

## **2. Feladat: Követelmény, projekt, funkcionalitás - beadás**

**14 – the\_minesweepers**

Konzulens:  
**Fetter László**

### **Csapattagok**

Iuhos Zoltán	J8LOUL	zoltan.iuhos@gmail.com
Kaposvári Márk Martin	RAYVF8	xmarko45@gmail.com
Sulyok Botond	BM2BBM	botond.sulyok99@gmail.com
Labancz Tamás Dominik	A9HZBF	labancz.tamas@gmail.com
Traxler Bálint	IXNOYB	traxlerbalint@gmail.com

2020.02.24.

## 2. Követelmény, projekt, funkcionalitás

### 2.1 Bevezetés

#### 2.1.1 Cél

Jelen dokumentum célja a készítendő szoftver alapszintű bemutatása, funkcióinak ismertetése.

#### 2.1.2 Szakterület

A készülő szoftver egy játékprogram lesz, melynek célja a felhasználók szórakoztatása. Nagyszerű szórakozást nyújt egyénileg és társasággal együtt is.

#### 2.1.3 Definíciók, rövidítések

JRE - Java Runtime Environment

#### 2.1.4 Hivatkozások

Szoftvertechnológia jegyzet (<https://www.iit.bme.hu/targyak/BMEVIIIAB01>)

A tárgy weboldalán található feladatkiírás, valamint a készítendő dokumentum mintája.

<https://www.iit.bme.hu/targyak/BMEVIIIAB02/feladat> (utolsó letöltés: 2020.02.19.)

<https://www.iit.bme.hu/file/810/dokumentum-sablonok> (utolsó letöltés: 2020.02.19.)

#### 2.1.5 Összefoglalás

A továbbiakban részletezésre kerülnek a szoftvertől elvárt funkciók és követelmények. Ezen kívül a megvalósítás tervezett módját, lépésein is részletezzük.

## 2.2 Áttekintés

### 2.2.1 Általános áttekintés

A felhasználók egy grafikus felület segítségével tájékozódhatnak a játék jelenlegi állapotáról. Indításkor a felhasználók a kezdőképernyőn találják magukat, ahonnan indíthatják az első játékukat. A szoftvernek a billentyűzet segítségével adhatnak ki parancsokat. Az adattárolás a játékot futtató felhasználó háttértárrán valósul meg. A program nem igényel internet kapcsolatot.

### 2.2.2 Funkciók

A feladat egy Jégmező nevű játék készítése, melynek lényege, hogy a játékosok egy csapatként megmeneküljenek egy tengerrel körülött jégmezőről. A csapat eszkimókból és sarkkutatókból áll. A játék elkezdéséhez legalább 3 játékos szükséges, valamint egy játékos akár több karaktert is irányíthat. A szereplőcsoportok számoságának eloszlására nincs megkötés, de hasznos lehet közel azonosra választani.

A jégmező kisebb, azonos méretű jégtáblából épül fel, melyek összeérnek, de eltérően nézhetnek ki, valamint viselkedésük sem teljesen azonos. A jégmező tehát jégtáblák összessége, amit tenger határol. Vannak stabil jégtáblák, amelyeken akárhány szereplő állhat, és vannak instabil jégtáblák, amik adott létszám felett átfordulnak és ilyenkor a rajtuk állók a vízbe esnek. A vízbe esett szereplők azonnal megfagynak és meghalnak, hacsak nem viselnek

búvárruhát, vagy nem áll mellettük egy köteles csapattársuk a szomszéd táblán, aki ilyen esetben azonnal kimenekíti, megmentve az életét.

A jégtáblákat a játék kezdetén eltérő mennyiségű hó borítja, de vannak olyan táblák is, ahol nincs jég, csak hó. Aki egy ilyenre rálép, az vízbe esik, hiszen nincs alatta jég, ami megtartaná. A vízbe esésre az előző szabály vonatkozik ebben az esetben is.

Minden jégtáblában található legfeljebb egy befagyott tárgy a következők közül: lapát, kötél, búvárruha, élelem, pisztoly, jelzőfény, patron. Befagyott tárgyat csak akkor lehet meglátni és kiásni, ha a jégtábla tiszta, nem borítja hó. A kiásás egy munkát igényel, és a tárgy ahhoz kerül, aki kiásta. minden játékos tetszőleges számú tárgyat hordhat magánál.

A szereplők körökre osztva tudnak tevékenységeket végezni. minden szereplő egy körben legfeljebb 4 egységyi munkát végezhet. Ilyen munka lehet: a jégtáblán levő egységyi mennyiségű hó eltakarítása (A lapáttal két egységyi hó takarítható el egy munkaráfordítással); egy szomszédos jégtáblára való lépés (a táblák között nem lehet átlósan mozogni); egy tárgy kiásása és felvétele; egy tárgy használata; a jelzőpisztoly összerakása. Kivétel: egy vízbeesett csapattag kimentése automatikus, nem használ munkát. A tárgyakat nem lehet átadni más játékosnak, végig annál marad, aki felvette. minden játékos dönthet úgy, hogy nem használja fel mind a négy egységyi munkáját az adott körben. Ezt a szándékát jelezheti, ekkor a következő szereplő következik.

A jégmezőn időnként ( minden körben azonos valószínűséggel, de csak a körök elején) feltámad a hóvíhar, és néhány érintett jégtáblát újabb adag, egy egységyi friss hóval borít be. A hóvíhar előtti körben jelez a játék, hogy fel lehessen készülni. A jégtáblákon legfeljebb 3 réteg hó lehet. A hóvíhar véletlenszerűen érinti a táblákat. Ha ezen jégtáblák valamelyikén áll valaki, annak a testhője egységyivel csökken. Az eszkimóknak a játék elején 5 egység testhője van, a sarkutatóknak csak 4. Egy élelem eggyel növeli a testhőt. Az éleelmet azonnal meg kell enni, mihelyt valaki felvette. Élelem fogyasztásával nem lehet a kezdeti testhő értéke fölé menni, maximális testhővel rendelkező szereplő élelemfogyasztása esetén nem növekszik a testhő, de az élelem eltűnik.

A szereplők jégtábláról-jégtáblára haladnak képességeiknek megfelelően. A sarkutató meg tudja nézni, hogy az a jégtábla, amire lépne, hány embert bír el (a luk egyet sem). Az eszkimó tud iglut építeni, amiben átvésselhetők a hóviharok. Egy igluba egy ember fér bele, és egy iglu egy hóvihart él túl. Egy-egy képesség alkalmazása egy-egy munkát jelent. Csapatjáték lévén az igluba mehet sarkutató, valamint a sarkutató által felderített mező tulajdonsága a csapat közös tudása lesz.

A játék célja egy jelzőrakéta alkatrészeinek (pisztoly, jelzőfény, patron) megtalálása. Az alkatrészek is a jégbe vannak fagyva. Ha ezeket a csapat összegyűjtí és ugyanarra a jégtáblára viszi, akkor egy munka felhasználásával összeszerelhetik és elsüthetik, amivel megnyerik a játékot. Ehhez azonban mindenkiüknek ugyanott kell állniuk. Ha valaki menet közben meghal (vízbe esve megfullad vagy elfogy a testhője és kihűl), akkor a játék véget ér, a csapat vesztett.

### 2.2.3 Felhasználók

A játékprogramot minden korosztály élvezheti. A szabályok ismeretén kívül más előismeretre nincs szükség a használatához. A játékot egyszerre több játékos is játszhatja.

### 2.2.4 Korlátozások

A játék csak ablakos üzemmódban futtatható.

### 2.2.5 Feltételezések, kapcsolatok

A szoftvertechnológia jegyzet segített feleleveníteni szoftvertervezési és projektmenedzsmenti ismereteinket és a diagram megtervezésében is segített.. A tárgyhonlapon lévő kiírás és minta alapján oldottuk meg a feladatot.

## 2.3 Követelmények

### 2.3.1 Funkcionális követelmények

Az ono sító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
1	A játékot legalább 3 karakternek kell játszania.	bemutatás	alapvető	A játék informális leírása.	Karakterek beállítása	
2	A jégmezőt tenger veszi körül.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégmező megtekintése	
3	A karaktereknek egy része lehet eszkimó.	bemutatás	alapvető	A játék informális leírása	Karakterek beállítása	
4	A karaktereknek egy része lehet sarkkutató.	bemutatás	alapvető	A játék informális leírása	Karakterek beállítása	
5	Egy játékkörben minden karakternek kell hogy legyen játékos köre.	bemutatás és kiértékelés	alapvető	Csapat	Játékkör átadása	
6	Egy játékkörben a karakterek játékos körei fix sorrendben következnek.	bemutatás és kiértékelés	opcionális	Csapat	Játékkör átadása	
7	A jégmező jégtáblából áll.	bemutatás és kiértékelés	fontos	A játék informális leírása	Jégmező megtekintése	
8	Vannak stabil jégtáblák.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégmező megtekintése	

9	A stabil jégtáblák akárhány karaktert elbírnak.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégtáblára lépés	
10	Vannak instabil jégtáblák.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégmező megtekintése	
11	Az instabil jégtáblák csak meghatározott számú karaktert bírnak el.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégtáblára lépés	
12	Ha egy instabil jégtáblán annak teherbíró képességét meghaladó számú karakter tartózkodik, az átkelik, és a rajta álló karakterek vízbe esnek.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégtáblára lépés	
13	A jégtáblákat a játék kezdetén hó borítja.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégmező megtekintése	
14	A jégtáblákat borító hó a játék elején lehet jégtáblánként eltérő mennyiségekű.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégmező megtekintése	
15	A jégtáblák között lehetnek hóval fedett lyukak.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégmező megtekintése	
16	Ha egy karakter rálép egy ilyen hóval fedett lyukra, akkor az vízbe esik.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégtáblára lépés	
17	A jégtáblákba tárgyak lehetnek belefagyva.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégmező megtekintése	
18	A tárgyak csak akkor áshatóak ki, vagy láthatóak, ha a tartalmazó jégtáblát nem borítja hó.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Tárgy kiásása	
19	Ha egy tárgyat kiás egy karakter, azt felveszi a tárgyai közé.	bemutatás és kiértékelés	alapvető	Csapat	Tárgy kiásása	Kivéve az élelmet, azt azonnal felhasználja.

20	Egyfélé tárgyból pontosan egy karakternél egy lehet.	bemutatás és kiértékelés	fontos	Csapat	Tárgy kiásása	
21	Ha egy karakter rendelkezik egy tárggyal, akkor ugyanazt a tárgyat nem áshatja ki.	bemutatás és kiértékelés	alapvető	Csapat	Tárgy kiásása	
22	Az élelem felhasználása a testhő értékét eggyel növeli.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Tárgy kiásása	
23	Ha éleelmet ás ki egy karakter, azt azonnal felhasználja.	bemutatás és kiértékelés	fontos	Csapat	Tárgy kiásása	
24	Az élelem elfogyasztásával nem lehet a karakter maximum testhője fölé emelni azt.	bemutatás és kiértékelés	opcionális	Csapat	Tárgy kiásása	
25	A játékban létezik lapát tárgy.	bemutatás	alapvető	A játék informális leírása	Jégmező megtekintése	
26	A lapáttal két egységnyi hó takarítható el egy munkaráfordítással.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Hó eltakarítása	
27	A játékban létezik bűvárruha tárgy.	bemutatás	alapvető	A játék informális leírása	Jégmező megtekintése	
28	Ha egy karakter vízbe esik, és rendelkezik bűvárruhával, akkor az nem fulladhat meg, és halhat meg azonnal emiatt.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégtábla terheltsége	
29	A vízbe esett bűvárruhás karakternek lehetősége van átúsznia egy szomszédos tengeri mezőre.	bemutatás és kiértékelés	fontos	Csapat	Jégtáblára lépés	
30	Ha a vízbe esett bűvárruhás karakter átúszik egy másik tengeri mezőre, az egy munkába kerül.	bemutatás és kiértékelés	opcionális	Csapat	Jégtáblára lépés	

31	Ha a vízbe esik egy bűvárruhás karakter, akkor azt a karaktert egy közelí kötéssel rendelkező karakter nem fogja kihúzni.	bemutatás és kiértékelés	opcionális	Csapat	Jégtáblára lépés	
32	A bűvárruhás karakter csak azután mozoghat más tengeri mezőre, hogy vízbe esett, tehát önmagától nem dönthet úgy, hogy egy jégtábláról vízbe mászik.	bemutatás és kiértékelés	opcionális	Csapat	Jégtáblára lépés	
33	Ha a bűvárruhát használó vízbe esett karakter egy tengeri mezőről egy jégtáblára mozdul el, az egy munkát vesz igénybe.	bemutatás és kiértékelés	opcionális	Csapat	Jégtáblára lépés	
34	A bűvárruha használata után (miután elhagyta a vizet) tönkremehet, és eltűnhet a karakter tárgyai közül.	bemutatás és kiértékelés	opcionális	Csapat	Jégtáblára lépés	Nyilván miután eltűnt, a karakter áshat ki újat.
35	A játékban létezik kötel tárgy.	bemutatás	alapvető	A játék informális leírása	Jégmező megtekintése	
36	Ha egy kötéssel rendelkező karakterrel szomszédos táblán egy másik karakter vízbe esik, akkor a kötéssel rendelkező karakter kimenti azt.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégtáblára lépés	
37	A kötéssel kihúzott karakter ugyanarra a mezőre kerül, mint amin az öt kihúzó karakter áll.	bemutatás és kiértékelés	fontos	Csapat	Jégtáblára lépés	
38	Egy karakter a játékos körében maximum 4 munkát hajthat végre.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Játékkör átadása	
39	Egy karakter a játékos körében nem muszáj mind a négy 4	bemutatás és kiértékelés	opcionális	Csapat	Játékkör átadása	

	munkáját felhasználnia, dönthet úgy, hogy befejezi a játékos körét.					
40	Elmozdulás másik jégtáblára egy munkát igényel.	bemutatás és kiértékelés	fontos	A játék informális leírása	Jégtáblára lépés	
41	Egy karakter csak a vele szomszédos jégtáblára mozdulhat el.	bemutatás	fontos	Csapat	Jégtáblára lépés	
42	Egységes hó eltakarítása egy munkát igényel.	bemutatás és kiértékelés	fontos	A játék informális leírása	Hó eltakarítása	
43	Egy tárgy felvétele és kiásása egy munkát igényel.	bemutatás és kiértékelés	fontos	A játék informális leírása	Tárgy kiásása	
44	Az eszkimó karaktereknek maximum 5 testhője lehet.	bemutatás	fontos	Csapat	Karakterek beállítása	
45	A sarkkutató karaktereknek maximum 4 testhője lehet.	bemutatás	fontos	Csapat	Karakterek beállítása	
46	Minden karakter a játék elején maximum testhővel kezd. (eszkimó:5, sarkkutató:4)	bemutatás és kiértékelés	alapvető	Csapat	Karakterek beállítása	
47	A jégmezőn feltámadhat a hóvihar.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Hóvihar vezérlés	
48	A hóvihar a játékkörök között támad fel.	bemutatás és kiértékelés	opcionális	Csapat	Hóvihar vezérlés	
49	Ha egy játékkört hóvihar fog követni, akkor annak a játékkörnek az elején ezt jelezni kell a játékosoknak.	bemutatás és kiértékelés	opcionális	Csapat	Hóvihar vezérlés, Jégmező megtekintése	
50	A hóvihar csökkenti minden karakter testhőjét eggyel, aki karakter nincs sehogy a hóvihartól védve.	kiértékelés	alapvető	A játék informális leírása	Hóvihar vezérlés	Ilyen védeottság például az igluban való

						tartózkodá s.
51	A karakterek a szereplőcsoporthoz megfelelő képességekkel és tulajdonságokkal rendelkeznek.	bemutatás	alapvető	A játék informális leírása	Karakterek beállítása	
52	Egy képesség használata 1 munkába kerül.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Karakterek beállítása	
53	Az eszkimó karakterek rendelkeznek az iglu építés képességevel.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Iglu építése	
54	Iglut csak hóval nem fedett, tárgyat nem tartalmazó jégtáblára lehet felépíteni.	bemutatás és kiértékelés	fontos	Csapat	Iglu építése	
55	Egy igluban maximum 1 karakter tartózkodhat.	bemutatás és kiértékelés	opcionális	Csapat	Iglu építése	
56	Az eszkimó karakterek iglut építeni csak azon a jégtáblán tudnak, amin tartózkodnak.	bemutatás	alapvető	Csapat	Iglu építése	
57	Az igluban tartózkodó karakterek testhője nem csökken a hóvihar miatt.	kiértékelés	alapvető	A játék informális leírása	Iglu építése, Hóvihar vezérlés	
58	A hóvihar összedönti a jégmezőn álló összes iglut.	bemutatás és kiértékelés	opcionális	Csapat	Hóvihar vezérlés	
59	A sarkutató karakterek rendelkeznek az jégtáblák teherbíró képességének vizsgálatával.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégtábla terheltsége	
60	A jégtáblák teherbíró képességének vizsgálata képesség segítségével meghatározható, hogy mekkora a vizsgált jégtábla teherbíró képessége.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégtábla terheltsége	

61	A jégtáblák teherbíró képességének vizsgálata képesség hatása maradandó, tehát később is visszaolvasható az eredmény.	bemutatás és kiértékelés	opcionális	Csapat	Jégtábla terheltsége	
62	A jégtáblák teherbíró képességének vizsgálata képesség többször is alkalmazható ugyanazon jégtáblára.	bemutatás és kiértékelés	opcionális	Csapat	Jégtábla terheltsége	Bár mivel az eredmény maradandó , ez nem egy taktikus döntés.
63	A jégtáblák teherbíró képességének vizsgálata képesség csak a sarkkutató karakter alatt található jégtáblán, illetve a szomszédos jégtáblákon alkalmazható.	bemutatás	fontos	Csapat	Jégtábla terheltsége	
64	Egy sarkkutató karakter egy játékban maximum egy meghatározott számszor használhatja a jégtáblák teherbíró képességének vizsgálata képességét.	bemutatás és kiértékelés	opcionális	Csapat	Jégtábla terheltsége	
65	A jelzőrakéta három kiátható tárgyból áll: patron, jelzőfény, pisztoly.	bemutatás	alapvető	A játék informális leírása	Jelzőrakéta összerakása	
66	Jelzőrakéta bármely darabja megtalálható lehet egy jégmező több jégtábláján is.	bemutatás és kiértékelés	opcionális	Csapat	Jégmező megtekintése	
67	A jelzőrakétai darabjait hordó karakterek ahhoz, hogy összerakják a jelzőrakétát, egy jégtáblán kell tartózkodniuk.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégmező megtekintése , Jelzőrakéta összerakása	

68	A jelzőrakéta darabjai stabil jégtáblából ásható ki.	bemutatás	fontos	Csapat	Tárgy kiásása	
69	A jelzőrakéta összerakásának költsége egy munka.	bemutatás és kiértékelés	fontos	A játék informális leírása	Jelzőrakéta összerakása	
70	A játékot akkor nyeri(k) meg a játékos(ok), ha sikerül összerakniuk a jelzőrakétát.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jelzőrakéta összerakása	
71	Ha egy karakter meghal, akkor a játéknak vége.	bemutatás és kiértékelés	alapvető	A játék informális leírása	Jégmező megtekintése	
72	A játékot maximum 8 karakter játszhatja.	bemutatás	opcionális	Csapat	Karakterek beállítása	
73	A karakterek szereplőcsoportja tetszőlegesen kiválasztható a játék megkezdése előtt.	bemutatás	opcionális	Csapat	Karakterek beállítása	
74	Az instabil jégtáblák teherbíró képessége 1 és 5 között lehet.	bemutatás	opcionális	Csapat	Jégmező megtekintése	Úgy kell érteni, hogy lehet 1 vagy 5 is.
75	A karakterek négy irányba mozoghatnak: előre, hátra, jobbra, balra.	bemutatás	opcionális	Csapat	Jégmező megtekintése	
76	Tetszőleges számú iglu létezhet egyszerre.	bemutatás	opcionális	Csapat	Iglu építése	

### 2.3.2 Erőforrásokkal kapcsolatos követelmények

Az osztó	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
1	A szoftver futtatásához legalább 4 GB RAM-mal rendelkező PC szükséges.	Kiértékelés	fontos	Csapat	
2	A szoftver használatához billentyűzet szükséges.	Bemutatás	alapvető	Csapat	

3	A szoftver helyes használatához szükséges valamilyen kijelző, például egy monitor.	Bemutatás	alapvető	Csapat	
4	A szoftver futtatásához a JRE legalább 1.8-as verziója szükséges	Bemutatás	alapvető	Csapat	

### 2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
1	Az elkészült szoftvernek futtathatónak kell lennie a .jar kiterjesztésű fájl elindításával.	bemutatás	alapvető	Csapat	
2	A szoftver elsősorban PC-re Windows 10 operációs rendszerre készül; azon kell telepíthetőnek lennie.	bemutatás	alapvető	Csapat	
3	A szoftvernek működnie kell, ha annak installációját a szoftver megfelelőnek találta (úgy találta, hogy minden szükséges fájl a futáshoz elérhető és megfelelő helyen van).	bemutatás	alapvető	Csapat	

### 2.3.4 Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
1	A felhasználó alapszinten tudja kezelnı a billentyüzetet.	bemutatás	alapvető	Csapat	

## 2.4 Lényeges use-case-ek

### 2.4.1 Use-case leírások

Use-case neve	Hó eltakarítása
Rövid leírás	A játékos eltakarítja a havat a jégtáblán, amin a karakter éppen rajta van.
Aktorok	Játékos
Forgatókönyv	1. A játékos eltakarít egy egységnyi havat a jégtáblán, amin a körön lévő karakter áll.
Alternatív forgatókönyv	1.A.1. Ha a karakternek megvan a lapát tárgy, akkor egy egység helyett két egységnyi havat tud eltakarítani.

Use-case neve	Jégtáblára lépés
Rövid leírás	A játékos átlép egy szomszédos jégtáblára.
Aktorok	Játékos
Forgatókönyv	1. A játékos átlépteti a körön lévő karaktert egy kiválasztott szomszédos jégtáblára. Felfelé, lefelé, balra, jobbra mozoghat.
Alternatív forgatókönyv	1.A.1. A játékos nem tud lépni a kiválasztott irányba, ha ott tenger van.
Alternatív forgatókönyv	1.B.1. A játékos nem tud lépni a kiválasztott irányba, ha ott elsüllyedt jégtábla van.
Alternatív forgatókönyv	1.C.1. Ha a karakter olyan jégtáblára lépett, ahol a karakter odaléptével a terheltségi szint nagyobb, mint kellene, akkor a játékos elsüllyed.

Use-case neve	Jégmező megtekintése
Rövid leírás	A játékos megtekinti a jégmezőt.
Aktorok	Játékos
Forgatókönyv	1. A rendszer kirajzolja a jégmező aktuális állapotát. 2. A játékos megtekinti a jégmező aktuális állapotát.

Use-case neve	Tárgy kiásása
Rövid leírás	A játékos kiássza a tárgyat azon a jégtáblán, amin éppen rajta van.
Aktorok	Játékos
Forgatókönyv	1. A játékos kiássza a tárgyat, ami azon a jégtáblán van, amin a körön lévő karakter éppen rajta áll. 2. A karakter felveszi a tárgyat és eltárolja, ilyenkor a tárgyat automatikusan használja.

<b>Alternatív forgatókönyv</b>	1.A.1. A játékos nem tudja kiásni az ott lévő tárgyat, ha a jégtáblán még van hó.
<b>Alternatív forgatókönyv</b>	1.B.1. A játékos nem tudja kiásni a tárgyat, ha a karakternél már van nála egy ugyanolyan fajta tárgy.
<b>Alternatív forgatókönyv</b>	2.A.1. Ha a játékos élelmet ás ki, akkor azt nem tárolja el a karakter, hanem egyből megeszi, ezzel növeli saját testhőjét eggel.
<b>Alternatív forgatókönyv</b>	2.A.1.A.1. Ha a karakter maximum testhővel rendelkezik, akkor nem növekszik eggel a testhője.

<b>Use-case neve</b>	<b>Játékkör átadása</b>
<b>Rövid leírás</b>	A játékos átadja a kört egy másik karakternek.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	1. A játékos átadja a kört a sorban következő karakternek.

<b>Use-case neve</b>	<b>Jelzőrakéta összerakása</b>
<b>Rövid leírás</b>	A játékos összerakja a három alkatrészből a jelzőrakétát, ezzel megnyerve a játékot.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	1. A játékos a körön lévő karakterrel összerakja a jelzőrakétát, ha az egy jégmezőn álló karakterek között megvan az összes jelzőrakéta alkatrész. 2. A játékos kilövi a jelzőfényt, ezzel megnyerve a játékot.

<b>Use-case neve</b>	<b>Hóvihar vezérlése</b>
<b>Rövid leírás</b>	A jégmezőn időnként feltámad a hóvihar.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	1. A vezérlő hóvihart csinál a jégmezőn, ami néhány érintett jégtáblát újabb adag hóval borít be.
<b>Alternatív forgatókönyv</b>	1.A.1. Ha egy karakter az érintett jégtáblán áll, akkor egy testhőt elveszít.
<b>Alternatív forgatókönyv</b>	1.A.1.A.1. Ha az érintett jégtáblán iglu található, akkor nem veszít el testhőt a karakter.
<b>Alternatív forgatókönyv</b>	1.A.1.A.1.A.1. Ha az igluban található már egy másik karakter, akkor szintén veszít egy testhőt.

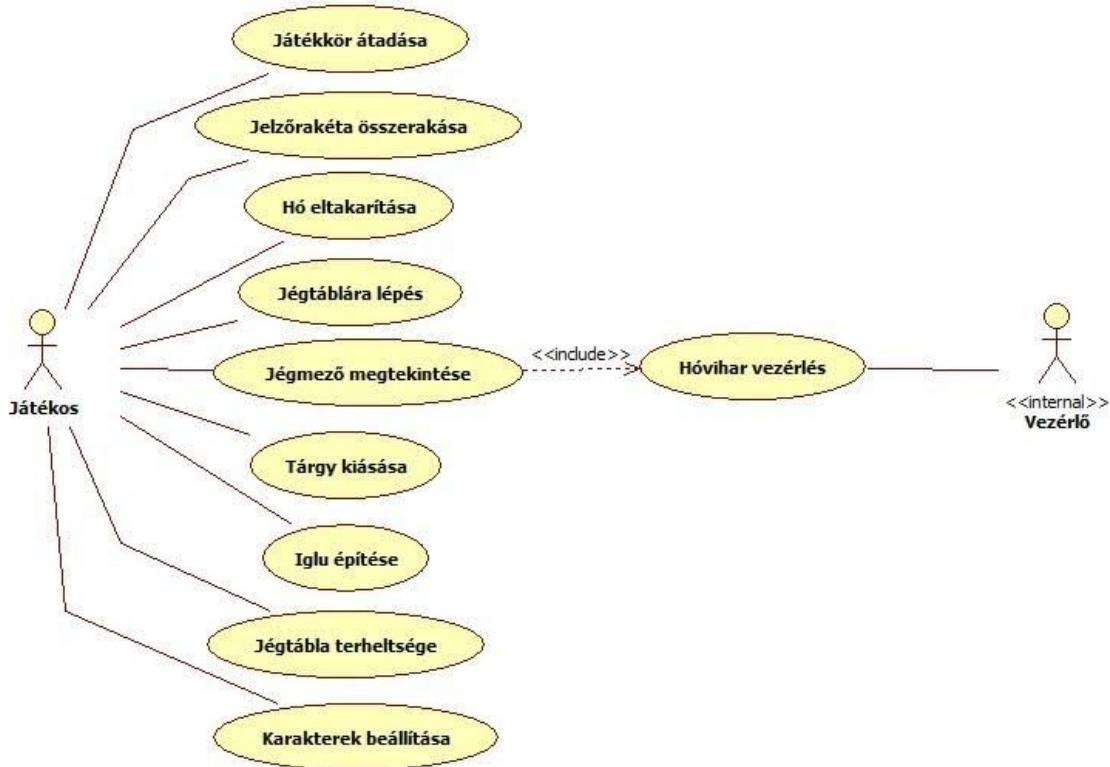
<b>Use-case neve</b>	<b>Karakterek beállítása</b>
<b>Rövid leírás</b>	A játékos beállítja a karaktereket.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	1. A játékos beállítja a karakterek számát. 2. A játékos beállítja a karakter típusokat.

<b>Use-case neve</b>	<b>Iglu építése</b>
<b>Rövid leírás</b>	A játékos iglut épít egy jégtáblára.

<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	1. A játékos az aktuális jégtáblán, amin éppen rajta van az eszkimó karaktere, épít egy új iglut.
<b>Alternatív forgatókönyv</b>	1.A.1. Ha van már a jégtáblán egy iglu, akkor nem tud új iglut építeni.
<b>Alternatív forgatókönyv</b>	1.B.1. Ha a jégtáblán hó található, akkor nem tud új iglut építeni.
<b>Alternatív forgatókönyv</b>	1.C.1. Ha a jégtáblán kiásatlan tárgy található, akkor nem tud új iglut építeni.

<b>Use-case neve</b>	<b>Jégtábla terheltsége</b>
<b>Rövid leírás</b>	A játékos megnézi egy jégtábla mennyi karaktert bír el.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	1. A játékos megnézi, hogy a szomszédos jégtáblára, mennyi karakter léphet rá ahoz, hogy ne forduljon át a jégtábla. Megnéheti a játékos a sarkkutató aktuális pozíóján lévő jégtáblát is.

#### 2.4.2 Use-case diagram



## 2.5 Szótár

**Búvárruha:** egy tárgy, átvészhető vele a tengerbe esés

**Egységterület:** egy egységes terület, ami vagy jégtábla vagy tenger, csak egy ilyen területen tartózkodhatnak a karakterek, akár hóval fedett is lehet

**Élelem:** egy a jégtáblán elhelyezkedő, egy karakter által felvehető segédeszköz, amivel a játékosok növelhetik a szereplőjük testhőjét értékét

**Eszkimó:** egy szereplőcsoport, mely 5 egység testhővel rendelkezik a játék elején, az eszkit tud igtut építeni, amiben átvészhetők a hóviharok

**Hó:** egy a játékot nehezítő tényező, ami egy egységterületen helyezkedik el és ilyenkor elfed minden ami alatta van (lehet alatta jégtábla, azon tárgyak vagy akár tenger is)

**Hóval fedett lyuk:** a jégtáblák között helyezkedik el, egy egységterület gyakorlatilag tenger, amit hó fed, ha valaki rálép, akkor beleesik a tengerbe

**Hóvihar:** egy véletlenszerűen bekövetkező esemény, mely jégtáblákat érint és a jégtáblákra, vagy az azokon tartózkodó karakterekre hatást gyakorol

**Iglu:** egy eszkit által létrehozott épület, melyben átvészhető testhő vesztés nélkül a hóvihar

**Instabil jégtábla:** olyan jégtábla, ami adott létszám felett átfordul és ilyenkor a rajtuk állók a vízbe esnek

**Játék kör:** egy egység, mely alatt a játékosok egymás után végrehajtják a műveleteiket a karaktereiken, egy játék körnek akkor van vége, ha minden játékos befejezte a saját körét

**Játékos:** a való életben létező személy, az aki a játékban irányítja a játék elején választott karakterét vagy karaktereit (egy játékoshoz több karakter is tartozhat)

**Játékos köre:** egy egység mely alatt egy játékos egy karakterrel cselekvéseket hajt végre, egy körnek akkor van vége, ha a játékosnak elfogyott a "munkája" vagy úgy dönt, hogy továbbadja a cselekvési lehetőséget

**Jégmező:** a fő játékteret alkotja, amin a szereplők közlekednek, tárgyakat vesznek fel és egyéb cselekvéseket hajtanak végre, a jégmező jégtáblákból áll, tehát a jégmező a jégtáblák együttese

**Jégtábla:** egy egység a jégmezőn, amin különböző fajta tárgyak lehetnek és a karakterek ezeken közlekednek, a játék folyamán minden karakter egy ilyenben tartózkodik

**Jelzőfény:** a jelzőrakéta egy alkatrésze, ami csak stabil jégtáblán vagy egy játékosnál lehet

**Jelzőrakéta:** egy összetett tárgy, mely több alkatrészből áll (pisztoly, jelzőfény, patron), összeszerelése a játék megnyerését jelenti

**Képesség:** minden a játékban lévő szereplő rendelkezik ilyen különleges tevékenységgel, melyek szereplőcsoportonként (sarkutató, eszkimó) egyediek és csak az adott szereplő csoportba tartozó karakter hajthat végre ilyen cselekvést, többek között ezek különböztetik meg a szereplőcsoportokat egymástól

**Kötél:** egy tárgy, amivel az egyik szereplő ki tud menteni egy másikat a tengerből

**Lapát:** egy tárgy, amivel a szereplők két egység havat takaríthatnak el a jégtábláról

**Mozgás:** a játékos megváltoztathatja a karakter pozícióját, egy lépés egy munkát jelent, a karakter átlósan nem mozoghat

**Munka:** egy egység, ami azt mondja meg, hogy egy szereplő milyen mennyiségű tevékenységet végezhet egy körben

**Patron:** a jelzőrakéta egy alkatrésze, ami csak stabil jégtáblán vagy egy játékosnál lehet

**Pisztoly:** a jelzőrakéta egy alkatrésze, ami csak stabil jégtáblán vagy egy játékosnál lehet

**Sarkutató:** egy szereplőcsoport, mely 4 egység testhővel rendelkezik a játék elején, képessége, hogy a sarkutató meg tudja nézni, hogy az a jégtábla, amire lépne, hány embert bír el

**Stabil jégtábla:** olyan jégtábla, amelyeken akárhány szereplő állhat nem fordul át soha

**Szereplő/karakter:** a játékban a játékos által irányított karakter, minden játékoshoz tartozik legalább egy szereplő

**Szereplőcsoport:** az azonos tulajdonságokkal és képességekkel rendelkező szereplők csoportja

**Szomszédos jégtábla:** egy jégtábla felett, mellett, vagy tőle jobbra vagy balra elhelyezkedő másik jégtábla, csak szomszédos jégtáblára léphet tovább egy karakter

**Tárgy:** olyan eszköz, amivel a karakterek különböző képességekre tehetnek szert amennyiben birtokolják, befagyott tárgynak akkor hívjuk, ha a tárgy nem egy karakternél van, hanem egy jégtáblán, csak akkor látható, ha a jégtáblát nem borítja hó

**Teherbíró képesség:** egy szám, ami arra utal, hogy egy instabil jégtábla hány embert bír el, azaz hány ember állhat rajta mielőtt átfordulna

**Tenger:** a jégmezőt veszi korül, és a jégmező alatt is megtalálható, a karakterek beleeshetnek, ekkor meghalnak és ekkor a játéknak vége.

**Testhő:** a karakterek egyik számszerű jellemzője, aminek értéke ha nullára csökken az adott karakter meghal, a hóvihar csökkenti, viszont élelemmel növelhető az értéke

**Tevékenység:** egy cselekvésre utal, ami lehet például helyváltoztatás, hó eltakarítása vagy tárgy felvétele, a tevékenységeknek munkaköltsége van, ami az egyes esetekben eltérő lehet

**Túlélés:** a játék célja a túlélés, azaz nem szabad meghalnia egy karakternek sem

## 2.6 Projekt terv

A szoftver és a dokumentáció létrehozása során fontos szempont számunkra a csapatmunka és egyetértés, ezért a nagyobb (heti) feladatokban mindenki részt vesz, viszont ezeket kisebb részekre bontva külön-külön oldjuk meg. Emiatt nincsenek a feladatok személyenként felosztva, mindenki abban vesz részt, amiben tud.

Határidő	Feladat
Február 24.	Követelmény, projekt, funkcionálitás - beadás
Március 2.	Analízis modell kidolgozása 1. - beadás
Március 9.	Analízis modell kidolgozása 2. - beadás
Március 16.	Szkeleton tervezése - beadás
Március 23.	Szkeleton - beadás és a forráskód herculesre való feltöltése
Március 30.	Prototípus koncepciója - beadás
Április 6.	Részletes tervezés - beadás
Április 20.	Prototípus készítése, tesztelése
Április 27.	Prototípus - beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése
Május 4.	Grafikus felület specifikációja - beadás
Május 11.	Grafikus változat készítése
Május 18.	Grafikus változat és Összefoglalás - beadás és a forráskód herculesre való feltöltése

A választott fejlesztési folyamat miatt fontos a folyamatos kommunikáció, amihez a Messenger-t és Discord-ot használjuk.

A dokumentáció írására Google Docs-ot használunk, a forráskód megosztása pedig Git segítségével történik. Ezekkel az eszközökkel egyszerre akár többen is tudunk ugyanazon a feladaton dolgozni.

Modellezéshez a WhiteStarUML-t használjuk, a kódot pedig IntelliJ-ben, illetve Eclipse-ben írjuk.

## 2.7 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.02.18. 17:30-20:00	2.5 óra	Kaposvári Labancz Iuhos Sulyok Traxler	Értekezlet: a feladat értelmezése, megbeszélése, munkafeladatok kiosztása, mindenki legkésőbb 2020.02.23. délig megoldja az általa választott feladatot
2020.02.19. 18:00-21:30	3.5 óra	Labancz	Tevékenység: Funkciók alfejezet megírása
2020.02.20. 20:00-23:00	3 óra	Sulyok	Tevékenység: Szótár
2020.02.21. 16:00-17:30	1.5 óra	Labancz	Tevékenység: Általános áttekintés, Cél, Felhasználók és egyéb kisebb alfejezetek szerkesztése, javítások
2020.02.21. 20:00-21:00	1 óra	Kaposvári Labancz Iuhos Sulyok Traxler	Értekezlet: discord konzultáció a hátralévő feladatok elosztásáról: Kaposvári elvállalja a use-casek elkészítését, Traxler a funkcionális követelményeket fogja megoldani
2020.02.22. 9:00-12:30	3.5 óra	Iuhos	Tevékenység: Projekt terv megírása, kisebb bevezető fejezetek megírása és az eddig elkészült dolgok javítása, javaslatok a további munkához

2020.02.22. 13:00-14:00	1 óra	Kaposvári Iuhos Traxler	Értekezlet: Funkcionális követelmények és use-casek megtervezése, egyeztetés, eddig felmerült apróbb hibák javítása
2020.02.22. 14:00-18:30	4.5 óra	Traxler	Tevékenység: Funkcionális követelmények megírása
2020.02.22 16:00-18:30	2.5 óra	Kaposvári	Tevékenység: use-case diagram
2020.02.22 20:00-22:00	2 óra	Kaposvári	Tevékenység: use-case leírások
2020.02.22. 20:00-22:00	2 óra	Sulyok	Tevékenység: Szótár bővítése a megjelent új fogalmakkal, megjegyzések, módosítási javaslatok felvetése
2020.02.23. 15:00-16:00	1 óra	Iuhos	Tevékenység: dokumentum formázása
2020.02.23. 20:00-22:00	2 óra	Kaposvári Labancz Iuhos Sulyok Traxler	Értekezlet: döntés a dokumentum formai megjelenéséről és a végleges változat elfogadása

### **3. Feladat: Analízis modell kidolgozása 1. - beadás**

**14 – the\_minesweepers**

Konzulens:  
**Fetter László**

#### **Csapattagok**

Iuhos Zoltán	J8LOUL	zoltan.iuhos@gmail.com
Kaposvári Márk Martin	RAYVF8	xmarko45@gmail.com
Sulyok Botond	BM2BBM	botond.sulyok99@gmail.com
Labancz Tamás Dominik	A9HZBF	labancz.tamas@gmail.com
Traxler Bálint	IXNOYB	traxlerbalint@gmail.com

2020.03.02.

## 3. Analízis modell kidolgozása

### 3.1 Objektum katalógus

#### 3.1.1 Cartridge (patron)

A jelzőpisztoly egyik alkatrésze, szükséges ahhoz, hogy meg lehessen nyerni a játékot, a jégtáblákról lehet felvenni.

#### 3.1.2 DivingSuit (búvárruha)

Ez által átvészélhető a tengerbe esés, de valamekkora valószínűsséggel tönkre tud menni. Alapból a jégtáblákon van és onnan vehetik fel a karakterek.

#### 3.1.3 Eskimo (eszkimó)

Az eszkimó is egy választható karakter a játékban, azonban a képessége eltér a kutatójától, ő iglut tud építeni a mezőre, amin áll és ebben átvészélhető a hóvihar.

#### 3.1.4 Explorer (kutató)

A játékban a sarkkutatót reprezentálja, egy vagy több játékos is irányíthatja, legfőbb célja a túlélés. Tud cselekedni, azaz különböző munkákat végez: léphet jégtábláról jégtáblára, használhatja a képességét (magnézheti egy szomszédos jégtábla vagy amin áll annak a teherbíró képességét), tárgyakat vehet fel és használhat, de bele tud esni a tengerbe is.

#### 3.1.5 Food (étel)

Ez egy olyan objektum, ami a karakterek testhőjét növeli. Amikor felszedik a jégtábláról, akkor egyből fel is használják.

#### 3.1.6 IceUnit (jégtábla)

A játékban ez az alapegység, amin a karakterek állhatnak és haladhatnak ezeken. Emellett tárgyak is lehetnek rajtuk. Kezeli a rajta lévő vagy rálépő karakterek utasításait. A játékosokat nehezíti, hogy hó is fedheti, ami elfedi a tárgyakat.

#### 3.1.7 Igloo (iglu)

Az eszkimók tudnak építeni iglukat, ami gyakorlatilag egyfajta épületként értelmezhető. Védelmet nyújt a hóviharok ellen.

#### 3.1.8 Instable (instabil jégtábla)

Az sima jégtáblák között helyezkedhet el egy ilyen objektum. Lényege, hogy csak bizonyos számú karaktert bír el és ha túllépte a teherbíró képességét, akkor átfordul (beszakad) és tenger lesz belőle, amibe a rajta álló játékosok belesesnek. Amikor beszakad és tenger lesz belőle, akkor már nem léphetnek rá játékosok.

#### 3.1.9 Light (jelzőfény)

A jelzőpisztoly egyik alkatrésze, szükséges ahhoz, hogy meg lehessen nyerni a játékot, a jégtáblákról lehet felvenni.

### 3.1.10 Pistol (pisztoly)

A jelzőpisztoly egyik alkatrésze, szükséges ahhoz, hogy meg lehessen nyerni a játékot, a jégtábláról lehet felvenni.

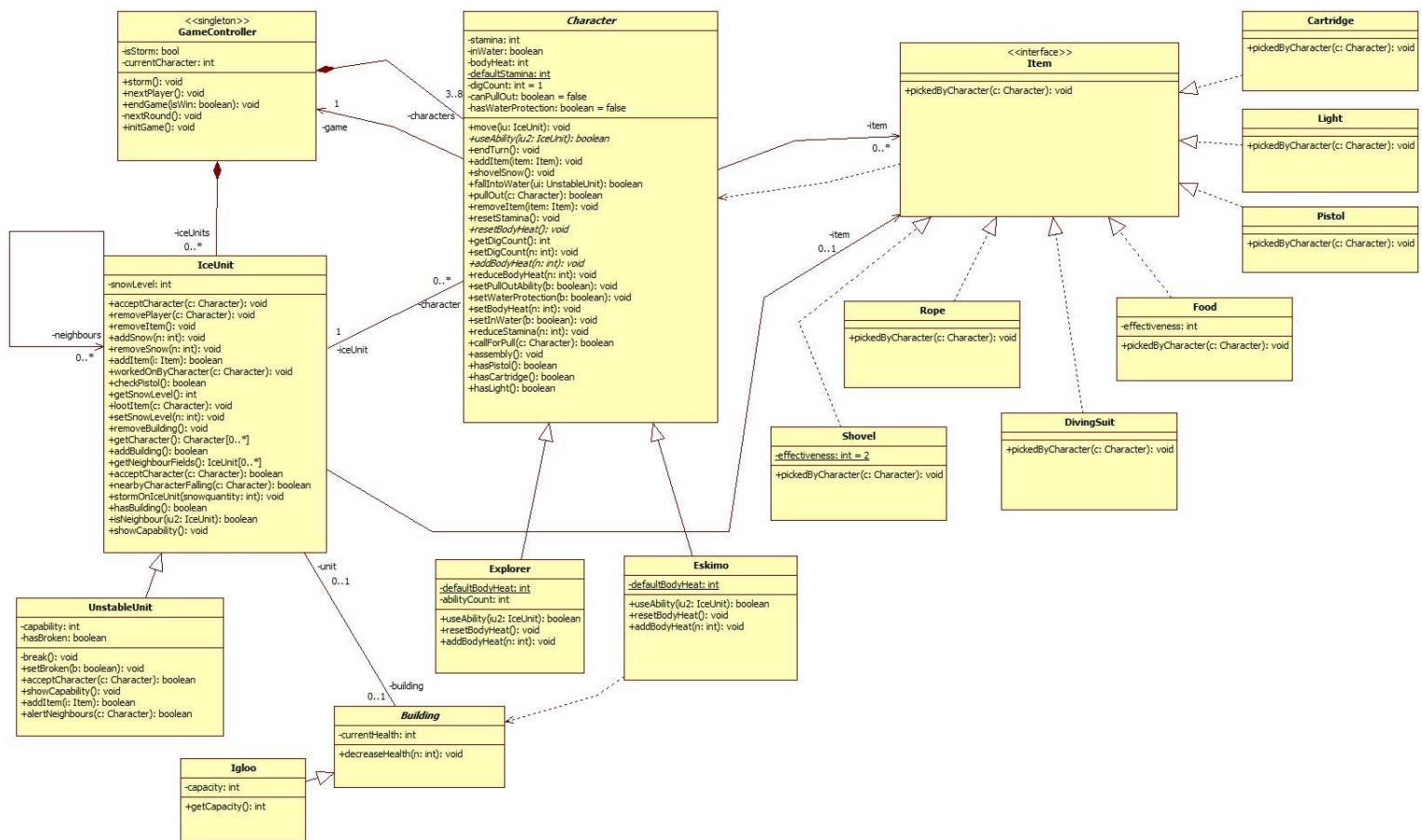
### 3.1.11 Rope (kötél)

Ez a jégtáblákon megtalálható tárgy arra képes, hogy aki felveszi, az ezt használva kimenekíthet egy vízbe esett karaktert, aki a vízbe esés pillanatában egy vele szomszédos jégtáblán helyezkedett el.

### 3.1.12 Shovel (lapát)

A játékban ez egy felvehető eszköz, ami segíti a játékost azzal, hogy a karaktere több havat tud ellapárolni. Tehát fel lehet venni és a használata előnyhöz juttatja a játékost.

## 3.2 Statikus struktúra diagramok



Megjegyzés: A nevesített asszociációvégekhez további implicit getter és setter függvények tartoznak, amelyeket az implementációtól függően lehet megvalósítani.

Az osztálydiagramot az átláthatóság kedvéért egy külön oldalon is csatoljuk.

### 3.3 Osztályok leírása

#### 3.3.1 Building

- **Felelősség**

A jégmezőre építhető épületeket kezelő absztrakt osztály.

- **Attribútumok**

- – **IceUnit unit:** Tárolja, hogy melyik táblán van.
- – **int currentHealth:** Az épület ennyi vihart él még túl.

- **Metódusok**

- **void decreaseHealth(int n):** Csökkenti a maradék életét az épületnek a paraméterként kapott mennyiséggel.

#### 3.3.2 Cartridge

- **Felelősség**

A jelzőpisztoly összerakásához szükséges tárgy. Magyarul patron.

- **Interfészek**

Item

- **Metódusok**

- **void pickedByCharacter(Character c):** A paraméterül kapott karakter tárolójába kerül a tárgy.

#### 3.3.3 Character

- **Felelősség**

A játékban lévő karaktereket kezeli.

- **Attribútumok**

- – **GameController game:** Eltárolja a játékvezérőt.
- – **IceUnit iceUnit:** Tárolja, hogy a karakter melyik jégtáblán áll.
- – **Item[0..\*] item:** A karakternél lévő tárgyakat tárolja el.
- – **int stamina:** A karakter egy körben elvégzett munkájának nyilvántartásához kell.
- – **boolean inWater:** Igaz értékű, ha a karakter vízbe esett.
- – **int bodyHeat:** A karakter testhője, ha elfogy, a karakter meghal.
- – **int defaultStamina:** Statikus adattag, azt tárolja, hogy mennyi munkát végezhet egy karakter egy körben.
- – **int digCount:** Azt mutatja, hogy mennyi havat tud eltakarítani a karakter egy munkával.
- – **boolean canPullOut:** Ha a karakternél van kötél, akkor igaz az értéke.
- – **boolean hasWaterProtection:** Ha van a karakternél búvárruha, akkor igaz az értéke.

- **Metódusok**

- **void move(IceUnit iu):** A karakter a paraméterként kapott jégtáblára lép, ha az nem tenger. Ennek következményeit a jégtábla kezeli le.
- **boolean useAbility(IceUnit iu):** A karakter a paraméterként kapott táblán cselekvést végez, ha tud. Ez lehet hó eltakarítása, tárgy kiásása, vagy képesség alkalmazása. Az előző sorrend szigorúan követendő, nem lehet tárgyat kiásni, amíg van hó, és nem lehet képességet alkalmazni, amíg van hó vagy tárgy a táblán. Absztrakt függvény.
- **void endTurn():** A karakter befejezi a körét.
- **void addItem(Item item):** A karakter új tárgyat vesz fel.
- **void shovelSnow():** A karakter havat lapátol, a useAbility() hívja meg. Ha van nála lapát, akkor dupla mennyiségű havat tud lapátolni.
- **boolean fallIntoWater(UnstableUnit ui):** A karakter beleesik a vízbe. Ha nincs bűvárruhája, és nem húzza ki senki, akkor meghal és vége a játéknak.
- **boolean pullOut(Character c):** Kihúzza a szomszédos tábláról vízbe esett karaktereket, ha tudja.
- **void removeItem(Item item):** A karakternél lévő tárgyak közül eltávolítja a paraméterként kapott tárgyat.
- **void addBodyHeat(int n):** Növeli a karakter testhőjét a kapott mennyiséggel. Absztrakt függvény.
- **void resetStamina():** Visszaállítja a karakter által elvégezhető munka mennyiségét a default értékre.
- **void resetBodyHeat():** Visszaállítja a karakter testhőjét az alap értékre. Absztrakt függvény.
- **void reduceBodyHeat(int n):** Csökkenti a karakter testhőjét a kapott értékkel.
- **int getDigCount():** Visszatér a digCount értékével.
- **void setDigCount(int n):** A digCount setttere.
- **void setPullOutAbility(boolean b):** A canPullOut setttere.
- **void setWaterProtection(boolean b):** A hasWaterProtection setttere.
- **void setBodyHeat(int n):** A bodyHeat setttere.
- **void setInWater(boolean b):** Az inWater setttere.
- **void reduceStamina(int n):** Csökkenti a karakter által adott körben elvégezhető munka mennyiségét a kapott paraméter értékével.
- **boolean callForPull(Character c):** Amennyiben a karakter vízbeesik, ezzel a függvényel kér segítséget a többiektől. Ha valaki kitudja húzni, akkor az a mező amin a kihúzó karakter áll befogadja a kihúzottat. Ha az adott karakter nem tudja kihúzni a játékos, akkor a függvény false-al tér vissza.
- **void assembly():** A jelzőpisztoly összerakásának lehetőségét vizsgálja úgy, hogy meghívja a mezőnek a checkPistol metódusát, ami tudja, hogy milyen karakterek állnak rajta. Amennyiben lehetséges az összeszerelés, tehát a tárgyak rendelkezésre állnak és a során lévő játékosnak van elég "munkája", akkor a játék véget ér, győzelemmel.
- **boolean hasPistol():** Igaz értékkel tér vissza, ha van pisztoly a karakternél.
- **boolean hasCartridge():** Igaz értékkel tér vissza, ha van patron a karakternél.
- **boolean hasLight():** Igaz értékkel tér vissza, ha van jelzőfény a karakternél.

### 3.3.4 DivingSuit

- **Felelősség**

A bűvárruhát megvalósító osztály.

- **Interfészek**  
Item
- **Metódusok**
  - **void pickedByCharacter(Character c):** A paraméterül kapott karakter tárolójába kerül a tárgy.

### 3.3.5 Eskimo

- **Felelősség**  
A játékban lévő eszkimókat kezeli.
- **Ősosztályok**  
Character
- **Attribútumok**
  - **-int defaultBodyHeat:** Statikus adattag, ami mutatja, hogy alapból mennyi testhője van az eszkimóknak.
- **Metódusok**
  - **void useAbility(IceUnit iu2):** A karakter ennek hatására vagy havat lapátol, vagy tárgyat vesz fel, vagy használja a képességét az adott mezőn.
  - **void resetBodyHeat():** Inicializáláskor beállítja a testhőt.
  - **void addBodyHeat(int n):** Növeli a karakter testhőjét a paraméterként kapott értékkal, de nem mehet a kezdeti érték fölé.

### 3.3.6 Explorer

- **Felelősség**  
A játékban lévő sarkutatókat kezeli.
- **Ősosztályok**  
Character
- **Attribútumok**
  - **-int defaultBodyHeat:** Statikus adattag, ami mutatja, hogy alapból mennyi testhője van a sarkutónak.
  - **- int abilityCount:** Számolja, hogy a karakter hányszor használta a képességét.
- **Metódusok**
  - **void useAbility(IceUnit iu2):** A karakter ennek hatására vagy havat lapátol, vagy tárgyat vesz fel, vagy használja a képességét az adott mezőn.
  - **void resetBodyHeat():** Inicializáláskor beállítja a testhőt.
  - **void addBodyHeat(int n):** Növeli a karakter testhőjét a paraméterként kapott értékkal, de nem mehet a kezdeti érték fölé.

### 3.3.7 Food

- **Felelősség**

A játékban lévő élelmet megvalósító osztály.

- **Interfészek**

Item

- **Attribútumok**

- – **int effectiveness:** Azt jelzi, hogy mennyi testhőt képes visszatölteni.

- **Metódusok**

- **void pickedByCharacter(Character c):** A paraméterül kapott karakter tárolójába kerül a tárgy.

### 3.3.8 GameController

- **Felelősség**

Ez az osztály irányítja a játékot, vagyis inicializál, kezeli a játékosokat és a köröket, valamint a játék végét.

- **Attribútumok**

- – **bool isStorm:** Azt jelzi, hogy van-e vihar az adott körben.
- – **Charater[3..8] characters:** A játékban lévő karakterek tárolása.
- – **IceUnit[0..\*] iceUnits:** A játékban lévő jégtáblák tárolása.
- – **int currentCharacter:** Nyilvántartja, hogy éppen melyik karakter köre van folyamatban.

- **Metódusok**

- **void storm():** minden kör elején megnézi, az isStorm attribútum értékét. Ha true, akkor néhány IceUnit-ra meghívja az addSnow() és destroyBuilding() függvényt, valamint az ezeken az IceUnitokon lévő Character-ek reduceBodyHeat() függényét, ha azok nem voltak iglus mezőn. Ha false, akkor bizonyos valószínűsséggel true-ra állítja és jelez, hogy vihar jön.
- **void nextPlayer():** Elkezdi a következő játékos körét.
- **void endGame(boolean isWin) :** Ha igaz értékkal hívódik meg, akkor véget vet a játéknak, vagyis jelzi, hogy nyertek vagy vesztettek a játékosok és nem kezdődik új kör.
- **void nextRound():** Elkezdi a következő kört, aminek az első lépéseként meghívja a strom() függvényt.
- **void initGame():** Inicializálja a játék kezdése előtt az összes objektumot.

### 3.3.9 IceUnit

- **Felelősség**

A játékban lévő stabil jégtáblákat kezeli.

- **Attribútumok**
  - - int **snowLevel**: A jégtáblán lévő hó mennyiségét tárolja.
  - – Character[0..\*] **character**: A mezőn álló játékosokat tárolja.
  - – Item **item** : A mezőbe fagyott tárgyat tárolja.
  - –Building **building**: A mezőn lévő épületet tárolja (például az iglut).
  - –IceUnit[0..\*] **neighbours**: A jégtábla tárolja a szomszédjait.
- **Metódusok**
  - **void acceptCharacter(Character c)**: A jégtáblára lépő karaktert hozzáadja az ott lévőkhöz.
  - **void removePlayer(Character c)**: A jégtábláról ellépő játékost kiveszi az ott állók közül.
  - **void removeItem()**: Eltávolítja a táblán lévő tárgyat.
  - **void addSnow(int n)**: Megnöveli a táblán lévő hó mennyiségét.
  - **void removeSnow(int n)**: Eltávolít a tábláról n egység havat.
  - **boolean addItem(Item i)**: Inicializáláskor hozzáadja a paraméterként kapott tárgyat a táblához, ha lehet.
  - **void workedOnByCharacter(Character c)**: A paraméterként kapott játékos ennek hatására havat lapátol, ha nincs hó, akkor kiássa a tárgyat, ha tárgy sincs, akkor pedig alkalmazza a képességét.
  - **void checkPistol(GameController gamecontroller)**: Azt vizsgálja, hogy a táblán álló játékosok össze tudják-e rakni a jelzőpisztolyt, amivel megnyerhetik a játékot. Ha igen, akkor szól a GameControllernek, hogy vége a játéknak, nyert a csapat.
  - **int getSnowLevel()**: Visszatér a táblán lévő hó mennyiséggel.
  - **void lootItem(Character c)**: A játékos kiássa a tárgyat, ami hozzákerül.
  - **void setSnowLevel(int n)**: Beállítja a táblán lévő hó mennyiségét.
  - **void removeBuilding()**: A táblán lévő épület lerombolása.
  - **Character[0..\*] getCharacter()**: Visszatér a táblán lévő karakterekkel.
  - **boolean addBuilding()**: Hozzáad a mezőhöz egy épületet.
  - **IceUnit[0..\*] getNeighbourFields()**: Visszatér a jégtáblával szomszédos jégtáblákkal.
  - **boolean acceptCharacter(Character c)**: A jégtábla lekezeli, hogy egy karakter rálépett. Ha nem tud rálépní, akkor false értékkal tér vissza.
  - **boolean nearbyCharacterFalling(Character c)**: A beszakadt jégtábla szomszédjain álló játékosokat értesíti, hogy húzzanak ki karaktereket, ha tudnak.
  - **void stormOnIceUnit(int snowquantity)**: A jégtáblára a hóvihar miatt újabb adag hó kerül, ha kerülhet rá több. Az érintett táblán lévő játékosok testhője is csökken, ha nincs védelmük.
  - **boolean hasBuilding()**: Igaz értékkel tér vissza, ha van épület a táblán.
  - **boolean isNeighbour(IceUnit iu2)**: Igaz értékkel tér vissza, ha a paraméterként kapott tábla az adott tábla szomszédja.
  - **void showCapability()**: Ha a sarkkutató megvizsgálja az adott táblát, akkor megjelenik a tábla teherbírása.

### 3.3.10 Igloo

- **Felelősség**

A pályán lévő igluk kezelése.

- **Ősosztályok**  
Building
- **Attribútumok**
  - - **int capacity:** Statikus adattag, alap értéke 1, az igluba beférő karakterek maximális száma.
- **Metódusok**
  - **int getCapacity():** Visszatér a capacity értékével.

### 3.3.11 Item

- **Felelősség**  
A játékban lévő tárgyak interfésze.
- **Metódusok**
  - **void pickedByCharacter(Character c):** A paraméterül kapott karakter tárolójába kerül a tárgy.

### 3.3.12 Light

- **Felelősség**  
A jelzőfényt megvalósító osztály. A játék megnyeréséhez szükséges tárgy.
- **Interfészek**  
Item
- **Metódusok**
  - **void pickedByCharacter(Character c):** A paraméterül kapott karakter tárolójába kerül a tárgy.

### 3.3.13 Pistol

- **Felelősség**  
A pisztolyt megvalósító osztály. A játék megnyeréséhez szükséges tárgy.
- **Interfészek**  
Item
- **Metódusok**
  - **void pickedByCharacter(Character c):** A paraméterül kapott karakter tárolójába kerül a tárgy.

### 3.3.14      **Rope**

- **Felelősség**

A kötelet megvalósító osztály.

- **Interfészek**

Item

- **Metódusok**

- **void pickedByCharacter(Character c):** A paraméterül kapott karakter tárolójába kerül a tárgy.

### 3.3.15      **Shovel**

- **Felelősség**

A lapátot megvalósító osztály.

- **Interfészek**

Item

- **Attribútumok**

- – **int effectiveness:** Statikus adattag, alap értéke 2. Azt mutatja, hogy hány réteg havat lehet eltakarítani a lapát használatával.

- **Metódusok**

- **void pickedByCharacter(Character c):** A paraméterül kapott karakter tárolójába kerül a tárgy.

### 3.3.16      **UnstableUnit**

- **Felelősség**

Az instabil jégtáblákat kezeli.

- **Ősosztályok**

IceUnit

- **Attribútumok**

- – **int capability:** A tábla teherbírását mutatja, vagyis azt, hogy hány karaktert bír el, mielőtt átfordulna.
- – **boolean hasBroken:** Igaz az értéke, ha a jégtábla beszakadt.

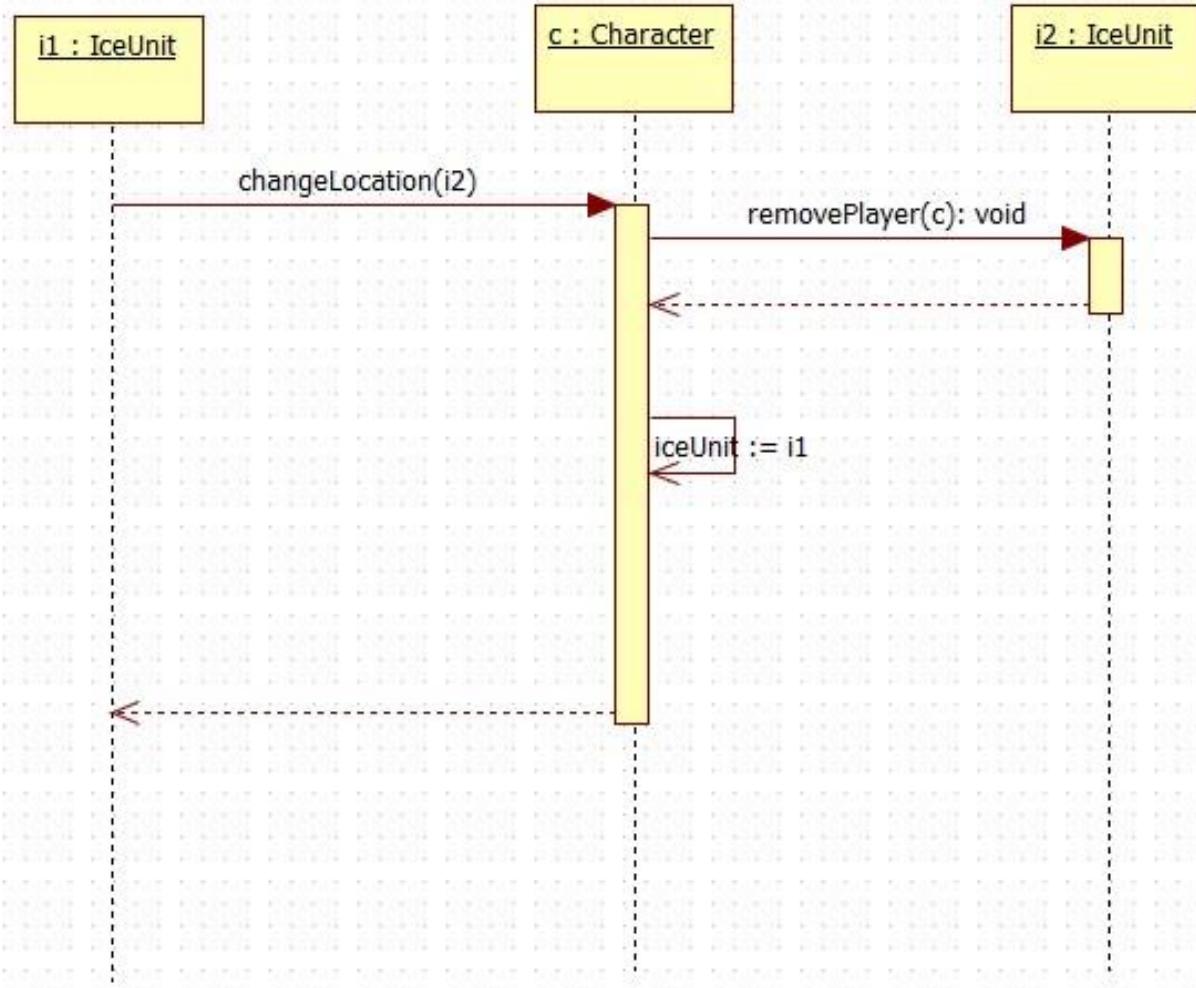
- **Metódusok**

- **void break():** A jégtábla beszakad, a rajta lévők vízbe esnek.
- **void setBroken(boolean b):** A hasBroken setttere.
- **boolean acceptCharacter(Character c):** A jégtáblára lépő karaktert hozzáadja az ott lévőkhöz.

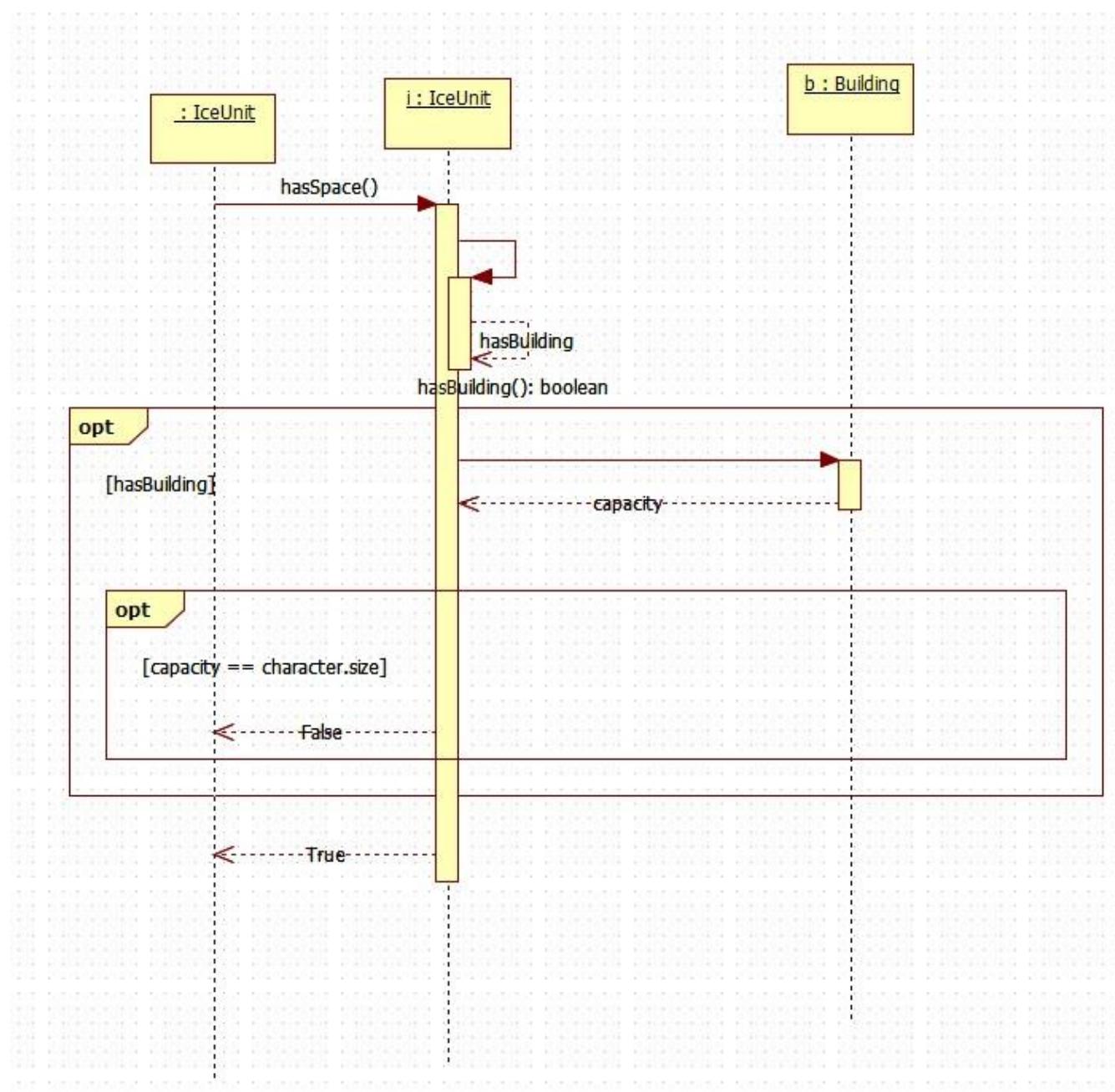
- **void showCapability()**: Ha a sarkkutató megvizsgálja az adott táblát, akkor megjelenik a tábla teherbírása.
- **boolean addItem(Item i)**: Inicializáláskor hozzáadja a paraméterként kapott tárgyat a táblához, ha lehet.
- **boolean alertNeighbours(Character c)**: Jelez a szomszédos tábláknak, hogy a rajta álló karakterek vízbe estek.

### 3.4 Szekvencia diagramok

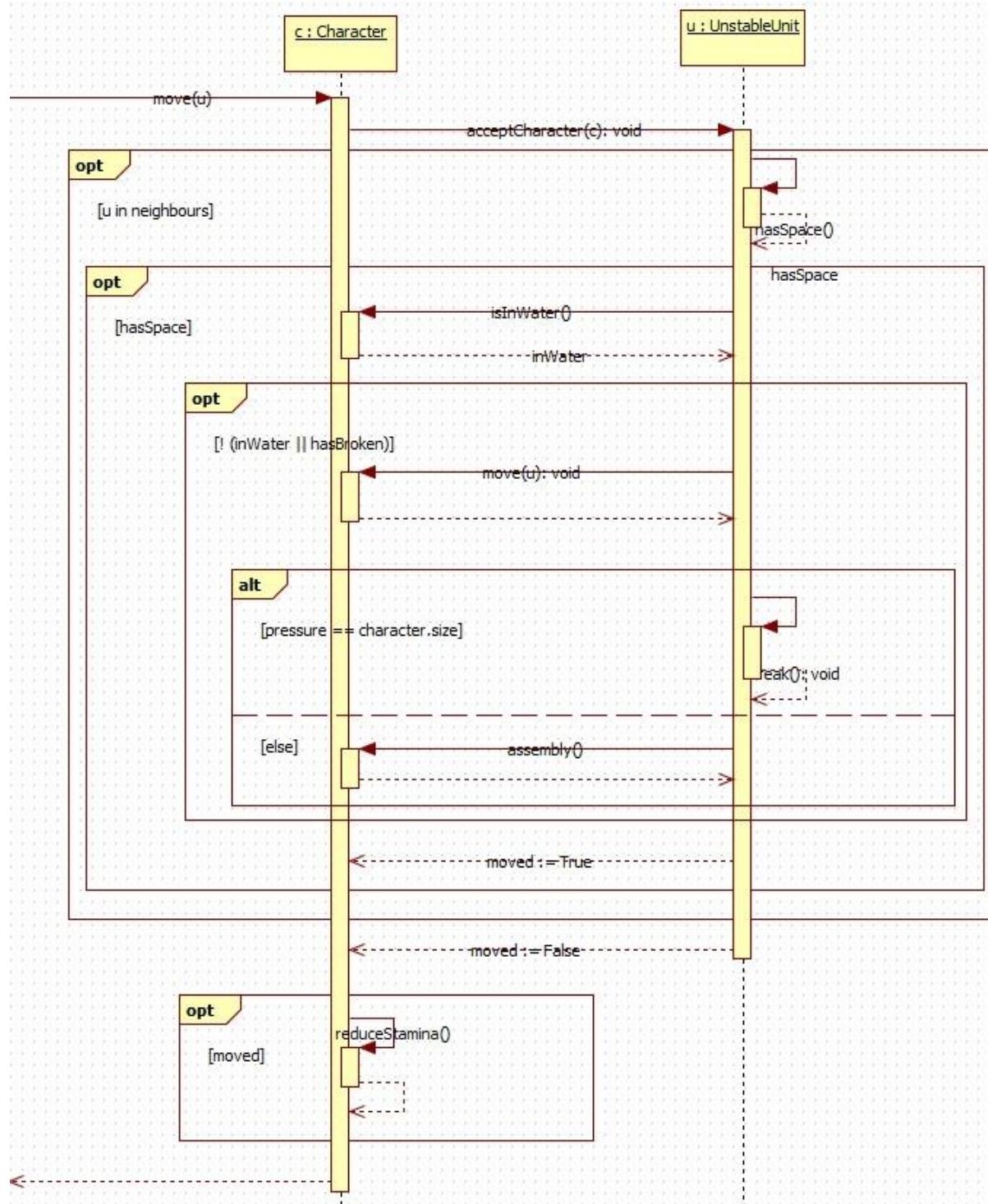
#### 3.4.1 changeLocation



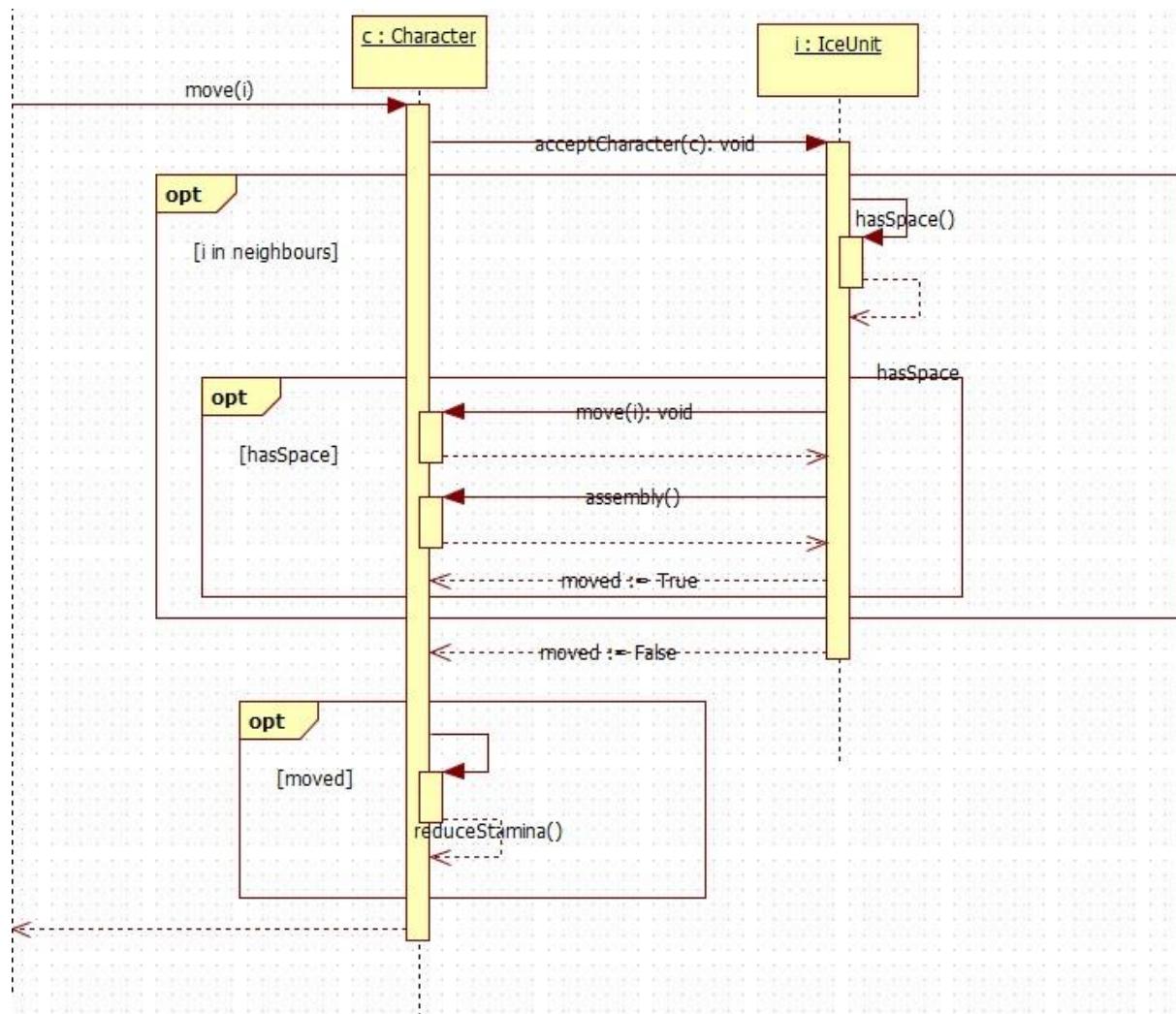
### 3.4.2 checkSpace



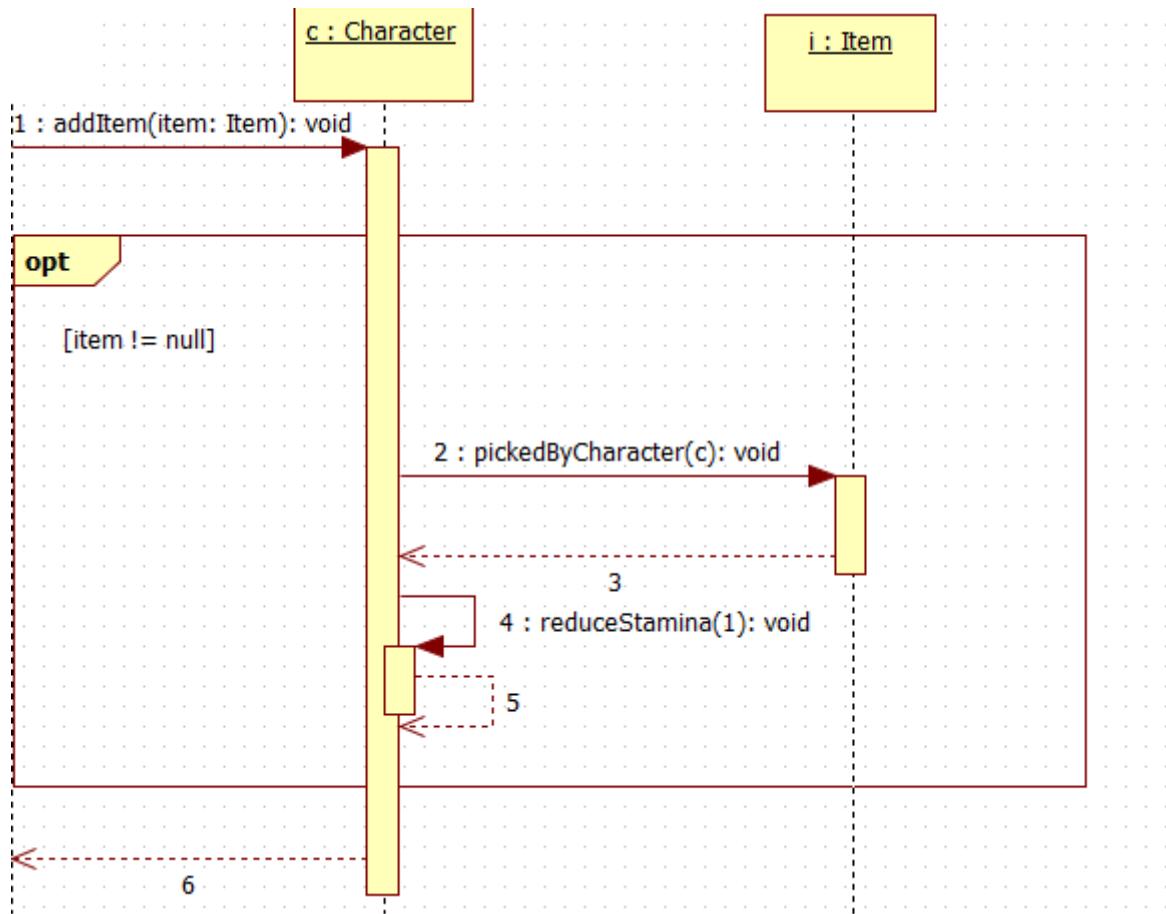
### 3.4.3 move\_IceUnit



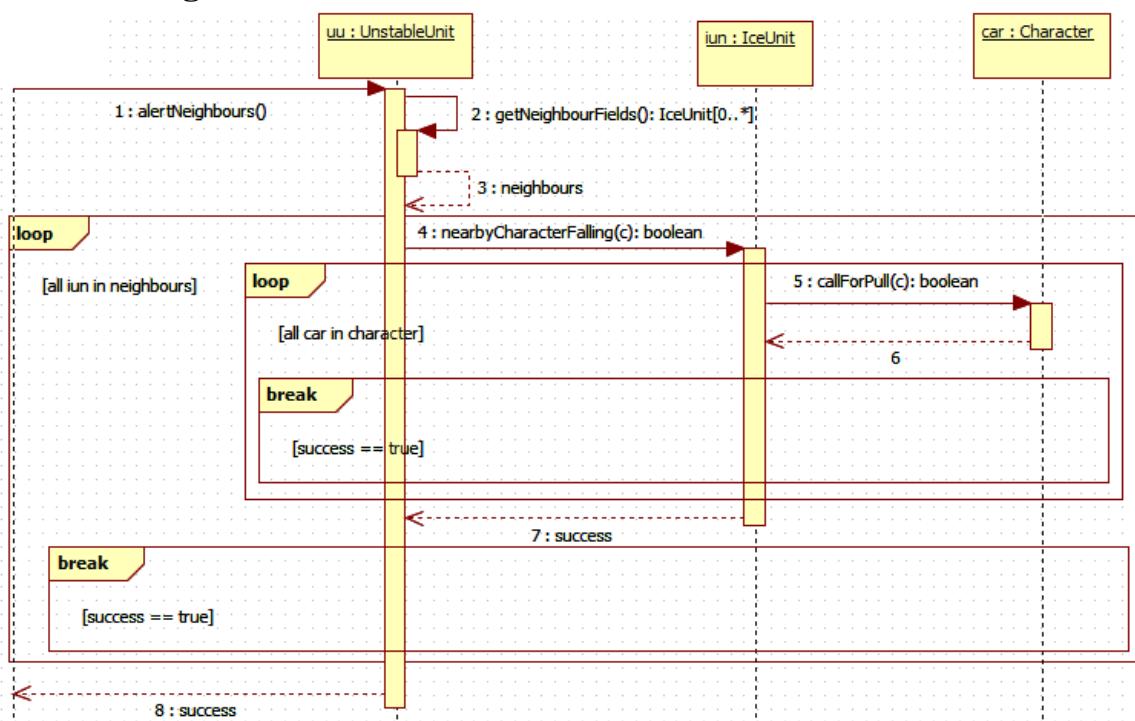
### 3.4.4 move\_Unstable



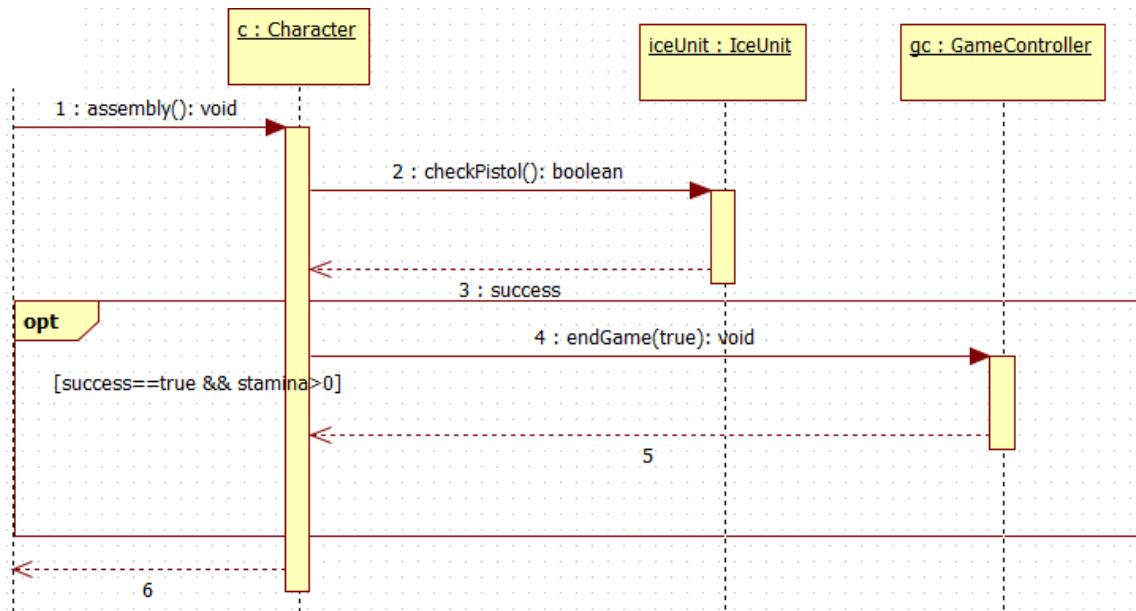
### 3.4.5 add\_Item



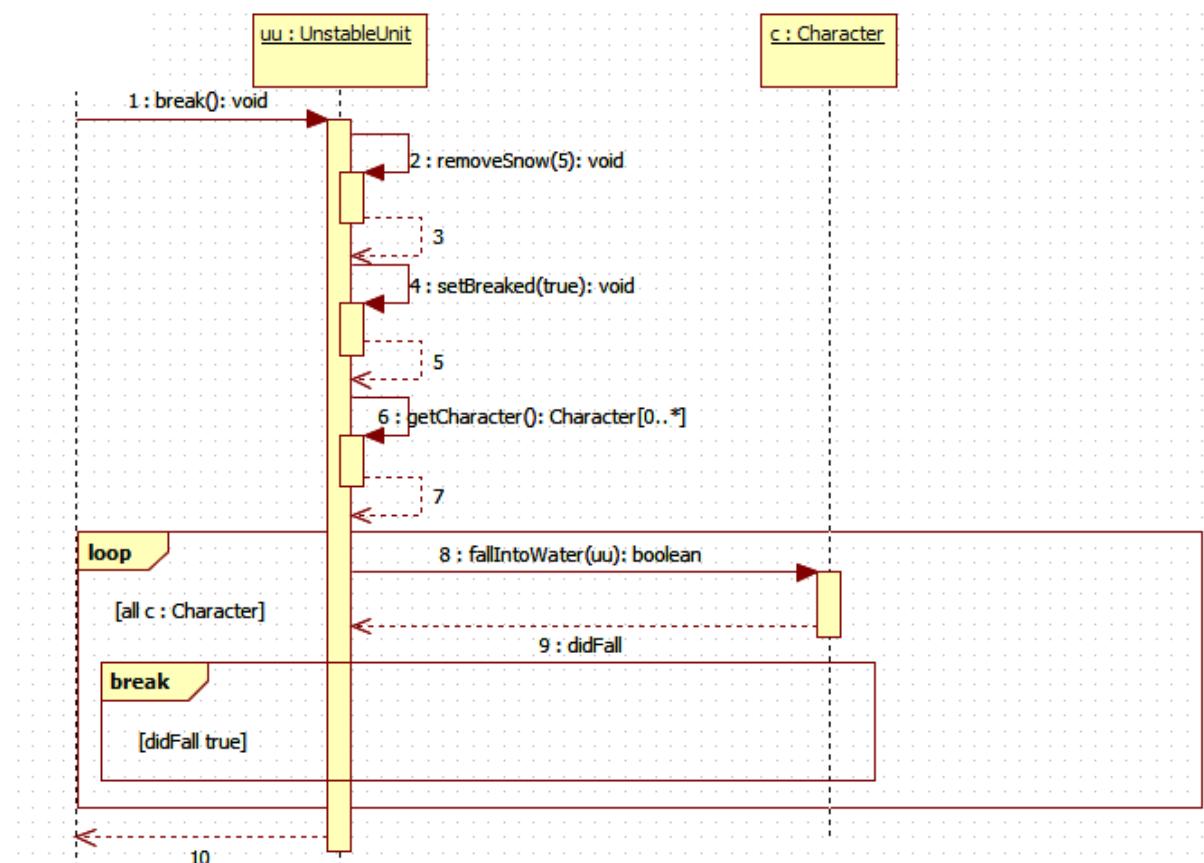
### 3.4.6 alert\_neighbours



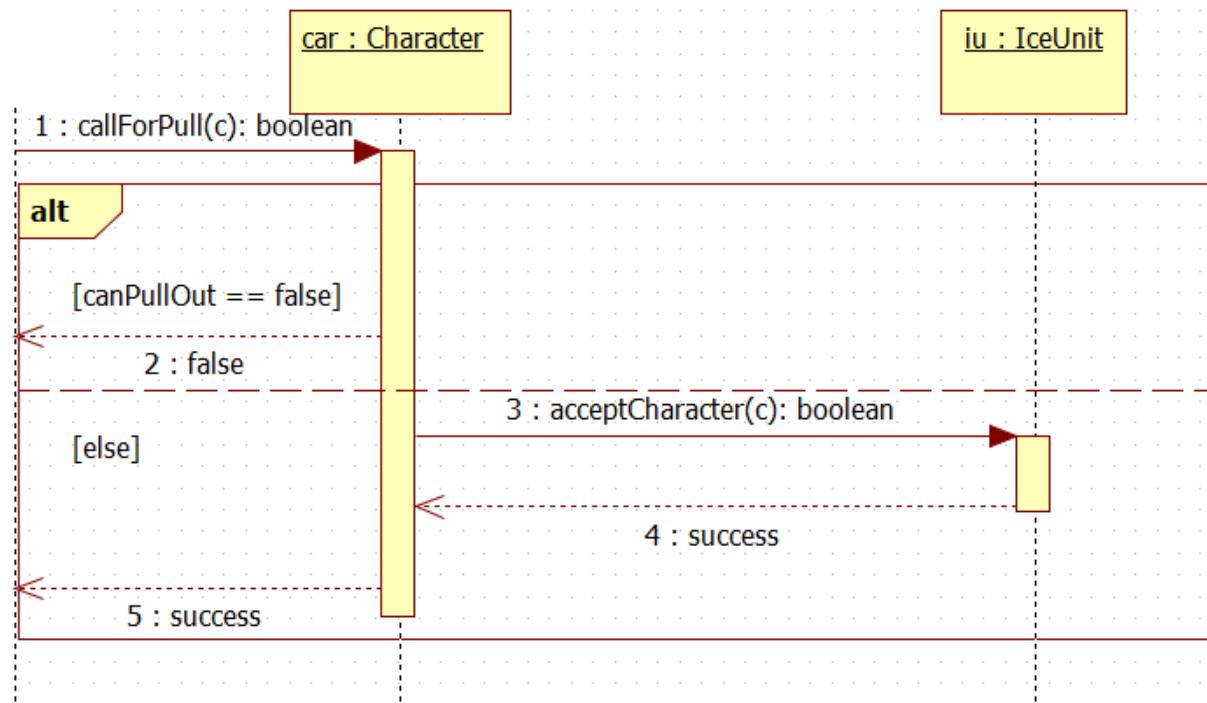
### 3.4.7 assembly



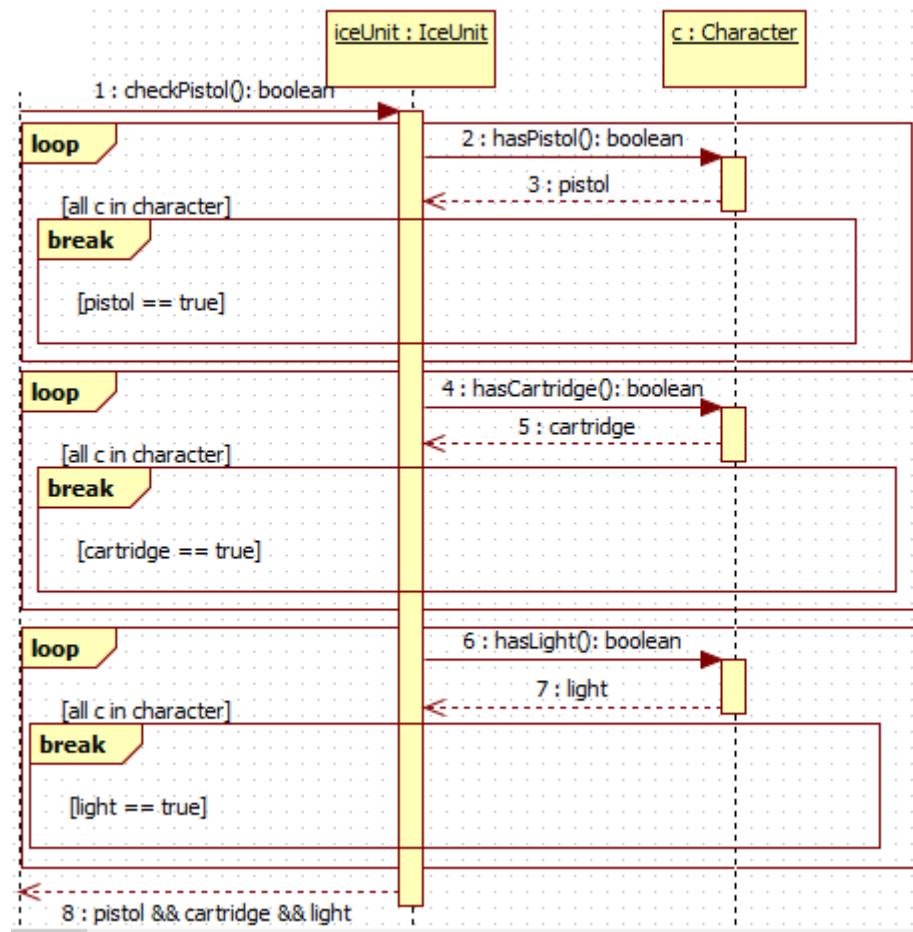
### 3.4.8 break



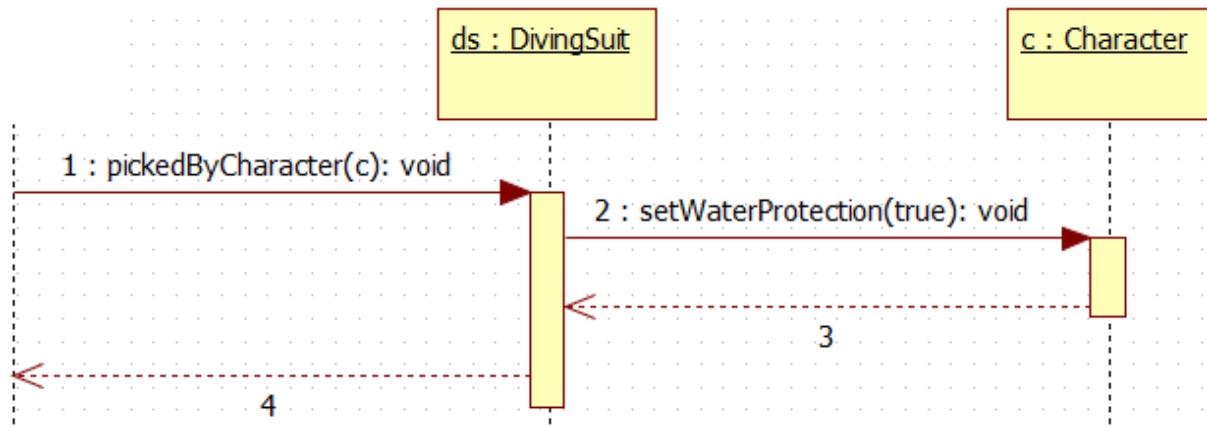
### 3.4.9 call\_for\_pull



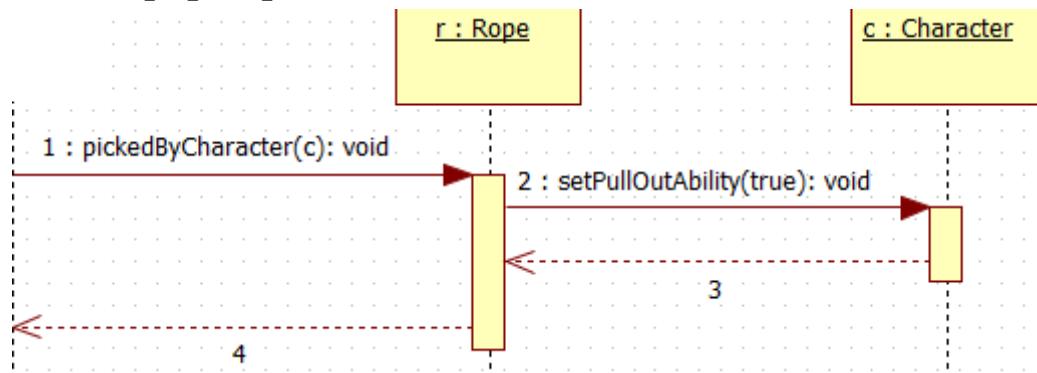
### 3.4.10 check\_pistol



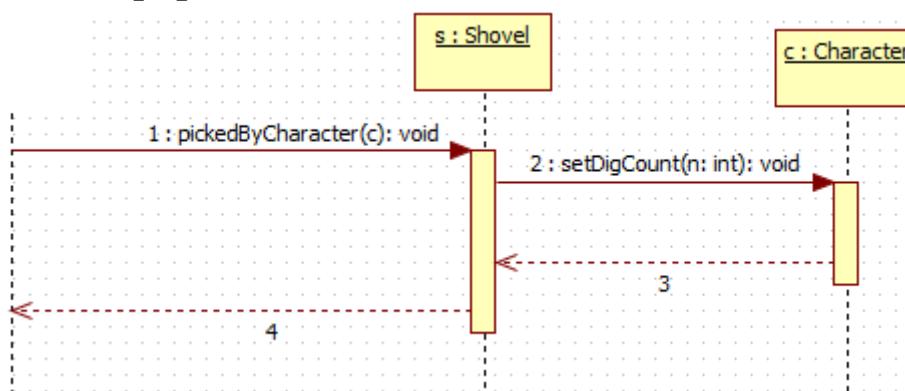
### 3.4.11 equip\_divingsuit

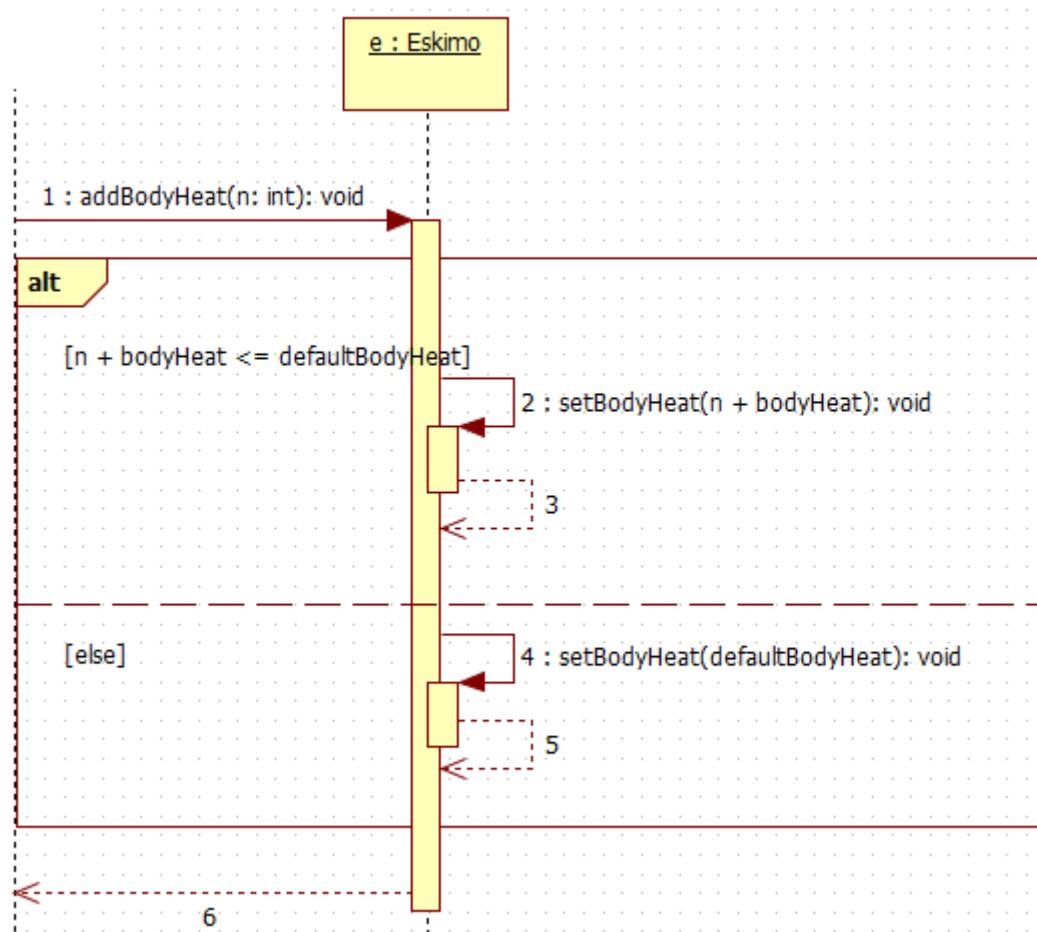


### 3.4.12 equip\_rope

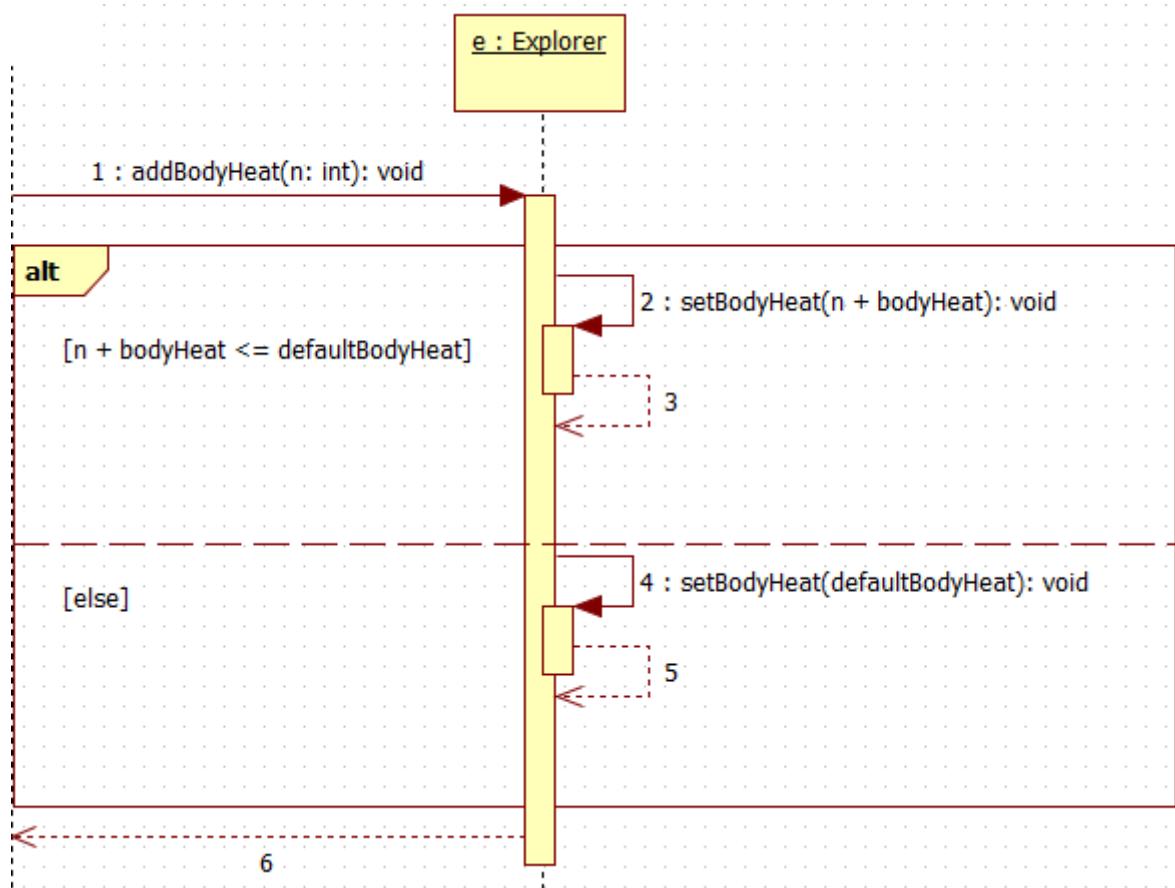


### 3.4.13 equip\_shovel

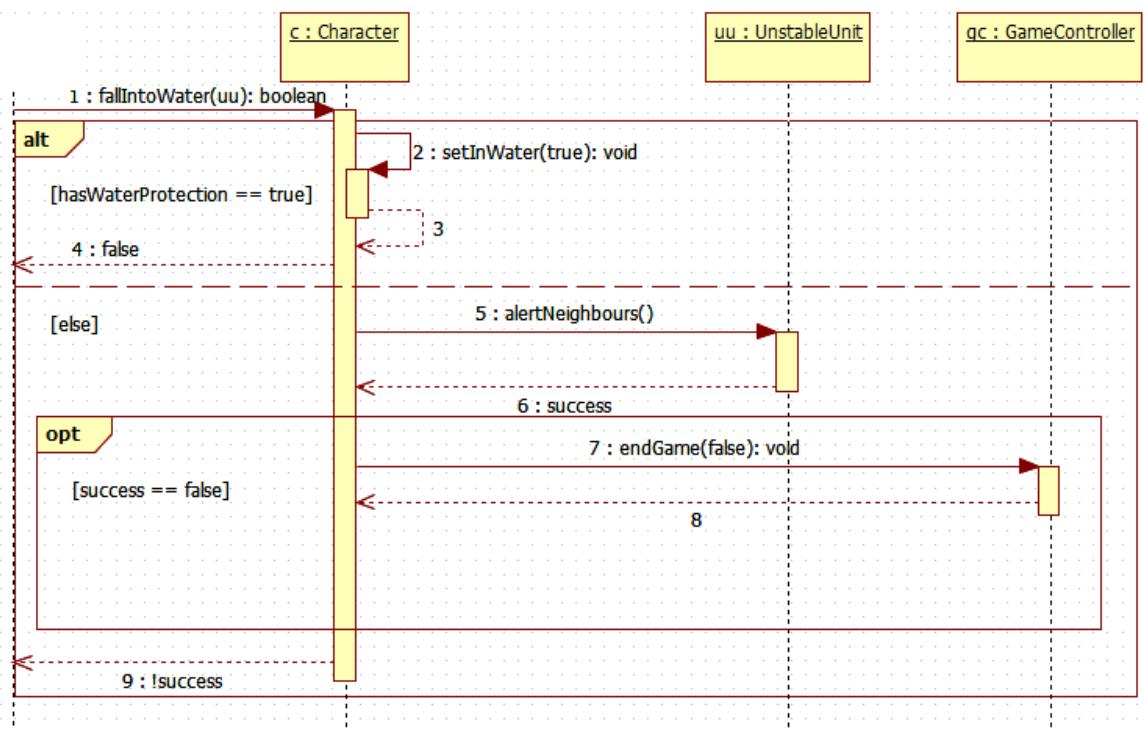


**3.4.14 eskimo\_add\_heat**

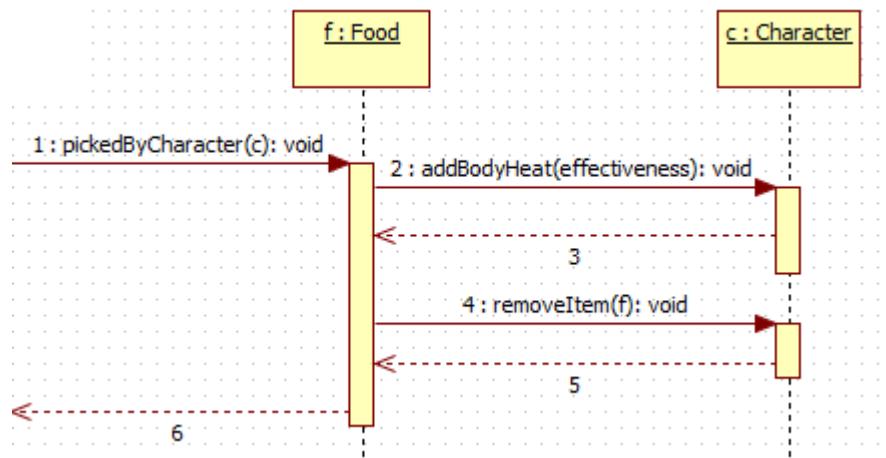
### 3.4.15 explorer\_add\_heat



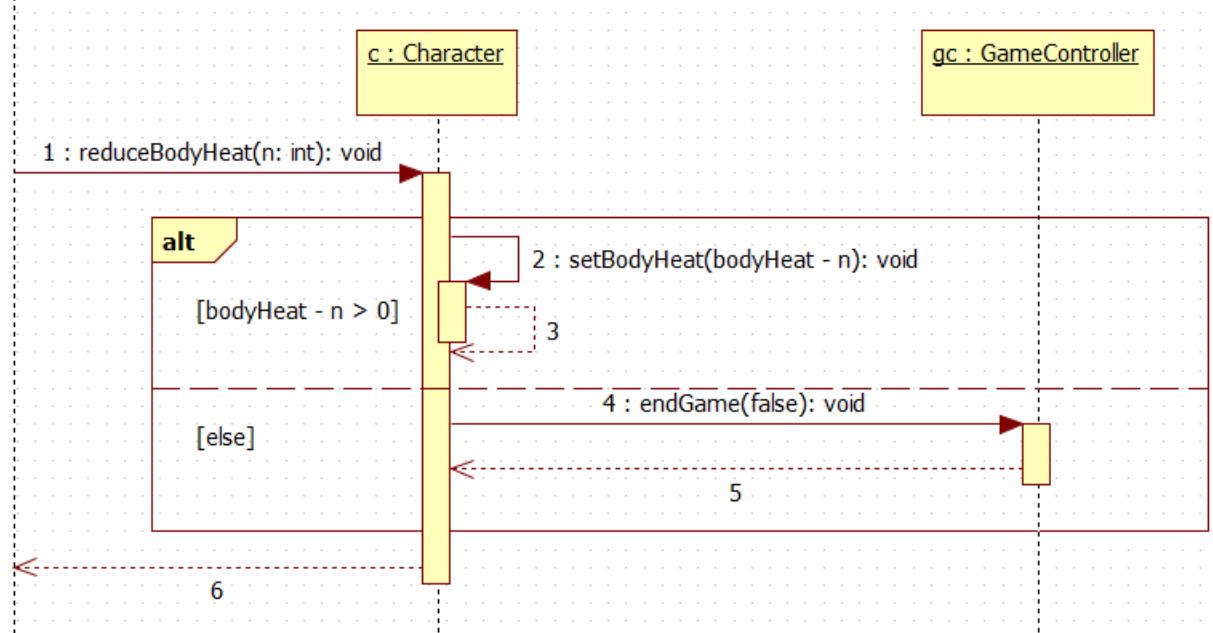
### 3.4.16 fall\_into\_water



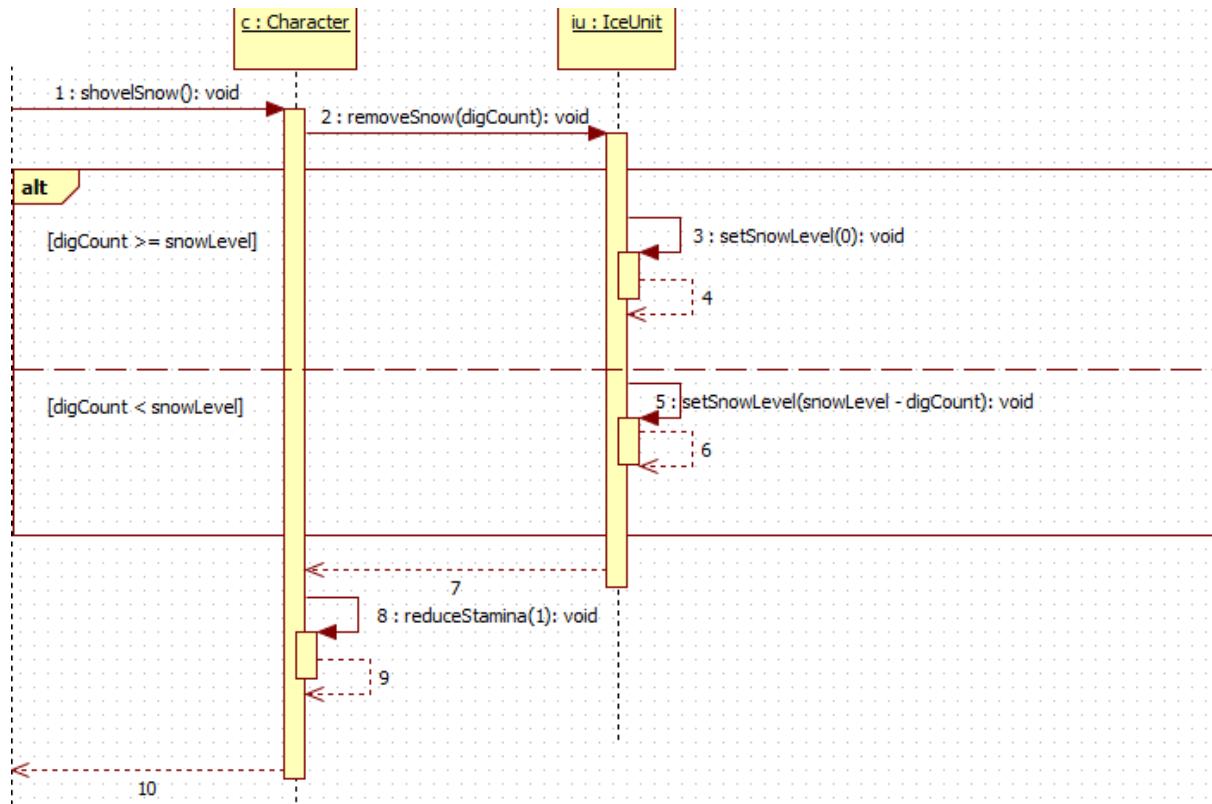
### 3.4.17 pickup\_food



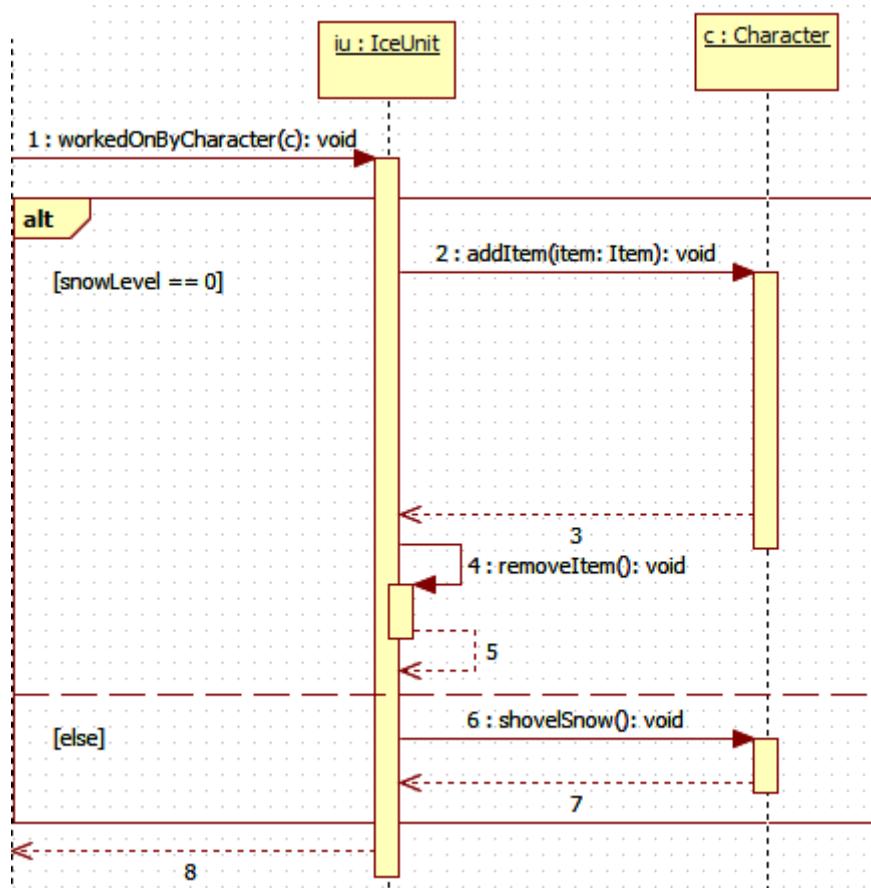
### 3.4.18 reduce\_bodyHeat



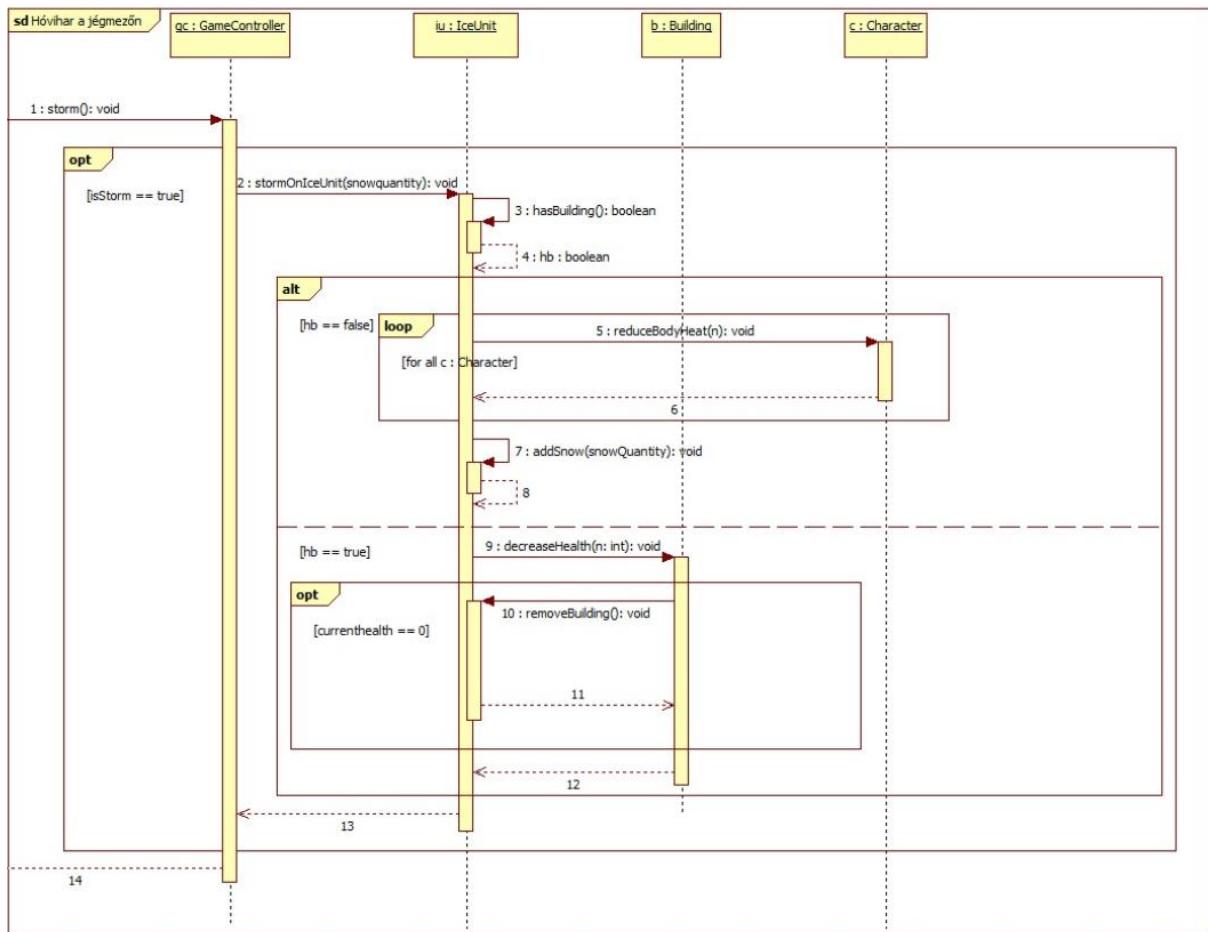
### 3.4.19 shovel



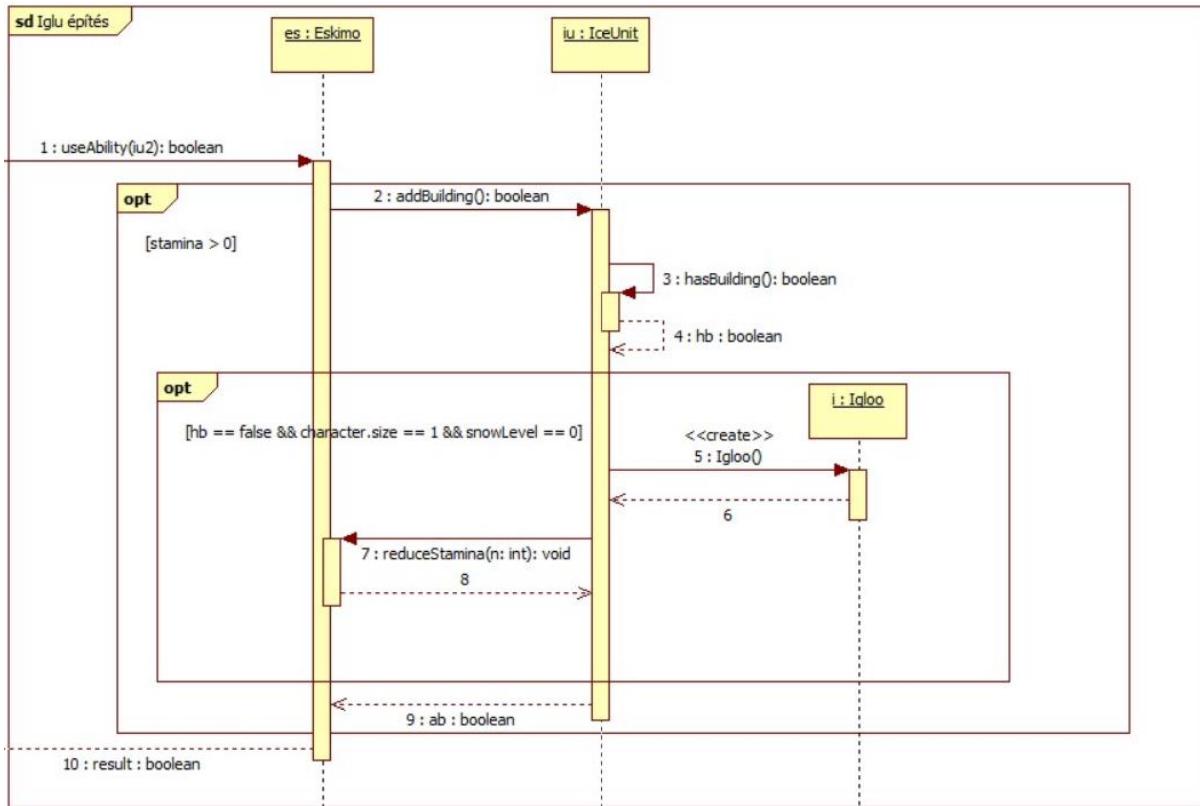
### 3.4.20 worked\_on



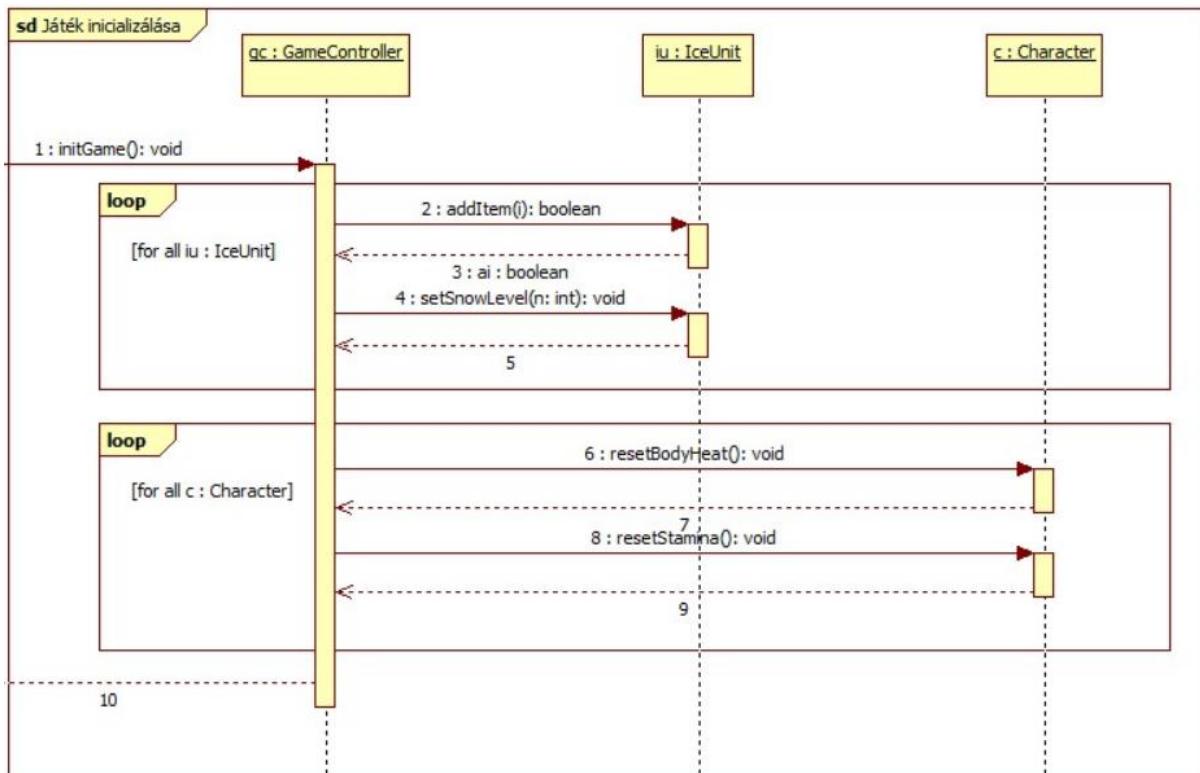
### 3.4.21 storm



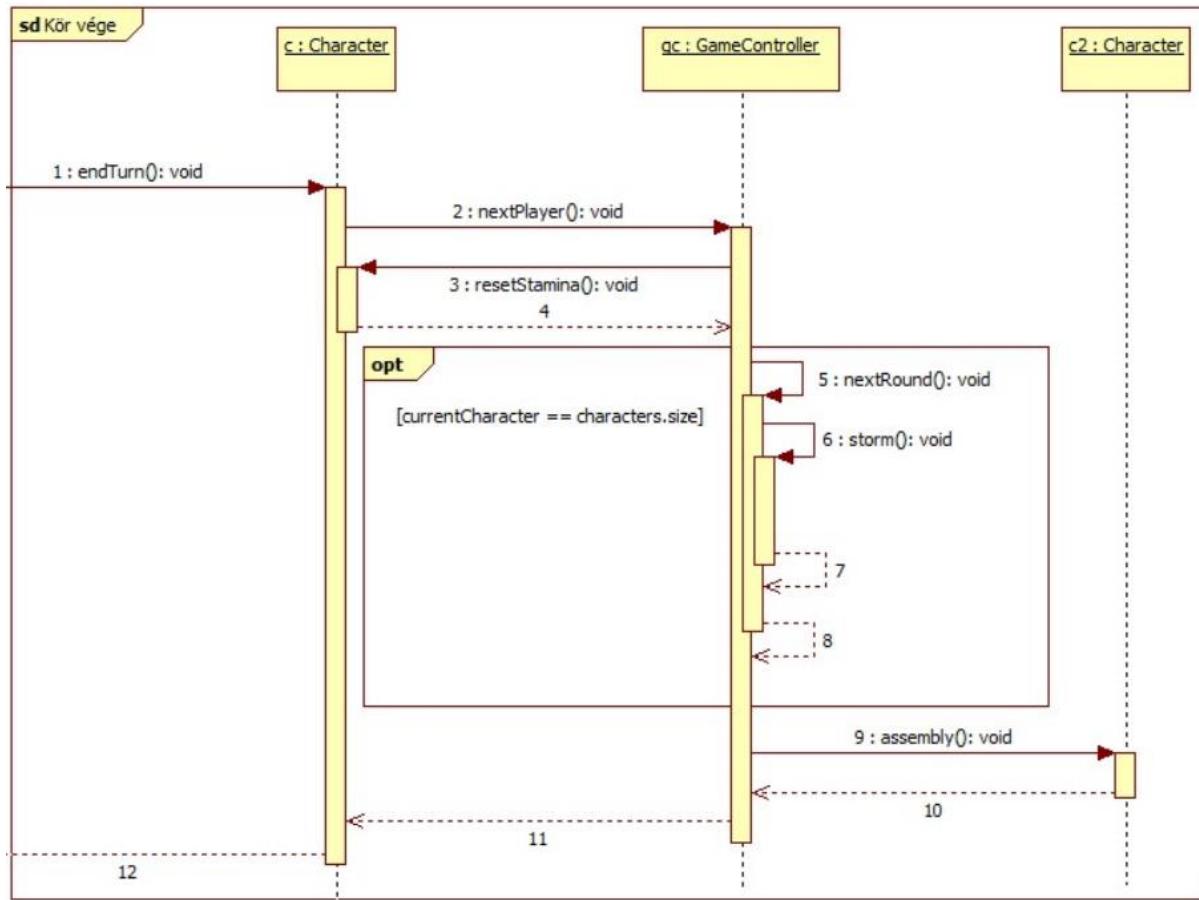
### 3.4.22 useAbility (Eskimo)



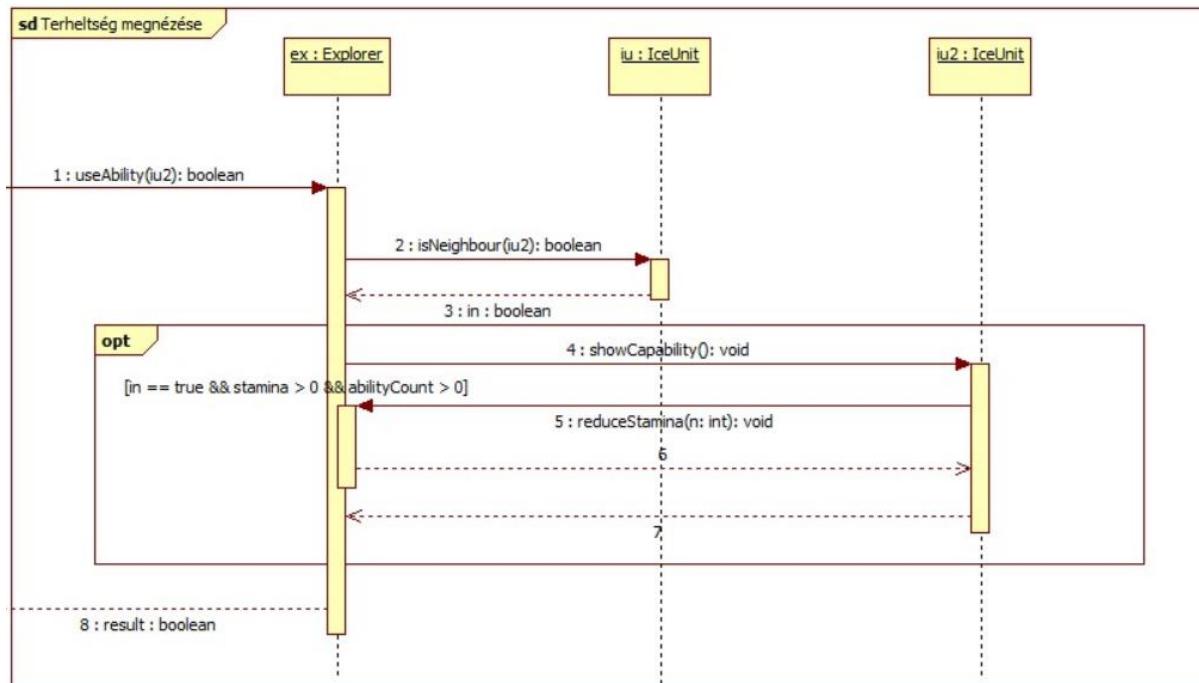
### 3.4.23 initGame



### 3.4.24 endTurn

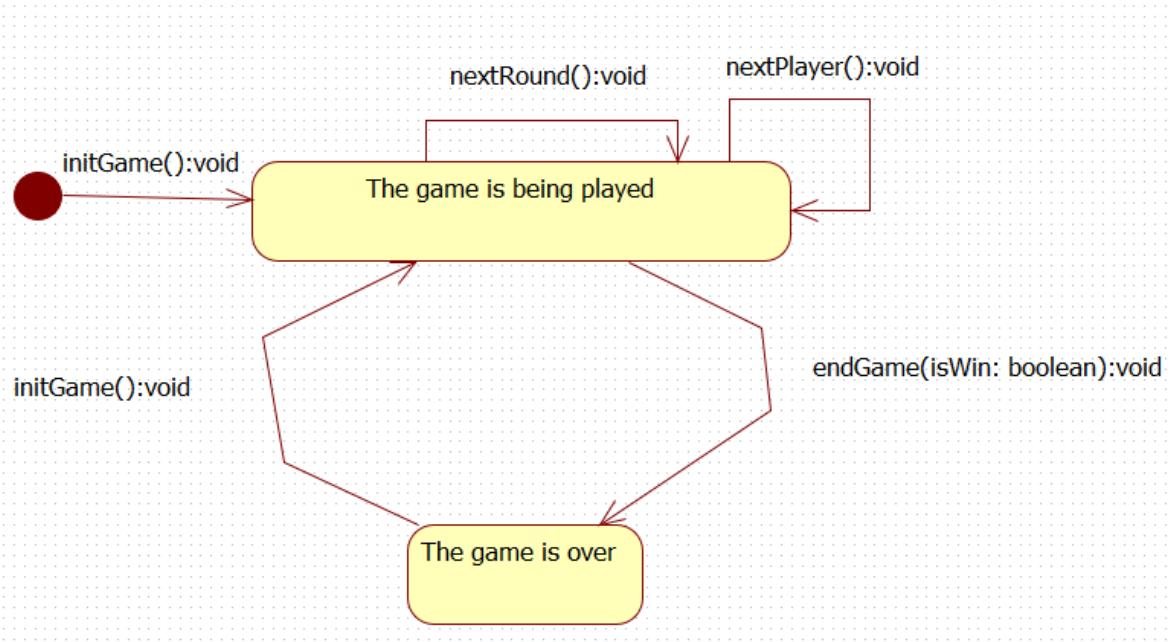


### 3.4.25 useAbility (Explorer)

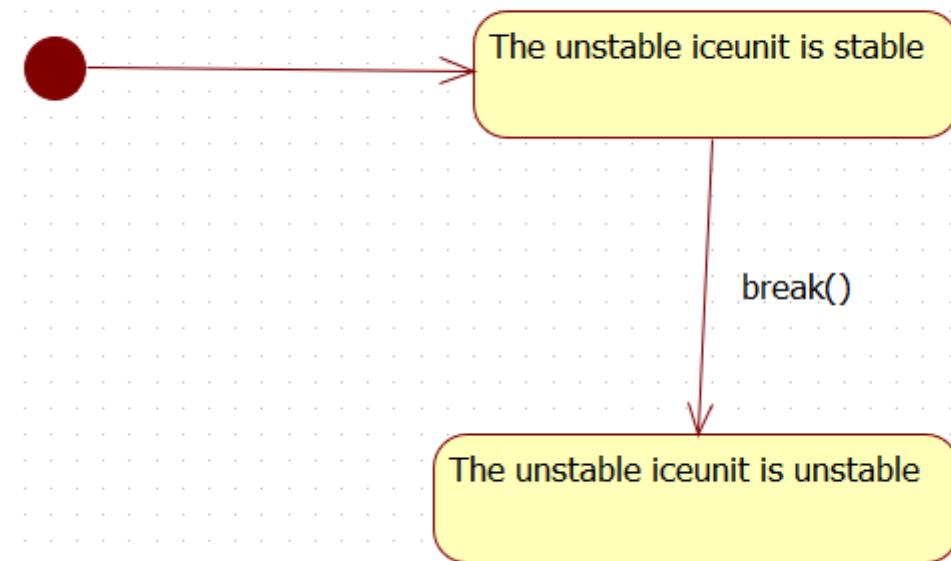


### 3.5 State-chartok

#### 3.5.1 Start Game



#### 3.5.2 IceUnit stability



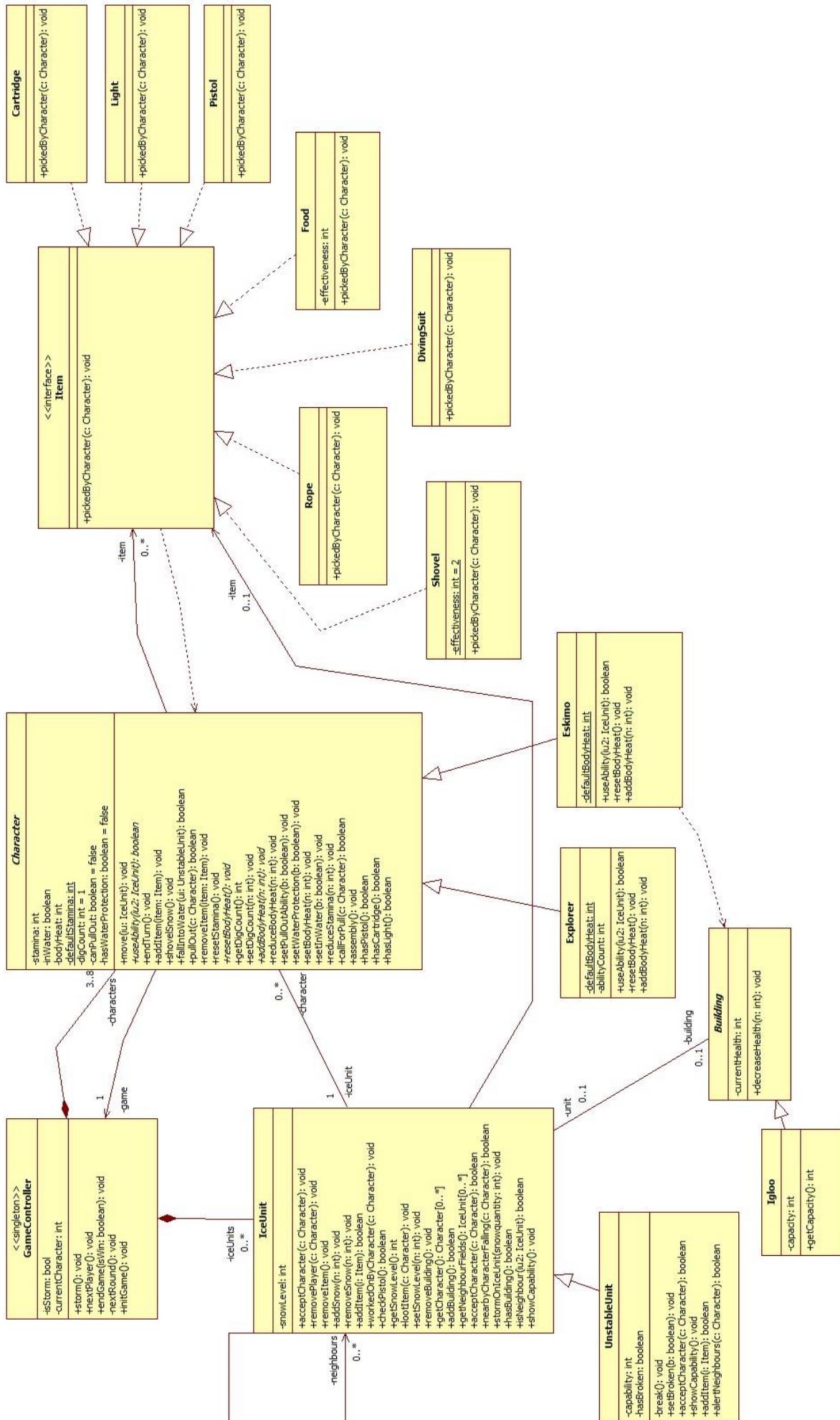
### 3.6 Napló

Kezdet	Időtartam	Résznevők	Leírás
2020.02.24. 15:30-18:30	3 óra	Iuhos Kaposvári Labancz Sulyok Traxler	Értekezlet: A feladat megbeszélése, munkafeladatok kiosztása, a program alapvető struktúrájának megteremtése
2020.02.25 15:00-19:00	4 óra	Sulyok	Tevékenység: Osztálydiagram kezdetleges elkészítése
2020.02.25 17:30-19:30	2 óra	Sulyok	Tevékenység: Alapvető szekvenciák felvázolása
2020.02.26 16:00-20:00	4 óra	Labancz	Tevékenység: Osztálydiagram bővítése
2020.02.27 15:00-17:30	2.5 óra	Sulyok	Tevékenység: Objektum katalógus elkészítése
2020.02.28 12:00-15:00	3 óra	Labancz	Tevékenység: Osztályok leírása
2020.02.28 13:00-14:00	1 óra	Iuhos Kaposvári Traxler	Értekezlet: Discord értekezlet a szekvencia diagramok felosztásáról
2020.02.28 15:00-20:00	5 óra	Traxler	Tevékenység: Szekvencia diagramok készítése
2020.02.28 15:00-19:00	4 óra	Kaposvári	Tevékenység: Szekvencia diagramok készítése
2020.02.29 10:00-13:00	3 óra	Iuhos	Tevékenység: Szekvencia diagramok készítése
2020.02.29 14:00-16:30	2.5 óra	Iuhos Kaposvári Labancz Sulyok Traxler	Értekezlet: Eddigi munka átbeszélése és ellenőrzése, megmaradt feladatok felosztása
2020.02.29 15:00-18:30	3.5 óra	Labancz	Tevékenység: Osztálydiagram bővítése, javítása

2020.02.29 15:30-19:00	3.5 óra	Iuhos	Tevékenység: Szekvencia diagramok javítása
2020.02.29 15:30-19:30	4 óra	Kaposvári	Tevékenység: Szekvencia diagramok készítése
2020.02.29 16:00-20:30	4.5 óra	Traxler	Tevékenység: Szekvencia diagramok készítése
2020.02.29 17:00-20:00	3 óra	Labancz	Tevékenység: Osztálydiagram bővítése új függvényekkel
2020.02.29 17:00-18:00	1 óra	Sulyok	Tevékenység: Segítség szekvencia diagramok készítésében
2020.03.01 10:00-13:00	3 óra	Iuhos	Tevékenység: Szekvencia diagramok készítése
2020.03.01 13:00-15:00	2 óra	Iuhos Kaposvári Sulyok Traxler	Értekezlet: Szekvencia diagramok ellenőrzése
2020.03.01 15:30-19:00	3.5 óra	Kaposvári	Tevékenység: Szekvencia diagramok javítása
2020.03.01 17:00-19:00	2 óra	Traxler	Tevékenység: Szekvencia diagramok készítése, javítása
2020.03.01 17:00-19:00	2 óra	Iuhos	Tevékenység: Dokumentáció készítése
2020.03.01 17:30-19:00	1.5 óra	Sulyok	Tevékenység: Dokumentáció készítése
2020.03.01 19:00-20:00	1 óra	Labancz Sulyok	Megbeszélés: Dokumentáció tartalmi és formai ellenőrzése
2020.03.01 20:30-21:00	0.5 óra	Sulyok	Tevékenység: Dokumentáció formázása

### 3. Analízis modell kidolgozása

## *the\_minesweepers*



# **4. Feladat: Analízis modell kidolgozása 2. - beadás**

**14 – the\_minesweepers**

Konzulens:  
**Fetter László**

## **Csapattagok**

Iuhos Zoltán	J8LOUL	zoltan.iuhos@gmail.com
Kaposvári Márk Martin	RAYVF8	xmarko45@gmail.com
Sulyok Botond	BM2BBM	botond.sulyok99@gmail.com
Labancz Tamás Dominik	A9HZBF	labancz.tamas@gmail.com
Traxler Bálint	IXNOYB	traxlerbalint@gmail.com

2020.03.09.

## 4. Analízis modell kidolgozása

### 4.1 Objektum katalógus

#### 4.1.1 Cartridge (patron)

A jelzőpisztoly egyik alkatrésze, szükséges ahhoz, hogy meg lehessen nyerni a játékot, a jégtábláról lehet felvenni.

#### 4.1.2 DivingSuit (búvárruha)

Ez által átvészethető a tengerbe esés, de valamekkora valószínűsséggel tönkre tud menni. Alapból a jégtáblákon van és onnan vehetik fel a karakterek.

#### 4.1.3 Eskimo (eszkimó)

Az eszkimó is egy választható karakter a játékban, azonban a képessége eltér a kutatójától, ő iglut tud építeni a mezőre, amin áll és ebben átvészethető a hóvihar.

#### 4.1.4 Explorer (kutató)

A játékban a sarkkutatót reprezentálja, egy vagy több játékos is irányíthatja, legfőbb célja a túlélés. Tud cselekedni, azaz különböző munkákat végez: léphet jégtábláról jégtáblára, használhatja a képességét (magnézheti egy szomszédos jégtábla vagy amin áll annak a teherbíró képességét), tárgyakat vehet fel és használhat, de bele tud esni a tengerbe is.

#### 4.1.5 Food (étel)

Ez egy olyan doleg, ami a karakterek testhőjét növeli. Amikor felszedik a jégtábláról, akkor egyből fel is használják.

#### 4.1.6 IceUnit (jégtábla)

A játékban ez az alapegység, amin a karakterek állhatnak és haladhatnak ezeken. Emellett tárgyak is lehetnek rajtuk. Kezeli a rajta lévő vagy rálépő karakterek utasításait. A játékosokat nehezíti, hogy hó is fedheti, ami elfedi a tárgyakat.

#### 4.1.7 Igloo (iglu)

Az eszkimók tudnak építeni iglukat, ami gyakorlatilag egyfajta épületként értelmezhető. Védelmet nyújt a hóviharok ellen.

#### 4.1.8 Instable (instabil jégtábla)

Az sima jégtáblák között helyezkedhet el egy ilyen objektum. Lényege, hogy csak bizonyos számú karaktert bír el és ha túllépte a teherbíró képességét, akkor átfordul (beszakad) és tenger lesz belőle, amibe a rajta álló játékosok belesesnek. Amikor beszakad és tenger lesz belőle, akkor már nem léphetnek rá játékosok.

#### 4.1.9 Light (jelzőfény)

A jelzőpisztoly egyik alkatrésze, szükséges ahhoz, hogy meg lehessen nyerni a játékot, a jégtábláról lehet felvenni.

#### 4.1.10      **Pistol (pisztoly)**

A jelzőpisztoly egyik alkatrésze, szükséges ahhoz, hogy meg lehessen nyerni a játékot, a jégtábláról lehet felvenni.

#### 4.1.11      **Rope (kötél)**

Ez a jégtáblákon megtalálható tárgy arra képes, hogy aki felveszi, az ezt használva kimenekíthat egy vízbe esett karaktert, aki a vízbe esés pillanatában egy vele szomszédos jégtáblán helyezkedett el.

#### 4.1.12      **Shovel (lapát)**

A játékban ez egy felvehető eszköz, ami segíti a játékost azzal, hogy a karaktere több havat tud ellapítani. Tehát fel lehet venni és a használata előnyhöz juttatja a játékost.

### 4.2 **Statikus struktúra diagramok**

Megjegyzés: A nevesített asszociációvégekhez további *implicit getter* és *setter* függvények tartoznak, amelyeket az implementációtól függően lehet megvalósítani.

Az osztálydiagramot az átláthatóság kedvéért egy külön oldalon csatoljuk.

## 4.3 Osztályok leírása

### 4.3.1 GameController

- **Felelősség**

A játék indításáért, irányításáért, befejezéséért és az alapvető működéséért felel ez az osztály. Az ő feladata, hogy a karakterek egymás után körökre osztva tevékenykednek. Az ő feladata jelezni a katasztrófáknak, hogy olyan időpillanatba jutott a játék, hogy bekövetkezésük szabályos lenne.

- **Attribútumok**
  - – **StormyField[0..\*] disastrousFields:** Eltárolja azokat a pályákat, amiket érhet vihar.
  - – **Commandable [1..\*] commandables:** Az irányítható dolgok összessége.
- **Metódusok**
  - **void startGame():** A metódus feladata a játék inicializálása, megfelelő állapotba hozása a játék elkezdéséhez.
  - **void endGame(boolean isWin):** A metódus feladata, hogy a paraméterként kapott változó alapján megfelelően befejezze a játékot.
  - **void nextCharacter():** A metódus feladata, hogy a segítségével változtassuk az éppen körön lévő karaktert.
  - **void addDisastrousNotifiable(DisastrousField df):** A metódus feladata, hogy a kapott paramétert eltároljuk az ismert katasztrófával sújtott pályák között.
  - **void notifyDisastrouseses():** A katasztrófával sújtott pályák értesítése, hogy a játék olyan szakaszba ért, ahol a szabályok szerint megtörténhet valamilyen katasztrófa (a jelenlegi követelmények szerint ez a katasztrófa a hóvihar).

### 4.3.2 IceField

- **Felelősség**

Tárolja, hogy milyen egységekből épül fel a pálya, és az egységeket vezérli.

- **Interfészek**
  - StormyField
- **Attribútumok**
  - **IceUnit[1..\*] iceUnits:** A játék mezői.
  - **Strikeable[1..\*] strikeables:** minden olyan objektum a játékban, amire hathat katasztrófa.

- **Stander[1..\*] standers:** minden egyes mezőn létező objektum.
- **Metódusok**
  - **void unleashStorm():** A StormyField interfész metódusának implementációja, a függvény hatására az összes strikable interfészű objektumot a vihar hatásával sújtja.

### 4.3.3 IceUnit

- **Felelősség**  
A jégtáblát reprezentáló osztály. Kezeli ezt a jégtáblát, azaz a jégtáblán levő hó mennyiségét, és azt, hogy mi helyezkedik el rajta. Az ő felelőssége, hogy meghatározza, hogy rajta milyen műveleteket lehet végrehajtani, azaz hogyan lehet vele interakcióba kerülni.
- **Ősosztályok**
  - BuildingPlot
- **Interfészek**
  - Digsite, CommunicationPlatform, Strikeable
- **Attribútumok**
  - **Puller[0..\*] pullers:** A mezőn álló minden olyan objektum, aki rendelkezik az elhúzás képességevel.
  - **Stander[0..\*] standers:** minden a mezőn tartózkodó objektum
  - **Pickable[0..1] pickable:** A mező által tárolt, tőle elvehető objektum.
  - **int snow:** Az attribútum azt méri, hogy hány réteg hó található ezen a mezőn.
  - **-IceUnit [0..\*] neighbours:** A tábla eltárolja a szomszédait.
  - **ComponentMaster[0..\*] masters:** A jégmezőn tartózkodó komponens mesterek.
- **Metódusok**
  - **void acceptStander(Stander s, IceUnit from):** Hozzáadja az eltárolt mezőn tartózkodó objektumokhoz az s paramétert, értesíti a from nevű paraméter mezőt, hogy elhagyta őt az s paraméter.
  - **void removeStander(Stander s):** Ezzel a metódussal lehet értesíteni az objektumot, hogy a rajta álló s objektum őt elhagyta.
  - **void removePickable():** Az eltárolt felvehető tárgy eltávolítása a metódus hatása.
  - **void show():** A metódus hatása, hogy hozzáférhetővé válik az az információ, hogy az objektumon hány réteg hó van.
  - **void addPuller(Puller pu):** Eltároljuk a paraméterként kapott objektumot azokkal az objektumokkal, akik a mezőre tudnak húzni más objektumokat.

- **void removePuller(Puller pu):** Eltávolítjuk a paraméterként kapott objektumot azokból az objektumokból, akik a mezőre tudnak húzni más objektumokat.

#### 4.3.4 UnstableIceUnit

- **Felelősség**

Egy instabil jégtáblát reprezentáló osztály. A sima jégtáblához képest annyiban tér el, hogy egy bizonyos terhelés hatására át tud fordulni, ekkor a rajta levő dolgok a vízbe esnek.

- **Ősosztályok**

*IceUnit*

- **Attribútumok**

- **int capacity:** Azt jelzi, hogy mennyi Stander-t bír el a jégtábla, ha ezt az értéket átlépi, akkor az instabil jégtábla beszakad.
- **boolean broken:** A jégtáblának két állapota lehet, beszakadadt vagy nem beszakadt. Ez a tagváltozó ezt az információt jelöli.

- **Metódusok**

- **void acceptStander(Stander s, IceUnit from):** Hozzáadja az eltárolt mezőn tartózkodó objektumokhoz az s paramétert, értesíti a from nevű paraméter mezőt, hogy elhagyta őt az s paraméter. Továbbá dönt a jégtábla beszakadásának sorsáról a terhelés és kapacitás függvényében.

#### 4.3.5 Storm

- **Felelősség**

A vihart reprezentáló osztály. Ez az osztály egy katasztrófa, aminek a felelőssége az, hogy megcsapja viharral a jégmezőt.

- **Interfészek**

*Disaster*

- **Metódusok**

- **void strikeStrikeable(Strikeable[1..\*] s):** A vihar lecsap a paraméterben kapott lecsapható dolgakra.

#### 4.3.6 BuldingPlot

- **Felelősség**

A jégmezőn levő épülethelyeket kezelő absztrakt osztály.

- **Attribútumok**
  - **Building building:** Tárolja a jégmezőn levő épületet.
- **Metódusok**
- **void build(Building b):**
- **void destructBuilding():** Szétesik az épület, ez miatt átállítja az épülethelyen levő épületet síkra(Plains osztály).
- **void searchShelter(Refugee r):** Az épülethelyen a paraméterben kapott menekült keres menedéket egy épületben.
- **void buildingStruckByStorm():** Ha a vihar éri ezt az épülethelyet, akkor megcsapja a rajta levő épületet.

#### **4.3.7 Building**

- **Felelősség**

Az épületeket kezelő absztrakt osztály.

- **Attribútumok**
  - **-BuildingPlot plot:** Eltárolja, hogy az épület melyik épülethelyen található.
- **Metódusok**
  - **void getIn(Refugee r):** Bemegy az épületbe a paraméterben kapott menekült. Ha tele van az épület, akkor nem tud bemenni.
  - **void struckByStorm():** A vihar elkapja az épületet és az épület típusától függően történik valami az épüettel.

#### **4.3.8 Plains**

- **Felelősség**

Egy épület, ahol nincs semmi. Alapból minden jégtáblán „sík épület” van, és kicsérélődik , ha építenek rá egy másik épületet, illetve erre cserélődik, ha szétesik egy épület.

- **Ősosztályok**

Building

#### **4.3.9 DestructableBuilding**

- **Felelősség**

Az elpusztítható épületeket kezelő absztrakt osztály.

- **Ősosztályok**

Building

- **Attribútumok**
  - **-int health:** Tárolja az elpusztítható épületnek az életét.
- **Metódusok**
  - **void getIn(Refugee r):** Bemegy az elpusztítható épületbe a paraméterben kapott menekült. Ha tele van az épület, akkor nem tud bemenni.
  - **void struckByStorm():** A vihar elkapja az épületet és az épület típusától függően történik valami az épülettel.
  - **void decreaseHealth():** Csökkenti az életét az épületnek, illetve felelős arra is, hogy ha nincs több élete az épületnek akkor szóljon, hogy elkell pusztítani.

#### **4.3.10 Igloo**

- **Felelősség**

A játékban levő iglu elpusztítható épületet valósítja meg.

- **Ősosztályok**
  - DestuctableBuilding
- **Attribútumok**
  - **-Refugee[0..1] refugees:** Eltárolja az igluban levő menekülteket.
  - **-int capacity:** Eltárolja azt, hogy mennyi menekült lehet az igluban.
- **Metódusok**
  - **void getIn(Refugee r):** Bemegy az igluba a paraméterben kapott menekült. Ha tele van az épület, akkor nem tud bemenni.
  - **void struckByStorm():** A vihar elkapja az iglut és csökkenti az életét.

#### **4.3.11 Stander**

- **Felelősség**

Egy olyan dolgot reprezentál a játékban, ami le tud esni a jégtábláról, így ennek megfelelően a felelőssége az esés műveletének végrehajtása.

- **Attribútumok**
  - **IceUnit iceUnit:** Tárolja, hogy melyik táblán tartózkodik.
- **Metódusok**

- **void fall():** Ez által kap figyelmeztetést, hogy a le fog esni, és levezényli az esés folyamatát.
- **void fallenDown():** Ennek a metódusnak a meghívása azt jelzi, hogy a metódussal rendelkező dolog már nem esik, hanem leesett.

#### 4.3.12 Pullable

- **Felelősség**

Egy olyan dolog, amiket a játékban, akár más közreműködő segítségével húzni.

- **Ősosztályok**

Stander

- **Metódusok**

- **void reposition(IceUnit iu):** Megkapja paraméterben, hogy melyik jégtáblára húzták ki. Ezután arra a jégtáblára “mozog”.

#### 4.3.13 LocationChanger

- **Felelősség**

Egy olyan dolog, ami képes más Pullable típusú dolgokat húzni.

- **Ősosztályok**

Pullable

- **Interfészek**

Puller, FiniteActionDoer

- **Attribútumok**

- **- MoveBehaviour behaviour:** Tárolja a LocationChanger példány mozgásstílusát.

- **Metódusok**

- **void pull(Pullable p):** Ezáltal a metódus által képes más objektumokat kihúzni. Paraméterben megkapja, hogy kit vagy mit kell kihúznia.

- **void move(IceUnit goal, IceUnit from):** A mozgást megvalósító függvény.

- **void changeMoveBehaviour(MoveBehaviour mb):** Átállítjuk a példány mozgásstílusát a paraméterként kapotttra.

#### 4.3.14 Mortal

- **Felelősség**

A testhő értékének tárolása és annak csökkentése illetve növelése a felelőssége ennek az osztálynak.

- **Ősosztályok**  
LocationChanger
- **Attribútumok**
  - **-int bodyHeat:** A testhő nagyságának értékét jelölő szám.
- **Metódusok**
  - **void reduceBodyHeat(int n):** A paraméterben kapott értékkel csökkenti a testhő nagyságát.
  - **void restoreBodyHeat(int n):** A paraméterben kapott értékkel növeli a testhő nagyságát.

#### 4.3.15 Character

- **Felelősség**  
Az osztály felelőssége, hogy a játék szabályainak megfelelően implementálja az interfészeit, határozza meg az összes irányítható karakter közös működéseit.
- **Ősosztályok**  
Mortal
- **Interfészek**  
Terrestrial, ComponentMaster, Equipper, Refugee, Commandable
- **Attribútumok**
  - **int stamina:** Ennek az attribútumnak a segítségével valósítjuk azt az elvárt működést, hogy a karakter egy körben korlátozott számú műveletet végezhet.
  - **WaterProtector waterProtector:** A karakter tárgya, ami megvédi ha vízbe esik.
  - **PullerDevice pullerDevice:** A karakter tárgya, ami segítségével tud arébb húzni más objektumokat.
  - **Component components:** A karakter által eltárolt komponensek. A karakterek ezek komponensekből összerakhat teljesen új dolgokat, hogy azok segítsék a játék során.
  - **Digger digger:** A karakter tárgya, ami segítségével elvégzi az ásást.
- **Metódusok**
  - **void pull():** A metódus a karakter működésének szerinti megfelelő megvalósítása, olyan értelemben, hogy itt definiáljuk a karakter elhúzó képességét.

- **void fall():** A metódus a karakter szabályos esésének szerinti megvalósítása.
- **void reduceBodyHeat():** A metódus a szabályoknak megfelelően csökkenti a karakter testhőjét, és tesz megfelelő lépéseket a szabályoknak megfelelően.
- **void fallenDown():** A metódus megvalósítja a karakter teendőjét, ha leesett egy mezőről.
- **void move(IceUnit goal, IceUnit from):** Az osztálynak megfelelő megvalósítása ennek a funkciónak. A különbség az, hogy jelezük a mezőnek, amin tartózkodunk, hogy új ComponentMaster lépett rá.

#### **4.3.16      Explorer**

- **Felelősség**

Az osztály felelőssége megvalósítani az egyik játszható karaktertípus, a felfedező működését.

- **Ősosztályok**

Character

- **Interfészek**

*IceUnitManipulator*

- **Metódusok**

- **void useAbility(IceUnit iu):** Megméri, hogy mennyit bír el a paraméterként kapott mező.

#### **4.3.17      Eskimo**

- **Felelősség**

Az osztály felelőssége megvalósítani az egyik játszható karaktertípus, az eszkimó működését.

- **Ősosztályok**

Character

- **Interfészek**

*IceUnitManipulator*

- **Metódusok**

- **void useAbility(IceUnit iu):** Iglut épít a kijelölt mezőre.

#### **4.3.18      Food**

- **Felelősség**

A játékban a felvehető ételt reprezentálja.

- **Interfészek**

Consumable

- **Attribútumok**

- **-int effectiveness:** Ez az érték azt adja meg, hogy mennyire hatékony az étel. Ezt a hatékonyságot úgy kell értelmezni, hogy hány testhő visszatöltésére képes.

- **Metódusok**

- **consume(m: Mortal): void :** A metódus hatására az ételt elfogyasztó, paraméterként kapott Mortalnak az étel effektívségének alapján testhőt tölt vissza.

### **4.3.19 Digger**

- **Felelősség**

Absztrakt osztály, aminek felelőssége, hogy az ásás műveletét levezényelje.

- **Attribútumok**

- **-int effectiveness:** Az ásás hatékonyságát jelöli.

- **Metódusok**

- **void dig(Character c, IceUnit iu):** Az ásást levezénlését végzi, a paraméterként megkapott IceUnit-nak meghívja a megfelelő metódusát, amivel az IceUnit ezt az interakciót kezelni tudja.

### **4.3.20 Shovel**

- **Felelősség**

Egy ásó fajta, amivel több havat lehet eltakarítani, ahhoz képest, mint ha nem rendelkeznénk ezzel az eszközzel.

- **Ősosztályok**

Digger

- **Interfészek**

Pickable

- **Metódusok**

- **void pickedByCharacter(Character c, IceUnit iu):** Amikor egy karakterhez kerül a tárgy, akkor ez metódus ennek a viselkedésnek a levezénlését végzi. Tehát beállítja a karakter osztálynak az egyik attribútumát, ami azt jelöli, hogy a karakter rendelkezik ezzel a tárggyal.

- **void dig(Character c, IceUnit iu):** Az ásást levezénylését végzi, a paraméterként megkapott IceUnit-nak meghívja a megfelelő metódusát, amivel az IceUnit ezt az interakciót kezelni tudja.

### 4.3.21 BareHand

- **Felelősség**

Egy ásó fajta, tud ájni, de nem rendelkezik semmilyen további viselkedéssel.

- **Ősosztályok**

Digger

- **Metódusok**

- **void dig(Character c, IceUnit iu):** Az ásást levezénylését végzi, a paraméterként megkapott IceUnit-nak meghívja a megfelelő metódusát, amivel az IceUnit ezt az interakciót kezelni tudja.

### 4.3.22 Lung

- **Felelősség**

Azt reprezentálja, amikor a karakter nem rendelkezik semmilyen a víz ellen védő eszközzel és ekkor a karakter meg tud halni.

- **Interfészek**

WaterProtecter

- **Metódusok**

- **void protectFromWater(Terrestrial t):** A paraméterként kapott Terrestrial objektumok meghívja a megfelelő metódusát, ami a karakter fulladását idézi elő, mivel ha a karakter egy ilyen eszközzel rendelkezik, az azt jelenti, hogy nincs védve a tengertől.

### 4.3.23 DivingSuit

- **Felelősség**

Buvárruha, amit fel lehet venni és a tulajdonosa ennek köszönhetően védeeltséget élvez a vízbe esés után esetlegesen bekövetkező haláltól.

- **Interfészek**

Pickable, WaterProtecter

- **Metódusok**

- **void pickedBy(Character c, IceUnit iu):** Amikor egy karakterhez kerül a tárgy, akkor ez metódus ennek a viselkedésnek a levezénylését végzi. Tehát beállítja a karakter

osztálynak az egyik attribútumát, ami azt jelöli, hogy a karakter rendelkezik ezzel a tárggyal és beállítja azt is, hogy a Character tudni

- **void protectFromWater(Terrestrial t):** Megvédi a vízbe esett Terrestrial objektumot a fulladástól.

#### **4.3.24      Rope**

- **Felelősség**

Egy felvehető tárgy, ami képes karakterek vízből való kimentésére.

- **Interfészek**

Pickable, PullerDevice

- **Metódusok**

- **void pickedBy (Character c, IceUnit iu):** Amikor egy karakterhez kerül a tárgy, akkor ez metódus ennek a viselkedésnek a vezérolását végzi. Tehát beállítja a karakter osztálynak az egyik attribútumát, ami azt jelöli, hogy a karakter rendelkezik ezzel a tárggyal.

- **void pull(Pullable p, IceUnit iu):** A paraméterként kapott Pullable objektumot, ami akár lehet egy karakter, kihúzza a vízből. A kihúzott objektum arra az IceUnit-ra kerül, ahol a kötél tárgyának tulajdonosa is áll.

#### **4.3.25      BasicPull**

- **Felelősség**

Azt jelzi, hogy a használói nem képesek egy másik jelen, alapértelmezett esetben karaktert a vízből kihúzni.

- **Interfészek**

PullerDevice

- **Metódusok**

- **void pullByDevice(Pullable p, IceUnit iu):** Ez az osztály nem képes kihúzni a paraméterként kapott objektumot, ezt a viselkedést valósítja meg ez a metódus.

#### **4.3.26      Pistol**

- **Felelősség**

A Pistol felelőssége az, hogy a jelzőfényespisztoly komponenseként belőle összeállítható legyen a jelzőfényespisztoly.

- **Interfészek**

Component, Pickable

- **Metódusok**

- **void assemble(ComponentMaster cc)** : A metódus hatására a pisztoly komponens összeáll a belőle létrehozható AlarmPistol-vá.

#### 4.3.27 Cartridge

- **Felelősség**

A Cartridge felelőssége az, hogy a jelzőfényespisztoly komponenseként belőle összeállítható legyen a jelzőfényespisztoly.

- **Interfészek**

Component, Pickable

- **Metódusok**

- **void assemble(ComponentMaster cc)** : A metódus hatására a patron komponens összeáll a belőle létrehozható AlarmPistol-vá.

#### 4.3.28 Light

- **Felelősség**

A Light felelőssége az, hogy a jelzőfényespisztoly komponenseként belőle összeállítható legyen a jelzőfényespisztoly.

- **Interfész**

Component, Pickable

- **Metódusok**

- **void assemble(ComponentMaster cc)** : A metódus hatására a jelzőfény komponens összeáll a belőle létrehozható AlarmPistol-vá.

#### 4.3.29 AlarmPistol

- **Felelősség**

A játék célja ezen osztály egy példányának létrehozása, ez a játék megnyerését jelenti.

- **Metódusok**

- **alarmPistol()**: Amikor létrejön ebből a típusból egy példány, az a játék győzelmét jelenti.

### 4.3.30 Amphibious

- **Felelősség**

Olyan mozgásforma, aminek követésével mozoghatunk bármely mezőre, legyen az víz vagy föld.

- **Interfészek**

MoveBehaviour

- **Metódusok**

**void move(IceUnit goal, IceUnit from, LocationChanger lc):** A paraméterként kapott LocationChanger elmozgatása a paraméterként kapott célba a kezdőpontból a kétéltű mozgás szerint.

### 4.3.31 Walk

- **Felelősség**

Olyan mozgásforma, aminek követésével mozoghatunk szárazföldön.

- **Interfész**

MoveBehaviour

- **Metódusok**

**void move(IceUnit goal, IceUnit from, LocationChanger lc):** A paraméterként kapott LocationChanger elmozgatása a paraméterként kapott célba a kezdőpontból a szárazföldi mozgás szerint.

### 4.3.32 StromyField

- **Felelősség**

A játékban levő viharos jégmezőknek az interfész.

- **Metódusok**

- **void unleashStorm():** Végig megy a vihar a jégmezőn.

### 4.3.33 Disaster

- **Felelősség**

A játékban levő katasztrófáknak az interfész.

- **Metódusok**

- **void strikeStrikeable(Strikeable[1..\*] s):** A katasztrófa elkapja a paraméterben kapott lecsapható dolgokra.

#### **4.3.34 CommunicationPlatform**

- **Felelősség**

Az interfész felelőssége, hogy az azt implementáló játékbeli mezők rendelkezzenek minden metódussal, ami a különböző objektumok egymással való kommunikációjához szükséges egy mezőn keresztül.

- **Metódusok**

- **Component[0..\*] askForComponents():** A metódus felelőssége az, hogy visszaadja a mezőn létező objektumok komponenseinek összességét.
- **void callForPullOut(Pullable p):** A metódus meghívásával jelezük az összes szomszédos mezőnek, hogy a rajtuk lévő objektumoknak ki kell húzniuk a paraméterként átadott objektumot.
- **void alertNeighbourForPullOut(Pullable p):** Ezzel a metódussal a mező figyelmeztetést kap, hogy a rajta álló Puller objektumoknak ki kell húzniak a paraméterként átadott objektumot.

#### **4.3.35 Refugee**

- **Felelősség**

A játékban levő menekülteknek az interfésze.

- **Ősosztály**

Strikeable

- **Metódusok**

- **void protectionFailed():** Ha a menekültnek nincsen fedezéke a katasztrófák ellen, akkor nincsen neki protekciója és testhőt veszít.
- **void struckByDisaster():** Elkapja a katasztrófa a menekültet. A menekült elmenekülhet ez elől.

#### **4.3.36 Commandable**

- **Felelősség**

Az interfész felelőssége, hogy az irányítható karakterekhez egy irányítási felületet biztosítan.

- **Metódusok**

- **void executeCommand(String s):** A paraméternek megfelelő parancs végrehajtása.

### 4.3.37 IceUnitManipulator

- **Felelősség**

Az interfész felelőssége, hogy biztosítja az eszközöket egy IceUnit osztályú objektumon való műveletvégzéshez.

- **Metódusok**

- **void useAbility(IceUnit iu ):** A metódus, ami elvégzi a paraméterként kapott iceUniton a szükséges műveleteket.

### 4.3.38 FiniteActionDoer

- **Felelősség**

Az interfész felelőssége, hogy definiálja a működését azoknak az objektumoknak, amelyek korlátozott cselekvőképességgel rendelkeznek.

- **Metódusok**

- **void tire():** A metódus felelőssége, hogy jelezzen vele, hogy a megvalósító objektumnak valamely mértékben csökkennie kell a további munkavégző képességének.
- **void rest():** A metódus felelőssége, hogy jelezzen vele, hogy a megvalósító objektum újra munkavégzézű képességének elérte a maximumát.

### 4.3.39 Equipper

- **Felelősség**

A tárgyakat felvenni képes objektumok interfésze.

- **Metódusok**

- **void equipConsumable(Consumable cons):** Előírja a fogyasztható tárgyak felvételének megvalósítását.
- **void equipDigger(Digger d):** Előírja az ásás működésének dinamikus megváltoztatását az interfész megvalósító osztályoknak.
- **void equipWaterProtecter(WaterProtecter wp):** Előírja az víz ellen védés működésének dinamikus megváltoztatását az interfész megvalósító osztályoknak.
- **void equipPullerDevice(PullerDevice p):** Előírja a húzás működésének megváltoztatását.

### 4.3.40 Pickable

- **Felelősség**

A felvehető tárgyak számára írja elő a felvétel megvalósítását.

- **Metódusok**

- **pickedBy(Character c, IceUnit iu):** A felvezés módjának megvalósításának implementálását írja elő.

#### **4.3.41 Consumable**

- **Felelősség**

Az elfogyasztható tárgyak interfésze.

- **Ősosztály**

Pickable

- **Metódusok**

- **void consume(Mortal m):** Előírja az elfogyasztás módjának implementálását.

#### **4.3.42 WaterProtector**

- **Felelősség**

A vízbeesés ellen védő játékmechanikának megvalósítását írja elő.

- **Metódusok**

- **void protectFromWater(Terrestrial t):** Előírja a víz ellen védés játékmechanikájának megvalósítását.

#### **4.3.43 Terrestrial**

- **Felelősség**

Az interfész felelőssége, hogy működési felületet biztosítson azoknak az objektumoknak, akiknek káros vízben tartózkodniuk.

- **Metódusok**

- **void drowning():** A metódus felelőssége, hogy jelezze a megvalósító példánynak, hogy éppen fullad a vízben.

#### **4.3.44 PullerDevice**

- **Felelősség**

A húzást megvalósító eszközök interfésze.

- **Metódusok**

- **void pullByDevice(Pullable p, IceUnit iu):** A húzás működésének implementálását írja elő.

#### **4.3.45 ComponentMaster**

- **Felelősség**

Az interfész felelőssége, hogy az őt implementáló osztályok rendelkezzenek a szükséges műveletekkel a komponensek használatához.

- **Metódusok**

- **void addToComponents(Component c):** A függvény feladata, hogy a paraméterként kapott komponenst hozzáadjuk a tárolni kívánt komponenseinkhez.
- **void removeComponents(Component c):** A metódus feladata, hogy amennyiben a paraméterként kapott komponenst tárolja a mester, azt el kell dobni.
- **Components[0..\*] getOtherMastersComponents():** A metódus felelőssége, hogy kommunikációt biztosítson az implementáló osztálynak a többi mesterrel, akiktől így elkéri használatra az ő komponenseiket.

#### **4.3.46 Components**

- **Felelősség**

Az osztály feladata, hogy definiálja a komponensek működését. A komponensek olyan objektumok, amik valami más objektummá összeépíthetők. További szabály, hogy egy komponens pontosan egy építhető tárgy része lehet.

- **Attribútumok**

- **string Type:** Azonosítja egyértelműen a komponens típusát.

- **Metódusok**

- **void assemble(ComponentMaster cc):** Utasítjuk a komponenst, hogy álljon össze a céltárgyává.

#### **4.3.47 MoveBehavior**

- **Felelősség**

Az interfész felelőssége, hogy megadja a keretet különböző mozgásformák megvalósításához.

- **Metódusok**

- **void move(IceUnit goal, IceUnit from, LocationChanger lc):** A metódus felelőssége, hogy megvalósítja az elmozdulását a paraméterként átadott LocationChanger típusú objektumnak a cél és kiinduló jégmező között.

#### **4.3.48 DigSite**

- **Felelősség**

Az interfész felelőssége, hogy az őt megvalósító objektumok áshatóak legyenek.

- **Metódusok**

- **void digsnow(Character c, int n):** A metódus felelőssége, hogy biztosítja az implementáló osztályokban a működését, amikor egy karakter ás a megvalósító objektumon.

#### **4.3.49 Puller**

- **Felelősség**

Az interfész felelőssége definiálni, hogy az implementáló objektum rendelkezik az elhúzás képességevel.

- **Metódusok**

- **void pull(Pullable p):** A metódus célja, hogy a paraméterként kapott Pullable objektumot elhúzza magához.

#### **4.3.50 Stikeable**

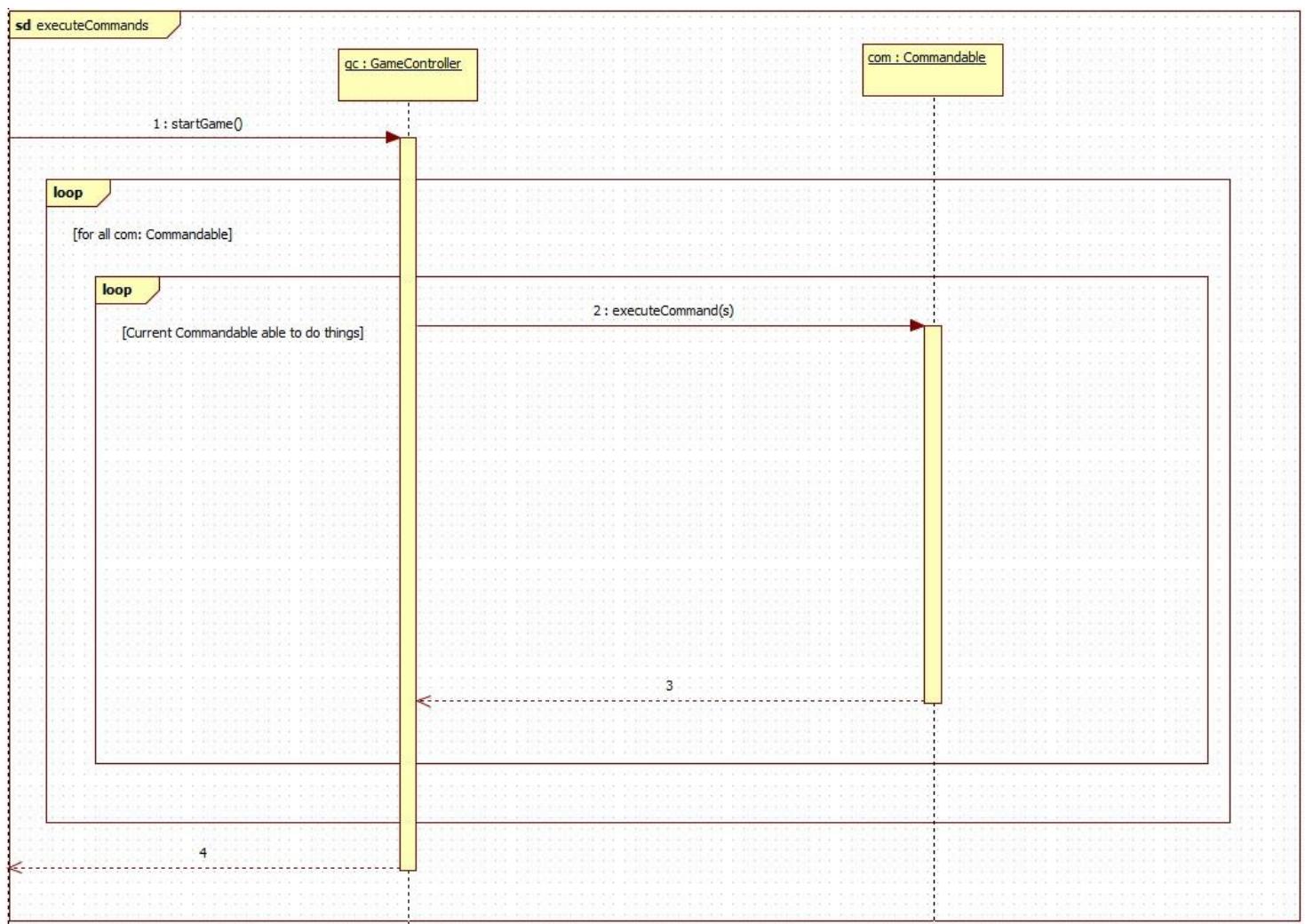
- **Felelősség**

A katasztrófák által megcsapható dolgok interfésze.

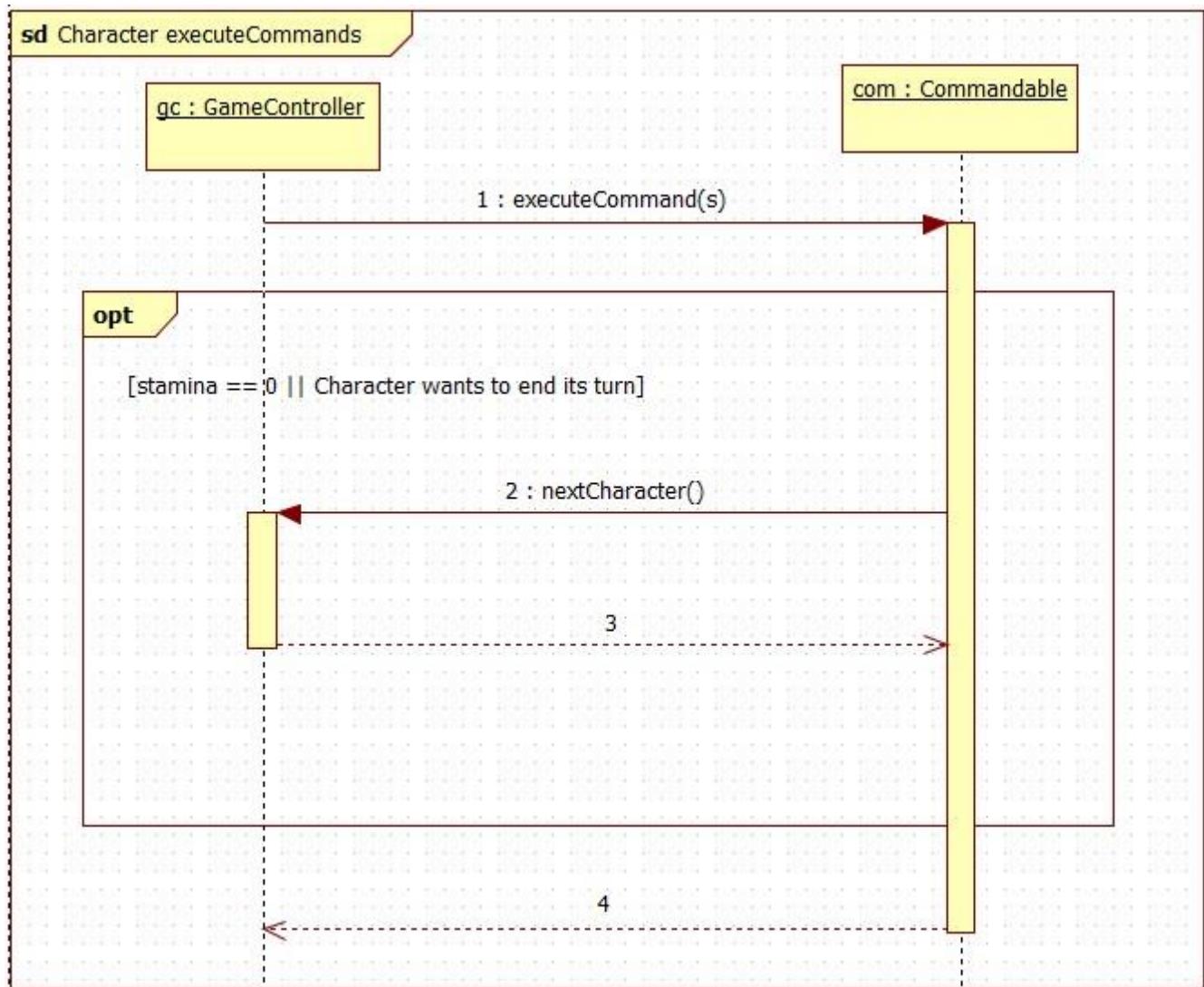
- **Metódusok**

- **void struckByDisaster():** Elkapja a katasztrófa a megcsapható dolgot.

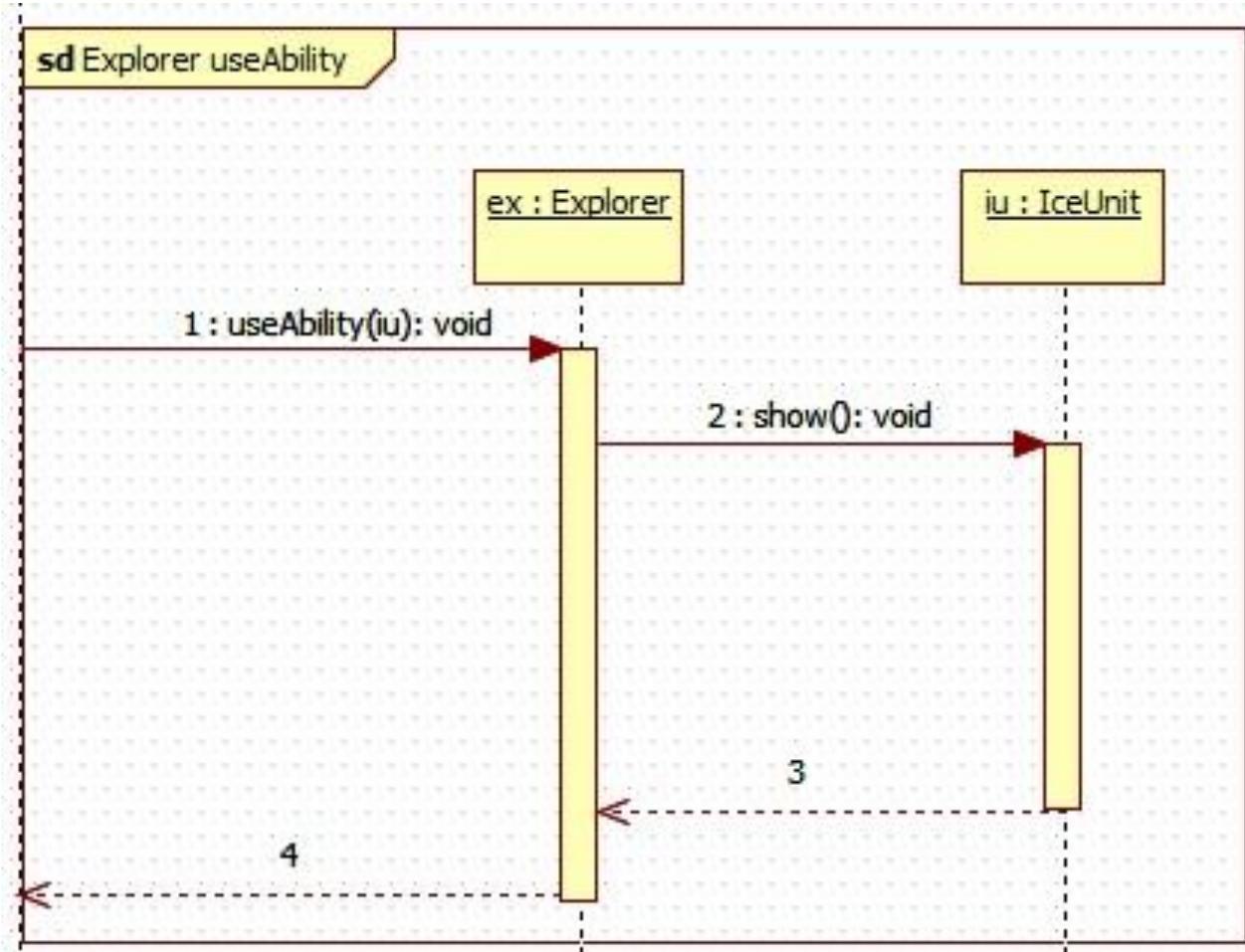
## 4.4 Szekvencia diagramok



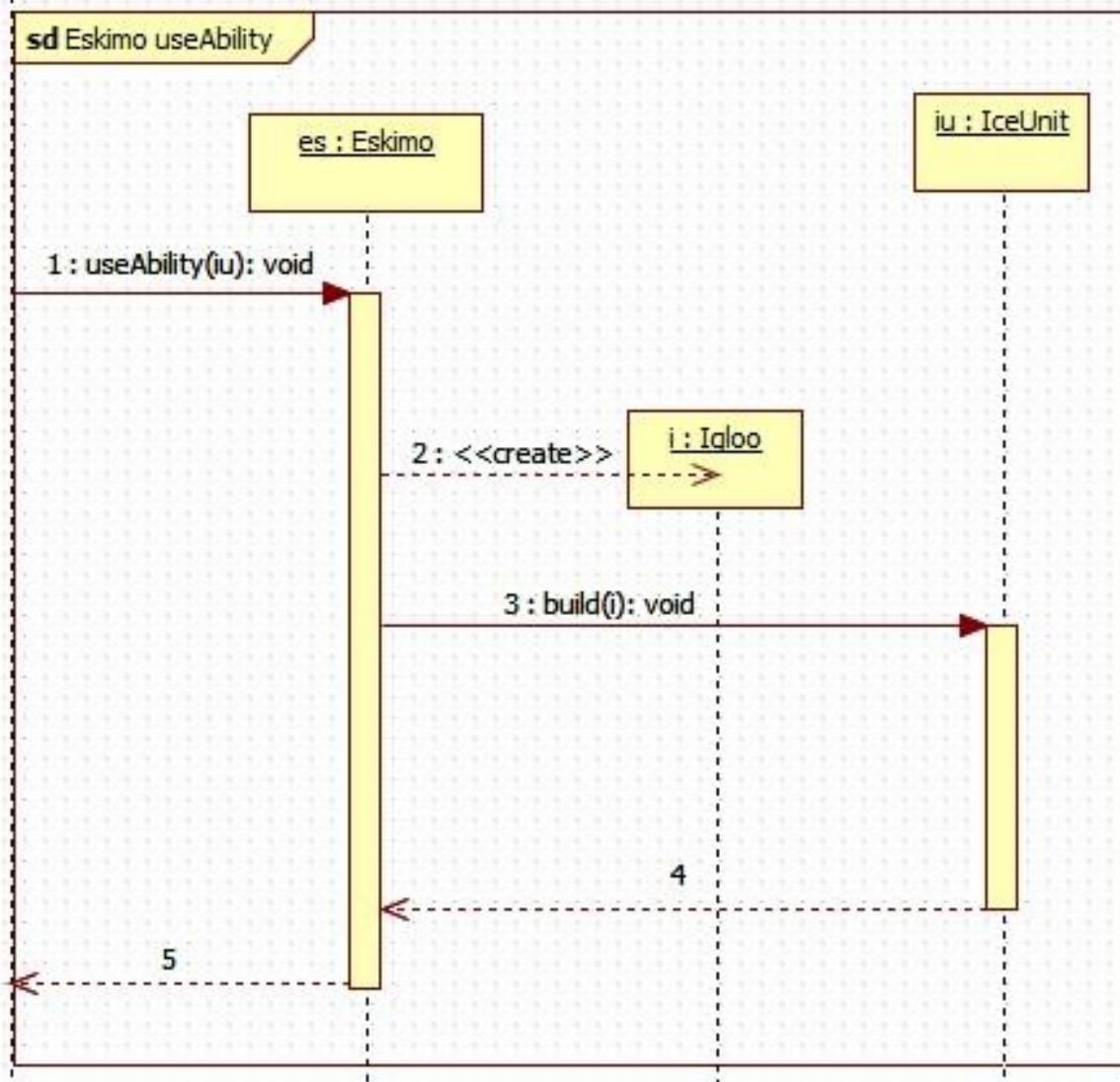
#### 4.4.2 Character executeCommands



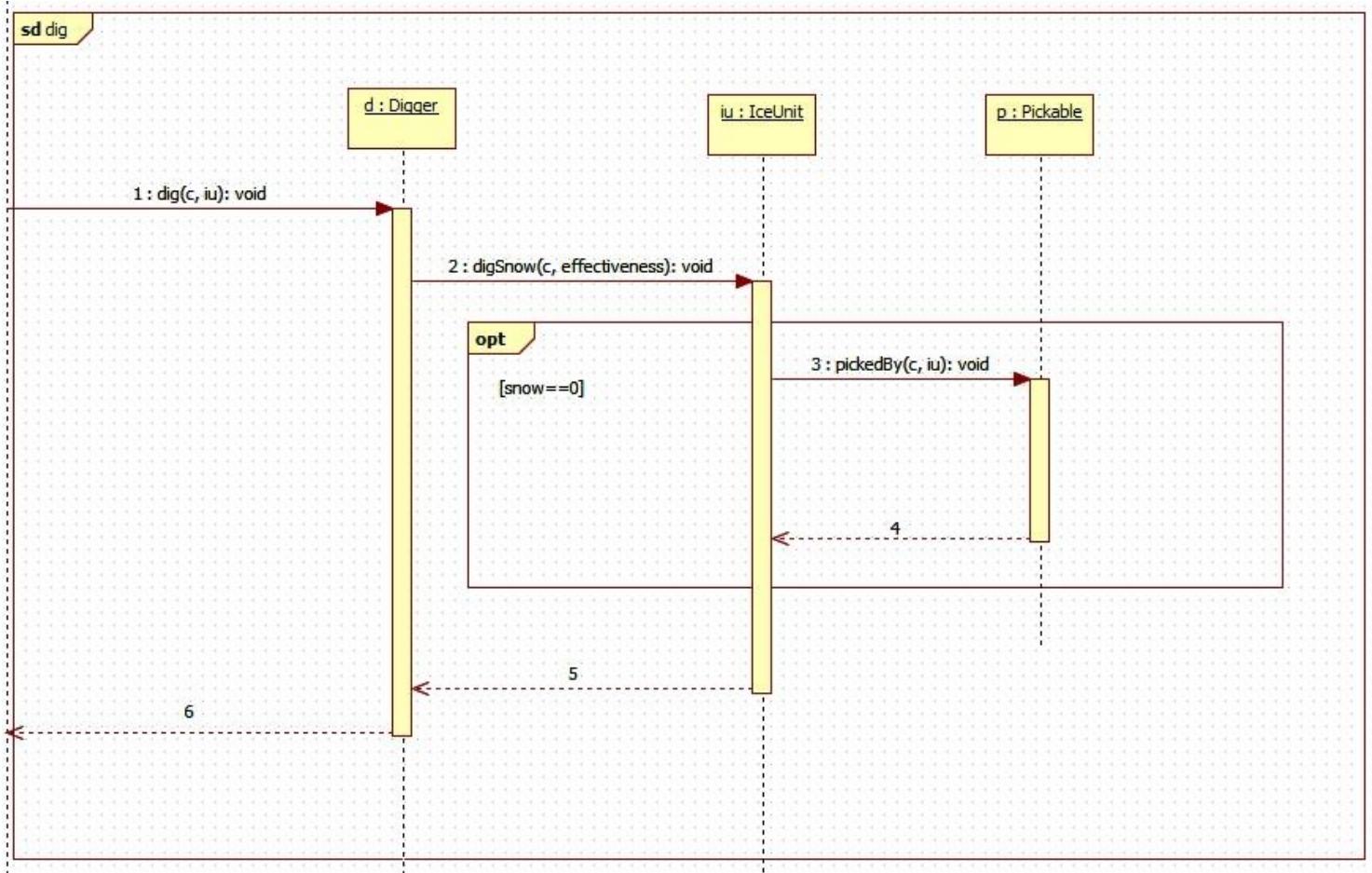
#### 4.4.3 Explorer useAbility



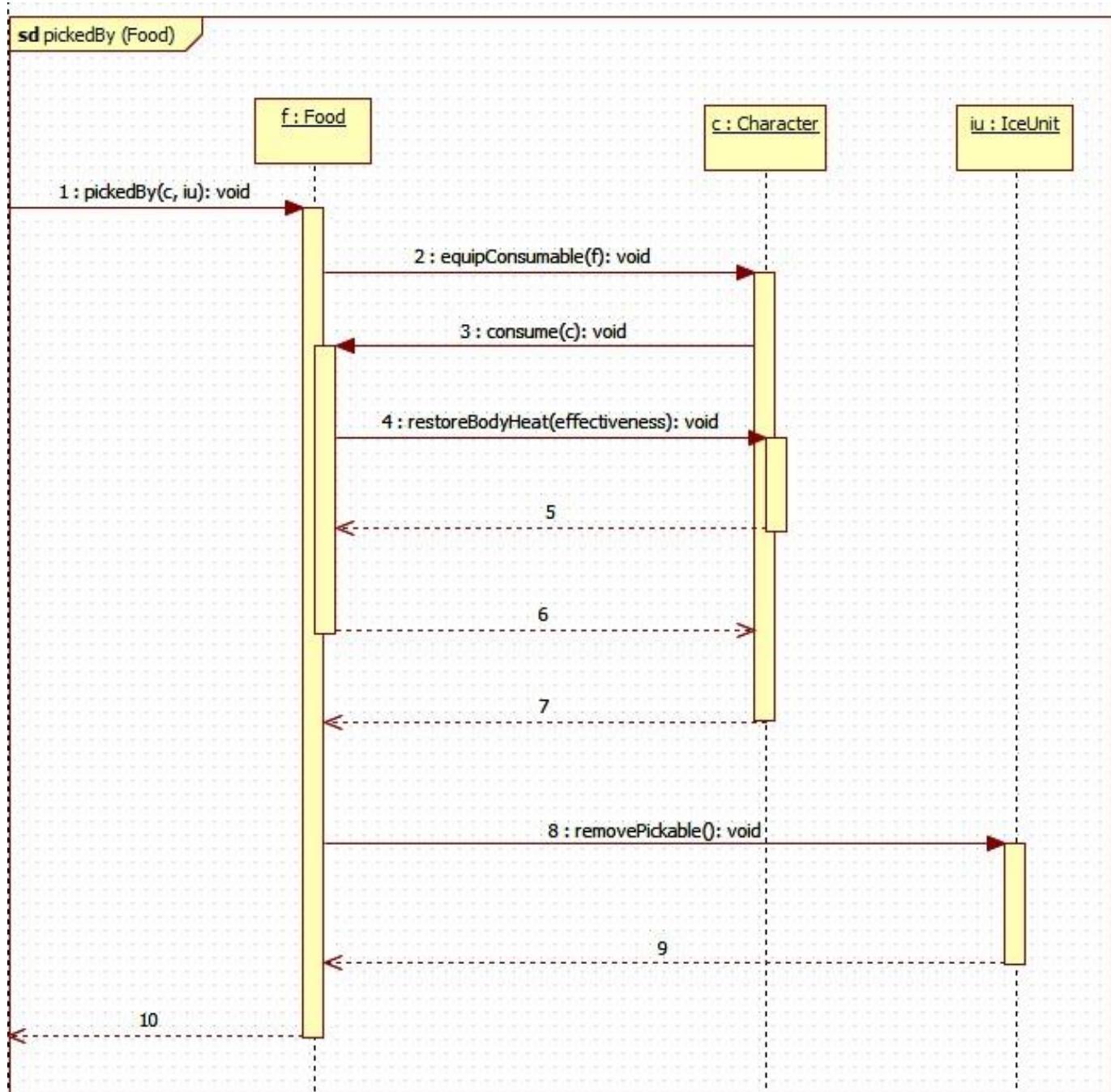
#### 4.4.4 Eskimo useAbility



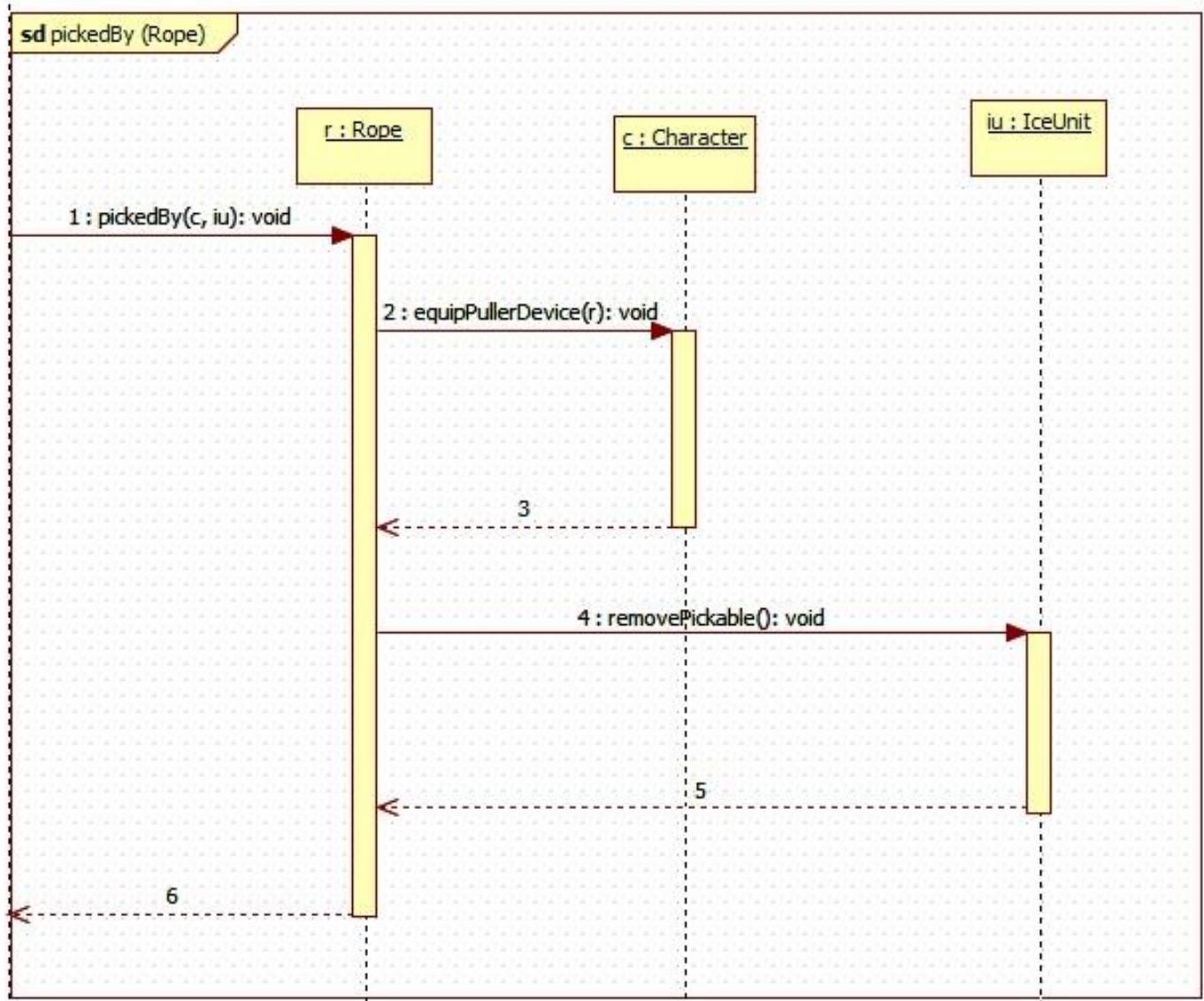
### 4.4.5 Dig



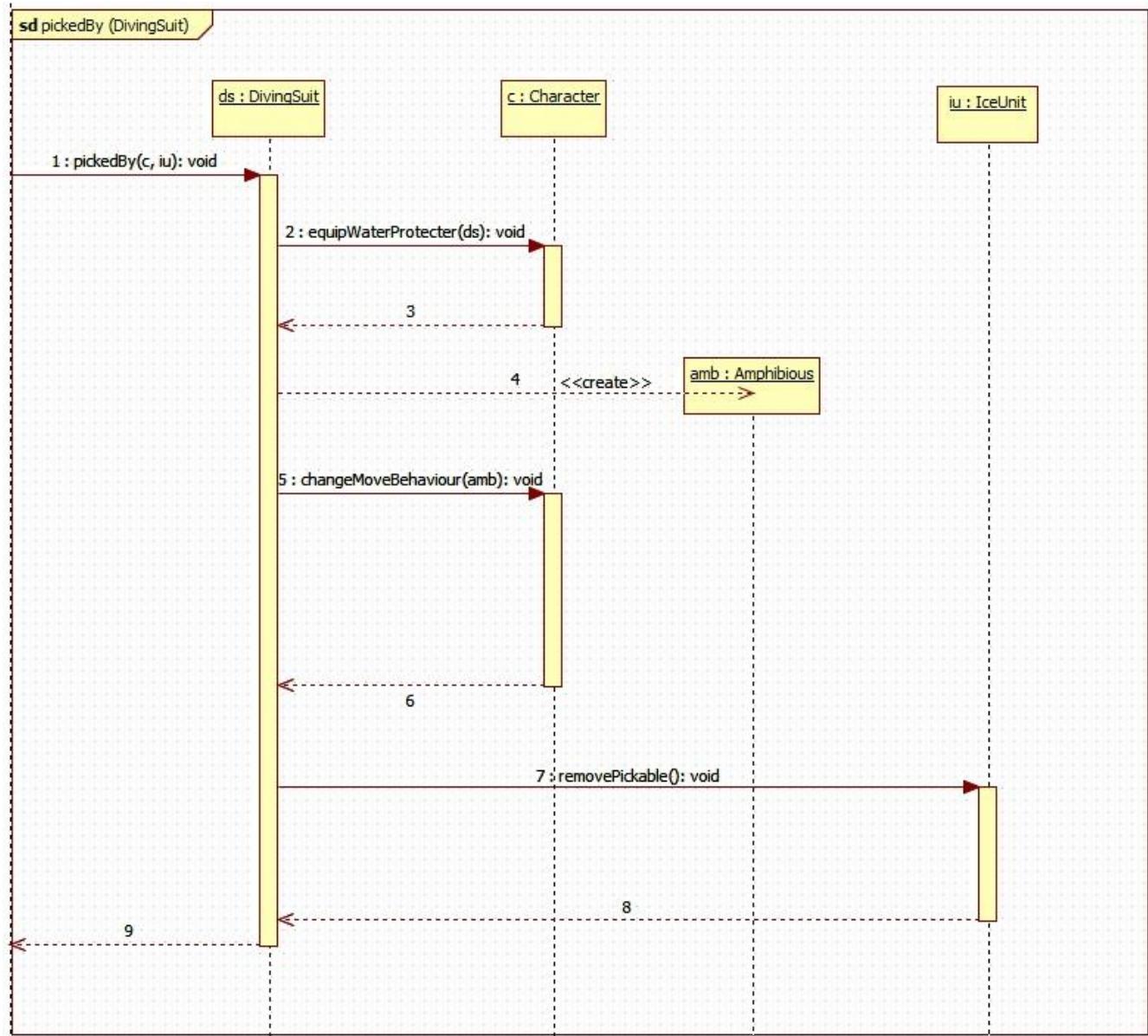
#### 4.4.6 pickedBy (Food)



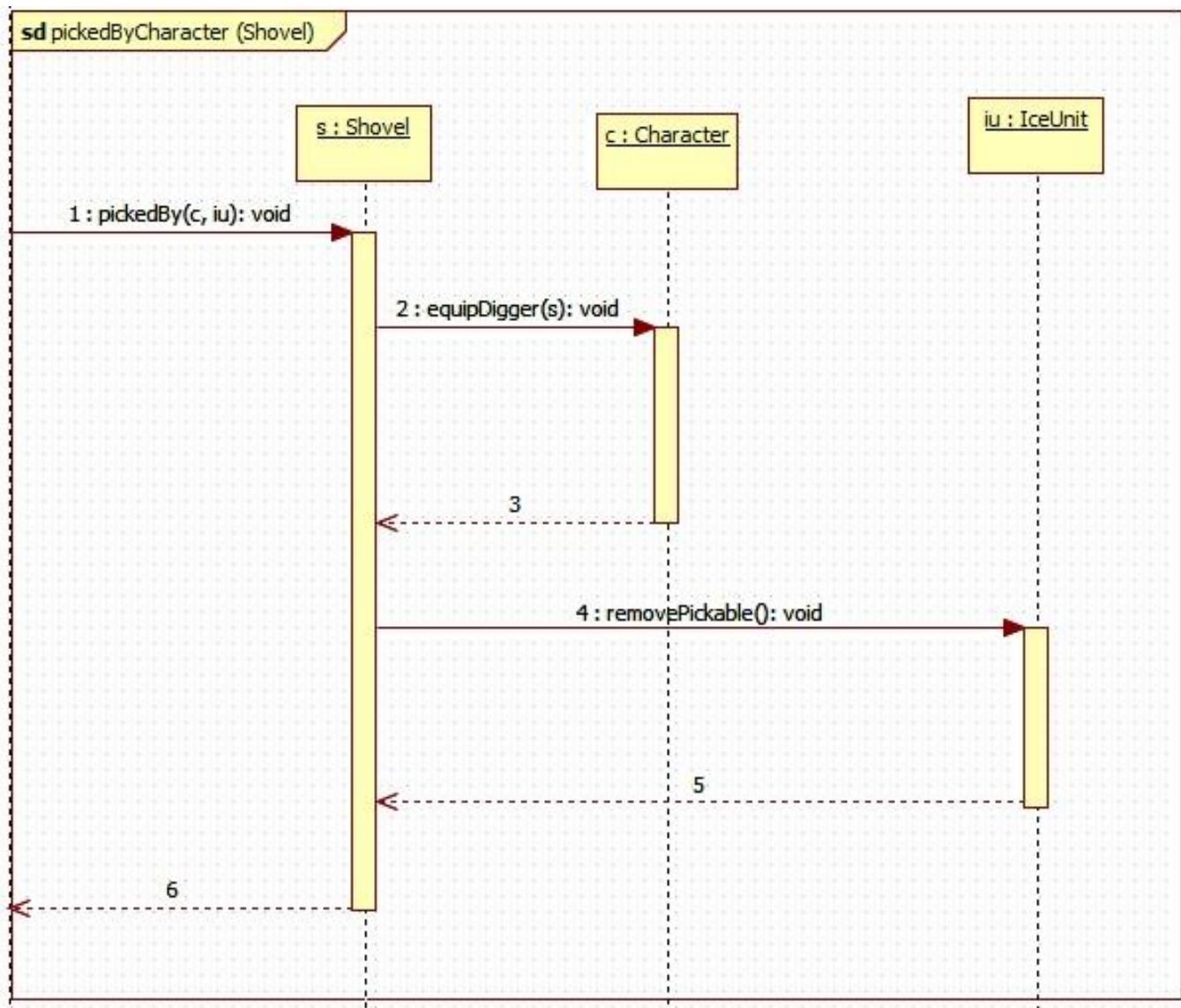
#### 4.4.7 pickedBy (Rope)



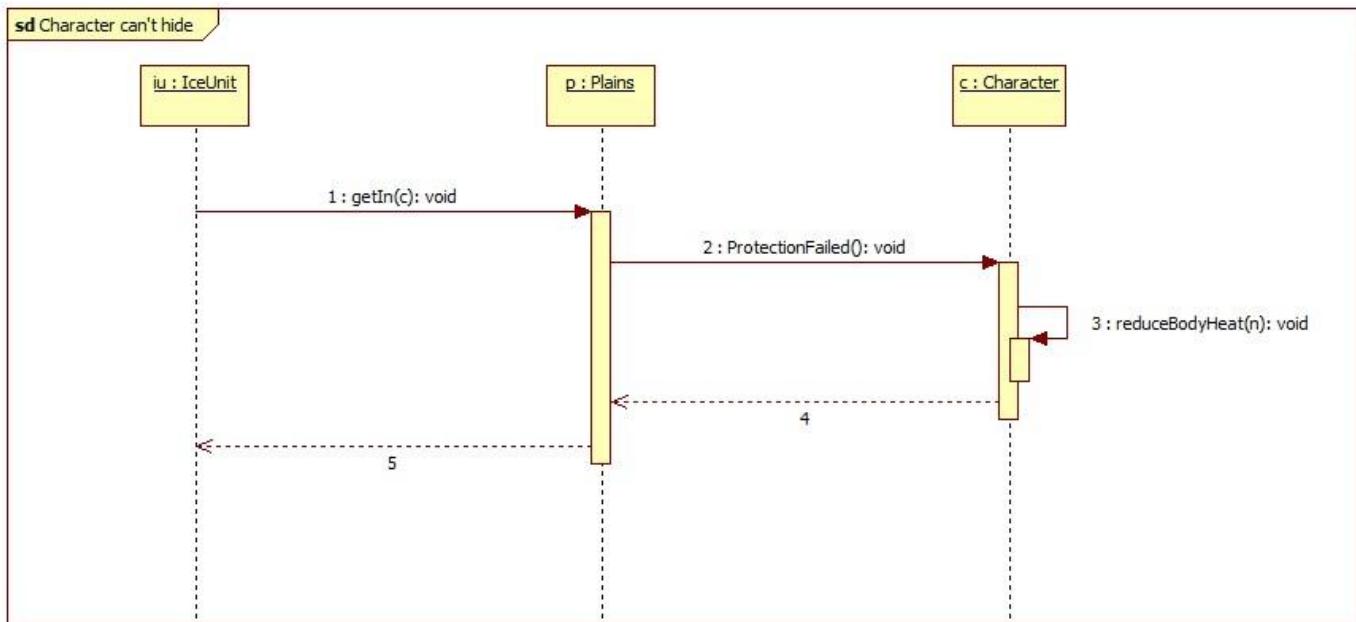
#### 4.4.8 pickedBy (DivingSuit)



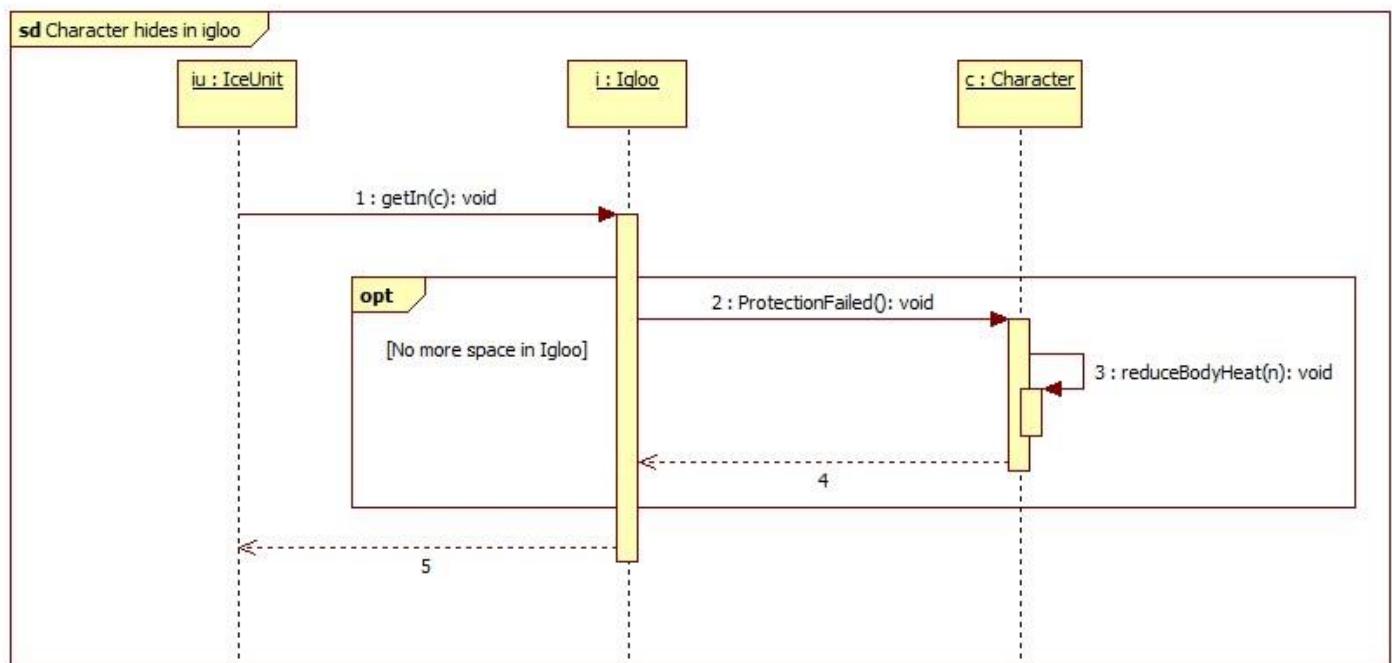
#### 4.4.9 pickedBy (Shovel)



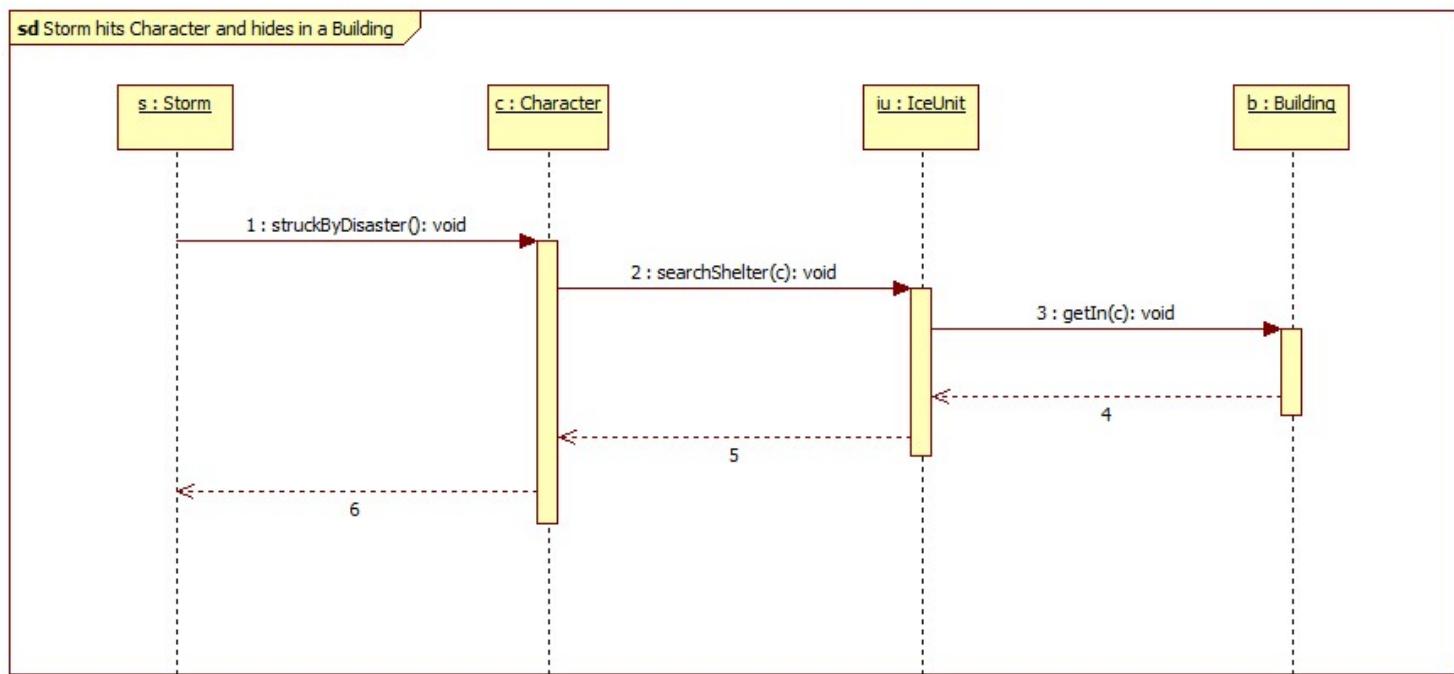
#### 4.4.10 Character can't hide



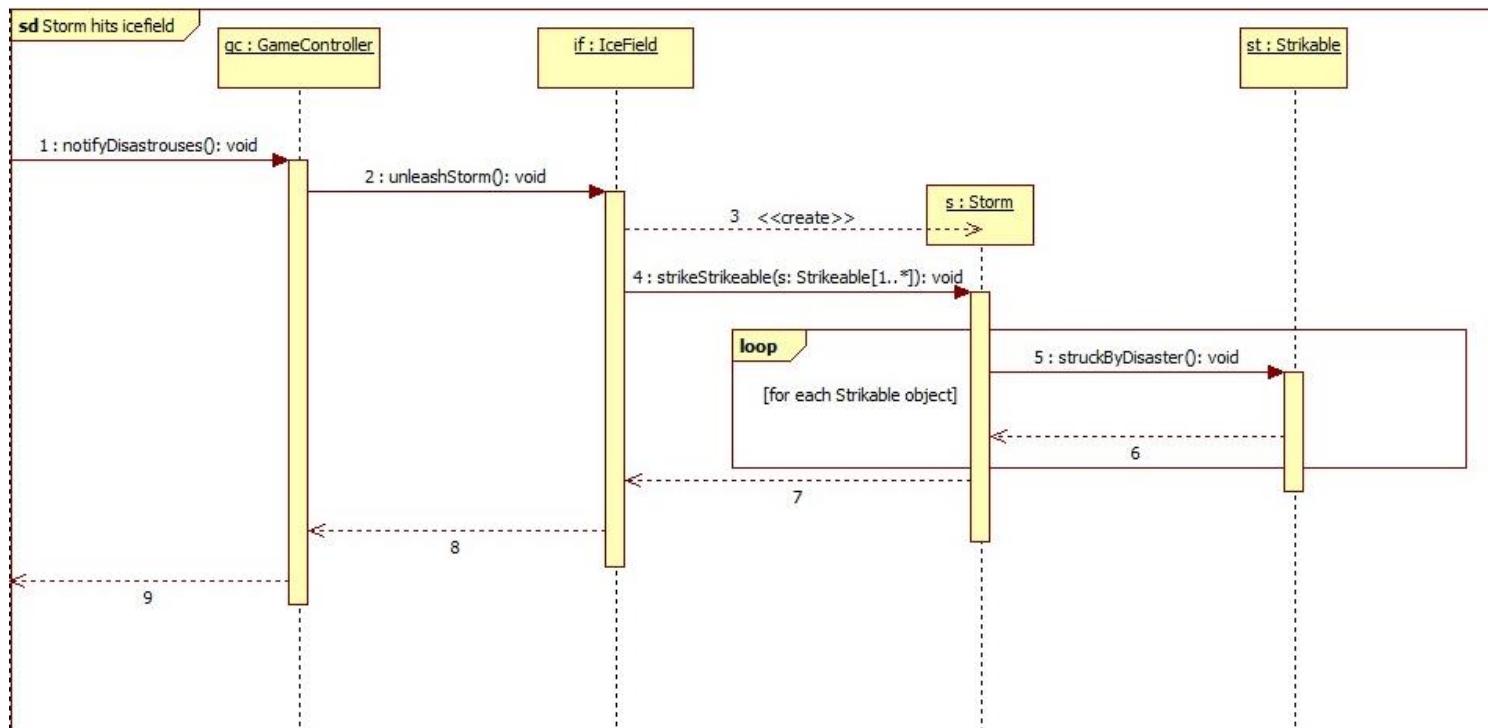
#### 4.4.11 Character hides in igloo



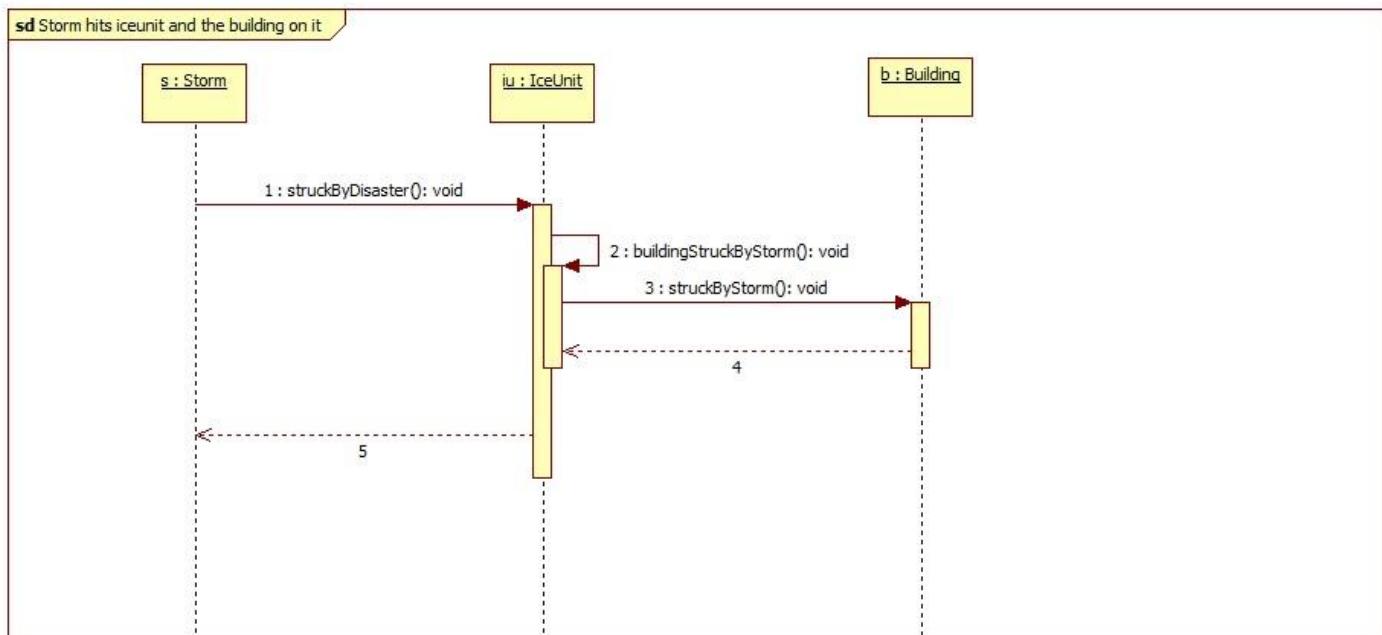
#### 4.4.12 Storm hits Character and hides in a Building



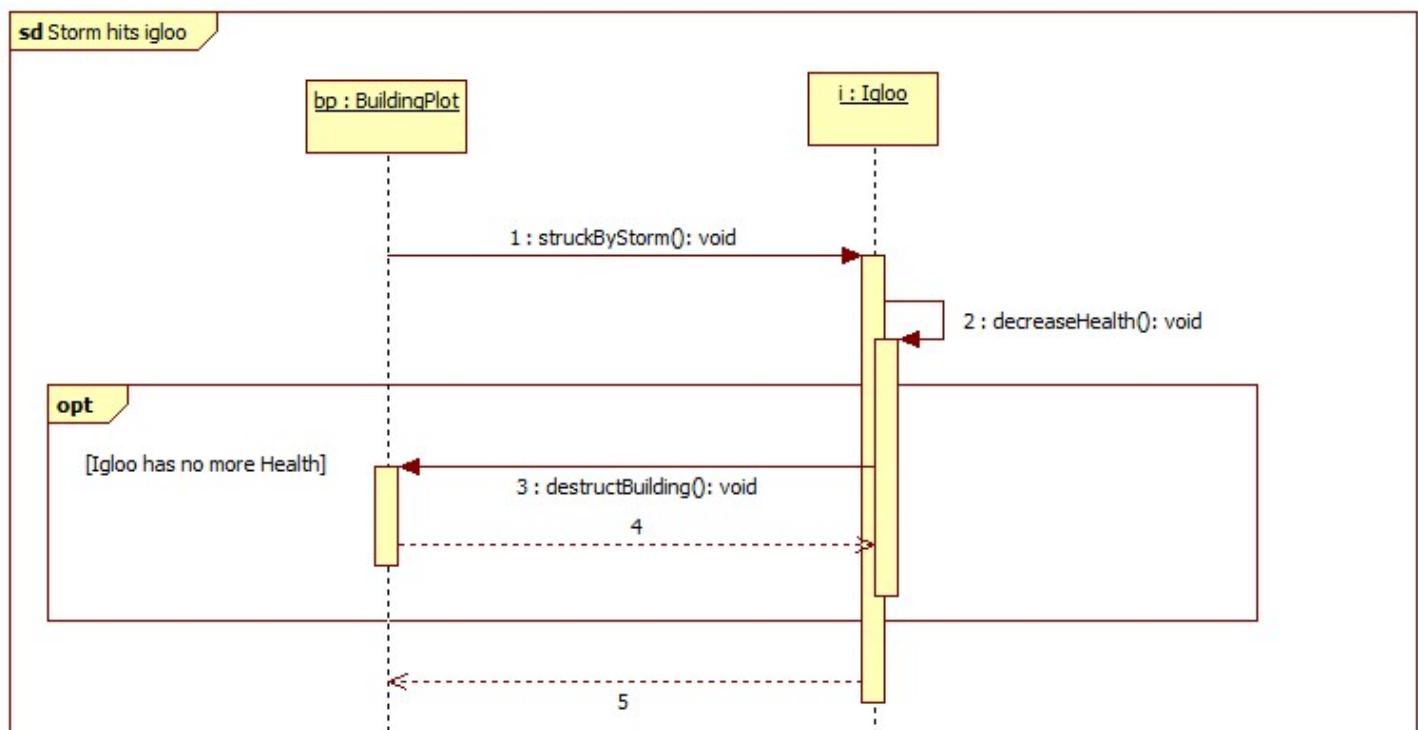
#### 4.4.13 Strom hits icefield



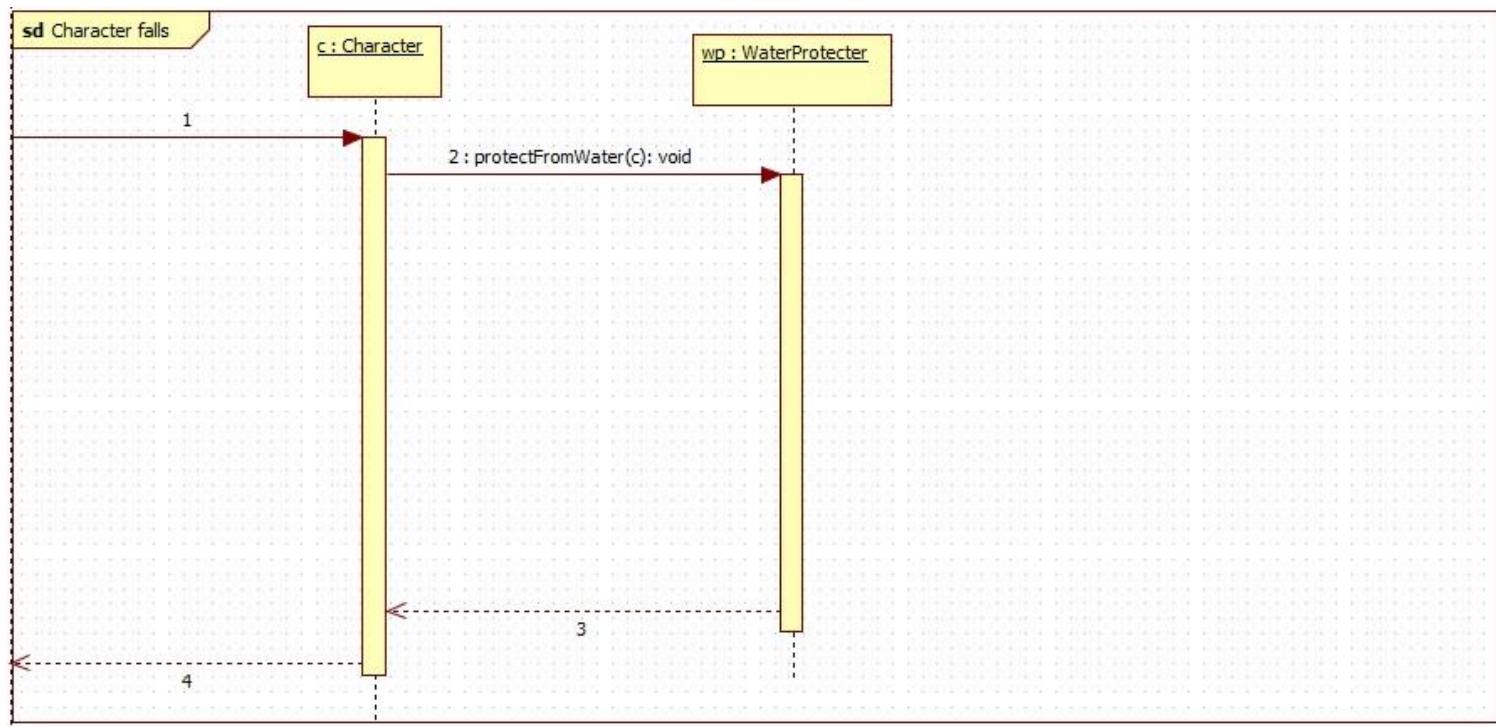
#### 4.4.14 Strom hits iceunit and the building on it



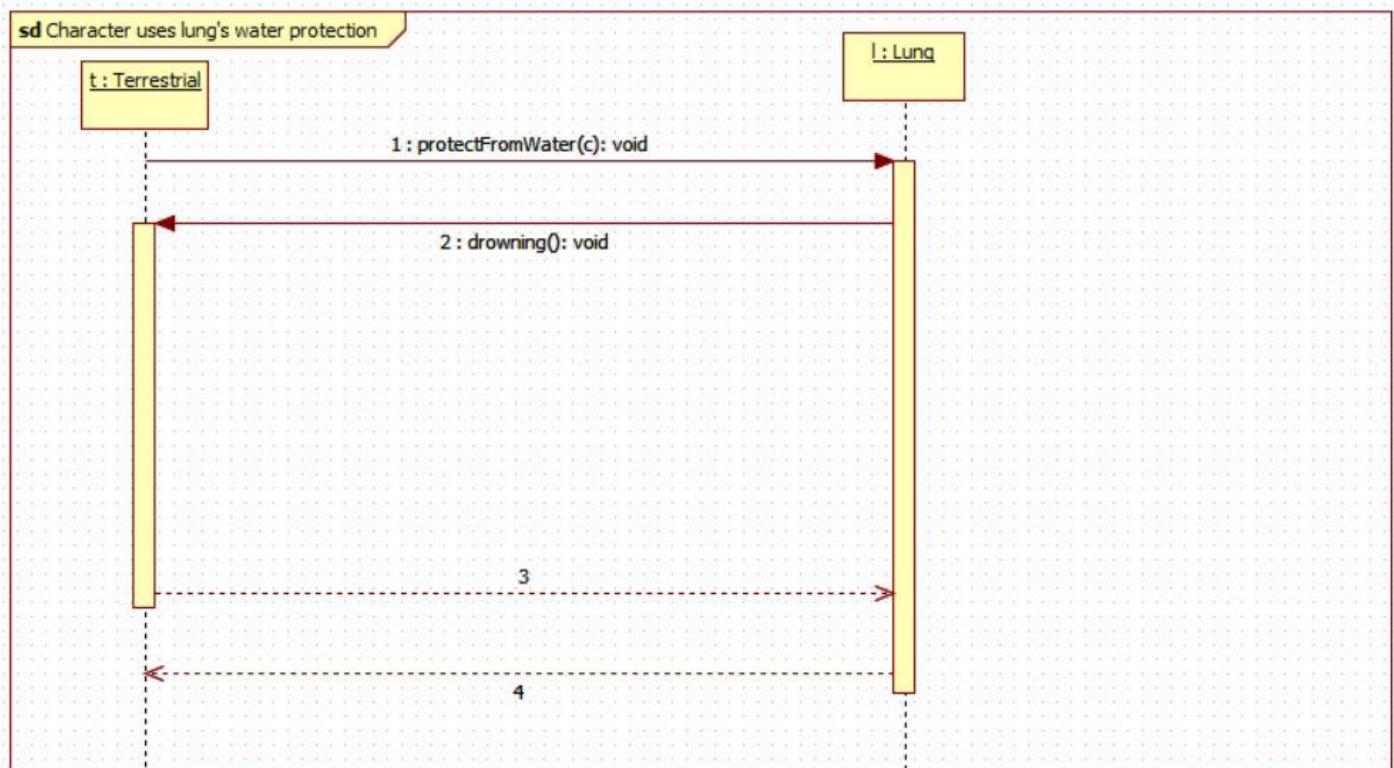
#### 4.4.15 Storm hits igloo



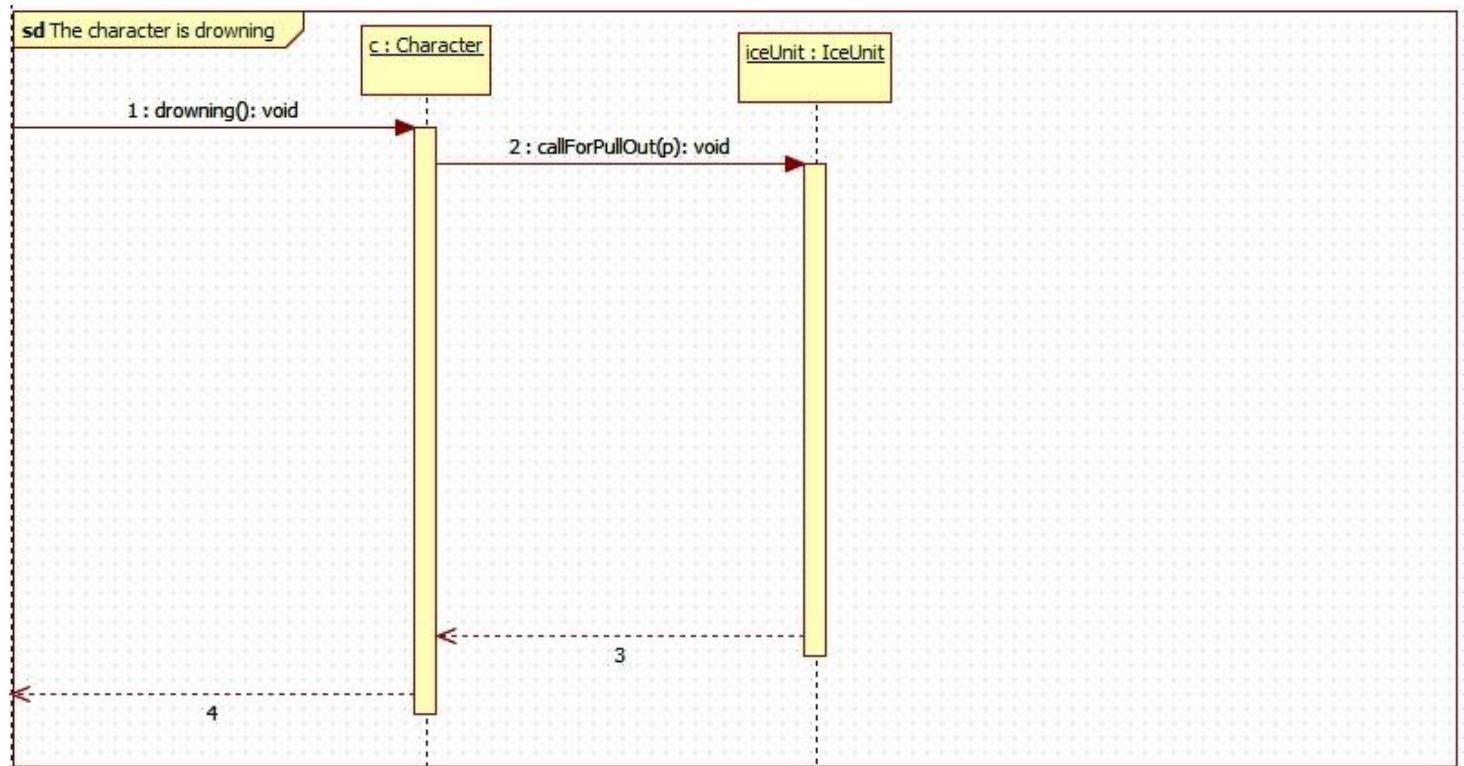
#### 4.4.16 Character falls



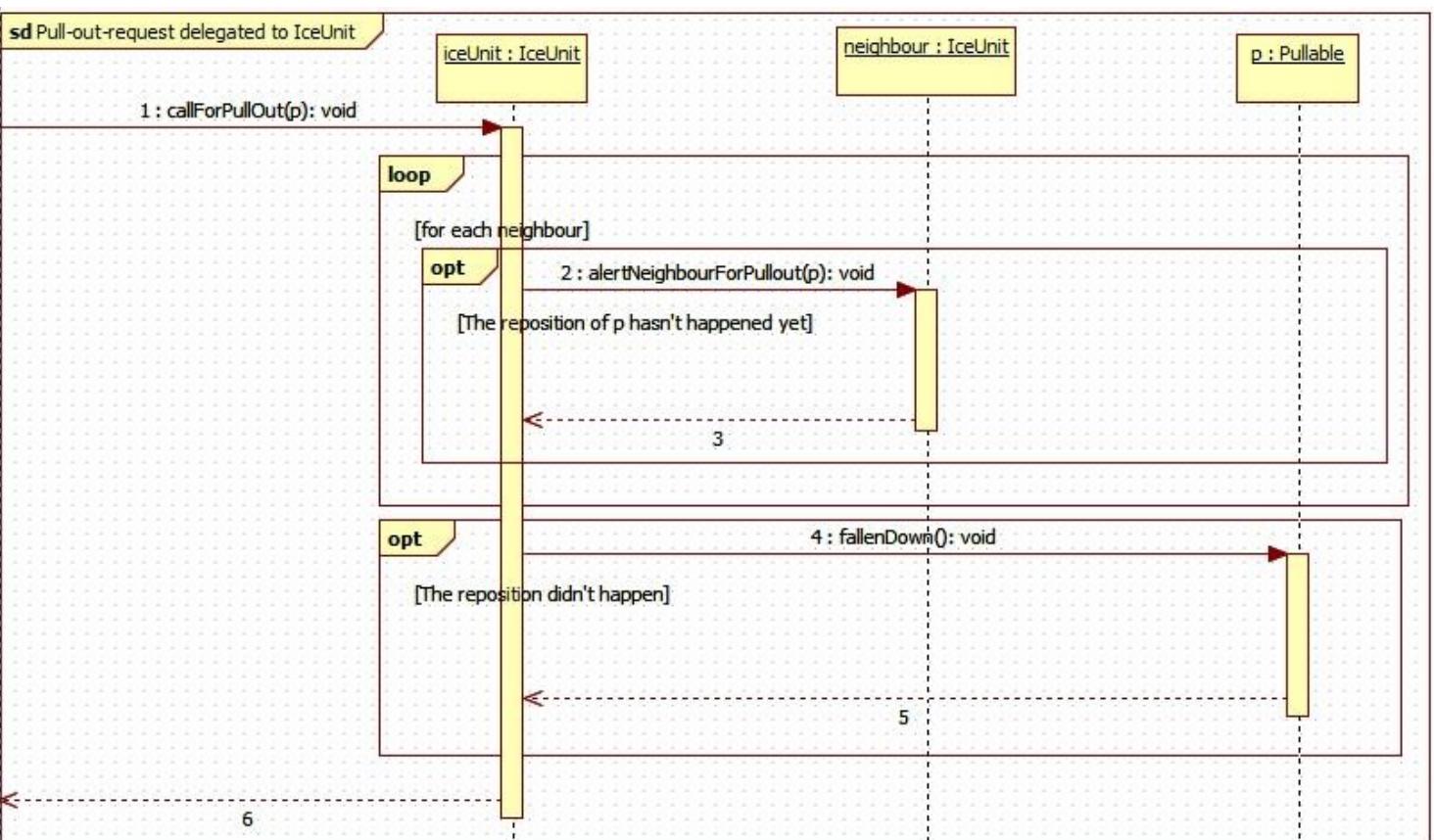
#### 4.4.17 Character uses lung's water protection



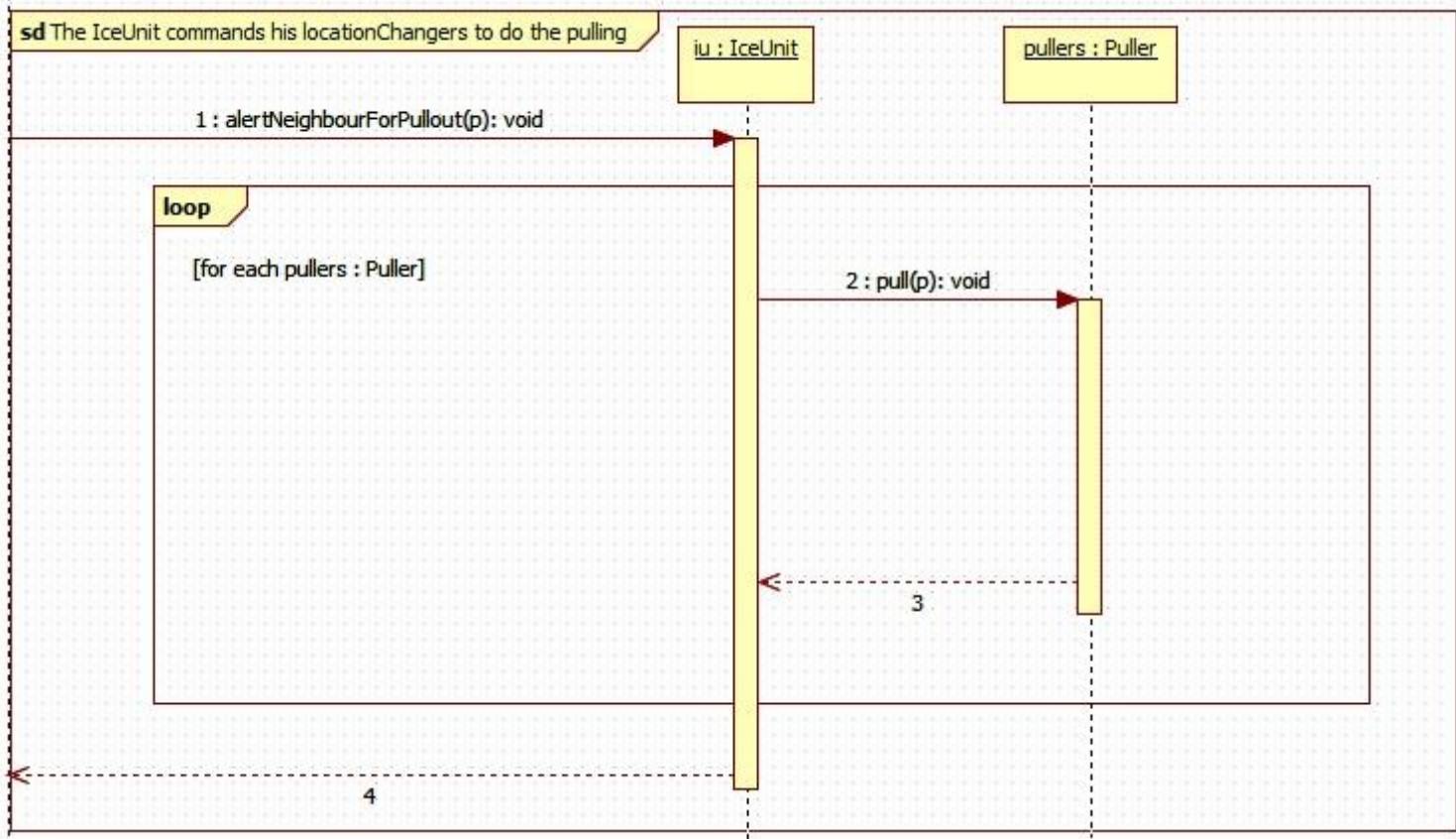
#### 4.4.18 Character drowning



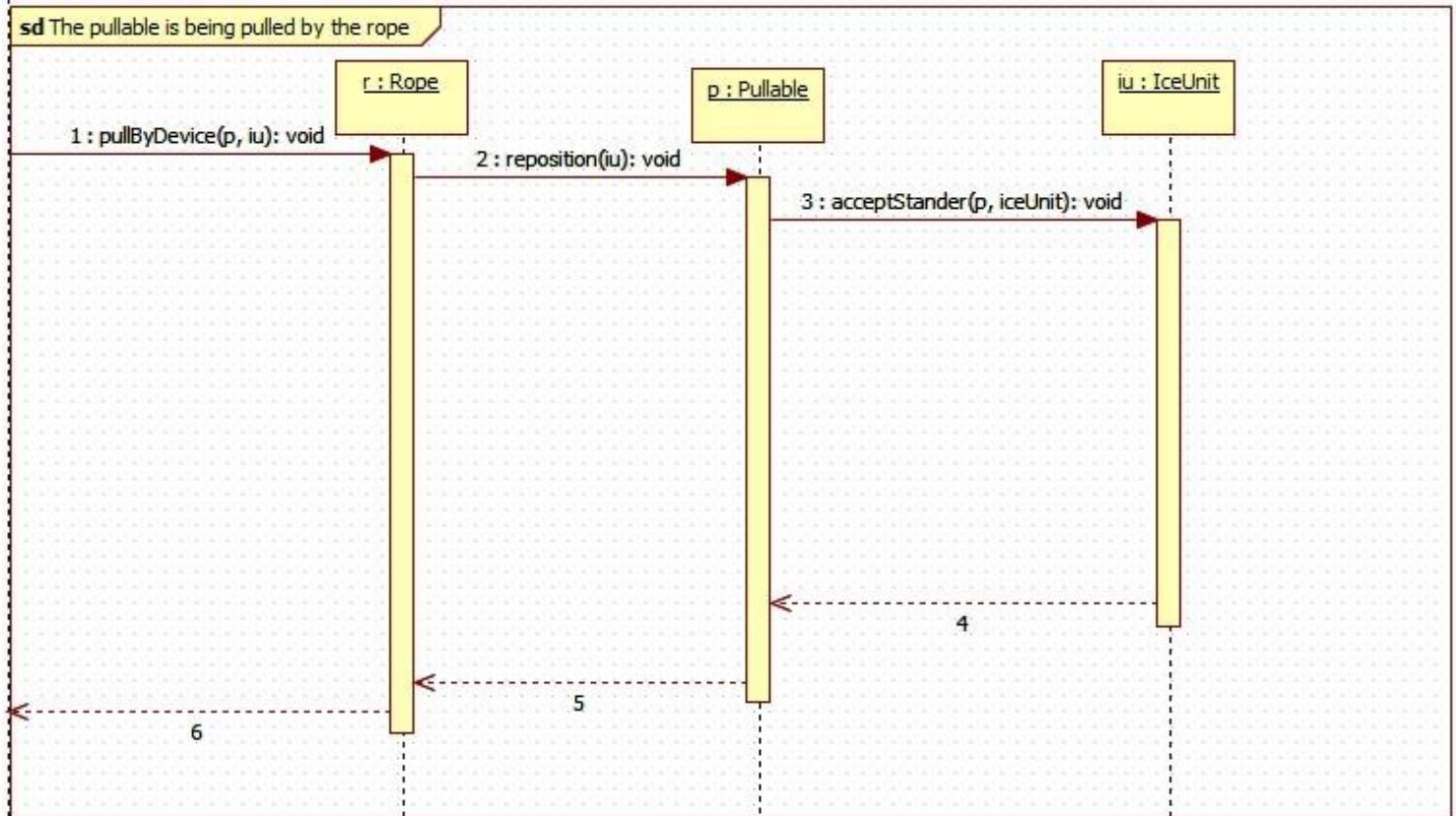
#### 4.4.19 Pull out request delegated to IceUnit

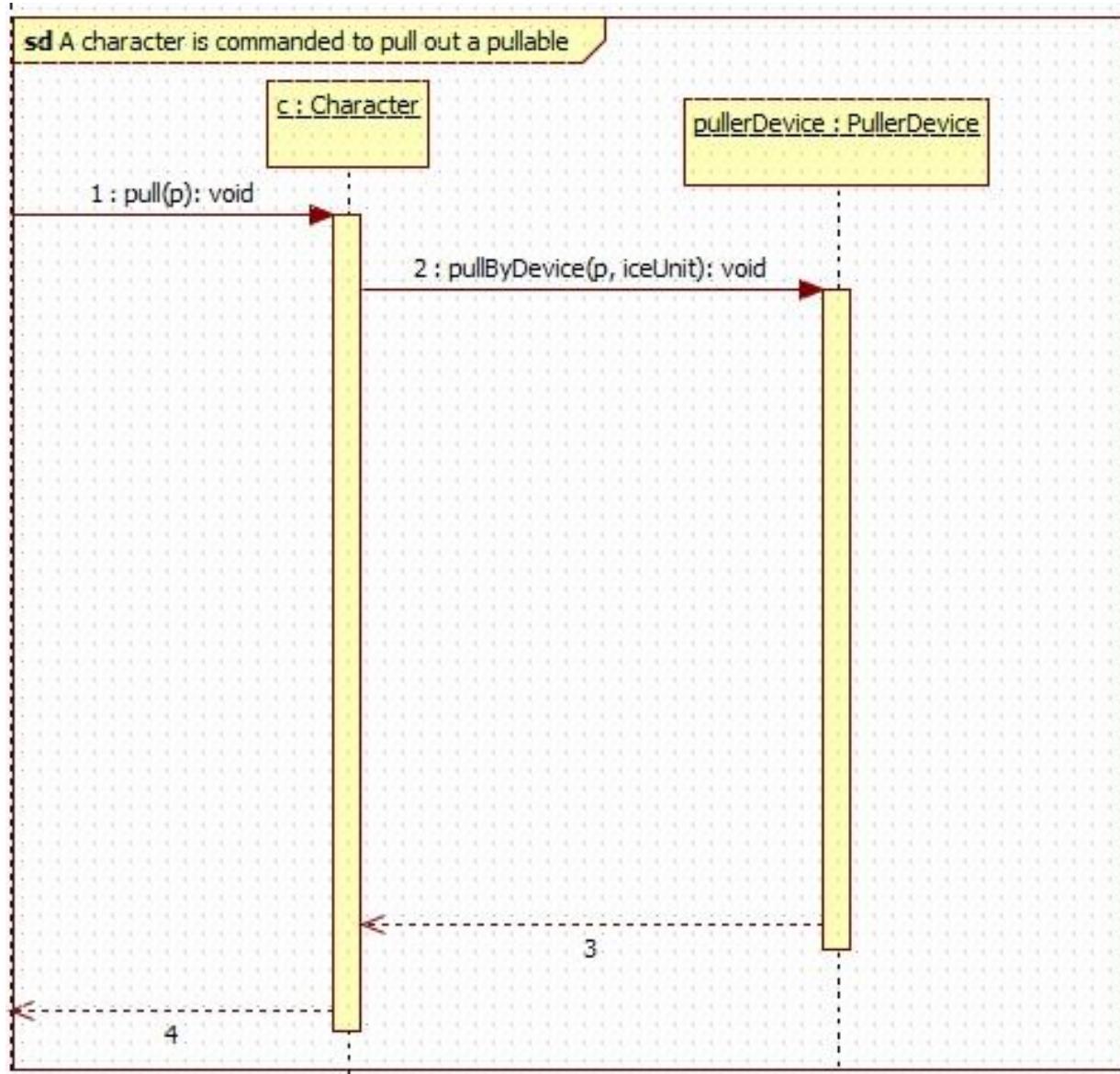


#### 4.4.20 The IceUnit commands his locationChangers to do the pulling

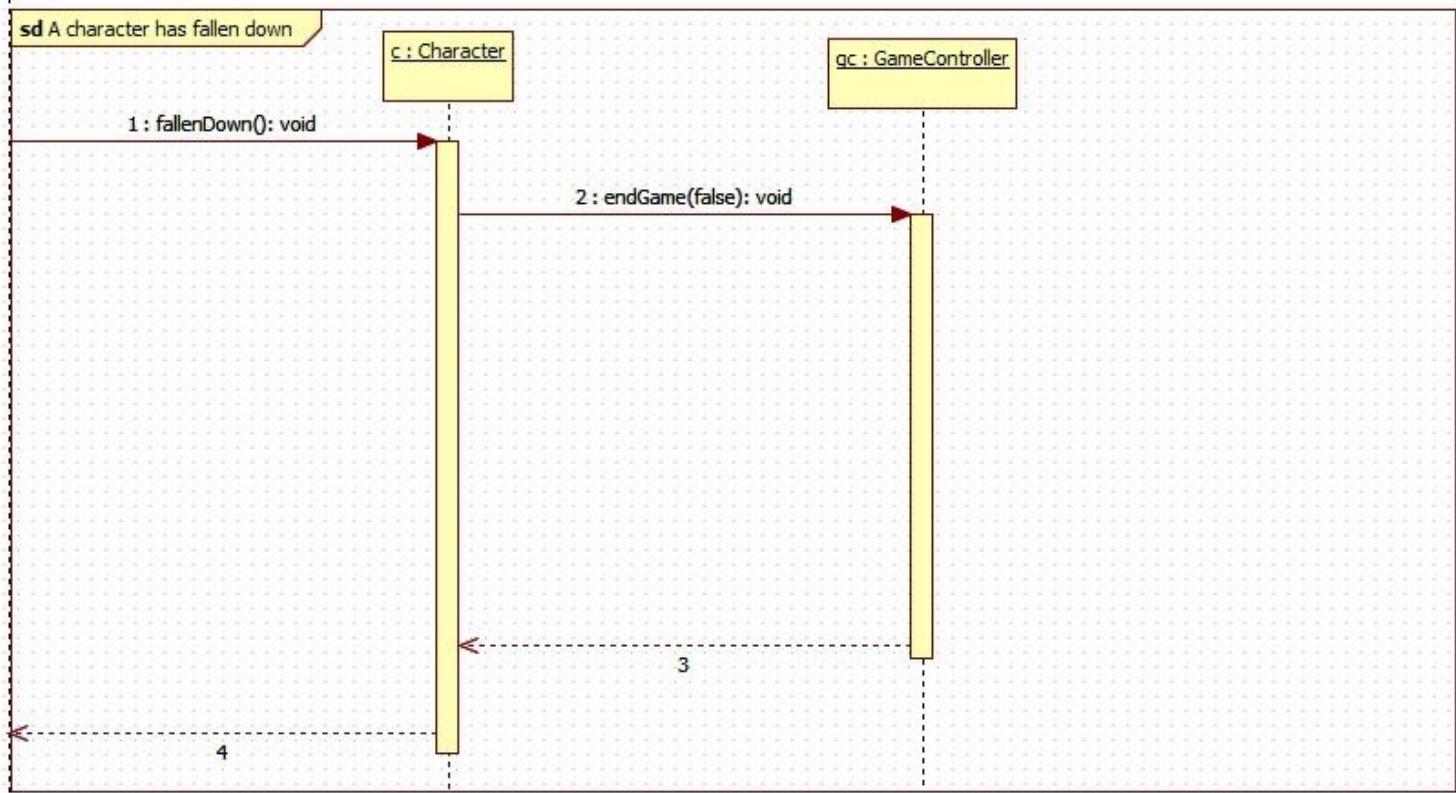


#### 4.4.21 The pullable is being pulled by the rope

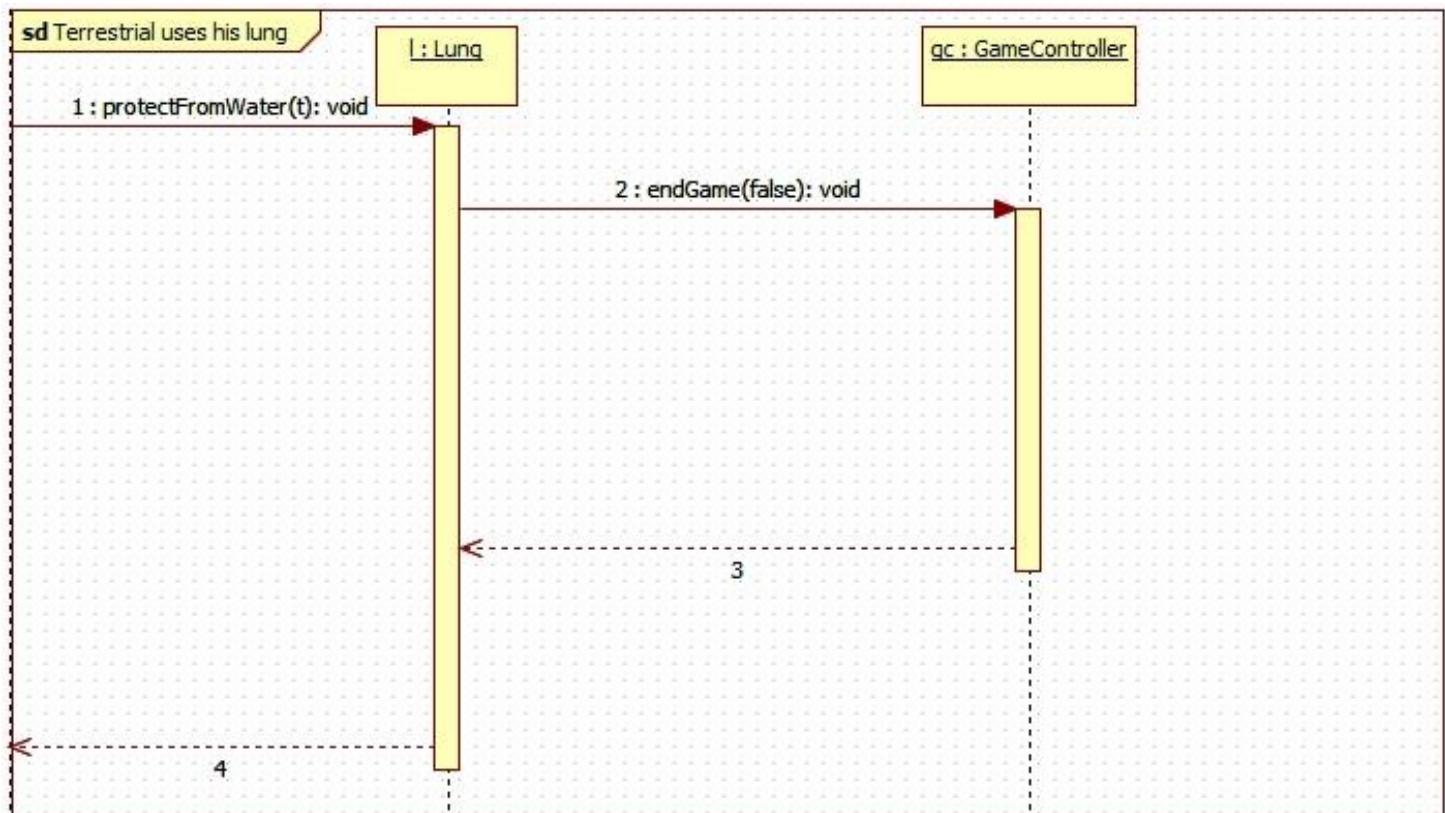


**4.4.22 A character is commanded to pull out a pullable**

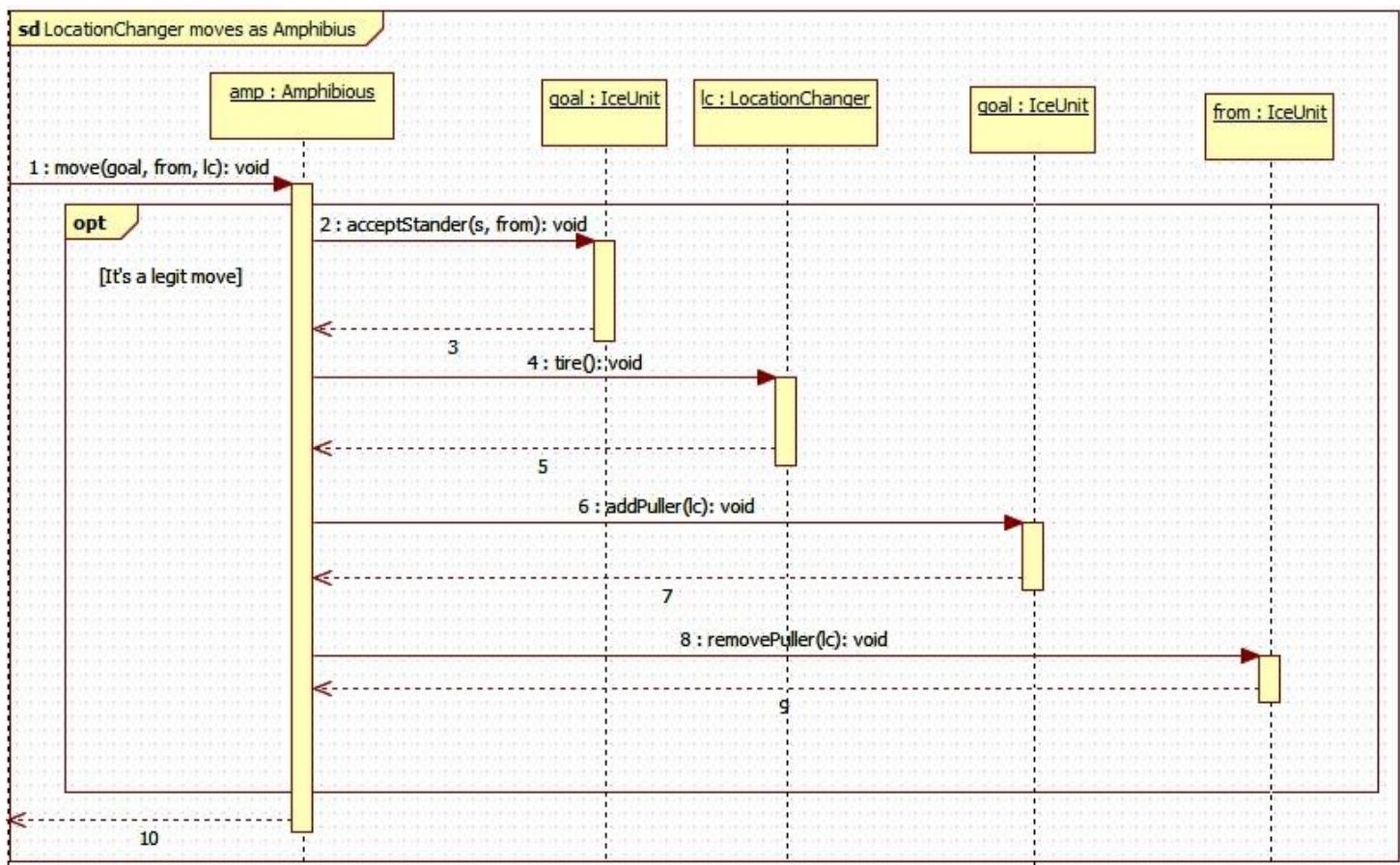
#### 4.4.23 A character has fallen down



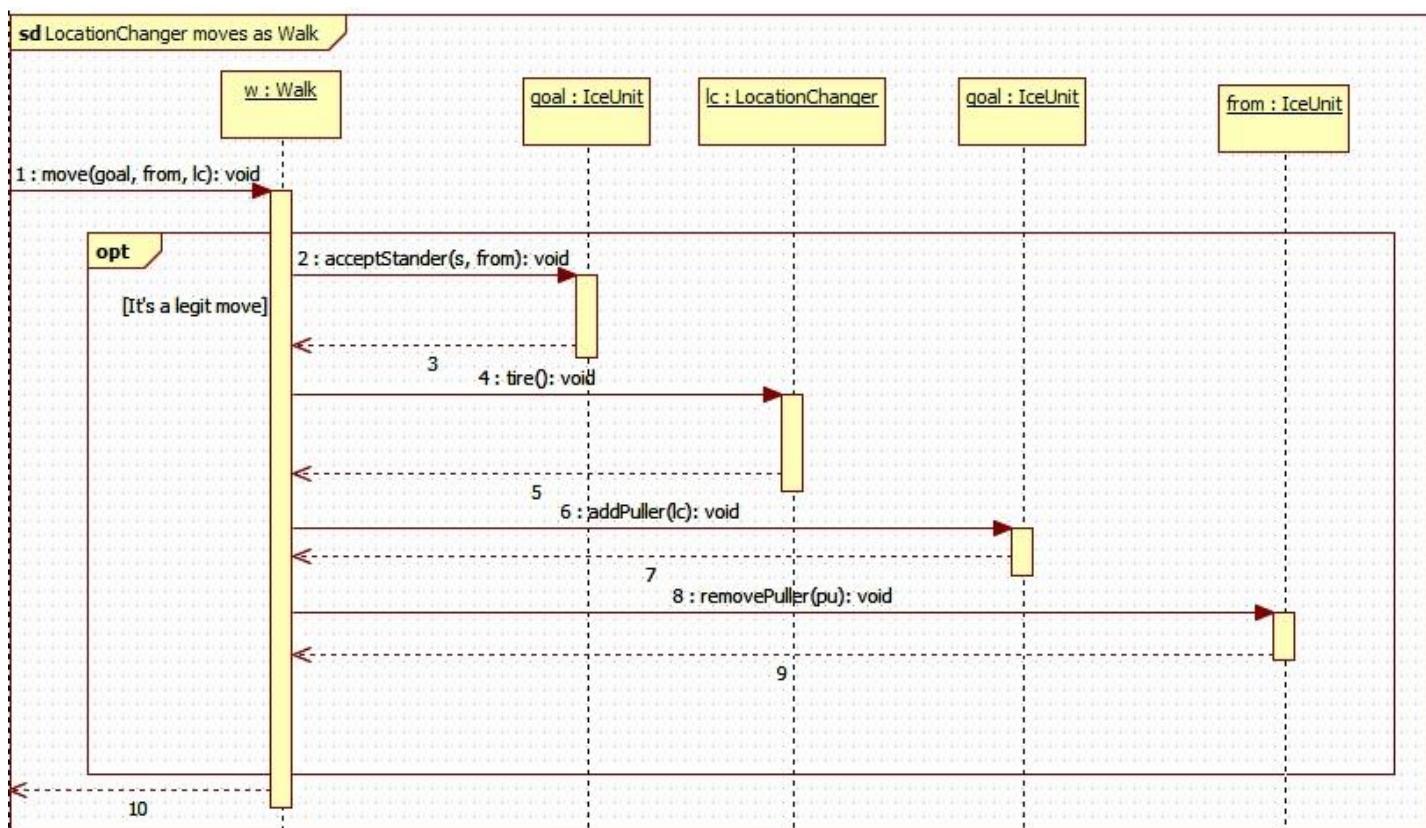
#### 4.4.24 Terrestrial uses his lung



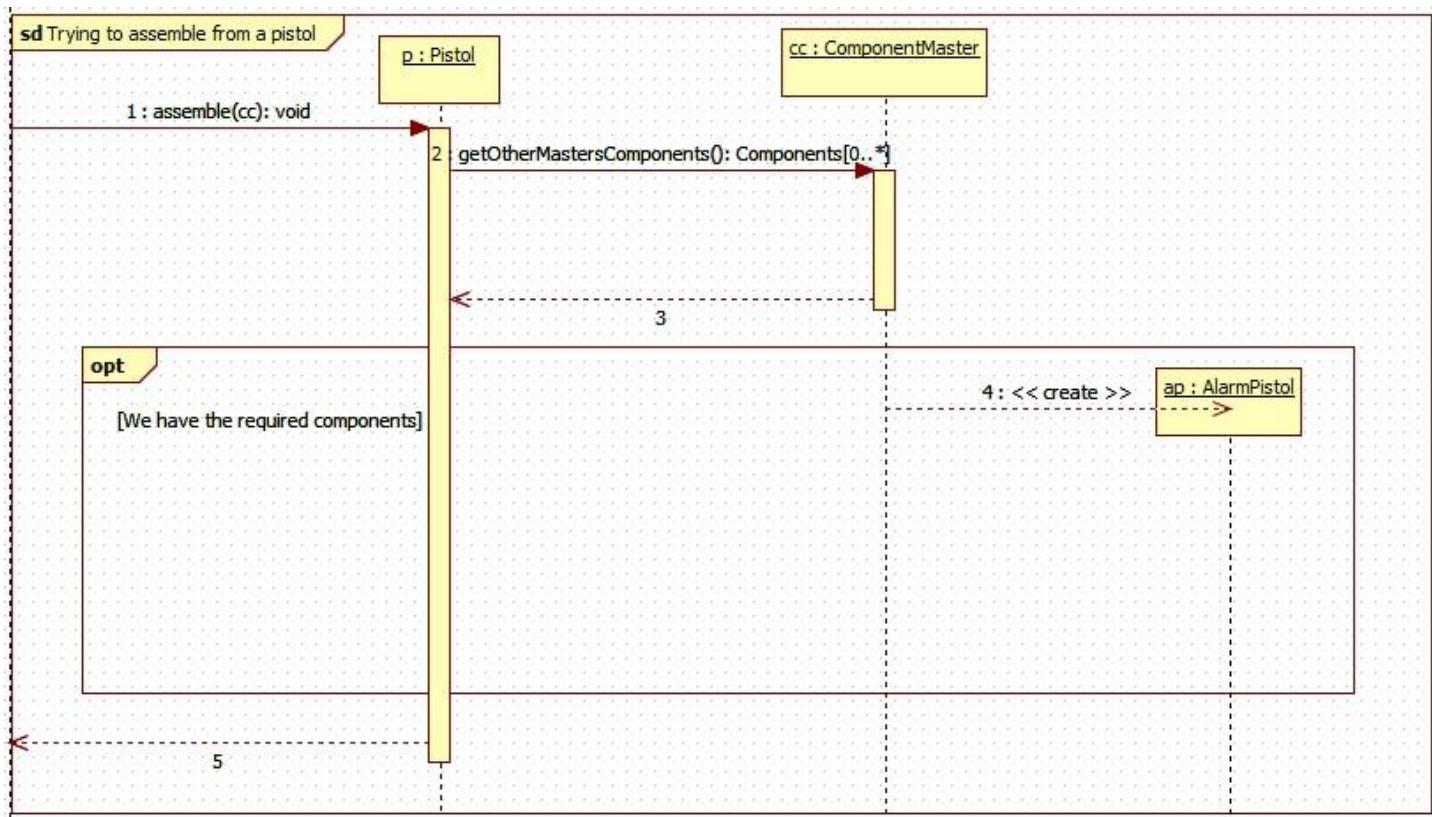
#### 4.4.25 LocationChanger moves as Amphibius



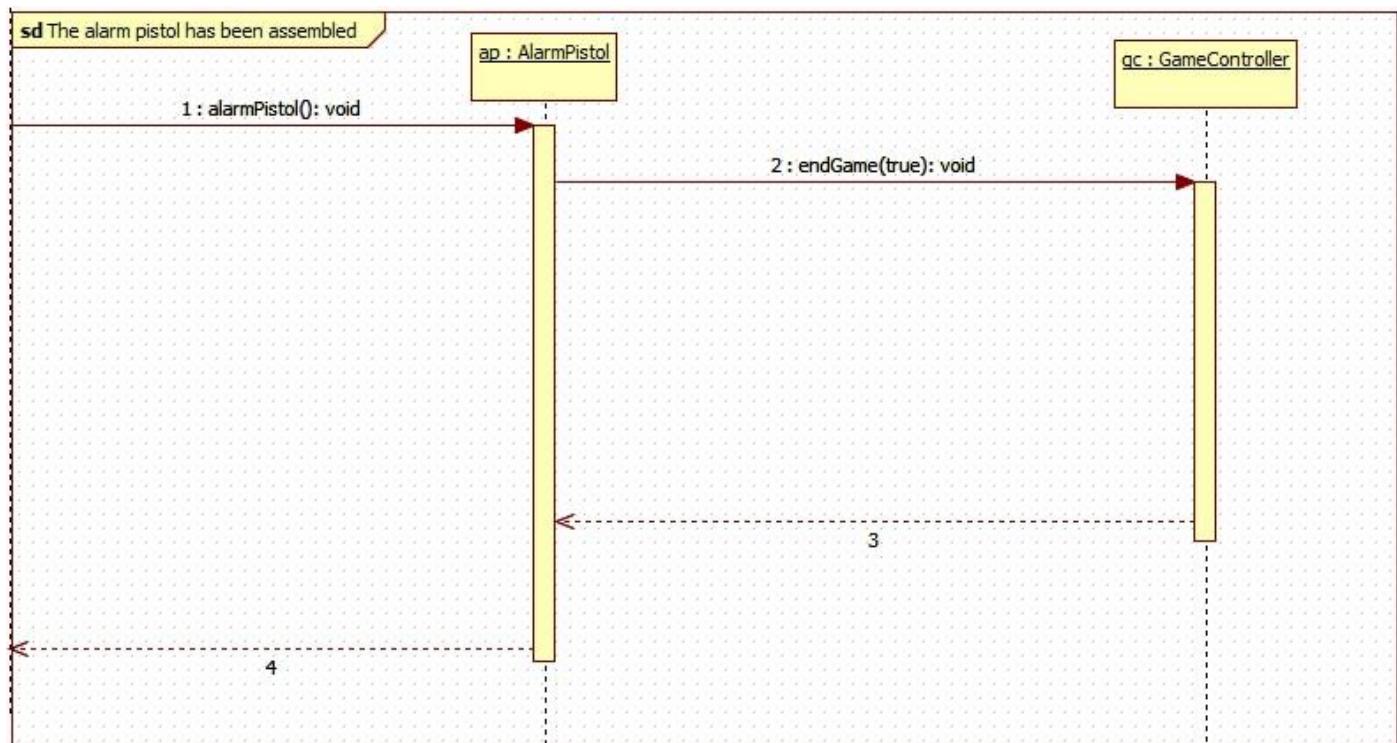
#### 4.4.26 LocationChanger moves as Walk



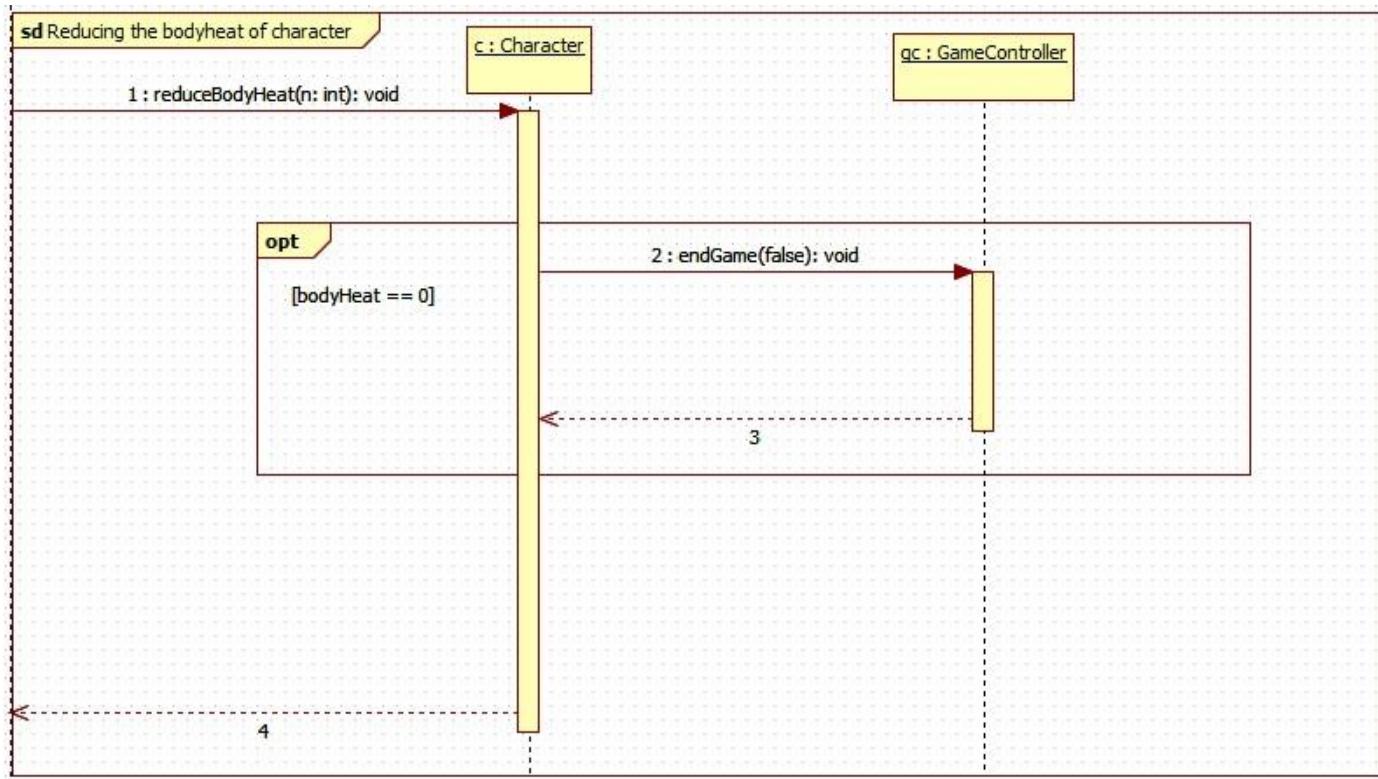
#### 4.4.27 Trying to assemble from a pistol



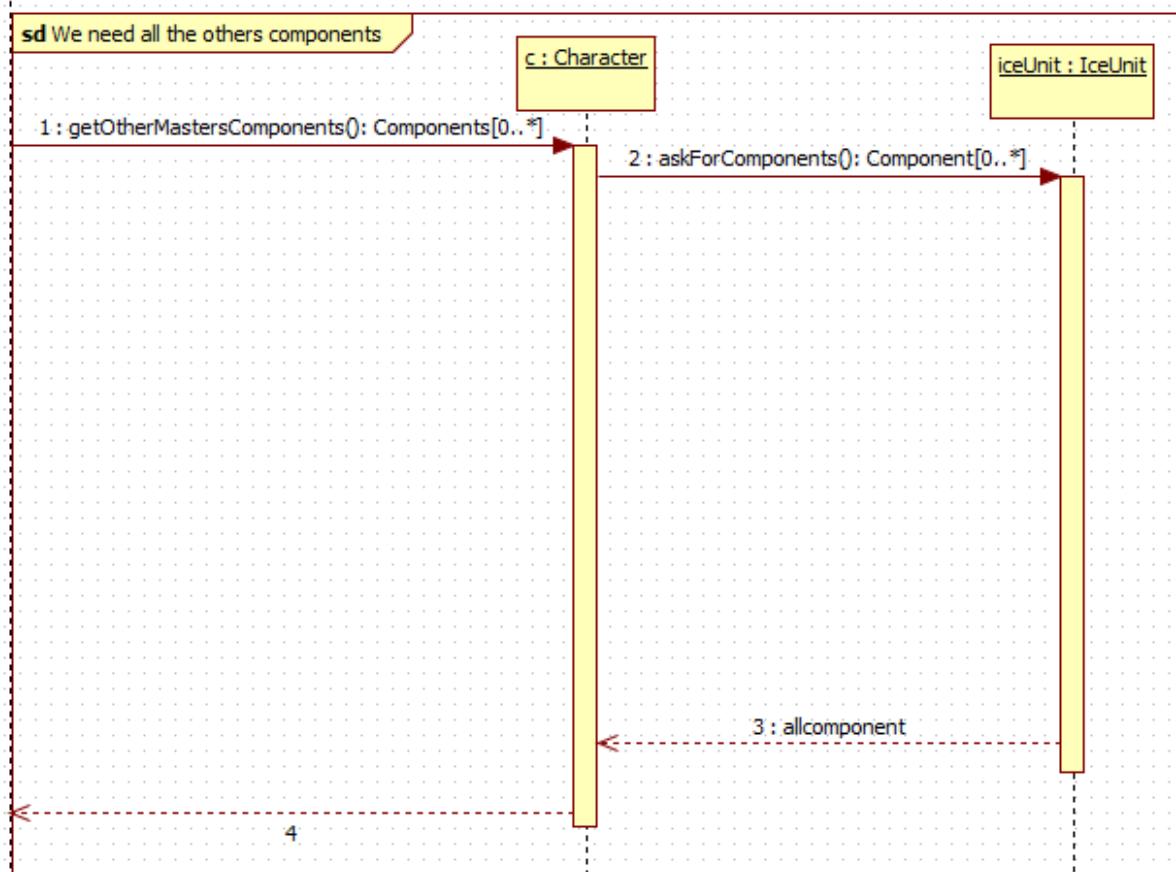
#### 4.4.28 The alarm pistol has been assembled



#### 4.4.29 Reducing the bodyheat of a character



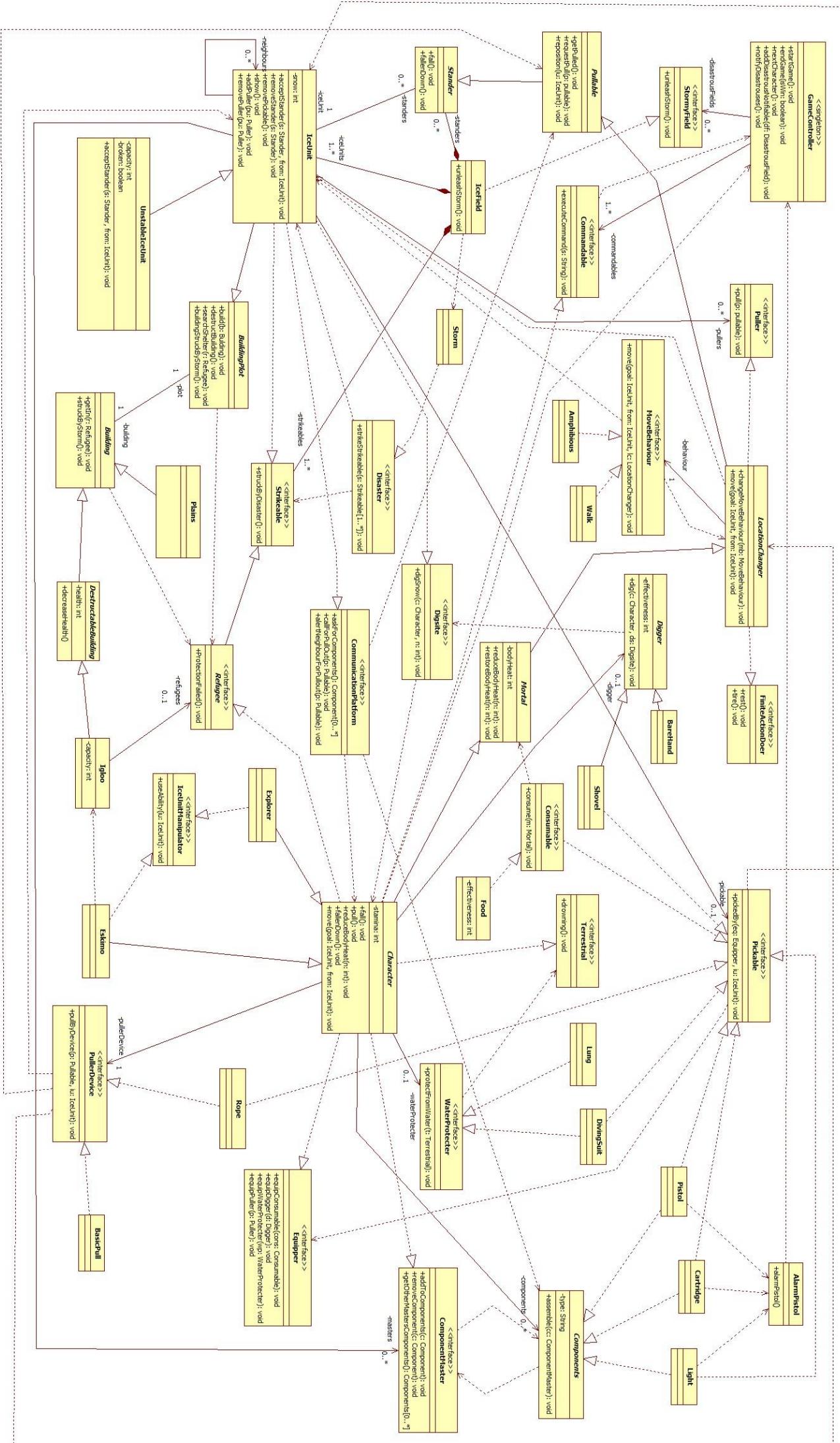
#### 4.4.30 We need all the others components



## 4.5 Napló

Kezdet	Időtartam	Résznevők	Leírás
2020.03.04. 17:30-21:30	4 óra	Iuhos Kaposvári Labancz Sulyok Traxler	Értekezlet: szoftver struktúrájának újragondolása
2020.03.05. 16:00-20:00	4 óra	Labancz	Tevékenység: Osztálydiagram újratervezése
2020.03.05. 20:00-22:00	2 óra	Sulyok	Tevékenység: Szekvencia diagramok készítése
2020.03.06 12:00-14:00	2 óra	Iuhos Kaposvári Traxler	Értekezlet: Feladatok kiosztása
2020.03.06 14:00-18:00	4 óra	Traxler	Tevékenység: Osztálydiagram bővítése
2020.03.06 15:00-18:00	3 óra	Kaposvári	Tevékenység: Szekvencia diagramok készítése
2020.03.06 18:00-20:00	2 óra	Iuhos	Tevékenység: Osztálydiagram javítása
2020.03.07 10:00-14:00	4 óra	Sulyok	Tevékenység: Szekvencia diagramok készítése
2020.03.07 14:00-15:00	1 óra	Iuhos Kaposvári Labancz	Értekezlet: Eddigi munka értékelése, további feladatok kiosztása
2020.03.07 15:00-18:00	3 óra	Kaposvári	Tevékenység: Szekvencia diagramok javítása
2020.03.07 15:30-18:30	3 óra	Traxler	Tevékenység: Osztálydiagram javítása
2020.03.07 18:00-20:00	2 óra	Iuhos	Tevékenység: Dokumentáció készítése
2020.03.07 19:00-21:00	2 óra	Labancz	Tevékenység: Szekvencia diagramok készítése
2020.03.08 10:00-12:00	2 óra	Iuhos Kaposvári Sulyok	Értekezlet:

			Szekvencia diagramok ellenőrzése
2020.03.08 12:00-14:00	2 óra	Labancz	Tevékenység: Osztálydiagram javítása
2020.03.08 12:00-14:00	2 óra	Traxler	Tevékenység: Szekvencia diagramok bővítése
2020.03.08 14:00-17:00	3 óra	Sulyok	Tevékenység: Szekvencia diagramok javítása
2020.03.08 17:00-19:00	2 óra	Labancz	Tevékenység: Dokumentáció készítése
2020.03.08 19:00-21:00	2 óra	Iuhos	Tevékenység: Dokumentáció javítása



# **5. Feladat: Szkeleton tervezése - beadás**

**14 – the\_minesweepers**

Konzulens:  
**Fetter László**

## **Csapattagok**

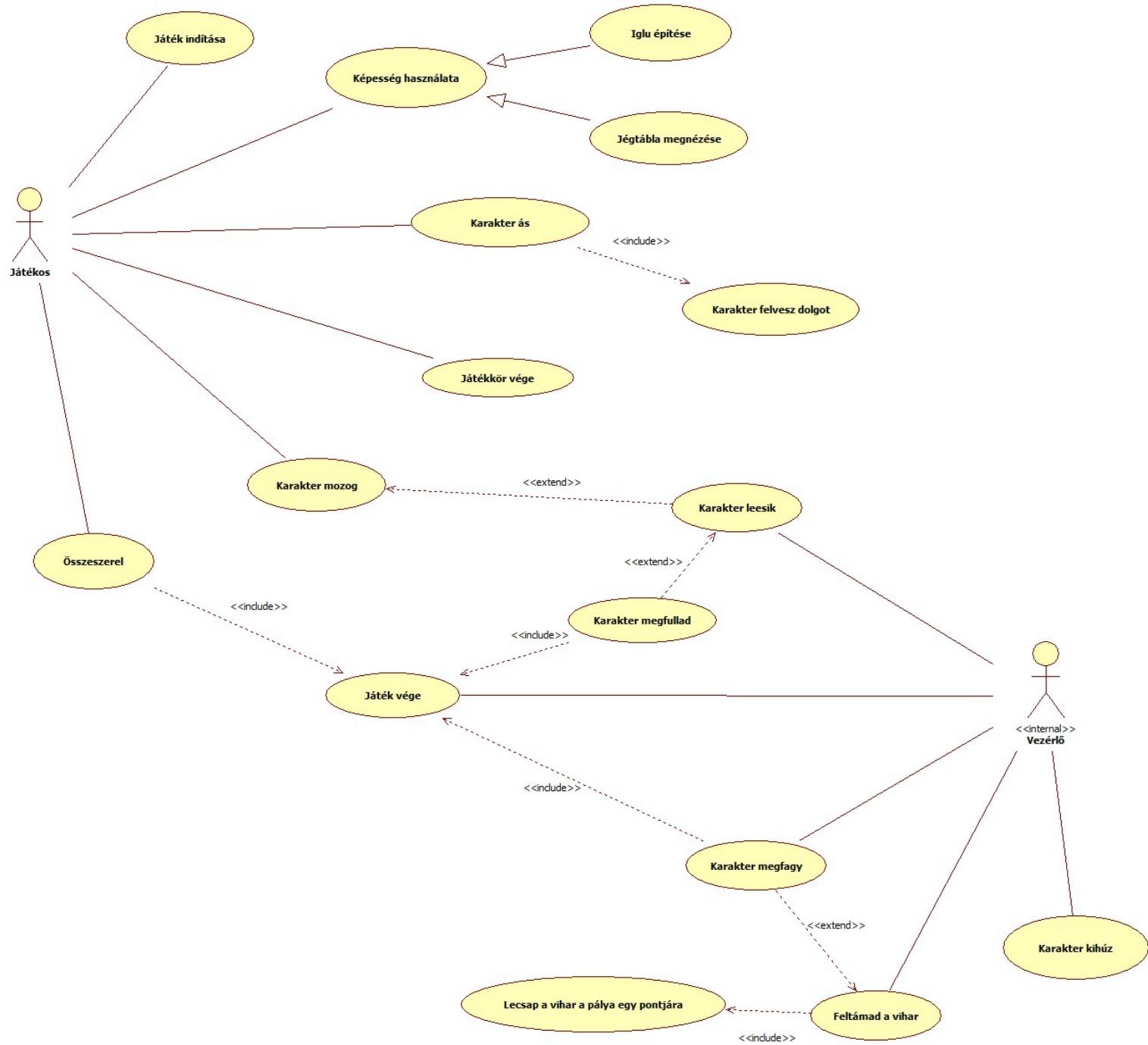
Iuhos Zoltán	J8LOUL	zoltan.iuhos@gmail.com
Kaposvári Márk Martin	RAYVF8	xmarko45@gmail.com
Sulyok Botond	BM2BBM	botond.sulyok99@gmail.com
Labancz Tamás Dominik	A9HZBF	labancz.tamas@gmail.com
Traxler Bálint	IXNOYB	traxlerbalint@gmail.com

2020.03.23.

## 5. Szkeleton tervezése

### 5.1 A szkeleton modell valóságos use-case-ei

#### 5.1.1 Use-case diagram



### 5.1.2 Use-case leírások

<b>Use-case neve</b>	Játék indítása
<b>Rövid leírás</b>	A játékos elindítja a játékot.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	<b>1.</b> A játékos elindítja a játékot
<b>Alternatív forgatókönyv</b>	<b>1.A.</b> Ha nincs elég játékos, akkor nem kezdődik el a játék.

<b>Use-case neve</b>	Képesség használata
<b>Rövid leírás</b>	A karakter használja a hozzá tartozó képességet.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	<b>1.</b> A karakter alkalmazza a neki megfelelő képességet.

<b>Use-case neve</b>	Iglu építése
<b>Rövid leírás</b>	A eszkimó iglut épít.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	<b>1.</b> Az eszkimó típusú karakter képesség használatkor iglut épít azon a jégtáblán, amelyiken áll.
<b>Alternatív forgatókönyv</b>	<b>1.A.</b> Ha van már iglu a jégtáblán, akkor nem tud építeni. <b>1.B.</b> Ha van befagyott tárgy a jégtáblán, akkor nem tud építeni. <b>1.C.</b> Ha van hó a jégtáblán, akkor nem tud iglut építeni.

<b>Use-case neve</b>	Jégtábla megnézése
<b>Rövid leírás</b>	A sarkkutató megvizsgálja a jégtáblát.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	<b>1.</b> A sarkkutató típusú karakter megvizsgálja, hogy a kiválasztott jégtábla hány karaktert bír el. Csak azt a jégtáblát tudja megnézni, amelyiken áll, vagy amelyik a közvetlen szomszédjában van.

<b>Use-case neve</b>	Karakter ás
<b>Rövid leírás</b>	A karakter munkát végezve egy jégtáblán tevékenykedik.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	<b>1.</b> A karakter munkát befektetve vagy havat lapátol, ha van hó.

<b>Use-case neve</b>	Karakter felvesz dolgot
<b>Rövid leírás</b>	A karakter munkát végezve egy befagyott tárgyat felvesz.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	<b>1.</b> A karakter felveszi a jégtáblába befagyott tárgyat.
<b>Alternatív forgatókönyv</b>	<b>1.A.</b> A karakter egy búvárruhát ás ki, amit később felhasználhat. <b>1.B.</b> A karakter egy ásót ás ki, amit később felhasználhat. <b>1.C.</b> A karakter egy kötelet ás ki, amit később felhasználhat. <b>1.D.</b> A karakter éleelmet ás ki, amit azonnal felhasznál, növelve a testhőjét. <b>1.E.</b> Nem tud tárgyat felvenni, ha a táblán még van hó.

<b>Use-case neve</b>	Játékkör vége
<b>Rövid leírás</b>	A játékos befejezi a körét.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos jelzi, hogy az adott karakterrel nem kíván a körben több tevékenységet végezni.</li> <li>2. A játékos továbbadja a kört a következő karakternek.</li> </ol>

<b>Use-case neve</b>	Karakter mozog
<b>Rövid leírás</b>	A karakter változtatja a helyzetét.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A karakter egy jégtábláról egy másikra megy.</li> </ol>
<b>Alternatív forgatókönyv</b>	<ol style="list-style-type: none"> <li>1.A. A karakter nem léphet olyan helyre, ahol tenger van.</li> <li>1.A.1. Ha a karakter búvárruhában van a tengerben, akkor úszhat szomszédos mezőre.</li> <li>1.B. A karakter stabil jégtáblára lép.</li> <li>1.C. A karakter instabil jégtáblára lép.</li> <li>1.C.1. Ha az instabil jégtáblára túl sokat léptek, akkor beszakad és a rajta állók vízbe esnek.</li> </ol>

<b>Use-case neve</b>	Összeszerel
<b>Rövid leírás</b>	A karakterek összerakják a pisztolyt.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. Ha minden kellék egy jégtáblán álló karakterknél van, akkor az éppen soron lévő, ott álló karakter egy munkával összeszerelheti a jelzőpisztolyt.</li> </ol>

<b>Use-case neve</b>	Játék vége
<b>Rövid leírás</b>	A játék győzelemmel vagy vereséggel véget ér.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játék véget ér.</li> </ol>
<b>Alternatív forgatókönyv</b>	<ol style="list-style-type: none"> <li>1.A. A játék vereséggel ér véget.</li> <li>1.A.1. Egy vagy több karakter vízbe esve megfullad.</li> <li>1.A.2. Egy vagy több karakter hóviharban megfagy.</li> <li>1.B. A játék győzelemmel ér véget.</li> <li>1.B.1. A játékosok összeszerelik és elsütik a jelzőpisztolyt.</li> </ol>

<b>Use-case neve</b>	Karakter leesik
<b>Rövid leírás</b>	Egy vagy több karakter a tengerbe esik.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. Egy instabil jégtáblát túlterhelve a rajta álló karakterek a tengerbe esnek.</li> </ol>

<b>Use-case neve</b>	Karakter megfullad
<b>Rövid leírás</b>	A tengerbe esett karakterek megfulladnak.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A túlterhelt instabil jégtábláról a karakterek a tengerbe esnek.</li> <li>2. Ha a karakteren nincs búvárruha, és senki sem tudja kihúzni, akkor megfullad, tehát a játék véget ér.</li> </ol>

<b>Use-case neve</b>	Karakter megfagy
<b>Rövid leírás</b>	A hóvihar miatt egy vagy több karakter megfagy.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A feltámadó hóvihar csökkenti pár szabadban lévő karakter testhőjét.</li> <li>2. A karakterek elfogy a testhője, ezért megfagy.</li> <li>3. A karakter halála miatt a játék véget ér.</li> </ol>

<b>Use-case neve</b>	Feltámad a vihar
<b>Rövid leírás</b>	A jégmezőn feltámad egy hóvihar.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A hóvihar érkezését a játék előre jelzi.</li> <li>2. A hóvihar a kör elején lesújt, néhány jégtáblára.</li> <li>3. Az ott lévő igluk összedőlnek, a szabadban lévő karakterek testhőt veszítenek.</li> </ol>

<b>Use-case neve</b>	Lecsap a vihar a pálya egy pontjára
<b>Rövid leírás</b>	A karakter bemenekül egy erre alkalmas helyre.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A feltámadó hóvihar elől a karakter menekülni kezd.</li> <li>2. Bemegy az adott jégtáblán álló épületbe, például egy igluba, ahol átvészeli a vihart, és nem veszít testhőt.</li> <li>3. Az iglu a vihar után elpusztul.</li> </ol>
<b>Alternatív forgatókönyv</b>	2.A. Ha az adott igluban már van valaki, akkor nem tud bemenni.

<b>Use-case neve</b>	Karakter kihúz
<b>Rövid leírás</b>	A karakter kimenekít egy vagy több karaktert a vízből.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. Egy vagy több karakter tengerbe esik.</li> <li>2. A szomszédos táblán álló karakter automatikusan kihúzza a beesett karaktereket, ha van nála kötél.</li> </ol>
<b>Alternatív forgatókönyv</b>	2.A. Ha emiatt többen lesznek egy instabil jégtáblán, akkor az is beszakad.

## **5.2 A szkeleton kezelői felületének terve, dialógusok**

### **Tervezett kimeneti formátum:**

```

függvényhívás1: f {object} {function}
    függvényhívás2: f {object} {function}
    visszatérés2: r {object} {function}
visszatérés1: r {object} {function}

```

A szkeleton felhasználói felülete a fent bemutatott formában fogja megjeleníteni a program működését. A függvényben történő függvényhívás a konzolon kicsit beljebb fog megjelenni, ezzel is növelve az átláthatóságot.

Azt, hogy melyik teszeset fusson le, egy menüből lehet kiválasztani, ehhez be kell írni a kiválasztott teszeset sorszámát.

### **Dialógusok:**

#### **Játék indítása**

Bemenet:

- a. Van elég játékos az indításhoz? (0: igen, 1: nem)

Kírja, hogy sikeresen elindult-e a játék vagy nem.

#### **Képesség használata**

Bemenet:

- a) Melyik karakter képességét használod? (0: explorer, 1: eskimo)

#### **Karakter ás**

Bemenet:

- a) Milyen eszközzel ás? (0: kéz, 1: ásó)

#### **Karakter felvesz dolgot**

Bemenet:

- a) Mit vegyen fel a karakter? (0: étel, 1: búvárruha, 2: ásó, 3: kötél, 4: pisztoly, 5: jelzőfény, 6: töltény)

#### **Játékkör vége**

Bemenet:

- a. Tovább adja a kört a karakter? (0: igen, 1: nem)

**Karakter mozog**

Bemenet:

- a. Szabályos volt a mozgás? (0: igen, 1: nem)
- b. Beszakad a jégtábla ahova lépett a karakter? (0: igen, 1: nem)

**Karakter leesik**

Bemenet:

- a) Van búvárruhája a karakternek? (0: igen, 1: nem)

**Karakter húz**

Bemenet:

- a. Milyen eszköze van a karakternek a húzáshoz? (0: kéz, 1: kötél)

**Karakter megfullad**

Bemenet: -

**Karakter összeszerel**

Bemenet:

- a. Össze tudjuk szerelni? (0: igen, 1: nem)
- b. Egyedül meg tudjuk csinálni az összeszerelést? (0: igen, 1: nem)

**Feltámad a vihar**

Bemenet: -

**Lecsap a vihar a pálya egy pontjára**

Bemenet:

- a. Volt iglu a jégtáblán? (0: igen, 1: nem)

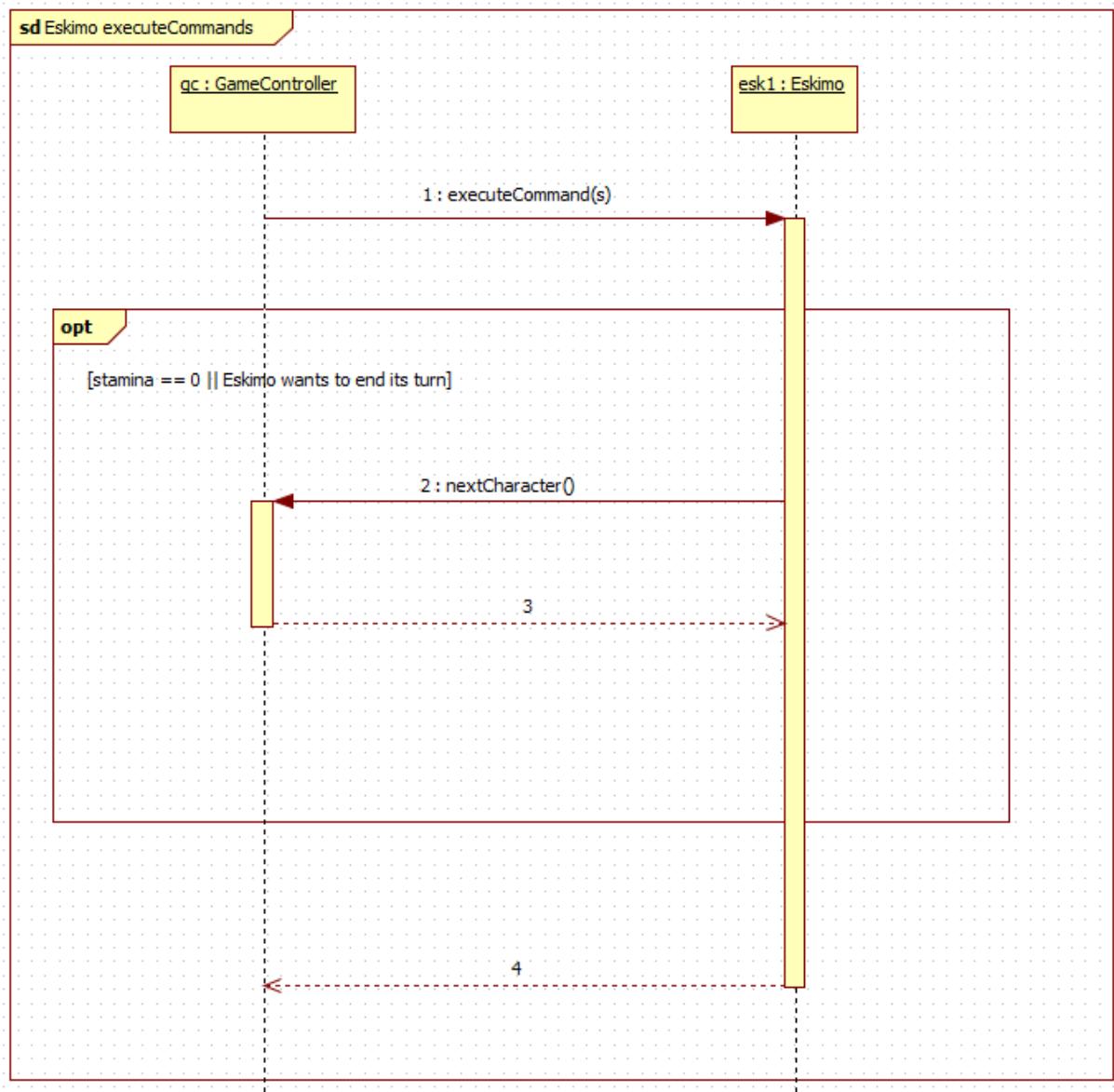
**Karakter megfagy**

Bemenet:

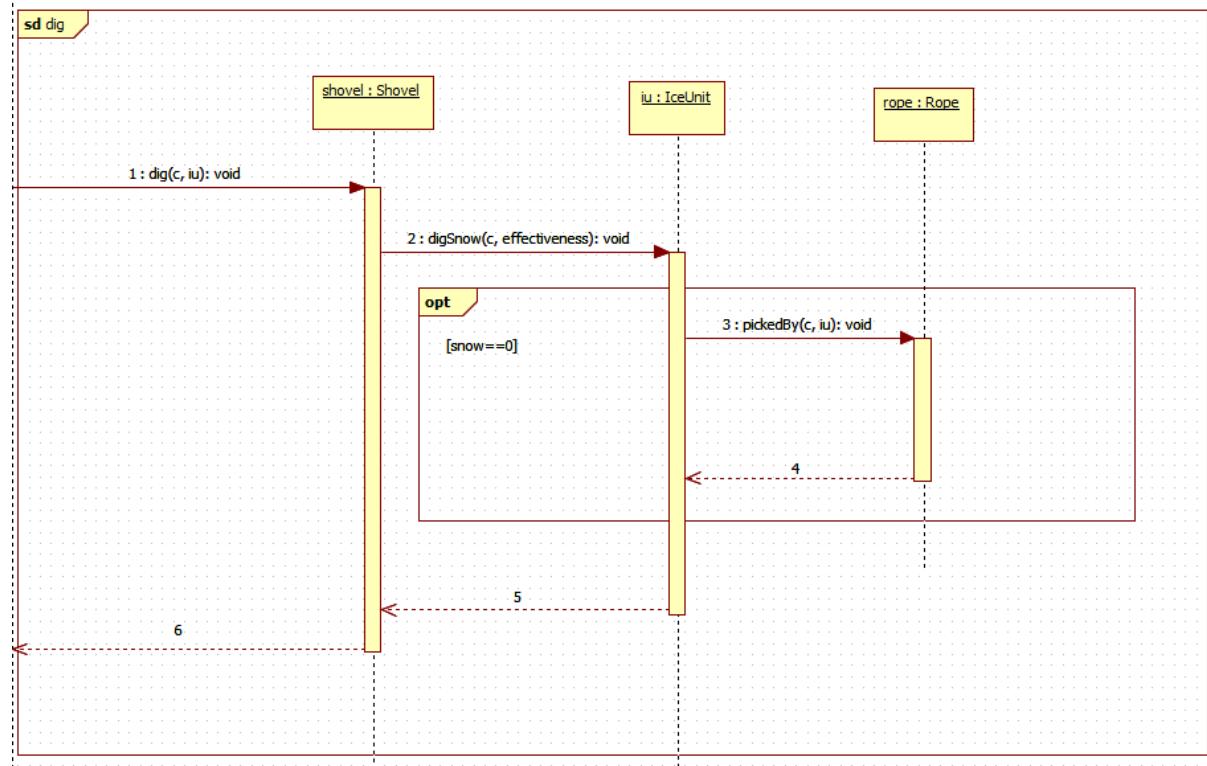
- a. Nullára csökkent a testhő? (0: igen, 1: nem)

### 5.3 Szekvencia diagramok a belső működésre

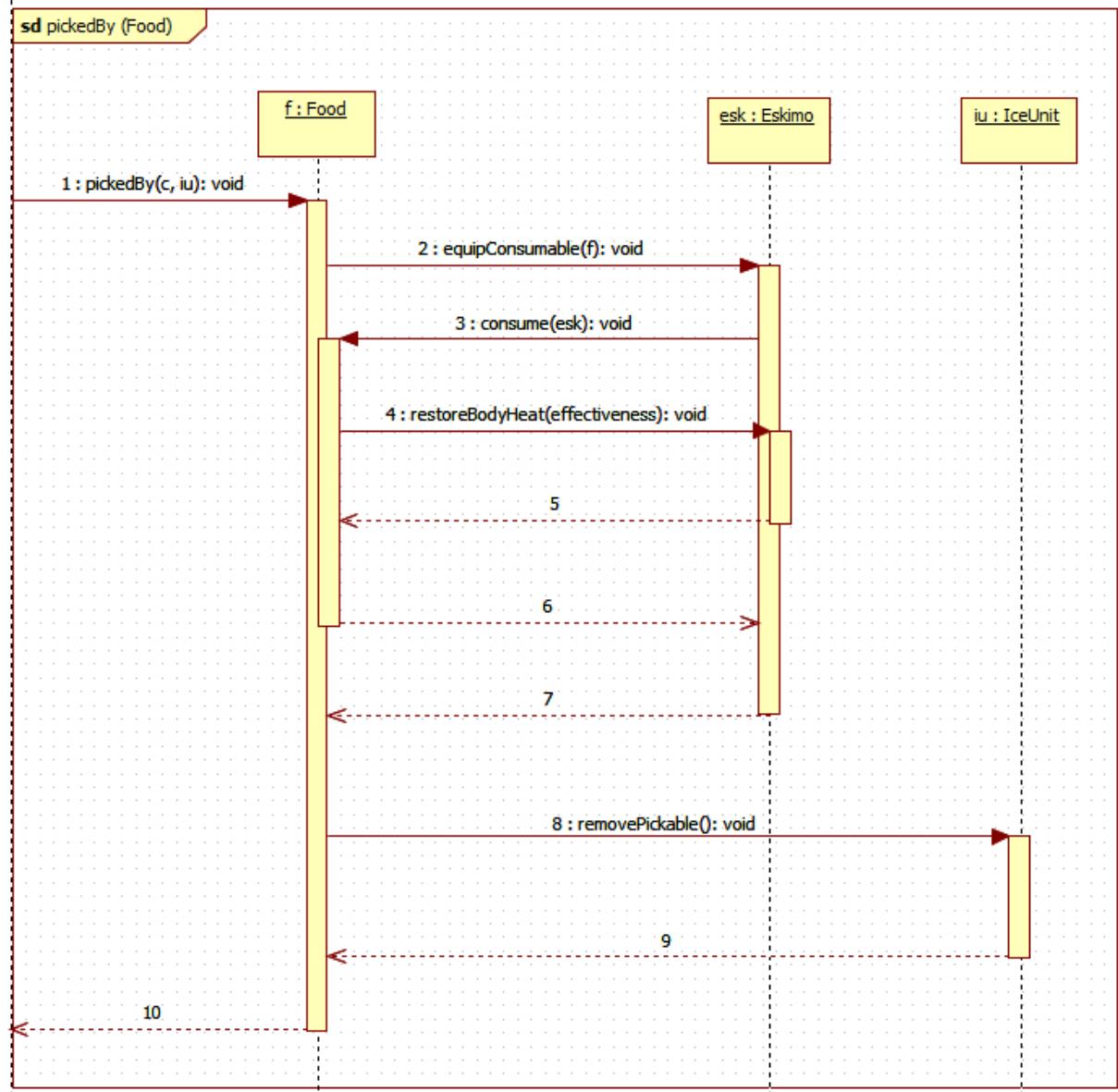
#### 5.3.1 Eskimo executeCommands

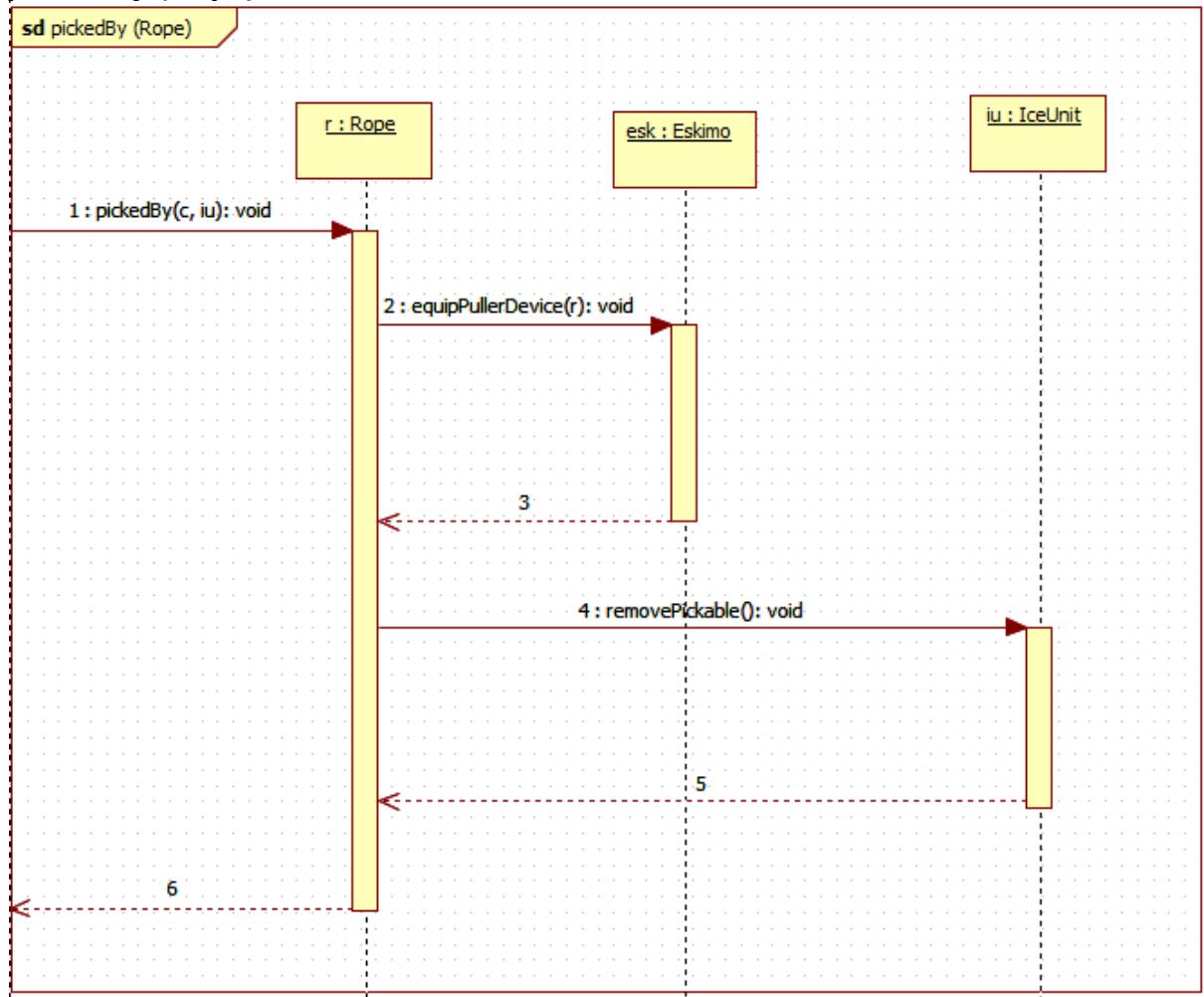


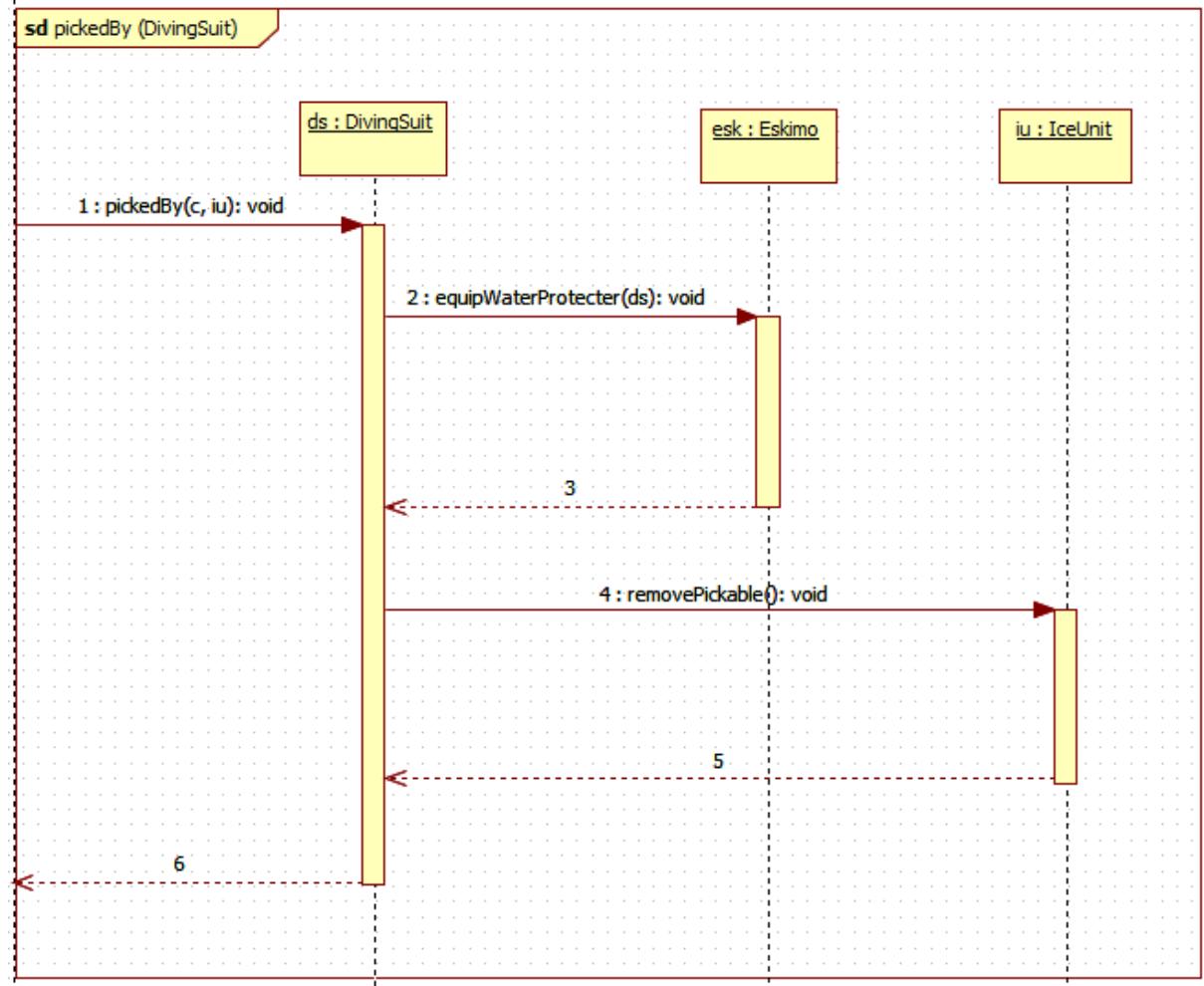
### 5.3.2 dig

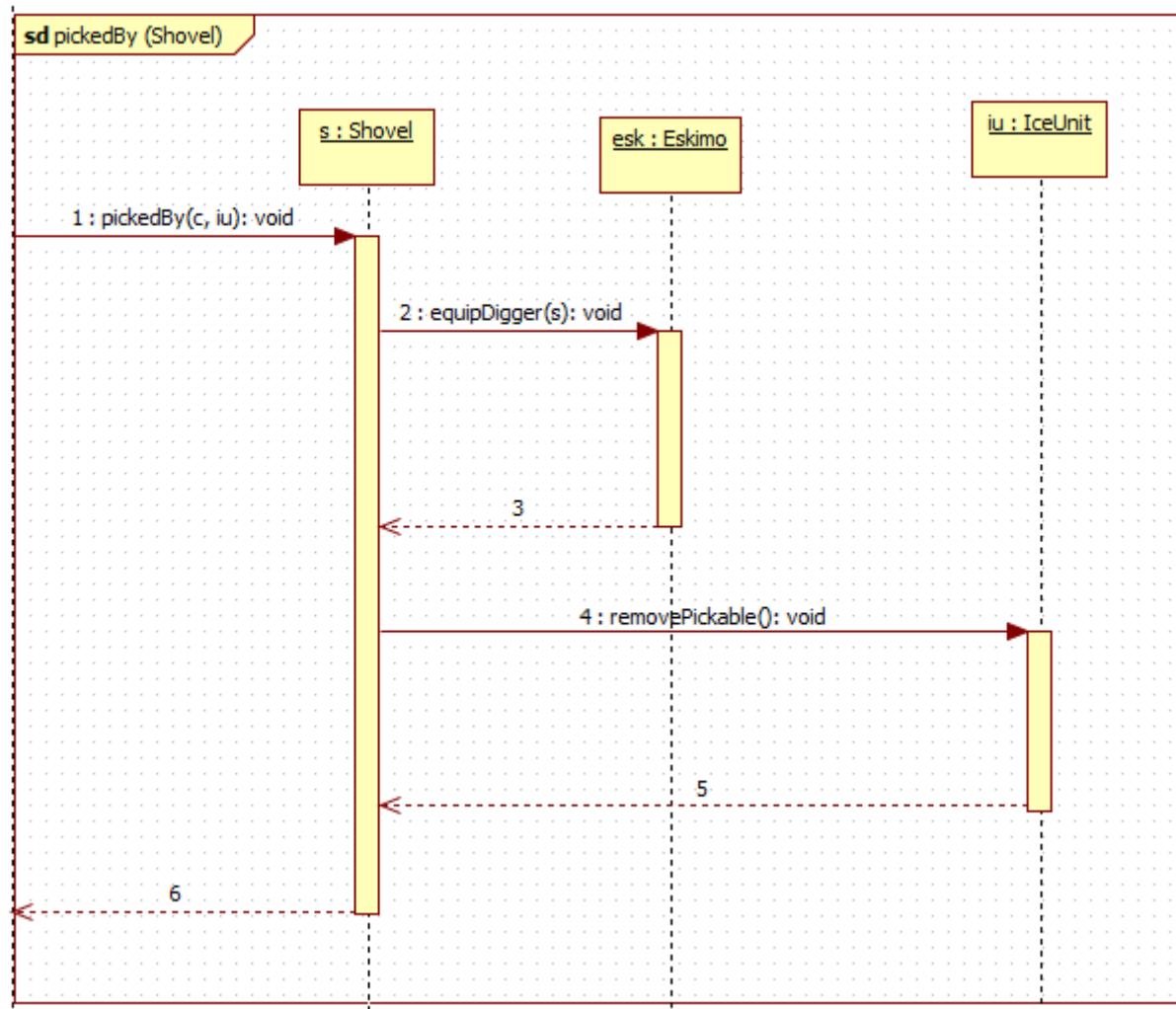


### 5.3.3 **pickedBy (Food)**

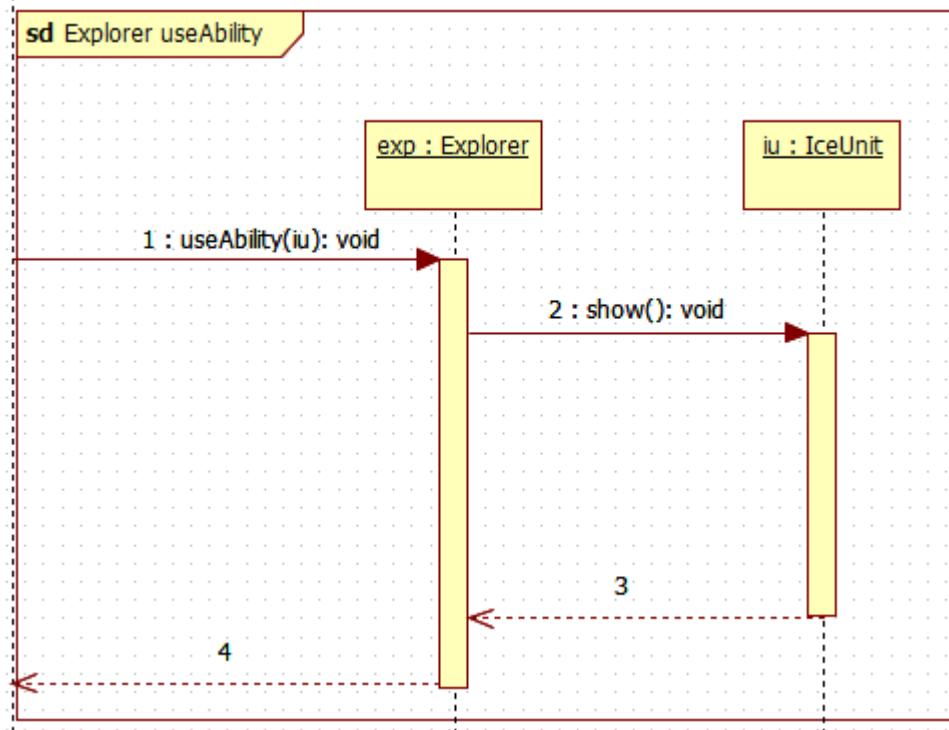


5.3.4 **pickedBy (Rope)**

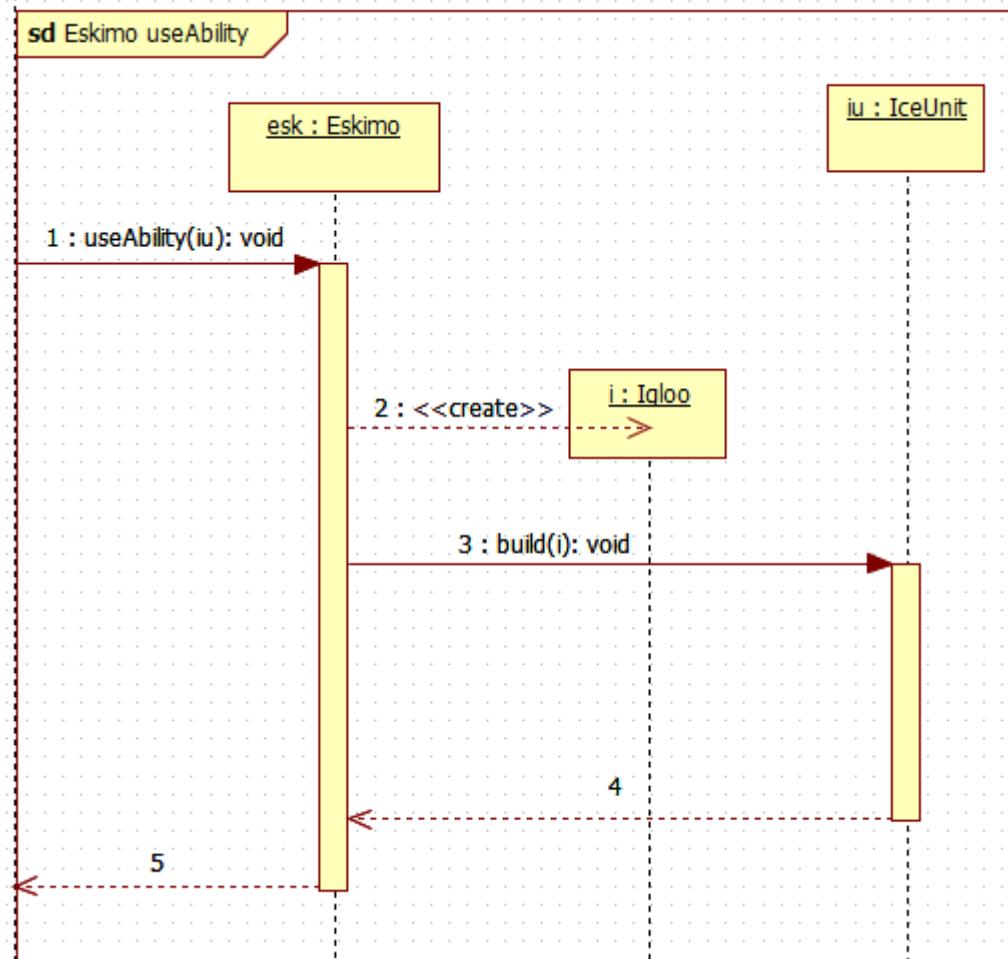
**5.3.5 *pickedBy (DivingSuit)***

**5.3.6 *pickedBy (Shovel)***

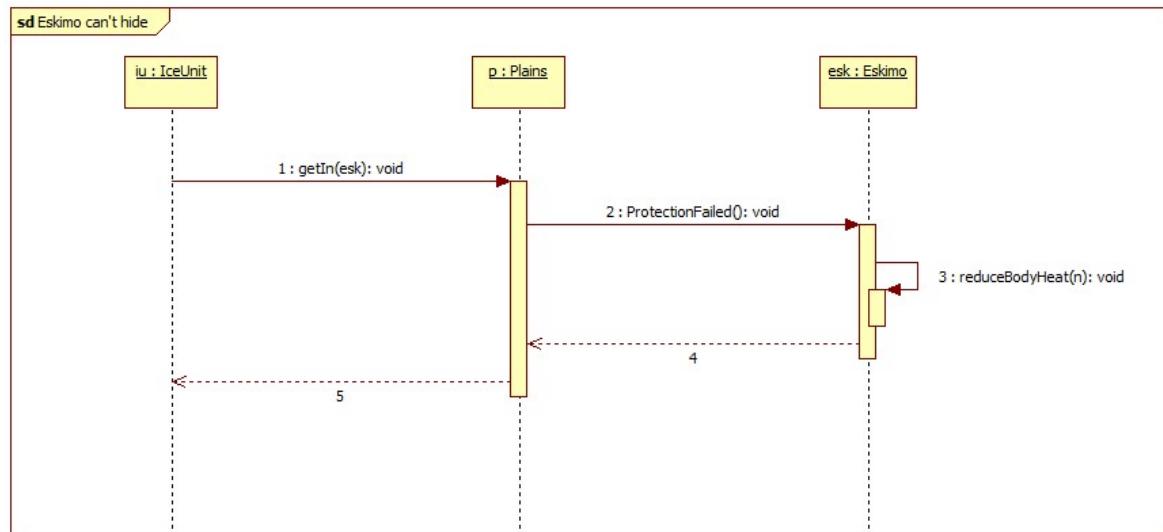
### 5.3.7 Explorer useAbility



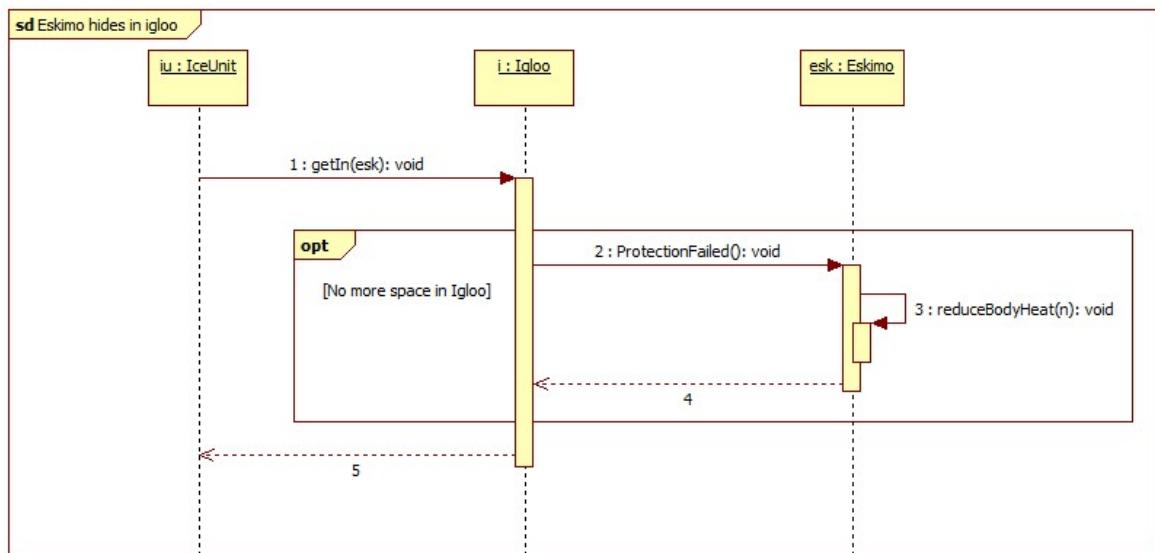
### 5.3.8 Eskimo useAbility



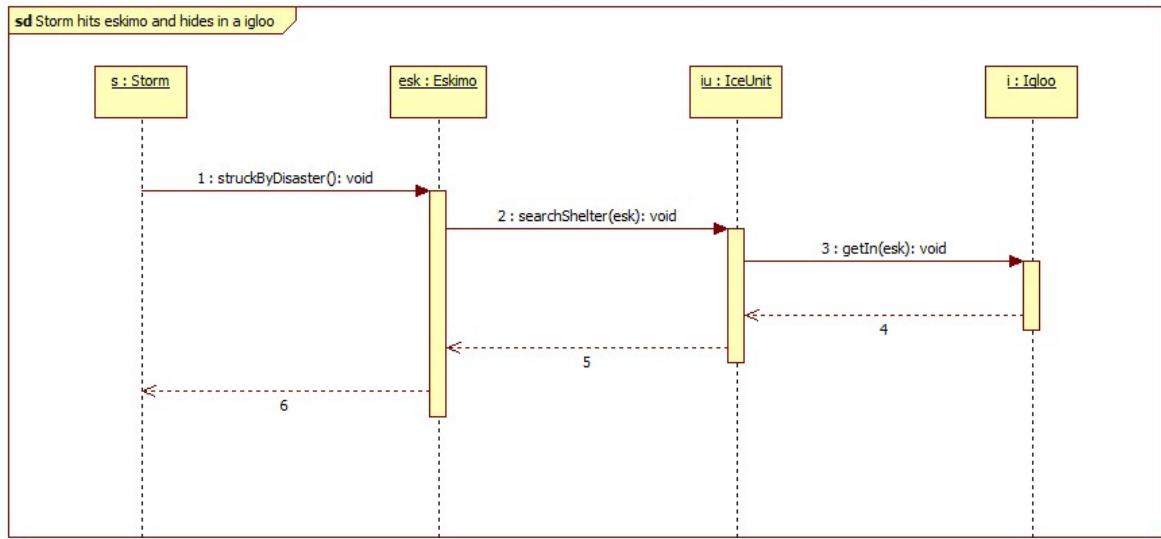
### 5.3.9 Eskimo can\_t hide



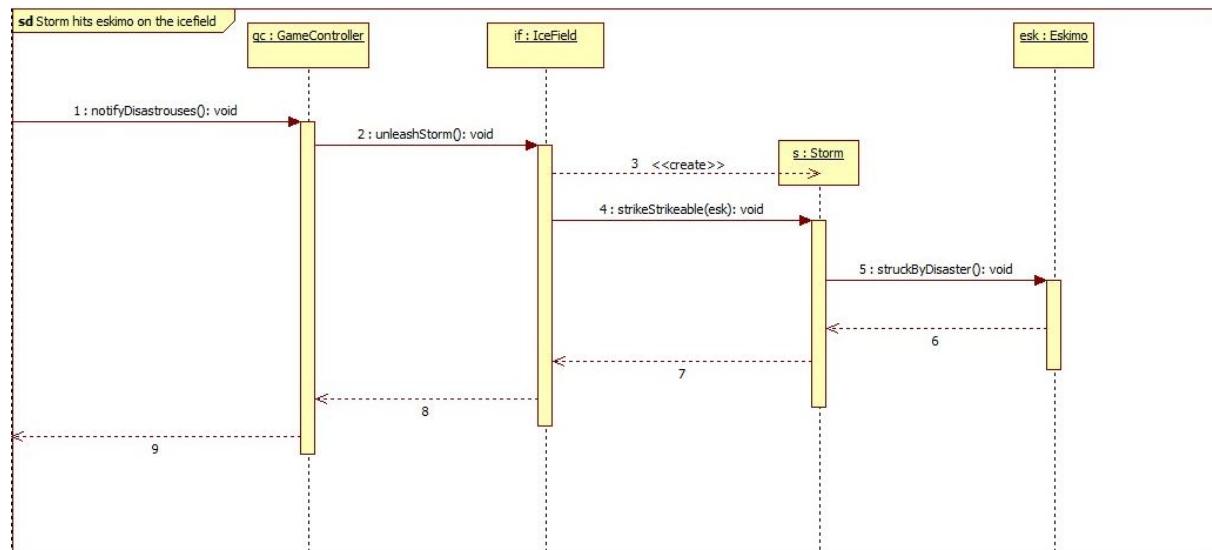
### 5.3.10 Eskimo hides in igloo



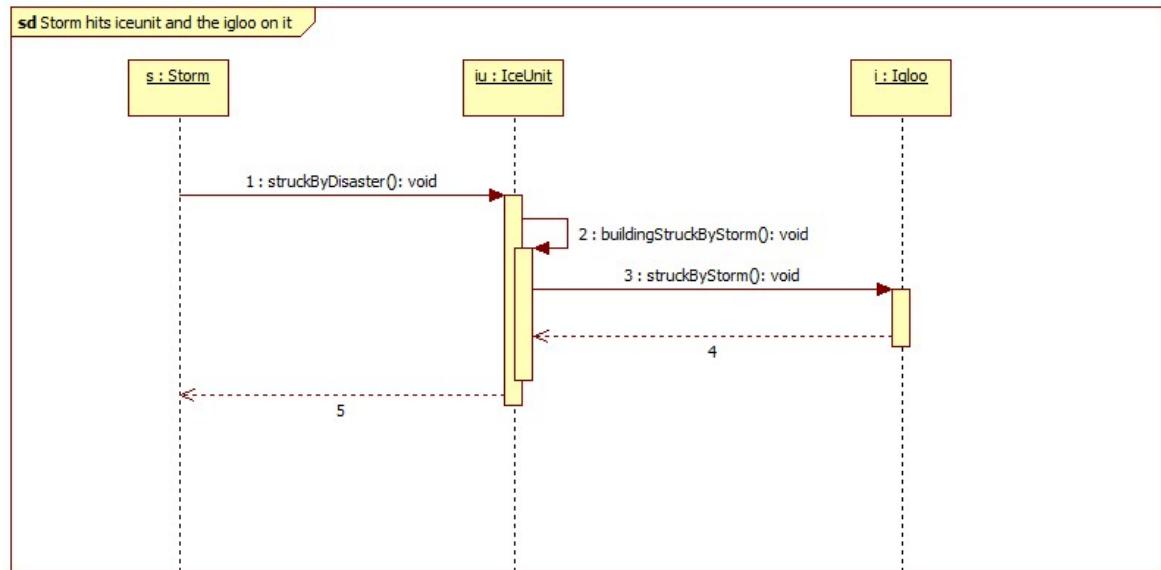
### 5.3.11 Storm hits eskimo and hides in a igloo



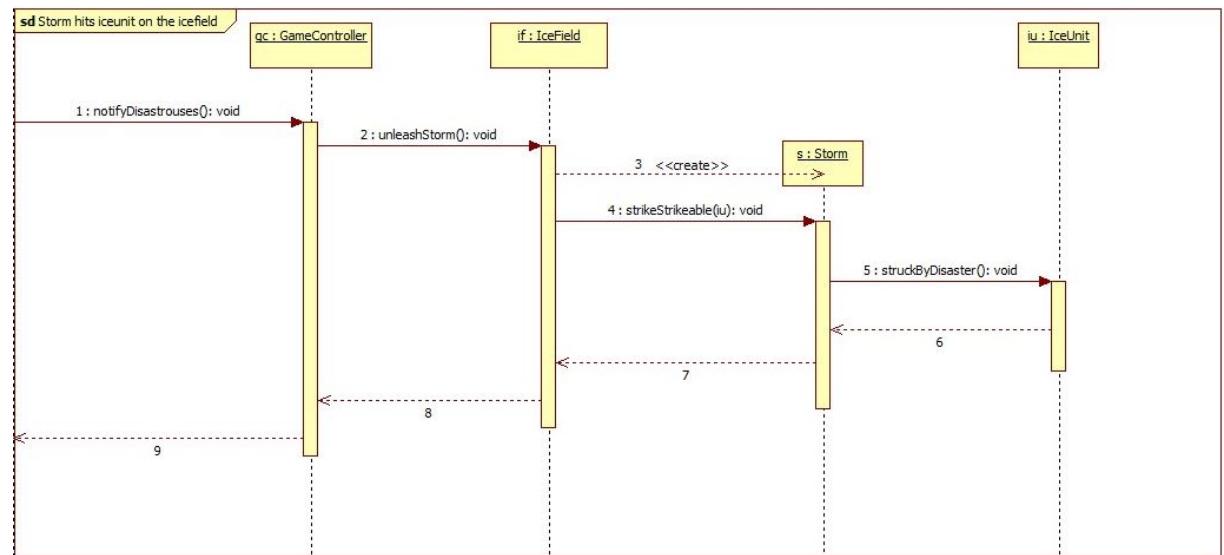
### 5.3.12 Storm hits eskimo on the icefield



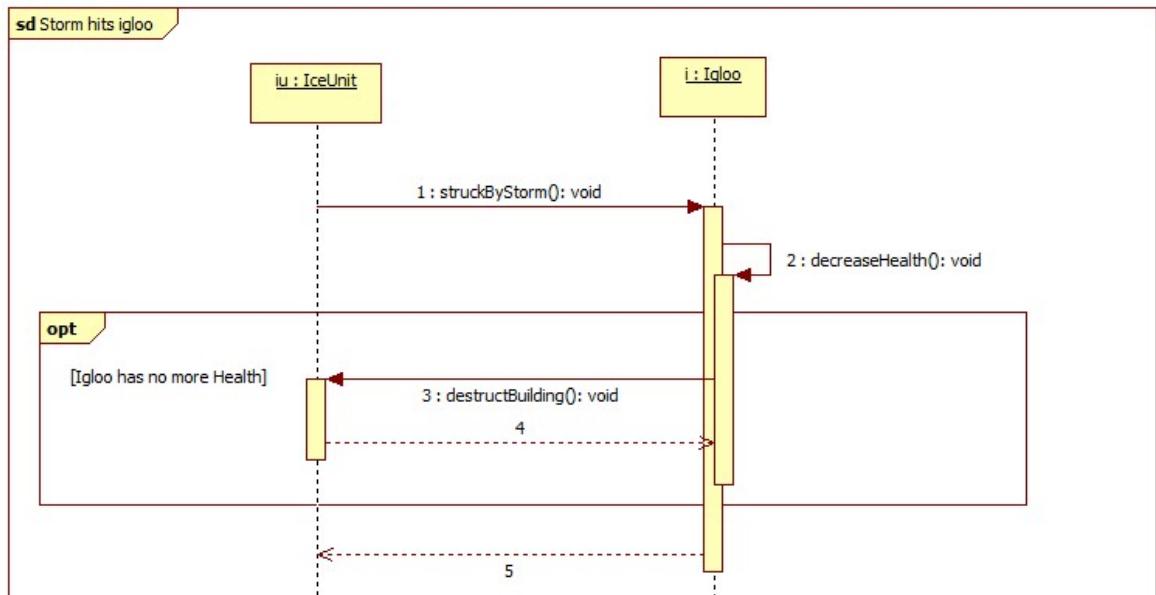
### 5.3.13 Storm hits iceunit and the igloo on it



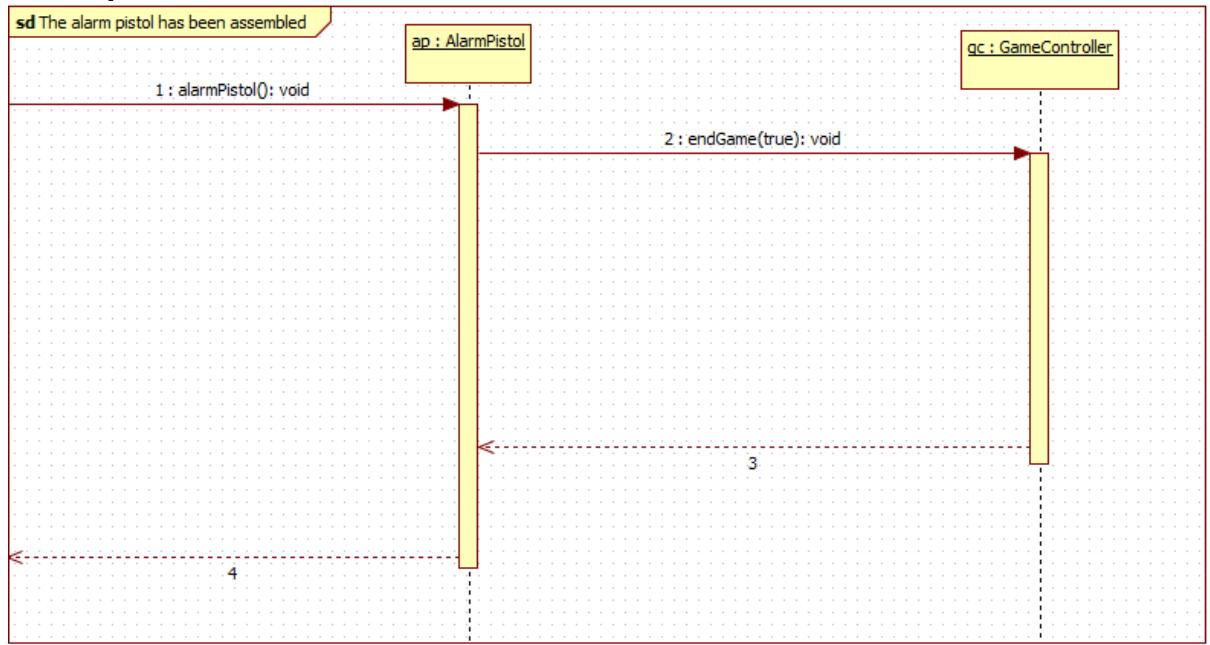
### 5.3.14 Storm hits iceunit on the icefield



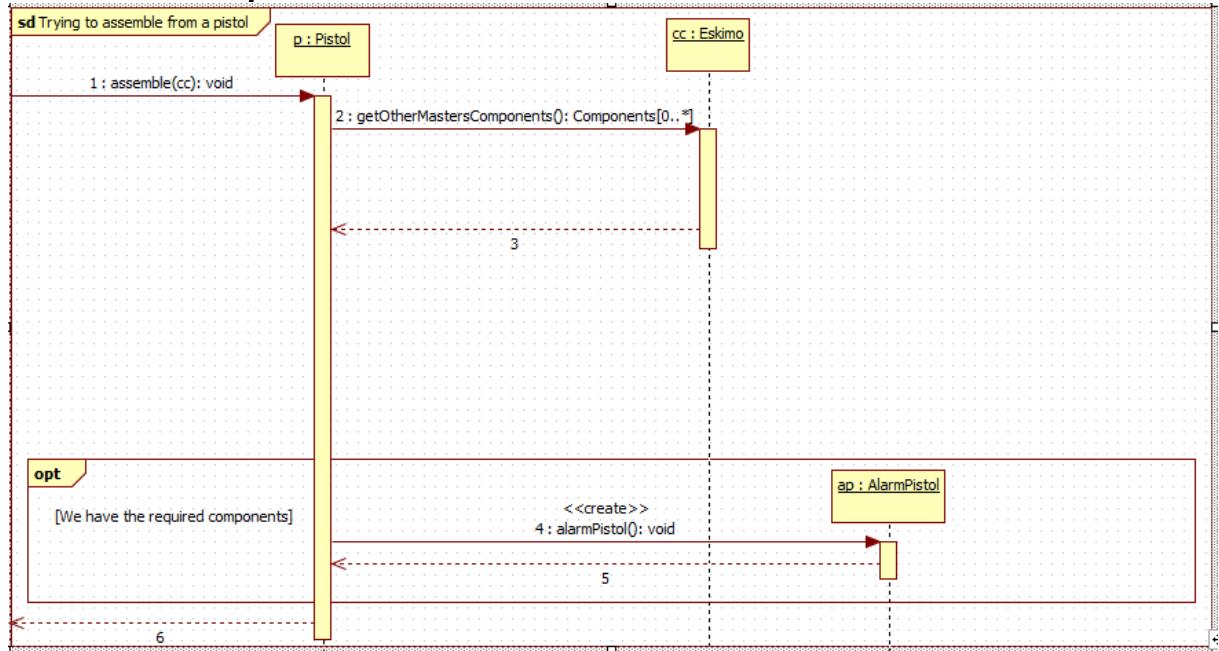
### 5.3.15 Storm hits igloo



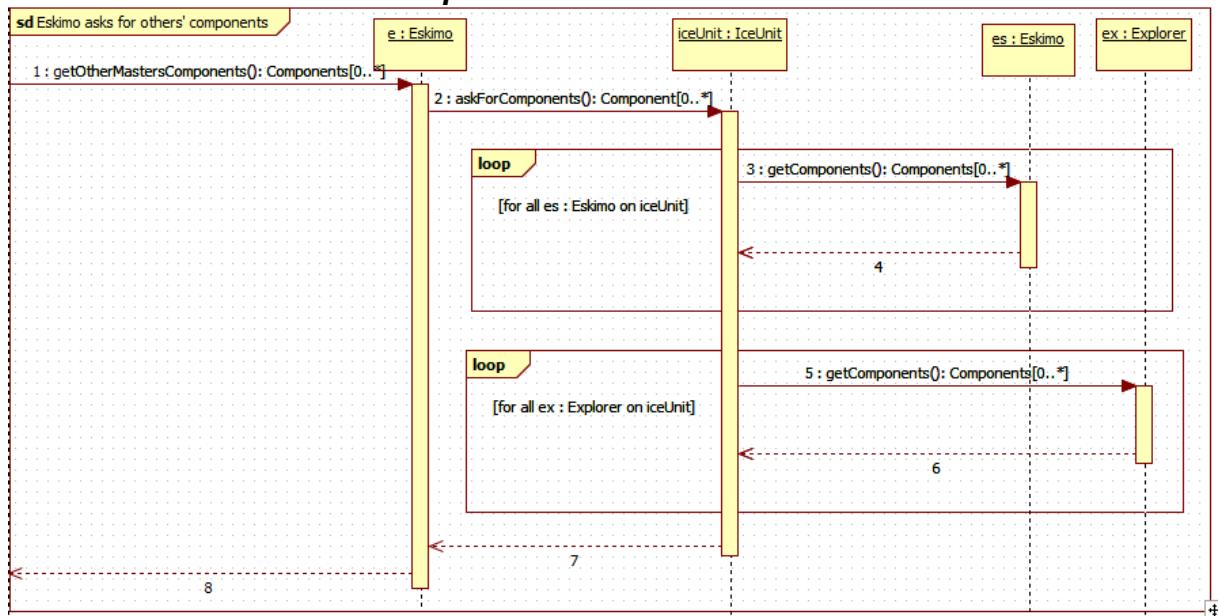
### 5.3.16 Alarm pistol assembled



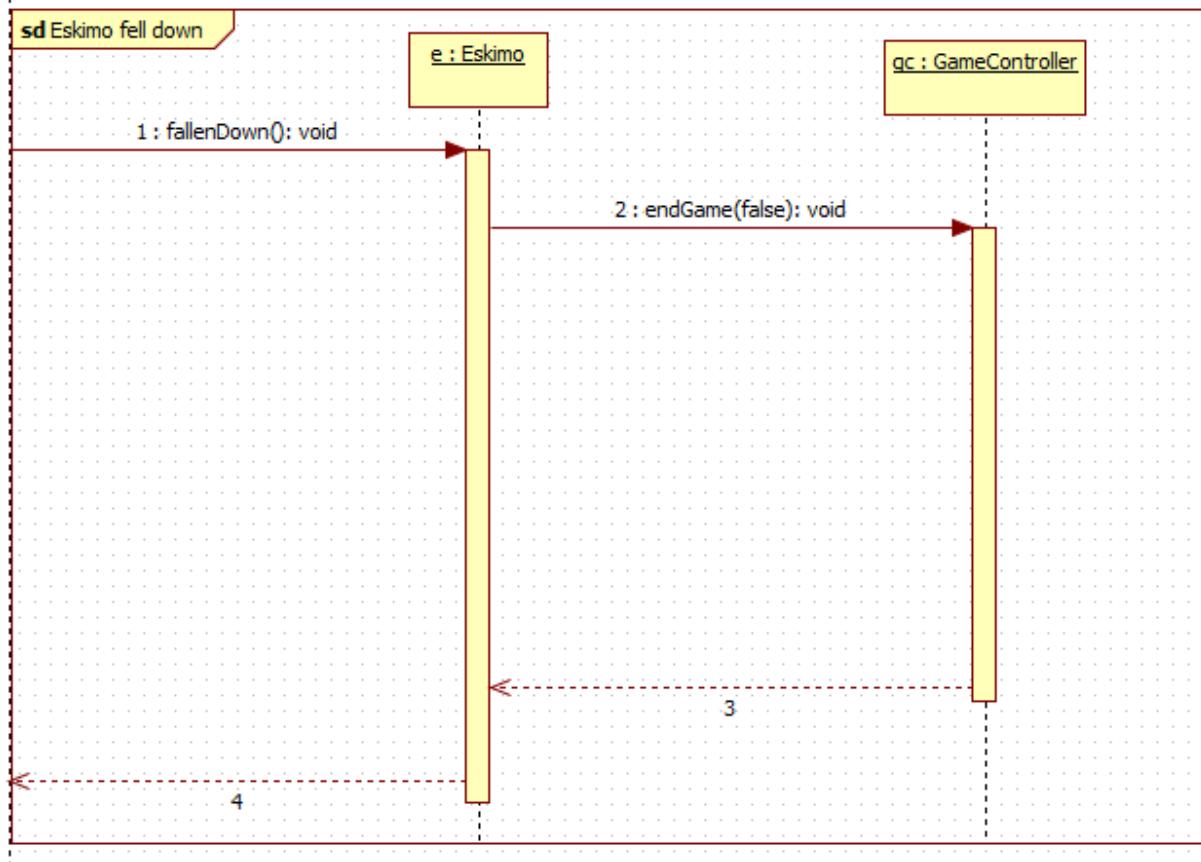
### 5.3.17 Assemble from pistol



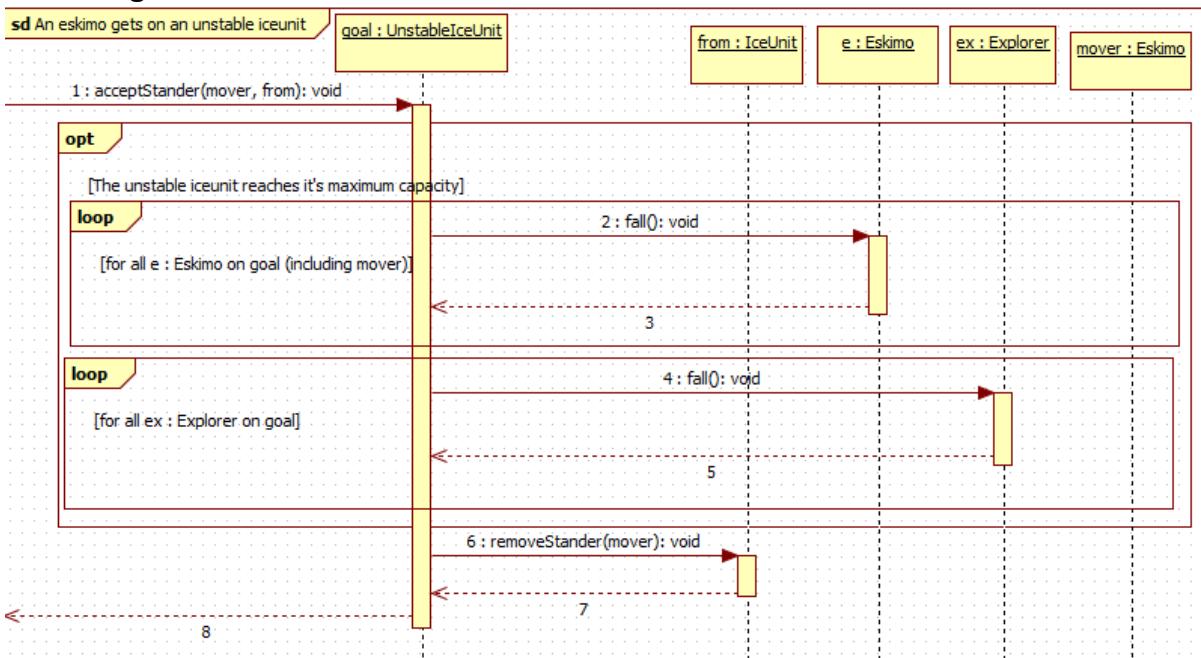
### 5.3.18 Eskimo asks for others' components



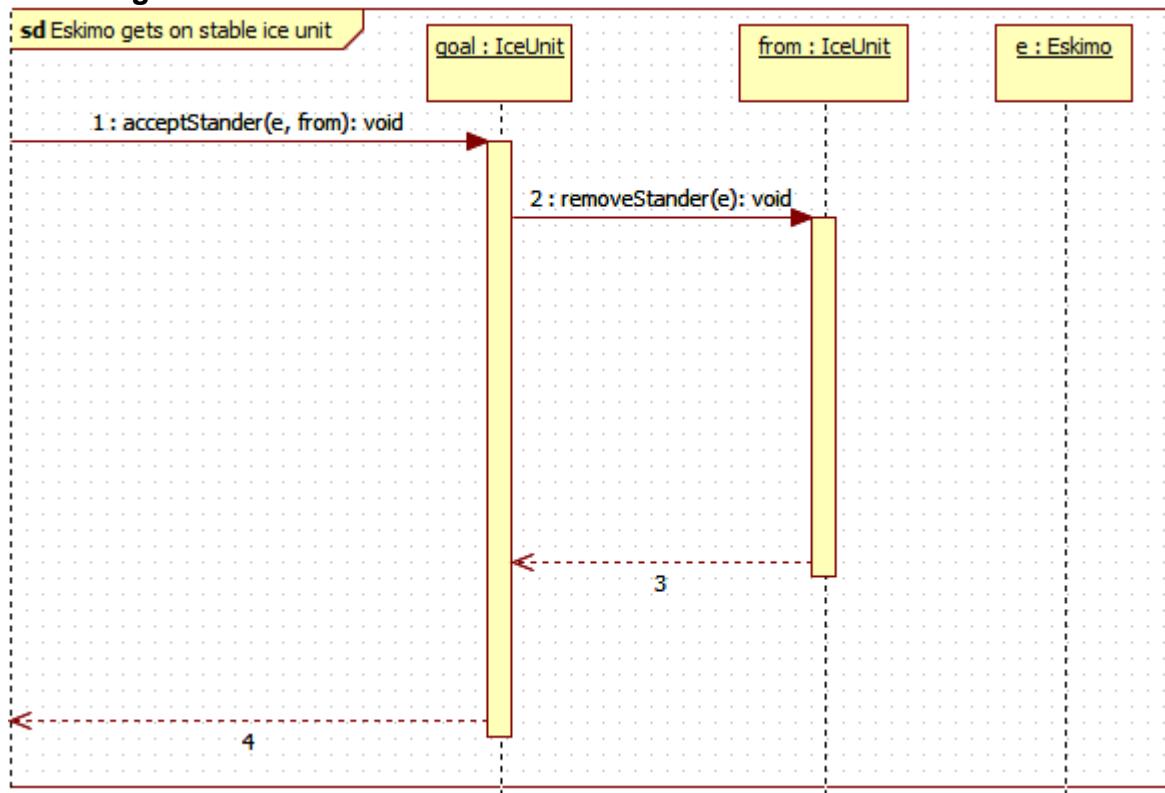
### 5.3.19 Eskimo fell down



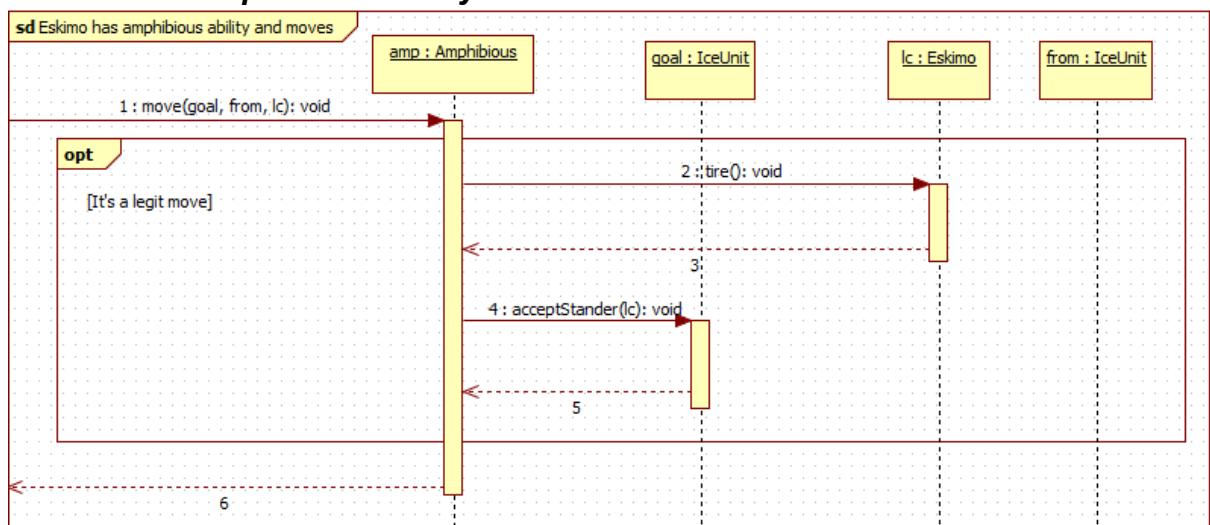
### 5.3.20 Eskimo gets on an unstable iceunit



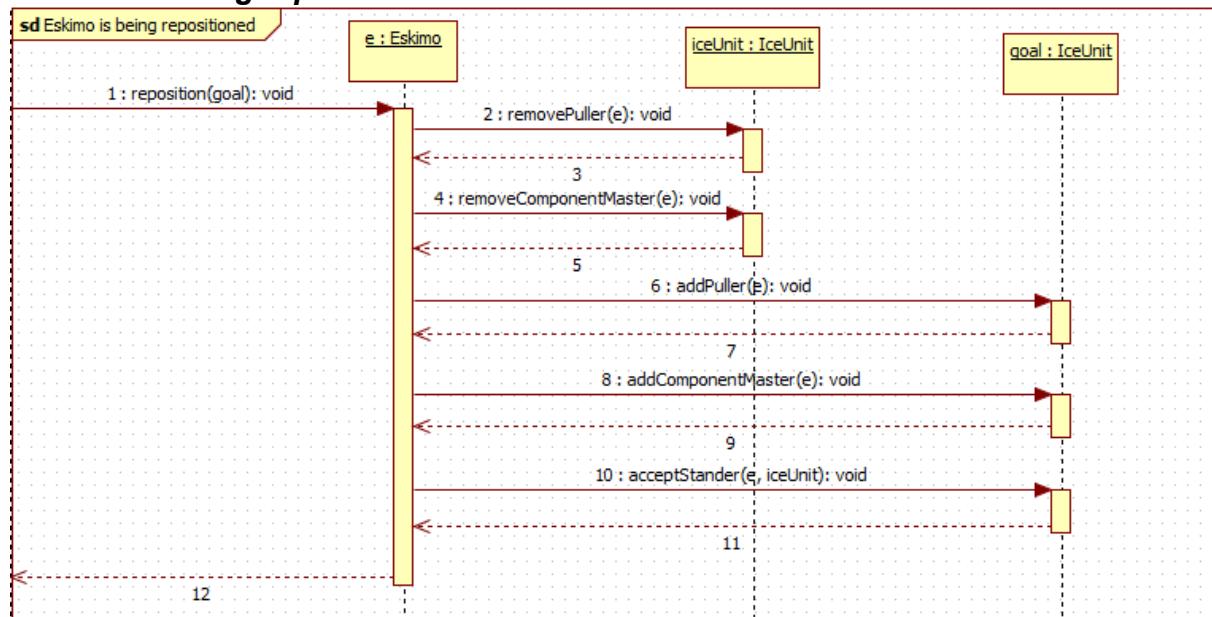
### 5.3.21 Eskimo gets on stable iceunit



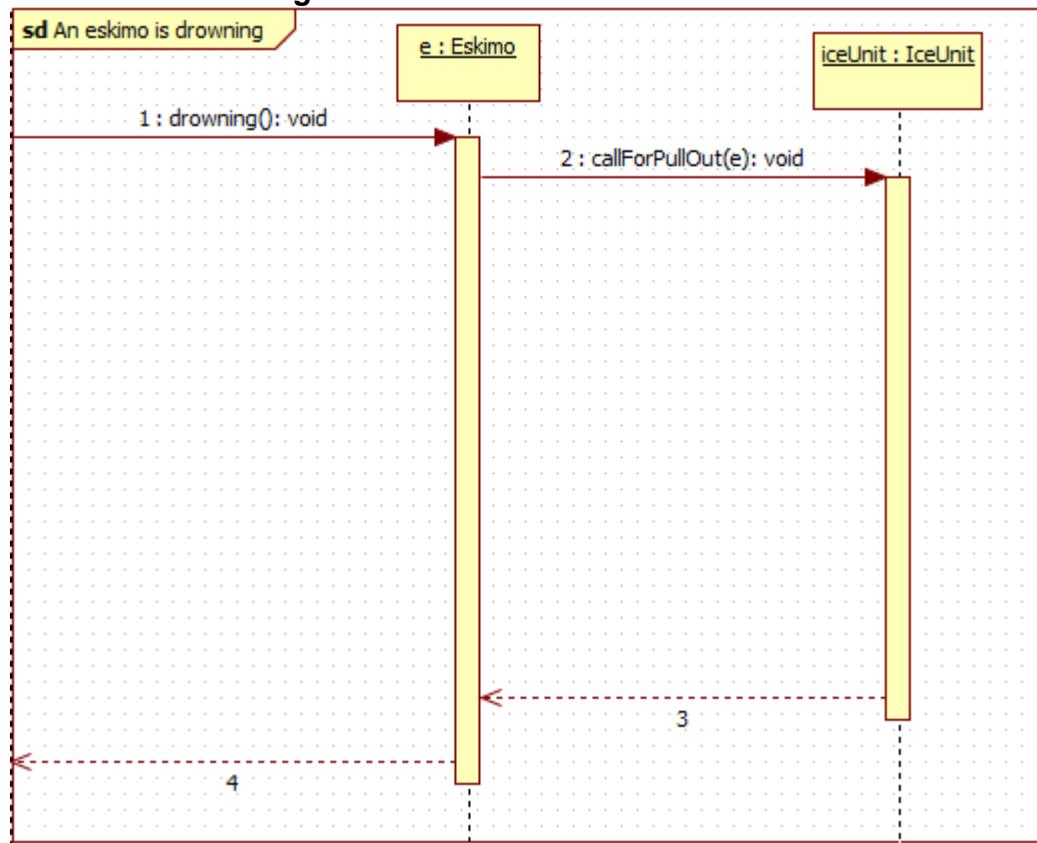
### 5.3.22 Eskimo has amphibious ability and moves



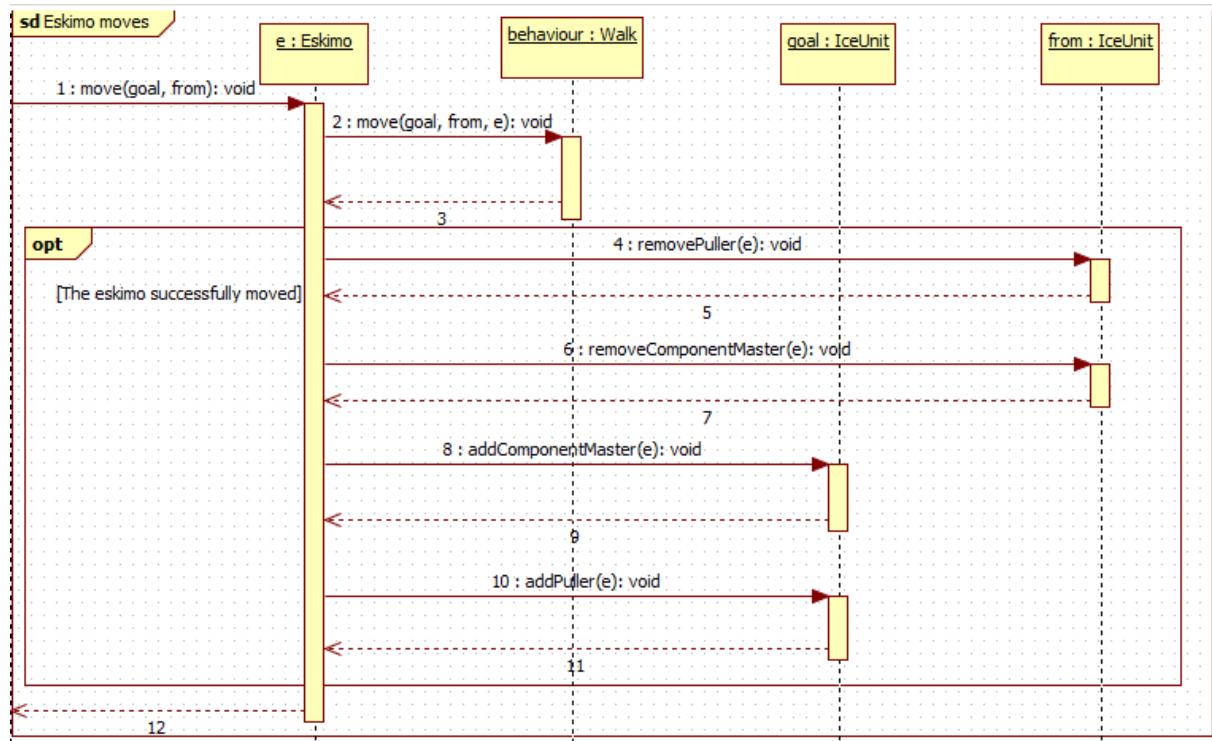
### 5.3.23 Eskimo is being repositioned



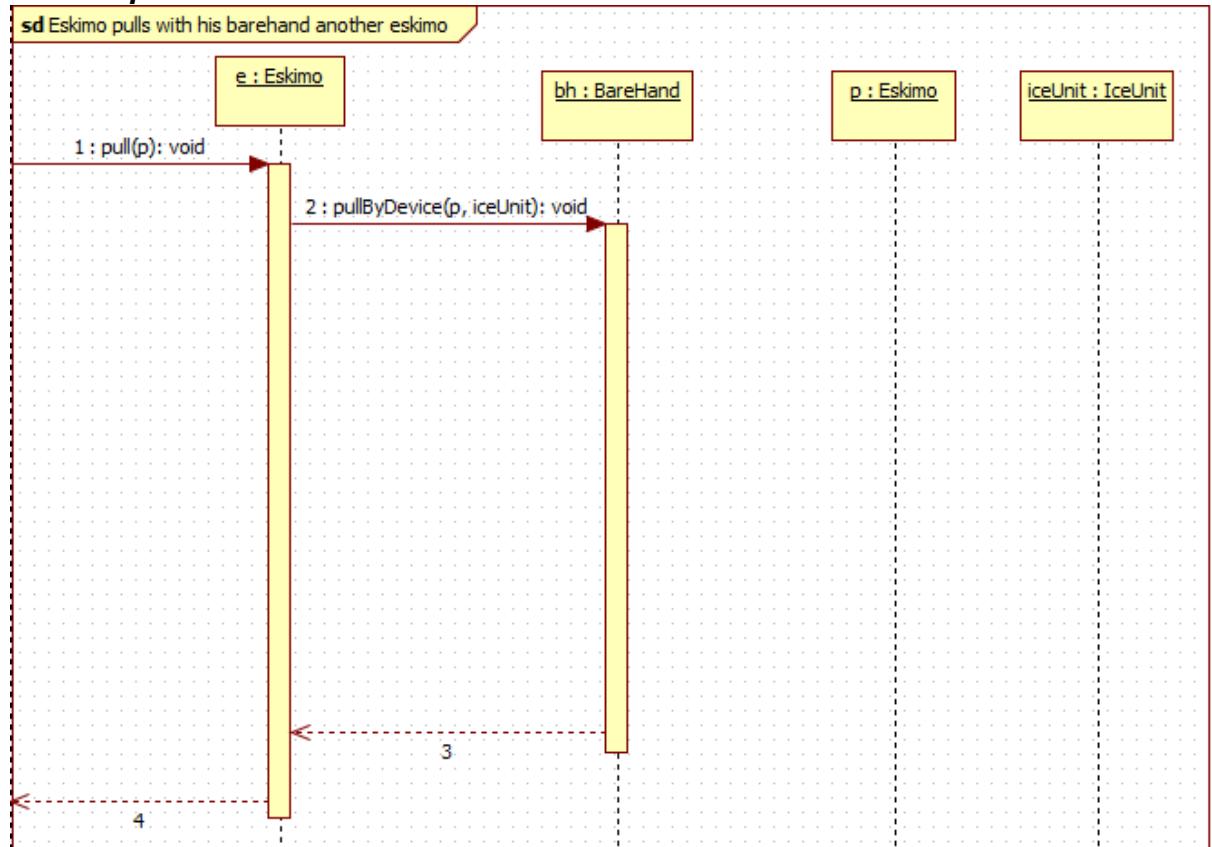
### 5.3.24 Eskimo is drowning



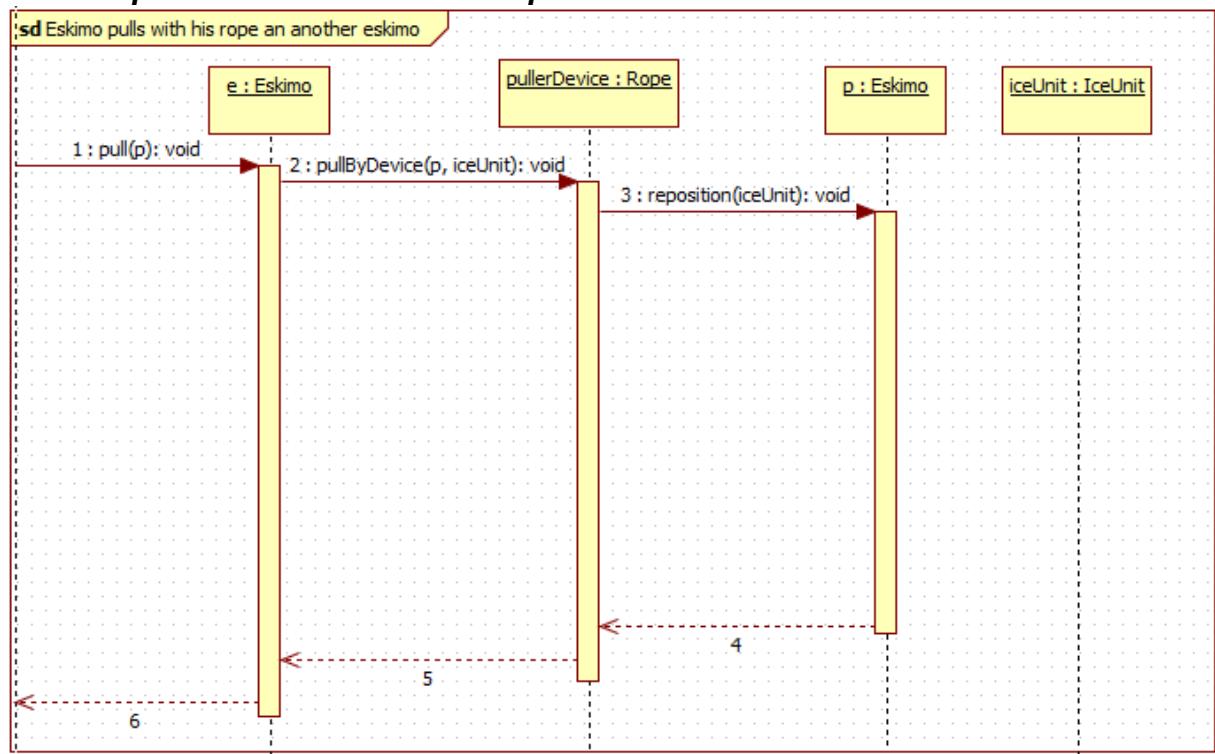
### 5.3.25 Eskimo moves



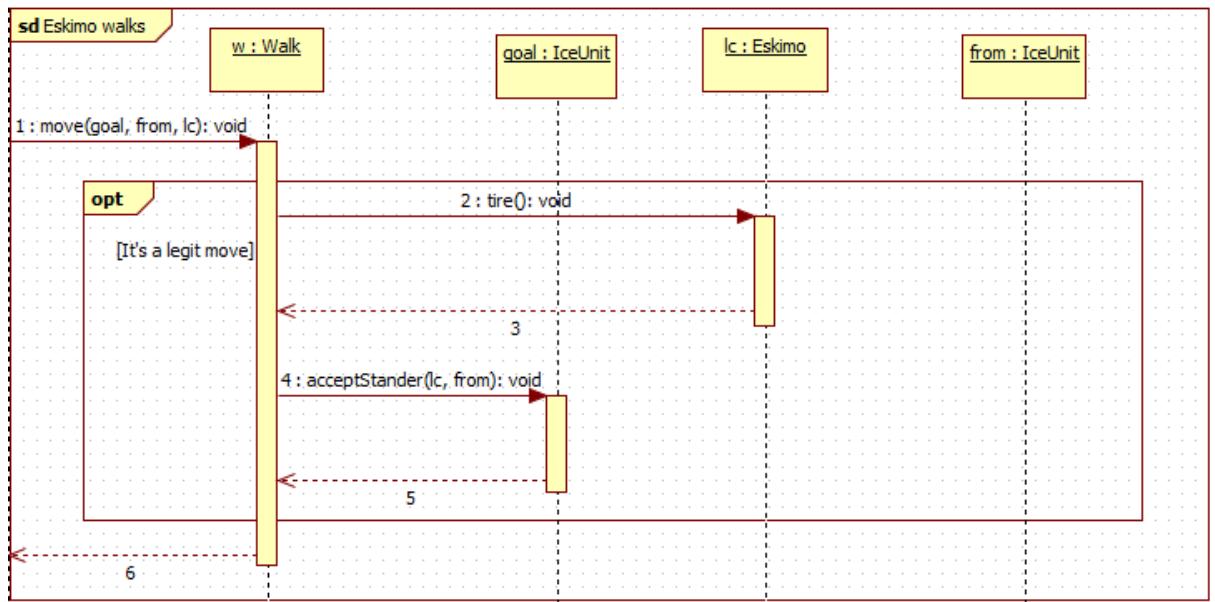
### 5.3.26 eskimo pulls another eskimo with barehand

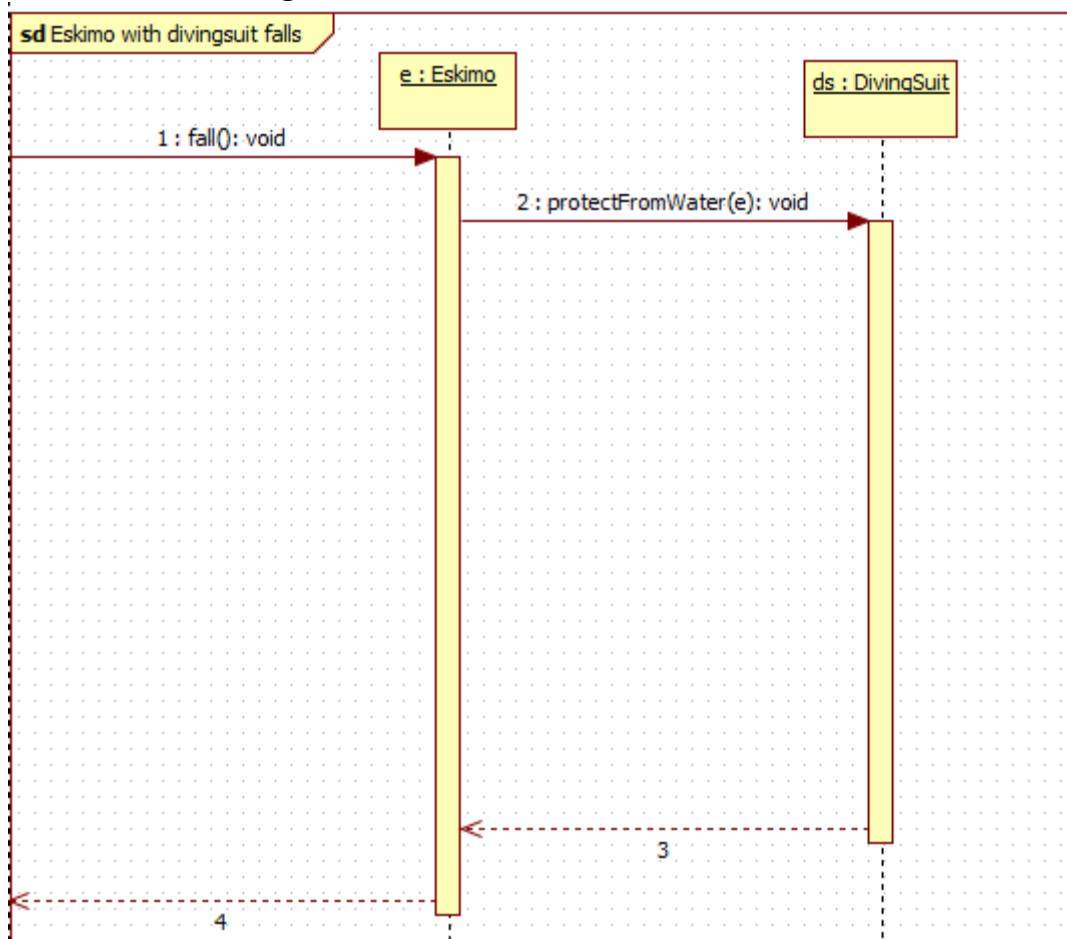


### 5.3.27 Eskimo pulls other eskimo with rope

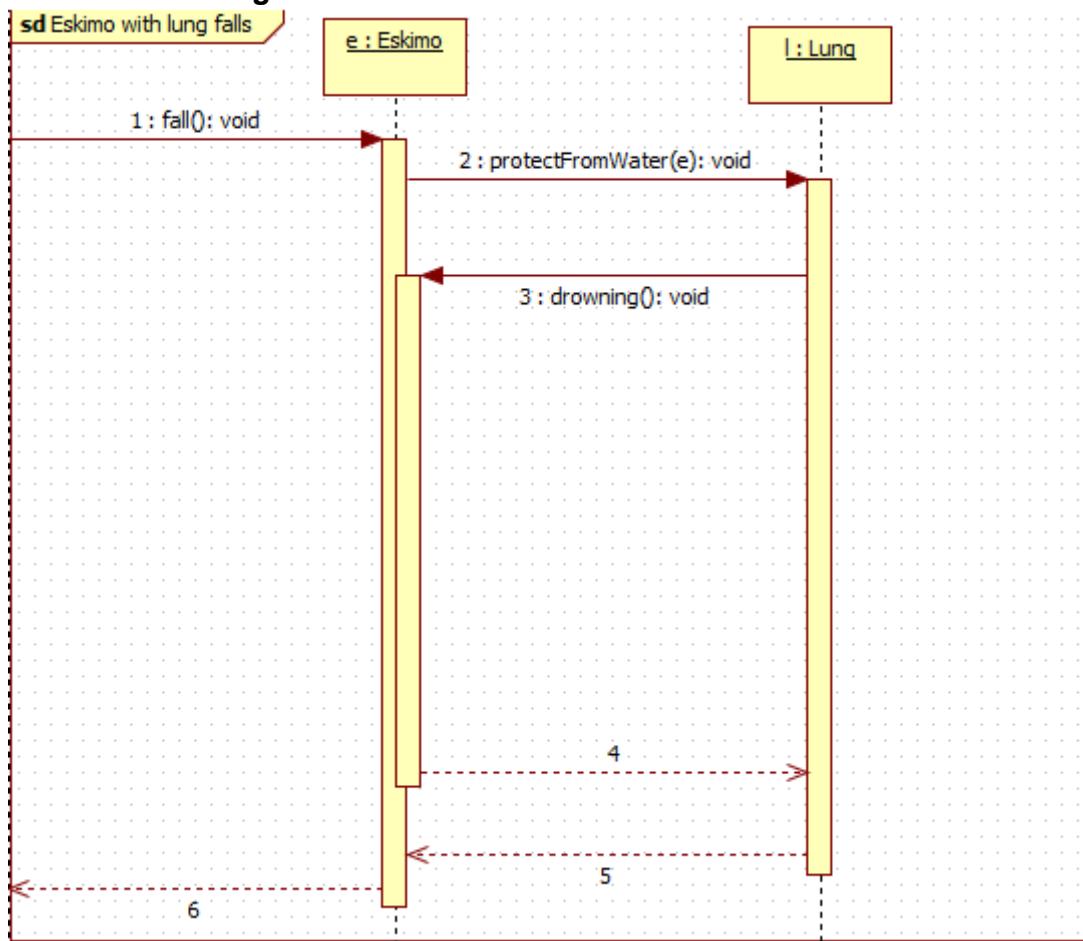


### 5.3.28 Eskimo walks

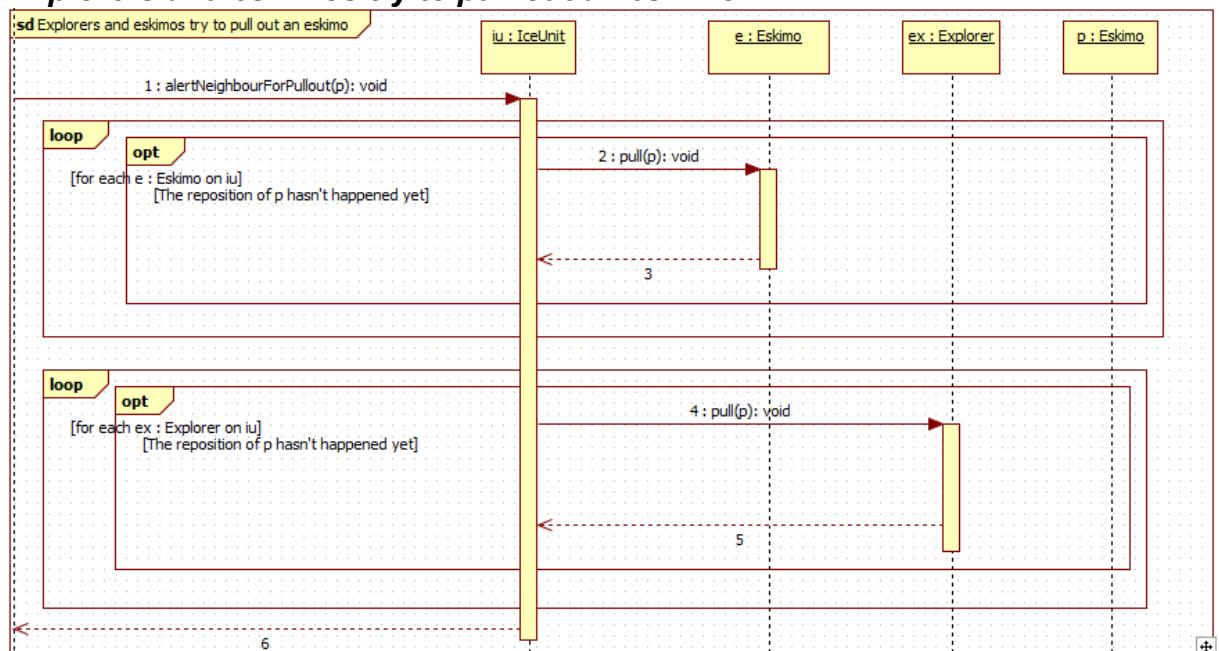


**5.3.29 Eskimo with divingsuit falls**

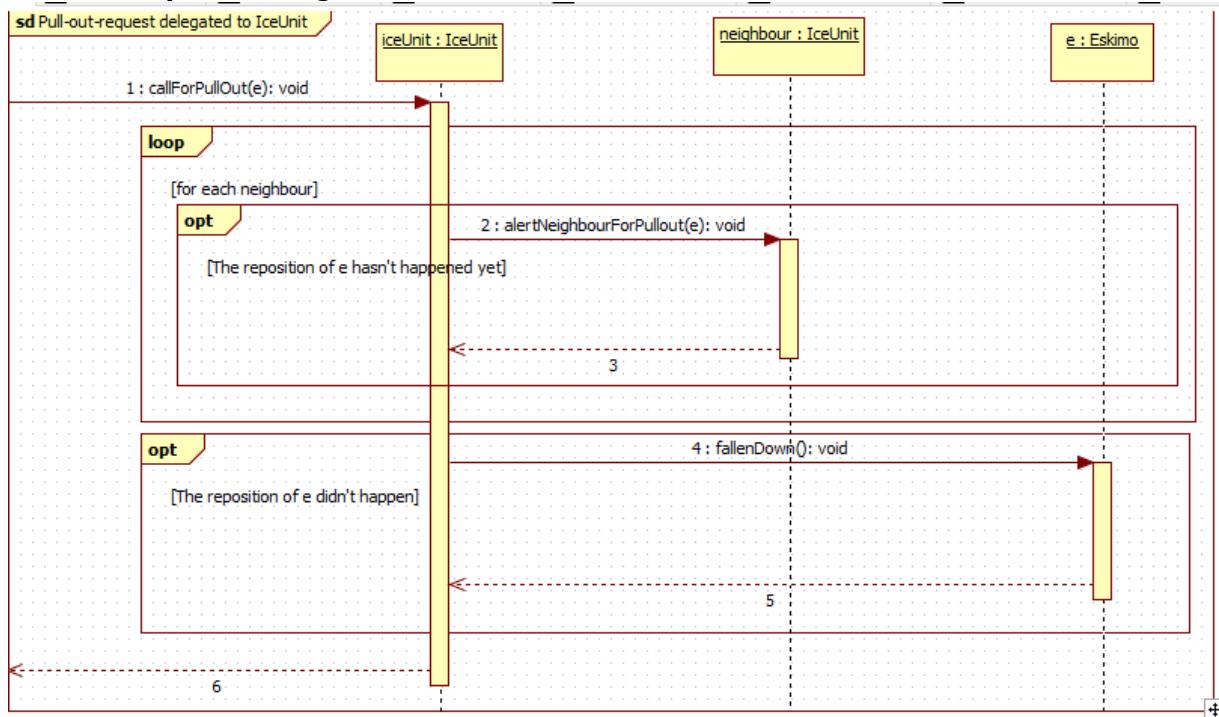
### 5.3.30 Eskimo with lung falls



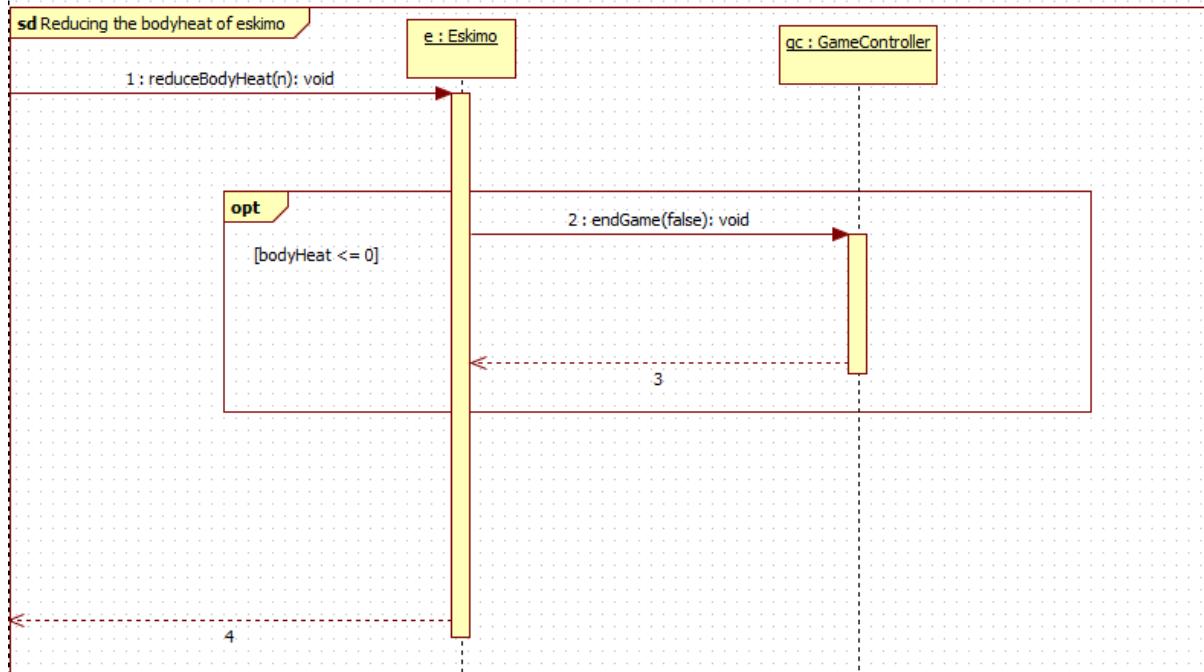
### 5.3.31 Explorers and eskimos try to pull out an eskimo



### 5.3.32 Pull-out request delegated to iceunit

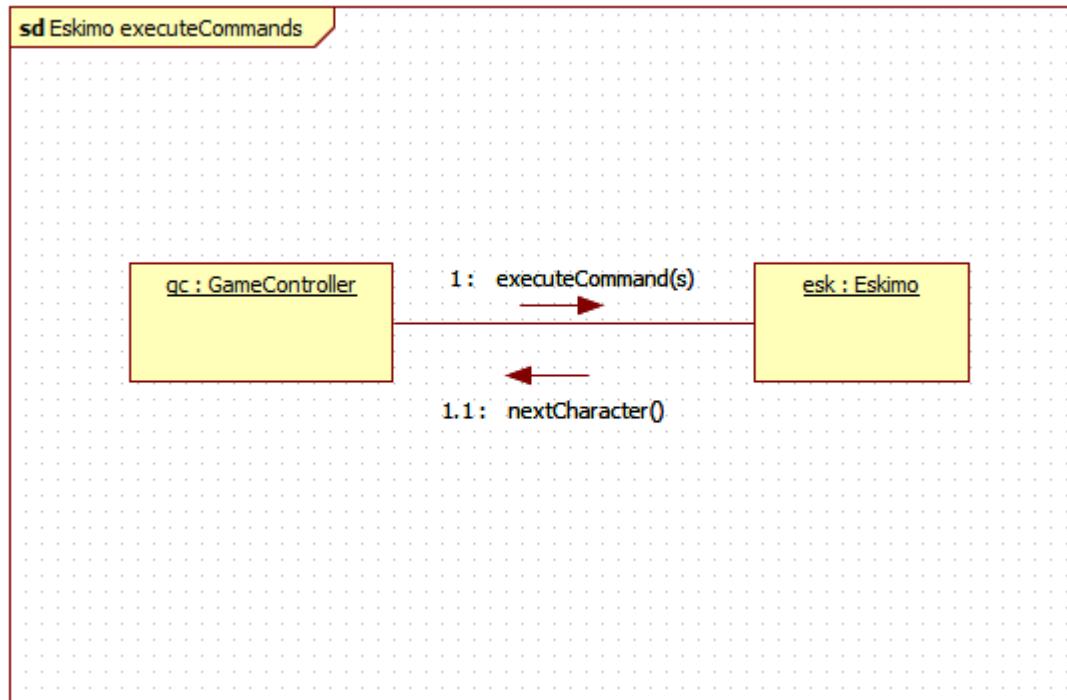


### 5.3.33 Reducing the bodyheat of eskimo

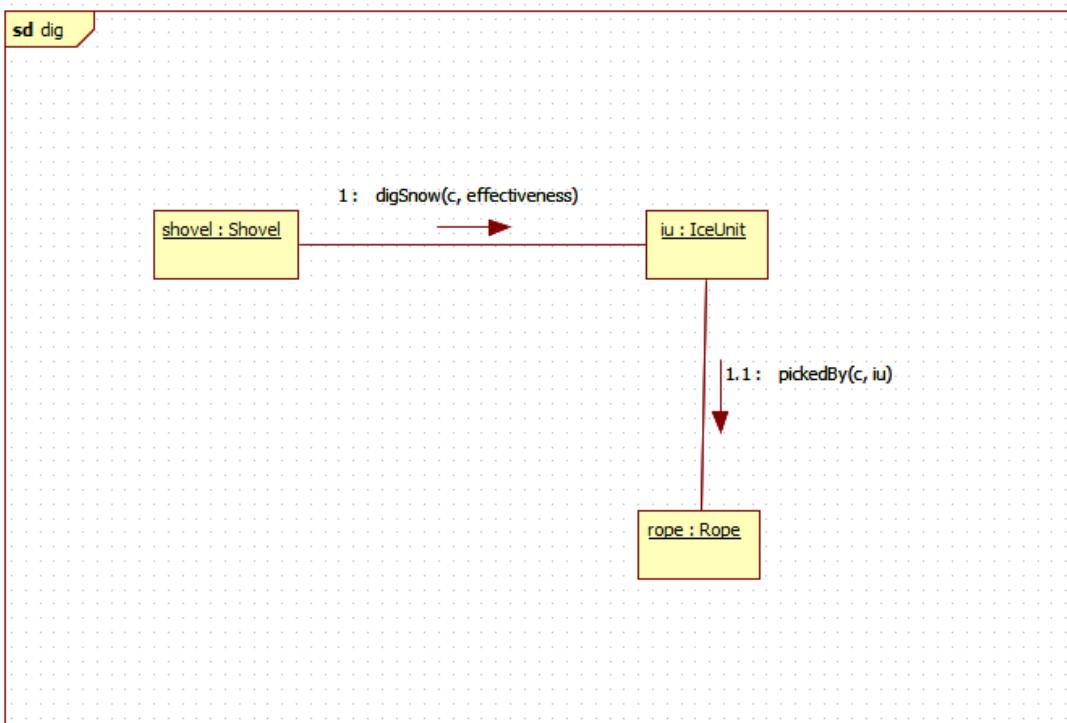


## 5.4 Kommunikációs diagramok

### 5.4.1 Eskimo executeCommands



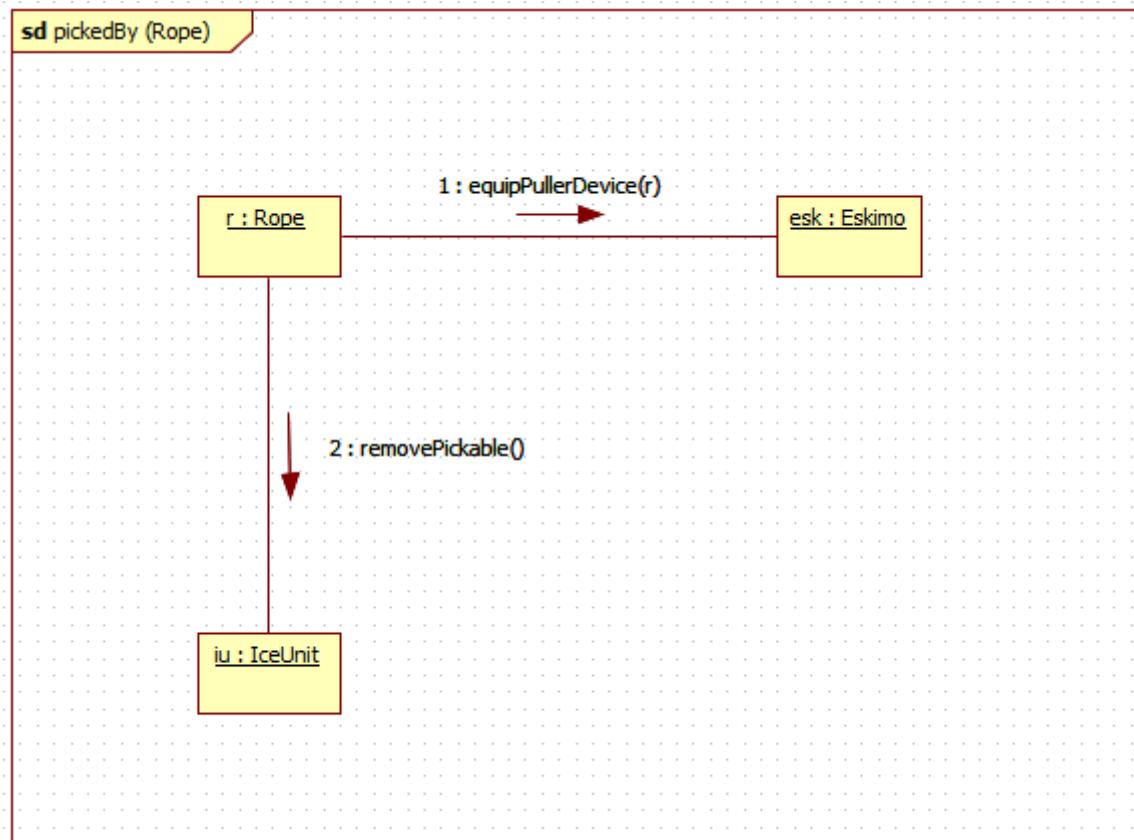
### 5.4.2 dig



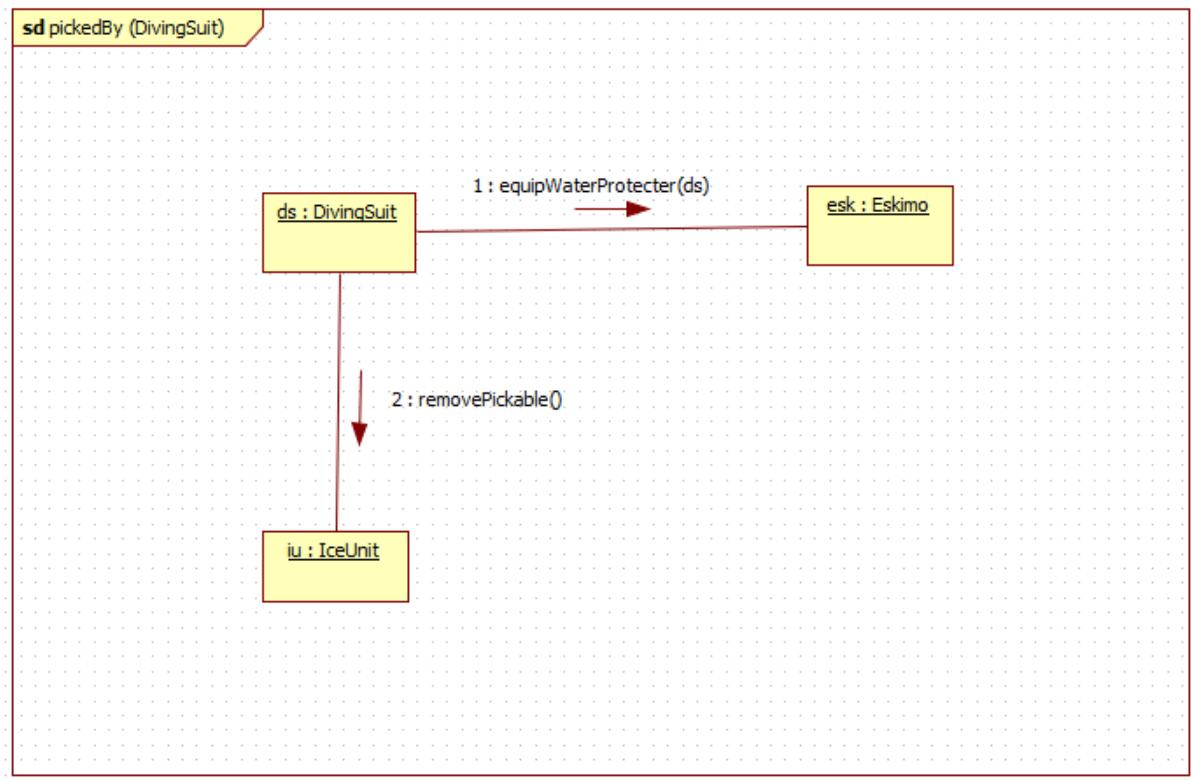
### 5.4.3 *pickedBy (Food)*



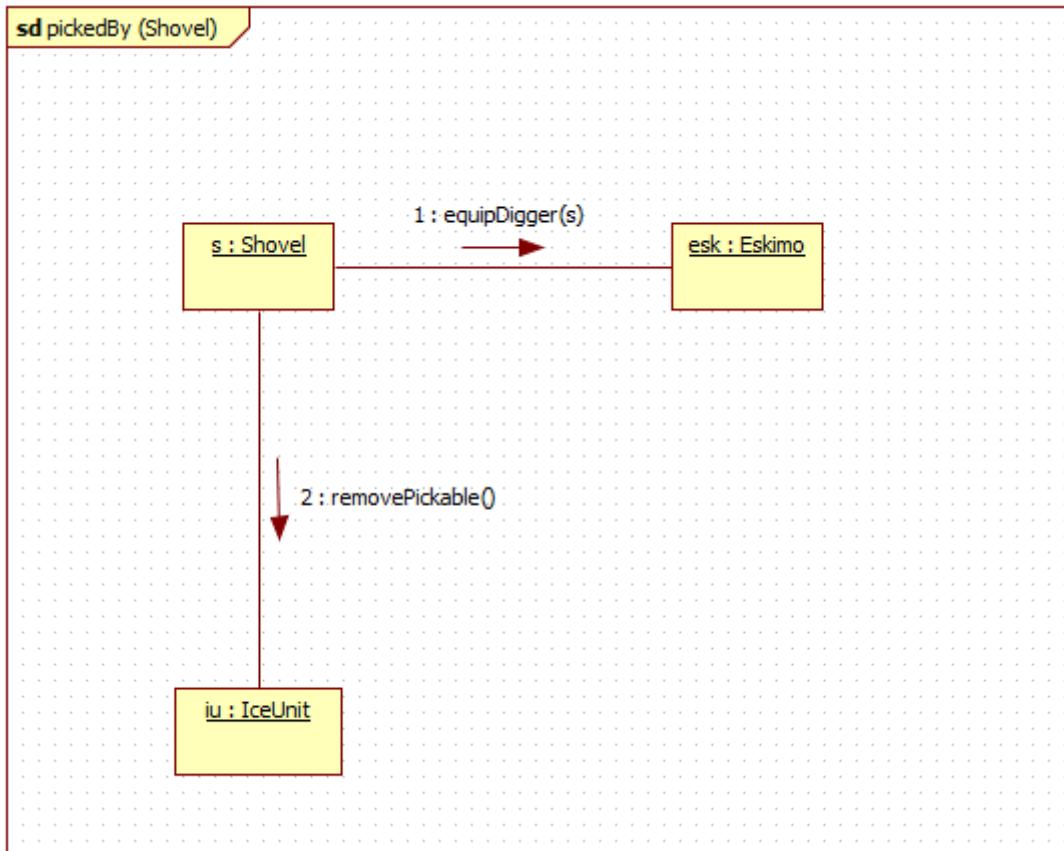
### 5.4.4 *pickedBy (Rope)*



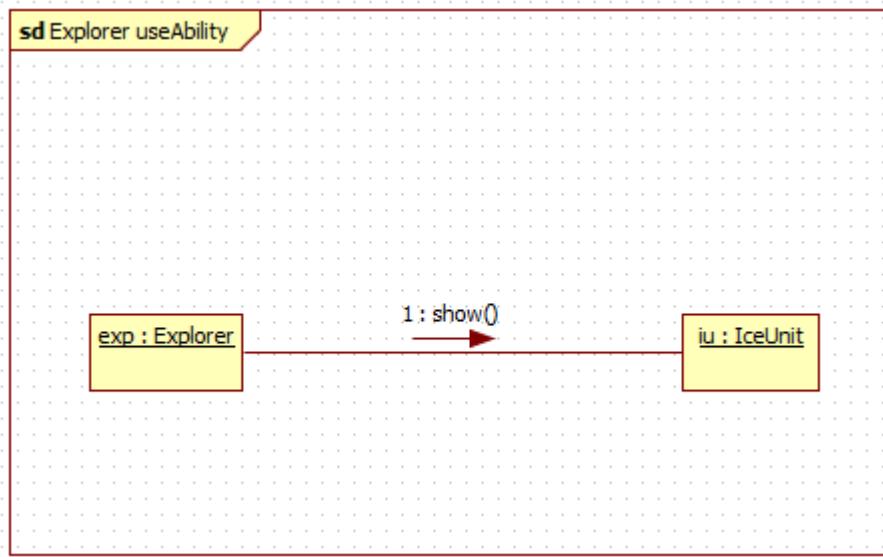
### 5.4.5 *pickedBy (DivingSuit)*



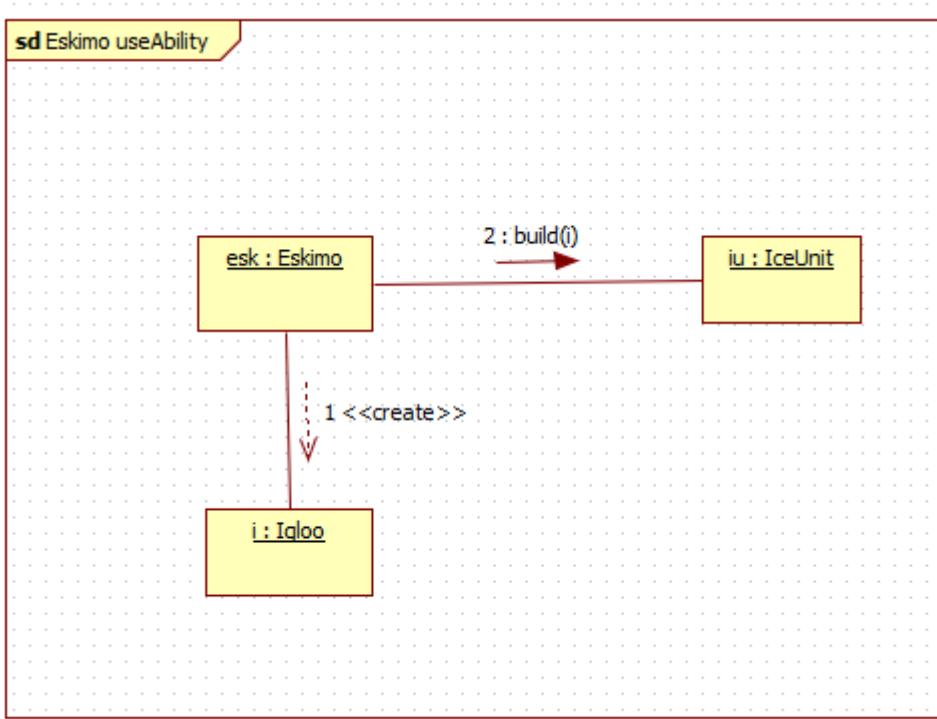
### 5.4.6 *pickedBy (Shovel)*



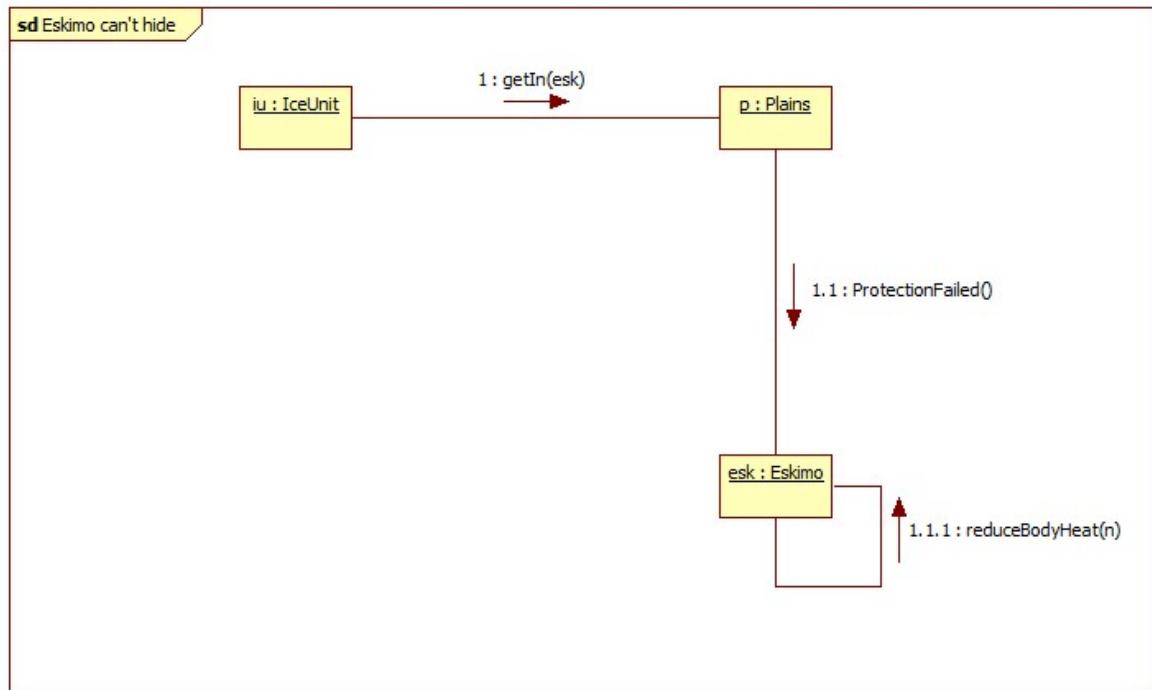
#### 5.4.7 Explorer useAbility



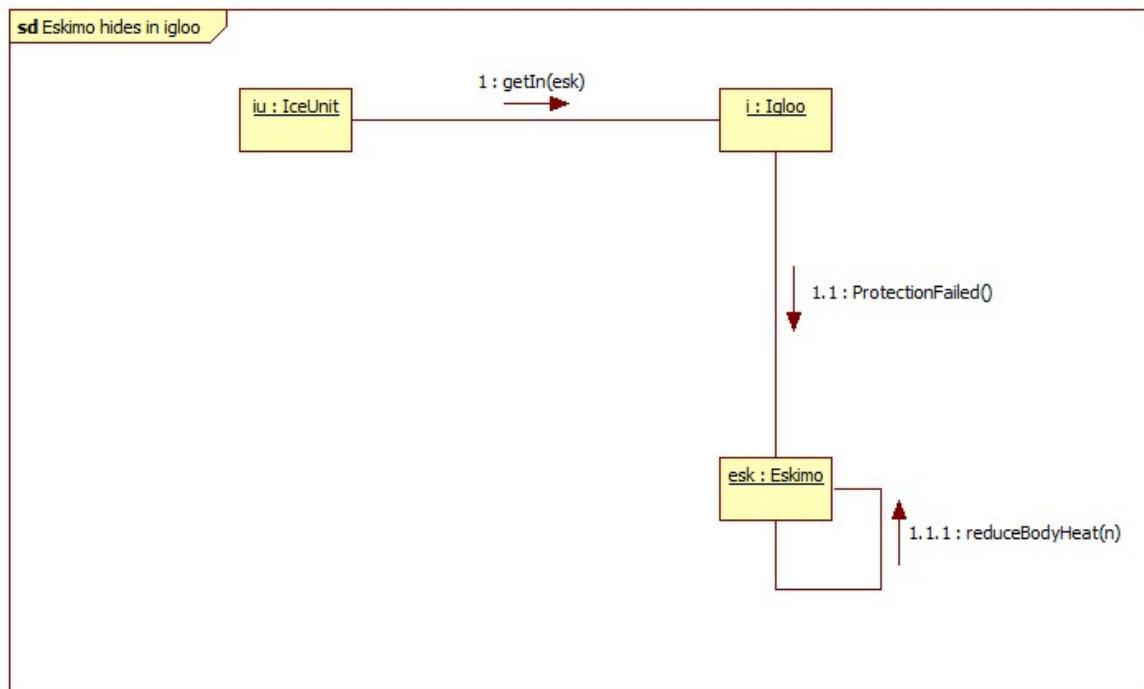
#### 5.4.8 Eskimo useAbility



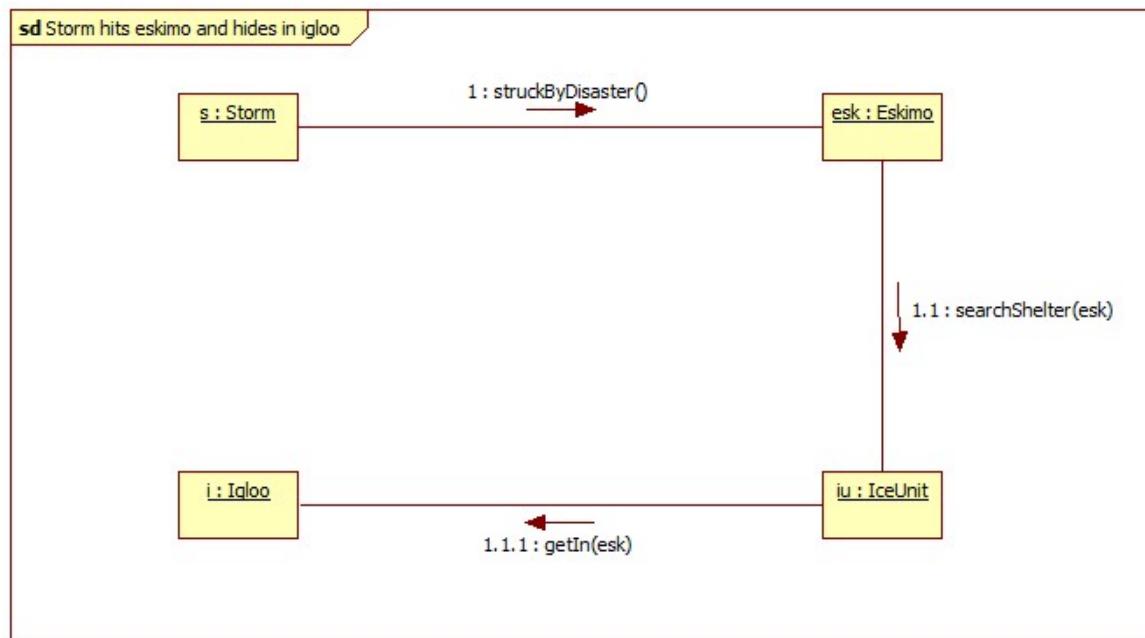
#### 5.4.9 *Eskimo can't hide*



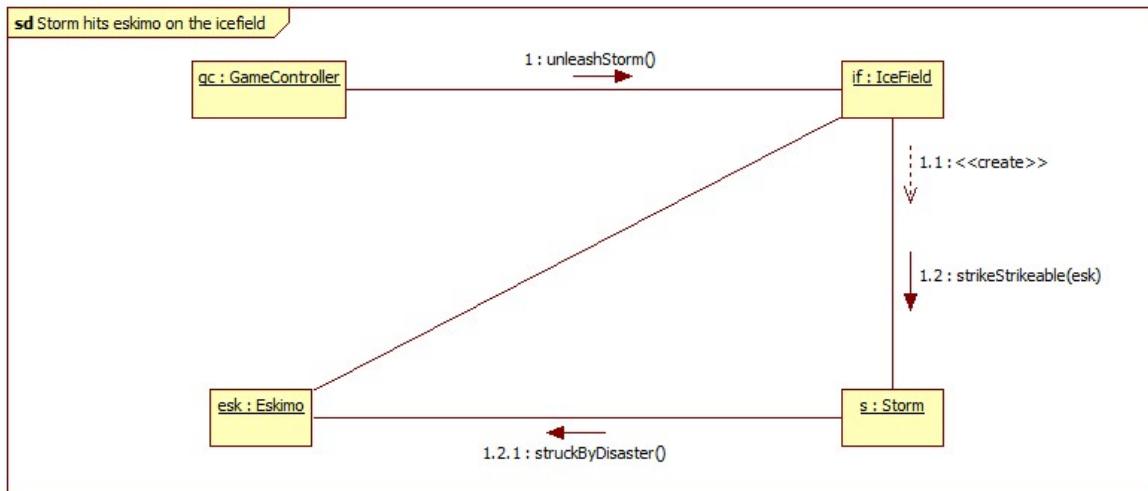
#### 5.4.10 *Eskimo hides in igloo*



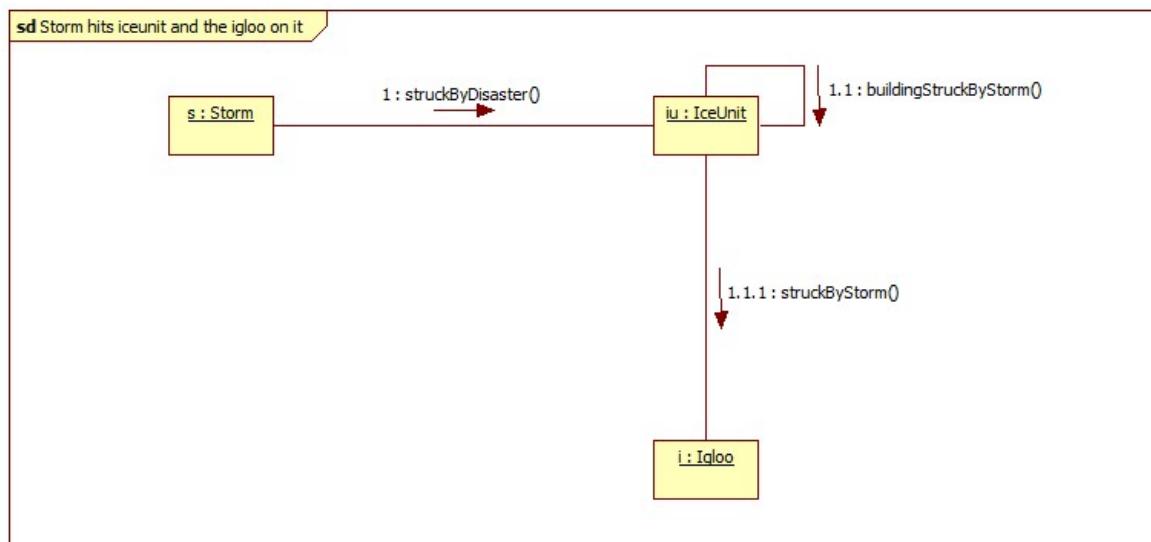
#### 5.4.11 Storm hits eskimo and hides in igloo



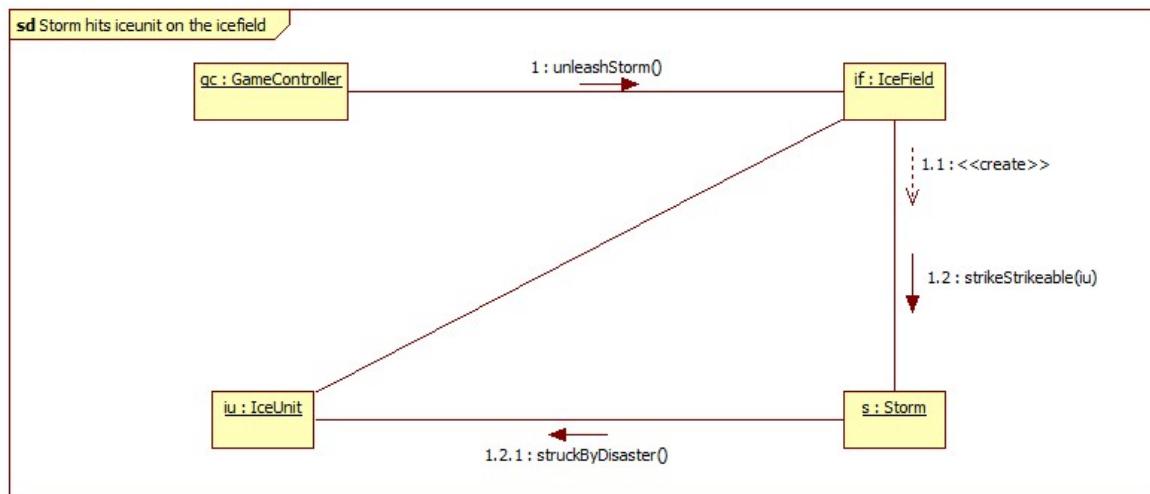
#### 5.4.12 Storm hits eskimo on the icefield



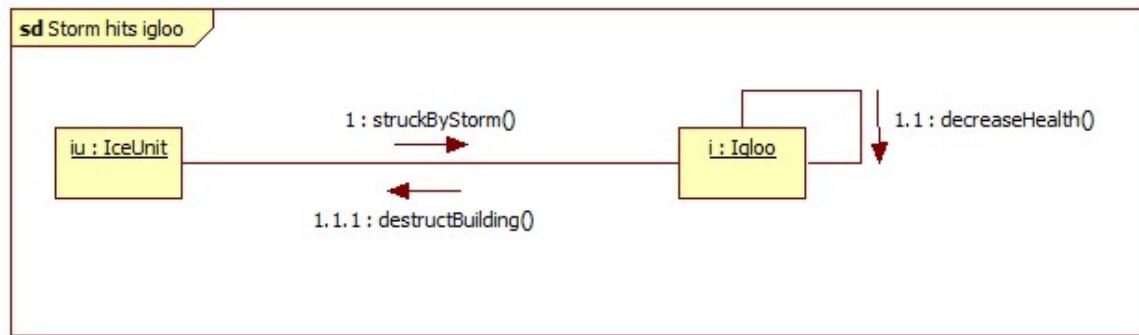
#### 5.4.13 Storm hits iceunit and the igloo on it



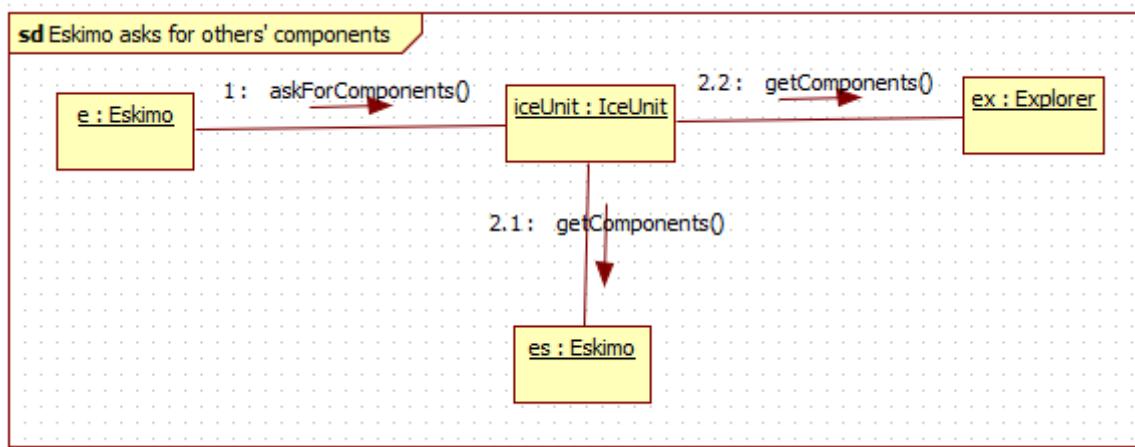
#### 5.4.14 Storm hits iceunit on the icefield



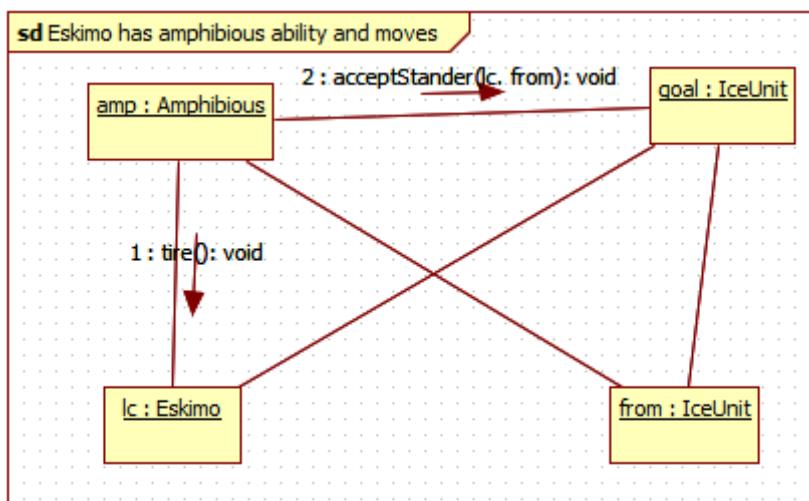
### 5.4.15 Storm hits igloo



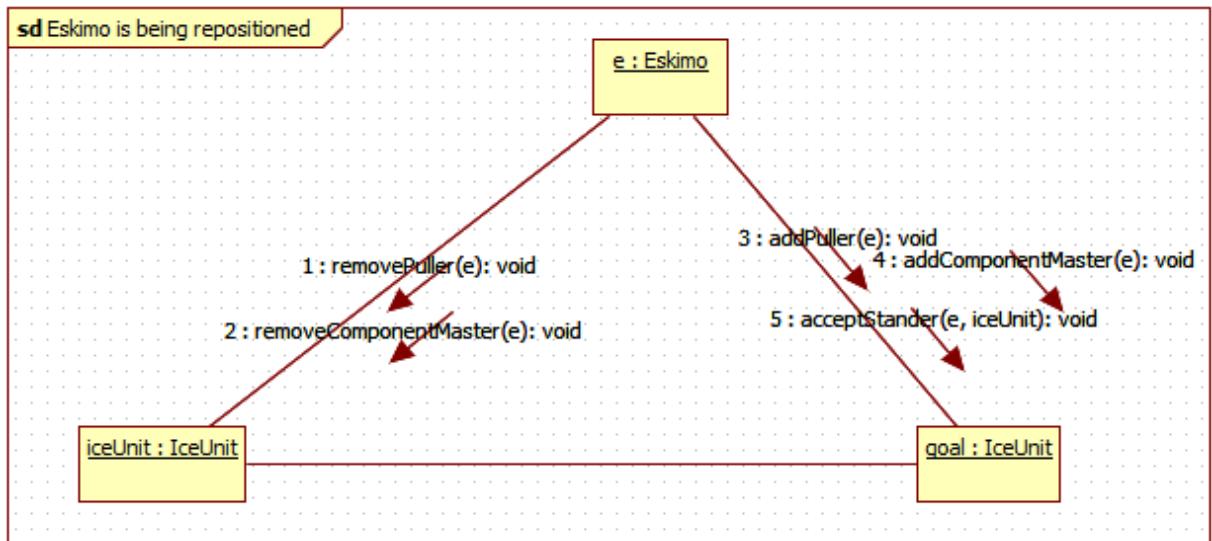
### 5.4.16 Eskimo asks for other's components



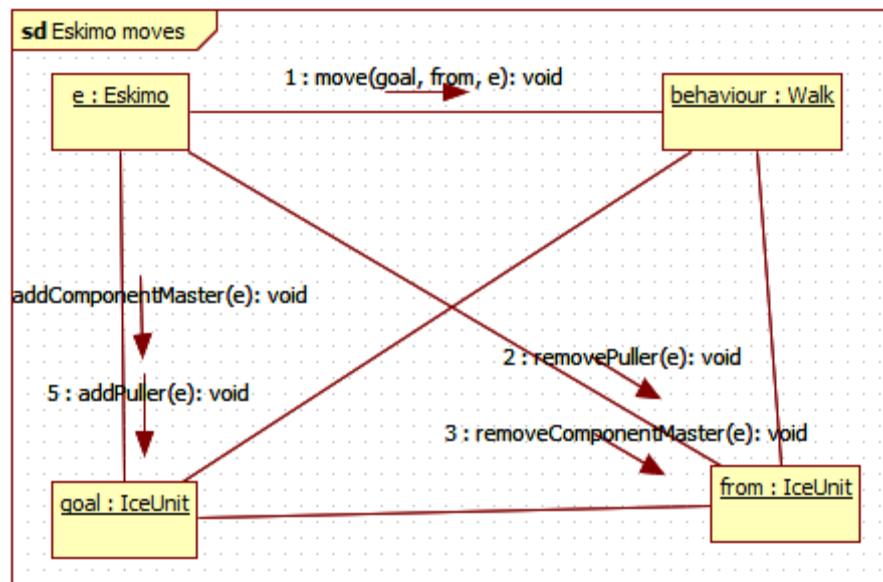
### 5.4.17 Eskimo has amphibious ability and moves



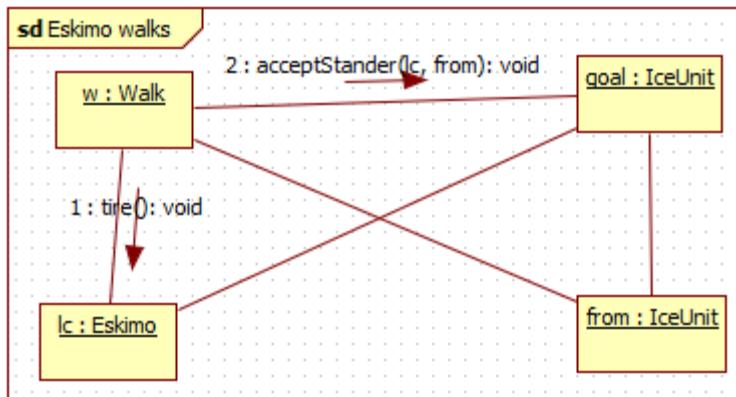
#### 5.4.18 Eskimo is being repositioned



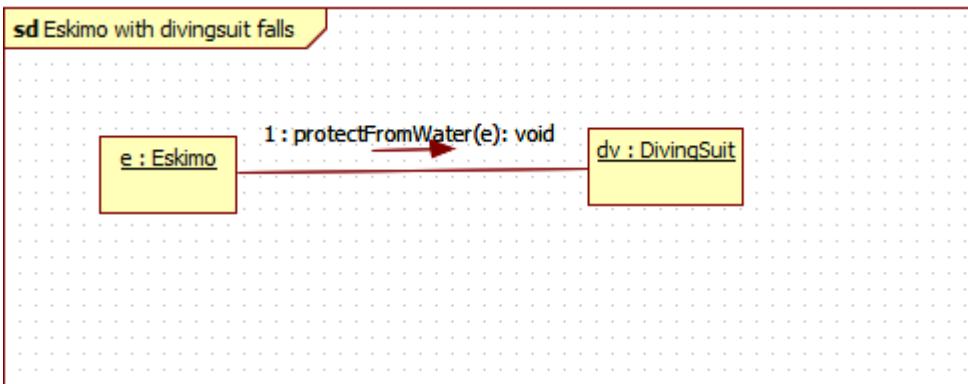
#### 5.4.19 Eskimo moves



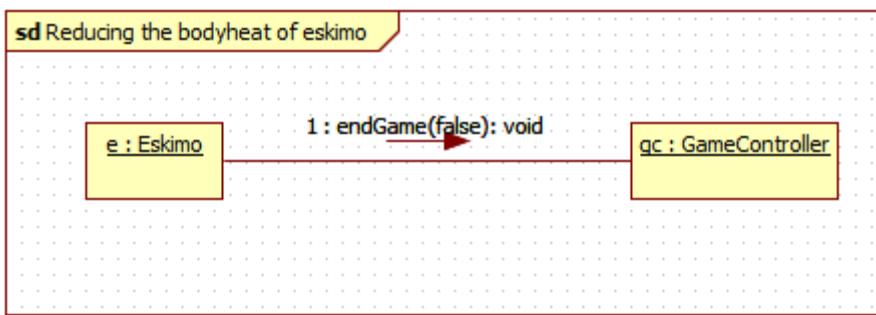
#### 5.4.20 Eskimo walks



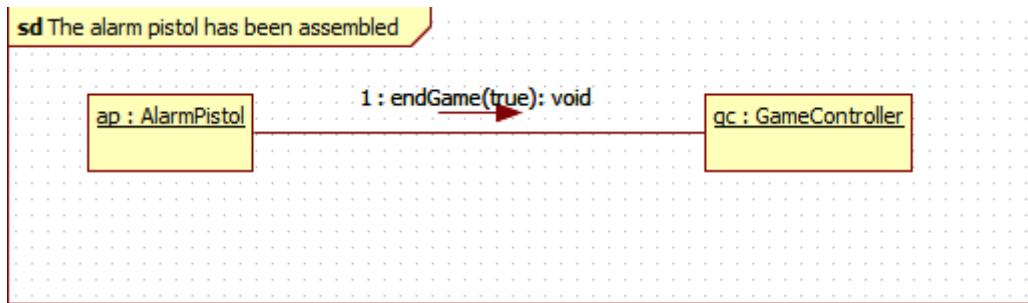
#### 5.4.21 Eskimo with divingsuit falls



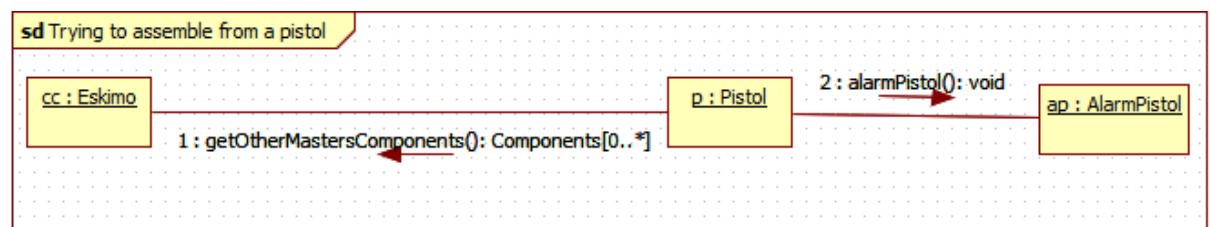
#### 5.4.22 Reducing the bodyheat of eskimo



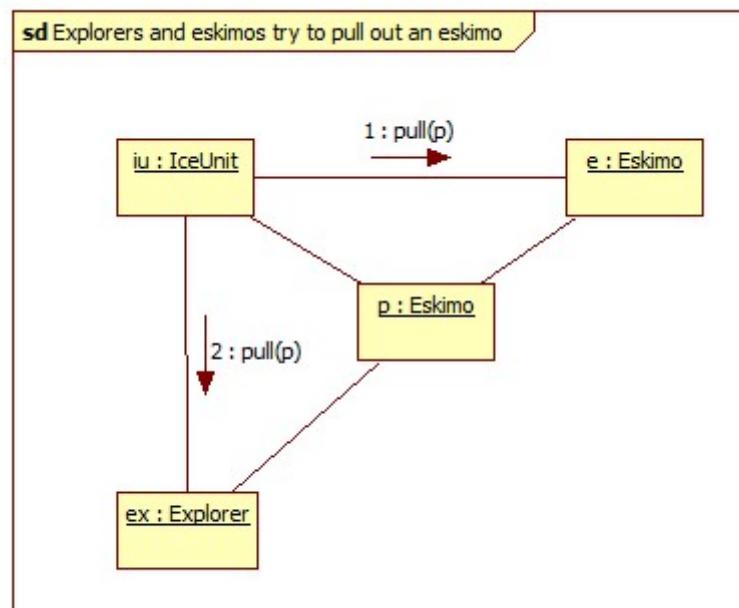
#### 5.4.23 The alarm pistol has been assembled



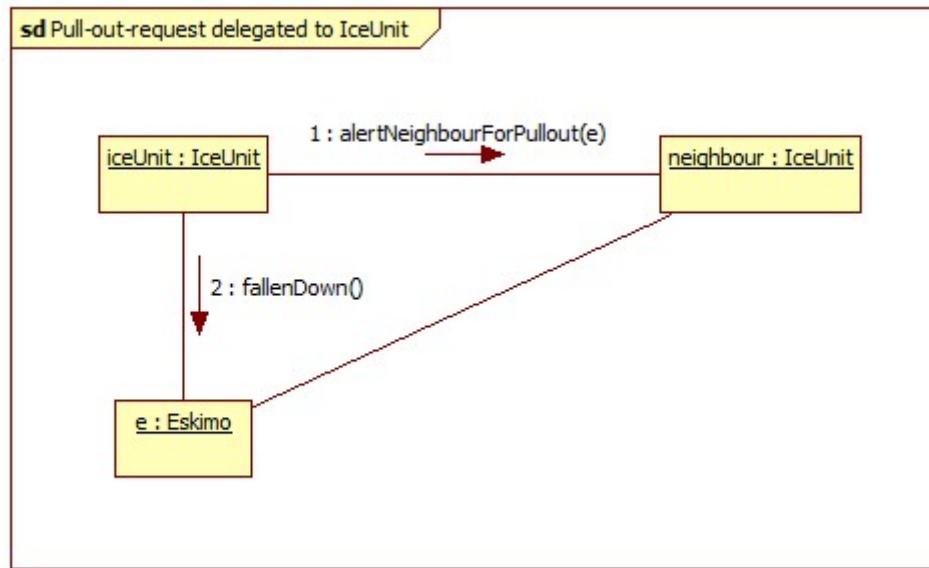
#### 5.4.24 Trying to assemble from a pistol



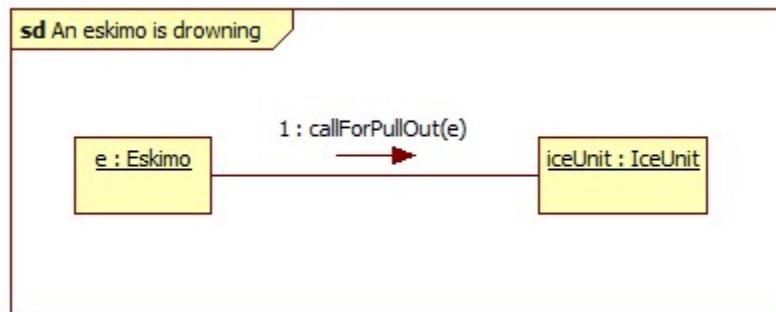
#### 5.4.25 alert\_neighbours



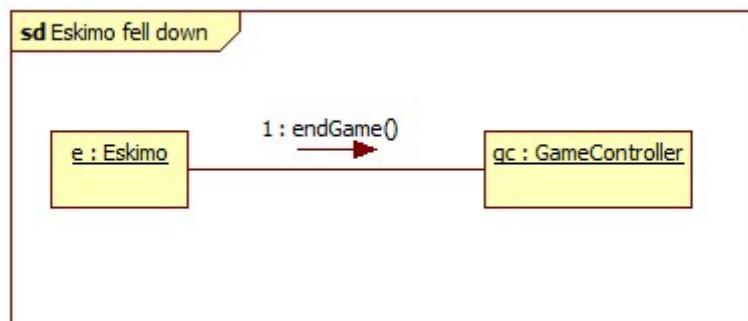
#### 5.4.26 *call\_for\_pullout*



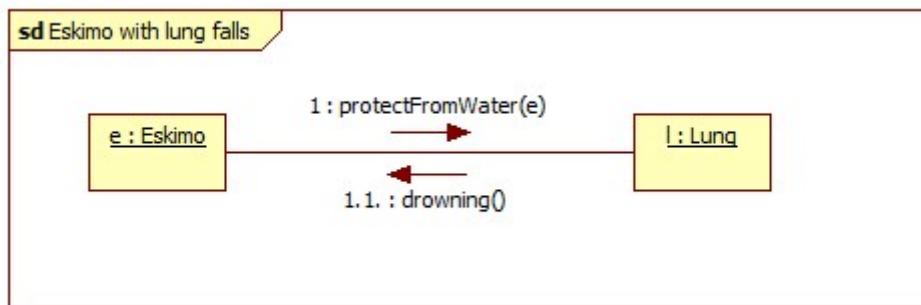
#### 5.4.27 *eskimo\_dorwning*



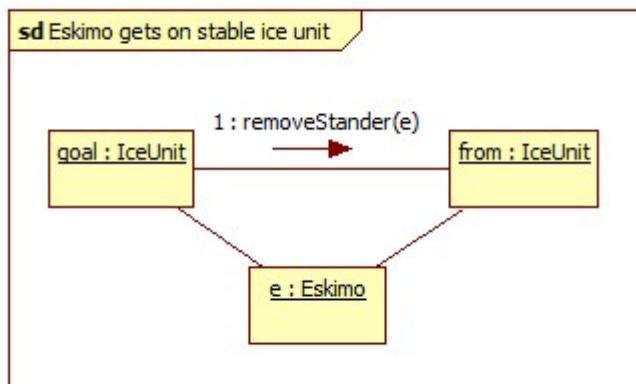
#### 5.4.28 *eskimo\_fallen\_down*



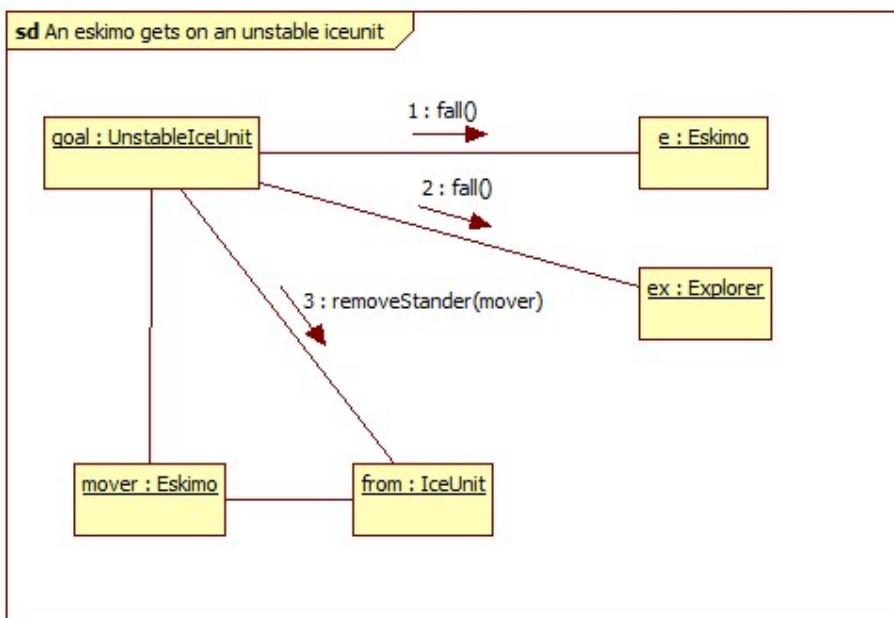
#### 5.4.29 eskimo\_falls\_lung



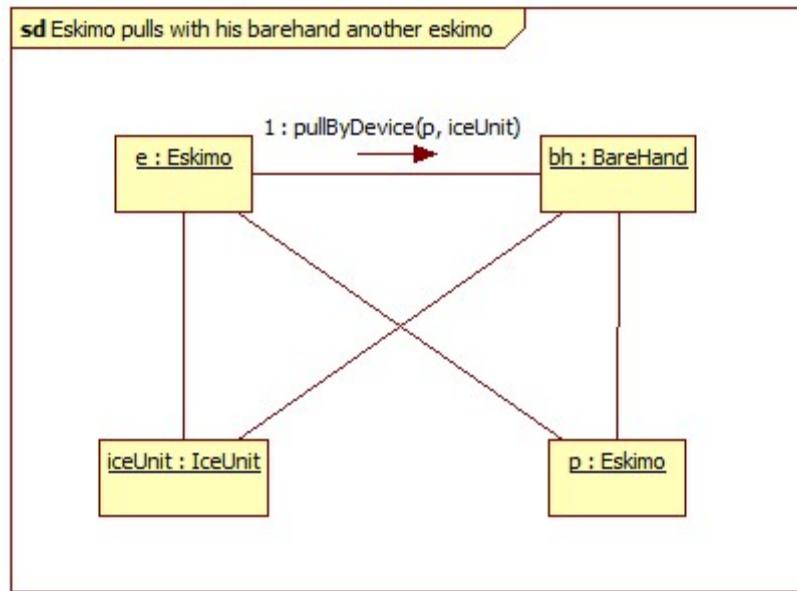
#### 5.4.30 eskimo\_getson\_stable



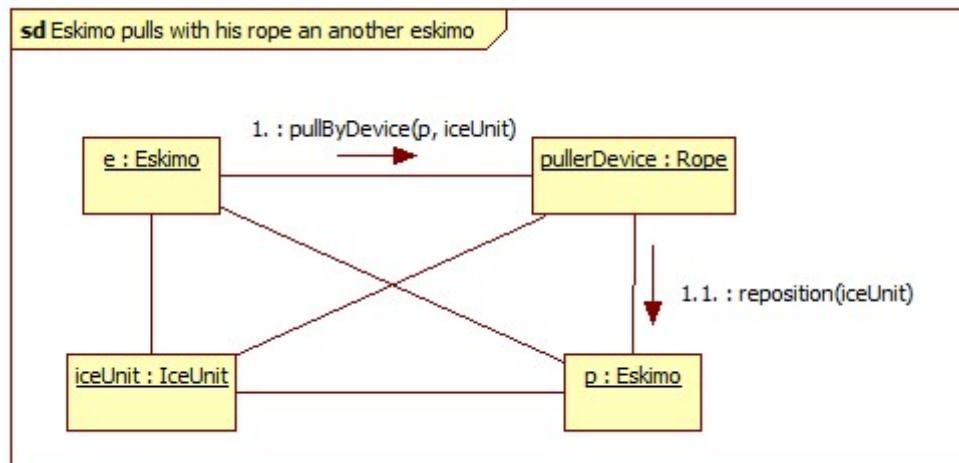
#### 5.4.31 eskimo\_getson\_unstable



### 5.4.32 eskimo\_pull\_bh



### 5.4.33 eskimo\_pull\_rope



## 5.5 Napló

Kezdet	Időtartam	Résznevők	Leírás
2020.03.18. 14:00	4 óra	Iuhos Kaposvári Labancz Sulyok Traxler	Értekezlet: Feladat értelmezése, feladatrészek kiosztása.
2020.03.18. 20:00	2 óra	Sulyok	Tevékenység: Use-case diagram tervezése.
2020.03.19. 9:00	3 óra	Labancz	Tevékenység: Use-case leírások készítése.
2020.03.19. 10:00	3 óra	Kaposvári	Tevékenység: Szekvenciadiagramok tervezése.
2020.03.19. 13:00	4 óra	Traxler	Tevékenység: Szekvenciadiagramok készítése.
2020.03.19. 14:00	2 óra	Labancz Sulyok	Értekezlet: Use-case javításának megbeszélése.
2020.03.19. 17:00	2 óra	Iuhos Kaposvári Traxler	Értekezlet: További szekvenciadiagramok pontos megbeszélése.
2020.03.19. 16:00	2 óra	Sulyok	Tevékenység: Use-case diagram módosítása.
2020.03.19. 18:00	2 óra	Labancz	Tevékenység: Use-case leírások javítása.
2020.03.20. 13:00	5 óra	Iuhos	Tevékenység: Szekvenciadiagramok szerkesztése, ellenőrzése.
2020.03.20. 14:00	3 óra	Labancz	Tevékenység: A szkeleton kezelői felületének tervének elkészítése.
2020.03.20. 16:00	3 óra	Traxler	Tevékenység: Kommunikációs diagramok tervezése.
2020.03.20. 20:00	3 óra	Sulyok	Tevékenység: Kommunikációs diagramok készítése.

2020.03.21. 14:00	4 óra	Kaposvári	Tevékenység: Kommunikációs diagramok készítése.
2020.03.21. 17:00	2 óra	Iuhos	Tevékenység: Kommunikációs diagramok készítése.
2020.03.22. 16:00	2 óra	Iuhos Kaposvári Sulyok Traxler	Értekezlet: Dokumentum ellenőrzése, hibajavítások elvégzése.
2020.03.22. 20:00	1 óra	Labancz	Tevékenység: Dokumentum formázása, ellenőrzése.

# **6. Feladat: Szkeleton - beadás és a forráskód herculesre való feltöltése**

**14 – the\_minesweepers**

Konzulens:  
**Fetter László**

## **Csapattagok**

Iuhos Zoltán	J8LOUL	zoltan.iuhos@gmail.com
Kaposvári Márk Martin	RAYVF8	xmarko45@gmail.com
Sulyok Botond	BM2BBM	botond.sulyok99@gmail.com
Labancz Tamás Dominik	A9HZBF	labancz.tamas@gmail.com
Traxler Bálint	IXNOYB	traxlerbalint@gmail.com

2020.03.30.

## 6. Szkeleton beadás

### 6.1 Fordítási és futtatási útmutató

#### 6.1.1 Fájllista

Fájl neve	Méret (B)	Keletkezés ideje	Tartalom
CommunicationPlatform.java	294	2020.03.25 15:03	Azon interfész váza, amelyik az egy mezőn keresztül történő kommunikációt lehetővé tévő függvényeket írja elő.
Pullable.java	589	2020.03.25 15:07	Húzható dolgokat kezelő absztrakt osztály váza.
Puller.java	434	2020.03.25 15:10	Azon interfész váza, amelyet megvalósító objektumok rendelkeznek az elhúzás képességével.
Stander.java	1 972	2020.03.25 15:14	Absztrakt osztály váza, leesni képes dolgok származnak le belőle.
Terrestrial.java	382	2020.03.25 15:21	Azon objektumok interfészének váza, amelyeknek káros, ha vízben vannak.
Amphibious.java	2 413	2020.03.25 15:24	Annak az osztálynak a váza, amelyik objektumai mozoghatnak jégtáblán és vízben is.
LocationChanger.java	3 109	2020.03.25 15:35	Azon absztrakt osztály váza, amely képes Pullable objektumokat húzni.
MoveBehaviour.java	590	2020.03.25 15:47	Azon interfész váza, amely megadja a szükséges függvényeket a különböző mozgásformákhoz.
Walk.java	2 346	2020.03.25 15:51	Ezen osztály objektumai mozoghatnak szárazföldön.
Eskimo.java	1 366	2020.03.25 15:58	Azon osztály váza, amelyik megvalósítja az eszkimó karaktertípust.
Explorer.java	1 386	2020.03.25 16:05	Azon osztály váza, amelyik megvalósítja a sarkutató karaktertípust.
Character.java	13 616	2020.03.25 16:11	Az összes irányítható karakter közös működését megvalósító absztrakt osztály.

ComponentMaster.java	232	2020.03.25 16:40	Interfész, az implementáló osztályok objektumai képesek komponenseket használni.
Equipper.java	518	2020.03.25 16:44	A tárgyakat felvenni képes objektumok interfésze.
FiniteActionDoer.java	500	2020.03.25 16:49	Korlátozott cselekvőképességgel rendelkező objektumok interfésze.
IceUnitManipulator.java	463	2020.03.25 16:52	Interfész, amely biztosítja a jégtáblákon való műveletvégzést.
Mortal.java	1 045	2020.03.25 16:55	Testhőt kezelő absztrakt osztály.
Refugee.java	257	2020.03.25 17:03	A játékban levő menekülteknek az interfésze.
Commandable.java	326	2020.03.25 17:07	Irányítási felületet biztosító interfész.
Disaster.java	355	2020.03.25 17:10	A játékban levő katasztrófáknek az interfésze.
GameController.java	4 638	2020.03.25 17:13	A játék indításáért, irányításáért, befejezéséért és az alapvető működéséért felelős osztály váza.
Storm.java	1 288	2020.03.25 17:38	A vihart reprezentáló osztály.
Strikeable.java	239	2020.03.25 17:49	A katasztrófák által megcsapható dolgok interfésze.
AlarmPistol.java	726	2020.03.25 17:52	Jelzőpisztolyt kezelő osztály.
Cartridge.java	1 581	2020.03.25 17:57	Jelzőpisztolyt alkotó patront kezelő osztály.
Component.java	1 916	2020.03.25 18:05	Komponenseket kezelő absztrakt osztály.
Light.java	1 556	2020.03.25 18:14	Jelzőpisztolyt alkotó jelzőfényt kezelő osztály.
Pistol.java	1 556	2020.03.25 18:20	Jelzőpisztolyt alkotó pisztolyt kezelő osztály.
Consumable.java	388	2020.03.25 18:23	Az elfogyasztható tárgyak interfésze.
Food.java	2 156	2020.03.25 18:26	Felvehető ételt reprezentáló osztály.
BareHand.java	483	2020.03.25 18:34	Az egyszerű ásást kezelő osztály váza.

Digger.java	1 426	2020.03.25 18:39	Az ásást kezelő absztrakt osztály.
Shovel.java	1 562	2020.03.25 18:43	Ásót kezelő osztály váza.
DivingSuit.java	2 235	2020.03.25 18:47	Búvárruhát kezelő osztály.
Lung.java	1 425	2020.03.25 18:56	Olyan osztály, amelyik lehetővé teszi a karakter megfulladását, ha nincs rajta védelem.
WaterProtecter.java	624	2020.03.25 19:06	Víz elleni védelmet kezelő interfész.
BasicPull.java	1 414	2020.03.25 19:10	Húzásra képtelenséget kezelő osztály.
PullerDevice.java	612	2020.03.25 19:16	A húzást megvalósító eszközök interfésze.
Rope.java	2 203	2020.03.25 19:20	Kötelet kezelő osztály.
Pickable.java	466	2020.03.25 19:26	Felvehető dolgok interfésze.
Logger.java	2 635	2020.03.28 13:49	A függvényhívások logolásáért felelős osztály.
testAssemble.java	1 515	2020.03.28 14:40	A jelzőpisztoly összeszerelését tesztelő osztály.
testBuildLogIglooAbility.java	830	2020.03.28 15:03	Eszkimó igluépítését tesztelő osztály.
testDigAndEquip.java	1 242	2020.03.28 15:30	Karakter ásását és tárgyfelvételét tesztelő osztály.
testEndTurn.java	726	2020.03.28 16:07	A karakter kör továbbadását tesztelő osztály.
testFallAndPull.java	1 793	2020.03.28 17:11	Esést és kihúzást tesztelő osztály.
testMovement.java	3 192	2020.03.29 14:16	Karakter mozgását tesztelő osztály.
testShowAbility.java	842	2020.03.29 14:20	Sarkkutató képességét tesztelő osztály.
testStartGame.java	496	2020.03.29 16:11	Játék elkezdését tesztelő osztály.
testStormOnIceField.java	1 310	2020.03.29 17:47	Vihart tesztelő osztály.
Building.java	1 613	2020.03.25 19:29	Az épületeket kezelő absztrakt osztály.
BuildingPlot.java	2 898	2020.03.25 19:36	A jégmezőn levő épülethelyeket kezelő absztrakt osztály.
DestructableBuilding.java	1 637	2020.03.25 19:45	Az elpusztítható épületeket kezelő absztrakt osztály.
Igloo.java	1 415	2020.03.25 19:50	Iglu típusú épületeket kezelő osztály.

Plains.java	1 221	2020.03.25 19:54	Olyan épülettípust kezelő osztály, ami nem rendelkezik semmilyen képességgel.
Digsite.java	419	2020.03.25 19:58	Ásható objektumok interfésze.
IceField.java	2 229	2020.03.25 20:01	A játékot felépítő egységeket kezelő osztály váza.
IceUnit.java	12 874	2020.03.25 20:07	Jégtáblát kezelő osztály.
StormyField.java	189	2020.03.25 20:51	A játékban levő viharos jégmezőknek az interfésze.
UnstableIceUnit.java	3 321	2020.03.25 20:57	Instabil jégtáblát kezelő osztály.
Main.java	2 924	2020.03.28 13:05	A tesztelés main osztályát tartalmazza, amely kezeli a teszteseteket.

### 6.1.2 Fordítás

1. Csomagoljuk ki az src.zip mappát.
2. Navigálunk a kicsomagolt src mappába.
3. Nyissunk egy terminált/parancssort az src mappához navigálva, majd hajtsuk végre a következő parancsokat:
  - 4a. Linux / MacOS alatt: `find -name "*.java" > sources.txt`
  - 4b. Windows alatt: `dir /s /B *.java > sources.txt`
5. `javac @sources.txt`

### 6.1.3 Futtatás

Az előző lépések után az src mappában maradva a terminál / parancssorban ki kell adni a következő parancsot: `java hu.minesweepers.Main`

## 6.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Iuhos Zoltán	J8LOUL	20 %
Kaposvári Márk Martin	RAYVF8	20 %
Sulyok Botond	BM2BBM	20 %
Labancz Tamás Dominik	A9HZBF	20 %
Traxler Bálint	IXNOYB	20 %

### 6.3 Napló

<b>Kezdet</b>	<b>Időtartam</b>	<b>Részttvevők</b>	<b>Leírás</b>
2020.03.25 15:00	6 óra	Iuhos	Tevékenység: Osztályok elkészítése
2020.03.27 13:00	6 óra	Sulyok	Tevékenység: Funkciók elkészítése
2020.03.27 18:00	1 óra	Sulyok	Tevékenység: Program kommentálása
2020.03.28 10:00	4 óra	Iuhos Kaposvári Sulyok Labancz Traxler	Értekezlet: Teszesetek kiosztása, megoldásuk módjának megbeszélése
2020.03.28 13:00	4 óra	Traxler	Tevékenység: Tesztek írása
2020.03.28 15:00	5 óra	Kaposvári	Tevékenység: Tesztek írása
2020.03.28 18:00	2 óra	Labancz	Tevékenység: Osztályok javítása
2020.03.29 10:00	2 óra	Labancz	Tevékenység: Kód kommentálása
2020.03.29 12:00	2 óra	Iuhos Kaposvári Sulyok Labancz Traxler	Értekezlet: Működés helyességének megbeszélése
2020.03.29 14:00	5 óra	Traxler	Tevékenység: Tesztek írása
2020.03.29 14:00	4 óra	Kaposvári	Tevékenység: Tesztek írása
2020.03.29 16:00	2 óra	Labancz	Tevékenység: Tesztek írása
2020.03.29 18:00	2 óra	Iuhos	Tevékenység: Hibák javítása
2020.03.29 21:00	1 óra	Iuhos	Tevékenység: Fordítás és futtatás módjának leírása
2020.03.29 18:00	3 óra	Labancz	Tevékenység: Ellenőrzés, kommentek írása
2020.03.30 8:00	2 óra	Sulyok	Tevékenység: Dokumentáció megírása

# **7. Feladat: Prototípus koncepciója - beadás**

**14 – the\_minesweepers**

Konzulens:  
**Fetter László**

## **Csapattagok**

Iuhos Zoltán

Kaposvári Márk Martin

Sulyok Botond

Labancz Tamás Dominik

Traxler Bálint

J8LOUL	zoltan.iuhos@gmail.com
RAYVF8	xmarko45@gmail.com
BM2BBM	botond.sulyok99@gmail.com
A9HZBF	labancz.tamas@gmail.com
IXNOYB	traxlerbalint@gmail.com

2020.04.06.

## 7. Prototípus koncepciója

### 7.0 Változás hatása a modellre

#### 7.0.1 Módosult osztálydiagram

Az olvashatóság érdekében a dokumentum végén helyeztük el.

#### 7.0.2 Új vagy megváltozó metódusok

##### 1) Collideable

**Új interfész, akik megvalósítják, azok tudnak találkozni egy másik objektummal.**

- **void HitBy(c: Character):** A metódus feladata, hogy levezényelje azt az eseményt, amikor egy az interfészt implementáló osztály találkozik egy másik objektummal (azaz ahol az interfészt megvalósító objektum állt, arra a helyre odakerült egy karakter), jelen esetben akivel találkozik az egy Character típusú objektum.

**void HitBy(ib: IceBear):** A metódus feladata, hogy levezényelje azt az eseményt, amikor egy az interfészt implementáló osztály találkozik egy másik objektummal (azaz ahol az interfészt megvalósító objektum állt, arra a helyre odakerült egy jegesmedve), jelen esetben akivel találkozik az egy IceBear típusú objektum.

##### 2) IceBear

**Új osztály, a jegesmedvét reprezentálja, aki megtámadhatja a karaktereket.**

- **void HitBy(c: Character):** A metódus feladata, hogy levezényelje azt az eseményt, amikor egy jegesmedve összetalálkozik egy karakterrel, azaz arra az egységre amin a jegesmedve áll odamegy egy Character típusú objektum. Ekkor a jegesmedve lehetőséget kap arra, hogy megtámadja a karaktert.

**void HitBy(ib: IceBear):** A metódus feladata, hogy levezényelje azt az eseményt, amikor egy jegesmedve összetalálkozik egy másik jegesmedvével, jelen esetben ekkor nem történik semmi.

##### 3) Character

**Bővült absztrakt osztály.**

- **void HitBy(c: Character):** A metódus feladata, hogy levezényelje azt az eseményt, amikor egy karakter összetalálkozik egy másik karakterrel.
- **void HitBy(ib: IceBear):** A metódus feladata, hogy levezényelje azt az eseményt, amikor egy karakter összetalálkozik egy jegesmedvével, azaz arra az egységre amin a karakter áll odamegy egy jegesmedve típusú objektum. Ekkor a jegesmedve lehetőséget kap arra, hogy megtámadja a karaktert. Természetesen a karakter elkezd menedéket keresni.
- **void removeTent():** Eltávolítja a karaktertől az általa birtokolt sárat.

#### 4) IceUnit

##### Módosult osztály.

- **void accept(c: Character, from: IceUnit):** Hozzáadja az eltárolt mezőn tartózkodó objektumokhoz az c paramétert, értesíti a from nevű paraméter mezőt, hogy elhagyta őt az c paraméter. Továbbá értesít minden az IceUniton tartózkodó Standert, hogy találkozik az éppen az IceUnitra került objektummal.
- **void remove(c: Character):** Ezzel a metódussal lehet értesíteni az IceUnitot, hogy a rajta levő Character típusú objektum elhagyta őt és ezáltal az IceUniton levő épületet is, ha abban tartózkodott.
- **void accept(ib: IceBear, from: IceUnit):** Hozzáadja az eltárolt mezőn tartózkodó objektumokhoz az ib paramétert, értesíti a from nevű paraméter mezőt, hogy elhagyta őt az ib paraméter. Továbbá értesít minden az IceUniton tartózkodó Standert, hogy találkozik az éppen az IceUnitra került objektummal.
- **void remove(ib: IceBear):** Ezzel a metódussal lehet értesíteni az IceUnitot, hogy a rajta levő IceBear típusú objektum elhagyta őt.
- **IceUnit[0..\*] getNeighbours():** Visszaadja az IceUnit objektummal szomszédos jégegységeket.

#### 5) UnstableIceUnit

##### Módosult osztály.

- **void accept(c: Character, from: IceUnit):** Hozzáadja az eltárolt mezőn tartózkodó objektumokhoz az c paramétert, értesíti a from nevű paraméter mezőt, hogy elhagyta őt az c paraméter. Továbbá értesít minden az UnstableIceUniton tartózkodó Standert, hogy találkozik az éppen az IceUnitra került objektummal. Továbbá dönt a jégtábla beszakadásának sorsáról a terhelés és kapacitás függvényében.
- **void accept(ib: IceBear, from: IceUnit):** Hozzáadja az eltárolt mezőn tartózkodó objektumokhoz az ib paramétert, értesíti a from nevű paraméter mezőt, hogy elhagyta őt az ib paraméter. Továbbá értesít minden az UnstableIceUniton tartózkodó Standert, hogy találkozik az éppen az UnstableIceUniton került objektummal. Továbbá dönt a jégtábla beszakadásának sorsáról a terhelés és kapacitás függvényében.

#### 6) BuildingPlot

##### Bővült absztrakt osztály.

- **void searchShelterFromIceBear(r: Refugee):** Az épülethelyen a paraméterben kapott menekült keres menedéket egy épületben, egy jegesmedve elől menekülve.

#### 7) Building

##### Bővült absztrakt osztály.

- **void getInFromIceBear(r: Refugee):** Bemegy az épületbe a paraméterben kapott jegesmedve elől menekülő objektum, ha tele van az épület, akkor nem tud bemenni és a

jegesmedve meg tudja támadni.

- **void leave(r: Refugee):** A menekült elhagyja az épületet.

## 8) Tent

**Új osztály, a sátor reprezentálja, ami megvédi a benne tartózkodókat a vihartól, de medvetámadástól viszont már nem.**

- **void getInFromIceBear(r: Refugee):** Bemegy a sátorba a paraméterben kapott jegesmedve elől menekülő objektum, azonban ez semmilyen fajta védelmet nem nyújt neki a medvetámadás elől.
- **void buildTent(c: Character, b: BuildingPlot):** Felépíti a sárat a paraméterként kapott építőhelyre és a karaktertől elveszi a sátorral rendelkezés jogát.
- **void executeCommand():** Végrehajtja a megfelelő parancsot, ez minden körben egyszer következik be a sátor esetében és ez azt eredményezi, hogy a sátor egy kör után eltűnik, eltávolítva magát az épülethelyéről.

## 9) Refugee

**Bővült interfész.**

- **void protectionFromBearFailed():** Ha a menekültnek nincsen fedezéke a jegesmedve ellen, akkor nincsen neki protekciója és meghalhat.

## 10) Equipper

**Bővült interfész.**

- **void equipTent(t: Tent):** Előírja a felvehető sátorral rendelkezés jogának megszerzését.

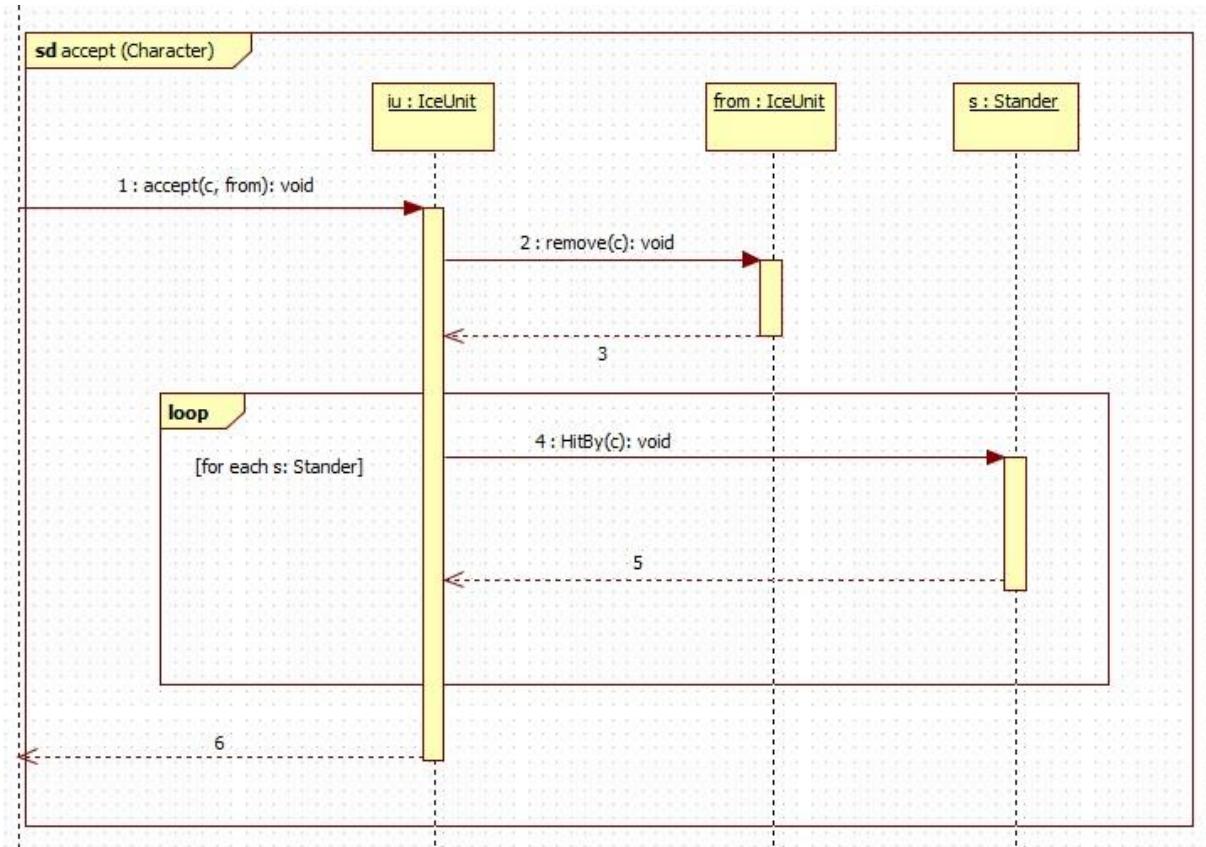
## 11) BreakableShovel

**Új osztály, a törhető ásó ugyanúgy működik, mint a hagyományos ásó, viszont ez el tud törni egy bizonyos számú használat után.**

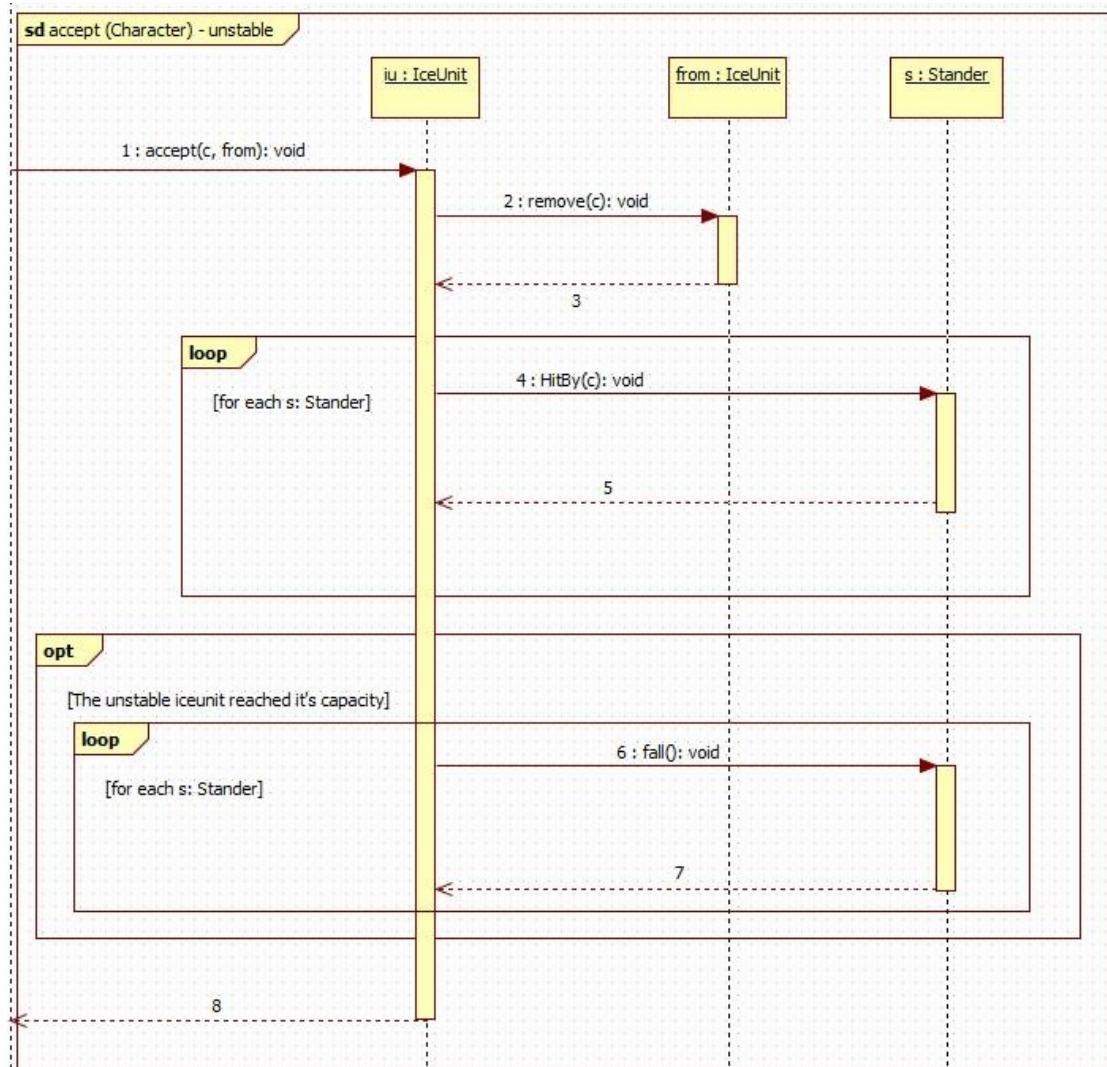
- **void dig(Character c, IceUnit iu):** Az ásást levezénylest végzi, a paraméterként megkapott IceUnit-nak meghívja a megfelelő metódusát, amivel az IceUnit ezt az interakciót kezelní tudja. Ha elfogyott az ásónak a tartóssága, akkor eltörök és az őt birtokló karakter is elveszti és a keze lesz az „ásóeszköze”.

### 7.0.3 Szekvencia-diagramok

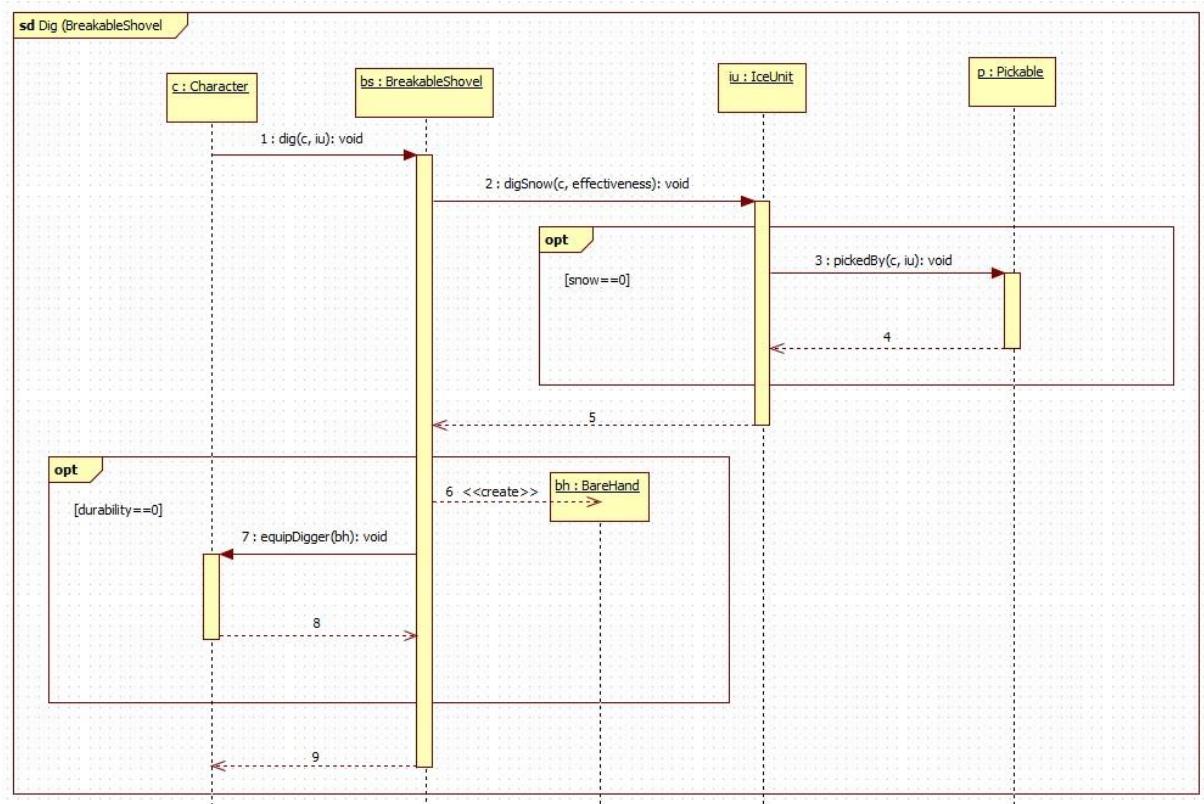
#### 1) accept (Character)



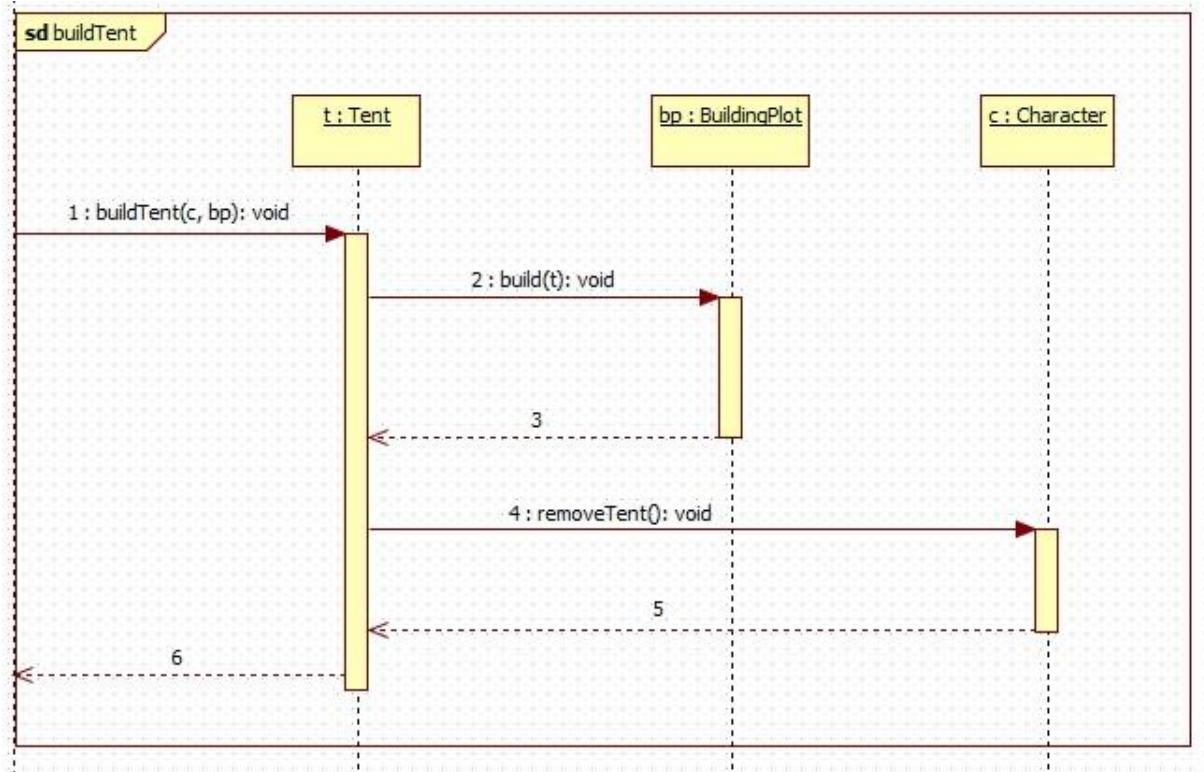
## 2) accept (Character)-unstable



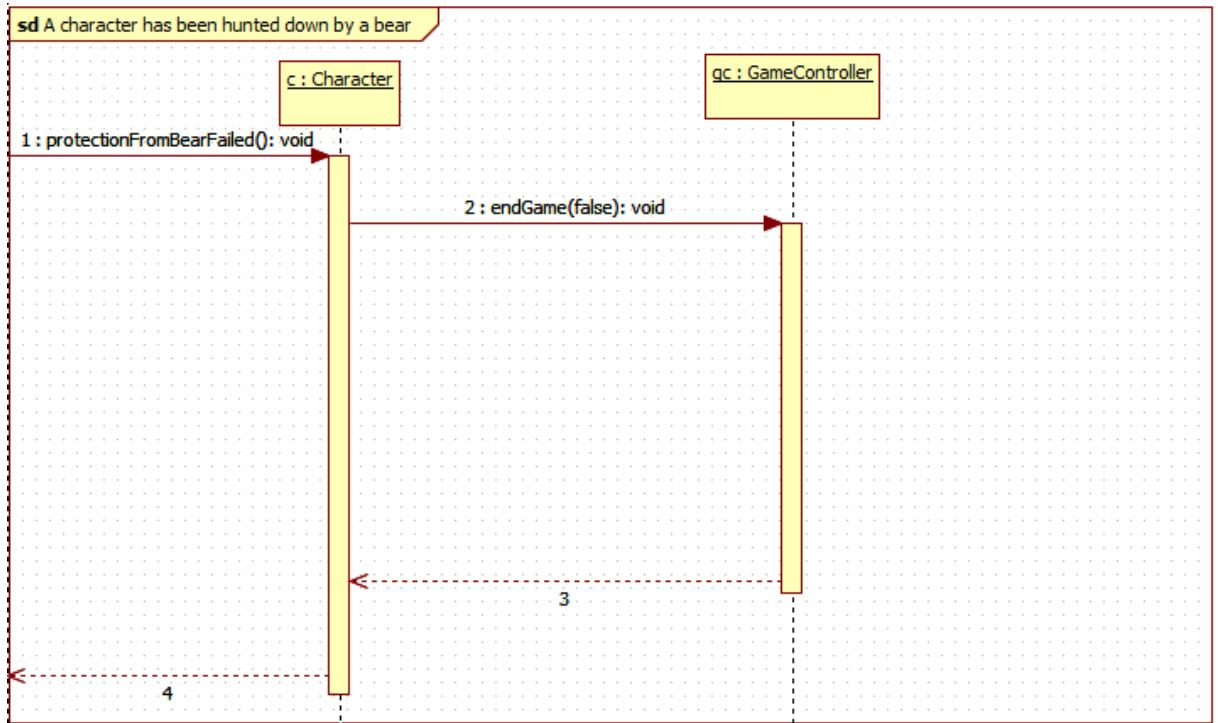
### 3) BreakableShovel



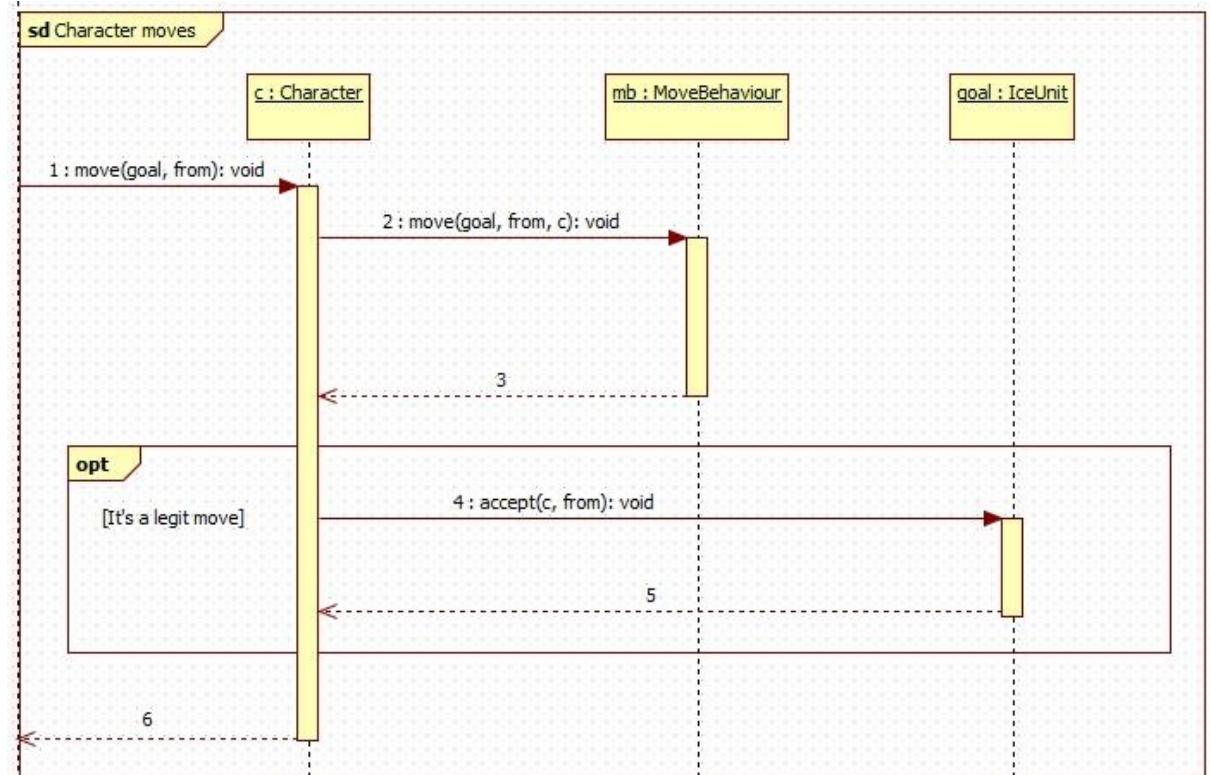
### 4) buildTent2



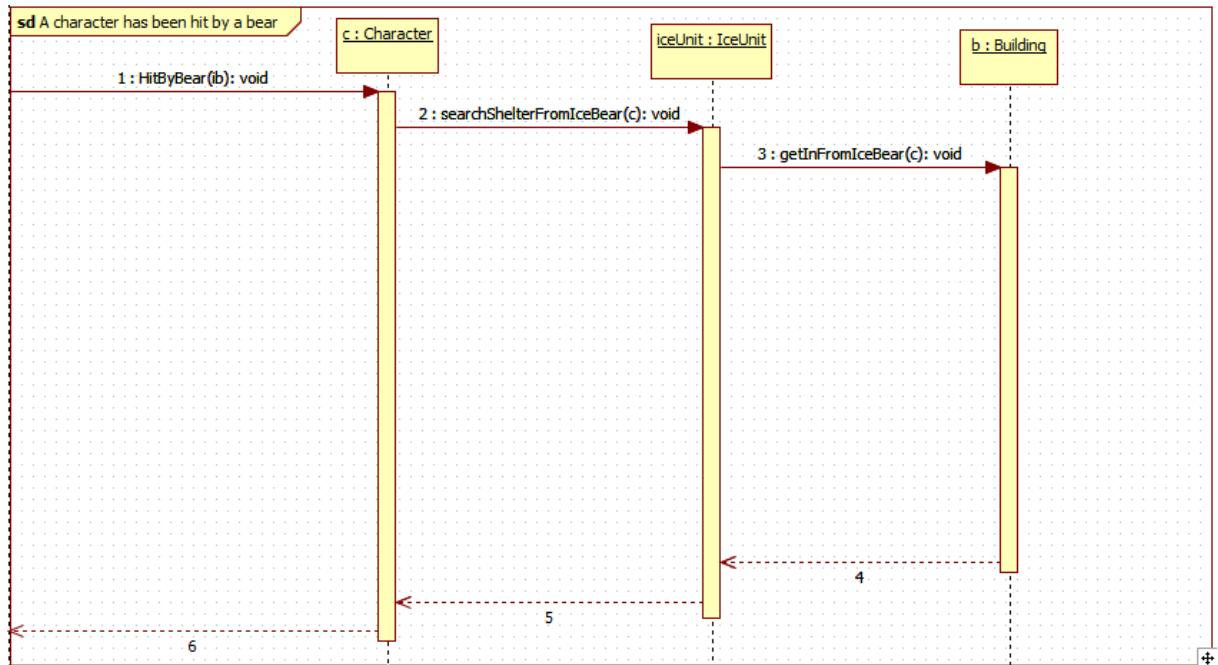
## 5) char\_protbearfail



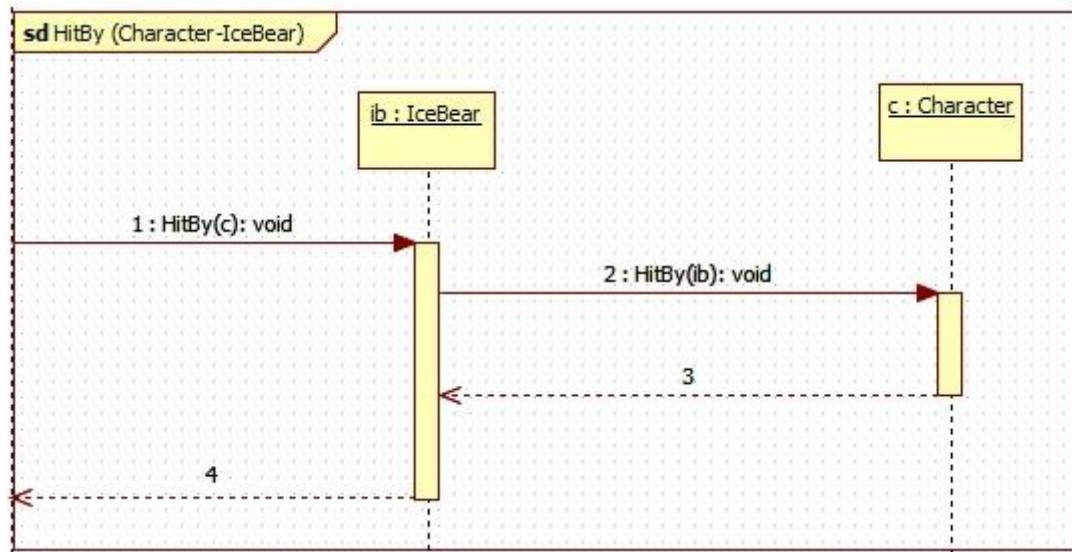
## 6) Character moves



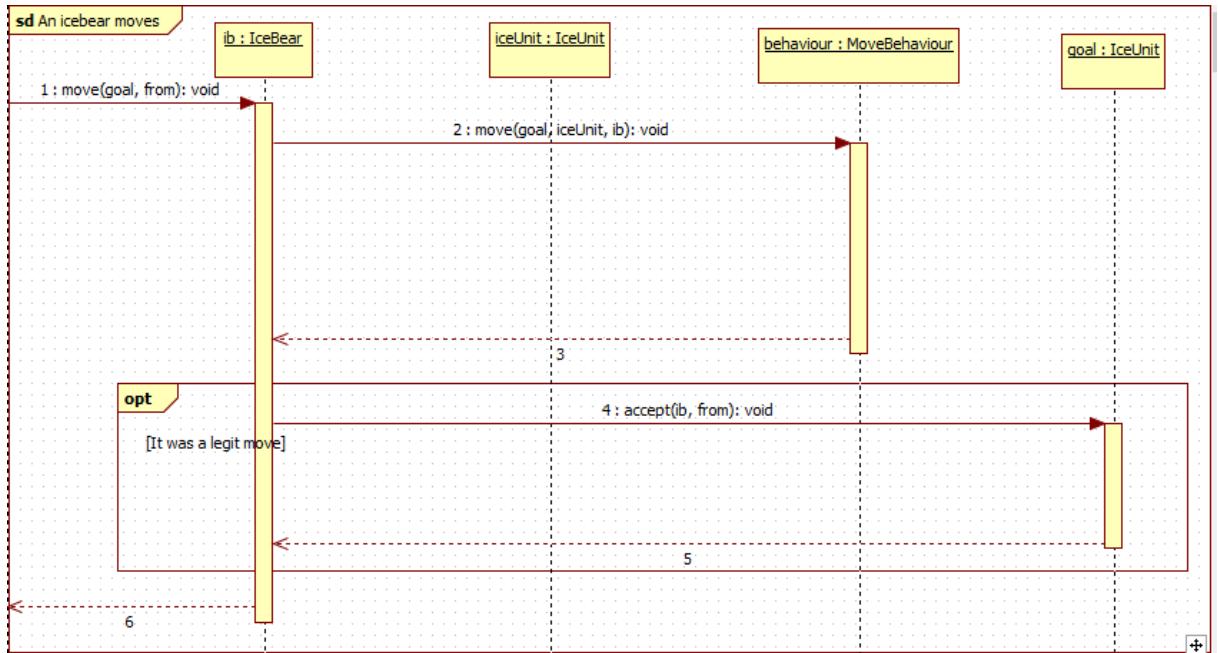
### 7) character\_hitby\_bear



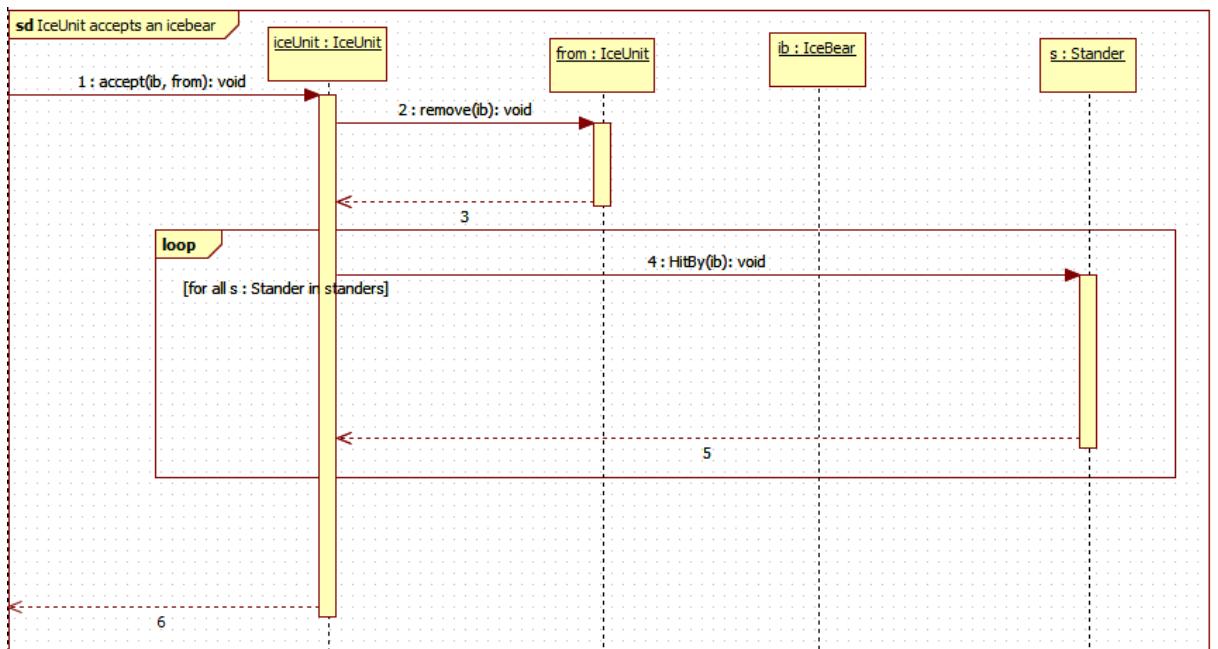
### 8) HitBy (Character-IceBear)



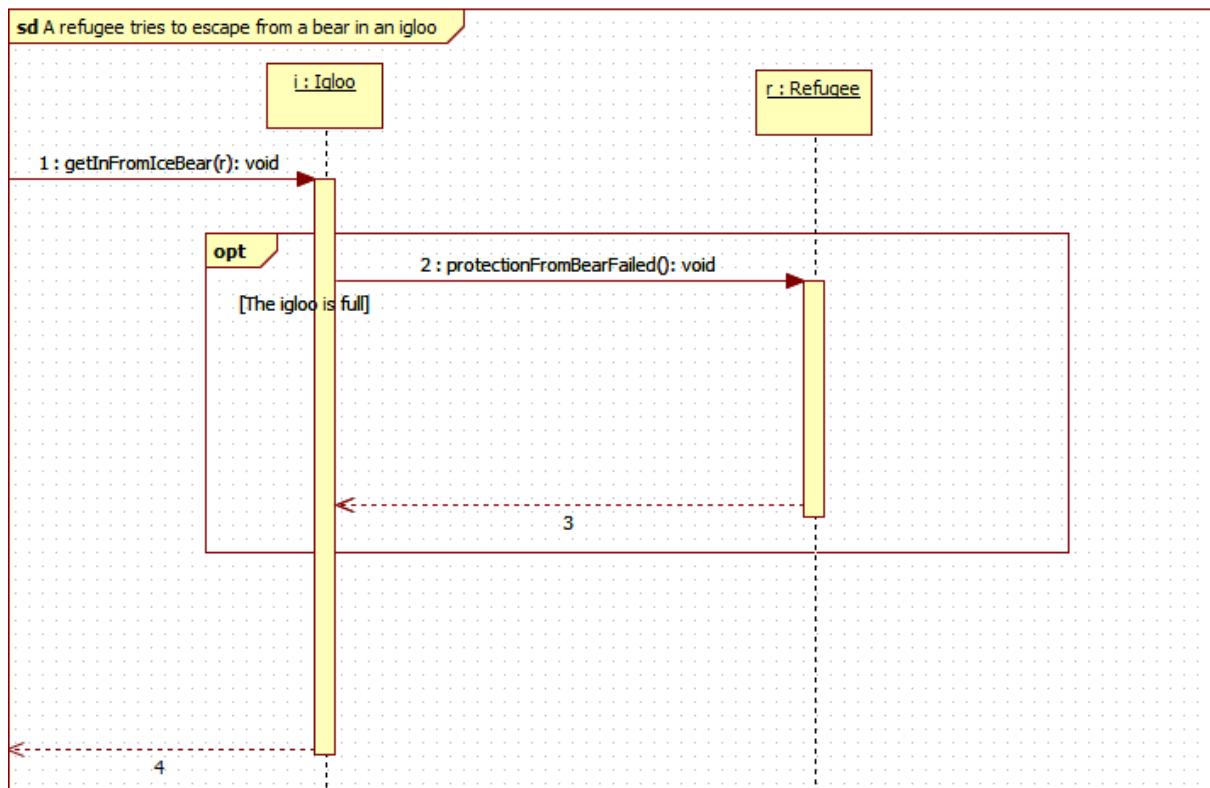
## 9) icebear\_moves



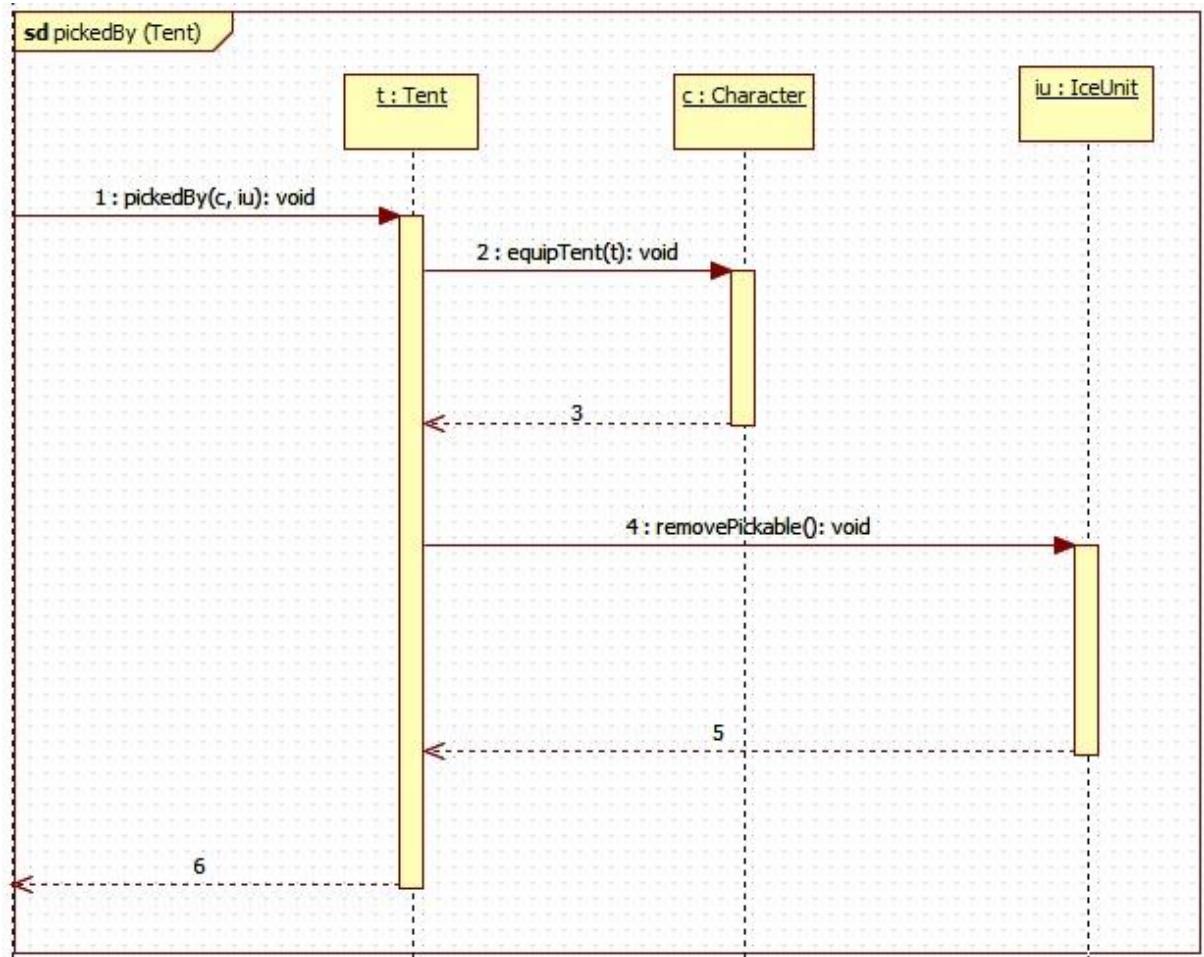
## 10) iceunit\_accepts\_bear



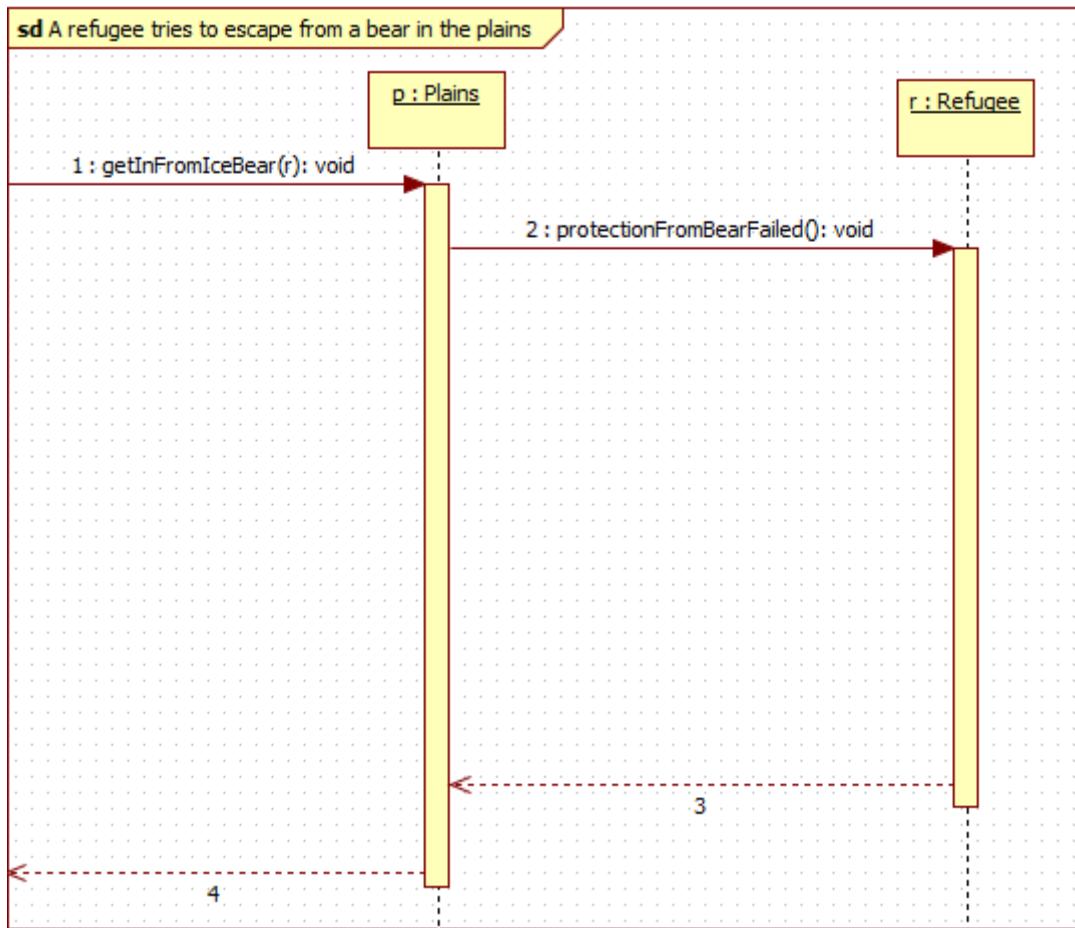
## 11) igloo\_getin\_bear



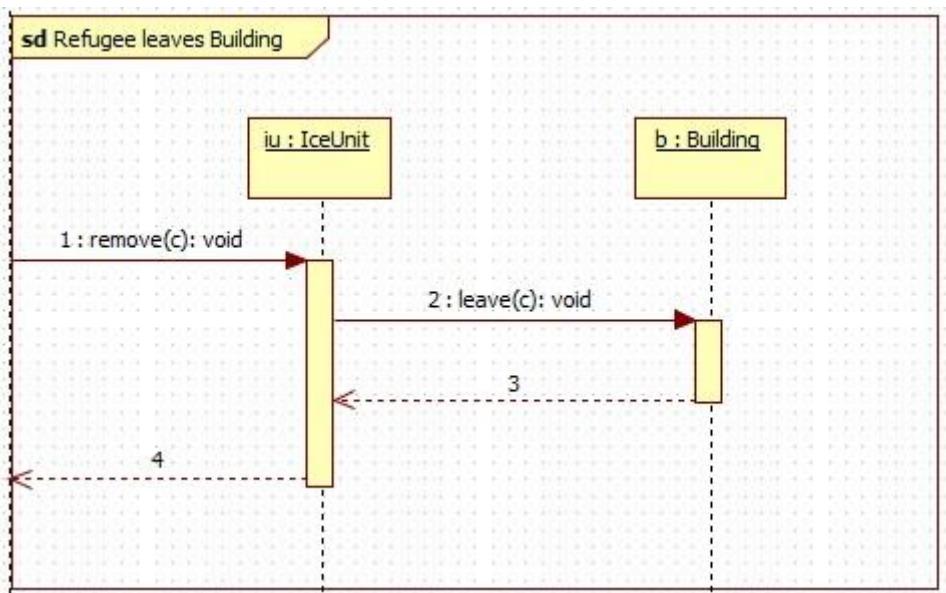
## 12) pickedBy (Tent)

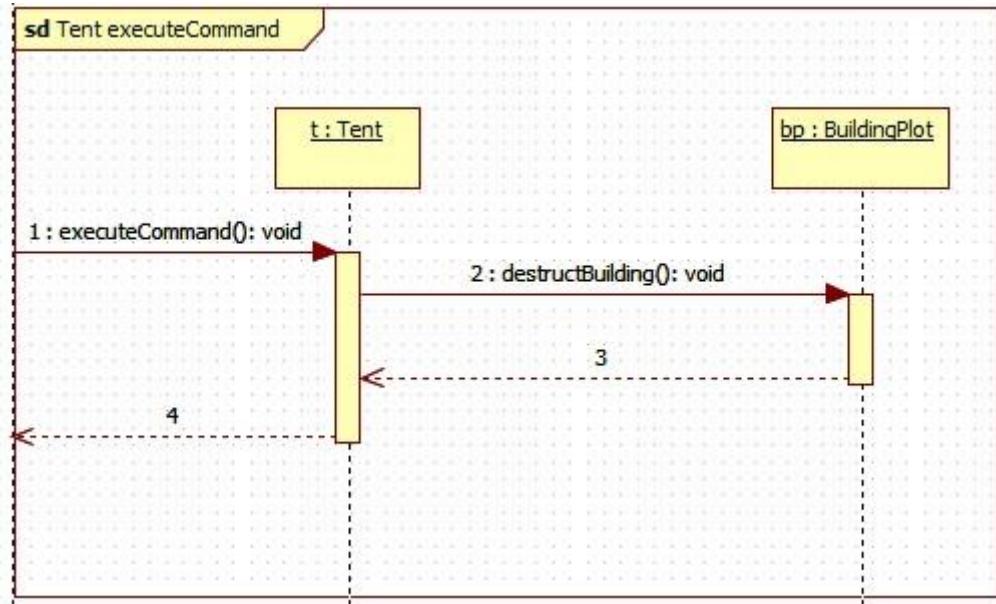
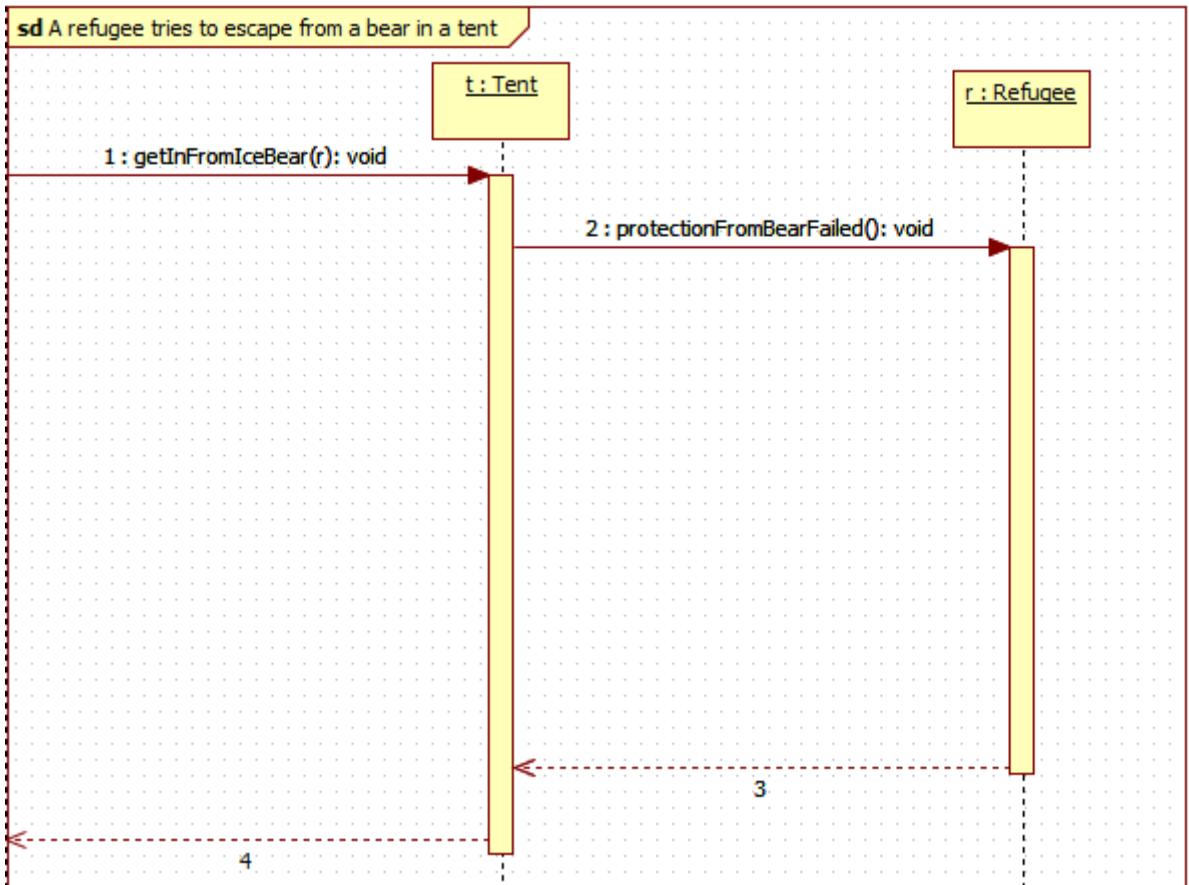


## 13) plains\_getin\_bear

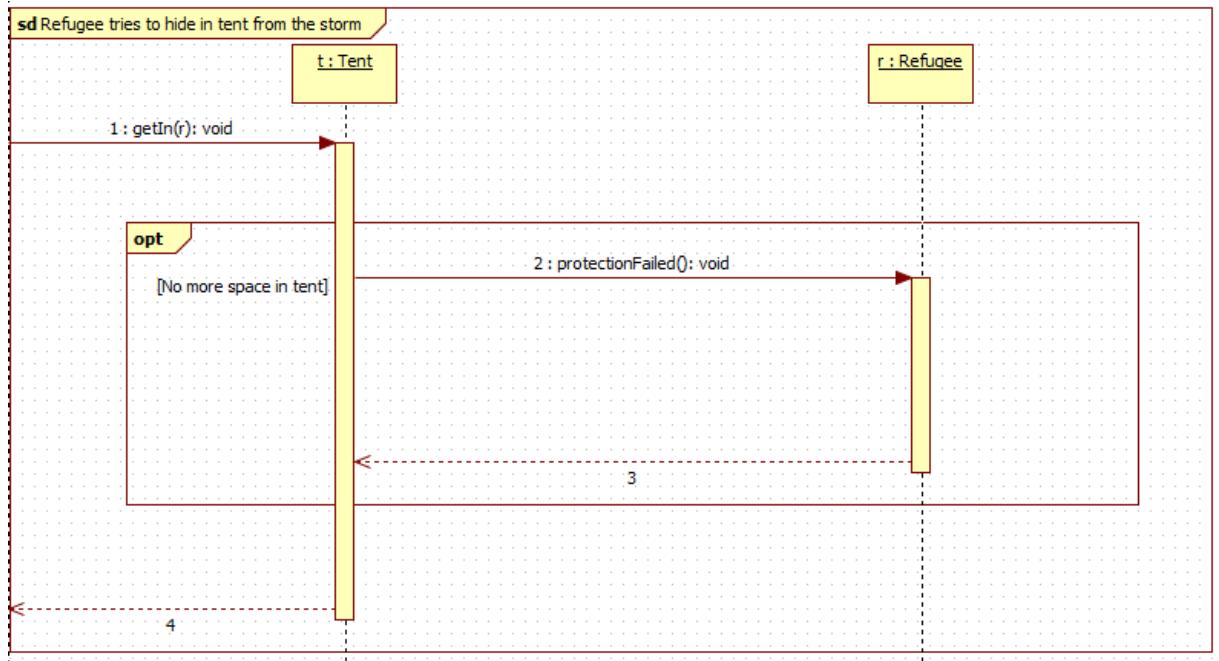


## 14) Refugee leaves Building

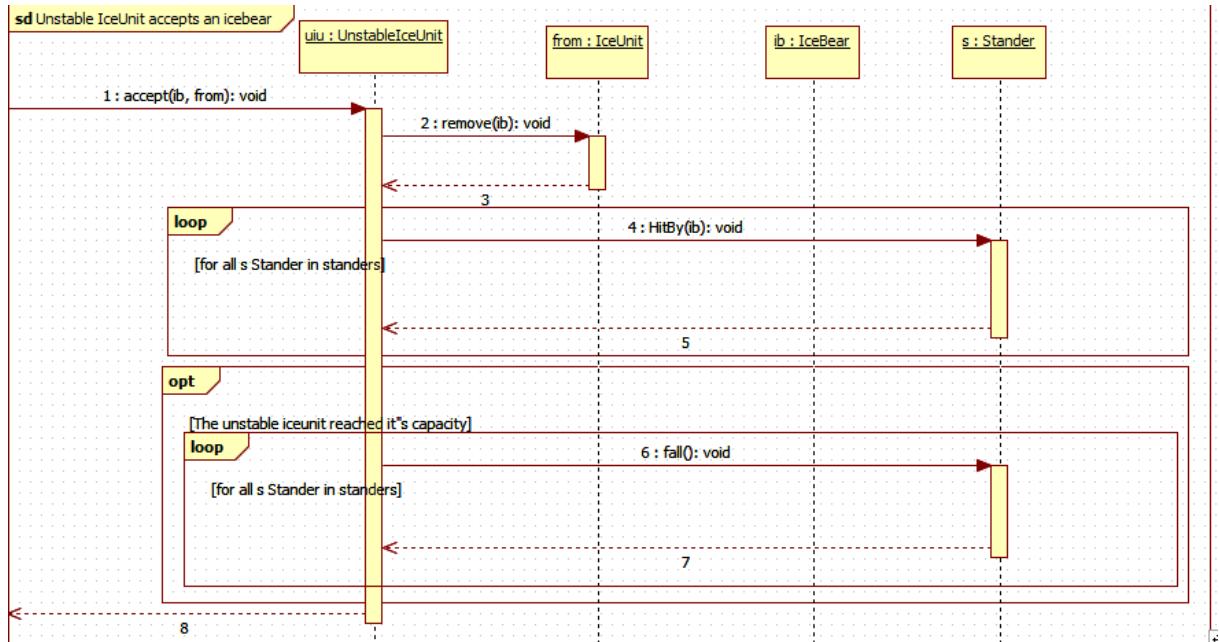


**15) Tent executeCommand****16) tent\_getin\_bear**

### 17) tent\_getin\_storm



### 18) unstable\_accepts\_bear



## 7.1 Prototípus interface-definíciója

### 7.1.1 Az interfész általános leírása

A prototípus program parancssoron keresztül lesz vezérelhető. A tesztelés megkönnyítése érdekében előre elkészített forgatókönyvek segítségével is képes lesz működni, valamint egy segédprogram segítségével a teljes tesztelési folyamat automatizálható lesz. A futtatás kimenete parancssorra és fájlba is kiíródhat.

### 7.1.2 Bemeneti nyelv

Tesztelést szolgáló parancsok:

**start**

**Leírás:** Letiltja a pályaépítést szolgáló parancsok használatát.

**end**

**Leírás:** Engedélyezi a pályaépítést szolgáló parancsok használatát.

**loadmap <name>**

**Leírás:** Betölt egy pályát egy fájlból.

**Opciók:** name: Pályafájl neve (fájlkiterjesztés nélkül). A pályafájlok egy külön mappában (maps) érhetőek el.

**list [<type>]**

**Leírás:** Kiírja a pályán lévő objektumok azonosítóját a szabványos kimenetre.

**Opciók:** type: Objektumtípus specifikálása. Lehetséges értékek: {tile, building, character, item, bear}.

Amennyiben nem adunk meg típust, az összes azonosító kiírásra kerül.

**stats [<id>]**

**Leírás:** Kiírja egy objektum tulajdonságait a szabványos kimenetre.

**Opciók:** id: Objektum azonosítója.

Amennyiben nem adunk meg azonosítót, az összes objektum kiírásra kerül.

**move <entityid> <fieldid>**

**Leírás:** Entitás léptetése.

**Opciók:** entityid: Entitás azonosítója.

fieldid: Annak a mezőnek az azonosítója, ahova az entitást akarjuk mozgatni.

Csak a jelenleg aktív körben lévő karakter tud lépni.

**ability <fieldid>**

**Leírás:** Kerekterképesség használata.

**Opciók:** fieldid: Annak a mezőnek az azonosítója, melyen a képességet akarjuk használni.

**dig**

**Leírás:** Ásás a jelenlegi karakterrel.

**tent**

**Leírás:** Sátor lerakása a jelenlegi karakterrel.

**storm ...**

... <fieldid>

... <characterid>

**Leírás:** Hóvihar indítás.

**Opciók:** fieldid: Annak a mezőnek az azonosítója, melyet hóvihar súlyt.

**characterid:** Annak a karakternek az azonosítója, melyet hóvihar súlyt.

### **assemble**

**Leírás:** Jelenlegi karakter utasítása pisztoly összerakására.

### **turn [<id>]**

**Leírás:** Átadja a kört a következő irányítható objektumnak.

**Opciók:** **id:** Következő objektum azonosítója.

Ha nem adjuk meg az azonosítót, akkor az alapértelmezés szerint választja ki a következő objektumot.

Pályaépítést szolgáló parancsok:

### **field <id> [<capacity> [(broken)]] <snow>**

**Leírás:** Létrehoz egy mezőt.

**Opciók:** **id:** Mező azonosítója.

**capacity:** Instabil jégmező teherbíró képessége.

**broken:** Törött-e már a jégmező.

**snow:** Hó mennyisége.

Ha nem adjuk meg a teherbíró képességet, akkor stabil jégtábla keletkezik.

### **neighbours <id> <id1> [<id2> [<id3> [...]]]**

**Leírás:** Beállítja egy mező szomszédjait.

**Opciók:** **id:** Annak a mezőnek az azonosítója, amelynek a szomszédjait állítjuk.

**id1, id2, id3, ...:** Szomszédos mezők azonosítója.

### **building <id> <type> <fieldid>**

**Leírás:** Létrehoz egy építményt.

**Opciók:** **id:** Építmény azonosítója.

**type:** Építmény típusa. Lehetséges értékek: {Plains, Tent, Igloo}.

**fieldid:** Annak a mezőnek az azonosítója, amelyen az építmény lesz.

### **item <id> <type> ...**

**... setcharacter <characterid>**

**... setfield <fieldid>**

**Leírás:** Létrehoz egy tárgyat.

**Opciók:** **id:** Tárgy azonosítója.

**type:** Tárgy típusa. Lehetséges értékek: {Pistol, Cartridge, Light, Lung, Suit, BasicPull, Rope, BareHand, Shovel, BreakableShovel, Food}

**setcharacter:** Hozzáadja a tárgyat egy karakterhez.

**characterid:** Karakter azonosítója.

**setfield:** Hozzáadja a tárgyat egy mezőhöz

**fieldid:** Mező azonosítója.

### **entity <id> <type> <fieldid>**

**Leírás:** Létrehoz egy entitást.

**Opciók:** **id:** Entitas azonosítója.

**type:** Entitás típusa. Lehetséges értékek: {Bear, Eskimo, Explorer}  
**fieldid:** Annak a mezőnek az azonosítója, amelyen az entitás lesz.

#### **savemap <name>**

**Leírás:** Jelenlegi pálya elmentése egy fájlba.

**Opciók:** name: Pálya neve.

A pályákat JSON formátumú fájlokba lehet elmenteni.

### 7.1.3 Kimeneti nyelv

A stats és list parancs az egyetlen, melynek van kimenete. A többi parancs csak súlyos hiba esetén ír ki üzenetet.

#### **list:**

A parancs minden objektumtípusra külön-külön kilistázza az azonosítókat, az alábbi formában:

```
tile: [t1]
building: [b1, b2, b3]
character: []
...
...
```

Amennyiben van megadva paraméter, csak annak megfelelő sort írja ki a parancs.

#### **stats:**

A parancs különböző típusú objektumokra különböző módon írja ki a tulajdonságokat.

#### **IceUnit:**

```
id: i0
snow: 3
building: b0
pickable: null
standers: [c1, c2]
pullers: []
masters: [c1]
neighbours: [i1, i2, u1]
```

#### **UnstableIceUnit:**

```
id: u0
snow: 1
capacity: 3
broken: false
building: b2
pickable: p1
standers: []
masters: []
neighbours: [i1, u1]
```

Plains:

```
id: b0
capacity: 0
plot: i0
refugees: []
```

Tent, Igloo:

```
id: b4
capacity: 0
health: 2
plot: i0
refugees: [c1]
```

IceBear:

```
id: e0
iceUnit: i1
```

Eksimo, Explorer:

```
id: c4
iceUnit: i3
bodyHeat: 2
stamina: 1
digger: d0
waterProtecter: w0
pullerDevice: r1
tent: null
components: [p0, p2]
```

Shovel, Barehand, Food:

```
id: d0
effectiveness: 1
```

BreakableShovel:

```
id: d3
durability: 2
effectiveness: 2
```

Rope, BaiscPill, Lung, DivingSuit, AlarmPistol:

```
id: 16
```

Pistol, Cartridge, Light

```
id: a4
type: "pistol"
```

## 7.2 Összes részletes use-case

<b>Use-case neve</b>	start
<b>Rövid leírás</b>	Elindítja a játékot a beállított játék tulajdonságokkal.
<b>Aktorok</b>	Játékos, vezérlő
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. Letiltja a pályaépítést szolgáló parancsok használatát.</li> <li>2. Játék elindítása.</li> </ol>

<b>Use-case neve</b>	end
<b>Rövid leírás</b>	Leállítja a játékot.
<b>Aktorok</b>	Játékos, vezérlő
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. Játék leállítása.</li> <li>2. Engedélyezi a pályaépítést szolgáló parancsok használatát.</li> </ol>

<b>Use-case neve</b>	loadmap
<b>Rövid leírás</b>	Egy fájlból betölt egy pályát.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. Fájl betöltése.</li> <li>2. Mezők létrehozása.</li> <li>3. Mezők összeköttetése.</li> </ol>

<b>Use-case neve</b>	list
<b>Rövid leírás</b>	Kiírja a pályán lévő objektumok azonosítóját a szabványos kimenetre.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	1. Meghatározott azonosítók kiírása.

<b>Use-case neve</b>	stats
<b>Rövid leírás</b>	Kiírja egy objektum tulajdonságait a szabványos kimenetre.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	1. Meghatározott objektum tulajdonságainak kiírása

<b>Use-case neve</b>	move
<b>Rövid leírás</b>	Entitás léptetése egy másik mezőre.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	1. Entitás léptetése a megadott mezőre.

<b>Use-case neve</b>	ability
<b>Rövid leírás</b>	A körön lévő karakter képességének használata egy mezőn.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	1. Képesség használata a meghatározott mezőn.

<b>Use-case neve</b>	dig
<b>Rövid leírás</b>	Ásás a körön lévő karakterrel.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	1. Ásás a karakterrel

<b>Use-case neve</b>	tent
<b>Rövid leírás</b>	Sátor lerakása a körön lévő karakterrel.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	1. Sátor lerakása

<b>Use-case neve</b>	storm
<b>Rövid leírás</b>	Hóvihar indítás egy adott karaktere vagy mezőre.
<b>Aktorok</b>	Vezérző
<b>Forgatókönyv</b>	1. Hóvihar indítás a meghatározott dologra.

<b>Use-case neve</b>	assemble
<b>Rövid leírás</b>	A körön lévő karakter összerakja a pisztolyt.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	1. Pisztoly összerakása.

<b>Use-case neve</b>	turn
<b>Rövid leírás</b>	Atadja a kört a körben következő irányítható objektumnak.
<b>Aktorok</b>	Játékos
<b>Forgatókönyv</b>	1. Kör átadása

<b>Use-case neve</b>	field
<b>Rövid leírás</b>	Létrehoz egy mezőt a pályán a megadott tulajdonságokkal.
<b>Aktorok</b>	Vezérző
<b>Forgatókönyv</b>	1. Mező létrehozása a meghatározott tulajdonságokkal.

<b>Use-case neve</b>	neighbour
<b>Rövid leírás</b>	Beállítja a pályán egy mező szomszédjait.
<b>Aktorok</b>	Vezérző
<b>Forgatókönyv</b>	1. Meghatározott szomszéd beállítása

<b>Use-case neve</b>	building
<b>Rövid leírás</b>	Létrehoz egy építményt egy mezőn.
<b>Aktorok</b>	Vezérző
<b>Forgatókönyv</b>	1. Építmény létrehozása a meghatározott mezőn.

<b>Use-case neve</b>	item
<b>Rövid leírás</b>	Létrehoz egy tárgyat, amit hozzálehet adni egy karakterhez vagy egy mezőhöz.
<b>Aktorok</b>	Vezérző
<b>Forgatókönyv</b>	1. Tárgy létrehozása a megadott helyen.

<b>Use-case neve</b>	entity
<b>Rövid leírás</b>	Létrehoz egy entitást egy mezőn.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	1. Entitás létrehozása a megadott mezőn.

<b>Use-case neve</b>	savemap
<b>Rövid leírás</b>	Jelenlegi pálya elmentése egy fájlba.
<b>Aktorok</b>	Vezérlő
<b>Forgatókönyv</b>	1. Pálya elmentése a megadott fájlba.

### 7.3 Tesztelési terv

<b>Teszt-eset neve</b>	Pálya betöltése
<b>Rövid leírás</b>	A teszt azt ellenőrzi, hogy egy forrásfájlból betöltött pályát helyesen olvasta-e be a fájl.
<b>Teszt célja</b>	A teszt célja, hogy ellenőrizzük, hogy a pálya beolvasása fájlból megfelelően működik.

<b>Teszt-eset neve</b>	Mozgó objektumok és mozgásformák tesztelése
<b>Rövid leírás</b>	A tesztben azt ellenőrizzük, hogy a követelményekben megfogalmazott szabályok szerint működnek a játékban a különböző mozgásformák, mint a szárazföldi és a kétéltű mozgásforma; és mozgó objektumok, mint a karakterek és jegesmedve mozgása helyes módon működik. A tesztben különböző tulajdonságú jégtáblák között próbálunk karaktereket/jegesmedvéket mozgatni, és megvizsgáljuk, hogy a mozgásformájuk ezt helyesen végrehajtotta, vagy akadályozta meg a szabálytalan mozgást; és ha megtörtént a mozgás végrehajtása, ez helyesen jelződött a cél és forrás jégtábláknak.
<b>Teszt célja</b>	A teszt célja annak ellenőrzése, hogy a különböző mozgástulajdonsággal rendelkező mozgó objektumok mozgástulajdonságuk szerint szabályosan és helyes módon mozognak.

<b>Teszt-eset neve</b>	Jégtábla betörése
<b>Rövid leírás</b>	A tesztben egy megadott instabil jégtábla terhelését lehet tesztelni. A tesztben a jégtáblára öt terhelő objektumokat helyezünk, és azt ellenőrizzük, hogy a helyes pillanatba törik be a jégtábla.
<b>Teszt célja</b>	A teszt célja annak az ellenőrzése, hogy a jégtábla betörése a követelményeknek megfelelően működik-e, a betörés helyes esetekben következik csak be.

Teszt-eset neve	Karakter megfelelő reagálása a jégtábla betörésére
<b>Rövid leírás</b>	Karaktereket helyezünk a jégtáblára, illetve a szomszédos jégtáblára is. A jégtáblát addig terheljük, amíg be nem törik. Ezután a karakterek reakcióját kell vizsgálni; a karakterek száma és tulajdonságuk olyan, hogy kimerítően tudjuk tesztelni összes lehetséges reakciót erre az eseményre. Tehát a teszt magába foglalja, ha egy karakter megfullad, ha egy karakter túlél a búvárruhájával, vagy ha egy karaktert kötéssel kiment egy másik.
<b>Teszt célja</b>	A teszt célja, hogy letesztelje az összes karakterreakciót arra az esetre, ha a karakter alatt betörik a jégtábla.

Teszt-eset neve	Eszkimó iglut épít
<b>Rövid leírás</b>	Elhelyezünk egy eszkimót egy jégtáblán, és megpróbálunk vele ráépíteni egy iglut, majd ellenőrizzük, hogy az iglu megfelelő módon jött létre. A tesztben fontos ellenőrizni, hogy a képesség használata után a karakter munkavégző képessége megfelelően változott-e (csökkent). Ezt az ellenőrzést különböző tulajdonságú jégtáblákon el kell végezni, és ellenőrizni kell a képesség használatának sikereségét, illetve hogy ez az elvárt eredmény-e.
<b>Teszt célja</b>	A teszt célja annak ellenőrzése, hogy az eszkimó iglu építő képessége a követelményeknek megfelelően működik.

Teszt-eset neve	Felfedező felfedi egy jégtábla teherbíró képességét
<b>Rövid leírás</b>	A tesztben egy felfedező képességét használjuk több jégtáblára is, és ellenőrizzük, hogy a képesség használata helyes eredményt mutat-e. A tesztben fontos ellenőrizni, hogy a képesség használata után a karakter munkavégző képessége megfelelően változott-e (csökkent).
<b>Teszt célja</b>	A teszt célja, hogy ellenőrizze, hogy a felfedező jégtábla teherbíró képességet ellenőrző képessége helyesen működik.

Teszt-eset neve	Jelzőpisztoly összeszerelése
<b>Rövid leírás</b>	A tesztnél ellenőriznünk kell, hogy a karakter csak a megfelelő komponensek összességéből tudja összerakni a jelzőpisztolyt. Továbbá fontos ellenőrzendő funkció, hogy amikor több karakter próbálja meg összeállítani együtt a jelzőpisztolyt, az is hibamentesen működik. A tesztnél ellenőrizzük azt is, hogy amikor összeáll a pisztoly, az a játékból megfelelő reakciót vált ki, tehát a játék érzékeli a győzelmet.
<b>Teszt célja</b>	A teszt célja annak ellenőrzése, hogy a jelzőpisztoly összeállítás a követelményekben megfogalmazottaknak megfelelően működik.

Teszt-eset neve	A vihar eléri a jégmezőt
<b>Rövid leírás</b>	A vihar egy olyan pályát fog elérni, ahol megvizsgálhatjuk az összes lehetséges reakciót a viharra. Ez magába foglalja azokat az ellenőrzéseket; hogy az elért jégtáblákra rakódik, és megfelelő mennyiségi hó rakódik le a vihar hatására; hogy a karaktereket a különböző épületek, esetleg azok telítettségének függvényében, megfelelően védi-e meg, illetve ha a védelem elbukik, a karaktereknek megfelelően csökken a test hőjük.
<b>Teszt célja</b>	A teszt célja annak ellenőrzése, hogy a vihar kiváltja-e megfelelő hatásokat a jégmező általa érinthető objektumaiiból.

Teszt-eset neve	Karakterek és jegesmedvék találkozása
<b>Rövid leírás</b>	A tesztnél egy mezőre mozgatunk egy karaktert egy jegesmedvével; illetve egy jegesmedvét és egy karakterrel is. Ekkor ellenőrizzük, hogy a jégtáblán található épület függvényében a játékra megfelelő hatása lesz ennek a két objektum találkozásának. A tesztnél azt is megvizsgáljuk, hogy amikor egy-egy karakter, vagy egy-egy jegesmedve találkozik, annak nem lesz semmilyen hatása a játéknak.
<b>Teszt célja</b>	A teszt célja annak az ellenőrzése, hogy a jegesmedve és karakter hatása minden esetben kiváltja a követelményeknek megfelelő hatásokat.

Teszt-eset neve	Ásás a jégtáblán
<b>Rövid leírás</b>	A tesztben ásni próbálunk különböző tulajdonságú jégtáblákon, különböző tulajdonságú ásási eszközökkel. A jégtáblán meg kell vizsgálni, hogy azon a hó mennyisége a vizsgált ásóeszköz tulajdonságainak megfelelően változott, illetve annak tárgyát ássuk ki, akkor az tovább nem lesz megtalálható a jégtáblán. Az ásóeszköznek meg kell vizsgálni, hogy az tulajdonságának megfelelően módosítja a jégtáblán található hó mennyiséget, illetve, ha az ásóeszköz tulajdonsága azt indokolja, akkor annak tulajdonsága megfelelően változik (pl. törékeny ásó esetén). Az ásást végző karakterek pedig ellenőrizni, hogy ha tárgyat vesz fel, azt sikeresen felveszi-e, illetve munkavégző képessége helyes módon változik.
<b>Teszt célja</b>	A teszt célja ellenőrizni, hogy a játékban a karakter ásás funkciója a követelményeknek megfelelően működik.

Teszt-eset neve	Sátor felállítása
<b>Rövid leírás</b>	A tesztben elhelyezünk egy karaktert egy jégtáblán, a karakter rendelkezni fog egy sátorral az eszközei között. Megpróbáljuk elhelyezni a sárat a jégtáblán, és ennek sikerességét ellenőrzük. Ezt az ellenőrzést el kell végezni többféle jégtáblára (pl. törött – nem törött), és ellenőrizni, hogy a sátor felépült vagy sem, illetve hogy ez elvárt viselkedés-e.
<b>Teszt célja</b>	A teszt célja annak ellenőrzése, hogy a sátor a követelményeknek megfelelően elhelyezhető a karakter jégtábláján, ha jégtábla tulajdonságai ezt engedik.

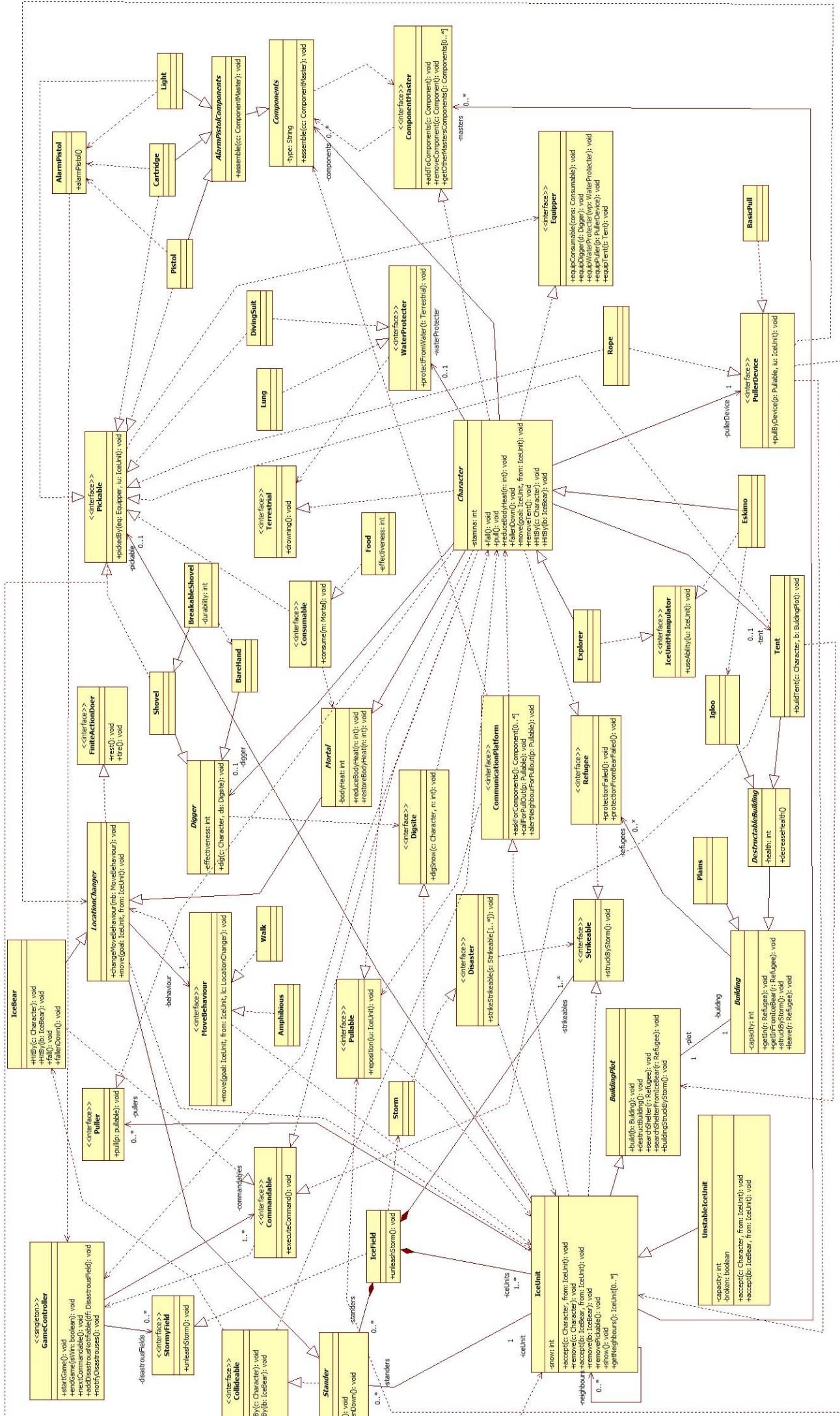
Teszt-eset neve	Játék körök szerinti vezérlése
<b>Rövid leírás</b>	A tesztben azt ellenőrzük, hogy megvalósul-e a játékban az a funkció, hogy a cselekedetek körökben hajtódnak végre, a körök műlása pedig megfelelően vezérelhető. A tesztben körök műlására érzékeny objektumokat helyezünk el, és megvizsgáljuk a viselkedésüket. Például, hogy a sátor megfelelő kör elteltével elpusztul-e, vagy a jegesmedvék és karakterek körei helyesen következnek-e egymás után.
<b>Teszt célja</b>	A teszt célja, hogy a játék megfelelően vezérelhető-e, illetve hogy az objektumok, amik érzékenyek a körök műlására, azok reakciója megfelelő-e.

#### **7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása**

A teszteléshez írni fogunk egy segédprogramot, ami a fájlokba mentett teszteseteket automatikusan lefuttatja, majd kiértékeli ezek sikerességét. A tesztesetek lefuttatása után terminálba kiírja, hogy hány teszteset lett sikeres. Sikertelen teszt esetén kiírja a visszakapott kimenetet.

**Napló**

<b>Kezdet</b>	<b>Időtartam</b>	<b>Résztvevők</b>	<b>Leírás</b>
2020.04.01. 16:00	3 óra	Kaposvári Labancz Iuhos Sulyok Traxler	Értekezlet. Változtatások megbeszélése, feladatok kiosztása.
2020.04.01. 20:00	3 óra	Sulyok	Tevékenység: Osztálydiagram módosítása
2020.04.02. 14:00	4 óra	Traxler	Tevékenység: Új vagy megváltozott metódusok megtervezése, leírások elkészítése.
2020.04.02. 17:00	3 óra	Iuhos	Tevékenység: Szekvenciadiagramok készítése.
2020.04.02. 19:00	2 óra	Kaposvári	Tevékenység: Szekvenciadiagramok készítése.
2020.04.03. 13:00	4 óra	Kaposvári Iuhos Sulyok Traxler	Értekezlet: Prototípus interfész definíciójának megbeszélése.
2020.04.03. 18:00	5 óra	Sulyok	Tevékenység: Interfész általános leírásának elkészítése, bemeneti nyelv leírása.
2020.04.03. 18:00	4 óra	Traxler	Tevékenység: Kimeneti nyelv leírása.
2020.04.04. 14:00	6 óra	Kaposvári	Tevékenység: Use-case leírások elkészítése.
2020.04.04. 16:00	5 óra	Iuhos	Tevékenység: Tesztelési terv elkészítése, tesztelést támogató segéd- és fordítóprogramok specifikálása.
2020.04.05. 17:00	3 óra	Labancz	Tevékenység: Dokumentum elkészítése, formázása, javítása.



# **8. Feladat: Részletes tervezek - beadás**

**14 – the\_minesweepers**

Konzulens:  
**Fetter László**

## **Csapattagok**

Iuhos Zoltán	J8LOUL	zoltan.iuhos@gmail.com
Kaposvári Márk Martin	RAYVF8	xmarko45@gmail.com
Sulyok Botond	BM2BBM	botond.sulyok99@gmail.com
Labancz Tamás Dominik	A9HZBF	labancz.tamas@gmail.com
Traxler Bálint	IXNOYB	traxlerbalint@gmail.com

2020.04.14.

## 8. Részletes tervez

### 8.1 Osztályok és metódusok tervez.

#### 8.1.1 Pickable

- **Felelősség**

A felvehető tárgyak számára írja elő a felvétel megvalósítását.

- **Metódusok**

- **+pickedBy(Character c, IceUnit iu):** A felvétel módjának megvalósításának implementálását írja elő.

#### 8.1.2 Consumable

- **Felelősség**

Az elfogyasztható tárgyak interfésze.

- **Ősosztály**

Pickable

- **Metódusok**

- **+void consume(Mortal m):** Előírja az elfogyasztás módjának implementálását.

#### 8.1.3 WaterProtector

- **Felelősség**

A vízbeesés ellen védő játékmechanikának megvalósítását írja elő.

- **Metódusok**

- **+void protectFromWater(Terrestrial t):** Előírja a víz ellen védés játékmechanikájának megvalósítását.

#### 8.1.4 PullerDevice

- **Felelősség**

A húzást megvalósító eszközök interfésze.

- **Metódusok**

- **+void pullByDevice(Pullable p, IceUnit iu):** A húzás működésének implementálását írja elő.

### 8.1.5 Food

- **Felelősség**

A játékban a felvehető ételt reprezentálja. Egy felvehető tárgy, melyet ha valaki felvesz, akkor a felvevőre bizonyos hatást gyakorol, a felvételt követően azonnal kifejti hatását. A Mortal típusú objektumok képesek a tárgy elfogyasztására, ezeknek az objektumoknak növeli a testhőjét, azt, hogy mennyivel növeli a testhőt egy privát attribútumban tárolja.

- **Interfészek**

Consumable

- **Attribútumok**

- **-int effectiveness:** Ez az érték azt adja meg, hogy mennyire hatékony az étel. Ezt a hatékonyságot úgy kell értelmezni, hogy hány testhő visszatöltésére képes. Ezt az értéket az objektum létrehozásakor kell megadni és később nem megváltoztatható.

- **Metódusok**

- **+void pickedBy (Character c, IceUnit iu):** Amikor egy karakterhez kerül a tárgy, akkor ez metódus ennek a viselkedésnek a levezénylését végzi. Tehát ha valaki felvette ezt a tárgyat, akkor annak meghívja az equipConsumable metódusát saját magával paraméterezve. A felvevő majd eldönti, hogy hogyan használja ezt a tárgyat. Alapból a karakterek az ételt azonnal elfogyasztják. A metódus gondokodik arról is, hogy a paraméterként kapott mezőn, tehát amin a felvehető tárgy volt, arról eltávolítsa magát, mivel felvette már valaki, tehát nem a mezőn van, hanem például egy karakternél. Ezt úgy teszi meg, hogy meghívja a removePickable függvényét a paraméterként kapott IceUnitnak saját magával paraméterezve.

- **+consume(m: Mortal): void :** A metódus hatására az ételt elfogyasztó, paraméterként kapott Mortalnak az étel effektívségének alapján testhőt tölt vissza. Ezt úgy teszi meg, hogy meghívja a restoreBodyHeat függvényét az effectiveness attribútum értékével paraméterezve.

### 8.1.6 Digger

- **Felelősség**

Absztrakt osztály, aminek felelőssége, hogy az ásás műveletét levezényelje.

- **Attribútumok**

- **-int effectiveness:** Az ásás hatékonyságát jelöli. Azaz, hogy mennyi havat képes eltakarítani egy jágtábláról.

- **Metódusok**

- **+void dig(Character c, IceUnit iu):** Az ásást levezénylését végzi, a paraméterként megkapott IceUnit-nak meghívja a digSnow metódusát, saját magával és az effectiveness attribútumának értékével paraméterezve, azaz, hogy mennyi havat tud eltakarítani, amivel az IceUnit ezt az interakciót kezelni tudja.

### 8.1.7 Shovel

- **Felelősség**

Egy ásó fajta, amivel több havat (2) lehet eltakarítani, ahhoz képest, mint ha nem rendelkeznénk ezzel az eszközzel. Ez az eszköz nem tud elhasználódni.

- **Ősosztályok**

Digger

- **Interfészek**

Pickable

- **Metódusok**

- **+void pickedByCharacter(Character c, IceUnit iu):** Amikor egy karakterhez kerül a tárgy, akkor ez metódus ennek a viselkedésnek a levezénylését végzi. Tehát ha valaki felvette ezt a tárgyat, akkor annak meghívja az equipDigger metódusát saját magával paraméterezve. A felvett majd eldönti, hogy hogyan használja ezt a tárgyat. A metódus gondolkodik arról is, hogy a paraméterként kapott mezőn, tehát amin a felvehető tárgy volt, arról eltávolítsa magát, mivel felvette már valaki, tehát nem a mezőn van, hanem például egy karakternél. Ezt úgy teszi meg, hogy meghívja a removePickable függvényét a paraméterként kapott IceUnitnak saját magával paraméterezve.

### 8.1.8 BreakableShovel

- **Felelősség**

A törhető ásó ugyanúgy működik, mint a hagyományos ásó, viszont ez el tud törni egy bizonyos számú (3) használat után.

`dig/digSnow, durability-- : [ durability is not 1 ]`

Works

`durability is 0 (broken)`

`do/equipDefaultDigger`

`dig/digSnow, durability-- : [ durability is 1 ]`

- **Ősosztályok**

Digger

- **Interfészek**

Pickable

- **Attribútumok**

**-int durability:** Az ásónak a tartósságát határozza meg, az ásó minden egyes használatakor csökken az értéke eggel.

- **Metódusok**

- **void dig(Character c, IceUnit iu):** Az ásást levezénylését végzi, a paraméterként megkapott IceUnit-nak meghívja a digSnow metódusát, saját magával és az effectiveness attribútumának értékével paraméterezve, azaz, hogy mennyi havat tud eltakarítani, amivel az IceUnit ezt az interakciót kezelní tudja. Ha elfogyott az ásónak a tartóssága, akkor eltörik és az őt birtokló karakter is elveszti és a keze lesz az „ásóeszköze”. Ezt úgy éri el, hogy ha a durability attribútumának értéke 0 lesz, akkor meghívja a használójának equipDefaultDigger metódusát.

### 8.1.9 BareHand

- **Felelősség**

Egy ásó fajta, tud ájni, de nem rendelkezik semmilyen további viselkedéssel. Egy hó eltakarítására képes.

- **Ősosztályok**

Digger

### 8.1.10 Lung

- **Felelősség**

Azt reprezentálja, amikor a karakter nem rendelkezik semmilyen a víz ellen védő eszközzel és ekkor a karakter meg tud halni.

- **Interfészek**

WaterProtector

- **Metódusok**

- **+void protectFromWater(Terrestrial t):** A paraméterként kapott Terrestrial objektumok meghívja a drowning metódusát, ami a karakter fulladását idézi elő, mivel ha a karakter egy ilyen eszközzel rendelkezik, az azt jelenti, hogy nincs védve a tengertől.

### 8.1.11 DivingSuit

- **Felelősség**

Buvárruha, amit fel lehet venni és a tulajdonosa ennek köszönhetően védettséget élvez a vízbe esés után esetlegesen bekövetkező haláltól.

- **Interfészek**

Pickable, WaterProtecter

- **Metódusok**

- **+void pickedBy(Character c, IceUnit iu):** Amikor egy karakterhez kerül a tárgy, akkor ez metódus ennek a viselkedésnek a levezénylését végzi. Tehát ha valaki felvette ezt a tárgyat, akkor annak meghívja az equipWaterProtecter metódusát saját magával paraméterezve. A felvettő majd eldönti, hogy hogyan használja ezt a tárgyat. A metódus gondokodik arról is, hogy a paraméterként kapott mezőn, tehát amin a felvehető tárgy volt, arról eltávolítsa magát, mivel felvette már valaki, tehát nem a mezőn van, hanem például egy karakternél. Ezt úgy teszi meg, hogy meghívja a removePickable függvényét a paraméterként kapott IceUnitnak saját magával paraméterezve.

- **+void protectFromWater(Terrestrial t):** Megvédi a vízbe esett Terrestrial objektumot a fulladástól. Ezt egyszerűen csak úgy teszi meg, hogy nem hívja meg a paraméterként kapott Terrestrial fulladást jelző drowning nevű függvényét.

### 8.1.12 Rope

- **Felelősség**

Egy felvehető tárgy, ami képes karakterek vízből való kimentésére.

- **Interfészek**

Pickable, PullerDevice

- **Metódusok**

- **+void pickedBy (Character c, IceUnit iu):** Amikor egy karakterhez kerül a tárgy, akkor ez metódus ennek a viselkedésnek a levezénylését végzi. Tehát ha valaki felvette ezt a tárgyat, akkor annak meghívja az equipPuller metódusát saját magával paraméterezve. A felvettő majd eldönti, hogy hogyan használja ezt a tárgyat. A metódus gondokodik arról is, hogy a paraméterként kapott mezőn, tehát amin a felvehető tárgy volt, arról eltávolítsa magát, mivel felvette már valaki, tehát nem a mezőn van, hanem például egy karakternél. Ezt úgy teszi meg, hogy meghívja a removePickable függvényét a paraméterként kapott IceUnitnak saját magával paraméterezve.

- **+void pullByDevice(Pullable p, IceUnit iu):** A paraméterként kapott Pullable objektumot, ami akár lehet egy karakter, kihúzza a vízből. A kihúzott objektum arra az IceUnit-ra kerül, ahol a kötél tárgyának tulajdonosa is áll. Ezt úgy éri el, hogy meghívja a

paraméterként kapott Pullable objektumnak (egy karakter is lehet ez) a reposition metódusát a paraméterként kapott IceUnit objektummal paraméterezi, ez az IceUnit ahol a kötél tulajdonosa, azaz a húzó tartózkodik és ide kerül a kihúzott objektum is.

### 8.1.13 BasicPull

- **Felelősség**

Azt jelzi, hogy a használói nem képesek egy másik jelen, alapértelmezett esetben karaktert a vízből kihúzni.

- **Interfészek**

PullerDevice

- **Metódusok**

- **+void pullByDevice(Pullable p, IceUnit iu):** Ez az osztály nem képes kihúzni a paréterként kapott objektumot, ezt a viselkedést valósítja meg ez a metódus. Tehát lényegében nem csinál semmit.

### 8.1.14 Tent

- **Felelősség**

A sátor reprezentálja, ami megvédi a benne tartózkodókat a vihartól, de medvetámadástól viszont már nem.

- **Ősosztályok**

Building → DestructableBuilding

- **Metódusok**

- **+void getInFromIceBear(r: Refugee):** Bemegy a sátorba a paraméterben kapott jegesmedve elől menekülő objektum, azonban ez semmilyen fajta védelmet nem nyújt neki a medvetámadás elől, azaz meghívja a menekülő objektum protectionFromBearFailed metódusát.

- **+void buildTent(c: Character, b: BuildingPlot):** Felépíti a sárat a paraméterként kapott építőhelyre és a karaktertől elveszi a sátorral rendelkezés jogát: meghívja a paraméterként kapott BuildingPlot típusú objektum build függvényét, majd a karakter removeTent metódusát.

- **+void executeCommand():** Végrehajtja a megfelelő parancsot, ez minden körben egyszer következik be a sátor esetében és ez azt eredményezi, hogy a sátor egy kör után eltűnik, eltávolítva magát az épülethelyéről: meghívja az épülethelyének a destructBuilding metódusát.

- **+void pickedByCharacter(Character c, IceUnit iu):** Amikor egy karakterhez kerül a tárgy, akkor ez metódus ennek a viselkedésnek a levezénylését végzi. Tehát ha valaki

felvette ezt a tárgyat, akkor annak meghívja az equipTent metódusát saját magával paraméterezve. A felvevő majd eldönti, hogy hogyan használja ezt a tárgyat. A metódus gondokodik arról is, hogy a paraméterként kapott mezőn, tehát amin a felvehető tárgy volt, arról eltávolítsa magát, mivel felvette már valaki, tehát nem a mezőn van, hanem például egy karakternél. Ezt úgy teszi meg, hogy meghívja a removePickable függvényét a paraméterként kapott IceUnitnak saját magával paraméterezve.

### 8.1.15 Components

- **Felelősség**

Az osztály feladata, hogy definiálja a komponensek működését. A komponensek olyan objektumok, amik valami más objektummá összeépíthetőek. További szabály, hogy egy komponens pontosan egy építhető tárgy része lehet.

- **Attribútumok**

**-string Type:** Azonosítja egyértelműen a komponens típusát.

- **Metódusok**

- **+void assemble(ComponentMaster cc):** Utasítjuk a komponenst, hogy álljon össze a céltárgyává.

### 8.1.16 AlarmPistolComponents

- **Felelősség**

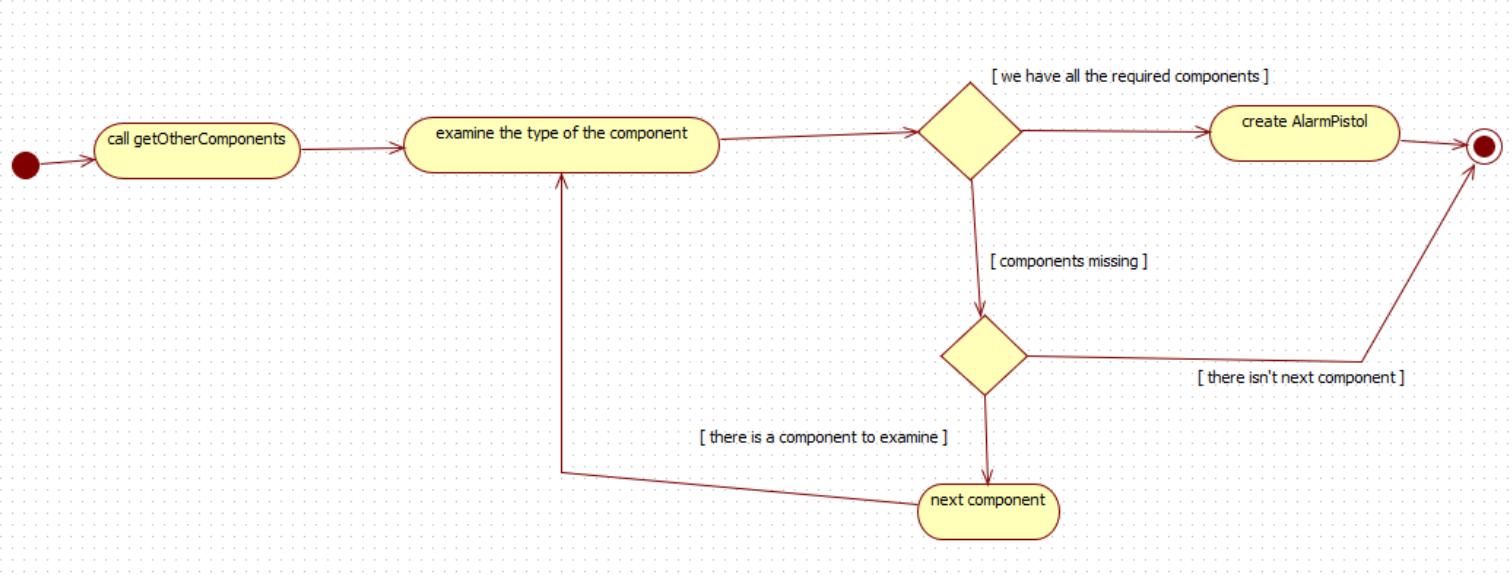
Az AlarmPistolComponents absztrakt űsosztály, gyakorlatilag egy olyan tárgyat reprezentál, ami szükséges a jelzőfényespisztoly összeszereléséhez. Felelőssége az, hogy a jelzőfényespisztoly komponenseként belőle összeállítható legyen a jelzőfényespisztoly.

- **Ősosztályok**

Components

- **Metódusok**

- **+void assemble(ComponentMaster cc):** Utasítjuk a komponenst, hogy álljon össze a céltárgyává. Ezt úgy teszi meg, hogy lekérdezi a tárgy birtokosától az összes komponensem, getOtherMasterComponents függvény, ami visszaad egy listát a komponensekkal, ezután megvizsgáljuk, hogy rendelkezésre áll-e az összes komponens a jelzőfényes pisztoly összeszereléséhez (legalább egy mindegyikből Pistol, Cartridge és Light), és ha igen, akkor létrehoz egy AlarmPistol példányt. Ha nem akkor nem történik semmi.



### 8.1.17 Pistol

- **Felelősség**

A Pistol felelőssége az, hogy a jelzőfényespisztoly komponenseként belőle összeállítható legyen a jelzőfényespisztoly.

- **Interfészek**

Pickable

- **Ősosztályok**

Components → AlarmPistolComponents

### 8.1.18 Cartridge

- **Felelősség**

A Cartridge felelőssége az, hogy a jelzőfényespisztoly komponenseként belőle összeállítható legyen a jelzőfényespisztoly.

- **Interfészek**

Pickable

- **Ősosztályok**

Components → AlarmPistolComponents

### **8.1.19 Light**

- **Felelősség**

A Light felelőssége az, hogy a jelzőfényespisztoly komponenseként belőle összeállítható legyen a jelzőfényespisztoly.

- **Interfészek**

Pickable

- **Ősosztályok**

Components → AlarmPistolComponents

### **8.1.20 AlarmPistol**

- **Felelősség**

A játék célja ezen osztály egy példányának létrehozása, ez a játék megnyerését jelenti.

- **Metódusok**

- **+alarmPistol():** Amikor létrejön ebből a típusból egy példány, az a játék győzelmet jelenti. Ezt úgy éri el, hogy a konstruktőrben meghívja a GameController endGame metódusát true értékkal paraméterezve, ami győzelmet jelenti.

### **8.1.21 ComponentMaster**

- **Felelősség**

Az interfész felelőssége, hogy az őt implementáló osztályok rendelkezzenek a szükséges műveletekkel a komponensek használatához. Tudják tárolni a komponenseket, és vissza tudják adni azokat.

- **Metódusok**

- **+void addToComponents(Component c):** Előírja annak a megvalósítását, hogy a paraméterként kapott komponenst hozzáadjuk a tárolandó komponensek listájához.
- **+void removeComponents(Component c):** Előírja annak a megvalósítását, hogy ha paraméterként kapott komponenst tárolja a függvényt implementáló mester, azt el kell dobni.

- **+Components[0..\*] getOtherMastersComponents():** Az interfész ezzel a metódussal előírja az implementáló osztályok számára annak a megvalósítását, hogy kommunikációt biztosítson az implementáló osztálynak a többi mesterrel, akiktől így elkéri használatra az ő komponenseiket.

### **8.1.22 Equipper**

- **Felelősség**

A tárgyakat felvenni képes objektumok interfésze.

- **Metódusok**

- **+void equipConsumable(Consumable cons):** Előírja a fogyasztható tárgyak felvételének megvalósítását.
- **+void equipDigger(Digger d):** Előírja az ásás működésének dinamikus megváltoztatását az interfész megvalósító osztályoknak.
- **+void equipWaterProtecter(WaterProtecter wp):** Előírja az víz ellen védés működésének dinamikus megváltoztatását az interfész megvalósító osztályoknak.
- **+void equipPullerDevice(PullerDevice p):** Előírja a húzás működésének megváltoztatását.
- **+void equipTent(t: Tent):** Előírja a felvehető sátorral rendelkezés jogának megszerzését.

### **8.1.23 DigSite**

- **Felelősség**

Az interfész felelőssége, hogy az őt megvalósító objektumok áshatóak legyenek.

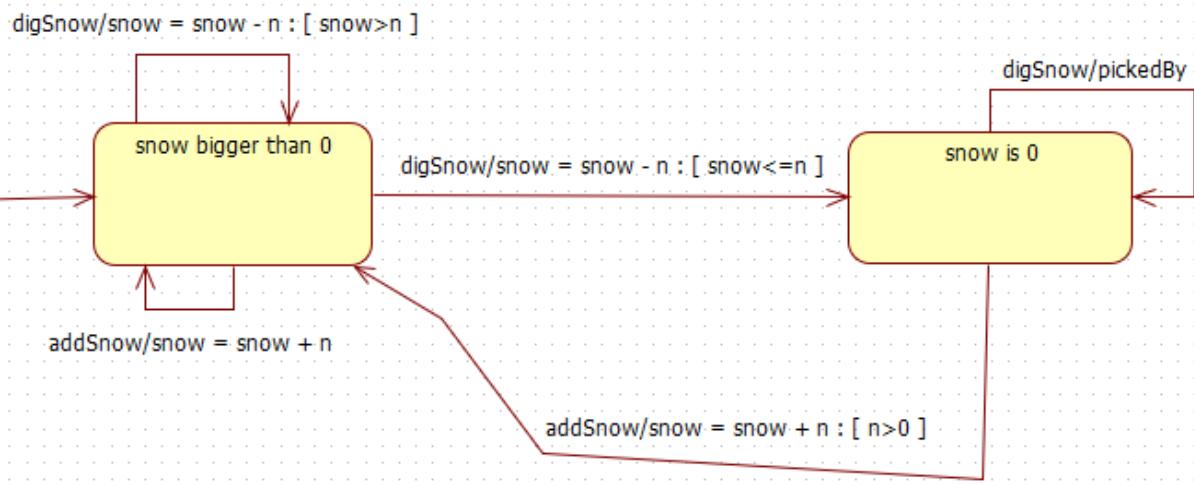
- **Metódusok**

- **+void digsnow(Character c, int n):** A metódus felelőssége, hogy biztosítja az implementáló osztályokban a működését, amikor egy karakter ás a megvalósító objektumon.

### **8.1.24 IceUnit**

- **Felelősség**

A jégtáblát reprezentáló osztály. Kezeli ezt a jégtáblát, azaz a jégtáblán levő hó mennyiségét, és azt, hogy mi helyezkedik el rajta. Az ő felelőssége, hogy meghatározza, hogy rajta milyen műveleteket lehet végrehajtani, azaz hogy hogyan lehet vele interakcióba kerülni. A sok megvalósított interfésznek köszönhetően, érheti vihar, lehet rá építeni és ásni is lehet rajta.



- Ősosztályok

- BuildingPlot

- Interfészek

- Digsite, CommunicationPlatform, Strikeable

- Attribútumok

- **-Puller[0..\*] pullers:** Tárolja egy listában a mezőn tartózkodó minden olyan objektumot, aki rendelkezik az elhúzás képességgel.
- **-Stander[0..\*] standers:** minden a mezőn tartózkodó objektum, aki Stander típusú.
- **-Pickable[0..1] pickable:** A mező által tárolt, róla felvehető objektum.
- **-int snow:** Az attribútum azt méri, hogy hány réteg hó található ezen a mezőn.
- **-IceUnit [0..\*] neighbours:** A tábla eltárolja a szomszédait.
- **-ComponentMaster[0..\*] masters:** A jégmezőn tartózkodó komponens mestereket tárolja egy listában.

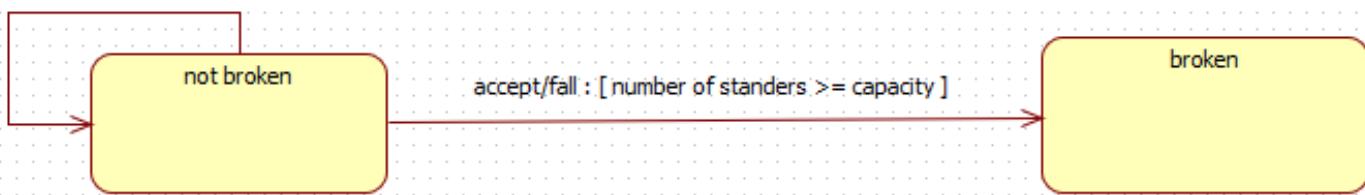
- Metódusok

- **+void accept(c: Character, from: IceUnit):** Hozzáadja a megfelelő tárolóhoz a c paramétert, jelen esetben mivel a karakter Puller, Stander és ComponentMaster is egyben, ezért az alábbi három listába rakja: pullers, standers, masters. Továbbá értesít minden az IceUniton tartózkodó Standert, hogy találkozik az éppen az IceUnitra került paraméterként kapott c objektummal, tehát végigiterál a standers nevű listán és meghívja minden egyes elemének a hitBy metódusát az IceUnitra lépett, a függvény paraméterében megkapott c objektummal paraméterezve.

- **+void remove(c: Character):** Ezzel a metódussal lehet értesíteni az IceUnitot, hogy a rajta levő Character típusú objektum elhagyta őt és ezáltal az IceUniton levő épületet is, ha abban tartózkodott. Ekkor meghívódik a mezőn levő épületnek a leave függvénye, a c objektummal paraméterezve és az IceUnit is eltávolítja a megfelelő listából (pullers, standers, masters) a c objektumot.
- **+void accept(ib: IceBear, from: IceUnit):** Hozzáadja a megfelelő tárolóhoz az ib paramétert, jelen esetben mivel a jegesmedve Stander, ezért a standers listába. Továbbá értesít minden az IceUniton tartózkodó Standert, hogy találkozik az éppen az IceUnitra került paraméterként kapott ib objektummal, tehát végigiterál a standers nevű listán és meghívja minden egyes elemének a hitBy metódusát az IceUnitra lépett, a függvény paraméterében megkapott ib objektummal paraméterezve.
- **+void remove(ib: IceBear):** Ezzel a metódussal lehet értesíteni az IceUnitot, hogy a rajta levő IceBear típusú objektum elhagyta őt. Ekkor csak annyi történik, hogy kitörli a megfelelő tárolóból (standers) a paraméterként kapott ib objektumot.
- **+IceUnit[0..\*] getNeighbours():** Visszaadja az IceUnit objektummal szomszédos járőrököt, ez azt jelenti, hogy a neighbours listával tér vissza.
- **+void removePickable():** Az eltárolt felvehető tárgy eltávolítása a metódus hatása, azaz nullra állítja a pickable attribútum értékét.
- **+void show():** A metódus hatása, hogy hozzáférhetővé válik az az információ, hogy az objektumon hány réteg hó van. Megjeleníti a snow attribútumának értékét.
- **+void digsnow(Character c, int n):** Egy karakter ás a jágtáblán. Ha a jágtáblán nincsen hó, azaz a snow tagváltozó értéke 0, akkor ha van Pickable típusú objektum a jágtáblán, akkor meghívja annak a pickedBy metódusát, saját magával és az ásást végző paraméterként megkapott karakterrel paraméterezve. Ha van hó az IceUniton, akkor a paraméterben megkapott mennyiségű havat eltávolítja a jágtábláról, azaz csökkenti a snow attribútum értékét az n paraméter értékével.

### 8.1.25 UnstableIceUnit

accept : [ number of standers < capacity ]



#### • Felelősség

Egy instabil jágtáblát reprezentáló osztály. A sima jágtáblához képest annyiban tér el, hogy egy bizonyos terhelés hatására át tud fordulni, ekkor a rajta levő dolgok a vízbe esnek.

## • Ősosztályok

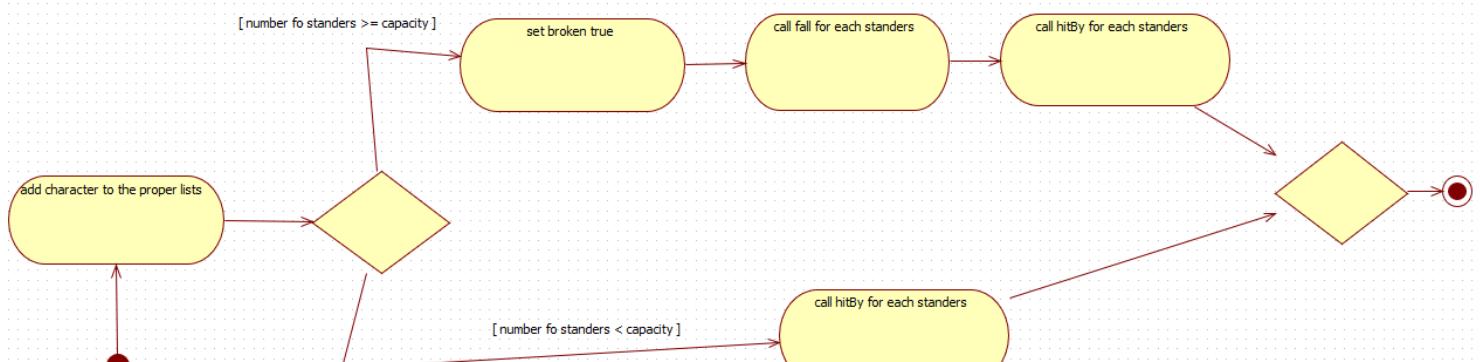
BuildingPlot → IceUnit

## • Attribútumok

- **-int capacity:** Azt jelzi, hogy mennyi Stander-t bír el a jégtábla, ha ezt az értéket átlépi, akkor az instabil jégtábla beszakad.
- **-boolean broken:** A jégtáblának két állapota lehet, beszakadadt vagy nem beszakadt. Ez a tagváltozó ezt az információt jelöli.

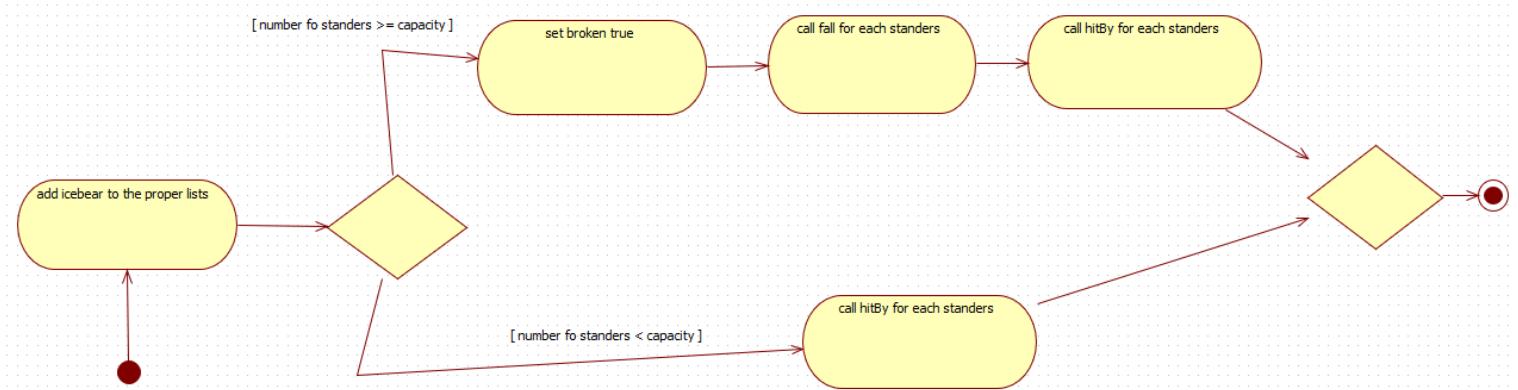
## • Metódusok

- **+void accept(c: Character, from: IceUnit):** Hozzáadja a megfelelő tárolóhoz a c paramétert, jelen esetben mivel a karakter Puller, Stander és ComponentMaster is egyben, ezért az alábbi három listába rakja: pullers, standers, masters. Ezután dönt a jégtábla beszakadásának sorsáról a terhelés és kapacitás függvényében, ez annyit jelent, hogy megvizsgálja, hogy mennyi Stander tartózkodik a mezőn és ha ez a szám nagyobb vagy egyenlő mint a capacity attribútum értéke, akkor végigiterál a standers tárolón és az összes elemének meghívja a fall metódusát. Ha nem szakadt be, akkor értesít minden az IceUniton tartózkodó Standert, hogy találkozik az éppen az IceUnitra került paraméterként kapott c objektummal, tehát végigiterál a standers nevű listán és meghívja minden egyes elemének a hitBy metódusát az IceUnitra lépett, a függvény paraméterében megkapott c objektummal paraméterezve.



- **+void accept(ib: IceBear, from: IceUnit):** Hozzáadja a megfelelő tárolóhoz az ib paramétert, jelen esetben mivel a jegesmedve Stander, ezért a standers listába. Ezután dönt a jégtábla beszakadásának sorsáról a terhelés és kapacitás függvényében, ez annyit jelent, hogy megvizsgálja, hogy mennyi Stander tartózkodik a mezőn és ha ez a szám nagyobb vagy egyenlő mint a capacity attribútum értéke, akkor végigiterál a standers tárolón és az összes elemének meghívja a fall metódusát. Ha nem szakadt be, akkor értesít minden az IceUniton tartózkodó Standert, hogy találkozik az éppen az IceUnitra került paraméterként kapott ib objektummal, tehát végigiterál a standers nevű listán és meghívja minden egyes

elemének a hitBy metódusát az IceUnitra lépett, a függvény paraméterében megkapott ib objektummal paraméterezve.



### 8.1.26 Collideable

Akik megvalósítják az interfészt, azok tudnak találkozni egy másik objektummal.

- **+void HitBy(c: Character):** A metódus feladata, hogy levezényelje azt az eseményt, amikor egy az interfészt implementáló osztály találkozik egy másik objektummal (azaz ahol az interfészt megvalósító objektum állt, arra a helyre odakerült egy karakter), jelen esetben akivel találkozik az egy Character típusú objektum.
- **+void HitBy(ib: IceBear):** A metódus feladata, hogy levezényelje azt az eseményt, amikor egy az interfészt implementáló osztály találkozik egy másik objektummal (azaz ahol az interfészt megvalósító objektum állt, arra a helyre odakerült egy jegesmedve), jelen esetben akivel találkozik az egy IceBear típusú objektum.

### 8.1.27 Stander

- **Felelősség**

Egy olyan dolgot reprezentál a játékban, ami le tud esni a jágtábláról, illetve meg tud jelenni egy jágtáblát, terhelve azt. Így ennek megfelelően a felelőssége az esés műveletének végrehajtása.

- **Interfészek**

Collideable

- **Attribútumok**

- **-IceUnit iceUnit:** Tárolja, hogy melyik jágtáblán tartózkodik.

- **Metódusok**

- **+void fall():** Ez által kap figyelmeztést, hogy a le fog esni, és levezényli az esés folyamatát.

- **+void fallenDown():** Ennek a metódusnak a meghívása azt jelzi, hogy a metódussal rendelkező dolog már nem esik, hanem leesett.

### 8.1.28 IceBear

- **Felelősség**

A jegesmedvét reprezentálja, aki megtámadhatja a karaktereket. Leszármazik a LocationChanger absztrakt osztályból, tehát le tud mozogni, esni, és tud találkozni objektumokkal. Ha leesik, akkor nem hal meg. Ha karakterekkel találkozik, azokat megtámadja, ha jegesmedvével, akkor ott nem támad.

- **Ősosztályok**

LocationChanger

- **+void HitBy(c: Character):** A metódus feladata, hogy levezényelje azt az eseményt, amikor egy jegesmedve összetalálkozik egy karakterrel, azaz arra az egységre amin a jegesmedve áll odamegy egy Character típusú objektum. Ekkor a jegesmedve lehetőséget kap arra, hogy megtámadja a karaktert. Meghívja a Character hitBy metódusát saját magával paraméterezve.

- **+void HitBy(ib: IceBear):** A metódus feladata, hogy levezényelje azt az eseményt, amikor egy jegesmedve összetalálkozik egy másik jegesmedvével, jelen esetben ekkor nem történik semmi.

• **+ void fall():** Ez által kap figyelmeztetést, hogy a le fog esni, és levezényli az esés folyamatát. Jegesmedvénél ez nem jár semmilyen következménnyel.

• **+void fallenDown():** Ennek a metódusnak a meghívása azt jelzi, hogy a metódussal rendelkező dolog már nem esik, hanem leesett. Jegesmedvénél ez szintén nem jár semmilyen következménnyel.

### 8.1.29 IceUnitManipulator

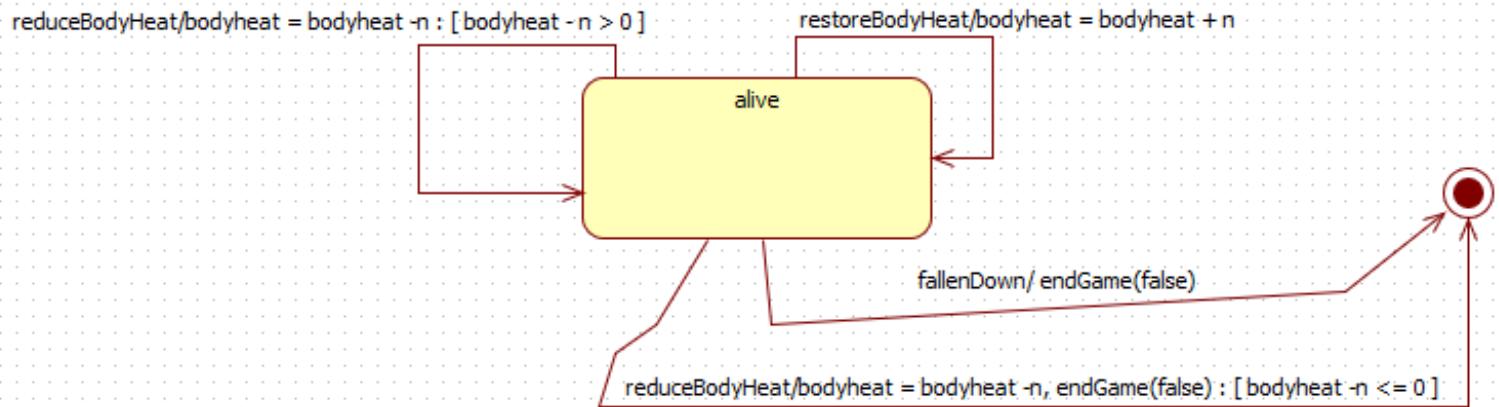
- **Felelősség**

Az interfész felelőssége, hogy biztosítja az eszközöket egy IceUnit osztályú objektumon való műveletvégzéshez.

- **Metódusok**

- **void useAbility(IceUnit iu):** A metódus, ami elvégzi a paraméterként kapott IceUniton a szükséges műveleteket.

### 8.1.30 Character



- **Felelősség**

Az osztály felelőssége, hogy a játék szabályainak megfelelően implementálja az interfészeit, határozza meg az összes irányítható karakter közös működéseit. A karakter tud műveleteket végrehajtani, amíg rendelkezik elég stamina-val, ilyen műveletek: húzás, ásás, mozgás. További dolgok is hathatnak rá, mint például esés, medvetámadás, találkozás egy másik karakterrel.

- **Ősosztályok**

Stander → LocationChanger → Mortal

- **Interfészek**

Terrestrial, ComponentMaster, Equipper, Refugee, Commandable, Puller

- **Attribútumok**

- **-int stamina:** Ennek az attribútumnak a segítségével valósítjuk azt az elvárt működést, hogy a karakter egy körben korlátozott számú műveletet végezhet. minden egyes elvégzett művelet után változik az értéke, a megfelelő mennyiséggel. A állóképesség egy adott időpillanatbeli értékét reprezentálja, mely pihenés árán (a körök végén) növekedhet.

- **-int defaultStamina:** A karakterek alapértelmezett stamina értéke, a körök végén pihenés során erre az értékre állhat vissza.

- **-WaterProtector waterProtector:** A karakter tárgya, ami megvédi ha vízbe esik.

- **-PullerDevice pullerDevice:** A karakter tárgya, ami segítségével tud arébb húzni más objektumokat.

- **-Component components:** A karakter által eltárolt komponensek. A karakterek ezek komponensekből összerakhat teljesen új dolgokat, hogy azok segítsék a játék során.

- **-Digger digger:** A karakter tárgya, ami segítségével elvégzi az ásást.

### • Metódusok

- **+void pull():** A karakter húzni fog, ez a metódus meghívja a karakter által birtokolt pullerDevice pullByDevice metódusát, azzal az IceUnit típusú objektummal paraméterezve, ahol a húzó objektum éppen elhelyezkedik.

- **+void fall():** A metódus a karakter szabályos esésének szerinti megvalósítása. Meghívja a waterProtecer objektumának a protectFromWater metódusát, saját magával paraméterezve.

- **+void reduceBodyHeat():** A metódus a szabályoknak megfelelően csökkenti a karakter testhőjét, és tesz megfelelő lépéseket a szabályoknak megfelelően, ezek a lépések: alapból a testhőt csak simán csökkenti, de ha a testhő 0-ra csökken, akkor meghívja a GameController-nek a endGame metódusát false-al paraméterezve, ami azt jelenti, hogy a játék véget ért vereséggel.

- **+void fallenDown():** A metódus megvalósítja a karakter teendőjét, ha leesett egy mezőről. Ez azt hogy a karakter meghalt, meghívja a GameController-nek a endGame metódusát false-al paraméterezve, ami azt jelenti, hogy a játék véget ért vereséggel.

- **+void HitBy(c: Character):** A metódus feladata, hogy levezényelje azt az eseményt, amikor egy karakter összetalálkozik egy másik karakterrel. Ilyenkor nem történik semmi.

- **+void HitBy(ib: IceBear):** A metódus feladata, hogy levezényelje azt az eseményt, amikor egy karakter összetalálkozik egy jegesmedvével, azaz arra az egységre amin a karakter áll odamegy egy jegesmedve típusú objektum. Ekkor a jegesmedve lehetőséget kap arra, hogy megtámadja a karaktert. Természetesen a karakter elkezd menedéket keresni, meghívja a searchShelterFromIceBear metódusát az IceUnitnak amin áll, saját magával paraméterezve.

- **+void removeTent():** Eltávolítja a karaktertől az általa birtokolt sárat.

- **+void drowning():** A karakter éppen fuldoklik a vízben, meghívja az IceUnitnak, amin tartózkodik a callForPullOut metódusát, saját magával paraméterezve, ezzel kér segítséget, hogy valaki húzza ki.

- **+void addComponents(Component c):** A függvény feladata, hogy a paraméterként kapott komponenst hozzáadjuk a tárolandó komponensek listájához.

- **+void removeComponents(Component c):** A metódus feladata, hogy amennyiben a paraméterként kapott komponenst tárolja a karakter, azt el kell dobna. Azaz el kell távolítania a listából.

- **+Components[0..\*] getOtherMastersComponents():** A metódus felelőssége, hogy kommunikációt biztosítson az implementáló osztálynak a többi mesterrel, akiktől így elkéri használatra az ő komponenseiket. Meghívja az IceUnit asdForComponents

függvényét, ami visszaad egy listát, majd ez a metódus is ezt a listát adja vissza. Tehát ez a függvény delegál csak a CommunicationPlatform interfészt megvalósító osztálynak.

- + **void tire()**: A metódus felelőssége, hogy a karakter ha valamilyen munkát végezett, akkor fáradjon, csökkenjen a stamina attribútumának értéke.
- + **void rest()**: A karakter képes a pihenésre, ekkor a stamina értéke a defaultStamina értékét veszi fel.

### 8.1.31 Explorer

- **Felelősség**

Az osztály felelőssége megvalósítani az egyik játszható karaktertípus, a felfedező működését.

- **Ősosztályok**

Stander → LocationChanger → Mortal → Character

- **Interfészek**

`IceUnitManipulator`

- **Metódusok**

- +**void useAbility(IceUnit iu)**: Megméri, hogy mennyit bír el a paraméterként kapott mező: meghívja a mező show függvényét.

### 8.1.32 Eskimo

- **Felelősség**

Az osztály felelőssége megvalósítani az egyik játszható karaktertípus, az eszkimó működését.

- **Ősosztályok**

Stander → LocationChanger → Mortal → Character

- **Interfészek**

`IceUnitManipulator`

- **Metódusok**

- +**void useAbility(IceUnit iu)**: Iglut épít a kijelölt mezőre: meghívja a létrehoz egy új Igloo példányt és ezzel paraméterezve meghívja az `IceUnit build` metódusát.

### 8.1.33 Puller

- **Felelősség**

Az interfész felelőssége definiálni, hogy az implementáló objektum rendelkezik az elhúzás képességevel.

- **Metódusok**

- **+ void pull(Pullable p):** A metódus célja, hogy a paraméterként kapott Pullable objektumot elhúzza magához.

### 8.1.34 LocationChanger

- **Felelősség**

Egy olyan doleg, ami képes a helyváltoztatásra.

- **Ősosztályok**

Stander

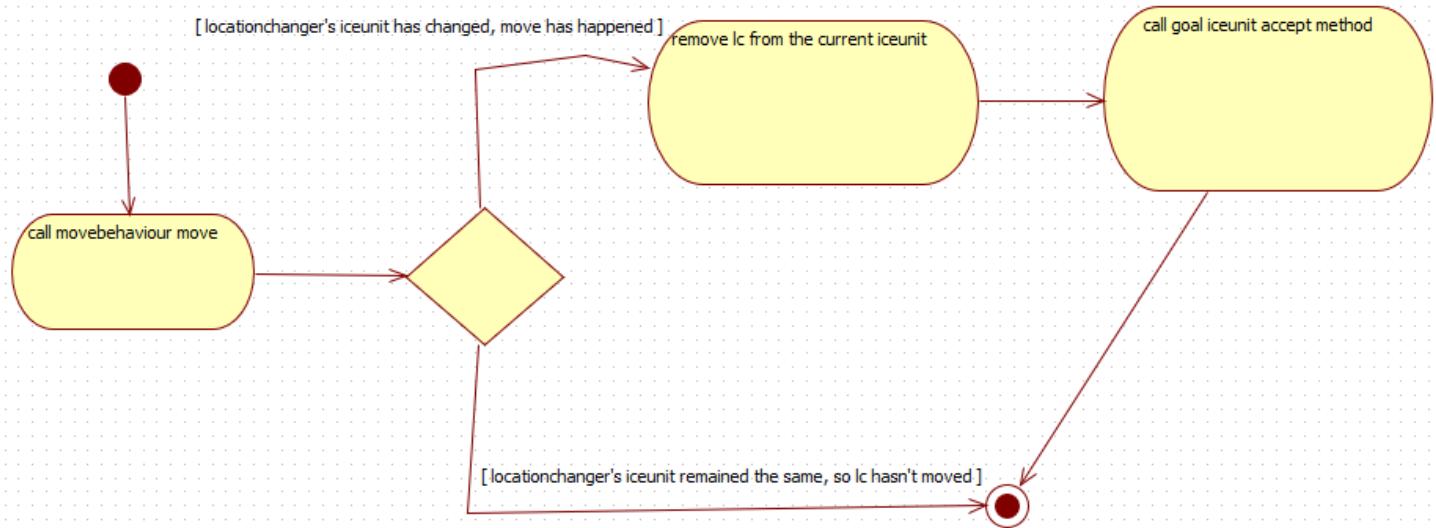
- **Interfészek**

FiniteActionDoer

- **Attribútumok**

- **-MoveBehaviour behaviour:** Tárolja a LocationChanger példány mozgásstílusát.

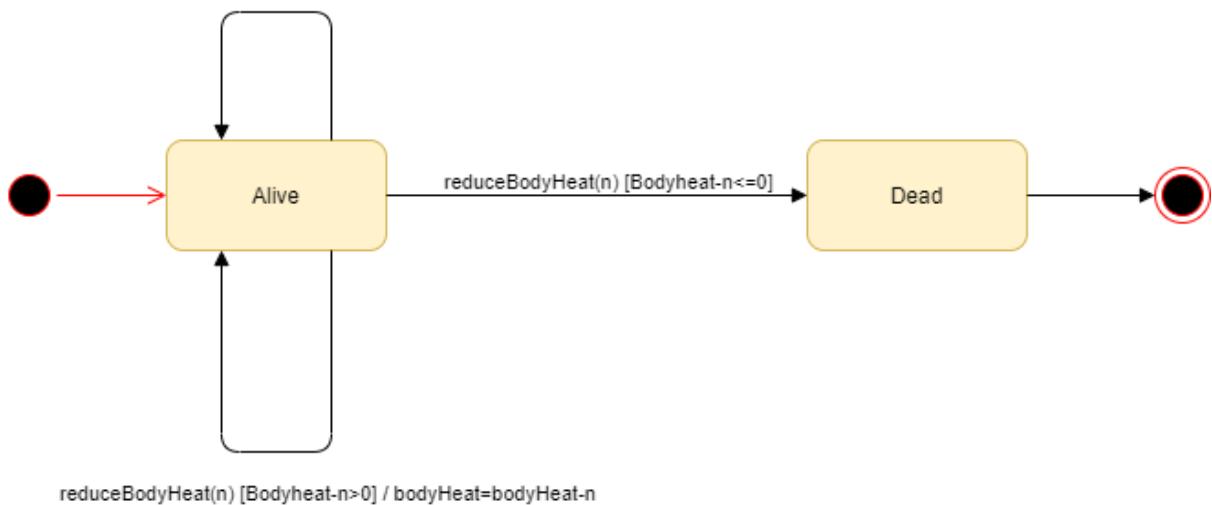
- **Metódusok**



- **+void move(IceUnit goal, IceUnit from):** A LocationChanger mozog, ez annyit jelent, hogy meghívjuk a mozgást végző tagváltozó (behaviour) move metódusát, a cél hellyel, a jelenlegi tartozkodási hellyel és saját magával paraméterezve. Ez a meghívott metódus ha szabályosnak ítéli meg a lépést, akkor megváltoztatja a tartózkodási helyét a LocationChangernek, innen lehet tudni, hogy ez egy szabályos lépés volt, ha pedig a lépés szabályos volt, akkor a metódus eltávolítja a LocationChangert arról a helyről, ahonnan készül ellépni, ennek az objektumnak meghívja a remove metódusát saját magával paraméterezve, majd ezután meghívja a cél mezőnek az accept metódusát, szintén saját magával paraméterezve.
- **+void changeMoveBehaviour(MoveBehaviour mb):** Beállítjuk a példány mozgásstílusát a paraméterként kapottra.

### 8.1.35 Mortal

`restoreBodyHeat(n) [Bodyheat+n<max] / bodyHeat=bodyHeat+n`



- **Felelősség**

Az osztály felelőssége az olyan jégtáblákon mozgásra képes objektumok modellezése, akik rendelkeznek testhővel.

- **Ősosztályok**

2. Stander → LocationChanger

- **Attribútumok**

- **-int bodyHeat:** A testhő nagyságának értékét jelölő szám.
- **-int maxBodyHeat:** A testhő nagyságának maximumát jelöli.

- **Metódusok**

- **+ void reduceBodyHeat(int n):** A paraméterben kapott értékkel csökkenti a testhő (bodyHeat) nagyságát.
- **+ void restoreBodyHeat(int n):** A paraméterben kapott értékkel növeli a testhő (bodyHeat) nagyságát, de bodyHeat attribútum értéke sosem lehet nagyobb, mint a maxBodyHeat.

### 8.1.36 Terrestrial

- **Felelősség**

Az interfész felelőssége, hogy működési felületet biztosítson azoknak az objektumoknak, akiknek káros vízben tartózkodniuk.

- **Metódusok**

- + **void drowning()**: A metódus felelőssége, hogy jelezze a megvalósító példánynak, hogy éppen fullad a vízben.

### 8.1.37 FiniteActionDoer

- **Felelősség**

Az interfész felelőssége, hogy definiálja a működését azoknak az objektumoknak, amelyek korlátozott cselekvőképességgel rendelkeznek.

- **Metódusok**

- + **void tire()**: A metódus felelőssége, hogy jelezzük vele, hogy a megvalósító objektumnak valamely mértékben csökkennie kell a további munkavégző képességének.
- + **void rest()**: Előírja, hogy a metódust implementáló osztályok képesek a pihenésre.

### 8.1.38 MoveBehavior

- **Felelősség**

Az interfész felelőssége, hogy megadja a keretet különböző mozgásformák megvalósításához.

- **Metódusok**

- + **void move(IceUnit goal, IceUnit from, LocationChanger lc)**: A metódus felelőssége, hogy megvalósítja az elmozdulását a paraméterként átadott LocationChanger típusú objektumnak a cél és kiinduló jégmező között.

### 8.1.39 Walk

- **Felelősség**

Olyan mozgásforma, aminek követésével mozoghatunk szárazföldön.

- **Interfész**

MoveBehaviour

- **Metódusok**

**void move(IceUnit goal, IceUnit from, LocationChanger lc)**: Ha a goal IceUnit broken változója false, akkor engedélyezi a mozgást. Ezt a függvényt a LocationChanger egyik leszármazottjának move függvénye hívja meg. Ha ez a mozgás nem engedélyezett, akkor a goal értékét fromra állítja, különben nem avatkozik bele a mozgásba.

### 8.1.40 Amphibious

- **Felelősség**

Olyan mozgásforma, aminek követésével mozoghatunk bármely mezőre, legyen az víz vagy föld.

- **Interfészek**

MoveBehaviour

- **Metódusok**

**void move(IceUnit goal, IceUnit from, LocationChanger lc):** Ha a goal IceUnit broken változója true, akkor engedélyezi a mozgást. Ezt a függvényt a LocationChanger egyik leszármazottjának move függvénye hívja meg. Ha ez a mozgás nem engedélyezett, akkor a goal értékét fromra állítja, különben nem avatkozik bele a mozgásba.

### 8.1.41 Pullable

- **Felelősség**

Egy olyan dolog, amiket a játékban, akár más közreműködő segítségével képes húzni.

- **Ősosztályok**

Stander

- **Metódusok**

- + **void reposition(IceUnit iu):** Megkapja paraméterben, hogy melyik jégtáblára húzták ki. Ezután arra a jégtáblára “mozog”.

### 8.1.42 CommunicationPlatform

- **Felelősség**

Az interfész felelőssége, hogy az azt implementáló játékbeli mezők rendelkezzenek minden metódussal, ami a különböző objektumok egymással való kommunikációjához szükséges egy mezőn keresztül.

- **Metódusok**

- + **Component[0..\*] askForComponents():** A metódus felelőssége az, hogy visszaadja a mezőn létező objektumok komponenseinek összességét.

- + void **callForPullOut(Pullable p)**: A metódus meghívásával jelezzük az összes szomszédos mezőnek, hogy a rajtuk lévő objektumoknak ki kell húzniuk a paraméterként átadott objektumot.

- + void **alertNeighbourForPullOut(Pullable p)**: Ezzel a metódussal a mező figyelmeztetést kap, hogy a rajta álló Puller objektumoknak ki kell húzniuk a paraméterként átadott objektumot.

### **8.1.43 BuildingPlot**

- **Felelősség**

A jégmezőn levő épülethelyeket kezelő absztrakt osztály. Épülethelyeken építhető valamelyen épület például iglu. Ha nincs semmi az épülethelyen, akkor minden „sík” épület van ott.

- **Attribútumok**

- -Building **building**: Tárolja a jégmezőn lévő épületet.

- **Metódusok**

- +void **build(Building b)**: Az épülethelyen felépül egy épület. A függvény átállítja a building attribútumot síkról a paraméterben kapott b épületre.
- +void **destructBuilding()**: Szétesik az épület az épülethelyen, ez miatt átállítja a building attribútumot a függvényen belül létrehozott síkra (Plains osztály).
- +searchShelter(Refugee r): Az épülethelyen a paraméterben kapott menekült keres menedéket egy épületben. Itt absztrakt függvény, a leszármazottaknak kell megvalósítani.
- +void **searchShelterFromIceBear(Refugee r)**: Az épülethelyen a paraméterben kapott menekült keres menedéket egy épületben, egy jegesmedve elől menekülve. Itt absztrakt függvény, a leszármazottaknak kell megvalósítani.

### **8.1.44 StormyField**

- **Felelősség**

A játékban lévő viharos jégmezőknek az interfésze.

- **Metódusok**

- +void **unleashStorm()**: Végig megy a vihar a pályán(mi esetünkben a jégmezőn).

### **8.1.45 IceField**

- **Felelősség**

Tárolja, hogy milyen egységekből épül fel a pálya, és az egységeket vezérli.

- **Interfészek**

StormyField

- **Attribútumok**

- **-IceUnit[1..\*] iceUnits:** A jégmező jégtáblái.
- **-Strikeable[1..\*] strikeables:** minden olyan objektum a játékban, amire hathat katasztrófa.
- **-Stander[1..\*] standers:** minden egyes jégtáblán létező objektum.

- **Metódusok**

- **+void unleashStorm():** A StormyField interfész metódusának implementációja, a függvény hatására sok strikable interfészű objektumot a vihar hatásaival sújtja. Először létrehoz egy Strikeable listát, amibe benne lesz az összes karakter és random valamennyi jégtábla, ezt a strikeables attribútumból szedi össze. Ezután létrehoz egy Storm példányt, majd meghívja a storm példánynak a strikeStrikeable metódusát az újonnan létrehozott strikeable listával.

### **8.1.46      Storm**

- **Felelősség**

A vihart reprezentáló osztály. Ez az osztály egy katasztrófa, aminek a felelőssége az, hogy megcsapja viharral a jégmezőt.

- **Interfészek**

Disaster

- **Metódusok**

- **+void strikeStrikeable(Strikeable[1..\*] s):** A vihar lesújt a paraméterben kapott lecsapható dolgokra. Végig megy a paraméterben kapott s listában és meghívja a listában lévő összes objektumnak a struckByDisaster metódusát.

### **8.1.47      Disaster**

- **Felelősség**

A játékban lévő katasztrófáknak az interfésze.

- **Metódusok**

- **+void strikeStrikeable(Strikeable[1..\*] s):** A katasztrófa lesújt a paraméterben kapott katasztrófa által elkapható dolgokra.

### **8.1.48      Strikeable**

- **Felelősség**

Katasztrófa által sújtható dolgok.

- **Metódusok**

- **+void struckByDisaster():** Katasztrófa sújtja az ezt a függvényt megvalósító objektumot.

### **8.1.49      Refugee**

- **Felelősség**

A játékban lévő menekülteknek az interfésze. Az interfész megvalósítói a játékban épületekbe elbújni képes különböző dolgok.

- **Ősosztályok**

Strikeable

- **Metódusok**

- **+void protectionFailed():** Ha a menekültnek nincsen fedezéke a katasztrófák ellen, akkor nincsen neki protekciója és testhőt veszít.
- **+void protectionFromBearFailed():** Ha a menekültnek nincsen fedezéke a jegesmedve ellen, akkor nincsen neki protekciója és meghalhat.

### **8.1.50      Building**

- **Felelősség**

Az épületeket kezelő absztrakt osztály. Az épület egy épülethelyen építhető fel és van valamennyi max kapacitása.

- **Attribútumok**

- **-BuildingPlot plot:** Eltárolja, hogy az épület melyik épülethelyen található.
- **-Refugee[0..\*] refugees:** Eltárolja az igluban lévő menekülteket.
- **-int capacity:** Eltárolja azt, hogy mennyi menekült lehet az igluban.

- **Metódusok**

- **+void getIn(Refugee r):** Bemegy az épületbe a paraméterben kapott menekült. Először megnézzük, hogy a menekült már benne van-e az épületben. Ez lehetséges, mert csak akkor hagyja el, amikor lelép a jágtábláról. Végig iterálunk a refugees listán és megnézzük, hogy melyiknek a referenciajára egyenlő a paraméterben kapott menekültével. Ha valamelyikkel megegyezik, akkor nem kell semmit sem csinálnunk, viszont, ha nem, akkor megkell nézni, hogy van-e még hely az épületben. Ha tele van az épület, akkor nem tud bemenni, tehát ha a capacity egyenlő a refugees lista nagyságával, akkor meghívódik a menekült protectionFailed függvénye.
- **+void struckByStorm():** A vihar elkapja az épületet és az épület típusától függően történik valami az épülettel. Itt absztrakt függvény, a leszármazottaknak kell megvalósítani.
- **+void getInFromIceBear(Refugee r):** Bemegy az épületbe a paraméterben kapott menekült. Először megnézzük, hogy a menekült már benne van-e az épületben. Ez lehetséges, mert csak akkor hagyja el, amikor lelép a jágtábláról. Végig iterálunk a refugees listán és megnézzük, hogy melyiknek a referenciajára egyenlő a paraméterben kapott menekültével. Ha valamelyikkel megegyezik, akkor nem kell semmit sem csinálnunk, viszont, ha nem, akkor megkell nézni, hogy van-e még hely az épületben. Ha tele van az épület, akkor nem tud bemenni, tehát ha a capacity egyenlő a refugees

lista nagyságával, akkor meghívódik a menekült protectionFromBearFailed függvénye.

- **+void leave(Refugee r):** Mivel a karakter átlép egy másik mezőre, ha esetleg benne volt az épületben, akkor kimegy. Végig iterálunk a refugees listán és megnézzük, hogy melyiknek a referenciajára egyenlő a paraméterben kapott meneküttel. Ha egyenlő valamelyikkkel, akkor kiveszszük a listából, ha nem, akkor nincs benne az épületben, nem csinálunk semmit.

### **8.1.51      Plains**

- **Felelősség**

Egy épület, ahol nincs semmi. Alapból minden jégtáblán „sík épület” van, és kicsérélődik, ha építének rá egy másik épületet, illetve erre cserélődik, ha szétesik egy épület.

- **Ősosztályok**

Building

- **Metódusok**

- **+void getIn(Refugee r):** Felülírja az ősosztály függvényét. A síkon a paraméterben kapott menekült nem tud sehol bemenekülni a vihar elől. Ezért mindenkor veszít testhőt, vagyis azonnal meghívódik a menekült protectionFailed metódusa.
- **+void struckByStorm():** A vihar nem tudja lerombolni a síkot, tehát nem történik semmi, de kell hogy meglegyen a függvény, mert az ősosztályban a függvény absztrakt.
- **+void getInFromIceBear(Refugee r):** Felülírja az ősosztály függvényét. A síkon a paraméterben kapott menekült nem tud sehol bemenekülni a jégmedve elől, mindenkor megfog halni. Meghívódik a menekült protectionFromBearFailed függvénye.

### **8.1.52      DestructableBuilding**

- **Felelősség**

Az elpusztítható épületeket kezelő absztrakt osztály.

- **Ősosztályok**

Building

- **Attribútumok**

- **-int health:** Tárolja az elpusztítható épületnek az életét.

- **Metódusok**

- **+void getInFromIceBear(Refugee r):** Felülírja az ősosztály függvényét. Nem csinál semmit a függvény, azért kell, hogy a leszármazott osztály is felül tudja írni (a Tent osztály felülírja, mert ott nincs protekció a medve ellen).

- **+void struckByStorm()**: A vihar elkapja az épületet. Az elpusztítható épületeknél ez azt jelenti, hogy csökkenti az életüket, vagy is meghívódik az osztály decreaseHealth függvénye.
- **+void decreaseHealth()**: Csökkenti az életerőt eggyel. Ha az életerő lecsökken nullára, akkor lekéri a Building osztályból a plot attribútumot, és meghívja az attribútumnak a destructBuilding metódusát.

### **8.1.53 Igloo**

- **Felelősség**

A játékban levő iglu elpusztítható épületet valósítja meg.

- **Ósosztályok**

Building → DestructableBuilding

### **8.1.54 Commandable**

- **Felelősség**

Az interfész felelőssége, hogy az irányítható karakterekhez egy irányítási felületet biztosítson.

- **Metódusok**

- **+void executeCommand(String s)**: A paraméternek megfelelő parancs végrehajtása.

### **8.1.55 IceUnit**

- **Metódusok**

- **+void struckByDisaster()**: Elkapja a katasztrófa az jégtáblát, és a rajta lévő épületet. Lekérjük a BuildingPlot osztály building attribútumát, majd meghívjuk a struckByStorm függvényét.
- **+void searchShelter(Refugee r)**: A jégtáblán a paraméterben kapott menekült keres menedéket egy épületben. Lekéri a BuildingPlot building attribútumát, majd meghívja a getIn metódusát a menekülltel paraméterevezve.
- **+void searchShelterFromIceBear(Refugee r)**: A jégtáblán a paraméterben kapott menekült keres menedéket egy épületben a jegesmedve elől. Lekéri a BuildingPlot building attribútumát, majd meghívja a getInFromIceBear metódusát a menekülltel paraméterevezve.

### **8.1.56 GameController**

- **Felelősség**

A játék indításáért, irányításáért, befejezéséért és az alapvető működéséért felel ez az osztály. Az ő feladata, hogy a karakterek egymás után körökre osztva tevékenykednek. Az ő feladata jelezni a katasztrófáknak, hogy olyan időpillanatba jutott a játék, hogy bekövetkezésük szabályos lenne, illetve felel azért is, hogy a sátrak eltűnjenek a kör végén. Statikus osztály, minden függvénye és adattagja külön jelzés nélkül is statikus.

- **Attribútumok**
  - **-StormyField[0..\*] disastrousFields:** Eltárolja azokat a pályákat, amiket érhet vihar.
  - **-Commandable [1..\*] commandables:** Az irányítható dolgok összesége.
- **Metódusok**
  - **+void startGame():** A metódus feladata a játék inicializálása, megfelelő állapotba hozása a játék elkezdéséhez.
  - **+void endGame(boolean isWin):** A metódus feladata, hogy a paraméterként kapott változó alapján megfelelően befejezze a játékot. Ha isWin az igaz, akkor nyert a játékos és kiírjuk neki, hogy „Game Won!”, ha nem igaz, akkor vesztett és kiírjuk, hogy „Game Over!“.
  - **+void nextCommandable():** A metódus feladata, hogy a segítségével egy másik Commandable objektumnak adjunk lehetőséget arra, hogy valamilyen akciót hajthasson végre.
  - **+void addDisastrousNotifiable(DisastrousField df):** A metódus feladata, hogy a kapott paramétert eltároljuk az ismert katasztrófával sújtott pályák között, tehát hozzáadjuk a df paramétert a disastrousFields listához.
  - **+void notifyDisastrouses():** A katasztrófával sújtott pályák értesítése, hogy a játék olyan szakaszba ért, ahol a szabályok szerint megtörténhet valamilyen katasztrófa (a jelenlegi követelmények szerint ez a katasztrófa a hóvihar). Végig iterálunk az összes pályán, azaz a disastrousFields listán és meghívja mindegyiknek az unleashStorm metódusát.

### **8.1.57      CommandReader**

- **Felelősség**

A felhasználóval tartja a kapcsolatot a játék során. Ez az osztály tartalmazza a játék irányítására használható parancsokat, amiket értelmez és végrehajt. Statikus osztály, minden függvénye és adattagja külön jelzés nélkül is statikus.

- **Metódusok**

- **- void start():** Letiltja a pályaépítést szolgáló parancsok használatát.
- **- void end():** Engedélyezi a pályaépítést szolgáló parancsok használatát.
- **- void loadmap(String name):** Betölt egy pályát egy fájlból.
- **- void list():** Kiírja a pályán lévő objektumok azonosítóját a szabványos kimenetre.

- - **void stats(string id):** Kiírja egy objektum tulajdonságait a szabványos kimenetre.
- - **void stats():** Kiírja az összes objektum tulajdonságait a szabványos kimenetre.
- - **void move(String fieldID, String entityId):** Entitás léptetése.
- - **void move(String fieldID):** Soron következő entitás léptetése.
- - **void ability(String fieldID, String entityID):** Kerekterképesség használata.
- - **void ability(String fieldID):** Soron következő kerekter képességének használata.
- - **void dig(String entityID):** Ásás karakterrel.
- - **void dig():** Ásás a jelenlegi karakterrel.
- - **void tent(String entityID):** Sátor lerakása.
- - **void tent():** Sátor lerakása a jelenlegi karakterrel.
- - **void storm(String fieldID):** Hóvihar indítás mezőn.
- - **void storm(String entityID):** Hóvihar indítás karakteren.
- - **void assemble(String entityID):** Karakter utasítása pisztoly összerakására.
- - **void turn():** Átadja a kört a következő irányítható objektumnak.
- - **void current():** Kiírja annak az objektumnak az azonosítóját, aminek jelenleg a köre van.
- - **void field(String fieldID, int snow):** Stabil jégtábla létrehozása.
- - **void field(String fieldID, int capacity, boolean broken, int snow):** Instabil jégtábla létrehozása.
- - **void neighbours(String fieldID, List<String> neighbourIDs):** Mező szomszédainak beállítása

- - **void building(String buildingID, String type, String fieldID):** Épület létrehozása.
- - **void entity(String entityID, String type, String fieldID):** Entitás létrehozása.
- - **void savemap(String name):** Jelenlegi pálya elmentése egy fájlba.
- + **void execute(String command):** Parancs értelmezése és futtatása.

## 8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

### 8.2.1 Egy jégtábla betöltésének ellenőrzése

- **Leírás**

A teszt célja annak ellenőrzése, hogy a jégmezőt alkotó jégtábla adatai kimentve, megfelelően visszaolvashatóak, és a beolvasható adatok konzisztensek a kiírt adatokkal.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az egy jégtáblából álló jégmező elmentésének és visszaolvasásának tesztje. Várható hibák, hogy az visszaolvasott objektum a kiírt objektummal inkonzisztens, tehát különbözőek a tulajdonságaik. Ilyen tulajdonság lehet például a teherbíró képesség, a jégtáblán tartózkodó objektumok (standerek, pullerek, componentMasterek), a töröttség állapotát tároló tagváltozó, épület a jégtáblán, a jégtáblába fagyott tárgy.

- **Bemenet**

```
field f1 5 false 3
entity e1 Eskimo f1
building b1 Tent f1
item i1 Pistol setfield f1
savemap testmap_1
loadmap testmap_1
stats f1
```

- **Elvárt kimenet**

```
id: f1
snow: 3
capacity: 5
broken: false
building: b1
pickable: i1
standers: e1
pullers: e1
masters: e1
neighbours: null
```

### 8.2.2 Jégmező betöltésének ellenőrzése

- **Leírás**

A teszt célja annak az ellenőrzése, hogy amennyiben bonyolultabb, több jégtáblából álló pályát töltünk be, akkor abban az esetben a pálya szerkezete, a jégtáblák képessége, hogy elmentse szomszédjait, helyes módon megőrződik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt ellenőrzi, hogy a pálya szerkezete, a szomszédos mezők elmentődnek megfelelően. Várható hibák például, hogy jégtáblák visszatöltés után már nem érik el a kimentés előtt beállított szomszédjaikat, vagy elérnek olyan szomszédot is, amit nem kellene.

- **Bemenet**

```
field f1 5 false 3
field f2 6 false 2
field f3 0
neighbours f1 f2 f3
neighbours f2 f1
neighbours f3 f1
savemap testmap_2
loadmap testmap_2
stats f1
stats f2
stats f3
```

- **Elvárt kimenet**

```
id: f1
snow: 3
capacity: 5
broken: false
building: plains1
pickable: null
standers: null
pullers: null
masters: null
neighbours: f2, f3
```

```
id: f2
snow: 2
capacity: 6
broken: false
building: plains2
pickable: null
standers: null
pullers: null
masters: null
neighbours: f1
```

```
id: f3
snow: 0
building: plains3
pickable: null
standers: null
pullers: null
masters: null
neighbours: f1
```

### **8.2.3 Jegesmedve mozgásának az ellenőrzése**

- **Leírás**

A jegesmedvék az egyik mozgásra képes objektumok a játékban, a tesztben ellenőrizzük, hogy mozgásuk megfelel a követelményeknek. A jegesmedve kételőként mozog, tehát tudnia kell bármilyen tulajdonságú mezőön járnia; legyen az szárazföld, vagy víz. Emiatt a jegesmedvét szárazföldről indítva, mozgatjuk szárazföldre, onnan vízbe, onnan megint vízre, majd szárazföldre ismét. Ezzel lefedjük az összes mozgási lehetőséget.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Várható hibajegy például, ha nem sikerül mozognia a jegesmedvének bármely esetben. Ezt onnan látjuk a kimeneten, hogy a mozgási parancsra válaszul a helyzete nem változott.

- **Bemenet**

```
field spawn 0
field dry1 3
field wet1 5 true 0
field wet2 3 true 2
field dry2 0
neighbours spawn dry1
neighbours dry1 wet1
neighbours wet1 wet2
neighbours wet2 dry2
entity ib1 Bear spawn
stats ib1
move ib1 dry1
stats ib1
move ib1 wet1
stats ib1
move ib1 wet2
stats ib1
move ib1 dry2
stats ib1
```

- **Elvárt kimenet**

```
id: ib1
iceUnit: spawn
```

```
id: ib1
iceUnit: dry1
```

```
id: ib1
iceUnit: wet1
```

```
id: ib1
iceUnit: wet2
```

```
id: ib1
iceUnit: dry2
```

## 8.2.4 Karakter szárazföldi mozgásának ellenőrzése

- **Leírás**

A karakterek az egyik mozgásra képes objektumok a játékban, a tesztben ellenőrizzük, hogy mozgásuk megfelel a követelményeknek. A karakterek kételőként vagy szárazföldiként

mozoghatnak, ebben a tesztben azt vizsgáljuk, ha szárazföldi mozgásformával rendelkeznek. Megvizsgáljuk, hogy a karakter, a tesztben egy eszkimó, sikeresen tud-e szárazföldről szárazföldre mozogni, majd ellenőrizzük, hogy ha vízbe próbáljuk mozgatni, az meghiúsul-e.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Várható hibajegy például, ha tévesen nem tudunk elmozdulni a szárazföldi mezőre; vagy ha sikerül belemennünk a karakterrel a vízbe.

- **Bemenet**

```
field spawn 0
field dry 0
field wet 5 true 0
neighbours spawn dry
neighbours dry wet
entity esk1 Eskimo spawn
stats esk1
move esk1 dry
stats esk1
move esk1 wet
stats esk1
```

- **Elvárt kimenet**

```
id: esk1
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: esk1
iceUnit: dry
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: esk1
iceUnit: dry
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
```

components: null  
moveBehaviour: walk1

### **8.2.5 Karakter kétéltű mozgásának ellenőrzése**

- **Leírás**

A karakterek az egyik mozgásra képes objektumok a játékban, a tesztben ellenőrizzük, hogy mozgásuk megfelel a követelményeknek. A karakterek kétéltüként vagy szárazföldiként mozoghatnak, ebben a tesztben azt vizsgáljuk, ha kétéltű mozgásformával rendelkeznek. Megvizsgáljuk, hogy a karakter, a tesztben egy eszkimó, sikeresen tud-e szárazföldről szárazföldre mozogni, szárazföldről vízbe, vízből szárazföldnek.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Várható hibajegy, ha akármelyik elmozdulás meghiúsul, hiszen a kétéltű mozgással rendelkező karakterek képesnek kell lennie a tesztelt elmozdulásokat minden végrehajtani.

- **Bemenet**

```
field spawn 0
field dry1 3
field wet1 5 true 0
field wet2 3 true 0
field dry2 0
neighbours spawn dry1
neighbours dry1 wet1
neighbours wet1 wet2
neighbours wet2 dry2
entity esk1 Eskimo spawn
moveBehaviour amph1 Amphibious esk1
stats esk1
move esk1 dry1
stats esk1
move esk1 wet1
stats esk1
move esk1 wet2
stats esk1
move esk1 dry2
stats esk1
```

- **Elvárt kimenet**

```
id: esk1
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: amph1
```

```
id: esk1
iceUnit: dry1
```

```
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: amph1
```

```
id: esk1
iceUnit: wet1
bodyHeat: 5
stamina: 2
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: amph1
```

```
id: esk1
iceUnit: wet2
bodyHeat: 5
stamina: 1
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: amph1
```

```
id: esk1
iceUnit: dry2
bodyHeat: 5
stamina: 0
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: amph1
```

### **8.2.6 Instabil jégtábla teherbíró képességének, és betörésének vizsgálata**

- **Leírás**

A teszt feladata, hogy ellenőrizze, hogy az instabil jégtáblák betörési mechanizmusa megfelelő módon történik. Ezt úgy fogjuk letesztni, hogy elhelyezünk fokozatosan jegesmedvéket egy ilyen instabil jégtáblán, és megvizsgáljuk, hogy a megfelelő pillanatban török-e be, tehát a kapacitásának elérésekor.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt során hiba lehet, ha a jégtábla nem a definiált kapacitás elérése esetén fog betörni; hiba a kapacitás elérésénél korábban, és hiba, ha később következik be a betörés.

- **Bemenet**

field b1 3 false 0

stats b1

entity ib1 Bear b1

stats b1

entity ib2 Bear b1

stats b1

entity ib3 Bear b1

stats b1

entity ib4 Bear b1

stats b1

entity ib5 Bear b1

stats b1

- **Elvárt kimenet**

id: b1

snow: 0

capacity: 3

broken: false

building: plains1

pickable: null

standers: null

pullers: null

masters: null

neighbours: null

id: b1

snow: 0

capacity: 3

broken: false

building: plains1

pickable: null

standers: ib1

pullers: null

masters: null

neighbours: null

id: b1

snow: 0

capacity: 3

broken: false

building: plains1

pickable: null

standers: ib1, ib2

pullers: null

masters: null

neighbours: null

id: b1

```
snow: 0
capacity: 3
broken: true
building: plains1
pickable: null
standers: ib1, ib2, ib3
pullers: null
masters: null
neighbours: null
```

```
id: b1
snow: 0
capacity: 3
broken: true
building: plains1
pickable: null
standers: ib1, ib2, ib3, ib4
pullers: null
masters: null
neighbours: null
```

```
id: b1
snow: 0
capacity: 3
broken: true
building: plains1
pickable: null
standers: ib1, ib2, ib3, ib4, ib5
pullers: null
masters: null
neighbours: null
```

### **8.2.7 Vizet reprezentáló jégtábla teherbíró képességének, és betörésének vizsgálata**

- **Leírás**

A tesztnben a vizes mezőket, tehát gyakorlatilag a betört jégtábla mezőket fogjuk tesztelni, hogy azok hogyan reagálnak a rajtuk tartózkodó objektumok terhelésére. A tesztnben létrehozunk egy betört instabil jégtáblát. Ezután fokozatosan karaktereket, eszkimót fogunk elhelyezni ezen a jégtáblán, akik nem rendelkeznek vízi védelemmel. Annyi ilyen eszkimót kell elhelyeznünk, hogy elérjük a kapacitását a jégtáblának, és azon még valamennyivel túl is menjünk karakterszámban.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az teszt célja annak az ellenőrzése, hogy a betört jégtábla nem törhet be újra. A tesztnben terheljük a jégtáblát eszkimókkal, akik nem rendelkeznek vízi védelemmel. Ezeknek az eszkimóknak csak akkor hívódik meg az esés metódusa, ha betörök a jégtábla. Mivel ezek az eszkimók nem rendelkeznek vízi védelemmel, ezért ha betörne a már betörött jégtábla, ezek a karakterek meghalnának, és a Game Over felirat megjelenne a kimeneten. Tehát ez az egyik hibahely. További hiba lehet még, ha nem jól érzékeli a betört, instabil jégtábla a ráérkező karaktereket.

- **Bemenet**

```
field unstable 2 true 0
stats unstable
entity e1 Eskimo unstable
entity e2 Eskimo unstable
entity e3 Eskimo unstable
entity e4 Eskimo unstable
stats unstable
```

- **Elvárt kimenet**

```
id: unstable
snow: 0
capacity: 2
broken: true
building: plains1
pickable: null
standers: null
pullers: null
masters: null
neighbours: null
```

```
id: unstable
snow: 0
capacity: 2
broken: true
building: plains1
pickable: null
standers: e1, e2, e3, e4
pullers: e1, e2, e3, e4
masters: e1, e2, e3, e4
neighbours: null
```

### **8.2.8 Védelemmel rendelkező karakter esésének vizsgálata**

- **Leírás**

Elhelyezünk egy karaktert, a tesztnben konkrétan egy eszkimót egy stabil jégtáblára. Ennek a stabil jégtáblának lesz egy instabil, egy-kapacitású jégtábla szomszédja, amire átléptetjük az eszkimót. Erre a stabil, szomszédos jégtáblára elhelyezünk egy húzára képes objektumot, ami most szintén egy eszkimó lesz. Mielőtt átléptetnénk az eszkimót, előtte felruházzuk vízi védelemmel, tehát adunk neki egy DivingSuit osztályú objektumot. Az eszkimó az instabil jégtáblára lépve, azt be fogja szakítani, ugyanakkor a karakterünknek túl kell élnie az esést; nem történhet játék vége. Meg kell vizsgálni azt is, hogy a stabil jégtáblán maradt eszkimó nem húzta-e vissza az eszkimót.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt célja annak vizsgálata, hogy ha egy karakter vízi védelemmel lép egy jégtáblára, és az beszakadva esésre készti a karaktert, akkor az a követelményeknek megfelelően fog-e reagálni. Tehát a karakter nem halhat meg, illetve más nem húzhatja el őt a helyéről; ezek hibának minősülnek. Ha a karakter meghal, azt onnan látnánk, hogy a „Game Over” kiíródna a kimenetre. Hiba továbbá, ha a jégtáblák nem érzékelik jól, az őket elhagyó, és a rájuk rálépő játékosokat.

- **Bemenet**

```
field stable 0
```

```

field unstable 1 false 0
neighbour stable unstable
neighbour unstable stable
entity mover Eskimo stable
entity puller Eskimo stable
stats stable
stats unstable
item rope1 Rope setcharacter puller
item ds1 DivingSuit setcharacter mover
stats mover
stats puller
move mover unstable
stats mover
stats unstable
stats stable

```

• Elvárt kimenet

```

id: stable
snow: 0
building: plains1
pickable: null
standers: mover, puller
pullers: mover, puller
masters: mover, puller
neighbours: unstable

```

```

id: unstable
snow: 0
capacity: 1
broken: false
building: plains2
pickable: null
standers: null
pullers: null
masters: null
neighbours: stable

```

```

id: mover
iceUnit: stable
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: ds1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

```

id: puller
iceUnit: stable

```

```
bodyHeat: 5
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: rope1
tent: null
components: null
moveBehaviour: walk2
```

```
id: mover
iceUnit: unstable
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: ds1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: amph1
```

```
id: unstable
snow: 0
capacity: 1
broken: true
building: plains2
pickable: null
standers: mover
pullers: mover
masters: mover
neighbours: stable
```

```
id: stable
snow: 0
building: plains1
pickable: null
standers: puller
pullers: puller
masters: puller
neighbours: unstable
```

### **8.2.9 Védelemmel nem rendelkező karakter esésének vizsgálata, környező kihúzók nélkül**

- **Leírás**

A tesztben egy vízi védelemmel nem rendelkező karaktert, egy eszkimót helyezünk el egy stabil jégtáblán. Ezzel a jégtáblával létrehozunk egy szomszédos, 1 teherbíró képességű, instabil jégtáblát. Az eszkimót átmozgatjuk erre a jégtáblára, ami beszakad, majd az esésben meghal a karakter.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt feladata, hogy ellenőrizze azt, amikor egy víztől védtelen karakter, környéki kihúzók nélkül vízbe kerül, halálával a játék végét okozva. Hiba, ha a játéknak nem lesz vége a karakter halála miatt, tehát ha nem jelenik meg a „Game Over” felirat a kimeneten. Hiba az is, ha nem török be az instabil jégtábla, ezzel elmarad a törés.

- **Bemenet**

```
field stable 0
field unstable 1 false 0
neighbour stable unstable
neighbour unstable stable
entity mover Eskimo stable
stats stable
stats unstable
stats mover
move mover unstable
stats mover
stats unstable
```

- **Elvárt kimenet**

```
id: stable
snow: 0
building: plains1
pickable: null
standers: mover
pullers: mover
masters: mover
neighbours: unstable
```

```
id: unstable
snow: 0
capacity: 1
broken: false
building: plains2
pickable: null
standers: null
pullers: null
masters: null
neighbours: stable
```

```
id: mover
iceUnit: stable
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

Game Over

id: mover

```

iceUnit: unstable
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

```

id: unstable
snow: 0
capacity: 1
broken: true
building: plains2
pickable: null
standers: mover
pullers: mover
masters: mover
neighbours: stable

```

### **8.2.10 Védelemmel nem rendelkező karakter esésének vizsgálata, egy szomszédos mezőn található kihúzával**

- **Leírás**

Létrehozunk egy egymással szomszédos stabil, és 1 kapacitású, instabil jégtáblát. Elhelyezünk a stabil jégtáblára két eszkimót; az egyiknek adunk kötelet, a másik pedig egyszerűen, az alapértelmezett tárgyakkal jön létre, tehát a víztől védetlen lesz. A víztől védetlen eszkimóval rálépünk az instabil jégtáblára, ami beszakad. Ez azt fogja okozni, hogy segítséget fog kérni a környező mezőn lévő kihúzóktól; tehát a stabil jégtáblán maradt eszkimó kihúzza őt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszttel azt ellenőrizzük, hogy ha pontosan egy szomszédos mezőn, pontosan egy húzó van jelen, akkor az sikeresen ki tudja húzni a karaktert, aki bajba került. Hibahely lehet, ha nem húzódik el a karaktert; tehát pozíciója nem változik és/vagy meghal, ezzel a játék végét okozva (Game Over felirat a kimeneten). Hiba az is, ha jégtáblák tulajdonságai nem változnak megfelelően; nem török be az instabil jégtábla, vagy a visszahúzott karaktert jelenlétét nem érzékeli a stabil mező.

- **Bemenet**

```

field stable 0
field unstable 1 false 0
neighbour stable unstable
neighbour unstable stable
entity mover Eskimo stable
entity puller Eskimo stable
item rope1 Rope setcharacter puller
stats stable
stats unstable
stats mover
stats puller

```

move mover unstable

stats mover

stats unstable

stats stable

- **Elvárt kimenet**

id: stable

snow: 0

building: plains1

pickable: null

standers: mover, puller

pullers: mover, puller

masters: mover, puller

neighbours: unstable

id: unstable

snow: 0

capacity: 1

broken: false

building: plains2

pickable: null

standers: null

pullers: null

masters: null

neighbours: stable

id: mover

iceUnit: stable

bodyHeat: 5

stamina: 4

digger: bareHand1

waterProtector: lung1

pullerDevice: basicPull1

tent: null

components: null

moveBehaviour: walk1

id: puller

iceUnit: stable

bodyHeat: 5

stamina: 4

digger: bareHand2

waterProtector: lung2

pullerDevice: rope1

tent: null

components: null

moveBehaviour: walk2

id: mover

iceUnit: stable

bodyHeat: 5

stamina: 3  
 digger: bareHand1  
 waterProtector: lung1  
 pullerDevice: basicPull1  
 tent: null  
 components: null  
 moveBehaviour: walk1

id: unstable  
 snow: 0  
 capacity: 1  
 broken: true  
 building: plains2  
 pickable: null  
 standers: null  
 pullers: null  
 masters: null  
 neighbours: stable

id: stable  
 snow: 0  
 building: plains1  
 pickable: null  
 standers: mover, puller  
 pullers: mover, puller  
 masters: mover, puller  
 neighbours: unstable

### **8.2.11 Védelemmel nem rendelkező karakter esésének vizsgálata, több szomszédos mezőn álló kihúzóval**

- **Leírás**

Létrehozunk egy egymással szomszédos stabil, és 1 kapacitású, instabil jégtáblát. Ezután létrehozunk ezzel az instabil jégtáblával szomszédos, stabil jégtáblát. Fontos, hogy az utóbb létrehozott stabil jégtáblát az első után állítjuk be, mint az instabil szomszédja. Elhelyezünk az első stabil jégtáblára két eszkimót; az egyiknek adunk kötelet, a másik pedig egyszerűen, az alapértelmezett tárgyakkal jön létre, tehát a víztől védetlen lesz. A másik stabil jégtáblára elhelyezünk egy kötéllel rendelkező eszkimót. A víztől védetlen eszkimóval rálépünk az instabil jégtáblára, ami beszakad. Ez azt fogja okozni, hogy segítséget fog kérni a környező mezőn lévő kihúzóktól, tehát a fenti beállításokkal az az eszkimó fogja őt elhúzni, aki azon a jégtáblán van, ahonnan indult a mozgó eszkimó.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt célja annak ellenőrzése, hogy amennyiben több elhúzó is akad segíteni egy bajba jutott karakternek, akkor a bajba jutott karakterek a megfelelő elhúzó segít. Az elhúzó mezőjének az először szomszédként hozzáadott stabil jégtáblának kell lennie. A tesztnben tehát hibahely, ha törést követően nem ide kerül a bajba jutott karakter. További hiba lehet, ha a mezők nem érzékelik helyes módon a mozgások és kihúzás után, hogy melyik karakterek tartózkodnak rajtuk.

- **Bemenet**

field stable1 0  
 field unstable 1 false 0  
 field stable2 0  
 neighbour stable1 unstable  
 neighbour unstable stable1, stable2  
 neighbour stable2 unstable  
 entity mover Eskimo stable1  
 entity puller1 Eskimo stable1  
 entity puller2 Eskimo stable2  
 item rope1 Rope setcharacter puller1  
 item rope2 Rope setcharacter puller2  
 stats stable1  
 stats unstable  
 stats stable2  
 stats mover  
 stats puller1  
 stats puller2  
 move mover unstable  
 stats mover  
 stats unstable  
 stats stable1  
 stats stable2

- **Elvárt kimenet**

id: stable1  
 snow: 0  
 building: plains1  
 pickable: null  
 standers: mover, puller1  
 pullers: mover, puller1  
 masters: mover, puller1  
 neighbours: unstable

id: unstable  
 snow: 0  
 capacity: 1  
 broken: false  
 building: plains2  
 pickable: null  
 standers: null  
 pullers: null  
 masters: null  
 neighbours: stable1, stable2

id: stable2  
 snow: 0  
 building: plains3  
 pickable: null  
 standers: puller2

pullers: puller2  
masters: puller2  
neighbours: unstable

id: mover  
iceUnit: stable1  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtecter: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: puller1  
iceUnit: stable1  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtecter: lung2  
pullerDevice: rope1  
tent: null  
components: null  
moveBehaviour: walk2

id: puller2  
iceUnit: stable2  
bodyHeat: 5  
stamina: 4  
digger: bareHand3  
waterProtecter: lung3  
pullerDevice: rope2  
tent: null  
components: null  
moveBehaviour: walk3

id: mover  
iceUnit: stable1  
bodyHeat: 5  
stamina: 3  
digger: bareHand1  
waterProtecter: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: unstable  
snow: 0

```
capacity: 1
broken: true
building: plains2
pickable: null
standers: null
pullers: null
masters: null
neighbours: stable1, stable2
```

```
id: stable1
snow: 0
building: plains1
pickable: null
standers: puller1, mover
pullers: puller1, mover
masters: puller1, mover
neighbours: unstable
```

```
id: stable2
snow: 0
building: plains3
pickable: null
standers: puller2
pullers: puller2
masters: puller2
neighbours: unstable
```

### **8.2.12 Jegesmedve vízbeesésének vizsgálata**

- **Leírás**

A tesztben létrehozunk egy stabil, és egy instabil, 1 kapacitású mezőt. A stabil jégtáblára elhelyezünk egy jegesmedvét. A jegesmedvével ezután elmozdulunk az instabil mezőre, ami ennek hatására betörök. A jegesmedvére a törésnek nincs semmilyen hatása.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt célja annak ellenőrzése, hogy abban az esetben, amikor egy jegesmedve tör be egy instabil jégtáblát, akkor az instabil jégtábla ugyanúgy betörhet, és a jegesmedvére ez nincs hatással. Tehát hiba, ha a jegesmedvére bárminemű hatása van a jégtábla betörésének, továbbá hiba az is, ha nem törik be a jégtábla. Várható hibahely az is, ha a két mező nem érzékeli jól a rajta mozgó jegesmedvét. További hibalehetőség, ha a szabályosan mozgó jegesmedve mozgása meghiúsul.

- **Bemenet**

```
field stable 0
field unstable 1 false 0
neighbour stable unstable
neighbour unstable stable
entity pb1 Bear stable
stats stable
stats unstable
stats pb1
move pb1 unstable
```

stats stable

stats unstable

stats pb1

- **Elvárt kimenet**

id: stable

snow: 0

building: plains1

pickable: null

standers: pb1

pullers: null

masters: null

neighbours: unstable

id: unstable

snow: 0

capacity: 1

broken: false

building: plains2

pickable: null

standers: null

pullers: null

masters: null

neighbours: stable

id: pb1

iceUnit: stable

id: stable

snow: 0

building: plains1

pickable: null

standers: null

pullers: null

masters: null

neighbours: unstable

id: unstable

snow: 0

capacity: 1

broken: true

building: plains2

pickable: null

standers: pb1

pullers: null

masters: null

neighbours: stable

id: pb1

iceUnit: unstable

### **8.2.13 Szárazföldi mozgással bíró karakter hóval fedte lukra lép**

- **Leírás**

Létrehozunk egy stabil jégtáblát, és egy azzal szomszédos hóval fedte lukat; a hóval fedte luk egy betörött instabil jégtábla, amin hó van. Elhelyezünk a stabil jégtáblára egy eszkimót, aki szárazföldi mozgásformával rendelkezik. Ezután eszkimóval elmegyünk a hóval fedte lukba, és beleesünk.

- **Ellenőrzött funkcionális, várható hibahelyek**

A teszt célja annak az ellenőrzése többek között, hogy a szárazföldi mozgásformával rendelkező karakterek el tudnak-e mozdulni egy hóval fedte lukra, engedi-e nekik ezt ez a mozgásforma. Ha nem engedi ezt, akkor az hibás működést jelent. További hibalehetőség, ha a mezők, akik részt vesznek a mozgásban helytelenül érzékelik a mozgókat.

- **Bemenet**

```
field stable 0
field hole 1 true 1
neighbour stable hole
neighbour hole stable
entity mover Eskimo stable
stat stable
stat hole
stat mover
move mover hole
stat stable
stat hole
stat mover
```

- **Elvárt kimenet**

```
id: stable
snow: 0
building: plains1
pickable: null
standers: mover
pullers: mover
masters: mover
neighbours: hole
```

```
id: hole
snow: 1
capacity: 1
broken: true
building: null
pickable: null
standers: null
pullers: null
masters: null
neighbours: stable
```

```
id: mover
iceUnit: stable
bodyHeat: 5
stamina: 4
digger: bareHand1
```

```
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

Game Over  
 id: stable  
 snow: 0  
 building: plains1  
 pickable: null  
 standers: null  
 pullers: null  
 masters: null  
 neighbours: hole

```
id: hole
snow: 0
capacity: 1
broken: true
building: null
pickable: null
standers: mover
pullers: mover
masters: mover
neighbours: stable
```

```
id: mover
iceUnit: hole
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

### **8.2.14 Vízben úszó elhúzó viselkedésének ellenőrzése, ha a szomszédos mezőn egy karakter vízbe esik és segítséget kér**

- **Leírás**

Létrehozunk egy stabil, és két instabil jégtáblát. A stabil jégtábla szomszédos az egyik instabil jégtáblával, és a két instabil jégtábla szomszédos egymással. A stabil jégtáblával szomszédos instabil jégtábla teherbíró képessége egy, és nem törött; míg a másik jégtábla be van törve. Elhelyezünk egy kötéllel rendelkező eszkimót a törött jégtáblán, a stabilon pedig elhelyezünk egy másik eszkimót. A stabil jégtáblán álló eszkimó rálép az instabil jégtáblára, ami betörök ennek hatására. Az vízben lévő eszkimó nem húzza el a lezuhan eszkimót, hiszen ő maga vízben van; ezért meghal az eszkimó.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt főcélja annak ellenőrzése, hogy a vízben lévő karakter ne húzza el egy bajba került társát, hiszen ez teljesen értelmetlen. Tehát az egyik hibahely, hogy az úszó eszkimó elhúzza a társát. Szintén hiba, ha nem fullad meg az eszkimó, aki alatt betörött a jégtábla. További hibalehetőségek, ha a jégtáblák nem kezelik jól az eszkimó mozgásának történő reakciókat.

- **Bemenet**

```
field spawn 0
field unstable 1 false 0
field swimpool 1 true 0
neighbour spawn unstable
neighbour unstable spawn, swimpool
neighbour swimpool unstable
entity drowner Eskimo spawn
entity swimmer Eskimo swimpool
item rope1 Rope setcharacter swimmer
stats spawn
stats unstable
stats swimpool
stats drowner
stats swimmer
move drowner unstable
stats spawn
stats unstable
stats swimpool
stats drowner
stats swimmer
```

- **Elvárt kimenet**

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: drowner
pullers: drowner
masters: drowner
neighbours: unstable
```

```
id: unstable
snow: 0
capacity: 1
broken: false
building: plains2
pickable: null
standers: null
pullers: null
masters: null
neighbours: spawn, swimpool
```

```
id: swimpool
snow: 0
capacity: 1
```

broken: true  
building: null  
pickable: null  
standers: swimmer  
pullers: swimmer  
masters: swimmer  
neighbours: unstable

id: drowner  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: swimmer  
iceUnit: hole  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: rope1  
tent: null  
components: null  
moveBehaviour: walk2

Game Over  
id: spawn  
snow: 0  
building: plains1  
pickable: null  
standers: null  
pullers: null  
masters: null  
neighbours: unstable

id: unstable  
snow: 0  
capacity: 1  
broken: true  
building: null  
pickable: null  
standers: drowner  
pullers: drowner  
masters: drowner  
neighbours: spawn, swimpool

```

id: swimpool
snow: 0
capacity: 1
broken: true
building: null
pickable: null
standers: swimmer
pullers: swimmer
masters: swimmer
neighbours: unstable

```

```

id: drowner
iceUnit: unstable
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

```

id: swimmer
iceUnit: swimpool
bodyHeat: 5
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: rope1
tent: null
components: null
moveBehaviour: walk2

```

### **8.2.15 Karakter alapértelmezett húzóeszközének vizsgálata**

- **Leírás**

Létrehozunk egy stabil; és egy vele szomszédos, 1 teherbíró képességű szomszédos instabil jégtáblát. A stabil jégtáblán elhelyezünk két eszkimót. Az egyik eszkimót elmozgatjuk az instabil jégtáblára, ami beszakad. Mivel nincs ennek a mozgó eszkimónak víz elleni védelme, se a közelben húzásra képes karakter, meghal.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak az ellenőrzése, hogy karakter alapértelmezett húzóeszköze nem tud kimenteni a vízből más karaktert. További hiba lehet, ha a mezők nem érzékelik jól a rajtuk éppen jelenlévő mozgó objektumokat; valamint, ha a mozgó objektumok sem kezelik jól, hogy melyik mezőn kell tartózkodniuk egy adott időpillanatban. További hibahely, ha a mozgó eszkimó nem hal bele az esésbe.

- **Bemenet**

```

field spawn 0
field unstable 1 false 0
neighbour spawn unstable

```

neighbour unstable spawn  
 entity mover Eskimo stable  
 entity falsepuller Eskimo stable  
 stats spawn  
 stats unstable  
 stats mover  
 stats falsepuller  
 mover mover unstable  
 stats spawn  
 stats unstable  
 stats mover  
 stats falsepuller

- **Elvárt kimenet**

id: spawn  
 snow: 0  
 building: plains1  
 pickable: null  
 standers: mover, falsepuller  
 pullers: mover, falsepuller  
 masters: mover, falsepuller  
 neighbours: unstable

id: unstable  
 snow: 0  
 capacity: 1  
 broken: false  
 building: plains2  
 pickable: null  
 standers: null  
 pullers: null  
 masters: null  
 neighbours: spawn

id: mover  
 iceUnit: spawn  
 bodyHeat: 5  
 stamina: 4  
 digger: bareHand1  
 waterProtector: lung1  
 pullerDevice: basicPull1  
 tent: null  
 components: null  
 moveBehaviour: walk1

id: falsepuller  
 iceUnit: spawn  
 bodyHeat: 5  
 stamina: 4  
 digger: bareHand2

```
waterProtector: lung2
pullerDevice: basicPull2
tent: null
components: null
moveBehaviour: walk2
```

Game Over

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: falsepuller
pullers: falsepuller
masters: falsepuller
neighbours: unstable
```

```
id: unstable
snow: 0
capacity: 1
broken: true
building: null
pickable: null
standers: mover
pullers: mover
masters: mover
neighbours: spawn
```

```
id: mover
iceUnit: unstable
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: falsepuller
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: rope1
tent: null
components: null
moveBehaviour: walk2
```

### **8.2.16 Eszkimó iglu építése épületlen mezőre**

- **Leírás**

Létrehozunk egy stabil jégtáblát, és elhelyezünk rajta egy eszkimót. Használjuk az eszkimó képességét, ennek hatására megjelenik rajta egy iglu.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja annak az ellenőrzése, hogy az eszkimó, ha használja a képességét egy épületlen mezőre, akkor azon sikeresen fel kell épülnie egy iglunak. Ha nem épül fel az iglu, az hibás működést jelent. További hiba, ha az eszkimó vagy a jégtáblája nem érzékeli jól, hogy melyik mezőn tartózkodik az eszkimó, vagy a jégtáblán az eszkimó tartózkodik.

- **Bemenet**

```
field spawn 0
entity builder Eskimo spawn
stats spawn
stats builder
ability spawn builder
stats spawn
stats builder
```

- **Elvárt kimenet**

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: builder
pullers: builder
masters: builder
neighbours: null
```

```
id: builder
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: spawn
snow: 0
building: igloo1
pickable: null
standers: builder
pullers: builder
masters: builder
neighbours: null
```

```
id: builder
iceUnit: spawn
bodyHeat: 5
```

stamina: 3  
 digger: bareHand1  
 waterProtector: lung1  
 pullerDevice: basicPull1  
 tent: null  
 components: null  
 moveBehaviour: walk1

### **8.2.17 Eszkimó iglu építése igluval már rendelkező mezőre**

- **Leírás**

Létrehozunk egy stabil jégtáblát, és elhelyezünk rajta egy eszkimót. Rakunk a jégtáblára egy iglut, ezután használjuk az eszkimó képességét, aminek a hatására nem történik semmi.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja annak az ellenőrzése, hogy az eszkimó, ha használja a képességét egy igluval már rendelkező mezőre, akkor azon nem szabad újra egy iglu megjelenjen. Ha ez mégis megtörténik, az hibás működést jelent. További hiba, ha az eszkimó vagy a jégtáblája nem érzékeli jól, hogy melyik mezőn tartózkodik az eszkimó, vagy a jégtáblán az eszkimó tartózkodik.

- **Bemenet**

field spawn 0  
 entity builder Eskimo spawn  
 building igloo1 Igloo spawn  
 stats spawn  
 stats builder  
 ability spawn builder  
 stats spawn  
 stats builder

- **Elvárt kimenet**

id: spawn  
 snow: 0  
 building: igloo1  
 pickable: null  
 standers: builder  
 pullers: builder  
 masters: builder  
 neighbours: null

id: builder  
 iceUnit: spawn  
 bodyHeat: 5  
 stamina: 4  
 digger: bareHand1  
 waterProtector: lung1  
 pullerDevice: basicPull1  
 tent: null  
 components: null  
 moveBehaviour: walk1

id: spawn

```

snow: 0
building: igloo1
pickable: null
standers: builder
pullers: builder
masters: builder
neighbours: null

id: builder
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

### **8.2.18 Eszkimó iglu építése sátras mezőre**

- **Leírás**

Létrehozunk egy stabil jégtáblát, és elhelyezünk rajta egy eszkimót. Rakunk a jégtáblára egy sárat, ezután használjuk az eszkimó képességét, aminek a hatására nem történik semmi.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja annak az ellenőrzése, hogy az eszkimó, ha használja a képességét egy sátorral már rendelkező mezőre, akkor azon nem szabad az iglu a sárat felülírja. Ha ez mégis megtörténik, az hibás működést jelent. További hiba, ha az eszkimó vagy a jégtáblája nem érzékeli jól, hogy melyik mezőn tartózkodik az eszkimó, vagy a jégtáblán az eszkimó tartózkodik.

- **Bemenet**

```

field spawn 0
entity builder Eskimo spawn
building tent1 Tent spawn
stats spawn
stats builder
ability spawn builder
stats spawn
stats builder

```

- **Elvárt kimenet**

```

id: spawn
snow: 0
building: tent1
pickable: null
standers: builder
pullers: builder
masters: builder
neighbours: null

```

```

id: builder
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

```

id: spawn
snow: 0
building: tent1
pickable: null
standers: builder
pullers: builder
masters: builder
neighbours: null

```

```

id: builder
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

### **8.2.19      Uszó Eszkimó iglut próbál építeni**

- **Leírás**

Létrehozunk egy már betört hó nélküli jégtáblát, és elhelyezünk rajta egy eszkimót. Ezután használjuk az eszkimó képességét, aminek a hatására nem történik semmi.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja annak az ellenőrzése, hogy az eszkimó, ha egy betört jégtáblán használja a képességét, akkor azon nem szabad iglu megjelenjen. Ha ez mégis megtörténik, az hibás működést jelent. További hiba, ha az eszkimó vagy a jégtáblája nem érzékeli jól, hogy melyik mezőn tartózkodik az eszkimó, vagy a jégtáblán az eszkimó tartózkodik.

- **Bemenet**

```

field spawn 1 true 0
entity builder Eskimo spawn
stats spawn
stats builder
ability spawn builder
stats spawn
stats builder

```

- **Elvárt kimenet**

```
id: spawn
snow: 0
capacity: 1
broken: true
building: plains1
pickable: null
standers: builder
pullers: builder
masters: builder
neighbours: null
```

```
id: builder
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: spawn
snow: 0
capacity: 1
broken: true
building: plains1
pickable: null
standers: builder
pullers: builder
masters: builder
neighbours: null
```

```
id: builder
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

### **8.2.20      Explorer felfedi egy stabil mező teherbíró képességét**

- **Leírás**

Létrehozunk egy stabil jégtáblát, és rajta egy felfedezőt. Használjuk a felfedező képességét a jégtáblán.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja annak ellenőrzése, hogy amennyiben stabil jégtáblára használjuk a felfedező képességét, az megfelelően működik; tehát a parancs hatására a kimenetre a megfelelő karakterlánc kerül. További hibahely, ha a jégtábla nem kezeli jól a rajta tartózkodó felfedezőt, vagy ha felfedező rosszul menti el, hogy melyik jégtáblán tartózkodik.

- **Bemenet**

```
field spawn 0
entity exp Explorer spawn
ability spawn exp
stats spawn
stats exp
```

- **Elvárt kimenet**

The iceunit is stable!

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: exp
pullers: exp
masters: exp
neighbours: null
```

```
id: exp
iceUnit: spawn
bodyHeat: 4
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

### **8.2.21      Explorer felfedi egy instabil mező teherbíró képességét**

- **Leírás**

Létrehozunk egy instabil jégtáblát, és rajta egy felfedezőt. Használjuk a felfedező képességét a jégtáblán.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja annak ellenőrzése, hogy amennyiben instabil jégtáblára használjuk a felfedező képességét, az megfelelően működik; tehát a parancs hatására a kimenetre a megfelelő karakterláncnak megfelelő struktúrájúnak kell lennie, illetve a megfelelő teherbíró képességet kell jeleznie a felhasználónak. További hibahely, ha a jégtábla nem kezeli jól a rajta tartózkodó felfedezőt, vagy ha felfedező rosszul menti el, hogy melyik jégtáblán tartózkodik.

- **Bemenet**

```
field spawn 3 false 0
entity exp Explorer spawn
ability spawn exp
stats spawn
stats exp
```

- **Elvárt kimenet**

The iceunit can hold 3 objects!

```
id: spawn
snow: 0
capacity: 3
broken: false
building: plains1
pickable: null
standers: exp
pullers: exp
masters: exp
neighbours: null
```

```
id: exp
iceUnit: spawn
bodyHeat: 4
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

### **8.2.22 Egy karakter összerakja a saját komponenseiből (pont elég)**

- **Leírás**

Létrehozunk egy jégtáblát, és rajta egy eszkimót. Az eszkimót felruházzuk a szükséges és elégséges számú és típusú jelzőpisztoly komponenssel, majd utasítjuk, hogy rakja össze azt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja annak az ellenőrzése, hogy a követelmények szerinti minimális számú komponensből összerakható a jelzőpisztoly. A jelzőpisztoly hatása, hogy a játékot megnyerjük, ennek meg kell jelennie a kimeneten (Game Won! felirat). Ha ez elmarad, az hibás jelentést jelent. Ez történhet amiatt is, mert nem jön létre a pisztoly, illetve amiatt is, hogy a pisztoly összerakása nem váltja ki a játék megnyerését. További hiba, ha az eszkimó nem érzékeli jól, hogy melyik jégtáblán van; illetve, ha a jégtábla nem érzékeli a rajta álló eszkimót megfelelő módon.

- **Bemenet**

```
field spawn 0
entity master Eskimo spawn
stats spawn
stats master
item pis1 Pistol setcharacter master
stats master
item cart1 Cartridge setcharacter master
stats master
item ligh1 Light setcharacter master
stats master
assamble master
stats spawn
```

stats master

- **Elvárt kimenet**

id: spawn

snow: 0

building: plains1

pickable: null

standers: exp

pullers: exp

masters: exp

neighbours: null

id: master

iceUnit: spawn

bodyHeat: 5

stamina: 4

digger: bareHand1

waterProtector: lung1

pullerDevice: basicPull1

tent: null

components: null

moveBehaviour: walk1

id: master

iceUnit: spawn

bodyHeat: 5

stamina: 4

digger: bareHand1

waterProtector: lung1

pullerDevice: basicPull1

tent: null

components: pis1

moveBehaviour: walk1

id: master

iceUnit: spawn

bodyHeat: 5

stamina: 4

digger: bareHand1

waterProtector: lung1

pullerDevice: basicPull1

tent: null

components: pis1, cart1

moveBehaviour: walk1

id: master

iceUnit: spawn

bodyHeat: 5

stamina: 4

digger: bareHand1

waterProtector: lung1

```

pullerDevice: basicPull1
tent: null
components: pis1, cart1, light1
moveBehaviour: walk1

```

Game Won!

```

id: spawn
snow: 0
building: plains1
pickable: null
standers: exp
pullers: exp
masters: exp
neighbours: null

```

```

id: master
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: pis1, cart1, light1
moveBehaviour: walk1

```

### **8.2.23 Egy karakter nem rakja össze, mert tetszőlegesen hiányzik valami**

- **Leírás**

Létrehozunk három jégtáblát, és mindegyikre elhelyezünk egy eszkimót, tehát összesen hármat. Először az egyik, komponenssel nem rendelkező eszkimóval megpróbáljuk összeszerelni a pisztolyt. Ezután felruházzuk mindenből eszkimót, egy darab, egymástól különböző jelzőpisztoly komponenssel; majd mindenből eszkimóval megpróbáljuk összeszerelni a jelzőpisztolyt. Ezután további komponenseket osztunk ki nekik, pontosan úgy; hogy mindenki pontosan egy típusú komponens hiányozzon a pisztoly összerakásához, és egyenként ez a hiányzó komponens legyen különböző az eszkimók között. Ezután megpróbáljuk ismét mindenből eszkimóval összeszerelni a pisztolyt. Ezután adunk mindenből eszkimónak további komponenseket, azonban úgy, hogy a hiányzó típusából ne adjunk nekik. Ezután megint megpróbáljuk összeszerelni a pisztolyt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt célja annak az ellenőrzése, hogy nem elegendő komponensekből nem tudjuk összeszerelni a jelzőpisztolyt. A tesztnél egyaránt ellenőrizzük, hogy a megfelelő számú, illetve különböző típusú komponensek hiányában mind sikertelen-e az összeszerelés. Ha sikerül az összeszerelés, abban az esetben a teszt sikertelen, a működés bizonyítottan hibás. További hiba, ha az eszkimó nem érzékeli jól, hogy melyik jégtáblán van; illetve ha a jégtábla nem érzékeli a rajta álló eszkimót megfelelő módon.

- **Bemenet**

```

field spawn0 0
field spawn1 0

```

```

field spawn2 0
entity master0 Eskimo spawn0
entity master1 Eskimo spawn1
entity master2 Eskimo spawn2
stats spawn0
stats spawn1
stats spawn2
stats master0
stats master1
stats master2
assemble master0
item pis0 Pistol setcharacter master0
item cart1 Cartridge setcharacter master1
item light2 Light setcharacter master2
stats master0
stats master1
stats master2
assemble master0
assemble master1
assemble master2
item cart0 Cartridge setcharacter master0
item light1 Light setcharacter master1
item pis2 Pistol setcharacter master2
stats master0
stats master1
stats master2
assemble master0
assemble master1
assemble master2
item cart00 Cartridge setcharacter master0
item light11 Light setcharacter master1
item pis22 Pistol setcharacter master2
stats master0
stats master1
stats master2
assemble master0
assemble master1
assemble master2
stats spawn0
stats spawn1
stats spawn2
stats master0
stats master1
stats master2

```

• **Elvárt kimenet**

```

id: spawn0
snow: 0
building: plains1
pickable: null
standers: master0

```

pullers: master0  
masters: master0  
neighbours: null

id: spawn1  
snow: 0  
building: plains2  
pickable: null  
standers: master1  
pullers: master1  
masters: master1  
neighbours: null

id: spawn2  
snow: 0  
building: plains3  
pickable: null  
standers: master2  
pullers: master2  
masters: master2  
neighbours: null

id: master0  
iceUnit: spawn0  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1  
id: master1  
iceUnit: spawn1  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: null  
moveBehaviour: walk2

id: master2  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand3  
waterProtector: lung3  
pullerDevice: basicPull3

tent: null  
components: null  
moveBehaviour: walk3

id: master0  
iceUnit: spawn0  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtecter: lung1  
pullerDevice: basicPull1  
tent: null  
components: pis0  
moveBehaviour: walk1

id: master1  
iceUnit: spawn1  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtecter: lung2  
pullerDevice: basicPull2  
tent: null  
components: cart1  
moveBehaviour: walk2

id: master2  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand3  
waterProtecter: lung3  
pullerDevice: basicPull3  
tent: null  
components: light2  
moveBehaviour: walk3

id: master0  
iceUnit: spawn0  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtecter: lung1  
pullerDevice: basicPull1  
tent: null  
components: pis0, cart0  
moveBehaviour: walk1

id: master1  
iceUnit: spawn1

```
bodyHeat: 5
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: basicPull2
tent: null
components: cart1, light1
moveBehaviour: walk2
```

```
id: master2
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand3
waterProtector: lung3
pullerDevice: basicPull3
tent: null
components: light2, pis2
moveBehaviour: walk3
```

```
id: master0
iceUnit: spawn0
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: pis0, cart0, cart00
moveBehaviour: walk1
```

```
id: master1
iceUnit: spawn1
bodyHeat: 5
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: basicPull2
tent: null
components: cart1, light1, light11
moveBehaviour: walk2
```

```
id: master2
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand3
waterProtector: lung3
pullerDevice: basicPull3
tent: null
```

components: light2, pis2, pis22  
moveBehaviour: walk3

id: spawn0  
snow: 0  
building: plains1  
pickable: null  
standers: master0  
pullers: master0  
masters: master0  
neighbours: null

id: spawn1  
snow: 0  
building: plains2  
pickable: null  
standers: master1  
pullers: master1  
masters: master1  
neighbours: null

id: spawn2  
snow: 0  
building: plains3  
pickable: null  
standers: master2  
pullers: master2  
masters: master2  
neighbours: null

id: master0  
iceUnit: spawn0  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: pis0, cart0, cart00  
moveBehaviour: walk1

id: master1  
iceUnit: spawn1  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: cart1, light1, light11

```
moveBehaviour: walk2
```

```
id: master2
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand3
waterProtector: lung3
pullerDevice: basicPull3
tent: null
components: light2, pis2, pis22
moveBehaviour: walk3
```

### **8.2.24 Egy karakter összerakja bőven több komponensből**

- **Leírás**

Létrehozunk egy jégtáblát, rajta egy eszkimót. Az eszkimót felruházzuk a pisztoly összeszereléséhez szükséges komponensek duplájával. Megpróbáljuk összeszerelni a pisztolyt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak vizsgálata, hogy ha a karakterek a kelleténél több komponens is rendelkezésére áll, az nem zavarja meg az összerakásban. Tehát hiba, ha meghiúsul az összerakás és hiba az is, ha nem egy pisztolyt rakunk össze. Az is hiba lehet, ha a pisztoly összeállítása nem implikálja a győzelmet. További hiba, ha az eszkimó nem érzékeli jól, hogy melyik jégtáblán van; illetve, ha a jégtábla nem érzékeli a rajta álló eszkimót megfelelő módon.

- **Bemenet**

```
field spawn 0
entity master Eskimo spawn
item pist1 Pistol setcharacter master
item cart1 Cartridge setcharacter master
item light1 Light setcharacter master
item pist2 Pistol setcharacter master
item cart2 Cartridge setcharacter master
item light2 Light setcharacter master
stats spawn
stats master
assemble master
stats spawn
stats master
```

- **Elvárt kimenet**

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: master
pullers: master
masters: master
neighbours: null
```

```
id: master
```

```

iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: pist1, cart1, ligh1, pist2, cart2, ligh2
moveBehaviour: walk1

```

Game Won!

```

id: spawn
snow: 0
building: plains1
pickable: null
standers: master
pullers: master
masters: master
neighbours: null

```

```

id: master
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: pist1, cart1, ligh1, pist2, cart2, ligh2
moveBehaviour: walk1

```

### **8.2.25 Több karakter összerakja együtt a pont megfelelő mennyiséggű és típusú komponensekből a jelzőpisztolyt**

- **Leírás**

Létrehozunk egy jégtáblát, rajta három eszkimóval. Mindegyik eszkimó megkapja az egyik jelzőpisztoly komponenst. Utasítjuk az egyik eszkimót az összeszerelésre.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja az együttes összerakás vizsgálata. Ebben a tesztesetben összesen pont a megfelelő komponens mennyiséggel és típussal rendelkeznek a karakterek ugyanazon mezőn, tehát sikeresnek kell lennie az összerakásnak, a művelet eredménye győzelem kell, hogy legyen. További hiba, ha az eszkimó nem érzékeli jól, hogy melyik jégtáblán van; illetve ha a jégtábla nem érzékeli a rajta álló eszkimót megfelelő módon.

- **Bemenet**

```

field spawn 0
entity master1 Eskimo spawn
entity master2 Eskimo spawn
entity master3 Eskimo spawn
item pist1 Pistol setcharacter master1
item cart1 Cartridge setcharacter master2

```

```
item light1 Light setcharacter master3  
stats spawn  
stats master1  
stats master2  
stats master3  
assemble master1  
stats spawn  
stats master1  
stats master2  
stats master3
```

- **Elvárt kimenet**

```
id: spawn  
snow: 0  
building: plains1  
pickable: null  
standers: master1, master2, master3  
pullers: master1, master2, master3  
masters: master1, master2, master3  
neighbours: null
```

```
id: master1  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: pist1  
moveBehaviour: walk1
```

```
id: master2  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: cart1  
moveBehaviour: walk2
```

```
id: master3  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand3  
waterProtector: lung3  
pullerDevice: basicPull3  
tent: null
```

components: light1

moveBehaviour: walk3

Game Won!

id: spawn

snow: 0

building: plains1

pickable: null

standers: master1, master2, master3

pullers: master1, master2, master3

masters: master1, master2, master3

neighbours: null

id: master1

iceUnit: spawn

bodyHeat: 5

stamina: 3

digger: bareHand1

waterProtector: lung1

pullerDevice: basicPull1

tent: null

components: pist1

moveBehaviour: walk1

id: master2

iceUnit: spawn

bodyHeat: 5

stamina: 4

digger: bareHand2

waterProtector: lung2

pullerDevice: basicPull2

tent: null

components: cart1

moveBehaviour: walk2

id: master3

iceUnit: spawn

bodyHeat: 5

stamina: 4

digger: bareHand3

waterProtector: lung3

pullerDevice: basicPull3

tent: null

components: light1

moveBehaviour: walk3

### **8.2.26 Több karakter nem rakja össze a jelzőpisztolyt, mert hiányzik egy komponens típus**

- Leírás

Létrehozunk egy mezőt, elhelyezünk rajta két eszkimót. Utasítjuk az egyik eszkimót az összeszerelésre. Ezután minden két eszkimót felruházzuk egy-egy különböző típusú komponenssel. Utasítjuk az egyik eszkimót ismét az összeszerelésre. Ezután minden két eszkimónak adunk egy-egy az komponenst, pontosan olyan típusút, amit korábban a másik kapott. Ezután megint megpróbálkozunk az összeszereléssel.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy néhány olyan esetben, amikor több karakter tartózkodik egy jégtáblán, és nincsenek meg a szükséges tulajdonságaik az összeszereléshez; az összeszerelés valóban sikertelen-e. Tehát ha a tesztben sikerül az összeszerelés, és ezzel azt okozzuk, hogy megnyerjük a játékot, az hibának számít. További hiba, ha az eszkimó nem érzékeli jól, hogy melyik jégtáblán van; illetve ha a jégtábla nem érzékeli a rajta álló eszkimót megfelelő módon, vagy valamilyen tulajdonságuk változik, amire nem számítunk.

- **Bemenet**

```
field spawn 0
entity master1 Eskimo spawn
entity master2 Eskimo spawn
item pist1 Pistol setcharacter master1
item cart1 Cartridge setcharacter master2
stats spawn
stats master1
stats master2
assemble master1
item cart2 Cartridge setcharacter master1
item pist2 Pistol setcharacter master2
stats master1
stats master2
assemble master2
stats spawn
stats master1
stats master2
```

- **Elvárt kimenet**

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: master1, master2
pullers: master1, master2
masters: master1, master2
neighbours: null
```

```
id: master1
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
```

components: pist1

moveBehaviour: walk1

id: master2

iceUnit: spawn

bodyHeat: 5

stamina: 4

digger: bareHand2

waterProtector: lung2

pullerDevice: basicPull2

tent: null

components: cart1

moveBehaviour: walk2

id: master1

iceUnit: spawn

bodyHeat: 5

stamina: 4

digger: bareHand1

waterProtector: lung1

pullerDevice: basicPull1

tent: null

components: pist1, cart2

moveBehaviour: walk1

id: master2

iceUnit: spawn

bodyHeat: 5

stamina: 4

digger: bareHand2

waterProtector: lung2

pullerDevice: basicPull2

tent: null

components: cart1, pist2

moveBehaviour: walk2

id: spawn

snow: 0

building: plains1

pickable: null

standers: master1, master2

pullers: master1, master2

masters: master1, master2

neighbours: null

id: master1

iceUnit: spawn

bodyHeat: 5

stamina: 4

digger: bareHand1

```
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: pist1, cart2
moveBehaviour: walk1
```

```
id: master2
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: basicPull2
tent: null
components: cart1, pist2
moveBehaviour: walk2
```

### **8.2.27 Több karakter összeszereli a jelző pisztolyt a szükségesnél sokkal több komponensből**

- **Leírás**

Létrehozunk egy jégtáblát, rajta négy eszkimót. Mindegyik eszkimót felruházzuk a szükséges mennyiségű és minőségű komponensekkel, majd megpróbáljuk összeszerelni a jelzőpisztolyt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja annak ellenőrzése, hogy amennyiben, egy olyan szituáció áll fent, hogy több karakter tartózkodik egy mezőn, és a náluk található komponensek összessége bőven meghaladja a szükséges komponens mennyiségi elvárásait a jelzőpisztoly összeszereléséhez, akkor is sikeresen rakjuk össze a pisztolyt. Ha tehát meghiúsul a pisztoly összerakása az hibának számít. Az is hibának számít, ha több, mint egy pisztoly jön létre. Hiba továbbá az is, ha pisztoly összeszerelése nem okoz a játékban győzelmet. További hiba, ha az eszkimó nem érzékeli jól, hogy melyik jégtáblán van; illetve ha a jégtábla nem érzékeli a rajta álló eszkimót megfelelő módon, vagy valamelyen tulajdonságuk változik, amire nem számítunk.

- **Bemenet**

```
field spawn 0
entity master1 Eskimo spawn
entity master2 Eskimo spawn
entity master3 Eskimo spawn
entity master 4 Eskimo spawn
item pist1 Pistol setcharacter master1
item pist2 Pistol setcharacter master2
item pist3 Pistol setcharacter master3
item pist4 Pistol setcharacter master4
item cart1 Cartridge setcharacter master1
item cart2 Cartridge setcharacter master2
item cart3 Cartridge setcharacter master3
item cart4 Cartridge setcharacter master4
item light1 Light setcharacter master1
item light2 Light setcharacter master2
item light3 Light setcharacter master3
item light4 Light setcharacter master4
stats spawn
```

```
stats master1
stats master2
stats master3
stats master4
assemble master1
stats spawn
stats master1
stats master2
stats master3
stats master4
```

- **Elvárt kimenet**

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: master1, master2, master3, master4
pullers: master1, master2, master3, master4
masters: master1, master2, master3, master4
neighbours: null
```

```
id: master1
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: pist1, cart1, light1
moveBehaviour: walk1
```

```
id: master2
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: basicPull2
tent: null
components: pist2, cart2, light2
moveBehaviour: walk2
```

```
id: master3
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand3
waterProtector: lung3
pullerDevice: basicPull3
```

tent: null  
components: pist3, cart3, light3  
moveBehaviour: walk3

id: master4  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand4  
waterProtecter: lung4  
pullerDevice: basicPull4  
tent: null  
components: pist4, cart4, light4  
moveBehaviour: walk4

Game Won!

id: spawn  
snow: 0  
building: plains1  
pickable: null  
standers: master1, master2, master3, master4  
pullers: master1, master2, master3, master4  
masters: master1, master2, master3, master4  
neighbours: null

id: master1  
iceUnit: spawn  
bodyHeat: 5  
stamina: 3  
digger: bareHand1  
waterProtecter: lung1  
pullerDevice: basicPull1  
tent: null  
components: pist1, cart1, light1  
moveBehaviour: walk1

id: master2  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtecter: lung2  
pullerDevice: basicPull2  
tent: null  
components: pist2, cart2, light2  
moveBehaviour: walk2

id: master3  
iceUnit: spawn  
bodyHeat: 5

stamina: 4  
 digger: bareHand3  
 waterProtector: lung3  
 pullerDevice: basicPull3  
 tent: null  
 components: pist3, cart3, light3  
 moveBehaviour: walk3

id: master4  
 iceUnit: spawn  
 bodyHeat: 5  
 stamina: 4  
 digger: bareHand4  
 waterProtector: lung4  
 pullerDevice: basicPull4  
 tent: null  
 components: pist4, cart4, light4  
 moveBehaviour: walk4

### **8.2.28 A vihar lecsap egy karakterre, és nincs épület a mezőn**

- **Leírás**

Létrehozunk egy jégtáblát, és elhelyezünk rajta egy eszkimót. Ezután vihart idézünk elő, aminek hatását a karakterre irányítjuk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja annak az ellenőrzése, hogy az alapértelmezett, épület nélküli (Plains) mezőn tartózkodó karakterre, ha lecsap a vihar, akkor az a megfelelő módon reagál; tehát test hője csökken eggyel. Hiba, ha ez a csökkenés nem történik meg. További hiba, ha az eszkimó nem érzékeli jól, hogy melyik jégtáblán van; illetve ha a jégtábla nem érzékeli a rajta álló eszkimót megfelelő módon, vagy valamilyen tulajdonságuk változik, amire nem számítunk.

- **Bemenet**

field spawn 0  
 entity esk Eskimo spawn  
 stats spawn  
 stats esk  
 storm esk  
 stats spawn  
 stats esk

- **Elvárt kimenet**

id: spawn  
 snow: 0  
 building: plains1  
 pickable: null  
 standers: esk  
 pullers: esk  
 masters: esk  
 neighbours: null

id: esk  
 iceUnit: spawn  
 bodyHeat: 5

```

stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

```

id: spawn
snow: 0
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null

```

```

id: esk
iceUnit: spawn
bodyHeat: 4
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

### **8.2.29 A vihar lecsap egy karakterre, és nincs épület a mezőn**

- **Leírás**

Létrehozunk egy jégtáblát, és elhelyezünk rajta egy eszkimót. Ezután vihart idézünk elő, aminek hatását a karakterre irányítjuk. Ezt megtesszük többször, egészen pontosan ötször. Az öt vihar hatására a karakter test hőjének nullára kell csökkennie, ami a játék végét jelenti.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja annak az ellenőrzése, hogy az alapértelmezett, épület nélküli (Plains) mezőn tartózkodó karakterre, ha lecsap a vihar, akkor az a megfelelő módon reagál; továbbá, hogy a vihar, ha lecsökkenti eléggé a karakter test hőjét, az meghal, és a játék végét okozza ezzel. Hiba, ha ez a testhő csökkenés nem történik meg. Potenciális hiba, ha a karakter halála nem okozza a játék végét. További hiba, ha az eszkimó nem érzékeli jól, hogy melyik jégtáblán van; illetve ha a jégtábla nem érzékeli a rajta álló eszkimót megfelelő módon, vagy valamilyen tulajdonságuk változik, amire nem számítunk.

- **Bemenet**

```

field spawn 0
entity esk Eskimo spawn
stats spawn
stats esk
storm esk
stats esk
storm esk
stats esk

```

storm esk  
stats esk  
storm esk  
stats esk  
storm esk  
stats spawn  
stats esk

- **Elvárt kimenet**

id: spawn  
snow: 0  
building: plains1  
pickable: null  
standers: esk  
pullers: esk  
masters: esk  
neighbours: null

id: esk  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: esk  
iceUnit: spawn  
bodyHeat: 4  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: esk  
iceUnit: spawn  
bodyHeat: 3  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

```

id: esk
iceUnit: spawn
bodyHeat: 2
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

```

id: esk
iceUnit: spawn
bodyHeat: 1
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

Game Over!

```

id: spawn
snow: 0
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null

```

```

id: esk
iceUnit: spawn
bodyHeat: 0
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

### **8.2.30 A vihar lecsap egy karakterre, és sátor van a mezőn (van+nincs hely)**

- **Leírás**

Létrehozunk egy jégtáblát, és elhelyezünk rajta két eszkimót. Az egyik eszkimó eszköztárához hozzáadunk egy sárat, amit ezután elhelyeztünk vele a jégtáblán. Ezután vihart idézünk elő, és annak hatását a két eszkimóra irányítjuk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak vizsgálása, hogy amennyiben sátor van a jégmezőn, és az azon tartózkodó karaktereket eléri a vihar, azok megfelelően reagálnak-e. A tesztnél elhelyezünk egy sárat a jégtáblára; ha ennek elhelyezése sikertelen, vagy lehelyezés után nem kerül ki az eszkimó eszköztárából; az a prototípus hibás működésére utal. A vihar előidézésével az első elérő karaktert meg kell védenie a sátornak, míg a másodikat nem, hiszen kapacitása ezt nem fogja engedni, hogy megvédje őt is. Az első karakterek tehát nem változhat a test hője a vihar miatt, míg a másodiknak csökkennie kell eggel. Ha ez nem így történik, az minden hibás működésre utal. További hiba, ha az eszkimó nem érzékeli jól, hogy melyik jégtáblán van; illetve ha a jégtábla nem érzékeli a rajta álló eszkimót megfelelő módon, vagy valamilyen tulajdonságuk változik, amire nem számítunk.

- **Bemenet**

```
field spawn 0
entity esk1 Eskimo spawn
entity esk2 Eskimo spawn
item tent1 Tent setcharacter esk1
stats spawn
stats esk1
stats esk2
tent esk1
stats spawn
stats esk1
stats esk2
storm esk1
stats esk1
stats esk2
storm esk2
stats spawn
stats esk1
stats esk2
```

- **Elvárt kimenet**

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: esk1, esk2
pullers: esk1, esk2
masters: esk1, esk2
neighbours: null
```

```
id: esk1
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: tent1
components: null
```

moveBehaviour: walk1

id: esk2  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: null  
moveBehaviour: walk2

id: spawn  
snow: 0  
building: tent1  
pickable: null  
standers: esk1, esk2  
pullers: esk1, esk2  
masters: esk1, esk2  
neighbours: null

id: esk1  
iceUnit: spawn  
bodyHeat: 5  
stamina: 3  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: esk2  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: null  
moveBehaviour: walk2

id: esk1  
iceUnit: spawn  
bodyHeat: 5  
stamina: 3  
digger: bareHand1  
waterProtector: lung1

pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: esk2  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: null  
moveBehaviour: walk2

id: esk1  
iceUnit: spawn  
bodyHeat: 5  
stamina: 3  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: esk2  
iceUnit: spawn  
bodyHeat: 4  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: null  
moveBehaviour: walk2

id: spawn  
snow: 0  
building: tent1  
pickable: null  
standers: esk1, esk2  
pullers: esk1, esk2  
masters: esk1, esk2  
neighbours: null

id: esk1  
iceUnit: spawn  
bodyHeat: 5

```

stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

```

id: esk2
iceUnit: spawn
bodyHeat: 4
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: basicPull2
tent: null
components: null
moveBehaviour: walk2

```

### **8.2.31 A vihar lecsap egy karakterre, és iglu van a mezőn (van+nincs hely)**

- **Leírás**

Létrehozunk egy jégtáblát, és elhelyezünk rajta két eszkimót. Az egyik eszkimónak használjuk a képességét, ezzel elhelyezve egy iglut a jégtáblán. Ezután vihart idézünk elő, és annak hatását a két eszkimóra irányítjuk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak vizsgálása, hogy amennyiben iglu van a jégmezőn, és az azon tartózkodó karaktereket eléri a vihar, azok megfelelően reagálnak-e. A tesztbén elhelyezünk egy iglut a jégtáblára; ha ennek elhelyezése sikertelen, az a prototípus hibás működésére utal. A vihar előidézésével az első elért karaktert meg kell védenie az iglunak, míg a másodikat nem, hiszen kapacitása ezt nem fogja engedni, hogy megvédje őt is. Az első karakterek tehát nem változhat a test hője a vihar miatt, míg a másodiknak csökkennie kell eggel. Ha ez nem így történik, az minden hibás működésre utal. További hiba, ha az eszkimó nem érzékeli jól, hogy melyik jégtáblán van; illetve ha a jégtábla nem érzékeli a rajta álló eszkimót megfelelő módon, vagy valamilyen tulajdonságuk változik, amire nem számítunk.

- **Bemenet**

```

field spawn 0
entity esk1 Eskimo spawn
entity esk2 Eskimo spawn
stats spawn
stats esk1
stats esk2
ability spawn esk1
stats spawn
stats esk1
stats esk2
storm esk1
stats esk1
stats esk2
storm esk2

```

stats spawn  
 stats esk1  
 stats esk2

- **Elvárt kimenet**

id: spawn  
 snow: 0  
 building: plains1  
 pickable: null  
 standers: esk1, esk2  
 pullers: esk1, esk2  
 masters: esk1, esk2  
 neighbours: null

id: esk1  
 iceUnit: spawn  
 bodyHeat: 5  
 stamina: 4  
 digger: bareHand1  
 waterProtector: lung1  
 pullerDevice: basicPull1  
 tent: null  
 components: null  
 moveBehaviour: walk1

id: esk2  
 iceUnit: spawn  
 bodyHeat: 5  
 stamina: 4  
 digger: bareHand2  
 waterProtector: lung2  
 pullerDevice: basicPull2  
 tent: null  
 components: null  
 moveBehaviour: walk2

id: spawn  
 snow: 0  
 building: igloo1  
 pickable: null  
 standers: esk1, esk2  
 pullers: esk1, esk2  
 masters: esk1, esk2  
 neighbours: null

id: esk1  
 iceUnit: spawn  
 bodyHeat: 5  
 stamina: 3  
 digger: bareHand1

```
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: esk2
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: basicPull2
tent: null
components: null
moveBehaviour: walk2
```

```
id: esk1
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: esk2
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: basicPull2
tent: null
components: null
moveBehaviour: walk2
```

```
id: esk1
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```

id: esk2
iceUnit: spawn
bodyHeat: 4
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: basicPull2
tent: null
components: null
moveBehaviour: walk2

```

```

id: spawn
snow: 0
building: igloo1
pickable: null
standers: esk1, esk2
pullers: esk1, esk2
masters: esk1, esk2
neighbours: null

```

```

id: esk1
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

```

id: esk2
iceUnit: spawn
bodyHeat: 4
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: basicPull2
tent: null
components: null
moveBehaviour: walk2

```

### **8.2.32 Jegesmedve karaktert támad épület nélküli mezőn**

- **Leírás**

Létrehozunk két-két, egymással szomszédos stabil jégtáblát. A párokra különböző jégtábláira elhelyezünk egy-egy jegesmedvét és egy-egy eszkimót. Az egyik szomszédos jégtáblán a medvével átlépünk a karakter jégtáblájára; a másikon pedig pont fordítva, az eszkimóval átlépünk a jegesmedve jégtáblájára.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja annak ellenőrzése, hogy a karakterek megfelelően reagálnak-e a jegesmedvével való találkozásra az alapértelmezett, épülettel nem rendelkező jégtáblán, történjen ez úgy, hogy a jegesmedve lép az ő mezejükre; vagy úgy, hogy ők lépnek egy jegesmedve mezőjére. Mindkét eset halálos a karakter számára, ezért hiba, ha bármelyik esetben a karakter nem hal meg, és ezzel nem okozza a játék végét. További hiba, ha az eszkimó, vagy jegesmedve nem érzékeli jól, hogy melyik jégtáblán van; illetve ha a jégtábla nem érzékeli a rajta álló eszkimót/jegesmedvét megfelelő módon. Az is hiba, ha egy objektum valamelyen tulajdonsága változik, amire nem számítunk.

- **Bemenet**

```
field start1 0
field end1 0
neighbour start1 end1
neighbour end1 start1
field start2 0
field end2 0
neighbour start2 end2
neighbour end2 start2
entity esk1 Eskimo start1
entity pb1 Bear end1
entity esk2 Eskimo end2
entity pb2 Bear start2
stats start1
stats end1
stats start2
stats end2
stats esk1
stats pb1
stats esk2
stats pb2
move esk1 end1
stats esk1
stats pb1
move pb2 end2
stats esk2
stats pb2
stats start1
stats end1
stats start2
stats end2
stats esk1
stats pb1
stats esk2
stats pb2
```

- **Elvárt kimenet**

```
id: start1
snow: 0
building: plains1
pickable: null
standers: esk1
pullers: esk1
```

masters: esk1  
neighbours: end1

id: end1  
snow: 0  
building: plains2  
pickable: null  
standers: pb1  
pullers: null  
masters: null  
neighbours: start1

id: start2  
snow: 0  
building: plains3  
pickable: null  
standers: pb2  
pullers: null  
masters: null  
neighbours: end2

id: end2  
snow: 0  
building: plains4  
pickable: null  
standers: esk2  
pullers: esk2  
masters: esk2  
neighbours: start2

id: esk1  
iceUnit: start1  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: pb1  
iceUnit: end1

id: esk2  
iceUnit: end2  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2

pullerDevice: basicPull2  
tent: null  
components: null  
moveBehaviour: walk2

id: pb2  
iceUnit: start2

Game Over!  
id: esk1  
iceUnit: end1  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: pb1  
iceUnit: end1

Game Over!  
id: esk2  
iceUnit: end2  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: null  
moveBehaviour: walk2

id: pb2  
iceUnit: end2

id: start1  
snow: 0  
building: plains1  
pickable: null  
standers: null  
pullers: null  
masters: null  
neighbours: end1

id: end1  
snow: 0  
building: plains2

pickable: null  
standers: pb1, esk1  
pullers: esk1  
masters: esk1  
neighbours: start1

id: start2  
snow: 0  
building: plains3  
pickable: null  
standers: null  
pullers: null  
masters: null  
neighbours: end2

id: end2  
snow: 0  
building: plains4  
pickable: null  
standers: esk2, pb2  
pullers: esk2  
masters: esk2  
neighbours: start2

id: esk1  
iceUnit: end1  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: pb1  
iceUnit: end1

id: esk2  
iceUnit: end2  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: null  
moveBehaviour: walk2

id: pb2

iceUnit: end2

### **8.2.33 Jegesmedve karaktert támad sátras mezőn**

- **Leírás**

Létrehozunk két, egymással szomszédos stabil jégtáblát. Az egyiken elhelyezünk egy eszkimót, a másikon pedig egy jegesmedvét. Az eszkimónak adunk egy sárat, amit elhelyezünk vele a mezőn. Ezután a jegesmedvét átmozgatjuk az eszkimó jégtáblájára.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja, hogy azt ellenőrizzük, hogy amennyiben egy jegesmedve és egy karakter találkozik egy sátorral rendelkező mezőn, akkor a jegesmedve sikeresen megöli-e a karaktert, ezzel a játék végét okozva. Tehát hiba, ha nem lesz a játéknak vége a találkozástól. További hiba lehet, ha a sátor elhelyezése sikertelen. További hiba, ha az eszkimó, vagy jegesmedve nem érzékelni jól, hogy melyik jégtáblán van; illetve ha a jégtábla nem érzékeli a rajta álló eszkimót/jegesmedvét megfelelő módon. Az is hiba, ha egy objektum valamelyen tulajdonsága változik, amire nem számítunk.

- **Bemenet**

```
field start 0
field end 0
neighbour start end
neighbour end start
entity pb Bear start
entity esk Eskimo end
item tent1 Tent setcharacter esk
tent esk
stats start
stats end
stats esk
stats pb
move pb end
stats start
stats end
stats esk
stats pb
```

- **Elvárt kimenet**

```
id: start
snow: 0
building: plains1
pickable: null
standers: pb
pullers: null
masters: null
neighbours: end
```

```
id: end
snow: 0
building: tent1
pickable: null
standers: esk
pullers: esk
masters: esk
```

neighbours: start

id: esk  
iceUnit: end  
bodyHeat: 5  
stamina: 3  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: pb  
iceUnit: start

Game Over!

id: start  
snow: 0  
building: plains1  
pickable: null  
standers: null  
pullers: null  
masters: null  
neighbours: end

id: end  
snow: 0  
building: tent1  
pickable: null  
standers: esk, pb  
pullers: esk  
masters: esk  
neighbours: start

id: esk  
iceUnit: end  
bodyHeat: 5  
stamina: 3  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: pb  
iceUnit: end

### **8.2.34 Jegesmedve karaktert támad iglus mezőn**

- **Leírás**

Létrehozunk két, egymással szomszédos stabil jégtáblát. Az egyiken elhelyezünk két eszkimót, a másikon pedig egy jegesmedvét. Az eszkimónak használjuk a képességét, amivel egy iglut rakunk az ő jégtáblájára. Ezután a jegesmedvét átmozgatjuk az eszkimó jégtáblájára.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja, hogy azt ellenőrizzük, hogy amennyiben egy jegesmedve és egy karakter találkozik egy igluval rendelkező mezőn, akkor a jegesmedve sikeresen megöli-e a karaktert, aki nem fér be az igluba, ezzel a játék végét okozva. Tehát hiba, ha nem lesz a játéknak vége a találkozástól. Az is hiba, ha minden karakter meghal, ezzel két „Game Over” kiírást okozva. További hiba lehet, ha az iglu elhelyezése sikertelen. További hiba, ha az eszkimó, vagy jegesmedve nem érzékeli jól, hogy melyik jégtáblán van; illetve ha a jégtábla nem érzékeli a rajta álló eszkimót/jegesmedvét megfelelő módon. Az is hiba, ha egy objektum valamilyen tulajdonsága változik, amire nem számítunk.

- **Bemenet**

```
field start 0
field end 0
neighbour start end
neighbour end start
entity pb Bear start
entity esk1 Eskimo end
entity esk2 Eskimo end
ability esk1 end
stats start
stats end
stats esk1
stats esk2
stats pb
move pb end
stats start
stats end
stats esk1
stats esk2
stats pb
```

- **Elvárt kimenet**

```
id: start
snow: 0
building: plains1
pickable: null
standers: pb
pullers: null
masters: null
neighbours: end
```

```
id: end
snow: 0
building: igloo1
pickable: null
standers: esk1, esk2
```

pullers: esk1, esk2

masters: esk1, esk2

neighbours: start

id: esk1

iceUnit: end

bodyHeat: 5

stamina: 3

digger: bareHand1

waterProtector: lung1

pullerDevice: basicPull1

tent: null

components: null

moveBehaviour: walk1

id: esk2

iceUnit: end

bodyHeat: 5

stamina: 4

digger: bareHand2

waterProtector: lung2

pullerDevice: basicPull2

tent: null

components: null

moveBehaviour: walk2

id: pb1

iceUnit: start

Game Over!

id: start

snow: 0

building: plains1

pickable: null

standers: null

pullers: null

masters: null

neighbours: end

id: end

snow: 0

building: igloo1

pickable: null

standers: esk1, esk2, pb

pullers: esk1, esk2

masters: esk1, esk2

neighbours: start

id: esk1

iceUnit: end

```

bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

```

id: esk2
iceUnit: end
bodyHeat: 5
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: basicPull2
tent: null
components: null
moveBehaviour: walk2

```

```

id: pb1
iceUnit: end

```

### **8.2.35    Jegesmedve és karakter találkozása úgy, hogy a jégmező betörök**

- **Leírás**

Létrehozunk egy stabil, és egy instabil jégtáblát. Az instabil jégtáblán elhelyezünk egy búvárruhás eszkimót, a stabil jégtáblára pedig egy jegesmedvét. Az instabil jégtábla kapacitását annak létrehozásakor kettőre állítjuk. Használjuk az eszkimó képességét az instabil jégtáblán, és építünk rá egy iglut. A jegesmedvét átmozgatjuk az instabil jégtáblára.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ennek a teszesetnek az az elsődleges célja, hogy ellenőrizze, hogy megfelelő-e a működés abban a kissé összetett esetben, amikor egy jegesmedve és egy karakter találkozik, azonban beszakad alattuk a jégtábla. Ebben az esetben először az esés zajlik le, majd utána csak a medve támadása a karakter ellen. A karakter nem halhat bele az esésbe, hiszen búvárruhát visel; ugyanakkor a medvetámadás számára végzetes. A teszt tehát sikertelen, ha a karakter nem hal meg, ezzel a játék végét okozva. Az is hibás működés jele, ha az esés, és a medvetámadás is végzetes a karakter számára. Az is hiba, ha egy objektum valamilyen tulajdonsága változik, amire nem számítunk.

- **Bemenet**

```

field stable 0
field unstable false 2 0
neighbour stable unstable
neighbour unstable stable
entity esk Eskimo unstable
entity pb Bear stable
item ds1 DivingSuit setcharacter esk
ability esk unstable
stats stable

```

stats unstable  
stats esk  
stats pb  
move pb unstable  
stats stable  
stats unstable  
stats esk  
stats pb

- **Elvárt kimenet**

id: stable  
snow: 0  
building: plains1  
pickable: null  
standers: pb  
pullers: null  
masters: null  
neighbours: unstable

id: unstable  
snow: 0  
building: igloo1  
pickable: null  
standers: esk  
pullers: esk  
masters: esk  
neighbours: stable

id: esk  
iceUnit: unstable  
bodyHeat: 5  
stamina: 3  
digger: bareHand1  
waterProtector: ds1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: amph1

id: pb1  
iceUnit: stable

Game Over!  
id: stable  
snow: 0  
building: plains1  
pickable: null  
standers: null  
pullers: null  
masters: null  
neighbours: unstable

```
id: unstable
snow: 0
building: null
pickable: null
standers: esk, pb
pullers: esk
masters: esk
neighbours: stable
```

```
id: esk
iceUnit: unstable
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtecter: ds1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: amph1
```

```
id: pb1
iceUnit: unstable
```

### **8.2.36 Havat ásunk a mezőn alapvető ásó eszközzel**

- **Leírás**

Létrehozunk egy jégtáblát néhány egység hóval, valamint a jégtáblára rárakunk egy karaktert, ami most egy eszkimó lesz. Nem adunk neki speciális ásó eszközt, így az alapvető eszközzel (kézzel) fog ájni. Ezután a karakterrel ásunk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt főcélja annak ellenőrzése, hogy az ásás hatására megfelelően változik-e a jégtáblán lévő hómennyiségek. Az alapvető ásó eszközzel való ásáskor azt várjuk el, hogy egyszerre 1 egység havat lehessen ellapoltani. Hibahely lehet, ha nem tudunk havat lapátolni, vagy nem megfelelő mennyiségű havat lapátolunk el. Ezen kívül az is hibát jelenthet az is, ha nem kezeli a jégtábla megfelelően az ásást 0 egységnyi hó esetén.

- **Bemenet**

```
field spawn 2
entity esk Eskimo spawn
stats spawn
stats esk
dig esk
stats spawn
stats esk
```

- **Elvárt kimenet**

id: spawn  
snow: 2  
building: plains1  
pickable: null  
standers: esk  
pullers: esk  
masters: esk  
neighbours: null

id: esk  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: spawn  
snow: 1  
building: plains1  
pickable: null  
standers: esk  
pullers: esk  
masters: esk  
neighbours: null

id: esk  
iceUnit: spawn  
bodyHeat: 5  
stamina: 3  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: spawn  
snow: 0  
building: plains1  
pickable: null  
standers: esk  
pullers: esk  
masters: esk  
neighbours: null

id: esk

```
iceUnit: spawn
bodyHeat: 5
stamina: 2
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null
```

```
id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 1
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

### **8.2.37 Havat ásunk a mezőn ásóval**

- **Leírás**

Létrehozunk egy jégtáblát néhány egység hóval, valamint a jégtáblára rárakunk egy karaktert, ami most egy eszkimó lesz. A karakter ásó eszköze egy sima ásó lesz. Ezután a karakterrel ásunk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt főcélja annak ellenőrzése, hogy az ásás hatására megfelelően változik-e a jégtáblán lévő hómennyiség. A sima ásóval való ásáskor azt várjuk el, hogy egyszerre 2 egység havat lehessen ellapárolni, valamint a törékeny ásóval ellentétben ezt bármennyiszer megtehessük. Hibahely lehet, ha nem tudunk havat lapárolni, vagy nem megfelelő mennyiségű havat lapátolunk el. Ezen kívül az is hibát jelenthet az is, ha nem kezeli a jégtábla megfelelően az ásást 0 vagy 1 egységnyi hó esetén.

- **Bemenet**

```
field spawn 3
entity esk Eskimo spawn
item shovel Shovel setcharacter esk
stats spawn
stats esk
dig esk
```

stats spawn  
stats esk  
dig esk  
stats spawn  
stats esk  
dig esk  
stats spawn  
stats esk

• **Elvárt kimenet**

id: spawn  
snow: 3  
building: plains1  
pickable: null  
standers: esk  
pullers: esk  
masters: esk  
neighbours: null

id: esk  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: shovel1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: spawn  
snow: 1  
building: plains1  
pickable: null  
standers: esk  
pullers: esk  
masters: esk  
neighbours: null

id: esk  
iceUnit: spawn  
bodyHeat: 5  
stamina: 3  
digger: shovel1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: spawn

```
snow: 0
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null
```

```
id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 2
digger: shovel1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null
```

```
id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 1
digger: shovel1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

### **8.2.38 Havat ásunk a mezőn törékeny ásóval**

- **Leírás**

Létrehozunk egy jégtáblát néhány egység hóval, valamint a jégtáblára rárakunk egy karaktert, ami most egy eszkimó lesz. A karakter ásó eszköze egy törékeny ásó lesz. Ezután a karakterrel ásunk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt főcélja annak az ellenőrzése, hogy az ásás hatására megfelelően változik-e a jégtáblán lévő hómennyiségek, valamint az, hogy a törékeny ásó néhány használat után eltörök-e. A törékeny ásóval való ásáskor azt várjuk el, hogy egyszerre 2 egység havat lehessen ellapítani, de 3 használat után eltörök. Hibahely lehet, ha nem tudunk havat lapátolni, vagy nem megfelelő mennyiségű havat lapátolunk el, vagy ha nem török el az ásó. Ezen kívül az is

hibát jelenthet az is, ha nem kezeli a jégtábla megfelelően az ásást 0 vagy 1 egységnnyi hó esetén.

- **Bemenet**

```
field spawn 3
entity esk Eskimo spawn
item shovel1 BreakableShovel setcharacter esk
stats spawn
stats esk
dig esk
stats spawn
stats esk
```

- **Elvárt kimenet**

```
id: spawn
snow: 3
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null

id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: shovel1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: spawn
snow: 1
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null
```

```
id: esk
iceUnit: spawn
bodyHeat: 5
```

stamina: 3  
digger: shovel1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: spawn  
snow: 0  
building: plains1  
pickable: null  
standers: esk  
pullers: esk  
masters: esk  
neighbours: null

id: esk  
iceUnit: spawn  
bodyHeat: 5  
stamina: 2  
digger: shovel1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: spawn  
snow: 0  
building: plains1  
pickable: null  
standers: esk  
pullers: esk  
masters: esk  
neighbours: null

id: esk  
iceUnit: spawn  
bodyHeat: 5  
stamina: 1  
digger: bareHand2  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

### **8.2.39 Ásó eszközt ásunk ki**

- **Leírás**

A különböző ásó eszközökhöz egyenként létrehozunk egy jégtáblát, és rájuk egy-egy eszkimót. minden jégtáblára egy különböző ásó eszközt rakunk, majd az eszkimókkal ezeket kiássuk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy az egyes ásó fajtákat megfelelően ki tudjuk-e ásni, valamint az, hogy az eszközök felvétele megtörténik-e. Hibahely lehet, ha a jégtábla nem kezeli helyesen a tárgy kiásását, vagy ha a karakter nem képes a tárgyat megfelelően felvenni, de az is hibát jelenthet, ha egy olyan tulajdonság változik, amire nem számítunk.

- **Bemenet**

```
field spawn1 0
field spawn2 0
entity esk1 Eskimo spawn1
entity esk2 Eskimo spawn2
item shovel Shovel setfield spawn1
item breakable BreakableShovel setfield spawn2
stats spawn1
stats spawn2
stats esk1
stats esk2
dig esk1
stats spawn1
stats esk1
dig esk2
stats spawn2
stats esk2
```

- **Elvárt kimenet**

```
id: spawn1
snow: 0
building: plains1
pickable: shovel
standers: esk1
pullers: esk1
masters: esk1
neighbours: null

id: spawn2
snow: 0
building: plains2
pickable: breakable
standers: esk2
pullers: esk2
masters: esk2
neighbours: null
```

```
id: esk1
iceUnit: spawn1
bodyHeat: 5
```

stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: esk2  
iceUnit: spawn2  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: null  
moveBehaviour: walk2

id: spawn1  
snow: 0  
building: plains1  
pickable: null  
standers: esk1  
pullers: esk1  
masters: esk1  
neighbours: null

id: esk1  
iceUnit: spawn1  
bodyHeat: 5  
stamina: 3  
digger: shovel  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: spawn2  
snow: 0  
building: plains2  
pickable: null  
standers: esk2  
pullers: esk2  
masters: esk2  
neighbours: null

id: esk2  
iceUnit: spawn2

bodyHeat: 5  
 stamina: 3  
 digger: breakable  
 waterProtector: lung2  
 pullerDevice: basicPull2  
 tent: null  
 components: null  
 moveBehaviour: walk2

### **8.2.40 Sátrat ásunk ki**

- **Leírás**

Létrehozunk egy jégtáblát egy eszkimóval. A jégtábla egy sárat rakunk. Ezután az eszkimóval a jégtáblán ásunk és felvesszük a sárat.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy a sárat megfelelően ki tudjuk-e ájni, valamint az, hogy a sátor felvétele megtörténik-e. Hibahely lehet, ha a jégtábla nem kezeli helyesen a tárgy kiásását, vagy ha a karakter nem képes a tárgyat megfelelően felvenni, de az is hibát jelenthet, ha egy olyan tulajdonság változik, amire nem számítunk.

- **Bemenet**

field spawn 0  
 entity esk Eskimo spawn  
 item tent1 Tent setfield spawn  
 stats spawn  
 stats esk  
 dig esk  
 stats spawn  
 stats esk

- **Elvárt kimenet**

id: spawn  
 snow: 0  
 building: plains1  
 pickable: tent1  
 standers: esk  
 pullers: esk  
 masters: esk  
 neighbours: null

id: esk  
 iceUnit: spawn  
 bodyHeat: 5  
 stamina: 4  
 digger: bareHand1  
 waterProtector: lung1  
 pullerDevice: basicPull1  
 tent: null  
 components: null  
 moveBehaviour: walk1

```

id: spawn
snow: 0
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null

```

```

id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: tent1
components: null
moveBehaviour: walk1

```

### **8.2.41 Kötelet ásunk ki**

- **Leírás**

Létrehozunk egy jégtáblát egy eszkimóval. A jégtáblába egy kötelet rakunk. Ezután az eszkimóval a jégtáblán ásunk és felvesszük a kötelet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy a kötelet megfelelően ki tudjuk-e ásni, valamint az, hogy a kötél felvétele megtörténik-e. Hibahely lehet, ha a jégtábla nem kezeli helyesen a tárgy kiásását, vagy ha a karakter nem képes a tárgyat megfelelően felvenni, de az is hibát jelenthet, ha egy olyan tulajdonság változik, amire nem számítunk.

- **Bemenet**

```

field spawn 0
entity esk Eskimo spawn
item rope1 Rope setfield spawn
stats spawn
stats esk
dig esk
stats spawn
stats esk

```

- **Elvárt kimenet**

```

id: spawn
snow: 0
building: plains1
pickable: rope1
standers: esk
pullers: esk
masters: esk
neighbours: null

```

```

id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

```

id: spawn
snow: 0
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null

```

```

id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: rope1
tent: null
components: null
moveBehaviour: walk1

```

### **8.2.42 Búvárruhát ásunk ki**

- **Leírás**

Létrehozunk egy jégtáblát egy eszkimóval. A jégtáblába egy búvárruhát rakunk. Ezután az eszkimóval a jégtáblán ásunk és felvesszük a búvárruhát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy a búvárruhát megfelelően ki tudjuk-e ásni, valamint az, hogy a búvárruha felvétele megtörténik-e. Hibahely lehet, ha a jégtábla nem kezeli helyesen a tárgy kiásását, vagy ha a karakter nem képes a tárgyat megfelelően felvenni, de az is hibát jelenthet, ha egy olyan tulajdonság változik, amire nem számítunk.

- **Bemenet**

```

field spawn 0
entity esk Eskimo spawn
item suit1 Suit setfield spawn
stats spawn
stats esk
dig esk
stats spawn
stats esk

```

- **Elvárt kimenet**

```
id: spawn
snow: 0
building: plains1
pickable: suit1
standers: esk
pullers: esk
masters: esk
neighbours: null
```

```
id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null
```

```
id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: suit1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: amph1
```

### **8.2.43 Jelzőpisztoly komponenst ásunk ki**

- **Leírás**

A különböző komponensekhez egyenként létrehozunk egy jégtáblát, és rájuk egy-egy eszkimót. minden jégtábla egy különböző komponenst rakunk, majd az eszkimókkal ezeket kiássuk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy az egyes komponenseket megfelelően ki tudjuk-e ásni, valamint az, hogy a tárgyak felvétele megtörténik-e. Hibahely lehet, ha a jégtábla nem kezeli helyesen a tárgy kiásását, vagy ha a karakter nem képes a tárgyat megfelelően felvenni, de az is hibát jelenthet, ha egy olyan tulajdonság változik, amire nem számítunk.

- **Bemenet**

```
field spawn1 0
field spawn2 0
field spawn3 0
entity esk1 Eskimo spawn1
entity esk2 Eskimo spawn2
entity esk3 Eskimo spawn3
item ps Pistol setfield spawn1
item cr Cartridge setfield spawn2
item lg Light setfield spawn3
stats spawn1
stats spawn2
stats spawn3
stats esk1
stats esk2
stats esk3
dig esk1
stats spawn1
stats esk1
dig esk2
stats spawn2
stats esk2
dig esk3
stats spawn3
stats esk3
```

- **Elvárt kimenet**

```
id: spawn1
snow: 0
building: plains1
pickable: ps
standers: esk1
pullers: esk1
masters: esk1
neighbours: null
```

```
id: spawn2
snow: 0
building: plains2
pickable: cr
standers: esk2
pullers: esk2
masters: esk2
neighbours: null
```

id: spawn3  
snow: 0  
building: plains3  
pickable: lg  
standers: esk3  
pullers: esk3  
masters: esk3  
neighbours: null

id: esk1  
iceUnit: spawn1  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: esk2  
iceUnit: spawn2  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: null  
moveBehaviour: walk2

id: esk3  
iceUnit: spawn3  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: null  
moveBehaviour: walk2

id: spawn1  
snow: 0  
building: plains1  
pickable: null  
standers: esk1  
pullers: esk1  
masters: esk1  
neighbours: null

id: esk1  
iceUnit: spawn1  
bodyHeat: 5  
stamina: 3  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: ps  
moveBehaviour: walk1

id: spawn2  
snow: 0  
building: plains2  
pickable: null  
standers: esk2  
pullers: esk2  
masters: esk2  
neighbours: null

id: esk2  
iceUnit: spawn2  
bodyHeat: 5  
stamina: 3  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: cr  
moveBehaviour: walk2

id: spawn3  
snow: 0  
building: plains3  
pickable: null  
standers: esk3  
pullers: esk3  
masters: esk3  
neighbours: null

id: esk3  
iceUnit: spawn3  
bodyHeat: 5  
stamina: 3  
digger: bareHand3  
waterProtector: lung3  
pullerDevice: basicPull3  
tent: null  
components: lg

moveBehaviour: walk3

### **8.2.44 Ételt ásunk ki maximális testhőn**

- **Leírás**

Létrehozunk egy jégtáblát egy eszkimóval. A jégtáblába egy ételt rakunk. Ezután az eszkimóval a jégtáblán ásunk és felveszük az ételt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy az ételt megfelelően ki tudjuk-e ásni és nem tölt vissza testhőt. Hibahely lehet, ha a jégtábla nem kezeli helyesen a tárgy kiásását, vagy ha az étel felvételekor a testhő a maximális érték fölé emelkedik, de az is hibát jelenthet, ha egy olyan tulajdonság változik, amire nem számítunk.

- **Bemenet**

```
field spawn 0
entity esk Eskimo spawn
item food1 Food setfield spawn
stats spawn
stats esk
dig esk
stats spawn
stats esk
```

- **Elvárt kimenet**

```
id: spawn
snow: 0
building: plains1
pickable: food1
standers: esk
pullers: esk
masters: esk
neighbours: null
```

```
id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
```

neighbours: null

```

id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

### **8.2.45 Ételt ásunk ki nem maximális testhőn**

- **Leírás**

Létrehozunk egy jégtáblát egy eszkimóval. A jégtáblába egy ételt rakunk. Ezután viharral kétszer lecsapunk az eszkimóra, a jégtáblán ásunk és felvesszük az ételt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy az ételt megfelelően ki tudjuk-e ásni, valamint az, hogy a testhő megfelelően növekedik-e. Hibahely lehet, ha a jégtábla nem kezeli helyesen a tárgy kiásását, vagy ha az étel felvételekor nem a megfelelő mértékben növeli a testhőt. Az is hibát jelenthet, ha a vihar nem csökkenti megfelelően a testhőt, vagy ha egy olyan tulajdonság változik, amire nem számítunk.

- **Bemenet**

```

field spawn 0
entity esk Eskimo spawn
item food1 Food setfield spawn
stats spawn
stats esk
storm esk
storm esk
stats esk
dig esk
stats spawn
stats esk

```

- **Elvárt kimenet**

```

id: spawn
snow: 0
building: plains1
pickable: food1
standers: esk
pullers: esk
masters: esk
neighbours: null

```

```

id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 4

```

```
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: esk
iceUnit: spawn
bodyHeat: 3
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null
```

```
id: esk
iceUnit: spawn
bodyHeat: 4
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

### **8.2.46 Tárgy felvétele hóval fedett jégtábláról**

- **Leírás**

Létrehozunk egy jégtáblát egy eszkimóval. A jégtáblába valamilyen tárgyat rakunk (most sárat), valamint a jégtáblát 1 egység hóval lefedjük. Ezután az eszkimóval egymás után kétszer ásunk, hogy felvehessük a tárgyat.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy egy hóval fedett jégtábláról való tárgyfelvétel helyesen működik-e. Hibás működést jelent, ha a tárgyat akkor is fel lehet venni, ha még van hó a jégtáblán, vagy ha nem sikerül felvenni, ha már nem fedi hó a jégtáblát. Az is hibát jelenthet, ha egy olyan tulajdonság változik, amire nem számítunk.

- **Bemenet**

field spawn 1  
entity esk Eskimo spawn  
item tent1 Tent setfield spawn  
stats spawn  
stats esk  
dig esk  
stats spawn  
stats esk  
dig esk  
stats spawn  
stats esk

- **Elvárt kimenet**

id: spawn  
snow: 1  
building: plains1  
pickable: tent1  
standers: esk  
pullers: esk  
masters: esk  
neighbours: null

id: esk  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: spawn  
snow: 0  
building: plains1  
pickable: tent1  
standers: esk  
pullers: esk  
masters: esk  
neighbours: null

id: esk  
iceUnit: spawn  
bodyHeat: 5  
stamina: 3  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null

```
moveBehaviour: walk1
```

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null
```

```
id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 2
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: tent1
components: null
moveBehaviour: walk1
```

### **8.2.47 Karakter sátor építése épületlen mezőre**

- **Leírás**

Létrehozunk egy jégtáblát egy eszkimóval. Az eszkimónak egy sárat adunk, majd az eszkimóval lerakjuk a sárat a jégtáblára.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt fő célja annak az ellenőrzése, hogy ha a karakter lerak egy sárat egy épületlen mezőre, akkor azon sikeresen fel kell épülnie egy sátornak. Ha nem épül fel a sátor, az hibás működést jelent. Hibahely lehet még, ha a sátor a karakternél marad, vagy ha egy olyan tulajdonság változik, amire nem számítunk.

- **Bemenet**

```
field spawn 0
entity builder Eskimo spawn
item tent1 Tent setcharacter builder
stats spawn
stats builder
tent builder
stats spawn
stats builder
```

- **Elvárt kimenet**

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: builder
pullers: builder
masters: builder
```

neighbours: null

```

id: builder
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: tent1
components: null
moveBehaviour: walk1

```

```

id: spawn
snow: 0
building: tent1
pickable: null
standers: builder
pullers: builder
masters: builder
neighbours: null

```

```

id: builder
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

### **8.2.48 Karakter sátor építése igluval már rendelkező mezőre**

- **Leírás**

Elhelyezünk egy stabil jégtáblán egy eszkimót. Használjuk az eszkimó képességét, ezzel létrehozunk egy iglut a jégtáblára. Hozzáadunk a karakter eszköztárához egy sárat, majd megpróbáljuk lerakni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy egy sárat nem tudunk ráépíteni egy másik épületre. Tehát a sátorépítési kísérlet meghiúsul, hiszen a jégtáblán található egy iglu. Bármilyen változás hibának minősül.

- **Bemenet**

```

field stable 0
entity esk Eskimo stable
ability esk stable
item tent1 Tent setcharacter esk
stats stable
stats esk
tent esk

```

stats stable

stats esk

- **Elvárt kimenet**

id: spawn

snow: 0

building: igloo1

pickable: null

standers: esk

pullers: esk

masters: esk

neighbours: null

id: esk

iceUnit: spawn

bodyHeat: 5

stamina: 3

digger: bareHand1

waterProtecter: lung1

pullerDevice: basicPull1

tent: tent1

components: null

moveBehaviour: walk1

id: spawn

snow: 0

building: igloo1

pickable: null

standers: esk

pullers: esk

masters: esk

neighbours: null

id: esk

iceUnit: spawn

bodyHeat: 5

stamina: 3

digger: bareHand1

waterProtecter: lung1

pullerDevice: basicPull1

tent: tent1

components: null

moveBehaviour: walk1

### **8.2.49 Karakter sátor építése sátras mezőre**

- **Leírás**

Elhelyezünk egy stabil jégtáblán egy eszkimót. Hozzáadunk a karakter eszköztárához egy sárat, majd elhelyezzük a jégtáblán. Hozzáadunk a karakter eszköztárához egy sárat, majd megpróbáljuk lerakni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy egy sátrat nem tudunk ráépíteni egy másik épületre. Tehát a második sátorépítési kísérlet meghiúsul, nem szabad történnie semmi változásnak; ha történik, az hibát jelent.

- **Bemenet**

```
field stable 0
entity esk Eskimo stable
item tent1 Tent setcharacter esk
tent esk
item tent2 Tent setcharacter esk
stats stable
stats esk
tent esk
stats stable
stats esk
```

- **Elvárt kimenet**

```
id: spawn
snow: 0
building: tent1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null
```

```
id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: tent2
components: null
moveBehaviour: walk1
```

```
id: spawn
snow: 0
building: tent1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null
```

```
id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
```

```

pullerDevice: basicPull1
tent: tent2
components: null
moveBehaviour: walk1

```

### **8.2.50 Karakter sátor építése úszás közben**

- **Leírás**

Elhelyezünk egy betört jégtáblára egy eszkimót. Kibővíjük az eszkimó eszköztárát egy sátorral, és megpróbáljuk elhelyezni a jégtáblán.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az ellenőrzés esődleges célja, hogy biztosítsuk, hogy egy uszó karakter nem tud elhelyezni a jégtábláján sárat. Ha sikerül a sátor elhelyezése, vagy akármilyen nem várt változás történik; az hibás működésre utal.

- **Bemenet**

```

field swimpool 2 true 0
entity esk Eskimo swimpool
item tent1 Tent setcharacter esk
stats swimpool
stats esk
tent esk
stats swimpool
stats esk

```

- **Elvárt kimenet**

```

id: swimpool
snow: 0
capacity: 2
broken: true
building: plains1
pickable: null
standers: esk
pullers: esk
masters: esk
neighbours: null

```

```

id: esk
iceUnit: swimpool
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: tent1
components: null
moveBehaviour: walk1

```

```

id: swimpool
snow: 0
capacity: 2
broken: true
building: plains1

```

pickable: null  
 standers: esk  
 pullers: esk  
 masters: esk  
 neighbours: null

id: esk  
 iceUnit: swimpool  
 bodyHeat: 5  
 stamina: 4  
 digger: bareHand1  
 waterProtector: lung1  
 pullerDevice: basicPull1  
 tent: tent1  
 components: null  
 moveBehaviour: walk1

### **8.2.51 A körátadás megfelelő sorrendben történik karakterek között**

- **Leírás**

Létrehozunk egy jégtáblát, amin elhelyezünk három darab eszkimót. Átadjuk a köröket négyeszer.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy karakterek között a körátadás helyesen működik. Ahogy a játékhoz adjuk hozzá a karaktereket, a karakterek következése a játékban akként alakul. minden körátadás után ellenőrizzük, hogy a megfelelő karakter van-e a körön; ha nem, akkor a teszt sikertelen. További potenciális hiba, ha olyan tulajdonság változik, amire nem számítottunk.

- **Bemenet**

```

field spawn 0
entity esk1 Eskimo spawn
entity esk2 Eskimo spawn
entity esk3 Eskimo spawn
stats field
stats esk1
stats esk2
stats esk3
current
turn
current
turn
current
turn
current
turn
current
turn
stats field
stats esk1
stats esk2
stats esk3
  
```

- Elvárt kimenet

id: spawn  
snow: 0  
building: plains1  
pickable: null  
standers: esk1, esk2, esk3  
pullers: esk1, esk2, esk3  
masters: esk1, esk2, esk3  
neighbours: null

id: esk1  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: esk2  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand2  
waterProtector: lung2  
pullerDevice: basicPull2  
tent: null  
components: null  
moveBehaviour: walk2

id: esk3  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand3  
waterProtector: lung3  
pullerDevice: basicPull3  
tent: null  
components: null  
moveBehaviour: walk3

esk1

esk2

esk3

esk1

esk2

```
id: spawn
snow: 0
building: plains1
pickable: null
standers: esk1, esk2, esk3
pullers: esk1, esk2, esk3
masters: esk1, esk2, esk3
neighbours: null
```

```
id: esk1
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: esk2
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand2
waterProtector: lung2
pullerDevice: basicPull2
tent: null
components: null
moveBehaviour: walk2
```

```
id: esk3
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand3
waterProtector: lung3
pullerDevice: basicPull3
tent: null
components: null
moveBehaviour: walk3
```

### **8.2.52 A körök műlásával a karakterek munkavégző képessége visszaáll az alapértelmezett állapotba**

- Leírás

Létrehozunk egy jégtáblát, és elhelyezünk rajta egy eszkimót. Az eszkimóval ásunk a jégtáblán, ezzel csökkentjük a munkavégző képességét. Ezután átadjuk a kört.

- **Ellenőrzött funkcionálitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy a körátadás visszaállítja-e a karakter munkavégző képességét. Ha ez nem történik meg a körátadás után, az hibás működésre utal. További hiba lehet, ha az ásás hatására a karakternek nem csökken a munkavégző képessége, vagy nem változik a hó mennyiség a jégtáblán. További potenciális hiba, ha olyan tulajdonság változik, amire nem számítottunk.

- **Bemenet**

```
field 3 spawn
entity esk Eskimo spawn
stats spawn
stats esk
dig esk
stats spawn
stats esk
current
turn
stats spawn
stats esk
current
```

- **Elvárt kimenet**

```
id: spawn
snow: 3
building: plains1
pickable: null
standers: esk1
pullers: esk1
masters: esk1
neighbours: null
```

```
id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1
```

```
id: spawn
snow: 2
building: plains1
pickable: null
standers: esk1
pullers: esk1
masters: esk1
neighbours: null
```

```

id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 3
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

esk

```

id: spawn
snow: 2
building: plains1
pickable: null
standers: esk1
pullers: esk1
masters: esk1
neighbours: null

```

```

id: esk
iceUnit: spawn
bodyHeat: 5
stamina: 4
digger: bareHand1
waterProtector: lung1
pullerDevice: basicPull1
tent: null
components: null
moveBehaviour: walk1

```

esk

### **8.2.53 A körök műlása megfelelően, és megfelelő időben hat a jegesmedve viselkedésére**

- **Leírás**

Létrehozunk két egymással szomszédos jégtáblát; és még egy, különálló jégtáblát. A különálló jégtáblán elhelyezünk egy eszkimót. A két szomszédos jégtábla egyikén elhelyezünk egy jegesmedvét. Átadjuk a kört.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja, hogy ellenőrizzük, hogy a jegesmedve megfelelően reagál-e az idő műlására, tehát körök eltelésére. Ha a jegesmedve kerül sorra, akkor az elmozdul egy szomszédos mezőre; a tesztesetben egy szomszédos mező van, tehát jósolható, hogy hová kell elmozdulnia. Sikertelen a teszt, ha körátadás után a jegesmedve nem mozdul el. Akkor is hibás a működés, ha a körátadás után nem a karakter köre következik ismét. További potenciális hiba, ha olyan tulajdonság változik, amire nem számítottunk.

- **Bemenet**

field n1 0  
 field n2 0  
 neighbour n1 n2  
 neighbour n2 n1  
 field spawn 0  
 entity esk Eskimo spawn  
 entity pb Bear n1  
 stats n1  
 stats n2  
 stats pb  
 stats spawn  
 stats esk  
 current  
 turn  
 stats n1  
 stats n2  
 stats pb  
 stats spawn  
 stats esk  
 current

- **Elvárt kimenet**

id: n1  
 snow: 0  
 building: plains1  
 pickable: null  
 standers: pb  
 pullers: null  
 masters: null  
 neighbours: n2

id: n2  
 snow: 0  
 building: plains2  
 pickable: null  
 standers: null  
 pullers: null  
 masters: null  
 neighbours: n1

id: spawn  
 snow: 0  
 building: plains3  
 pickable: null  
 standers: esk  
 pullers: esk  
 masters: esk  
 neighbours: null

id: esk

iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null  
components: null  
moveBehaviour: walk1

id: pb  
iceUnit: n1

esk

id: n1  
snow: 0  
building: plains1  
pickable: null  
standers: null  
pullers: null  
masters: null  
neighbours: n2

id: n2  
snow: 0  
building: plains2  
pickable: null  
standers: pb  
pullers: null  
masters: null  
neighbours: n1

id: spawn  
snow: 0  
building: plains3  
pickable: null  
standers: esk  
pullers: esk  
masters: esk  
neighbours: null

id: esk  
iceUnit: spawn  
bodyHeat: 5  
stamina: 4  
digger: bareHand1  
waterProtector: lung1  
pullerDevice: basicPull1  
tent: null

components: null

moveBehaviour: walk1

id: pb

iceUnit: n2

esk

### **8.2.54 A körök műlása megfelelően, és megfelelő időben hat a sáorra**

- **Leírás**

Létrehozunk egy jégtáblát, rajta egy eszkimót és egy sárat, majd átadjuk a kört.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt elsődleges célja annak ellenőrzése, hogy a kör végeztével eltűnik-e a sátor. Ha ez nem történik meg a körátadás után, az hibás működésre utal. További potenciális hiba, ha olyan tulajdonság változik, amire nem számítottunk.

- **Bemenet**

field spawn 0

entity esk Eskimo spawn

building tent1 Tent spawn

stats spawn

current

turn

stats spawn

current

- **Elvárt kimenet**

id: spawn

snow: 0

building: tent1

pickable: null

standers: esk

pullers: esk

masters: esk

neighbours: null

esk

id: spawn

snow: 0

building: plains2

pickable: null

standers: esk

pullers: esk

masters: esk

neighbours: null

esk

### **8.3 A tesztelést támogató programok tervei**

A tesztelést egy segédprogram fogja segíteni. Ez egy Windows PowerShell script lesz, ami a teszesetek automatikus kiértékelését fogja segíteni.

A script működését paraméterek segítségével lehet kontrollálni. Alapvetően kétféle módon lehet futtatni: Képes az összes teszesetet egyszerre, vagy egy előre kiválasztott tesztet külön is kiértékelni.

Amennyiben az összes tesztet ki akarjuk értékelni, akkor a script indításakor a `testall` paramétert kell megadni. Egy tesztet a `test <teszeset neve>` paraméterek segítségével futthatunk le. Amennyiben valamelyik paramétert rosszul adjuk meg, a script kiírja a használható paraméterek listáját azok jelentésével együtt:

Usage:

```
testall - run all tests
test <test name> - run one test
```

Ha a `<test name>` paraméterhez olyan teszesetet adunk meg, amit nem képes megtalálni a script, akkor hibát dob:

```
Error: Test "<test name>" not found
```

Az egyes teszesetek bemenetét és elvárt kimenetét fájlokban olvassa be a script. minden teszesetben tartozik egy `<teszeset neve>.in` és egy `<teszeset neve>.out` szövegfájl. Az előbbi a bemenetet, az utóbbi az elvárt kimenetet tartalmazza. Ezek a fájlok ugyanabban a mappában kell legyenek, mint a script, különben nem fogja megtalálni őket.

A script az egyes teszesetek futtatása után összehasonlítja a programtól kapott kimenetet a teszesetben tartozó elvárt kimenettel. Amennyiben a két kimenet megegyezik, akkor azt a `<teszeset neve> passed` sor kiírásával jelzi a script. Ha bármiféle eltérés van a két kimenet között, akkor a teszt eredménye sikertelen lesz. Ekkor a script kiírja a kapott kimenetet és az elvárt kimenetet is, hogy a felhasználó összehasonlíthassa őket. Ezt az alábbi módon teszi meg:

```
<teszeset neve> failed
Expected:
[elvárt kimenet]
```

Got:

```
[kapott kimenet]
```

Amennyiben a scriptet a `testall` paraméterrel futtatjuk, az összes teszt lefutása után kiír egy statisztikát a sikeresen lefutott tesztek arányáról (pl. 36 / 42).

A segédprogramot azért PowerShell segítségével fogjuk elkészíteni, mert ezzel könnyen lehet imitálni azt, amit egy felhasználó is csinálna a terminálon, és emiatt viszonylag egyszerűen megvalósítható az elvárt működés. Természetesen más nyelveknek is megvan ez a tulajdonsága, de a PowerShell minden Windows operációs rendszeren elérhető, így nem szükséges egyéb eszközök telepítése. A script belső működését két egyszerűbb részre lehet bontani.

Az első rész végzi egy teszeset kiértékelését. Ez a rész egy külön függvény (`test([string]$testname)`) lesz, aminek a paramétere a teszeset neve. A kiértékelést úgy valósítja meg, hogy először beolvassa a teszesetben tartozó input parancsokat a teszeset `.in` fájljából, majd ezeket a Java program szabványos bemenetére küldi. Ezután a Java

programtól kapott kimenetet összehasonlítja a tesztesethez tartozó .out fájlban lévő elvárt kimenettel (erre használható például PowerShell-ben a compare parancs), és a korábban specifikált működést követve kiírja a teszt eredményét. A függvény visszatérési értéke egy logikai változó, ami akkor igaz, ha a teszt sikeresen lefutott.

A második rész feladata az összes teszteset futtatásának elvégzése. Ehhez végigmegy a mappában lévő fájlokban, hogy megkereshesse a tesztesetek nevét. Az előbb leírt `test()` függvény segítségével elvégzi egyenként a tesztek kiértékelését, és a függvény visszatérési értékének segítségével számolja, hány teszt futott le sikeresen.

## 8.4 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.04.10. 15:00	4 óra	Kaposvári Labancz Iuhos Sulyok Traxler	Értekezlet. A feladat értelmezése, megoldási javaslatok és a feladatok kiosztása.
2020.04.10. 19:15	2 óra	Traxler	Tevékenység: Tesztesetek előzetes kidolgozása, vázlat készítése.
2020.04.10. 21:00	2 óra	Iuhos Traxler	Értekezlet: Tesztekkel kapcsolatos észrevételek megbeszélése, tesztnyelv megbeszélése.
2020.04.10. 21:00	2 óra	Kaposvári Labancz Sulyok	Értekezlet: Osztályok és metódusok dokumentálásával kapcsolatos értekezlet, döntés a diagramok típusáról, megjelenéséről.
2020.04.11. 09:00	5 óra	Traxler	Tevékenység: Tesztesetek előzetes kidolgozásának folytatása, a felmerült hibák javítása, a tesztesetek dokumentálása.
2020.04.11. 11:00	2 óra	Labancz	Tevékenység: A korábbi beadott anyagok tanulmányozása, azok alapján a dokumentáció kidolgozása.
2020.04.11. 15:00	3 óra	Traxler	Tevékenység: A tesztesetek részletes kidolgozása, leírása.
2020.04.11. 18:00	2 óra	Kaposvári	Tevékenység:

			Az osztályok dokumentálása, részletes leírások készítése.
2020.04.12. 09:00	4 óra	Sulyok	Tevékenység: Az osztályok dokumentálása, állapotdiagramok rajzolása.
2020.04.12. 15:00	6 óra	Iuhos	Tevékenység: A tesztelést támogató program terveinek kidolgozása.
2020.04.12. 14:00	6 óra	Traxler	Tevékenység: A tesztesetek részletes kidolgozásának folytatása dokumentálása. A dokumentáció felülvizsgálata, megoldási javaslatok tétele.
2020.04.12. 16:00	4 óra	Kaposvári	Tevékenység: Tesztesetek és dokumentáció hibáinak javítása.
2020.04.12. 21:00	1 óra	Iuhos Traxler	Értekezlet: Tesztesetekkel kapcsolatban felmerült kérdések megbeszélése.
2020.04.13. 09:00	4 óra	Labancz	Tevékenység: Dokumentáció bővítése, diagramok szerkesztése, hibák javítása.
2020.04.13. 10:00	4 óra	Sulyok	Tevékenység: Metódusok részletes leírása.
2020.04.13. 10:00	2 óra	Kaposvári	Tevékenység: Aktivitásdiagramok tervezése, rajzolása.
2020.04.14. 13:00	6 óra	Iuhos	Tevékenység: Tesztesetek írása.
2020.04.14. 14:00	5 óra	Kaposvári Labancz Sulyok	Értekezlet: Tesztesetek és dokumentáció együttes átnézése,

			hibák javítása.
2020.04.14. 20:00	2 óra	Kaposvári Labancz Iuhos Sulyok Traxler	Értekezlet. Az aznap elvégzett munka értékelése, a hátralevő feladatok megbeszélése.
2020.04.15. 12:00	3 óra	Labancz	Tevékenység: A dokumentumot egyesítése, formázása.

# **10. Feladat: Prototípus - beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése**

**14 – the\_minesweepers**

Konzulens:  
**Fetter László**

## **Csapattagok**

Iuhos Zoltán	J8LOUL	zoltan.iuhos@gmail.com
Kaposvári Márk Martin	RAYVF8	xmarko45@gmail.com
Sulyok Botond	BM2BBM	botond.sulyok99@gmail.com
Labancz Tamás Dominik	A9HZBF	labancz.tamas@gmail.com
Traxler Bálint	IXNOYB	traxlerbalint@gmail.com

2020.04.27.

## 10. Prototípus beadása

### 10.1 Fordítási és futtatási útmutató

#### 10.1.1 Fájllista

Fájl neve	Méret (B)	Keletkezés ideje	Tartalom
CommunicationPlatform.java	1 069	2020.03.25 15:03	Azon interfész, amelyik az egy mezőn keresztül történő kommunikációt lehetővé tévő függvényeket írja elő.
Pullable.java	895	2020.03.25 15:07	Húzható dolgokat kezelő absztrakt osztály.
Puller.java	757	2020.03.25 15:10	Azon interfész, amelyet megvalósító objektumok rendelkeznek az elhúzás képességével.
Stander.java	2 212	2020.03.25 15:14	Absztrakt osztály, leesni képes dolgok származnak le belőle.
Terrestrial.java	382	2020.03.25 15:21	Azon objektumok interfésze, amelyeknek káros, ha vízben vannak.
Amphibious.java	2 171	2020.03.25 15:24	Osztály, amelyik objektumai mozoghatnak jégtáblán és vízben is.
Collideable.java	866	2020.04.22. 13:04	Ütközni képes objektumok interfésze.
LocationChanger.java	2 461	2020.03.25 15:35	Azon absztrakt osztály, amely képes Pullable objektumokat húzni.
MoveBehaviour.java	972	2020.03.25 15:47	Azon interfész, amely megadja a szükséges függvényeket a különböző mozgásformákhoz.
Walk.java	2 355	2020.03.25 15:51	Ezen osztály objektumai mozoghatnak szárazföldön.
Eskimo.java	2 134	2020.03.25 15:58	Azon osztály, amelyik megvalósítja az eszkimó karaktertípushat.
Explorer.java	2 052	2020.03.25 16:05	Azon osztály, amelyik megvalósítja a sarkkutató karaktertípushat.
Character.java	13 222	2020.03.25 16:11	Az összes irányítható karakter közös működését megvalósító absztrakt osztály.
ComponentMaster.java	1 340	2020.03.25 16:40	Interfész, az implementáló osztályok objektumai képesek komponenseket használni.
Equipper.java	1 276	2020.03.25 16:44	A tárgyakat felvenni képes objektumok interfésze.

FiniteActionDoer.java	500	2020.03.25 16:49	Korlátozott cselekvőképességgel rendelkező objektumok interfésze.
IceBear.java	5 179	2020.04.22 15:51	A jegesmedvét kezelő osztály.
IceUnitManipulator.java	829	2020.03.25 16:52	Interfész, amely biztosítja a jégtáblákon való műveletvégzést.
Mortal.java	2 406	2020.03.25 16:55	Testhőt kezelő absztrakt osztály.
Refugee.java	895	2020.03.25 17:03	A játékban levő menekülteknek az interfésze.
CommandReader.java	7 805	2020.04.23. 14:13	Ez a statikus osztály tartalmazza a játék irányítására használható parancsokat, amiket értelmez és végrehajt.
CommandUtils.java	8 310	2020.04.23. 14:13	Ez az osztály tárolja el a játék objektumait és ezzel az osztályval bármilyen azonosítójú objektumot letudunk kérdezni a jégmezőről.
MapParser.java	6 390	2020.04.23. 14:20	A statikus osztály egyetlen felelőssége egy fájlból betöltött pályának a kezelése és játékba állítása.
ParsedObject.java	2 915	2020.04.23. 14:26	Az osztály felelőssége a Parsable interfészt implementáló osztályok objektumaiból egy olyan HashMap-et tárolni, * amelynek adatait könnyen fel lehet dolgozni, és azt mentési vagy vizsgálati célból lehet használni.
AbilityCommand.java	1 225	2020.04.23. 14:31	A statikus osztálynak az egyetlen felelőssége kezelni az "ability" parancsot.
AssembleCommand.java	943	2020.04.23. 14:40	A statikus osztálynak az egyetlen felelőssége kezelni az "assemble" parancsot.
BuildingCommand.java	3 754	2020.04.23. 14:51	A statikus osztálynak a felelőssége kezelni a

			"building" parancsot, illetve tartalmazza az épülettípusokat.
CurrentCommand.java	538	2020.04.23. 15:02	A CurrentCommand osztály felelőssége megvalósítani, és biztosítani a helyes működését a bemeneti nyelv részét képző * current parancsnak.
DigCommand.java	871	2020.04.23. 15:07	A statikus osztálynak az egyetlen felelőssége kezelni a "dig" parancsot.
EntityCommand.java	5 583	2020.04.23. 15:16	A statikus osztálynak a felelőssége kezelni az "entity" parancsot, illetve tartalmazza az entitásípusokat (azok az objektumok, amik "élnek", tudnak mozogni a pályán).
FieldCommand.java	2 696	2020.04.23. 15:26	A statikus osztálynak a felelőssége kezelni a "field" és a "neighbours" parancsot.
ItemCommand.java	5 551	2020.04.23. 15:31	A statikus osztálynak a felelőssége kezelni az "item" parancsot, illetve tartalmazza a tárgytípusokat.
LoadCommand.java	929	2020.04.23. 15:36	A statikus osztálynak az egyetlen felelőssége kezelni a "load" parancsot.
MoveCommand.java	1 140	2020.04.23. 15:45	A statikus osztálynak az egyetlen felelőssége kezelni a "move" parancsot.
SaveCommand.java	949	2020.04.23. 15:51	A statikus osztálynak az egyetlen felelőssége kezelni a "save" parancsot.
StatsCommand.java	4 017	2020.04.23. 15:58	A statikus osztálynak az egyetlen felelőssége kezelni a "stats" parancsot.
StormCommand.java	1 029	2020.04.23. 16:08	A statikus osztálynak az egyetlen felelőssége kezelni a "storm" parancsot.

TentCommand.java	852	2020.04.23. 16:14	A statikus osztálynak az egyetlen felelőssége kezelni a "tent" parancsot.
TurnCommand.java	479	2020.04.23. 16:25	A CurrentCommand osztály felelőssége megvalósítani, és biztosítani a helyes működését a bementi nyelv részét képző turn parancsnak.
Commandable.java	522	2020.03.25 17:07	Irányítási felületet biztosító interfész.
Disaster.java	355	2020.03.25 17:10	A játékban levő katasztrófáknak az interfésze.
GameController.java	5 106	2020.03.25 17:13	A játék indításáért, irányításáért, befejezéséért és az alapvető működéséért felelős osztály.
Identifiable.java	839	2020.04.23. 17:12	Az osztály felelőssége reprezentálni az azonosítható objektumokat.
Parseable.java	674	2020.04.23. 17:42	Az interfész felelőssége közös működési felületet biztosítani az olyan implementáló osztályok objektumnainak, amelynek adatait fel lehet dolgozni, és azt mentési vagy vizsgálati célból lehet használni.
Storm.java	718	2020.03.25 17:38	A vihart reprezentáló osztály.
Strikeable.java	239	2020.03.25 17:49	A katasztrófák által megcsapható dolgok interfésze.
AlarmPistol.java	432	2020.03.25 17:52	Jelzőpisztolyt kezelő osztály.
Pickable.java	849	2020.03.25 19:26	Felvehető dolgok interfésze.
AlarmPistolComponent.java	2 696	2020.04.22. 15:11	Az osztály felelőssége az, hogy a jelzőfényespisztoly komponenseként belőle összeállítható legyena jelzőfényespisztoly.
Cartridge.java	2 280	2020.03.25 17:57	Jelzőpisztolyt alkotó patron kezelő osztály.
Component.java	1 642	2020.03.25 18:05	Komponenseket kezelő absztrakt osztály.
Light.java	2 272	2020.03.25 18:14	Jelzőpisztolyt alkotó jelzőfényt kezelő osztály.

Pistol.java	2 268	2020.03.25 18:20	Jelzőpisztolyt alkotó pisztolyt kezelő osztály.
Consumable.java	810	2020.03.25 18:23	Az elfogyasztható tárgyak interfész.
Food.java	2 985	2020.03.25 18:26	Felvehető ételt reprezentáló osztály.
BareHand.java	1 575	2020.03.25 18:34	Az egyszerű ásást kezelő osztály.
BreakableShovel.java	2 463	2020.04.22 15:50	Eltörni képes ásót reprezentáló osztály.
Digger.java	2 388	2020.03.25 18:39	Az ásást kezelő absztrakt osztály.
Shovel.java	2 401	2020.03.25 18:43	Ásót kezelő osztály.
DivingSuit.java	2 615	2020.03.25 18:47	Búvárruhát kezelő osztály.
Lung.java	1 984	2020.03.25 18:56	Olyan osztály, amelyik lehetővé teszi a karakter megfulladását, ha nincs rajta védelem.
WaterProtecter.java	1 053	2020.03.25 19:06	Víz elleni védelmet kezelő interfész.
BasicPull.java	1 980	2020.03.25 19:10	Húzásra képtelenséget kezelő osztály.
PullerDevice.java	996	2020.03.25 19:16	A húzást megvalósító eszközök interfész.
Rope.java	2 672	2020.03.25 19:20	Kötelelet kezelő osztály.
Logger.java	2 635	2020.03.28 13:49	A függvényhívások logolásáért felelős osztály.
testAssemble.java	1 596	2020.03.28 14:40	A jelzőpisztoly összeszerelését tesztelő osztály.
testBuildLogIglooAbility.java	874	2020.03.28 15:03	Eszkimó igluépítését tesztelő osztály.
testDigAndEquip.java	1 316	2020.03.28 15:30	Karakter ásását és tárgyfelvételét tesztelő osztály.
testEndTurn.java	772	2020.03.28 16:07	A karakter kör továbbadását tesztelő osztály.
testFallAndPull.java	1 899	2020.03.28 17:11	Esést és kihúzást tesztelő osztály.
testMovement.java	3 356	2020.03.29 14:16	Karakter mozgását tesztelő osztály.
testShowAbility.java	884	2020.03.29 14:20	Sarkkutató képességét tesztelő osztály.
testStartGame.java	530	2020.03.29 16:11	Játék elkezdését tesztelő osztály.
testStormOnIceField.java	1 386	2020.03.29 17:47	Vihart tesztelő osztály.
Building.java	4 931	2020.03.25 19:29	Az épületeket kezelő absztrakt osztály.

BuildingPlot.java	2 850	2020.03.25 19:36	A jégmezőn levő épülethelyeket kezelő absztrakt osztály.
DestructableBuilding.java	2 235	2020.03.25 19:45	Az elpusztítható épületeket kezelő absztrakt osztály.
Igloo.java	1 511	2020.03.25 19:50	Iglu típusú épületeket kezelő osztály.
Plains.java	2 870	2020.03.25 19:54	Olyan épülettípus kezelő osztály, ami nem rendelkezik semmilyen képességgel.
Tent.java	2 904	2020.04.22 16:18	Sátort reprezentáló osztály.
Digsite.java	558	2020.03.25 19:58	Ásható objektumok interfésze.
IceField.java	1 840	2020.03.25 20:01	A játékot felépítő egységeket kezelő osztály.
IceUnit.java	12 467	2020.03.25 20:07	Jégtáblát kezelő osztály.
StormyField.java	215	2020.03.25 20:51	A játékban levő viharos jégmezőknek az interfésze.
UnstableIceUnit.java	5 558	2020.03.25 20:57	Instabil jégtáblát kezelő osztály.
Main.java	308	2020.03.28 13:05	A tesztelés main osztályát tartalmazza, amely kezeli a teszeseteket.

autotester.ps1	2 384	2020.04.24. 13:07	A teszteket futtató szkript.
test1.1.in	102	2020.04.24. 13:09	A névnek megfelelő teszt inputja.
test1.1.out	137	2020.04.24. 13:11	A fájlnévnek megfelelő teszt elvárt outputja.
test1.2.in	25	2020.04.24. 13:12	A névnek megfelelő teszt inputja.
test1.2.out	137	2020.04.24. 13:13	A fájlnévnek megfelelő teszt elvárt outputja.
test10.in	284	2020.04.24. 13:15	A névnek megfelelő teszt inputja.
test10.out	1 158	2020.04.24. 13:16	A fájlnévnek megfelelő teszt elvárt outputja.
test11.in	442	2020.04.24. 13:17	A névnek megfelelő teszt inputja.
test11.out	1 647	2020.04.24. 13:18	A fájlnévnek megfelelő teszt elvárt outputja.
test12.in	213	2020.04.24. 13:20	A névnek megfelelő teszt inputja.
test12.out	697	2020.04.24. 13:22	A fájlnévnek megfelelő teszt elvárt outputja.
test13.in	195	2020.04.24. 13:25	A névnek megfelelő teszt inputja.
test13.out	928	2020.04.24. 13:26	A fájlnévnek megfelelő teszt elvárt outputja.

test14.in	397	2020.04.24. 13:28	A névnek megfelelő teszt inputja.
test14.out	1 668	2020.04.24. 13:29	A fájlnévnek megfelelő teszt elvárt outputja.
test15.in	282	2020.04.24. 13:30	A névnek megfelelő teszt inputja.
test15.out	1 368	2020.04.24. 13:32	A fájlnévnek megfelelő teszt elvárt outputja.
test16.in	115	2020.04.24. 13:33	A névnek megfelelő teszt inputja.
test16.out	625	2020.04.24. 13:35	A fájlnévnek megfelelő teszt elvárt outputja.
test17.in	143	2020.04.24. 13:36	A névnek megfelelő teszt inputja.
test17.out	624	2020.04.24. 13:38	A fájlnévnek megfelelő teszt elvárt outputja.
test18.in	141	2020.04.24. 13:39	A névnek megfelelő teszt inputja.
test18.out	622	2020.04.24. 13:40	A fájlnévnek megfelelő teszt elvárt outputja.
test19.in	122	2020.04.24. 13:42	A névnek megfelelő teszt inputja.
test19.out	688	2020.04.24. 13:44	A fájlnévnek megfelelő teszt elvárt outputja.
test2.1.in	146	2020.04.24. 13:46	A névnek megfelelő teszt inputja.
test2.1.out	421	2020.04.24. 13:47	A fájlnévnek megfelelő teszt elvárt outputja.
test2.2.in	43	2020.04.24. 13:50	A névnek megfelelő teszt inputja.
test2.2.out	421	2020.04.24. 13:51	A fájlnévnek megfelelő teszt elvárt outputja.
test20.in	69	2020.04.24. 13:52	A névnek megfelelő teszt inputja.
test20.out	146	2020.04.24. 13:53	A fájlnévnek megfelelő teszt elvárt outputja.
test21.in	87	2020.04.24. 13:54	A névnek megfelelő teszt inputja.
test21.out	358	2020.04.24. 13:56	A fájlnévnek megfelelő teszt elvárt outputja.
test22.in	221	2020.04.24. 13:57	A névnek megfelelő teszt inputja.
test22.out	1 194	2020.04.24. 13:58	A fájlnévnek megfelelő teszt elvárt outputja.
test23.in	832	2020.04.24. 14:00	A névnek megfelelő teszt inputja.
test23.out	3 624	2020.04.24. 14:02	A fájlnévnek megfelelő teszt elvárt outputja.

test24.in	262	2020.04.24. 14:03	A névnek megfelelő teszt inputja.
test24.out	704	2020.04.24. 14:04	A fájlnévnek megfelelő teszt elvárt outputja.
test25.in	303	2020.04.24. 14:05	A névnek megfelelő teszt inputja.
test25.out	1 464	2020.04.24. 14:08	A fájlnévnek megfelelő teszt elvárt outputja.
test26.in	321	2020.04.24. 14:09	A névnek megfelelő teszt inputja.
test26.out	1 426	2020.04.24. 14:10	A fájlnévnek megfelelő teszt elvárt outputja.
test27.in	590	2020.04.24. 14:11	A névnek megfelelő teszt inputja.
test27.out	1 778	2020.04.24. 14:12	A fájlnévnek megfelelő teszt elvárt outputja.
test28.in	91	2020.04.24. 14:14	A névnek megfelelő teszt inputja.
test28.out	594	2020.04.24. 14:15	A fájlnévnek megfelelő teszt elvárt outputja.
test29.in	171	2020.04.24. 14:16	A névnek megfelelő teszt inputja.
test29.out	1 300	2020.04.24. 14:19	A fájlnévnek megfelelő teszt elvárt outputja.
test3.in	295	2020.04.24. 14:21	A névnek megfelelő teszt inputja.
test3.out	291	2020.04.24. 14:23	A fájlnévnek megfelelő teszt elvárt outputja.
test30.in	240	2020.04.24. 14:25	A névnek megfelelő teszt inputja.
test30.out	1 832	2020.04.24. 14:28	A fájlnévnek megfelelő teszt elvárt outputja.
test31.in	228	2020.04.24. 14:29	A névnek megfelelő teszt inputja.
test31.out	1 830	2020.04.24. 14:31	A fájlnévnek megfelelő teszt elvárt outputja.
test32.in	498	2020.04.24. 14:32	A névnek megfelelő teszt inputja.
test32.out	2 430	2020.04.24. 14:33	A fájlnévnek megfelelő teszt elvárt outputja.
test33.in	236	2020.04.24. 14:35	A névnek megfelelő teszt inputja.
test33.out	962	2020.04.24. 14:36	A fájlnévnek megfelelő teszt elvárt outputja.
test34.in	272	2020.04.24. 14:38	A névnek megfelelő teszt inputja.
test34.out	1 352	2020.04.24. 14:39	A fájlnévnek megfelelő teszt elvárt outputja.

test35.in	295	2020.04.24. 14:42	A névnek megfelelő teszt inputja.
test35.out	1 052	2020.04.24. 14:46	A fájlnévnek megfelelő teszt elvárt outputja.
test36.in	149	2020.04.24. 14:47	A névnek megfelelő teszt inputja.
test36.out	1 188	2020.04.24. 14:48	A fájlnévnek megfelelő teszt elvárt outputja.
test37.in	173	2020.04.24. 14:50	A névnek megfelelő teszt inputja.
test37.out	1 180	2020.04.24. 14:51	A fájlnévnek megfelelő teszt elvárt outputja.
test38.in	176	2020.04.24. 14:52	A névnek megfelelő teszt inputja.
test38.out	1 180	2020.04.24. 14:55	A fájlnévnek megfelelő teszt elvárt outputja.
test39.in	253	2020.04.24. 14:56	A névnek megfelelő teszt inputja.
test39.out	1 216	2020.04.24. 14:58	A fájlnévnek megfelelő teszt elvárt outputja.
test4.in	170	2020.04.24. 15:00	A névnek megfelelő teszt inputja.
test4.out	521	2020.04.24. 15:02	A fájlnévnek megfelelő teszt elvárt outputja.
test40.in	111	2020.04.24. 15:03	A névnek megfelelő teszt inputja.
test40.out	596	2020.04.24. 15:05	A fájlnévnek megfelelő teszt elvárt outputja.
test41.in	111	2020.04.24. 15:07	A névnek megfelelő teszt inputja.
test41.out	590	2020.04.24. 15:08	A fájlnévnek megfelelő teszt elvárt outputja.
test42.in	111	2020.04.24. 15:09	A névnek megfelelő teszt inputja.
test42.out	595	2020.04.24. 15:11	A fájlnévnek megfelelő teszt elvárt outputja.
test43.in	361	2020.04.24. 15:12	A névnek megfelelő teszt inputja.
test43.out	1 806	2020.04.24. 15:13	A fájlnévnek megfelelő teszt elvárt outputja.
test44.in	111	2020.04.24. 15:15	A névnek megfelelő teszt inputja.
test44.out	595	2020.04.24. 15:16	A fájlnévnek megfelelő teszt elvárt outputja.
test45.in	141	2020.04.24. 15:17	A névnek megfelelő teszt inputja.
test45.out	769	2020.04.24. 15:20	A fájlnévnek megfelelő teszt elvárt outputja.

test46.in	141	2020.04.24. 15:22	A névnek megfelelő teszt inputja.
test46.out	894	2020.04.24. 15:24	A fájlnévnek megfelelő teszt elvárt outputja.
test47.in	130	2020.04.24. 15:25	A névnek megfelelő teszt inputja.
test47.out	626	2020.04.24. 15:26	A fájlnévnek megfelelő teszt elvárt outputja.
test48.in	133	2020.04.24. 15:28	A névnek megfelelő teszt inputja.
test48.out	598	2020.04.24. 15:30	A fájlnévnek megfelelő teszt elvárt outputja.
test49.in	143	2020.04.24. 15:33	A névnek megfelelő teszt inputja.
test49.out	596	2020.04.24. 15:34	A fájlnévnek megfelelő teszt elvárt outputja.
test5.in	340	2020.04.24. 15:35	A névnek megfelelő teszt inputja.
test5.out	885	2020.04.24. 15:36	A fájlnévnek megfelelő teszt elvárt outputja.
test50.in	129	2020.04.24. 15:38	A névnek megfelelő teszt inputja.
test50.out	670	2020.04.24. 15:39	A fájlnévnek megfelelő teszt elvárt outputja.
test51.in	243	2020.04.24. 15:40	A névnek megfelelő teszt inputja.
test51.out	1 404	2020.04.24. 15:42	A fájlnévnek megfelelő teszt elvárt outputja.
test52.in	132	2020.04.24. 15:43	A névnek megfelelő teszt inputja.
test52.out	901	2020.04.24. 15:44	A fájlnévnek megfelelő teszt elvárt outputja.
test53.in	233	2020.04.24. 15:46	A névnek megfelelő teszt inputja.
test53.out	1 194	2020.04.24. 15:48	A fájlnévnek megfelelő teszt elvárt outputja.
test54.in	108	2020.04.24. 15:49	A névnek megfelelő teszt inputja.
test54.out	254	2020.04.24. 15:51	A fájlnévnek megfelelő teszt elvárt outputja.
test6.in	182	2020.04.24. 15:53	A névnek megfelelő teszt inputja.
test6.out	942	2020.04.24. 15:55	A fájlnévnek megfelelő teszt elvárt outputja.
test7.in	157	2020.04.24. 15:58	A névnek megfelelő teszt inputja.
test7.out	350	2020.04.24. 16:00	A fájlnévnek megfelelő teszt elvárt outputja.

test8.in	304	2020.04.24. 16:02	A névnek megfelelő teszt inputja.
test8.out	1 138	2020.04.24. 16:04	A fájlnévnek megfelelő teszt elvárt outputja.
test9.in	207	2020.04.24. 16:08	A névnek megfelelő teszt inputja.
test9.out	818	2020.04.24. 16:12	A fájlnévnek megfelelő teszt elvárt outputja.
testmap1.map	516	2020.04.24. 17:15	A map teszteléshez szükséges fájl.
testmap2.map	607	2020.04.24. 17:31	A map teszteléshez szükséges fájl.

### 10.1.2 Fordítás

1. Csomagoljuk ki az projlab-2020.zip mappát.
2. Navigálunk a kicsomagolt projlab-2020 mappába.
3. Nyissunk egy terminált/parancssort az projlab-2020 mappában található src mappához navigálva, majd hajtsuk végre a következő parancsokat:
  - 4a. Linux / MacOS alatt: find -name "\*.java" > sources.txt
  - 4b. Windows alatt: dir /s /B \*.java > sources.txt
5. javac @sources.txt

### 10.1.3 Futtatás

Az előző lépések után az src mappában maradva a terminál / parancssorban ki kell adni a következő parancsot: java hu.minesweepers.Main

Az autotester használatához navigálunk egy PowerShell admin módban futó ablakot a kicsomagolt mappa test mappájába, majd adjuk ki a következő parancsot:

Összes teszt futtatása: .\autotester.ps1 testall

Egy teszt futtatása: .\autotester.ps1 test <teszt száma>

## 10.2 Tesztek jegyzőkönyvei

### 10.2.1 Egy jégtábla betöltésének ellenőrzése

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

### 10.2.2 Jégmező betöltésének ellenőrzése

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

### 10.2.3 Jegesmedve mozgásának az ellenőrzése

Tesztelő neve	Iuhos Zoltán
Teszt időpontja	2020.04.26 14:00

**10.2.4 Karakter szárazföldi mozgásának ellenőrzése**

<b>Tesztelő neve</b>	Iuhos Zoltán
<b>Teszt időpontja</b>	2020.04.26 14:00

**10.2.5 Karakter kételtű mozgásának ellenőrzése**

<b>Tesztelő neve</b>	Iuhos Zoltán
<b>Teszt időpontja</b>	2020.04.26 14:00

**10.2.6 Instabil jégtábla teherbíró képességének, és betörésének vizsgálata**

<b>Tesztelő neve</b>	Iuhos Zoltán
<b>Teszt időpontja</b>	2020.04.26 14:00

**10.2.7 Vizet reprezentáló jégtábla teherbíró képességének, és betörésének vizsgálata**

<b>Tesztelő neve</b>	Iuhos Zoltán
<b>Teszt időpontja</b>	2020.04.26 14:00

**10.2.8 Védelemmel rendelkező karakter esésének vizsgálata**

<b>Tesztelő neve</b>	Iuhos Zoltán
<b>Teszt időpontja</b>	2020.04.26 14:00

**10.2.9 Védelemmel nem rendelkező karakter esésének vizsgálata, környező kihúzók nélkül**

<b>Tesztelő neve</b>	Iuhos Zoltán
<b>Teszt időpontja</b>	2020.04.26 14:00

**10.2.10 Védelemmel nem rendelkező karakter esésének vizsgálata, egy szomszédos mezőn található kihúzával**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00
<b>Teszt eredménye</b>	Failed, StackBufferOverflowException
<b>Lehetséges hibaok</b>	A karakter és/vagy a jégtábla nem érzékeli jól a kihúzott karakter elmozdulását.
<b>Változtatások</b>	Character osztály reposition függvényének bővítése úgy, hogy a karakter IceUnitját átállítsa arra a jégtáblára, ahová a karaktert elhúzták.

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:23

Teszt eredménye	Success
Lehetséges hibaok	-
Változtatások	-

#### 10.2.11 Védelemmel nem rendelkező karakter esésének vizsgálata, több szomszédos mezőn álló kihúzával

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

#### 10.2.12 Jegesmedve vízbeesésének vizsgálata

Tesztelő neve	Iuhos Zoltán
Teszt időpontja	2020.04.26 14:00

#### 10.2.13 Szárazföldi mozgással bíró karakter hóval fedte lukra lép

Tesztelő neve	Iuhos Zoltán
Teszt időpontja	2020.04.26 14:00

#### 10.2.14 Vízben úszó elhúzó viselkedésének ellenőrzése, ha a szomszédos mezőn egy karakter vízbe esik és segítséget kér

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

#### 10.2.15 Karakter alapértelmezett húzóeszközének vizsgálata

Tesztelő neve	Iuhos Zoltán
Teszt időpontja	2020.04.26 14:00

#### 10.2.16 Eszkimó iglu építése épületlen mezőre

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

#### 10.2.17 Eszkimó iglu építése igluval már rendelkező mezőre

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

#### 10.2.18 Eszkimó iglu építése sátras mezőre

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

**10.2.19 Úszó Eszkimó iglut próbál építeni**

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

**10.2.20 Explorer felfedi egy stabil mező teherbíró képességét**

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

**10.2.21 Explorer felfedi egy instabil mező teherbíró képességét**

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

**10.2.22 Egy karakter összerakja a saját komponenseiből (pont elég)**

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

**10.2.23 Egy karakter nem rakja össze, mert tetszőlegesen hiányzik valami**

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

**10.2.24 Egy karakter összerakja bőven több komponensből**

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

**10.2.25 Több karakter összerakja együtt a pont megfelelő mennyiséggű és típusú komponensekből a jelzőpisztolyt**

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

**10.2.26 Több karakter nem rakja össze a jelzőpisztolyt, mert hiányzik egy komponens típus**

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

**10.2.27 Több karakter összeszereli a jelző pisztolyt a szükségesnél sokkal több komponensből**

Tesztelő neve	Traxler Bálint
Teszt időpontja	2020.04.26. 14:00

**10.2.28 A vihar lecsap egy karakterre, és nincs épület a mezőn**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.29 A vihar lecsap egy karakterre, és nincs épület a mezőn**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.30 A vihar lecsap egy karakterre, és sátor van a mezőn  
(van+nincs hely)**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.31 A vihar lecsap egy karakterre, és iglu van a mezőn  
(van+nincs hely)**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.32 Jegesmedve karaktert támad épület nélküli mezőn**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.33 Jegesmedve karaktert támad sátras mezőn**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.34 Jegesmedve karaktert támad iglus mezőn**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.35 Jegesmedve és karakter találkozása úgy, hogy a jégmező betörik**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.36 Havat ásunk a mezőn alapvető ásó eszközzel**

<b>Tesztelő neve</b>	Traxler Bálint
----------------------	----------------

<b>Teszt időpontja</b>	2020.04.26. 14:00
------------------------	-------------------

**10.2.37 Havat ásunk a mezőn ásóval**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.38 Havat ásunk a mezőn törékeny ásóval**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.39 Ásó eszközt ásunk ki**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.40 Sárat ásunk ki**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.41 Kötelet ásunk ki**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.42 Búvárruhát ásunk ki**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.43 Jelzőpisztoly komponenst ásunk ki**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.44 Ételt ásunk ki maximális testhőn**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.45 Ételt ásunk ki nem maximális testhőn**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.46 Tárgy felvétele hóval fedett jégtábláról**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.47 Karakter sátor építése épületlen mezőre**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.48 Karakter sátor építése igluval már rendelkező mezőre**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.49 Karakter sátor építése sátras mezőre**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.50 Karakter sátor építése úszás közben**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26. 14:00

**10.2.51 A körátadás megfelelő sorrendben történik karakterek között**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26 14:00
<b>Teszt eredménye</b>	Failed, a program NullPointerExceptiont dobott
<b>Lehetséges hibaok</b>	Nincs helyes módon inicializálva a Gamecontroller, vagy annak attribútumainak a beállítása
<b>Változtatások</b>	GameController addCommandable nevű metódusának átírása, Main osztály main függvényének megváltoztatása

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26 14:45
<b>Teszt eredménye</b>	Success
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	Az előző változtatások megoldották a problémát.

**10.2.52 A körök műlásával a karakterek munkavégző képessége visszaáll az alapértelmezett állapotba**

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26 14:00
<b>Teszt eredménye</b>	Failed, a program NullPointerExceptiont dobott
<b>Lehetséges hibaok</b>	Nincs helyes módon inicializálva a Gamecontroller, vagy annak attribútumainak a beállítása

<b>Változtatások</b>	GameController addCommandable nevű metódusának átírása, Main osztály main függvényének megváltoztatása
----------------------	--

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26 14:45
<b>Teszt eredménye</b>	Success
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	Az előző változtatások megoldották a problémát.

### 10.2.53 A körök műlása megfelelően, és megfelelő időben hat a jegesmedve viselkedésére

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26 14:00
<b>Teszt eredménye</b>	Failed, a program NullPointerExceptiont dobott
<b>Lehetséges hibaok</b>	Nincs helyes módon inicializálva a Gamecontroller, vagy annak attribútumainak a beállítása
<b>Változtatások</b>	GameController addCommandable nevű metódusának átírása, Main osztály main függvényének megváltoztatása

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26 14:45
<b>Teszt eredménye</b>	Success
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	Az előző változtatások megoldották a problémát.

### 10.2.54 A körök műlása megfelelően, és megfelelő időben hat a sátorra

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26 14:00
<b>Teszt eredménye</b>	Failed, a program NullPointerExceptiont dobott
<b>Lehetséges hibaok</b>	Nincs helyes módon inicializálva a Gamecontroller, vagy annak attribútumainak a beállítása
<b>Változtatások</b>	GameController addCommandable nevű metódusának átírása, Main osztály main függvényének megváltoztatása

<b>Tesztelő neve</b>	Traxler Bálint
<b>Teszt időpontja</b>	2020.04.26 14:45
<b>Teszt eredménye</b>	Success
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	Az előző változtatások megoldották a problémát.

### 10.3 Értékelés

Tag neve	Tag neptun	Munka százalékban
Iuhos Zoltán	J8LOUL	20 %
Kaposvári Márk Martin	RAYVF8	20 %
Sulyok Botond	BM2BBM	20 %
Labancz Tamás Dominik	A9HZBF	20 %
Traxler Bálint	IXNOYB	20 %

## 10.4 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.04.22. 13:00	4 óra	Kaposvári Labancz Iuhos Sulyok Traxler	Értekezlet. A feladat értelmezése, megoldási javaslatok és a feladatok kiosztása. Fájlok létrehozása.
2020.04.23. 17:00	5 óra	Sulyok	Tevékenység: Játékmechanika implementálása a korábbi tervezetek alapján.
2020.04.24. 09:00	3 óra	Sulyok Traxler	Értekezlet. A program apróbb hibáinak megbeszélése és javítása.
2020.04.24. 13:00	7 óra	Iuhos	Tevékenység: Parancsértelmező, tesztelő leprogramozása.
2020.04.25. 09:00	6 óra	Iuhos	Tevékenység: Teszteléshez kapcsolódó programok egy részének implementálása.
2020.04.25 09:30	5 óra	Traxler	Tevékenység: Teszteléshez kapcsolódó programok egy részének implementálása.
2020.04.25 15:30	7 óra	Labancz	Tevékenység: Teszteléshez kapcsolódó fájlok készítése, a tesztek ellenőrzése, javítása.
2020.04.25 16:30	7 óra	Kaposvári	Tevékenység: Teszteléshez kapcsolódó fájlok készítése, a tesztek ellenőrzése, javítása.
2020.04.26. 09:30	4 óra	Sulyok	Tevékenység: Javadoc kommentek írása a programba. (a)

			játékmechanikához kapcsolódó osztályokban)
2020.04.26. 10:30	4 óra	Kaposvári	Tevékenység: Javadoc kommentek írása a programba. (a teszteléshez kapcsolódó osztályokban)
2020.04.26 14:00	1 óra	Traxler	Tevékenység: Tesztelés.
2020.04.26 14:00	1 óra	Iuhos	Tevékenység: Tesztelés.
2020.04.26. 15:00	2 óra	Kaposvári Labancz	Értekezlet: Az eddig elkészült anyagok átnézése.
2020.04.26 15:30	5 óra	Traxler	Tevékenység: Hibajavítások a programban, tesztjegyzőkönyv megírása.
2020.04.26 22:00	1 óra	Iuhos Traxler	Értekezlet. Hibás tesztesetek javítása.
2020.04.27 10:00	1 óra	Iuhos	Tevékenység: Időközben felmerült hibák javítása a parancskezelőben.
2020.04.27 10:00	1 óra	Traxler	Tevékenység: Néhány hibás tesztfájl és javadoc komment javítása.
2020.04.27 10:00	1 óra	Kaposvári	Tevékenység: A teszteléshez kapcsolódó új osztályok javadoc commenttel való ellátása.
2020.04.27 11:00	1 óra	Sulyok	Tevékenység: Az elkészült forráskódok egyesítése.
2020.04.27 10:00	4 óra	Labancz	Tevékenység: Dokumentáció, annak formázása, fájllista elkészítése. Az elkészült anyagok feltöltése.

# **11. Feladat: Grafikus felület specifikációja - beadás**

**14 – the\_minesweepers**

Konzulens:  
**Fetter László**

## **Csapattagok**

Iuhos Zoltán	J8LOUL	zoltan.iuhos@gmail.com
Kaposvári Márk Martin	RAYVF8	xmarko45@gmail.com
Sulyok Botond	BM2BBM	botond.sulyok99@gmail.com
Labancz Tamás Dominik	A9HZBF	labancz.tamas@gmail.com
Traxler Bálint	IXNOYB	traxlerbalint@gmail.com

2020.05.04.

## 11. Grafikus felület specifikációja

### 11.1 A grafikus interfész

Miután elindítjuk a szoftvert, a játék főmenüje jelenik meg. A főmenüben lehetőségünk nyílik elindítani a játékot, és erre előkészületeket tenni. Ilyen előkészület például annak **beállítása, hogy hány karakterrel**; tehát hány eszkimóval, és felfedezővel szeretnénk játszani az indítandó játékot. Lehetőségünk van továbbá a „**How to play**” gombra kattintva tanulmányoznunk a játék szabályait, illetve annak az irányítását. Ha ezeket a lépéseket megtettük; akkor a „**New game**” gombra kattintva elkezdhetünk játszani a játékkal. Meg fog jelenni a játéknak az ablaka, ahol leírt irányítási szabályoknak megfelelően játszhatunk vele.



A játékpálya megjelenítéséhez a következő képeket fogjuk használni:

- **Cartride (töltény):**



- **Divingsuit (búvárruha):**



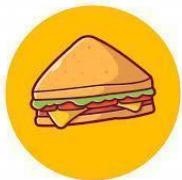
- **Eskimo (eszkimo):**



- **Explorer (sarkkutató):**



- **Food (étel):**



- **Icebear (jegesmedve):**



- **Igloo (iglu):**



- **Light (jelzőfény):**



- **Pistol (jelzőpisztoly):**



- **Rope (kötél):**



- **Shovel (ásó):**



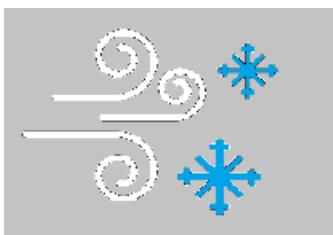
- **Tent (sátor):**



- **Water (víz):**



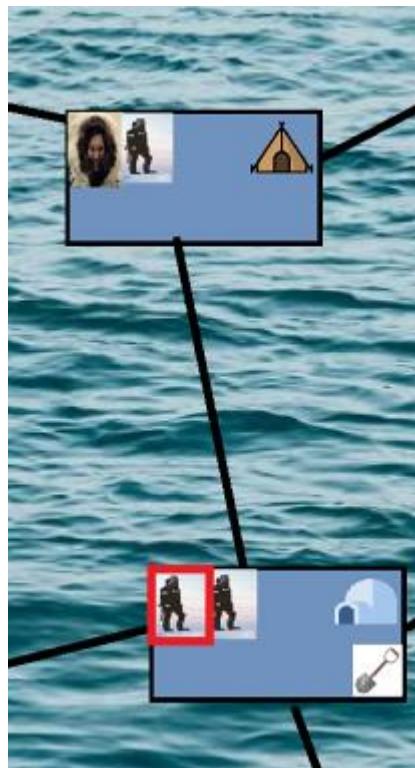
- **Storm (hóvihar)**



A játékpálya az információkat tehát a fenti képek segítségével fogja megjeleníteni. Ezek az információk sokszor összetartoznak. Ilyen például, amikor a játékban egy jégtáblát jellemzünk, ekkor a hozzá tartozó adatokat és információkat jégtábla egységenként fogjuk megjeleníteni, hasonlóan a következő példához:

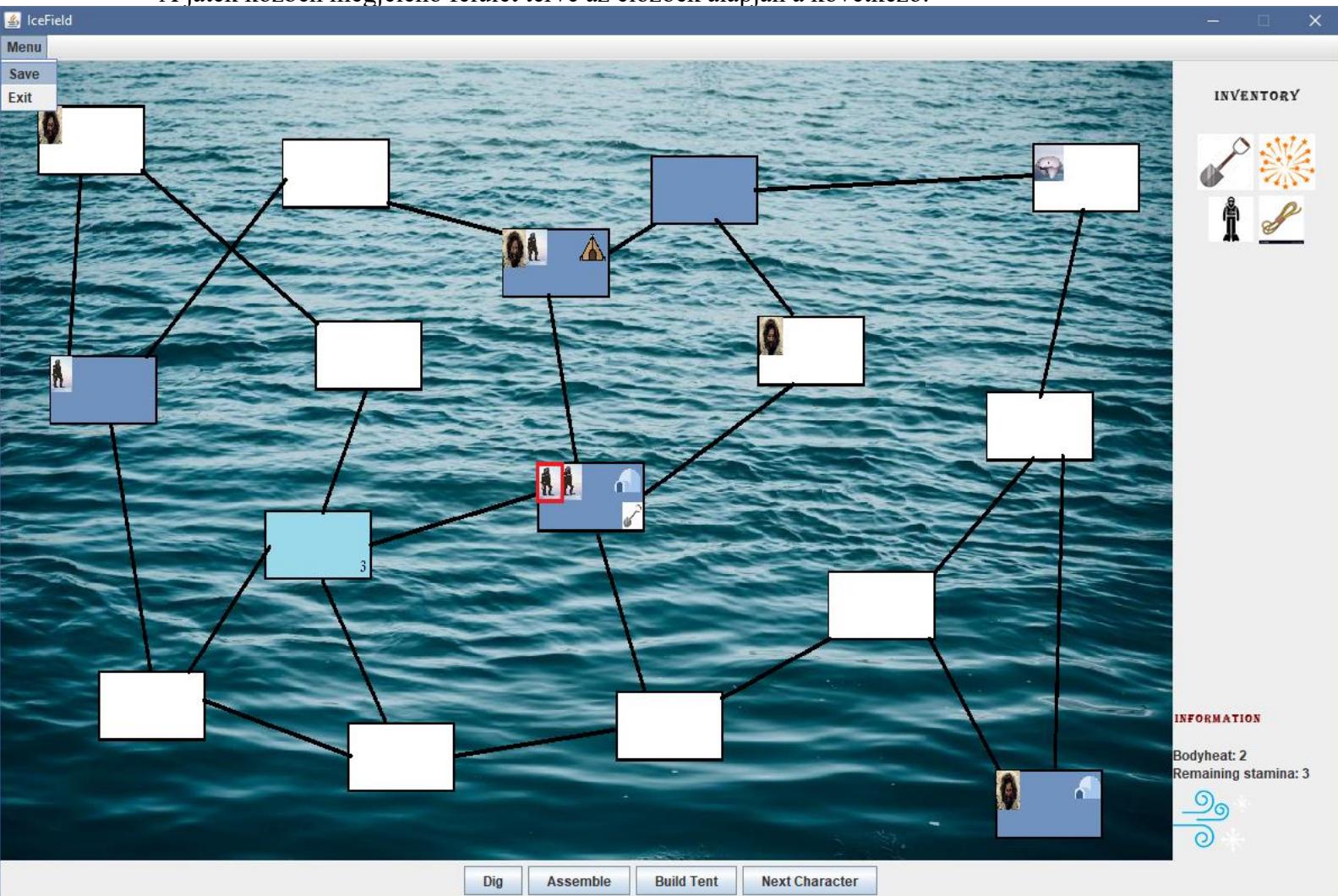


Fontos alapelveink, hogy támogassuk a grafikus felhasználói interfésszel azt, hogy akármelyik jégtáblának a játékban tetszőleges számú szomszédos jégtáblája legyen. Ezt úgy valósítjuk meg, hogy amennyiben szomszédos két jégtábla, abban az esetben vonalat rajzolunk közéjük.



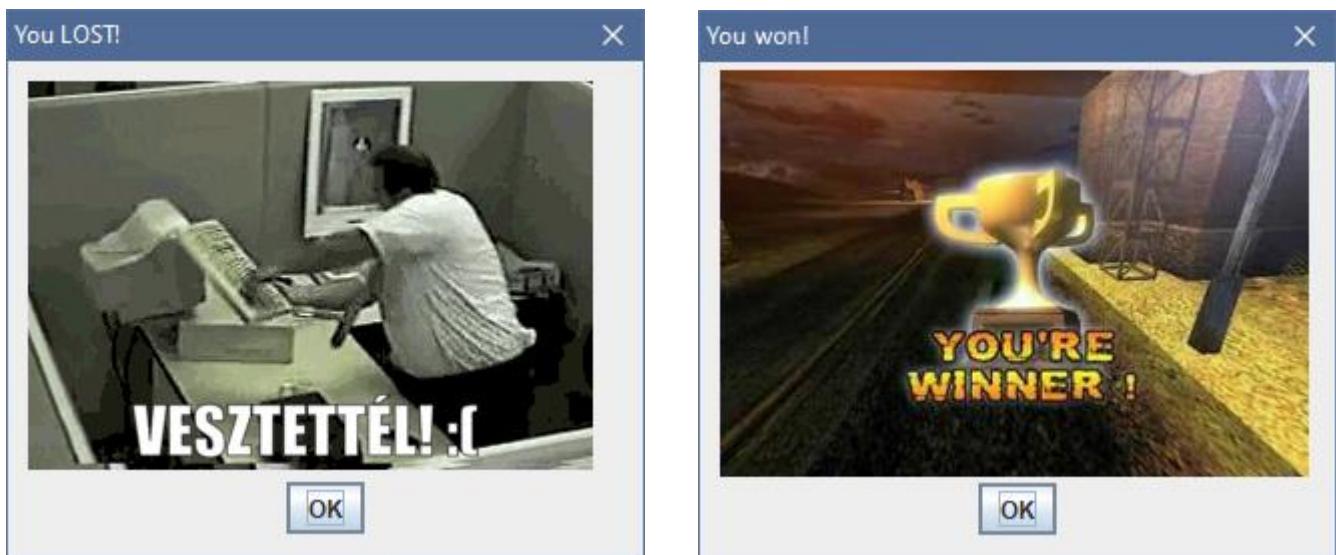
Az irányítás alapvetően egérkattintásokkal fog történni, mivel minden jégtábla rendelkezik szélességgel, magassággal és koordinátákkal, ezért egyszerűen meg lehet mondani, hogy a felhasználó a kattintásával melyiket választotta ki. Ilyenkor két esetet különböztetünk meg, jobb vagy bal egérkattintás szerint. Ha bal egérgombbal kattintott egy jégtáblára, akkor azt jelenti, hogy oda szeretne mozogni (move), ha jobb kattintás történt, akkor pedig azon a jégtáblán képességet szeretne használni (use ability). A többi akcióhoz nem szükséges jégtáblát választani, mivel azok a műveleteket minden vagy egy adott jégtáblára vonatkoznak, vagy független attól a jégtáblától, amin a karakter tartózkodik. Ilyenek a Dig, Assemble, Build Tent, Next Character parancsok, melyeket a felhasználó egy-egy gombra kattintás segítségével adhat ki.

A játék közben megjelenő felület terve az előzőek alapján a következő:



A jégtáblák egy gráf csúcsaiban helyezkednek el. minden táblán megjelenik a rajta tartózkodó karakterek (sarkutatók és eszkimók), jegesmedvék, épületek (iglu vagy sátor), és éppen kiásás alatt lévő tárgy képe. Az aktuális karaktert a játék kijelöli, hogy könnyebb legyen azonosítani. Ennek a karakternek jobb oldalon megjelenik az inventory-ja, vagyis azon tárgyak halmaza, amik nála vannak. Megjelenik továbbá a karakter testhője, maradék munkavégző képessége (stamina), valamint egy vihar jel, ha hóvihar várható a következő körben. A jégtáblák színei eltérőek. A fehér hó borítja, a sötétkéket nem, a világoskék pedig azt jelöli, hogy instabil, valamint erről a tényről egy sarkutató megbizonyosodott. A sarkutató vizsgálata után a jégtábla teherbírása is megjelenik a jégtáblán. Jégtáblák között csak akkor lehet lépni, ha a két tábla között van vonal.

A játék vége képernyő tervei:



## 11.2 A grafikus rendszer architektúrája

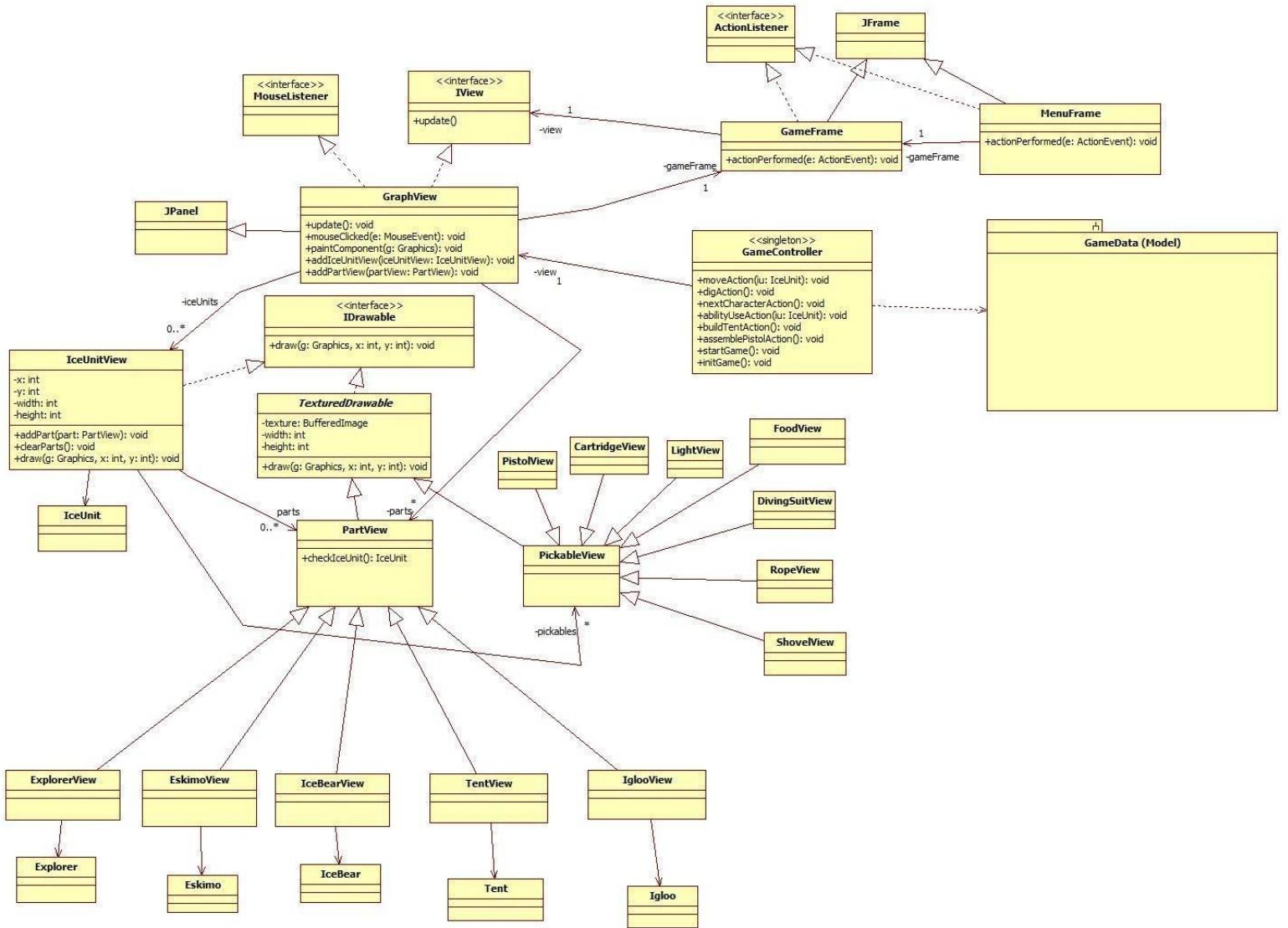
### 11.2.1 A felület működési elve

A grafikus felületet az **MVC architektúra** elveinek megfelelően próbáltuk kialakítani. Ennek hatalmas előnye, hogy a helyes tervezés esetén a játék modelljének működését nem kell változtatni. Ez azért nagyon előnyös, hiszen ezt a részt már korábban teszteltük, így továbbra is bízhatunk ezen nagyon meghatározó rész működésének helyességében. Az MVC architektúra kontroller részének helyes megvalósításáért a GameController osztály felelős. Ez a kontrollért, az irányításért felelős osztály példánya felületet kínál a nézeteknek, hogy a segítségével a megfelelő jelzések elérhessék a modellt; ezzel egyfajta összeköttetést biztosítva a felhasználó által kiváltott események és a modell viselkedések között. A játék megjelenítésére a GameFrame osztályú ablakot használjuk. Ez az ablak tartalmazza a játékpályát felrajzoló GraphView osztály egy példányát, illetve további, az irányításhoz szükséges komponenseket. A játékpálya megjelenítésének helyes működésének érdekében létrehoztunk több, a modell egyes elemeit modellező osztályt. Ezekben az osztályokban közös, hogy megvalósítják az IDrawable interfészét. Amennyiben a megjelenítés érzékeny az általa megjelenített modellelem állapotára, abban az esetben, hogy garantáljuk a helyes megjelenítést, a nézeti osztályok eltárolják az általuk megjelenített modellelem referenciáját. A nézetünk megalkotásában a **push alapelv** szerinti tervezést követtük, amennyiben valamilyen változás történhetett a modellben, abban az esetben értesítük a nézetet, hogy ez alapján aktualizálnia kell magát.

A pályafájlokban annyi változtatás történik, hogy a jégtáblákhoz eltároljuk azt is, hogy a grafikus felületen milyen pozícióban (x, y koordináta) jelennek meg, így a pályák mentésénél és betöltésénél ez a plusz információ meg kell jelenjen. Ez és az eddig eltárolt információk alapján a pályákat már konzisztensen vissza lehet tölteni a grafikus változatban is.

A pályák betöltését és mentését a grafikus felületen a menüsor segítségével lehet elvégezni. A "Load Game" menüpontot kiválasztva megjelenik egy fájlválasztó ablak, ahol megadhatjuk, melyik pályafájlt akarjuk betölteni. A "Save" menüpont is hasonlóan egy fájl dialógus nyílik meg, viszont itt azt adhatjuk meg, hogy a jelenlegi állást milyen nevű fájlbba mentük el. A pálya betöltés csak a főmenüből, a mentés pedig csak játék közben érhető el.

## **11.2.2 A felület osztály-struktúrája**



## 11.3 A grafikus objektumok felsorolása

### 11.3.1 GameController

- **Felelősség**

Ez az osztály az MVC architektúrában a Controller, azaz ez teremti meg a nézet és a játéklogika között a kapcsolatot. Feladata, hogy a felhasználói események hatását közvetítse a modellnek és a nézetnek pedig jelezze, hogy frissítenie kell magát ha változott a modell. Továbbá a játék indításáért, irányításáért, befejezéséért és az alapvető működéséért felel ez az osztály. Az ő feladata jelezni a katasztrófáknak, hogy olyan időpillanatba jutott a játék, hogy bekövetkezésük szabályos lenne, illetve felel azért is, hogy a sátrak eltűnjenek a kör végén.

- **Ősosztályok**
- 

- **Interfészek**
- 

- **Attribútumok**

- -**StormyField[0..\*] disastrousFields:** Eltárolja azokat a pályákat, amiket érhet vihar.
- -**Commandable [1..\*] commandables:** Az irányítható dolgok összesége.
- - **Character [1..\*] character:** A karakter objektumok összessége, a felhasználó ezeket tudja irányítani.
- - **Character activeCharacter:** Ez az a karakter akit éppen irányíthatnak, tehát a felhasználótól jövő bemenetek, parancsok amik egy karakterre vonatkoznak, az mindenig az éppen aktív karaktert fogja cselekvésre késztetni.

- **Metódusok**

- **+void startGame():** A metódus feladata a játék elindítása, vezénylése.
- **+void initGame():** Ezen metódus feladata a játék kezdeti inicializációs lépéseinél elvégzése, azaz létrehozza a megfelelő objektumokat (modellt és nézetet, továbbá egy ablakot a megjelenítés számára), a pályát, és a grafikus elemekhez rendeli a hozzá tartozó modell elemeket. Feltölti a nézet tárolót a grafikus objektumokkal.
- **+void moveAction(iu: IceUnit):** Amikor a felhasználói felületen egy olyan esemény következett be, aminek hatására egy karakternek mozognia kell egy kijelölt IceUnitra, akkor ennek az eseménynek, parancsnak a továbbítása a megfelelő Character felé a metódus feladata. Ez a függvény meghívja az aktív karakter move metódusát a kijelölt IceUnittal paraméterezve.
- **+void digAction():** A felhasználói felületen egy olyan esemény következett be, aminek hatására egy karakternek ásnia kell, ennek az eseménynek, parancsnak a továbbítása a megfelelő Character felé a metódus feladata. Ez a függvény meghívja az aktív karakter dig.
- **+void nextCharacterAction():** A felhasználói felületen egy olyan esemény következett be (egy gomb lenyomásának vagy akár bármilyen más tetszőleges input hatására), aminek következményeképpen egy karakter befejezi a körét, akkor ezt ez a függvény vezényli le. Ez a függvény megváltoztatja az éppen aktív karaktert, közben akár lehetőséget adhat más Commandable objektumoknak a cselekvésre.
- **+void abilityUseAction(iu: IceUnit):** A felhasználói felületen egy olyan esemény következett be, aminek hatására egy karakternek képességet kell használnia egy kijelölt IceUniton, ennek az eseménynek, parancsnak a továbbítása a megfelelő Character felé a metódus feladata. Ez a függvény meghívja az aktív karakter useAbility metódusát a kijelölt IceUnittal paraméterezve.

- **+void buildTentAction():** A felhasználói felületen egy olyan esemény következett be, aminek hatására egy karakternek építenie kell egy sárat, ennek az eseménynek, parancsnak a továbbítása a megfelelő Character felé a metódus feladata. Ez a függvény meghívja az aktív karakter buildTent metódusát.
- **+void assemblePistolAction():** felhasználói felületen egy olyan esemény következett be, aminek hatására egy karakternek meg kell próbálnia összeszerelni egy pisztolyt, ennek az eseménynek, parancsnak a továbbítása a megfelelő Character felé a metódus feladata. Ez a függvény meghívja az aktív karakter assemble metódusát.

### 11.3.2 MenuFrame

- **Felelősség**

A játék indulásakor ez jelenik meg először, ez egy önálló ablak, ami gombokat tartalmaz és az egyik ilyen gombra kattintva lehet elindítani a játékot.

- **Ősosztályok**

JFrame

- **Interfészek**

ActionListener

- **Attribútumok**

• **-GameFrame gameFrame:** Tartalmazza a játék megjelenítésért felelős objektumot, ha a felhasználó a megfelelő gombra kattintva elindítja a játékot, akkor ezt az ablakot jeleníti meg.

- **Metódusok**

• **+void actionPerformed(ActionEvent e):** A különböző gombokra kattintás eseményeit kezeli.

### 11.3.3 GameFrame

- **Felelősség**

A játék ebben az ablakban jelenik meg, egy JPanel objektumon, tehát a felelőssége, hogy az ablak a megfelelő tartalmat jelenítse meg és ezen az ablakon a panelek és gombok megfelelően helyezkedjenek el.

- **Ősosztályok**

JFrame

- **Interfészek**

ActionListener

- **Attribútumok**

• **-IView view:** Az ablakhoz tartozó nézet.

- **Metódusok**

• **+void actionPerformed(ActionEvent e):** A különböző gombokra kattintás eseményeit kezeli.

### 11.3.4 IView

- **Felelősség**

A különböző nézetek interfésze, előírja a megjeleníthető nézetek számára az update metódus megvalósítását.

- **Interfészek**

-

- **Metódusok**

- **+void update(): void:** A nézetet frissíteni kell a modell adatai alapján.

### 11.3.5 GraphView

- **Felelősség**

A játék ezen a panelen rajzolódik ki, tehát az osztály feladata ennek a megvalósítása, ez egy fajta nézet, ami a játékokat gráf formában jeleníti meg.

- **Ősosztályok**

JPanel

- **Interfészek**

IView, MouseListener

- **Attribútumok**

- **-IceUnitView[0..\*] iceUnits:** Tárolja az IceUnitokat, amiket ki lehet rajzolni, ezek helyezkednek el a legmagasabb szinten, ezeknek a koordinátája a modell alapján meghatározható.

- **-PartView parts[0..\*]:** Tárolja azokat a kirajzolható objektumokat, amik majd egy IceUnitView-n lesznek rajta a grafikus megjelenítés során.

- **-GameFrame gameFrame:** Tartalmazza azt, hogy a nézethez melyik ablak tartozik.

- **Metódusok**

- **+void mouseClicked(MouseEvent e):** Az egérkattintás eseményének kezelése. Ez képes meghatározni, hogy melyik koordinátában történt a kattintás, és hogy abban a koordinátában mi helyezkedik el.

- **+void update(): void:** A nézetet frissíteni kell. A játékmagyarázatban valamelyen változás történt, ezért a GraphView nézetnek frissítenie, azaz újra rajzolnia kell magát az változott adatoknak megfelelően. Meghívja az összes IceUnitView clearParts metódusát. Majd meghívja a tartalmazott PartView objektumoknak a checkIceUnit metódusát, majd elvégzi a megfelelő IceUnitView-hoz való rendelését (addPart függvény hívása). Ezután pedig meghívja az összes IceUnitView-jának a draw metódusát.

- **+void paintComponent(Graphics g):** Ez a metódus vége a rajzolást, a paraméterként kapott Graphics objektumra, ennek során ki kell rajzolni minden olyan objektumot a játékban, amit meg szeretnénk jeleníteni. Meghívja a kirajzolható IceUnitView objektumoknak a draw függvényét a Graphics g objektumával paraméterezve.

- **+void addIceUnitView(iceUnitView: IceUnitView):** Hozzáad egy IceUnitView objektumot a megfelelő tárolóhoz.

- **+void addPartView(partView: PartView):** Hozzáad egy PartView objektumot a megfelelő tárolóhoz.
- 

### **11.3.6      IceUnit View**

- **Felelősség**

Egy a modellben is szereplő IceUnit kirajzolása a feladata, annak teljes tartalmával együtt. Az IceUniton vizuálisan is megjelenhetnek más objektumok.

- **Ősosztályok**
- 

- **Interfészek**

IDrawable

- **Attribútumok**

- **-PartView[0..\*] parts:** Az IceUniton megjelenő objektumok.
- **-int x:** Az IceUnitView objektum elhelyezkedésének x koordinátája.
- **-int y:** Az IceUnitView objektum elhelyezkedésének y koordinátája.
- **-int width:** Az IceUnitView szélessége.
- **-int height:** Az IceUnitView magassága.
- **IceUnit iceUnitModell:** A nézethez tartozó modell, aminek az adatai alapján jelenik meg a nézet.

- **Metódusok**

- **+void draw(g: Graphics, x: int, y: int):** A paraméterként kapott Graphics objektumra kirajzolja magát, és a tartalmazott IDrawable interfész megvalósító objektumoknak is meghívja a draw függvényeiket.

### **11.3.7      IDrawable**

- **Felelősség**

A kirajzolható objektumok interfésze.

- **Interfészek**
- 

- **Metódusok**

- **+void draw(g: Graphics, x: int, y: int):** Előírja a rajzolás megvalósítását az implementáló osztályok számára.

### **11.3.8      TexturedDrawable**

- **Felelősség**

Egy olyan absztrakt osztály, ami kirajzolható és rendelkezik valamilyen textúrával, amit egy képfájlból lehet beolvasni.

- **Ősosztályok**  
-

- **Interfészek**

IDrawable

- **Attribútumok**

- **-BufferedImage texture:** Az objektum textúrája.
- **-int width:** Az IceUnitView szélessége.
- **-int height:** Az IceUnitView magassága.

- **Metódusok**

- **+void draw(g: Graphics, x: int, y: int):** A paraméterként kapott Graphics objektumra kirajzolja magát, a textúrája, a modell adatai és a kapott koordináták alapján.

### 11.3.9 PartView

- **Felelősség**

Egy olyan kirajzolható objektum, amit kirajzolódik bizonyos koordinátákra.

- **Ősosztályok**

TexturedDrawable

- **Interfészek**  
-

- **Attribútumok**  
-

- **Metódusok**

- **+IceUnit checkIceUnit():** Visszatér azzal, hogy melyik IceUnit modell objektum tartozik a nézethez.

### 11.3.10 PickableView

- **Felelősség**

Olyan elemeket reprezentál ez az osztály, amik kirajzoláskor egy IceUnitView-n jelennek meg.

- **Ősosztályok**

TexturedDrawable

- **Interfészek**  
-

- **Attribútumok**  
-

- **Metódusok**

-

### 11.3.11 ExplorerView

- **Felelősség**

Egy a modellben is szereplő Explorer objektum kirajzolása a feladata, annak adatai alapján.

- **Ősosztályok**

TextureDrawable -> PartView

- **Interfészek**

-

- **Attribútumok**

• **-Explorer explorer:** A nézethez tartozó modell, aminek az adatai alapján jelenik meg, rajzolódik ki a nézet.

- **Metódusok**

-

### 11.3.12 EskimoView

- **Felelősség**

Egy a modellben is szereplő Eskimo objektum kirajzolása a feladata, annak adatai alapján.

- **Ősosztályok**

TextureDrawable -> PartView

- **Interfész**

-

- **Attribútumok**

• **-Eskimo eskimo:** A nézethez tartozó modell, aminek az adatai alapján jelenik meg, rajzolódik ki a nézet.

- **Metódusok**

-

### 11.3.13 IceBearView

- **Felelősség**

Egy a modellben is szereplő IceBear objektum kirajzolása a feladata, annak adatai alapján.

- **Ősosztályok**

TextureDrawable -> PartView

- **Interfészek**
  - **Attribútumok**
  - **-Icebear iceBear:** A nézethez tartozó modell, aminek az adatai alapján jelenik meg, rajzolódik ki a nézet.
- **Metódusok**
  -

### 11.3.14 TentView

- **Felelősség**

Egy a modellben is szereplő Tent objektum kirajzolása a feladata, annak adatai alapján.

- **Ősosztályok**

TextureDrawable -> PartView

- **Interfészek**
  -
- **Attribútumok**
- **- Tent tent:** A nézethez tartozó modell, aminek az adatai alapján jelenik meg, rajzolódik ki a nézet.
- **Metódusok**
  -

### 11.3.15 IglooView

- **Felelősség**

Egy a modellben is szereplő Igloo objektum kirajzolása a feladata, annak adatai alapján.

- **Ősosztályok**

TextureDrawable -> PartView

- **Interfészek**
  -
- **Attribútumok**
- **- Igloo igloo:** A nézethez tartozó modell, aminek az adatai alapján jelenik meg, rajzolódik ki a nézet.
- **Metódusok**
  -

### 11.3.16     **PistolView**

- **Felelősség**

Egy a modellben is szereplő Pistol objektum kirajzolása a feladata.

- **Ősosztályok**

TextureDrawable → PickableView

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

-

### 11.3.17     **CartridgeView**

- **Felelősség**

Egy a modellben is szereplő Cartridge objektum kirajzolása a feladata.

- **Ősosztályok**

TextureDrawable → PickableView

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

-

### 11.3.18     **LightView**

- **Felelősség**

Egy a modellben is szereplő Light objektum kirajzolása a feladata.

- **Ősosztályok**

TextureDrawable → PickableView

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

-

### 11.3.19 FoodView

- **Felelősség**

Egy a modellben is szereplő Food objektum kirajzolása a feladata.

- **Ősosztályok**

TextureDrawable → PickableView

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

-

### 11.3.20 DivingSuitView

- **Felelősség**

Egy a modellben is szereplő DivingSuit objektum kirajzolása a feladata.

- **Ősosztályok**

TextureDrawable → PickableView

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

-

### 11.3.21 RopeView

- **Felelősség**

Egy a modellben is szereplő Rope objektum kirajzolása a feladata.

- **Ősosztályok**

TextureDrawable → PickableView

- **Interfészek**

-

- **Attribútumok**
- 

- **Metódusok**
- 

### 11.3.22 ShovelView

- **Felelősség**

Egy a modellben is szereplő Shovel objektum kirajzolása a feladata.

- **Ősosztályok**

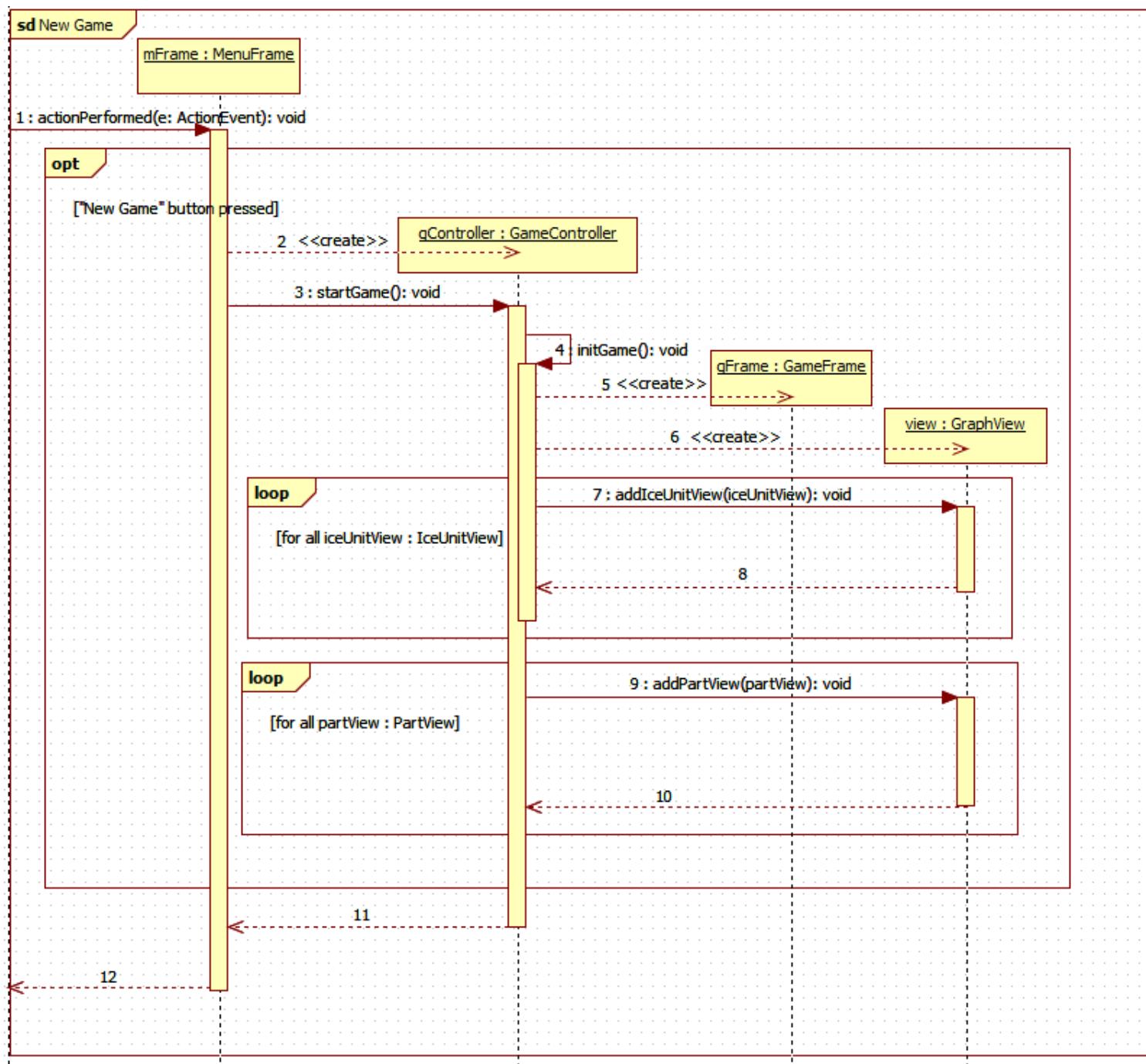
TextureDrawable -> PickableView

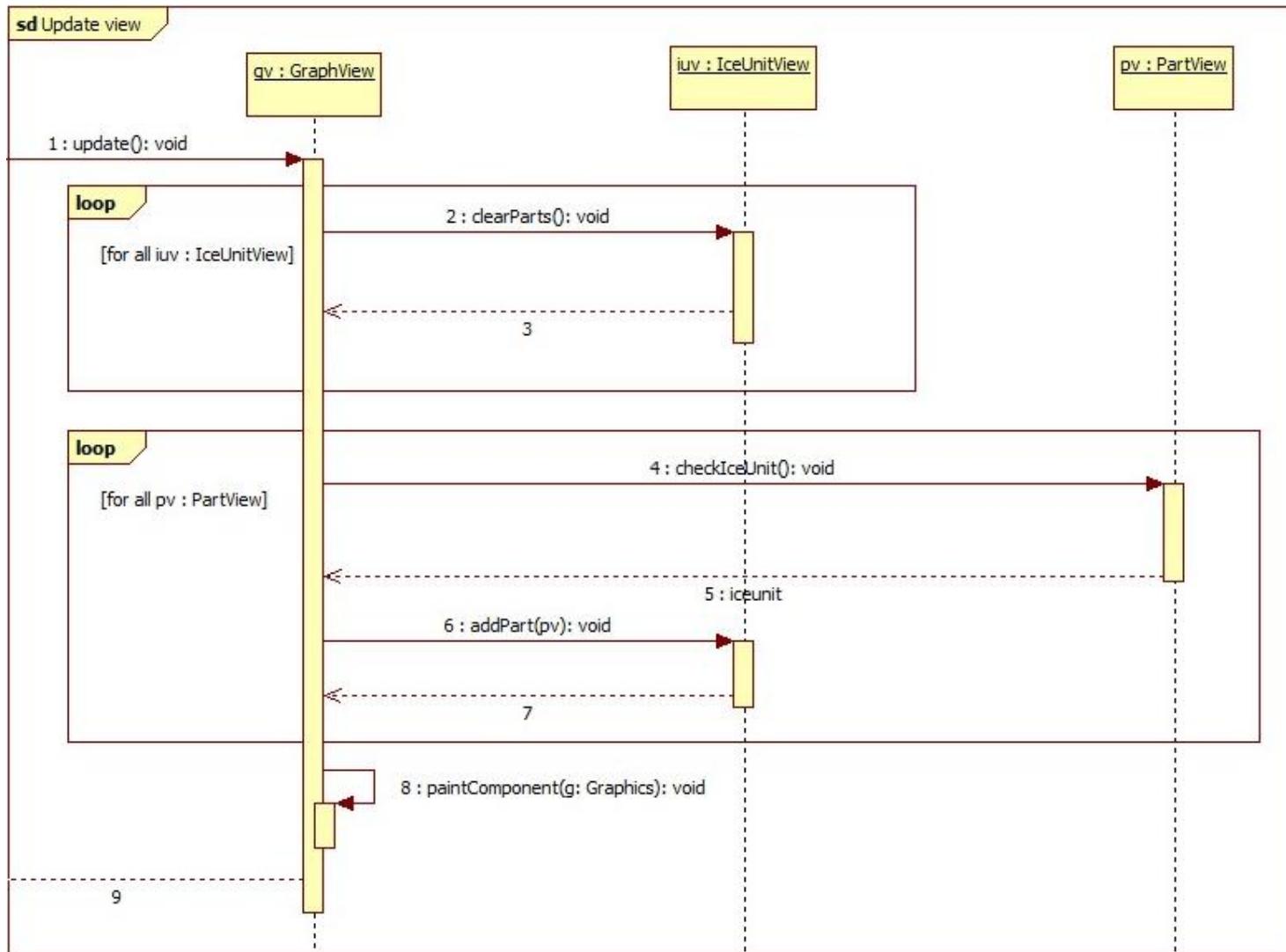
- **Interfészek**
- 

- **Attribútumok**
- 

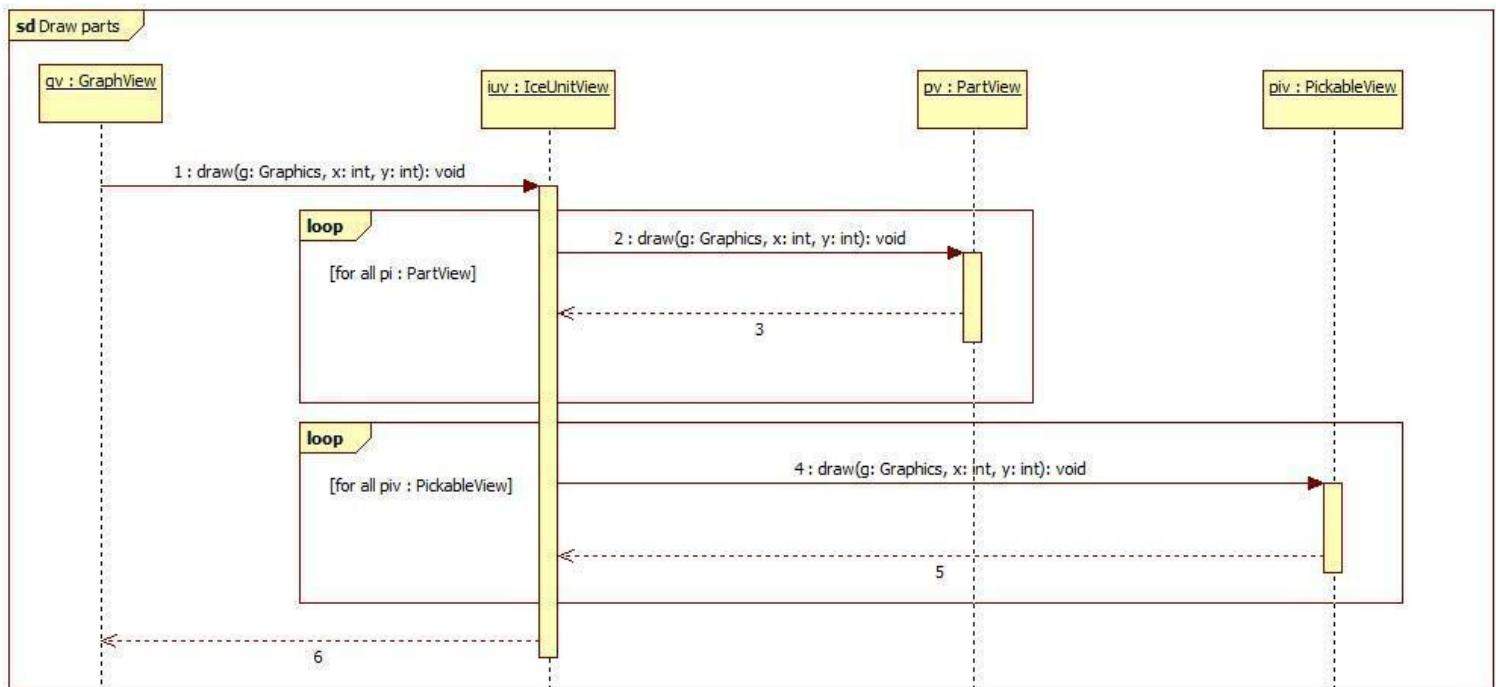
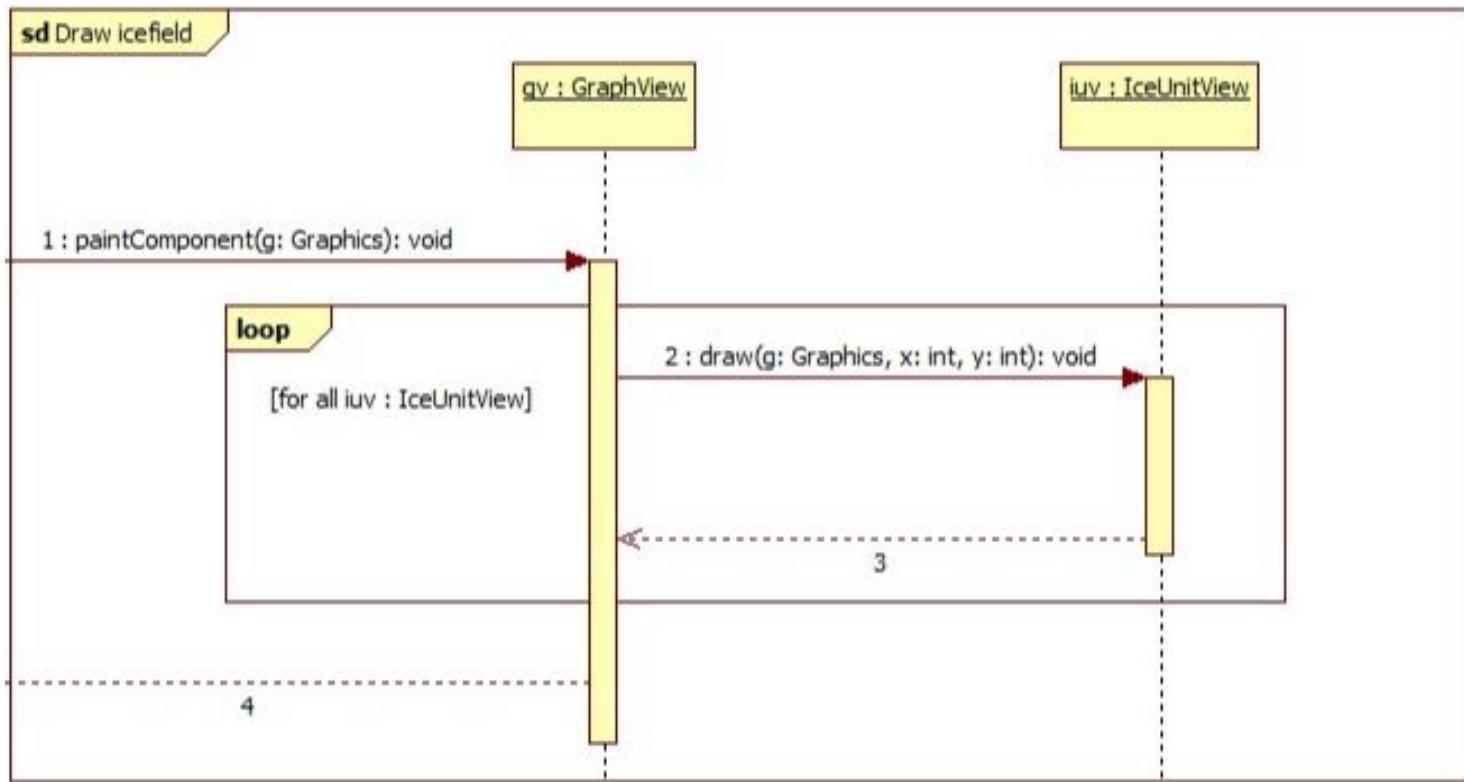
- **Metódusok**
-

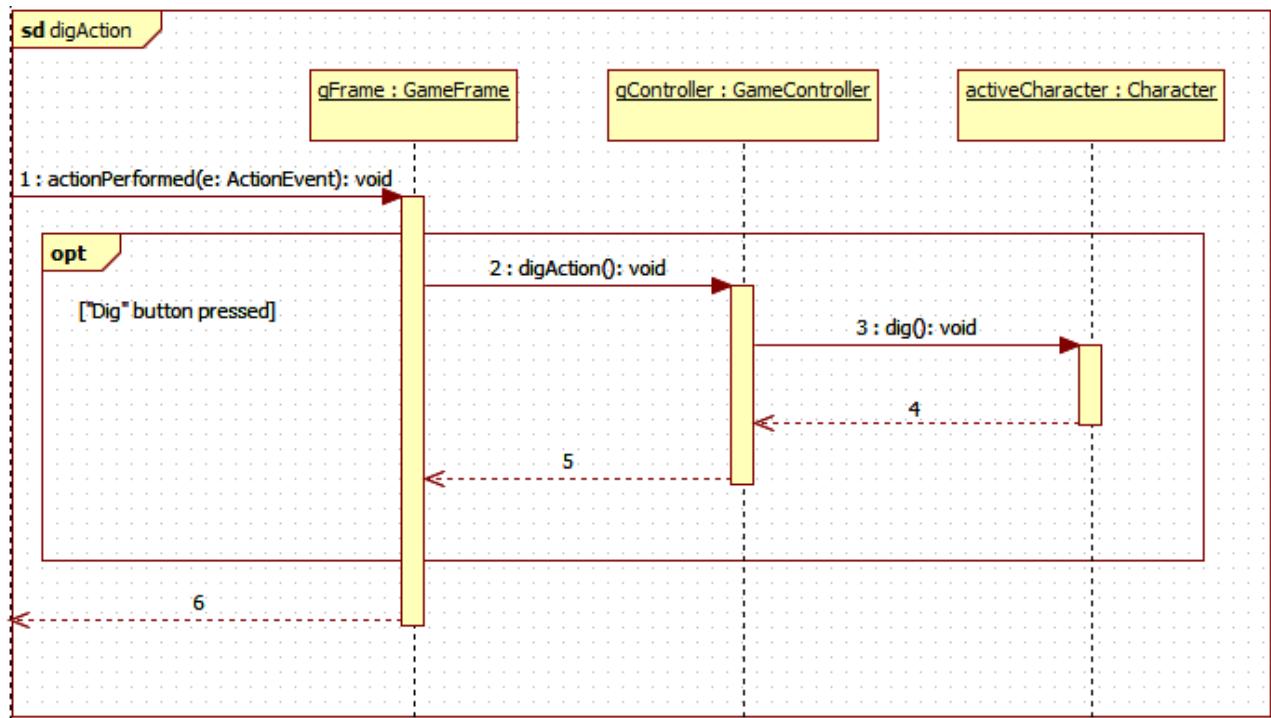
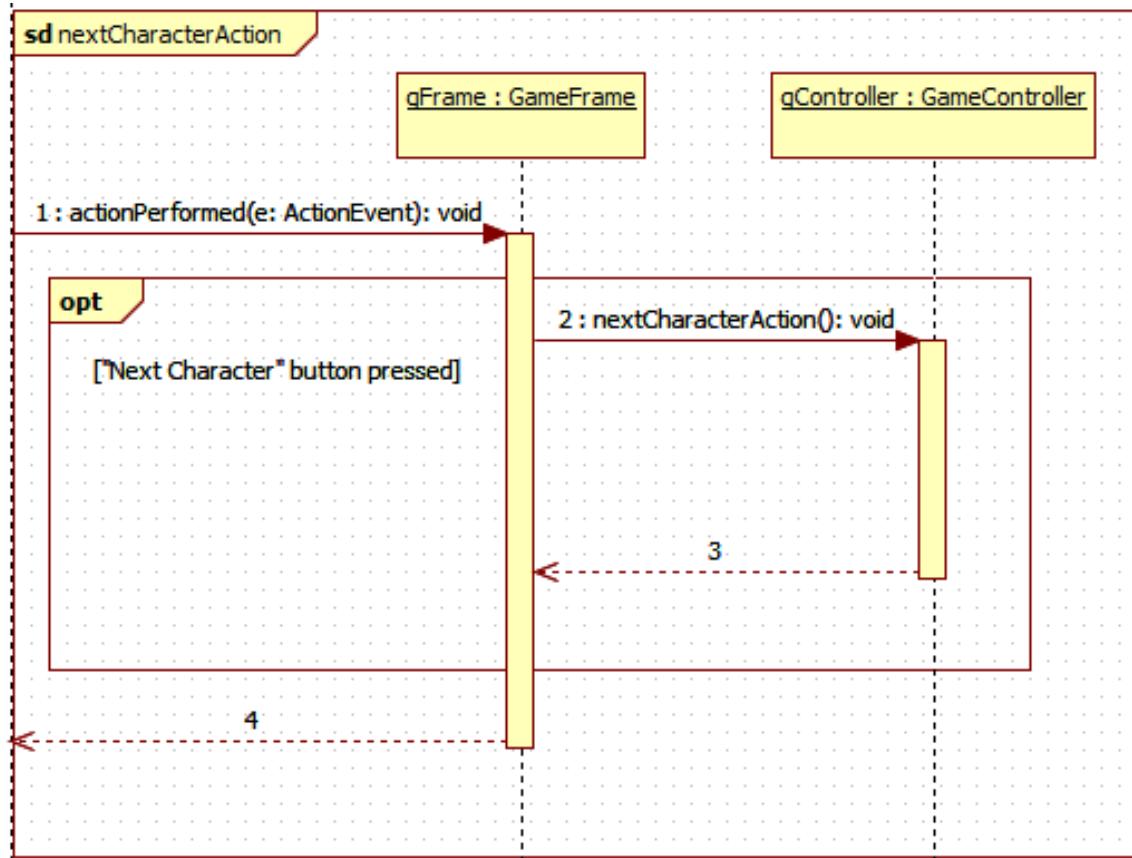
## 11.4 Kapcsolat az alkalmazói rendszerrel

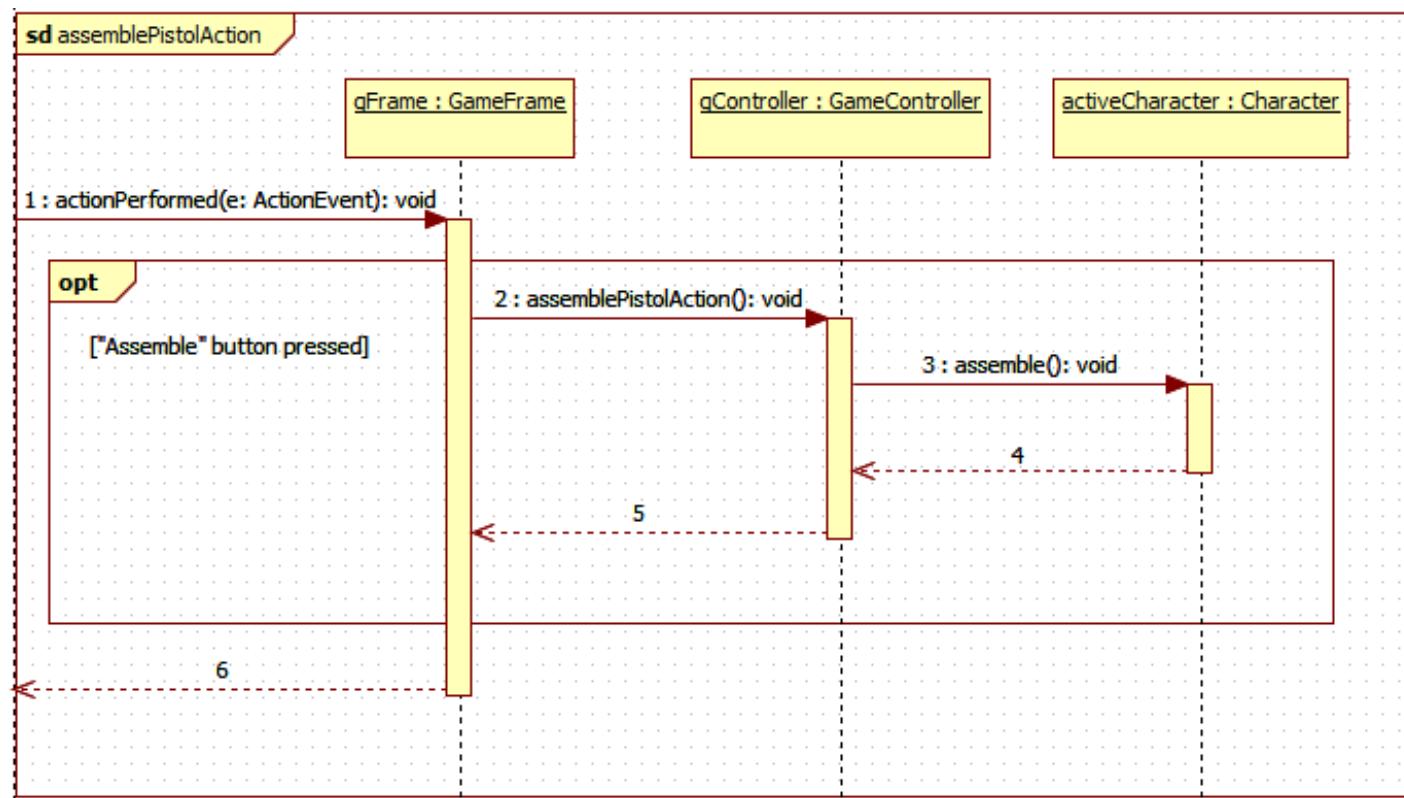
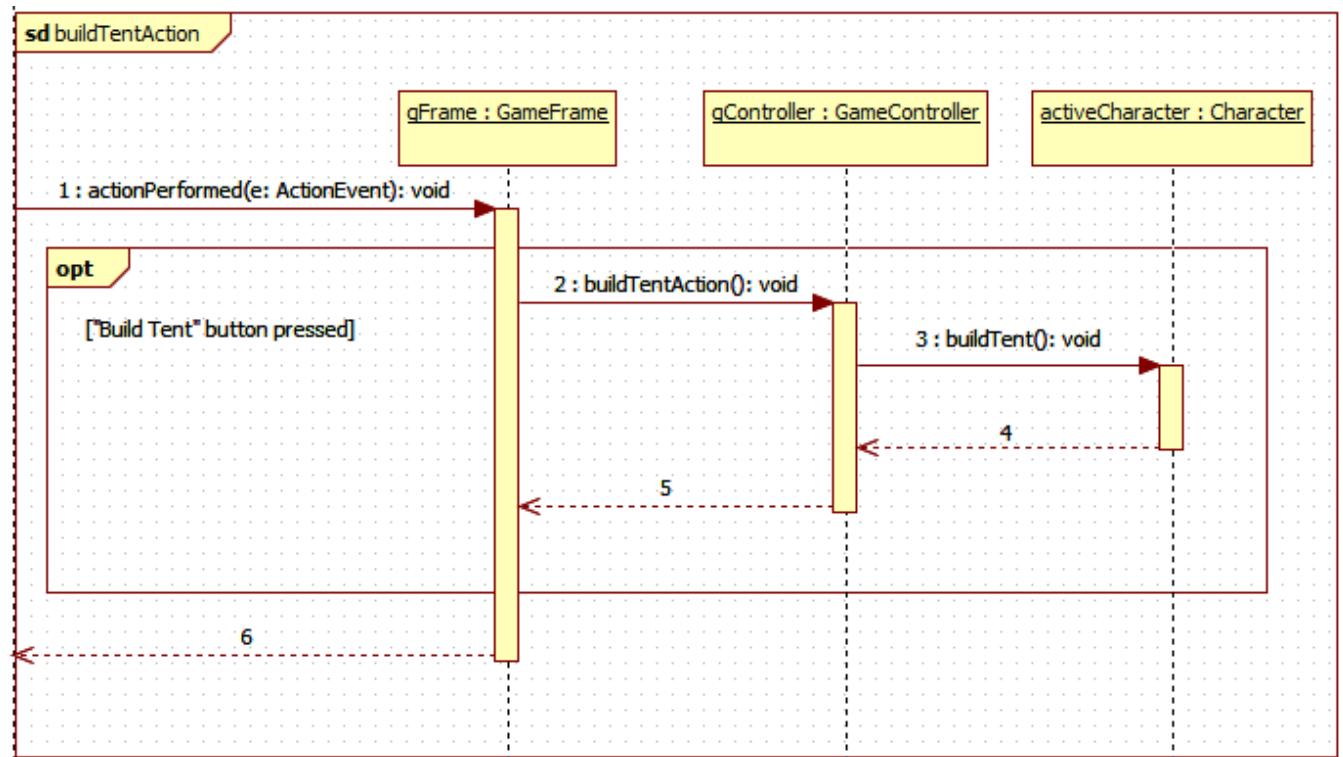


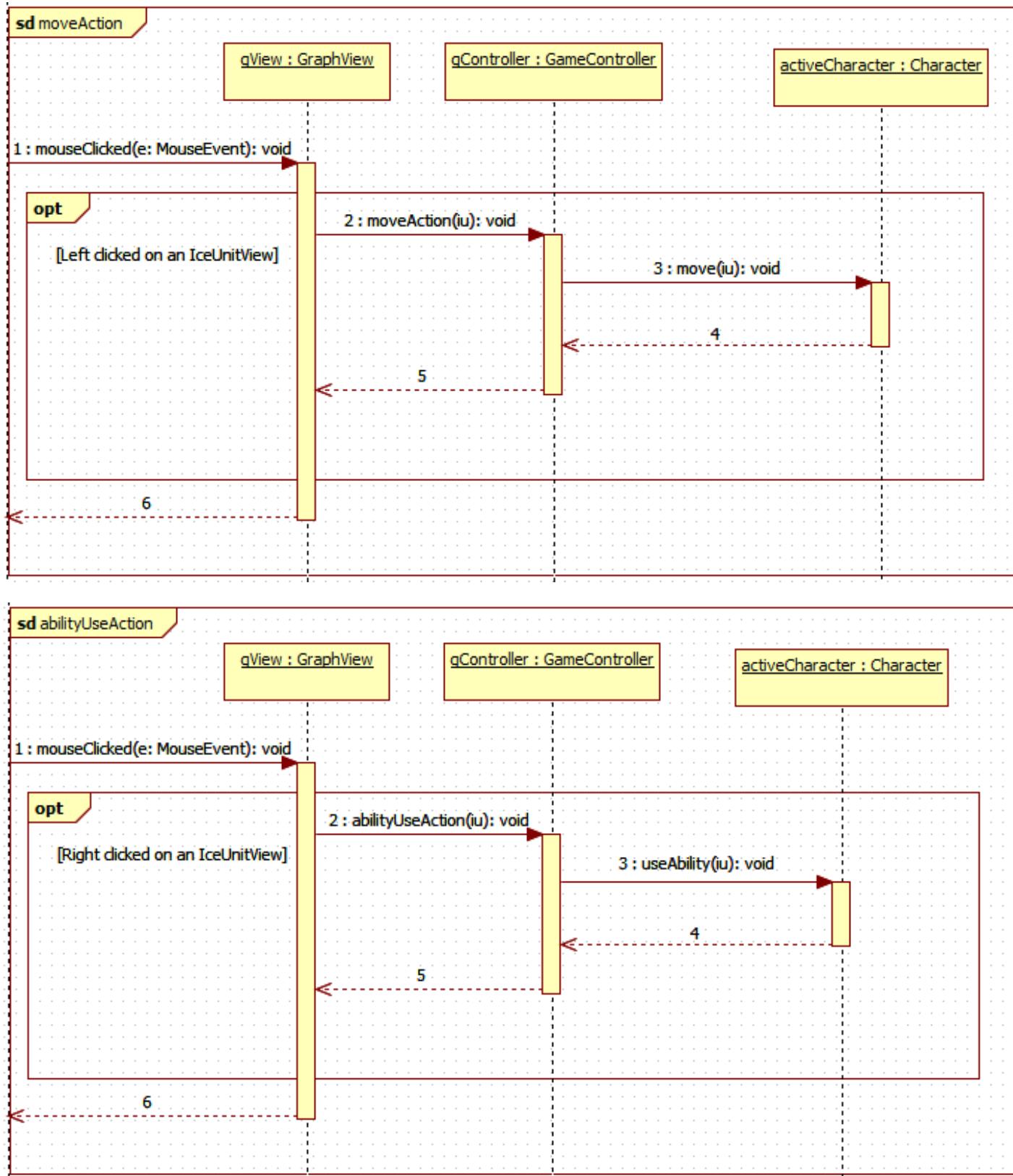


Megjegyzés: a `paintComponent` metódus meghívása nem közetlenü fog történni, hanem mivel ez a `JPanel`-től örökölt függvény felüldefiniálása, így ezt egy `repaint` függvényhívással tudjuk elérni









**Napló**

<b>Kezdet</b>	<b>Időtartam</b>	<b>Résznevők</b>	<b>Leírás</b>
2020.05.02 10:00	3 óra	Kaposvári Labancz Iuhos Sulyok Traxler	Értekezlet: Feladat értelmezése, részfeladatok kiosztása
2020.05.02 14:00	4 óra	Sulyok	Tevékenység: Grafikus felület programozása
2020.05.02 14:00	3 óra	Traxler	Tevékenység: Oszálydiagramok rajzolása
2020.05.02 18:00	3 óra	Sulyok	Tevékenység: Osztálydiagramok rajzolása
2020.05.03 8:00	3 óra	Kaposvári Labancz Iuhos Traxler	Értekezlet: Eddigi munka ellenőrzése, további feladatok kiosztása
2020.05.03 11:00	4 óra	Kaposvári	Tevékenység: Szekvencia diagramok rajzolása
2020.05.03 12:00	4 óra	Labancz	Tevékenység: Szekvencia diagramok rajzolása, dokumentáció írása
2020.05.03 15:00	4 óra	Traxler	Tevékenység: Osztálydiagramok javítása, dokumentáció írása
2020.05.04 9:00	62 óra 12 perc	Iuhos	Tevékenység: Napló megírása

# **13. Feladat: Grafikus változat beadása**

**14 – the\_minesweepers**

Konzulens:  
**Fetter László**

## **Csapattagok**

Iuhos Zoltán  
Kaposvári Márk Martin  
Sulyok Botond  
Labancz Tamás Dominik  
Traxler Bálint

J8LOUL	zoltan.iuhos@gmail.com
RAYVF8	xmarko45@gmail.com
BM2BBM	botond.sulyok99@gmail.com
A9HZBF	labancz.tamas@gmail.com
IXNOYB	traxlerbalint@gmail.com

2020.05.18.

## 13. Grafikus változat beadása

### 13.1 Fordítási és futtatási útmutató

#### 13.1.1 Fájllista

Fájl neve	Méret (B)	Keletkezés ideje	Tartalom
GameFrame.java	5 002	2020.05.14 14:12	Az ablakot kezelő osztályt tartalmazza.
GraphView.java	13 118	2020.05.14 15:18	A játék ezen a panelen rajzolódik ki, tehát az osztály feladata ennek a megvalósítása.
IView.java	292	2020.05.14 14:57	A különböző nézetek interfésze.
MenuFrame.java	6 202	2020.05.14 15:46	A játék indulása előtti képernyőt kezelő osztály.
TextureType.java	1 214	2020.05.14 14:32	A textúratípusokat definiáló felsorolást tartalmazó fájl.
BuildingView.java	3 131	2020.05.14 16:02	Az épületek megjelenítését kezelő osztály.
CharacterView.java	2 715	2020.05.14 16:24	A karakterek megjelenítését kezelő osztály.
IceBearView.java	1 886	2020.05.14 16:48	A jegesmedve megjelenítését kezelő osztály.
IceUnitView.java	5 640	2020.05.14 17:21	A jégtáblák megjelenítését kezelő osztály.
IDrawable.java	553	2020.05.14 16:12	A kirajzolható objektumok interfészét tartalmazó fájl.
PartView.java	574	2020.05.14 18:11	Koordinátára rajzolható objektumokat kezelő osztály.
PickableView.java	3 710	2020.05.14 18:33	Felvehető tárgyak kirajzolását kezelő osztály.
TexturedDrawable.java	1 832	2020.05.14 18:49	Kirajzolható, textúrával rendelkező objektumokat kezelő osztályt tartalmaz.
backgroundwater.jpg	707 337	2020.04.29 16:11	Ez a kép a játékban a háttér.
breakable.jpg	3 328	2020.04.29 16:12	Ez a kép a játékban a törékeny ásó.
cartridge.png	894	2020.04.29 16:12	Ez a kép a játékban a töltény.
defaulttexture.png	360	2020.05.13. 17:06	Ez a kép a játékban arra való, hogy hiba esetén ezt töltődjön be és ne kivételel dobjon a program.
divingsuit.jpg	8 958	2020.04.29 16:13	Ez a kép a játékban a búvárruha.
eskimo.jpg	73 084	2020.04.29 16:14	Ez a kép a játékban az eszkimo.

explorer.jpg	11 556	2020.04.29 16:16	Ez a kép a játékban a sarkkutató.
food.jpg	7 241	2020.04.29 16:17	Ez a kép a játékban az étel.
howtoplay.txt	2 869	2020.05.15 15:32	Egy rövid leírást tartalmaz a játék használatáról.
icebear.jpg	12 140	2020.04.29 16:18	Ez a kép a játékban a jegesmedve.
igloo.png	1 932	2020.04.29 16:18	Ez a kép a játékban az iglu.
info.png	2 466	2020.05.13. 17:18	Ez a kép a játékban a karakter információi felett jelenik meg.
inventory.png	2 428	2020.05.13. 17:22	Ez a kép a játékban a karakter tárgyai felett jelenik meg.
light.png	5 949	2020.04.29 16:19	Ez a kép a játékban a jelzőfény.
looser1.gif	1 288 674	2020.04.29 16:20	Ez a gif a játékban akkor jelenik meg, ha a csapat vesztett.
mainimage.png	113 802	2020.05.13. 17:39	Ez a kép a játékban a főképernyőn jelenik meg.
main_map.map	5 906	2020.05.16. 14:02	Ez az alap térképet tartalmazó fájl.
pickabletent.png	3 594	2020.05.13. 17:49	Ez a kép a játékban a befagyott sátor.
pistol.png	4 042	2020.04.29 16:21	Ez a kép a játékban a jelzőpisztoly.
plains.png	1 617	2020.04.29 16:22	Ez a kép a játékban a hó.
rope.png	3 342	2020.04.29 16:22	Ez a kép a játékban a kötél.
shovel.jpg	10 453	2020.04.29 16:24	Ez a kép a játékban az ásó.
snow.png	6 031	2020.05.13. 17:57	Ez a kép a játékban a hóvihart jelzi.
tent.png	1 861	2020.04.29 16:25	Ez a kép a játékban a sátor.
titleimage.png	4 503	2020.05.13. 18:11	Ez a kép a játékban a címnél jelenik meg.
winner1.jpg	54 045	2020.05.13. 18:40	Ez a kép a játékban győzelem esetén jelenik meg.
BuildingType.java	2 214	2020.05.16. 13:32	Az épülettípusok felsorolását tartalmazó fájl.
EntityType.java	1 128	2020.05.16. 14:06	Az entitástípusokat (lénytípusokat) tartalmazó fájl.
ItemType.java	3 781	2020.05.16. 14:28	A tárgytípusok felsorolását tartalmazó fájl.
ParseHelper.java	4 377	2020.05.16. 15:11	A gamecontroller osztály segédosztályát tartalmazó fájl.
CommunicationPlatform.java	1 083	2020.03.25 15:03	Azon interfész, amelyik az egy mezőn keresztül történő kommunikációt lehetővé tévő függvényeket írja elő.

Pullable.java	909	2020.03.25 15:07	Húzható dolgokat kezelő absztrakt osztály.
Puller.java	764	2020.03.25 15:10	Azon interfész, amelyet megvalósító objektumok rendelkeznek az elhúzás képességevel.
Stander.java	2 261	2020.03.25 15:14	Absztrakt osztály, leesni képes dolgok származnak le belőle.
Terrestrial.java	389	2020.03.25 15:21	Azon objektumok interfésze, amelyeknek káros, ha vízben vannak.
Amphibious.java	2 117	2020.03.25 15:24	Osztály, amelyik objektumai mozoghatnak jégtáblán és vízben is.
Collideable.java	887	2020.04.22. 13:04	Ütközni képes objektumok interfésze.
LocationChanger.java	2 439	2020.03.25 15:35	Azon absztrakt osztály, amely képes Pullable objektumokat húzni.
MoveBehaviour.java	993	2020.03.25 15:47	Azon interfész, amely megadja a szükséges függvényeket a különböző mozgásformákhoz.
Walk.java	2 317	2020.03.25 15:51	Ezen osztály objektumai mozoghatnak szárazföldön.
Eskimo.java	2 256	2020.03.25 15:58	Azon osztály, amelyik megvalósítja az eszkimó karaktertípust.
Explorer.java	2 167	2020.03.25 16:05	Azon osztály, amelyik megvalósítja a sarkkutató karaktertípust.
Character.java	13 794	2020.03.25 16:11	Az összes irányítható karakter közös működését megvalósító absztrakt osztály.
ComponentMaster.java	1 354	2020.03.25 16:40	Interfész, az implementáló osztályok objektumai képesek komponenseket használni.
Equipper.java	1 318	2020.03.25 16:44	A tárgyat felvenni képes objektumok interfésze.
FiniteActionDoer.java	507	2020.03.25 16:49	Korlátozott cselekvőképességgel rendelkező objektumok interfésze.
IceBear.java	5 183	2020.04.22 15:51	A jegesmedvét kezelő osztály.
IceUnitManipulator.java	801	2020.03.25 16:52	Interfész, amely biztosítja a jégtáblákon való műveletvégzést.
Mortal.java	2 390	2020.03.25 16:55	Testhőt kezelő absztrakt osztály.

Refugee.java	867	2020.03.25 17:03	A játékban levő menekülteknek az interfésze.
CommandReader.java	7 945	2020.04.23. 14:13	Ez a statikus osztály tartalmazza a játék irányítására használható parancsokat, amiket értelmez és végrehajt.
CommandUtils.java	8 394	2020.04.23. 14:13	Ez az osztály tárolja el a játék objektumait és ezzel az osztályval bármilyen azonosítójú objektumot letudunk kérdezni a jégmezőről.
MapParser.java	6 495	2020.04.23. 14:20	A statikus osztály egyetlen felelőssége egy fájlból betöltött pályának a kezelése és játékba állítása.
ParsedObject.java	2 922	2020.04.23. 14:26	Az osztály felelőssége a Parsable interfész implementáló osztályok objektumaiból egy olyan HashMap-et tárolni, * amelynek adatait könnyen fel lehet dolgozni, és azt mentési vagy vizsgálati célból lehet használni.
AbilityCommand.java	1 225	2020.04.23. 14:31	A statikus osztálynak az egyetlen felelőssége kezelní az "ability" parancsot.
AssembleCommand.java	964	2020.04.23. 14:40	A statikus osztálynak az egyetlen felelőssége kezelní az "assemble" parancsot.
BuildingCommand.java	3 827	2020.04.23. 14:51	A statikus osztálynak a felelőssége kezelní a "building" parancsot, illetve tartalmazza az épülettípusokat.
CurrentCommand.java	564	2020.04.23. 15:02	A CurrentCommand osztály felelőssége megvalósítani, és biztosítani a helyes működését a bemeni nyelv részét képző * current parancsnak.

DigCommand.java	878	2020.04.23. 15:07	A statikus osztálynak az egyetlen felelőssége kezelní a "dig" parancsot.
EntityCommand.java	5 677	2020.04.23. 15:16	A statikus osztálynak a felelőssége kezelní az "entity" parancsot, illetve tartalmazza az entitásípusokat (azok az objektumok, amik "élnek", tudnak mozogni a pályán).
FieldCommand.java	2 724	2020.04.23. 15:26	A statikus osztálynak a felelőssége kezelní a "field" és a "neighbours" parancsot.
ItemCommand.java	5 649	2020.04.23. 15:31	A statikus osztálynak a felelőssége kezelní az "item" parancsot, illetve tartalmazza a tárgytípusokat.
LoadCommand.java	950	2020.04.23. 15:36	A statikus osztálynak az egyetlen felelőssége kezelní a "load" parancsot.
MoveCommand.java	1 168	2020.04.23. 15:45	A statikus osztálynak az egyetlen felelőssége kezelní a "move" parancsot.
SaveCommand.java	963	2020.04.23. 15:51	A statikus osztálynak az egyetlen felelőssége kezelní a "save" parancsot.
StatsCommand.java	3 934	2020.04.23. 15:58	A statikus osztálynak az egyetlen felelőssége kezelní a "stats" parancsot.
StormCommand.java	1 057	2020.04.23. 16:08	A statikus osztálynak az egyetlen felelőssége kezelní a "storm" parancsot.
TentCommand.java	859	2020.04.23. 16:14	A statikus osztálynak az egyetlen felelőssége kezelní a "tent" parancsot.
TurnCommand.java	505	2020.04.23. 16:25	A CurrentCommand osztály felelőssége megvalósítani, és biztosítani a helyes működését a bementi nyelv részét képző turn parancsnak.

Commandable.java	547	2020.03.25 17:07	Irányítási felületet biztosító interfész.
Disaster.java	362	2020.03.25 17:10	A játékban levő katasztrófáknak az interfésze.
GameController.java	22 541	2020.03.25 17:13	A játék indításáért, irányításáért, befejezéséért és az alapvető működéséért felelős osztály.
Identifiable.java	1 133	2020.04.23. 17:12	Az osztály felelőssége reprezentálni az azonosítható objektumokat.
Parseable.java	688	2020.04.23. 17:42	Az interfész felelőssége közös működési felületet biztosítani az olyan implementáló osztályok objektumainak, amelynek adatait fel lehet dolgozni, és azt mentési vagy vizsgálati célból lehet használni.
Storm.java	687	2020.03.25 17:38	A vihart reprezentáló osztály.
Strikeable.java	246	2020.03.25 17:49	A katasztrófák által megcsapható dolgok interfésze.
AlarmPistol.java	419	2020.03.25 17:52	Jelzőpisztolyt kezelő osztály.
Pickable.java	877	2020.03.25 19:26	Felvehető dolgok interfésze.
AlarmPistolComponent.java	2 717	2020.04.22. 15:11	Az osztály felelőssége az, hogy a jelzőfényespisztoly komponenseként belőle összeállítható legyena jelzőfényespisztoly.
Cartridge.java	2 358	2020.03.25 17:57	Jelzőpisztolyt alkotó patront kezelő osztály.
Component.java	1 640	2020.03.25 18:05	Komponenseket kezelő absztrakt osztály.
Light.java	2 350	2020.03.25 18:14	Jelzőpisztolyt alkotó jelzőfényt kezelő osztály.
Pistol.java	2 346	2020.03.25 18:20	Jelzőpisztolyt alkotó pisztolyt kezelő osztály.
Consumable.java	793	2020.03.25 18:23	Az elfogyasztható tárgyak interfésze.
Food.java	2 943	2020.03.25 18:26	Felvehető ételt reprezentáló osztály.
BareHand.java	1 471	2020.03.25 18:34	Az egyszerű ásást kezelő osztály.
BreakableShovel.java	2 571	2020.04.22 15:50	Eltörni képes ásót reprezentáló osztály.

Digger.java	2 365	2020.03.25 18:39	Az ásást kezelő absztrakt osztály.
Shovel.java	2 398	2020.03.25 18:43	Ásót kezelő osztály.
DivingSuit.java	2 711	2020.03.25 18:47	Búvárruhát kezelő osztály.
Lung.java	2 055	2020.03.25 18:56	Olyan osztály, amelyik lehetővé teszi a karakter megfulladását, ha nincs rajta védelem.
WaterProtecter.java	1 056	2020.03.25 19:06	Víz elleni védelmet kezelő interfész.
BasicPull.java	2 058	2020.03.25 19:10	Húzásra képtelenséget kezelő osztály.
PullerDevice.java	1 051	2020.03.25 19:16	A húzást megvalósító eszközök interfésze.
Rope.java	2 764	2020.03.25 19:20	Kötelet kezelő osztály.
Logger.java	2 614	2020.03.28 13:49	A függvényhívások logolásáért felelős osztály.
testAssemble.java	1 641	2020.03.28 14:40	A jelzőpisztoly összeszerelését tesztelő osztály.
testBuildLogIglooAbility.java	902	2020.03.28 15:03	Eszkimó igluépítését tesztelő osztály.
testDigAndEquip.java	1 368	2020.03.28 15:30	Karakter ásását és tárgyfelvételét tesztelő osztály.
testEndTurn.java	803	2020.03.28 16:07	A karakter kör továbbadását tesztelő osztály.
testFallAndPull.java	1 941	2020.03.28 17:11	Esést és kihúzást tesztelő osztály.
testMovement.java	3 415	2020.03.29 14:16	Karakter mozgását tesztelő osztály.
testShowAbility.java	912	2020.03.29 14:20	Sarkutató képességét tesztelő osztály.
testStartGame.java	540	2020.03.29 16:11	Játék elkezdését tesztelő osztály.
testStormOnIceField.java	1 438	2020.03.29 17:47	Vihart tesztelő osztály.
Building.java	5 049	2020.03.25 19:29	Az épületeket kezelő absztrakt osztály.
BuildingPlot.java	2 894	2020.03.25 19:36	A jégmezőn levő épülethelyeket kezelő absztrakt osztály.
DestructableBuilding.java	2 256	2020.03.25 19:45	Az elpusztítható épületeket kezelő absztrakt osztály.
Igloo.java	1 545	2020.03.25 19:50	Iglu típusú épületeket kezelő osztály.
Plains.java	2 936	2020.03.25 19:54	Olyan épülettípust kezelő osztály, ami nem rendelkezik semmilyen képességgel.

Tent.java	3 240	2020.04.22 16:18	Sátort reprezentáló osztály.
Digsite.java	579	2020.03.25 19:58	Ásható objektumok interfésze.
IceField.java	1 844	2020.03.25 20:01	A játékot felépítő egységeket kezelő osztály.
IceUnit.java	13 003	2020.03.25 20:07	Jégtáblát kezelő osztály.
StormyField.java	471	2020.03.25 20:51	A játékban levő viharos jégmezőknek az interfésze.
UnstableIceUnit.java	5 503	2020.03.25 20:57	Instabil jégtáblát kezelő osztály.
Main.java	273	2020.03.28 13:05	A tesztelés main osztályát tartalmazza, amely kezeli a teszeseteket.

autotester.ps1	2 384	2020.04.24. 13:07	A teszteket futtató szkript.
test1.1.in	102	2020.04.24. 13:09	A névnek megfelelő teszt inputja.
test1.1.out	137	2020.04.24. 13:11	A fájlnévnek megfelelő teszt elvárt outputja.
test1.2.in	25	2020.04.24. 13:12	A névnek megfelelő teszt inputja.
test1.2.out	137	2020.04.24. 13:13	A fájlnévnek megfelelő teszt elvárt outputja.
test10.in	284	2020.04.24. 13:15	A névnek megfelelő teszt inputja.
test10.out	1 158	2020.04.24. 13:16	A fájlnévnek megfelelő teszt elvárt outputja.
test11.in	442	2020.04.24. 13:17	A névnek megfelelő teszt inputja.
test11.out	1 647	2020.04.24. 13:18	A fájlnévnek megfelelő teszt elvárt outputja.
test12.in	213	2020.04.24. 13:20	A névnek megfelelő teszt inputja.
test12.out	697	2020.04.24. 13:22	A fájlnévnek megfelelő teszt elvárt outputja.
test13.in	195	2020.04.24. 13:25	A névnek megfelelő teszt inputja.
test13.out	928	2020.04.24. 13:26	A fájlnévnek megfelelő teszt elvárt outputja.
test14.in	397	2020.04.24. 13:28	A névnek megfelelő teszt inputja.
test14.out	1 668	2020.04.24. 13:29	A fájlnévnek megfelelő teszt elvárt outputja.
test15.in	282	2020.04.24. 13:30	A névnek megfelelő teszt inputja.
test15.out	1 368	2020.04.24. 13:32	A fájlnévnek megfelelő teszt elvárt outputja.
test16.in	115	2020.04.24. 13:33	A névnek megfelelő teszt inputja.

test16.out	625	2020.04.24. 13:35	A fájlnévnek megfelelő teszt elvárt outputja.
test17.in	143	2020.04.24. 13:36	A névnek megfelelő teszt inputja.
test17.out	624	2020.04.24. 13:38	A fájlnévnek megfelelő teszt elvárt outputja.
test18.in	141	2020.04.24. 13:39	A névnek megfelelő teszt inputja.
test18.out	622	2020.04.24. 13:40	A fájlnévnek megfelelő teszt elvárt outputja.
test19.in	122	2020.04.24. 13:42	A névnek megfelelő teszt inputja.
test19.out	688	2020.04.24. 13:44	A fájlnévnek megfelelő teszt elvárt outputja.
test2.1.in	146	2020.04.24. 13:46	A névnek megfelelő teszt inputja.
test2.1.out	421	2020.04.24. 13:47	A fájlnévnek megfelelő teszt elvárt outputja.
test2.2.in	43	2020.04.24. 13:50	A névnek megfelelő teszt inputja.
test2.2.out	421	2020.04.24. 13:51	A fájlnévnek megfelelő teszt elvárt outputja.
test20.in	69	2020.04.24. 13:52	A névnek megfelelő teszt inputja.
test20.out	146	2020.04.24. 13:53	A fájlnévnek megfelelő teszt elvárt outputja.
test21.in	87	2020.04.24. 13:54	A névnek megfelelő teszt inputja.
test21.out	358	2020.04.24. 13:56	A fájlnévnek megfelelő teszt elvárt outputja.
test22.in	221	2020.04.24. 13:57	A névnek megfelelő teszt inputja.
test22.out	1 194	2020.04.24. 13:58	A fájlnévnek megfelelő teszt elvárt outputja.
test23.in	832	2020.04.24. 14:00	A névnek megfelelő teszt inputja.
test23.out	3 624	2020.04.24. 14:02	A fájlnévnek megfelelő teszt elvárt outputja.
test24.in	262	2020.04.24. 14:03	A névnek megfelelő teszt inputja.
test24.out	704	2020.04.24. 14:04	A fájlnévnek megfelelő teszt elvárt outputja.
test25.in	303	2020.04.24. 14:05	A névnek megfelelő teszt inputja.
test25.out	1 464	2020.04.24. 14:08	A fájlnévnek megfelelő teszt elvárt outputja.
test26.in	321	2020.04.24. 14:09	A névnek megfelelő teszt inputja.

test26.out	1 426	2020.04.24. 14:10	A fájlnévnek megfelelő teszt elvárt outputja.
test27.in	590	2020.04.24. 14:11	A névnek megfelelő teszt inputja.
test27.out	1 778	2020.04.24. 14:12	A fájlnévnek megfelelő teszt elvárt outputja.
test28.in	91	2020.04.24. 14:14	A névnek megfelelő teszt inputja.
test28.out	594	2020.04.24. 14:15	A fájlnévnek megfelelő teszt elvárt outputja.
test29.in	171	2020.04.24. 14:16	A névnek megfelelő teszt inputja.
test29.out	1 300	2020.04.24. 14:19	A fájlnévnek megfelelő teszt elvárt outputja.
test3.in	295	2020.04.24. 14:21	A névnek megfelelő teszt inputja.
test3.out	291	2020.04.24. 14:23	A fájlnévnek megfelelő teszt elvárt outputja.
test30.in	240	2020.04.24. 14:25	A névnek megfelelő teszt inputja.
test30.out	1 832	2020.04.24. 14:28	A fájlnévnek megfelelő teszt elvárt outputja.
test31.in	228	2020.04.24. 14:29	A névnek megfelelő teszt inputja.
test31.out	1 830	2020.04.24. 14:31	A fájlnévnek megfelelő teszt elvárt outputja.
test32.in	498	2020.04.24. 14:32	A névnek megfelelő teszt inputja.
test32.out	2 430	2020.04.24. 14:33	A fájlnévnek megfelelő teszt elvárt outputja.
test33.in	236	2020.04.24. 14:35	A névnek megfelelő teszt inputja.
test33.out	962	2020.04.24. 14:36	A fájlnévnek megfelelő teszt elvárt outputja.
test34.in	272	2020.04.24. 14:38	A névnek megfelelő teszt inputja.
test34.out	1 352	2020.04.24. 14:39	A fájlnévnek megfelelő teszt elvárt outputja.
test35.in	295	2020.04.24. 14:42	A névnek megfelelő teszt inputja.
test35.out	1 052	2020.04.24. 14:46	A fájlnévnek megfelelő teszt elvárt outputja.
test36.in	149	2020.04.24. 14:47	A névnek megfelelő teszt inputja.
test36.out	1 188	2020.04.24. 14:48	A fájlnévnek megfelelő teszt elvárt outputja.
test37.in	173	2020.04.24. 14:50	A névnek megfelelő teszt inputja.

test37.out	1 180	2020.04.24. 14:51	A fájlnévnek megfelelő teszt elvárt outputja.
test38.in	176	2020.04.24. 14:52	A névnek megfelelő teszt inputja.
test38.out	1 180	2020.04.24. 14:55	A fájlnévnek megfelelő teszt elvárt outputja.
test39.in	253	2020.04.24. 14:56	A névnek megfelelő teszt inputja.
test39.out	1 216	2020.04.24. 14:58	A fájlnévnek megfelelő teszt elvárt outputja.
test4.in	170	2020.04.24. 15:00	A névnek megfelelő teszt inputja.
test4.out	521	2020.04.24. 15:02	A fájlnévnek megfelelő teszt elvárt outputja.
test40.in	111	2020.04.24. 15:03	A névnek megfelelő teszt inputja.
test40.out	596	2020.04.24. 15:05	A fájlnévnek megfelelő teszt elvárt outputja.
test41.in	111	2020.04.24. 15:07	A névnek megfelelő teszt inputja.
test41.out	590	2020.04.24. 15:08	A fájlnévnek megfelelő teszt elvárt outputja.
test42.in	111	2020.04.24. 15:09	A névnek megfelelő teszt inputja.
test42.out	595	2020.04.24. 15:11	A fájlnévnek megfelelő teszt elvárt outputja.
test43.in	361	2020.04.24. 15:12	A névnek megfelelő teszt inputja.
test43.out	1 806	2020.04.24. 15:13	A fájlnévnek megfelelő teszt elvárt outputja.
test44.in	111	2020.04.24. 15:15	A névnek megfelelő teszt inputja.
test44.out	595	2020.04.24. 15:16	A fájlnévnek megfelelő teszt elvárt outputja.
test45.in	141	2020.04.24. 15:17	A névnek megfelelő teszt inputja.
test45.out	769	2020.04.24. 15:20	A fájlnévnek megfelelő teszt elvárt outputja.
test46.in	141	2020.04.24. 15:22	A névnek megfelelő teszt inputja.
test46.out	894	2020.04.24. 15:24	A fájlnévnek megfelelő teszt elvárt outputja.
test47.in	130	2020.04.24. 15:25	A névnek megfelelő teszt inputja.
test47.out	626	2020.04.24. 15:26	A fájlnévnek megfelelő teszt elvárt outputja.
test48.in	133	2020.04.24. 15:28	A névnek megfelelő teszt inputja.

test48.out	598	2020.04.24. 15:30	A fájlnévnek megfelelő teszt elvárt outputja.
test49.in	143	2020.04.24. 15:33	A névnek megfelelő teszt inputja.
test49.out	596	2020.04.24. 15:34	A fájlnévnek megfelelő teszt elvárt outputja.
test5.in	340	2020.04.24. 15:35	A névnek megfelelő teszt inputja.
test5.out	885	2020.04.24. 15:36	A fájlnévnek megfelelő teszt elvárt outputja.
test50.in	129	2020.04.24. 15:38	A névnek megfelelő teszt inputja.
test50.out	670	2020.04.24. 15:39	A fájlnévnek megfelelő teszt elvárt outputja.
test51.in	243	2020.04.24. 15:40	A névnek megfelelő teszt inputja.
test51.out	1 404	2020.04.24. 15:42	A fájlnévnek megfelelő teszt elvárt outputja.
test52.in	132	2020.04.24. 15:43	A névnek megfelelő teszt inputja.
test52.out	901	2020.04.24. 15:44	A fájlnévnek megfelelő teszt elvárt outputja.
test53.in	233	2020.04.24. 15:46	A névnek megfelelő teszt inputja.
test53.out	1 194	2020.04.24. 15:48	A fájlnévnek megfelelő teszt elvárt outputja.
test54.in	108	2020.04.24. 15:49	A névnek megfelelő teszt inputja.
test54.out	254	2020.04.24. 15:51	A fájlnévnek megfelelő teszt elvárt outputja.
test6.in	182	2020.04.24. 15:53	A névnek megfelelő teszt inputja.
test6.out	942	2020.04.24. 15:55	A fájlnévnek megfelelő teszt elvárt outputja.
test7.in	157	2020.04.24. 15:58	A névnek megfelelő teszt inputja.
test7.out	350	2020.04.24. 16:00	A fájlnévnek megfelelő teszt elvárt outputja.
test8.in	304	2020.04.24. 16:02	A névnek megfelelő teszt inputja.
test8.out	1 138	2020.04.24. 16:04	A fájlnévnek megfelelő teszt elvárt outputja.
test9.in	207	2020.04.24. 16:08	A névnek megfelelő teszt inputja.
test9.out	818	2020.04.24. 16:12	A fájlnévnek megfelelő teszt elvárt outputja.
testmap1.map	516	2020.04.24. 17:15	A map teszteléshez szükséges fájl.

testmap2.map	607	2020.04.24. 17:31	A map teszteléshez szükséges fájl.
--------------	-----	-------------------	------------------------------------

### 13.1.2 Fordítás és telepítés

1. Csomagoljuk ki az projlab-2020.zip mappát.
2. Navigálunk a kicsomagolt projlab-2020 mappába.
3. Nyissunk egy terminált/parancssort az projlab-2020 mappában található src mappához navigálva, majd hajtsuk végre a következő parancsokat:
4. a.) Linux / MacOS alatt: find -name "\*.java" > sources.txt
5. b.) Windows alatt: dir /s /B \*.java > sources.txt
5. javac @sources.txt -encoding utf8

### 13.1.3 Futtatás

Az előző lépések után az src mappában maradva a terminál / parancssorban ki kell adni a következő parancsot: java hu.minesweepers.Main

## 13.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Iuhos Zoltán	J8LOUL	20 %
Kaposvári Márk Martin	RAYVF8	20 %
Sulyok Botond	BM2BBM	20 %
Labancz Tamás Dominik	A9HZBF	20 %
Traxler Bálint	IXNOYB	20 %

**13.3Napló**

<b>Kezdet</b>	<b>Időtartam</b>	<b>Résztvevők</b>	<b>Leírás</b>
2020.05.13. 13:00	4 óra	Kaposvári Labancz Iuhos Sulyok Traxler	Értekezlet. A feladat értelmezése, megvalósítás megtervezése, feladatok kiosztása.
2020.05.13. 17:00	2 óra	Labancz	Tevékenység: Felhasználó képek keresése, készítése.
2020.05.14. 14:00	5 óra	Sulyok	Tevékenység: Megjelenítő osztályok implementálása.
2020.05.15. 14:00	5 óra	Traxler	Tevékenység: Kezelői felület módosítása, javítása, bővítése.
2020.05.16. 13:00	5 óra	Kaposvári	Tevékenység: Mentést és betöltést kezelő osztályok implementálása, ellenőrzések, tesztelések.
2020.05.16. 13:00	5 óra	Iuhos	Tevékenység: Enumok készítése, kódba illesztése. A kód általános átnézése. Program tesztelése.
2020.05.16. 18:00	5 óra	Kaposvári Labancz Iuhos Sulyok Traxler	Értekezlet: Elvégzett munka értékelése, utolsó hibák javítása, tesztelések. Összefoglaló dokumentum részleteinek megbeszélése, féléves teljesítmény értékelése.
2020.05.17. 14:00	3 óra	Labancz	Tevékenység: Dokumentumok elkészítése, formázása, véglegesítése.

# **14. Feladat: Összefoglalás**

**14 – the\_minesweepers**

Konzulens:  
**Fetter László**

## **Csapattagok**

Iuhos Zoltán	J8LOUL	zoltan.iuhos@gmail.com
Kaposvári Márk Martin	RAYVF8	xmarko45@gmail.com
Sulyok Botond	BM2BBM	botond.sulyok99@gmail.com
Labancz Tamás Dominik	A9HZBF	labancz.tamas@gmail.com
Traxler Bálint	IXNOYB	traxlerbalint@gmail.com

2020.05.18.

## 14. Összefoglalás

### 14.1 A projektre fordított összes munkaidő

Tag neve	Munkaidő (óra)
Iuhos Zoltán	159
Kaposvári Márk Martin	156
Sulyok Botond	163
Labancz Tamás Dominik	151
Traxler Bálint	161
<b>Összesen</b>	<b>790</b>

- A feltöltött programok forrássorainak száma**

Fázis	Kódsorok száma
Szkeleton	2930 (+2930)
Prototípus	5688 (+2758)
Grafikus változat	8014 (+2326)
<b>Összesen</b>	<b>8014</b>

Megjegyzés: A zárójelben lévő számok az adott fázisban írt kódok számát jelentik, mig az előtte lévő számok a beadáskor meglévő összes sor számát.

### 14.2 Projekt összegzés

#### 14.2.1 Mit tanultak a projektből konkrétan és általában?

Megtanultuk a szoftverfejlesztés alapjait, mégghozzá úgy hogy az elkészült munka könnyen bővíthető és újra felhasználható, köszönhetően az OO szemléletnek és a különböző architekturális és tervezési mintáknak, melyekkel szintén megismerkedhettünk. Emellett megtanultunk csapatban, közösen dolgozni és a git alapjait is elsajátítottuk, mivel munkánk során verziókezelést is használtunk.

#### 14.2.2 Mi volt a legnehezebb és a legkönnyebb?

A legnehezebb számunkra a kezdeti analízis modell kidolgozása volt, és az első beadásban sok hibát is vétettünk, szerencsére ezt később sikerült kijavítani, és úgy érezzük ezen a téren sokat fejlődtünk.

Legkönnyebb az volt, amikor a diagramok alapján meg kellett írni a kódot.

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

Nagyon sokat dolgoztunk a feladatokkal, de ennek meglett az eredménye, mivel magas pontszámokat kaptunk, tehát úgy érezzük megvolt az összhang.

#### 14.2.4 Ha nem, akkor hol okozott ez nehézséget?

-

#### **14.2.5 Milyen változtatási javaslatuk van?**

Talán lehetne egy kicsit részletesebb szöveges leírás az első néhány beadáshoz, főleg az analízis részhez. Továbbá olyan dolgokat is kellett használnunk, amiket nem tanultunk még (például a tervezési minták csak a félév későbbi szakaszában jött elő egy másik tárgy keretében), ez végül nem jelentett problémát, mert az interneten utáranéztünk, de itt is jobb lett volna egy kicsit részletesebb leírás vagy megfogalmazás, hogy pontosan mit kell csinálni és mi az elvárás.

#### **14.2.6 Milyen feladatot ajánlanának a projektre?**

##### **Ötletek:**

- 1.) Antitest vírust öl játék.
- 2.) 4D sakk, amiben tetszőlegesen sok irányba lehet lépni.
- 3.) Kalózok ürrakétéban dimenziókapukon keresztül utaznak, és minél több ananászt kell eljuttatniuk a célohoz.

#### **14.2.7 Egyéb kritika és javaslat**

Legyen minden laborsezető egységesen szigorú és csak tényleg azokat a munkákat pontozzák magas pontszámmal, amik megfelelnek az elvártaknak, tehát a program ténylegesen legyen objektum-orientált és feleljen meg az OO tervezési elveknek is.