

14. Összegző leadás

8 – aranyaszok

Konzulens:
Fetter László

Csapattagok

Bodó István
Gaál Gergely
Nagy Márton
Rédai Attila
Varga Levente

LU93RR LU93RR@neptunkod.hszk.bme.hu
GOXEHI GOXEHI@neptunkod.hszk.bme.hu
DMMP94 DMMP94@neptunkod.hszk.bme.hu
LYYOST LYYOST@neptunkod.hszk.bme.hu
JIE1OY JIE1OY@neptunkod.hszk.bme.hu

2020-05-18

2. Követelmény, projekt, funkcionális

2.1 Bevezetés

2.1.1 Cél

Jelen dokumentum segít megérteni, hogy előzetesen pontosan mit is várunk el majd a kész projekttől. Specifikálja azokat a jellemzőket, amiket a feladatkiírás nem tisztázott egyértelműen és leírja a majd elkészült projektet futtató környezettel kapcsolatos elvárásokat. Továbbá segít behatárolni a fejlesztés menetét és meghatároz olyan tulajdonságokat, melyektől a kész szoftver nem térhet el.

2.1.2 Szakterület

A kész szoftver egy játék lesz, így nem sorolhatjuk be egyik szakterületre sem. A játékok közül talán a stratégiai játékokhoz fog hasonlítani leginkább. Elvárásaink szerint az elkészült szoftver szórakozási és időtöltési lehetőséget nyújt majd egyszerre akár több ember számára is.

2.1.3 Definíciók, rövidítések

BME = Budapesti Műszaki És Gazdaságtudományi Egyetem

GitHub = A Git verziókezelő rendszerre épülő internetes szolgáltatás

Háttértár = Tárhely, a számítógép kikapcsolása után is megőrzi az adatokat

IIT = Irányítástechnika és Informatika Tanszék

JRE8 = Java Runtime Environment 8 rövidítése. Szükséges a Java-ban írt programok futtatásához

Szoftver = futtatható program

Verziókezelő = Egy adott fájl változásának követését segíti elő. Lehetővé teszi, hogy egy fájl korábbi verziót visszaállítsuk

2.1.4 Hivatkozások

BME IIT által tartott Szoftver projekt laboratórium, Szoftvertechnológia és A programozás alapjai 3 nevű tárgyak oldalait használjuk többek között ezen dokumentum készítésekor forrásként. Ezen tantárgyak honlapjai az alábbi linkről indulva érhetők el.

<https://www.iit.bme.hu/targyak/BMEVIIIAB02>

<https://www.iit.bme.hu/targyak/BMEVIIIAB01>

<https://www.iit.bme.hu/targyak/BMEVIIIAB00>

A dokumentum szerkesztéséhez továbbá segítségül hívjuk a Google online tárhely szolgáltatását, a Google Drive-ot, melynek segítségével a csapat minden tagja akár egy időben is hozzáfér és szerkesztheti azt. Ez a következő linken érhető el.

<https://www.google.com/drive/>

A későbbiekben, vagyis a projekt előrehaladtával, valószínüleg élünk a GitHub online fájlmegosztó és verziókezelő szolgáltatása nyújtotta lehetőségekkel is.

<https://github.com>

2.1.5 Összefoglalás

A dokumentum alább található részeiben pontosítjuk, hogy kiknek és milyen célra szánjuk a majd elkészülő projektet, hogy precízen milyen követelményeket támasztunk a kész szoftverrel szemben és pontosítjuk a feladat kiírásában nem egyértelműen meghatározott tulajdonságokat. Megszabjuk várhatóan milyen, a rendszerre vonatkozó követelmények szükségesek a projekt fejlesztéséhez és a kész program futtatásához. Továbbá definiálunk egy egységes jelentésrendszert, (egy szótárat) aminek a segítségével a projekt fejlesztésének későbbi szakaszaiiban is egyértelműsíthető bizonyos szövegrészletek tartalma.

2.2 Áttekintés

2.2.1 Általános áttekintés

Fontosabb alrendszerök:

- felhasználói grafikus felület
- játék menü
- pályageneráló
- kontroller
- pályafájlok

A felhasználó közvetlenül a grafikus felülettel van kapcsolatban, elsősorban ez szolgáltatja a játék megjelenítését a játékosok felé, másrészt ezen keresztül irányítható a kontroller.

A felhasználók a menüben állíthatják be a fejlesztés későbbi szakaszában specifikált opciókat.

A játék indulásakor a pályageneráló felépíti a pályát. A grafikus felületen való megjelenítéshez még szükség van a pályafájlokra, melyet a program a háttértárról tölt be. Itt tárolódnak például a karakterek képei.

A kontroller lesz a legfontosabb egység. Ez lesz gyakorlatilag a játékmotor. Feladata, hogy irányítsa, hogy éppen minnek kell megjelennie a képernyön és játszassa a játékosokat. Tulajdonképpen ez teremti meg a kapcsolatot, az interfést, a különböző komponensek, a játékosok és grafikus felület között.

Hálózatot a szoftver egyáltalán nem használ. Az előzőekben említetteken kívül nem igényel kitüntetett adatbázist.

2.2.2 Funkciók

A játéktér egy tengerrel körülvett jégmező, ami jégtáblából áll. minden jégtábla négyzet alakú és rendelkezik bizonyos tulajdonságokkal, amik véletlenszerűen kerülnek szétszórtára a játék kezdetén, ezek a következők:

- A jégtáblák között vannak stabilak, amikre akárhány játékos ráállhat, de vannak instabilak is, amik csak bizonyos számú karaktert bírnak el, amennyiben többen állnak rá, akkor a tábla átfordul és a rajta álló játékosok a vízbe esnek.
- Ezen felül előfordulhat, hogy a jégtáblák között vannak esetleg nem befagyott területek is, amire rálépve a játékos a vízbe esik.
- A táblákat és akár a nem befagyott területeket fedheti hó is, akár több réteg is, de legfeljebb négy.
- A táblákba lehetnek különböző tárgyak befagyva, de nem biztos, hogy mindegyikben van, sőt az is lehet hogy egy táblába több minden van befagyva különböző mélyen, ekkor többször is lehet kiásni belőle tárgyat.

Minden résztvevő egy körben pontosan négy egységnnyi munkát végezhet, amit tetszőlegesen használhat fel, de egy körben nem kötelező elhasználnia az összes munkáját. Munkának számít a szomszédos jégtáblára való lépés, egységnnyi mennyiségű hó eltakarítása, tárgy kiásása, felvétele vagy a karaktere egyedi képességeinek használata.

Egy adott jégtáblából csak akkor lehet kiásni a benne lévő tárgyat, ha a táblán nincsen hó. Ezeket a tárgyakat a játékosok akár át is adhatják egymásnak, de ez is egy egységnnyi munkába kerül és egy játékosnál csak maximum négy tárgy lehet. Az egyes tárgyak különböző plusz képességekkel ruházzák fel megtalálójukat, ilyen tárgyak lehetnek:

- Lapát: két egységnnyi hó takarítható el egy munkaráfordítással
- Kötél: vízbe esett játékos ki lehet menekíteni az adott körben, de csak a szomszédos jégtábláról, ekkor a kimentett személy a megmentője jégtáblájára kerül
- Búvárruha: tulajdonosa túléli a vízbeesést, ekkor visszakerül az előző táblára, ahol azelőtt állt, hogy átlépett volna arra a táblára, amiről a vízbe esett
- Élelem: Ezt felhasználva a játékos testhője eggel megnő (csak egyszer használható)
- Pálinka: csak az informatikus használhatja a saját képességéhez
- Egyéb tárgyak is megjelenhetnek, amik valójában nem használhatók

A játék kezdetén mindegyik felhasználó véletlenszerűen kap egy karaktert, ami lehet eszkimó, sarkkutató, fóka vagy informatikus. Mindegyik karakter rendelkezik bizonyos tulajdonságokkal, amikre csak ő képes, de minden ilyen képesség használata egy egységnnyi munkába kerül. Továbbá minden karakter különböző testhővel kezdi a játékot:

- Az eszkimó tud iglut építeni egy jégtáblára, ami megvédi az adott jégtáblán álló játékosokat egy hóvihar esetén, testhője 5.

- A sarkkutató meg tudja nézni lépés előtt, hogy a tábla amire lépne hány játékost bír el, testhője 4.
- A fóka amennyiben vízbe esik, túléli és visszakerül a vízbeesés előtti táblára, vagyis oda ahol azelőtt állt, hogy átlépett volna arra a táblára, amiről a vízbe esett, testhője 6.
- Az informatikus, ha megszerez egy üveg pálinkát, akkor a tárgy használatakor a vele egy táblán álló játékosok testhője visszaáll az eredeti értékre, testhője 3.

A játékot tovább nehezíti, hogy néha két kör között feltámad egy hóvihar egy véletlenszerű táblán, és a hóvihar epicentrumával szomszédos táblákat is befedi egy, de akár több újabb réteg hóval, attól függően, hogy mennyire intenzív a vihar, és ezen helyeken álló játékosok testhőjét eggyel csökkenti. A hóvihar akár egyszerre több helyen is lecsaphat egyszerre.

A játékot úgy lehet megnyerni, hogy a játékosok összegyűjtik egy jelzőrakéta alkatrészeit (pisztoly, jelzőfény, patron), amik szintén a jégbe vannak zárva véletlen helyeken. Ha ezeket a csapat összegyűjt, egy táblára állnak rá és az alkatrészeket egy játékosnak adják át, az egy munka felhasználásával összerakja és egy másik egységnyi munkával használja, akkor a játék véget ér. A csapat veszít viszont, ha valamelyik csapattag meghal játék közben, ami úgy történhet meg, hogy vagy nullára csökken a testhője és kihűl vagy vízbe esve megfullad az adott körben, ha senki sem menekíti ki.

2.2.3 Felhasználók

A felhasználók számára nincs szükség semmilyen különösebb előismeretre vagy tapasztalatra. Jelen dokumentum, illetve a későbbiekben elkészülő dokumentációk értelmezésével könnyen elsajátítható a program használata.

2.2.4 Korlátozások

A programnak egy játéktól elvárható mértékben stabilan és helyesen kell működnie: nem fagyhat le, nem léphet ki a felhasználó akarata nélkül és a követelményekben meghatározott módon kell játszania és a felhasználókat játszani engednie. Maximum négy felhasználót engedhet egyszerre játszani.

2.2.5 Feltételezések, kapcsolatok

Az előadások anyagait és a jegyzeteinket a projekt szakszerűbb átgondolásához használtuk.

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
1.01	Munkát végez a játékos	bemutatás	alapvető	megrendelő	Munkát végez	
1.02	Munkavégzés egy pont költségű	bemutatás	alapvető	megrendelő	Munkát végez	
1.03	Egy játékosnak	bemutatás	alapvető	megrendelő	Munkát végez	

	<i>nincs több pontja az ō köre véget ér</i>					
1.04	<i>Csak a tiszta jégtáblából lehet kiásni tárgyakat</i>	bemutatás	alapvető	megrendelő	Ásás	
1.05	<i>Az ásás munkának minősül</i>	bemutatás	alapvető	megrendelő	Ásás	
1.06	<i>Egy jégtáblában több tárgy van egyet lehet véletlenszerűen kiásni</i>	bemutatás	opcionális	csapat	Ásás	
1.07	<i>A játékos pusztá kézzel egy egységnyi hót képes letakarítani</i>	bemutatás	alapvető	megrendelő	Hó takarítása	
1.08	<i>Ha a játékos birtokában van egy ásonak akkor képes egy munkával 2 egységnyi havat eltakarítani</i>	bemutatás	alapvető	megrendelő	Hó takarítása	
1.09	<i>A kiásott tárgyakat a játékos fel tudja venni és elrakni azt ez is egy munka pontba kerül</i>	bemutatás	alapvető	megrendelő	Tárgy felvétele	
1.10	<i>Csak 4 tárgy lehet egy játékosnál</i>	kiértékelés	opcionális	csapat	Tárgy felvétele	
1.11	<i>A játékosok képesek az egyik tábláról egy másik táblára lépni 1 munka pontért cserébe</i>	bemutatás	alapvető	megrendelő	Lépés	

<i>1.12</i>	<i>Egyedi karakterek vannak a játékban különböző képességekkel</i>	<i>bemutatás</i>	<i>alapvető</i>	<i>megrendelő</i>		
<i>1.13</i>	<i>Alap karakterek: Eszkimók Sarkutatók</i>	<i>bemutatás</i>	<i>alapvető</i>	<i>megrendelő</i>		
<i>1.14</i>	<i>Extra karakterek: Fóka Informatikus</i>	<i>bemutatás</i>	<i>opcionális</i>	<i>csapat</i>		
<i>1.15</i>	<i>Eszkimók képessége, hogy iglut tud építeni</i>	<i>bemutatás</i>	<i>alapvető</i>	<i>megrendelő</i>	Iglut építése	
<i>1.16</i>	<i>Sarkutatók képessége, hogy megtudja vizsgálni hány játékost bír el egy adott jégtábla</i>	<i>bemutatás</i>	<i>alapvető</i>	<i>megrendelő</i>	Jégtábla vizsgála-ta	
<i>1.17</i>	<i>Fóka képessége, hogy a vízbe esést túl élje és visszakerüljön az utolsó jégtáblára amin volt</i>	<i>kiértékelés</i>	<i>opcionális</i>	<i>csapat</i>	Kimá-szik a vízből	
<i>1.18</i>	<i>Informatikus képessége, hogy a pálinka tárgy használatával képes minden vele egy csapatban és egy táblán álló játékos testhőjét visszaállítani az eredeti testhőjére</i>	<i>kiértékelés</i>	<i>opcionális</i>	<i>csapat</i>	Pálinkák-zunk	

1.19	Tárgyakat képesek egyik játékos egy másik játékosnak átadni	bemutatás	opcionális	csapat	Tárgy átadása	
------	---	-----------	------------	--------	---------------	--

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
2.01	A program Java nyelven készül.	bemutatás	alapvető	megrendelő	
2.02	A termékhez ajánlott a Windows 10 operációs rendszer.	bemutatás	opcionális	csapat	Az egyszerűség kedvéért csak a windows operációs rendszerre biztosítjuk a helyes működést attól függetlenül, hogy a programozási nyelv cross-platform.
2.03	A programnak szüksége lehet fájlkezelési jogokra.	bemutatás	fontos	csapat	Abban az esetben fontos, ha a játékban a mentés/betöltés funkció implementálásra kerül.

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
3.11	A program futtatásához JRE v8 környezet szükséges.	bemutatás	alapvető	megrendelő	
3.12	A futtatáshoz a Windows 10 operációs rendszer ajánlott.	bemutatás	alapvető	csapat	Csak ebben az OS-ben tudjuk garantálni a hibátlan futást.
3.13	Minden egyéb követelmény teljesülése.	bemutatás	alapvető	csapat	Átadásra a programterv minden alpontról teljesülnie kell.

2.3.4 Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
4.11	<i>A programot egy "jar" kiterjesztésű fájl segítségével lesz hordozható, illetve könnyen indítható.</i>	bemutatás	opcionális	csapat	
4.12	<i>A program megbízhatóságát, tesztelhetőségét saját tesztfüggvények biztosítják.</i>	bemutatás	alapvető	csapat	
4.13	<i>A programhoz felhasználói útmutatót biztosítunk a felhasználó számára.</i>	bemutatás	alapvető	csapat	

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	Munkát végez
Rövid leírás	Ez az a tevékenység, amely az adott körben a játékosnak meglévő 4 egységnyi munkájából egyet elvesz.
Aktorok	Játékosok
Forgatókönyv	1. Ha az egyik játékos munkát végez akkor a meglévő 4 pont munkából egyet elveszít.
Alternatív forgatókönyv	1.A Ha a játékosnak elfogyott az összes munka pontja, akkor a kör véget ért számára.

Use-case neve	Ásás
Rövid leírás	Egy munka amely a letakarított jégtáblából lehet egyes tárgyakat kiásni.
Aktorok	Játékosok
Forgatókönyv	1. Ha a jégtábla tiszta, akkor a játékos képes kiásni a tárgyat a jégtáblából.

Alternatív forgatókönyv	1.A Ha a jégtábla nem tartalmaz semmilyen tárgyat, akkor nem lehet ásni ezen a mezőn.
Alternatív forgatókönyv	1.B Ha egy jégtáblában több tárgy van akkor véletlenszerűen ás ki egy tárgyat.

Use-case neve	Hó takarítása
Rövid leírás	A játékos azon a jégtáblán, amin áll eltakarítja a havat.
Aktorok	Játékosok
Forgatókönyv	1. Az adott jégtábláról eltakarít 1 egységnyi havat.
Alternatív forgatókönyv	1.A Ha a játékosnál ásó van, akkor 2 egységnyi havat képes eltávolítani.

Use-case neve	Tárgy felvétele
Rövid leírás	A már kiásott, tárgyakat a játékos fel tudja venni és magánál tudja tartani.
Aktorok	Játékosok
Forgatókönyv	1. A játékos felveszi a tárgyat és elrakja azt.
Alternatív forgatókönyv	1.A Nem fér el több tárgy a játékosnál. (Max 4 tárgy lehet egy játékosnál)

Use-case neve	Tárgy átadása
Rövid leírás	Az egyik játékos képes a nála lévő (választható) tárgyak egyéket átadni a vele egy táblán álló játékosnak.
Aktorok	Játékosok
Forgatókönyv	1. A játékos átadja a tárgyat amit választott a másik játékosnak és az elrakja azt a tárgyat
Alternatív forgatókönyv	1.A A játékos át akar adni egy tárgyat a másik játékosnak de annak nincs több helye az táskjában a tárgy eldobja és a jégtáblára kerül és bárki felveheti azt

Use-case neve	Lépés
Rövid leírás	A játékos egyik jégtábláról a másikra mozog.
Aktorok	Játékosok
Forgatókönyv	1. A játékos egyik mezőről a másikra lép és a jégtábla teherbírása csökken eggyel.
Alternatív forgatókönyv	1.A Ha a jégtábla nem stabil és több játékos áll rajta, mint amennyit a tábla elbír, akkor a jégtábla széttörök.
Alternatív forgatókönyv	1.A.1 A széttört jégtáblán állók a vízbe esnek.

Alternatív forgatókönyv	1.B Egy olyan mezőre lépnek ahol nincs jégtábla, de hó fedи, azt akkor a játékos a vízbe esik.
--------------------------------	---

Use-case neve	Iglu építése
Rövid leírás	A jégtáblán az eszkimó egy iglut épít a hóvihar ellen.
Aktorok	<i>Eszkimók</i>
Forgatókönyv	1. Ha egy hóvihar van azért, hogy az eszkimó ne veszítsen testhőt egy iglut építhet és megbújhat benne, míg a vihar tart.

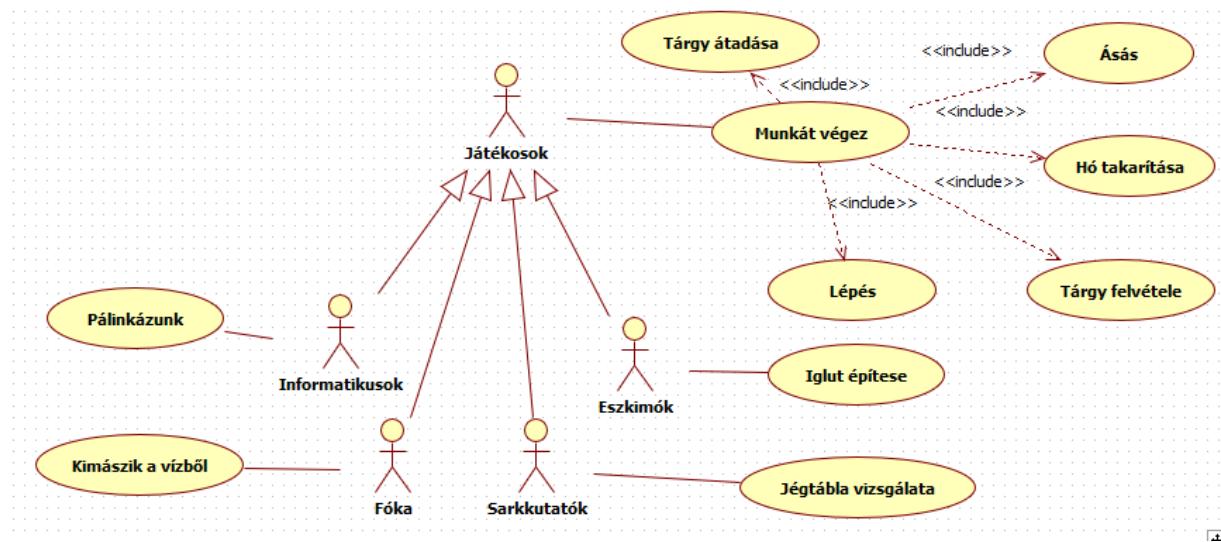
Use-case neve	Jégtábla vizsgálata
Rövid leírás	A sarkutatók egy jégtábla teherbírását képesek megvizsgálni.
Aktorok	<i>Sarkutatók</i>
Forgatókönyv	1. A sarkutató a maga alatti és körülötte lévő jégtáblákat képes vizsgálni

2.4.2

Use-case neve	Pálinkázunk
Rövid leírás	Ha egy pálinka van az informatikus játékosnál használhatja azt hogy visszaállítsa a egy táblán lévő csapattagjainak a testhőjét az eredeti értékre.
Aktorok	<i>Informatikus</i>
Forgatókönyv	1. Az informatikus kibontja a pálinkát és megosztja azt csapattársaival, hogy jól felmelegítség magukat

Use-case neve	Kimászik a vízből
Rövid leírás	Ha a fóka beleesik a vízbe akkor képes onnan kimászni
Aktorok	<i>Fóka</i>
Forgatókönyv	1. Beleesett a vízbe és kimászik arra a jégtáblára amin utoljára még állt

2.4.3 Use-case diagram



2.5 Szótár

alkatrész	Egy tárgy. Három különböző alkatrész szükséges a jelzőrakéta összeszereléséhez.
búvárruha	Egy tárgy. A birtoklója tud vízbe esést túlélni.
eszkimó	Egy kaszt.
élelem	Egy tárgy. Birtoklója elhasználhatja a testhője egy egységgel növeléséhez.
felszerelés	A szereplő tárgykészlete. Maximális mérete: 4.
fóka	Egy kaszt.
hó eltakarítása	Egy hóval fedett jégtábláról egy réteg hó eltávolítása.
hóréteg	A jégtáblákon fordulhat elő. Alatta nem látszanak a jégtáblán lévő tárgyak.
hóval fedett jégtábla	A jégtáblát 1-4 réteg hó borítja.
hóvihar	Egy véletlenszerű esemény, ami körök között fordul elő. Egy véletlenszerű jégtáblára és annak szomszédjaira egy réteg hó kerül. Az ezeken a jégtáblákon álló szereplők testhője 1-el csökken, ha a jégtáblán amin állnak nincs iglu.
iglu	Egy jégtáblán helyezkedhet el. Iglu-t tartalmazó jégtáblán álló szereplők a hóvihar alatt nem veszítenek testhőt. Egy jégtáblán csak egy iglu állhat.
iglu építése (eszkimó)	Az eszkimó kaszt képessége. Arra a jégtáblára, amin áll, egy iglu-t helyez.
informatikus	Egy kaszt.
instabil jégtábla	Olyan jégtábla, amin csak egy szereplő állhat egyszerre.
instabil jégtábla átfordul	Ha egynél több szereplő áll egy instabil jégtáblára, a szereplők vízbe esnek.

ivás	Egy tevékenység. Egy informatikus megiszik egy pálinkát és a vele egy jégmezőn állók testhője újratöltődik.
jégmező	A játék területe, jégtáblából áll.
jégtábla	Egységes méretű terület (négyzet alakú). Tartalmazhat tárgyat, havat és szereplő(ke)t.
jégtábla vizsgálata (sarkkutató)	A sarkkutató kaszt képessége. Megnézi, hány szereplő bír el egy szomszédos jégtábla.
jelzőfény	A jelzőrakéta egyik alkatrésze.
jelzőrakéta	A játék célja ennek összeszerelése és elsütése.
jelzőrakéta elsütése	Egy tevékenység. A játék célja. Ha megtörténik, vége a játéknak. Egy munkába kerül.
jelzőrakéta összeszerelése	Egy tevékenység. Ha egy szereplő birtokolja a jelzőrakéta három különböző alkatrészét, akkor egy munkáért cserébe megkaphatja a jelzőrakétát (ekkor eltűnnék tőle az alkatrészek).
kaszt	A szereplő változata egyedi képességgel és adott maximális munkával.
képesség	Egy kaszthoz tartozó tevékenység (csak a hozzá tartozó kaszt képes elvégezni).
kihűlés	Ha egy szereplő testhője 0-ra csökken, meghal.
kimentés	Egy tevékenység. Egy szomszédos jégmezőn vízbe esett szereplőt a saját jégmezőjére tud juttatni az a szereplő, akinek kötele van.
kör	A játék egy időegysége.
kötél	Egy tárgy. A birtoklója tud kimenteni.
lapát	Egy tárgy. A birtoklója tud lapátolni.
lapátolás	Egy tevékenység. Egy hóval fedett jégtábláról két réteg hó eltávolítása.
lépés	Egy tevékenység. Egy szereplő mozgatása egy szomszédos jégtáblára.
lyuk	Olyan jégtábla, amin nem állhatnak szereplők. A rálépő szereplő vízbe esik.
munka	Egy tevékenység elvégzése egy egységgel csökkenti az adott körben elvégezhető tevékenységek számát a tevékenységet végző szereplónél. Körönként visszaáll 4-re.
pálinka	Egy tárgy. Egy informatikus megihatja
patron	A jelzőrakéta egyik alkatrésze.
pisztoly	A jelzőrakéta egyik alkatrésze.
sarkkutató	Egy kaszt.
stabil jégtábla	Olyan jégtábla, amin akárhány szereplő állhat egyszerre.
szereplő	Egy játékos által irányított karakter.
szereplő meghal	Testhője 0-ra csökken vagy vízbe fullad. Ekkor vége a játéknak.

szomszédos jégtábla	Egy jégtábla szomszédos egy másik jégtáblával, ha van közös élük.
tárgy	Egyes jégtáblákon megtalálható eszköz. Fedheti hó.
tárgy átadása	Egy tevékenység. Egy tárgyat egy szereplő átadhat egy másik szereplőnek, ha ugyan azon a jégtáblán állnak.
tárgy felvétele	Egy tevékenység. Egy tárgy elvételre a jégtábláról és a szereplő felszereléséhez adása.
tárgy kiásása	Egy tevékenység. Egy szereplő felvesz egy tárgyat, ami a vele megegyező jégtáblán található és nem fedi hó. Csak tiszta jégtáblán lehet elvégezni.
táska	Egy olyan eszköz amivel minden játékos rendelkezik a játék eleje óta és ebben tárolhatják a azokat a tárgyakat amiket a játék során felvesznek (4 db tárgy lehet maximum benne).
testhő	Egy szereplőkhöz kötött szám, ami ha nullára csökken, az adott szereplő meghal.
testhő újratöltés	A testhő visszaállítása maximumra (ez függ a szereplő kasztjától).
tevékenység	Bármilyen mozgás, képesség- vagy tárgyhasználat. Elvégzése egy egység munkába kerül.
tiszta jégtábla	A jégtáblát nem borítja hó.
vízbe esés	Amennyiben a vízbe eső szereplőnél nincs búvárruha és nem fóka, valamint nem tudják kimenteni, meghal.
vízbe esés túlélése	Egy búvárruhával rendelkező szereplő vízbe esve nem hal meg.

2.6 Projekt terv

2.6.1 Határidők

Határidő	Feladat	Felelős
febr. 24.	Követelmény, funkcionálitás, projekt - beadás	Bodó
márc. 2.	Analízis modell kidolgozása 1. - beadás	Gaál
márc. 9.	Analízis modell kidolgozása 2. - beadás	Nagy
márc. 16.	Szkeleton tervezése - beadás	Rédai
márc. 23.	Szkeleton - beadás és a forráskód herculesre való feltöltése	Varga
márc. 30.	Prototípus koncepciója - beadás	Bodó

ápr. 6.	Részletes tervezet - beadás	Gaál
ápr. 20.	Prototípus készítése, tesztelése	Nagy
árp. 27.	Prototípus - beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése	Rédai
máj. 4.	Grafikus felület specifikációja - beadás	Varga
máj. 18	Grafikus változat és Összefoglalás - beadás és a forráskód herculesre való feltöltése	Bodó

2.6.2 Erőforrások

Dokumentálásra használt eszközök: Microsoft Word, Google Docs

Kommunikáció: Személyesen, Messenger

Modellező eszköz: WhiteStarUML, StarUML

Fejlesztőeszköz: Eclipse/IntelliJ

Dokumentumok megosztása: Google Drive

Forráskód megosztása, verziókezelése: GitHub

2.7 Napló

Kezdet	Időtartam	Résznevők	Leírás
2020.02.20 20:00	2 óra	Gaál Bodó Rédai Nagy Varga	Értekezlet. Döntés: Rédai elkészíti az use-case leírásokat. Gaál elkészíti a Bevezetést és a Funkciók leírástát. Varga a Szótár elkészítése
2020.02.20 18:00	3	Gaál	Bevezető rész megírása és a Funkciók pontosítása
2020.02.21 8:00	4 óra	Rédai	Tevékenység: Rédai elkészíti a use-case-t és leírását
2020.02.22 10:00	3 óra	Rédai	Tevékenység: Rédai javította a use-case leírásokat és a diagramot. Elkészítette a funkcionális követelményeket
2020.02.21 18:00	3 óra	Varga	Funkciók leírása, szótár elkészítése. Bevezetés javítása.
2020.02.23 12:00	3 óra	Bodó Nagy	erőforrások, azok követelményei, átadásokkal-, egyéb követelmények, korlátozások, feltételezések, kapcsolatok, a dokumentum javítása, átnézése
2020.02.23 15:00	0.5 óra	Gaál Bodó Rédai Nagy Varga	Napló megírása

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Alkatrész

Három ilyen alkatrész szükséges a jelzőpisztoly összeszereléséhez.

3.1.2 Búvárruha

Viselésével a szereplő életben marad ha vízbe esett, akár két kör között is, de ebben az esetben csak menni tud, más munkát végezni nem.

3.1.3 Eszkimó

Iglut tud építeni egy adott jégtáblára, de az épített igluk a következő körre eltűnnek.

3.1.4 Élelem

Az élelem elfogyasztásával nő a szereplő testhője eggel.

3.1.5 Fóka

Vízbe esve életben marad két kör között is, de ilyenkor csak mozogni tud, más munkát végezni nem.

3.1.6 Informatikus

Csak ő tudja használni a pálinka tárgyat.

3.1.7 Jelzőpisztoly

Elsütésével vége a játéknak.

3.1.8 Jégtábla

Ez egy olyan objektum, melyen állhat szereplő és lehet rajta, illetve belefagyva is tárgy. A jégtáblát boríthatja több réteg hó is egyszerre. Az eszkimó építhet rajta iglut. Feladata, hogy ismerje a szomszédos jégmezőket, rajta lévő játékosokat és tárgyakat, tudjon játékosokat átadni és fogadni más jégmezőktől, valamint lehet belőle tárgyakat kiásni és tárgyakat elhelyezni rajta.

3.1.9 Kötél

Segítségével ki lehet menteni egy vízbe esett szereplőt, aki a mentés után a kimentője jégtáblájára kerül.

3.1.10 Lapát

Kétszer annyi havat lehet vele eltakarítani, mint kézzel.

3.1.11 Manager

A játék menedzser dolga, hogy a játék elején elkészítse a játékteret és kisorsolja a tárgyak kezdeti helyét. Játék közben körönként bizonyos lehetőséggel generál hóvihart és egy kör ideje alatt egymás után ad lehetőséget a szereplöknek munkára.

3.1.12 Pálinka

Csak az Informatikus tudja meginni. Használatkor az informatikus jégtábláján minden szereplőnek a testhője beáll a lehető legmagasabb szintre.

3.1.13 Sarkkutató

Meg tudja vizsgálni egy szomszédos jégtábla teherbírását.

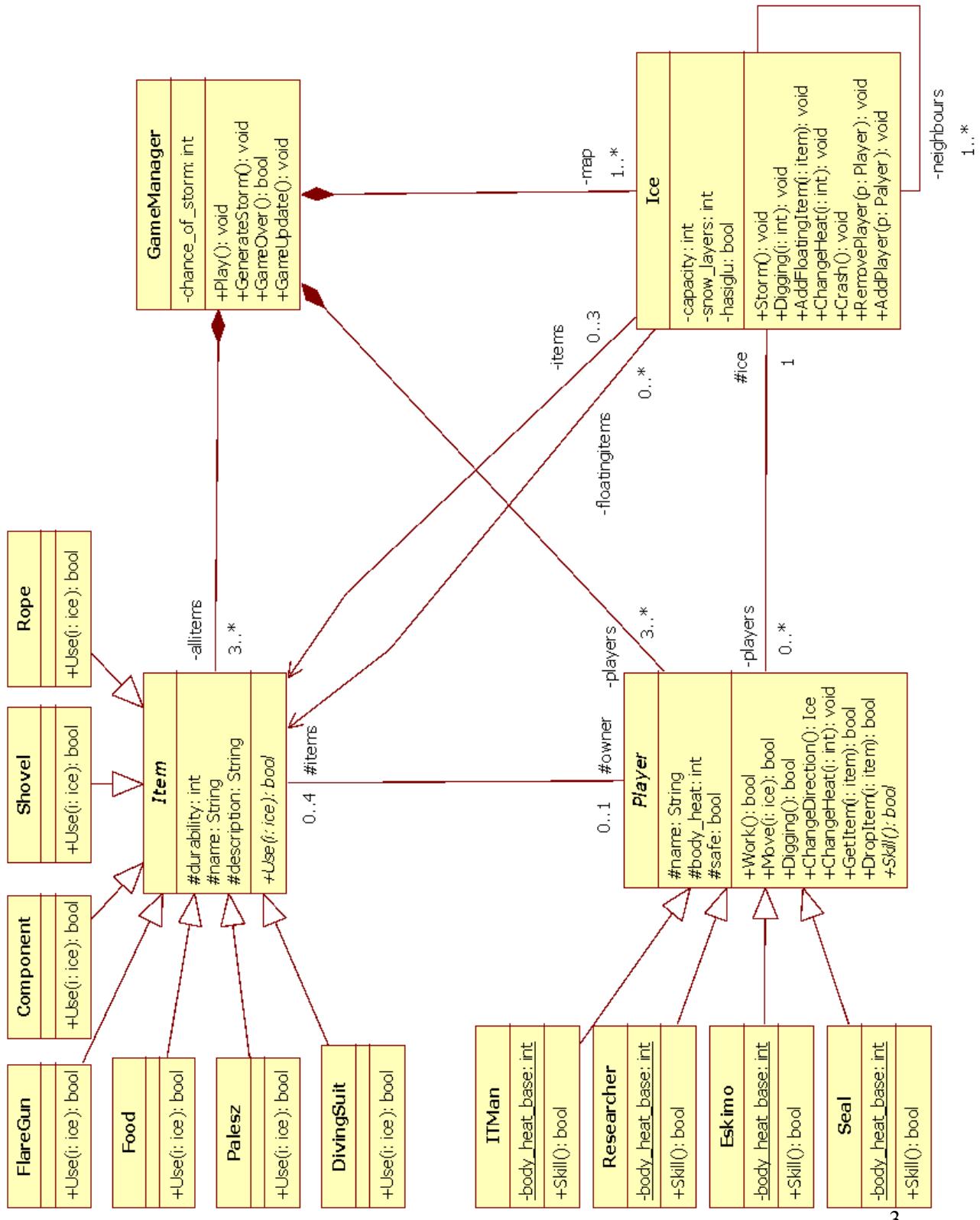
3.1.14 Szereplő

Egy karakter, amit a felhasználó irányít. Tulajdonságai, hogy legfeljebb 4 tárgyat tud tárolni és a játék kezdetén van bizonyos mennyiségű testhője. Továbbá képes jégtáblákon mozogni, havat ásni, nála lévő tárgyakat használni és vízbe tud esni. Meg tud halni, ha elfogy a testhője, vagy bűvárruha nélkül vízbe esik és nem mentik ki. Ez a játék végét is jelenti. A szereplőnek típusai vannak, és azoknak egyéni képessége.

3.1.15 Tárgy

Egy olyan eszköz, amit használva különböző képességekkel ruházza fel a tulajdonosát.

3.2 Statikus struktúra diagramok



3.3 Osztályok leírása

3.3.1 Component

- **Felelősség**

Három komponensből állítható elő a FlareGun.

- **Ősosztályok**

- Item

- **Metódusok**

- **bool Use(Ice i):** Ha a használó játékosnál van három komponens, akkor azokat kitörli a játékból, és a játékos felszereléséhez adja a FlareGun-t. Ez nem kerül munkába.

3.3.2 DivingSuit

- **Felelősség**

Ha egy játékos viseli és vízbe esik, túléli. Ha csak a felszerelésében van, de még nem használta akkor nem.

- **Ősosztályok**

- Item

- **Metódusok**

- **bool Use(Ice i):** A használó játékos felveszi a búvárruhát, ezzel eltűnik a tárgy a felszerelésből és a safe attribútuma true-ra változik. Használatakor eltűnik. (durability értéke 1)

3.3.3 Eskimo

- **Felelősség**

Iglut tud építeni arra a jégtáblára, amin áll. Az iglu csak az adott körben érvényes, a következő kör elején eltűnik.

- **Ősosztályok**

- Player

- **Attribútumok**

- **int body_heat_base:** Maximális testhőmérséklet (ennyiről indul).

- **Metódusok**

- **bool Skill():** Épít egy Igut a saját jégtáblájára. Az Igut egy körön át marad ott.

3.3.4 FlareGun

- **Felelősség**

El lehet sütni. Elsütésével a játékosok megnyerik a játékot.

- **Ősosztályok**

- Item

- **Metódusok**

- **bool Use(Ice i):** A birtokló játékos elsüti és megnyeri a játékot.

3.3.5 Food

- **Felelősség**

Használatával a használó testhője megnő.

- **Ősosztályok**

- Item

- **Metódusok**

- **bool Use(Ice i):** A játékos megeszi és megnöveli a testhőjét eggel.

3.3.6 GameManager

- **Felelősség**

A játékot vezérli. Generálja a játékot, körönként meghívja a játékosokat, hogy mozogjanak, végezzenek munkát. A körök végén véletlenszerűen hóvihart hoz létre, megnézi, hogy meghalt-e valaki, valamint ellenőrzi, hogy elszüttötték-e a jelzőpisztolyt.

- **Attribútumok**

- **int chance_of_storm:** A hóvihar keletkezésének esélye. Körönként nő.
- **Player players[]:** A játékosok listája.
- **Ice map[]:** A játékban szereplő összes jégmező.
- **allitems[]:** A játékban szereplő összes tárgy listája.

- **Metódusok**

- **void Play():** A játékosok egymás utáni játszatásáért felelős.
- **void GenerateStorm():** Kiválasztva egy (nagyobb pálya esetén több) jeigtáblát, arra és annak szomszédaira egy hóréteget tesz, továbbá az azokon a táblákon tartózkodó játékosok testhőjét eggel csökkenti (ha nincs iglu). A hóvihar esélye körönként nő.
- **void GameUpdate():** A játékosok sorrendje minden kör után változik. (A kör végéig még egy éppen vízben lévő játékost még egy soron lévő játékos ki tud menteni. Ha éppen a kör végén esik valaki a vízbe akkor őt már nem lehet

- menteni. A cserére azért van szükség hogy ne minden ugyanaz a játékos kerüljön ilyen helyzetbe, hogy már a kör végén nem lehet megmenetlen.)
- **bool GameOver()**: minden kör végén kiértékeli hogy vége van-e a játéknak. Egy játék véget ér, ha van olyan játékos akinek a testhője nem több 0-nál vagy egy játékos úgy maradt vízben hogy nincs búvárruhája és nem fóka.

3.3.7 Ice

- **Felelősség**

Egy területegységet reprezentál. Tárgyakat és játékosokat tárol. Valamennyi réteg hó fedheti. Ismeri a szomszédait. Ha egy adott számnál több ember áll rajta, a tábla beszakad. Léteznek olyan táblák is, melyek nulla embert bírnak el (víz), de ezeken is lehet hóréteg.

- **Attribútumok**

- **int capacity**: A jégtábla ennyi játékosot bír el.
- **int snow_layers**: A jégtáblán ennyi réteg hó van.
- **bool hasiglu**: Van-e a jégtáblán iglu.
- **Player players[]**: A jégtáblán tartózkodó játékosok száma.
- **Ice neighbours[]**: A szomszédos jégtáblák.
- **Item items[]**: A jégbefagyott tárgyakra referenciák.
- **Item floatingitems[8]**: A jégtáblán lebegő tárgyakra referenciák.

- **Metódusok**

- **void Storm()**: Hóvihar tombol a jégmezőn. A hóréteg megvastagodik, illetve a mezőn álló játékosok testhője csökken.
- **void Digging(int i)**: Eltávolít i darab hóréteget. Ha a hórétegek száma 0-ra csökken, tárggyal tér vissza.
- **void AddFloatingItem(Item i)**: A lebegő tárgyakhoz hozzáad egy új tárgyat.
- **void ChangeHeat(int i)**: A jégtáblán álló összes játékos testhőjét változtatja.
- **bool Crash()**: Vizsgálja hogy a jégtábla elbírja e a játékosokat. Ha a capacity adattag kevesebb mint a jégtáblán álló játékosok száma beszakad.
- **void RemovePlayer(Player p)**: Eltávolít egy játékosat a jégtábláról.
- **void AddPlayer(Player p)**: Hozzáad egy játékosat a jégtáblához.

3.3.8 Item

- **Felelősség**

Absztrakt osztály. Egy eszköz, ami a birtoklójának különböző tevékenységek elvégzésére adnak lehetőséget. El lehet őket dobni egy jégtáblára, fel lehet őket venni jégtábláról, ki lehet őket ásni jégtáblákból.

- **Attribútumok**

- **int durability**: A tárgy használtsága. 0-ra csökkenéskor a tárgy elhasználódik és eltűnik.
- **String description**: a tárgy tulajdonságainak ismertetése a felhasználó számára.
- **Player owner**: A játékos, akinél a tárgy van.

- **String name:** A tárgyak neve.
- **Metódusok**
 - **bool Use(Ice i):** Absztrakt metódus. minden tárgynak egyedi képessége van.

3.3.9 ITMan

- **Felelősség**

Meg tudja inni a Pálinkát.

- **Ősosztályok**
 - Player
- **Attribútumok**
 - **int body_heat_base:** Maximális testhőmérséklet (ennyiről indul).
- **Metódusok**
 - **bool Skill():** Megissza a nála lévő pálinkát. Azon a jégtáblán, amin áll, minden játékos testhője visszaáll a maximumra.

3.3.10 Palesz

- **Felelősség**

Csak az ITMan tudja meginni. Használatkor a használó jégtábláján álló játékosok testhője maximumra nő.

- **Ősosztályok**
 - Item
- **Metódusok**
 - **bool Use(Ice i):** A használó jégtábláján álló játékosok testhője maximumra nő. Használat után eltűnik.

3.3.11 Player

- **Felelősség**

Absztrakt osztály. Egy ember által irányított játékosot reprezentál. Tud lépni, munkát végezni és meg tud halni. Egy munkának számít egy tárgy használata és a karaktertípus saját képességének használata.

- **Attribútumok**
 - **String name:** A játékos neve.
 - **int body_heat:** A játékos testhője. Ha nullára csökken, meghal.
 - **Ice ice:** Az a jégtábla, amin a játékos éppen áll.
 - **Item items[4]:** A játékosnál lévő tárgyak.

- **Metódusok**

- **bool Work()**: A játékos dolgoztató függvény. Ha a játékos vízben van, akkor búvárruha nélkül nem dolgozhat. Ha van búvárruhája akkor a vízben csak mozoghat. A jégtáblán bármilyen munkát végezhet. Ha igazzal tér vissza a pisztoly el lett sütve és vége a játéknak. Ha hamis a játékosok befejezte a munkát és a következő játékos jön.
- **bool Move(Ice i)**: A játékos átkerül egy másik jégtáblára.
- **bool Digging()**: A játékos vagy havat takarít a jégtábláról amin áll, vagy egy tárgyat ás ki a jégtáblából amin áll.
- **Ice ChangeDirection()**: A játékos irányváltására szolgál. Meghívva a visszaadja a játékos jégtáblájával szomszédos jégtáblák közül a következőt egy bizonyos sorrend szerint.
- **void ChangeHeat(int i)**: A játékos testhőjét változtatja.
- **bool GetItem(Item i)**: Egy lebegő i itemet felvesz a játékos felszerelése közé.
- **bool DropItem(Item i)**: A játékos lerak egy tárgyat a felszereléséből arra a jégtáblára amin áll. Onnantól kezdve az a tárgy "lebeg" a jégtáblán.
- **bool Skill()**: Absztrakt metódus, a leszármazottak egyedi képességének használatához.

3.3.12 Researcher

- **Felelősség**

Meg tudja vizsgálni egy szomszédos jégtábla teherbírását.

- **Ősosztályok**

- Player

- **Attribútumok**

- **int body_heat_base**: Maximális testhőmérséklet (ennyiről indul).

- **Metódusok**

- **bool Skill()**: Megvizsgál egy szomszédos jégtáblát és visszakapja annak a teherbírását (hány embert bír el).

3.3.13 Rope

- **Felelősség**

Ki lehet vele menteni egy szomszédos jégtáblán vízbeesett játékest.

- **Ősosztályok**

- Item

- **Metódusok**

- **bool Use(Ice i)**: A használóval szomszédos jégtáblán vízbeesett játékest kimenekíti maga mellé

3.3.14 Seal

- **Felelősség**

A kör végén búvárruha nélkül is túléli a vízbeesést.

- **Ősosztályok**

- Player

- **Attribútumok**

- **int body_heat_base:** Maximális testhőmérséklet (ennyiről indul).

- **Metódusok**

- **bool Skill():** Üres metódus. A különleges képessége karakterből adódóan automatikusan működik.

3.3.15 Shovel

- **Felelősség**

Használatával egyszerre két réteg havat lehet ellapárolni arról a jégtábláról, amin a játékos áll. Használata csökkenti a Durability-t.

- **Ősosztályok**

- Item

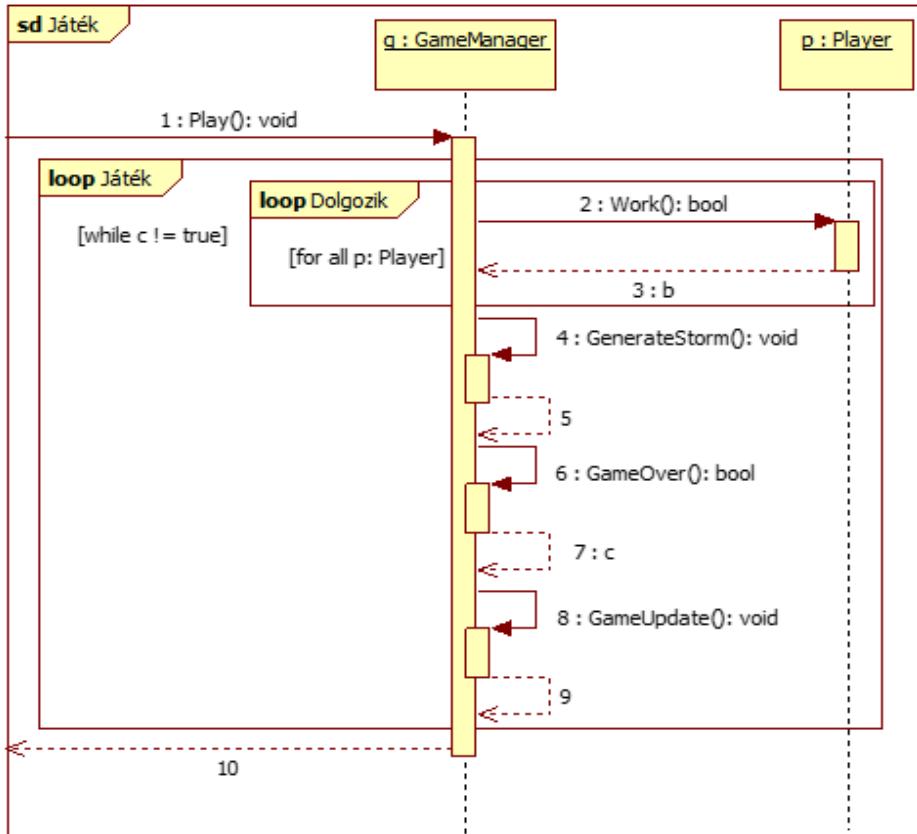
- **Metódusok**

- **bool Use(Ice i):** A használó jégtáblájáról eltávolít két réteg havat. Ezzel csökken a durability. Ha a durability 0 vagy kevesebb, a Shovel eltűnik.

3.4 Szekvencia diagramok

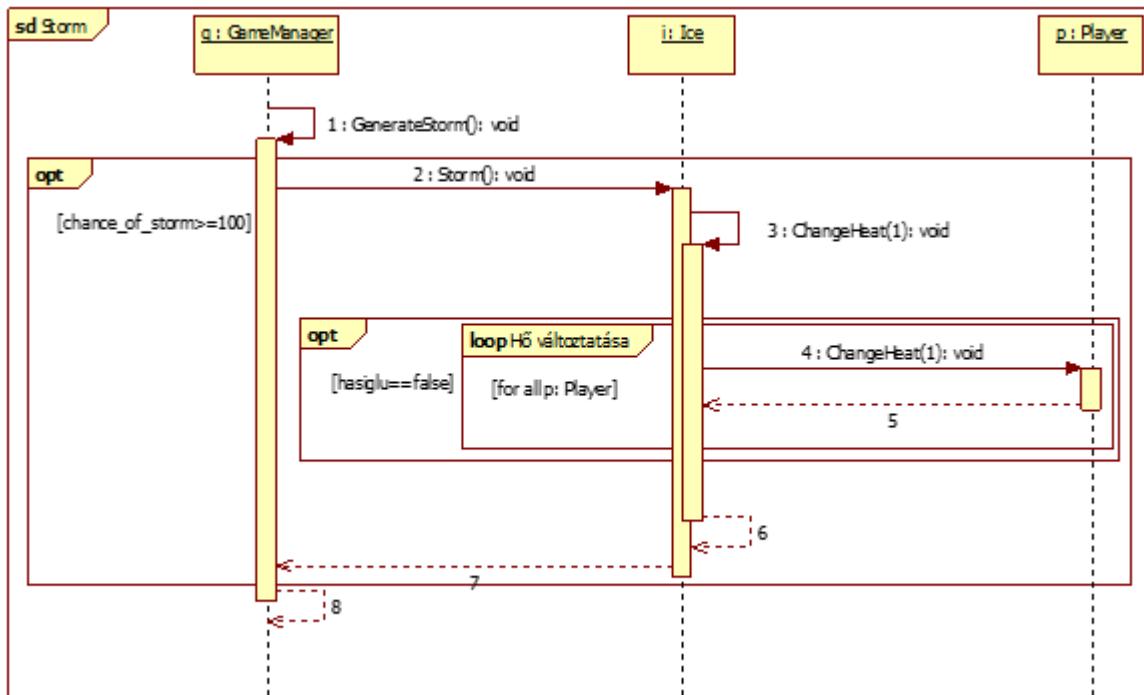
3.4.1.

Egy kör lefolyása a játékban. A GameManager Play függvénye játszatja az összes játékosot addig, amíg el nem fogy az egyes játékosok munkái. Majd a hóvihar esélye változik. (Az is lehet, hogy hóvihar keletkezik). Majd a GameOver fv. vizsgálja, hogy vége van-e a játéknak. Majd a játékosok sorrendjét cseréli a GameUpdate fv.



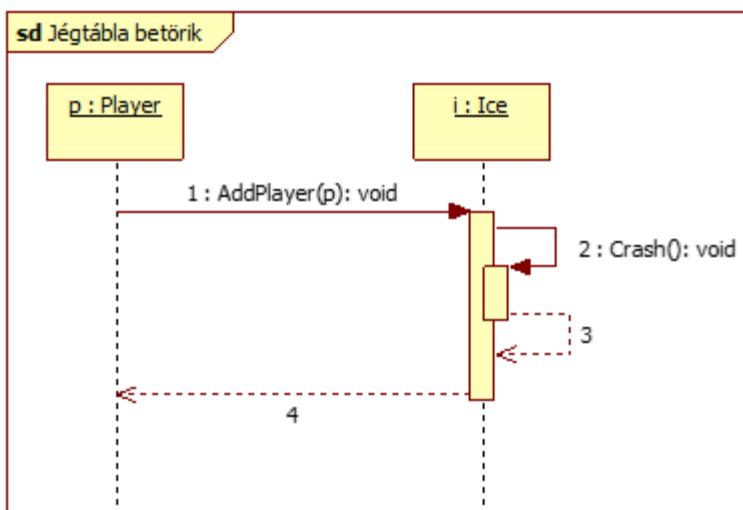
3.4.2

Hóvíhar generálódik, ha a GameManager chance_of_storm adattjának értékéhez legalább 100. (Ha nem lenne akkor egy random számmal növeljük az értékét és vége a hívásnak.) Hóvíharban azon játékosok, akiknek jégtábláján nincs iglu, tehát nincs védeeltségük, 1 testhő csökkenést szenvednek el.



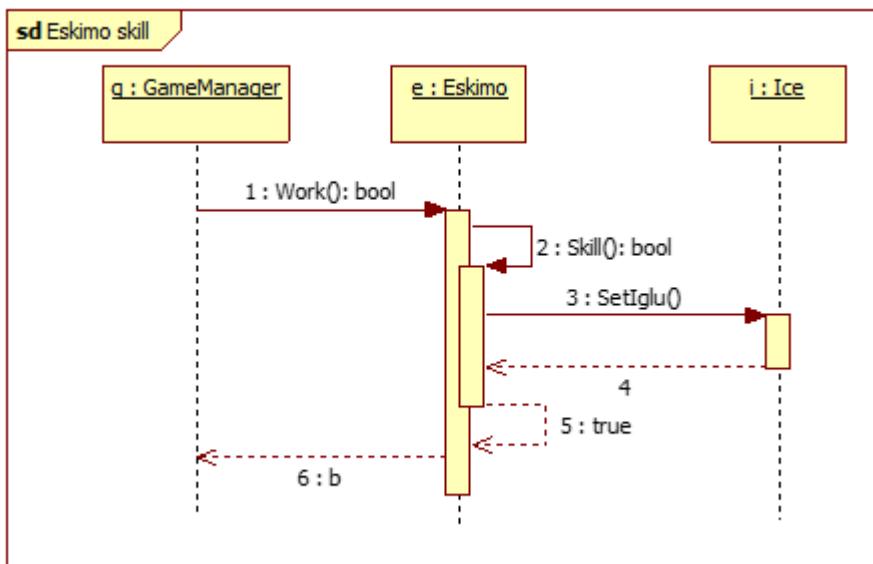
3.4.3

Az alábbi szekvencia diagramon látható egy jégtábla betörése. A játékos hozzáadása után túllépjük a jégtábla kapacitását, majd az betörök.

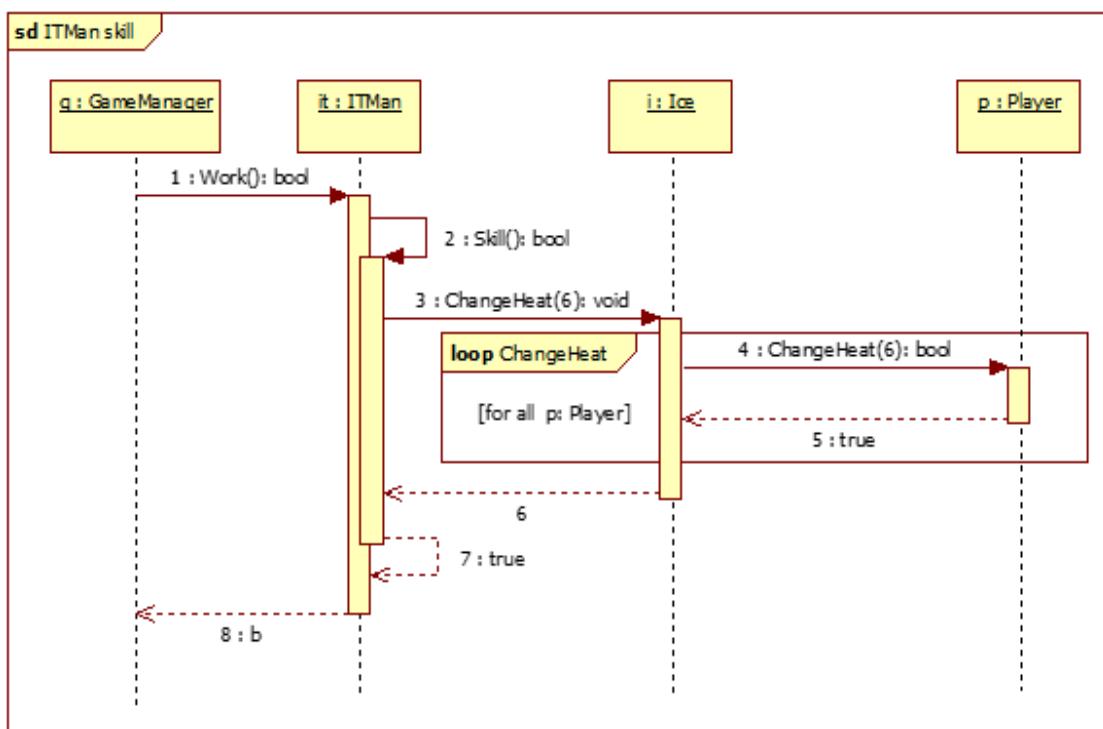


3.4.4

Az alábbi szekvencia diagramon látható az eszkimó képességének használata.

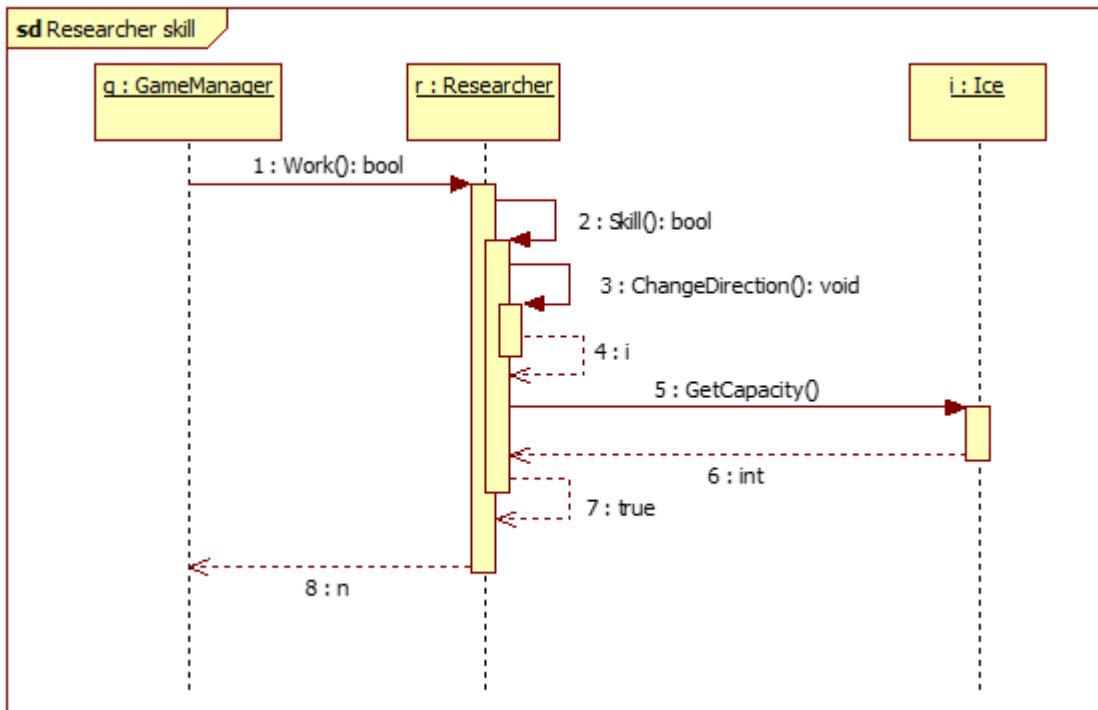
**3.4.5**

Az alábbi szekvencia diagramon látható az informatikus képességének a használata.



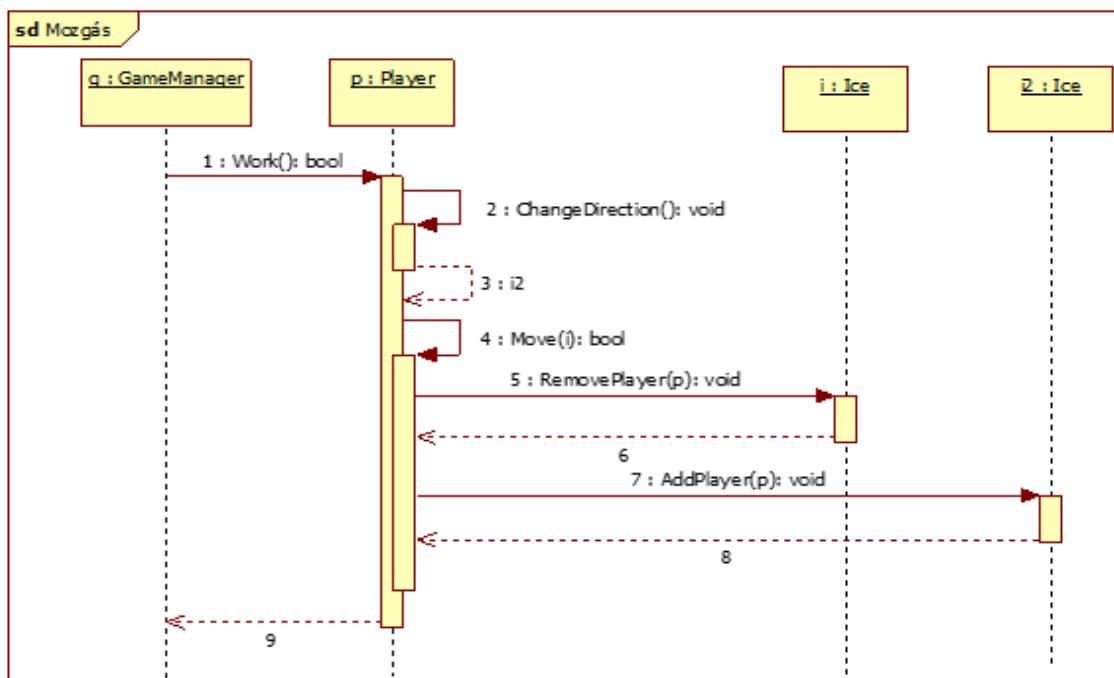
3.4.6

Az alábbi szekvencia diagramon látható a kutató képességének a használata.



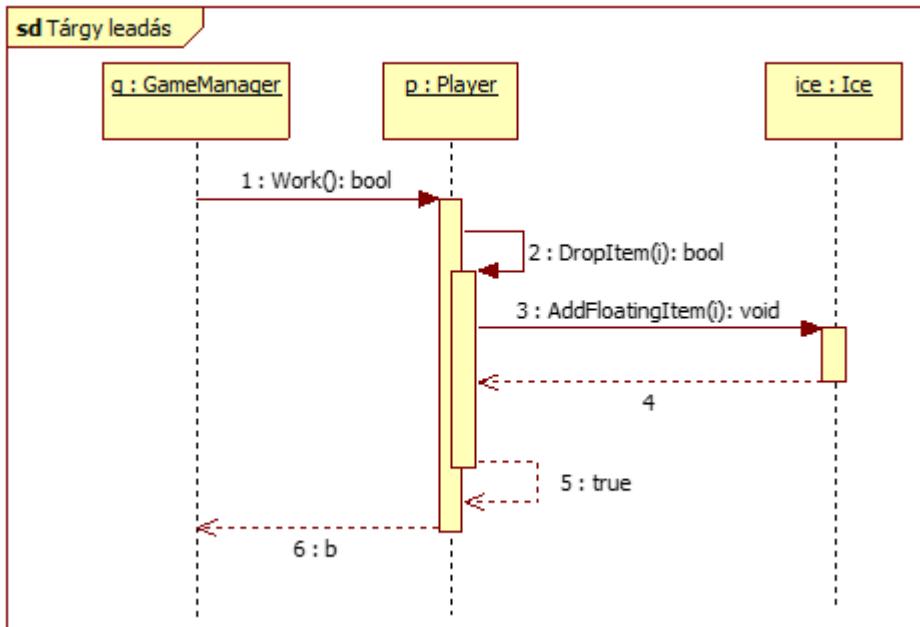
3.4.7

Az alábbi szekvencia diagramon egy játékos mozgása látható



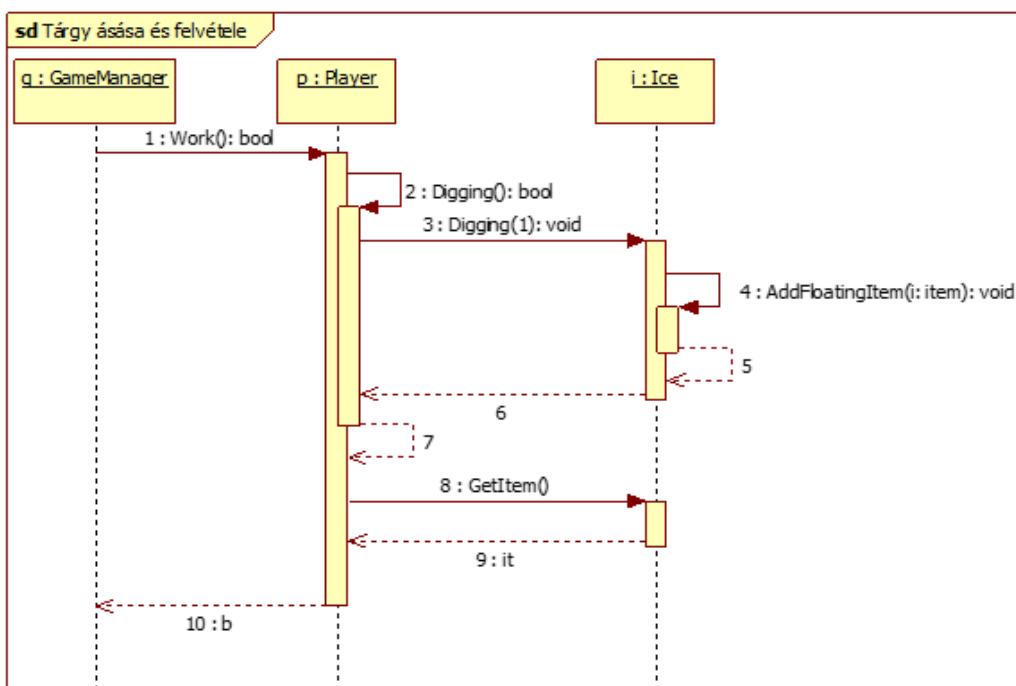
3.4.8

Az alábbi szekvencia diagramon az látható, hogy a játékos eldob egy Item-et és az a jégtábla “levegő Item”-jei közé kerül.



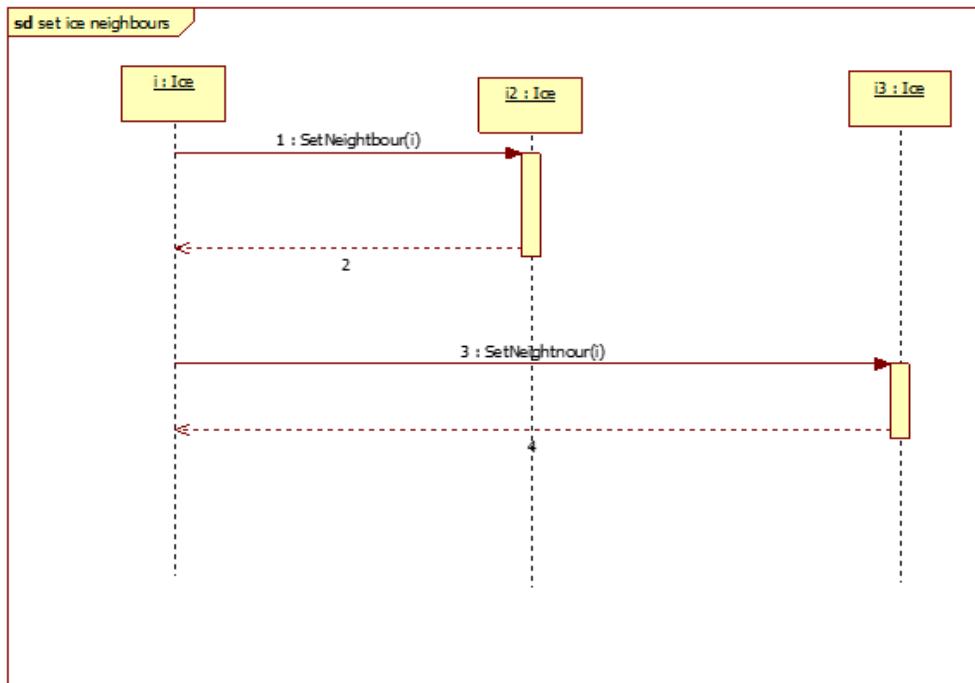
3.4.9

Az alábbi szekvencia diagramon látható egy tárgy kiásása jégből, majd annak felvétele.



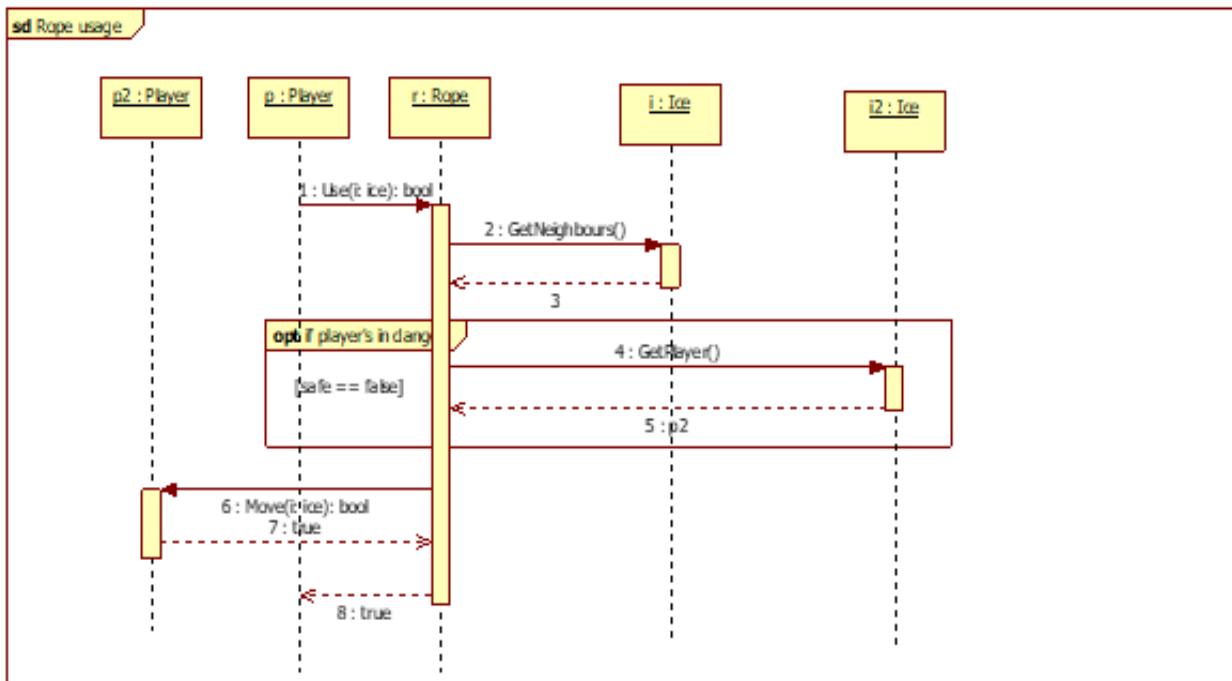
3.4.10

Az alábbi szekvencia diagramon látható a jégtáblák szomszédsági beállítása.



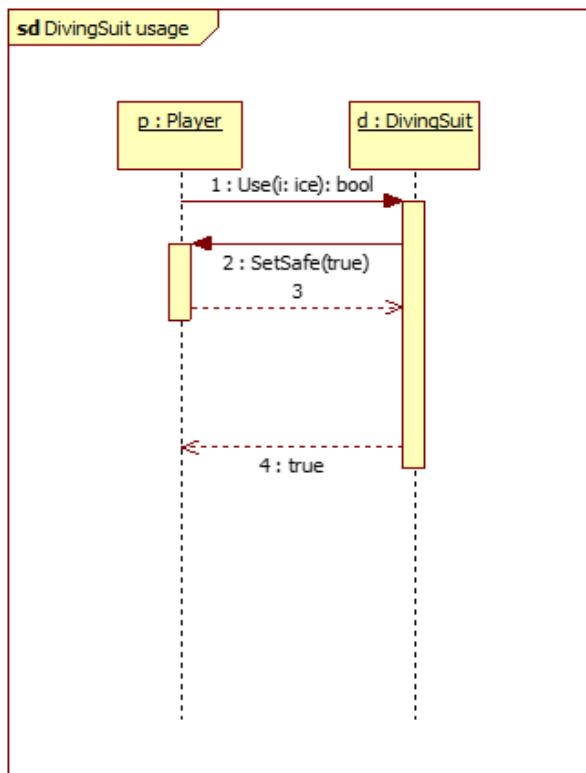
3.4.11

Az alábbi szekvencia diagramon látható a kötél tárgy használata.

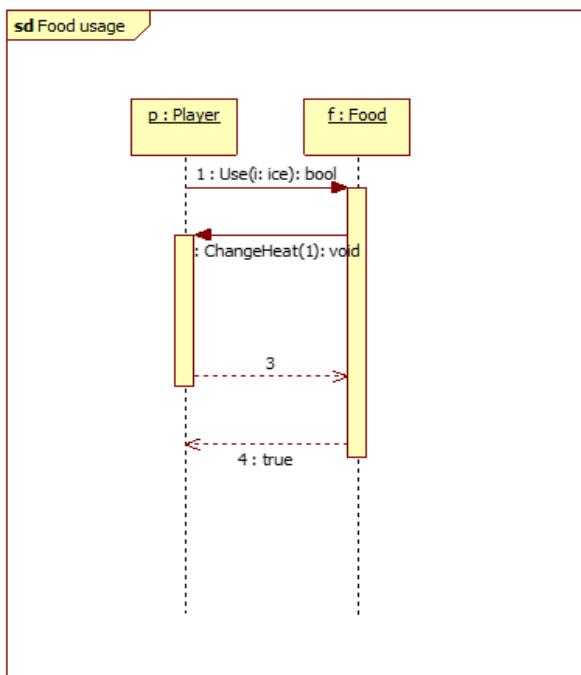


3.4.12

Az alábbi szekvencia diagramon látható a búvárruha használata.

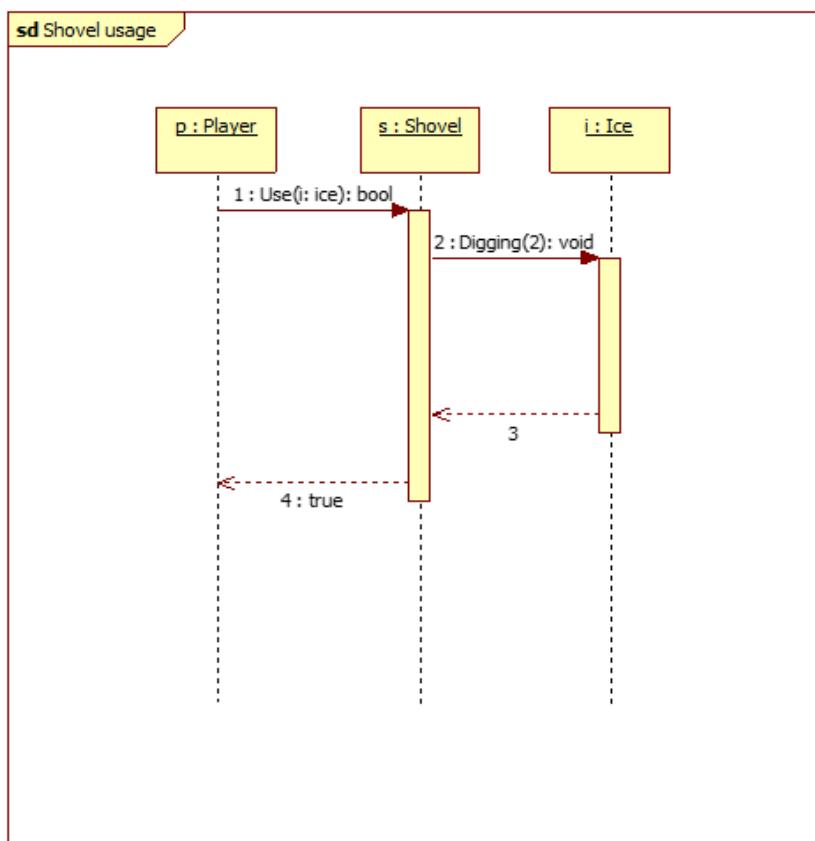
**3.4.13**

Az alábbi szekvencia diagramon látható egy élelem elfogyasztása.



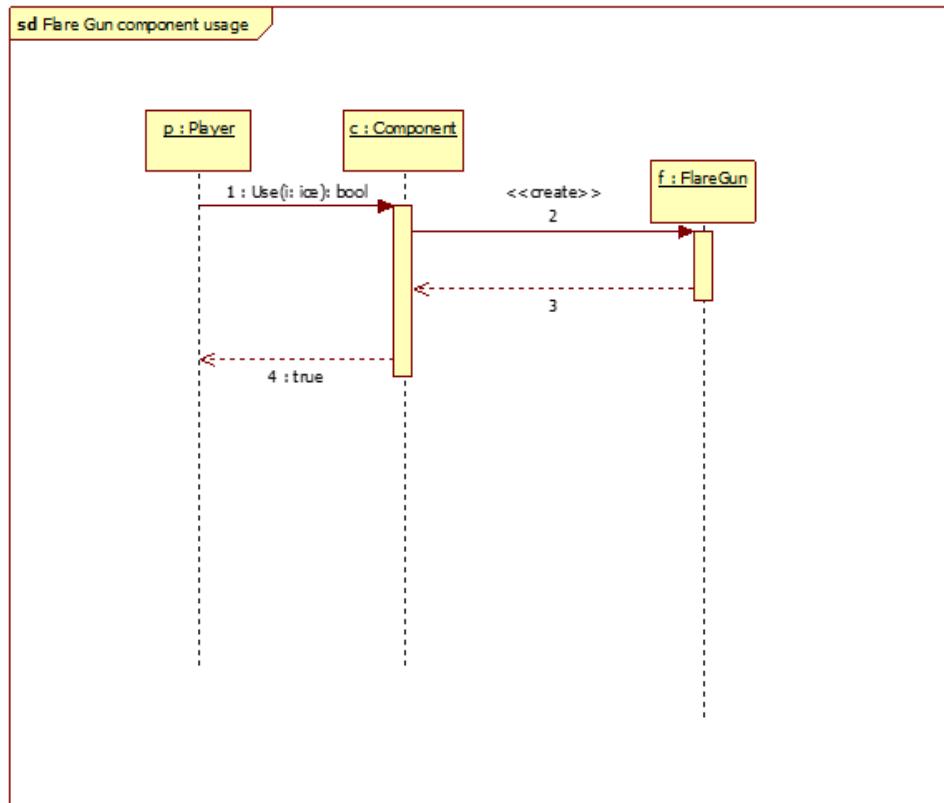
3.4.14

Az alábbi szekvencia diagramon látható, ahogy egy játékos a lapátot használja, amivel két havat is le tud takarítani egyszerre.



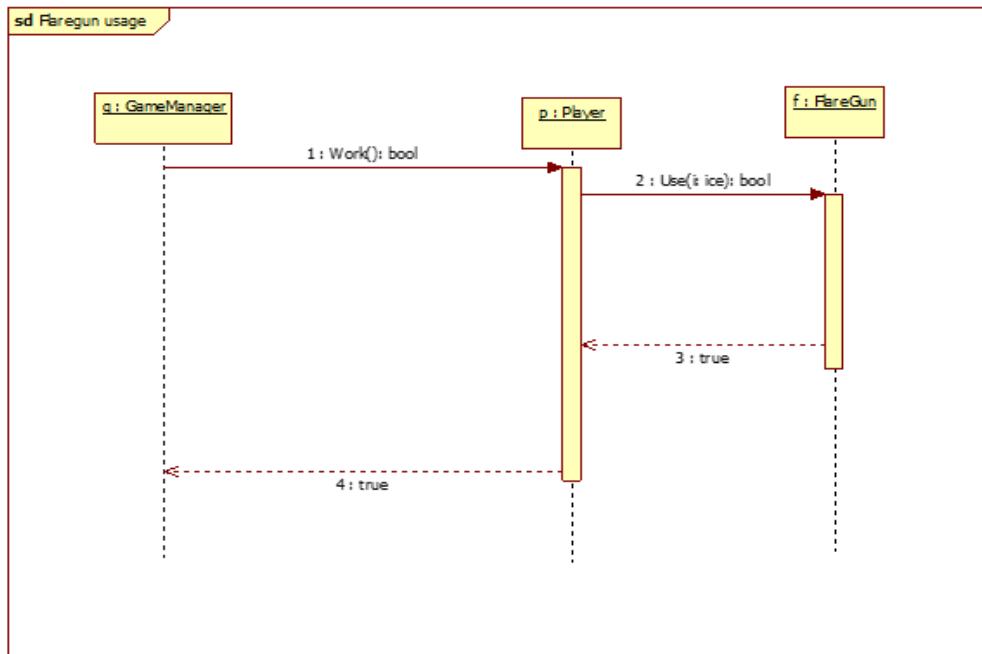
3.4.15

Egy fegyverdarab használatakor (ha van a játékosnál 3db) létrejön a FlareGun



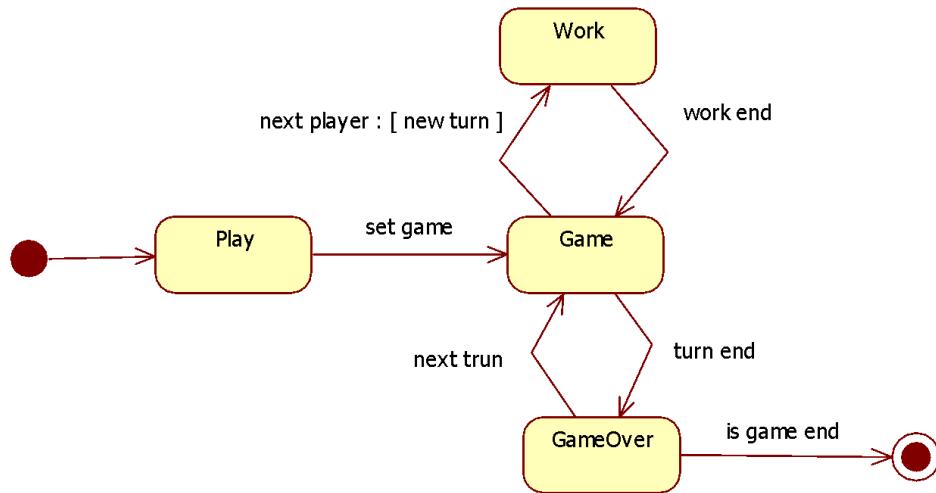
3.4.16

Ha a játékos elszüti a FlareGun-t Work fv. igazzal tér vissza és a játéknak vége.

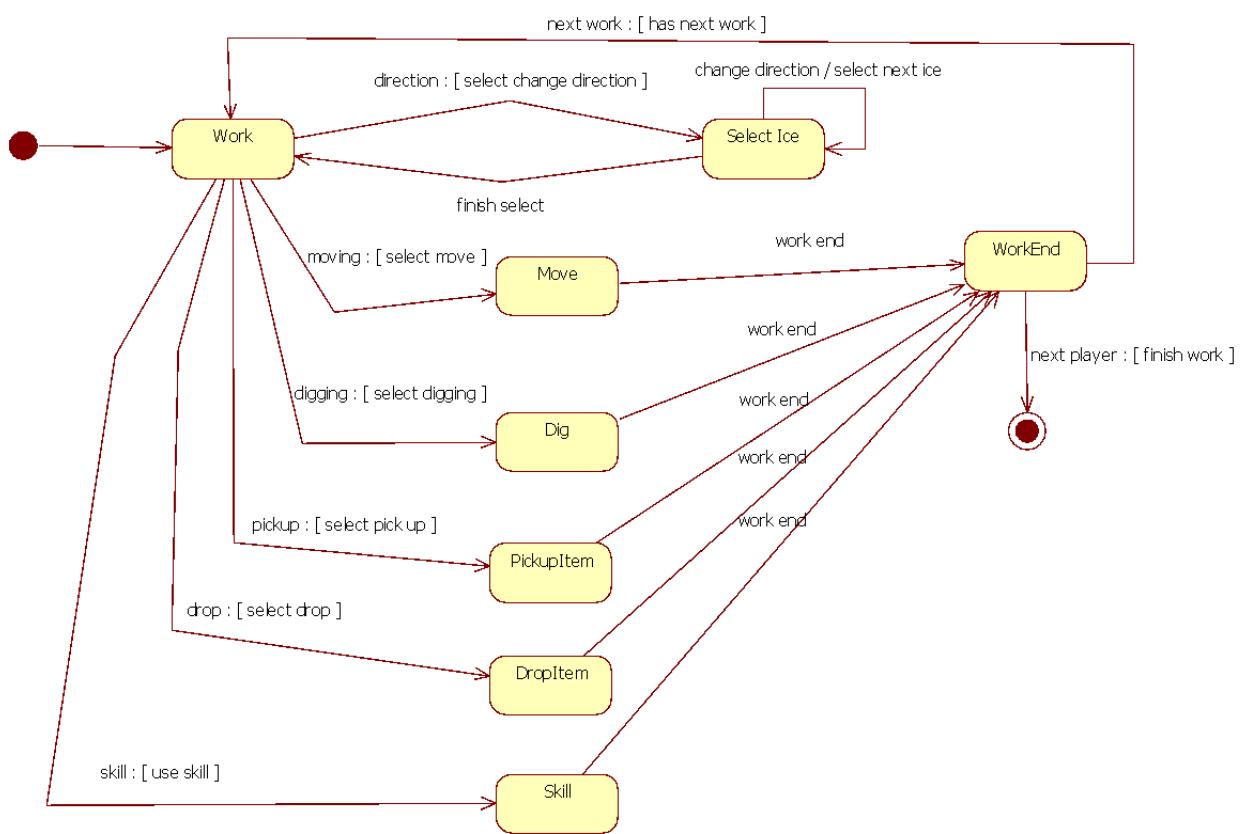


3.5 State-chartok

3.5.1 Game állapotgép



3.5.2 Work állapotgép



3.6

Napló

Kezdet	Időtartam	Részttvevők	Leírás
2020.02.27. 18:00	3 óra	Gaál Bodó Rédai Nagy Varga	Értekezlet. Program mechanika átbeszélése, osztálydiagram felvázolása.
2020.02.28. 21:00	3 óra	Gaál Bodó Rédai Nagy Varga	Osztálydiagram készítése.
2020.02.29. 10:00	2 óra	Gaál	Osztálydiagram javítása
2020.02.29. 14:00	3 óra	Gaál Bodó Nagy	Funkcionalitás átbeszélése, osztálydia javítása
2020.02.29 18:00	2 óra	Bodó Nagy Rédai Varga	Osztálydiagramok javítása Szekvenciadiagram készítése
2020.02.29 21:00	4 óra	Gaál Bodó Nagy Varga	Szekvencia diagram készítésének folytatása
2020.03.01	2 óra	Gaál Bodó Nagy Rédai	Szekvencia diagram javítása Állapotgépek készítése Dokumentum javítása és formázása

4. Analízis modell kidolgozása

4.1 Objektum katalógus

4.1.1 Alkatrész

Három ilyen alkatrész szükséges a jelzőpisztoly összeszereléséhez.

4.1.2 Búvárruha

Viselésével a szereplő életben marad, ha vízbe esett, akár két kör között is, de ebben az esetben csak menni tud, más munkát végezni nem.

4.1.3 Effektus

Az objektum a különböző, más objektumok közötti üzenetcserét fogja megvalósítani. Különböző változatai vannak a hóvihar, és egyéb tárgyak, például a kötél és a pálinka használatának esetére. Az objektum használatával egy adott területen lévő játékosok értesíthetőek ezen különböző eseményekről, de csak ők, és az adott esemény következménye is csak rájuk fog vonatkozni.

4.1.4 Eszkimó

Iglut tud építeni egy adott jégtáblára.

4.1.5 Élelem

Az élelem elfogyasztásával nő a szereplő testhője eggyel.

4.1.6 Épület

Alapvetően minden terület, amire játékos léphet, rendelkezik egy ilyen objektummal. Amennyiben az adott területen hóvihar keletkezik, akkor erről értesítve a játékosokat, azok "bemenekülnek" az adott területen lévő épületbe. Az épület típusától függ, hogy a játékosok, hogyan reagálnak a viharra. Ha az épület egy iglu, akkor semmilyen hatás nem éri őket, amennyiben, az adott épület nem rendelkezik ilyen tulajdonságokkal, akkor a játékosok veszíthatnak testhőjükön.

4.1.7 Fóka

Vízbe esve életben marad két kör között is, de ilyenkor csak mozogni tud, más munkát végezni nem.

4.1.8 Informatikus

Csak ő tudja használni a pálinka tárgyat.

4.1.9 Instabil jégtábla

Ez gyakorlatilag a jég egy speciális változata. Egy fontos dologban különböznek, ha a kapacitásánál több játékos tartózkodik a táblán, akkor az széttörök.

4.1.10 Jelzőpisztoly

Elsütésével vége a játéknak, azt megnyerték.

4.1.11 Jégtábla

A víz egy speciális változata, egy olyan víz ami be van fagyva. Ez egy olyan objektum, melyen állhat tetszőleges számú szereplő és lehet rajta, illetve belefagyva is tárgy. Az eszkimó építhet rajta iglut. Feladata, hogy ismerje a szomszédos jégmezőket, rajta lévő játékosokat és tárgyakat, tudjon játékosokat átadni és fogadni más jégmezőktől, valamint lehet belőle tárgyakat kiásni és tárgyakat elhelyezni rajta.

4.1.12 Kötél

Segítségével ki lehet menteni a szomszédos, beszakadt jégtábláról a bajba jutott társakat, akik a mentés után a kimentőjük jégtáblájára kerülnek. A kötél a mentés után eltűnik.

4.1.13 Lapát

Kétszer annyi havat lehet vele eltakarítani, mint kézzel. A lapátot is csak előre meghatározott alkalommal lehet használni.

4.1.14 Manager

A játék menedzser dolga, hogy a játék elején elkészítse a játékteret és kisorsolja a tárgyak kezdeti helyét. Játék közben körönként bizonyos lehetőséggel generál hóvihart és egy kör ideje alatt egymás után ad lehetőséget a szereplöknek munkára. Ha a szereplők valamelyike összeszereli a jelzőpisztolt, akkor az jelzi a menedzser felé a győzelmet és a játék véget ér.

4.1.15 Pálinka

Csak az Informatikus tudja meginni. Használatkor az informatikus jégtábláján minden szereplőnek a testhője beáll a lehető legmagasabb szintre.

4.1.16 Sarkkutató

Meg tudja vizsgálni egy szomszédos jégtábla teherbírást.

4.1.17 Szereplő

Egy karakter, amit a felhasználó irányít. Tulajdonságai, hogy legfeljebb 4 tárgyat tud tárolni és a játék kezdetén van bizonyos mennyiségi testhője. Továbbá képes jégtáblákon mozogni, havat ásni, nála lévő tárgyakat használni és vízbe tud esni. Meg tud halni, ha elfogy a testhője, vagy bűvárruha nélkül vízbe esik és nem mentik ki. Ez a játék végét is jelenti. A szereplőnek típusai vannak, és azoknak egyéni képessége.

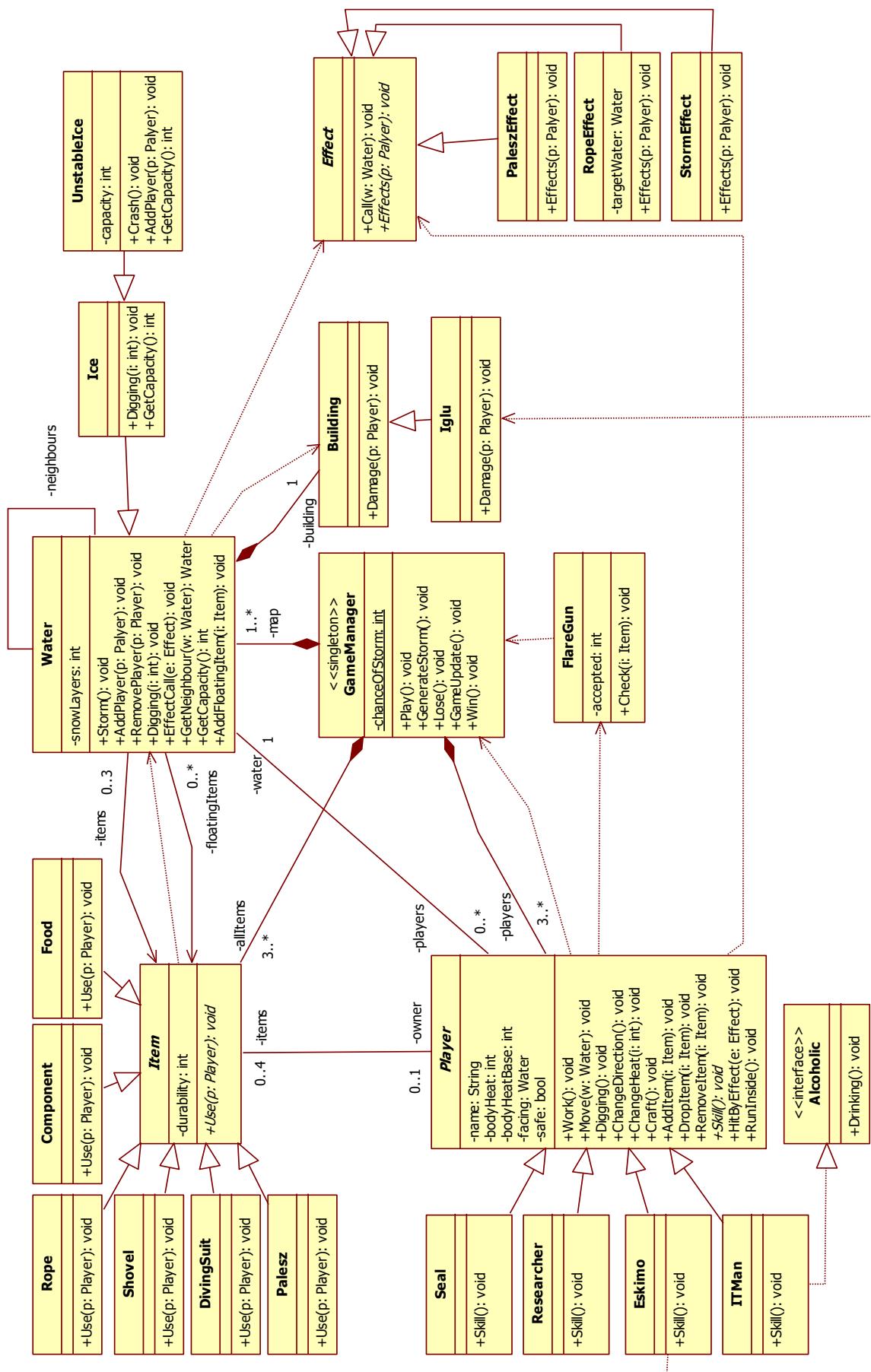
4.1.18 Tárgy

Egy olyan eszköz, amit használva, az különböző képességekkel ruházza fel a tulajdonosát.

4.1.19 Víz

A mezők alapegysége. Ez az ōse a jégtáblának. Ezt az objektumot több höréteg is takarhatja

4.2 Statikus struktúra diagramok



4.3 Osztályok leírása

4.3.1 Alcoholic

- **Felelősség**
A megvalósító objektum képes lesz a Palesz item-et használni. (ezáltal megnövelni a jégtömbön álló játékosok testhőjét)
- **Metódusok**
 - **void Drinking():** Használatkor egy PaleszEffect jön létre ami a jégtáblán álló játékosok testhőjét maximumra állítja.

4.3.2 Building

- **Felelősség**
A Water táblán elhelyezkedő “épület”. A játékosok ide menekülhetnek be hóvihar esetén.
- **Metódusok**
 - **void Damage(Player p):** Az épületbe berohanó játékos testhőjét csökkenti.

4.3.3 Component

- **Felelősség**
Három komponensből állítható elő a FlareGun.
- **Ősosztályok**
Item
- **Metódusok**
 - **void Use(Player p):** Ha egy játékos használja és van a birtokában összesen 3 komponens egy FlareGun jön létre (craftolás) ami autómatikusan elsül és vége a játéknak.

4.3.4 DivingSuit

- **Felelősség**
Ha egy játékos viseli és vízbe esik, túléli. Ha csak a felszerelésében van, de még nem használta akkor nem.
- **Ősosztályok**
Item
- **Metódusok**
 - **void Use(Player p):** A használó játékos felveszi a búvárruhát, ezzel eltűnik a tárgy a felszereléséből és a játékos safe attribútuma true-ra változik.

4.3.5 Effect

- **Felelősség**
Absztrakt osztály. A jégtömbön álló játékosokat befolyásoló történés. (testhőváltozás, kimentés)
- **Metódusok**
 - **void Call(Water w):** Egy hívás a célzott területen (Water-en). Később az a terület irányítja tovább a hívást a játékosokhoz. (lásd Water osztály.)
 - **void Effects(Player p):** Maga a hatás, mely a játékost befolyásolja.

4.3.6 Eskimo

- **Felelősség**
Iglut tud építeni arra a jégtáblára, amin áll. Az Iglu csak az adott körben érvényes, a következő kör elején eltűnik.
- **Ősosztályok**
Player
- **Metódusok**
 - **void Skill():** Épít egy Iglut a saját jégtáblájára. Az Iglu egy körön át marad ott.

4.3.7 FlareGun

- **Felelősség**
A komponensek összerakásával keletkezik a játékban. Automatikusan elsül és vége a játéknak.
- **Attribútumok**
 - **accepted:** Egy játékos mikor szereli össze a FlareGun-t, azt számolja hány Komponenst adott át eddig a játékos.
- **Metódusok**
 - **void Check(Item i):** A játékos átadja a tárgyat, melyről kiértékelődik komponens-e. Ha talál egyet növeli az accepted tagváltozót. Ha meg van a 3 elsül a fegyver, vége a játéknak.

4.3.8 Food

- **Felelősség**
Használatával az azt használó játékos testhője megnő eggyel.
- **Ősosztályok**
Item
- **Metódusok**
 - **void Use(Player p):** A játékos megeszi és megnöveli a testhőjét eggyel.

4.3.9 GameManager

- **Felelősség**
A játékot vezérli. Generálja a játékot, körönként meghívja a játékosokat, hogy mozogjanak, végezznek munkát. A körök végén véletlenszerűen hóvihart hoz létre, megnézi, hogy meghalt-e valaki, valamint ha egy játékos elsüti a jelzőpisztolyt, akkor jelezni tudja ezen objektumnak, ami leállítja a játékot.
- **Attribútumok**
 - **int chanceOfStorm:** A hóvihar keletkezésének esélye. Körönként nő.
 - **Player players[]:** A játékosok listája.
 - **Water map[]:** A játékban szereplő összes mező.
 - **allitems[]:** A játékban szereplő összes tárgy listája.
- **Metódusok**
 - **void Play():** A játékosok egymás utáni játszatásáért felelős.
 - **void GenerateStorm():** Kiválasztva egy (nagyobb pálya esetén több) mezőt, arra és annak szomszédaira egy hóréteget tesz, továbbá az azonon a táblákon tartózkodó játékosok testhőjét eggyel csökkenti (ha nincs iglu). A hóvihar esélye körönként nő.
 - **void GameUpdate():** A játékosok sorrendje minden kör után változik. (A kör végéig még egy éppen vízben lévő játékos még egy soron lévő játékos ki tud menteni. Ha éppen a kör végén esik valaki a vízbe akkor őt már nem lehet megmenteni. A cserére azért van szükség hogy ne minden ugyanaz a játékos kerüljön ilyen helyzetbe, hogy már a kör végén nem lehet megmenteni.)
 - **void Lose():** minden kör végén kiértékeli hogy vége van-e a játéknak. Egy játék véget ér, ha van olyan játékos akinek a testhője nem több 0-nál vagy egy játékos úgy maradt vízben hogy nincs búvárruhája és nem fóka.
 - **void Win():** A jelzőpisztoly elsütésével ez a függvény hívódik meg, ami leállítja a játékot.

4.3.10 Ice

- **Felelősség**
A stabil jégtábla. A játékos tud rajta bármilyen munkát végezni. Nem tud beszakadni, tehát akárhányan állhatnak rajta.
- **Ősosztályok**
Water
- **Metódusok**
 - **void Digging(i: int):** Letakarít i hóréteget a jégtábláról. Ha már nem lenne hó, akkor pedig egy tárgyat helyez ez a floatingItem-ek közé.
 - **int GetCapacity():** Visszadja mekkora a teherbírása a jégtáblának.

4.3.11 Iglu

- **Felelősség**
Ezen objektum gyakorlatilag egy épület típus. Ha egy mező rendelkezik ilyen épülettel, akkor hóvihar esetén az adott mezőn álló játékosok bemenekülhetnek az épületbe, ami megvédi őket a hóvihar hatásaitól.
- **Ősosztályok**
Building
- **Metódusok**
 - **void Damage(Player p):** Hóvihar esetén ezen függvény hívásával tud az adott felhasználó “bemenni” az épületbe. A függvény gyakorlatilag a hóvihar hatását valósítja meg az épülettől függően. Ennél az épülettípusnál a hóvihar semmilyen hatását nem érzékelik az adott mezőn álló játékosok.

4.3.12 Item

- **Felelősség**
Absztrakt osztály. Egy eszköz, ami a birtoklójának különböző tevékenységek elvégzésére adnak lehetőséget. El lehet őket dobni egy jégtáblára, fel lehet őket venni jégtábláról, ki lehet őket ásni jégtáblákból.
- **Attribútumok**
 - **int durability:** A tárgy “életereje”. 0-ra csökkenéskor a tárgy elhasználódik és eltűnik. minden leszármazottnak egyedi értékű.
 - **Player owner:** A játékos, aki nél a tárgy van.
- **Metódusok**
 - **void Use(Ice i):** Absztrakt metódus. minden tárgynak egyedi használata van.

4.3.13 ITMan

- **Felelősség**
Meg tudja inni a Pálinkát.
- **Ősosztályok**
Player
- **Interfész:**
Alcoholic

- **Metódusok**

- **void Skill()**: Üres metódus. Saját egyedi képessége nem kerül megvalósításra, hanem az interfésze és a Palesz item-je segítségével.

- **void Drinking()**: Megissza a nála lévő Pálinkát. Azon a jégtáblán, amin áll, minden játékos testhője visszaáll a maximumra (bodyHeatBase).

4.3.14 Palesz

- **Felelősség**

Csak az ITMan tudja meginni. Használatkor a használó jégtábláján álló játékosok testhője maximumra nő.

- **Ősosztályok**

Item

- **Metódusok**

- **void Use(Ice i)**: Használatakor meghívja az ITMan Drinking függvényét.

4.3.15 PaleszEffect

- **Felelősség**

A jégtábla hívódik, mely továbbítja a kérést. Az “eltatlált” játékosokat testhőjét maximumra állítja.

- **Ősosztályok**

Effect

- **Metódusok**

- **void Effects(Player p)**: A játékos bodyHeatBase adattagját felhasználva visszaállítja maximumra a testhőjét.

4.3.16 Player

- **Felelősség**

Absztrakt osztály. Egy ember által irányított játékos reprezentál. Tud lépni, munkát végezni és meg tud halni. Egy munkának számít egy tárgy használata és a karaktertípus saját képességének használata is.

- **Attribútumok**

- **String name**: A játékos neve.
- **int bodyHeat**: A játékos testhője. Ha nullára csökken, meghal.
- **int bodyHeatBase**: A játékos maximális testhője, ennél soha nem lehet több. A játék elején ennyivel indul.
- **Water water**: Az a jégtábla/víz (vagyis mező), amin a játékos éppen áll.
- **Item items[4]**: A játékosnál lévő tárgyak.
- **Water facing**: Az a mező, amire a játékos éppen mutat/néz, ez néhány képesség és a mozgás megvalósításához szükséges.

- bool safe: Ha igazra van állítva az azt jelenti, hogy az adott játékos vízbeeséskor nem hal meg, ha nem menekítik ki a kör végéig. (Akkor vált igara a változó, ha az adott játékos egy fóka, vagy használta a búvárruhát.)

- **Metódusok**

- **void Work()**: A játékost dolgoztató függvény. Ha a játékos vízben van, akkor búvárruha nélkül nem dolgozhat. Ha van búvárruhája akkor a vízben csak mozoghat. A jégtáblán bármilyen munkát végezhet.
- **void Move(Water w)**: A játékos átléphet egy a paraméterben szereplő másik mezőre.
- **void Digging()**: A játékos vagy havat takarít a jégtábláról amin áll, vagy egy tárgyat ás ki belőle.
- **void ChangeDirection()**: A játékos irányváltására szolgál. Meghívva, módosítja a játékos facing attribútumát a saját jégtáblájának szomszédlistájában szereplő facing-et követő tagra (így ezt a függvényt hívogatva az összes szomszédos jégtáblát el lehet érni).
- **void ChangeHeat(int i)**: A játékos testhőjét változtatja a kapott értékkel.
- **void Craft()**: A jelzőpisztoly összeszerelését megvalósító függvény, csak egy komponens use()-a hívhatja meg. Megpróbál elkészíteni egy jelzőpisztolyt a játékos saját tárgyaiból, ha sikerül, akkor a játék véget ér, ha nem, akkor a játék folytatódik.
- **void AddItem(Item i)**: Egy azonos jégtáblán lebegő i Itemet felvesz a játékos felszerelése közé.
- **void DropItem(Item i)**: A játékos lerak egy tárgyat a felszerelésből arra a jégtáblára amin áll. Onnantól kezdve az a tárgy "lebeg" a jégtáblán.
- **void RemoveItem(i: Item)**: Elveszi a kapott tárgyat a játékos felszerelésből.
- **void Skill()**: Absztrakt metódus, a leszármazottak egyedi képességének használatához.
- **void HitByEffect(e: Effect)**: Az adott táblára érkező effektus "eltalálja" a rajta álló játékosokat, ami ezen függvény hívásával történik. Az effektus típusától függően valamilyen hatást vált ki a játékosra.
- **void RunInside()**: A függvény használatával a játékos "bemenekülhet" a mezőn lévő épületbe hóvihar esetén. (Ez a függvény effektus érkezése esetén automatikusan hívódik.)

4.3.17 Researcher

- **Felelősség**

Meg tudja vizsgálni egy szomszédos jégtábla teherbírását.

- **Ősosztályok**

Player

- **Metódusok**

- **void Skill()**: Megvizsgál a facing által meghatározott szomszédos jégtáblát és visszakapja annak a teherbírását (hány embert bír el).

4.3.18 Rope

- **Felelősség**
Ki lehet vele menteni egy szomszédos jégtáblán vízbeesett játékosokat.
- **Ősosztályok**
Item
- **Metódusok**
 - **void Use(Player p):** Használatával a facing által meghatározott szomszédos jégtábára RopeEffect-et hoz létre. A vízbeesett játékosokat kimenekíti a saját jégtáblájára (mindenkit).

3.3.19 RopeEffect

- **Felelősség**
A jégtábla hívódik, mely továbbítja a kérést. Az “eltatlált” játékosok, akik vízben vannak és nincsenek biztonságban (safe) kimenekíti.
- **Ősosztályok**
Effect
- **Attribútumok**
 - **Water targetWater:** Ide kell kimenekíteni a játékosokat. Ezen a jégtáblán áll a kimentő játékos aki használta a kötelet.
- **Metódusok**
 - **void Effect(Player p):** A játékosra van hatással, meghívja rájuk a move-ot a targetWater irányba.

4.3.20 Seal

- **Felelősség**
A kör végén búvárruha nélkül is túléli a vízbeesést.
- **Ősosztályok**
Player
- **Metódusok**
 - **void Skill():** Üres metódus. A különleges képessége karakterből adódóan automatikusan működik.

4.3.21 Shovel

- **Felelősség**
Használatával egyszerre két réteg havat lehet ellapárolni arról a jégtábláról, amin a játékos áll. Használata csökkenti a durability-t. 0 durability elérésekor eltűnik a játékos felszereléséből.
- **Ősosztályok**
- Item
- **Metódusok**
- **void Use(Player p):** A használó jégtáblájáról eltávolít két réteg havat. Ezzel csökken a durability. Ha a durability 0 vagy kevesebb, a Shovel eltűnik.

4.3.22 StormEffect

- **Felelősség**
Egy jégtáblára hívódik, a rajta lévő játékosok testhőjét csökkenti. (közvetlenül csak a játékosokat beküldi az épületbe)
- **Ősosztályok**
Effect
- **Metódusok**
- **void Effects(Player p):** Az “eltalált” játékost beküldi a jégtáblán (Water) lévő épületbe.

4.3.23 UnstableIce

- **Felelősség**
Azt a jégtáblát valósítja meg, amin már csak egy bizonyos számú ember állhat és ha többen lépnek rá, akkor beszakad.
- **Ősosztályok**
Ice
- **Attribútumok**
- **int capacity:** Az adott jégtábla teherbírása, megadja, hogy egyszerre maximum hány ember állhat rajta.
- **Metódusok**
 - **AddPlayer(Player p):** Egy paraméterben megadott játékost hozzáad a saját magán álló játékosok listájához. Az örökölt ugyanilyen nevű metódust definiálja felül.
 - **Crash():** Az AddPlayer(Player p) metódus hívja meg. A függvény ellenőrzi, hogy a jégtáblán többen állnak-e, mint amennyit elbír és ha igen, akkor “beszakad”. (Egy új Water példány jön létre a helyén.)
 - **int GetCapacity():** Visszatér capacity értékkel, vagyis hogy hányan állhatnak egyszerre az adott jégtáblán.

4.3.24 Water

- **Felelősség**

Egy területegységet reprezentál, ami tárolhat tárgyakat és játékosokat, valamint fedheti valamennyi réteg hó.

- **Attribútumok**

- **int snowLayers:** A jégtáblán ennyi réteg hó van.
- **Player players[]:** A jégtáblán tartózkodó játékosok száma.
- **Water neighbours[]:** A szomszédos jégtáblák.
- **Item items[]:** A jégbefagyott tárgyakra referenciák.
- **Item floatingItems[]:** A jégtáblán “lebegő” tárgyakra referenciák.
- **Building building:** Az adott területen lévő épület.

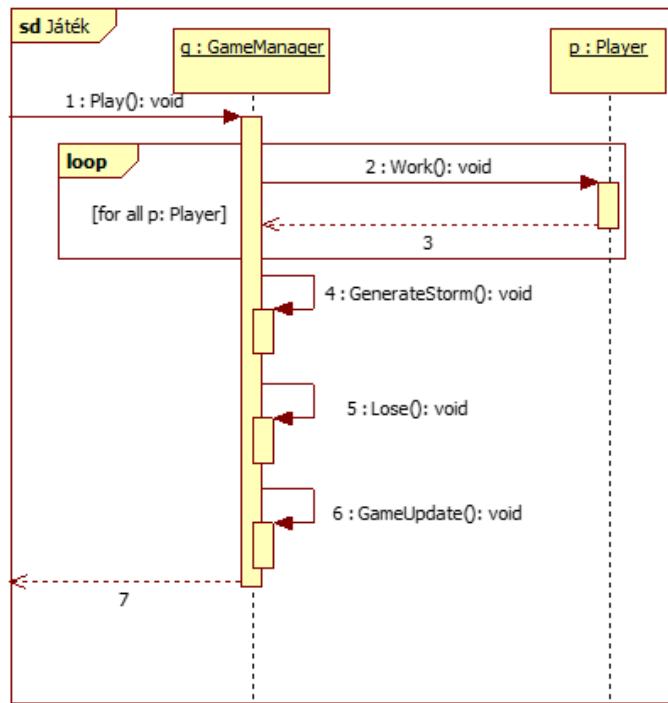
- **Metódusok**

- **void Storm():** Hóvihar tombol a jégmezőn. (epicentrum) A hóréteg megvastagodik, illetve a mezőn, és a szomszédos mezőn álló játékosokat értesíti a hóviarról.
- **void Digging(int i):** Eltávolít i darab hóréteget. Ha a hórétegek száma 0-ra csökken, tárggyal tér vissza további ásás esetén.
- **void AddFloatingItem(Item i):** A lebegő tárgyakhoz hozzáad egy új tárgyat.
- **void RemovePlayer(Player p):** Eltávolít egy játékost a jégtábláról.
- **void AddPlayer(Player p):** Hozzáad egy játékost a jégtáblához.
- **void EffectCall(Effect e):** A jégtáblán álló játékosokat “eltalálja” az effektus.
- **Water: GetNeighbour(Water w):** Visszatér az adott mező sorban következő szomszédjával, vagyis azzal ami sorban a paraméterként kapott (facing) mező után jön. (Egyirányú körbeforgást tesz lehetővé.)

4.4 Szekvencia diagramok

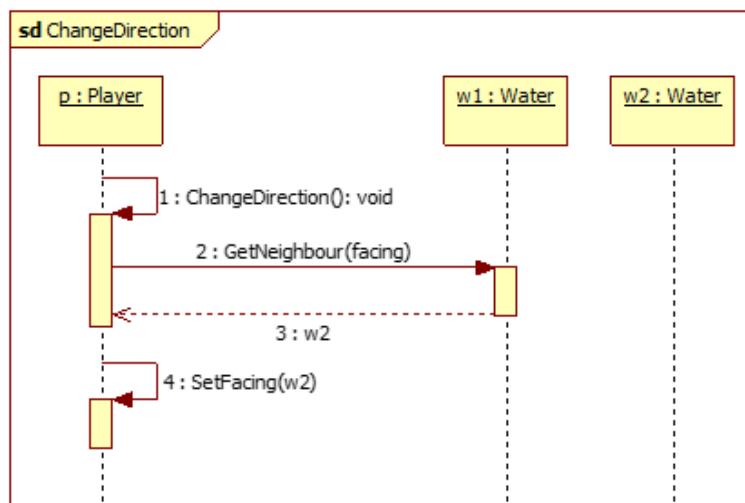
4.4.1

Egy kör lefolyása a játékban. A GameManager Play függvénye játszatja az összes játékosot addig, amíg el nem fogy az egyes játékosok munkái. Majd a hóvihar esélye változik. (Az is lehet, hogy hóvihar keletkezik). Majd a Lose fv. vizsgálja, hogy vége van-e a játéknak. Majd a játékosok sorrendjét cseréli a GameUpdate függvény.



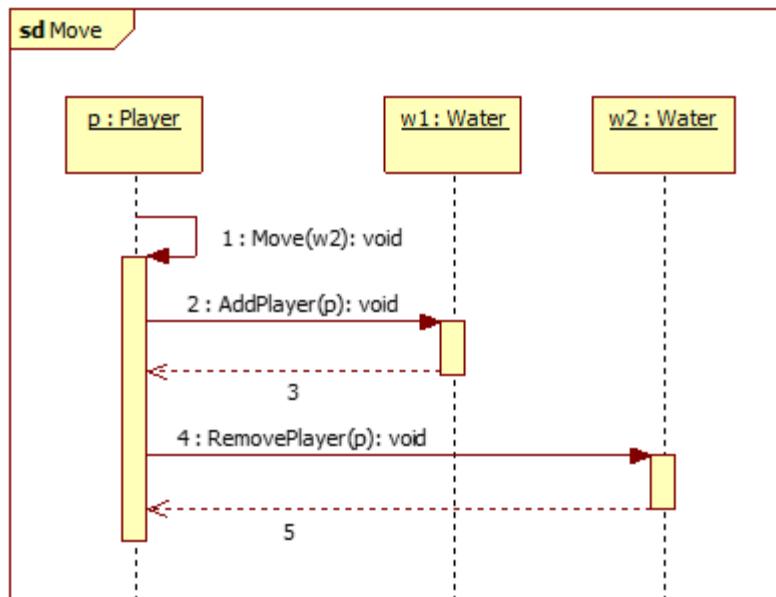
4.4.2

Megkérdezi a jégtáblától, hogy melyik jégtábla van amellett a jégtábla mellett melyre a játékos néz (facing). Beállítja hogy a játékos melyik irányba néz.



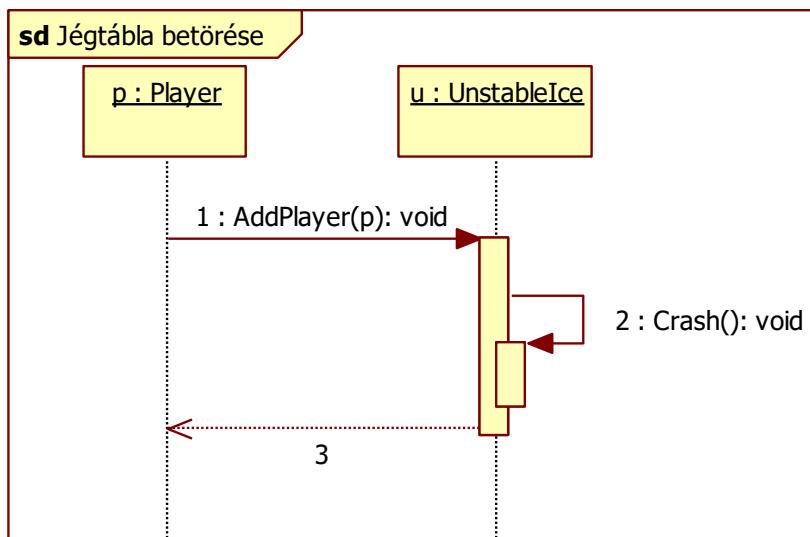
4.4.3

Az alábbi szekvencia diagramon látható, hogy egy játékos hogyan lép át egy másik mezőre, arra amire a lépéskor néz.



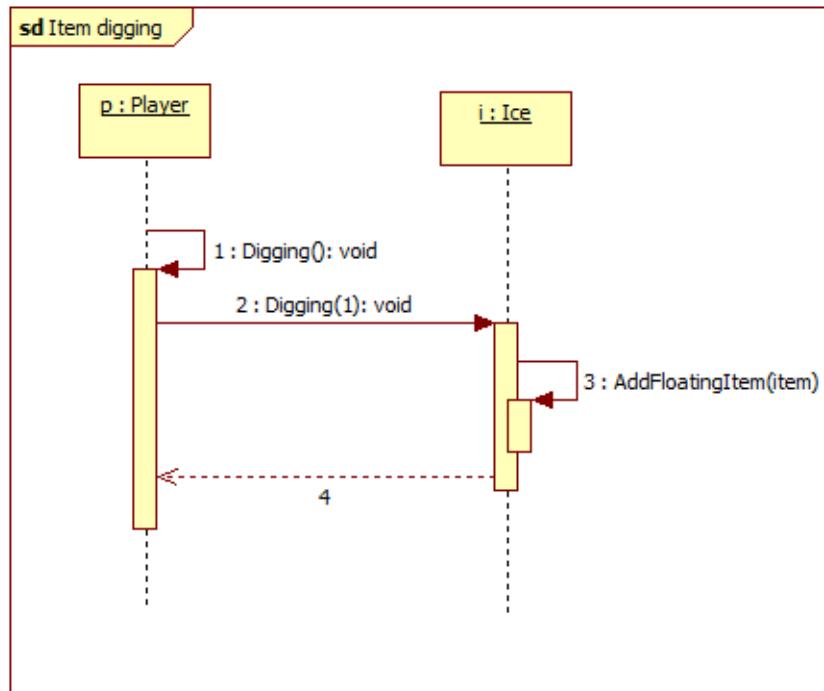
4.4.4

Amikor egy játékos átkerül egy másik unstable jegtáblára akkor az betörhet.

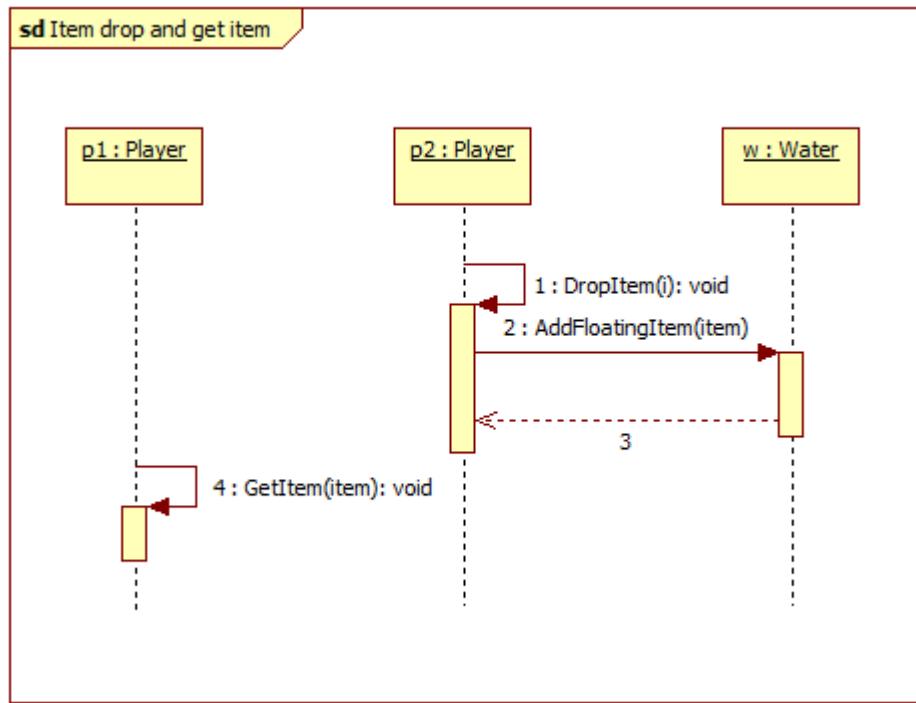


4.4.5

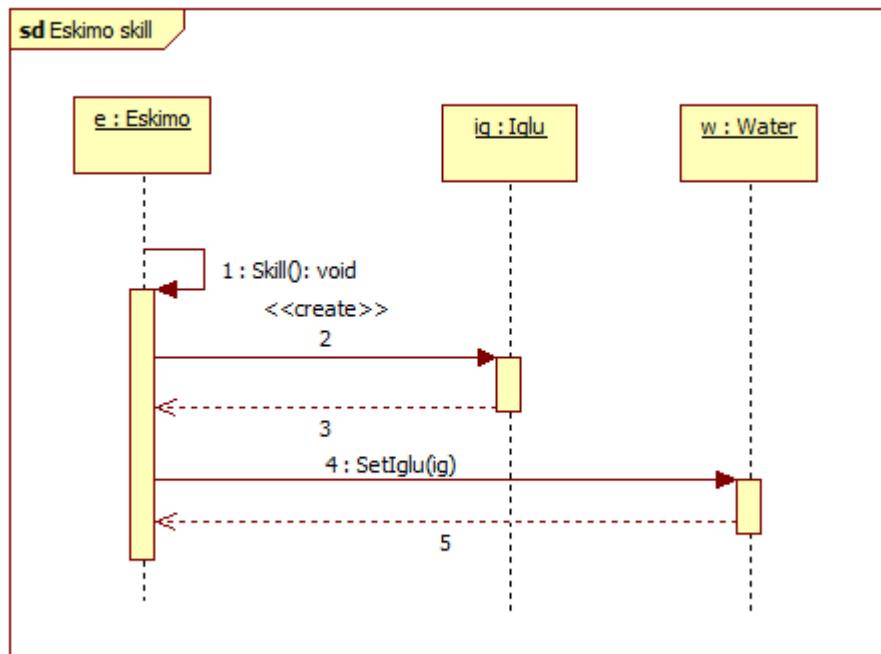
A játékos kiás egy Item-et a jégtömbből. Az Item a floatingItem-ekhez kerül.

**4.4.6**

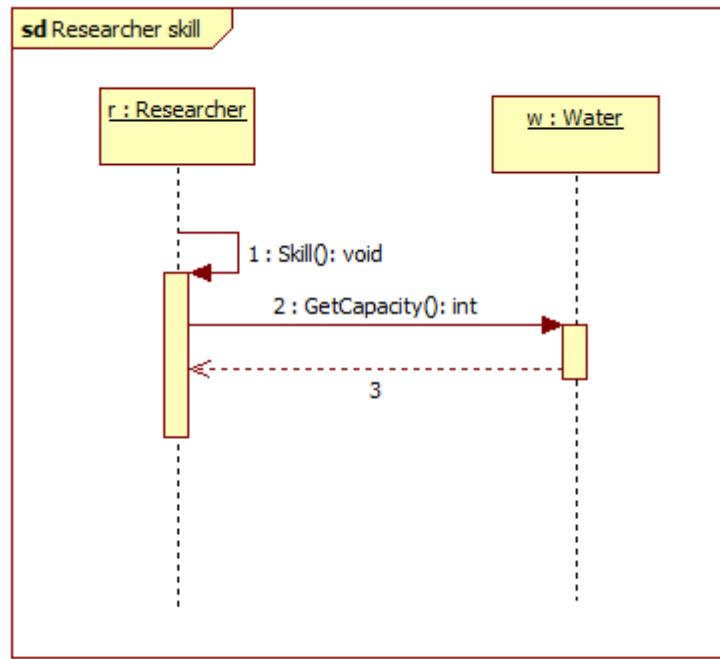
A játékos kaphat egy új itemet a felszerelései közé, de akár el is dobhat egyet, ilyenkor az eldobott item az adott jégtáblához kerül.

**4.4.7**

Az alábbi szekvencia diagramon látható az eszkimó képességének a használata.

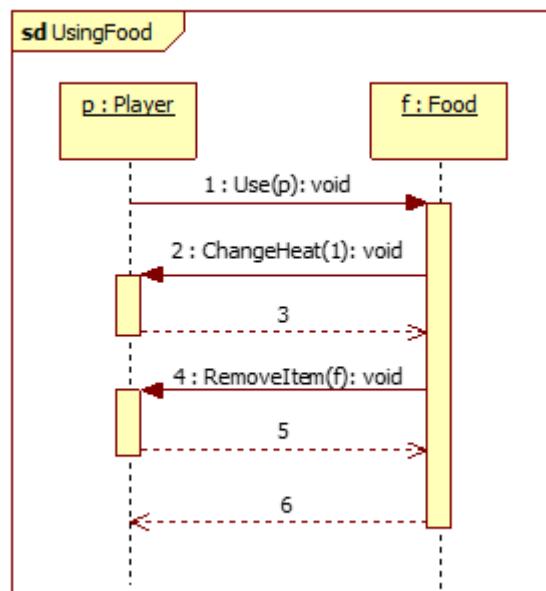
**4.4.8**

Az alábbi szekvencia diagramon látható a kutató képességeinek a használata



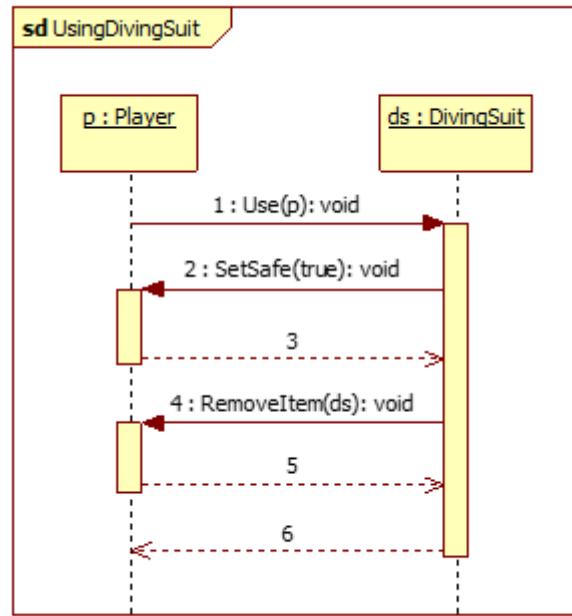
4.4.9

Egy Food elfogyasztásával megnő a játékos testhője eggyel.



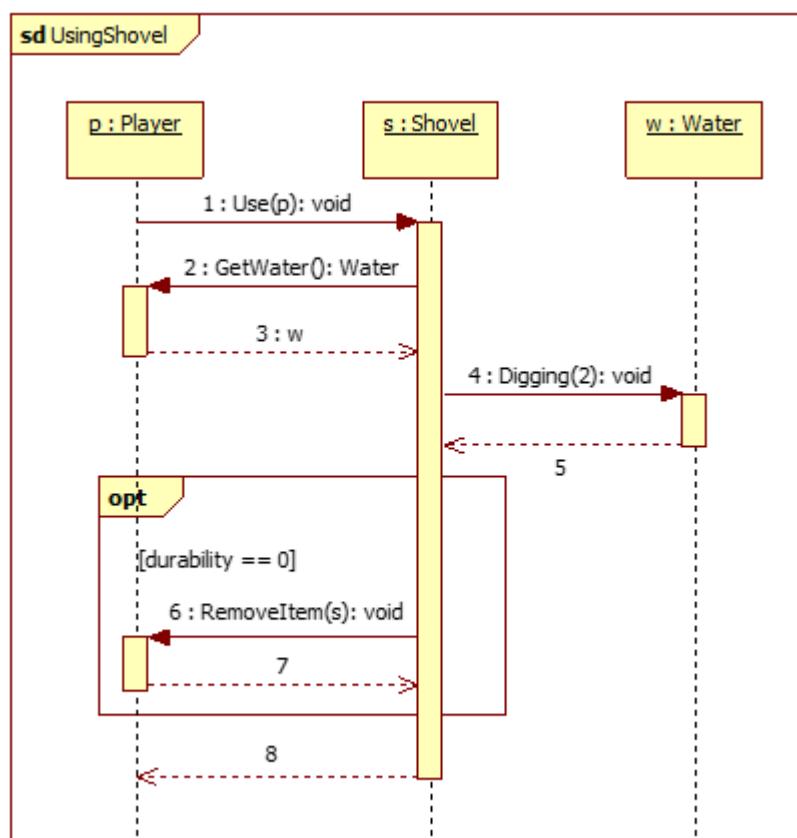
4.4.10

A DivingSuit item használatával a játékos safe-e igazra állítódik és az item eltűnik.



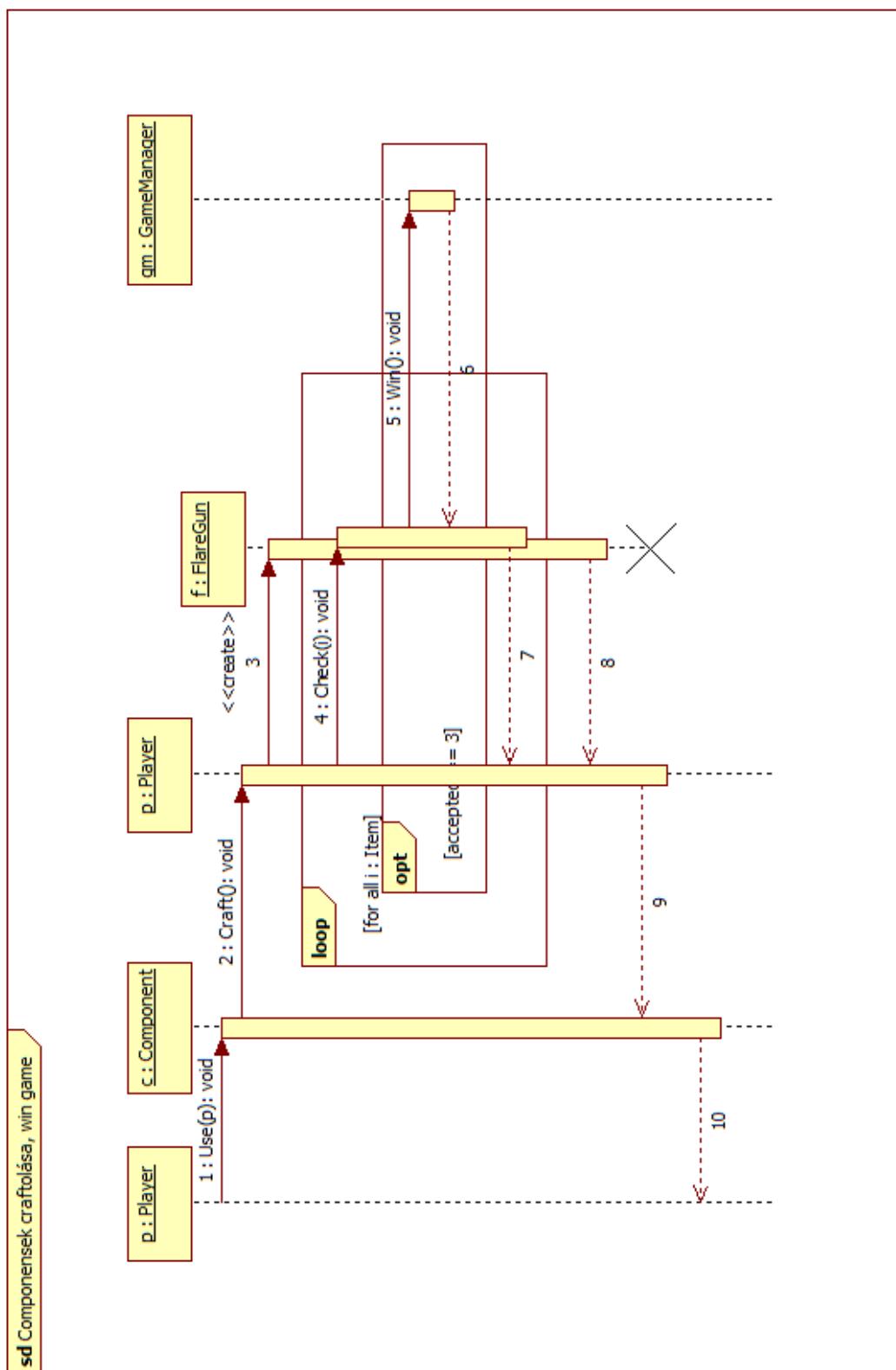
4.4.11

A lapát item használatával egyszerre két réteg hó eltakarítható egy munka felhasználásával.



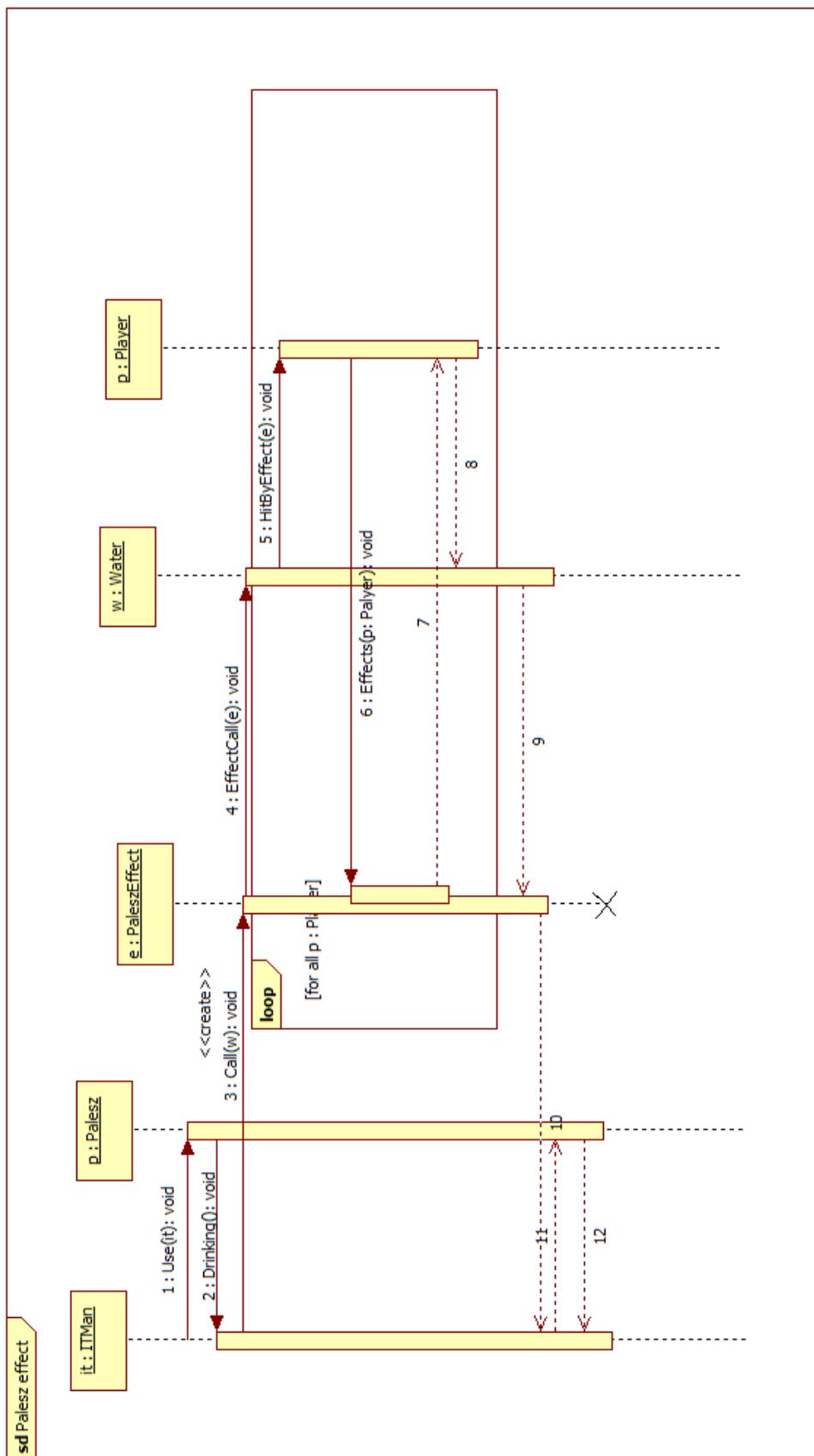
4.4.12

3 komponens felhasználásával összeállítható a jelzőpisztoly, ezzel megnyerve a játékot.



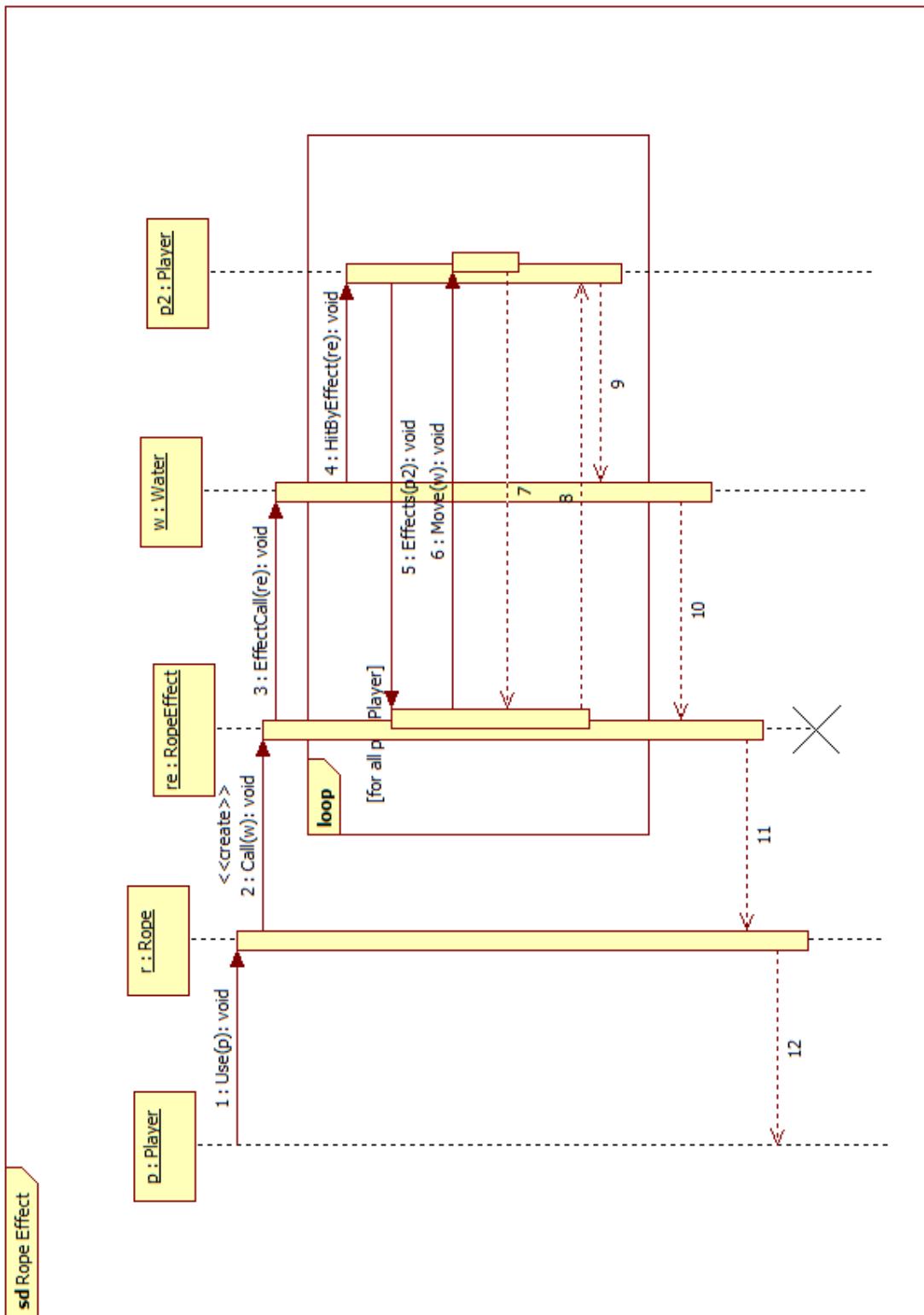
4.4.13

Az informatikus használhatja a Palesz itemjét, ekkor az adott jégtáblán minden játékosnak visszaáll eredeti értékre a testhője.



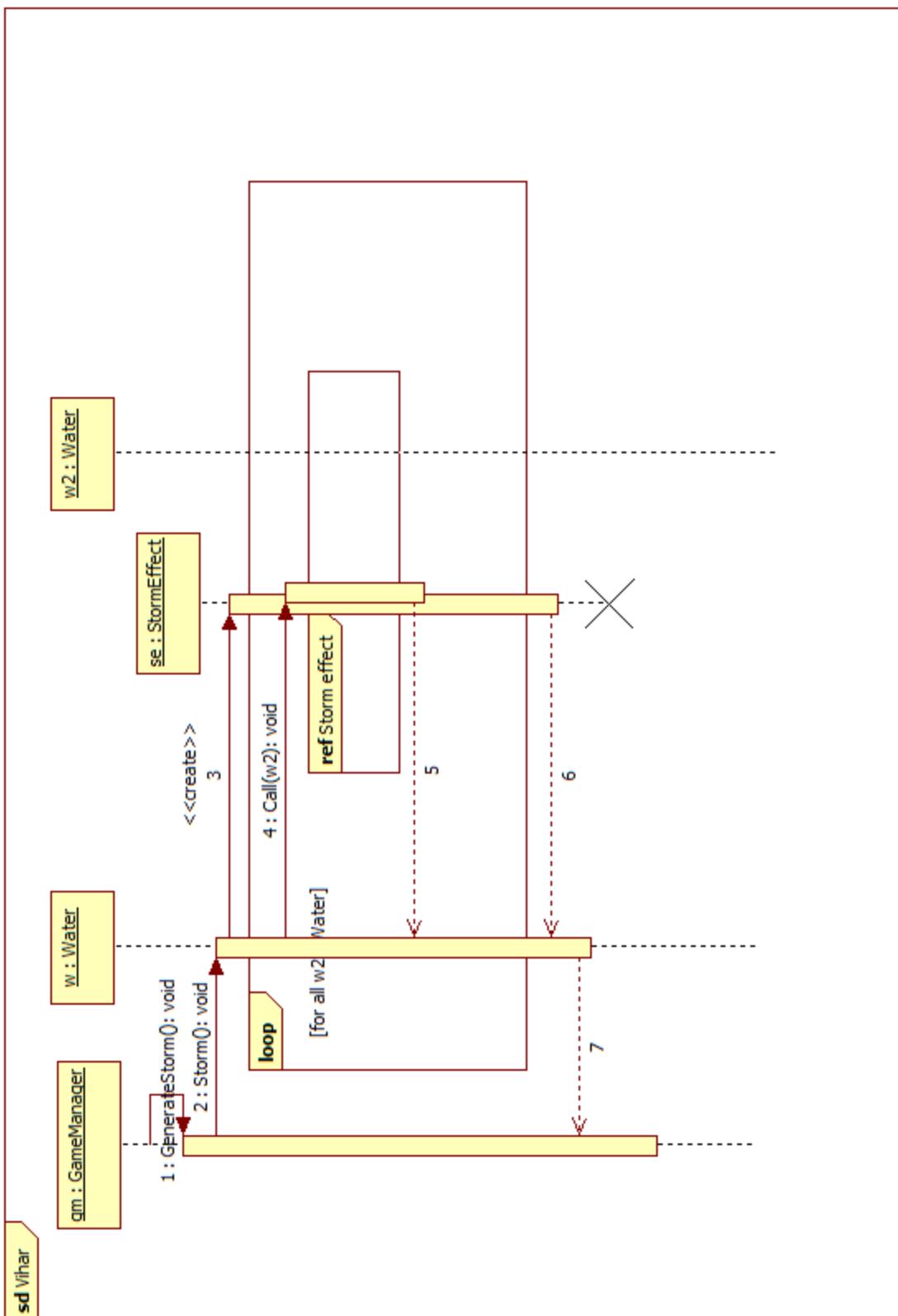
4.4.14

A kötél item használatával a szomszédos mezőről (arról amelyikre a játékos néz) kimenthetők az ott vízbe esett játékosok, akik mentés után a megmentőjük táblájára kerülnek. Használat után a kötél eltűnik.



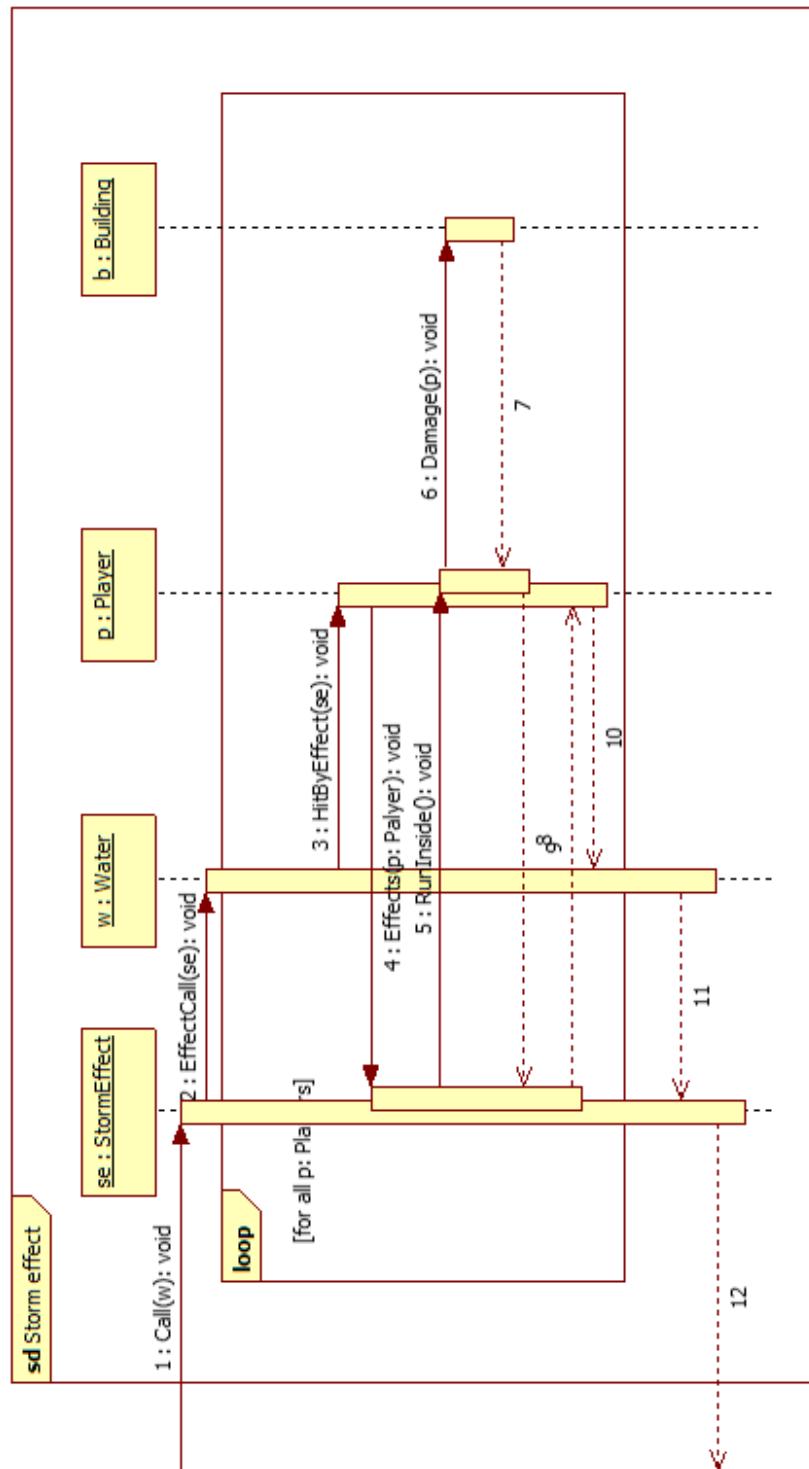
4.4.15

A GameManager a körök között bizonyos valószínűsséggel generál vihart, ha ez megtörténik, akkor a StormEffect-et használva érinti a mezőket és a rajtuk álló játékosokat.



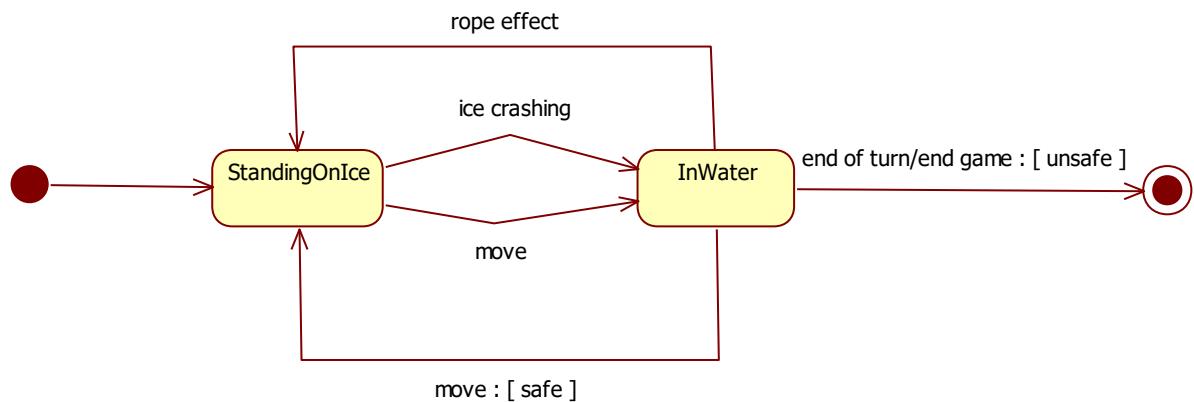
4.4.16

Ezt az effektust használja a vihar, amennyiben generálódik egy.

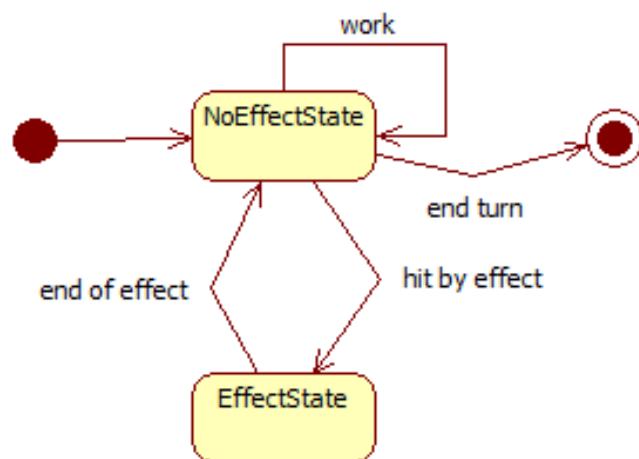


4.5 State-chartok

4.5.1 A kötél használata állapotokkal



4.5.2 Az effektusok működése állapotokkal



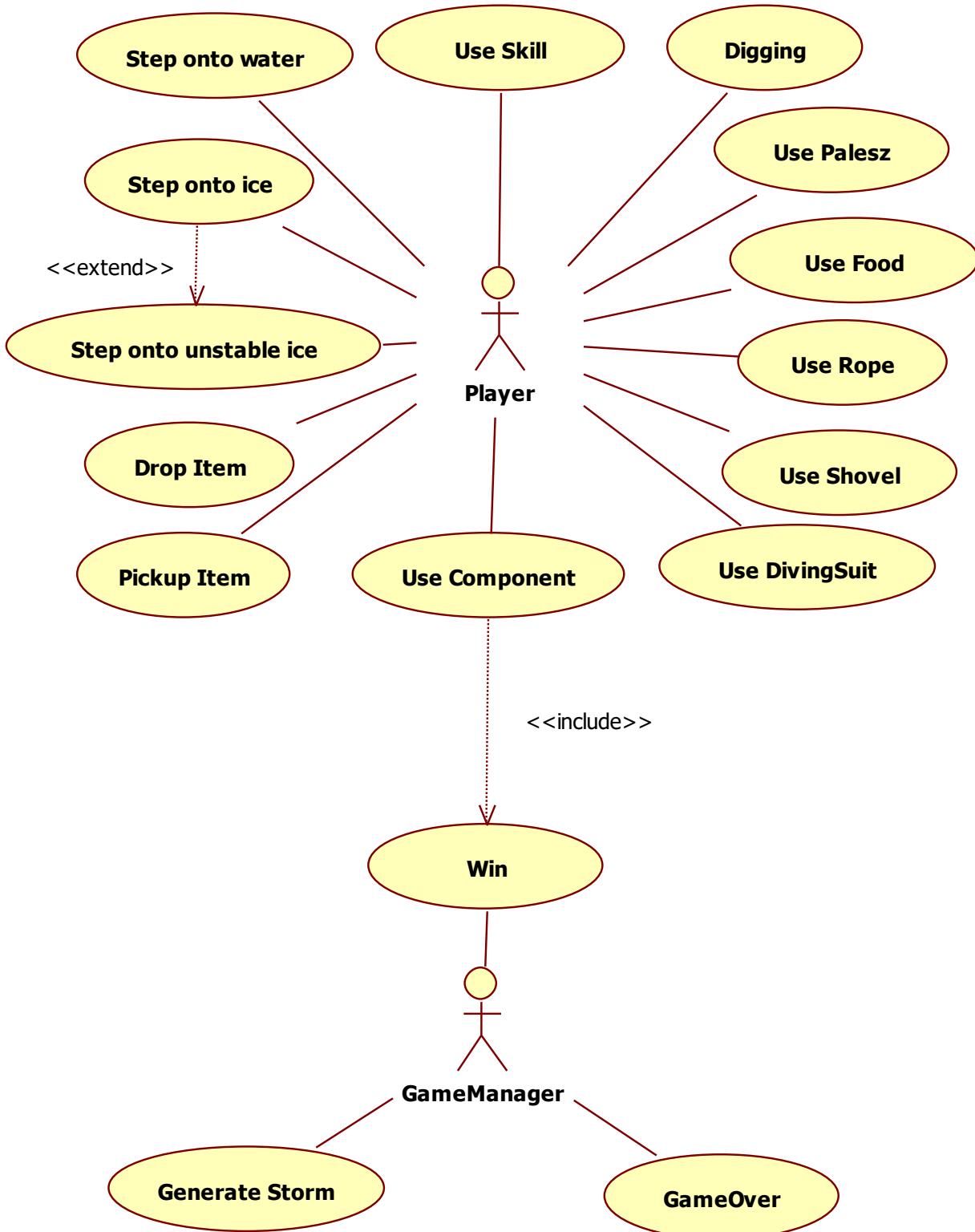
Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.03.05. 10:00	1 óra	Gaál Bodó Nagy Varga	Értekezlet. Felmerült problémák átbeszélése.
2020.03.05. 12:00	1 óra	Bodó Nagy	Funkciók átgondolása, megvalósíthatóság-nak tesztelése.
2020.03.06. 21:00	3,5 óra	Gaál Bodó Nagy Varga	Osztálydiagram javítása.
2020.03.07. 10:00	1 óra	Bodó Nagy	Funkcionalitás átbeszélése, osztálydia javítása.
2020.03.07. 16:00	4 óra	Bodó Gaál Nagy Rédai Varga	Osztálydia véglegesítése, szekvencia diagramok elkezdése.
2020.03.07. 21:00	2 óra	Bodó Gaál Nagy Rédai Varga	Előzőek folytatása
2020.03.08. 10:00	3 óra	Bodó Gaál	Osztályok leírása
2020.03.08. 14:00	2 óra	Bodó Gaál Nagy	Szekvenciák javítása
2020.03.08. 16:00	2 óra	Bodó Gaál Nagy Rédai Varga	Dokumentum javítása és state-chartok elkészítése
2020.03.08. 18:00	2 óra	Bodó Gaál Nagy Rédai Varga	Dokumentum véglegesítése, esetleges apróbb javítások

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	Step onto water
Rövid leírás	A játékos vízre lép.
Aktorok	Player
Forgatókönyv	1. A játékos karaktere vízbe kerül, ezáltal csak lépni képes.
Alternatív forgatókönyv	1.A Ha a játékos karaktere nem egy fóka és nem használta a vízbe esés előtt a búvárruha tárgyat (vagyis nincs biztonságban), akkor mozogni sem tud. 1.B Ha az adott karakter nincs biztonságban és az adott kör végéig senki sem menti ki, akkor a játék véget ér.

Use-case neve	Step onto ice
Rövid leírás	A játékos jégre lép.
Aktorok	Player
Forgatókönyv	1. A játékos olyan jégtáblára lép, amin bármennyien állhatnak, ezen a helyen munkát végezhet.

Use-case neve	Step onto unstable ice
Rövid leírás	A játékos instabil jégtáblára lép.
Aktorok	Player
Forgatókönyv	1. A játékos olyan jégtáblára lépett, amin csak egy bizonyos számú játékos állhat.
Alternatív forgatókönyv	1.A Ha a játékossal együtt már többen állnak az adott táblán, mint amennyit elbír, akkor az beszakad és a rajta álló játékosok vízbe kerülnek.

Use-case neve	Drop Item
Rövid leírás	A játékos eldob egy tárgyat.
Aktorok	Player
Forgatókönyv	1. A játékos eldob egy kiválasztott tárgyat, ami arra a jégtáblára kerül, amin a játékos éppen áll.

Use-case neve	Pickup Item
Rövid leírás	A játékos felvesz egy tárgyat.
Aktorok	Player
Forgatókönyv	1. A játékos felvesz egy tárgyat, arról a jégtábláról, amin éppen áll, ekkor az adott tárgy a saját tárolójába kerül. (Egyszerre maximum négy tárgy lehet egy játékosnál.)

Use-case neve	Digging
Rövid leírás	A játékos ás.
Aktorok	Player
Forgatókönyv	1. A játékos azon a jégtáblán ás, amin éppen áll.
Alternatív forgatókönyv	1.A Ha az adott jégtáblát hó fedi, akkor az ásással eltakarít egy réteg havat. 1.B Ha az adott tábla tiszta volt, akkor a táblába befagyott tárgyak közül ás ki egyet, ami a jégtábla felszínére kerül, így felvehető a játékos által.

Use-case neve	Use Skill
Rövid leírás	A játékos használja a saját képességét.
Aktorok	Player
Forgatókönyv	1. A játékos a saját karaktere speciális képességét használja.
Alternatív forgatókönyv	1.A Ha a karakter egy kutató, akkor le tudja kérdezni, hogy a jégtábla, amelyikre néz, az hány játékest bír el. 1.B Ha a karakter egy eszkimó, akkor az adott jégtáblára, amin áll, építhet egy iglut, ami egy esetleges hóvihar esetén megvédi a táblán álló játékosokat a vihar hatásaitól. 1.C Ha az adott karakter egy informatikus és van egy pálinkája, akkor annak használatával az adott mezőn álló minden játékos testhőjét visszaállítja az eredeti értékre. 1.D Ha az adott karakter egy fóka, akkor ezen képesség használata felesleges, mert hatása alapvetően is érvényesül (karakter sajátossága), ami az, hogy biztonságban tudhatja magát, vagyis, ha vízbe esik, akkor is ki tud belőle lépni.

Use-case neve	Use Food
Rövid leírás	Étel tárgy használata.
Aktorok	Player
Forgatókönyv	1. Az étel elfogyasztásával (használatával), a játékos testhője eggyel megnő, a tárgy használat után eltűnik.

Use-case neve	Use Shovel
Rövid leírás	Lapát tárgy használata.
Aktorok	Player
Forgatókönyv	1. A lapát tárgy használatával, az adott jégtábláról egyszerre két réteg havat is el lehet takarítani, az adott tárgy tartóssága eggyel csökken.
Alternatív forgatókönyv	1.A Ha használat után a tartósság nullára csökken, akkor a tárgy eltűnik.

Use-case neve	Use Palesz
Rövid leírás	Pálinka tárgy használata.
Aktorok	Player
Forgatókönyv	1. Ezt a tárgyat csak az informatikus karakter tudja használni. A tárgy használatával, az adott mezőn álló minden játékos testhőjét visszaállítja az eredeti értékre. Használat után a tárgy eltűnik.

Use-case neve	Use DivingSuit
Rövid leírás	Búvárruha tárgy használata.
Aktorok	Player
Forgatókönyv	1. A tárgy használatával, annak használója biztonságos állapotba kerül, vagyis, ha vízbe esik, akkor is ki tud belőle lépni. Használat után a tárgy eltűnik.

Use-case neve	Use Rope
Rövid leírás	Kötél tárgy használata.
Aktorok	Player
Forgatókönyv	1. A tárgy használatával, arról a mezőről, amire a játékos éppen néz, kimenthető minden játékos, amennyiben vízben vannak, ekkor tartóssága eggyel csökken.. Ebben az esetben a kimentett játékosok a megmentőjük mezőjére kerülne.
Alternatív forgatókönyv	1.A Ha használat után a tartósság nullára csökken, akkor a tárgy eltűnik.

Use-case neve	Use Component
Rövid leírás	Komponens tárgy használata.
Aktorok	Player, GameManager
Forgatókönyv	1. Egy komponens lehet patron, jelzőfény vagy pisztoly. (A program szempontjából mindegy.) Egy ilyen tárgy használatával a játékos megpróbálhat összerakni egy jelzőpisztolyt.
Alternatív forgatókönyv	1.A A játéknak sikerül összerakni egy jelzőpisztolyt, ha rendelkezik mindhárom komponenssel, ekkor a játék győzelemmel véget ér. 1.B Ha nincs meg az összerakáshoz minden alkatrész, akkor a játék folytatódik tovább.

Use-case neve	Win
Rövid leírás	A játék győzelemmel véget ér.
Aktorok	GameManager, Player
Forgatókönyv	1. A jelzőpisztoly sikeres elkészítése esetén a GameManager véget vet a játéknak, ebben az esetben a játékosok győznek.

Use-case neve	GameOver
Rövid leírás	A játék végének ellenőrzése.
Aktorok	GameManager
Forgatókönyv	1. A GameManager ellenőrzi, hogy a játék folytatható-e.
Alternatív forgatókönyv	<p>1.A Ha egy kör végén egy játékos vízben marad, de nincs biztonságban (nem fóka és nem használt búvárruhát) vagy van olyan játékos akinek a testhője nullára csökkent, akkor a játék véget ér és a játékosok veszítettek.</p> <p>1.B Ha a kör végén minden játékos testhője nagyobb mint nulla és nincs vízben olyan játékos, aki nincs biztonságban, akkor a játék folytatódik.</p>

Use-case neve	Generate Storm
Rövid leírás	Hóvihar generálása.
Aktorok	GameManager
Forgatókönyv	1. Egy adott mezőn, és annak szomszédain, hóvihar esetén egy újabb réteg hó rakódik rá a táblára (de maximum három lehet rajta), valamint a táblán álló játékosok bemenekülnek a táblán lévő épületbe.
Alternatív forgatókönyv	<p>1.A Ha a táblán lévő épület egy iglu, akkor a vihar semmilyen hatással nem lesz az adott táblán álló játékosokra.</p> <p>1.B Ha az adott táblán nincs iglu, akkor a vihar eggel csökkenti a rajta álló játékosok testhőjét.</p>

5.2 A szkeleton kezelői felületének terve, dialógusok

A program fejlesztésének ezen fázisában egy előre definiált tesztkészlet futtatására ad lehetőséget konzolon. Az indítás után megjelenik a főmenü, ahol az előre definiált szcenáriók közül választhatunk, ezt a választást a megfelelő menüpont előtt álló szám bevitelével tehetjük meg. (Nagyságrendileg annyi menüpont lesz, ahány use-case) Kiválasztás után megjelenik a szcenárió neve és alatta sorban a végrehajtott függvények a következő formátumban:

[tabulálás][objektum neve, amin hívunk].[függvény, amit hívunk][soremelés]

A tabulálás mértékét az határozza meg, hogy milyen mélyen vagyunk a hívási láncban. Kezdetben ez azt jelenti, hogy a tabulálás nulla, de ha a főprogram meghív egy függvényt, akkor ott már egyes lesz a tabulálás, majd ugyanígy a továbbiakban.

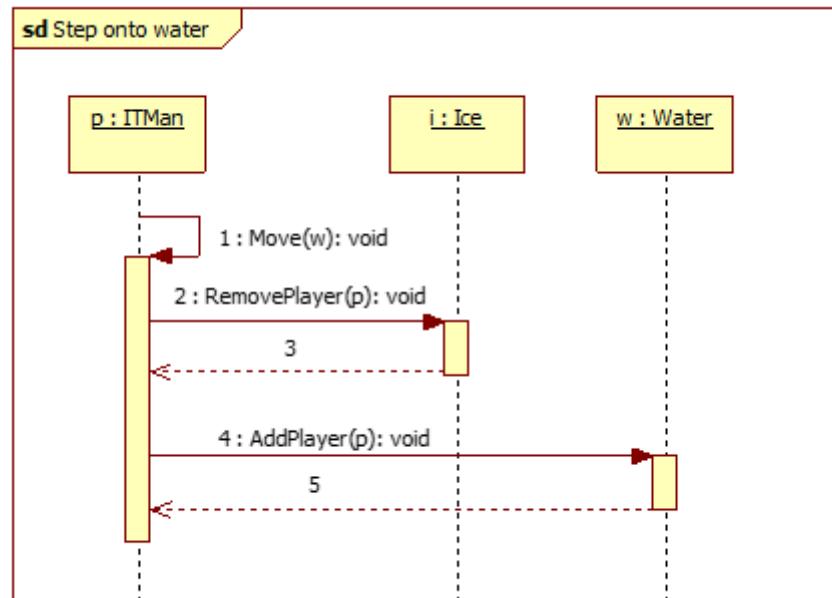
Néhány szcenárióban szükség van a felhasználó interakciójára. Ez az esetek túlnyomó részében azt jelenti, hogy az adott függvényhívás alatti sorban megjelenik egy eldöntendő kérdés, amire I/H billentyűk egyikével tud válaszolni. (Egy-egy kivételes esetben azonban szükség lehet számszerű válaszra is.) A felhasználó ezen egyszerű válaszok segítségével tudja alakítani az események történtét.

Megjegyzés: Ha a későbbi kódolás során kiderülne, hogy néhány szcenárióban olyan mélyre megy a függvényhívás, hogy a tabulálás csak rontana a szöveg olvashatóságán egy idő után, ott ezt külön kezeljük, vagyis nem tabulálunk.

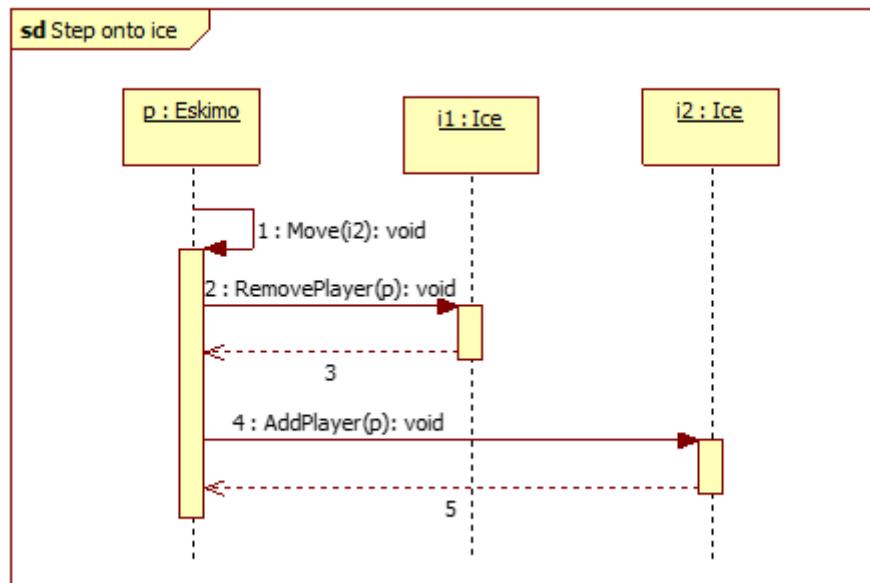
5.3 Szekvencia diagramok a belső működésre

A következő szekvenciadiagramokon egy-egy konkrét példát mutatunk a működésre, de a legtöbb esetben az adott tárgy vagy az adott karakter kicsérélhető bármely másikra, ahol ez nem teljesül, vagyis a lefutás csak egy adott karakterrel vagy tárggyal működik, azt külön jelezzük.

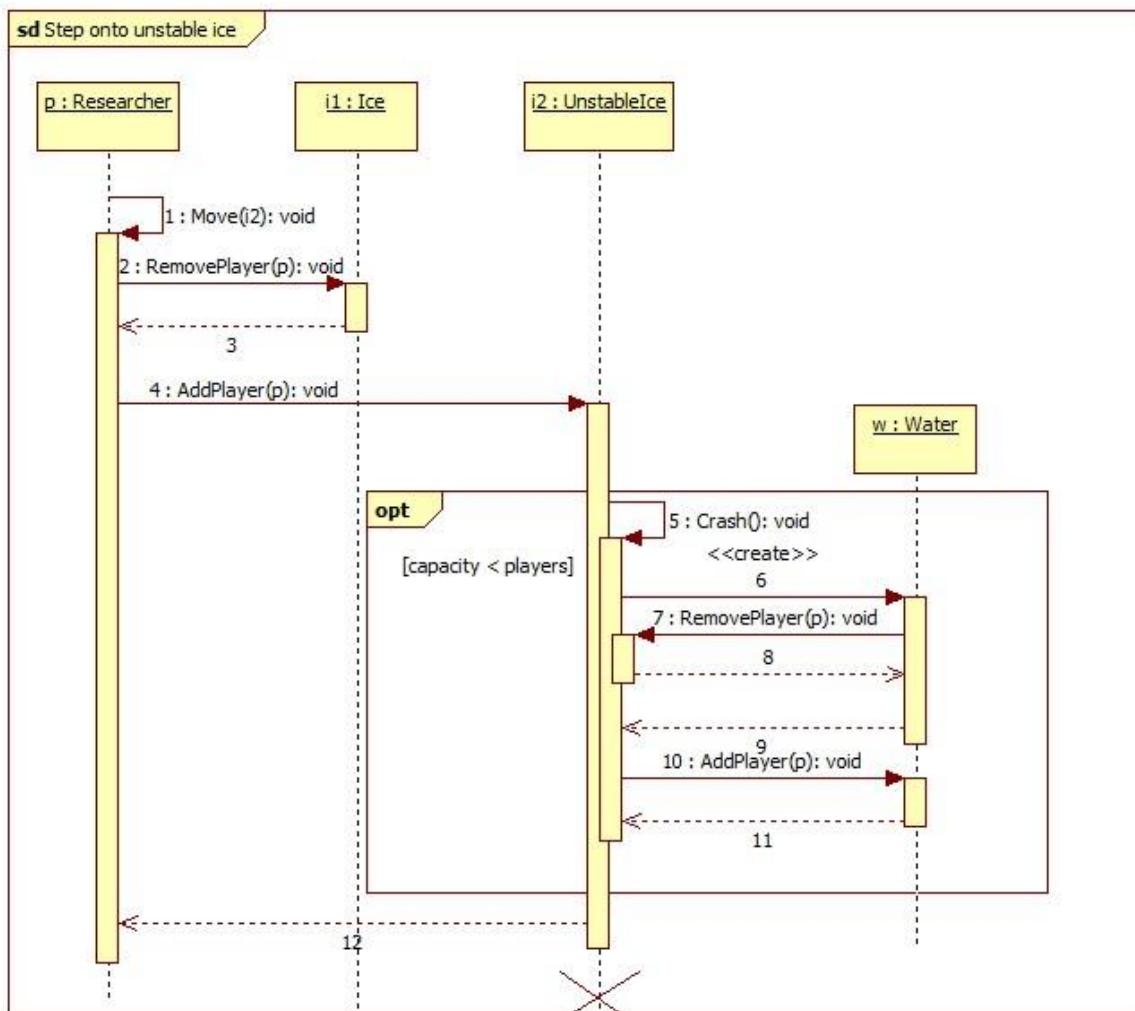
5.3.1 Egy Player átlépése egy Water-re



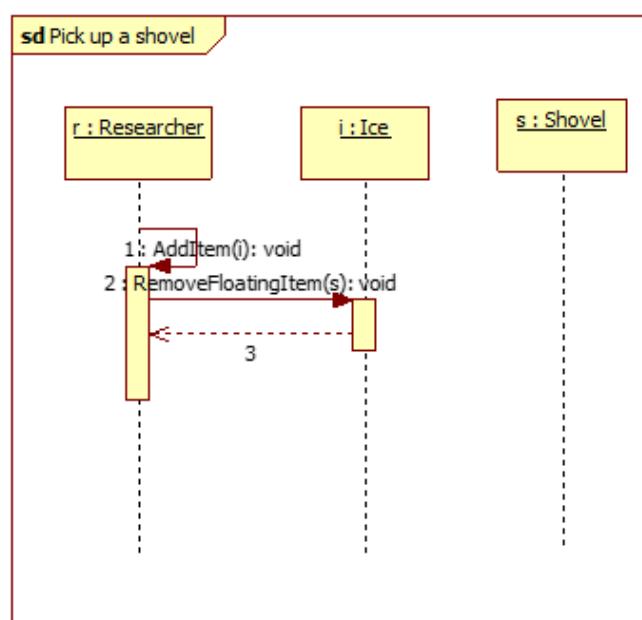
5.3.2 Egy Player átlépése egy Ice-ra



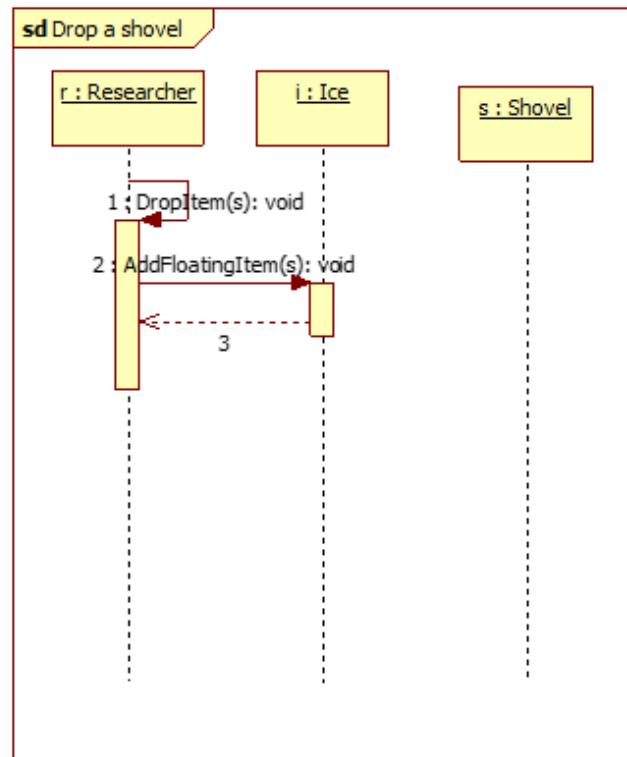
5.3.3 Egy Player átlépése UnstableIce-ra



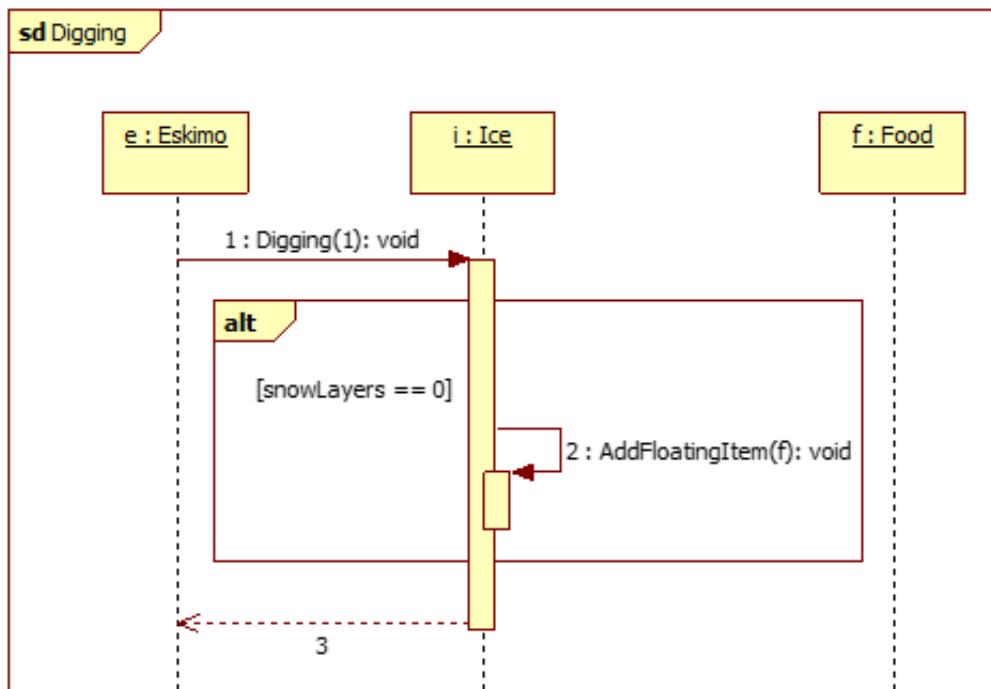
5.3.4 Egy Item felvétele egy Player által



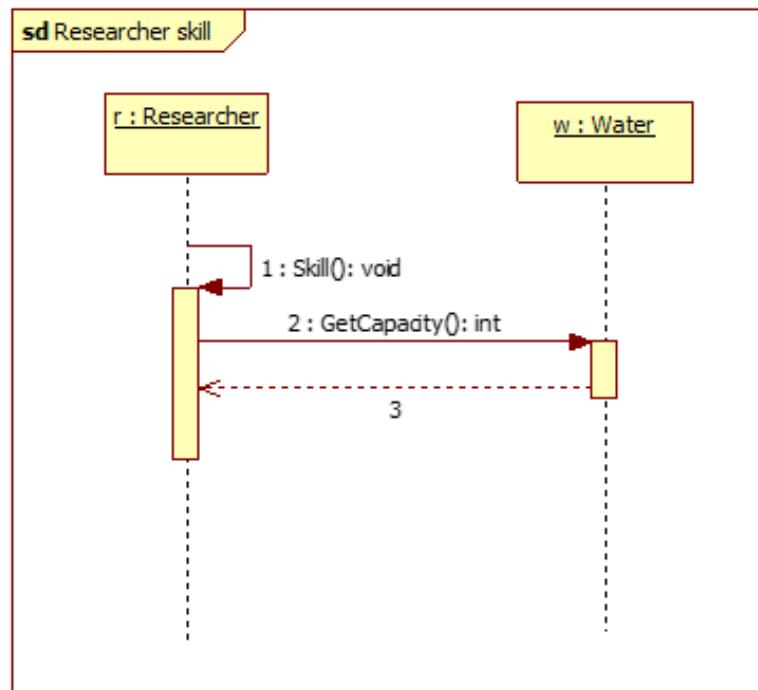
5.3.5 Egy Item eldobása egy Player által



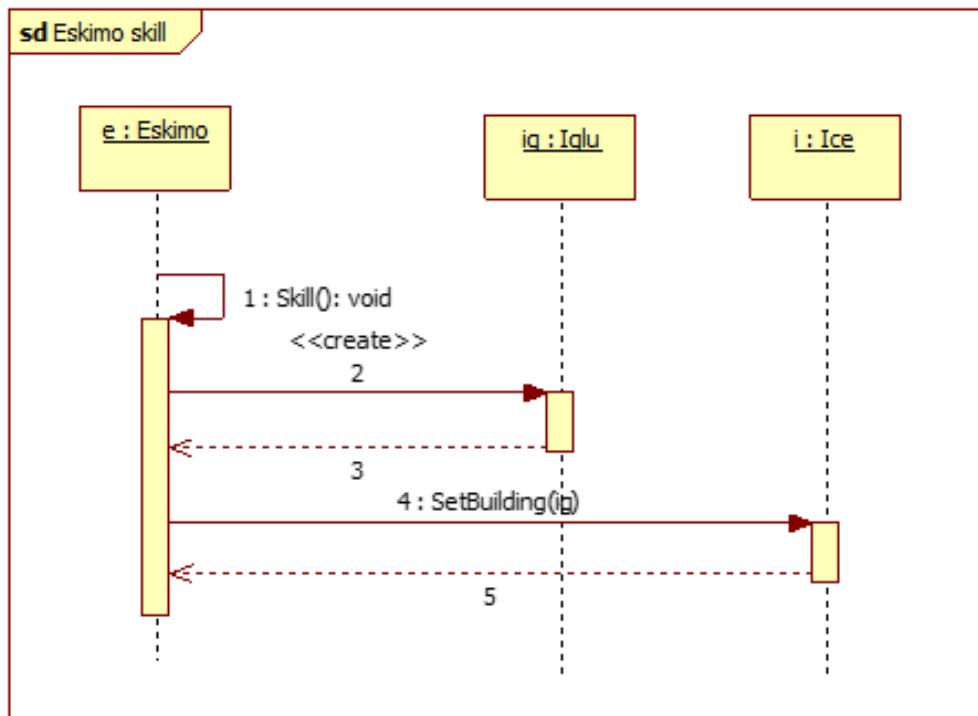
5.3.6 Ásás egy jégtáblán, egy Player által



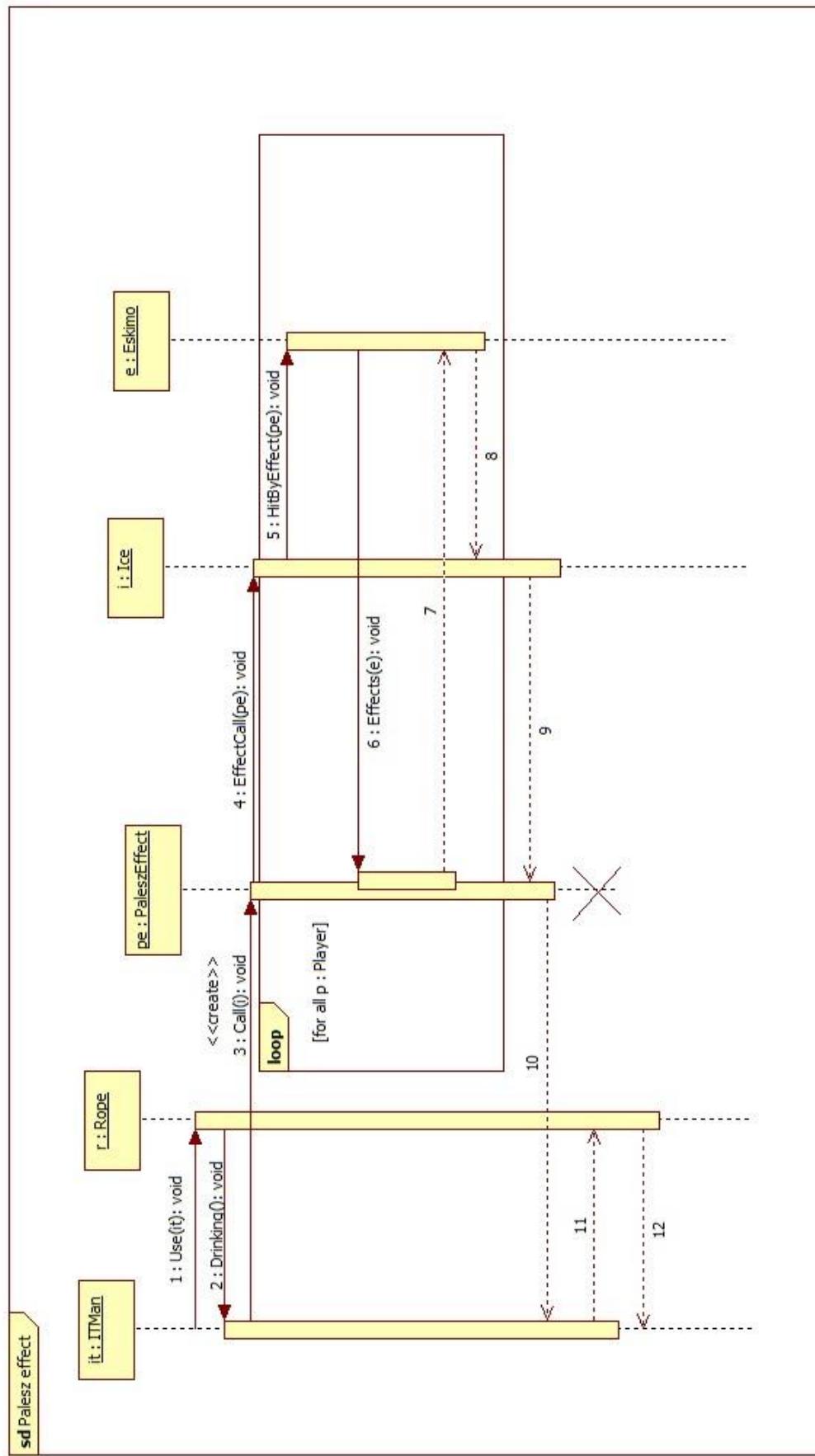
5.3.7 A Researcher saját képességének használata



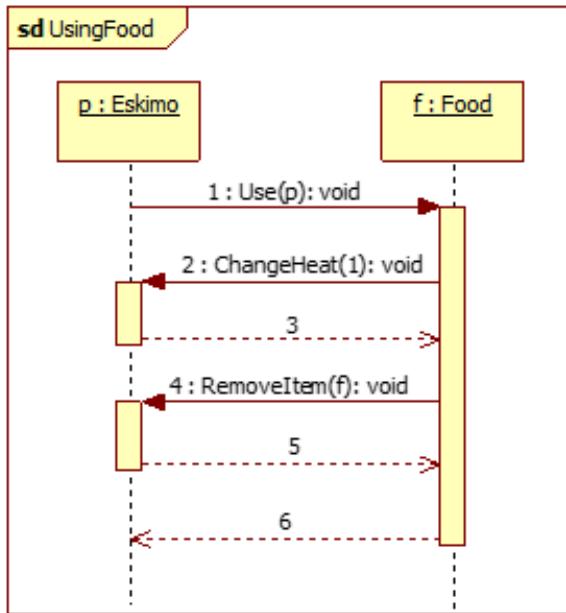
5.3.8 Az Eskimo saját képességének használata



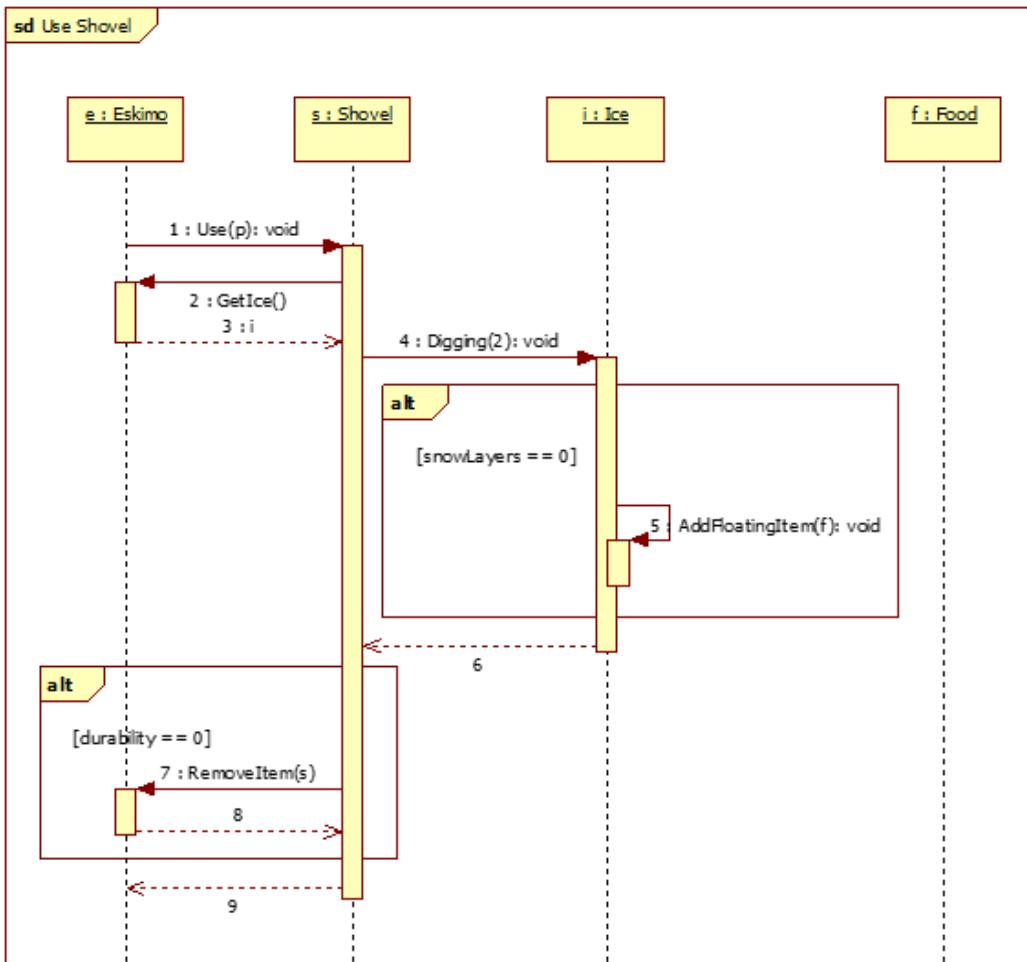
5.3.9 Az ITMan saját képességének használata

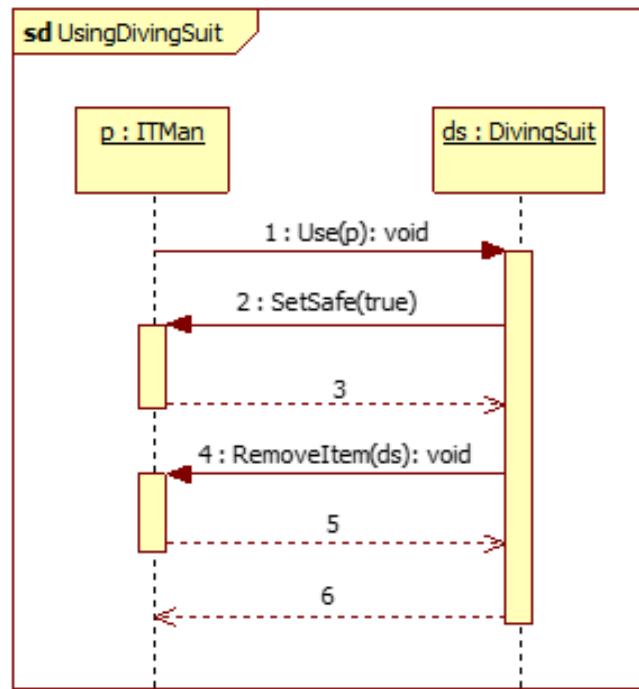


5.3.10 Food tárgy használata egy karakter által

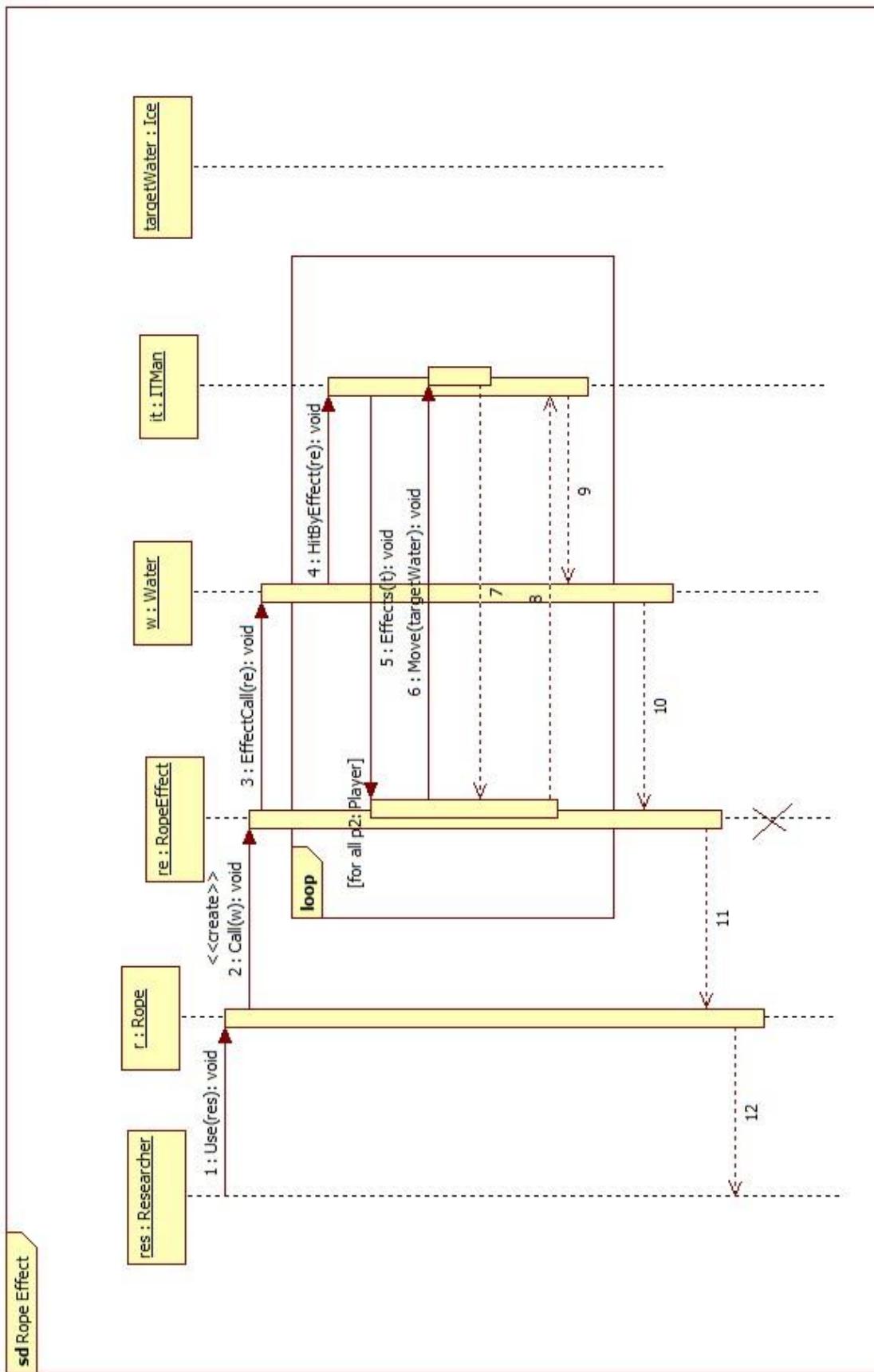


5.3.11 A Shovel item használata egy karakter által

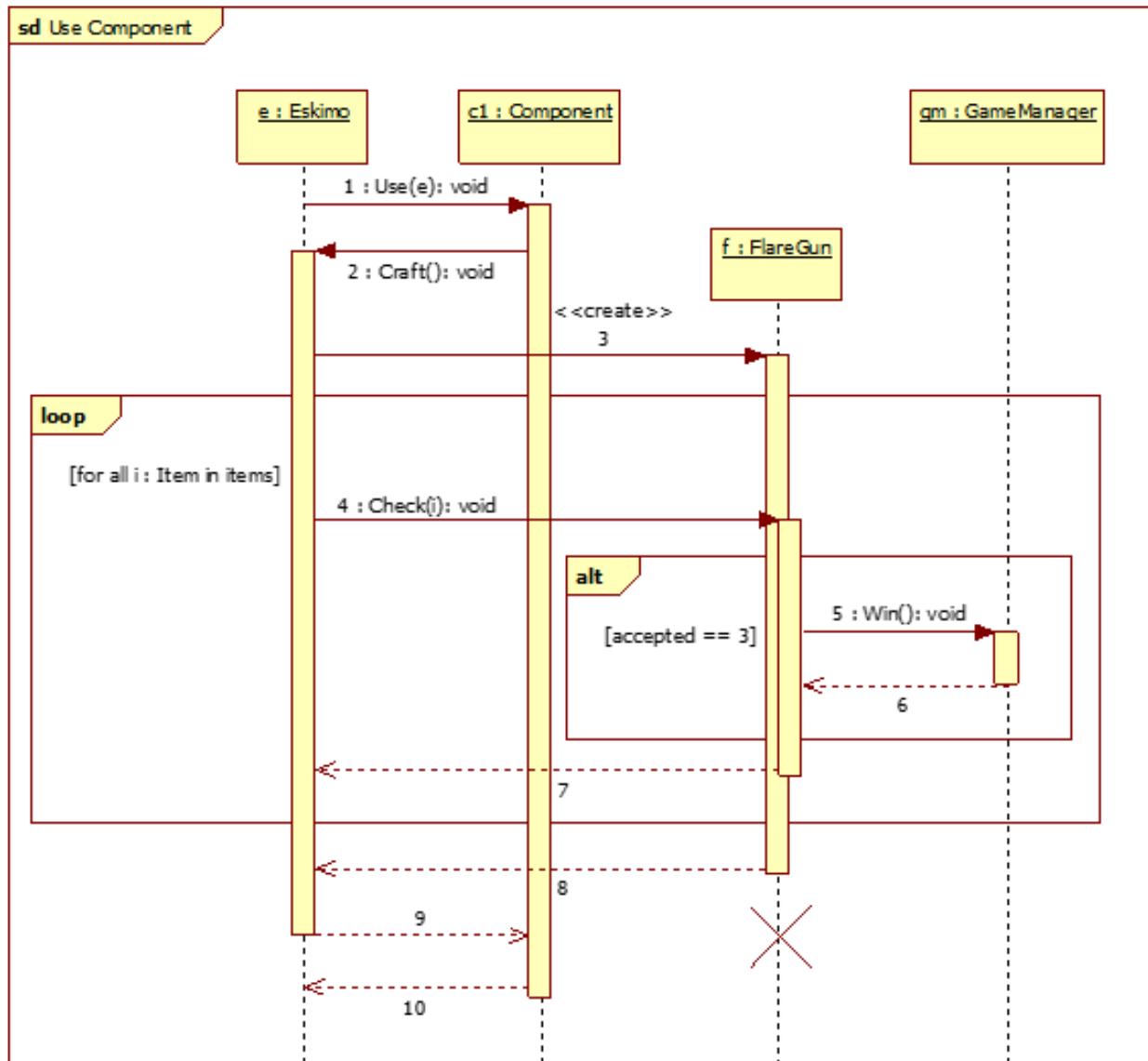


5.3.12 Egy DivingSuit item használata egy karakter által

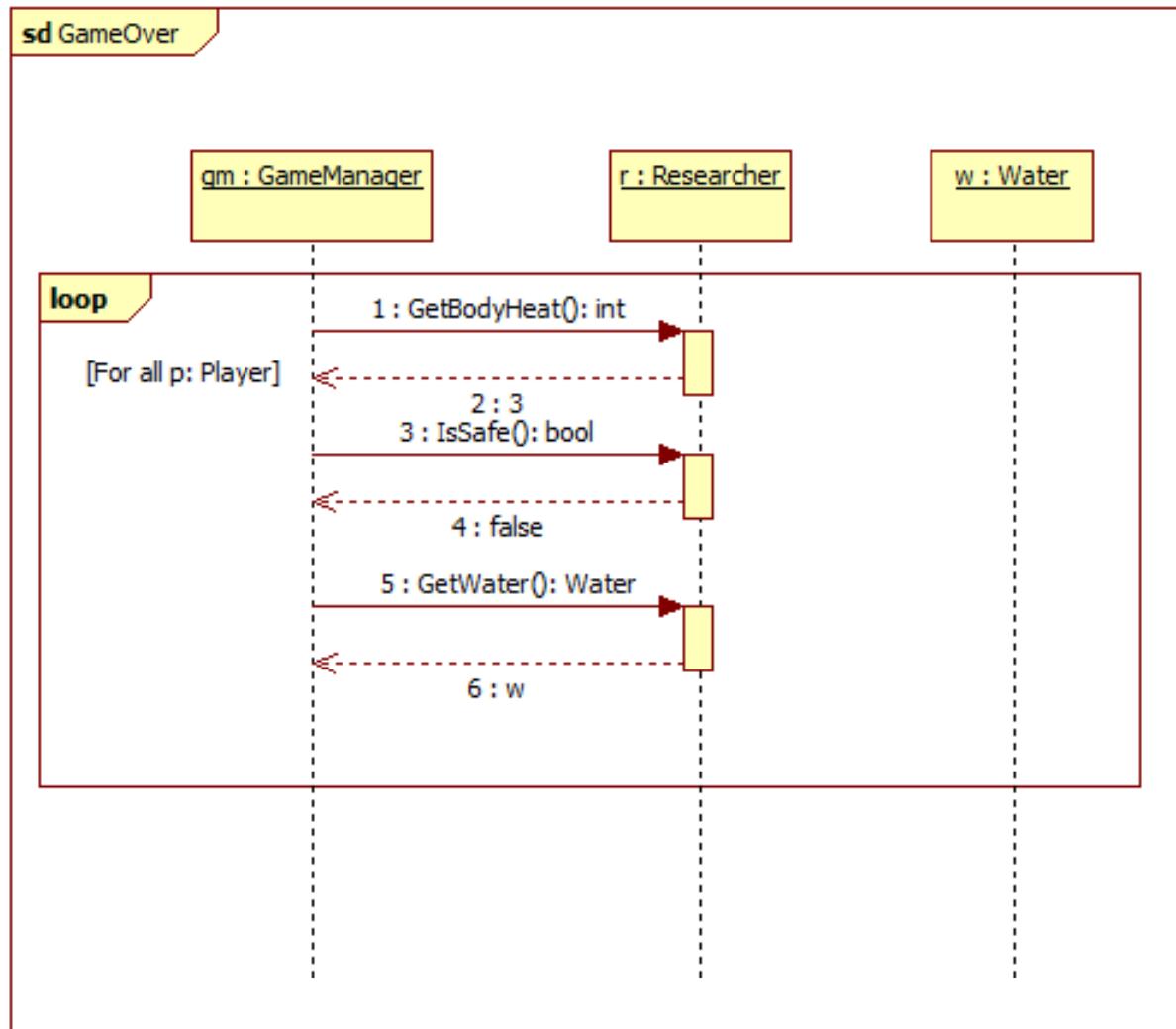
5.3.13 Egy Rope item használata egy karakter által



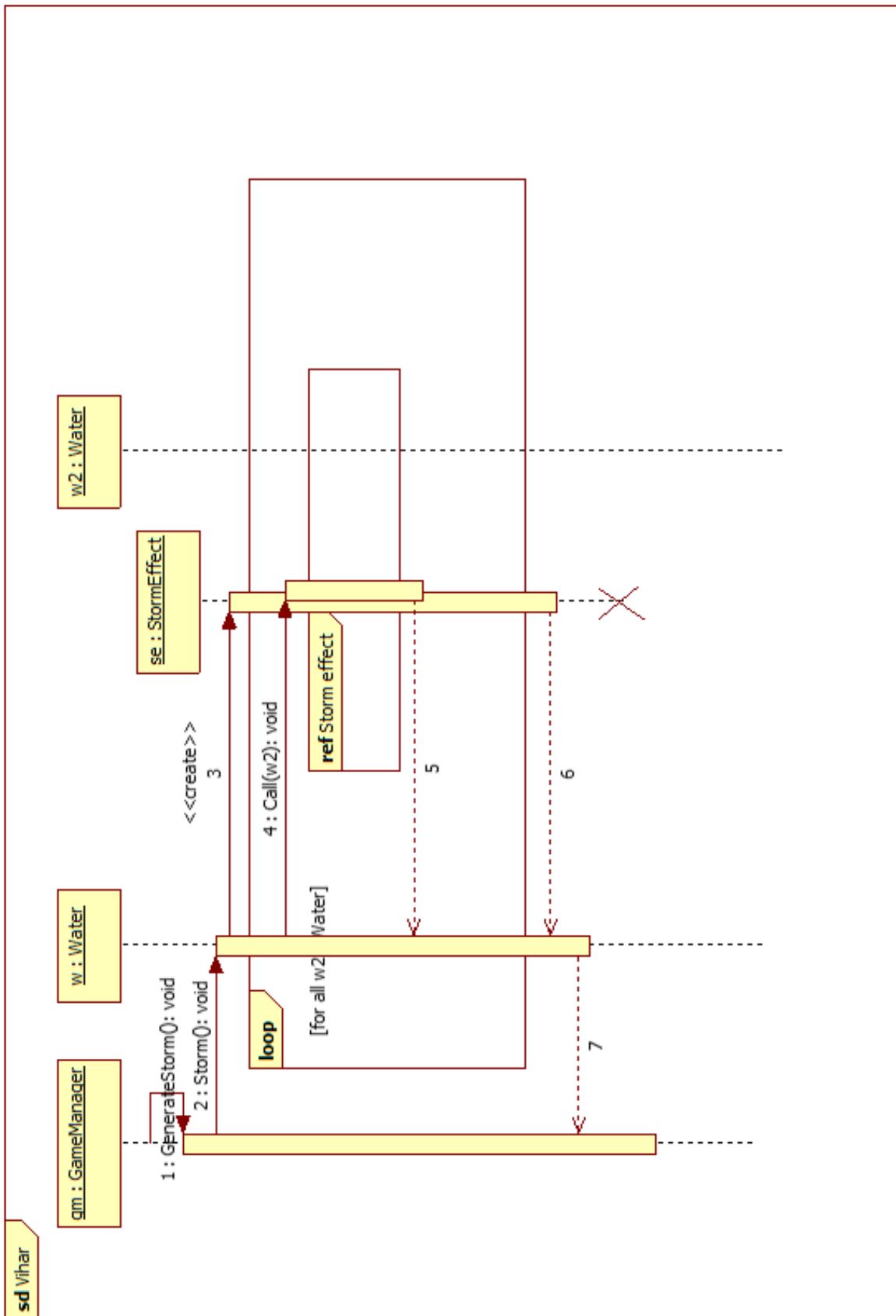
5.3.14 Egy Component item használata egy karakter által, illetve a legjobb esetben, a játék megnyerése a GameManager Win-je segítségével



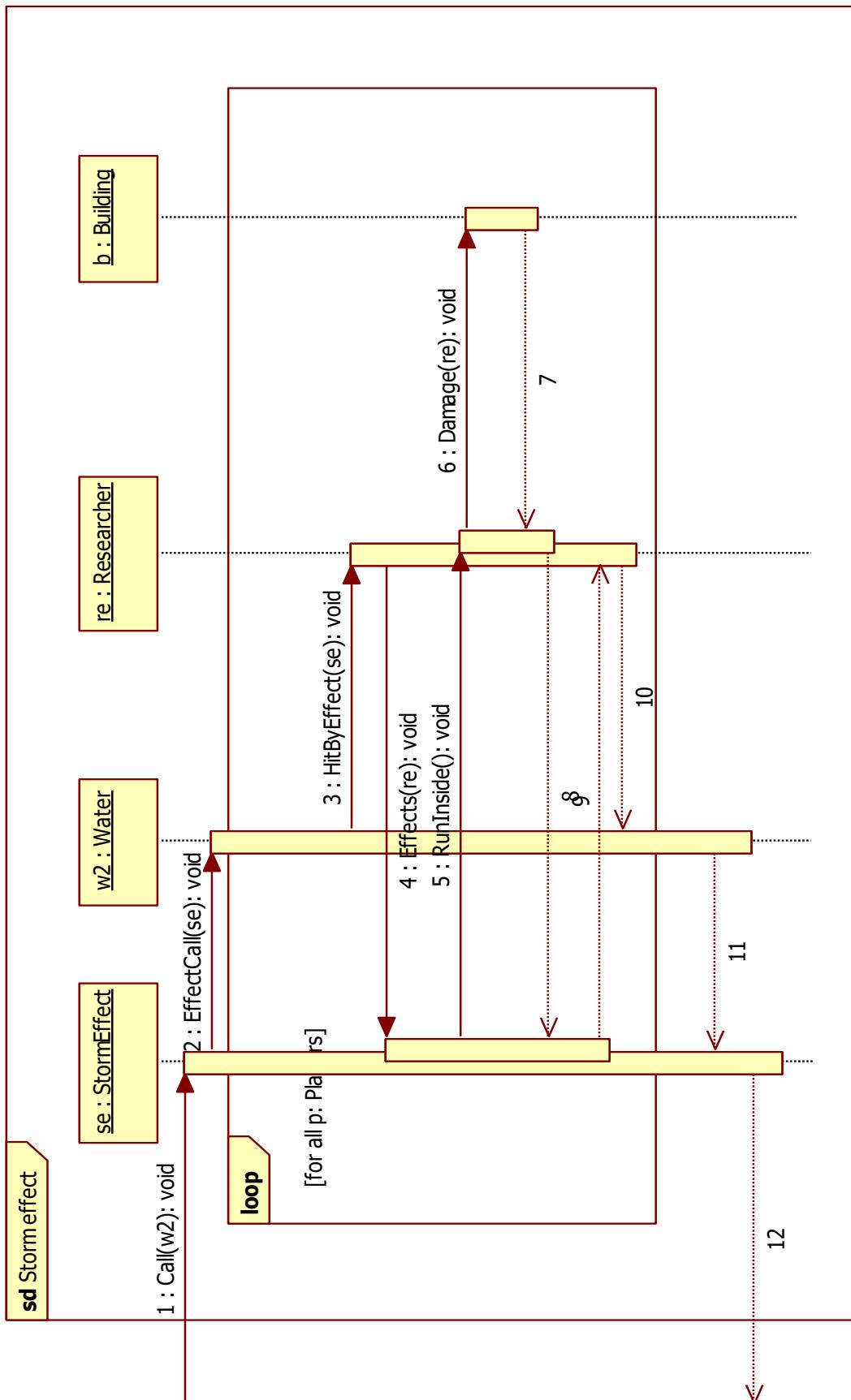
5.3.15 A GameManager ellenőrzi, hogy véget ért-e a játék



5.3.16 Hóvihar generálása a GameManager által

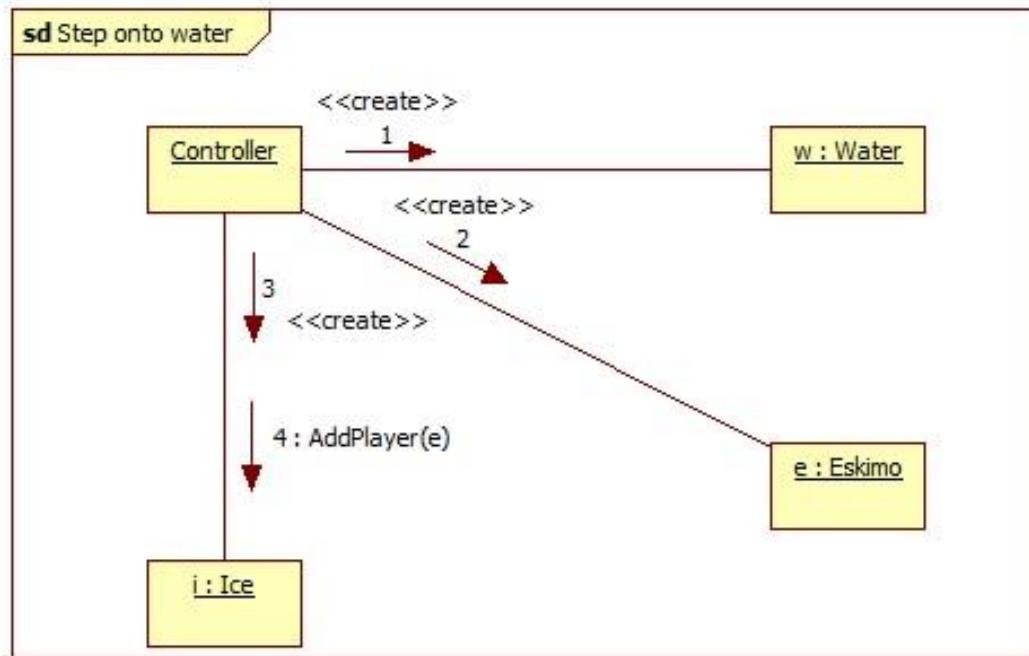


5.3.17 A hóvihar generálásához szükséges Storm Effect

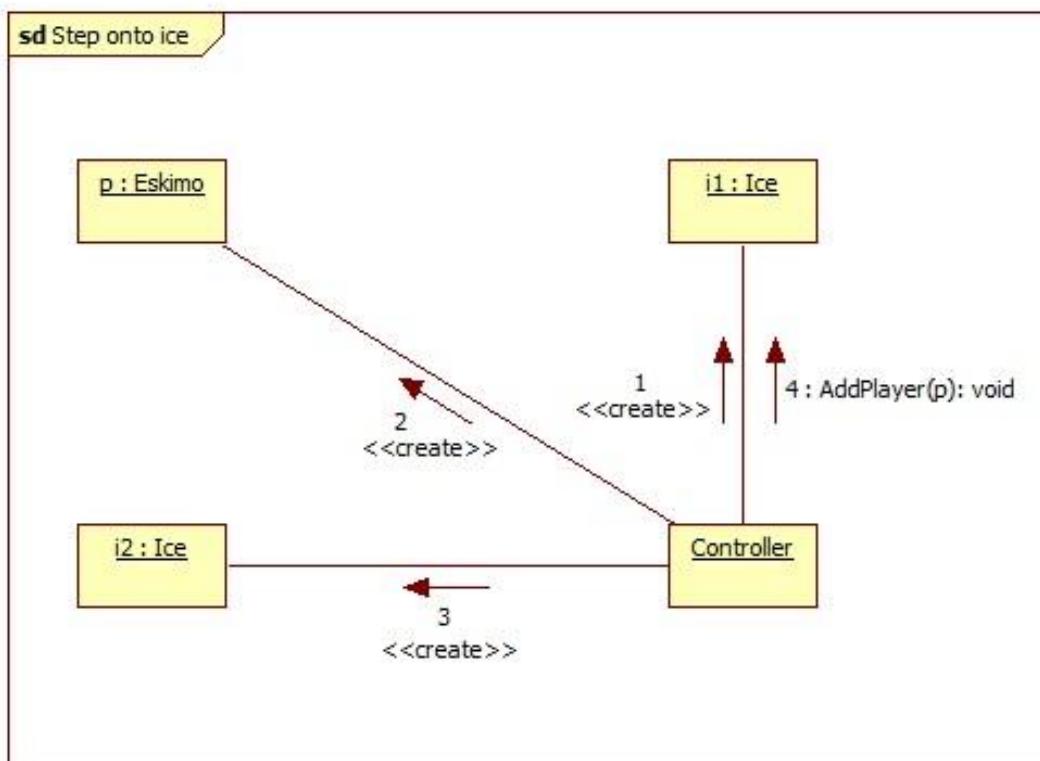


5.4 Kommunikációs diagramok

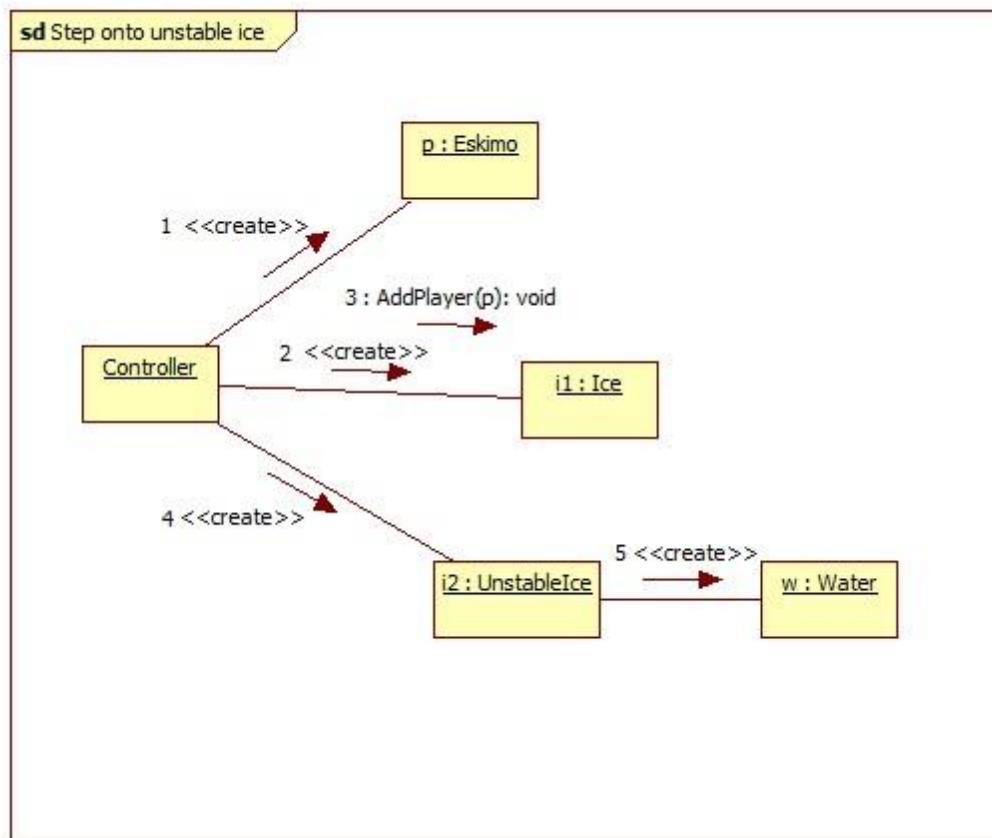
5.4.1 Egy Player átlépése egy Water-re



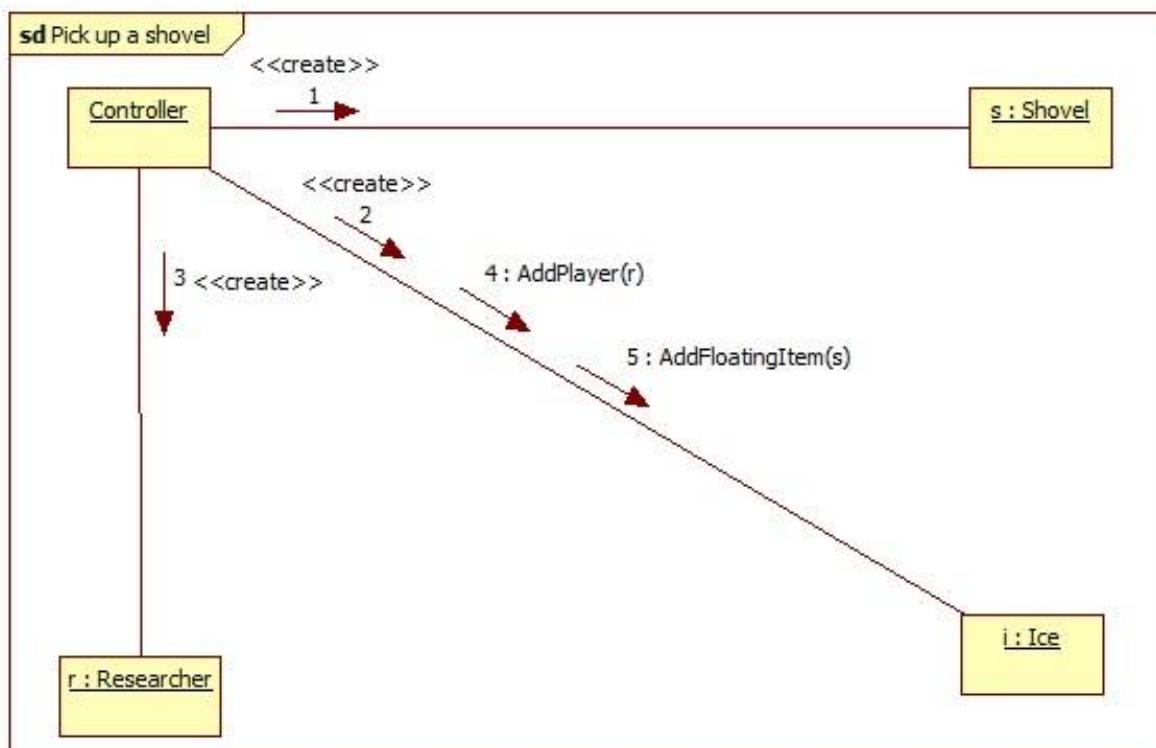
5.4.2 Egy Player átlépése egy Ice-ra



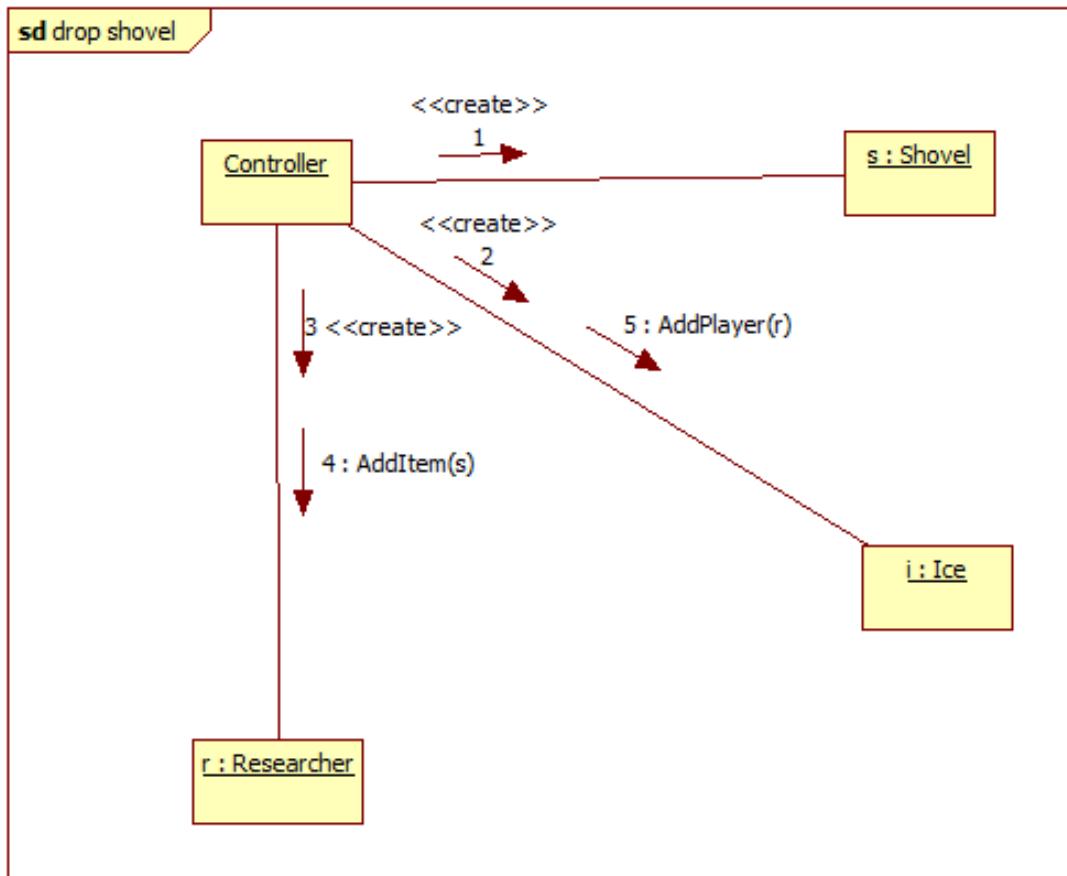
5.4.3 Egy Player átlépése UnstableIce-ra



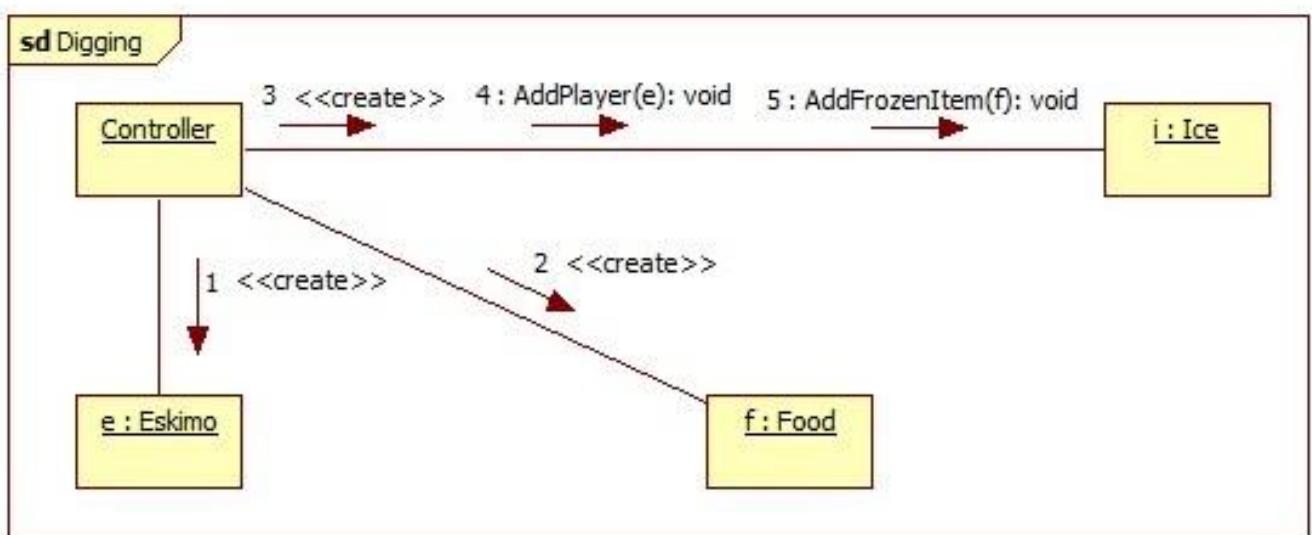
5.4.4 Egy Item felvétele egy Player által



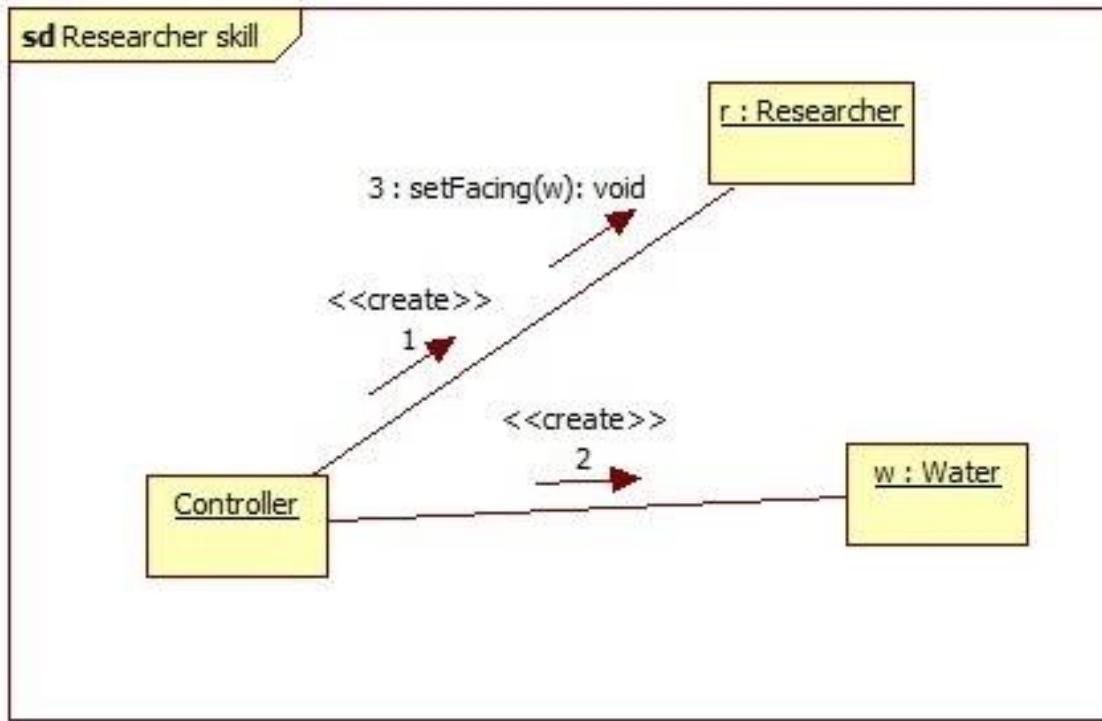
5.4.5 Egy Item eldobása egy Player által



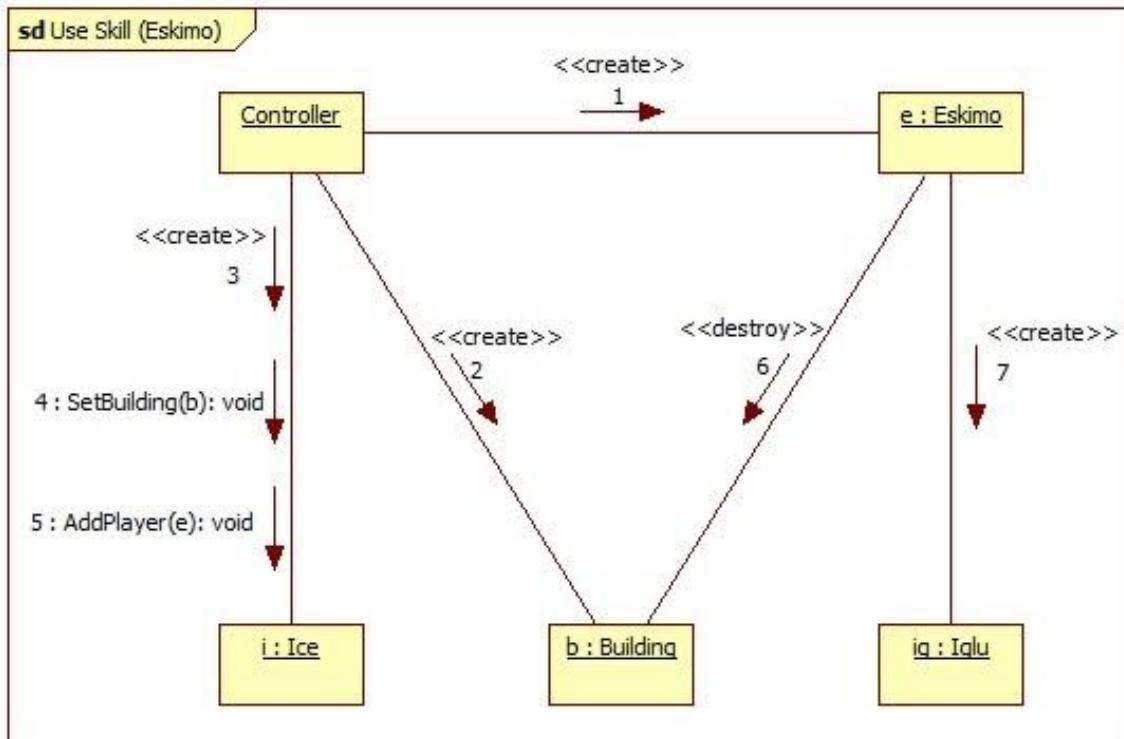
5.4.6 Ásás egy jégtáblán, egy Player által



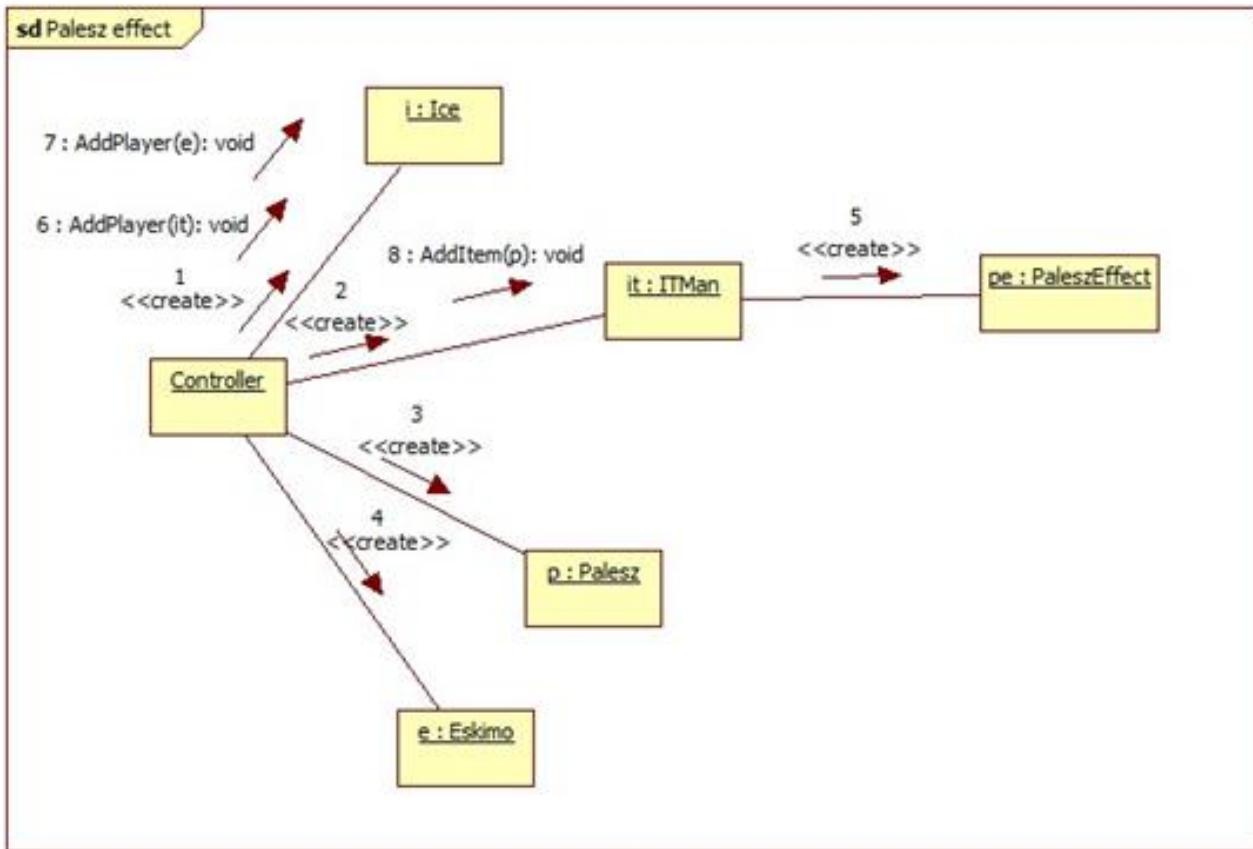
5.4.7 A Researcher saját képességének használata



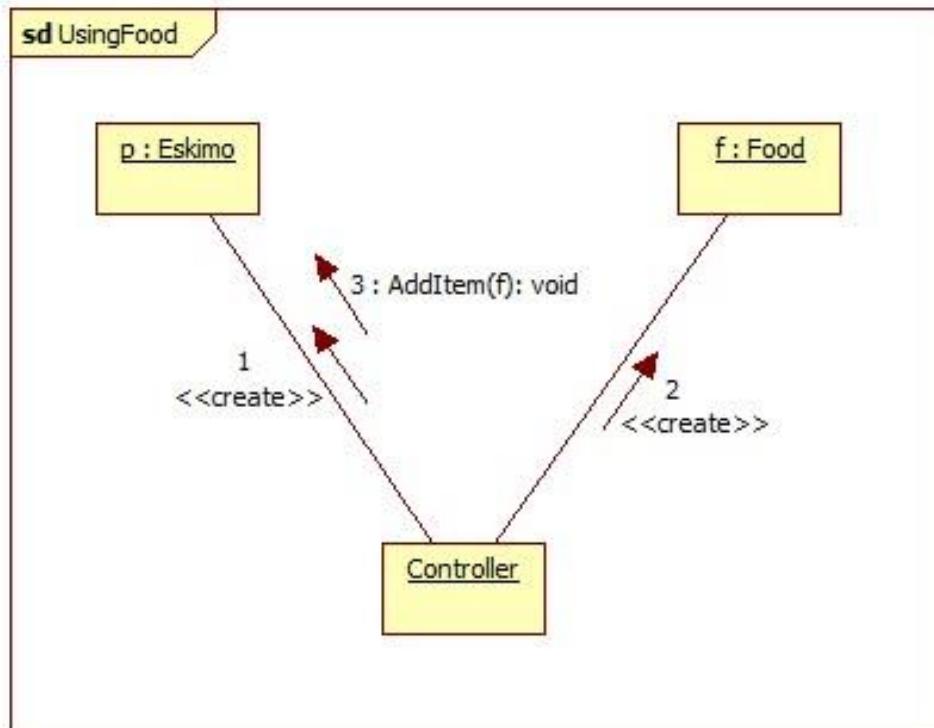
5.4.8 Az Eskimo saját képességének használata



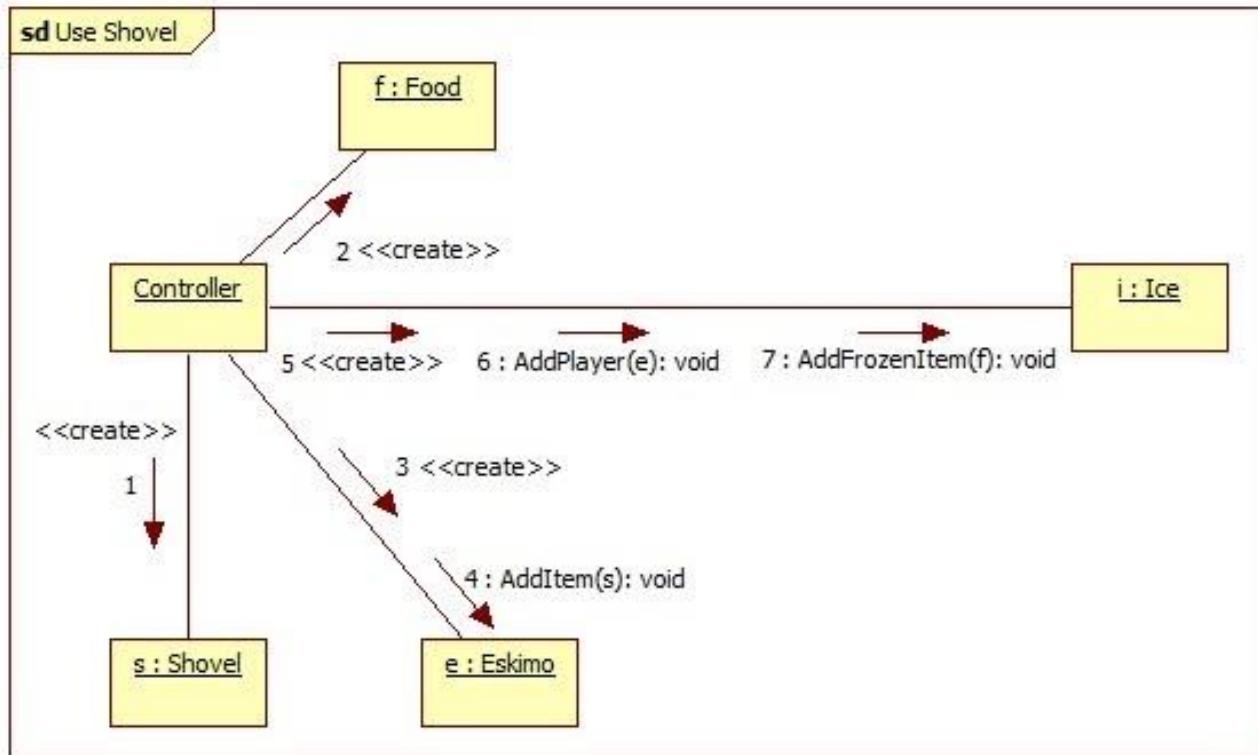
5.4.9 Az ITMan saját képességének használata



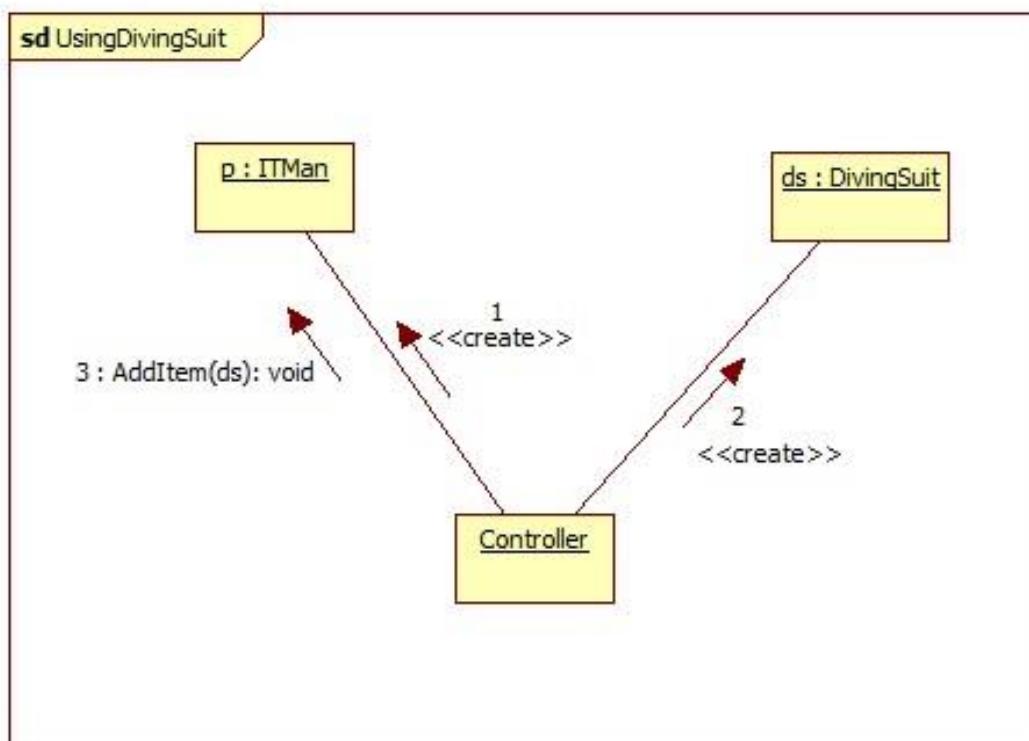
5.4.10 Food tárgy használata egy karakter által



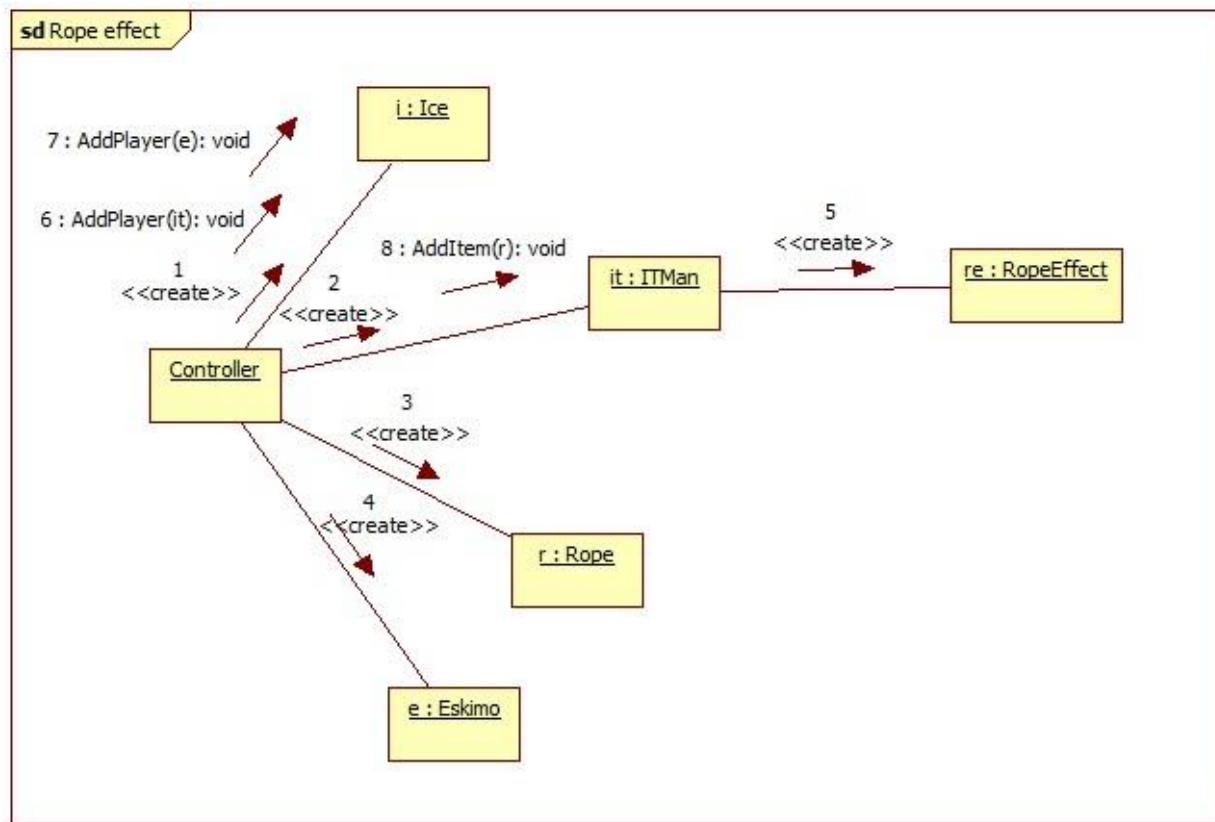
5.4.11 A Shovel item használata egy karakter által



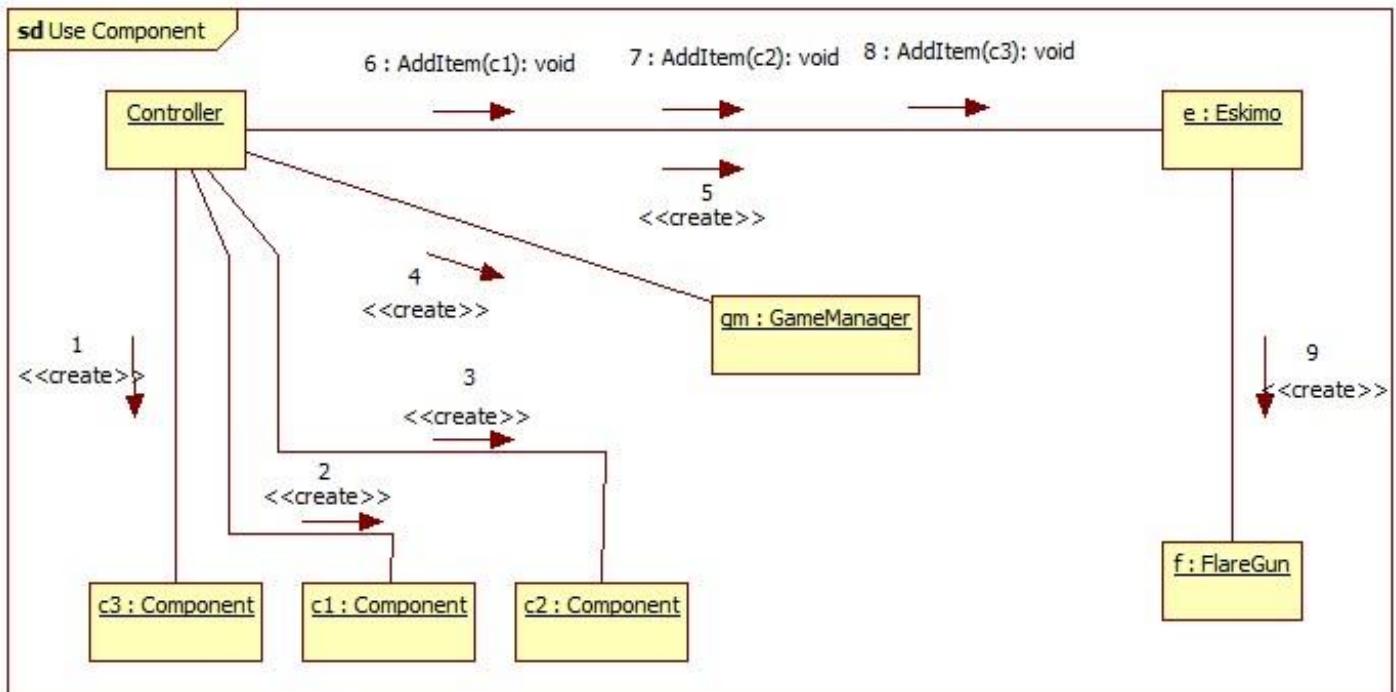
5.4.12 Egy DivingSuit item használata egy karakter által



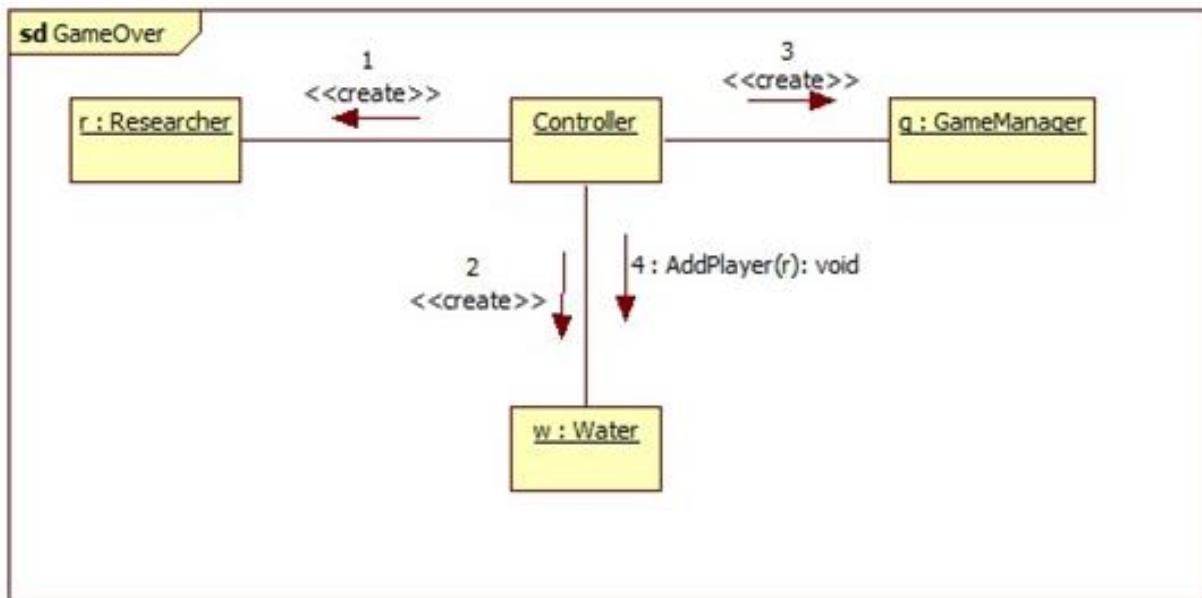
5.4.13 Egy Rope item használata egy karakter által



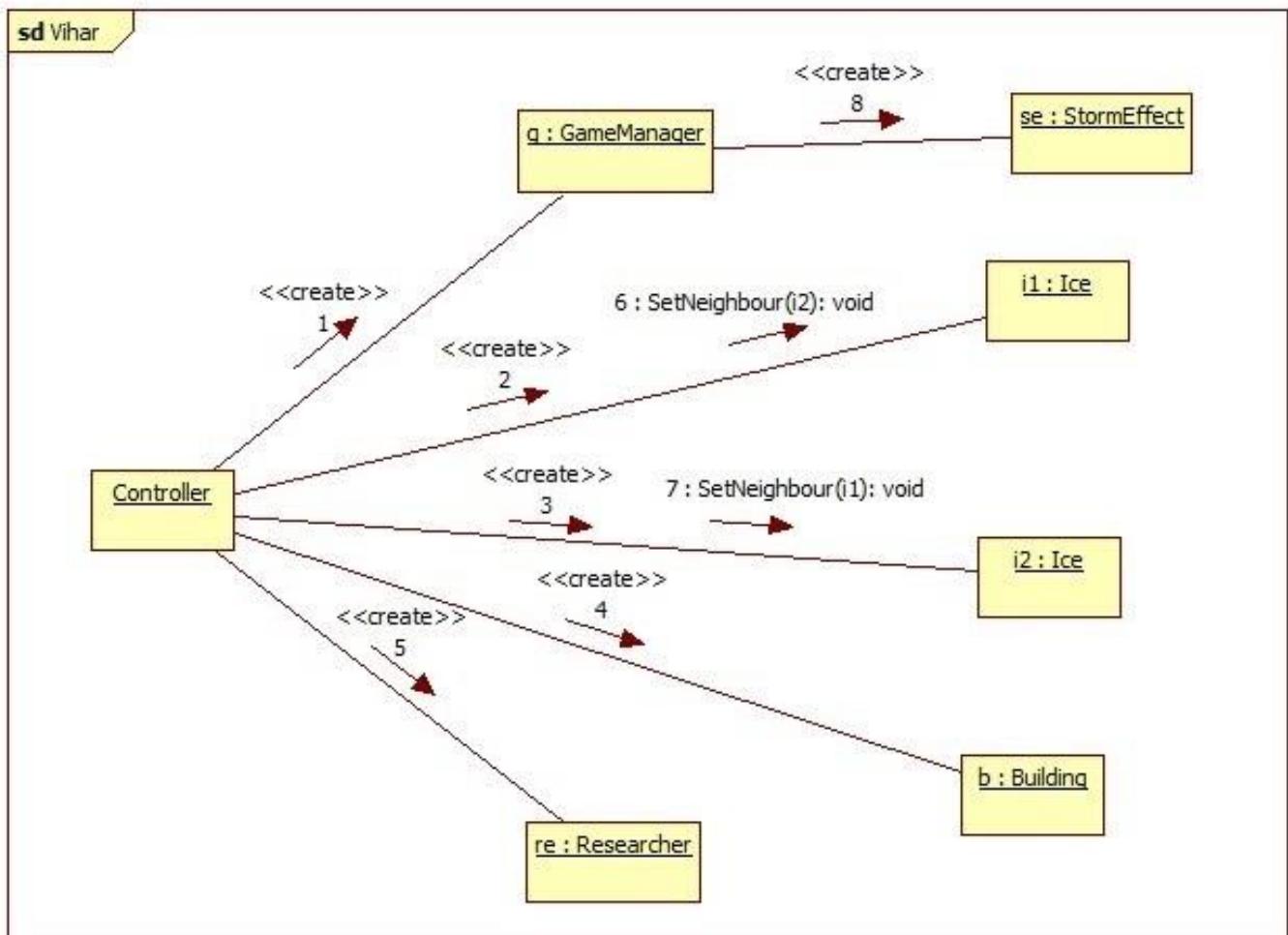
5.4.14 Egy Component item használata egy karakter által, illetve a legjobb esetben, a játék megnyerése a GameManager Win-je segítségével



5.4.15 A GameManager ellenőrzi, hogy véget ért-e a játék



5.4.16 Hóvihar generálása a GameManager által



5.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.03.11. 12:00	1 óra	Bodó Nagy Rédai	A Use-Case-k összegyűjtése.
2020.03.11. 16:00	1 óra	Bodó Gaál Nagy Rédai Varga	Gyors megbeszélés és a következő összejövetelek egyeztetése.
2020.03.12.12:00	5 óra	Bodó Gaál Nagy Rédai Varga	A Use-Case diagram véglegesítése.
2020.03.18. 14:00	4 óra	Bodó Gaál Nagy Rédai Varga	A szekvencia diagramok megtervezése.
2020.03.19. 15:00	3 óra	Bodó Gaál Nagy Rédai Varga	Szekvencia diagramok véglegesítése és a kommunikációs diagramok elkezdése.
2020.03.20. 11:00	6 óra	Bodó Gaál Nagy Rédai Varga	Kommunikációs diagramok folytatása, javítások a szekvencia diagramokon.
2020.03.20. 20:00	3 óra	Bodó Gaál Nagy Rédai Varga	Kommunikációs diagramok véglegesítése, dokumentum szerkesztése.
2020.03.21. 12:00	2 óra	Bodó Gaál Rédai Varga Nagy	Napló megírása. Dokumentáció átnézése.

6. Szkeleton beadás

A szkeleton készítése során selfedezett kisebb hiányosságokat, vagy a korábbi szekvenciákban található hibákat, jelenlegi szkeleton készítésekor javítva valósítottuk meg, illetve a problémás részeket visszamenőleg javítottuk. Ezen hibák alatt értünk paraméterezi hibákat vagy valahol külön get/set függvények bevezetését. Továbbá a tesztelés sikeressége miatt némely osztályokban szükség volt új függvények bevezetésére, illetve a teszteléshez segédosztályok készítésére, melyeket az adott helyeken a kommentekben egyértelműen jeleztünk. (pl.: Logger, Test - Ezeket a kész kód nem fogja tartalmazni.)

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Fájl neve	Méret (byte)	Keletkezés ideje	Tartalom
Alcoholic.java	339	2020.03.28 14:04	Alcoholic interfész
Building.java	583	2020.03.28 14:04	Building osztály
Component.java	772	2020.03.28 14:04	Component osztály
DivingSuit.java	715	2020.03.28 14:04	Diving Suit osztály
Effect.java	762	2020.03.28 14:04	Effect osztály
Eskimo.java	700	2020.03.28 14:04	Eskimo osztály
FlareGun.java	709	2020.03.29 20:46	FlareGun osztály
Food.java	594	2020.03.28 14:24	Food osztály
GameManager.java	2833	2020.03.29 21:30	gamemanager osztály
Ice.java	2333	2020.03.28 14:24	Ice osztály
Iglu.java	581	2020.03.28 14:24	Iglu osztály
Item.java	592	2020.03.28 14:24	Item osztály
ITMan.java	731	2020.03.29 21:03	ITMan osztály
Main.java	3036	2020.03.28 22:51	A main függvényt tartalmazó osztály
Palesz.java	770	2020.03.29 21:03	Palesz osztály
PaleszEffect.java	774	2020.03.29 21:08	PaleszEffect osztály
Player.java	7169	2020.03.29 21:47	Player osztály
Researcher.java	686	2020.03.28 14:04	Researcher osztály
Rope.java	686	2020.03.28 14:04	Rope osztály
RopeEffect.java	890	2020.03.28 14:04	RopeEffect osztály
Seal.java	572	2020.03.28 14:04	Seal osztály
Shovel.java	762	2020.03.28 14:04	Shovel osztály
StormEffect.java	639	2020.03.28 14:04	StormEffect osztály
UnstableIce.java	1653	2020.03.28 14:04	UnstableIce osztály
Water.java	5239	2020.03.28 14:04	Water osztály

Az alábbi fájlok mindegyike egy-egy teszeset megvalósítását tartalmazza, minden teszesethez egy osztály tartozik, illetve fellelhető köztük a teszthez készült segédosztályok is.

Fájl neve	Méret	Keletkezés ideje	Tartalom
Digging.java	582	2020.03.29. 21:03	A Player ás egy Ice-on
DropaShovel.java	557	2020.03.27. 21:03	Egy Player eldob egy Shovel-t egy Ice-on
EskimoSkill.java	531	2020.03.29 21:03	Az Eskimo használja a Skill-jét
GameOver.java	497	2020.03.29 21:03	A játék végének ellenőrzése
GenerateStorm.java	743	2020.03.29 21:03	A GameManager vihart generál
Logger.java	2090	2020.03.29 21:38	Logger osztály
PaleszEffectTest.java	457	2020.03.29 21:03	A Palesz Item működése
PickUpaShovel.java	571	2020.03.27 19:23	Egy Player felvesz egy Shovel-t egy Ice-ról
ResearcherSkill.java	435	2020.03.29. 21:47	A researcher használja a Skill-jét
RopeEffectTest.java	959	2020.03.27. 16:17	A Rope Item működése
StepOntoIce.java	508	2020.03.27. 16:17	Egy Player Ice-ra lép
StepOntoUnstableIce.java	567	2020.03.27. 16:37	Egy Player UnstableIce-ra lép
StepOntoWater.java	498	2020.03.27. 15:30	Egy Player Water-re lép
Test.java	247	2020.03.29. 21:47	Test osztály, a tesztek ősosztálya
UseComponent.java	556	2020.03.29. 21:03	Egy Player használja a Component-et
UsingDivingSuit.java	464	2020.03.29. 21:03	Egy Player használja a DivingSuit-ot
UsingFood.java	469	2020.03.27. 17:53	Egy Player használja a Food-ot
UsingShovel.java	636	2020.03.29. 21:03	Egy használja a Shovel-t

6.1.2 Fordítás

A fordításhoz el kell navigálni a terminálban a projekt mappájához:

```
set PATH=%PATH%;c:\Program Files\Java\jdk1.8.0_231\bin
```

```
set JAVA_HOME=c:\Program Files\Java\jdk1.8.0_231
```

```
javac aranyaszok/*.java aranyaszok/test/*.java
```

6.1.3 Futtatás

A fordítás után az alábbi parancssal futtatható a program:

java aranyaszok.Main

6.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Bodó István	LU93RR	20%
Gaál Gergely	GOXEHI	20%
Nagy Márton	DMMP94	20%
Rédai Attila	LYYOST	20%
Varga Levente	JIE1OY	20%

6.3 Napló

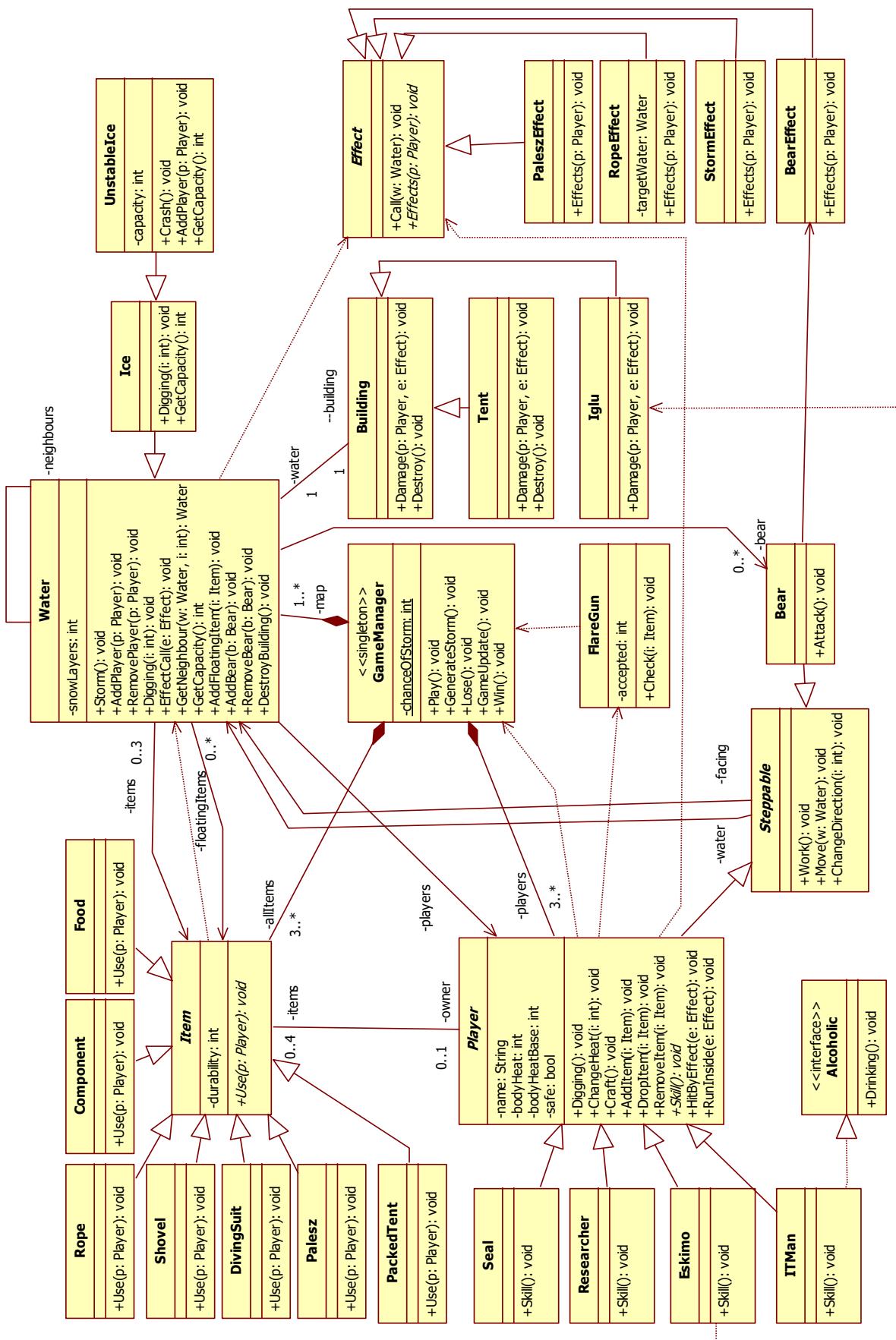
Kezdet	Időtartam	Résztvevők	Leírás
2020.03.26 10:00	2 óra	Bodó Gaál Nagy Rédai Varga	Osztályok és függvényeik fejlécének implementálása
2020.03.27 13:00	3 óra	Bodó Gaál Nagy Rédai Varga	Tesztosztályok elkészítése és a tesztelés menetének egyeztetése
2020.03.27 18:00	4 óra	Bodó Gaál Nagy Rédai Varga	Tesztesetek implementálása
2020.03.28 12:00	2 óra	Bodó Gaál Nagy Rédai Varga	A fejlécek ellátása javadoc kommentekkel
2020.03.28 14:00	3 óra	Bodó Gaál Nagy Rédai Varga	Programban hibák javítása és java dokumentáció elkészítése
2020.03.29 20:00	3 óra	Bodó Gaál Nagy Rédai Varga	Hibák javítása, Szkeleton dokumentáció elkészítése

7. Prototípus koncepciója

7.0 Változás hatása a modellre

Az egyik módosított követelmény, hogy legyen olyan ásó tárgyunk, ami három használat után “eltörök”. Mivel a mi modellünkben eredetileg is úgy működött az ásó tárgy, hogy volt strapabírása, vagyis bizonyos számú használat után eltűnt, ezért ez a követelmény, majd csak az adott konstruktőrban jelent változást. Ezek alapján ez az új követelmény ilyen szempontból nem okozott változást a modellben.

7.0.1 Módosult osztálydiagram



Új vagy megváltozó metódusok

A *-al jelölt metódusok és adattagok leírása megegyezik a korábbiakban specifikáltakkal, működésükben változás nem történt. (néhány esetben új osztályba helyeztük át)

Új osztályok:

Bear:

- **Leírás:** A jegesmedvét modellezzi. Gyakorlatilag egy mozogni képes karakter, akit a GameManager irányít és képességét a függvényei valósítják meg. Őse a Steppable.
- **Metódusok:**
 - Attack(): void - Azon a mezőn ahol éppen áll, ott használja képességét, vagyis, ha azon a mezőn áll valaki, akkor vége a játéknak.

BearEffect:

- **Leírás:** A medve támadására létrehozott új osztály. Ennek az osztálynak a felhasználásával valósítjuk meg a támadást.
- **Metódusok:**
 - Effects(p:Player) : void - A játékosnak csökentjük a testhőjét attól függően, hogy melyik épületbe menekül be.

Steppable:

- **Leírás:** Ez egy olyan absztrakt osztály, ami egy karakter mozgását és ahhoz szükséges adattagokat valósítja meg, ebből örököl a Player és a Bear is.
- **Adattagok:**
 - facing: Water *
 - water: Water *
- **Metódusok:**
 - Move(w: Water): void *
 - Work(): void *
 - ChangeDirection(i: int):void - Annak az iránynak a megváltoztatása, amerre az adott karakter néz. A paraméterként kapott értékkel adhatjuk meg, hogy mennyivel forduljon el a karakter.

Tent:

- **Leírás:** A Water táblán elhelyezkedő “sátor”. Őse a Building osztály.
- **Metódusok:**
 - Damage(p:Player,e:Effect):void - Az épületbe berohanó játékos testhőjét csökkenti attól függően, hogy mi az effect.
 - Destroy():void - A sárat “semmisíti meg”, kicseréli egy Building-re.

PackedTent:

- **Leírás:** Egy új tárgy, amit a játékosok ki tudnak ásni.
- **Adattagok:**
 - durability: int *
- **Metódusok:**
 - Use(p: Player): void - A tárgy használatával azon a jégmezőn, amin a tárgyat használó játékos áll, létrehoz egy sárat, ami menedéket nyújt az ott álló játékosoknak vihar esetén.

Régi, de megváltozott osztályok:

Building:

- **Leírás:** A Water táblán elhelyezkedő “épület”. A játékosok ide menekülhetnek be hóvihar esetén.
- **Metódusok:**
 - Damage(p:Player,e:Effect):void - Az épületbe berohanó játékos testhőjét csökkenti attól függően, hogy mi az effect.
 - Destroy():void - Nem okoz változást a modellben.

Iglu:

- **Leírás:** Ezen objektum gyakorlatilag egy épület típus. Ha egy mező rendelkezik ilyen épülettel, akkor hóvihar esetén az adott mezőn álló játékosok. Őse a Building osztály.
- **Metódusok:**
 - Damage(p:Player,e:Effect):void - Az épületbe berohanó játékos testhőjét csökkenti attól függően, hogy mi az effect.

Water:

- **Leírás:** Egy területegységet reprezentál, ami tárolhat tárgyakat és játékosokat, valamint fedheti valamennyi réteg hó.
- **Adattagok:**
 - snowLayers:int *
- **Metódusok:**
 - Storm():void*
 - AddPlayer(p:Player)*
 - RemovePlayer(p:Player)*
 - Digging(i:int):void*
 - EffectCall(e:Effect):void*
 - GetNeighbour(w:Water,i:int):Water *
 - GetCapacity():int*
 - AddFloatingItem(i:Item):void*
 - AddBear(b:Bear):void - Hozzáad egy medve objektumot a területegységhez.
 - RemoveBear(b:Bear):void - Eltávolít egy medve objektumot a területegységről.
 - DestroyBuiling():void - Ez a metódus hívja meg a rajta álló “épület” destroy() metódusát.

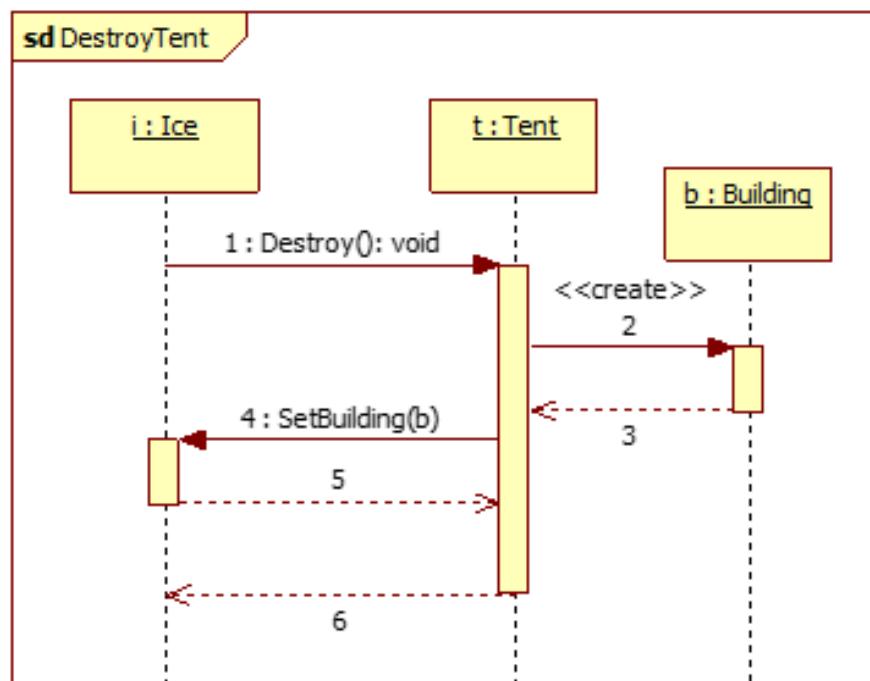
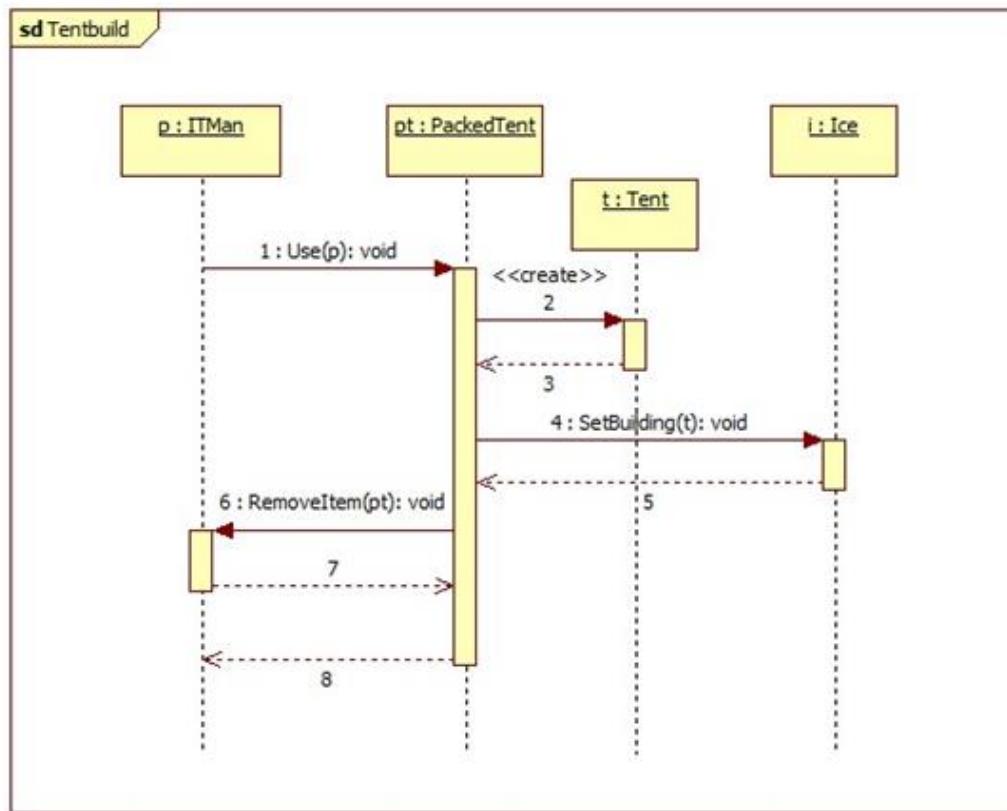
Player:

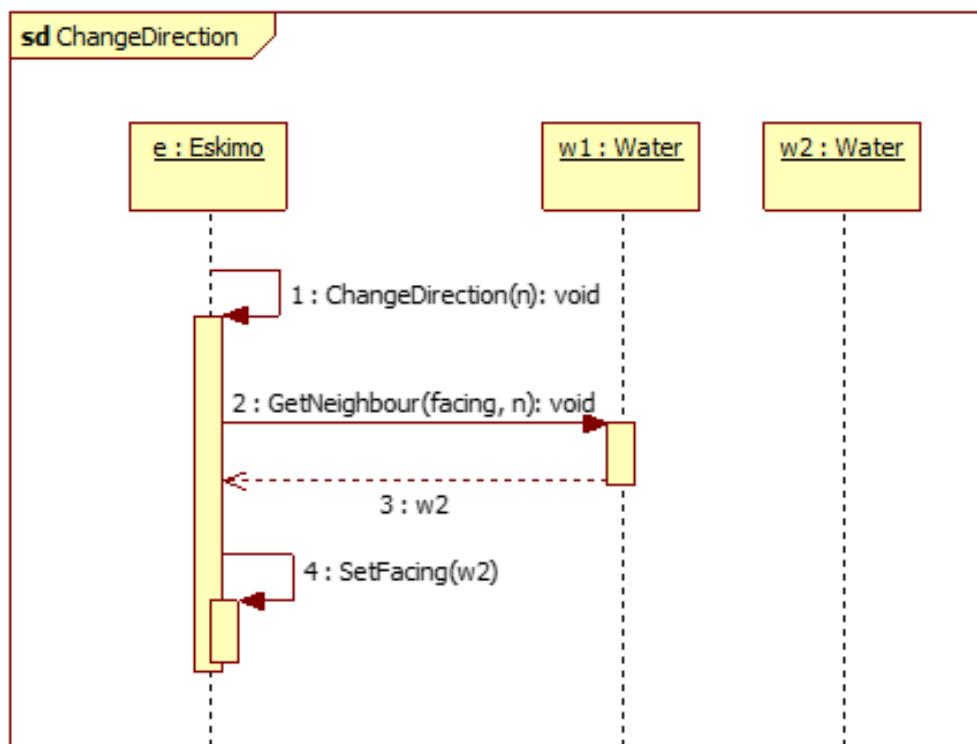
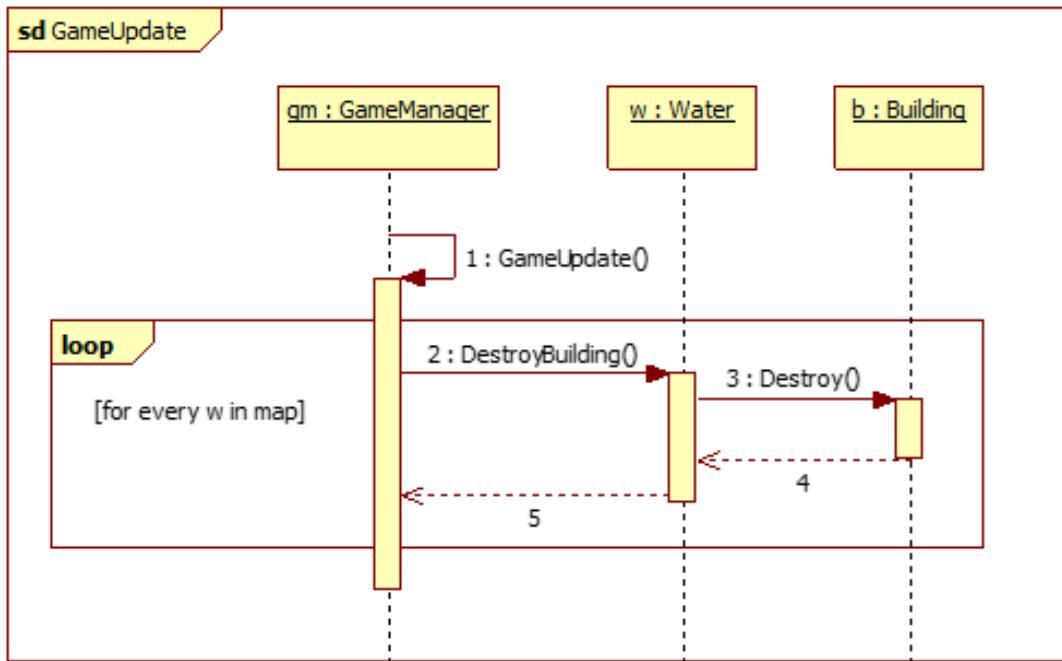
- **Leírás:** Egy absztrakt osztály, ami egy konkrét játékost reprezentál, ebből örökölnek a konkrét karakterek. Őse a Steppable.
- **Adattagok:**
 - name: String*
 - bodyHeat: int*
 - bodyHeatBase: int*
 - safe: bool*

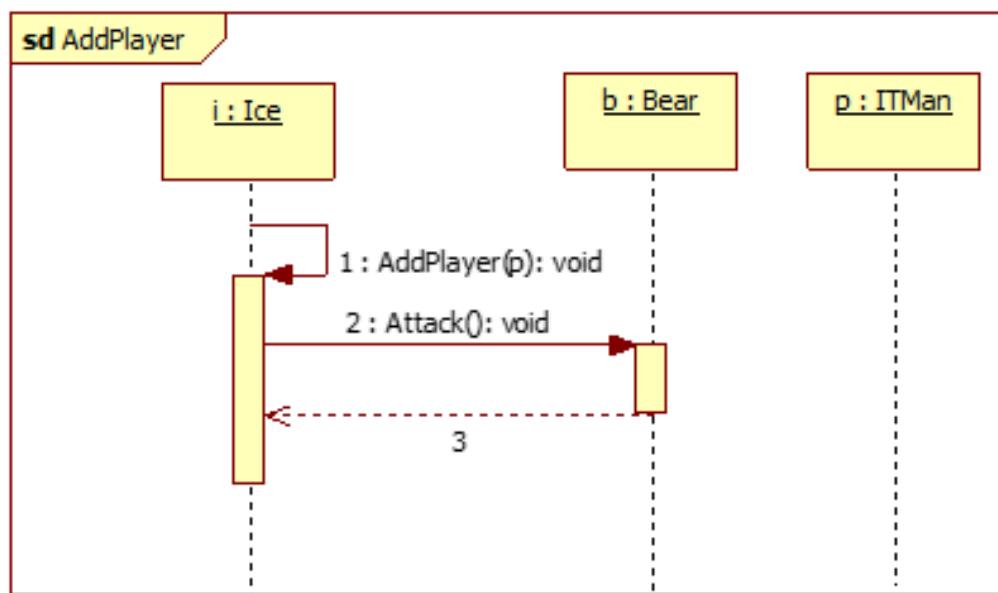
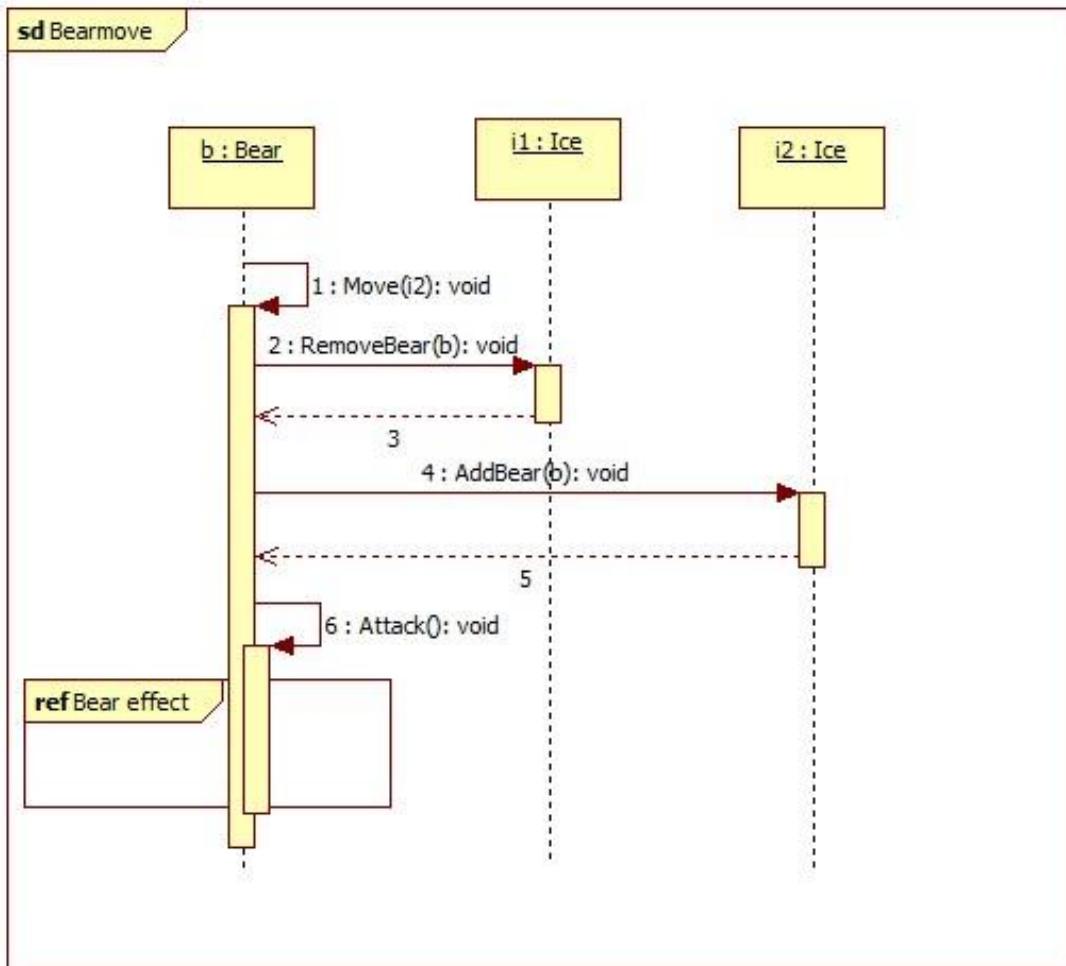
- **Metódusok:**

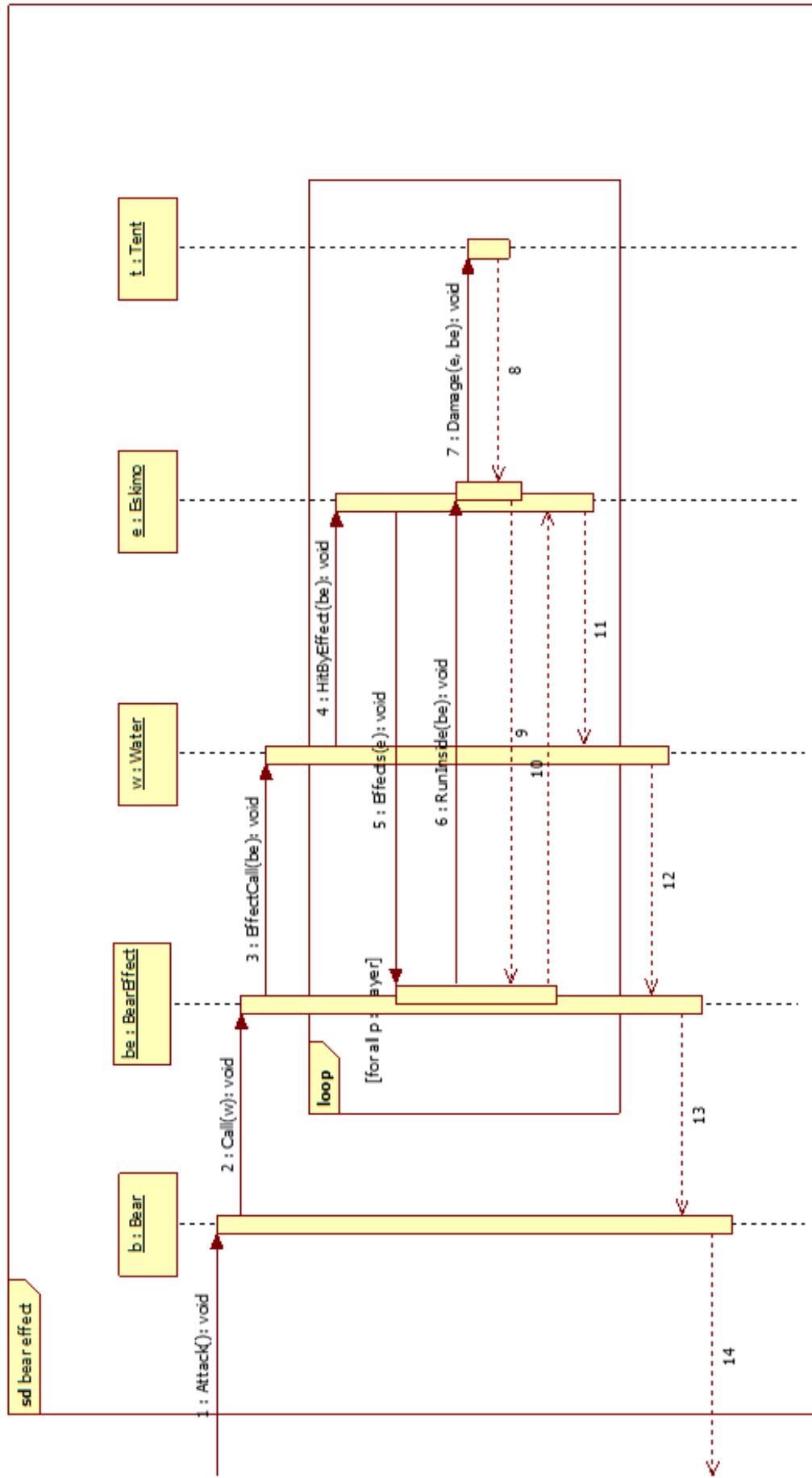
- Digging(): void*
- ChangeHeat(i: int): void*
- Craft(): void*
- AddItem(i: Item): void*
- DropItem(i: Item): void*
- RemoveItem(i: Item): void*
- Skill(): void*
- HitByEffect(e: Effect): void*
- RunInside(e: Effect): void - Működésében megegyezik az előzőekben specifikáltakkal, azzal a különbséggel, hogy paraméterül megkapja azt az effektust, ami érte a játékos, erre azért volt szükség, mert az új követelmények miatt, az épület típusától és az effektus fajtájától is függ, hogy mennyire érinti a hatása az adott játékost.

7.0.2 Szekvencia-diagramok









7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A szabványos bemenet segítségével adhatunk parancsokat az interfésznek, amennyiben a parancs valamilyen válasszal vagy üzenettel tér vissza, akkor az a szabványos kimeneten jelenik meg. Ezek alapján a be- és kimenteket átirányíthatjuk akár fájlba is, amit a segédprogram használni fog az automatikus tesztelés megvalósítása során.

Az általunk megvalósított parancsok lehetőséget teremtenek egy fájlban eltárolt parancsok beolvasására és futtatására egy parancssal. A tesztelés folyamán nem engedélyezzük egy mező szomszédainak manuális beállítását, mivel könnyen létrehozható olyan játéktér, ahol a játékosok számára nem lehetséges a szomszédos mezőkre való átlépés vagy a referenciaik rosszul kerülnek beállításra, így ezt a feladatot a konstruktora bízzuk.

7.1.2 Bemeneti nyelv

A beviteli nyelvben azok az értékek, amelyek nem számszerűsíthetők, pl. egy játékos “típusa”, egy meghatározott halmaz értékei közül kell választani. A halmaz a parancsok opcionálisan látható a következő formában: param: {value1, value2, ..., valueN}

AddFloatingItem

Leírás: Létrehoz egy adott típusú tárgyat a paraméterben megadott helyre és hozzáadja lebegő tárgyként.

Opciók: addfloatingitem <type> <water>

type: {"shovel", "packedtent", "food", "palesz", "rope", "divingsuit", "component"}

AddItem

Leírás: Létrehoz egy adott típusú tárgyat a paraméterben megadott játékosnak

Opciók: additem <type> <player>

type: {"shovel", "packedtent", "food", "palesz", "rope", "divingsuit", "component"}

ChangeDirection

Leírás: A paraméterben kapott játékos “facing” jeǵtábláját változtatja meg.

Opciók: changedirection <player>

Digging

Leírás: A paraméterben kapott játékos ásását valósítja meg a parancs, szintén paraméterben kapott területre.

Opciók: digging <player> <water>

DropItem

Leírás: Paraméterben meghatározott tárgyat dob el egy játékostól.

Opciók: dropitem <player> <item>

Load

Leírás: Betölti a paraméterben megadott pályát

Opciók: load <filename>

MovePlayer

Leírás: A játékost elmozgatja a paraméterben megadott helyre

Opciók: moveplayer <player> <water>

PickUpItem

Leírás: Paraméterben meghatározott tárgyat ad hozzá egy játékoshoz.

Opciók: pickupitem <player> <item>

Save

Leírás: Kimenti a paraméterben megadott fájlba az épp aktuális pályát

Opciók: save <filename>

Script

Leírás: A paraméterben kapott, parancsok neveit tartalmazó fájlból beolvasott utasításokat hajtja végre.

Opciók: script <filename>

Skill

Leírás: A paraméterben megadott játékos képességét használja

Opciók: skill <player>

TestMode

Leírás: Tesztmódba lépés. Adott parancsok használatának engedélyezése és letiltását lehet beállítani

Opciók: testmode <bool>

bool: {"true", "false"}

UseItem

Leírás: Paraméterként kapott tárgy használatát valósítja meg a parancs. Szintén paraméterben adható meg a játékos, akinél az item van.

Opciók: useitem <item> <player>

A tesztmódban elérhető további parancsok:**AddFrozenItem**

Leírás: Létrehoz egy adott típusú tárgyat a paraméterben megadott helyre és hozzáadja befagyotttárgyként.

Opciók: addfrozenitem <type> <water>

type: {"shovel", "packedtent", "food", "palesz", "rope", "divingSuit", "component"}

CreateBear

Leírás: Létrehoz egy medve típusú objektumot a paraméterben megadott helyre.

Opciók: createbear <water>

CreateBuilding

Leírás: Létrehoz egy épületet és elhelyezi a paraméterben kapott mezőre.

Opciók: createbuilding <type> <water>

type: {"building", "iglu", "tent"}

CreateMap

Leírás: A paraméterben megkapott sor- és oszlopszámú pályát létrehozza, illetve beállítja a szomszédsági viszonyokat. Az egyes területeket feltölti itemekkel.

Opciók: createmap <sor> <oszlop>

CreatePlayer

Leírás: Létrehoz egy adott típusú játékos a paraméterben megadott helyre

Opciók: createplayer <type> <water>

type : {“itman”, “eskimo”, “seal”, “researcher” }

CreateStorm

Leírás: Létrehoz egy storm-ot abban az esetben, ha a véletlenszerűség ki lett kapcsolva. A paraméterben kapott területre jön létre.

Opciók: createstorm <water>

List

Leírás: Kilistáz minden egyes objektumot a játékban azok attribútumaival és értékeivel együtt.

Opciók: list

LoseGame

Leírás: Játék végének ellenőrzése.

Opciók:losegame

MoveBear

Leírás: Paraméterben kapott medve mozgatása adott területre.

Opciók:movebear <bear> <water>

RandomStorm

Leírás: A vihar létrejöttének véletlenszerűségét beállító parancs.

Opciók: randomstorm <bool>

bool:{“true”,”false”}

7.1.3 Kimeneti nyelv

A bemeneti nyelvek közül egyiknek sincs kimenete, kivéve a ”list” parancsnak. Az összes bemeneti parancs hiba esetén hibaüzenetet ír ki.

A *list* parancs a szabványos kimeneten írja ki az adatokat, ezeket JSON fájlba fogjuk menteni, melynek szerkezete a következő struktúrát fogja követni:

Fields:

```

uid: f_x_x
neighbours:
    uid1
    uid2
    uid3
    uid4
frozenItems:
    uid1
    ....
floatingItems:
    uid1
    ....
building: b_(b/t/i)_x

```

Players:

```

uid: p_(e/i/s/r)
field:f_x_x
items:
    uid1
    uid2
    ...
bodyHeat: number
bodyHeatBase: number
safe: boolean

```

Items:

```

uid: i_(r/p/f/d/s/pt/c)_x
owner: uid
ice: uid
capacity: number

```

Bears:

```

uid: b_x
field: uid

```

7.2 Összes részletes use-case

A továbbiakban minden use-case-ben feltételezzük alternatív forgatókönyvként, hogy amennyiben egy parancs el lett írva, vagy rossz paraméterek lettek hozzárendelve, akkor a parancs nem fut le, beadását semmisnek tekintjük.

Use-case neve	CreateMap
Rövid leírás	A megadott pálya létrehozása
Aktorok	Controller
Forgatókönyv	1. A beadott paraméterek alapján felépít egy pályát, mely során létrehozza a mezőket, beállítja szomszédsági viszonyait és feltölti a jégtáblákat tárgyakkal.

Use-case neve	MovePlayer
Rövid leírás	A megadott játékos mozgatása
Aktorok	Player
Forgatókönyv	1 A játékos átlép a megadott táblára
Alternatív forgatókönyv	1.A.1 Ha víz akkor beleesik a játékos 1.A.2 Ha jégtábla akkor csak átlép a táblára

Use-case neve	CreatePlayer
Rövid leírás	A megadott táblára létrehozása egy játékost
Aktorok	Controller
Forgatókönyv	1.A megadott paraméterek alapján létrehoz egy adott játékos és hozzá adja a kiválasztott jégtáblára

Use-case neve	AddFloatingItem
Rövid leírás	Létrehoz és hozzáad egy tárgyat a jégtábla felszínéhez
Aktorok	Controller
Forgatókönyv	1. Egy tárgy hozzáadása a paraméterekekben meghatározott jégtáblára, a lebegő tárgyakhoz.

Use-case neve	AddFrozenItem
Rövid leírás	Létrehoz és hozzáad egy tárgyat a jégtáblába
Aktorok	Controller
Forgatókönyv	1. Egy tárgy hozzáadása a paraméterekekben meghatározott jégtáblára, a befagyott tárgyakhoz.

Use-case neve	AddItem
Rövid leírás	Létrehoz és hozzáad egy tárgyat a játékoshoz
Aktorok	Controller
Forgatókönyv	1. Egy tárgy létrehozása amelynek típusát és helyét a paraméterekekben határozzuk meg.

Use-case neve	ChangeDirection
Rövid leírás	Az adott játékos kiválasztott jégtábláját változtatja meg
Aktorok	Player
Forgatókönyv	1. A játékosnak megváltozik a kiválasztott jégtáblája és arra fog nézni

Use-case neve	Digging
Rövid leírás	A jégtábláról lapátol havat ha van rajta, ha nincs akkor a befagyott tárgyból ás ki egyet
Aktorok	Player
Forgatókönyv	1. A játékos a jégtáblán ásik
Alternatív forgatókönyv	1.A.1 Ha a jégtáblán hó van akkor azt fogja a játékos eltávolítani 1.A.2 Nincs hó akkor a jégből kiásik egy tárgyat a játékos

Use-case neve	Skill
Rövid leírás	Az adott játékos képességét használja
Aktorok	Player
Forgatókönyv	1. A játékos használja az egyedi képességét

Use-case neve	UseItem
Rövid leírás	A kiválasztott tárgyat használja
Aktorok	Player
Forgatókönyv	1. A játékos használja a kiválasztott a tárgyat

Use-case neve	CreateBuilding
Rövid leírás	Létrehoz és hozzáad egy épületet a jégtáblára
Aktorok	Controller
Forgatókönyv	1. Létrejön egy adott épület a megadott mezőn

Use-case neve	CreateBear
Rövid leírás	Létrehoz és hozzáad egy macit a jégtáblára
Aktorok	Controller
Forgatókönyv	1. Létrejön egy maci a kijelölt táblán

Use-case neve	CreateStorm
Rövid leírás	Létrehoz egy vihart az adott jégtáblán
Aktorok	Controller
Forgatókönyv	1. Létrehoz egy vihart azon a jégtáblán amit paraméterben megadtunk

Use-case neve	RandomStorm
Rövid leírás	A vihar véletlenszerűségét lehet ki és bekapsolni
Aktorok	Controller
Forgatókönyv	1. Beállítja a vihar véletlenség szerűségét
Alternatív forgatókönyv	1.A.1 A véletlenség szerűségét ki van kapcsolva akkor be lehet kapcsolni 1.A.2 A véletlenség szerűségét be van kapcsolva akkor ki lehet kapcsolni

Use-case neve	List
Rövid leírás	Kilistáz minden egyes objektumot a játékban azok attribútumaival és értékeivel együtt
Aktorok	Controller
Forgatókönyv	1. Kilistázza az összes objektumot és a hozzájuk tartozó adatokat az adott formátumban

Use-case neve	Script
Rövid leírás	A test eset automatizálásáért felelős
Aktorok	Controller
Forgatókönyv	1. Lefutattja a scriptet a parancs, amit paraméterben kapott.

Use-case neve	Load
Rövid leírás	Betölti egy pályát az adott fájlból
Aktorok	Controller
Forgatókönyv	1. A paraméterben kapott fájlból betölti a pályát.

Use-case neve	Save
Rövid leírás	Kimenti a pályát az adott fájlba
Aktorok	Controller
Forgatókönyv	1. Elmenti a játék pályát a paraméterben kapott fájlba.

Use-case neve	TestMode
Rövid leírás	Adott parancsok használatának engedélyezése és letiltását lehet beállítani
Aktorok	Controller
Forgatókönyv	1. Bekapcsolja a test mode-ot. 1.A Bekapcsolja a test mode-ot, de az már be volt kapcsolva. A test mode bekapcsolva marad.
Alternatív forgatókönyv	2. Kikapcsolja a test mode-ot. 2.A Kikapcsolja a test mode-ot, de az már ki volt kapcsolva. A test mode marad kikapcsolt állapotban.

Use-case neve	LoseGame
Rövid leírás	Játékos ellenőrzi, hogy véget ért-e,
Aktorok	Controller
Forgatókönyv	1. A játék állapotát ellenőrzi, majd a játékot megállítja.
Alternatív forgatókönyv	2. A játék állapotát ellenőrzi, majd nem állítja meg a játékot.

Use-case neve	MoveBear
Rövid leírás	Egy adott medvét mozgat egy paraméterben kapott területre.
Aktorok	Controller
Forgatókönyv	1. A paraméterben kapott medvét mozgatja az adott területre.

Use-case neve	DropItem
Rövid leírás	Paraméterben kapott játékostól eldobja a paraméterben kapott tárgyat.
Aktorok	Player
Forgatókönyv	1. Paraméterben kapott játékos eldobja az adott tárgyat.

Use-case neve	PickUpItem
Rövid leírás	Paraméterben kapott játékos felveszi a paraméterben kapott tárgyat.
Aktorok	Player
Forgatókönyv	1. Paraméterben kapott játékos felveszi az adott tárgyat.

7.3 Tesztelési terv

Teszt-eset neve	DiggingSnowLayer
Rövid leírás	Egy Player ás egy jégtáblán, amin még van höréteg.
Teszt célja	Azt vizsgáljuk, hogy mezőn lévő hörétegek száma csökkene-e.

Teszt-eset neve	DiggingItem
Rövid leírás	Egy Player ás egy jégtáblán, amin már nincs höréteg, és kiás egy Item-et.
Teszt célja	Azt vizsgáljuk, hogy a mezőből az ásás hatására egy Item kikerül a FrozenItem-ek közül (ha még van) és a FloatingItem-ekhez.

Teszt-eset neve	DropAnItem
Rövid leírás	Egy Player lehelyez egy Item-et a saját jégtáblájára
Teszt célja	Azt vizsgáljuk hogy a játékos tárgykészletéből eltűnt-e és a jégtábla FloatingItems listájába került-e az eldobott Item.

Teszt-eset neve	PickUpAnItem
Rövid leírás	Egy Player felvesz egy Item-et a saját jégtáblájáról.
Teszt célja	Azt vizsgáljuk hogy a játékos tárgykészletébe került-e és a jégtábla FloatingItems listájából eltűnt-e a felvett Item.

Teszt-eset neve	EskimoSkillOnIceWithTent
Rövid leírás	Az Eskimo használja a Skill-jét olyan jégen, amin Iglu van.
Teszt célja	Azt vizsgáljuk hogy azon a mezőn, amin az Eskimo áll, a Skill használata után Iglu áll-e, nem pedig Tent.

Teszt-eset neve	ITManSkill
Rövid leírás	Az Informatikus használja a képességét, megissza a Pálinkát.
Teszt célja	Azt vizsgáljuk, hogy a mezőn álló játékosok élete maximális lesz-e és hogy a Palesz eltűnt-e a játékos tárgykészletéből.

Teszt-eset neve	ResearcherSkill
Rövid leírás	A Sarkutató használja a képességét, megvizsgálja egy szomszédos mező teherbírását.
Teszt célja	Azt vizsgáljuk, hogy a mező teherbírása megegyezik ezzel, amit a sarkutató eredményül kapott.

Teszt-eset neve	StepOntoIce
Rövid leírás	Egy Player egy jégtáblára lép.
Teszt célja	Azt vizsgáljuk, hogy a játékos az egyik mezőből eltávolításra került, míg a másikhoz hozzáadódott.

Teszt-eset neve	StepOntoUnstableIce
Rövid leírás	Egy Player egy instabil jégtáblára lép.
Teszt célja	Megvizsgáljuk, hogy ha egy új játékos lép az instabil jégtáblára - ezzel túllépve annak teherbírását - akkor az beszakad-e.

Teszt-eset neve	StepOntoWater
Rövid leírás	Egy Player vízre jégtáblára lép.
Teszt célja	Azt vizsgáljuk, hogy a játékos az egyik mezőből eltávolításra kerül, míg a másikhoz hozzáadódik.

Teszt-eset neve	StepOntoFieldWithBear
Rövid leírás	A Player olyan jégre lép, amin Bear van.
Teszt célja	Azt vizsgáljuk, hogy ha egy játékos egy olyan mezőre lép amin egy medve van akkor a medve megtámadja-e a játékost.

Teszt-eset neve	UseComponent
Rövid leírás	Egy Player megkísérli összeépíteni a FlareGun-t, de nem jön össze.
Teszt célja	Egy játékos akinél kevesebb, mint 3 komponens van, használja az egyiket, a létrehozott FlareGun check attribútuma nem éri el a 3 értéket és a játéknak nincs vége.

Teszt-eset neve	UseComponentAndWin
Rövid leírás	Egy Player összeépíti a FlareGun-t.
Teszt célja	Egy játékos akinél 3 db komponens van, használja az egyiket, a létrehozott FlareGun check attribútuma eléri a 3 értéket és a játéknak vége.

Teszt-eset neve	UseFood
Rövid leírás	Egy Player elhasznál egy Food-ot.
Teszt célja	A Food eltűnik a játékos tárgykészletéből és a testhője eggel nő.

Teszt-eset neve	UseFoodOnMaxBodyHeat
Rövid leírás	Egy Player maximális testhőre használja a Food-ot.
Teszt célja	A Food eltűnik a játékos tárgykészletéből, de a testhője nem nő, mivel már maximumon van.

Teszt-eset neve	UseShovelToDigSnowLayer
Rövid leírás	Egy Player használja a Shovel-t olyan jégen, ahol még van höréteg.
Teszt célja	A Shovel durability-je csökken (ha 0-ra csökken, el is tűnik a játékos tárgykészletéből), a jégen lévő hörétegek száma max. 2-vel csökken.

Teszt-eset neve	UseShovelToDigItem
Rövid leírás	Egy Player használja a Shovel-t olyan jégen, ahol 0 höréteg van, és kiás egy Item-et.
Teszt célja	A Shovel durability-je csökken (ha 0-ra csökken, el is tűnik a játékos tárgykészletéből), a jégből egy FrozenItem (ha van) a FloatingItem-ekhez kerül.

Teszt-eset neve	UseRope
Rövid leírás	Egy Player használja a Rope-ot.
Teszt célja	A Rope-ot használó játékos facing attribútuma által meghatározott mezőről az összes játékosnak a használó játékos mezőjére kell kerülnie, a Rope-nak el kell tűnnie a használó játékos tárgykészletéből.

Teszt-eset neve	UseRopeInWrongDirection
Rövid leírás	Egy Player rossz irányban használja a Rope-ot, ezzel nem ment ki senkit.
Teszt célja	Nem mentődik ki senki (a Rope nem mozgat át játékost), de a Rope eltűnik a használó játékos tárgykészletéből.

Teszt-eset neve	UseDivingSuit
Rövid leírás	Egy Player használja a DivingSuit-ot.
Teszt célja	A játékos safe attribútumának igaznak kell lennie a teszt végén és el kell tűnnie a DivingSuitnak a játékos tárgykészletéből.

Teszt-eset neve	UseDivingSuitWithSafeAttributeOn
Rövid leírás	Egy Player használja a DivingSuit-ot úgy, hogy a safe attribútuma True.
Teszt célja	A safe attribútumnak True-nak kell maradnia.

Teszt-eset neve	UsePackedTent
Rövid leírás	Egy Player használja az PackedTent-et és lerak egy Tent-et a saját jégtáblájára..
Teszt célja	Egy Tent-nek kell létrejönnie a mezőn és a PackedTent-nek el kell tünnie a játékos tárgykészletéből.

Teszt-eset neve	UsePackedTentOnIceWithIglu
Rövid leírás	A Player PackedTent-et használ olyan jégen, amin Iglu van.
Teszt célja	A mezőn lévő Iglu-nak le kell cserélődni Tent-re.

Teszt-eset neve	GenerateStorm
Rövid leírás	A GameManager vihart generál.
Teszt célja	A kapott epicentrumon és annak környékén nő a hórétegek száma és az ott álló játékosos testhője csökken.

Teszt-eset neve	BearMoves
Rövid leírás	Egy Bear egy szomszédos mezőre lép.
Teszt célja	Azt vizsgáljuk, hogy a Bear az egyik mezőből eltávolításra került, míg a másikhoz hozzáadódott.

Teszt-eset neve	BearMovesAndAttacks
Rövid leírás	Egy Bear olyan szomszédos mezőre lép, ahol Player van.
Teszt célja	Azt vizsgáljuk, hogy a Bear az egyik mezőről eltávolításra került, míg a másikhoz hozzáadódott, és megtámadja-e a játékost, aki azon a mezőn van, amelyikre lépett.

Teszt-eset neve	StormEffectOnIceWithIglu
Rövid leírás	Vihar hatása jégen amin Iglu van.
Teszt célja	Vizsgáljuk, hogy azon a mezőn amelyiken Iglu van, a játékosok testhője nem változik.

Teszt-eset neve	StormEffectOnIceWithTent
Rövid leírás	Vihar hatása jégen amin Tent van.
Teszt célja	Vizsgáljuk, hogy azon a mezőn amelyiken Tent van, a játékosok testhője nem változik.

Teszt-eset neve	StormEffectOnIceWithBuilding
Rövid leírás	Vihar hatása jégen amin Building van.
Teszt célja	Vizsgáljuk, hogy azon a mezőn amelyiken Building van, a játékosok testhője csökken.

Teszt-eset neve	GameOver
Rövid leírás	A GameManager ellenőrzi, hogy vége lett-e a játéknak.
Teszt célja	Vizsgáljuk, hogy vége lett-e a játéknak úgy, hogy nem teljesül az alábbi feltételek egyike sem: <ul style="list-style-type: none"> ● Van olyan játékos, akinek a testhője 0 ● Van olyan játékos, aki vízben van és safe attribútuma False ● Van olyan játékos, aki egy medvével egy mezőn áll

Teszt-eset neve	GameOverWithPlayerOnWaterWithSafeAttributeFalse
Rövid leírás	A GameManager ellenőrzi, hogy vége lett-e a játéknak.
Teszt célja	Vizsgáljuk, hogy vége lett-e a játéknak úgy, hogy egy játékos, akinek a safe attribútuma False, vízben van.

Teszt-eset neve	GameOverWithPlayerWithZeroBodyHeat
Rövid leírás	A GameManager ellenőrzi, hogy vége lett-e a játéknak.
Teszt célja	Vizsgáljuk, hogy vége lett-e a játéknak úgy, hogy egy játékos testhője nulla.

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

Az automatikus tesztelés három különböző komponens együttműködéseként valósul meg. A proto program, ami magát játékot futtatja és rendelkezik parancsértelmezővel, ami az előzőekben definiált parancsokat fogadja el. A komparátor, ami képes eldönteni két fájl tartalmáról mennyire egyeznek meg. Valamint a PowerShell script (master), ami összehangolja az előző programok működését és irányítja azokat. (Utóbbi két komponens jelentik a segédprogramokat.)

Az automatikus tesztelés során a master script fordítja és futtatja a proto programot, illetve bead neki egy pályafájt betöltésre, majd a teszthez kapcsolódó parancsok listáját. Miután lefutottak a parancsok és a kimenet belekerült egy fájlba, a master elindítja komparátorit és összeveti a kimenetet az előre definiált elvárt kimenettel. Az eredményről tájékoztatást ad, hiba esetén is.

A fent említett folyamat annyiszor hajtódik végre, amennyi teszeset van. Miután lefutott minden teszt, a master megjeleníti, hogy az összes teszt hány százaléka végződött sikeresen.

7.5 Napló

Kezdet	Időtartam	Résznevők	Leírás
2020.04.02. 10:00	2,5 óra	Bodó Gaál Rédai Varga Nagy	Feladat megbeszélése, tisztázása.
2020.04.02. 18:00	3 óra	Bodó Gaál Rédai Varga Nagy	Osztálydiagram bővítése, módosítások elvégzése.
2020.04.03. 12:00	7 óra	Bodó Gaál Rédai Varga Nagy	Új szekvencia diagramok létrehozása, régebbiek frissítése. Parancsértelmező modul átbeszélése.
2020.04.04. 10:00	3,5 óra	Bodó Gaál Rédai Varga Nagy	Parancsértelmező modul definiálása. Use-case-ek meghatározása.
2020.04.04. 18:00	4 óra	Bodó Gaál Rédai Varga Nagy	Tesztesetek meghatározása és leírása.
2020.04.05. 10:00	3 óra	Bodó Gaál Rédai Varga Nagy	Dokumentum befejezése, véglegesítések, hibajavítások.

8. Részletes tervez

8.1. Osztályok és metódusok tervei.

Az egyes osztályokat a parancsokkal irányíthatóság miatt *unique id*-kal láttuk el, ahol szükséges volt. Ezzel oldottuk meg, hogy grafikus felület hiányában is egyedileg beazonosíthatók legyenek az objektumok, így elérve, hogy a parancs hatása a megfelelő példányokhoz jussanak el. Ez a módszer néhány osztályban plusz függvények használatát igényelte, melyek az alábbi leírásban megtalálhatóak.

8.1.1 Bear

- **Felelősség**

A játékban egy nem játszható karaktert valósít meg. Feladata a játékosok megölése, hogy a játék vereséggel érjen véget.

- **Ősosztályok**

Steppable

- **Attribútumok**

- **private BearEffect bearEffect:** a jegesmedve saját effectje (hogy ne kelljen minden Attack() hívásnál újat létrehozni)

- **Metódusok**

- **public void Attack():** Ez a függvény hívja meg a BearEffect egyik példányának a Call metódusát, ami a Bear osztály attribútum.

- **public void Work():** A steppable metódusa melyet felül definál a Bear osztály a medve tudjon véletlenszerűen lépni

8.1.2 BearEffect

- **Felelősség**

A játékban a medve rendelkezik egy BearEffect-el. A medve támadásakor egy effekt találja el a medvével egy mezőn lévő játékosokat. Ha egy játékos eltalál, vége a játéknak.

- **Ősosztályok**

Effect

- **Metódusok**

- **public void Effects(Player p):** Ez a függvény a GameManager Lose() metódusát hívja meg ami a játék végét eredményezi.

8.1.3 Building

- **Felelősség:**

A Water táblán elhelyezkedő “épület”. A játékosok ide menekülhetnek be (RunInside()) hóvihar esetén. Ez egy absztrakt osztály.

- **Attribútumok**

- **private Water water:** A Water amin a Building áll.

- **Metódusok**

- **public abstract void Damage(Player p, Effect e):** Egy absztrakt metódus amit majd az osztály gyermekei fognak megvalósítani.
- **public abstract void Destroy():** Egy absztrakt metódus amit majd az osztály gyermekei fognak megvalósítani.
- **public void SetWater(Water w):** A water attribútum értékét w-re állítja
- **public Water GetWater():** Visszaadja a water attribútum értékét
- **public void ReplaceBuilding(Building newB):** Az ezen a mezőn álló building referenciát cseréli ki a paraméterben kapott újra

8.1.4 Component

- **Felelősség**

Három komponensből állítható elő a FlareGun.

- **Ősosztályok**

Item

- **Metódusok**

- **public void Use(Player p):** Ha egy játékos használja, akkor meghívja a használó játékos Craft() függvényét.

8.1.5 DivingSuit

- **Felelősség**

Ha egy játékos viseli és vízbe esik, túléli. Ha csak a felszerelésében van, de még nem használta akkor nem.

- **Ősosztályok**

Item

- **Metódusok**
- **public void Use(Player p):** A használó játékos felveszi a búvárruhát, ezzel a játékos safe attribútumát true-ra állítja és a DivingSuit Item-et eltávolítja (RemoveItem()) a játékos eszköztárából. Meghívódik az ōsosztály Use() metódusa is.

8.1.6 Effect

- **Felelősség**

Absztrakt osztály. A jégtömbön álló játékosokat befolyásoló történés. (testhőváltozás, kimentés, medvetámadás, pálesz)

- **Metódusok**
- **public void Call(Water w):** A paraméterben megadott Water-nek meghívja a EffectCall() függvényét.
- **public abstract void Effects(Player p):** Maga a hatás, mely a játékost befolyásolja. Egy absztrakt metódus amit majd az osztály gyermekei fognak megvalósítani.

8.1.7 Eskimo

- **Felelősség**

Iglut tud építeni arra a jégtáblára, amin áll. Az Iglu a játék végéig ott marad (kivéve ha másik épülettel felül nem írják).

- **Ósosztályok**

Player

- **Metódusok**
- **public void Skill():** A függvényben létrehozunk egy Iglu, és a játékos jégtáblájának building referenciáját rá állítjuk (ezzel a mezőn korábban álló Building eltűnik a játékból).

8.1.8 FlareGun

- **Felelősség**

A komponensek összerakásával keletkezik a játékban. Automatikusan elsül és vége a játéknak.

- **Attribútumok**
- **private int accepted:** Mikor egy játékos szereli össze a FlareGun-t, azt számolja hány komponenst adott át eddig a játékos.
- **Metódusok**
- **public void Check(Item i):** Megvizsgálja, hogy a paraméterben átadott item egy komponens-e és ha igen, akkor az accepted értékét növeli eggyel. Ha az accepted értéke eléri a hármat akkor meghívódik a GameManager Win() függvénye. Meghívódik az ōsosztály Use() metódusa is.

8.1.9 Food

- **Felelősség**

Használatával az azt használó játékos testhője megnő eggyel.

- **Ősosztályok**

Item

- **Metódusok**

- **public void Use(Player p):** A Food item use metódusa meghívja a játékos ChangeHeat() metódusát (1) paraméterrel és utána a RemoveItem()-mel eltávolítja a játékos eszköztárából. Meghívódik az ősosztály Use() metódusa is.

8.1.10 GameManager

- **Felelősség** A játékot vezérli. Generálja a játékot, körönként meghívja a játékosokat, hogy mozogjanak, végezznek munkát. A körök végén véletlenszerűen hóvihart hoz létre, megnézi, hogy meghalt-e valaki, ha egy játékos elsüti a jelzőpisztolyt, akkor jelezni tudja ezen objektumnak, ami leállítja a játékot valamint ha egy Bear megtámad egy játékost, akkor a Lose() függvény hívódik meg amivel vége a játéknak.

- **Attribútumok**

- **private final int chanceOfStormLimit:** ha a chanceOfStorm értéke ennél kisebb, akkor garantáltan nincs esély hóvihar létrejöttére
- **private static int chanceOfStorm:** A hóvihar keletkezésének esélye. Körönként nő.
- **private static int chanceOfStormStep:** A hóvihar véletlen növekedésének mértéke.
- **private List<Player> players:** A játékosok listája.
- **private List<Water> map:** A játékban szereplő összes mező.
- **private List<Item> allItems:** A játékban szereplő összes tárgy listája.
- **private List<Bear> bears:** A jegesmedvék listája.

- **Metódusok**

- **public void Play():** Jelenleg nincs funkciója. A játékosok egymás utáni játszatásáért felelős. (Jelen verzióban a működését a parancsok helyettesítik)
- **public void GenerateStorm():** A függvényben vizsgáljuk a chanceOfStorm-ot és ha az értke 100% vagy nagyobb akkor kiválasztunk egy random Water-t a map listából és azon meghívjuk a Storm() függvényt. Ha viszont a feltétel nem teljesül akkor a chanceOfStorm-ot növeljük a egy random számmal ami a chanceOfStormStep intervallumában van.
- **public void SetChanceOfStormStep(int ch):**
- **public void GameUpdate():** A játékosok sorrendje minden kör után változik. (A kör végéig még egy éppen vízben lévő játékosot még egy soron lévő játékos ki tud menteni). A függvényben lekérdezi az összes játékos testhőjét a GetBodyHeat()-tel és vizsgálja, hogy a játékosok testhője 0-e vagy kisebb, ha igaz akkor meghívja a Lose() metódust. Itt vizsgáljuk azt is, hogy a játékosok a kör végén milyen mezőn vannak a GetCapacity()-val ha a visszatérési érték 0-val egyenlő akkor is meghívódika a Lose() függvény. Ezen kívül az összes Water-en meghívódik a DestroyBuilding().
- **public void Lose():** Véget vet a játéknak. Ez a függvény meghívódhat, ha GameUpdate()-ból vagy ha egy embert megtámad egy medve.

- **public void Win():** A jelzőpisztoly elsütésével ez a függvény hívódik meg, ami leállítja a játékot.
- **public void RelocateCrashedItems(List<Item> list):** Amikor egy instabil jégtábla betörök, az összes Item, ami a jégtáblán volt (ezt paraméterben kapja meg) átkerülnek véletlenszerűen választott jégtáblák frozenItems listáiba
- **public void ReplaceCrashedFields(Water oldW, Water newW):** A map lista oldW elemét lecseréli a kapott newW elemre (ezzel a helye nem változik a listában)
- **public void Reset():** a map, players és allItems listákat teljesen törli
- **public Object LookUp(String id):** A paraméterben megadott id-jű objektumot adja vissza (a keresett objektum lehet a map, players, bears illetve allItems listában is). Ha nincs ilyen id a játékban akkor egy null pointerrel tér vissza
- **public void CreatePlayer(String type, String uid):** Létrehoz egy adott típusú (itman, seal, eskimo, researcher) és id-jű játékost, ha még az az id-jű játékos nem létezik
- **public void CreateBear(String uid):** Létrehoz egy medvét a adott id-vel, ha olyan id-vel medve még nincs
- **public void CreateItem(String type, String uid):** Létrehoz egy adott típusú (shovel, food, packedtent, palesz, rope, divingsuit, component) és id-jű Item-et, ha még az az id-jű Item nem létezik
- **public void ListJson():** az összes objektumot kilistázza json formátumban
- **public void SaveGame(String file):** A paraméterben megadott fájlba menti ki a játékot json formátumban
- **public void LoadGame(String file):** A paraméterben megadott fájlból töltődik be a játék amely json formátumú

8.1.11 Ice

- **Felelősség**

A stabil jégtábla. Erre a jégtáblára akárhány játékos ráállhat, munkát végezhet rajta.

- **Ősosztályok**

Water

- **Metódusok**

- **public void Digging(int i):** Letakarít i hóréteget a jégtábláról. Ha már nem lenne hó, akkor pedig a frozenItems lista legfelső elemét helyezi el a floatingItems listába.
- **public void SetBuilding(Building b):** A building attribútumot a kapott paramétere állítja
- **public int GetCapacity():** Visszaadja mekkora a teherbírása a jégtáblának. Visszatérésben -1-et ad vissza, ami azt jelzi hogy bármennyien állhatnak rajta

8.1.12 Iglu

- **Felelősség**

Az iglu egy olyan épület amely megvédi a vele egy jégtáblán lévő játékosokat egy hóvihar vagy medvetámadás esetén.

- **Ősosztályok**

Building

- **Metódusok**

- **public void Damage(Player p, Effect e):** A berohanó játékosra van hatással az effect függvényében. Konkrétan ennél az épület típusnál nem történik semmi a játékossal, sem hóvihar, sem medvetámadás esetén szóval ez a függvény üres.
- **public void Destroy():** Ez a függvény nem csinál semmit az igluval, egy üres függvény.

8.1.13 Item

- **Felelősség**

Absztrakt osztály. Egy eszköz, ami a birtoklójának különböző tevékenységek elvégzésére ad lehetőséget. El lehet dobni egy jégtáblára, fel lehet venni jégtábláról, ki lehet ájni jégtáblákból.

- **Attribútumok**

- **private int durability:** A tárgy “életereje”. 0-ra csökkenéskor a tárgy elhasználódik és eltűnik a birtokló tárgykészletéből. minden leszármazottnak saját értékű.
- **private string uid:** Az objektum egyedi azonosítója.
- **private Player owner:** Referencia a játékosra, akinek a tárgy az items listájában van.

- **Metódusok**

- **public void Use(Player p):** Csökkenti a durability értékét 1-el. Ha eléri a 0-t, akkor törli a tárgyat a használó tárgykészletéből. Ezt a metódust minden leszármazott meghívja a saját Use() metódusában.
- **public int GetDurability():** Visszatér a durability értékével
- **public void SetDurability(int newDurability):** Beállítja a durability értékét a newDurability-re
- **public Player GetOwner():** Visszatér az owner-ben tárolt referenciával
- **public void SetOwner(Player newOwner):** Beállítja az owner értékét newOwner-re
- **public String GetUid():** Visszatér az uid értékével

8.1.14 ITMan

- **Felelősség**

Meg tudja inni a Pálinkát.

- **Ősosztályok**

Player

- **Metódusok**

- **public void Skill():** Üres metódus. Saját egyedi képessége nem itt kerül megvalósításra, hanem az interfésze és a Palesz item-je segítségével.

8.1.15 NullBuilding

- **Felelősség**

A Water táblán elhelyezkedő “épület”. A játékosok ide menekülhetnek be hóvihar esetén.

- **Ősosztályok**

Building

- **Metódusok**

- **public void Damage(Player p, Effect e):** Ha e egy bearEffect példánya, meghívja a GameManager Lose() metódusát. Ha e egy StormEffect, akkor az épületbe berohanó játékosok ChangeHeat()-ét meghívja -1 paraméterrel.
- **public void Destroy():** Egy üres függvény mivel a NullBuilding az alap épület, nem lehet megsemmisíteni.

8.1.16 PackedTent

- **Felelősség**

Egy olyan tárgy, aminek a használatával egy Tent jelenik meg a játékos mezőjén. A Tent megvédi a vele egy jégtáblán lévő játékosokat a hóvihar esetén, de egy medve támadásától nem. Csak egy körre marad a jégtáblán, utána lecserélődik NullBuilding-re.

- **Ősosztályok**

Item

- **Metódusok**

- **public void Use(Player p):** Létrehoz egy Tent példányt és a használó játékos mezőjének building paraméterét beállítja a létrehozott Tent referenciájára (SetBuilding()-et használva) és ezt követően eltávolítja a PackedTent-et a játékostól. Meghívódik az ősosztály Use() metódusa is.

8.1.17 Palesz

- **Felelősség** Csak az ITMan tudja meginni. Használatkor a használó jégtábláján álló játékosok testhője maximumra nő.

- **Ősosztályok:**

Effect

- **Metódusok**

- **public void Use(Player p):** Vizsgálja, hogy a paraméterben kapott játékos megvalósítja-e az Alcoholic interface-t, és ha igen, akkor létrehoz egy új PaleszEffect-et és meghívja rajta a Call() függvényt. Meghívódik az ősosztály Use() metódusa is.

8.1.18 PaleszEffect

- **Felelősség**

Arra a jégtáblára hívódik, amelyen a Palesz-t használó áll, mely továbbítja a kérést minden az adott mezőn álló játékosnak. Az “eltalált” játékosokat testhőjét maximumra állítja.

- **Ősosztályok**

Effect

- **Metódusok**

- **public void Effects(Player p):** A játékos bodyHeat ttribútumát a bodyHeatBase értékére állítja.

8.1.19 Player

- **Felelősség**

Absztrakt osztály. Egy ember által irányított játékosot reprezentál. Tud lépni, munkát végezni és meg tud halni. Egy munkának számít egy tárgy használata és a karaktertípus saját képességének használata is.

- **Ősosztályok**

Steppable

- **Attribútumok**

- **private final int maxInventory:** Egy játéknál lévő maximális Item-ek száma
- **private int bodyHeat:** A játékos testhője. Ha nullára csökken, meghal.
- **private int bodyHeatBase:** A játékos maximális testhője, ennél soha nem lehet több. A játék elején ennyivel indul.
- **private List<Item> items:** A játékos tárgyai itt tárolódnak el.
- **private boolean safe:** Ha igazra van állítva az azt jelenti, hogy az adott játékos ha a kör végén vízben van, akkor nem hal meg. (A fóka játéknál ez az érték alapértelmezetten igaz, míg más játékosok úgy tudják módosítani, hogy egy DivingSuit Item-et használnak)

- **Metódusok**

- **public void Work():** Jelenleg nincs funkcionálása. A működését jelen verzióban a parancsok helyettesítik.
- **public void Digging():** A játékos annak a Waternek a Digging metódusát hívja meg 1-es paraméterrel amin áll.
- **public void ChangeHeat(int i):** A paraméterben megadott értékkel változik a játékos bodyHeat-je, és ha több lenne, mint a bodyHeatBase, akkor a bodyHeat egyenlő lesz a bodyHeatBase értékével.
- **public void Craft():** Létrehoz egy FlareGun-t és a Check() metódusát a játékos összes Item-ével meghívjuk.
- **public void AddItem(Item i):** Megvizsgáljuk, hogy az items listában lévő eszközök száma kisebb-e mint a maxInventory. Ha ez igazzal tér vissza akkor hozzáadódik a Item a játékoshoz, egyébként visszakerül a jég FloatingItem-jeihez.

- **public void DropItem(Item i):** A kiválasztott Item, amit paraméterben kap meg, azt hozzáadja a jégtábla FloatingItems listájához és eltávolítja a játékos Itemei közül.
- **public void PickUpItem(Item i):** A kapott i Item-et törli a játékos mezőjének floatingItems listájából a RemoveFloatingItems() metódus hívásával, majd megpróbálja hozzáadni a saját tárgykészletéhez (items lista)
- **public void RemoveItem(Item i):** Elveszi a kapott tárgyat a játékos felszereléséből (items).
- **public void Skill():** Absztrakt metódus a leszármazottak egyedi képességének használatához.
- **public void HitByEffect(Effect e):** Az adott mezőre érkező effektus “eltalálja” a rajta álló játékosokat, ami ezen függvény hívásával történik. Az effektus típusától függően valamilyen hatást vált ki a játékosra.
- **public void RunInside(Effect e):** A függvény használatával a játékos “bemenekülhet” a mezőn lévő épületbe hóvihar esetén. Meghívjuk a jégtáblán lévő épület Damage() függvényét.
- **public int GetBodyHeat():** Visszatér a bodyHeat értékével
- **public void SetBodyHeat(int i):** Beállítja a bodyHeat attribútum értékét i-re
- **public int GetBodyheatBase():** Visszatér a bodyHeatBase értékével
- **public void SetBodyHeatBase(int i):** Beállítja a bodyHeatBase attribútum értékét i-re
- **public void SetSafe(boolean newSafe):** Beállítja a safe attribútum értékét newSafe-re
- **public boolean GetSafe():** Visszatér a safe értékével
- **public void Die():** Megpróbálja “megölni” a játékost. Ez akkor sikerül, ha a játékos testhője 0, vagy False értékű safe attribútum mellett a játékos Water-en áll. Siker esetén meghívja a GameManager Lose() metódusát.

8.1.20 Researcher

- **Felelősség**

Meg tudja vizsgálni egy szomszédos jégtábla teherbírást.

- **Ősosztályok**

Player

- **Metódusok**

- **public void Skill():** Megvizsgál a facing által meghatározott szomszédos jégtáblát (GetCapacity()) és visszakapja annak a teherbírást (hány embert bír el).

8.1.21 Rope

- **Felelősség**

Ki lehet vele menteni egy szomszédos mezőn vízbe esett játékosokat.

- **Ősosztályok**

Item

- **Metódusok**

- **public void Use(Player p):** Használatával a facing által meghatározott szomszédos jegtáblára meghívja a RopeEffect egy új példányát. A vízbe esett játékosokat kimenekíti a használó jegtáblájára (mindenkit). Meghívódik az ōsosztály Use() metódusa is.

8.1.22 RopeEffect

- **Felelősség**

A jegtáblára hívódik, amely továbbítja a kérést. Az “eltalált” játékosok, akik a kötél használatának irányában (facing) vannak, azokat kimenekíti a hívó mezőjére.

- **Ósosztályok**

Effect

- **Attribútumok**

- **private Water targetWater:** Ide kell kimenekíteni a játékosokat. Ezen a jegtáblán áll a kimentő játékos, aki használta a kötelet.

- **Metódusok**

- **public void Effects(Player p):** A játékosra van hatással, meghívja rájuk a move-ot a targetWater irányba.

8.1.23 Seal

- **Felelősség**

A kör végén búvárruha nélkül is túléli a vízbe esést.

- **Ósosztályok**

Player

- **Metódusok**

- **public void Skill():** Üres metódus. A különleges képessége karakterből adódóan automatikusan működik.

8.1.24 Shovel

- **Felelősség**

Használatával egyszerre két réteg havat lehet ellapárolni arról a jegtábláról, amin a játékos áll. Használata csökkenti a durability-t. 0 durability elérésekor eltűnik a Játékos felszerelései közül. (Követelmények miatt lesz olyan ásó aminek 3 a durability-je)

- **Ósosztályok**

Item

- **Metódusok**
- **public void Use(Player p):** A használó jégtáblájáról eltávolít két réteg havat vagy kiás egy darab tárgyat, ha a tábla előzetesen tiszta volt. Meghívódik az űosztály Use() metódusa is.

8.1.25 Steppable

- **Felelősség**

Absztrakt osztály. Azok az osztályok származhatnak ebből az osztályból, amelyek mozoghatnak a játékban.

- **Attribútumok**
- **private Water facing:** az a mező, amelyikre a leszármazott “néz” (Move esetén erre a mezőre kerül)
- **private Water water:** az a mező, amelyiken a leszármazott áll
- **private String uid:** Az objektum egyedi azonosítója.
- **Metódusok**
- **public void Work():** A játékost dolgoztató függvény. Ha a játékos vízben van, akkor nem dolgozhat. Ha van bűvárruhája akkor a vízben csak mozoghat. A jégtáblán bármilyen munkát végezhet.
- **public void Move(Water w):** A játékos átléphet egy a paraméterben szereplő másik mezőre.
- **public void ChangeDirection(int n):** A játékos irányváltására szolgál. Meghívva módosítja a játékos facing attribútumát a saját jégtáblájának szomszéelistájában szereplő water-re, amely a paraméterben kapott “n” változó távolságra van tőle. Ha a paraméterben kapott szám túl nagy, akkor egy újabb kör kezdődik a kiválasztásnál.
- **public Water GetWater():** Visszatér a water attribútumban tárolt referenciával
- **public void SetWater(Water w):** A water attribútum értékét a kapott referenciára állítja
- **public void SetFacing(Water w):** A facing attribútum értékét a kapott referenciára állítja
- **public String GetUid():** Visszatér a uid értékével

8.1.26 StormEffect

- **Felelősség**

Egy jégtáblára hívódik, a rajta lévő játékosoknak meghívja a RunInside() metódusát

- **Ősztályok**

Effect

- **Metódusok**
- **public void Effects(Player p):** Meghívja az “eltalált” játékos RunInside() metódusát.

8.1.27 Tent

- **Felelősség**

Megvédi a hóvihar hatása elől a mezőn álló játékosokat (nem csökken a testhőjük), de a jegesmedve elől nem nyújt védelmet. minden kör végén eltűnik.

- **Ősosztályok**

Building

- **Metódusok**

- **public void Damage(Player p, Effect e):** Üres metódus
- **public void Destroy():** Létrehoz egy NullBuilding-et. Az összes referenciát ami rá (Tent) mutat, az újonnan létrehozott NullBuilding-re állítja (ezzel kitörli magát a játékból).

8.1.28 UnstableIce

- **Felelősség**

Játékosokat (Player), tárgyakat (Item), jegesmedvéket (Bear), szomszédos mezőket (Water) és épületet (Building) tárol. At tud adni és át tud venni játékost és jegesmedvét más mezőktől (Add-/RemovePlayer/-Bear), Effect-et tud továbbítani a rajta álló játékosok felé. Ha a játékosok száma túllépi a capacity értékét, a jég vízzé alakul.

- **Ősosztályok**

Ice

- **Attribútumok**

- **private int capacity:** A maximális játékosok száma, akik a jégtáblán egyszerre állhatnak

- **Metódusok**

- **private void Crash():** Létrehoz egy Water-t. Az összes referenciát ami rá (UnstableIce) mutat (a szomszédos Water-ek neighbour listájában és a GameManager map listájában), az újonnan létrehozott Water-re állítja (ezzel kitörli magát a játékból).
- **public int GetCapacity():** Visszaadja a capacity értékét
- **public void AddPlayer(Player p):** Úgy működik, ahogy a Water-nél, kivéve, ha a játékosok száma túllépi a capacity értékét, meghívódik a Crash() metódus

8.1.29 Water

- **Felelősség**

Játékosokat (Player), tárgyakat (Item), jegesmedvéket (Bear), szomszédos mezőket (Water) és épületet (Building) tárol. At tud adni és át tud venni játékost és jegesmedvét más mezőktől (Add-/RemovePlayer/-Bear), Effect-et tud továbbítani a rajta álló játékosok felé.

- **Attribútumok**
- **private final int maxSnowLayers:** A mezőn lévő hörétegek maximális száma
- **private String uid:** Az objektum egyedi azonosítója.
- **private int snowLayers:** A mezőn lévő hörétegek száma
- **private List<Water> neighbours:** A szomszédos mezők
- **private List<Player> players:** A mezőn lévő játékosok
- **private List<Bear> bears:** A mezőn lévő jegesmedvék
- **private List<Item> frozenItems:** A mezőbe fagyott Item-ek
- **private List<Item> floatingItems:** A mezőn lévő tárgyak
- **private Building building:** A mezőn lévő épület

- **Metódusok**
- **public void AddSteppable(Steppable s):** Hozzáadja a kapott steppable objektumot a mező bears vagy players listájához - az objektum típusának megfelelő listába
- **public void RemoveSteppable(Steppable s):** Törli a kapott Steppable objektumot a mező bears vagy players listájából - az objektum típusának megfelelő listából
- **public void Storm():** Létrehoz egy StormEffect-et, amivel meghívja a players lista összes elemének HitByEffect() metódusát
- **public void StormCenter():** Meghívja a saját és a neighbours lista összes elemének Storm() metódusát
- **public void Digging(int i):** Üres metódus
- **public void EffectCall(Effect e):** A kapott e Effect-el meghívja a players listában szereplő összes Player HitByEffect() metódusát
- **public Water GetNextNeighbour(Water w):** Visszaadja a neighbours listából azt az elemet, ami a kapott w elem utáni első eleme a listának (túlindexelés esetén a lista első elemét adja vissza)
- **public void AddNeighbour(Water w):** A neighbours lista végére teszi a kapott w Water-t
- **public int GetCapacity():** 0-t ad vissza
- **public int GetSnowLayers():** Visszatér a snowLayers értékével
- **public void AddSnowLayers(int i):** Megnöveli a snowLayers értékét i-vel
- **public void SetSnowLayers(int i):** Beállítja a snowLayers értékét i-re
- **public Building GetBuilding():** Visszaadja a building attribútumban tárolt Building referenciáját
- **public void SetBuilding(Building b):** Beállítja a building attribútum értékét a kapott referenciára
- **public void AddFrozenItem(Item i):** A frozenItems lista végére teszi a kapott Item-et
- **public void RemoveFrozenItem(Item i):** A frozenItems lista 0. elemét törli a listából
- **public void AddFloatingItem(Item i):** A floatingItems lista végére teszi a kapott Item-et
- **public void RemoveFloatingItem(Item i):** A floatingItems lista 0. elemét törli a listából
- **public void DestroyBuilding():** Meghívja a building attribútumban tárolt Building leszármazott Destroy() függvényét
- **public void SetNeighbour(Water oldW, Water newW):** A neighbours listában szereplő oldW mezőt a newW mezőre cseréli (a listában elfoglalt helye nem változik)
- **public String GetUid():** Visszatér a mező egyéni azonosítójával (uid)

8.2. A tesztek részletes tervezet, leírásuk a teszt nyelvén

A tesztek során a felhasználó számára nem engedélyezzük a pálya felépítését, vagy új mezők hozzáadását. minden teszt egy előre létrehozott és teljesen felépített tesztpályán történik. A pálya az alább látható négy mezőből áll, melyeken kezdetben szereplőket és tárgyakat helyeztünk el, valamint beállítottuk azok tulajdonságait. A tesztek elején ezen játékeret töltjük be és ahol szükséges további tárgyakkal vagy játékosokkal egészítjük ki, majd a bemenetben megadott parancsokat futtatjuk, ezzel lefut a teszt. A kimenetet a *list* parancs segítségével állítjuk elő, melynek minden esetben átadjuk az objektum uid-kat, melyeket végig nézve megállapítható, hogy jól futnak-e a tesztek.

{}: teherbírás

(): hőrétegek száma

<>: (safe ? "s" : "us") + ", " + teshő

[]: items

<p>[0, 0]</p> <p>Type</p> <p>Ice:f_0_0 {inf} (2)</p> <p>Frozen[]</p> <p>Floating[Food:i_f_0]</p> <p>Steppables</p> <p>ITMan:p_i<us, 1></p> <p>[]</p> <p>Eskimo:p_e<us, 0></p> <p>[]</p> <p>Building</p> <p>Tent</p>	<p>[0, 1]</p> <p>Type</p> <p>UnstableIce:f_0_1 {2} (0)</p> <p>Frozen[Food:i_f_1, Component:i_c_0]</p> <p>Floating[]</p> <p>Steppables</p> <p>Seal:p_s<s, max></p> <p>[]</p> <p>Researcher:p_r<us, 3></p> <p>[]</p> <p>Building</p> <p>Iglu</p>
<p>[1, 0]</p> <p>Type</p> <p>Ice:f_1_0{inf}</p> <p>Frozen[]</p> <p>Floating[]</p> <p>Steppables</p> <p>-</p> <p>Building</p> <p>NullBuilding</p>	<p>[1, 1]</p> <p>Type</p> <p>Water:f_1_1 {0} (1)</p> <p>Frozen[]</p> <p>Floating[]</p> <p>Steppables</p> <p>-</p> <p>Building</p> <p>NullBuilding</p>

8.2.1. DiggingSnowLayer

- **Leírás**

Egy Player ás egy jégtáblán, amin még van höréteg.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt vizsgáljuk, hogy mezőn lévő hörétegek száma csökken-e.

Lehetséges hibák:

- A jégtáblán nem csökken a höréteg
- A höréteg csökkentése helyet egy tárgyat ás ki a jégtáblából

- **Bemenet**

```
load map.json
digging p_e
list f_0_0
```

- **Elvárt kimenet**

```
[
  {
    "uid": "f_0_0",
    "type": "Ice",
    "neighbours": [
      {
        "uid": "f_0_1"
      },
      {
        "uid": "f_1_0"
      }
    ],
    "frozenItems": [],
    "floatingItems": [
      {
        "uid": "i_f_0"
      }
    ],
    "snowlayers": 1,
    "building": "NullBuilding",
    "players": [
      {
        "uid": "p_i"
      },
      {
        "uid": "p_e"
      }
    ],
    "bears": []
  }
]
```

8.2.2. DiggingItem

- **Leírás**

Egy Player ás egy jégtáblán, amin már nincs höréteg, és kiás egy Item-et.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt vizsgáljuk, hogy a mezőből az ásás hatására egy Item kikerül-e a FrozenItem-ek közül (ha még van) és bekerül a FloatingItem-ekhez.

Lehetséges hibák:

- Nem kerül ki Item a jégtáblára akkor se ha van még Item a jégtáblában
- Több Item kerül ki egy ásással
- A kiásott Item nem kerül a FloatingItem-ekhez
- A kiásott Item nem tűnik el a FrozenItem-ek közül

- **Bemenet**

```
load map.json
digging p_r
list f_0_1
```

- **Elvárt kimenet**

```
[
  {
    "uid": "f_0_1",
    "type": "UnstableIce",
    "neighbours": [
      {
        "uid": "f_1_1"
      },
      {
        "uid": "f_0_0"
      }
    ],
    "frozenItems": [
      {
        "uid": "i_c_0"
      }
    ],
    "floatingItems": [
      {
        "uid": "i_f_1"
      }
    ],
    "snowlayers": 0,
    "building": "NullBuilding",
    "players": [
      {
        "uid": "p_s"
      },
      {
        "uid": "p_r"
      }
    ]
  }
]
```

```

        }
    ],
    "bears": []
}
]
```

8.2.3. DiggingItemWithEmptyFrozenItemList

- **Leírás**

Egy Player ás egy jégtáblán, amin már nincs hóréteg, és kiás egy Item-et.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt vizsgáljuk, hogy a mezőből az ássás hatására egy Item kikerül-e a FrozenItem-ek közül (ha még van) és bekerül a FloatingItem-ekhez.

Lehetséges hibák:

- Nem kerül ki Item a jégtáblára akkor se ha van még Item a jégtáblában
- Több Item kerül ki egy ásással
- A kiásott Item nem kerül a FloatingItem-ekhez
- A kiásott Item nem tűnik el a FrozenItem-ek közül

- **Bemenet**

```
load map.json
setfacing p_e f_1_0
move p_e
digging p_e
list f_1_0
```

- **Elvárt kimenet**

```
[
  {
    "uid": "f_1_0",
    "type": "UnstableIce",
    "neighbours": [
      {
        "uid": "f_0_0"
      },
      {
        "uid": "f_1_1"
      }
    ],
    "frozenItems": [],
    "floatingItems": [],
    "snowlayers": 0,
    "building": "NullBuilding",
    "players": [
      {
        "uid": "p_e"
      }
    ]
}
```

```
],
  "bears": []
}
]
```

8.2.4. DropAnItem

- **Leírás**

Egy Player lehelyez egy Item-et a saját jégtáblájára

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt vizsgáljuk hogy a játékos tárgykészletéből eltűnt-e és a jégtábla FloatingItems listájába került-e az eldobott Item.

Lehetséges hibák:

- A tárgy benne marad a játékos tárgykészletében
- A tárgy nem kerül a FloatingItems listájába de eltűnik a játékos tárgykészletéből

- **Bemenet**

```
load map.json
additem shovel i_s_0 p_r
dropitem p_r i_s_0
list p_r f_0_1
```

- **Elvárt kimenet**

```
[
  {
    "uid": "p_r",
    "type": "Researcher",
    "field": "f_0_1",
    "facing": "f_1_1",
    "items": [],
    "bodyHeat": 3,
    "bodyHeatBase": 4,
    "safe": false
  },
  {
    "uid": "f_0_1",
    "type": "UnstableIce",
    "neighbours": [
      {
        "uid": "f_1_1"
      },
      {
        "uid": "f_0_0"
      }
    ],
    "frozenItems": [
      {
        "uid": "f_1_1"
      }
    ]
  }
]
```

```

    "uid": "i_f_1"
  },
  {
    "uid": "i_c_0"
  }
],
"floatingItems": [
  {
    "uid": "i_s_0"
  }
],
"snowlayers": 0,
"building": "NullBuilding",
"players": [
  {
    "uid": "p_s"
  },
  {
    "uid": "p_r"
  }
],
"bears": []
}
]

```

8.2.5. PickUpAnItem

- **Leírás**

Egy Player felvesz egy Item-et a saját jégtáblájáról.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt vizsgáljuk hogy a játékos tárgykészletébe került-e és a jégtábla FloatingItems listájából eltűnt-e a felvett Item.

Lehetséges hibák:

- A tárgy nem kerül be a játékos tárgykészletébe, de a FloatingItems listájából eltűnt
- A tárgy amit felvettünk nem tűnik el a FloatingItems listájából

- **Bemenet**

load map.json
 pickupitem p_e i_f_0
 list p_e f_0_0

- **Elvárt kimenet**

```
[
  {
    "uid": "p_e",
    "type": "Eskimo",
    "field": "f_0_0",
    "facing": "f_0_1",
  }
]
```

```

"items": [
  {
    "uid": "i_f_0"
  }
],
"bodyHeat": 0,
"bodyHeatBase": 5,
"safe": false
},
{
  "uid": "f_0_0",
  "type": "Ice",
  "neighbours": [
    {
      "uid": "f_0_1"
    },
    {
      "uid": "f_1_0"
    }
  ],
  "frozenItems": [],
  "floatingItems": [],
  "snowlayers": 2,
  "building": "NullBuilding",
  "players": [
    {
      "uid": "p_i"
    },
    {
      "uid": "p_e"
    }
  ],
  "bears": []
}
]

```

8.2.6. EskimoSkillOnIceWhitTent

- **Leírás**

Az Eskimo használja a Skill-jét olyan jégen, amin Tent van.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt vizsgáljuk hogy azon a mezőn, amin az Eskimo áll, a Skill használata után Iglu áll-e, nem pedig Tent.

Lehetséges hibák:

- Az Iglu nem jön létre a jégtáblán továbbra is a Tent van ott

- **Bemenet**

load map.json

skill p_e

list f_0_0

- **Elvárt kimenet**

```
[  
  {  
    "uid": "f_0_0",  
    "type": "Ice",  
    "neighbours": [  
      {  
        "uid": "f_0_1"  
      },  
      {  
        "uid": "f_1_0"  
      }  
    ],  
    "frozenItems": [],  
    "floatingItems": [  
      {  
        "uid": "i_f_0"  
      }  
    ],  
    "snowlayers": 2,  
    "building": "Iglu",  
    "players": [  
      {  
        "uid": "p_i"  
      },  
      {  
        "uid": "p_e"  
      }  
    ],  
    "bears": []  
  }  
]
```

8.2.7. ITManSkill

- **Leírás**

Az Informatikus használja a képességét, megissza a Pálinkát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt vizsgáljuk, hogy a mezőn álló játékosok élete maximális lesz-e és hogy a Palesz eltűnt-e a játékos tárgykészletéből.

Lehetséges hibák:

- Nem az összes játékos testhője lesz maximum
- Nem lesz maximum a játékosok testhője

- A Palesz nem tűnik el a ITMan tárgykészletéből

- **Bemenet**

```
load map.json
additem palesz i_p_0 p_i
useitem i_p_0 p_i
list p_i p_e
```

- **Elvárt kimenet**

```
[
  {
    "uid": "p_i",
    "type": "ITMan",
    "field": "f_0_0",
    "facing": "f_0_1",
    "items": [],
    "bodyHeat": 3,
    "bodyHeatBase": 3,
    "safe": false
  },
  {
    "uid": "p_e",
    "type": "Eskimo",
    "field": "f_0_0",
    "facing": "f_0_1",
    "items": [],
    "bodyHeat": 5,
    "bodyHeatBase": 5,
    "safe": false
  }
]
```

8.2.8. ResearcherSkill

- **Leírás**

A Sarkkutató használja a képességét, megvizsgálja egy szomszédos mező teherbírását.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt vizsgáljuk, hogy a mező teherbírása megegyezik eazzal, amit a sarkkutató eredményül kapott.

Lehetséges hibák:

- A Researcher rossz értéket kap vissza a teherbírásról
- Nem kap semmilyen értéket

- **Bemenet**

```
load map.json
setfacing p_r f_0_0
skill p_r
list f_0_0
```

- Elvárt kimenet**

Eszkimo skill lefutott, a facing water capacity-je: -1

```
[  
 {  
   "uid": "f_0_0",  
   "type": "Ice",  
   "neighbours": [  
     {  
       "uid": "f_0_1"  
     },  
     {  
       "uid": "f_1_0"  
     }  
   ],  
   "frozenItems": [],  
   "floatingItems": [  
     {  
       "uid": "i_f_0"  
     }  
   ],  
   "snowlayers": 2,  
   "building": "NullBuilding",  
   "players": [  
     {  
       "uid": "p_i"  
     },  
     {  
       "uid": "p_e"  
     }  
   ],  
   "bears": []  
 }
```

8.2.9. StepOntoIce

- Leírás**

Egy Player egy jégtáblára lép.

- Ellenőrzött funkcionalitás, várható hibahelyek**

Azt vizsgáljuk, hogy a játékos az egyik mezőből eltávolításra került, míg a másikhoz hozzáadódott.

Lehetséges hibák:

- A játékos nem lesz eltávolítva a jégtábláról, amin állt
- A játékos el lett távolítva a jégtábláról, de nem lett hozzáadva az újhoz

- Bemenet**

```
load map.json  
setfacing p_r f_0_0  
move p_r  
list f_0_1 f_0_0
```

- **Elvárt kimenet**

```
[
{
  "uid": "f_0_1",
  "type": "UnstableIce",
  "neighbours": [
    {
      "uid": "f_1_1"
    },
    {
      "uid": "f_0_0"
    }
  ],
  "frozenItems": [
    {
      "uid": "i_f_1"
    },
    {
      "uid": "i_c_0"
    }
  ],
  "floatingItems": [],
  "snowlayers": 0,
  "building": "NullBuilding",
  "players": [
    {
      "uid": "p_s"
    }
  ],
  "bears": []
},
{
  "uid": "f_0_0",
  "type": "Ice",
  "neighbours": [
    {
      "uid": "f_0_1"
    },
    {
      "uid": "f_1_0"
    }
  ],
  "frozenItems": [],
  "floatingItems": [
    {
      "uid": "i_f_0"
    }
  ],
}
```

```

"snowlayers": 2,
"building": "NullBuilding",
"players": [
  {
    "uid": "p_i"
  },
  {
    "uid": "p_e"
  },
  {
    "uid": "p_r"
  }
],
"bears": []
}
]

```

8.2.10. StepOntoUnstableIce

- **Leírás**

Egy Player egy instabil jégtáblára lép.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Megvizsgáljuk, hogy ha egy új játékos lép az instabil jégtáblára - ezzel túllépve annak teherbírássát - akkor az beszakad-e.

Lehetséges hibák:

- Nem szakad be a jégtábla
- A beszakadt jégtábla nem cserélődik ki egy víz táblára
- Az játékosok nem kerülnek be a vízbe
-

- **Bemenet**

```

load map.json
setfacing p_e f_0_1
move p_e
list f_0_0 f_0_1

```

- **Elvárt kimenet**

```

[
  {
    "uid": "f_0_0",
    "type": "Ice",
    "neighbours": [
      {
        "uid": "f_0_1"
      },
      {
        "uid": "f_1_0"
      }
    ]
  }
]
```

```
],
  "frozenItems": [],
  "floatingItems": [
    {
      "uid": "i_f_0"
    }
  ],
  "snowlayers": 2,
  "building": "NullBuilding",
  "players": [
    {
      "uid": "p_i"
    }
  ],
  "bears": []
},
{
  "uid": "f_0_1",
  "type": "UnstableIce",
  "neighbours": [
    {
      "uid": "f_1_1"
    },
    {
      "uid": "f_0_0"
    }
  ],
  "frozenItems": [
    {
      "uid": "i_f_1"
    },
    {
      "uid": "i_c_0"
    }
  ],
  "floatingItems": [],
  "snowlayers": 0,
  "building": "NullBuilding",
  "players": [
    {
      "uid": "p_s"
    },
    {
      "uid": "p_r"
    },
    {
      "uid": "p_e"
    }
  ],
  "bears": []
}
```

```

    }
]
```

8.2.11. StepOntoWater

- **Leírás**

Egy Player víz mezőre lép.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt vizsgáljuk, hogy a játékos az egyik mezőből eltávolításra kerül, míg a másikhoz hozzáadódik.

Lehetséges hibák:

- A játékos nem lesz eltávolítva a jégtábláról amin állt
- A játékos el lett távolítva a jégtábláról de nem lett hozzáadva a vízhez

- **Bemenet**

```
load map.json
setfacing p_r f_1_1
move p_r
list f_0_1 f_1_1
```

- **Elvárt kimenet**

```
[
  {
    "uid": "f_0_1",
    "type": "UnstableIce",
    "neighbours": [
      {
        "uid": "f_1_1"
      },
      {
        "uid": "f_0_0"
      }
    ],
    "frozenItems": [
      {
        "uid": "i_f_1"
      },
      {
        "uid": "i_c_0"
      }
    ],
    "floatingItems": [],
    "snowlayers": 0,
    "building": "NullBuilding",
    "players": [
      {
        "uid": "p_s"
      }
    ],
  ]
```

```

    "bears": []
},
{
  "uid": "f_1_1",
  "type": "UnstableIce",
  "neighbours": [
    {
      "uid": "f_1_0"
    },
    {
      "uid": "f_0_1"
    }
  ],
  "frozenItems": [],
  "floatingItems": [],
  "snowlayers": 1,
  "building": "NullBuilding",
  "players": [
    {
      "uid": "p_r"
    }
  ],
  "bears": []
}
]

```

8.2.12 StepOntoFieldWhitBear

- **Leírás**

A Player olyan jégre lép, amin Bear van.

- **Ellenőrzött funkcionális, várható hibahelyek**

Azt vizsgáljuk, hogy ha egy játékos egy olyan mezőre lép amin egy medve van akkor a medve megtámadja-e a játékost.

Lehetséges hibák:

- A medve nem támadja meg a játékost és nem lesz vége a játéknak
- A medve megtámadja a játékos de nem lesz vége a játéknak

- **Bemenet**

```

load map.json
addbear b_1 f_1_1
setfacing p_r f_1_1
move p_r
list f_0_1 f_1_1

```

- **Elvárt kimenet**

```

[
  {
    "uid": "f_0_1",
    "type": "UnstableIce",
    "neighbours": [
      {
        "uid": "f_1_1"
      }
    ],
    "frozenItems": [],
    "floatingItems": [],
    "snowlayers": 1,
    "building": "NullBuilding",
    "players": [
      {
        "uid": "p_r"
      }
    ],
    "bears": []
  }
]

```

```
"neighbours": [
  {
    "uid": "f_1_1"
  },
  {
    "uid": "f_0_0"
  }
],
"frozenItems": [
  {
    "uid": "i_f_1"
  },
  {
    "uid": "i_c_0"
  }
],
"floatingItems": [],
"snowlayers": 0,
"building": "NullBuilding",
"players": [
  {
    "uid": "p_s"
  }
],
"bears": []
},
{
  "uid": "f_1_1",
  "type": "UnstableIce",
  "neighbours": [
    {
      "uid": "f_1_0"
    },
    {
      "uid": "f_0_1"
    }
  ],
  "frozenItems": [],
  "floatingItems": [],
  "snowlayers": 1,
  "building": "NullBuilding",
  "players": [
    {
      "uid": "p_r"
    }
  ],
  "bears": [
    {
      "uid": "b_1"
    }
  ]
}
```

```

        ]
    }
]
LOSE!
```

8.2.13. UseComponentWith1Components

- **Leírás**

Egy Player megkísérli összeépíteni a FlareGun-t, de nem jön össze.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Egy játékos akinél kevesebb, mint 3 komponens van, használja az egyiket, a létrehozott FlareGun check attribútuma nem éri el a 3 értéket és a játéknak nincs vége.

Lehetséges hibák:

- A játéknak vége lesz akkor is, ha nincs 3 komponense a játékosnak

- **Bemenet**

```
load map.json
additem component i_c_2 p_r
useitem i_c_2 p_r
list p_r
```

- **Elvárt kimenet**

```
[
  {
    "uid": "p_r",
    "type": "Researcher",
    "field": "f_0_1",
    "facing": "f_1_1",
    "items": [
      {
        "uid": "i_c_2"
      }
    ],
    "bodyHeat": 3,
    "bodyHeatBase": 4,
    "safe": false
  }
]
```

8.2.14. UseComponentWith2Components

- **Leírás**

Egy Player megkísérli összeépíteni a FlareGun-t, de nem jön össze.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Egy játékos akinél kevesebb, mint 3 komponens van, használja az egyiket, a létrehozott FlareGun check attribútuma nem éri el a 3 értéket és a játéknak nincs vége.

Lehetséges hibák:

- A játéknak vége lesz akkor is, ha nincs 3 komponense a játékosnak

- **Bemenet**

```
load map.json
additem component i_c_1 p_r
additem component i_c_2 p_r
useitem i_c_1 p_r
list p_r
```

- **Elvárt kimenet**

```
[ {
  "uid": "p_r",
  "type": "Researcher",
  "field": "f_0_1",
  "facing": "f_1_1",
  "items": [
    {
      "uid": "i_c_1"
    },
    {
      "uid": "i_c_2"
    }
  ],
  "bodyHeat": 3,
  "bodyHeatBase": 4,
  "safe": false
}
```

8.2.15. UseComponentWith3Components

- **Leírás**

Egy Player összeszereli a FlareGun-t.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Component által létrehozott FlareGun check attribútuma eléri a 3 értéket, majd meghívja a GameManager Win() metódusát, ezzel vége a játéknak.

Lehetséges hibák:

- A játéknak nincs vége

- **Bemenet**

```
load map.json
additem component i_c_1 p_r
additem component i_c_2 p_r
additem component i_c_3 p_r
```

useitem i_c_1 p_r
list p_r

- **Elvárt kimenet**

```
[  
 {  
   "uid": "p_r",  
   "type": "Researcher",  
   "field": "f_0_1",  
   "facing": "f_1_1",  
   "items": [  
     {  
       "uid": "i_c_1"  
     },  
     {  
       "uid": "i_c_2"  
     },  
     {  
       "uid": "i_c_3"  
     }  
   ],  
   "bodyHeat": 3,  
   "bodyHeatBase": 4,  
   "safe": false  
 }  
 ]
```

WIN!

8.2.16. UseFood

- **Leírás**

Egy Player elhasznál egy Food-ot (úgy, hogy a testhője nem maximális).

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos testhője 1-el nő. A Food-nak el kell tűnnie a játékos tárgykészletéből.

Lehetséges hibák:

- A játékos testhője nem 1-el nőtt
- A Food nem tűnt el a játékostól

- **Bemenet**

```
load map.json  
additem food i_f_2 p_r  
useitem i_f_2 p_r  
list p_r
```

- **Elvárt kimenet**

```
[  
 {  
   "uid": "p_r",  
   "type": "Researcher",  
   "field": "f_0_1",  
   "facing": "f_1_1",  
   "items": [  
     {  
       "uid": "i_f_2"  
     }  
   ],  
   "bodyHeat": 2,  
   "bodyHeatBase": 4,  
   "safe": false  
 }  
 ]
```

```

    "field": "f_0_1",
    "facing": "f_1_1",
    "items": [],
    "bodyHeat": 4,
    "bodyHeatBase": 4,
    "safe": false
}
]

```

8.2.17. UseFoodOnMaxBodyHeat

- **Leírás**

Egy Player maximális testhőre használja a Food-ot.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos testhője nem nőhet. A Food-nak el kell tűnnie a játékos tárgykészletéből.

Lehetséges hibák:

- A játékos testhője nőtt, ezzel átlépve a maximumot
- A Food nem tűnt el a játékostól

- **Bemenet**

```

load map.json
additem food i_f_2 p_s
useitem i_f_2 p_s
list p_s

```

- **Elvárt kimenet**

```

[
  {
    "uid": "p_s",
    "type": "Seal",
    "field": "f_0_1",
    "facing": "f_1_1",
    "items": [],
    "bodyHeat": 6,
    "bodyHeatBase": 6,
    "safe": true
  }
]

```

8.2.18. UseShovelToDigSnowLayer

- **Leírás**

Egy Player használja a Shovel-t olyan jégen, ahol még van höréteg.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Shovel durability-je 1-el csökken. A jégtábla snowLayers attribútuma 2-vel csökkent. Ezek után a snowLayers attribútum értéke nem lehet kevesebb, mint 0!

Lelhetőséges hibák:

- A Shovel durability-je nem 1-el csökkent
- A snowLayers értéke nem 2-vel csökkent úgy, hogy értéke korábban ≥ 2 volt
- A snowLayers értéke nem lett 0 úgy, hogy értéke korábban 1 volt
- Egy Item ki lett ásva

- **Bemenet**

```
load map.json
additem shovel i_s_0 p_i
useitem i_s_0 p_i
list f_0_0 p_i
```

- **Elvárt kimenet**

```
[
  {
    "uid": "f_0_0",
    "type": "Ice",
    "neighbours": [
      {
        "uid": "f_0_1"
      },
      {
        "uid": "f_1_0"
      }
    ],
    "frozenItems": [],
    "floatingItems": [
      {
        "uid": "i_f_0"
      }
    ],
    "snowlayers": 0,
    "building": "NullBuilding",
    "players": [
      {
        "uid": "p_i"
      },
      {
        "uid": "p_e"
      }
    ],
    "bears": []
  },
  {
    "uid": "p_i",
    "type": "ITMan",
    "field": "f_0_0",
```

```

    "facing": "f_0_1",
    "items": [
        {
            "uid": "i_s_0"
        }
    ],
    "bodyHeat": 3,
    "bodyHeatBase": 3,
    "safe": false
}
]

```

8.2.19. UseShovelToDigItem

- **Leírás**

Egy Player használja a Shovel-t olyan jégen, ahol nincs hörétek, és kiás egy Item-et.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Shovel durability-je 1-el csökken. A jégtábla frozenItems listájának legfelső eleme a floatingItems listába kerül.

Lehetséges hibák:

- A Shovel durability-je nem 1-el csökkent
- A kiásott Item nem a frozenItems lista első eleme
- Nem lett Item kiásva (és a frozenItems lista nem üres)
- A kiásott Item nem került bele a floatingItems listába

- **Bemenet**

```

load map.json
additem shovel i_s_0 p_r
useitem i_s_0 p_r
list f_0_1 p_r

```

- **Elvárt kimenet**

```

[
    {
        "uid": "f_0_1",
        "type": "UnstableIce",
        "neighbours": [
            {
                "uid": "f_1_1"
            },
            {
                "uid": "f_0_0"
            }
        ],
        "frozenItems": [
            {
                "uid": "i_c_0"
            }
        ]
    }
]

```

```

        },
        ],
        "floatingItems": [
        {
            "uid": "i_f_1"
        }
        ],
        "snowlayers": 0,
        "building": "NullBuilding",
        "players": [
        {
            "uid": "p_s"
        },
        {
            "uid": "p_r"
        }
        ],
        "bears": []
    },
    {
        "uid": "p_r",
        "type": "Researcher",
        "field": "f_0_1",
        "facing": "f_1_1",
        "items": [
        {
            "uid": "i_s_0"
        }
        ],
        "bodyHeat": 3,
        "bodyHeatBase": 4,
        "safe": false
    }
]

```

8.2.20. UseRope

- Leírás**

Egy Player használja a Rope-ot és a saját jégtáblájára mozgatja az eltalált játékosokat.

- Ellenőrzött funkcionalitás, várható hibahelyek**

A játékostól eltűnik a Rope. A Rope az eltalált játékosokat átmozgatja a használó játékos jégtáblájára.

Lehetséges hibák:

- Az eltalált játékosok nem mozdulnak
- Az eltalált játékosok helytelen jégtáblára kerülnek
- Nem az összes játékos lett elmozgatva az eltalált jégtábláról
- A játékostól nem tünt el a Rope

● **Bemenet**

```
load map.json
additem rope i_r_0 p_i
setfacing p_i f_0_1
useitem i_r_0 p_i
list f_0_0 f_0_1 f_1_0 f_1_1 p_i
```

● **Elvárt kimenet**

```
[
  {
    "uid": "f_0_0",
    "type": "Ice",
    "neighbours": [
      {
        "uid": "f_0_1"
      },
      {
        "uid": "f_1_0"
      }
    ],
    "frozenItems": [],
    "floatingItems": [
      {
        "uid": "i_f_0"
      }
    ],
    "snowlayers": 2,
    "building": "NullBuilding",
    "players": [
      {
        "uid": "p_i"
      },
      {
        "uid": "p_e"
      }
    ],
    "bears": []
  },
  {
    "uid": "f_0_1",
    "type": "UnstableIce",
    "neighbours": [
      {
        "uid": "f_1_1"
      },
      {
        "uid": "f_0_0"
      }
    ],
    "frozenItems": [
      {
        "uid": "i_f_1"
      }
    ],
    "floatingItems": [
      {
        "uid": "i_f_0"
      }
    ],
    "snowlayers": 1,
    "building": "NullBuilding",
    "players": [
      {
        "uid": "p_e"
      }
    ],
    "bears": []
  }
]
```

```

"frozenItems": [
  {
    "uid": "i_f_1"
  },
  {
    "uid": "i_c_0"
  }
],
"floatingItems": [],
"snowlayers": 0,
"building": "NullBuilding",
"players": [
  {
    "uid": "p_s"
  },
  {
    "uid": "p_r"
  }
],
"bears": []
},
{
  "uid": "f_1_0",
  "type": "UnstableIce",
  "neighbours": [
    {
      "uid": "f_0_0"
    },
    {
      "uid": "f_1_1"
    }
  ],
  "frozenItems": [],
  "floatingItems": [],
  "snowlayers": 0,
  "building": "NullBuilding",
  "players": [],
  "bears": []
},
{
  "uid": "f_1_1",
  "type": "UnstableIce",
  "neighbours": [
    {
      "uid": "f_1_0"
    },
    {
      "uid": "f_0_1"
    }
  ],

```

```

    "frozenItems": [],
    "floatingItems": [],
    "snowlayers": 1,
    "building": "NullBuilding",
    "players": [],
    "bears": []
},
{
    "uid": "p_i",
    "type": "ITMan",
    "field": "f_0_0",
    "facing": "f_0_1",
    "items": [
        {
            "uid": "i_p_0"
        }
    ],
    "bodyHeat": 3,
    "bodyHeatBase": 3,
    "safe": false
}
]

```

8.2.21. UseRopeInWrongDirection

- **Leírás**

Egy Player rossz irányban használja a Rope-ot, ezzel nem ment ki senkit.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékosról eltűnik a Rope. A Rope nem mozgat át senkit másik mezőre.

Lehetséges hibák:

- A Rope hatására egyes Player-ek átmozogtak másik mezőre
- A játékosról nem tünt el a Rope

- **Bemenet**

```

load map.json
additem rope i_r_0 p_i
setfacing p_i f_1_0
useitem i_r_0 p_i
list f_0_0 f_1_0 f_0_1 f_1_1 p_i

```

- **Elvárt kimenet**

```

[
{
    "uid": "f_0_0",
    "type": "Ice",
    "neighbours": [
        {
            "uid": "f_0_1"
        },
        {

```

```
        "uid": "f_1_0"
    },
],
"frozenItems": [],
"floatingItems": [
{
    "uid": "i_f_0"
},
],
"snowlayers": 2,
"building": "NullBuilding",
"players": [
{
    "uid": "p_i"
},
{
    "uid": "p_e"
}
],
"bears": []
},
{
    "uid": "f_1_0",
    "type": "UnstableIce",
    "neighbours": [
{
    "uid": "f_0_0"
},
{
    "uid": "f_1_1"
}
],
"frozenItems": [],
"floatingItems": [],
"snowlayers": 0,
"building": "NullBuilding",
"players": [],
"bears": []
},
{
    "uid": "f_0_1",
    "type": "UnstableIce",
    "neighbours": [
{
    "uid": "f_1_1"
},
{
    "uid": "f_0_0"
}
],
```

```

"frozenItems": [
  {
    "uid": "i_f_1"
  },
  {
    "uid": "i_c_0"
  }
],
"floatingItems": [],
"snowlayers": 0,
"building": "NullBuilding",
"players": [
  {
    "uid": "p_s"
  },
  {
    "uid": "p_r"
  }
],
"bears": []
},
{
  "uid": "f_1_1",
  "type": "UnstableIce",
  "neighbours": [
    {
      "uid": "f_1_0"
    },
    {
      "uid": "f_0_1"
    }
  ],
  "frozenItems": [],
  "floatingItems": [],
  "snowlayers": 1,
  "building": "NullBuilding",
  "players": [],
  "bears": []
},
{
  "uid": "p_i",
  "type": "ITMan",
  "field": "f_0_0",
  "facing": "f_1_0",
  "items": [],
  "bodyHeat": 3,
  "bodyHeatBase": 3,
  "safe": false
}
]

```

8.2.22. UseDivingSuit

- **Leírás**

Egy Player használja a DivingSuit-ot.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Használat után a safe attribútum True kell hogy legyen. A DivingSuit-nak el kell tűnnie a játékos tárgykészletéből.

Lehetséges hibák:

- A safe attribútum False értékű
- A játékostól nem tűnt el a DivingSuit

- **Bemenet**

```
load map.json
additem divingsuit i_d_0 p_r
useitem i_d_0 p_r
list p_r
```

- **Elvárt kimenet**

```
[
  {
    "uid": "p_r",
    "type": "Researcher",
    "field": "f_0_1",
    "facing": "f_1_1",
    "items": [],
    "bodyHeat": 3,
    "bodyHeatBase": 4,
    "safe": true
  }
]
```

8.2.23. UseDivingSuitWhitSafeAttributeOn

- **Leírás**

Egy Player használja a DivingSuit-ot úgy, hogy a safe attribútuma True.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A safe attribútum továbbra is True kell hogy legyen. A DivingSuit-nak el kell tűnnie a játékos tárgykészletéből.

Lehetséges hibák:

- A safe attribútum False-ra változott
- A játékostól nem tűnt el a DivingSuit

- **Bemenet**

```
load map.json
additem divingsuit i_d_0 p_s
useitem i_d_0 p_s
list p_s
```

- **Elvárt kimenet**

```
[
  {
    "uid": "p_s",
    "type": "Seal",
    "field": "f_0_1",
    "facing": "f_1_1",
    "items": [],
    "bodyHeat": 6,
    "bodyHeatBase": 6,
    "safe": true
  }
]
```

8.2.24. UsePackedTent

- **Leírás**

Egy Player használja az PackedTent-et és lerak egy Tent-et a saját jégtáblájára.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A jégen lévő Iglu-nak Tent-re kell cserélődni. A PackedTent-nek el kell tűnnie a játékos tárgykészletéből.

Lehetséges hibák:

- Nem jött létre Tent a jégen
- A játékostól nem tűnt el a PackedTent

- **Bemenet**

```
load map.json
additem packedtent i_pt_0 p_e
setfacing p_e f_1_0
move p_e
useitem i_pt_0 p_e
list f_1_0 p_e
```

- **Elvárt kimenet**

```
[
  {
    "uid": "f_1_0",
    "type": "UnstableIce",
    "neighbours": [
      {
        "uid": "f_0_0"
      },
      {
        "uid": "f_1_1"
      }
    ],
    "frozenItems": [],
    "floatingItems": [],
    "snowlayers": 0,
  }
]
```

```

"building": "Tent",
"players": [
{
  "uid": "p_e"
},
],
"bears": []
},
{
  "uid": "p_e",
  "type": "Eskimo",
  "field": "f_1_0",
  "facing": "f_0_0",
  "items": [],
  "bodyHeat": 0,
  "bodyHeatBase": 5,
  "safe": false
}
]

```

8.2.25. UsePackedTentOnIceWhitIglu

- Leírás**

A Player PackedTent-et használ olyan jégen, amin Iglu van.

- Ellenőrzött funkcionalitás, várható hibahelyek**

A jégen lévő Iglu-nak Tent-re kell cserélődni. A PackedTent-nek el kell tünnie a játékos tárgykészletéből.

Lehetséges hibák:

- Az Iglu nem cserélődött le Tent-re
- Az Iglu eltűnt, de nem lett Tent a helyén
- A játékostól nem tünt el a PackedTent

- Bemenet**

```

load map.json
additem packedtent i_pt_0 p_r
useitem i_pt_0 p_r
list f_0_1 p_r

```

- Elvárt kimenet**

```

[
{
  "uid": "f_0_1",
  "type": "UnstableIce",
  "neighbours": [
    {
      "uid": "f_1_1"
    }
  ]
}
]
```

```

    },
    {
      "uid": "f_0_0"
    }
  ],
  "frozenItems": [
    {
      "uid": "i_f_1"
    },
    {
      "uid": "i_c_0"
    }
  ],
  "floatingItems": [],
  "snowlayers": 0,
  "building": "Tent",
  "players": [
    {
      "uid": "p_s"
    },
    {
      "uid": "p_r"
    }
  ],
  "bears": []
},
{
  "uid": "p_r",
  "type": "Researcher",
  "field": "f_0_1",
  "facing": "f_1_1",
  "items": [],
  "bodyHeat": 3,
  "bodyHeatBase": 4,
  "safe": false
}
]

```

8.2.26. BearMoves

- **Leírás**

Egy Bear egy olyan szomszédos mezőre lép, ahol nem áll Player.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játéknak tovább kell haladnia, mivel a medve nem támadott meg senkit

Lehetséges hibák:

- A játék befejeződött

- **Bemenet**

load map.json

```
addbear b_0 f_1_0
setfacing b_0 f_1_1
move b_0
list f_1_0 f_1_1
```

• **Elvárt kimenet**

```
[
  {
    "uid": "f_1_0",
    "type": "UnstableIce",
    "neighbours": [
      {
        "uid": "f_0_0"
      },
      {
        "uid": "f_1_1"
      }
    ],
    "frozenItems": [],
    "floatingItems": [],
    "snowlayers": 0,
    "building": "NullBuilding",
    "players": [],
    "bears": []
  },
  {
    "uid": "f_1_1",
    "type": "UnstableIce",
    "neighbours": [
      {
        "uid": "f_1_0"
      },
      {
        "uid": "f_0_1"
      }
    ],
    "frozenItems": [],
    "floatingItems": [],
    "snowlayers": 1,
    "building": "NullBuilding",
    "players": [],
    "bears": [
      {
        "uid": "b_0"
      }
    ]
  }
]
```

8.2.27. BearMoveAndAttacks

- **Leírás**

Egy Bear olyan szomszédos mezőre lép, ahol Player áll, és megtámadja azt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játéknak be kell fejeződnie.

Lehetséges hibák:

- A játéknak nincs vége

- **Bemenet**

```
load map.json
addbear b_0 f_1_0
setfacing b_0 f_0_0
move b_0
list f_1_0 f_0_0
```

- **Elvárt kimenet**

```
[
  {
    "uid": "f_1_0",
    "type": "UnstableIce",
    "neighbours": [
      {
        "uid": "f_0_0"
      },
      {
        "uid": "f_1_1"
      }
    ],
    "frozenItems": [],
    "floatingItems": [],
    "snowlayers": 0,
    "building": "NullBuilding",
    "players": [],
    "bears": []
  },
  {
    "uid": "f_0_0",
    "type": "Ice",
    "neighbours": [
      {
        "uid": "f_0_1"
      },
      {
        "uid": "f_1_0"
      }
    ],
    "frozenItems": []
```

```

"floatingItems": [
  {
    "uid": "i_f_0"
  }
],
"snowlayers": 2,
"building": "NullBuilding",
"players": [
  {
    "uid": "p_i"
  },
  {
    "uid": "p_e"
  }
],
"bears": [
  {
    "uid": "b_0"
  }
]
}
LOSE!

```

8.2.28. StormEffectOnIceWhitIglu

- **Leírás**

Vihar hatása jégen, amin Iglu van.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A jégen, amin Iglu van, a játékosok testhője nem csökkenhet.

Lehetséges hibák:

- A játékosok teshője csökken

- **Bemenet**

```

load map.json
createstorm f_0_1
list f_0_0 p_i p_e f_0_1 p_s p_r f_1_1

```

- **Elvárt kimenet**

```

[
  {
    "uid": "f_0_0",
    "type": "Ice",
    "neighbours": [
      {
        "uid": "f_0_1"
      },
      {
        "uid": "f_1_0"
      }
    ]
  }
]

```

```

        }
    ],
    "frozenItems": [],
    "floatingItems": [
        {
            "uid": "i_f_0"
        }
    ],
    "snowlayers": 3,
    "building": "NullBuilding",
    "players": [
        {
            "uid": "p_i"
        },
        {
            "uid": "p_e"
        }
    ],
    "bears": []
},
{
    "uid": "p_i",
    "type": "ITMan",
    "field": "f_0_0",
    "facing": "f_0_1",
    "items": [],
    "bodyHeat": 2,
    "bodyHeatBase": 3,
    "safe": false
},
{
    "uid": "p_e",
    "type": "Eskimo",
    "field": "f_0_0",
    "facing": "f_0_1",
    "items": [],
    "bodyHeat": 0,
    "bodyHeatBase": 5,
    "safe": false
},
{
    "uid": "f_0_1",
    "type": "UnstableIce",
    "neighbours": [
        {
            "uid": "f_1_1"
        },
        {
            "uid": "f_0_0"
        }
    ]
}

```

```
],
  "frozenItems": [
    {
      "uid": "i_f_1"
    },
    {
      "uid": "i_c_0"
    }
  ],
  "floatingItems": [],
  "snowlayers": 1,
  "building": "NullBuilding",
  "players": [
    {
      "uid": "p_s"
    },
    {
      "uid": "p_r"
    }
  ],
  "bears": []
},
{
  "uid": "p_s",
  "type": "Seal",
  "field": "f_0_1",
  "facing": "f_1_1",
  "items": [],
  "bodyHeat": 5,
  "bodyHeatBase": 6,
  "safe": true
},
{
  "uid": "p_r",
  "type": "Researcher",
  "field": "f_0_1",
  "facing": "f_1_1",
  "items": [],
  "bodyHeat": 2,
  "bodyHeatBase": 4,
  "safe": false
},
{
  "uid": "f_1_1",
  "type": "UnstableIce",
  "neighbours": [
    {
      "uid": "f_1_0"
    },
    {
      "uid": "f_1_2"
    }
  ]
}
```

```

        "uid": "f_0_1"
    },
],
"frozenItems": [],
"floatingItems": [],
"snowlayers": 2,
"building": "NullBuilding",
"players": [],
"bears": []
}
]

```

8.2.29. StormEffectOnIceWhitTent

- **Leírás**

Vihar hatása jégen, amin Tent van.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A jégen, amin Tent van, a játékosok testhője nem csökkenhet.

Lehetséges hibák:

- A játékosok teshője csökken

- **Bemenet**

load map.json

createstorm f_0_0

list f_0_0 p_i p_e f_0_1 p_s p_r f_1_0

- **Elvárt kimenet**

```
[
{
    "uid": "f_0_0",
    "type": "Ice",
    "neighbours": [
        {
            "uid": "f_0_1"
        },
        {
            "uid": "f_1_0"
        }
    ],
    "frozenItems": [],
    "floatingItems": [
        {
            "uid": "i_f_0"
        }
    ],
    "snowlayers": 3,
    "building": "NullBuilding",
    "players": [
        {

```

```

    "uid": "p_i"
  },
  {
    "uid": "p_e"
  }
],
"bears": []
},
{
  "uid": "p_i",
  "type": "ITMan",
  "field": "f_0_0",
  "facing": "f_0_1",
  "items": [],
  "bodyHeat": 2,
  "bodyHeatBase": 3,
  "safe": false
},
{
  "uid": "p_e",
  "type": "Eskimo",
  "field": "f_0_0",
  "facing": "f_0_1",
  "items": [],
  "bodyHeat": 0,
  "bodyHeatBase": 5,
  "safe": false
},
{
  "uid": "f_0_1",
  "type": "UnstableIce",
  "neighbours": [
    {
      "uid": "f_1_1"
    },
    {
      "uid": "f_0_0"
    }
  ],
  "frozenItems": [
    {
      "uid": "i_f_1"
    },
    {
      "uid": "i_c_0"
    }
  ],
  "floatingItems": [],
  "snowlayers": 1,
  "building": "NullBuilding",

```

```

"players": [
  {
    "uid": "p_s"
  },
  {
    "uid": "p_r"
  }
],
"bears": []
},
{
  "uid": "p_s",
  "type": "Seal",
  "field": "f_0_1",
  "facing": "f_1_1",
  "items": [],
  "bodyHeat": 5,
  "bodyHeatBase": 6,
  "safe": true
},
{
  "uid": "p_r",
  "type": "Researcher",
  "field": "f_0_1",
  "facing": "f_1_1",
  "items": [],
  "bodyHeat": 2,
  "bodyHeatBase": 4,
  "safe": false
},
{
  "uid": "f_1_0",
  "type": "UnstableIce",
  "neighbours": [
    {
      "uid": "f_0_0"
    },
    {
      "uid": "f_1_1"
    }
  ],
  "frozenItems": [],
  "floatingItems": [],
  "snowlayers": 1,
  "building": "NullBuilding",
  "players": [],
  "bears": []
}
]

```

8.2.30. StromEffectOnIceWithNullBuilding

- **Leírás**

Vihar hatása jégen, amin NullBuilding (~levegő) van.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A jégen, amin NullBuilding van, a játékosok testhője 1-el csökken.

Lehetséges hibák:

- A játékosok teshője nem csökken
- A játékosok testhője nem 1-el, de csökken

- **Bemenet**

```
load map.json
setfacing p_i f_1_0
move p_i
createstorm f_1_0
list f_1_0 p_i
```

- **Elvárt kimenet**

```
[
  {
    "uid": "f_1_0",
    "type": "UnstableIce",
    "neighbours": [
      {
        "uid": "f_0_0"
      },
      {
        "uid": "f_1_1"
      }
    ],
    "frozenItems": [],
    "floatingItems": [],
    "snowlayers": 1,
    "building": "NullBuilding",
    "players": [
      {
        "uid": "p_i"
      }
    ],
    "bears": []
  },
  {
    "uid": "p_i",
    "type": "ITMan",
    "field": "f_1_0",
    "facing": "f_0_0",
    "items": [],
    "bodyHeat": 2,
    "bodyHeatBase": 3,
    "safe": false
  }
]
```

```

    }
]
```

8.2.31. GameOver

- **Leírás**

A GameManager ellenőrzi, hogy vége lett-e a játéknak úgy, hogy nem teljesül egy feltétel sem az alábbiakból:

- Van olyan játékos, akinek testhője 0
- Van olyan játékos, aki vízben van és safe attribútuma False

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játéknak nem szabad befejeződni

Lehetséges hibák:

- A játék befejeződik

- **Bemenet**

```
load map.json
additem food i_f_2 p_e
useitem i_f_2 p_e
list f_0_0 p_e
losegame
```

- **Elvárt kimenet**

```
[
  {
    "uid": "f_0_0",
    "type": "Ice",
    "neighbours": [
      {
        "uid": "f_0_1"
      },
      {
        "uid": "f_1_0"
      }
    ],
    "frozenItems": [],
    "floatingItems": [
      {
        "uid": "i_f_0"
      }
    ],
    "snowlayers": 2,
    "building": "NullBuilding",
    "players": [
      {
        "uid": "p_i"
      },
      {
        "uid": "p_e"
      }
    ]
  }
]
```

```

        }
    ],
    "bears": []
},
{
    "uid": "p_e",
    "type": "Eskimo",
    "field": "f_0_0",
    "facing": "f_0_1",
    "items": [],
    "bodyHeat": 1,
    "bodyHeatBase": 5,
    "safe": false
}
]

```

8.2.32. GameOverWithPlayerOnWaterWhitSafeAttributeFalse

- **Leírás**

A GameManager ellenőrzi, hogy vége lett-e a játéknak úgy, hogy egy játékos vízben van és a safe attribútuma False értékű.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játéknak véget kell érnie, mivel teljesül a feltétel miszerint egy false safe attribútumú játékos vízben fejezte be a kört.

Lehetséges hibák:

- A játéknak nincs vége

- **Bemenet**

```

load map.json
setfacing p_r f_1_1
move p_r
list f_1_1 p_r
losegame

```

- **Elvárt kimenet**

```

[
{
    "uid": "f_1_1",
    "type": "UnstableIce",
    "neighbours": [
        {
            "uid": "f_1_0"
        },
        {
            "uid": "f_0_1"
        }
    ],
    "frozenItems": [],
    "floatingItems": []
}
]
```

```

"snowlayers": 1,
"building": "NullBuilding",
"players": [
  {
    "uid": "p_r"
  }
],
"bears": []
},
{
  "uid": "p_r",
  "type": "Researcher",
  "field": "f_1_1",
  "facing": "f_1_0",
  "items": [],
  "bodyHeat": 3,
  "bodyHeatBase": 4,
  "safe": false
}
]
LOSE!

```

8.2.33. GameOverWithPlayerWithZeroBodyHeat

- Leírás**

A GameManager ellenőrzi, hogy vége lett-e a játéknak úgy, hogy egy játékosnak 0 testhője van.

- Ellenőrzött funkcionalitás, várható hibahelyek**

A játéknak véget kell érnie, mivel teljesül a feltétel miszerint egy játékos testhője elérte a 0-t.

Lehetséges hibák:

- A játéknak nincs vége

- Bemenet**

load map.json

list f_0_0 p_e

losegame

- Elvárt kimenet**

```
[
  {
    "uid": "f_0_0",
    "type": "Ice",
    "neighbours": [
      {
        "uid": "f_0_1"
      },
      {
        "uid": "f_1_0"
      }
    ]
  }
]
```

```

],
"frozenItems": [],
"floatingItems": [
{
  "uid": "i_f_0"
},
],
"snowlayers": 2,
"building": "NullBuilding",
"players": [
{
  "uid": "p_i"
},
{
  "uid": "p_e"
}
],
"bears": []
},
{
  "uid": "p_e",
  "type": "Eskimo",
  "field": "f_0_0",
  "facing": "f_0_1",
  "items": [],
  "bodyHeat": 0,
  "bodyHeatBase": 5,
  "safe": false
}
]

```

LOSE!

8.3. A tesztelést támogató programok tervezetek

8.3.1 Fő-tesztelő program

- **Rövid leírás**

A master script egy összefogló segédprogram, amely a többi segéd program segítségével teszteteket futtat, kiértékel, majd minden összegezve kiírja az eredményt a console-ra.

- **Paraméterek**

.\master.ps1 <case>

case: A tesztet neve, amit a felhasználó tesztelni szeretne. E paraméter nélkül a master script az összes tesztet futtatja.

- **Működés**

A script a paraméterül kapott érték alapján futtatja a tesztesetet. minden tesztesethez tartozik egy a tesztesettel azonos nevű <case>.input és egy <case>.ref kiterjesztésű fájl.

A master script a proto programot hívja megfelelően paramétere zve. Ez esetben a tesztesethez tartozó .\input\<case>.input fájllal.

A proto futása alapján, ha az nem állt meg hiba miatt, hívja a comparator.jar segédprogramot, mely az előállított .\temp.ref-et összehasonlítja az elvárt kimenettel, ami a .\ref\<case>.ref állományban található, majd a teszt eredményét a képernyőn közli a felhasználóval. Ha egy teszteset sikertelen, akkor annak okaival együtt.

Az egyes segédprogramok (ideérte a protot is) más-más exit code-al állnak le, így jelezve ha valami hiba történt. (sikeress futás exit kódja : 0)

Ha a proto futása közben rossz input fájlt kap, tehát értelmezhetetlen parancsot, akkor hibakódval áll le. A hibát a temp.ref ideiglenes fájlba írja. Ezt tudja kiolvasni a master script majd közli a felhasználóval. Ha a comparator szerint hibás a kimenet, exit code-al áll le. A hibát a temp.ref fájlba írja.

Ha a master script paraméter nélkül fut az input és ref könyvtárak alapján tartalma alapján futtatja az összes tesztesetet. Majd összegezve kiírja a tesztesetek futásának eredményét. (összes teszteset / sikeres teszteset)

- **Fájlok/könyvtárak**

master.ps1 - futtatható script

temp.ref - ideiglenes tároló a proto kimenetére

teszteset.input - egy teszt esethez tartozó parancsok összessége

teszteset.ref - egy teszteset futása után az elvárt kimenet

A tesztesetek inputja és az elvár kimenetek külön könyvtárban vannak tárolva ("input", "ref" könyvtárak)

8.4. Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.04.08. 10:00	3,5 óra	Bodó Gaál Varga Nagy Rédai	Értekezlet. Osztályok leírása.
2020.04.09. 16:00	3,5 óra	Bodó Gaál Varga Nagy Rédai	Osztályok írásának folytatása. Tesztesetek leírásának elkezdése.
2020.04.10. 16:00	3 óra	Bodó Gaál Varga Nagy Rédai	Tesztesetek leírásának folytatása.
2020.04.11. 16:00	4,5 óra	Bodó Gaál Varga Nagy Rédai	Tesztesetek leírásának befejezése. A tesztelést támogató programok tervének elkezdése.
2020.04.12. 10:00	1,5 óra	Bodó Gaál Varga Nagy Rédai	A tesztelést támogató programok tervének befejezése.
2020.04.12. 16:00	4 óra	Bodó Gaál Varga Nagy Rédai	Dokumentáció véglegesítése, átnézése.

10. Prototípus beadása

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

.java

Fájl neve	Méret (Byte)	Keletkezés ideje	Tartalom
Bear.java	1455	2020.04.24. 14:19	A medve osztálya
BearEffect.java	564	2020.04.24. 14:19	A medve effektus osztálya
Building.java	1493	2020.04.24. 14:19	Az épület osztálya
Component.java	683	2020.04.24. 14:19	A komponens osztálya
DivingSuit.java	828	2020.04.24. 14:19	A búvárruha osztálya
Effect.java	819	2020.04.24. 14:19	Effektusok ősosztálya
Eskimo.java	824	2020.04.24. 14:19	Az eszkimó osztálya
FlareGun.java	675	2020.04.24. 14:19	A jelzőpisztoly osztálya
Food.java	757	2020.04.24. 14:19	Az étel osztálya
GameManager.java	9267	2020.04.24. 20:56	A gamemanager osztálya
Ice.java	1346	2020.04.24. 14:19	A fég terület osztálya
Iglu.java	777	2020.04.24. 14:19	Az iglu osztálya
Item.java	1975	2020.04.24. 14:19	Az eszközök ősosztálya
ITman.java	537	2020.04.24. 14:19	Az informatikus osztálya
Main.java	1055	2020.04.24. 20:55	A Main osztálya
NullBuilding.java	953	2020.04.24. 14:16	A NullBuilding osztálya
PackedTent.java	847	2020.04.24. 14:10	A csomagolt sátor osztálya
Palesz.java	838	2020.04.24. 14:19	A pálesz osztálya
PaleszEffect.java	646	2020.04.24. 14:19	A pálesz effektus osztálya
Player.java	5254	2020.04.24. 14:15	A játékosok ősosztálya
Researcher.java	923	2020.04.24. 14:02	A kutató osztálya
Rope.java	715	2020.04.24. 14:02	A kötél osztálya
RopeEffect.java	804	2020.04.24. 13:59	A kötél effektus osztálya
Seal.java	626	2020.04.24. 13:59	A fóka osztálya
Shovel.java	1032	2020.04.24. 13:54	A lapát osztálya
Steppable.java	2242	2020.04.24. 13:53	A mozgathatók ősosztálya
StormEffect.java	562	2020.04.24. 13:44	A storm effektus osztálya
Tent.java	1058	2020.04.24. 13:44	A sátor osztálya
UnstableIce.java	1782	2020.04.24. 13:58	Az instabil jég osztálya
Water.java	7477	2020.04.24. 13:56	A víz osztálya

.class

Fájl neve	Méret (Byte)	Keletkezés ideje	Tartalom
Bear.class	1327	2020.04.24. 20:58	A medve osztálya
BearEffect.class	594	2020.04.24. 20:58	A medve effektus osztálya
Building.class	961	2020.04.24. 20:58	Az épület osztálya
Component.class	646	2020.04.24. 20:58	A komponens osztálya
DivingSuit.class	673	2020.04.24. 20:58	A búvárruha osztálya

Effect.class	643	2020.04.24. 20:58	Effektusok őosztálya
Eskimo.class	894	2020.04.24. 20:58	Az eszkimó osztálya
FlareGun.class	761	2020.04.24. 20:58	A jelzőpisztoly osztálya
Food.class	684	2020.04.24. 20:58	A étel osztálya
GameManager.class	8601	2020.04.24. 20:58	A gamemanager osztálya
Ice.class	1139	2020.04.24. 20:58	A fég terület osztálya
Iglu.class	678	2020.04.24. 20:58	A iglu osztálya
Item.class	1421	2020.04.24. 20:58	A eszközök őosztálya
ITman.class	569	2020.04.24. 20:58	A informatikus osztálya
Main.class	1678	2020.04.24. 20:58	A Main osztálya
NullBuilding.class	1006	2020.04.24. 20:58	A NullBuilding osztálya
PackedTent.class	960	2020.04.24. 20:58	A csomagolt sátor osztálya
Palesz.class	876	2020.04.24. 20:58	A pálesz osztálya
PaleszEffect.class	583	2020.04.24. 20:58	A pálesz effektus osztálya
Player.class	3820	2020.04.24. 20:58	A játékosok őosztálya
Researcher.class	977	2020.04.24. 20:58	A kutató osztálya
Rope.class	822	2020.04.24. 20:58	A kötél osztálya
RopeEffect.class	671	2020.04.24. 20:58	A kötél effektus osztálya
Seal.class	602	2020.04.24. 20:58	A fóka osztálya
Shovel.class	864	2020.04.24. 20:58	A lapát osztálya
Steppable.class	1509	2020.04.24. 20:58	A mozgathatók őosztálya
StormEffect.class	598	2020.04.24. 20:58	A storm effektus osztálya
Tent.class	1045	2020.04.24. 20:58	A sátor osztálya
UnstableIce.class	2053	2020.04.24. 20:58	Az instabil jég osztálya
Water.class	6027	2020.04.24. 20:58	A víz osztálya

.input

Fájl neve	Méret (Byte)	Keletkezés ideje	Tartalom
BearMoveAndAttacks.input	80	2020.04.24. 15:08	Bemeneti parancsok
BearMoves.input	80	2020.04.24. 15:08	Bemeneti parancsok
DiggingItem.input	37	2020.04.24. 15:08	Bemeneti parancsok
DiggingItemWithEmptyFrozenItemsList.input	68	2020.04.24. 15:08	Bemeneti parancsok
DiggingSnowLayer.input	37	2020.04.24. 15:08	Bemeneti parancsok
DropAnItem.input	74	2020.04.24. 15:08	Bemeneti parancsok
EskimoSkillOnIceWithTent.input	35	2020.04.24. 15:08	Bemeneti parancsok
GameOver.input	82	2020.04.24. 15:08	Bemeneti parancsok
GameOverWithPlayerOnWaterWithSafeAttributeFalse.input	69	2020.04.24. 15:08	Bemeneti parancsok
GameOverWithPlayerWithZeroBodyHeat.input	38	2020.04.24. 15:08	Bemeneti parancsok
ITManSkill.input	71	2020.04.24. 15:08	Bemeneti parancsok
PickUpAnItem.input	50	2020.04.24. 15:08	Bemeneti parancsok
ResearcherSkill.input	56	2020.04.24. 15:08	Bemeneti parancsok
StepOntoFieldWithBear.input	80	2020.04.24. 15:08	Bemeneti parancsok
StepOntoIce.input	61	2020.04.24. 15:08	Bemeneti parancsok

StepOntoUnstableIce.input	61	2020.04.24. 15:08	Bemeneti parancsok
StepOntoWater.input	61	2020.04.24. 15:08	Bemeneti parancsok
StormEffectOnIceWhitIglu.input	71	2020.04.24. 15:08	Bemeneti parancsok
StormEffectOnIceWhitTent.input	71	2020.04.24. 15:08	Bemeneti parancsok
StromEffectOnIceWithNullBuilding.input	98	2020.04.24. 15:08	Bemeneti parancsok
UseComponentWith1Components.input	70	2020.04.24. 15:08	Bemeneti parancsok
UseComponentWith2Components.input	99	2020.04.24. 15:08	Bemeneti parancsok
UseComponentWith3Components.input	128	2020.04.24. 15:08	Bemeneti parancsok
UseDivingSuit.input	71	2020.04.24. 15:08	Bemeneti parancsok
UseDivingSuitWhitSafeAttributeOn.input	71	2020.04.24. 15:08	Bemeneti parancsok
UseFood.input	65	2020.04.24. 15:08	Bemeneti parancsok
UseFoodOnMaxBodyHeat.input	65	2020.04.24. 15:08	Bemeneti parancsok
UsePackedTent.input	110	2020.04.24. 15:08	Bemeneti parancsok
UsePackedTentOnIceWhitIglu.input	80	2020.04.24. 15:08	Bemeneti parancsok
UseRope.input	110	2020.04.24. 15:08	Bemeneti parancsok
UseRopeInWrongDirection.input	110	2020.04.24. 15:08	Bemeneti parancsok
UseShovelToDigItem.input	73	2020.04.24. 15:08	Bemeneti parancsok
UseShovelToDigSnowLayer.input	73	2020.04.24. 15:08	Bemeneti parancsok

.ref

Fájl neve	Méret (Byte)	Keletkezés ideje	Tartalom
BearMoveAndAttacks.ref	810	2020.04.24. 16:42	Elvárt kimenet
BearMoves.ref	681	2020.04.24. 12:47	Elvárt kimenet
DiggingItem.ref	502	2020.04.24. 12:47	Elvárt kimenet
DiggingItemWithEmptyFrozenItemsList.ref	365	2020.04.24. 12:47	Elvárt kimenet
DiggingSnowLayer.ref	446	2020.04.24. 12:47	Elvárt kimenet
DropAnItem.ref	730	2020.04.24. 12:47	Elvárt kimenet
EskimoSkillOnIceWhitTent.ref	438	2020.04.24. 12:47	Elvárt kimenet
GameOver.ref	627	2020.04.24. 12:47	Elvárt kimenet
GameOverWithPlayerOnWaterWhitSafeAttributeFalse.ref	446	2020.04.24. 16:44	Elvárt kimenet
GameOverWithPlayerWithZeroBodyHeat.ref	866	2020.04.24. 16:44	Elvárt kimenet
ITManSkill.ref	364	2020.04.24. 12:47	Elvárt kimenet
PickUpAnItem.ref	627	2020.04.24. 12:47	Elvárt kimenet
ResearcherSkill.ref	446	2020.04.24. 12:47	Elvárt kimenet

StepOntoFieldWhitBear.ref	866	2020.04.24. 16:44	Elvárt kimenet
StepOntoIce.ref	940	2020.04.24. 12:47	Elvárt kimenet
StepOntoUnstableIce.ref	940	2020.04.24. 12:47	Elvárt kimenet
StepOntoWater.ref	818	2020.04.24. 12:47	Elvárt kimenet
StormEffectOnIceWhitIglu.ref	1980	2020.04.24. 12:47	Elvárt kimenet
StormEffectOnIceWhitTent.ref	1980	2020.04.24. 12:47	Elvárt kimenet
StromEffectOnIceWithNullBuilding.ref	545	2020.04.24. 12:47	Elvárt kimenet
UseComponentWith1Components.ref	236	2020.04.24. 12:47	Elvárt kimenet
UseComponentWith2Components.ref	279	2020.04.24. 12:47	Elvárt kimenet
UseComponentWith3Components.ref	324	2020.04.24. 16:45	Elvárt kimenet
UseDivingSuit.ref	187	2020.04.24. 12:47	Elvárt kimenet
UseDivingSuitWhitSafeAttributeOn.ref	181	2020.04.24. 12:47	Elvárt kimenet
UseFood.ref	188	2020.04.24. 12:47	Elvárt kimenet
UseFoodOnMaxBodyHeat.ref	181	2020.04.24. 12:47	Elvárt kimenet
UsePackedTent.ref	538	2020.04.24. 12:47	Elvárt kimenet
UsePackedTentOnIceWhitIglu.ref	674	2020.04.24. 12:47	Elvárt kimenet
UseRope.ref	1749	2020.04.24. 16:47	Elvárt kimenet
UseRopeInWrongDirection.ref	1752	2020.04.24. 12:47	Elvárt kimenet
UseShovelToDigItem.ref	735	2020.04.24. 12:47	Elvárt kimenet
UseShovelToDigSnowLayer.ref	674	2020.04.24. 12:47	Elvárt kimenet

Megjegyzés: a .input bemeneti fájlokhoz a programnak az azonos nevű .ref fájl tartalmát kell előállítania. Ennek ellenőrzésére szolgál a Comparator program.

Comparator

Fájl neve	Méret (Byte)	Keletkezés ideje	Tartalom
Main.java	2472	2020.04.24 20:59	Két fájlt összehasonlító kód

10.1.2 Fordítás

A fordításhoz el kell navigálni a terminálban a projekt mappájához és a következő parancsokat kell kiadni:

```
set PATH=%PATH%;c:\Program Files\Java\jdk1.8.0_231\bin
set JAVA_HOME=c:\Program Files\Java\jdk1.8.0_231
```

```
jar cf proto.jar aranyaszok/*.java aranyaszok/command/*.java
```

(a comparator fordítása analóg módon megegyezik a proto fordításával)

10.1.3 Futtatás

Alapértelmezetten a Windows 10 rendszeren nincs engedélyezve a digitálisan nem aláírt scriptek futtatása. Az engedélyezéshez a következő parancsra van szükség:

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
```

A master script futtatásához el kell navigálni a tesztelő könyvtárhoz és a .\master.ps1 parancsot kell kiadni. A scriptet argumentummal indítva csak az adott teszteset fut le. Argumentum nélkül minden teszteset ami az Input könyvtáron belül van lefut. Fontos hogy legyen minden tesztesethez egy elvárt kimenet a Reference könyvtárban. Az esetleges hibákról, a szabványok kimeneten tájékoztatást kapunk.

A következő képen egy futási eredmény látható az összes tesztesettel. A kimeneten az első oszlopban a ‘+’ -jel a proto helyes futását jelöli. (pl. nincs értelmezhetetlen input parancs) A második ‘+’-jel a comparator sikeres futását jelenti, tehát a proto által generált kimenet megegyezik az elvárt kimenettel.

Hiba esetén a temp.data tartalmazza a generált kimenetet, ezt lehet összevetni az adott referenciával. A futás közbeni hibákat az error.data fájl tartalmazza.

```
Administrator: Windows PowerShell
PS C:\WINDOWS> cd..
PS C:\> cd ..\Users\Marci\Desktop\projlab\
PS C:\Users\Marci\Desktop\projlab> .\master.ps1

++ BearMoveAndAttacks
++ BearMoves
++ DiggingItem
++ DiggingItemWithEmptyFrozenItemList
++ DiggingSnowLayer
++ DropAnItem
++ EskimoSkillOnIceWhitTent
++ GameOver
++ GameOverWithPlayerOnWaterWhitSafeAttributeFalse
++ GameOverWithPlayerWithZeroBodyHeat
++ ITManSkill
++ PickUpAnItem
++ ResearcherSkill
++ StepOntoFieldWhitBear
++ StepOntoIce
++ StepOntoUnstableIce
++ StepOntoWater
++ StormEffectOnIceWhitIglu
++ StormEffectOnIceWhitTent
++ StromEffectOnIceWithNullBuilding
++ UseComponentWith1Components
++ UseComponentWith2Components
++ UseComponentWith3Components
++ UseDivingSuit
++ UseDivingSuitWhitSafeAttributeOn
++ UseFood
++ UseFoodOnMaxBodyHeat
++ UsePackedTent
++ UsePackedTentOnIceWhitIglu
++ UseRope
++ UseRopeInWrongDirection
++ UseShovelToDigItem
++ UseShovelToDigSnowLayer

Sikerarany: 33/33

PS C:\Users\Marci\Desktop\projlab>
```

Név	Módosítás dátuma	Típus	Méret
Input	2020. 04. 24. 16:38	Fájlmappa	
Maps	2020. 04. 24. 16:18	Fájlmappa	
References	2020. 04. 24. 15:25	Fájlmappa	
comp	2020. 04. 24. 16:13	Executable Jar File	2 KB
master	2020. 04. 24. 16:38	Windows PowerS...	3 KB
proto	2020. 04. 24. 16:20	Executable Jar File	485 KB

A **Tester** nevű mappa tartalma fent látható, ebből:

- Input: Azok az input fájlok, melyekből a proto futása során kiolvassa, az egyes tesztekhez tartozó bemeneti parancsokat, majd végrehajtja azokat.
- Maps: A tesztek alapjául szolgáló játéktér fájlba mentve, minden teszt elején betöltyük ezt a map-ot, majd ezen végezzük el a teszteket.
- comp.jar: A komparátorból generált .jar fájlt futtatja a master script, a referencia fájlok segítségével ellenőrzi a proto által generált kimenetet, amit a proto előzetesen a temp.data fájlba ment el.
- master: A szkript futtatásával a tesztek végrehajthatóak.
- proto: A proto programból generált .jar fájlt futtatja a master script.

Ha a program futása során bármilyen hiba lép fel, akkor a fenti mappában egy error.data nevű fájlba íródik ki a hibaüzenet, amit a szkript futása során is megjelenít futása során.

10.2 Tesztek jegyzőkönyvei

10.2.1 DiggingSnowLayer

Teszteleő neve	Rédai Attila
Teszt időpontja	2020.04.22.

10.2.2 DiggingItem

Teszteleő neve	Rédai Attila
Teszt időpontja	2020.04.22.

10.2.3 DiggingItemWithEmptyFrozenItemList

Teszteleő neve	Rédai Attila
Teszt időpontja	2020.04.22.

10.2.4 DropAnItem

Teszteleő neve	Rédai Attila
Teszt időpontja	2020.04.22.

10.2.5 PickUpAnItem

Teszteleő neve	Rédai Attila
Teszt időpontja	

Teszt időpontja	2020.04.22.
------------------------	-------------

10.2.6 EskimoSkillOnIceWhitTent

Tesztelő neve	Rédai Attila
Teszt időpontja	2020.04.22.

10.2.7 ITManSkill

Tesztelő neve	Nagy Márton
Teszt időpontja	2020.04.22.

10.2.8 ResearcherSkill

Tesztelő neve	Nagy Márton
Teszt időpontja	2020.04.22.

10.2.9 StepOntoIce

Tesztelő neve	Nagy Márton
Teszt időpontja	2020.04.22.

10.2.10 StepOntoUnstableIce

Tesztelő neve	Nagy Márton
Teszt időpontja	2020.04.22.

10.2.11 StepOntoWater

Tesztelő neve	Nagy Márton
Teszt időpontja	2020.04.22.

10.2.12 StepOntoFieldWhitBear

Tesztelő neve	Nagy Márton
Teszt időpontja	2020.04.22.

10.2.13 UseComponentWith1Components

Tesztelő neve	Nagy Márton
Teszt időpontja	2020.04.22.

10.2.14 UseComponentWith2Components

Tesztelő neve	Bodó István
Teszt időpontja	2020.04.22.

10.2.15 UseComponentWith3Components

Tesztelő neve	Bodó István
Teszt időpontja	2020.04.22.

10.2.16 UseFood

Tesztelő neve	Bodó István
Teszt időpontja	2020.04.22.

10.2.17 UseFoodOnMaxBodyHeat

Tesztelő neve	Bodó István
Teszt időpontja	2020.04.22.

10.2.18 UseShovelToDigSnowLayer

Tesztelő neve	Bodó István
Teszt időpontja	2020.04.22.

10.2.19 UseShovelToDigItem

Tesztelő neve	Bodó István
Teszt időpontja	2020.04.22.

10.2.20 UseRope

Tesztelő neve	Bodó István
Teszt időpontja	2020.04.22.

10.2.21 UseRope (hibás teszt)

Tesztelő neve	Bodó István
Teszt időpontja	2020.04.22
Teszt eredménye	A teszt eredményét, miután ellenőrizte a komparátor, jelezte, hogy az egyik mezőn álló játékosok között eltérés található.
Lehetséges hibaok	Futás közben a lista változott, amivel nem számoltunk.
Változtatások	Ideiglenes lista megoldotta a problémát.

10.2.22 UseRopeInWrongDirection

Tesztelő neve	Bodó István
Teszt időpontja	2020.04.22.

10.2.23 UseDivingSuit

Tesztelő neve	Gaál Gergely
Teszt időpontja	2020.04.22.

10.2.24 UseDivingSuitWhitSafeAttributeOn

Tesztelő neve	Gaál Gergely
Teszt időpontja	2020.04.22.

10.2.25 UsePackedTent

Tesztelő neve	Gaál Gergely
Teszt időpontja	2020.04.22.

10.2.26 UsePackedTentOnIceWhitIglu

Tesztelő neve	Gaál Gergely
Teszt időpontja	2020.04.22.

10.2.27 BearMoves

Tesztelő neve	Gaál Gergely
Teszt időpontja	2020.04.22.

10.2.28 BearMoveAndAttacks

Tesztelő neve	Gaál Gergely
Teszt időpontja	2020.04.22.

10.2.29 StormEffectOnIceWhitIglu

Tesztelő neve	Gaál Gergely
Teszt időpontja	2020.04.22.

10.2.30 StormEffectOnIceWhitTent

Tesztelő neve	Varga Levente
Teszt időpontja	2020.04.22.

10.2.31 StromEffectOnIceWithNullBuilding

Tesztelő neve	Varga Levente
Teszt időpontja	2020.04.22.

10.2.32 GameOver

Tesztelő neve	Varga Levente
---------------	---------------

Teszt időpontja	2020.04.22.
------------------------	-------------

10.2.33 GameOverWithPlayerOnWaterWhitSafeAttributeFalse

Teszteleő neve	Varga Levente
Teszt időpontja	2020.04.22.

10.2.34 GameOverWithPlayerWithZeroBodyHeat

Teszteleő neve	Varga Levente
Teszt időpontja	2020.04.22.

10.3 Értékelés

Tag neve	Tag neptun	Munka százalékban
Bodó István	LU93RR	20%
Gaál Gergely	GOXEHI	20%
Varga Levente	JIE1OY	20%
Nagy Márton	DMMP94	20%
Rédai Attila	LYYOST	20%

10.4 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.04.17	4 óra	Bodó Gaál Varga Nagy Rédai	Proto programozása
2020.04.18	5 óra	Bodó Gaál Varga Nagy Rédai	Proto programozása és javítása
2020.04.19	3 óra	Bodó Gaál Varga Nagy Rédai	Script program elkészítése
2020.04.20	4 óra	Bodó Gaál Varga Nagy Rédai	Komparátor programozása. Script javítása
2020.04.21	5 óra	Bodó Gaál Varga Nagy Rédai	Proto átnézése és javítása. Java komment megírása
2020.04.22	3.5 óra	Bodó Gaál Varga Nagy Rédai	Proto tesztelése Script program tesztelése. Teszt dokumentálás

11. Grafikus felület specifikációja

11.1 A grafikus interfész

A program irányítását az egér használatával végezhetjük, billentyűzetről az irányítás nem megvalósított. Az egyes gombok hatásának érvényre jutását a bal egérgomb lenyomásával kezdeményezhetjük az adott gomb képe fölött.

A program támogatja a különböző oldalszámú mezőket is, jelen példában négyzet alakú mezőket hozunk példaként, de mivel a tárolás során gráfkként kezeljük a területet, ezért a későbbiekben bármilyen formájú mezők megjelenítését támogatjuk.

A program egyetlen ablakból áll majd. A játék grafikus ábrázolása során a következő komponenseket jelenítjük meg:

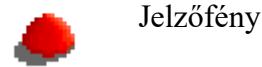
Karakterek:

(A karakterknél sorban láthatjuk a komponenseket alapállapotukban, búvárruhában, vízben búrruha nélkül és vízben búvárruhában)



Tárgyak:





Jelzőfény



Kötél



Patron



Pálinka



Pisztoly



Sátor tárgy (kiásható)

Egyéb komponensek:

Iglu



Jégtábla

(Egy táblán maximum öt réteg hó lehet, a fenti képek azt szemléltetik, hogy hogyan látjuk a jégtáblát különböző számú hőrétegek esetén, 0 - 5)



Medve



Sátor



Víz

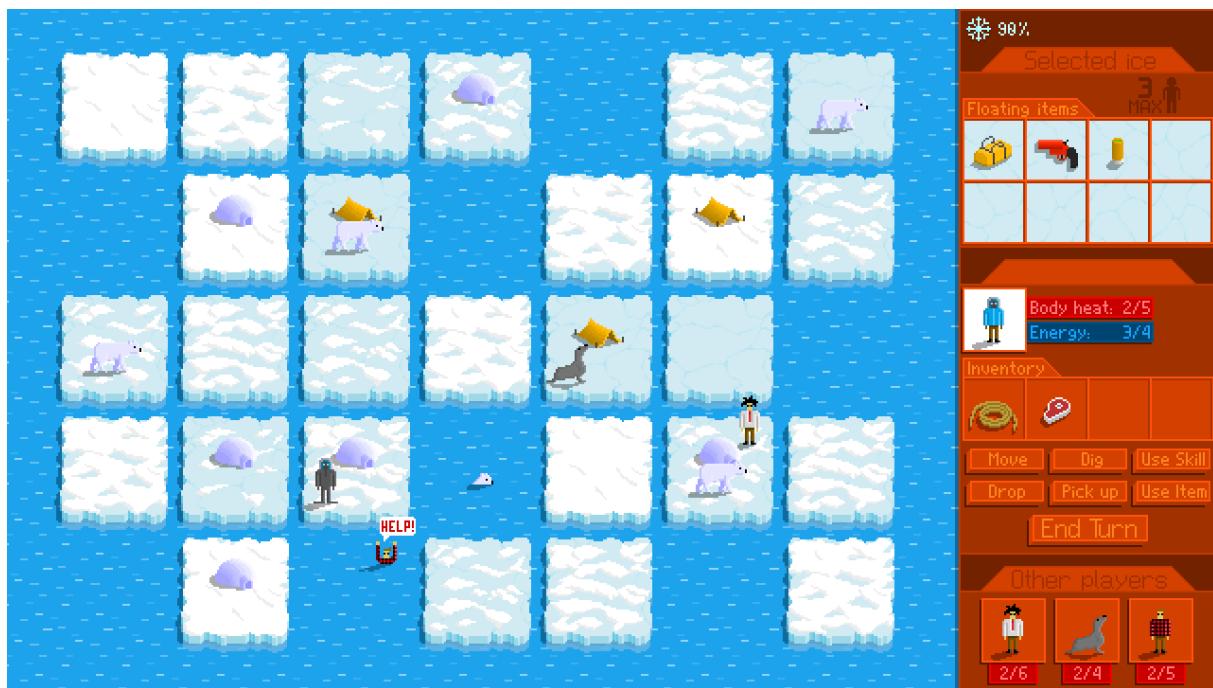
A fentebb vázolt komponensekből és további grafikus elemekből az alábbi kész megjelenítések fognak elkészülni:

Menü



A játék indításakor ez a képernyő fogad minket, a fenti menüpontok közül választhatunk.

Játék állapot minta



A képen egy folyamatban lévő játékról készült kép látható. Bal oldalt a jégtáblák ból és vízből álló játéktér jelenik meg, egyes táblákon hóval, igluval és sátorral, valamint a játékosok és medvék karaktereivel.

Jobb oldalt egy menü látható, ami három fő részből épül fel:

Legfelül vagy azon a jégtáblán lévő tárgyak láthatóak, amin a soron lévő játékos épp áll, vagy amelyikre a játékos legutoljára kattintott. Ezen ablak felső részén jelenik meg az esetleges hóvihar kialakulásának aktuális esélye, továbbá látható az adott tábla teherbírása is, amennyiben azt a kutató karakter korábban már felfedte képességének használatával.

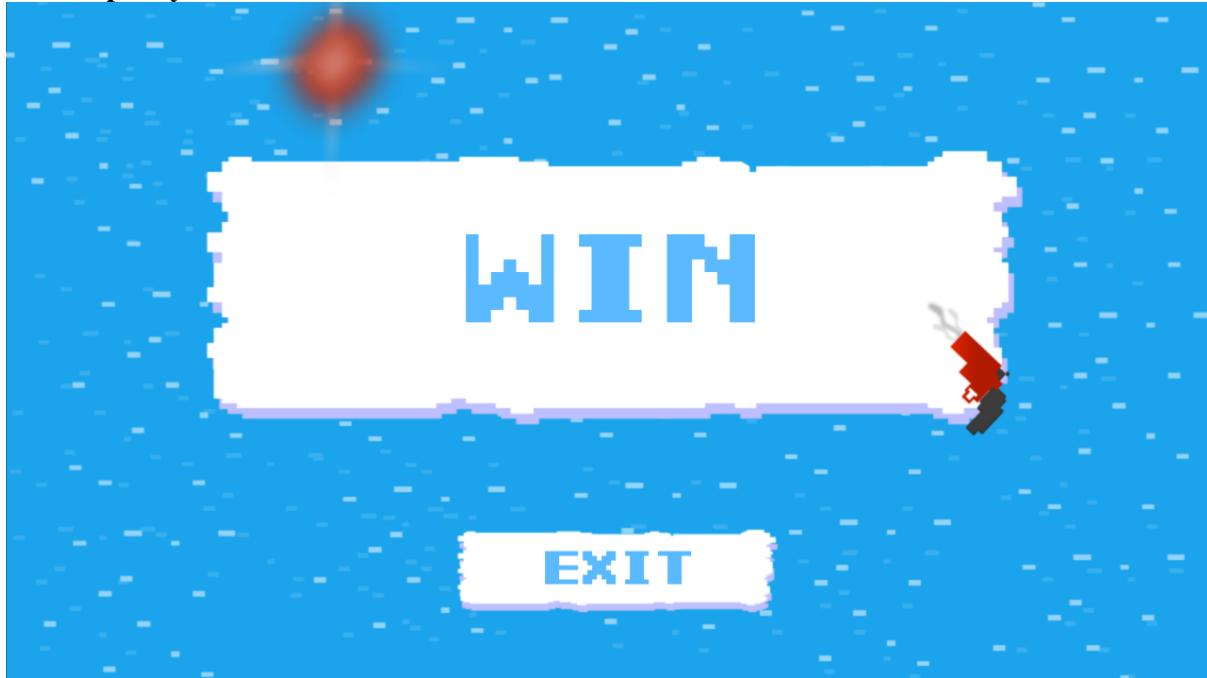
Alatta az aktuálisan soron lévő játékos adatai láthatóak. A kiemelt játékosnak látszik a tárgykészlete, életereleje (teshője) és energiája (hány munkát képes még elvégezni az adott körben). Az "Inventory" alatt minden egyes tevékenységhez találhatók gombok, valamint az "End Turn" gombbal lehet a játékos körét lezárni.

Az aktuális játékos alatt a többi játékos és azok testhői vannak felsorolva.

Lose képernyő



Amikor a játék vereséggel ér véget akkor ez a képernyő fogadja a játékosot. Az "Exit" gomb lenyomásával visszatérhetünk a főmenübe.

Win képernyő

Amikor a játék győzelemmel ér véget akkor ez a képernyő fogadja a játékosat. Az “Exit” gomb lenyomásával visszatérhetünk a főmenübe.

11.2 A grafikus rendszer architektúrája

11.2.1 A felület működési elve

A grafikus rendszer felépítése során törekedtünk az MVC minták követésére. Ezek alapján a program három elkülöníthető részre fog tagolódni, melyek szoros ismerettségen állnak egymással:

- Modell: A korábban leadott proto-nak felel meg. Maga a játék ezen keresztül történik, ez tartja karban az adatokat és hajtja végre a műveleteket a korábbiakban definiált szabályok szerint, a folyamatokat egy GameManager indítja és felügyeli.
- Nézet: Elkülönül a modelltől, de a modellhez hasonlóan egy ViewManager tartja karban és felügyeli az aktuális nézetet, miközben minden megjelenítendő komponens saját magát rajzolja ki.
- Kontroller: Az irányítást gombok segítségével oldjuk meg, minden egyes gombhoz tartozik egy megjelenítésért és egy irányításért felelős rész. A kontroller a nézettel van szorosan összekapcsolva, a nézet pedig magával a modellekkel, így ha gombnyomásra változás történik akkor a nézet és a modell is kap róla értesítést, majd a modellben lezajlott változások megjelenítése miatt frissül a nézet is.

A fentiek alapján is sejthető, hogy a grafikus rendszer tervezésekor a következő alapvetőt követtük:

- Alapvetően a modell és a nézet kapcsolatában a “push modellt” alkalmazzuk, ha a modellben változás történik, akkor a megváltozott komponens értesíti a hozzá tartozó nézetet a frissítés szükségességéről, majd jelez a grafikus rendszernek, hogy a képek újboli kirajzolására van szükség. Ez a kapcsolat úgy valósul meg, hogy a GameManager és a ViewManager között erős ismerettség van
- A grafikus rendszeren belül a “pull modell” érvényesül. Amennyiben a grafikus rendszer értesítést kap a kép újra rajzolásának szükségességéről, akkor minden komponenstől lekérdezi annak állapotát és összeállítja belőle a képet. Ez úgy fog megvalósulni, hogy a ViewManager egy üres képet elindítva, minden egyes View-nak átadva az adott nézet rárajzolhatja magát a képre, majd ha ez megtörtént minden egyes nézettel, akkor a ViewManager kirajzolja a képet a képernyőre.

A program indulásakor megjelenő menü, a játék végén megjelenő győzelem/vereség, illetve maga a játék is egy-egy különböző panel valósít meg, melyeket a ViewManager tart karban, ahogy a program különböző állapotokba lép át a működése során, úgy cserélődnek a képernyön a panelek. Ebből a szempontból a nyitó- és záróképernyők nem különülnek el a játéktól, ugyanaz a megjelenítő kezeli őket.

A menüben van lehetőség korábbi játékállapotok betöltésére és a jelenlegi játék elmentésére. Az ehhez használt pályaformátum megegyezik a korábbiakban használt beépített sserializálással. Mentés vagy betöltés esetén gyakorlatilag csak az aktuális GameManager és ViewManager sserializálás/deserializálása történik tetszőleges fájlba/fájlóból.

Az egyes paneleken található gombok leírása és funkciójuk:

Menü panel:

- Play - A játék indítása
- Load - Tetszőleges játék betöltése
- Save - Aktuális játék mentése
- Exit - Kilépés

Win panel:

- Exit - Kilépés a menübe

Lose panel:

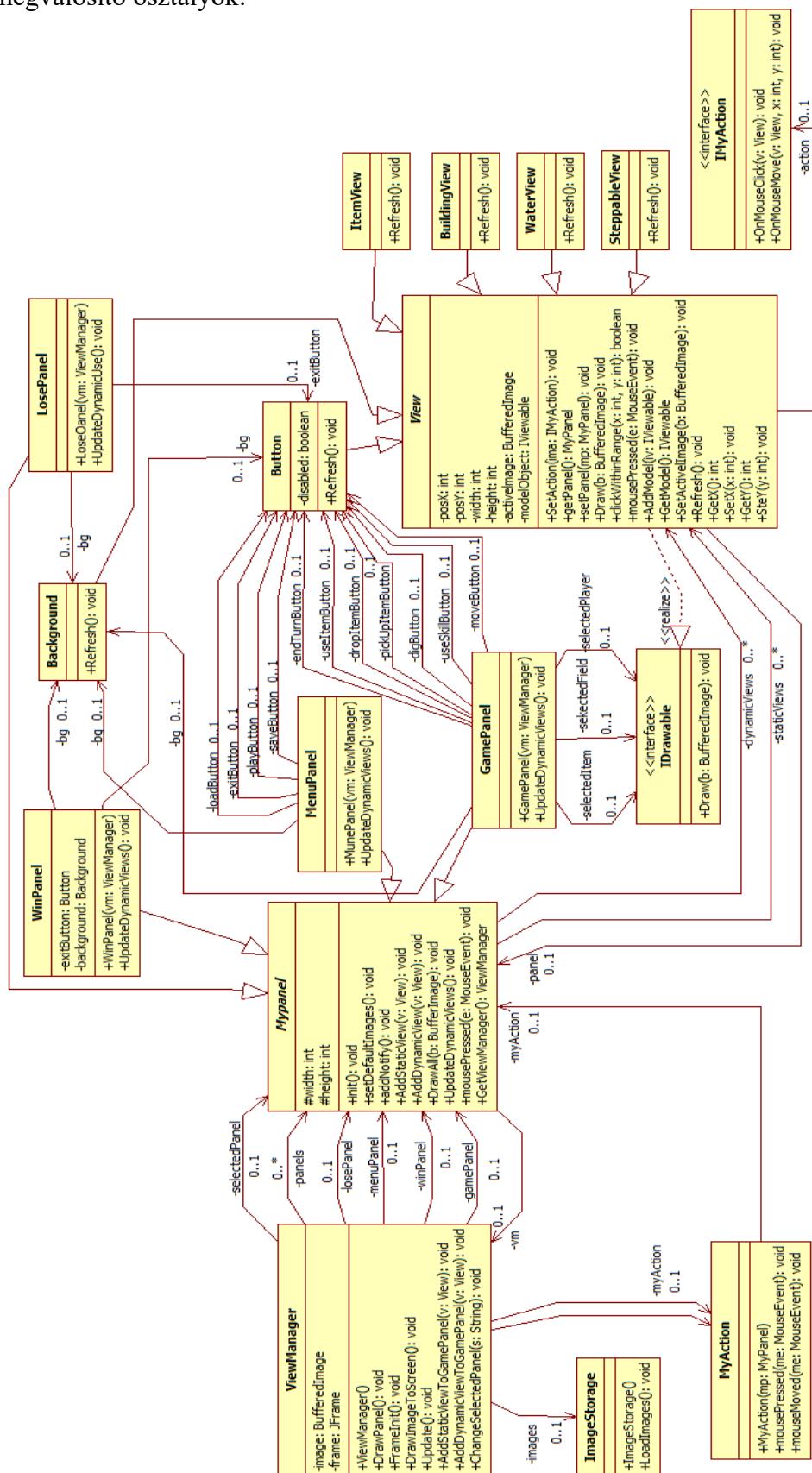
- Exit - Kilépés a menübe

Game panel:

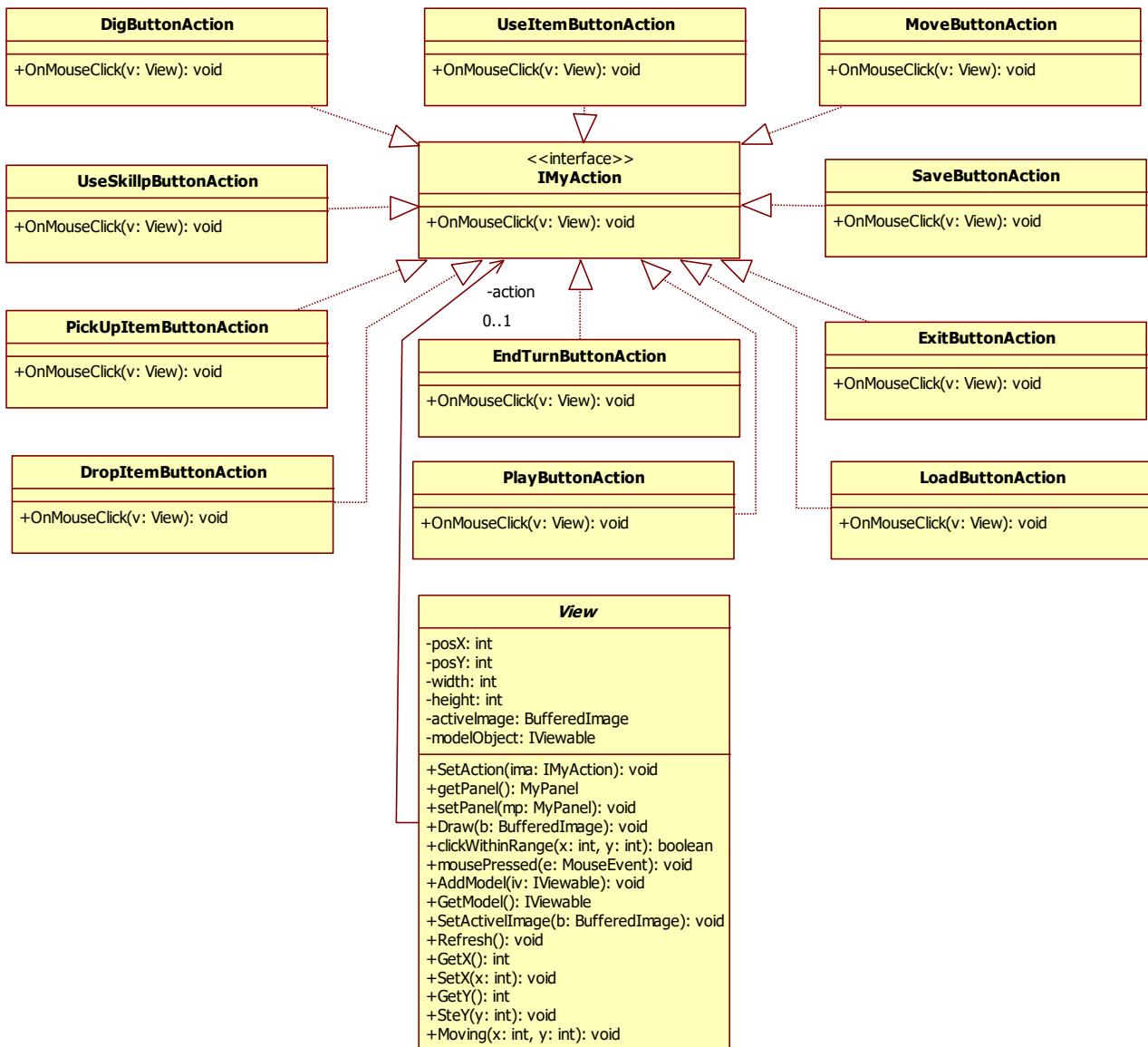
- Move - Az aktuális játékos átléptetése az aktív szomszédos mezőre
- Dig - Az aktuális játékos ás azon a táblán, amin áll
- Use Skill - A játékos különleges képességének használata
- Drop - A játékos eszközei közül a kiválasztott tárgy eldobása a jégtáblára
- Pick up - Arról a jégtábláról, amin a játékos éppen áll felvesz egy kijelölt tárgyat
- Use Item - A játékosnál lévő tárgyak közül egy kijelölt tárgy használata
- End Turn - A játékosnak még lenne lehetősége munkát végezni az adott körben, de lemond ezen jogáról és a következő játékos következik
- Exit - A játék megszakad, visszatérünk a menübe

11.2.2 A felület osztály-struktúrája

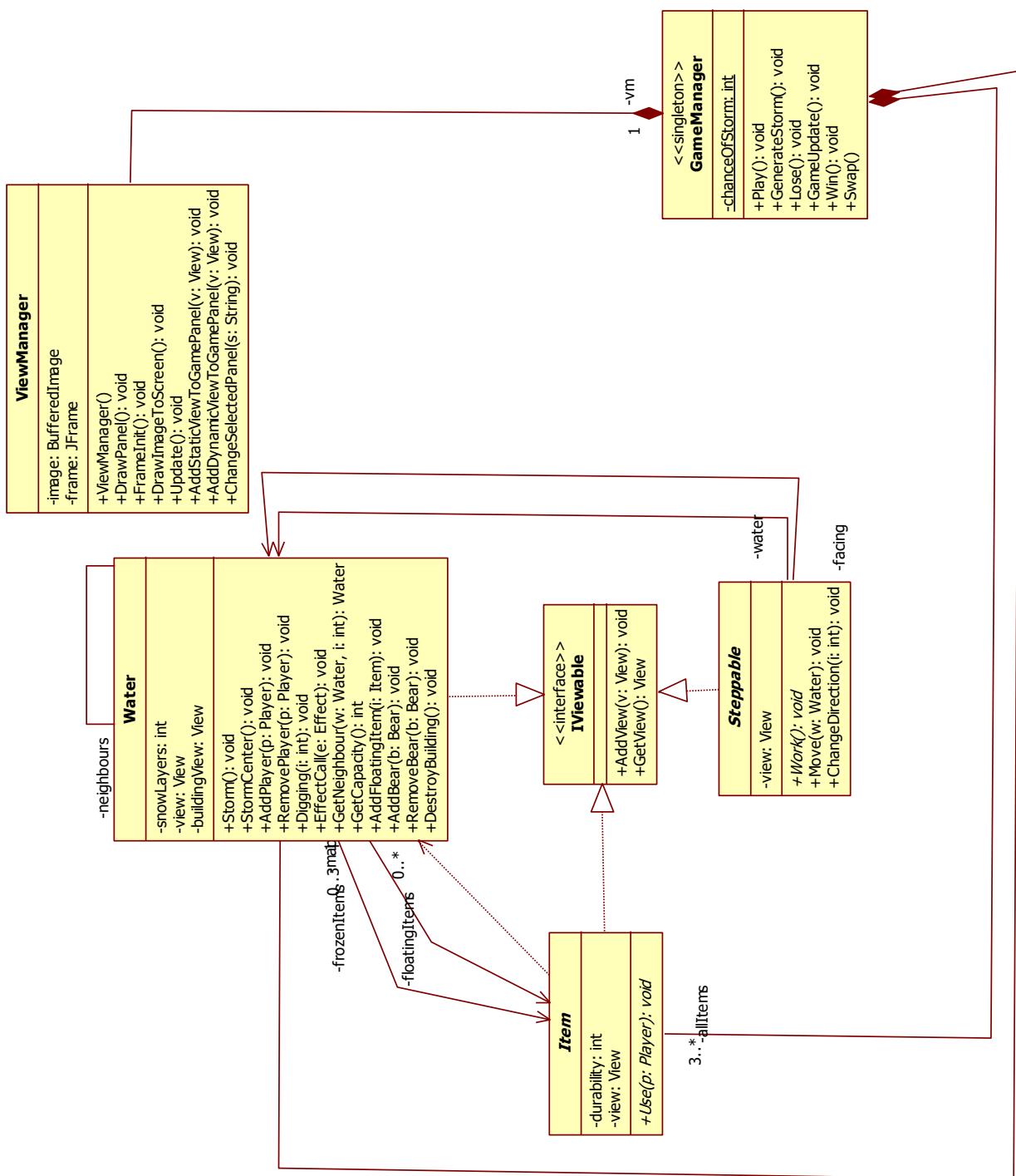
A nézetet megvalósító osztályok:



A kontrollert megvalósító osztályok a nézethez kapcsolva:



A prototípus azon osztályai, amik az új osztályok valamelyikéhez közvetlenül kapcsolódnak:



11.3 A grafikus objektumok felsorolása

A korábban létező, jelenleg módosult osztályok megváltoztatott részletei:

11.3.1 Steppable

- **Attribútumok**
 - **private View view:** Egy Steppable-hez tartozó nézeti objektum

11.3.2 GameManager

- **Metódusok**
 - **public void Swap():** Ha egy játékos köre véget ér akkor ez fogja elindítani az új játékost

11.3.3 Player

- **Attribútumok**
 - **private int workCount:** Ez jelzi, hogy egy játékos mennyi munkája van még

11.3.3 Water

- **Attribútumok**
 - **private View view:** Egy Water-hez tartozó grafikus objektum.
 - **private buildingView:** A Water épületéhez tartozó grafikus objektum.

11.3.4 Item

- **Attribútumok**
 - **private View view:** Egy item-hez tartozó grafikus objektum.

Új grafikus osztályok:

11.3.5 Background

- **Felelősség**
Egy panel háttérét tárolja.
- **Ősosztályok**
View
- **Metódusok**
 - **public void Refresh()**: A kép frissítése.

11.3.6 BuildingView

- **Felelősség**
Egy jégmező tartozik hozzá. A jégmezőn található épület alapján rajzol ki egy sárat vagy iglut.
- **Ősosztályok**
View
- **Metódusok**
 - **public void Refresh()**: A grafikus objektum tárolja a field-et. A field épülete alapján eldönti, hogy melyik képet kell aktívvá tenni.

11.3.7 Button

- **Felelősség**
A felhasználó által kattintható grafikus objektum.
- **Ősosztályok**
View
- **Attribútumok**
 - **private bool disabled**: Az adott körben, az adott játékos használhatja e a gombot.
- **Metódusok**
 - **public void Refresh()**: A gomb képe cserélődik az állapota alapján.

11.3.8 GamePanel

- **Felelősség**
A játék színtere. Ezen a MyPanel-en jelenik meg maga a játék, illetve a játékot segítő jobb oldali menü.
- **Ősosztályok**
MyPanel

- **Attribútumok**
 - **private Button endTurnButton:** Az endTurn gombot reprezentálja.
 - **private Button useItemButton:** Az useItem gombot reprezentálja
 - **private Button dropItemButton:** Az dropItem gombot reprezentálja
 - **private Button pickUpItemButton:** Az pickUpItem gombot reprezentálja
 - **private Button moveButton:** Az move gombot reprezentálja
 - **private Button digButton:** Az dig gombot reprezentálja
 - **private Button useSkillButton:** Az useSkill gombot reprezentálja
 - **private Background bg:** A Panel háttere.
 - **private IDrawable selectedPlayer:** A jobb oldali menüben megjelenítendő játékos.
 - **private IDrawable selectedItem:** A felhasználó által kijelölt floating item vagy egy player inventory-jában lévő item.
 - **private IDrawable selectedField:** Az a mező amelyre a felhasználó kattintott a játéktéren.
- **Metódusok**
 - **public void UpdateDynamicViews():** Frissíti a dinamikusan változó nézeteket.

11.3.9 IDrawable

- **Felelősség**
Azon objektumok amiknek van képe.
- **Metódusok**
 - **public void Draw(BufferedImage image):** A paraméterben kapott képre rajzol a grafikus objektum.

11.3.10 ImageStorage

- **Felelősség**
A játékban használt összes kép fájlt (textúrát) indításkor betölti és eltárolja, hogy a program futása közben már ne kelljen I/O műveleteket végezni.
- **Attribútumok**
 - **public List<BufferedImages> images:** A memóriában tárolt textúrák összessége.
- **Metódusok**
 - **private void LoadImages():** Az összes képet betölti az images tárolóba.
 -

11.3.11 ItemView

- **Felelősség**
Egy Item grafikus nézete.
- **Ősosztályok**
View
- **Metódusok**
 - **public void Refresh():** Egy Item képének frissítése.

11.3.12 LosePanel

- **Felelősség**
Az a MyPanel, mely a játék végét jelenti. Ekkor már csak a játék menübe tud lépni a felhasználó.
- **Ősosztályok**
MyPanel
- **Attribútumok**
 - **private Button exitButton:** A kilépéshez használt gomb.
 - **private Background bg:** A panel háttere.
- **Metódusok**
 - **public void UpdateDynamicViews():** A dinamikusan változó grafikus nézetek frissítése.

11.3.13 MenuPanel

- **Felelősség**
Az a MyPanel mely a játék főmenüjét jeleníti meg.
- **Ősosztályok**
MyPanel
- **Attribútumok**
 - **Button playButton:** A menün a “Play” feliratot reprezentáló gomb.
 - **Button exitButton:** A menün a “Exit” feliratot reprezentáló gomb.
 - **Button loadButton:** A menün a “Load” feliratot reprezentáló gomb.
 - **Button saveButton:** A menün a “Save” feliratot reprezentáló gomb.
 - **Background bg:** A menü háttere.
- **Metódusok**
 - **public void UpdateDynamicViews():** A menün dinamikusan változó képeket frissíti.

11.3.14 MyPanel

- **Felelősség**
Absztrakt osztály. Az egyes állapotokat megjelenítő Panelek ősosztálya.
- **Ősosztályok**
Jpanel

- **Attribútumok**
 - **protected ArrayList<View> staticViews:** Az ebben tárolt View-k számossága csak inicializáláskor változik.
 - **protected ArrayList<View> dynamicViews:** A játék folyamán megjelenhetnek extra View-ok különböző állapotok esetén. Ezek a változások itt tárolódnak.
 - **protected int width:** A panel szélessége.
 - **protected int height:** A panel magassága.
 - **protected ViewManager vm:** ViewManager referenciaja.
- **Metódusok**
 - **public void init():** A panel beállítása.
 - **public void AddStaticView(View view):** A statikus view listához új view hozzáadása.
 - **public void AddDynamicView(View view):** A dinamikusview listához új view hozzáadása.
 - **public void DrawAll(BufferedImage image):** A panelben lévő view-k kirajzolása.
 - **public void UpdateDynamicViews():** Dinamikus view-ok frissítése.
 - **public void mousePressed(MouseEvent e):** Mouse pressed event.

11.3.15 SteppableView

- **Felelősség**
Egy Steppable objektumhoz tartozó grafikus nézet.
- **Ősosztályok**
View
- **Metódusok**
 - **public void Refresh():** A megjelenítendő Steppable pozíóját frissíti.

11.3.16 View

- **Felelősség**
Absztrakt osztály. Az összes view ebből származik. Ez az osztály valósítja meg a megjelenítendő objektumokat.
- **Attribútumok**
 - **private int posX:** View x koordinátája.
 - **private int posY:** View y koordinátája.
 - **private int width:** View szélessége.
 - **private int height:** View magassága.
 - **private MyPanel panel:** MyPanel referencia, amelyhez a View tartozik.
 - **private BufferedImage activeImage:** Azt a képet tartalmazza, melyik jelenleg aktív. Ez kerül rajzolásra.
 - **private IViewable modelObject:** A view-hoz tartozó logikai model.
 - **private IMyAction action:** A grafikus nézethez tartozó eseménykezelő. A játék futása közben keletkező események (triggerek) kezelése.

- **Metódusok**
 - **public void SetAction(IMyAction a):** Egy eseménykezelő beállítása az adott a jelenlegi grafikus nézethez.
 - **public void Draw(BufferedImage image):** View által reprezentált objektum kirajzolása.
 - **public boolean clickWithinRange(int x, int y):** Annak eldöntése, hogy az adott kattintás erre a View-ra vonatkozik-e.
 - **public void mousePressed(MouseEvent e):** A nézethez tartozó action event elsütése.
 - **public void AddModel(IViewable model):** View-hoz tartozó model referencia beállítása.
 - **public void SetActiveImage(BufferedImage image) :** A nézet képének beállítása. Ez a kép rajzolódik rá a MyPanel-re.
 - **public void Refresh():** View frissítése.

11.3.17 ViewManager

- **Felelősség**
A modell és a megjelenés közötti kapcsolatot valósítja meg. A grafikus felület megjelenítésséért felel. Az egyes nézetekben elsült triggereket továbbírja a megfelelő eseménykezelőhöz.
- **Ősosztályok**
View
- **Attribútumok**
 - **private BufferedImage image:** Az a kép melyet a program állít elő minden frissítéskor. Ez jelenik meg a képernyőn.
 - **private JFrame frame:** Az ablak ami megjelenik a képernyőn.
 - **public ImageStorage images:** A képtár létrehozása.
 - **private MyAction myAction:** A framen történő eseményeket elkapó függvény.
 - **private Map<String,MyPanel> panels:** Lista a panelekkel. (az alábbiakat tároljuk itt)
 - **private MyPanel winPanel:** A játék megnyerése során kerül megjelenítésre.
 - **private MyPanel losePanel:** A játék elvesztése során kerül megjelenítésre.
 - **private MyPanel menuPanel:** A játék indulásakor kerül megjelenítésre.
 - **private MyPanel gamePanel:** A játékmenetet elindítva kerül megjelenítésre.
 - **private MyPanel selectedPanel:** A frame-ben az aktuális MyPanel.
- **Metódusok**
 - **public void DrawPanel():** A selectedPanel képe rárajzolódik a megjelenítendő képre. Ez a kép kerül majd kirajzolásra.
 - **public void FrameInit():** A képernyőn megjelenő frame inicializájója.
 - **public void AddDynamicViewToGamePanel(View view):** Egy view-t hozzáad a GamePanel view-oka listájához. A játék init- jehez szükséges.
 - **public void DrawImageToScreen():** A MyPanel által rajzolt kép megjelenik a képernyőn
 - **public void Update():** Történik egy képfrissítés majd kirajzolódik a képernyőre.
 - **public void AddStaticViewToGamePanel(View view):** Egy view-t hozzáad a GamePanel view-oka listájához. A játék init- jehez szükséges.

- **public void ChangeSelectedPanel(String):** Egy kulcsszó alapján kicseréli a frame-ben megjelenítendő MyPanelt.

11.3.18 WaterView

- **Felelősség**
Egy Water objektumhoz tartozó grafikus nézet.
- **Ősosztályok**
View
- **Metódusok**
 - **public void Refresh():** A grafikus nézet beállítása a Water tulajdonságai alapján.

11.3.19 WinPanel

- **Felelősség**
A játék megnyerése során megjelnő MyPanel.
- **Ősosztályok**
MyPanel
- **Attribútumok**
 - **private Button exitButton:** Az a gomb a Penelen mellyek vissza lehet lépni a menübe.
 - **private Backgrund bg:** A panel hátttere.
- **Metódusok**
 - **public void UpdateDynamicViews():** A panelhez tartozó dinamikus nézetek frissítése.

Új kontroller osztályok:

11.3.20 DigButtonAction

- **Felelősség**
Az aktuális játékos ásatása azon a táblán, amin áll.
- **Interfészek**
IMyAction
- **Metódusok**
 - **public void OnMouseClick(View w):** Ha az adott gombra kattint a felhasználó, akkor végrehajtja a gomb funkcióját.

11.3.21 DropItemButtonAction

- **Felelősség**
A játékos eszközei közül a kiválasztott tárgy eldobása a jégtáblára.
- **Interfészek**
IMyAction
- **Metódusok**
 - **public void OnMouseClick(View w):** Ha az adott gombra kattint a felhasználó, akkor végrehajtja a gomb funkcióját.

11.3.22 EndTurnButtonAction

- **Felelősség**
A játékosnak még lenne lehetősége munkát végezni az adott körben, de lemond ezen jogáról és a következő játékos következik.
- **Interfészek**
IMyAction
- **Metódusok**
 - **public void OnMouseClick(View w):** Ha az adott gombra kattint a felhasználó, akkor végrehajtja a gomb funkcióját.

11.3.23 ExitButtonAction

- **Felelősség**
A játék megszakítása, visszatérünk a menübe.
- **Interfészek**
IMyAction
- **Metódusok**
 - **public void OnMouseClick(View w):** Ha az adott gombra kattint a felhasználó, akkor végrehajtja a gomb funkcióját.

11.3.24 IMyAction

- **Felelősség**
Interfész. Miden egyes gombhoz tartozó „Action” ezt valósítja meg.
- **Metódusok**
 - **public void OnMouseClick(View w):** Ha az adott gombra kattint a felhasználó, akkor végrehajtja a gomb funkcióját.

11.3.25 LoadButtonAction

- **Felelősség**
Egy korábbi játékállapot betöltése.
- **Interfészek**
IMyAction
- **Metódusok**
 - **public void OnMouseClick(View w):** Ha az adott gombra kattint a felhasználó, akkor végrehajtja a gomb funkcióját.

11.3.26 MoveButtonAction

- **Felelősség**
Az aktuális játékos átléptetése az aktív szomszédos mezőre.
- **Interfészek**
IMyAction
- **Metódusok**
 - **public void OnMouseClick(View w):** Ha az adott gombra kattint a felhasználó, akkor végrehajtja a gomb funkcióját.

11.3.27 PickUpItemButtonAction

- **Felelősség**
Arról a jégtábláról, amin a játékos éppen áll felvesz egy kijelölt tárgyat.
- **Interfészek**
IMyAction
- **Metódusok**
 - **public void OnMouseClick(View w):** Ha az adott gombra kattint a felhasználó, akkor végrehajtja a gomb funkcióját.

11.3.28 PlayButtonAction

- **Felelősség**
Játék indítása.
- **Interfészek**
IMyAction
- **Metódusok**
 - **public void OnMouseClick(View w):** Ha az adott gombra kattint a felhasználó, akkor végrehajtja a gomb funkcióját.

11.3.29 SaveButtonAction

- **Felelősség**
Aktuális játékmenet elmentése.
- **Interfészek**
IMyAction
- **Metódusok**
 - **public void OnMouseClick(View w):** Ha az adott gombra kattint a felhasználó, akkor végrehajtja a gomb funkcióját.

11.3.30 UseItemButtonAction

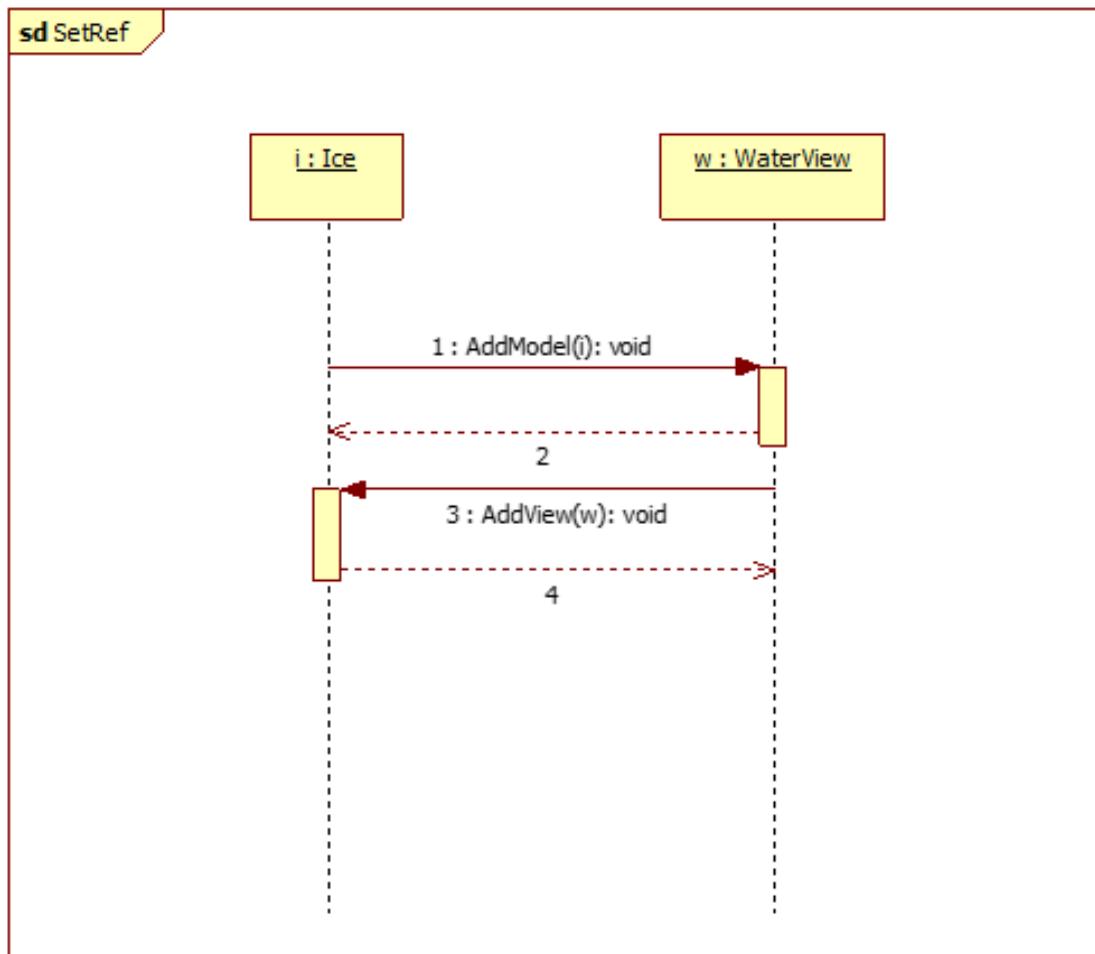
- **Felelősség**
A játékosnál lévő tárgyak közül egy kijelölt tárgy használata.
- **Interfészek**
IMyAction
- **Metódusok**
 - **public void OnMouseClick(View w):** Ha az adott gombra kattint a felhasználó, akkor végrehajtja a gomb funkcióját.

11.3.31 UseSkillButtonAction

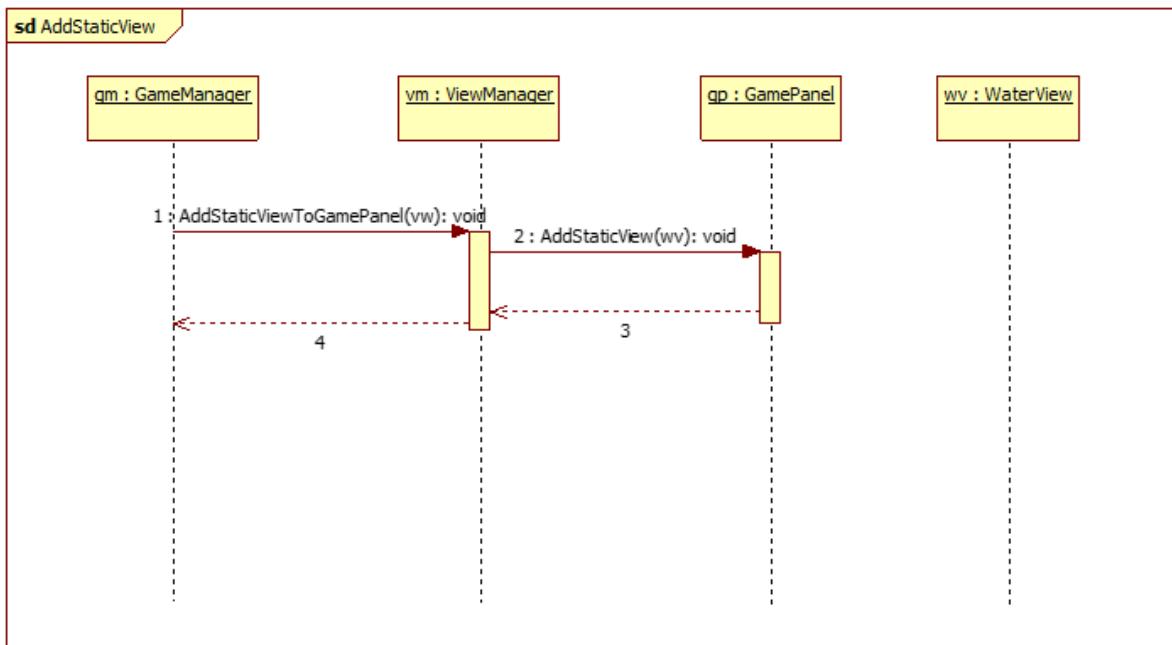
- **Felelősség**
A játékos különleges képességének használata.
- **Interfészek**
IMyAction
- **Metódusok**
 - **public void OnMouseClick(View w):** Ha az adott gombra kattint a felhasználó, akkor végrehajtja a gomb funkcióját.

11.4 Kapcsolat az alkalmazói rendszerrel

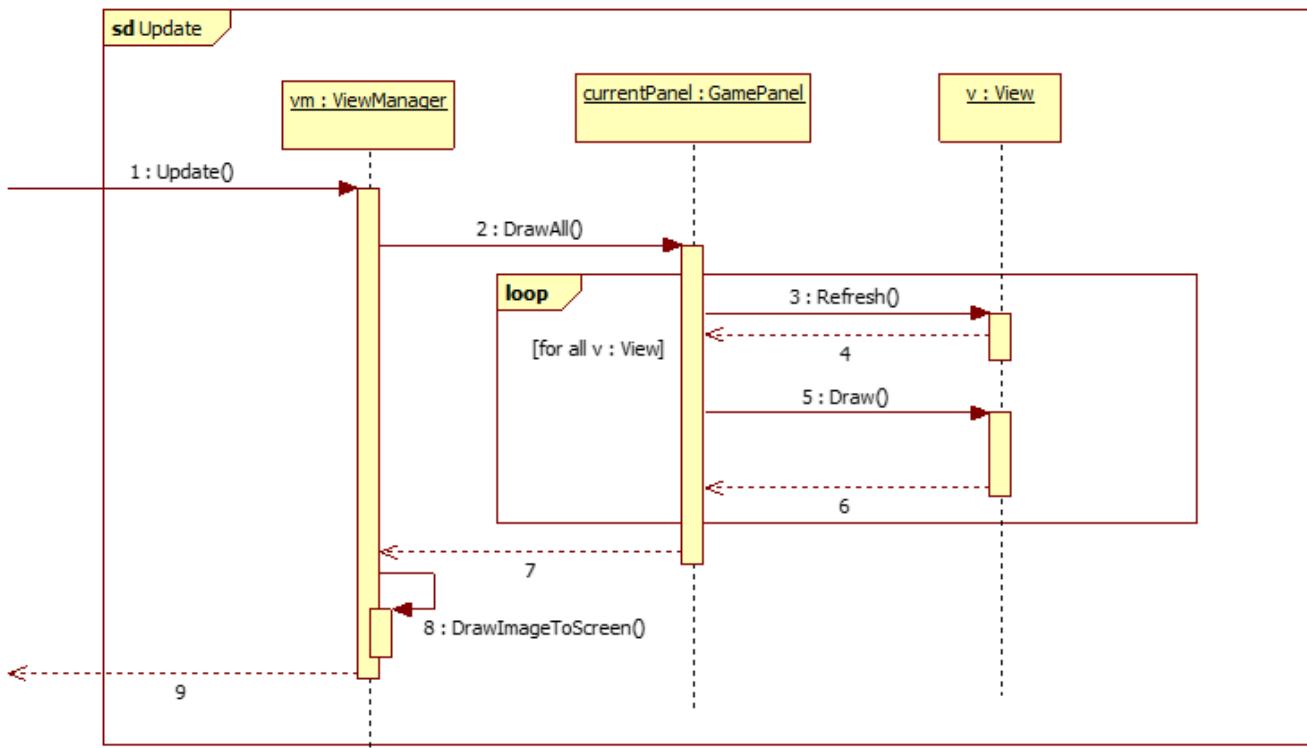
A következő szekvenciadiagramokon egy-egy példát hozunk az egyes működésekre (egy konkrét gombnyomásra, egy objektum kirajzolására, ...stb.), de a legtöbb esetben hasonlóan történik minden elemnél.



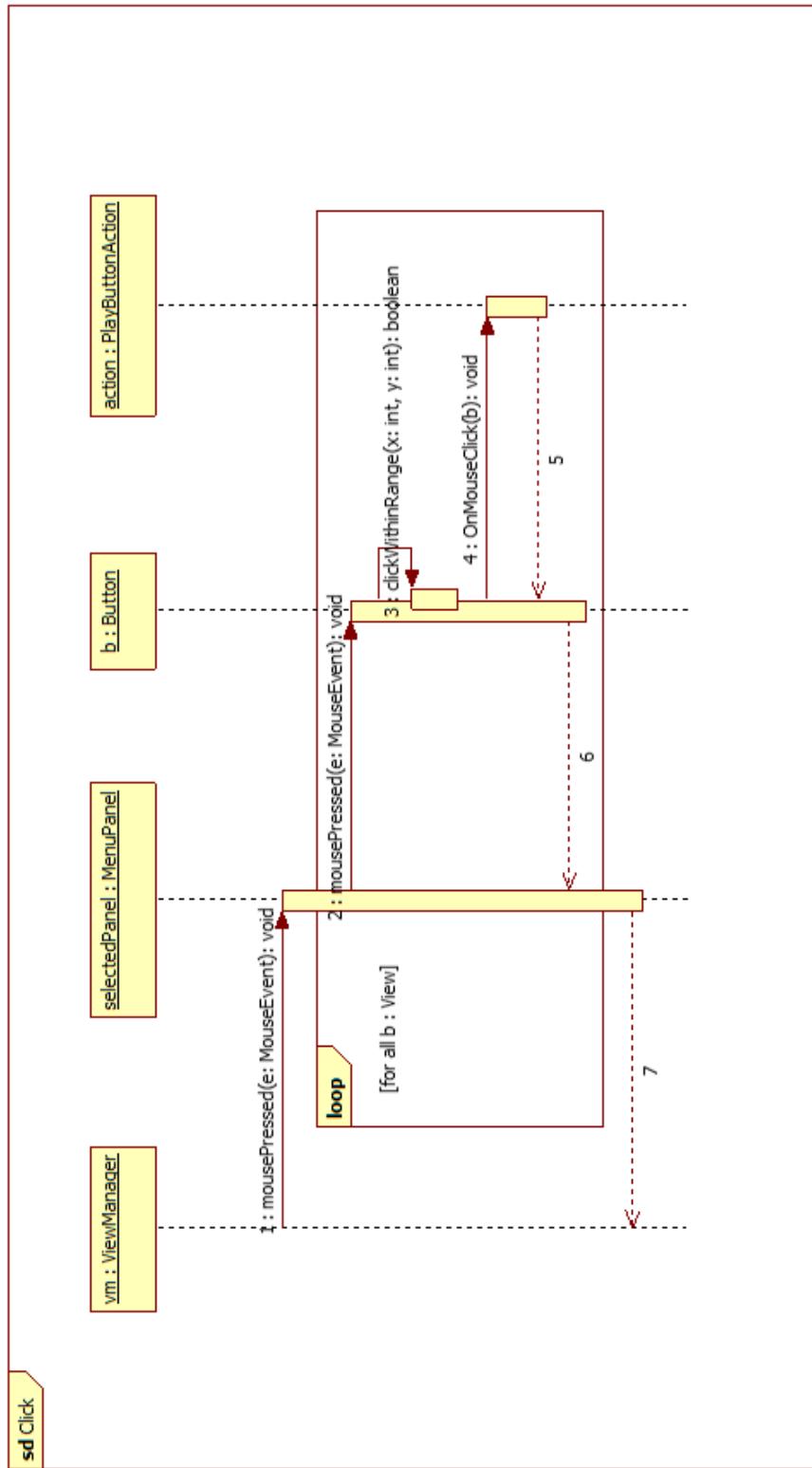
11.4.1 Inicializáláskor a modell objektumhoz létrejött grafikus nézet egymás közötti referenciai beállítása.



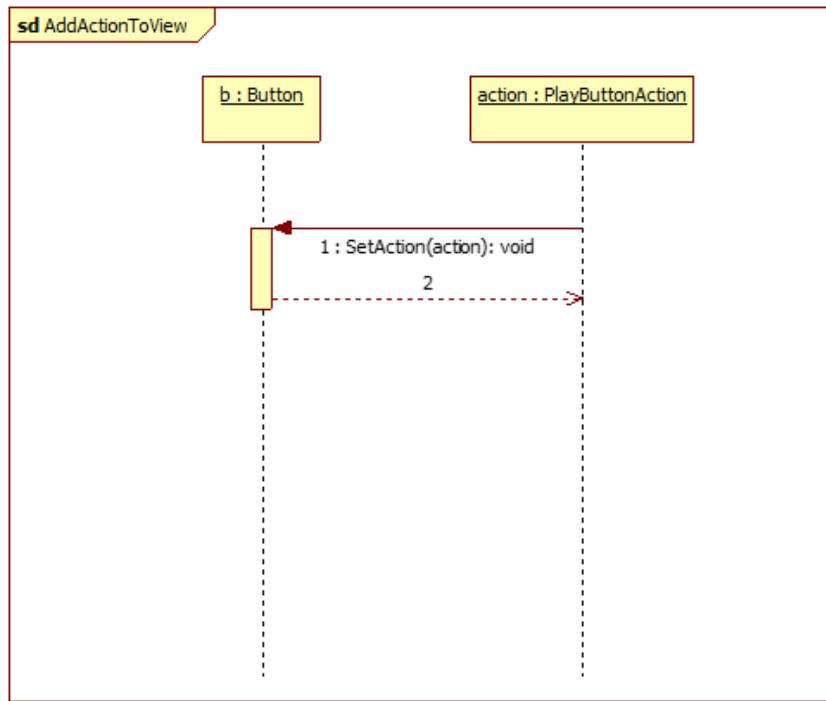
11.4.2 Inicializáláskor a létrejött View-ok hozzáadásra kerülnek a GamePanel statikus view tárolójához.



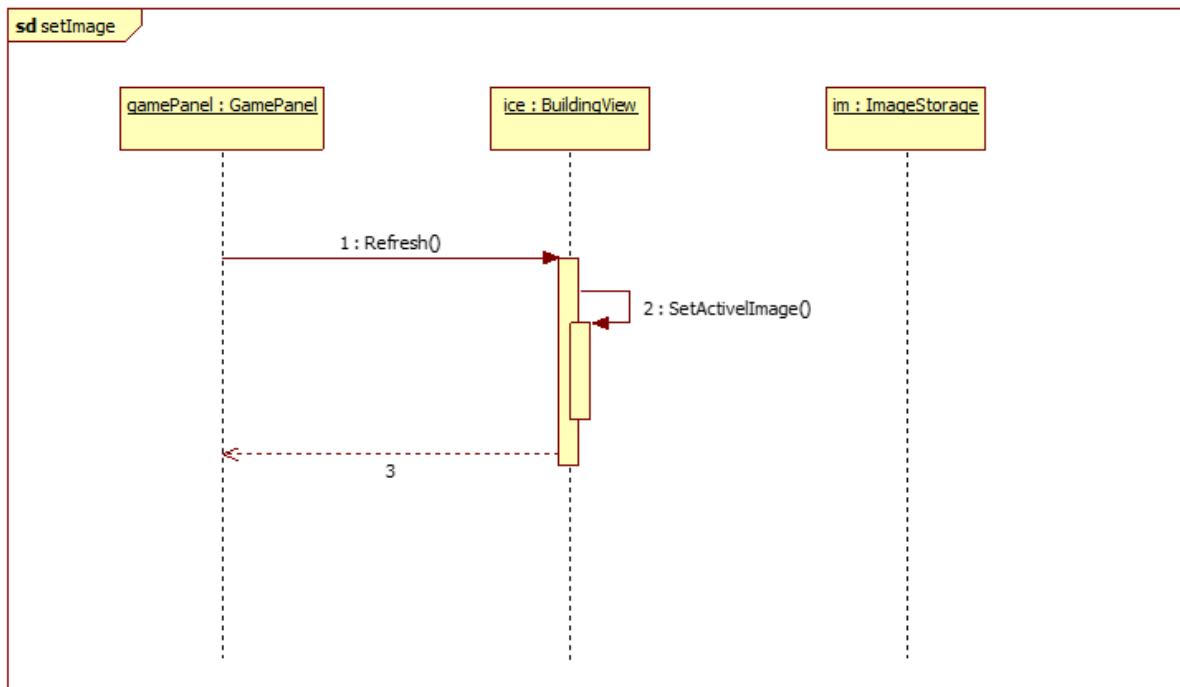
11.4.3 A ViewManager `Update()` hívása újra rajzolja a teljes képet, majd megjeleníti a képernyőn. A rajzolás során minden az aktuális panelhez tartozó view aktuális képe rárajzolódik egy Image-re. Ezt a kész Image-t cseréli le. (double buffering)



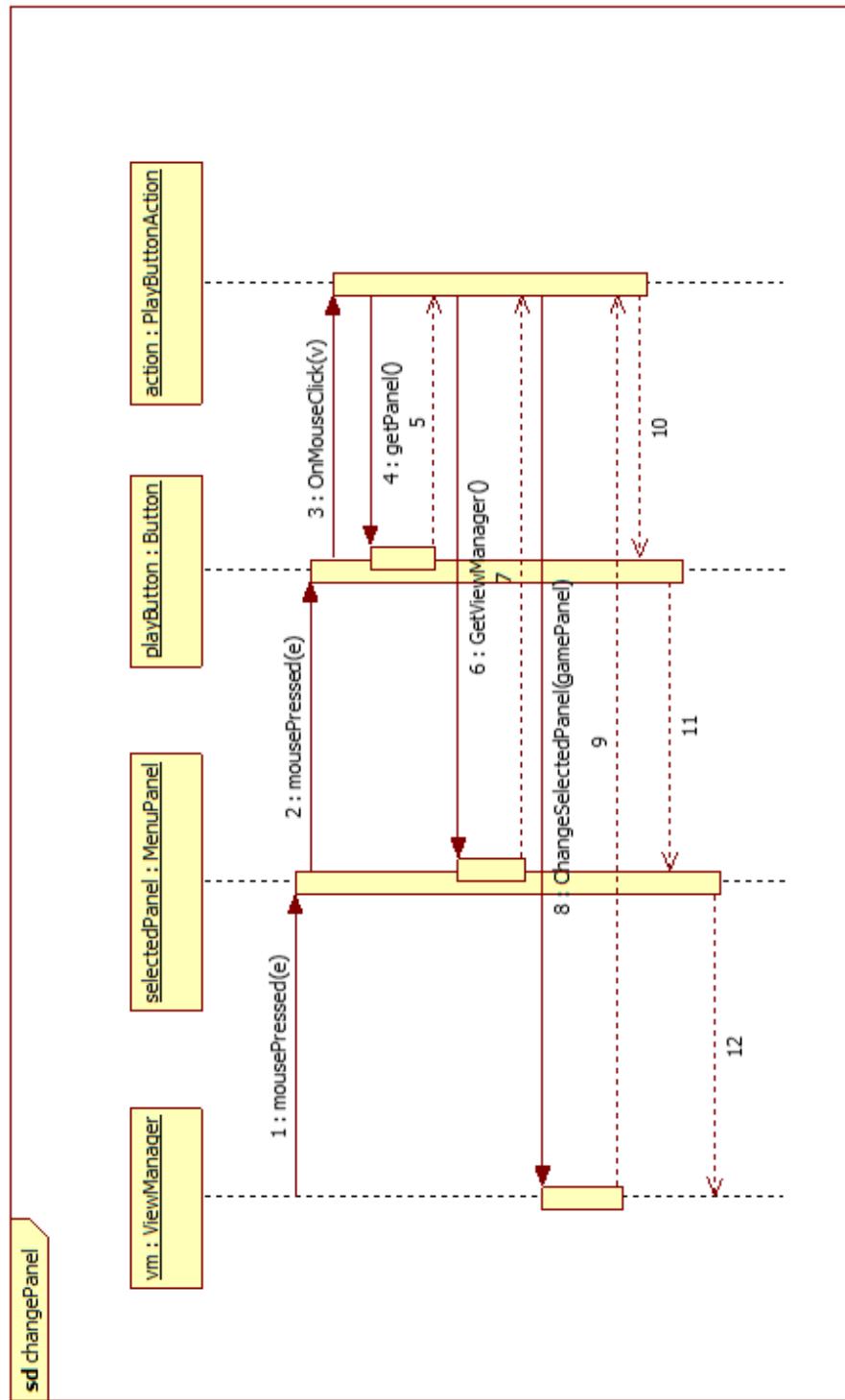
11.4.4 Egy kattintás kezelése első körben az aktuális panelen hívódik. Az esemény tovább adódik az aktuális panelen lévő összes view-ra. Az egyes nézetek előírták hogy a kattintás az ő általuk közrefogott területen történt-e. Ha igen, akkor a megfelelő action click függvénye hívódik.



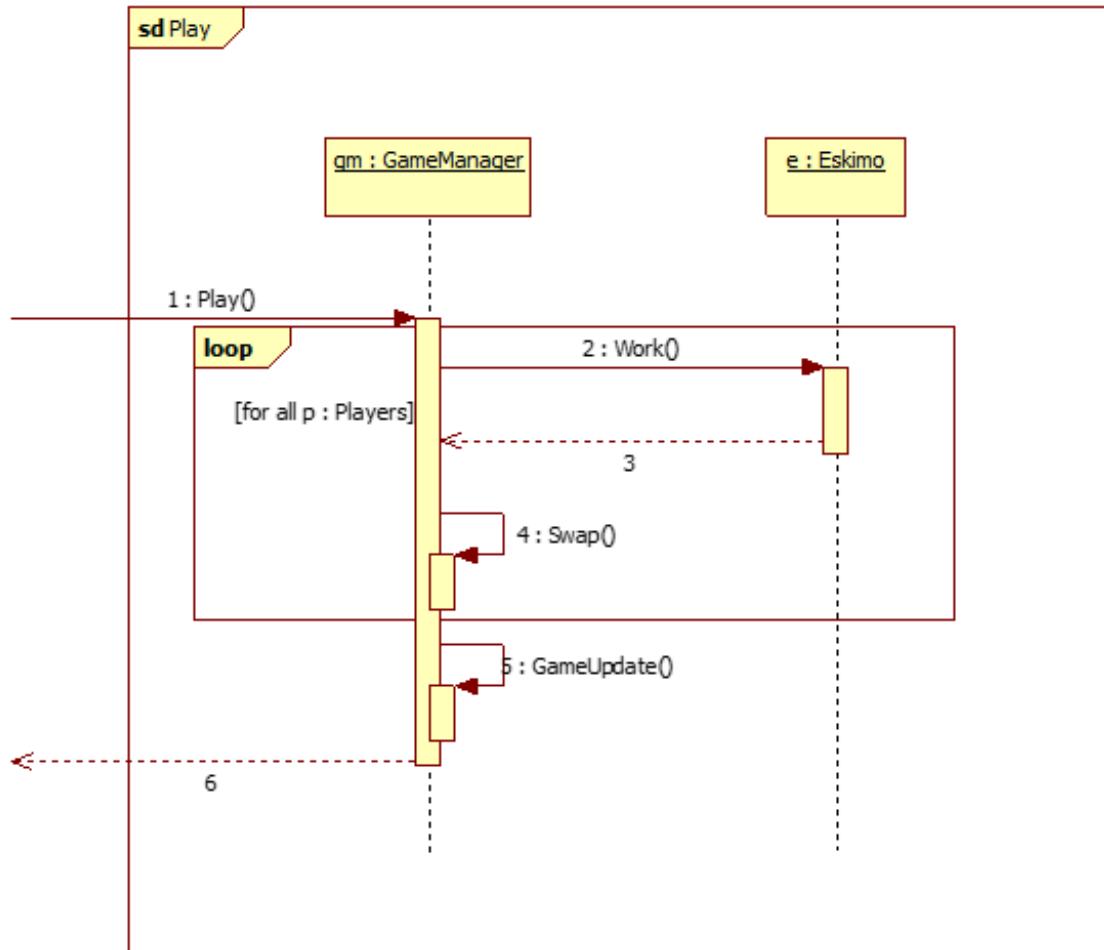
11.4.5 Egy grafikus nézet (view) eseménykezelőjének a beállítása.



11.4.6 A logikai modellben egy Ice Building-je lecserélődött egy Iglu-ra. Ekkor a hozzá tartozó grafikus nézetben frissíteni kell a képet. A kép az ImageStorage tárolóból kerül betöltésre.

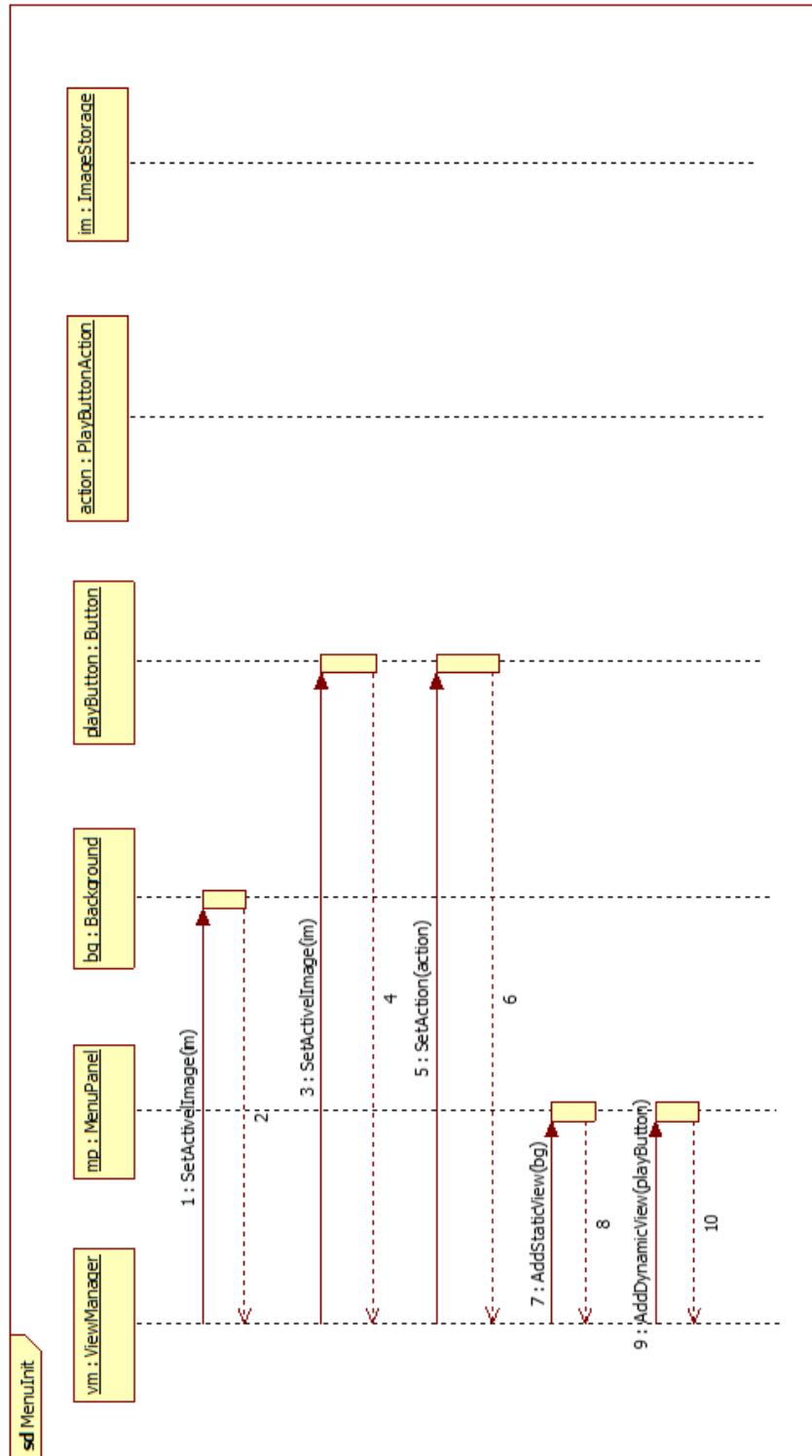


11.4.7 Egy Panel cseréje. Jelen esetben a főmenü “Play” gombjára kattintva a MenuPanel lecserélődik a GamePanel-re.

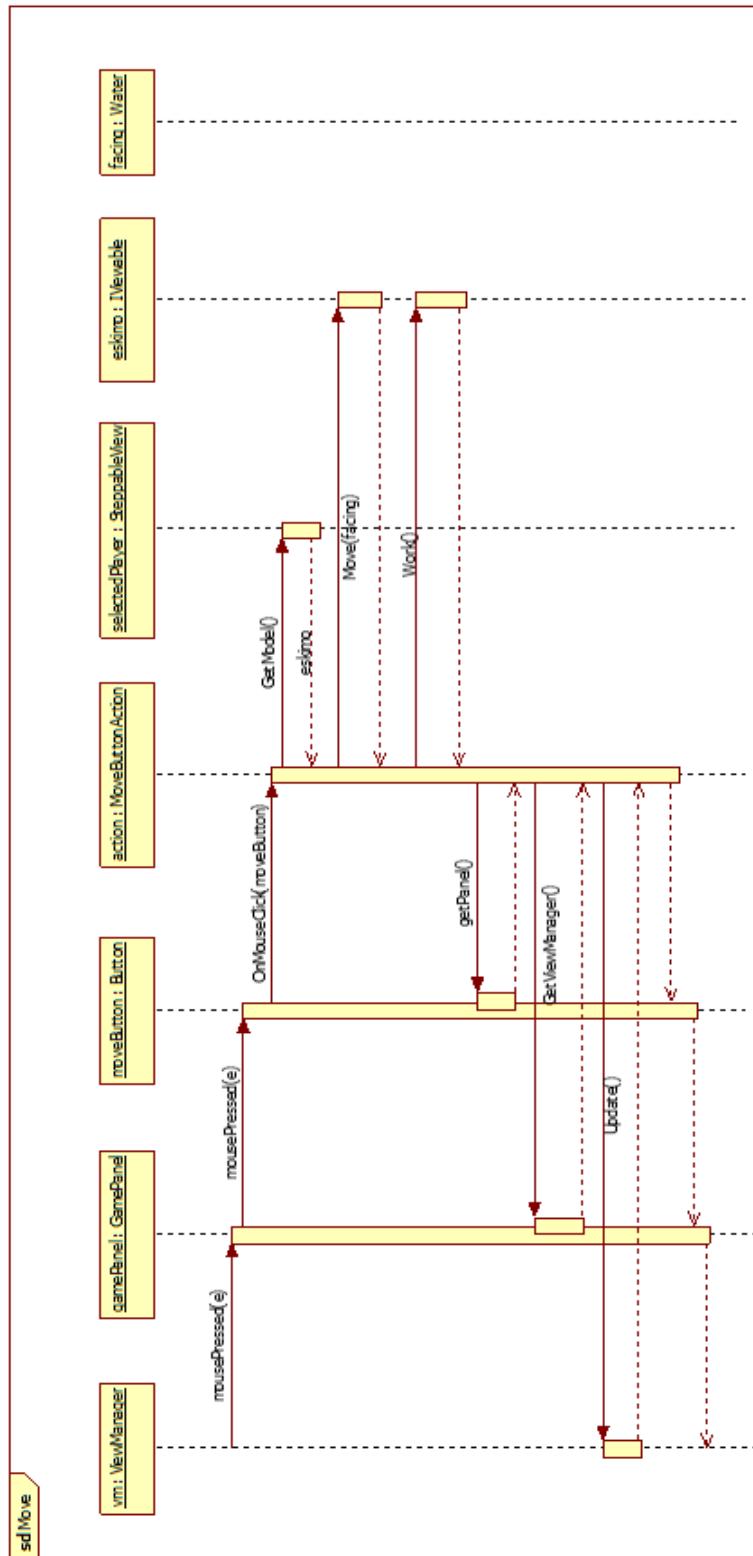


11.4.8 A játék futása. Egy körben minden játékosot addig dolgoztatunk amíg van “energiája”.

Ha már minden játékos végzett az adott körben GameUpdate() hívódik. (játékosok sorrend cseréje, játék végé kiértékelés, medvemozgatás, stormgenerálás)



11.4.9 Panel felépítése (pl. MenuPanel).



11.4.10 Egy játékos mozgatása. A játékpanel-en a felhasználó kattint a “Move” gombra. A jelenleg aktív játékos (selectedPlayer) fog átlépni arra a mezőre, amerre éppen néz (facing). Majd frissíti a képet.

Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.04.29 12:30	2 óra	Bodó Gaál Varga Nagy Rédai	Értekezlet.
2020.04.30 12:00	3 óra	Bodó Gaál Varga Nagy Rédai	Osztálydiagramok elkészítése
2020.04.30 19:00	2 óra	Bodó Gaál Varga Nagy Rédai	Osztálydiagram javítása
2020.05.01 10:00	4 óra	Bodó Gaál Varga Nagy Rédai	Osztálydiagram javítás, Szekvenciadiagramok elkészítése
2020.05.02 10:00	3 óra	Bodó Gaál Varga Nagy Rédai	Szekvenciadiagramok javítása
2020.05.02 18:00	4 óra	Bodó Gaál Varga Nagy Rédai	A játékhoz használt képek elkészítése
2020.05.03 12:00	3 óra	Bodó Gaál Varga Nagy Rédai	Dokumentáció elkészítése

13. Grafikus változat beadása

13.1 Fordítási és futtatási útmutató

13.1.1 Fájllista

Model .java

Fájl neve	Méret (byte)	Keletkezési ideje	Tartalom
Bear.java	1055	2020.05.15 16:24	A medve osztály.
BearEffect.java	600	2020.05.15 12:01	A medve effektus osztály.
Building.java	1525	2020.05.15 12:23	Az épületek ősosztálya.
Component.java	645	2020.05.15 12:23	A komponens osztály.
DivingSuit.java	788	2020.05.15 12:23	A búvárruha osztály.
Effect.java	889	2020.05.15 12:01	Az effektusok ősosztálya.
Eskimo.java	874	2020.05.15 17:48	Az eszkimó osztály.
FlareGun.java	676	2020.05.15 12:23	A jelzőpisztoly osztály.
Food.java	721	2020.05.15 16:55	Az élelem osztály.
GameManager.java	16807	2020.05.15 17:51	A GameManager osztály.
Ice.java	1439	2020.05.15 12:06	A jég osztály.
Iglu.java	818	2020.05.15 12:30	Az iglu osztály.
Item.java	1334	2020.05.15 12:33	A tárgyak ősosztálya.
ITMan.java	612	2020.05.15 12:35	Az informatikus osztály.
IViewable.java	408	2020.05.15 12:47	Az IViewable interfész.

Main.java	196	2020.05.15 16:48	A Main osztály.
NullBuilding.java	1182	2020.05.15 12:47	A NullBuilding (= levegő) osztály.
PackedTent.java	761	2020.05.15 12:08	A csomagolt sátor osztály.
Palesz.java	858	2020.05.15 12:47	A pálinka osztály.
PaleszEffect.java	714	2020.05.15 12:44	A pálinka effektus osztály.
Player.java	5309	2020.05.15 16:24	A játékosok ősosztálya.
Researcher.java	902	2020.05.15 12:08	A sarkkutató osztály.
Rope.java	742	2020.05.15 12:39	A kötél osztály.
RopeEffect.java	878	2020.05.15 12:08	A kötél effektus osztály.
Seal.java	677	2020.05.15 12:38	A fóka osztály.
Shovel.java	911	2020.05.15 12:38	Az ásó osztály.
Steppable.java	2308	2020.05.15 12:37	A Steppable osztály.
StormEffect.java	637	2020.05.15 12:08	A vihar effektus osztály.
Tent.java	1133	2020.05.15 12:33	A sátor osztály.
UnstableIce.java	1939	2020.05.15 12:32	Az instabil osztály.
Water.java	8143	2020.05.15 12:32	A vízosztály.

View .java

Fájl neve	Méret (byte)	Keletkezési ideje	Tartalom
Background.java	478	2020.05.15 12:10	A háttér osztály.
BuildingView.java	1411	2020.05.15 12:10	Az épület nézet osztály.
Button.java	1007	2020.05.15 12:10	A gomb nézet osztály.
FieldView.java	1223	2020.05.15 12:51	A terület nézet osztály.
GamePanel.java	14360	2020.05.15 16:24	A játékpanel nézet osztály.
IDrawable.java	378	2020.05.15 10:17	A rajzolható interface.
ImageStorage.java	18420	2020.05.15 16:24	A képtároló osztály.
ItemView.java	504	2020.05.15 12:09	Az item nézet osztály.
LosePanel.java	1071	2020.05.15 16:24	A losepanel osztály.
MenuPanel.java	3129	2020.05.15 12:15	A menü panel osztály.
MyFileFilter.java	1920	2020.05.15 12:30	A egyedi szűrő osztály.
MyPanel.java	4867	2020.05.15 12:08	A panel ősosztál.
SelectedPlayer.java	576	2020.05.15 16:24	A kiválasztott játékost megjelenítő osztály.
SteppableView.java	3627	2020.05.15 12:08	A steppable nézet osztály.
View.java	4641	2020.05.15 16:24	A nézet ősosztály osztály.
ViewManager.java	4810	2020.05.15 16:24	A nézetek menedzseléséért felelős osztály.
WinPanel.java	1123	2020.05.15 16:24	A győztes panel osztály.

Control .java

Fájl neve	Méret (byte)	Keletkezési ideje	Tartalom
BackToMenuAction.java	514	2020.05.15 12:20	A Vissza a menübe gomb action osztály.
DigButtonAction.java	1022	2020.05.15 12:20	Az ásó gomb action osztály.
DropItemButtonAction.java	1147	2020.05.15 12:19	Az item drop gomb action osztály.
EndTurnButtonAction.java	856	2020.05.15 12:19	A vége a körnek gomb action osztály.
ExitButtonAction.java	449	2020.05.15 11:05	A kilépés gomb action osztály.
FieldAction.java	1783	2020.05.15 11:04	A mező action osztály.
IMyAction.java	521	2020.05.15 11:00	Az IMyAction interface.
ItemAction.java	1917	2020.05.15 11:03	Az item action osztály.
LoadButtonAction.java	1230	2020.05.15 17:53	A betöltő gomb action osztály.
MoveButtonAction.java	1038	2020.05.15 10:58	A move gomb action osztály.
NewGameButtonAction.java	681	2020.05.15 13:04	Az új játék action osztály.
PickUpItemButtonAction.java	1155	2020.05.15 10:57	A pick up item gomb action osztály.
ResumeButtonAction.java	539	2020.05.15 11:01	A vissza a játékba gomb action osztály.
SaveButtonAction.java	1354	2020.05.15 17:53	A mentés gomb action osztály.
UseItemButtonAction.java	1148	2020.05.15 11:01	Az item használat action osztály.
UseSkillButtonAction.java	1000	2020.05.15 16:34	A skill használat gomb action osztály.

Model .class

Fájl neve	Méret (byte)	Keletkezési ideje	Tartalom
Bear.class	820	2020.05.15 19:03	A medve osztály.
BearEffect.class	514	2020.05.15 19:03	A beareffect osztály
Building.class	842	2020.05.15 19:03	A building osztály
Component.class	487	2020.05.15 19:03	A component osztály
DivingSuit.class	519	2020.05.15 19:03	A diving suit osztály
Effect.class	568	2020.05.15 19:03	Az effect osztály
Eskimo.class	753	2020.05.15 19:03	Az eskimo osztály
FlareGun.class	621	2020.05.15 19:03	A flaregun osztály
Food.class	529	2020.05.15 19:03	A food osztály
GameManager.class	12390	2020.05.15 19:03	A gamemanager osztály
Ice.class	1144	2020.05.15 19:03	Az ice osztály
Iglu.class	515	2020.05.15 19:03	Az iglu osztály
Item.class	864	2020.05.15 19:03	Az item osztály
ITMan.class	468	2020.05.15 19:03	Az ITman osztály
IViewable.class	199	2020.05.15 19:03	Az Iviewable interface
Main.class	526	2020.05.15 19:03	A main
NullBuilding.class	835	2020.05.15 19:03	A nullbuilding osztály

PackedTent.class	765	2020.05.15 19:03	A packedtent osztály
Palesz.class	677	2020.05.15 19:03	A palesz osztály
PaleszEffect.class	501	2020.05.15 19:03	A paleszeffect osztály
Player.class	3273	2020.05.15 19:03	A player osztály
Researcher.class	595	2020.05.15 19:03	A researcher osztály
Rope.class	633	2020.05.15 19:03	A rope osztály
RopeEffect.class	577	2020.05.15 19:03	A ropeeffect osztály
Seal.class	501	2020.05.15 19:03	A seal osztály
Shovel.class	569	2020.05.15 19:03	A shovel osztály
Steppable.class	1367	2020.05.15 19:03	A steppable osztály
StormEffect.class	517	2020.05.15 19:03	A stormeffect osztály
Tent.class	858	2020.05.15 19:03	A tent osztály
UnstableIce.class	1783	2020.05.15 19:03	Az unstable ice osztály
Water.class	5163	2020.05.15 19:03	A water osztály

View .class

Fájl neve	Méret (byte)	Keletkezési ideje	Tartalom
Background.class	289	2020.05.15 19:03	A háttér osztály.
BuildingView.class	1518	2020.05.15 19:03	Az épület nézet osztály.
Button.class	516	2020.05.15 19:03	A gomb nézet osztály.

FieldView.class	1243	2020.05.15 19:03	A terület nézet osztály.
GamePanel.class	9383	2020.05.15 19:03	A játékpanel nézet osztály.
IDrawable.class	165	2020.05.15 19:03	A rajzolható interface.
ImageStorage.class	11046	2020.05.15 19:03	A képtároló osztály.
ItemView.class	285	2020.05.15 19:03	Az item nézet osztály.
LosePanel.class	1176	2020.05.15 19:03	A losepanel osztály.
MenuPanel.class	2800	2020.05.15 19:03	A menü panel osztály.
MyFileFilter.class	1225	2020.05.15 19:03	A egyedi szűrő osztály.
MyPanel.class	3176	2020.05.15 19:03	A panel ősosztál.
SelectedPlayer.class	297	2020.05.15 19:03	A kiválasztott játékos megjelenítő osztály.
SteppableView.class	2335	2020.05.15 19:03	A steppable nézet osztály.
View.class	2452	2020.05.15 19:03	A nézet ősosztály osztály.
MyAction.class	649	2020.05.15 19:03	Inner class, action listener-ért felelős osztály. ViewManager-en belüli.
ViewManager.class	3600	2020.05.15 19:03	A nézetek menedzseléséért felelős osztály.
WinPanel.class	1173	2020.05.15 19:03	A győztes panel osztály.

Control .class

Fájl neve	Méret (byte)	Keletkezési ideje	Tartalom
BackToMenuAction.class	685	2020.05.15 19:03	A Vissza a menübe gomb action osztály.
DigButtonAction.class	1079	2020.05.15 19:03	Az ásó gomb action osztály.
DropItemButtonAction.class	1225	2020.05.15 19:03	Az item drop gomb action osztály.
EndTurnButtonAction.class	1017	2020.05.15 19:03	A vége a körnek gomb action osztály.
ExitButtonAction.class	462	2020.05.15 19:03	A kilépés gomb action osztály.
FieldAction.class	1463	2020.05.15 19:03	A mező action osztály.
IMyAction.class	197	2020.05.15 19:03	Az IMyAction interface.
ItemAction.class	1706	2020.05.15 19:03	Az item action osztály.
LoadButtonAction.class	1890	2020.05.15 19:03	A betöltő gomb action osztály.
MoveButtonAction.class	1238	2020.05.15 19:03	A move gomb action osztály.
NewGameButtonAction.clas s	879	2020.05.15 19:03	Az új játék action osztály.
PickUpItemButtonAction.cl ass	1231	2020.05.15 19:03	A pick up item gomb action osztály.
ResumeButtonAction.class	689	2020.05.15 19:03	A vissza a játékba gomb action osztály.
SaveButtonAction.class	1909	2020.05.15 19:03	A mentés gomb action osztály.
UseButtonItemAction.class	1220	2020.05.15 19:03	Az item használat action osztály.
UseSkillButtonAction.class	1087	2020.05.15 19:03	A skill használat gomb action osztály.

A grafikus megjelenítéshez használt képek és alappálya

Fájl neve	Méret (byte)	Keletkezési ideje	Tartalom
gamebg.png	10120	2020.05.12 17:19	A játék háttérképe
infoPanel.png	7158	2020.05.12 17:19	A játékon belüli oldalsó menü képe
losebg.png	87184	2020.05.12 17:19	A vesztés képernyő háttérképe
menubg.png	58022	2020.05.12 17:19	A főmenü háttérképe
selectedFieldFrame.png	1258	2020.05.12 17:27	A kiválasztott mező keretének képe
selectedItemFrame.png	656	2020.05.12 17:19	A kiválasztott tárgy keretének képe
winbg.png	76604	2020.05.12 17:19	A győztes képernyő háttérképe
polarbear0.png	1318	2020.05.12 17:19	A jegesmedve képe
polarbear1.png	1349	2020.05.12 17:19	A jegesmedve vízben lévő képe
iglu.png	1584	2020.05.12 17:19	Az iglu képe
tent.png	1790	2020.05.12 17:19	A sátor képe
backToMenuButton.png	713	2020.05.12 17:19	A “Menu” gomb képe
0digNormal.png	7575	2020.05.12 17:19	Az aktív “Dig” gomb képe
1digDisabled.png	7106	2020.05.12 17:19	Az inaktív “Dig” gomb képe
0dropNormal.png	7780	2020.05.12 17:19	Az aktív “Drop” gomb képe
1dropDisabled.png	7352	2020.05.12 17:19	Az inaktív “Drop” gomb képe
0endTurnNormal.png	7293	2020.05.12 17:19	Az aktív “End Turn” gomb képe

1endTurnDisabled.png	6763	2020.05.12 17:19	Az inaktív “End Turn” gomb képe
exitButton.png	5005	2020.05.12 17:19	Az aktív “Exit” gomb képe
loadButton.png	4951	2020.05.12 17:19	Az inaktív “Exit” gomb képe
0moveNormal.png	7958	2020.05.12 17:19	Az aktív “Move” gomb képe
1moveDisabled.png	7393	2020.05.12 17:19	Az inaktív “Move” gomb képe
newGameButton.png	6201	2020.05.15 12:22	A “New Game” gomb képe
0pickUpNormal.png	8202	2020.05.12 17:19	Az aktív “Pick Up” gomb képe
1pickUpDisabled.png	7645	2020.05.12 17:19	Az inaktív “Pick Up” gomb képe
resumeButton0.png	5319	2020.05.15 12:22	Az aktív “Resume” gomb képe
resumeButton1.png	4940	2020.05.15 12:22	Az inaktív “Resume” gomb képe
saveButton.png	5062	2020.05.12 17:19	Az aktív “Save” gomb képe
saveButtonon1.png	4981	2020.05.15 12:23	Az inaktív “Save” gomb képe
0useItemNormal.png	8664	2020.05.12 17:19	Az aktív “Use Item” gomb képe
1useItemDisabled.png	7925	2020.05.12 17:19	Az inaktív “Use Item” gomb képe
0useSkillNormal.png	8772	2020.05.12 17:19	Az aktív “Use Skill” gomb képe
1useSkillDisabled.png	8148	2020.05.12 17:19	Az inaktív “Use Skill” gomb képe
capacity0.png	741	2020.05.12 17:19	A 0 értékű kapacitásjelző képe
capacity1.png	719	2020.05.12 17:19	Az 1 értékű kapacitásjelző képe
capacity2.png	728	2020.05.12 17:19	A 2 értékű kapacitásjelző képe

capacity3.png	729	2020.05.12 17:19	A 3 értékű kapacitásjelző képe
capacity4.png	719	2020.05.12 17:19	A 4 értékű kapacitásjelző képe
capacityInfinite.png	750	2020.05.12 17:19	A végtelen értékű kapacitásjelző képe
capacityUnknown.png	725	2020.05.12 17:19	Az ismeretlen értékű kapacitásjelző képe
component0.png	1810	2020.05.12 17:19	A patron komponens képe
component1.png	1765	2020.05.12 17:19	A jelzőfény komponens képe
component2.png	1738	2020.05.12 17:19	A pisztoly komponens képe
empty128x128.png	754	2020.05.12 17:19	A mezők üres képe
empty64x64.png	617	2020.05.12 17:19	A tárgyak/játékosok üres képe
ice .png	2299	2020.05.12 17:19	A 0 hórétegű jég képe
iceWith1Snow.png	12464	2020.05.12 17:19	Az 1 hórétegű jég képe
iceWith2Snow.png	13132	2020.05.12 17:19	A 2 hórétegű jég képe
iceWith3Snow.png	14109	2020.05.12 17:19	A 3 hórétegű jég képe
iceWith4Snow.png	12412	2020.05.12 17:19	A 4 hórétegű jég képe
iceWith5Snow.png	1977	2020.05.12 17:19	Az 5 hórétegű jég képe
divingSuit.png	1659	2020.05.12 17:19	A búvárruha tárgy képe
food.png	1800	2020.05.12 17:19	Az élelem tárgy képe
packedTent.png	1745	2020.05.12 17:19	A csomagolt sátor tárgy képe
palesz.png	1735	2020.05.12 17:19	A pálinka tárgy képe

rope.png	1662	2020.05.12 17:19	A kötél képe
shovel.png	1769	2020.05.12 17:19	Az ásó tárgy képe
mapBase.data	12388	2020.05.15 17:51	A pálya háttérképe
0of4.png	3107	2020.05.12 17:19	0 hátralévő ember képe
1of4.png	2761	2020.05.12 17:19	1 hátralévő ember képe
2of4.png	2304	2020.05.12 17:19	2 hátralévő ember képe
3of4.png	1866	2020.05.12 17:19	3 hátralévő ember képe
4of4.png	1360	2020.05.12 17:19	4 hátralévő ember képe képe
blue0.png	608	2020.05.12 17:19	Kék 0-ás szám képe.
blue1.png	569	2020.05.12 17:19	Kék 1-es szám képe.
blue2.png	617	2020.05.12 17:19	Kék 2-es szám képe.
blue3.png	596	2020.05.12 17:19	Kék 3-as szám képe.
blue4.png	613	2020.05.12 17:19	Kék 4-es szám képe.
blue5.png	589	2020.05.12 17:19	Kék 5-ös szám képe.
blue6.png	601	2020.05.12 17:19	Kék 6-os szám képe.
blue7.png	593	2020.05.12 17:19	Kék 7-es szám képe.
blue8.png	581	2020.05.12 17:19	Kék 8-as szám képe.
blue9.png	596	2020.05.12 17:19	Kék 9-es szám képe.
lightblue0.png	608	2020.05.12 17:19	Világoskék 0-s szám képe.

lightblue1.png	573	2020.05.12 17:19	Világoskék 1-es szám képe.
lightblue2.png	618	2020.05.12 17:19	Világoskék 2-es szám képe.
lightblue3.png	593	2020.05.12 17:19	Világoskék 3-as szám képe.
lightblue4.png	611	2020.05.12 17:19	Világoskék 4-es szám képe.
lightblue5.png	588	2020.05.12 17:19	Világoskék 5-ös szám képe.
lightblue6.png	601	2020.05.12 17:19	Világoskék 6-os szám képe.
lightblue7.png	596	2020.05.12 17:19	Világoskék 7-es szám képe.
lightblue8.png	581	2020.05.12 17:19	Világoskék 8-as szám képe.
lightblue9.png	598	2020.05.12 17:19	Világoskék 9-es szám képe.
lightbluePercent.png	621	2020.05.12 17:19	Világoskék százalékjel képe.
red0.png	596	2020.05.12 17:19	Piros 0-ás szám képe.
red1.png	2315	2020.05.12 17:19	Piros 1-es szám képe.
red2.png	599	2020.05.12 17:19	Piros 2-es szám képe.
red3.png	577	2020.05.12 17:19	Piros 3-as szám képe.
red4.png	596	2020.05.12 17:19	Piros 4-es szám képe.
red5.png	585	2020.05.12 17:19	Piros 5-ös szám képe.
red6.png	582	2020.05.12 17:19	Piros 6-os szám képe.
red7.png	581	2020.05.12 17:19	Piros 7-es szám képe.
red8.png	578	2020.05.12 17:19	Piros 8-as szám képe.

red9.png	588	2020.05.12 17:19	Piros 9-es szám képe.
eskimo0.png	970	2020.05.12 17:19	Az Eskimo alapértelmezett képe
eskimo1.png	969	2020.05.12 17:19	Az Eskimo bűvárruhás képe
eskimo2.png	1135	2020.05.12 17:19	Az Eskimo vízbe esett képe
eskimo3.png	1146	2020.05.12 17:19	Az Eskimo vízbe esett (bűvárruhában) képe
itman0.png	1047	2020.05.12 17:19	Az ITMan alapértelmezett képe
itman1.png	956	2020.05.12 17:19	Az ITMan bűvárruhás képe
itman2.png	1243	2020.05.12 17:19	Az ITMan vízbe esett képe
itman3.png	1133	2020.05.12 17:19	Az ITMan vízbe esett (bűvárruhában) képe
researcher0.png	1043	2020.05.12 17:19	A researcher alapértelmezett képe
researcher1.png	1145	2020.05.12 17:19	A researcher bűvárruhás képe
researcher2.png	1179	2020.05.12 17:19	A researcher vízbe esett képe
researcher3.png	1128	2020.05.12 17:19	A researcher vízbe esett (bűvárruhában) képe
seal0.png	3614	2020.05.12 17:19	A fóka alapértelmezett képe
seal1.png	1367	2020.05.12 17:19	A fóka vízbe esett képe

13.1.2 Fordítás és telepítés

A fordításhoz cmd-ben vagy powershellben el kell navigálni a projekt könyvtárához majd a következő parancsokat kell kiadni:

```
set path=C:\Program Files\Java\jdk-11.0.7\bin
set JAVA_HOME=c:\Program Files\Java\jdk-11.0.7
javac aranyaszok/*.java aranyaszok/listener/*.java aranyaszok/gui/*.java
jar cfm aranyaszok.jar MANIFEST.mf aranyaszok/*.class /aranyaszok/gui/*.class
aranyaszok/listener/*.class
```

13.1.3 Futtatás

Futtatáshoz a létrejött .jar fájlhoz kell navigálni és a *java -jar aranyaszok.jar* parancsal futtatható az állomány. (vagy csak szimplán dupla kíkkel file explorer-ben)

A követelmények között szerepelt, hogy tisztáznunk kell a billentyűzetkiosztást is, ahogy az a 11. dokumentumban szerepelt a programunk csak egérrel lesz irányítható, billentyűlenyomások kezelése továbbra sincs megvalósítva.

13.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Bodó István	LU93RR	20%
Gaál Gergely	GOXEHI	20%
Varga Levente	JIE1OY	20%
Nagy Márton	DMMP94	20%
Rédai Attila	LYYOST	20%

13.3 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.05.05 12:00	6 óra	Bodó Gaál Varga Nagy Rédai	Kódolás
2020.05.06 13:30	5 óra	Bodó Gaál Varga Nagy Rédai	Kódolás, Bugfix
2020.05.08 12:00	6 óra	Bodó Gaál Varga Nagy Rédai	Kódolás, Tesztelés
2020.05.10 15:00	5,5 óra	Bodó Gaál Varga Nagy Rédai	Kódolás, Bugfix
2020.05.11 12:00	7 óra	Bodó Gaál Varga Nagy Rédai	Kódolás, Tesztelés, Bugfix
2020.05.12 13:00	6 óra	Bodó Gaál Varga Nagy Rédai	Kódolás, Bugfix
2020.05.13 10:00	8 óra	Bodó Gaál Varga Nagy Rédai	Bugfix, Tesztelés
2020.05.15 12:00	1 óra	Bodó Gaál Varga Nagy Rédai	Dokumentáció és futási hibák javítása

14. Összefoglalás

14.1 A projektre fordított összes munkaidő

Tag neve	Munkaidő (óra)
Bodó István	252
Gaál Gergely	252
Varga Levente	252
Nagy Márton	252
Rédai Attila	252
Összesen	1260

- **A feltöltött programok forrássorainak száma**

Fázis	Kódsorok száma
Szkeleton	2392
Prototípus	2737
Grafikus változat	5438
Összesen	10567

14.2 Projekt összegzés

14.2.1 Mit tanultak a projektből konkrétan és általában?

Általában tapasztalatot szerezhettünk csoportmunkával kapcsolatba, konkrétan pedig elmélyíthettük a tudásunkat a SWING-gel és az objektumorientált világénezettel kapcsolatban.

14.2.2 Mi volt a legnehezebb és a legkönnyebb?

A legkönnyebb talán a kódolás, habár több hibára csak eközben derült fény. A legnehezebb pedig az, hogy öt embernek kellett összeegyeztetni a gondolatmenetét, ami a kezdetben nem volt egyszerű, de így legalább szereztünk tapasztalatot csapatmunka terén is.

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

Nem teljesen. (lásd lentebb)

14.2.4 Ha nem, akkor hol okozott ez nehézséget?

Mi sokkal több időt fordítottunk a grafikai részre, mint azt a pontszámok aránya mutatja. Gyakorlatilag a programot a kétszeresére kellett bővíteni a sok új grafikus osztály miatt, és a rajzolás is sok időt elvitt. Lehet, hogy túlzásba estünk a grafikus rész kidolgozásában, de mi így tudtuk csak létrehozni azt a “minőséget”, amivel elégedettek vagyunk.

14.2.5 Milyen változtatási javaslatuk van?

A grafikus rész szempontjából fontos lett volna, ha idővel tisztázódik a pálya alakja. Az első 8 héten teljesen általános, gráf-szerű pályákkal dolgoztunk, de mikor elérünk a grafikus részhez, ennél több információra lett volna szükségünk ahhoz, hogy a pályát meg tudjuk rajzolni. Legalább egy fokszám limitációt szerettünk volna, mivel ha általában gráf alakú pályákkal dolgozunk, akkor a mezőknek korlátlan számú szomszédjuk lehet, és ezt nem tudjuk kirajzolni. Mi ezért limitáltuk a fokszámot 4-re, majd kiterítettük a gráfot négyzetrácsként az egyszerű rajzolás miatt.

14.2.6 Milyen feladatot ajánlanának a projektre?

Mivel a jelenlegi feladat és a pár példa, amit az elmúlt évekből láttunk arra épít, hogy egy pályán mozognak a karakterek / mozgatnak valamit, és bizonyos objektumokat gyűjtenek / használnak, valamint előfordulnak véletlenszerű külső tényezők is, ezért egy ezeket a típusú jellemzőket megvalósítandó játék lehet, ha a pálya egy farm lenne. A farmon változó, de akár összevonható területeken különböző növényeket lehetne ültetni, majd a magokat újra begyűjteni. A játék közben egy-egy jágeső vagy más tényező letarolhatna egy-egy területet így nehezítve a játékosok dolgát. A játékot akkor nyernék meg, ha minden fajta növényből sikerülne egy bizonyos mennyiséget begyűjteni.

14.2.7 Egyéb kritika és javaslat

Nincs