

3. Analízis modell kidolgozása

12 – Külföldiek és András

Konzulens:
Goldschmidt Balázs

Csapattagok

Kurcsi Norbert

Barabási Zalán Botond

Ganzer Attila

Nagy Szabolcs

Palásti András

Y3ZTEI

FWHHHB

Z44FF6

Z73X7L

IDNGIS

kurcsi.n@gmail.com

barabasizalan@gmail.com

ganzer.attila@gmail.com

n.szaby4@gmail.com

andraspalasti29@gmail.com

2023. 03. 10.

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Saboteur

Ez egy játékos által felvehető szerep. Ezzel a szereppel az a dolga a játékosnak, hogy különböző műveleteket végrehajtva (pumpa átállítása, csövek kilyukasztása) elérje, hogy minél több víz kifolyjon a homokba. A szabotőr a pumpákon és csöveken léphet, azzal a megkötéssel, hogy ha valaki áll már egy csövön, legyen az szabotőr vagy szerelő, akkor nem léphet arra a mezőre.

3.1.2 Mechanic

Ez egy játékos által felvehető szerep. Ezzel a szereppel az a dolga a játékosnak, hogy a játék során minél több vizet juttasson a forrásból a ciszternákba. Ezt a következő műveletekkel valósíthatja meg: pumpák átállítása, pumpák megjavítása, csövek megfoltozása, új pumpák elhelyezése egy-egy csővezeték közepén, illetve csövek végének az elmozdítása. A szerelő a pumpákon és csöveken léphet, azzal a megkötéssel, hogy egy csőre csak akkor léphet rá, ha más játékos nem áll éppen rajta.

3.1.3 Pipe

Az a pályaelem amelyen keresztül a játékosok a vizet szállíthatják a pumpák között, és ezáltal a forrásból a ciszternákba. A szabotőrök kilyukasztathatják, a szerelők pedig megjavíthatják, valamint áthelyezhetik. Egy cső mindig két játékelemet (ciszterna, forrás, pompa) köt össze. A csőben csak akkor folyhat víz, ha az egyik vége egy pumpához van csatlakoztatva. Egy csőben mindig csak egy irányba folyhat a víz, tehát van egy bemeneti illetve egy kimeneti pumpája. Emellett minden cső rendelkezik egy kapacitással, ami korlátozza annak a víznek a mennyiségét, amit átereszt. A ciszternáknál keletkező szabad végű csöveket a szerelők felvehetik, majd egy kiválasztott helyen letehetik, amiután letették, már nem vehető fel a cső, csak a cső egyik vége köthető át.

3.1.4 Pump

Az a pályaelem, amelyen keresztül a víz pumpálódik egyik csőből a másikba. A szerelők, valamint a szabotőrök egyaránt állíthatják, hogy honnan hová pumpáljon. Egyszerre több játékos is állhat rajta. A pompa rendelkezik egy tartállyal, amelybe akkor kerül víz ha a csövön érkező víz kapacitása meghaladja a kimeneti cső kapacitását. A pumpák véletlen időközönként elromolhatnak, amely esetben nem képesek vizet továbbítani.

3.1.5 Cistern

Az a játékelem, ahová a szerelők a vizet eljuttatva pontokat szerezhetnek. A ciszternát a játékosok nem képesek elmozdítani a játék során. A játék során folyamatosan termelődnek ezen a helyen a szabad csövek, valamint pumpák, amelyeket a szerelők felhasználhatnak a pálya építéséhez.

3.1.6 Spring

Az a játékelem, ahonnan a pályán lévő víz származik, amelyet a játékosok a szerepüknek megfelelően adott helyre próbálnak eljuttatni. A szabotőrök, és a szerelők sem képesek a pályaelem elmozdítására, annak bármilyen megváltoztatása nem lehetséges.

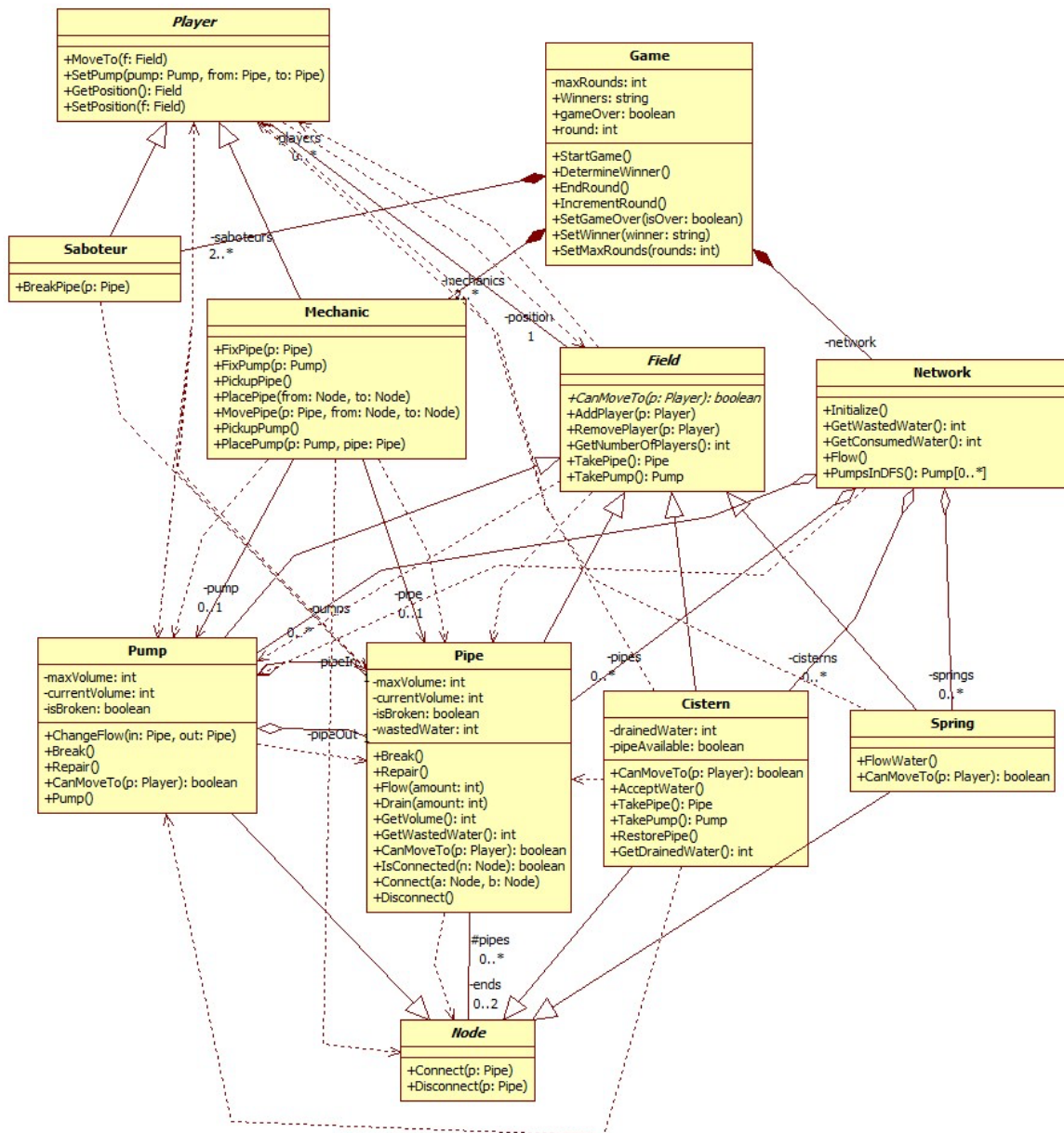
3.1.7 Game

A játékosok hozzák létre, ez felelős a játékmenet lebonyolításáért, annak megfelelő, elvárt működéséért.

3.1.8 Network

A játék folyamán létrehozott csőhálózat, amelyet bármely játékos megtekinthet, és a szerepétől függően interaktálhat a hálózat egyes elemeivel. Csövekből, pumpákból, forrásból, valamint ciszternákból áll.

3.2 Statikus struktúra diagram



3.3 Osztályok leírása

3.3.1 Cistern

- **Felelősség**

A ciszternákért felelős osztály. A szerelők célja, hogy minél több vizet juttassanak a forrásoktól a ciszternákig .

- **Ősosztályok**

Field, Node.

- **Interfészek**

- **Attribútumok**

- **drainedWater:** A vízmennyiség ami befolyt a ciszternákba. Ennek függvényében számolódnak a szerelő csapat pontjai.

- **pipeAvailable:** Van-e elérhető cső, amelyet a szerelők felvehetnek az eszköztárukba.

- **Metódusok**

- **boolean CanMoveTo(Player p):** Visszatéríti, hogy a játékos ráléphet-e a pályaelemre.

- **void AcceptWater():** Befogadja a vizet.

- **Pipe TakePipe():** Ha még ebben a körben nem vettek el csövet róla akkor visszatér egy pumpával különben pedig null-al.

- **Pump TakePump():** Mivel bármikor felvehető pumpa a ciszternából ezért mindig egy pumpával tér vissza.

- **void RestorePipe():** Visszaállítja a ciszterna működését hogy újabb csövet lehessen elvenni róla.

- **int GetDrainedWater():** A cisztárnába belefolyt víz mennyiségét téríti vissza.

3.3.2 Field

- **Felelősség**

A pályaelemekért felelős absztrakt osztály.

- **Ősosztályok**

- **Interfészek**

- **Attribútumok**

- **players:** A pályaelemen álló játékosok listája.

- **Metódusok**

- **void CanMoveTo(Player p):** Visszatéríti, hogy a játékos ráléphet-e a pályaelemre.

- **void AddPlayer(Player p):** Ha egy játékos rálép a pályaelemre, ezt hozzáadja a játékosok listájához.

- **void RemovePlayer():** Ha egy játékos ellép a pályaelemről, ezt kitörli a játékosok listájából.
- **int GetNumberOfPlayers():** Visszatéríti a pályaelemen álló játékosok számát. A cső esetében, egyszerre több mint egy játékos nem állhat rajta.
- **Pipe TakePipe():** Megpróbál elvenni egy csövet az adott mezőről, ha nem megy akkor null értékkel tér vissza.
- **Pump TakePump():** Megpróbál elvenni egy pumpát az adott mezőről, ha nem megy akkor null értékkel tér vissza.

3.3.3 Game

- **Felelősség**

Az osztály felelős a játémezők és az azon lévő objektumok létrehozásáért és az újabb, játékokban résztvevő szereplők csatlakoztatásáért. A játék továbbá meg tud szünni, győzelem vagy vereség esetén.

- **Összostályok**

- **Interfészek**

- **Attribútumok**

- **maxRounds:** A játéknak köreinek a maximális száma.
- **Winners:** A játék győztes csapata.
- **gameOver:** A játék végét jelzi.
- **round:** Az aktuális kör sorszáma, amelynek tájékoztató jellege is van, hogy a játéknak milyen pontjában állunk.
- **network:** A csőhálózat, amelyből ismerjük az elfolyt, valamint a ciszternákba jutott vízmennyiséget.
- **mechanics:** Szerelőkölő álló tömb, amely a szerelők csapatát képviseli.
- **saboteurs:** Szabotőrölő álló tömb, amely a szabotőrök csapatát képviseli.

- **Metódusok**

- **void StartGame():** Elindítja a játékot.
- **void DetermineWinner():** Meghatározza az elfolyt vízmennyiség és a ciszternákba jutott vízmennyiség alapján a győztes csapatot.
- **void EndRound():** Befejeződik egy kör.
- **void IncrementRound():** Minden kör után meghívódik, és növeli az aktuális kör sorszámát. Amint az aktuális körszám eléri a maximális körök számát, a kör lejátszása után a játék befejeződik.
- **SetGameOver(isOver: boolean):** A játék végét állítja be az osztály gameOver boolean típusú változójában.
- **SetWinner(winner: string):** A játék győztesét állítja be, ill. ha döntetlen, akkor "Tie" értéket ad a Winners attribútumnak.
- **SetMaxRounds(rounds: int):** A játék köreinek maximális számát állítja be a maxRounds attribútumban.

3.3.4 Mechanic

- **Felelősség**

A lyukas csövek és az elromlott pumpák megjavításáért felelős, valamint a csőrendszert úgy kell alakítsa, hogy minél több víz jusson a forrástól a ciszternákba.

- **Ősosztályok**

Player.

- **Interfészek**

- **Attribútumok**

- **pump**: Az eszköztárban a játékosnak lehet egy vagy nulla pumpája, amelyet majd lehelyezhet.
- **pipe**: Az eszköztárban a játékosnak lehet egy vagy nulla csöve, amelyet majd lehelyezhet.

- **Metódusok**

- **void FixPipe(Pipe p)**: A szerelő megjavít egy kilyukasztott csövet. Ezután a csövön keresztül, a víz átjut egyik pumpától a másikig, anélkül, hogy elfolyjon a homokba.
- **void FixPump(Pump p)**: A szerelő megjavít egy meghibásodott pumpát. A pumpa ezután képes lesz a csöveken tovább pumpálni a vizet más csövek felé.
- **void PickupPipe(Pipe p)**: A szerelő felvesz egy csövet a ciszternáktól. Ezután az eszköztárba elérhető lesz egy cső, amelyet ezt lehelyezhet egy pumpánál.
- **void PlacePipe(Node from, Node to)**: A szerelő letesz egy csövet, amelyet a ciszternáktól vett fel. Egyik végét arra a pumpára köti amelyiken áll, a másik végét viszont egy olyan pumpára, amelyikre még lehetséges a csatlakoztatás. Ezután az eszköztára kiürül, és ha a ciszternáknál jár az elkövetkezendő körökben, akkor újabb csövet vehet fel.
- **void MovePipe(Pipe p, Node to)**: A szerelő a kiválasztott csövet áthelyezi, úgy, hogy azt a csövet mozdítja, amelyik arra a pumpára van kötve, amelyiken a szerelő áll, viszont a csőnek csak azt a végét mozdíthatja, amelyik távolabb van tőle.
- **void PickupPump(Pump p)**: A szerelő felvesz egy pumpát. Ez a pumpa a szerelő eszköztárába kerül, majd ezt a pályán bárhol lehelyezheti, ahol ez megengedett.
- **void PlacePump(Pump p)**: A szerelő letesz egy pumpát, amely az eszköztárban volt eddig. Ezután az eszköztára kiürül és felvehet egy más pumpát az elkövetkezendő körökben.

3.3.5 Network

- **Felelősség**

A csőhálózat karbantartásáért felelős osztály.

- **Ősosztályok**

Nincsenek.

- **Interfészek**
- **Attribútumok**
 - **pumps:** A pumpák összessége, amelyek a csőhálózathoz tartoznak. Ezekbe nem tartoznak bele a szerelők által felvett, de még le nem helyezett pumpák.
 - **pipes:** A csövek összessége, amelyek a csőhálózathoz tartoznak. Ezekbe nem tartoznak bele a szerelők által felvett, de még le nem helyezett csöveket.
 - **cisterns:** A ciszternák összessége, amelyek a csőhálózathoz tartoznak. Ide jön a csöveken és pumpákon keresztül a forrásból a víz.
 - **springs:** A források összessége, ahonnan a vizet továbbítjuk a ciszternák felé.
- **Metódusok**
 - **void Initialize():** Létrehozza a kezdő pályát, az alap csőrendszert.
 - **int GetWastedWater():** Visszatéríti az elfolyt vízmennyiséget. Ez alapján számolódik a szabotőr csapat pontszáma.
 - **int GetConsumedWater():** Visszatéríti a ciszternákba folyt vízmennyiséget. Ez alapján számolódik a szerelő csapat pontszáma.
 - **void Flow():** Elindítja a víz folyásának folyamatát.
 - **Pump[] PumpsInDFS():** A forrásoktól kezdődően egy DFS-beli sorrendben adja vissza a hálózat pumpáit.

3.3.6 Node

- **Felelősség**

Egy absztrakt osztály ami egy csomópontot jelöl mindenhez amihez csöveket lehet csatlakoztatni.

- **Ósosztályok**

Nincsenek.

- **Interfészek**

- **Attribútumok**

- **pipes:** a csatlakoztatható csövek halmaza

- **Metódusok**

- **void Connect(p: Pipe):** A csomóponthoz csatlakoztatja az adott csövet.
- **void Disconnect(p: Pipe):** Lecsatlakoztatja a csomóponttól az adott csövet.

3.3.7 Pipe

- **Felelősség**

A víz szállításáért felelős, vagy a többi játékelemből szállít el vizet vagy azokba szállítja. valamint ezen képesek a játékosok mozogni.

- **Ósosztályok**

Field, Node.

- **Interfészek**
- **Attribútumok**
 - **maxVolume**: a cső kapacitása hogy maximum mennyi víz fér bele
 - **currentVolume**: a csövön érkező aktuális vízmennyiség
 - **isBroken**: a cső állapotát jelzi, midőn ki van lyukasztva vagy sem
 - **ends** :a cső végeihez csatlakoztatott egy-egy másik játékelem
- **Metódusok**
 - **void Brake()**: A cső kilyukadt állapotba kerül. Ezentúl a csövön érkező víz mind elfolyik, ezzel növelve a szabotőrök pontszámát.
 - **void Repair()**: A cső megjavítása, amiután képes lesz vizet szállítani a pumpák között, valamint a ciszternák irányába.
 - **void Flow(int amount)**: Egy pumpa vagy forrás hívja meg a csövön amikor betölti a vizet az adott csőbe.
 - **void Drain(int amount)**: Egy pumpa vagy ciszterna hívja meg a csövön amikor kiszívja annak tartalmát.
 - **boolean CanMoveTo(Player p)**: Visszatéríti, hogy a játékos ráléphet-e a pályaelemre.
 - **int GetVolume()**: Visszatéríti a cső kapacitását.
 - **boolean IsConnected(Node n)**: Visszatéríti. hogy a csőnek a paraméterként kapott vége hozzá van csatlakoztatva egy pumpához vagy egy ciszternához, vagy sem.
 - **void Connect(Node a, Node b)**: A csövet rácsatlakoztatja két pályaelemre(forrás, pumpa vagy ciszterna).
 - **void Disconnect()**: Lecsatlakoztatja a csövet.
 - **int GetWastedWater()**: A csövön kifolyt víz mennyiségét téríti vissza.

3.3.8 Player

- **Felelősség**
A játékosokért felelős és azokat megvalósító absztrakt osztály, amiből két további osztály származik: Mechanic, Saboteur.
- **Ősosztályok**
Nincsenek.
- **Interfészek**
- **Attribútumok**
 - **position**: a játékos pozíciójának tárolására használt attribútum.
- **Metódusok**
 - **void MoveTo(Field f)**: A játékos pozíciót vált.
 - **void SetPump(Pump pump, Pipe from, Pipe to)**: A játékos a paraméterként kapott pumpát átállítja, hogy melyik csőből melyikbe pumpálja a vizet.
 - **void SetPosition(Field f)**: Beállítja a játékos aktuális pozícióját.

3.3.9 Pump

- **Felelősség**

A víz pumpálásáért felelős osztály.

- **Ősosztályok**

Field, Node.

- **Interfészek**

- **Attribútumok**

- **maxVolume:** A pumpa tartályába tölthető maximális vízmennyiség.
- **currentVolume:** A pumpa tartályában lévő vízmennyiség.
- **isBroken:** A pumpa állapotát jelzi, ha működőképes vagy meghibásodott
- **pipeIn:** Az a cső, amelyről érkezik a víz.
- **pipeOut:** Az a cső, amelyiken keresztül továbbítódik a víz.

- **Metódusok**

- **void ChangeFlow(Pipe in, Pipe out):** Megváltoztatja, hogy melyik csőből melyikbe pumpálja a vizet.
- **void Break():** A pumpa meghibásodik, ezután ez használhatatlanná válik, nem képes vizet pumpálni, addig amíg egy szerelő meg nem javítja.
- **void Repair():** A pumpa megjavítása, ezután már használható - újra pumpálhat vizet.
- **boolean CanMoveTo(Player p):** Visszatéríti, hogy a játékos ráléphet-e a pályaelemre.
- **void Pump():** Egyik csőből a másikba pumpálja a vizet.

3.3.10 Saboteur

- **Felelősség**

A szabotőrök a csövek kilyukasztásáért, valamint a csövek átállításáért felelősek, ezzel minél több vizet a homokba folytatnak, hogy ők legyenek a győztesek.

- **Ősosztályok**

Player.

- **Interfészek**

- **Attribútumok**

- **Metódusok**

- **void BreakPipe(Pipe p):** A szabotőr egy paraméterként kapott csövet kilyukaszt. Ezután a kilyukasztott csőből a víz a homokba folyik, ezzel növelve a csapat pontszámát.

3.3.11 Spring

- **Felelősség**

A rendszert, pontosabban a belőle kivezető csöveket látja el vízzel. Ezekbe minden kör végén folytatja a megfelelő mennyiségű vizet.

- **Ősosztályok**

Field, Node.

- **Interfészek**

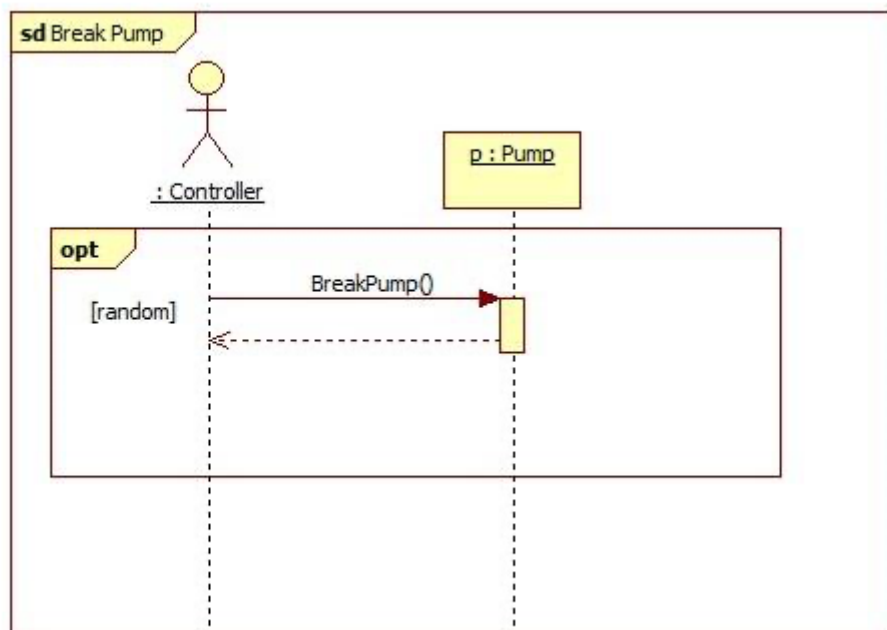
- **Attribútumok**

- **Metódusok**

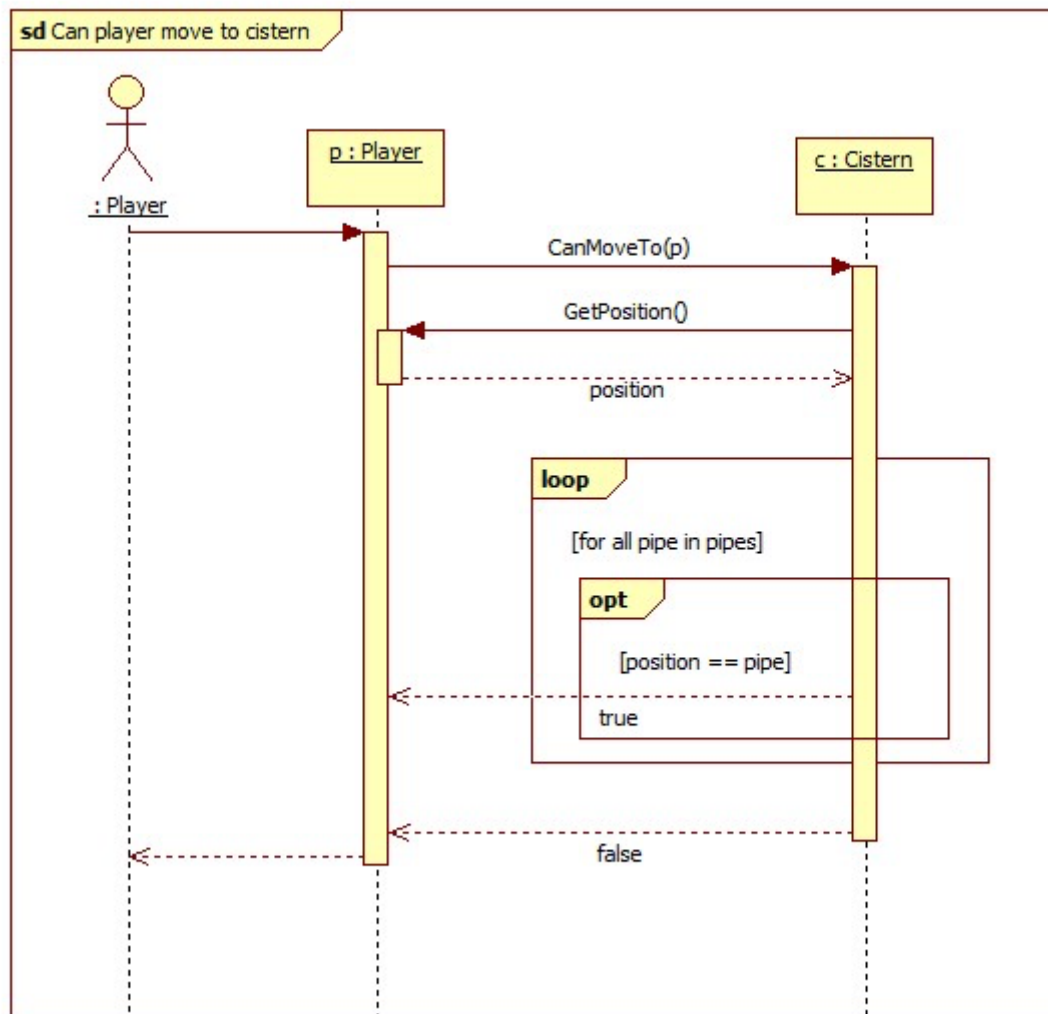
- **void FlowWater():** A víz áramlik a forrásokból.
- **boolean CanMoveTo(Player p):** Visszatéríti, hogy a játékos ráléphet-e a pályaelemre.

3.4 Szekvencia diagramok

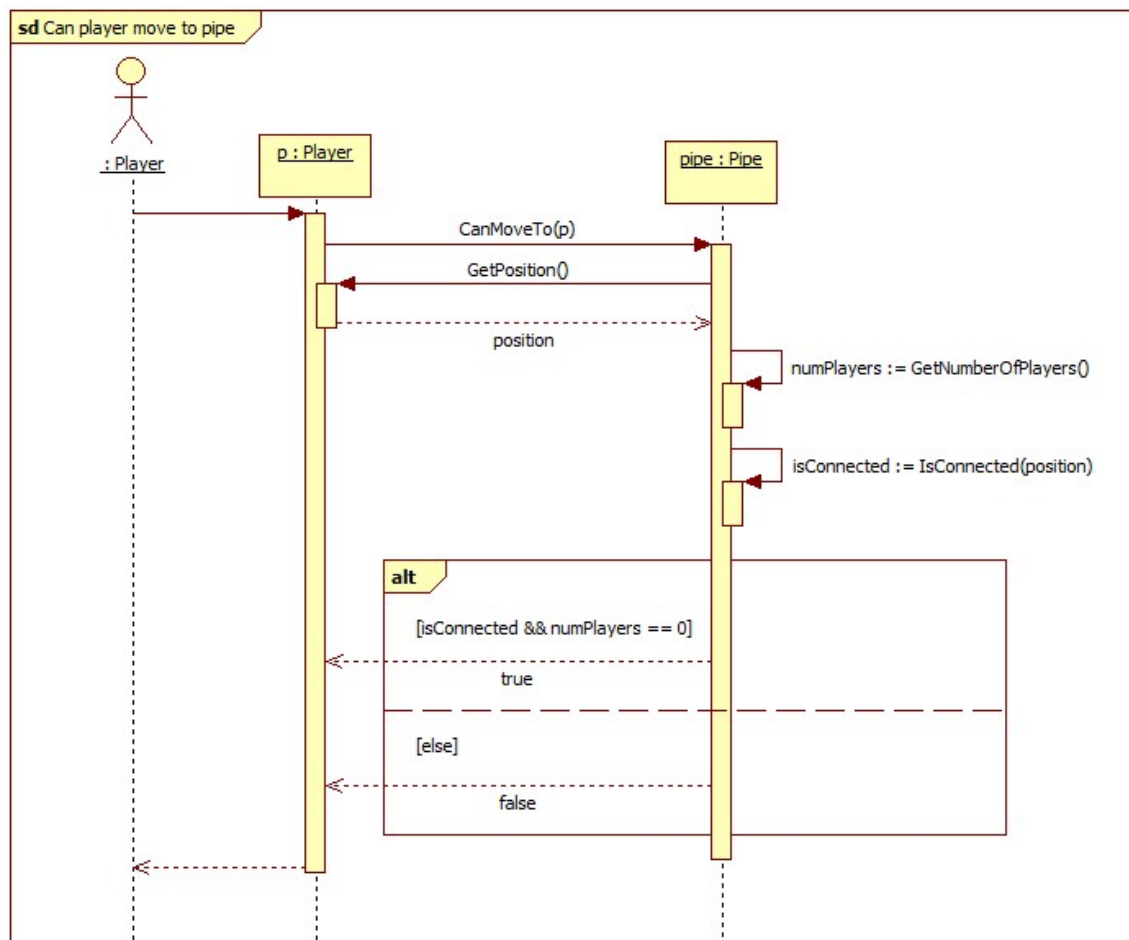
3.4.1 Break Pump



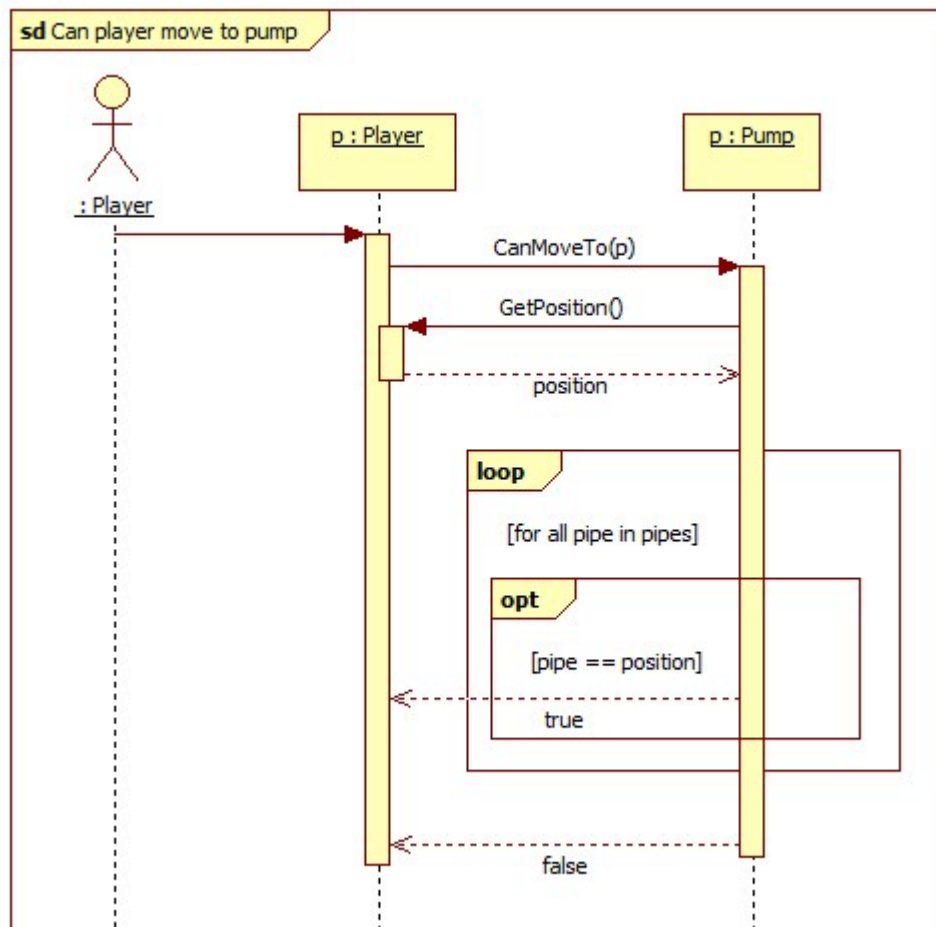
3.4.2 Can Player Move To Cistern



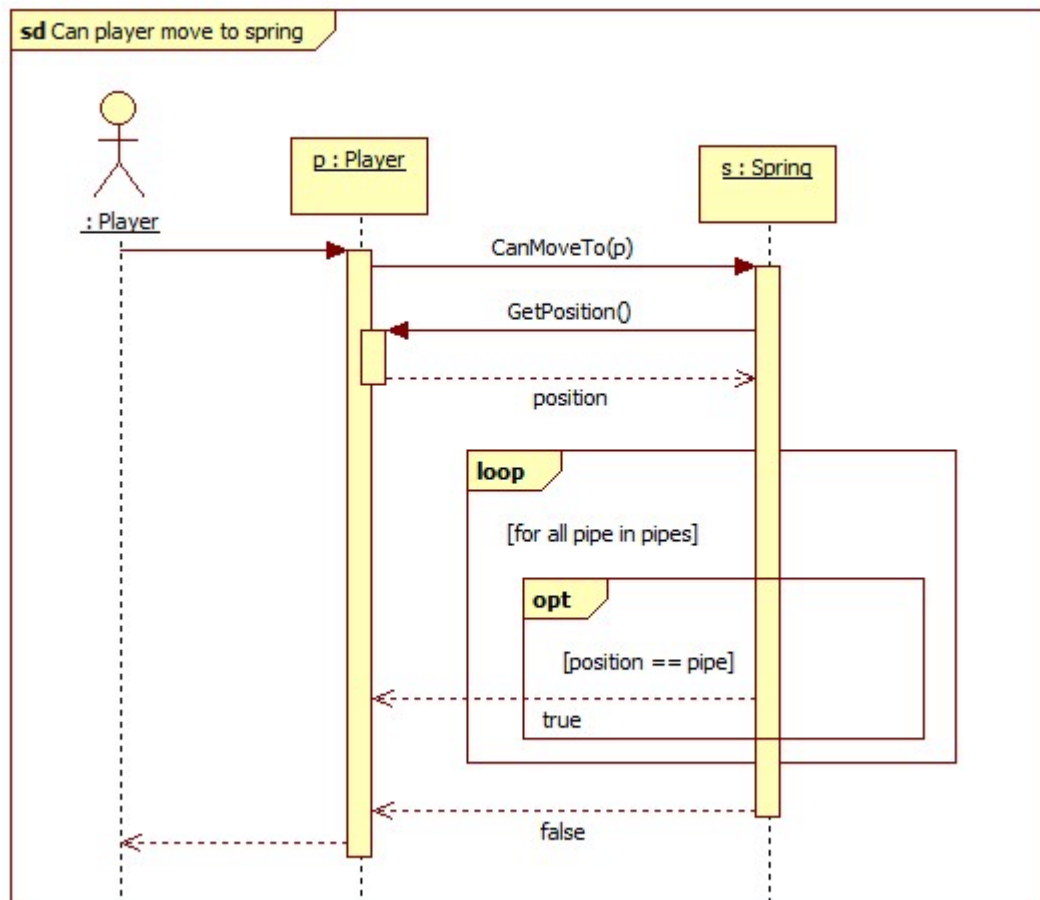
3.4.3 Can Player Move To Pipe

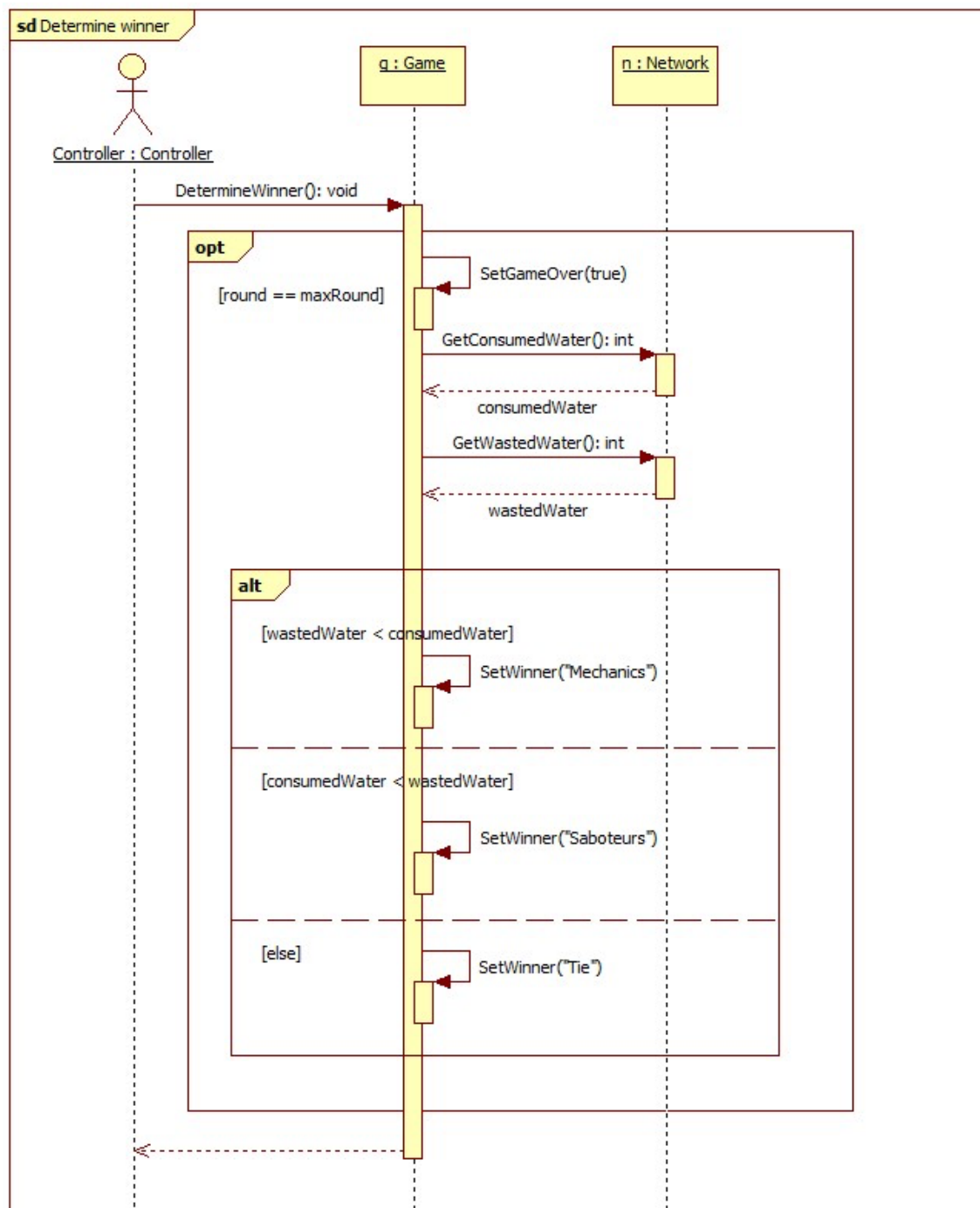


3.4.4 Can Player Move To Pump

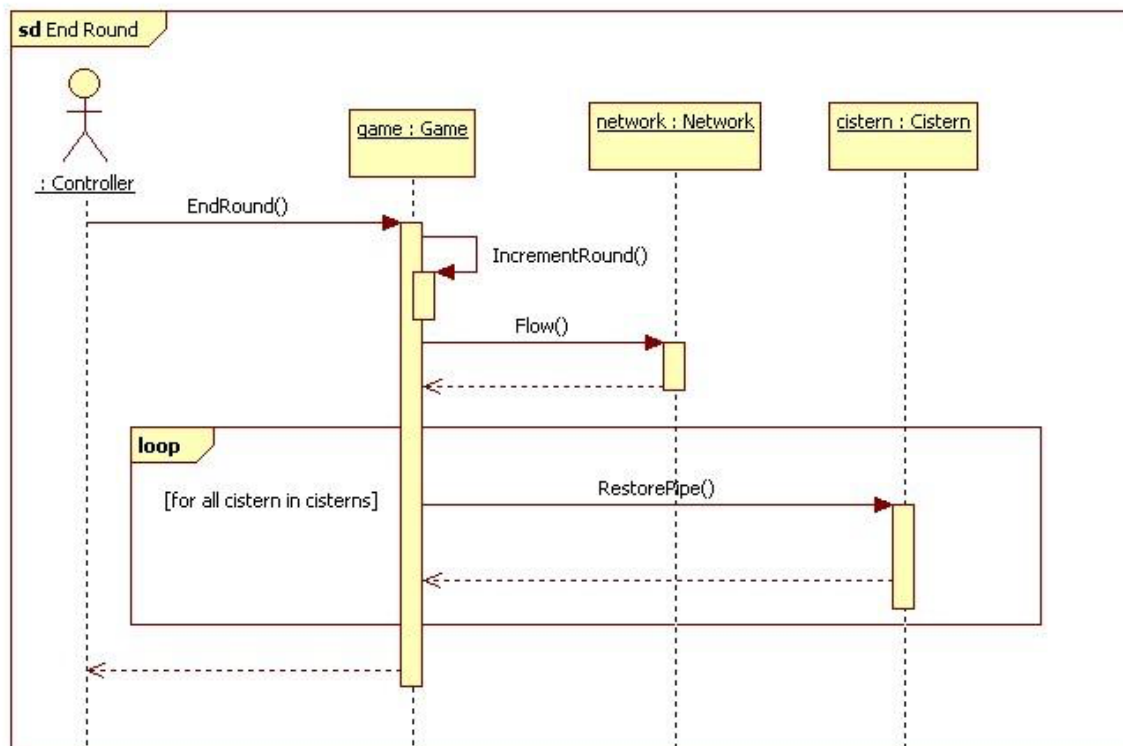


3.4.5 Can Player Move To Spring

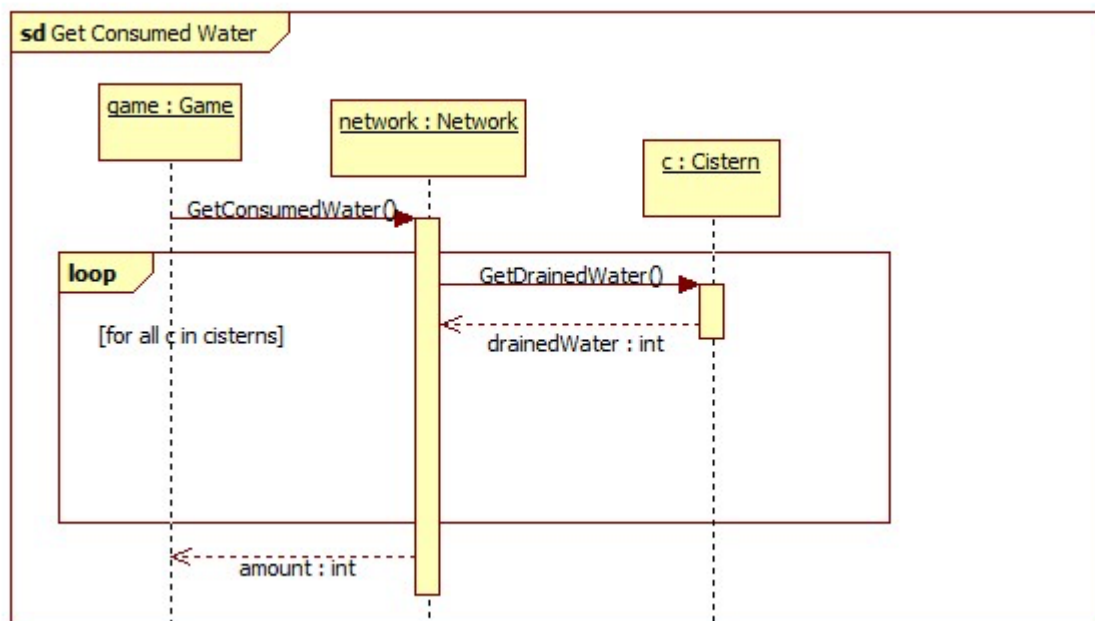


3.4.6 Determine Winner

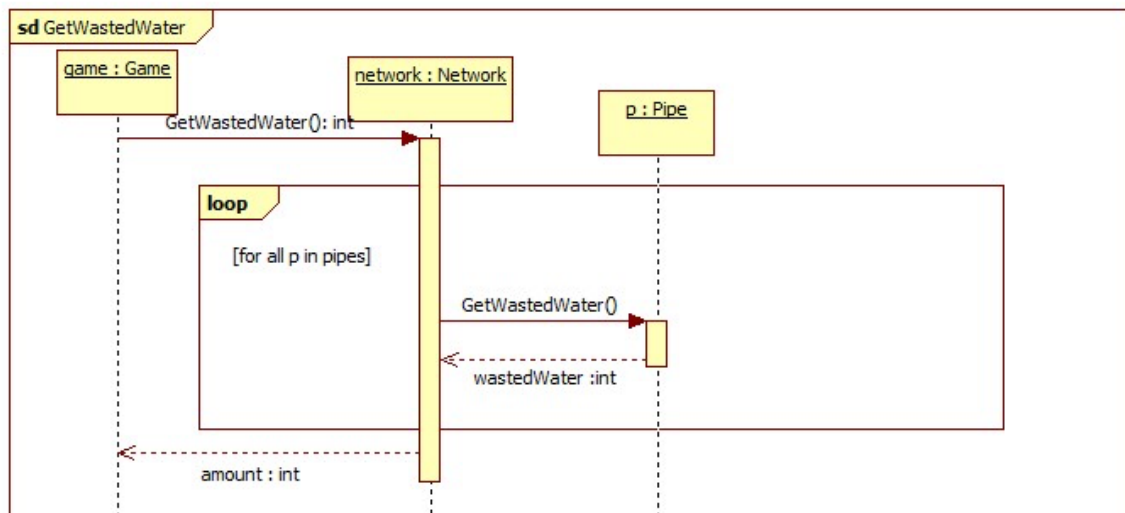
3.4.7 End Round



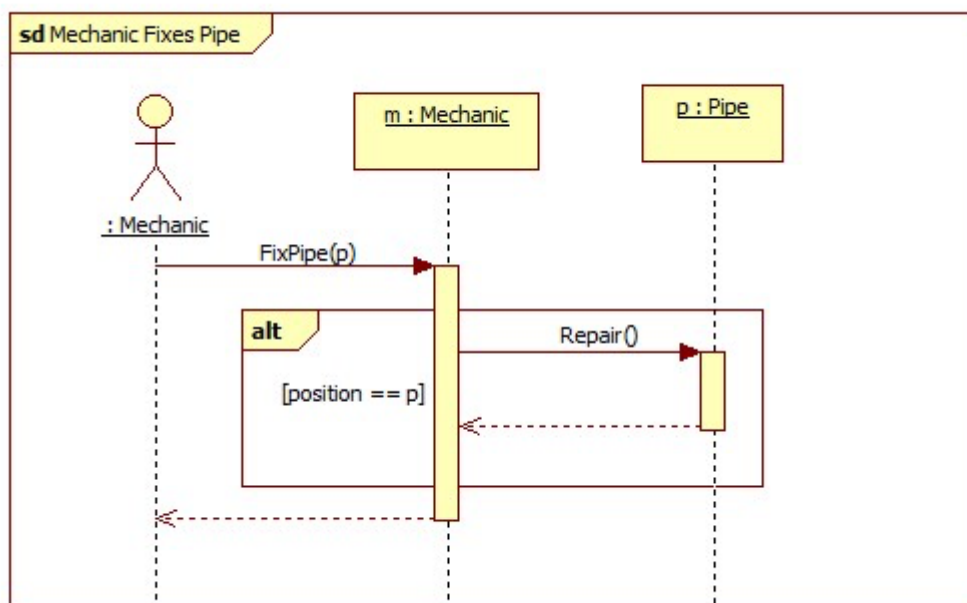
3.4.8 3.4.8 Get Consumed Water



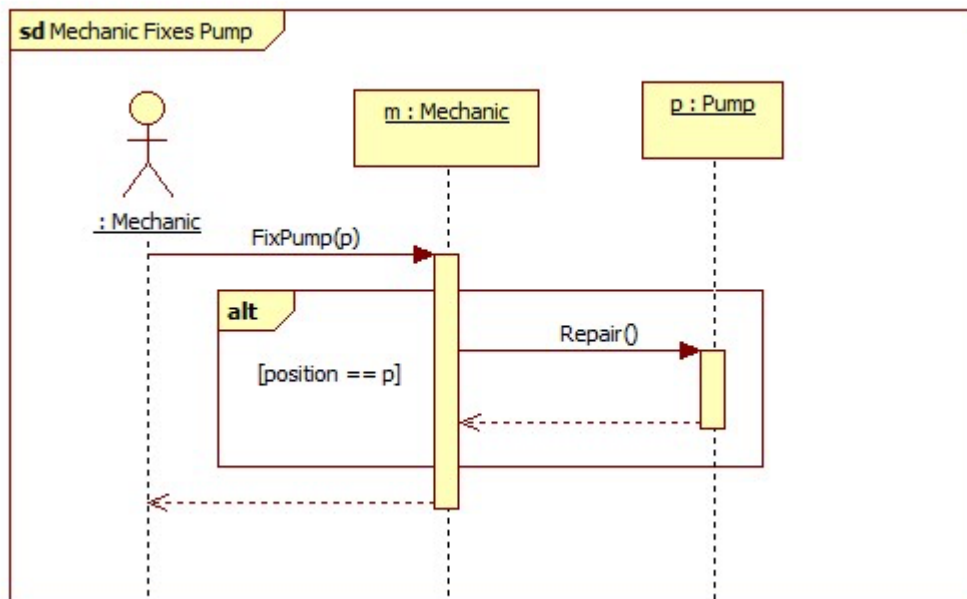
3.4.9 Get Wasted Water



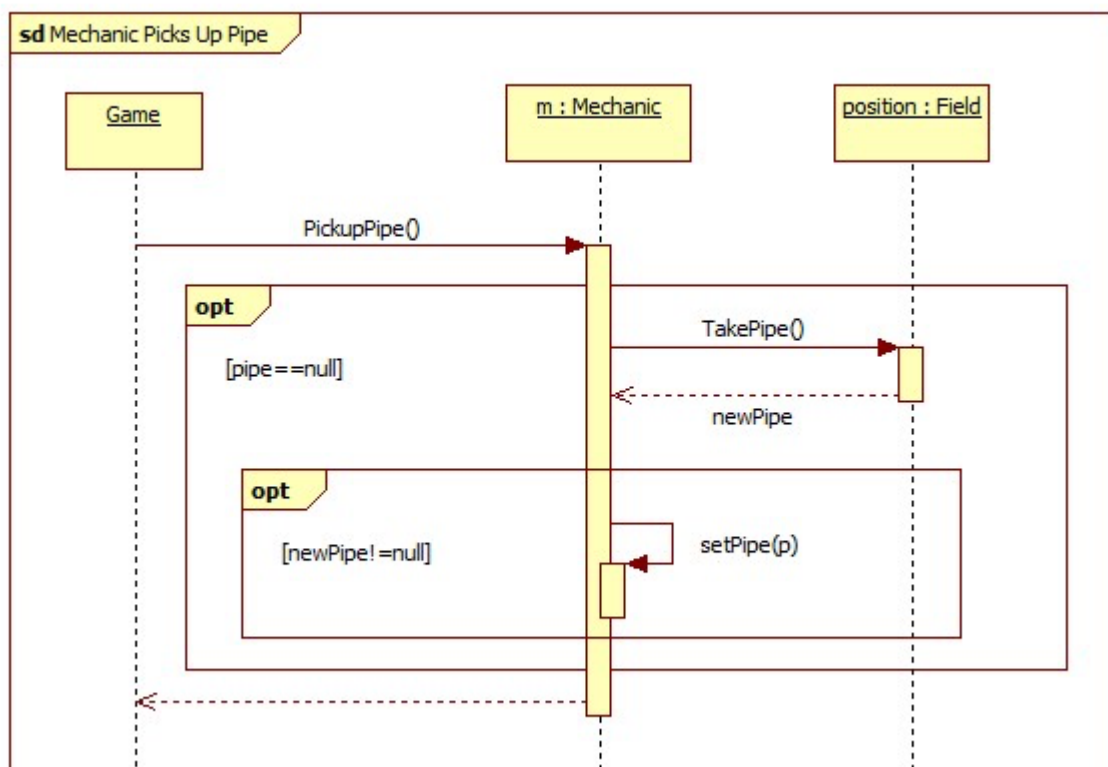
3.4.10 Mechanic Fixes Pipe



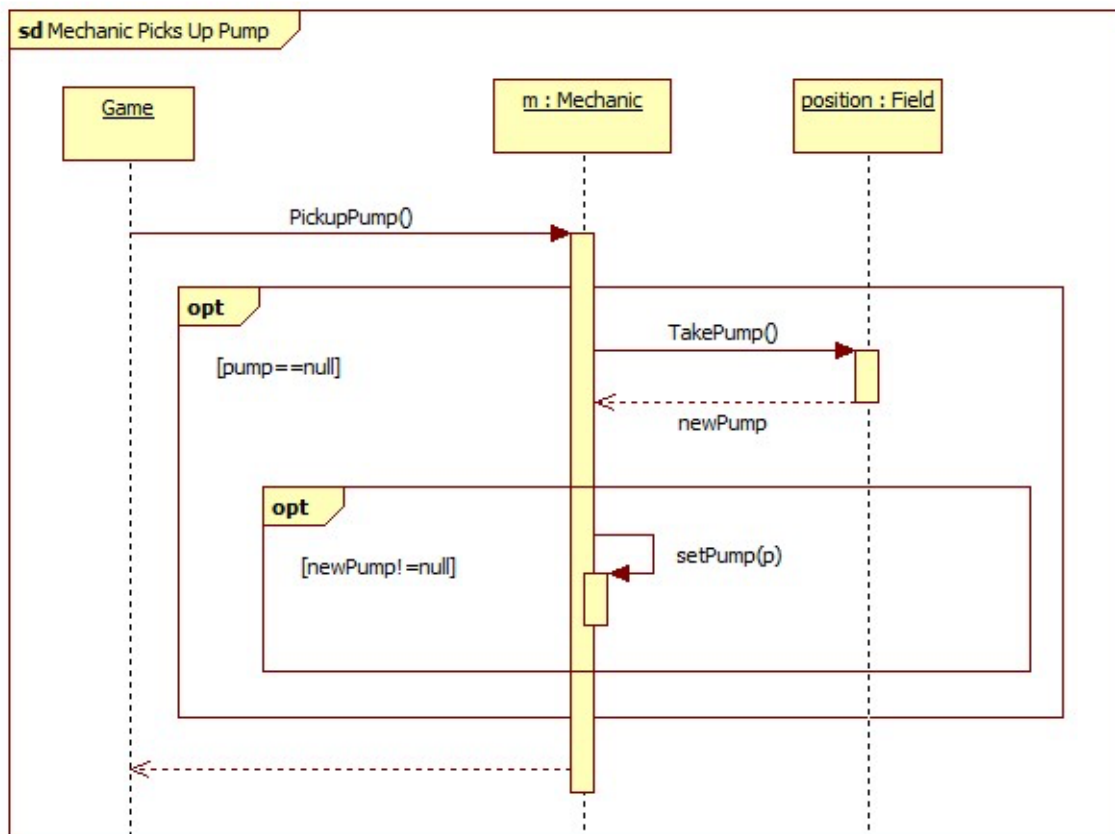
3.4.11 Mechanic Fixes Pump



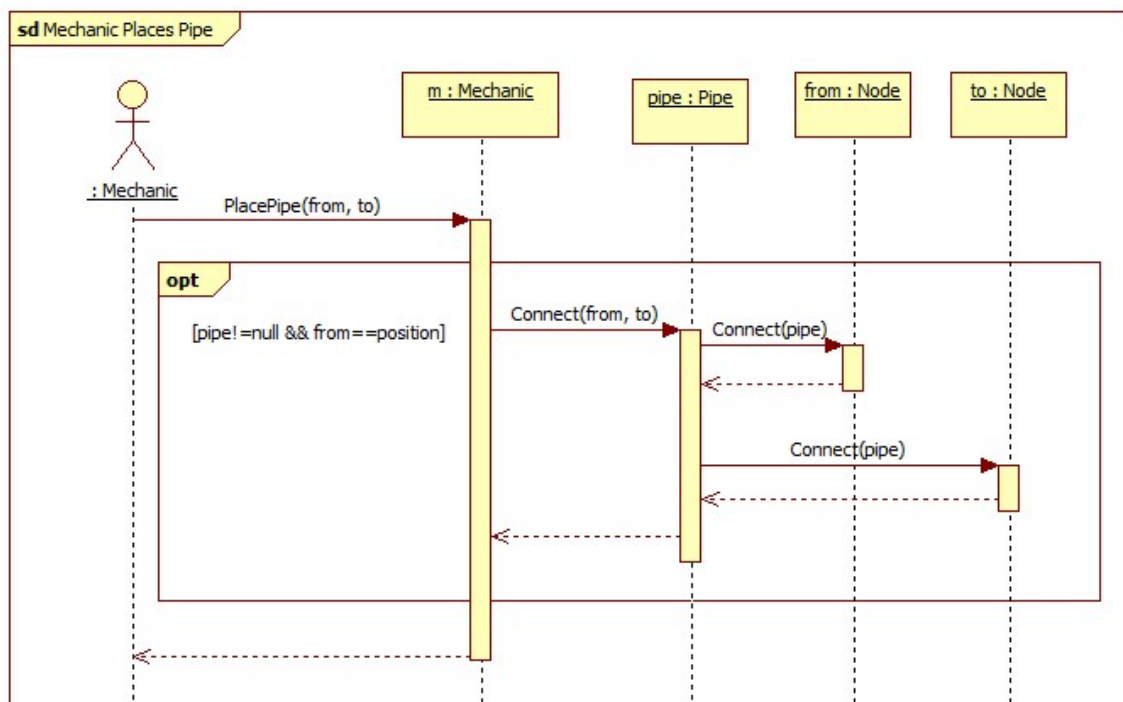
3.4.12 Mechanic Picks Up Pipe

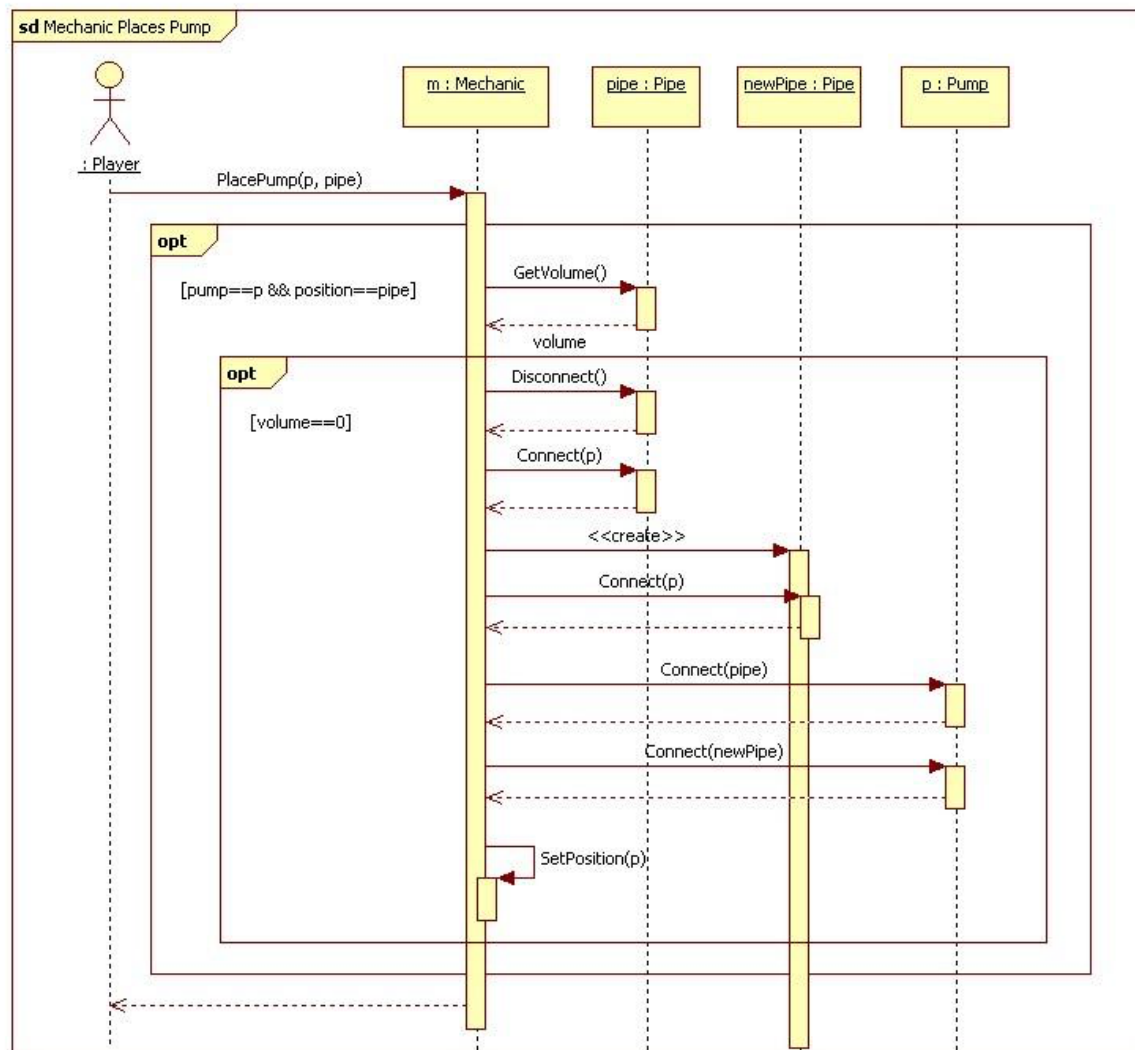


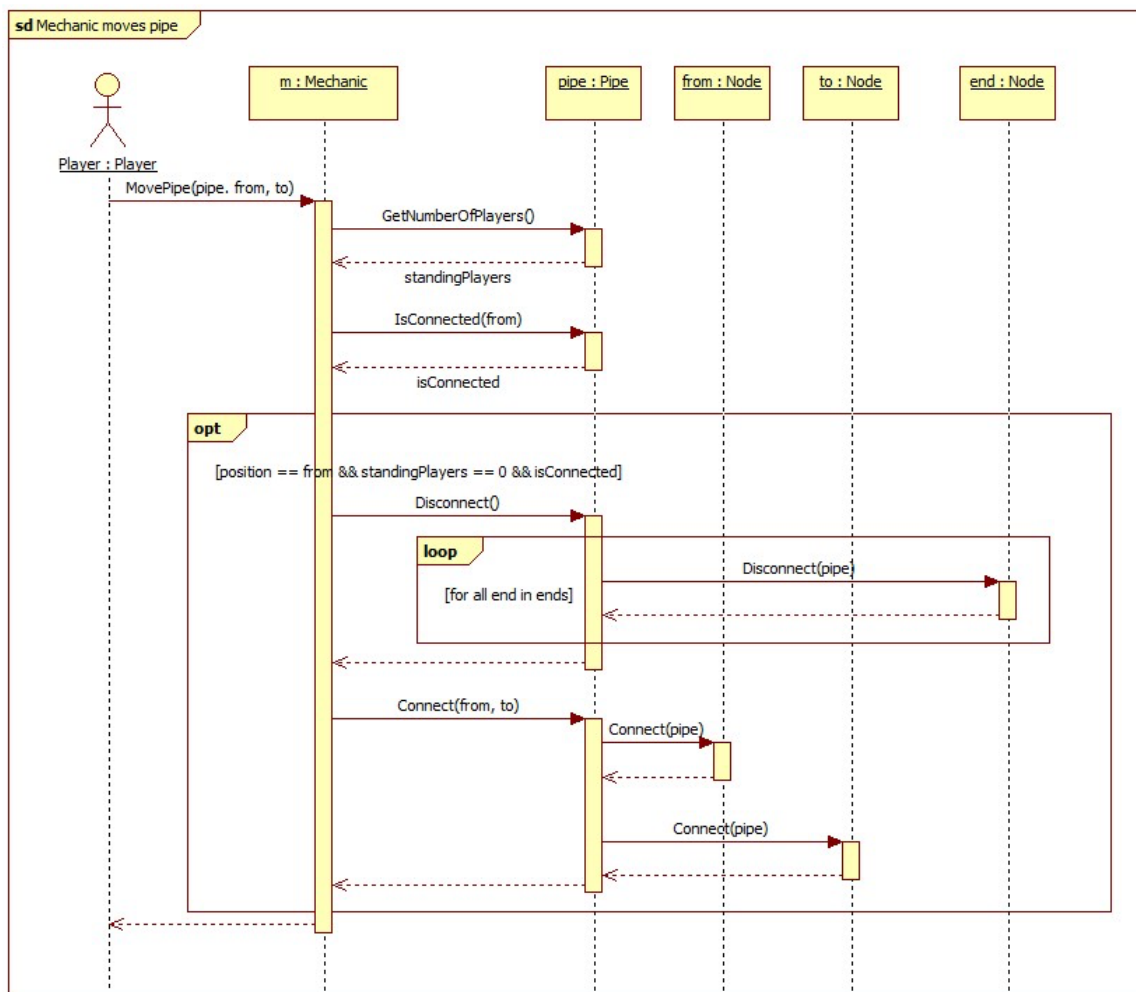
3.4.13 Mechanic Picks Up Pump

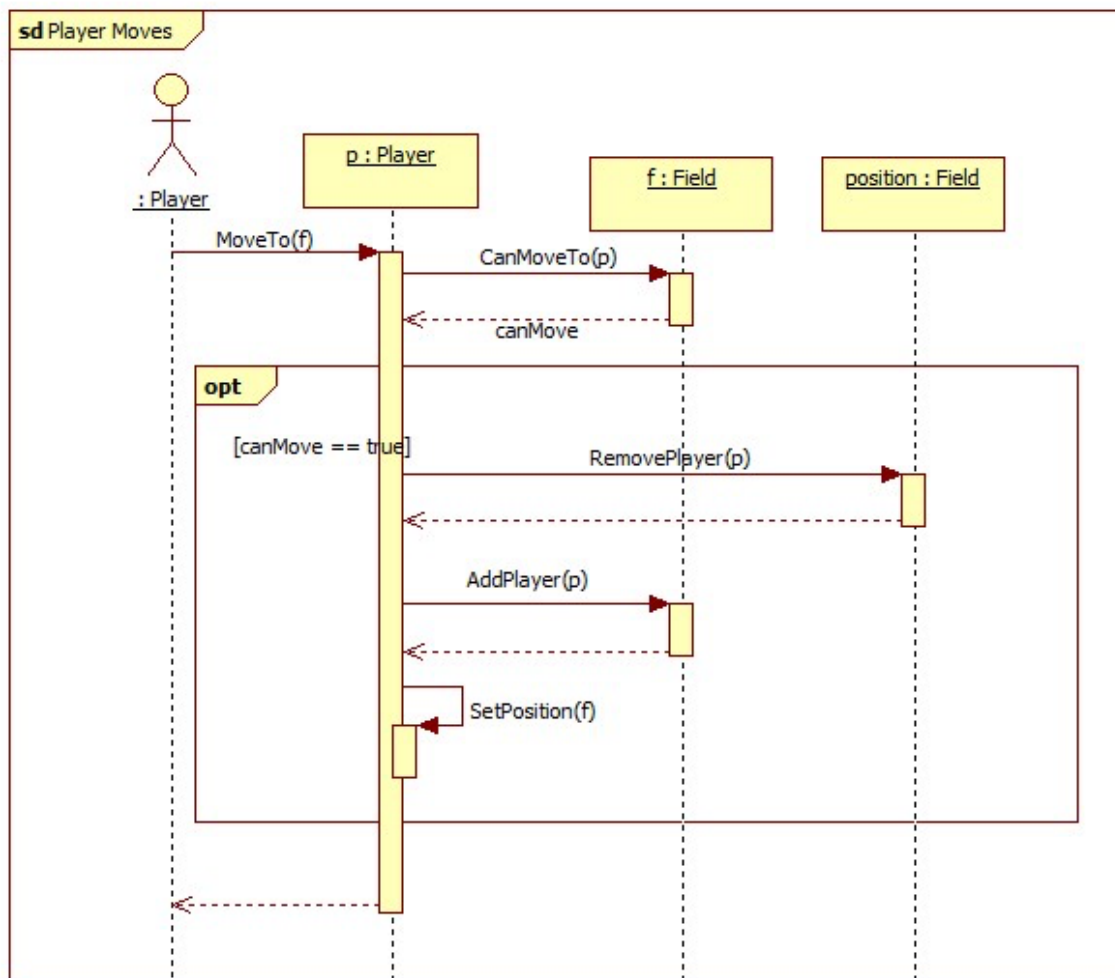
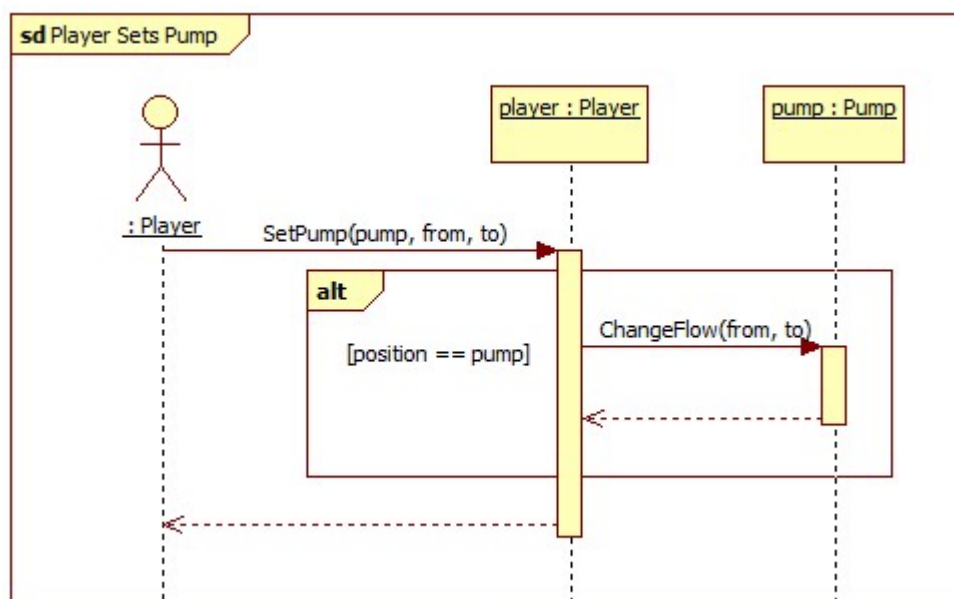


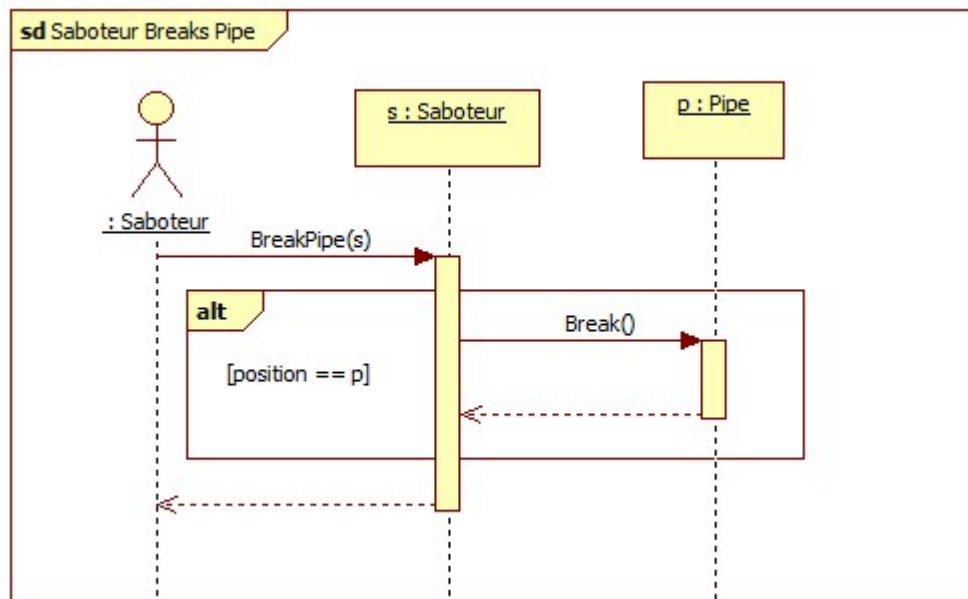
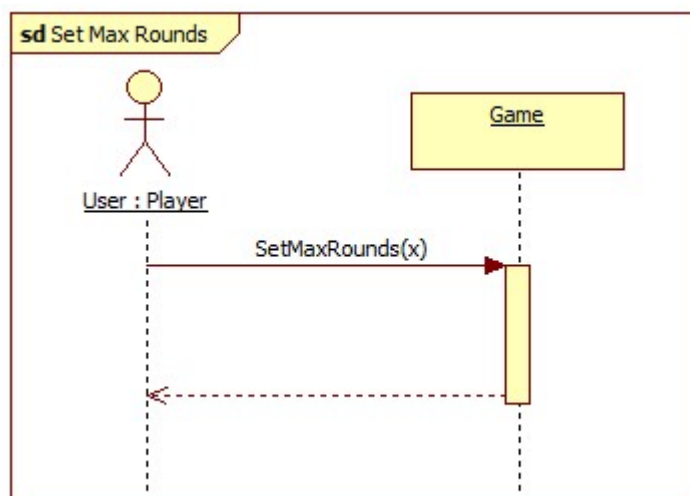
3.4.14 Mechanic Places Pipe

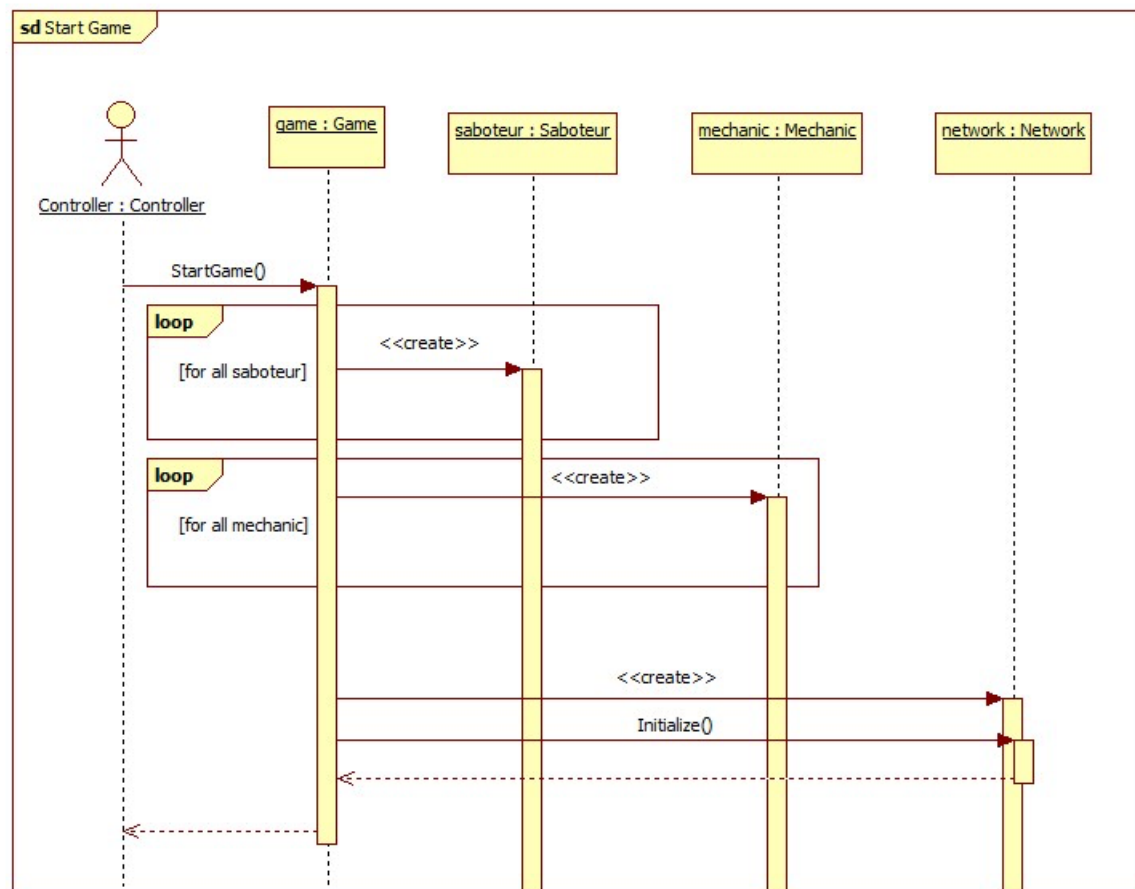


3.4.15 Mechanic Places Pump

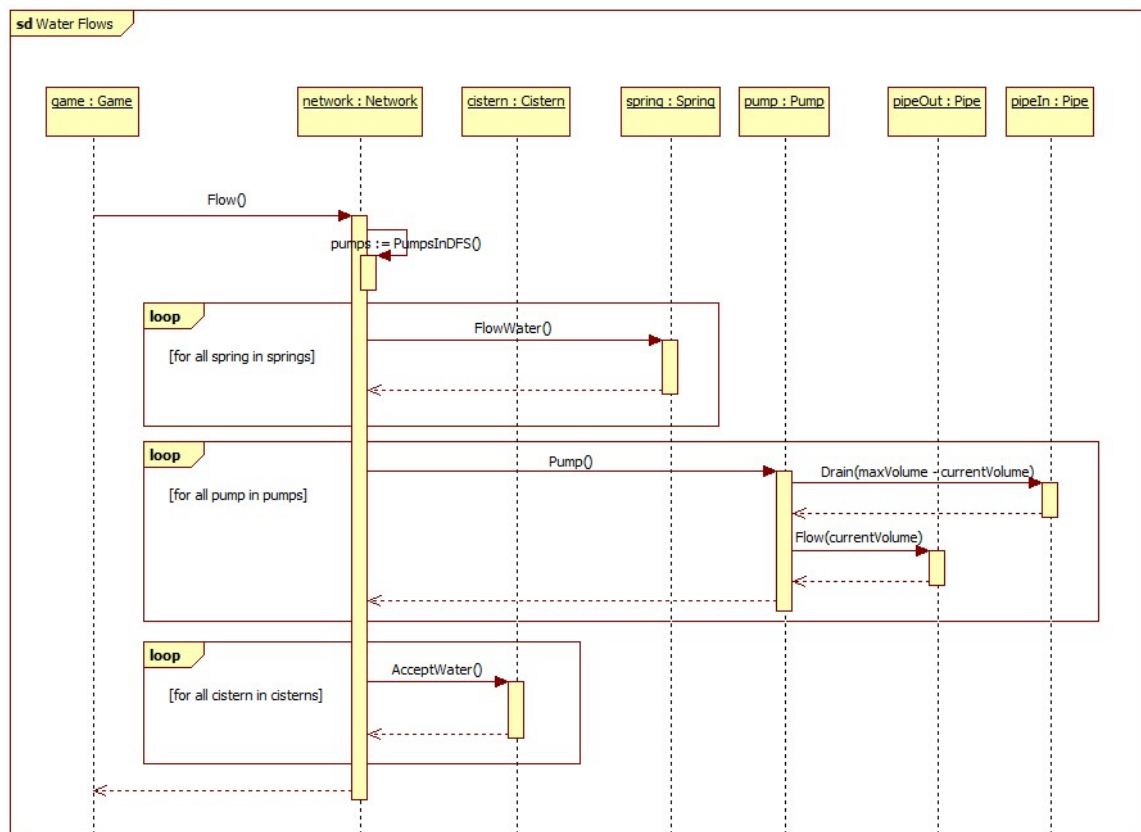
3.4.16 Mechanic Moves Pipe

3.4.17 Player Moves**3.4.18 Player Sets Pump**

3.4.19 Saboteur Breaks Pipe**3.4.20 Set Max Rounds**

3.4.21 Start Game

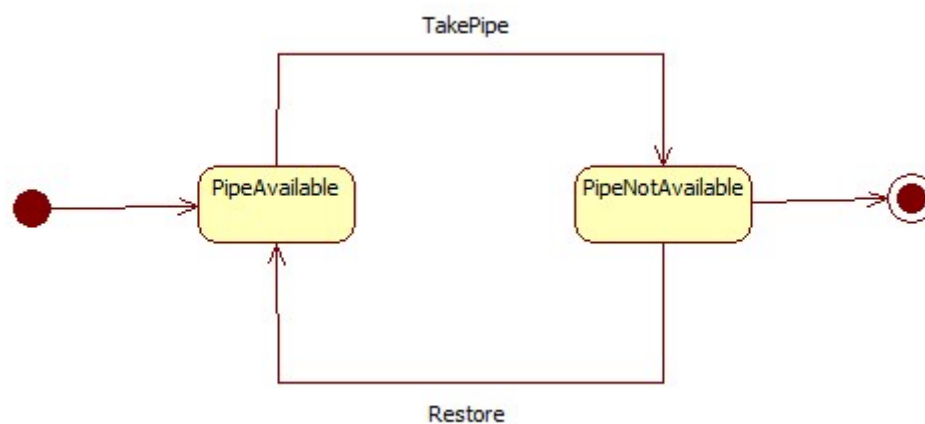
3.4.22 Water Flows



3.4.23

3.5 State-chartok

3.5.1 Cső elérhetősége a ciszternánál



4. Napló

Kezdet	Időtartam	Résztevők	Leírás
2023.03.14. 10:00	3 óra	Kurcsi Barabási Nagy Ganzer Palásti	Értekezlet. Döntés: Kurcsi elkészíti az osztálydiagramot, Barabási megírja az objektum katalógust és az osztályok leírását, Nagy, Ganzer és Palásti elkészítik a szekvencia diagramokat és state chartokat.
2023.03.14. 20:00	2,5 óra	Kurcsi	Tevékenység: Statikus struktúra diagram kezdetleges vázának megrajzolása (osztályok, kompozíciók, aggregációk, öröklődések), attribútumok nélkül
2023.03.14. 20:00	1,5 óra	Barabási	Tevékenység: Objektum katalógus megtervezése, megírása
2023.03.15. 15:00	2,5 óra	Kurcsi	Tevékenység: Struktúradiagram osztályaiba attribútumok és metódusok felvétele
2023.03.15. 20:00	2 óra	Kurcsi Nagy Ganzer Barabási Palásti	Értekezlet. Döntés: Osztálydiagram átnézése, szekvencia diagramok átbeszélése a további feladatokhoz.
2023.03.16. 17:30	4,5 óra	Nagy	Tevékenység: Szekvencia diagramok: 3.4.8, 3.4.9, 3.4.13, 3.4.14, 3.4.15, 3.4.18, 3.4.19
2023.03.17. 11:20	5 óra	Ganzer	Tevékenység: Szekvencia diagramok: 3.4.1, 3.4.6, 3.4.7, 3.4.10, 3.4.11, 3.4.12, 3.4.16, 3.4.20
2023.03.17. 11:20	2,5 óra	Barabási	Tevékenység: Osztályok részletes leírása.

2023.03.17. 12:00	4,5 óra	Palásti	Tevékenység: Szekvencia diagramok: 3.4.2, 3.4.3, 3.4.4, 3.4.5, 3.4.17, 3.4.21, 3.4.22 Valamint a state chart diagram.
2023.03.18. 13:30	30 perc	Barabási	Tevékenység: Dokumentum véglegesítése, apróbb hibajavítások