

3. Analízis modell (II. változat)

5 – runtime_error

Konzulens:

Dobos-Kovács Mihály

Csapattagok

Mizser Ádám Zoltán	SHKGZW	mizser.adam@gmail.com
Tepliczky Olivér Zoltán	WB6LC5	tepliczkyo@edu.bme.hu
Fekete Álmos Valér	KR5WPC	feketealmos0@gmail.com
Váradi Kristóf	BP17IB	kristofvaradi@edu.bme.hu
Sasvári Szabolcs Attila	TWOZG6	szabolcs.attila.sasvari@edu.bme.hu

2023.03.26.

Analízis modell kidolgozása

3.1 Objektum katalógus

fontos megjegyzés: A Game (játék) objektum a játékvezérlő kontrollert valósítja meg. A nem játékosokhoz köthető használati esetekért felel, illetve ez a híd a View és Model között. Bár nem képezi az analízis modell részét, meg kell jelenítenünk, mert számos szekvenciadiagramhoz elengedhetetlen, valamint ez a tervezést is jelentősen megkönnyíti.

3.1.1 Játék

A játékmenetet kezeli (vizsgálja, hogy véget ért-e a játék, valamint a köröket vezérli). Számontartja és módosítja a csőrendszer felépítését (turn-önként véletlenszerűen pumpákat ront el, round-onként pedig ciszternákkal szomszédos üres helyekre új csöveket helyez). A turn-ök elején a forrásokból áramoltatja a vizet a csövekig, vagy ciszternáig. Pontokat oszt a csapatoknak, és számon is tartja azokat.

Írányokat / célpontokat tud bekérni a felhasználótól. Játékosmozgatást, cső- és pumpahelyezést, illetve csőfelvételt kezdeményezhet vele. Pumpát adhat a ciszternán található szerelőnek, ha üres a tárolója.

3.1.2 Csőrendszer

Számontartja a játék elemeit (csöveket, pumpákat, forrásokat és ciszternákat). Ezek közül eltávolíthat csöveket, ha a csőrendszer ettől még összefüggő marad. Az eltávolított csövet odaadja annak a szerelőnek, aki az eltávolítást kezdeményezte. Kezeli a csövek és pumpák elhelyezését, valamint az erre irányuló utasítások helyességét.

3.1.3 Forrás

Olyan elem, amely vizet juttat a vele szomszédos csövekbe, azaz a számontartott kimeneteibe, illetve a sivatagba, ha nincsen szomszédja az adott irányban. Ahányszor előfordul az utóbbi, annyi pontot oszt a szabotőrök csapatának.

Játékosokat tárol és távolít el az adott pozícióról.

3.1.4 Ciszterna

Olyan elem, amely számontartja a bemeneteit. Amennyi csőből folyik víz az adott ciszternába, annyi pontot oszt a szerelők csapatának.

Számontartja a játékosokat, akik rajta állnak. Játékosokat tud feltenni, illetve eltávolítani magáról.

3.1.5 Pumpa

Olyan elem, amely számontartja a bemenetét, a kimenetét és a szomszédait. A szomszédai közül be tudja állítani, hogy melyik a bemenete és melyik a kimenete. Víz képes juttatni a bemeneti csővéből, önmagán keresztül (ő maga is tárol vizet) a kimeneti csővébe. El tud romlani, és ekkor nem képes víz átengedésére. Amikor víz megy át rajta, pontot oszthat a szabotőrök csapatának, ha a kimenete egy szabad végű / lyukas cső vagy sivatag (nincs kimenete).

Számontartja a játékosokat, akik rajta állnak ezzel együtt játékosokat tud feltenni, illetve eltávolítani magáról.

3.1.6 Cső

Olyan elem, amely számontartja a bemenetét és kimenetét, és az előbbiből az utóbbiba vizet képes juttatni. Tárolja a benne lévő vizet. Ki lehet lyukasztani, és meg is lehet javítani. Lyukas állapotában nem képes vizet átengedni magán. A víz átfolyásakor pontot oszt a szabotőröknek, amennyiben a cső lyukas, vagy kimenete a sivatag.

Számontartja a játékost, aki rajta áll, valamint egy játékost tud feltenni, illetve eltávolítani magáról (egyszerre egy játékos lehet egy csövön).

3.1.7 Szerelő

Számontartja azt az elemet, amin éppen áll. Szomszédos elemre léphet, hogyha ez lehetséges (aza, ha a szomszédos csövön esetleg nem áll másik játékos). A ciszternák között direkt módon mozoghat.

Manipulálni tudja az elemet, amin éppen áll (csövet és pumpát tud javítani, illetve működő pumpát tud átállítani).

A tárolója tárolhat vagy egy csövet, vagy egy pumpát. El tud távolítani egy szomszédos, üres csövet, hogy ha a tárolója üres, valamint ezt az csövet később elhelyezni egy szomszédos, üres helyre. Pumpát olyan cső helyére tud letenni, aminek a bemenete és a kimenete is egy cső.

3.1.8 Szabotőr

Számontartja azt az elemet, amin éppen áll. Szomszédos elemre léphet, hogyha ez lehetséges (aza, ha a szomszédos csövön esetleg nem áll másik játékos). A ciszternák között direkt módon mozoghat.

Manipulálni tudja azt az elemet, amin éppen áll (csövet tud lyukasztani, illetve pumpát tud átállítani).

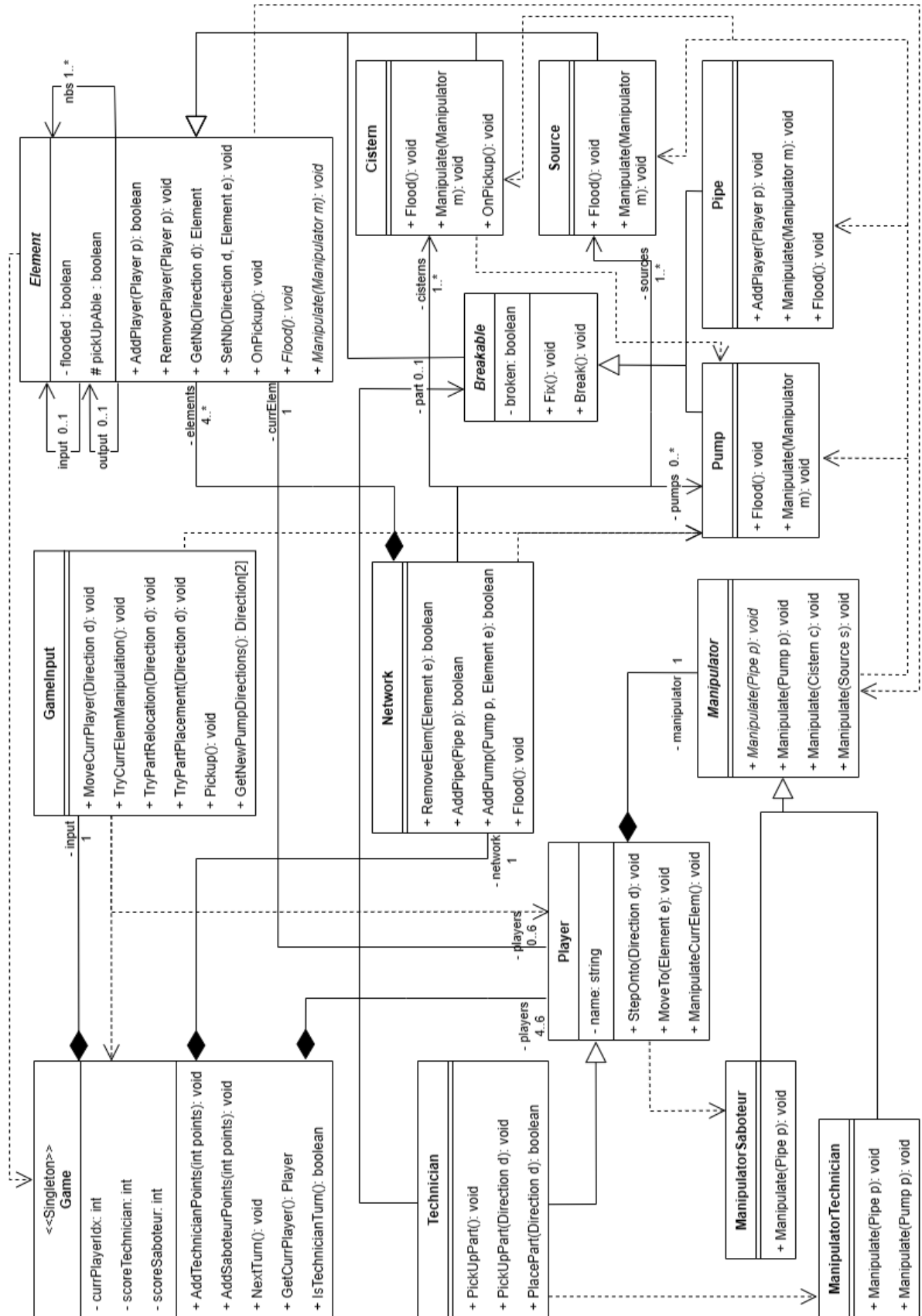
3.2 Statikus struktúra diagramok

fontos megjegyzések:

- A geometriát igyekeztünk eltávolítani. A Direction típus a modell szempontjából tetszőleges absztrakció lehet.
- A triviális (paraméter nélküli/egyparaméteres) setter/getter függvényeket nem jelöltük, viszont az "Osztályok leírása" szekcióban fel vannak ezek is sorolva.
- Ha egy ősnek van egy dependenciája, akkor az érvényes minden leszármazottra. A dependenciákat csak akkor jelöltük, ha nincsen erősebb kapcsolat az osztályok között (leszármazás, kompozíció, aggregáció, asszociáció).

osztálydiagram magyarázatok:

- GameInput-nak akár a Game része is lehetne, de számunkra ez egy fontos logikai elkülönítés:
 - GameInputba azok a függvények kerültek, amik az interfészt szolgáltatják a View és Controller között (kommunikáció a felhasználóval).
 - Game-be közvetlenül pedig azok a függvények kerültek, amik a Controller részei.
- A visitor minta egy trade-off: a jelenlegi osztályhierarchiát bonyolítja, de új elemek potenciális megjelenésével a Player és leszármazott osztálya(i) könnyen túlságosan összetetté válhatnának, ha nem szerveznénk ki az elemek manipulációjának logikáját önálló osztályba.



3.3 Osztályok leírása

3.3.1 Breakable

- **Felelősség**

Ez az elem elromolhat, és értelemszerűen meg is lehet javítani. Ilyen elemet tudnak tárolni a szerelők (más néven part).

- **Ősosztályok**

Element

- **Attribútumok**

- - **broken: boolean** azt jelzi, hogy az adott elem hibás-e vagy sem, azaz át tud-e eresztetni magán vizet, vagy sem.

- **Metódusok**

- + **void Fix()** : Hibás elem megjavítására szolgáló függvény.
- + **void Break()** : Működőképes elem elrontására szolgáló függvény.
- + **void GetBroken()** : Visszaadja, hogy az adott elem működőképes-e.

3.3.2 Cistern

- **Felelősség**

Amennyi csőből folyik víz ebbe az elembe, annyi pontot oszt a szerelők csapatának. Erről az elemről a játékosok közvetlen más ciszternákra tudnak lépni.

- **Ősosztályok**

Element

- **Metódusok**

- + **void OnPickUp()** : megpróbál pumpát adni a soron lévő játékos tárolójába.
- + **void Flood()** : Vízzel tölti fel magát, és pontot ad a szerelőknek. A pálya végpontjaként nem áramoltat más elemekbe vizet.
- + **void Manipulate(Manipulator m)** : A paraméterként kapott manipulátorral manipulálja ezt a konkrét elemet.

3.3.3 Element

- **Felelősség**

Player-eket tud felrakni, illetve eltávolítani magáról, és számon is tartja a rajta tartózkodókat. Tárolja a bemenet/kimenetét, valamint a szomszédjait is, ezeket tudja változtatni, illetve lekérdezni. Továbbá tárolja, hogy van-e víz benne.

- **Attribútumok**

- - **flooded: boolean** : Értéke igaz, ha víz van benne (jelenleg vizet tárol).
- # **pickUpAble: boolean** : Értéke igaz, ha az adott elem felvehető, különben hamis.
- - **players: Player[0..6]** : Azon játékosok halmaza, amelyek az adott elemen tartózkodnak

- - **nbs: Element[1..*]** : A vele szomszédos elemekre hivatkozik.
- - **input: Element[0..1]**: A bemenő szomszédra hivatkozik, ha van ilyen.
- - **output: Element[0..1]** : A kimenő szomszédra hivatkozik, ha van ilyen.

- **Metódusok**

- **+ void AddPlayer(Player p)** : Felhelyez magára egy játékost, ha ez lehetséges a rajta állók száma szerint.
- **+ void RemovePlayer(Player p)** : Eltávolítja a számontartott játékosai közül azt, amelyiket paraméterként kapott.
- **+ Element GetNb(Direction d)** : Megadott irányba lévő szomszédos elemét adja vissza, ha van ilyen.
- **+ void SetNb(Direction d, Element e)** : Megadott irányú szomszédjának állítja be a paraméterként kapott elemet.
- **+ void OnPickup()** : Ha egy elem támogatni akarja azt a funkcionalitást, hogy egy szerelő rajta állva egy part-ot kaphasson a tárolójába, akkor ezt a függvényt kell felülírnia. Alapértelmezett megvalósítása üres törzsű függvény (alapjáraton nem támogatják az elemek ezt a funkcionalitást).
- **+ void Flood()** : Absztrakt metódus, amely a víz áramoltatásáért felelős.
- **+ void Manipulate(Manipulator m)** : Absztrakt metódus. A paraméterként kapott manipulátorral manipulálja az elemet.
- **+ boolean GetPickUpAble()** : Visszaadja, hogy az elem felvehető-e (pickUpAble és nincs benne víz)
- **+ int GetNbCnt()** : A szomszédos elemek számát adja vissza.
- **+ int GetPlayerCnt()** : A rajta tartózkodó játékosok számát adja vissza.
- **+ boolean GetFlooded()** : Igazat ad vissza, ha az víz van benne (vagy víz folyik ki belőle), egyéb esetben hamisat.
- **+ Element GetInput()** : Visszaadja a bemeneti elemét.
- **+ Element GetOutput()** : Visszaadja a kimeneti elemét.
- **+ void SetFlooded(boolean f)** : Az átadott érték szerint állítja be, hogy van-e víz jelenleg benne.
- **+ void SetInput(Element e)** : A bemenetét az átadott elemre állítja be.
- **+ void SetOutput(Element e)** : A kimenetét az átadott elemre állítja be.

3.3.4 Game

- **Felelősség**

A játékmenetet kezeli (vizsgálja, hogy véget ért-e a játék, valamint a köröket vezérli). Számontartja és módosítja a csőrendszer felépítését (turn-önként véletlenszerűen pumpákat ront el, round-onként pedig ciszternákkal szomszédos üres helyekre új csöveket helyez). A turn-ök elején a forrásokból áramoltatja a vizet a csövekig, vagy ciszternáig. Pontokat oszt a csapatoknak, és számon is tartja azokat.

- **Attribútumok**

- **- currPlayerIdx: int** : A soron lévő játékos indexe a „players” gyűjteményben.
- **- scoreTechnician: int** : A szerelők pontszámát tárolja.
- **- scoreSaboteur: int** : A szabotőrök pontszámát tárolja.
- **- input: Gameinput** : A felhasználóval kommunikáló objektum. (Híd a View és a Model között.)
- **- players: Player[4..6]** : A játékban résztvevő játékosokat tárolja.
- **- network: Network** : A pálya elemeit tárolja, szortírozza.

- **Metódusok**

- **+ void AddTechnicianPoints(int points)** : Ez a metódus ad a szerelőknek pontot.

- + **void AddSaboteurPoints(int points)** : Ez a metódus ad a szabotőröknek pontot.
- + **void NextTurn()** : A következő turn indítása (áramoltatja a vizet, pontokat oszt, véletlenszerűen pumpákat ront el, és ha véget ért egy teljes kör (round), akkor új csöveket teremt a ciszternák üres szomszédjai helyén). Ha az egyik csapat elérte a győzelemhez szükséges pontok számát, véget vet a játéknak.
- + **Player GetCurrPlayer()** : Visszaadja a játékost, aki éppen soron van (akinek turn-je van jelenleg).
- + **boolean IsTechnicianTurn()** : Igazat ad vissza, ha éppen szerelő jön az adott körben.
- + **GameInput GetInput()** : Visszaadja a felhasználóval kommunikáló objektumot.
- + **Network GetNetwork** : Visszaadja a pálya elemeit számontartó objektumot.

3.3.5 GameInput

- **Felelősség**

A felhasználót és a játékot köti össze. Irányokat / célpontokat tud bekérni a felhasználótól. Játékosmozgatást, cső- és pumpalehelyezést, illetve csőfelvételt kezdeményezhet vele. Pumpát is tud adni a jelenlegi játékosnak, amennyiben egy szerelő, ciszternán áll és üres a tárolója.

- **Metódusok**

- + **void MoveCurrPlayer(Direction d)** : d irányba próbálja mozgatni az épp soron lévő játékost, arról az elemről, amelyiken éppen tartózkodik.
- + **void TryCurrElemManipulation()** : akkor hívandó függvény, amikor éppen azt az elemet szeretné manipulálni / interakcióba lépni vele (csövet lyukasztani / foltozni, pumpát javítani / átállítani vagy a következő ciszternára ugrani) a soron lévő játékos, amelyiken éppen áll.
- + **void TryPartRelocation(Direction d)** : akkor hívandó függvény, amikor a soron lévő játékos megpróbál egy tőle d irányba lévő szomszédos csövet felvenni, és a tárolójában eltárolni.
- + **void TryPartPlacement(Direction d)** : akkor hívandó függvény, amikor a soron lévő játékos megpróbálja a tárolt part-ját tőle d irányba lehelyezni.
- + **void Pickup()** : akkor hívandó függvény, amikor egy játékos egy part-ot próbálna a tárolójába tenni (nem a játéktérrel, hanem közvetlenül odaadva). Ez csak akkor lesz sikeres, ha a játékos szerelő, üres a tárolója, és az elem amin áll, támogat ilyen funkcionalitást.
- + **Direction[2] GetNewPumpDirections()** : pumpa átállításához visszaadja a soron lévő játékos által bevitt irányokat.

3.3.6 Manipulator

- **Felelősség**

Absztrakt osztály, amely a Visitor tervezési mintát valósítja meg a leszármazottaival együtt. A leszármazottai egy játékos típusnak készítik el a „gazda” elem manipulálását megvalósító viselkedést minden elem esetére (pumpák, csövek, források, ciszternák).

A szerelők konkrét manipulátora például definiálja, hogy mit tud tenni egy szerelő, ha az előbb említett elemeken állva próbál interaktálni. A függvényei gyakran a kört is léptetik.

- **Metódusok**

- + **void Manipulate(Pipe p)** : Absztrakt metódus a játékosok cső manipulálásának leírására.
- + **void Manipulate(Pump p)** : Átállítja az átadott pumpát (GameInput-ot használja a bemenetért). Ezután véget ér a játékos köre (turn).
- + **void Manipulate(Cistern c)** : Átlépteti a jelenlegi játékost a következő ciszternára. Ezzel nem ér véget a játékos köre (turn).
- + **void Manipulate(Source s)** : Üres törzsű függvény, jelenleg a játékosok nem tesznek semmit a forráson állva. (A jövőbeli bővíthetőségre fenntartva, illetve a Dynamic Dispatch hibátlan működése miatt kell, hogy ne kelljen Type-checkingelni a hívóoldalon.)

3.3.7 ManipulatorSaboteur

- **Felelősség**

A szabotőrök konkrét manipulátora, ami definiálja, hogy hogyan manipulálhatják az összes lehetséges „gazda” elemüket.

A csövek kilyukasztásának viselkedését valósítja meg. A többi viselkedés megfelel az ősbélivel.

- **Ősosztályok**

Manipulator

- **Metódusok**

- + **void Manipulate(Pipe p)** : Kilyukasztja az átadott p csövet. Utána pedig véget ér a jelenlegi játékos köre.

3.3.8 ManipulatorTechnician

- **Felelősség**

A szerelők konkrét manipulátora, ami definiálja, hogy hogyan manipulálhatják az összes lehetséges „gazda” elemüket.

A csövek javításának viselkedését valósítja meg, illetve a pumpák megjavításának viselkedésével helyettesíti a pumpaátállítást, ha az adott pumpa rossz.

A többi viselkedés megfelel az ősbélivel.

- **Ősosztályok**

Manipulator

- **Metódusok**

- + **void Manipulate(Pipe p)** : Megjavítja az átadott p csövet. Utána pedig véget ér a jelenlegi játékos köre.
- + **void Manipulate(Pump p)** : Megjavítja az átadott p pumpát, ha az elromlott, különben pedig átállítja (ez esetben meghívja az ősbéli megvalósítását a függvénynek).

3.3.9 Network

- **Felelősség**

Tárolja a pálya elemeit. Csövek és pumpák adhatók hozzá, a csöveket pedig el is lehet távolítani róla. Vizsgálja, hogy ezek a műveletek lehetségesek-e egyáltalán.

A benne tárolt elemek azok, amik ténylegesen meg is lesznek jelenítve a pályán.

Továbbá ez az osztály indítja el a vízfolyást forrásokból.

- **Attribútumok**

- - **elements: Element[4..*]** : A pálya összes elemét tárolja.
- - **cisterns: Cistern[1..*]** : Külön csoportosítja minden elem közül a ciszternákat ez a gyűjtemény.
- - **sources: Source[1..*]** : Külön csoportosítja minden elem közül a forrásokat ez a gyűjtemény.

- - **pumps: Pump[0..*]** : Külön csoportosítja minden elem közül a pumpákat ez a gyűjtemény.
- **Metódusok**
 - + **boolean RemoveElem(Element e)** : Megkísérli eltávolítani az átadott elemet a pályáról. Ha ennek semmi akadálya nem volt (pl. nem esne komponensekre a pálya, mint gráf), akkor igazzal tér vissza. Ellenkező esetben nem módosít semmi a pályán.
 - + **boolean AddPipe(Pipe p)** : Az átadott csövet megkísérli hozzáadni a pályához. Ennek sikerességét adja vissza.
 - + **boolean AddPump(Pump p, Element e)** : Az átadott pumpával kísérli meg felülírni a másik átadott elemet a pályán. Ennek sikerességét adja vissza.
 - + **void Flood()** : Felasztja az összes vizet a pályáról, majd minden forrásból elindítja a vizet, aminek következtében el lesz árasztva vízzel a pálya, és pontokat fognak kapni a csapatok.
 - + **Element[4..*] GetElements()** : Visszaadja a pálya minden elemét.
 - + **Pump[0..*] GetPumps()** : Visszaadja a pályán lévő pumpákat.
 - + **Cistern[1..*] GetCisterns()** : Visszaadja a pályán lévő ciszternákat.

3.3.10 Pipe

- **Felelősség**

Olyan elromolható, felvehető elem, amelyen legfeljebb egy játékos tartózkodhat.

Ki lehet lyukasztani, a lyukas csövet pedig meg is lehet javítani. Ha a cső nem lyukas akkor a kimenetéhez vizet tud juttatni.

Ha lyukas, vagy a kimenete üres, akkor a szabotőrök pontot kapnak, amikor víz érkezik belé.

- **Össztályok**

Element → Breakable

- **Metódusok**

- + **void AddPlayer(Player p)** : Felhelyezi az átadott játékost magára, ha nem áll rajta már más valaki.
- + **void Manipulate(Manipulator m)** : Az átvett manipulátorral manipulálja ezt a konkrét elemet.
- + **void Flood()** : Vízrel tölti meg magát. Ha lyukas a cső vagy nincs output-ja akkor pontot kapnak a szabotőrök, különben pedig hívja tovább az output-ján a Flood() függvényt (ereszti tovább a vizet).

3.3.11 Player (Saboteur)

- **Felelősség**

A Player osztály a szabotőr osztállyal ekvivalens, tehát a szabotőrök csapatának játékosai Player típusúak. Eltárolja a játékos nevét, az elemet, amelyen éppen áll, és a manipulátorát. Tud mozogni irányokba, és manipulálni tudja a „gazda” elemét.

- **Attribútumok**

- - **currElem: Element** : Az adott elem, amelyen a játékos tartózkodik
- - **name: string** : A játékos neve, azonosítója
- - **manipulator: Manipulator** : A játékos által használt manipulator objektum a „gazda” elemmel való interakciók lekezeléséhez.

- **Metódusok**

- + **void MoveTo(Element e)** : Az átadott elemre helyezi a játékost, ha ez lehetséges.
- + **void StepOnto(Direction d)** : A játékost a megadott irányban lévő elemre lépteti, ha ez lehetséges.
- + **void ManipulateCurrElem()** : Manipulálja azt az elemet, amelyen éppen tartózkodik. A manipulátora határozza meg, hogy mely „gazda” elem típus esetén mit tesz.
- + **Element GetCurrElem()** : Visszaadja az elemet, amelyen a játékos éppen tartózkodik.

3.3.12 Pump

- **Felelősség**

Olyan elromolható, felvehető elem, amely vizet képes áteresztetni a bemenetként beállított csővéből a kimenetként beállított csővébe. Pontot oszt a szabotőröknek, ha nincsen kimenete, amikor vizet eresztene át.

- **Ősosztályok**

Element → Breakable

- **Metódusok**

- + **void Flood()** : Ha a pumpa el van romolva, nem tesz semmit. Ha nincs elromolva akkor vízzel tölti fel magát. Ha nincsen outputja, akkor a szabotőrök pontot kapnak, de ha van outputja, akkor az outputra hívja tovább a Flood() függvényt.
- + **void Manipulate(Manipulate m)** : Az átvett manipulátorral manipulálja ezt a konkrét elemet.

3.3.13 Source

- **Felelősség**

Olyan elem, amely vizet juttat a vele szomszédos elemekbe, és soha sincsen bemenete.

- **Ősosztályok**

Element

- **Metódusok**

- + **void Flood()** : Vízzel tölti fel magát, és minden nem lyukas szomszédjára hívja tovább a Flood() függvényt, azaz ereszti tovább a vizet. Ha nem tudja ezt megtenni az adott szomszédja felé, mert lyukas, vagy nincs arra szomszédja, akkor pontot ad a szabotőröknek.
- + **void Manipulate(Manipulator m)** : Az átvett manipulátorral manipulálja ezt a konkrét elemet.

3.3.14 Technician

- **Felelősség**

Olyan játékos, aki csövek lyukasztása helyett javítani tudja őket, illetve pumpákat is. Emellett képes vagy egy csövet, vagy egy pumpát tárolni magánál, ami el is tud helyezni a megfelelő feltételek fennállása esetén.

- **Ősosztályok**

Player

- **Attribútumok**

- - **part: Breakable[0..1]** : Az az elem, amelyet a szerelő felvett a pályáról.

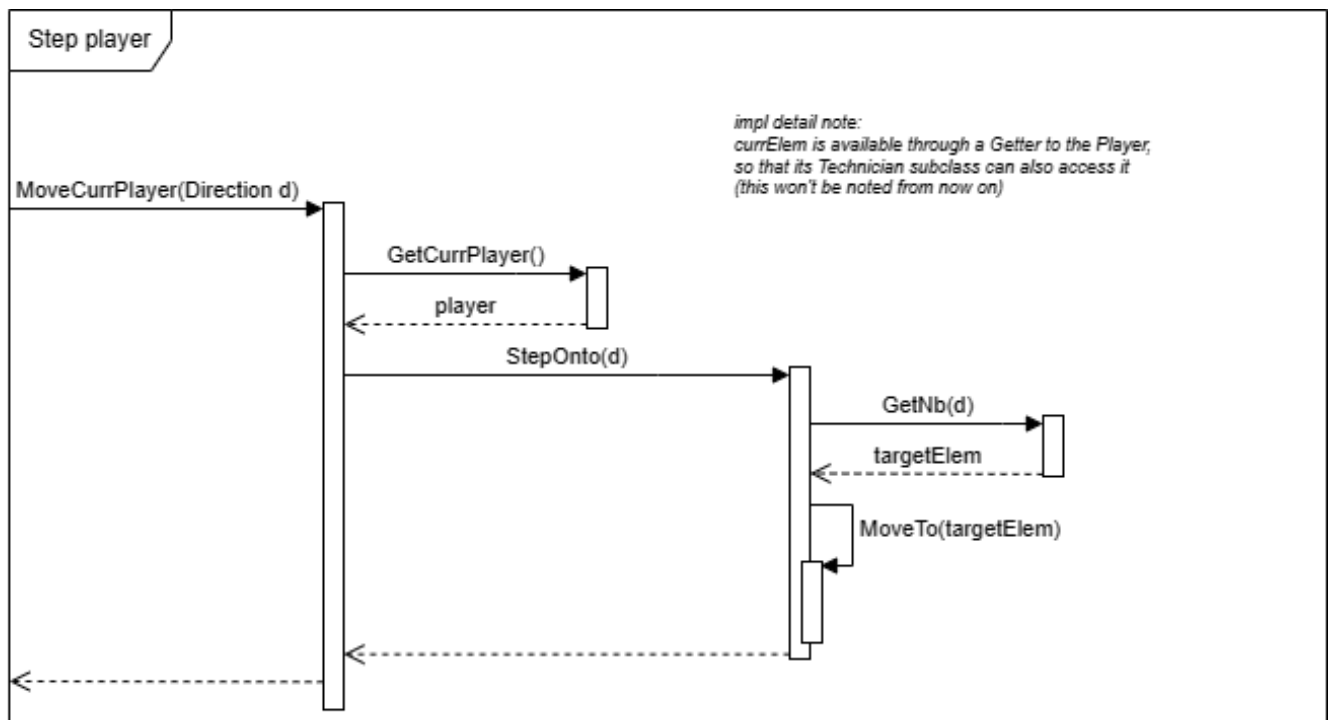
- **Metódusok**

- + **void PickUpPart()** : Megkísérel felvenni egy part-ot a jelenlegi elemtől.
- + **void PickUpPart(Direction d)** : Megkísérel felvenni a d irányban lévő part-ot.

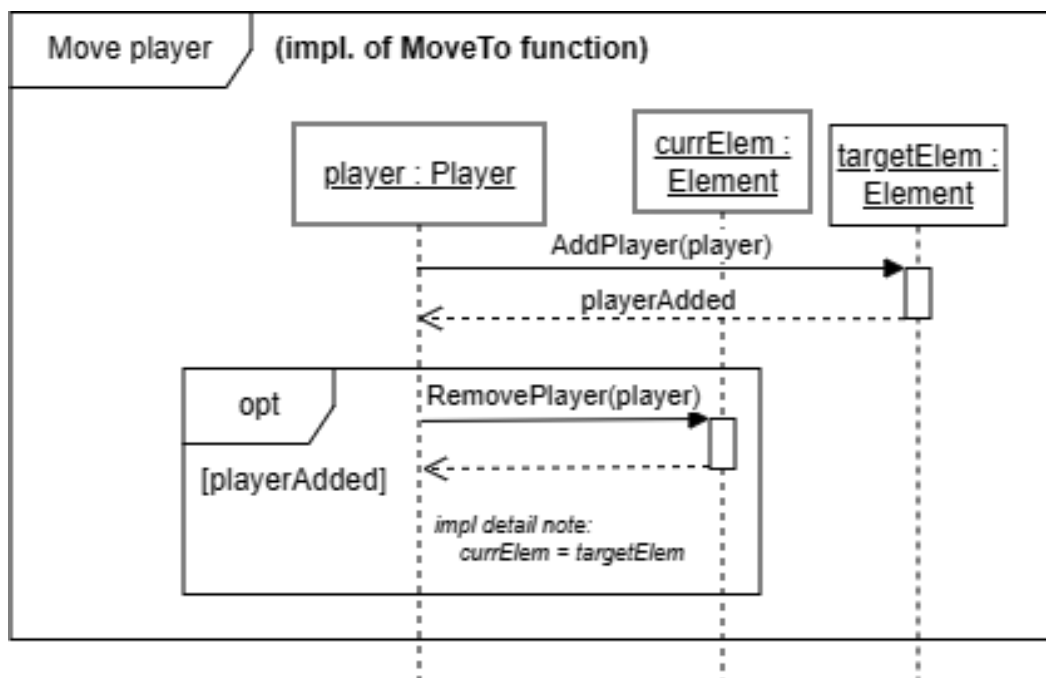
- + **boolean PlacePart(Direction d)** : Megkísérli elhelyezni a tárolt part-ját d irányba. A művelet sikerességével tér vissza.
- + **Breakable GetPart()** : Visszaadja azt a Breakable-t („part” attribútumot), ami a szerelőnél van.
- + **void SetPart(Breakable b)** : Az átadott Breakable-re állítja a „part” attribútumot.

3.4 Szekvencia diagramok

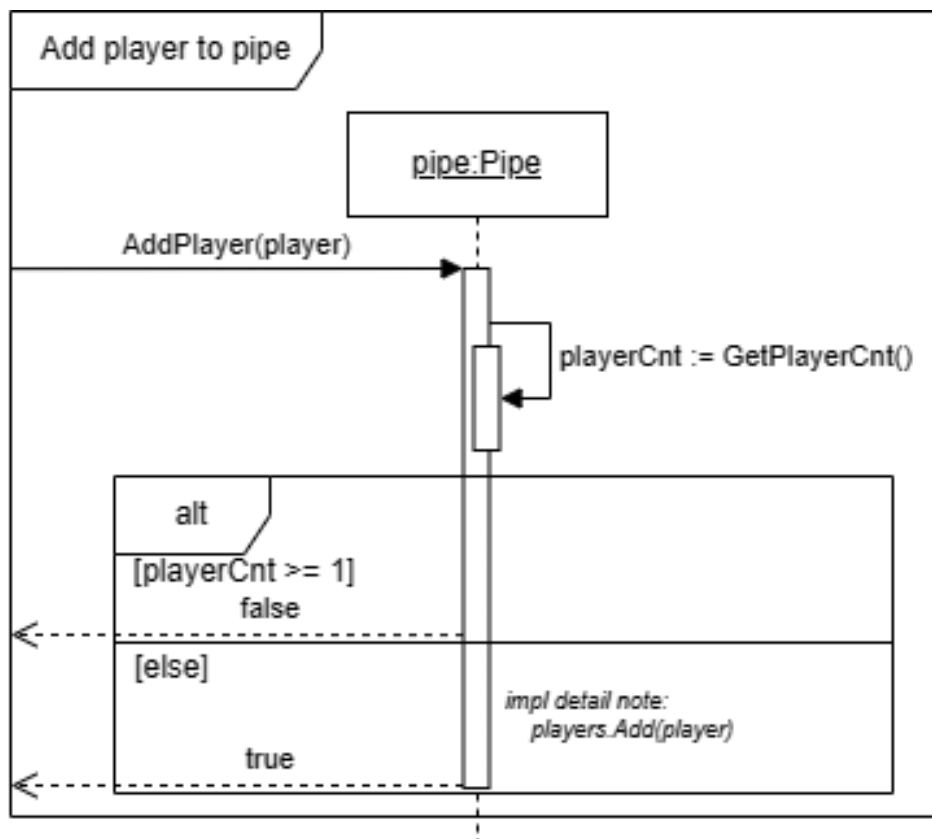
3.4.1 Step Player



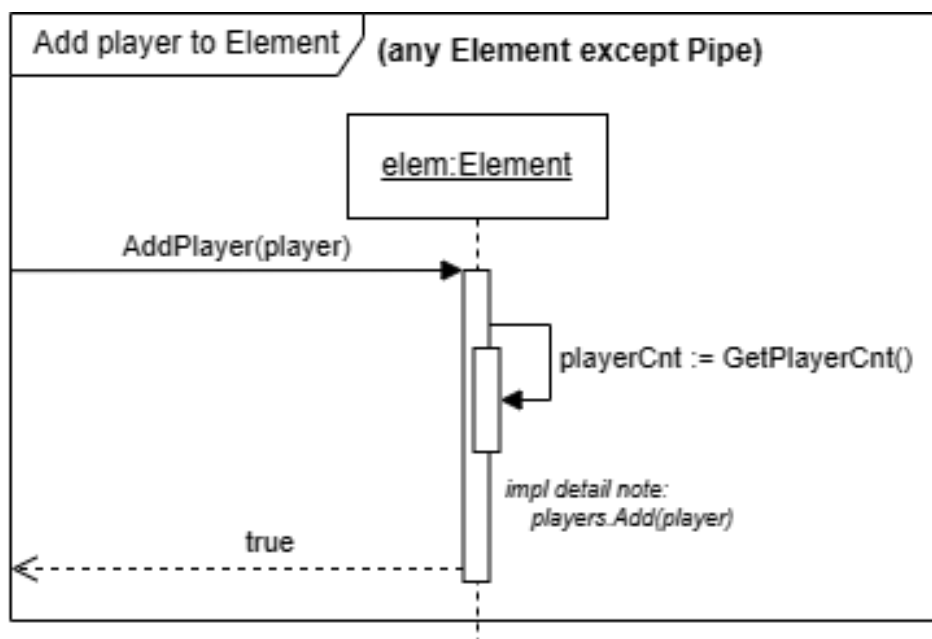
3.4.2 Move Player



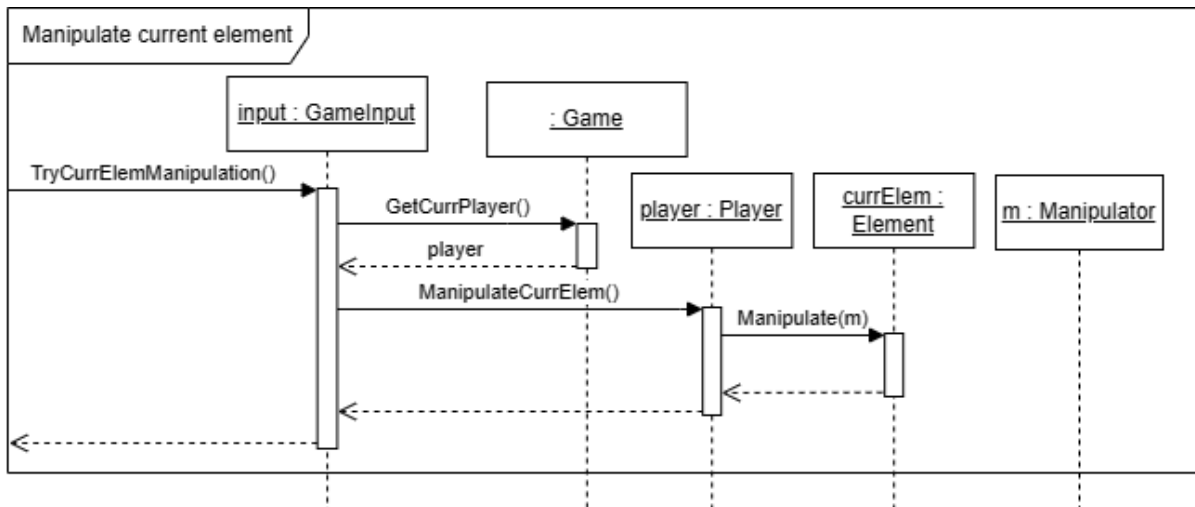
3.4.3 Add Player to Pipe



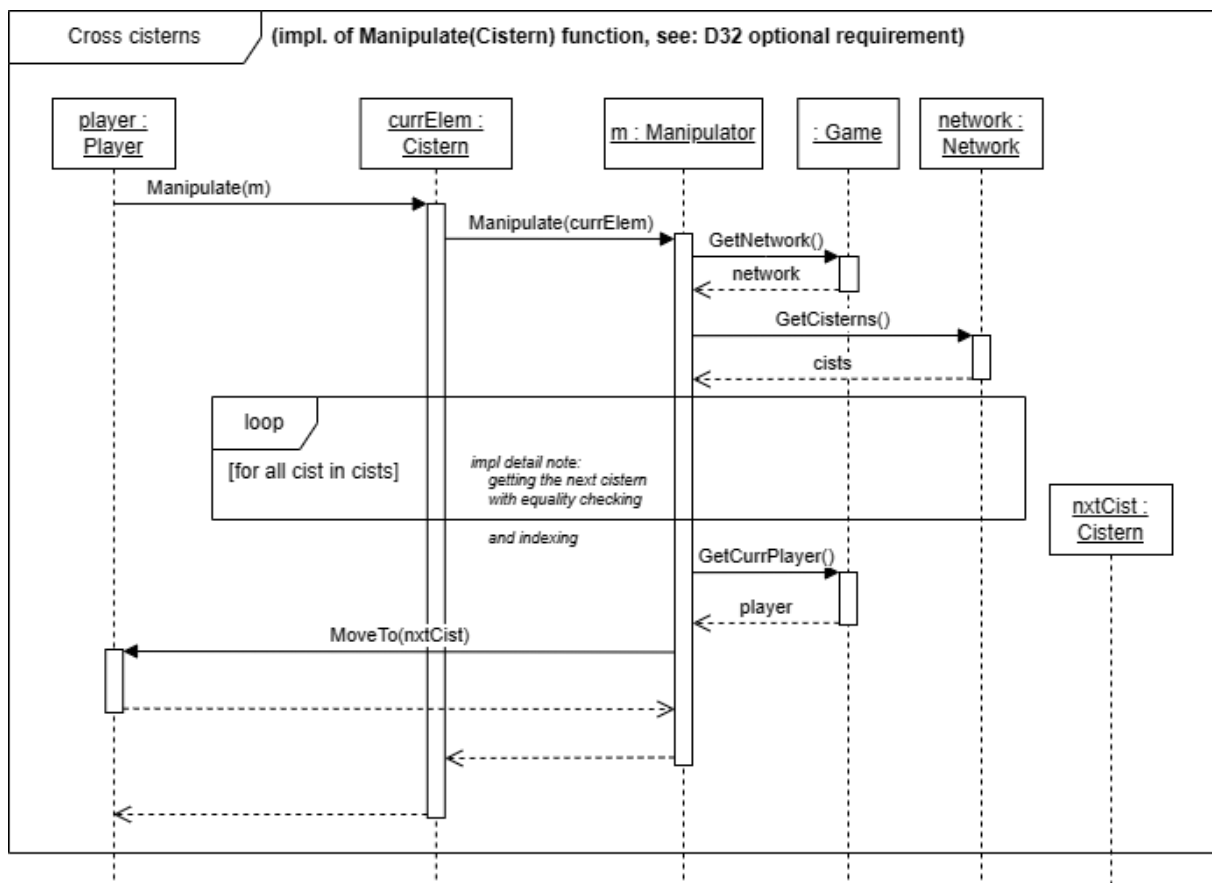
3.4.4 Add Player to Element



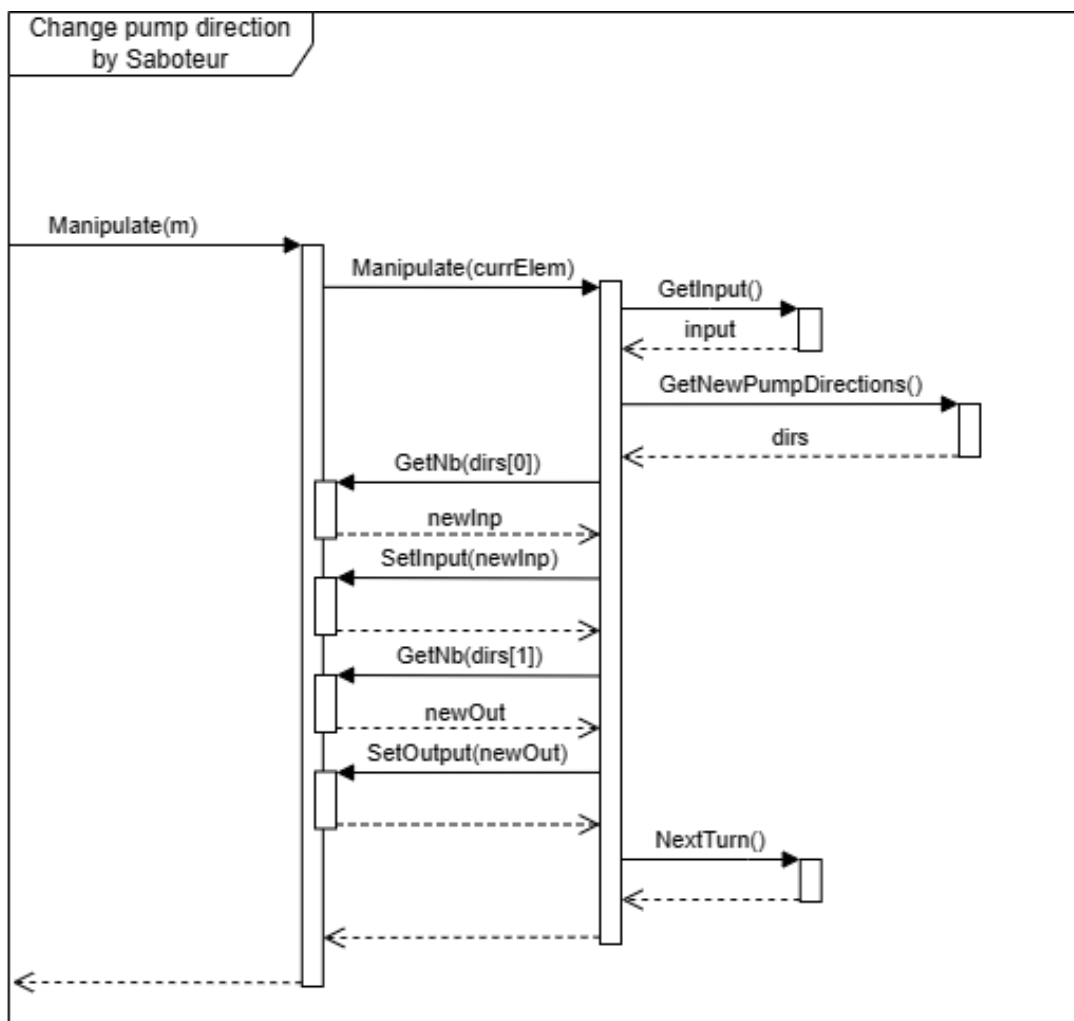
3.4.5 Manipulate Current Element



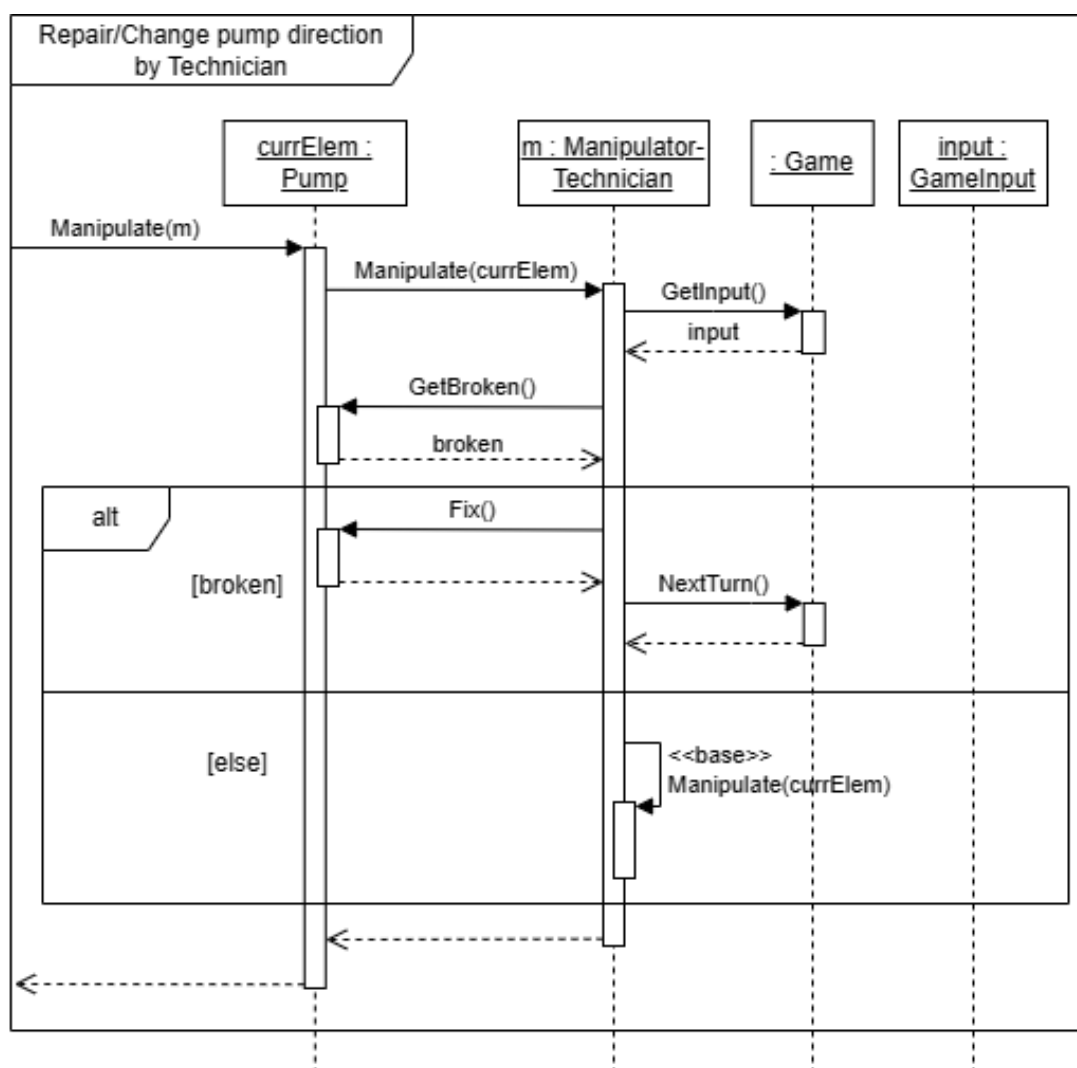
3.4.6 Cross Cisterns



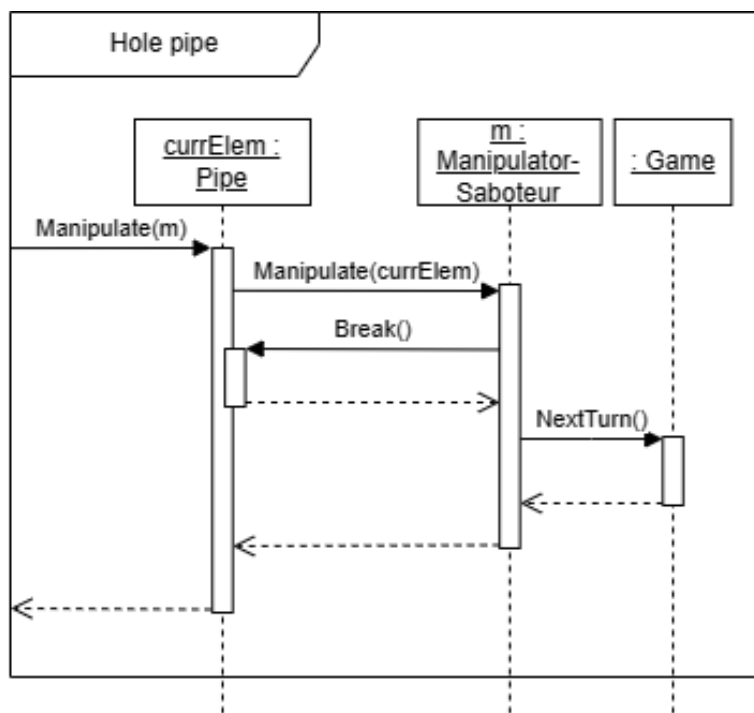
3.4.7 Change Pump Direction (by Saboteur)



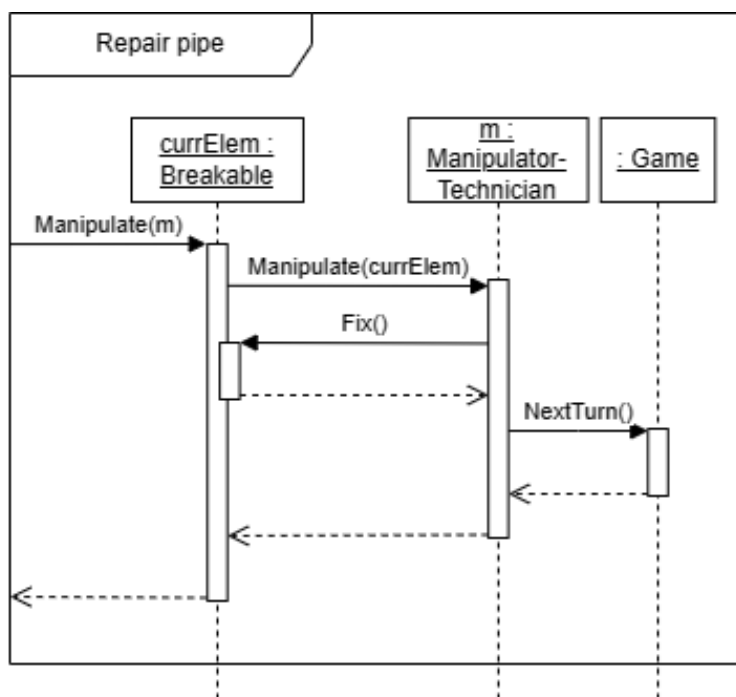
3.4.8 Repair or Change Pump Direction (by Technician)



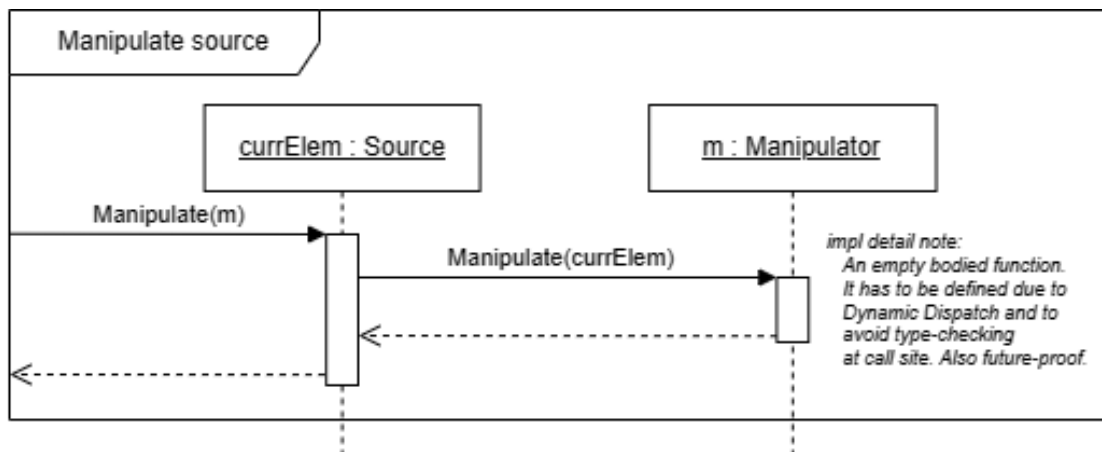
3.4.9 Hole Pipe



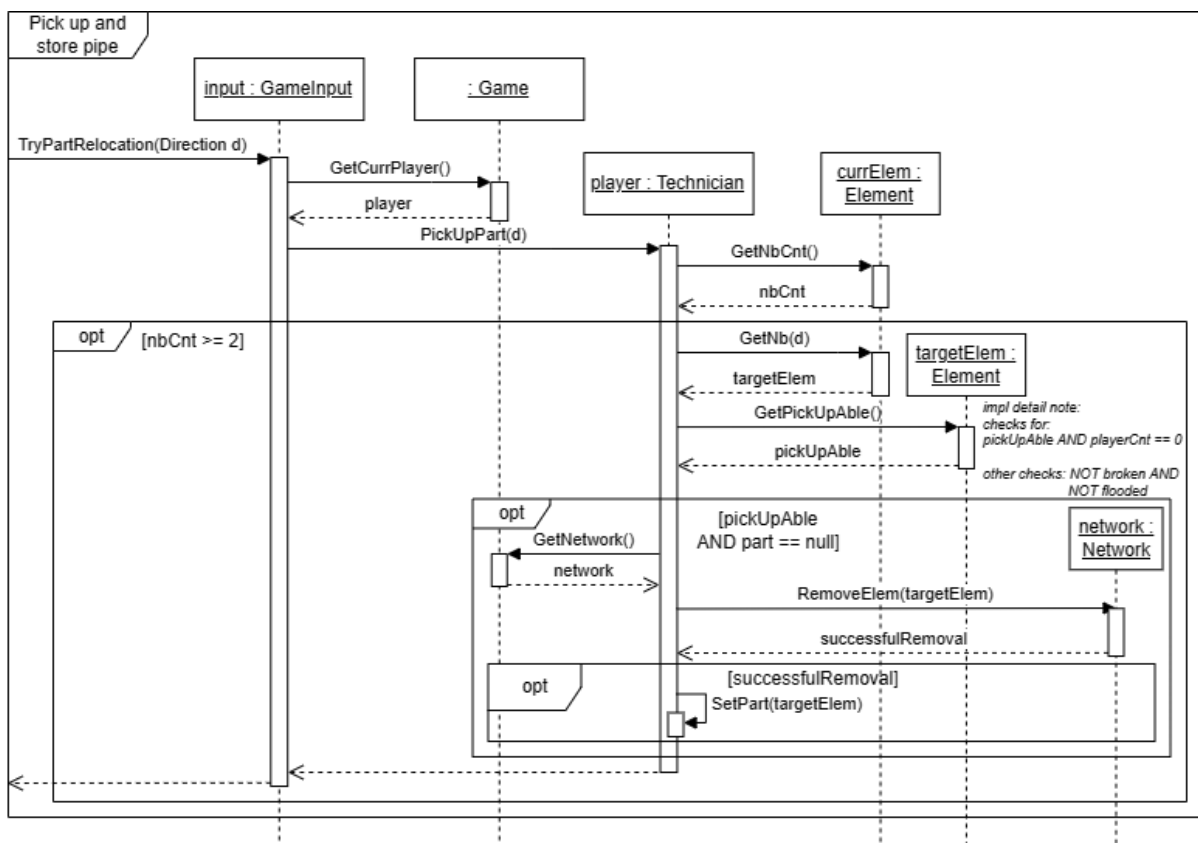
3.4.10 Repair Pipe

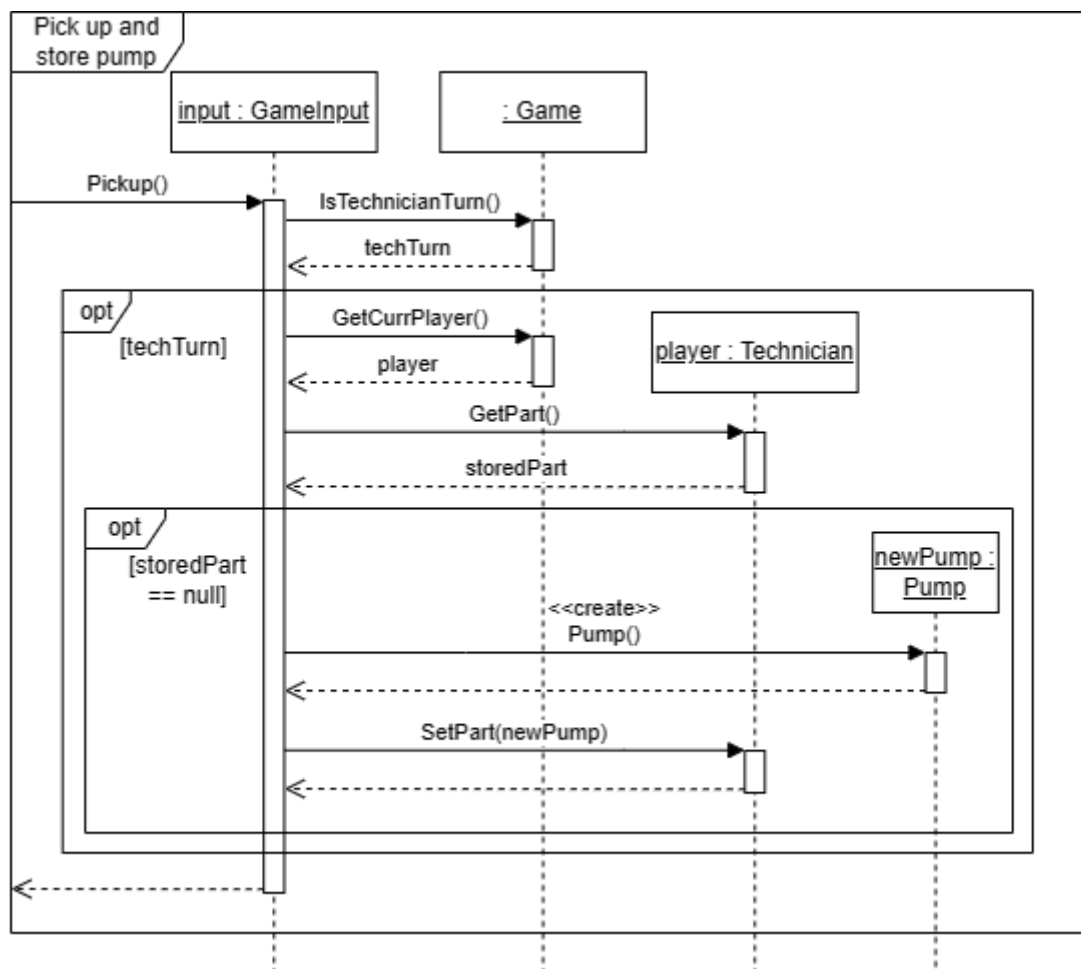


3.4.11 Manipulate Source

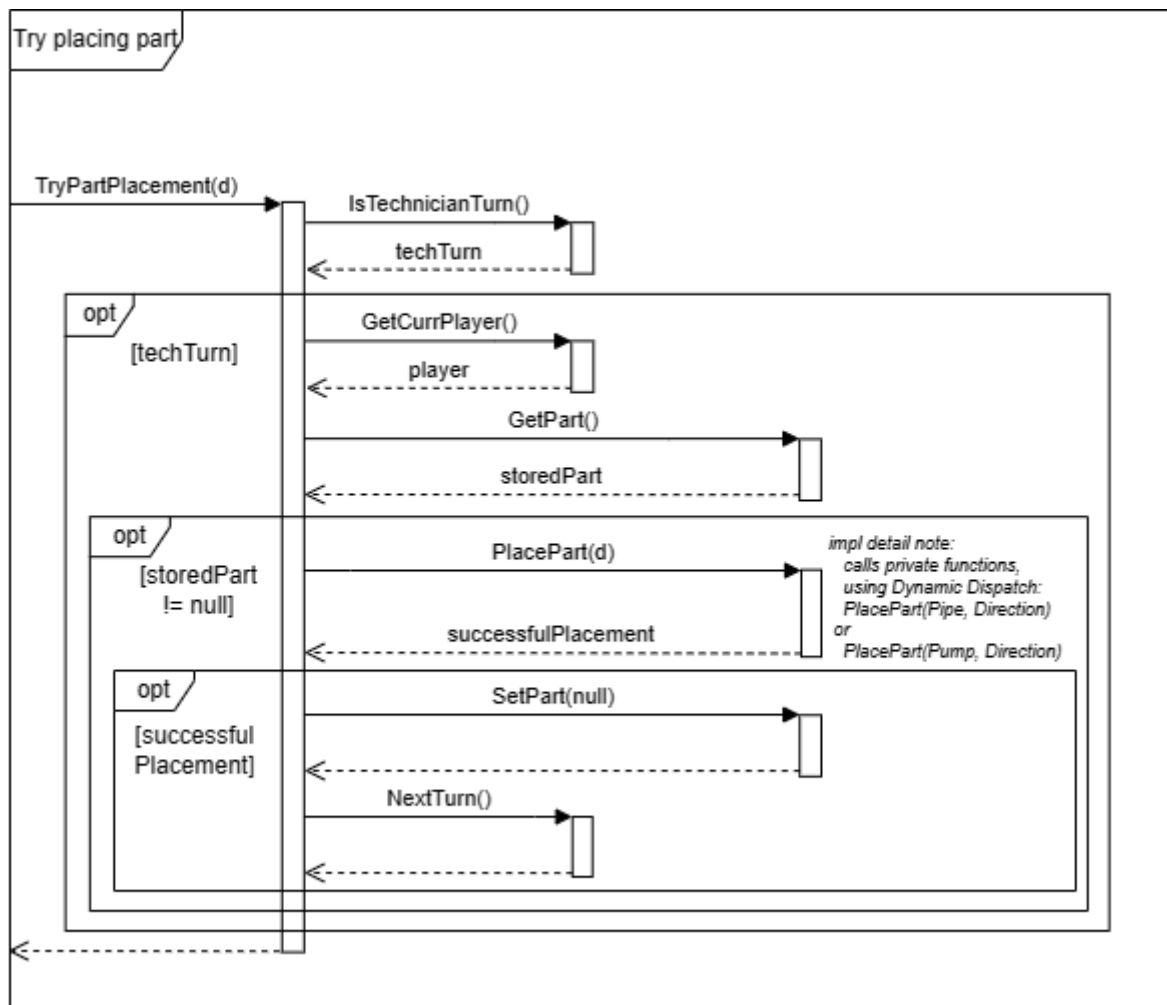


3.4.12 Pick Up and Store Pipe

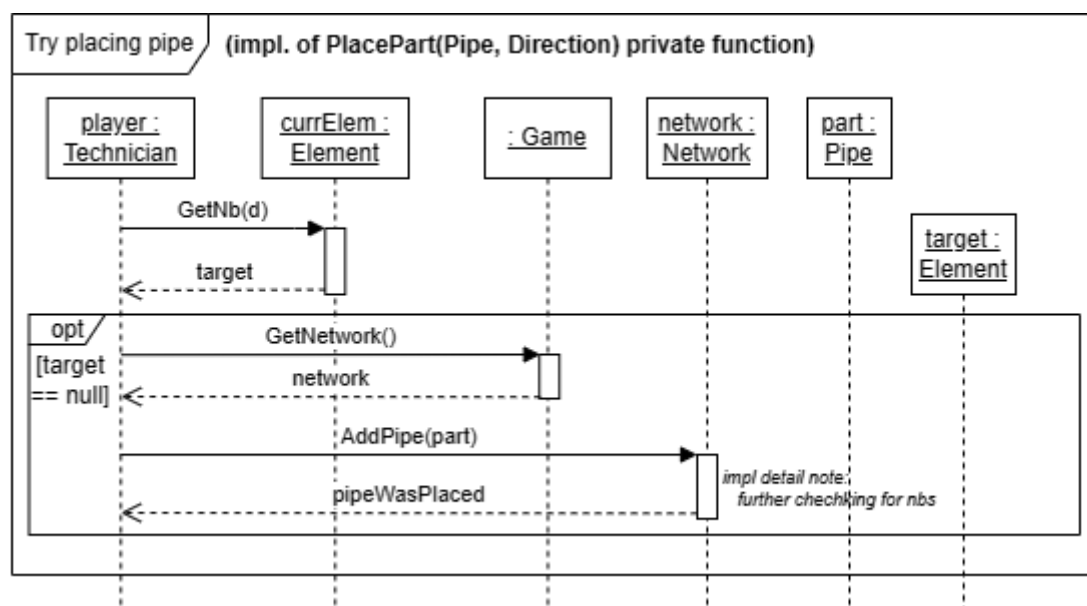


3.4.13 Pick Up and Store Pump

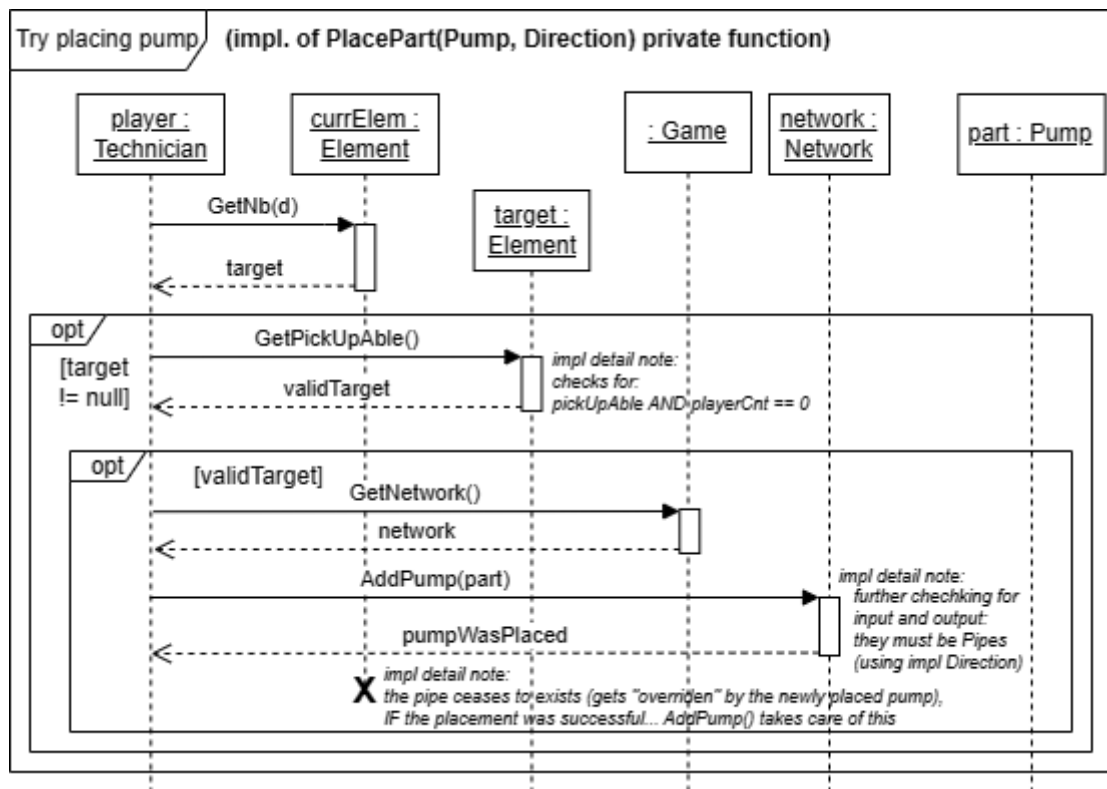
3.4.14 Try Placing Part



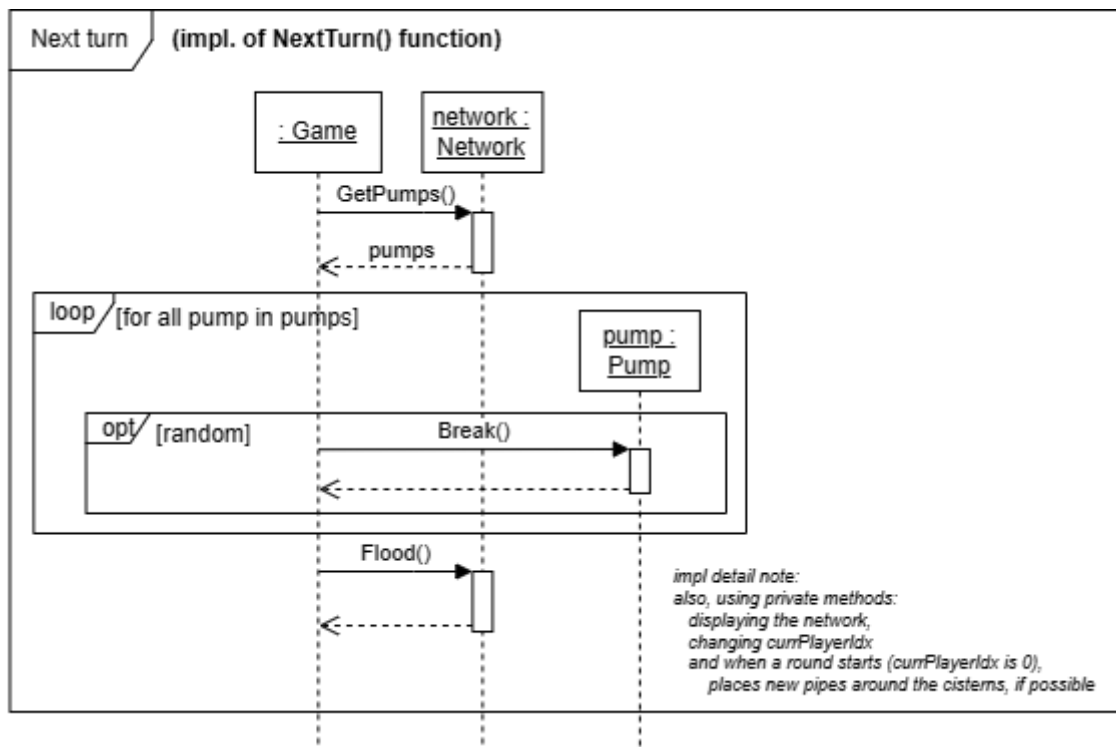
3.4.15 Try Placing Pipe



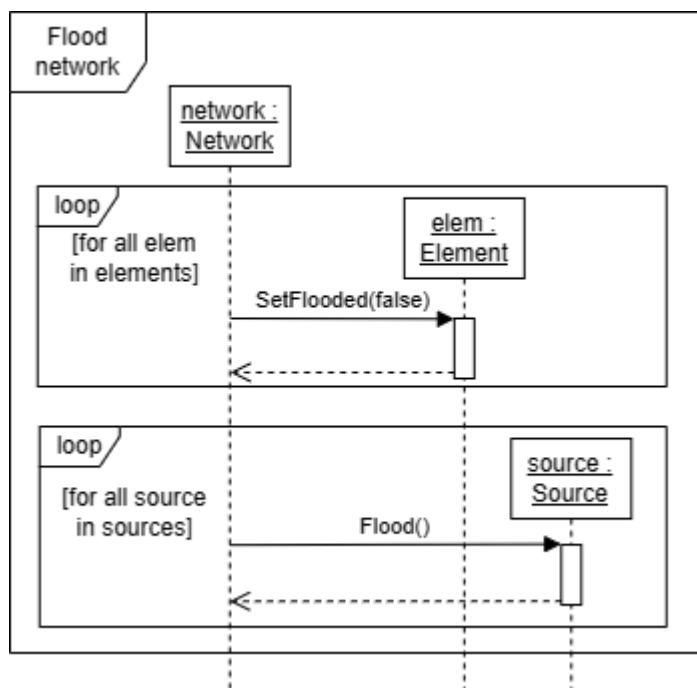
3.4.16 Try Placing Pump



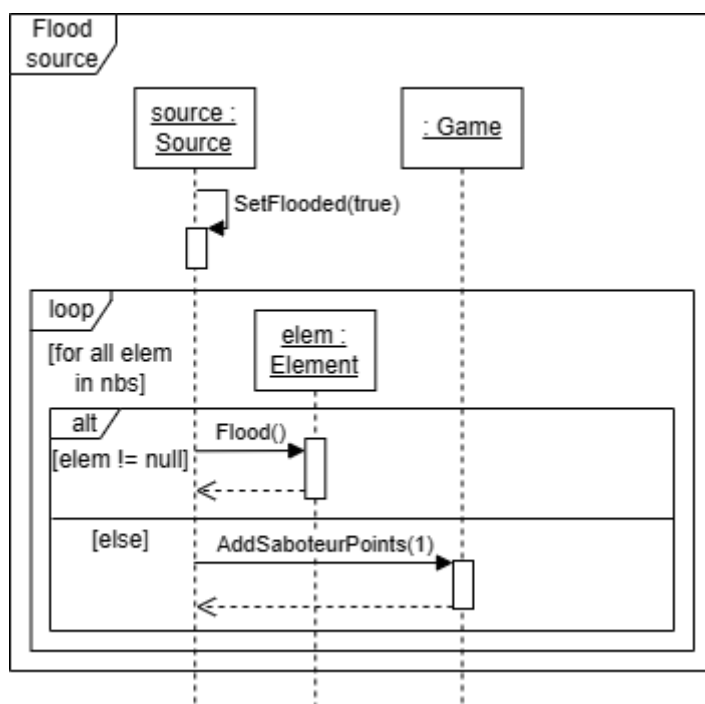
3.4.17 Next Turn



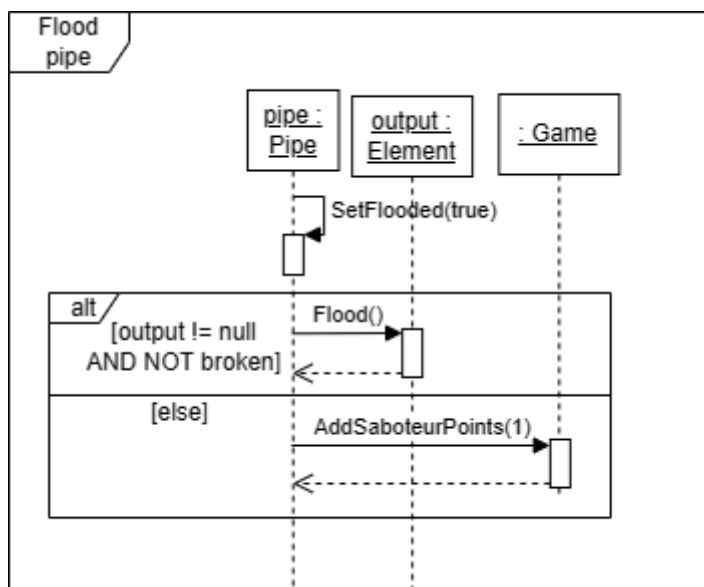
3.4.18 Flood Network



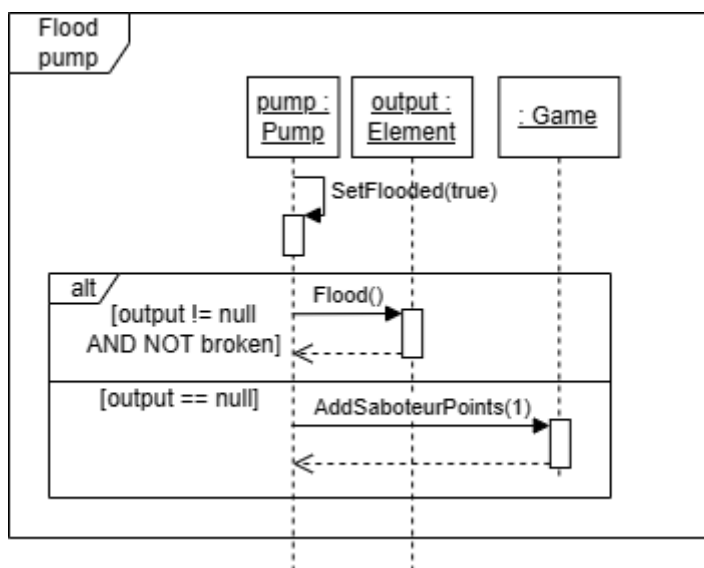
3.4.19 Flood Source

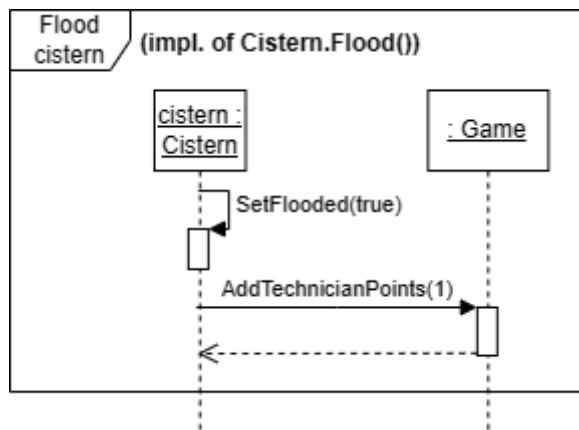


3.4.20 Flood Pipe



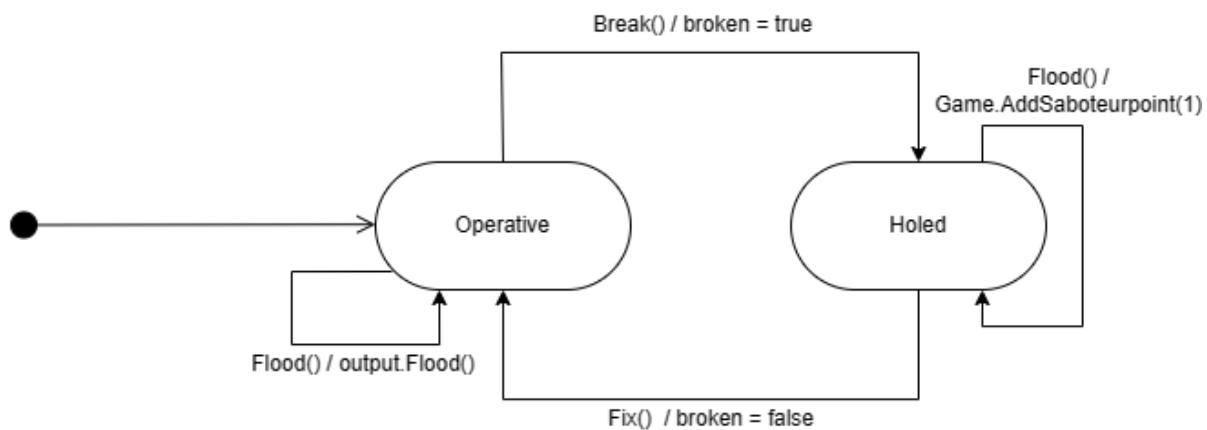
3.4.21 Flood Pump



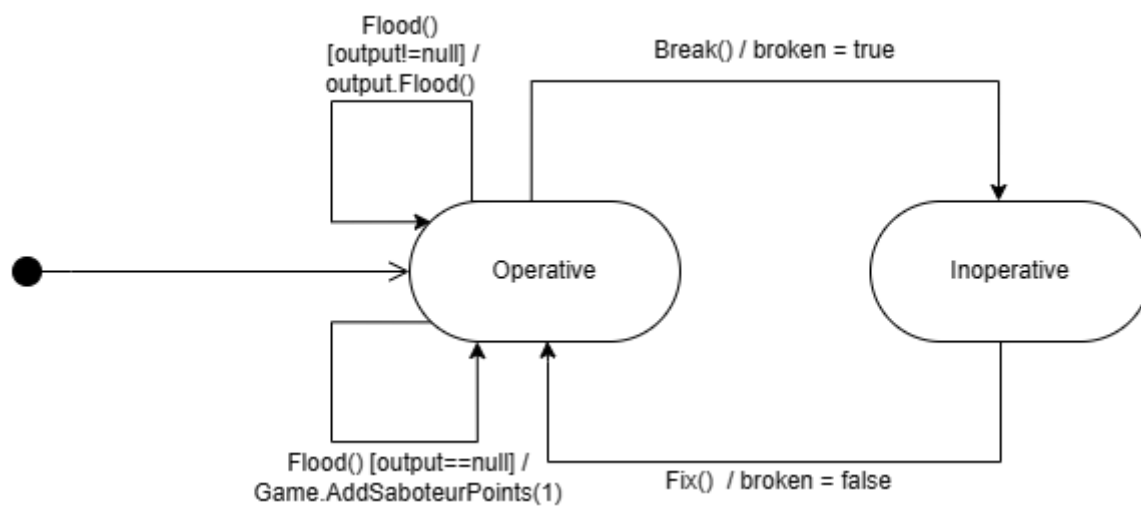
3.4.22 Flood Cistern

3.5 State-chartok

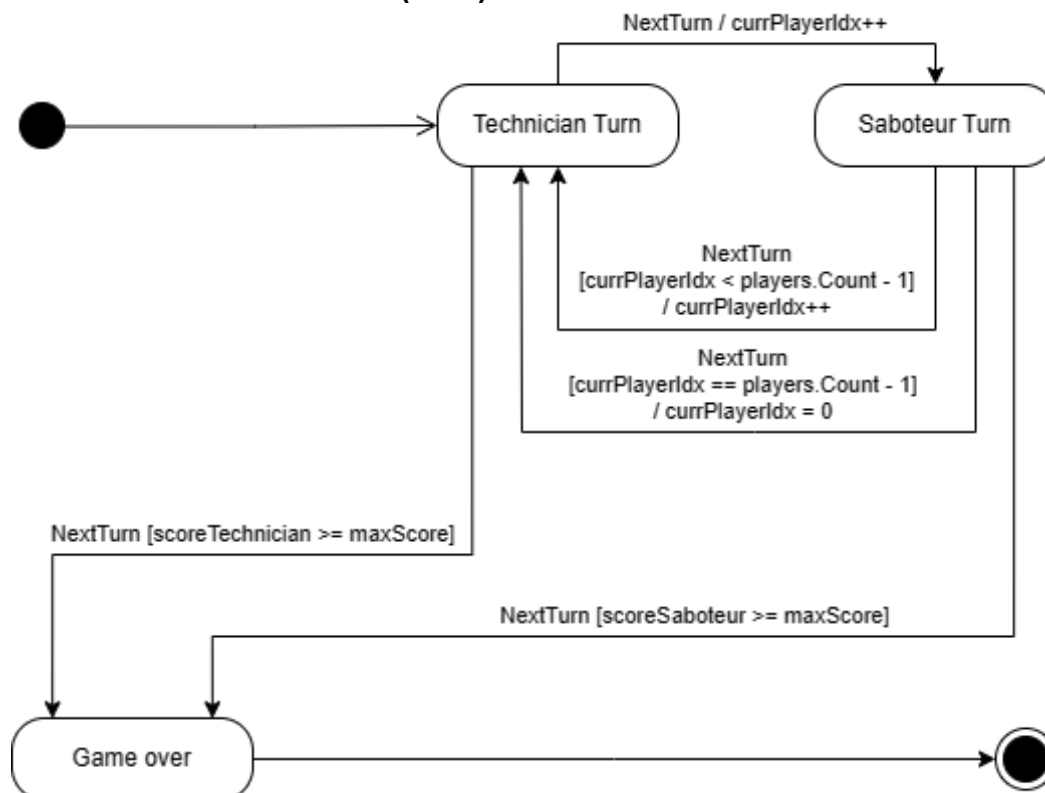
3.5.1 Pipe States



3.5.2 Pump States



3.5.3 Játékos köreinek (turn) váltakozása



3.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2023.03.22. 19:00	0,5 óra	Mizser Váradi Tepliczky Fekete Sasvári	<p>Értekezlet. Döntés:</p> <ul style="list-style-type: none"> • Mizser és Sasvári javítja az osztálydiagramot és a szekvenciadiagramokat • Fekete, Tepliczky és Sasvári elkészíti a 2 hiányzó szekvenciadiagramot • Váradi javítja a dokumentum formai és tartalmi hiányosságait <p>-----</p> <p>Diagram javítások Határidő: 2023.03.25. 22:00. Elkészült: 2023.03.25. 20:00</p> <p>Kész dokumentum (formázással) Határidő: 2023.03.27. 0:00 Elkészült: 2023.03.26 21:00</p>
2023.03.23. 18:00	3 óra	Mizser Sasvári	<p>Osztálydiagram és szekvenciadiagramok hibáinak feltárása:</p> <ul style="list-style-type: none"> • cső életvonalának megszakadása amikor felülíródik a lerakott pumpával • "Repair/Change pump direction by Technician"-ben a felesleges GetPumpDirections hívás törlése • GetCurrElem getter hozzáadása Playerhez, ennek jelölése szekvencián megjegyzéssel • MoveTo(targetElem) felvétele osztálydiagramra • GetPickUpAble függvény leírásának megváltoztatása
2023.03.23 20:00	2 óra	Mizser	<p>Az előzőek tényleges módosítása a diagramokon. További javítások:</p> <ul style="list-style-type: none"> • „osztálydiagram magyarázatok” rész megírása • nem egyértelmű szekvencia diagramokra az illusztrált függvény nevének kiírása • konzisztens függvénynevek
2023.03.25. 18:30	1,5 óra	Tepliczky, Fekete, Sasvári	<p>Pipe States és Pump States állapotdiagramok elkészítése. Korábbi javítások áttekintése.</p>
2023.03.26. 14:00	1 óra	Váradi	<p>A dokumentum formai és tartalmi javítása:</p> <ul style="list-style-type: none"> • A hiányzó oldalszámok pótlása • Objektumkatalógus pontosítása

2023.03.26. 17:00	1 óra	Váradi	A dokumentum tartalmi javítása: <ul style="list-style-type: none">• Pontosítás és korrekció az Objektumkatalógus szövegében és az osztályleírásokban.
2023.03.26. 19:00	2 óra	Váradi	A dokumentum tartalmi és formai javítása: <ul style="list-style-type: none">• További apróbb korrekciók az osztályok leírásában.• Szekvenciadiagramok frissítése.• Állapotdiagramok frissítése.• Dokumentum tördelése és formázása.