# Analízis modell kidolgozása

## Objektum katalógus

### 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.

### 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.

### Pipe

Az a pályaelem amelyen keresztül a játékosok a vizet szállíthatják a pumpák között, és ezáltal a forrásból a ciszternákba. A szabotőrök kilyukaszthatják, a szerelők pedig megjavíthatják, valamint áthelyezhetik. Egy cső mindig két játékelemet (ciszterna, forrás, pumpa) 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.

### 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 pumpa 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.

### 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.

### 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.

## Statikus struktúra diagram

Diagram

Description automatically generated

## Osztályok leírása

### 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 AddPlayer(Player p)**: Ha egy játékos rálép a pályaelemre, ezt hozzáadja a játékosok listájához. True-t térít vissza, ha sikerült játékost hozzáadni, False-t ha nem.
* **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.
* **int GetDrainedWater()**: A cisztárnába belefolyt víz mennyiségét téríti vissza.

### Field

#### Felelősség

A pályaelemekért felelős absztrakt osztály.

#### Ősosztályok

#### Interfészek

#### Attribútumok

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

#### Metódusok

* **boolean AddPlayer(Player p)**: Ha egy játékos rálép a pályaelemre, ezt hozzáadja a játékosok listájához. True-t térít vissza, ha sikerült játékost hozzáadni, False-t ha nem.
* **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.

### 4.3.3 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():** A szerelő felvesz egy csövet a ciszternáktól. Ezután az eszköztárába elérhető lesz egy cső, amelyet ezt lehelyezhet egy pumpánál.
* **void PlacePipe(Node n):** A szerelő letesz egy csövet, amelyet a ciszternáktól vett fel. Egyik végét arra a pumpára köti amelyiken áll, a másik vége szabadon lóg. Ezután az eszköztára kiürül, és ha a ciszternáknál jár az elkövetkezendő körökben, akkor újabb csövet vehet fel.
* **void PickupPump():** A szerelő felvesz egy pumpát ha van a mezőn. Ez a pumpa a szerelő eszköztárába kerül, majd ezt a pályán bárhol lehelyezheti, ahol ez megengedett.
* **void PlacePump(Pump p, Pipe pipe):** A szerelő letesz egy pumpát, amely az eszköztárában volt eddig. A  *pipe*ra letett pumpa a csövet kettéválasztja, egy új cső létrejön és a megfelelő végeit a régi és új csőnek csatlakoztatja a pumpához. Ezután az eszköztára kiürül és felvehet egy más pumpát az elkövetkezendő körökben.
* **void ConnectPipe(Pipe p, Node n):** A paraméterként kapott csomópontra csatlakoztatjuk fel a paraméterként kapott csövet.
* **void DisconnectPipe(Pipe p, Node n):** A paraméterként kapott csomópontra csatlakoztatott *p* csövet lecsatlakoztatja.

### 4.3.4 Node

#### Felelősség

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

#### Ősosztályok

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 csomópontról az adott csövet.

### 4.3.5 Pipe

#### Felelősség

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

#### Ősosztályok

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
* **wastedWater**: a lyukakon elfolyt vízmennyiség

#### 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.
* **int Flow(int amount)**: Egy pumpa vagy forrás hívja meg a csövön amikor betölti a vizet az adott csőbe. Visszatéríti azt a mennyiséget, ami befolyt a csőbe.
* **int Drain(int amount)**: Egy pumpa vagy ciszterna hívja meg a csövön amikor kiszívja annak tartalmát. Visszatéríti azt a mennyiséget, ami kifolyt a csőből.
* **boolean AddPlayer(Player p)**: Ha egy játékos rálép a pályaelemre, ezt hozzáadja a játékosok listájához. True-t térít vissza, ha sikerült játékost hozzáadni, False-t ha nem.
* **int GetVolume()**: Visszatéríti a csőben lévő víz mennyiségét.
* **int GetCapacity():** Visszatéríti a cső kapacitását.
* **void Connect(Node n)**: A cső szabad végét rácsatlakoztatja az *n* csomópontra.
* **void Disconnect(Node n)**: Lecsatlakoztatja a cső azon végét, ami az *n* csomóponthoz van csatlakoztatva.
* **Node[] GetEnds():** A cső végeit, azaz a végeire csatlakoztatott játék elemeket téríti vissza.

### 4.3.6 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 poziciót 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.

### 4.3.7 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.
* **void Pump()**: Egyik csőből a másikba pumpálja a vizet.
* **boolean AddPlayer(Player p)**: Ha egy játékos rálép a pályaelemre, ezt hozzáadja a játékosok listájához. True-t térít vissza, ha sikerült játékost hozzáadni, False-t ha nem.

### 4.3.8 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.

### 4.3.9 Spring

#### Felelősség

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

#### Ősosztályok

Field, Node.

#### Interfészek

#### Attribútumok

#### Metódusok

* **void FlowWater()**: A víz áramlik a forrásokból.
* **boolean AddPlayer(Player p)**: Ha egy játékos rálép a pályaelemre, ezt hozzáadja a játékosok listájához. True-t térít vissza, ha sikerült játékost hozzáadni, False-t ha nem.

## Szekvencia diagramok

### Diagram Description automatically generatedBreak Pump

### Diagram Description automatically generated Can Player Move To Pipe

### Diagram Description automatically generated Cistern Drains Water

### Diagram Description automatically generatedMechanic Connects Pipe

### Diagram Description automatically generatedMechanic Disconnects Pipe

### Diagram Description automatically generatedMechanic Fixes Pipe

### Diagram Description automatically generatedMechanic Fixes Pump

### Diagram Description automatically generatedMechanic Picks Up Pipe

### Diagram Description automatically generatedMechanic Picks Up Pump

### Diagram Description automatically generatedMechanic Places Pipe

### Diagram, schematic Description automatically generatedMechanic Places Pump

### Diagram Description automatically generatedPlayer Moves

### Diagram Description automatically generatedPlayer Sets Pump

### Diagram, box and whisker chart Description automatically generatedSaboteur Breaks Pipe

### Diagram Description automatically generatedWater Flows To Pipe From Spring

### Diagram Description automatically generatedWater Is Pumped

## State-chartok

### Diagram Description automatically generatedIs Pipe Broken

### Diagram Description automatically generatedIs Pump Broken

### Diagram Description automatically generatedMechanic Has Pipe

### Diagram Description automatically generatedPipe Availability At Cistern

### Diagram Description automatically generatedPipe Has Standing Players On It

### Diagram Description automatically generatedMechanic Has Pump

### Diagram Description automatically generatedWater Volume In Pipe

# *Napló*

|  |  |  |  |
| --- | --- | --- | --- |
| **Kezdet** | **Időtartam** | **Résztvevők** | **Leírás** |
| 2023.03.22. 18:00 | 2.5 óra | Barabási  Ganzer  Kurcsi  Nagy  Palásti | Értekezlet.  Döntés: Miket kell kijavítani, pontosítani, pótolni, feladatok kiosztása. |
| 2023.03.23. 17:00 | 2 óra | Barabási | Javítások: osztálydiagram javítása.  4.5.7 állapot diagram elkészítése |
| 2023.03.23. 15:00 | 2.5 óra | Nagy | Javítások: szekvenciadiagramok javítása.  4.5.6 állapot diagram elkészítése |
| 2023.03.24. 13:00 | 3 óra | Palásti | Javítások: szekvenciadiagramok javítása.  4.5.4, 4.5.5 állapot diagramok elkészítése |
| 2023.03.24. 12:00 | 2.5 óra | Ganzer | Javítások: szekvenciadiagramok javítása.  4.5.3 állapot diagram elkészítése |
| 2023.03.25. 18:00 | 2 óra | Kurcsi | Javítások: dokumentum szerkesztése.  4.5.2 állapot diagram elkészítése |