

2. feladat: Követelmény, projekt, funkcionálitás

36 – press_f

Konzulens:
Ludmány Balázs

Csapattagok

Karmacsi Péter	L94HDE	karmacsi.peter@freemail.hu
Suciú Barnabás	ASOG9J	szubar10@gmail.com
Bartsch Dávid	BXURDI	bartsch.david.a@gmail.com
Seres Ádám	JF204C	seres.adm@gmail.com
Takács Marcell Adrián	QITM0G	marci.takacs@gmail.com

2020.02.23.

2. Követelmény, projekt, funkcionálitás

2.1 Bevezetés

2.1.1 Cél

A dokumentum célja hogy a „press_f” csapat „Jégmező” programjának általános tudnivalót összefoglalja.

2.1.2 Szakterület

A szoftver elsődleges célja a Szoftver projekt laboratórium című tárgy teljesítése. Másodlagos célja pedig hogy szoftvert használó emberek jól szórakozzanak.

2.1.3 Definíciók, rövidítések

-

2.1.4 Hivatkozások

A feladatkiírás itt található meg: <https://www.iit.bme.hu/targyak/BMEVIIIB02>

2.1.5 Összefoglalás

A továbbiakban ismertetni fogunk minden olyan információt a szoftverrel kapcsolatban, aminek nincsenek szakmai vonzatai. Ennek elolvasása után az olvasó tisztában lesz a program felhasználásával, céljaival.

2.2 Áttekintés

2.2.1 Általános áttekintés

A szoftver legfontosabb elemei: egy pálya; a pályán lévő elemek; egy felhasználói interfész; ahol befolyásolni lehet a pályán lévő elemeket; és egy szabályrendszer, ami az elemek közötti interakciók, és a kör menete felett felügyel.

2.2.2 Funkciók

A szoftver egy videojáték, ahol minimum három játékos küzdhet a sarkvidék veszélyeivel, egymással szorosan együtt dolgozva. A játékosok eszkimók és sarkkutatók közül választhatnak, de mindenki egy csapatként dolgozik a cél érdekében. Egy véletlenszerűen elkészített jégmezőn játszódik az egész játék, ami négyzet alakú jégtáblákra van darabolva. A jégtábláknak sok különböző tulajdonságaik lehetnek: A rajtuk álló játékosok száma, a felületüket borító hó mennyisége, a hó alatt rejző esetleges tárgy, és magának a jégtáblának a stabilitása: ettől függ hogy hányan állhatnak maximum az adott jégtáblán anélkül hogy belelesnének a vízbe. Vannak olyan jégtáblák is, ahol a hó alatt egy luk rejzőzik: ebben az esetben a rálépő játékos egyből a vízben landol, bár ezt a büntetést tárgyakkal lehet enyhíteni.

A játék körökre van osztva: A játékosok maguk között előnnyel bírnak, hogy ki kezdje előbb a saját lépéseiit tartalmazó lépését, amíg mindenki nem cselekedett. (Érdemes a lehető legnagyobb előnyhöz jutni a fordulók helyes sorrendben végrehajtásával) Egy játékos a saját lépésein belül 4 akciót tehet meg, ezek az akciók rendkívül sokfajtákat lehetnek: egy mozgás akciót tehet meg egy szomszédos jégtáblára (az átlós tábla nem számít szomszédosnak), eltakaríthat egy egységes havat arról a jégtábláról amin áll, vagy használhatja valamelyik tárgyat, ami egészen különböző eredményekkel járhat, ezekről később.

A játékban időnként (minden játékos lépése végén) hóvihar csap le a jégmezőre: néhány véletlenszerűen kiválasztott jégmezőn növeli a hómennyiséget, és a rajta tartózkodó játékosok veszítenek egy egységes testhőt. (Ez gyakorlatilag az életerő pontok koncepciójának felel meg, a vízbe esés pedig kör végi halálhoz vezet) Ez természetesen nehezebbé teszi a még ki nem ásott tárgyak felfedezését, illetve elfedheti a lyukakat amik már ki voltak ásva, úgyhogy ezek helyeit érdemes megjegyezni.

Ha minden játékos lejátszotta a saját lépését, akkor azt egy körnek hívjuk: Ilyenkor a használt tárgyak hatásai véget érnek (újabb akciótáblára használatot kaphatnak), a következő körben, már ha még meg van a tárgy), a vízben lévő nem búvárruhás és nem kimentett játékosok meghalnak, az igluk elolvadnak.

A különböző karaktertípusok különböző képességekkel bírnak: A kutatók meg tudnak nézni egy táblát egy akcióért, és ekkor megtudja hogy az adott tábla hány embert bír el, míg az eszkimók iglut tudnak építeni (szintén egy akcióért), amiben átvészélhetnek egy hóvihart hűlés nélkül, és egyel több testhővel kezdenek. (4 helyet 5)

A játékosok a talált tárgyaikból szabadon bármennyit magukkal vihetnek, és tárgyaik valamint a képességeik is akár egy lépésen belül is többször rendelkezésükre állnak, természetesen minden használat után egy akciótáblára használatot kaphatnak. A kutatók és az eszkimók képességei különböző veszélyeket fednek le: A kutatók a stabilitás-alapú veszélyektől védik meg a csapatot, míg az eszkimók a fentről érkező hóvihartól. Fontos ezáltal az együtt dolgozás, de a szétválás pedig segít a játék gyors befejezésében.

Ezen kívül minden játékos használhatja egy akció áráért a tárgyat amiket a hóban találtak, ezek például:

- A lapát, amivel kétszer annyi havat lehet eltakarítani egy mezőről, mint kézzel.
- Az élelem, amivel vissza lehet gyógyítani egy testhőt. (Ilyenkor az élelem maga elfogy, és eltűnik a táskából)
- A kötél, amivel kihúzhatjuk a szomszédos mezőn vízben lévő bajtársainkat.
- A búvárruha, ami saját magunkat védi meg a vízbeeséstől.
- A pisztoly, a jelzőfény, és a patron. Ez három különböző tárgy, amit ha a csapat összegyűjt egy adott táblára, akkor egy akcióért megnyerhetik a játékot, azonban ehhez minden táblán kell tartózkodniuk.

Ha valaki vízbe esik búvárruha nélkül, akkor a lépésének vége, és a maradék akció pontjai elvesztek. (Arra koncentrál hogy ne süllyedjen el a hideg vízben) Ha a kör végén is a vízben van (nem mentették ki a csapattársai), akkor minden játékos veszít, és vége a játéknak.

2.2.3 Felhasználók

A felhasználóknak semmi, a programon kívül megtalálható előképzésre sincs szükségük a használatához. Legalább 3 játékosra van szükség.

2.2.4 Korlátozások

A szoftvernek stabilan kell működnie, és a felhasználók számára egyértelmű és konzisztens módon kell működnie, nem kívánatos funkcionálisok és programhibák nélkül.

2.2.5 Feltételezések, kapcsolatok

A megoldásunknak a saját megvalósításainktól függetlenül illeszkednie kell a feladatkiírásra.

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
R00	Létezik pálya	bemutatás	alapvető	feladatkiírás	View Level	
R01	Létre lehet hozni véletlenszerű pályát	bemutatás, kiértékelés	fontos	csapat	View Level	
R02	Az egyes jégtáblák különböző stabilitással rendelkeznek	bemutatás	alapvető	feladatkiírás	Control Character	
R03	Az egyes jégtáblák különböző hőmennyiséget tudnak magukon tárolni	bemutatás	alapvető	feladatkiírás	View Level	
R04	Az egyes jégtáblákon tárgyak vannak	bemutatás	alapvető	feladatkiírás	Control Character	
R05	A tárgyak nem látszanak ha hő van rajtuk	bemutatás	alapvető	feladatkiírás	View Level	
R06	A tárgyakat nem lehet felvenni ha hő van rajtuk	bemutatás	alapvető	feladatkiírás	Control Character	
R07	A játékosok parancsokat adhatnak a karaktereknek	bemutatás, kiértékelés	alapvető	feladatkiírás	Control Character	
R08	A parancsok tartalmazzák a mozgást	bemutatás, kiértékelés	alapvető	feladatkiírás	Control Character	
R09	A parancsok tartalmazzák a képesség használatát	bemutatás, kiértékelés	alapvető	feladatkiírás	Control Character	
R10	A parancsok	bemutatás,	alapvető	feladatkiírás	Control	

	tartalmazzák az ásást	kiértékelés			Character	
R11	A parancsok tartalmazzák a tárgyak használatát	bemutatás, kiértékelés	alapvető	feladatkiírás	Control Character	
R12	Implementálva van a „kötél” tárgy	bemutatás	alapvető	feladatkiírás	Control Character	
R13	Implementálva van a „lapát” tárgy	bemutatás	alapvető	feladatkiírás	Control Character	
R14	Implementálva van az „élelem” tárgy	bemutatás	alapvető	feladatkiírás	Control Character	
R15	Implementálva van a „búvárruha” tárgy	bemutatás	alapvető	feladatkiírás	Control Character	
R16	Implementálva vannak a győzelemhez szükséges tárgyak	bemutatás	alapvető	feladatkiírás	Control Character	
R17	Az eszkimók egygyel több élettel kezdenek	bemutatás	alapvető	feladatkiírás	Control Character	
R18	Implementálva vannak egyéb tárgyak	bemutatás	opcionális	csapat	Control Character	
R19	Egy teljes kör végén hóvihar lesz	bemutatás, kiértékelés	alapvető	feladatkiírás	View Level	
R20	A hóvihar havat szór a megfelelő mezőkre	bemutatás	alapvető	feladatkiírás	Control Character	
R21	A hóvihar kihűti a megfelelő mezőkön tartózkodó játékosokat	bemutatás	alapvető	feladatkiírás	Control Character	
R22	A jégtáblák átfordulnak ha meghaladja a rálépő játékosok száma a	bemutatás, kiértékelés	alapvető	feladatkiírás	Control Character	

	stabilitásukat					
R23	Egy játékos 4 akciót lehet egy fordulóban	bemutatás, kiértékelés	alapvető	feladatkiírás	Control Character	
R24	4 akció végeztével ki kell választani a következő játékos karakterét	bemutatás, kiértékelés	fontos	feladatkiírás	Control Character	
R25	Ha egy játékos meghal, akkor vége a játéknak	bemutatás, kiértékelés	alapvető	feladatkiírás	Control Character	
R26	Ha sikerül használni a győzelemhez szükséges tárgyakat a megfelelő módon, akkor vége a játéknak	bemutatás, kiértékelés	alapvető	feladatkiírás	Control Character	
R27	Egy menürendszerrel lehet konfigurálni a játékot	bemutatás, kiértékelés	fontos	csapat	View Level	

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
R101	A program Java nyelven készül.	bemutatás	alapvető	feladatkiírás	
R102	A programnak szüksége lehet fájlkezelési jogokra.	bemutatás	opcionális	csapat	
R103	A termék fő beviteli eszközei az egér és a billentyűzet.	bemutatás	alapvető	csapat	
R104	Az elkészült programkódnak fordíthatónak kell lenni Java SE Developer Kit 8-on.	bemutatás, kiértékelés	fontos	feladatkiírás	

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
R111	Minden alapvető követelménynek teljesülnie kell.	bemutatás, kiértékelés	alapvető	feladatkiírás	
R112	A termék futtatásához szükséges a Java Runtime Environment 8 megléte.	bemutatás	alapvető	feladatkiírás	

2.3.4 Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
R121	A forráskódnak könnyen olvashatónak kell lennie	bemutatás	fontos	csapat	Az „olvashatóság” egy szubjektív definíció, de szeretnénk minnél több ember véleményét lefedni

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

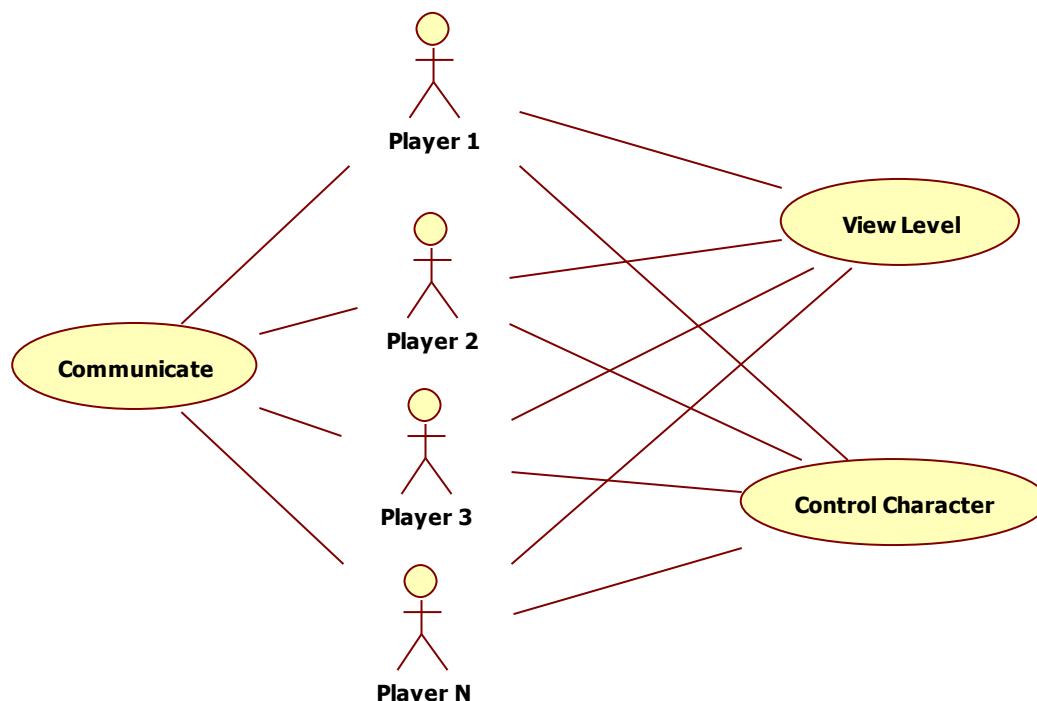
Use-case neve	Communicate
Rövid leírás	A játékosok közösen eldöntik a stratégiát
Aktorok	Player (1,2,3,N)
Főforgatókönyv	1. A A játékosok egyetértenek
Alternatív forgatókönyv	1. B A játékosok nem értenek egyet

Use-case neve	View Level
Rövid leírás	A játékosok megtekintik a játék pillanatnyi állását
Aktorok	Player (1,2,3,N)
Főforgatókönyv	1. A program kirajzolja a pályát
Főforgatókönyv	2. A A játékosok megtekintik a pályát
Alternatív forgatókönyv	3. B A játékos valamelyen képesség vagy tárgy segítségével extra információt tekint meg

Use-case neve	Control Character
---------------	-------------------

Rövid leírás	A játékos irányíthatja a karakterét 4 akció erejéig
Aktorok	Player (1,2,3,N)
Főforgatókönyv	1. A játékos kiválasztja a karakterét
Főforgatókönyv	2. A A játékos négy akciót pontig cselekedhet
Alternatív forgatókönyv	2. B A játékos meghal a négy akciót pontja elhasználása előtt
Főforgatókönyv	3. A következő játékos következik

2.4.2 Use-case diagram



2.5 Szótár

Akció: Egy egységnyi cselekvés, ahol a játékos az alábbi cselekvések közül választhat: mozgás, képesség használata, ásás, tárgy használata.

Ás: Egy egységnyi havat eltakarít a saját mezőjéről az ásást használó játékos. Fontos megkülönböztetés, hogy a lapáttal ásás az egy tárgy használata.

Búvárruha: Ha használjuk, a kör végéig megment minket a vízbe esés általi haláltól.

Élelem: Egy testhőt gyógyul a használója.

Eszkimó: Egy választható karaktertípus. Több élettel, és hóvihartól védő képességgel bír.

Hó: A jégtáblákat hó boríthatja, ez elfedi a tárgyat és a lyukakat.

Hóvihar: Játékosok lépései között véletlenszerű mezőkre hóvihar csap le: Ilyenkor az ott tartózkodó Játékosok testő-veszteséget szenvednek el, és a mezőn lévő hó mennyisége megnő.

Jégmező: Lásd pálya

Jégtábla: Lásd mező

Jelzőfény: Egyike a három tárgynak, amiket ha összegyűjtünk egy mezőre az összes játékossal együtt, akkor megnyerjük a játékot. Lásd még: Pisztoly, Patron

Kör: Ha minden játékosnak véget ért a saját lépése, akkor lejárt egy teljes kör: Ilyenkor számíthatunk járnak le a használt tárgyak hatásai, fulladnak meg a vízben lévő játékosok, majd mindenki újra cselekedhet egy következő körben.

Kötél: Amikor használatra kerül, akkor a kör végéig kimenti a szomszédos mezőkön vízbe eső bajtársainkat.

Lapát: Kétszer annyi havat lehet eltakarítani vele egy mezőről, mint kézzel.

Lépés: Egy játékos egy körön belül egyszer elkezdheti a lépését: ilyenkor, és csak ilyenkor cselekedhet maximum négy akciót. A lépések sorrendjét a játékosok határozzák meg egymás között.

Lépés: Egy játékos egy körön belüli akcióinak (maximum 4) az összessége. Lépések között jön a hóvihar.

Lyuk: A jégtáblákban lyukak lehetnek. A lyukra lépés hasonló következményekkel jár, mint ha átfordulna a tábla ahol a játékos karaktere tartózkodik.

Mező: A pálya négyzetes mezőkre van osztva, ahol egy adott mennyiséggű hó, játékos, és tárgy fér el

Mozgás: Az irányított karakter az általunk választott, szomszédos mezőre lép. (Az átlós nem szomszédos)

Pálya: A játékosok, és minden egyéb játékelem egy megadott képzeletbeli jégmezőn tartózkodnak, és minden játékakciót csak ezen belül végezhetnek el

Patron: Egyike a három tárgynak, amiket ha összegyűjtünk egy mezőre az összes játékossal együtt, akkor megnyerjük a játékot. Lásd még: Jelzőfény, Pisztoly

Pisztoly: Egyike a három tárgynak, amiket ha összegyűjtünk egy mezőre az összes játékossal együtt, akkor megnyerjük a játékot. Lásd még: Jelzőfény, Patron

Sarkkutató: Egy választható karaktertípus. Jégtáblák stabilitását megvizsgáló képességgel bír.

Tárgy: Különböző eszközök, amiket a játékosok felhasználhatnak arra hogy megnyerjék a játékot. Ezeknek egy része kötelezően kell a győzelemhez, egy másik része csak opcionális.

Testhő: Egy játékos életereje. Ha nullára csökken, akkor a játékos meghal, és vége a játéknak

2.6 Projekt terv

Feladat	Határidő	Felelős
Követelmény, projekt, funkcionálitás – beadás	febr. 24.	Bartsch
Analízis modell kidolgozása 1. – beadás	márc. 2.	
Analízis modell kidolgozása 2. – beadás	márc. 9.	
Szkeleton tervezése – beadás	márc. 16.	
Szkeleton – beadás és a forráskód herculesre való feltöltése	márc. 23.	
Prototípus koncepciója – beadás	márc. 30.	
Részletes tervezés – beadás	ápr. 6.	
Prototípus készítése, tesztelése	ápr. 20.	
Prototípus – beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése	ápr. 27.	
Grafikus felület specifikációja – beadás	máj. 4.	

Grafikus változat készítése	máj. 11.	
Grafikus változat és Összefoglalás - beadás és a forráskód herculesre való feltöltése	máj. 18.	

2.7 Napló

Kezdet	Időtartam	Részvevők	Leírás
2020.02.13. 19:00	1 óra	Karmacsi Suciu Bartsch Seres Takács	Értekezlet. Döntés: Suciu elkészíti a git repositoryt, Bartsch az első beadandót és a fedőlapot, megbeszélések időpontjai, csapatnév választás
2020.02.15. 16:00	0.5 óra	Karmacsi Suciu Bartsch Seres Takács	Értekezlet. Döntés: Git repository tesztelése és bejáratása, feladat megbeszélése
2020.02.15. 16:30	1.5 óra	Bartsch	2.1, 2.2
2020.02.15. 19:40	1.5 óra	Bartsch	2.3, 2.4
2020.02.16. 16:30	1.25 óra	Bartsch	2.5, 2.6, 2.2 javítása
2020.12.22. 15:50	1 óra	Karmacsi Suciu Bartsch Seres Takács	Értekezlet. Döntés: Javításokat kell eszközölni a dokumentációban.
2020.12.22. 16:00	0.75 óra	Bartsch	Javítások eszközölése a dokumentációban.

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Sarkkutató

Az egyik játszható karaktertípus. A játék elején 4 egységnnyi testhővel kezd és meg tudja nézni, hogy egy szomszédos jégtábla hány embert bír el.

3.1.2 Eszkimó

Az egyik játszható karaktertípus. A játék elején 5 egységnnyi testhővel kezd és képes iglut építeni.

3.1.3 Jégtábla

Ezen állhatnak a játékosok, de ha túl sokan állnak rajta egyszerre, akkor felborul és mindenki vízbeesik, aki rajta állt. Lehetnek tárgyak belefagyva, amiket a jégtáblát fedő höréteg eltávolítása után lehet kiásni.

3.1.4 Lapát

Egyike a jégből kiátható tárgyaknak. Ha a játékosnak van ásója, akkor egyszerre 2 egységnnyi havat takaríthat el egy munkaráfordítással.

3.1.5 Kötél

Egyike a jégből kiátható tárgyaknak. A játékosnak kötélre van szüksége ahhoz, hogy kimenekítsen egy másik, vízbe esett játékoszt.

3.1.6 Búvárruha

Egyike a jégből kiátható tárgyaknak. Ha egy vízbeesett játékosnak van búvárruhája, akkor nem kell másnak kimenekítenie és folytathatja a körét, mintha semmi se történt volna.

3.1.7 Élelem

Egyike a jégből kiátható tárgyaknak. A felhasználásával a játékosok növelhetik a testhőjüköt egygyel.

3.1.8 Iglu

Az eszkimók tudnak iglukat építeni. Az igluban el lehet bújni a vihar elől, így elkerülve a vihar miatti testhő csökkenést.

3.1.9 Hóvihar

Időnként feltámad egy hóvihar, ami néhány jégtáblát újabb réteg hóval borít be. Akit elkap a hóvihar, annak egységnnyivel csökken a testhője.

3.1.10 Jelzőrakéta

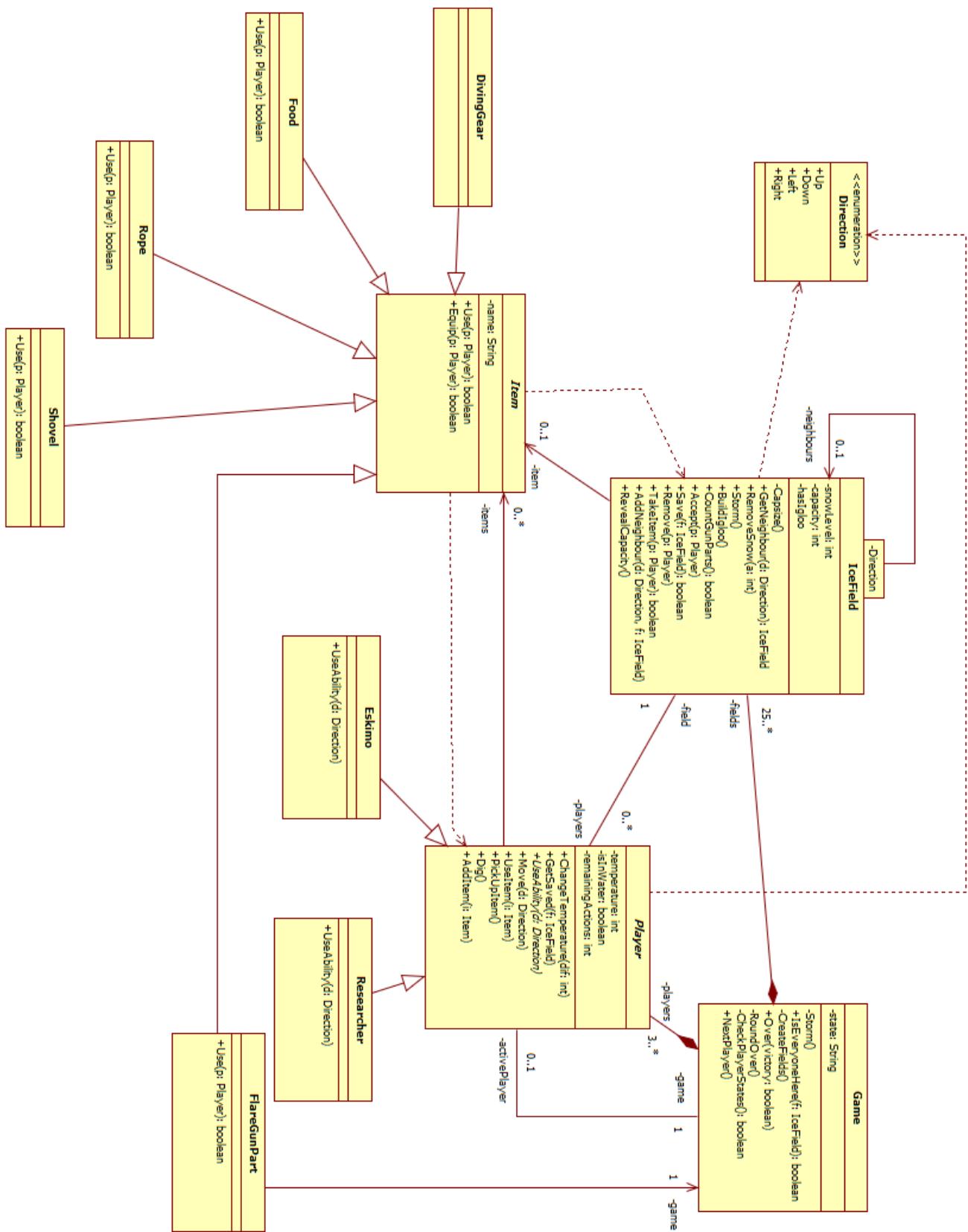
A játékot ennek az elszűrésével lehet megnyerni, amikor minden játékos egy jégtáblán áll.

3.1.11 Jelzőfény, pisztoly, patron

Ezekre van szükség a jelzőrakéta összerakásához. Ezek a játék kezdetekor jégbe vannak fagyva és ugyanúgy lehet kiásni őket, mint a többi tárgyat.

3.2 Statikus struktúra diagramok

(Megjegyzés: a nevesített asszociációvégekhez és attribútumokhoz implicit getter és setter függvények tartoznak, amelyeket a diagram az olvashatóság kedvéért nem jelöl.)



3.3 Osztályok leírása

3.3.1 DivingGear

- **Felelősség**

A játékos tulél a vízben is kör végén, ha van neki, illetve ki tűd jutni a vízből.

- **Ősosztályok**

Item → DivingGear

- **Metódusok**

Nem írja felül az Item metódusait. Az eltérő megjelenítés miatt külön osztály.

3.3.2 Eskimo

- **Felelősség**

Képes Iglut építeni.

- **Ősosztályok**

Player → Eskimo

- **Metódusok**

- **boolean UseAbility(Direction: d):** Iglut épít oda, ahol éppen áll. True-val tér vissza, ha sikerült.

3.3.3 FlareGunPart

- **Felelősség**

Ezekből kell összegyűlyteni mindenárrat és használni egy helyen állva a játék megnyeréséhez.

- **Ősosztályok**

Item → FlareGunPart

- **Metódusok**

- **boolean Use(Player p):** Megnézi, hogy teljesülnek-e a győzelem feltételei és jelez a Game-nek, ha igen. True-val tér vissza, ha sikerült használni.

3.3.4 Food

- **Felelősség**

Fel lehet használni a játékos testhőjének növelésére.

- **Ősosztályok**

Item → Food

- **Metódusok**

- **boolean Use(Player p):** A játékos testhőjét növeli. True-val tér vissza.

3.3.5 Game

- **Felelősség**

Inicializálja a játékot. Nyilvántartja a játék állapotát. minden kör végén vihart indít és ellenőrzi a játékosok állapotát.

- **Attribútumok**

- **-state:** String: A játék állapota (folyamatban van, vége + nyert/vesztett)
- **-fields:** IceField[]: A jégtáblák.
- **-players:** Player[]: A játékosok.
- **-activePlayer:** player: A soron levő játékos.

- **Metódusok**

- **boolean IsEveryoneHere(IceField f):** Megnézi, hogy mindenki az átvett fielden áll e.
- **void Over(boolean victory):** Befejezi a játékot. Ha a paraméter true, akkor a játékosok nyertek.
- **void NextPlayer():** Átállítja, hogy ki az aktuális játékos és átállítja a hátralevő akciót 4-re. Ha vége egy körnek, akkor meghívja a RoundOver()-t.

3.3.6 IceField

- **Felelősség**

Tárolja a jégtábla adatait (szomszédok, rajta álló játékosok, hószint, befagyott tárgy, iglu). Lehet mozogni közöttük, felborul, ha sokan állnak rajta, ki lehet ájni a tárgyat, ha nincs rajta hó. (A hóval fedett lyuk egy 0 kapacitású jégtábla.)

- **Attribútumok**

- **-neighbours:** IceField: A szomszédos jégtáblák a megadott irányba.
- **-snowLevel:** int: A jégtáblán levő hőréteg vastagsága.
- **-players:** Player[]: A rajta álló játékosok.
- **-item:** Item: A jégtáblába fagyott tárgy.
- **-capacity:** int: Ennyi játékos fér el rajta anélkül, hogy felborulna.
- **-hasIgloo:** True, ha van rajta iglu.

- **Metódusok**

- **IceField GetNeighbour(Direction d):** Visszaadja a szomszédos jégtáblát a megadott irányba.
- **void removeSnow(int a):** Csökkenti a hőréteg vastagságát a megadott értékkal.
- **void Storm():** Növeli a hőréteget és csökkenti a rajtalevő játékosok testhőjét, ha nincs a jégtáblán iglu.
- **void BuildIgloo():** Iglut épít a jégtáblára
- **boolean CountGunParts():** Megszámolja, hogy hány játékosnak van rajta alkatrésze a jelzőpisztolyhoz. True-val tér vissza, ha az összes alkatrész megvan.
- **void Accept(Player p):** Felveszi a játékost a rajta álló játékosok közé. Felborul ha, túl sokan állnak így már rajta.
- **boolean Save(IceField f):** Végighívja a GetSaved(Icefield f)-et a rajta álló játékosokon. True-val tér vissza, ha legalább 1 játékos ki lett mentve.
- **void Remove(Player p):** Kiveszi a játékost a rajta állók listájából.

- **boolean TakeItem(Player p):** A kapott játékosnak kiássa a befagyott tárgyat. True-val tér vissza, ha sikerült.
- **void AddNeighbour(Direction d, IceField f):** Beállítja kapott irányba levő szomszédnak a kapott IceFieldet.
- **void RevealCapacity():** Felfedi, hogy hányan férnek el a jégtáblán anélkül, hogy felborulna. Innentől kezdve végig látszik.

3.3.7 Item

- **Felelősség**

A használható tárgyak heterogén kollekciója.

- **Attribútumok**

- **-name:** string: A tárgy neve.(hogy azonosítani lehessen)

- **Metódusok**

- **boolean Use(Player p):** A kapott játékos használja a tárgyat. True-val tér vissza, ha sikerült.
- **boolean Equip(Player p):** A kapott player felveszi a tárgyat. Ha sikerül True-val tér vissza.

3.3.8 Player

- **Felelősség**

Nyilvántartja a játékos adatait. A játékos akciót végrehajtja. Ha elfogynak az akciói, szól a Game-nek, hogy jöhet a következő.

- **Attribútumok**

- **-field:** IceField: A jégtábla, amin éppen áll
- **-temperature:** int: A játékos jelenlegi testhője
- **-isInWater:** boolean: True, ha vízbe van esve.
- **-remainingActions:** int: A hátralévő akciók száma a játékos körében.
- **-items:** Item[]: A játékos birtokolt tárgyai.
- **-game:** Game: A játék.

- **Metódusok**

- **void ChangeTemperature(int dif):** Változtatja a játékos testhőjét a megadott értékkel(nem lehet 0 alá)
- **boolean GetSaved(IceField: f):** Kímenekítik a vízből a kapott IceField-re, ha vízbe van esve. True-val tér vissza, ha ki lett mentve.
- **void UseAbility(Direction: d):** Használja a játékos képességét a megadott irányba. Ha null-t vesz át, akkor ott, ahol áll a játékos. Levon egy akciót, ha sikerült.
- **void Move(Direction: d):** Másik jégtáblára mozog a megadott irányba. Levon egy akciót, ha sikerült.
- **void UseItem(Item i):** Használja a megadott tárgyat a játékos. **void PickUpItem():** Kiássa a jégből a belefagyott tárgyat. Levon egy akciót, ha sikerült.
- **void Dig():** Eltakarít egy réteg havat a jégmezőről, ahol áll. **void AddItem(Item i):** Hozzáadja az Item-et a játékos tárgyaihoz. Levon egy akciót, ha sikerült.

- **void PickUpItem()**: Kiássa a jégtáblából a befagyott tárgyat.
- **void AddItem(Item i)**: A játékos táskájába teszi a tárgyat.

3.3.9 Researcher

- **Felelősség**

Meg tudja nézni egy IceField kapacitását.

- **Ősosztályok**

Player → Researcher

- **Metódusok**

- **boolean UseAbility(Direction: d)**: Meg tudja nézni egy szomszédos(vagy ahol áll, ha d null) IceField kapacitását. True-val tér vissza.

3.3.10 Rope

- **Felelősség**

Ki lehet menteni a játékosokat a vízből a környező mezőkről.

- **Ősosztályok**

Item → Rope

- **Metódusok**

- **boolean Use(Player p)**: Kimenti a vízbeesett játékosokat a szomszédos mezőkről. True-val tér vissza, ha legalább egy játékest kimentett.

3.3.11 Shovel

- **Felelősség**

Havat lehet vele eltakarítani a jégmezőről.

- **Ősosztályok**

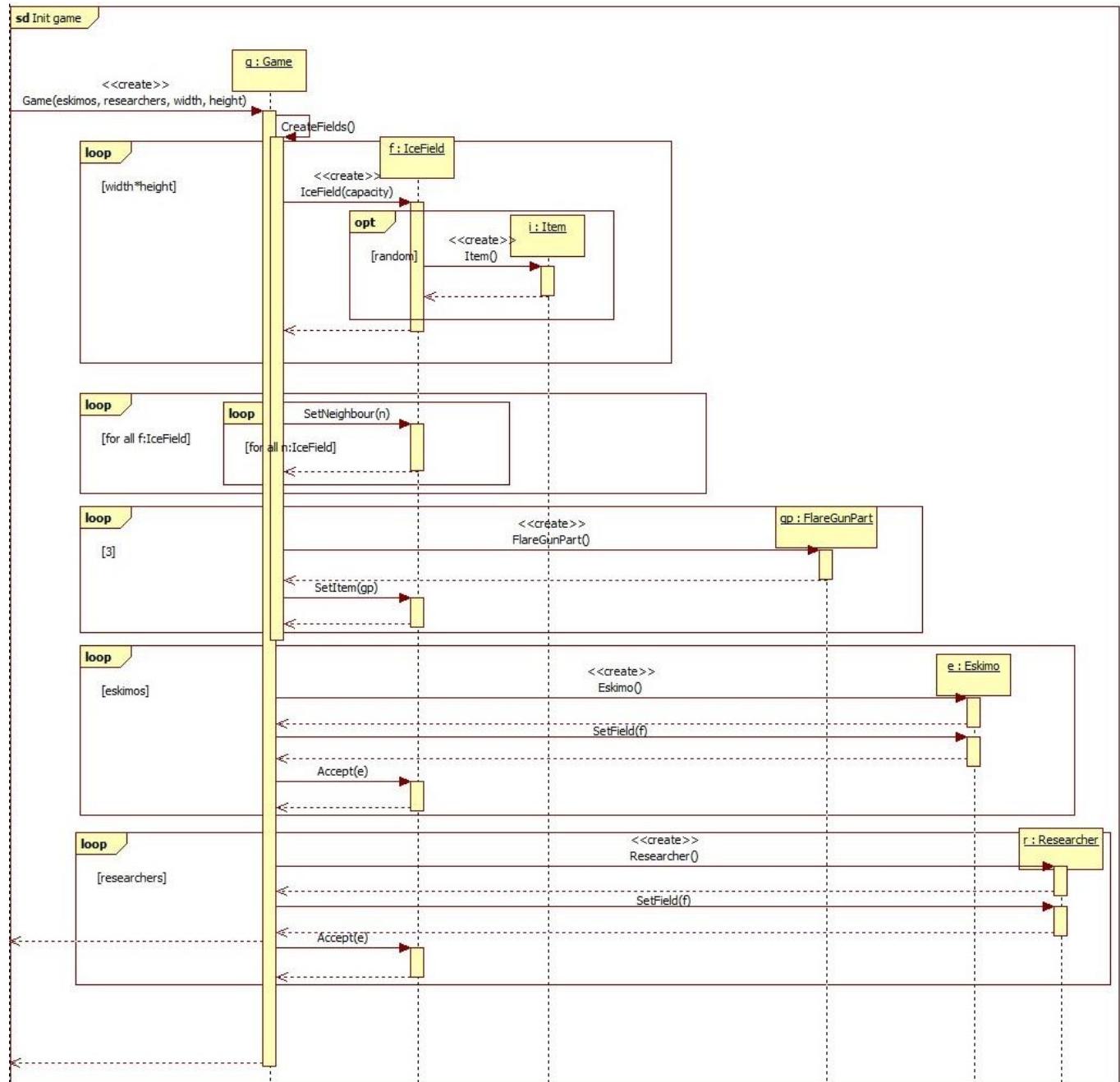
Item → Food

- **Metódusok**

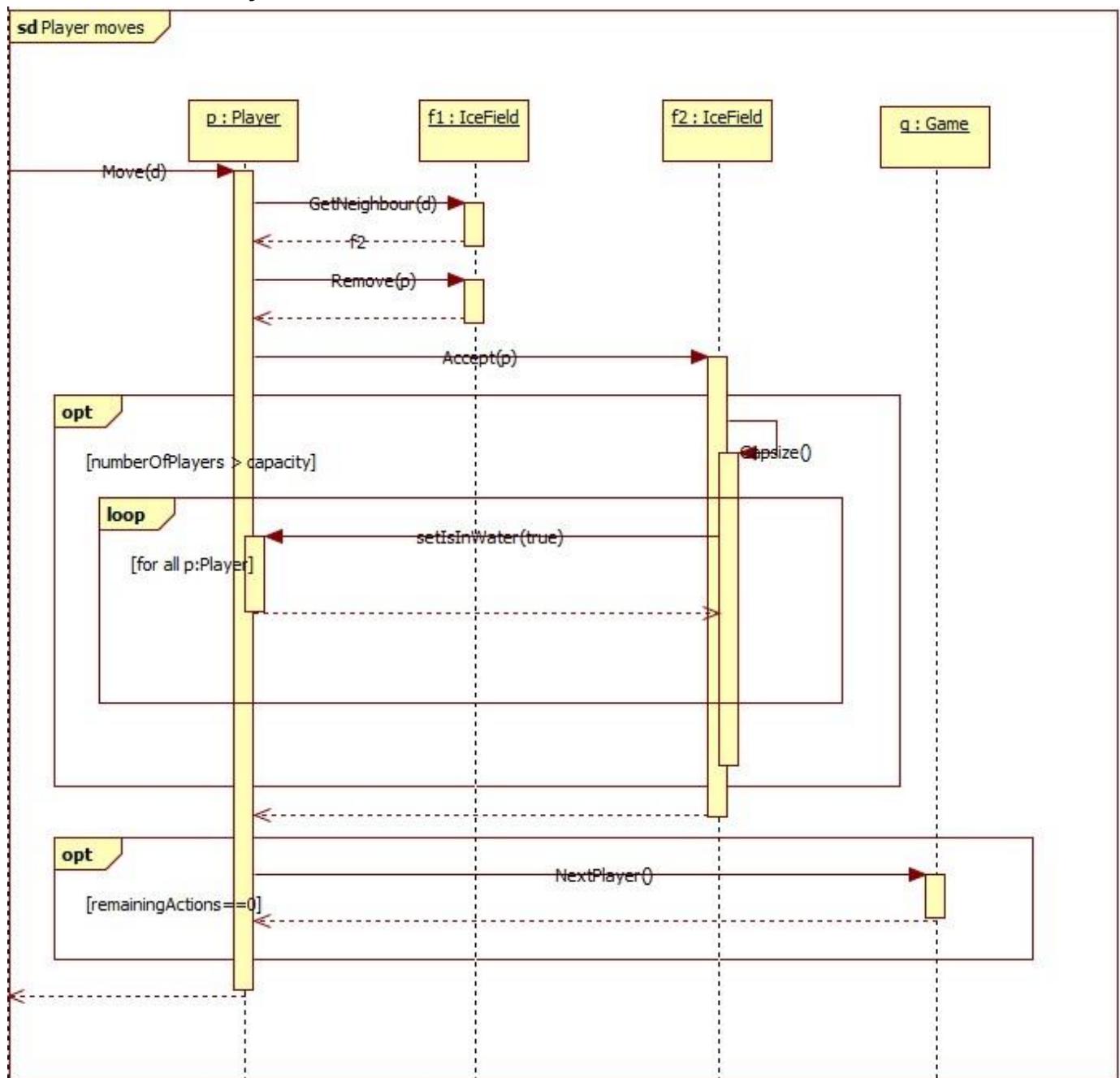
- **boolean Use(Player p)**: Két egységnyi hóréteget takarít el a jégmezőről, ahol a használója áll.

3.4 Szekvencia diagramok

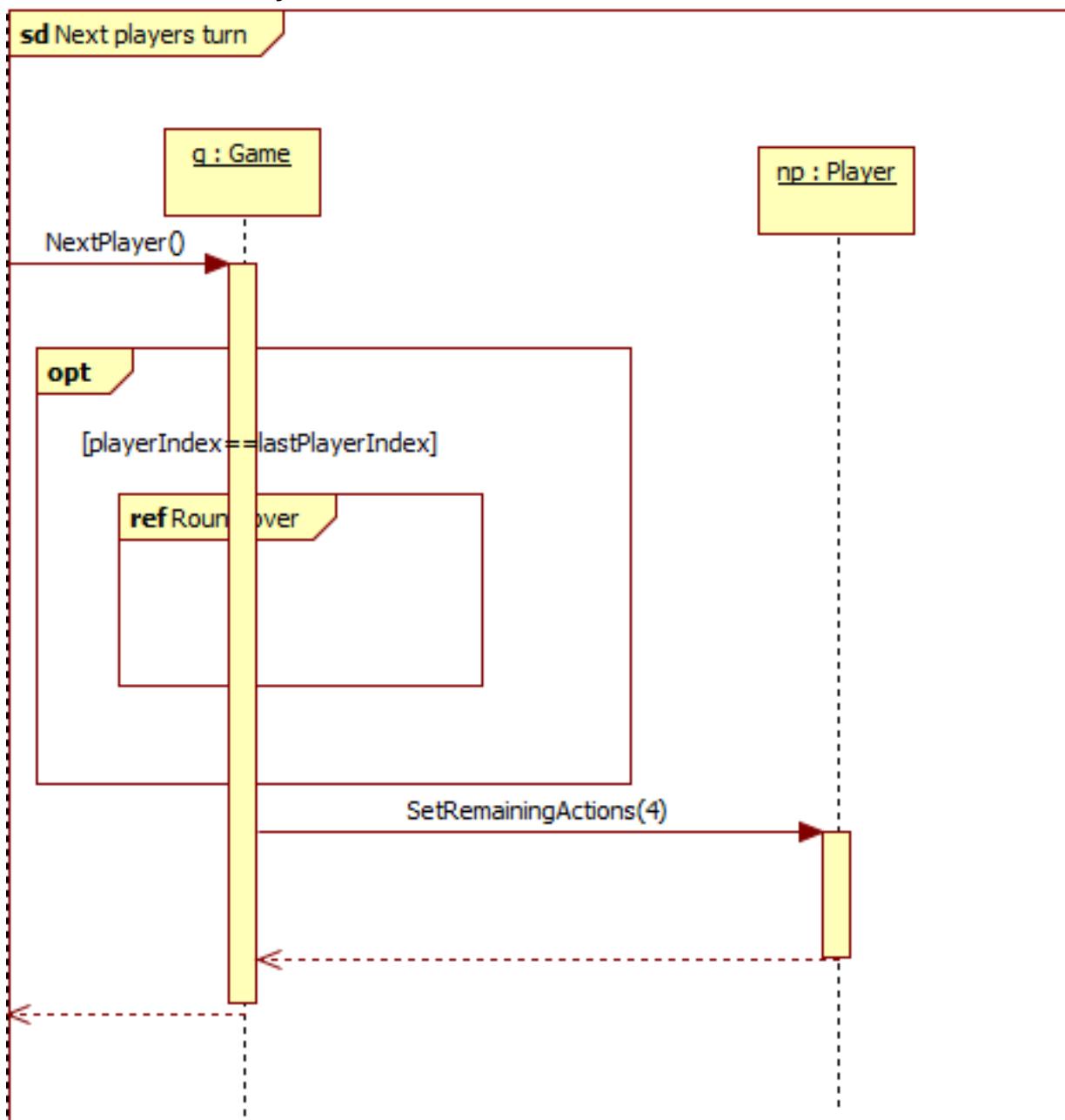
3.4.1 Init Game



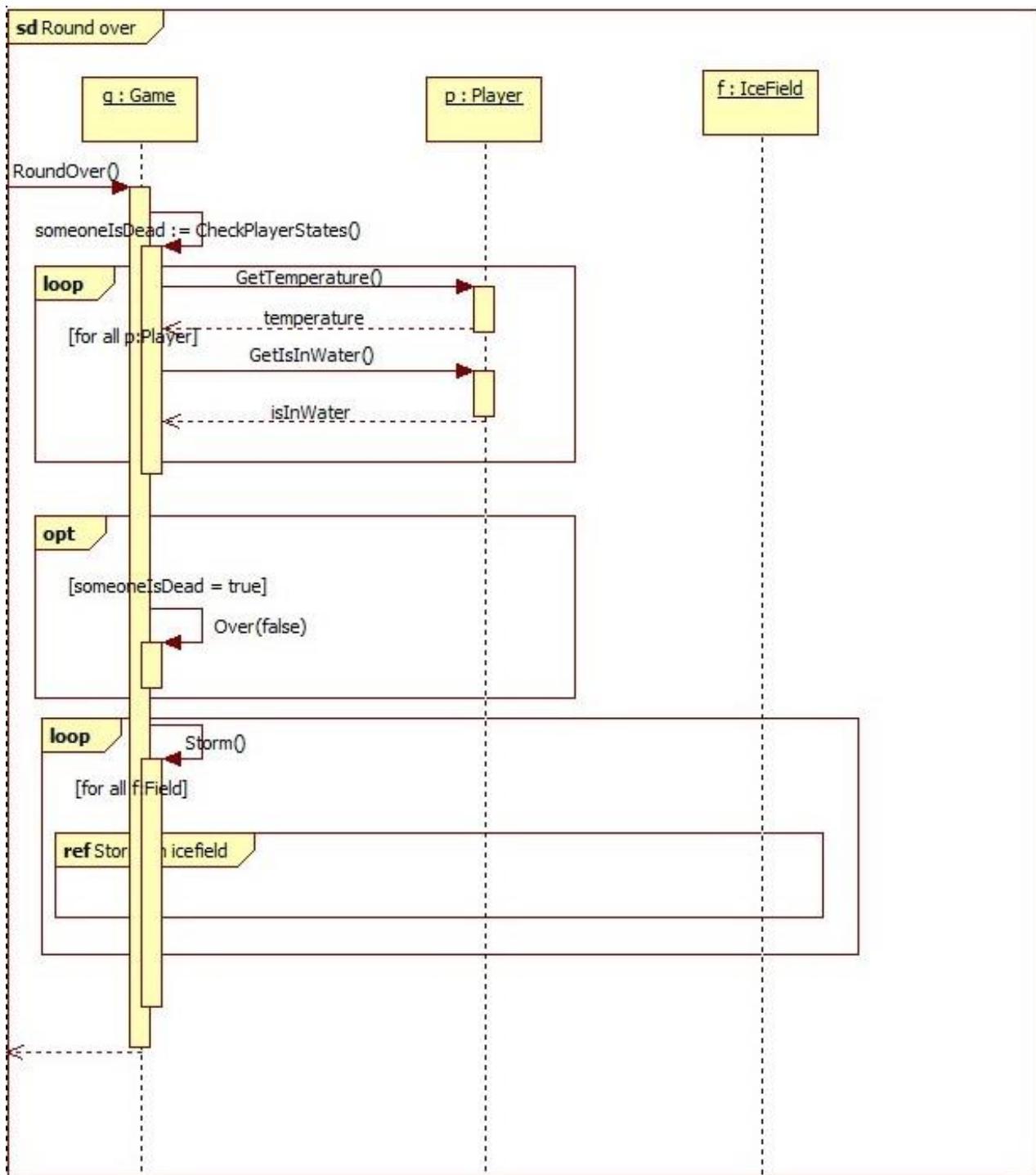
3.4.2 Player Moves



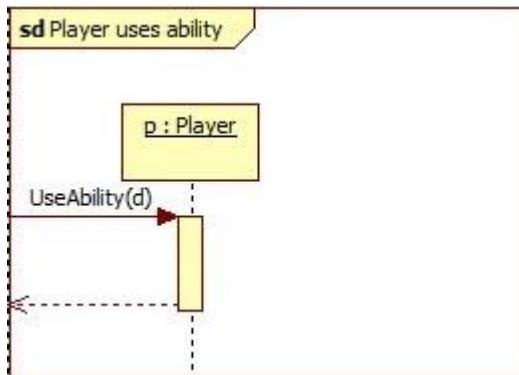
3.4.3 Next Player's turn



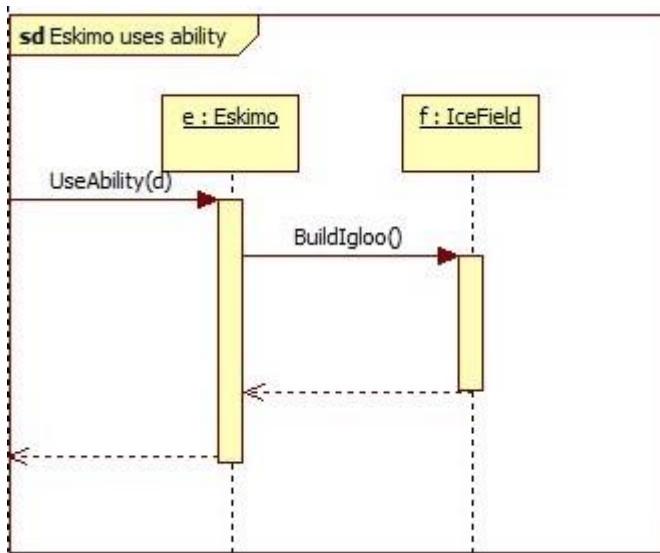
3.4.4 Round Over



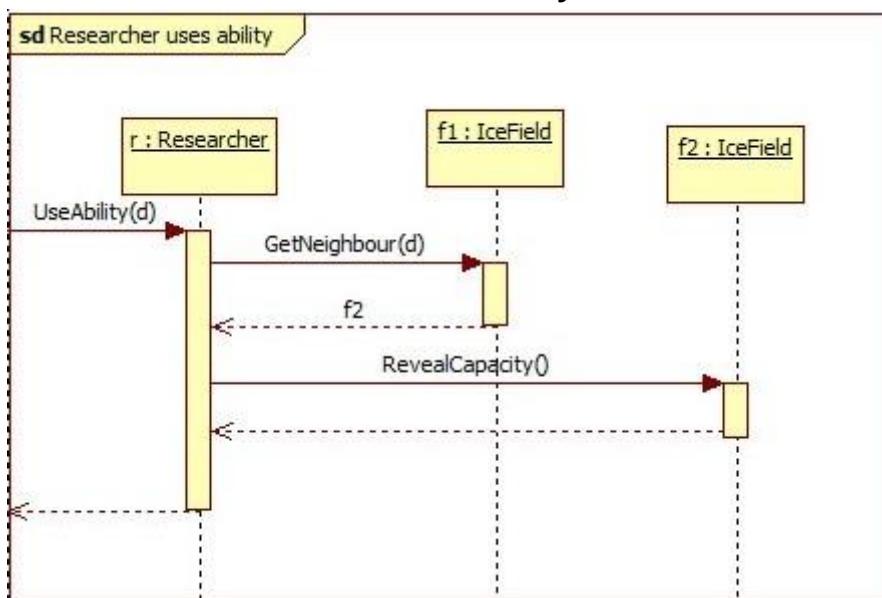
3.4.5 Player Uses Ability



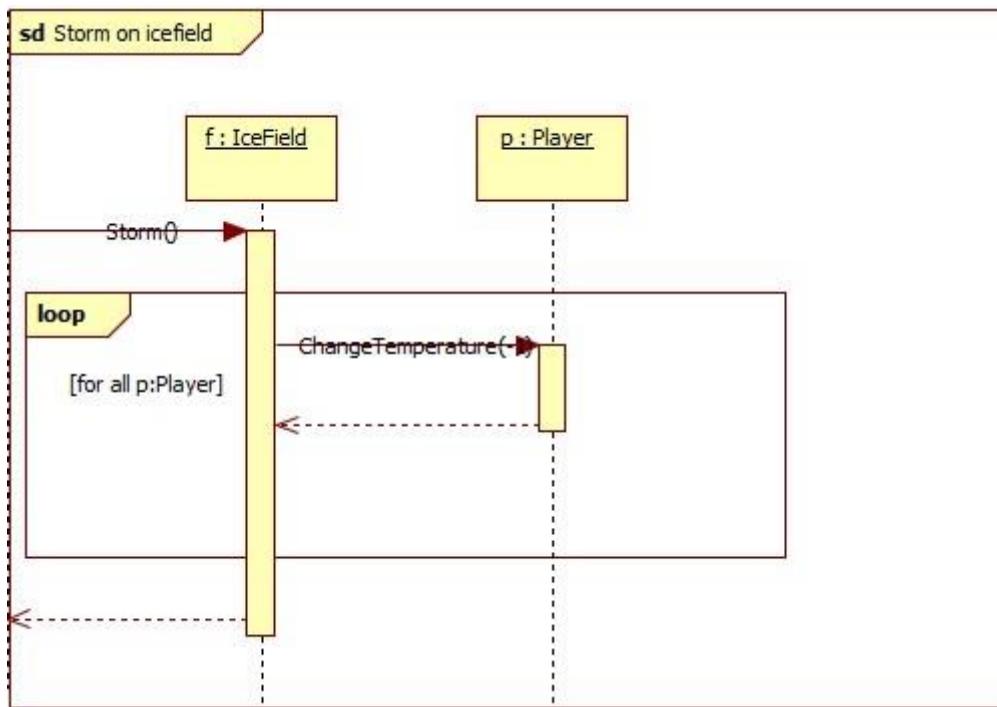
3.4.6 Eskimo Uses Ability



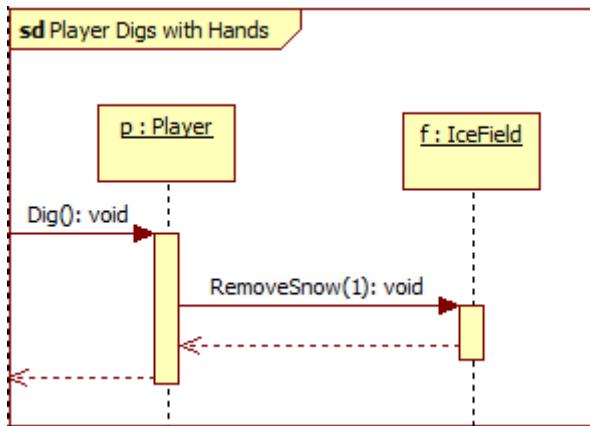
3.4.7 Researcher Uses Ability



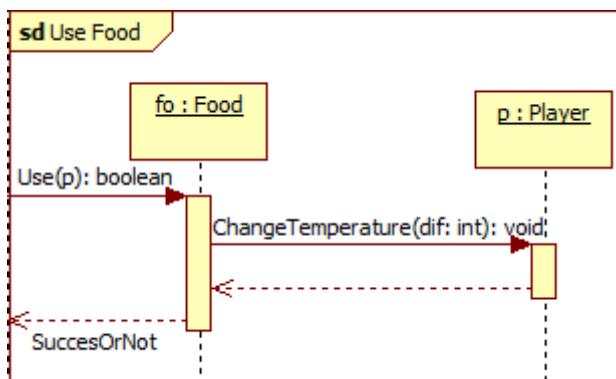
3.4.8 Storm On Icefield



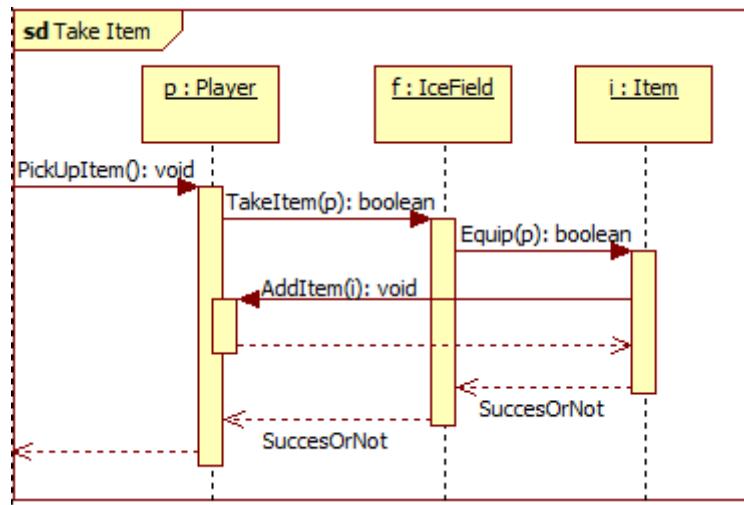
3.4.9 Player Digs with Hands



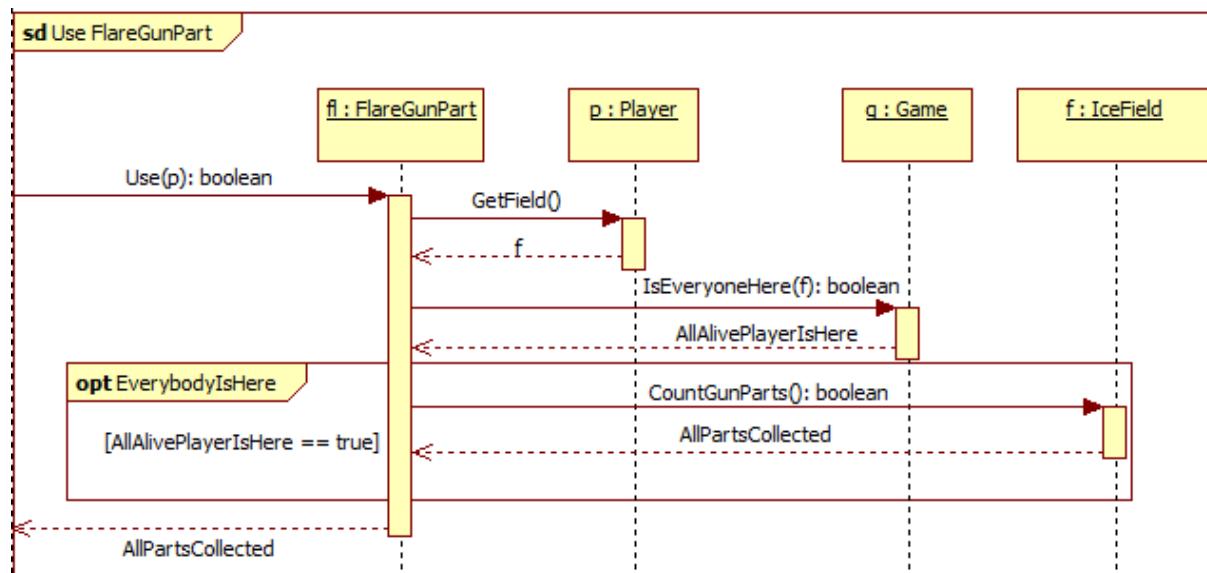
3.4.10 Use Food



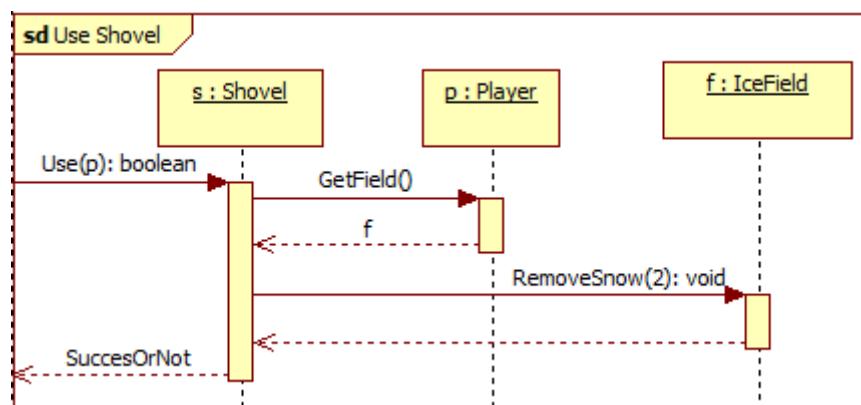
3.4.11 Take Item



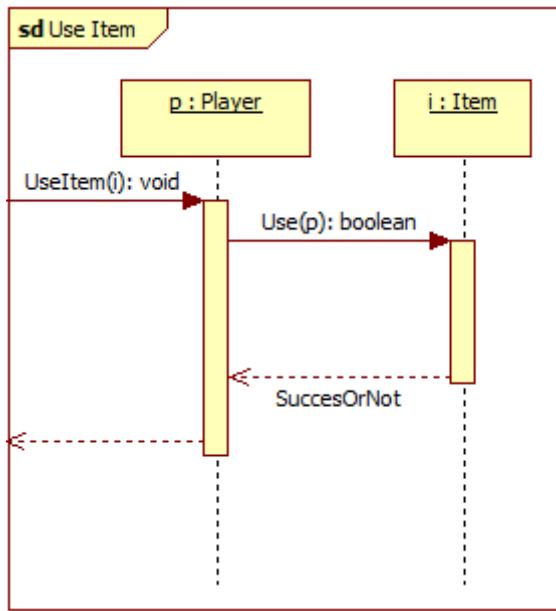
3.4.12 Use Flare Gun Part



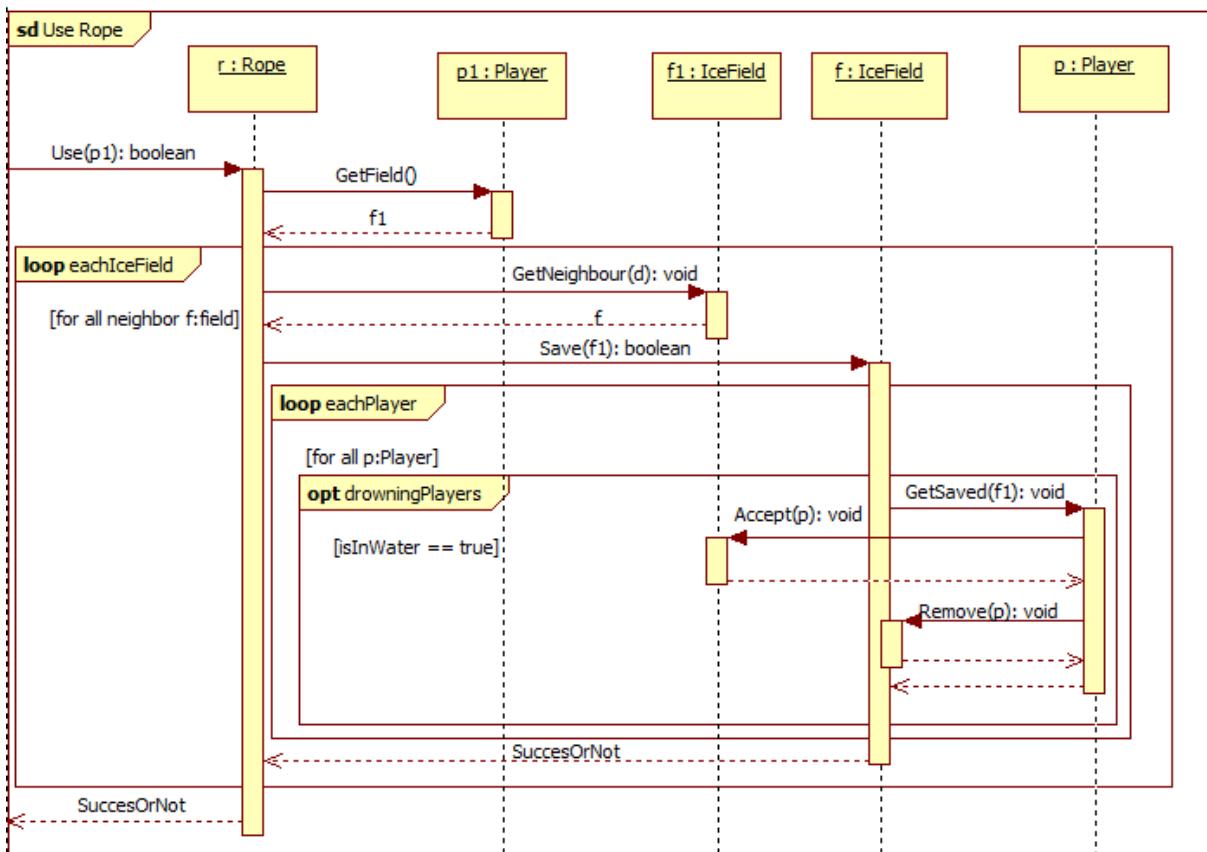
3.4.13 Use Shovel



3.4.14 Use Item



3.4.15 Use Rope



3.5 State-chartok

Állapotgépre nincs szükség a programterv bemutatásához.

3.6 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.02.24. 13:00	6 óra	Karmacsi	Objektum katalógus, osztálydiagram. 3.1 3.2
2020.02.24. 13:00	2 óra	Suciú	Osztálydiagram
2020.02.25. 20:00	1 óra	Karmacsi Suciú Seres Takács	Megbeszélés: Osztálydiagram tervezése
2020.02.26. 20:00	2 óra	Seres Bartsch Suciú Takács	Megbeszélés: Osztálydiagram tervezése
2020.02.27. 21:00	1 óra	Seres Karmacsi Suciú Takács	Megbeszélés: Osztálydiagram tervezése
2020.02.28. 20:00	2 óra	Karmacsi Suciú Bartsch Seres Takács	Megbeszélés: Szekvenciadiagramok vázlatos megtervezése feladatok beosztása.
2020.02.29. 09:00	4 óra	Karmacsi	Osztályok leírása. 3.3
2020.02.29. 11:00	4 óra	Takács	Szekvenciadiagramok
2020.03.01. 11:00	4 óra	Seres	Szekvenciadiagramok
2020.03.01. 16:10	1 óra	Takács	Szekvenciadiagramok
2020.03.01. 20:00	2 óra	Karmacsi Suciú	Dokumentum vélegesítése, nyomtatás

4. Analízis modell kidolgozása

4.1 Objektum katalógus

4.1.1 Sarkkutató

Az egyik játszható karaktertípus. A játék elején 4 egységnnyi testhővel kezd és meg tudja nézni, hogy egy szomszédos jégtábla hány embert bír el.

4.1.2 Eszkimó

Az egyik játszható karaktertípus. A játék elején 5 egységnnyi testhővel kezd és képes iglut építeni.

4.1.3 Jégtábla

Ezen állhatnak a játékosok, de ha túl sokan állnak rajta egyszerre, akkor felborul és mindenki vízbeesik, aki rajta állt. Lehetnek tárgyak belefagyva, amiket a jégtáblát fedő höréteg eltávolítása után lehet kiásni.

4.1.4 Lapát

Egyike a jégből kiátható tárgyaknak. Ha a játékosnak van ásója, akkor egyszerre 2 egységnnyi havat takaríthat el egy munkaráfordítással.

4.1.5 Kötél

Egyike a jégből kiátható tárgyaknak. A játékosnak kötélre van szüksége ahhoz, hogy kimenekítsen egy másik, vízbe esett játékoszt.

4.1.6 Búvárruha

Egyike a jégből kiátható tárgyaknak. Ha egy vízbeesett játékosnak van búvárruhája, akkor nem kell másnak kimenekítenie és folytathatja a körét, mintha semmi se történt volna.

4.1.7 Élelem

Egyike a jégből kiátható tárgyaknak. A felhasználásával a játékosok növelhetik a testhőjüköt egygyel.

4.1.8 Iglu

Az eszkimók tudnak iglukat építeni. Az igluban el lehet bújni a vihar elől, így elkerülve a vihar miatti testhő csökkenést.

4.1.9 Hóvihar

Időnként feltámad egy hóvihar, ami néhány jégtáblát újabb réteg hóval borít be. Akit elkap a hóvihar, annak egységnnyivel csökken a testhője.

4.1.10 Jelzőrakéta

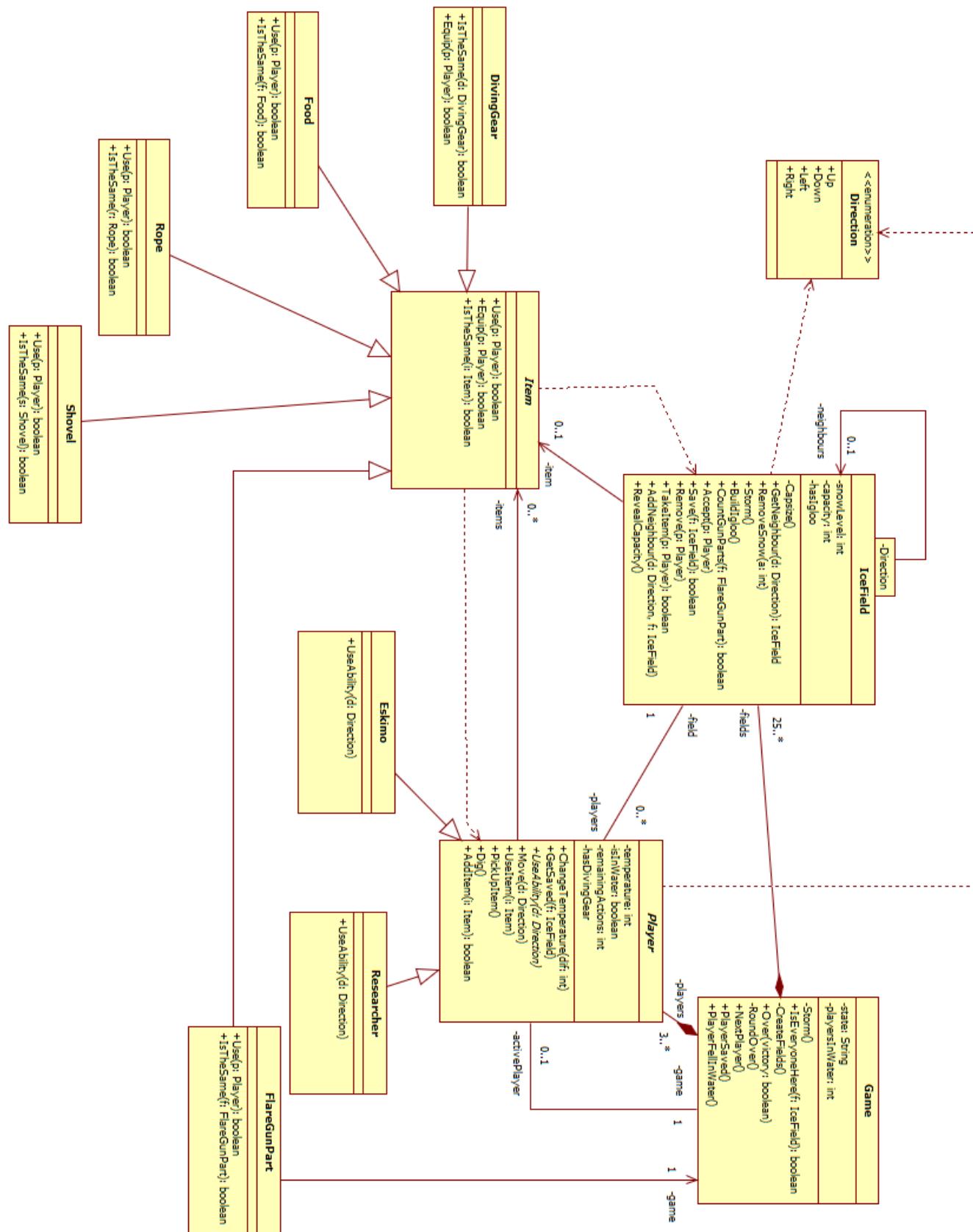
A játékot ennek az elszűrésével lehet megnyerni, amikor minden játékos egy jégtáblán áll.

4.1.11 Jelzőfény, pisztoly, patron

Ezekre van szükség a jelzőrakéta összerakásához. Ezek a játék kezdetekor jégbe vannak fagyva és ugyanúgy lehet kiásni őket, mint a többi tárgyat.

4.2 Statikus struktúra diagramok

(Megjegyzés: a nevesített asszociációvégekhez és attribútumokhoz implicit getter és setter függvények tartoznak, amelyeket a diagram az olvashatóság kedvéért nem jelöl.)



4.3 Osztályok leírása

4.3.1 DivingGear

- **Felelősség**

A játékos tulél a vízben is kör végén, ha van neki, illetve ki tud jutni a vízből.

- **Ősosztályok**

Item → DivingGear

- **Metódusok**

- **boolean IsTheSame(DivingGear: d):** True-val tér vissza.
- **boolean Equip(Player p):** Ugyan azt csinálja, mint az Item, csak pluszba beállítja a játékoson a hasDivingGear-t.

4.3.2 Eskimo

- **Felelősség**

Képes Iglut építeni.

- **Ősosztályok**

Player → Eskimo

- **Metódusok**

- **boolean UseAbility(Direction: d):** Iglut épít oda, ahol éppen áll. True-val tér vissza, ha sikerült.

4.3.3 FlareGunPart

- **Felelősség**

Ezekből kell összegyűlyteni mindenárrat és használni egy helyen állva a játék megnyeréséhez.

- **Ősosztályok**

Item → FlareGunPart

- **Metódusok**

- **boolean Use(Player p):** Megnézi, hogy teljesülnek-e a győzelem feltételei és jelez a Game-nek, ha igen. True-val tér vissza, ha sikerült használni.
- **boolean IsTheSame(FlareGunPart: f):** True-val tér vissza.

4.3.4 Food

- **Felelősség**

Fel lehet használni a játékos testhőjének növelésére.

- **Ősosztályok**

Item → Food

- **Metódusok**

- **boolean Use(Player p):** A játékos testhőjét növeli. True-val tér vissza.

4.3.5 Game

- **Felelősség**

Inicializálja a játékot. Nyilvántartja a játék állapotát. minden kör végén vihart indít és ellenőrzi a játékosok állapotát.

- **Attribútumok**

- **-state:** String: A játék állapota (folyamatban van, vége + nyert/vesztett)
- **-fields:** IceField[]: A jégtáblák.
- **-players:** Player[]: A játékosok.
- **-activePlayer:** player: A soron levő játékos.
- **-playersInWater:** int: A vízbeesett játékosok száma.

- **Metódusok**

- **boolean IsEveryoneHere(IceField f):** Megnézi, hogy mindenki az átvett fielden áll e.
- **void Over(boolean victory):** Befejezi a játékot. Ha a paraméter true, akkor a játékosok nyertek.
- **void NextPlayer():** Átállítja, hogy ki az aktuális jázékos és átállítja a hátralevő akcióit 4-re. Ha vége egy körnek, akkor meghívja a RoundOver()-t.
- **void PlayerSaved():** Jelzés, hogy egy játékos kimentettek a vízből.
- **void PlayerFellInWater():** Jelzés, hogy egy jétkos vízbe esett.

4.3.6 IceField

- **Felelősség**

Tárolja a jégtábla adatait(szomszédok, rajta álló játékosok, hőszint, befagyott tárgy, iglu). Lehet mozogni közöttük, felborul, ha sokan állnak rajta, ki lehet ásni a tárgyat, ha nincs rajta hó. (A hóval fedett lyuk egy 0 kapacitású jégtábla.)

- **Attribútumok**

- **-neighbours:** IceField: A szomszédos jégtáblák a megadott irányba.
- **-snowLevel:** int: A jégtáblán levő hőréteg vastagsága.
- **-players:** Player[]: A rajta álló játékosok.
- **-item:** Item: A jégtáblába fagyott tárgy.
- **-capacity:** int: Ennyi játékos fér el rajta anélkül, hogy felborulna.
- **-hasIgloo:** True, ha van rajta iglu.

- **Metódusok**

- **IceField GetNeighbour(Direction d):** Visszaadja a szomszédos jégtáblát a megadott irányba.
- **void removeSnow(int a):** Csökkenti a hőréteg vastagságát a megadott értékkal.
- **void Storm():** Növeli a hőréteget és csökkenti a rajtalevő játékosok testhőjét, ha nincs a jégtáblán iglu.
- **void BuildIgloo():** Iglut épít a jégtáblára

- **boolean CountGunParts(FlareGunPart: f):** Megszámolja, hogy hány játékosnak van itt alkatrésze a jelzőpisztolyhoz. True-val tér vissza, ha az összes alkatrész megvan.
- **void Accept(Player p):** Felveszi a játékost a rajta álló játékosok közé. Felborul ha, túl sokan állnak így már rajta.
- **boolean Save(IceField f):** Végighívja a GetSaved(Icefield f)-et a rajta álló játékosokon. True-val tér vissza, ha legalább 1 játékos ki lett mentve.
- **void Remove(Player p):** Kiveszi a játékost a rajta állók listájából.
- **boolean TakeItem(Player p):** A kapott játékosnak kiássa a befagyott tárgyat. True-val tér vissza, ha sikerült.
- **void AddNeighbour(Direction d, IceField f):** Beállítja kapott irányba levő szomszédnak a kapott IceFieldet.
- **void RevealCapacity():** Felfedi, hogy hányan férnek el a jégtáblán anélkül, hogy felborulna. Innentől kezdve végig látszik.
- **boolean IsTheSame(Food: f):** True-val tér vissza.

4.3.7 Item

- **Felelősség**

A használható tárgyak heterogén kollekciója.

- **Attribútumok**

- **-name:** string: A tárgy neve.(hogy azonosítani lehessen)

- **Metódusok**

- **boolean Use(Player p):** A kapott játékos használja a tárgyat. True-val tér vissza, ha sikerült.
- **boolean Equip(Player p):** A kapott player felveszi a tárgyat. Ha sikerül True-val tér vissza.
- **boolean IsTheSame(Item: i):** False-al tér vissza. A leszármazottaknak van egy ugyan ilyen nevű függvényük, amiben magukból vesznek át egy példányt paraméterként és azok true-val térnek vissza. Így meg lehet nézni, hogy a kapott item ugyanolyan fajta-e, mint ez.

4.3.8 Player

- **Felelősség**

Nyilvántartja a játékos adatait. A játékos akciót végrehajtja. Ha elfogynak az akciói, szól a Game-nek, hogy jöhet a következő.

- **Attribútumok**

- **-field:** IceField: A jégtábla, amin éppen áll
- **-temperature:** int: A játékos jelenlegi testhője
- **-isInWater:** boolean: True, ha vízbe van esve.
- **-remainingActions:** int: A hátralévő akciók száma a játékos körében.
- **-items:** Item[]: A játékos birtokolt tárgyai.
- **-game:** Game: A játék.
- **-hasDivingGear:** True, ha van búvárruhája.

- **Metódusok**

- **void ChangeTemperature(int dif):** Változtatja a játékos testhőjét a megadott értékkel(nem lehet 0 alá)
- **boolean GetSaved(IceField: f):** Kimenekítik a vízből a kapott IceField-re, ha vízbe van esve. True-val tér vissza, ha ki lett mentve.
- **void UseAbility(Direction: d):** Használja a játékos képességét a megadott irányba. Ha null-t vesz át, akkor ott, ahol áll a játékos. Levon egy akciót, ha sikerült.
- **void Move(Direction: d):** Másik jégtáblára mozog a megadott irányba. Levon egy akciót, ha sikerült.
- **void UseItem(Item i):** Használja a megadott tárgyat a játékos. **void PickUpItem():** Kiássa a jégből a belefagyott tárgyat. Levon egy akciót, ha sikerült.
- **void Dig():** Eltakarít egy réteg havat a jégmezőről, ahol áll. **void AddItem(Item i):** Hozzáadja az Item-et a játékos tárgyaihoz. Levon egy akciót, ha sikerült.
- **void PickUpItem():** Kiássa a jégtáblából a befagyott tárgyat.
- **boolean AddItem(Item i):** A játékos táskájába teszi a tárgyat, ha még nincs neki egy ilyen. True-val tér vissza, ha sikerült.

4.3.9 Researcher

- **Felelősség**

Meg tudja nézni egy IceField kapacitását.

- **Ősosztályok**

Player → Researcher

- **Metódusok**

- **boolean UseAbility(Direction: d):** Meg tudja nézni egy szomszédos(vagy ahol áll, ha d null) IceField kapacitását. True-val tér vissza.

4.3.10 Rope

- **Felelősség**

Ki lehet menteni a játékosokat a vízből a környező mezőkről.

- **Ősosztályok**

Item → Rope

- **Metódusok**

- **boolean Use(Player p):** Kimenti a vízbeesett játékosokat a szomszédos mezőkről. True-val tér vissza, ha legalább egy játékest kimentett.
- **boolean IsTheSame(Rope: r):** True-val tér vissza.

4.3.11 Shovel

- **Felelősség**

Havat lehet vele eltakarítani a jégmezőről.

- **Ősosztályok**

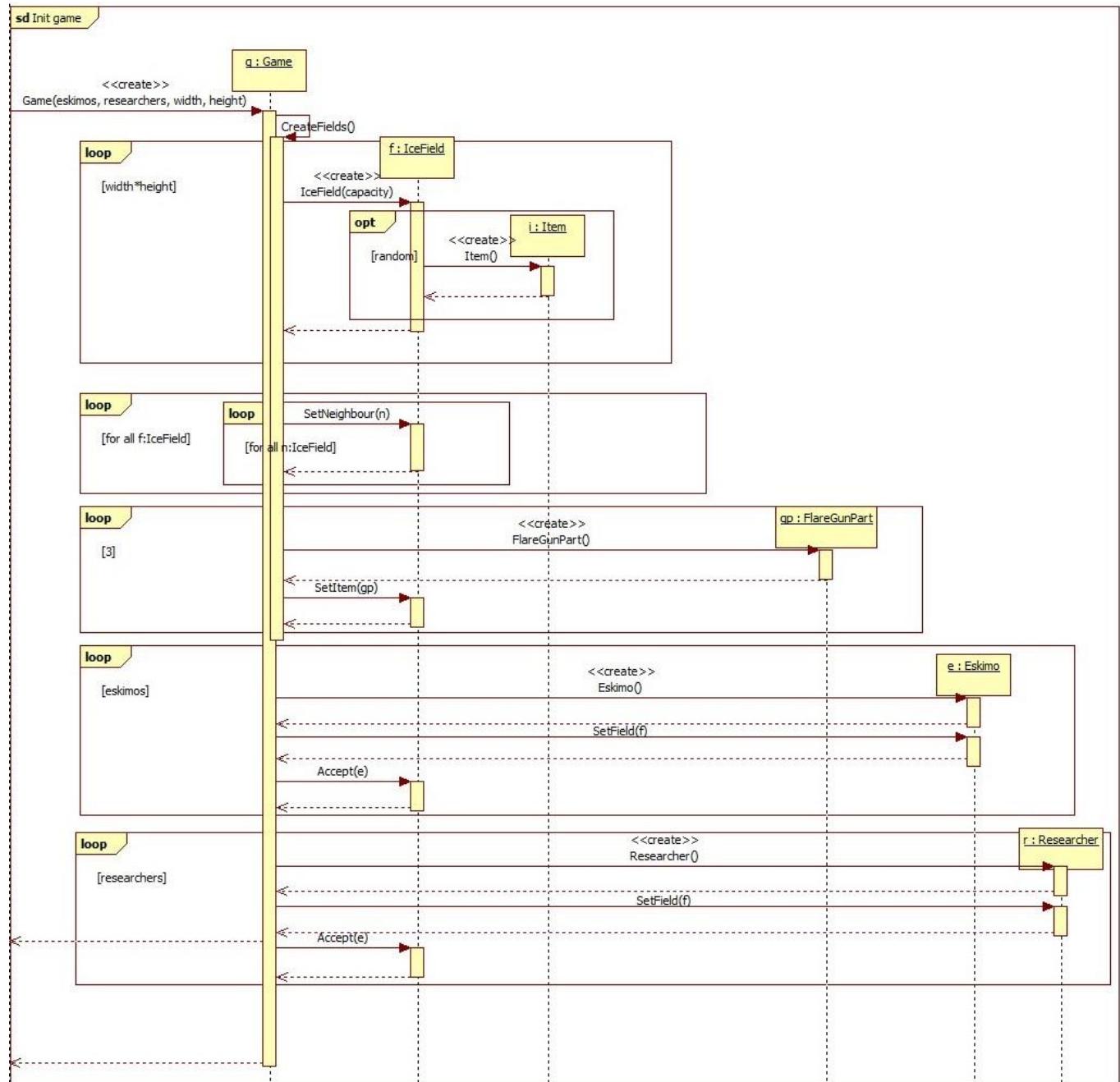
Item → Food

- **Metódusok**

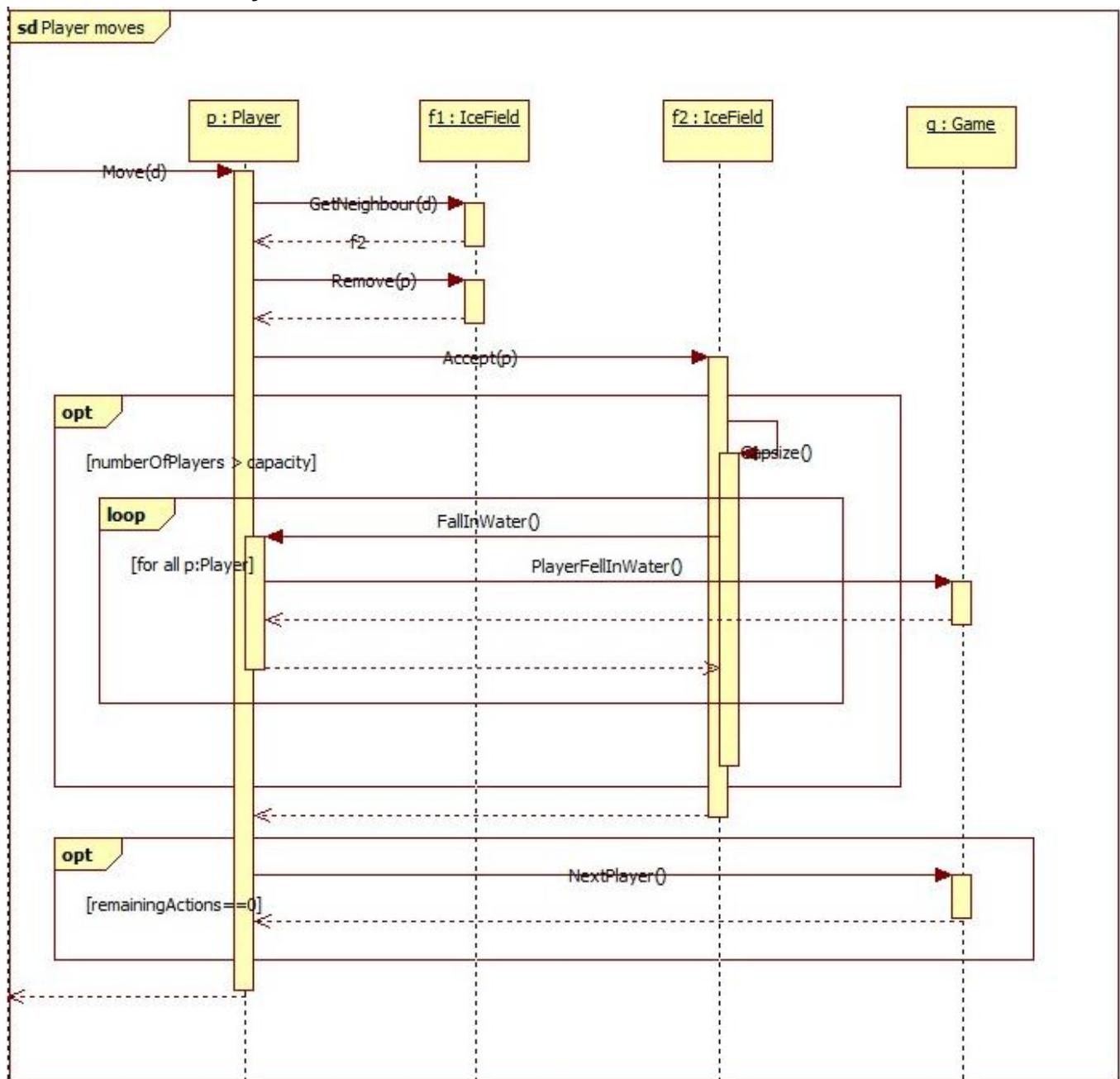
- **boolean Use(Player p):** Két egységnyi höréteget takarít el a jégmezőről, ahol a használója áll.
- **boolean IsTheSame(Shovel: s):** True-val tér vissza.

4.4 Szekvencia diagramok

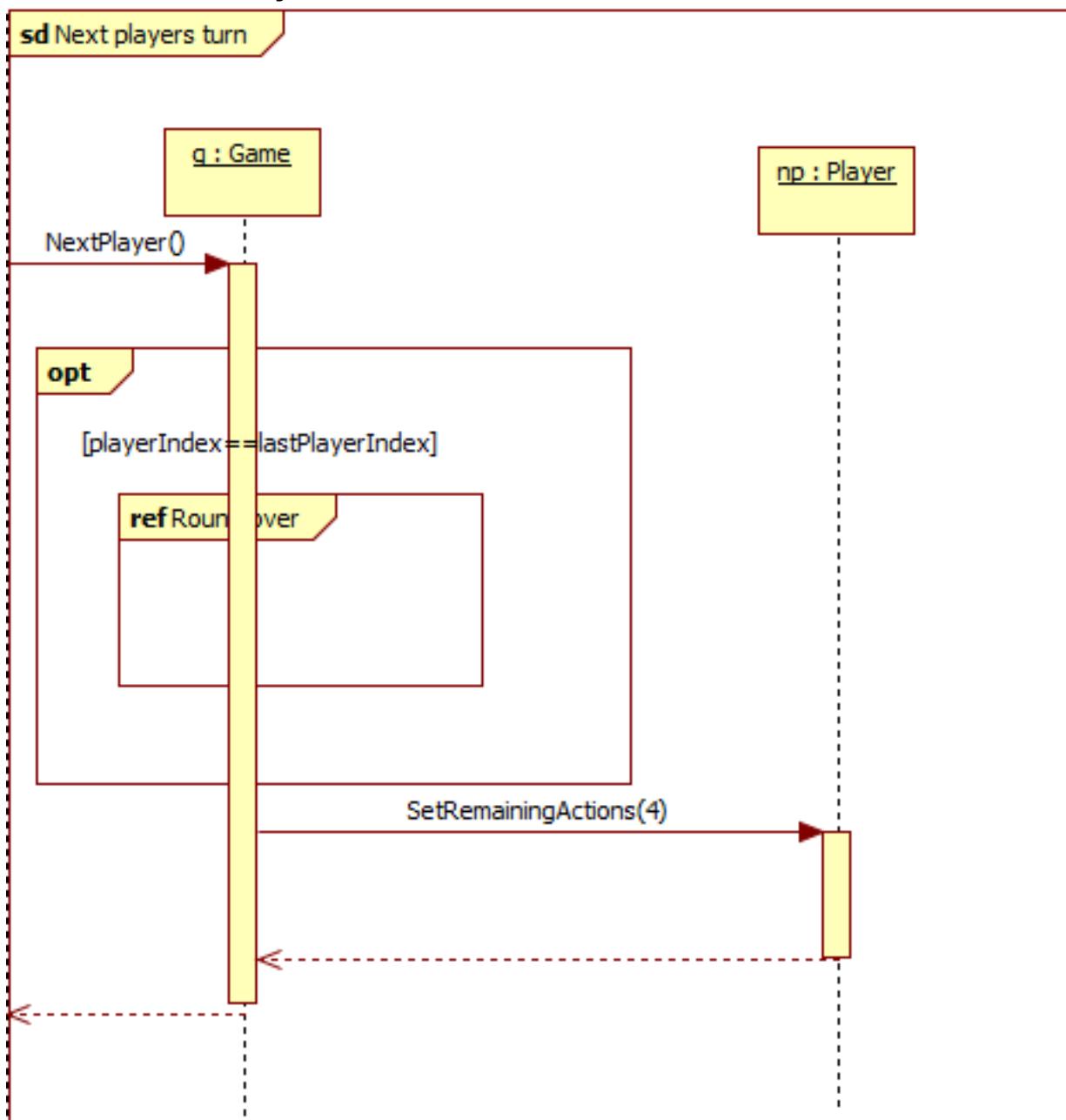
4.4.1 Init Game



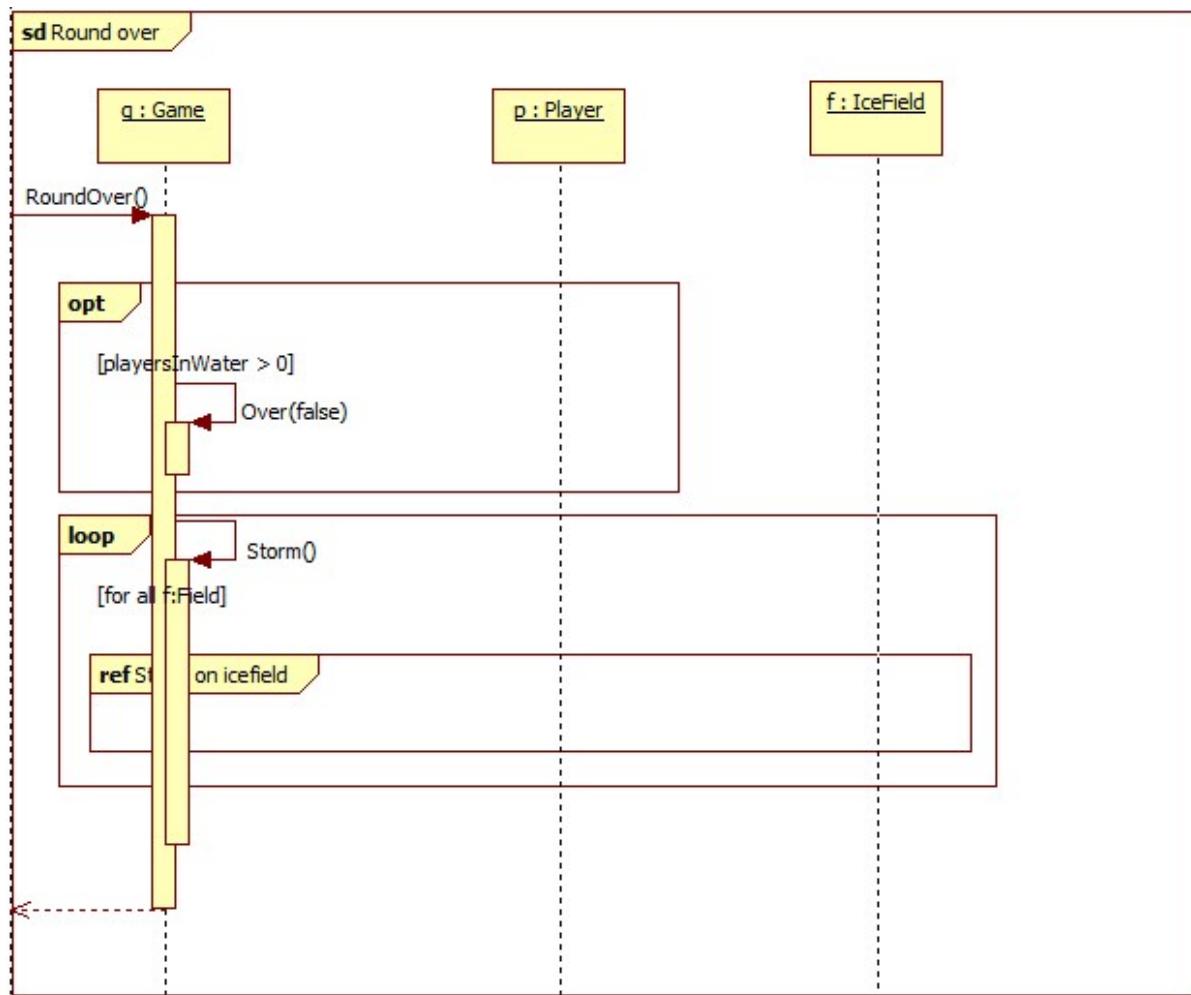
4.4.2 Player Moves



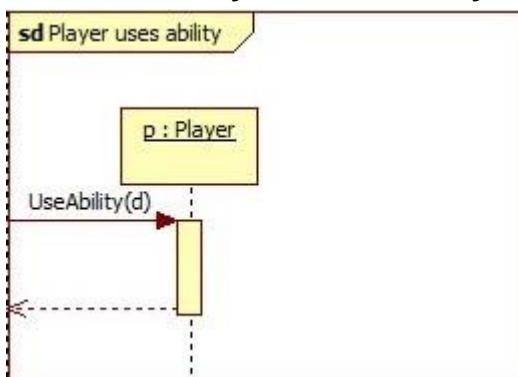
4.4.3 Next Player's turn



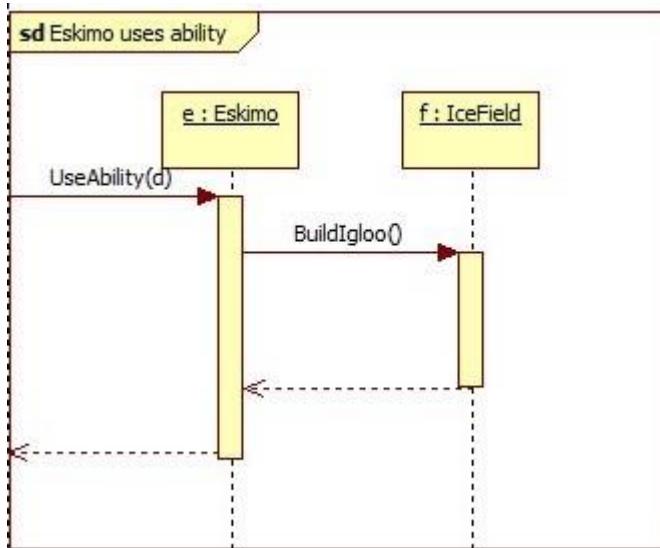
4.4.4 Round Over



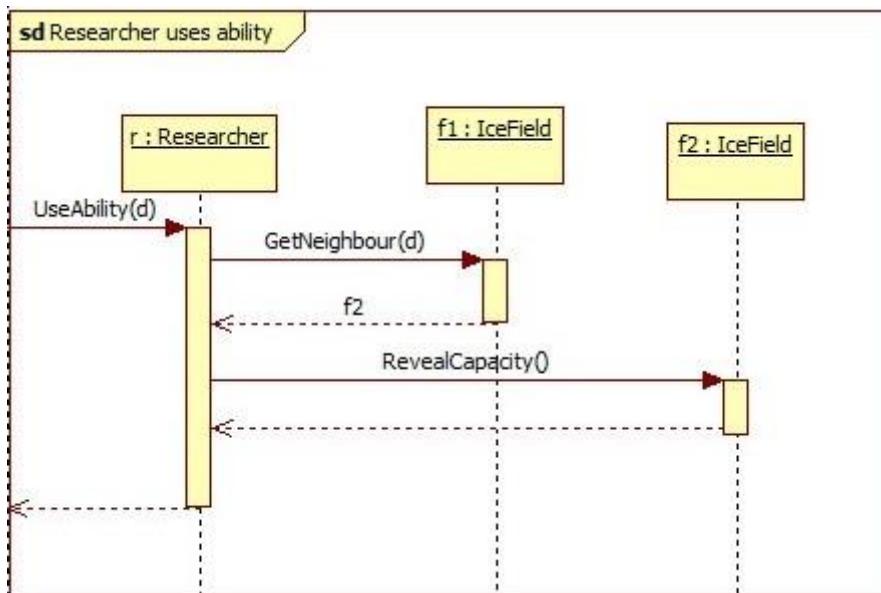
4.4.5 Player Uses Ability



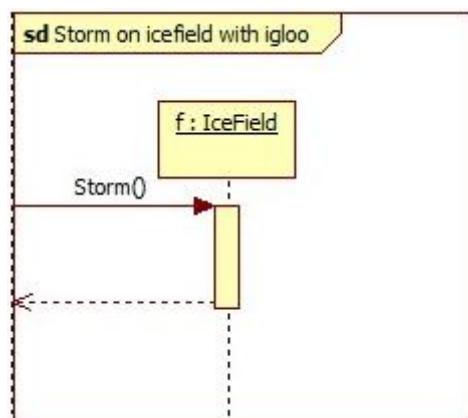
4.4.6 Eskimo Uses Ability



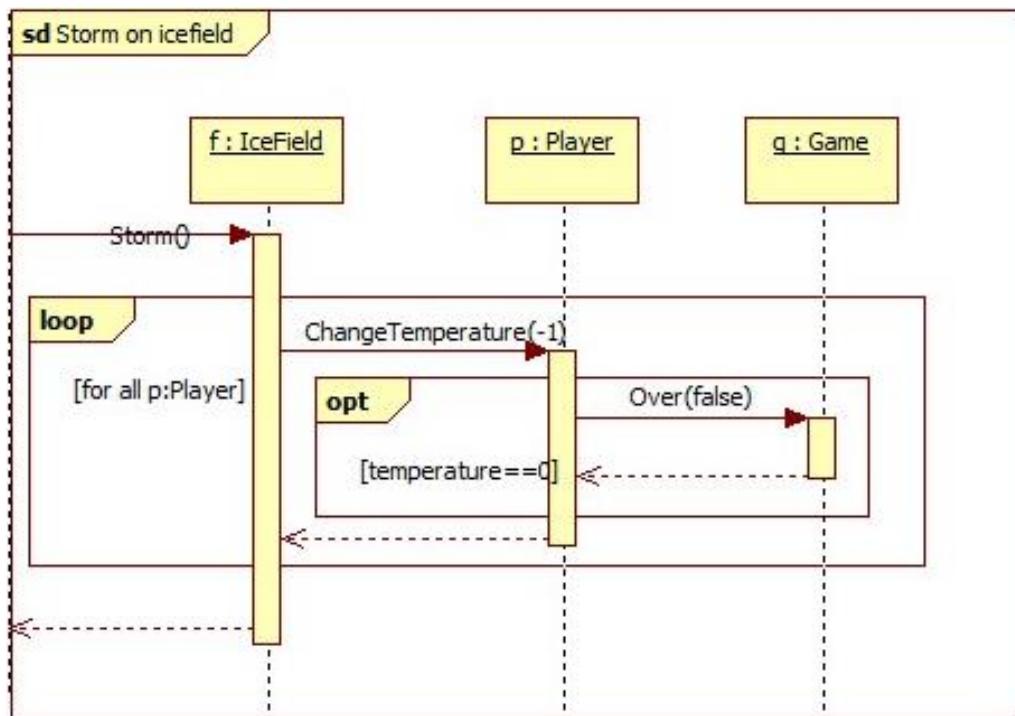
4.4.7 Researcher Uses Ability



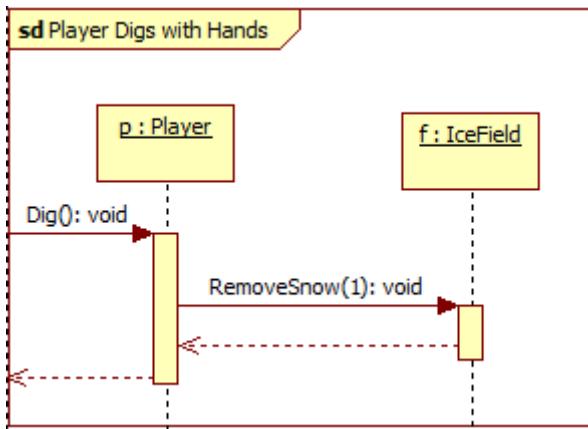
4.4.8 Storm on Icefield with Igloo



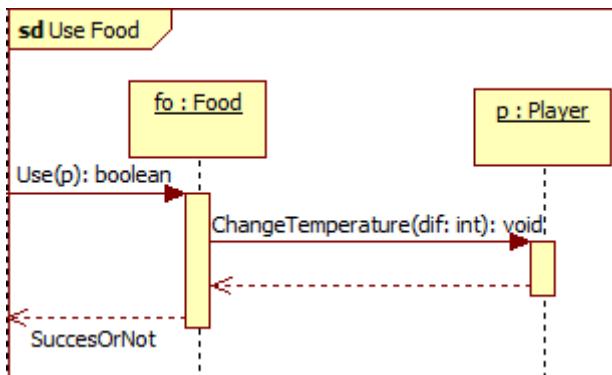
4.4.9 Storm On Icefield



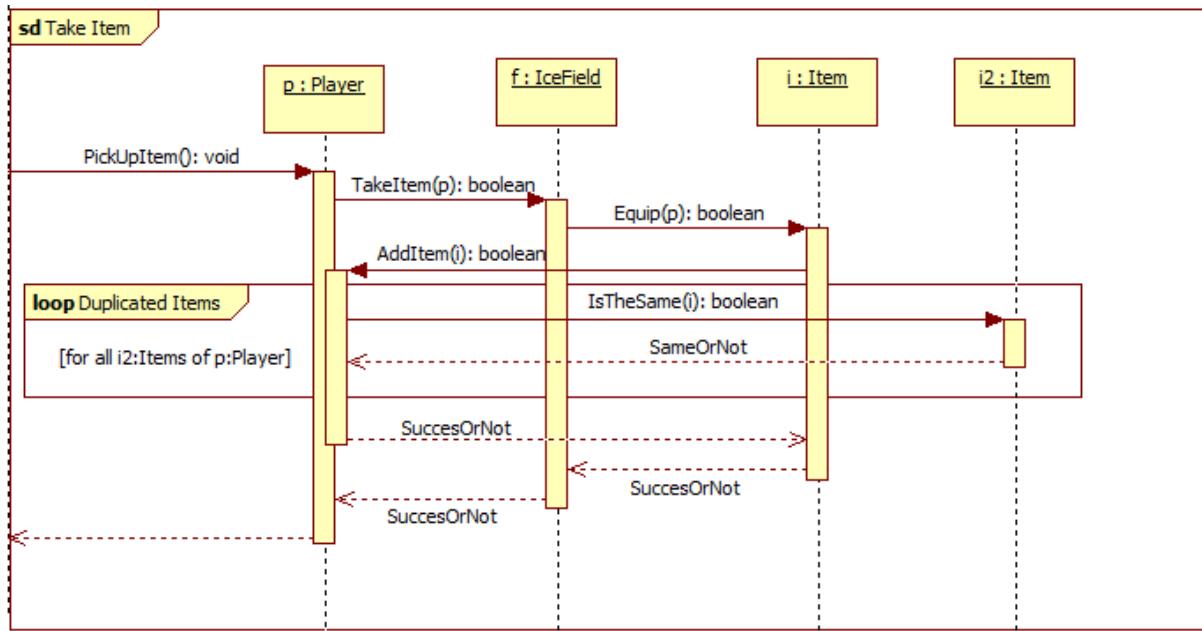
4.4.10 Player Digs with Hands



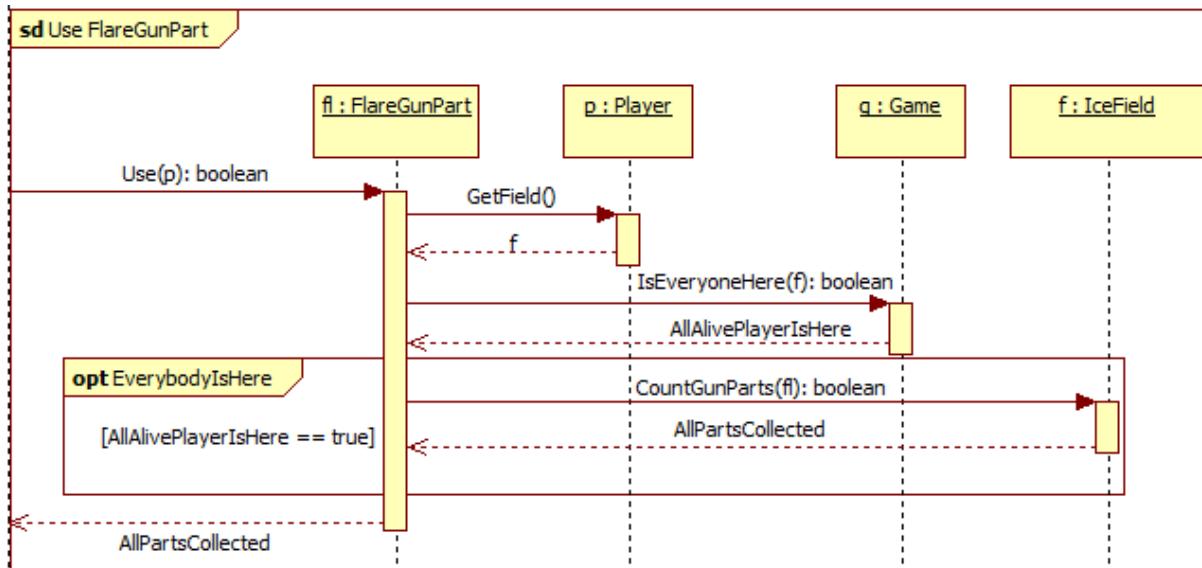
4.4.11 Use Food



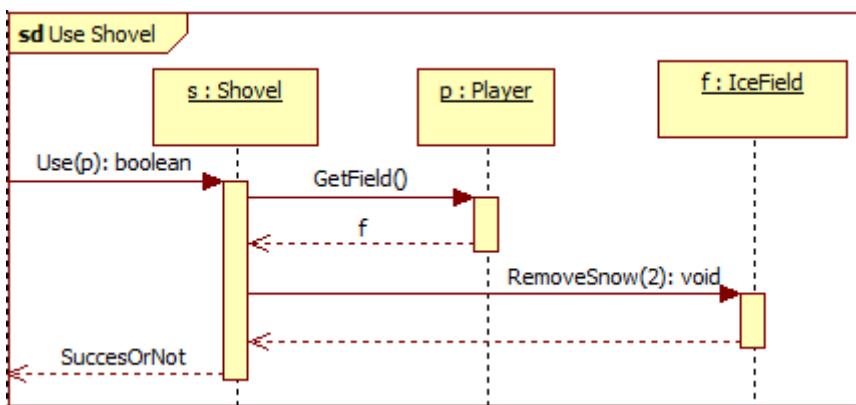
4.4.12 Take Item



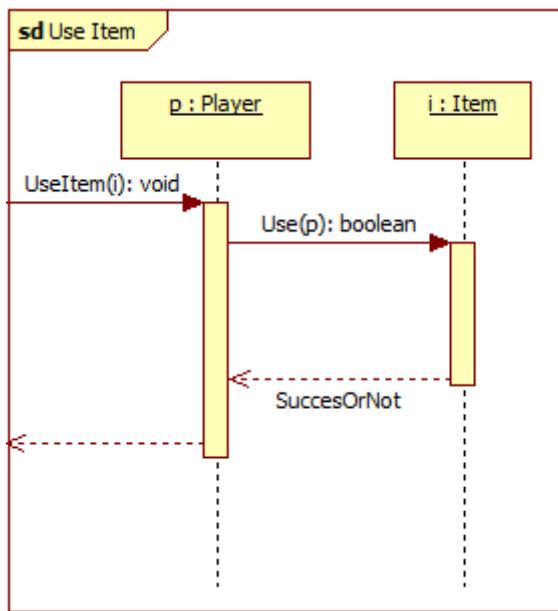
4.4.13 Use Flare Gun Part



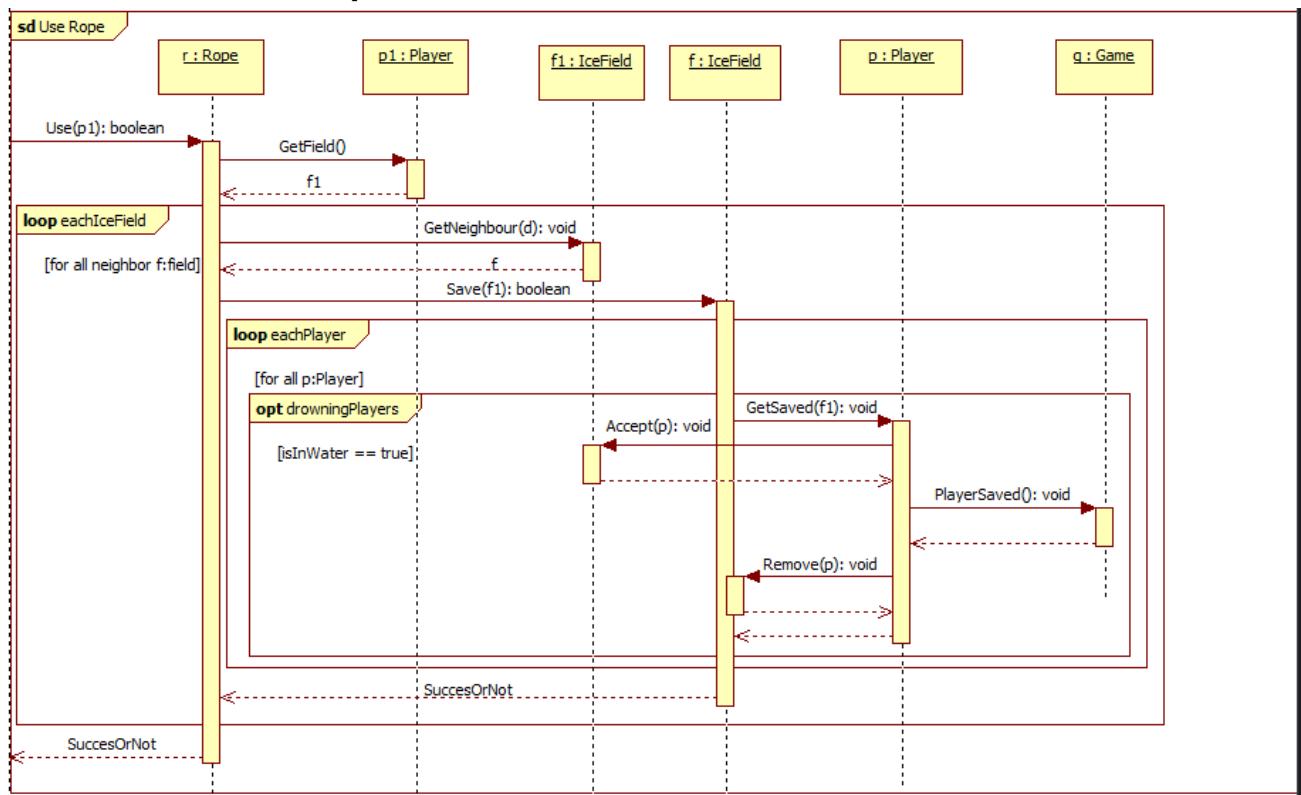
4.4.14 Use Shovel



4.4.15 Use Item



4.4.16 Use Rope



4.5 State-chartok

Állapotgépre nincs szükség a programterv bemutatásához.

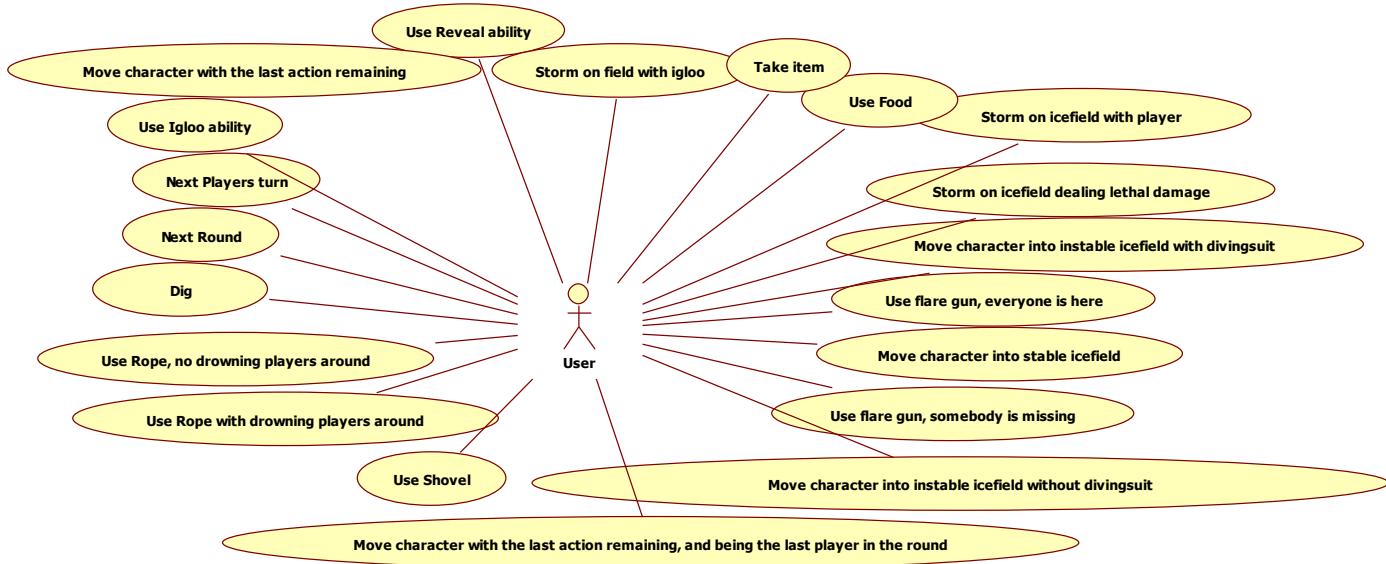
4.6 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.03.07. 16:00	1 óra	Karmacsi Suciu Seres Takács	Megbeszélés: Változtatások egyeztetése.
2020.03.07. 16:30	1 óra	Seres	Szekvenciadiagramokon változtatások eszközlése: 3.4.11., 3.4.12., 3.4.15
2020.03.07. 16:30	1 óra	Karmacsi	Osztálydiagramon és leíráson változtatások: 3.2, 3.3
2020.03.07. 10:30	1 óra	Takács	Szekvenciadiagramokon változtatások eszközlése: 3.4.2., 3.4.4., 3.4.8
2020.03.09. 11::00	1 óra	Suciu	Dokumentum vélegesítése, nyomtatás

5. Szkeleton tervezése

5.1 A szkeleton modell valóságos use-case-ai

5.1.1 Use-case diagram



5.1.2 Use-case leírások

Use-case neve	Next Players turn
Rövid leírás	Egy játékos kifogy az akcióiból, és a következő játékos jön
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a függvényhívásokat, majd kiírja az aktív játékos sorszámának a változását (+1)

Use-case neve	Next Round
Rövid leírás	Egy játékos kifogy az akcióiból, és mivel ő volt az utolsó játékos, a következő kör kezdődik
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a függvényhívásokat, majd kiírja az aktív játékos sorszámának a változását (1)

Use-case neve	Move character into instable icefield without divingsuit
Rövid leírás	Egy karakter rásétál egy mezőre, ahol a vízbe esik, és véget ér a köre.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a lépéssel és az eséssel járó függvényhívásokat

Use-case neve	Move character into instable icefield with divingsuit
----------------------	---

Rövid leírás	Egy karakter rásétál egy mezőre, ahol biztonságosan a vízbe esik, és nem ér véget a köre.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a lépéssel és az eséssel járó függvényhívásokat

Use-case neve	Move character into stable icefield
Rövid leírás	Egy karakter rásétel egy másik mezőre.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a lépéssel járó függvényhívásokat

Use-case neve	Move character with the last action remaining, and being the last player in the round
Rövid leírás	Egy karakter rásétel egy másik mezőre, és mivel ez volt az utolsó akciója, véget ér a köre, és mivel ez volt az utolsó akcióval bíró játékos, ezért véget ér a nagy kör is.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a lépéskor, a kör végekor, és a nagy kör végén lefutó függvényhívásokat

Use-case neve	Move character with the last action remaining
Rövid leírás	Egy karakter rásétel egy másik mezőre, és mivel ez volt az utolsó akciója, véget ér a köre.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a lépéssel és a kör végével járó függvényhívásokat

Use-case neve	Storm on icefield dealing lethal damage
Rövid leírás	A vihar lecsap egy mezőre, ahol karakter tartózkodik, és a vihar megöl ezt a karaktert.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a viharral, a kihűléssel, és a játék végével kapcsolatos függvényhívásokat

Use-case neve	Storm on icefield with players
Rövid leírás	A vihar lecsap egy mezőre, ahol karakter tartózkodik, és a karakter kihűl.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a viharral és a kihűléssel kapcsolatos függvényhívásokat, és az elszenvedő karakter hőváltozását

Use-case neve	Storm on field with igloo
Rövid leírás	A vihar lecsap egy mezőre, ahol igloo van, ezért nem hűl ki senki.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a viharral járó függvényhívásokat

Use-case neve	Use Reveal ability
Rövid leírás	Egy kutató használja a képességét egy szomszédos mezőn
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a képesség használatával járó függvényhívásokat

Use-case neve	Use Igloo ability
Rövid leírás	Egy eszkimó használja a képességét a saját mezőjén
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a képesség használatával járó függvényhívásokat

Use-case neve	Dig
Rövid leírás	Egy karakter kézzel ás.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja az ásással járó függvényhívásokat, és egy IceField hómennyiségeknek változását

Use-case neve	Take Item
Rövid leírás	Egy karakter felvesz egy itemet.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a felvétellel járó függvényhívásokat, és a karakter táskjának tartalmán történő változást

Use-case neve	Use Rope with drowning players around
Rövid leírás	Egy karakter sikeresen használja a kötelét.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a használattal járó függvényhívásokat, és célpont karakter állapotában bekövetkező változást

Use-case neve	Use Rope, no drowning players around
Rövid leírás	Egy karakter sikertelenül használja a kötelét.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a használattal járó függvényhívásokat

Use-case neve	Use Food
Rövid leírás	Egy karakter megesz egy adag éleelmet.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a használattal járó függvényhívásokat, illetve a karakter testhőjének és táskjának a változását

Use-case neve	Use Shovel
----------------------	------------

Rövid leírás	Egy karakter ásásra használja az ásóját.
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a használattal járó függvényhívásokat, és egy IceField hómennyiségek változását

Use-case neve	Use flare gun, everyone is here
Rövid leírás	Egy karakter sikeresen használ egy flare gun darabot
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a használattal járó függvényhívásokat

Use-case neve	Use flare gun, somebody is missing
Rövid leírás	Egy karakter sikertelenül használ egy flare gun darabot
Aktorok	User
Forgatókönyv	1. A program a képernyőre írja a használattal járó függvényhívásokat

5.2 A szkeleton kezelői felületének terve, dialógusok

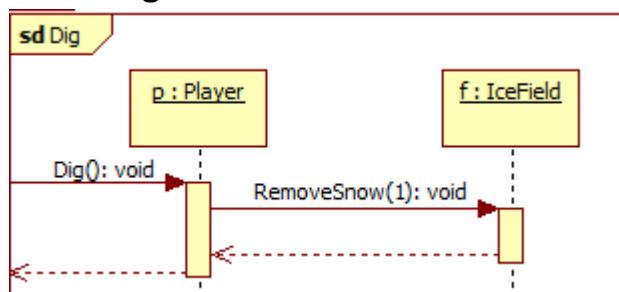
A program indításakor, és a „help” parancs kiadásakor kiíródik a fenti lehetőségek parancsszavai. Egy parancsszó begépelésekor részletes adatokat fogunk látni arra hogy az adott interakció milyen függvényeket hívott, és milyen mezőket módosított volna.

Minden függvény hívásakor a függvény, és a függvényen belüli függvényhívásokban szereplő objektumok típusai is kiírásra kerülnek.

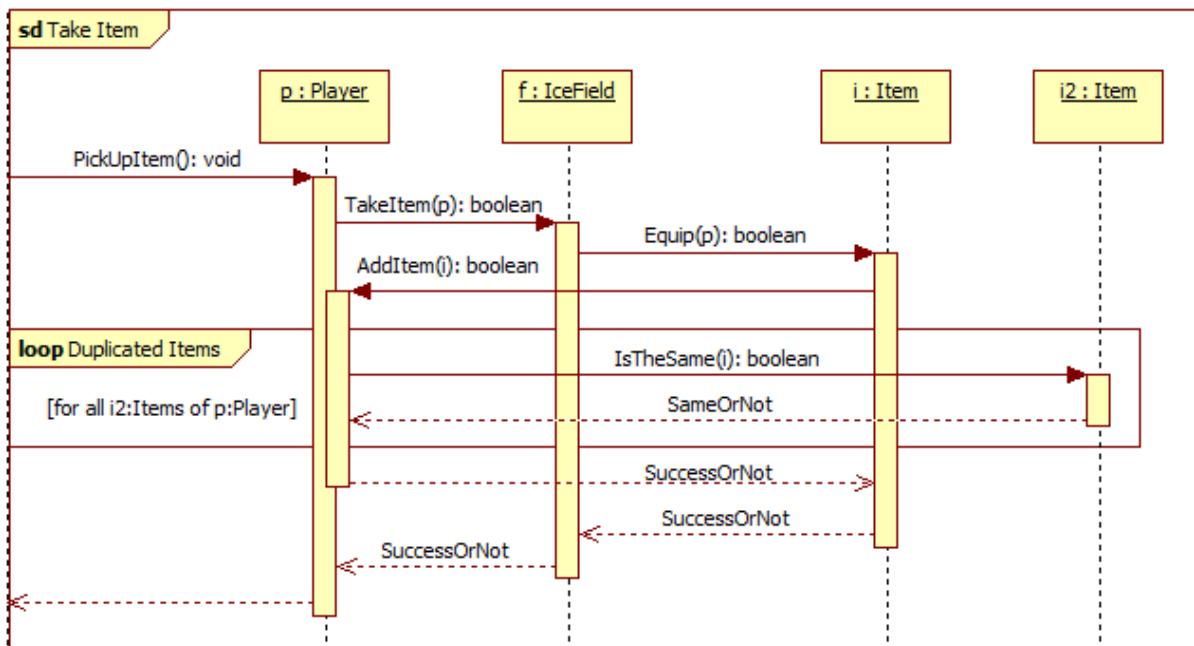
A kimenet tabulálással szándékozik szemléltetni a hívások egymásba egyázottságának szintjeit, de a tulajdonságok kiírása a függvények futása után egy síkban lesz olvasható, az objektumok nevével ellátva.

5.3 Szekvencia diagramok a belső működésre

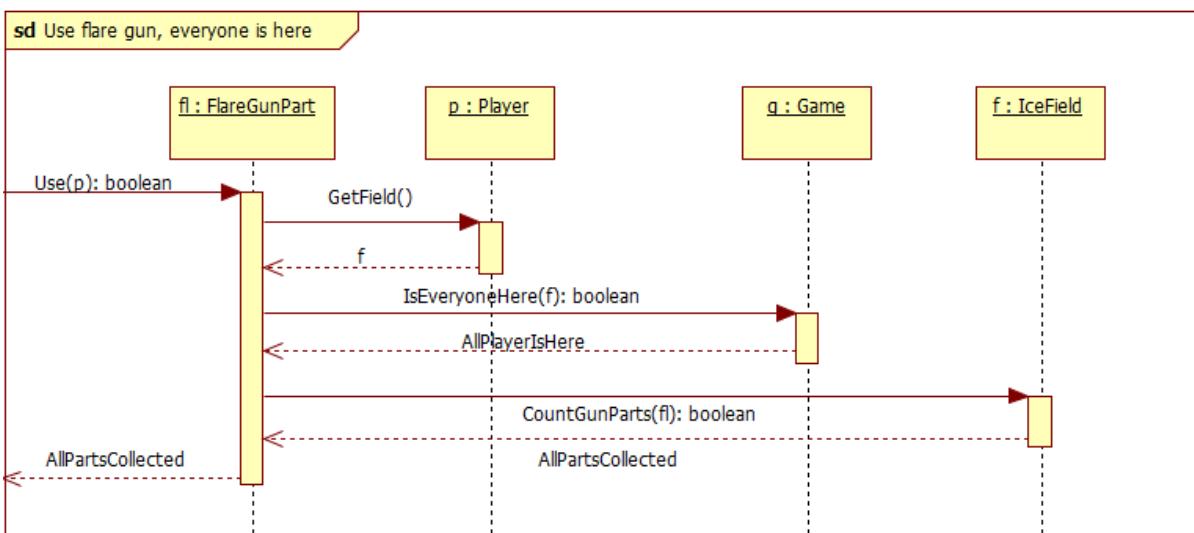
5.3.1 Dig



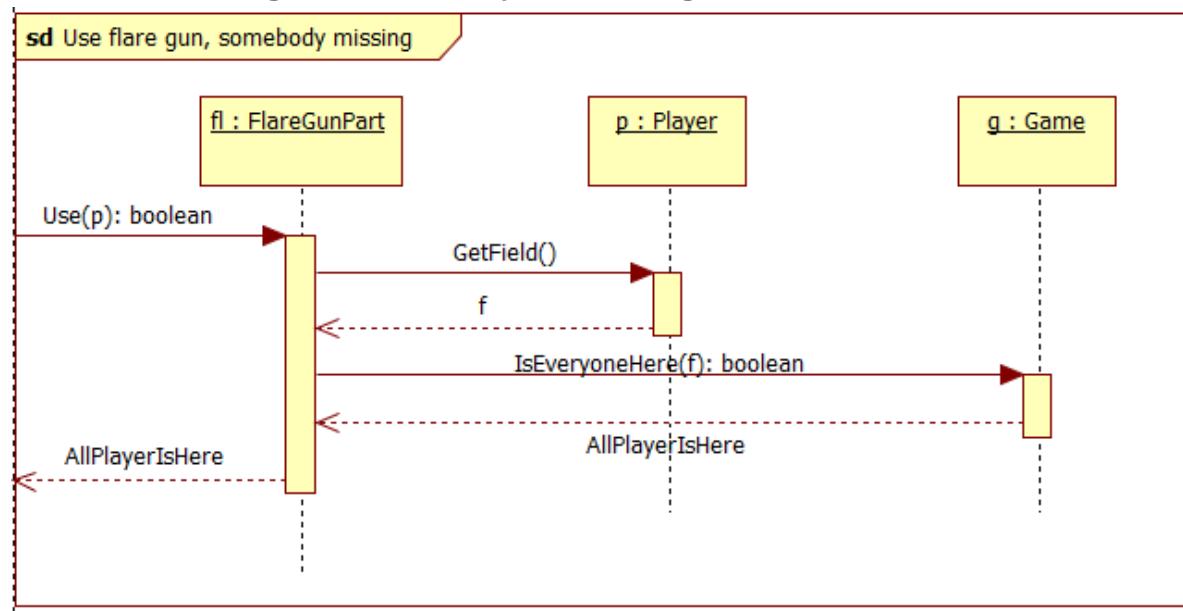
5.3.2 Take Item



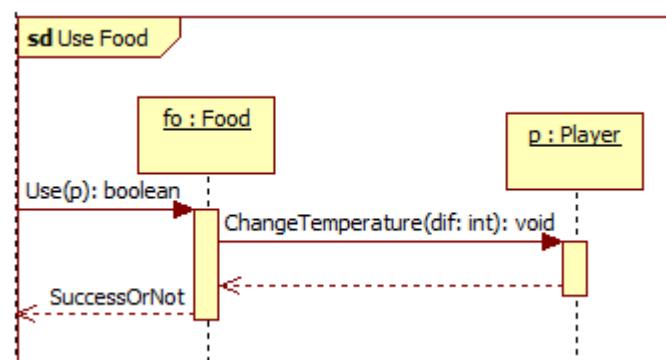
5.3.3 Use flare gun, everyone is here



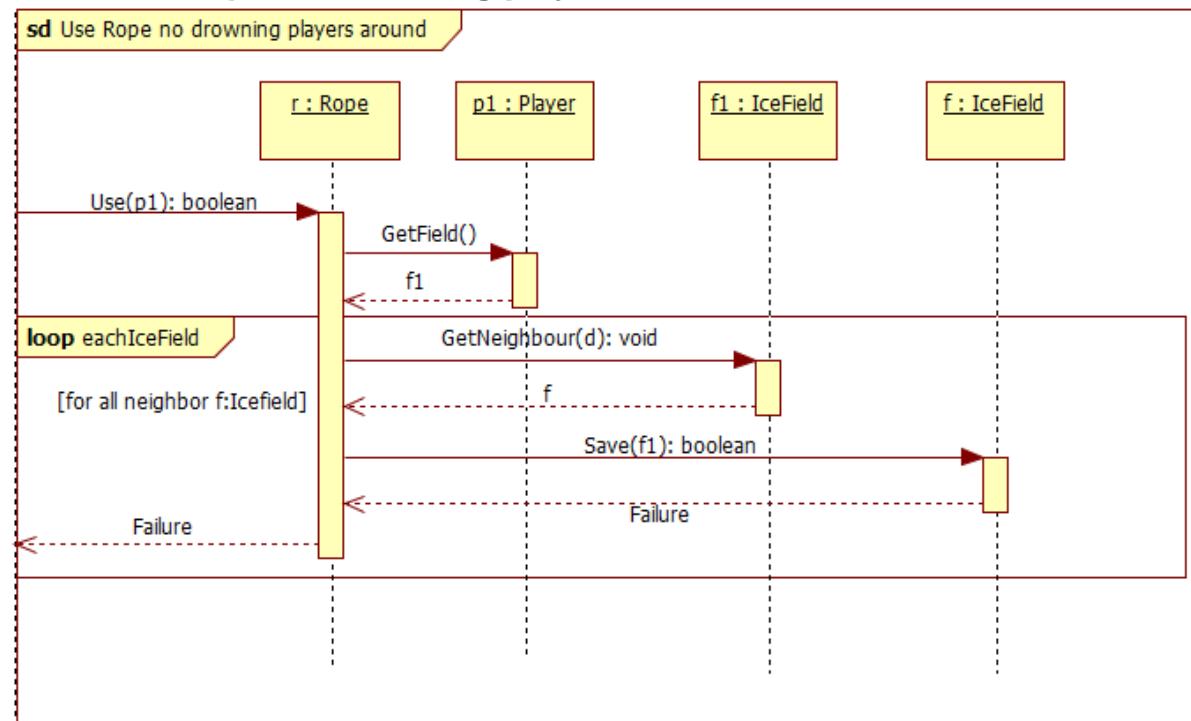
5.3.4 Use flare gun, somebody is missing



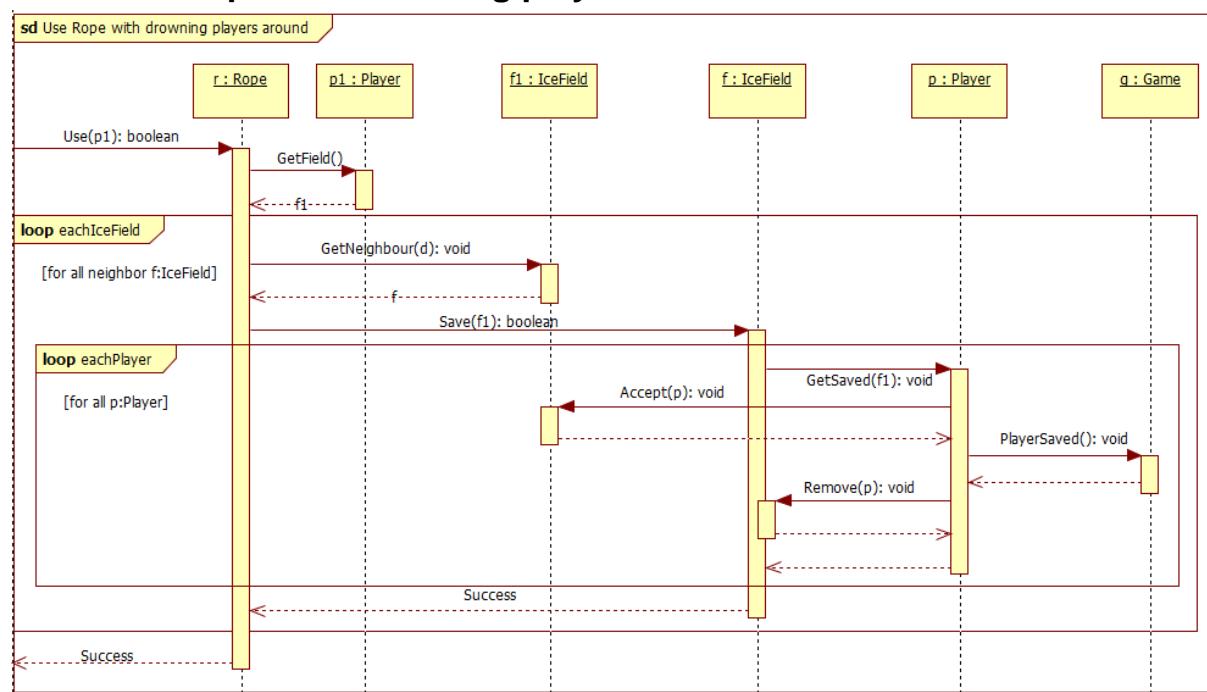
5.3.5 Use Food



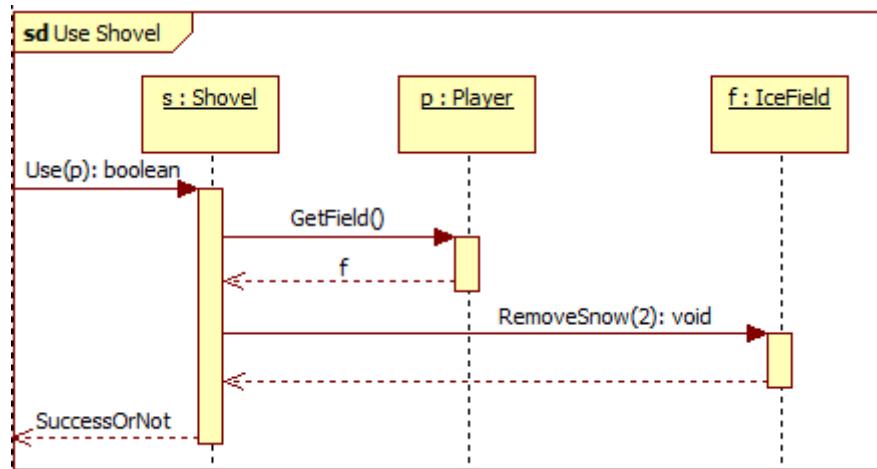
5.3.6 Use Rope, no drowning players around



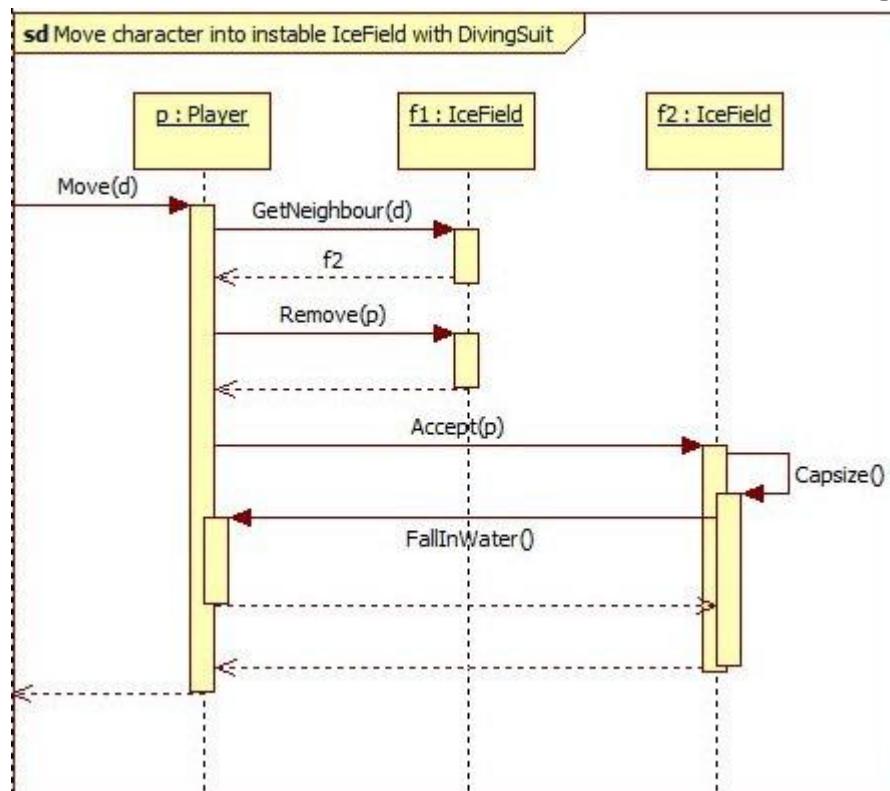
5.3.7 Use Rope with drowning players around



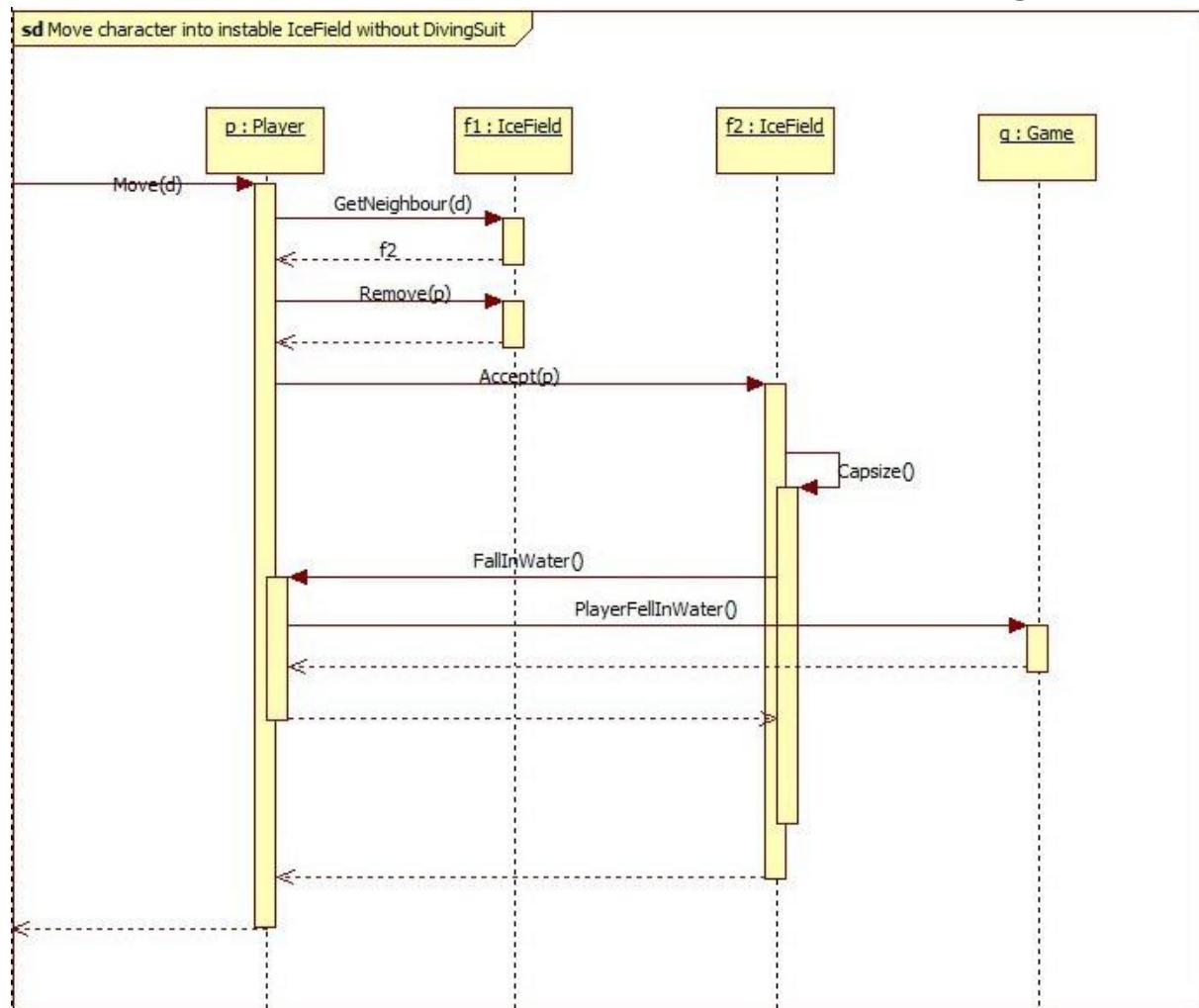
5.3.8 Use Shovel



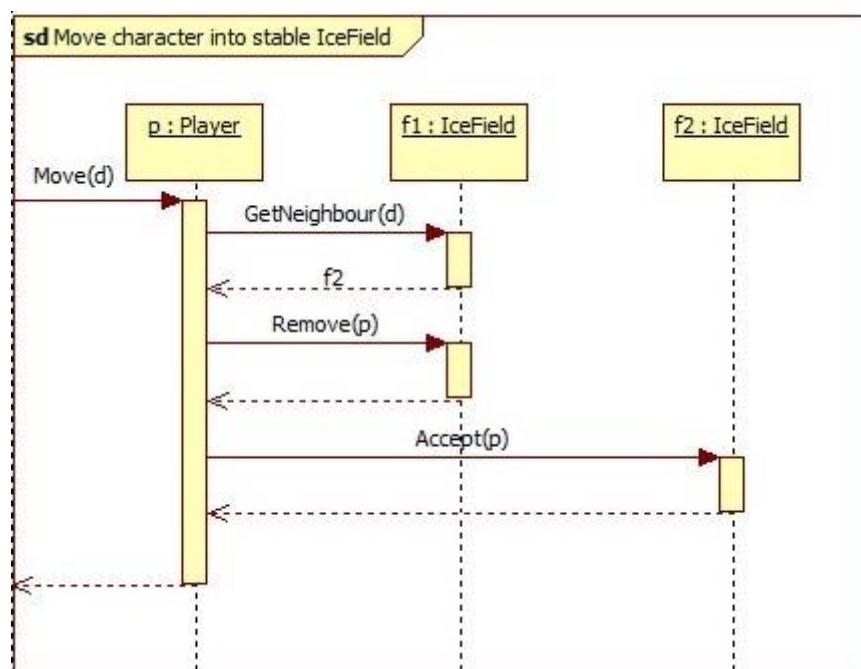
5.3.9 Move character into instable icefield with divingsuit



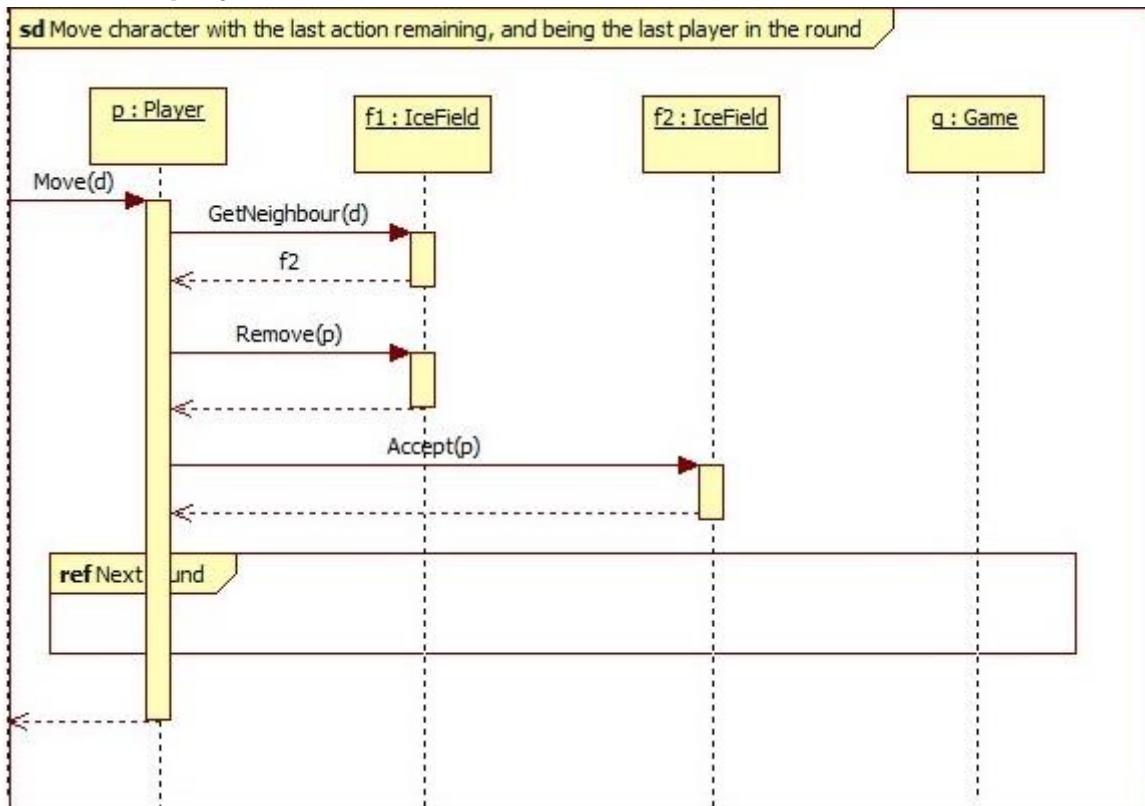
5.3.10 Move character into instable icefield without divingsuit



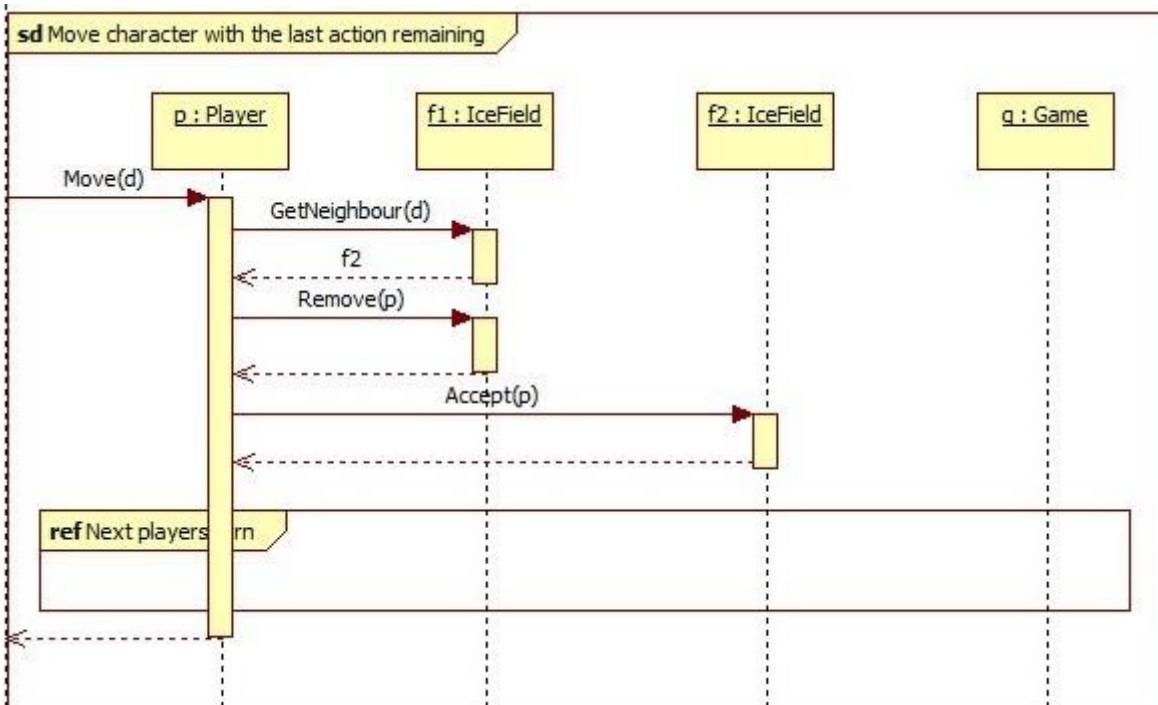
5.3.11 Move character into stable icefield



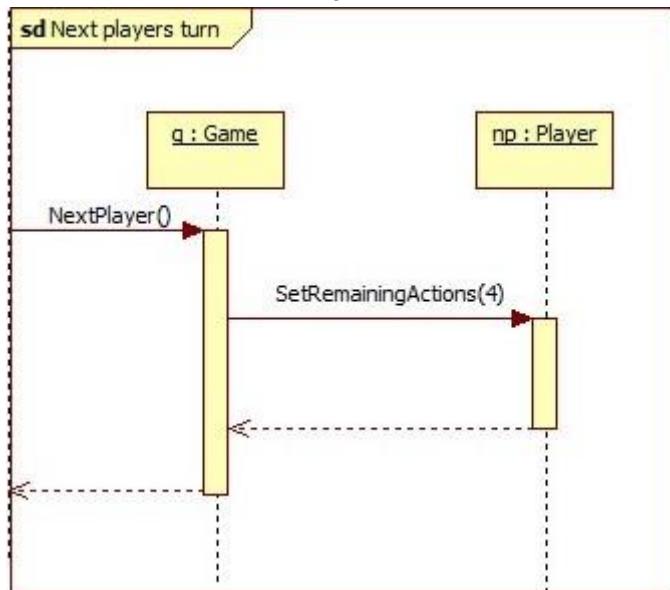
5.3.12 Move character with the last action remaining, and being the last player in the round



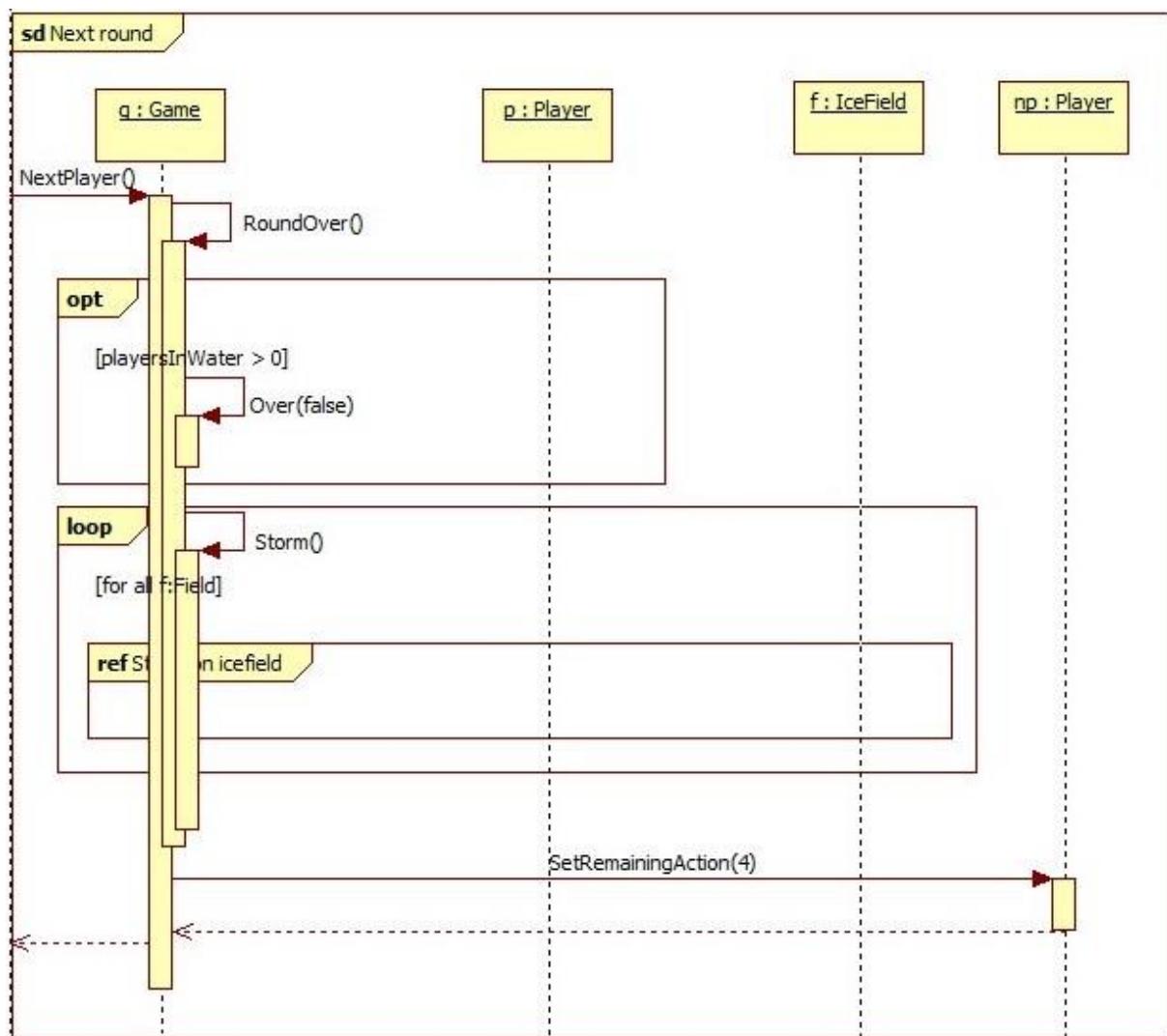
5.3.13 Move character with the last action remaining



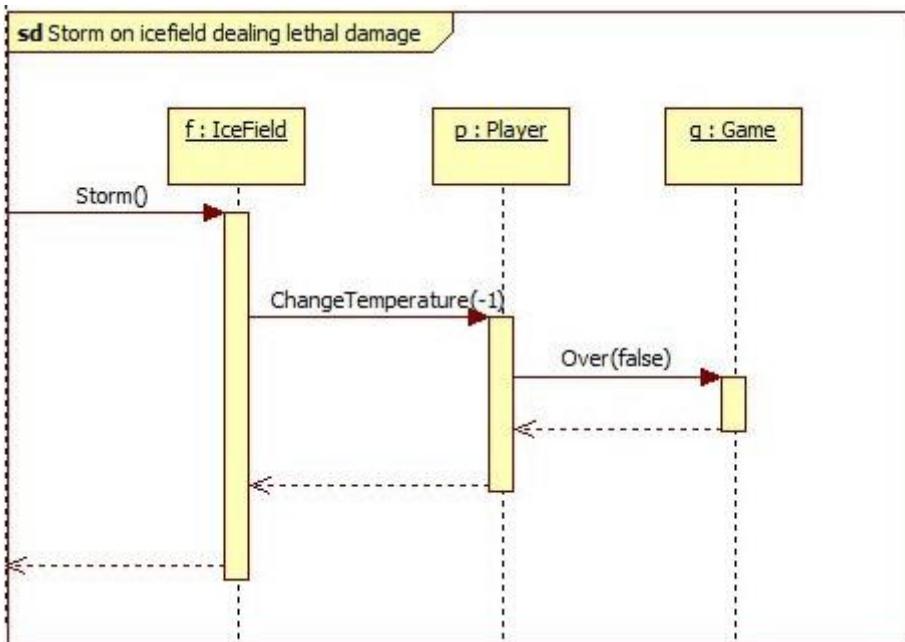
5.3.14 Next Players turn



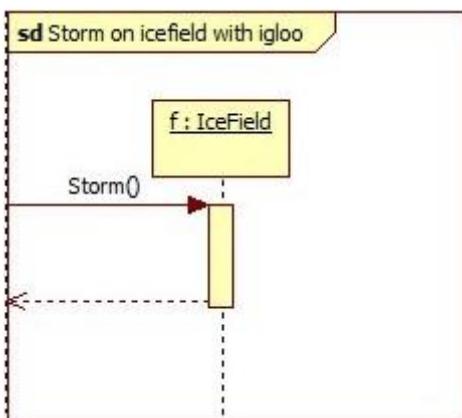
5.3.15 Next round



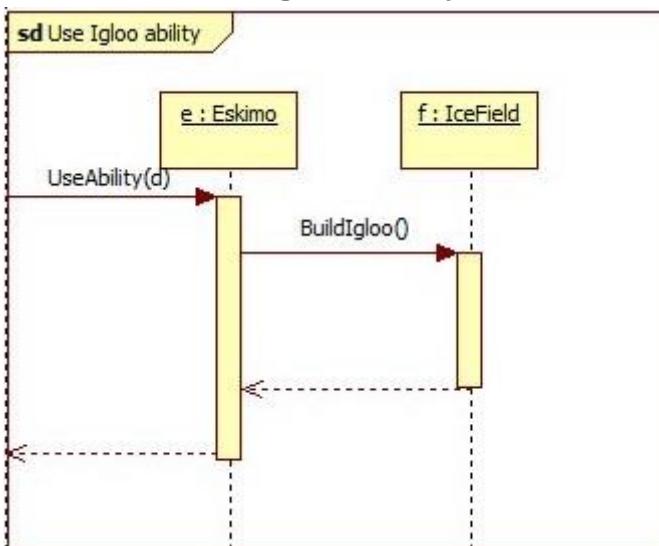
5.3.16 Storm on icefield dealing lethal damage



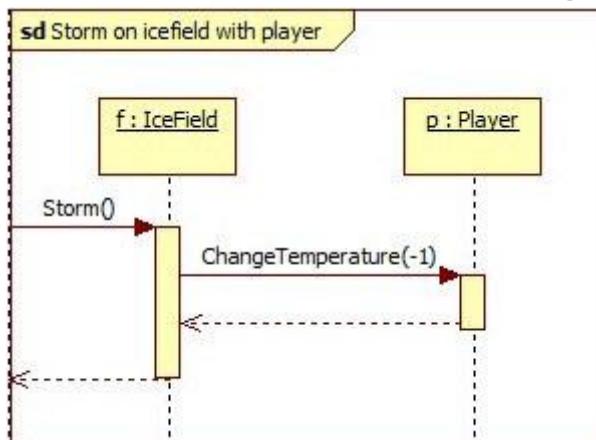
5.3.17 Storm on icefield with igloo



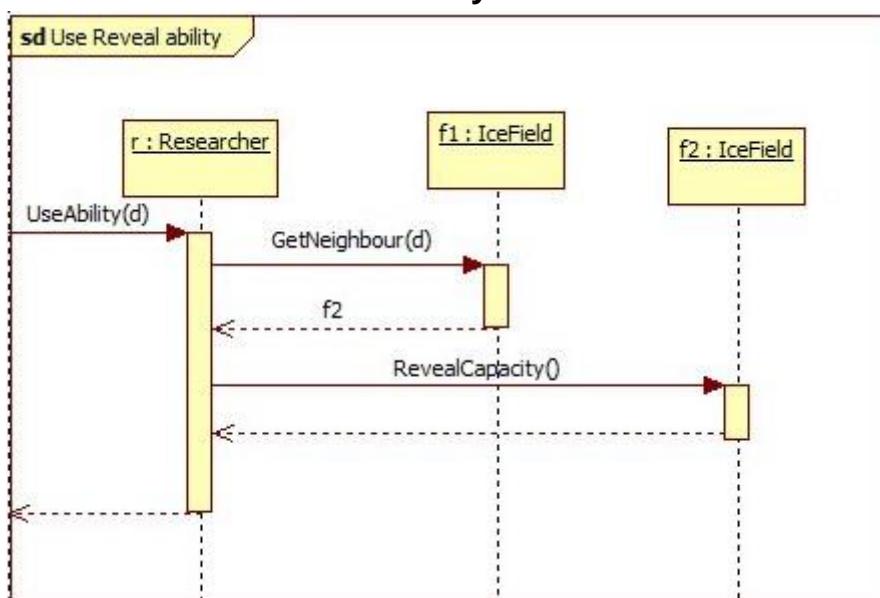
5.3.18 Use Igloo Ability



5.3.19 Storm on icefield with player

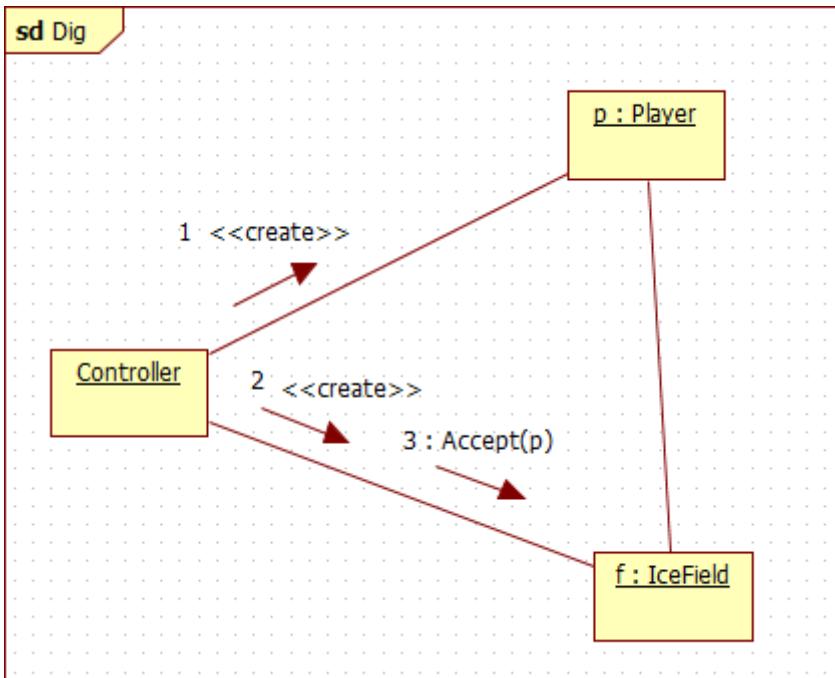


5.3.20 Use Reveal ability

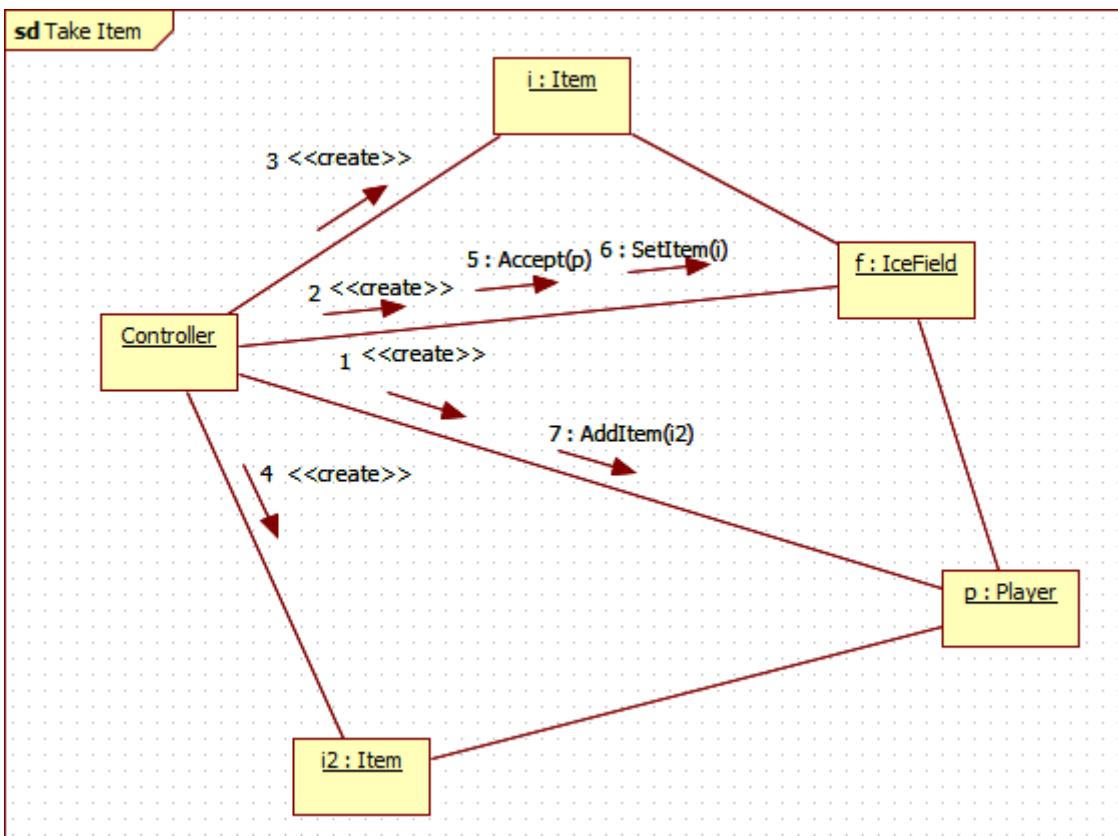


5.4 Kommunikációs diagramok

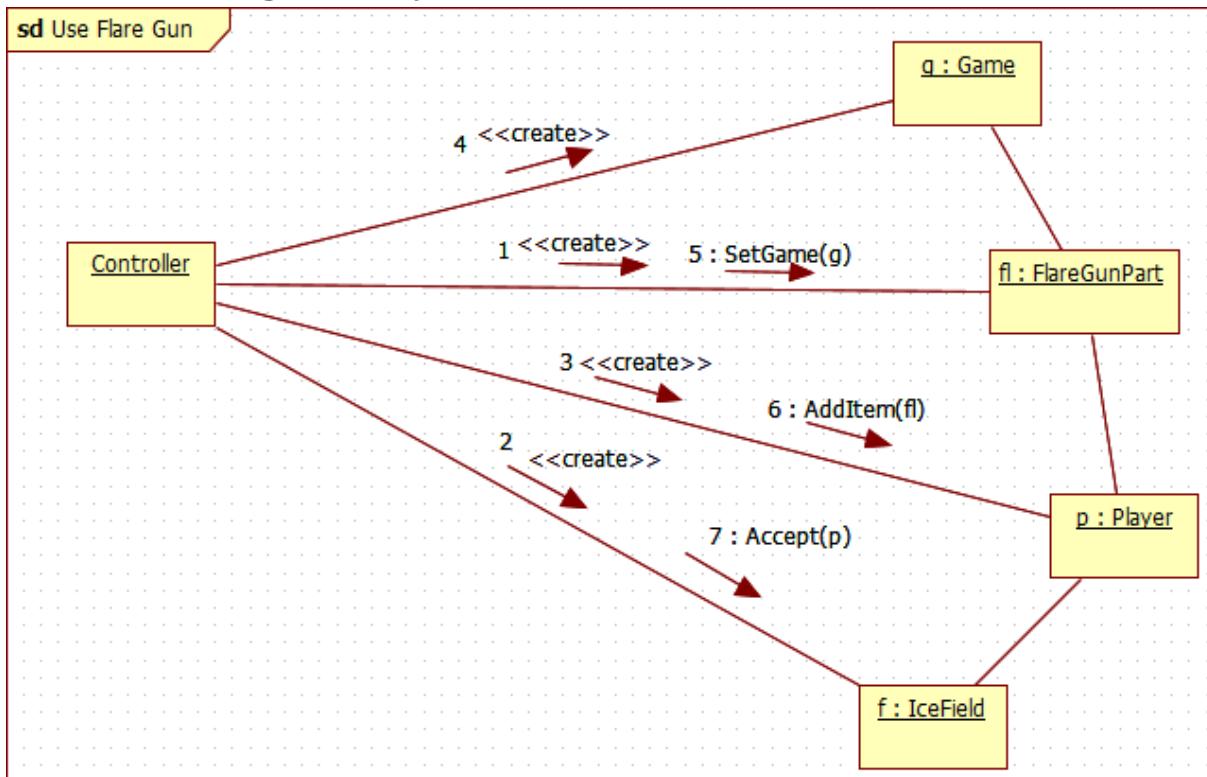
5.4.1 Dig



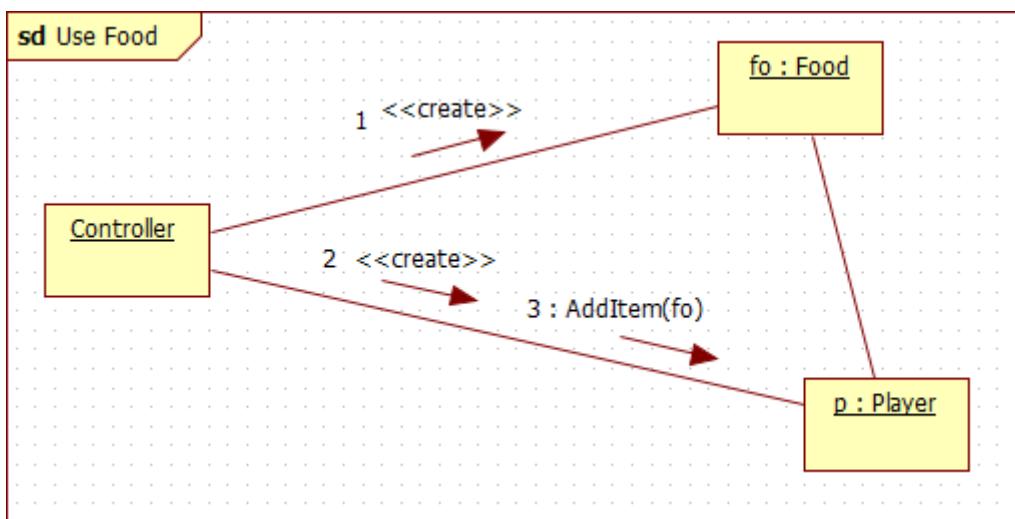
5.4.2 Take Item



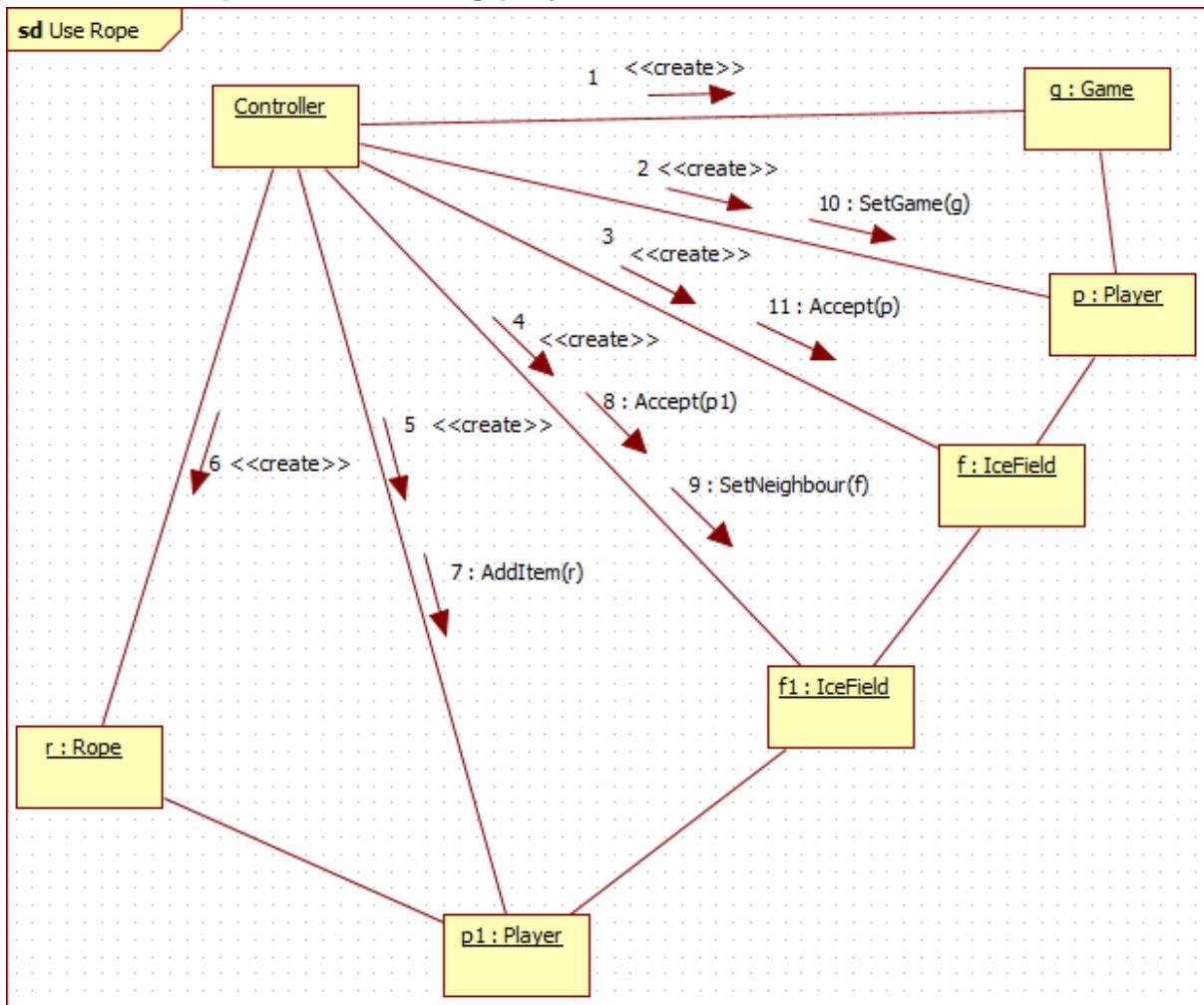
5.4.3 Use flare gun, everyone is here



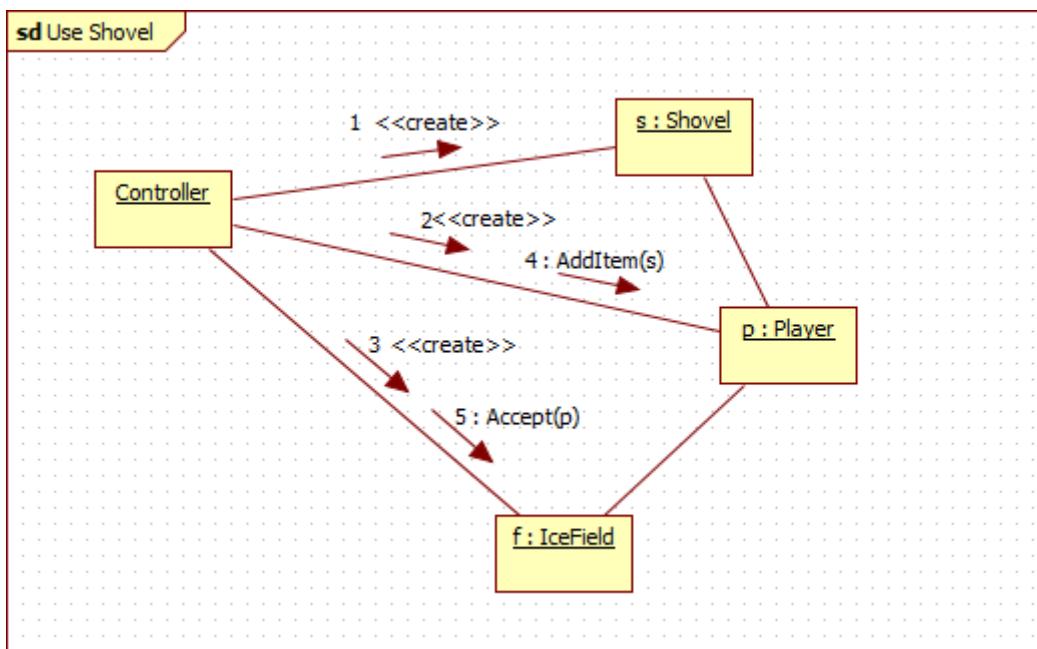
5.4.4 Use Food



5.4.5 Use Rope, no drowning players around

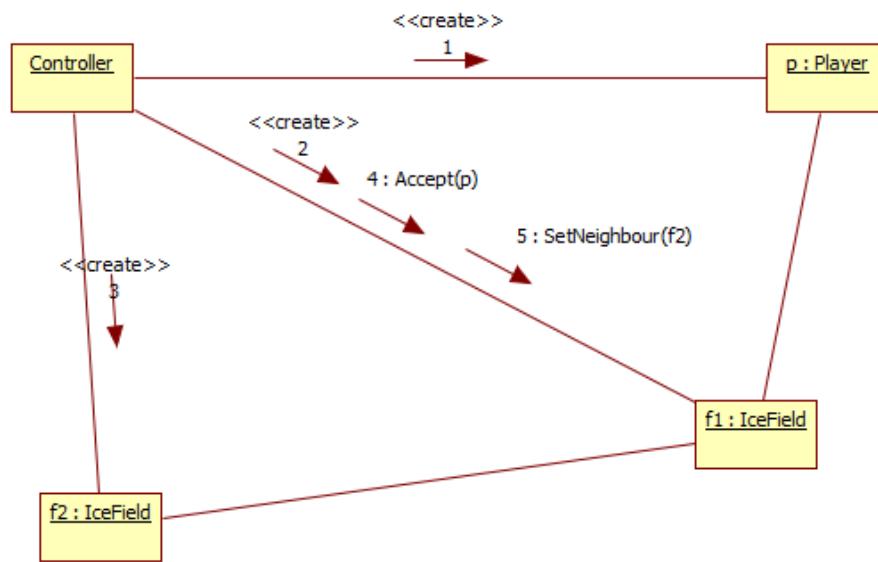


5.4.6 Use Shovel



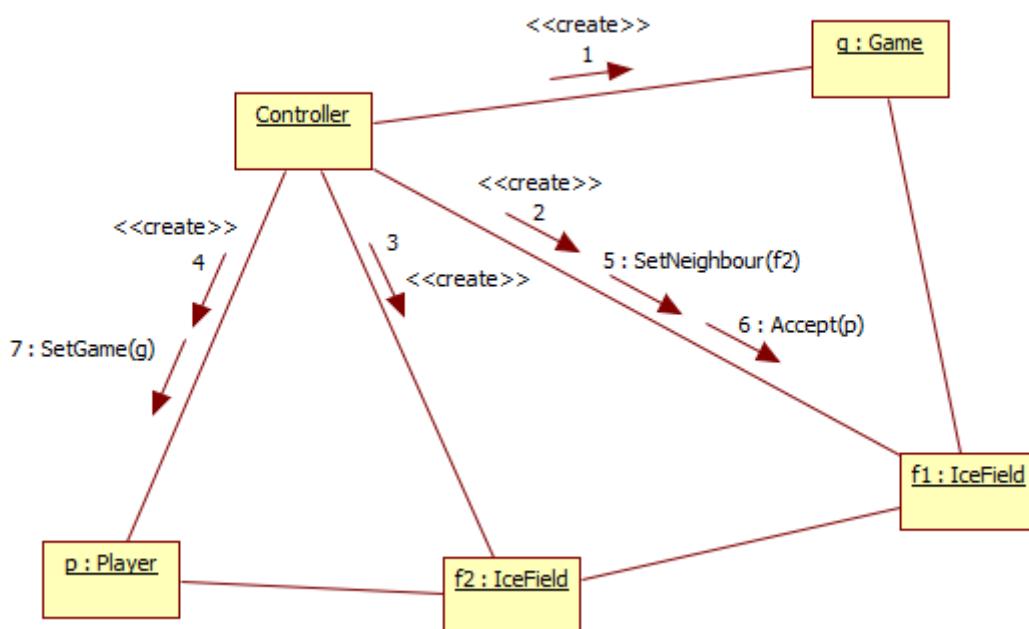
5.4.7 Move character into instable icefield with divingsuit

sd Move character into instable IceField with DivingSuit



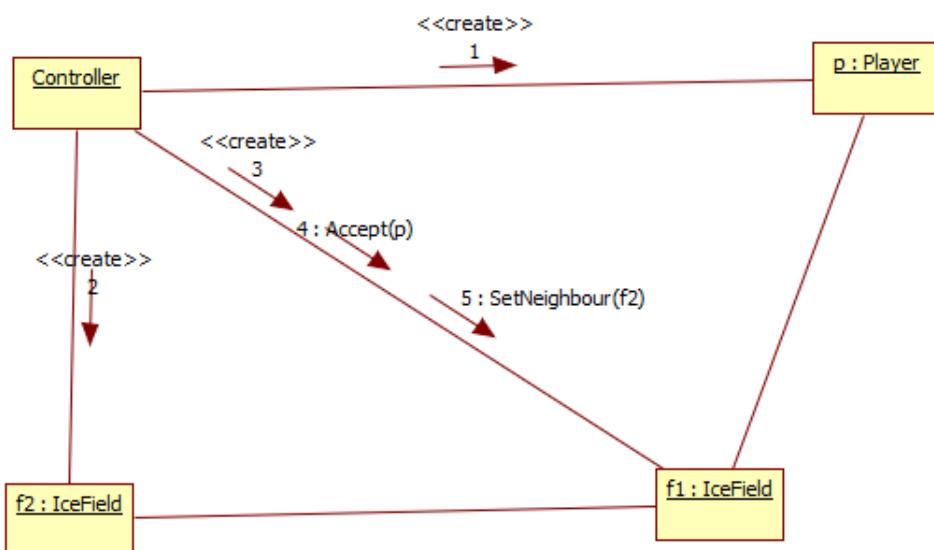
5.4.8 Move character into instable icefield without divingsuit

sd Move character into instable IceField without DivingSuit



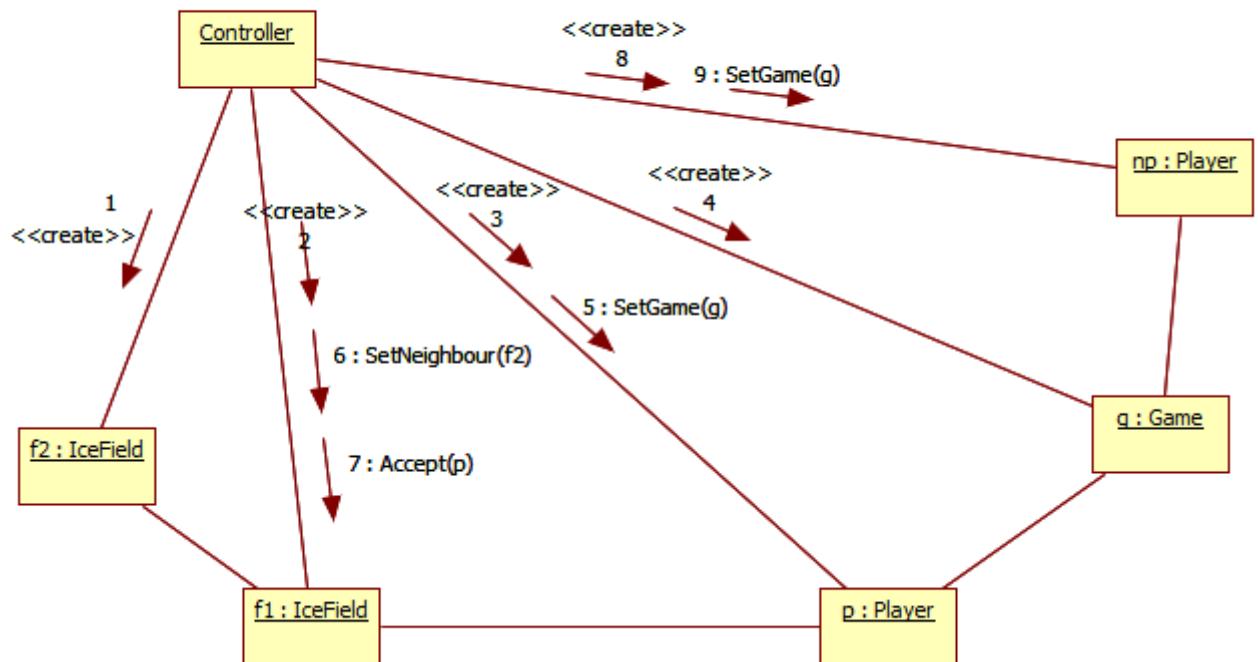
5.4.9 Move character into stable icefield

sd Move character into stable IceField



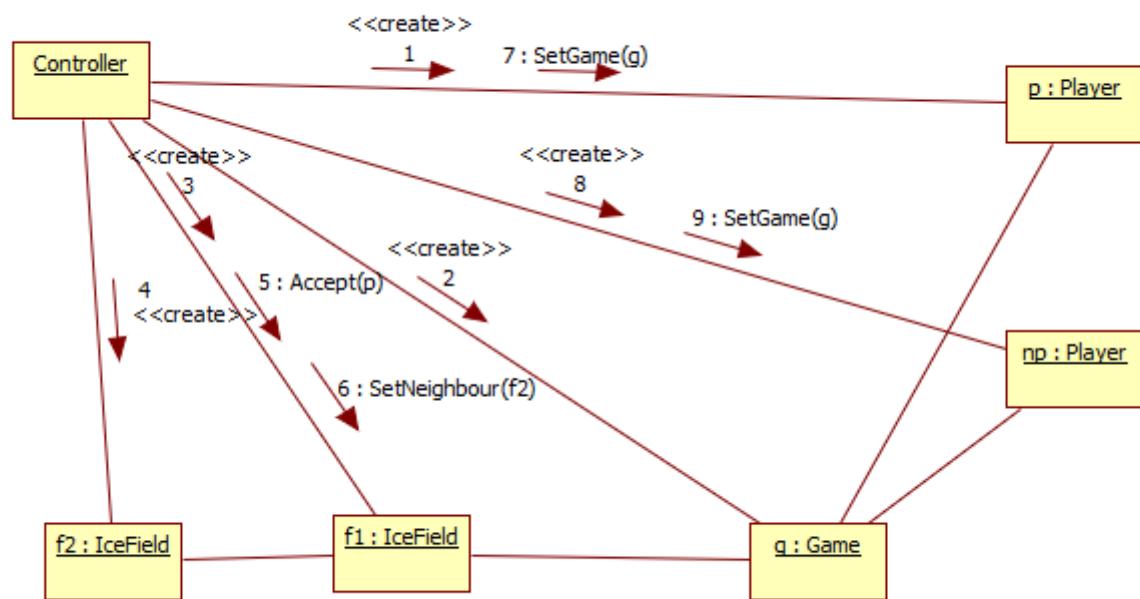
5.4.10 Move character with the last action remaining, and being the last player in the round

sd Move character with the last action remaining, and being the last player in the round



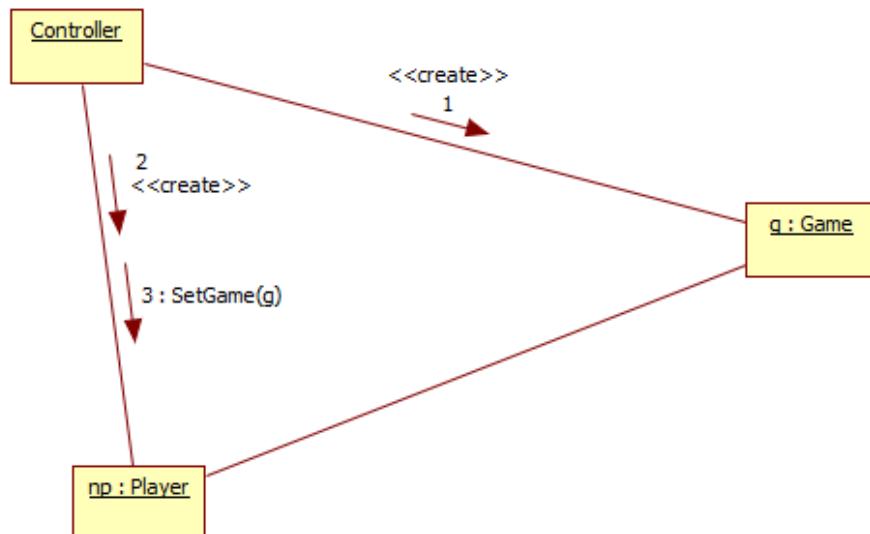
5.4.11 Move character with the last action remaining

sd Move character with the last action remaining



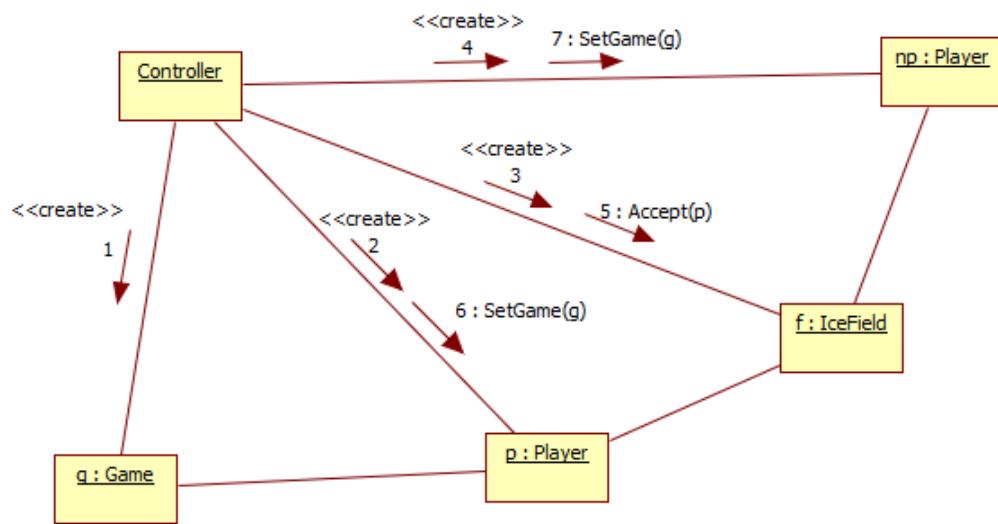
5.4.12 Next Players turn

sd Next players turn



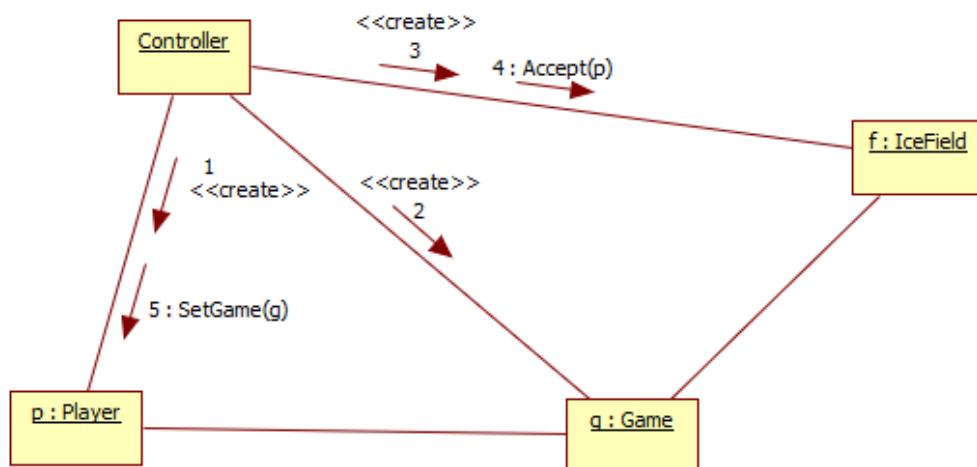
5.4.13 Next round

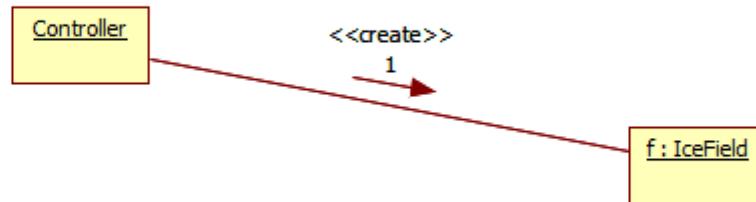
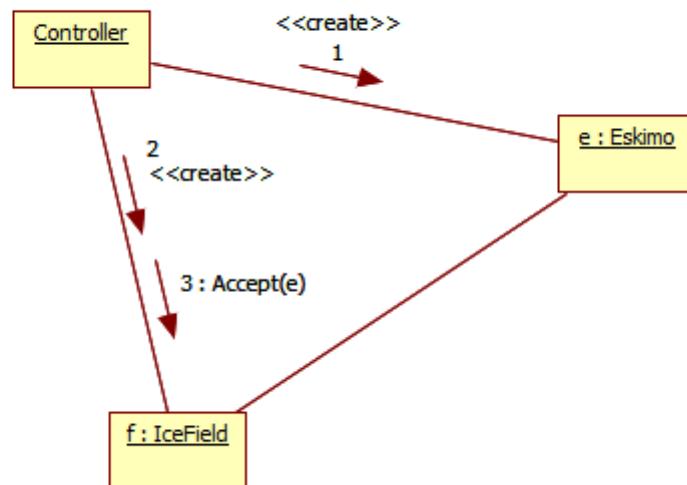
sd Next round



5.4.14 Storm on icefield dealing lethal damage

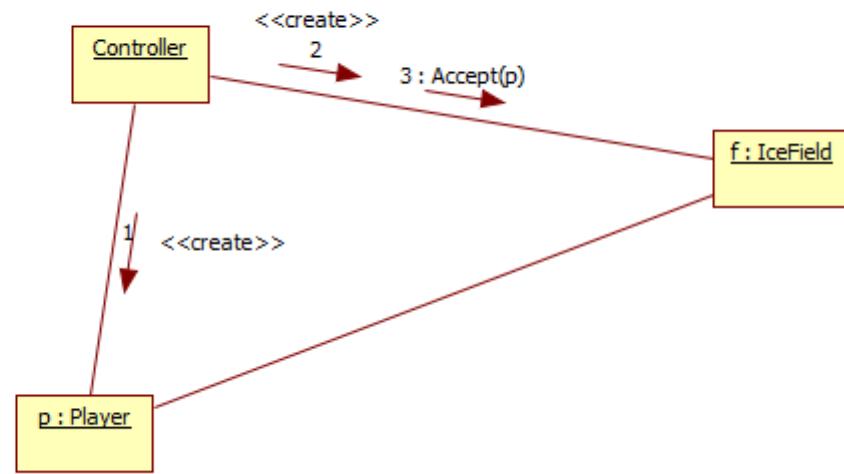
sd Storm on icefield dealing lethal damage



5.4.15 Storm on icefield with igloo**sd** Storm on icefield with igloo**5.4.16 Use Igloo Ability****sd** Use Igloo ability

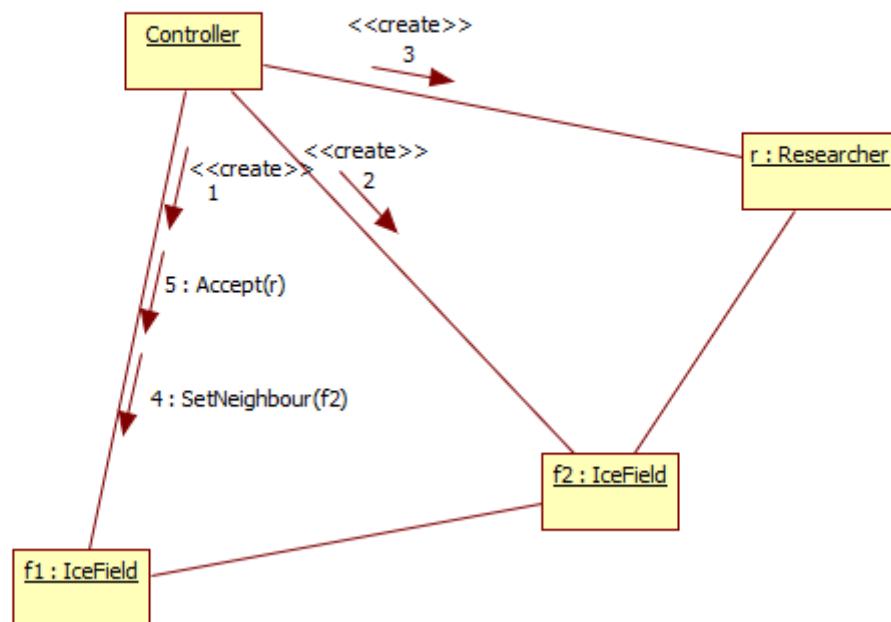
5.4.17 Storm on icefield with player

sd Storm on icefield with player



5.4.18 Use Reveal ability

sd Use Reveal Ability



5.4.19

Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.03.8. 12:00	1 óra	Bartsch	Tervező piszkozat készítése
2020.03.10. 18:30	1,5 óra	Bartsch	5.1, 5.2 ,5.3
2020.03.11. 21:00	0.5 óra	Karmacsi Suciu Seres Takács Bartsch	Megbeszélés: A szerdai foglalkozáson kapott visszajelzések implementálásának megtervezése
2020.03.12. 14:00	2 óra	Bartsch	5.1, 5.2
2020.03.21. 18:00	3 óra	Takács	Szekvencia diagramok, 5.3.9 – 5.3.12, 5.3.16 – 5.3.20
2020.03.21. 17:00	1 óra	Takács	Szekvencia diagramok, 5.3.12 – 5.3.15
2020.03.22. 18:00	2 óra	Suciu	Kommunikációs diagramok készítése
2020.03.22. 21:00	1 óra	Bartsch	5.1, 5.2
2020.03.22. 13:00	3 óra	Seres	Szekvencia diagramok, 5.3.1.- 5.3.8.
2020.03.22 16:00	3 óra	Karmacsi	Kommunikációs diagramok készítése
2020.03.23. 01:00	2 óra	Suciu	Dokumentum szerkesztése, képek vágása

6. Szkeleton beadás

36 – press_f

Konzulens:
Ludmány Balázs

Csapattagok

Karmacsi Péter	L94HDE	karmacsi.peter@freemail.hu
Suciú Barnabás	ASOG9J	szubar10@gmail.com
Bartsch Dávid	BXURDI	bartsch.david.a@gmail.com
Seres Ádám	JF204C	seres.adm@gmail.com
Takács Marcell Adrián	QITM0G	marci.takacs@gmail.com

2020.03.30.

6. Szkeleton beadás

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Direction.java	1 KB	2020.03.28 19:09	Az irányok enum-ja
DivingGear.java	1 KB	2020.03.25 19:23	A búvárruha osztálya
Eskimo.java	1 KB	2020.03.25 19:23	Az eszkimók osztálya
FlareGunPart.java	2 KB	2020.03.25 19:23	A jelzőpisztoly alkatrész osztálya
Food.java	1 KB	2020.03.25 19:23	Az étel osztálya
Game.java	4 KB	2020.03.25 19:23	A játék osztálya
IceField.java	5 KB	2020.03.25 19:23	A jégmezők osztálya
Item.java	2 KB	2020.03.25 19:23	A tárgyak ősosztálya
Main.java	5 KB	2020.03.28 16:54	A main függvény
MethodPrinter.java	2 KB	2020.03.28 14:52	A függvényhívásokat kiíró osztály
Player.java	5 KB	2020.03.25 19:23	A játékos karakterek ősosztálya
Researcher.java	1 KB	2020.03.25 19:23	A kutatók osztálya
Rope.java	1 KB	2020.03.25 19:23	A kötél osztálya
Shovel.java	1 KB	2020.03.25 19:23	Az ásó osztálya
Skeleton.java	12 KB	2020.03.28 16:54	A use-case-ek

6.1.2 Fordítás

1. Győződjünk meg róla, hogy a JDK mappája hozzá van adva a JAVA_HOME környezeti változóhoz, illetve a JDK bin mappája hozzá van adva a PATH-hez.(Ha nincs JAVA_HOME változó, akkor hozza létre és adja hozzá a JDK mappájának elérési útját!)
2. Navigálunk valamelyen paracssorral a kitömörtet forrásfájlokat tartalmazó mappába.
3. Adjuk ki az alábbi parancsokat(Windows alatt):

```
mkdir .\class
javac -d .\class .\Direction.java .\DivingGear.java
.\Eskimo.java .\FlareGunPart.java .\Food.java .\Game.java
.\IceField.java .\Item.java .\Main.java
.\MethodPrinter.java .\Player.java .\Researcher.java
.\Rope.java .\Shovel.java .\Skeleton.java
```

(Linux alatt ugyanez, csak “\” helyett “/” karakterekkel.)

6.1.3 Futtatás

Windows alatt adjuk ki a következő parancsot:

```
java -cp .\class szkeleton.Main
```

(Linux alatt ugyanez, csak “\” helyett “/” karakterrel.)

6.2 Megjegyzések

A modellen végrehajtottunk néhány változtatást, hogy a jelzőpisztoly működése átláthatóbb legyen.

Az új megoldásban a FlareGunPurt osztály felülírja az ōs Equip metódusát, melyben értesíti a Game osztályt, annak egy új metódusával (FlareGunPartFound), hogy megtalálták. A Game eltárolja a még megtalálandó darabok számát, így az IceField CountGunParts metódusára nincs szükség.

6.3 Értékelés

Tag neve	Tag neptun	Munka százalékban
Karmacsi Péter	L94HDE	20
Bartsch Dávid Anasztáz	BXURDI	20
Seres Ádám	JF204C	20
Takács Marcell Adrián	QITM0G	20
Suciú Barnabás	ASOG9J	20

6.4 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.03.25. 19:00	1 óra	Karmacsí Suciu Seres Takács Bartsch	Értekezlet. Döntés: Feladatok felosztása, egységes kiírási sablon kialakítása
2020.03.28. 16:00	1 óra	Karmacsí Suciu Seres Takács	Megbeszélés. Részletek egyeztetése.
2020.03.28. 17:00	2 óra	Karmacsí	Game osztály függvényei, fordítás és futtatás leírása. 6.1.2, 6.1.3
2020.03.28. 17:00	2 óra	Takács	Icefield osztály
2020.03.29. 12:15	4 óra	Bartsch	Skeleton osztály
2020.03.29. 16:00	3 óra	Suciu	Player osztály
2020.03.29. 17:00	3 óra	Seres	Item osztály
2020.03.29. 18:00	5 óra	Takács Bartsch Seres Karmacsí Suciu	Hibakeresés, menü tervezése, dokumentum szerkesztése
2020.03.30 11:00	2 óra	Takács Bartsch Karmacsí Suciu	Dokumentum véglegesítése, menü for

7. Prototípus koncepciója

36 – press_f

Konzulens:
Ludmány Balázs

Csapattagok

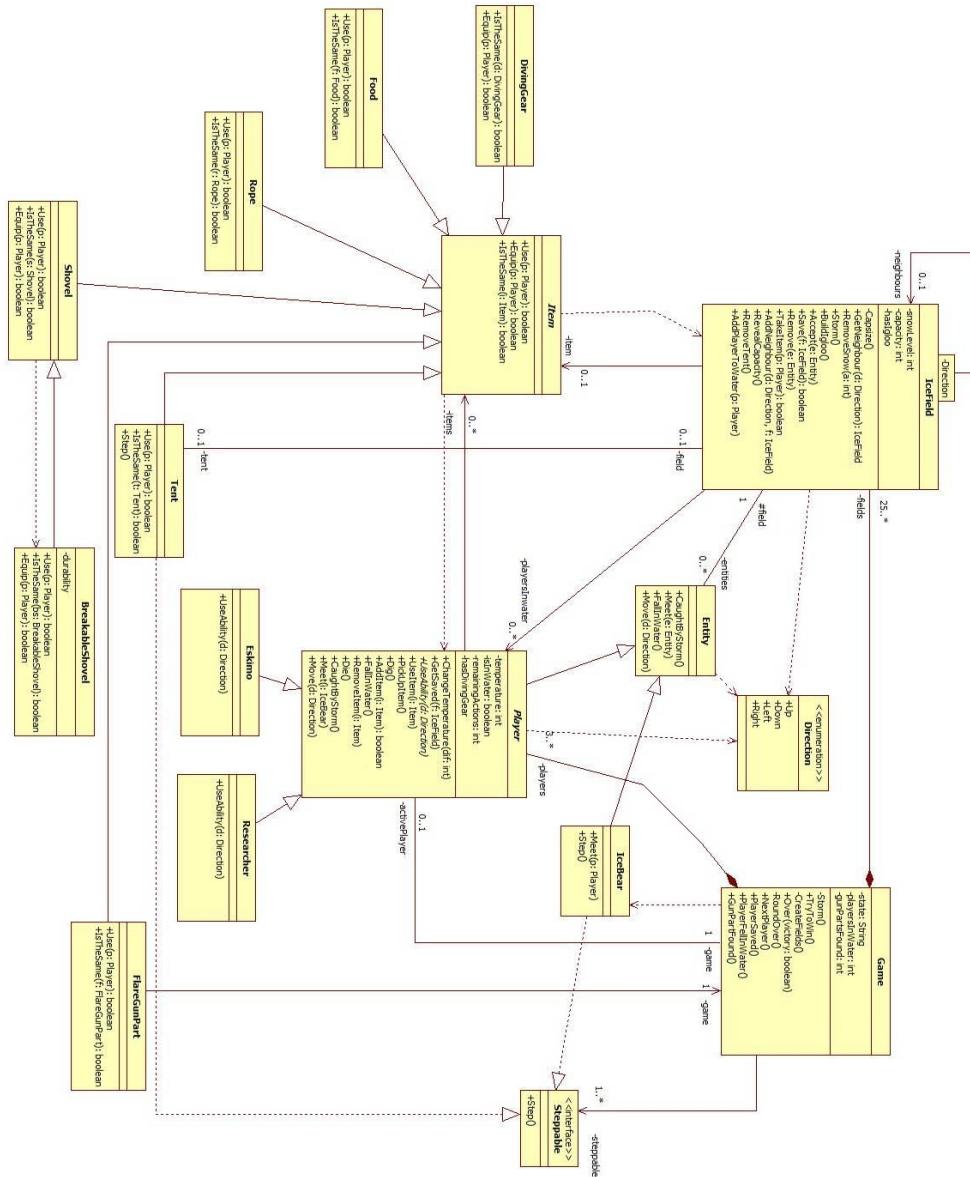
Karmacsi Péter	L94HDE	karmacsi.peter@freemail.hu
Suciú Barnabás	ASOG9J	szubar10@gmail.com
Bartsch Dávid	BXURDI	bartsch.david.a@gmail.com
Seres Ádám	JF204C	seres.adm@gmail.com
Takács Marcell Adrián	QITM0G	marci.takacs@gmail.com

2020.04.06.

7. Prototípus koncepciója

7.0 Változás hatása a modellre

7.0.1 Módosult osztálydiagram



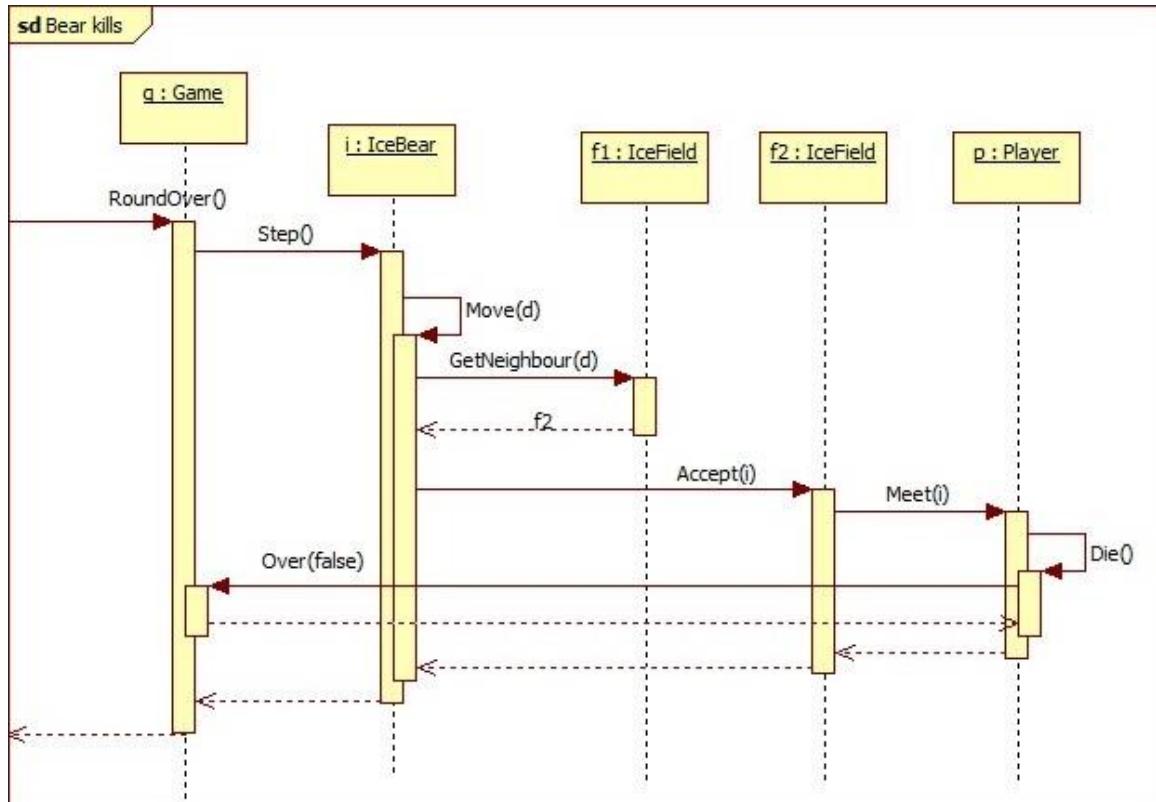
7.0.2 Új vagy megváltozó metódusok

- Player.RemoveItem(i: Item)**: Eltávolítja a paraméterként kapott item-et a játékos tárgyai közül.
- Player.Die()**: A játékos elkapta a jegesmedve. Jelez a Game-nek, hogy vesztettek.
- Player.CaughtByStorm()**: Csökkenti a játékos testhőjét eggyel.
- Player.Meet(i: IceBear)**: Meghal, szól a Game-nek, hogy vesztettek.
- Game.TryToWin()**: Ha minden játékos egy helyen áll és ki van ásva az összes jelzőpisztoly alkatrész, akkor a játékosok megnyerték a játékot.

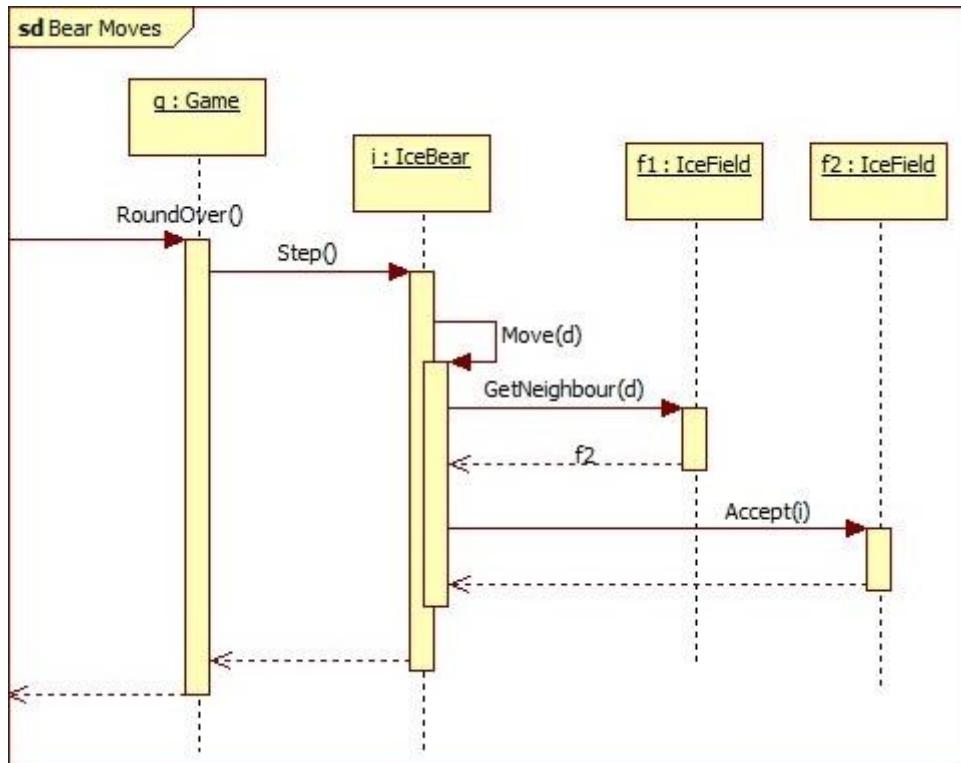
- **Game.RoundOver()**: Kiegészült annyival, hogy a medvének szól, hogy mozogjon és szól a sátraknak, hogy rombolják le magukat ha már fel vannak építve valahol.
- **Game.GunpartFound()**: Növeli a megtalált alkatrészek számát eggyel.
- **IceField.RemoveTent()**: Lerombolja a sárat.
- **IceField.AddPlayerToWater**: Hozzáad egy játékost a vízben lévő játékosok listájához.
- **BreakableShovel.Use(P: Player)**: Eltávolít 2 rétegnyi havat a játékos mezőjéről és csökkenti a durability-t eggyel. Ha nullára csökkent, akkor elveszi a játékostól (ezt) az ásót.
- **BreakableShovel.IsTheSame(b: BreakableShovel)**: True-val tér vissza.
- **BreakableShovel.Equip(p: Plyaer)**: A játékosnak adja a törhető ásót, ha nincs még nála rendes ásó.
- **Shovel.Equip(p: Plyaer)**: Lecseréli a törhető ásóját a erre, ha már van a játéknál, egyébként csak simán felveszi.
- **Tent.Use(p: Player)**: Egy sárat épít a játékos mezőjére.
- **Tent.IsTheSame(t: Tent)**: True-val tér vissza.
- **Tent.Step()**: Lerombolja magát, ha fel van építve egy jégmezőn.
- **IceBear.Step()**: Egy véletlenszerű szomszédos mezőre lép a medve.
- **IceBear.Meet(p: Player)**: Megöli a Player-t.
- **Entity.CaughtByStorm()**: Nem csinál semmit alapból, a Player felülírja.
- **Entity.Meet(e: Entity)**: Nem csinál semmit. A Player és IceBear specializálják más paraméterrel, hogy észrevegyék, ha IceBear és Player találkozik.
- **Entity.Move(d: Direction)**: Átlép a d irányba szomszédos jégmezőre.
- **FallInWater()**: Ezzel szól az IceField, hogy vízbe esett.

7.0.3 Szekvencia-diagramok

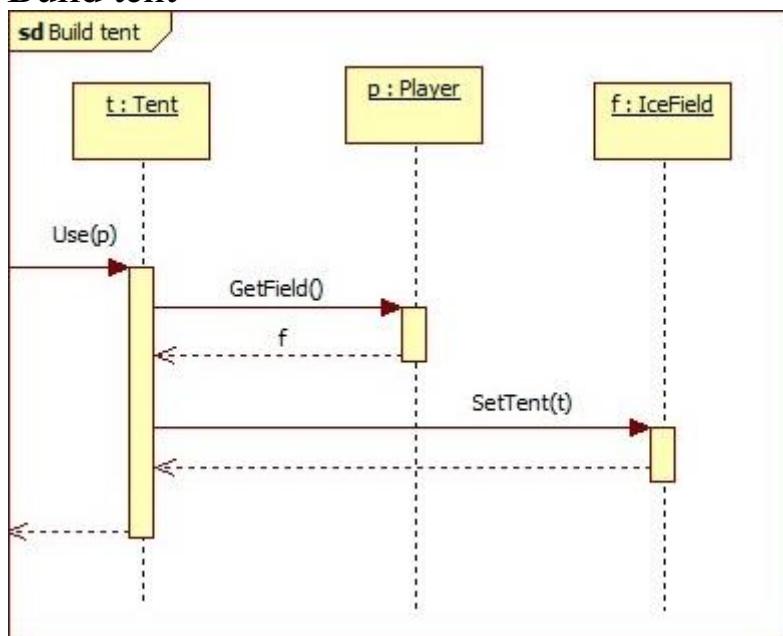
1. Bear kills



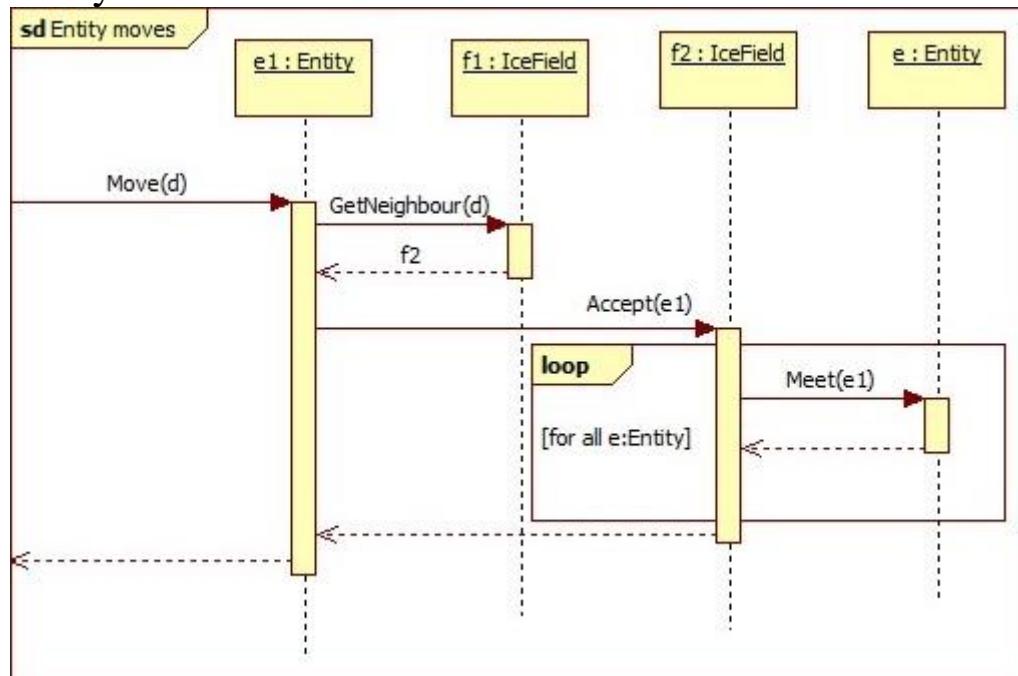
2. Bear moves



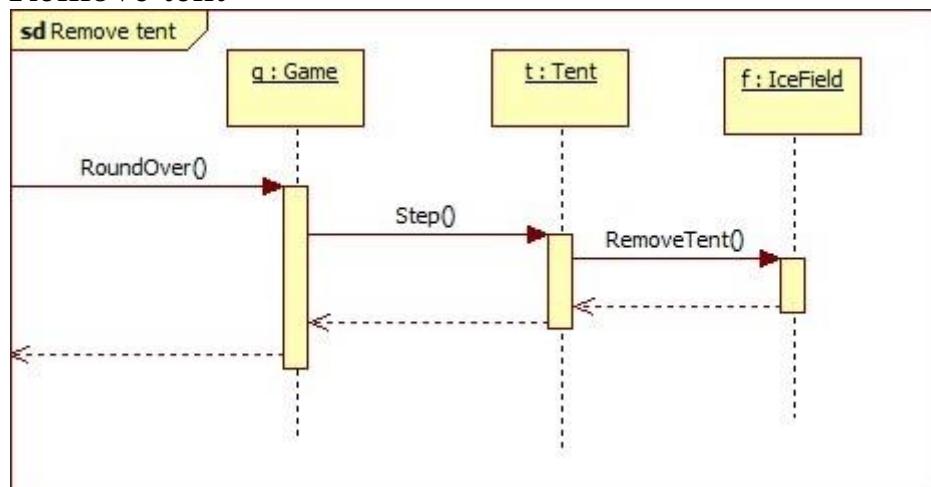
3. Build tent



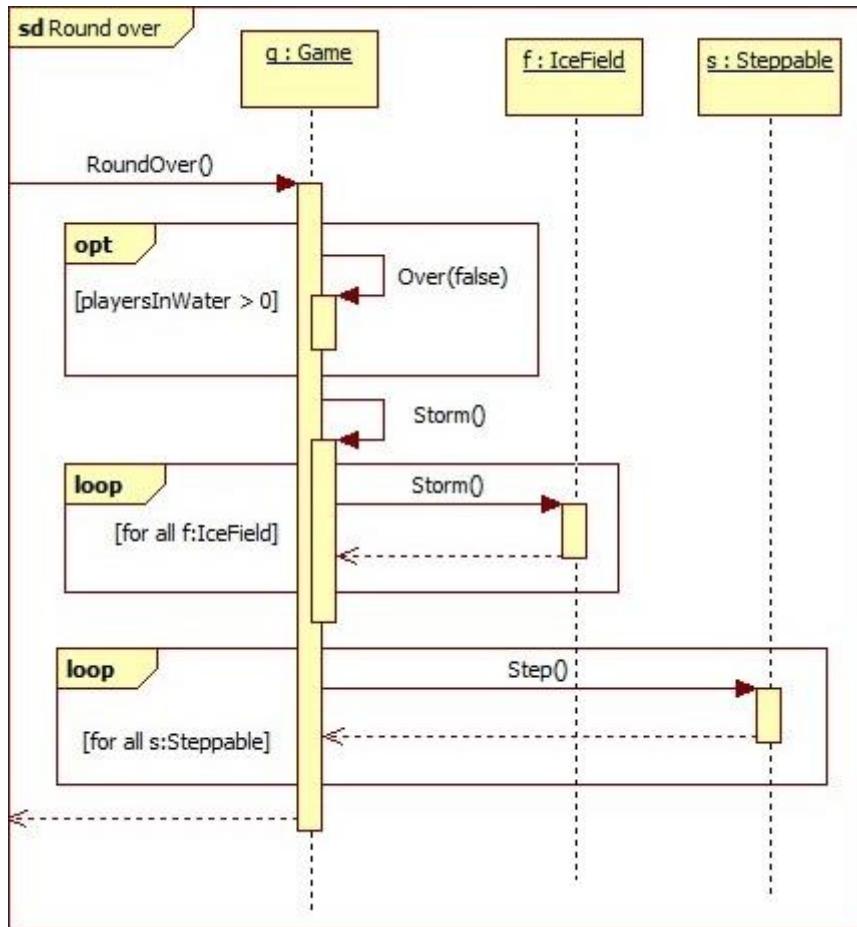
4. Entity moves



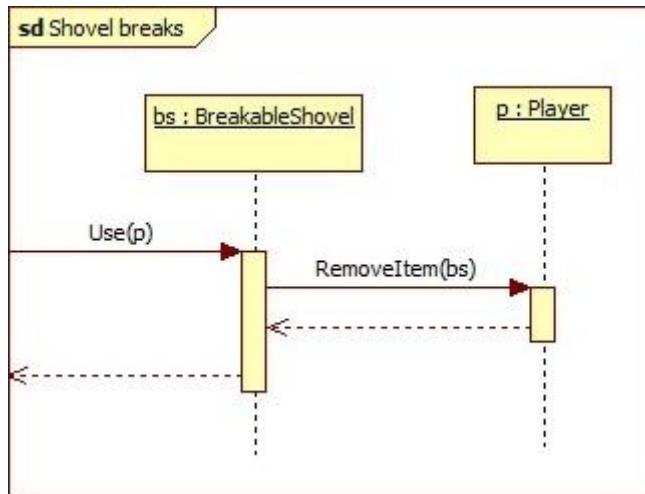
5. Remove tent



6. Round over



7. Shovel breaks



7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A prototípus egy konzolos kezelőfelülettel fog rendelkezni, ahol parancsok begépelésével tudjuk vezérelni a programot. Egy megfelelő formátumú file-al is lehet bevinni ezeket a parancsokat, a tesztelés során ezt ki fogjuk használni.

7.1.2 Bemeneti nyelv

CreateMap

Leírás: Létrehozza a paraméterekben megadott méretű pályát, majd utána beolvassa az egyes mezők adatait, amiket külön sorokban adhatunk meg.

Opciók: createmap <sorok száma> <oszlopok száma>

<hó mennyiség> <kapacitás>

...

<hó mennyiség> <kapacitás>

A mezők adatainak megadása n^*m sorban történik, ahol m a sorok száma, n pedig az oszlopok száma. Az első sor tartozik az első sor első oszlopában lévő mezőhöz, a második az első sor második oszlopában lévő mezőhöz, és így tovább. (Azaz az első n sor adja meg a játéktábla első sorának mezőit, a második n db sor a tábla második sorának mezőit, stb.)

Pl: createmap 2 3

1 2

3 2

4 3

1 1

3 1

4 1

Entity

Leírás: Létrehoz egy entitást, mely lehet egy player vagy jegesmedve, és elhelyezi a névvel megadott mezőn.

Opciók: entity <név> <típus> <célmező>

Pl: entity player1 researcher f_1_1

Pl: entity bear1 bear f_2_2

CreateItem

Leírás: Egy item objektumot hoz létre a paraméterben megadott típusból a szintén paraméterben megadott mezőre.

Opciók: createitem <név> <típus> <célmező>

Pl: createitem shover f_1_1

UseItem

Leírás: Az éppen körön lévő játékossal végrehajtja a paraméterben megadott Item-hez tartozó akciót.

Opciók: useitem <név>

Pl: useitem food1

UseAbility

Leírás: Az éppen körön lévő játékos végrehajtja az akcióját, mely kutató esetében feltérképez egy szomszédos mezőt, és eszkimó esetében épít egy iglut.

Opciók: useability

Pl: useability

Move

Leírás: Mozgatja a körön lévő játékost a paraméterben megadott irányba.

Opciók: move <irány>

Pl: move up

Placeigloo

Leírás: A paraméterben megadott mezőre épít egy iglut.

Opciók: placeigloo <célmező>

Pl: placeigloo f_1_1

SetActivePlayer

Leírás: Beállítja, hogy melyik játékos legyen éppen aktív.

Opciók: setactiveplayer <játékosnév>

Pl: setactiveplayer player1

SetActions:

Leírás: Beállítja, hogy az aktív játékosnak hány akciót pontja legyen.

Opciók: setremainingactions <szám>

Pl: setactions 2

Info

Leírás: Kílistázza a névvel megadott objektum információit.

Opciók: info <név>

Pl: info player1

Save

Leírás: Kiírja az összes objektum adatait, az Info parancs kimeneti nyelvét felhasználva.

Opciók: save

Pl: save

SetRandom

Leírás: Beállítja, hogy a pályagenerálás és a körvégi események végrehajtásakor véletlenszerű értékekkel dolgozzon, vagy a programba előre beépített, fix lefutást alkalmazza.

Opciók: setrandom <logikai változó>

Pl: setrandom true

Dig

Leírás: Az aktív játékossal végrehajtat egy ásás akciót.

Opciók: dig

Pl: dig

PickUpItem

Leírás: Az aktív játékossal felveteti a mezején található tárgyat.

Opciók: pickupitem

Pl: pickupitem

Load

Leírás: Egy elmentett játékállapotot betölt a paraméterben megadott file-ból.

Opciók: load <file név>

Pl: load mentes.txt

//A fileban a parancsok paramétereihez hasonló formátumban tárolt adatokat tölt be a program.

Példa a file formátumára:

1 2 //fieldek száma

<hómennyiségt> <kapacitás>

<hómennyiségt> <kapacitás>

1 //entitások száma

player1 player f_1_1 //név, típus, mező

1 //itemek száma

shovel f_1_1

1 //igluk száma

f_1_2

AddItem

Leírás: Egyből a kiválasztott játékos eszköztárába helyez egy paraméterben megadott tárgyat.

Opciók: additem <játékosnév> <típus>

pl: additem player1 shovel

7.1.3 Kimeneti nyelv

Alapesetben a parancsok valamilyen hiba esetén jeleznek a szokásos módon, emellett az info parancs kiadásával lehet egy objektumról információkat kapni.

Az info parancs a következő példában szemléltetett módon adja meg az objektum információit:

IceField:

Neighbor up: f_1_3

Neighbor right: f_2_4

Neighbor down: f_3_3

Neighbor left: f_2_2

Snowlevel: 0

Capacity: 3

Hasigloo: true

Item: shovel

Entity1: player1

Entity2: player2

...

EntityN: player3

Player:

Field: f_1_1

Temperature: 2

Isinwater: false

Remainingactions: 3

Hasdivingear: true

Item1: shovel

Item2: food

Item3: flaregunpart

...

ItemN: rope

Tent: tent1

IceBear:

Field: f_1_2

Game:

State: running

Playersinwater: 2

Gunpartsfound: 1

7.2 Összes részletes use-case

Use-case neve	Create map
Rövid leírás	A játéktábla létrehozása.
Aktorok	User, Controller
Főforgatókönyv	1. A felhasználó kiadja a createmap parancsot, valamint megadja a sorok és az oszlopok számát. 2. A felhasználó megadja az egyes jégtáblák adatait. 3. Létrejön a megadott adatoknak megfelelő játéktábla.
Alternatív forgatókönyv	2.A.1. A véletlenszerű viselkedés engedélyezve van, így véletlen pálya jön létre.

Use-case neve	Create entity
Rövid leírás	Jegesmedve, eszkimó, vagy kutató létrehozása, majd adott mezőre helyezése.
Aktorok	User
Forgatókönyv	1. A felhasználó kiadja az entity parancsot, valamint megadja a létrehozandó entitás nevét, típusát, és a jégtábla nevét. 2. Létrejön az új entitás, majd a megfelelő táblára kerül.

Use-case neve	Create item
Rövid leírás	Új tárgy létrehozása, majd elhelyezése a megadott mezőn.
Aktorok	User
Forgatókönyv	1. A felhasználó kiadja a createitem parancsot, valamint megadja a tárgy nevét, típusát, és a jégtábla nevét. 2. Létrejön a tárgy, és a megfelelő jégtáblára kerül.

Use-case neve	Add item
Rövid leírás	Új tárgy létrehozása, majd elhelyezése egy játékos eszköztárában.
Aktorok	User
Forgatókönyv	1. A felhasználó kiadja az additem parancsot, valamint megadja a játékos nevét és a választott tárgy típusát.

Use-case neve	Place igloo
Rövid leírás	Iglu elhelyezése megadott jégtáblára.
Aktorok	
Forgatókönyv	1. A felhasználó kiadja a placeigloo parancsot, valamint megadja a jégtábla nevét. 2. Iglu jön létre a megadott jégtáblán.

Use-case neve	Use item
Rövid leírás	Az aktív játékos használ egy nála lévő tárgyat.
Aktorok	User
Főforgatókönyv	1. A felhasználó kiadja a useitem parancsot, megadva a használandó tárgy nevét. 2. A tárgy használata megtörténik, a játékos veszít egy akciót pontot
Alternatív forgatókönyv	2.A.1 A tárgy használata sikertelen, nem változik semmi.

Use-case neve	Move
Rövid leírás	Az aktív játékos mozgatása egy adott irányba.
Aktorok	User
Főforgatókönyv	<p>1. A felhasználó kiadja a move parancsot, valamint megadja a mozgás irányát.</p> <p>2. A mozgás megtörténik a játékos veszít egy akciót pontot.</p>
Alternatív forgatókönyv	2.A.1. A mozgás sikertelen, a játékos marad a helyén.

Use-case neve	Get info
Rövid leírás	A program objektumairól adatok kiíratása.
Aktorok	User
Forgatókönyv	<p>1. A felhasználó kiadja az info parancsot, valamint megadja az objektum nevét.</p> <p>2. A program kiírja a választott objektum mezőinek értékét.</p>

Use-case neve	Set random
Rövid leírás	A véletlenszerű viselkedés be-, illetve kikapcsolása.
Aktorok	Felhasználó
Forgatókönyv	<p>1. A felhasználó kiadja a setrandom parancsot, valamint megadja a logikai változó értékét.</p> <p>2. A beállítás a megfelelő értékre változik.</p>

Use-case neve	Set actions
Rövid leírás	Aktív játékos akciót pontjának beállítása megadott értékre.
Aktorok	User
Forgatókönyv	<p>1. A felhasználó kiadja a setactions parancsot, valamint megadja az értéket.</p> <p>2. A játékos akciót pontjainak száma a megadott értékre módosul.</p>

Use-case neve	Dig
Rövid leírás	Aktív játékos utasítása ásásra.
Aktorok	Felhasználó
Főforgatókönyv	<p>1. A felhasználó kiadja a dig parancsot.</p> <p>2. Az ásás sikeresen megtörténik, a játékos veszít egy akciót pontot.</p>
Alternatív forgatókönyv	2.A.1. Az ásás sikertelen, nem történik semmi.

Use-case neve	Pick up item
Rövid leírás	Az aktív játékos utasítása tárgy felvételére.
Aktorok	Felhasználó
Főforgatókönyv	1. Az aktív játékos utasítása tárgy kiásására. 2. A játékos sikeresen kiássa a tárgyat és veszít egy akciót pontot.
Alternatív forgatókönyv	2.A.1. Az akció sikertelen, nem történik semmi.

Use-case neve	Load
Rövid leírás	Játékállás betöltése fájlból.
Aktorok	Felhasználó.
Forgatókönyv	1. A felhasználó kiadja a load parancsot, valamint megadja a fájl nevét. 2. A játékállás betöltődik.
Use-case neve	Create item
Rövid leírás	Új tárgy létrehozása, majd elhelyezése a megadott mezőn.
Aktorok	User
Forgatókönyv	1. A felhasználó kiadja a createitem parancsot, valamint megadja a tárgy nevét, típusát, és a jégtábla nevét. 2. Létrejön a tárgy, és a megfelelő jégtáblára kerül.

Use-case neve	Save
Rövid leírás	Játék állapotának elmentése.
Aktorok	User
Forgatókönyv	1. A felhasználó kiadja a save parancsot. 2. A program kiírja az összes objektum adatait a kimenetére.

Use-case neve	Use ability
Rövid leírás	Aktív játékos utasítása képessége használatára.
Aktorok	Felhasználó
Forgatókönyv	1. A felhasználó kiadja a useability parancsot, valamint megadja a szükséges irányt is.

Use-case neve	Lose action point
Rövid leírás	Az aktív játékos akciót pontot veszít.
Aktorok	User, Controller
Főforgatókönyv	1. A játékos veszít egy akciót pontot, de nem az utolsót.
Alternatív forgatókönyv	1.A.1 A játékos elveszti az utolsó akciót pontját. 1.A.2 A következő játékos következik.
Alternatív forgatókönyv	1.A.1.A.1. A játékos elveszti az utolsó akciót pontját, és ő az utolsó a körben. 1.A.1.A.2 Véletlenszerűen lépnek a jegesmedvék és véletlenszerű mezőket érint vihar, majd a következő játékos jön.

Alternatív forgatókönyv	1.A.1.A.2.A.1 A felhasználó megadja az irányokat, amerre a jegesmedvék lépni fognak, a vihar által érintett mezők számát, majd a mezők nevét. 1.A.1.A.2.A.2 A jegesmedvék lépnek a megadott irányba, megtörténik a vihar, majd a következő játékos jön.
--------------------------------	--

7.3 Tesztelési terv

Teszt-eset neve	Next Players turn
Rövid leírás	Egy játékos kifogy az akcióiból, és a következő játékos jön
Teszt célja	Az akciók elfogyásának, és a kontroll átadásának működését teszteli.

Teszt-eset neve	Next Round
Rövid leírás	Egy játékos kifogy az akcióiból, és mivel ő volt az utolsó játékos, a következő kör kezdődik
Teszt célja	Az akciók elfogyásának, kontroll átadásának, és az automatizált események (például fulladás) működését teszteli.

Teszt-eset neve	Move character into instable icefield without divingsuit
Rövid leírás	Egy karakter rásétál egy mezőre, ahol a vízbe esik, és véget ér a köre.
Teszt célja	A vízbe esés mechanikáit teszteli.

Teszt-eset neve	Move character into instable icefield with divingsuit
Rövid leírás	Egy karakter rásétál egy mezőre, ahol biztonságosan a vízbe esik, és nem ér véget a köre.
Teszt célja	A diving suit működését teszteli.

Teszt-eset neve	Move character into stable icefield
Rövid leírás	Egy karakter rásétel egy másik mezőre.
Teszt célja	A mozgás mechanikáit teszteli.

Teszt-eset neve	Storm on icefield dealing lethal damage
Rövid leírás	A vihar lecsap egy mezőre, ahol karakter tartózkodik, és a vihar megöli ezt a karaktert.
Teszt célja	A halálos kihülés mechanikáit teszteli.

Teszt-eset neve	Storm on icefield with players
Rövid leírás	A vihar lecsap egy mezőre, ahol karakter tartózkodik, és a karakter kihül.
Teszt célja	A vihar és a kihülés mechanikáit teszteli.

Teszt-eset neve	Storm on icefield with igloo
Rövid leírás	A vihar lecsap egy mezőre, ahol igloo van, ezért nem hül ki senki.
Teszt célja	Az igloo mechanikáit teszteli.

Teszt-eset neve	Use Reveal ability
Rövid leírás	Egy kutató használja a képességét egy szomszédos mezőn
Teszt célja	A reveal képesség mechanikáit teszteli.

Teszt-eset neve	Use Igloo ability
Rövid leírás	Egy eszkimó használja a képességét a saját mezőjén
Teszt célja	Az Igloo rakás képesség mechanikáit teszteli.

Teszt-eset neve	Dig
Rövid leírás	Egy karakter kézzel ás.
Teszt célja	A ásást teszteli.

Teszt-eset neve	Take Item
Rövid leírás	Egy karakter felvessz egy itemet.
Teszt célja	A item felvezést teszteli.

Teszt-eset neve	Use Rope with drowning players around
Rövid leírás	Egy karakter sikeresen használja a kötelét.
Teszt célja	A kötél sikeres használatát teszteli.

Teszt-eset neve	Use Rope with no drowning players around
Rövid leírás	Egy karakter sikertelenül használja a kötelét.
Teszt célja	A kötél üres használatát teszteli.

Teszt-eset neve	Use Food
Rövid leírás	Egy karakter megesz egy adag élelmet.
Teszt célja	A étel működését teszteli.

Teszt-eset neve	Use Shovel
Rövid leírás	Egy karakter ásásra használja az ásóját.
Teszt célja	Az ásó működését teszteli.

Teszt-eset neve	Use flare gun, everyone is here
Rövid leírás	Egy karakter sikeresen használ egy flare gun darabot.
Teszt célja	A flare gun működését, és a játék vége események kezelését teszteli.

Teszt-eset neve	Use flare gun, somebody is missing
Rövid leírás	Egy karakter sikertelenül használ egy flare gun darabot.
Teszt célja	A flare gun működését teszteli.

Teszt-eset neve	Breakable Shovel Break
Rövid leírás	Egy karakter ás egy törhető ásó segítségével, ami ezzel a használattal össze is törik.
Teszt célja	A törhető ásót teszteli.

Teszt-eset neve	Storm on icefield with tent
Rövid leírás	A vihar lecsap egy mezőre, ahol sátor van, ezért nem hűl ki senki.
Teszt célja	A sátort teszteli.

Teszt-eset neve	Bear meets Meat
Rövid leírás	Egy medve találkozik egy karakterrel, és megeszi.
Teszt célja	A medvét mozgását teszteli.

Teszt-eset neve	Bear meets Meat with Tent
Rövid leírás	Egy medve találkozik egy karakterrel aki sátorban van, és megeszi.
Teszt célja	A sátort teszteli.

Teszt-eset neve	Bear meets Meat with Igloo
Rövid leírás	Egy medve találkozik egy karakterrel aki iglooban van, és ezért nem tudja megenni.
Teszt célja	A medve és az igloo interakcióját teszteli.

Teszt-eset neve	Bear meets Bear
Rövid leírás	Egy medve találkozik egy medvével.
Teszt célja	Az icefield és a medve interakcióját teszteli.

Teszt-eset neve	Bear flips icefield
Rövid leírás	Egy medve rálép egy mezőre ahol játékosok vannak, és az ő súlyával együtt átfordul az icefield.
Teszt célja	Az evés és a vízbe esés által készített helyzetben feljövő folyamatok sorrendjét teszteli.

Teszt-eset neve	Generate level
Rövid leírás	Egy random pálya generálása.
Teszt célja	A generáló algoritmust teszteli.

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A prototípust PowerShell scriptekkel lehet majd tesztelni. Ezeket a scripteket paraméterekkel megadva leghet futtatni.

A *tesztnév*.tst-be kell elhelyezni a teszesetre szánt bemenetet, a kimenetet pedig a *tesztnév*.exp-ből lehet kiolvasni. A szkript minden tesztelésre a tesztelés sikerességéről (vagy annak mértékéről) jelentést fog adni.

7.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.04.01. 22:50	2 óra	Karmacsí	Változások dokumentálása.
2020.04.03. 18:00	2 óra	Bartsch	7.3, 7.4
2020.04.04. 16:00	2 óra	Takács Suciu	7.1
2020.04.05. 20:00	1.5 óra	Takács	7.2
2020.04.05. 20:00	1 óra	Takács Suciu	Bemeneti nyelv pontosítása
2020.04.05. 21:00	0.5 óra	Suciu	Kimeneti nyelv

8. Részletes tervezettségi részletek

36 – press_f

Konzulens:
Ludmány Balázs

Csapattagok

Karmacsi Péter	L94HDE	karmacsi.peter@freemail.hu
Suciú Barnabás	ASOG9J	szubar10@gmail.com
Bartsch Dávid	BXURDI	bartsch.david.a@gmail.com
Seres Ádám	JF204C	seres.adm@gmail.com
Takács Marcell Adrián	QITM0G	marci.takacs@gmail.com

2020.04.13.

8. Részletes tervez

8.1 Osztályok és metódusok tervez.

8.1.1 BreakableShovel

- **Felelősség**

Havat lehet vele eltakarítani a jégmezőről. minden használat csökkenti a durability-t, ha 0-ra csökken, akkor eltörik.

- **Ősosztályok**

Item → Shovel → BreakableShovel

- **Attribútumok**

- **-durability:** int: Az ásó tartóssága.

- **Metódusok**

- **+boolean Use(Player p):** Két egységnyi höréteget takarít el a jégmezőről, ahol a használója áll. Csökkenti eggyel a durability-t. Ha 0-ra csökkent a durability, akkor a p-n meghívja a RemoveItem(bs)-t.
- **+boolean IsTheSame(BreakableShovel: bs):** True-val tér vissza.

8.1.2 DivingGear

- **Felelősség**

A játékos túlél a vízben is kör végén, ha van neki, illetve ki tud jutni a vízből.

- **Ősosztályok**

Item → DivingGear

- **Metódusok**

- **+boolean IsTheSame(DivingGear: d):** True-val tér vissza.
- **+boolean Equip(Player p):** Ugyan azt csinálja, mint az Item, csak pluszba beállítja a játékoson a hasDivingGear-t.

8.1.3 Entity

- **Felelősség**

A jegesmedvék és játékosok ősosztálya. Tud mozogni jégtáblákon, vízbe is eshet róluk, találkozhat más entitásokkal és viharba tud kerülni.

- **Attribútumok**

- **#field: IceField:** A jégtábla, amin az entitás áll.

- **Metódusok**

- **+void CaughtByStorm():** Viharba került az entitás.
- **+void Meet(Entity e):** Találkozik az e entitással.

- **+void FallInWater():** Vízbe esik.
- **+void Move(Direction d):** Átmegy a d irányba lévő jégtáblára.

8.1.4 Eskimo

- **Felelősség**

Képes Iglut építeni, valamint egygyel több testhővel rendelkezik.

- **Ősosztályok**

Entity → Player → Eskimo

- **Metódusok**

- **boolean UseAbility(Direction: d):** Iglut épít arra a IceField-re, ahol éppen áll. True-val tér vissza, ha sikerült.

8.1.5 FlareGunPart

- **Felelősség**

Ezekből kell összegyűjteni mindhármat és használni egy helyen állva a játék megnyeréséhez.

- **Ősosztályok**

Item → FlareGunPart

- **Metódusok**

- **+boolean Use(Player p):** Megnézi, hogy teljesülnek-e a győzelem feltételei és jelez a Game-nek, ha igen. True-val tér vissza, ha sikerült használni.
- **+boolean IsTheSame(FlareGunPart: f):** True-val tér vissza.

8.1.6 Food

- **Felelősség**

Fel lehet használni a játékos testhőjének növelésére.

- **Ősosztályok**

Item → Food

- **Metódusok**

- **+boolean Use(Player p):** A játékos testhőjét növeli. True-val tér vissza.
- **+boolean IsTheSame(Food: f):** True-val tér vissza.

8.1.7 Game

- **Felelősség**

Inicializálja a játékot. Nyilvántartja a játék állapotát. Számon tartja, hogy hányan vannak vízbe esve és hány alkatrésze van kiásva a jelzőpisztolynak. A játék véget ér, ha a kör végén valaki vízbe van esve. minden kör végén lépteti a Steppable interfész megvalósító objektumokat.

- **Attribútumok**

- **-state:** String: A játék állapota (folyamatban van, vége + nyert/vesztett)
- **-playersInWater:** int: A búvárruha nélkül vízbe esett játékosok száma.
- **-gunPartsFound:** int: A jelzőpisztoly megtalált alkatrészeinek száma.
- **-fields:** IceField[]: A jégtáblák.
- **-players:** Player[]: A játékosok.
- **-activePlayer:** player: A soron lévő játékos.
- **-steppables:** Steppable: A körök végén léptetendő objektumok.

- **Metódusok**

- **-void init(int row, int col, int e, int r, int ib):** Inicializálja a játékot row db sorral, col db oszloppal, e db eszkimóval, r db researcher-el és ib db jedesmedvével. A pálya elrendezése majdnem teljesen véletlenszerű.(pl. biztosan lesz olyan jégmező, amire egyszerre rá tud állni mindenki anélkül, hogy felborulna.)
- **-void CreateFields():** Inicializálja a jégtáblákat.
- **-void Storm():** Vihart indít néhány jégtáblán.
- **-void RoundOver():** Végrehajtja a kör végén elvégzendő dolgokat: Véget ér a játék, ha valaki vízbe van esve búvárruha nélkül, vihart indít és lépteti a steppable-ket.
- **+void TryToWin():** Ha ki van ásva minden jelzőpisztoly alkatrész és mindenki egy jégtáblán áll, akkor a játék véget ér és nyertek a játékosok.
- **+void Over(boolean victory):** Befejezi a játékot. Ha a paraméter true, akkor a játékosok nyertek, false esetén vesztettek.
- **+void NextPlayer():** Átállítja, hogy ki az aktuális játékos és átállítja a hátralevő akciót 4-re. Ha vége egy körnek, akkor meghívja a RoundOver()-t.
- **+void PlayerSaved():** Jelzés, hogy egy játékest kimentettek a vízből. Levon egyet a playersInWater értékéből.
- **+void PlayerFellInWater():** Jelzés, hogy egy jétkos vízbe esett. hozzáad egyet a playersInWater értékéhez.
- **+void GunPartFound():** Jelzés, hogy kiástak egy jelzőpisztoly alkatrészt egy jégtáblából. Növeli a gunPartsFound értékét eggyel.

8.1.8 IceBear

- **Felelősség**

Léptetni, lehet, hogy véletlenszerűen átlépjen egy másik szomszédos jégtáblára. Megöli a játékosokat, akikkel találkozik.

- **Ósosztályok**

Entity → IceBear

- **Interfészek**

Steppable

- **Metódusok**

- **+void Meet(Player p):** Találkozik a p játékossal és megöli.
- **+void Step():** Véletlenszerűen választ egy szomszédos mezőt és átlép rá.

8.1.9 IceField

- **Felelősség**

Tárolja a jégtábla adatait (szomszédok, rajta álló játékosok, hószint, befagyott tárgy, iglu). Lehet mozogni közöttük, felborul, ha sokan állnak rajta, ki lehet ásni a tárgyat, ha nincs rajta hó. (A hóval fedett lyuk egy 0 kapacitású jégtábla.)

- **Attribútumok**

- **-neighbours:** IceField: A szomszédos jégtáblák a megadott irányba.
- **-snowLevel:** int: A jégtáblán lévő hőréteg vastagsága.
- **-entities:** Entity[]: A rajta álló entitások..
- **-playersInWater:** Player[]: A mezőről vízbe esett játékosok.
- **-item:** Item: A jégtáblába fagyott tárgy.
- **-capacity:** int: Ennyi entitás tartózkodhat rajta anélkül, hogy felborulna.
- **-capacityRevealed:** boolean: True, ha már látható a jégtábla kapacitása, mert egy Researcher megvizsgálta.
- **-hasIgloo:** boolean: True, ha van rajta iglu.
- **-tent:** Tent: A jégtáblára épített sátor.

- **Metódusok**

- **-void Capsize():** Felborul a jégtábla. Jelez a rajta álló entitásoknak, hogy vízbe estek.
- **+IceField GetNeighbour(Direction d):** Visszaadja a szomszédos jégtáblát a megadott irányba.
- **+void removeSnow(int a):** Csökkenti a hőréteg vastagságát a megadott értékkel.
- **+void Storm():** Növeli a hőréteget és csökkenti a rajta lévő játékosok testhőjét, ha nincs a jégtáblán iglu vagy sátor.
- **+void BuildIgloo():** Iglut épít a jégtáblára
- **+void Accept(Entity e):** Felveszi az entitást a rajta álló entitások közé és találkoztatja a már ott lévő entitásokkal. Felborul ha, túl sokan állnak így már rajta.
- **+boolean Save(IceField f):** Végig hívja a GetSaved(Icefield f)-et a rajta álló játékosokon. True-val tér vissza, ha legalább 1 játékos ki lett mentve.
- **+void Remove(Entity e):** Kiveszi az entitást a rajta álló entitások listájából.
- **+boolean TakeItem(Player p):** A kapott játékosnak kiássa a befagyott tárgyat. True-val tér vissza, ha sikerült.
- **+void AddNeighbour(Direction d, IceField f):** Beállítja kapott irányba lévő szomszédnak a kapott IceFieldet.
- **+void RevealCapacity():** Felfedi, hogy hányan férnek el a jégtáblán anélkül, hogy felborulna. Innentől kezdve végig látszik.
- **+void RemoveTent():** Lerombolja a jégtáblán lévő sárat.
- **+void AddPlayerToWater(Player p):** Hozzáadja a p játékosat a vízbe esett játékosok listájához.

8.1.10 Item

- **Felelősség**

A használható tárgyak heterogén kollekciója.

- **Metódusok**

- **+boolean Use(Player p):** A kapott játékos használja a tárgyat. True-val tér vissza, ha sikerült.
- **+boolean Equip(Player p):** A kapott player felveszi a tárgyat. Ha sikerül True-val tér vissza.
- **+boolean IsTheSame(Item: i):** False-al tér vissza. A leszármazottaknak van egy ugyanilyen nevű függvényük, amiben magukból vesznek át egy példányt paraméterként és azok true-val térnek vissza. Így meg lehet nézni, hogy a kapott item ugyanolyan fajta-e, mint ez.

8.1.11 Player

- **Felelősség**

Nyilvántartja a játékos adatait. A játékos akcióit végrehajtja. Ha elfogynak az akciói, szól a Game-nek, hogy jöhet a következő játékos.

- **Ősosztályok**

Entity → Player

- **-temperature:** int: A játékos jelenlegi testhője
- **-isInWater:** boolean: True, ha vízbe van esve.
- **-remainingActions:** int: A hátralévő akciók száma a játékos körében.
- **-items:** Item[]: A játékos birtokolt tárgyai.
- **-game:** Game: A játék.
- **-hasDivingGear:** True, ha van búvárruhája.

- **Metódusok**

- **+void ChangeTemperature(int dif):** Változtatja a játékos testhőjét a megadott értékkel(nem lehet 0 alá)
- **+boolean GetSaved(IceField: f):** Kimenekítik a vízből a kapott IceField-re, ha vízbe van esve. True-val tér vissza, ha ki lett mentve.
- **+void UseAbility(Direction: d):** Használja a játékos képességét a megadott irányba. Ha null-t vesz át, akkor ott, ahol áll a játékos. Levon egy akciót, ha sikerült.
- **+void Move(Direction: d):** Másik jégtáblára mozog a megadott irányba. Levon egy akciót, ha sikerült.
- **+void UseItem(Item i):** Használja a megadott tárgyat a játékos.
- **+void PickUpItem():** Kiássa a jégből a belefagyott tárgyat. Levon egy akciót, ha sikerült.
- **+void Dig():** Eltakarít egy réteg havat a jégmezőről, ahol áll.
- **+void AddItem(Item i):** Hozzáadja az i-t a játékos tárgyaihoz. Levon egy akciót, ha sikerült.
- **+void PickUpItem():** Kiássa a jégtáblából a befagyott tárgyat.
- **+boolean AddItem(Item i):** A játékos táskájába teszi a tárgyat, ha még nincs neki egy ilyen. True-val tér vissza, ha sikerült.
- **+void FallInWater():** Beállítja az isInWater-t true-ra. Jelez a game-nek a PlayerFallInWater-rel.
- **+void RemoveItem(Item i):** Kiveszi az i tárgyat a player tárgyai közül.
- **+void Die():** Jelez a game-nek, hogy vesztettek Over(false)-al.
- **+void CaughtByStorm():** Csökken a testhője 1-el. Ha 0-ra csökkent, akkor meghívja a Die()-t.

- +void Meet(IceBear i): Találkozik egy medvével és meghívja a Die()-t.

8.1.12 Researcher

- **Felelősség**

Meg tudja nézni egy szomszédos IceField kapacitását.

- **Ősosztályok**

Entity → Player → Researcher

- **Metódusok**

- **boolean UseAbility(Direction: d):** Meg tudja nézni egy szomszédos(vagy ahol áll, ha d null) IceField kapacitását, ha azt már nem nézték meg kötában. True-val tér vissza, ha sikerült.

8.1.13 Rope

- **Felelősség**

Ki lehet menteni a játékosokat a vízből a környező mezőkről.

- **Ősosztályok**

Item → Rope

- **Metódusok**

- **+boolean Use(Player p):** Kimenti a vízbeesett játékosokat a szomszédos mezőkről. True-val tér vissza, ha legalább egy játékos kimentett.
- **+boolean IsTheSame(Rope: r):** True-val tér vissza.

8.1.14 Shovel

- **Felelősség**

Havat lehet vele eltakarítani a jégmezőről.

- **Ősosztályok**

Item → Shovel

- **Metódusok**

- **+boolean Use(Player p):** Két egységes hóréteget takarít el a jégmezőről, ahol a p Player áll.
- **+boolean IsTheSame(Shovel: s):** True-val tér vissza.
- **+Equip(p: Player):** A p játékosnak adja az ásót, ha még nincs neki, vagy csak törhető van.

8.1.15 Steppable

- **Felelősség**

A kör végén lehet léptetni a step() függvénnyel.

- **Metódusok**

- **+boolean Step()**: Ezt kell implementálnia az interfész megvalósító osztályoknak.

8.1.16 Tent

- **Felelősség**

Fel lehet állítani egy mezőn, ahol megvéd a vihartól. A kör végén összeomlik.

- **Ősosztályok**

Item → Tent

- **Interfészek**

Steppable

- **Attribútumok**

- **-field: IceField**: A jégmező, ahol fel van állítva.

- **Metódusok**

- **+boolean Use(Player p)**: Felépíti a sárat azon a jégtáblán, ahol a játékos áll.
- **+boolean IsTheSame(Tent: t)**: True-val tér vissza.
- **+void Step()**: Lerombolja a sárat, ha fel van építve egy IceField-en.

8.2 A tesztek részletes tervez, leírásuk a teszt nyelvén

Megjegyzés: A be- és kimeneti nyelvben a következő változtatásokat eszközöltük:

- A CreateMap helyett létrehoztunk egy CreateField és egy SetNeighbor parancsot, hogy a nem négyzetrács alakú pályát is tudjuk tesztelni
- Létrehoztunk egy MoveBear parancsot a következő szintaxissal:
movebear <név> <irány>
- A UseAbility kapott egy irány paramétert
- Létrehoztunk egy SetTemp parancsot, ami az aktuális játékos testhőjét állítja, a következő szintaxissal:
settemp <érték>
- A field-ek az info parancs hatására kiírják azt is, hogy a kapacitásuk látható-e
- A game az info parancs hatására az aktív játékosot is kiírja

8.2.1 Next Players turn

- **Leírás**

Egy játékos kifogy az akcióiból, és a következő játékos jön

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos fordulójának végén lefutó folyamatok. Hibalehetőségek: A game kontroller játékos menedzselése.

- **Bemenet**

```
createfield f_1_1 1 3
createfield f_1_2 1 3
setneighbor f_1_1 1 f_1_2 3
entity player1 researcher f_1_1
entity player2 researcher f_1_1
setactiveplayer player1
```

8. Részletes tervezettségek

press_f

setactions 1
move 1

info game

- **Elvárt kimenet**

game:

State: running

Playersinwater: 0

Gunpartsfound: 0

Activeplayer: player2

8.2.2 Next Round

- **Leírás**

Egy játékos kifogy az akcióiból, és mivel ő volt az utolsó játékos, a következő kör kezdődik

- **Ellenőrzött funkcionális, várható hibahelyek**

Egy teljes kör végén lefutó folyamatok. Hibalehetőségek: A game kontroller játékos menedzselése, a fulladások helytelen lefutása.

- **Bemenet**

createfield f_1_1 1 3
createfield f_1_2 1 3
setneighbor f_1_1 1 f_1_2 3
entity player1 researcher f_1_1
entity player2 researcher f_1_1
setactiveplayer player2
setactions 1
move 1

info game

- **Elvárt kimenet**

game:

State: running

Playersinwater: 0

Gunpartsfound: 0

Activeplayer: player1

8.2.3 Move character into instable icefield without divingsuit

- **Leírás**

Egy karakter rásétál egy mezőre, ahol a vízbe esik, és véget ér a köre.]

- **Ellenőrzött funkcionális, várható hibahelyek**

Az Icefield limitjének átlépése, és a vízbe esés. Hibalehetőség: Az vízbeesés flag helytelen nem átfordulása.

- **Bemenet**

```
createfield f_1_1 1 3
createfield f_1_2 1 0
setneighbor f_1_1 1 f_1_2 3
entity player1 researcher f_1_1
setactiveplayer player1
setactions 4
move 1
```

```
info player1
info game
```

- **Elvárt kimenet**

player1:

Field: f_1_2

Temperature: 4

Isinwater: true

Remainingactions: 0

Hasdivingear: false

game:

State: ended

Playersinwater: 1

Gunpartsfound: 0

Activeplayer: -

8.2.4 Move character into instable icefield with divingsuit

- **Leírás**

Egy karakter rásétál egy mezőre, ahol biztonságosan a vízbe esik, és nem ér véget a köre.

- **Ellenőrzött funkcionális, várható hibahelyek**

A divingsuit működése. Hibalehetőség: Az vízbeesés flag helytelen átfordulása.

- **Bemenet**

```
createfield f_1_1 1 3
createfield f_1_2 1 0
setneighbor f_1_1 1 f_1_2 3
entity player1 researcher f_1_1
setactiveplayer player1
setactions 4
```

```
additem player1 divinggear
useitem divinggear
move 1
```

```
info player1
info game
```

- **Elvárt kimenet**

player1:

Field: f_1_2

Temperature: 4

Isinwater: true

Remainingactions: 3

Hasdivingear: true

Item1: divinggear

game:

State: running

Playersinwater: 1

Gunpartsfound: 0

Activeplayer: player1

8.2.5 Move character into stable icefield

- **Leírás**

Egy karakter rásétel egy másik mezőre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az Icefield stabilitásnak helyes működése. Hibalehetőség: Az vízbeesés flag helytelen átfordulása.

- **Bemenet**

```
createfield f_1_1 1 3
createfield f_1_2 1 3
setneighbor f_1_1 1 f_1_2 3
entity player1 researcher f_1_1
setactiveplayer player1
setactions 4
move 1
```

```
info player1
```

- **Elvárt kimenet**

player1:

Field: f_1_2

Temperature: 4

Isinwater: false

Remainingactions: 3

Hasdivingear: false

8.2.6 Storm on icefield dealing lethal damage

- **Leírás**

A vihar lecsap egy mezőre, ahol karakter tartózkodik, és a vihar megöli ezt a karaktert.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A game kontroller játékos menedzselése. Hibalehetőség: A game kontroller nem fejezi be a játékot.

- **Bemenet**

createfield f_1_1 2 3

setrandom false

entity player1 researcher f_1_1

setactiveplayer player1

sethealth 1

storm 1

f_1_1

info player1

info f_1_1

info game

- **Elvárt kimenet**

player1:

Field: f_1_1

Temperature: 0

Isinwater: false

Remainingactions: 0

Hasdivingear: false

f_1_1:

Snowlevel: 3

Capacity: 3

Hasigloo: false

Item: -

Entity1: player1

Capacityrevealed: false

game:

State: ended

Playersinwater: 0

Gunpartsfound: 0

Activeplayer: -

8.2.7 Storm on icefield with players

- **Leírás**

A vihar lecsap egy mezőre, ahol karakter tartózkodik, és a karakter kihűl.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A storm működése. Hibalehetőség: A karakter nem hül ki.

- **Bemenet**

createfield f_1_1 2 3

setrandom false

entity player1 researcher f_1_1

entity player2 researcher f_1_1

setactiveplayer player1

sethealth 2

setactiveplayer player2

sethealth 3

storm 1

f_1_1

info player1

info player2

info f_1_1

- **Elvárt kimenet**

player1:

Field: f_1_1

Temperature: 1

Isinwater: false

Remainingactions: 0

Hasdivingear: false

player2:

Field: f_1_1

Temperature: 2

Isinwater: false

Remainingactions: 4

Hasdivingear: false

game:

State: running

Playersinwater: 0

Gunpartsfound: 0

Activeplayer: player2

8.2.8 Storm on icefield with igloo

- **Leírás**

A vihar lecsap egy mezőre, ahol igloo van, ezért nem hűl ki senki.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az igloo működése. Hibalehetőség: A karakter kihűl.

- **Bemenet**

```
createfield f_1_1 2 3
setrandom false
entity player1 eskimo f_1_1
setactiveplayer player1
sethealth 2
setactions 1
useability
storm 1
f_1_1
```

```
info player1
info f_1_1
```

- **Elvárt kimenet**

player1:

Field: f_1_1

Temperature: 5

Isinwater: false

Remainingactions: 4

Hasdivingear: false

f_1_1:

Snowlevel: 3

Capacity: 3

Hasigloo: true

Item: -

Entity1: player1

Capacityrevealed: false

8.2.9 Use Reveal ability

- **Leírás**

Egy kutató használja a képességét egy szomszédos mezőn.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A képesség működése. Hibalehetőség: Helytelen adatok, a rossz mező adatai, rossz formátum.

- **Bemenet**

```
createfield f_1_1 1 3
```

```
createfield f_1_2 2 4
```

```
setneighbor f_1_1 1 f_1_2 3
```

```
entity player1 researcher f_1_1
```

```
setactiveplayer player1
```

```
setactions 4
```

```
useability 1
```

```
info f_1_2
```

- **Elvárt kimenet**

f_1_2:

Snowlevel: 2

Capacity: 4

Hasigloo: false

Item: -

Capacityrevealed: true

8.2.10 Use Igloo ability

- **Leírás**

Egy eszkimó használja a képességét a saját mezőjén.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Egy igloo építése. Hibalehetőség: Nem kerül az Icefieldre igloo, vagy rossz icefieldre kerül igloo.

- **Bemenet**

```

createfield f_1_1 1 3
createfield f_1_2 1 3
setneighbor f_1_1 1 f_1_2 3
entity player1 eskimo f_1_1
setactiveplayer player1
setactions 4
useability 1

```

info f_1_1

- **Elvárt kimenet**

f_1_1:

Snowlevel: 1

Capacity: 3

Hasigloo: true

Item: -

Entity1: player1

Capacityrevealed: false

8.2.11 Dig

- **Leírás**

Egy karakter kézzel ás.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az ásás működése. Hibalehetőség: Nem megfelelően módosul az icefield hómennyisége.

- **Bemenet**

```

createfield f_1_1 3 3
entity player1 researcher f_1_1
setactiveplayer player1
setactions 4
dig

```

info f_1_1

- **Elvárt kimenet**

f_1_1:

Snowlevel: 2

Capacity: 3

Hasigloo: false

Item: -

Entity1: player1

Capacityrevealed: false

8.2.12 Take Item

- **Leírás**

Egy karakter felvesz egy itemet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Egy item felvevése. Hibalehetőség: Az item nem kerül be az inventoryba, és/vagy nem tűnik el az Icefield-ről.

- **Bemenet**

```
createfield f_1_1 0 3
entity player1 researcher f_1_1
setactiveplayer player1
setactions 4
createitem shovel f_1_1
pickupitem

info f_1_1
info player1
```

- **Elvárt kimenet**

f_1_1:

Snowlevel: 0

Capacity: 3

Hasigloo: false

Item: -

Entity1: player1

Capacityrevealed: false

player1:

Field: f_1_1

Temperature: 4

Isinwater: false

Remainingactions: 3

Hasdivingear: false

Item1: shovel

8.2.13 Use Rope with drowning players around

- **Leírás**

Egy karakter sikeresen használja a kötelét.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A rope működése. Hibalehetőség: A fuldokló játékosok water flagje és pozíciója nem megfelelően módosul.

- **Bemenet**

```
createfield f_1_1 0 3
createfield f_1_2 0 0
setneighbor f_1_1 1 f_1_2 3
entity player1 researcher f_1_1
entity player2 researcher f_1_1
setactiveplayer player1
setactions 4
move 1
additem player2 rope
setactiveplayer player2
setactions 4
useitem rope

info player1
info player2
info game
```

- **Elvárt kimenet**

player1:

Field: f_1_1
Temperature: 4
Isinwater: false
Remainingactions: 3
Hasdivingear: false
Item1: rope

player2:

Field: f_1_1
Temperature: 4
Isinwater: false
Remainingactions: 0
Hasdivingear: false

game:

State: running
Playersinwater: 0
Gunpartsfound: 0
Activeplayer: player1

8.2.14 Use Rope with no drowning players around

- **Leírás**

Egy karakter sikeresen használja a kötelét.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A rope működése. Hibalehetőség: A nem fuldokló játékosok pozíciójának módosulása.

- **Bemenet**

```
createfield f_1_1 0 3
entity player1 researcher f_1_1
entity player2 researcher f_1_1
additem player2 rope
setactiveplayer player2
setactions 4
useitem rope
```

```
info player1
info player2
```

- **Elvárt kimenet**

player1:

Field: f_1_1
Temperature: 4
Isinwater: false
Remainingactions: 3
Hasdivingear: false
Item1: rope

player2:

Field: f_1_1
Temperature: 4
Isinwater: false
Remainingactions: 0

Hasdivingear: false

8.2.15 Use Food

- **Leírás**

Egy karakter megesz egy adag élelmet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az étel működésének tesztelése. Hibalehetőség: A testhő nem nő egyel és/vagy nem tűnik el a food az inventoryból.

- **Bemenet**

```
createfield f_1_1 0 3
entity player1 researcher f_1_1
additem player1 food
setactiveplayer player1
sethealth 1
setactions 4
useitem food
```

info player1

- **Elvárt kimenet**

player1:

Field: f_1_1

Temperature: 2

Isinwater: false

Remainingactions: 3

Hasdivingear: false

8.2.16 Use Shovel

- **Leírás**

Egy karakter ásásra használja az ásóját.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az ásó tesztelése. Hibalehetőség: Nem megfelelően módosul az icefield hómennyisége.

- **Bemenet**

```
createfield f_1_1 3 3
entity player1 researcher f_1_1
additem player1 shovel
setactiveplayer player1
setactions 4
useitem shovel
```

info f_1_1

- **Elvárt kimenet**

f_1_1:

Snowlevel: 1

Capacity: 3

Hasigloo: false

Item: -

Entity1: player1

Capacityrevealed: false

8.2.17 Use flare gun, everyone is here

- **Leírás**

Egy karakter sikeresen használ egy flare gun darabot.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A flare gun működése. Hibalehetőség: A game kontroller nem vet véget a játéknak.

- **Bemenet**

```
createfield f_1_1 3 3
entity player1 researcher f_1_1
entity player2 researcher f_1_1
additem player1 flaregunpart
additem player1 flaregunpart
additem player1 flaregunpart
setactiveplayer player1
setactions 4
useitem flaregunpart
```

info game

- **Elvárt kimenet**

game:

State: ended

Playersinwater: 0

Gunpartsfound: 3

Activeplayer: -

8.2.18 Use flare gun, somebody is missing

- **Leírás**

Egy karakter sikertelenül használ egy flare gun darabot.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A flare gun működése. Hibalehetőség: A game kontroller véget vet a játéknak és/vagy helytelenül jelez vissza a flare gun használatában vétett hibáról.

- **Bemenet**

```
createfield f_1_1 3 3
createfield f_1_2 3 3
entity player1 researcher f_1_1
```

```
entity player2 researcher f_1_2
additem player1 flaregunpart
additem player1 flaregunpart
additem player2 flaregunpart
setactiveplayer player1
setactions 4
useitem flaregunpart
```

info game

- **Elvárt kimenet**

game:

State: running

Playersinwater: 0

Gunpartsfound: 3

Activeplayer: player1

8.2.19 Breakable Shovel Break

- **Leírás**

Egy karakter ás egy törhető ásó segítségével, ami ezzel a használattal össze is törik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A törhető ásó törésteszte. Hibalehetőség: Nem tünik el az ásó.

- **Bemenet**

```
createfield f_1_1 5 3
entity player1 researcher f_1_1
additem player1 breakableshovel
setactiveplayer player1
setactions 4
useitem breakableshovel
useitem breakableshovel
useitem breakableshovel
useitem breakableshovel
```

info player1

- **Elvárt kimenet**

player1:

Field: f_1_1

Temperature: 4

Isinwater: false

Remainingactions: 4

Hasdivingear: false

8.2.20 Storm on icefield with tent

- **Leírás**

A vihar lecsap egy mezőre, ahol sátor van, ezért nem hűl ki senki.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A sátor teszje. Hibalehetőség: Sebződik a jégmezőn álló játékos a sátor ellenére.

- **Bemenet**

```
createfield f_1_1 3 3
setrandom false
entity player1 researcher f_1_1
additem player1 tent
setactiveplayer player1
setactions 1
useitem tent
storm 1
f_1_1
```

info player1

- **Elvárt kimenet**

player1:

Field: f_1_1

Temperature: 4

Isinwater: false

Remainingactions: 4

Hasdivingear: false

8.2.21 Bear meets Meat

- **Leírás**

Egy medve találkozik egy karakterrel, és megeszi.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A medve teszje. Hibalehetőség: Nem ér véget a játék és/vagy rosszul mozog a medve.

- **Bemenet**

```
createfield f_1_1 1 3
createfield f_1_2 1 3
setrandom false
setneighbor f_1_1 1 f_1_2 3
entity player1 researcher f_1_1
entity bear1 bear f_1_2
setactiveplayer player1
setactions 1
move 1
```

info game

info bear1

- **Elvárt kimenet**

game:

State: ended

Playersinwater: 0

Gunpartsfound: 0

Activeplayer: -

bear1:

Field: f_1_2

8.2.22 Bear meets Meat with Tent

- **Leírás**

Egy medve találkozik egy karakterrel aki sátorban van, és megeszi.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A medve és a sátor teszje. Hibalehetőség: Nem ér véget a játék és/vagy rosszul mozog a medve.

- **Bemenet**

```
createfield f_1_1 1 3
createfield f_1_2 1 3
setrandom false
setneighbor f_1_1 1 f_1_2 3
entity player1 researcher f_1_1
entity bear1 bear f_1_2
additem player1 tent
setactiveplayer player1
setactions 1
useitem tent
movebear bear1 3
```

info game

info bear1

- **Elvárt kimenet**

game:

State: ended

Playersinwater: 0

Gunpartsfound: 0

Activeplayer: -

bear1:

Field: f_1_1

8.2.23 Bear meets Meat with Igloo

- **Leírás**

Egy medve találkozik egy karakterrel aki iglooban van, és ezért nem tudja megenni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A medve és az igloo teszje. Hibalehetőség: Véget a játék és/vagy rosszul mozog a medve.

- **Bemenet**

```
createfield f_1_1 1 3
createfield f_1_2 1 3
setrandom false
setneighbor f_1_1 1 f_1_2 3
entity player1 eskimo f_1_1
entity bear1 bear f_1_2
setactiveplayer player1
setactions 1
useability 1
movebear bear1 3
```

```
info game
info bear1
```

- **Elvárt kimenet**

game:

State: running

Playersinwater: 0

Gunpartsfound: 0

Activeplayer: player1

bear1:

Field: f_1_1

8.2.24 Bear meets Bear

- **Leírás**

Egy medve találkozik egy medvével.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A medve és a medve találkozása. Hibalehetőség: Történik valami anomália a medve mozgásfüggvényében. (A medve megeszi / megpróbálja megenni a medvét)

- **Bemenet**

```
createfield f_1_1 1 3
createfield f_1_2 1 3
```

```

createfield f_1_3 1 3
createfield f_2_1 1 3
setrandom false
setneighbor f_1_1 1 f_1_2 3
setneighbor f_1_2 1 f_1_3 3
entity player1 eskimo f_2_1
entity bear1 bear f_1_1
entity bear2 bear f_1_3
movebear bear1 3
movebear bear2 1

info bear1
info bear2
info f_1_2

```

- **Elvárt kimenet**

bear1:

Field: f_1_2

bear2:

Field: f_1_2

game:

State: running

Playersinwater: 0

Gunpartsfound: 0

Activeplayer: -

8.2.25 Bear flips icefield

- **Leírás**

Egy medve rálép egy mezőre ahol játékosok vannak, és az ō súlyával együtt átfordul az icefield.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az icefield-medve-ember hármas interakciója. Hibalehetőségek: Az evés és a vízbeesés sorrendje, az medve vízbeesése, az emberek vízbeesése.

- **Bemenet**

```

createfield f_1_1 1 1
createfield f_1_2 1 1
setrandom false
setneighbor f_1_1 1 f_1_2 3
entity player1 researcher f_1_1
entity bear1 bear f_1_2
movebear bear1 3

```

*info player1
info game
info bear1*

- **Elvárt kimenet**

player1:

*Field: f_1_1
Temperature: 4
Isinwater: true
Remainingactions: 0
Hasdivingear: false*

game:

*State: ended
Playersinwater: 0
Gunpartsfound: 0
Activeplayer: -*

bear1:

Field: f_1_1

8.2.26 Test scenario

- **Leírás**

Előre meghatározott teszthelyzet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt helyes lefutása sok, fentebb tesztelt funkcionális egyszerre futtatását teszteli.

Hibalehetőség: Az objektumok közötti helytelen interakció.

- **Bemenet**

*createfield f_1_1 2 1
createfield f_1_2 2 3
createfield f_1_3 0 0*

*createfield f_2_1 0 0
createfield f_2_2 2 3
createfield f_2_3 2 1
createfield f_2_4 0 0*

*createfield f_3_1 2 3
createfield f_3_2 2 3*

```

createfield f_3_3 0 0

createfield f_4_1 0 0
createfield f_4_2 2 1
createfield f_4_3 2 3

setneighbor f_1_1 1 f_1_2 1
setneighbor f_1_1 2 f_2_1 1
setneighbor f_1_1 3 f_3_1 1

setneighbor f_1_2 2 f_1_2 2
setneighbor f_1_2 3 f_2_1 1
setneighbor f_1_2 4 f_2_2 1

setneighbor f_1_3 2 f_2_2 2
setneighbor f_1_3 3 f_2_3 1

setneighbor f_2_1 3 f_2_2 3
setneighbor f_2_1 4 f_3_1 2
setneighbor f_2_1 5 f_3_2 1

setneighbor f_2_2 4 f_2_3 2
setneighbor f_2_2 5 f_3_1 2
setneighbor f_2_2 6 f_3_2 1

setneighbor f_2_3 3 f_2_4 1
setneighbor f_2_3 4 f_3_3 2
setneighbor f_2_3 5 f_4_3 1

setneighbor f_2_4 2 f_4_3 2

setneighbor f_3_1 3 f_3_2 3
setneighbor f_3_1 4 f_4_1 1

setneighbor f_3_2 4 f_3_3 3
setneighbor f_3_2 5 f_4_1 2
setneighbor f_3_2 6 f_4_2 1

setneighbor f_3_3 4 f_4_2 3
setneighbor f_3_3 5 f_4_3 2

setneighbor f_4_1 3 f_4_2 3

setneighbor f_4_2 4 f_4_3 4

entity bear1 bear f_1_2

entity player1 eskimo f_2_3
entity player2 researcher f_4_2
entity player3 researcher f_4_3

createitem flaregunpart f_1_1
createitem flaregunpart f_2_2
createitem flaregunpart f_3_1
createitem shovel f_3_2

```

*setactiveplayer player1
setactions 4
useability 1
move 2*

*setactiveplayer player3
setactions 4
move 4*

movebear bear1 4

*info game
info player1
info player2
info player3*

- **Elvárt kimenet**

game:

State: ended

Playersinwater: 0

Gunpartsfound: 0

Activeplayer: -

player1:

Field: f_2_2

Temperature: 5

Isinwater: false

Remainingactions: 2

Hasdivingear: false

player2:

Field: f_4_2

Temperature: 4

Isinwater: true

Remainingactions: 0

Hasdivingear: false

player3:

Field: f_4_2

Temperature: 4

Isinwater: true

Remainingactions: 0

Hasdivingear: false

8.3 A tesztelést támogató programok tervei

A teszteseteket futtató script egy bemeneti és egy kimeneti fájlt vár el, <tesztnév>.in és <tesztnév>.out formában. A <tesztnév>.in fájl tartalmazza a teszteset esetén elvárt beírandó bemeneteket, a <tesztnév>.out fájl pedig az erre a bemenetre elvárt kimeneteket. A fájloknek a scriptel azonos mappában kell lenniük.

Ha a teszt sikeres, akkor a „Test successfully passed” szöveget írja ki a script a képernyőre, ha nem, akkor pedig azt hogy „Test failed”.

Mindkét esetben kiírja az elvárt és kapott outputot, ebben a formátumban:

Generated output:

<kapott kimenet>

Pre-written output:

<.out fájl tartalma>

A szkript futtatni fogja a programon a <tesztnév>.in fájl tartalmában szereplő függvényhívásokat, majd a programból a kimeneten kapott szöveget összehasonlítja a <tesztnév>.out fájl tartalmával, és az eltéréstől vagy nem eltéréstől függően elkészíti a kiírandó szöveget, majd kiírja.

8.4 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.04.11. 10:00	2 óra	Bartsch	8.2
2020.04.12. 15:00	5 óra	Bartsch	8.2, 8.3
2020.04.12. 16:00	3 óra	Karmacsi	8.1
2020.04.12. 17:00	2 óra	Suciú Takács	Be- és kimeneti nyelv pontosítása, tesztelésnél felmerülő kérdések tisztázása
2020.04.13. 24:00	1 óra	Bartsch	Teszt-scenario dokumentálása

10. Prototípus beadása

36 – press_f

Konzulens:
Ludmány Balázs

Csapattagok

Karmacsi Péter	L94HDE	karmacsi.peter@freemail.hu
Suciú Barnabás	ASOG9J	szubar10@gmail.com
Bartsch Dávid	BXURDI	bartsch.david.a@gmail.com
Seres Ádám	JF204C	seres.adm@gmail.com
Takács Marcell Adrián	QITM0G	marci.takacs@gmail.com

2020.04.29.

10. Prototípus beadása

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
BreakableShovel.java	1 KB	2020.04.24.	A törhető ásó osztálya
ConsoleApp.java	5 KB	2020.04.26.	A ConsoleApp osztály
Direction.java	1 KB	2020.04.12.	A irányok enum-ja
DivingGear.java	1 KB	2020.04.12.	A búvárruha osztálya
Entity.java	1 KB	2020.04.12.	A entitások osztálya
Eskimo.java	1 KB	2020.04.12.	A eszkimók osztálya
FlareGunPart.java	1 KB	2020.04.12.	A jelzőpisztoly alkatrész osztálya
Food.java	1 KB	2020.04.12.	A étel osztálya
Game.java	10 KB	2020.04.12.	A játék osztálya
IceBear.java	2 KB	2020.04.12.	A jegesmedvék osztálya
IceField.java	6 KB	2020.04.12.	A jégmezők osztálya
Item.java	1 KB	2020.04.12.	A tárgyak ősosztálya
Main.java	3 KB	2020.04.12.	A main függvény
Player.java	5 KB	2020.04.12.	A játékos karakterek ősosztálya
Researcher.java	1 KB	2020.04.12.	A kutatók osztálya
Rope.java	1 KB	2020.04.12.	A kötél osztálya
Shovel.java	1 KB	2020.04.12.	A ásó osztálya
Steppable.java	1 KB	2020.04.12.	A Steppable interfész
Tent.java	1 KB	2020.04.24.	A sátor osztálya

10.1.2 Fordítás

1. Győződjünk meg róla, hogy a JDK mappája hozzá van adva a JAVA_HOME környezeti változóhoz, illetve a JDK bin mappája hozzá van adva a PATH-hez.(Ha nincs JAVA_HOME változó, akkor hozza létre és adja hozzá a JDK mappájának elérési útját!)
2. Navigálunk valamelyen paracssorral a kitömörtetett forrásfájlokat tartalmazó mappába.
3. Adjuk ki az alábbi parancsokat(Windows alatt):

```
mkdir .\class
javac -d .\class .\BreakableShovel.java .\ConsoleApp.java
.\Direction.java .\DivingGear.java .\Entity.java
.\Eskimo.java .\FlareGunPart.java .\Food.java .\Game.java
.\IceBear.java .\IceField.java .\Item.java .\Main.java
.\Player.java .\Printable.java .\Researcher.java
.\Rope.java .\Shovel.java .\Steppable.java .\Tent.java
```

(Linux alatt ugyanez, csak “\” helyett “/” karakterekkel.)

10.1.3 Futtatás

Windows alatt adjuk ki a következő parancsot:

```
java -cp .\class proto.Main
```

(Linux alatt ugyanez, csak “\” helyett “/” karakterrel.)

Tesztesetek futtatása:

Windows PowerShell alatt adjuk ki a következő parancsot:

```
.\Test.ps1 <x.in> <x.out>
```

Ahol x.in és x.out az egy teszthez tartozó ki- és bemeneti file-ok.

vagy

```
.\Test.ps1
```

Ekkor az összes teszteset lefut.

(Linux alatt ugyanez, csak “\” helyett “/” karakterrel.)

10.2 Tesztek jegyzőkönyvei

10.2.1 BearFlipsIceField

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:04

10.2.2 BearMeetsBear

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:07

10.2.3 BearMeetsMeat

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:14

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:09
Teszt eredménye	A medve nem ette meg a játékest
Lehetséges hibaok	Rossz függvényfejlécek/paraméterek miatt nem a megfelelő függvényváltozat futott le.
Változtatások	Függvényfejlécek átírása.

10.2.4 BearMeetsMeatWithIgloo

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:16

10.2.5 BearMeetsMeatWithTent

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:19

10.2.6 BreakableShovelBreak

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:21

10.2.7 Dig

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:24

10.2.8 MoveCharacterIntoInstableIcefieldWithDivingsuit

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:31

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:27
Teszt eredménye	Vízbe esett, mikor nem kéne.
Lehetséges hibaok	Nem vette figyelembe, hogy van búvárruhája.
Változtatások	A hasDivingGear ellenőrzése.

10.2.9 MoveCharacterIntoStableIcefieldWithoutDivingsuit

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:35

10.2.10 MoveCharacterIntoStableIceField

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:39

10.2.11 NextPlayersTurn

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:47

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:42
Teszt eredménye	Nem állította át a következő játékos akciópontjait 4-re.
Lehetséges hibaok	Hiányos függvény.
Változtatások	Függvény kiegészítése.

10.2.12 NextRound

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:50

10.2.13 StormOnIcefieldDealingLethalDamage

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:59

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 15:54
Teszt eredménye	A játék állapota "running"
Lehetséges hibaok	Az Over nem állítja át az állapotot.
Változtatások	Over() kiegészítése.

10.2.14 StormOnIcefieldWithIgloo

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 16:02

10.2.15 StormOnIcefieldWithPlayers

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 16:06

10.2.16 StormOnIcefieldWithTent

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 16:15

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 16:10
Teszt eredménye	A testhő csökkent, mikor nem kéne.
Lehetséges hibaok	A sátor használatkor nem kerül a jégtáblára.
Változtatások	A sátor Use függvénye.

10.2.17 TakeItem

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 16:18

10.2.18 TestScenario

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 16:21

10.2.19 UseFlareGunEveryonelsHere

Tesztelő neve	Suci Barnabás
Teszt időpontja	2020.04.28 14:00
Teszt eredménye	Nullpointerexceptions-t dob.
Lehetséges hibaok	A FlareGunPart-nak nem állítottuk be a game attribútumát.
Változtatások	ConsoleApp módosítása.

Tesztelő neve	Karmacs Péter
Teszt időpontja	2020.04.27 16:24
Teszt eredménye	Nullpointerexceptions-t dob.
Lehetséges hibaok	Nem állít be valamit a ConsoleApp.
Változtatások	-

10.2.20 UseFlareGunSomebodyIsMissing

Tesztelő neve	Suci Barnabás
Teszt időpontja	2020.04.28 14:00
Teszt eredménye	A game gunpartsfound számlálója nem jó értéket ad.
Lehetséges hibaok	Az additem parancsra a Player AddItem-je hívódik, nem az Equip.
Változtatások	ConsoleApp módosítása.

Tesztelő neve	Karmacs Péter
Teszt időpontja	2020.04.27 16:35
Teszt eredménye	Nullpointerexceptions-t dob.
Lehetséges hibaok	Nem állít be valamit a ConsoleApp.
Változtatások	-

10.2.21 UseFood

Tesztelő neve	Karmacs Péter
Teszt időpontja	2020.04.27 16:38

10.2.22 UselglooAbility

Tesztelő neve	Karmacs Péter
Teszt időpontja	2020.04.27 16:42

10.2.23 UseRevealAbility

Tesztelő neve	Karmacs Péter
Teszt időpontja	2020.04.27 16:44

10.2.24 UseRopeWithDrowningPlayersAround

Tesztelő neve	Karmacs Péter
----------------------	---------------

Teszt időpontja	2020.04.27 17:17
------------------------	------------------

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 17:10
Teszt eredménye	Nem menti ki a játékosat a kötél.
Lehetséges hibaok	A kötél Use függvénye hibás.
Változtatások	Use függvény javítása.

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 16:50
Teszt eredménye	Nem menti ki a játékosat a kötél.
Lehetséges hibaok	A player Save függvénye nem állítja be az isInWater-t false-re.
Változtatások	Save függvény javítása.

10.2.25 UseRopeWithNoDrowningPlayersAround

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 17:20

10.2.26 UseShovel

Tesztelő neve	Karmacsi Péter
Teszt időpontja	2020.04.27 17:23

10.3 Értékelés

Tag neve	Tag neptun	Munka százalékban
Karmacsi Péter	L94HDE	20
Bartsch Dávid Anasztáz	BXURDI	20
Seres Ádám	JF204C	20
Takács Marcell Adrián	QITM0G	20
Suciú Barnabás	ASOG9J	20

10.4 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.14.23. 18:00	4 óra	Karmacsí	Game osztály, 10.1
2020. 04. 27. 01:00	1 óra	Seres	Item osztály és leszármazottai
2020. 04. 27. 11:00	2 óra	Suciú	ConsoleApp egy része, Entity és leszármazottai
2020. 04. 27. 6:30	2 óra	Seres	Item osztály és leszármazottai
2020.04.27 15:00	2 óra	Karmacsí	Tesztelés
2020.04.28. 13:00	2 óra	Suciú, Karmacsí	Hibakeresés, tesztelés
2020. 04. 28. 21:00	4 óra	Seres, Karmacsí, Suciu	Hibakeresés, tesztelés

11. Grafikus felület specifikációja

36 – press_f

Konzulens:
Ludmány Balázs

Csapattagok

Karmacsi Péter	L94HDE	karmacsi.peter@freemail.hu
Suciú Barnabás	ASOG9J	szubar10@gmail.com
Bartsch Dávid	BXURDI	bartsch.david.a@gmail.com
Seres Ádám	JF204C	seres.adm@gmail.com
Takács Marcell Adrián	QITM0G	marci.takacs@gmail.com

2020.05.04.

11. Grafikus felület specifikációja

11.1 A grafikus interfész

A pálya megrajzolása során a következő ikonokat fogjuk használni:



Jégmező



Kutató



Eszkimó



Búvár (Kutató vagy Eszkimó búvárruhában)



Jegesmedve



Táska



Étel



Flare Gun



Ásó



Törhető ásó



Iglu



Sátor



Kötél

Ez alapján a példa pálya így fog kinézni:



Természetesen a pálya tényleges mérete változtatható, de a minimum mérete az 4x4 lesz, ami egyenként egy 100x100 jégmezőt és a rajta lévő elemeket tartalmazza, ami lehet valamilyen fajta entitás (kutató, eszkimó, medve, búvár), táska, ami jelzi, hogy van-e tárgy a mezőn (Ha rávisszük az egeret, akkor meg tudjuk tekinteni a tárgy típusát), iglu/sátor, illetve az esetlegesen vízbe esett játékosok.

A fentebb említett pálya az ablak közepén fog helyet foglalni, mellette jobb oldalt 2 információs panel lesz látható egymás alatt. A felső az épp aktív játékosról fog információkat közölni. Az aktív játékos nevét, maximum és maradék akció pontjai számát, az életét és a nála lévő tárgyakat. Az alsó arról a jégmezőről fog információt szolgáltatni, amire rá visszük az egerünket. Ezek a jégmező kapacitása, a mezőn lévő hó mennyisége, illetve a mezőn lévő tárgy típusa.

A játékosokat ez a menü fogja fogadni, ha elindítják a programot, innen kezdhetnek új játékot vagy léphetnek ki a programból.



Az új játék gomb megnyomása után ebben a menüben adhatjuk meg az elkövetkezendő játék kezdeti adatait. A pálya méretét, az eszkimók, a kutatók és a jegesmedvék számát.



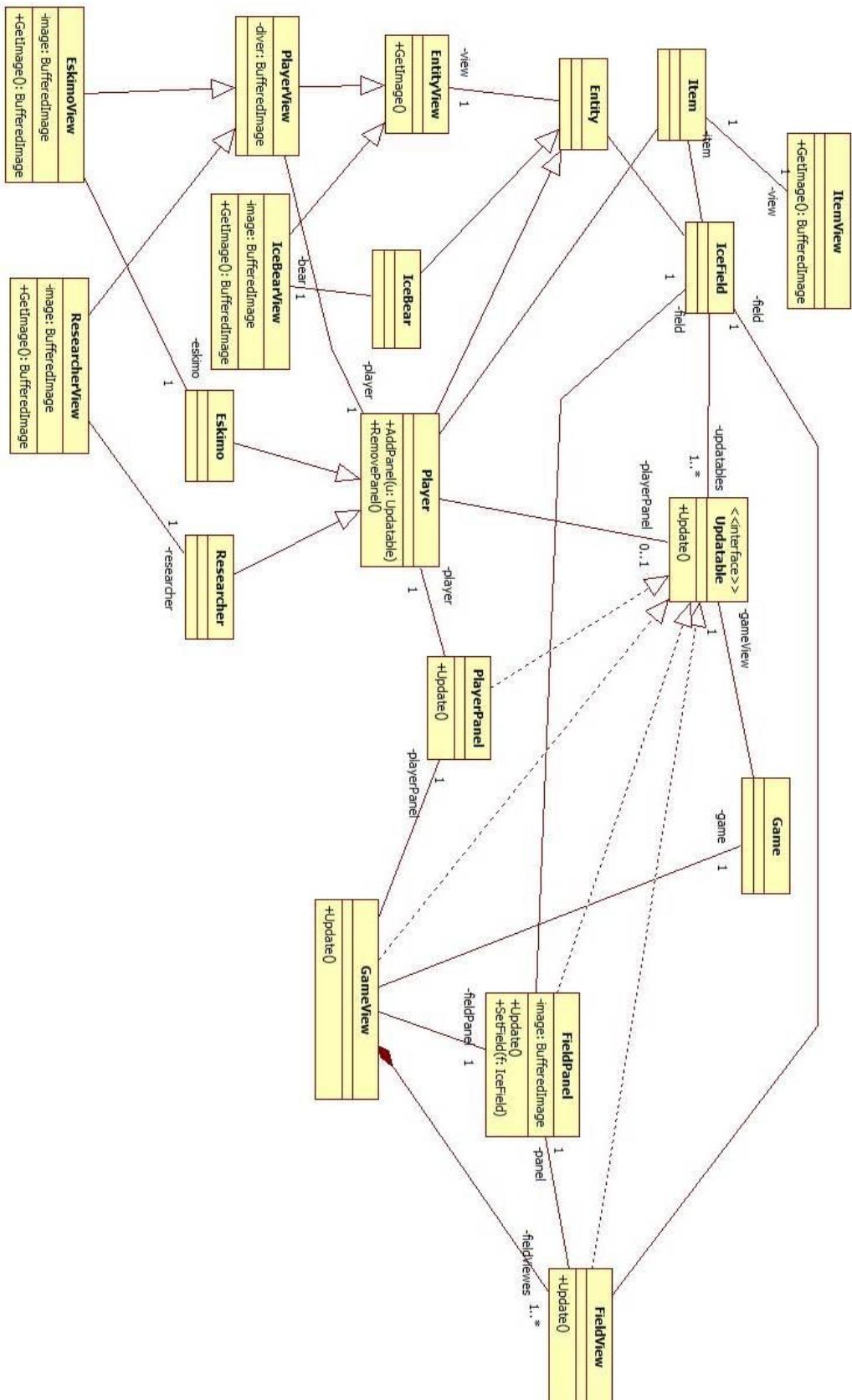
11.2 A grafikus rendszer architektúrája

A felület tervezése során az MVC tervezési mintát igyekeztünk követni.

11.2.1 A felület működési elve

A grafikus felület elsősorban push alapú megoldásokat használ. A modellbeli osztályokhoz tartozik egy View osztály, ami megvalósítja az Updatable interfacet, és amikor változás történik a modellben (például változik a Player hőmérdéklete, eltűnik hó az IceFieldről), akkor a modellbeli osztály szól a View osztályának, hogy frissítse magát. Ennek hatására a View osztály lekérdezi az adatokat, amiket meg kell jelenítenie, majd kirajzolja.

11.2.2 A felület osztály-struktúrája



Az osztálydiagramon az attribútumokhoz tartozó implicit getter és setter metódusokat nem ábrázoltuk.

11.3A grafikus objektumok felsorolása

11.3.1 Updatable

- **Felelősség**

Interfész, melyet a megtekinthető objektumoknak meg kell valósítaniuk, hogy a grafikát kezelő osztályok kommunikálhassanak velük.

- **Ősosztályok**
 -

- **Interfészek**
 -

- **Attribútumok**
 -

- **Metódusok**
 - **Update(): void:** Az a függvény, ami az interfacet megvalósító osztályok frissíthetőségeért felelős. Ezt meghívva a grafikus felület újrarajzolja az objektumot a friss adatokkal.

11.3.2 Game

- **Felelősség**

Inicializálja a játékot. Nyilvántartja a játék állapotát. minden kör végén vihart indít és ellenőrzi a játékosok állapotát.

- **Ősosztályok**
 -

- **Interfészek**
 -

- **Attribútumok**
 - -gameview: Updatables

- **Metódusok**
 -

11.3.3 GameView

- **Felelősség**

A fő View osztály, amely felelős a játék fő eseményeinek grafikus kezeléséért, valamint tárolja a playerpanelt, fieldpanelt, és az összes fieldview-t.

- **Ősosztályok**
 -

- **Interfészek**

Updatable

- **Attribútumok**

- -game: Game
- -List<FieldView>: fieldViews
- -PlayerPanel: playerPanel
- -FieldPanel: fieldPanel

- **Metódusok**

- **+Update(): void:** A játék aktuális állapotától kiírja, hogy ha vége a játéknak, vagy aktív játékos változásakor frissíti a playerpanelt az új játékossal.

11.3.4 IceField

- **Felelősség**

Tárolja a jégtábla adatait (szomszédok, rajta álló játékosok, hószint, befagyott tárgy, iglu). Lehet mozogni közöttük, felborul, ha sokan állnak rajta, ki lehet ájni a tárgyat, ha nincs rajta hó. (A hóval fedett lyuk egy 0 kapacitású jégtábla.)

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- -updatables: Updatable

- **Metódusok**

-

11.3.5 FieldView

- **Felelősség**

Megjeleníti az Icefield-et és az azon történt változásokat.

- **Ősosztályok**

-

- **Interfészek**

Updatables

- **Attribútumok**

- -field: IceDield

- **Metódusok**

- **+Update(): void:** Amikor valami változik az icefielden, akkor lekérdezi a friss adatokat a modelltől, és kirajzolja ezeket.

11.3.6 FieldPanel

- **Felelősség**

Jégtáblák adatait jeleníti meg a pálya melletti információs panelen.

- **Ősosztályok**

- **Interfészek**

Updatables

- **Attribútumok**

- -field: Icefield
- -image: BufferedImage

- **Metódusok**

- **+Update(): void:** A kijelölt icefield friss adatait írja ki.
- **+SetField(f: Icefield): void:** A kiválasztott mező változásakor beállítja a fieldpanel által tárolt fieldet a paraméterben kapott icefielddel, majd hív egy Update()-et.

11.3.7 PlayerPanel

- **Felelősség**

Az aktív játékos adatait jeleníti meg a pálya melletti információs panelen.

- **Ősosztályok**

- **Interfész**

Updatables

- **Attribútumok**

- -player: Player

- **Metódusok**

- **Update(): void:** Az eltárolt játékos lekérdezett friss adatait jeleníti meg.

11.3.8 Player

- **Felelősség**

Nyilvántartja a játékos adatait. A játékos akciót végrehajtja. Ha elfogynak az akciói, szól a Game-nek, hogy jöhet a következő.

- **Ősosztályok**

Entity

- **Interfészek**

-

- **Attribútumok**

- -playerPanel: Updatable

- **Metódusok**

- **+AddPanel(u: Updatables): void:** Kap egy Updatable interfacet megvalósító objektumot, ami egy PlayerPanel lesz, hogy tudja a köre közben frissíteni a playerpanel által kijelzett adatokat.
- **+RemovePanel(): void:** Amikor vége a játékos körének, akkor ezt meghívva nullra állítjuk a Player által eltárolt Updatable-t.

11.3.9 PlayerView

- **Felelősség**

Eltárolja a búvárruhás játékost ábrázoló képet.

- **Ősosztályok**

EntityView

- **Interfész**

-

- **Attribútumok**

- -diver: BufferedImage
- -player: Player

- **Metódusok**

-

11.3.10 IceBear

- **Felelősség**

Léptetni, lehet, hogy véletlenszerűen átlépjen egy másik szomszédos jégtáblára. Megöli a játékosokat, akikkel találkozik.

- **Ősosztályok**

Entity

- **Interfész**

-

- **Attribútumok**

-

- **Metódusok**
 -

11.3.11 IceBearView

- **Felelősség**

A jegesmedvék megjelenítésével foglalkozik.

- **Ősosztályok**

EntityView

- **Interfészek**
 -

- **Attribútumok**

- -image: BufferedImage
 - -bear: IceBear

- **Metódusok**

- +GetImage(): BufferedImage: Visszatér az eltárolt képpel.

11.3.12 Entity

- **Felelősség**

A jegesmedvék és játékosok ősosztálya. Tud mozogni jégtáblákon, vízbe is eshet róluk, találkozhat más entitásokkal és viharba tud kerülni.

- **Ősosztályok**
 -

- **Interfész**
 -

- **Attribútumok**

- -view: EntityView

- **Metódusok**
 -

11.3.13 EntityView

- **Felelősség**

- **Ősosztályok**
 -

- **Interfészek**
 -
- **Attribútumok**
 -
- **Metódusok**
 - **GetImage(): void:** A leszármazottaknak ezt a függvényt kell megvalósítaniuk.

11.3.14 Eskimo

- **Felelősség**

Képes Iglut építeni, valamint egygyel több testhővel rendelkezik.

- **Ősosztályok**

Player

- **Interfész**
 -
- **Attribútumok**
 -
- **Metódusok**
 -

11.3.15 EskimoView

- **Felelősség**

Az eszkimók megjelenítésével foglalkozik.

- **Ősosztályok**

PlayerView

- **Interfész**
 -
- **Attribútumok**
 - -eskimo: Eskimo
- **Metódusok**
 - **GetImage(): BufferedImage:** Visszatér az eltárolt képpel.

11.3.16 Researcher

- **Felelősség**

Meg tudja nézni egy szomszédos IceField kapacitását.

- **Ősosztályok**

Player

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

-

11.3.17 ResearcherView

- **Felelősség**

Kutatók megjelenítésével foglalkozik.

- **Ősosztályok**

Playerview

- **Interfészek**

-

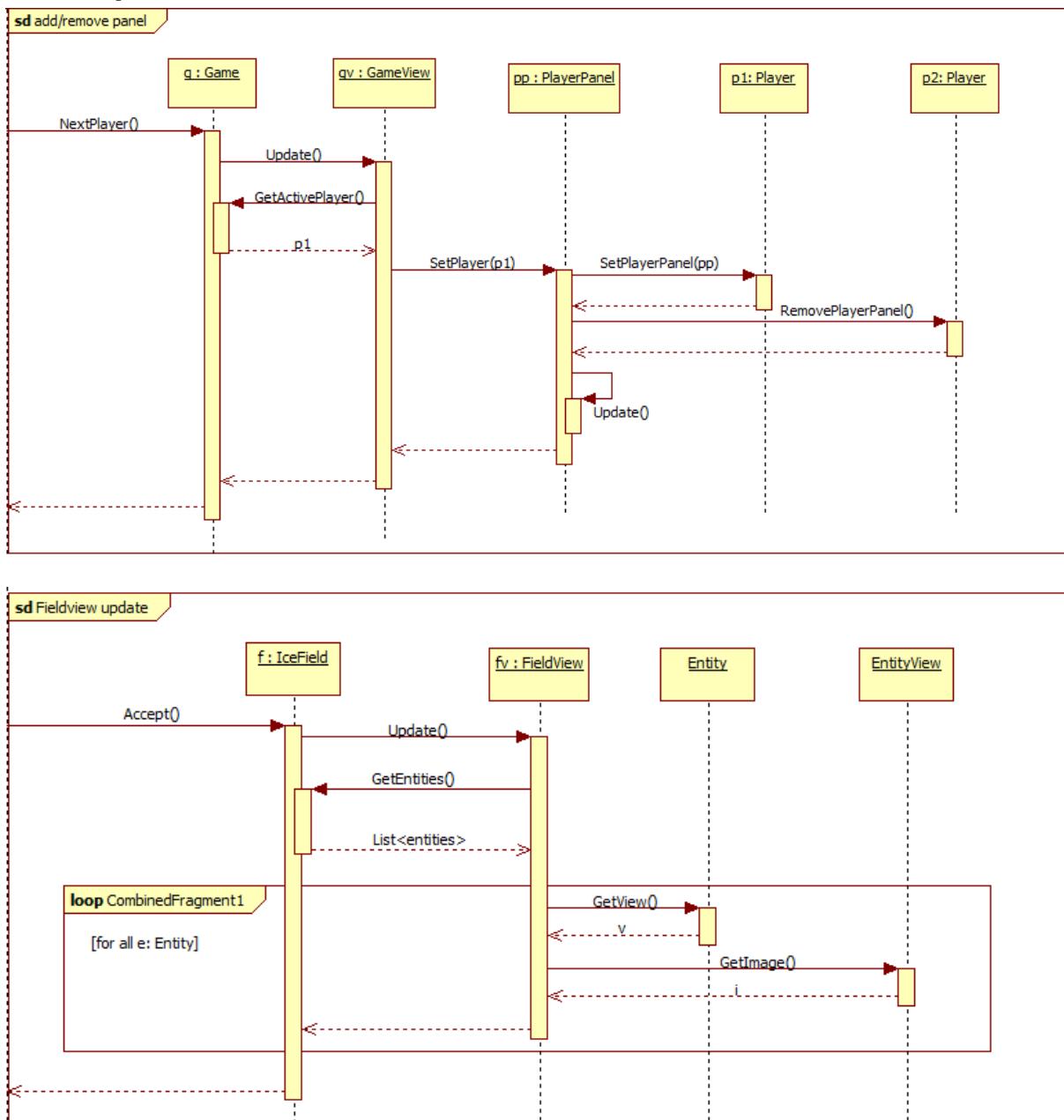
- **Attribútumok**

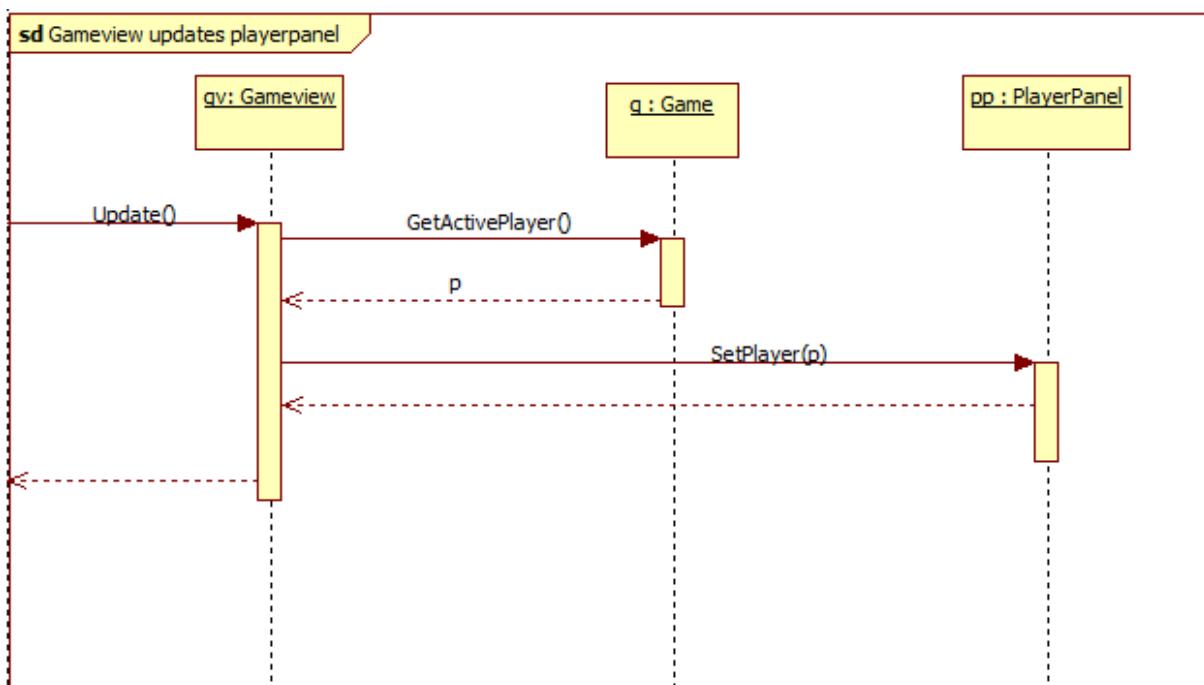
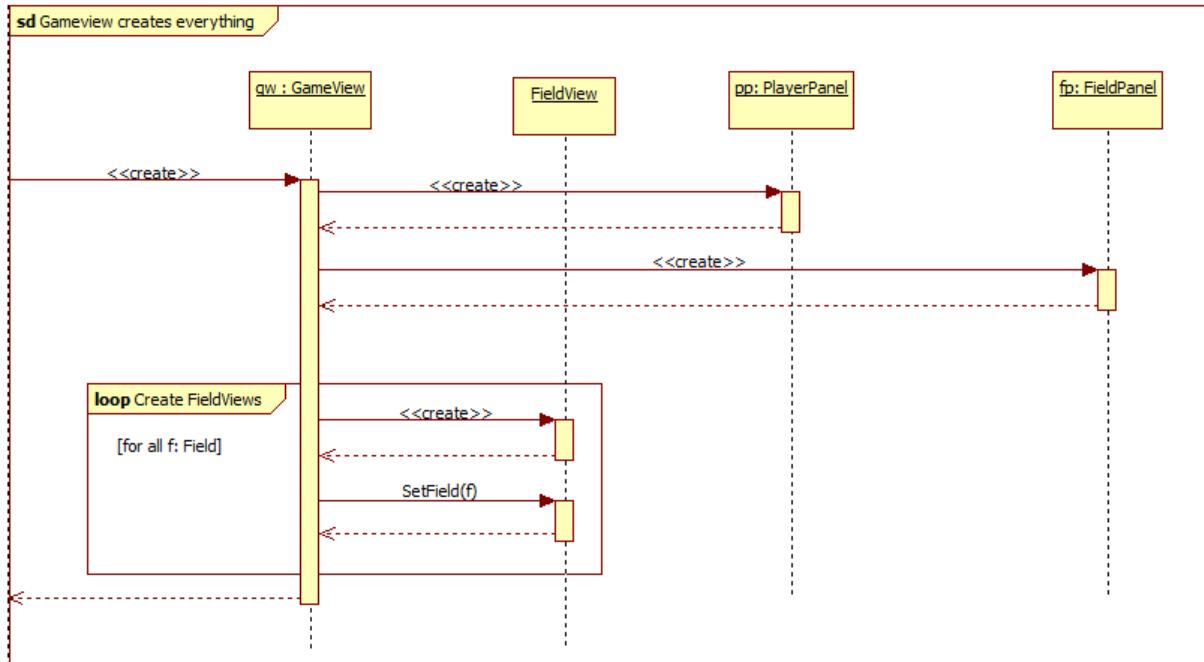
- -image: BufferedImage
- -researcher: Researcher

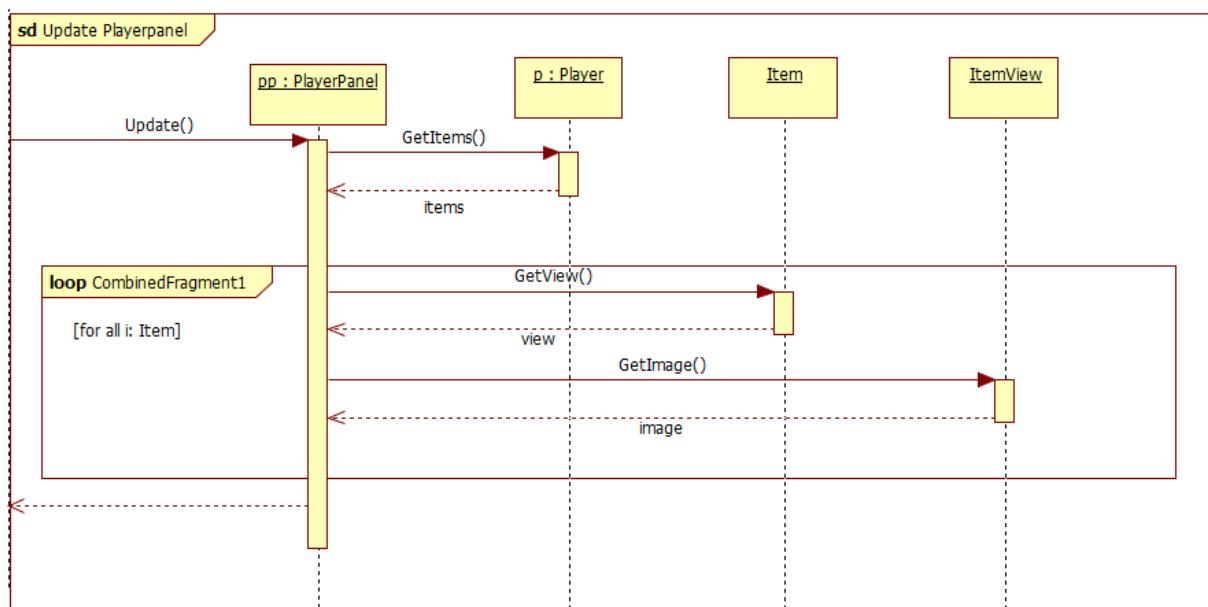
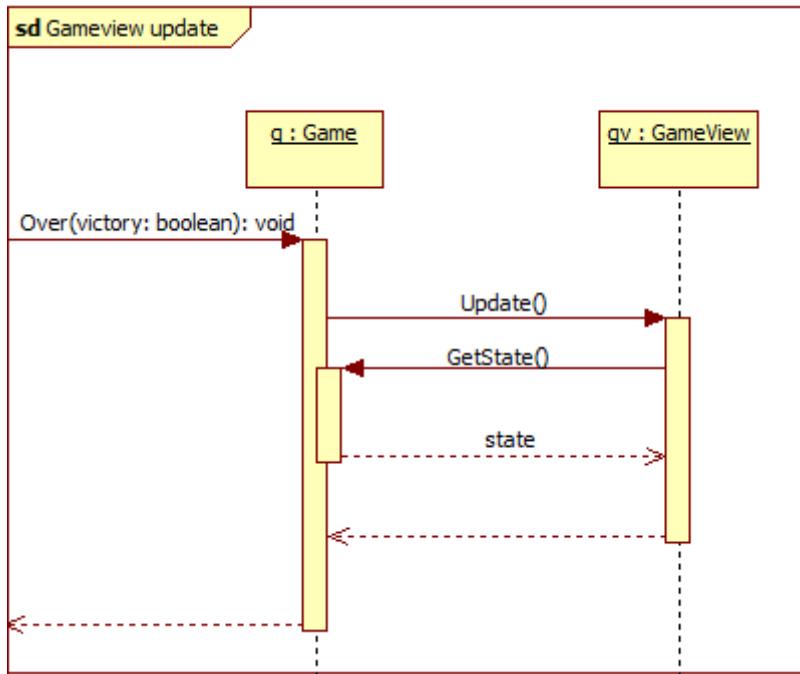
- **Metódusok**

- +GetImage(): BufferedImage: Visszatér az eltárolt képpel.

11.4 Kapcsolat az alkalmazói rendszerrel







11.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.05.02. 12:00	6 óra	Bartsch	Képek készítése
2020.05.03. 12:00	2 óra	Takács Suciu	Osztálydiagram tervezése
2020.05.04. 18:00	2 óra	Karmacsi Seres Takács Suciu	Megbeszélés: osztálydiagram pontosítása, tervezése
2020.05.04. 10:00	4 óra	Karmacsi Seres Takács Suciu	Szekvenciák készítése, metódusok leírása

13. Grafikus változat beadása

13.1 Fordítási és futtatási útmutató

13.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Area.java	1 KB	2020.05.02	Terület adatai
BreakableShovel.java	1 KB	2020.05.02	Törhető ásó
Direction.java	1 KB	2020.05.02	Irány enumeráció
DivingGear.java	1 KB	2020.05.02	Búvárruha
Entity.java	1 KB	2020.05.02	Ember/medve ősosztály
EntityView.java	1 KB	2020.05.02	A fenti megjelenítése
Eskimo.java	1 KB	2020.05.02	Eszkimó
EskimoView.java	3 KB	2020.05.02	A fenti megjelenítése
FieldPanel.java	5 KB	2020.05.02	Kijelölt field adatai
FieldView.java	4 KB	2020.05.02	A fenti megjelenítése
FlareGunPart.java	1 KB	2020.05.02	Flare gun
Food.java	1 KB	2020.05.02	Étel
Game.java	11 KB	2020.05.02	Játék-kontroller
GameView.java	4 KB	2020.05.02	A fenti megjelenítése
IceBear.java	2 KB	2020.05.02	Jegesmedve
IceBearView.java	1 KB	2020.05.02	A fenti megjelenítése
IceField.java	7 KB	2020.05.02	Jégtábla adatai
Item.java	1 KB	2020.05.02	Tárgy ősosztály
ItemView.java	1 KB	2020.05.02	A fenti megjelenítése
MainMenu.java	2 KB	2020.05.02	Főmenü
MainWindow.java	1 KB	2020.05.02	Az alkalmazás ablaka
NewGameMenu.java	5 KB	2020.05.02	Új játék menüelem
Player.java	5 KB	2020.05.02	Eszkimó/Kutató ősosztály
PlayerPanel.java	7 KB	2020.05.02	Felhasználói felület
PlayerView.java	1 KB	2020.05.02	Ember megjelenítése
Researcher.java	1 KB	2020.05.02	Kutató
ResearcherView.java	3 KB	2020.05.02	A fenti megjelenítése
Rope.java	1 KB	2020.05.02	Kötél
Shovel.java	1 KB	2020.05.02	Ásó
Steppable.java	1 KB	2020.05.02	Automatikus játékakciók interface
Tent.java	1 KB	2020.05.02	Sátor
Updatable.java	1 KB	2020.05.02	Grafikai frissítés interface

13.1.2 Fordítás és telepítés

1. Győződjünk meg róla, hogy a JDK mappája hozzá van adva a JAVA_HOME környezeti változóhoz, illetve a JDK bin mappája hozzá van adva a PATH-hez.(Ha nincs JAVA_HOME változó, akkor hozza létre és adja hozzá a JDK mappájának elérési útját!)
2. Navigálunk valamelyen paracssorral a kitömörtetett forrásfájlokat és a képeket tartalmazó images mappát tartalmazó mappába.
3. Adjuk ki az alábbi parancsokat(Windows alatt):


```
mkdir .\class
javac -d .\class *.java
```

(Linux alatt ugyanez, csak “\” helyett “/” karakterekkel.)

13.1.3 Futtatás

Windows alatt adjuk ki a következő parancsot:

```
java -cp .\class graphic.MainWindow
```

(Linux alatt ugyanez, csak “\” helyett “/” karakterrel.)

13.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Karmacsi Péter	L94HDE	20
Suciú Barnabás	ASOG9J	20
Bartsch Dávid	BXURDI	20
Seres Ádám	JF204C	20
Takács Marcell	QITM0G	20

13.3 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.05.16. 18:00	3 óra	Karmacsi	Menük, GameView kezdet
2020.05.18. 11:00	6 óra	Suciu	Grafikus felület implementációja
2020.05.18. 15:00	8 óra	Seres	PlayerPanel gombok, tesztelés, bugfix
2020.05.18. 16:00	2 óra	Bartsch	Dokumentum készítése
2020.05.19. 12:00	2.5 óra	Seres	Tesztelés, bugfix
2020.05.18. 11:00	6 óra	Takács	Grafikus felület implementációja
2020.05.19. 11:00	2 óra	Takács	Tesztelés, bugfix, storm animáció

14. Összefoglalás

14.1 A projektre fordított összes munkaidő

Tag neve	Munkaidő (óra)
Karmacsi Péter	50
Suciú Barnabás	48
Bartsch Dávid	46,5
Seres Ádám	44,5
Takács Marcell	52,5
Összesen	241,5

- A feltöltött programok forrássorainak száma**

Fázis	Kódsorok száma
Szkeleton	1558
Prototípus	1526
Grafikus változat	2489
Összesen	5573

14.2 Projekt összegzés

14.2.1 Mit tanultak a projektből konkrétan és általában?

- Powershell script
- Csapatmunka
- Git Extensions, és a verziókezelés szépségei úgy általában

14.2.2 Mi volt a legnehezebb és a legkönnyebb?

Nehéz:

- A program logikájának a megtervezése, de jó kihívás volt, sokat tanul belőle az ember.
- Csapatnevet kitalálni

Könnyű:

- Dokumentációk írása, különösen az ismétlődő részek (osztályok leírásai)

14.2.3 Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?

- Az idő igen, a pontszám a skeleton és a prototípus feladatoknál nem annyira, túl sok pont járt rájuk, az érdemi logikai tervezésre pedig néha kevesebb.

14.2.4 Ha nem, akkor hol okozott ez nehézséget?

14.2.5 Milyen változtatási javaslatuk van?

- Egy hivatalos példa a dokumentációra, vagy pontosabb dokumentáció követelmény leírás
- Több logikai és kódolós rész, kevesebb skeleton és prototípus készítés

14.2.6 Milyen feladatot ajánlanának a projektre?

- XCOM(lite) játék (körökre osztott stratégia)
- Monopoly játék
- Rizikó játék

14.2.7 Egyéb kritika és javaslat