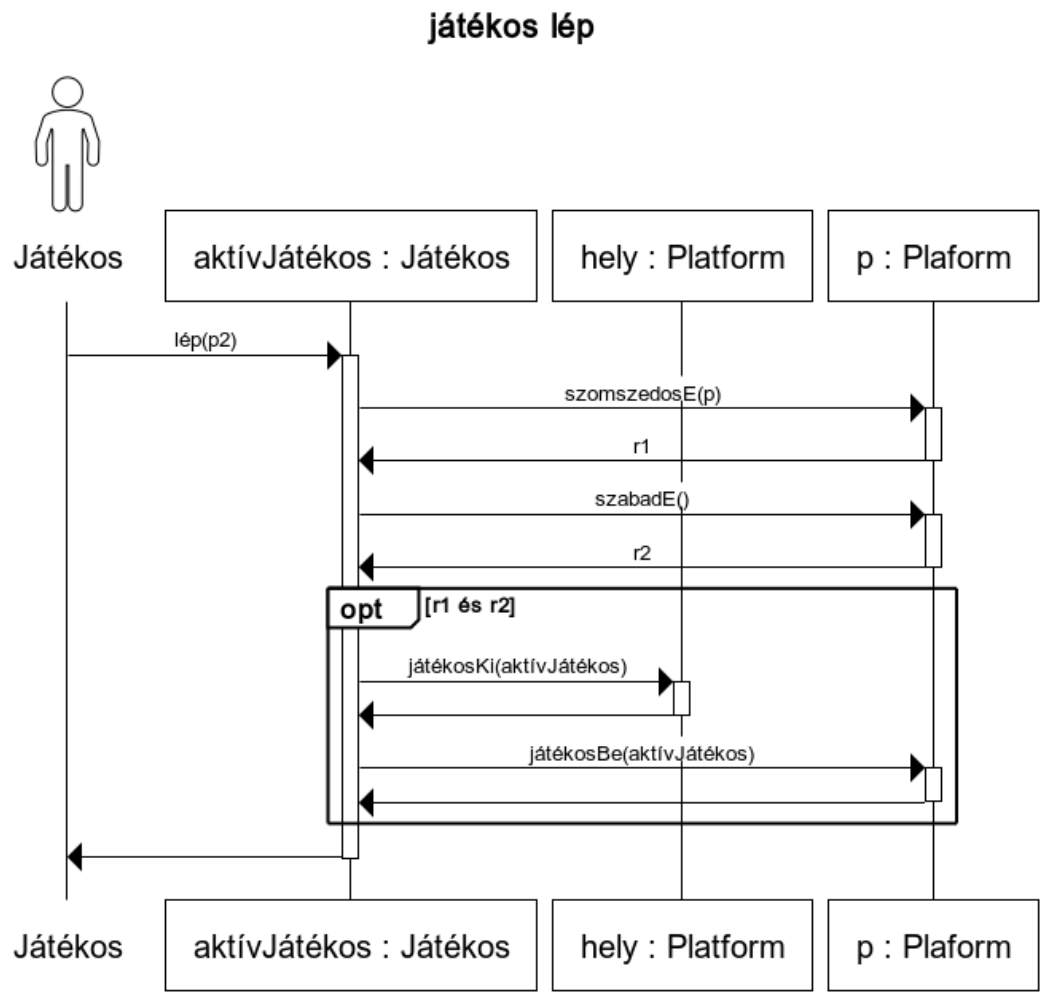


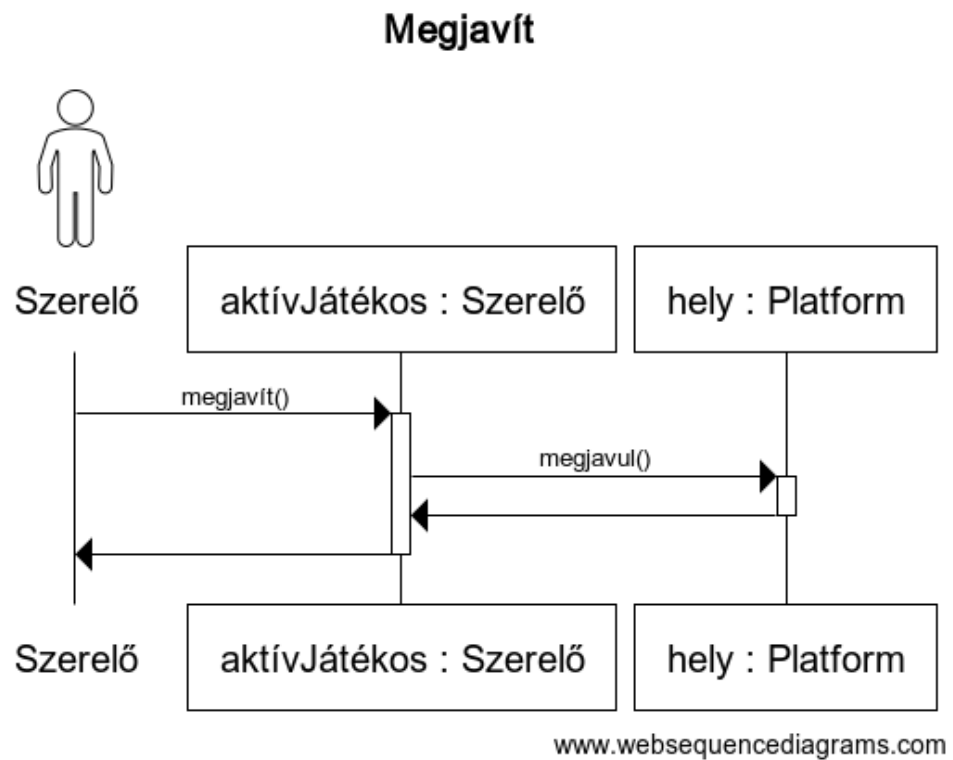


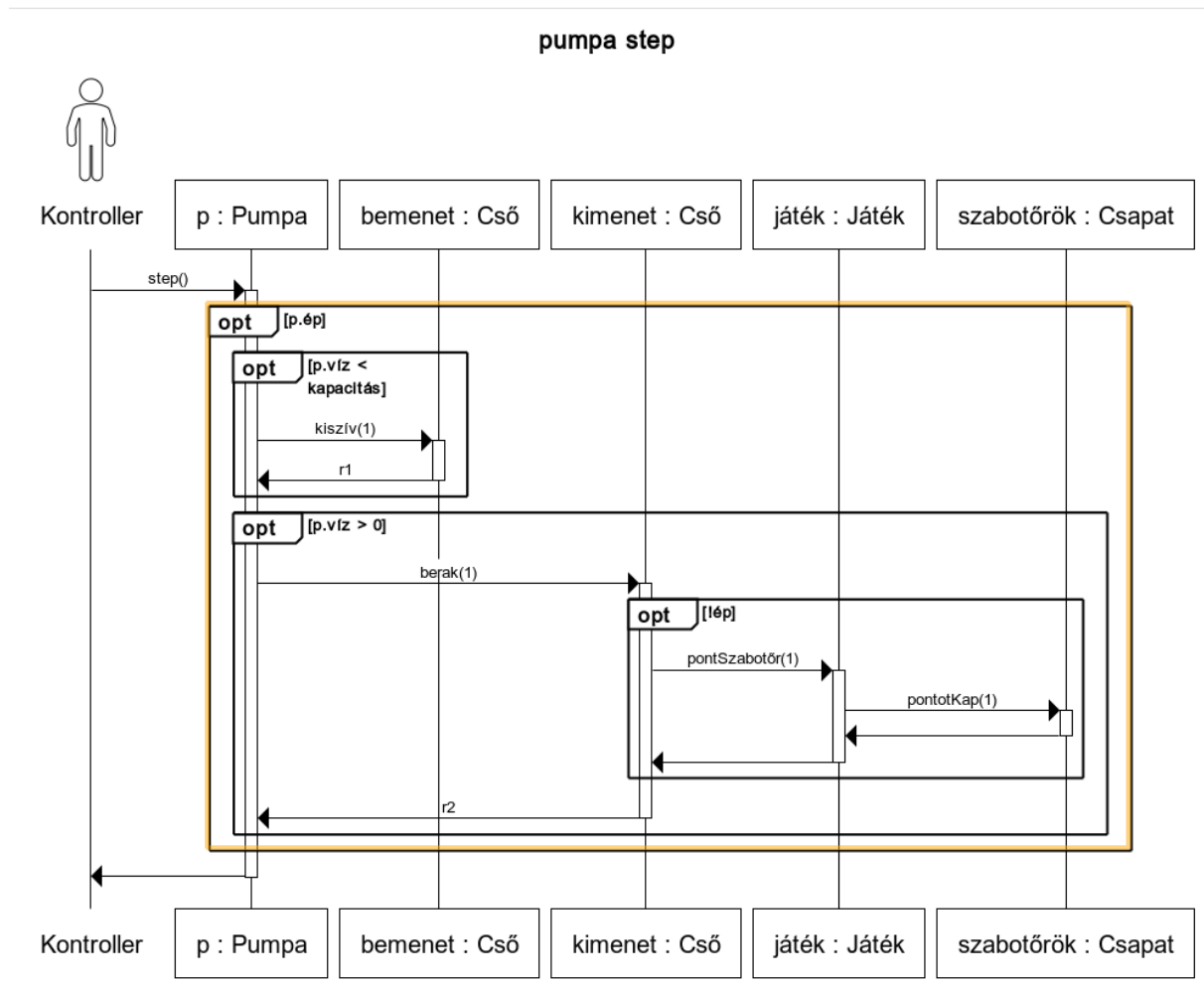
www.websequencediagrams.com



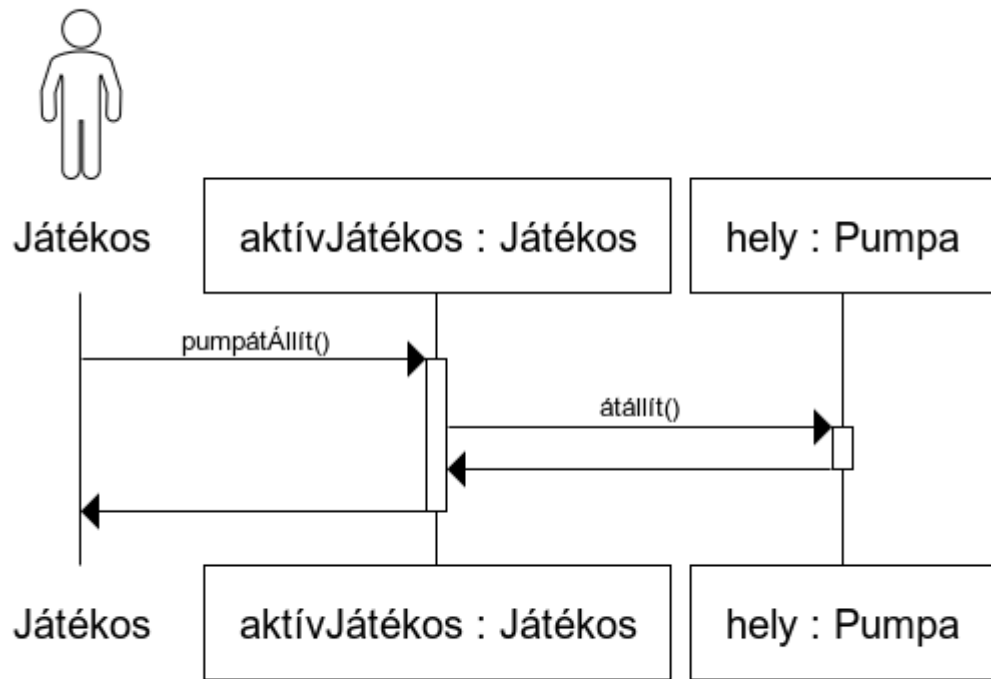
Itt a pálya loopjaiban mindig a megfelelő számú elem kerül létrehozásra. (ezt előre nem tudjuk, hogy mennyi így a loopba nem írtunk semmit)





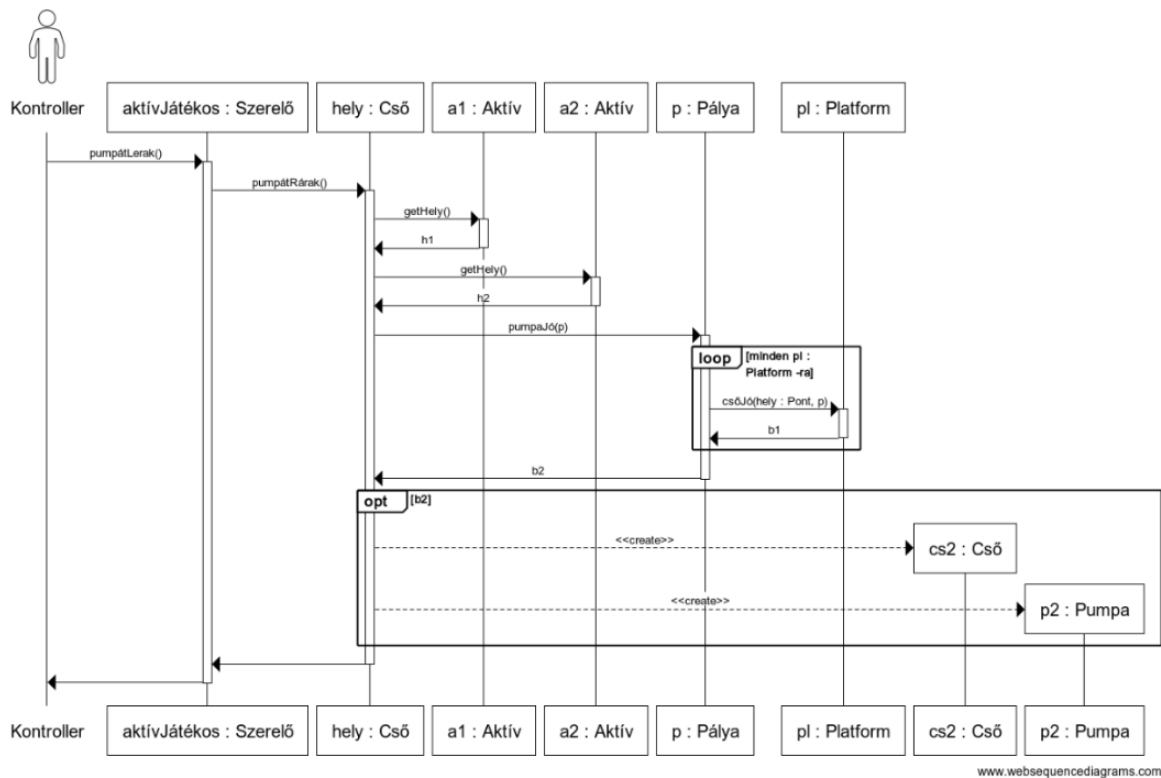


Pumpát állít



www.websequencediagrams.com

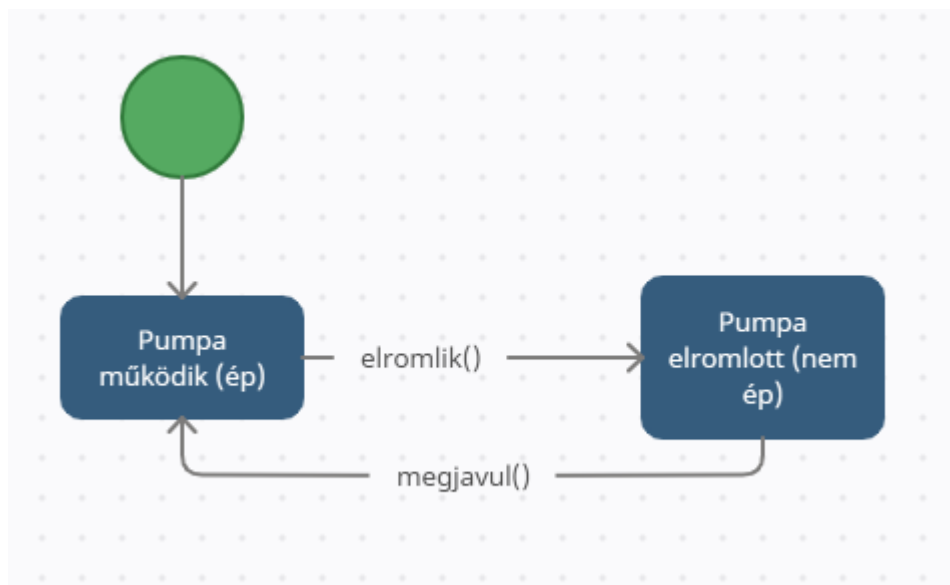
Pumpát lerak



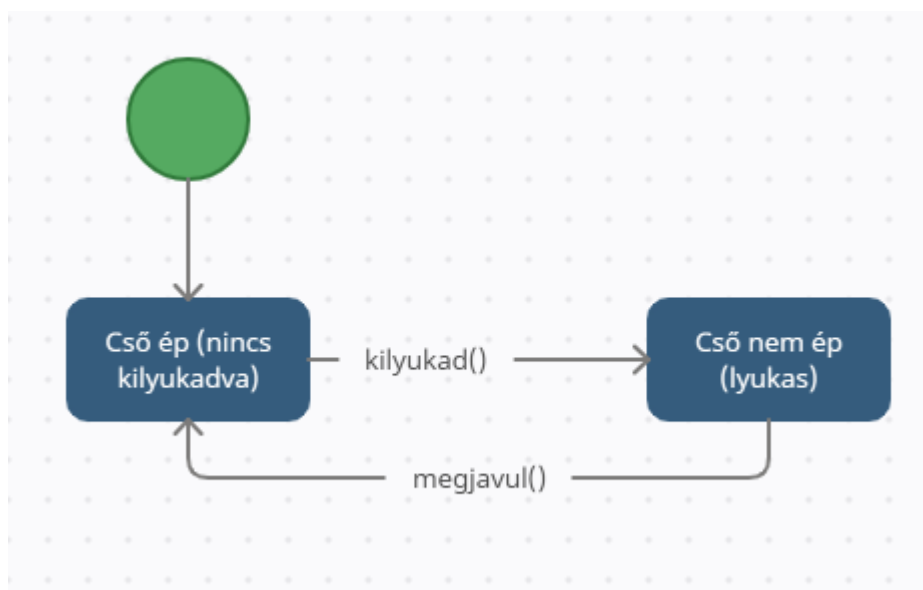
www.websequencediagrams.com

Itt is megintcsak a hely : Cső az a cső amin a szerelő áll, később pedig a hely: Pont a pálya egy pontja.

3.5 State-chartok



Az `elromlik()` metódus a `Pumpa` osztály **private** metódusaként van jelen.



3.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2023.03.16. 14:30	2,5 óra	Kovács Gegő Koleszár	Értekezlet: Class Diagram megtervezése
2023.03.16. 18:30	1,5 óra	Kovács Gegő	Class Diagram megszerkesztése
2023.03.18 12:00	3 óra	Tóth Szautner	Az osztályok leírása.
2023.03.18. 12:00	3 óra	Koleszár	Az osztályok leírásai, class diagram szerkesztés
2023.03.18 14:00	5 óra	Kovács	Az osztályok leírásai
2023.03.18 14:00	4 óra	Gegő	Osztálydiagram szerkesztése és szekvencia diagramok elkezdése
2023.03.18 20:00	3 óra	Tóth	Szekvenciadiagramok ellenőrzése és befejezése.
2023.03.18. 21:00	1,5 óra	Koleszár	Eddigi munka ellenőrzése
2023.03.18 19:00	3 óra	Szautner	Objektum katalógus véglegesítése, osztálydiagram bővítése.
2023.03.19 10:00	4 óra	Gegő Koleszár Tóth Szautner	Értekezlet: Osztálydiagram bővítése, teljes dokumentum ismételt ellenőrzése, hibák átbeszélése, kijavítása.
2023.03.19 11:00	3 óra	Kovács	Osztálydiagram bővítése, teljes dokumentum ismételt ellenőrzése, hibák átbeszélése, kijavítása.

3. Analízis modell kidolgozása

3.1 Objektum katalógus

Csapat: 2 vagy több játékost összefoglaló, a játékosok összpontszámáért és összidejéért, valamint a soron következő (csapaton belüli) játékos kiválasztásért felel. A szerelők, illetve szabotőrök is külön-külön alkotnak egy-egy csapatot.

Szerelő: Olyan játékbeli karakter, azaz belépési pont a játékosok számára a játék (sivatagi vízhálózat) világába, amely azért felel, hogy a csőrendszeren keresztül minél több víz folyjon be a ciszternákba. Feladatai közé tartozik a lyukas csövek megjavítása, új cső lerakása, pumpa lerakása, pumpa átállítása.

Szabotőr: Olyan játékbeli karakter, azaz belépési pont a játékosok számára a játék (sivatagi vízhálózat) világába, amely azért felel, hogy minél több víz folyjon ki a csőhálózatból, valamint minél kevesebb víz folyjon be a ciszternákba. Feladatai közé tartozik a csövek kilyukasztása, pumpa átállítása.

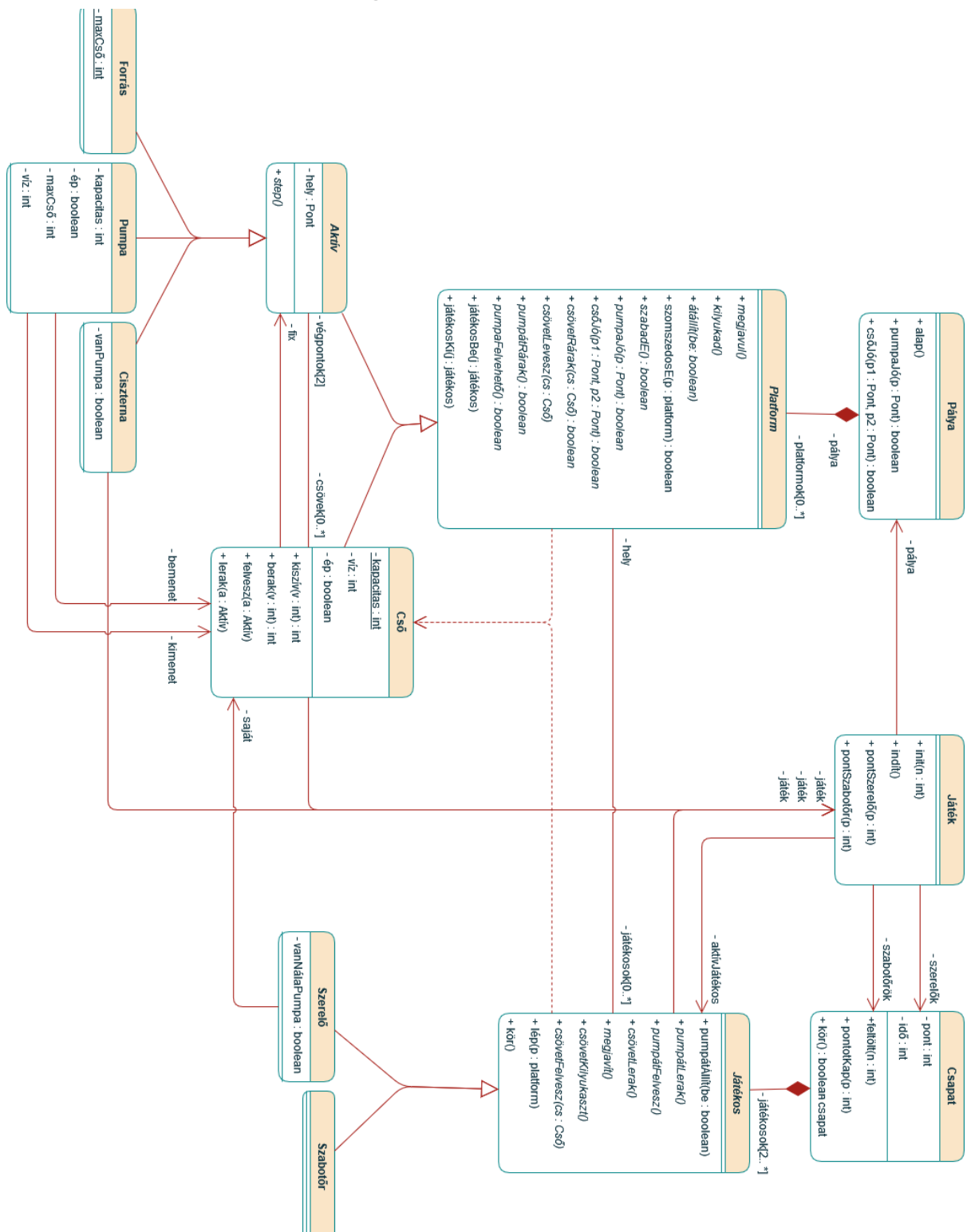
Cső: Olyan objektum a játékban, amelynek a felelősségei közé tartozik, hogy két pumpa, forrás-pumpa, pumpa-ciszterna között vizet tudjon szállítani, amennyiben ép, és vizet engedjen ki a sivatagba, amennyiben lyukas. A csövekhez kapcsolódnak a játékbeli aktív elemek.

Forrás: Olyan objektum, amely azért felel, hogy játékba víz kerülhessen. Ez teszi lehetővé, hogy a rá csatlakoztatott csőben víz legyen. A Forrást nem lehet "tönkretenni", így mindig áramlik be víz a játékba.

Pumpa: Olyan objektum, amely lehetővé teszi több megadott hosszúságú cső összekapcsolását, valamint felelőssége kiterjed arra is, hogy meg lehet határozni melyik két csövet kapcsoljon aktívan össze, azaz melyik csőből melyik másik csőbe kerüljön az egységnyi víz, amennyiben a pumpa üzemel. A pumpa csak 1 bemeneti és 1 kimeneti csövet tud összekötni, azaz több csőből több csőbe nem tud pumpálni.

Ciszterna: Olyan objektum, amely azért felel, hogy játékban lévő víz pontként jóváírható legyen. Hasonlóan működik mint a Forrás objektum, azaz ezt sem lehet "tönkretenni", tehát amennyiben a hozzá kapcsolódó csőben van x egységnyi víz, abban az esetben az x egységnyi víz befolyik a Ciszternába, és pontként jóváíródik a Szerelő csapatnak.

3.2 Statikus struktúra diagramok



3.3 Osztályok leírása

3.3.1 Aktív

- **Felelősség**

A pálya aktív elemeinek az ősosztálya. Biztosítja a léptethetőséget. Absztrakt osztály.

- **Ősosztályok**

Platform

- **Interfészek**

-

- **Attribútumok**

- **Cső[] csövek:** Az aktív elembe csatlakoztatott csövek.
- **Pont hely:** Az aktív elem helye a pályán (grafikus megjelenítéshez).

- **Metódusok**

- **void step():** Az aktív elem minden gametick-ben frissíti az állapotát. **Absztrakt**
- **void csövetLevesz(Cső cs):** A paraméterként kapott cs csövet leveszi magáról.
- **boolean csövetRáak(Cső cs):** A paraméterként kapott cs csövet ráakja magára. Visszatérési értéke, hogy ez megtehető volt-e.

3.3.2 Ciszterna

- **Felelősség**

A játékban egy ciszternát reprezentál. A játék során a ciszterna minden csőből, amely hozzá van kötve kivesz egy egység vizet, ez egyenlő azzal, hogy a ciszternába befolyt az a vízmennyiség. Minden befolyt vízmennyiség után ad a szerelők csapatának egy pontot.

- **Ősosztályok**

Platform → Aktív

- **Interfészek**

-

- **Attribútumok**

- **boolean vanPumpa:** Megadja, hogy a ciszternán van-e pumpa.
- **Játék játék:** A játék példánya.

- **Metódusok**

- **void step():** Minden gametick-ben a ciszterna minden csőből, amely hozzá kapcsolódik kivesz egy egység vizet, majd a befolyt vízmennyiséget pontokban a szerelők csapatának adja.
- **boolean pumpaFelvehető():** Megadja, hogy a ciszternáról fel lehet-e venni pumpát.

3.3.3 Csapat

- **Felelősség**

A játékot játszó egy csapatba tartozó játékosokat tartalmazza. Számon tartja a csapat pontjait és hátralévő összidejét, illetve lépteti, hogy ki a soron következő játékos.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **Játékos[] játékosok:** A csapatba tartozó játékosok.
- **int pont:** A csapat pontjai.
- **int idő:** A csapat összideje mp-ben.

- **Metódusok**

- **void pontotKap(int p):** A csapat a p paraméterben megadott pontot kap, így növelve annyival a pontjainak számát.
- **boolean kör():** Minden a csapatba tartozó játékost felkéri a körére. Folyamatosan csökkenti a csapat összidejét addig, amíg minden, a csapatba tartozó játékos be nem fejezte a körét.
- **void feltölt(int n):** Létrehoz a csapatba n db játékost. Ezeket hozzá is adja a csapathoz.

3.3.4 Cső

- **Felelősség**

A cső a játékban két aktív elem összekötésére és víz tárolására szolgál.

- **Ősosztályok**

Platform

- **Interfészek**

-

- **Attribútumok**

- **static int kapacitás:** A cső kapacitása víz egységben mérve. Ennél több víz nem kerülhet bele.
- **int víz:** Tárolja, hogy mennyi víz van éppen a csőben.
- **boolean ép:** A cső állapotát tárolja, ha a cső nem ép, a benne lévő víz elfolyik.
- **Játék játék:** A játék példánya.
- **Aktív fix:** Az az oldal, amelyet éppen nem mozgatunk.
- **Aktív[] végpontok:** A cső végpontjai.

- **Metódusok**

- **int kiszív(int v):** A csőből v számú vizet próbál kiszívni. Visszatérési értéke a vízmennyiség, amennyit ki lehet szívni (pl.: víz = 3, v = 4 → return = 3), ennyivel kevesebb lesz a csőben a víz.
- **int berak(int v):** A csőbe v számú vizet próbál berakni. Ha a cső túllépné így a kapacitását, akkor kapacitás mennyiségű víz lesz benne. Visszatérési értéke (v - az a vízmennyiség amennyi a csőbe bele tudott menni).
- **void felvesz (Aktív a):** Egy cső felvétele. Az argumentum a cső oldalát adja meg.
- **void lerak (Aktív a):** Egy cső lerakása. Az argumentum azt adja meg, hogy melyik aktív elemre rakjuk rá a csövet.
- **boolean pumpátRáarak():** Egy cső közepére egy pumpát megpróbál elhelyezni. Visszaadja, hogy ez sikeres volt-e.
- **void kilyukad():** Cső kilyukasztása.
- **void megjavul():** Cső megjavítása.

3.3.5 Forrás

- **Felelősség**

A forrás szolgáltatja a vizet a hozzá csatlakoztatott csövekbe.

- **Össztályok**

Platform → Aktív

- **Interfészek**

-

- **Attribútumok**

- **static int maxCső:** A forrásba kapcsolódó csövek maximális száma.

- **Metódusok**

- **void step():** Minden gametick-ben a forrás a hozzá csatlakoztatott csövekbe egy egység vizet rak. Bizonyos időközönként a forráson cső keletkezik. Ezeknek a csöveknek az egyik vége a forrásba, a másik végük a semmibe csatlakozik.
- **boolean csövetRáarak(Cső cs):** A paraméterként kapott cs csövet rárakja magára. Visszatérési értéke, hogy ez megtehető volt-e.

3.3.6 Játék

- **Felelősség**

A játék lebonyolításáért felelős osztály.

- **Össztályok**

-

- **Interfészek**

-

- **Attribútumok**
 - **Játékos aktívJátékos:** Az éppen soron következő játékos.
 - **Csapat szabotőrök:** A szabotőrök alkotta csapat.
 - **Csapat szerelők:** A szerelők alkotta csapat.
 - **Pálya pálya:** A jelenlegi pálya. Ez alapesetben az alapértelmezett pálya.
- **Metódusok**
 - **void indít():** Ez a metódus indítja el a játékot, illetve tölti be a pályát és kezdi meg a játékot.
 - **void pontSzerelo(int p):** A szerelő csapat p pontot kap.
 - **void pontSzabotor(int p):** A szabotőr csapat p pontot kap.
 - **void init(int n):** A játék kezdeti állapotát inicializáló függvény. Az n paraméter megadja a játékosok számát.

3.3.7 Játékos

- **Felelősség**

A játékosok karaktereit reprezentáló absztrakt őszosztály. Biztosítja a karakterek alapvető akcióit.

- **Őszosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **Platform hely:** Az a Platform, amin a játékos áll.
- **Játék játék:** A játék példánya.

- **Metódusok**

- **void pumpátÁllít(boolean be) :** A játékos beállítja a pumpát, amin éppen áll. Ha nem azon áll, akkor nem történik semmi.
- **void pumpátLerak():** A játékos megpróbálja a nála lévő pumpát lerakni. **Absztrakt**
- **void pumpátFelvesz():** A játékos megpróbálja a platformról amin áll felvenni a pumpát. **Absztrakt**
- **void csövetLerak():** A játékos megpróbálja lerakni a nála lévő csövet. **Absztrakt**
- **void csövetFelvesz(Cső cs):** A játékos megpróbálja felvenni a cs csövet. **Absztrakt**
- **void lép(platform p):** A játékos megpróbál a p platformra lépni. Ha ez nem sikerül, akkor a saját platformján marad.
- **void kör():** A játékost felkéri a csapata a körre, ilyenkor kap egy előre meghatározott időt a játékos, ami alatt cselekedhet akármennyit. Az idő lejártával a játék automatikusan elveszi az irányítást a játékostól. A játékos az idő lejártá előtt is befejezheti a körét.
- **void megjavít():** A játékos megpróbálja megjavítani azt a Platformot, amin éppen tartózkodik. **Absztrakt**
- **void csövetKilyukaszt():** A játékos megpróbálja kilyukasztani azt a Platformot, amin éppen áll.

3.3.8 Pálya

- **Felelősség**

A játék pályáját reprezentálja. Ez az osztály tárolja a pálya elemeit és ellenőrzi, hogy a pályán kért módosítás (pl. pumpa lerakása) legális-e.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **platform[] platformok:** A pálya platformjai.

- **Metódusok**

- **boolean pumpaJó(Pont p):** Visszaadja, hogy egy pumpát az adott p pontra el lehet-e helyezni.
- **boolean csőJó(Pont p1, Pont p2):** Visszaadja, hogy egy, a p1 és p2 pontokat összekötő csövet le lehet-e rakni.
- **void alap():** Beállítja az alapértelmezett pályát (ez magába foglalja a pálya elemeinek legenerálását/létrehozását).

3.3.9 Platform

- **Felelősség**

A játék mezőit reprezentáló absztrakt őosztály. Biztosítja a mezők számára az alapvető viselkedéseket.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **Játékos[] játékosok:** A platformon tartózkodó játékosok listája.
- **Pálya pálya:** A pálya amelybe a platform be van töltve.

- **Metódusok**

- **void megjavul():** A platform újra rendesen működik. **Absztrakt**
- **void kilyukad():** A platformot tönkreteszik. **Absztrakt**
- **void átállít():** A platformot átállítják. **Absztrakt**
- **boolean szomszedosE(Platform p):** Visszaadja, hogy két platform szomszédos-e.
- **boolean szabadE():** Visszaadja, hogy az adott platform szabad-e.
- **boolean pumpaJó(Pont p):** Visszaadja, hogy a platform és a p pontra helyezett pumpa közt nincs-e konfliktus.
- **boolean csőJó(Pont p1, Pont p2):** Visszaadja, hogy a platform és a p1 és p2 pontokat összekötő cső között nincs-e konfliktus.
- **boolean pumpaFelvehető() :** Visszaadja, hogy a pumpát felveheti-e egy játékos. **Absztrakt**
- **void pumpátRáarak():** A pumpát elhelyezi egy cső közepén. Ehhez a csövet középen ketté kell vágni. **Absztrakt**
- **void játékosBe(Játékos j):** A j játékost hozzáadja a platform saját magához.
- **void játékosKi(Játékos j):** A j játékost a platform eltávolítja saját magáról.
- **boolean csövetRáarak(Cső cs):** A platformra megpróbálunk felcsatlakoztatni egy csövet. **Absztrakt**
- **void csövetLevesz(Cső cs):** A cs csövet levesszük a platformról. **Absztrakt**

3.3.10 **Pumpa**

- **Felelősség**

A pumpa egy olyan aktív elem, amihez a hozzá csatlakoztatott csövek közül - ha a pumpa ép - az egyik kiválasztott csőből pumpálja pontosan egy másik kiválasztott csőbe a vizet. Ezt úgy teszi, hogy a bejövő csőből kiveszi, a kimenő csőbe pedig berak egy egység vizet gametick-enként. Emellett minden gametick-ben megvan az esélye, hogy a pumpa elromlik.

- **Ősosztályok**

Platform → Aktív

- **Interfészek**

-

- **Attribútumok**

- **boolean ép:** A pumpa állapotát tárolja, ha a pumpa nem ép, tehát elromlott, akkor nem pumpálja a vizet.

- **int víz:** A pumpa belső tartályában lévő víz mennyisége.
- **int kapacitás:** A pumpa tartályának kapacitása.
- **Cső kimenet:** A pumpa ebbe a csőbe pumpálja a vizet.
- **Cső bemenet:** A pumpa ebből a csőből pumpálja a vizet.
- **int maxCső:** A pumpára maximálisan csatlakoztatható csövek száma.
- **Metódusok**
 - **void step():** A pumpa minden gametick-ben a bejövő csőből egy egység vizet megpróbál kivenni, majd a kivett vízmennyiséget berakja a kimenő csőbe. Ebben a metódusban történik a pumpa véletlenszerű elromlása is.
 - **void átállít(boolean be):** A függvény a paraméterként kapott kibe alapján (kimenet állítás: false, bemenet állítás: true) beállítja a kimeneti vagy bemeneti csövet a “sorban következőre” (a Cső[] csövek arrayban következő-re).
 - **void megjavul():** A pumpán amikor meghívják ezt a függvényt, akkor ezek után ismét működni fog, mindaddig, amíg ismét el nem romlik.
 - **boolean csövetRáak(Cső cs):** A paraméterként kapott cs csövet ráakja magára. Visszatérési értéke, hogy ez megtehető volt-e.

3.3.11 Szerelő

- **Felelősség**

A játékban egy szerelő karakter. Ő tudja a csöveket és pumpákat javítani, felvenni és lerakni.

- **Ősosztályok**

Játékos

- **Interfészek**

-

- **Attribútumok**

- **boolean vanNálaPumpa():** Ha a szerelőnél van pumpa igaz, ha nincs hamis.
- **Cső saját:** A szerelőnél lévő cső.

- **Metódusok**

- **void pumpátLerak():** A szerelő megpróbálja lerakni a nála lévő pumpát a platformra, amin áll.
- **void pumpátFelvesz():** A szerelő megpróbál felvenni egy pumpát arról a platformról, amin áll.
- **void csövetFelvesz(Cső cs):** A szerelő megpróbálja felvenni a cs csövet.
- **void csövetLerak():** A szerelő megpróbálja a csövet lerakni a platformra, amin áll.
- **void megjavít():** A szerelő megpróbálja megjavítani a platformot, amin áll.

3.3.12 Szabotőr

- **Felelősség**

A játékban egy szabotőr karakter. Ő tudja a csöveket kilyukasztani és a pumpákat átállítani.

- **Össztályok**

Játékos

- **Interfészek**

-

- **Attribútumok**

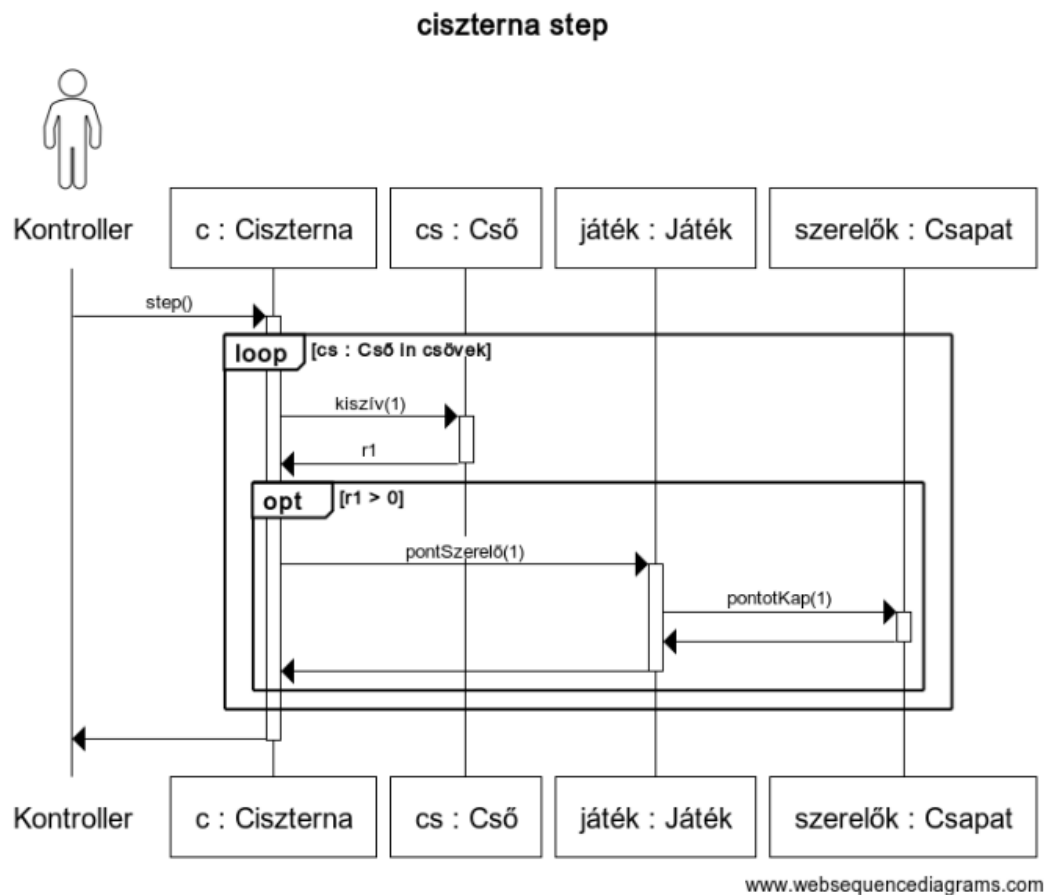
-

- **Metódusok**

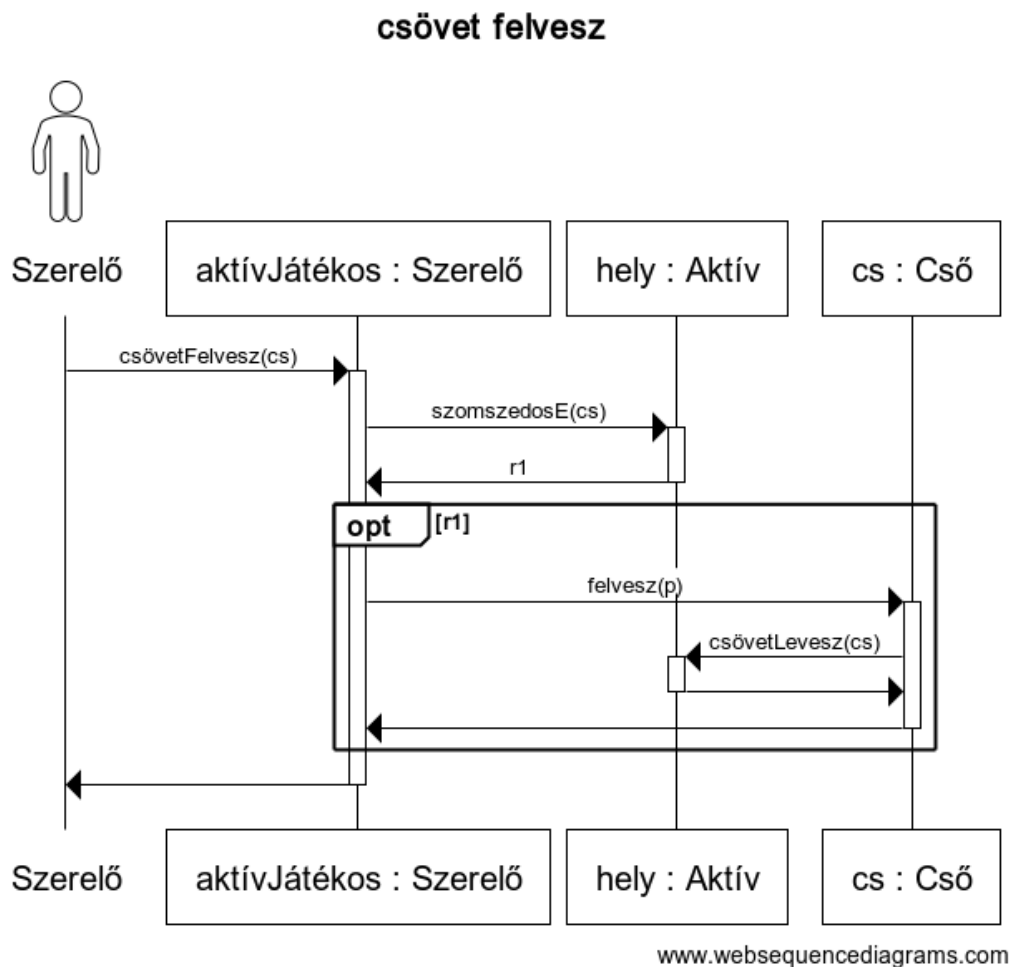
- **void csövetKilyukaszt():** A szabotőr megpróbálja kilyukasztani a csövet amin áll.

3.4 Szekvencia diagramok

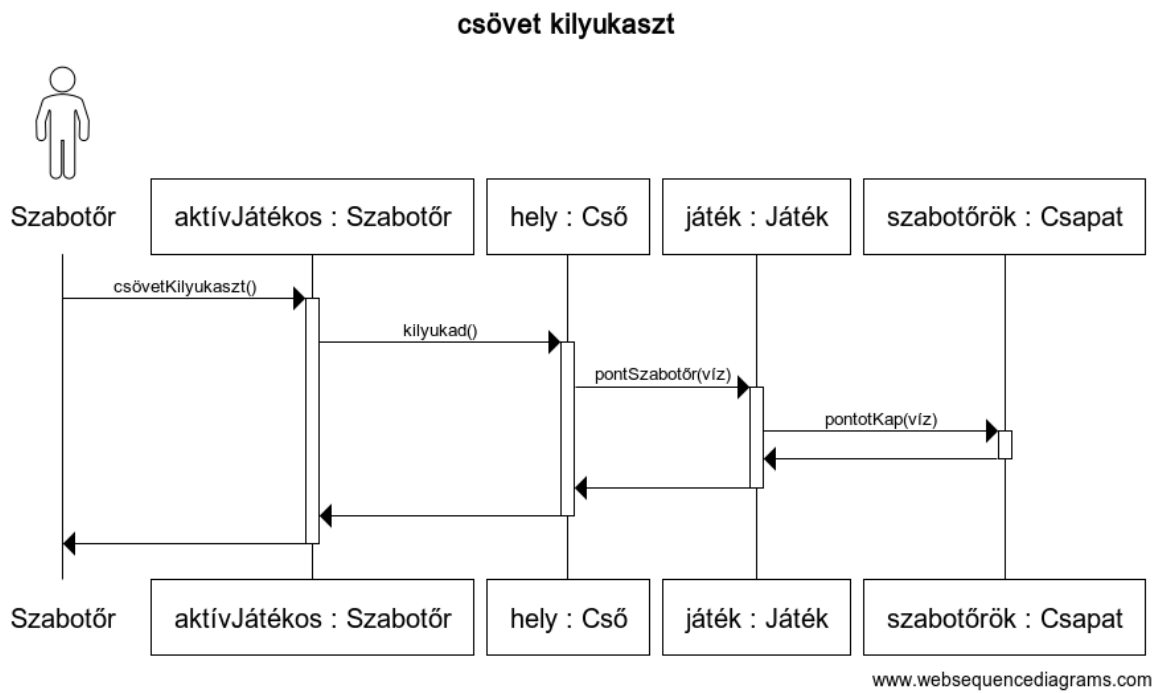
3.4.1 Ciszterna step



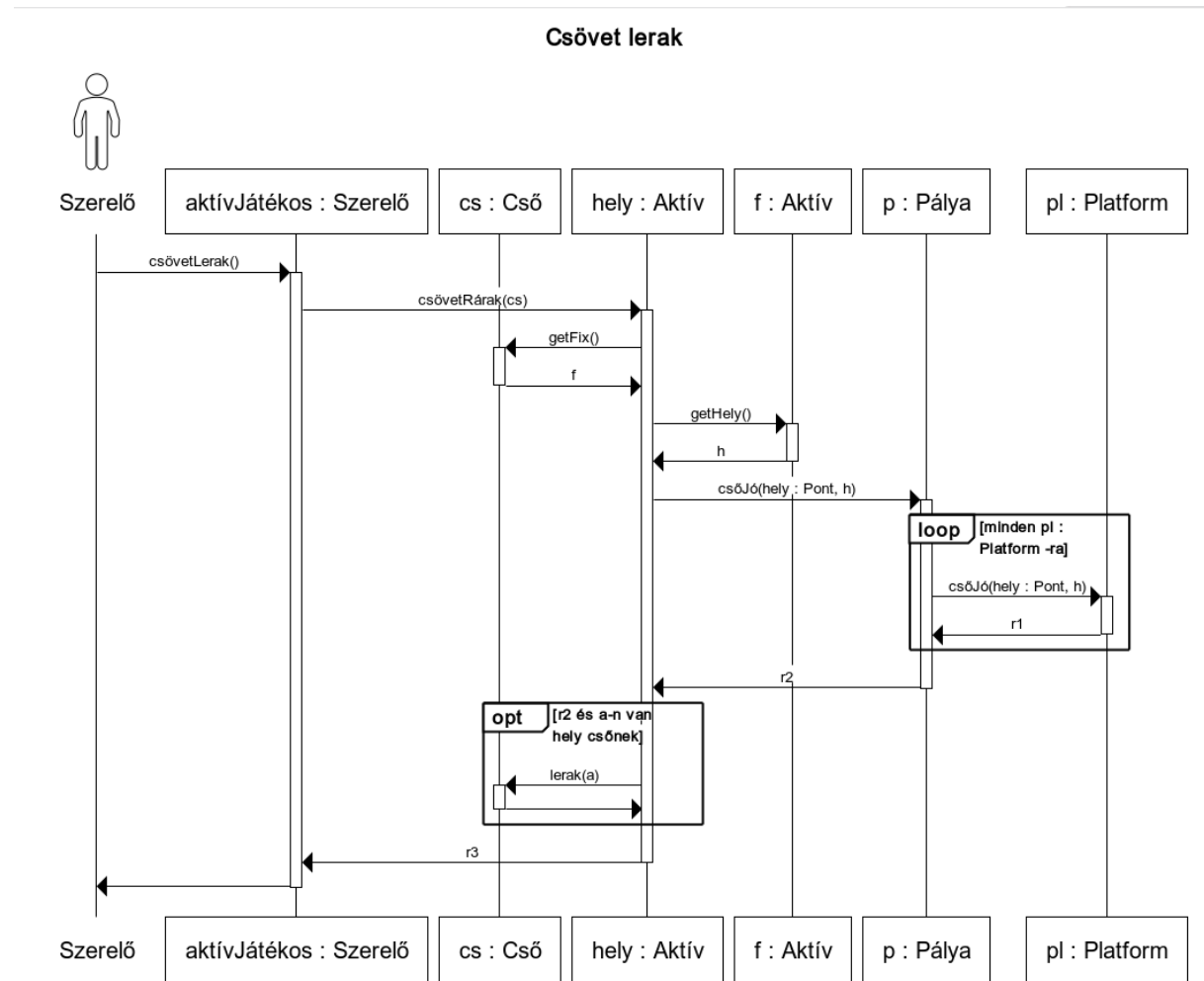
3.4.2 Csövet felvesz



3.4.3 Csövet kilyukaszt

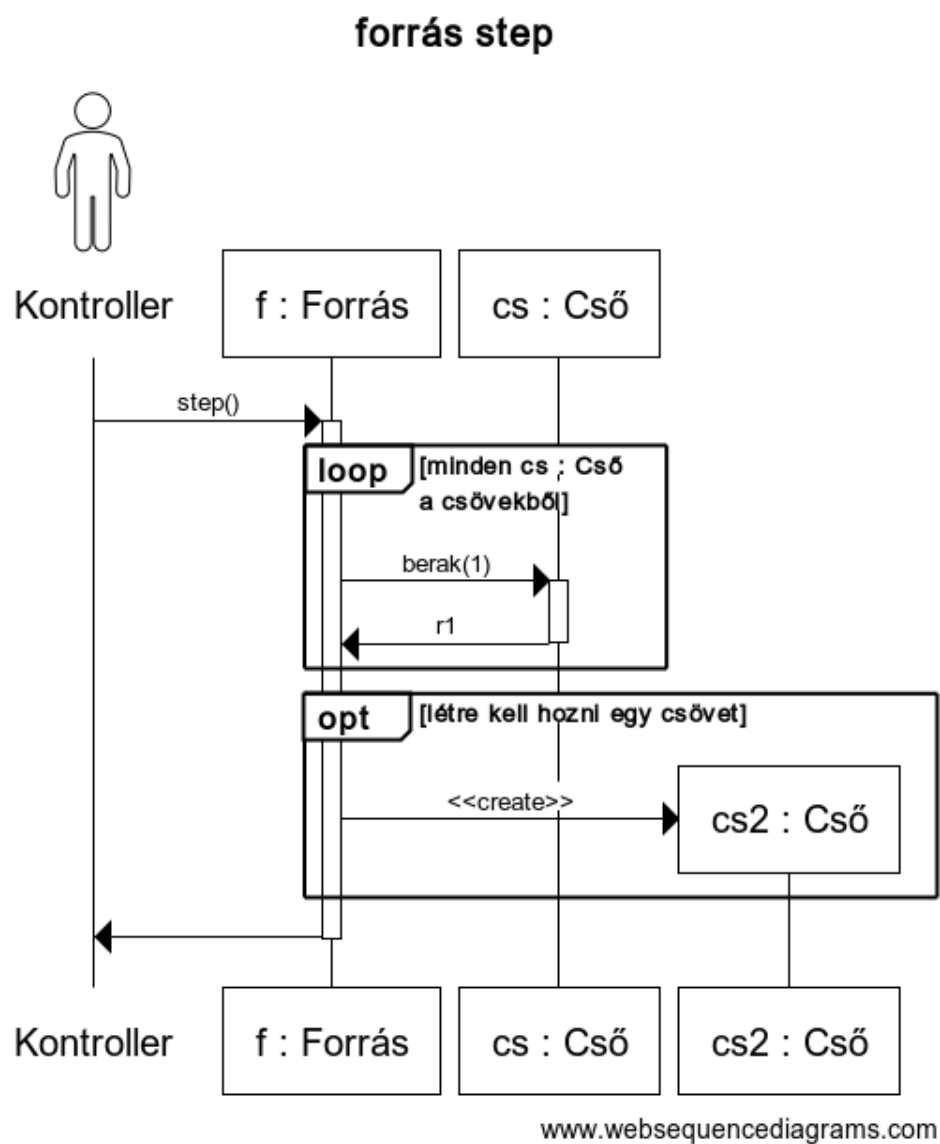


3.4.4 Csövet lerak

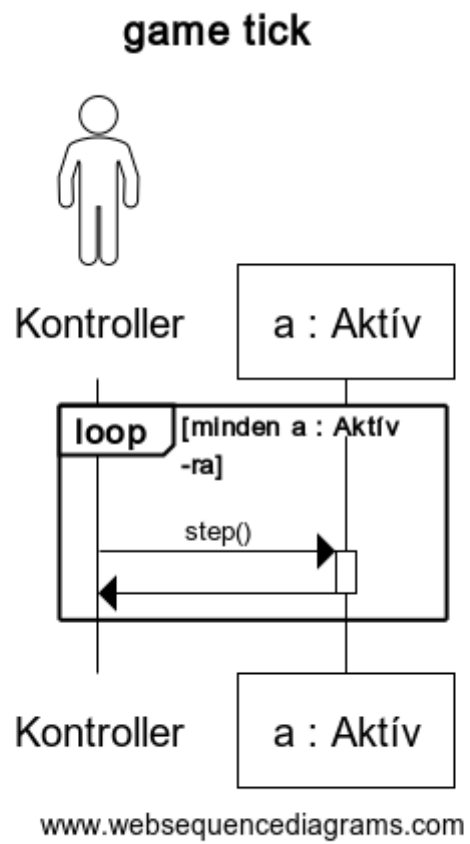


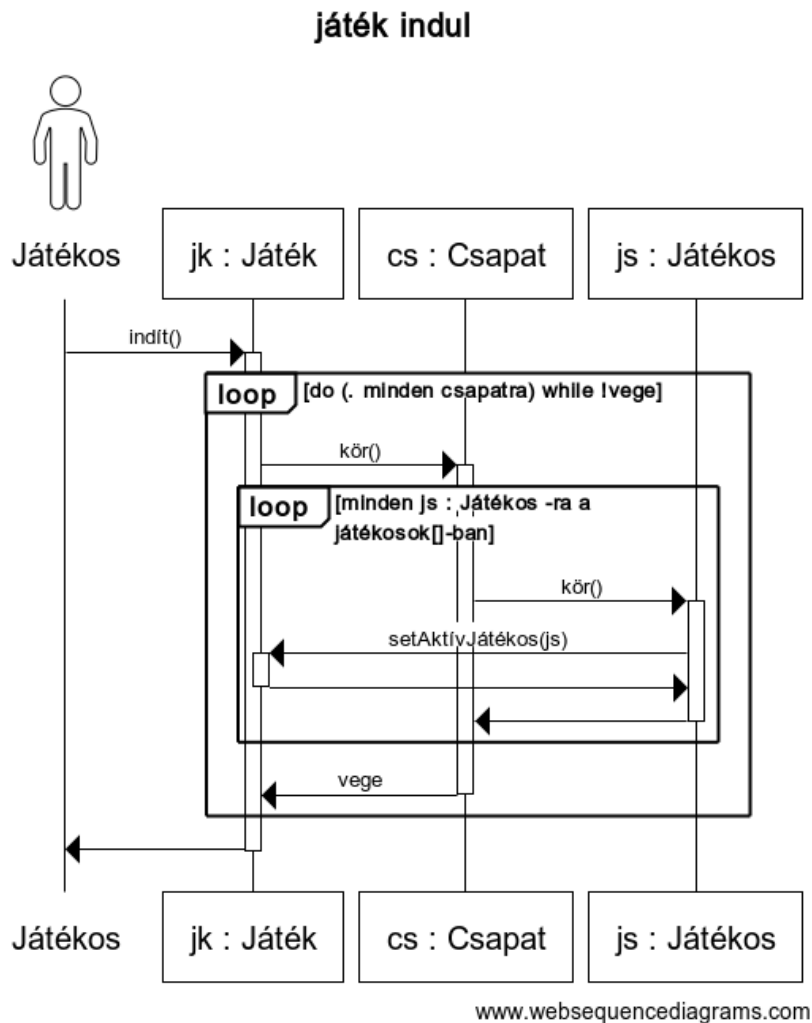
A csövet lerak szekvencia diagramban a hely : Aktív azt az aktív elemet reprezentálja amelyen a szerelő áll és ezáltal ismeri. A hely: Pont viszont a pálya egy megadott pontját jelzi (ez a grafikus megjelenítéshez kell).

3.4.5 Forrás step

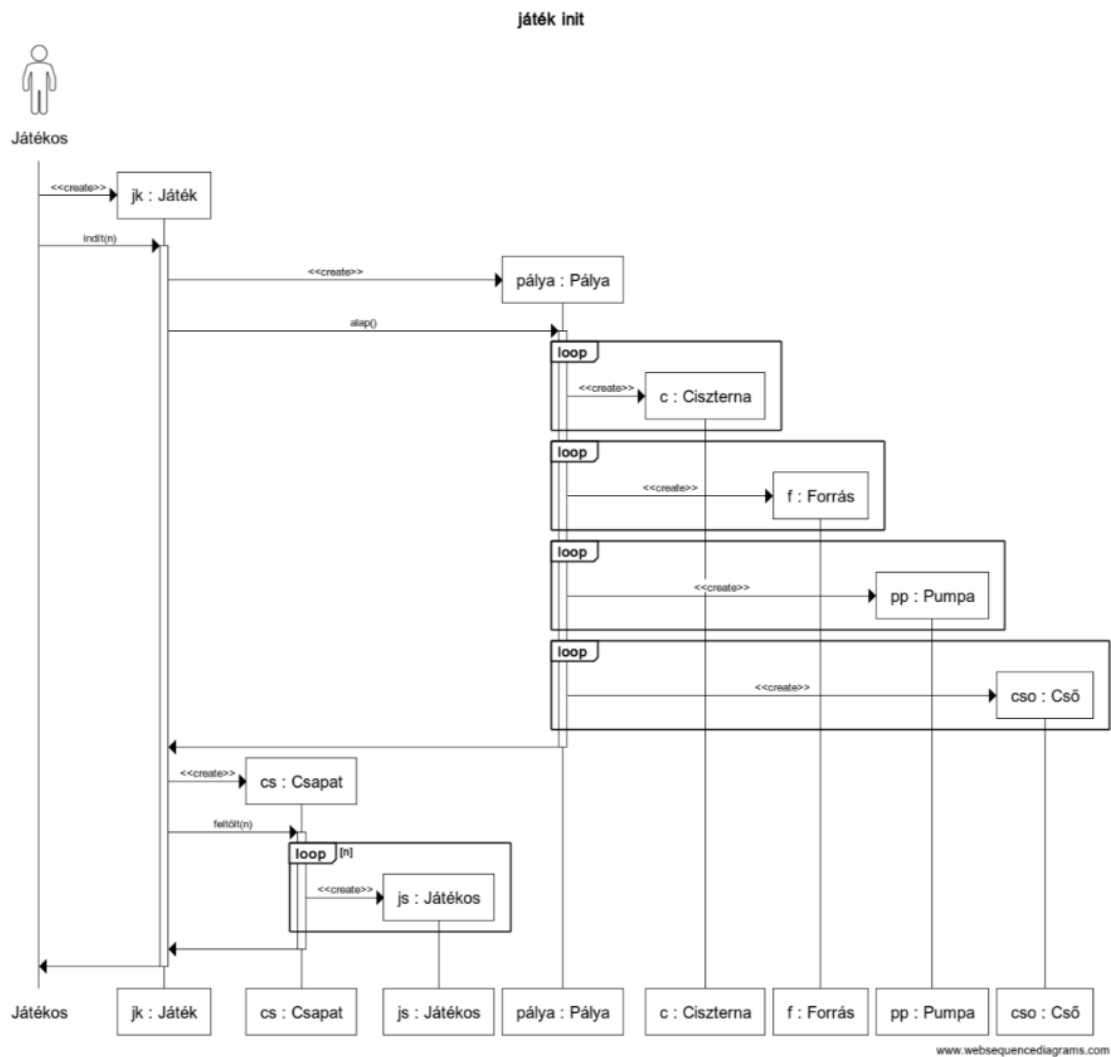


3.4.6 Game tick



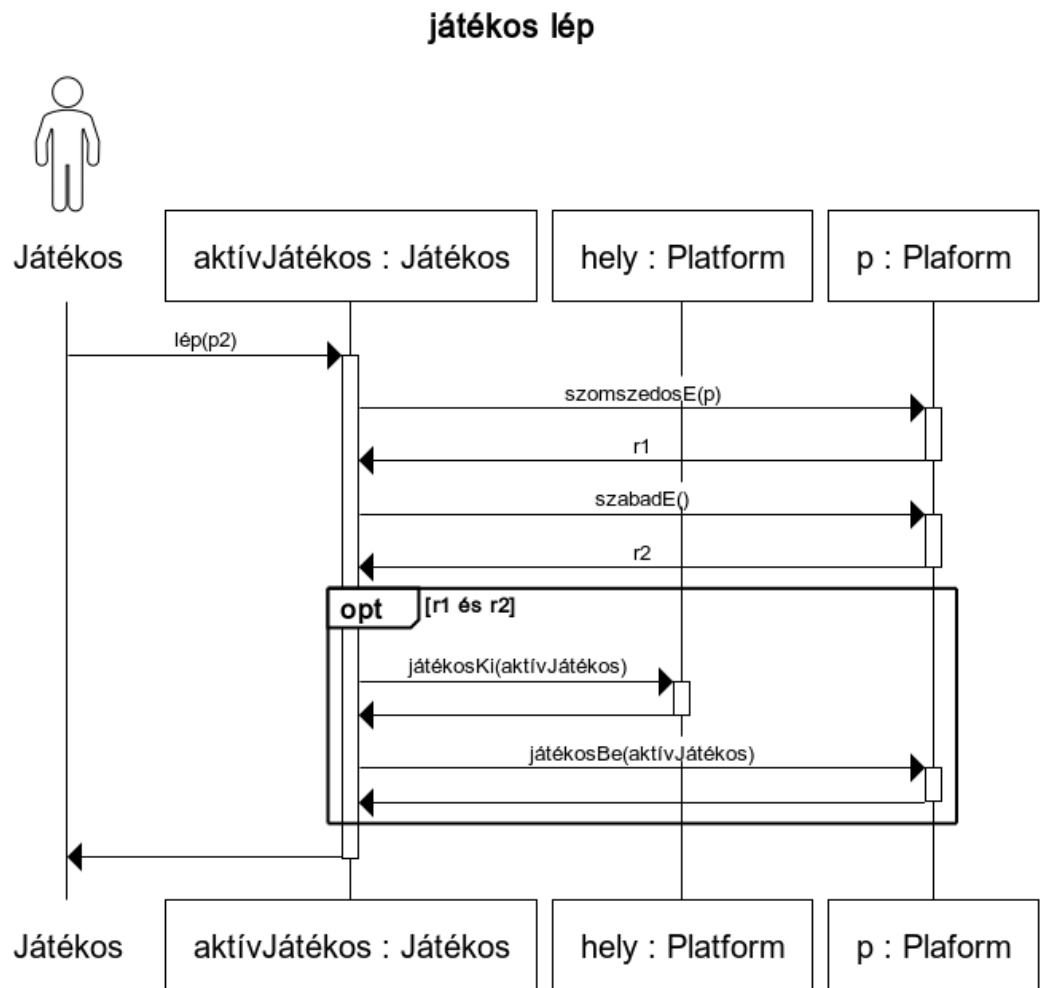
3.4.7 Játék indul

3.4.8 Játék init

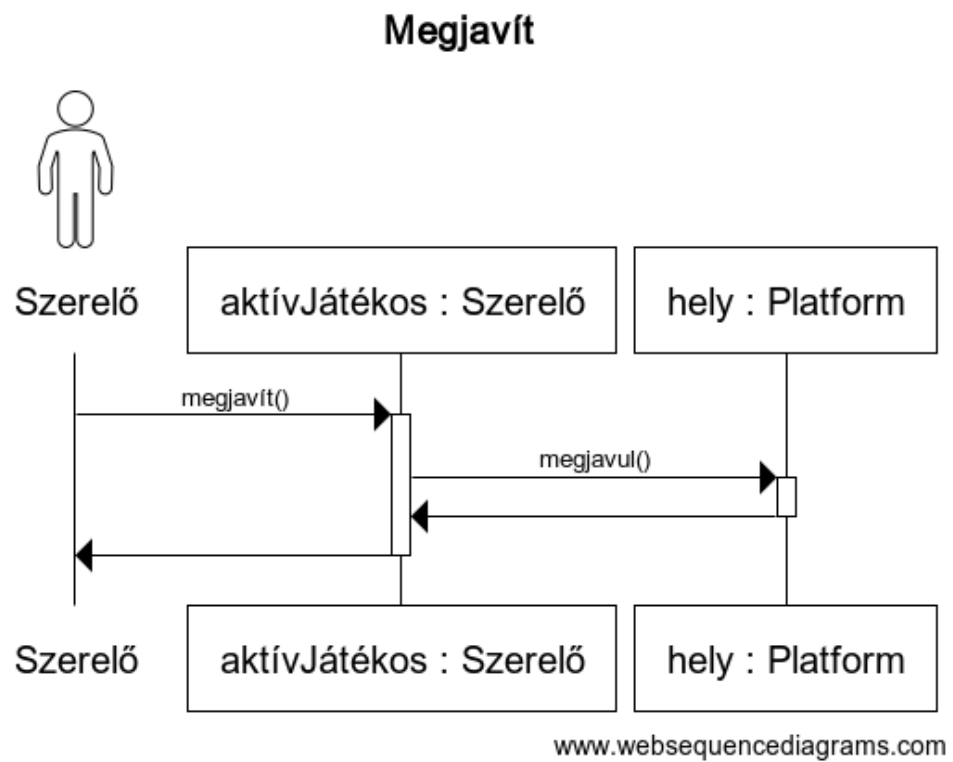


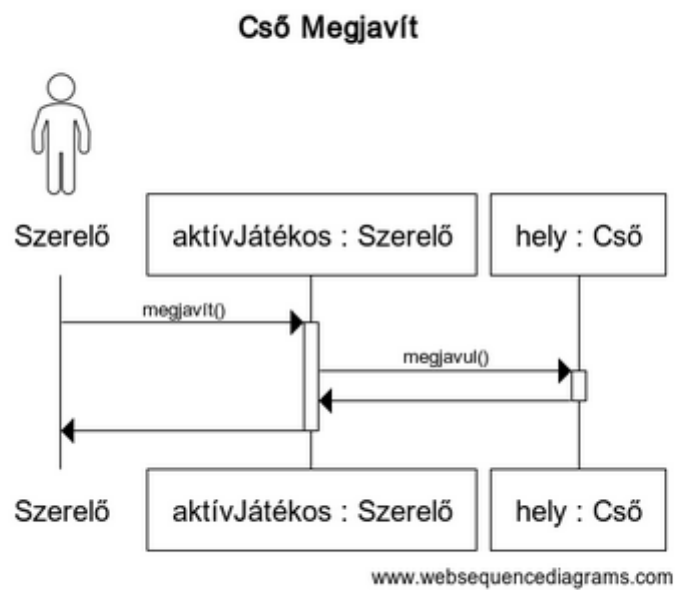
Itt a pálya loopjaiban mindig a megfelelő számú elem kerül létrehozásra. (ezt előre nem tudjuk, hogy mennyi így a loopba nem írtunk semmit)

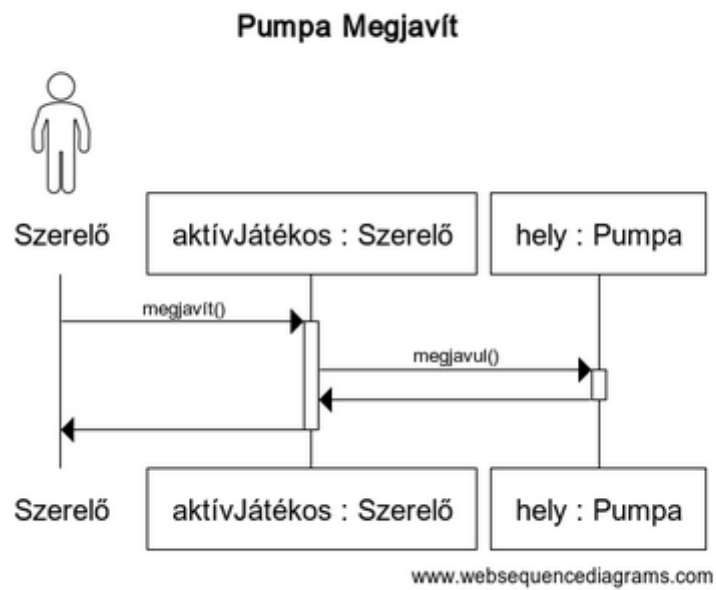
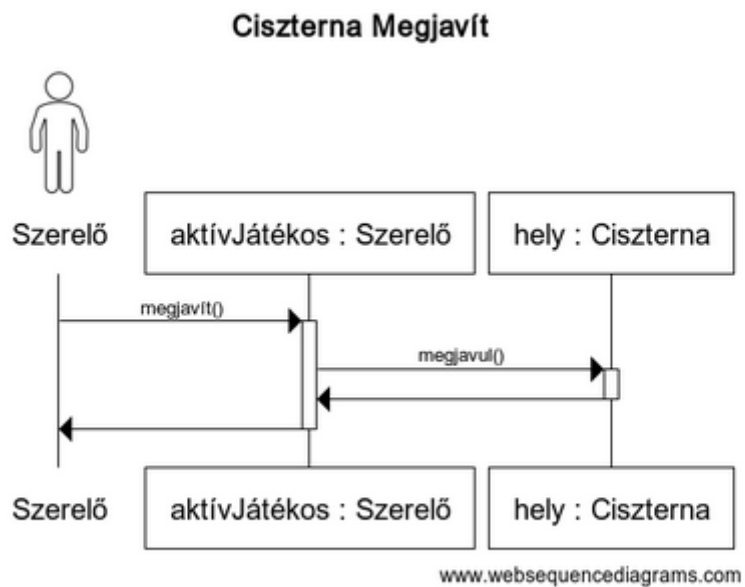
3.4.9 Játékos lép

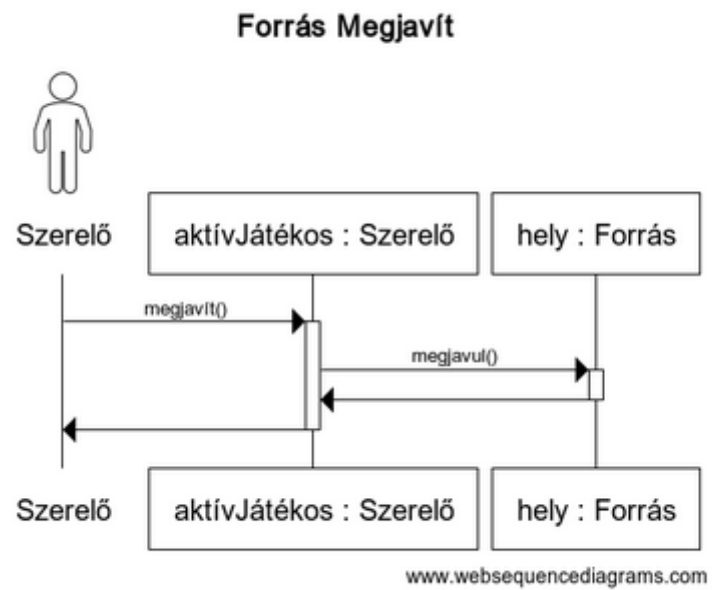


www.websequencediagrams.com

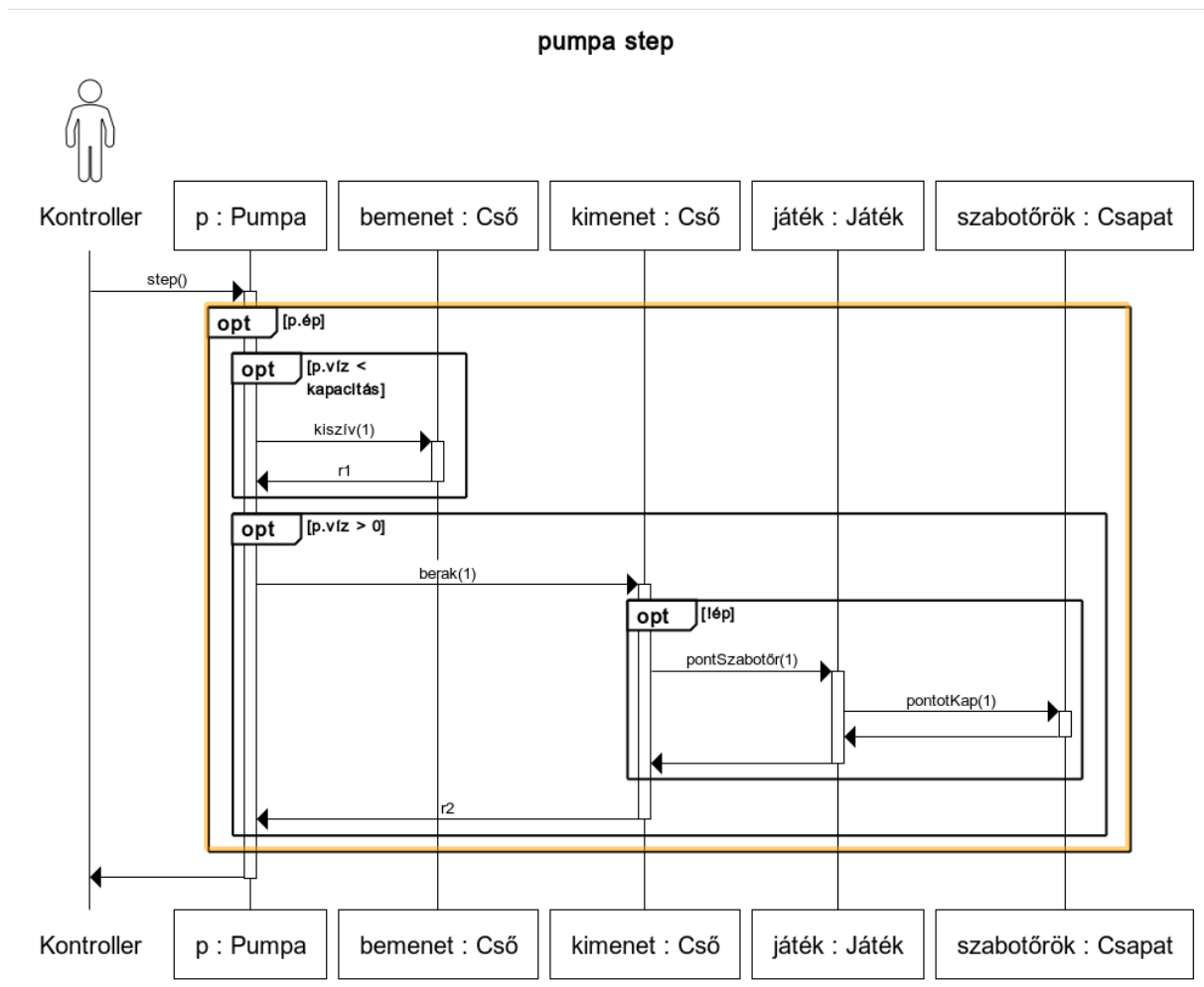
3.4.10 Megjavít

3.4.11 Cső Megjavít

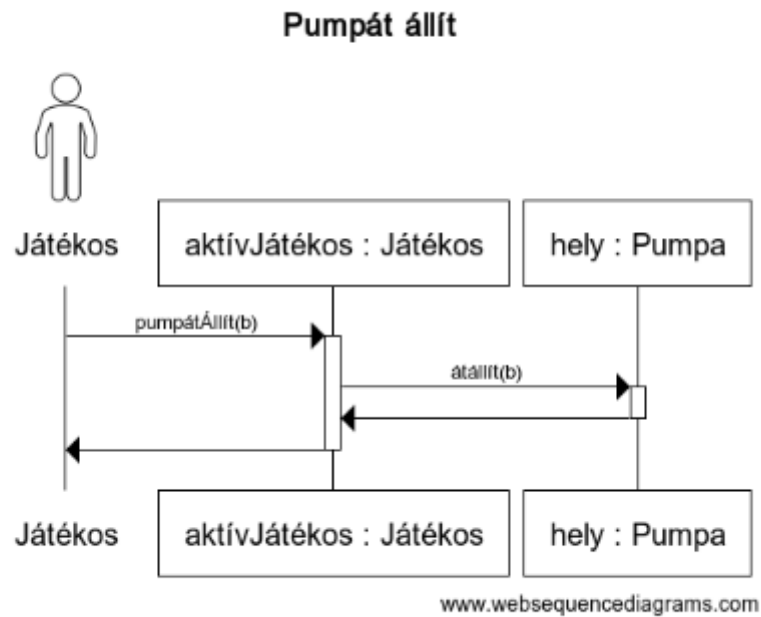
3.4.12 *Pumpa Megjavít***3.4.13 *Ciszterna Megjavít***

3.4.14 Forrás Megjavít

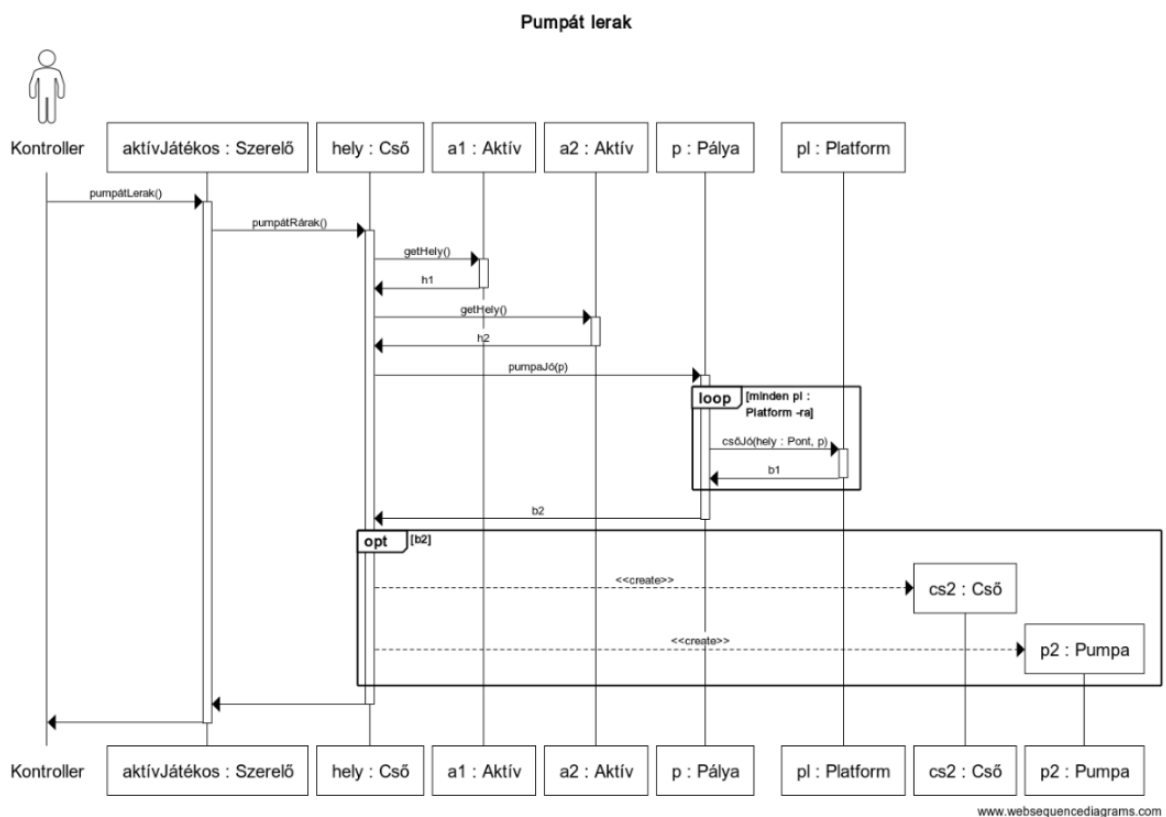
3.4.15 *Pumpa step*



3.4.16 Pumpát állít



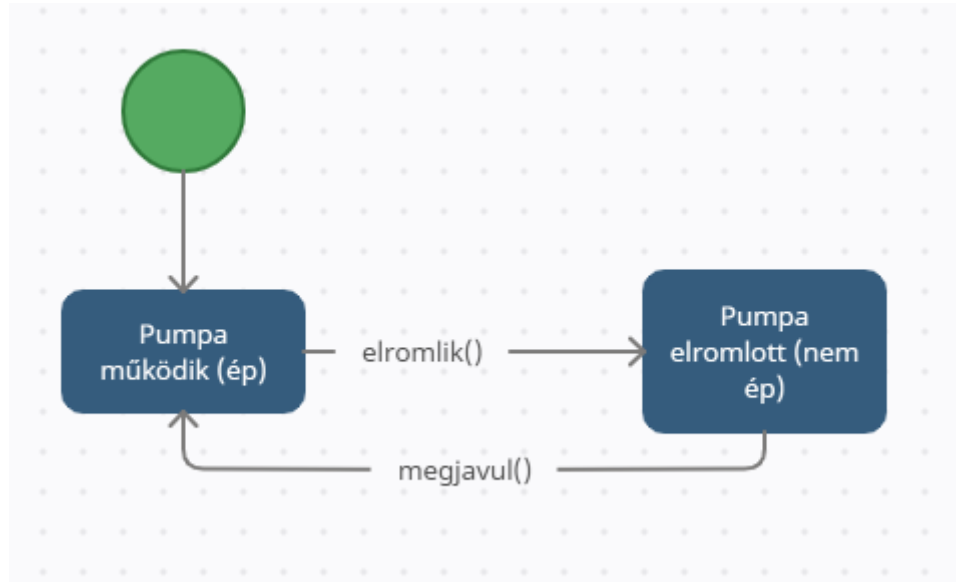
3.4.17 Pumpát lerak



Itt is megintcsak a hely : Cső az a cső amin a szerelő áll, később pedig a hely: Pont a pálya egy pontja.

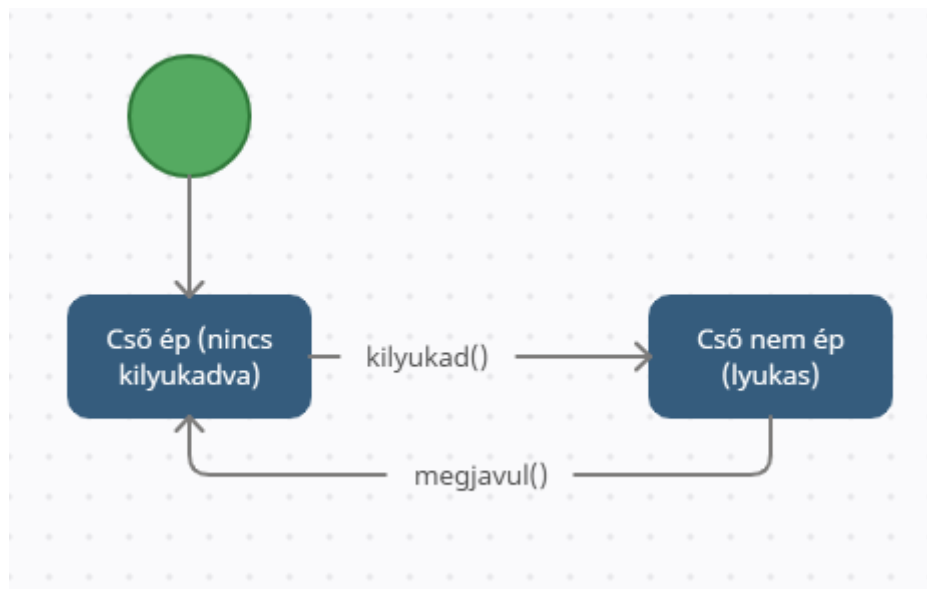
3.5 State-chartok

3.5.1 Pumpa statechart



Az `elromlik()` metódus a `Pumpa` osztály **private** metódusaként van jelen.

3.5.2 Cső statechart



3.6 Napló

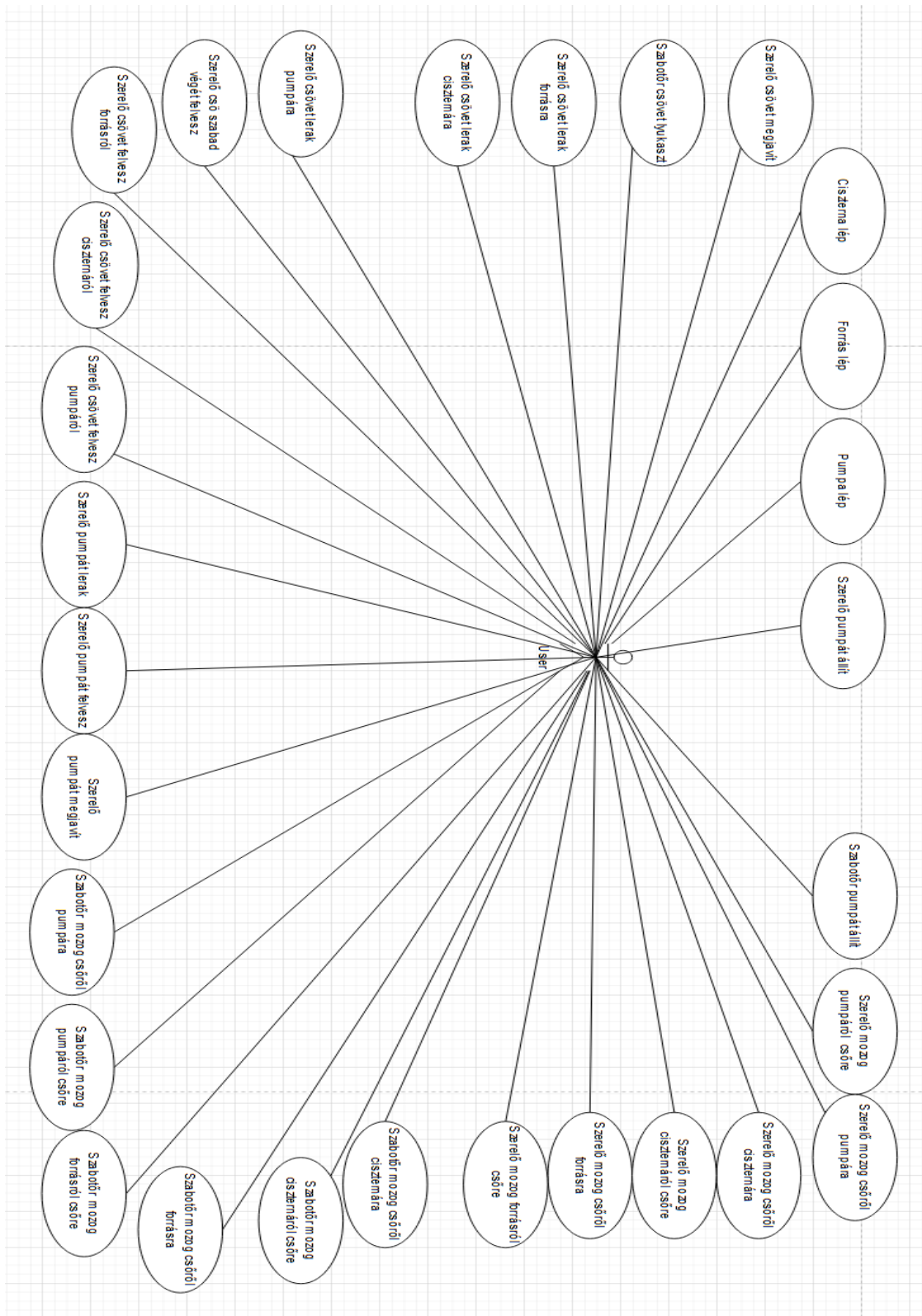
Kezdet	Időtartam	Résztevők	Leírás
2023.03.16. 14:30	2,5 óra	Kovács Gegő Koleszár	Értekezlet: Class Diagram megtervezése
2023.03.16. 18:30	1,5 óra	Kovács Gegő	Class Diagram megszerkesztése
2023.03.18 12:00	3 óra	Tóth Szautner	Az osztályok leírása.
2023.03.18. 12:00	3 óra	Koleszár	Az osztályok leírásai, class diagram szerkesztés
2023.03.18 14:00	5 óra	Kovács	Az osztályok leírásai
2023.03.18 14:00	4 óra	Gegő	Osztálydiagram szerkesztése és szekvencia diagramok elkezdése
2023.03.18 20:00	3 óra	Tóth	Szekvenciadiagramok ellenőrzése és befejezése.
2023.03.18. 21:00	1,5 óra	Koleszár	Eddigi munka ellenőrzése
2023.03.18 19:00	3 óra	Szautner	Objektum katalógus véglegesítése, osztálydiagram bővítése.
2023.03.19 10:00	4 óra	Gegő Koleszár Tóth Szautner	Értekezlet: Osztálydiagram bővítése, teljes dokumentum ismételt ellenőrzése, hibák átbeszélése, kijavítása.
2023.03.19 11:00	3 óra	Kovács	Osztálydiagram bővítése, teljes dokumentum ismételt ellenőrzése, hibák átbeszélése, kijavítása.
2023.03.25 11:00	1 óra	Gegő Kovács Szautner	Konzultáción megbeszélte hibák javítása, szekvenciák bővítése, dokumentum módosítása.

2023.03.26 11:00	1 óra	Gegő Szautner Koleszár Tóth	Teljes dokumentum átnézése, közben kisebb hibák javítása.
------------------	-------	--------------------------------------	--

5. Szkeleton tervezése

5.1 A szkeleton modell valóságos use-case-e

5.1.1 Use-case diagram



5.1.2 Use-case leírások

Use-case neve	Ciszterna lép
Rövid leírás	A vízhálózatban a Ciszterna elem kitol magából egy egység vizet.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. Amennyiben hozzá van kapcsolva egy cső, abban az esetben a csőbe pumpálja a vizet. 2. Amennyiben nincs hozzá kapcsolva cső, akkor az az egység víz nem kerül bele a rendszerbe.

Use-case neve	Forrás lép
Rövid leírás	A vízhálózatban a Forrás elem befogad magába egy egység vizet.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. Amennyiben a Forrás elemhez hozzá van kapcsolva a cső, és a csőben van víz, akkor "kiveszi" a csőből a vizet, és pontként jóváírja a Szerelő csapatnak.

Use-case neve	Pumpa lép
Rövid leírás	A vízhálózatban a Pumpa elem átmozgat magán keresztül egy egység vizet a hozzá csatlakoztatott csőből, a szintén hozzá csatlakoztatott cső elembe. A szívócső és a kimeneti cső a pumpában van egyértelműen beállítva.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A pumpa a kiszívja a hozzá kapcsolt bemeneti csőből a vizet és a kimeneti csőbe beletesz egy egység vizet. 2. Amennyiben a pumpához kapcsolt csőben nincs víz, akkor nem szív ki vizet a bemeneti csőből, és nem tesz vizet a kimeneti csőbe

Use-case neve	Szerelő pumpát állít
Rövid leírás	A játék folyamata során a Szerelő játékos átállítja azt a pumpát, amin tartózkodik.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos átállítja a pumpát, ezzel megváltoztatja a pumpa bementi és kimeneti csöveit.

Use-case neve	Szabotőr pumpát állít
Rövid leírás	A játék folyamata során a Szabotőr játékos átállítja azt a pumpát, amin tartózkodik.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szabotőr játékos átállítja a pumpát, ezzel megváltoztatja a pumpa bementi és kimeneti csöveit.

Use-case neve	Szerelő mozog pumpáról csőre
Rövid leírás	A Szerelő játékos megpróbál mozogni egy pumpáról egy csőre.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos megváltoztatja helyét, azaz egy pumpáról egy csőre lép. 2. A Szerelő játékos nem tudja megváltoztatni a helyét, mivel a "cél" csövön, ahova lépni szeretne, már áll másik játékos. Ekkor helyét megtartja a Szerelő játékos, azaz nem lép át a csőre, és így a pumpán marad.

Use-case neve	Szerelő mozog csőről pumpára
Rövid leírás	A Szerelő játékos átmozog a vízhálózat egy cső eleméről egy pumpa elemére.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos átmozog egy csőről egy pumpára, azaz megváltoztatja helyét.

Use-case neve	Szerelő mozog csőről ciszternára
Rövid leírás	A Szerelő játékos átmozog a vízhálózat egy cső eleméről egy ciszterna elemére.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos átmozog egy csőről egy ciszternára, azaz megváltoztatja helyét.

Use-case neve	Szerelő mozog ciszternáról csőre
Rövid leírás	A Szerelő játékos megpróbál mozogni a vízhálózat egy ciszterna eleméről egy cső elemére.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos megváltoztatja helyét, azaz egy ciszternáról egy csőre lépett. 2. A Szerelő játékos nem tudja megváltoztatni a helyét, mivel a "cél" csövön, ahova lépni szeretne, már áll másik játékos. Ekkor helyét megtartja a Szerelő játékos, azaz nem lép át a csőre, és így a ciszternán marad.

Use-case neve	Szerelő mozog csőről forrásra
Rövid leírás	A Szerelő játékos átmozog a vízhálózat egy cső eleméről egy forrás elemére.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos átmozog egy csőről egy ciszternára, azaz megváltoztatja helyét.

Use-case neve	Szerelő mozog forrásról csőre
Rövid leírás	A Szerelő játékos megpróbál mozogni a vízhálózat egy forrás eleméről egy cső elemére.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos megváltoztatja helyét, azaz egy ciszternáról egy csőre lépett. 2. A Szerelő játékos nem tudja megváltoztatni a helyét, mivel a "cél" csövön, ahova lépni szeretne, már áll másik játékos. Ekkor helyét megtartja a Szerelő játékos, azaz nem lép át a csőre, és így a ciszternán marad.

Use-case neve	Szabotőr mozog ciszternáról csőre
Rövid leírás	A Szabotőr játékos megpróbál mozogni a vízhálózat egy ciszterna eleméről egy cső elemére.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szabotőr játékos megváltoztatja helyét, azaz egy ciszternáról egy csőre lépett. 2. A Szabotőr játékos nem tudja megváltoztatni a helyét, mivel a "cél" csövön, ahova lépni szeretne, már áll másik játékos. Ekkor helyét megtartja a Szabotőr játékos, azaz nem lép át a csőre, és így a ciszternán marad.

Use-case neve	Szabotőr mozog csőről ciszternára
Rövid leírás	A Szabotőr játékos átmozog a vízhálózat egy cső eleméről egy ciszterna elemére.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szabotőr játékos átmozog egy csőről egy ciszternára, azaz megváltoztatja helyét.

Use-case neve	Szabotőr mozog csőről forrásra
Rövid leírás	A Szabotőr játékos átmozog a vízhálózat egy cső eleméről egy forrás elemére.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szabotőr játékos átmozog egy csőről egy forrásra, azaz megváltoztatja helyét.

Use-case neve	Szabotőr mozog forrásról csőre
Rövid leírás	A Szabotőr játékos megpróbál mozogni a vízhálózat egy forrás eleméről egy cső elemére.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szabotőr játékos megváltoztatja helyét, azaz egy forrásról egy csőre lépett.

	2. A Szabotőr játékos nem tudja megváltoztatni a helyét, mivel a “cél” csövön, ahova lépni szeretne, már áll másik játékos. Ekkor helyét megtartja a Szabotőr játékos, azaz nem lép át a csőre, és így a forráson marad.
--	--

Use-case neve	Szabotőr mozog csőről pumpára
Rövid leírás	A Szabotőr játékos átmozog a vízhálózat egy cső eleméről egy pumpa elemére.
Aktorok	User
Forgatókönyv	1. A Szerelő játékos átmozog egy csőről egy pumpára, azaz megváltoztatja helyét.

Use-case neve	Szabotőr mozog pumpáról csőre
Rövid leírás	A Szabotőr játékos megpróbál mozogni a vízhálózat egy pumpa eleméről egy cső elemére.
Aktorok	User
Forgatókönyv	1. A Szabotőr játékos megváltoztatja helyét, azaz egy pumpáról egy csőre lépett. 2. A Szabotőr játékos nem tudja megváltoztatni a helyét, mivel a “cél” csövön, ahova lépni szeretne, már áll másik játékos. Ekkor helyét megtartja a Szabotőr játékos, azaz nem lép át a csőre, és így a pumpán marad.

Use-case neve	Szerelő pumpát megjavít
Rövid leírás	A Szerelő játékos megpróbálja megjavítani azt a pumpát amin áll.
Aktorok	User
Forgatókönyv	1. A Szerelő játékos megjavítja a pumpát amin éppen áll, azaz a pumpa ismét működik, ellátja a funkcióját. 2. Amennyiben a pumpa működik, abban az esetben tovább fog működni.

Use-case neve	Szerelő pumpát felvesz
Rövid leírás	A Szerelő játékos megpróbálja felvenni egy pumpát amely azon a ciszterna elemen van amin éppen áll.
Aktorok	User
Forgatókönyv	1. A Szerelő játékos felveszi a pumpa elemet a ciszternáról. 2. Amennyiben a ciszternán -amin áll a Szerelő- nincs felvehető pumpa, abban az esetben nem tud felvenni pumpát.

Use-case neve	Szerelő pumpát lerak
Rövid leírás	A Szerelő játékos megpróbálja lerakni a pumpát arra a helyre a vízhálózaton, ahol éppen tartózkodik.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos lerakja a pumpát -amennyiben van nála lerakható pumpa- arra cső elemre amin éppen áll. Ekkor a cső "félbe lesz vágva", és a félbevágott két csövet fogja összekötni a lehelyezett pumpa. 2. Amennyiben nincs nála lerakható pumpa, akkor nem történik változás a vízhálózatban. 3. Amennyiben nem csövön próbálja lerakni a pumpát, akkor ez sikertelen lesz, azaz a vízhálózat felépítése nem fog megváltozni.

Use-case neve	Szerelő csövet felvesz pumpáról
Rövid leírás	A Szerelő játékos megpróbálja felvenni a cső végét arról a pumpáról amin éppen áll.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő felveszi a cső azon végét ami ahhoz a pumpához van közelebb amin éppen áll. 2. Ha a Szerelőnél már van cső elem szabad vége, akkor sikertelen lesz a cső felvétel kísérlet.

Use-case neve	Szerelő csövet felvesz ciszternáról
Rövid leírás	A Szerelő játékos megpróbálja felvenni a csövet arról a ciszternáról amin éppen áll.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos felveszi a cső azon végét, ami ahhoz a ciszternához van közel, amin a Szerelő éppen áll. 2. Amennyiben a Szerelőnél már van egy cső elem szabad vége, akkor sikertelen lesz a kísérlet.

Use-case neve	Szerelő csövet felvesz forrásról
Rövid leírás	A Szerelő játékos megpróbálja felvenni a csövet arról a forrásról amin éppen áll.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos felveszi a cső azon végét, ami ahhoz a forráshoz van közel, amin a Szerelő éppen áll. 2. Amennyiben a Szerelőnél már van egy cső, akkor sikertelen lesz a kísérlet.

Use-case neve	Szerelő cső szabad végét felvesz
Rövid leírás	A Szerelő játékos megpróbálja felvenni a cső szabad végét arról a ciszternáról amin éppen áll.

Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos felveszi a cső szabad végét a ciszternáról. 2. Amennyiben már birtokolja valamely másik cső szabad végét, a kísérlet sikertelen lesz.

Use-case neve	Szerelő csövet lerak pumpára
Rövid leírás	A Szerelő játékos lerakja a cső végét arra a pumpára amin éppen áll
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő lerakja a nála lévő cső végét arra a pumpára amin éppen áll.

Use-case neve	Szerelő csövet lerak ciszternára
Rövid leírás	A Szerelő játékos lerakja a cső elemet arra a ciszternára amin éppen áll.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos lerakja a nála lévő cső szabad végét arra a ciszternára amin éppen áll.

Use-case neve	Szerelő csövet lerak forrásra
Rövid leírás	A Szerelő játékos megpróbálja lerakni a cső szabad végét arra a forrásra amin éppen áll.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos lerakja a nála lévő cső szabad végét arra a forrásra amin éppen áll.

Use-case neve	Szabotőr csövet lyukaszt
Rövid leírás	A Szabotőr játékos megpróbálja kilyukasztani azt a cső elemet amin éppen áll.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szabotőr játékos kilyukasztja a csövet, azaz a bele kerülő egység víz kifolyik. 2. Amennyiben a cső már lyukas, a kísérlet nem hajtodik végre, így a cső állapota nem változik meg.

Use-case neve	Szerelő csövet megjavít
Rövid leírás	A Szerelő játékos megjavítja azt a cső elemet amin éppen áll.
Aktorok	User
Forgatókönyv	<ol style="list-style-type: none"> 1. A Szerelő játékos megjavítja azt a cső elemet amin éppen áll.

- A "Szabotőr csövet felvesz", "Szerelő csövet lyukaszt", valamint a játék kiírásával implicit és explicit ellentmondó lépéseket nem modelleztük, és nem teszteltük, mert a feladat kiírása nem teszi lehetővé ezen funkciók a játékba való integrálását.
- "Játék indítása", "Játék vége" lépéseket itt nem lehet modellezni.
- A "Játék init" implicit tesztelve van, minden esetben egy kis tesztpálya generálódik.

5.2 A szkeleton kezelői felületének terve, dialógusok

A program elindulása után a tesztesetek egy lista formájában jelennek meg, amelyek közül a felhasználó választhat. A választáshoz az adott eset számát kell begépelni. Miután ez megtörtént, a képernyőn megjelennek a következő információk, a következő módon: tabulálás mértéke, objektumnév, függvénynév, sor változtatása.

Ekkor az objektumok az adott teszteset során résztvevő objektumokat jelentik, a függvények az ezekhez tartozó publikus függvények, a tabulálás nagysága pedig az alapján kerül meghatározásra, hogy a függvények által alkotott hívási lánc elemei közül hányadiknak került meghívásra az adott függvény.

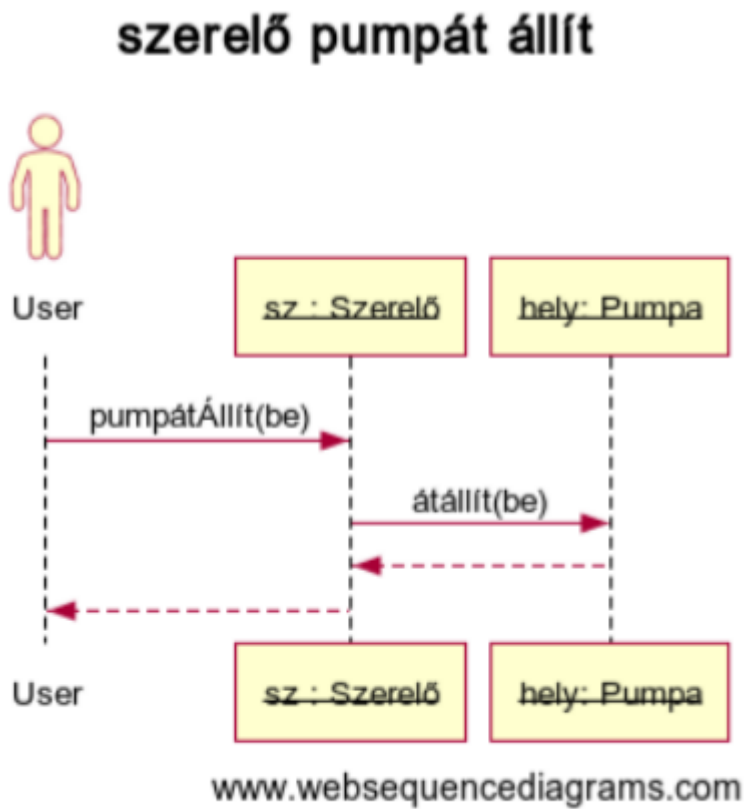
Tehát, ha a "fv1()" meghívja a "fv2()" -t, akkor a kettő függvény egymás alatt, és utóbbi 1 tabulálásnyi mértékkel beljebb fog megjelenni a képernyőn. Ha ezután a "fv2()" meghívja a "fv3()" függvényt, akkor a "fv3()" a "fv2()" alatti sorban, ahhoz képest 1 tabulálás mértékkel beljebb jelenik meg. Abban az esetben, ha a "fv1()" meghívja a "fv2()" -t, majd az visszatér, majd ismételten a "fv1()" meghívja a "fv3()" -t, akkor a képernyőn a három függvény egymás alatt, de a "fv2()" és "fv3()" a "fv1()" -hez képest azonosan, 1 tabulálás mértékkel beljebb jelenik meg. Az elsőként meghívott függvény esetében a tabulálás mértéke mindig 0.

Egyes függvények, bizonyos esetekben a felhasználóval kommunikálnak. Ebben az esetben a képernyőn megjelenik a függvény, ami a felhasználóval kommunikál. Ez alatt megjelenik a kommunikáció szövege, pl. "Van-e víz a csőben? (Y/N)", majd a további függvények. Ekkor a felhasználó az Y vagy N betűk begépelésével választhat a kettő lehetőség közül. A választ ebben az esetben ugyanabba a sorba írjuk, mint a kérdés, és a kérdést nem tabuláljuk beljebb az esztétikusabb megjelenés érdekében.

Minden felsorolt use-case-hez tartozik egy opció a menüben.

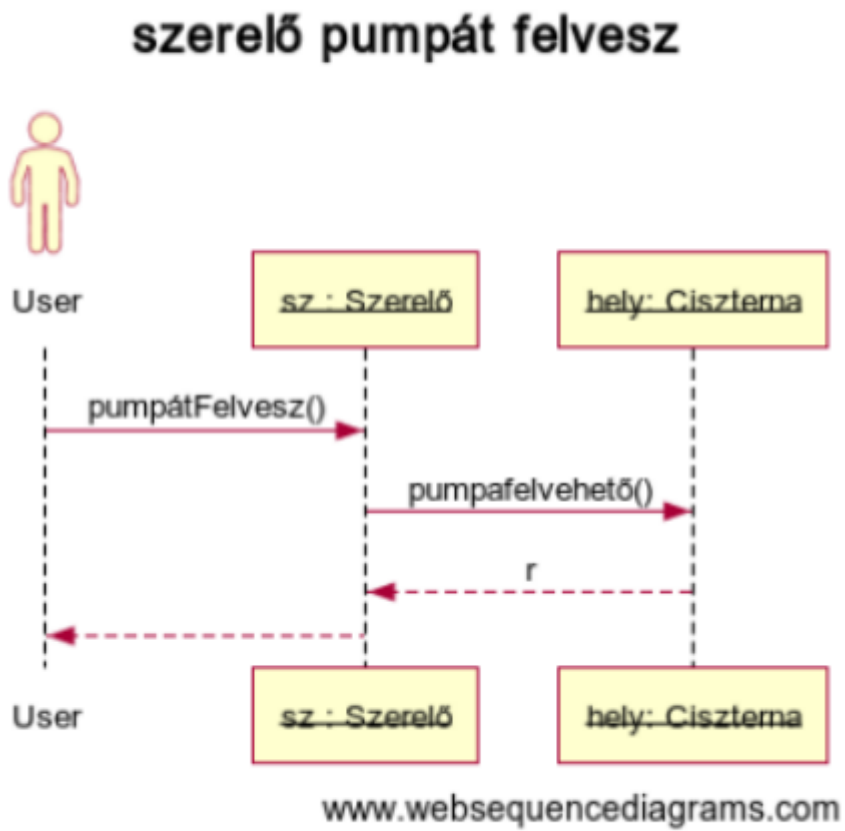
5.3 Szekvencia diagramok a belső működésre

5.3.1 Pumpát állít



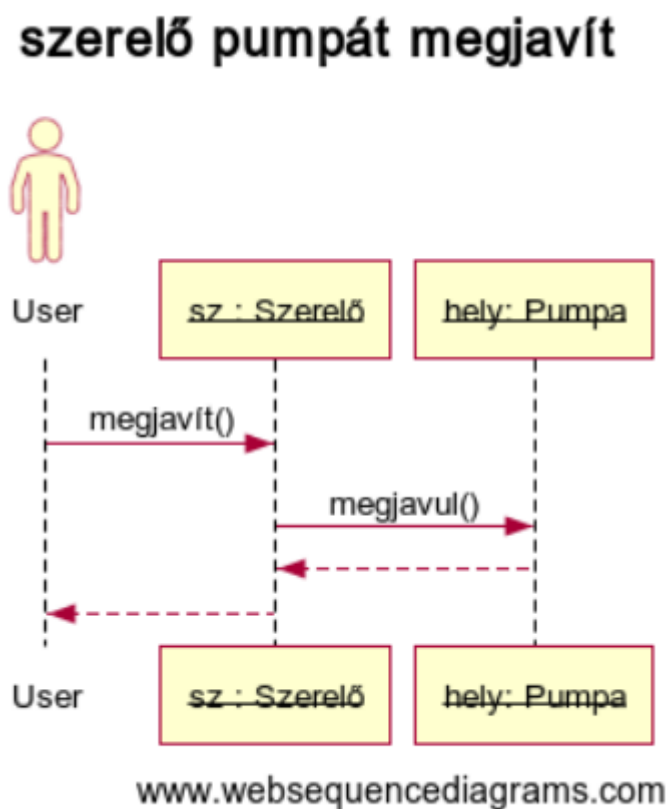
Itt a szerelő szabotőr is lehet, a diagram ugyanígy néz ki. Use-casek: szerelő pumpát állít, szabotőr pumpát állít.

5.3.2 Szerelő pumpát felvesz



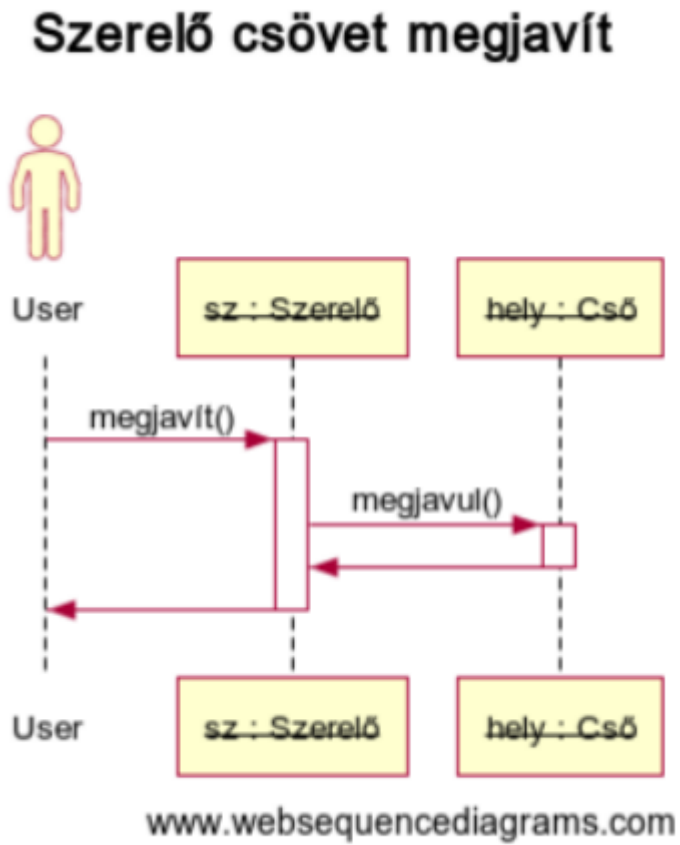
Use-case(k): szerelő pumpát felvesz

5.3.3 Szerelő pumpát megjavít



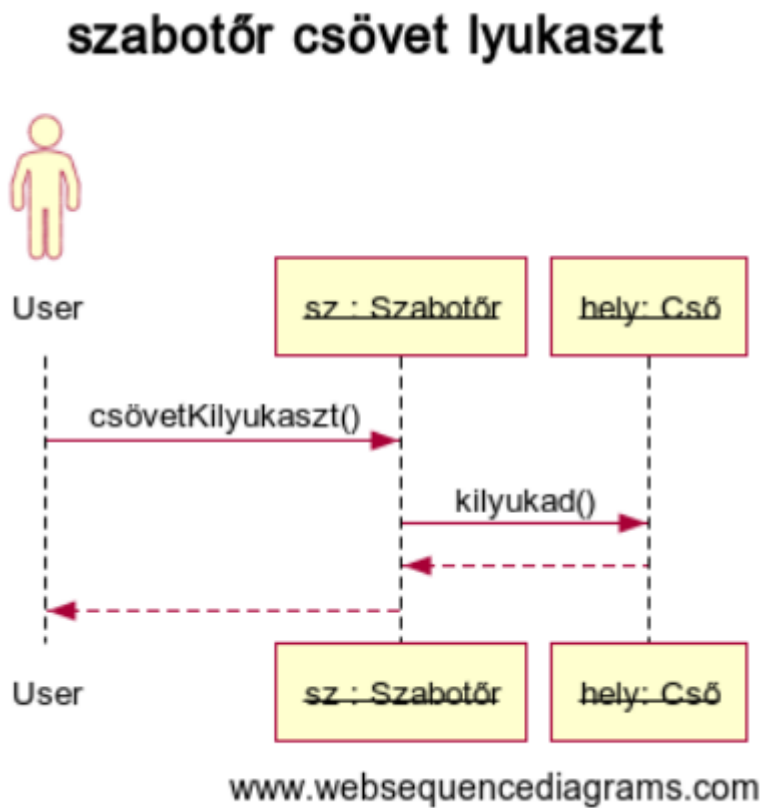
Use-case(k): szerelő pumpát megjavít

5.3.4 Szerelő csövet megjavít



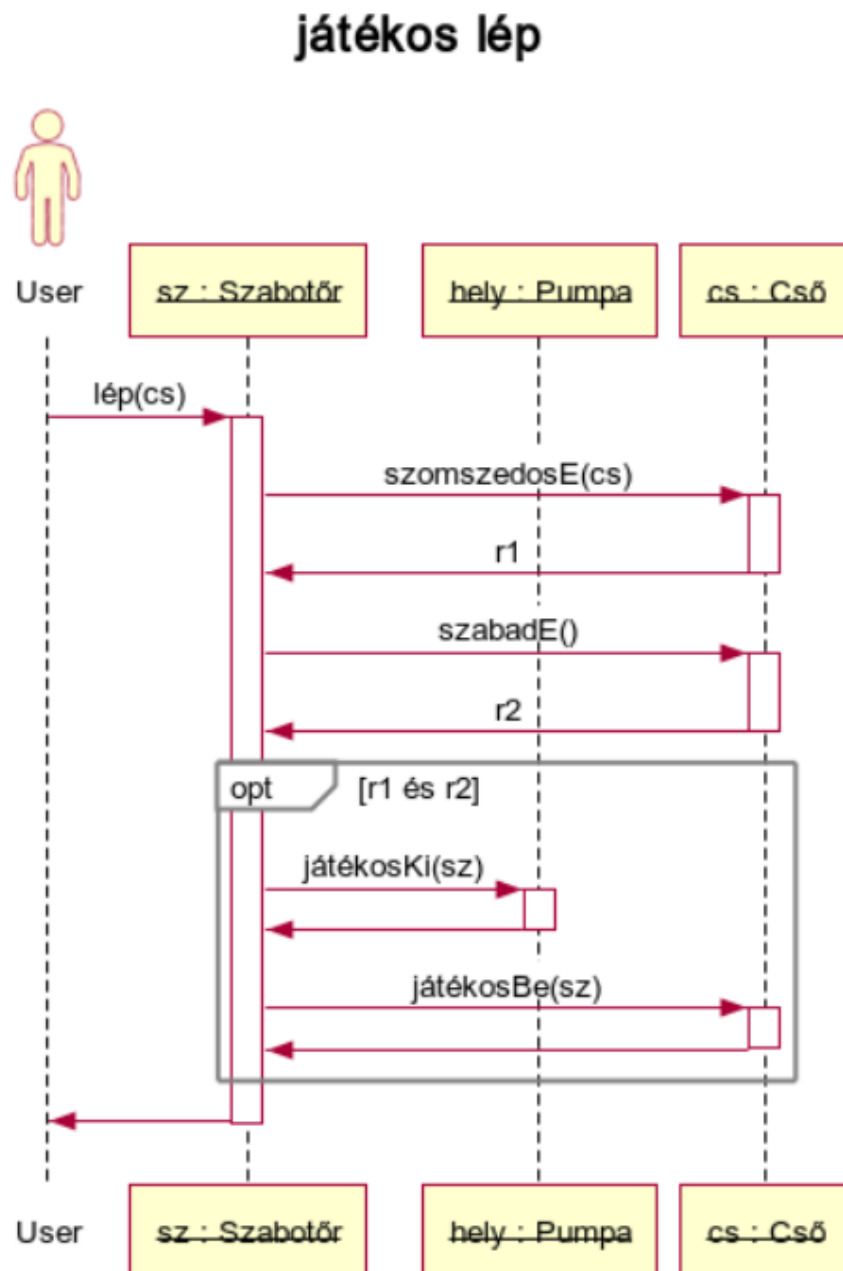
Use-case(k): szerelő csövet megjavít

5.3.5 Szabotőr csövet lyukaszt



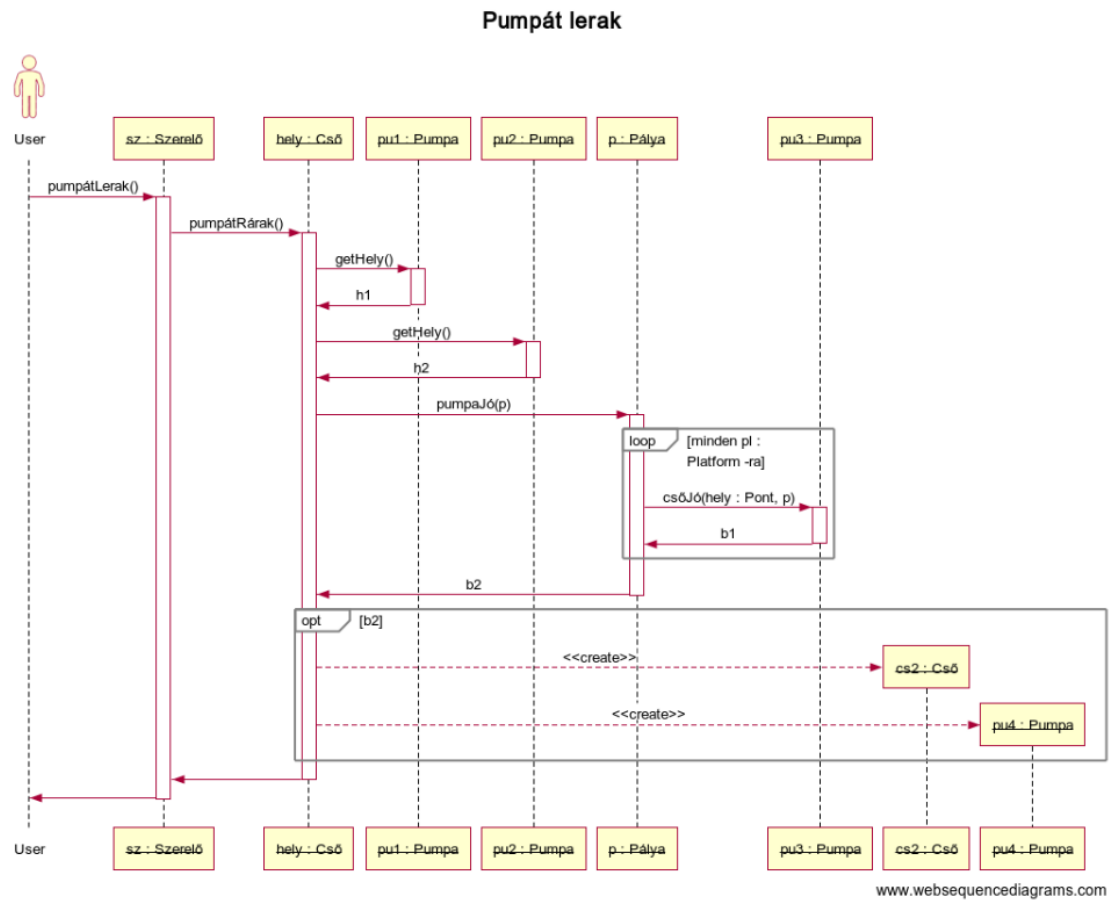
Use-case(k): szabotőr csövet lyukaszt

5.3.6 Szabotőr mozog pumpáról csőre



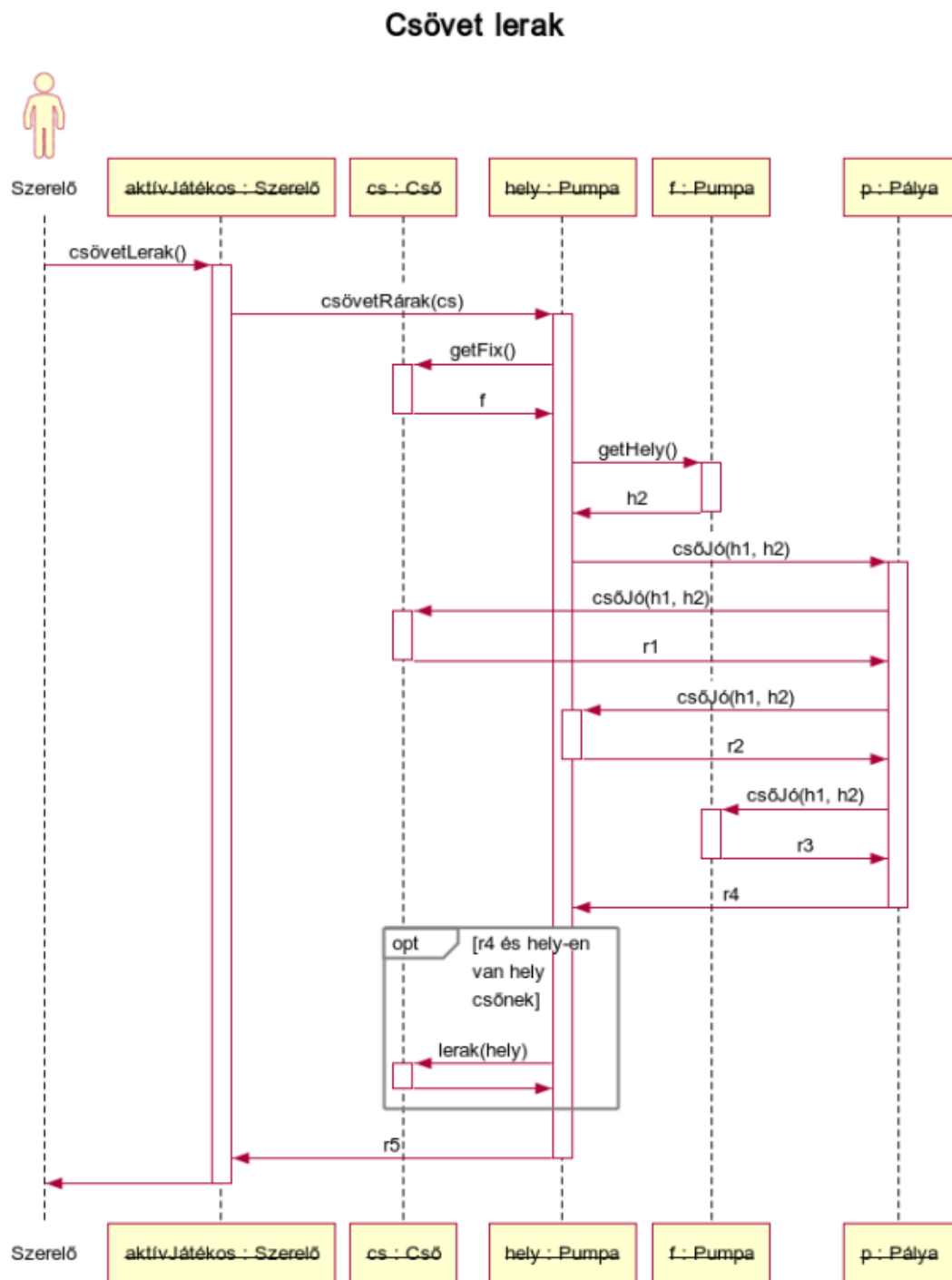
Itt a hely pumpán kívül lehet ciszterna vagy forrás is, az ábra ugyanígy nézne ki. Use-case(k): szabotőr mozog pumpáról csőre, szabotőr mozog ciszternáról csőre, szabotőr mozog forrásról csőre, szerelő mozog pumpáról csőre, szerelő mozog ciszternáról csőre, szerelő mozog forrásról csőre.

5.3.7 Szerelő pumpát lerak



Itt a pu1, pu2 lehet pumpa, ciszterna vagy forrás és pu3 lehet pumpa ciszterna forrás vagy cső is a diagram ugyanaz marad. Use-case(k): szerelő pumpát lerak.

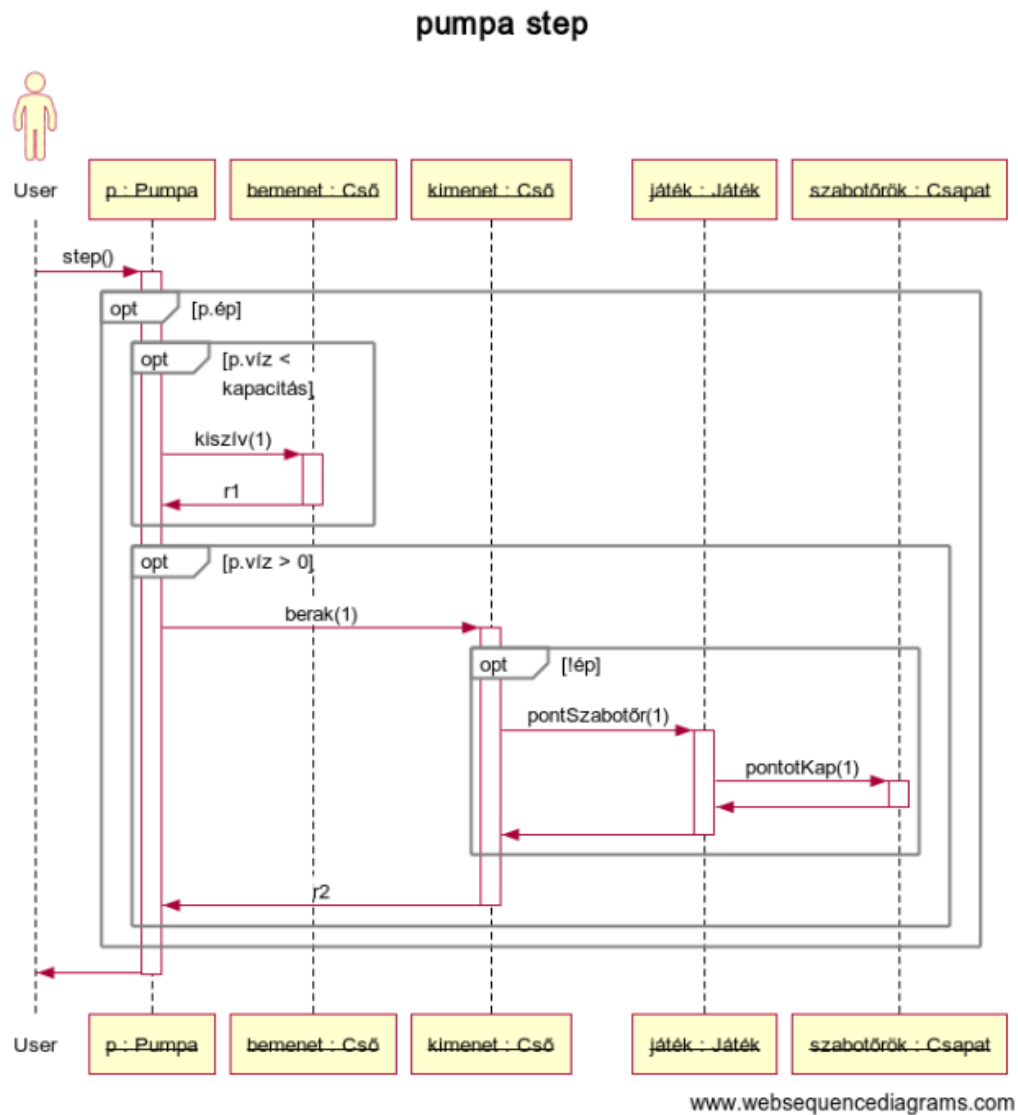
5.3.8 Szerelő csövet lerak



www.websequencediagrams.com

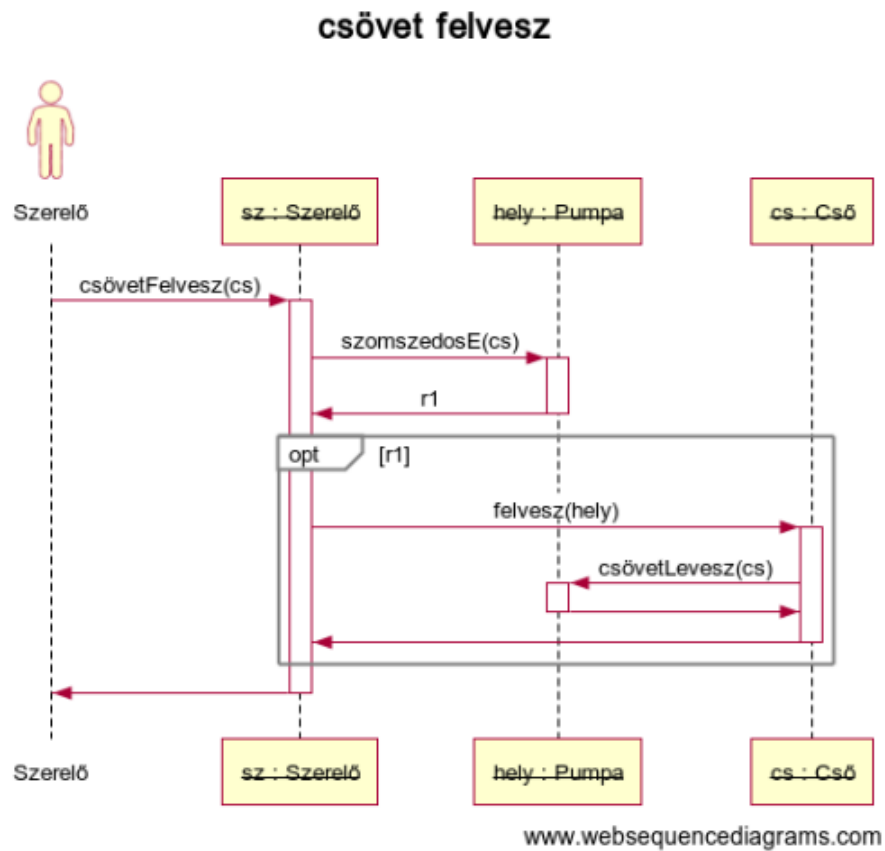
Itt f lehet pumpa, ciszterna vagy forrás. Use-case(k): szerelő csövet lerak pumpára, szerelő csövet lerak ciszternára, szerelő csövet lerak forrásra

5.3.9 Pumpa lép



Use-case(k): Pumpa lép.

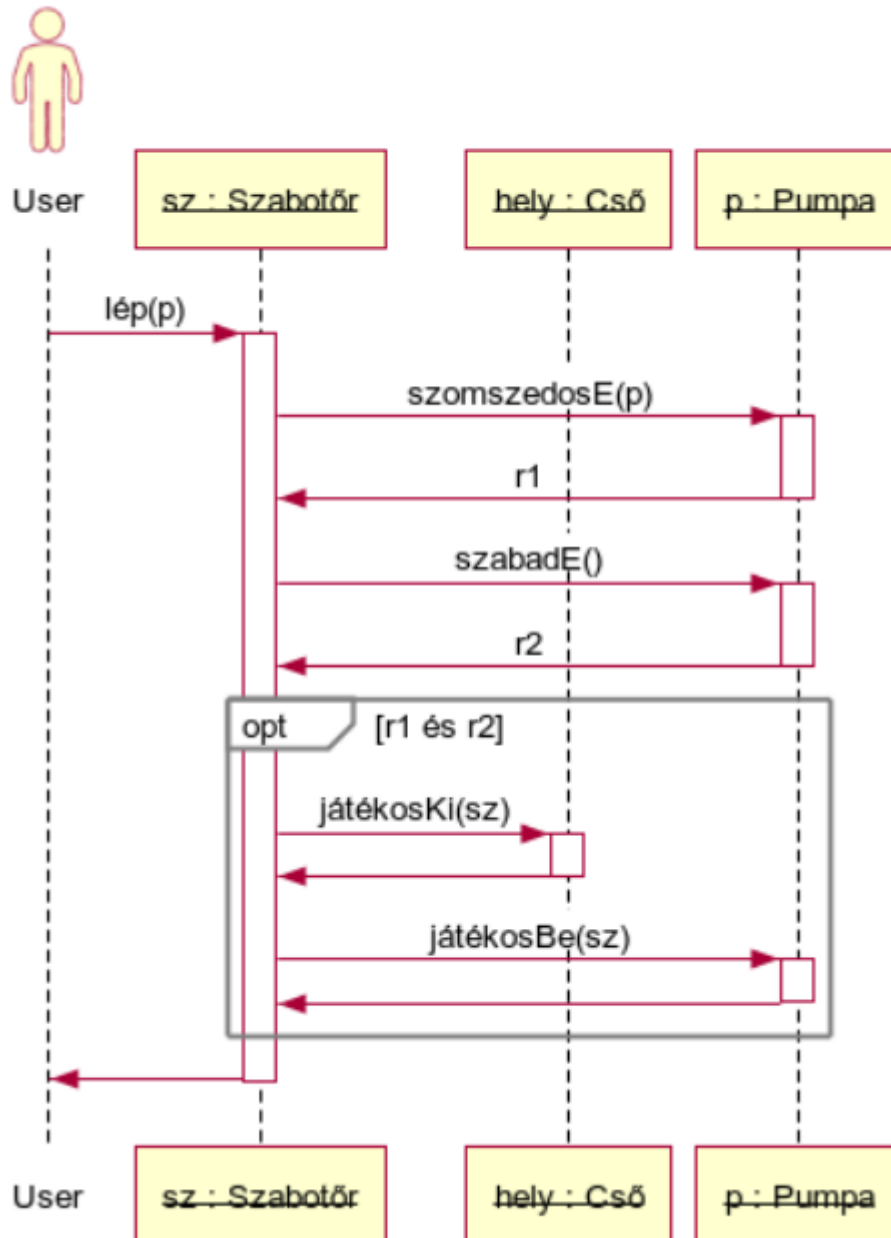
5.3.10 Szerelő csövet felvesz



Itt a hely lehet pumpa, ciszterna vagy forrás is. Use-case(k): szerelő csövet felvesz pumpáról, szerelő csövet felvesz ciszternáról, szerelő csövet felvesz forrásról, szerelő cső szabad végét felvesz.

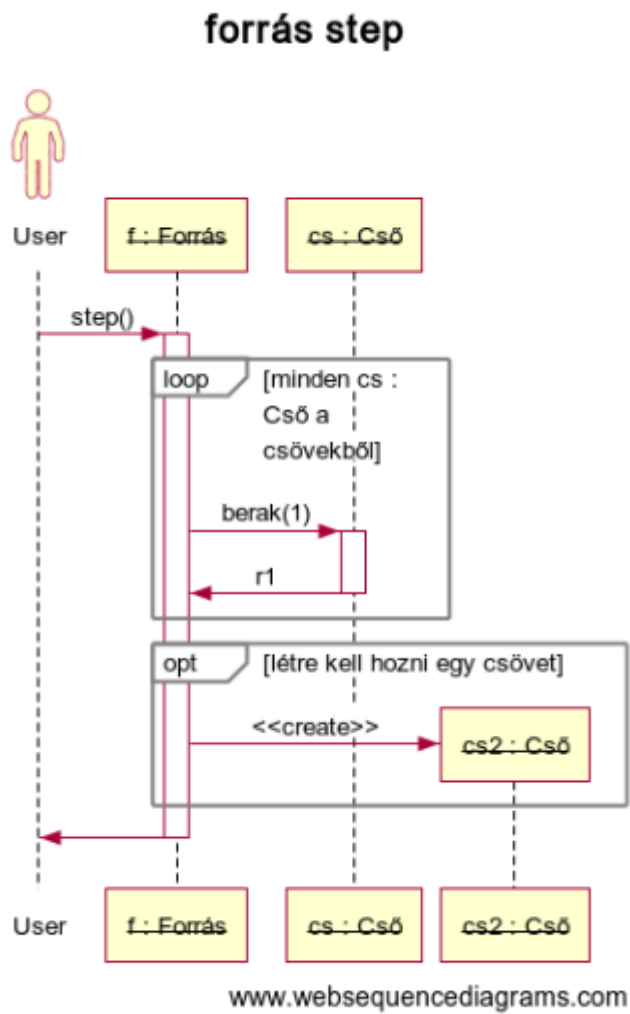
5.3.11 Szabotőr mozog csőről pumpára

játékos lép



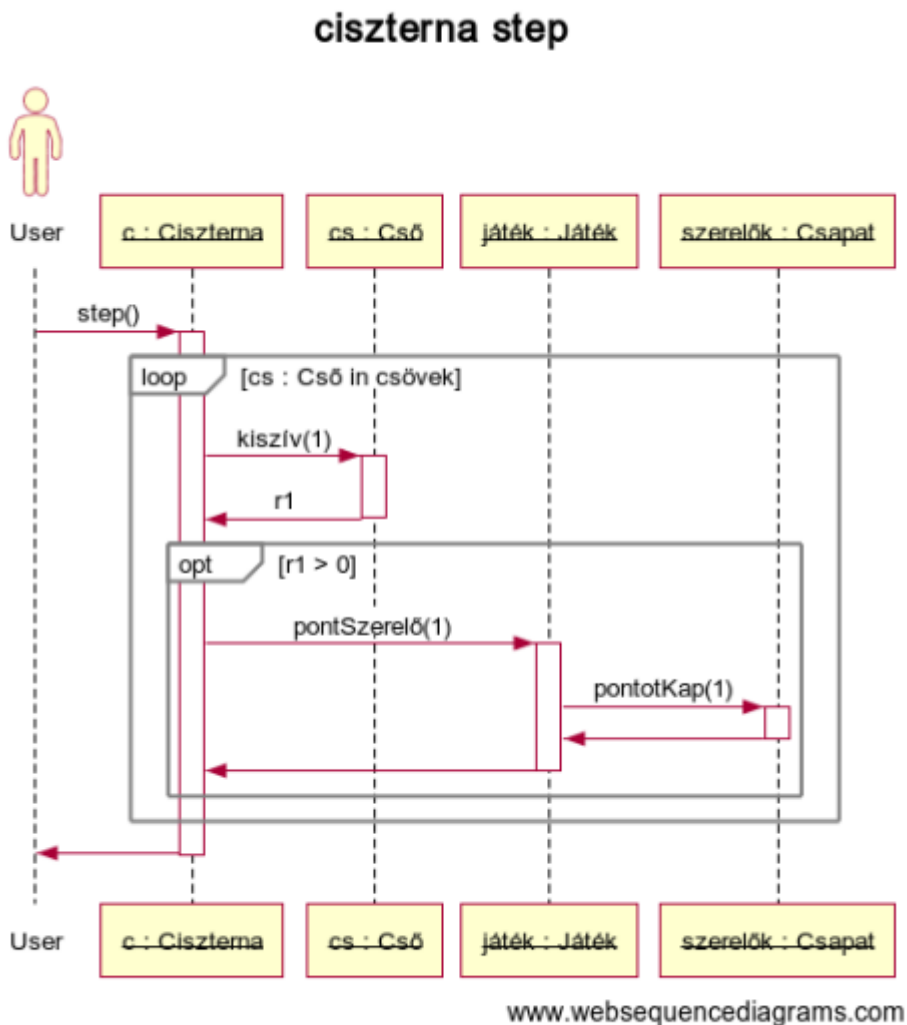
Itt a p lehet pumpa, ciszterna vagy forrás is. Use-case(k): szabotőr mozog csőről pumpára, szabotőr mozog csőről ciszternára, szabotőr mozog csőről forrásra, szerelő mozog csőről pumpára, szerelő mozog csőről ciszternára, szerelő mozog csőről forrásra.

5.3.12 Forrás lép



Use-case(k): Forrás lép.

5.3.13 Ciszterna lép

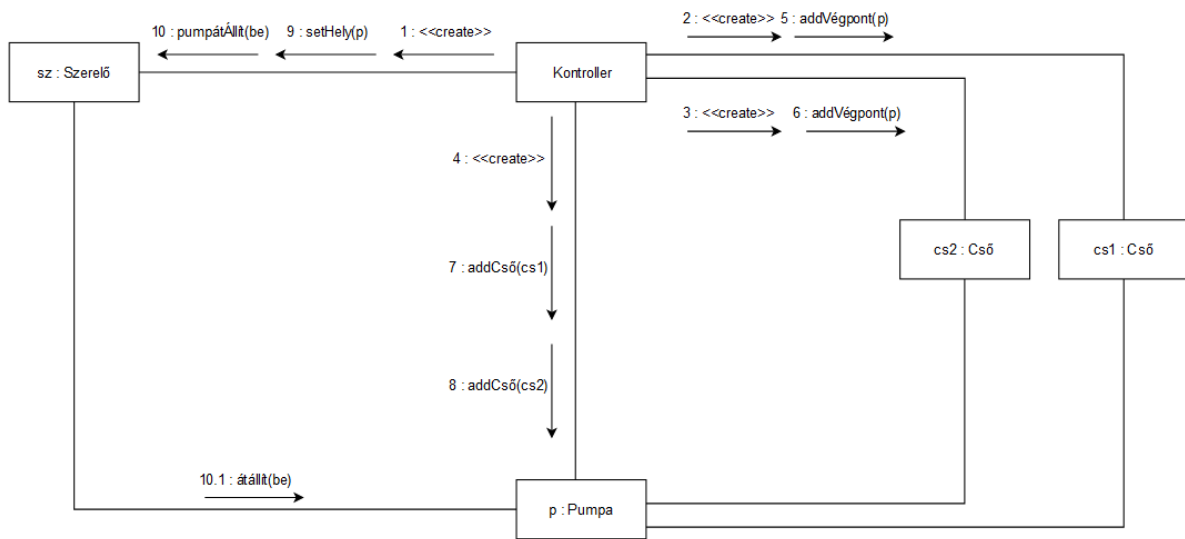


Use-case(k): Forrás lép.

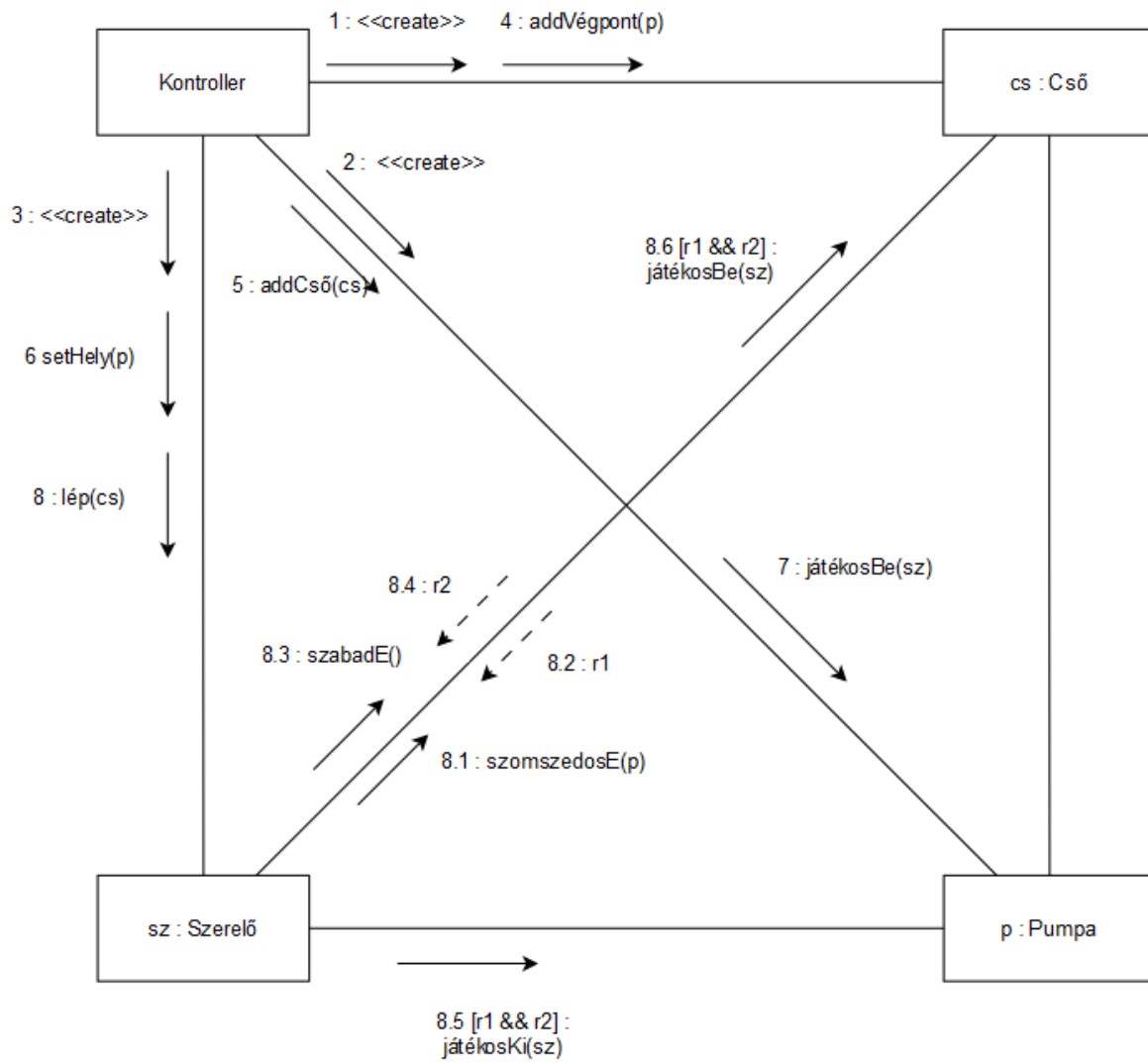
A diagramok és a use-casek számossága nem egyezik meg, mivel pl. a játékos lépésnél cső->pumpa, cső->forrás az külön use-case mégis a szekvencia diagram ugyanaz lenne. Ez ahol fontos ki is van emelve a diagram alatt közvetlen, emelet ott van mely use-caseket foglalja magába. Vannak olyan use-casek amelyeket nem szedtünk külön, mivel nem lenne értelme annyi (pl.: ~36) use-case-t csinálni, amikor annak egy szekvencia-diagramja lesz (ilyen például a csövet lerak). Vannak olyan use-casek amelyeket viszont azért hagytunk el mivel a játék leírása vagy előzőekben bemutatott osztálymodell nem engedélyezi azt a bizonyos lépés sorozatot (pl.: a játék specifikációjából adódik hogy egy játékos nem léphet egy csőről rá egy másik csőre vagy hogy szerelő nem tud csövet kilyukasztani). Végül vannak olyan dolgok amelyek itt nem lehet modellezni pl.: a játék befejezése, hiszen ahhoz le kellene játszani egy egész játékot. A játék init azért nincs feltüntetve, mivel minden esetben a program generál magának egy kis “sandbox” pályát amelyen elvégzi a kül. use-casekhez tartozó lépéseket, így az implicit tesztelve van.

5.4 Kommunikációs diagramok

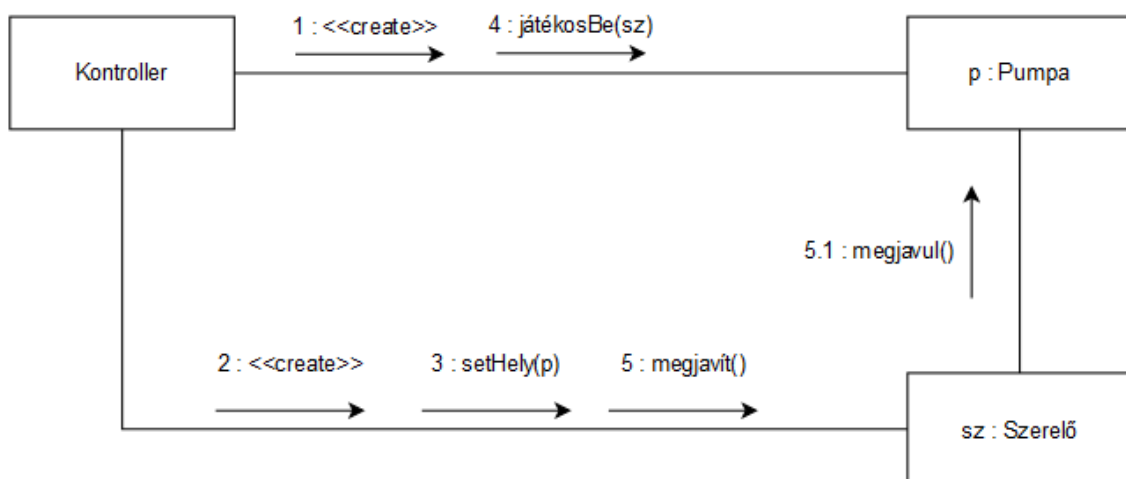
5.4.1 Pumpát állít



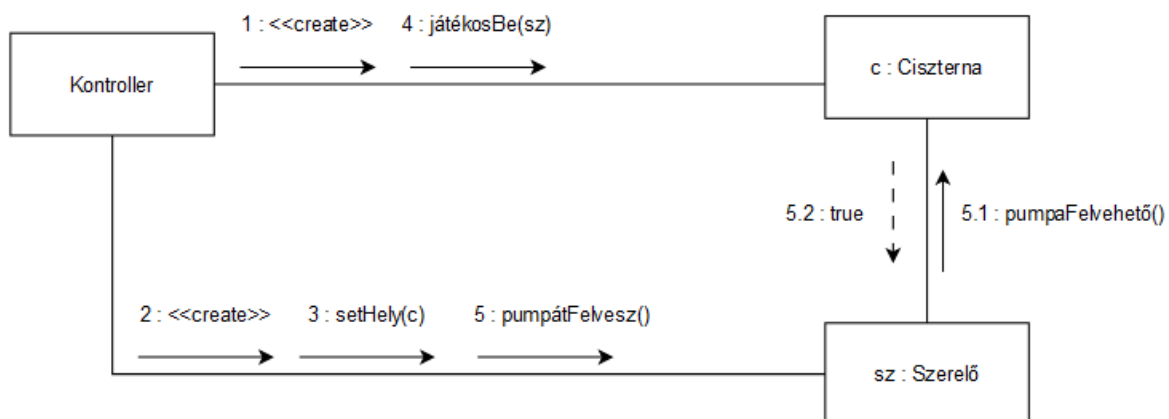
5.4.2 Játékos mozog



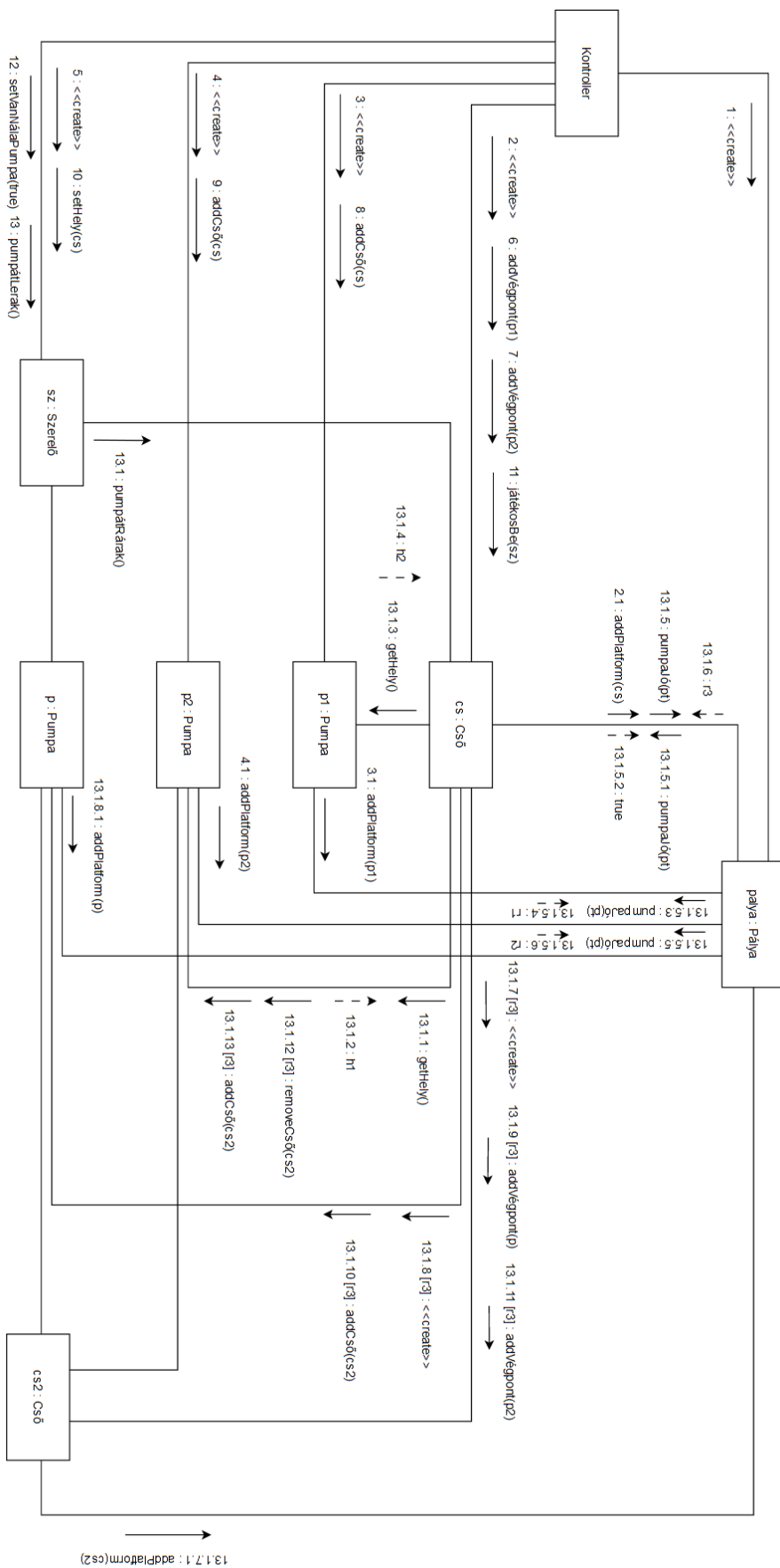
5.4.3 Javít



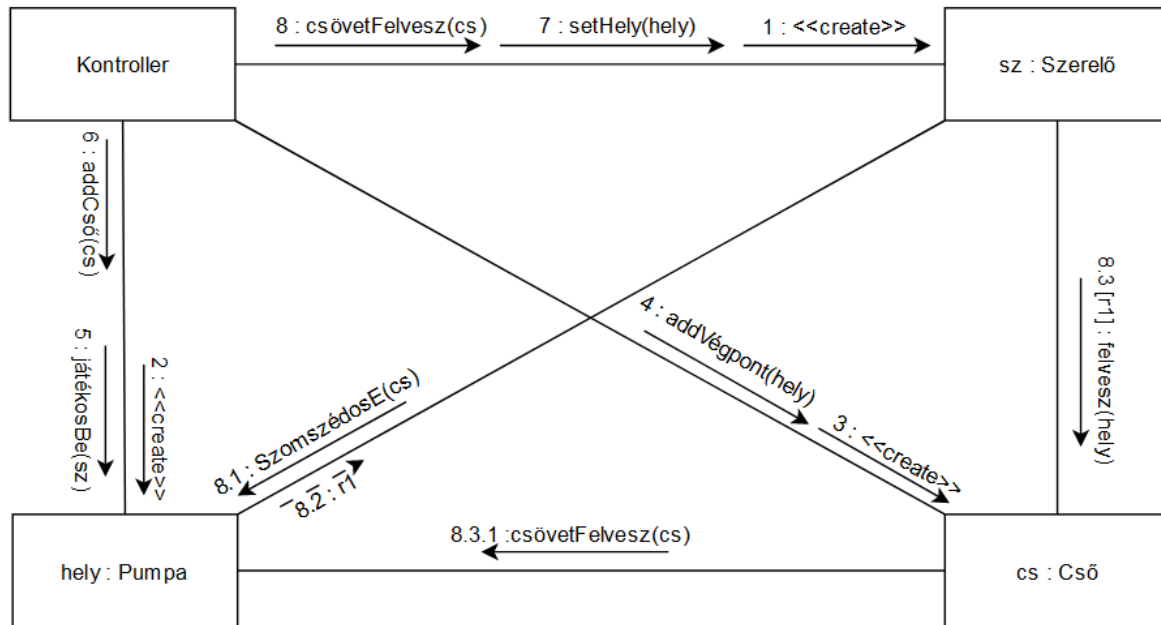
5.4.4 Pumpát felvesz



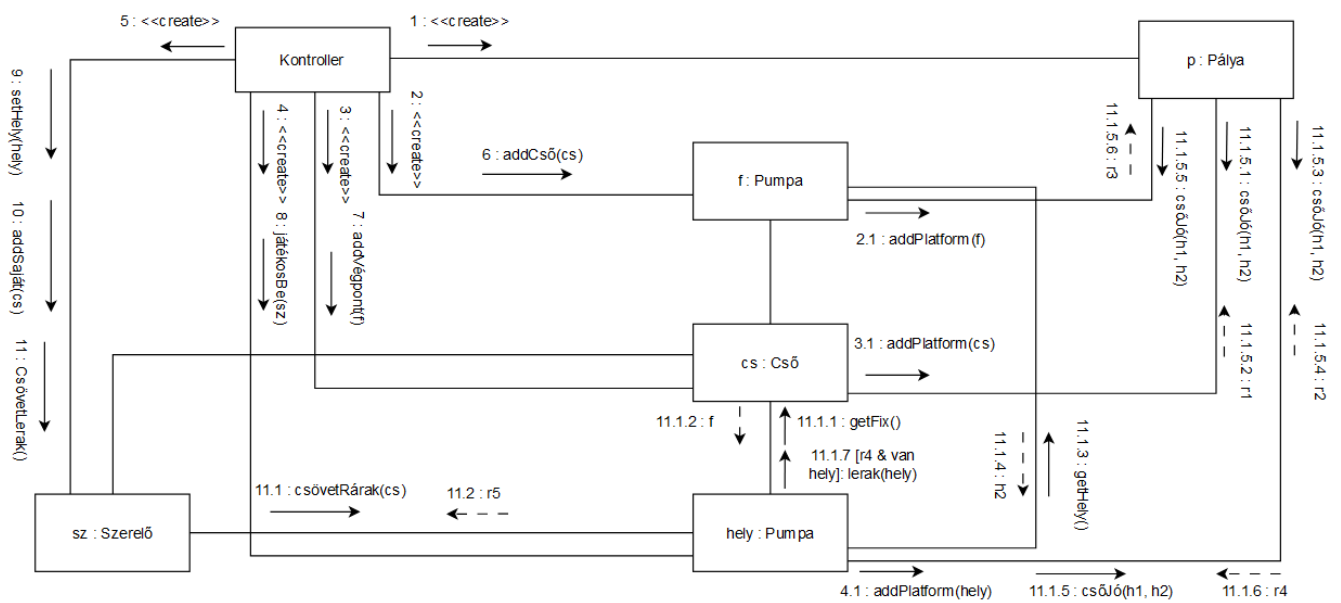
5.4.5 Pumpát lerak



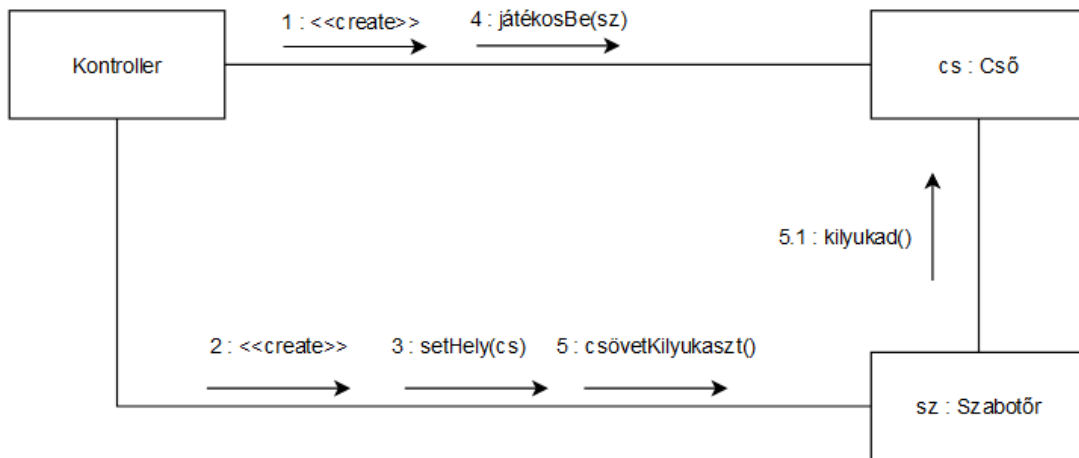
5.4.6 Csövet felvesz



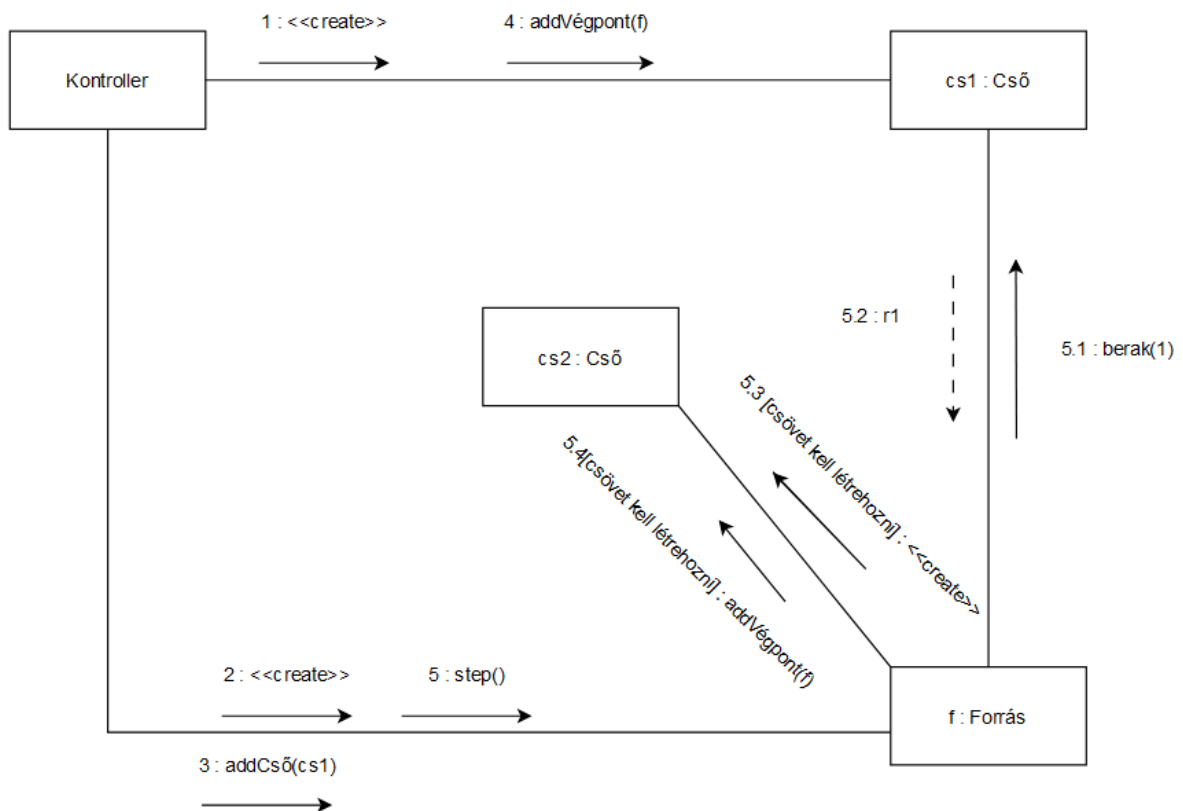
5.4.7 Csövet lerak



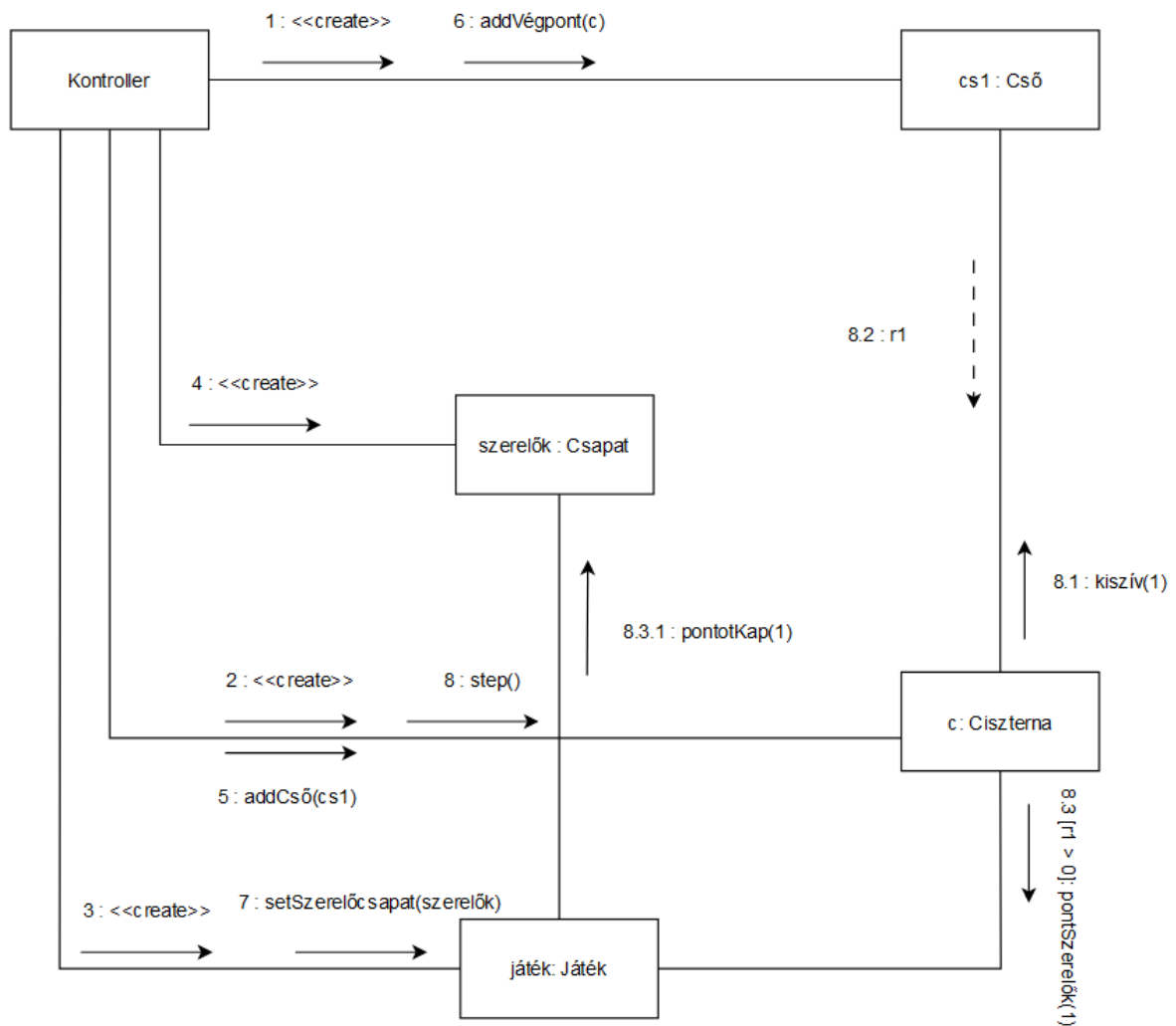
5.4.8 Csövet lyukaszt



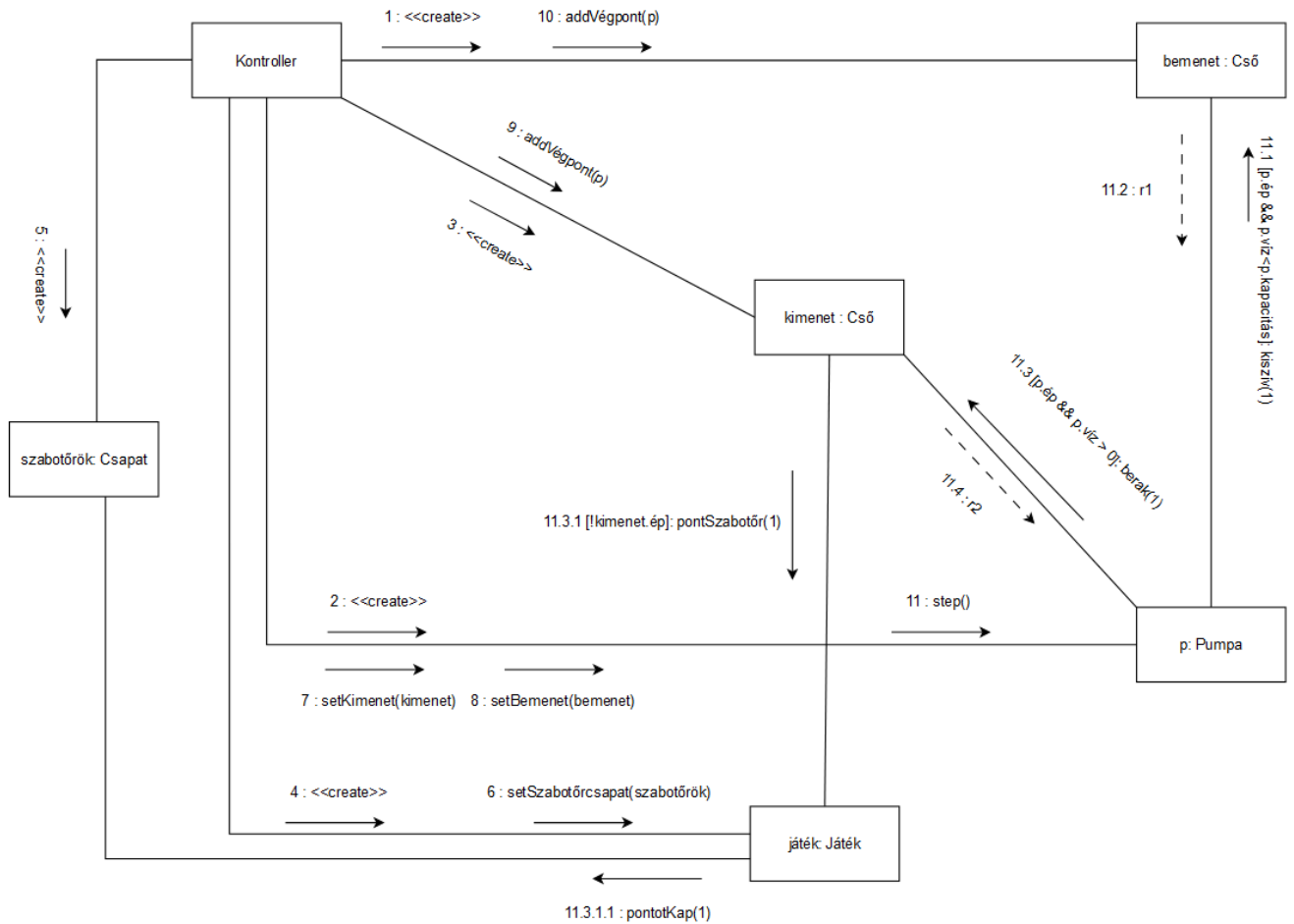
5.4.9 Forrás lép



5.4.10 Ciszterna lép



5.4.11 Pumpa lép



5.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2023.04.01. 11:00	2 óra	Tóth, Gegő, Koleszár, Kovács, Szautner	Értekezlet
2023.04.01. 13:00	4 óra	Gegő, Tóth	Kommunikációs diagramok
2023.04.01. 13:00	4 óra	Kovács, Koleszár	Szekvenciadiagramo k
2023.04.01. 13:00	4 óra	Szautner	Use-case diagramok
2023.04.02. 10:30	2 óra	Tóth	Kommunikációs diagramok
2023.04.02 18:00	3 óra	Gegő Tóth	Kommunikációs diagramok
2023.04.02 18:30	2.5 óra	Kovács	Kommunikációs diagramok
2023.04.02 19:00	3.5 óra	Koleszár	5.4 Fejezet dokumentálásának elkészítése Lektorálás, javítások
2023.04.02 21:00	1.5 óra	Szautner	Dokumentum átnézése, leadásra előkészítése

6. Szkeleton beadás

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
mainPackage/Main.java	4.47 KB	2023.04.15 16:00	Tesztesetek kezelése, megadott tesztelés futtatása.
mainPackage/SkeletonController.java	2.94 KB	2023.04.15 16:00	Tesztesetek
mainPackage/SkeletonIO.java	1.32 KB	2023.04.15 16:00	Skeleton Input-Output osztály
model/Active.java	2.7 KB	2023.04.15 16:00	Aktív elemet megvalósító őosztály
model/Cistern.java	652 Bytes	2023.04.15 16:00	Ciszternát megvalósító osztály
model/Game.java	1.22 KB	2023.04.15 16:00	Játékot megvalósító osztály
model/Map.java	1.02 KB	2023.04.15 16:00	Pályát megvalósító osztály
model/Mechanic.java	1.63 KB	2023.04.15 16:00	Szerelőt megvalósító osztály
model/Pipe.java	4.81 KB	2023.04.15 16:00	Csővet megvalósító osztály
model/Platform.java	1.14 KB	2023.04.15 16:00	Platformot megvalósító osztály
model/Player.java	1.02 KB	2023.04.15 16:00	Játékost megvalósító osztály
model/Point.java	331 Bytes	2023.04.15 16:00	Pontot megvalósító osztály
model/Pump.java	2.48 KB	2023.04.15 16:00	Pumpát megvalósító osztály
model/Saboteur.java	1.04 KB	2023.04.15 16:00	Szabotőrt megvalósító osztály
model/Spring.java	477 Bytes	2023.04.15 16:00	Forrást megvalósító osztály
model/Team.java	489 Bytes	2023.04.15 16:00	Csapatokat megvalósító osztály

6.1.2 Fordítás

A Fordításhoz a számítógépen, Windows 10 operációs rendszer alatt van lehetőség a következő lépések szerint:

1. Windows parancssor (cmd) megnyitása
2. Az aktuális mappába (forráskód főmappája) való navigálás után a következő parancsot kell futtatni: `javac model/*.java mainPackage/*.java`
3. Ezek után, ha nincs hibaüzenet, akkor a fordítás sikeres volt.

6.1.3 Futtatás

A Futtatáshoz a számítógépen, Windows 10 operációs rendszer alatt van lehetőség a következő lépések szerint:

1. Windows parancssor (cmd) megnyitása
2. Az aktuális mappába (forráskód főmappája) való navigálás után a következő parancsot kell futtatni: `java mainPackage/Main.java`
3. Ezek után a programnak meg kell adni a program által felsoroltak közül a futtatni kívánt tesztelés számát.

6.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Kovács Zoltán	W4YDIT	23
Gegő Levente	LYNUXB	23
Koleszár Kristóf	I1TLYH	18
Tóth Gábor	F041OM	18
Szautner Béla	EEID48	18

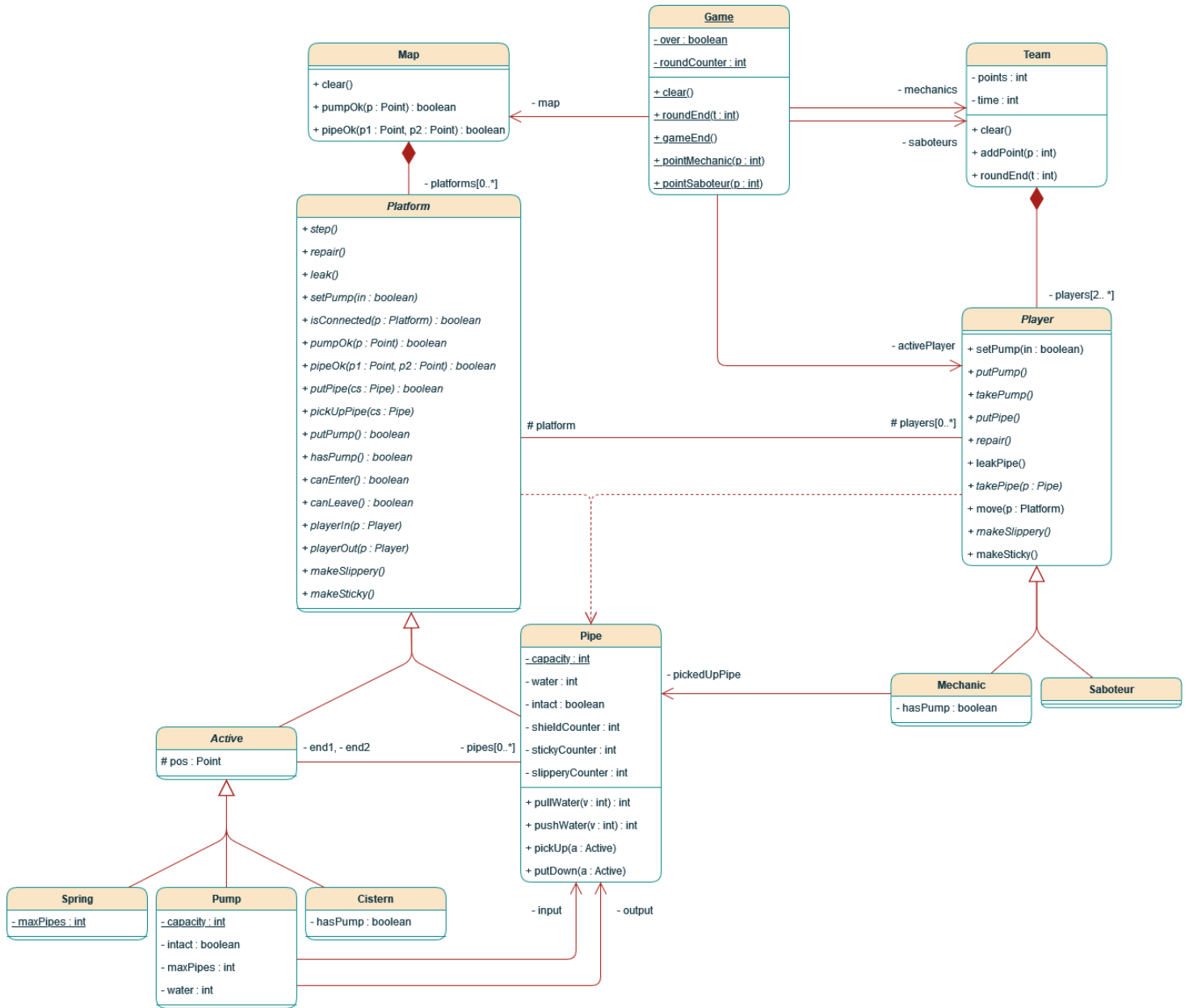
6.3 Napló

Kezdet	Időtartam	Résztevők	Leírás
2023. 04. 09. 14:00	3 óra	Gegő	Kódolás elkezdése
2023. 04. 11. 16:00	3 óra	Gegő	Kódolás folytatása
2023. 04. 13. 16:00	3 óra	Gegő	Kódolás befejezése
2023. 04. 14. 12:00	1,5 óra	Koleszár	Kód javítása
2023.04.15 11:00	2 óra	Kovács	Tesztesetek írása
2023. 04. 15. 13:00	5 óra	Gegő Kovács Szautner Koleszár	Értekezlet: Tesztesetek véglegesítése, dokumentáció elkezdése, kommentek készítése.
2023.04.16. 10:00	1 óra	Koleszár	Kommentek írása
2023.04.16 10:00	1 óra	Gegő	Kommentek írása
2023.04.16 10:00	3 óra	Tóth	Kommentek készítése
2023.04.16 12:00	3,5 óra	Szautner	Map, Point, SkeletonIO fájlokban kommentek készítése. Tesztek virtuális gépen futtatása. Dokumentáció elkészítése.
2023.04.16 13:00	2 óra	Kovács	Kommentek készítése a következő osztályokhoz: Platform, Player, Active
2023.04.16. 17:00	2 óra	Koleszár	Dokumentáció javítása, eddig munka ellenőrzése, kiseb hibák javítása
2023.04.16 18:00	3 óra	Kovács Gegő	A forrásfájlok kommentjeinek ellenőrzése és javítása

7. Prototípus koncepciója

7.0 Változás hatása a modellre

7.0.1 Módosult osztálydiagram



7.0.2 Új vagy megváltozó metódusok

Eltávolított metódusok:

- Map.default()

- `Game.start()`
- `Game.init()`
- `Team.fill()`
- `Player.putPipeBack()`
- `Player.round()`
- `Team.round()`

Létező de megváltozott metódusok / tagváltozók:

- `Game` referenciák helyett a program mostantól egy statikus `Game`-t használ
- `Player.leakpipe()`: mostantól nem absztrakt, implementációja megegyezik az eddigi szabotőrben lévő implementációval
- `Player.pos` át lett nevezve `Player.platform-ra`
- `Player.platform`, `Platform.players` és `Active.pos` mostantól `protected`
- `Platform.playerIn()`: a platformban absztrakt; `Active` ugyanúgy implementálja, mint ahogy eddig a platform; `Pipe` kiegészíti az implementációt a következőkkel: ha a cső csúszós ledobja a játékost, amikor rálép.
- `Platform.isFree()` mostantól `Platform.canEnter()`; mostantól absztrakt
- `Active.step()`: mostantól itt külön nem implementált, örökli a `Platform`tól
- `Platform.set()` mostantól `Platform.setPump()`
- A `Platform` osztály eddigi nem absztrakt metódusai mostantól absztraktok; Ezek implementációi változatlanul mostantól az `Active` osztályba kerültek át
- `Pipe.pickUp(Active a)`: most már fel lehet venni a cső mindkét végét, ha a csövet felveszik akkor mostantól a csőben lévő víz a homokba kifolyik, viszont ezért nem jár senkinek sem pont.

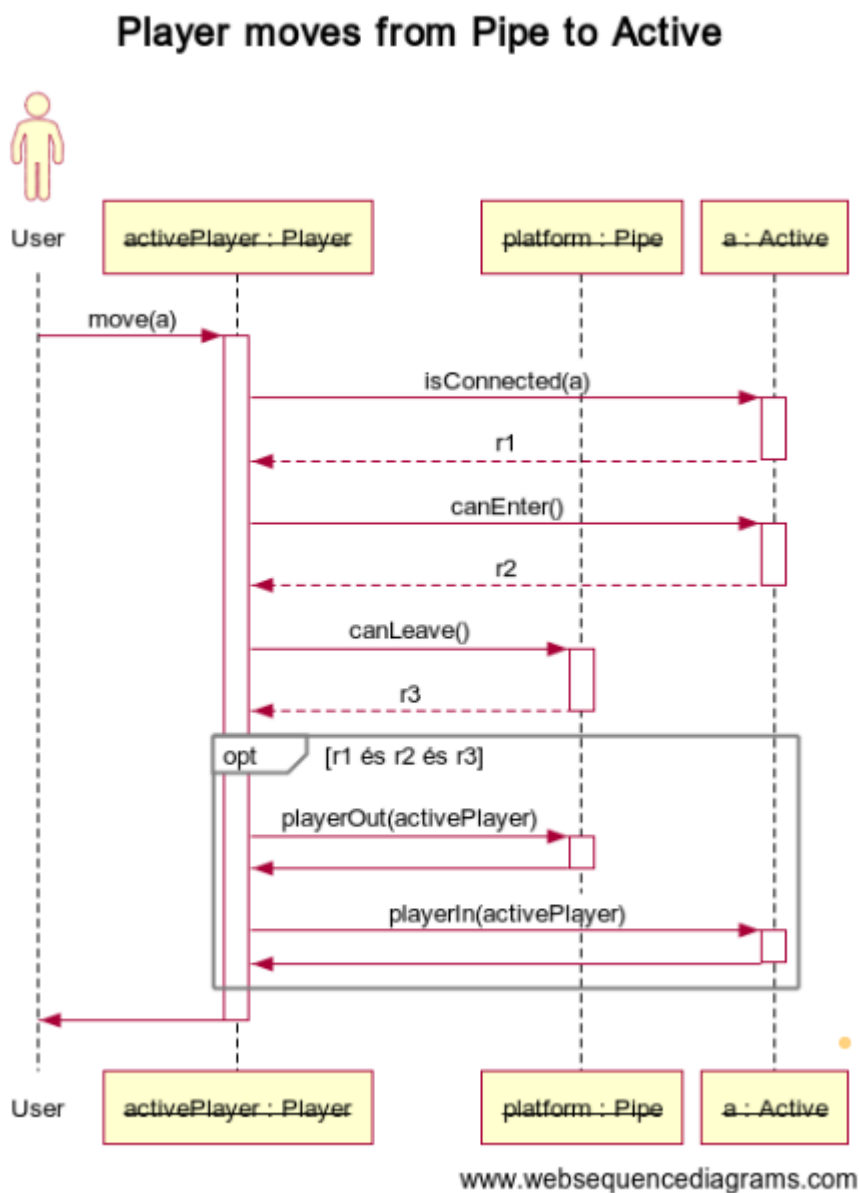
Új metódusok / tagváltozók:

- `Game.clear()`: törli a pályát, csapatokat és minden ezekhez tartozó objektumot.
- `Team.clear()`: törli a csapatban lévő `Player`-eket
- `Map.clear()`: törli a pályán lévő `Platform`-okat
- `Game.roundEnd(int t)`: a kör végén meghívja a megfelelő csapaton a `Team.roundEnd(int t)` metódust
- `Game.gameEnd()`: a játék végét kezeli le
- `Team.roundEnd(int t)`: levonja a csapat idejéből "t"-t, ha ezzel elfogyna a csapat ideje akkor befejezi a játékot (`Game.gameEnd()`)
- `Game.roundCounter`: számolja, hogy a játék hányadik körben jár, ezt használja fel az éppen soron következő játékos kiválasztásához
- `Platform.step()`: itt absztrakt (a `Pump`, a `Cistern` és a `Spring` osztályokban ennek a metódusnak a megvalósítása nem változik)
- `Pipe.step()`: a cső itt kezeli az idő múlását
- `bool Platform.canLeave()`: itt absztrakt; `Active`-nál mindig `true`; `Pipe`-nál attól, függ hogy ragadós-e a cső
- `Player.makeSlippery()`: itt absztrakt; a szabotőr implementálja; a szabotőr a csövet amin áll csúszóssá teszi
- `Player.makeSticky()`: a játékos a csövet amin áll csúszóssá teszi
- `Platform.makeSlippery()`: itt absztrakt; `Pump`, `Cistern`, `Spring`-ben üres; `Pipe`-nál a cső csúszóssá válik
- `Platform.makeSticky()`: itt absztrakt; `Pump`, `Cistern`, `Spring`-ben üres; `Pipe`-nál a cső ragadóssá válik
- `int Pipe.slipperyCounter`, `int Pipe.stickyCounter`, `int Pipe.shieldCounter`: ezek azt számolják, hogy a cső mennyi ideig csúszósak, ragadósak, védettek (ebben a sorrendben)

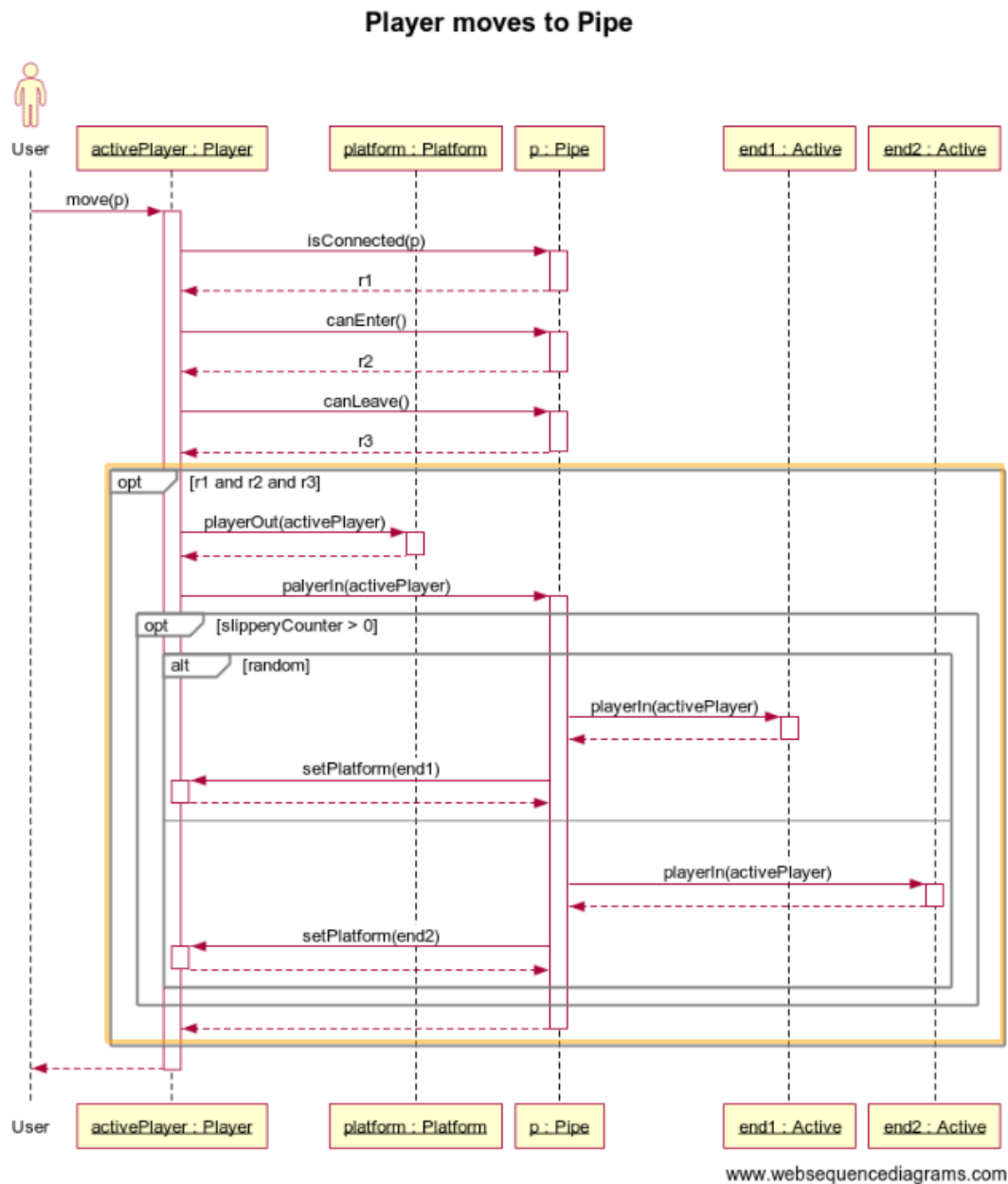
7.0.3 Szekvencia-diagramok

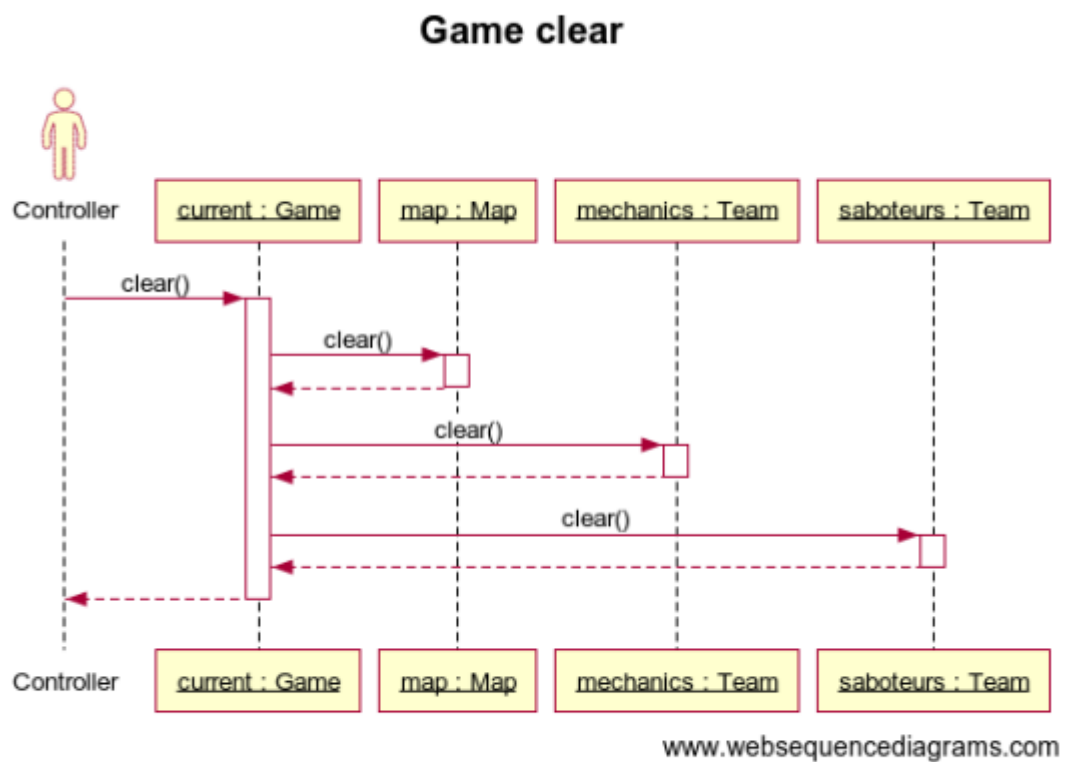
Csak azokhoz a metódusokhoz csináltunk szekvencia diagramot, melyekben érdemi logikai / algoritmikus változás történt. Egyik osztályból másik osztályba átkerülő, de változatlan metódusokhoz nincsenek itt szekvencia diagramok. Nevükben igen, de implementációjukban nem változó metódusokhoz nincsenek itt szekvencia diagramok. Mivel Java-ban GC van, ezért a `Game.clear()` szekvencia diagramján destruktork nem szerepel, emellett így a `Team.clear()` és `Map.clear()` metódusoknak nincsen szekvencia diagramja.

7.0.3.1 Player moves from Pipe to Active

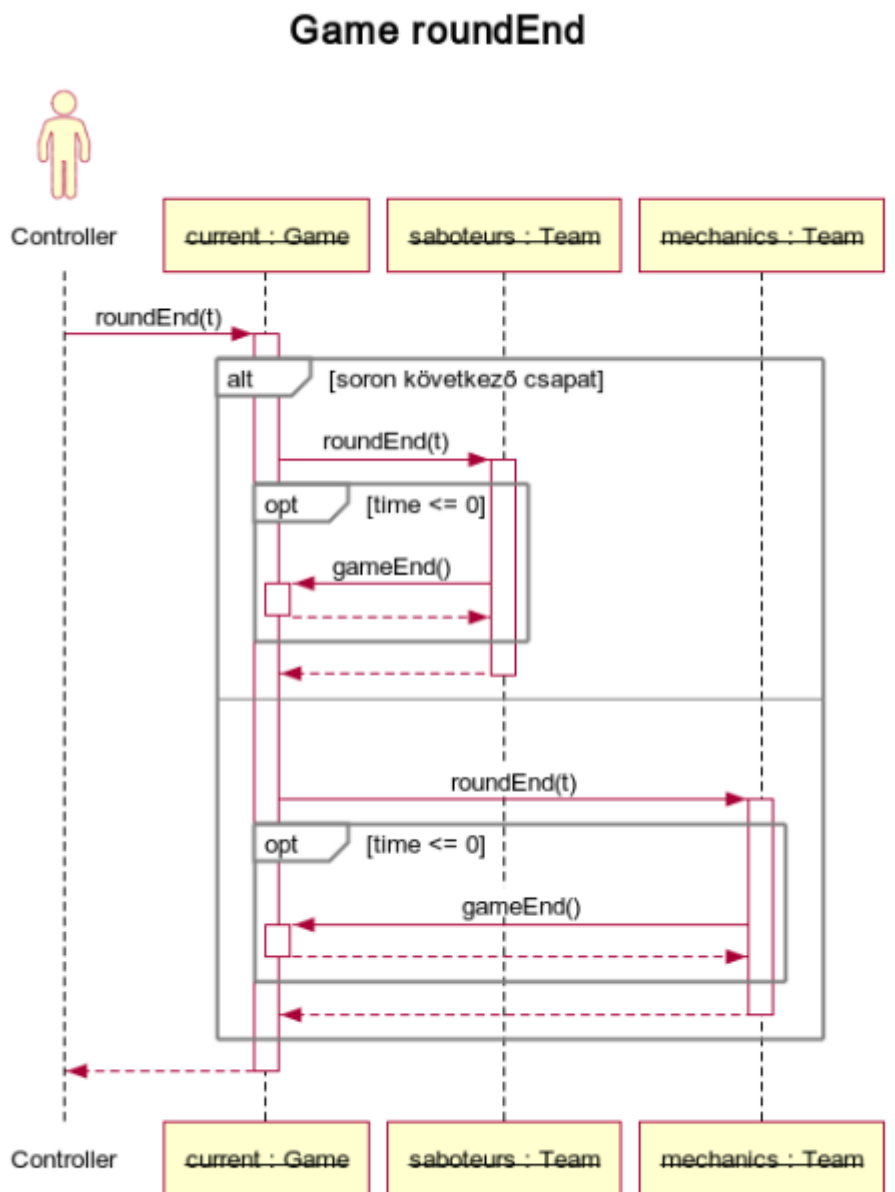


7.0.3.2 Player moves from Active to Pipe

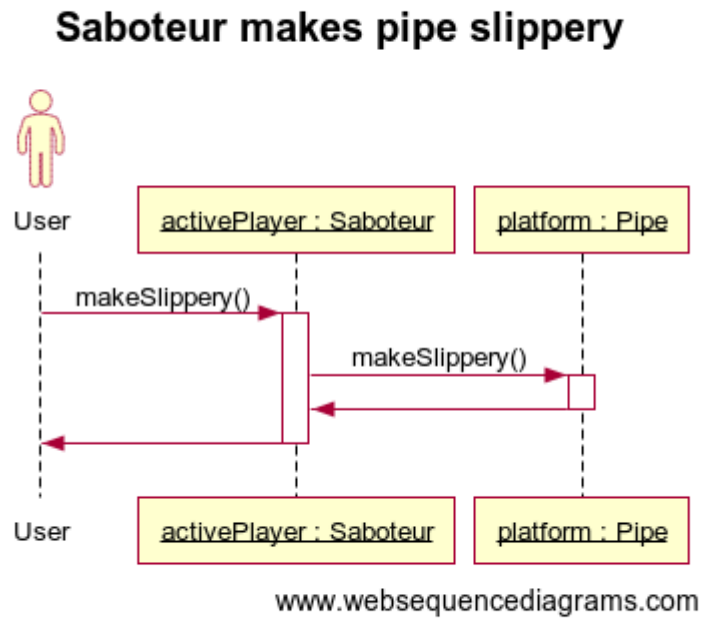


7.0.3.3 Game clear

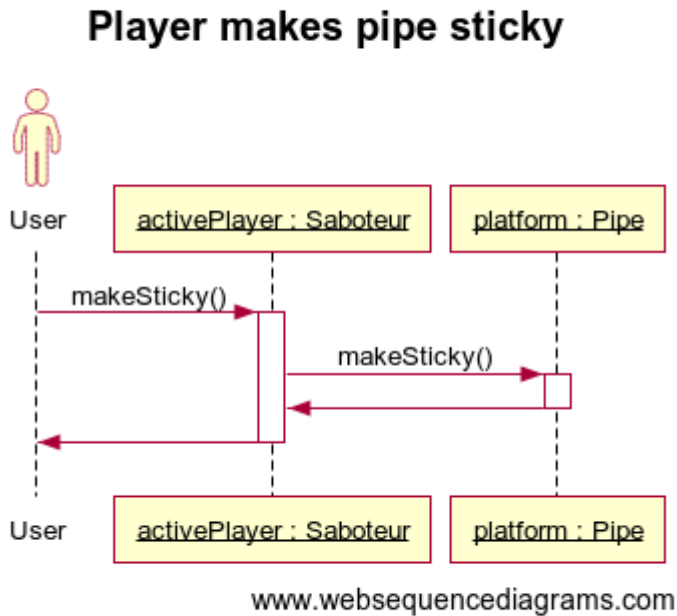
7.0.3.4 Game roundEnd



www.websequencediagrams.com

7.0.3.5 Saboteur make pipe slippery

7.0.3.6 Player makes Pipe sticky



7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A prototípus program szöveges interfészen keresztül vezérelhető.

A parancsok futtatása után a játék állapota konfigurációs fájlba menthető. A mentett konfiguráció betölthető. A konfigurációs fájl java szerializációval jön létre, annak formátuma manuális szerkesztésre nem alkalmas.

7.1.2 Bemeneti nyelv

A prototípus nevet ad a játék objektumainak. Ezzel a névvel hivatkozhatunk rájuk a kiadott parancsokban

```
create_spring <point> <name>
```

Leírás: Létrehoz egy forrást a megadott pontban.

```
create_pump <point> <int> <name>
```

Leírás: Létrehoz egy pumpát a megadott pontban. A második paraméter a csatlakoztatható csövek számát határozza meg.

```
create_cistern <point> <name>
```

Leírás: Létrehoz egy ciszternát a megadott pontban.

```
create_pipe <active> <active> <name>
```

Leírás: Létrehoz egy csövet a két megadott aktív elem közt.

```
create_mechanic <platform> <name>
```

Leírás: Létrehoz egy szerelőt a megadott platformon.

```
create_saboteur <platform> <name>
```

Leírás: Létrehoz egy szabotőrt a megadott platformon

```
setPump <player> <boolean>
```

Leírás: A megadott játékos beállítja a pumpát. A második paraméter azt jelzi, hogy a bemenetet állítjuk-e

```
putPump <player>
```

Leírás: A megadott játékos megpróbál lerakni egy pumpát.

```
takePump <player>
```

Leírás: A megadott játékos megpróbál felvenni egy pumpát.

```
putPipe <player>
```

Leírás: A megadott játékos megpróbálja lerakni a nála lévő csövet.

```
repair <player>
```

Leírás: A megadott játékos megpróbálja megjavítani a platformot, amin áll.

```
leakPipe <player>
```

Leírás: A megadott játékos megpróbálja kilyukasztani a platformot, amin áll.

```
takePipe <player> <pipe>
```

Leírás: A megadott játékos megpróbálja felvenni a megadott csövet.

```
move <player> <platform>
```

Leírás: A megadott játékos megpróbál a megadott platformra lépni.

```
makeSlippery <player>
```

Leírás: A megadott játékos megpróbálja csúszóssá tenni a platformot, amin áll.

```
makeSticky <player>
```


Leírás: A megadott játékos megpróbálja ragadóssá tenni a platformot, amin áll.

step <platform>

Leírás: A megadott platform állapotának frissítése.

step_all

Leírás: Minden platform állapotának frissítése

round_end <int>

Leírás: Ennek hatására a játékban véget ér a kör. A paraméter a játékban eltelt időt határozza meg.

randomize <boolean>

Leírás: Ki/Be kapcsolja a random számok generálását.

random_default <int>

Leírás: A paraméter meghatározza, hogy kikapcsolt randomizáció esetén milyen számot adjon vissza a random szám generátor.

save <file>

Leírás: A játék állapotát fájlba menti.

load <file>

Leírás: A játék állapotát fájlból betölti.

clear

Leírás: A játékot alapállapotba állítja.

stat

Leírás: Információkat ír ki a játék objektumairól.

run <file>

Leírás: A megadott fájlban leírt tesztet futtatása.

7.1.3 Kimeneti nyelv

A parancsok csak hibaüzeneteket írhatnak ki. Sikeres lefutás esetén a kimenet üres. Kivétel a stat parancs, ami a teljes játék állapotát írja ki. Az egyes objektumokra az alábbi formában:

Pipe: <name>

end1: <name>

end2: <name>

```

water: <int>
intact: <boolean>
shieldCounter: <int>
slipperyCounter: <int>

```

```

Pump: <name>
pipes: <name> <name> ...
intact: <boolean>
maxPipes: <int>
water: <int>
input: <name>
output: <name>

```

```

Spring: <name>
pipes: <name> <name> ...

```

```

Cistern: <name>
pipes: <name> <name> ...
hasPump: <boolean>

```

```

Mechanic: <name>
platform: <name>
pickedUpPipe: <name>
hasPump: <boolean>

```

```

Saboteur: <name>
platform: <name>

```

Itt a <name> a létrehozáskor megadott névre utal. Ennek megvalósításához lesz egy Named segédosztály, ami az elnevezhető objektumok közös őse.

7.2 Összes részletes use-case

Egy cső egyszerre nem lehet csúszós és ragadós, a ragadóssá / csúszóssá tétel mindig felülírja a cső állapotát (makeSlippery -> makeSticky = ragadós lesz a cső).

Use-case neve	Create Spring
Rövid leírás	Forrás létrehozása adott helyen
Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> 1. Forrás létrehozása 2. Forrás adott helyen való elhelyezése

Use-case neve	Create Pump
Rövid leírás	Pumpa létrehozása adott helyen
Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> 1. Pumpa létrehozása 2. Pumpa adott helyen való elhelyezése

Use-case neve	Create Cistern
Rövid leírás	Ciszterna létrehozása adott helyen
Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> 1. Ciszterna létrehozása 2. Ciszterna adott helyen való elhelyezése

Use-case neve	Create Pipe
Rövid leírás	Cső létrehozása adott helyen
Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> 1. Cső létrehozása 2. Cső adott helyen való elhelyezése

Use-case neve	Create Mechanic
Rövid leírás	Szerelő játékos létrehozása adott helyen
Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> 1. Szerelő játékos létrehozása 2. Szerelő játékos adott helyen való elhelyezése

Use-case neve	Create Saboteur
Rövid leírás	Szabotőr játékos létrehozása adott helyen
Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> 1. Szabotőr játékos létrehozása 2. Szabotőr játékos adott helyen való elhelyezése

Use-case neve	Set Pump
Rövid leírás	Pumpa beállítása
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. Pumpa beállítása

Use-case neve	Put Pump
Rövid leírás	Pumpa lerakása
Aktorok	Player
Forgatókönyv	<ol style="list-style-type: none"> 1. Pumpa lerakása
Alternatív Forgatókönyv	<ol style="list-style-type: none"> 1.A.1 Pumpa lerakása sikertelen, mert a játékos nem csövön áll 1.B.1 Pumpa lerakása sikertelen, mert a pumpa nem fér el

Use-case neve	Take Pump
Rövid leírás	Pumpa felvétele
Aktorok	Player

Forgatókönyv	1. Pumpa felvétele
Alternatív forgatókönyv	1.A.1 Pumpa felvétele sikertelen, mert szabotőr szeretne pumpát felvenni

Use-case neve	Put Pipe
Rövid leírás	Cső lerakása
Aktorok	Player
Forgatókönyv	1. Cső lerakása
Alternatív Forgatókönyv	1.A.1 Cső lerakása sikertelen, mert az nem fér el

Use-case neve	Repair
Rövid leírás	A platform megjavítása, amin a játékos áll
Aktorok	Player
Forgatókönyv	1. Platform megjavítása
Alternatív forgatókönyv	1.A.1 A platform megjavítása sikertelen, mert már eddig is meg volt javítva 1.B.1. A javítás sikertelen, mert szabotőr szeretne platformot javítani

Use-case neve	Leak Pipe
Rövid leírás	Cső kilyukasztása
Aktorok	Player
Forgatókönyv	1. Cső kilyukasztása
Alternatív forgatókönyv	1.A.1 A cső kilyukasztása sikertelen, mert nem telt el megfelelő mennyiségű idő, a legutóbbi kilyukasztás óta. 1.B.1 A cső kilyukasztása sikertelen, mert már eddig is ki volt lyukasztva. 1.C.1 A cső kilyukasztása sikertelen, mert nem szabotőr próbálja kilyukasztani

Use-case neve	Take pipe
Rövid leírás	Cső felvétele
Aktorok	Player
Forgatókönyv	1. Cső felvétele
Alternatív forgatókönyv	1.A.1 A cső felvétele sikertelen, mert a játékos egy szabotőr

Use-case neve	Move
Rövid leírás	A játékos lépése
Aktorok	Player
Forgatókönyv	1. A platform kiválasztása 2. A lépés megkísérlése az adott platformra

Alternatív forgatókönyv	2.A.1 A lépés sikeres 2.B.1 A lépés sikertelen 2.B.1.A.1 Az adott csővön már van játékos, oda nem lehet mozogni 2.B.1.B.1 Az adott platformra való mozgás nem lehetséges 2.B.1.C.1 A cső csúszóssága miatt, a játékos a cső valamelyik végéhez kapcsolódó elemre kerül 2.B.1.D.1 A cső ragadóssága miatt nem lehet onnan továbblépni
--------------------------------	---

Use-case neve	Make Slippery
Rövid leírás	A cső csúszóssá tétele
Aktorok	Player
Forgatókönyv	1. A játékos csúszóssá teszi a csövet egy rövid időre
Alternatív Forgatókönyv	1.A.1 A játékos a csövet nem tudja csúszóssá tenni, mert szerelő

Use-case neve	Make Sticky
Rövid leírás	A cső ragadóssá tétele
Aktorok	Player
Forgatókönyv	1. A játékos sikeresen ragadóssá teszi a csövet

Use-case neve	Step
Rövid leírás	Az adott platform állapotának frissítése
Aktorok	Controller
Forgatókönyv	1. A platform állapota frissül

Use-case neve	Step All
Rövid leírás	Minden platform állapotának frissítése
Aktorok	Controller
Forgatókönyv	1. Minden állapot frissül

Use-case neve	Round End
Rövid leírás	A játék egy köre véget ér
Aktorok	Controller
Forgatókönyv	1. A kör véget ér 2. Új kör kezdődik
Alternatív Forgatókönyv	2.A.1 A játék véget ért, nem kezdődik új kör

Use-case neve	Randomize
----------------------	-----------

Rövid leírás	Random szám generátor kikapcsolása illetve bekapcsolása.
Aktorok	Controller
Forgatókönyv	1. A randomizáció bekapcsolása
Alternatív Forgatókönyv	1.A.1 A randomizáció kikapcsolása

Use-case neve	Random Default
Rövid leírás	Kikapcsolt random generátor esetén az alapértelmezett érték.
Aktorok	Controller
Forgatókönyv	1. Az érték beállítása

Use-case neve	Save
Rövid leírás	A játék állapotának fájlba mentése
Aktorok	Player, Controller
Forgatókönyv	1. A játék fájlba mentése
Alternatív Forgatókönyv	1.A.1 A mentés sikertelen

Use-case neve	Load
Rövid leírás	A játék állapotának fájlból betöltése
Aktorok	Player, Controller
Forgatókönyv	1. A játék betöltése
Alternatív Forgatókönyv	1.A.1 A betöltés sikertelen

Use-case neve	Clear
Rövid leírás	A játék alapállapotba állítása
Aktorok	Controller
Forgatókönyv	1. Az alapállapotba állítás sikeres

Use-case neve	Stat
Rövid leírás	Az adott objektum adatainak lekérdezése
Aktorok	Controller
Forgatókönyv	1. Adatok kiírása 2. Adatok megtekintése

Use-case neve	Run
Rövid leírás	A megadott fájlban leírt tesztet futtatása. A kimenet a standard outputra íródik.
Aktorok	Controller

Forgatókönyv	<ol style="list-style-type: none"> 1. A fájl tartalmának beolvasása. 2. Script futtatása. 3. Eredmény kiírása.
Alternatív Forgatókönyv	Hiba esetén a hibaüzenet is kiíródik, a script futtatása megszakad, a run parancs visszatér.

7.3 Tesztelési terv

Teszt-eset neve	1.a Szerelő foglalt csőre lép
Rövid leírás	A szerelő egy aktív elemről egy foglalt csőre próbál lépni. Sikeres teszt esetén ez nem teljesül.
Teszt célja	Teszteli, hogy a cső és az aktív elemek közti mozgás helyessége megfelelően biztosított-e.

Teszt-eset neve	1.b. A Szerelő szabad csőre lép.
Rövid leírás	A szerelő egy aktív elemről egy szabad csőre próbál lépni. Sikeres teszt esetén ez teljesül.
Teszt célja	Teszteli, hogy a cső és az aktív elemek közti mozgás megfelelően biztosított-e.

Teszt-eset neve	1.c. A Szerelő egy csúszós csőre lép.
Rövid leírás	A szerelő egy aktív elemről egy csúszós csőre lép. Sikeres teszt esetén a szerelő a cső valamelyik végpontjához kapcsolt aktív elemre kerül.
Teszt célja	Teszteli, hogy a cső és az aktív elemek közti mozgás megfelelően biztosított-e.

Teszt-eset neve	1.d. A Szerelő egy ragadós csőre lép
Rövid leírás	A szerelő egy aktív elemről egy ragadós csőre lép. Sikeres teszt esetén a szerelő ezt követően egy ideig nem tud mozogni.
Teszt célja	Teszteli, hogy a cső és az aktív elemek közti mozgás megfelelően biztosított-e.

Teszt-eset neve	2.a. A Szabotőr egy foglalt csőre lép.
Rövid leírás	A szabotőr egy aktív elemről egy foglalt csőre próbál lépni. Sikeres teszt esetén ez nem teljesül.
Teszt célja	Teszteli, hogy a cső és az aktív elemek közti mozgás helyessége megfelelően biztosított-e.

Teszt-eset neve	2.b A Szabotőr egy szabad csőre lép.
Rövid leírás	A szabotőr egy aktív elemről egy szabad csőre próbál lépni. Sikeres teszt esetén ez teljesül.

Teszt célja	Teszteli, hogy a cső és az aktív elemek közti mozgás helyessége megfelelően biztosított-e.
--------------------	--

Teszt-eset neve	2.c A Szabotőr egy csúszós csőre lép.
Rövid leírás	A szabotőr egy aktív elemről egy csúszós csőre próbál lépni. Sikeres teszt esetén a szabotőr a cső valamelyik végpontjához kapcsolt aktív elemre kerül.
Teszt célja	Teszteli, hogy a cső és az aktív elemek közti mozgás helyessége megfelelően biztosított-e.

Teszt-eset neve	2.d A Szabotőr egy ragadós csőre lép.
Rövid leírás	A szerelő egy aktív elemről egy ragadós csőre lép. Sikeres teszt esetén a szabotőr ezt követően egy ideig nem tud mozogni.
Teszt célja	Teszteli, hogy a cső és az aktív elemek közti mozgás helyessége megfelelően biztosított-e.

Teszt-eset neve	3. A Szerelő egy aktív elemre lép.
Rövid leírás	A Szerelő egy csőről egy aktív elemre lép. Sikeres teszt esetén ez teljesül.
Teszt célja	Teszteli, hogy a cső és az aktív elemek közti mozgás helyessége megfelelően biztosított-e.

Teszt-eset neve	4. A Szabotőr egy aktív elemre lép.
Rövid leírás	A Szabotőr egy csőről egy aktív elemre lép. Sikeres teszt esetén ez teljesül.
Teszt célja	Teszteli, hogy a cső és az aktív elemek közti mozgás helyessége megfelelően biztosított-e.

Teszt-eset neve	5.a A Szabotőr egy olyan csövet lyukaszt ki, amin engedélyezve van a lyukasztás művelete.
Rövid leírás	A Szabotőr egy olyan csövön áll, amelyen a lyukasztás megkezdésének pillanatában engedélyezve van a lyukasztás művelete. Sikeres teszt esetén ki tudja lyukasztani a csövet.

Teszt célja	Teszteli, hogy a cső konzisztensen viselkedik-e, azaz ilyen esetben kilyukasztható-e.
--------------------	---

Teszt-eset neve	5.b A Szabotőr egy olyan csövet lyukaszt ki, amin nincs engedélyezve a lyukasztás művelete.
Rövid leírás	A Szabotőr egy olyan csövön áll, amelyen a lyukasztás megkezdésének pillanatában nincs engedélyezve a lyukasztás művelete. Sikeres teszt esetén nem tudja kilyukasztani a csövet.
Teszt célja	Teszteli, hogy a cső konzisztensen viselkedik-e, azaz ilyen esetben kilyukasztható-e.

Teszt-eset neve	6.a A Szerelő egy olyan csövet lyukaszt ki, amin engedélyezve van a lyukasztás művelete.
Rövid leírás	A Szerelő egy olyan csövön áll, amelyen a lyukasztás megkezdésének pillanatában engedélyezve van a lyukasztás művelete. Sikeres teszt esetén ki tudja lyukasztani a csövet.
Teszt célja	Teszteli, hogy a cső konzisztensen viselkedik-e, azaz ilyen esetben kilyukasztható-e

Teszt-eset neve	6.b A Szerelő egy olyan csövet lyukaszt ki, amin nincs engedélyezve a lyukasztás művelete.
Rövid leírás	A Szerelő egy olyan csövön áll, amelyen a lyukasztás megkezdésének pillanatában nincs engedélyezve a lyukasztás művelete. Sikeres teszt esetén nem tudja kilyukasztani a csövet.
Teszt célja	Teszteli, hogy a cső konzisztensen viselkedik-e, azaz ilyen esetben kilyukasztható-e.

Teszt-eset neve	7.a A Szabotőr minden pálya elemet javít
Rövid leírás	A Szabotőr megpróbál javítani egy-egy csövet, pumpát, ciszternát és forrást amin áll. A teszt akkor sikeres, ha egyik sem teljesül.
Teszt célja	Teszteli, hogy a szabotőrnek, tud-e javítani.

Teszt-eset neve	7.b A Szerelő megjavítja a lyukas csövet.
Rövid leírás	A Szerelő megpróbálja megjavítani azt a csövet amin éppen áll. A cső amit megpróbál megjavítani kezdetben lyukas, a teszt akkor sikeres, ha a cső a művelet után nem lyukas.
Teszt célja	Teszteli hogy a Szerelő meg tudja-e javítani a lyukas csövet.

Teszt-eset neve	7.c A Szerelő megjavítja az ép csövet.
Rövid leírás	A Szerelő megpróbálja megjavítani azt a csövet amin áll. A teszt elején a cső nem lyukas és a teszt végeztével is épnek kell maradnia. Ekkor a teszt sikeres.
Teszt célja	Teszteli hogy a Szerelő meg tudja-e javítani az ép csövet.

Teszt-eset neve	7.d A Szerelő megjavít egy elromlott pumpát
Rövid leírás	A Szerelő megpróbálja megjavítani azt a pumpát amin áll. A pumpa kezdetben hibás a művelet után pedig épnek kell lennie. Ha ez teljesült, a teszt sikeres.
Teszt célja	Teszteli, hogy a Szerelő tud-e pumpát javítani.

Teszt-eset neve	7.e A Szerelő megjavít egy ép pumpát
Rövid leírás	A Szerelő megpróbál megjavítani egy olyan pumpát ami kezdetben is ép, a művelet után is épnek kell maradnia. Amennyiben ez teljesül, a teszt sikeres.
Teszt célja	Teszteli, hogy a Szerelő tud-e ép pumpát javítani.

Teszt-eset neve	7.e A Szerelő javít ciszternát és forrást.
Rövid leírás	A szerelő megpróbál megjavítani egy forrást és utána egy ciszternát. Ha egyik sem sikerül, nem történik változás az állapotában, akkor a teszt sikeres.
Teszt célja	Teszteli, hogy a Szerelő tud-e ciszternát vagy forrást javítani.

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A program parancssorból futtatható és csak a szöveges interfészt használja. A teszt scriptek fájlból olvasása és a kimenet fájlba írása a program standard input/outputjának átirányításával és a run paranccsal lehetséges.

A prototípus tesztelése egy PowerShell szkript futtatásával elérhető. A szkript egy paramétert vár, a teszteset nevét, ez után lefuttatja a tesztet és összehasonlítja a kapott kimenetet az elvárt kimenettel.

Minden teszthez két fájl szükséges, az bemeneti fájl “<teszt_név>.in” és az elvárt kimeneti fájl: “<teszt_név>.exp”. Ezek mellé a program generál egy “<teszt_név>.out” fájlt, amit összehasonlít a hozzá tartozó .exp fájlal.

A szkript minden teszt futása után kiírja annak sikerességét vagy sikertelenségét és ha sikertelen a teszt, kiírja a kimenet és az elvárt kimenet közti különbséget.

Napló

Kezdet	Időtartam	Résztevők	Leírás
2023.04.21 22:00	3 óra	Gegő	Változtatások dokumentálása, osztálydiagram szerkesztése.
2023.04.22 10:00	1,5 óra	Gegő	Kimeneti/Bemeneti nyelv leírása.
2023.04.22 12:00	1 óra	Kovács Gegő	Az osztálydiagram átbeszélése, új és megváltozó metódusok megbeszélése, tervek
2023.04.22. 17:00	1 óra	Koleszár	7.2 tervezésének elkezdése
2023.04.22 18:00	1,5 óra	Kovács	Új és megváltozó metódusok leírása (7.0.2)
2023.04.22 23:00	0.5 óra	Kovács Gegő	7.0.1 és 7.0.2 átbeszélése, javítások
2023.04.23 11:00	2 óra	Kovács	Szekvencia diagramok az új és megváltozó metódusokhoz (7.0.3)
2023.04.23. 11:00	2 óra	Koleszár	7.2 Use-Case-ek véglegesítése, a táblázatok kitöltése
2023.04.23 14:30	1,5 óra	Szautner	7.2 Use-Case átnézése, 7.3 Tesztelési terv írása.
2023.04.23 23:30	0,5 óra	Kovács	7.2 Use-Casek átnézése, javítások
2023.04.24 0:00	1 óra	Gegő	Dokumentum átnézése, 7.1 és 7.2 kisebb módosítása, 7.4 újraírása.
2023.04.24 19:00	0,5 óra	Tóth	7.4 Tesztelést támogató segédprogramok.
2023.04.24 10:00	2 óra	Tóth, Szautner	7.3 Tesztesetek bővítése, javítása.

8. Részletes tervek

8.0.1 Prototípus kimeneti nyelv kiegészítése (7.1.3)

A következővel egészítjük ki az eddigi, prototípusban megadott kimeneteket:

```
Round: <int>
mechanics: <name> <name> ...
mechanicTeamTimeLeft: <int>
mechanicTeamPoints: <int>
saboteurs: <name> <name> ...
saboteurTeamTimeLeft: <int>
saboteurTeamPoints: <int>
lastActivePlayer: <name>
```

8.1 Osztályok és metódusok tervei.

8.1.1 Game

- **Felelősség**

A teljes játékra vonatkozó információkat tartalmazó osztály. Statikus megvalósítású, így mindenhol könnyen elérhető. Számolja a köröket és kiválasztja a soron következő játékost.

- **Attribútumok**

- - **static Map map:** a pálya amin a játék játszódni fog
- - **static Team mechanics:** a szerelők csapata
- - **static Team saboteurs:** a szabotőrök csapata
- - **static boolean end:** Vége van-e a játéknak.
- - **static int roundCounter:** A kör száma.

- **Metódusok**

- + **static void clear():** Törli a csapatokat és a pályát.
- + **static void roundEnd(int t):** A kör végén levonja az időt a megfelelő csapattól. A játék végén beállítja az "end" tagváltozót true-ra, ezzel jelezve, hogy a játék véget ért.
- + **static gameEnd():** Befejezi a játékot és kihirdeti a győztes csapatot.
- + **static pointMechanic(int p):** Ad p pontot a szerelők csapatának.
- + **static pointSaboteur(int p):** Ad p pontot a szabotőrök csapatának.

8.1.2 Map

- **Felelősség**

A pálya platformjait csoportosítja

- **Attribútumok**

- - **List<Active> actives**: A pályán elhelyezkedő aktív elemek listája
- - **List <Pipe> pipes**: A pálya csöveinek listája.

- **Metódusok**

- + **void clear()**: Kitörli az összes platformot.
- + **Pipe getPipe(String name)**: Név alapján kikeres egy csövet és visszaadja. Ha nincs találat, akkor null-t ad.
- + **Active getActive(String name)**: Név alapján kikeres egy aktív elemet és visszaadja. Ha nincs találat, akkor null-t ad.
- + **boolean pumpOk(Point p)**: Visszaadja, hogy az adott pontba lerakható-e egy pumpa. (Elfér-e a többi elem mellett.)
- + **boolean pipeOk(Point p1, Point p2)**: Visszaadja, hogy az adott pontok között lerakható-e egy cső. (Elfér-e a többi elem mellett.)

8.1.3 Team

- **Felelősség**

Egy csapat tagjátékosait csoportosítja. Számon tartja a pontokat és a hátralévő időt. A kör végén, ha a csapat ideje lejárt, értesíti a statikus Game objektumot a játék végétől.

- **Attribútumok**

- - **int points**: A csapat pontjai.
- - **int time**: A csapat ideje.
- - **List<Player> players**: A csapathoz tartozó játékosokat tároló lista.

- **Metódusok**

- + **void clear()**: Kitörli az összes játékost.
- + **Player getPlayer(String name)**: Név alapján kikeres egy játékost és visszaadja. Ha nincs találat, akkor null-t ad.
- + **void addPoint(int p)**: Pontot ad a csapatnak.
- + **void roundEnd(int t)**: A kör végén levonja az időt a csapattól. Ha elfogyott az idő, akkor Game.gameEnd()-et hív, ezzel jelzi, hogy vége a játéknak.

8.1.4 Platform

- **Felelősség**

A játékban a csőrendszer egy elemét reprezentálja. A platformok egymáshoz kapcsolódnak, a játékosok a platformokon mozognak és azokkal különböző műveleteket tudnak végezni. Teljesen absztrakt osztály, mert az elemek viselkedése nagy mértékben eltér.

- **Attribútumok**

- **# List<Player> players:** A platformon álló játékosok listája. Ennek kezelése is az alosztályok feladata.

- **Metódusok**

- **+ abstract void step():** Az elem állapotát frissíti, ezzel az idő múlását szimulálja.
- **+ abstract void repair():** Megjavítja, ha meg lehet javítani.
- **+ abstract void leak():** Kilyukasztja, ha ki lehet lyukasztani.
- **+ abstract void setPump(boolean in):** Beállítja a pumpa kimenetét vagy bemenetét. Az in paraméter dönti el, hogy melyiket.
- **+ abstract boolean isConnected(Platform p):** Visszaadja, hogy a paraméterként kapott platform csatlakozik-e ehhez.
- **+ abstract boolean pumpOk(Point p):** Visszaadja, hogy az adott pontban elér-e egy pumpa a platform mellett.
- **+ abstract boolean pipeOk(Point p1, Point p2):** Visszaadja, hogy az adott pontok közt elér-e egy cső a platform mellett.
- **+ abstract boolean putPipe(Pipe p):** Az adott csövet rácsatlakoztatja a platformra. Visszaadja, hogy a rátétel sikeres volt-e.
- **+ abstract void pickUpPipe(Pipe p):** Az adott csövet lecsatlakoztatja a platformról.
- **+ abstract boolean putPump():** Megpróbál rárakni egy pumpát a platformra. Visszatérési értékében jelzi, hogy sikerült-e.
- **+ abstract boolean hasPump():** Visszaadja, hogy a platformon felvehető-e pumpa.
- **+ abstract boolean canEnter():** Visszaadja, hogy egy játékos ráléphet-e.
- **+ abstract boolean canLeave():** Visszaadja, hogy egy játékos elhagyhatja-e a platformot.
- **+ abstract void playerIn(Player p):** A p játékos beléptetése. Hozzáadja a játékost a listához.
- **+ abstract void playerOut(Player p):** A p játékos kiléptetése. Kiveszi a játékost a listából.
- **+ abstract void makeSlippery():** Csúszóssá teszi a platformot, ha lehetséges.
- **+ abstract void makeSticky():** Ragadóssá teszi a platformot, ha lehetséges.

8.1.5 Pipe

- **Felelősség**

A csőbe víz rakása, és csőből víz kivétele. A cső felvétele és lerakása.

- **Ősosztályok**

Platform

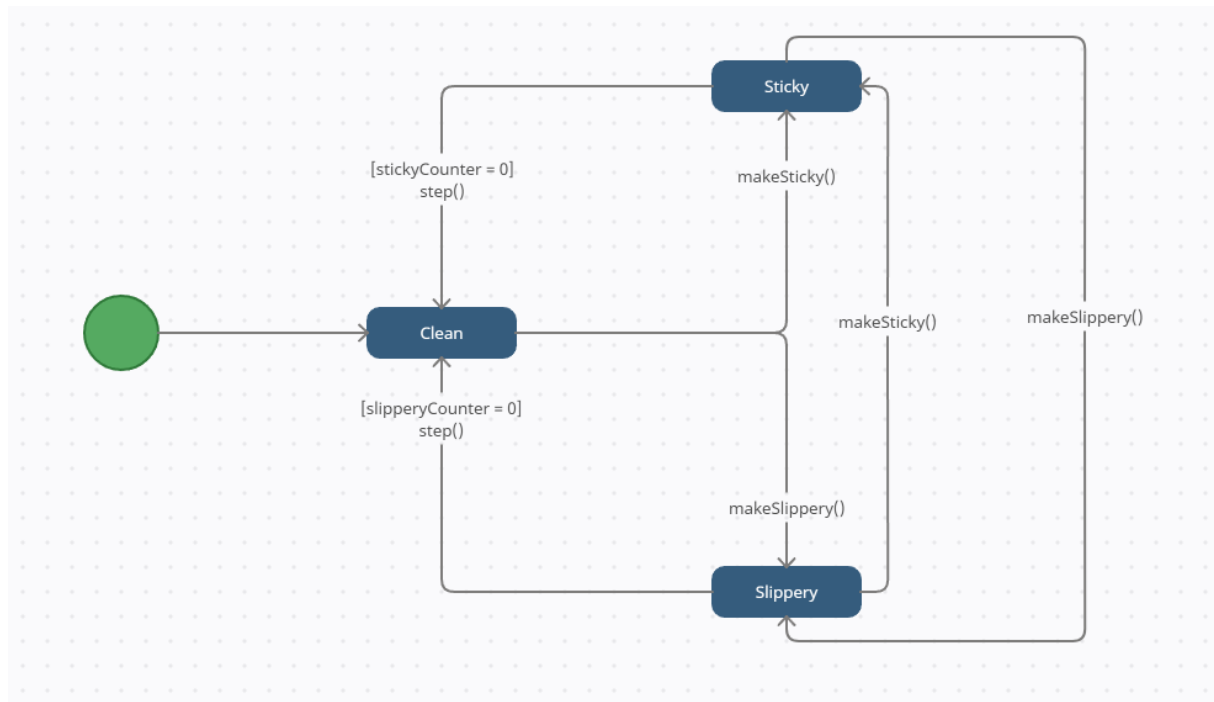
- **Attribútumok**

- - **static int capacity**: a csövek kapacitása
- - **int water**: a csőben lévő víz mennyisége
- - **boolean intact**: megadja, hogy a cső ép-e
- - **int shieldCounter**: számláló, hogy még hány tick-ig nem lyukasztható a cső
- - **int stickyCounter**: számláló, hogy még mennyi ideig ragadós a cső
- - **int slipperyCounter**: számláló, hogy még mennyi ideig csúszós a cső
- - **Active end1**: az az aktív elem, amelyhez a cső egyik vége csatlakozik
- - **Active end2**: az az aktív elem, amelyhez a cső másik vége csatlakozik

- **Metódusok**

- + **int pullWater(int v)**: a csőből min(v, water) mennyiségű vizet szív ki. Visszaadja a kivett víz mértékét.
- + **int pushWater(int v)**: a csőbe min(v, (capacity - water)) vizet tesz be. Visszaadja a betett víz értékét. Ha a cső ki van lyukadva, akkor a beleteendő v víz elfolyik. Az elfolyt vízmennyiségnek megfelelően ad pontot a szabotőrök csapatának.
- + **pickUp(Active a)**: A paraméterként megadott aktív elemről felvevődik a cső.
- + **putDown(Active a)**: A paraméterként megadott aktív elemre letevődik a cső.
- + **void step()**: A cső állapotát frissíti, ezzel az idő múlását szimulálja. Minden lépésben a megfelelő nem 0 értékű számlálók csökkennek 1-el.
- + **void repair()**: A cső megjavul. Ekkor beállítódik a shieldCounter a megfelelő nem 0 értékre.
- + **void leak()**: A cső kilyukad. Ekkor azonnal minden víz kifolyik belőle és az így elfolyt vízmennyiségnek megfelelően ad pontokat a szabotőrök csapatának. Amíg a shieldCounter nem 0 addig a csövet nem lehet kilyukasztani.
- + **void setPump(boolean in)**: Üres.
- + **boolean isConnected(Platform p)**: Visszaadja, hogy a paraméterként kapott platform csatlakozik-e ehhez.
- + **boolean pumpOk(Point p)**: Visszaadja, hogy az adott pontban elfér-e egy pumpa a platform mellett.
- + **boolean pipeOk(Point p1, Point p2)**: Visszaadja, hogy az adott pontok közt elfér-e egy cső a platform mellett.
- + **boolean putPipe(Pipe p)**: Mindig false-t ad vissza.
- + **void pickUpPipe(Pipe p)**: Üres.
- + **boolean putPump()**: Megpróbál rárakni egy pumpát a csőre. Visszatérési értékében jelzi, hogy sikerült-e.
- + **boolean hasPump()**: Mindig false-t ad vissza.
- + **boolean canEnter()**: Visszaadja, hogy a játékos ráléphet-e a csőre.
- + **boolean canLeave()**: Visszaadja, hogy egy játékos elhagyhatja-e a csövet. Ezt akkor nem teheti meg, ha a cső ragadós.

- + **void playerIn(Player p):** A p játékos beléptetése. Ha a cső csúszós a belépő játékos átkerül a cső egy random végén lévő aktív elemre.
- + **void playerOut(Player p):** A p játékos kiléptetése.
- + **void makeSlippery():** A cső csúszóssá válik. Itt beállítódik a slipperyCounter a megfelelő nem 0 értékre.
- + **void makeSticky():** A cső ragadóssá válik. Itt beállítódik a stickyCounter a megfelelő nem 0 értékre.



8.1.6 Active

- **Felelősség**

Aktív elemet reprezentáló abstract osztály. Az aktív elemekre a játékosok mindig léphetnek.

- **Ősosztályok**

Platform

- **Attribútumok**

- # **List<Pipe> pipes:** Az aktív elembe csatlakozó csöveket tároló lista
- # **Point pos:** Az aktív elem pozíciója a pályán

- **Metódusok**

- + **void repair():** Üres.
- + **void leak():** Üres.
- + **void setPump(boolean in):** Üres.
- + **boolean isConnected(Platform p):** Visszaadja, hogy a paraméterként kapott platform csatlakozik-e ehhez.

- + **boolean pumpOk(Point p)**: Visszaadja, hogy az adott pontban elér-e egy pumpa az aktív elem mellett.
- + **boolean pipeOk(Point p1, Point p2)**: Visszaadja, hogy az adott pontok közt elér-e egy cső az aktív elem mellett.
- + **boolean putPipe(Pipe p)**: Az adott csövet rácsatlakoztatja az aktív elemre.
- + **void pickUpPipe(Pipe p)**: Az adott csövet lecsatlakoztatja az aktív elemről.
- + **boolean putPump()**: Mindig false-t ad vissza.
- + **boolean hasPump()**: Mindig false-t ad vissza.
- + **boolean canEnter()**: Visszaadja, hogy egy játékos ráléphet-e. (Mindig true-t ad vissza)
- + **boolean canLeave()**: Visszaadja, hogy egy játékos elhagyhatja-e az aktív elemet. (Mindig true-t ad vissza)
- + **void playerIn(Player p)**: A p játékos beléptetése.
- + **void playerOut(Player p)**: A p játékos kiléptetése.
- + **void makeSlippery()**: Üres.
- + **void makeSticky()**: Üres.

8.1.7 Cistern

- **Felelősség**

Ciszternát reprezentáló osztály. Vizet juttat a rendszerbe és pumpákat lehet felvenni róla.

- **Össztályok**

Platform→Active

- **Attribútumok**

- - **boolean hasPump**: megadja, hogy a ciszternán van-e pumpa

- **Metódusok**

- + **void step()**: Minden bekötött csőből kiszív egy egység vizet. A kiszívott vízmennyiségnek megfelelően ad a szerelők csapatának pontot. Bizonyos időközönként pumpa “terem” a ciszternán.
- + **override boolean hasPump()**: Visszaadja, hogy éppen van-e pumpa a ciszternán.

8.1.8 Pump

- **Felelősség**

Pumpát reprezentáló osztály. Vizet mozgat az input és output csövek között. Véletlen időközönként elromlik. Meghatározott számú cső lehet rácsatlakoztatva.

- **Össztályok**

Platform→Active

- **Attribútumok**

- - **static int capacity**: a pumpák kapacitása

- - **boolean intact:** megadja, hogy a pumpa ép-e
- - **int maxPipes:** megadja, hogy a pumpára maximum hány cső csatlakozhat
- - **int water:** a pumpa belső tankjában lévő vízmennyiség
- - **Pipe input:** a pumpa bemeneti csöve, innen fog vizet kiszívni
- - **Pipe output:** a pumpa kimeneti csöve, ide fog vizet pumpálni
- **Metódusok**
 - + **void step():** Ha van hely a tartályában, akkor megpróbál a bemeneti csőből átrakni egy egységnyi vizet. Ha ezután van víz a tartályban (nem feltétlen a jelen step során kiszívott víz), akkor megpróbál egy egységnyit átrakni a kimeneti csőbe.
 - + **override void repair():** A pumpa megjavul.
 - + **override void setPump(boolean in):** Az in paramétertől függően átállítódik a pumpa ki- vagy bemenete. In = True → bemenet állítódik, In = False → kimenet állítódik.
 - + **override boolean putPipe(Pipe p):** Az adott csövet rácsatlakoztatja a pumpára, ha az elfér rajta és beállítja a pumpát. Visszaadja, hogy a rátétel sikeres volt-e.
 - + **override void pickUpPipe(Pipe p):** Az adott csövet lecsatlakoztatja a pumpáról és beállítja a pumpát.

8.1.9 Spring

- **Felelősség**

Forrást reprezentáló osztály. Csöveket hoz létre

- **Össztályok**

Platform→Active

- **Attribútumok**

- - **static int maxPipes:** megadja, hogy a forrásokra maximum hány cső csatlakoztatható

- **Metódusok**

- + **void step():** Minden bekötött csőbe egy egység vizet pumpál.

8.1.10 Player

- **Felelősség**

Egy Játékost reprezentál. A játékos a csőrendszer elemein mozog, a platformokkal különböző műveleteket végez. A játékosok szerelők vagy szabotőrök, a Player ezek abstract osztálya.

- **Attribútumok**

- # **Platform platform:** A platform amin a játékos áll.

- **Metódusok**

- + **void setPump(boolean in):** Megpróbálja beállítani a pumpát amin áll. A paramétertől függően a bemenetet/kimenetet állítja.
- + **abstract void putPump():** Megpróbál lerakni egy pumpát, ha megteheti.
- + **abstract void takePump():** Megpróbál felvenni egy pumpát, ha megteheti.
- + **abstract void putPipe():** Megpróbál lerakni egy csövet, ha megteheti.
- + **abstract void repair():** Megpróbálja megjavítani a platformot, amin áll, ha megteheti.
- + **void leakPipe():** Megpróbálja kilyukasztani a csövet amin áll.
- + **abstract void takePipe(Pipe p):** Megpróbál felvenni egy csövet, ha megteheti.
- + **void move(Platform p):** Megpróbál az adott platformra lépni.
- + **abstract void makeSlippery():** Megpróbálja csúszóssá tenni a platformot, amin áll, ha megteheti.
- + **void makeSticky():** Megpróbálja ragadóssá tenni a platformot, amin áll.

8.1.11 Mechanic

- **Felelősség**

A szerelő által végezhető műveleteket valósítja meg.

- **Össztályok**

Player

- **Attribútumok**

- - **Pipe pickedUpPipe:** a szerelő által felvett cső, ha van
- - **boolean hasPump:** megadja, hogy a szerelőnél van-e pumpa

- **Metódusok**

- + **void putPump():** Megpróbál lerakni egy pumpát a platformra, amin áll.
- + **void takePump():** Megpróbál felvenni egy csövet.
- + **void putPipe():** Megpróbálja lerakni a csövet, ha van nála.
- + **void repair():** Megpróbálja megjavítani a platformot, amin áll.
- + **void takePipe(Pipe p):** Megpróbálja felvenni a paraméterként adott csövet.
- + **void makeSlippery():** Üres.

8.1.12 Saboteur

- **Felelősség**

A szabotőr által végezhető műveleteket valósítja meg.

- **Össztályok**

Player

- **Metódusok**

- + **void putPump():** Üres.

- + **void takePump():** Üres.
- + **void putPipe():** Üres.
- + **void repair():** Üres.
- + **void takePipe(Pipe p):** Üres.
- + **void makeSlippery():** Csúszóssá teszi a platformot, amin áll.

8.1.13 Point

- **Felelősség**

A pályán egy pontot meghatározó segédosztály.

- **Attribútumok**

- - **int x:** helyvektor vízszintes komponense pixelekben mérve
- - **int y:** helyvektor függőleges komponense pixelekben mérve

- **Metódusok**

- + **static Point avg(Point p1, Point p2):** Visszaadja a két paraméterként kapott pont által alkotott szakasz felezőpontját. Ez is Point típusú.

8.1.14 Named

- **Felelősség**

Nevet tároló segédosztály.

- **Attribútumok**

- - **static int sequence:** az osztály utolsó példányának sorszámát tárolja. Ezzel tudja biztosítani a saját nevének egyediségét.
- - **String name:** az objektum neve

- **Metódusok**

8.1.15 Randomizer

- **Felelősség**

Random számok generálását végzi. A véletlenszerűség ki/be kapcsolható.

- **Attribútumok**

- - **static Random rd:** a Random osztály példánya, random szám generátor
- - **static boolean on:** megadja, hogy a véletlenszám generátor be van-e kapcsolva
- - **static int defaultVal:** kikapcsolt állapotban a véletlen számot helyettesítő alapértelmezett érték

- **Metódusok**

- **+ static getInt(int max):** Ha a véletlenszám generátor be van kapcsolva visszaad egy véletlen számot [0, max) intervallumon. Ha nincs bekapcsolva, akkor az alapértelmezett értéket adja vissza.

8.1.16 CmdParser

- **Felelősség**

A parancsok értelmezéséért és futtatásáért felelős osztály.

- **Metódusok**

- **private static Point parsePoint(String str):** Stringből Point-ot csinál, ha a formátum megfelelő. A megfelelő formátum: (X,Y)
- **private static boolean isNatural(String str):** Visszaadja, hogy a paraméterként kapott String egy nem negatív Integer-t ír-e le.
- **private static Named getNamed(String name):** Név alapján megkeresi a megfelelő objektumot (bármilyen Platform / Player).
- **private static Player getPlayer(String name):** Név alapján megkeresi a megfelelő Playert.
- **private static Platform getPlatform(String name):** Név alapján megkeresi a megfelelő Platformot.
- **public static void start(InputStream is):** Elkezd a megadott InputStream parse-olását.
- **private static void create_”X”(String[] args):** Létrehozza a megfelelő objektumot. A létrehozandó objektum neve áll mindig az X helyén. Az argumentumlista (args[]) az előző dokumentumban specifikált argumentumokat várja string-ek formájában. A függvények kivételt nem dobnak, de jelzik a felhasználónak, ha az argumentumlista hibás.
 - o **private static void create_spring(String[] args)**
 - o **private static void create_pump(String[] args)**
 - o **private static void create_cistern(String[] args)**
 - o **private static void create_pipe(String[] args)**
 - o **private static void create_mechanic(String[] args)**
 - o **private static void create_saboteur(String[] args)**
- **private static void ”X”(String[] args):** Elvégzi a megfelelő objektumokon a megfelelő műveleteket és vagy változtatásokat. Az argumentumlista (args[]) az előző dokumentumban specifikált argumentumokat várja string-ek formájában. A függvények kivételt nem dobnak, de jelzik a felhasználónak, ha az argumentumlista hibás.
 - o **private static void setPump(String[] args) :** Az adott játékos beállít egy pumpát
 - o **private static void putPump(String[] args) :** Az adott játékos lerak egy pumpát.
 - o **private static void takePump(String[] args) :** Az adott játékos felvesz egy pumpát.
 - o **private static void putPipe(String[] args) :** Az adott játékos lerak egy csövet.

- o **private static void repair(String[] args):** Az adott játékos megjavít egy pumpát vagy csövet.
- o **private static void leakPipe(String[] args):** Az adott játékos kilyukaszt egy csövet.
- o **private static void takePipe(String[] args):** Az adott játékos felvesz egy csövet.
- o **private static void move(String[] args):** Az adott játékos mozog.
- o **private static void makeSlippery(String[] args):** Az adott játékos egy csövet csúszóssá tesz.
- o **private static void makeSticky(String[] args):** Az adott játékos egy csövet ragadóssá tesz.
- o **private static void step(String[] args) :** Az adott platform léptetése.
- o **private static void step_all(String[] args):** Minden platform léptetése.
- o **private static void round_end(String[] args) :** A kört befejezi, az adott mennyiségű időt levonja.
- **private static void randomize(String[] args):** Az első argumentum alapján beállítja a véletlenszám generátor állapotát (bekapcsolt / kikapcsolt).
- **private static void random_default(String[] args):** Az első argumentum alapján beállítja a véletlenszám generátor alapértelmezett értékét
- **private static void save(String[] args):** Menti az argumentum listában megadott nevű fájlba a játékot.
- **private static void load(String[] args):** Betölti az argumentum listában megadott nevű fájlból a játékot.
- **private static void clear(String[] args):** Visszaállítja a játék állapotát. Törli a platformokat és a játékosokat.
- **private static void stat(String[] args):** Kiírja a megfelelő információkat.
- **private static void run(String[] args):** Rekurzívan elindítja a parancsok értelmezését az argumentumban megadott fájlból.

Használt algoritmusok pszeudokódjai:

```

Lineáris keresés, arra, hogy egy objektum eleme-e egy megfelelő kollekciónak:
    for minden objektum a tömbben:
        ha megfelelő feltétel teljesül akkor
            visszatér az objektummal
        visszatér null

```

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

8.2.1 Teszteset1

- **Leírás**

Órán megadott teszteset.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szerelő mozog, Szabotőr mozog, Szabotőr csövet lyukaszt, Szabotőr csövet ragadóssá tesz, pumpa véletlenszerűen elromlik, Szerelő pumpát javít, Szerelő csövet javít. Várható hibahely, hogy a függvények bármelyike nem megfelelően működik.

- **Bemenet**

```

create_cistern (1,5) c
create_pump (3,5) 3 p1
create_pump (5,7) 3 p2
create_pump (5,3) 3 p3

```

```
create_spring (7,3) f1
create_spring (7,7) f2
```

```
create_pipe c p1 cs1
create_pipe p1 p3 cs2
create_pipe p1 p2 cs3
create_pipe p3 p2 cs4
create_pipe p2 f2 cs5
create_pipe p3 f1 cs6
```

```
create_saboteur c s
create_mechanic p3 m
```

```
move s cs1
move s p1
move s cs3
move s p2
move s cs5
makeSticky s
move s p2
move s cs3
leakPipe s
move s p2
```

```
randomize false
random_default 1
step p1
randomize true
```

```
move m cs2
move m p1
repair m
move cs3
repair m
```

```
stat
```

● Elvárt kimenet

```
Round: 1
mechanics: m
mechanicTeamTimeLeft: 280
mechanicTeamPoints: 20
saboteurs: s
saboteurTeamTimeLeft: 280
saboteurTeamPoints: 20
lastActivePlayer: m
```

```
Cistern: c
pipes: cs1
hasPump: false
```

```
Pump: p1
pipes: cs1 cs2 cs3
intact: true
maxPipes: 3
water: 0
input: cs1
output: cs2
```

```
Pump: p2
pipes: cs3 cs4 cs5
intact: true
maxPipes: 3
water: 0
input: cs3
output: cs4
```


Pump: p3
pipes: cs2 cs4 cs6
intact: true
maxPipes: 3
water: 0
input: cs2
output: cs4

Spring: f1
pipes: cs6

Spring: f2
pipes: cs5

Pipe: cs1
end1: c
end2: p1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs2
end1: p1
end2: p3
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs3
end1: p1
end2: p2
water: 0
intact: true
shieldCounter: 0
stickyCounter: 10
slipperyCounter: 0

Pipe: cs4
end1: p3
end2: p2
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs5
end1: p2
end2: f2
water: 0
intact: true
shieldCounter: 10
stickyCounter: 0
slipperyCounter: 0

Pipe: cs6
end1: p3
end2: f1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Saboteur: s
platform: p2

Mechanic: m
platform: cs3
pickedUpPipe: null
hasPump: false

8.2.2 Teszteset2

- **Leírás**

Szerelő megjavít egy Szabotőr által kilyukasztott csövet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szerelő mozog, Szabotőr mozog, Szerelő csövet javít, Szabotőr csövet lyukaszt. Várható hibahely, hogy a függvények bármelyike nem megfelelően működik.

- **Bemenet**

create_cistern (1,1) c1
create_pump (1,3) 2 p1
create_pump (1,5) 2 p2
create_spring (1,7) f1
create_pipe c1 p1 cs1
create_pipe p1 p2 cs2
create_pipe p2 f1 cs3
create_saboteur p2 s1
create_mechanic p1 m1

move s1 cs2
leakPipe s1
move s1 p1

move m1 cs2
repair m1
step cs2
move m1 p2

stat

- **Elvárt kimenet**

Round: 1
mechanics: m1
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 10
saboteurs: s1
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 10
lastActivePlayer: m1

Pipe: cs1
end1: c1
end2: p1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs2
end1: p1
end2: p2
water: 10
intact: true
shieldCounter: 10
stickyCounter: 0
slipperyCounter: 0

Pipe: cs3
end1: p2
end2: f1
water: 10
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pump: p1
pipes: cs1 cs2
intact: true
maxPipes: 2
water: 0
input: cs2
output: cs1

Pump: p2
pipes: cs2 cs3
intact: true
maxPipes: 2
water: 0
input: cs3
output: cs2

Spring: f1
pipes: cs3

Cistern: c1
pipes: cs1
hasPump: false

Mechanic: m1
platform: p2
pickedUpPipe: null
hasPump: false

Saboteur: s1
platform: p1

8.2.3 Teszteset3

- **Leírás**

Szerelő megkísérel megjavítani egy ép csövet, aminek hatására semmi sem történik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szerelő mozog, Szerelő csövet javít. Várható hibahely, hogy az ép cső javításának hatására semminek sem kell történnie, illetve, hogy a függvények bármelyike nem megfelelően működik.

- **Bemenet**

create_cistern (1,1) c1
create_pump (1,3) 2 p1
create_pump (1,5) 2 p2
create_spring (1,7) f1
create_pipe c1 p1 cs1
create_pipe p1 p2 cs2
create_pipe p2 f1 cs3
create_mechanic p1 m1

move m1 cs2
repair m1
step cs2
move m1 p2

stat

- **Kimenet**

Round: 1
mechanics: m1
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: null
saboteurTeamTimeLeft: null
saboteurTeamPoints: null
lastActivePlayer: m1

Pipe: cs1
end1: c1
end2: p1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs2
end1: p1
end2: p2
water: 10
intact: true
shieldCounter: 10
stickyCounter: 0
slipperyCounter: 0

Pipe: cs3
end1: p2
end2: f1
water: 10
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pump: p1
pipes: cs1 cs2
intact: true
maxPipes: 2
water: 0
input: cs2
output: cs1

Pump: p2
pipes: cs2 cs3
intact: true
maxPipes: 2
water: 0
input: cs3
output: cs2

Spring: f1
pipes: cs3

Cistern: c1
pipes: cs1
hasPump: false
Mechanic: m1
platform: p2
pickedUpPipe: null
hasPump: false

8.2.4 Teszteset4

- **Leírás**

Szabotőr, illetve szerelő mozgása minden fajta szabad, aktív elemre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szerelő mozog, Szabotőr mozog. Várható hibahely, hogy a függvények bármelyike nem megfelelően működik.

- **Bemenet**

```
create_cistern (1,1) c1
create_pump (1,3) 2 p1
create_spring (1,5) f1
create_pipe c1 p1 cs1
create_pipe p1 f1 cs2
create_mechanic p1 m1
create_saboteur p1 s1
```

```
move s1 cs2
move s1 f1
move s1 cs2
move s1 p1
move s1 cs1
move s1 c1
```

```
move m1 cs2
move m1 f1
move m1 cs2
move m1 p1
move m1 cs1
move m1 c1
```

```
stat
```

- **Elvárt kimenet**

```
Round: 1
mechanics: m1
mechanicTeamTimeLeft: 270
mechanicTeamPoints: 0
saboteurs: s1
saboteurTeamTimeLeft: 270
saboteurTeamPoints: 0
lastActivePlayer: m1
```

```
Pipe: cs1
end1: c1
end2: p1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0
```

```
Pipe: cs2
end1: p1
end2: f1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0
```

```
Pump: p1
pipes: cs1 cs2
intact: true
maxPipes: 2
water: 0
```

input: cs1
output: cs2

Spring: f1
pipes: cs2

Cistern: c1
pipes: cs1
hasPump: false

Mechanic: m1
platform: c1
pickedUpPipe: null
hasPump: false

Saboteur: s1
platform: c1

8.2.5 Teszteset5

- **Leírás**

Szabotőr foglalt csőre lép.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szabotőr mozog. Várható hibahely, hogy egyszerre egy csövön csak 1 játékos tartózkodhat, illetve, hogy a függvények bármelyike nem megfelelően működik.

- **Bemenet**

create_cistern (1,1) c1
create_pump (1,3) 2 p1
create_spring (1,5) f1
create_pipe c1 p1 cs1
create_pipe p1 f1 cs2
create_mechanic p1 m1
create_saboteur p1 s1

move m1 cs1
move s1 cs1

stat

- **Kimenet**

Round: 1
mechanics: m1
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: null
saboteurTeamTimeLeft: null
saboteurTeamPoints: null
lastActivePlayer: m1

Pipe: cs1
end1: c1
end2: p1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs2
end1: p1
end2: f1
water: 0
intact: true

shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pump: p1
pipes: cs1 cs2
intact: true
maxPipes: 2
water: 0
input: cs1
output: cs2

Spring: f1
pipes: cs2

Cistern: c1
pipes: cs1
hasPump: false

Mechanic: m1
platform: cs1
pickedUpPipe: null
hasPump: false

Saboteur: s1
platform: p1

8.2.6 Teszteset6

- **Leírás**

Szerelő foglalt csőre lép.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szerelő mozog. Várható hibahely, hogy egyszerre egy csövön csak 1 játékos tartózkodhat, illetve, hogy a függvények bármelyike nem megfelelően működik.

- **Bemenet**

create_cistern (1,1) c1
create_pump (1,3) 2 p1
create_spring (1,5) f1
create_pipe c1 p1 cs1
create_pipe p1 f1 cs2
create_mechanic p1 m1
create_saboteur p1 s1

move s1 cs1
move m1 cs1

stat

- **Kimenet**

Round: 1
mechanics: m1
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: s1
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: m1

Pipe: cs1
end1: c1
end2: p1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0

slipperyCounter: 0

*Pipe: cs2
end1: p1
end2: f1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0*

*Pump: p1
pipes: cs1 cs2
intact: true
maxPipes: 2
water: 0
input: cs1
output: cs2*

*Spring: f1
pipes: cs2*

*Cistern: c1
pipes: cs1
hasPump: false*

*Mechanic: m1
platform: p1
pickedUpPipe: null
hasPump: false*

*Saboteur: s1
platform: cs1*

8.2.7 Teszteset7

- **Leírás**

Szerelő, illetve szabotőr is ragadós csőre lép, és onnan nem tudnak továbblépni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szerelő mozog, Szabotőr mozog, Ragadós csőre lépés. Szabotőr mozog. Várható hibahely, hogy a csövek ragadóssá tétele után a játékosok onnan nem tudnak továbblépni, illetve, hogy a függvények bármelyike nem megfelelően működik.

- **Bemenet**

*create_cistern (1,1) c1
create_pump (1,3) 2 p1
create_pump (1,5) 2 p2
create_spring (1,7) f1
create_pipe c1 p1 cs1
create_pipe p1 p2 cs2
create_pipe p2 f1 cs3
create_mechanic p1 m1
create_saboteur p2 s1*

*move m1 cs1
makeSticky m1
move m1 c1*

*move s1 cs3
makeSticky s1
move s1 f1*

stat

- **Kimenet**

Round: 1
mechanics: m1
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: s1
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: s1

Pipe: cs1
end1: c1
end2: p1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 10
slipperyCounter: 0

Pipe: cs2
end1: p1
end2: p2
water: 10
intact: true
shieldCounter: 10
stickyCounter: 0
slipperyCounter: 0

Pipe: cs3
end1: p2
end2: f1
water: 10
intact: true
shieldCounter: 0
stickyCounter: 10
slipperyCounter: 0

Pump: p1
pipes: cs1 cs2
intact: true
maxPipes: 2
water: 0
input: cs2
output: cs1

Pump: p2
pipes: cs2 cs3
intact: true
maxPipes: 2
water: 0
input: cs3
output: cs2

Spring: f1
pipes: cs3

Cistern: c1
pipes: cs1
hasPump: false

Mechanic: m1
platform: cs1
pickedUpPipe: null
hasPump: false

Saboteur: s1
platform: cs3

8.2.8 Teszteset8

- **Leírás**

Szerelő, illetve Szabotőr csúszós csőre lépnek.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szerelő mozog, Szabotőr mozog, csúszós csőre lépnek. Várható hibahely, hogy a csövek csúszóssa tétele után a játékosok a cső valamely végéhez kerülnek véletlenszerűen, illetve, hogy a függvények bármelyike nem megfelelően működik.

- **Bemenet**

```
create_cistern (1,1) c1
create_pump (1,3) 2 p1
create_pump (1,5) 2 p2
create_spring (1,7) f1
create_pipe c1 p1 cs1
create_pipe p1 p2 cs2
create_pipe p2 f1 cs3
create_mechanic p1 m1
create_saboteur p2 s1
```

```
move m1 cs1
makeSlippery m1
randomizer false
random default 1
step cs1
randomizer true
```

```
move s1 cs3
makeSlippery s1
randomizer false
random default 0
step cs3
randomizer true
```

stat

- **Kimenet**

```
Round: 1
mechanics: m1
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: s1
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: s1
```

```
Pipe: cs1
end1: c1
end2: p1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 10
```

```
Pipe: cs2
end1: p1
end2: p2
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0
```

```
Pipe: cs3
end1: p2
end2: f1
```

water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 10

Pump: p1
pipes: cs1 cs2
intact: true
maxPipes: 2
water: 0
input: cs2
output: cs1

Pump: p2
pipes: cs2 cs3
intact: true
maxPipes: 2
water: 0
input: cs3
output: cs2

Spring: f1
pipes: cs3

Cistern: c1
pipes: cs1
hasPump: false

Mechanic: m1
platform: p1
pickedUpPipe: null
hasPump: false

Saboteur: s1
platform: p2

8.2.9 Teszteset9

- **Leírás**

Szerelő, és szabotőr is pumpát állítanak, aminek következtében bizonyos csövekbe víz jut. A pálya szerkezete az órainak megfelelő.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szerelő mozog, Szabotőr mozog, Szerelő pumpát állít, Szabotőr pumpát állít, víz folyik. Várható hibahely, hogy a függvények bármelyike nem megfelelően működik.

- **Bemenet**

create_cistern (1,5) c
create_pump (3,5) 3 p1
create_pump (5,7) 3 p2
create_pump (5,3) 3 p3
create_spring (7,3) f1
create_spring (7,7) f2

create_pipe c p1 cs1
create_pipe p1 p3 cs2
create_pipe p1 p2 cs3
create_pipe p3 p2 cs4
create_pipe p2 f2 cs5
create_pipe p3 f1 cs6

create_saboteur c s
create_mechanic p3 m

setPump m true

move s cs1
move s p1
move s cs3
move s p2
setPump s true

stat

● **Kimenet**

Round: 1
mechanics: m
mechanicTeamTimeLeft: 295
mechanicTeamPoints: 20
saboteurs: s
saboteurTeamTimeLeft: 275
saboteurTeamPoints: 0
lastActivePlayer: s

Cistern: c
pipes: cs1
hasPump: false

Pump: p1
pipes: cs1 cs2 cs3
intact: true
maxPipes: 3
water: 0
input: cs1
output: cs2

Pump: p2
pipes: cs3 cs4 cs5
intact: true
maxPipes: 3
water: 0
input: cs5
output: cs4

Pump: p3
pipes: cs2 cs4 cs6
intact: true
maxPipes: 3
water: 0
input: cs6
output: cs4

Spring: f1
pipes: cs6

Spring: f2
pipes: cs5

Pipe: cs1
end1: c
end2: p1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs2
end1: p1
end2: p3
water: 0
intact: true
shieldCounter: 0

stickyCounter: 0
slipperyCounter: 0

Pipe: cs3
end1: p1
end2: p2
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs4
end1: p3
end2: p2
water: 10
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs5
end1: p2
end2: f2
water: 10
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs6
end1: p3
end2: f1
water: 10
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Saboteur: s
platform: p2

Mechanic: m
platform: p3
pickedUpPipe: null
hasPump: false

8.2.10 Teszteset10

- **Leírás**

Pumpa véletlenszerűen elromlik

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Pumpa elromlik A pumpa véletlenszerű elromlását szimulálja és teszteli, hiba lehet, hogy a pumpa utána is működőképes lesz

- **Bemenet**

create_pump (1,1) 2 p

randomize false
random_default 1
step p
randomize true

stat

- **Kimenet**

Round: 1

mechanics:
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs:
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer:

Pump: *p*
pipes:
intact: *false*
maxPipes: 2
water: 0
input:
output:

8.2.11 Teszteset11

- **Leírás**

A Szabotőr egy olyan csövet lyukaszt ki, amin engedélyezve van a lyukasztás művelete.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Szabotőr kilyukasztja a csövet amin áll, ez után a csőnek hibásnak kell lennie

- **Bemenet**

create_cisztern (1,1) *c*
create_pump (1,3) 1 *p*
create_pipe *c* *p* *cs*
create_saboteur *cs* *s*

leakPipe *s*

stat

- **Kimenet**

Round: 1
mechanics:
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: *s*
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: *s*

Cistern: *c*
pipes: *cs*
hasPump: *false*

Pump: *p*
pipes: *cs*
intact: *true*
maxPipes: 1
water: 0
input: *cs*
output:

Pipe: *cs*
end1: *c*
end2: *p*
water: 0
intact: *false*
shieldCounter: 0
slipperyCounter:

Saboteur: *s*

platform: cs

8.2.12 Teszteset12

- **Leírás**

A Szerelő egy olyan csövet lyukaszt ki, amin engedélyezve van a lyukasztás művelete.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Szerelő kilyukasztja a csövet amin áll, ez után a csőnek hibásnak kell lennie

- **Bemenet**

create_cisztern (1,1) c
create_pump (1,3) 1 p
create_pipe c p cs
create_mechanic cs m

leakPipe m

stat

- **Kimenet**

Round: 1
mechanics: m
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs:
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: m

Cistern: c
pipes: cs
hasPump: false

Pump: p
pipes: cs
intact: true
maxPipes: 1
water: 0
input: cs
output:

Pipe: cs
end1: c
end2: p
water: 0
intact: false
shieldCounter: 0
slipperyCounter:

Mechanic: m
platform: cs
pickedUpPipe: null
hasPump: false

8.2.13 Teszteset13

- **Leírás**

A Szerelő egy olyan csövet lyukaszt ki, amin nincs engedélyezve a lyukasztás művelete.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Szerelő kilyukasztja a csövet amin áll, megjavítja, majd amíg még nem lehetne újra lyukasztani megpróbálja újra lyukasztani, hiba lehet, hogy sikerül neki.

- **Bemenet**

create_cisztern (1,1) c
create_pump (1,3) 1 p

```
create_pipe c p cs
create_mechanic cs m
```

```
randomize false
random_default 10
leakPipe m
repair m
leakPipe m
randomize true
```

```
stat
```

- **Kimenet**

```
Round: 1
mechanics: m
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs:
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: m
```

```
Cistern: c
pipes: cs
hasPump: false
```

```
Pump: p
pipes: cs
intact: true
maxPipes: 1
water: 0
input: cs
output:
```

```
Pipe: cs
end1: c
end2: p
water: 0
intact: true
shieldCounter: 10
slipperyCounter:
```

```
Mechanic: m
platform: cs
pickedUpPipe: null
hasPump: false
```

8.2.14 Teszteset14

- **Leírás**

A Szabotőr egy olyan csövet lyukaszt ki, amin nincs engedélyezve a lyukasztás művelete

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Szabotőr kilyukasztja a csövet amin áll, megjavítja, majd amíg még nem tudná újra kilyukasztani megpróbálja ezt újra megtenni, hiba lehet, hogy sikerül neki.

- **Bemenet**

```
create_cistern (1,1) c
create_pump (1,3) 1 p
create_pipe c p cs
create_saboteur cs s
create_mechanic c m
```

```
randomize false
random_default 10
```


leakPipe s
move s p
move m cs
repair m
move m c
move s cs
leakPipe s
randomize true

stat

• Kimenet

Round: 1
mechanics: m
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: s
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: s

Cistern: c
pipes: cs
hasPump: false

Pump: p
pipes: cs
intact: true
maxPipes: 1
water: 0
input: cs
output:

Pipe: cs
end1: c
end2: p
water: 0
intact: true
shieldCounter: 10
slipperyCounter:

Saboteur: s
platform: cs

Mechanic: m
platform: c
pickedUpPipe: null
hasPump: false

8.2.15 Teszteset15

• Leírás

A Szerelő egy elromlott pumpát javít.

• Ellenőrzött funkcionalitás, várható hibahelyek

A szerelő egy hibás pumpát javít, a művelet után a pumpának jónak kell lennie

• Bemenet

create_pump (1,3) 2 p
create_mechanic p m

randomize false
random_default 1
step p
randomize true

repair m

stat

- **Kimenet**

Round: 1

mechanics: m

mechanicTeamTimeLeft: 300

mechanicTeamPoints: 0

saboteurs:

saboteurTeamTimeLeft: 300

saboteurTeamPoints: 0

lastActivePlayer: m

Pump: p

pipes:

intact: true

maxPipes: 2

water: 0

input:

output:

Mechanic: m

platform: p

pickedUpPipe: null

hasPump: false

8.2.16 Teszteset16

- **Leírás**

A Szerelő javít ciszternát és forrást.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Szerelő ciszternát és forrást javít, egyikre sem lehet hatása.

- **Bemenet**

create_cisztern (1,1) c

create_spring (1,5) f

create_pump (1,3) 2 p

create_pipe c p cs1

create_pipe p f cs2

create_mechanic c m

repair m

move m cs1

move m p

move m cs2

move m f

repair m

stat

- **Kimenet**

Round: 1

mechanics: m

mechanicTeamTimeLeft: 300

mechanicTeamPoints: 0

saboteurs:

saboteurTeamTimeLeft: 300

saboteurTeamPoints: 0

lastActivePlayer: m

Cistern: c

pipes: cs1

hasPump: false

Spring: f
pipes: cs2

Pump: p
pipes: cs1 cs2
intact: true
maxPipes: 2
water: 0
input: cs1
output: cs2

Pipe: cs1
end1: c
end2: p
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs2
end1: p
end2: f
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Mechanic: m
platform: f
pickedUpPipe: null
hasPump: false

8.2.17 Teszteset

- **Leírás**

A Szabotőr minden pálya elemet javít

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szabotőr javít, Szabotőr mozog. Várható hibahely, valamely függvény nem megfelelő működése, illetve, hogy a Szabotőr nem javíthat.

- **Bemenet**

create_cisztern (1,1) c
create_spring (1,5) f
create_pump (1,3) 2 p
create_pipe c p cs1
create_pipe p f cs2
create_saboteur c s

repair s
move s cs1
repair s
move s p
repair p
move s cs2
move s f
repair s

stat

- **Kimenet**

Round: 1
mechanics:
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: s

saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: s

Cistern: c
pipes: cs1
hasPump: false

Spring: f
pipes: cs2

Pump: p
pipes: cs1 cs2
intact: true
maxPipes: 2
water: 0
input: cs1
output: cs2

Pipe: cs1
end1: c
end2: p
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs2
end1: p
end2: f
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Saboteur: s
platform: f

8.2.18 Teszteset18

- **Leírás**

A Szerelő illetve a Szabotőr egy-egy csövet ragadóssá tesznek.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szerelő csövet ragadóssá tesz, Szabotőr csövet ragadóssá tesz, Szerelő mozog, Szabotőr mozog. Várható hibahely bármelyik függvény nem megfelelő viselkedése.

- **Bemenet**

create_spring (1,1) sp
create_cistern(1,5) c
create_pump (1,3) p
create_pipe sp p1 cs1
create_pipe p1 c cs2
create_saboteur cs1 s
create_mechanic cs2 m

makeSticky s
move s p
makeSticky m
move m p
stat

- **Kimenet**

Round: 1
mechanics: m
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: s
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: m

Pump: p
pipes: cs1 cs2
intact: true
maxPipes: 2
water: 0
input: cs1
output: cs2

Spring: sp
pipes: cs1

Cistern: c
pipes: cs2
hasPump: false

Pipe: cs1
end1: sp
end2: p
water: 0
intact: true
shieldCounter: 0
stickyCounter: 9
slipperyCounter: 0

Pipe: cs2
end1: p
end2: c
water: 0
intact: true
shieldCounter: 0
stickyCounter: 9
slipperyCounter: 0

Saboteur: s
platform: p

Mechanic: m
platform: p
pickedUpPipe: null
hasPump: false

8.2.19 Teszteset19

- **Leírás**

A Szabotőr egy csövet csúszóssá tud tenni

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szabotőr csövet csúszóssá tesz. Várható hibahely valamelyik függvény nem megfelelő működése.

- **Bemenet**

create_spring (1,1) sp
create_cistern(1,5) c
create_pump (1,3) p
create_pipe sp p1 cs1
create_pipe p1 c cs2

create_saboteur cs1 s
create_mechanic cs2 m

makeSlippery s
move s p
stat

- **Kimenet**

Round: 1
mechanics: m
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: s
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: s

Pump: p
pipes:cs1 cs2
intact: true
maxPipes: 2
water: 0
input: cs1
output: cs2

Spring: sp
pipes: cs1

Cistern:c
pipes: cs2
hasPump: false

Pipe: cs1
end1: sp
end2: p
water: 0
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:9

Pipe: cs2
end1: p
end2: c
water: 0
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Saboteur: s
platform: p

Mechanic: m
platform: p
pickedUpPipe: null
hasPump: false

8.2.20 Teszteset20

- **Leírás**

Szerelő felvesz egy pumpát a ciszternáról, majd elhelyezi a csövön

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

```
create_spring (1,1) sp
create_cistern(1,5) c
create_pipe sp c cs1
create_mechanic cs1 m
create_saboteur c s
```

```
move m c
randomize false
random_default 0
step c
takePump m
move m cs1
putPump m
```

stat

- **Kimenet**

```
Round: 1
mechanics: m
mechanicTeamTimeLeft: <int>
mechanicTeamPoints: 0
saboteurs: s
saboteurTeamTimeLeft: <int>
saboteurTeamPoints: 0
lastActivePlayer: m
```

```
Spring: sp
pipes: cs1
```

```
Cistern:c
pipes: cs1
hasPump: false
```

```
Pipe: cs1
end1: sp
end2: p
water: 0
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0
```

```
Pipe: cs2
end1: p
end2: c
water: 0
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0
```

```
Pump: p1
pipes:cs1, cs2
```

intact: true
maxPipes: 2
water: 0
input: cs1
output: cs2

Saboteur: s
platform: c

Mechanic: m
platform: p
pickedUpPipe: null
hasPump: false

8.2.21 Teszteset21

- **Leírás**
Szerelő megpróbál felvenni pumpát a ciszternáról de nem tud
- **Ellenőrzött funkcionalitás, várható hibahelyek**
-

- **Bemenet**

create_spring (1,1) sp
create_cistern(1,3) c
create_pipe sp c cs1
create_mechanic cs1 m
create_saboteur c s

move m c
randomize false
random_default 1
step c
takePump m
move m cs1

stat

- **Kimenet**

Round: 1
mechanics: m
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: s
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: m

Cistern: c
pipes: cs1
hasPump: false

Spring: sp
pipes: cs1

Pipe: cs1
end1: sp
end2: c
water: 0

intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Saboteur: s
platform: c

Mechanic: m
platform: p
pickedUpPipe: null
hasPump: false

8.2.22 Teszteset22

- **Leírás**

A Szerelő felvesz egy pumpát viszont nem tudja lerakni

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Pumpát felvesz, Pumpát lerak. Hibahely, hogy bármelyik függvény nem megfelelően működik, illetve, hogy a pumpát nem lehet lerakni

- **Bemenet**

create_spring (1,1) sp
create_cistern(1,3) c
create_pipe sp c cs1
create_mechanic cs1 m
create_saboteur c s

move m c
randomize false
random_default 0
step c
takePump m
move m cs1
putPump m

stat

- **Kimenet**

Round: 1
mechanics: m
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: s
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: m

Cistern: c
pipes: cs1
hasPump: false

Spring: sp
pipes: cs1

Pipe: cs1
end1: sp
end2: c
water: 0
intact: true

shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Saboteur: s
platform: c

Mechanic: m
platform: cs1
pickedUpPipe: null
hasPump: true

8.2.23 Teszteset23

- **Leírás**

Egy kör véget ér, a játékosok mozognak, Szabotőr csövet lyukaszt, ennek következtében a csapata ponthoz jut.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szerelő mozog, Szabotőr mozog, kör véget ér, Szabotőr csövet lyukaszt. Várható hibahely, hogy a függvények bármelyike nem megfelelően működik.

- **Bemenet**

create_cistern (1,1) c1
create_pump (1,3) 2 p1
create_pump (1,5) 2 p2
create_spring (1,7) f1
create_pipe c1 p1 cs1
create_pipe p1 p2 cs2
create_pipe p2 f1 cs3
create_saboteur p2 s1
create_saboteur f1 s2
create_mechanic p1 m1
create_mechanic c1 m2

move m1 cs1
move m1 c1
move m2 cs1
move s1 cs2
leakPipe s1
move s2 cs3
move s2 p2

round_end 20

stat

- **Kimenet**

Round: 1
mechanics: m1 m2
mechanicTeamTimeLeft: 290
mechanicTeamPoints: 0
saboteurs: s1 s2...
saboteurTeamTimeLeft: 290
saboteurTeamPoints: 10
lastActivePlayer: s2

Pipe: cs1
end1: c1
end2: p1
water: 0
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pipe: cs2
end1: p1
end2: p2
water: 0
intact: false
shieldCounter: 10
stickyCounter: 0
slipperyCounter: 0

Pipe: cs3
end1: p2
end2: f1
water: 10
intact: true
shieldCounter: 0
stickyCounter: 0
slipperyCounter: 0

Pump: p1
pipes: cs1 cs2
intact: true
maxPipes: 2
water: 0
input: cs2
output: cs1

Pump: p2
pipes: cs2 cs3
intact: true
maxPipes: 2
water: 0
input: cs3
output: cs2

Spring: f1
pipes: cs3

Cistern: c1
pipes: cs1
hasPump: false

Mechanic: m1
platform: c1
pickedUpPipe: null
hasPump: false

Mechanic: m2
platform: cs1
pickedUpPipe: null
hasPump: false

Saboteur: s1
platform: cs2

Saboteur: s2
platform: p2

8.2.24 Teszteset24

- **Leírás**

Összetett teszt: mozgás, cső lyukasztás, pumpa kapcsolás, pumpa lerakás, cső ragadóssá tétele, cső csúszóssá tétele

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Mozgás, cső lyukasztás, pumpa kapcsolás, pumpa lerakás, cső ragadóssá tétele, cső csúszóssá tétele. Hibahely, hogy bármelyik függvény nem megfelelően működik.

- **Bemenet**

```
create_spring (1,1) sp1
create_spring (1,3) sp2
create_spring (1,5) sp3
create_cistern(10,5) c
create_pump (3,1) p1
create_pump (3,5) p2
```

```
create_pipe sp3 p2 cs1
create_pipe sp2 p2 cs2
create_pipe sp1 p1 cs3
create_pipe p2 c cs3
create_pipe p1 c cs5
create_pipe p1 p2 cs6
```

```
create_mechanic cs1 m
create_saboteur cs4 s
```

```
leakPipe m
makeSticky m
randomize false
random_default 0
move m p2
move m cs6
setPump m true
move m cs3
move m c
takePump m
move m cs3
putPump
move s p1
leakPipe s
makeSlippery s
move s cs6
```

```
stat
```

- **Kimenet**

```
Round: 1
mechanics: m
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: s
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 1
lastActivePlayer: m
```

```
Cistern: c
pipes: cs3, cs5
hasPump: false
```

```
Spring: sp1
pipes: cs4
```

```
Spring: sp2
pipes: cs2
```

```
Spring: sp3
pipes: cs1
```

Cistern: c
pipes: cs1

Pipe: cs1
end1: sp3
end2: p2
water: 0
intact: false
shieldCounter:0
stickyCounter:9
slipperyCounter:0

Pipe: cs2
end1: sp2
end2: p2
water: 0
intact: true
shieldCounter:0
stickyCounter:9
slipperyCounter:0

Pipe: cs3
end1: p2
end2: p3
water: 0
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Pipe: cs4
end1: sp1
end2: p1
water: 0
intact: false
shieldCounter:0
stickyCounter:0
slipperyCounter:5

Pipe: cs5
end1: p1
end2: c
water: 0
intact: false
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Pipe: cs6
end1: p1
end2: p2
water: 0
intact: false
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Pipe: cs7
end1: p3
end2: c

water: 0
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Pump: p1
pipes: cs4 cs5
intact: true
maxPipes: 3
water: 0
input: cs4
output: cs5

Pump:p2
pipes: cs1 cs2 cs3 cs6
intact: true
maxPipes: 4
water: 0
input: cs2
output: cs3

Pump:p3
pipes: cs3 cs7
intact: true
maxPipes: 2
water: 0
input: cs3
output: cs7

Saboteur: s
platform: p1

Mechanic: m
platform: cs6
pickedUpPipe: null
hasPump: false

8.2.25 Teszteset25

- **Leírás**

Több kör játszódik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Körök változásával kapcsolatos funkcionalitás tesztelése, hiba, ha nem működik valamelyik függvény helyesen.

- **Bemenet**

create_cistern (1,1) c
create_spring (3,7) f
create_pump (3,1) p1
create_pump (3,5) p2
create_pipe p1 p2 cs1
create_pipe c p1 cs2
create_pipe p2 f cs3
create_mechanic p1 m
create_saboteur p2 s

move m cs1

round_end

move m p2
move s cs1

round_end

move m cs1
move s p2

round_end

stat

- **Kimenet**

Round: 3
mechanics: m
mechanicTeamTimeLeft: 300
mechanicTeamPoints: 0
saboteurs: s
saboteurTeamTimeLeft: 300
saboteurTeamPoints: 0
lastActivePlayer: s

Pipe: cs1
end1: p1
end2: p2
water: 0
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Pipe: cs2
end1: c
end2: p1
water: 0
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Pipe: cs3
end1: p2
end2: f
water: 0
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Pump: p1
pipes: cs1 cs2
intact: true
maxPipes: 2
water: 0
input: cs1
output: cs2

Pump: p2
pipes: cs1 cs3
intact: true
maxPipes: 2
water: 0
input: cs3
output: cs1

Spring: f
pipes: cs3

Cistern: c
pipes: cs2

Saboteur: s
platform: p2

Mechanic: m
platform: cs1
pickedUpPipe: null
hasPump: false

8.2.26 Teszteset26

- **Leírás**

Komplex teszt: 1-1 Szerelő és Szabotőr mozog a csőhálózatban, mindketten csövet lyukasztanak, előbbi csövet szerel, csövet vesz fel ciszternánál és rak le pumpánál, illetve csövet csúszóssá tesz; míg utóbbi pumpát állít és csövet tesu ragadóssá.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

create_cistern (1,5) c1
create_pump (3,5) 3 p1
create_pump (5,7) 3 p2
create_pump (5,3) 3 p3
create_spring (7,3) f1
create_spring (7,7) f2

create_pipe c p1 cs1
create_pipe p3 p1 cs2
create_pipe p2 p1 cs3
create_pipe p2 p3 cs4
create_pipe f2 p2 cs5
create_pipe f1 p3 cs6

create_saboteur f1 s1
create_mechanic f2 m1

move m1 cs5
move s1 cs6
move m1 p2
move s1 p3
move m1 cs3
setPump s1 true
move m1 cs3
leakPipe m1
move s1 cs2
move s1 p1
move s1 cs3
leakPipe s1
repair m1

round_end


```

makeSlippery m1
randomizer false
random default 1
step cs1
randomizer true
move s1 p2
move s1 cs5
makeSticky s1
move m1 p1
move m1 cs1
move m1 c1
takePipe m1
move m1 cs1
move m1 p1
move m1 cs2
move m1 p3
putPipe m1

```

```
round_end
```

```
stat
```

● Kimenet

```

Round: 2
mechanics: m1
mechanicTeamTimeLeft: 225
mechanicTeamPoints: 10
saboteurs: s1
saboteurTeamTimeLeft: 250
saboteurTeamPoints: 10
lastActivePlayer: m1

```

```

Cistern:c1
pipes: cs1, cs7
hasPump: false

```

```

Spring: f1
pipes: cs6

```

```

Spring: 42
pipes: cs5

```

```

Pipe: cs1
end1: c1
end2: p1
water: 10
intact: true
shieldCounter:0
stickyCounter:9
slipperyCounter:0

```

```

Pipe: cs2
end1: p3
end2: p1
water: 10
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0

```

```
Pipe: cs3
```

end1: p2
end2: p1
water: 0
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Pipe: cs4
end1: p2
end2: p3
water: 10
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Pipe: cs5
end1: f2
end2: p2
water: 10
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Pipe: cs6
end1: f1
end2: p3
water: 0
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Pipe: cs7
end1: c1
end2: p3
water: 0
intact: true
shieldCounter:0
stickyCounter:0
slipperyCounter:0

Pump: p1
pipes: cs1 cs2 cs3
intact: true
maxPipes: 3
water: 0
input: cs2
output: cs1

Pump:p2
pipes: cs3 cs4 cs5
intact: true
maxPipes: 3
water: 0
input: cs5

output: cs4

Pump:p3

pipes: cs2 cs4 cs6 cs7

intact: true

maxPipes: 3

water: 0

input: cs4

output: cs2

Saboteur: s1

platform: cs5

Mechanic: m1

platform: cs7

pickedUpPipe: null

hasPump: false

8.3 A tesztelést támogató programok tervei

A tesztelés a bemeneti nyelvvel végezhető el, a teszt automatizálását és a kimenet ellenőrzését egy PowerShell szkript végzi, ami lefuttatja a paraméterként kapott tesztet.

A tesztet a teszt számával azonosítható és indítható. Ez után a szkript kiírja a teszt sorszámát, majd azt, hogy sikerült-e a teszt.

A tesztek egy bemeneti *<azonosító>.in* és egy *<azonosító>.exp* szöveges fájlból állnak, emellé a program generál egy *<azonosító>.out* fájlt. A szkript átirányítja a *<azonosító>.in* fájl tartalmát, majd összehasonlítja kimeneti fájlt és az elvárt kimenetet, ha egyezik, akkor a teszt sikeres. Ha nem egyeznek jelzi, hogy a teszt sikertelen és kiírja a valós és az elvárt kimenet közötti különbséget.

Továbbá a szkript futtatható egy megkülönböztetett *“all-tests”* paraméterrel, ebben az esetben az összes tesztet lefuttatja, minden tesztet egyesével, úgy mint amikor egy tesztet futtat, majd a végén kiírja, hogy hány teszt volt sikeres és hány tesztet futtatott összesen.

A szkript futtatásához Windows-on engedélyezni kell a szkriptek futtatását, ha ezt nem tettük meg korábban. Ezt PowerShell-ben megtehetjük a *“Set-ExecutionPolicy Unrestricted”* parancs futtatásával rendszergazdai jogokkal, majd az *“Y”* beírásával elfogadva a módosításokat.

8.3.1 Napló

Kezdet	Időtartam	Résztevők	Leírás
2023.04.28. 20:00	2 óra	Gegő Kovács	8.1 osztályok leírása
2023.04.30. 16:00	2 óra	Koleszár	8.2 Tesztesetek írásának elkezdése
2023.04.30. 21:00	1 óra	Gegő	Struktúra újratervezés
2023.04.30 23:00	1,5 óra	Gegő Kovács	Új és változott osztályok dokumentálása
2023.05.01. 10:30	1 óra	Szautner Tóth Koleszár	Értekezlet: 8.2 Tesztesetekkel kapcsolatos megbeszélés, tesztek felosztása, pontosítása
2023.05.01. 11:30	1,5 óra	Koleszár	8.2 További tesztesetek írása
2023.05.01. 14:00	2 óra	Gegő Kovács	8.1 osztályok leírásának befejezése
2023.05.01. 11:30	3 óra	Tóth	8.2 Tesztesetek írása
2023.05.01. 21:00	3 óra	Szautner	8.2 Tesztesetek írása, tesztesetek javítása
2023.05.02. 10:00	1,5 óra	Koleszár	8.2 Korábbi tesztesetek kiegészítése, javítása, újak írása
2023.05.02. 18:00	2,5 óra	Koleszár	8.2 Tesztesetek bővítése
2023.05.02 20:00	2 óra	Tóth	8.2 Tesztesetek javítása 8.3 A támogató programok leírása
2023.05.02. 21:00	1 óra	Koleszár	8.2 Tesztesetek bővítése
2023.05.02 22:30	1,5 óra	Szautner	8.2 Tesztek bővítése, javítása

10. Prototípus beadása

10.0 Prototípus kimeneti nyelv módosítása (7.1.3)

A következőket módosítottuk a kimenetben (ezáltal a korábban megfogalmazott kimenetek is módosításra kerülnek):

A **Cistern** esetében a kimenetből eltávolítottuk a *“hasPump: boolean”* sort.

A **Pipe** esetében a kimenethez hozzáadtuk a *“willStuck: boolean”* sort. Ez azt jelzi, hogy a következő játékos, aki rálép a csőre, az ragadni fog. Ez azért fontos, hogy ne az ragadjon rá a csőre, aki ragadóssá tette.

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Active.java	3028 B	2023. 04. 28. 17:07	Aktív elemet megvalósító ősosztály
Cistern.java	709 B	2023. 04. 28. 17:07	Ciszternát megvalósító osztály
Game.java	3176 B	2023. 04. 28. 17:07	Játékot megvalósító osztály
Map.java	3197 B	2023. 04. 28. 17:07	Pályát megvalósító osztály
Mechanic.java	1973 B	2023. 04. 28. 17:07	Szerelőt megvalósító osztály
Pipe.java	7936 B	2023. 04. 28. 17:07	Csővet megvalósító osztály
Platform.java	3059 B	2023. 04. 28. 17:07	Platformot megvalósító osztály
Player.java	2339 B	2023. 04. 28. 17:07	Játékost megvalósító osztály
Pump.java	2799 B	2023. 04. 28. 17:07	Pumpát megvalósító osztály
Saboteur.java	1295 B	2023. 04. 28. 17:07	Szabotőrt megvalósító osztály
Spring.java	615 B	2023. 04. 28. 17:07	Forrást megvalósító osztály
Team.java	1275 B	2023. 04. 28. 17:07	Csapatokat megvalósító osztály
Cmdparser.java	16956 B	2023. 04. 28. 17:07	Tesztesetek kezelése, megadott tesztet futtatása.
Main.java	261 B	2023. 04. 28. 17:07	A program belépési pontja.
Named.java	402 B	2023. 04. 28. 17:07	A Platform és a Játékos ősosztálya, a kiírásért felel.
Point.java	2214 B	2023. 04. 28. 17:07	Pontot megvalósító osztály
Randomizer.java	524 B	2023. 04. 28. 17:07	A játék randomnesságát kezelő osztály.

build.ps1	55 B	2023. 04. 28. 17:07	A program futtathatóvá fordítását végző script.
clear.ps1	90 B	2023. 04. 28. 17:07	A fordított programfájlok törlését végző script
run.ps1	21 B	2023. 04. 28. 17:07	A program futására a szkript.
test.ps1	809 B	2023.04.28. 17:07	A fordított programmal a tesztek futtatsása.
testX.in	1 KB	2023. 05. 14. 10:30	Az X. teszteset bemenete.
testX.out	3 KB	2023. 05. 14. 10:30	Az X. teszteset kimenete, a program futáskor felülírja.
testX.exp	3 KB	2023. 05. 14. 10:30	Az X. teszteset elvárt kimenete.

Megjegyzés: Az utolsó három sorban a X lehet bármilyen egész szám az [1, 25] intervallumon, így mind a három sor egyenként 25 fájlt jelent.

10.1.2 Fordítás

Ha powershell scripttel szeretnénk a programot fordítani, akkor először engedélyeznünk kell azokat. Adminisztrátorként indított powershellbe írjuk a következő parancsot:
set-executionpolicy unrestricted, majd fogadjunk el mindent.

A proto mappában a következő cmd utasítást kell kiadni a fordításhoz:

```
javac "mainPackage\*.java" "model\*.java" "util\*.java"
```

Ha a powershell scripttel akarjuk fordítani:

a proto mappában nyissunk egy powershell-t
majd adjuk a következő parancsot:
scripts/build

10.1.3 Futtatás

Ha powershell scripttel szeretnénk a programot futtatni, akkor először engedélyeznünk kell azokat. Adminisztrátorként indított powershellbe írjuk a következő parancsot:
set-executionpolicy unrestricted, majd fogadjunk el mindent.

A futtatáshoz a proto mappában a következő utasítást kell kiadni:

```
java mainPackage.Main
```

Ha a powershell scripttel akarjuk futtatni:

a proto mappában nyissunk egy powershell-t
majd adjuk a következő parancsot:
scripts/run

Ha a powershell scripttel akarjuk valamelyik tesztet futtatni és összehasonlítani az elvárt kimenetével:

a proto mappában nyissunk egy powershell-t
majd adjuk a következő parancsot:
scripts/test testX
vagy ha az összes tesztet szeretnénk:
scripts/test all_tests

10.2 Tesztek jegyzőkönyvei

Megjegyzés: A 10.0 alapján minden kimenetből eltűnt a Ciszternánál a hasPump, minden Cső kimenetéhez hozzáadódott a willStuck. Az ezekből adódó hibákat nem tüntetjük fel.

10.2.1 Teszteset1

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 10:30

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 9:30
Teszt eredménye	Hiba
Lehetséges hibaok	Hibás bemenet. Hátról a 3. sor (move cs3) hibás, kimaradt belőle, hogy melyik játékoskal lépünk. Emellett kimaradt a kör végét jelző sorok.
Változtatások	A hibás sort a következőre cseréltem: <i>“move m cs3”</i> . Illetve, a szabotőr mozgása utáni (move s p2 sor után), illetve az utolsó előtti sorba is belekerült egy <i>“round_end 20”</i> tartalmú sor. randomize false, random_default 1 es randomize true sorokkal vettem körbe a repair parancsot.

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 10:15
Teszt eredménye	Hiba
Lehetséges hibaok	Kimenet eltér, rosszul adtuk meg.
Változtatások	A Round értéke 2 (Round: 2). A Szerelő és Szabotőr pontjai 20 helyett 0 értékűek. A <i>“cs5”</i> cső esetében a shieldCounter értéke 0, és a cs3 esetében a stickyCounter értéke is 0, willStuck hozzáadásra került, értéke minden csőnél false. A cs3 cső shieldCounter értéke 1.

10.2.2 Teszteset2

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 11:00

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 10:50
Teszt eredménye	Hiba
Lehetséges hibaok	Kimenet eltér, rosszul adtuk meg.
Változtatások	A Round értéke 0, a csapatok pontjainak értéke is 0. A cs2 cső water értéke 0, míg a shieldCounter értéke 6. A p1 pumpa bemenete cs1, kimenete cs2. A <i>“p2”</i> pumpa bemenete cs2, kimenete cs1, willStuck hozzáadásra került, értéke minden csőnél false.

10.2.3 Teszteset3

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 11:20

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 11:10
Teszt eredménye	Hiba
Lehetséges hibaok	Kimenet eltér, rosszul adtuk meg.
Változtatások	A Round értéke 0, a Szabotőrök hátralévő ideje 300. A cs2 cső water értéke 0, shieldCounter értéke 0. A cs3 cső water értéke 0. A p1 pumpa bemenete cs1, kimenete cs2. A p2 pumpa bemenete cs2, kimenete cs3, , willStuck hozzáadásra került, értéke minden csőnél false.

10.2.4 Teszteset4

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 19:10

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 19:00
Teszt eredménye	Hiba
Lehetséges hibaok	Kimenet eltér, rosszul adtuk meg, illetve a bemenetben is található egy hiba.
Változtatások	A Round értéke 2, willStuck hozzáadásra került, értéke minden csőnél false. A bemenetben a "create_pipe p1 f1 cs2" (5. sor) tartalmazott egy plusz szóközt, ezt javítottam.

10.2.5 Teszteset5

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 19:25

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 19:15
Teszt eredménye	Hiba
Lehetséges hibaok	Kimenet eltér, rosszul adtuk meg, illetve a bemenetben is található egy hiba.
Változtatások	Round értéke 0, saboteurs értéke null helyett semmi, saboteurTeamTimeLeft értéke 300, saboteurTeamPoints értéke 0, willStuck hozzáadásra került, értéke minden csőnél false. A bemenetben a "create_pipe p1 f1 cs2" (5. sor) tartalmazott egy plusz szóközt, ezt javítottam.

10.2.6 Teszteset6

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 19:25

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 19:20
Teszt eredménye	Hiba
Lehetséges hibaok	Kimenet eltér, rosszul adtuk meg, illetve a bemenetben is található egy hiba.
Változtatások	Round értéke 0, willStuck hozzáadásra került, értéke minden csőnél false. A bemenetben a "create_pipe p1 f1 cs2" (5. sor) tartalmazott egy plusz szóközt, ezt javítottam.

10.2.7 Teszteset7

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 19:35

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 19:28
Teszt eredménye	Hiba
Lehetséges hibaok	Kimenet eltér, rosszul adtuk meg, illetve a bemenetben is található egy hiba.
Változtatások	A cs2 cső water értéke 0, shieldCounter értéke 0, cs3 water és stickyCounter értéke is 0, p1 pumpa bemenete cs1, kimenete cs2, p2 pumpa bemenete cs2, kimenete cs3, Szabotőr platform értéke f1, willStuck hozzáadásra került, értéke minden csőnél false. A bemenethez a stat elé a következő két sort adtuk hozzá: "move m1 cs1", és "round_end 0" sor (nem célunk az idő szimulálását tesztelni, a lastActivePlayer miatt szükséges).

10.2.8 Teszteset8

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 19:45

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 19:36
Teszt eredménye	Hiba
Lehetséges hibaok	Kimenet eltér, rosszul adtuk meg, illetve a bemenetben is található egy hiba.
Változtatások	Kimenetben cs1 slipperyCounter értéke 0, cs3 slipperyCounter értéke 9, p1 pumpa bemenete cs1, kimenete cs2, p2 pumpa bemenete cs2, kimenete cs3. Szerelő platform értéke cs1, Szabotőr platform értéke cs3, willStuck hozzáadásra került, értéke minden csőnél false. Bemenetben randomizer helyett randomize-nak kell szerepelnie több helyen is. A 13. sorban a random_default parancsból kimaradt a "_" jel. A stat elé bekerült a "round_end 0" sor (nem célunk az idő szimulálását tesztelni, a lastActivePlayer miatt szükséges).

10.2.9 Teszteset9

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 20:00

Tesztelő neve	Koleszár Kristóf
Teszt időpontja	2023.05.14. 19:45
Teszt eredménye	Hiba
Lehetséges hibaok	Kimenet eltér, rosszul adtuk meg, illetve a bemenetben is található egy hiba.
Változtatások	Round értéke 2, cs4 water értéke 0, cs5 water értéke 0, cs6 water értéke 0, p2 bemenete cs4, p3 bemenete cs4. A bemenetet módosítottam azzal, hogy a "setPump m true" sor után beszúrtam a "round_end 5" sort, a "setPump s true" sor után beszúrtam a "round_end 25" sort.

10.2.10 Teszteset10

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 11:07

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 10:49
Teszt eredménye	Hiba
Lehetséges hibaok	Kimenet hibás, rosszul adtuk meg, illetve változtatás a nyelvben.
Változtatások	<i>A Round értéke a kimenetben 0. A lastActivePlayer az null nem üres. Az intact true, nem false. A Pump-ban az input és output értéke null.</i>

10.2.11 Teszteset11

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 11:54

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 11:30
Teszt eredménye	Hiba
Lehetséges hibaok	Elírás a bemeneti fájlban. Kimenet hibás, rosszul adtuk meg, illetve változtatás a nyelvben.
Változtatások	<i>A test11.in fájl 1. sorában create_cistern create_cisztern helyett. A Round értéke a kimenetben 0. A kiíratás sorrendje megváltozott így átrendezés aszerint. A Pipe-ban stickyCounter tulajdonság korábban kimaradt. Pipe-ban a SlipperyCounter értéke 0. Pipe-ban willStuck új adat. A Pump-ban az output értéke null. A Cistern-ben a hasPump az már nem szükséges, ezért kivettük.</i>

10.2.12 Teszteset12

Tesztelő neve	Tóth Gábor
----------------------	------------

Teszt időpontja	2023.05.14. 12:01
Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 11:56
Teszt eredménye	Hiba
Lehetséges hibaok	Elírás a bemeneti fájlban. Kimenet hibás, rosszul adtuk meg, illetve változtatás a nyelvben.
Változtatások	<i>A test12.in fájl 1. sorában create_cistern create_cisztern helyett. A Round értéke a kimenetben 0. A kiíratás sorrendje megváltozott így átrendezés aszerint. A Pipe-ban stickyCounter tulajdonság korábban kimaradt. Pipe-ban a SlipperyCounter értéke 0. Pipe-ban willStuck új adat. A Pump-ban az output értéke null. A Cistern-ben a hasPump az már nem szükséges, ezért kivettük.</i>

10.2.13 Teszteset13

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 12:22
Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 12:02
Teszt eredménye	Hiba
Lehetséges hibaok	Elírás a bemeneti fájlban. Kimenet hibás, rosszul adtuk meg, illetve változtatás a nyelvben.
Változtatások	<i>A test13.in fájl 1. sorában create_cistern elírás create_cisztern helyett. A Round értéke a kimenetben 0. A kiíratás sorrendje megváltozott így átrendezés aszerint. A Pipe-ban stickyCounter tulajdonság korábban kimaradt. Pipe-ban a SlipperyCounter értéke 0. Pipe-ban willStuck új adat. A Pump-ban az output értéke null. A Cistern-ben a hasPump az már nem szükséges, ezért kivettük. Illetve javítani kellett kicsit a kódot, mert a cső kilyukadt, amikor nem szabadott volna.</i>

10.2.14 Teszteset14 - másolásra

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 12:39
Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 12:30
Teszt eredménye	Hiba
Lehetséges hibaok	Elírás a bemeneti fájlban. Kimenet hibás, rosszul adtuk meg, illetve változtatás a nyelvben.
Változtatások	<i>A test14.in fájl 1. sorában create_cistern elírás create_cisztern helyett. A kiíratás sorrendje megváltozott így átrendezés aszerint. A Pipe-ban stickyCounter tulajdonság korábban kimaradt. Pipe-ban a SlipperyCounter értéke 0. Pipe-ban willStuck új adat. A Pump-ban az output értéke null. A Cistern-ben a hasPump az már nem</i>

	<i>szükséges, ezért kivettük. A test14.in fájl 7. sorában randomize elírás javítása. A tesztbe szükséges egy round_end parancs kiadása, ezt pótoltuk a bemeneti fájlban, és így a mechanicTimeLeft új értéke 299.</i>
--	---

10.2.15 Teszteset15

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 12:42

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 12:41
Teszt eredménye	Hiba
Lehetséges hibaok	Elírás a bemeneti fájlban. Kimenet hibás, rosszul adtuk meg.
Változtatások	<i>A bemeneti test15.in fájlban a repair szó elírásának javítása. Az elvárt kimenetben a Pump-ban az input és az output értéke null.</i>

10.2.16 Teszteset16

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 14:42

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 14:37
Teszt eredménye	Hibás
Lehetséges hibaok	Elírás a bemeneti fájlban. Kimenet hibás, rosszul adtuk meg, illetve változtatás a nyelvben.
Változtatások	<i>A test16.in fájl 1. sorában create_cistern elírás create_cisztern helyett. A Round értéke a kimenetben 0. A kiírás sorrendje megváltozott így átrendezés aszerint. Pipe-ban willStuck új adat. A Cistern-ben a hasPump az már nem szükséges, ezért kivettük.</i>

10.2.17 Teszteset17

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 14:59

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 14:43
Teszt eredménye	Hiba
Lehetséges hibaok	Elírás a bemeneti fájlban. Kimenet hibás, rosszul adtuk meg, illetve változtatás a nyelvben.
Változtatások	<i>A test16.in fájl 1. sorában create_cistern elírás create_cisztern helyett. A Round értéke a kimenetben 0. A kiírás sorrendje megváltozott így átrendezés aszerint. Pipe-ban willStuck új adat. A Cistern-ben a hasPump az már nem szükséges, ezért kivettük. A bemeneti fájl 11. sorában repair p volt hibásan, helyesen: repair s</i>

10.2.18 Teszteset18

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 15:21

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 15:01
Teszt eredménye	Hiba
Lehetséges hibaok	Elírás a bemeneti fájlban. Kimenet hibás, rosszul adtuk meg, illetve változtatás a nyelvben.
Változtatások	<i>A Round értéke a kimenetben 0. A kiíratás sorrendje megváltozott így átrendezés aszerint. A bemeneti fájlban a create_cistern után kell egy szóköz, javítva. A create_pump parancs 2. argumentuma lemaradt, pótolva. A create_pipe parancsban p1 helyett p. Pipe-ban willStuck új adat. A Cistern-ben a hasPump az már nem szükséges, ezért kivettük. A Cistern: után szóköz lemaradt, javítva. Pipe-oknál szóközők lemaradtak. A Pumpnál a szóköz lemaradt. A Pipe-ban a stckyCounter értéke 0.</i>

10.2.19 Teszteset19

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 15:32

Tesztelő neve	Tóth Gábor
Teszt időpontja	2023.05.14. 15:22
Teszt eredménye	Hiba
Lehetséges hibaok	Elírás a bemeneti fájlban. Kimenet hibás, rosszul adtuk meg, illetve változtatás a nyelvben.
Változtatások	<i>A kiíratás sorrendje megváltozott így átrendezés aszerint. A bemeneti fájlban a create_cistern után kell egy szóköz, javítva. A create_pump parancs 2. argumentuma lemaradt, pótolva. A create_pipe parancsban p1 helyett p. Pipe-ban willStuck új adat. A tesztbe szükséges egy round_end parancs kiadása, ezt pótoltuk a bemeneti fájlban, és így a mechanicTimeLeft új értéke 299. A Pipe-ban szóközők hiányoztak. A cs1-ben slipperyCounter értéke 10. A Pump-ban szóközők hiányoztak. A Cistern-ben a hasPump az már nem szükséges, ezért kivettük. A Cistern-ben szóközők hiányoztak. A Mechanic platformjának az értéke cs2.</i>

10.2.20 Teszteset20

Tesztelő neve	Szautner Béla
Teszt időpontja	2023.05.14 19:53

Tesztelő neve	Szautner Béla
Teszt időpontja	2023.05.14 19:40
Teszt eredménye	Hiba

Lehetséges hibaok	Elírás a bemeneti fájlban, változás volt a kimeneti nyelvben, valamint néhány kimeneti érték rosszul lett megadva
Változtatások	<i>A hiányzó whitespace karakterekkel ki lett bővítve a bemenet. A Round értéke 0 lett, a "mechanicTeamTimeLeft", valamint a "saboteurTeamTimeLeft" értéke 300-ra lett beállítva egységesen. Ezek mellett átrendezésre került a kimeneti fájl szerkezete (csak a sorrendiség változott), valamint hozzáadásra került a "willStuck: false" és a Mechanic:m hez tartozó platform értéke cs1-re változott.</i>

10.2.21 Teszteset21

Tesztelő neve	Szautner Béla
Teszt időpontja	2023.05.14 22:27

Tesztelő neve	Szautner Béla
Teszt időpontja	2023.05.14 22:00
Teszt eredménye	Hiba
Lehetséges hibaok	A teszt az utólagos játék struktúra módosítás miatt "A Szerelő karakter felvesz egy pumpát a ciszternáról". A struktúrális módosítás tartalma: a Szerelő a ciszternán mindig fel tud venni pumpát, míg korábban csak véletlen időközönként tudott felvenni. Elírás a bemeneti fájlban, változás volt a kimeneti nyelvben, valamint néhány kimeneti érték rosszul lett megadva.
Változtatások	<i>A hiányzó whitespace karakterekkel ki lett bővítve a bemenet. A Round értéke 0 lett. Ezek mellett átrendezésre került a kimeneti fájl szerkezete (csak a sorrendiség változott), valamint hozzáadásra került a "willStuck: false" és a Mechanic:m hez tartozó "hasPump" értéke true lett.</i>

10.2.22 Teszteset22

Tesztelő neve	Szautner Béla
Teszt időpontja	2023.05.14 23:00

Tesztelő neve	Szautner Béla
Teszt időpontja	2023.05.14 22:30
Teszt eredménye	Hiba
Lehetséges hibaok	Elírás a bemeneti fájlban, változás volt a kimeneti nyelvben, valamint néhány kimeneti érték rosszul lett megadva
Változtatások	<i>A hiányzó whitespace karakterekkel ki lett bővítve a bemenet. A Round értéke 0 lett. Ezek mellett átrendezésre került a kimeneti fájl szerkezete (csak a sorrendiség változott), valamint hozzáadásra került a "willStuck: false" és a Mechanic:m hez tartozó platform értéke cs1-re változott.</i>

10.2.23 Teszteset23

Tesztelő neve	Szautner Béla
----------------------	---------------

Teszt időpontja	2023.05.14 22:59
Tesztelő neve	Szautner Béla
Teszt időpontja	2023.05.14 22:30
Teszt eredménye	Hiba
Lehetséges hibaok	Elírás a bemeneti fájlban, változás volt a kimeneti nyelvben, a kimeneti értékek jól lettek megadva, azonban a teszt szerkezetileg egyszerűbb lett, ugyanazon eredmény mellett. (Elemibb lett a teszt)
Változtatások	<p><i>A teszt egyszerűsítve lett (eredetileg sem tartozott az összetett tesztek közé), viszont ugyanazokat a funkciókat valósítja meg mint a teszt előző verziója. A teszt a következőkben így néz ki, tördelés és whitespace karakterekkel bővítve</i></p> <pre> randomize false random_default 1 create_spring (1,1) sp create_cistern (1,5) c create_pipe sp c cs create_mechanic cs m create_saboteur cs s leakPipe s step sp step c round_end 20 round_end 20 stat </pre>

10.2.24 Teszteset24

Tesztelő neve	Szautner Béla
Teszt időpontja	2023.05.15 10:00
Tesztelő neve	Szautner Béla
Teszt időpontja	2023.05.15 9:30
Teszt eredménye	Hiba
Lehetséges hibaok	A putPump paramétere lemaradt, ezek mellett több kimeneti hiba is található.
Változtatások	<p><i>A kimeneti fájlban a hiányzó whitespaceket pótolni kellett. A Pipe: cs1 hez tartozó stickyCounter értékét 0 kellett állítani (hibás volt az eredeti tesztben megadott 9 érték). A Pipe: cs3 -nél az end1: sp1-re, az end2: p1-re módosult. A cs4-nél a slipperyCounter értéke 10re módosult. A Pump: p1 pipes-hez hozzá lett adva a cs6 cső. A cs7 cső, valamint Pump: p3 pumpa ki lett véve a teszesetből, mivel az nem lett létrehozva. P2 pumpa outputja cs2 lett. Spring: sp1 pipes</i></p>

	<i>paraméterének értéke cs4 helyett cs3 lett. Cistern pipes-ben a cs3 cs4-re lett javítva. A Cistern:c duplikáció ki lett véve. A Saboteur:s-hez tartozó platform cs4-re lett kijavítva</i>
--	---

10.2.25 Teszteset25

Tesztelő neve	Szautner Béla
Teszt időpontja	2023.05.15 11:00

Tesztelő neve	Szautner Béla
Teszt időpontja	2023.05.15 10:30
Teszt eredménye	A teszt sikeresen lefutott.
Lehetséges hibaok	-
Változtatások	-

10.3Értékelés

Tag neve	Tag neptun	Munka százalékban
Kovács Zoltán	W4YDIT	20
Koleszár Kristóf	I1TLYH	20
Gegő Levente	LYNUXB	20
Tóth Gábor	F041OM	20
Szautner Béla	EEID48	20

10.4 Napló

Kezdet	Időtartam	Résztevők	Leírás
2023.05.08 10:00	2 óra	Gegő	Kódolás megkezdése
2023.05.10 12:00	2 óra	Kovács	Kódolás
2023.05.13 12:00	2 óra	Gegő Kovács	Kód befejezése
2023.05.14 9:00	2,5 óra	Koleszár	10.2 Tesztesetek jegyzőkönyveinek elkezdése.
2023.05.14 10:00	2 óra	Gegő	Hibák javítása
2023.05.14 22:00	1,5 óra	Kovács	Tesztesetek átnézése, javítása, tesztelése
2023.05.14 18:00	4 óra	Koleszár	10.2 Tesztesetek jegyzőkönyveinek befejezése
2023.05.14 10:00	6 óra	Tóth	10.2 Tesztesetek jegyzőkönyvezése
2023.05.15 10:00	3 óra	Tóth	10.2 Tesztesetek bővítése.
2023.05.14 15:00	4 óra	Szautner	10.2 Tesztesetek jegyzőkönyvezése
2023.05.15 10:00	1,5 óra	Szautner	10.2 Tesztesetek jegyzőkönyvezése

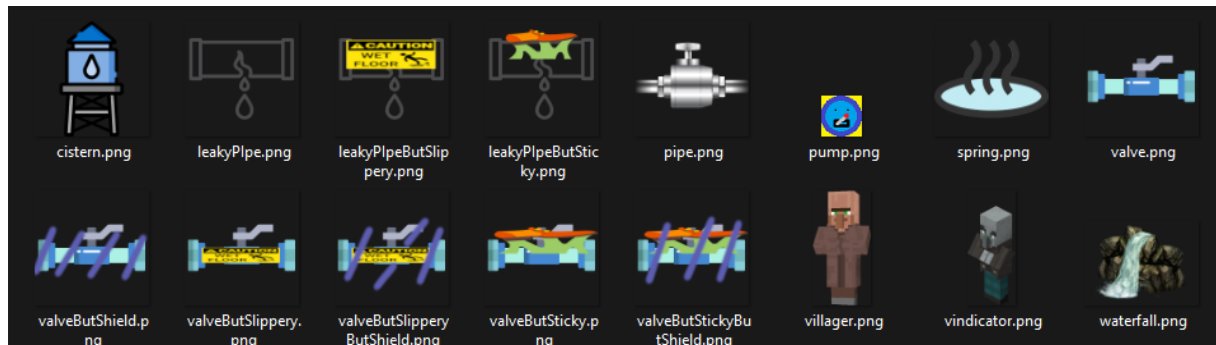
11. Grafikus felület specifikációja

11.0 Osztálydiagramot érintő változás

A régi osztályok közül eltűnt a “Named”.

11.1 A grafikus interfész

A pályaelemek és játékosok a következő ikonokkal lesznek ábrázolva:



cistern.png: a ciszterna ikonja

leakyPipe.png: a lyukas cső ikonja

valve.png: az ép cső ikonja

pump.png: a pumpa ikonja

waterfall.png: a forrás ikonja

villager.png: a szerelő ikonja

vindicator.png: a szabotőr ikonja

valveButShield.png: olyan cső ikonja ami éppen nem lyukasztható

valveButSlipperyButShield.png: olyan csúszós cső ikonja ami éppen nem lyukasztható

valveButStickyButShield.png: olyan ragadós cső ikonja ami éppen nem lyukasztható

leakyPipeButSlippery.png: olyan lyukas cső ikonja ami éppen csúszós

leakyPipeButSticky.png: olyan lyukas cső ikonja ami éppen ragadós

valveButSlippery.png: olyan cső ikonja ami éppen csúszós

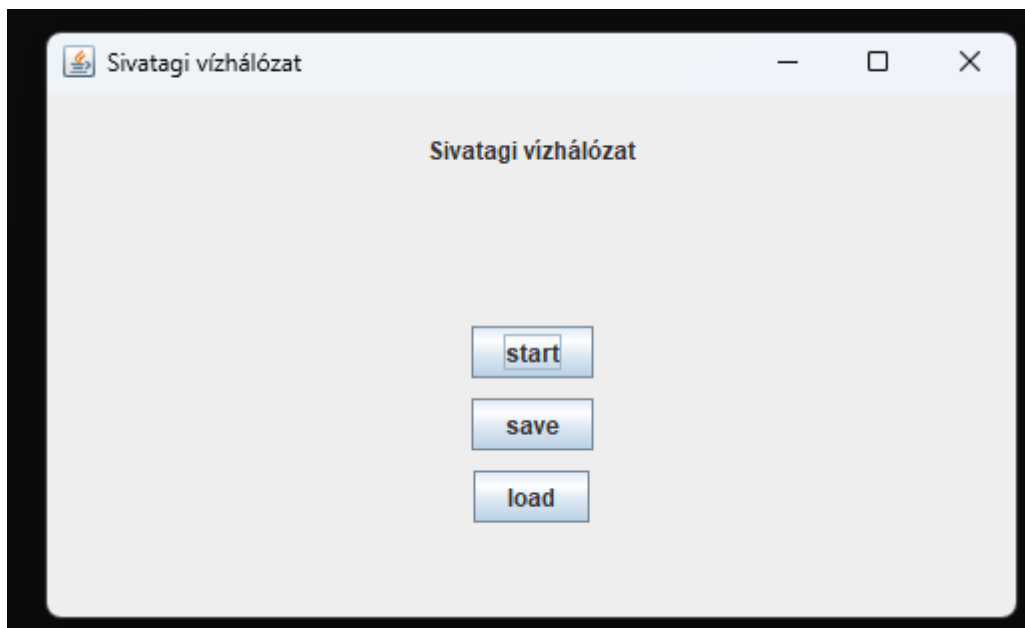
valvePipeButSticky.png: olyan cső ikonja ami éppen ragadós

A játék indításakor egy főmenü jelenik meg ahol a játék különböző beállításait lehet elvégezni.

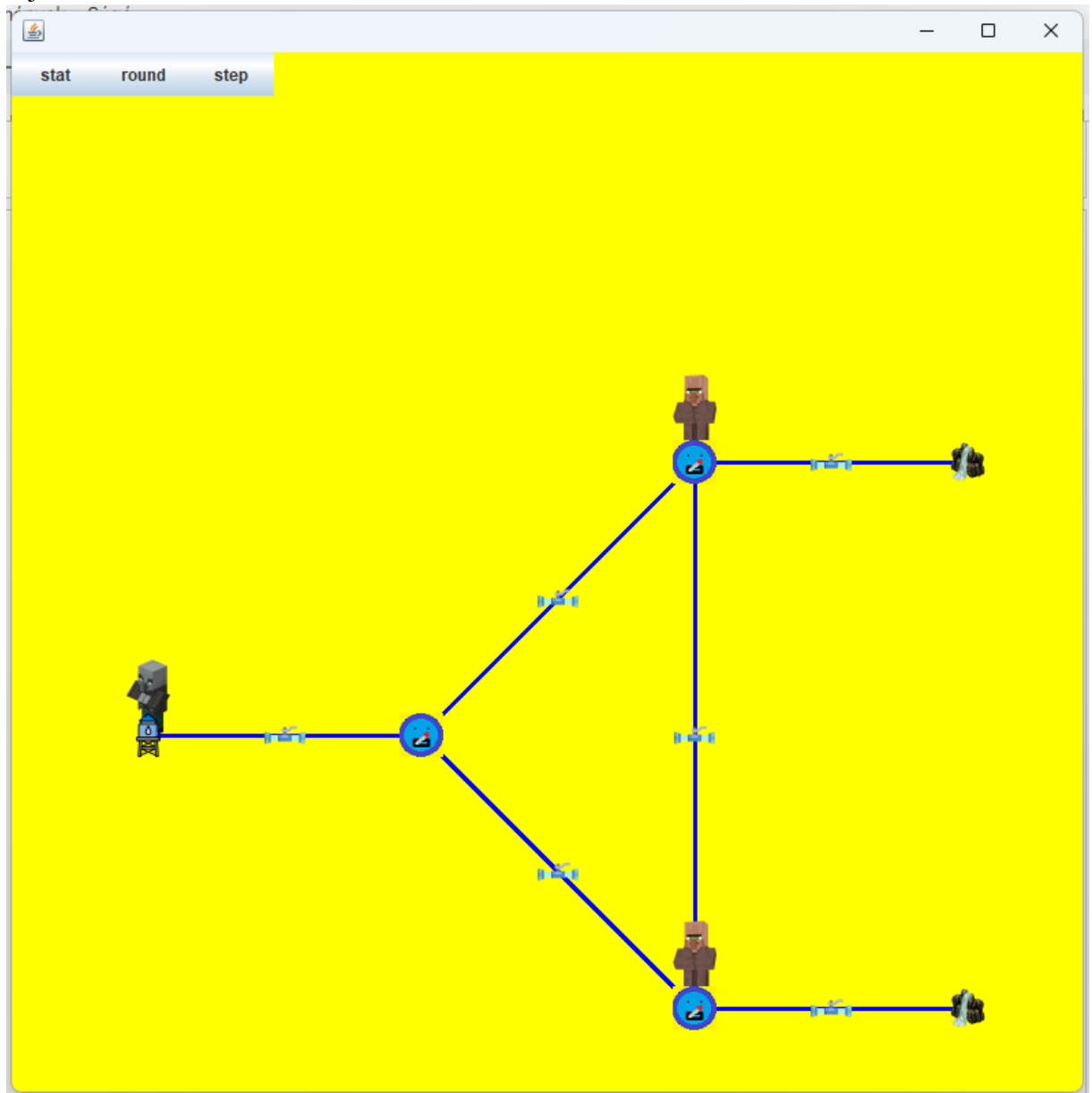
Ezután a “Start” gomb megnyomásával lehet a játékot elindítani, ami bezárja a menü ablakát és létrehozza a játék ablakát.

Néhány kép a grafikus felületről:

A főmenü:



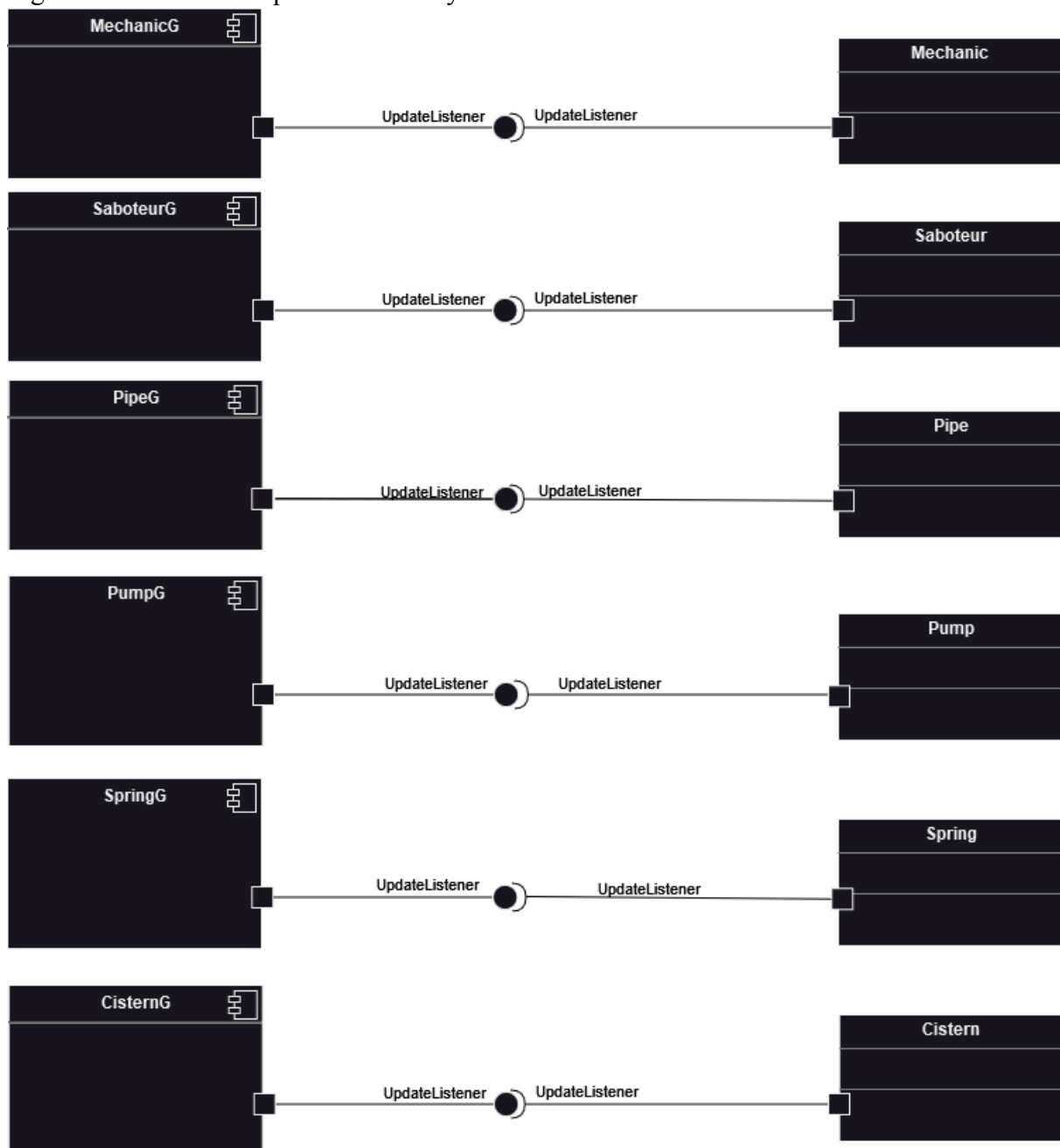
A játék ablaka:



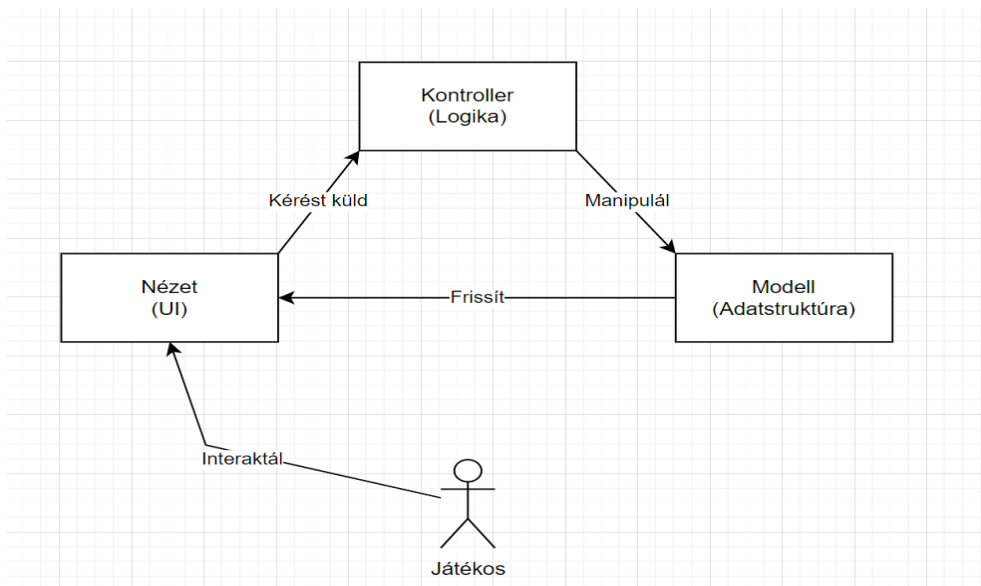
(a végső verzióban a felső gombok nem lesznek részei a grafikus megjelenítésnek, most csak debug/test célokkal vannak ott)

11.2 A grafikus rendszer architektúrája

A grafikus felülethez kapcsolódó osztályok a következők:



11.2.1A felület működési elve

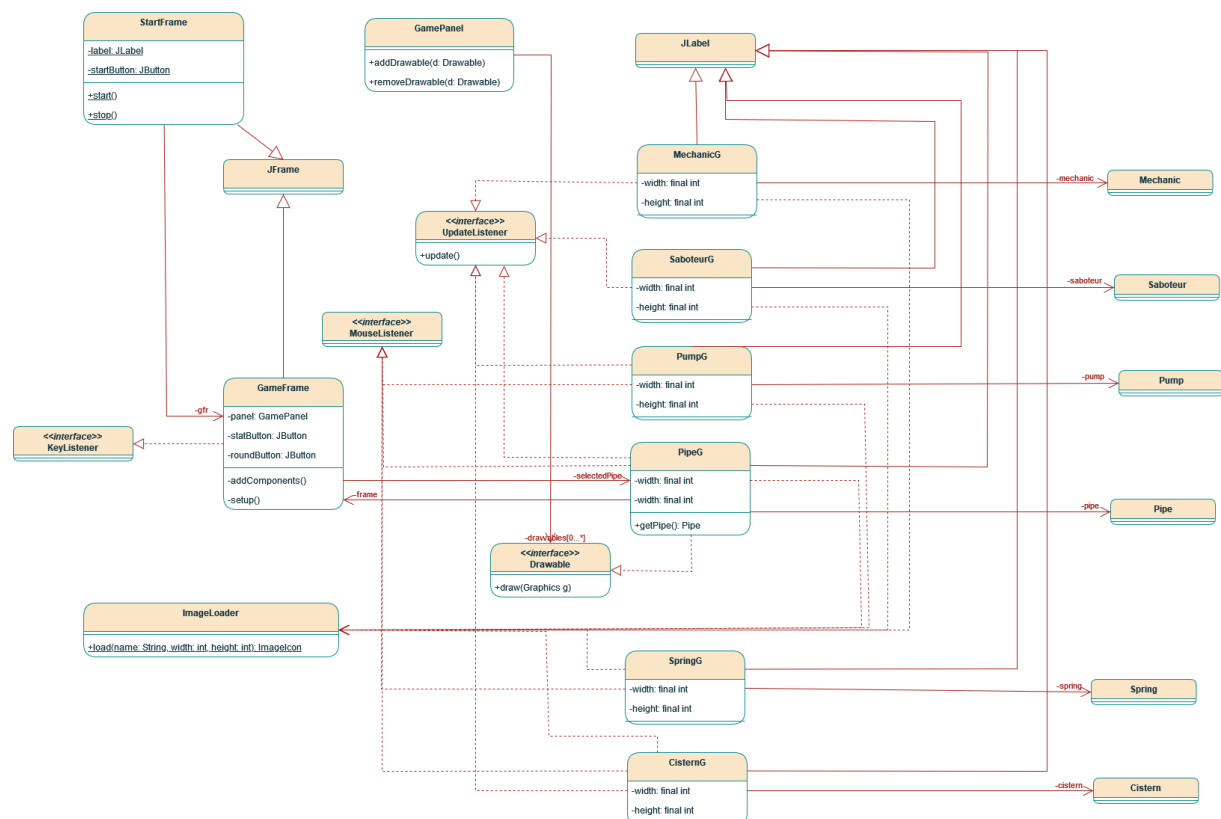


A programunkat már az elejétől fogva egy push alapú MVC-modellnek megfelelően alakítottuk ki.

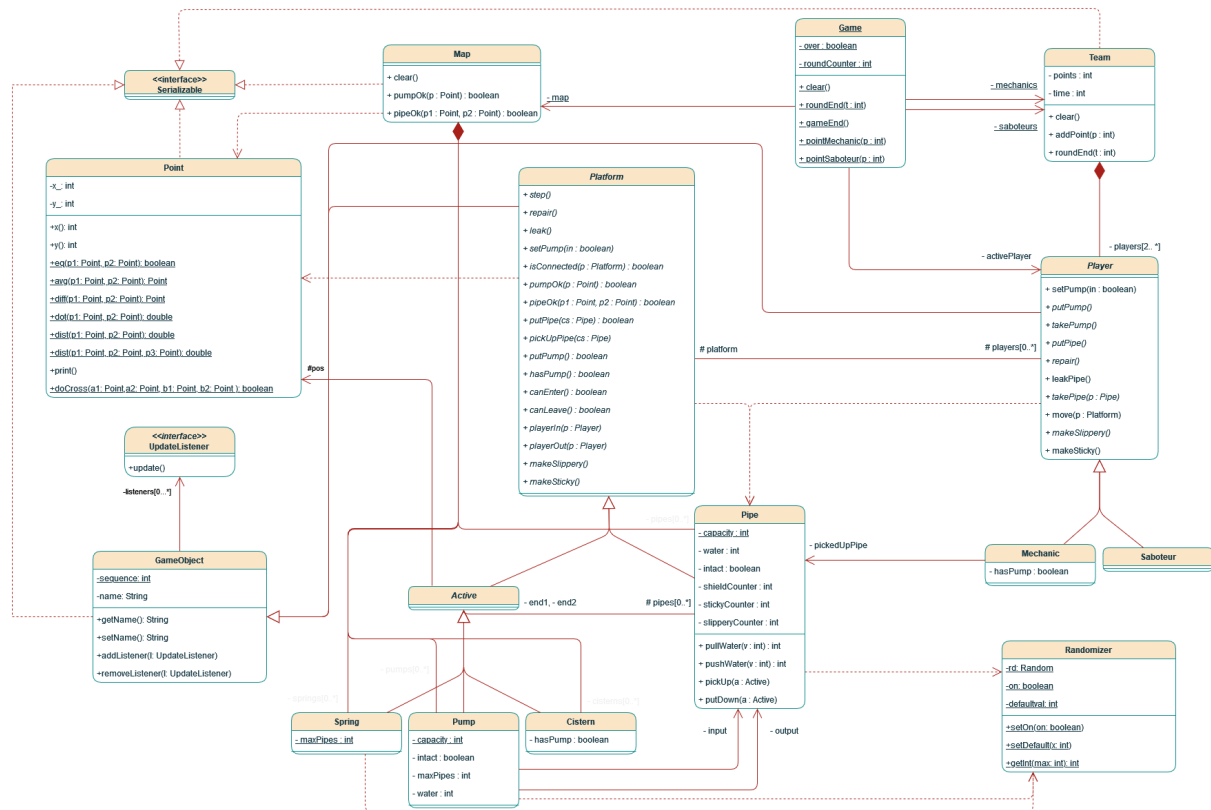
A grafikai megjelenésért felelős osztályok, objektumok, az observer mintának megfelelően az UpdateListener interfészt implementálva tudnak kommunikálni a modell osztályaival, objektumaival, amelyek pedig az GameObject osztályból örökölnék.

11.2.2A felület osztály-struktúrája

A grafikus osztálydiagram:



A módosított modellhez tartozó osztálydiagram:



11.3 A grafikus objektumok felsorolása

Az alábbi részletes leírásokban azért vannak kihagyott -interfészek általi- metódusok, mivel ezeknek a nincs szerepük a játék működésében. Ez a programkódban úgy van megvalósítva, hogy létre vannak hozva az interfészek általi metódusok, azonban ezeknek a függvény törzse üres.

11.3.1 UpdateListener

- **Felelősség**

A léptethetőséget lehetővé tevő interface.

- **Metódusok**

+ void update(): Ezzel értesíti az interfészt megvalósító modell elem a grafikus felületet a változásokról.

11.3.2 Drawable

- **Felelősség**

A kirajzolhatóságot lehetővé tevő interface.

- **Metódusok**

+ void draw(Graphics g): Egy komponens ikonján kívüli geometriai objektumok kirajzolását teszi lehetővé.

11.3.3 GameFrame

- **Felelősség**

Ez az osztály felel a játék irányíthatóságáért, és a fő megjelenítésért, és főbb funkciókért.

- **Ősosztályok**

JFrame

- **Interfészek**

KeyListener

- **Attribútumok**

- -GamePanel panel: Egy Gamepanel objektumra referál, ezen jelenik meg a játék grafikus felülete.
- - PipeG selectedPipe: Az utolsó kurzorral érintett csőre mutató referencia.

- **Metódusok**

- +void addPipe(Pipe p): Cső hozzáadása a hálózathoz.
- +void addPump(Pump p): Pumpa hozzáadása a hálózathoz.
- +void addComponents(): Az eddigiekben a pályához hozzáadott elemeket elhelyezi grafikusán, azaz hozzáadja a panelhez
- +void setup(): Egy előkészítő kódot futtató metódus.
- + void keyPressed(KeyEvent e): A billentyűzet gombjainak kezelése.

11.3.4 StartFrame

- **Felelősség**

Az osztály valósítja meg a Játék indítását. Ez a főmenü / launcher.

- **Ősosztályok**

JFrame

- **Interfészek**

-

- **Attribútumok**

- -JLabel label: A kezdőképernyőn a megjelenítendő kezdő szöveget tartalmazó mező.
- -JButton startButton: Egy gombot definiál, amelyet megnyomva a játék elkezdődik.
- -JButton loadButton: Egy gombot definiál, amelyet megnyomva egy felugró ablakból választhatjuk ki a betöltendő mentést.
- -JButton saveButton: Egy gombot definiál, amelyet megnyomva a játékállás egy megadható nevű fájlba mentődik.
- -GameFrame gfr: A a játék egyik fő Osztályára (GameFrame) referál, és felelős érte.

- **Metódusok**

- +void start(): A Játék indítását teszi lehetővé.
- +void load(): Egy játékállás betöltését teszi lehetővé.
- +void save(): A játékállás mentését teszi lehetővé.
- +void stop(): A Játék megállítását teszi lehetővé.

11.3.5 GamePanel

- **Felelősség**

Ez az osztály felel a grafikus, interaktív felület megjelenítéséért.

- **Ősosztályok**

JPanel

- **Interfészek**

-

- **Attribútumok**

- -List<Drawable> drawables: A kirajzolható komponensek tárolását teszi lehetővé.

- **Metódusok**

- +void addDrawable (Drawable d): A kirajzolható komponens (d paraméter) hozzáadását teszi lehetővé.
- +void removeDrawable (Drawable d): Az adott kirajzolható komponens (d paraméter) a tárolóból(drawables) való eltávolítását teszi lehetővé.
- +void paintComponent(Graphics g): A rajzolás megvalósítása.

11.3.6 ImageLoader

- **Felelősség**

A különböző karakterek és pálya elemekhez tartozó ikon betöltéséért felelős osztály.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- + void ImageIcon load(String name, int width, int height): Az adott nevű (name) képet betölti, majd a megadott méretre állítja

11.3.7 MechanicG

- **Felelősség**

Ez az osztály felelős a grafikus Szerelő karakterért.

- **Ősosztályok**

JLabel

- **Interfészek**

UpdateListener

- **Attribútumok**

- -Mechanic mechanic: A modell belüli szerelő objektum. Ennek definiálása a modellben már megtörtént.
- +final int width: A grafikus objektum szélessége.
- +final int height: A grafikus objektum magassága.

- **Metódusok**

- +void update(): Az UpdateListener interfészben definiált metódus megvalósítása, amely itt arra szolgál, hogy beállítsa az aktuális platformhoz tartozó pozíciót a szerelő objektumban.

11.3.8 PipeG

- **Felelősség**

A cső a játékban való grafikus megjelenítéséért felelős.

- **Ősosztályok**

JLabel

- **Interfészek**

UpdateListener, Drawable, MouseListener

- **Attribútumok**

- +final int width: A grafikus objektum szélessége.
- +final int height: A grafikus objektum hosszúsága.
- -Pipe pipe: A modell belüli cső objektum. Ennek definiálása a modellben már megtörtént.
- -GameFrame frame: A játékhoz tartozó GameFrame referencia.

- **Metódusok**

- +void draw(Graphics g): Az interfészben definiált metódus megvalósítása, itt kirajzolja a csövet a grafikus felületre.
- +void update(): Az UpdateListener interfész által definiált metódus, amely jelen esetben a léptethetőségért felel.
- -void create(): Létrehoz egy új csövet a grafikus felületen.
- +Pipe getPipe(): Visszaadja az aktuális csövet, amelynek a megjelenítését teszi lehetővé

11.3.9 PumpG

- **Felelősség**

A Pumpa a játékban való grafikus megjelenítéséért felelős osztály.

- **Ősosztályok**

JLabel

- **Interfészek**

UpdateListener, MouseListener

- **Attribútumok**

- +final int width: A grafikus objektum szélessége
- +final int height: A grafikus objektum magassága
- -Pump pump: A modell belüli pumpa objektum. Ennek definiálása a modellben már megtörtént.

- **Metódusok**

- +void update(): Az UpdateListener interfész által definiált metódus, amely jelen esetben a léptethetőségért felel.
- +void mousePressed(MouseEvent e): Lehetővé teszi azt, hogy az elem "úgy viselkedjen, mint egy gomb".

11.3.10 SaboteurG

- **Felelősség**

- Ez az osztály felelős a grafikus Szabotőr karakterért.

- **Ősosztályok**

- JLabel

- **Interfészek**

- UpdateListener

- **Attribútumok**

- -Saboteur saboteur: A modell belüli szabotőr objektum. Ennek definiálása a modellben már megtörtént.
- +final int width: a grafikus objektum szélessége.
- +final int height: a grafikus objektum magassága.

- **Metódusok**

- +void update(): az interfészben definiált metódus megvalósítása, amely itt arra szolgál, hogy beállítsa az aktuális platformhoz tartozó pozíciót a szabotőr objektumban.
- + void paintComponent(Graphics g): A grafikus megjelenítésért felelős metódus.

11.3.11 SpringG

- **Felelősség**

A Forrás a játékban való grafikus megjelenítéséért felelős osztály.

- **Ősosztályok**

JLabel

- **Interfészek**

UpdateListener, MouseListener

- **Attribútumok**

- +final int width: A grafikus objektum szélessége.
- +final int height: A grafikus objektum magassága.
- -Spring spring: A modell beli forrás objektum. Ennek definiálása a modellben már megtörtént.
- -GameFrame frame: A játékhoz tartozó GameFrame referencia.

- **Metódusok**

- +void update(): Az UpdateListener interfész által definiált metódus, amely jelen esetben a léptethetőségért felel.
- -void create(): Létrehoz egy új forrást a grafikus felületen.
- +void paintComponent(Graphics g): A grafikus megjelenítésért felel
- +void mousePressed(MouseEvent e): Lehetővé teszi azt, hogy az elem “úgy viselkedjen, mint egy gomb”.

11.3.12 CisternG

- **Felelősség**

A Ciszterna a játékban való grafikus megjelenítéséért felelős osztály.

- **Ősosztályok**

JLabel

- **Interfészek**

UpdateListener, MouseListener

- **Attribútumok**

- +final int width: A grafikus objektum szélessége
- +final int height: A grafikus objektum magassága
- -Cistern cistern: A modell beli forrás objektum. Ennek definiálása a modellben már megtörtént.
- -GameFrame frame: A játékhoz tartozó GameFrame referencia.

- **Metódusok**

- +void update(): Az UpdateListener interfész által definiált metódus, amely jelen esetben a léptethetőségért felel.
- -void create(): Létrehoz egy új ciszternát a grafikus felületen.
- +void paintComponent(Graphics g): A grafikus megjelenítésért felel
- +void mousePressed(MouseEvent e): Lehetővé teszi azt, hogy az elem “úgy viselkedjen, mint egy gomb”.

11.3.13 Map

Változás: Az attribútumok közül kikerültek a következők:

- - List<Pipe> pipes: A pálya csöveinek listája.
- - List<Cistern> cisterns: A pálya ciszternáinak listája
- - List<Spring> springs: A pálya forrásainak listája
- - List<Pump> pumps: A pálya pumpáinak listája

11.3.14 Randomizer

- **Felelősség**

Random számok generálásával kapcsolatos feladatokat ellátó osztály.

- **Össosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- -Random rd: A Random osztály egy példánya.
- -boolean on: Azt jelzi, hogy a random számok generálása be van-e kapcsolva.
- -int defaultval: Alapértelmezett random egész szám.

- **Metódusok**

- +void setOn(boolean on): Random számok generálásának bekapcsolása.
- +void setDefault(int x): Alapértelmezett egész szám értékének beállítása.
- +int getInt(int max): Visszaad egy random egész számot a [0,max) intervallumon.

11.3.15 Point

- **Felelősség**

Egy (x és y koordinátákkal rendelkező) pontot reprezentáló osztály.

- **Össosztályok**

-

- **Interfészek**

Serializable

- **Attribútumok**

- -int x_: A pont x koordinátája.
- -int y_: A pont y koordinátája.

- **Metódusok**

- +int x(): Visszaadja a pont x koordinátáját.
- +int y(): Visszaadja a pont y koordinátáját.
- +boolean eq(Point p1, Point p2): Visszaadja, hogy a két pont megegyezik-e.
- +Point avg(Point p1, Point p2): Visszaadja a két pont által meghatározott szakasz középpontját.
- +Point diff(Point p1, Point p2): Visszaadja a különbség vektor végpontját ((0,return) a vektor).
- +double dot(Point p1, Point p2): Visszaadja a két vektor skaláris szorzatát.
- +double dist(Point p1, Point p2): Visszaadja a két pont távolságát.
- +double dist(Point p1, Point p2, Point p3): Visszaadja a (p1, p2) által meghatározott szakasz és p3 pont távolságát.
- +boolean doCross(Point a1, Point a2, Point b1, Point b2): Visszaadja, hogy a két szakasz - (a1,a2) és (b1,b2) - keresztezi-e egymást.

11.3.16 GameObject

- **Felelősség**

Játékbeli objektumokat reprezentáló osztály.

- **Ősosztályok**

-

- **Interfészek**

Serializable

- **Attribútumok**

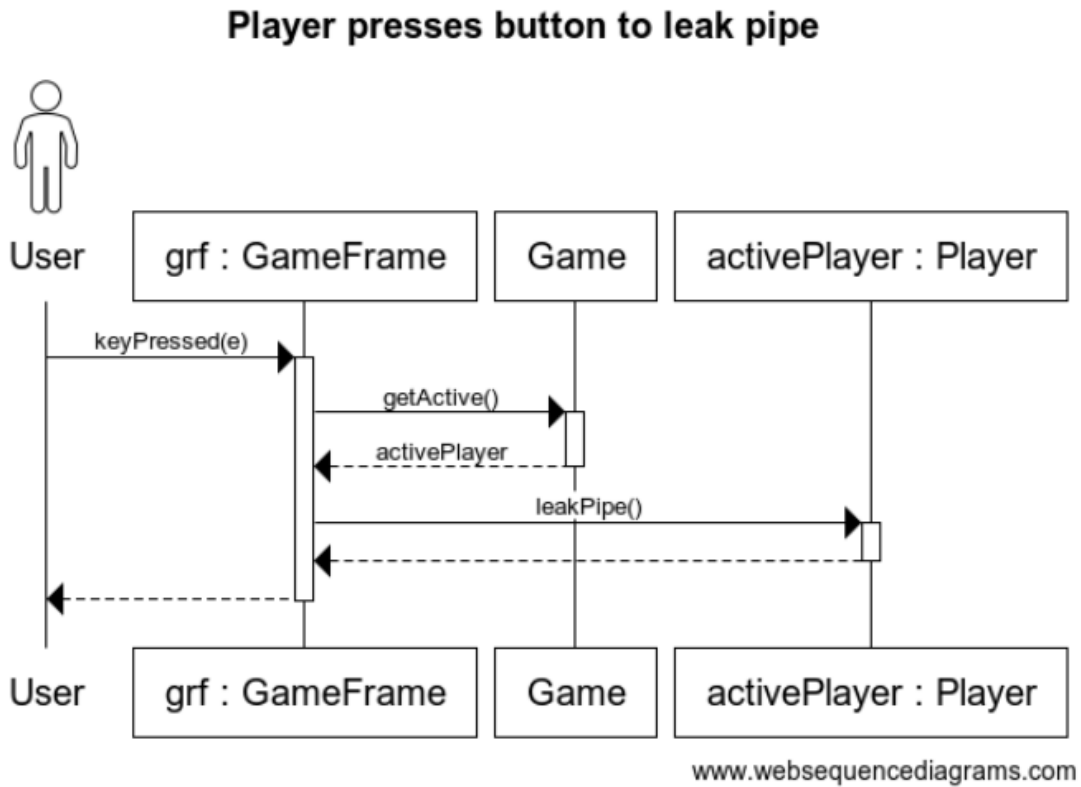
- -int sequence: A nevek egyediségét biztosító számláló.
- -String name: Objektum neve.

- **Metódusok**

- +String getName(): Visszaadja az objektum nevét.
- +String setName(): Beállítja az objektum nevét.
- +void addListener(UpdateListener l): Új UpdateListener hozzáadása.
- +void removeListener(UpdateListener l): UpdateListener eltávolítása.

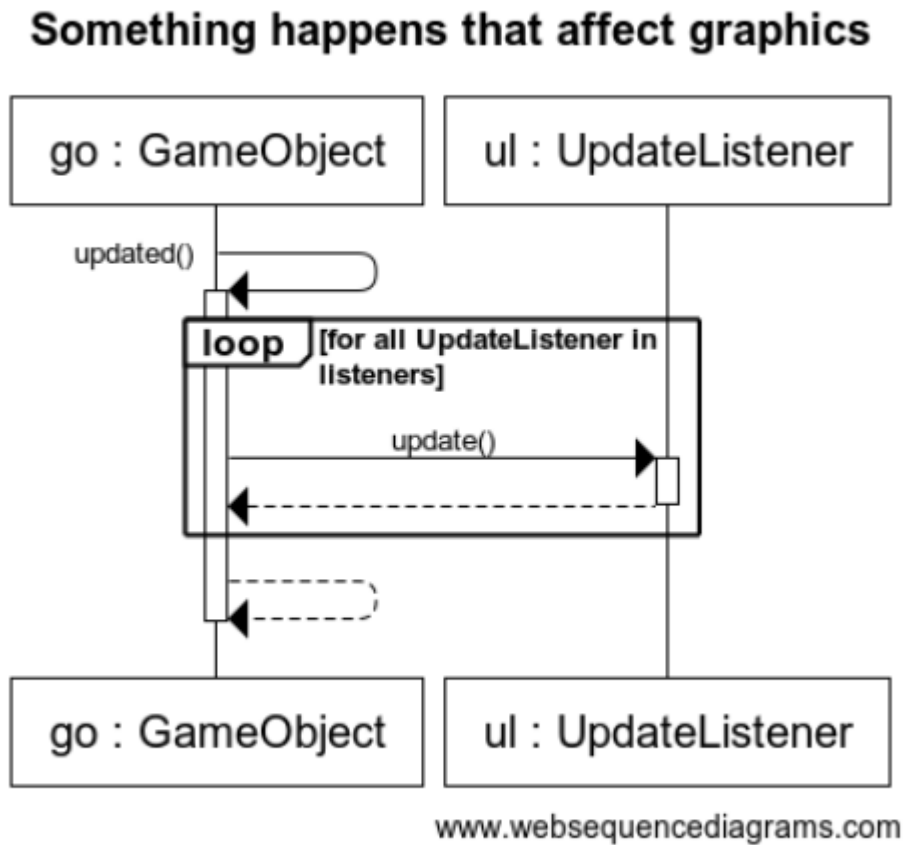
11.4 Kapcsolat az alkalmazói rendszerrel

11.4.1 Player presses button to leak pipe



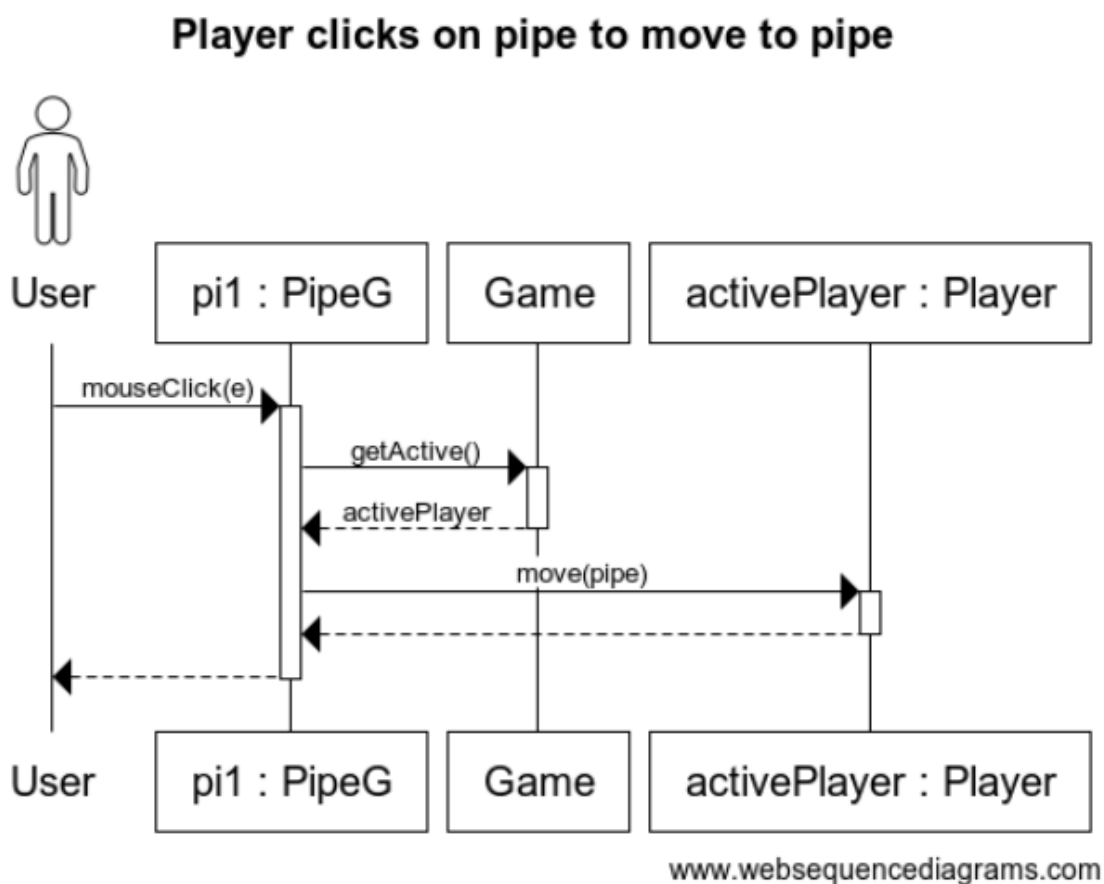
Itt a leakPipe helyett a következők mehetnek még végbe: makeSticky, makeSlippery, takePump, putPump, takePipe, putPipe, setPump(false), setPump(true), repair.

11.4.2 Something happens that affect graphics



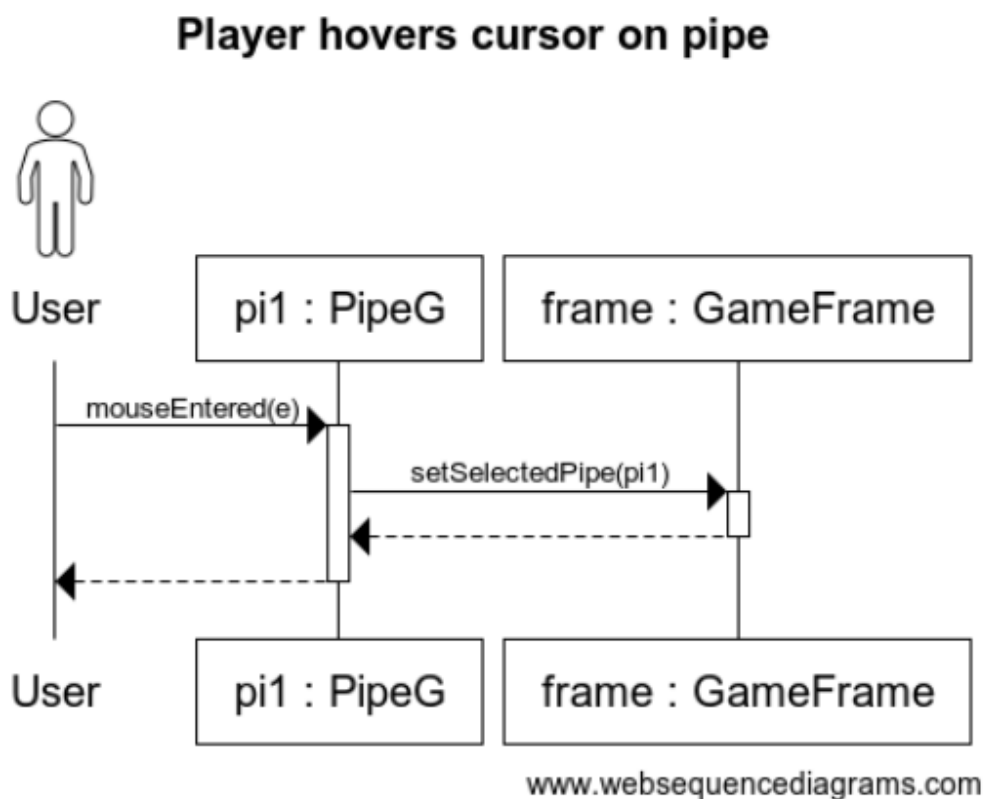
Amikor a modell belső állapota megváltozik, a GameObject leszármazottja meghívja az `updated()` függvényt, így a grafikus felület is frissül. Így kommunikál egymással a grafikus felület és a modell.

11.4.3 Player clicks on pipe to move to pipe



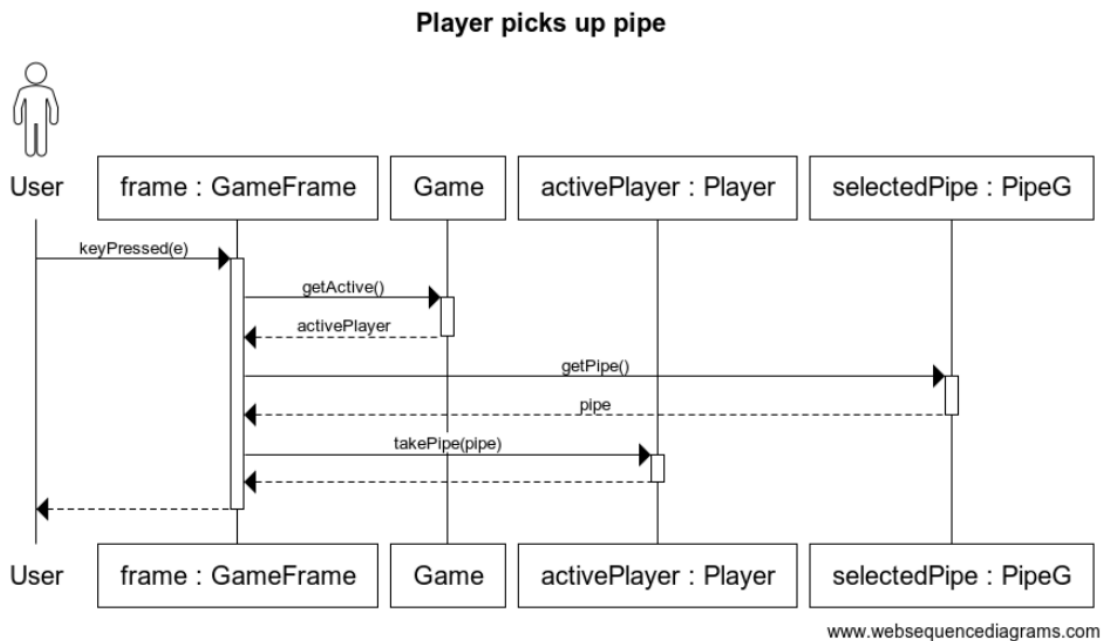
Itt bármely másik kattintással elvégezhető művelet hasonlóan történik. (pl.: PipeG lehetne akár PumpG is, stb...).

11.4.3 Player hovers cursor on pipe

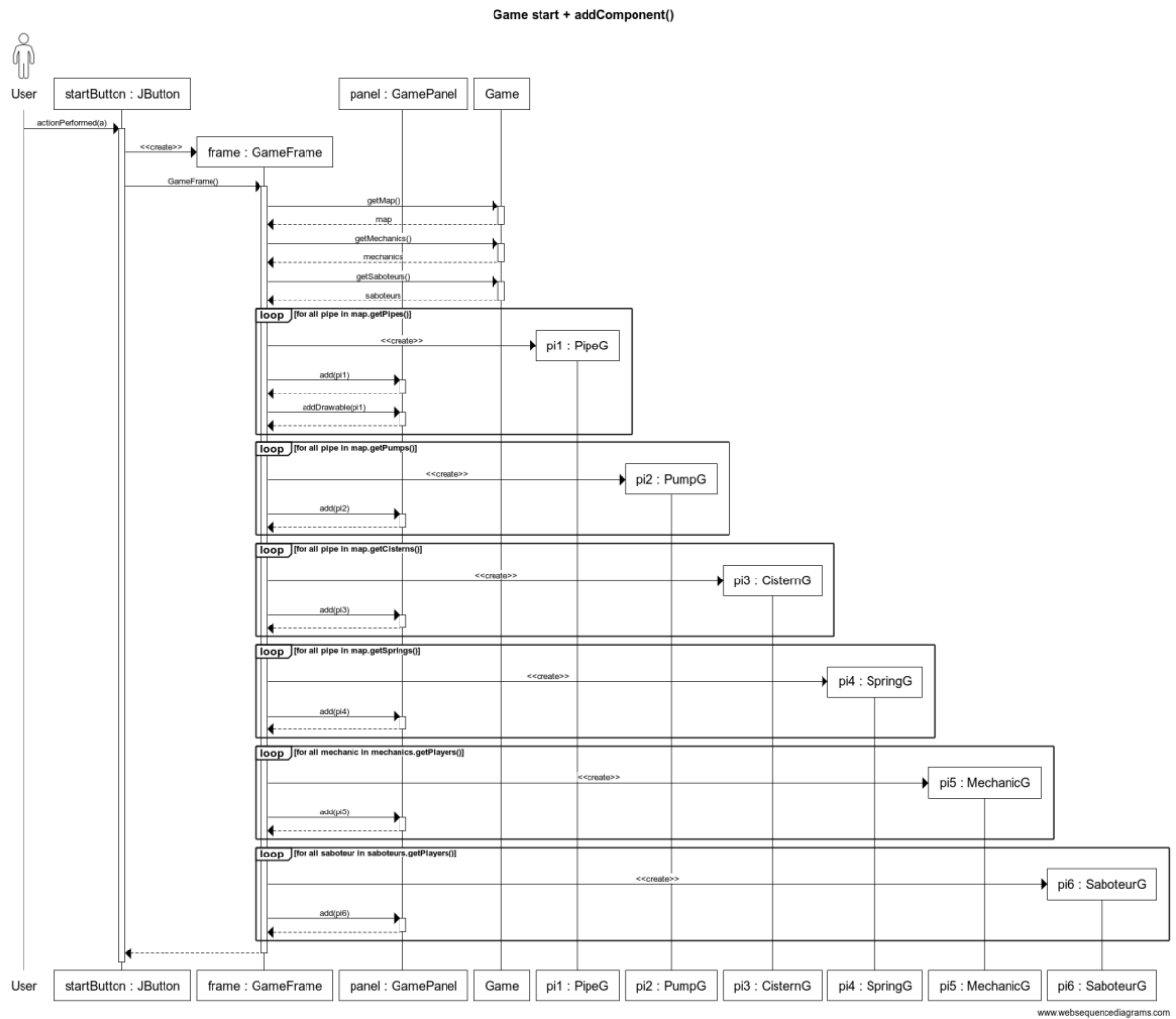


Ha a felhasználó ráviszi a kurzort a cső ikonjára akkor az beállítja magát az utoljára kiválasztott csőnek.

11.4.4 Player picks up pipe



11.4.5 Játék és grafikus felület inicializálása



11.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2023.05.20 14:00	4 óra	Gegő Kovács	Dokumentáció elkezdése
2023.05.21. 12:00	1,5 óra	Gegő Kovács Szautner	Értekezlet, 11.3. Grafikus objektumok áttekintése.
2023.05.21. 14:00	7 óra	Szautner	11.3 Grafikus objektumokhoz tartozó osztálydiagram, leírások. Modellhez tartozó diagram kibővítése.
2023.05.21. 0:00	1,5 óra	Koleszár	11.3 Grafikus objektumok felsorolásának elkezdése
2023.05.21. 10:00	2 óra	Koleszár	11.3 Eddig elkészült Grafikus objektumok felsorolásainak ellenőrzése, újak írása
2023.05.21. 11:00	3 óra	Szautner	Osztálydiagramok bővítése felmerülő hibák javítása. Struktúra diagramok elkezdése
2023.05.21. 18:00	5 óra	Gegő Szautner Kovács Koleszár	Értekezlet: Struktúra diagramok és szekvencia diagramok befejezése, kisebb hibák javítása
2023.05.21 23:00	1 óra	Koleszár Szautner	Struktúra diagramok megrajzolása
2023.05.22 00:00	1 óra	Gegő Kovács	Struktúra diagramok ellenőrzése
2023.05.22 01:00	1 óra	Kovács Gegő	Dokumentáció befejezése, utolsó ellenőrzések

13. Grafikus változat beadása

13.1 Fordítási és futtatási útmutató

13.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
mainPackage/CmdParser.java	24 KB	2023.05.27	A parancssori bemeneti argumentumokat betöltését és elemzését megvalósító fájl.
mainPackage/Main.java	1KB	2023.05.20	A program belépési pontját tartalmazó fájl.
model/Active.java	3 KB	2023.05.27	Aktív objektumot reprezentáló modell fájl.
model/Cistern.java	1 KB	2023.05.14	Ciszterna objektumot reprezentáló modell fájl.
model/Game.java	4 KB	2023.05.27	A játék működését reprezentáló modell fájl.
model/ Map.java	4 KB	2023.05.22	A játékkeret reprezentáló modell fájl.
model/Mechanic.java	3 KB	2023.05.20	A Szerelőt reprezentáló modell fájl.
model/Pipe.java	9 KB	2023.05.27	A csövet reprezentáló modell fájl.
model/Platform.java	4 KB	2023.05.20	Egy egységnyi területet reprezentáló modell fájl.
model/Player.java	3 KB	2023.05.20	A játékost reprezentáló modell fájl.
model/Pump.java	4 KB	2023.05.27	A pumpát reprezentáló modell fájl.
model/Saboteur.java	2 KB	2023.05.14	A szabotőrt reprezentáló modell fájl.
model/ Spring.java	1 KB	2023.05.21	A forrást reprezentáló modell fájl.
model/Team.java	2 KB	2023.05.20	Egy csapatot reprezentáló modell fájl.
util/GameObject.java	2 KB	2023.05.22	
util/Point.java	4 KB	2023.05.22	Egy játék beli pontot reprezentáló fájl.
util/Randomizer.java	1 KB	2023.05.14	Egy random szám generátort reprezentáló fájl.
util/UpdateListener.java	1 KB	2023.05.20	Egy léptetést megvalósító interfészt magába foglaló fájl.
view/CisternG.java	2 KB	2023.05.20	Egy grafikus Ciszterna objektumot megvalósító fájl.

view/Drawable.java	1 KB	2023.05.20	Egy kirajzolhatóságot megvalósító interfészt magába foglaló fájl.
view/GameFrame.java	7 KB	2023.05.27	A játék framet megvalósító fájl.
view/GamePanel.java	1 KB	2023.05.27	A játék panelt megvalósító fájl.
view/ImageLoader.java	1 KB	2023.05.20	A grafikus megjelenítéshez szükséges képbetöltőt megvalósító fájl.
view/MechanicG.java	2 KB	2023.05.27	A szerelőt, mint grafikus objektumot megvalósító fájl.
view/PipeG.java	5 KB	2023.05.27	A csövet, mint grafikus objektumot megvalósító fájl.
view/PumpG.java	3 KB	2023.05.27	A pumpát, mint grafikus objektumot megvalósító fájl.
view/SaboteurG.java	2 KB	2023.05.27	A szabotőr karaktert, mint grafikus objektumot megvalósító fájl.
view/SpringG.java	2 KB	2023.05.21	A ciszternát, mint grafikus objektumot megvalósító fájl.
view/StartFrame.java	4 KB	2023.05.27	A játék kezdeti felületét megvalósító fájl.
resources/activeIndicator.png	858 bájt	2023.05.25	Aktív állapotot jelölő kép.
resources/camel.png	899 KB	2023.05.25	Háttérkép.
resources/cistern.png	22.7 KB	2023.05.25	Ciszterna képe.
resources/leakyPIpe.png	17.9 KB	2023.05.25	Lyukas cső képe.
resources/leakyPIpeButSlippery.png	52.4 KB	2023.05.25	Lyukas és csúszós cső képe.
resources/leakyPIpeButSticky.png	30.1 KB	2023.05.25	Lyukas és ragadós cső képe.
resources/pipe.png	19.7 KB	2023.05.25	Cső képe.
resources/pump.png	1.71 KB	2023.05.25	Pumpa képe.
resources/pumpButBroken.png	1.44 KB	2023.05.25	Elromlott pumpa képe.
resources/pumpInput.png	834 bájt	2023.05.25	Pumpa bementének képe.
resources/pumpOutput.png	834 bájt	2023.05.25	Pumpa kimenetének képe.
resources/spring.png	24.7 KB	2023.05.25	Forrás képe.
resources/valve.png	4.79 KB	2023.05.25	Szelep képe.
resources/valveButShield.png	37.9 KB	2023.05.25	Védett szelep képe.
resources/valveButSlippery.png	27.7KB	2023.05.25	Csúszós szelep képe.

resources/valveButSlipperyButShield.png	55.3 KB	2023.05.25	Csúszós és védett szelep képe.
resources/valveButSticky.png	20.3 KB	2023.05.25	Ragadós szelep képe.
resources/valveButStickyButShield.png	48.5 KB	2023.05.25	Ragadós és védett szelep képe.
resources/villager.png	6.96 KB	2023.05.25	Szerelő képe.
resources/vindicator.png	13.00KB	2023.05.25	Szabotőr képe.
resources/waterfall.png	725 KB	2023.05.25	Vízesés képe.

13.1.2 Fordítás és telepítés

A fordításhoz a számítógépen, Windows 10 operációs rendszer alatt van lehetőség a következő lépések szerint:

1. Windows parancssor (azaz cmd.exe) megnyitása
2. Az aktuális mappába (forráskód főmappája) való navigálás után a következő parancsot kell futtatni: `javac model/*.java mainPackage/*.java util/*.java view/*.java`
3. Ha minden rendben volt, azaz nincs hibaüzenet, akkor sikeresen megtörtént a fordítás.

13.1.3 Futtatás

A Futtatáshoz a számítógépen, Windows 10 operációs rendszer alatt van lehetőség a következő lépések szerint:

1. Windows parancssor (cmd) megnyitása
2. Az aktuális mappába (forráskód főmappája) való navigálás után a következő parancsot kell futtatni: `java mainPackage/Main.java`
3. Ha nincs hibaüzenet akkor elindult a szoftver, és kis idő elteltével meg fog jelenni a grafikus felület.

13.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Gegő Levente	LYNUXB	23
Kovács Zoltán	W4YDIT	23
Koleszár Kristóf	I1TLYH	23
Tóth Gábor	F041OM	8
Szautner Béla László	EEID48	23

13.3Napló

Kezdet	Időtartam	Résztevők	Leírás
2023.05.20 13:00	10 óra	Kovács Gegő	Kódolás
2023.05.23 20:00	2 óra	Kovács Gegő	Kódolás befejezése
2023.05.29 01:00	2,5 óra	Szautner	Dokumentum elkészítése
2023.05.26. 10:00	1 óra	Koleszár	Dokumentum javítása