

# Követelmény, projekt, funkcionálitás

45 – *nullpointer*

Konzulens:  
Ludmány Balázs

## Csapattagok

Háromi Bence	AK51GI	bence.haromi@gmail.com
Holczer Zsolt	XL9Y0E	zsolti@holczer.net
Kedves Áron Csanád	AV43UW	aron.kedves@gmail.com
Kurdi Barnabás	F0DJNW	barnabas.kurdi@gmail.com
Marton Gergely	FC2WNS	martongergely11@gmail.com

2020.02.24.

## **2. Követelmény, projekt, funkcionalitás**

### **2.1. Bevezetés**

#### **2.1.1. Cél**

A dokumentum a jégtáblás játékprogrammal szemben támasztott követelményeket, a felhasználók a játékkal való interakcióit hivatott leírni.

#### **2.1.2. Szakterület**

Játékprogram

#### **2.1.3. Definíciók, rövidítések**

IIT Tanszék: a feladatot kiíró, illetve az egyes fejlesztési egységek követelményeinek meghatározó szervezete.

Felhőalkalmazások: olyan alkalmazások összessége, amelyek alkalmasak fájlok online felhőben való tárolására.

Inventory: a játékosok által bitrokolt tárgyak tára.

#### **2.1.4. Hivatkozások**

Feladatkiírás, infók:

- IIT Tanszék: <https://www.iit.bme.hu/>

Felhőalkalmazás:

- Google Drive: <https://www.drive.google.com>

Dokumentumszerkesztés:

- Microsoft Office: [https://www.office.com/](https://www.office.com)

Operációs rendszer:

- Windows 10: <https://www.microsoft.com/hu-hu/software-download/windows10>

Verziókezelés:

- BitBucket: <https://bitbucket.org/>
- TortoiseGit: <https://tortoisegit.org/>

Fejlesztőeszközök:

- Java környezet: <https://www.oracle.com/java/>
- IntelliJ IDEA: <https://www.jetbrains.com/idea/>

## **2.1.5. Összefoglalás**

A lentebbi fejezetekben részletesen bemutatásra kerülnek a játék szabályai, funkciói, tervezett fejlesztésének ütemezése, azok részeinek felelősei és a használt eszközök.

## **2.2. Áttekintés**

### **2.2.1. Általános áttekintés**

A játék egy számítógépes program, melynek fő alkotóelemei a pálya, a karakterek és a tárgyak. A pálya a jégmező, ez táblából áll, hóval borított és a hó alatt helyezkednek el a tárgyak. A karakterek a pálya egy adott tábláján kezdenek, innen mozoghatnak a többi táblára, illetve beleeshetnek a vízbe bizonyos esetekben.

A tárgyaknak 2 csoportja van: az egyik amelyek valami plusz segítséget adnak, illetve a jelzőrakéta alkatrészek, amelyek összegyűjtése a játék megnyeréséhez szükséges.

### **2.2.2. Funkciók**

A jégmező egy számítógépes, 3-6 fő által játszható körökre osztott, kooperatív társasjáték, melynek célja kijutni egy veszélyekkel teli jégmezőről egy jelzőrakéta alkatrészeinek felkutatásával, majd annak összeszerelésével.

A felhasználó a játék elindításakor kiválaszthatja, hogy hány játékos fog játszani, és minden játékos kiválaszthatja, hogy sarkutatóként vagy eszkimóként szeretne-e játszani. Miután a kezdeti beállítások kiválasztása megtörtént, a játék elkezdődik.

A játéktér négyzet alakú, a pályán megtalálható mezők szintén. A pályát tenger veszi körül, ez képezi a játéktér szélét. Az, hogy mekkora a pálya, attól függ, hány ember játszik a játékkal (minél többen, annál nagyobb). A játékosok egy mezőről indulnak. A játékmezők jeigtáblákat, vagy a táblán található lyukakat szimbolizálják. minden jeigtábla négy másikkal szomszédos, kivéve a jégmező szélén. A jeigtáblákon különböző tárgyak lehetnek elrejtve, például a jelzőrakéta alkatrészei, ásó, bűvárruha, kötél vagy élelem. A táblákat a játék kezdetén fedheti hó (0-4 egység), ebben az esetben nem látszik, hogy van-e rajtuk elrejtve tárgy.

A jeigtábláknak két típusa van: vannak stabil és instabil jeigtáblák. A stabil jeigtáblák minden épen maradnak, bárhányan állnak rá, azonban az instabiloknak van egy adott teherbírása (minimum 0, maximum a játékosok száma), aminél ha többen állnak rá, a jeigtábla felborul. A 0 teherbírású jégmezők lyukak a táblán. A tábla teherbírása nem látszik, hogyha hó fedi azt, ellenben a kutató képes ellenőrizni a szomszédos táblák teherbírását. Amennyiben egy játékos lyukra lép, vagy rajta áll egy táblán, amely épp felborul, a játékos belesik a vízbe. Hogyha vele szomszédos mezőn tartózkodik olyan játékos, aki rendelkezik egy előzőleg megtalált kötéllel, képes kimenteni társát a vízből, ez az akció automatikus. Egy másik eszköz a megmenekülésre a bűvárruha, hogyha egy játékos bűvárruhát birtokol, mikor vízbe esik, könnyedén kiúszik a vízből, és tovább mozoghat bármelyik szomszédos mezőre.

Minden játékosnak van valamennyi testhője a játék kezdetén, a kutatók 4, míg az eszkimók 5 testhővel kezdenek. A testhő két módon csökkenhet: hogyha egy játékos belesik a vízbe, hogyha senki nem mentette ki kötél segítségével a vízből, és nincs nála bűvárruha, a körének vége, és testhője 0-ra csökken. A játéktér néhány (random) jeigtábláját bizonyos időközönként hóvihar sújtja, ilyenkor az összes érintett mezőn lévő hőmennyiséget eggyel nő, és hogyha egy játékos az érintett mezőn áll, a testhője csökken eggyel. A játékosok nem látják, hogy mikor

érkezik a következő hóvihar, és azt sem tudják, hogy melyik mezőket fogja érinteni. Az eszkimó képes iglut építeni arra a jégmezőre, amin épp áll, ezzel megvédi az ott tartózkodó játékosokat a jégvihar testhőcsökkentő hatásától. Hogyha egy játékosnak az összes testhője elfogy, a játékos megfagy és meghal, ezzel a játéknak mindenki számára vége. Azonban szerencsére van lehetőség a testhő visszatöltésére: egy élelem elhasználásával eggyel nő a testhő, viszont soha nem nőhet az eredeti kezdőérték fölé.

Minden játékos 4 staminával kezdi a körét, amit felhasználhat (de nem kötelezően) bizonyos akciók elvégzésére. Az összes akció egy staminát fogyaszt el. Ilyen akció például a mozgás. minden játéknak lehetősége van átmozogni egy vele szomszédos (szomszédos élő) jégtáblára arról, amin épp áll. Lehetséges akció még az ásás, amivel a játékos képes eltávolítani egy egységenyi havat arról a mezőről, amin épp elhelyezkedik. Amennyiben a játékos birtokol ásót, egy akcióval két egység havat takaríthat el a tábláról. Hogyha egy táblán már nincs hó, megjelenik a rajta elrejtett tárgy, amit szintén egy akció kiásni. Az a játékos kapja meg az adott tárgyat, aki kiásta azt. Egy akcióba kerül továbbá a szerepspecifikus képesség használata (eszkimóknál az iglu megépítése, kutatóknál a szomszédos mezők teherbírásának vizsgálata).

A játék célja összegyűjteni a jelzőrakéta darabjait, ezekből három különböző található meg a pályán: a pisztoly, a jelzőfény és a patron. Hogyha a játékosok az összes alkatrészt megtalálták és kiásták, egy (bármelyik) mezőre kell vinniük, hogyha mindegyik egy mezőn található meg, és az összes játékos összegyűlt ezen a mezőn, egy akcióért összeszerelhetik a rakétát, és elsüthetik azt, ebben az esetben a játéknak vége, és a játékosok megnyerték a játékot.

### **2.2.3. Felhasználók**

A szoftver használatára csak a Budapesti Műszaki és Gazdaság tudományi Egyetemen nappali tanulói jogviszonnyal rendelkező hallgatói, illetve az Irányítástechnikai és Informatikai Tanszék dolgozói jogosultak.

### **2.2.4. Korlátozások**

A szoftvert kereskedelmi, marketinges célból nem lehet használni. Egyéb korlátozások nincsenek.

### **2.2.5. Feltételezések, kapcsolatok**

Az IIT tanszék oldalát a feladat, illetve az instrukciók és az időpontok megismeréséhez használtuk fel.

Az alkalmazásokat pedig a játék fejlesztéséhez és a csapatban való közös munka megkönnyítésének érdekében alkalmazzuk.

## **2.3. Követelmények**

### **2.3.1. Funkcionális követelmények**

Azono sító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case
FR1	A játékosok lehetnek eszkimók vagy sarkkutatók.	bemutatás	alapvető	IIT tanszék	Játék indítása
FR2	A szerepek választhatók.	bemutatás	alapvető	IIT tanszék	Játék indítása
FR3	Az eszkimók képesek iglut építeni, amiben átvészélhetők a hóviharok.	bemutatás	alapvető	IIT tanszék	Karakter irányítása
FR4	A kutatók képesek megnézni a szomszédos mezők teherbírást.	bemutatás	alapvető	IIT tanszék	Karakter irányítása
FR5	A játék körökre osztott.	bemutatás	alapvető	IIT tanszék	Karakter irányítása
FR6	A játékosok staminával rendelkeznek, amit felhasználhatnak interakciókra.	bemutatás	alapvető	IIT tanszék	Karakter irányítása
FR7	A játékosok minden kört 4 staminával kezdenek.	bemutatás	alapvető	IIT tanszék	Karakter irányítása
FR8	A stamina felhasználható tárgyak felvételére, mozgásra és hóeltakarításra, illetve a	bemutatás	alapvető	IIT tanszék	Karakter irányítása, Tárgyak kezelése

	szereplőspecifikus képesség használatára.				
FR9	Minden akció egy staminába kerül.	bemutatás	alapvető	IIT tanszék	Karakter irányítása,Tárgyak kezelése
FR10	Ha egy játékos lyukra mozog, akkor vízbe esik.	bemutatás	alapvető	IIT tanszék	Karakter irányítása
FR11	Ha lyukba esne a játékos, de szomszédos mezőn van olyan társa, akinél van kötél, akkor a társa mellé kerül.	bemutatás	alapvető	IIT tanszék	Tárgyak kezelése
FR12	Játékos csak szomszédos mezőre léphet.	bemutatás	alapvető	Csapat	Karakter irányítása
FR13	Hogyha egy játékos instabil táblára mozog, és ezzel az túlterhelődik, akkor a tábla átfordul.	bemutatás	alapvető	IIT tanszék	Karakter irányítása
FR14	Ha valaki belesik egy lyukba, akkor a hőenergiája 0-ra csökken a köre végén.	bemutatás	alapvető	Csapat	Karakter irányítása
FR15	A táblákat boríthatja hó.	bemutatás	alapvető	IIT tanszék	Hóvihar indítása
FR16	Egy táblán 0-4 egységnnyi hó lehet.	bemutatás	alapvető	IIT tanszék	Hóvihar indítása
FR17	A táblákba befagyva lehetnek eszközök, hogyha olyan táblán vannak, amin van hó, nem látszanak.	bemutatás	alapvető	IIT tanszék	Tárgyak kezelése
FR18	Nem látható tárgyat nem lehet felvenni.	bemutatás	alapvető	IIT tanszék	Tárgyak kezelése
FR19	A játékosok birtokolhatnak tárgyakat.	bemutatás	alapvető	IIT tanszék	Tárgyak kezelése
FR20	A tárgyak lehetnek pisztoly, alkatrészek, búvárruha, élelem, kötél és lapát.	bemutatás	alapvető	IIT tanszék	Tárgyak kezelése
FR21	Az ásóval két egység hó takarítható el egy stamina felhasználásával.	bemutatás	alapvető	IIT tanszék	Tárgyak kezelése, Karakter irányítása

FR22	Ha a táblán hó van, nem látszik, hogy van-e rajta tárgy.	bemutatás	alapvető	IIT tanszék	Tárgyak kezelése
FR23	A búvárruha megakadályozza a kihűlést a vízben.	bemutatás	alapvető	IIT tanszék	Tárgyak kezelése
FR24	Az élelem visszatölti a hőenergiát.	bemutatás	alapvető	IIT tanszék	Karakter irányítása, tárgyak kezelése
FR25	A játék során keletkezhet hóvihar, amely csökkentheti a testhőt, illetve helyezhet havat néhány mezőre.	bemutatás	alapvető	IIT tanszék	Hóvihar indítása
FR26	Ha egy játékosnak elfogy a hőenergiája, akkor meghal.	bemutatás	alapvető	IIT tanszék	Karakter irányítása
FR27	Ha bármelyik játékos meghal, akkor a játéknak vége.	bemutatás	alapvető	IIT tanszék	Játék befejezése
FR28	Ha a játékosok egy táblára állnak az összes rakétaalkatrésszel, és valemelyikük összeszereli, részeket, akkor megnyerték a játékot.	bemutatás	alapvető	IIT tanszék	Játék befejezése

### 2.3.2. Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás
RR01	A szoftvert Java nyelven kell fejleszteni.	bemutatás	alapvető	Csapat
RR02	A szoftver futás közben nem haladhatja meg az 1 GB-os memóriahasználatot.	bemutatás	fontos	Csapat
RR03	A szoftver teljes mérete nem haladhatja meg az 1,5 GB-ot.	bemutatás	fontos	Csapat
RR04	A szoftvert Windows 10 operációs rendszerre kell írni.	bemutatás	alapvető	Csapat
RR05	A futtató számítógép operációs rendszere Windows 10 kell legyen	bemutatás	alapvető	Csapat

### 2.3.3. Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás
A01	Minden követelménynek teljesülnie kell.	bemutatás	alapvető	feladat
A02	A játék futtatásához java környezet szükséges.	bemutatás	alapvető	feladat

### 2.3.4. Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás
NF01	A projekt szöveges dokumentumait, illetve a kódhoz kapcsolódó fájlokat felhőben kell tárolni.	bemutatás	fontos	Csapat
NF02	A szöveges dokumentumok tárolására a Google Drive használandó.	bemutatás	opcionális	Csapat
NF03	A kóddal kapcsolatos fájlokat a csapat egy Git alkalmazáson (Bitbucket) keresztül kell megossza egymással.	bemutatás	opcionális	Csapat
NF04	A szoftver egyes moduljaihoz teszteseteket kell készíteni, amelyek megfelelő lefutását ellenőrizni kell.	bemutatás	fontos	Csapat

## 2.4. Lényeges use-case-ek

### 2.4.1. Use-case leírások

<b>Use-case neve</b>	Hóvihar indítása
<b>Rövid leírás</b>	A rendszer hóvihart indít, amely a pálya néhány mezőjét érinti.
<b>Aktorok</b>	System
<b>Forgatókönyv</b>	<p>1. A rendszer a játékosok lépései után véletlenszerűen dönt arról, hogy keletkezik-e hóvihar.</p> <p><b>1.A.1.</b> A rendszer úgy dönt hogy ne keletkezzen hóvihar.</p> <p><b>1.B.1.</b> A rendszer úgy dönt hogy legyen hóvihar.</p> <p><b>1.B.1.A.1.</b> A rendszer kiválaszt a pályán adott számú mezőt, és végigmegy azokon.</p> <p><b>1.B.1.A.1.A.1.</b> A rendszer megpróbál elhelyezni egy egység havat a mezőn.</p> <p><b>1.B.1.A.1.A.1.B.1.</b> Elhelyezi a havat a mezőn.</p> <p><b>1.B.1.A.1.A.2.</b> A rendszer megnézi vannak-e játékosok a mezőn, ha vannak, végigmegy rajtuk.</p> <p><b>1.B.1.A.1.A.2.A.1.</b> Ha nincs iglu a mezőn, akkor csökkenti a mezőn lévő játékosok testhőjét eggel.</p> <p><b>1.B.1.A.1.A.2.B.1.</b> Ha van iglu a mezőn, akkor nem csökkenti a játékosok testhőjét.</p>

<b>Use-case neve</b>	Játék indítása
<b>Rövid leírás</b>	A program indításakor a játékosok beállítják, hogy hogyan szeretnének játszani.
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<p>1. Az első belépő kiválasztja, hogy hányan fognak játszani.</p> <p><b>1.A.1.</b> Ez a szám 3-6 közötti, ekkor létrejönnek a karakterek.</p> <p><b>1.B.1.</b> Nem 3-6 közötti játékosot akar beállítani, ekkor nem jönnek létre a karakterek, és a program kijelzi a hibát.</p> <p><b>1.A.1.A.1.</b> A játékosok egymás után kiválasztják a karakterüket.</p> <p><b>1.A.1.A.1.A.1.</b> A játékos eszkimó típusú karakter mellett dönt.</p> <p><b>1.A.1.A.1.B.1.</b> A játékos sarkkutató típusú karakter mellett dönt.</p> <p>2. A játékos elindítja a játékot.</p> <p><b>2.A.1.</b> A játékosok nincsenek megfelelően beállítva, ekkor nem kezdődik a játék és a program hibát dob.</p> <p><b>2.B.1.</b> A játékosok jól vannak beállítva.</p> <p><b>2.B.1.A.1.</b> Legenerálódik a pálya, a rajta lévő mezőkkel és tárgyakkal.</p> <p><b>2.B.1.A.1.A.1.</b> A játékosok karakterei elhelyeződnek a pálya egy stabil mezőjén.</p>

Use-case neve	Karakter mozgatása
Rövid leírás	A karakterek mozgatása a pályán és a játék megnyeréséhez szükséges jelzőrakéta kilövése.
Aktorok	Player, System
Forgatókönyv	<p><b>1.</b> A játékos mozgást kezdeményez egy irányba, ha van legalább egy staminája.</p> <p><b>1.A.1.</b> A rendszer ellenőrzi, hogy tud-e arra mozogni és ha tud, elmozdítja (pl. pálya szélén nem tud kifelé mozogni).</p> <p><b>1.A.1.A.1.</b> A játékos stabil mezőre lépett, a játékos staminája egyetlen csökken és az akció véget ér.</p> <p><b>1.A.1.B.1.</b> A játékos instabil mezőre lépett.</p> <p><b>1.A.1.B.1.A.1.</b> A mező terhelhetősége nagyobb, mint a rajta lévő játékosok száma. A játékos staminája egyetlen csökken és az akció véget ért.</p> <p><b>1.A.1.B.1.B.1.</b> A játékosok száma nagyobb a mezőn, mint a mező terhelhetősége, ekkor a mező lyukká válik és minden rajta lévő játékos vízbe esik. A játékos köre megszakad</p> <p><b>1.A.1.B.1.B.1.A.1.</b> Ha az új lyuk melletti mezők valamelyikén van játékos akinél van kötél, akkor az össze játékos aki a lyukban volt, mellé kerül.</p> <p><b>1.A.1.B.1.B.1.A.1.</b> Ha nincs, akkor mindenki a lyukban marad, azok, akiken nincs bűvárruha, elvesztik az összes testhőjüket.</p> <p><b>1.A.1.C.1.</b> A játékos lyukba lépett.</p> <p><b>1.A.1.C.1.A.1.</b> A játékosnak van bűvárruhája, ekkor a játékos staminája egyetlen csökken és az akció véget ér, de a játékos köre nem szakad meg.</p> <p><b>1.A.1.C.1.B.1.</b> A játékosnak nincs bűvárruhája, ekkor a játékos köre megszakad, de ez az akció még lefut.</p> <p><b>1.A.1.C.1.B.1.A.1.</b> Ha a játékos mezője melletti mezők közül valamelyiken van olyan játékos akinél van kötél, akkor automatikusan mellé kerül. A játékos staminája egyetlen csökken, az akció és a játékos köre véget ér.</p> <p><b>1.A.1.C.1.B.1.B.1.</b> Ha nincs, aki kihúzza, akkor a játékos testhője 0-ra csökken, az akció és a köre véget ér.</p> <p><b>1.A.1.A.2.A.1.B.1.</b> Ha nincs köteles ember a közelben a bűvárruhát nem viselők hőenergiája 0-ra csökken és véget ér a kör.</p> <p><b>2.</b> minden mozgás után a rendszer végigmegy a játékosok inventory-ján.</p> <p><b>2.A.1.</b> A játékosoknak még nincs meg minden jelzőpisztoly-alkatrész.</p> <p><b>2.B.1.</b> A játékosoknak megvan minden jelzőpisztoly-alkatrész.</p> <p><b>2.B.1.A.1.</b> A játékosok nincsenek ugyanazon a mezőn</p> <p><b>2.B.1.B.1.</b> A játékosok ugyanazon a mezőn vannak.</p> <p><b>2.B.1.B.1.A.1.</b> A játékosnak még van legalább egy staminája.</p> <p><b>2.B.1.B.1.A.1.A.1.</b> A játékos nem dönt a rakéta fellövése mellett.</p> <p><b>2.B.1.B.1.A.1.B.1.</b> A játékos a rakéta fellövése mellett dönt, és ezzel a játékosok megnyerték a játékot. A játékos staminája egyetlen csökken és az akció véget ér.</p>

<b>Use-case neve</b>	Ásás
<b>Rövid leírás</b>	A játékosok mezőkről való hó eltakarítását irányítja.
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<p><b>1.</b> A játékos ásást kezdeményezhet, ha van legalább egy staminája.</p> <p><b>1.A.1.</b> Ha a mezőn nincs hó, az akció nem számít és a játékos új akciót kezdhet.</p> <p><b>1.B.1.</b> Ha a mezőn van hó, a játékos áshat.</p> <p><b>1.B.1.A.1.</b> Ha a játékosnak nincs lapátja, a mezőn lévő hómennyiségek és a játékos staminája eggyel csökken és az akció véget ér.</p> <p><b>1.B.1.B.1.</b> Ha a játékosnak van lapátja az akció folytatódik.</p> <p><b>1.B.1.B.1.A.1.</b> Ha a mezőn legalább 2 egység hó van, akkor a mezőn lévő hó 2-vel csökken, a játékos staminája egyel csökken és az akció véget ér.</p> <p><b>1.B.1.B.1.B.1.</b> Ha a mezőn csak egy hó van, akkor a mezőn lévő hómennyiségek és a játékos staminája eggyel csökken és az akció véget ér.</p>

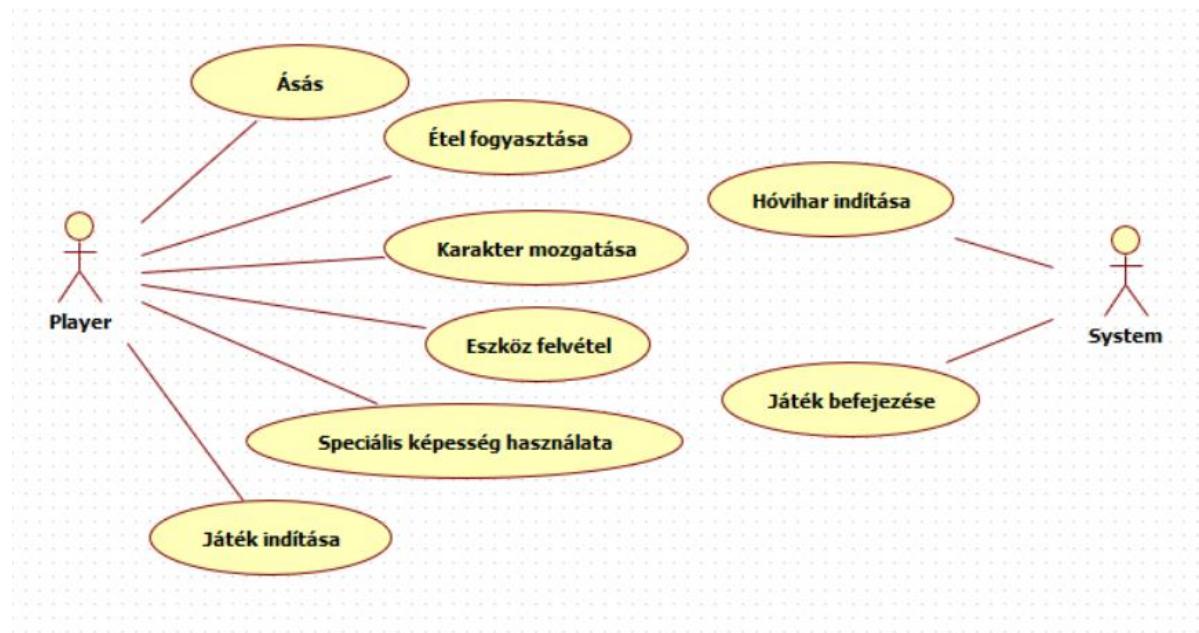
<b>Use-case neve</b>	Speciális képesség használata
<b>Rövid leírás</b>	A játékosok használhatnak speciális képességeket
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<p><b>1.</b> A játékos kezdeményezheti a speciális képességének használatát, ha van legalább egy staminája.</p> <p><b>1.A.1.</b> A játékos karaktere sarkkutató, ekkor megjelenik a játékos mezője melletti mezők teherbírása, a játékos staminája eggyel csökken és az akció véget ér.</p> <p><b>1.B.1.</b> A játékos karaktere eszkimó, ekkor az akció folytatódik.</p> <p><b>1.B.1.A.1.</b> Ha a mezőn van már iglu, akkor nem történik semmi, és az akció véget ér.</p> <p><b>1.B.1.B.1.</b> Ha a mezőn még nincs iglu, akkor az eszkimó épít rá iglut, a staminája eggyel csökken és véget ér az akció.</p>

<b>Use-case neve</b>	Eszköz felvétel
<b>Rövid leírás</b>	A játékosok vehetnek fel tárgyakat ezzel az interakcióval.
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<p><b>1.</b> A játékos kezdeményezheti tárgy felvételét, ha van legalább egy staminája.</p> <p><b>1.A.1.</b> A mezőn hó van, ekkor nem történik semmi és az akció véget ért.</p> <p><b>1.B.1.</b> A mezőn nincs hó, ekkor a tárgy felvételre kerül, a játékos staminája csökken, és az akció véget ér.</p>

<b>Use-case neve</b>	Étel fogyasztása
<b>Rövid leírás</b>	A játékosoknál lehet élelem, amit felhasználhatnak testhőjük növelésére.
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<p><b>1.</b> A játékos kezdeményezhet étkezés akciót, ha van legalább egy staminája.</p> <p><b>1.A.1.</b> Ha a játékosnak nincs élelme, akkor nem történik semmi és az akció véget ér.</p> <p><b>1.B.1.</b> Ha a játékosnak maximális a testhője, akkor nem történik semmi és az akció véget ér.</p> <p><b>1.C.1.</b> Ha a játékosnak van legalább egy élelme, és nem maximális a testhője, akkor a játékos élelmeinek száma és staminája eggyel csökken, a játékos testhője eggyel nő, és az akció véget ér.</p>

<b>Use-case neve</b>	Játék befejezése
<b>Rövid leírás</b>	A rendszer kiértékeli a játék állapotát minden játékos körének végén. A játék befejezéséért felelős.
<b>Aktorok</b>	System
<b>Forgatókönyv</b>	<p><b>1.</b> A rendszer végigmegy a játékosok hőenergiáján.</p> <p><b>1.A.1.</b> Nincs olyan játékos, akinek 0 a hőenergiája.</p> <p><b>1.B.1.</b> Van olyan játékos, akinek 0 a hőenergiája, ez a játékos meghalt. A játék véget ért, a játékosok veszítettek.</p>

### 2.4.2. Use-case diagram



## 2.5. Szótár

- **Akció elvégzése:** A saját körében minden játékos képes négy akció elvégzésére.
- **Ásás:** A játékosok képesek eltávolítani 1 (vagy ha ásóval rendelkeznek, 2) egység havat a mezőről, amin állnak. Hogyha már nincs hó a mezőn, kiáshatják az ott elrejtett tárgyat.
- **Borulás:** Hogyha egy instabil jegtábla túlterhelődik, a tábla felborul, és az összes rajta álló játékos a vízbe esik.
- **Búváruha:** Egy tárgy, amelyet ha visel egy karakter, akkor túlélí a vízbe esést.
- **Eszkimó:** Az egyik választható karakter, amelynek a képessége, hogy iglut tud építeni, 5 testhővel rendelkezik.
- **Evés:** Hogyha egy játékos élelemmel rendelkezik, felhasználhatja azt, hogy visszatöltsön egy egység testhőt, azonban nem mehet a kezdőértéke fölé.
- **Élelem:** Egy tárgy, melynek használatakor 1 egységgel visszatöltődik a karakter testhője.
- **Hóvihar:** Időnként körök végén feltámadhat, melynek hatására az érintett táblákat változó mennyiségű hóval boríthatja be, illetve az ezeken a táblákon tartózkodó karaktereknek csökkenti a testhőjét (feltéve, hogy nincsenek igluban).
- **Iglu:** Eszkimó által építhető menedék. Hóviharban, ha egy táblán van iglu, akkor az azon lévő játékosok az igluban átvézelhetik a hóvihart, így nem csökken a hőenergiájuk.
- **Igluépítés:** Az eszkimó képes iglut építeni arra a mezőre, amin épp áll, amiben átvézelhető a hóvihar testhőveszteség nélkül.
- **Interakció:** Karakterek által történő mozgás, képességhasználat, eszközhasználat vagy ásás.
- **Jelzőrakéta alkatrész:** 3 tárgy tartozik ebbe a kategóriába: pisztoly, jelzőfény, patron. Külön-külön nem használhatóak, viszont egy táblára összegyűjtve az összeset (akár különböző karakterek birtokában) összeépíthetik és elsüthetik, ezzel megnyerve a játékot.
- **Jegtábla:** A jégmező egy egységnyi része, stabilitással rendelkezik, különböző mennyiségű hó boríthatja (0-4 egység), illetve lehetnek rajta belefagyva tárgyak.
- **Kihűlés:** Hogyha egy játékosnak 0-ra csökken a testhője, kihűl és meghal, ezzel a játéknak vége, a játékosok vesztettek.
- **Kimentés:** Hogyha egy játékos vízbe esik, viszont szomszédos mezőn áll másik, olyan játékos, akinél van kötél, kimenti társát, és maga mellé húzza.
- **Kiúszás:** Hogyha olyan játékos esik vízbe, akinek van búvárruhája, kiúszhat a lyukból bármelyik szomszédos mezőre.
- **Kör:** a játék körökből áll, minden körben minden játékos egyszer kerül sorra, és irányíthatja a karakterét. minden kör elején minden karakter 4 staminát kap, amit használhatnak interakciókra.
- **Kötél:** Egy tárgy, melyet ha birtokol egy vízbe eső karakterrel szomszédos táblán álló karakter, akkor megmenti azt a vízbe eséstől.
- **Lapát:** Egy tárgy, amelyet ha birtokol egy karakter, akkor 2 egység havat tud eltakarítani egy staminával az eredeti 1 helyett.
- **Lyuk:** Olyan jegtábla, melyre rálépve vízbe esik a játékos.
- **Mozgás:** a játékosoknak egy stamina felhasználásával minden körben lehetőségük van szomszédos mezőre lépni.
- **Összeszerelés:** hogyha az összes játékos egy mezőn áll, és megszerezték a jelzőrakéta mindhárom alkatrészét, egy akcióért összeszerelhetik, és elsüthetik azt, ezzel megnyerve (és befejezve) a játékot.
- **Sarkkutató:** A másik karakter, amelynek a képessége, hogy a mellette lévő mezők stabilitását meg tudja nézni, 4 testhővel rendelkezik.

- Stabilitás: a jégtáblák jellemzője, lehetnek stabilak, ilyenkor bármennyi karaktert elbírnak, illetve lehetnek instabilak, ilyenkor van egy értékük, ami azt adja meg, hogy mennyi karakter tartózkodhat rajtuk anélkül, hogy felborulna.
- Stamina: Az interakciókra felhasználható fizetőeszköz.
- Tárgy: A karakterek által felvehető, illetve használható eszközök, melyek valamiben segítik a játékost. Felvehető eszközök: búvárruha, lapát, élelem, kötél, valamint a jelzőrakéta-alkatrészek.
- Teherbírás ellenőrzése: A kutató képes a szomszédos táblák teherbírásának ellenőrzésére.
- Testhő: Karakterek tulajdonsága, bizonyos mennyiséggel kezdik a játékot (sarkkutató: 4-gyel, illetve eszkimó 5-tel), és különböző események hatására ennek értéke csökkenhet, illetve visszatölthető, de nem lépheti túl a kezdőértéket. Ha eléri a 0-át a karakter meghal, és a játéknak vége.

## **2.6. Projekt terv**

### **2.6.1. Projekt leadások és határidők**

Határidő	Feladat	Felelős
<b>Február 24.</b>	Követelmény, projekt, funkcionalitás	
<b>Március 2.</b>	Analízis modell kidolgozása 1	Kedves Áron Csanád
<b>Március 9.</b>	Analízis modell kidolgozása 2	Marton Gergely
<b>Március 16.</b>	Szkeleton tervezése	Háromi Bence
<b>Március 23.</b>	Szkeleton - beadás és a forráskód herculesre való feltöltése	Marton Gergely
<b>Március 30</b>	Prototípus koncepciója	Holczer Zsolt
<b>Április 6.</b>	Részletes tervezek	Kurdi Barnabás
<b>Április 20.</b>	<i>Prototípus készítése, tesztelése</i>	Háromi Bence
<b>Április 27.</b>	Prototípus - beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése	Marton Gergely
<b>Május 4.</b>	Grafikus felület specifikációja	Holczer Zsolt
<b>Május 11.</b>	Grafikus változat készítése	Kedves Áron Csanád
<b>Május 18.</b>	Grafikus változat és Összefoglalás - beadás és a forráskód herculesre való feltöltése	Kurdi Barnabás

\*A feladatok felelőseinek nem feladata, hogy többet dolgozzanak a témaban, csak az ő dolguk a csapat haladását felügyelni és szólni ha valahol elmaradt valaki.

## 2.6.2. Fájlok megosztásának módszere, fejlesztési eszközök

- A dokumentumok megosztása és közös szerkesztése Google Drive-on megosztott közös mappán keresztül.
- A fejlesztés során készült szoftver változatok megosztása verzió kezeléssel Bitbuckettel (GIT) történik. A könnyebb használhatóság érdekében TortoiseGit-et is használunk.
- Közös Java fejlesztő környezetként a IntelliJ IDEA fejlesztőkörnyezetet fogjuk használni.
- A dokumentumok szerkesztésére a Microsoft Word-öt és a Google beépített szövegszerkesztőjét fogjuk használni.
- A csapat haladását és a feladatok elosztását a közös Google Driveon lévő progress táblával követjük nyomon.

## 2.7. Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.02.14. 12:30	1,5 óra	Kurdi Háromi Holczer Kedves Marton	Értekezlet. Döntés: Kurdi felállítja a funkcionális követelményeket 2020.02.15. estig.
2020.02.14. 20:00	1 óra	Kurdi	Tevékenység: Kurdi felállította a funkcionális követelményeket.
2020.01.16. 18:00	2 óra	Kurdi Háromi Holczer Kedves Marton	Értekezlet. Döntés: Marton kitölti a 2.2.2, 2.2.4, 2.3.2-es részeket 2020.02.20. estig. Háromi megszerkeszti a fedőlapot és a 2.3.3-as részt.
2020.01.17. 19:00	0,5 óra	Marton	Tevékenység: Marton a 2.2.3; 2.2.4; 2.3.2-es pontokat kitöltötte.
2020.01.17. 20:00	0,5 óra	Háromi	Tevékenység: Háromi elkészítette a fedőlapot és a 2.3.3; 2.3.4-es pontokat.
2020.02.20. 14:00	1 óra	Kurdi Holczer Kedves Marton	Értekezlet. Döntés: Kurdi elkészíti a funkciók részt és feltölти a szótárba az igéket. Háromi feltölti a szótárt főnevekkel, és ellenőrzi az addigi szócikekkel. Holczer elkészíti a Use-case diagramot és feltölti a szótárt fő- és melléknevekkel. Marton kiegészíti és javítja addigi munkáját. Kedves elkészíti a projekttervet és a Use-case leírásokat. Határidő: 2020.02.22. este.
2020.02.20. 18:00	1,5 óra	Kurdi	Tevékenység: Kurdi elkészítette a funkciók (2.2.2.) részt és a szótárból az igéket.
2020.02.20. 20:00	1 óra	Háromi	Tevékenység: Háromi elkészítette a 2.1.5-ös ponot és javította a 2.3.3; 2.3.4-es pontokat.
2020.02.20. 21:00	1 óra	Holczer	Tevékenység:

			Holczer elkészítette a Use-case diagramot és feltöltötte a szótárt melléknevekkel(2.4.2, 2.5).
2020.02.21. 16:00	1 óra	Marton	Tevékenység: Marton javításokat és kiegészítéseket végzett a korábban elkészített munkáján.
2020.02.21. 17:00	1 óra	Háromi	Tevékenység: Háromi elkészítette a szótár főneveit és a 2.2.1-es pontot.
2020.02.21. 18:00	2 óra	Kedves	Tevékenység: Kedves: Projekt terv, Use-case leírások elkészítése (2.6, 2.4.1).
2020.02.22. 16:00		Kurdi Háromi Holczer Kedves Marton	Értekezlet. Döntés: Kedves frissíti a Use-case leírásokat 2020.02.23. estig.
2020.02.22. 18:00	0,5 óra	Kedves	Tevékenység: Kedves: Use-case leírások javítása.

# **Analízis modell kidolgozása 1.**

## **45 – nullpointer**

Konzulens:  
**Ludmány Balázs**

### **Csapattagok**

Háromi Bence	AK51GI	bence.haromi@gmail.com
Holczer Zsolt	XL9Y0E	zsolti@holczer.net
Kedves Áron Csanád	AV43UW	aron.kedves@gmail.com
Kurdi Barnabás	F0DJNW	barnabas.kurdi@gmail.com
Marton Gergely	FC2WNS	martongergely11@gmail.com

## **3.1 Objektum katalógus**

### **3.1.1 Búvárruha**

Vízbeeséskor, ha a karakter visel búvárruhát, akkor nem fullad meg, ki tud úszni egy másik táblára.

### **3.1.2 Eszkimó**

A játékos által választható egyik karakter, amelynek a speciális tulajdonsága, hogy 5 élettel kezd, valamint iglut tud építeni. Az igluban bárki és bárhányan tartózkodhatnak és ebben tartózkodók átvészhetik a hóvihart hőveszteség nélkül.

### **3.1.3 Étel**

A karakter kiáshatja, illetve utána bármikor felhasználhatja 1 hőenergia visszatöltésére.

### **3.1.4 Jégtábla**

A játék jégtáblából áll, amelyek lehetnek stabilak, illetve instabilak is. A stabil jégtáblákon akármennyi játékos állhat, az instabilokon azonban csak meghatározott számú játékos tartózkodhat. Ugyanis, ha az adott instabil mezőn túl sok játékos tartózkodik, akkor az a feje tetejére fordul és mindenkit maga alá temet.

### **3.1.5 Kötél**

Ha egy karakter vízbe esik, nincs búvárruhája, és egy szomszédos táblán lévő barátjának van kötele, akkor az magához húzza, ezzel megmentve a megfulladástól.

### **3.1.6 Lapát**

A játékban az egyik tárgy. Ezt a tárgyat, ha birtokolja a karakterünk akkor 2 egység havat tud eltakarítani 1 egység munkával a kezdeti 1 helyett.

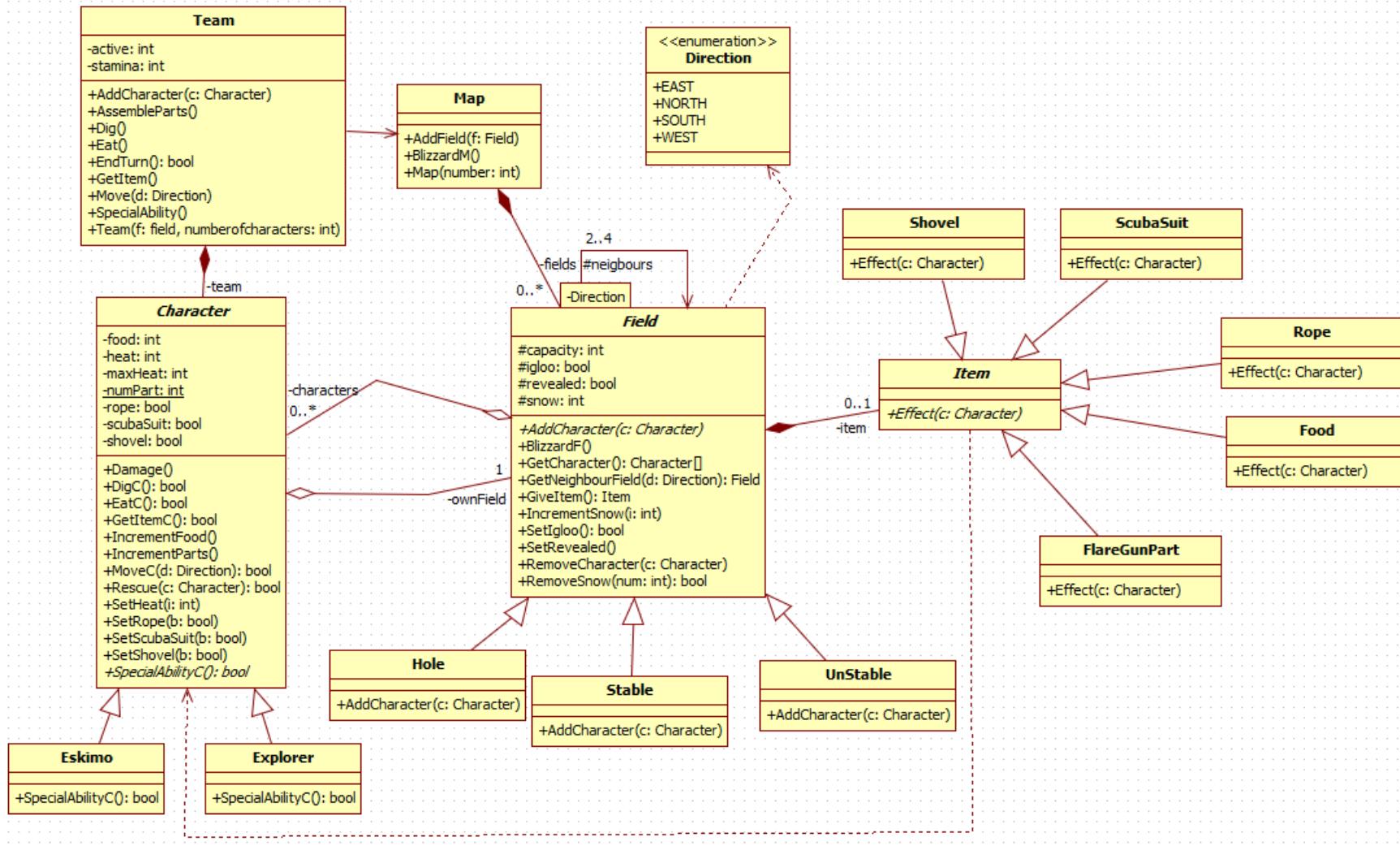
### **3.1.7 Pisztolyalkatrészek**

A játék során gyűjthető tárgyak egy fajtája, amely három elemből áll. A végső cél ezen 3 elem megtalálása, majd az összes játékos egy mezőre állítása.

### **3.1.8 Sarkkutató**

A játékos által egyik választható karakter, aminek a speciális képessége, hogy a mellette lévő mezők terhelhetőségét tudja megnézni. 4 hőenergiával kezd.

## 3.2 Statikus struktúra diagramok



## 3.3 Osztályok leírása

### 3.3.1 Character

- **Felelősség**

A karakter képességeit mozgását kezeli, valamint a tárgyakkal lévő viselkedést.

- **Attribútumok**

- **-food: int:** A karakternél lévő ételek számát tárolja
- **-heat: int:** A karakter életét tárolja
- **-maxHeat: int:** A karakter maximális hőenergiáját tárolja azért, hogy ne lehessen túl lépni és a kétféle karakternek másféle a maximuma
- **-numPart: int:** A karaktereknél lévő rakétaalkatrészek számát tárolja (statikus változó)
- **-rope: bool:** A karakternél van-e kötél
- **-scubaSuit: bool:** A karakternél van-e búvárruha
- **-shovel: bool:** A karakternél van-e lapát

- **Metódusok**

- **void Damage():** A karakter életét (heat) csökkenti pl hóvihar esetén
- **bool DigC():** A mezőn hó eltakarítása (1 vagy 2 lapáttal) ha tudott ásni akkor tér vissza true-val egyébként false-t add vissza
- **bool EatC():** A karakter enni fog egy ételt és növeli az életét eggel ha tudott enni akkor tér vissza true-val egyébként false-t add vissza
- **bool GetItemC():** Felvesz egy itemet a mezőről ha fel tudta venni az eszközt akkor tér vissza true-val egyébként false-t add vissza
- **void IncrementFood():** A karakternél lévő élelem számát növeli 1-gyel
- **void IncrementParts():** A karakternél lévő rakétaalkatrészek számát növeli
- **bool MoveC(d: Direction):** A karakter lép d irányba ha tudott lépni akkor tér vissza true-val egyébként false-t add vissza
- **bool Rescue():** A mellett lévő mezőn lyukba esett társát ha van nála kötél kimenti magához húzza ha sikerül akkor true-val tér vissza
- **void SetHeat(i: int):** A karakter életét lehet beállítani a megadott "i" értékre
- **void SetRope(b: bool):** Beállítja hogy a karakternél van-e kötél
- **bool SpecialAbilityC():** abstract A karakterhez tartozó speciális képességet hajtja végre amennyiben sikerül true-val ha nem akkor false-al tér vissza
- **void setScubaSuit(b: bool):** Beállítja hogy a karakternél van-e búvárruha
- **void setShovel(b: bool):** Beállítja hogy a karakternél van-e lapát
- **bool assamblePartsC():** A karakter megpróbálja összeszerelni jelzőrakétát ha sikerül true-val tér vissza ha nem akkor false-al.

### 3.3.2 Direction

- **Felelősség**

Enumeráció osztály az irányok felsorolására.

- **Attribútumok**

- +NORTH
- +SOUTH
- +EAST
- +WEST

### 3.3.3 Eskimo

- **Felelősség**

Az eszkimóra jellemző dolgokat kezeli

- **Ősosztályok**

Character -> Eskimo

- **Metódusok**

- **bool SpecialAbilityC()**: Az eszkimó speciális képeségét hajtja végre, ami egy iglu építése.

### 3.3.4 Explorer

- **Felelősség**

A sarkkutatóra jellemző dolgokat kezeli

- **Ősosztályok**

Character -> Explorer

- **Metódusok**

- **bool SpecialAbilityC()**: A kutató speciális képességét hajtja végre amiaz hogy a mellette lévő mezők terhelhetőségét megtekinti.

### 3.3.5 Field

- **Felelősség**

Absztrakt osztály, amely azon mezőket kezeli, amelyek a térképet képezik, ahol a karakterek mozoghatnak, illetve ahol a tárgyak elhelyezkedhetnek.

- **Attribútumok**

- **#Field[Direction]: neighbours**: minden mező ismeri az adott irányban lévő szomszédját.
- **Character[0..\*]: characters**: A mezőn álló játékosok összessége.
- **#capacity: int**: Az adott mező kapacitása.
- **#igloo: bool**: Megadja, hogy az adott mezőn van-e iglu.
- **#revealed: bool**: Megadja, hogy egy Sarkkutató felfedte-e a stabilitását a mezőnek
- **#snow: int**: Az adott mezőn lévő hó aktuális mennyisége.

- **Metódusok**

- **void AddCharacter(Character c):** Absztakt függvény, amely hozzáadja a paraméterként megkapott karaktert a Field-hez, illetve egyéb változásokat tesz, attól függően, hogy mely leszármazottakban valósul meg.
- **BlizzardF():** A mezőt a hóvihar evvel a függvénytel sújtja, különböző módosításokat végez a mezővel, illetve az esetlegesen rajta álló játékos(ok)on.
- **Character[] GetCharacter():** Visszaadja a mezőn álló karaktert, vagy karaktereket.
- **Field GetNeighbourField(Direction d):** Visszaadja a paraméterként kapott irányban tartózkodó mezőt. Amennyiben nincs ott mező, akkor null-lal tér vissza.
- **Item GiveItem():** Visszatér a mezőn lévő tárggyal, majd a mező Item attribútumából eltávolítja azt.
- **IncrementSnow(int howmuch):** Megnöveli a hó mennyiséget a mezőn 1-gyel.
- **bool RemoveSnow(int num):** Havat takarít el a megadott mennyiséget(1 vagy 2). Ha sikerül az eltakarítás true-val ha nem akkor false-al tér vissza.
- **void SetIgloo():** Beállítja a mező igloo paraméterét true-ra.
- **void SetRevealed():** Beállítja a mező revealed attribútumát true-ra, ami azt jelenti, hogy egy Sarkkutató megnézte, hogy az adott mező stabil-e vagy sem.
- **void RemoveCharacter(Character c):** Eltávolítja a paraméterként kapott karaktert a mezőről.

### 3.3.6 FlareGunParts

- **Felelősség**

A jelzőrakéta alkatrészeit reprezentálja. Összesen három szerepel belőlük a játékban. Amikor egy játékos felszedi ezt a tárgyat, az növeli a játékos numParts attribútumát eggyel (ezzel az összes többi játékosét is, mivel a numParts statikus).

Megjegyzés: azért nincs külön osztály a különböző alkatrészeknek, mivel funkciójukban egyáltalán nem különböznek, a grafikus megjelenítéshez később az összes Item rendelkezni fog egy texture attribútummal, amely alapján a különböző alkatrészeket meg lehet majd különböztetni, de ezt nem foglalja magába az analízis modell.

- **Ősosztályok**

Item → FlareGunParts

- **Metódusok**

- **Effect(Character c): void:** meghívja a paraméterként megkapott Character IncrementParts() függvényét, ezzel növeli a numParts attribútumot a karakterben. A karakter hívja meg, mikor felszedi a tárgyat.

### 3.3.7 Food

- **Felelősség**

Az étel tárgyat reprezentálja. Amikor a játékos felveszi, akkor a character food attribútumát eggyel növeli.

- **Ősosztályok**

Item -> Food

- **Metódusok**

- **void Effect(Character c):** meghívja a paraméterként kapott Character IncrementFood() függvényét, ami megnöveli eggyel a food attribútumának értékét, nem tér vissza semmivel.

### 3.3.8 Hole

- **Felelősség**

Lyukakat reprezentáló osztály. Ezen mezők stabilitása 0.

- **Ősosztályok**

Field->Hole

- **Attribútumok**

Az ősosztály attribútumai.

- **Metódusok**

- **void AddCharacter(Character c):** Hozzáadja a játékosat a lyukhoz, illetve a szomszédos mezőkön álló játékosoknak (ha vannak) meghívja a Rescue() függvényét.

### 3.3. Item

- **Felelősség**

A tárgyaknak az absztrakt ősosztályát reprezentálja.

- **Metódusok**

- **void Effect(Character c):** ez egy absztrakt metódus, amely a paraméterként kapott Character setttereit hívja meg ezzel beállítva valamely attribútumát.

### 3.3.10 Map

- **Felelősség**

Ez az osztály reprezentálja a pályát. Tárolja a pálya mezőit és felelős a hóviharok végrehajtásáért.

- **Attribútumok**

- **-fields : Field:** A játék mezőit tároló dinamikus tömb.

- **Metódusok**

- **void BlizzardM():** ez a metódus dönti el, hogy történik-e hóvihar valamint végrehajtja azt. Ha van hóvihar, akkor a véletlenszerűen kiválaszt az általa tárolt field-ek közül néhányat és meghívja azoknak a BlizzardF() függvényét.

### 3.3.11 Rope

- **Felelősség**

A kötél tárgyat reprezentálja. Amikor a játékos felveszi, akkor a character rope attribútuma true-ra állítódik.

- **Ősosztályok**

Item -> Rope

- **Metódusok**

- **void Effect(Character c):** meghívja a paraméterként kapott Character SetRope() függvényét, ami a Character rope attribútumát true-ra állítja, nem tér vissza semmivel.

### 3.3.12 ScubaSuit

- **Felelősség**

A búvárruha tárgyat reprezentálja. Ha a karakter viseli, akkor nem fullad meg a vízben.

- **Ősosztályok**

Item->Shovel

- **Metódusok**

- **void Effect(Character c):** meghívja a paraméterként kapott Character SetScubaSuit() függvényét, ami a Character scubaSuit attribútumát true-ra állítja, nem tér vissza semmivel.

### 3.3.13 Shovel

- **Felelősség**

A ásó tárgyat reprezentálja. Ha birtokosnak van ásója, akkor eggyel több egység havat tud eltakarítani.

- **Ősosztályok**

Item -> Shovel

- **Metódusok**

- **void Effect(Character c):** meghívja a paraméterként kapott Character SetShovel() függvényét, ami a Character shovel attribútumát true-ra állítja, nem tér vissza semmivel.

### 3.3.14 Stable

- **Felelősség**

A stabil jégmezőt reprezentálja. Ezen mezők stabilitása -1, ami a végtelent jelenti.

- **Ősosztályok**

Field - > Stable

- **Attribútumok:**

- Az ōsosztály attribútumait örökli.
- **Metódusok**
  - Az ōsosztály metódusait örökli, a AddCharacter() metódust pedig nem írja felül.

### 3.3.15 UnStable

- **Felelősség**

Az instabil jégmezőt reprezentálja. Ezen mezők stabilitása korlátozott, amit, ha meglépne

- **Ősosztályok**

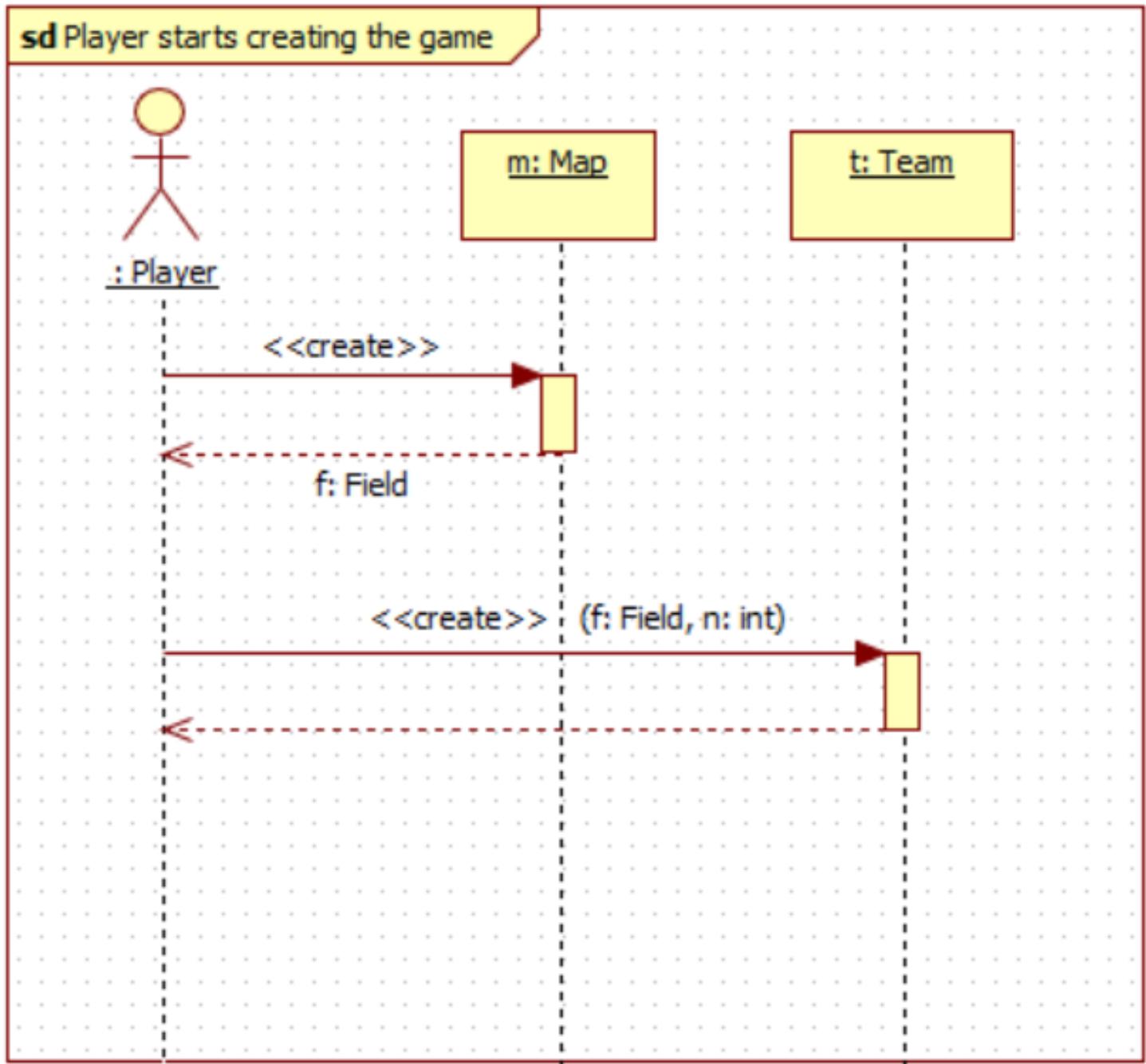
Field->UnStable

- **Attribútumok:**
  - Az ōsosztály attribútumait örökli.
- **Metódusok**
  - **void AddCharacter(Character c):** Hozzáadja a paraméterként kapott karaktert a mezőhöz, illetve ellenőrzi, hogy ezzel a mező átlépte-e a kapacitásának határát. Ha igen, akkor a mező a fejére fordul, ezzel minden rajta álló játékos hőenergiáját 0-ra állítja.

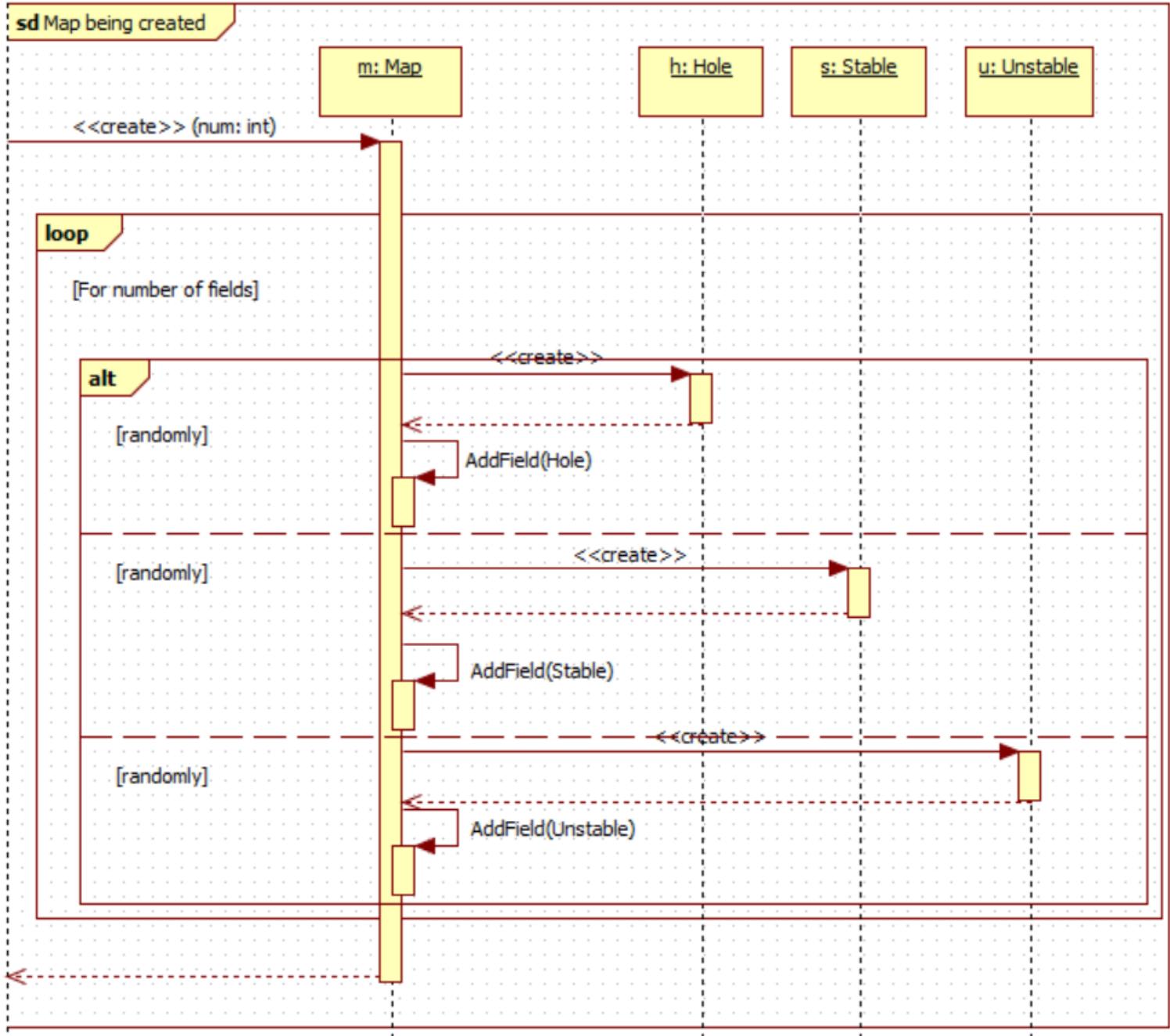
## 3.4 Szekvenciadiagramok

### 3.4.1 Inicializáció

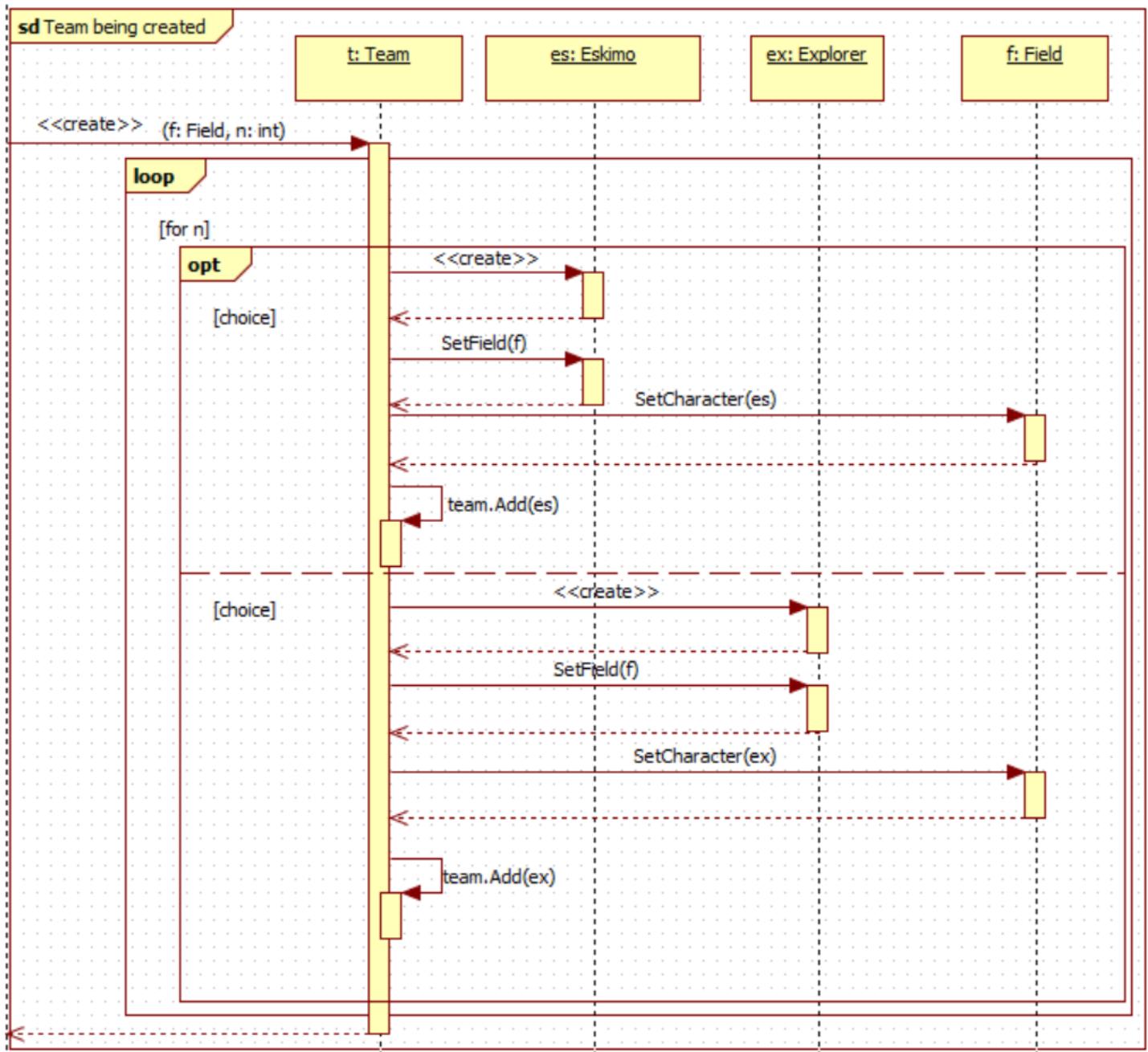
#### 3.4.1.1 Player starts creating the game



### 3.4.1.2 Map being created

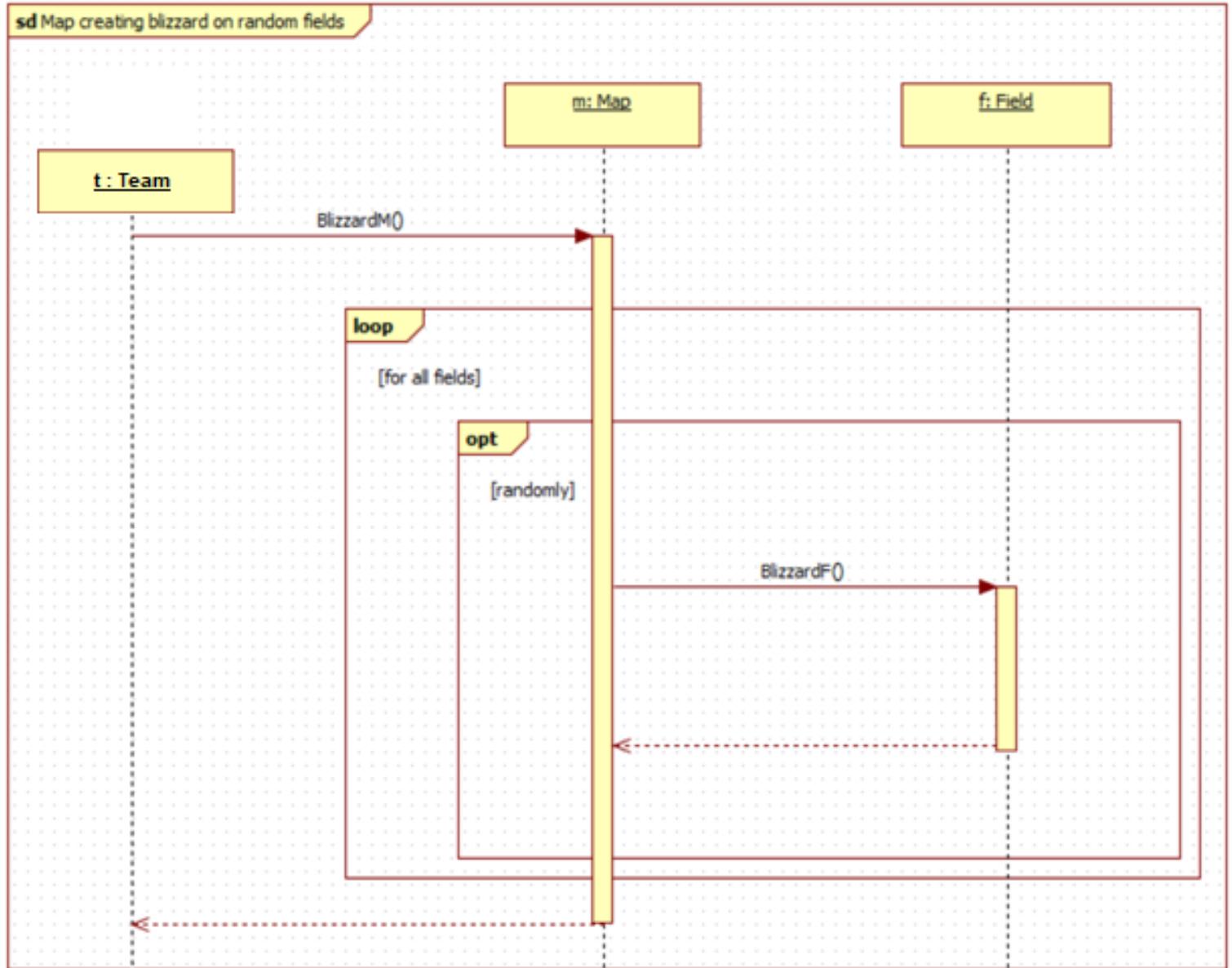


### 3.4.1.3 Team being created

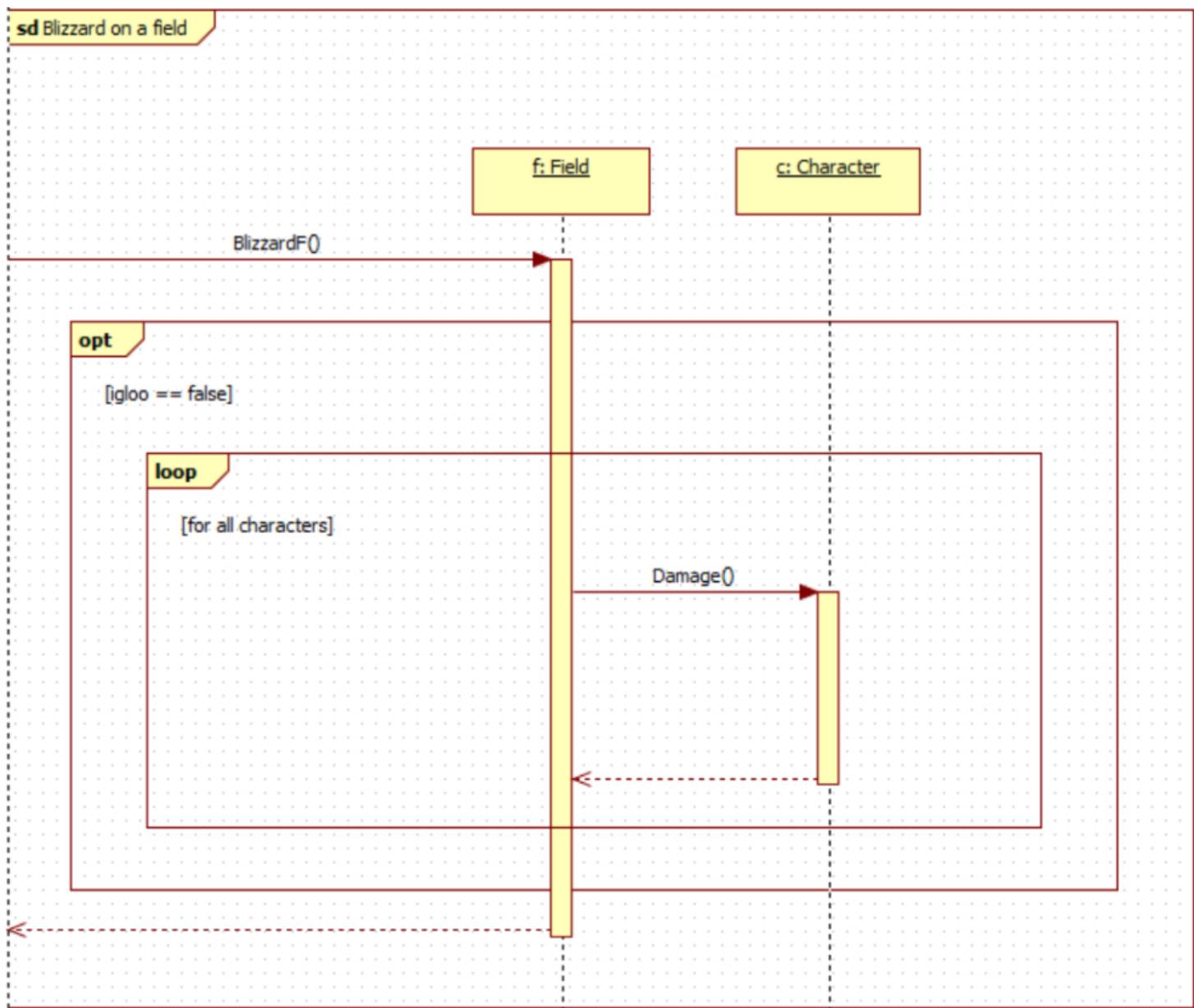


### 3.4.2 Kör befejezése

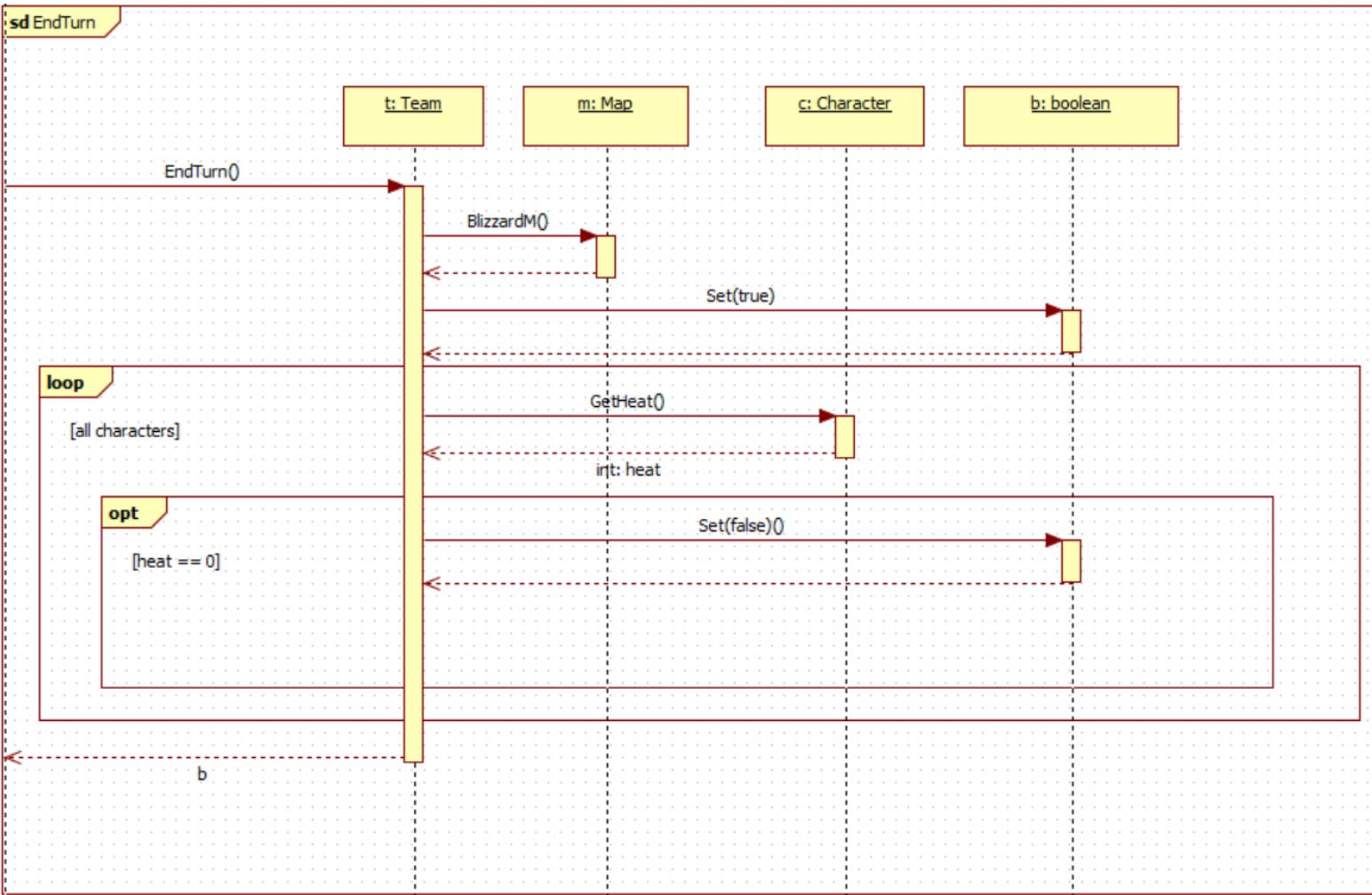
#### 3.4.2.1 Map creating blizzard on random fields



### 3.4.2.2 Blizzard on a field

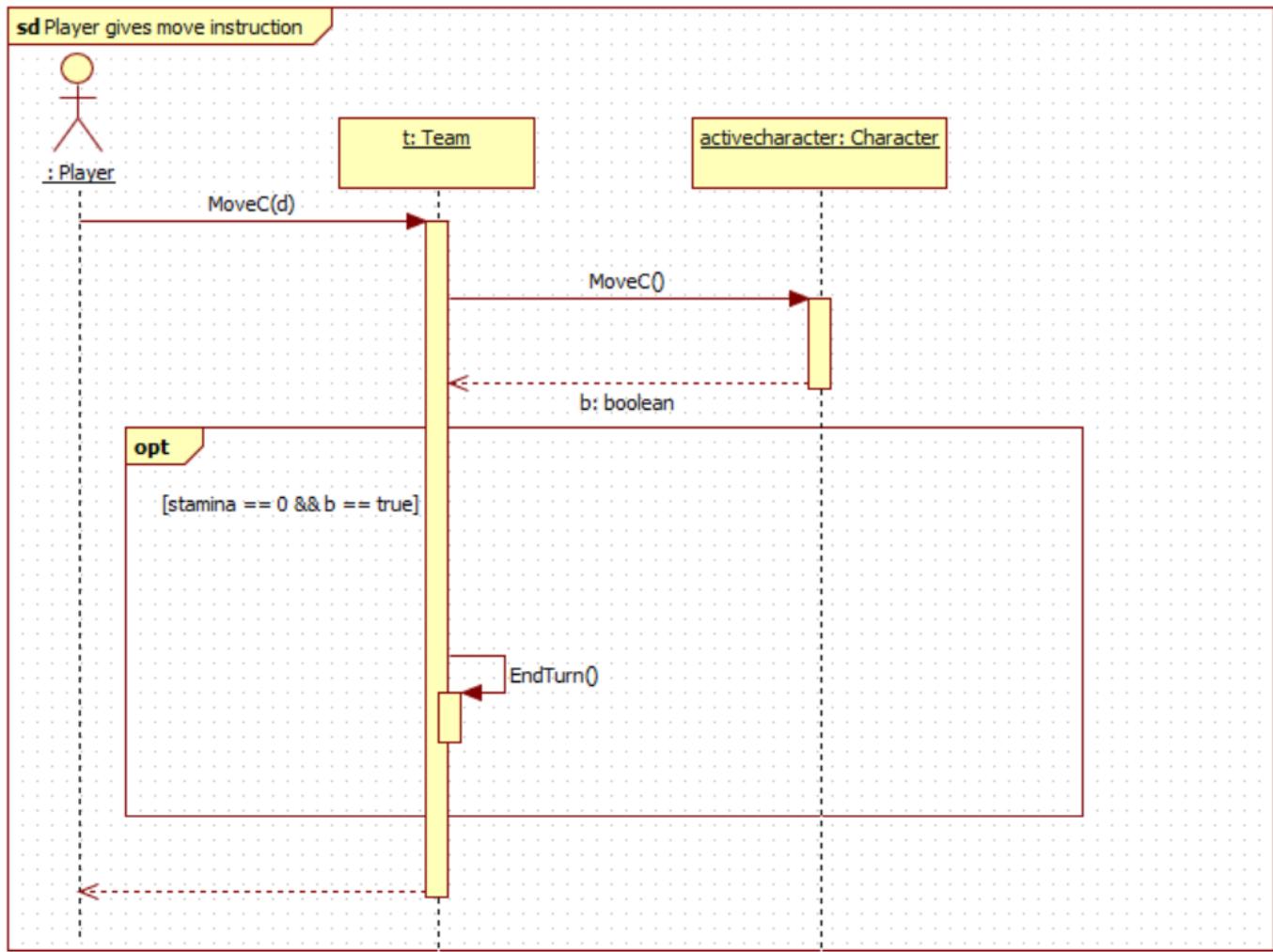


### 3.4.2.3 EndTurn

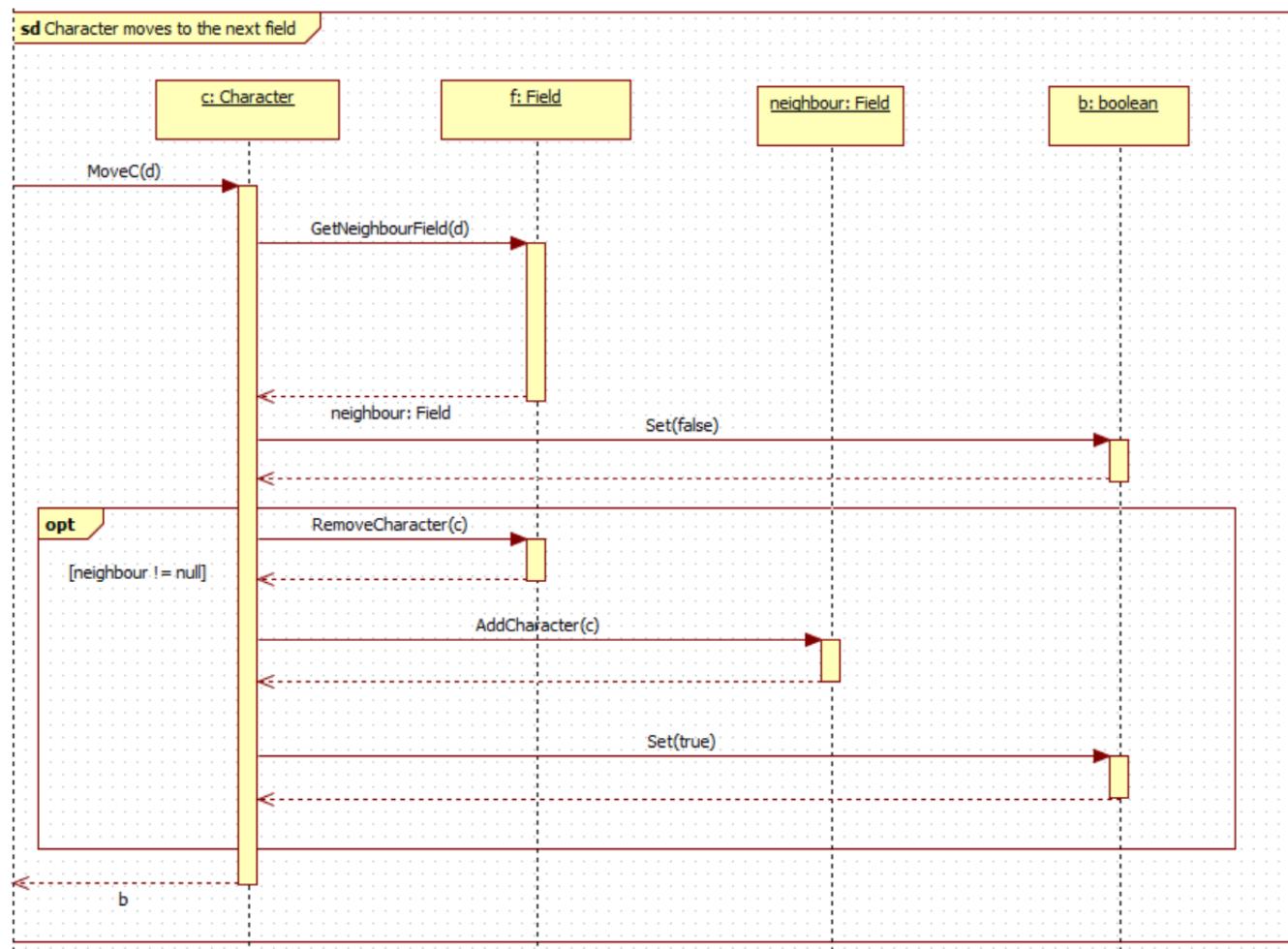


### 3.4.3 Mozgás

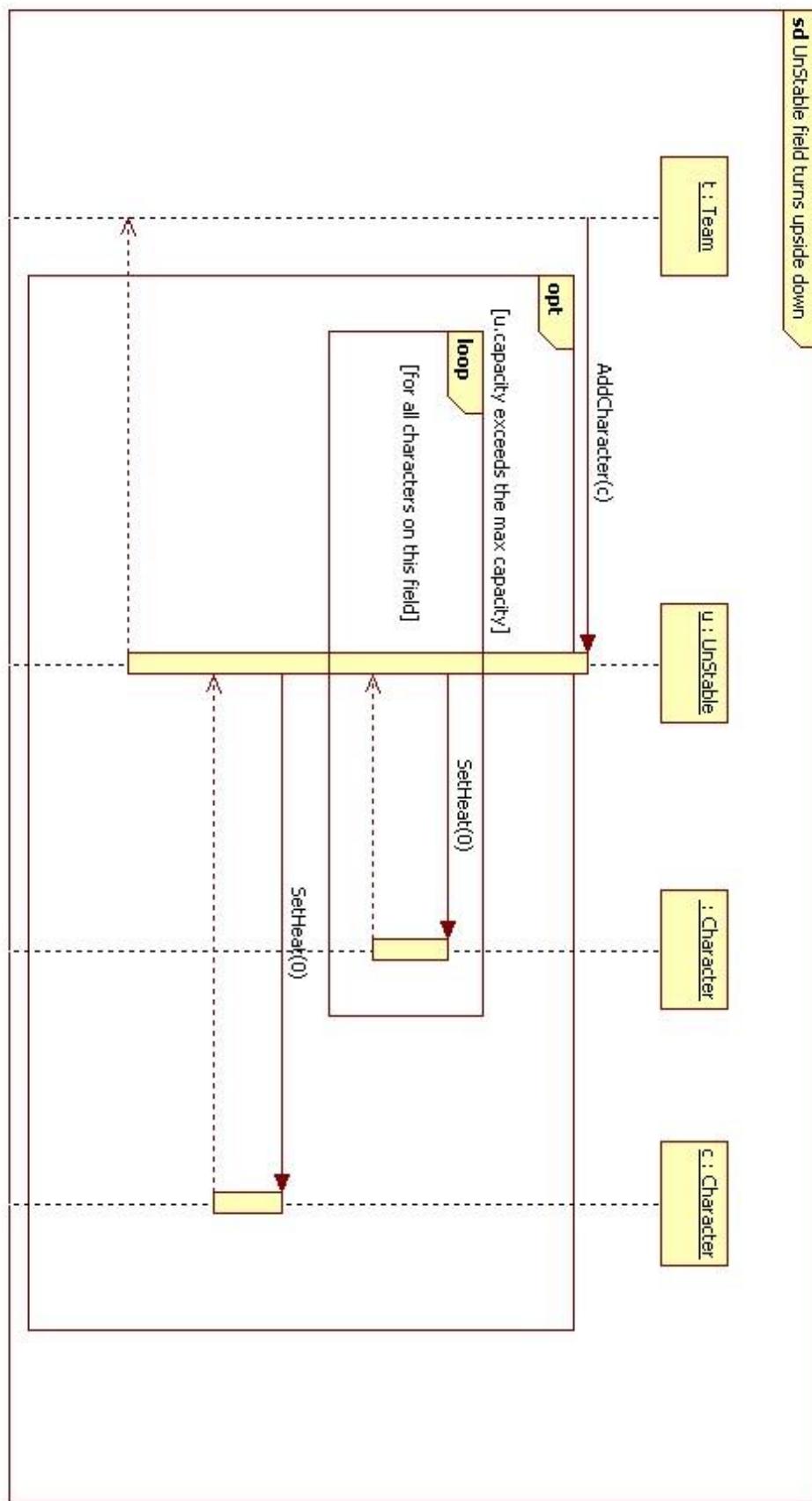
#### 3.4.3.1 Player gives move instruction



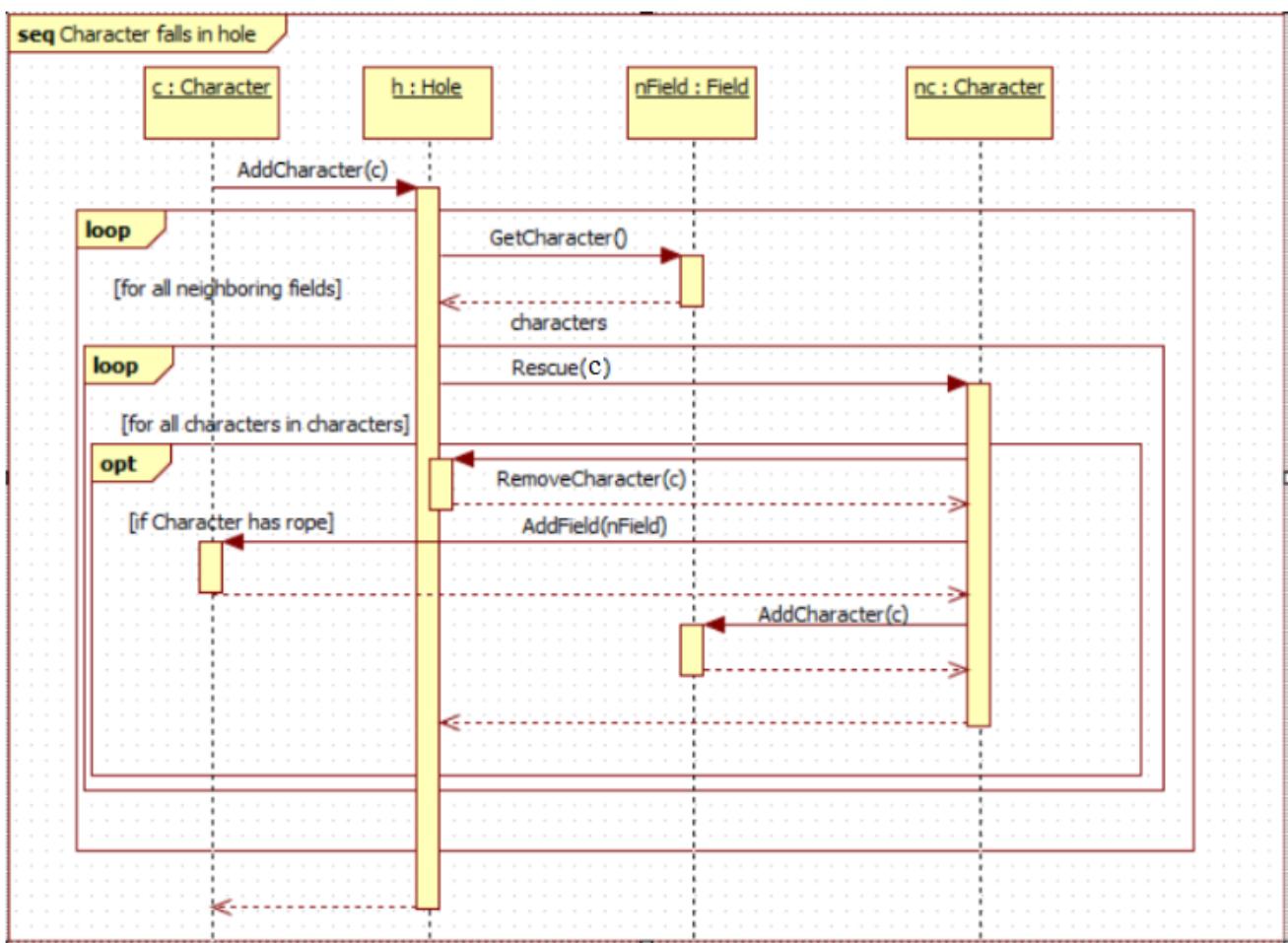
### 3.4.3.2 Character moves to next field



### 3.4.3.3 Unstable field turns upside down

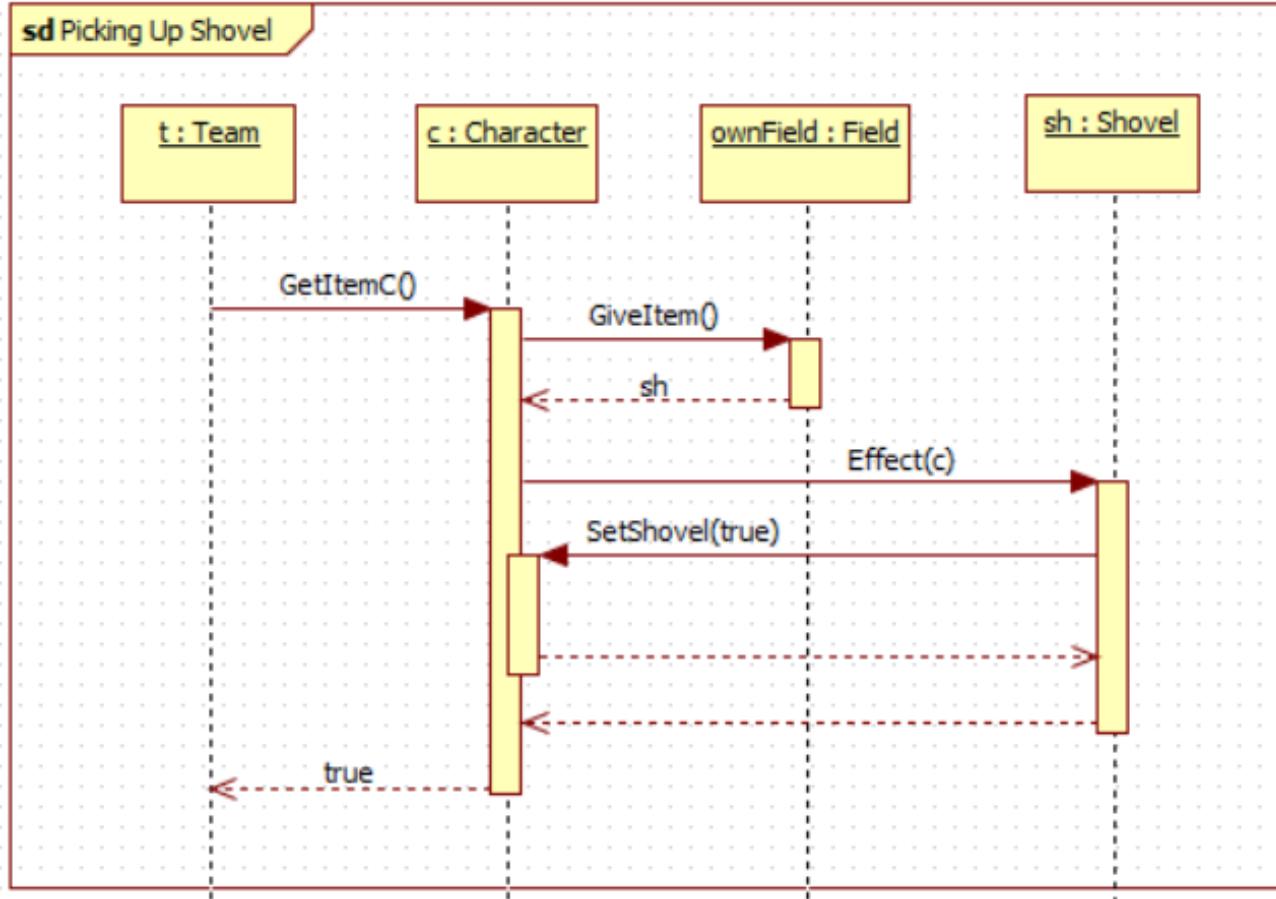


### 3.4.3.4 Character falls in hole

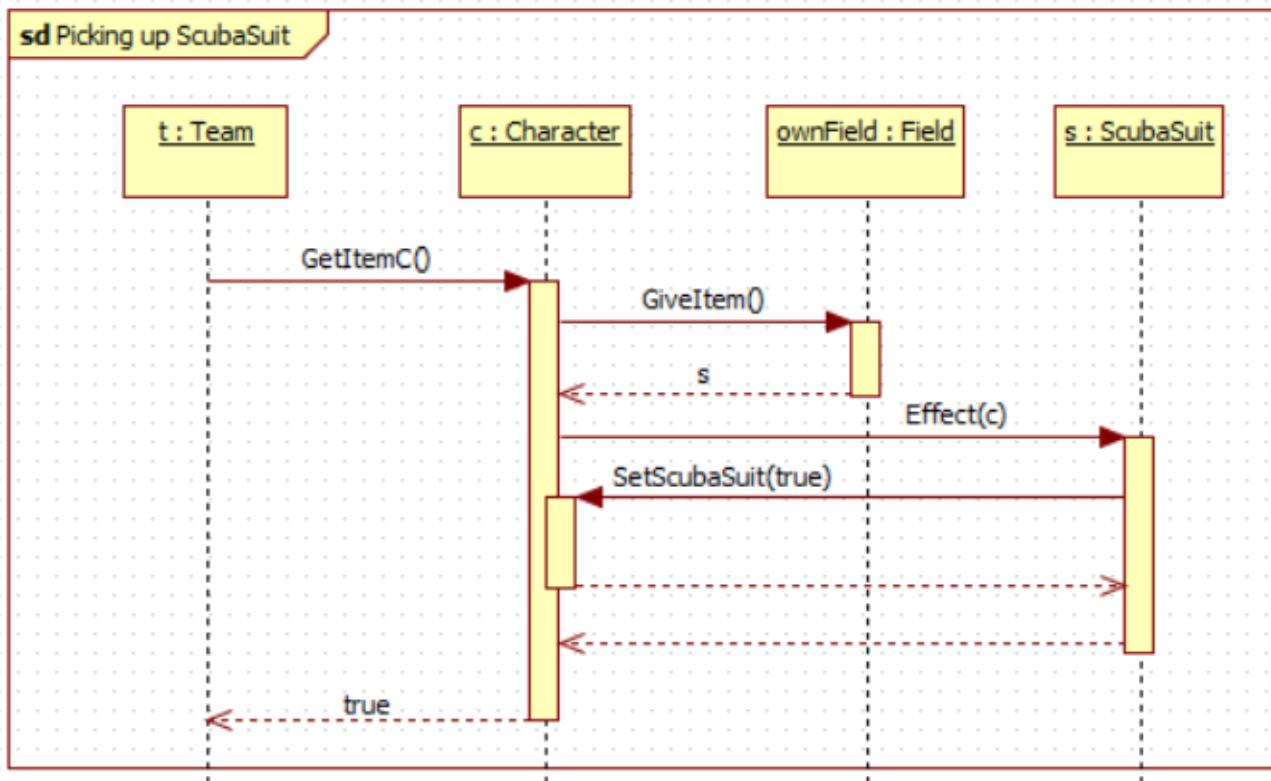


### 3.4.4 Tárgykezelés

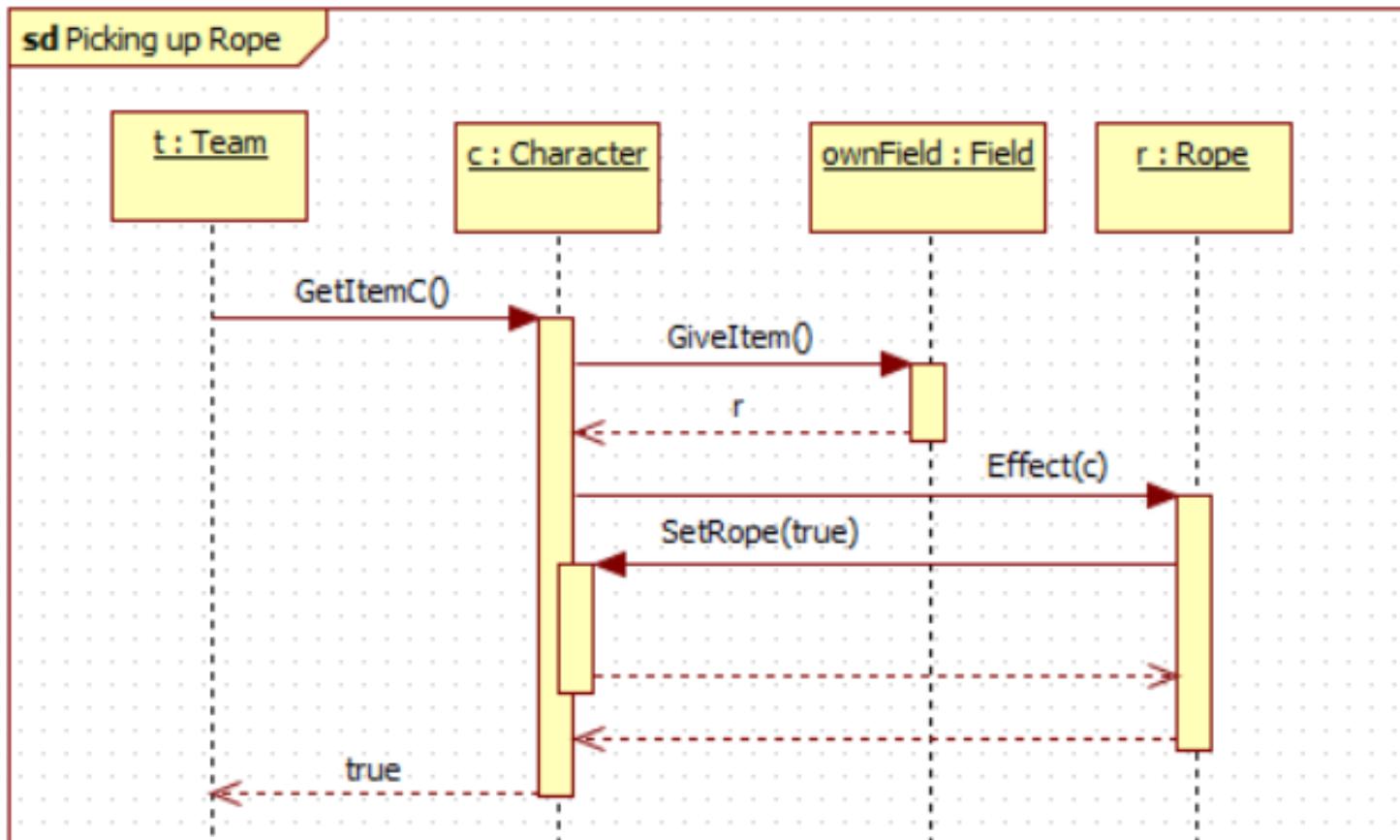
#### 3.4.4.1 Picking up shovel



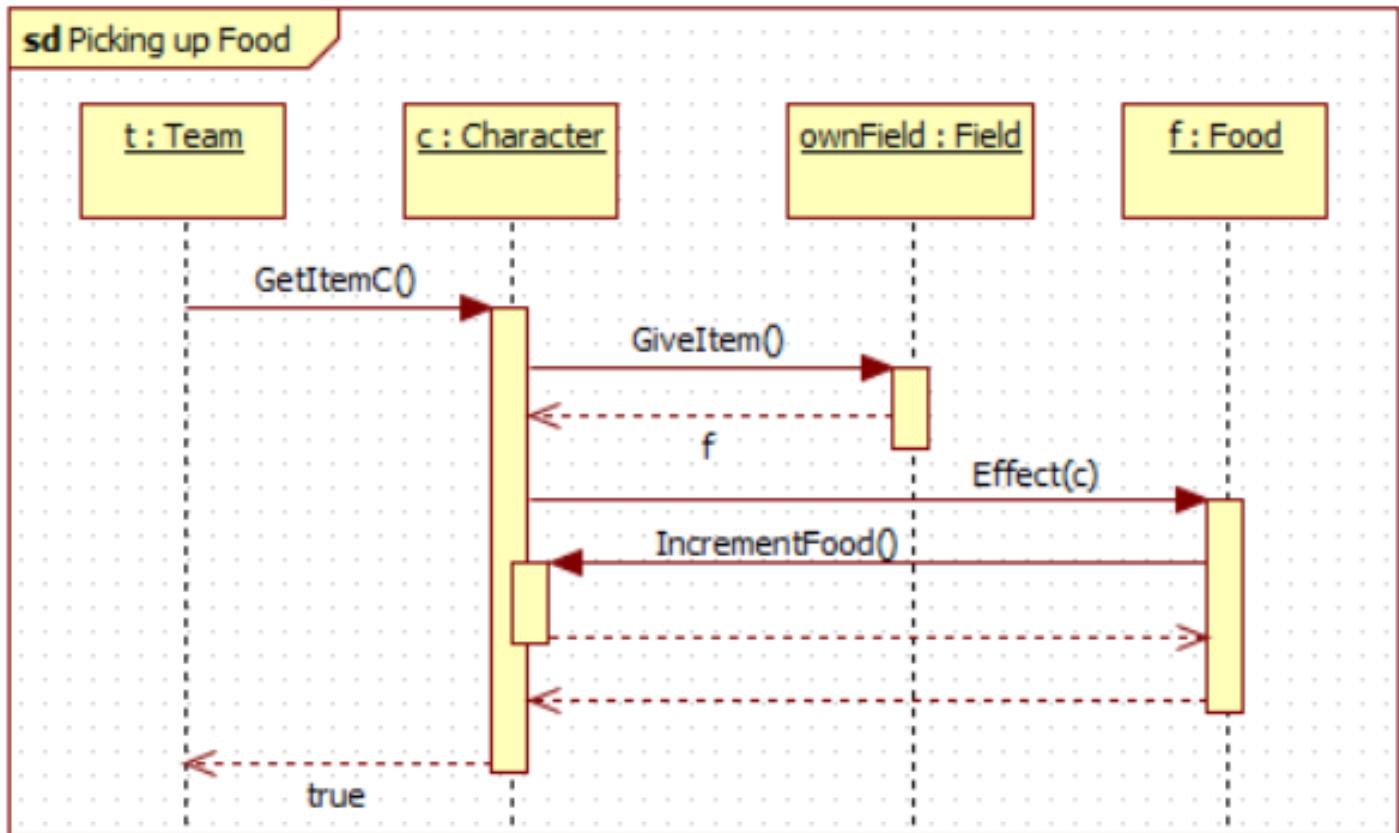
### 3.4.4.2 Picking up ScubaSuit



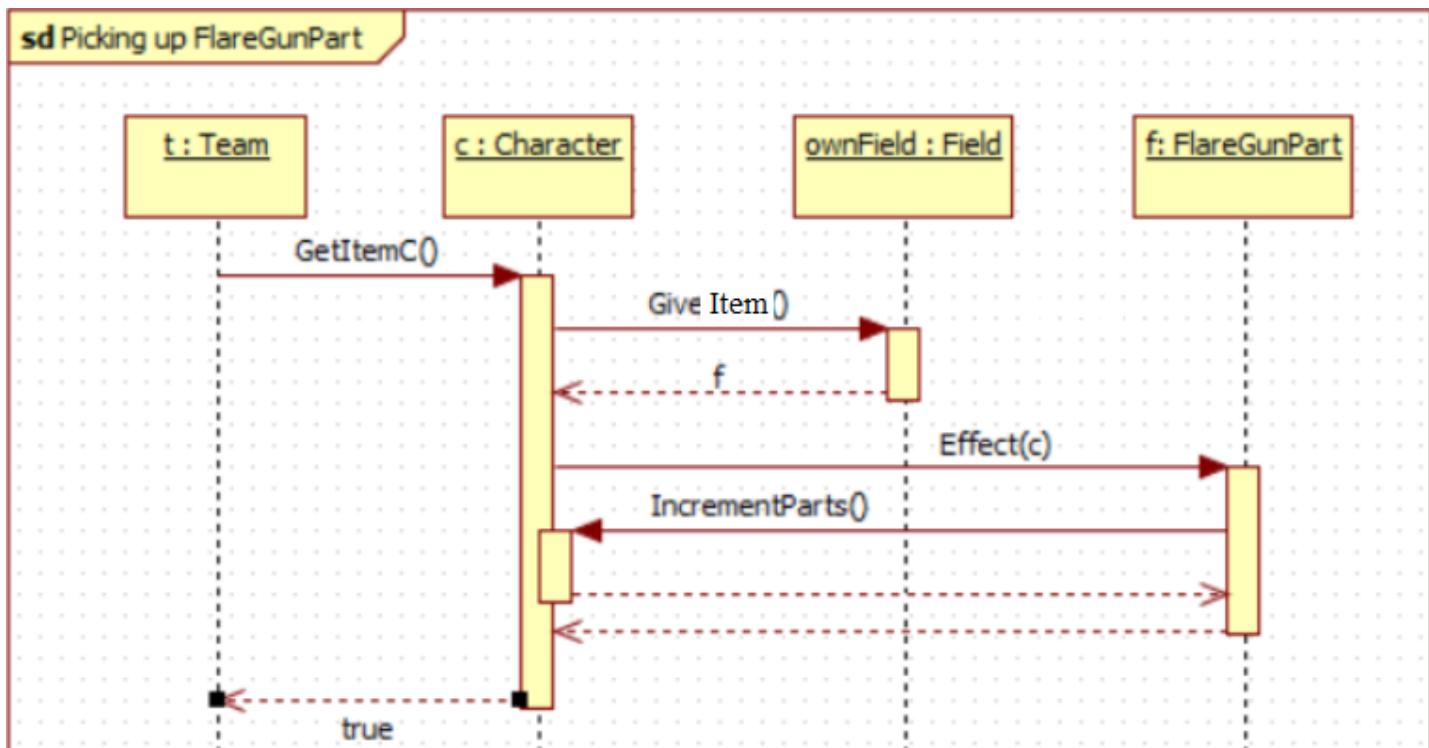
### 3.4.4.3 Picking up Rope



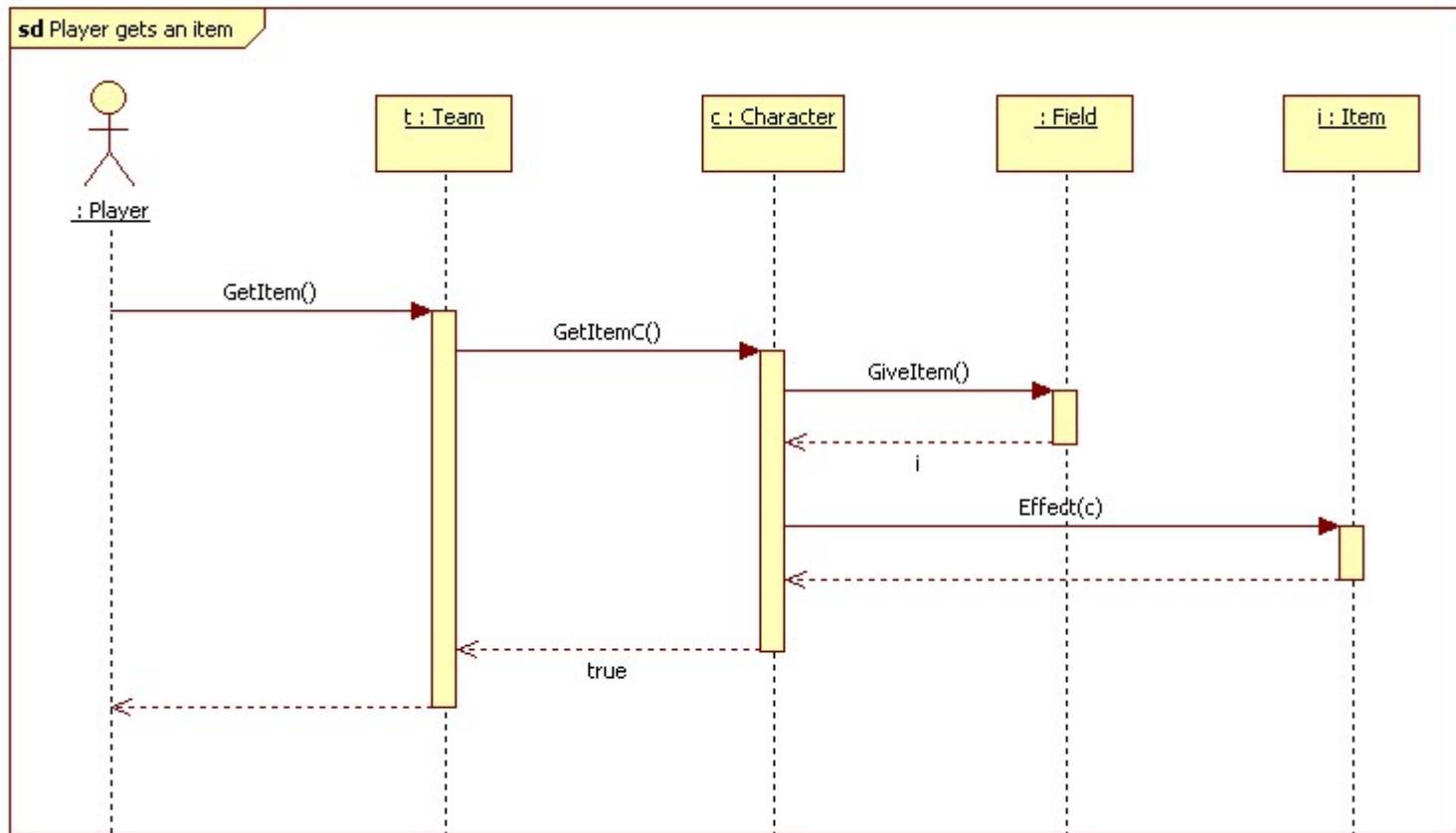
### 3.4.4.4 Picking up food



### 3.4.4.5 Picking up FlareGunPart

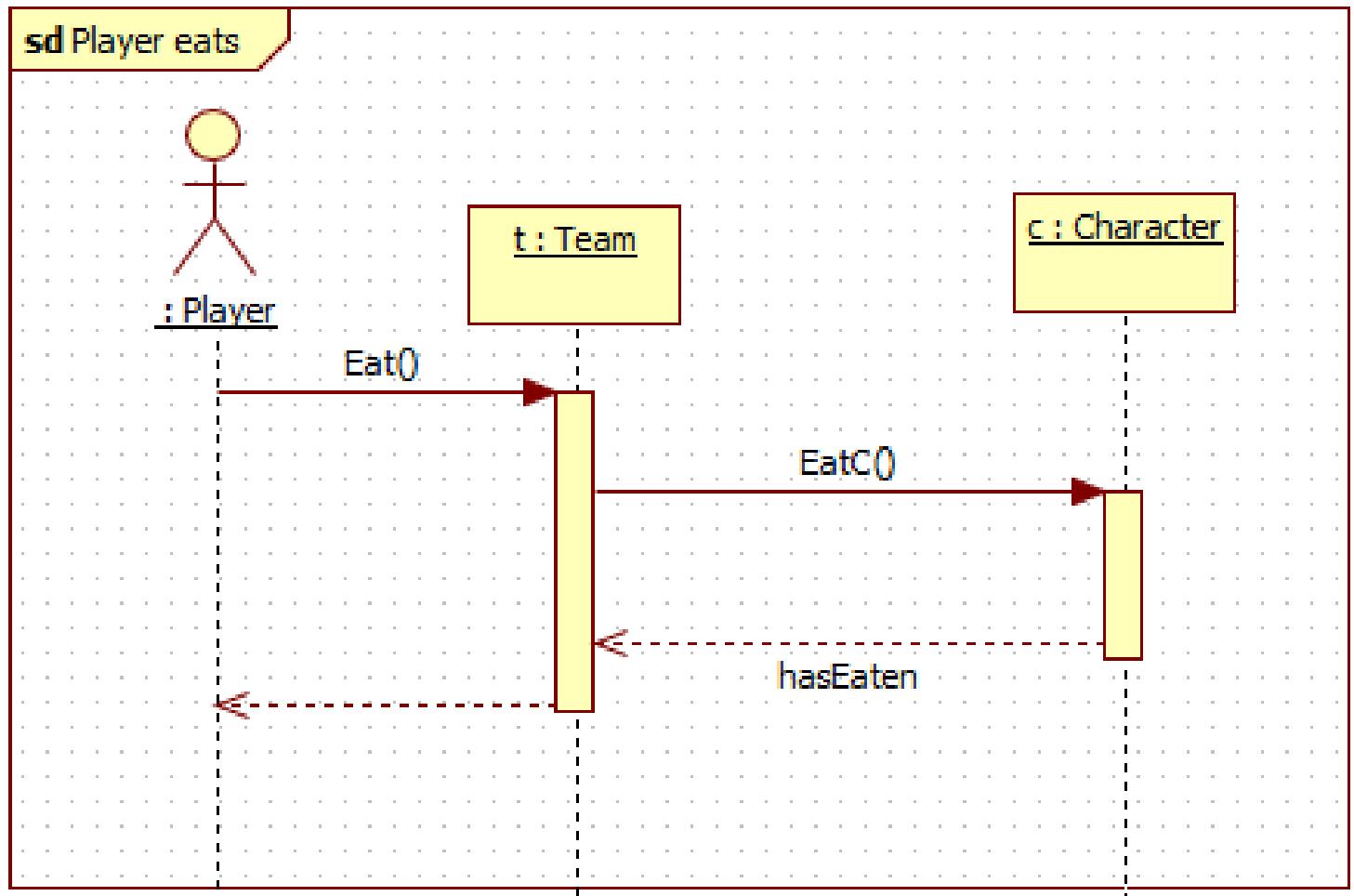


### 3.4.4.6 Player gets an item

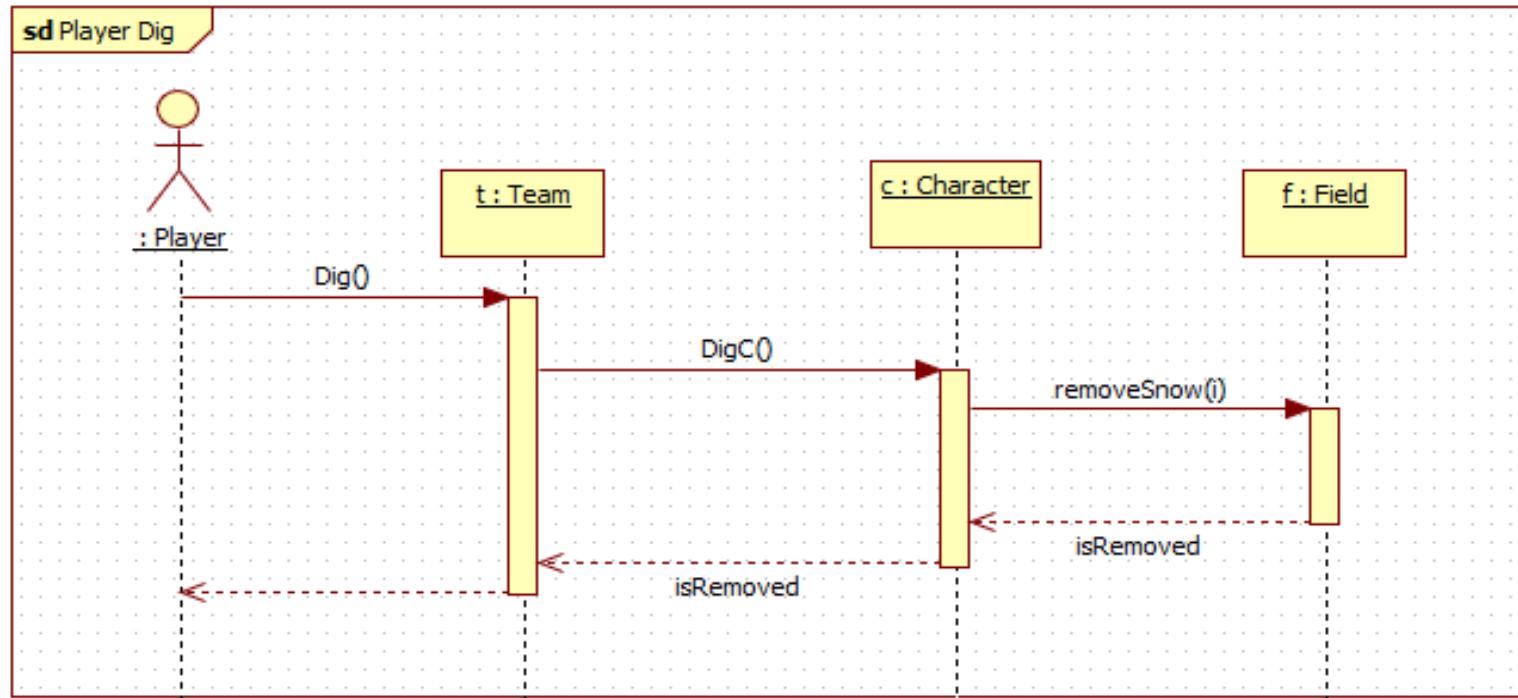


### 3.4.5 Akciók

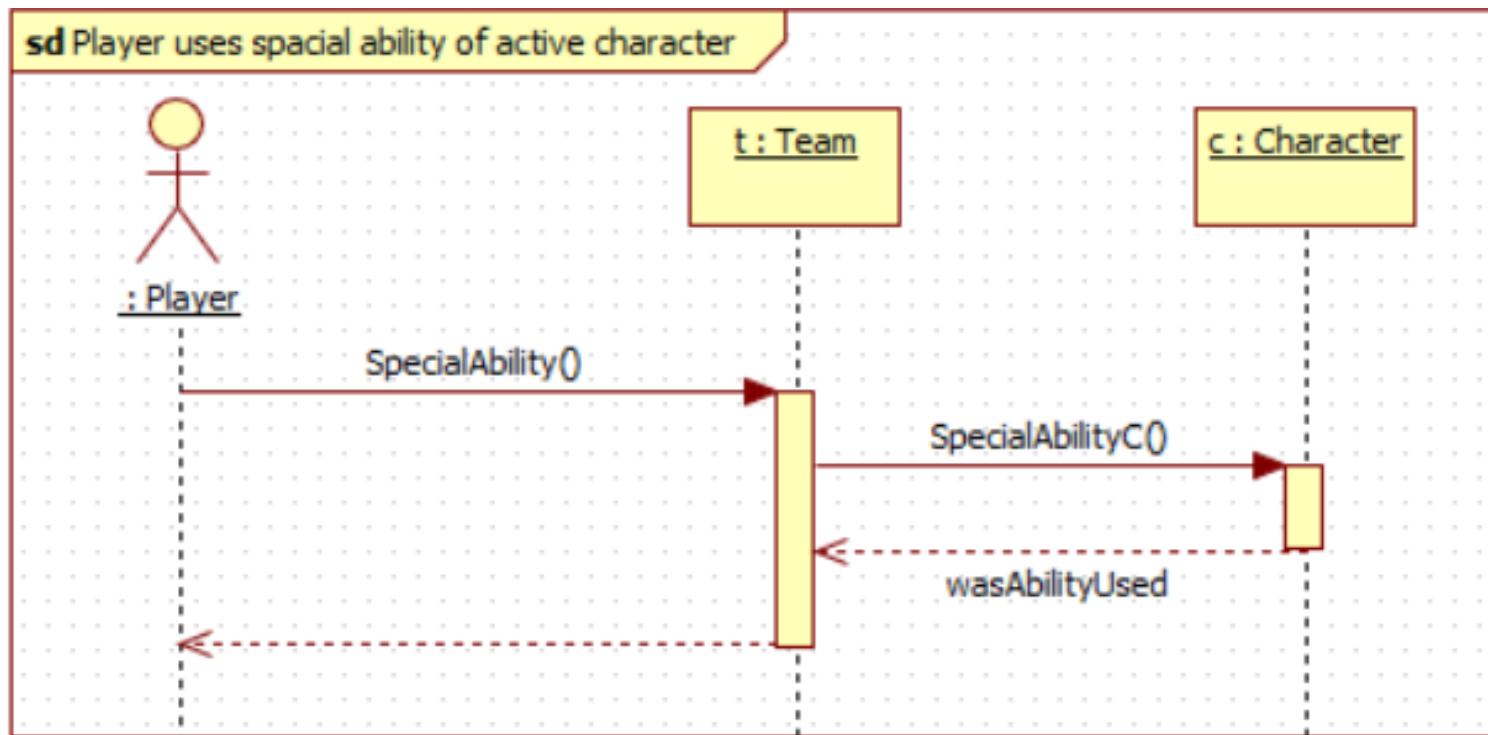
#### 3.4.5.1 Player eats



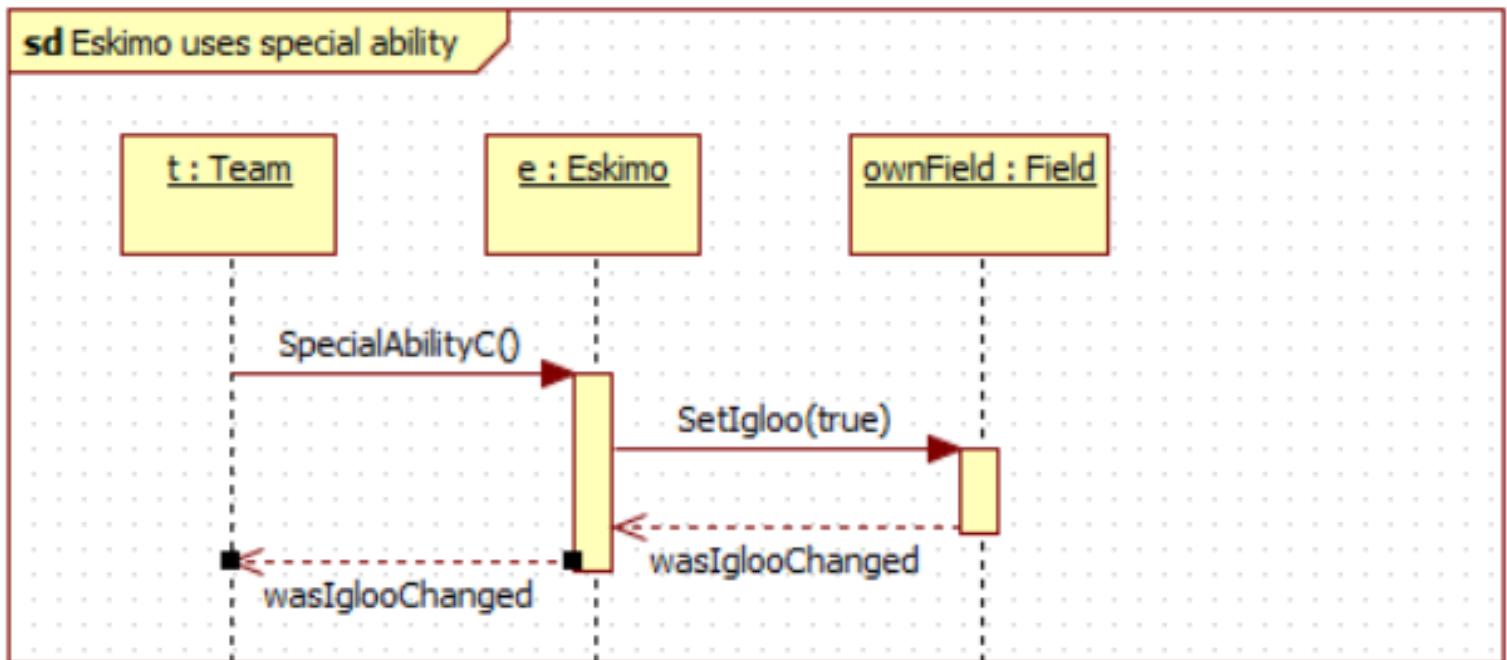
#### 3.4.5.2 Player digs



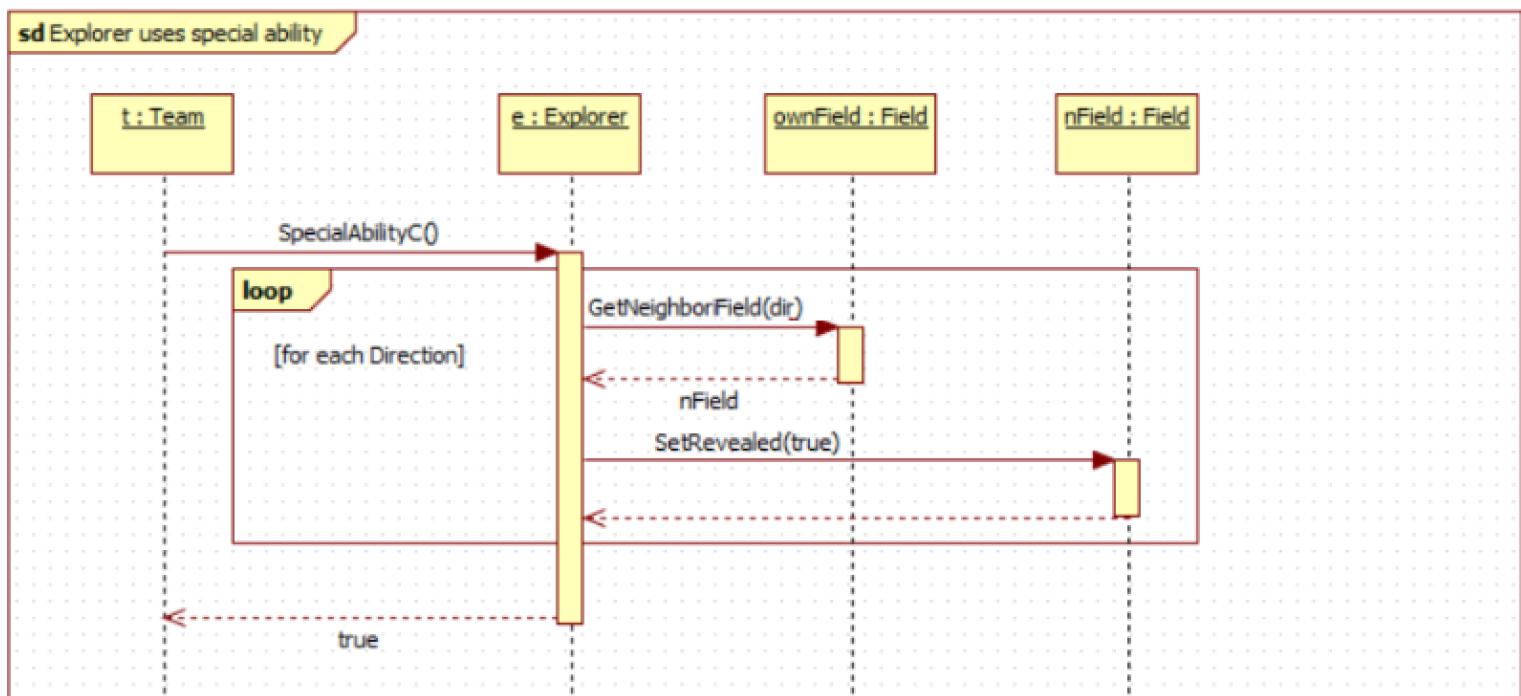
### 3.4.5.3 Player uses special ability of active character



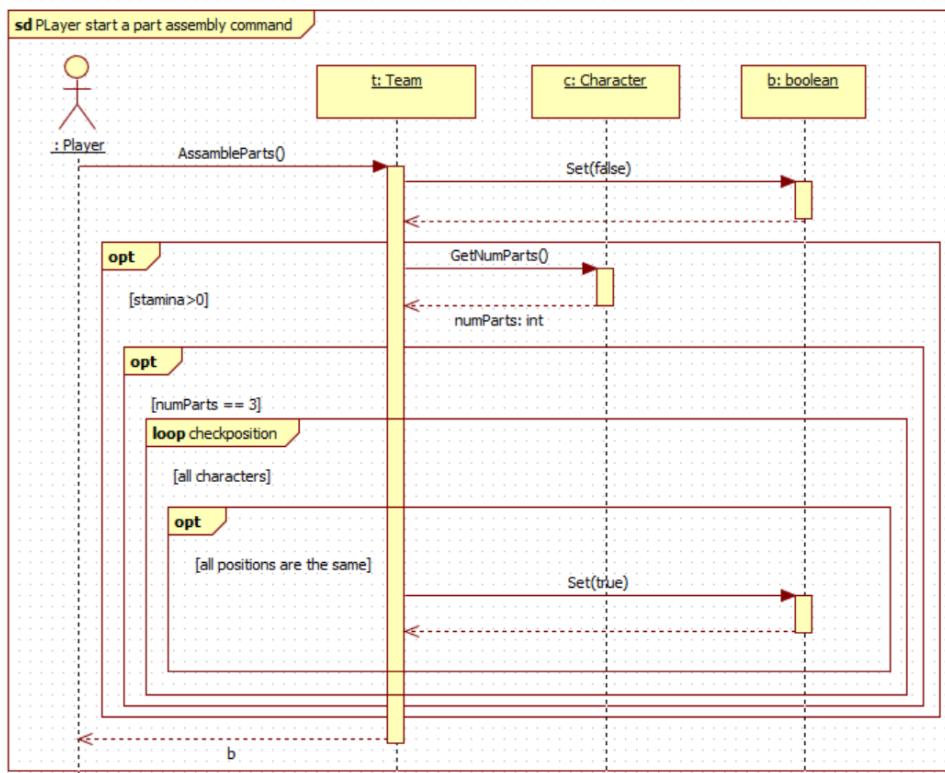
### 3.4.5.4 Eskimo uses special ability



### 3.4.5.5 Explorer uses special ability



### 3.4.5.6 Player starts part assembly command



### 3.3.5Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.02.24. 18:00	2 óra	Háromi Holczer Kedves Kurdi Marton	Értekezlet. Döntés: Háromi megszerkeszti az osztálydiagramot 2020.02.25. estig.
2020.02.25. 17:00	2 óra	Háromi	Háromi elkészíti az osztálydiagramot.
2020.02.27 20:00	1 óra	Kedves Háromi Holczer Marton	Értekezlet. Döntés: Holczer készíti el a Character osztályhoz tartozó elemeket. Marton készíti el a Fielddel kapcsolatos elemeket. Háromi készíti az osztálydiagramot és néhány tárgy leírását. Kedves készíti a Mappel kapcsolatos elemeket. Kurdi készíti el a különleges képességek használatához tartozó szekvenciákat, és a lyukba esés szekvenciát.
2020.02.27 21:00	3 óra	Holczer	Holczer osztály leírásokat készít (Character, Eskimo, Explorer) objektum leírások (eszkimó, kutató) szekvencia diagramokat készít(eat, dig)
2020.02.27 21:00	3 óra	Marton	Marton osztálydiagramm javításokat tesz, osztályleírásokat (Field, Hole, Stable, Unstable, Direction) és szekvenciadiagramm okat készít hozzájuk.

2020.02.27. 21:00	3 óra	Háromi	Háromi elkészíti a Food, Rope, ScubaSuit, Shovel osztályleírásokat és a hozzá tartozó szekvenciadiagramokat.
2020.02.28. 11:00	3 óra	Kurdi	A megbeszélt szekvenciák elkészítése
2020.02.28. 13:00	2 óra	Háromi	Háromi javítja a usecase diagramot és rendezi a szótárat, hozzáadja a fedőlapot, kiegészíti az osztálydiagramot, és megszerkeszti a Shovel szekvenciadiagramot .
2020.02.28. 20:00	2 óra	Háromi	Háromi javítja a Food, Rope, ScubaSuit, Shovel szekvenciadiagramokat.
2020.02.28. 15:00	2 óra	Kedves	Kedves elkészíti a Team, Map, osztályleírását, a BlizzardM, BéizzardF, EndTurn, Move és Move C szekvencia diagramjait,
2020.02.29. 12:00	2 óra	Háromi Holczer Kedves Kurdi Marton	Értekezlet. Döntés: Kedves, Marton, Kurdi, Holczer kijavítják a szekvenciadiagramjaikat
2020.02.29. 14:00	1 óra	Holczer	Javítások character osztályon metódusok hozzáadása és szekvencia diagrammok javítása
2020.02.29. 14:00	1 óra	Háromi	Háromi véglegesíti az osztálydiagramot.
2020.02.29. 15:00	2 óra	Kedves	Kedves javítja a megbeszélés alapján

			a szekvencia diagramokat és elkészíti az inicializálási diagramok elkészítése
2020.02.29 17:00	1 óra	Marton	Marton javításokat végez a korábban elkészített osztályain és szekvenciadiagramm jain
2020.03.01.15:00	2 óra	Háromi Holczer Kedves Kurdi Marton	Értekezlet. Döntés: Kurdi letisztázza a dokumentumot
2020.03.01. 17:00	2 óra	Kurdi	Dokumentum letisztázása

# **Analízis modell kidolgozása 2.**

## **45 – nullpointer**

Konzulens:  
**Ludmány Balázs**

### **Csapattagok**

Háromi Bence	AK51GI	bence.haromi@gmail.com
Holczer Zsolt	XL9Y0E	zsolti@holczer.net
Kedves Áron Csanád	AV43UW	aron.kedves@gmail.com
Kurdi Barnabás	F0DJNW	barnabas.kurdi@gmail.com
Marton Gergely	FC2WNS	martongergely11@gmail.com

## **3.1 Objektum katalógus**

### **3.1.1 Búvárruha**

Vízbeeséskor, ha a karakter visel búvárruhát, akkor nem fullad meg, ki tud úszni egy másik táblára.

### **3.1.2 Eszkimó**

A játékos által választható egyik karakter, amelynek a speciális tulajdonsága, hogy 5 élettel kezd, valamint iglut tud építeni. Az igluban bárki és bárhányan tartózkodhatnak és ebben tartózkodók átvészhetik a hóvihart hőveszteség nélkül.

### **3.1.3 Étel**

A karakter kiáshatja, illetve utána bármikor felhasználhatja 1 hőenergia visszatöltésére.

### **3.1.4 Jégtábla**

A játék jégtáblából áll, amelyek lehetnek stabilak, illetve instabilak is. A stabil jégtáblákon akármennyi játékos állhat, az instabilokon azonban csak meghatározott számú játékos tartózkodhat. Ugyanis, ha az adott instabil mezőn túl sok játékos tartózkodik, akkor az a feje tetejére fordul és mindenkit maga alá temet.

### **3.1.5 Kötél**

Ha egy karakter vízbe esik, nincs búvárruhája, és egy szomszédos táblán lévő barátjának van kötele, akkor az magához húzza, ezzel megmentve a megfulladástól.

### **3.1.6 Lapát**

A játékban az egyik tárgy. Ezt a tárgyat, ha birtokolja a karakterünk akkor 2 egység havat tud eltakarítani 1 egység munkával a kezdeti 1 helyett.

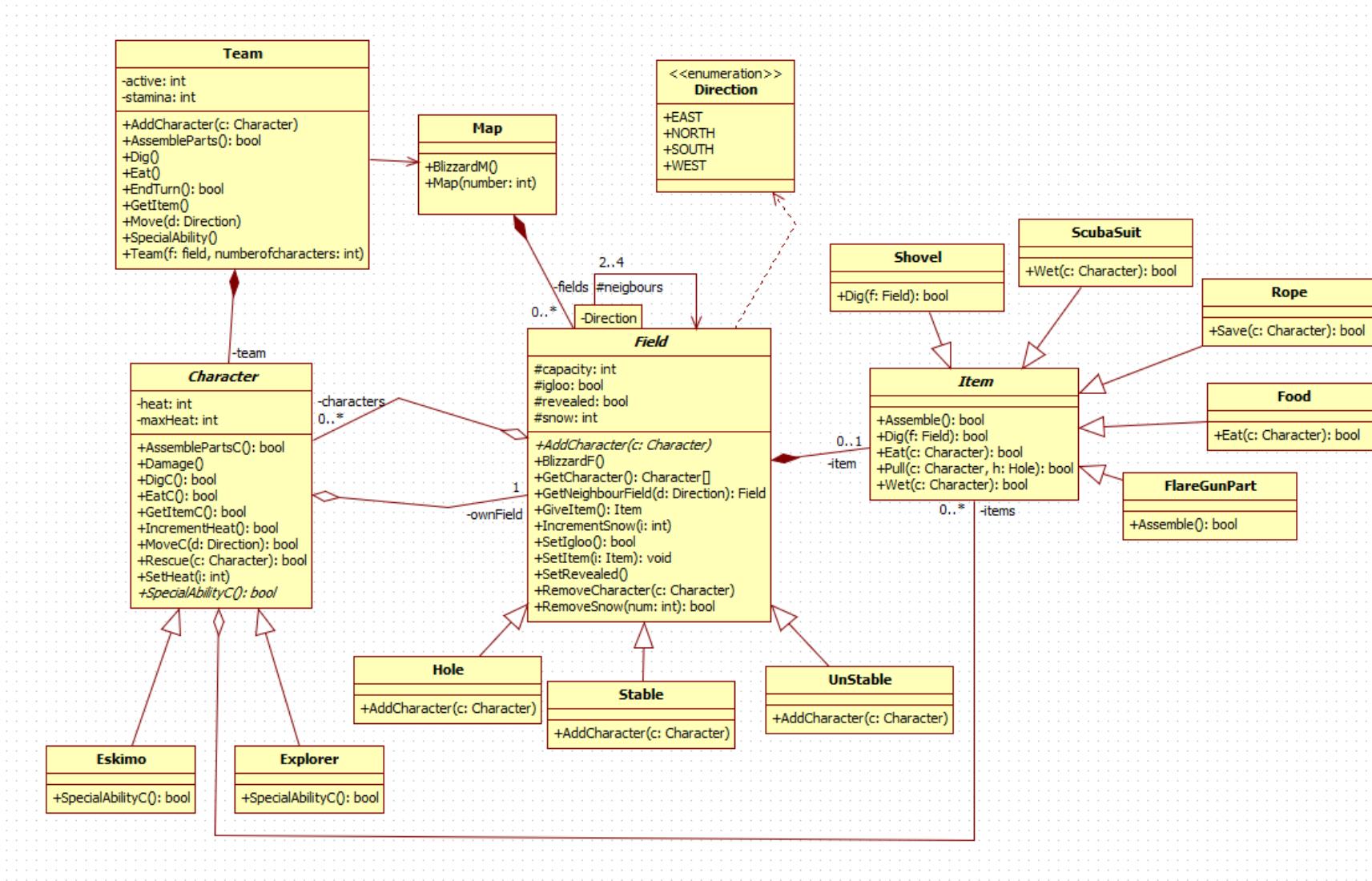
### **3.1.7 Pisztolyalkatrészek**

A játék során gyűjthető tárgyak egy fajtája, amely három elemből áll. A végső cél ezen 3 elem megtalálása, majd az összes játékos egy mezőre állítása.

### **3.1.8 Sarkkutató**

A játékos által egyik választható karakter, aminek a speciális képessége, hogy a mellette lévő mezők terhelhetőségét tudja megnézni. 4 hőenergiával kezd.

### **3.2 Statikus struktúra diagramok**



## 3.3 Osztályok leírása

### 3.3.1 Character

- **Felelősség**

A karakter képességeit mozgását kezeli, valamint a tárgyakkal lévő viselkedést.

- **Attribútumok**

- **-heat: int:** A karakter életét tárolja
- **-maxHeat: int:** A karakter maximális hőenergiáját tárolja azért, hogy ne lehessen túl lépni és a kétféle karakternek másféle a maximuma
- **-item: Item:** A karakternél lévő eszközöket tárolja

- **Metódusok**

- **bool assamblePartsC():** A karakter megpróbálja összeszerelni jelzőrakétát ha sikerül true-val tér vissza ha nem akkor false-al.
- **void Damage():** A karakter életét (heat) csökkenti pl hóvihar esetén
- **bool DigC():** A mezőn hó eltakarítása (1 vagy 2 lapáttal) ha tudott ásni akkor tér vissza true-val egyébként false-t add vissza
- **bool EatC():** A karakter enni fog egy ételt és növeli az életét eggel ha tudott enni akkor tér vissza true-val egyébként false-t add vissza
- **bool GetItemC():** Felvesz egy itemet a mezőről ha fel tudta venni az eszközt akkor tér vissza true-val egyébként false-t add vissza
- **void IncrementFood():** A karakternél lévő élelem számát növeli 1-gyel
- **bool MoveC(d: Direction):** A karakter lép d irányba ha tudott lépni akkor tér vissza true-val egyébként false-t add vissza
- **bool Rescue():** A mellett lévő mezőn lyukba esetttársát ha van nála kötél kimenti magához húzza ha sikerül akkor true-val tér vissza
- **void SetHeat(i: int):** A karakter életét lehet beállítani a megadott "i" értékre
- **void SetRope(b: bool):** Beállítja hogy a karakternél van-e kötél
- **bool SpecialAbilityC():** abstract A karakterhez tartozó speciális képességet hajtja végre amennyiben sikerül true-val ha nem akkor false-al tér vissza

### 3.3.2 Direction

- **Felelősség**

Enumeráció osztály az irányok felsorolására.

- **Attribútumok**

- +NORTH
- +SOUTH
- +EAST
- +WEST

### 3.3.3 Eskimo

- **Felelősség**

Az eszkimóra jellemző dolgokat kezeli

- **Ősosztályok**

Character -> Eskimo

- **Metódusok**

- **bool SpecialAbilityC()**: Az eszkimó speciális képeségét hajtja végre, ami egy iglu építése.

### 3.3.4 Explorer

- **Felelősség**

A sarkkutatóra jellemző dolgokat kezeli

- **Ősosztályok**

Character -> Explorer

- **Metódusok**

- **bool SpecialAbilityC()**: A kutató speciális képességét hajtja végre amiaz hogy a mellette lévő mezők terhelhetőségét megtekinti.

### 3.3.5 Field

- **Felelősség**

Absztrakt osztály, amely azon mezőket kezeli, amelyek a térképet képezik, ahol a karakterek mozoghatnak, illetve ahol a tárgyak elhelyezkedhetnek.

- **Attribútumok**

- **#Field[Direction]: neighbours**: minden mező ismeri az adott irányban lévő szomszédját.
- **Character[0..\*]: characters**: A mezőn álló játékosok összessége.
- **#capacity: int**: Az adott mező kapacitása.
- **#igloo: bool**: Megadja, hogy az adott mezőn van-e iglu.
- **#revealed: bool**: Megadja, hogy egy Sarkkutató felfedte-e a stabilitását a mezőnek
- **#snow: int**: Az adott mezőn lévő hó aktuális mennyisége.

- **Metódusok**

- **void AddCharacter(Character c):** Absztakt függvény, amely hozzáadja a paraméterként megkapott karaktert a Field-hez, illetve egyéb változásokat tesz, attól függően, hogy mely leszármazottakban valósul meg.
- **BlizzardF():** A mezőt a hóvihar evvel a függvénytel sújtja, különböző módosításokat végez a mezővel, illetve az esetlegesen rajta álló játékos(ok)on.
- **Character[] GetCharacter():** Visszaadja a mezőn álló karaktert, vagy karaktereket.
- **Field GetNeighbourField(Direction d):** Visszaadja a paraméterként kapott irányban tartózkodó mezőt. Amennyiben nincs ott mező, akkor null-lal tér vissza.
- **Item GiveItem():** Visszatér a mezőn lévő tárggyal, majd a mező Item attribútumából eltávolítja azt.
- **IncrementSnow(int howmuch):** Megnöveli a hó mennyiséget a mezőn 1-gyel.
- **bool RemoveSnow(int num):** Havat takarít el a megadott mennyiséget(1 vagy 2). Ha sikerül az eltakarítás true-val ha nem akkor false-al tér vissza.
- **void SetIgloo():** Beállítja a mező igloo paraméterét true-ra.
- **void SetRevealed():** Beállítja a mező revealed attribútumát true-ra, ami azt jelenti, hogy egy Sarkkutató megnézte, hogy az adott mező stabil-e vagy sem.
- **void RemoveCharacter(Character c):** Eltávolítja a paraméterként kapott karaktert a mezőről.

### 3.3.6 FlareGunParts

- **Felelősség**

A jelzőrakéta alkatrészeit reprezentálja. Összesen három szerepel belőlük a játékban. Hogyha mindenhol összegyűjtötték a játékosok, az Assamble függvényel lehet befejezni a játékot. Megjegyzés: azért nincs külön osztály a különböző alkatrészeknek, mivel funkciójukban egyáltalán nem különböznek, a grafikus megjelenítéshez később az összes Item rendelkezni fog egy texture attribútummal, amely alapján a különböző alkatrészeket meg lehet majd különböztetni, de ezt nem foglalja magába az analízis modell.

- **Ősosztályok**

Item → FlareGunParts

- **Metódusok**

- **Assamble(Character c): bool:** A játékosnak egy true értékkal tér vissza és ezzel jelzi, hogy van FlareGunPartja a játékosnak. Ezeket a true visszatéréseket számolja meg a rendszer.

### 3.3.7 Food

- **Felelősség**

Az étel tárgyat reprezentálja. A játékos testhőjének növelésére tudja használni.

- **Ősosztályok**

Item -> Food

- **Metódusok**

- **bool Eat(Character c):** meghívja a paraméterként kapott Character IncrementHeat() függvényét, ami megnöveli eggyel a hőenergia attribútumának értékét, true-val tér vissza, ha sikerül hőenergiát növelni ha nem akkor false-szal.

### 3.3.8 Hole

- **Felelősség**

Lyukakat reprezentáló osztály. Ezen mezők stabilitása 0.

- **Ősosztályok**

Field->Hole

- **Attribútumok**

Az ősosztály attribútumai.

- **Metódusok**

- **void AddCharacter(Character c):** Hozzáadja a játékosot a lyukhoz, illetve a szomszédos mezőkön álló játékosoknak (ha vannak) meghívja a Rescue() függvényét.

### 3.3.9 Item

- **Felelősség**

A tárgyak ősosztálya, benne az összes függvény megvan, ami a tárgyakban, azonban mindegyik törzse üres. Csak abban az osztályban van megvalósítva az összes függvény, ahol releváns.

- **Metódusok**

- **bool Assmeble():** a tárgyak azon függvénye, amely a három pisztolyalkatrész “összeszerelését” idézi elő. A visszatérés logikai tartalma attól függ, hogy az adott akció sikeres volt-e vagy sem. Ez a függvény csak a FlareGunParts osztályban van megvalósítva.
- **bool Dig(Field: f):** a tárgyak azon függvénye, amely segítségével a paraméterként kapott mezőn áshat az adott tárggyal (2 egységnyi havat). A visszatérés logikai tartalma attól függ, hogy az adott akció sikeres volt-e vagy sem. Ez a függvény csak a Shovel osztályban van megvalósítva.
- **bool Eat(Character c):** a tárgyak azon függvénye, amely segítségével a paraméterként kapott karakter megeheti az adott tárgyat. A visszatérés logikai tartalma attól függ, hogy az adott akció sikeres volt-e vagy sem. Ez a függvény csak a Food osztályban van megvalósítva.
- **bool Pull(Character c, Hole h):** a tárgyak azon függvénye, amely segítségével a paraméterként átadott karaktert kihúzzák a paraméterként kapott lyukból. A visszatérés logikai tartalma attól függ, hogy az adott akció sikeres volt-e vagy sem. Ez a függvény csak a Rope osztályban van megvalósítva.
- **bool Wet(Character c):** a tárgyak azon függvénye, amely segítségével a paraméterként kapott karakter amikor vízbe esik megmenekülhet. A visszatérés logikai tartalma attól függ, hogy az adott akció sikeres volt-e vagy sem. Ez a függvény csak a ScubaSuit osztályban van megvalósítva.

### 3.3.10 Map

- **Felelősség**

Ez az osztály reprezentálja a pályát. Tárolja a pálya mezőit és felelős a hóviharok végrehajtásáért.

- **Attribútumok**

- **-fields : Field:** A játék mezőit tároló dinamikus tömb.

- **Metódusok**

- **void BlizzardM():** ez a metódus dönti el, hogy történik-e hóvihar valamint végrehajtja azt. Ha van hóvihar, akkor a véletlenszerűen kiválaszt az általa tárolt field-ek közül néhányat és meghívja azoknak a BlizzardF() függvényét.

### 3.3.11 Rope

- **Felelősség**

A kötél tárgyat reprezentálja. A játékos ennek segítségével ki tudja húzni egy mellette lévő játékosat a lyukból.

- **Ősosztályok**

Item -> Rope

- **Metódusok**

- **void bool Pull(Character c, Hole h):** függvény, amellyel a paraméterként kapott karakter kihúzását valósítja meg a paraméterként kapott lyukból. A logikai visszatérés értéke attól függ, hogy a művelet sikeres volt-e vagy sem.

### 3.3.12 ScubaSuit

- **Felelősség**

A búvárruha tárgyat reprezentálja. Ha a karakter viseli, akkor nem fullad meg a vízben.

- **Ősosztályok**

Item->Shovel

- **Metódusok**

- **bool Wet(Character c):** visszatér egy true értékkal, ezzel jelzi, hogy a játékosnál van búvárruha.

### 3.3.13 Shovel

- **Felelősség**

Az ásó tárgyat reprezentálja. Ha birtokosnak van ásója, akkor eggyel több egység havat tud eltakarítani.

- **Ősosztályok**

Item -> Shovel

- **Metódusok**

- **bool Dig(Field: f):** az ásó osztály azon függvénye, amely segítségével a paraméterként kapott mezőn áshat 2 egységnyi havat. A visszatérés logikai tartalma attól függ, hogy az adott akció sikeres volt-e vagy sem.

### 3.3.14 Stable

- **Felelősség**

A stabil jégmezőt reprezentálja. Ezen mezők stabilitása -1, ami a végtelent jelenti.

- **Ősosztályok**

Field -> Stable

- **Attribútumok:**

- Az ősosztály attribútumait örökli.

- **Metódusok**

- Az ősosztály metódusait örökli, a AddCharacter() metódust pedig nem írja felül.

### 3.3.15 UnStable

- **Felelősség**

Az instabil jégmezőt reprezentálja. Ezen mezők stabilitása korlátozott, amit, ha meglépne

- **Ősosztályok**

Field->UnStable

- **Attribútumok:**

- Az ősosztály attribútumait örökli.

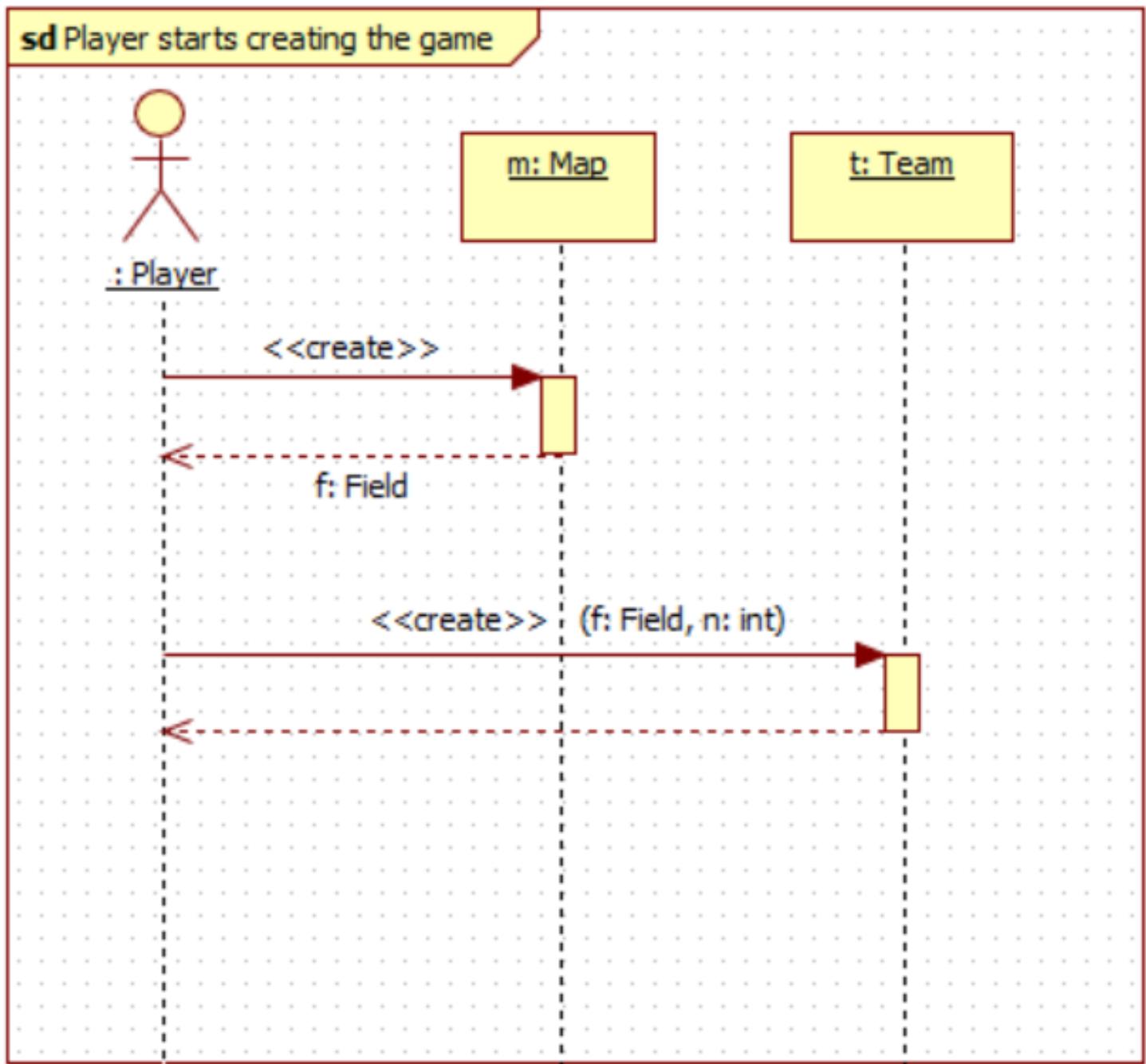
- **Metódusok**

- **void AddCharacter(Character c):** Hozzáadja a paraméterként kapott karaktert a mezőhöz, illetve ellenőrzi, hogy ezzel a mező átlépte-e a kapacitásának határát. Ha igen, akkor a mező a fejére fordul, ezzel minden rajta álló játékos hőenergiáját 0-ra állítja.

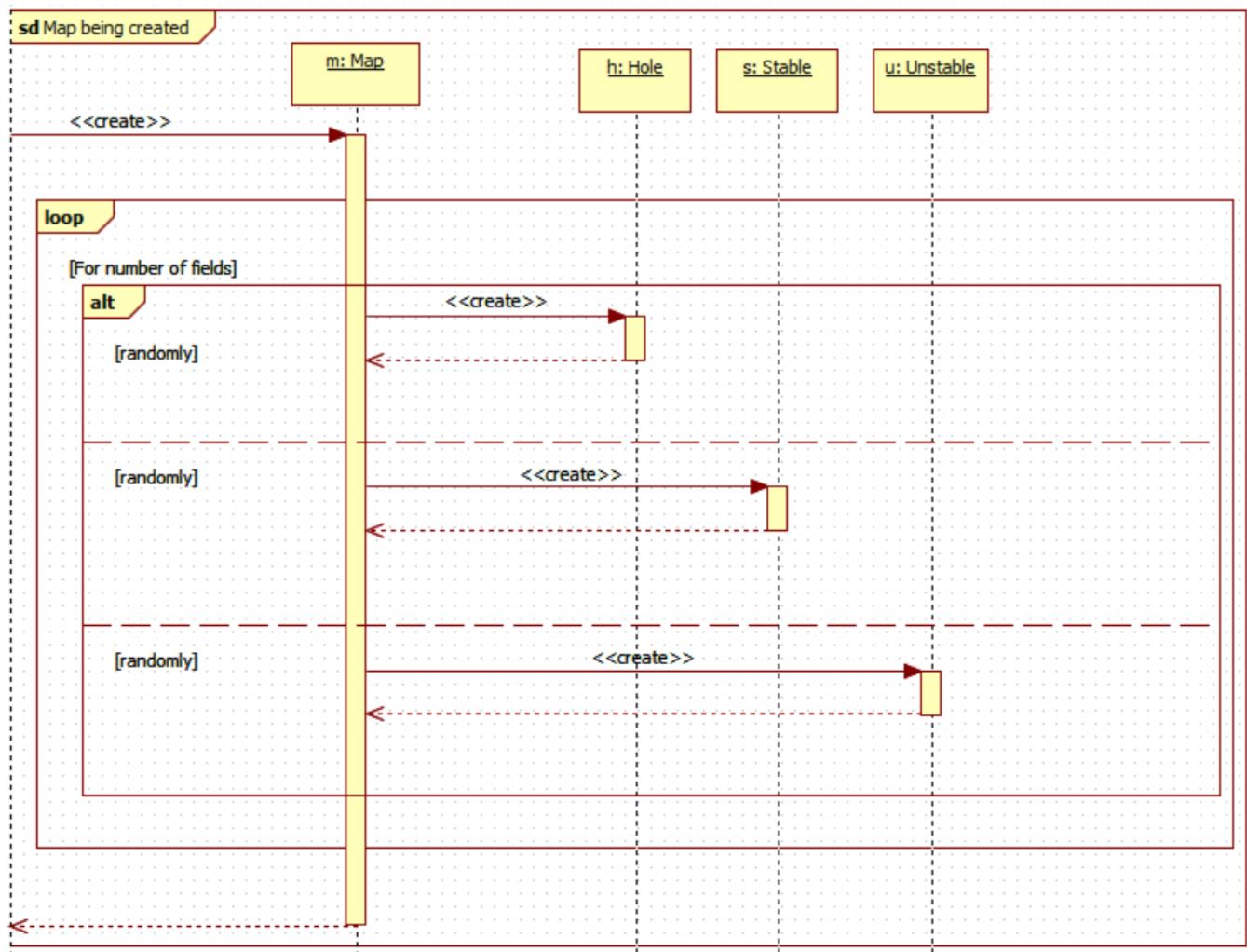
## 3.4 Szekvenciadiagramok

### 3.4.1 Inicializáció

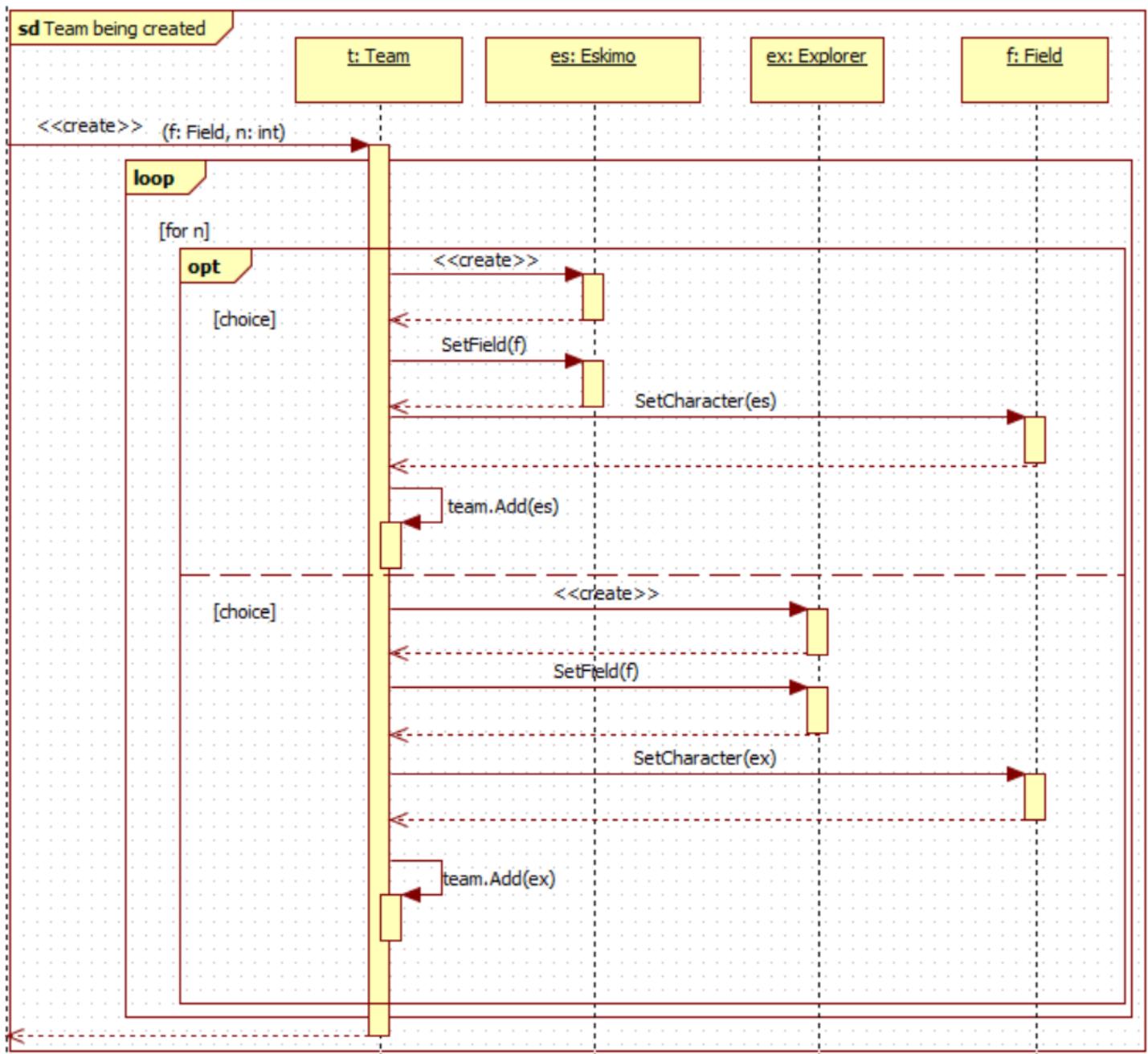
#### 3.4.1.1 Player starts creating the game



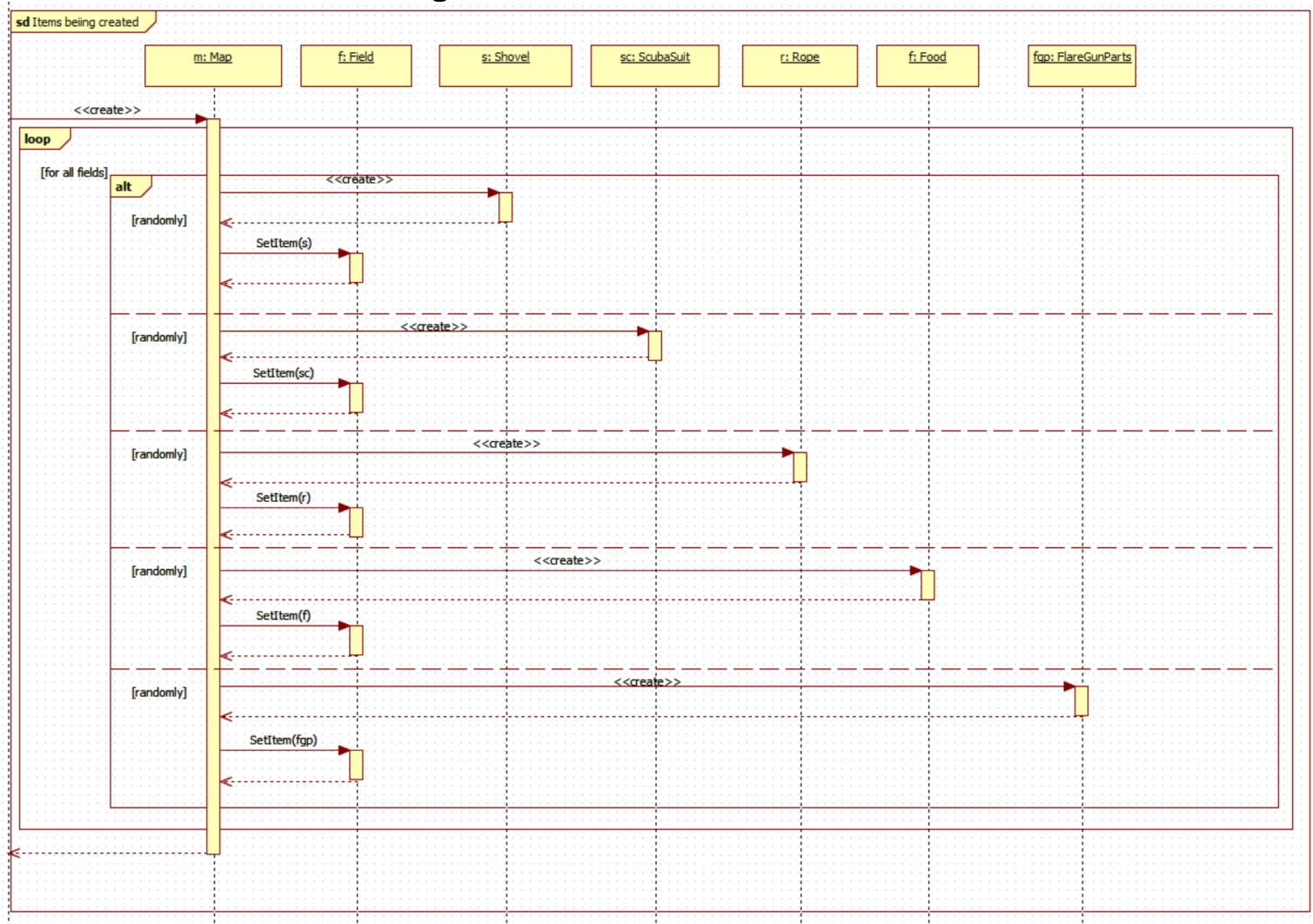
### 3.4.1.2 Map being created



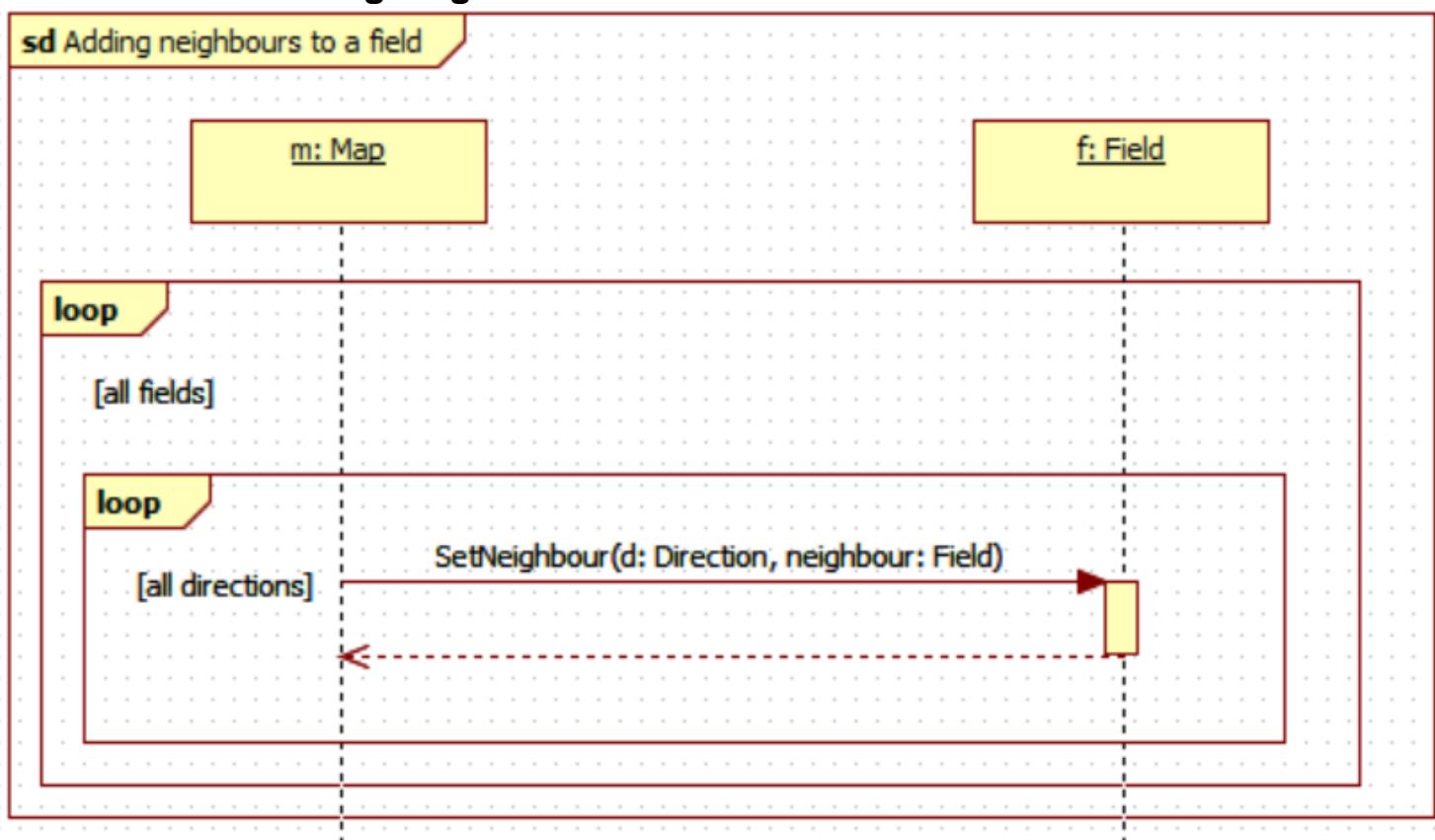
### 3.4.1.3 Team being created



### 3.4.1.4 Items being created

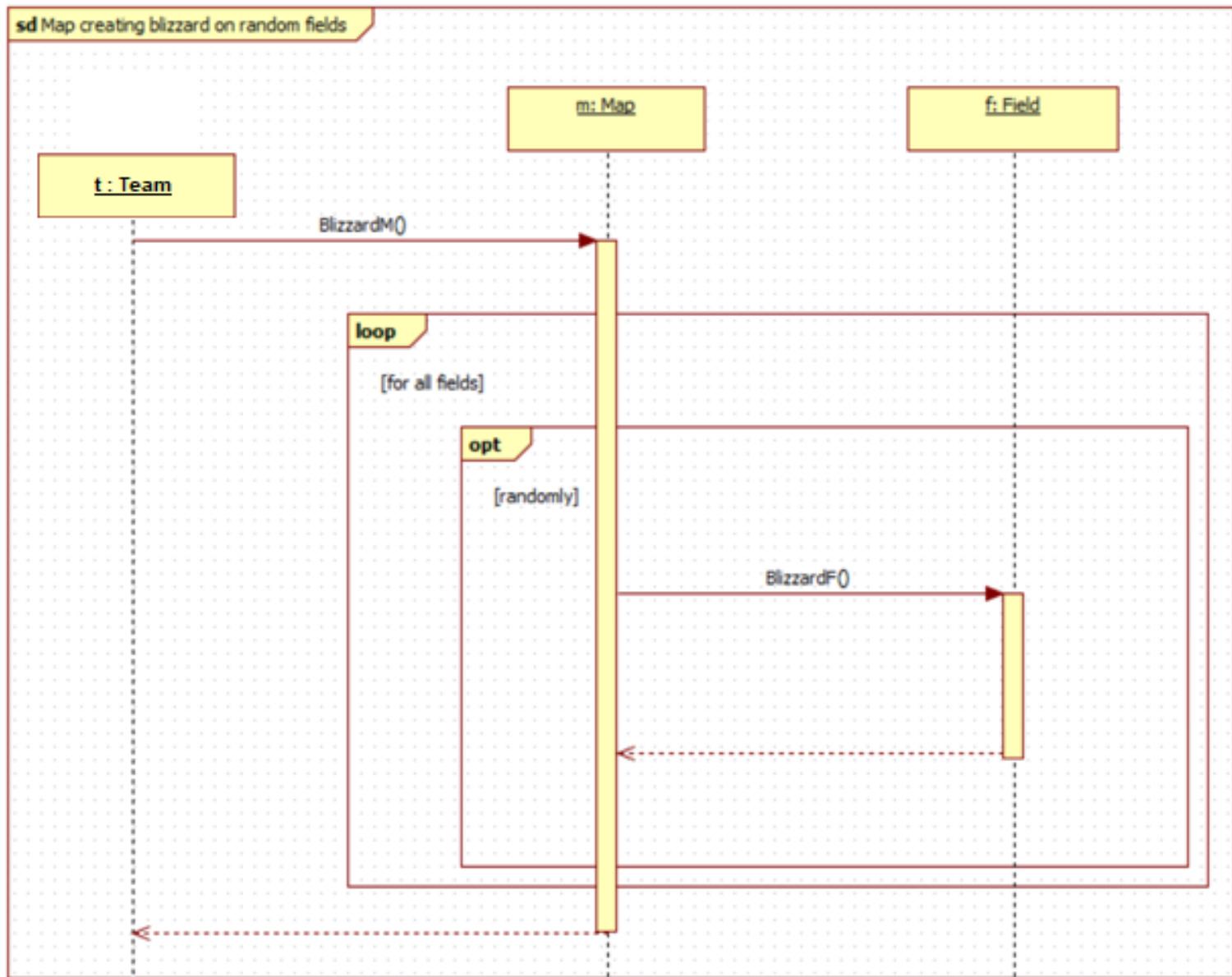


### 3.4.1.5 Adding neighbours to a field

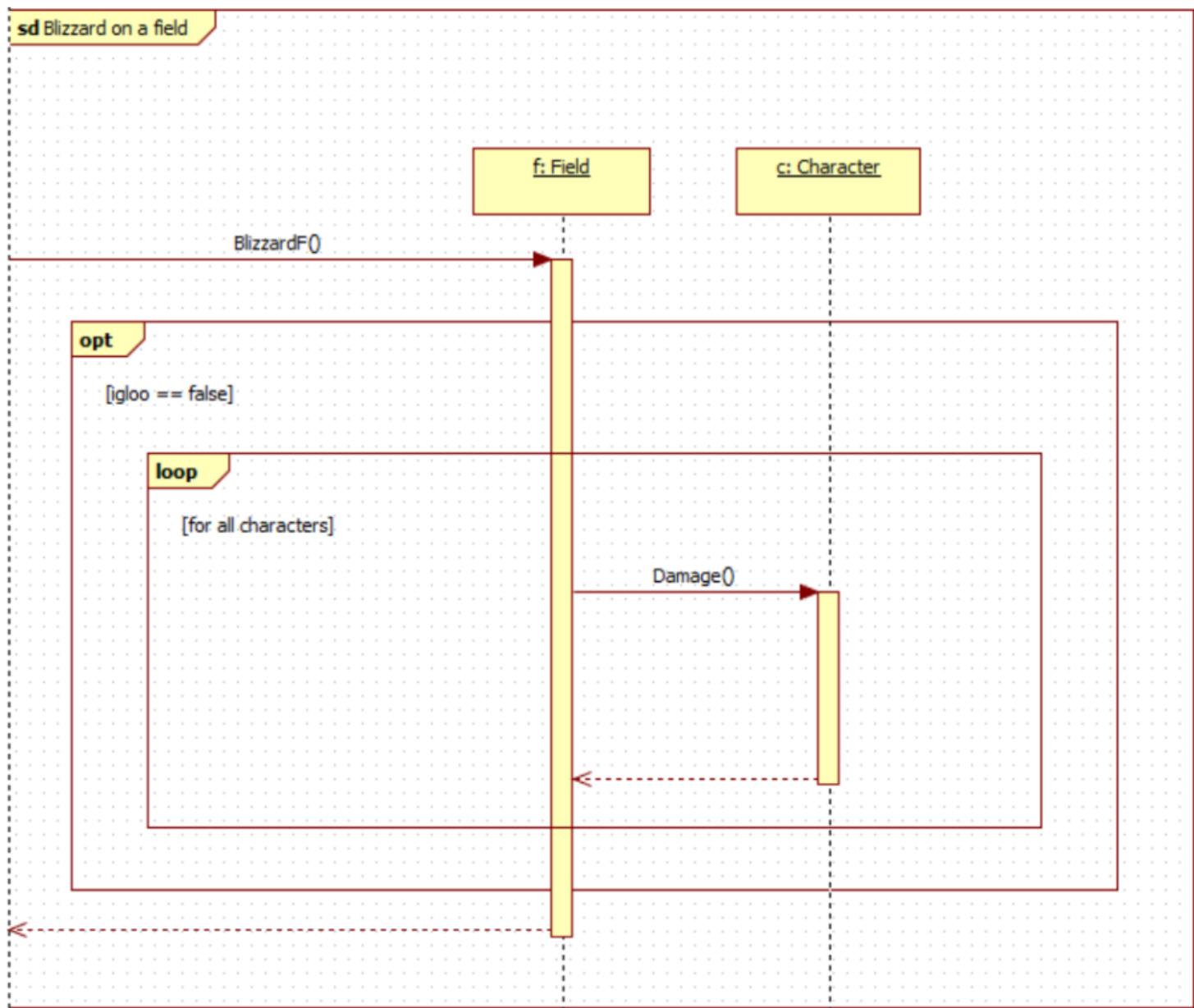


### 3.4.2 Kör befejezése

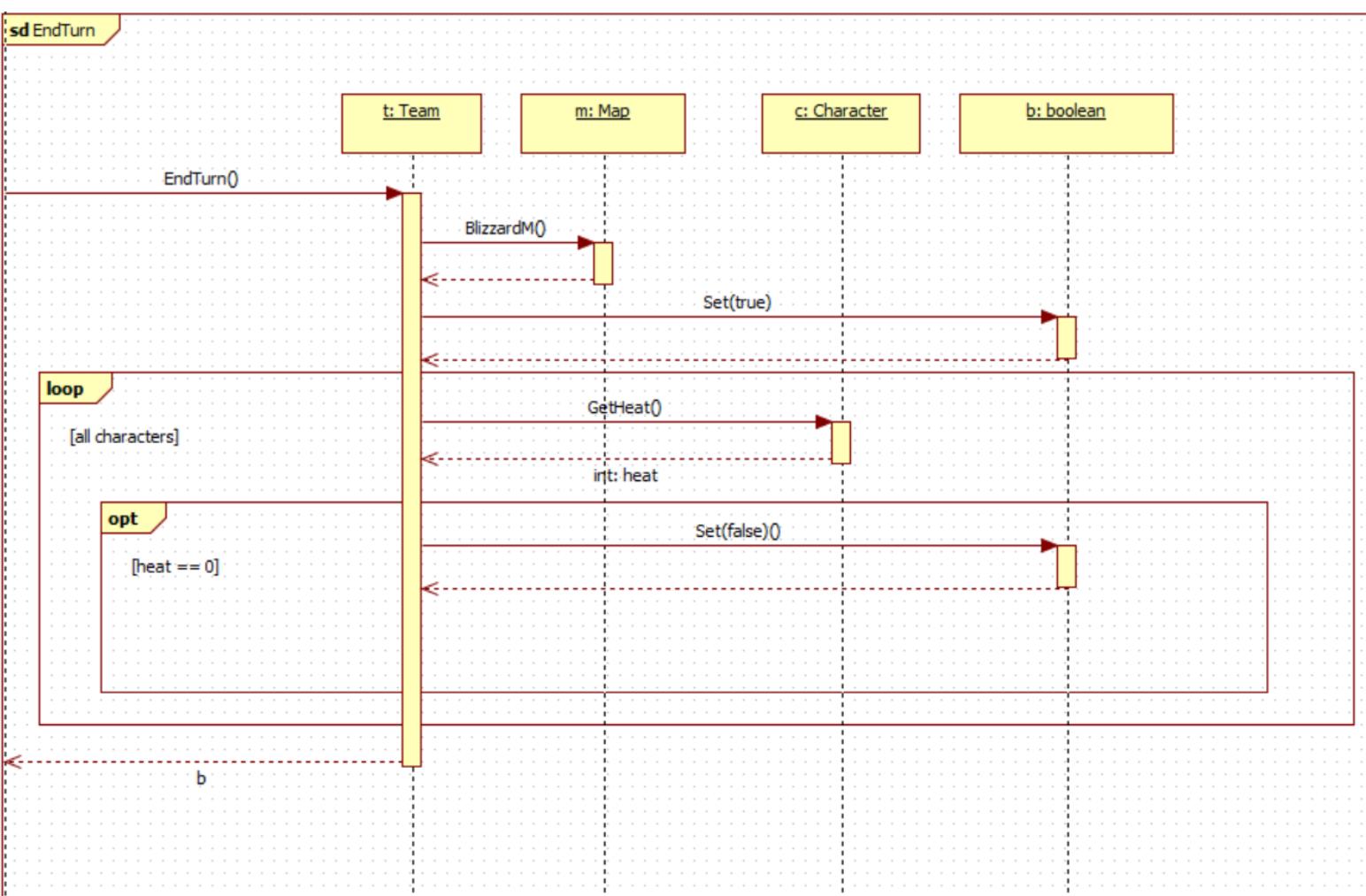
#### 3.4.2.1 Map creating blizzard on random fields



### 3.4.2.2 Blizzard on a field

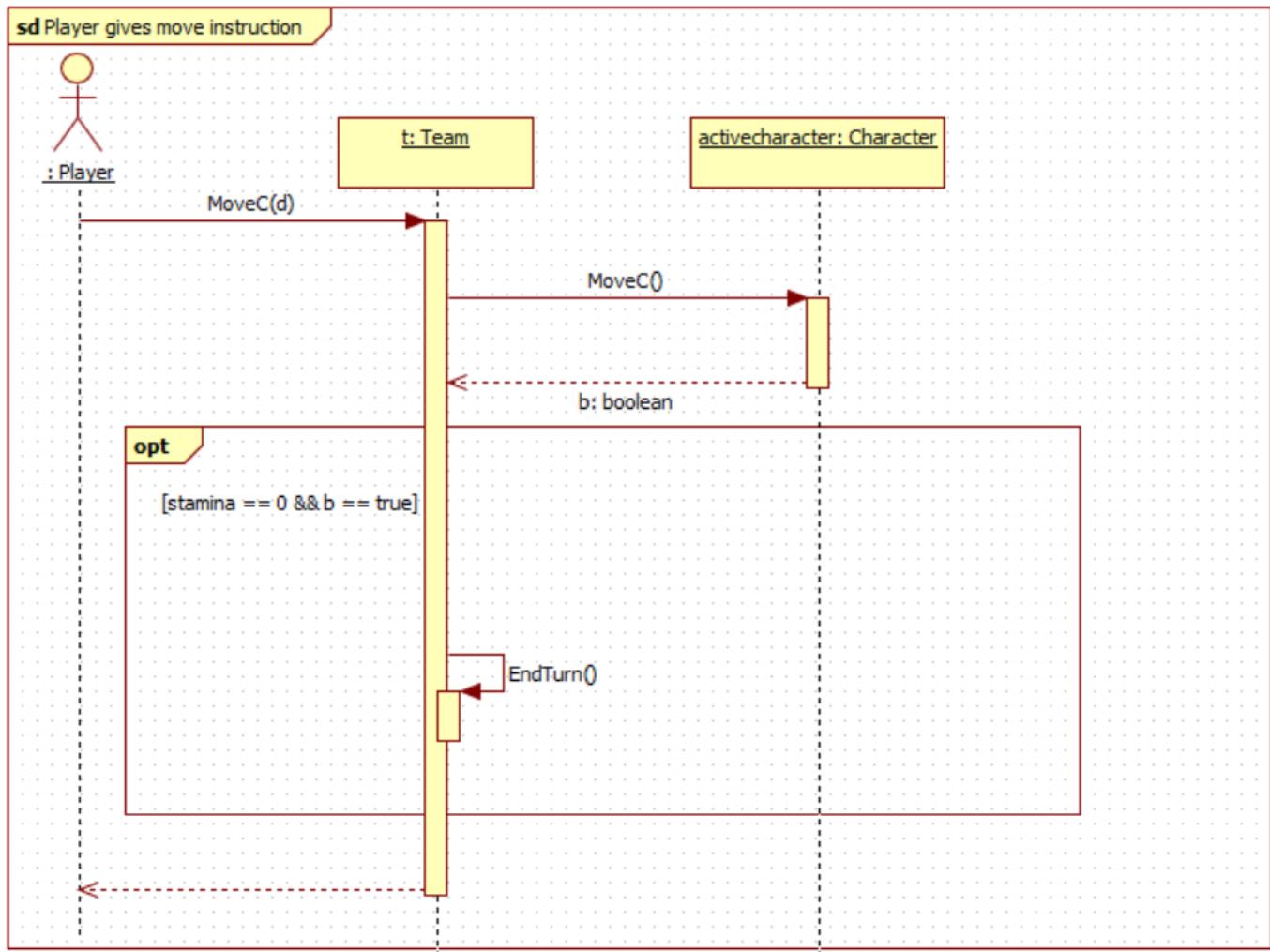


### 3.4.2.3 EndTurn

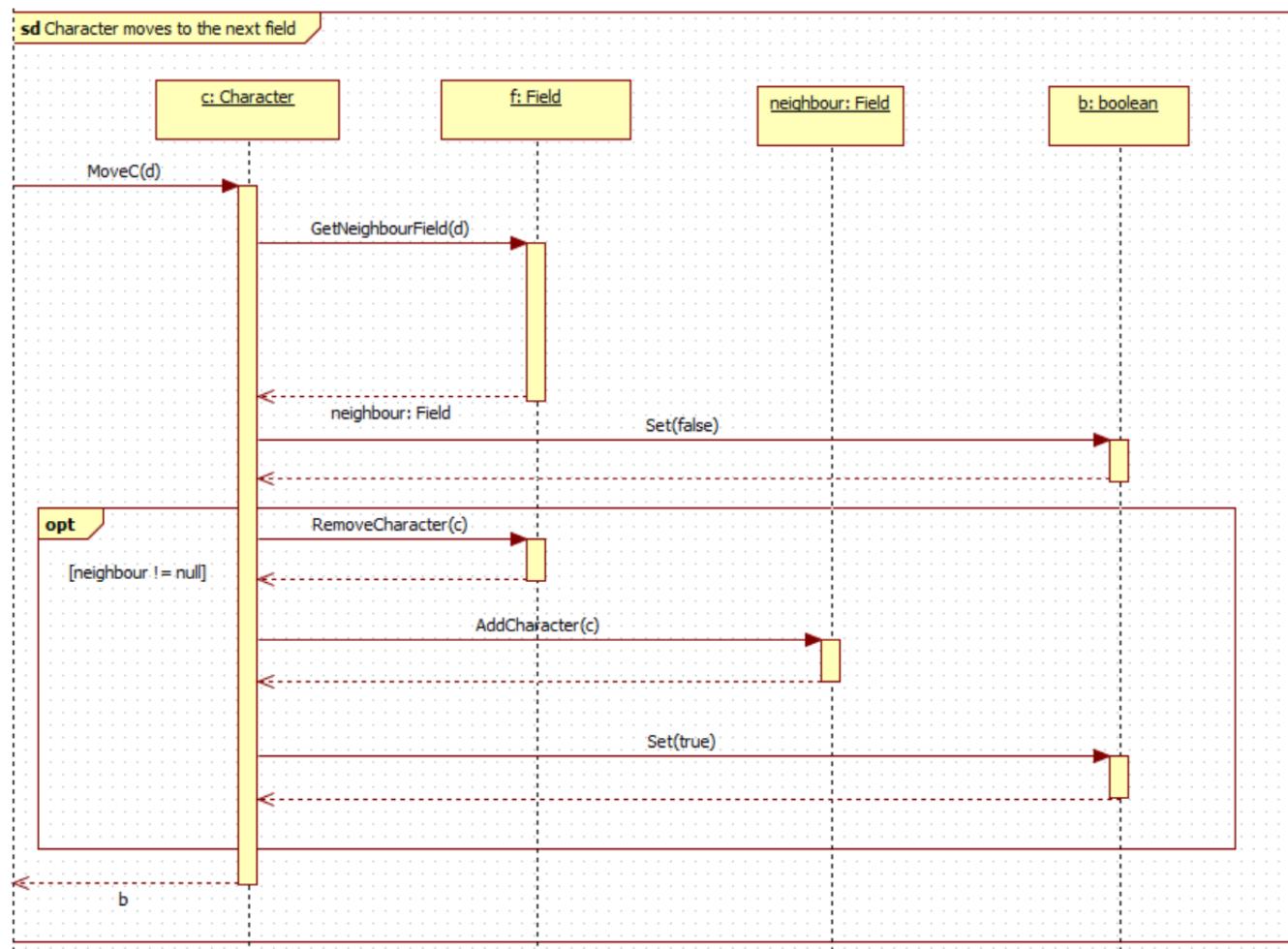


### 3.4.3 Mozgás

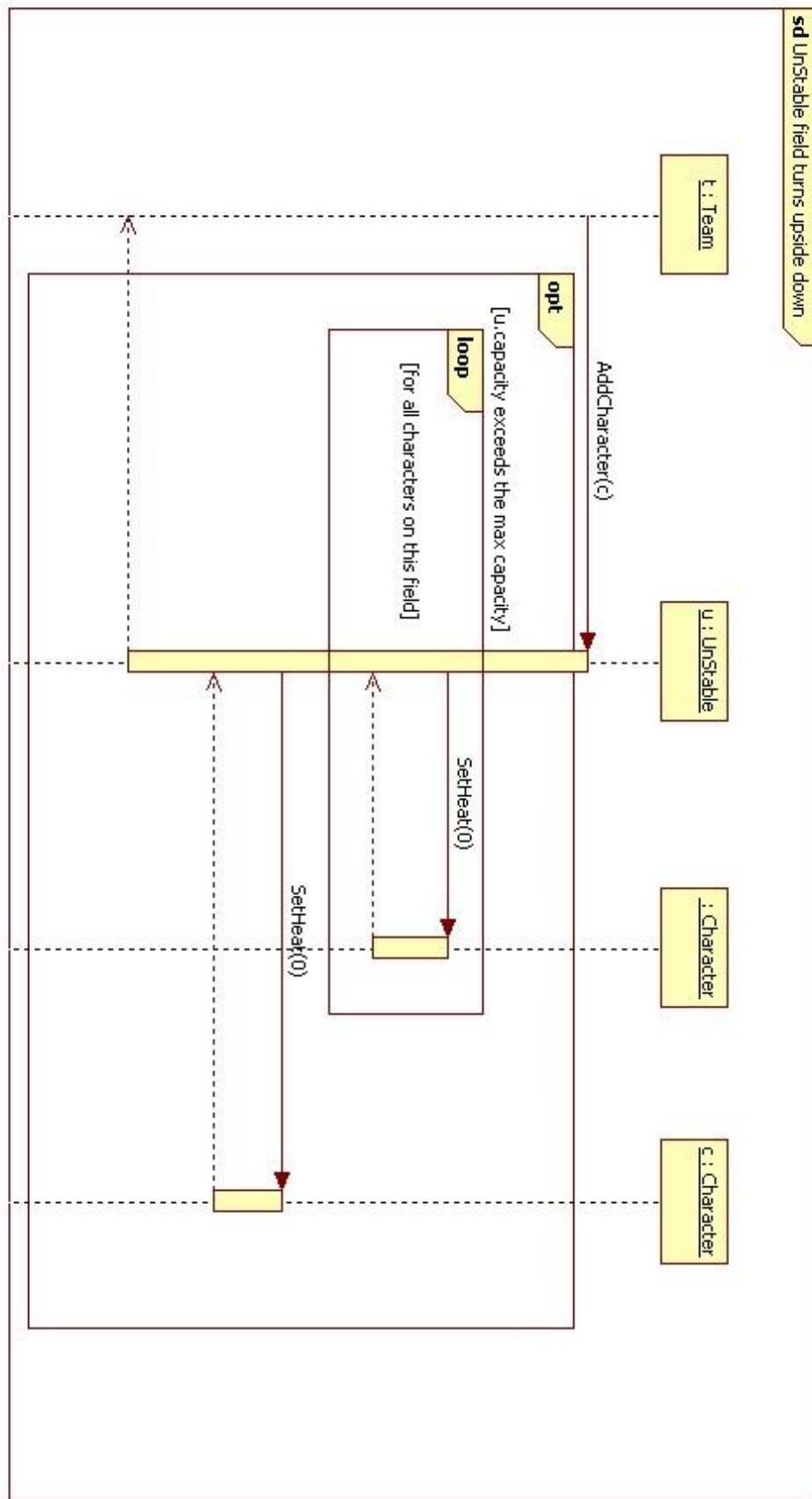
#### 3.4.3.1 Player gives move instruction



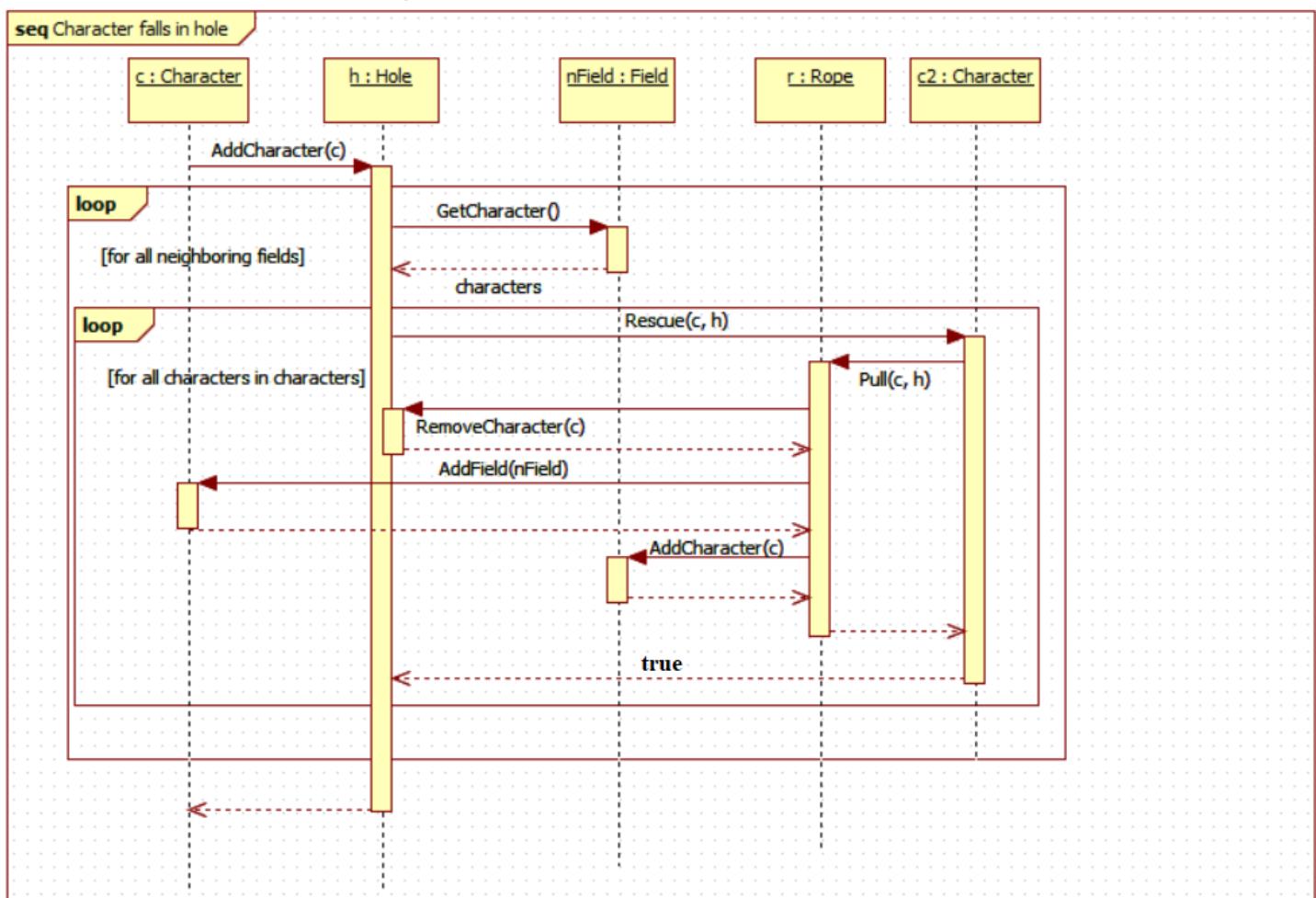
### 3.4.3.2 Character moves to next field



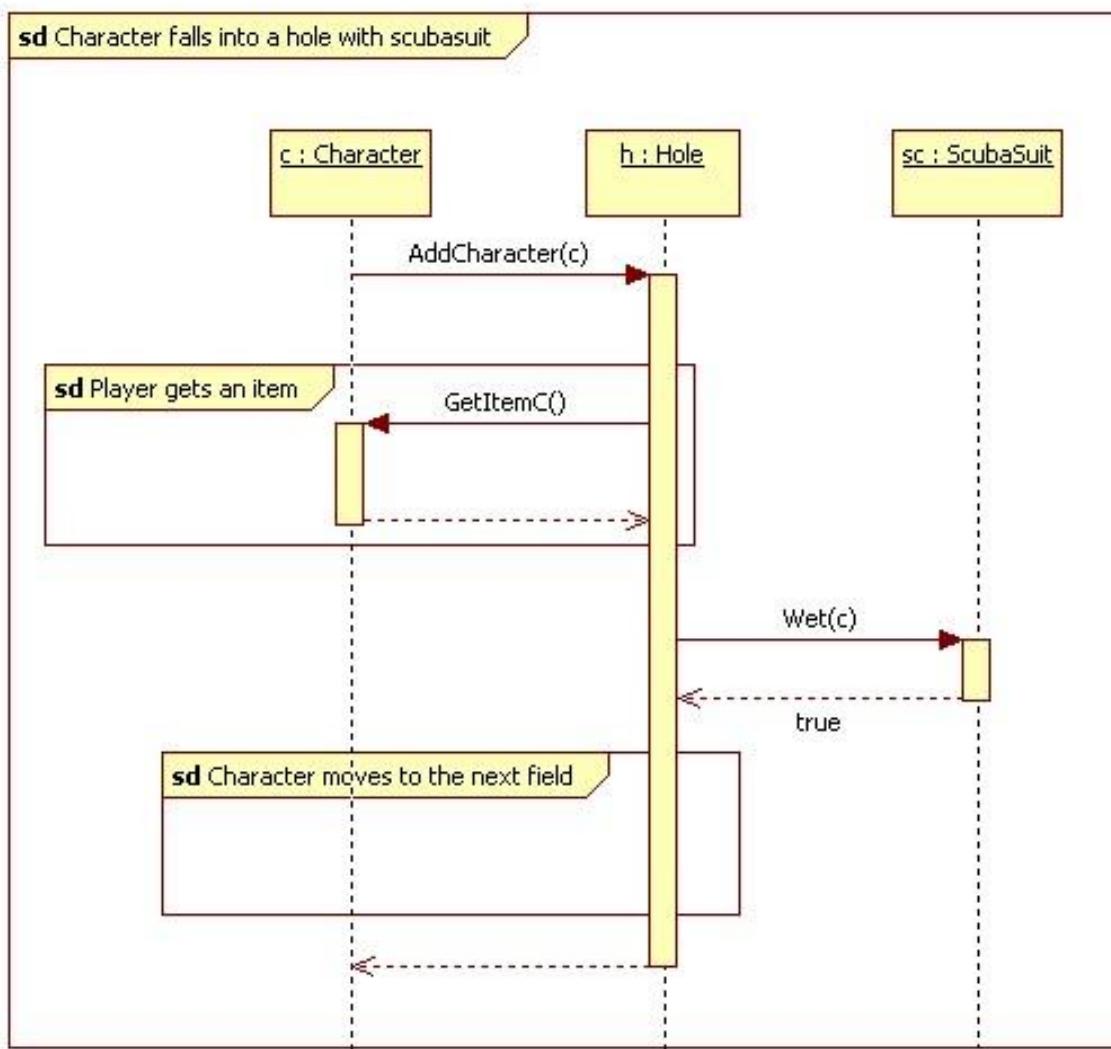
### 3.4.3.3 Unstable field turns upside down



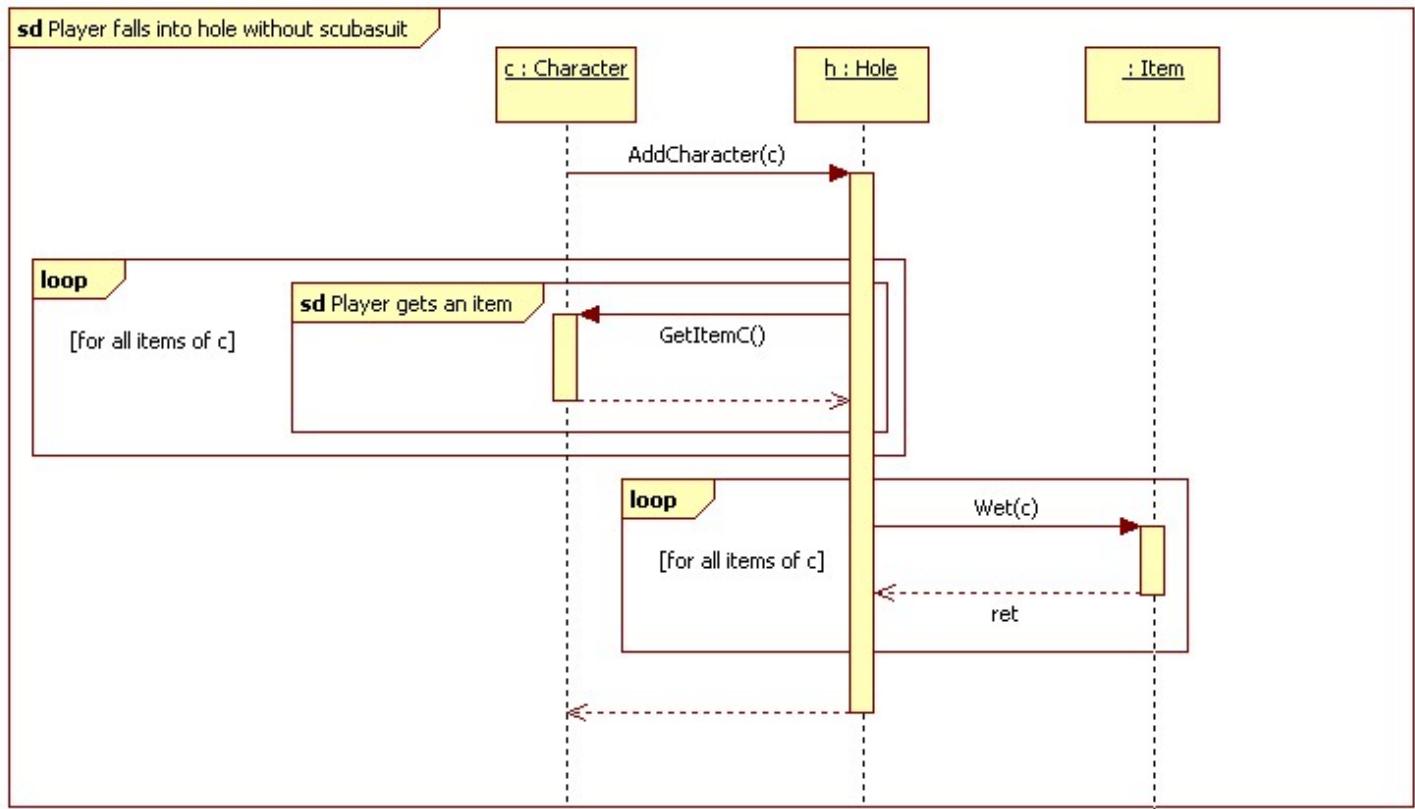
### 3.4.3.4 Character gets rescued



### 3.4.3.5 Character falls in hole with scuba suit

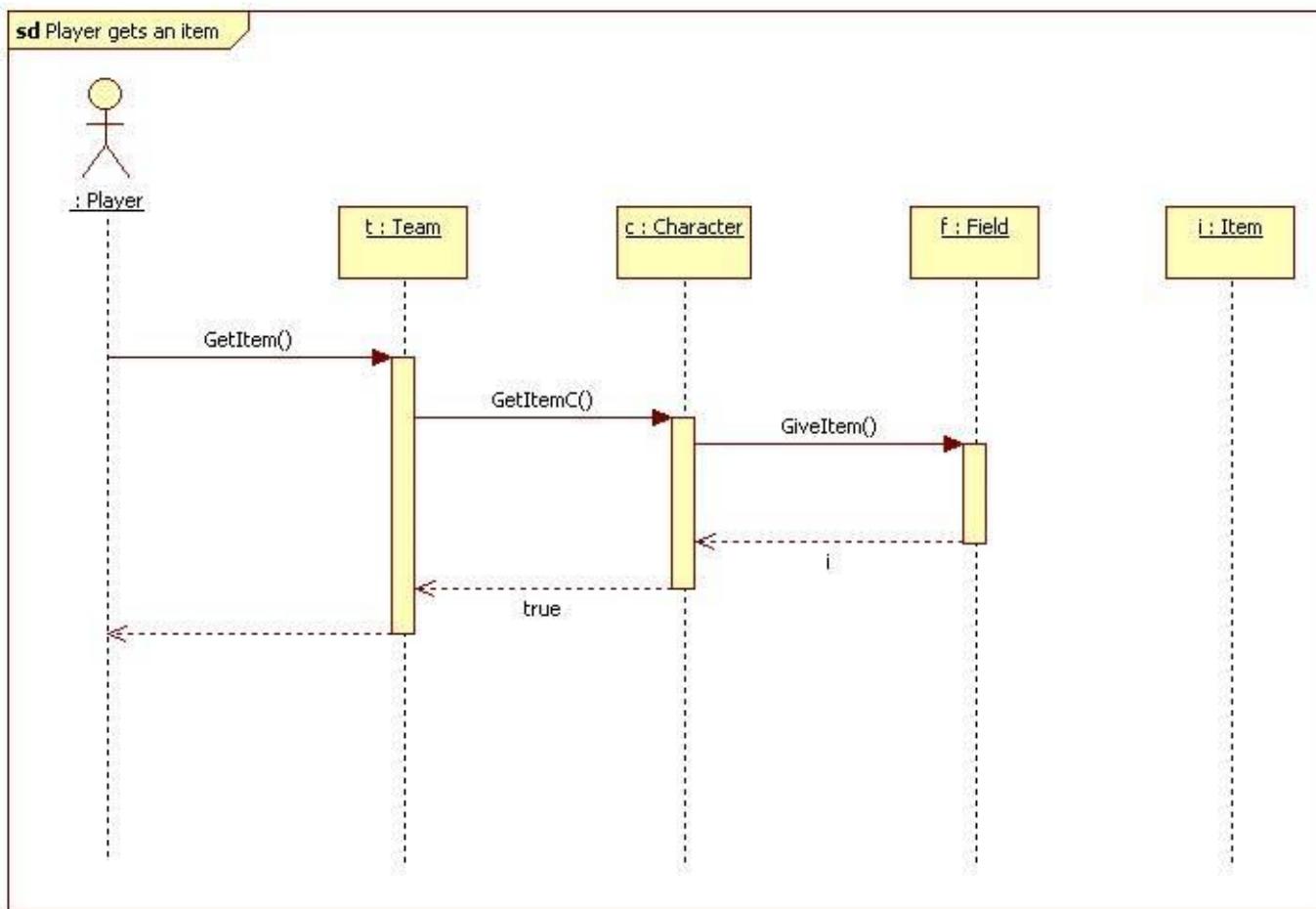


### 3.4.3.6 Player falls into hole without scubasuit



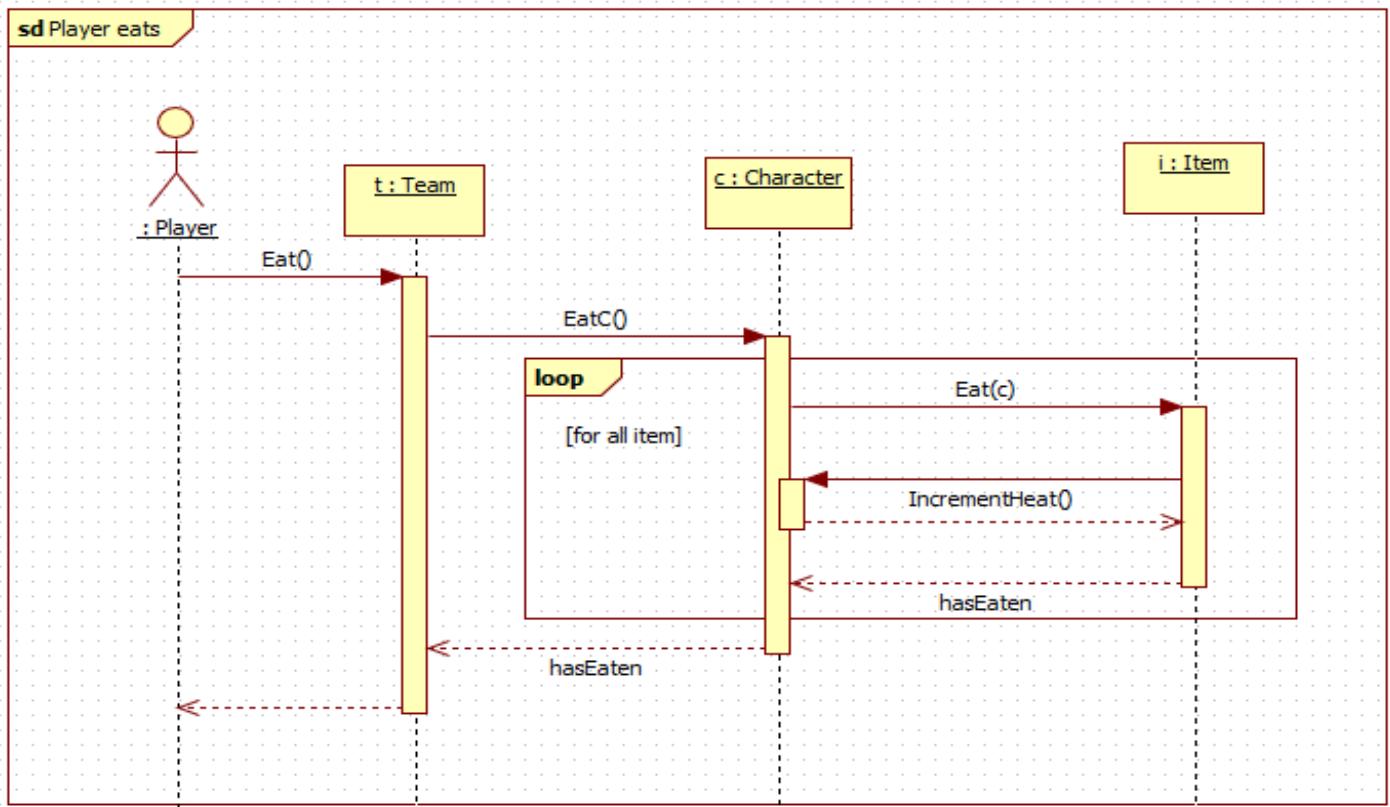
### 3.4.4 Tárgykezelés

#### 3.4.4.1 Player gets an item

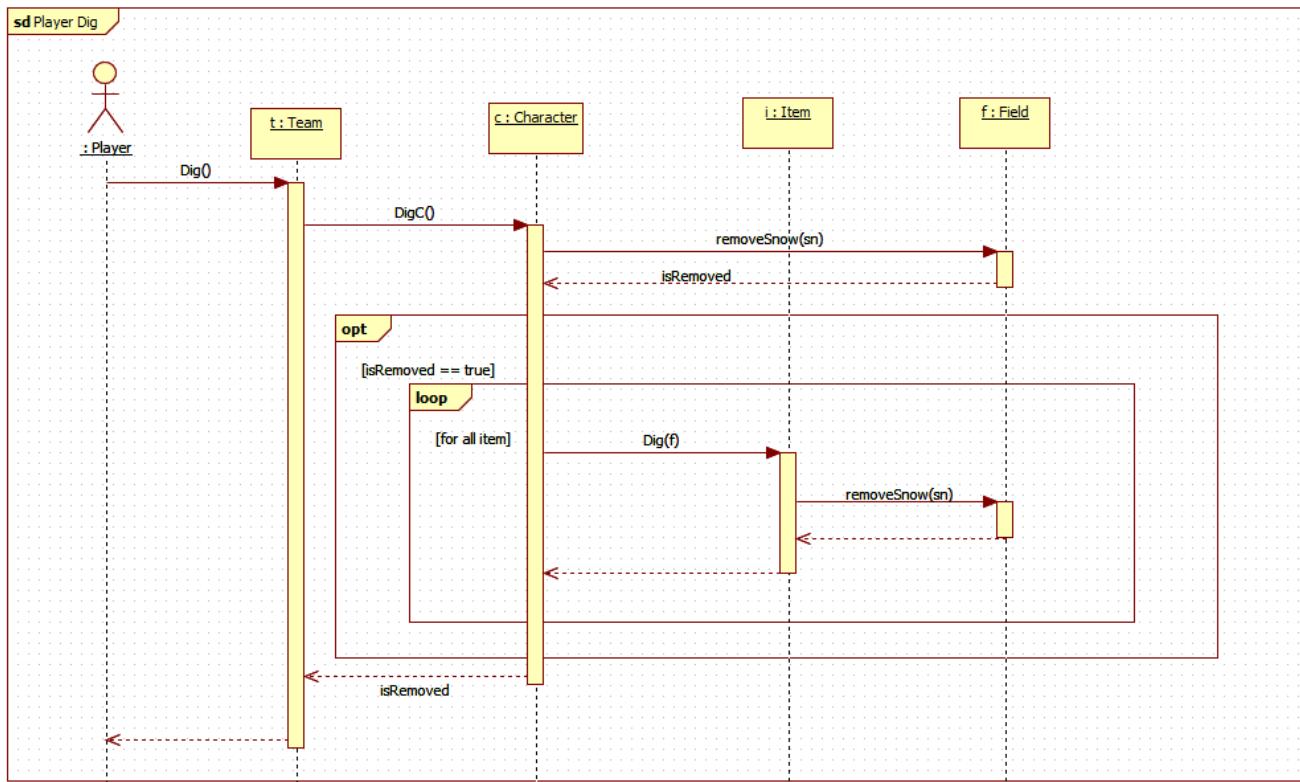


### 3.4.5 Akciók

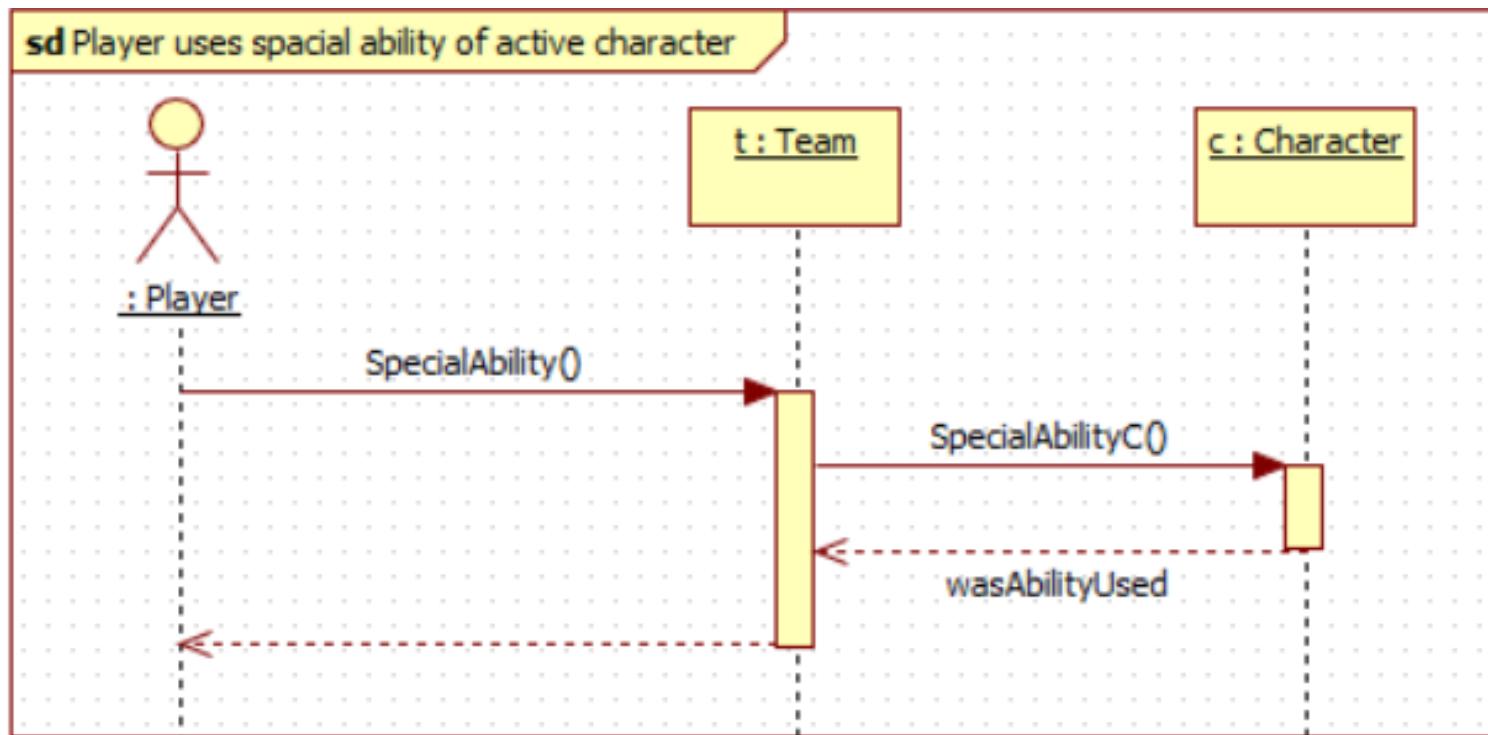
#### 3.4.5.1 Player eats



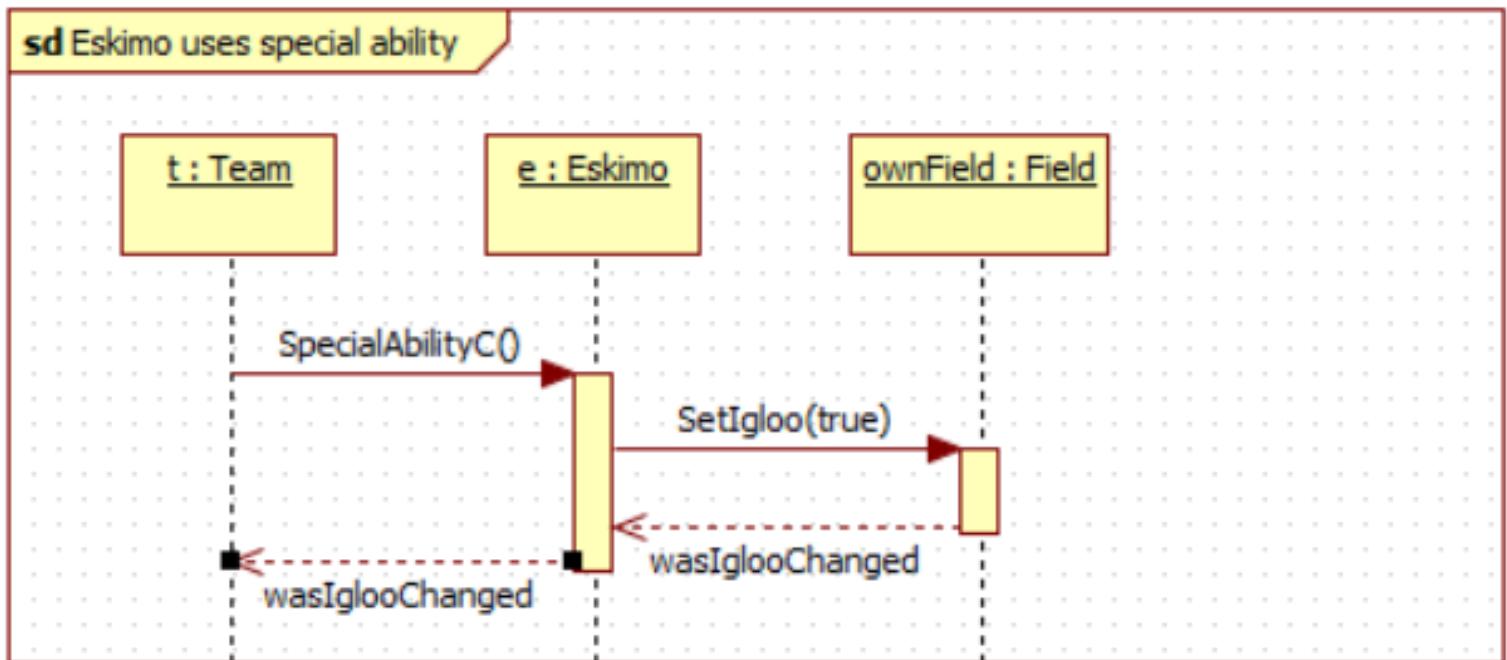
#### 3.4.5.2 Player digs



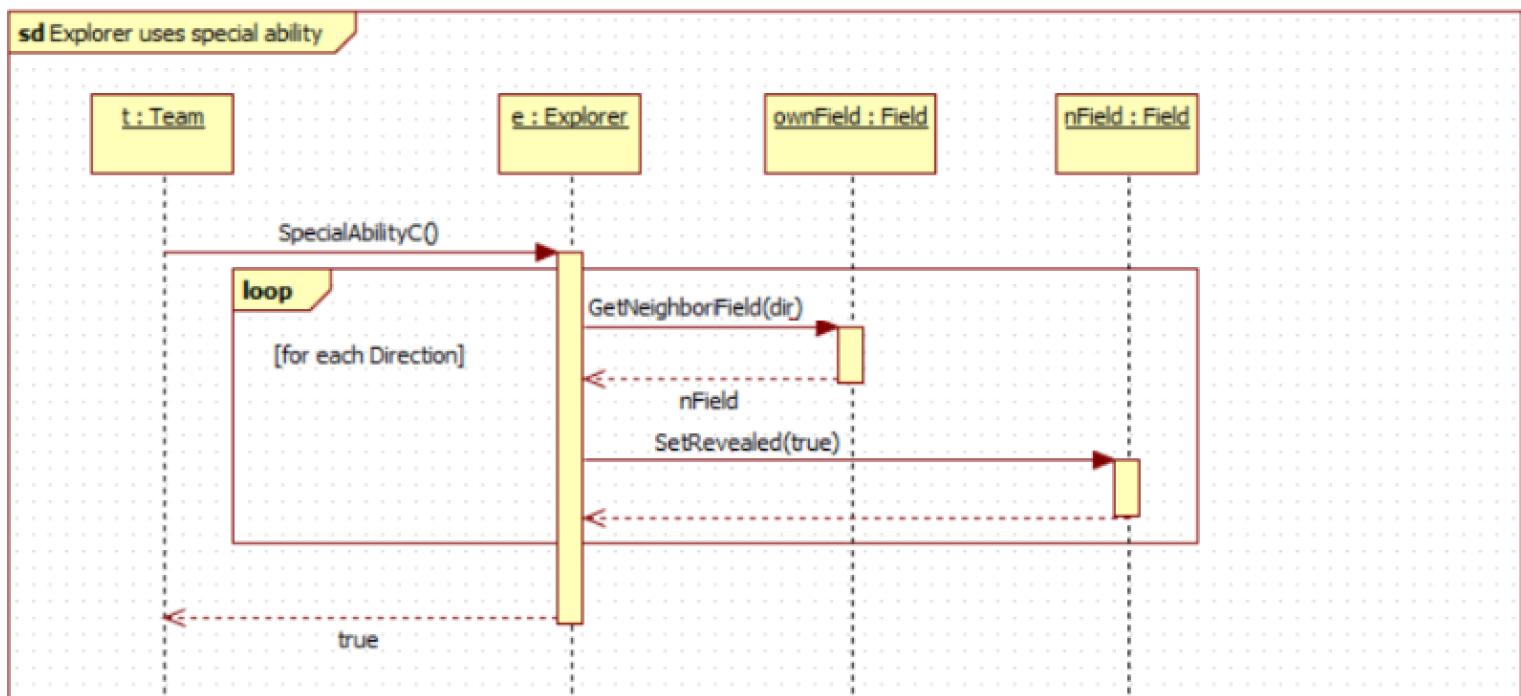
### 3.4.5.3 Player uses special ability of active character



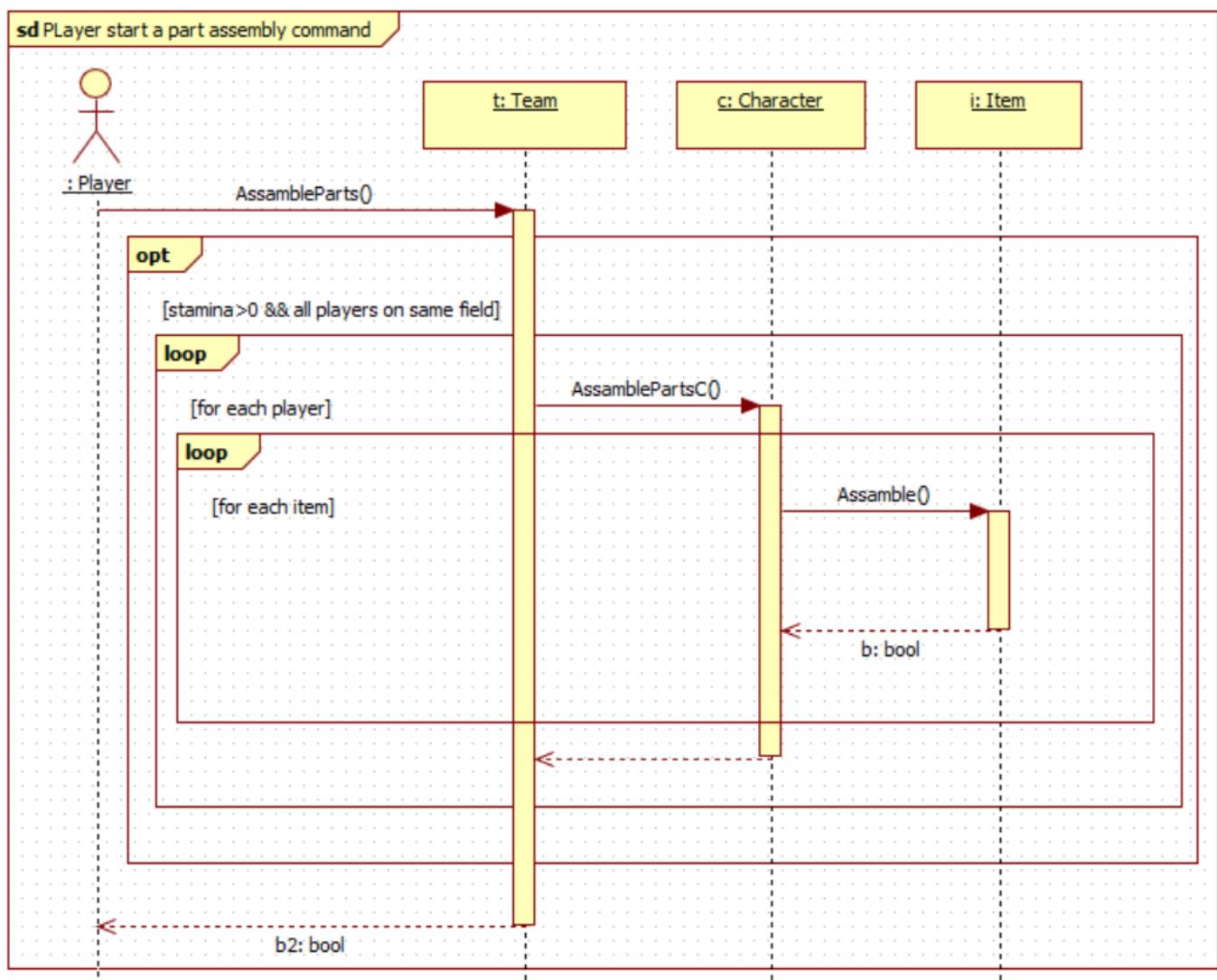
### 3.4.5.4 Eskimo uses special ability



### 3.4.5.5 Explorer uses special ability



### 3.4.5.6 Player starts part assembly command



### 3.3.5Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.03.04. 18:00	2 óra	Háromi, Holczer, Kedves, Kurdi, Marton	<p>Értekezlet.</p> <p>Döntés:</p> <p>Háromi felújítja az osztálydiagramot, Holczer megcsinálja az új tárgystruktúrához tartozó szekvenciadiagramok egyik felét, Kedves kiegészíti az inicializálást, Kurdi elkészíti a tárgyakhoz tartozó szekvenciadiagramok másik felét, Marton felújítja az osztályleírásokat, és a mozgással kapcsolatos szekvenciadiagramokat egészíti ki.</p> <p>Határidő: 2020. 03. 08. este</p>
2020.03.05. 16:00	2,5 óra	Kedves	Tevékenység: Kedves elkészíti az inicializáláshoz tartozó hiányzó diagramokat.
2020.03.07. 19:00	2 óra	Holczer	Tevékenység: Holczer elkészíti az Eat() és Dig() szekvenciadiagramokat, kiegészít osztályleírásokat.
2020.03.07. 20:00	1 óra	Kurdi	Tevékenység: Kurdi elkészíti a karakter kihúzásához és lyukba eséséhez tartozó szekvenciadiagramokat.
2020.03.08. 15:00	2 óra	Marton	Tevékenység: Marton kiegészíti a mozgással kapcsolatos szekvenciadiagramokat, és véglegesíti az osztályleírásokat.
2020.03.08. 18:00	2 óra	Háromi	Tevékenység: Háromi a többiek munkája alapján frissíti az osztálydiagramot, és ellenőrzi a konzisztenciát az osztálydiagram és a szekvenciadiagram között.
2020.03.08. 21:00	1 óra	Kurdi	Tevékenység: Kurdi letisztázza a dokumentumot

## 5. Szkeleton tervezése

45 – nullpointer

Konzulens:  
Ludmány Balázs

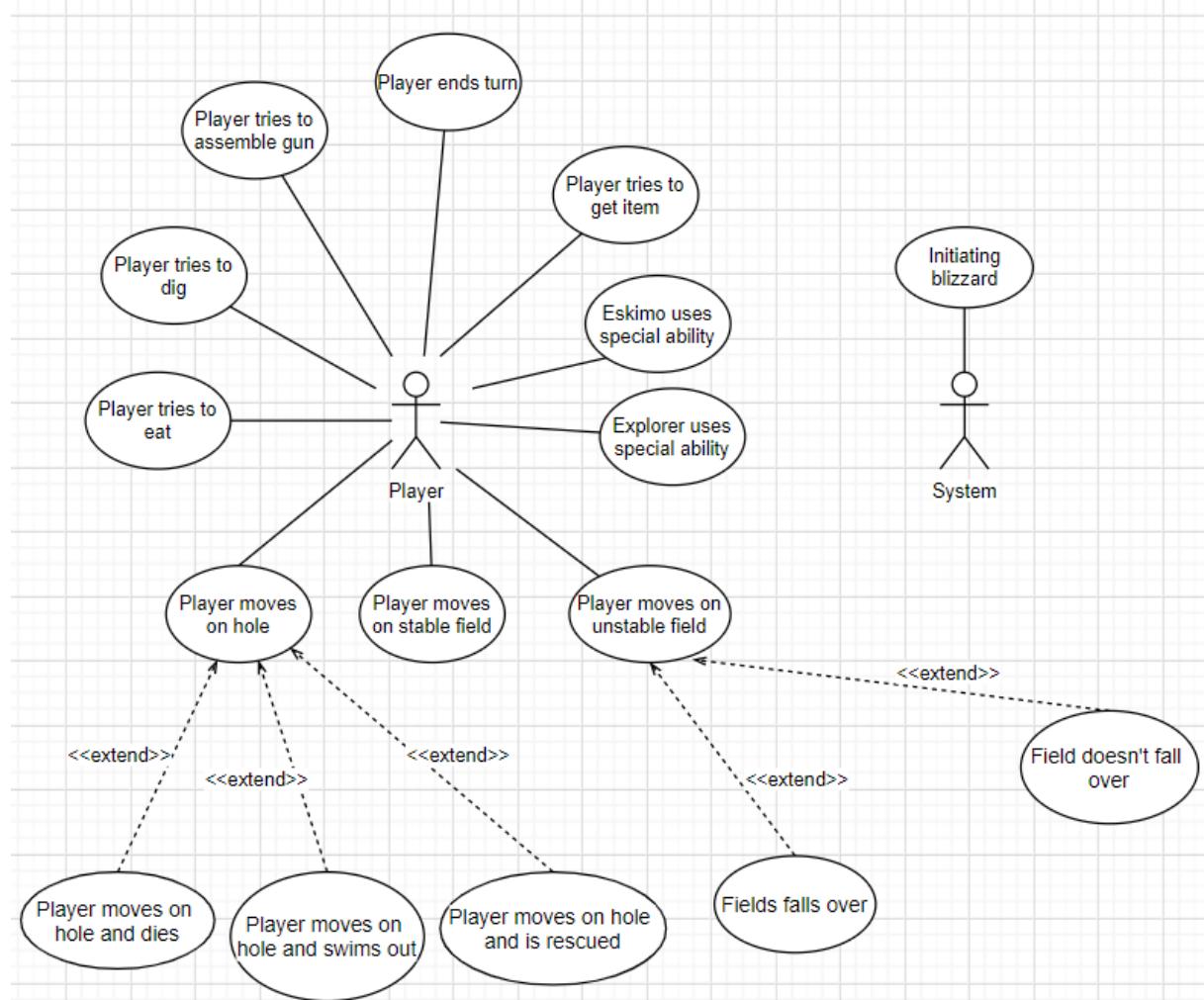
### Csapattagok

Háromi Bence	AK51GI	bence.haromi@gmail.com
Holczer Zsolt	XL9Y0E	zsolti@holczer.net
Kedves Áron Csanád	AV43UW	aron.kedves@gmail.com
Kurdi Barnabás	F0DJNW	barnabas.kurdi@gmail.com
Marton Gergely	FC2WNS	martongergely11@gmail.com

2020.03.23.

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

### 5.1.1 Use-case diagram



Az extend use-case-ek a nagyobb use-case-ek alternatív forgatókönyveiben szerepelnek. Az ábrán az átláthatóság kedvéért szerepelnek ilyen módon.

## 5.1.2 Use-case leírások

### 5.1.2.1. Eskimo uses special ability

<b>Use-case neve</b>	Eskimo uses special ability
<b>Rövid leírás</b>	A játékos eszkimóval van és a speciális képességét próbálja használni.
<b>Aktorok</b>	Player
<b>Fő forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos eszkimóval van.</li> <li>2. A játékos kiadja a speciális képesség használatára szolgáló parancsot.</li> <li>3. A játékos alatti mezőre épít egy iglut.</li> <li>4. A játékos staminája eggyel csökken és a játékos akciója véget ér.</li> </ol>
<b>Alternatív forgatókönyv</b>	<ol style="list-style-type: none"> <li>2.A.1. A játékosnak nincs staminája, ekkor az akció megszakad és a játékos staminája nem csökken.</li> <li>3.A.1. A játékos alatti mezőn már van iglu, ekkor az akció megszakad és a játékos staminája nem csökken.</li> </ol>

### 5.1.2.2. Explorer uses special ability

<b>Use-case neve</b>	Explorer uses special ability
<b>Rövid leírás</b>	A játékos kutatóval van és a speciális képességét próbálja használni.
<b>Aktorok</b>	Player
<b>Fő forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos kutatóval van.</li> <li>2. A játékos kiadja a speciális képesség használatára szolgáló parancsot.</li> <li>3. A játékos felderíti a körülötte lévő mezők teherbírását.</li> <li>4. A játékos staminája eggyel csökken és a játékos akciója véget ér.</li> </ol>
<b>Alternatív forgatókönyv</b>	<ol style="list-style-type: none"> <li>2.A.1. A játékosnak nincs staminája, ekkor az akció megszakad és a játékos staminája nem csökken.</li> </ol>

### 5.1.2.3. Player starts initialisation

<b>Use-case neve</b>	Player starts initialization
<b>Rövid leírás</b>	A játékos inicializálja a játékot.
<b>Aktorok</b>	Player
<b>Fő forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos elindítja a játék létrehozását.</li> <li>2. Létrejön a játék.</li> </ol>

### 5.1.2.4. Player ends turn

<b>Use-case neve</b>	Player ends turn
<b>Rövid leírás</b>	A játékos dönthet úgy, hogy a staminájának elfogyása előtt befejezi a körét.
<b>Aktorok</b>	Player
<b>Fő forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos kiadja a kör korai befejezésére szolgáló parancsot.</li> <li>2. A játékos köre véget ér.</li> </ol>

### 5.1.2.5. Player tries to eat

<b>Use-case neve</b>	Player tries to eat
<b>Rövid leírás</b>	A játékos megpróbál ételt fogyasztani.
<b>Aktorok</b>	Player
<b>Fő forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos kiadja az étel fogyasztására szolgáló parancsot.</li> <li>2. A játékos staminája, élelmiszere egyel csökken és a testhője eggyel nő.</li> <li>3. Az akció véget ér.</li> </ol>
<b>Alternatív forgatókönyv</b>	<ol style="list-style-type: none"> <li>2.A.1. A játékosnak nincs élelme, a játékosnak nem változnak a tulajdonságai és a harmadik pontra ugrunk.</li> <li>2.B.1. A játékosnak nincs staminája, a játékosnak nem változnak a tulajdonságai és a harmadik pontra ugrunk.</li> <li>2.C.1. A játékos testhője maximális, a játékosnak nem változnak a tulajdonságai és a harmadik pontra ugrunk.</li> </ol>

### 5.1.2.6. Player tries to dig

<b>Use-case neve</b>	Player tries to dig
<b>Rövid leírás</b>	A játékos megpróbálja ellapátolni a havat a mezőről, amin áll.
<b>Aktorok</b>	Player
<b>Fő forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos kiadja az ásásra szolgáló parancsot.</li> <li>2. A játékos alatt lévő hó mennyisége és staminája egyel csökken.</li> <li>3. Az akció véget ér.</li> </ol>
<b>Alternatív forgatókönyv</b>	<ol style="list-style-type: none"> <li>2.A.1. A játékosnak alatti mezőn nincs hó, a játékosnak nem csökken a staminája.</li> <li>2.B.1. A játékosnak nincs staminája, a játékosnak nem csökken a staminája.</li> <li>2.C.1. A játékosnak van ásója, ekkor kettővel próbálja csökkenteni a játékos alatti mezőn lévő hó mennyiséget.</li> <li>2.C.1.A.1. A játékos alatti mezőn csak egy hó van, ekkor csak eggyel csökken a hómennyiség.</li> </ol>

### 5.1.2.7. Player tries to assemble gun

<b>Use-case neve</b>	Player tries to assemble gun
<b>Rövid leírás</b>	A játékos megpróbálja összeszerelni a játék megnyeréséhez szükséges jelzőrakétát.
<b>Aktorok</b>	Player
<b>Fő forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékosok minden mezőn vannak.</li> <li>2. A játékosok rendelkeznek az összes alkatrésszel.</li> <li>3. A játékos kiadja a jelzőpisztoly összeszerelésére szolgáló parancsot.</li> <li>4. Az akció és a játék véget ér.</li> </ol>
<b>Alternatív forgatókönyv</b>	<ol style="list-style-type: none"> <li>1.A.1. A játékosok nem minden mezőn vannak, ezért az akció megszakad és a parancsot kiadó játékos staminája nem csökken.</li> <li>2.A.1. A játékosok nem rendelkeznek az összes alkatrésszel, ezért az akció megszakad és a parancsot kiadó játékos staminája nem csökken.</li> <li>3.A.1. A játékosnak nincs staminája, ezért az akció megszakad és a parancsot kiadó játékos staminája nem csökken.</li> </ol>

### 5.1.2.8. Player tries to get item

<b>Use-case neve</b>	Player tries to get item
<b>Rövid leírás</b>	A játékos megpróbál tárgyat felvenni a mezőről, amin áll.
<b>Aktorok</b>	Player
<b>Fő forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos olyan mezőn áll, ahol nincs hó.</li> <li>2. A játékos alatti mezőn van tárgy.</li> <li>3. A játékos kiadja a tárgy felvételére szolgáló parancsot.</li> <li>4. A játékoshoz kerül a mezőn lévő tárgy, a játékos staminája egygyel csökken és akció véget ér.</li> </ol>
<b>Alternatív forgatókönyv</b>	<ol style="list-style-type: none"> <li>1.A.1. A játékos alatti mezőn van hó, ekkor az akció megszakad és a játékos staminája nem csökken.</li> <li>2.A.1. A játékos alatti mezőn nincs tárgy, ekkor az akció megszakad és a játékos staminája nem csökken.</li> <li>3.A.1. A játékosnak nincs staminája, ekkor az akció megszakad és a játékos staminája nem csökken.</li> </ol>

### 5.1.2.9. Player tries to move out of play area

<b>Use-case neve</b>	Player tries to move out of play area
<b>Rövid leírás</b>	A játékos megpróbál kimozogni a pályáról
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos olyan irányba ad mozgási parancsot, amerre nincs már mező.</li> <li>2. A játékos az eredeti pozíójában marad.</li> </ol>

### 5.1.2.10. Player moves on hole

<b>Use-case neve</b>	Player moves on hole
<b>Rövid leírás</b>	A játékos lyukra mozog
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<p>1. A játékos lyukra mozog.</p> <p>2.A.1. A játékos rendelkezik búvárruhával.</p> <p>2.A.2. A játékos folytathatja körét, elmozoghat a lyukról.</p> <p>2.B.1. A játékos nem rendelkezik búvárruhával.</p> <p>2.B.1.A.1. A lyukkal szomszédos mezőn áll olyan játékos, akinél van kötél.</p> <p>2.B.1.A.2. A lyukba esett játékos automatikusan a kötéllel rendelkező játékos mezőjére mozog, a lyukba esett játékos köre folytatódik.</p> <p>2.B.1.B.1. A lyukkal szomszédos mezőn nem áll olyan játékos, akinél van kötél.</p> <p>2.B.1.B.2. A játékos beleset a lyukba, a köre befejeződik, a játékos hőereje 0-ra csökken, és a játéknak vége.</p>

### 5.1.2.11. Player moves on stable field

<b>Use-case neve</b>	Player moves on stable field
<b>Rövid leírás</b>	A játékos stabil mezőre mozog
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	1. A játékos átmozog a stabil mezőre.

### 5.1.2.12. Player moves on unstable field

<b>Use-case neve</b>	Player moves on unstable field
<b>Rövid leírás</b>	A játékos instabil mezőre mozog
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<p>1. A játékos átmozog az instabil mezőre.</p> <p>1.A.1. A mező nem terhelődött túl az új játékossal.</p> <p>1.A.2. A játékos köre folytatódik.</p> <p>1.B.1. A mező túlterhelődött az új játékossal együtt.</p> <p>1.B.2. A mező felborul, az összes rajta álló játékos belesik a vízbe, ezzel 0-ra csökken a hőerejük, és rögtön vége van a játékos körének. A játéknak vége.</p>

### 5.1.2.13. System initiates blizzard

<b>Use-case neve</b>	System initiates blizzard
<b>Rövid leírás</b>	A pálya hóvihart indít
<b>Aktorok</b>	System
<b>Forgatókönyv</b>	<p>1. A rendszer úgy dönt, hogy egy játékos köre végén hóvihart indít.</p> <p>2. A rendszer kiválasztja a pálya néhány mezőjét, ezeket hóviharral sújtja.</p> <p>3. A hóviharral sújtott mezőkön eggyel nő a hó.</p> <p>4.A.1. Nem áll játékos hóviharral sújtott mezőn.</p> <p>4.B.1. Áll játékos hóviharral sújtott mezőn.</p> <p>4.B.2.A.1. A mezőn van iglu.</p> <p>4.B.2.A.2. A játékossal nem történik semmi.</p> <p>4.B.2.B.1. A mezőn nincs iglu.</p> <p>4.B.2.B.2. A mezőn álló játékosok hőereje eggyel csökken.</p>

## 5.2 A szkeleton kezelői felületének terve, dialógusok

A programot elindítva kiíródnak a forgatókönyvek, amelyek közül az egyiknek a sorszámát begépelve indíthatjuk el. Ezután a lejjebb szemléltetett módon íródnak ki a forgatókönyv szerint meghívott függvények nevei:

*[behúzás][objektum, melyen a függvény hívódott].[a függvény neve]*

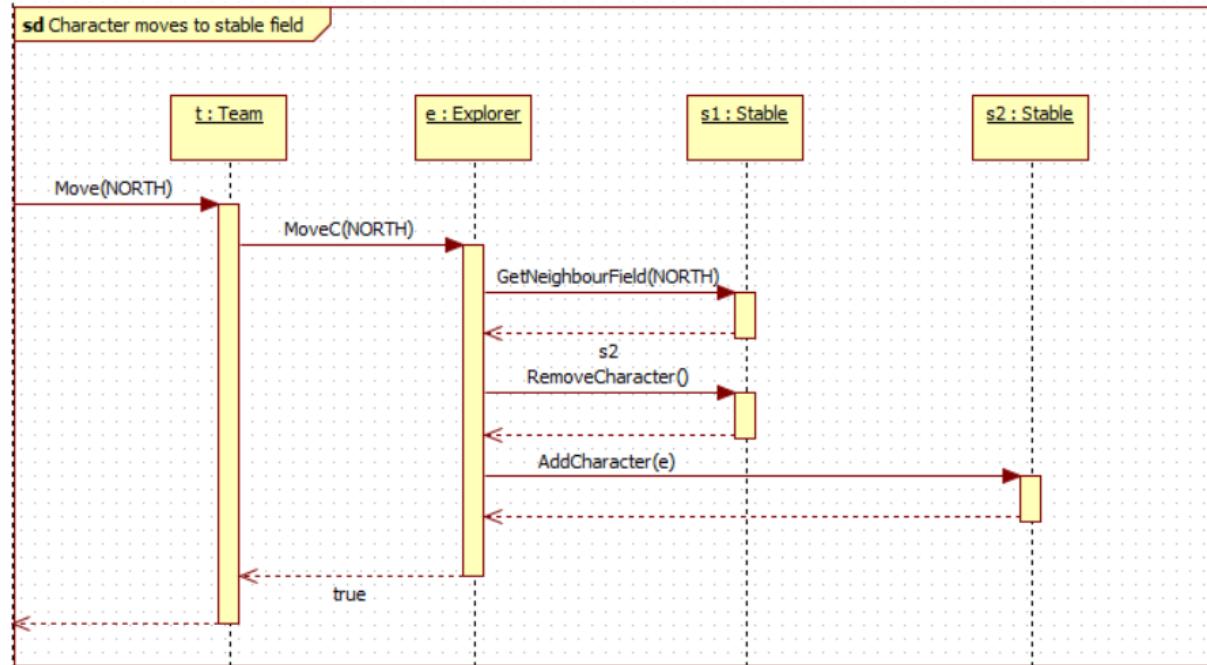
A main-ből hívott függvény előtt nincs behúzás. Ha egy függvényen belül hívódott egy másik függvény, akkor nő a behúzás mértéke, ha viszont visszatér egy függvény és utána egy másik hívódik, akkor ugyanakkora behúzás lesz minden két függvény előtt.

A program írhat ki kérdéseket, amelyek a forgatókönyv elágazó eseteikor különböző végkifejletet okozhatnak (pl. ásáskor mennyi hó van a táblán?), ilyenkor a kérdés utáni zárójelben jelzi a program, hogy mit vár (egész számot, vagy igen/nem választást) és a válasz szerint fut tovább.

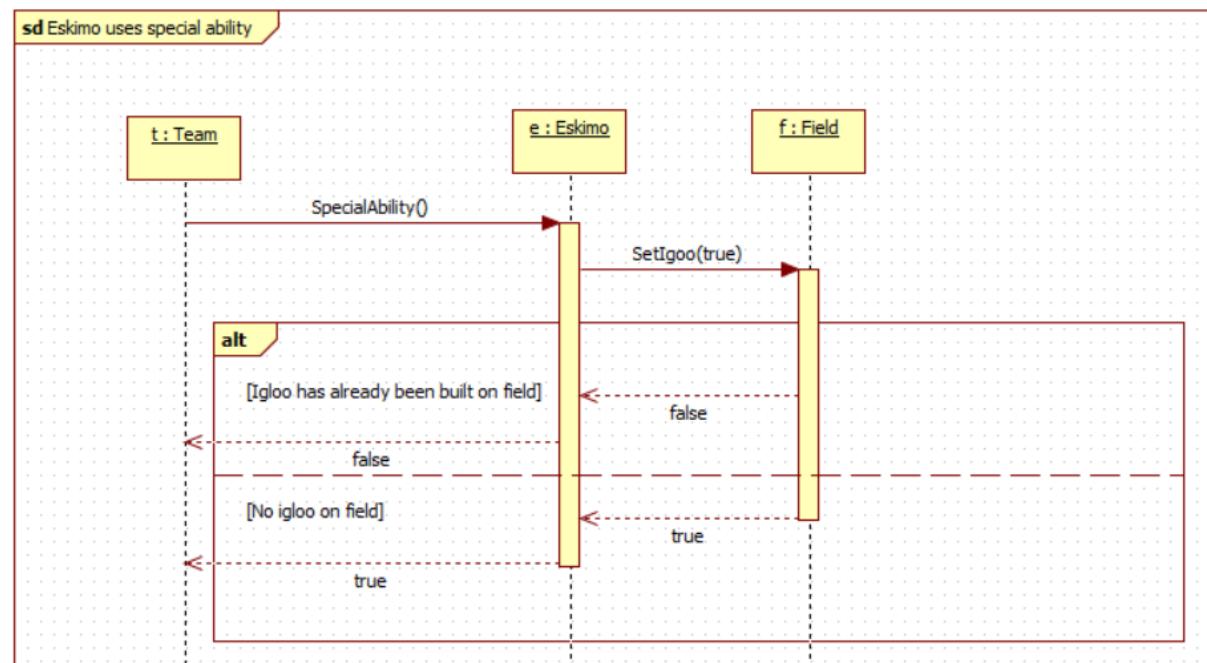
### 5.3 Szekvencia diagramok a belső működésre

A szekvencia diagramok egyrészt csak a karakterek és mező fajták egy fajtájával ábrázoltuk a forgatókönyvet, ezen esetekben az eredmény a fajtától függetlenül ugyanaz.

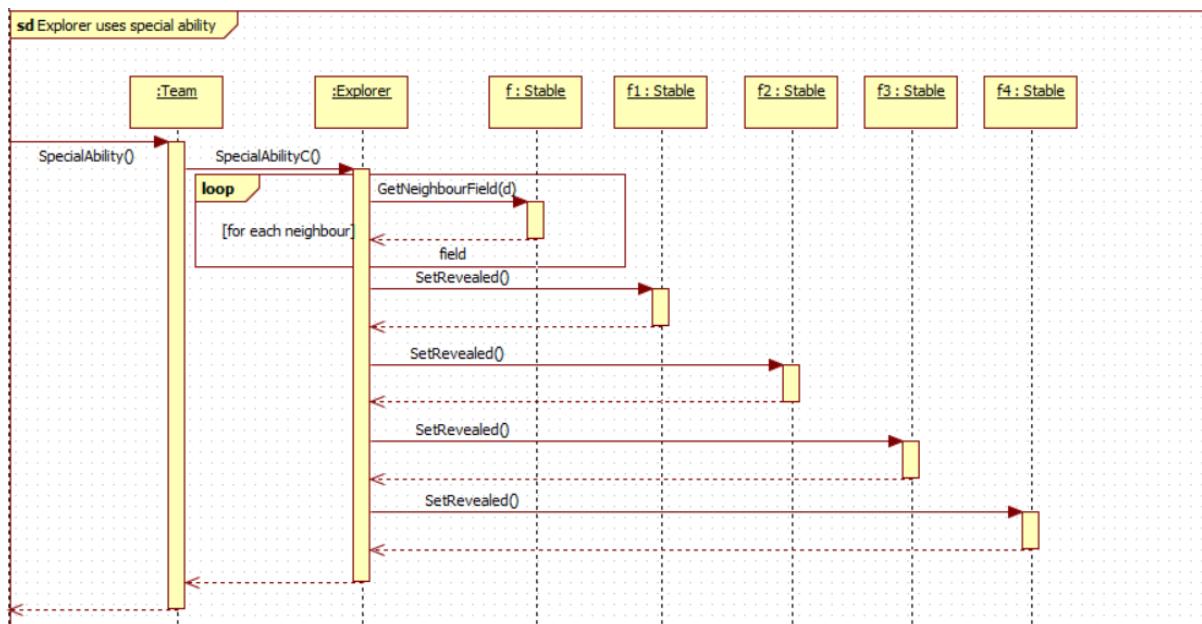
#### 5.3.1. Character moves to stable field



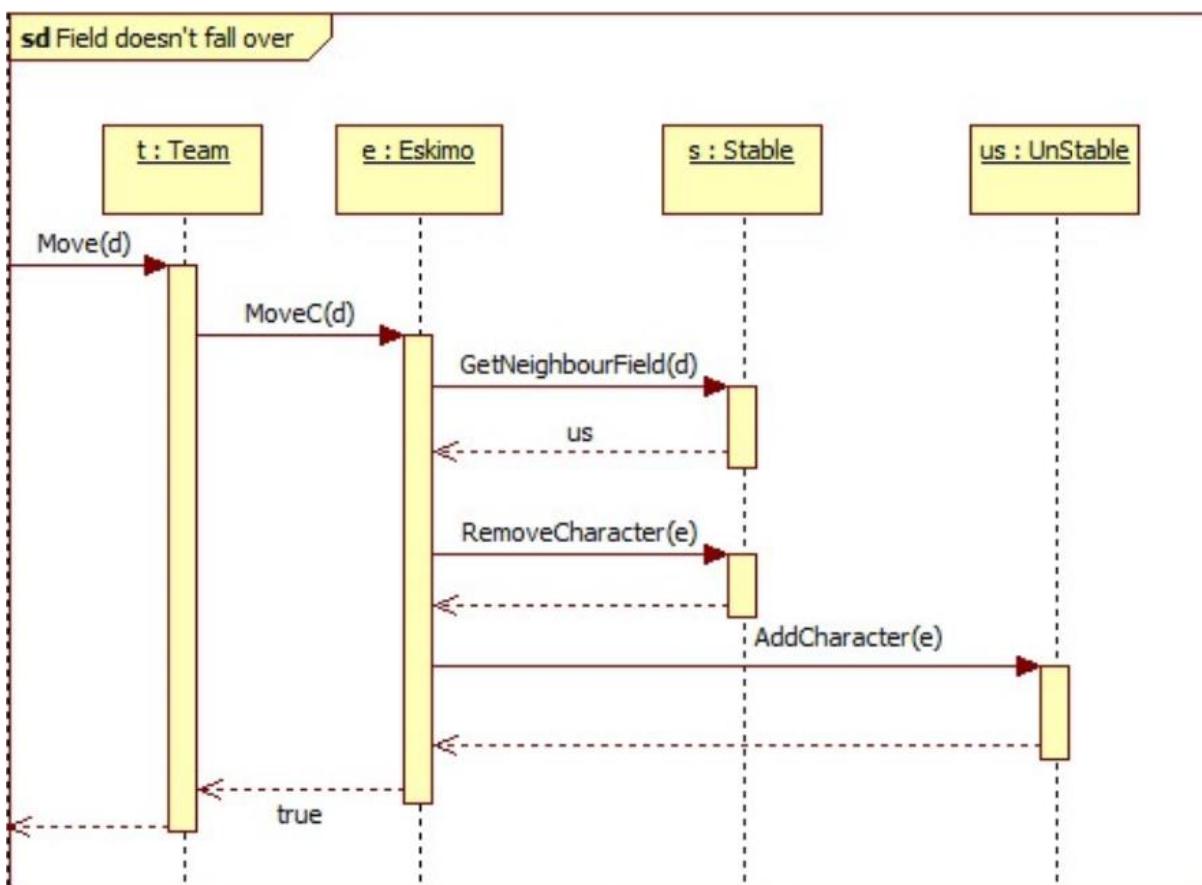
#### 5.3.2. Eskimo uses special ability



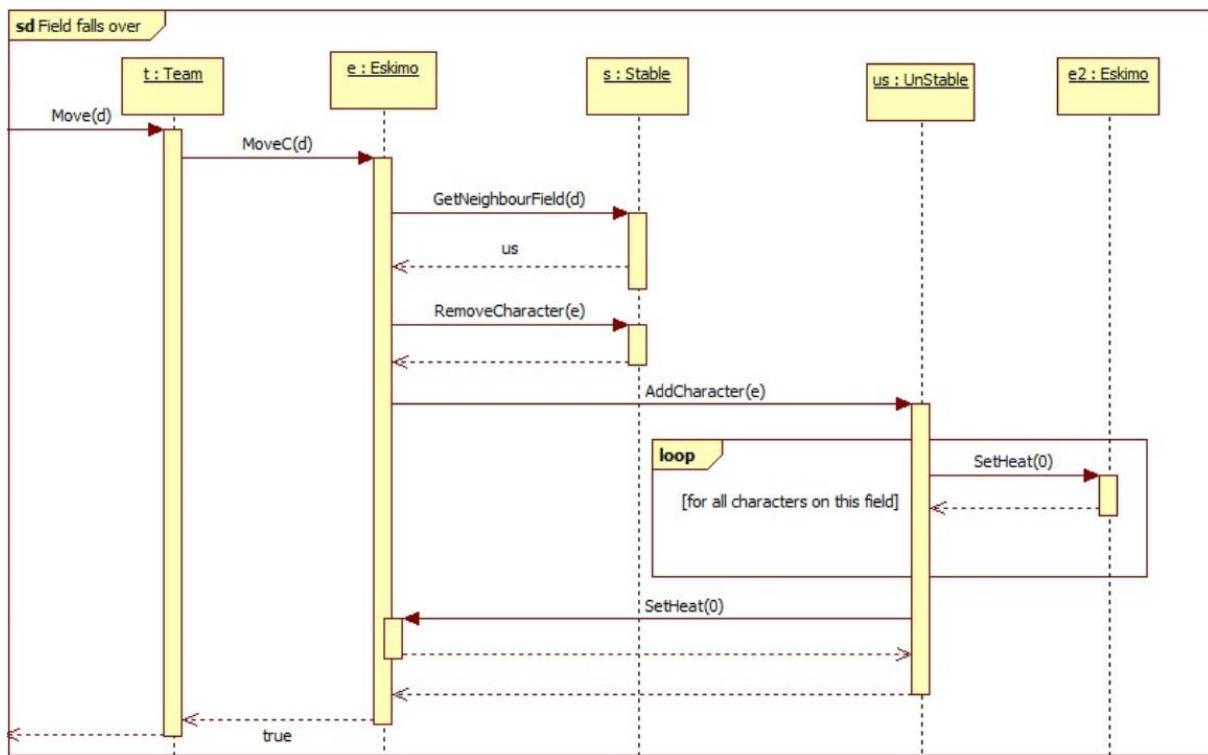
### 5.3.3. Explorer uses special ability



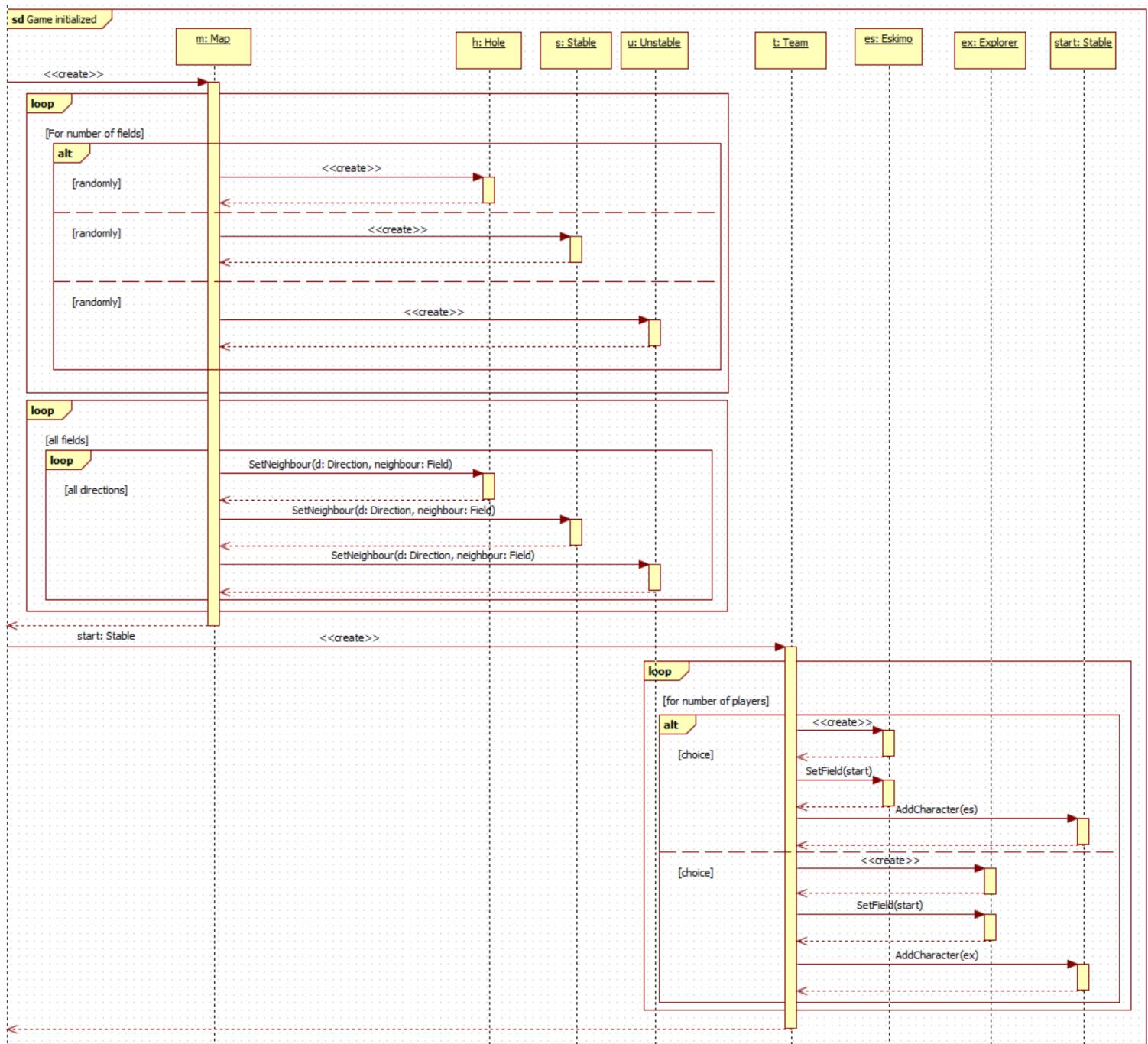
### 5.3.4. Field doesn't fall over



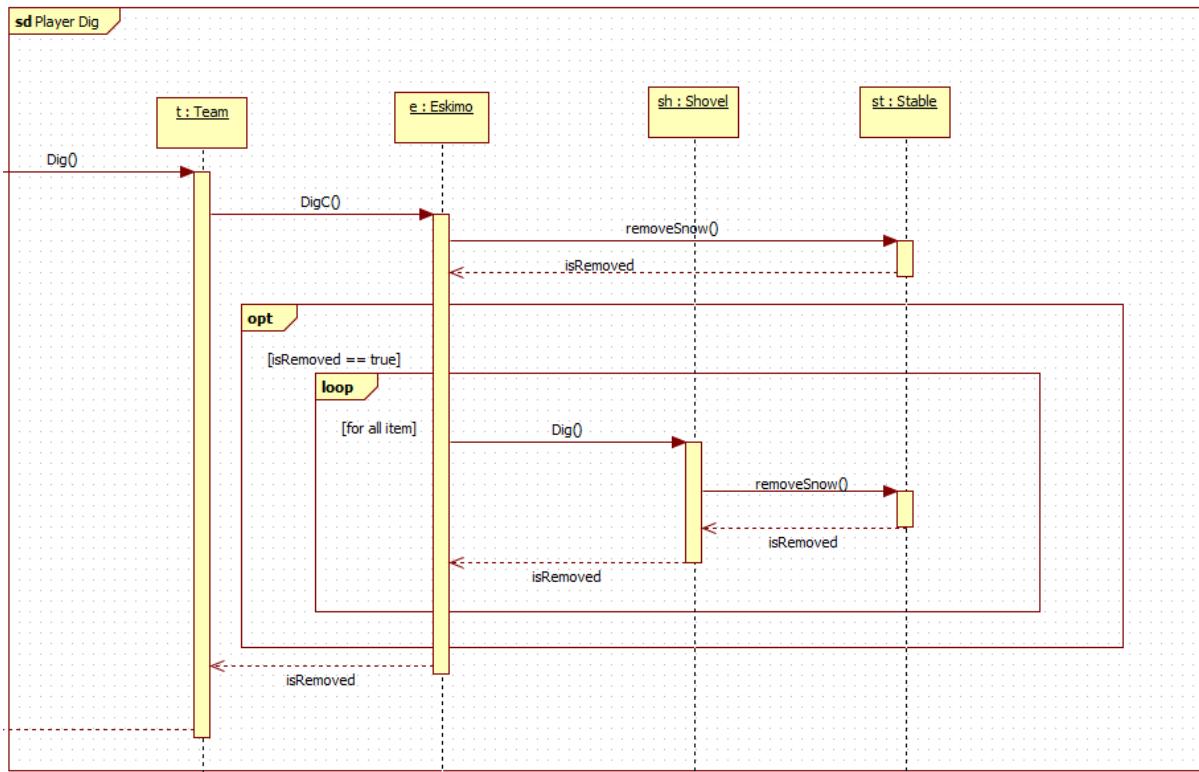
### 5.3.5. Field falls over



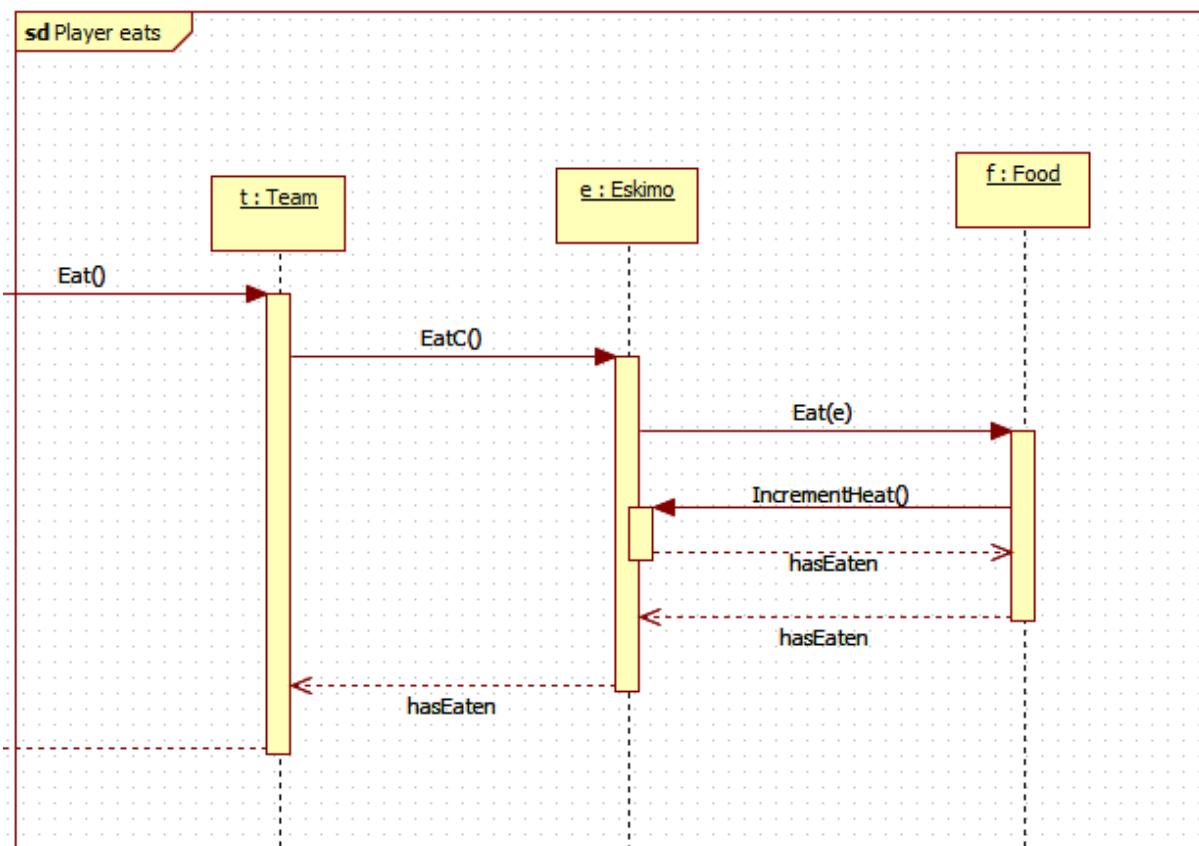
### 5.3.6. Game initialised



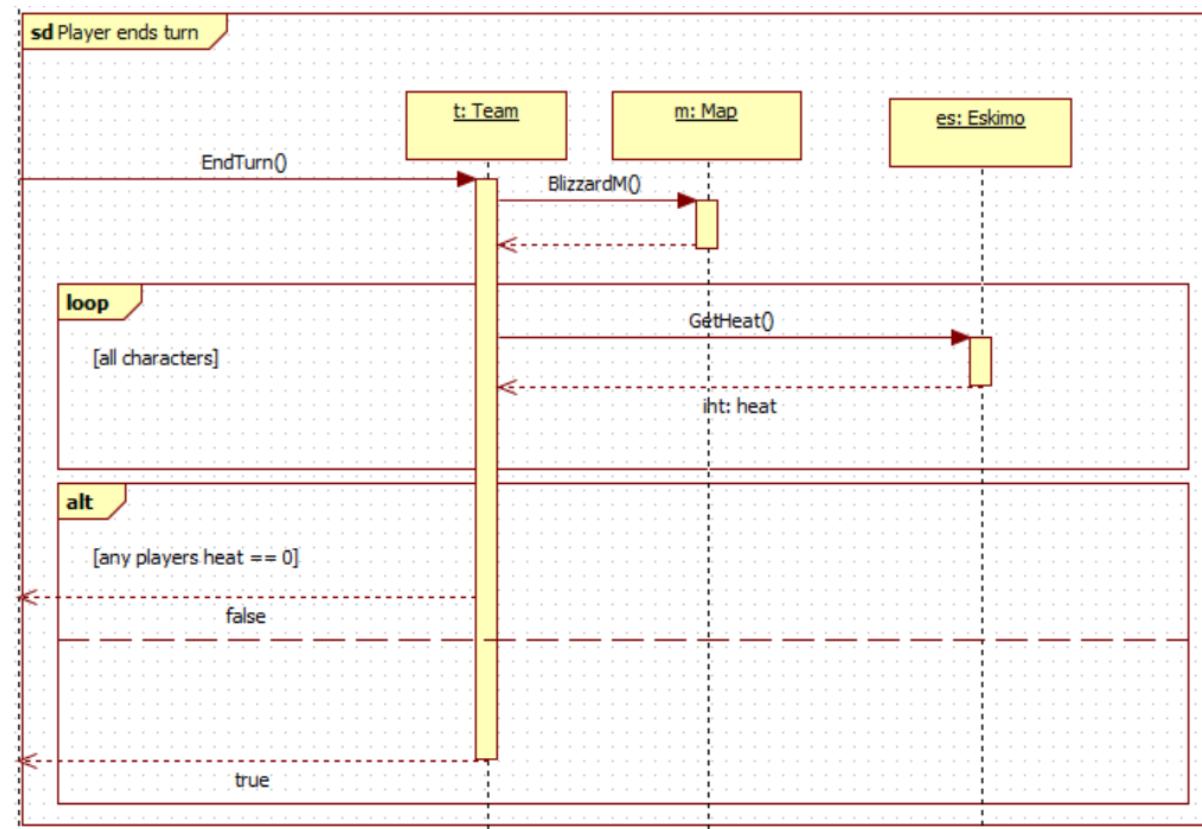
### 5.3.7. Player dig



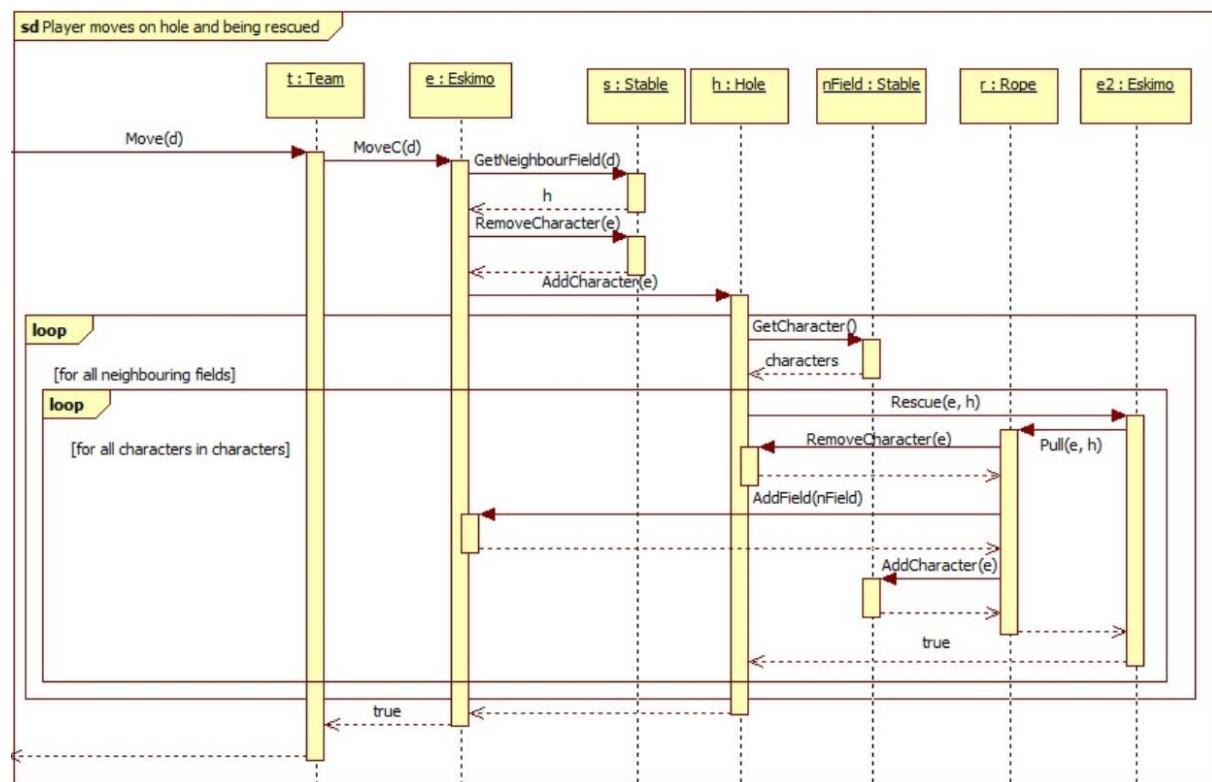
### 5.3.8. Player eats



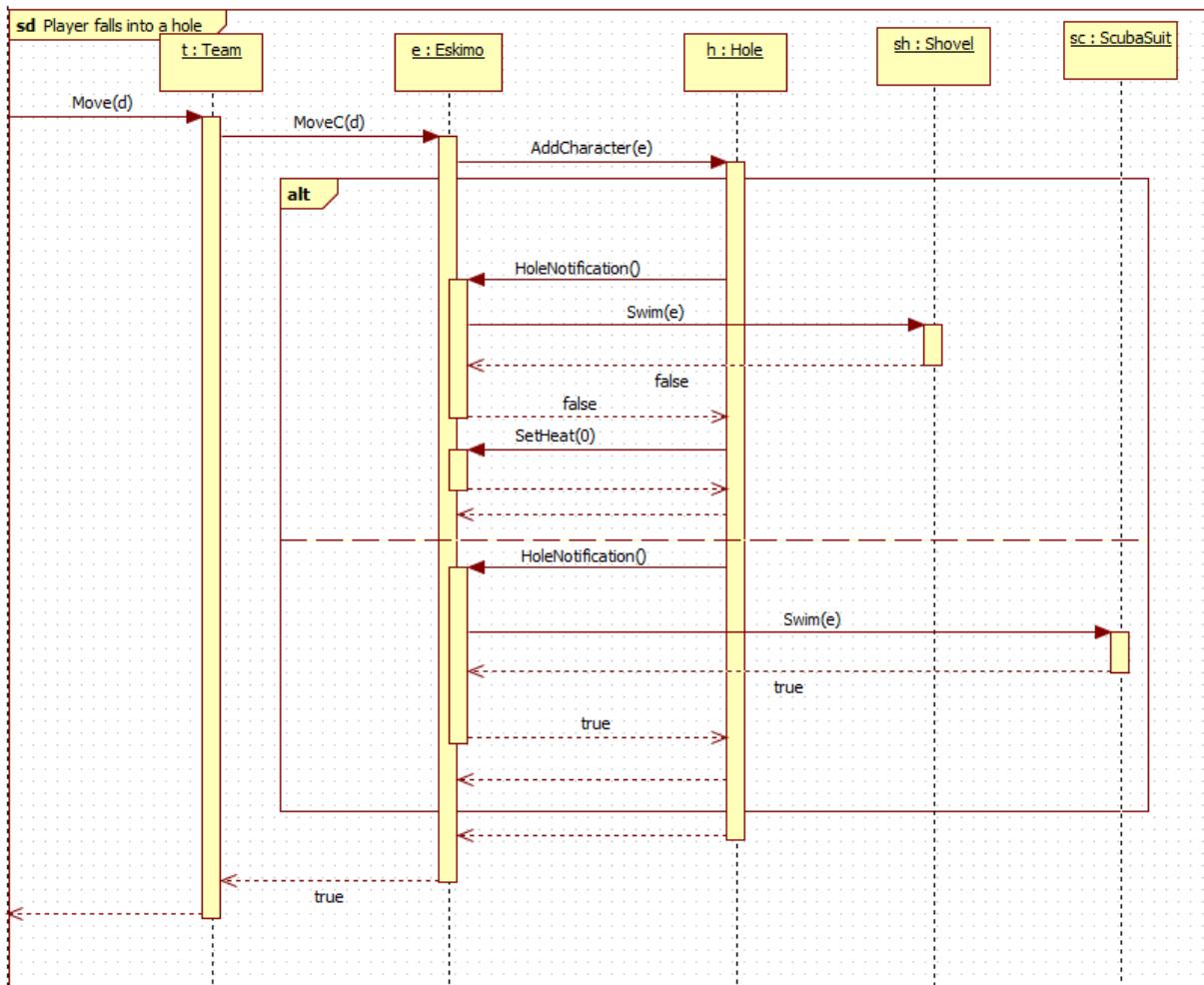
### 5.3.9. Player ends turn



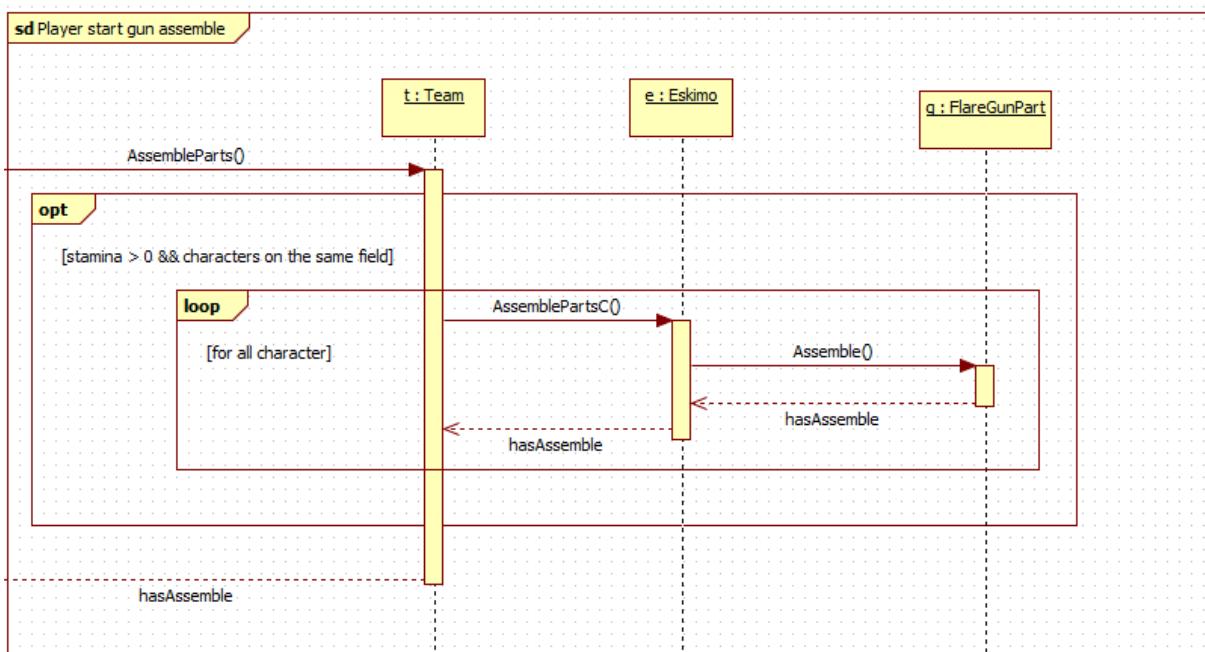
### 5.3.10. Player moves on hole and being rescued



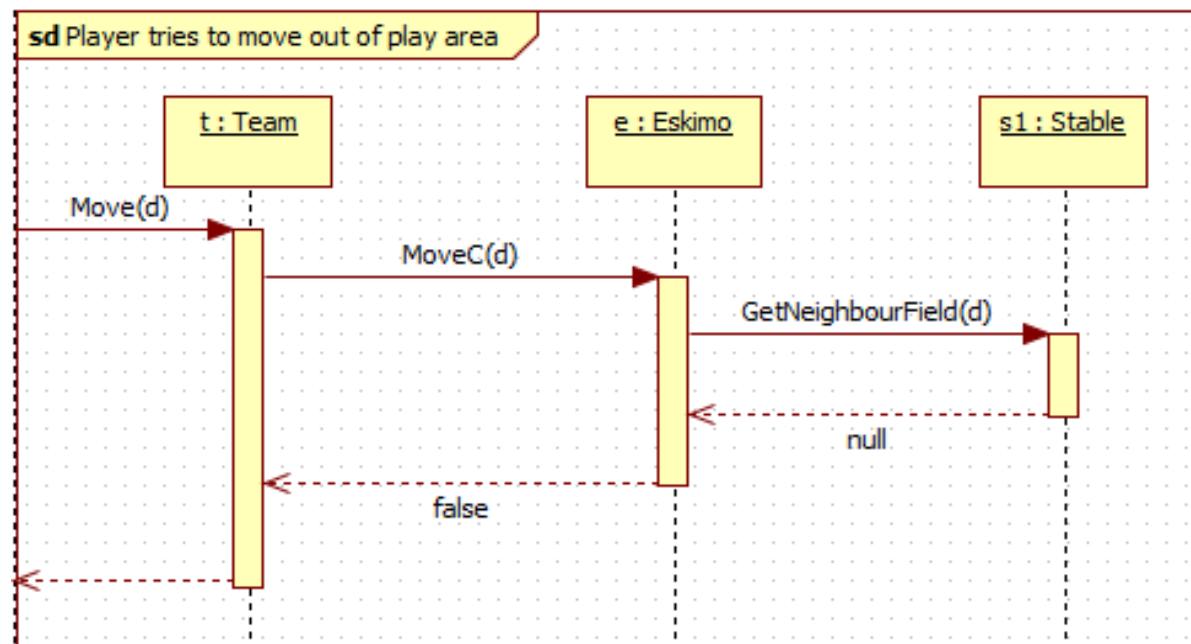
### 5.3.11. Player falls into a hole



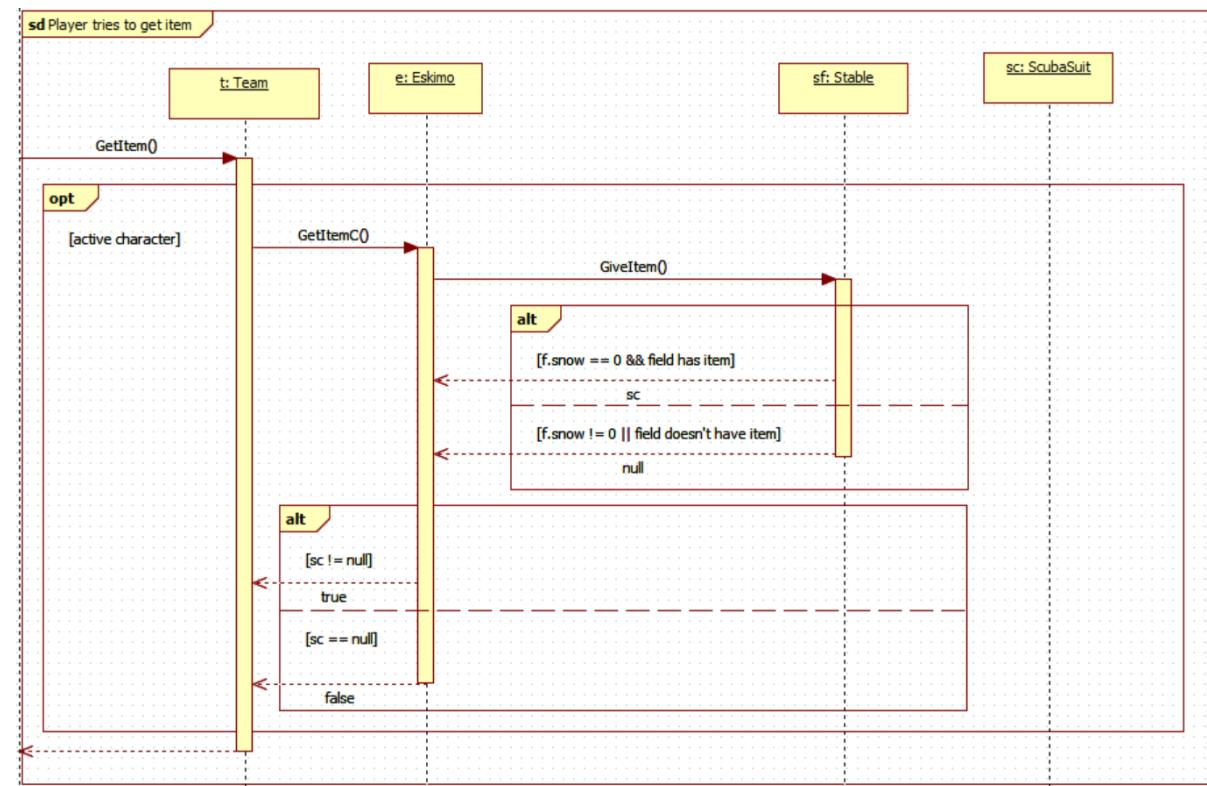
### 5.3.12. Player start gun assemble



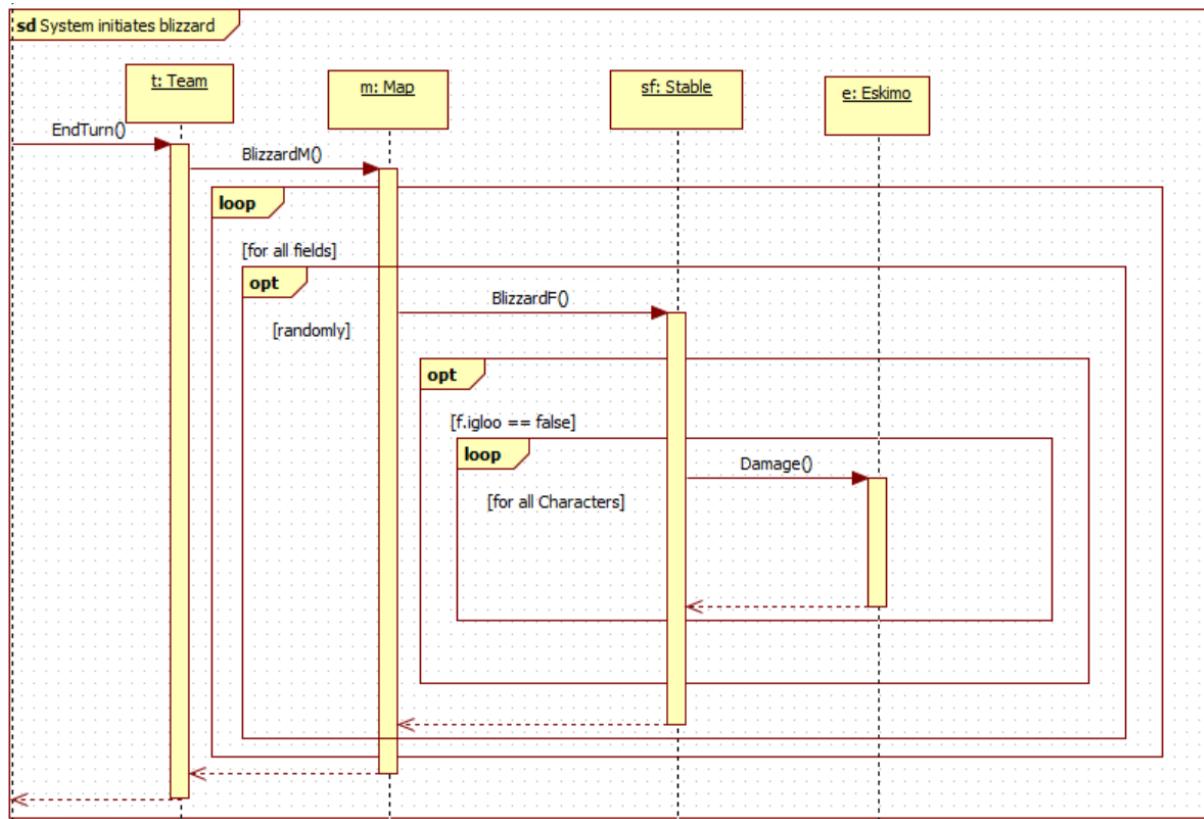
### 5.3.13. Player tries to move out of play area



### 5.3.14. Player tries to get item



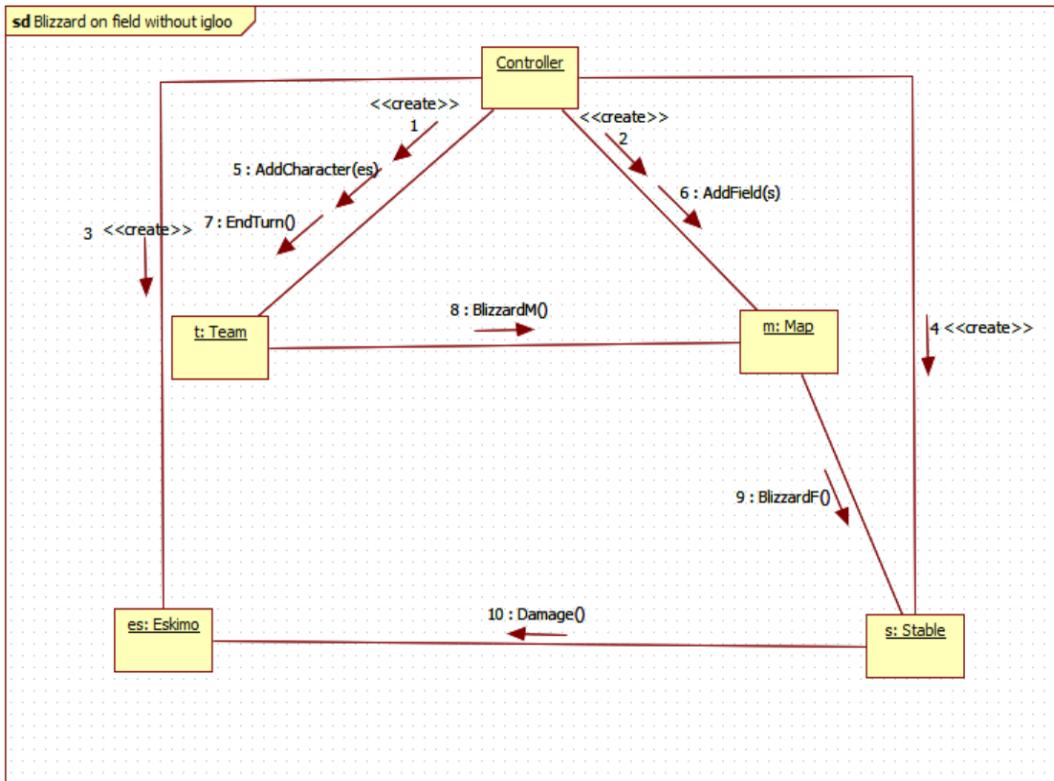
### 5.3.15. System initiates blizzard



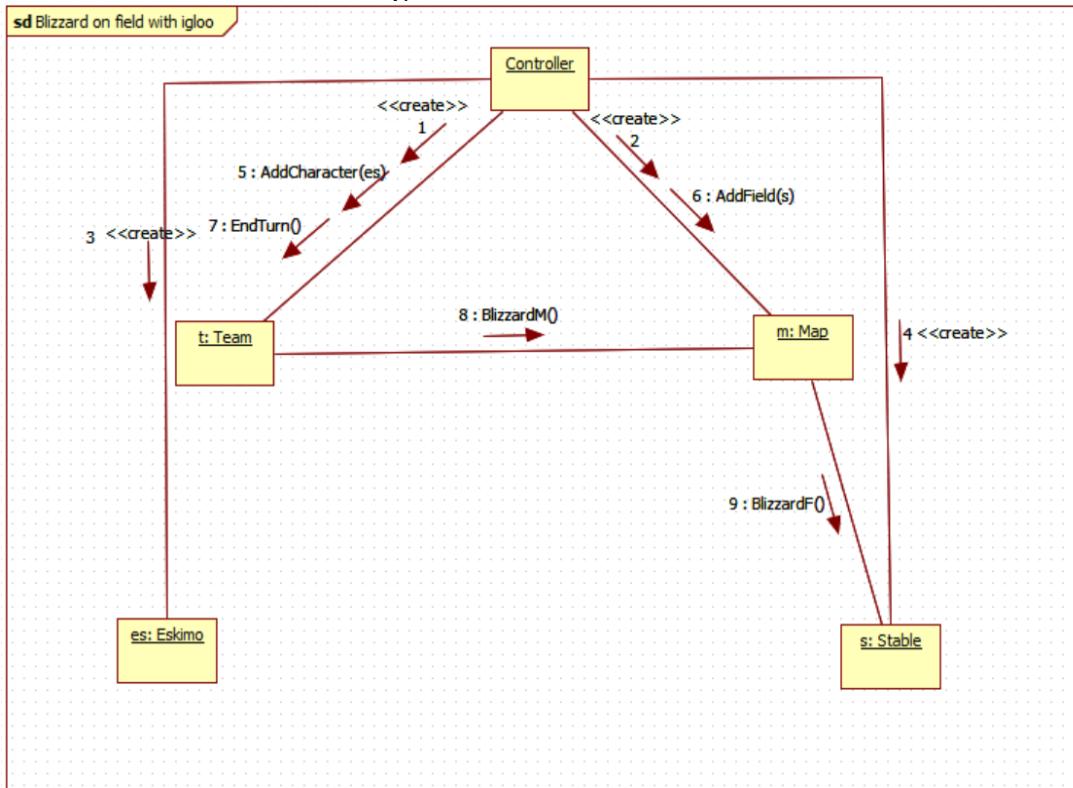
## 5.4 Kommunikációs diagramok

A kommunikációs diagramok nagyrészén csak a karakterek és mező fajták egy fajtájával ábrázoltuk a forgatókönyvet, ezen esetekben az eredmény a fajtától függetlenül ugyanaz.

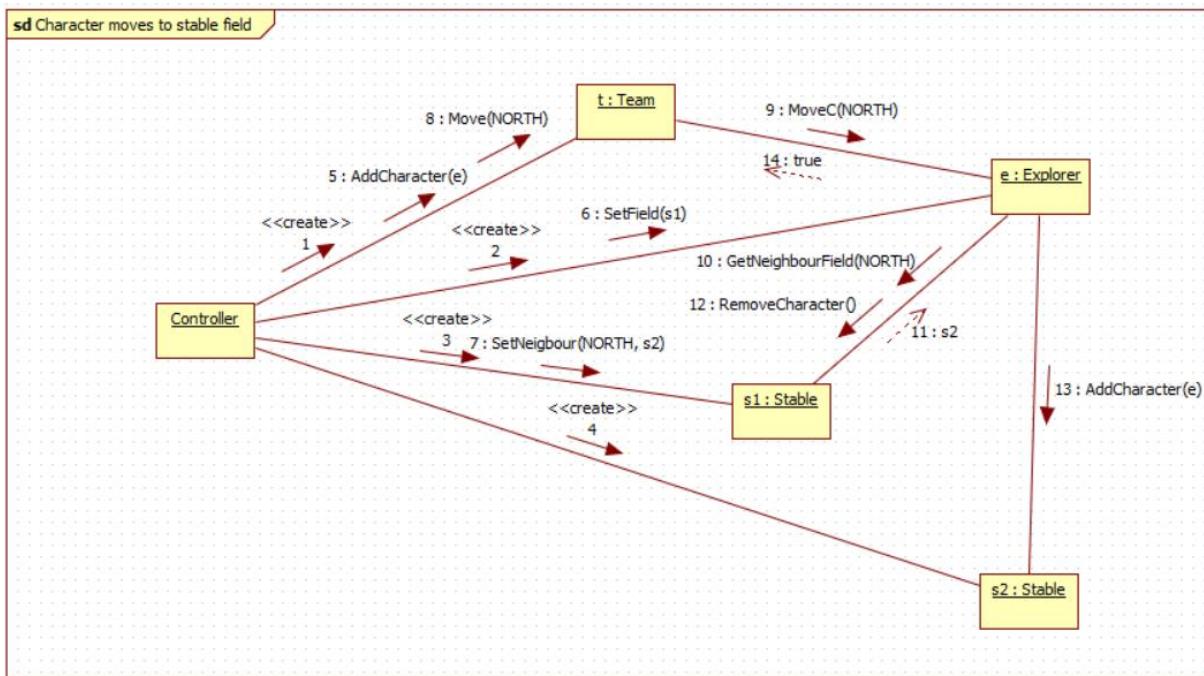
### 3.4.1. Blizzard on field without igloo



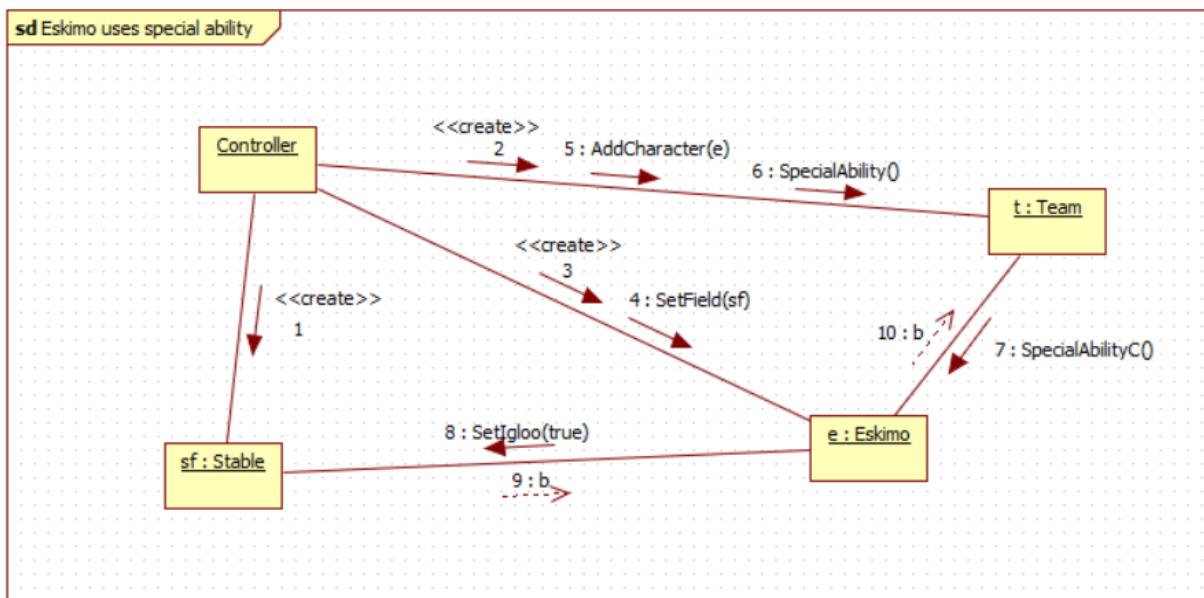
### 3.4.2. Blizzard on field with igloo



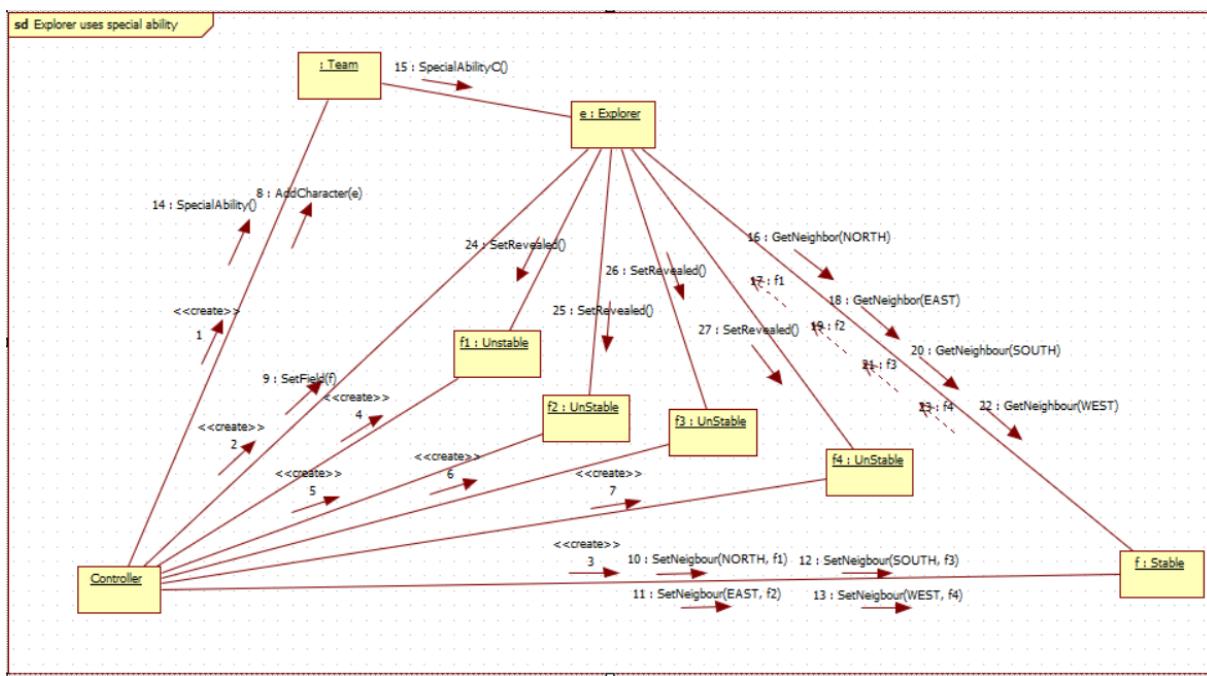
### 3.4.3. Character moves to stable field



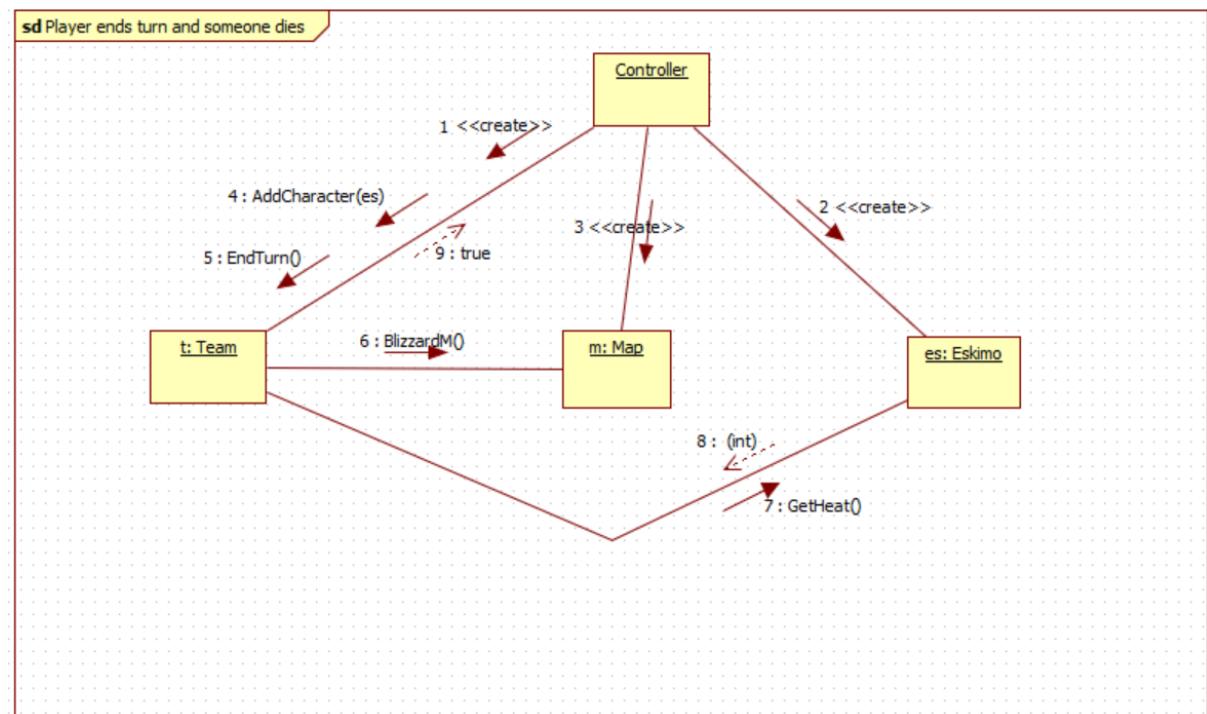
### 3.4.4. Eskimo uses special ability



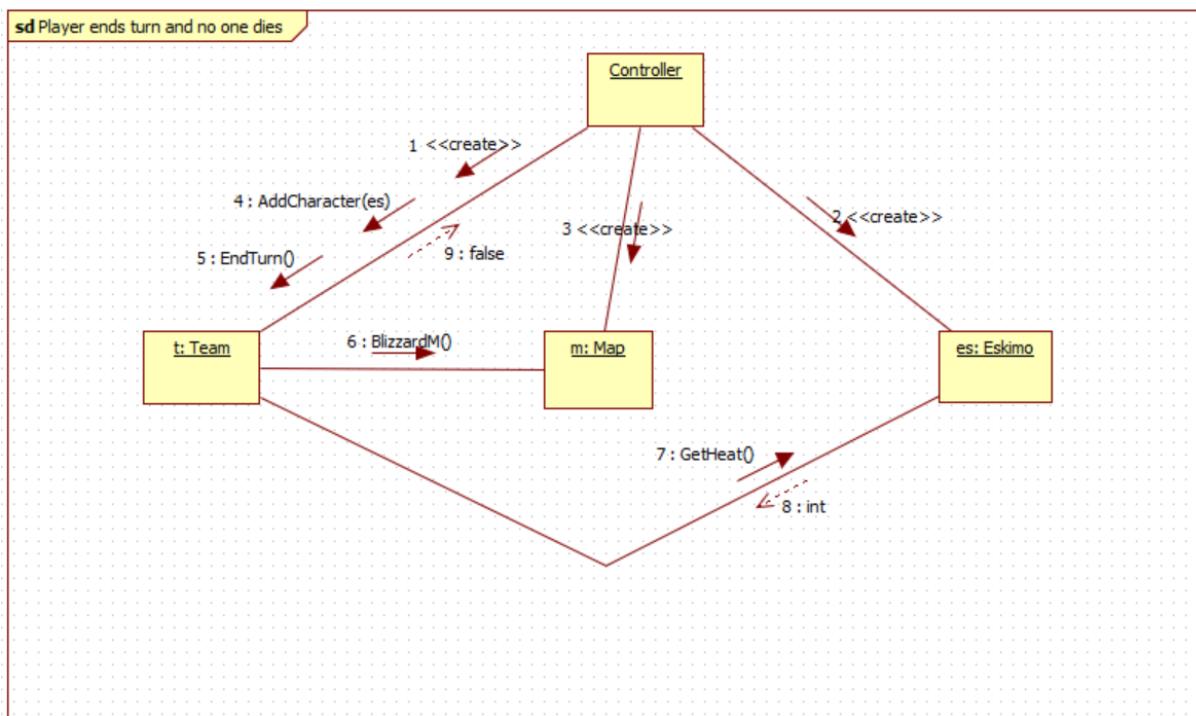
### 3.4.5. Explorer uses special ability



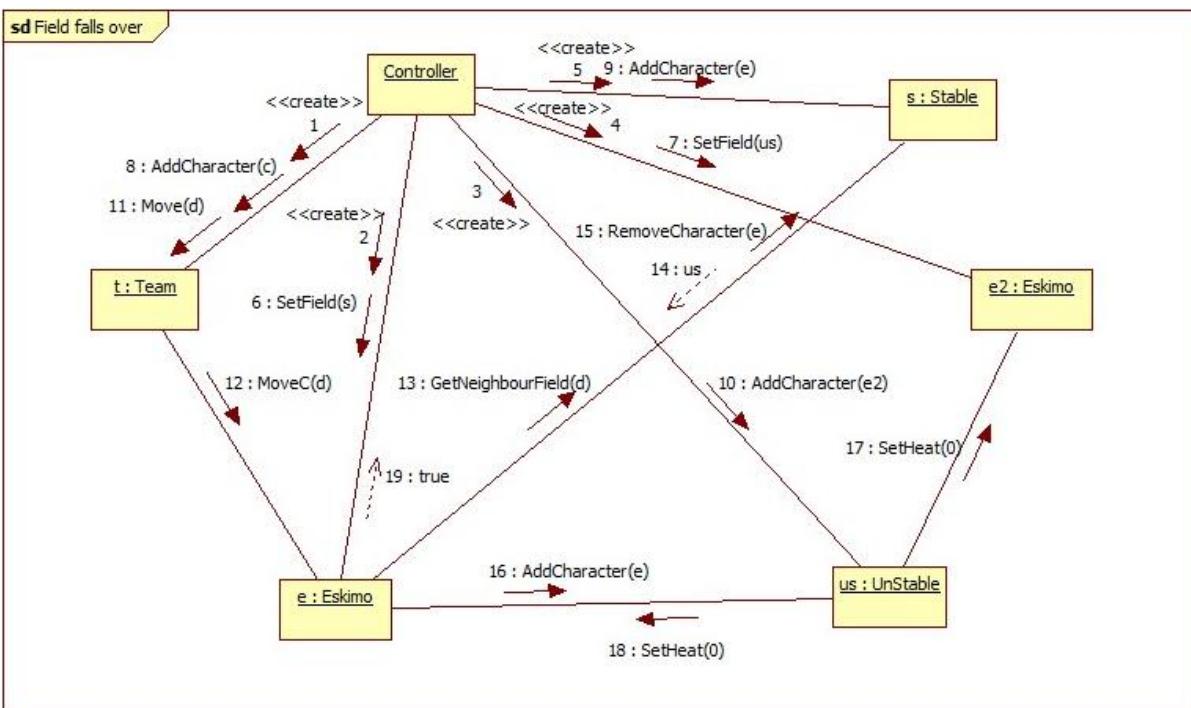
### 3.4.6. Player ends turn and someone dies



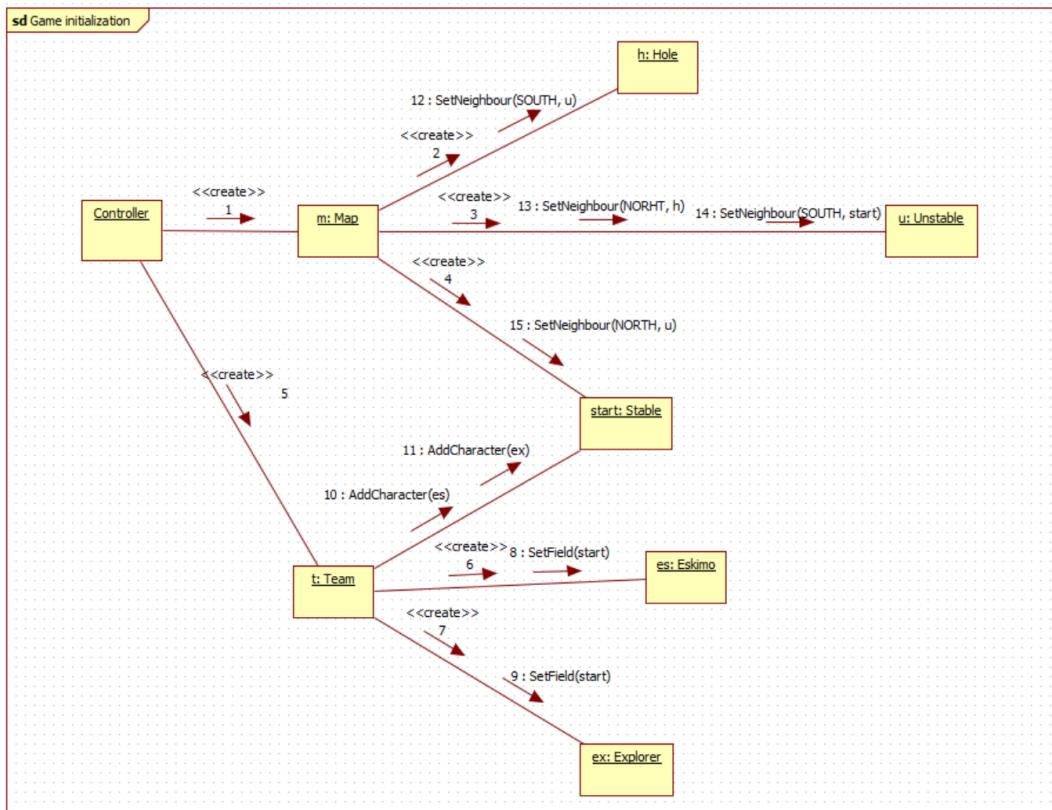
### 3.4.7. Player ends turn and no one dies



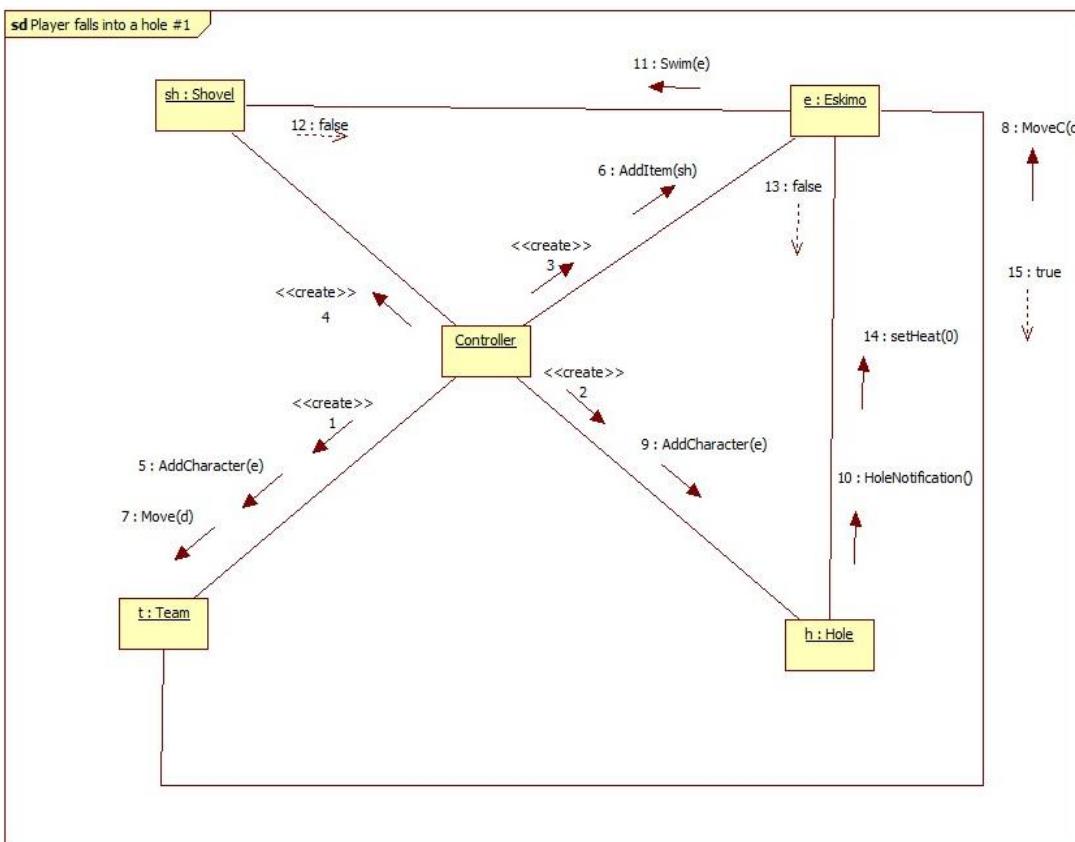
### 3.4.8. Field falls over



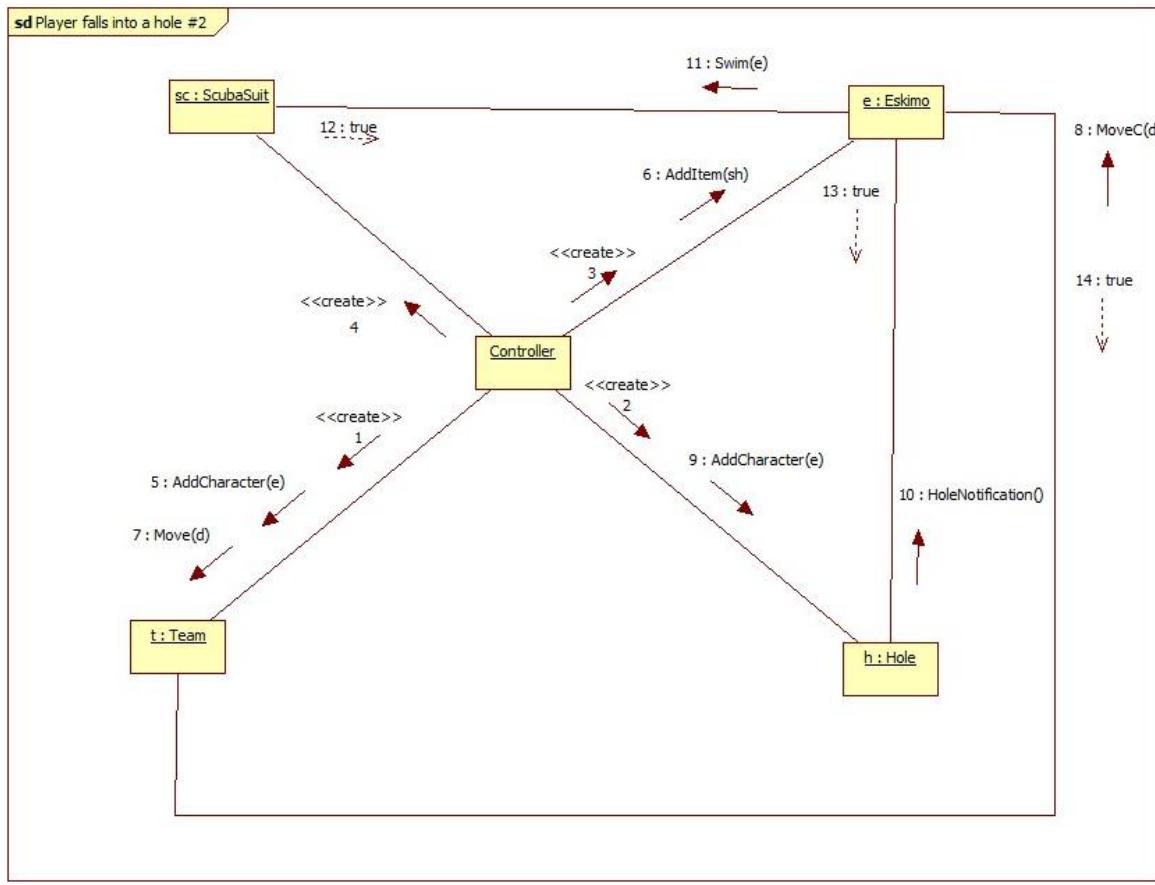
### 3.4.9. Game initialization



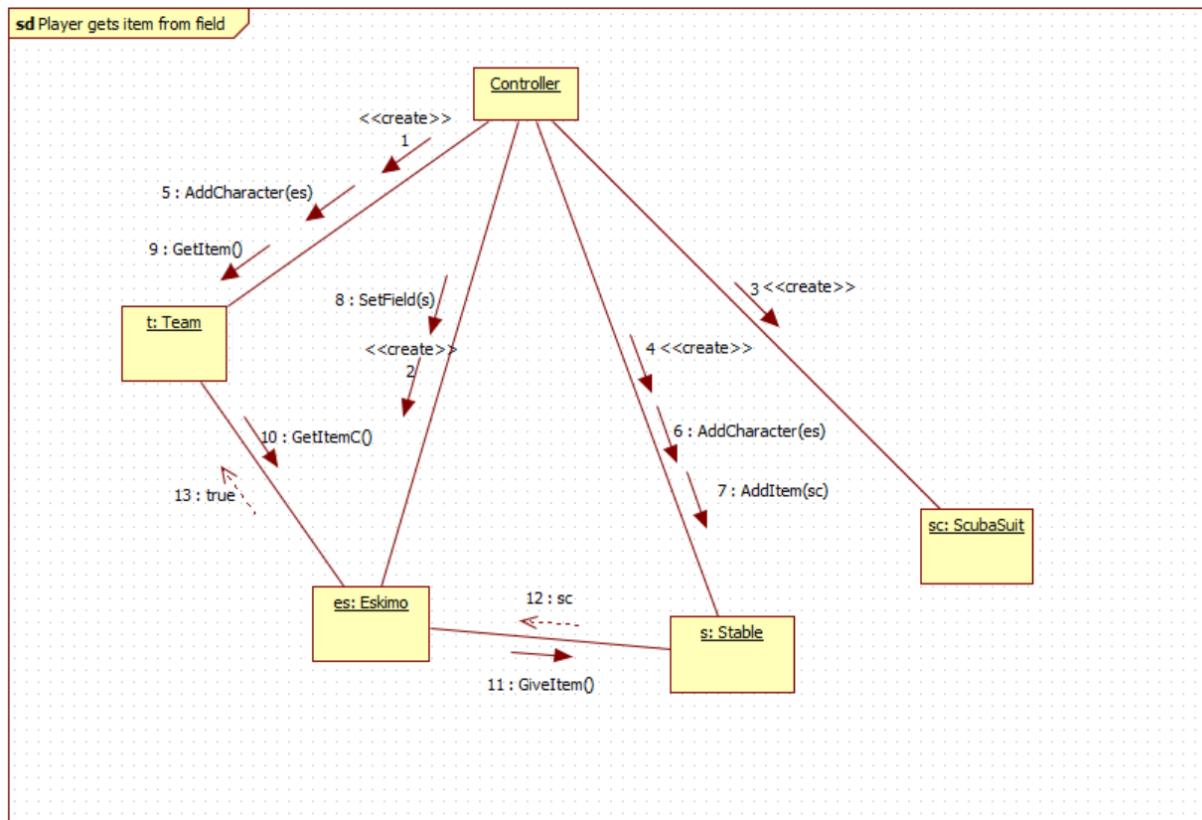
### 3.4.10. Player falls into a hole #1



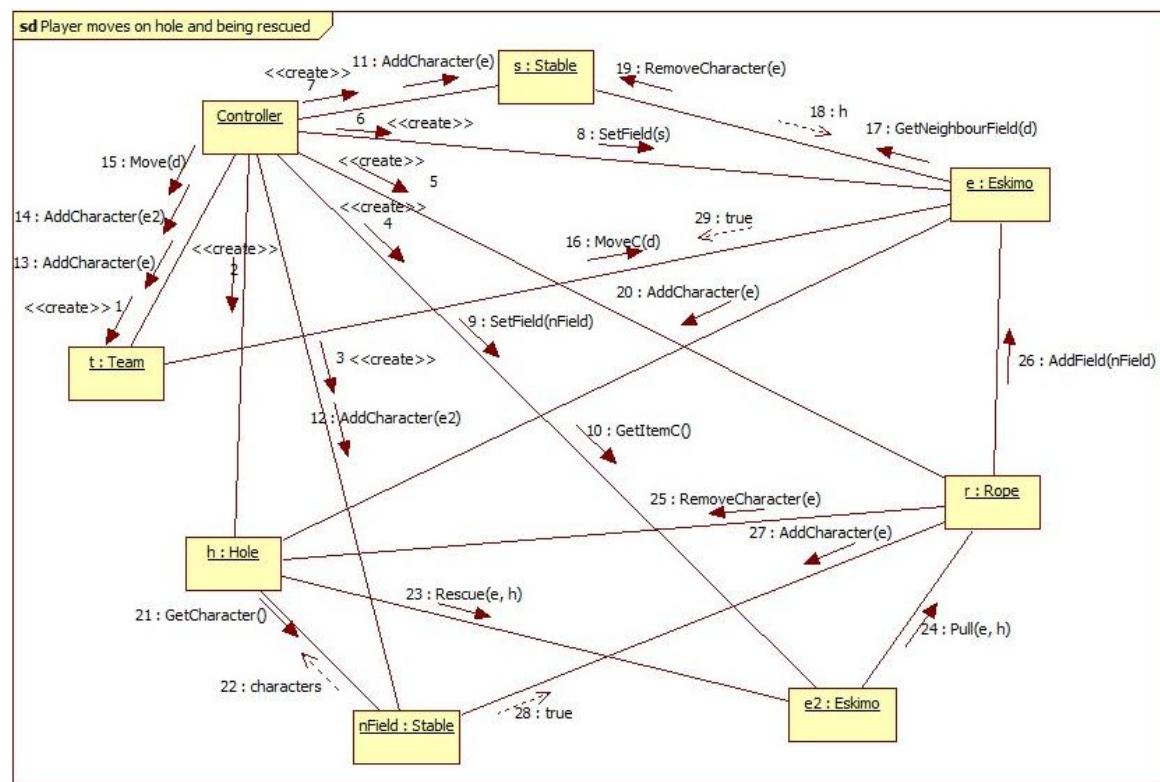
### 3.4.11. Player falls into a hole #2



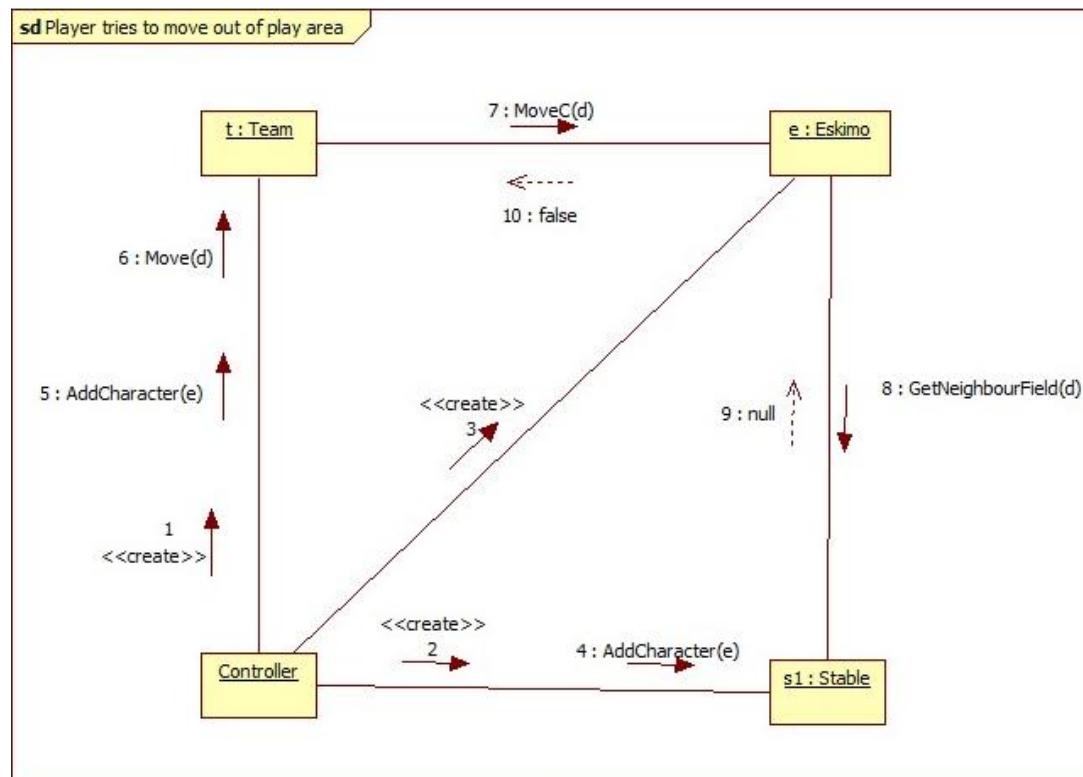
### 3.4.12. Player gets item from field



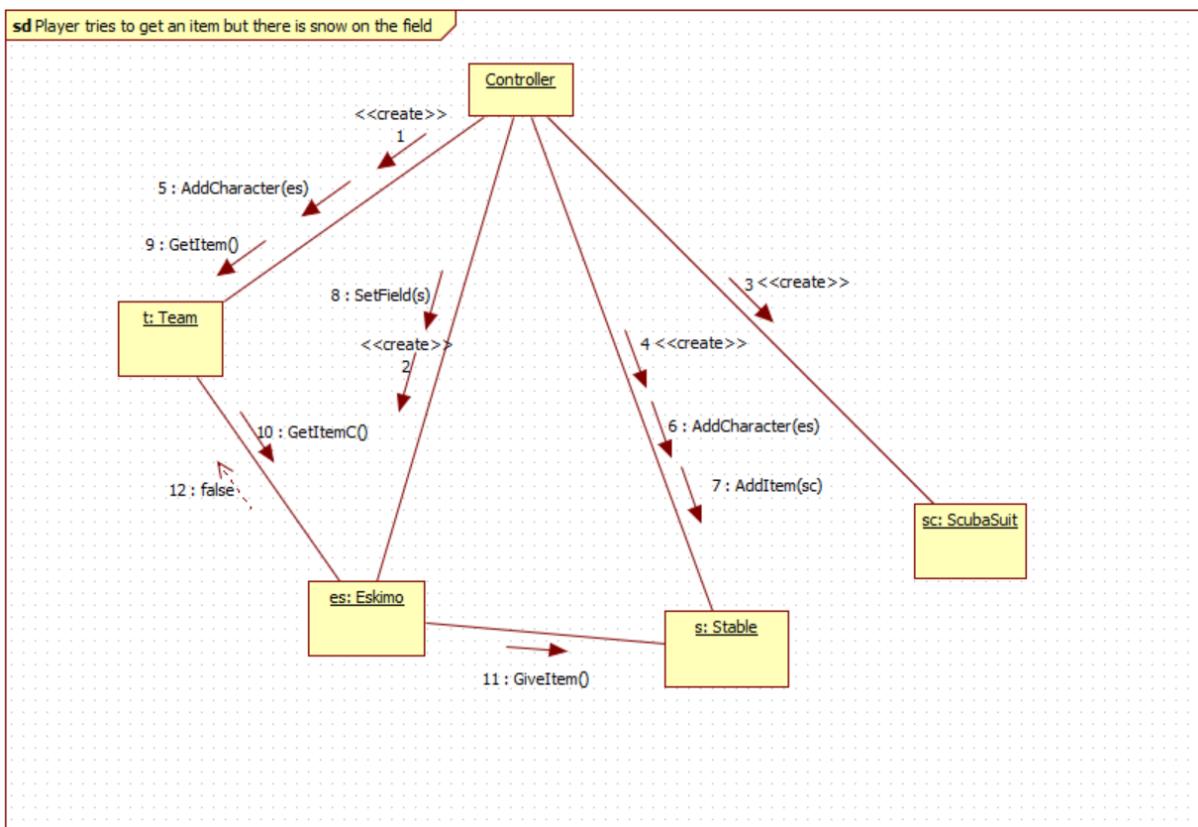
### 3.4.13. Player moves on hole and being rescued



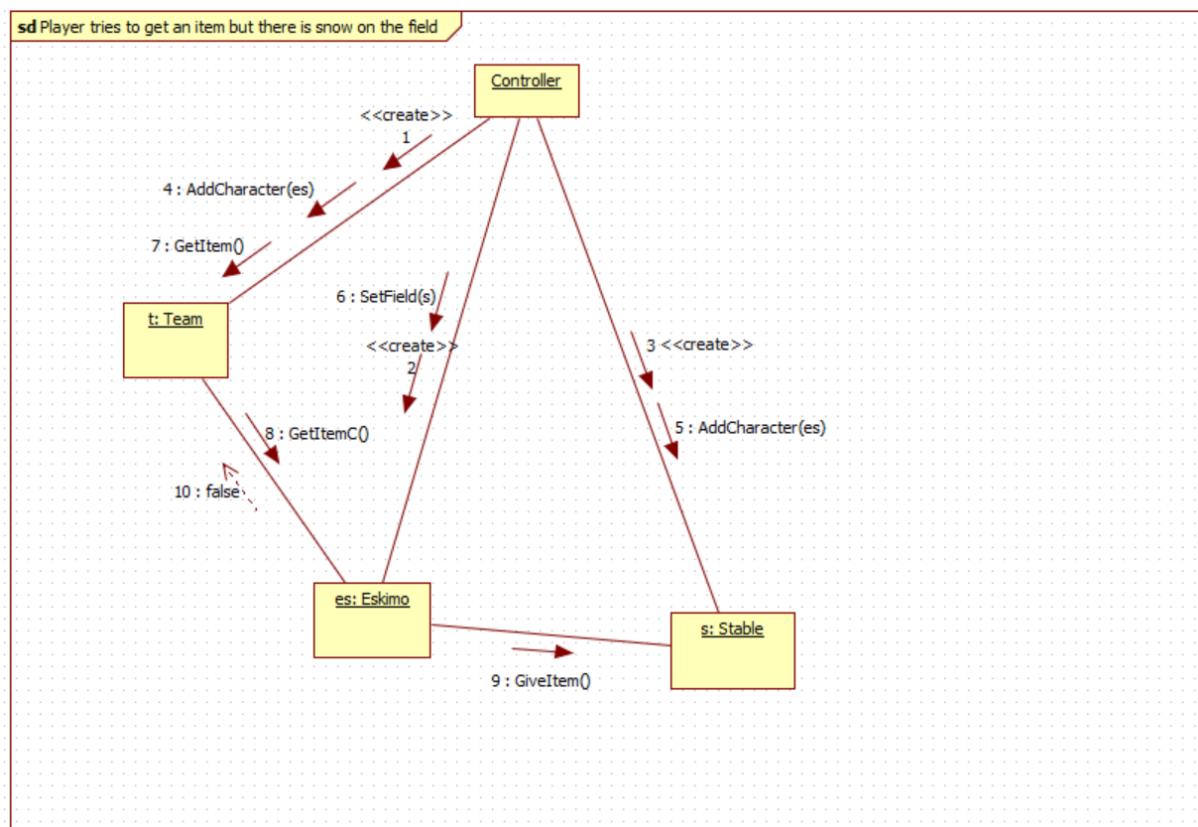
### 3.4.14. Player tries to move out of play area



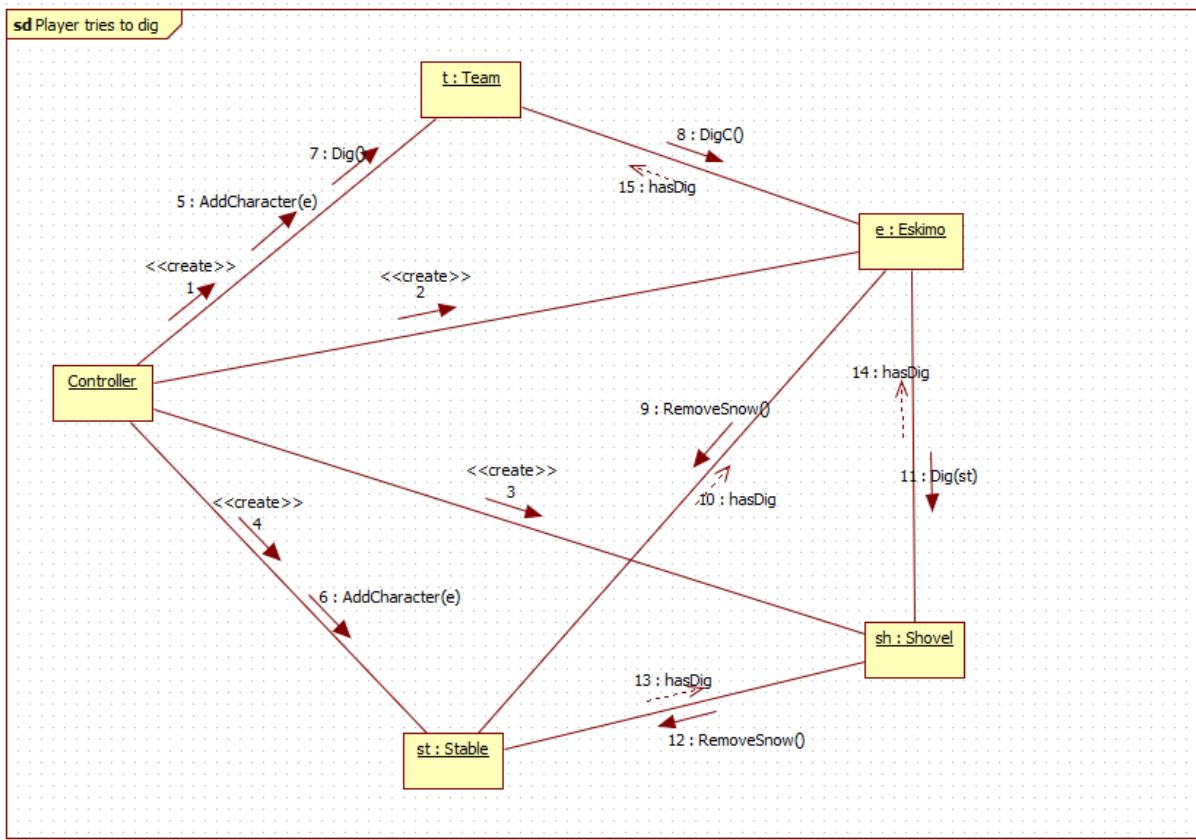
### 3.4.15. Player tries to get an item but there is snow on the field



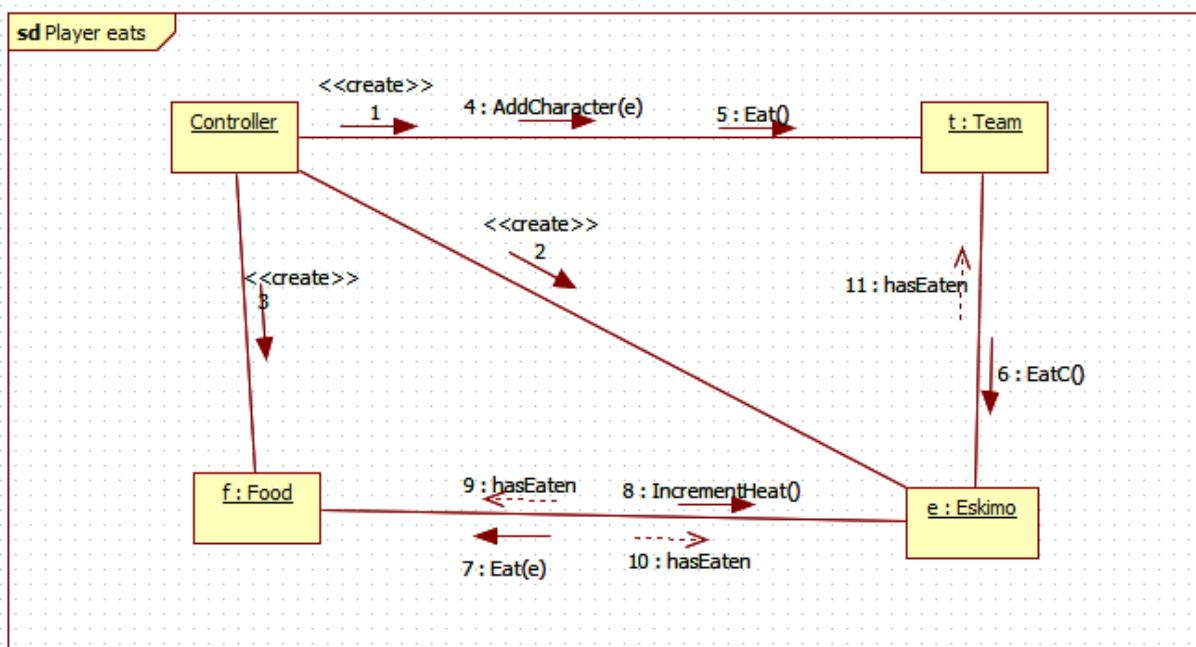
### 3.4.16. Player tries to get an item but there is no item on the field

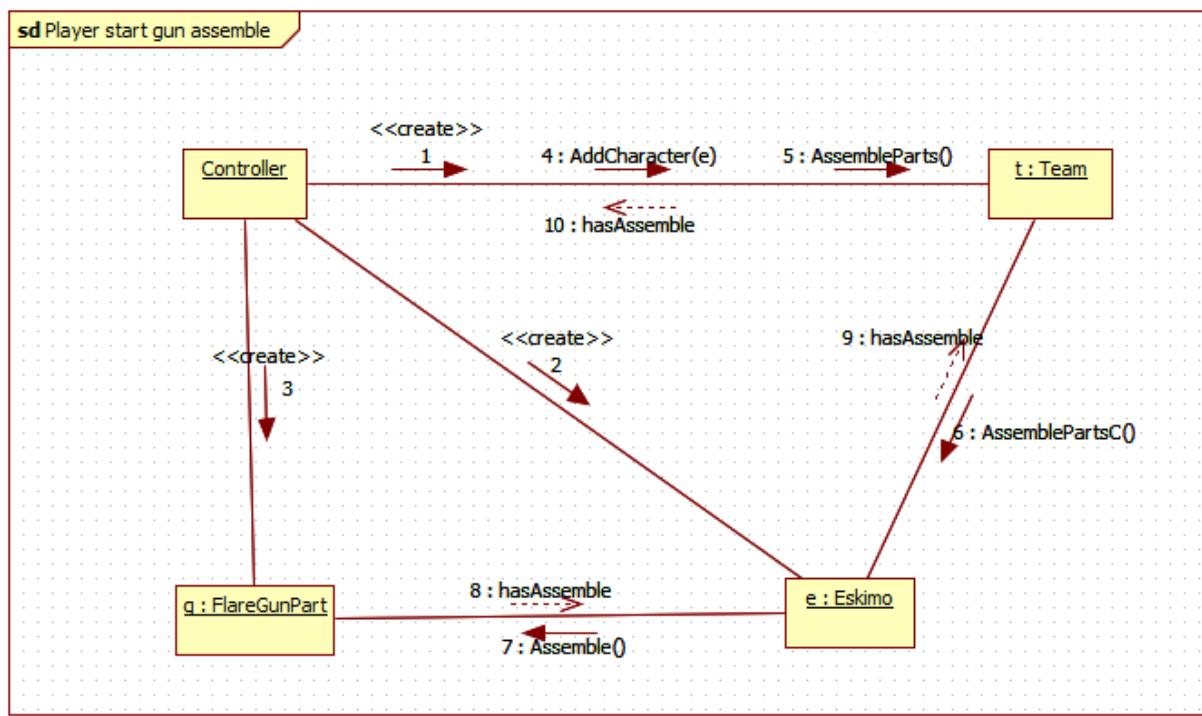


### 3.4.17. Player tries to dig



### 3.4.18. Player eats



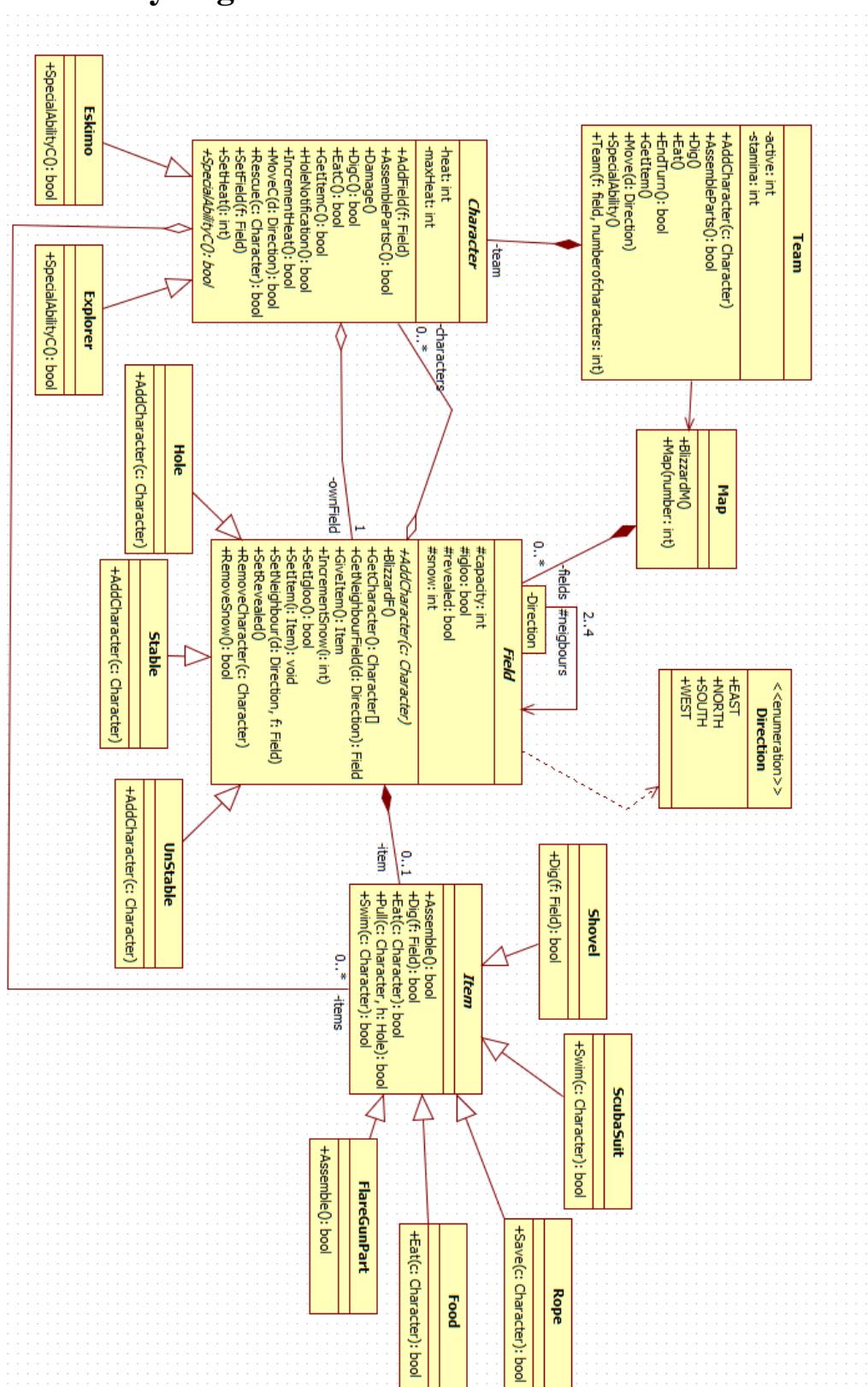
**3.4.19. Player start gun assemble**

## 5.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.03.15 14:00	1 óra	Kedves	Tevékenység: Kedves elkészít az eat, dig, assemble gun, end turn, get item special ability use-casek leírását
2020.03.15. 16:00	2 óra	Kurdi	Tevékenység: Kurdi elkészít a maradék use case-ek leírását és a use case diagramot
2020.03.17 18:00	1 óra	Háromi Kedves Holczer Kurdi Marton	Megbeszélés. DÖNTÉS: a csapat felosztja egymás között az elkészítendő diagramokat Hatóidő: 2020.03.18, 19:00
2020.03.18 10:00	1 óra	Kedves	Tevékenység: Kedves elkészít az ends turn, get item blizzard use-casek szekvencia diagramjait
2020.03.18. 12:00	2 óra	Háromi	Tevékenység: Háromi elkészít a Player moves on hole and rescued, Field falls over, Field doesn't fall over szekvencia- és a hozzá tartozó kommunikációs diagramokat.
2020.03.18 15:00	1 óra	Marton	Tevékenység: Marton elkészít a rá kiszabott szekvencia diagramokat (Player falls into hole and dies / swims out, Player tries to move out of play area) és a hozzájuk tartozó komm. diagrammokat.
2020.03.18. 17:00	1 óra	Holczer	Tevékenység: Holczer elkészít a rá kiszabott szekvenciákat (player start eat, start dig, start flare gun)
2020.03.18. 18:00	1 óra	Háromi Kedves Holczer Kurdi Marton	Megbeszélés: Az elkészített diagramok ellenőrzése.
2020.03.18. 20:00	1 óra	Háromi	Tevékenység: Háromi javítja a megbeszéltek szerint a Player moves on hole and rescued, Field falls over, Field doesn't fall over szekvencia- és a hozzá tartozó kommunikációs diagramokat.
2020.03.19 10:00	1,5 óra	Kedves	Tevékenység: Kedves kijavítja a szekvencia diagramjait és elkészít a hozzájuk tartozó kommunikációs diagramokat
2020.03.20. 15:00	1 óra	Kedves	Tevékenység:

			Kommunikációs diagramok frissítése, inicializáló use-case, szekvencia és kommunikációs diagram
2020.03.20. 16:00	2 óra	Kurdi	Tevékenység: Kurdi elkészíti a különleges képességek használatához és a stalib mezőre való mozgáshoz tartozó szekvencia- és kommunikációs diagramokat.
2020.03.20. 18:00	1,5 óra	Háromi	Tevékenység Háromi javítja a megbeszéltek szerint a Player moves on hole and rescued, Field falls over, Field doesn't fall over kommunikációs diagramokat, 5.2
2020.03.20. 18:00	2 óra	Holczer	Holczer elkészíti a rá kiszabott kommunikációs diagramokat és javítja a szekvencia diagramot(player start eat, start dig, start flare gun)
2020.03.20. 20:00	1 óra	Háromi Kedves Holczer Kurdi Marton	Megbeszélés. Döntés: Kommunikációs diagramok javítása.
2020.03.21. 15:00	1 óra	Marton	Tevékenység: Marton kijavítja a hibás kommunikációs diagramjait.
2020.03.22. 11:00	1 óra	Kedves	Tevékenység: Kedves letisztázza a dokumentumot

## A frissített osztálydiagram:



# 6 – Szkeleton beadás

## 45 – nullpointer

Konzulens:  
Ludmány Balázs

### Csapattagok

Háromi Bence	AK51GI	bence.haromi@gmail.com
Holczer Zsolt	XL9Y0E	zsolti@holczer.net
Kedves Áron Csanád	AV43UW	aron.kedves@gmail.com
Kurdi Barnabás	F0DJNW	barnabas.kurdi@gmail.com
Marton Gergely	FC2WNS	martongergely11@gmail.com

## 6. Szkeleton beadás

### 6.1 Fordítási és futtatási útmutató

#### 6.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Character.java	5,613 byte	2020.03.29 8:21 PM	A karakter osztály.
Eskimo.java	1,143 byte	2020.03.29 8:08 PM	Az eszkimó osztály.
Explorer.java	1,364 byte	2020.03.29 8:08 PM	Az explorer osztály.
Direction.java	164 byte	2020.03.29 8:08 PM	A direction enum.
Field.java	5,411 byte	2020.03.29 7:30 PM	A field osztály.
Hole.java	2,332 byte	2020.03.29 8:12 PM	A lyuk osztály.
Stable.java	892 byte	2020.03.29 8:08 PM	A stabil mező osztály.
UnStable.java	1278 byte	2020.03.29 8:12 PM	A instabil mező osztály.
FlareGunPart.java	1,229 byte	2020.03.29 8:13 PM	A pisztoly alkatrész osztály.
Food.java	830 byte	2020.03.29 8:13 PM	A élelmiszer osztály.
Item.java	2,631 byte	2020.03.29 8:23 PM	A tárgy absztrakt osztály.
Rope.java	978 byte	2020.03.29 8:14 PM	A kötél osztály.
ScubaSuit.java	710 byte	2020.03.29 8:14 PM	A búvárruha osztály.
Shovel.java	848 byte	2020.03.29 8:14 PM	Az ásó osztály.
Main.java	29,469 byte	2020.03.29 8:08 PM	A teszteket tartalmazó osztály és a program belépési pontja.
Map.java	1170 byte	2020.03.29 8:15 PM	A térkép osztály.
Skeleton.java	2,223 byte	2020.03.29 7:30 PM	A kiírásáért felelős singelton osztály.
Team.java	4,635 byte	2020.03.29 8:12 PM	A csapat osztály.

#### 6.1.2 Fordítás

##### 6.1.2.1. Navigálunk el a mappához cd parancs ismételt használata segítségével és lépjünk be a szketelon mappába

*cd szkeleton*

##### 6.1.2.2. Menjünk be az src mappába

Ebben a mappában találhatók a szkeleton java file-jai packagegenként mappákba rendezve.

*cd src*

##### 6.1.2.3. Hozzunk létre az src mappán belül egy teszt mappát

A mappára ezért van szükség, hogy ne ugyanazokban a mappákban legyenek a forrás fájlok, mint a fordított fájlok. Ez kényelmes lehet, ha később szeretnénk mozgatni a forrás fájljainkat.

*mkdir teszt*

#### 6.1.2.4. Fordítsuk le a programot, úgy, hogy a teszt mappát választjuk a fájlok tárolására

Amennyiben a JRE verzióval van problémája a rendszernek a target és source beállításával orvosolható.

```
javac character/*.java direction/*.java field/*.java item/*.java main/*.java  
map/*.java skeleton/*.java team/*.java -d teszt -h teszt
```

#### 6.1.2.5. Menjünk be a teszt mappába

Itt már lefordítva vár minket a projekt. A további lépéseket a Futtatás alatt részleteztem.

```
cd teszt
```

### 6.1.3 Futtatás

#### 6.1.3.1. A program elindítása

Ha a fordítás lépései pontosan követtük, akkor most a teszt mappában kell lennünk. Itt a main package-ben található Main fordított class indításával el is indul a programunk.

```
java main.Main
```

#### 6.1.3.2. A program inputja

A programunk időnként kér inputot tölünk, hogy eldöntse mi fog történni. A kezdeti menüből faszerűen ágaznak le forgatókönyvek. A billentyűzeten adott az elvártnak megfelelő érték megadása után az Enter billentyű lenyomásával adhatjuk meg a válaszunkat a programnak. A megadott értékek segítségével állíthatjuk be az általunk tesztelni kíván forgatókönyvet. Két fajta kérdést kaphatunk. Elsősorban menükben választhatunk forgatókönyvet, másodsorban egyszerűbb értékekre feltett kérdésekre válaszolhatunk.

#### 6.1.3.3. A program outputja

A program csak a képernyőn keresztül ad outputot, a Command Promptban történő szövegek kiírása segítségével. A program menüket ír ki választási lehetőségekkel, valamint a végrehajtás során kiírásokat hajt végre a különböző konstruktor és egyéb függvények hívása során, hogy megmutassa a függvények kommunikációját és hatását.

#### 6.1.3.4. A program bezárása

A program minden forgatókönyv végén és nem megfelelő input esetén visszatér a kezdeti menübe. A főmenü tartalmaz egy "8. Kilepes" menüpontot, ennek választása esetén a program futása véget ér.

## 6.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Háromi Bence	AK51GI	21%
Holczer Zsolt	XL9Y0E	17,5%
Kedves Áron Csanád	AV43UW	22%
Kurdi Barnabás	F0DJNW	22%
Marton Gergely	FC2WNS	17,5%

## 6.3 Napló

Kezdet	Időtartam	Résznevők	Leírás
2020.03.26. 15:00	1 óra	Háromi Kedves Holczer Kurdi Marton	Értekezlet. Döntés: a csapat eldönti, hogy ki melyik osztályt készítse el. Határidő: 2020.03.27 14:00
2020.03.26. 17:00	3 óra	Marton	Tevékenység: Marton elkészíti a számára kiosztott osztályokat (Field, Hole, Stable, Unstable)
2020.03.26. 18:00	1 óra	Kurdi	Tevékenység: Item és Itemből leszármazó osztályok elkészítése
2020.03.26. 18:00	1,5 óra	Háromi	Tevékenység: Team és Map osztályok elkészítése.
2020.03.27. 10:00	1,5 óra	Holczer	Tevékenység: Character és a leszármazott osztályok elkészítése
2020.03.27 14:00	2 óra	Háromi Kedves Holczer Kurdi Marton	Értekezlet: A csapat javításokat végez az elkészített osztályokon
2020.03.27 15:00	1 óra	Marton	Tevékenység: Marton javításokat végez az osztályain.
2020.03.27. 16:00	1 óra	Háromi	Tevékenység: Osztályok javítása.
2020.03.27 17:00	4 óra	Kedves	Tevékenység: Az osztályok tisztázása a tesztírás kezdéséhez Main teszeset kezelő részének írása
2020.03.28 10:00	3 óra	Kedves	Tevékenység: A teszeset kezelő rendszer befejezése
2020.03.28. 11:00	5 óra	Kurdi	Tevékenység:

			Összes osztály tisztázása, függvényhívások rendezése, skalárok kiiktatása
2020.03.29 17:00	1,5 óra	Kedves	Tevékenység: A projekt CommandPrompt-ban való működésének megoldása A dokumentáció megírása
2020.03.29 17:00	3 óra	Holczer	Tevékenység: szcenáriók áttekintése és lyukba mozgás javítása
2020.03.29. 18:00	2 óra	Kurdi	Tevékenység: Osztályok kommentelése, szcenáriók áttekintése
2020.03.29. 18:00	2 óra	Háromi	Kommentelés, hibák javítása.
2020.03.29 18:00	1 óra	Marton	Tevékenység: szenáriók kommentelése és javítása.
2020.03.29 20:00	1 óra	Kedves	Tevékenység: Dokumentáció tisztázása

# 7. Prototípus koncepciója

45 – nullpointer

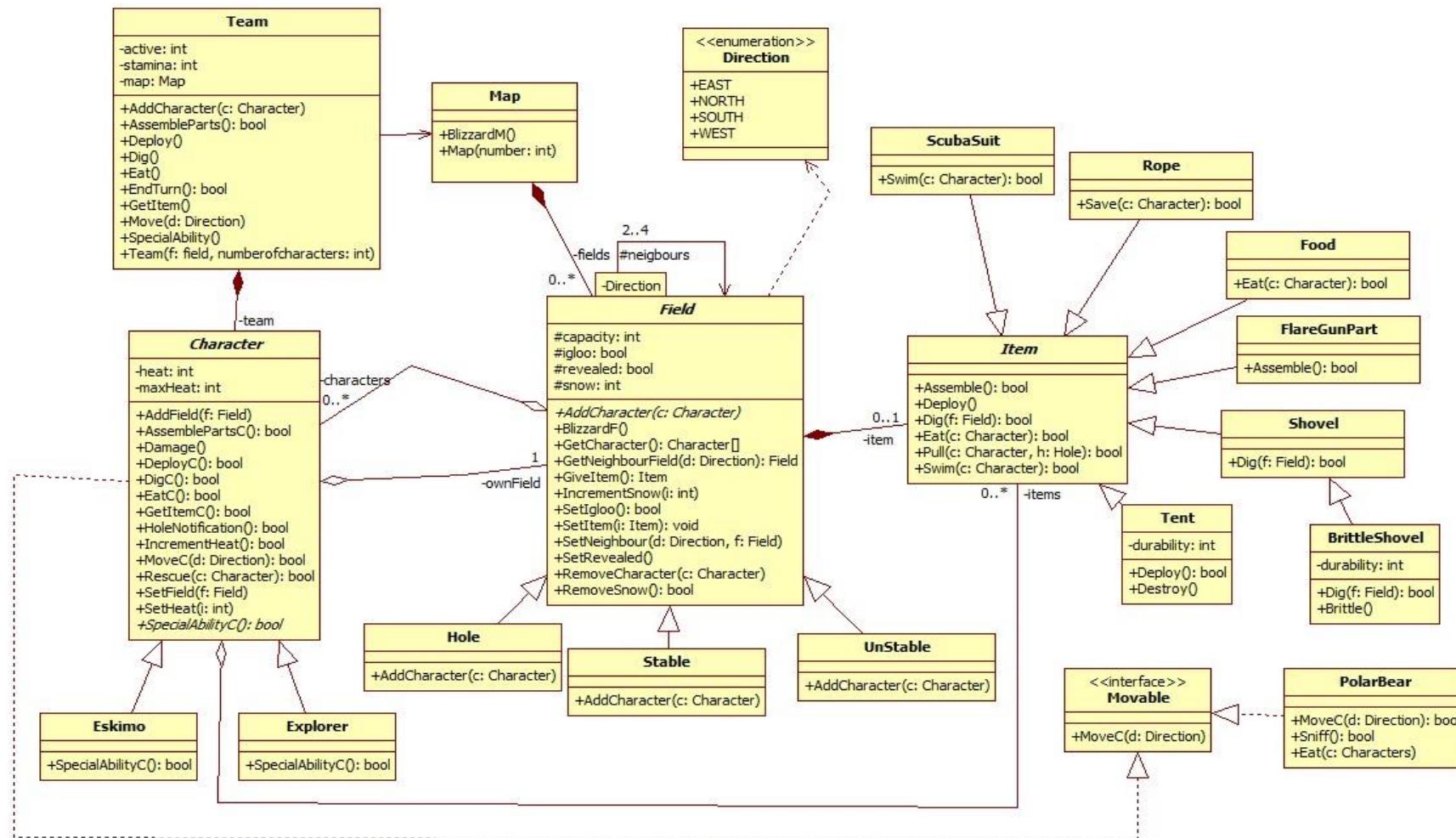
Konzulens:  
Ludmány Balázs

## Csapattagok

Háromi Bence	AK51GI	bence.haromi@gmail.com
Holczer Zsolt	XL9Y0E	zsolti@holczer.net
Kedves Áron Csanád	AV43UW	aron.kedves@gmail.com
Kurdi Barnabás	F0DJNW	barnabas.kurdi@gmail.com
Marton Gergely	FC2WNS	martongergely11@gmail.com

## 7.0 Változás hatása a modellre

### 7.0.1 Módosult osztálydiagram



## 7.0.2 Új vagy megváltozó metódusok

### Változott osztályok vagy metódusok:

#### Character:

- **Változás:** A karakter most már megvalósítja a Movable interfésszt (lásd lejjebb), így a bool MoveC(Direction d) metódusa innen származik, azonban a működés változatlan.
- **Metódusok:**
- **bool DeployC():** A játékos sátorállítását kezdeményezi.

#### Team:

- **Ősosztályok:** -
- **Felelősség:**
  - A karakterek és a játék interakciót kezelő osztály, a játékos közvetlenül ezzel áll kapcsolatban.
- **Attribútumok:**
  - **-team: Character[]:** A csapatot reprezentáló karakterek tömbje.
  - **-active: int:** Az aktív játékos indexe.
  - **-stamina: int:**
- **Metódusok:**
  - **void AddCharacter(Character c):** Hozzáad egy játékosat a csapathoz.
  - **bool AssembleParts():** A pisztoly összeszerelését elvégző függvény. Visszatérése a művelet sikereségétől függ.
  - **void Deploy():** A játékos sárat állít műveletet kezelő metódus.
  - **void Dig():** A játékos ásás műveletet kezelő metódus.
  - **void Eat():** A játékos evés műveletet kezelő metódus.
  - **bool EndTurn():** A kör befejezését irányító metódus. <Visszatérés?>
  - **void GetItem():** A mezőről való tárgyfelvétel elindítását végzi.
  - **void Move(Direction d):** A mozgatható karakterek / NPC-k (Non Playable Character, azaz Nem Játszható Karakter) mozgását végző metódus.
  - **void SpecialAbility():** A karakterek spec. képességét irányító metódus.

### Új osztályok:

#### BrittleShovel:

- **Ősosztályok:** Item → Shovel → BrittleShovel
- **Felelősség:**
  - A törékeny ásót kezelő osztály. Ugyanúgy működik, mint az ősosztálya (Shovel), leszámítva azt, hogy 3 használat után használhatatlanná válik.
- **Attribútumok:**
  - **-durability: int:** Az ásó “életét” reprezentálja. Három használat után 0-ra csökken. Kezdeti értéke tehát 3.
- **Metódusok:**
  - **bool dig(Field f):** Az ásást elvégző művelet. Ha A művelet sikeres true-val, ha nem false-szal tér vissza. Mindent hívást követően csökkenti a durability-t.
  - **void Brittle():** Az ásó összetörését kezelő függvény, amely akkor hívódik, ha az ásó durability-je 0-ra csökken.

**Tent:**

- **Ősosztályok:** Item → Tent
- **Felelősség:**
  - A sátrakat kezelő osztály. A sátor is egy kiátható tárgy, amely védelmet nyújt a hóviharok elől, azonban egy kör után eltűnik.
- **Attribútumok:**
  - **-durability: int:** A sátor “életét” reprezentálja. Az elhelyezést követően 1 kör után 0-ra csökken, így a sátor eltűnik.
- **Metódusok:**
  - **bool Deploy():** A sátor felállítását elvégző függvény. Amennyiben a mezőn már van egy építmény (tehát iglu vagy egy sátor), akkor false-al tér vissza, tehát a felállítás nem sikeres. Olyan mező esetén, ami építés szempontjából üres a visszatérés true.
  - **void Destroy():** A sátor eltűnésséért felelős metódus, amely a sátor lerakását követően hívódik.

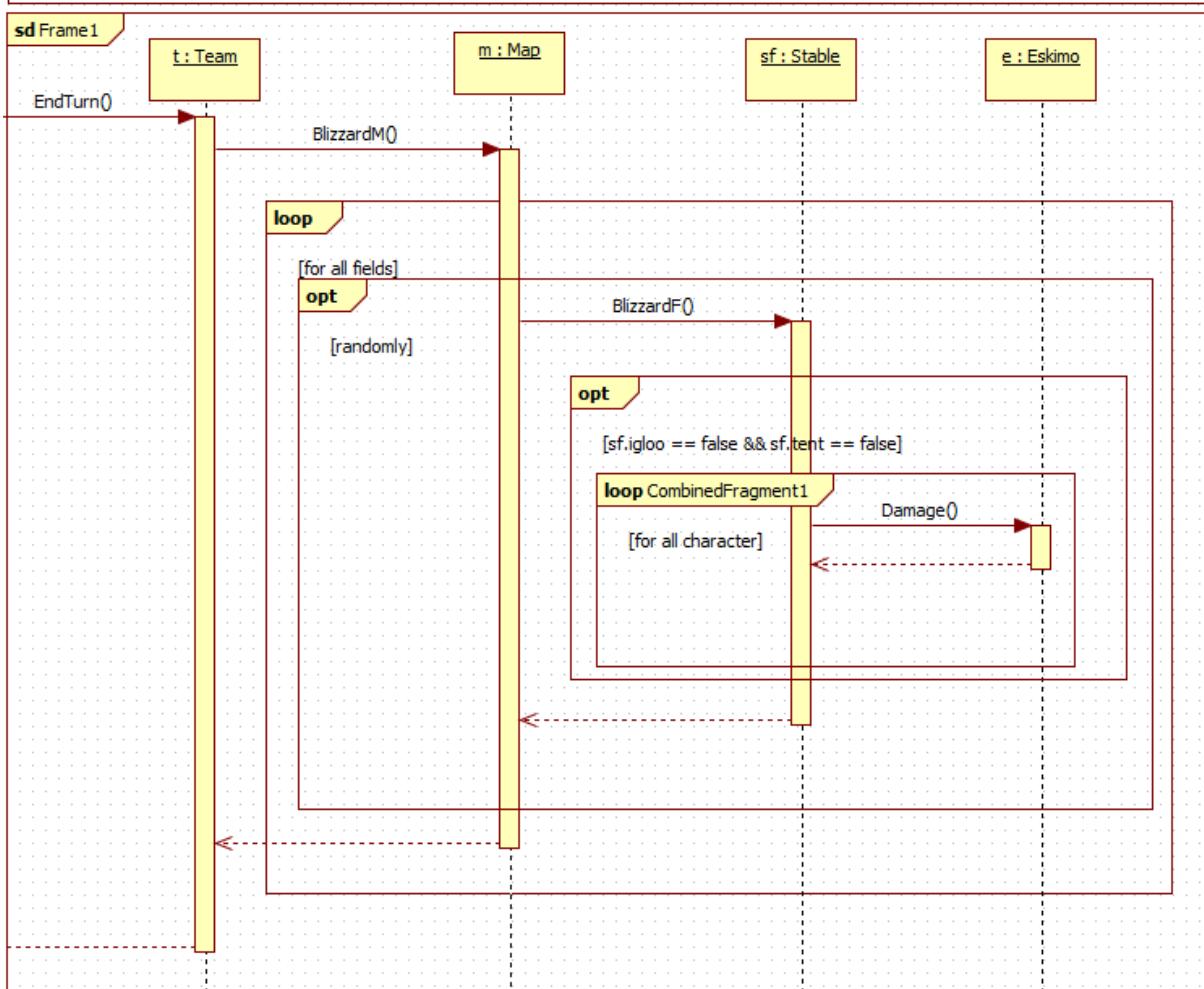
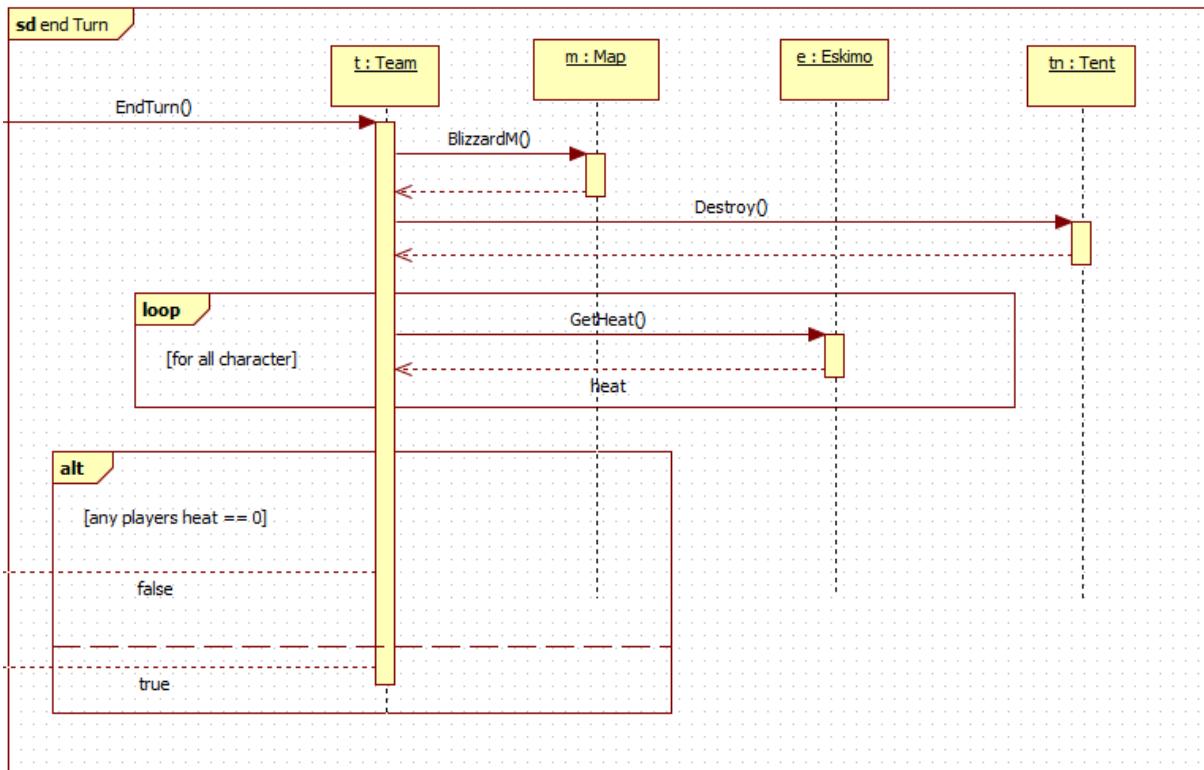
**PolarBear:**

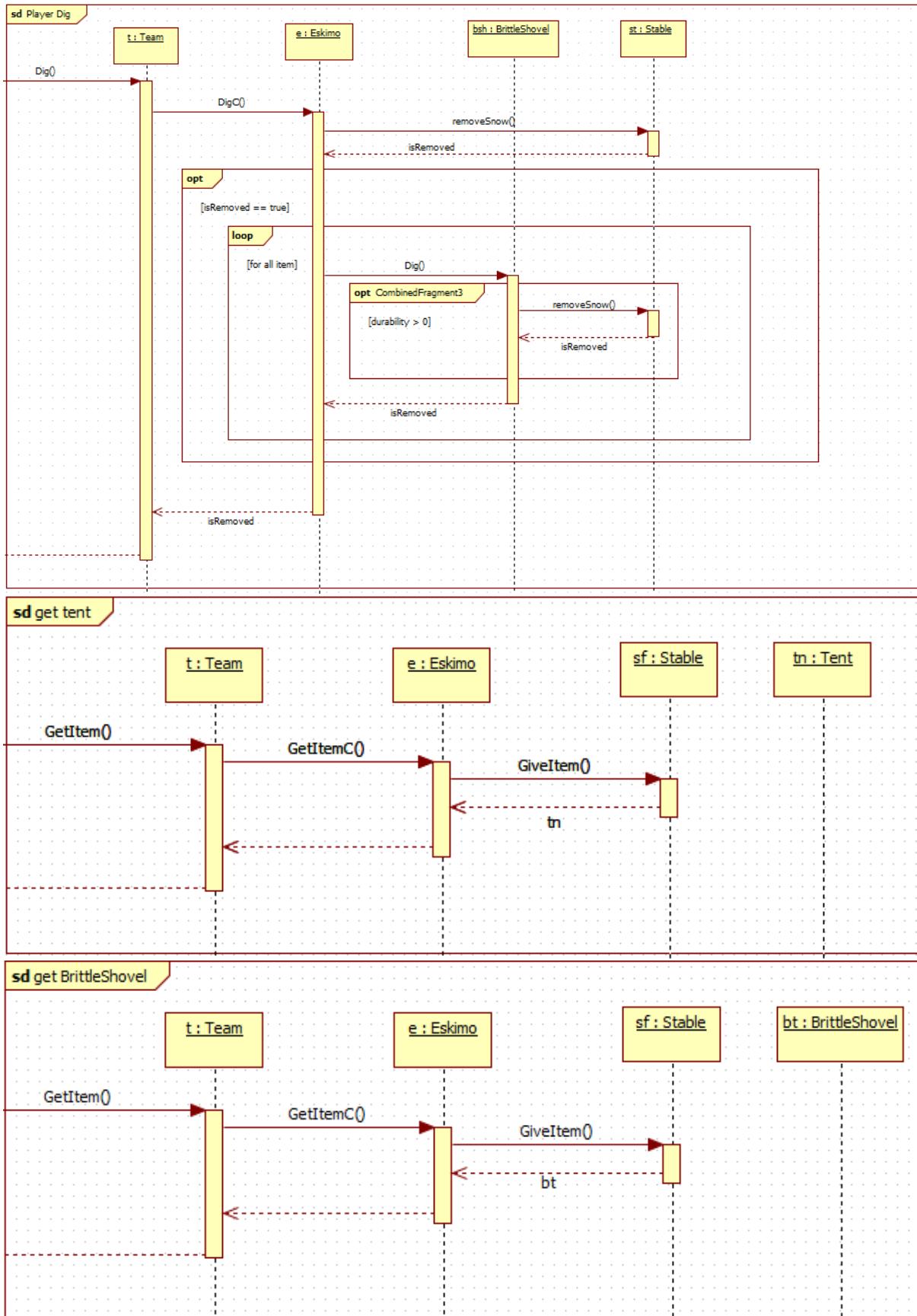
- **Felelősség:** A jegesmedvéket kezelő osztály, amely kezeli azok mozgását, és interakcióit a játékosokkal.
- **Attribútumok:**
  - -
- **Metódusok:**
  - **bool MoveC(Direction d):** (Moveable interfész metódusa) A medve lép egy d irányba. Ha sikerült elmozdulnia true-val tér vissza egyébként pedig false-sal.
  - **bool Sniff():** A medve minden mezőre lépéskor körbeszaggalászik, hogy tartózkodik-e karakter a mezőn. Amennyiben a karakterek igluban vannak, úgy nem érzi meg őket, és a visszatérés hamis, egyéb esetben pedig kiszagolja őket és a visszatérés igaz.
  - **void Eat(Characters c):** Ha a medve kiszagolta a karaktereket, akkor az elfogyasztásukat ezzel a függvénnyel végzi el.

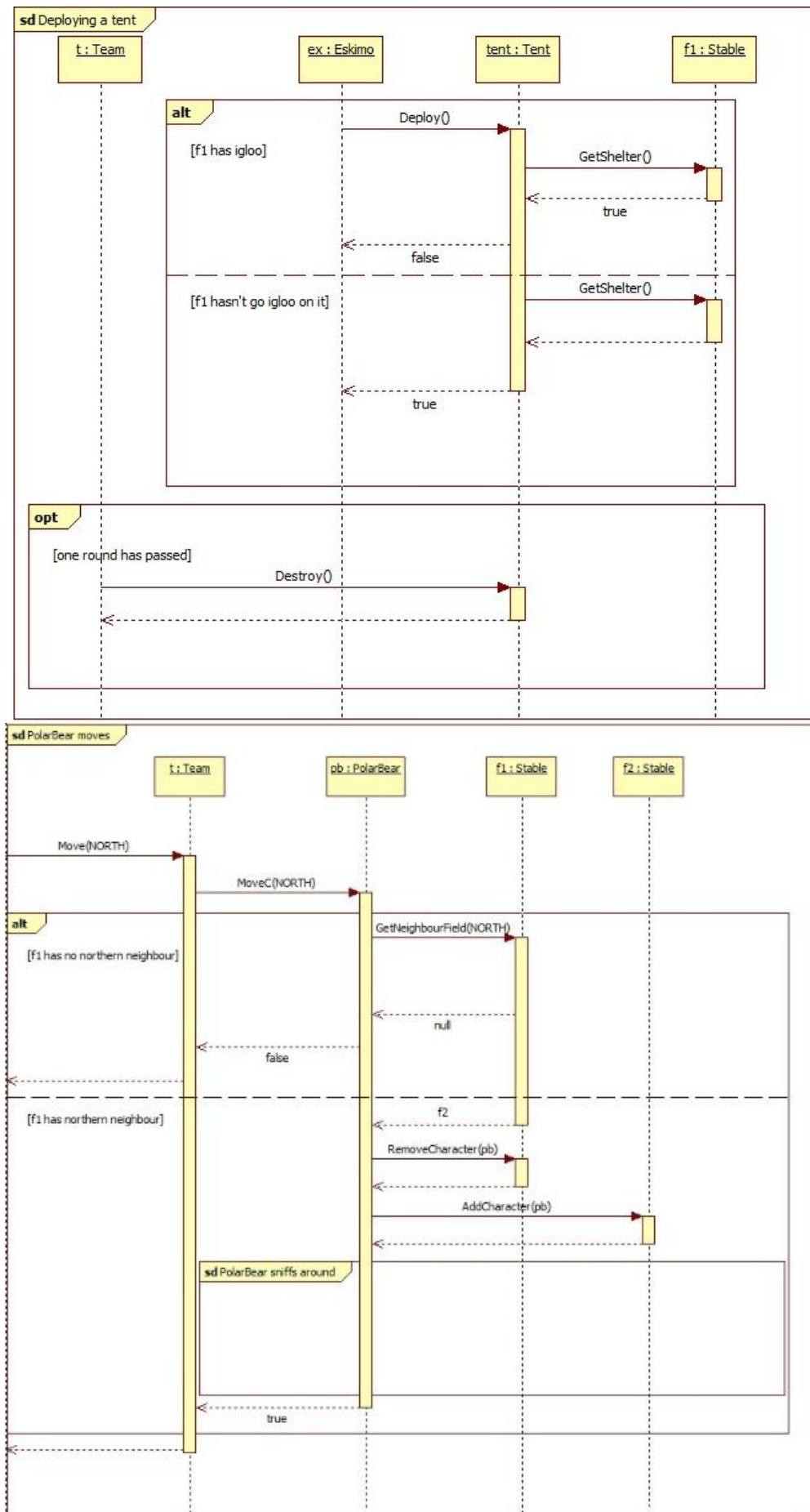
**Movable interfész:**

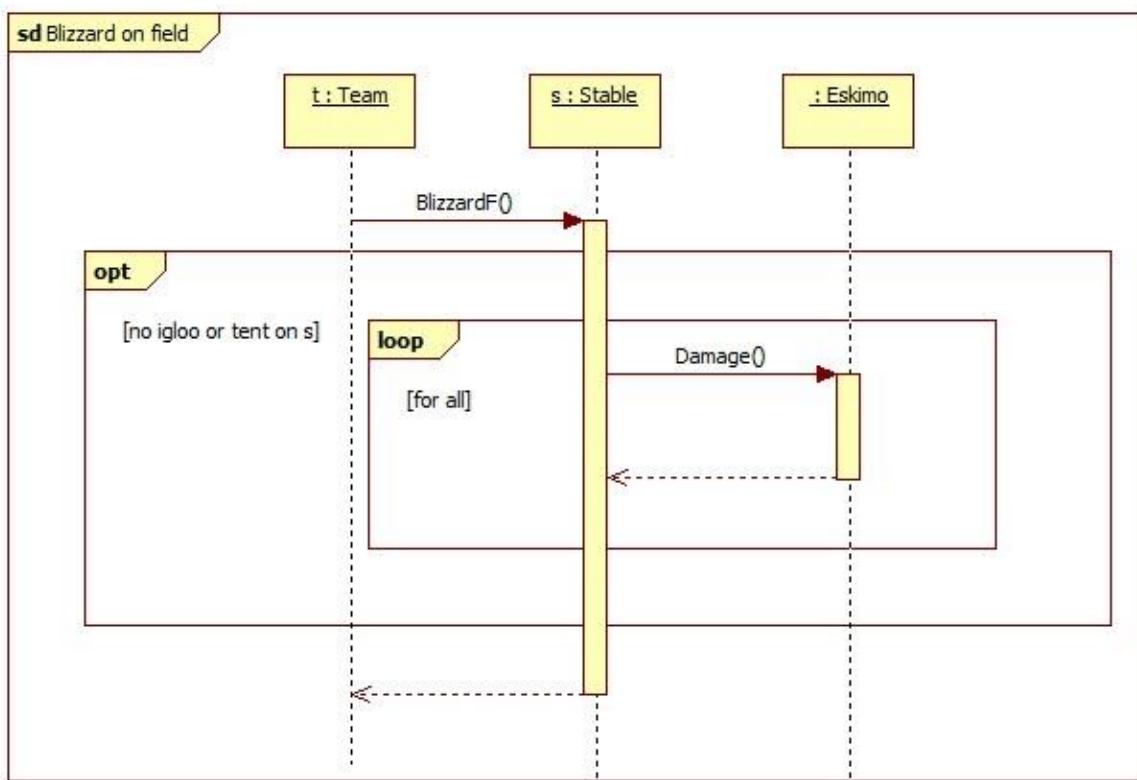
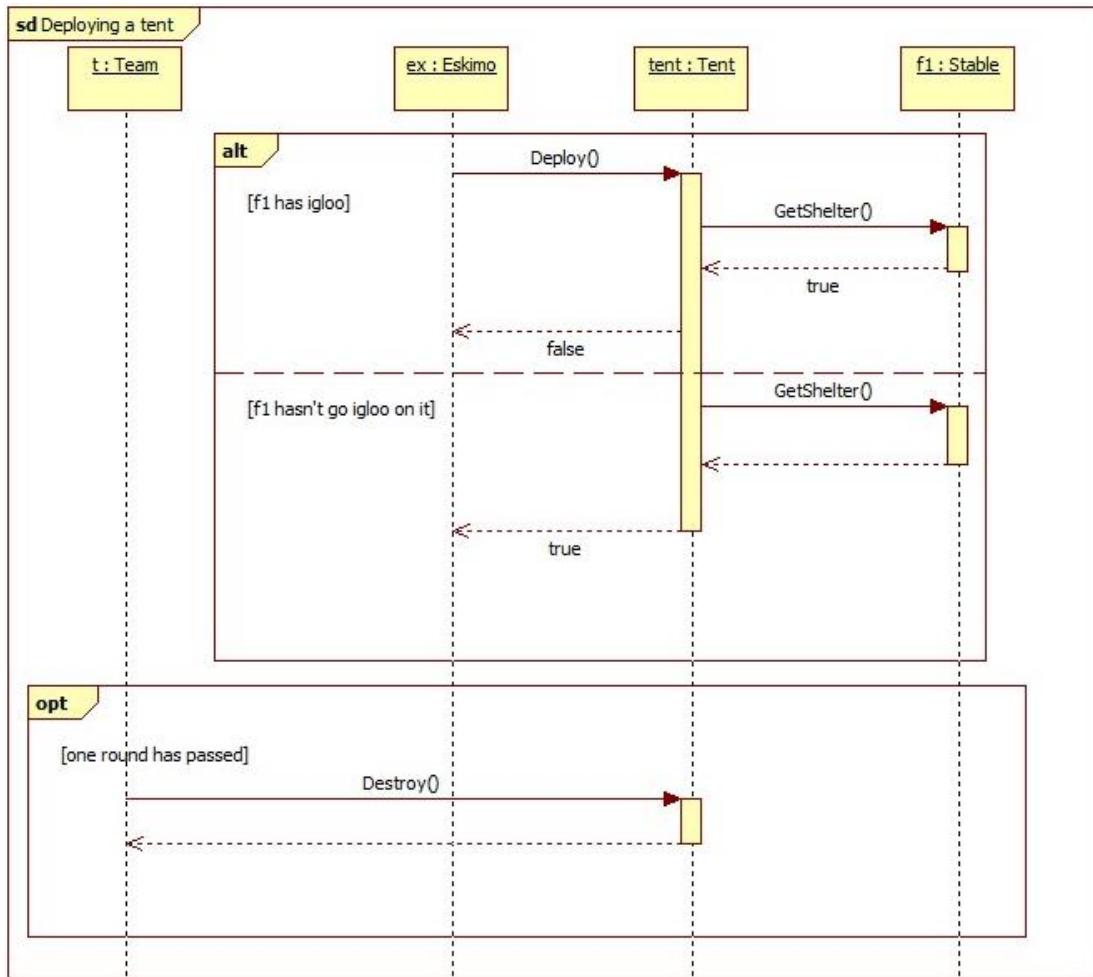
- **Felelősség:** A mozgatható karakterek mozgást kezelő metódusait foglalja össze.
- **Attribútumok:**
  - -
- **Metódusok:**
  - **bool MoveC(Direction d):** A mozgást megvalósító függvény.

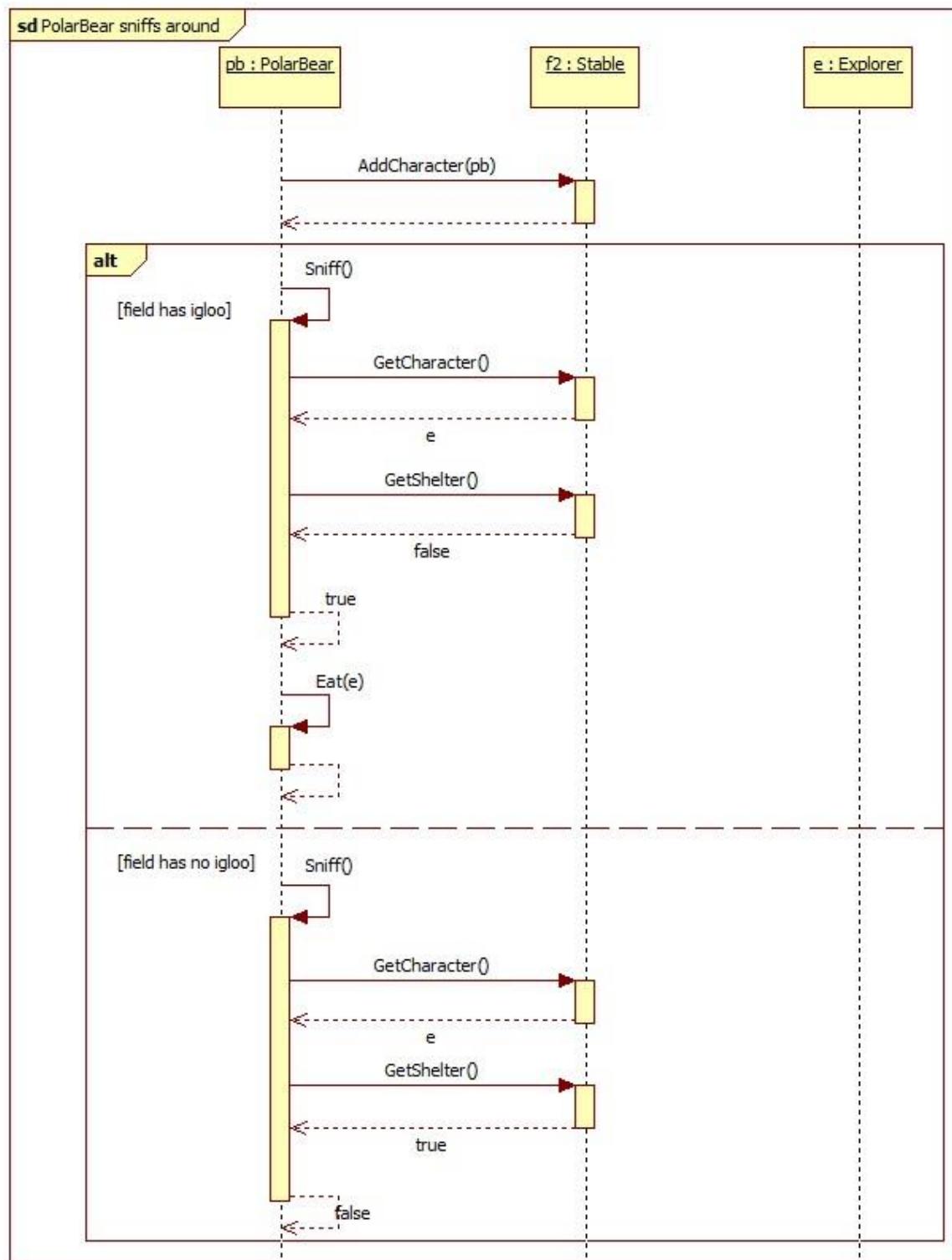
### 7.0.3 Szekvencia-diagramok











## 7.1 Prototípus interface-definíciója

### 7.1.1 Az interfész általános leírása

A prototípus a parancssoros megjelenítéssel, illetve azon parancsok kiadásával irányítható. A jó működés ellenőrizhetőségének érdekében a program előre elkészített teszteseteket batch program segítségével képes lesz futtatni, illetve ellenőrizni és az ellenőrzés sikerességét a felhasználóval tudatni.

### 7.1.2 Bemeneti nyelv

#### *AssambleParts*

**Leírás:** Az éppen aktív karakterre fog hatni ez az utasítás.

**Opciók:** AssembleParts

**pl.:** AssembleParts

Az éppen aktív karakter megpróbálja összerakni az alkatrészeket, amikkel a csapat rendelkezik, ha egy mezőn vannak.

#### *Blizzard*

**Leírás:** A vihart irányító parancs. Ez csak akkor tudjuk használni, ha RandomBlizzard false-ra van állítva, tehát ha a vihar irányításáért mi felelünk. A vihar által érintett mezőket felsorolással adhatjuk meg.

**Opciók:** Blizzard <érintett mező 1 sor szám> <érintett mező 1 oszlop szám> ...  
 <érintett mező n sor szám> <érintett mező n sor szám>

**pl.:** Blizzard 1 1 2 2

A megadott mezőkre meghívódik ezután a BlizzardF() függvény.

#### *Dig*

**Leírás:** Az éppen aktív karakterre fog hatni ez az utasítás.

**Opciók:** Dig

**pl.:** Dig

Az éppen aktív karakter megpróbál ájni.

#### *Eat*

**Leírás:** Az éppen aktív karakterre fog hatni ez az utasítás.

**Opciók:** Eat

**pl.:** Eat

Az éppen aktív karakter megpróbál enni.

#### *EndTurn*

**Leírás:** Az éppen aktív játékosnak jogában áll hamarabb befejezni a körét. Ezt ezzel az utasítással teheti meg. Ha a vihar és a jegesmedve irányítása random, akkor a rendszer ezeket vagy ezután vagy a stamina elfogyása után hajtja végre.

**Opciók:** EndTurn

**pl.:** EndTurn

A rendszer ezután végrehajtja a vihar és a medve mozgatást, ha úgy van beállítva. Ha a medve mozgatása a mi dolgunk, akkor a következő aktív karakternek a medvét állítja be.

### **Eskimo**

**Leírás:** Létrehoz egy eszkimó típusú karakter. Létrehozásánál meg kell adni a karakter pozíóját. Opcionálisan megadhatjuk a karakterünk testhőjét (ha nem adjuk meg akkor alapból a maximális értéket veszi fel), illetve felsorolhatjuk milyen tárgyak vannak nála.

**Opciók:** Eskimo <sorszám> <oszlopszám> <testhő> <tárgy1> ... <tárgyn>

**pl.:** Eskimo 1 1 3 shovel scubasuit

A: Eskimo (1,1) heat: 3 items: Shovel, ScubaSuit

A pálya megjelenítése alatt felsorolás szerűen szerepelnek a karakterek a tulajdonságaikkal.  
Itt fog megjelenni a fentebb mutatott bejegyzés az eszkimóról.

### **Explorer**

**Leírás:** Létrehoz egy kutató típusú karakter. Létrehozásánál meg kell adni a karakter pozíóját. Opcionálisan megadhatjuk a karakterünk testhőjét (ha nem adjuk meg akkor alapból a maximális értéket veszi fel), illetve felsorolhatjuk milyen tárgyak vannak nála.

**Opciók:** Explorer <sorszám> <oszlopszám> <testhő> <tárgy1> ... <tárgyn>

**pl.:** Explorer 1 1 food food

A: Explorer (1,1) heat: 3 items: Food, Food

A pálya megjelenítése alatt felsorolás szerűen szerepelnek a karakterek a tulajdonságaikkal.  
Itt fog megjelenni a fentebb mutatott bejegyzés a kutatóról.

### **GameStart**

**Leírás:** Ez a parancs arra szolgál, hogy a programot ne csak tesztelni tudjuk, de ki is tudjuk próbálni. A parancs paramétereiben megadjuk a pálya méretét, és felsoroljuk a játékosok karaktereinek típusát. A program ezután generálja a pályát, a tárgyakat, a medvét és a karaktereket.

**Opciók:**

GameStart <sor-oszlop szám> <karakter1> .. <karaktern>

**pl.:** GameStart 5 Eszkimo Explorer Eszkimo

Az n karakter számát ellenőrizzük természetesen, hogy megfelelő számú karaktert lehessen csak megadni.

### **GetItem**

**Leírás:** Az éppen aktív karakterre fog hatni ez az utasítás.

**Opciók:** GetItem

**pl.:** GetItem

Az éppen aktív karakter megróbál tárgyat felvenni az alatta lévő mezőről.

### ***Itemcreate***

**Leírás:** Ez a parancs létrehozza a paraméterekkel megadott tárgyat/tárgyakat. Működése a paraméterektől függ. Ha nem adunk meg paramétereket, akkor úgy veszi, hogy a pályát fel kell töltenie véletlenszerűen tárgyakkal. A paraméterek segítségével megadhatjuk, hogy konkrétan mit szeretnénk elhelyezni és hova.

**Opciók:**

Itemcreate <sor szám> <oszlop szám> <típus>

**pl.:** Itemcreate 2 1 shovel

+	====	====	====	+
	3	2	2	
	S	S	H	
+	====	====	====	+
	2	0	0	
	XH	U	S	
		2		
+	====	====	====	+
	0	4	3	
	S	H	S	
+	====	====	====	+

A pályán az itemeknek csak a helyét, jelöljük X karakterrel, ha be van állítva, hogy mutassa, ha egy mezőn van tárgy még akkor is, ha van rajta hó. A tárgy típusát nem mutatjuk.

### ***Loadstate***

**Leírás:** A tesztek futtatásához szükségünk előre megadott játék állapotokra, amiket külső fájlokban tárolunk. Ezek betöltésére szolgál ez a parancs.

**Opciók:** Loadstate <a játékállapot leírást tartalmazó fájl>

**pl.:** Loadstate stablemap.txt

### **Mapcreate**

**Leírás:** Ez a parancs létrehozza a paraméterekkel megadott pályát. Meg kell adni a sorok- oszlop számot a létrehozáshoz. A létrehozott mezőket összeköti egymással. A csak oszlop számot tartalmazó parancs esetén random módon jön létre a pálya, ha konkrétan meg akarjuk adni a mezők típusát azt felsorolással tudjuk megtenni. minden mezőhöz meg kell adnunk a hó mennyiséget, ha nem tesszük meg ezt automatikusan maximálisnak veszi a rendszer.

**Opciók:**

Mapcreate <sor-oszlop szám> <típus> <hómennyiség> <típus> .. <típus> <hómennyiség>

**pl.:** Mapcreate 4 s 2 s 2 h 3 u3 2 u2 1 u3 1 s 4 s 4 s 4 s 4 h 4 u1 3 s 2 h 2 s 4 s 3

+	====	====	====	====	+			
	2		1		3		2	
	S		S		H		U	
							3	
+	====	====	====	====	+			
	1		1		4		4	
	U		U		S		S	
	2		3					
+	====	====	====	====	+			
	4		4		4		3	
	S		S		H		U	
							1	
+	====	====	====	====	+			
	2		2		4		3	
	S		H		S		S	
+	====	====	====	====	+			

A betűk a mező típusát jelzik. s: *Stalbe*, u: *UnStable*, h: *Hole*. Az U megnevezésű mezőkben az U alatt található még a teherbírási is. A példán is látszik, hogy az UnStable mezők típusának megadásába beletartozik a teherbírási is. A

A fájlba írásnál és az olvasásánál természetesen nem fog szerepelni az átláthatóságot könnyítő "pálya" kinézet csak a mezők típusának és hómennyiségének felsorolása.

### **Move**

**Leírás:** Az éppen aktív karakterre fog hatni ez az utasítás. A paraméterben a mozgás irányát adjuk meg.

**Opciók:** Move <irány>

**pl.:** Move NORTH

Az éppen aktív karakter megróbál mozogni a megadott irányba.

### **Polarbear**

**Leírás:** Létrehoz egy jegesmedve típusú karakter. Létrehozás után random mezőre kerül, kivéve, ha paraméterként megadjuk a mező sor és oszlop azonosítóját, ahova szeretnénk tenni.

**Opciók:** Polarbear <sorszám> <oszlopszám>

**pl.:** Polarbear 1 1

P: Polarbear (1,1)

A pálya megjelenítése alatt felsorolás szerűen szerepelnek a karakterek a tulajdonságaikkal. Itt fog megjelenni a fentebb mutatott bejegyzés a jegesmedvéről.

### ***Save***

**Leírás:** Szeretnénk a játék állapotát és a változásokat megfigyelni és szeretnénk elmenteni a dolgokat. Ezzel a parancsal a paraméterezéstől függően el tudjuk elmenteni a játék állapotát és a futás közben készült logokat egy fájlba.

**Opciók:** Save <típus> <a cél fájl neve>

**pl.:** Save log tesztlog.txt

Ha a log helyett state-et írunk a típus paraméterbe, akkor a játék állapotát fogjuk elmenteni.

### ***SpecialAbility***

**Leírás:** Az éppen aktív karakterre fog hatni ez az utasítás. Az utasítás hatása attól függ, hogy milyen típusú az éppen aktív karakter

**Opciók:** SpecialAbility

**pl.:** SpecialAbility

Az éppen aktív karakter meopróbálja használni a speciális képességét.

### ***Setting***

**Leírás:** Ez a parancs megnyitja, vagy módosítja a játék beállításait. A paraméter nélküli verzió csak kiírja az éppen használatban lévő beállításokat.

**Opciók:**

Setting <a változtatni kívánt beállítás> <új érték>

**pl.:** Setting RandomBlizzard false

Az állítható beállítások:

- **RandomBlizzard:** random vihar, a felhasználó dönti el, hogy lesz-e vihar és azt is, hogy hol
- **ControlPolarbear:** random medve mozgás ,ha ez false a felhasználó irányítja a medvét is
- **ItemAlwaysVisible:** a tárgyak minden láthatóak-e
- **CapacityVisible:** látszik-e a mezők teherbírása
- **FieldTypeShow:** mutatja a mező típusát vagy ne

### ***Teszt***

**Leírás:** A tesztek futtatásához szükségünk van betöltő parancsra. Ennek a parancsnak a hatására a rendszer betölti a paraméterként kapott fájlt és végrehajtja az abban megadott utasításokat.

**Opciók:** Teszt <teszt leírást tartalmazó fájl>

**pl.:** Teszt mozgasteszt.txt

### 7.1.3 Kimeneti nyelv

#### A vizuális kimenet

A program futása közben egyszerre szeretnénk egy kipróbálható játékot mutatni és egy olyan modellt adni, amit lehet tesztelni. A képernyőn mátrixba rendezve jelennek meg a mezők. A mezőkön beállításoktól függően a pálya. Alatta egy "status bar" szerűen látható a pályán lévő karakterek típusa, helyzete, testhője (ha van ilyen) és az inventory tartalma. Ez alatt jelenik meg az utolsó utasítás hatására érkező válasz. Itt minden csak a legutolsó utasítás utáni válaszok láthatóak, azonban a ezeket a válaszokat eltároljuk és a Save log utasítás segítségével menthetjük a tartalmát egy fájlba. A log írás kezdete mindenig a legutóbbi betöltött állapot betöltésétől indul. A megjelenítésre adott példa után definiálom a különböző válaszokat.

Ebben a példában, amiben megmutatom a program kinézetét a megjelenítésre vonatkozó beállítások minden be vannak kapcsolva, hogy jobban meg lehessen nézni. A példában egy 5x5-ös pályán egy medve, két eszkimó, egy kutató és néhány tárgy látható.

```
=====+
| 2 | 1 | 2 | 2 | 4 |
|XS | S | H | XU | S |
|   |   |   |   | 2 |
=====+
| 0 | 1 | 3 | 4 | 2 |
| S | H | S | U | S |
|   |   |   |   | 2 |
=====+
| 3 | 1 | 0 | 2 | 4 |
|XU | S | XH | H | S |
| 2 |   |   |   |   |
=====+
| 2 | 0 | 3 | 2 | 0 |
| H | XU | H | S | XS |
| 3 |   |   |   |   |
=====+
| 4 | 1 | 3 | 0 | 4 |
| S | S | H | XU | S |
|   |   |   |   | 2 |
=====+
A: Eskimo (1,2)    heat: 2      items: Rope
B: Eskimo (5,1)    heat: 5      items: Food, Shovel
+C: Explorer (4,5) heat: 3      items: FlareGunPart, ScubaSuit
P: Polarbear (3,5)
A legutolsó esemény: C Dig (4,5)
Kovetkezo lepes: |
```

A példán látható, hogy a status bar-on az aktív karakter egy + jelrel jelöljük, a könnyű kezelés érdekében. A mezőkön a felső szám a mezőn lévő hó mennyiséget mutatja, alatta látható a mező típusa. Ha a mező unstable típusú a teherbírása szerepel alatta. Néhány mezőn szerepel egy X, ami azt jelzi, hogy az adott mezőn tárgy található.

## A parancsokról készül log bejegyzések

### *Blizzard*

- **Blizzard (x,y) no effect:** Az x,y mezőn vihar történt, de a mezőn a hómennyiségek növelésén kívül nem történt semmi, mert nem volt karakter (kivéve jegesmedve) a mezőn.
- **Blizzard(x,y) safe:** Az x,y mezőn vihar történt, de az ott levő karaktereknek nem esett bája, mert sátor/iglu van a mezőn.
- **Blizzard(x,y) A damaged:** Az x,y mezőn vihar történt és az ott lévő A karakter sérülést szerzett. Ez annyiszor jelenik meg, ahány karakter sérült meg a mezőn.

### *Move*

- **A moved from (x,y) to (i,j):** Az A játékos sikeresen átment az i,j mezőre, az x,y mezőről.
- **A moved from(x,y) to (i,j) in scubasuit:** Az A játékos sikeresen át úszott az i,j mezőre az x,y mezőről a búvárruhájában.
- **A moved from (x,y) to (i,j) the field flipped:** Az A játékos átment az i,j mezőre, az x,y mezőről és ezzel az i,j mező túlterhelődött és átfordult és vége a játéknak.
- **A moved from (x,y) to (i,j) and drowned:** Az A játékos átment az i,j lyukra, az x,y mezőről és nem tudták megmenteni, ezért meghalt és vége a játéknak.
- **A moved from (x,y) to (i,j) and was rescued from (l,m):** Az A játékos átment az i,j lyukra, az x,y mezőről, de az l,m mezőn lévő játékos megmentette és magához húzta.
- **Polarbear moved from (x,y) to (i,j):** A jegesmedve átment az i,j mezőre az x,y mezőről.
- **Polarbear moved from (x,y) to (i,j) and attacked:** A jegesmedve átment az i,j mezőre az x,y mezőről és megtámadta a játékosokat, a játéknak vége.

### *Dig*

- **A Dig (x,y):** Az A karakter az x,y mezőn ásott.
- **A Dig (x,y) no snow:** Az A karakter nem ásott az x,y mezőn, mivel azon nincs hó.
- **A Dig (x,y) shovel:** Az A karakter az x,y mezőn ásott az ásóval is. Ez a második üzenet csak, akkor jelenik meg, ha a játékos ásóval is ásott.

### *Eat*

- **A Eat X Y:** Az A karakter evett és a testhője X-ről Y-ra nőtt.
- **A Eat no food:** A játékos nem tudott enni, mert nincs élelmiszer.
- **A Eat max:** A játékos nem evett, mert a testhője maximális már így is.

### *EndTurn*

- **A ended turn early:** A játékos befejezte a körét mielőtt elfogyott volna a staminája. Az, hogy ezután mi történik az a játék beállításától függ.

### *GetItem*

- **A GetItem (x,y):** Az A karakter tárgyat vett fel az x,y mezőn.
- **A GetItem (x,y) snow:** Az A karakter az x,y mezőn, nem tudott tárgyat felvenni, mert hó van a mezőn.

- **A GetItem (x,y) no item:** Az A karakter az x,y mezőn nem tudott tárgyat felvenni, mert nincs tárgy a mezőn.

### *AssambleParts*

- **Victory:** Az A karakter sikeresen összeszerelte a jelző pisztolyt és ezzel a játékosok megyerték a játékot.
- **A AssambleParts missing pieces X:** Az A karakter nem tudta összeszerelni a jelző pisztolyt, mert X darab alkatrész hiányzik.
- **A AssambleParts different fields:** Az A karakter nem tudta összeszerelni a jelző pisztolyt, mert a játékosok nem egy mezőn vannak.

### *SpecialAbility*

- **A Eskimo (x,y) built igloo:** Az A eszkimó típusú karakter az x,y mezőn iglут épített.
- **A Eskimo (x,y) already have igloo:** Az A eszkimó típusú karakter az x,y mezőn nem tud iglут építeni, mert már van a mezőn iglu.
- **A Explorer explored (i,j) capacity X :** Az A kutató típusú karakter felfedezte, hogy az i,j mező kapacitása X.

### *Loadstate*

- **State loaded from X.Y log restarted:** Játék állapotot töltöttünk be az X nevű Y kiterjesztésű fájlból és a log írása újraindult.
- **Can't load state from X.Y:** Nem lehet játékállást betölteni az X nevű és Y kiterjesztésű fájlból.

### *Save*

- **Game state saved to X.Y:** A játék állás mentve lett az X nevű és Y kiterjesztésű fájlba.
- **Game log saved to X.Y:** A játék log mentve lett az X nevű és Y kiterjesztésű fájlba.

## 7.2 Összes részletes use-case

<b>Use-case neve</b>	Hóvihar indít
<b>Rövid leírás</b>	A rendszer hóvihart indít, amely a pálya néhány mezőjét érinti.
<b>Aktorok</b>	System, Player
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. Hóvihar indul, de a mezőre nem tud havat rakni mivel már maximális mennyiségű hó van rajta</li> <li>2. Hóvihar megpróbálja sebezni a mezőn álló karaktereket, de azok igluban vagy sátorban tartózkodnak</li> <li>3. Hóvihar sebzi a mezőn álló karaktereket 1 egység hőenergiával csökkentve</li> </ol>

<b>Use-case neve</b>	Karakter mozgatása
<b>Rövid leírás</b>	A karakterek mozgatása a pályán és a játék megnyeréséhez szükséges jelzőrakéta kilövése.
<b>Aktorok</b>	Player, System
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos sikeresen át mozog egy másik mezőre</li> <li>2. A játékos át mozog egy instabil mezőre és az túlterhelődik így felfordul és az ott lévők meghalnak így vége a játéknak</li> <li>3. A játékos lyukra lép és meghalt mivel nem tudják kimenteni és nincs nála búvárruha</li> <li>4. A játékos lyukra lép és megmentik a szomszédos mezőről egy kötéssel és arra a mezőre fog kerülni.</li> <li>5. A jegesmedve át mozog egy másik mezőre</li> <li>6. A jegesmedve át mozog egy másik mezőre és megtámadja az ott lévő játékosokat megtámadja, de azok igluban vannak</li> <li>7. A jegesmedve át mozog egy másik mezőre és megtámadja az ott lévő játékosokat megtámadja és azok meghalnak és vége a játéknak</li> </ol>

<b>Use-case neve</b>	Ásás
<b>Rövid leírás</b>	A játékosok mezőkről való hó eltakarítását irányítja.
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos ásott a mezőn</li> <li>2. A játékos megpróbál ájni, de nincs hó a mezőn</li> <li>3. A játékos lapáttal ásott</li> <li>4. A játékos törékeny lapáttal ásott</li> </ol>

<b>Use-case neve</b>	Étel fogyasztása
<b>Rövid leírás</b>	A játékosok lehet élelem, amit felhasználhatnak testhőjük növelésére.
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<ol style="list-style-type: none"> <li>1. A játékos evett és 1-gyel nő a testhője</li> <li>2. A játékos megpróbál enni, de nem tud mert nincs nála étel</li> <li>3. A játékos megpróbál enni, de nem tud mert maximális hőenergiája van</li> </ol>

<b>Use-case neve</b>	Kör befejezése
<b>Rövid leírás</b>	A játékosok előbb befejezi a körét.
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<b>1.</b> A játékos előbb befejezi a körét, mint elfogyna a staminája.

<b>Use-case neve</b>	Eszköz felvétel
<b>Rövid leírás</b>	A játékosok vehetnek fel tárgyat ezzel az interakcióval.
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<b>1.</b> A játékos tárgyat vesz fel a mezőről. <b>2.</b> A játékos megpróbál tárgyat felvenni, de nem tud mert hó van a mezőn <b>3.</b> A játékos megpróbál tárgyat felvenni, de nem tud mert nincs tárgy a mezőn

<b>Use-case neve</b>	Jelzőpisztoly összeszerelése
<b>Rövid leírás</b>	A játékosok jelző pisztoly összeszerelése.
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<b>1.</b> A játékos sikeresen összeszereli a jelző pisztolyt és nyernek <b>2.</b> A játékos nem tudja összeszerelni a jelző pisztolyt, mert nincs meg az összes alkatrész. <b>3.</b> A játékos sikeresen összeszereli a jelző pisztolyt, de a játékosok nem egy mezőn vannak.

<b>Use-case neve</b>	Speciális képesség használata
<b>Rövid leírás</b>	A játékosok használhatnak speciális képességeket
<b>Aktorok</b>	Player
<b>Forgatókönyv</b>	<b>1.</b> Az Eszkimó típusú karakter a mezőn iglut épített <b>2.</b> Az eszkimó típusú karakter megpróbál iglut építeni, de nem tud mer már van a mezőn egy iglu <b>3.</b> A Kutató típusú karakter felfedi a mező kapacitását

### 7.3 Tesztelési terv

A tesztesetekhez különböző játékállásokat fogunk betölteni, hogy a tesztekben csak a releváns parancsokat kelljen kiadni, ne kelljen minden a játék legelejétől kezdeni őket. Vannak olyan tesztesetek, amelyeket a részletes kidolgozásban több esetre fogunk bontani, de ezek annyira hasonlóak, hogy a magas szintű forgatókönyveiket nem érdemes szétbontani.

<b>Teszt-eset neve</b>	Karakter stabil mezőre mozog / karakter instabil mezőre mozog, de az nem fordul át
<b>Rövid leírás</b>	Az éppen aktív karakter a Move parancs kiadása után egy stabil mezőre mozog. A logban a ... moved to... bejegyzés jelenik csak meg.
<b>Teszt célja</b>	Character MoveC, Stable AddCharacter függvényeinek bemutatása

<b>Teszt-eset neve</b>	Karakter instabil mezőre mozog, amely utána átfordul
<b>Rövid leírás</b>	Az éppen aktív karakter a Move parancs kiadása után instabil mezőre mozog, amely átfordul. Ez a logban a “... moved to ... and it flipped over” bejegyzés mutatja.
<b>Teszt célja</b>	Character MoveC, UnStable AddCharacter függvényeinek bemutatása

<b>Teszt-eset neve</b>	Karakter lyukba esik és meghal
<b>Rövid leírás</b>	Az éppen aktív karakter a Move parancs kiadása után lyukra mozog, és megfullad. Ez a logban “... moved from ... and drowned” bejegyzésként jelenik meg.
<b>Teszt célja</b>	Character MoveC, Hole AddCharacter függvényeinek bemutatása

<b>Teszt-eset neve</b>	Karakter lyukba esik, de kimentik belőle
<b>Rövid leírás</b>	Az éppen aktív karakter a Move parancs kiadása után lyukra mozog, de kihúzzák belőle. “Ez a logban ...moved from ... and was rescued from...” bejegyzésként jelenik meg.
<b>Teszt célja</b>	Character MoveC, Rescue, Hole AddCharacter függvényeinek bemutatása

<b>Teszt-eset neve</b>	Karakter lyukba esik, de kiúszik belőle
<b>Rövid leírás</b>	Az éppen aktív karakter a Move parancs kiadása után lyukra mozog, de van rajta búvárruha, szóval kiúszik a lyukból.
<b>Teszt célja</b>	Character MoveC, Hole AddCharacter, ScubaSuit Swim függvényeinek bemutatása

<b>Teszt-eset neve</b>	Karakter eszik
<b>Rövid leírás</b>	Az éppen aktív karakter az Eat parancs kiadása után elfogyasztja a tárgyai között lévő ételt, ezzel egygel megnő a testhője. Ilyenkor az étel eltűnik a karakter tárgyai közül.
<b>Teszt célja</b>	Character EatC, Food Eat függvényeinek bemutatása

<b>Teszt-eset neve</b>	Karakter enne, de nincs nála étel
<b>Rövid leírás</b>	Az éppen aktív karakterrel az Eat parancs kiadása után nem történik semmi, mivel nincs étel a tárgyai között.
<b>Teszt célja</b>	Character EatC, tárgyak Eat függvényének bemutatása

<b>Teszt-eset neve</b>	Karakter enne, de már maximális a testhője
<b>Rövid leírás</b>	Az éppen aktív karakterrel az Eat parancs kiadása után nem történik semmi, mivel már maximális a testhője.
<b>Teszt célja</b>	Character EatC, Food Eat függvényeinek bemutatása

<b>Teszt-eset neve</b>	Karakter ásó nélkül ás
<b>Rövid leírás</b>	Az éppen aktív karakter a Dig parancs kiadása után a mezőről, amin áll, eltávolt egy egység hót.
<b>Teszt célja</b>	Character DigC, Field RemoveSnow függvényeinek bemutatása

<b>Teszt-eset neve</b>	Karakter ásóval ás
<b>Rövid leírás</b>	Az éppen aktív karakter a Dig parancs kiadása után a mezőről, amin áll, eltávolt két egység hót.
<b>Teszt célja</b>	Character DigC, Shovel Dig, Field RemoveSnow függvényeinek bemutatása

<b>Teszt-eset neve</b>	Karakter próbál ájni, de nincs hó a mezőn
<b>Rövid leírás</b>	A karakter a Dig parancs kiadására ájni próbál, de nem tudom, mert nincs hó a mezőn. Ilyenkor staminát sem veszít.
<b>Teszt célja</b>	Character DigC, Field RemoveSnow függvényeinek bemutatása

<b>Teszt-eset neve</b>	Karakter törékeny ásóvával ásik, amely eltörök
<b>Rövid leírás</b>	Az éppen aktív karakter a Dig parancs kiadása után a mezőről, amin áll, eltávolt két egység hót. Ez a törékeny ásó 3. használata, így az eltörök.
<b>Teszt célja</b>	Character DigC, Field RemoveSnow, BrittleShovel Dig, Shatter függvényeinek bemutatása

<b>Teszt-eset neve</b>	Jegesmedve mozog
<b>Rövid leírás</b>	A Settings parancs segítségével be lehet állítani, hogy a jegesmedvét a felhasználó irányítsa. Egy játékos köre végén a medve válik aktívvá, és Move parancccsal lehet irányítani. A medve új, üres mezőre lép.
<b>Teszt célja</b>	PolarBear Move függvényének bemutatása

<b>Teszt-eset neve</b>	Jegesmedve megeszi az egyik játékos
<b>Rövid leírás</b>	A Settings parancs segítségével be lehet állítani, hogy a jegesmedvét a felhasználó irányítsa. Egy játékos köre végén a medve válik aktívává, és Move parancccsal lehet irányítani. A medve olyan mezőre lép, ahol tartózkodik már játékos, ilyenkor a játéknak vége.
<b>Teszt célja</b>	PolarBear Move függvényének bemutatása

<b>Teszt-eset neve</b>	Kutató a speciális képességét használja
<b>Rövid leírás</b>	Az éppen aktív kutató a SpecialAbility parancsra felfedi a szomszédos mezők teherbírását, ez az információ innentől elérhetővé válik a játékos számára.
<b>Teszt célja</b>	Explorer SpecialAbility, Field SetRevealed függvényének bemutatása

<b>Teszt-eset neve</b>	Eszkimó a speciális képességét használja
<b>Rövid leírás</b>	Az éppen aktív eszkimó a SpecialAbility parancsra iglut épít a mezőre, ha volt már ott iglu, nem történik semmi, ha nem, az iglu megépül a mezőre.
<b>Teszt célja</b>	Eskimo SpecialAbility függvényémnak bemutatása

<b>Teszt-eset neve</b>	Karakter sárat épít
<b>Rövid leírás</b>	Az éppen aktív karakter leépíti a tárgyai között megtalálható sárat. Hogyha nincs nála sátor, akkor nem történik semmi.
<b>Teszt célja</b>	Tent Deploy függvényének bemutatása

<b>Teszt-eset neve</b>	Játékos tárgyat szed föl
<b>Rövid leírás</b>	A játékos megpróbálja kiásni a tárgyat arról a mezőn, amin áll. Hogyha van hó a mezőn, az akció nem sikeres, hogyha nincs, a karakter megkapja a mezőn tárolt tárgyat (vagy semmit, ha a mezőn nincs tárgy).
<b>Teszt célja</b>	Character GetItemC, Field GiveItem függvényeinek bemutatása

<b>Teszt-eset neve</b>	Játékos befejezi a körét
<b>Rövid leírás</b>	Az EndTurn parancccsal a játékos befejezi a körét, ilyenkor hívódik a Team EndTurn metódusa.
<b>Teszt célja</b>	Team EndTurn, Map BlizzardM, Filed BlizzardF metódusainak bemutatása

<b>Teszt-eset neve</b>	Játékos megpróbálja összeszerelni a jelzőrakétát
<b>Rövid leírás</b>	Az Assemble parancs hatására az éppen aktív játékos megpróbálja összeszerelni a jelzőrakétát. Hogyha az összes játékos egy mezőn áll, és összeszedték a jelzőrakéta-alkatrészeket, akkor a játéknak vége, egyébként a játék folytatódik.
<b>Teszt célja</b>	Team AssembleParts, FlareGunPart Aseemble függvényeinek bemutatása

<b>Teszt-eset neve</b>	Hóvihar olyan mezőt sújt, amelyen nincs játékos
<b>Rövid leírás</b>	Az összes hóvihar-tesztet be lehet állítani manuálisan a Settings Blizzard parancs segítségével. Hogyha a hóvihar olyan mezőt érint, ahol nincs játékos, a mezőn lévő hőmennyiségek egy egységgel nő.
<b>Teszt célja</b>	Map BlizzardM, Field BlizzardF metódusainak bemutatása

<b>Teszt-eset neve</b>	Hóvihar olyan mezőt sújt, amelyen van játékos, és nincs se iglu, se sátor
<b>Rövid leírás</b>	Hasonló az előző teszthez, annyival kiegészítve, hogy a játékosnak egygyel csökken a hőenergiája.
<b>Teszt célja</b>	Map BlizzardM, Field BlizzardF, Character SetHeat metódusainak bemutatása

<b>Teszt-eset neve</b>	Hóvihar olyan mezőt sújt, amelyen van játékos, és van iglu
<b>Rövid leírás</b>	Ugyanaz a kimenetele, mint a "Hóvihar olyan mezőt sújt, amelyen nincs játékos" tesztnek.
<b>Teszt célja</b>	Map BlizzardM, Field BlizzardF metódusainak bemutatása

<b>Teszt-eset neve</b>	Hóvihar olyan mezőt sújt, amelyen van játékos, és van sátor
<b>Rövid leírás</b>	A játékos ilyenkor nem sebződig, viszont a sátor megsemmisül.
<b>Teszt célja</b>	Tent Destroy, Map BlizzardM, Field BlizzardF metódusainak bemutatása

## 7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A prototípus tesztelése egy batch file elindításával végezhető, ami 2 txt fájlt hasonlít össze, az elvárt, illetve a tényleges kimenettel. A tesztelő mappájában találhatóak az elvárt kimenetek txt fájljai (pl. "test1\_expected.txt") illetve a tényleges kimenetek (pl. "test1\_output.txt").

Az "IceField\_Tester.bat" batch programot elindítva kiválasztható a kívánt teszteset. Ha sikeresen lefutott és nincs különbség akkor "no differences encountered" felirat látható, viszont, ha bármi nem megfelelő módon történt, akkor kiírja az elvárt, alatta pedig a tényleges kimenetet összehasonlítva.

## 7.5 Napló

Kezdet	Időtartam	Résznevők	Leírás
2020.04.01 10:00	1 óra	Marton	Tevékenység: Marton megírja az új osztályokat (BrittleShovel, PolarBear, Tent), illetve megtervezzi az ezek miatt bekövetkezett változásokat.
2020.04.01. 17:00	0,5 óra	Háromi	Tevékenység: Háromi frissít az osztálydiagramot a változtatásoknak megfelelően.
2020.04.02 11:00	1 óra	Kedves	Tevékenység: Kedves megírja a bemeneti nyelv pontban szereplő utasításokat
2020.04.02. 13:00	1 óra	Háromi	Tevékenység: Háromi kidolgozza a tesztelési kimenetek összehasonlításának módját, elkészíti a hozzá tartozó leírást, illetve javítja az osztálydiagramot.
2020.04.02 21:00	1,5 óra	Kedves	Tevékenység: A bemeneti nyelv leírás bővítése, A kimeneti nyelv megírása
2020.04.03. 11:00	2 óra	Holczer	Tevékenység: Megírja a use case-et
2020.04.04. 15:00	2 óra	Kurdi	Tevékenység: Tesztesetek elkészítése
2020.04.05. 10:00	1,5 óra	Marton	Tevékenység: Marton elkészíti a rá kiszabott szekvenciadiagrammokat,
2020.04.05 12:00	2,5 óra	Holczer	Tevékenység: Javítja a use case-eket valamint megrajzolja a rákiszánt szekvencia diagramokat
2020.04.05 20:00	1 óra	Kedves	Tevékenység: Bemeneti nyelv javítása és a dokumentum szerkesztése

# 8 Részletes tervezet

## 45 – nullpointer

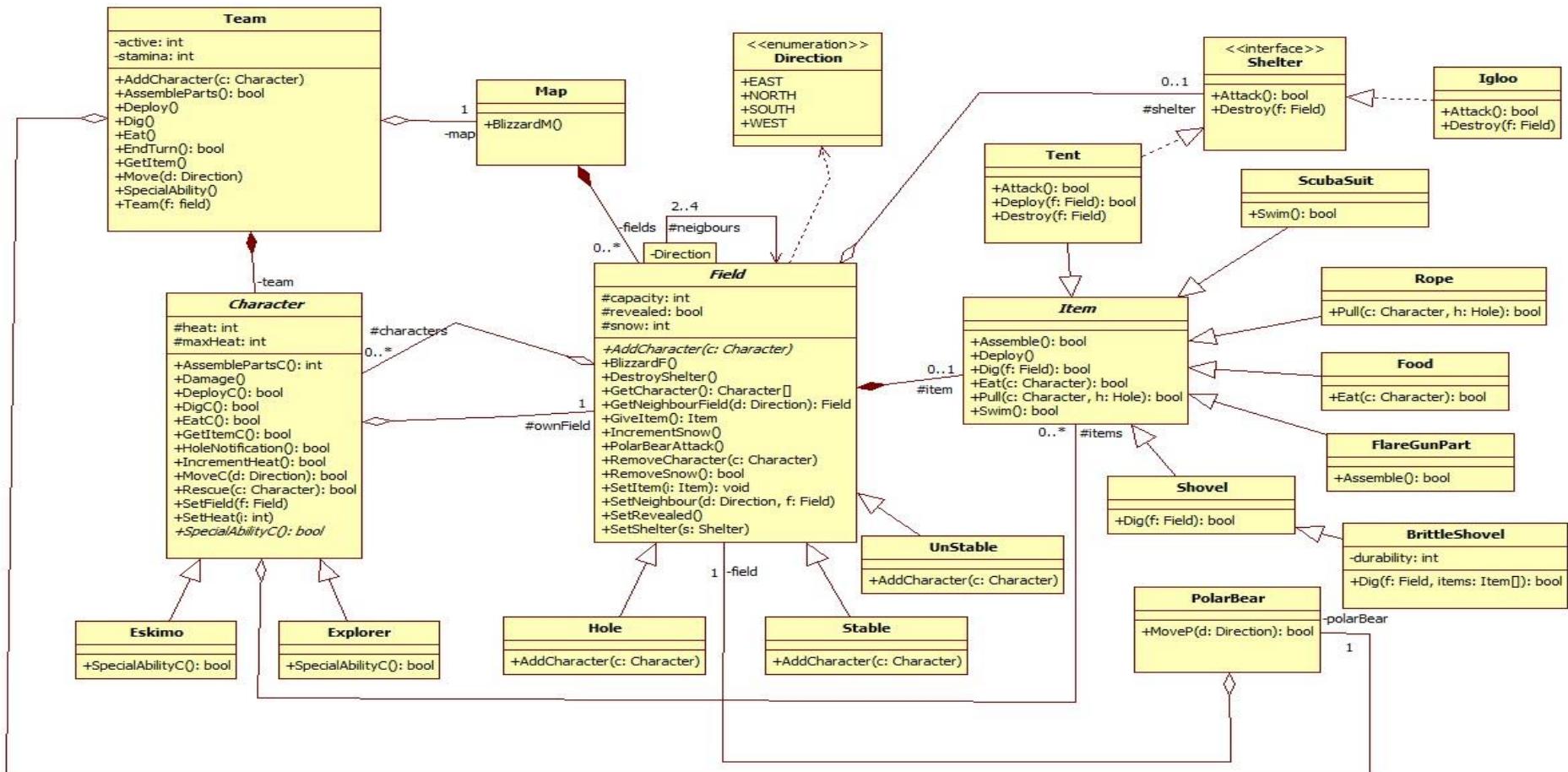
Konzulens:  
Ludmány Balázs

### Csapattagok

Háromi Bence	AK51GI	bence.haromi@gmail.com
Holczer Zsolt	XL9Y0E	zsolti@holczer.net
Kedves Áron Csanád	AV43UW	aron.kedves@gmail.com
Kurdi Barnabás	F0DJNW	barnabas.kurdi@gmail.com
Marton Gergely	FC2WNS	martongergely11@gmail.com

## Változtatások, javítások és pontosítások az előző leadás óta

Módosult osztálydiagram:



**Bemeneti nyelv:**

A módosítások oka, hogy a kezelést megkönnyítő x-y koordináta használata nem volt definiálva a modellben, azért átírtuk a mező megadására szolgáló bemeneti parancsok paramétereit. A leírás és egyéb leíró részek nem változtak a parancsokhoz. Az 1-es számú mezőnek a bal felső sarokban lévő mezőt tekintjük egy pálya esetében, innen jobbra, majd a sor végén a következő sor elejére ugorva folytatjuk a számozást. Erre az azonosítóra a továbbiakban sorszámként fogunk hivatkozni.

***Blizzard***

**Opciók:** Blizzard <sorszám 1> ... <sorszám n>

**pl.:** Blizzard 1 2 (Ekkor az első és második mező lesz érintve a felsorolásban.)

***Deploy***

**Leírás:** Az éppen aktív karakter lerakja a tárgyai között lévő sárat (Tent deploy metódusa) a saját mezőjére.

***Eskimo***

**Opciók:** Eskimo <sorszám> <testhő> <tárgy1> ... <tárgyn>

**pl.:** Eskimo 1 3 shovel scubasuit

A: Eskimo 1 heat: 3 items: Shovel, ScubaSuit

***Explorer***

**Opciók:** Explorer <sorszám> <testhő> <tárgy1> ... <tárgyn>

**pl.:** Explorer 1 3 food food

A: Explorer 1 heat: 3 items: Food, Food

***Itemcreate***

**Opciók:** Itemcreate <sorszám> <típus>

**pl.:** Itemcreate 2 shovel

***Polarbear***

**Opciók:** Polarbear <sorszám>

**pl.:** Polarbear 3

P: Polarbear 3

## Kimeneti nyelv:

A kimeneti nyelvnél a tesztek sikerességének és helyességének vizsgálatára a teszteset lefutása után kimentett log fájl és egy referencia fájl tartalmának összehasonlítását fogjuk használni. Az x,y koordináták használata helyett az egyértelműség kedvéért megváltoztattunk néhány korábbi kimenetet, valamint a tesztek teljessége érdekében új kimeneteket is felvettünk.

## A parancsokról készülő log bejegyzések

### *Blizzard*

- **Blizzard X no effect:** Az X sorszámú mezőn vihar történt, de a mezőn a hómennyiséget növelésén kívül nem történt semmi, mert nem volt karakter (kivéve esetleg jegesmedve) a mezőn.
- **Blizzard X safe:** Az X sorszámú mezőn vihar történt, de az ott levő karaktereknek nem esett baja, mert sátor/iglu van a mezőn.
- **Blizzard X A damaged:** Az X sorszámú mezőn vihar történt és az ott lévő A karakter sérülést szerzett. Ez annyiszor jelenik meg, ahány karakter sérült meg a mezőn.
- **Blizzard X no effect, snow at maximum:** A hóvihar a karakterek támadásán kívül a mezőn lévő hómennyiséget is növeli. Ha maximális a hómennyiség az érintett X azonosítójú mezőn, akkor nem tudja, ekkor jelenik meg ez a kimenet.

### *Deploy*

- **A built tent on X:** Az A játékos sárat helyezett el az X sorszámú mezőn.
- **A has no tent:** Az A játékos nem tudott sárat lehelyezni, mert nincs nála sátor.
- **A didn't build X has shelter:** Az A játékos nem tudott sárat elhelyezni az X sorszámú mezőn, mert a mezőn már van valamilyen menedék.

### *Dig*

- **A Dig X:** Az A karakter az X sorszámú mezőn ásott.
- **A Dig X no snow:** Az A karakter nem ásott az X sorszámú mezőn, mivel azon nincs hó.
- **A Dig X shovel:** Az A karakter az X sorszámú mezőn ásott az ásóval is. Ez a második üzenet csak, akkor jelenik meg, ha a játékos ásóval is ásott.

### *GetItem*

- **A GetItem X:** Az A karakter tárgyat vett fel az X mező sorszámú mezőn.
- **A GetItem X snow:** Az A karakter az X mező sorszámú mezőn, nem tudott tárgyat felvenni, mert hó van a mezőn.
- **A GetItem X no item:** Az A karakter az X mező sorszámú mezőn nem tudott tárgyat felvenni, mert nincs tárgy a mezőn.

### *Move*

- **A moved from X to Y:** Az A játékos sikeresen átment az Y sorszámú mezőre, az X sorszámú mezőről.
- **A moved from X to Y in scubasuit:** Az A játékos sikeresen átúszott az Y sorszámú lyukra az X sorszámú mezőről a búvárruhájában.
- **A moved from X to Y the field flipped:** Az A játékos átment az Y sorszámú mezőre az X sorszámú mezőről, és ezzel az Y mező túlerhelődött és átfordult, vége a játéknak.
- **A moved from X to Y and drowned:** Az A játékos átment az Y sorszámú lyukra, az X sorszámú mezőről és nem tudták megmenteni, ezért meghalt és vége a játéknak.

- **A moved from X to Y and was rescued from Z:** Az A játékos átment az Y sorszámú lyukra, az X mezőről, de a Z sorszámú mezőn lévő játékos megmentette és magához húzta.
- **Polarbear moved from X to Y:** A jegesmedve átment az Y sorszámú mezőre az X sorszámú mezőről.
- **Polarbear moved from X to Y and attacked:** A jegesmedve átment az Y sorszámú mezőre az X sorszámú mezőről és megtámadta a játékosokat, a játéknak vége.

#### *SpecialAbility*

- **A Eskimo X built igloo:** Az A eszkimó típusú karakter az X mező sorszámú mezőn ighut épített.
- **A Eskimo X already have shelter:** Az A eszkimó típusú karakter az X mezősorszámú mezőn nem tud ighut építeni, mert már van a mezőn menedék.
- **A Explorer explored Y capacity X :** Az A kutató típusú karakter felfedezte, hogy az Y mező sorszámú mező kapacitása X.

#### **Automatic events:**

A többi kimenettől eltérően ezek a kimenetek nem akkor hívódnak meg, ha egy konkrét parancsot hajtunk végre, hanem automatikusan meghívódik, ha változás következik be az általa figyelt területen.

- **A I1 item removed I2 ... IN remained:** Az A játékos inventoryjából törlésre került az I1 tárgy, az inventoryban csak az I2 ... IN tárgyak maradtak.
- **Tent on X destroyed:** Az X mező sorszámú mezőn lévő sátor előregedett és elpusztult.
- **GameOver:** Van olyan játékos, aki meghalt és ezzel a játék véget ért.
- **A is active player:** Az A játékos lett az aktív játékos. Ennek a megjelenítése a nyomonkövethetőséget segíti.

## 8. Részletes tervez

### 8.1. Osztályok és metódusok tervez.

#### 8.1.1. BrittleShovel

- **Felelősség:**  
A BrittleShovel osztály a törékeny ásót reprezentálja a programban, amely 3 használat után összetörök. Hogyha ásás közben van egy karakternél BrittleShovel, akkor 1 helyett 2 egység havat képes eltakarítani egy mezőről.
- **Ősosztályok:** Item → BrittleShovel
- **Attribútumok:**
  - **-int durability:** alapértéke 3, azt tartja számon, hányszor lehet még használni az ásót.
- Metódusok:
  - **+bool Dig(Field f, Item[] items):** akkor hívja meg a karakter, mikor ás, ilyenkor a megadott Fieldről a BrittleShovel eltávolít egy egység havat, és true-val tér vissza. Hogyha az ásóval sikeresen ástak, durability attribútuma eggyel csökken, hogyha eléri a 0-t, akkor kitörli magát a karakter tárgyai közül.

#### 8.1.2. Character

- **Felelősség**  
A különböző karakterek absztrakt ősosztálya, ami leírja a közös viselkedéseket és absztrakt függvényként adja meg a karakter típus-specifikus függvényt.
- **Attribútumok**
  - **# int heat:** A karakter testhőjét tárolja egy integer változóban.
  - **# int maxHeat:** A karakter maximális testhőjét tárolja azért, hogy ne lehessen túllépni. A kétféle karakternek más a maximuma.
  - **# ArrayList<Item> items:** A karakternél lévő eszközöket tárolja.
  - **# Field ownField:** A mező, amin a karakter áll.
- **Metódusok:**
  - **+ int AssamblePartsC():** A karakter megpróbálja összeszerelni jelzőrakétát, a nála lévő itemekre meghívja az Assemble függvényt. Megszámolja, hány tárgy tért vissza true-val, majd a true-k számával visszatér.
  - **+ void Damage():** A karakter testhőjét (heat) csökkenti egyel. pl hóvihar esetén
  - **+ bool DeployC():** A játékos sárat próbál állítani a mezőn, amin áll. Végigmegy a tárolt tárgyakon és meghívja rájuk a Deploy függvényt, azzal a mezővel, amin áll. Ha kap true visszatérést, akkor true-val tér vissza, egyébként false-szal tér vissza.
  - **+ bool DigC():** A mezőn hó eltakarítása, ha tudott ásni akkor tér vissza true-val egyébként false-t add vissza. A karakternél tárolt tárgyakon meghívja a Dig függvényt, azzal a field-el, amin áll. Ha van ásó nála, akkor ez is elvégzi az ásást.
  - **+ bool EatC():** A karakter enni próbál egy ételt, ami növeli az életét eggyel, ha tudott enni. Meghívja a nála lévő tárgyak Eat függvényét saját magával. Ha tudott enni akkor tér vissza true-val egyébként false-t add vissza
  - **+ bool GetItemC():** Megpróbál felvenni egy itemet az alatta lévő mező GiveItem függvény meghívásával. Ha a függvény ad vissza valamit, akkor tér vissza true-val egyébként false-t add vissza. A kapott item-et berakja az items tárolóba.
  - **+ bool HoleNotification():** Ez, akkor hívódik meg ha lyukba esett a karakter. Meghívja a nála lévő tárgyakon a Swim függvényt saját magával. Ha bármelyik true-val tér vissza, akkor true-val tér vissza egyébként false.
  - **+ bool IncrementHeat():** Ez a függvény eggyel növeli a játékos életét, ha az nem maximális. Ha tudta növelni, akkor true-val tér vissza egyébként false-szal.

- + **bool MoveC(Direction d):** A karakter lép d irányba úgy, hogy az alatta lévő Field d irányú szomszédjához hozzáadja magát az AddCharacter függvényel az eredeti mezőről pedig törli magát. Ha tudott lépni akkor tér vissza true-val egyébként false-t add vissza. False-t abban az esetben kapunk, ha a karakter ki akar menni a pályáról, ekkor értelemszerűen nem kerül áthelyezésre.
- + **bool Rescue(Character c):** Ez a függvény, akkor hívódik meg, ha egy mellette lévő mezőn lévő társa lyukba esett. A karakter meghívja a tárgyaira a Pull függvényt a bajba esett karakterrel és a társa alatt lévő mezővel. Ha van nála kötél, akkor kimenti a társát és true-val tér vissza.
- + **void SetField(Field f):** Ezzel a függvénytel adhatjuk hozzá a karaktert egy mezőhöz. A játék és a különböző forgatókönyvek létrehozásához szükséges egyszerű setter függvény.
- + **void SetHeat(int i):** A karakter életét lehet beállítani a megadott "i" értékre ezzel a függvénytel, elsősorban a játékos életének 0-ra állítására szolgál.
- + **bool SpecialAbilityC(): abstract** A karakterhez tartozó speciális képességet hajtja végre amennyiben sikerül true-val ha nem akkor false-szal tér vissza.

### 8.1.3. Direction

- **Felelősség**  
Enumeráció osztály az irányok felsorolására. A könnyebb és egyértelműbb kezelés és programozás megkönnyítésére szolgál.
- **Attribútumok**
  - +NORTH
  - +SOUTH
  - +EAST
  - +WEST

### 8.1.4. Eskimo

- **Felelősség**  
Az eszkimó típusú karakterek osztálya. Az eszkimókra jellemző, Character-től eltérő viselkedést és tulajdonságokat tárolja.
- **Ősosztályok:** Character → Eskimo
- **Attribútumok**
  - - int **maxHeat = 5** : A karakter specifikus maximális élet értéke.
- **Metódusok**
  - + **bool SpecialAbilityC():** Az eszkimó speciális képességét hajtja végre, ami egy iglu építése. Az eszkimó létrehoz egy iglut, és meghívja az alatta lévő mező SetShelter függvényét az új iglu-val, mint paraméterrel. Ha sikerült lerakni az iglut true-val tér vissza egyébként false-szal.

### 8.1.5. Explorer

- **Felelősség**
- A kutató típusú karakterek osztálya. Az kutatóra jellemző, Character-től eltérő viselkedést és tulajdonságokat tárolja.
- **Ősosztályok:** Character → Explorer
- **Attribútumok**
  - - int **maxHeat = 4** : A karakter specifikus maximális élet értéke.
- **Metódusok**
  - + **bool SpecialAbilityC():** A kutató speciális képességét hajtja végre. A kutató meghívja az alatta lévő mező összes szomszédjának SetRevealed függvényét, és végül true-val tér vissza.

### 8.1.6. Field

- **Felelősség:**  
Egy absztrakt osztály, mely a pályát alkotó mezőket testesíti meg.
- **Attribútumok:**
  - **# int capacity:** a mezőn lévő karakterek maximális száma
  - **# bool revealed:** a mező capacity értéke látható-e a játékos számára
  - **# int snow:** a mezőn lévő hó mennyisége
  - **# ArrayList<Character> characters:** A mezőn lévő karakterek
  - **# Item item:** a mezőn lévő eszköz
  - **#HashMap<Direction, Field> neighbours:** minden mező ismeri az adott irányban lévő szomszédját.
  - **# Shelter shelter:** ha a mezőn van menedék, akkor azt ebben az attribútumban tároljuk
- **Metódusok:**
  - **+ void AddCharacter(c: Character):** a mezőhöz hozzáad egy karaktert
  - **+ void BlizzardF():** a mező vihar függvénye, amely növelheti a hó mennyiségét, valamint a karakterek testhőjét csökkentheti, ha nincsenek igluban vagy sátorban
  - **+ void DestroyShelter():** A mezőn lévő shelter-en meghívja a Destroy függvényt.
  - **+ ArrayList<Character> GetCharacter():** A mezőn lévő karakter tömböt ad vissza
  - **+ Field GetNeighbourField(Direction d):** Visszaadja a megadott irányba lévő mezőt
  - **+ Item GiveItem():** A mezőn lévő eszközt adja vissza, majd eltávolítja a mezőről
  - **+ void IncrementSnow():** Megnöveli a hó mennyiségét 1-gyel
  - **+ void PolarBearAttack():** meghívja a mezőn lévő shelter Attack függvényét, ha van rajta shelter. Amennyiben nincs rajta shelter vagy a menedék false-szal tér vissza a mezén lévő karakterek SetHeat függvényét 0 paraméterrel. Ellenkező esetben nem csinál semmit.
  - **+ void RemoveCharacter(Character c):** A mezőről eltávolítja a megadott karaktert
  - **+ bool RemoveSnow():** Megpróbál 1 egység havat eltávolítani, ha sikerül true-val, ha nem akkor false-szal tér vissza.
  - **+ void SetItem(Item i):** A mező megkapja az eszközt, amit tárolnia kell
  - **+ void SetNeighbour(Direction d, Field f):** A mező szomszédját állítja be a paraméterként kapott mezőre a paraméterként kapott irányban.
  - **+ void SetRevealed():** Felfedi a mező kapacitását, vagyis a revealed értékét true-ra állítja
  - **+ bool SetShelter(Shelter s):** Beállítja a megadott menedéket a mezőre.

### 8.1.7. FlareGunPart:

- **Felelősség:**  
A jelzőrakéta alkatrészeit reprezentálja. Összesen három szerepel belőlük a játékban. Amennyiben minden harmat összegyűjtötték a játékosok, hogyha egy mezőre gyűlnek, befejezhetik a játékot. Megjegyzés: azért nincs külön osztály a különböző alkatrészeknek, mivel funkcióukban egyáltalán nem különböznek, a grafikus megjelenítéshez később az összes Item rendelkezni fog egy texture attribútummal, amely alapján a különböző alkatrészeket meg lehet majd különböztetni, de ezt nem foglalja magába a modell.
- **Ősosztályok:** Item → FlareGunPart
- **Metódusok:**
  - + bool Assamble(): A játékosnak true értékkal tér vissza, és ezzel jelzi, hogy van FlareGunPartja a játékosnak. Ezeket a true visszatéréseket számolja meg a rendszer.

### 8.1.8. Food

- **Felelősség:**  
Az étel tárgyat reprezentálja. A játékos testhőjének növelésére tudja használni.
- **Ősosztályok:** Item → Food
- **Metódusok:**
  - **bool Eat(Character c):** meghívja a paraméterként kapott Character IncrementHeat() függvényét, ami megnöveli eggyel a testhő attribútumának értékét, true-val tér vissza, ha sikerül testhőjét növelni, ha nem, akkor false-szal.

### 8.1.9. Hole

- **Felelősség:**  
Egy lyuk típusú mezőt reprezentál
- **Ősosztályok:** Field → Hole
- **Attribútumok:**
  - #int capacity = 0
- **Metódusok:**
  - + void AddCharacter(Character c): Hozzáadja a játékosat a lyukhoz, illetve a karaktereken meghívja a HoleNotificationt, valamint, ha kell, a szomszédos mezőkön álló játékosoknak (ha vannak) meghívja a Rescue() függvényét.

### 8.1.10. Igloo

- **Felelősség:**  
Egy építhető osztály, ami megépülhet egy mezőn, és véd a hóviharok és a medve ellen.
- **Interfészek:** Shelter
- **Metódusok:**
  - + bool Attack(): True-val tér vissza, mert meg tudja védeni a mezőn lévő karaktereket a medvétől.
  - + Destroy(Field f): Nem csinál semmit, hiszen az iglu nem tud elpusztulni.

### 8.1.11. Item

- **Felelősség:**  
A tárgyak absztrakt ősosztálya, benne az összes függvény megvan, ami a tárgyakban, azonban mindegyik törzse csak egy false-szal való visszatérést tartalmaz. Csak abban az osztályban van megvalósítva az összes függvény, ahol releváns. A metódusok működését nem részletezzük egyesével, mivel törzsük csak egy hamis visszatérést tartalmaz.
- **Metódusok:**
  - bool Assemble()
  - bool Deploy(Field f)
  - bool Dig(Field: f)
  - bool Eat(Character c)
  - bool Pull(Character c, Hole h)
  - bool Swim()

### 8.1.12. Map

- **Felelősség:**  
A térképet megvalósító singleton osztály, amely tárolja azon mezőket, amelyből a térkép felépül, illetve ez az osztály kezeli a hóvihart a térképen.
- **Attribútumok:**
  - - **ArrayList<Field> fields:** A térképen megjelenő mezők összessége.
- **Metódusok:**
  - + **void BlizzardM():** Ezen metódus minden kör végén meghívott metódus. Meghívás után a beállításoktól függően valamely módon eldöl, hogy lesz-e vihar a kör végén vagy sem. Amennyiben van vihar a megadott vagy véletlenszerűen kiválasztott mezőkön a Map meghívja a BlizzardF függvényt. A vihar létrejöttétől függetlenül visszatérés előtt meghívja a mezők DestroyShelter függvényét, hogy elpusztítsák a sátrákat.

### 8.1.13. PolarBear

- **Felelősség:**  
A jegesmedvét kezelő osztály, amely kezeli annak mozgását és interakcióit a játékosokkal.
- **Attribútumok:**
  - - **Field field:** A mező, amelyen a jegesmedve áll.
- **Metódusok:**
  - + **bool MoveP(Direction d):** A medve lép d irányba. Ha ki próbálna lépni a pályáról, akkor false-szal tér vissza, ellenkező esetben sikeres elmozdulnia, vagyis true-val. Ha sikeresen átlépett a másik mezőre, beállítja azt a saját field-jének és meghívja a mező PolarBearAttack függvényét.

### 8.1.14. Rope

- **Felelősség:**  
A kötél tárgyat reprezentálja. A játékos ennek segítségével ki tud húzni egy mellette lévő játékost egy lyukból.
- **Ősosztályok:** Item -> Rope
- **Metódusok:**
  - **bool Pull(Character c, Hole h):** függvény, amellyel a paraméterként kapott karakter kihúzását valósítja meg a paraméterként kapott lyukból. A logikai visszatérés értéke attól függ, hogy a művelet sikeres volt-e vagy sem.

### 8.1.15. ScubaSuit

- **Felelősség:**  
A búvárruha tárgyat reprezentálja. Ha a karakter viseli, akkor nem fullad meg a vízben.
- **Ősosztályok:** Item → ScubaSuit
- **Metódusok:**
  - **bool Swim():** visszatérít egy true értékkel, ezzel jelzi, hogy a játékosnál van búvárruha.

### 8.1.16. Shelter

- **Felelősség:**  
Az építhető dolgok interfésze.
- **Metódusok:**
  - + **bool Attack():** Ha a medve lép a mezőre meghívódik ez a függvény, ha van shelter a mezőn. A medve ellen biztosít védelmet vagy nem. Alapértelmezett módon false-szal tér vissza, ha ez nincs felülírva.
  - + **void Destroy(Field f):** Törli tudja magát a paraméterül kapott mezőről ezzel a függvénnyel a shelter.

### 8.1.17. Shovel

- **Felelősség:**  
Az ásó tárgyat reprezentálja. Ha birtokosnak van ásója, akkor egygel több egység havat tud eltakarítani ásáskor.
- **Ősosztályok:** Item → Shovel
- **Metódusok:**
  - **bool Dig(Field: f):** az ásó osztály azon függvénye, amely segítségével a paraméterként kapott mezőn áshat 2 egységnyi havat. A visszatérés logikai tartalma attól függ, hogy az adott akció sikeres volt-e vagy sem.

### 8.1.18. Stable

- **Felelősség:**  
Egy stabil típusú mezőt reprezentál, ahova akárhány karakter léphet.
- **Ősosztályok:** Fiel → Hole
- **Metódusok:**
  - + **void AddCharacter(Character c):** A megadott karaktert hozzáadja a mezőhöz.

### 8.1.19. Team

- **Felelősség:**

A team osztály fogja össze a játékosok karaktereit. Ez az osztály tartalmazza a karaktereket és tudja róluk, hogy melyik az éppen soron lévő karakter és hogy annak a karakter hány lépést tehet még a körében. Ez az osztály hajtja végre a játékosokkal a kapott műveleteket.

- **Attribútumok**

- - **int active:** Az éppen aktív karakter indexe.
- - **int stamina:** Az tárolja, hogy a játékos még hány lépést tehet meg a körében. minden kör elején a kezdő értéke 4.
- - **Map map:** A játékteret összefogó Map osztály.
- - **ArrayList<Character> team:** A játékosokat tároló lista.
- - **PolarBear polarbear:** A pályán mászkáló

- **Metódusok**

- + **void AddCharacter(Character c):** Ezzel a függvényel tudunk karaktereket hozzáadni a csoporthoz.
- + **bool AssembleParts():** Meghívja a játékosok AssemblePartsC függvényeit. A visszatérésként kapott számokat összeadja. Ha az összeg három, akkor a játékosok nyertek és true-val ér vissza egyébként false-t ad vissza. Eggyel csökkenti a staminát, ha a stamina elérte a 0-át, akkor meghívja az EndTurn függvényt.
- + **void Deploy():** Meghívja az aktív karakter DeployC függvényét. True visszatérés esetén csökkenti a staminát. Ha a stamina elérte a 0-át, akkor meghívja az EndTurn függvényt.
- + **void Dig():** Meghívja az aktív karakter DigC függvényét. True visszatérés esetén csökkenti a staminát. Ha a stamina elérte a 0-át, akkor meghívja az EndTurn függvényt.
- + **void Eat():** Meghívja az aktív karakter EatC függvényét. True visszatérés esetén csökkenti a staminát. Ha a stamina elérte a 0-át, akkor meghívja az EndTurn függvényt.
- + **bool EndTurn():** Véget ér az aktív játékos köre döntés vagy elfogyott stamina miatt. Az active attribútumot a következő játékos indexére állítja (0, ha végig ért a listán). A staminát négyre állítja. Meghívja a Map BlizzardM függvényét. Ezután megnézi, hogy volt-e játékos, ainek a testhője elérte a 0-át. Ha volt ilyen true-val tér vissza és vége a játéknak egyébként false-szal. Ha a játék megy tovább, és a játék során mi irányítjuk a medvét is, akkor a következő játékos körének megkezdése előtt vár egy irányt, amelybe elmozdítja a medvét, annak MoveP függvényének meghívásával a kapott irány paraméterrel. (Ha a medve false-szal tér vissza újabb irányt vár, hiszen csak sikeres medvemozgatás esetén tud továbblépni a játék)
- + **void GetItem():** Meghívja az aktív karakter GetItemC függvényét. True visszatérés esetén csökkenti a staminát. Ha a stamina elérte a 0-át, akkor meghívja az EndTurn függvényt.
- + **void Move(Direction d):** Meghívja az aktív karakter MoveC függvényét a kapott d paraméterrel. True visszatérés esetén csökkenti a staminát. Ha a stamina elérte a 0-át, akkor meghívja az EndTurn függvényt.
- + **void SpecialAbility():** Meghívja az aktív karakter SpecialAbility függvényét. True visszatérés esetén csökkenti a staminát. Ha a stamina elérte a 0-át, akkor meghívja az EndTurn függvényt.

### 8.1.20. Tent

- **Felelősség:**  
A játékban felállítható sárat reprezentálja, amely egy kör után, de egyébként úgy funkcionál, mint egy iglu, attól eltekintve, hogy nem véd meg a medve támadástól.
- **Interfészek:** Shelter
- **Ősosztályok:** Item → Tent
- **Metódusok:**
  - **+bool Deploy(Field F):** a sátor megpróbálja lerakni magát a paraméterként kapott F mezőre a field SetShelter függvényének magával való meghívásával. A set shelter visszatérését adja tovább.
  - **+ void Destroy(Field f) :** Eltörli magát arról a paraméterül kapott mezőről.

### 8.1.21. Unstable

- **Felelősség:**  
Az instabil mezőt reprezentál, ami azt tudja, hogy ha túl sok karakter van rajta, felborul, és a rajta lévő karakterek meghalnak.
- **Ősosztályok:** Field → Hole
- **Attribútumok:**
  - **- int capacity:** A mező teherbírása
- **Metódusok:**
  - **+ void AddCharacter(Character c):** Hozzáadja a megadott karaktert a mezőhöz, majd ellenőrzi, hogy túlterhelődött-e a mező, és ez alapján meghívja a karaktereken a SetHeat függvényt, amivel 0-ra állítja a mezőn álló karakterek testhőjét.

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

### JÁTÉKÁLLÁSOK:

#### JÁTÉKÁLLÁS 1:

Jeges, három mezőből álló Mapet hoz létre, az egyik mezőn áll egy jegesmedve.

#### Parancsok:

```
MapCreate 1 3 s 2 s 0 s 0  
Polarbear 1 3  
Setting ControlPolarbear true  
Save state jeges.txt
```

#### JÁTÉKÁLLÁS 2:

Egy 6 mezőből álló térképet hoz létre, 2 karakterrel valamelyik mezőn.

#### Parancsok:

```
MapCreate 2 3 s 4 s 2 s 2 s 0 s 1 s 0  
Setting RandomBlizzard false  
Save state stableMap2x3.txt
```

### 8.2.1. Eszkimó a speciális képességét használja

- **Leírás:** Egy eszkimó típusú karakter speciális képességét teszteli le. Az Eskimo parancccsal létrehozhatjuk a karaktert, majd a SpecialAbility parancccsal aktiválhatjuk annak speciális képességét. Ilyenkor a karakter arra a mezőre, ahol áll, egy iglут épít, amennyiben az nem tartalmaz már építményt.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - Az eszkimó az adott mezőre iglut épít.
  - hibahely1: A mezőn már van valamilyen építmény, ezért az utasítás mégis végre hajtódik.
- **Bemenet:**
  - LoadState stableMap2x3.txt
  - Eskimo 2 5
  - SpecialAbility
  - SpecialAbility
  - Save log eskimospecial.txt
- **Elvárt kimenet:**
  - State loaded from stableMap2x3.txt log restarted
  - A is active player
  - A Eskimo 2 built igloo
  - A Eskimo 2 already have shelter
  - Game log saved to eskimospecial.txt

### 8.2.2. Hóvihar olyan mezőt sújt, amelyen nincs játékos

- **Leírás:** A Blizzard parancs segítségével hóvihart indíthatunk, amely a térképen kiválasztott mezőket sújt. A hóvihar növeli az általa sújtott mezőkön a hó mennyiségét, amennyiben nem éri el a maximális értéket (4).
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - A hóvihar mindenhol növeli a hómennyiséget, ahol lesújt
  - hibahely1: A hómennyiség az adott mezőn maximális mégis nő ahómennyiség
- **Bemenet:**
  - LoadState stableMap2x3.txt
  - Eskimo 2 5 food
  - Blizzard 1 3
  - Save log blizzardtest1.txt
- **Elvárt kimenet:**
  - State loaded from stableMap2x3.txt log restarted
  - A is active player
  - Blizzard 1 no effect, snow at maximum
  - Blizzard 3 no effect
  - Game log saved to blizzardtest1.txt

### 8.2.3. Hóvihar olyan mezőt sújt, amelyen van játékos, és nincs se iglu, se sátor

- **Leírás:** A Blizzard parancs segítségével hóvihart indíthatunk, amely a térképen kiválasztott mezőket sújtja. A hóvihar növeli az általa sújtott mezőkön a hó mennyiségét, amennyiben nem éri el a maximális értéket (4), illetve a mezőn tartózkodó védetlen játékosok testhőjét csökkenti.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - A hóvihar olyan mezőt sújt, amelyen van játékos, ezzel a testhőjét csökkenti, amit a játékos túlél, illetve másik esetben nem él túl.
  - hibahely 1: A karakterek nem szenvednek sebzést.
  - hibahely 2: A nulla életű karakter nem hal meg.
- **Bemenet:**
  - LoadState stableMap2x3.txt
  - Explorer 1 5
  - Explorer 1 1
  - Blizzard 1
  - Save log blizzardtest2.txt
- **Elvárt kimenet:**
  - State loaded from stableMap2x3.txt log restarted
  - A is active player
  - Blizzard 1 A damaged
  - Blizzard 1 B damaged
  - Blizzard 1 no effect, snow at maximum
  - GameOver
  - Game log saved to blizzardtest2.txt

#### **8.2.4. Hóvihar olyan mezőt sújt, amelyen van játékos, és van iglu**

- **Leírás:** A Blizzard parancs segítségével hóvihart indíthatunk, amely a térképen kiválasztott mezőket sújtja. A hóvihar növeli az általa sújtott mezőkön a hó mennyiségét, amennyiben nem éri el a maximális értéket, illetve a mezőn tartózkodó védett játékosok testhőjét nem csökkenti.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - A hóvihar olyan mezőt sújt, amelyen van játékos, de nem sebzi őket, mert azok védve vannak.
  - hibahely 1: A játékosok sebződnek.
- **Bemenet:**
  - LoadState stableMap2x3.txt
  - Eskimo 1 5
  - Explorer 1 4
  - SpecialAbility
  - Blizzard 1
  - Save log blizzardtest3.txt
- **Elvárt kimenet:**
  - State loaded from stableMap2x3.txt log restarted
  - A is active player
  - A Eskimo 1 built igloo
  - Blizzard 1 safe
  - Blizzard 1 no effect, snow at maximum
  - Game log saved to blizzardtest3.txt

#### **8.2.5. Hóvihar olyan mezőt sújt, amelyen van játékos, és van sátor**

- **Leírás:** A Blizzard parancs segítségével hóvihart indíthatunk, amely a térképen kiválasztott mezőket sújtja. A hóvihar növeli az általa sújtott mezőkön a hó mennyiségét, illetve a mezőn tartózkodó védett játékosok testhőjét nem csökkenti. A sátor ezt követően megsemmisül.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - A hóvihar olyan mezőt sújt, amelyen van játékos, de nem sebzi őket, mert azok védve vannak. A sátor ezt követően megsemmisül.
  - hibahely 1: a játékosok sebzést szenvednek el
  - hibahely 2: a sátor nem semmisül meg
- **Bemenet:**
  - LoadState stableMap2x3
  - Itemcreate 6 tent
  - Eskimo 6 5
  - GetItem
  - Deploy
  - Blizzard 6
  - Save log blizzardtest4.txt
- **Elvárt kimenet:**
  - State loaded from stableMap2x3 log restarted
  - A is active player
  - A Eskimo GetItem 6
  - A built tent on 6
  - Blizzard 6 safe
  - Tent on 6 destroyed
  - Game log saved to blizzardtest4.txt

### 8.2.6. Játékos befejezi a körét

- **Leírás:** A játékos befejezi a körét, miután az általa kiválasztott tevékenységeket elvégzi. Az Eskimo / Explorer parancssal létrehozhatunk egy játékost, majd pár parancs segítségével (ameddig van staminánk) tevékenységeket végezhetünk. Jogunkban áll a körünket előbb is befejezni, az EndTurn parancs segítségével. Egy játékos köre akkor is véget ér, ha elfogy a staminája.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - A játékos vagy elfogyasztja a stamináját, vagy meghívja az EndTurn parancsot, ezzel befejezve a körét.
  - hibahely 1: Nem jön a következő játékos
- **Bemenet:**
  - LoadState stableMap2x3.txt
  - Explorer 2 3 food
  - Explorer 1 2 tent
  - Eat
  - Dig
  - EndTurn
  - Dig
  - Dig
  - Dig
  - Move EAST
  - Save log endturntest1.txt
- **Elvárt kimenet:**
  - State loaded from stableMap2x3.txt log restarted
  - A is active player
  - A Eat 2 3
  - A Dig 2
  - A ended turn early
  - B is active player
  - B Dig 1
  - B Dig 1
  - B Dig 1
  - Move B from 1 to 2
  - A is active player
  - Game log saved to endturntest1.txt

### 8.2.7. Játékos megpróbálja összeszerelni a jelzőrakétát

- **Leírás:** Az AssembleParts parancs hívásával megkísérelhetjük a jelzőpisztoly összerakását. Ez csak akkor sikeres, ha minden karakter egy mezőn áll.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - A játékosok megkísérelnek összerakni a jelzőpisztolyt.
  - hibahely1: nincs meg minden alkatrész
  - hibahely2: az összes játékos nem ugyanazon mezőn áll
- **Bemenet:**
  - LoadState stableMap2x3.txt
  - Eskimo 6 3 gunpart1 gunpart2 gunpart3
  - Explorer 6 4 food
  - AssembleParts
  - Save log assembleTest.txt
- **Elvárt kimenet:**
  - State loaded from stableMap2x3.txt log restarted
  - A is active player
  - Victory
  - Game log saved to assembleTest.txt

### 8.2.8. Játékos tárgyat szed fel

- **Leírás:** A játékos a GetItem parancccsal felveheti a tárgyat arról a mezőről, ahol van tárgy, illetve nincsen már hó.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - A játékos egy tárgyat szed fel.
  - hibahely1: az adott tárgyat hó fedi, ezért nem lehet felszedni.
  - hibahely2: nincs tárgy az adott mezőn.
- **Bemenet:**
  - LoadState stableMap2x3.txt
  - Itemcreate 6 food
  - Itemcreate 1 shovel
  - Eskimo 6 5
  - Eskimo 1 4
  - Explorer 4 3
  - GetItem
  - EndTurn
  - GetItem
  - EndTurn
  - GetItem
  - Save log getitemtest.txt
- **Elvárt kimenet:**
  - State loaded from stableMap2x3 log restarted
  - A is active
  - A GetItem 6
  - A ended turn early
  - B is active
  - B GetItem 6 snow
  - B ended turn early
  - C is active player
  - C GetItem 6 no item
  - Game log saved to getitemtest.txt

### 8.2.9. Jegesmedve megeszi az egyik játékost

- **Leírás:** A Settings parancs segítségével be lehet állítani, hogy a jegesmedvét a felhasználó irányítsa. Egy játékos köre végén a medve válik aktívvá, és Move parancccsal lehet irányítani. A medve olyan mezőre lép, ahol tartózkodik már játékos, ilyenkor a játéknak vége.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - PolarBear Move metódusa.
  - Hibahely 1: a jegesmedve olyan mezőre mozog, amelyen nem áll játékos, vagy van játékos, de iglu is.
- **Bemenet:**
  - Load jeges.txt
  - Eskimo 1 1
  - EndTurn
  - Move WEST
  - Save log bearattack.txt
- **Elvárt kimenet:**
  - State loaded from jeges.xml log restarted
  - A is active player
  - A ended turn early
  - Polarbear moved from 2 to 1 and attacked
  - GameOver
  - Game log saved to bearattack.txt

### 8.2.10. Jegesmedve mozog

- **Leírás:** A Settings parancs segítségével be lehet állítani, hogy a jegesmedvét a felhasználó irányítsa. Egy játékos köre végén a medve válik aktívvá, és Move parancccsal lehet irányítani. A medve új, üres mezőre lép.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - PolarBear Move metódusa.
- **Bemenet:**
  - Load jeges.txt
  - Eskimo 1 1
  - EndTurn
  - Move EAST
  - Save log bearmove.txt
- **Elvárt kimenet:**
  - State loaded from jeges.xml log restarted
  - A is active player
  - A ended turn early
  - Polarbear moved from 2 to 3
  - Game log saved to bearmove.txt

### 8.2.11. Karakter ásó nélkül ás

- **Leírás:** Az éppen aktív karakter a Dig parancs kiadása után a mezőről, amin áll, eltávolít egy egység hót.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - Chatacter Dig metódusa.
  - hibahely 1: a játékos nem tud ájni
- **Bemenet:**
  - Load jeges.txt
  - Eskimo 1 1
  - Dig
  - Save log dignoshovel.txt
- **Elvárt kimenet:**
  - State loaded from jeges.xml log restarted
  - A is active player
  - A Dig 1
  - Game log saved to dignoshovel.txt

### 8.2.12. Karakter ásóval ás

- **Leírás:** Az éppen aktív karakter a Dig parancs kiadása után a mezőről, amin áll, eltávolít két egység hót.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - Character Dig metódusa.
  - hibahely 1: A játékos ásója nem ás.
- **Bemenet:**
  - Load jeges.txt
  - Eskimo 1 4 shovel
  - Dig
  - Save log digshovel.txt
- **Elvárt kimenet:**
  - State loaded from jeges.xml log restarted
  - Eskimo Dig 1
  - Eskimo Dig 1 shovel
  - Game log saved to digshovel.txt

### 8.2.13. Karakter enne, de már maximális a testhője

- **Leírás:** Ezzel a teszttel azt a forgatókönyvet teszteljük, amikor karakterünk egy mezőn áll, van étel és enni próbál, de a karakter típusára jellemző maximális testhője van. A vizsgálatban megnézzük, hogy a karakter testhője a maximális felé megy-e, a stamina illetve elfogyasztja-e az ételt. Mindkét karakterre viszgálni fogjuk ezt a funkciót az eltérő maximális hő miatt. A tárgy használat tesztelés érdekében több tárgy lesz a karakternél.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - A játékos evés képessége
  - hibahely 1 : A testhő maximális fölé nő.
  - hibahely 2 : A játékosnak elpusztul az élelme.
- **Bemenet:**
  - Mapcreate 1 1 s 0
  - Eskimo 1 5 food shovel rope
  - Explorer 1 4 scubasuit brittleshovel food
  - Eat
  - EndTurn
  - Eat
  - Save log eatteszt1.txt
- **Elvárt kimenet:**
  - A is active player
  - A Eat max
  - A ended turn early
  - B is active player
  - B Eat max
  - Game log saved to eatteszt1.txt

### 8.2.14. Karakter enne, de nincs nála étel

- Leírás: Ezzel a teszttel azt a forgatókönyvet teszteljük, amikor karakterünk egy mezőn áll és enni próbál, de a karakternél nincs étel. A vizsgálatban megnézzük, hogy a karakter testhője a nő-e. Mindkét karakterre viszgálni fogjuk ezt a funkciót, annak ellenére, hogy a viselkedés azonos a teszt teljessége érdekében. A tárgy használat tesztelés érdekében több tárgy lesz a karakternél.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - A játékos evés képessége
  - hibahely 1 :A testhő növekszik
- **Bemenet:**
  - Mapcreate 1 1 s 0
  - Eskimo 1 5 shovel rope
  - Explorer 1 4 scubasuit brittleshovel tent
  - Eat
  - EndTurn
  - Eat
  - Save log eatteszt2.txt
- **Elvárt kimenet:**
  - A is active player
  - A Eat no food
  - A ended turn early
  - B is active player
  - B Eat no food
  - Game log saved to eatteszt2.txt

### 8.2.15. Karakter eszik

- **Leírás:** Ezzel a teszttel azt a forgatókönyvet teszteljük, amikor karakterünk egy mezőn áll és enni próbál, van nála étel és nem maximális a testhője. A vizsgálatban megnézzük, hogy a karakter testhője nő-e. Mindkét karakterre viszgálni fogjuk ezt a funkciót, az eltérő maximális testhő miatt. A tárgy használat tesztelés érdekében több tárgy lesz a karakternél.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - A karakter evés képessége
  - hibahely 1 : A karakterek nem nő a testhője.
  - hibahely 2 : A karakterek nem tünik el az étel az inventoryjából.
- **Bemenet:**
  - Mapcreate 1 1 s 0
  - Eskimo 1 4 shovel rope food
  - Explorer 1 3 scubasuit food brittleshovel tent
  - Eat
  - EndTurn
  - Eat
  - Save log eatteszt3.txt
- **Elvárt kimenet:**
  - A is active player
  - A Eat 4 5
  - A food removed shovel rope remained
  - A ended turn early
  - B is active player
  - B Eat 3 4
  - B food removed scubasuit brittleshovel tent remained
  - Game log saved to eatteszt3.txt

### 8.2.16. Karakter instabil mezőre mozog, amely utána átfordul

- **Leírás:** Ezzel a teszt esettel azt a forgatókönyvet teszteli amikor a karakter instabil mezőre lép ami ezzel túlterhelődik és meghalnak a mezőn álló karakterek. Először is egy karakter lép egyet a szomszédos intabli mezőre, ami így túlterhelődik és felborul így az ott levő karakterek élete 0-ra állítódik és így vége a játéknál.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - karakter mozgása és instabil mezőre és felborulását teszteli
  - hibahely: A karakterek nem lesz 0-ra a testhőhely
  - hibahely: Nem minden karakter hal meg
- **Bemenet:**
  - Mapcreate 1 2 s 1 u1 2
  - Eskimo 1 3 food
  - Explorer 2 2 shovel
  - Move EAST
  - Save log flip.txt
- **Elvárt kimenet:**
  - Eskimo moved from 1 to 2 and the field flipped
  - GameOver
  - Game log saved to flip.txt

### 8.2.17. Karakter lyukba esik és meghal

- **Leírás:** Ezzel a teszt esettel azt a forgatókönyvet teszteljük amikor a karakter lyuk mezőre lép és mivel nincs nála búvárruha és a szomszédos mezőkön nincs olyan karakter amelyiknél lenne kötél ezáltal ki tudná menteni így meghal és vége a játéknak
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - Karakter lyukba lépése és halálának tesztelése
  - hibahely: nem lett 0 a hőenergiája
- **Bemenet:**
  - Mapcreate 1 2 s 1 h 2
  - Eskimo 1 2 food
  - Move EAST
  - Save log holedie.txt
- **Elvárt kimenet:**
  - A is active player
  - Eskimo moved from 1 to 2 and drowned
  - GameOver
  - Game log saved to holedie.txt

### 8.2.18. Karakter lyukba esik, de kimentik belőle

- **Leírás:** Ezzel a tesztesettel azt a forgatókönyvet teszteljük amikor a karakter lyuk mezőre lép és nincs nála búvárruha, de az egyik szomszédos mezőn áll egy játékos amelyiknél van kötél így ki tudják húzni a lyukból
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - Karakter lyukba lépésének és kimentésének tesztelése
- **Bemenet:**
  - Mapcreate 1 2 s 1 h 2
  - Eskimo 1 3 food
  - Explorer 1 4 tent rope
  - Move EAST
  - Save log holeAndSave.txt
- **Elvárt kimenet:**
  - A is active player
  - Eskimo moved from 1 to 2 and was rescued from 1
  - Game log saved to holeAndSave.txt

### 8.2.19. Karakter lyukba esik, de kiúszik belőle

- **Leírás:** Ezzel a teszt esettel azt a forgatókönyvet teszteljük amikor a karakter lyuk mezőre lép, de van nála búvárruha így kiúszik a lyukból
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - Scubasuit Swim függvénye
  - hibahely : A karakter sebződik a búvárruha ellenére
- **Bemenet:**
  - Mapcreate 1 2 s 1 h 2
  - Eskimo 1 3 food scubasuit
  - Move EAST
  - Save log holeAndScubaSuit.txt
- **Elvárt kimenet:**
  - A is active player
  - Eskimo moved from 1 to 2 in scubasuit
  - Game log saved to holeAndScubaSuit.txt

### 8.2.20. Karakter próbál ájni, de nincs hó a mezőn

- **Leírás:** A karakter a Dig parancs kiadására ásni próbál, de nem tud, mert nincs hó a mezőn. Ilyenkor staminát sem veszít.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - A teszeset ellenőrzi, hogy a karakternek kiadott ásás parancs nem megy végbe, ha nincs hó a táblán
  - hibahely 1: Az ásás végbe megy
- **Bemenet:**
  - Mapcreate 1 1 s 0
  - Eskimo 1 3 rope
  - Dig
  - Save log nosnow.txt
- **Elvárt kimenet:**
  - A is active player
  - A Dig 1 no snow
  - Game log saved nosnow.txt

### 8.2.21. Karakter sárat épít, de nincs nála sátor

- **Leírás:** Az éppen aktív karakter leépíténe a tárgyai között megtalálható sárat. Nincs nála sátor, nem történik semmi.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - Ellenőrzi, hogy a karakter csak akkor tud sárat építeni, ha van nála sátor, így ha nincsen, A has no tent kimenetet várunk
- **Bemenet:**
  - Mapcreate 1 1 s 0
  - Eskimo 1 3
  - Deploy
  - Save log deploynotent.txt
- **Elvárt kimenet:**
  - A is active player
  - A has no tent
  - Game log saved deploynotent.txt

### 8.2.22. Karakter sárat épít, úgy, hogy van nála sátor

- **Leírás:** Az éppen aktív karakter leépíti a tárgyai között megtalálható sárat.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - Ellenőrzi, hogy a karakter tényleg leépíti-e a sárat, ha van nála sátor és kiadjuk a Deploy parancsot
- **Bemenet:**
  - Mapcreate 1 1 s 1
  - Eskimo 1 3 tent shovel
  - Deploy
  - Save log buildtent.txt
- **Elvárt kimenet:**
  - A is active player
  - A built tent on 1
  - A tent removed shovel remained
  - Game log saved to buildtent.txt

### **8.2.23. Karakter stabil mezőre mozog / karakter instabil mezőre mozog, de az nem fordul át**

- **Leírás:** Az éppen aktív karakter a Move parancs kiadása után egy instabil és egy stabil mezőre mozog. A logban a ... moved from .. to... bejegyzés jelenik csak meg.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - Amikor a karakter az instabil mezőre mozog, annak nem szabad felfordulnia, mert a teherbírása 3-ra van állítva és az egyetlen karakter az eszkimó lesz rajta, ezért csak a a moved kimenet kell, hogy megjelenjen.
- **Bemenet:**
  - Mapcreate 1 3 s 1 u3 1
  - Eskimo 1 3
  - Move EAST
  - Move EAST
  - Save log move.txt
- **Elvárt kimenet:**
  - A is active player
  - A moved from 1 to 2
  - A moved from 2 to 3
  - Game log saved to move.txt

### **8.2.24. Karakter törékeny ásóval ás, amely eltörök**

- **Leírás:** Az éppen aktív karakter a Dig parancs kiadása után a mezőről, amin áll, eltávolt két egység hót. Ez a törékeny ásó 3. használata, így az eltörök.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - 3 ásás után az ásónak el kell törnie, ezt ellenőrzi a teszteset, hogy valóban megtörténik-e
- **Bemenet:**
  - Mapcreate 1 2 s 4 s 4
  - Eskimo 1 tent brittleshovel
  - Dig
  - Dig
  - Move EAST
  - Dig
  - Save log shovelbreaks.txt
- **Elvárt kimenet:**
  - A is active player
  - A Dig 1
  - A Dig 1 Shovel
  - A Dig 1
  - A Dig 1 Shovel
  - A moved from 1 to 2
  - A Dig 2
  - A Dig 2 Shovel
  - A shovel item removed remained tent
  - Game log saved to shovelbreaks.txt

### 8.2.25. Kutató a speciális képességét használja

- **Leírás:** Az éppen aktív kutató a SpecialAbility parancsra felfedi a szomszédos mezők teherbírását, ez az információ innentől elérhetővé válik a játékos számára.
- **Ellenőrzött funkcionalitás, várható hibahelyek:**
  - A kutató felfedi a körülötte lévő táblák teherbírását
- **Bemenet:**
  - Mapcreate 3 3 u1 1 u4 1 u5 1 u8 1 u2 1 u5 1 u3 1 u4 1 u7 1
  - Explorer 5 3
  - SpecialAbility
  - Save log explorerspec.txt
- **Elvárt kimenet:**
  - A is active player
  - A Explorer explored 2 capacity 4
  - A Explorer explored 6 capacity 5
  - A Explorer explored 8 capacity 4
  - A Explorer explored 4 capacity 8
  - Game log saved to explorerspec.txt

## 8.3. A tesztelést támogató programok tervezése

A prototípus tesztelése egy tesztelő batch program (“IceField\_Tester.bat”) elindításával végezhető. Ha elindítottuk a programot választhatunk a tesztesetek közül (az előtte lévő számot kell beírni, majd enter nyomni).

A tesztelőnek szüksége van 2 fájlra, egy tényleges, illetve egy elvárt kimenetet tartalmazó txt fájlra (pl. “test1\_output.txt” és “test1\_expected.txt”). A program a tényleges kimenet fájl tartalmát (ami egy save paranccsal kimentett log file) összehasonlítja az elvárt kimeneti fájllal (ami a kimeneti nyelv azon elemeit tartalmazó log fájl, amelyeket akkor ír ki a program, ha sikeresen lefut). Ha sikeresen lefutott és nincs különbség akkor “no differences encountered” felirat látható, viszont, ha bármilyen megfelelő módon történt, akkor kiírja az elvárt, alatta pedig a tényleges kimenetet összehasonlítva.

## 8.4. Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.04.06. 14:00	1 óra	Háromi Holczer Kedves Kurdi Marton	Értekezlet. Döntés: osztályleírások elosztása, elkészítése 2020.04.07. 18:00 határidőre.
2020.04.07. 10:30	1 óra	Kedves	Tevékenység: A character, eskimo, explorer, direction,team,shelter, igloo osztályok leírásának elkezdése
2020.04.07. 12:00	1,5 óra	Háromi	Tevékenység: PolarBear osztályok leírása, osztálydiagram frissítése.
2020.04.07. 18:00	1,5 óra	Háromi Holczer Kedves Kurdi Marton	Értekezlet. Döntés: A tesztesetek részletes leírásának elkészítése 2020.04.09. 18:00 időpontra.
2020.04.08 11:00	2 óra	Kedves	Tevékenység: A character, eskimo, explorer, direction,team,shelter, igloo osztályok leírása. A karakter evéssel kapcsolatos tesztek leírása.
2020.04.08 10:00	2 óra	Marton	Tevékenység: Marton leírja a Map osztály jellemzőit, illetve elkészíti az első 8 tesztesetet.
2020.04.08. 14:00	2 óra	Holczer	Tevékenység: Holczer leírja a Field, Unstable, Stable, Hole osztályok leírását
2020.04.08 18:00	1 óra	Holczer Kedves Kurdi Marton	Értekezlet. Döntés: Tesztek pontos kidolgozása, korábbi ki- és bemeneti nyelv frissítése 2020.04.10. 18:00-ig
2020.04.08. 22:00	1 óra	Háromi	Tevékenység: Háromi elkészíti a saját teszteseteinek a leírását.
2020.04.09 11:00	0,5 óra	Kedves	Tevékenység: Kedves javítja és frissíti a rászabott osztályokat és teszteseteket.
2020.04.09.12:00	1 óra	Háromi	Tevékenység: Háromi kidolgozza a tesztesetei ki- és bemeneteit.
2020.04.09. 13:00	3 óra	Kurdi	Tevékenység: Jegesmedve, ásás tesztek, tárgy osztályok elkészítése
2020.04.09 14:00	2 óra	Holczer	Tevékenység:

			Javítja az osztály leírásokat és megírja a teszteseteket(8.2.16, 8.2.17, 8.2.18, 8.2.19)
2020.04.10. 11:00	1 óra	Kedves	Tevékenység: A bemeneti és kimeneti nyelv javítások
2020.04.10. 11:00	1 óra	Marton	Tevékenység: Teszteset javítások
2020.04.10. 14:00	2 óra	Háromi	Tevékenység: Háromi a tesztelést támogató segédprogramot átdolgozza, a leírását frissíti. Elkezdi a teszteseteinek a be-, illetve kimeneteinek leírását.
2020.04.10. 18:00	0,5 óra	Háromi Holczer Kedves Kurdi Marton	Megbeszélés: Minden résztvevő elvégzi a javításokat a saját tesztesetein 2020.04.12. 18:00-ig.
2020.04.11. 11:00	1 óra	Háromi	Tevékenység: Háromi befejezi a teszteseteinek leírását, javítja a PolarBear osztály leírását és a segédprogram leírását a megbeszéltek szerint.
2020.04.11. 11:30	0,5 óra	Kedves	Tevékenység: Kedves átnézi és javítja a rá kiszabott teszteket és osztályokat
2020.04.11. 11:00	0,5 óra	Marton	Tevékenység: Marton javításokat végez a tesztesetein.
2020.04.11. 16:00	1 óra	Háromi Holczer Kurdi Kedves	Megbeszélés: A csapat átnézi az elkészült tervezeteket, javításokat végez Döntés: Moveable interfész eltávolítása.
2020.04.11. 17:00	1,5 óra	Kedves	Tevékenység: Kedves javítja a rá kiszabott osztály leírásokat és teszteseteket
2020.04.11. 17:00	1,5 óra	Kurdi	Tevékenység: Kurdi javítja a rá kiszabott osztályleírásokat és teszteseteket
2020.04.11. 17:00	1,5 óra	Háromi	Tevékenység: Háromi javítja a rá kiszabott osztály leírásokat és teszteseteket.
2020.04.11. 17:00	1,5 óra	Holczer	Tevékenység: Kedves javítja a rá kiszabott osztály leírásokat és teszteseteket
2020.04.11. 18:00	1,5 óra	Marton	Tevékenység: Marton utólagos ellenőrzést végezt
2020.04.11. 20:00	1 óra	Kedves	Tevékenység:

			Kedves szerkeszti és letisztázza a dokumentációt.
--	--	--	---------------------------------------------------

# 10. Prototípus beadása

45 – nullpointer

Konzulens:  
Ludmány Balázs

## Csapattagok

Háromi Bence	AK51GI	bence.haromi@gmail.com
Holczer Zsolt	XL9Y0E	zsolti@holczer.net
Kedves Áron Csanád	AV43UW	aron.kedves@gmail.com
Kurdi Barnabás	F0DJNW	barnabas.kurdi@gmail.com
Marton Gergely	FC2WNS	martongergely11@gmail.com

## 10.1 Fordítási és futtatási útmutató

### 10.1.1 Fájllista

Fájl neve	Méret bytes	Keletkezés ideje	Tartalom
BlizzardParam.java	154	2020 Apr 27 00:57	enum a Blizzard kiíráshoz
BrittleShovel.java	1145	2020 Apr 27 00:57	ásó, amely eltörök
Character.java	12337	2020 Apr 27 00:57	a játékosokat reprezentálja
Controller.java	25161	2020 Apr 27 00:57	a konzolos kiírást végzi
Direction.java	121	2020 Apr 27 00:57	enum az irányokra
EatParam.java	133	2020 Apr 27 00:57	enum az evés kiírásához
Eskimo.java	2074	2020 Apr 27 00:57	eszkimó karakter
Explorer.java	1724	2020 Apr 27 00:57	felfedező karakter
Field.java	7453	2020 Apr 27 00:57	a játéktér egy mezője, abszt.
FlareGunPart.java	1029	2020 Apr 27 00:57	a jelzőpisztoly
Food.java	754	2020 Apr 27 00:57	étel, amely növeli a testhőt
Hole.java	1647	2020 Apr 27 00:57	lyuk a pályán
Igloo.java	606	2020 Apr 27 00:57	iglu, amely véd
Item.java	1028	2020 Apr 27 00:57	tárgyak abszt. ősosztálya
ItemParam.java	157	2020 Apr 27 00:57	enum tárgyhaszn. kiírásához
Main.java	23884	2020 Apr 27 00:57	a Main osztály
Map.java	6649	2020 Apr 27 00:57	a pálya mezőit tárolja
MovingParam.java	207	2020 Apr 27 00:57	enum a mozgás kiírásához
PolarBear.java	1526	2020 Apr 27 00:57	jegesmedve, amely vadászik
Rope.java	1094	2020 Apr 27 00:57	kötél osztály
ScubaSuit.java	659	2020 Apr 27 00:57	búvárruha osztály
Shelter.java	430	2020 Apr 27 00:57	interfész iglunak és sátornak
Shovel.java	692	2020 Apr 27 00:57	ásó osztály
ShovelParam.java	141	2020 Apr 27 00:57	enum ásás kiírásához
SpecParam.java	147	2020 Apr 27 00:57	enum speciális képességek kií.
Stable.java	979	2020 Apr 27 00:57	stabil mező osztálya
Team.java	8374	2020 Apr 27 00:57	a játékosokat tárolja
Tent.java	1294	2020 Apr 27 00:57	sátor osztály
TentParam.java	150	2020 Apr 27 00:57	enum sátor kiírásához
Unstable.java	1362	2020 Apr 27 00:57	instabil mező osztálya

### 10.1.2 Fordítás

Navigáljunk el a mappához cd parancs ismételt használata segítségével és lépjünk be a szkkelon mappába

*cd proto*

Ezzel a parancssal tudjuk lefordítani a porgramot:

```
javac src/character/*.java src/enums/*.java src/field/*.java src/item/*.java src/main/*.java
src/map/*.java src/team/*.java -d bin
```

Létrejött egy bin mappa, ebbe kerültek a lefordított fájlok.

### 10.1.3 Futtatás

A proto mappán belül ennek a parancsnak a kiadásával:

*java -cp bin main.Main*

Ezzel a bin mappát állítjuk be classpath-nek.

A teszteléshez egy sokkal kényelmesebben használható batch prgramot alkalmazunk, ehhez az Icefield\_Tester.bat fájlon duplán kattintva elénk kerül a program:

```
IceField Tester

Enter the number of the test case which would you like to start:
1) assembletest
2) bearattack
3) bearmove
4) blizzardtest1
5) blizzardtest2
6) blizzardtest3
7) blizzardtest4
8) buildtent
9) deploynotent
10) dignoshovel
11) digshovel
12) eatteszt1
13) eatteszt2
14) eatteszt3
15) endturntest1
16) eskimospecial
17) explorerspec
18) flip
19) getitemtest
20) holeAndSave
21) holeAndScubaSuit
22) holedie
23) move
24) nosnow
25) shovelbreaks
26) Exit

Enter a number:
```

Itt kiválaszthatjuk a tesztelni kívánt tesztesetet. A batch program a fentebb bemutatott java parancsokat használja a fordításhoz, illetve a futtatáshoz. A proto/test/input mappában találhatóak a tesztesetek bemeneti fájljai, a proto/test/output mappában pedig a prototípus adott bemenettel lefutatott log fájljai kerülnek mentésre, az elvárt kimeneti logfájlok, amik helyes működés esetén keletkeznek a proto/test/expected mappában találhatóak. A batch fájl

lefordítja a java prototípust majd egy számot vár, itt választhatunk egy tesztesetet, a látható névhez tartozó `nev_input.txt`-ben található parancsokat megadja a programnak, ami pedig egy `nev.txt` fajlt generál az `output` mappába. Ezután összehasonlítja az elvárt kimenettel (`nev_expected.txt`) a tényleges kimenetet (`nev.txt`) és ha nem volt különbség akkor a *FC: no differences encountered* felirat tárul elénk:

```
*Result of assembletest*

Comparing files TEST\EXPECTED\assembleTest_expected.txt and TEST\OUTPUT\ASSEMBLETEST.TXT
FC: no differences encountered
```

Ebben az esetben a tesztünk sikeresnek tekinthető, az elvárt módon futott le. Ellenkező esetben ha valami nem a megfelelően működött és nem a megfelelő kimenet keletkezett az elvárt fájlokat összehasonlítja a tényleges kimenettel:

```
*Result of assembletest*

Comparing files EXPECTED\assembleTest_expected.txt and OUTPUT\ASSEMBLETEST.TXT
***** EXPECTED\assembleTest_expected.txt
A is active player
Victory
Game log saved to assembleTest.txt
***** OUTPUT\ASSEMBLETEST.TXT
A is active player
Game log saved to assembleTest.txt
*****
```

Ebben az esetben a tesztünk sikertelen.

## 10.2 Tesztek jegyzőkönyvei

Bizonyos tesztbe- és kimeneteken szisztematikusan változtattunk, ezek a következők: Az elvárt kimenetekben szereplő “moved”-ot bizonyos esetekben “move”-ra cseréltük, a játékosok és tárgyak elnevezési konvencióját használjuk az elvárt kimenetekben (A, B, C... és Shovel1, Scubasuit2, stb.). A save és load state parancsokat végül nem valósítottuk meg, így az olyan teszteket, amelyekhez volt szükség játékállás betöltésére, kiegészítettük a megfelelő parancsokkal, az elvárt kimenetet is ehhez mérten módosítottuk.

### 10.2.1 Játékos megpróbálja összeszerelni a jelzőrakétát (assembletest.txt)

<b>Tesztelő neve</b>	Háromi Bence
<b>Teszt időpontja</b>	2020.04.25. 23:00

<b>Tesztelő neve</b>	Háromi Bence
<b>Teszt időpontja</b>	2020.04.25. 21:11
<b>Teszt eredménye</b>	Hiba
<b>Lehetséges hibaok</b>	Victory után aktív karakter váltás, nem fejeződött be ezért ténylegesen a játék victory után sem ért véget. Az teszeset leírásának inputjában el volt írva a tárgy neve.
<b>Változtatások</b>	Victory után aktív karakter váltás megszüntetése. Inputnál tárgyak neve le lett cserélve flaregunpart-ra.

### 10.2.2 Jegesmedve megeszi az egyik játékost (bearattack.txt)

<b>Tesztelő neve</b>	Háromi Bence
<b>Teszt időpontja</b>	2020.04.25. 21:20

<b>Tesztelő neve</b>	Háromi Bence
<b>Teszt időpontja</b>	2020.04.25. 21:15
<b>Teszt eredménye</b>	Hiba
<b>Lehetséges hibaok</b>	A tesztleírásnál rossz helyre volt inicializálva a medve.
<b>Változtatások</b>	Medve áthelyezése alapból a 2-es mezőre.

### 10.2.3 Jegesmedve mozog (bearmove.txt)

<b>Tesztelő neve</b>	Háromi Bence
<b>Teszt időpontja</b>	2020.04.25. 23:35

<b>Tesztelő neve</b>	Háromi Bence
<b>Teszt időpontja</b>	2020.04.25. 21:25
<b>Teszt eredménye</b>	Hiba
<b>Lehetséges hibaok</b>	A tesztleírásnál rossz helyre volt inicializálva a medve. Az elvárt kimenet hibás volt, nem számított az “A is active player”-re, ez a tényleges kimeneten megjelent, helyesen. Az irány megadás parancs bemenete megváltozott.

<b>Változtatások</b>	Medve áthelyezése alapból a 2-es mezőre. Az elvárt kimenet javítva. Az irány megadó parancs át lett írva Move EAST-ről EAST-re.
----------------------	---------------------------------------------------------------------------------------------------------------------------------

#### 10.2.4 Hóvihar olyan mezőt sújt, amelyen nincs játékos (blizzardtest1.txt)

<b>Tesztelő neve</b>	Háromi Bence
<b>Teszt időpontja</b>	2020.04.25. 21:10

<b>Tesztelő neve</b>	Háromi Bence
<b>Teszt időpontja</b>	2020.04.25. 21:00
<b>Teszt eredménye</b>	Hiba
<b>Lehetséges hibaok</b>	Indexelés az elvárt kimenetek esetén 1-től indulnak, a programban 0-tól, így elcsúsztak a sorszámozások.
<b>Változtatások</b>	Indexelés átalakítása 1-től indulóra.

#### 10.2.5 Hóvihar olyan mezőt sújt, amelyen van játékos, és nincs se iglu, se sátor (blizzardtest2.txt)

<b>Tesztelő neve</b>	Háromi Bence
<b>Teszt időpontja</b>	2020.04.25. 22:55

<b>Tesztelő neve</b>	Háromi Bence
<b>Teszt időpontja</b>	2020.04.22. 22:45
<b>Teszt eredménye</b>	Hiba
<b>Lehetséges hibaok</b>	Logolás kötben fordítva írta ki, a karakter sebzést és a maximum értékű hómennyiséget elérését blizzard után.
<b>Változtatások</b>	A kiíratás sorrendjének cseréje.

#### 10.2.6 Hóvihar olyan mezőt sújt, amelyen van játékos, és van iglu (blizzardtest3.txt)

<b>Tesztelő neve</b>	Háromi Bence
<b>Teszt időpontja</b>	2020.04.25. 23:00

#### 10.2.7 Hóvihar olyan mezőt sújt, amelyen van játékos, és van sátor(blizzardtest4.txt)

<b>Tesztelő neve</b>	Holczer Zsolt
<b>Teszt időpontja</b>	2020.04.25 23:10

<b>Tesztelő neve</b>	Holczer Zsolt
<b>Teszt időpontja</b>	2020.04.25 23:00
<b>Teszt eredménye</b>	Hiba

<b>Lehetséges hibaok</b>	Inputban rosszul indexelt fieldek, a sátor nem pusztult el a kör végén
<b>Változtatások</b>	Inputban indexek javítása valamint a kódban sátor elpusztítása javítás

**10.2.8 Karakter sátrat épít, úgy, hogy van nála sátor(buildtent.txt)**

<b>Tesztelő neve</b>	Holczer Zsolt
<b>Teszt időpontja</b>	2020.04.25 23:25

<b>Tesztelő neve</b>	Holczer Zsolt
<b>Teszt időpontja</b>	2020.04.25 23:15
<b>Teszt eredménye</b>	Hiba
<b>Lehetséges hibaok</b>	Nem szedte ki az inventory-ból a sárat
<b>Változtatások</b>	Kódban javítás

**10.2.9 Karakter sátrat épít, de nincs nála sátor(deploynotent.txt)**

<b>Tesztelő neve</b>	Holczer Zsolt
<b>Teszt időpontja</b>	2020.04.25 23:30

**10.2.10 Karakter ásó nélkül ás(dignoshovel.txt)**

<b>Tesztelő neve</b>	Holczer Zsolt
<b>Teszt időpontja</b>	2020.04.25 23:45

**10.2.11 Karakter asóval ás(digshovel.txt)**

<b>Tesztelő neve</b>	Holczer Zsolt
<b>Teszt időpontja</b>	2020.04.26 00:10

<b>Tesztelő neve</b>	Holczer Zsolt
<b>Teszt időpontja</b>	2020.04.25 23:50
<b>Teszt eredménye</b>	out lényegi része: A Dig shovel expected lényegi része: A Dig A Dig shovel
<b>Lehetséges hibaok</b>	csak akkor írta ki a "A dig"-et ha szimplán ásott ha ásóval akkor csak "A Dig shovel" szöveg jelent meg
<b>Változtatások</b>	Kódban javítás, hogy "A Dig" a karakter 1-el majd "A Dig shovel" pedig még 1-el való hó csökkentést jelentse

**10.2.12 Karakter enne, de már maximális a testhője**

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26. 9:00-10:30

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26. 9:00-10:30
<b>Teszt eredménye</b>	Sikeres teszt, a kimenet és az elvárt kimenet megegyezik
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

#### 10.2.13 Karakter enne, de nincs nála étel

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26. 9:00-10:30

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26 9:00-10:30
<b>Teszt eredménye</b>	Hiba, eltérő elvárt és való kimenet, max heat jelenik a no food helyett
<b>Lehetséges hibaok</b>	Kimenet rossz helyen való meghívása
<b>Változtatások</b>	Kimenet meghívás változtatása

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26 9:00-10:30
<b>Teszt eredménye</b>	Sikeres teszt, a kimenet és az elvárt kimenet megegyezik
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

#### 10.2.14 Karakter eszik

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26. 9:00-10:30

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26 9:00-10:30
<b>Teszt eredménye</b>	Hibás eredmény
<b>Lehetséges hibaok</b>	Elvárt kimenetben nem a futásnak megfelelő sorrendben vannak felírva a kimenetek
<b>Változtatások</b>	Elvárt kimenet javítása, hogy a kimenet sorrendje megegyezzen a futás során lévő sorrenddel.

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26 9:00-10:30
<b>Teszt eredménye</b>	Sikeres teszt, az elvárt és a valós kimenet egyezik
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

### 10.2.15 Játékos befejezi a körét

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26. 9:00-10:30

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26 9:00-10:30
<b>Teszt eredménye</b>	Hiba, eltérő elvárt és való kimenet
<b>Lehetséges hibaok</b>	Hibás elvárt kimenet. Nem szerepel rajta a tárgy törléséről készült log bejegyzés.
<b>Változtatások</b>	Elvárt kimenet javítása. Hiányzó kimenet bevétele.

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26 9:00-10:30
<b>Teszt eredménye</b>	Sikeres teszt, az elvárt és a valós kimenet egyezik
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

### 10.2.16 Eszkimó a speciális képességét használja

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26. 9:00-10:30

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26 9:00-10:30
<b>Teszt eredménye</b>	Eltérő elvárt és valós kimenet
<b>Lehetséges hibaok</b>	A valós kimenetben "already" szerepel az "already" helyett
<b>Változtatások</b>	A kimenet megfelelő részének javítása.

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26 9:00-10:30
<b>Teszt eredménye</b>	Sikeres teszt, az elvárt és a valós kimenet egyezik
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

### 10.2.17 Kutató a speciális képességét használja

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26. 9:00-10:30

<b>Tesztelő neve</b>	Kedves Áron Csanád
<b>Teszt időpontja</b>	2020.04.26 9:00-10:30
<b>Teszt eredménye</b>	Sikeres teszt, az elvárt és a valós kimenet egyezik
<b>Lehetséges hibaok</b>	-

<b>Változtatások</b>	-
----------------------	---

### 10.2.18 Unstable túlterhelődik és felborul

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:00

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:00
<b>Teszt eredménye</b>	Sikertelen, az elvárt kimenet elején nem jelenik meg az A is active player üzenet
<b>Lehetséges hibaok</b>	Rossz elvárt kimenet
<b>Változtatások</b>	Elvárt kimenet megváltoztatása

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:00
<b>Teszt eredménye</b>	Sikeress teszt
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

### 10.2.19 GetItem parancs tesztelése

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:05

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:05
<b>Teszt eredménye</b>	Sikertelen, az elvárt kimenetben rossz a mezők számozása
<b>Lehetséges hibaok</b>	Rossz elvárt kimenet
<b>Változtatások</b>	Elvárt kimenet megváltoztatása

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:05
<b>Teszt eredménye</b>	Sikeress
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

### 10.2.20 Rescue tesztelése

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:10

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:10
<b>Teszt eredménye</b>	Sikertelen, az elvárt kimenetben nem jelenik meg a megmentés után, hogy a játékos átmozgott arra a mezőre, ahonnan kimentették
<b>Lehetséges hibaok</b>	Rossz elvárt kimenet
<b>Változtatások</b>	Úgy döntöttünk, hogy tartozhat két mozgás kimenet a kimentéshez, mivel ténylegesen két mezőnyi mozgás történik.

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:10
<b>Teszt eredménye</b>	Sikeres
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

**10.2.21 Úszás scubasuittal**

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:15

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:15
<b>Teszt eredménye</b>	Sikeres
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

**10.2.22 Karakter lyukba esik és meghal**

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:20

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:20
<b>Teszt eredménye</b>	Sikertelen, az elvárt kimenetben máshogyan van megfogalmazva a parancs
<b>Lehetséges hibaok</b>	Rossz elvárt kimenet
<b>Változtatások</b>	Elvárt kimenet igazítása

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:20
<b>Teszt eredménye</b>	Sikeres
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

**10.2.23 Sima mozgás**

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:25

<b>Tesztelő neve</b>	Kurdi Barnabás
<b>Teszt időpontja</b>	2020. 04. 26. 12:25
<b>Teszt eredménye</b>	Sikeres
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

**10.2.24 Karakter olyan helyen ás ahol nincs hó**

<b>Tesztelő neve</b>	Marton Gergely
<b>Teszt időpontja</b>	2020.04.26 13:00

<b>Tesztelő neve</b>	Marton Gergely
<b>Teszt időpontja</b>	2020.04.26 13:00
<b>Teszt eredménye</b>	Sikertelen.
<b>Lehetséges hibaok</b>	Az elvárt kimenetben már nem használt parancsok vannak használva
<b>Változtatások</b>	Ezen parancsok eltávolítása.

<b>Tesztelő neve</b>	Marton Gergely
<b>Teszt időpontja</b>	2020.04.26. 13:10
<b>Teszt eredménye</b>	Sikeress
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

### 10.2.25 Karakter törékeny ásóval ásik, ami eltörik

<b>Tesztelő neve</b>	Marton Gergely
<b>Teszt időpontja</b>	2020.04.26 13:10

<b>Tesztelő neve</b>	Marton Gergely
<b>Teszt időpontja</b>	2020.04.26 13:10
<b>Teszt eredménye</b>	Sikeress.
<b>Lehetséges hibaok</b>	-
<b>Változtatások</b>	-

## 10.3 Értékelés

Tag neve	Tag neptun	Munka százalékban
Háromi Bence	AK51GI	21,5%
Holczer Zsolt	XL9Y0E	19,5%
Kedves Áron Csanád	AV43UW	21%
Kurdi Barnabás	F0DJNW	20,5%
Marton Gergely	FC2WNS	17,5%

## 10.4 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.04.14. 18:00	1 óra	Háromi Holczer Kedves Kurdi Marton	Értekezlet. Döntés:
2020.04.15. 10:00	1 óra	Háromi	Tevékenység: Háromi elkészíti a Tent osztály-tés a többi osztály függvényeinek fejléceit.
2020.04.15 11:00	1,5 óra	Kedves	Tevékenység: Kedves elkészíti a Character, Eskimo és Explorer osztályokat
2020.04.15 16:00	1 óra	Holczer	Tevékenység: Megírja a rá kiszabott osztályokat(Item, BrittleShovel, Shelter, Igloo)
2020.04.18 13:30	1,5 óra	Kedves	Tevékenység: Kedves javítja a Character, Eskimo és Explorer osztályokat és elkezdi elkészteni a Main osztályt
2020.04.19. 12:00	3.5 óra	Kurdi	Tevékenység: Kurdi megírja a Field osztályt és a leszármazottjait
2020.04.20. 15:00	2 óra	Háromi	Tevékenység: Háromi implementálja a Tent, PolarBear, Map, Team osztályokat
2020.04.21. 16:00	2 óra	Kedves	Tevékenység: Kedves javítja a Character, Eskimo, Explorer osztályokat és kibővíti a Main osztályt és megkezdi az osztályainak kommentelését
2020.04.21 10:00	1 óra	Marton	Tevékenység: Marton javít a Rope, ScubaSuit, Shovel, Food osztályokon.
2020.04.22 11:00	1,5 óra	Kedves	Tevékenység: Kedves elvégzi a Main osztály befejezését, valamint kommenteli a Character, Eskimo, Explorer és Main osztályokat
2020.04.22. 14:00	2 óra	Háromi	Tevékenység: Háromi javítja a Map, Team konstruktorkat, javítja a jegesmedve mozgásának a logikáját, illetve elkészíti az elvárt kimenetek fájljait.
2020.04.22. 18:00	2.5 óra	Marton	Tevékenység: Marton elkészíti a Controller osztály rá kiszabott részét, elkészít új enum

			osztályokat (BlizzardParam, ItemParam, ShovelParam, TentParam).
2020.04.22. 22:00	2.5 óra	Holczer	Tevékenység: elkészítí a Controller osztály rá kiszabott részét, új enumok készítése
2020.04.23. 11:00	2 óra	Háromi	Háromi javítja a Map osztályt, a tesztelő batch fájlt és az osztálydiagramot, befejezi a kimeneti fájlokat.
2020.04.23 20:00	2 óra	Marton	Tevékenység: Marton javít az általa elkészített munkán, extra függvényeket definiál
2020.04.23. 20:00	3 óra	Kurdi	Tevékenység: map kirajzolásának elkészítése, javítások eszközölése
2020.04.24 10:00	1 óra	Marton	Tevékenység: Marton teszteléseket végez az elkészített munkán.
2020.04.24. 10:00	3.5 óra	Kurdi	Tevékenység: Controller osztály függvényeinek meghívása, javítások
2020.04.24. 10:00	1,5 óra	Kedves	Tevékenység: Kedves javítja a Main osztályt, megírja a serializalásra és teszt futtatásra szolgáló parancsot. Felveszi az oztályaiba a szükséges kimeneti függvényeket
2020.04.24. 12:00	2 óra	Háromi	Tevékenység: Controller függvényeinek meghívása, javítások.
2020.04.24 14:00	2 óra	Holczer	Tevékenység: Controller javítása
2020.04.24 16:00	4 óra	Háromi Holczer Kedves Kurdi Marton	Döntés: A csapat eldönti, hogy miken kell javítani
2020.04.24 23:00	1 óra	Holczer	Tevékenység: evés és az eszköz felvétel függvények javítása
2020.04.25 13:30	5 óra	Háromi Holczer Kedves Kurdi Márton	Döntés: A csapat eldönti, hogy miken kell javítani
2020.04.25 23:00	2 óra	Holczer	Tevékenység: Ellenőrzi és javítja a rá kiszabott teszteket(10.2.7-10.2.11) és dokumentáció írás
2020.04.25. 23:00	2 óra	Háromi	Tevékenység:

			Kimenetek javítása.
2020.04.25. 23:00	5 óra	Kurdi	Tevékenység: Polarbear fiexelése, Team EndAction, tesztelés
2020.04.26 9:00	1,5 óra	Kedves	Tevékenység: Kedves ellenőrzi és javítja a rá kiszabott teszteket(10.2.13-10.2.18) és dokumentáció írás
2020.04.26. 13:00	2 óra	Háromi	Tevékenység: Tesztesetek naplózása, javítások.

# 11 Grafikus felület specifikációja

45 – nullpointer

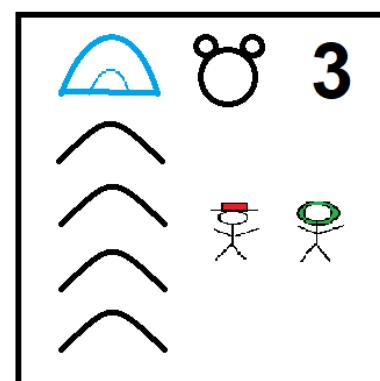
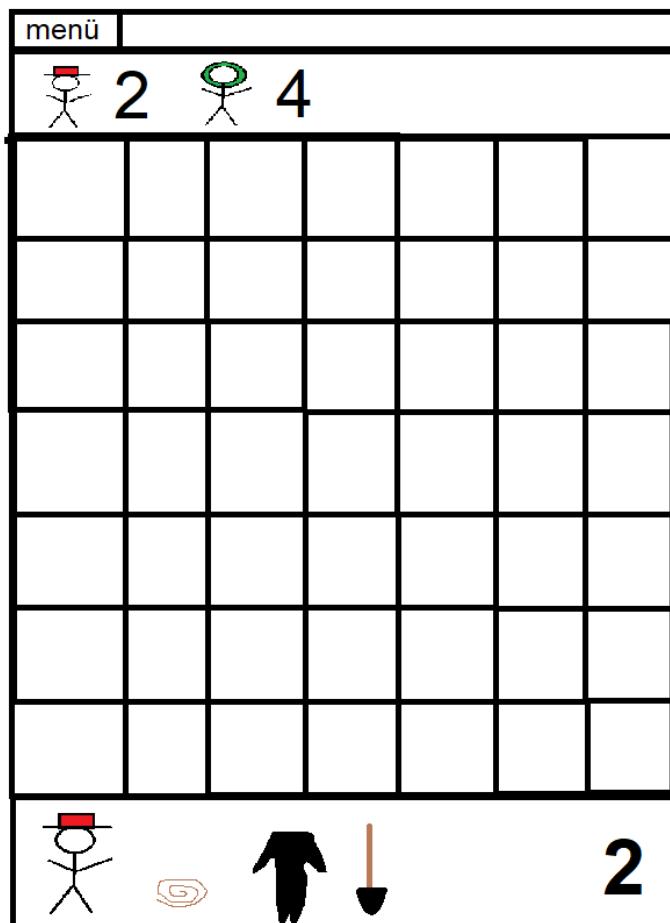
Konzulens:  
**Ludmány Balázs**

## Csapattagok

Háromi Bence	AK51GI	bence.haromi@gmail.com
Holczer Zsolt	XL9Y0E	zsolti@holczer.net
Kedves Áron Csanád	AV43UW	aron.kedves@gmail.com
Kurdi Barnabás	F0DJNW	barnabas.kurdi@gmail.com
Marton Gergely	FC2WNS	martongergely11@gmail.com

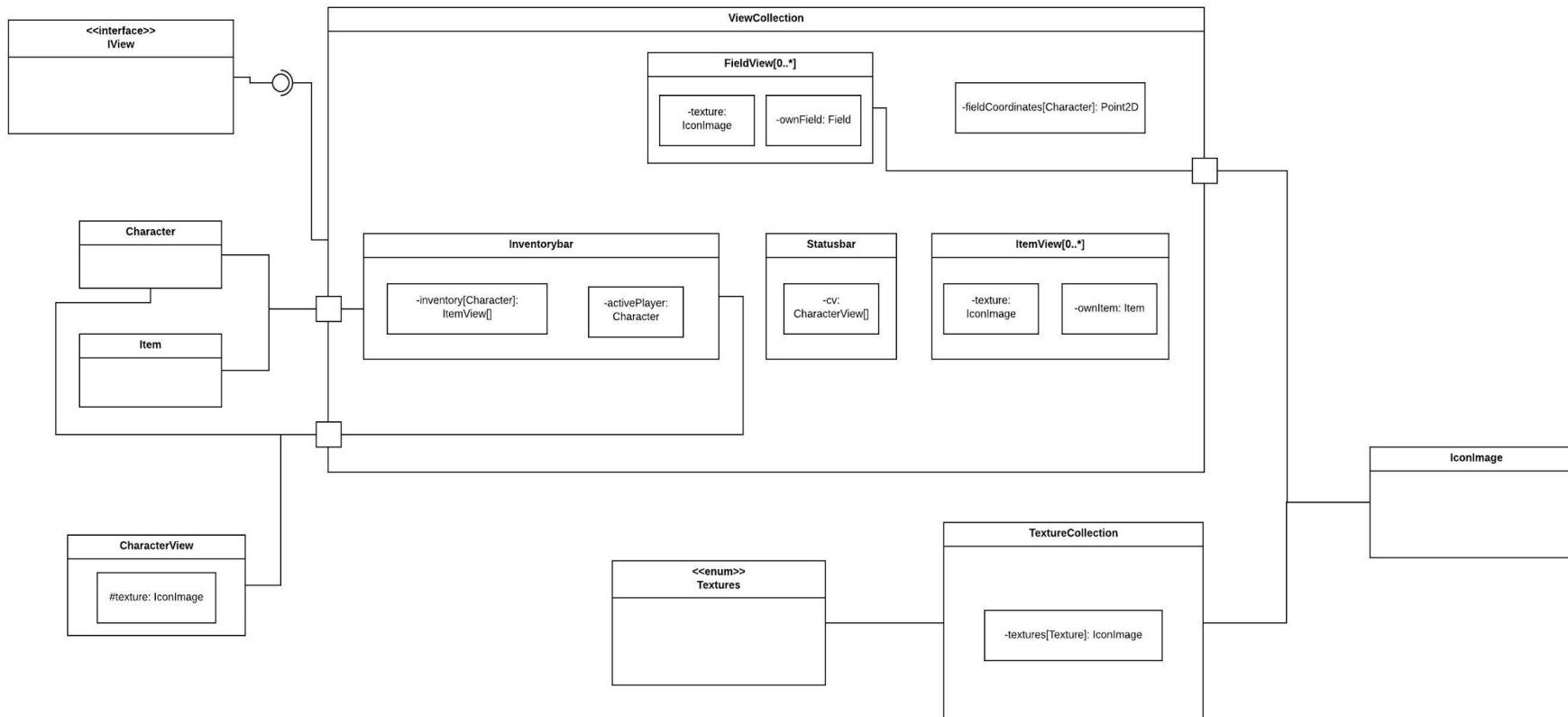
## 11. Grafikus felület specifikációja

### 11.1A grafikus interfész



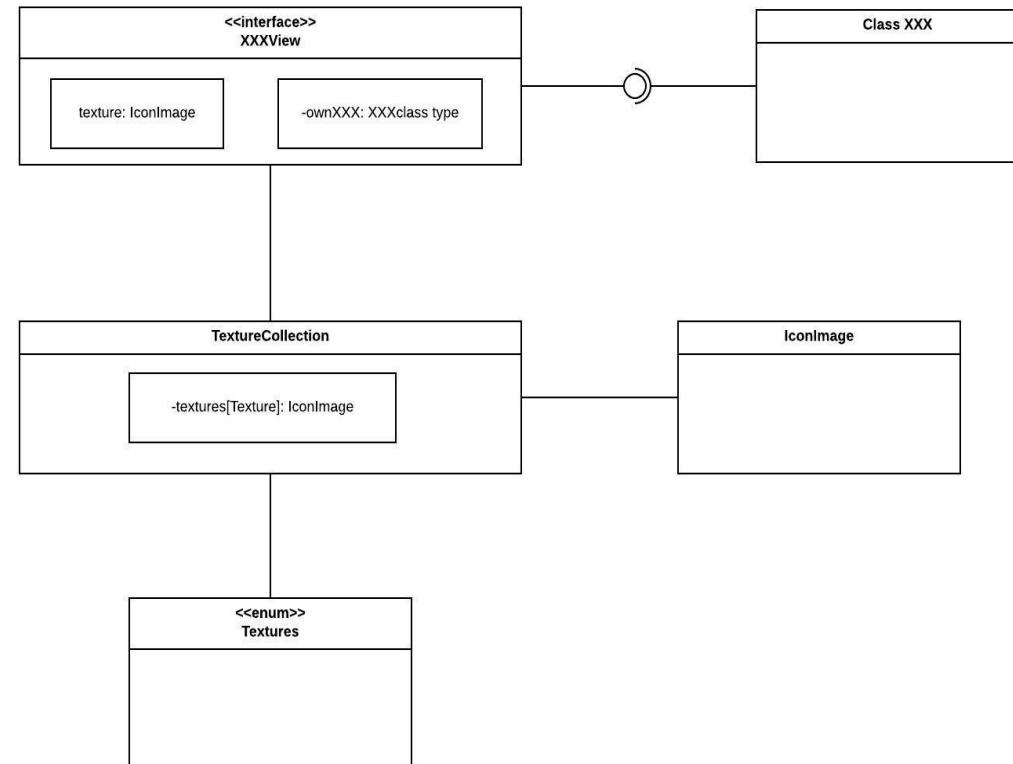
## 11.2 A grafikus rendszer architektúrája

A grafikus rendszer két nagy kollekciója:



Egyes megjelenítendő objektumok kapcsolata:

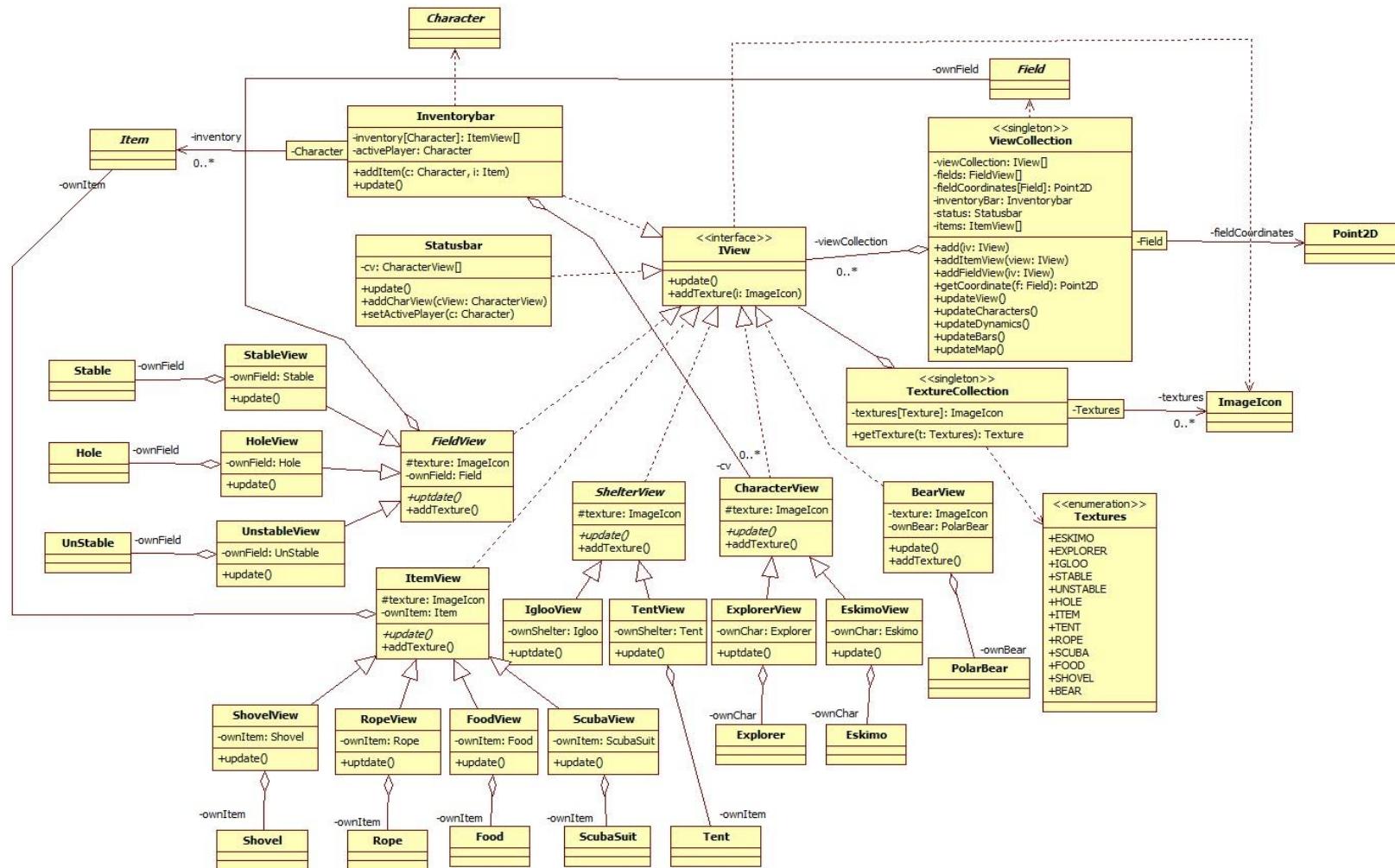
- Magyarázat: egy ponttal lejjebb



### 11.2.1 A felület működési elve

A grafikus felület megvalósítása kevert. Az eddigi modell összes osztályához tartozik egy View osztály (pl. Stable-höz StableView), amely az adott objektum grafikus megjelenítéséért felelős, a View objektumok ismerik a hozzájuk tartozó objektumot. A modell annyiban pull alapú, hogy a kirajzolásnál a View objektumok lekérík a hozzájuk tartozó objektum állapotát, és a kirajzolást ez alapján végzik. Ez azonban push alapon akkor történik meg, amikor az eredeti modell objektumai meghívják a ViewCollection singleton osztály egyes függvényeit (pl. az UpdateDynamics kirajzolja az olyan objektumokat, amelyek nem mezők). A ViewCollection facilitálja a kirajzolást, eltárolja a View objektumokat, és meghívja a megfelelő objektumok update() függvényét. A View objektumok létrehozáskor egy TextureCollection nevű singleton osztálytól kérík le a textúrákat, amelyeket megjeleníthetnek (pl. mező 0-1-2-3-4 hóval). Az, hogy hova rajzolódjanak ki az objektumok a Fieldknél egyértelmű, a FieldView-k létrehozáskor eltárolják azt, hogy milyen koordinátán vannak (Point2D osztály). A dinamikus objektumok pozícióját úgy oldjuk meg, hogy a ViewCollectionben van egy Hashmap, amely Field kulcsokhoz tárol Point2D értékeket, mikor egy View kirajzolódik, lekéri a hozzá tartozó dinamikus objektumhoz tartozó Fieldet (pl. egy játékostól, hogy melyik mezőn áll), majd a ViewCollectiontől azt, hogy ahhoz a Fieldhez milyen kirajzolási koordináta tartozik.

## **11.2.2 A felület osztály-struktúrája**



### **11.3A grafikus objektumok felsorolása**

#### **11.3.1      BearView**

- **Felelősség**

A medvét rajzoló osztály

- **Interfész:** IView

- **Attribútumok**

- **-texture: Image:** Medve képe
- **-ownBear: PolarBear:** A medve akihez tartozik

- **Metódusok**

- **+void update():** medve kirajzolása a helyére
- **+void addTexture(Image i):** beállítja a megadott texturát

#### **11.3.2      CharacterView**

- **Felelősség**

Absztrakt osztály a karakter rajzolásához

- **Interfész:** IView

- **Attribútumok**

- **#texture Image:** a karakter képe

- **Metódusok**

- **+void update():** karakter kirajzolása a helyére
- **+void addTexture(Image i):** beállítja a megadott texturát

#### **11.3.3      EskimoView**

- **Felelősség**

Eszkimó rajzolásért felelős osztály

- **Ősosztályok:** CharacterView → eskimoview

- **Attribútumok**

- **-ownChar: Eskimo:** Eszkimó akit reprezentál

- **Metódusok**

- **+void update():** Eszkimo kirajzolása a helyére

### 11.3.4 ExplorerView

- **Felelősség**

Kutató rajzolásért felelős osztály

- **Ősosztályok:** CharacterView → eskimoview
- **Attribútumok**
  - **ownChar:** Explorer: kutató akit reprezentál
- **Metódusok**
  - **+void update():** Kutató kirajzolása a helyére

### 11.3.5 FieldView

- **Felelősség**

Mezőket rajzoló absztrakt osztály

- **Interfészek:** IView
- **Attribútumok**
  - **#texture Image:** A mező képe
  - **-ownField:** Field: A mező amit reprezentál
- **Metódusok**
  - **+void update():** mező kirajzolása a helyére
  - **+void addTexture(Image i):** beállítja a megadott texturát

### 11.3.6 FoodView

- **Felelősség**

Ételt rajzoló osztály

- **Ősosztályok:** Itemview → FoodView
- **Attribútumok**
  - **-ownItem:** Item: A étel eszköz amit reprezentál
- **Metódusok**
  - **+void update():** kirajzolja az ételt

### 11.3.7 HoleView

- **Felelősség**

Lyukat rajzoló osztály

- **Ősosztályok**

FieldView → HoleView

- **Attribútumok**

- **-ownField: Hole:** A lyuk mező amit reprezentál

- **Metódusok**

- **+void update():** lyuk típusú mező kirajzolása a helyére

### 11.3.8 IgloView

- **Felelősség**

Iglut rajzoló osztály

- **Ősosztályok:** ShelterView → HoleView

- **Metódusok**

- **+void update():** iglu típusú mező kirajzolása a helyére

### 11.3.9 Inventorybar

- **Felelősség**

A karakter eszköz listáját rajzoló osztály

- **Interfészek**

IView

- **Attribútumok**

- **-HashMap<Character, Item[]> inventory:** A karakterek és a hozzá tartozó eszközöket tárolja
- **-Character activePlayer:** az aktív karaktert eltároló attribútum.

- **Metódusok**

- **+void update():** Eszközlista kirajzolása a helyére
- **+void addItem(Character c, Item i):** Hozzáadja a megadott eszközt a megadott karakterhez

### 11.3.10 ItemView

- **Felelősség**

Eszközöket rajzoló osztály

- **Interfészek**

IView

- **Attribútumok**

- **#texture ImageIcon:** az eszköz képe
- **-ownItem: Item:** az eszköz amit reprezentál

- **Metódusok**

- **+void update():** Eszköz kirajzolása a helyére
- **+void addTexture(ImageIcon i):** beállítja a megadott texturát

### 11.3.11 IView

- **Felelősség**

A kirajzoló interfész. minden kirajzoló osztály ezt valósítja meg.

- **Metódusok**

- **+void update():** a kirajzolást végző metódus.
- **+void addTexture(ImageIcon i):** az objektumhoz tartozó textúra hozzárendelése a nézethez.

### 11.3.12 RopeView

- **Felelősség**

A kötél kirajzolásáért felelős osztály.

- **Ősosztályok:** ItemView → RopeView

- **Attribútumok**

- **-Rope ownItem:** a hozzá tartozó Rope objektumot eltároló attribútum.

- **Metódusok**

- **+void update():** ekéri a hozzá tartozó Rope állapotát és az alapján elvégzi a kirajzolást.

### 11.3.13 ScubaView

- **Felelősség**

A búvárruha kirajzolásáért felelős osztály.

- **Ősosztályok:** ItemView → ScubaView

- **Attribútumok**

- **-ScubaSuit ownItem:** a hozzá tartozó ScubaSuit objektumot eltároló attribútum.

- **Metódusok**

- **+void update():** lekéri a hozzá tartozó ScubaSuit állapotát és az alapján elvégzi a kirajzolást.

### 11.3.14 ShelterView

- **Felelősség**

A menedék kirajzolásáért felelős osztály.

- **Interfészek:** IView

- **Attribútumok**

- **#ImageIcon texture:** a menedékhez tartozó textúra.

- **Metódusok**

- **+void addTexture():** a menedékhez tartozó textúra hozzárendelése a menedékhez.
- **+void update():** a menedék kirajzolását végző absztrakt osztály, a leszármazottak valósítják meg.

### 11.3.15 ShovelView

- **Felelősség**

Az ásó kirajzolásáért felelős osztály.

- **Ősosztályok:** ItemView → ShovelView

- **Attribútumok**

- **-Shovel ownItem():** a hozzá tartozó Shovel objektumot eltároló attribútum.

- **Metódusok**

- **+void update():** lekéri a hozzá tartozó Shovel állapotát és az alapján elvégzi a kirajzolást.

### 11.3.16      **StableView**

- **Felelősség**

A stabil mező kirajzolásáért felelős osztály.

- **Ősosztályok:** *FieldView* → *StableView*

- **Attribútumok**

- **-Stable ownField():** a hozzá tartozó Stable objektumot eltároló attribútum.

- **Metódusok**

- **+void update():** lekéri a hozzá tartozó Stable állapotát és az alapján elvégzi a kirajzolást.

### 11.3.17      **Statusbar**

- **Felelősség**

Az éppen aktív karakter tulajdonságait rajzoló osztály.

- **Interfészek:** *IView*

- **Attribútumok**

- **-CharacterView[] cv:** a karakterek nézetét tároló tömb

- **Metódusok**

- **+void update():** a statusbar kirajzolását végzi.
- **+void addCharacterView(CharacterView: cv):** a paraméterként kapott characterViewt hozzáadja a tömbhöz.

### 11.3.18      **TentView**

- **Felelősség**

A sátor kirajzolásáért felelős osztály.

- **Ősosztályok:** *ItemView* → *TentView*

- **Attribútumok**

- **+Tent ownShelter():** a hozzá tartozó Shelter objektumot eltároló attribútum.

- **Metódusok**

- **+void update():** lekéri a hozzá tartozó shelter állapotát és az alapján elvégzi a kirajzolást.

### 11.3.19 TextureCollection

- **Felelősség**

A textúrákat egy hashmapben eltároló szingleton osztály.

- **Attribútumok**

- **-HashMap<Enum, ImageIcon> textures:** a textúrákat tároló kollekció.

- **Metódusok**

- **+Texture getTexture(Textures t):** a paraméterként megadott enumhoz rendelt textúrát adja vissza.

### 11.3.20 Textures

- **Felelősség**

Enumeráció osztály a textúrák kezelésére. A programozás megkönnyítésére szolgál.

- **Attribútumok**

- **+BEAR**
- **+ESKIMO**
- **+EXPLORER**
- **+FOOD**
- **+HOLE**
- **+IGLOO**
- **+ITEM**
- **+ROPE**
- **+SCUBA**
- **+SHOVEL**
- **+STABLE**
- **+TENT**
- **+UNSTABLE**

### 11.3.21 UnstableView

- **Felelősség**

Az instabil mező objektumok kirajzolásáért felelős osztály.

- **Ősosztályok:** FieldView → UnstableView

- **Attribútumok**

- **-UnStable ownField():** a hozzá tartozó UnStable objektumot eltároló attribútum.

- **Metódusok**

- **+void update():** lekéri a hozzá tartozó unstableField állapotát és az alapján elvégzi a kirajzolást.5

- 

### 11.3.22 ViewCollection

- **Felelősség**

A view objektumokat eltároló, illetve azok kirajzolását kezdeményező szingleton osztály.

- **Attribútumok**

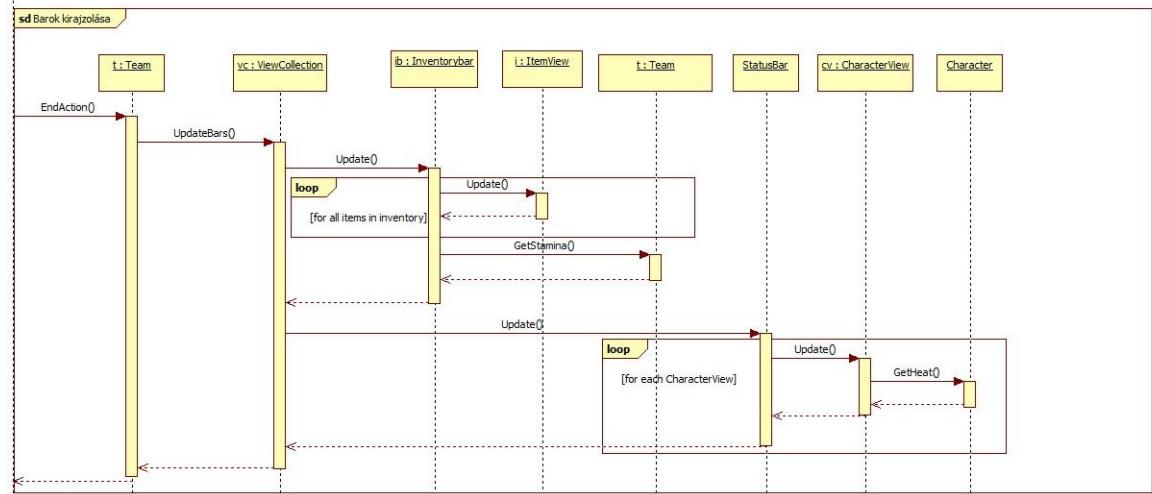
- **-List<IView> viewCollection:** azoknak a kirajzolandó tárgyaknak a kollekciója, melyeknek helyzete változik a játék folyamán.
- **-ArrayList<FieldView> fields:** a fieldek nézeteinek az osztályait tároló kollekció.
- **-HashMap<Field, Point2D> fieldCoordinates:** A fieldekhez tartozó kirajzolási kordinátákat tárolja.
- **-InventoryBar inventoryBar:** az inventorybart tároló attribútum
- **-StatusBar status:** a statusbar-t tároló attribútum
- **-ArrayList<ItemView> items:** az inventorybarban lévő tárgynézeteket tároló kollekció.

- **Metódusok**

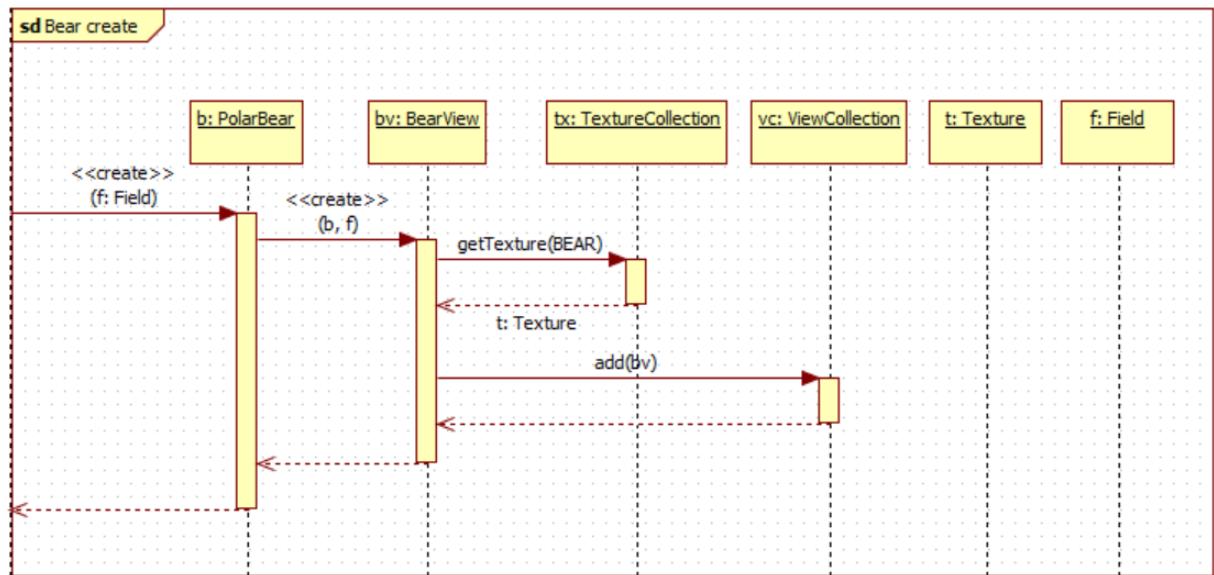
- **+void add(IView view):** a viewcollection-höz a paraméterként megadott IVView objektumot adja ozzá.
- **+void addItemView(IView: iview):** az items ArrayList-hez a paraméterként megadott objektumot adja hozzá.
- **+void addFieldView(IView iview):** a fields ArrayList-hez a paraméterként megadott objektumot adja hozzá.
- **+Point2D getCoordinate(Field f):** a paraméterként megadott field koordinátáját adja vissza a fieldCoordinates hashmap-ból.
- **+void updateView():**
- **+void updateCharacters():** a karakterek pozícióját frissíti.
- **+void updateDynamics():** a nem-mező (nem állandó pozíciójú) objektumokat rajzolja ki.
- **+void updateBars():** a bar-okat rajzolja ki.
- **+updateMap():** a mező objektumokat rajzolja ki.

## 11.4 Kapcsolat az alkalmazói rendszerrel

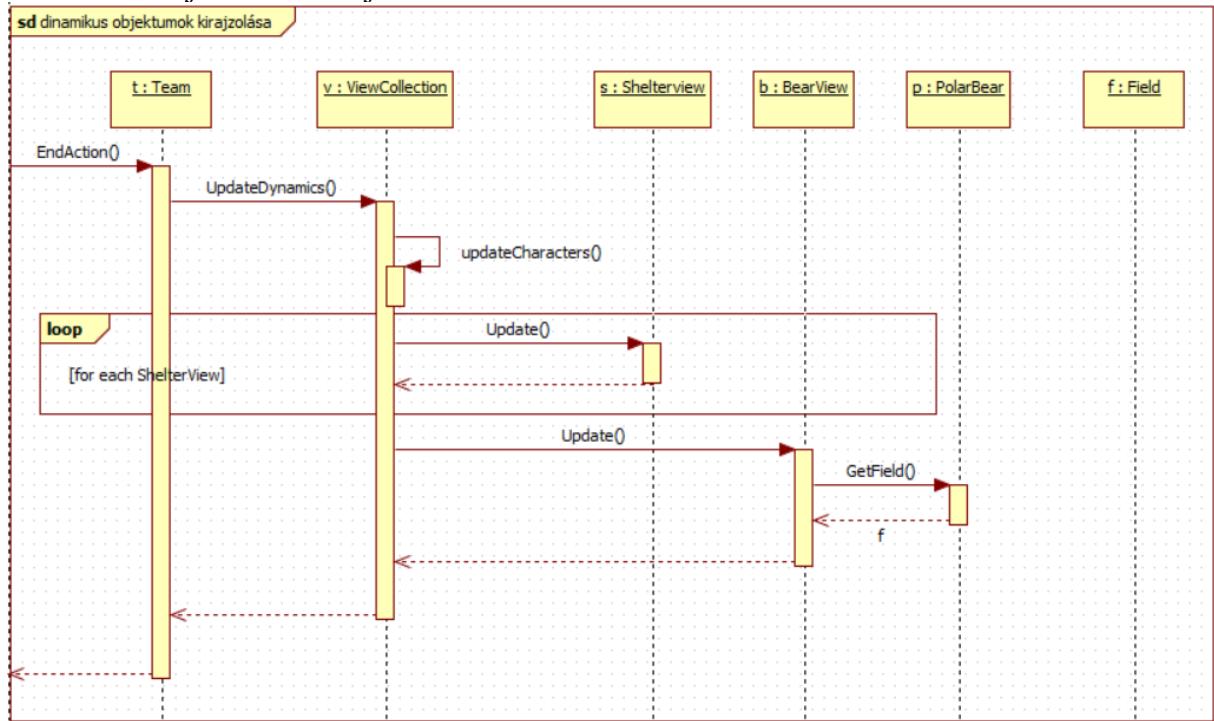
### 11.4.1 Barok update-elése



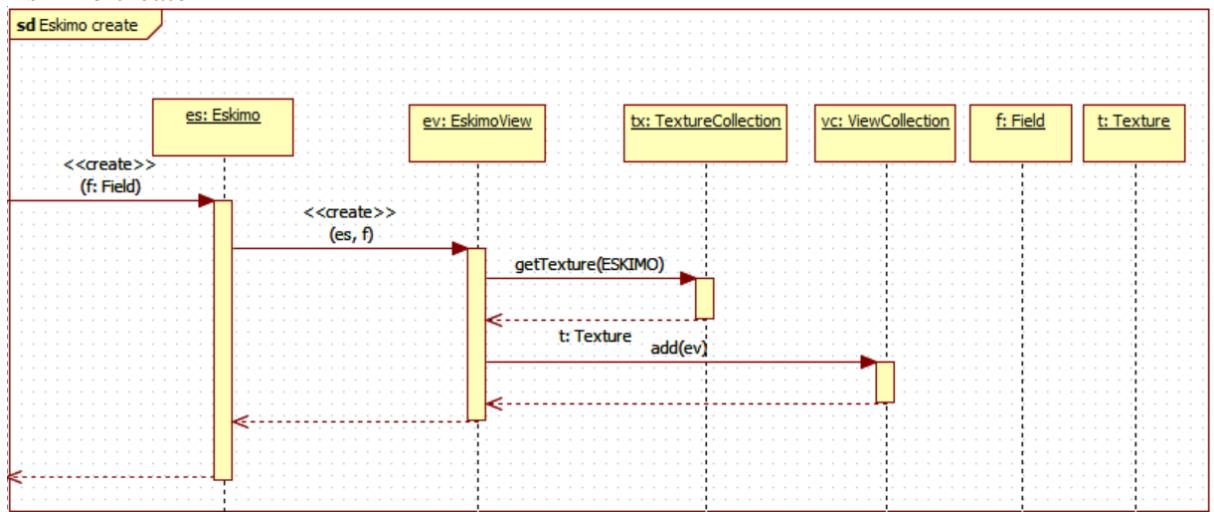
### 11.4.2 Bear create



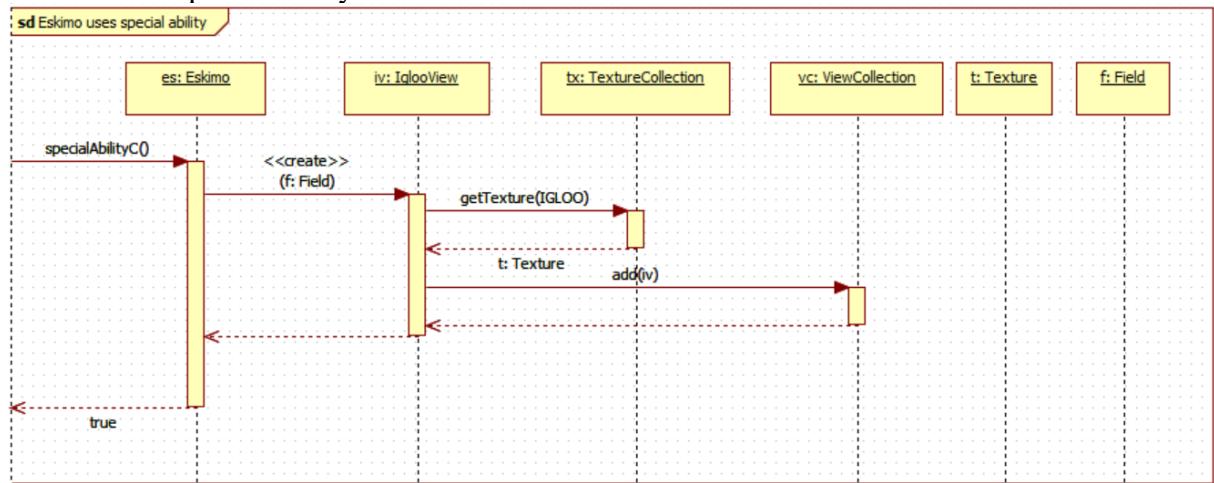
## 11.4.3 Dinamikus objektumok kirajzolása



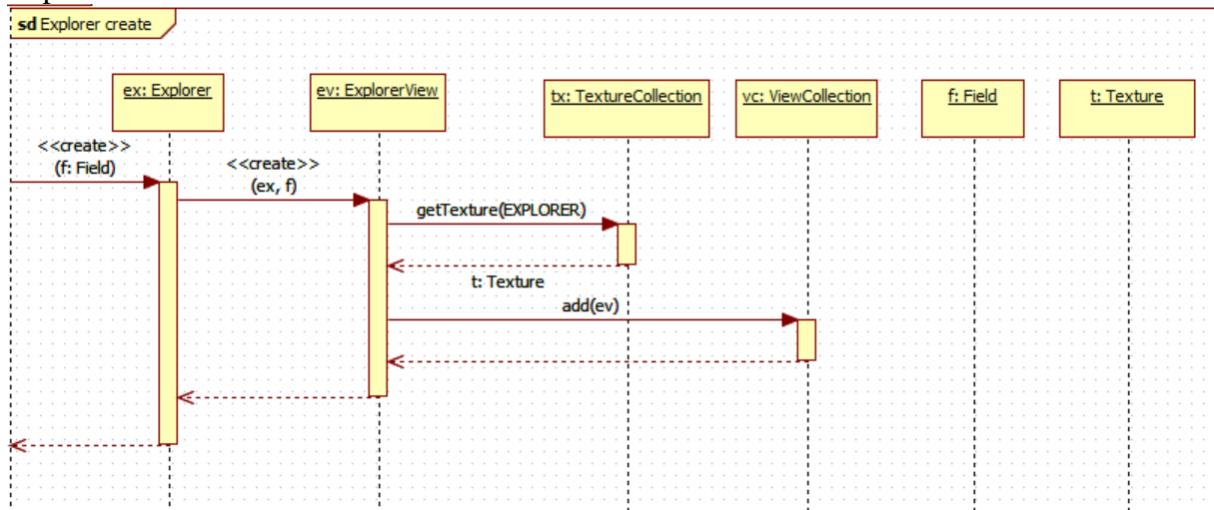
## 11.4.4 Eskimo create



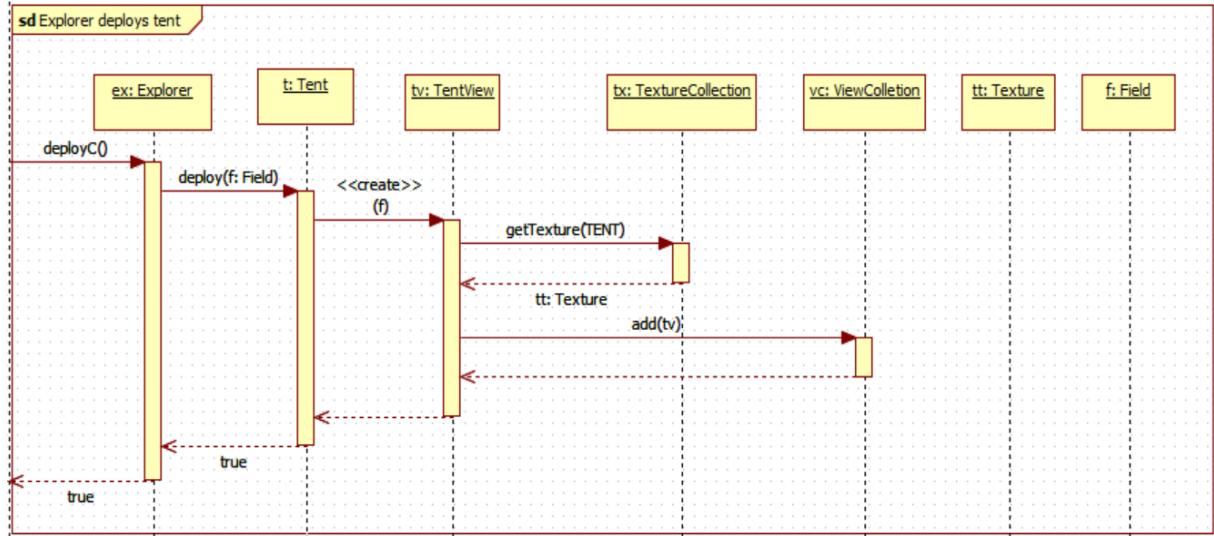
## 11.4.5 Eskimo uses special ability



## 11.4.6 Explorer create

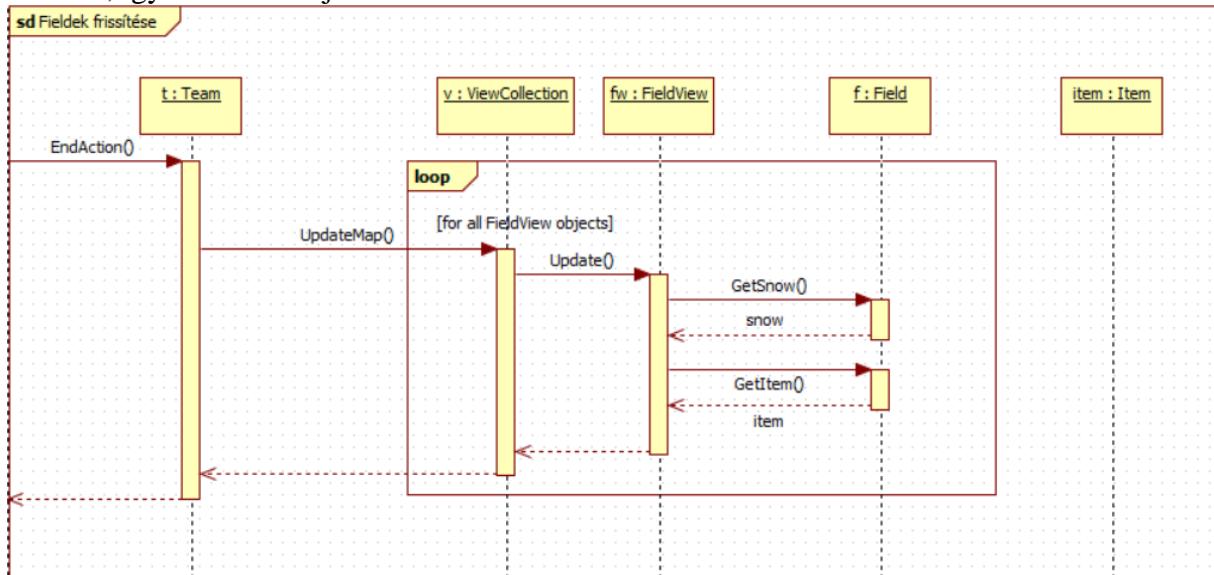


## 11.4.7 Explorer deploys tent

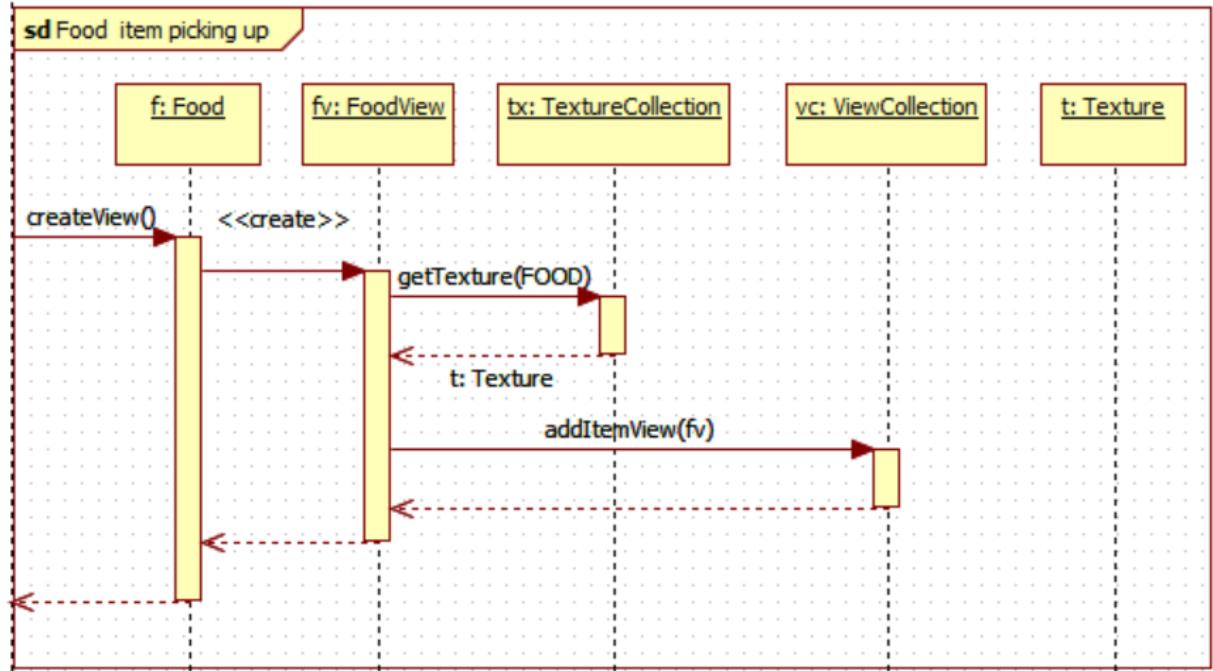


## 11.4.8 Fieldek frissítése

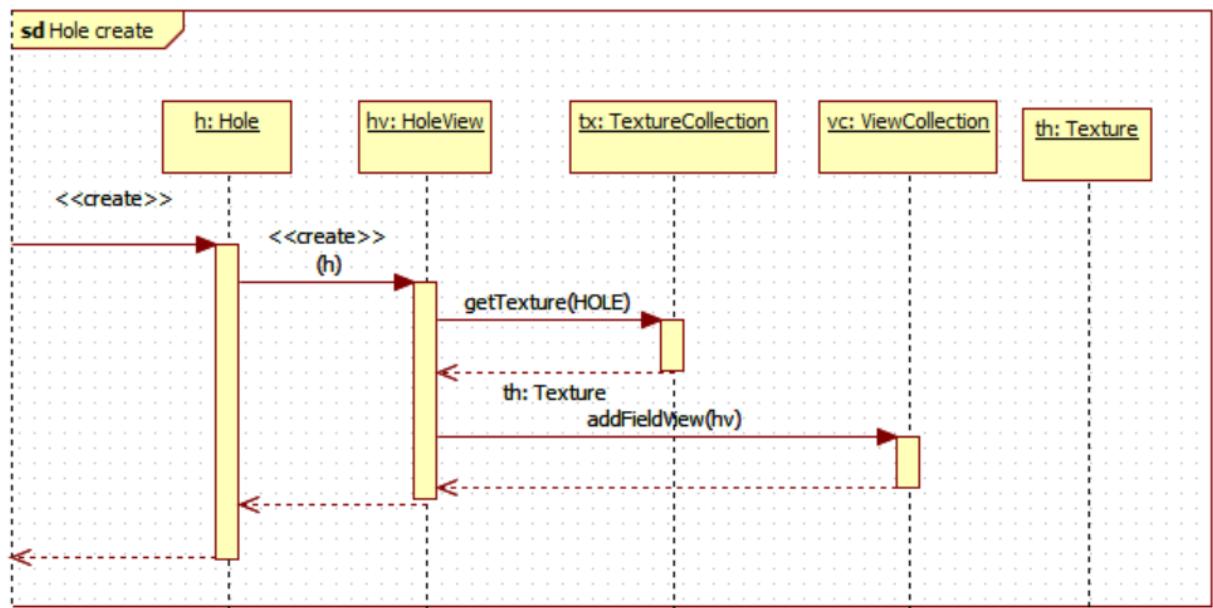
Mj: A Stable-, Unstable- és HoleView frissítése ugyanazokkal a függvényhívásokkal történik, így ezeket nem jelöltük külön



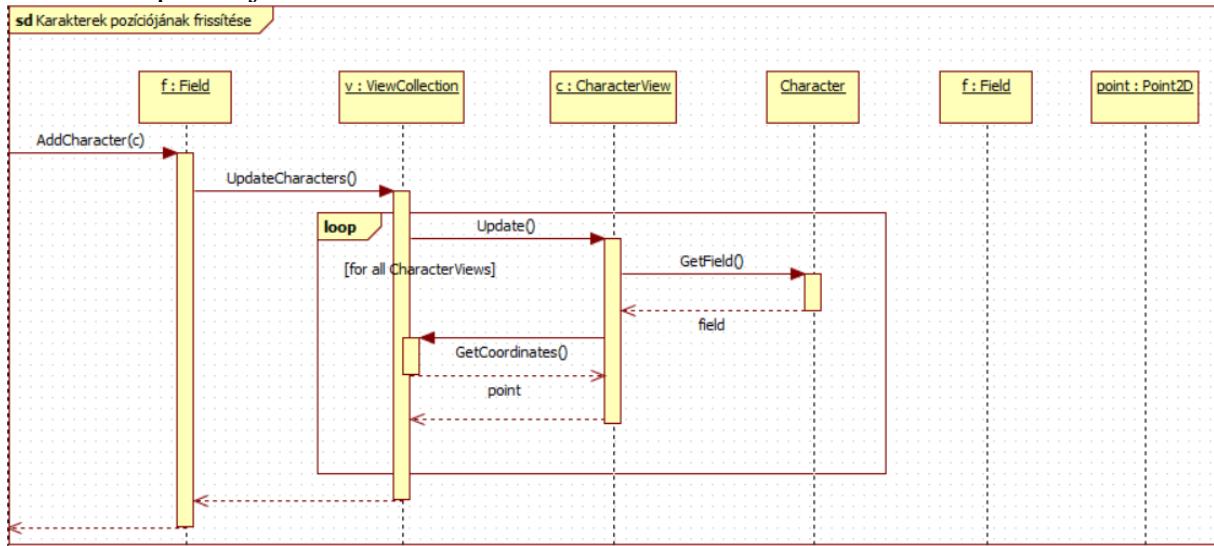
## 11.4.9 Food view creating when it is picked up



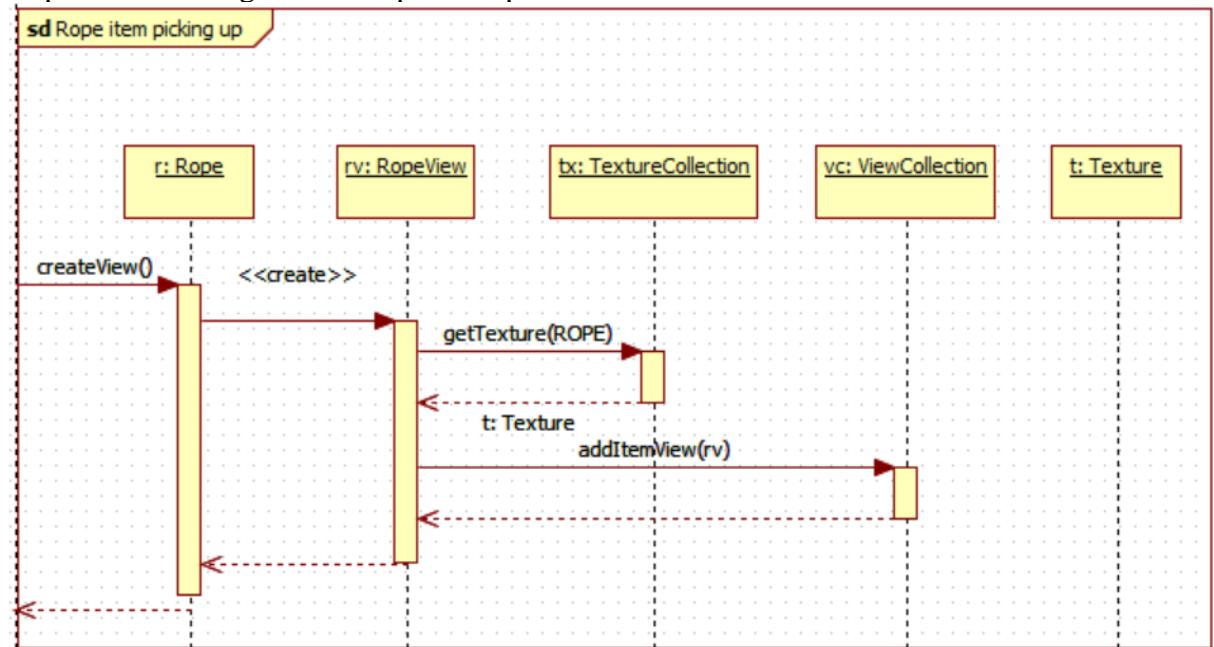
## 11.4.10 Hole create



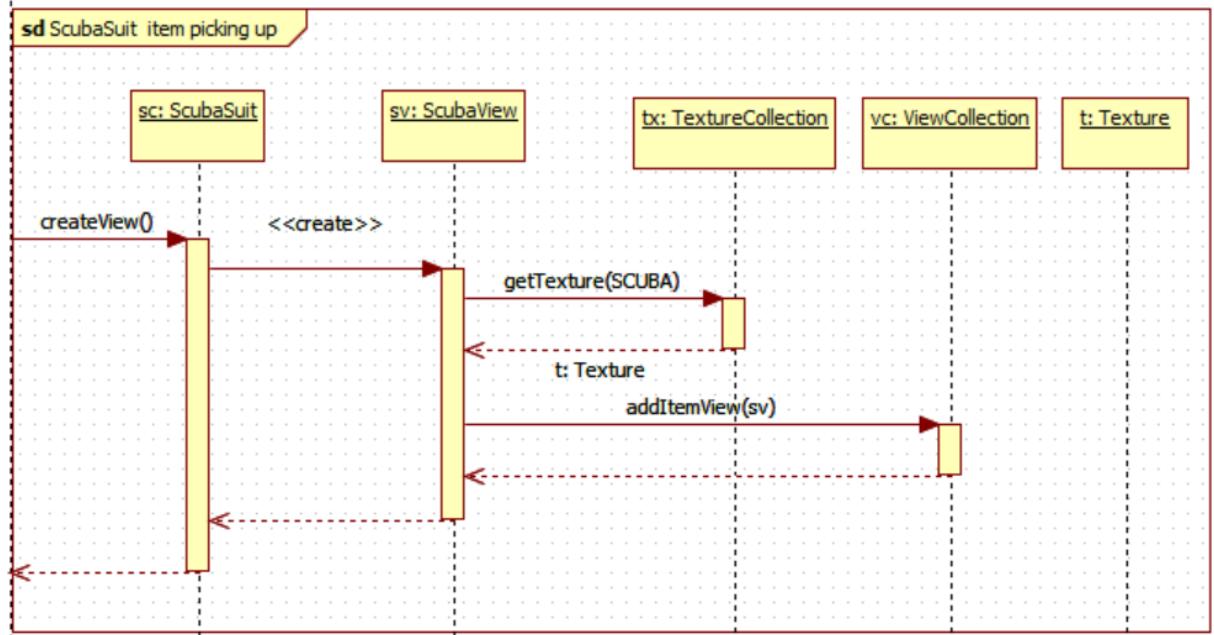
## 11.4.11 Karakterek pozíciójának frissítése



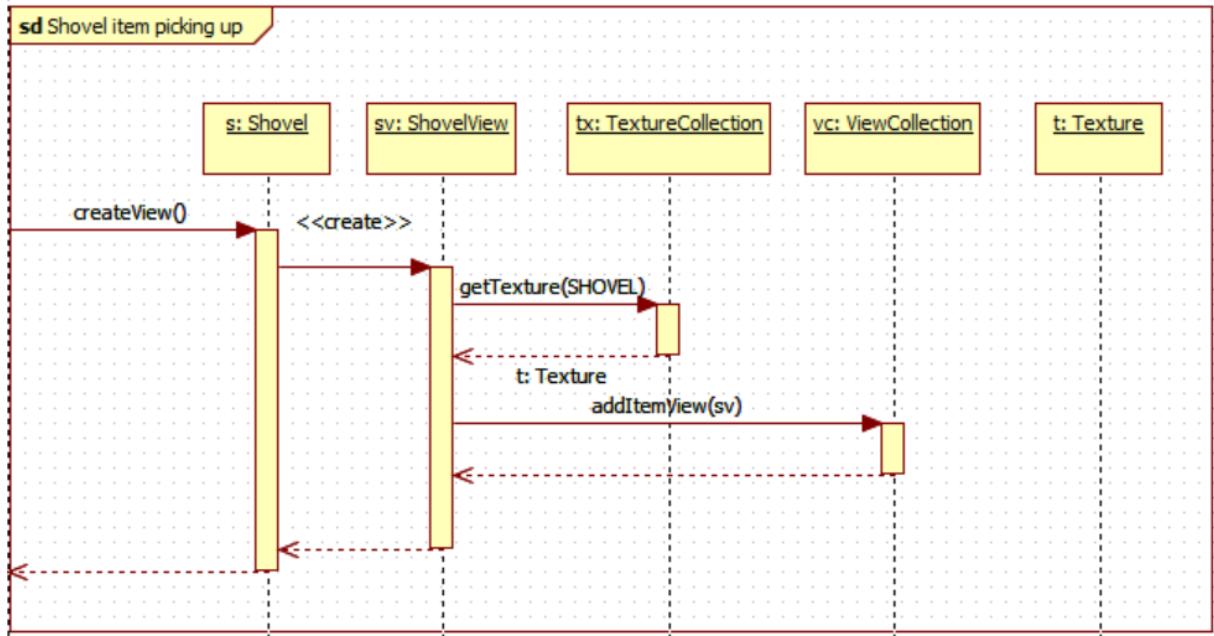
## 11.4.12 Rope view creating when it is picked up



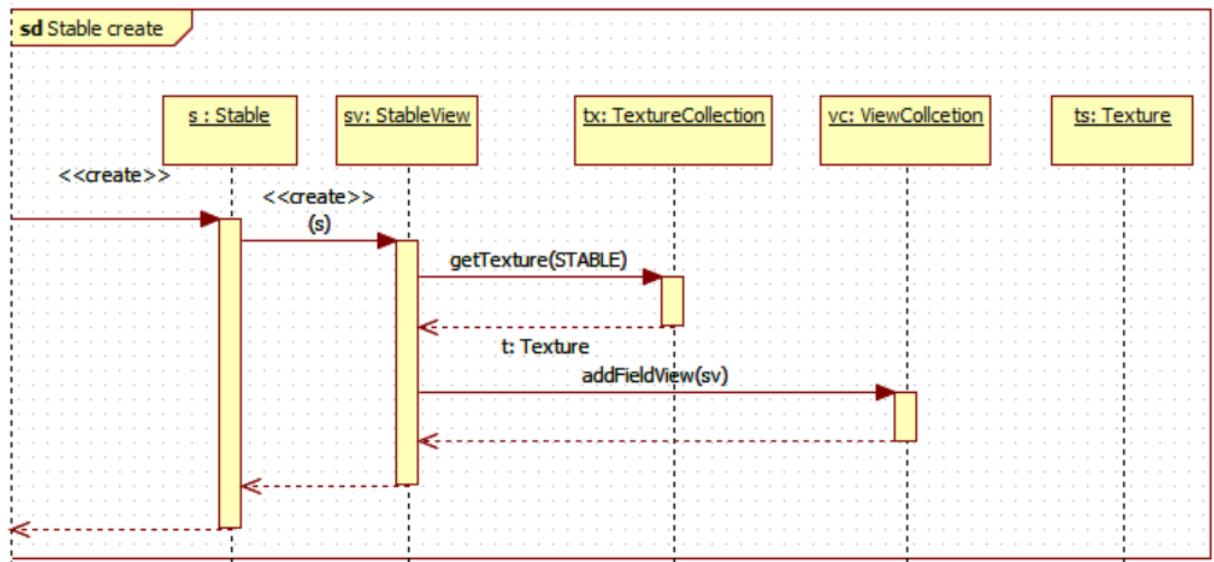
## 11.4.13 ScubaSuit view creating when it is picked up



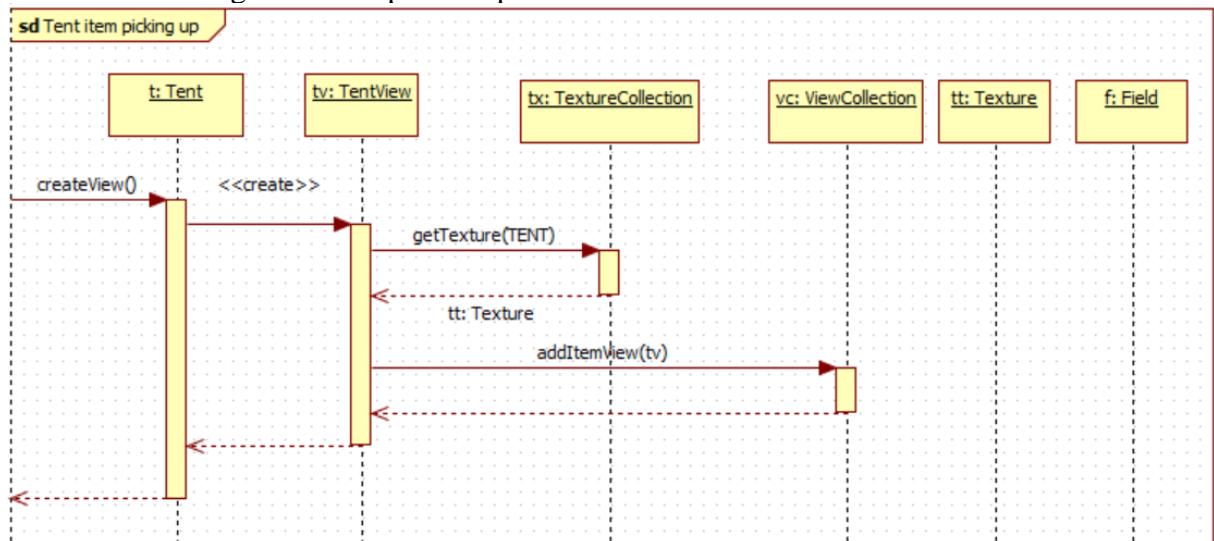
## 11.4.14 Shovel view creating when it is picked up



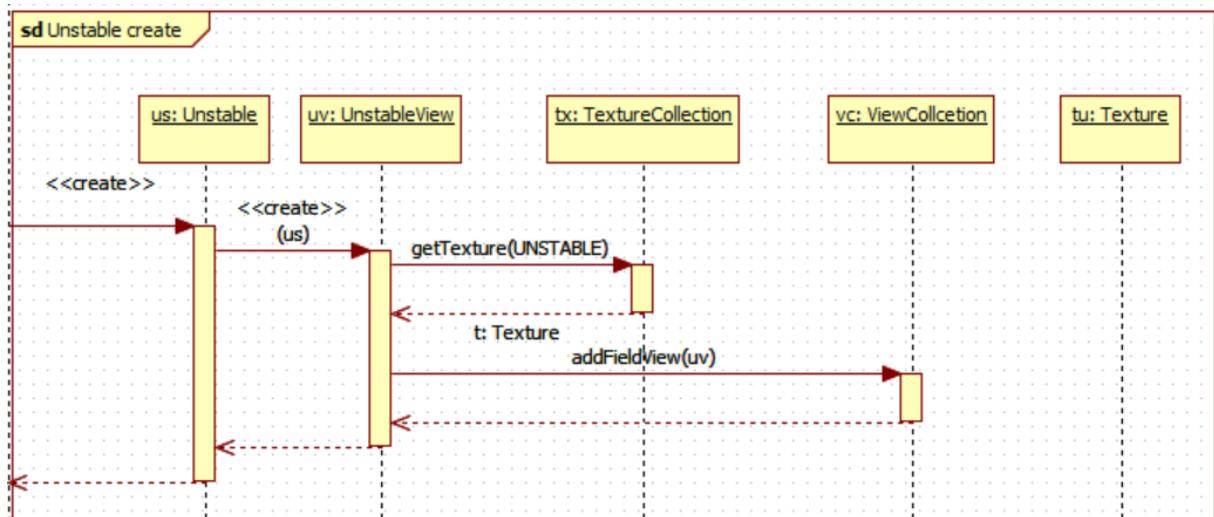
## 11.4.15 Stable create



## 11.4.16 Tent view creating when it is picked up



## 11.4.17 Unstable create



## 11.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.04.29. 16:00	2 óra	Háromi Holczer Kedved Kurdi Marton	Értekezlet. Döntés: Marton elkészíti az osztálydiagramot, Kedves és Kurdi a szekvencia-diagramokat, Háromi és Holczer az osztályok leírását 2020.05.01. 18:00 időpontra.
2020.04.30. 19:00	1 óra	Háromi	Tevékenység: Háromi elkezdi az osztályok leírását.
2020.04.30. 11:00	1 óra	Marton	Tevékenység: Marton megtervez egy kezdetleges osztálydiagramot.
2020.04.30 20:00	1 óra	Holczer	Tevékenység: Holczer elkezdi az osztály leírásait.
2020.05.01 11:00	1 óra	Kedves	Tevékenység: Kedves elkészíti a Eskimo c., Explorer c., Eskimo special ability, Stable c., Unstable c., Hole c., Stable c. with item szekvencia diagramokat.
2020.05.01 14:00	1 óra	Kedves	Tevékenység: Kedves elkészíti a tent, rope, scubasuit, bear, food create és a deploy tent szekvencia diagramjait
2020.05.01. 14:30	1 óra	Kurdi	Tevékenység: Kurdi elkészíti az update-es szekvenciadiagramokat
2020.05.01 15:00	1 óra	Marton	Tevékenység:

			Marton javít az elkészített diagramon
2020.05.01 18:00	2 óra	Háromi Holczer Kedved Kurdi Martón	Értekezlet.
2020.05.01. 21:00	0,5 óra	Háromi	Tevékenység: Háromi befejezi a rá kiszabott osztályleírásokat.
2020.05.02 10:00	1 óra	Kedves	Tevékenység: Keves javítja a rá kiszabott szekvencia diagramokat.
2020.05.02 14:00	1 óra	Háromi Holczer Kedved Kurdi Martón	Értekezlet
2020.05.02 20:00	0,5 óra	Martón	Tevékenység: Martón javít az osztálydiagrammon
2020.05.02.21:00	0,5 óra	Háromi	Tevékenység: Háromi javítja az osztályleírásokat.
2020.05.03. 9:00	1 óra	Kurdi	Tevékenység: Kurdi elkészíti a grafikus felület leírását és elkészíti a grafikus interfész kezdetleges képét
2020.05.03 10:00	1 óra	Martón	Tevékenység: Martón befejezi az osztálydiagrammot
2020.05.03. 12:00	1.5 óra	Holczer	Tevékenység: Holczer javítja a rá kiszabott osztály leírásokat
2020.05.03 13:00	1,5 óra	Martón	Tevékenység: Martón kutat, hogy hogyan kell megcsinálni a struktúra diagramot, majd elkészíti azt.
2020.05.03. 16:00	1 óra	Háromi Holczer Kedved Kurdi	Tevékenység: megbeszélés.

		Marton	
2020.05.17:00	1 óra	Háromi	Tevékenység: Háromi javítja az osztályleírásokat, megszerkeszti a dokumentumot.

# 13 Grafikus változat beadása

## 45 – nullpointer

Konzulens:  
Ludmány Balázs

### Csapattagok

Háromi Bence	AK51GI	bence.haromi@gmail.com
Holczer Zsolt	XL9Y0E	zsolti@holczer.net
Kedves Áron Csanád	AV43UW	aron.kedves@gmail.com
Kurdi Barnabás	F0DJNW	barnabas.kurdi@gmail.com
Marton Gergely	FC2WNS	martongergely11@gmail.com

## 13.1 Fordítási és futtatási útmutató

### 13.1.1 Fájllista

Fájl neve	Méret (bytes)	Keletkezés ideje	Tartalom
eskimo1.png	677	2020.05.11.20:55	A játéklemekhez tartozó kép
eskimo2.png	682	2020.05.11.20:55	A játéklemekhez tartozó kép
eskimo3.png	676	2020.05.11.20:55	A játéklemekhez tartozó kép
eskimo4.png	673	2020.05.11.20:55	A játéklemekhez tartozó kép
eskimo5.png	692	2020.05.11.20:55	A játéklemekhez tartozó kép
eskimo6.png	665	2020.05.11.20:55	A játéklemekhez tartozó kép
explorer.png	606	2020.05.11.20:55	A játéklemekhez tartozó kép
explorer1.png	606	2020.05.11.20:55	A játéklemekhez tartozó kép
explorer2.png	595	2020.05.11.20:55	A játéklemekhez tartozó kép
explorer3.png	602	2020.05.11.20:55	A játéklemekhez tartozó kép
explorer4.png	600	2020.05.11.20:55	A játéklemekhez tartozó kép
explorer5.png	610	2020.05.11.20:55	A játéklemekhez tartozó kép
explorer6.png	599	2020.05.11.20:55	A játéklemekhez tartozó kép
food.png	388	2020.05.11.20:55	A játéklemekhez tartozó kép
gunpart1.png	607	2020.05.17.20:23	A játéklemekhez tartozó kép
gunpart2.png	473	2020.05.17.20:23	A játéklemekhez tartozó kép
gunpart3.png	464	2020.05.17.20:23	A játéklemekhez tartozó kép
holefield1.png	38267	2020.05.16.13:38	A játéklemekhez tartozó kép
holefield2.png	38422	2020.05.16.13:38	A játéklemekhez tartozó kép
holefield3.png	38490	2020.05.16.13:38	A játéklemekhez tartozó kép
holefield4.png	38496	2020.05.16.13:38	A játéklemekhez tartozó kép
holefield5.png	38546	2020.05.16.13:38	A játéklemekhez tartozó kép
igloo.png	506	2020.05.11.20:55	A játéklemekhez tartozó kép
polarbear.png	478	2020.05.11.20:55	A játéklemekhez tartozó kép
rope.png	557	2020.05.11.20:55	A játéklemekhez tartozó kép

scuba.png	506	2020.05.11.20:55	A játékelemekhez tartozó kép
shovel.png	446	2020.05.11.20:55	A játékelemekhez tartozó kép
stablefield1.png	39438	2020.05.16.13:38	A játékelemekhez tartozó kép
stablefield2.png	38267	2020.05.16.13:38	A játékelemekhez tartozó kép
stablefield3.png	38422	2020.05.16.13:38	A játékelemekhez tartozó kép
stablefield4.png	38490	2020.05.16.13:38	A játékelemekhez tartozó kép
stablefield5.png	38496	2020.05.16.13:38	A játékelemekhez tartozó kép
stablefield6.png	38546	2020.05.16.13:38	A játékelemekhez tartozó kép
tent.png	425	2020.05.11.20:55	A játékelemekhez tartozó kép
unstablefield1.png	39438	2020.05.16.13:38	A játékelemekhez tartozó kép
unstablefield2.png	38267	2020.05.16.13:38	A játékelemekhez tartozó kép
unstablefield3.png	38422	2020.05.16.13:38	A játékelemekhez tartozó kép
unstablefield4.png	38490	2020.05.16.13:38	A játékelemekhez tartozó kép
unstablefield5.png	38496	2020.05.16.13:38	A játékelemekhez tartozó kép
unstablefield6.png	38546	2020.05.16.13:38	A játékelemekhez tartozó kép
Inventorybar.java	3665	2020.05.16.13:42	Az aktív státuszú játékos inventoryját illetve stamináját, és képét jeleníti meg.
Statusbar.java	2167	2020.05.16.14:29	A teljes csapat alapadatait jeleníti meg.
BearView.java	1944	2020.05.16.13:38	Egy jegesmedvét jelenít meg.
CharacterView.java	1084	2020.05.17.20:23	A játékosok megjelenítésének űsosztálya
EskimoView.java	2393	2020.05.17.20:23	A eszkimók megjelenítéséért felelős osztály
ExplorerView.java	2396	2020.05.17.20:23	A felfedezők megjeénenítéséért felelős osztály
FieldView.java	575	2020.05.17.20:23	A mezők megjelenítésének űsosztálya
HoleView.java	2378	2020.05.17.20:23	A lyukak megjelenítésére

StableView.java	2146	2020.05.17.20:23	A stabil mezők megjelenítésére
UnstableView.java	2083	2020.05.17.20:23	Az instabil mezők megjelenítésére
FlareGunPartView.java	1666	2020.05.16.13:38	A jelzőpisztoly darabok megjelenítésére
FoodView.java	1492	2020.05.16.13:38	A ételek megjelenítésére
IglooView.java	1488	2020.05.16.13:38	A igluk megjelenítésére
ItemView.java	900	2020.05.16.13:38	A tárgyak megjelenítésének ősosztálya
RopeView.java	1412	2020.05.16.13:38	A kötelek megjelenítésére
ScubaView.java	1519	2020.05.16.13:38	A búvárruha megjelenítésére
ShelterView.java	742	2020.05.16.13:38	A menedékek megjelenítésének ősosztálya
ShovelView.java	1524	2020.05.16.13:38	A ások megjelenítésére
TentItemView.java	1502	2020.05.16.13:38	A sátrak megjelenítésére az Inventorybarban
TentView.java	1802	2020.05.16.13:38	A sátrak megjelenítésére
ArrowListener.java	2981	2020.05.13.20:20	Egy KeyListener az ablakok kezelésére
PlayerSettings.java	2470	2020.05.16.14:46	A játékosok beállítására
ContainerView.java	3763	2020.05.17.20:23	A konténerek megjelenítésére
IView.java	121	2020.05.10.21:13	Interfész a megjelenítéshez
TextureCollection.java	4747	2020.05.17.20:23	A textúrák homogén kollekciója
Textures.java	147	2020.05.11.13:07	enum a textúrák rendszerezésére
ViewCollection.java	6212	2020.05.17.20:23	A megjeleníthető objektumok heterogén kollekciója
ViewController.java	6546	2020.05.17.20:23	A singleton osztály az adatok kezelésére

### 13.1.2 Fordítás és telepítés

Első lépésként hozzunk létre egy új projektet az INTELLIJ IDEA-n belül (miután elindítottuk a programot: “+” jel → Create New Project)

Amint kész van a projektünk a létrejött src mappát a gépünk bármely filekezelő programját használva (pl. Total Commander) írjuk felül a kapott file-ok közül a **Code\graf\src** mappával. Persze megtehetjük, hogy a src-ból az összes filet kimásoljuk a projektünk src mappájába. Ezen felül még szükségünk lesz átmásolni a graf/images mappát is, amely a megjelenítéshez szükséges **.png** fileokat tartalmazza. Ezt másoljuk át a projektünk **gyökér** mappájába.

Ezután az ablakunk jobb felső sarkában nyomjunk a zöld kalapács ikonra (), vagy

nyomjunk Shift+F9-et, így fordíthatjuk a programunkat. Ha esetleg olyan hibaüzenetet kapnánk, hogy a program nem találja a main() függvényt, akkor a kalapács melletti kis boxban nyomjunk Edit Configurations...-t (), majd adjuk meg, hogy a

projektünk Main class-a melyik osztályban található.

### 13.1.3 Futtatás

A futtatás innen már nagyon egyszerű. Egyszerűen adjuk ki a Shift+F10 billentyűzetkombinációt, vagy nyomjunk a ikonra. Mivel a programunk már tudja, hogy

hol a main() függvény, így futtatni is tudja a programot. Ha esetleg hiba lépne fel, mert a program nem találja a main() függvényt, egyszerűen a Project Explorerben a Main osztályunkunk nyomjunk egy Run ‘Main.main()’ gombot, és már kész is!

## 13.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Háromi Bence	AK51GI	20
Holczer Zsolt	XL9Y0E	20
Kedves Áron Csanád	AV43UW	20
Kurdi Barnabás	F0DJNW	20
Marton Gergely	FC2WNS	20

### 13.3 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.05.06 16:00	1 óra	Háromi Holczer Kurdi Marton Kedves	<b>Megbeszélés:</b> <b>Döntés:</b> Marton készíti a kezdőmenüt, az inventory és a statusbарт Holczer készíti a Character és FieldView osztályokat Kurdi készíti a játék elkezdését, Team és Map osztályok frissítését, A ViewCollection egy részét Háromi készíti a ShelterView, BearView és item View osztályokat Kedves készíti a ViewCollection egy részét, a TextureCollection és Textures osztályokat
2020.05.06 17:00	1 óra	Kedves	<b>Tevékenység:</b> Main osztály átalakítása, Textures enum leírása, ViewController osztály létrehozása, ViewCollection osztály készítése, TextureCollection osztály elkészítése
2020.05.08 14:00	1 óra	Háromi	<b>Tevékenység:</b> Háromi elkészíti a BearView, ItemView és abból származó osztályokat.
2020.05.08 15:00	2 óra	Kurdi	<b>Tevékenység:</b> Kurdi elkezdi team és map osztályok átalakításának
2020.05.08 16:00	2 óra	Kedves	<b>Tevékenység:</b> Textures enum, ViewController, ViewCollection és TextureCollection javítása
2020.05.08 17:00	2 óra	Marton	<b>Tevékenység:</b> Marton elkezdi a kezdőmenü létrehozását
2020.05.08 18:00	3 óra	Holczer	<b>Tevékenység:</b> Holczer létrehozza a CharacterView és FieldView osztályokat és nekilát azoknak a kidolgozásának
2020.05.09 10:00	1 óra	Háromi	<b>Tevékenység:</b> Háromi megvalósítja a ShelterView, TentView, IgloView
2020.05.09 16:00	2 óra	Háromi Holczer Kurdi Marton Kedves	<b>Megbeszélés:</b> <b>Döntés:</b> ViewController osztályt el kell készíteni és minden ennek megfelelően frissíteni kell
2020.05.09 18:00	2 óra	Holczer	<b>Tevékenység:</b>

			Holczer kidolgozza a számára kiszabott osztályokat
2020.05.09 18:00	1 óra	Kedves	<b>Tevékenység:</b> Tesztképek elkészítése, kép betöltés megírása és tesztelése
2020.05.09 18:00	3 óra	Kurdi	<b>Tevékenység:</b> Kurdi befejezi a team és map fejlesztését, a játék automatikus kezdésének megalkotása, inicializálások elkészítése
2020.05.09 19:00	5 óra	Holczer	<b>Tevékenység:</b> Holczer elkezdi a programrészek összerakása után felmerülő problémák javítása
2020.05.09 18:00	2 óra	Háromi	<b>Tevékenység:</b> Háromi a megbeszélteknek megfelelően átdolgozza a saját osztályait.
2020.05.10 11:00	1 óra	Kedves	<b>Tevékenység:</b> ViewController, ViewCollection, TextureCollection osztályok írása és javítása
2020.05.10 11:00	2,5 óra	Háromi Holczer Kurdi Marton Kedves	<b>Megbeszélés:</b> <b>Döntés:</b> A view modell újradolgozása, mindenkinél frissítenie kell, amiket eddig megcsinált
2020.05.10 14:00	2 óra	Holczer	<b>Tevékenység:</b> Holczer átdolgozza a Character és Field view osztályokat
2020.05.10 14:00	2 óra	Kurdi	<b>Tevékenység:</b> Kurdi átalakítja és javítja a rá kiszabott osztályokat és függvényeket
2020.05.10 15:00	4 óra	Marton	<b>Tevékenység:</b> Marton befejezi a kezdő menü megalkotását, beépíti azt a kódba és elkezdi a StatusBar és az Inventorybar létrehozását
2020.05.10 19:00	4 óra	Holczer	<b>Tevékenység:</b> Holczer javítja a módosítások miatt fellépő problémákat
2020.05.12 16:00	3 óra	Marton	<b>Tevékenység:</b> Marton ellenőrzi a kódot, és javít a minőségén, kommenteket készít
2020.05.13 10:00	4.5 óra	Marton	<b>Tevékenység:</b> Marton további javításokat végez, kommentel, javít pár funkciót.
2020.05.13 15:00	2 óra	Holczer	<b>Tevékenység:</b> Holczer javításokat végez az osztályain, valamint megkezdi az osztályok kommentelését

2020.05.15 12:00	3 óra	Marton	<b>Tevékenység:</b> Marton átdolgozza a status és inventory bart, majd megvalósítja azokat
2020.05.15 17:00	1 óra	Háromi	<b>Tevékenység:</b> Háromi cseréli a textúrákat, átméretezi az ablakot, teszteli a funkciók megfelelő működését, hibákat keres, a templateket elkezdi kitölteni.
2020.05.16. 10:00	0.5 óra	Háromi	<b>Tevékenység:</b> Háromi javításokat végez, kommenteli az általa elkészített osztályokat.
2020.05.16. 10:30	2.5 óra	Marton	<b>Tevékenység:</b> Marton befejezi és kommenteli a rá kiszabott kód részeket
2020.05.16 11:00	1 óra	Kurdi	<b>Tevékenység:</b> Javításokat végez a számára kijelölt megoldásokon a tesztek során szerzett tapasztalatok alapján
2020.05.16. 14:00	3 óra	Háromi Holczer Kurdi Marton Kedves	<b>Tevékenység:</b> <b>Döntés:</b> A csapat átvizsgálja a kódot és a készült dokumentációt, végül megegyeznek, hogy az elkészült termék megfelel az elvárásoknak.

# 14. Összefoglalás

## 45 – nullpointer

Konzulens:  
Ludmány Balázs

### Csapattagok

Háromi Bence	AK51GI	bence.haromi@gmail.com
Holczer Zsolt	XL9Y0E	zsolti@holczer.net
Kedves Áron Csanád	AV43UW	aron.kedves@gmail.com
Kurdi Barnabás	F0DJNW	barnabas.kurdi@gmail.com
Marton Gergely	FC2WNS	martongergely11@gmail.com

## 14.1 A projektre fordított összes munkaidő

Tag neve	Munkaidő (óra)
Háromi Bence	102
Holczer Zsolt	102
Kedves Áron Csanád	102
Kurdi Barnabás	102
Marton Gergely	102
<b>Összesen</b>	<b>510</b>

- A feltöltött programok forrássorainak száma**

Fázis	Kódsorok száma
Szkeleton	1195
Prototípus	1973
Grafikus változat	3554
<b>Összesen</b>	<b>6722</b>

## 14.2 • Projekt összegzés

### 14.2.1 Mit tanultak a projektből konkrétan és általában?

Megtanultuk, mennyire nehéz a csapatmunka, mégis, amikor működik, nagyon jó, hogy ki tudják egészíteni egymás hiányosságait a csapattagok. Kiderült, hogy egy aránylag bonyolult, grafikus felülettel is rendelkező játék elkészítése nem lehetetlen egy félév alatt, a többi tárgy teljesítése mellett sem. Rájöttünk, mennyire fontos, hogy senki ne hozzon egyedül a teljes projektet befolyásoló döntéseket, és hogy transzparens legyen, hogy a csapatból ki mivel foglalkozik.

### 14.2.2 Mi volt a legnehezebb és a legkönnyebb?

A legnehezebb általánosságban a modellek megtervezése volt, a megvalósítás viszonylag gördülékenyen ment. Azonban nehézségekbe ütközünk, mikor a különböző csapattagok által megírt modulokat kellett összeilleszteni, ilyenkor mindenkor sok órán keresztül kellett debugolnunk. A git kezelésével is akadtak problémáink, de a projekt végére (majdnem) mindenki megtanulta kezelni. A projekt elkészítésének legkönnyebb aspektusa talán a dokumentációk elkészítése volt, ezzel soha nem akadt problémánk.

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

Összességében igen, elég idő volt az összes feladat elkészítésére, azonban a feladat megvalósítása érhetne több pontot a modell megtervezésénél.

**14.2.4 Ha nem, akkor hol okozott ez nehézséget?**

Semmiilyen problémánk nem merült fel ezzel kapcsolatban.

**14.2.5 Milyen változtatási javaslatuk van?**

Ez valószínűleg nem megvalósítható, csak egy ötlet: lehetne minden beadás előtt egy lehetőség arra, hogy a teljes dokumentumot ellenőrizze a konzulens, mert előfordult, hogy félreérgettük egymást, és ez a konzultáció derült ki, utólag pedig nem lehet javítani a pontszámon. Esetleg lehetne egy pontozási útmutató minden leadáshoz, hogy a csapatok tudják, hogy mindenkel megfelelően elkészültek-e a leadáshoz, mert voltak olyan részfeladatok, amelyek nem voltak teljesen egyértelműek. Esetleg lehetne a félreérthetőbb részfeladatokhoz egy-két példa a mintában, hogy kiderüljön, pontosan mi tartozik oda.

**14.2.6 Milyen feladatot ajánlanának a projektre?**

A társasjáték jó volt szerintünk, esetleg lehetne valami olyan játék, amelyhez kell szálkezelés is, pl. idle játék (bár ehhez bonyolultabbak lennének a szekvenciadiagramok, de érdekes lehetne).

**14.2.7 Egyéb kritika és javaslat**

Nagyon hasznos a tárgy, fontos tapasztalatokat szereztünk, egyéb kritikánk/javaslatunk most nincs.