

Sivatagi Vízhálózat

Szoftvertechnológia házi feladat

Glocker Endre

KBC838

1 A FELADAT LEÍRÁSA

A drukmákori sivatagon át bonyolult csőrendszer szállítja a vizet a hegyi forrásokból a sivatagon túl elterülő városok ciszternáiba. A csőrendszer egyszerű, elágazás nélküli csövekből és napelemmel működő pumpákból áll. Egy pumpa több csövet is összeköthet, és minden pumpán külön-külön állítható, hogy melyik belekötött csőből melyik másikba pumpáljon. A többi rákötött cső eközben el van zárva. A pumpák véletlen időközönként el tudnak romlani, ilyenkor megszűnik az adott pumpánál a vízáramlás.

A csőhálózatot a szerelők tartják karban. Ők javítják meg az elromlott pumpákat, ők állítják át a pumpákat, hogy mindig a lehető legtöbb víz tudjon áthaladni a hálózaton, és ha egy cső kilyukad, az ő dolguk a cső megfoltozása is. A kilyukadt csövekből a víz kifolyik, a csövek végén lévő pumpához már nem jut belőle.

Mivel a sivatag veszélyes hely, a szerelők csak a csöveken és a pumpákon haladhatnak. A pumpáknál kikerülhetik egymást, de a csöveken már nem tudnak elmenni egymás mellett.

A hálózaton élnek a nomád szabotőrök is, akik a pumpákat tudják átállítani és a csöveket szokták kilyukasztani. A mozgásuk ugyanúgy a csőhálózathoz kötött, mint a szerelőké.

A játékot a két csapat legalább 2-2 játékosal játssza. A szabotőrök dolga, hogy minél több víz folyjon el a lyukakon, a szerelők pedig azon dolgoznak, hogy minél több víz jusson a ciszternákba. Az a csapat nyer, amelyik a játék végére több vizet szerez

2 FUNKCIONÁLIS KÖVETELMÉNYEK

2.1 ELSŐDLEGES KÖVETELMÉNYEK

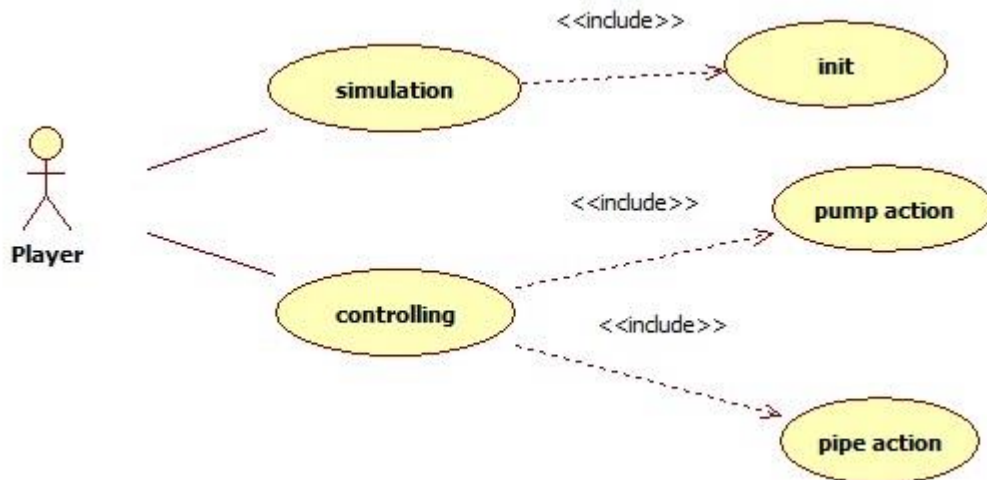
Azonosító	Leírás	Use-case
R01	A pálya elágazásmentes csövekből áll, amik pumpákba torkollanak	simulation
R02	A pumpák véletlenszerűen tönkre mennek	simulation
R03	A játékos irányítja a szabotőrt és/vagy a szerelőt	controlling
R04	A szerelő és a szabotőr a pumpákon és csöveken mozognak	controlling
R05	Bármely karakter (szerelő, szabotőr) kitudja egymást kerülni a pumpán	controlling
R06	A csöveken a karakterek nem tudják egymást kikerülni	controlling
R07	A szerelő képes megjavítani a tönkrement pumpát	pump action
R08	A szerelő képes megjavítani a kilyukadt csövet	pipe action
R09	A szerelő tudja állítani a pumpa irányát	pump action
R10	A szabotőr kitudja lyukasztani a csövet	pipe action
R11	A szabotőr tudja állítani a pumpa irányát	pump action
R12	A szerelő a pumpába befutó csövek közül tudja kiválasztani merre haladjon	controlling
R13	Egy pumpán egyszerre 1 irányból folyhat be, illetve ki a víz	simulation
R14	A kilyukadt csőből elfolyik a víz	simulation
R15	A csőhálózat vége egy ciszterna	simulation
R16	A szabotőrök a lyukakon elfolyt vízért kapnak pontot	simulation
R17	A szerelők a ciszternába folyt vízért kapnak pontot	simulation
R18	A víz folyását el kell indítani a hegyekből	init

2.2 TOVÁBBI KÖVETELMÉNYEK

Azonosító	Leírás	Use-case
R19	A kilyukadt csőből a víz nem jut el a pumpáig	simulation
R20	Az elromlott pumpából a víz nem folyik semerre	simulation

3 USE-CASE-EK

3.1 USE-CASE DIAGRAM



3.2 USE-CASE LEÍRÁSOK

Cím	simulation
Leírás	A pumpa és csőrendszeren keresztül áramlik a víz
Aktorok	Player
Főforgatókönyv	1.) Az init által megalkotott pályán elkezd áramlani a víz a ciszterna felé
Alternatív forgatókönyv	

Cím	init
Leírás	a csőhálózat komponenseinek létrehozása, közöttük a kapcsolat/csatlakozási pontok felépítése, a működés elindítása
Aktorok	Player
Főforgatókönyv	1.) A csőhálózat struktúrájának meghatározása és konkrét felépítése, pumpaszabályzás meghatározása, víz folyásának elindítása
Alternatív forgatókönyv	

Cím	controlling
Leírás	A játékos irányítja a szerelőt és a szabotőrt
Aktorok	Player
Főforgatókönyv	1.) A játékos a szerelőnek megadja a pumpán melyik csőre szeretnének menni
Alternatív forgatókönyv	1.A.) A játékos a szabotőrnek megadja a pumpán melyik csőre szeretne menni

Cím	pump action
Leírás	A pumpa állapotának módosítását teszi lehetővé
Aktorok	Player
Főforgatókönyv	1.) Működő pumpánál a szabotőr és a szerelő állíthatja a pumpa irányát
Alternatív forgatókönyv	1.A.) Nem működő pumpát megjavíthatja a szerelő

Cím	pipe action
Leírás	A cső állapotának módosítását teszi lehetővé
Aktorok	Player
Főforgatókönyv	1.) A működő csövet kilyukasztja a szabotőr
Alternatív forgatókönyv	1.A.) A lyukas csövet megjavítja a szerelő

4 STRUKTURÁLIS LEÍRÁS

4.1 AZ OSZTÁLYOK LEÍRÁSA

4.1.1 ScoreBoard

Felelősségek

Játékmenet lebonyolítását végzi el az osztály

Attribútumok

-saboteur_points: long	szabotőr csapat által gyűjtött víz
-mechanic_points: long	szerelő csapat által gyűjtött víz
-characters: Character[]	egy tömb, ami a karaktereket tárolja
-desert: Desert	Desert típusú sivatagot tárolunk

Metódusok

+calculate_points()	kiszámítja a csapatok pontszámát
+breakRandom()	a véletlenszerű eltörésért felelős függvény

4.1.2 Characters

Felelősségek

A játékban lévő karakterek absztrakt ősztyála

Attributumok

#stepable: Stepable	az az elem, amin jelenleg áll a játékos (pumpa, cső)
----------------------------	--

Metódusok

+edit(s: Stepable)	absztrakt függvény, ezzel lehet szerkeszteni egy cső állapotát
+turn(s: Stepable, p1: Pipe, p2: Pipe)	beállítja a pumpa állását p1 bemenetre, és p2 kimenetre az s pumpán
step(s: Stepable)	egy Stepable elemről lép át s-re (cső, pompa)

4.1.3 Saboteur

Felelősségek

A szabotőr karaktert valósítja meg

Metódusok

+edit(s: Stepable)	a szabotőr eltöri a csövet
---------------------------	----------------------------

4.1.4 Mechanic

Felelősségek

A szerelő karaktert valósítja meg

Metódusok:

+edit(s: Stepable)	ha csövön áll, akkor megjavítja azt, ha pumpán, akkor azt javítja meg
---------------------------	---

4.1.5 Desert

Felelősségek

A sivatagot valósítja meg, itt található a ciszterna, a víz forrását biztosító hegyek, a csövek és a pumpák

Attributumok

-st: Stepable[]	a csövek és pumpák összesége
-c: Cistern	a vizet tároló ciszterna
-m: Mountains	a víz forrásául szolgáló hegyek

Metódusok

+flowWater()	elindít egy egység vizet, amit megpróbál a ciszternáig eljuttatni
+breakRandomb()	véletlenszerűen eltör egy pumpát
+cistern_water(): int	megadja a ciszternában lévő össz vízmennyiséget
+all_water(): int	megadja az összes hegyek által "kibocsátott" vízmennyiséget

4.1.6 Field

Felelősségek

Egy interface, aminek a célja, hogy a megvalósítói kommunikálni tudjanak a csővel

Metódusok

+pushedInto(p: Pipe): bool	Alapértelmezett megvalósítása nincs
-----------------------------------	-------------------------------------

4.1.7 Cistern

Felelősségek

A vizet tároló ciszternát valósítja meg, ez jelenti a pálya végét, ide gyűlik a pályán sikeresen átfutó vízmennyiség.

Attributumok

-water: int	tárolt vízmennyiség
-------------	---------------------

Metódusok

+pushedInto(p: Pipe): bool	A ciszternáig eljutó vízmennyiséget nyeli el. Miután elnyelte ezt, akkor true értéket ad vissza
+storedWater(): int	visszaadja a tárolt vizet

4.1.8 Mountains

Felelősségek

A víz forrásául szolgáló hegyeket valósítja meg, innen kezd el folyni a víz.

Attributumok

-water: int	hegyekből elfolyt össz vízmennyiség
-first_pipe: Pipe	az első cső, amin távozik a víz

Metódusok

+start()	elindítja a víz folyását first_pipe irányába
+sourceWater(): int	visszaadja az elfolyt vizet

4.1.9 Stepable

Felelősségek

Olyan elemeket jelöl, amire a karakterek léphetnek (cső, pumpa)

Attributumok

#working: bool	működik-e az adott elem
#character: Character	a rajta álló karakter

Metódusok

+stepOn(ch: Character): bool	rálépteti a ch karaktert magára, siker esetén true értéket ad vissza
+stepOff(ch: Character): bool	lelépteti a ch karaktert magáról, siker esetén true értéket ad vissza
+break():	eltöri magát
+breakRandom()	véletlenszerűen eltöri magát
+repair()	megjavítja magát

4.1.10 Pump

Felelősségek

A pumpa működését valósítja meg, megvalósítja a Field interfacet, és a Stepable leszármazottja.

Metódusok

+pushedInto(p: Pipe): bool	p irányába folytatja a vizet
+turnPump(p1: Pipe, p2: Pipe): int	elfordítja a pumpát, úgy, hogy p1 lesz a bemenete, p2 pedig a kimenete

4.1.11 Pipe Felelősségek

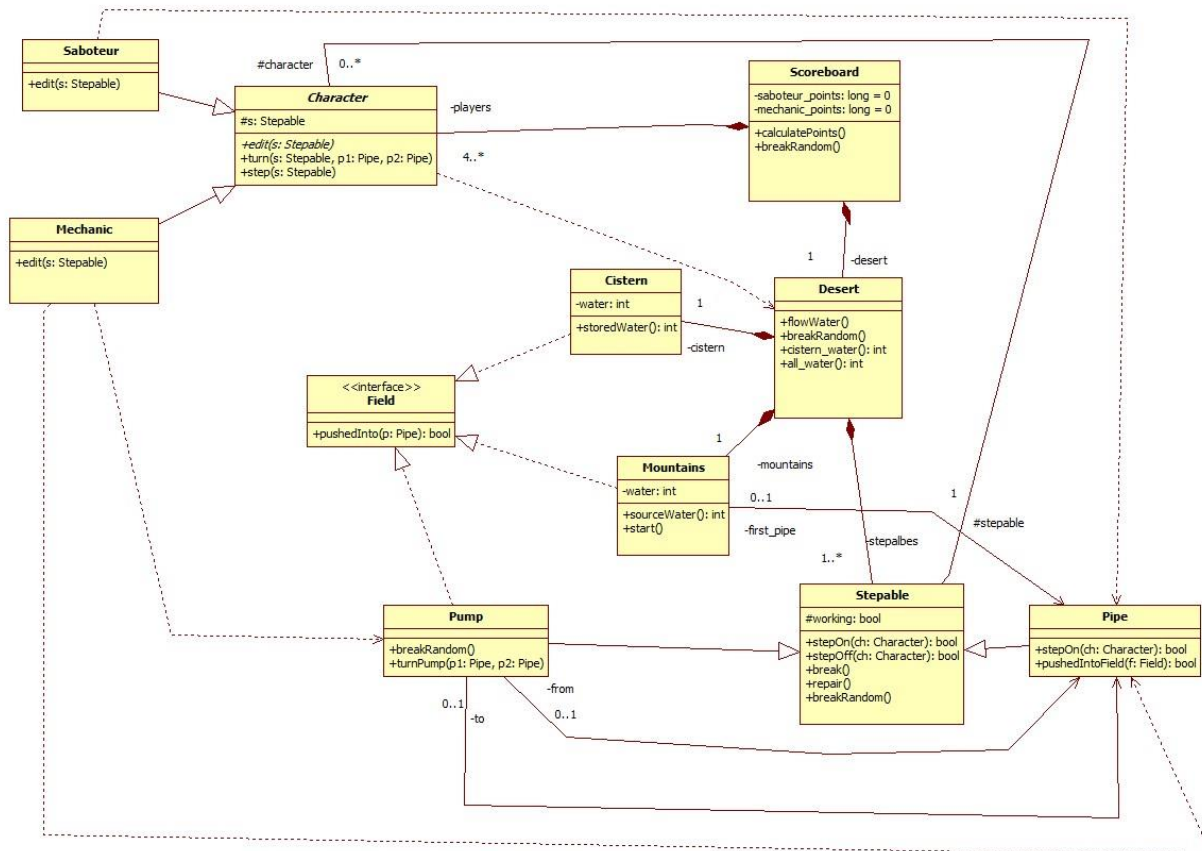
A cső működését megvalósító osztály.

Metódusok

+stepOn(ch: Character): bool	ch karakter rááll
+pushedIntoField(f: Field): bool	belefolyatja a vizet a Field -et megvalósító elemekbe

4.2 OSZTÁLYDIAGRAM

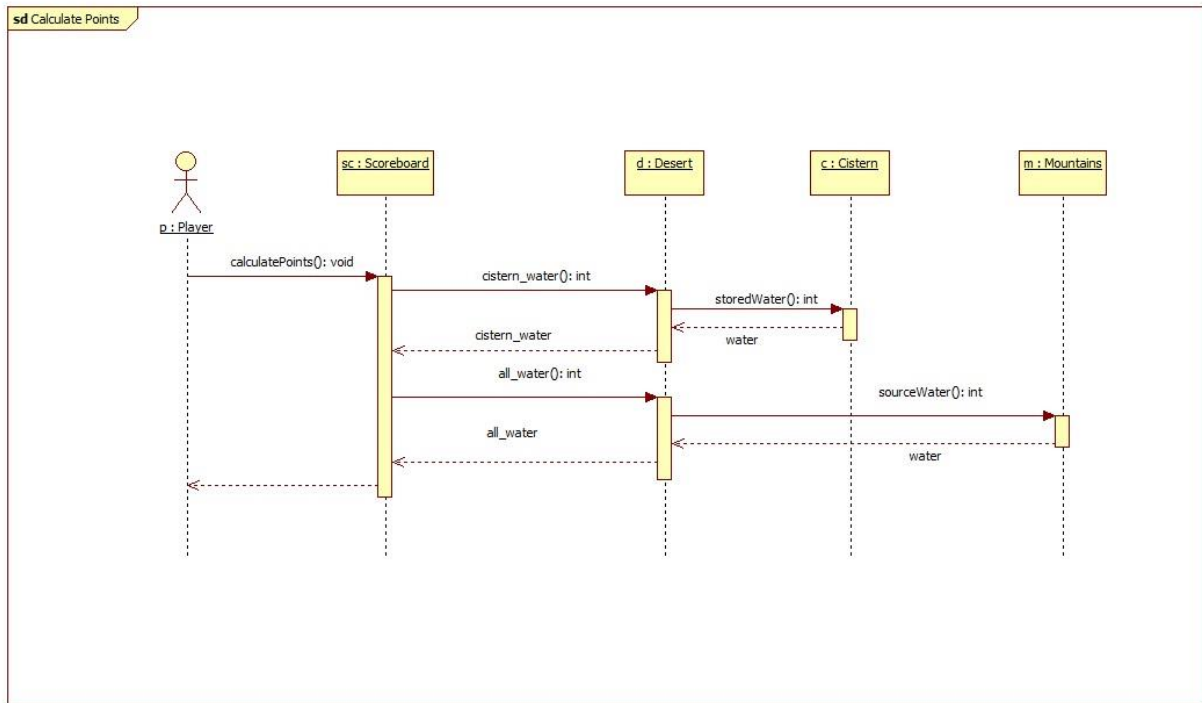
A nevesített asszociációvégekhez implicit getter és setter függvények tartoznak, amelyeket a diagram az olvashatóság kedvéért nem jelöl.



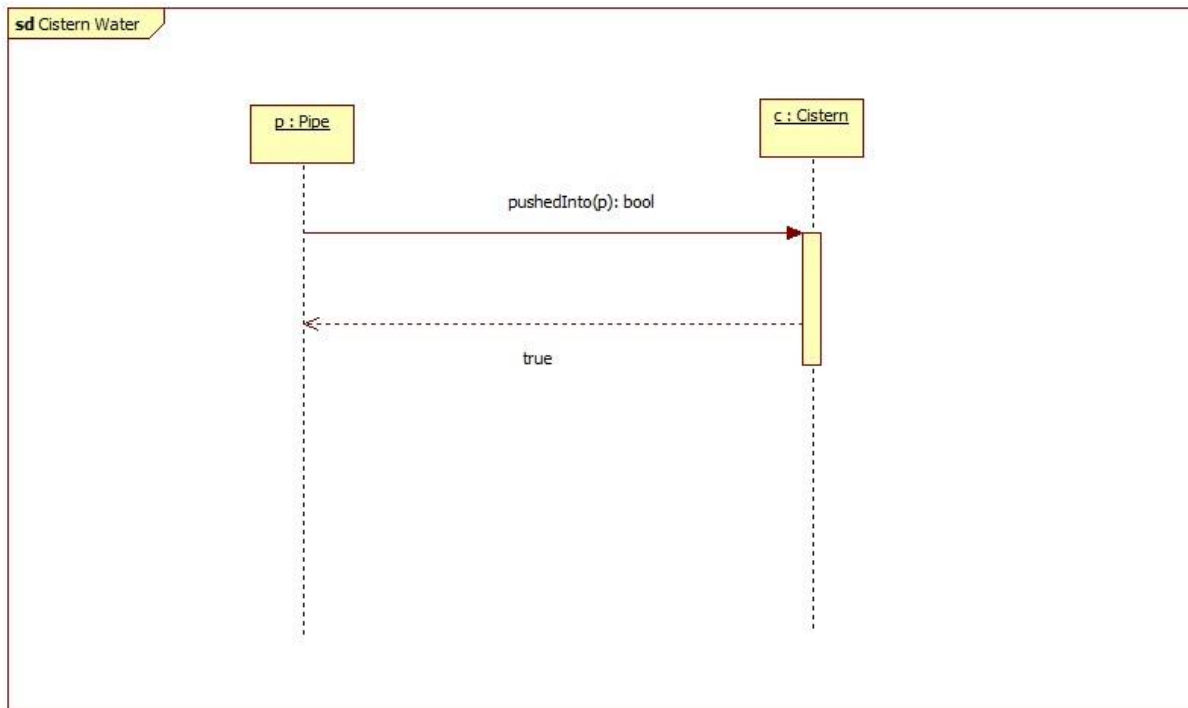
5 VISELKEDÉS LEÍRÁSA

5.1 SZEKVENCIA DIAGRAMOK

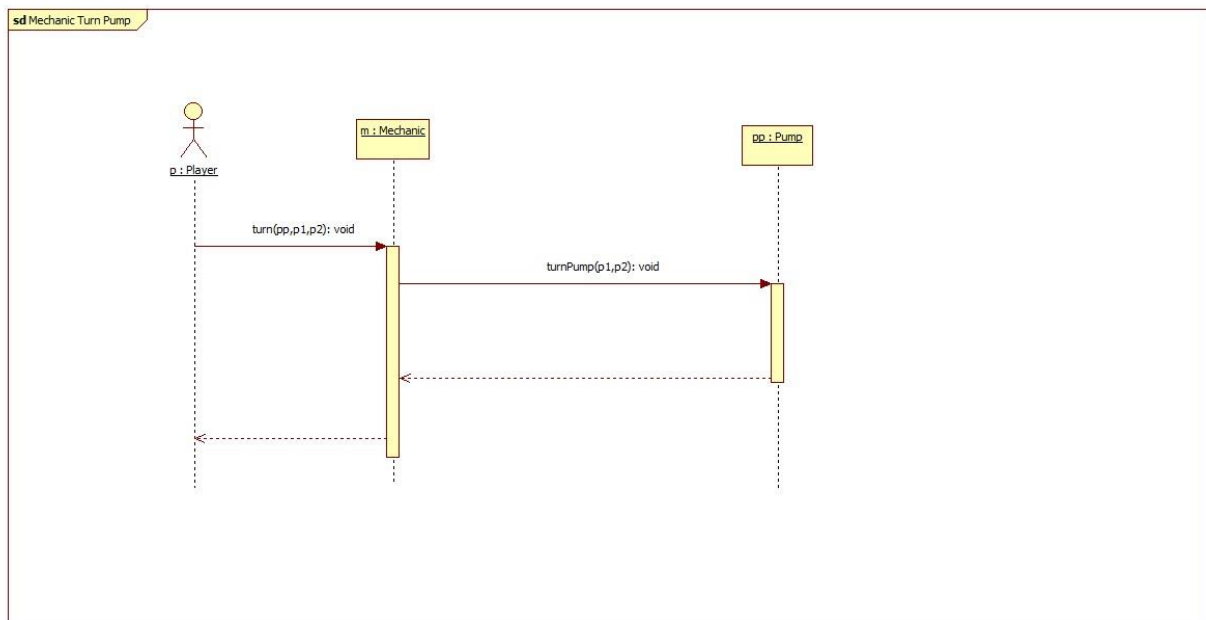
5.1.1 Pontok kiszámítása



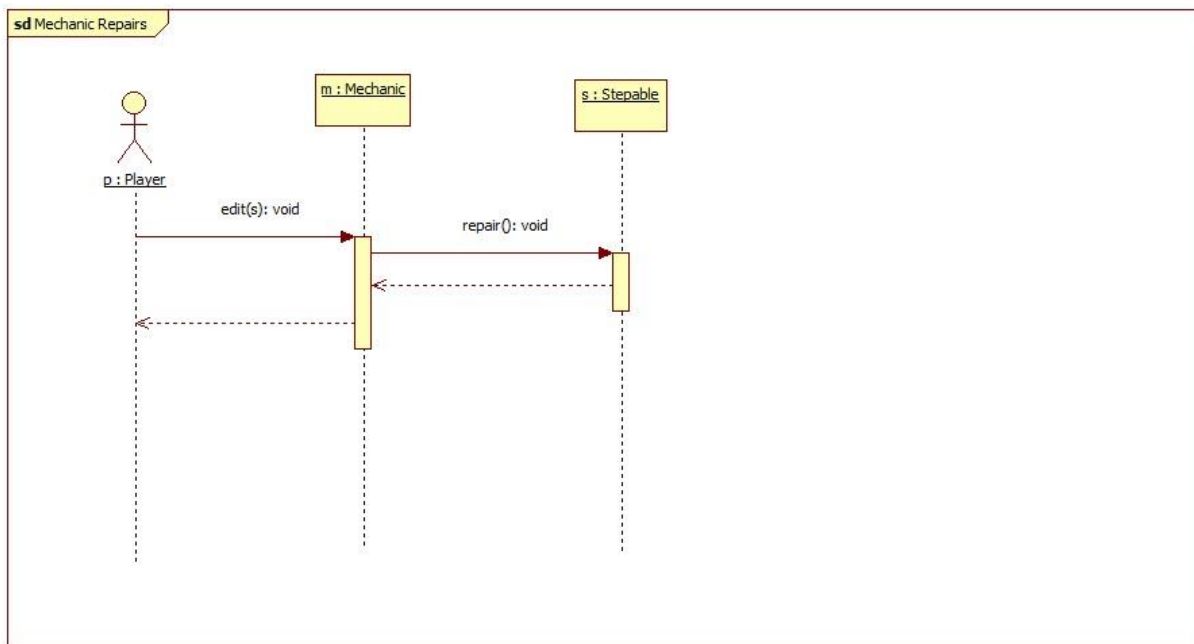
5.1.2 Ciszternába a víz áramlása



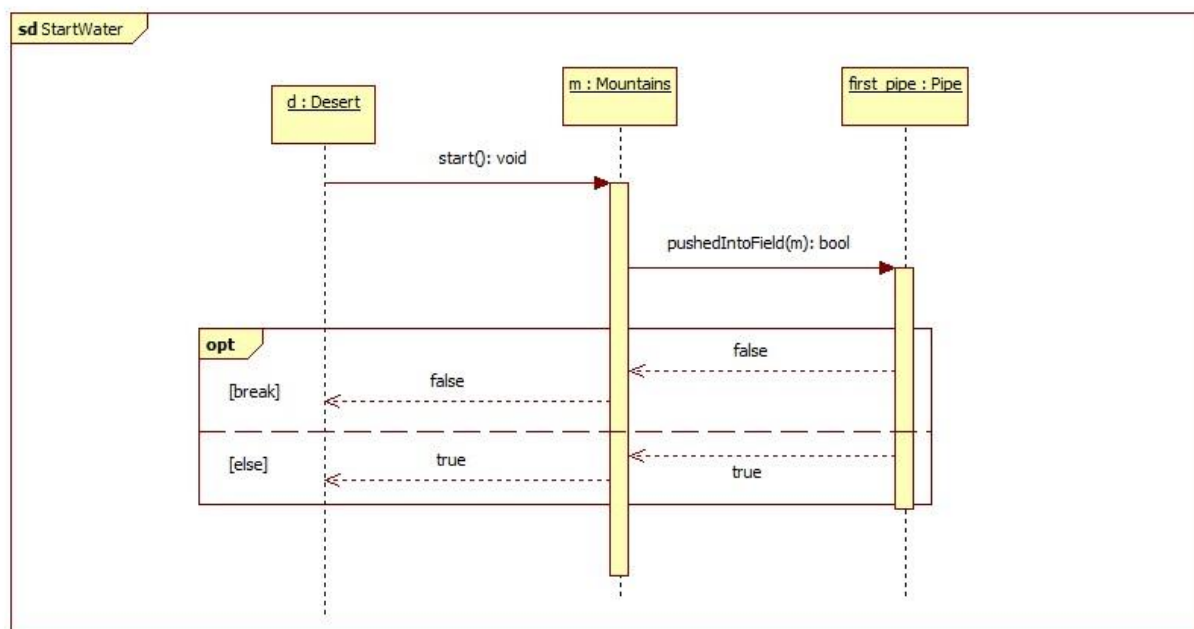
5.1.3 Szerelő átállítja a pumpát



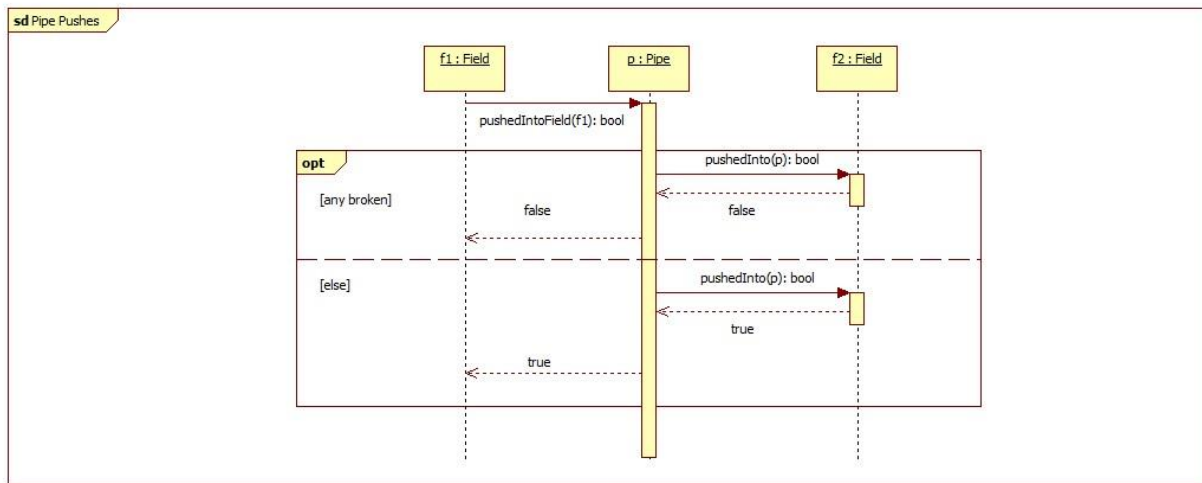
5.1.4 Szerelő megjavít valamit



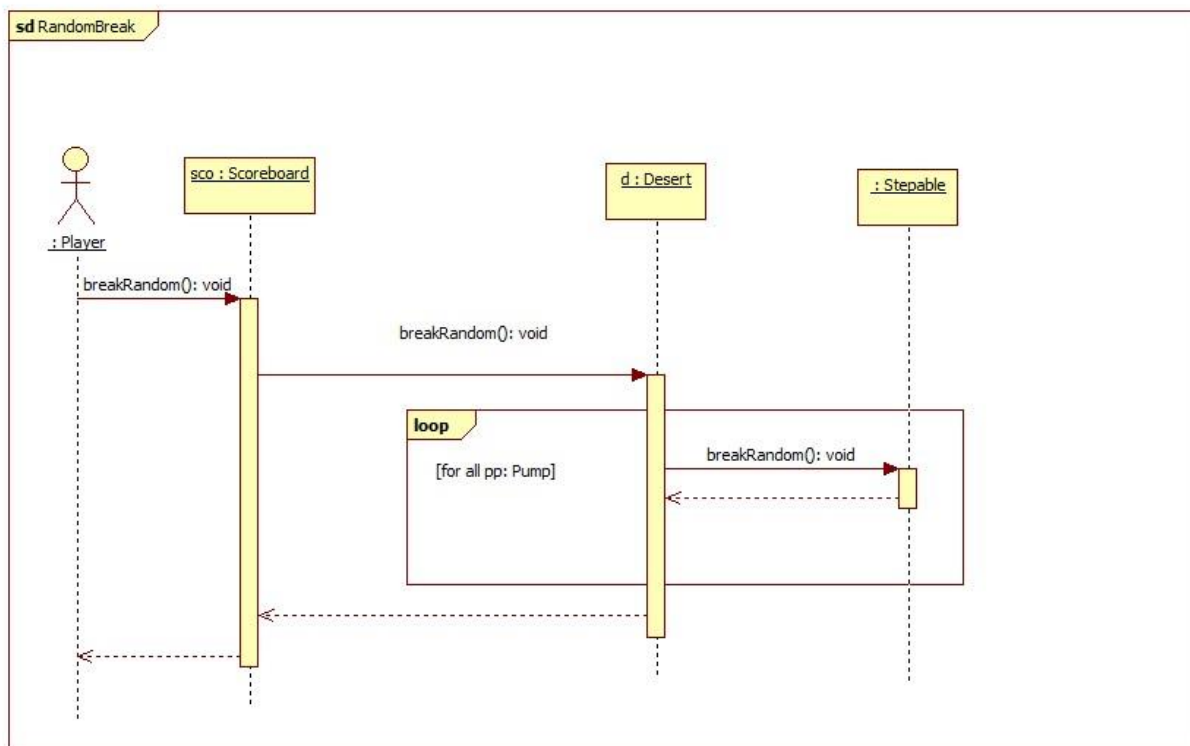
5.1.5 Hegyekből folyik a víz a sivatagba



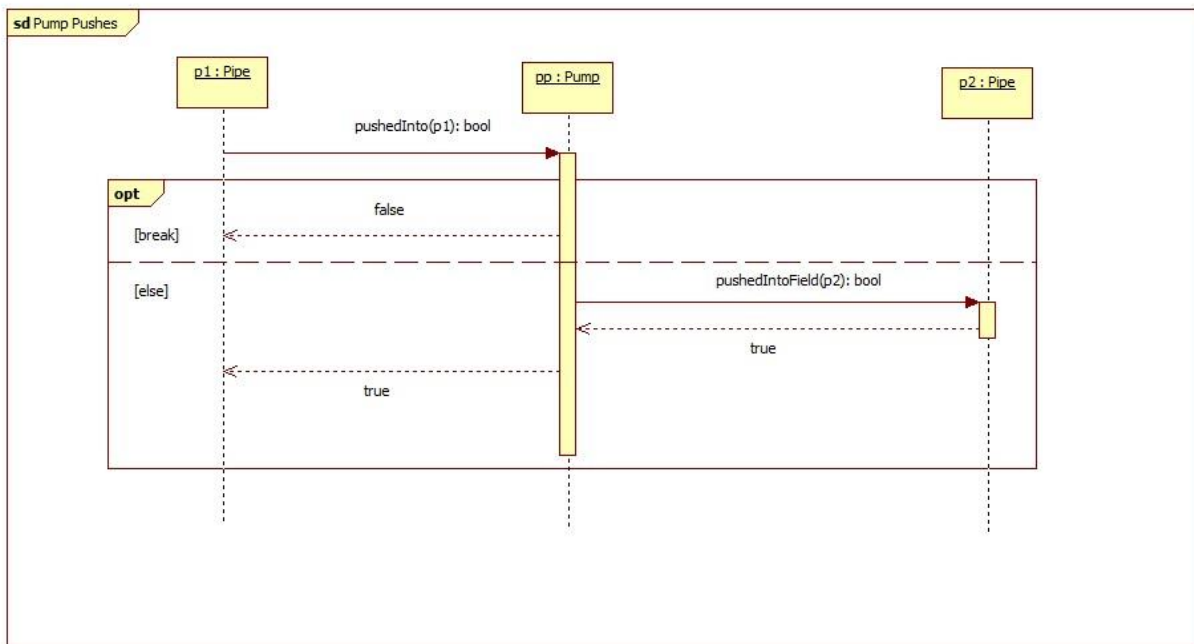
5.1.6 Cső folytatja a vizet



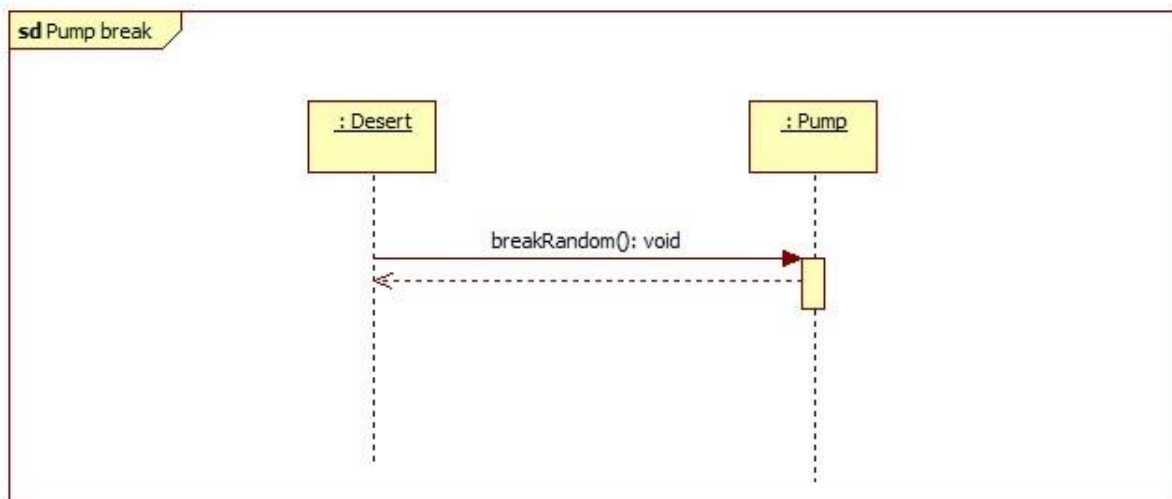
5.1.7 Véletlen eltörünk valamit



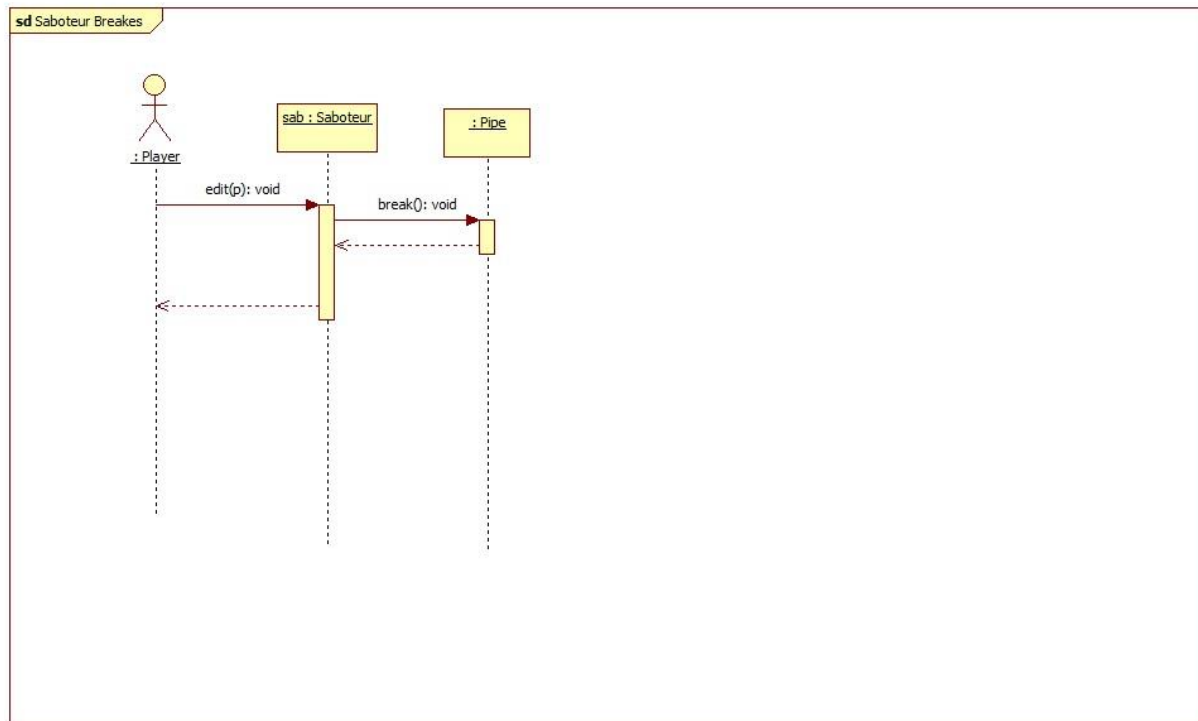
5.1.8 Pumpán átfolyatjuk a vizet



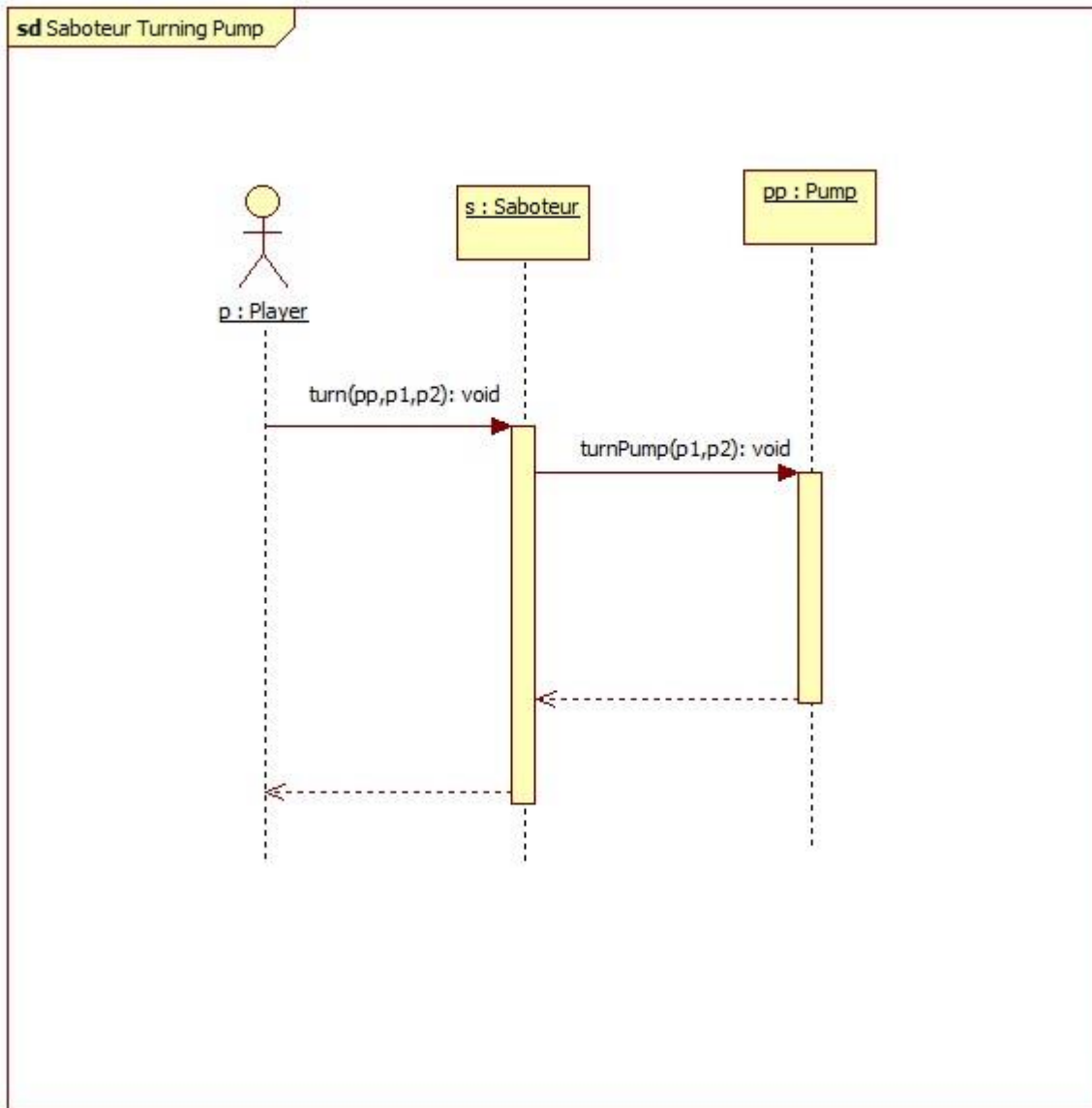
5.1.9 Véletlen eltörjük a pumpát



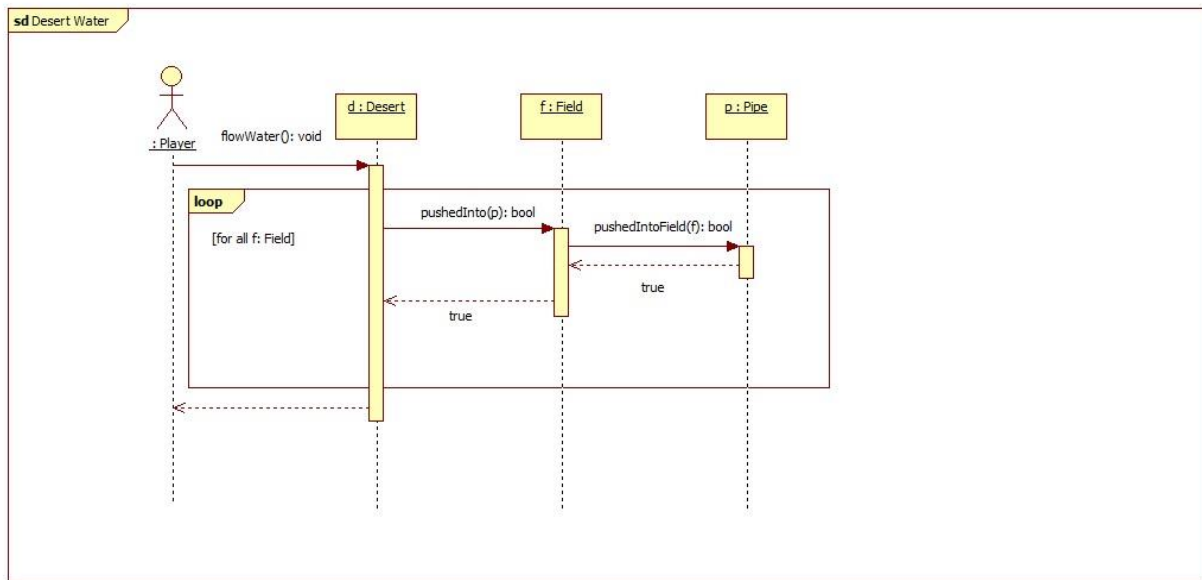
5.1.10 Szabotőr eltöri a pumpát



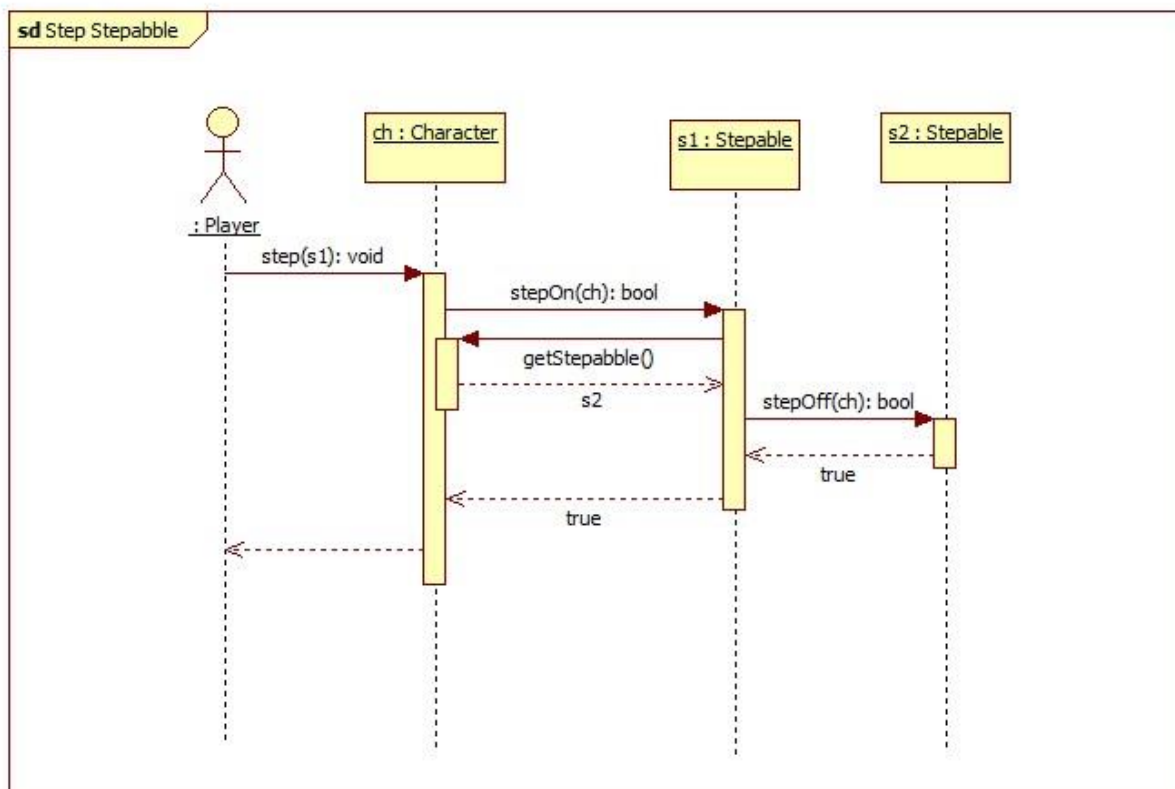
5.1.11 Szabotőr állítja a pumpát



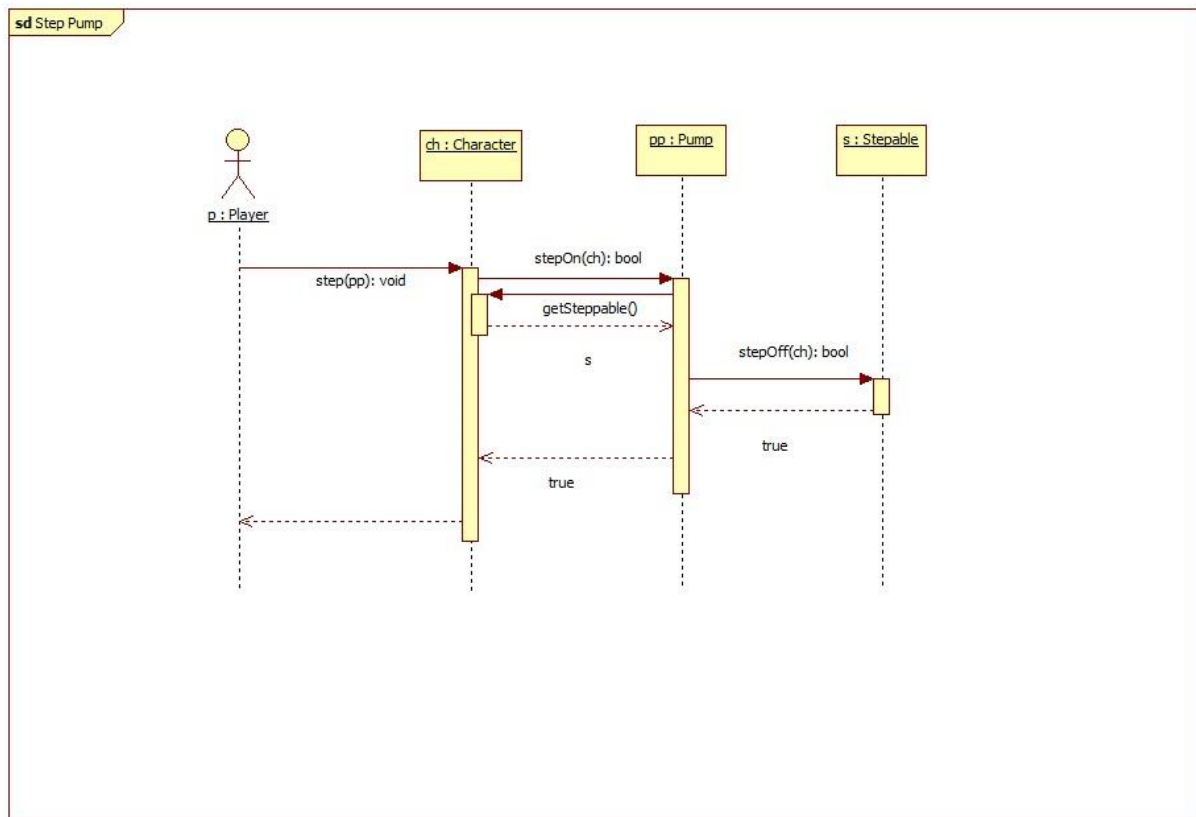
5.1.12 Víz folyatása a sivatagban



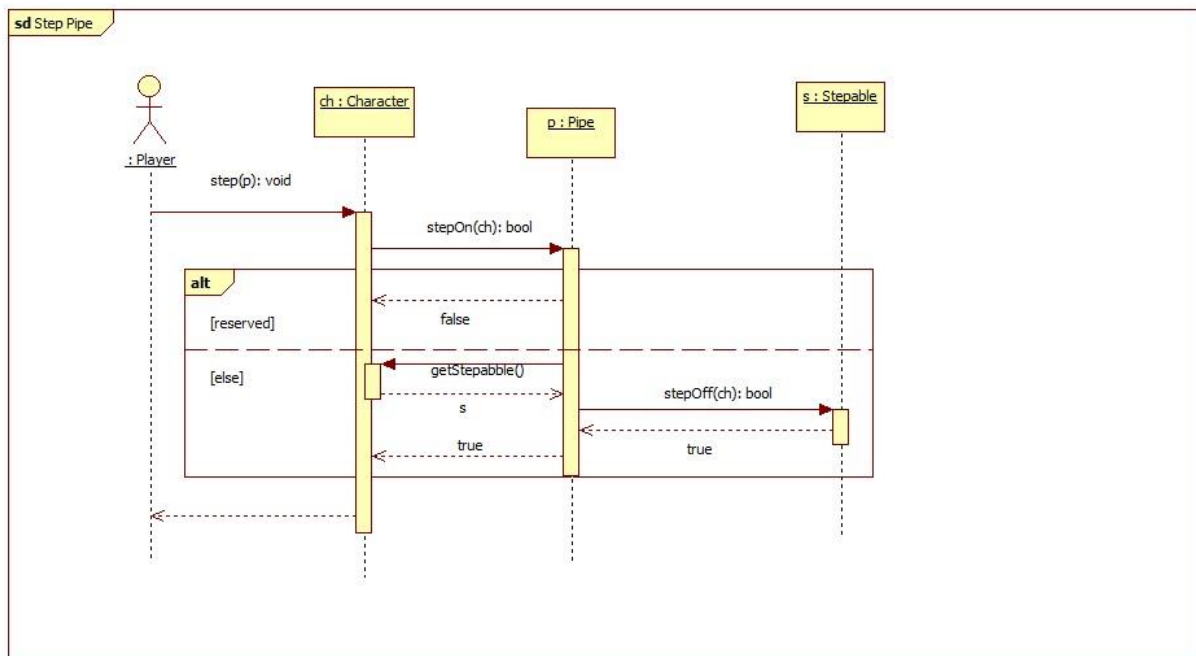
5.1.13 Elemre lépünk



5.1.14 Pumpára lépünk



5.1.15 Csőre lépünk



6 NAPLÓ

Kezdet	Időtartam	Elvégzett munka	Hivatkozások
2022.11.01	3 óra	funkcionális követelmények	2
2022.11.01	2 óra	User-Case diagramm & leírások	3
2022.11.03	2 óra	Osztálydiagramm készítés	4.2
2022.11.03	1 óra	Az osztályok leírása	4.1
2022.11.04	3 óra	Szekvencia diagrammok készítése	5.1
2022.11.05	2 óra	Hibás osztálydiagram javítása	4.2, 4.1
2022.11.05	1.5 óra	Hibás szekvencia diagrammok javítása	5.2
2022.11.06	1.5 óra	Osztálydiagrammok javítása	4.2,4.1
2022.11.06	2 óra	Szekvenciadiagrammok javítása	5.1

Összes elvégzett munka: 18

Modellező eszköz: WhiteStar UML