

Cereal Killers

Icefield

Dokumentáció

Konzulens:
Vörös András

Csapattagok

Filep Leonárdó Márk	F6JQ0X	filep.leonardo@gmail.com
Kálmán Bendegúz Bence	PTW6BD	kalmanbendeguz@gmail.com
Jakabos Csenge Apolka	KO6MG0	jakabos11@gmail.com
Puskás Levente	SN8YG9	puskas.levente99@gmail.com
Kuzsner Sára	HVE3FV	kuzsnersara@gmail.com

Tartalomjegyzék

Követelmény, projekt, funkcionalitás	3.
Analízis modell kidolgozása 1.	22.
Analízis modell kidolgozása 2.	43.
Szkeleton tervezése	63.
Szkeleton beadása	91.
Prototípus koncepciója	96.
Részletes tervezek	122.
Prototípus beadása	152.
Grafikus felület specifikációja	162.
Grafikus változat beadása	177.
Összefoglalás	185.

(A pdf oldalszámai)

02-Követelmény, projekt, funkcionálitás

47 – cereal_killers

Konzulens:
Micskei Zoltán
Vörös András

Csapattagok

Filep Leonárdó Márk	F6JQ0X	filep.leonardo@gmail.com
Kálmán Bendegúz Bence	PTW6BD	kalmanbendeguz@gmail.com
Jakabos Csenge Apolka	KO6MG0	jakabos11@gmail.com
Puskás Levente	SN8YG9	puskas.levente99@gmail.com
Kuzsner Sára	HVE3FV	kuzsnersara@gmail.com

2020.02.22.

2. Követelmény, projekt, funkcionalitás

2.1 Bevezetés

2.1.1 Cél

A dokumentum célja, hogy áttekintést nyújtson a cereal_killers csapat "Jégmező" című projektmunkájához a Szoftver Projekt Laboratórium tárgy keretében. Emellett meghatározza az elkészítendő munka kritériumait, és a megoldás tervét.

2.1.2 Szakterület

A szoftver egy stratégiai, gondolkodtatós, körökre osztott játék, amit bárki használhat, aki egy kis szórakozásra vágyik, vagy éppen a gondolkodási készségeit akarja fejleszteni.

2.1.3 Definíciók, rövidítések

MVC = Model, View and Controller

TDA = Tell, Don't Ask

2.1.4 Hivatkozások

Szoftvertechnológia előadás diák

Konzultáció elhangzottak (jegyzet)

Pacman minta házi feladat

(https://www.iit.bme.hu/system/files/uploads/module_files/StHf_Pacman.pdf)

2.1.5 Összefoglalás

A következőkben részletesen bemutatjuk a készülő szoftver jellemzőit felhasználói és fejlesztői szempontból is, utóbbi szempontot részletesebben kidolgozva.

2.2 Áttekintés

2.2.1 Általános áttekintés

A program alapvetően a MVC szerkezeti mintára épül. Kezdetben a menüből a játékosnak van lehetősége új játékot kezdeni. Ekkor megjelenik egy új ablak és inicializálónak a játékot felépítő objektumok. Ezek a pálya, az azt felépítő mezők, a szereplők, és az egyes tárgyak, élelmek. Ami logikailag a valóságban összetartozik, az az objektumok szintjén is meg fog ugyanúgy jelenni, például a pálya objektum tartalmazza a mezőket, vagy a szereplőket reprezentáló objektumok és a birtokolt tárgyaik között asszociációs kapcsolat van. A felhasználói interakciókra a listener-ek figyelnak, amik az egér és/vagy billentyűzetről jövő inputok hatására meghívják az adott cselekvést biztosító függvényeket, például a mozgást, vagy tárgyak felvezését. Ezek a függvények az adott szituációban az objektumokon a megfelelő változtatást fogják végrehajtani a TDA elvet követve. minden objektummal az interfészén keresztül lehet kommunikálni, például egy jégtáblára rálépve annak egy függvényét hívjuk, hogy rá szeretnénk lépni, az pedig a belső állapota szerint viselkedik. Fontos a jogosultságok figyelése, hogy melyik szereplő léphet éppen, ezért szükség lesz egy, a játék menetét kezelő osztályt is létrehozni. Ez felelhet a hóviharokért is, amelyek véletlenszerű időpontokban véletlenszerű jégtáblákat érintenek. A program egy dicsőséglistát

is nyilván fog tartani, ehhez az adatfájlt a háttértáron a programmal egy környezetben kell tartani.

2.2.2 Funkciók

A feladat egy játékprogram, amelyben egy stratégiai, körökre osztott játéket lehet játszani. A tábla egy jéggel borított sziget, tengerrel körülvéve. A sziget jeǵtáblából áll, minden jeǵtábla négyzet alakú, és négyzetrács alakban helyezkednek el. A játéket 3 vagy több szereplő játszhatja. minden szereplő eszkimó vagy sarkutató, különböző képességekkel. A szereplök körökre osztva tevékenykednek, és egymással vannak egy csapatban, a számítógép ellen. A játék célja a túlélés, azaz egy jelzörakéta használatával segítséget kérni.

A jégmező jeǵtáblái többfélék lehetnek, de ez látszatra nem mindig egyértelmű. Vannak stabil jeǵtáblák, amelyek tetszőleges számú szereplőt elbírnak, tehát bárhányan ráállhatnak következmények nélkül. Ezzel ellentétesen vannak instabil jeǵtáblák, amelyek nem bírnak el akárhány szereplőt, de hogy hányat bírnak el, az véletlenszerű. Ha egy ilyen táblán túl sokan állnak, az adott jeǵtábla átfordul, és a rajtuk állók a vízbe esnek. Azok a szereplök, akik vízbe esnek, akár meg is halhatnak, de egyes esetekben van mód a megmenekülésre. A jeǵtáblákat a játék kezdetén eltérő mennyiségű hó borítja, ami nehezíti a tárgyak megtalálását, ráadásul hóviharok hatására még több hó jelenhet meg.

Minden jeǵtábla egy-egy titkot rejteget, ezért óvatosan kell játszani. Ezek a titkok lehetnek számunkra szerencsések, például befagyott tárgyak és élelem. A befagyott tárgyak nagyon fontosak a játékmenet szempontjából, mert szinte elengedhetetlenek a győzelemhez. Ezek a tárgyak lehetnek lapátok, kötelek, bűvárruhák, jelzörakéta-alkatrészek és persze élelem. Csak akkor lehet meglátni és felvenni egy ilyen tárgyat, ha a jeǵtábla tiszta, azaz nem borítja hó. Ellentétes esetben a jeǵtáblán szabadon mozoghatunk, de az értékes tárgyról valószínűleg fogalmunk sincs, kivéve, ha emlékszünk rá, hogy ott volt, de időközben egy hóvihar betakarta hóval. A titkok számunkra kellemetlenek is lehetnek, azaz hóval fedett lukak is lehetnek rajtuk. Ezekbe beleesve csak azon szereplök élhetnek túl, akiken bűvárruha van, és azok akik olyan szerencsések, hogy a mellettük lévő mezőn van egy szereplő, akinél van kötél. Ebben az esetben azonnal kimenekíti, és a vízből a társával ugyanarra a jeǵtáblára kerül.

Minden körben minden szereplőnek korlátozva van, hogy mennyit tevékenykedhet. Ez a játékban úgy valósul meg, hogy mindenki minden körre 4 egység munkája van. A jeǵtáblát borító egy egységnyi hó eltakarítása, egy szomszédos jeǵtáblára való átlépés, vagy egy kiásott tárgy felvétele vagy letevése mind egy-egy egység munkát vesznek igénybe. Az éleelmet megenni is egy egység munka. Ha egy szereplónél van lapát, akkor ő két egységnyi havat tud egy egységnyi munkával eltakarítani. Ha elfogy az adott körre rendelkezésre álló munkamennyiség, akkor a játékos cselekvőképtelen lesz. Nem kötelező minden egység munkát felhasználni, de a fel nem használt munka nem vihető tovább a következő körre. minden szereplónél minden fajta tárgyból egy-egy lehet.

A játékbeli jégmezőn, ahogyan az lenni szokott, időnként feltámadnak hóviharok. Ezek hatására véletlenszerűen néhány jeǵtáblát újonnan hullott, friss hó borít be, pontosan egy egységnyi. A játékban látszódni fog, hogy az adott hóvihar mely mezőket érintette. Ha egy szereplő egy ilyen hóviharral sújtott jeǵtáblán állt, akkor nem borítja be a hó, de a testhője egy egységgel csökken. A testhő jelenti a játékban az életpontokat. Az eszkimóknak a játék kezdetén 5 egység, a sarkutatóknak viszont csak 4 egység testhőjük van. A megtalált és elfogyasztott élelem egy egységgel növeli a testhőt.

A szereplök minden körben mozoghatnak, egyik jeǵtábláról a másikra. Mindig csak a szomszédos jeǵtáblákra lehet átkelni. Szomszédos jeǵtáblának csak az alsó, felső és oldalsó mezők számítanak, az átlósan elhelyezkedők nem. A sarkutatók képesek arra, hogy

megnézzék, hogy a velük szomszédos jégtábla hány embert bír el, de ez csak az ő körükben látszódik. A lyukas jégtábla egy embert sem bír el, ebben az esetben ez a szám 0. Az eszkimók tudnak iglut építeni, ezek a helyükön maradnak, és bennük átvészelhetőek a hóviharok. Egy igluban bármennyien elférnek. Az olyan mező, amelyre iglu épült, mindenképpen stabil lesz.

A játék célja, hogy a szereplők megtalálják egy jelzőrakéta alkatrészeit. Ezek a pisztoly, a jelzőfény, és a patron. Ezek a tárgyak ugyanúgy jégbe vannak fagyva, mint más tárgyak. Azonban nem elég csak megtalálni ezeket, hanem ugyanarra a jégtáblára kell vinniük az összeset, és mindenkiuknak oda kell mennie arra a jégtáblára. Ekkor egy-egy munka felhasználásával összeszerelhetik, majd elsüthetik. Ekkor megnyerik a játékot. Ha valaki még ezelőtt meghal, akkor a játék véget ér, ekkor már nincs esély a győzelemre. Meghalni úgy lehet, hogy egy szereplő a vízbe esve megfullad, vagy elfogy a testhője, és kihűl.

2.2.3 Felhasználók

A játékot játszó felhasználónak az alapműveltségen kívül nem szükséges semmilyen előképzettséggel rendelkeznie. Előny a jó helyzetfelismerés és memória.

2.2.4 Korlátozások

A programnak stabilan kell futnia minden és a játék szabályaihoz megfelelően helyesen működnie: nem állhat le a felhasználó beavatkozása nélkül, nem fagyhat ki. A követelményeknek eleget kell tennie.

2.2.5 Feltételezések, kapcsolatok

A Szoftvertechnológia című tárgy diáit, példa házi feladatát (pacman: https://www.iit.bme.hu/system/files/uploads/module_files/StHf_Pacman.pdf) felhasználtuk egyes kérdések tisztázásához, a program megtervezéséhez.

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Use-case	Komment
R1	A játéktér egy jéggel borított sziget, tengerrel körülvéve.	bemutatás	alapvető	feladatkiírás	Move	
R2	A sziget jégtáblákból áll, minden jégtábla négyzet alakú, és négyzetrács alakban helyezkednek el.	bemutatás	alapvető	feladatkiírás, csapat	Move	
R3	A játékot 3 vagy több szereplő játszhatja.	bemutatás, kiértékelés	alapvető	feladatkiírás	Move	

R4	Minden szereplő eszkimó vagy sarkkutató, különböző képességekkel.	bemutatás, kiértékelés	alapvető	feladatkiírás	Move	
R5	A szereplők körökre osztva tevékenykednek, és egymással vannak egy csapatban, a számítógép ellen.	bemutatás	alapvető	feladatkiírás, csapat	Move	
R6	Vannak stabil jégtáblák, és instabil jégtáblák, amik nem bírnak el akárhány embert.	bemutatás, kiértékelés	alapvető	feladatkiírás	Move, Inspect ice-float	
R7	Ha egy instabil táblán túl sokan állnak, az adott jégtábla átfordul, és a rajtuk állók a vízbe esnek.	bemutatás, kiértékelés	alapvető	feladatkiírás	Move, Inspect ice-float	
R8	A jégtáblába fagyott tárgyat csak akkor lehet meglátni és felvenni, ha azt nem borítja hó.	bemutatás, kiértékelés	alapvető	feladatkiírás	Pick something up	
R9	Jégbe fagyott tárgy lehet élelem, lapát, kötél, búvárruha, vagy jelzőrakéta alkatrész.	bemutatás	alapvető	feladatkiírás	Pick something up	
R10	Lehet olyan jégtábla, amin hóval fedett lyuk van.	bemutatás	alapvető	feladatkiírás	Inspect ice-float	

R11	A lyukba eső szereplő csak akkor élhet túl, ha búvárruha van rajta, vagy egy szomszédos mezőn lévő szereplő rendelkezik kötéllel.	kiértékelés	alapvető	feladatkiírás	Wear diving suit, Save player	
R12	Ha elfogy az adott köorre rendelkezésre álló munkamennyisé g, akkor a játékos cselekvőképte le lesz.	bemutatás, kiértékelés	alapvető	feladatkiírás , csapat	Move	
R13	A játékbeli jégmezőn időnként feltámadnak hóviharok, melyek hatására egyes mezőkön egy-egy egységnnyi hó jelenik meg.	bemutatás, kiértékelés	alapvető	feladatkiírás	Start snowstorm	
R14	Ha egy szereplő egy ilyen hóviharral sújtott jégtáblán állt, akkor nem borítja be a hó, de a testhője egy egységgel csökken.	kiértékelés	alapvető	feladatkiírás , csapat	Start snowstorm	
R15	A megtalált és elfogyasztott élelem egy egységgel növeli a testhőt.	kiértékelés	alapvető	feladatkiírás	Eat	
R16	A szereplők minden körben mozoghatnak.	kiértékelés	alapvető	feladatkiírás	Move	

R17	Egy mozgás egy egységnyi munka.	kiértékelés	alapvető	feladatkiírás	Move	
R18	Mindig csak a szomszédos jégtáblákra lehet átkelni.	bemutatás, kiértékelés	alapvető	feladatkiírás, csapat	Move	
R19	A sarkkutatók képesek arra, hogy megnézzék, hogy a velük szomszédos jégtábla hány embert bír el.	bemutatás, kiértékelés	alapvető	feladatkiírás	Inspec-ice -float	
R20	Az eszkimók tudnak igtut építeni, ezek a helyükön maradnak, és bennük átvészhetőek a hóviharok.	kiértékelés	alapvető	feladatkiírás	Build igloo	
R21	A játék célja, hogy a szereplők megtalálják egy jelzőrakéta alkatrészeit.	bemutatás	alapvető	feladatkiírás	Assemble and fire pistol	
R22	A jelzőrakéta alkatrészei a pisztoly, a jelzőfény, és a patron.	kiértékelés	alapvető	feladatkiírás	Pick something up	
R23	A győzelemhez a szereplőknek el kell össze kell szerelniük és el kell sütniük a rakétát.	kiértékelés	alapvető	feladatkiírás	Assemble and fire pistol	

R24	A rakéta csak akkor szerelhető össze és süthető el, ha az összes alkatrész és szereplő egyazon jégtáblán vannak.	kiértékelés	alapvető	feladatkiírás	Drop something	
R25	Ha valaki még a gyözelem előtt meghal, akkor a játék véget ér.	bemutatás, kiértékelés	alapvető	feladatkiírás	Move	
R26	Ha egy szereplőnek elfogy a testhője, meghal.	kiértékelés	alapvető	feladatkiírás	Move	
R27	Ha egy szereplő vízbe esik és senki nem menti ki, meghal.	kiértékelés	alapvető	feladatkiírás	Save player	
R28	A jégtáblákat a játék kezdetén eltérő mennyiségű hó borítja.	bemutatás, kiértékelés	fontos	feladatkiírás	Start snowstorm	
R29	A kötéllel kimentett szereplő a segítő társa jégtáblájára kerül.	bemutatás	fontos	csapat	Save player	
R30	Mindenkinet minden körre 4 egység felhasználható munkája van.	kiértékelés	fontos	feladatkiírás, csapat	Move	

R31	Egy élelmet megenni, egy jégtáblát borító egy egységnyi hó eltakarítása, egy szomszédos jégtáblára való átlépés, vagy egy kiásott tárgy felvétele vagy letevése mind egy-egy egység munkát vesznek igénybe.	kiértékelés	fontos	feladatkiírás , csapat	Eat, Remove snow, Move, Pick something up, Drop something	
R32	Ha egy szereplőnél van lapát, akkor ő két egységnyi havat tud egy egységnyi munkával eltakarítani.	kiértékelés	fontos	feladatkiírás	Pick something up, Remove snow	
R33	Az eszkimóknak a játék kezdetén 5 egység, a sarkkutatóknak viszont csak 4 egység testhőjük van.	bemutatás, kiértékelés	fontos	feladatkiírás	Move	
R34	Szomszédos jégtáblának csak az alsó, felső és oldalsó mezők számítanak, az átlósan elhelyezkedők nem.	bemutatás	fontos	csapat	Move	
R35	Egy igluban bármennyien elférnek.	kiértékelés	fontos	csapat	Build igloo, Move	

R36	Az olyan mező, amelyre iglu épült, mindenkorban stabil lesz.	kiértékelés	fontos	csapat	Build igloo	
R37	A rakétát a szereplők egy-egy munka felhasználásával szerelhetik össze, és süthetik el.	bemutatás, kiértékelés	fontos	feladatkiírás, csapat	Assemble and fire pistol	
R38	Nem kötelező minden egység munkát felhasználni, de a fel nem használt munka nem vihető tovább a következő körre.	kiértékelés	opcionális	csapat	Move	
R39	Minden szereplónél minden fajta tárgyból egy-egy lehet.	bemutatás, kiértékelés	opcionális	csapat	Pick something up, drop something	

2.3.2 Erőforrásokkal kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
R40	A program Java nyelven készül.	bemutatás	alapvető	feladatkiadó	
R41	Az elkészült programkódnak fordíthatónak kell lenni Java SE Developer Kit 8-on.	bemutatás, kiértékelés	fontos	feladatkiadó	

R42	A termékhez ajánlott operációs rendszer a Windows 10 és a Linuxon belül az Ubuntu disztribúció.	bemutatás	opcionális	csapat	A felsorolt két operációs rendszerre tudjuk biztosítani a helyes működést.
R43	A játék fő bevitteli eszközei az egér és billentyűzet.	bemutatás	fontos	csapat	

2.3.3 Átadással kapcsolatos követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
R44	A termék futtatásához szükséges a Java Runtime Environment 8 megléte.	bemutatás	alapvető	feladat	
R45	Minden követelménynek teljesülnie kell.	bemutatás, kiértékelés	alapvető	feladat	

2.3.4 Egyéb nem funkcionális követelmények

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
R46	A felhasználónak előnyös az alapfokú angol tudás	bemutatás	opcionális	csapat	

2.4 Lényeges use-case-ek

2.4.1 Use-case leírások

Use-case neve	Move
Rövid leírás	A játékos mozog.
Aktorok	Player
Forgatókönyv	1. A játékos egyik jégtábláról a másikra lép (1 munkaegység).
Alternatív forgatókönyv	1.a. A jégtáblán nincs hely, ezért a játékos a helyén marad.
Alternatív forgatókönyv	1.b. A jégtábla átfordul és a rajta tartózkodó szereplő(k) a vízbe esik (esnek).

Use-case neve	Pick something up
Rövid leírás	A játékos felvesz egy tárgyat (lapát, kötél, bűvárruha, élelem, jelzőrakéta alkatrészei)
Aktorok	Player
Forgatókönyv	1. A játékos felveszi a tárgyat 1 munkaegység árán.
Alternatív forgatókönyv	1.a. A játékosnak nem áll elég hely rendelkezésére ezért nem tudja felvenni a tárgyat.

Use-case neve	Drop something
Rövid leírás	A játékos letesz egy tárgyat.
Aktorok	Player
Forgatókönyv	1. A játékos leteszi a tárgyat 1 munkaegység árán.
Alternatív forgatókönyv	1.a. Nincs szabad hely a jégtáblán a tárgy számára.

Use-case neve	Remove snow
Rövid leírás	A játékos eltakarítja a havat a jégtábláról.
Aktorok	Player
Forgatókönyv	1. A játékos eltakarít 1 egységnnyi havat 1 egységnnyi munkaráfordítással.
Alternatív forgatókönyv	1.a. A játékosnak van lapátja, így 2 egységnnyi havat tud eltakarítani 1 egységnnyi munkaráfordítással.

Use-case neve	Save player
Rövid leírás	A játékos megmenthet egy másik, vízbe esett szereplőt ha van kötele.
Aktorok	Player
Forgatókönyv	1. A játékos kimenekíti a vízbe esett szereplőt a kötéllel, 1 egységnnyi munkaráfordítással.

Use-case neve	Remove diving suit
Rövid leírás	A játékos levetheti a bűvárruhát.
Aktorok	Player
Forgatókönyv	1. A játékos leveti a bűvárruhát.

Use-case neve	Wear diving suit
Rövid leírás	A játékos felveheti a bűvárruhát, ami megóvja a haláltól ha vízbe esik.
Aktorok	Player
Forgatókönyv	1. A játékos felveszi a bűvárruhát.

Use-case neve	Eat
Rövid leírás	A szereplő megehet 1 egységnnyi munka árán 1 élelmet, így a testhője 1-el növekszik, vagy nem változik ha sarkkutatók esetén 4, eszkimók esetén 5 marad.
Aktorok	Player

Forgatókönyv	1. A szereplő megeszi az élelmet és testhője 1-el növekszik.
Alternatív forgatókönyv	1.a. A szereplő testhője változatlan marad.

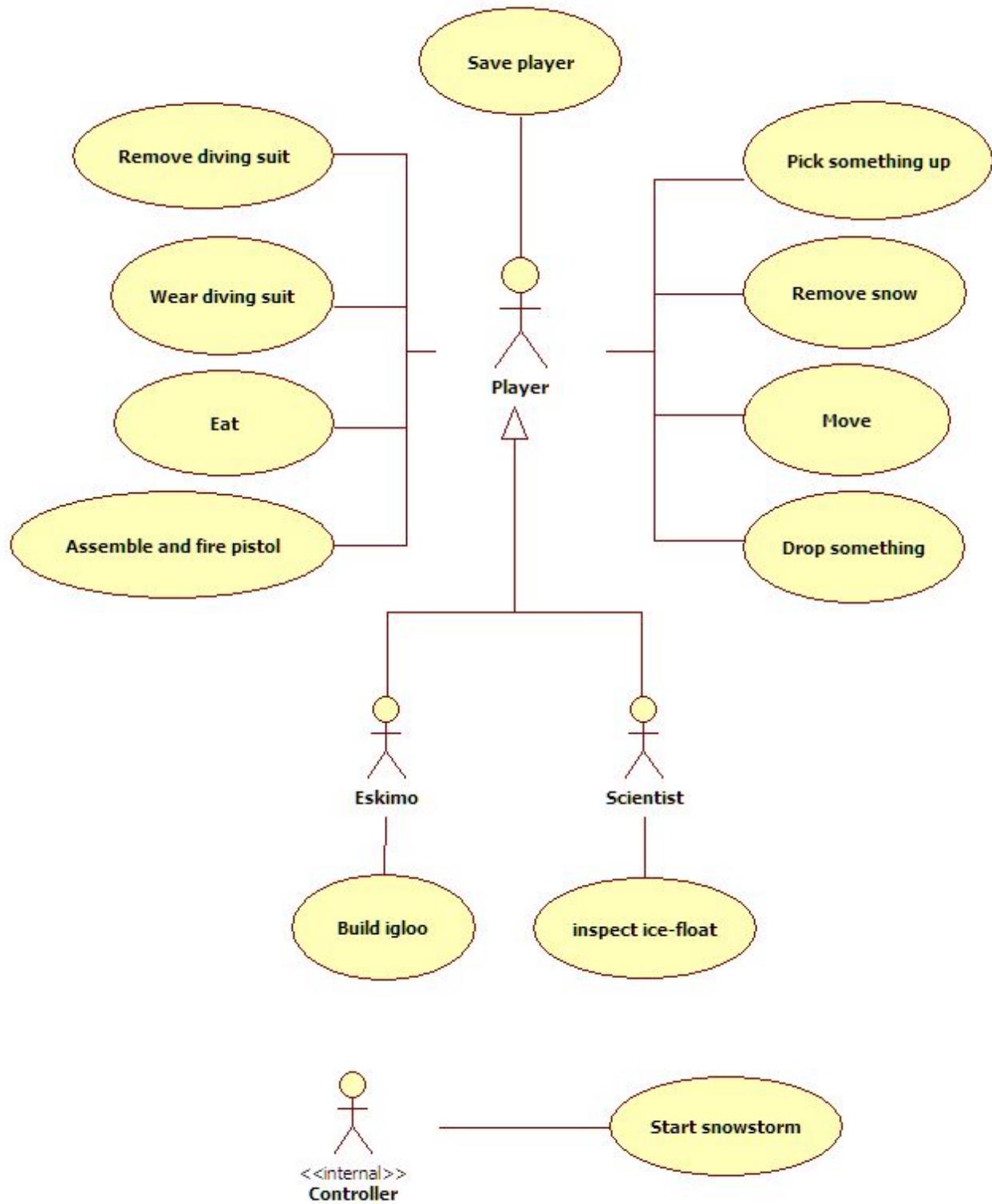
Use-case neve	Assemble and fire pistol
Rövid leírás	Az alkatrészeket a csapat összegyűjti és ugyanarra a jégtáblára viszi, akkor egy munka felhasználásával összeszerelhetik, majd a csapat leadja a jelzőlövést egy egységes munka árán és megnyeri a játékot.
Aktorok	Player
Forgatókönyv	1. A szereplők összeszerelik a rakétát és leadják a jelzőlövést, a játék véget ér.

Use-case neve	Build igloo
Rövid leírás	Az eszkimó tud iglut építeni egy egységes munka felhasználásával, amiben átvészhetők a hóviharok.
Aktorok	Eskimo
Forgatókönyv	1. Az eszkimó felépíti az iglut és az igluba kerül.

Use-case neve	Inspect ice-float
Rövid leírás	A sarkkutató meg tudja nézni, hogy az a jégtábla, amire lépne, hány embert bír el egy egységes munka árán.
Aktorok	Scientist
Forgatókönyv	1. A sarkkutató megvizsgálja a jégtáblát.

Use-case neve	Start snowstorm
Rövid leírás	A jégmezőn időnként feltámad a hóvihar, és néhány érintett jégtáblát újabb adag friss hóval borít be. Akit elkap, annak a testhője egységesivel csökken.
Aktorok	Controller
Forgatókönyv	1. A vihar hóval borítja az érintett jégtáblákat. 2. A vihar miatt csökken egy egységesivel az érintettek testhője.

2.4.2 Use-case diagram



2.5 Szótár

Ásás	A folyamat amikor a karakter eltávolít egy höréteget egy mezőről.
Búvárruha	A hideg sarki víz túlélésére alkalmas eszköz (nélküle vízbe esés esetén egy társ által történő azonnali mentés hiányában meghal a vízbe zuhant karakter).
Energia	A játékban a munkaegység neve, amiből minden körben minden szereplőnek 4 van.
Élelem	Használata a testhőmérsékletet növeli eggyel.
Építés	Az a folyamat, amikor egy mezőre igtut helyeznek.
Hóvihar	Amikor a pálya höréteget helyez el adott mezőkön, illetve csökkenti a játékosok testhőjét.
Iglu	Az eszkimók által a hóvihar kivédésére épített eszköz.
Instabil jégtábla	Olyan jégtábla, amin csak adott számú karakter állhat.
Játékos	A számítógép előtt ülő személy.
Jégtábla	Egységes felület ahova a karakter léphet.
Jelzőfény	A pályán található elem, a jelzőrakéta része.
Jelzőrakéta	Az az eszköz, aminek elsütése a játék végét okozza (összetevői: pisztoly, jelzőfény, patron).
Kör	Egy olyan egység amin belül minden karakter tevékenykedik, amíg az energiájuk el nem fog..
Kötél	Mások vízből való kihúzására alkalmas eszköz.
Lapát	A hó gyorsabb eltakarítására alkalmas eszköz.

Lépés	A körnek az a része amikor a karakter cselekedhet. Maximum négy egységnyi munka elvégzése.
Luk	Olyan instabil jégtábla amire egy karakter sem tud ráállni.
Mentés	Amikor egy karakter kötél segítségével kihúzza a vízbe esett társát.
Pálya	A jégtáblák összessége ahol a történések zajlanak.
Patron	A pályán található elem, a jelzőrakéta része.
Pisztoly	A pályán található elem, a jelzőrakéta része.
Stabil jégtábla	Olyan jégtábla amin akármennyi karakter állhat.
Szereplő	A játék egy karaktere, lehet eszkimó vagy kutató.
Tenger	A mezők között és körül lévő terület.
Testhő	A karakterek egészségét jelző mennyiség.

2.6 Projekt terv

A projekt végrehajtásának lépései:

Határidő	Feladat	Felelős (beosztás)
2020.02.24.	Követelmény, projekt, funkcionalitás - beadás	Kálmán (csapatvezető)
2020.03.02.	Analízis modell kidolgozása 1. - beadás	Kálmán (csapatvezető)
2020.03.09.	Analízis modell kidolgozása 2. - beadás	Kálmán (csapatvezető)
2020.03.16.	Szkeleton tervezése - beadás	Jakabos(fejlesztő,designer)
2020.03.23.	Szkeleton - beadás és a forráskód herculesre való feltöltése	Jakabos(fejlesztő, designer)
2020.03.30.	Prototípus koncepciója - beadás	Kuzsner (projektmenedzser)
2020.04.06.	Részletes tervek - beadás	Jakabos(fejlesztő, designer)

2020.04.20.	Prototípus készítése, tesztelése	Puskás (fejlesztő)
2020.04.27.	Prototípus - beadás és a forráskód, a tesztbemenetek és az elvárt kimenetek herculesre való feltöltése	Puskás (fejlesztő)
2020.05.04.	Grafikus felület specifikációja - beadás	Filep (ütemező, fejlesztő)
2020.05.11.	Grafikus változat készítése	Filep (ütemező, fejlesztő)
2020.05.18.	Grafikus változat és Összefoglalás - beadás és a forráskód herculesre való feltöltése	Kuzsner (projektmenedzser)

Dokumentálásra használt eszközök: Google Docs

Kommunikáció: Személyesen, Slack

Modellező eszköz: WhiteStarUML

Fejlesztőkörnyezet: IntelliJ

Dokumentumok megosztása: Google Drive

Forráskód megosztása, verziókezelése: Github

2.7 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.02.12. 18:00	1 óra 30 perc	Jakabos Kuzsner Puskás Filep Kálmán	Értekezlet. Értekezés a csapat megalakulásáról, névválasztás, "stratégia megtervezése". Döntés: a fő kommunikációs platform a Slack, a dokumentumokat Google Drive-n, a kódot githubon tartjuk karban. Kuzsner majd létrehozza a Slack organizationt, Kálmán a Drive mappát, Puskás a Github felületet hozza létre.
2020-02-19 13:00	20 perc	Kuzsner Puskás Filep Kálmán	Értekezlet. Értekezés a feladat megoldásának módjáról, a munka elosztásáról. Döntés: Kuzsner vesz laptartót.
2020-02-20 13:00	30 perc	Kuzsner	Jegyzetek feltöltése, ellenőrzése, napló dokumentum alakítása, Slack beállítása.
2020-02-20 18:00	20 perc	Kálmán	Cél és szakterület definiálása.

2020-02-20 19:00	30 perc	Jakabos	Feladat elolvasása és értelmezése, UML diagram vázlat (Use-Case).
2020-02-21 12:00	1 óra	Kuzsner Puskás Filep Kálmán	Értekezlet. Feladat értelmezése, értelmezésbeli különbségek, játékmenet és funkciók átgondolása. Döntés: Kálmán elkészíti a funkciók leírását, Jakabos a Use-Case leírásokat.
2020-02-21 14:00	1 óra	Jakabos Kálmán Puskás	Értekezlet. Dokumentum írása, hétvégi feladatok megbeszélése. Döntés: Jakabos csinálja a Use-Case leírásokat és diagramokat, Puskás az erőforrásokkal és átadással kapcsolatos követelményeket, Kálmán pedig a funkciókat és a funkcionális követelményeket. Megbeszélik, hogy az egyéb feladatokat, például a szótár és definíciók készítését a jelenlegi munka után végzik el.
2020-02-21 17:00	2 óra	Kálmán	Általános áttekintés és funkciók megírása.
2020-02-22 15:00	1 óra 20 perc	Jakabos	Use-Case leírások első verziójának elkészítése.
2020-02-22 18:00	1 óra 30 perc	Kálmán	Funkcionális követelmények leírása.
2020-02-22 18:45	30 perc	Puskás	Erőforrásokkal és átadással kapcsolatos követelmények, erőforrások megírása.
2020-02-22 22:00	1 óra 30 perc	Filep	Use-Case összekötése a funkcionális követelményekkel.
2020-02-22 22:00	1 óra 30 perc	Jakabos	Use-Case leírások második verziójának elkészítése.
2020-02-23 1:00	15 perc	Kálmán	Projekt terv vázának elkészítése.
2020-02-23 16:00	20 perc	Puskás	Definíciók, rövidítések, hivatkozások, összefoglalás elkészítése.
2020-02-23 19:00	1 óra	Jakabos Kálmán Filep Kuzsner Puskás	Értekezés: Elkészült feladatok ellenőrzése, Visszalévő feladatok elosztása. Döntés: határidő definiálása: csütörtök 10 óra.
2020-02-23 20:00	15 perc	Jakabos	Use-Casek javítása.
2020-02-23 20:00	10 perc	Kálmán	Projekt terv befejezése.

2020-02-23 20:00	1 óra	Jakabos	Hivatkozások, feltételezések, kapcsolatok leírása, fedlap elkészítése.
2020-02-23 21:00	4 óra	Kuzsner	Egyéb nem funkcionális követelmény megírása, szótár elkészítése.

Analízis modell kidolgozása 1.

47 – cereal_killers

Konzulens:

**Micskei Zoltán
Vörös András**

Csapattagok

Filep Leonárdó Márk	F6JQ0X	filep.leonardo@gmail.com
Kálmán Bendegúz	PTW6BD	kalmanbendeguz@gmail.com
Bence		
Jakabos Csenge Apolka	KO6MG	jakabos11@gmail.com
	0	
Puskás Levente	SN8YG9	puskas.levente99@gmail.co m
Kuzsner Sára	HVE3FV	kuzsnersara@gmail.com

2020.03.01.

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Eszkimó

A két választható szereplőtípus közül az egyik. Energia- és életpontokkal rendelkezik minden szereplő. minden kör elején 4 energiapontja van, életpontja az eszkimónak 5 van. Ez a szereplő képes tárgyakat felvenni/lerakni, ezeket használni, lépni. Továbbiakban még felelőssége, hogy ha elfogynak az életpontjai, akkor meghaljon és a körvégét jelezze, ha nincs több energiapontja vagy ha nem kíván több munkát végezni. Az eszkimó egy különleges képességgel is rendelkezik, igyekezni tud építeni egy jégtáblára, ami a hóvihar ellen véd.

3.1.2 Sarkkutató

Ez a másik fajta szereplőtípus. Mindazon attribútumokkal és képességekkel rendelkezik, mint a eszkimó, leszámítva az eszkimó különleges képességét. Neki is megvan a számára egyedi képesség, hogy megtudja nézni mi van egy szomszédos jégtáblán. Továbbá még abban különbözik az eszkimótól, hogy 4 életponttal rendelkezik.

3.1.3 Jégtábla

Ezeknek az összessége alkotja a pályát. A jégtábla az az egységes terület, amin a szereplők és a tárgyak tartózkodhatnak. Felelőssége még az is, hogy ismerje a mellette lévő szomszédos jégtáblákat, és biztosítsa a szereplők ezekre való átlépését.

3.1.4 Kontroller

Ez irányítja a játék menetét. Feladatai közé tartozik, hogy elindítja a játékmenetet, kezeli a hóviharokat, befejezi a játékot, ha nyert a csapat vagy ha meghalt egy szereplő.

3.1.5 Lapát

Segítségével a szereplő havat tud ellapárolni egy jégtábláról. Jégbe fagyva található meg, mint ahogy a többi tárgy is.

3.1.6 Búvárruha

Használata életet menthet olyan esetben, ha egy szereplő belesik a vízbe, miközben ez rajta van. A búvárruha nélkül meghal a vízbe esett szereplő.

3.1.7 Kötél

Abban segít a kötél, hogyha egy szereplő rendelkezik vele és egy szomszédos mezőn egy szereplő belesett a vízbe, akkor segítségével kihúzhatja a vízből.

3.1.8 Élelem

Ha egy játékos elhasználja, akkor növeli az életpontjait 1-el. Hasonlóan a többi tárgyhoz, ez is a pályán véletlenszerűen helyezkedik el.

3.1.9 Jelzőrakéta

A játék célja, hogy a csapat a rakéta alkatrészeit megtalálja és összeszerelje, majd közösen elszüssék. Három ilyen alkatrészre van szükség: pisztoly, jelzőfény, patron. Felelőssége, hogy össze lehessen szerelni és elszüntetni, és tudja magáról, hogy egy jégtáblán van.

3.1.10 Pisztoly

A jelzőrakéta egyik alkatrésze, elengedhetetlen kellék a rakéta összeszereléséhez. A pályán ugyanúgy megtalálható, mint a többi tárgy.

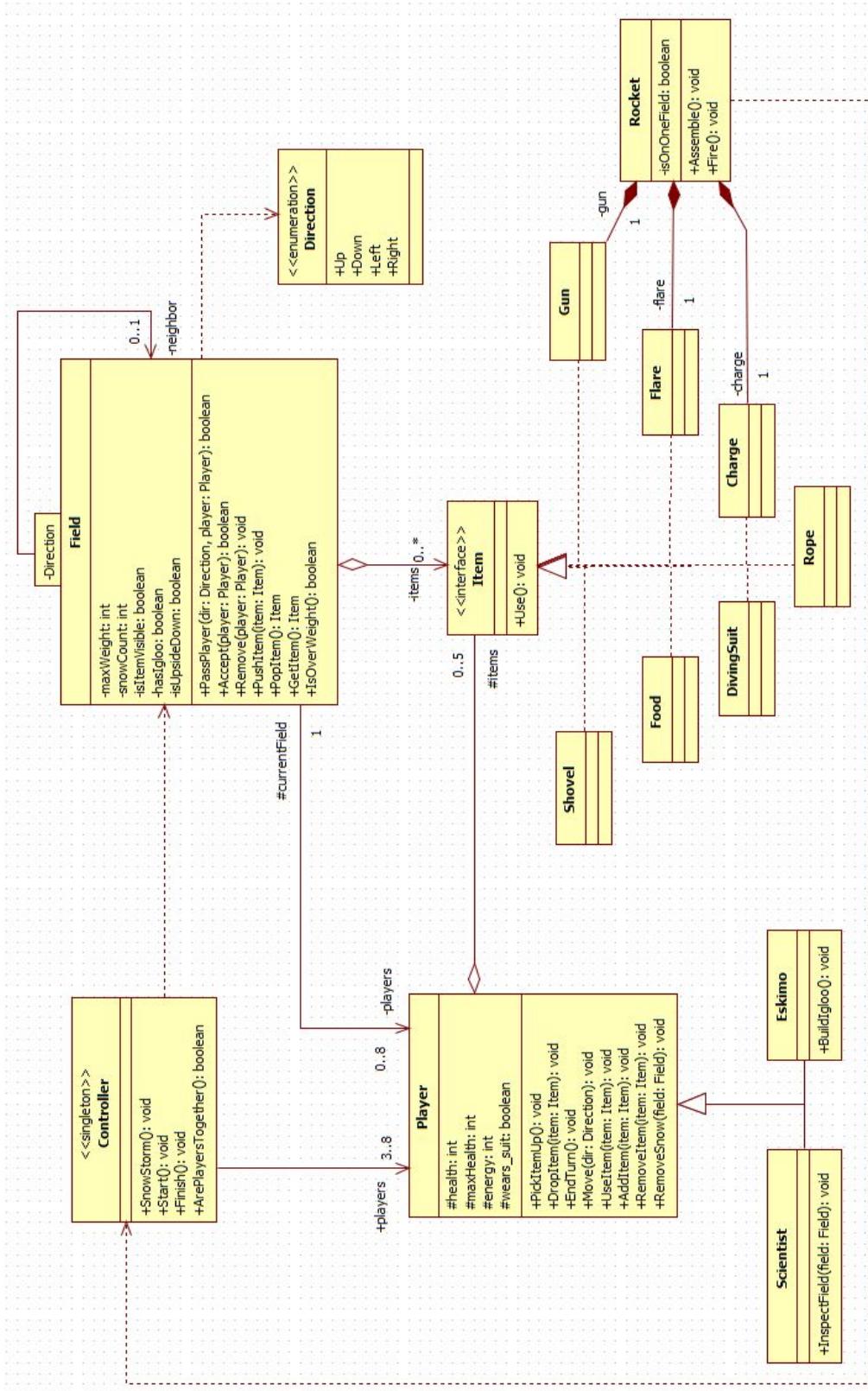
3.1.11 Jelzőfény

A jelzőrakéta egy másik alkatrésze, erre is szükség van az összeszereléséhez.

3.1.12 Patron

A jelzőrakéta harmadik és utolsó alkatrésze, ugyanaz jellemző rá, mint a többi alkatrészre.

3.2 Statikus struktúra diagramok



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

3.3 Osztályok leírása

3.3.1 Charge

- **Felelősség**

A Charge osztály a jelzőrakéta egy alkatrészét reprezentálja, a patron. A rakéta összeszereléséhez van hozzá szükség. A szereplők tudják az eszköztárukban hordozni.

- **Interfészek**

Item

3.3.2 Controller

- **Felelősség**

A Controller osztály a játékbeli hóviharokat kezeli, valamint ő indítja a játékot és ő is fejezi be, azaz nyeréskor vagy vesztéskor vele kell kommunikálni.

- **Metódusok**

- **void SnowStorm():** A játékbeli jégmezőn hóvihart indít, melynek hatására véletlenszerű jégmezőkön egy-egy újabb höréteg jelenik meg.
- **void Start():** Elkezdi a játékot, inicializálja az objektumokat, és odaadja az első szereplőnek a kezdés jogát.
- **void Finish():** Ennek hatására befejeződik a játék.
- **boolean ArePlayersTogether():** Azt vizsgálja ez a függvény, hogy a szereplők egy jégtáblán vannak-e, ekkor igazat ad vissza. Erre azért van szükség, mert a rakéta csak ekkor válik összeszerelhetővé.

3.3.3 Direction

- **Felelősség**

Egy felsorolt típus, amely a játékban az irányokat adja meg, ez négyféle lehet, ugyanis az átlósan lévő mezőkre nem lehet átkelni.

- **Attribútumok**

- **Up:** A felfelé mutató irány.
- **Down:** A lefelé mutató irány.
- **Left:** A balra mutató irány.
- **Right:** A jobbra mutató irány.

3.3.4 DivingSuit

- **Felelősség**

A DivingSuit osztály egy búvárruhát reprezentál. Ezek alapból a jége vannak fagyva, és ugyanúgy fel lehet őket venni, mint más eszközöket. Ha egy szereplő megszerzi, akkor onnantól kezdve ha vízbe esik, nem fog meghalni.

- **Interfészek**

Item

3.3.5 Field

- **Felelősség**

A Field osztály reprezentál egy játékbeli jégtáblát. Ilyenből áll össze a pálya, négyzetrács alakban helyezkednek el. Felelőssége, hogy kezelje a rajta lévő tárgyakat, és átforduljon, ha a teherbíró képességénél több szereplő helyezkedik el rajta. Ismeri a rajta lévő szereplőket, és mozgásnál ő rakja át a karaktereket a szomszédos jégtáblára.

- **Attribútumok**

- **-maxWeight: int:** A jégtábla maximum teherbíró képessége. Ha ennél többen állnak rajta, átfordul.
- **-snowCount: int:** Azt mondja meg, hogy a jégtáblán hány réteg hó van.
- **-isItemVisible: boolean:** Igaz, ha a jégtáblán látható a rajta lévő eszköz, azaz egy réteg hó sincs a jégtáblán. Származtatott attribútum a snowCount-ból.
- **-hasIgloo: boolean:** Igaz, ha a jégtáblán van iglu.
- **-isUpsideDown: boolean:** Igazat ad vissza, ha a jégtábla fel van fordulva.

- **Metódusok**

- **boolean PassPlayer(Direction dir, Player player):** Az argumentumban megadott szereplőt átrakja az argumentumban megadott irányban elhelyezkedő szomszédos mezőre, majd visszatér azzal, hogy ez sikerült-e (sikertelen az áthelyezés például ha abban az irányban tenger van).
- **boolean Accept(Player player):** A paraméterben megadott szereplőt rárakja az adott jégtáblára. Igazzal tér vissza, ha ez sikerült.
- **void Remove(Player player):** A paraméterben megadott szereplőt eltávolítja a jégtábláról.
- **void PushItem(Item item):** A paraméterben megadott eszközt leteszi a mezőre.
- **Item PopItem():** Visszaadja a mezőn lévő legfelső eszközt, és eltávolítja a mezőről. Ha ez alatt volt másik eszköz, akkor innentől az fog látszani.
- **Item GetItem():** Visszaadja a mezőn lévő legfelső eszközt.
- **boolean IsOverWeight():** Igaz lesz, ha többen állnak a mezőn, mint annak a teherbíró képessége. Ekkor a mező átfordul.

3.3.6 Eskimo

Felelősség

Az Eskimo osztály egy játékbeli szereplő, az eszkimó. Különleges képessége, hogy tud ighut építeni, amelyben átvészhetőek a hóviharok. A játékot 5 életponttal kezdi.

- **Ősosztályok**

Player

- **Metódusok**

- **void BuildIgloo():** Az eszkimó ighut épít, ehhez egy munkát fel kell használnia. Csak stabil mezőre tud ighut építeni az eszkimó.

3.3.7 Flare

- **Felelősség**

A Flare osztály a jelzőrakéta egy alkatrészét reprezentálja, a jelzőfény. A rakéta összeszereléséhez van hozzá szükség. A szereplők tudják az eszköztárukban hordozni.

- **Interfészek**

Item

3.3.8 Food

- **Felelősség**

Az Food osztály egy felvehető tárgyat reprezentál, az élelmet, ami használatkor 1-el növeli a szereplő életpontját. Ez a tárgy is, a többihez hasonlóan, jégtáblán található, jégbe fagyva, és az eszköztárban 1 darab tárolható belőle. Ha a játékos elérte a maximum életpontját, akkor élelem megevésének hatására sem nő az életpontjainak száma.

- **Interfészek**

Item

3.3.9 Gun

- **Felelősség**

A Gun osztály a jelzőrakéta egy alkatrészét reprezentálja, a pisztolyt. A rakéta összeszereléséhez van hozzá szükség. A szereplők tudják az eszköztárban hordozni.

- **Interfészek**

Item

3.3.10 Item

- **Felelősség**

Az Item egy interfész, a játékbeli eszközöket reprezentálja. Ilyeneket tárol a szereplő az eszköztárában. A szereplő amikor használ egy eszközt, akkor ezen az interfészen keresztül kommunikál vele.

- **Metódusok**

- **void Use()**: Az eszköz használata.

3.3.11 Player

- **Felelősség**

A Player osztály reprezentál egy játékbeli szereplőt. Ez az objektum reagál a felhasználó interakcióira, azaz ő a játék aktív szereplője. Van egy eszköztára, amiben a kiásott tárgyakat tárolhatja. Tud mozogni, és dolgozni, azaz a tárgyait használni. Körökre osztva tevékenykedik, és minden körben adott számú energiája van, amely a munkáját korlátozza. Életpontjai is vannak, amiket ha elveszít, meghal.

- **Attribútumok**

- **#health : int** : A szereplő életpontjainak száma. Élelem megevésével lehet növelni. Ha elfogy, a játékos meghal.
- **#maxHealth: int**: A szereplő maximális életpontjainak száma. Ez az életpont felé nem lehet jutni élelemmel sem.
- **#energy : int** : A szereplő energiapontjainak száma az adott körben. Ez az attribútum a munkavégző-képességét számosítja. minden körben 4-et használhat belőle, ha elfogy, cselekvőképtelen lesz. Ha nem használja fel az összeset egy körben, akkor a megmaradt energiát nem viheti tovább a következő körre.
- **#wears_suit : boolean** : Megmondja, hogy a szereplő rendelkezik-e bűvárruhával. Ha igen, akkor minden esetben túléli a vízbe zuhanást.
- **#items: Item[0..5]** : A karakter eszköztára, ami egy ötelemű lista, és eszközök tárolódnak benne.
- **#currentField: Field**: Az a mező, amin a karakter aktuálisan áll.

- **Metódusok**

- **void PickItemUp()**: A szereplő ennek hatására veszi fel az aktuális mezőn lévő tárgyat, amennyiben az ki van ásva.
- **void DropItem(Item item)**: A szereplő ezzel a függvénytel a nála lévő, a paraméterében kiválasztott tárgyat leteszi a járatábra, amin áll.
- **void EndTurn()**: Ez a függvény hívódik meg, amikor a szereplő végez a körével, ekkor a következő szereplő köre következik.
- **void Move(Direction dir)**: Mozgatja a szereplőt a paraméterben megadott irányban lévő mezőre.
- **void UseItem(Item item)**: A szereplő használja a paraméterben kiválasztott eszközöt.
- **void AddItem(Item item)**: A szereplő eszköztárába beleteszi a paraméterben lévő eszközt.

- **void RemoveItem(Item item):** A szereplő eszköztárából kiveszi a paraméterben kiválasztott eszközt.

3.3.12 Rocket

- **Felelősség**

A Rocket osztály reprezentálja a játékban a jelzőrakétát, amelyet a szereplőknek össze kell szerelniük. Három részből áll, a pisztolyból, a jelzőfényből és a patronból. Kezdetben ez a három rész a pályán elszórva helyezkedik el. Az osztály felelőssége, hogy nyilvántartsa, hogy éppen melyik alkatrésze hol található. Csak akkor válik összeszerelhetővé és elsüthetővé, ha minden rész ugyanazon a mezőn található, és mindegyik játékos ezen a mezőn van.

- **Attribútumok**

- **boolean isOnOneField():** Igaz az attribútum értéke, ha a rakéta minden rész ugyanazon a jégtáblán található.
- **Gun gun:** A rakéta pisztoly alkotóeleme.
- **Flare flare:** A rakéta jelzőfény alkotóeleme.
- **Charge charge:** A rakéta patron alkotóeleme.

- **Metódusok**

- **void Assemble():** Összeszereli a rakétát az alkatrészeiből, amennyiben minden rész ugyanazon a mezőn van.
- **void Fire():** Elsüti a rakétát, amennyiben az össze van szerelve.

3.3.13 Rope

- **Felelősség**

A kötél segítségével egyik szereplő kihúzhatja egyik társát, ha az vízbe esett. A többi szereplő hasonlóan jégekhez hasonlóan megfagyva található meg és miután a szereplő fölveszi, válik használhatóvá.

- **Interfészek**

Item

3.3.14 Scientist

- **Felelősség**

A sarkkutató egy játékbeli szereplő. Különleges képessége, hogy meg tudja nézni, hogy az a jégtábla amire lépne, hány embert bír el. Gyengesége, hogy csak 4 életpontja van.

- **Ősosztályok**

Player

- **Metódusok**

- **void InspectField()**: A sarkkutató meg tudja vizsgálni a jégtáblákat stabilitás szempontjából, melyhez egy egységes munkát használ. Így megtudhatja hogy az adott jégtábla hány szereplőt bír el.

3.3.15 Shovel

- **Felelősség**

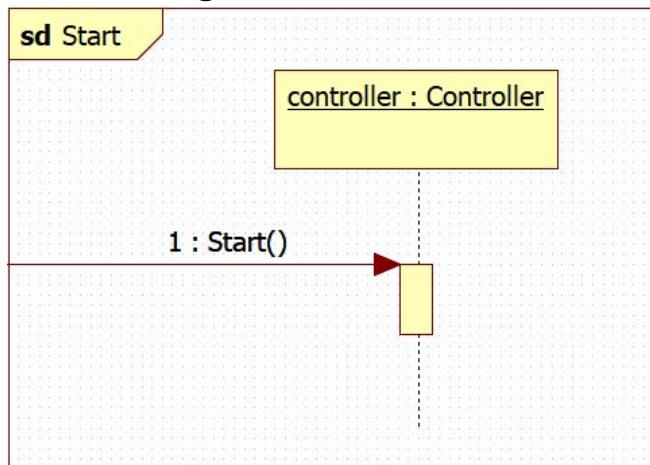
A Shovel osztály reprezentálja az lapátot, amelyet a szereplők hóeltakarításra használhatnak. Ha ez az eszköz a szereplőnél van, két egység havat tud ellapárolni. Az eszköz a többi tárgyhoz hasonlóan jégbe fagyva található meg.

- **Interfészek**

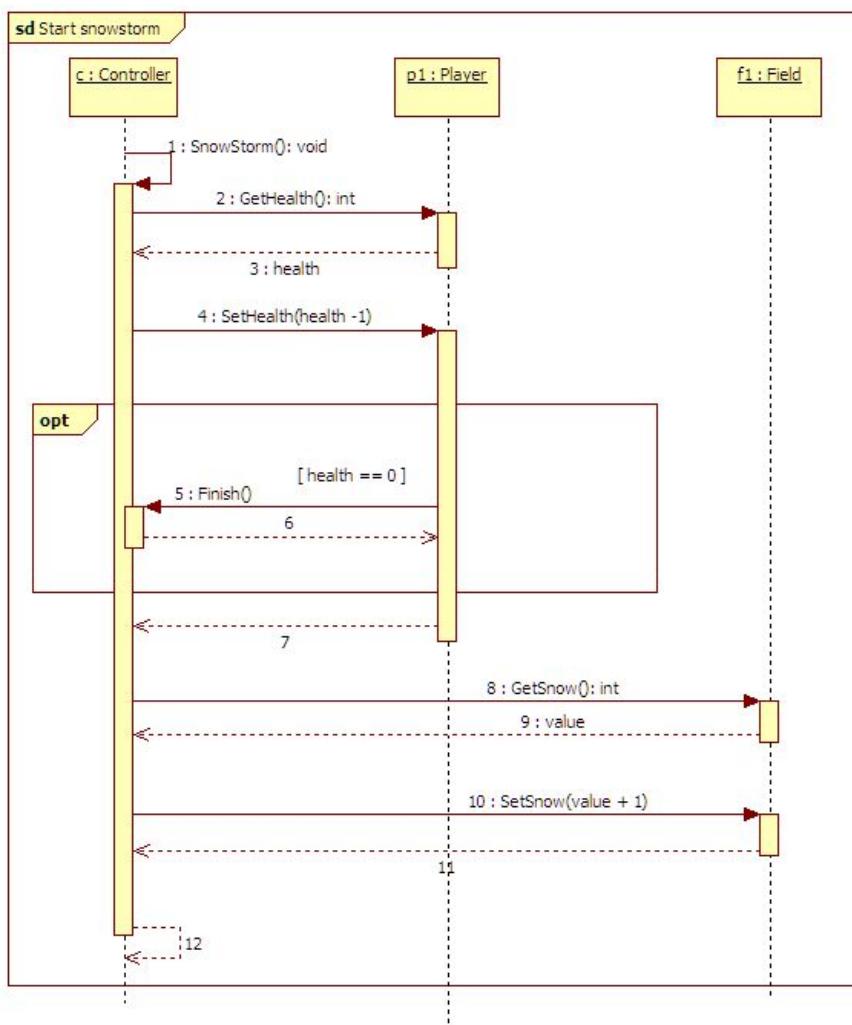
Item

3.4 Szekvencia diagramok

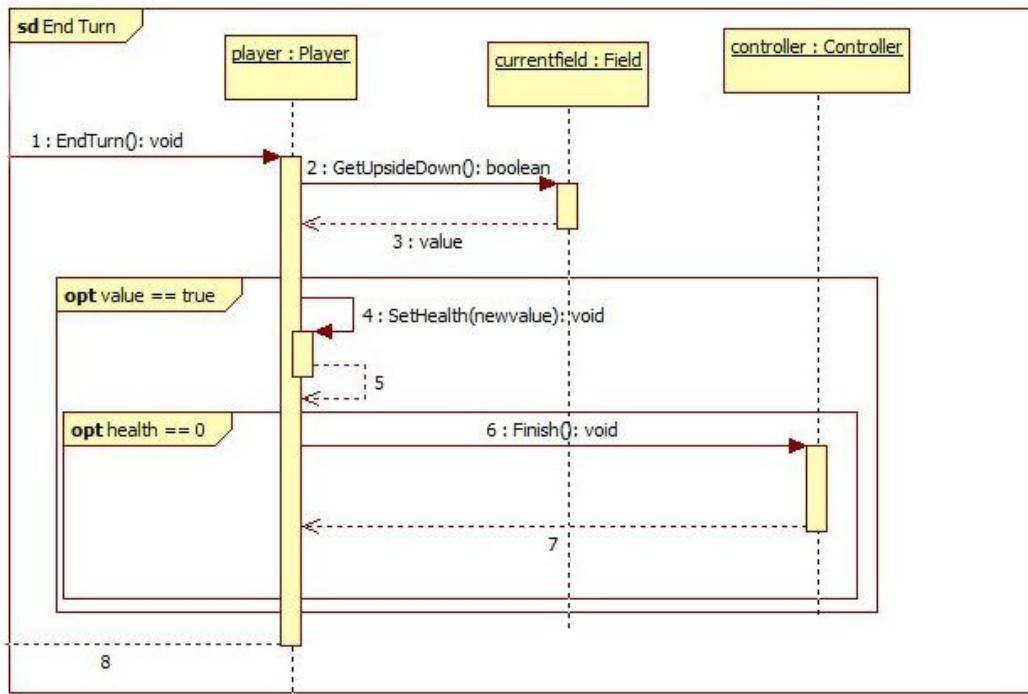
3.4.1 Start game



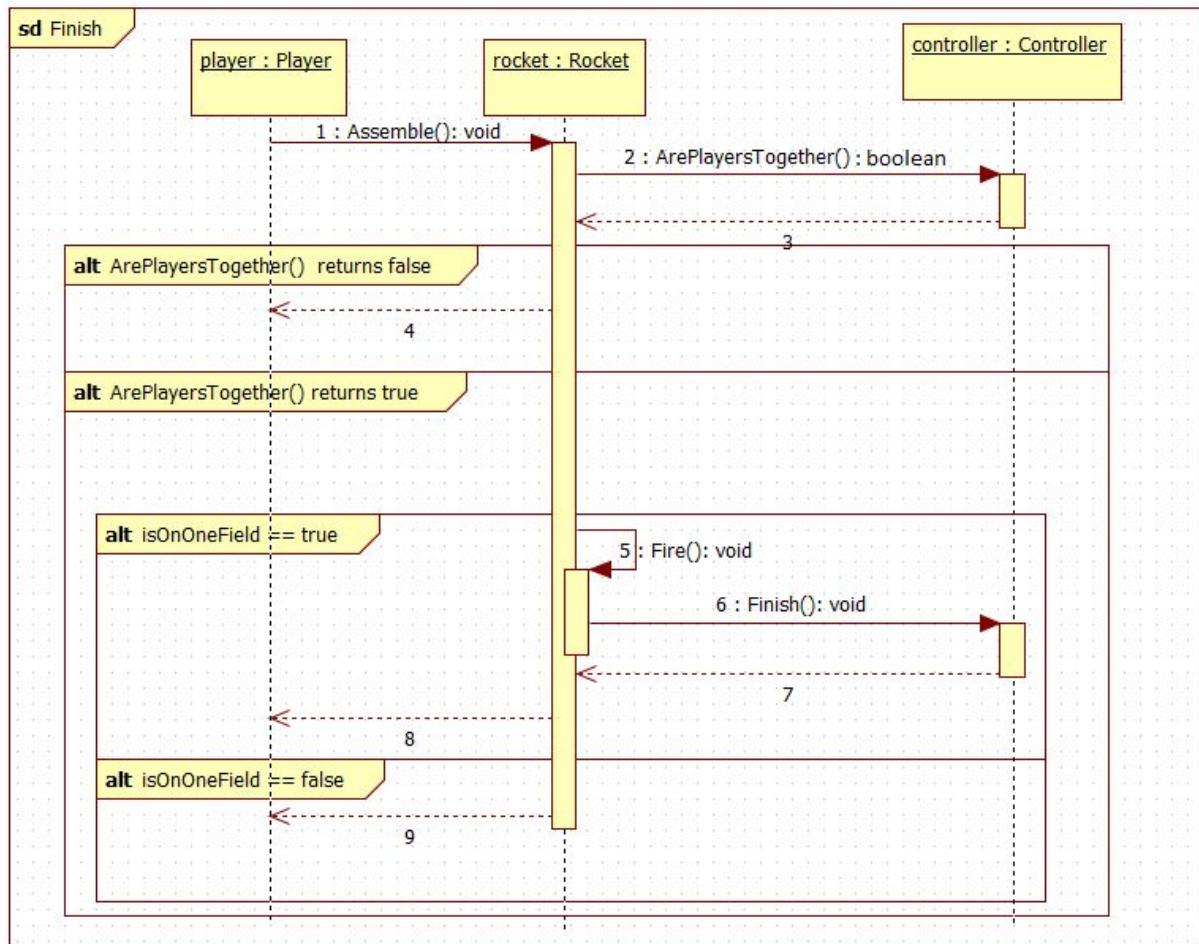
3.4.2 Start snowstorm



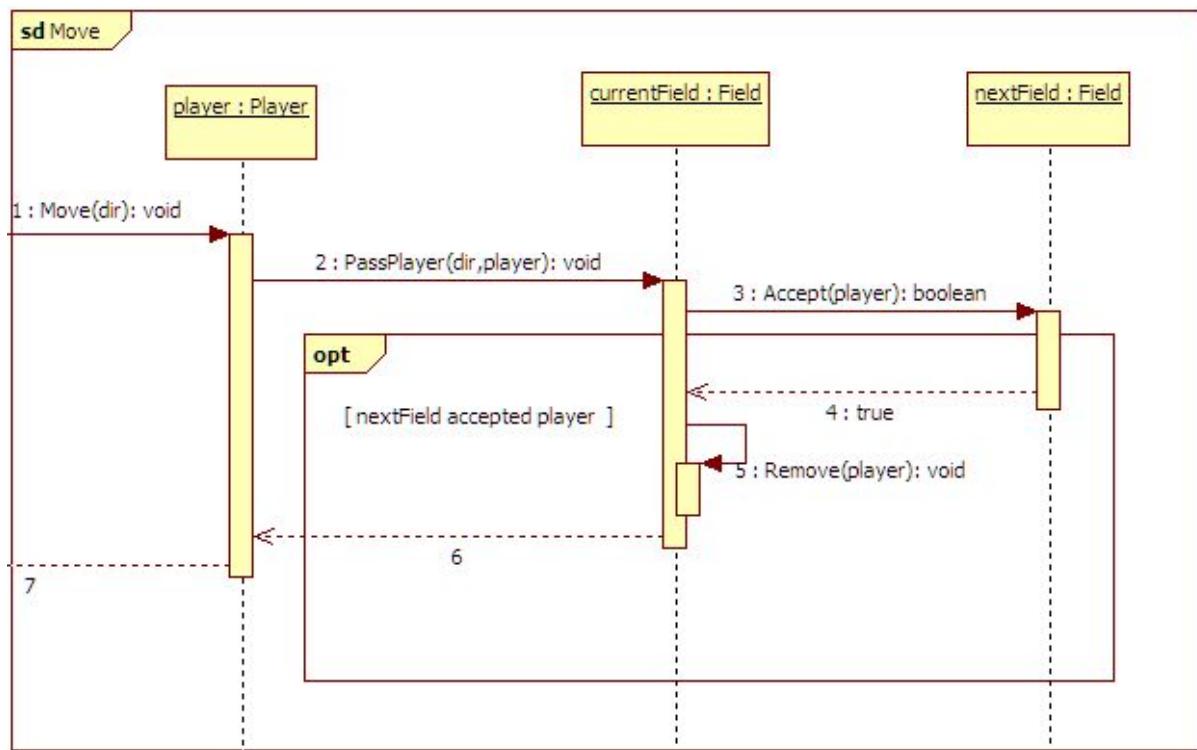
3.4.3 End turn



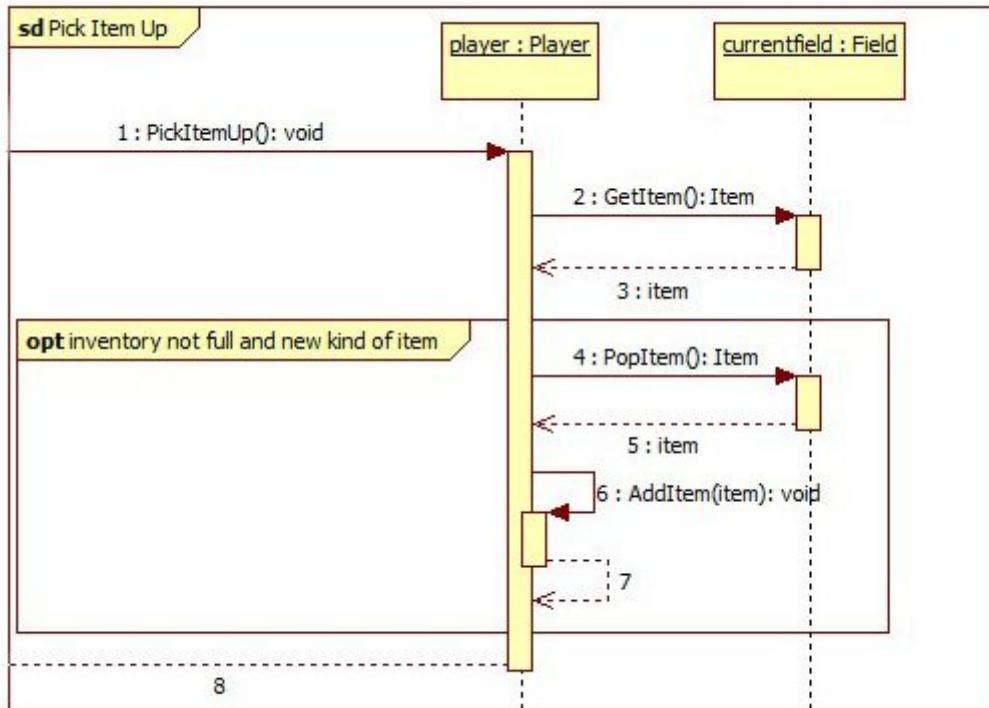
3.4.4 Finish



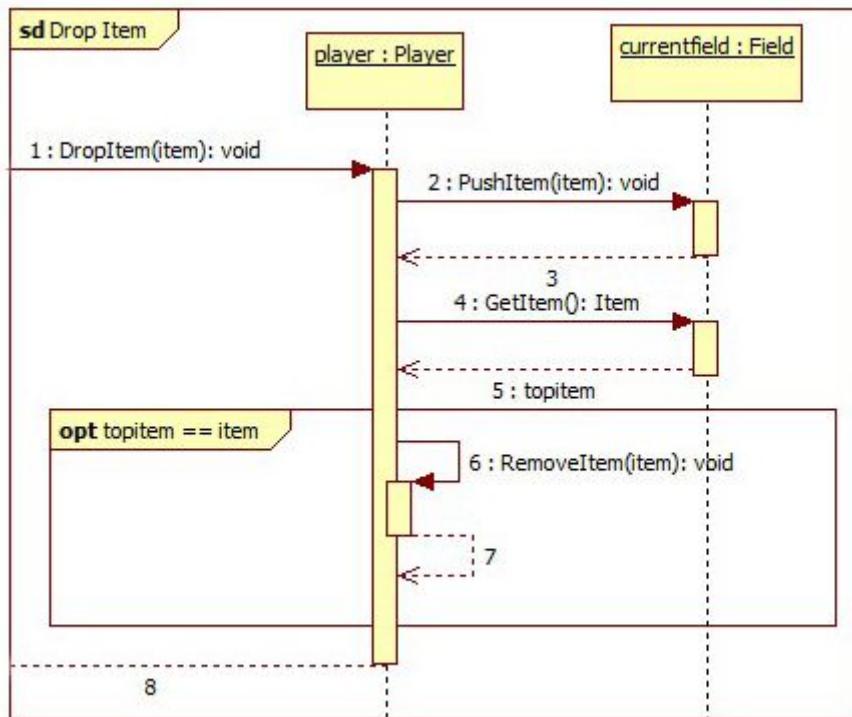
3.4.5 Move



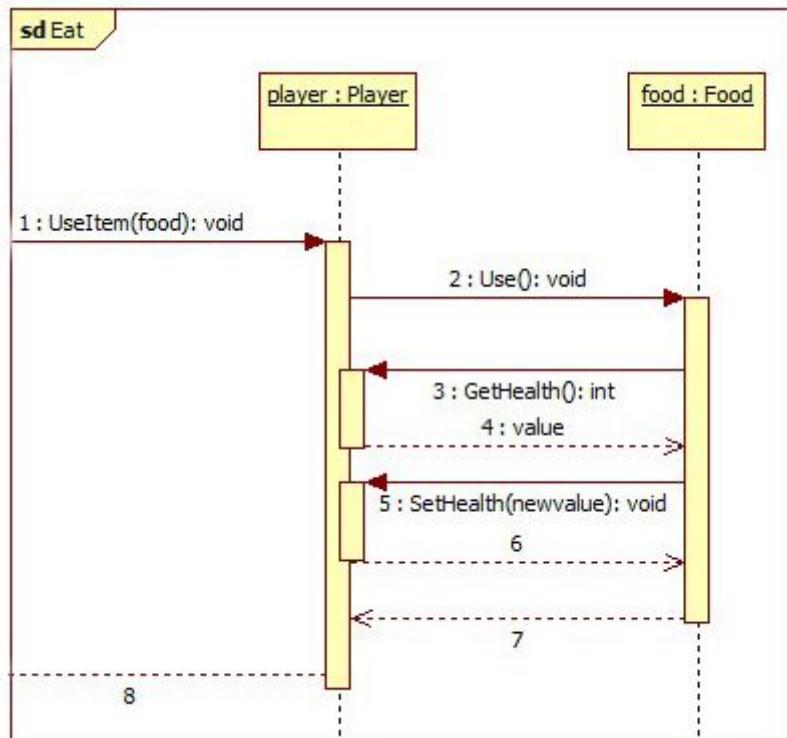
3.4.6 Pick item up



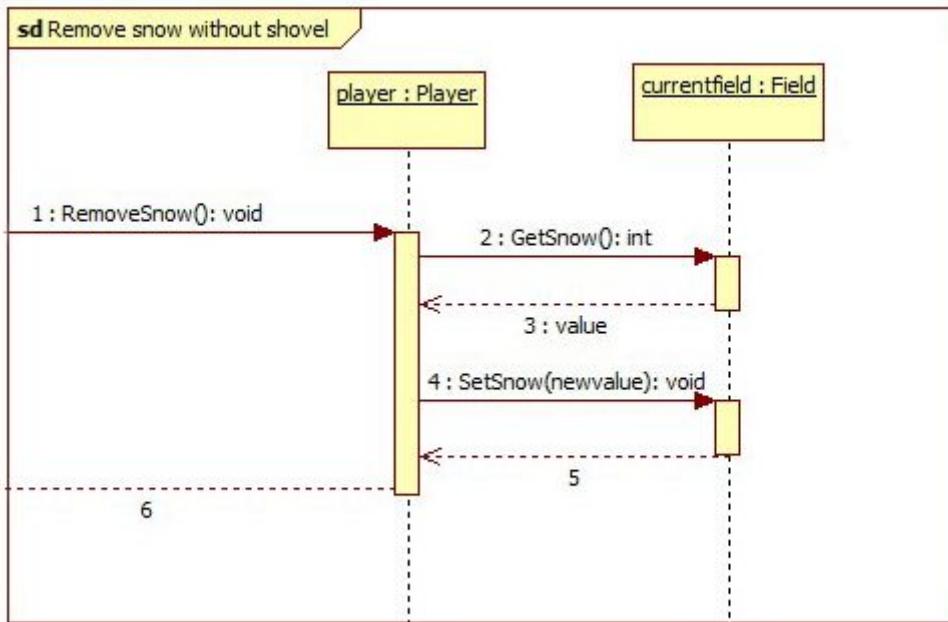
3.4.7 Drop item



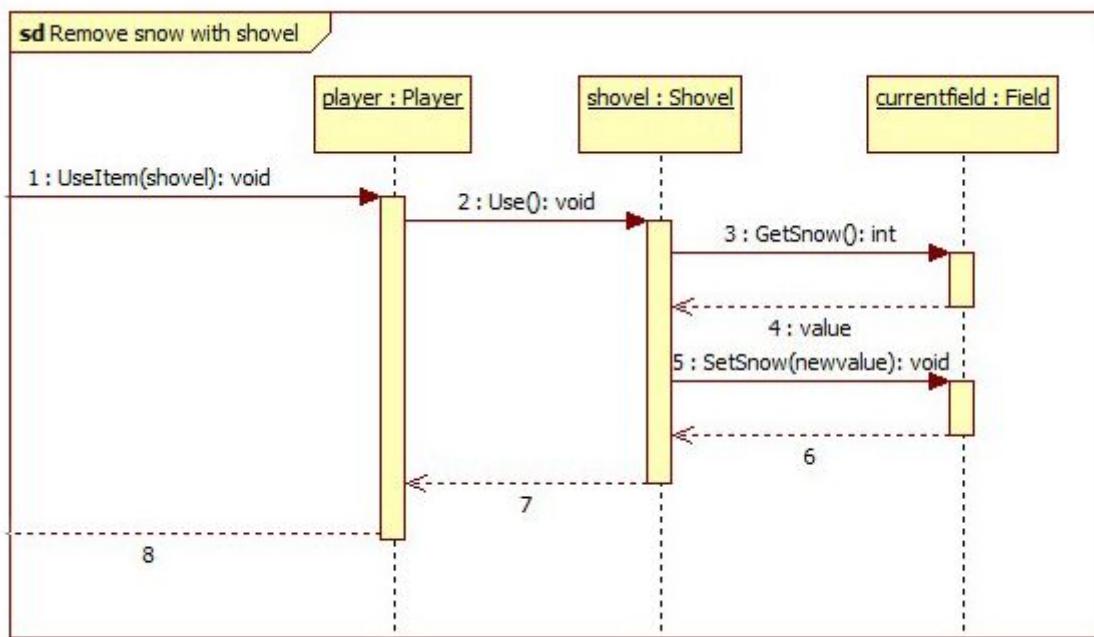
3.4.8 Eat



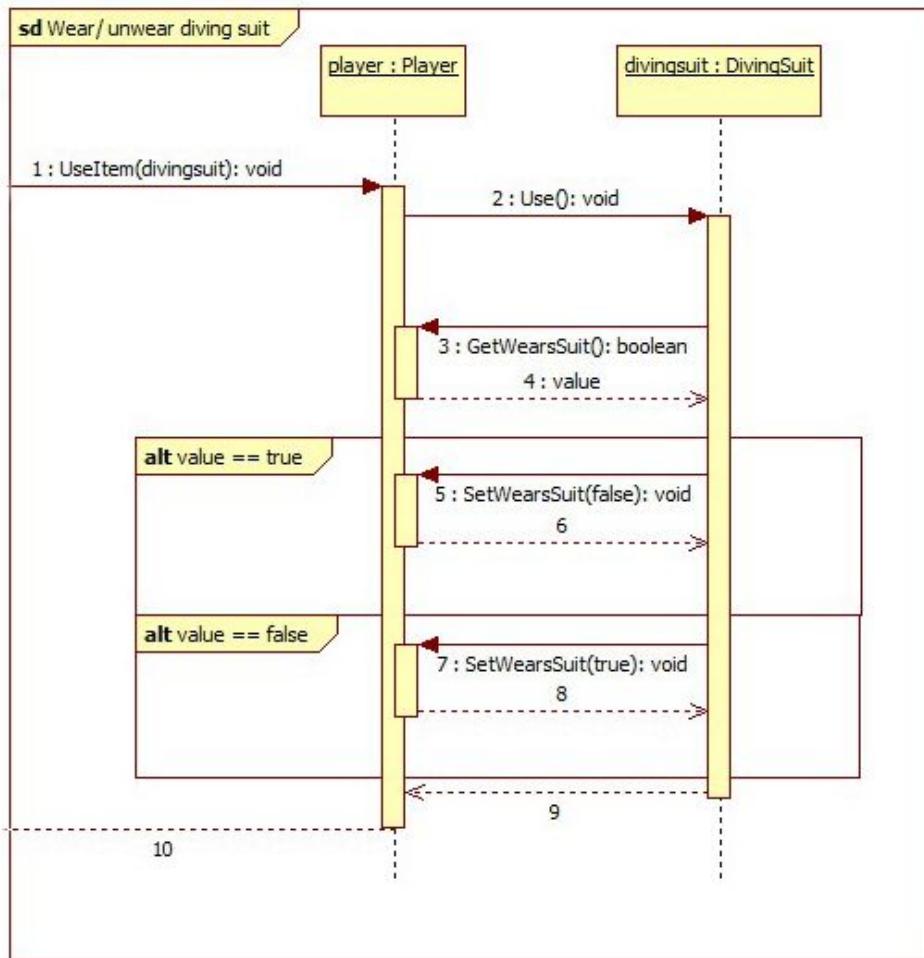
3.4.9 Remove snow without shovel



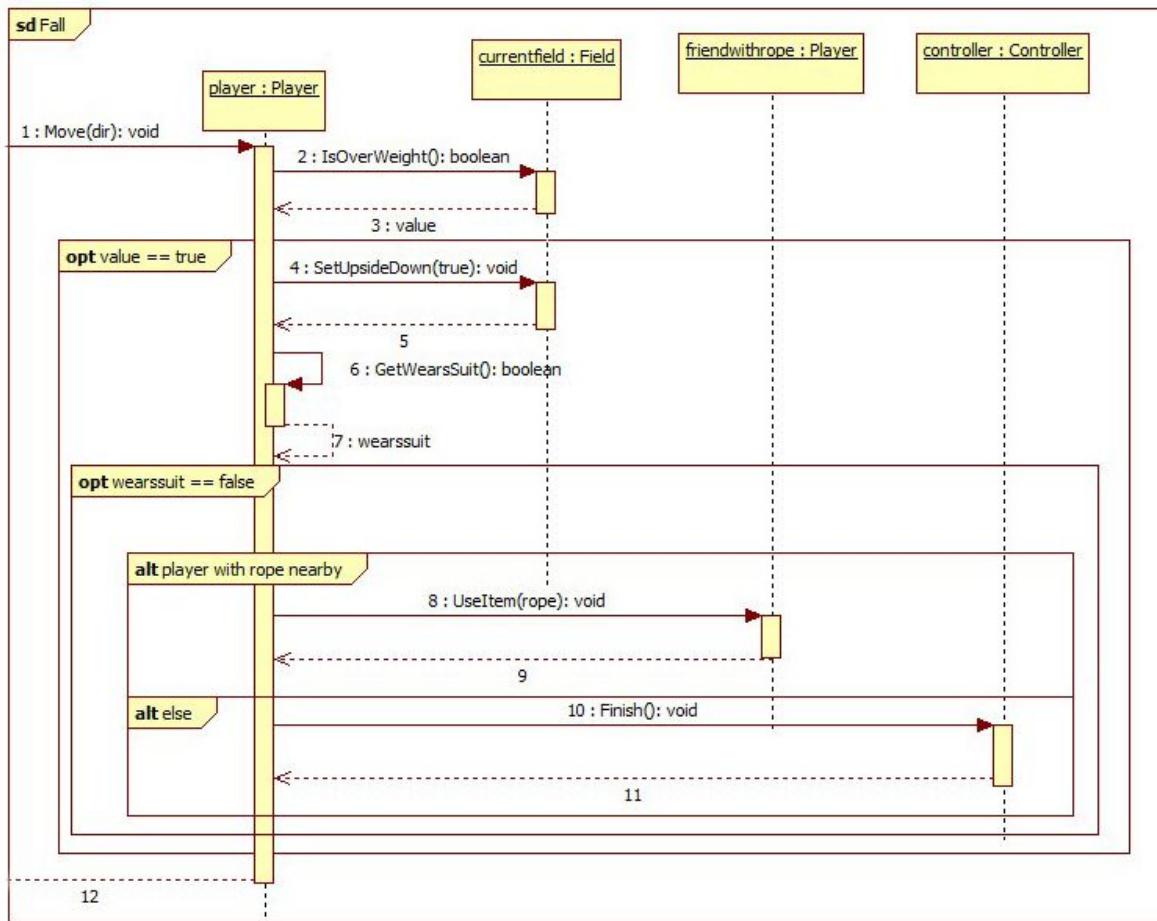
3.4.10 Remove snow with shovel



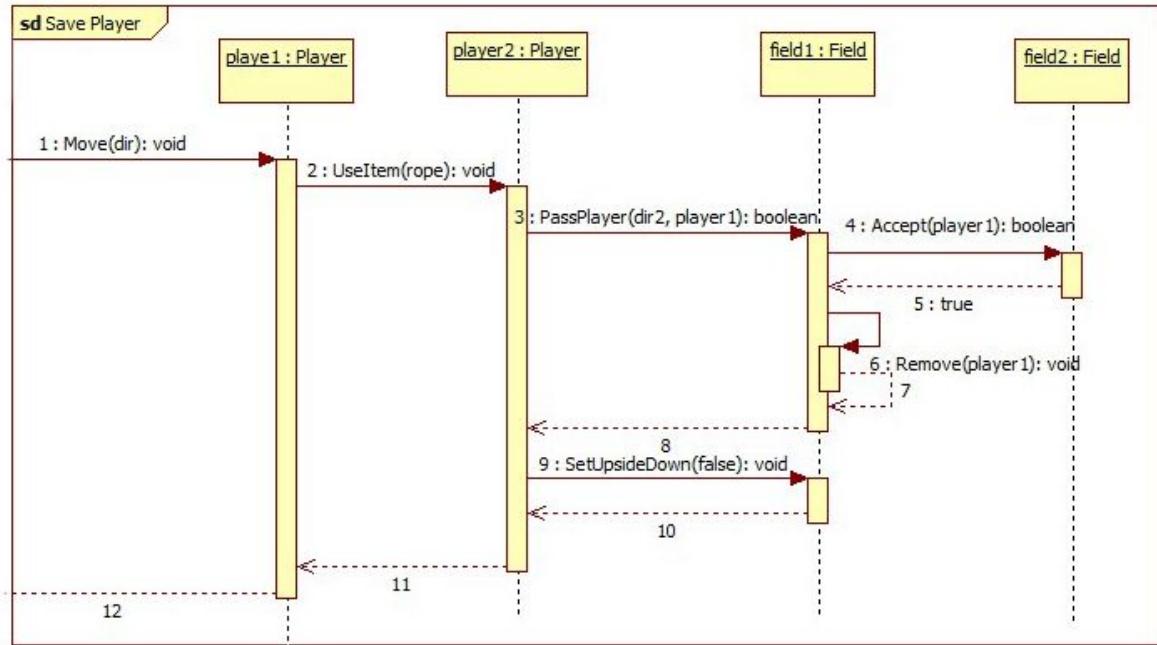
3.4.11 Wear/take off diving suit



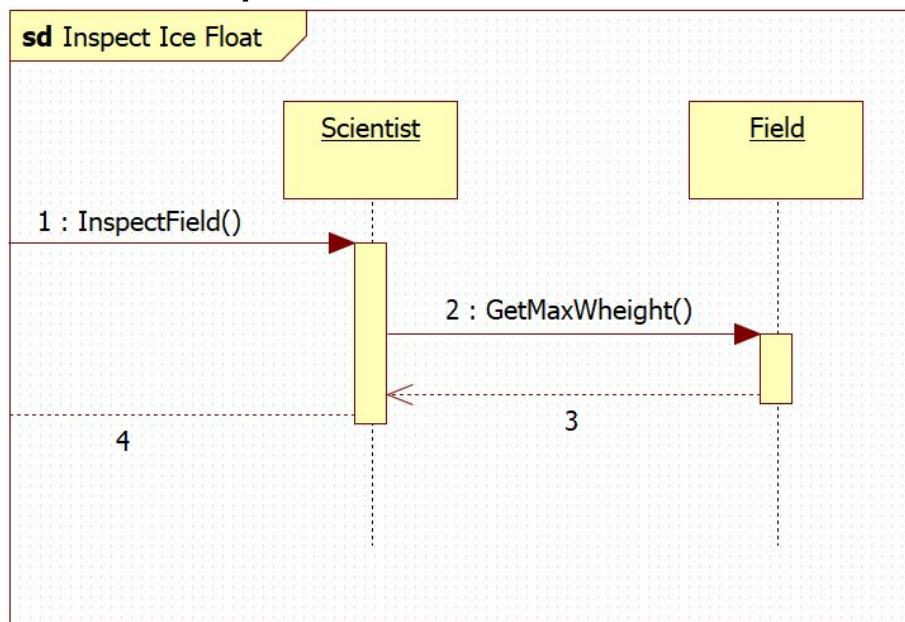
3.4.12 Fall



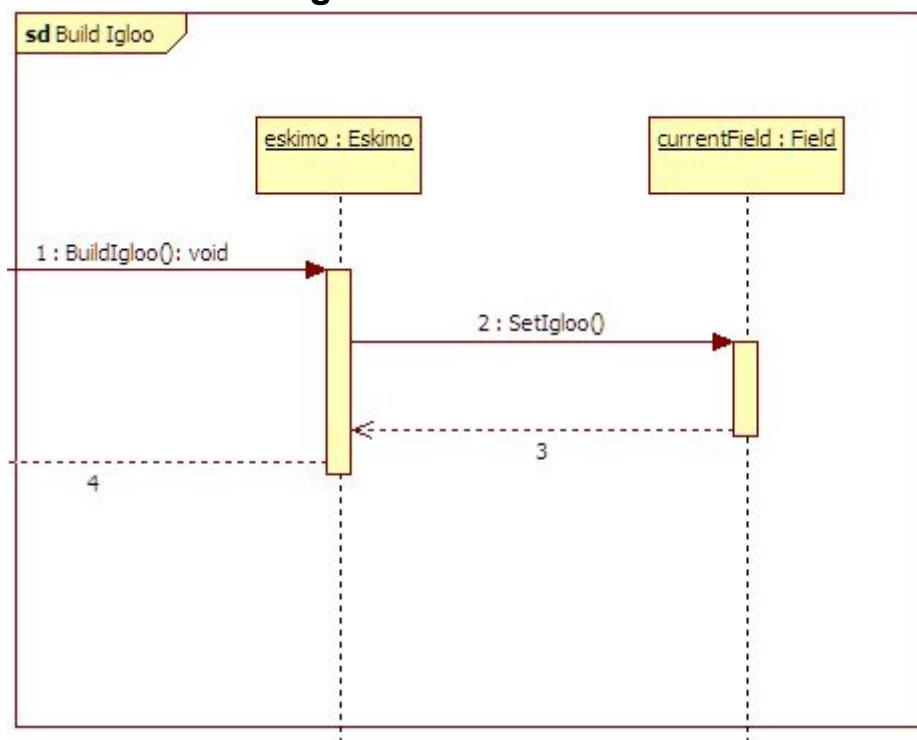
3.4.13 Save player



3.4.14 Inspect ice float



3.4.15 Build igloo



3.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.02.23. 22:00	2 óra	Filep	Tevékenység: Filep elkészíti az osztálydiagram első verzióját.
2020.02.24. 20:00	2 óra	Filep Jakabos Kálmán Kuzsner Puskás	Értekezés: Döntés: Puskás elkészíti az objektum katalógust, Jakabos és Kálmán az osztálydiagramot, Kuzsner és Filep Jakabossal együtt a szekvenciadiagramokat. Az osztályok leírásán megosztva mindenki dolgozik.
2020.02.26. 8:00	1 óra	Puskás	Tevékenység: Puskás elkészíti az objektum katalógus első verzióját. Felsorolja az objektumokat, illetve egy kezdetleges leírást írt nekik.
2020.02.26. 9:00	1 óra	Jakabos	Tevékenység: Jakabos elkészíti az osztálydiagram második verzióját illetve megrajzolta a Move szekvencia diagramot.
2020.02.26. 11:00	1 óra	Jakabos Kuzsner Filep	Értekezés a szekvencia diagramok elkészítéséről. Döntés: A diagramok felosztása, WhiteStarUML használata
2020.02.26. 12:00	1 óra	Kálmán	Tevékenység: Kálmán kiegészíti az osztálydiagramot javításokkal, illetve új osztályokkal.
2020.02.26. 20:00	1 óra	Jakabos	Tevékenység: Jakabos megrajzolta a Build Igloo és Start snowstorm szekvencia diagramokat.
2020.02.27. 12:00	2 óra	Filep	Tevékenység: Filep elkészíti az Eat, Remove Snow, Wear/ Unwear DivingSuit szekvencia diagramokat.
2020.02.27. 12:00	4 óra	Kuzsner	Tevékenység: Kuzsner elkészíti a Start, Finish, és Inspect Ice Float szekvencia diagramokat.
2020.02.27 16:00	3 óra	Filep	Tevékenység: Filep elkészíti a Drop Item, Pick Item Up, Save Player, Fall, End Turn szekvencia diagramokat.
2020.02.27 19:00	30 perc	Jakabos	Tevékenység: Jakabos kijavítja a Start snowstorm szekvencia diagramot.
2020.02.27 19:00	3 óra	Kálmán	Tevékenység: Kálmán elkészíti az osztályok leírását.
2020.02.27	2 óra	Puskás	Tevékenység: Objektum katalógus megírásának folytatása, osztályok leírásában segítség
2020.02.27 21:00	1 óra	Jakabos	Tevékenység: Jakabos elkészíti a Rope, Shovel és Scientist osztályok leírását.

2020.02.28 19:00	2 óra	Filep Jakabos Kálmán Kuzsner	Értekezés: Szekvencia diagramok átnézése, összevetése az osztály diagrammal, függvények viselkedésének megbeszélése. Döntés: Kálmán átdolgozza az osztálydiagramot, amit majd Kuzsner leellenőriz, és megformáz.
2020.02.29 12:00	2 óra	Kuzsner	Tevékenység: Inspect Ice Float szekvencia diagram bővítése/kijavítása, Diagramok lefedettségének vizsgálata, Osztály diagram kiegészítése és formázása, Kész diagramok dokumentumhoz csatolása.
2020.02.29 17:00	1 óra	Puskás	Tevékenység: Szekvencia diagramok elrendezése a dokumentumba, hibák javítása, dokumentum formázása, utolsó simítások.

04-Analízis modell kidolgozása 2.

47 – cereal_killers

Konzulens:

**Micskei Zoltán
Vörös András**

Csapattagok

Filep Leonárdó Márk	F6JQ0X	filep.leonardo@gmail.com
Kálmán Bendegúz	PTW6BD	kalmanbendeguz@gmail.com
Bence		
Jakabos Csenge Apolka	KO6MG	jakabos11@gmail.com
	0	
Puskás Levente	SN8YG9	puskas.levente99@gmail.co m
Kuzsner Sára	HVE3FV	kuzsnersara@gmail.com

2020.03.08.

4. Analízis modell kidolgozása

4.1 Objektum katalógus

4.1.1 Eszkimó

A két választható szereplőtípus közül az egyik. Energia- és életpontokkal rendelkezik minden szereplő. minden kör elején 4 energiapontja van, életpontja az eszkimónak 5 van. Ez a szereplő képes tárgyakat felvenni/lerakni, ezeket használni, lépni. Továbbiakban még felelőssége, hogy ha elfogynak az életpontjai, akkor meghaljon és a körvégét jelezze, ha nincs több energiapontja vagy ha nem kíván több munkát végezni. Az eszkimó egy különleges képességgel is rendelkezik, igyekezni tud építeni egy jégtáblára, ami a hóvihar ellen véd.

4.1.2 Sarkkutató

Ez a másik fajta szereplőtípus. Mindazon attribútumokkal és képességekkel rendelkezik, mint a eszkimó, leszámítva az eszkimó különleges képességét. Neki is megvan a számára egyedi képesség, hogy megtudja nézni mi van egy szomszédos jégtáblán. Továbbá még abban különbözik az eszkimótól, hogy 4 életponttal rendelkezik.

4.1.3 Jégtábla

Ezeknek az összessége alkotja a pályát. A jégtábla az az egységes terület, amin a szereplők és a tárgyak tartózkodhatnak. Felelőssége még az is, hogy ismerje a mellette lévő szomszédos jégtáblákat, és biztosítsa a szereplők ezekre való átlépését.

4.1.4 Kontroller

Ez irányítja a játék menetét. Feladatai közé tartozik, hogy elindítja a játékmenetet, kezeli a hóviharokat, befejezi a játékot, ha nyert a csapat vagy ha meghalt egy szereplő.

4.1.5 Lapát

Segítségével a szereplő havat tud ellapárolni egy jégtábláról. Jégből fagyva található meg, mint ahogy a többi tárgy is.

4.1.6 Búvárruha

Használata életet menthet olyan esetben, ha egy szereplő beleszik a vízbe, miközben ez rajta van. A búvárruha nélkül meghal a vízbe esett szereplő.

4.1.7 Kötél

Abban segít a kötél, hogyha egy szereplő rendelkezik vele és egy szomszédos mezőn egy szereplő beleszedett a vízbe, akkor segítségével kihúzhatja a vízből.

4.1.8 Élelem

Ha egy játékos elhasználja, akkor növeli az életpontjait 1-el. Hasonlóan a többi tárgyhoz, ez is a pályán véletlenszerűen helyezkedik el.

4.1.9 Jelzőrakéta

A játék célja, hogy a csapat a rakéta alkatrészeit megtalálja és összeszerelje, majd közösen elszüsse. Három ilyen alkatrészre van szükség: pisztoly, jelzőfény, patron. Felelőssége, hogy össze lehessen szerelni és elszütni, és tudja magáról, hogy egy jégtáblán van.

4.1.10 Pisztoly

A jelzőrakéta egyik alkatrésze, elengedhetetlen kellék a rakéta összeszereléséhez. A pályán ugyanúgy megtalálható, mint a többi tárgy.

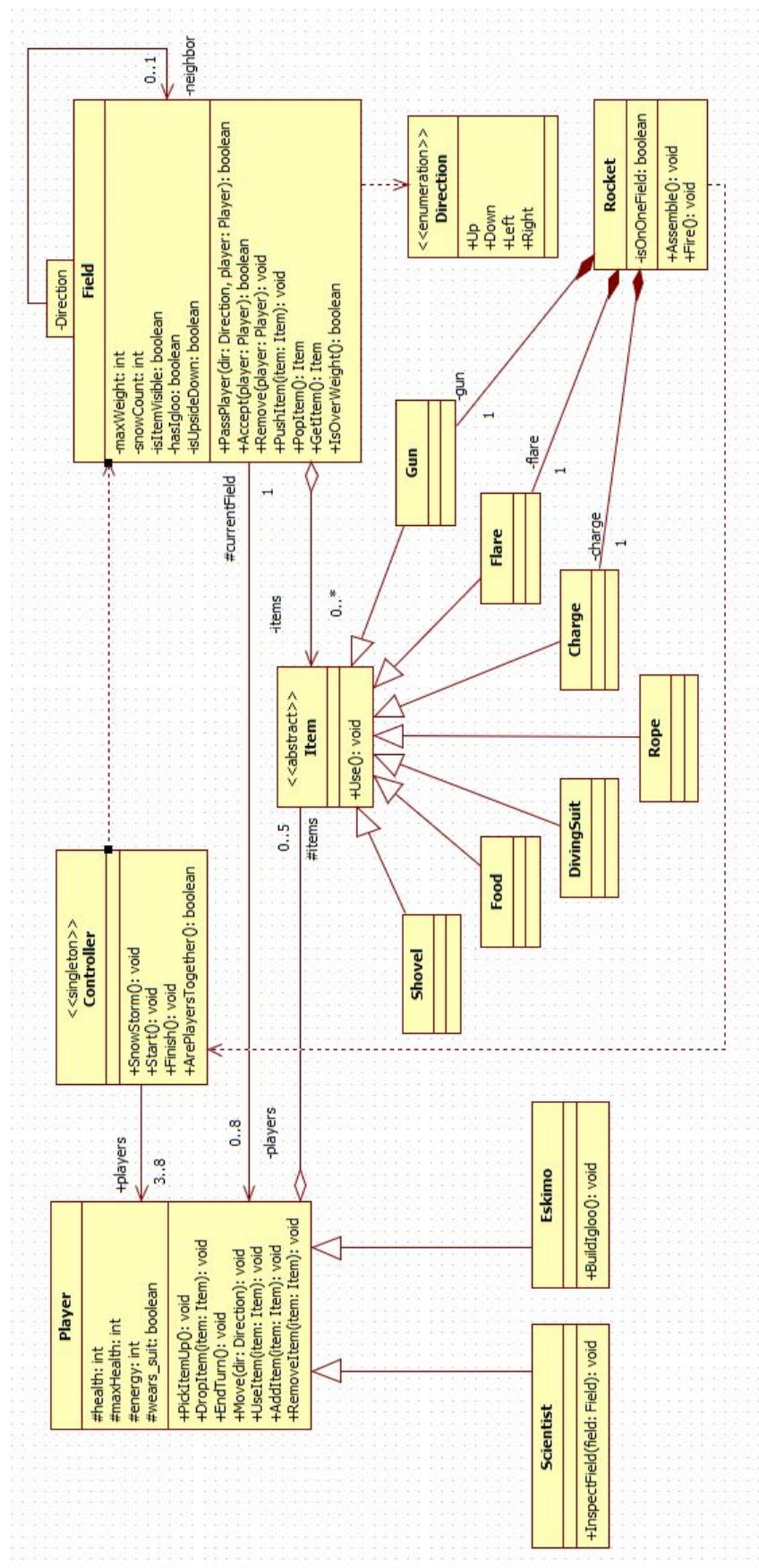
4.1.11 Jelzőfény

A jelzőrakéta egy másik alkatrésze, erre is szükség van az összeszereléséhez.

4.1.12 Patron

A jelzőrakéta harmadik és utolsó alkatrésze, ugyanaz jellemző rá, mint a többi alkatrészre.

4.2 Statikus struktúra diagramok



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

4.3 Osztályok leírása

4.3.1 Charge

- **Felelősség**

A Charge osztály a jelzőrakéta egy alkatrészét reprezentálja, a patront. A rakéta összeszereléséhez van hozzá szükség. A szereplők tudják az eszköztárukban hordozni.

Az Item osztály leszármazottja.

4.3.2 Controller

- **Felelősség**

A Controller osztály a játékbeli hóviharokat kezeli, valamint ő indítja a játékot és ő is fejezi be, azaz nyeréskor vagy vesztéskor vele kell kommunikálni.

- **Metódusok**

- **void SnowStorm()**: A játékbeli jégmezőn hóvihart indít, melynek hatására véletlenszerű jégmezőkön egy-egy újabb höréteg jelenik meg.
- **void Start()**: Elkezdi a játékot, inicializálja az objektumokat, és odaadja az első szereplőnek a kezdés jogát.
- **void Finish()**: Ennek hatására befejeződik a játék.
- **boolean ArePlayersTogether()**: Azt vizsgálja ez a függvény, hogy a szereplők egy jégtáblán vannak-e, ekkor igazat ad vissza. Erre azért van szükség, mert a rakéta csak ekkor válik összeszerelhetővé.

4.3.3 Direction

- **Felelősség**

Egy felsorolt típus, amely a játékban az irányokat adja meg, ez négyfélé lehet, ugyanis az átlósan lévő mezőkre nem lehet átkelni.

- **Attribútumok**

- **Up**: A felfelé mutató irány.
- **Down**: A lefelé mutató irány.
- **Left**: A balra mutató irány.
- **Right**: A jobbra mutató irány.

4.3.4 DivingSuit

- **Felelősség**

A DivingSuit osztály egy búvárruhát reprezentál. Ezek alapból a jége vannak fagyva, és ugyanúgy fel lehet őket venni, mint más eszközöket. Ha egy szereplő megszerzi, akkor onnantól kezdve ha vízbe esik, nem fog meghalni.

Az Item osztály leszármazottja.

4.3.5 Field

- **Felelősség**

A Field osztály reprezentál egy játékbeli jégtáblát. Ilyenkből áll össze a pálya, négyzetrács alakban helyezkednek el. Felelőssége, hogy kezelje a rajta lévő tárgyakat, és átforduljon, ha a teherbíró képességénél több szereplő helyezkedik el rajta. Ismeri a rajta lévő szereplőket, és mozgásnál ő rakja át a karaktereket a szomszédos jégtáblára.

- **Attribútumok**

- **-maxWeight: int:** A jégtábla maximum teherbíró képessége. Ha ennél többen állnak rajta, átfordul.
- **-snowCount: int:** Azt mondja meg, hogy a jégtáblán hány réteg hó van.
- **-isItemVisible: boolean:** Igaz, ha a jégtáblán látható a rajta lévő eszköz, azaz egy réteg hó sincs a jégtáblán. Származtatott attribútum a snowCount-ból.
- **-hasIgloo: boolean:** Igaz, ha a jégtáblán van iglu.
- **-isUpsideDown: boolean:** Igazat ad vissza, ha a jégtábla fel van fordulva.

- **Metódusok**

- **boolean PassPlayer(Direction dir, Player player):** Az argumentumban megadott szereplőt átrakja az argumentumban megadott irányban elhelyezkedő szomszédos mezőre, majd visszatér azzal, hogy ez sikerült-e (sikertelen az áthelyezés például ha abban az irányban tenger van).
- **boolean Accept(Player player):** A paraméterben megadott szereplőt rárakja az adott jégtáblára. Igazzal tér vissza, ha ez sikerült.
- **void Remove(Player player):** A paraméterben megadott szereplőt eltávolítja a jégtábláról.
- **void PushItem(Item item):** A paraméterben megadott eszközt leteszi a mezőre.
- **Item PopItem():** Visszaadja a mezőn lévő legfelső eszközt, és eltávolítja a mezőről. Ha ez alatt volt másik eszköz, akkor innentől az fog látszani.
- **Item GetItem():** Visszaadja a mezőn lévő legfelső eszközt.
- **boolean IsOverWeight():** Igaz lesz, ha többen állnak a mezőn, mint annak a teherbíró képessége. Ekkor a mező átfordul.

4.3.6 Eskimo

• Felelősség

Az Eskimo osztály egy játékbeli szereplő, az eszkimó. Különleges képessége, hogy tud ighut építeni, amelyben átvészhetőek a hóviharok. A játékot 5 életponttal kezdi.

• Ősosztályok

Player

• Metódusok

- **void BuildIgloo():** Az eszkimó ighut épít, ehhez egy munkát fel kell használnia. Csak stabil mezőre tud ighut építeni az eszkimó.

4.3.7 Flare

• Felelősség

A Flare osztály a jelzőrakéta egy alkatrészét reprezentálja, a jelzőfény. A rakéta összeszereléséhez van hozzá szükség. A szereplők tudják az eszköztárakban hordozni.

Az Item osztály leszármazottja.

4.3.8 Food

• Felelősség

Az Food osztály egy felvehető tárgyat reprezentál, az élelmet, ami használatkor 1-el növeli a szereplő életpontját. Ez a tárgy is, a többihez hasonlóan, jégtáblán található, jégbe fagyva, és az eszköztárban 1 darab tárolható belőle. Ha a játékos elérte a maximum életpontját, akkor élelem megevésének hatására sem nő az életpontjainak száma.

Az Item osztály leszármazottja.

4.3.9 Gun

• Felelősség

A Gun osztály a jelzőrakéta egy alkatrészét reprezentálja, a pisztolyt. A rakéta összeszereléséhez van hozzá szükség. A szereplők tudják az eszköztárakban hordozni.

Az Item osztály leszármazottja.

4.3.10 Item

• Felelősség

Az Item egy absztrakt osztály, a játékbeli eszközöket reprezentálja. Ilyeneket tárol a szereplő az eszköztárában. A szereplő amikor használ egy eszközt, akkor ezen az interfészen keresztül kommunikál vele.

- **Metódusok**
 - **void Use()**: Az eszköz használata.

4.3.11 Player

- **Felelősség**

A Player osztály reprezentál egy játékbeli szereplőt. Ez az objektum reagál a felhasználó interakcióira, azaz ő a játék aktív szereplője. Van egy eszköztára, amiben a kiásott tárgyakat tárolhatja. Tud mozogni, és dolgozni, azaz a tárgyait használni. Körökre osztva tevékenykedik, és minden körben adott számú energiája van, amely a munkáját korlátozza. Életpontjai is vannak, amiket ha elveszít, meghal.

- **Attribútumok**
 - **#health : int** : A szereplő életpontjainak száma. Élelem megevésével lehet növelni. Ha elfogy, a játékos meghal.
 - **#maxHealth: int**: A szereplő maximális életpontjainak száma. Ez az életpont felé nem lehet jutni élelemmel sem.
 - **#energy : int** : A szereplő energiapontjainak száma az adott körben. Ez az attribútum a munkavégző-képességét számosítja. minden körben 4-et használhat belőle, ha elfogy, cselekvőképtelen lesz. Ha nem használja fel az összeset egy körben, akkor a megmaradt energiát nem viheti tovább a következő körre.
 - **#wears_suit : boolean** : Megmondja, hogy a szereplő rendelkezik-e bűvárruhával. Ha igen, akkor minden esetben túlél a vízbe zuhanást.
 - **#items: Item[0..5]** : A karakter eszköztára, ami egy ötelemű lista, és eszközök tárolódnak benne.
 - **#currentField: Field**: Az a mező, amin a karakter aktuálisan áll.
- **Metódusok**
 - **void PickItemUp()**: A szereplő ennek hatására veszi fel az aktuális mezőn lévő tárgyat, amennyiben az ki van ásva.
 - **void DropItem(Item item)**: A szereplő ezzel a függvénytel a nála lévő, a paraméterében kiválasztott tárgyat leteszi a jégtáblára, amin áll.
 - **void EndTurn()**: Ez a függvény hívódik meg, amikor a szereplő végez a körével, ekkor a következő szereplő köre következik.
 - **void Move(Direction dir)**: Mozgatja a szereplőt a paraméterben megadott irányban lévő mezőre.
 - **void UseItem(Item item)**: A szereplő használja a paraméterben kiválasztott eszközöt.
 - **void AddItem(Item item)**: A szereplő eszköztárába beleteszi a paraméterben lévő eszközt.
 - **void RemoveItem(Item item)**: A szereplő eszköztárából kiveszi a paraméterben kiválasztott eszközt.

4.3.12 Rocket

- **Felelősség**

A Rocket osztály reprezentálja a játékban a jelzőrakétát, amelyet a szereplőknek össze kell szerelniük. Három részből áll, a pisztolyból, a jelzőfényből és a patronból. Kezdetben ez a

három rész a pályán elszórva helyezkedik el. Az osztály felelőssége, hogy nyilvántartsa, hogy éppen melyik alkatrésze hol található. Csak akkor válik összeszerelhetővé és elsüthetővé, ha minden rész ugyanazon a mezőn található, és minden rész ezen a mezőn van.

- **Attribútumok**

- **boolean isOnOneField()**: Igaz az attribútum értéke, ha minden rész ugyanazon a jégtáblán található.
- **Gun gun**: A rakéta pisztoly alkotóeleme.
- **Flare flare**: A rakéta jelzőfény alkotóeleme.
- **Charge charge**: A rakéta patron alkotóeleme.

- **Metódusok**

- **void Assemble()**: Összeszereli a rakétát az alkatrészeiből, amennyiben minden rész ugyanazon a mezőn van.
- **void Fire()**: Elsüti a rakétát, amennyiben az össze van szerelve.

4.3.13 **Rope**

- **Felelősség**

A kötél segítségével egyik szereplő kihúzhatja egyik társát, ha az vízbe esett. A többi eszközhöz hasonlóan jégbe fagyva található meg és miután a szereplő fölveszi, válik használhatóvá.

Az Item osztály leszármazottja.

4.3.14 Scientist

- **Felelősség**

A sarkkutató egy játékbeli szereplő. Különleges képessége, hogy meg tudja nézni, hogy az a jégtábla amire lépne, hány embert bír el. Gyengesége, hogy csak 4 életpontja van.

- **Ősosztályok**

Player

- **Metódusok**

- **void InspectField()**: A sarkkutató meg tudja vizsgálni a jégtáblákat stabilitás szempontjából, melyhez egy egységes munkát használ. Így megtudhatja hogy az adott jégtábla hány szereplőt bír el.

4.3.15 Shovel

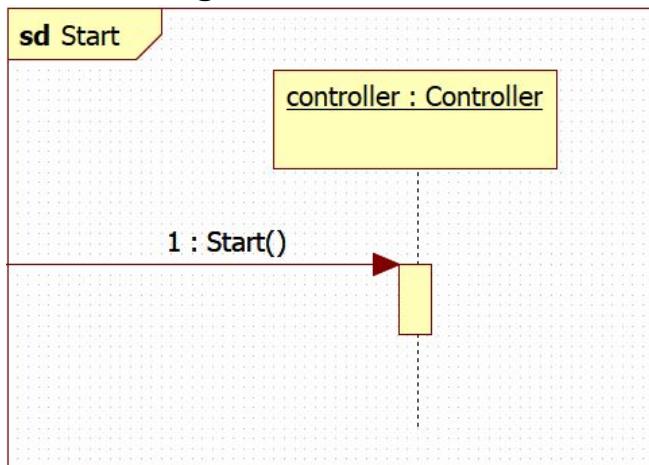
- **Felelősség**

A Shovel osztály reprezentálja az lapátot, amelyet a szereplők hóeltakarításra használhatnak. Ha ez az eszköz a szereplőnél van, két egység havat tud ellapárolni. Az eszköz a többi tárgyhoz hasonlóan jégbe fagyva található meg.

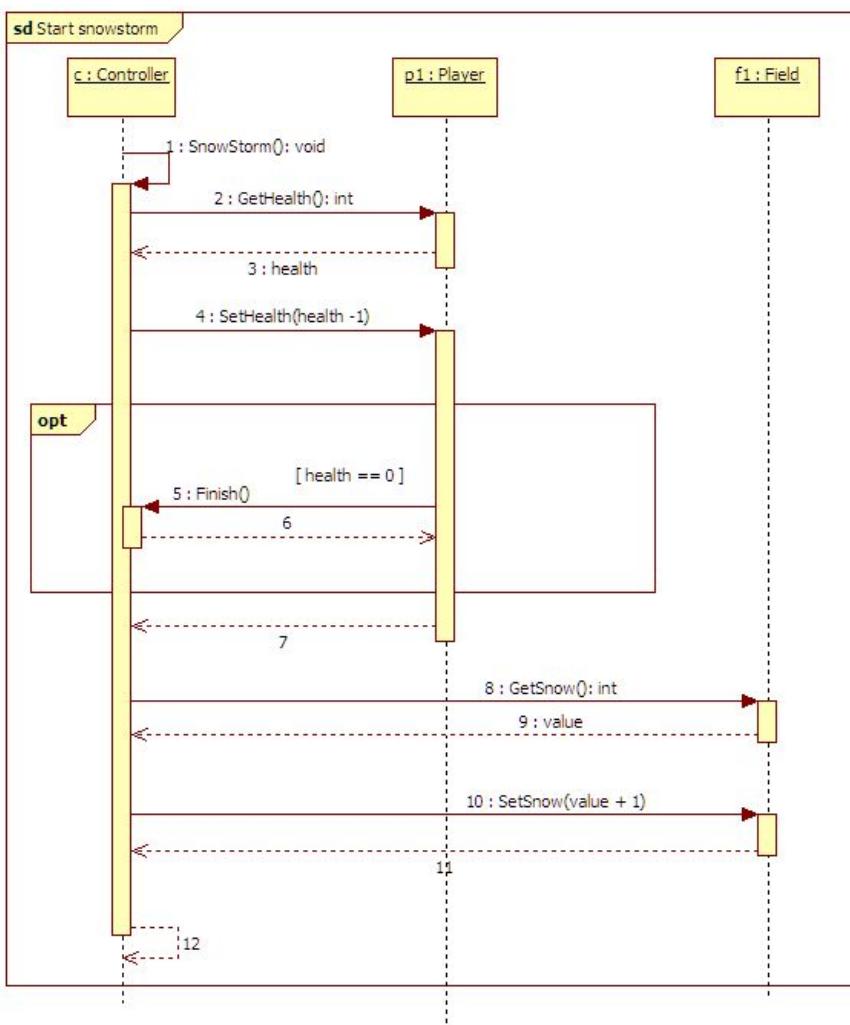
Az Item osztály leszármazottja.

4.4 Szekvencia diagramok

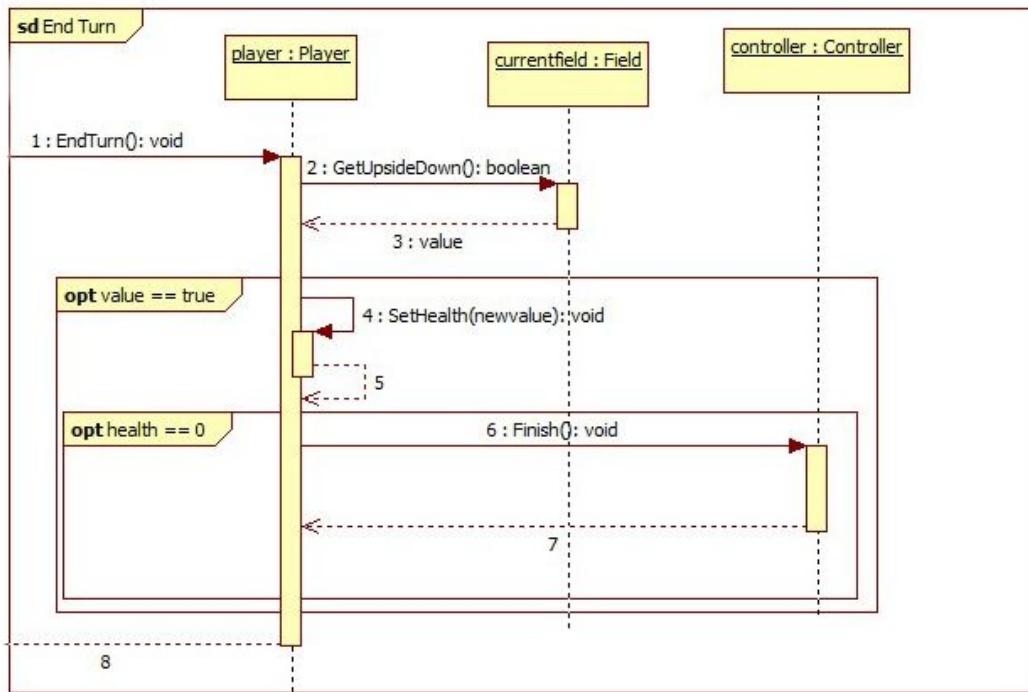
4.4.1 Start game



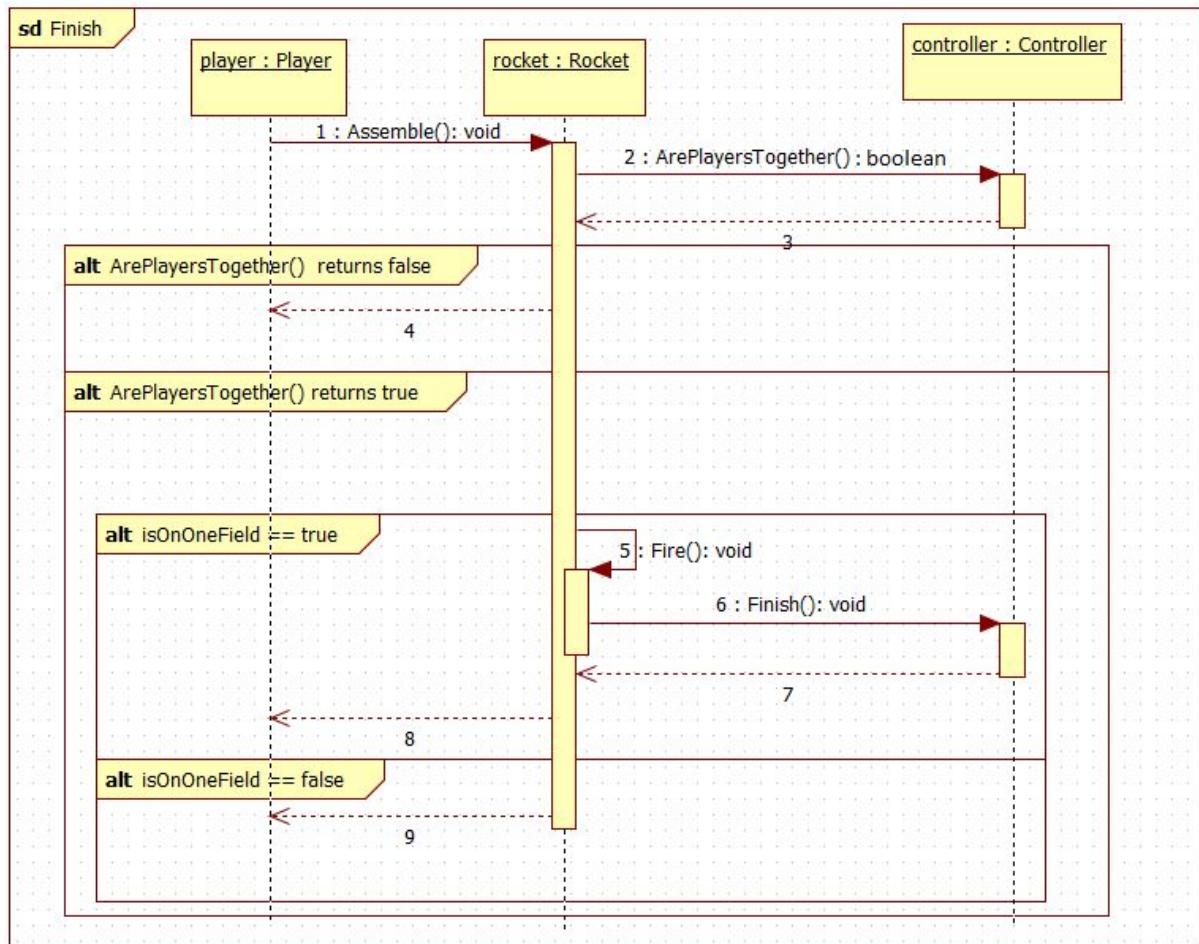
4.4.2 Start snowstorm



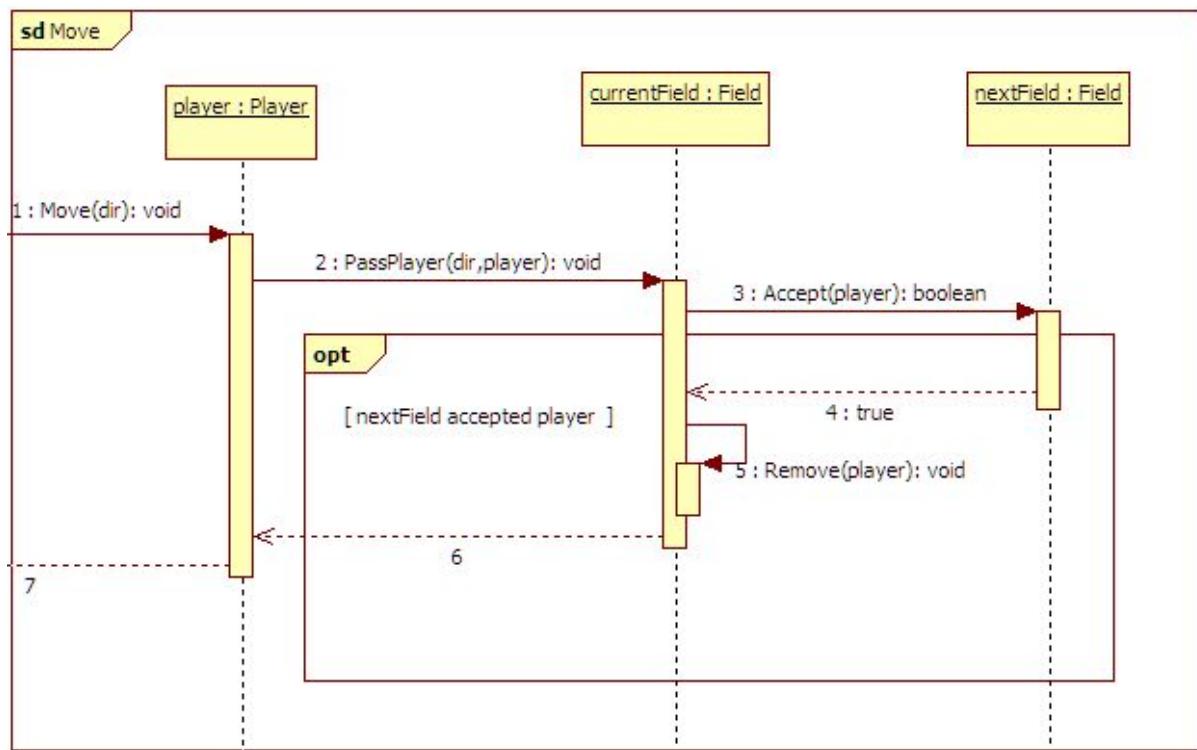
4.4.3 End turn



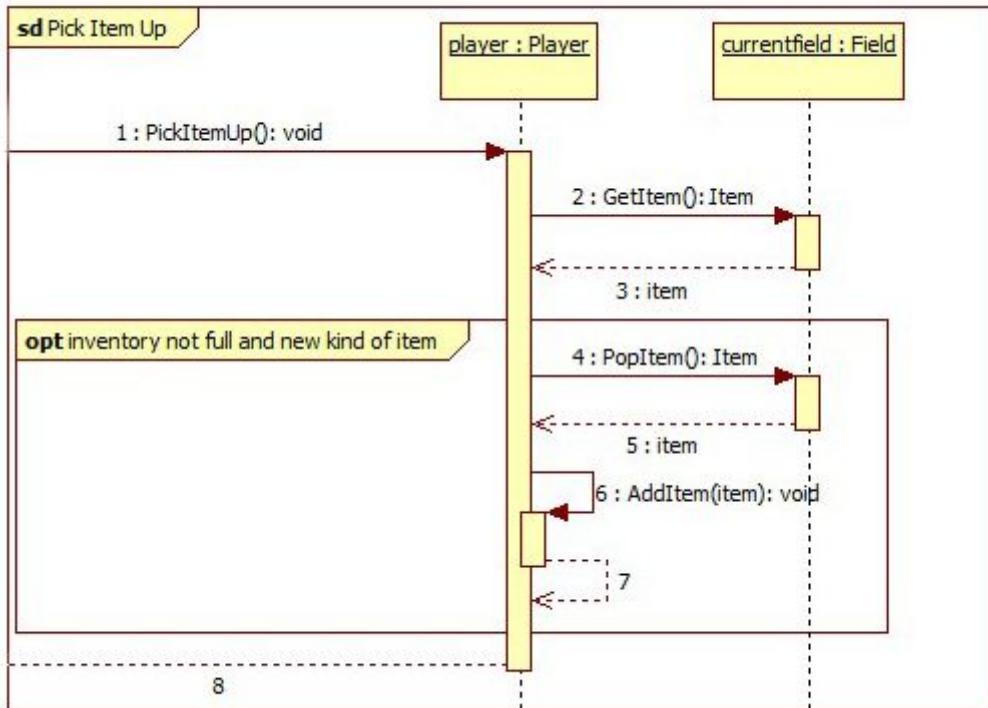
4.4.4 Finish



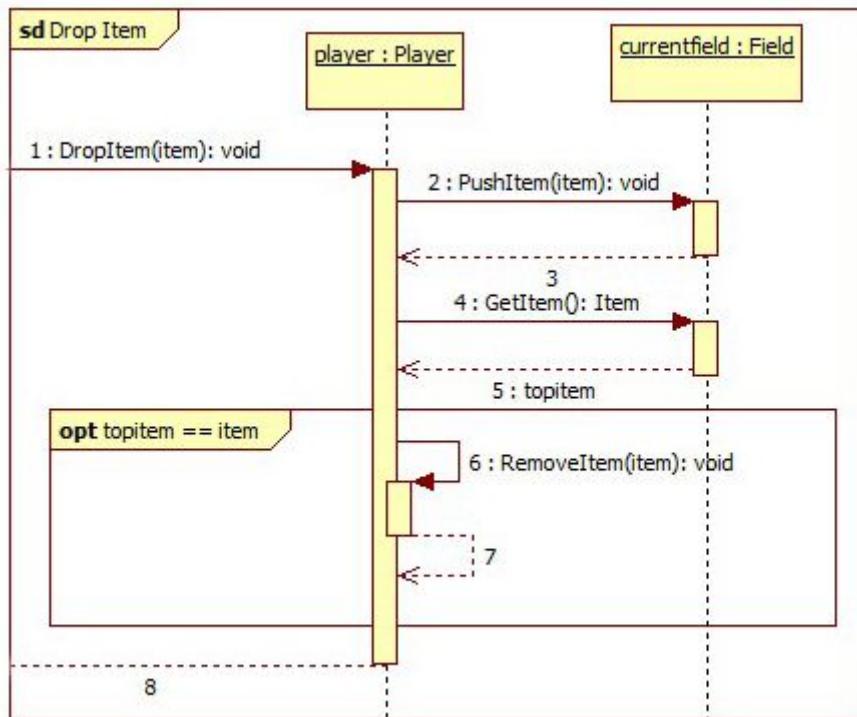
4.4.5 Move



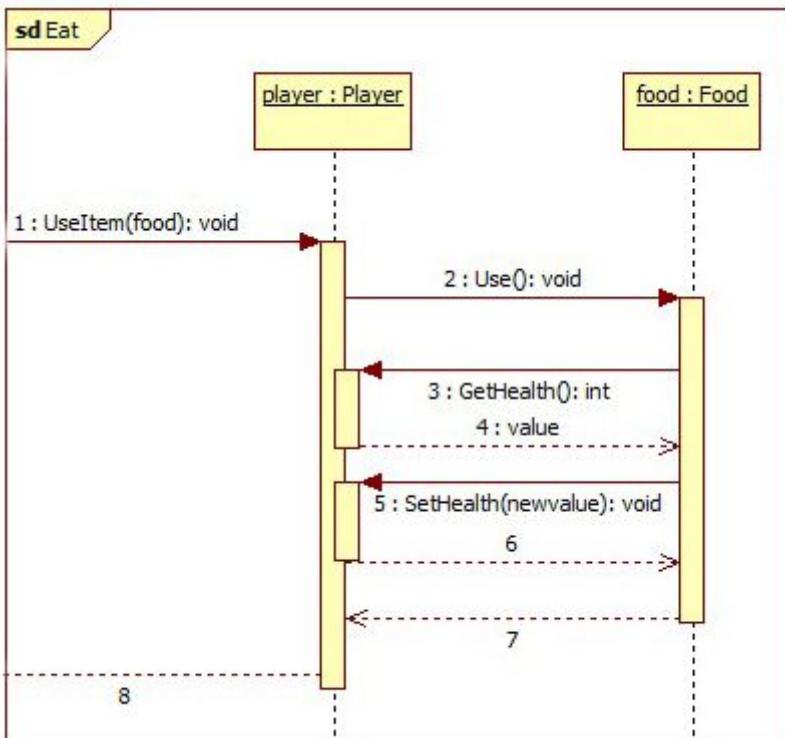
4.4.6 Pick item up



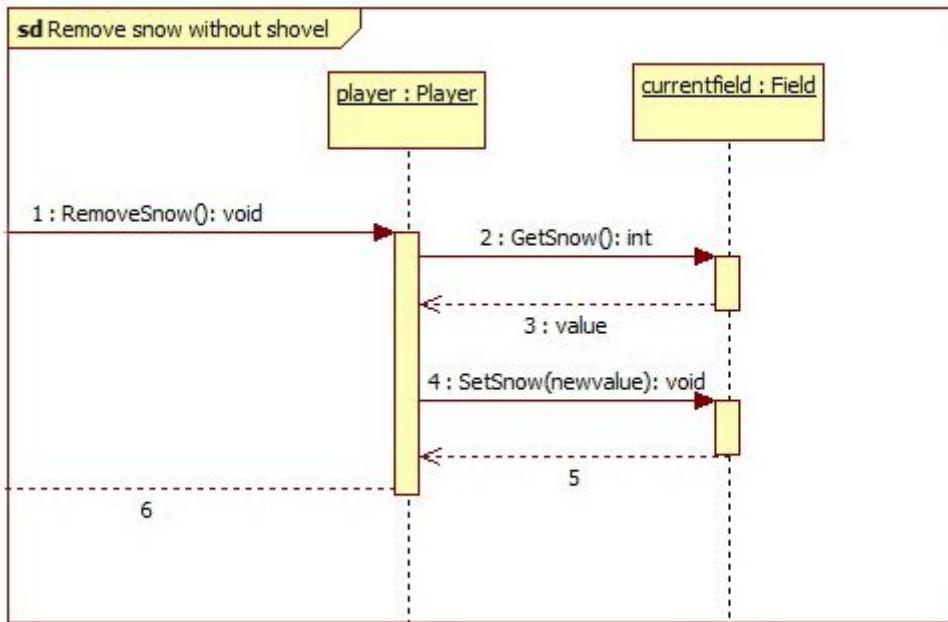
4.4.7 Drop item



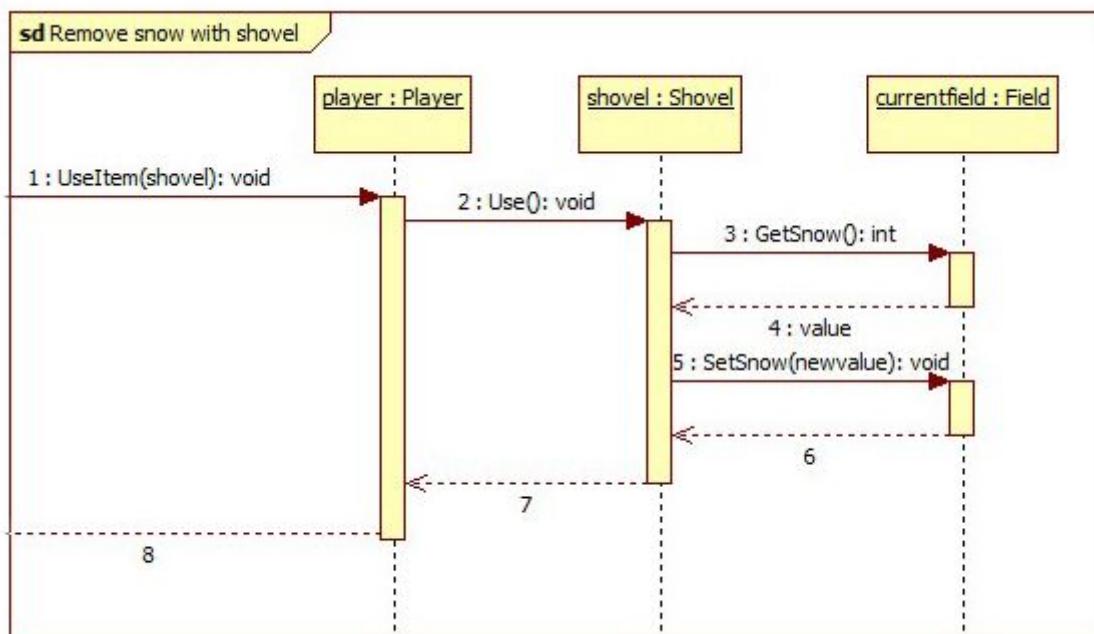
4.4.8 Eat



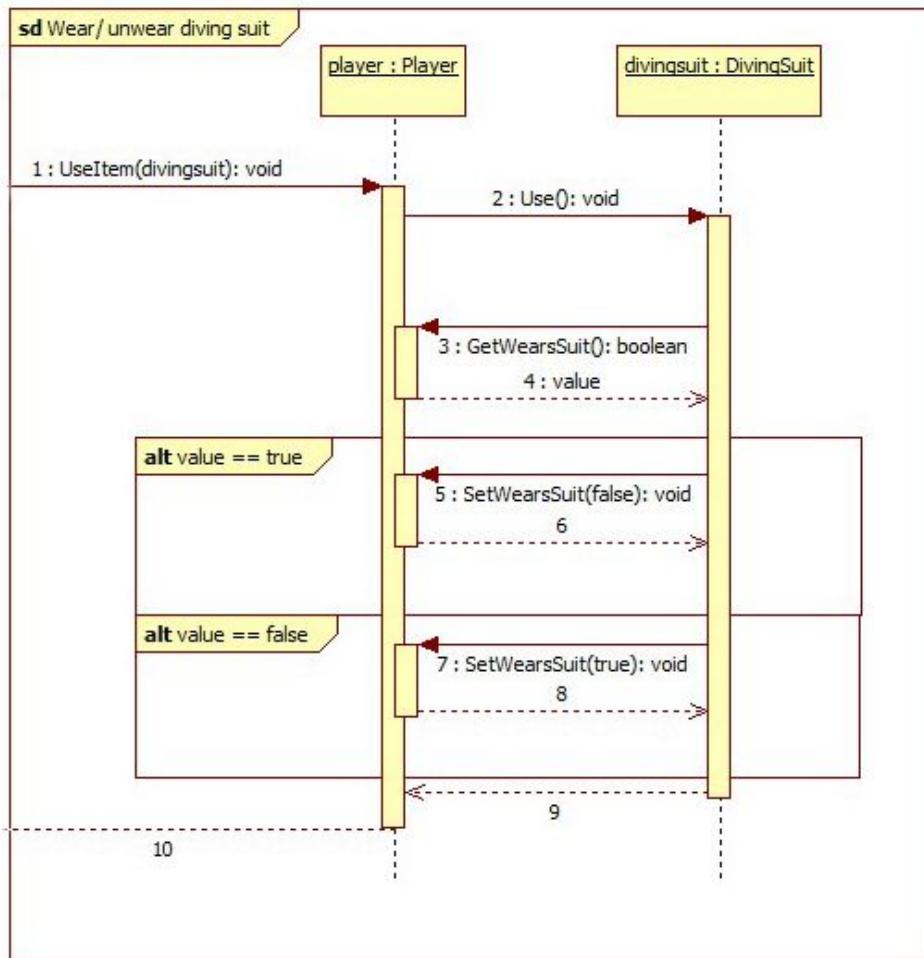
4.4.9 Remove snow without shovel



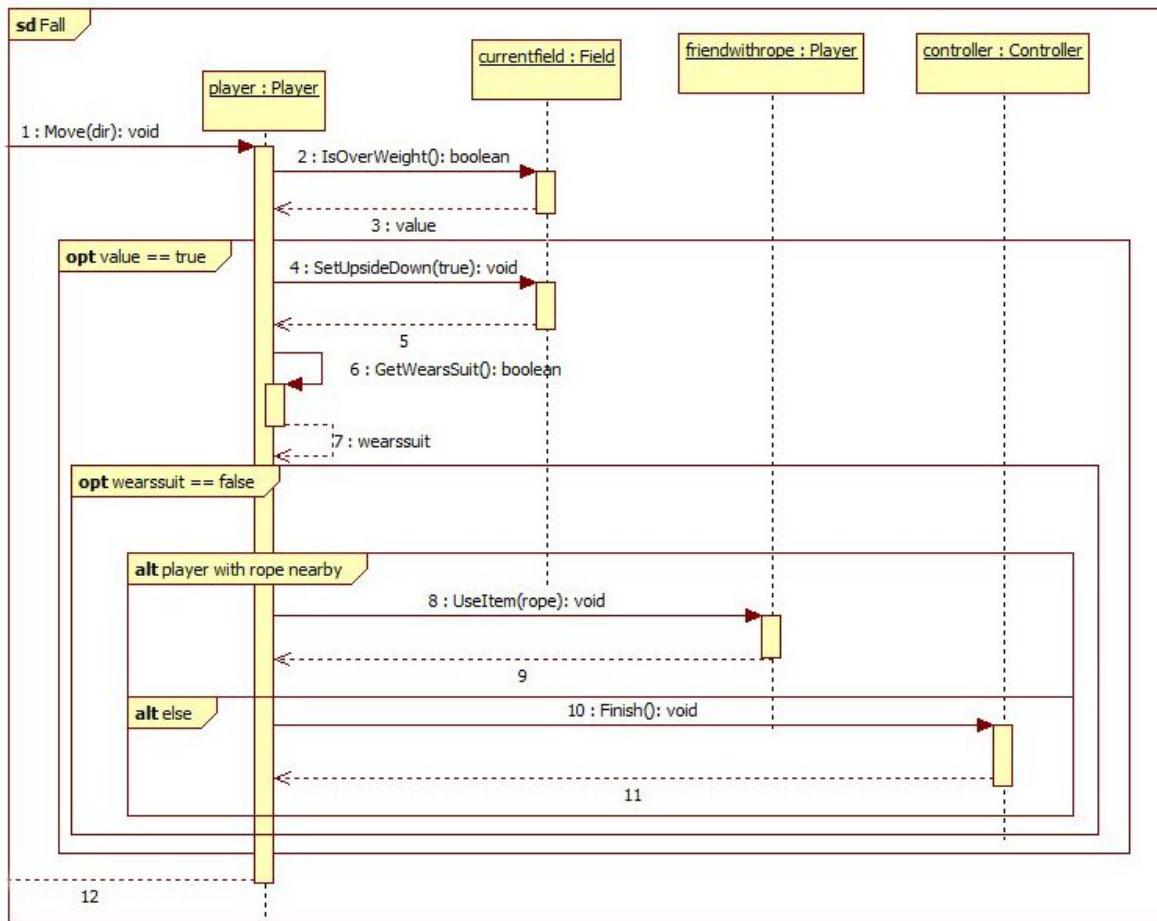
4.4.10 Remove snow with shovel



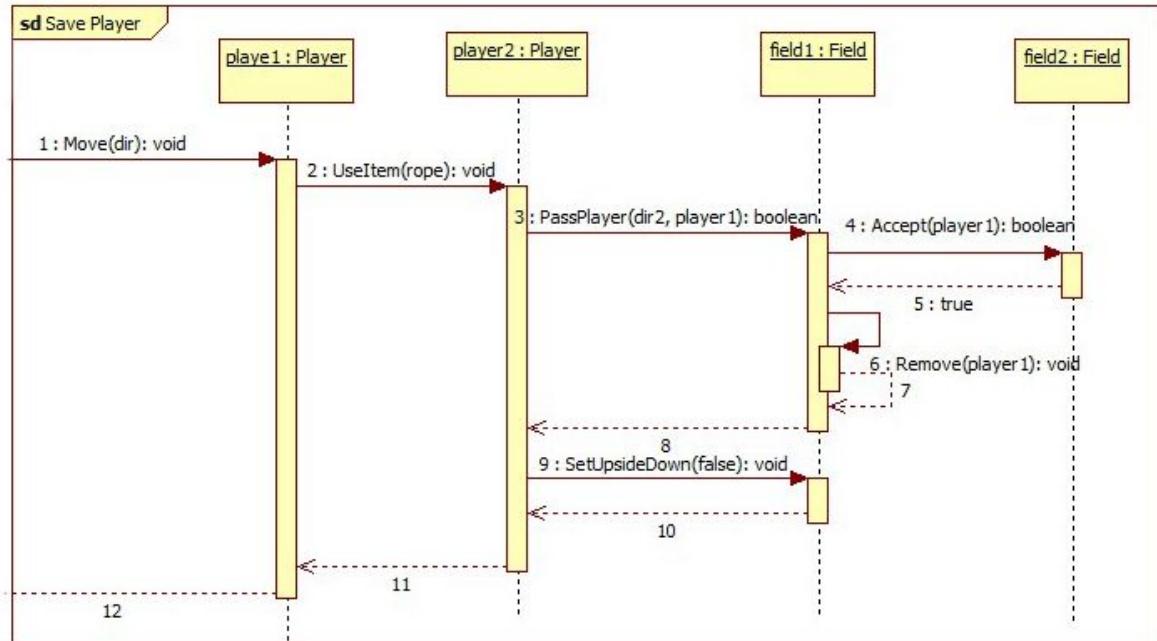
4.4.11 Wear/take off diving suit



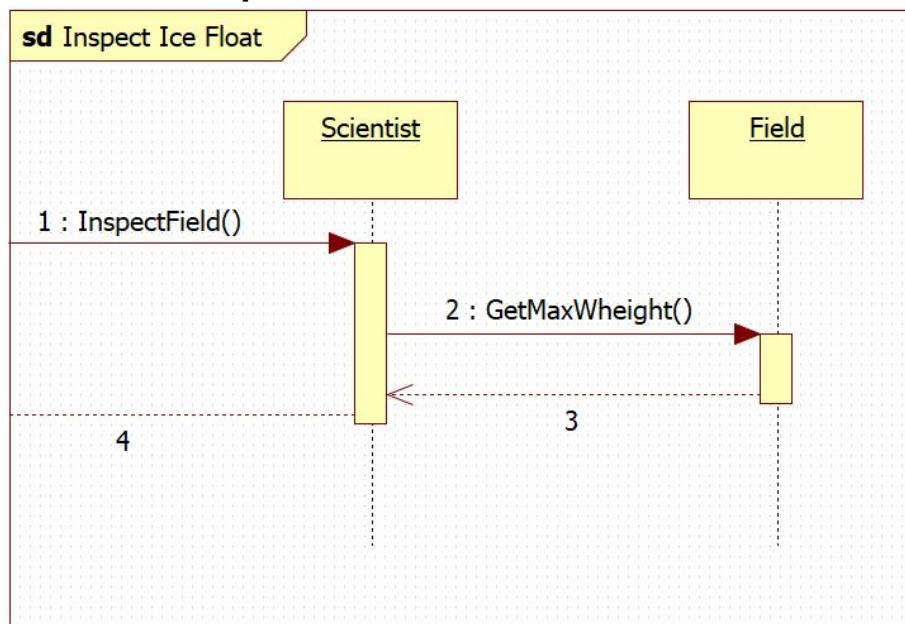
4.4.12 Fall



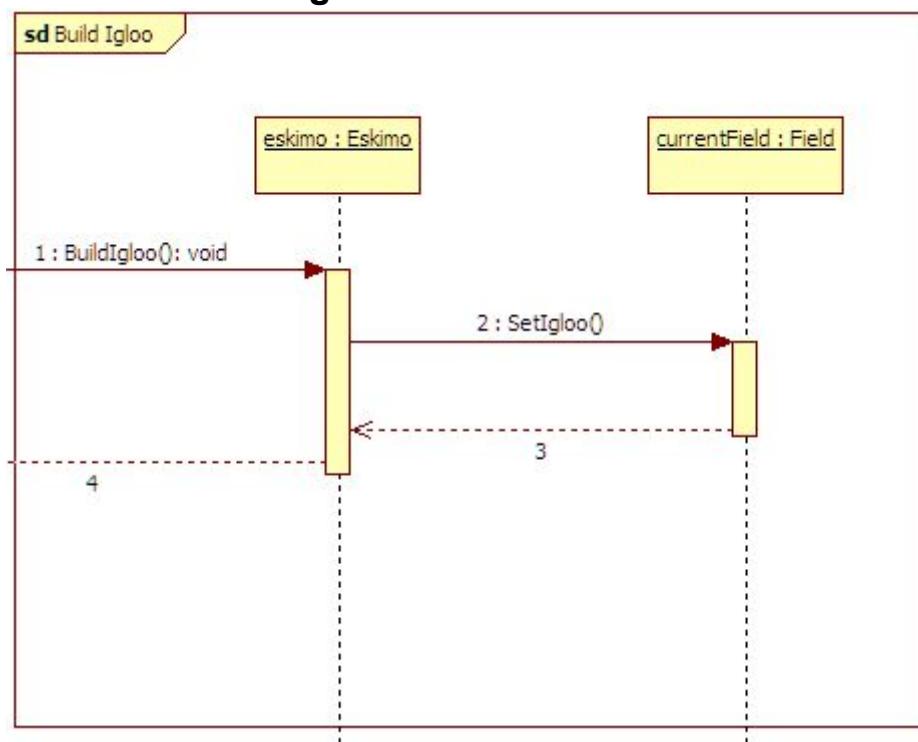
4.4.13 Save player



4.4.14 Inspect ice float

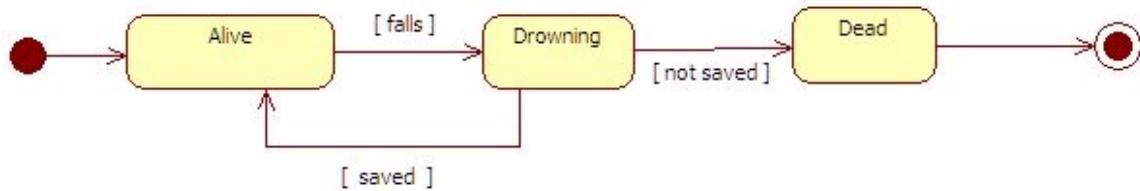


4.4.15 Build igloo



4.5 State-chartok

4.5.1 Player



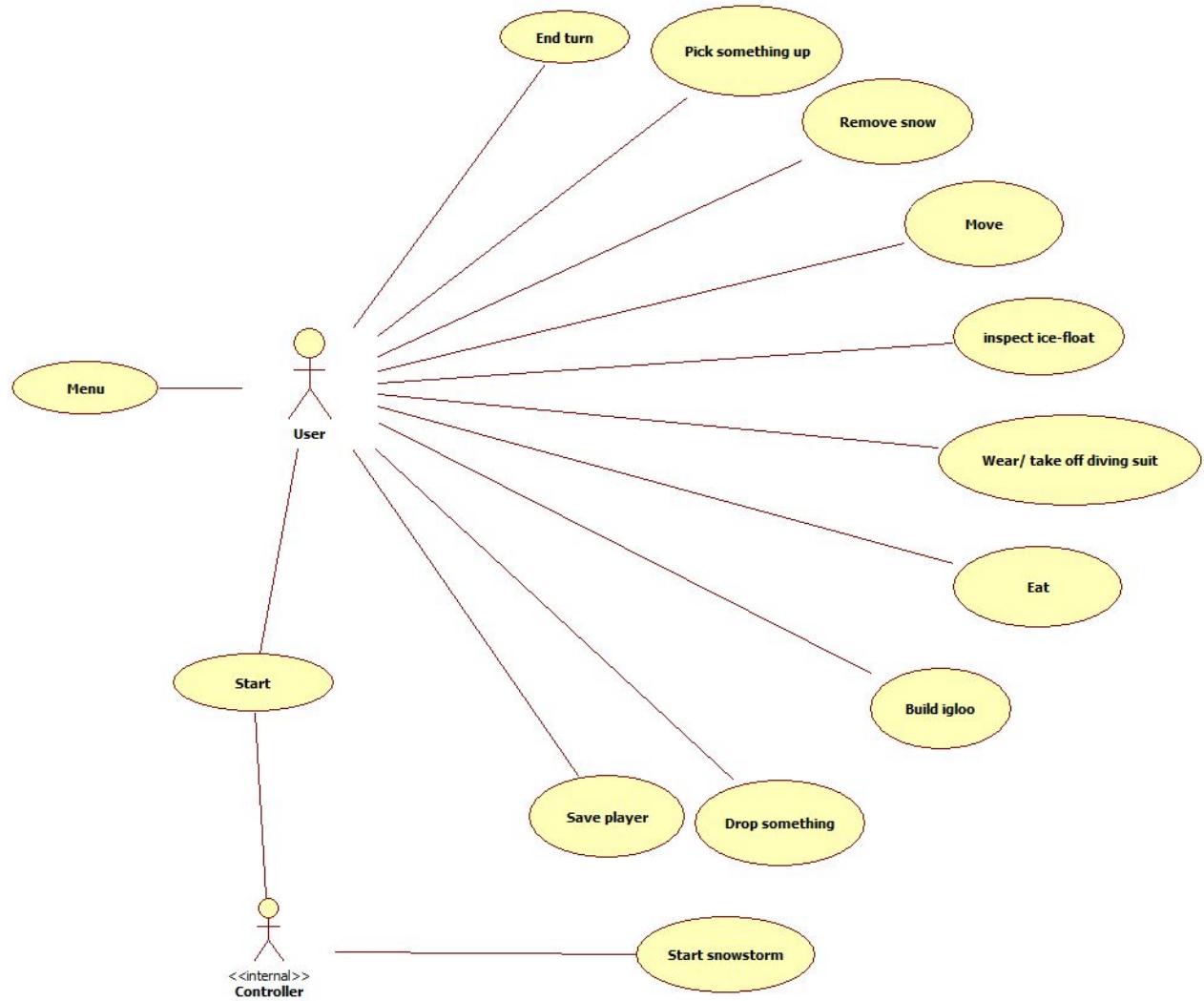
4.6 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.03.05. 18:00	1 óra 30 perc	Filep Jakabos Kálmán Kuzsner Puskás	Értekezés. Döntés: Az Item interfész helyett absztrakt osztály lesz, és ennek megfelelően átírára kerül a dokumentációban az osztályok leírása is. Állapotgép-diagram egyeztetése.
2020.03.05 20:00	1 óra	Kuzsner	Tevékenység: Munkabeosztások táblázatba felvitele, táblázatok létrehozása, elhelyezése, formázása.
2020.03.05. 21:00	30 perc	Jakabos	Tevékenység: Jakabos elkészíti a Player állapotgépet.
2020.03.08. 5:00	1 óra	Kuzsner	Tevékenység: Az Item osztály Interface-ből Absztrakt osztállyá való alakítása, és a változtatások dokumentálása az Osztályok leírásában.

5. Szkeleton tervezése

5.1 A szkeleton modell valóságos use-case-ai

5.1.1 Use-case diagram



5.1.2 Use-case leírások

Use-case neve	Menu
Rövid leírás	A felhasználó a menüben a játékkal, annak kezdésével, és opcionálisan kapcsolatos interakciókat végez.
Aktorok	User
Forgatókönyv	<p>1. Megjelennek a konzolon a választási lehetőségek: <i>New Game, Options, Best Scores, Exit.</i></p> <p>2. A felhasználó kiválasztja az egyik menüpontot, és az ezt lekezelő függvény neve kiíródik a képernyőre.</p> <p>3. Ha a "New Game"-t választja, új játék kezdődik, és az első szereplőnek adódik a vezérlés.</p> <p>3.A.1. Ha az "Options"-t választja, felajánlódnak a beállítási lehetőségek: <i>Difficulty, Player name, Number of players, Music.</i> Azonban ezeket itt még nem lehet beállítani.</p> <p>3.B.1. Ha a "Best Scores"-t választja, akkor az ezt kiolvasó és kiíró logika futna le, de még itt nem lesz megvalósítva.</p> <p>3.C.1. Ha 4-est ír, kilép a játékból, bezáródik a program.</p>

Use-case neve	Move
Rövid leírás	A játékos egyik mezőről a másikra irányítja az egyik szereplőt.
Aktorok	User
Forgatókönyv	<p>1. A szereplő átlépéséhez szükséges függvények nevét a program a szabványos kimenetre írja.</p> <p>1.A.1. Ha a szereplő egy lyuk lépne a vízbeesés folyamatához szükséges függvények kerülnek megjelenítésre.</p> <p>1.A.2. A szereplő olyan jágtáblára érkezik ami átfordul, a vízbeesés folyamatához szükséges függvények kerülnek megjelenítésre.</p> <p>2. A szereplő nem tud mozogni, mert például falba ütközne, az eredeti helyén marad.</p>

Use-case neve	Save Player
Rövid leírás	A játékos megmenthet egy másik, vízbe esett szereplőt ha van kötele.
Aktorok	User
Forgatókönyv	<p>1. A szereplő kimentéséhez szükséges függvények nevét a program a szabványos kimenetre írja.</p>

Use-case neve	Start snowstorm
Rövid leírás	A jégmezőn időnként feltámad a hóvihar, és néhány érintett jegtáblát újabb adag friss hóval borít be. Akit elkap, annak a testhője egységnivel csökken.
Aktorok	Controller
Forgatókönyv	<p>1. A program a kimenetre írja a SnowStorm() függvény lefutása közben hívódó függvények nevét (melyek a szekvencia diagramon láthatóak részletesen).</p>

Use-case neve	Build igloo
Rövid leírás	A eszkimó tud igtul építeni egy egységnyi munka felhasználásával, amiben átvészhetők a hóviharok.
Aktorok	User
Forgatókönyv	<p>1. A program a kimenetre írja a BuildIgloo() függvény lefutása közben hívódó függvények nevét.</p>

Use-case neve	Eat
Rövid leírás	A játékos életerei növelése érdekében egy egységnyi munka árán megeszik egy nála lévő ételt.
Aktorok	User
Forgatókönyv	<p>1. A program a kimenetre írja a UseItem(food) függvény lefutása közben hívódó függvények nevét és a játékos esetlegesen megnövelt életerejét.</p>

Use-case neve	Drop something
Rövid leírás	A játékos egy munka ellenében elhelyez egy nála lévő tárgyat arra a mezőre amin éppen rajta áll.
Aktorok	User
Forgatókönyv	<p>1. A program a kimenetre írja a DropItem(item) függvény lefutása során hívódó függvények nevét, a mezőn lévő összes tárgy nevét, és az elhelyezett tárgy nevét a letevés utáni állapotban.</p>

Use-case neve	Pick something up
Rövid leírás	A játékos felveszi az aktuális mezőn lévő tárgyak közül a legkorábban elhelyezettet egy munka árán.
Aktorok	User
Forgatókönyv	<p>1. A program a kimenetre írja a PickItemUp() függvény lefutása során hívódó függvények nevét, a játékosnál lévő összes tárgy nevét és a megszerzett tárgy nevét a felvétés utáni állapotban.</p>

Use-case neve	Remove snow
Rövid leírás	A játékos havat takarít el az aktuális mezőről. Lapát birtokában kettőt, anélkül egyet.
Aktorok	User
Forgatókönyv	<p>1. A program a kimenetre írja a hó takarításához szükséges függvények nevét.</p> <p>1.A.1. Ha a játékos lapát nélkül szeretne havat takarítani, akkor a program a kimenetre írja a RemoveSnow() függvény lefutása során hívódó függvények nevét.</p> <p>1.A.2. Ha a játékos lapáttal szeretne havat takarítani, akkor a program a kimenetre írja a UseItem(shovel) függvény lefutása során hívódó függvények nevét.</p> <p>2. A program a kiemelte írja, hogy havas-e még az aktuális mező a takarítás után.</p>

Use-case neve	Start
Rövid leírás	Akkor indul el a mikor a játékos a menüben a <i>New Game</i> opciót választja. Elindítja a játék működéséért felelős folyamatokat.
Aktorok	User, Controller
Forgatókönyv	1. A program a kimenetre írja a <i>Start()</i> függvény lefutása közben hívódó függvények nevét.

Use-case neve	Finish
Rövid leírás	A játék befejezésére szolgál.
Aktorok	Controller
Forgatókönyv	1. A program a kimenetre írja a <i>Finish()</i> függvény lefutása közben hívódó függvények nevét.

Use-case neve	Inspect Ice Float
Rövid leírás	A Sarkutató megvizsgál egy adott mezőt, és megállapítja annak teherbírását.
Aktorok	User
Forgatókönyv	1. A program a kimenetre írja az <i>InspectIceFloat()</i> függvény lefutása közben hívódó függvények nevét.

Use-case neve	Wear/take off diving suit
Rövid leírás	A DivingSuit objektum használatával hívódik meg. Ha a szereplőn nincs búvárruha, akkor felveszi a búvárruhát, egyébként pedig leveszi.
Aktorok	User
Forgatókönyv	<p>1. A felhasználó az adott szereplővel fel akarja vetetni a búvárruhát. Amikor ezt közli a programmal, a <i>Player UseItem()</i> függvénye hatására a <i>DivingSuit Use()</i> függvénye fut le, és kiíródik a függvények neve.</p> <p>2. A szereplő rendelkezik búvárruhával az eszköztárában, vagy éppen hord egyet.</p> <p>2.A.1. Az aktuális szereplő nem rendelkezik az eszköztárában, sem saját magán búvárruhával. Ekkor a konzolon erről visszajelzést kapunk, és visszatér a <i>Use()</i> függvény.</p> <p>3. Ha a szereplőn nincs, de az eszköztárában van búvárruha, akkor a szereplőre felkerül a búvárruha, és a konzolon erről visszajelzést kapunk.</p> <p>3.A.1. Ha a szereplőn már volt búvárruha, akkor leveszi a búvárruhát, és bekerül az eszköztárába. Erről is értesül a felhasználó.</p>

Use-case neve	End turn
Rövid leírás	Kör vége.
Aktorok	User
Forgatókönyv	<p>1. Az adott kör véget ér, ekkor a program a kimenetre írja az <i>EndTurn()</i> függvény lefutása közben hívódó függvények nevét.</p>

5.2 A szkeleton kezelői felületének terve, dialógusok

A szkeleton program egy parancssoros program, a felhasználó az interakcióhoz karaktereket ír be, amiket Enterrel zár.

Minden függvény meghívásakor kiíródik a függvény neve és a neki átadott paraméterek.

Menu

Közvetlenül indítás után a menü jelenik meg, kiíródnak a lehetőségek, soronként egy:

1. New Game
2. Options
3. Best Scores
4. Exit

A lehetőségekre a sorszámmal lehet hivatkozni, ezt az egy karaktert kell a kiválasztáshoz beírni.

A "New Game" lehetőséget választva új játék kezdődik.

Az "Options" lehetőség a beállításokhoz vinne, de a szkeleton verzióban ez még nem lesz implementálva, ezért egy "Coming soon" visszajelzést kapunk.

A "Best Scores" menüpont alatt a ranglistát tekinthetnénk meg, de a szkeleton verzióban ez még nem lesz implementálva, ezért egy "Coming soon" visszajelzést kapunk.

Az "Exit" lehetőség hatására kilépünk a játékból, bezáródik a program.

End turn

A játék indításával az első szereplőnek adódik a vezérlés. Ekkor kiíródik a konzolra: "Player N's turn", ahol N a játékos sorszáma.

Ha a felhasználó az adott szereplővel már nem kíván tevékenykedni, akkor a kör végét a konzolra beírt "end turn" karakterezettel lehet jelezni. Ekkor értelemszerűen "Player N+1's turn" íródik a következő sorba.

Move

Mozgatni a soron lévő szereplőt a beírt "up" / "down" / "left" / "right" parancsokkal lehet. Ekkor kiíródik a karakter új koordinátája. A koordináta a képernyő sarkától a mező x és y koordinátája ahol az egység a mező.

Ha egyéb dolgok is történnek mozgás során (pl. a szereplő lyukba esik), akkor ez is kiíródik, ezt lásd alább.

Ha nem sikerült mozogni, mert a pálya szélét elértek, vagy már nincs elég energiánk hozzá, azt is kiírja a program. Ezeket a visszajelzéseket kaphatjuk: "No room to move", "Not enough energy".

Save Player

Ha egy játékos vízbe esik, akkor addig, amíg következőnek sorra nem kerül, kimentheti egy csapattársa. A kimentés lényegében a Rope (kötél) objektum használatát jelenti.

Tehát ha egy olyan szereplő van körön, aki egy vízbe esett szereplő melletti jégtáblán tartózkodik, akkor ő beírhatja, hogy "use rope". Ekkor a vízbe esett szereplő az ő jégtáblájára kerül.

Start snowstorm

A kontroller indíthat hóvihart véletlenszerűen, a körök között. Ekkor kiíródik a képernyőre, hogy "Snowstorm happened", és az érintett mezők koordinátái, soronként egy.

Build igloo

Ha a soron lévő szereplő eszkimó, a felhasználó beírhatja, hogy "build igloo". Ekkor kiíródik a képernyőre, hogy "Igloo built", feltéve hogy a szereplőnek van elég energiája. Ha nincs, akkor ez íródik ki: "Not enough energy".

Eat

A felhasználó beírhatja, hogy "use food". Ekkor a szereplő életpontja növekedik. Az új életpont kiíródik a képernyőre, pl.: "Health = 4".

Drop something

A felhasználó beírhatja, hogy "drop ITEM", ahol az ITEM helyére tetszőleges tárgy nevét beírhatja ezek közül: shovel, food, divingsuit, rope, charge, flare, gun. Pl. "drop rope". Ekkor az adott tárgy a játékos eszköztárából a mezőre kerül.

Ha a szereplónél nincs ilyen tárgy, akkor ez íródik: "player doesn't have this"

Pick something up

Ha a szereplő egy mezőn van, amin látszódik egy vagy több tárgy, akkor beírhatja a felhasználó, hogy "pick up item", ekkor a felhasználó felveszi a mezőn található legelső tárgyat. Ha a mezőn nincs tárgy, akkor visszajelzést kapunk: "No item on field" ..

Ha van a mezőn tárgy, de a szereplő eszköztára tele van akkor ez íródik ki: "player's inventory is full".

Ha van tárgy a mezőn, és bele is fér a szereplő eszköztárába, akkor ez íródik ki: "Successfully picked ITEM up", ahol az ITEM az előbbiekben felsorolt tárgy lehet.

Remove snow

A felhasználó beírhatja, hogy "remove snow". Ennek hatására kiíródik, hogy egy egység hó el lett takarítva: "1 snow removed".

Ha a szereplőnél van ásó, akkor ezt is lehet használni: "use shovel". Ennek hatására 2 egység hó takarítódik el, ekkor kiíródik: "2 snow removed".

Ha nem volt hó a mezőn, akkor ezeknek nincs hatása, ekkor a kiírt üzenet: "0 snow removed".

Start

A játék kezdetekor kiíródnak soronként minden mező ismert adatai: milyen játékosok vannak rajta, látszik-e rajta tárgy, van-e rajta iglu (iglu a játék elején nyilván nem lesz, mert azt a szereplők hozhatják csak létre).

Valamint az első soron lévő szereplő neve is megjelenik.

Finish

Ha a szereplők összeszerelték és elsütötték a rakétát, akkor megnyerik a játékot, és a konzolon megjelenik, hogy "Game Won!". Ezután a menübe k erülünk vissza.

Inspect IceFloat

Ha éppen Scientist van soron, a felhasználó beírhatja, hogy "inspect icefloat". Ekkor a mellette lévő mezőknek kiíródik a teherbíráruk

Soronként pl.: Up: 2, Down: 5, Right: 3, Left: 4.

Wear/take off DivingSuit

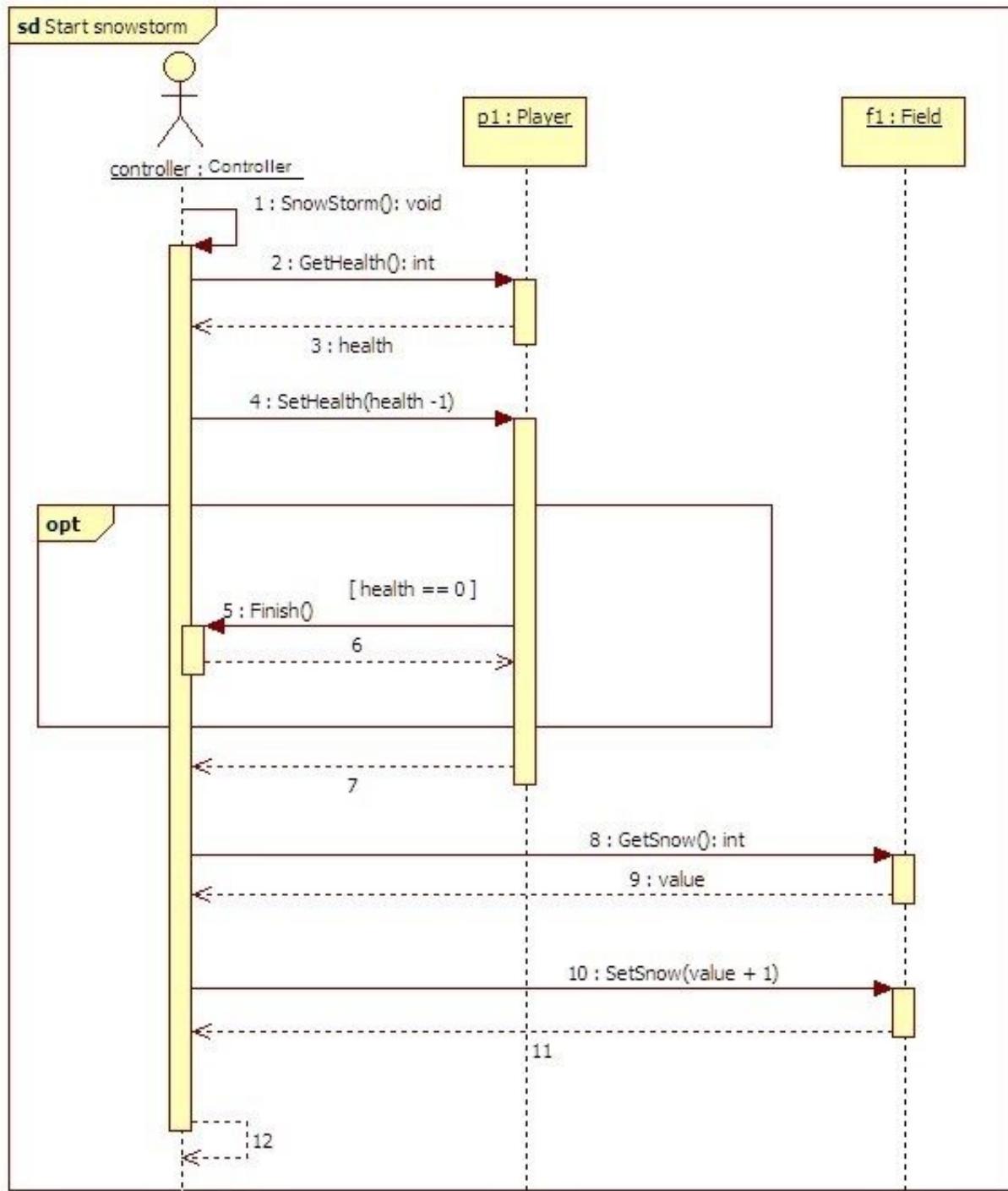
Ha a felhasználó beírja a konzolra, hogy "use divingsuit", akkor a következők történhetnek:
Az aktuális szereplő nem rendelkezik az eszköztárában, sem saját magán búvárruhával. Ekkor a konzolra ez íródik ki: „player doesn't have divingsuit”.

Ha a szereplőn nincs, de az eszköztárában van búvárruha, akkor a szereplőre felkerül a búvárruha, és a konzolra ez íródik ki: „player now wears divingsuit”.

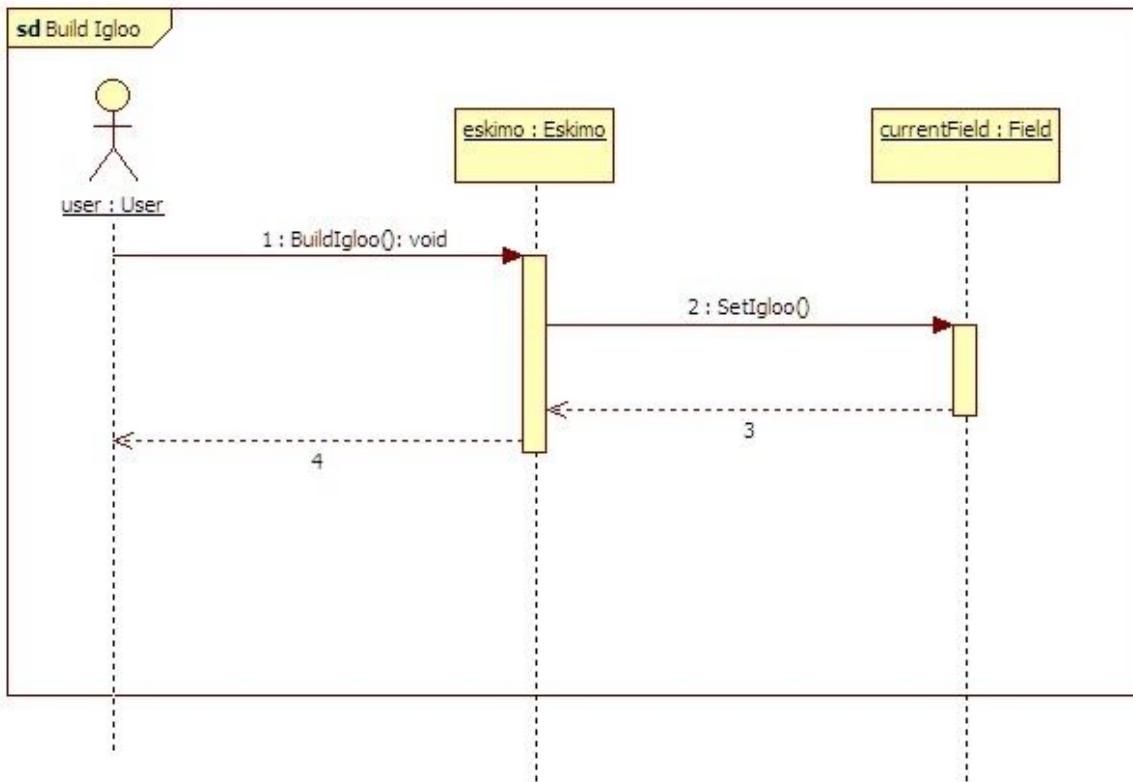
Ha a szereplőn már volt búvárruha, akkor leveszi a búvárruhát, és bekerül az eszköztárába. A konzolra ez íródik ki: „player took off divingsuit”.

5.3 Szekvencia diagramok a belső működésre

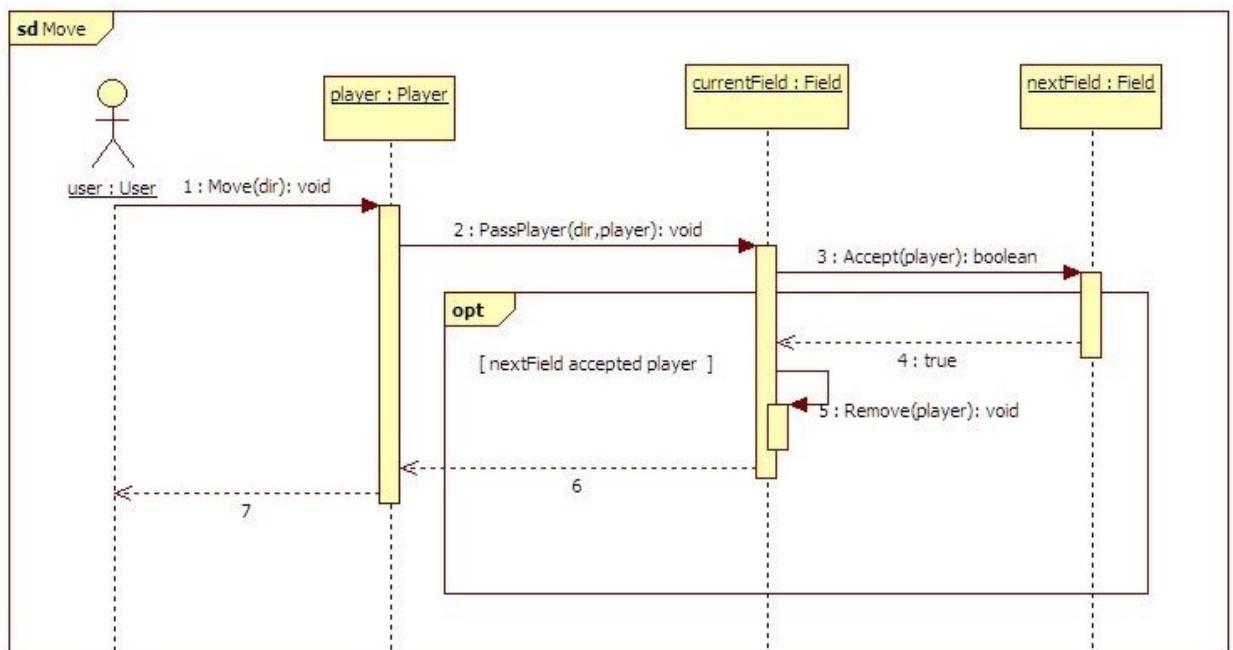
Start snowstorm

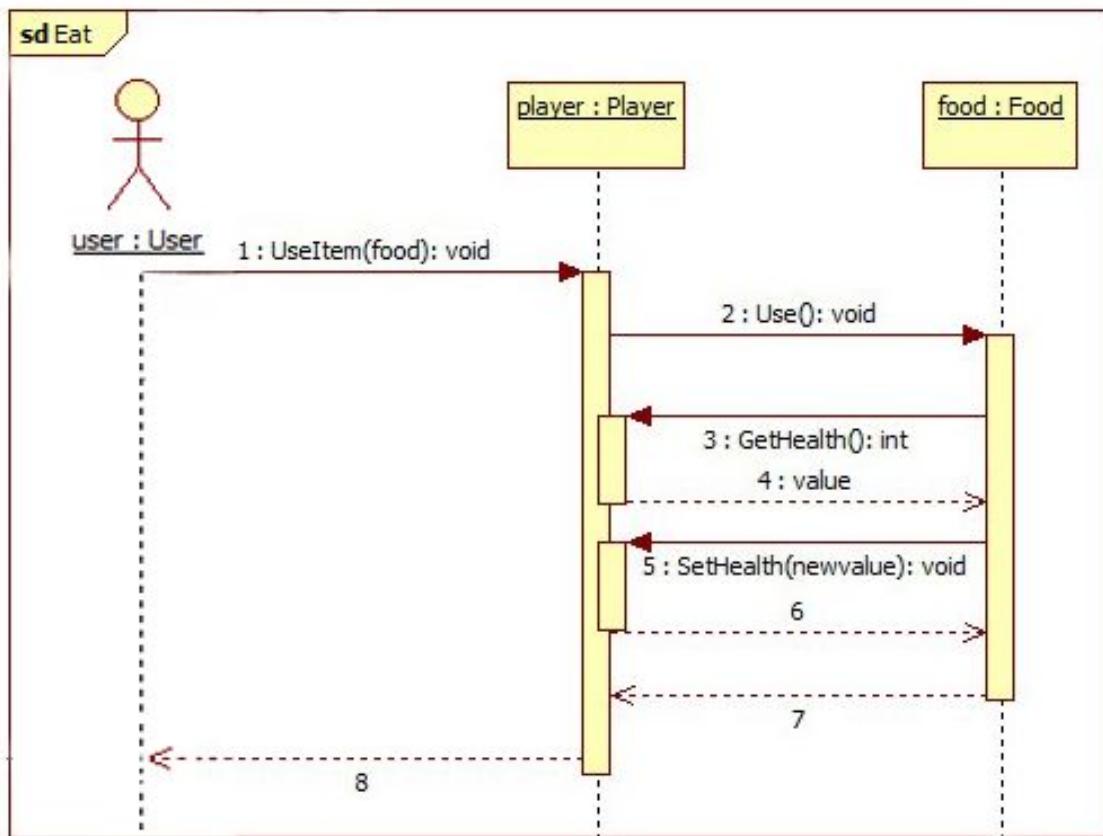


Build Igloo

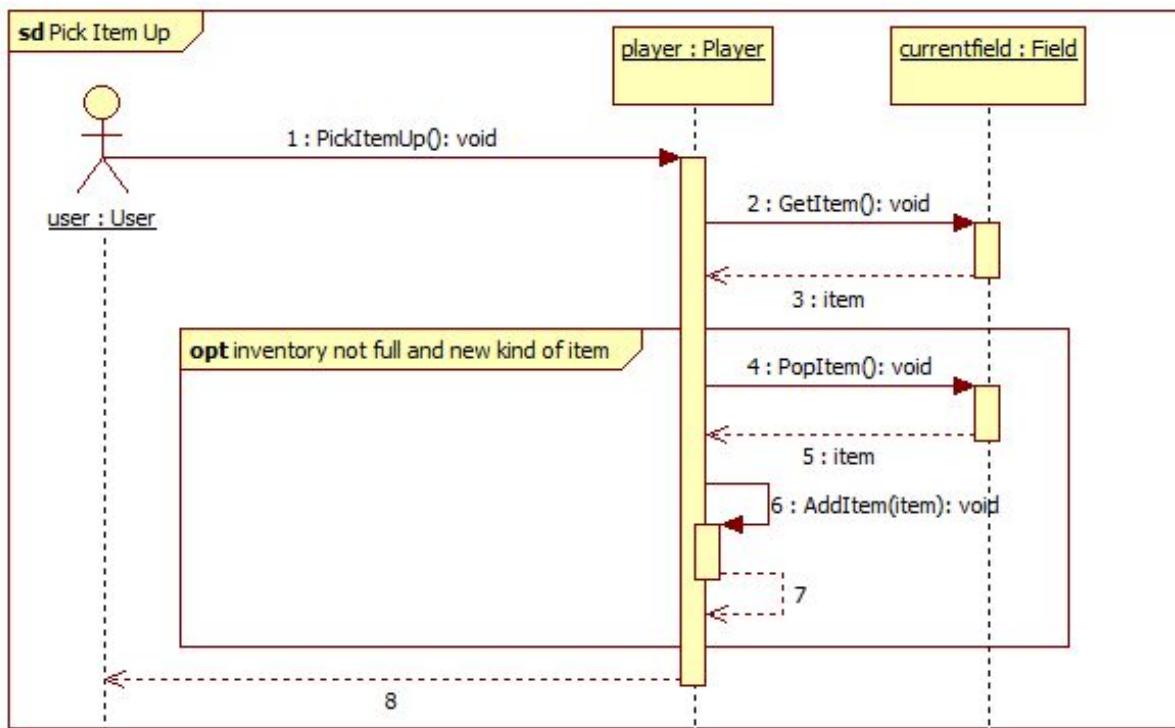


Move

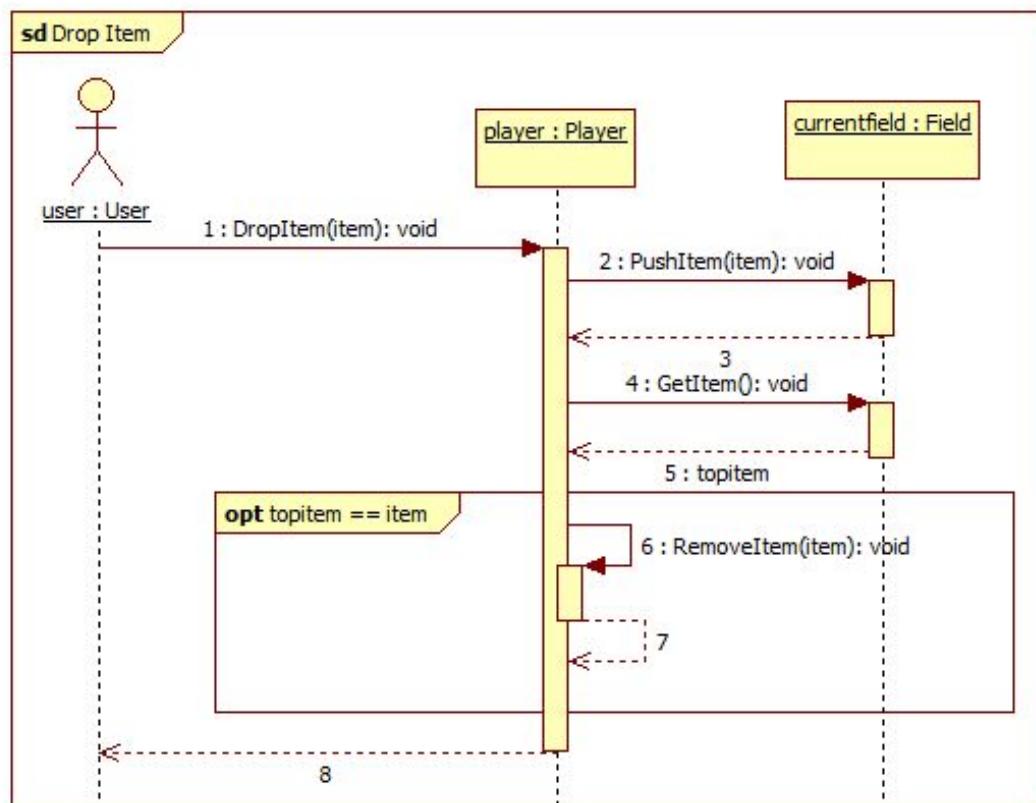


Eat

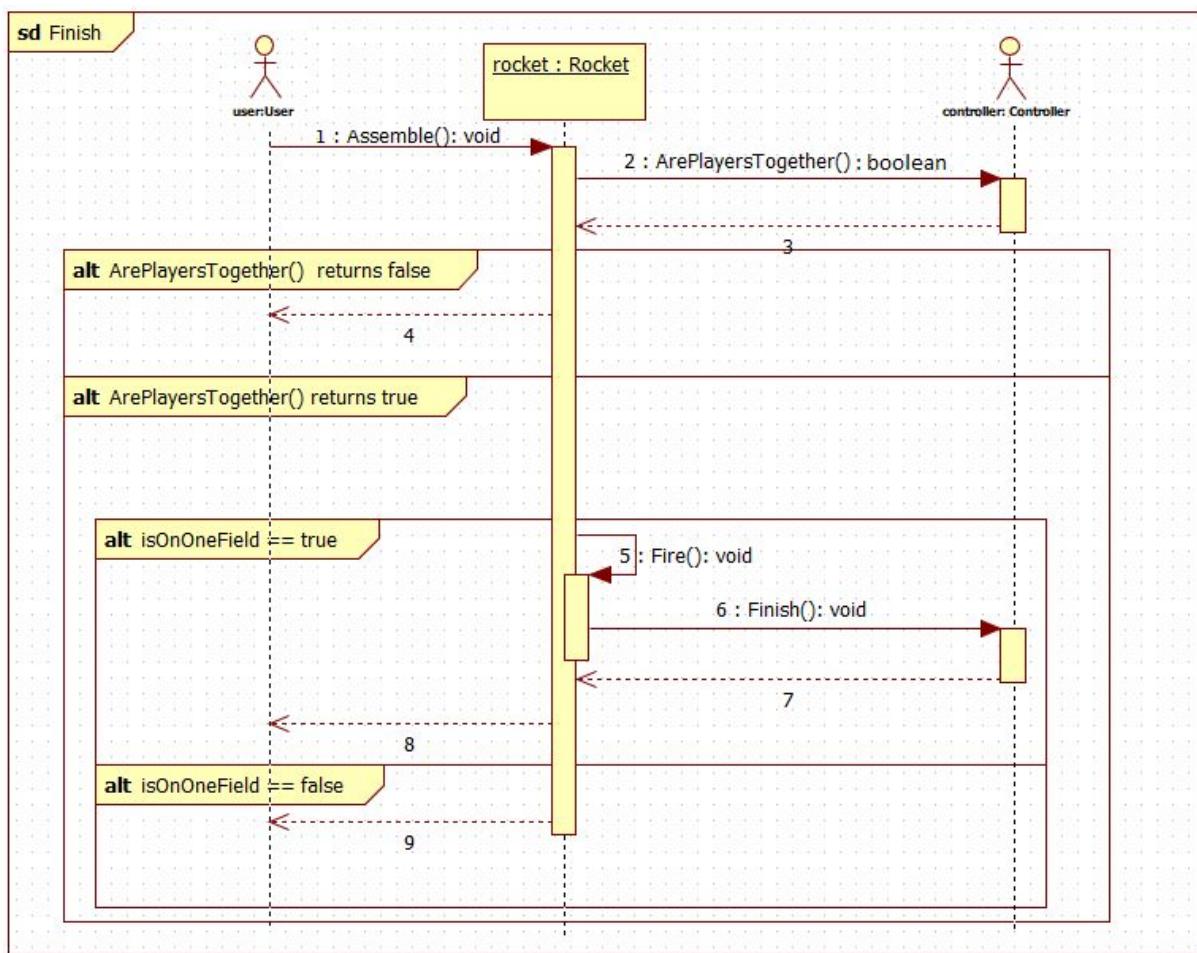
Pick item up



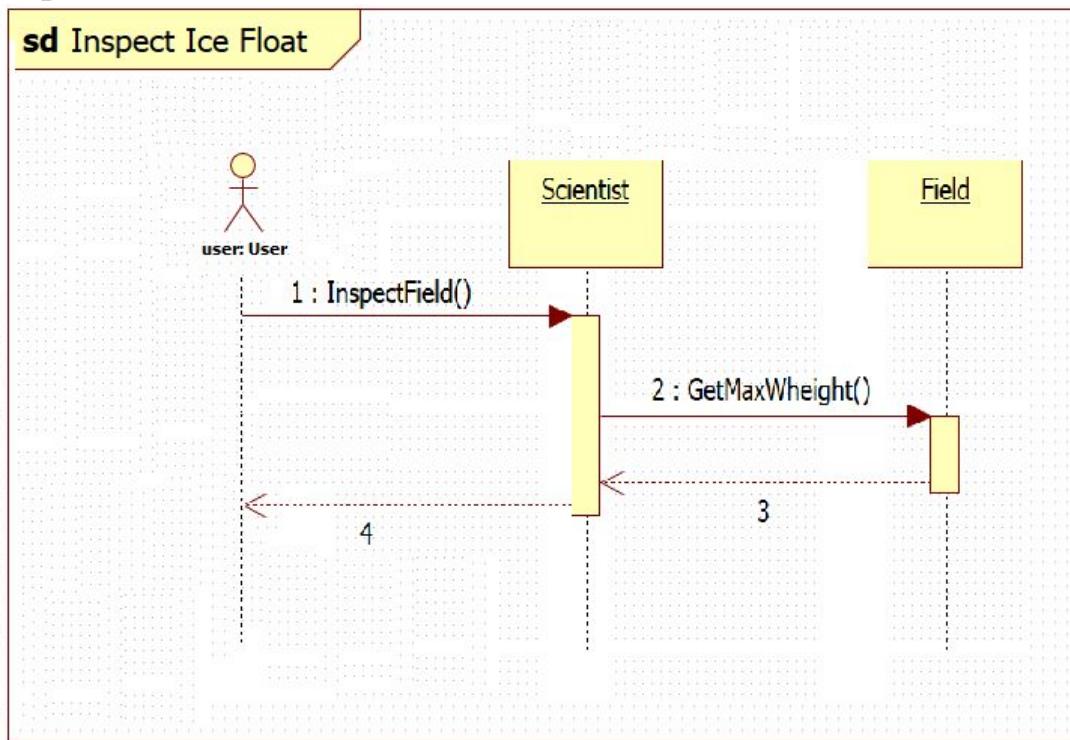
Drop item



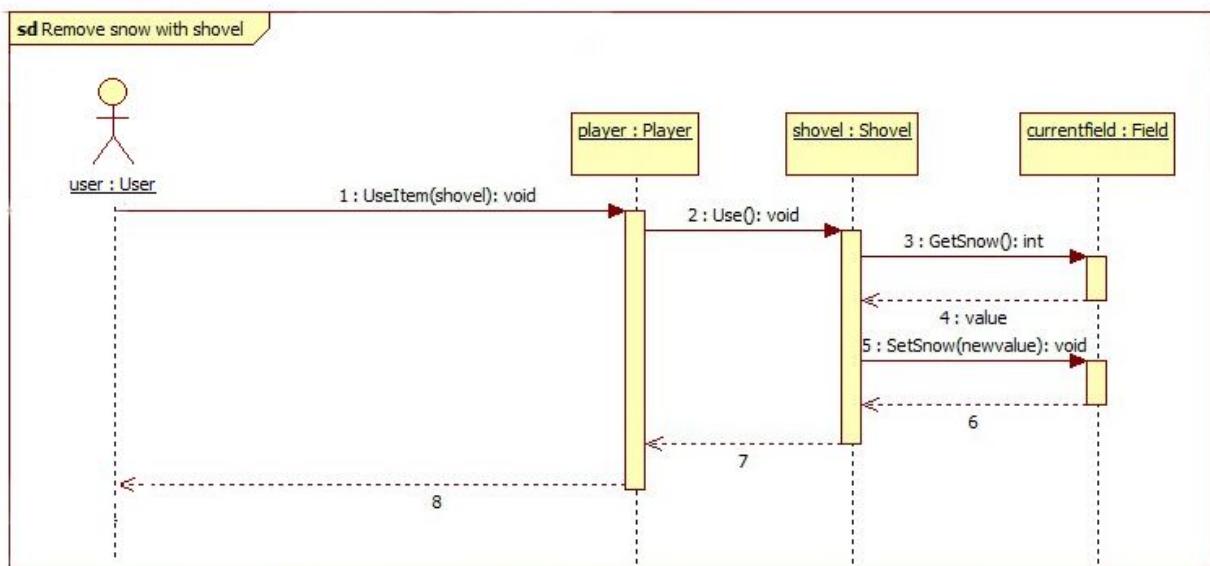
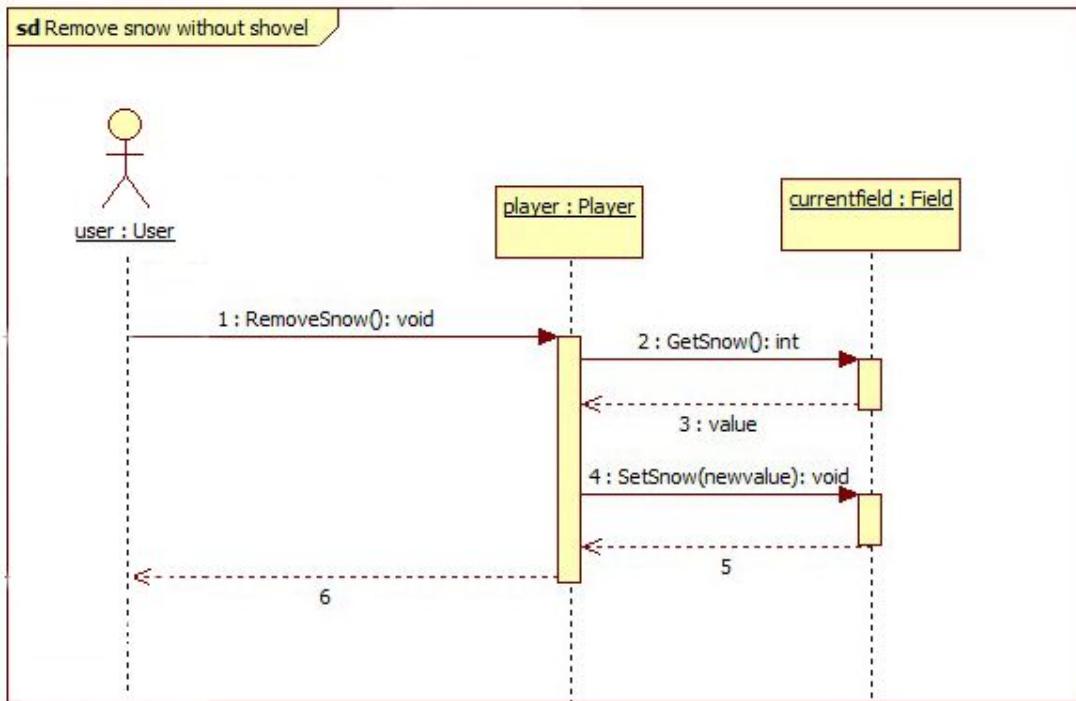
Finish

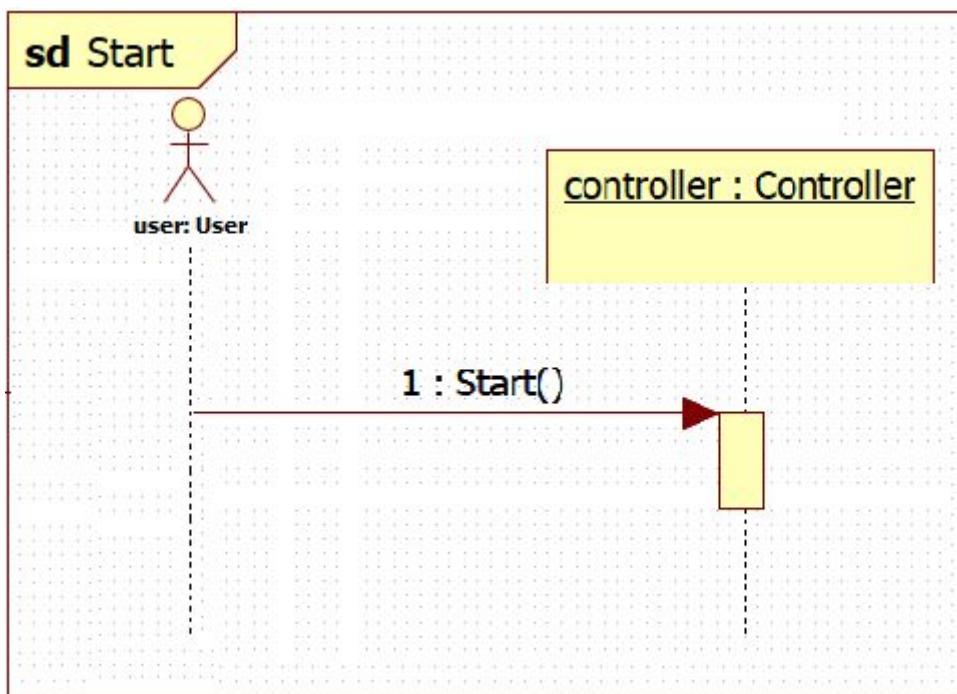


Inspect Ice Float



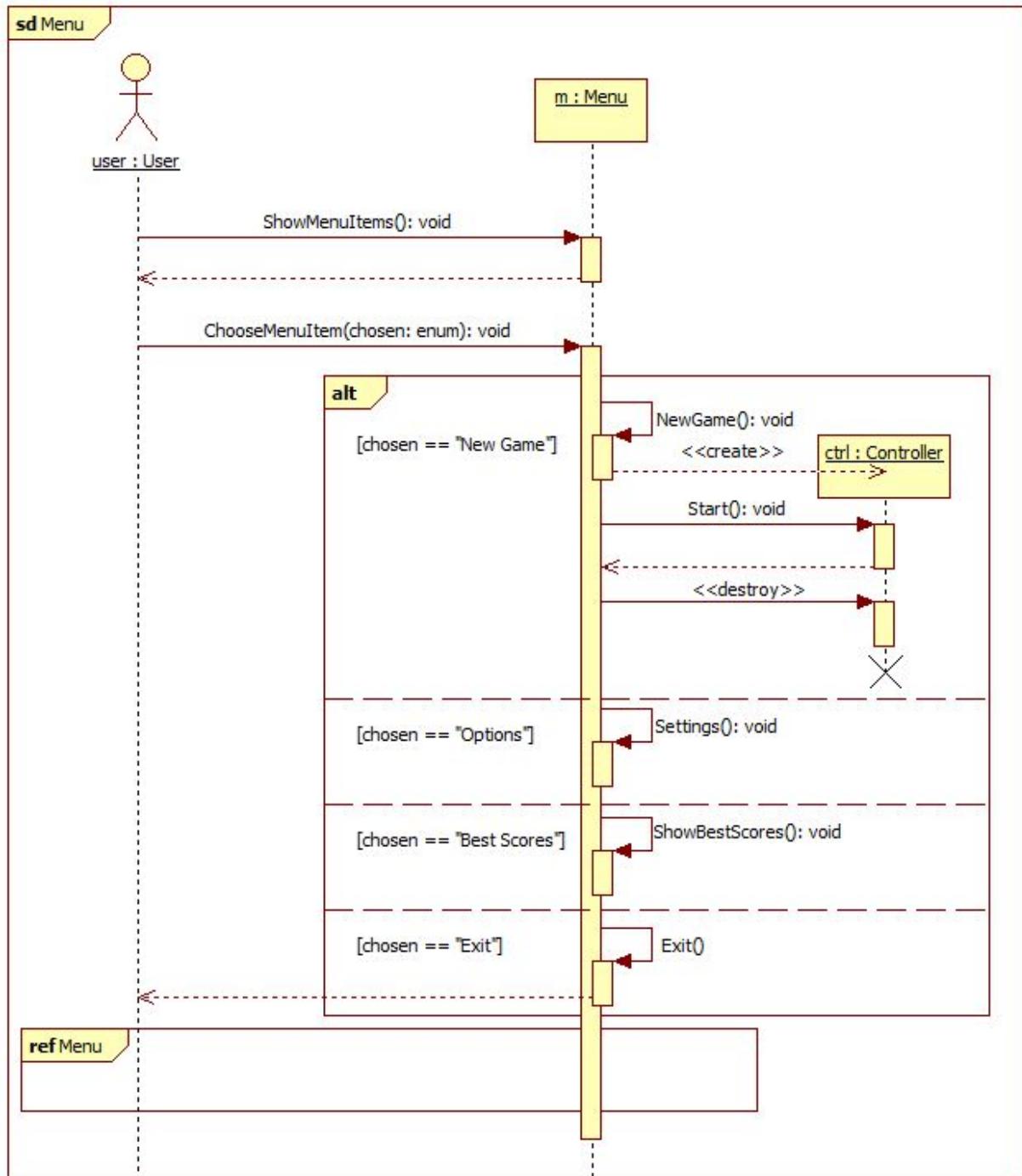
Remove snow



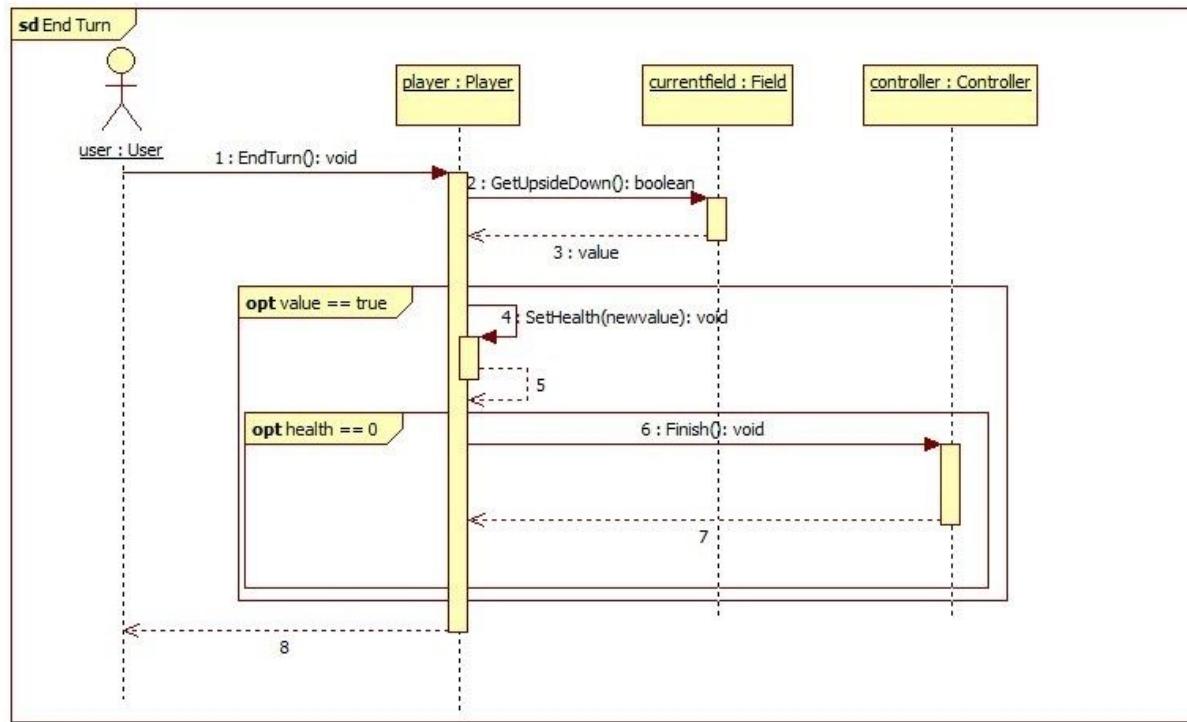
Start

Menu

Megjegyzés: A Controller Start() függvénye inicializálja az objektumot, és kezdi a játékot. Akkor tér vissza a Start() függvény, ha meghívják a Controller-en a Finish() függvényt. Ha visszatért a Start(), akkor a Menu tudja, hogy végetért a játék, és megszüntetheti a Controller-t. A menüpont kiválasztása után rekurzívan erre a diagramra van referencia, mert ugyanez fog lefutni, tehát pl. ha megnéztük a ranglistát, a menübe kerülünk vissza, és minden oda kerülünk vissza, amíg ki nem lépünk a programból.

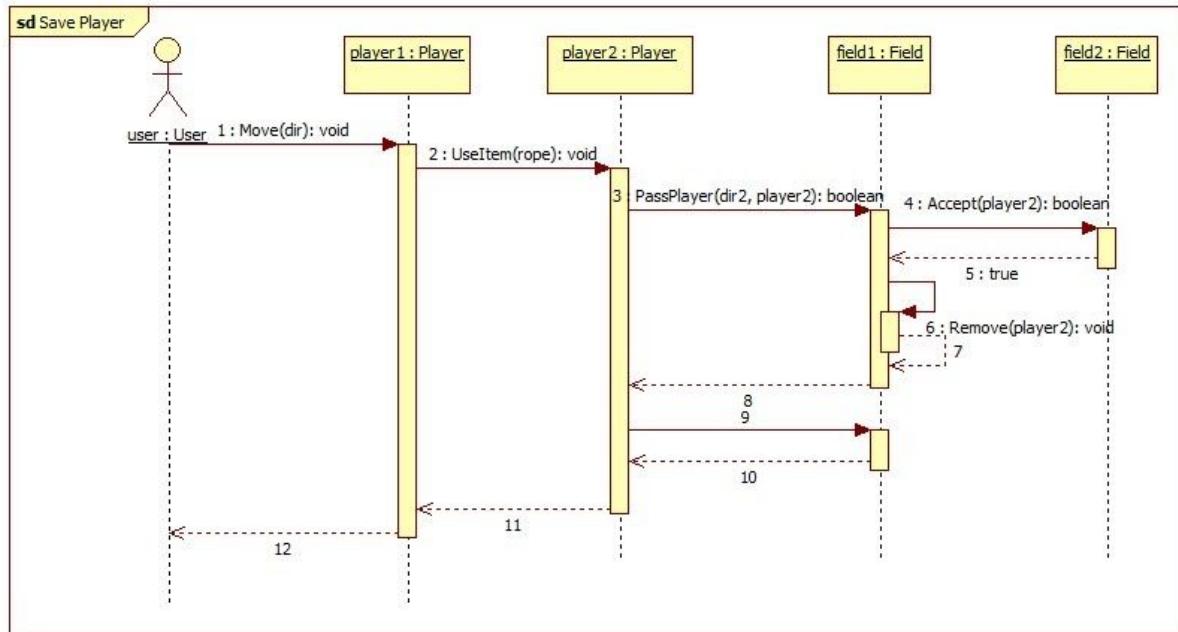


End turn



Megj.: Ebben az esetben a Player szerepét felveheti egyaránt az Eskimo és a Scientist is.

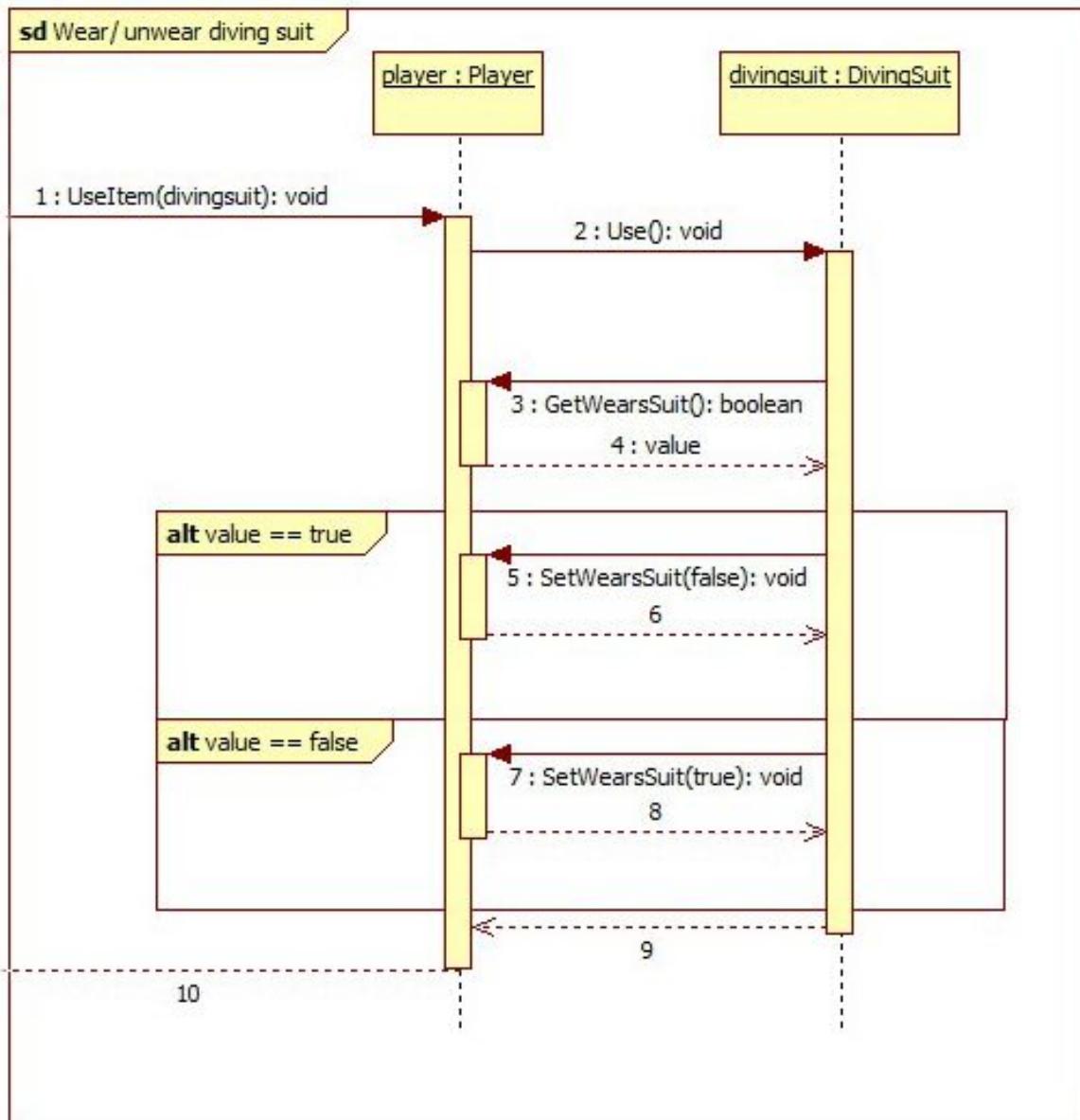
Save Player



Megj.: Ebben az esetben a két Player szerepét bármilyen kombinációban felveheti egyaránt az Eskimo és a Scientist is.

Wear/ take off DivingSuit

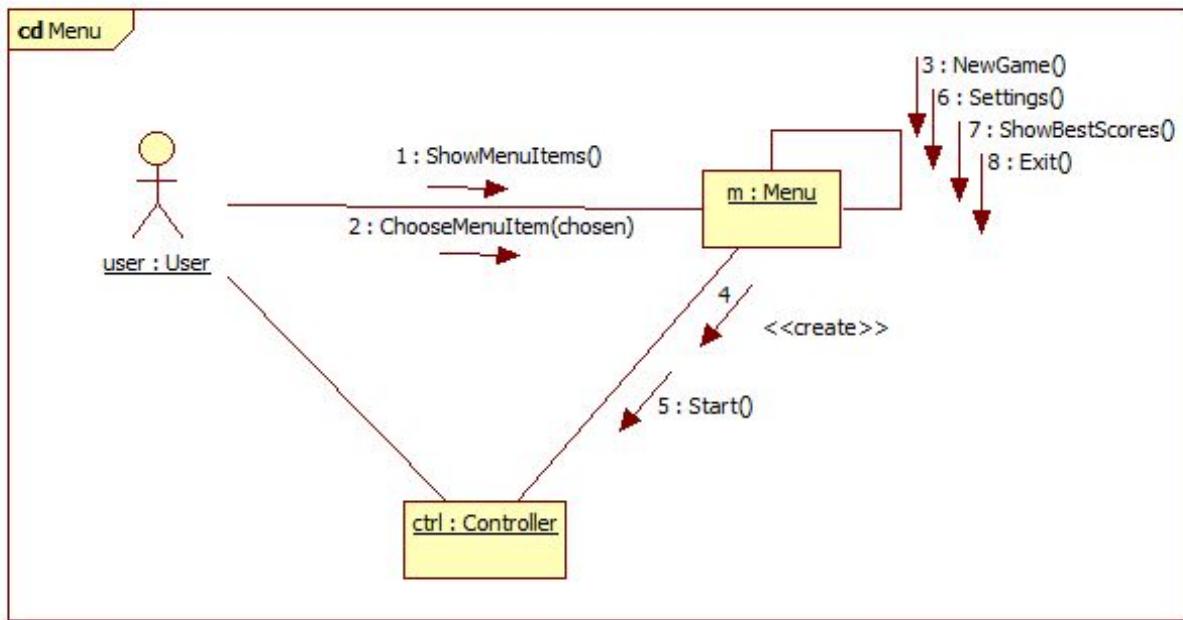
Megjegyzés: A Player lehet Eskimo és Scientist is, pontosan ugyanúgy fog lefutni.



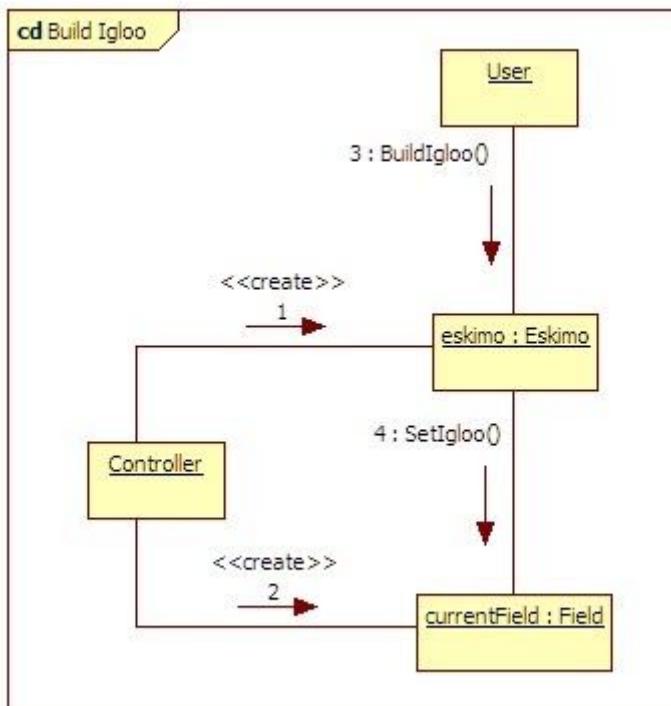
5.4 Kommunikációs diagramok

Menu

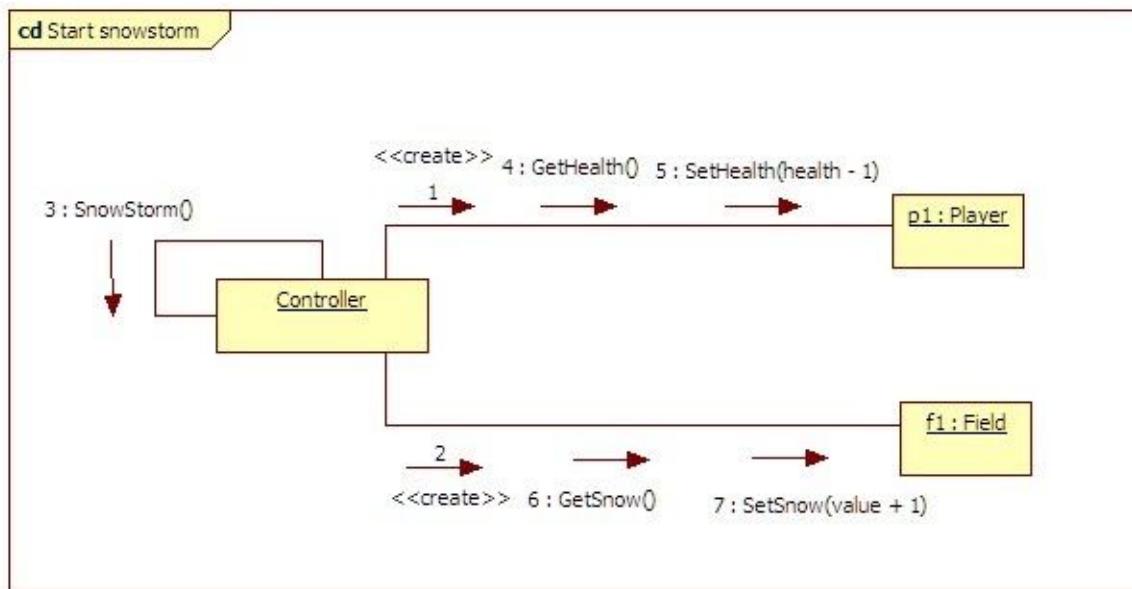
Megjegyzés: Itt az aktort is feltüntettem, mert egyes függvényeket közvetlenül ő hív meg. A Menu közvetlenül a program indítása után jön létre, tehát nem objektum, hanem közvetlenül a Main statikus osztály hozza létre, ez a diagramon nincs jelölve.



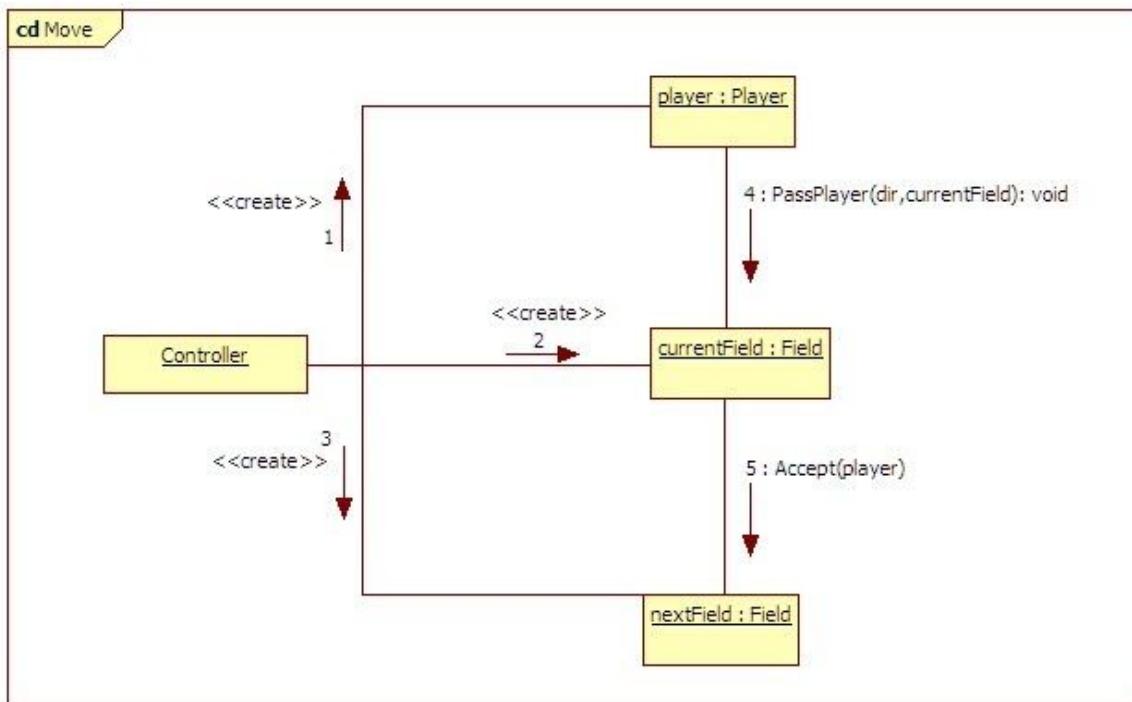
Build Igloo



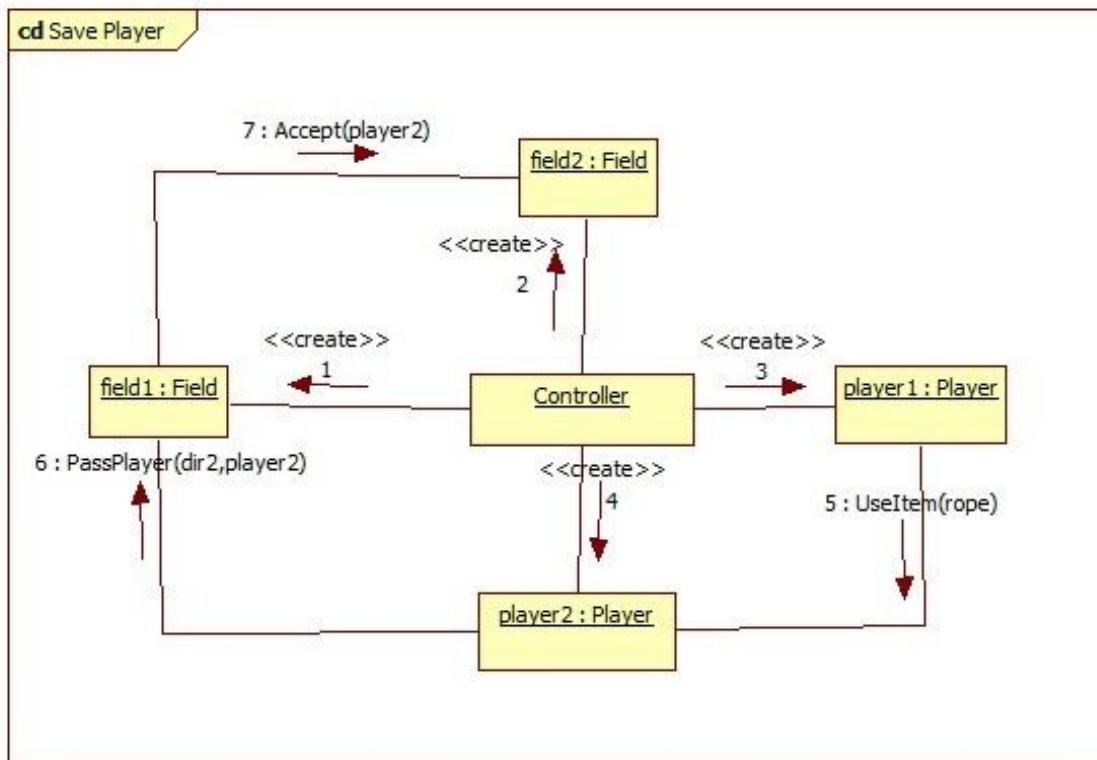
Start snowstorm



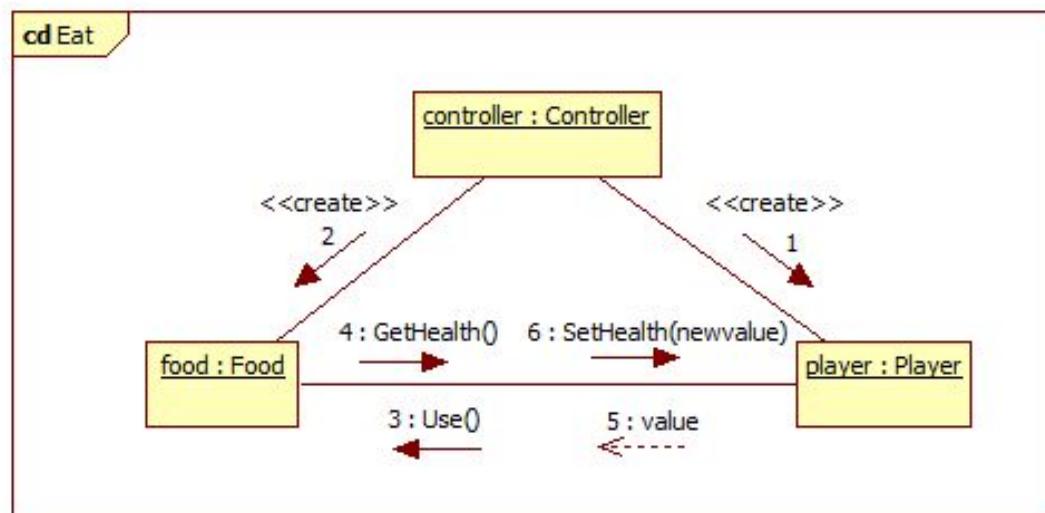
Move



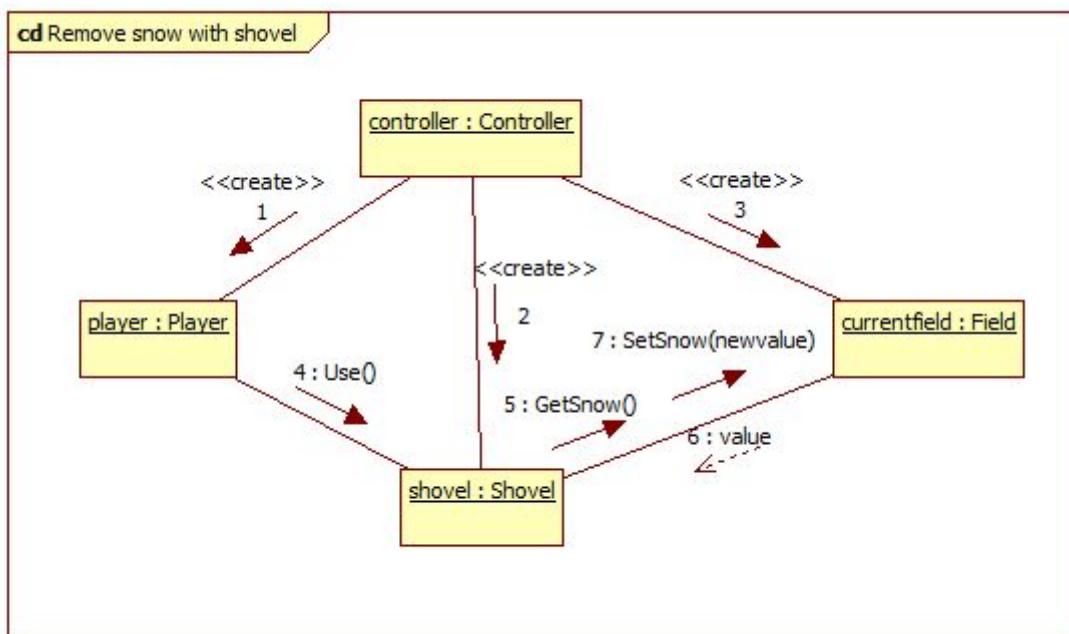
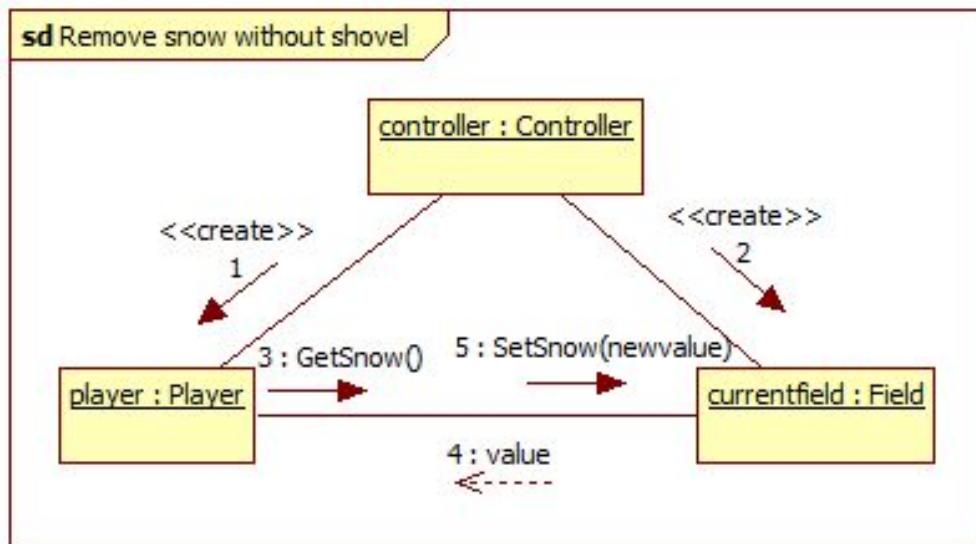
Save Player



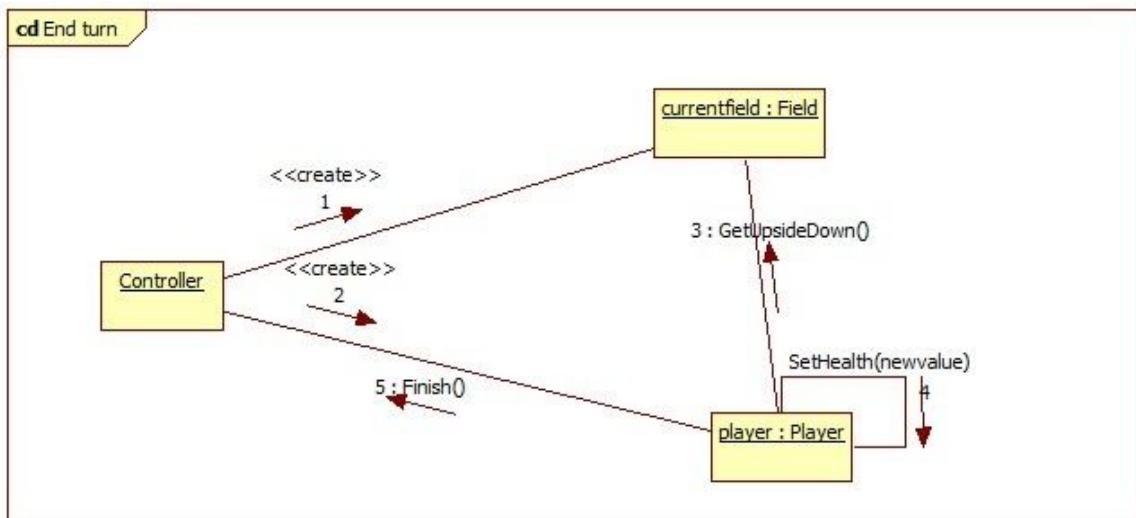
Eat



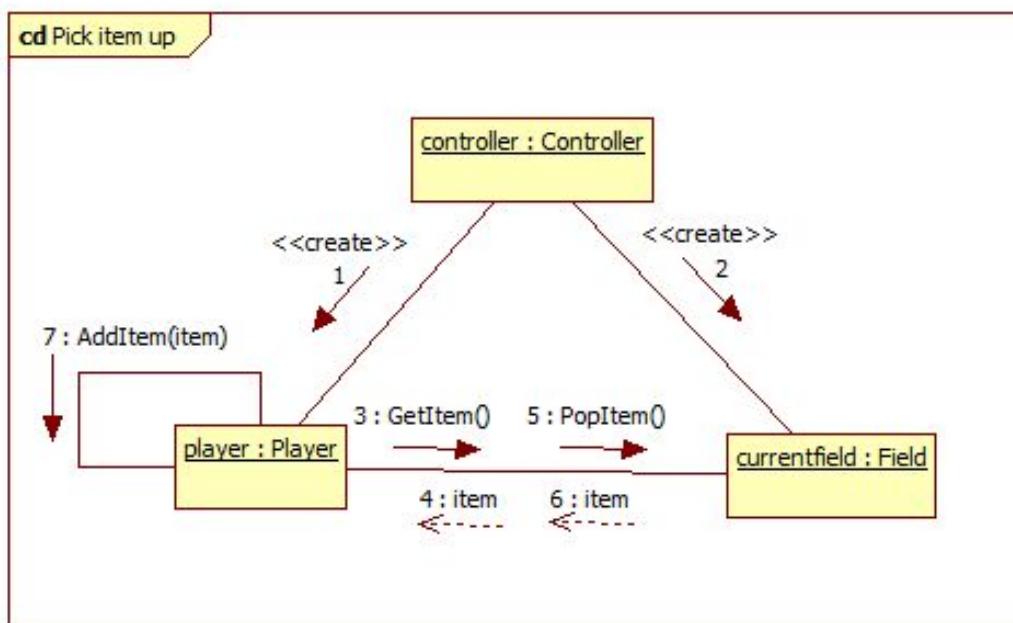
Remove snow



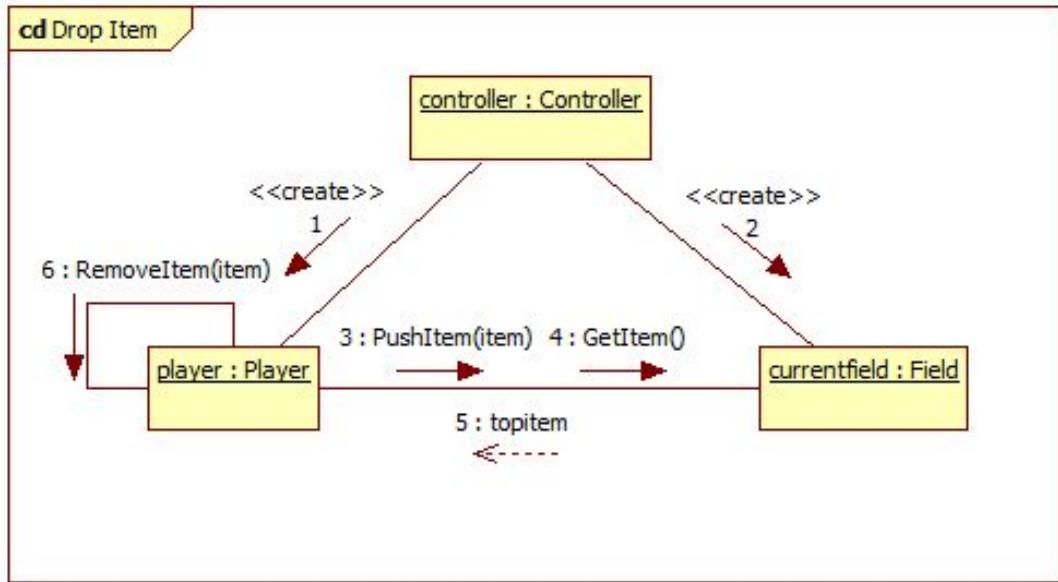
End turn



Pick something up

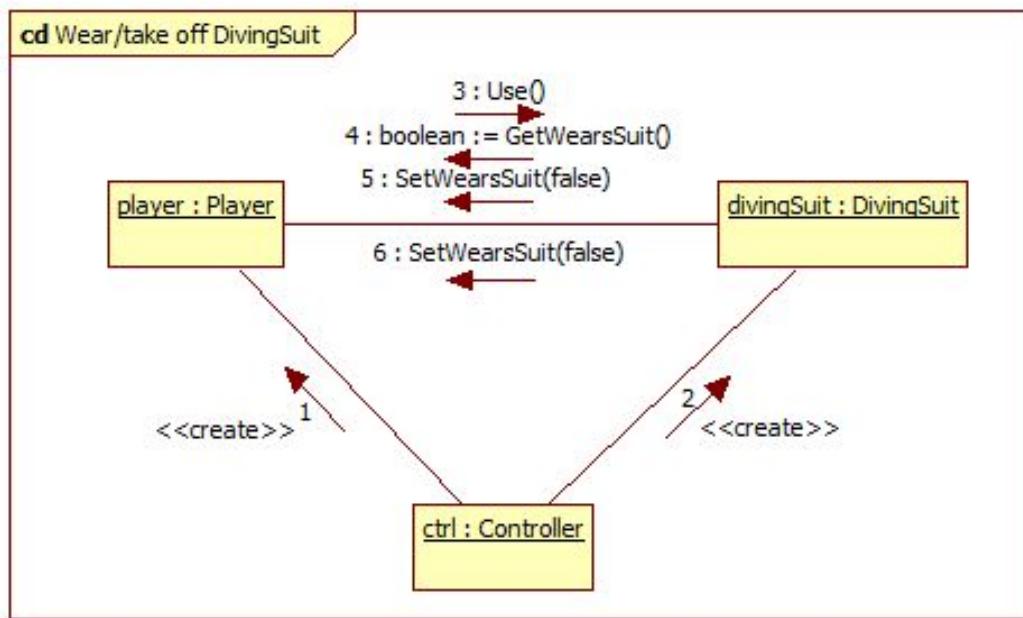


Drop something

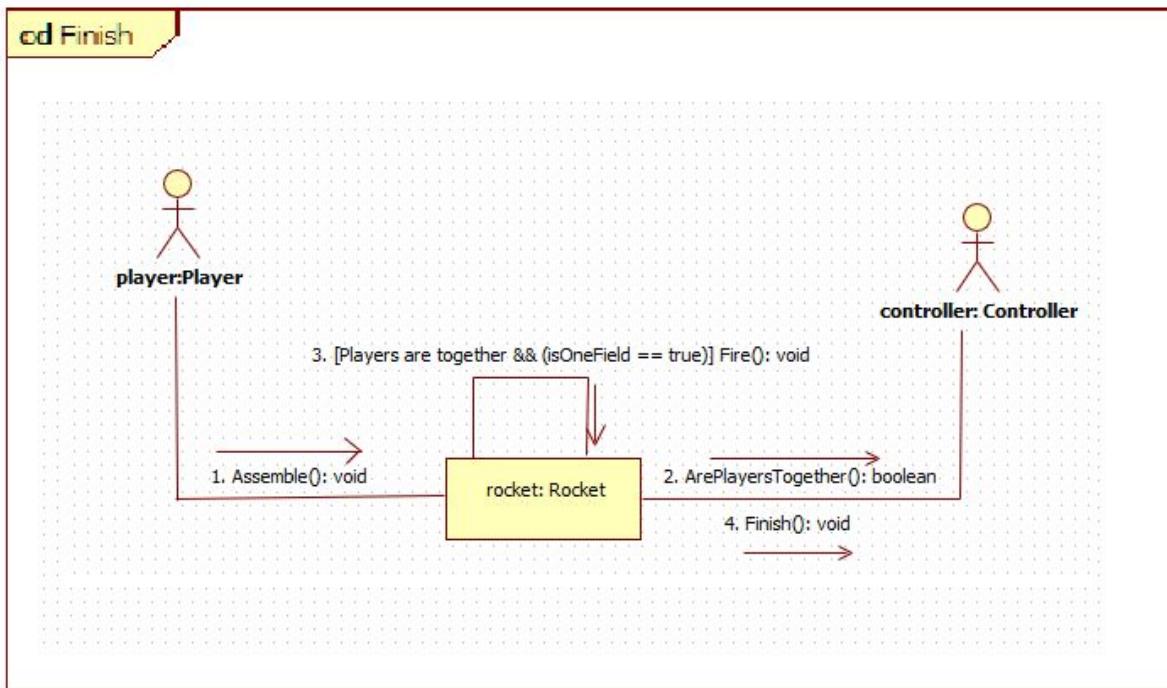


Wear/ take off DivingSuit

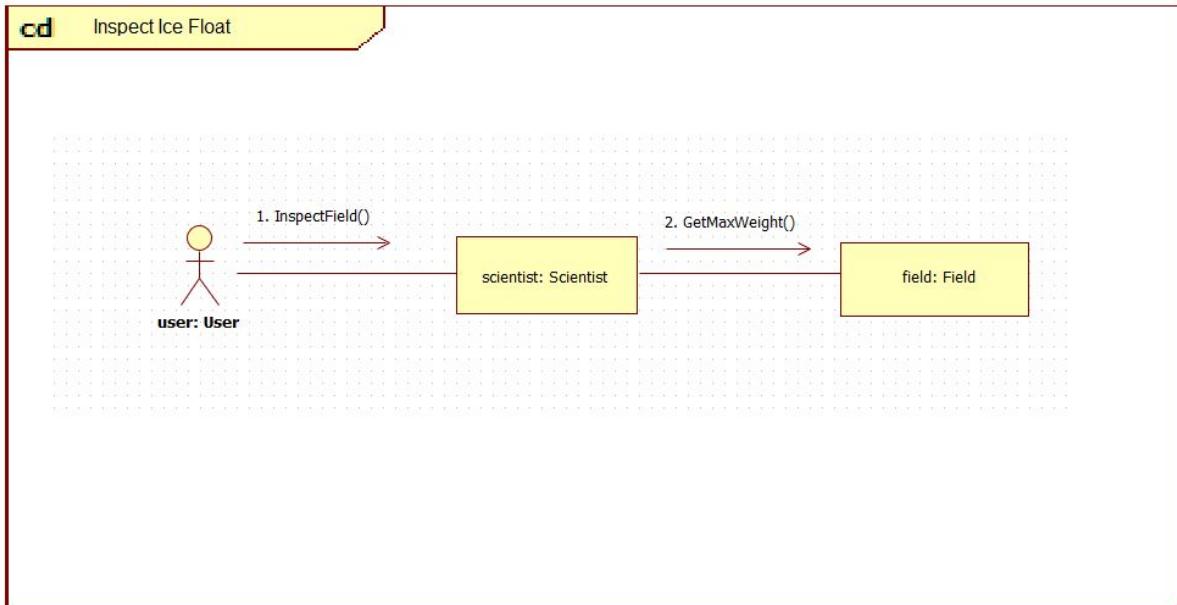
Megjegyzés: A Player lehet Eskimo és Scientist is, pontosan ugyanúgy fog lefutni.



Finish



Inspect Ice Float



Start

A Start Game implicit szerepel van a többi diagramon, ezért nem ábrázoljuk külön, mert így átláthatóbb a folyamat.

5.5 Napló

Kezdet	Időtartam	Résznevők	Leírás
2020.03.09. 17:00	2 óra	Filep Jakabos Kálmán Kuzsner Puskás	Értekezés: Heti feladatok megbeszélése, szkeleton-programra vonatkozó eddigi és új Use-Case-k megvitatása, szétoztatása a csapattagok között.
2020.03.10. 16:00	1 óra	Kálmán	Tevékenység: Kálmán elkészíti a Use-Case diagram szkeleton-programra specifikált változatát.
2020.03.20. 19:00	1 óra	Jakabos	Tevékenység: Move, Start snowstorm, Build igloo use case leírások elkészítése.
2020.03.19. 17:00	3 óra	Kálmán	Tevékenység: Kálmán elkészíti a Menu use-case leírását, szekvencia- és kommunikációs diagramját.
2020.03.21. 14:00	1 óra	Filep	Tevékenység: Filep elkészíti az Eat, Drop something, Pick something up, Remove snow use case leírásokat.
2020.03.21. 15:00	2 óra	Jakabos	Tevékenység: Move, Start snowstorm, Build igloo szekvenciadiagramok és kommunikációs diagramok elkészítése.
2020.03.21. 19:00	3 óra	Filep	Tevékenység: Eat, Pick item up, Drop item, Remove snow with/without shovel szekvenciadiagramok és kommunikációs diagramok elkészítése.

2020.03.22. 12:00	2 óra	Kálmán	Tevékenység: Menu Use-Case leírásának, szekvenciadiagramjának és kommunikációs diagramjának módosítása.
2020.03.21 11:00	1 óra	Puskás	Tevékenység: Save Player Use-Case leírásának, szekvenciadiagramjának elkészítése.
2020.03.21: 13:00	2 óra	Kuzsner	Tevékenység: Diagram átnézése, Start, Finish, Inspect Ice Float Use-casek megírása és szekvencia diagramok megrajzolása
2020.03.21: 17:30	1 óra	Filep Jakabos Kálmán Kuzsner	Értekezés: A Start függvény kommunikációs diagramjáról, Belső határidőt egyeztettünk: Későbbi hetekben továbbra is Csütörtök este 8
2020.03.22 14:00	2 óra	Puskás	Tevékenység: Save Player kommunikációs diagram elkészítése, illetve End turn Use-Case leírásának és diagramjainak elkészítése.
2020.03.22 15:00	2 óra	Kálmán	Tevékenység: Wear/take off DivingSuit Use-Case leírásának, szekvencia diagramjának és kommunikációs diagramjának elkészítése.
2020.03.22 17:00	30 perc	Kálmán	Tevékenység: Use-Case diagram módosítása.
2020.03.22 19:00	3 óra	Kálmán	Tevékenység: Szkeleton kezelői felületének és dialógusok tervének elkészítése.

2020.03.22 22:00	20 perc	Kálmán	Tevékenység: Dokumentum formázása.
2020.03.23 10:00	3 óra	Kuzsner	Tevékenység: Finish és Inspect Ice float kommunikációs diagramok megírása, Inspect Ice Float szekvenciadiagram újrarajzolása

06-Szkeleton beadása

47 – cereal_killers

Konzulens:

**Micskei Zoltán
Vörös András**

Csapattagok

Filep Leonárdó Márk	F6JQ0X	filep.leonardo@gmail.com
Kálmán Bendegúz Bence	PTW6BD	kalmanbendeguz@gmail.com
Jakabos Csenge Apolka	KO6MG0	jakabos11@gmail.com
Puskás Levente	SN8YG9	puskas.levente99@gmail.com
Kuzsner Sára	HVE3FV	kuzsnersara@gmail.com

2020.03.29.

6. Szkeleton beadás

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Field.java	5428 byte	Március 26, 2020	Mező osztály
Direction.java	127 byte	Március 26, 2020	Az irányt meghatározó osztály
Controller.java	6599 byte	Március 24, 2020	A játékot vezérlő osztály
EndOfGame.java	69 byte	Március 26, 2020	A játék befejezéséhez használt interfész
Charge.java	264 byte	Március 25, 2020	Töltény osztály
DivingSuit.java	683 byte	Március 25, 2020	Búvárruha osztály
Flare.java	258 byte	Március 25, 2020	Jelzőfény osztály
Food.java	503 byte	Március 25, 2020	Étel osztály
Gun.java	253 byte	Március 25, 2020	Pisztoly osztály
Item.java	386 byte	Március 25, 2020	Eszköz osztály
Rocket.java	599 byte	Március 25, 2020	Rakéta osztály
Rope.java	1619 byte	Március 25, 2020	Kötél osztály
Shovel.java	1397 byte	Március 25, 2020	Ásó osztály
Main.java	308 byte	Március 26, 2020	A játék indításához szükséges osztály
Menu.java	1270 byte	Március 26, 2020	Menü osztály
MenuItem.java	94 byte	Március 26, 2020	A menü elemeit összefogó osztály
Eskimo.java	563 byte	Március 25, 2020	Eszkimó osztály
Player.java	9042 byte	Március 25, 2020	Játékos osztály
Scientist.java	628 byte	Március 25, 2020	Kutató osztály

6.1.2 Fordítás

A.: Eclipse

Hozunk létre egy új projektet Eclipse-ben, majd importáljuk a forrásfájlokat. Az egyes fájlok package-ekben vannak (mappákban) csoportosítva. Úgy importáljuk ezeket, hogy a hierarchia maradjon meg. Állítsuk át a kódolást UTF-8-as kódolásra (ha szeretnénk, hogy a kommentek ékezetei helyesen kódolva jelenjenek meg). Ctrl+B-vel tudunk fordítani. (Bár a futtatás előtt fordít az Eclipse)

B.: Parancsos

Ha a gépünkre telepítve van a JDK és a környezeti változók be vannak állítva, lépjünk a projekt mappájába. Ezen belül navigálunk az src mappába, és adjuk ki a következő parancsot:

```
javac -encoding ISO-8859-1 -release 11 field/*.java icefield/*.java item/*.java menu/*.java player/*.java
```

Megjegyzés: a parancs fordított per-jelekkel működik.

Ekkor a projekt lefordítva készen áll a futtatásra.

6.1.3 Futtatás

A.: Eclipse

Állítsuk be a “Run configuration”-ben a menu package Main osztályát a main osztálynak, majd ezután futtatjuk.

B.: Parancssor

Ha a JRE a gépünkre telepítve van, miután a projektet lefordítottuk, ugyanebből a mappából adjuk ki a következő parancsot:

java menu.Main

6.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Filep Leonardó Márk	F6JQ0X	20
Kálmán Bendegúz Bence	PTW6BD	20
Jakabos Csenge Apolka	KO6MG0	20
Puskás Levente	SN8YG9	20
Kuzsner Sára	HVE3FV	20

6.3 Napló

Kezdet	Időtartam	Részttvevők	Leírás
2020.03.24. 20:00	2 óra	Filep Jakabos Kálmán Kuzsner Puskás	Értekezés: A skeleton megírásáról Döntés: A munka felosztása öt részegységre
2020.03.25 12:00	2 óra	Puskás	Tevékenység: Player, Eskimo, Scientist osztály implementálása, függvények létrehozása
2020.03.25. 16:00	1 óra	Kálmán	Tevékenység: Item, Shovel, Food, Charge, DivingSuit, Flare, Gun, Rocket, Rope osztály törzsének létrehozása és Use() metódusának létrehozása.

2020.03.25. 23:30	1 óra	Kuzsner	Tevékenység: A Controller osztály megtervezése, változtatások behozása
2020.03.26. 8:00	4 óra	Jakabos	Tevékenység: A Field osztály implementálása, egyeztetés esetleges változtatásokkal kapcsolatosan, hibák javítása.
2020.03.26. 10:00	4 óra	Kuzsner	Tevékenység: A Controller osztály írása, függvények létrehozása, End Of Game interface létrehozása
2020.03.26 10:00	3 óra	Puskás	Tevékenység: A Player, Eskimo, Scientist függvényeinek megírása, kommentelése, egyeztetés esetleges változtatásokról, hibák javítása
2020.03.26. 19:00	3 óra	Kuzsner	Tevékenység: A Controller osztály függvényeinek befejezése
2020.03.26. 20:00	30 perc	Puskás	Tevékenység: Player, Eskimo, Scientist osztály konstruktörának változtatása, egyeztetés miatt új függvények bevezetése.
2020.03.26. 21:00	1 óra	Jakabos	Tevékenység: Field osztály hibáinak javítása, GetUpsideDown(), SetHasIgloo() implementálása
2020.03.27 18:00	4 óra	Puskás	Tevékenység: Player osztálynak új osztályok implementálása, Turn(), UseItem(String) megírása, hibák javítása, egyeztetés változtatásokról, függvények változtatása
2020.03.26. 10:00	2 óra	Filep	Tevékenység: A Menu osztály implementálása, és a Main() függvény létrehozása.
2020.03.27. 18:00	1 óra	Filep	Tevékenység: A Menu osztály függvényeinek csiszolása javítása.
2020.03.28 12:00	2 óra	Puskás	Tevékenység: A Player osztály Turn() függvényének bővítése, RemoveSnow() implementálása, javítások, változtatások.

2020.03.28. 15:00	10 perc	Jakabos	Tevékenység: DecrementSnow() és ListItems() implementálása.
2020.03.29. 21:00	1 óra	Jakabos	Tevékenység: Dokumentum szerkesztése, fájllista, értékelés.
2020.03.29 22:00	1 óra	Puskás	Tevékenység: Dokumentum szerkesztése, fordítás, futtatás. Forráskód tesztelése virtuális gépen.
2020.03.30 10:00	30 perc	Kálmán	Tevékenység: Parancssoros fordítási és futtatási útmutató elkészítése és tesztelése a virtuális gépen.

07-Prototípus koncepciója

47 – cereal_killers

Konzulens:

**Micskei Zoltán
Vörös András**

Csapattagok

Filep Leonárdó Márk	F6JQ0X	filep.leonardo@gmail.com
Kálmán Bendegúz	PTW6BD	kalmanbendeguz@gmail.com
Bence		
Jakabos Csenge Apolka	KO6MG	jakabos11@gmail.com
	0	
Puskás Levente	SN8YG9	puskas.levente99@gmail.co m

Kuzsner Sára

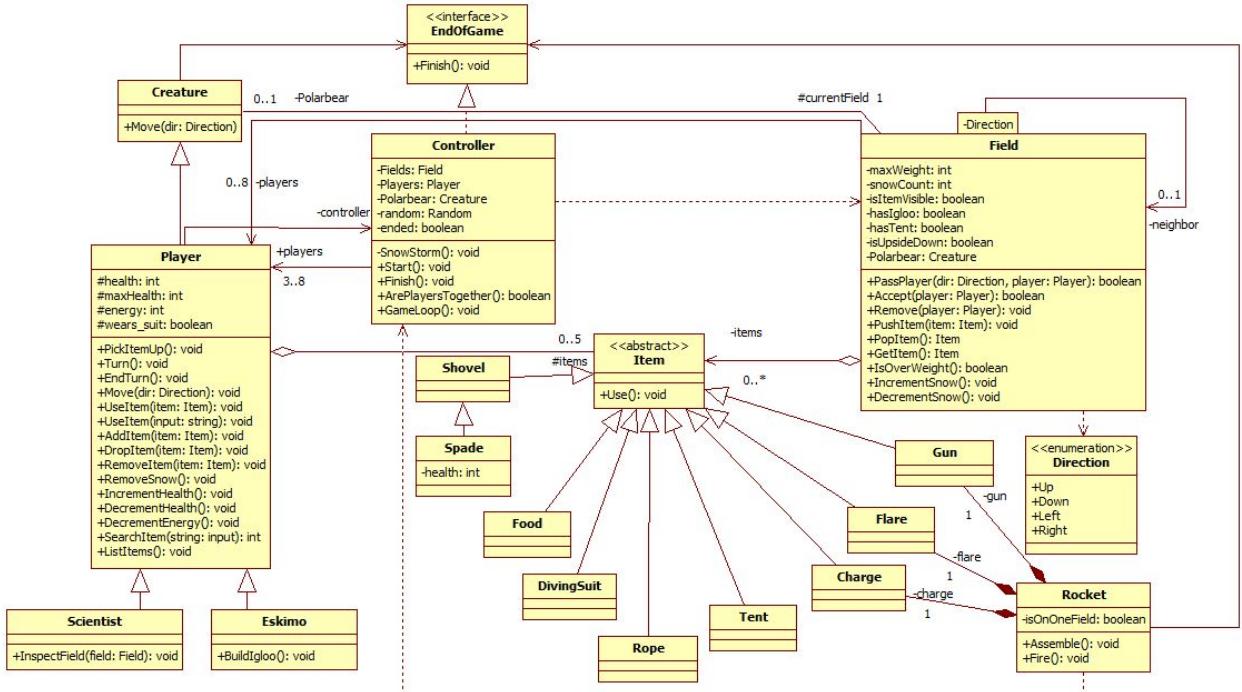
HVE3FV kuzsnersara@gmail.com

2020.04.05.

7. Prototípus koncepciója

7.0 Változás hatása a modellre

7.0.1 Módosult osztálydiagram



7.0.2 Új vagy megváltozó metódusok

Controller:

- **void Start()**: Létrehozza a pályát, a tárgyakat, a játékosokat, és elhelyezi őket-
- **void GameLoop()**: Ez a metódus felel a játék futásáért, ebben lépnek a szereplők és indulnak a hóviharok.

EndOfGame

- **void Finish()**: A játék végén meghívódó függvény, a GameLoopból való kiléést biztosítja.

Field:

- **IncrementSnow()**: Egyel csökkenti az adott mezőn található hó mennyiségét.
- **DecrementSnow()**: Egyel növeli az adott mezőn található hó mennyiségét.

Creature:

- **Move()**: Mozgatja a szereplőt a paraméterben megadott irányban lévő mezőre.

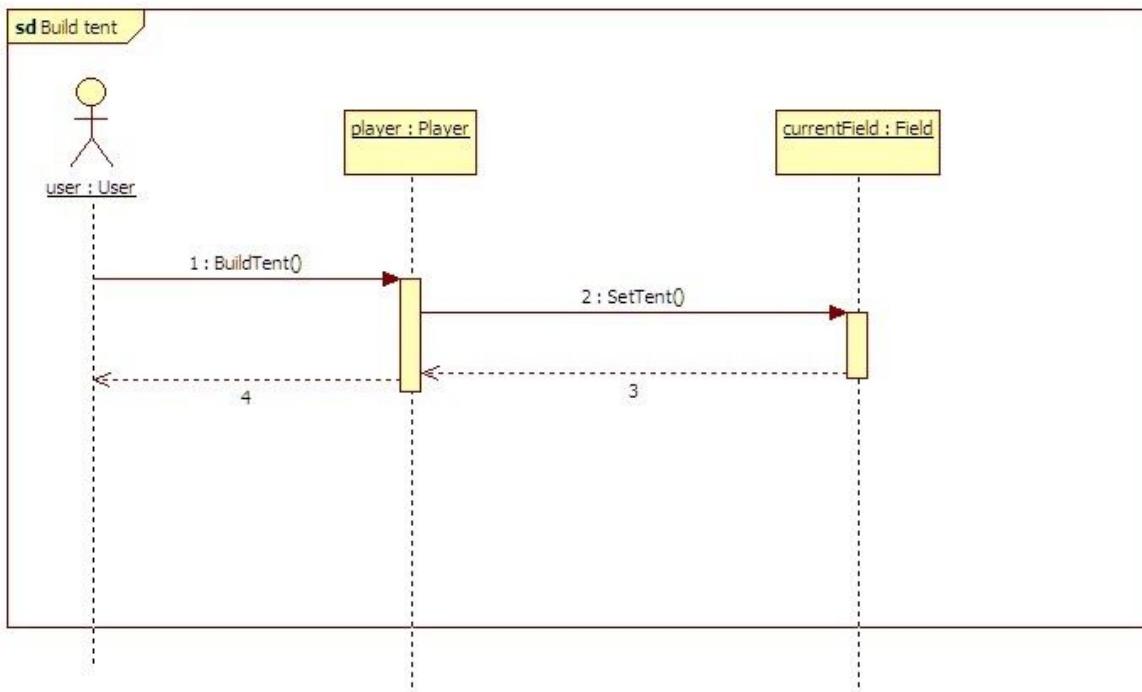
Player:

- **void Turn()**: A játékos által lejátszható lépésekért felelő metódus.
- **void RemoveSnow()**: Egy egység hó letakarítása kézzel.
- **void IncrementHealth()**: Egyel csökkenti a szereplő testhőmérsékletének értékét.

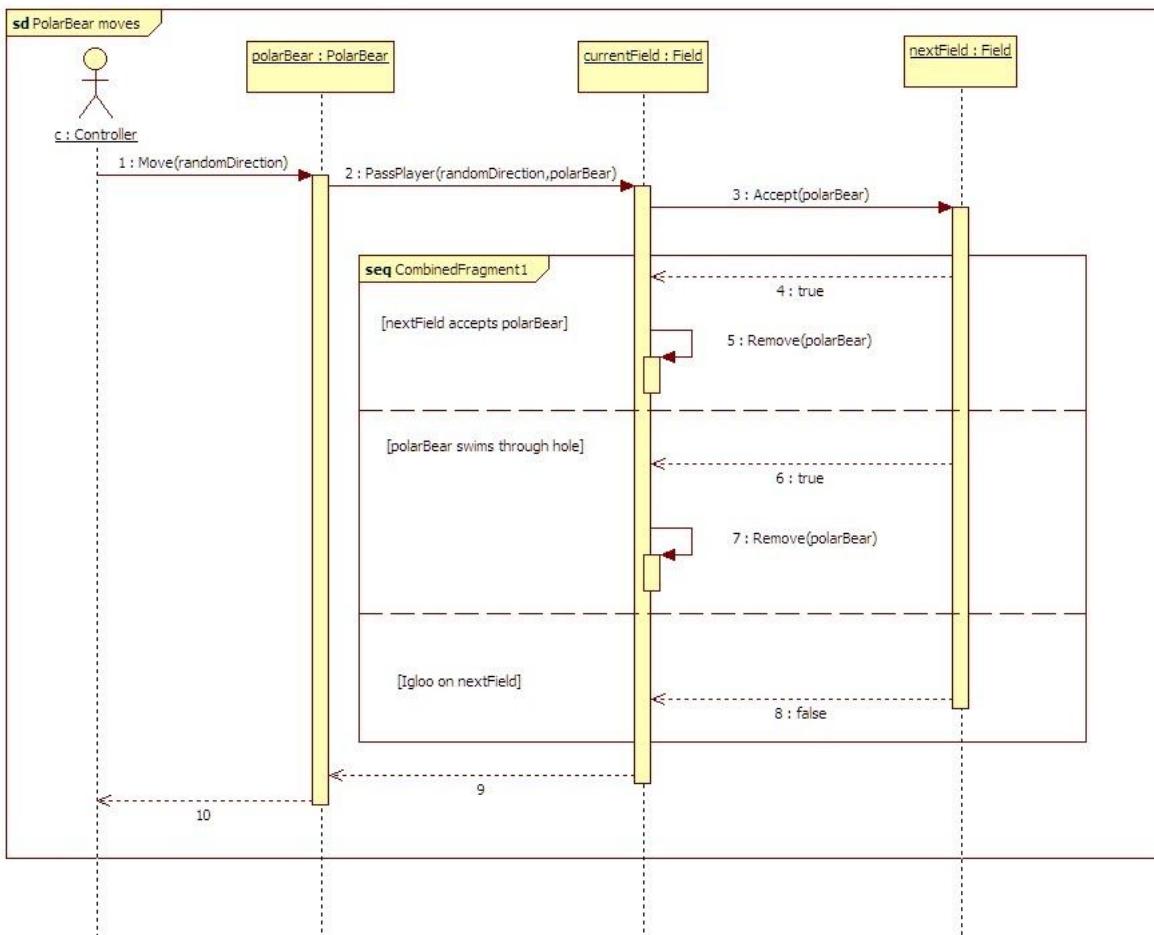
- **void DecrementHealth()**: Egyel növeli a szereplő testhőmérsékletének értékét.
- **void DecrementEnergy()**: Egyel csökkenti a szereplő energiájának értékét.
- **int SearchItem(string input)**:
- **void UseItem(String input)**: Név alapján keresi az eszközt és ha megtalálta használja.

7.0.3 Szekvencia-diagramok

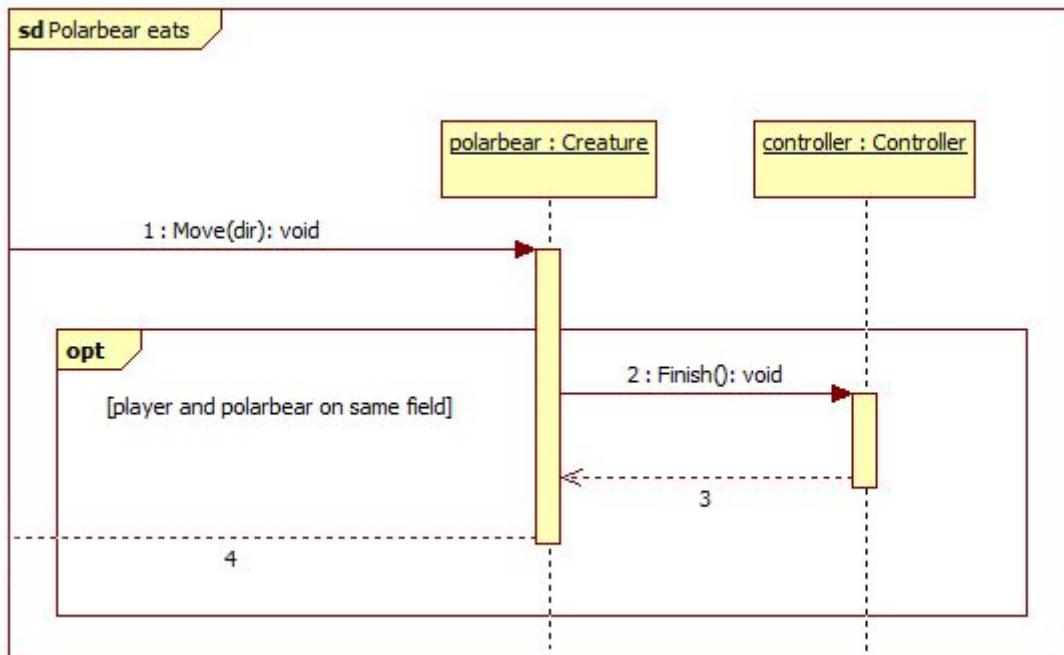
Build tent



Polarbear moves



Polarbear Eats



7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A program prototípusa parancssoron keresztül vezérelhető, és a kimenete is a konzolra íródik. Szituációtól függően más-más parancsok elérhetőek, például a menü parancsai nem érhetőek el játék közben, és fordítva. Az *inspect* parancs csak sarkkutatóként, az *igloo* parancs csak eszkimóként érhető el. Ha a parancs kontextusba illik, de nem lehet végrehajtani, (pl. rakéta elsütése, amikor még nincs összeszerelve), akkor arról informatív visszajelzést kap a felhasználó. Alapvetően azok a parancsok, amelyek logikailag összetartoznak, egy parancsot alkotnak, más argumentummal (pl. use parancs argumentuma, hogy milyen tárgyat szeretnénk használni). Ha a felhasználó érvénytelen parancsot ad ki, a program tájékoztatja a felhasználót, hogy ilyen parancs nem létezik. Ha érvényes a parancs, de az argumentum hibás, arról is visszajelzést kapunk. Az utasításokat fájlból is be tudjuk majd olvastatni, amelyben soronként egy parancsot helyezhetünk el.

7.1.2 Bemeneti nyelv

random

Leírás: A játékban történő véletlen események determinisztikusságát lehet ezzel a parancssal állítani. A játék bármely pillanatában a parancs kiadása azt jelenti, hogy az adott állapotból indulva akárhányszor lefuttatva a programot ugyanazzal a bemenettel, pontosan ugyanazt a kimenetet kapjuk. Ha a játék kezdete előtt adjuk ki, akkor minden alkalommal pontosan ugyanolyan játékmenetben lesz részünk, amennyiben ugyanazokat a parancsokat adjuk ki.

Opciók: random <kapcsoló>

kapcsoló: on / off

Az “on” opcióval be, az “off” opcióval kikapcsolhatjuk a program véletlenszerűségét.

Menüben: <szám>

Leírás: Amikor a menüben vagyunk, felajánlódnak a lehetőségek, hogy melyik menüpontot tudjuk kiválasztani. Ezeket számokkal tudjuk kiválasztani, és kiválasztás után az adott menüpontnak megfelelő folyamat játszódik le.

Opciók: <szám> [mentett_fájl]

szám: 1-4-ig terjedhet

1 [mentett_fájl]: Új játék kezdődik. Ha megadunk paraméterben egy mentett játékállást, akkor az amennyiben értelmes, onnan tölti be a játékot.

2: Megjelenik a beállítások almenü.

3: Megjelenik a ranglista.

4: Kilépünk a programból.

menu

Leírás: A játék során adhatjuk ki ezt a parancsot, amely hatására a menübe lépünk.

Opciók: menu

A menu parancs nem menti el a játékállást, arról külön kell gondoskodni.

endturn

Leírás: Amikor egy szereplővel már nem kívánunk tevékenykedni, ezzel a parancssal jelezzük a kör végét, és adjuk át a vezérlést a soron következő szereplőnek.

Opciók: endturn

pickup

Leírás: A szereplő az adott mezőn lévő látszódó tárgyat tudja felvenni ezzel a parancssal. Mindig a sorban legfelső Itemhez jutunk így hozzá, amennyiben van hely az eszköztárunkban.

Opciók: pickup

drop

Leírás: A szereplő a nála lévő tárgyak valamelyikét tudja ezzel a parancssal letenni az aktuális mezőre. Csak teljesen letakarított mezőre lehet tárgyat tenni.

Opciók: drop <item>

item: spade / shovel / food / divingsuit / rope / tent / charge / flare / gun

dig

Leírás: A szereplő ennek a parancsnak a hatására egy energia árán egy egység havat takarít el arról a mezőről, amin áll. Ha a mezőn nincs hó, akkor minden változatlan marad. Az ásóval és törékeny ásóval való ásás másik parancshoz tartozik.

Opciók: dig

move

Leírás: A szereplőt ezzel a parancssal tudjuk mozgatni a pályán. Egy mezőnyi mozgás egy energiát vesz igénybe. Ha a pálya széléről próbálunk a pálya szélénél irányába mozogni, akkor azt nem tudjuk megtenni.

Opciók: move <irány>

irány: up / down / left / right

inspect

Leírás: Amennyiben az éppen vezérelt szereplő sarkkutató, akkor lesz ennek a parancsnak hatása, a paraméterül kapott mezőnek kiíródik a teherbírása. A sarkkutatónak ez egy energiájába kerül.

Opciók: inspect <irány>

irány: up / down / left / right

Ha például ezt írjuk: "inspect right", akkor a sarkkutatótól jobbra lévő mező teherbírását tudjuk meg.

use

Leírás: A szereplő ezzel a parancsal tudja használni az eszköztárában található dolgokat. Mivel sokféle eszköz van, a parancsot sokféleképpen lehet használni. Egyes eszközök használata egy energiát, más eszközök használata egy energiát sem igényel.

Opciók: use <item>

item: spade / shovel / food / divingsuit / rope / tent

spade: Törékeny ásó, használatakor egy energiával két havat lehet vele ellapárolni, de három használat után eltörik, és megszűnik létezni.

shovel: Ásó, használatakor egy energiával két havat lehet vele ellapárolni.

food: Élelem. Használatakor a szereplő életpontja eggel nő. Használat után eltűnik. Használata nem kerül energiába.

divingsuit: Búvárruha. Használatakor a szereplőre felrakhatjuk a búvárruháját amennyiben van az eszköztárában, vagy levehetjük róla, ha már visel búvárruhát. Használata nem kerül energiába.

rope: Kötél. Használatakor a szereplő kötéllel kimenti a körülötte lévő mezőkön elhelyezkedő vízbe esett összes szereplőt, akik az ő mezőjére kerülnek. Egy energiába kerül a használata.

tent: Sátor. Használatakor a szereplő lerakja az aktuális mezőre a sárat. Csak teljesen letisztított mezőre lehet sárat rakni. A sátorállítás egy energiába kerül. A sátor egy kör után eltűnik, addig viszont pontosan úgy viselkedik, mint egy iglu.

igloo

Leírás: Eszkimónak kiadható parancs. Ennek hatására az eszkimó egy iglut épít arra a mezőre, amin áll. Iglut csak stabil mezőre lehet építeni. Az igluépítés az eszkimó egy energiájába kerül.

Opciók: igloo

assemble

Leírás: Amennyiben a szereplők az összes rakéta-alkatrészt letették ugyanarra a mezőre, és minden szereplő szintén ugyanezen a mezőn van, az egyik szereplővel kiadható ez a parancs, ekkor egy energia felhasználásával összeszerelődik a rakéta.

Opciók: assemble

fire

Leírás: Amennyiben a rakéta össze van szerelve egy mezőn, és minden szereplő szintén ugyanezen a mezőn van, az egyik szereplővel kiadható ez a parancs, ekkor egy energia felhasználásával elsütődik a rakéta, és a játék győzelemmel véget ér.

Opciók: fire

save

Leírás: Bármely játékállásban kiadhatjuk ezt a parancsot, mely segítségével elmenthetjük az adott játékállást egy fájlba, amiből később betölthetjük, és folytathatjuk a játékot. Ha a determinisztikusságot bekapsoljuk, ugyanaz a mentés minden betöltés után ugyanazt a játékmenetet eredményezi, amennyiben sorban ugyanazokat a parancsokat adjuk ki.

Opciók: save <fájlnév>
fájlnév: pl. "game.txt"

A mentés fájl a játék mappájába kerül.

Beállításoknál: <szám>

Leírás: Ha a menüből a beállítások (Options) menüpontba navigálunk, választhatunk a beállítási lehetőségek közül. Ezekre a sorszámmal lehet hivatkozni, 1-től 4-ig.

Opciók: <szám>
szám: 1-4-ig terjedhet

1: Játékos nevének beállítása. Ha ezt a menüpontot választjuk, utána beírhatunk egy tetszőleges karaktersorozatot, ami az Enter leütése után beállítódik a játékos nevének. Ez fog a ranglistán megjelenni.

2: A játékbeli szereplők számának megadása. Ez a szám 3-tól 8-ig terjedhet. Ha páratlan számot írunk, akkor véletlenszerű lesz, hogy eszkimóból, vagy sarkutatóból lesz-e több eggel.

3: A játék determinisztikusságának állítása, ide on / off -t írhatunk.

4: A játékbeli zene be- és kikapcsolása. Ide is on / off opciókat tudunk írni.

7.1.3 Kimeneti nyelv

menu

A menü amikor megjelenik, a következő 4 sort írja ki:

1. *New Game*
2. *Options*
3. *Best Scores*
4. *Exit*

Ha az 1-es számot írjuk be, új játék indul, és ez íródik ki: *Player I's turn*. Ezután a program várja az első parancsot. Ha fájlból indítjuk a játékot, akkor az éppen soron következő játékos következik. Ha abban a körben épp egy hóvihar jön, akkor mielőtt kiíródik a *Player N's turn*, ez íródik ki: *snowstorm happened*.

Ha a 2-es számot írjuk be, megjelennek a beállítási lehetőségek a következő 4 sorral:

1. *Player name*
2. *Number of players*
3. *Music*

Ha 3-as számot írunk be, megjelenik a ranglista, soronként egy sorszámmal és egy névvel, pl.:

1. *Hamupipőke*
2. *Terminátor*
3. *Szíriusz Blekk*
4. *Sarkkutató Sándor*

Ha a 4-es számot írjuk be, nem íródik ki semmi, hanem bezáródik a program.

random

A parancsra az opciókra tekintettel a következő kimenetek várhatók:

random off: *random switched off*
random on: *random switched on*

endturn

A parancsra következő kimenet várható:

player <n+1>'s turn : <playertype>

Ahol n+1 a következő játékos sorszáma, playertype pedig hogy eszkimó, vagy sarkkutató-e az illető.

playertype: eskimo / scientist

Példa:

player 3's turn

pickup

A parancsra következő kimenetek várhatók:

<item> picked up

Példa:

shovel picked up

Ha nem látható tárgy a mezőn:

no item visible

Ha nincs hely az eszköztárunkban:

inventory full

drop

A parancsra a következő kimenet várható:

<item> dropped

Ha az adott mező nincs teljesen letakarítva:

field not clear

Példa:

food dropped

tent dropped

dig

A parancsra a következő kimenetek várhatók:

Ha a mezőn van hó és a szereplőnek van legalább 1 energiája:

1 snow removed

remaining energy: <N>

Ahol N a szereplő megmaradt energiája.

Ha a játékosnak nincs elég energiája:

no energy

Ha a mezőn nincs hó:

0 snow removed

remaining energy: <N>

move

A parancsra következő kimenetek várhatók:

Alapesetben, ha az egyik mezőről átmegyünk egy másikra:

remaining energy: <N>

Példa:

remaining energy: 2

Ha lyukba esünk:

player fell into hole

remaining energy: <N>

Ha nincs elég energia a mozgáshoz:

no energy

inspect

A parancsra következő kimenetek várhatók:

stability: <N>

remaining energy: <M>

Ahol N a megfigyelt mező teherbírása, M a sarkkutató megmaradó energiája.

Példa:

stability: 4

remaining energy: 3

Ha nincs elég energia a képességhez:

no energy

use

A parancsra az opciókra tekintettel a következő kimenetek várhatók:

Mindegyikre vonatkozik:

Minden tárgy használata egy egység energiába kerül.

Ha a szereplőnél nincs az adott tárgyból:

player doesn't have this item

use shovel:

Ha nincs elég energia:

no energy

Ha nincs hó a mezőn:

0 snow removed

remaining energy: <N>

Ahol N a szereplő energiája, ami most nem csökkent.

Ha 1 egység hó van a mezőn:

1 snow removed

remaining energy: <N>

Ha 2 vagy több egység hó van a mezőn:

2 snow removed

remaining energy: <N>

Az utóbbi két esetben 1-el csökken az energia.

use spade:

Ha nincs elég energia:

no energy

Ha nincs hó a mezőn:

0 snow removed

remaining energy: <N>

spade condition: <M>

Ahol N a szereplő energiája, ami most nem csökkent, és M az ásó állapota.

Ha 1 egység hó van a mezőn:

1 snow removed

remaining energy: <N>

spade condition: <M>

Ha 2 vagy több egység hó van a mezőn:

2 snow removed

remaining energy: <N>

spade condition: <M>

Az utóbbi két esetben 1-el csökken az energia.

use food:

player health: <N>

remaining energy: <M>

Ahol N a szereplő új életpontja, M a fennmaradó energiaszintje.

Ha nincs elég energia:

no energy

use divingsuit:

Ha rajta van a búvárruha:

divingsuit off

remaining energy: <N>

Ha nincs rajta:

divingsuit on

remaining energy: <N>

Ha nincs elég energia:

no energy

use rope:

<N> players saved

remaining energy: <M>

Ahol N az a szám, ahány szereplőt sikerült kimenteni, M pedig a fennmaradó energiaszint.

Ha nincs elég energia:

no energy

use tent:

Alapeset:

tent set

remaining energy: <N>

Ha nincs elég energia:

no energy

Ha a mező nem stabil:

field unstable

igloo

A parancsra a következő kimenetek várhatók:

igloo built

remaining energy: <N>

Ahol N az eszkimó fennmaradó energiája.

Ha nincs elég energia:

no energy

Ha a mező, amire az iglут építénk, nem stabil:

field unstable

assemble

A parancsra a következő kimenetek várhatók:

rocket assembled

Ha az alkatrészek nincsenek egy mezőn:

pieces not on same field

Ha az alkatrészek igen, de a játékosok nincsenek mind ugyanezen a mezőn:

not all players on rocket field

Ha az összeszerelő szereplőnek nincs elég energiája:

no energy

fire

A parancsra a következő kimenetek várhatók:

rocket fired

victory

1. New Game

2. Options

3. Best Scores

4. Exit

Tehát ekkor visszajutunk a menübe.

Ha a rakéta nincs összeszerelve:

rocket not assembled

Ha a játékosok nincsenek mind ugyanazon a mezőn amin a rakéta van:

not all players on rocket field

Ha az elsütő szereplőnek nincs elég energiája:

no energy

save

A parancsra a következő kimenetek várhatók:

game saved to <fájlnév>

fájlnév pl: *savegame.txt*

7.2 Összes részletes use-case

Use-case neve	Save parancs kiadása
Rövid leírás	A felhasználó elmentheti a játékot.
Aktorok	User
Forgatókönyv	1. A játék mentésre kerül.

Use-case neve	New Game kiválasztása
Rövid leírás	A felhasználó megnyitja a játékot elindító menüpontot.
Aktorok	User
Forgatókönyv	1. A játék elindul, a pálya megjelenik a rajta lévő tárgyakkal és szereplőkkel, elindul az első kör.

Use-case neve	Menü kiválasztása
Rövid leírás	A felhasználó megnyitja a menü opciót.
Aktorok	User
Forgatókönyv	1. A Játék megjeleníti a lehetséges opciókat, amiből a játékos választani tud.

Use-case neve	Settings kiválasztása
Rövid leírás	A felhasználó megnyitja a beállítások opciót a menüben.
Aktorok	User
Forgatókönyv	1. A Játék megjeleníti a lehetséges beállításokat, amiből a játékos választani tud.

Use-case neve	Best Scores kiválasztása
Rövid leírás	A felhasználó megnyitja a toplistát a menüben.
Aktorok	User
Forgatókönyv	1. A játék megjeleníti a legjobb játékosokat.

Use-case neve	Exit kiválasztása
Rövid leírás	A felhasználó kilép a játékból.
Aktorok	User
Forgatókönyv	1. A játék bezárja az ablakot és leáll.

Use-case neve	Szereplő mozgatása move paranccsal
Rövid leírás	A felhasználó lép az egyik játékossal, ami ennek hatására elmozdul a pályán.
Aktorok	User
Forgatókönyv	<p>1. A játékos a kiválasztott mezőre lép, és eggyel csökken a játékos energiaszintje.</p>
Alternatív forgatókönyv	<p>1.A.1. A játékos a kiválasztott irányban lévő szomszédos mezőre lép, amennyiben nem bírja el, a vízbe esik.</p> <p>1.B.1. A játékos a kiválasztott irányban lévő szomszédos mezőre lép, amennyiben ott jegesmedve van meghal és a játéknak vége.</p> <p>1.C.1. A játékosnak nincs energiája mozogni, ezért a helyén marad.</p>

Use-case	Tárgy használata use parancssal
Rövid leírás	A felhasználó használja a kiválasztott tárgyat.
Aktorok	User
Forgatókönyv	<p>1. A játékos használja az adott tárgyat amennyiben van nála olyan.</p>
Alternatív forgatókönyv	1.A.1. Nem történik semmi amennyiben a játékosnál nincs olyan tárgy.

Use-case	Kötél használata use rope paranccsal
Rövid leírás	A játékos kimenti a társát.
Aktorok	User
Forgatókönyv	<p>1. A játékos kimenti a társát a kötéssel amennyiben van nála olyan.</p>
Alternatív forgatókönyv	1.A.1. Nem történik semmi, amennyiben a játékosnál nincs kötél.

Use-case	A játékos eszik a use food parancsal
Rövid leírás	A felhasználó meghívja a use food parancsot, ezzel növelte a játékos testhőmérsékletét.
Aktorok	User
Forgatókönyv	1. A játékos eszik, az étel megszűnik létezni és a játékos testhőmérséklete megnő eggyel.
Alternatív forgatókönyv	1.A.1. Amennyiben nincs a játékosnál étel nem történik semmi.

Use-case	Sátor használata use tent parancsal
Rövid leírás	A felhasználó meghívja a use tent parancsot, a játékos sárat épít a mezőre amin áll.
Aktorok	User
Forgatókönyv	1. A játékos sárat épít a mezőre amin áll.
Alternatív forgatókönyv	1.A.1. Nem történik semmi amennyiben a játékosnál nincs sátor.

Use-case	Különleges képesség használata inspect parancsal Scientist szereplő tevékenységeként
Rövid leírás	A felhasználó kiadja az inspect parancsot, aminek hatására a játékos használja a különleges képességét.
Aktorok	User
Forgatókönyv	1. A sarkutató megvizsgálja a jégtáblát, energiaszintje eggyel csökken.
Alternatív forgatókönyv	1.A.1. A szereplőnek nincs elég energiája a művelethez.

Use-case	Különleges képesség használata igloo parancsal Eskimo szereplő tevékenységeként
Rövid leírás	A felhasználó kiadja az igloo parancsot, aminek hatására a játékos használja a különleges képességét.
Aktorok	User
Forgatókönyv	1. Az eszkimó felépíti az iglut, energiaszintje eggyel csökken.
Alternatív forgatókönyv	1.A.1. A szereplőnek nincs elég energiája a művelethez.
Alternatív forgatókönyv	1.B.1. Nem lehet iglut építeni az adott mezőre, mert nem stabil.

Use-case	Tárgy felvétele pickup parancsal
Rövid leírás	A tárgy bekerül a játékos eszköztárába.
Aktorok	User
Forgatókönyv	1. Az eszköz az eszköztárba kerül.
Alternatív forgatókönyv	1.A.1. Az eszköztárban nincs hely a tárgy számára, ekkor az eredeti helyén marad.

Use-case	Tárgy lerakása drop <item> parancssal
Rövid leírás	A tárgy a szereplő eszköztárából a jégtáblára kerül.
Aktorok	User
Forgatókönyv	1. Az eszköz a jégtáblára kerül.
Alternatív forgatókönyv	1.A.1. A jégtáblán nincs hely, ezért az eszköztárban marad.

Use-case	Hó eltávolítása use shovel vagy dig parancsokkal
Rövid leírás	Hó eltávolítása egy adott mezőről.
Aktorok	User
Forgatókönyv	1. Amennyiben a játékosnál nincsen ásó eggel csökken a mezőn található hó mennyisége, eggel csökken a játékos energiaszintje.
Alternatív forgatókönyv	1.A.1. Amennyiben a játékosnál van ásó kettővel csökken a mezőn található hó mennyisége, eggel csökken a játékos energiaszintje.

Use-case neve	Rakéta összeszerelése és elsütése az assemble és a fire parancsok hatására
Rövid leírás	A szereplők ugyan arra a jégtáblára lépnek és egy egység munka árán összeszerelik és elsülik a rakétát a komponensekből.
Aktorok	User
Forgatókönyv	1. A szereplők összeszerelik és elsülik a rakétát.

Use-case neve	Szereplő vízből való mentése use rope parancssal
Rövid leírás	Ha egy szereplő vízbe esik, egy, a közelben levő szereplő akinek van kötele kimentheti.
Aktorok	User
Forgatókönyv	1. A szereplő kimenekíti társát egy kötéssel

Use-case neve	Játékból való kilépés a menu parancs kiadásával
Rövid leírás	A felhasználó kiléphet a játékból a menu parancs kiadásával.
Aktorok	User
Forgatókönyv	1. A játékos kilép a játékból a menübe.

Use-case neve	Játék zenéjének ki/be kapcsolása
Rövid leírás	A felhasználó ki/be kapcsolhatja a játék zenéjét.
Aktorok	User
Forgatókönyv	1. Zene bekapcsolása.
Alternatív forgatókönyv	1.A.1. Zene kikapcsolása.

Use-case neve	Random parancs kiadása
Rövid leírás	A felhasználó ki/be kapcsolhatja a játék determinisztikusságát.
Aktorok	User
Forgatókönyv	1. Funkció bekapcsolása.
Alternatív forgatókönyv	1.A.1. Funkció kikapcsolása.

Use-case neve	Játékos nevének beállítása
Rövid leírás	A felhasználó beállíthatja a menüben a nevét, hogy a best scores listán ez kerüljön megjelenítésre.
Aktorok	User
Forgatókönyv	1. A játékos beállítja a nevét.

Use-case neve	Kör befejezése endturn parancssal
Rövid leírás	A játékos befejezi a tevékenységét a jelenlegi körben.
Aktorok	User
Forgatókönyv	1. A játékos befejezi tevékenységét és megjelenik a soron következő játékos száma..

Use-case neve	Játékosok számának beállítása
Rövid leírás	A felhasználó beállíthatja a menüben a játékosok számát.
Aktorok	User
Forgatókönyv	1. A játékos beállítja a játékosok számát.

7.3 Tesztelési terv

Teszt-eset neve	1. Player steps on Field
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos egy szomszédos Field-re lép.
Teszt célja	A játékos mezőre lép.

Teszt-eset neve	2. Player steps on Hole
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos olyan mezőre lép, ami egy lyuk.
Teszt célja	A játékos vízbe esik.

Teszt-eset neve	3. Player steps on side of map
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos a pálya szélén túlra próbál lépni.
Teszt célja	A játékos lépése nem történik meg.

Teszt-eset neve	4. Player steps on unstable Field that flips
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos olyan mezőre lép, ami instabil és már elérte a súlykapacitásának maximumát.
Teszt célja	A mező átfordul és a rajta lévő játékosok vízbe esnek.

Teszt-eset neve	5. Player steps in Igloo
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos olyan mezőre lép, amin iglu van.
Teszt célja	A játékos iglus mezőre lép.

Teszt-eset neve	6. Player steps in Tent
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos olyan mezőre lép, amin sátor van.
Teszt célja	A játékos sátras mezőre lép.

Teszt-eset neve	7. Player steps on Field with polar bear on it
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos olyan mezőre lép, amin jegesmedve van.
Teszt célja	A játéknak vége, a csapat veszít.

Teszt-eset neve	8. Polar bear steps on Field with Player on it
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a jegesmedve olyan mezőre lép, amin játékos van.
Teszt célja	A jegesmedve olyan mezőre lép, amin játékos van.

Teszt-eset neve	9. Polar bear steps on Field with igloo on it
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a jegesmedve olyan mezőre lép, amin iglu van és az igluban játékos.
Teszt célja	A játék folytatódik.

Teszt-eset neve	10. Polar bear steps on Field with tent on it
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a jegesmedve olyan mezőre lép, amin sátor van és a sátorban játékos.
Teszt célja	A játéknak vége, a csapat veszít.

Teszt-eset neve	11. Player saves Player
Rövid leírás	Lemodellezük azt a folyamatot, ahogy egy játékos kiment egy másik játékosat vízből kötél segítségével.
Teszt célja	Egy játékos kihúz egy másik játékosat a vízből.

Teszt-eset neve	12. Snowstorm hits Field
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a hóvihar lesúlyt egy mezőre és annak hómennyisége eggel nő.
Teszt célja	A mező hómennyisége eggel nő.

Teszt-eset neve	13. Snowstorm hits Player
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a hóvihar lesúlyt egy játekosra és annak életpontjait eggel csökkenti.
Teszt célja	A játekos életpontjai eggel csökkennek.

Teszt-eset neve	14. Eskimo builds an igloo
Rövid leírás	Lemodellezük azt a folyamatot, ahogy egy eszkimo iglut épít, szigorúan stabil mezőre.
Teszt célja	Az Eskimo létrehoz egy iglut, azon a mezőn, amelyiken éppen tartózkodik.

Teszt-eset neve	15. Scientist inspects Ice Float
Rövid leírás	Lemodellezük azt a folyamatot, ahogy egy kutató megállapítja egy szomszédos mező kapacitását.
Teszt célja	A kutató megállapítja egy szomszédos mező kapacitását.

Teszt-eset neve	16. Player picks up item
Rövid leírás	Lemodellezük azt a folyamatot, ahogy egy játekos felvesz egy tárgyat.
Teszt célja	A játekos felvesz egy tárgyat.

Teszt-eset neve	17. Player drops an item
Rövid leírás	Lemodellezük azt a folyamatot, ahogy egy játekos lerak egy tárgyat az eszköztárából.
Teszt célja	A játekos lerak egy tárgyat.

Teszt-eset neve	18. Player eats food
Rövid leírás	Lemodellezük azt a folyamatot, ahogy egy játékos megeszik egy ételt.
Teszt célja	A játékos megeszi az ételt.

Teszt-eset neve	19. Player removes snow without Shovel
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos eltávolít egy egységnnyi havat a mezőről, lapát használata nélkül.
Teszt célja	A játékos havat távolít el lapát nélkül

Teszt-eset neve	20. Player removes snow with Shovel
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos ellapátol maximum 2 egységnnyi havat.
Teszt célja	A játékos havat lapátol el.

Teszt-eset neve	21. Player removes snow with Spade
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos ellapátol maximum 2 egységnnyi havat, törékeny ásóval.
Teszt célja	A játékos havat lapátol el törékeny lapáttal.

Teszt-eset neve	22. Player wears diving suit
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos felvesz egy búvárruhát.
Teszt célja	A játékos felvesz egy búvárruhát.

Teszt-eset neve	23. Player takes off diving suit
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos leveszi magáról a búvárruhát.
Teszt célja	A játékos leveszi magáról a búvárruhát.

Teszt-eset neve	24. Not enough energy for moving
Rövid leírás	Lemodellezük azt a folyamatot, amikor is a játékos lépni próbál, de nincs elég energiája
Teszt célja	A játékosnak nincs energiája, ezért nem tud lépni.

Teszt-eset neve	25. Not enough energy for using item
Rövid leírás	Lemodellezük azt a folyamatot, amikor is a játékos tárgyat próbál használni, de nincs elég energiája.
Teszt célja	A játékosnak nincs elég energiája, ezért nem tudja használni a tárgyat.

Teszt-eset neve	26. Player signals end turn
Rövid leírás	Lemodellezük azt a folyamatot, ahogy a játékos kör végét jelez.
Teszt célja	A játékos turnus végét jelez, a következő játékos jön.

Teszt-eset neve	27. Spade breaks after using it
Rövid leírás	Lemodelezük azt a folyamatot, amikor is a törékeny ásó használat után eltörök, mivel már háromszor volt használva.
Teszt célja	A törékeny ásó eltörök.

Teszt-eset neve	28. Tent disappears at the start of a new turn
Rövid leírás	Lemodelezük azt a folyamatot, amikor is az előző körben egy mezőre felhúzott sátor új kör elején eltűnik.
Teszt célja	A sátor eltűnik a mezőről.

Teszt-eset neve	29. Players assemble and fire Rocket
Rövid leírás	Lemodelezük azt a folyamatot, ahogy a játékosok összeszerelik és elszülik a rakétát.
Teszt célja	A rakéta elsül, a csapat nyer.

Teszt-eset neve	30. Snowstorm doesn't affect Players in Igloo
Rövid leírás	Lemodelezük azt a folyamatot, ahogy a hóvihar lesújt egy olyan mezőre, amin a játékosok igluban vannak.
Teszt célja	A játékosok életpontjai nem változnak.

Teszt-eset neve	31. Snowstorm doesn't affect Players in Tent
Rövid leírás	Lemodelezük azt a folyamatot, ahogy a hóvihar lesújt egy olyan mezőre, amin a játékosok sátorban vannak.
Teszt célja	A játékosok életpontjai nem változnak.

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A legegyszerűbben a prototípus tesztelése JUnit segítségével fog működni. minden teszt-esetre lesz egy JUnit Test Case, amik egy packageben lesznek csoportosítva. Egy teszt-esetnek a neve Test<i>.java lesz, ahol az <i> a teszt-eset sorszáma. Továbbá minden teszt-esethez tartozzik egy bemeneti- és kimeneti file, test<i>.in és egy test<i>.out. Az test<i>.in bemeneti file tartalmazza a programra szánt bemenetet, a test<i>.out kimeneti file pedig az elvárt kimenetet.

A Test<i>.java-ban hasonlítsuk össze a bemenetre kapott tényleges kimenetet és az elvárt kimenetet.

A standard kimenetre üzenetet küldünk, hogy egy teszteset sikeresen futott-e le vagy sem, emelett megjelenítődik a valós és elvárt kimenet.

7.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.03.30 18:00	1 óra	Kálmán Kuzsner Filep Jakabos Puskás	Értekezés: Feladatváltozások megbeszélése, heti feladatak kiosztása. Döntés: Filep módosítja az osztálydiagramot, Kuzsner az új metódusok leírását készíti el, Puskás a tesztelési tervet. Kálmán az interfész leírását, Jakabos a szekvenciadiagramokat és a Use-Case leírásokat készíti el, amiben Kuzsner is segít neki.
2020.03.31 9:00	1 óra	Filep	Tevékenység: Filep elkészíti az osztálydiagram módosult változatát.
2020.04.01 13:00	2 óra	Kuzsner	Tevékenység: Új és megváltozott metódusok összegyűjtése a Controller() és EndOfGame() osztályokban, diagramra való felvitelük és dokumentálásuk. Slack poll kírása a diagramon feltüntetett metódusok jellegére vonatkozóan.
2020.04.01 21:00	2 óra	Puskás	Tevékenység: Teszt -eseket megírása
2020.04.02 15:00	1 óra	Filep	Tevékenység: Filep javítja az osztálydiagramot.
2020.04.02 16:00	2 óra	Kuzsner	Tevékenység: Use-Case-ek kiválasztásának megtervezése. Új vagy megváltozó metódusok összegyűjtése és megírása, a kód és diagram kapcsolatának elemzése.
2020.04.02 21:00	1 óra	Jakabos	Tevékenység: Jakabos elkészíti a Build tent és PolarBear moves szekvenciadiagramokat.
2020.04.02 18:00	2 óra	Jakabos	Tevékenység: Jakabos elkészíti a Játékosok számának beállítása, Játékos nevének beállítása, Játék zenéjének ki/be kapcsolása, Szereplő vízből való mentése, Rakéta összeszerelése az assambe parancs hatására, Tárgy lerakása drop <item> parancssal, Tárgy felvétele pickup parancssal use case leírásokat.
2020.04.03 12:00	3 óra	Kuzsner	Tevékenység: New Game kiválasztása, Settings kiválasztása, ShowBestScores kiválasztása, Exit kiválasztása, Irányító billentyű lenyomása, Tárgy használata, Különleges képesség használata, Hó eltávolítása Use-Casek megírása.

2020.04.03 16:00:00	1 óra	Kálmán Kuzsner Filep Jakabos Puskás	Értekezés: Az addig elkészült feladatokról, a Use-Case-ek formájáról, és a bemeneti nyelvről. Döntés: A Jegesmedvének létrehozunk egy külön osztályt abból a célból hogy megkülönböztethető legyen más élőlényektől.
2020.04.03 20:00	10 perc	Filep	Tevékenység: Filep elkészíti a Polarbear eats szekvencia-diagramot.
2020.04.05 15:00	3 óra	Jakabos	Tevékenység: Use case-ek átnézése, hibajavítás, Kör befejezése endturn paranccsal, Különleges képesség használata igloo paranccsal Eskimo szereplő tevékenységeként, Különleges képesség használata inspect paranccsal Scientist szereplő tevékenységeként, Save parancs kiadása, Tárgy felvétele pickup paranccsal, Tárgy lerakása drop <item> paranccsal, Rakéta összeszerelése és elsütése az assemble és a fire parancsok hatására, Játék zenéjének ki/be kapcsolása, Random parancs kiadása use case-ek javítása/megírása.
2020.04.05 19:00	1 óra	Kálmán Kuzsner Filep Jakabos	Értekezés: A Use-Case-ek és a bemeneti nyelv kapcsolatáról. Döntés: A különleges képesség Use-Case kettéválasztása és további nyolc Use-Case bevezetése.
2020.04.05 19:00	2 óra	Kuzsner	Tevékenység: Use Item Use-Case átalakítása, Use Rope, Use Food és Use Tent interface-ek megírása. A játékosok mozgását leíró Use-Case-ek átalakítása, a dokumentum ellenőrzése és formázása.
2020.04.05 20:00	1 óra	Puskás	Tevékenység: Tesztelési terv javítása, tesztelést támogató segéd- és fordítóprogramok specifikálása
2020.04.05 22:00	3 óra	Kálmán	Tevékenység: Bemeneti és kimeneti nyelv javítása és véglegesítése a megbeszéltek alapján, dokumentum formázása.

08-Részletes tervezet

47 – cereal_killers

Konzulens:

**Micskei Zoltán
Vörös András**

Csapattagok

Filep Leonárdó Márk	F6JQ0X	filep.leonardo@gmail.com
Kálmán Bendegúz	PTW6BD	kalmanbendeguz@gmail.com
Bence		
Jakabos Csenge Apolka	KO6MG	jakabos11@gmail.com
	0	
Puskás Levente	SN8YG9	puskas.levente99@gmail.co m
Kuzsner Sára	HVE3FV	kuzsnersara@gmail.com

2020.04.14.

8. Részletes tervezés

8.1 Osztályok és metódusok tervezése.

8.1.1 Charge

Felelősség

A patront valósítja meg, ami a rakéta egyik alkatrésze. Ha ez nincs a többi alkatrésszel ugyanazon a mezőn, a rakétát nem lehet összeszerelni.

Ősosztályok

Item

Interfészek

Nem valósít meg interfészt.

Attribútumok

-

Metódusok

- **public void Use(Player player)** : A patron használatát megvalósító függvény. Mivel a patront önmagában nem lehet használni, csak egy üzenetet ír ki a képernyőre, miszerint “charge cannot be used individually”.
- **public String GetName()** : Visszaadja a patron nevét (charge). Innen tudja a szereplő, hogy patron van nála.

8.1.2 PolarBear

- Felelősség

Egy lényt valósít meg, ami tud lépni.

- Ősosztályok

Nem leszármazott osztály.

- Interfészek

Moveable

- Attribútumok

- **Field currentField**: tárolja, hogy a lény melyik mezőn tartózkodik. Láthatóság: #

- Metódusok

- **void Move(int nextField)**: A paraméterben megadott irányban lévő szomszédos mezőre próbálja léptetni a lényt, a currentField PassPlayer függvényét hívja meg. Láthatósága: +

8.1.3 Controller

- **Felelősség**

Feladata a játék menetének és folyamatainak irányítása. Idetartozik a pálya elemeinek és a játékosoknak a létrehozása és elhelyezése, a GameLoop, valamint a hóvihar indítása és a jegesmedve mozgatása.

- **Ősosztályok**

Nem leszármazott osztály.

- **Interfészek**

Megvalósítja az EndOfGame interface-t.

- **Attribútumok**

- **Field fields:** A játék mezőinek tárolására szolgál. Láthatósága:-
- **Player players:** A játékosok tárolására szolgál. Láthatósága:-
- **Creature polarBear:** A Pályán barangoló Jegesmedvét tárolja. Láthatósága:-
- **Random random:** A hóviharok bekövetkezésének valószínűségéhez használt változó. Láthatósága:-
- **boolean ended:** A játék végét jelző változó, aminek a segítségével kilépünk a GameLoopból. Láthatósága:-

- **Metódusok**

- **void SnowStorm():** Egy új hóvihar 30% valószínűsséggel való indítására alkalmas függvény. Láthatósága:-
- **void Start():** A pálya elemeinek létrehozásáért, a játékosok létrehozásáért, és mindenek elhelyezéséért felelős függvény. Láthatósága:+
- **void Finish():** A Játék végekor meghívódó függvény, ennek segítségével tudunk kilépni a GameLoopból. Láthatósága:+
- **boolean ArePlayersTogether():** Visszaadja, hogy a játékosok egy mezőn vannak-e. Láhatósága:+
- **void GameLoop():** A játék menetéért felelős függvény. Ebben zajlik a játék körforgása: a jegesmedve elmozdul egy mezővel, a játékosok egyesével lépnek, majd adott valószínűsséggel jön egy hóvihar. Láthatósága:+

8.1.4 Moveable

Felelősség

A Move() függvényt megvalósító interfész, a szereplők és a jegesmedve mozgásában van szerepe.

Ősosztályok

Nem leszármazott osztály.

Interfészek

Nem valósít meg interfészt.

Attribútumok

- **Field currentField:** A mező ami jelenleg tartózkodik a szereplő/jegesmedve.

Metódusok

- **void Move(int directionIndex):** A paraméterként kapott index irányába mozog a szereplő vagy jegesmedve.

8.1.5 DivingSuit

Felelősség

A búvárruhát megvalósító osztály. Ha ezt viseli a szereplő, nem hal meg miután vízbe esett.

Ősosztályok

Item

Interfészek

Nem valósít meg interfészt.

Attribútumok

-

Metódusok

- **public void Use(Player player) :** A búvárruha használatakor ez a függvény hívódik. A használat itt azt jelenti, hogy a szereplő felveszi a ruhát, vagy ha rajta volt eddig, akkor leveszi.
- **public String GetName() :** A búvárruha nevével tér vissza (divingsuit). Innen tudja a szereplő, hogy búvárruha van nála.

8.1.6 Flare

Felelősség

A jelzőfényt megvalósító osztály. Ez a rakéta egyik alkatrészze, amíg nincs ugyanazon a mezőn mint a többi alkatrész, addig nem lehet összeszerelni a rakétát.

Ősosztályok

Item

Interfészek

Nem valósít meg interfészt.

Attribútumok

Metódusok

- **public void Use(Player player)** : A jelzőfény használatakor ez a függvény hívódik. Mivel a jelzőfényt önmagában nem lehet használni, ezért ekkor csak egy üzenet íródik ki a képernyőre: “flare cannot be used individually”.
- **public String GetName()** : A jelzőfény nevével tér vissza (flare) . Innen tudja a szereplő, hogy jelzőfény van nála.

8.1.7 Food

Felelősség

Az éleelmet megvalósító osztály. Az élelem használatakor a szereplő megeszi azt, és életpontja egygyel megnövekedik.

Ősosztályok

Item

Interfészek

Nem valósít meg interfészt.

Attribútumok

Metódusok

- **public void Use(Player player)** : Az étel használata az étel megevését jelenti. Ekkor a játékos életpontja egygyel növekedik, amennyiben még nincs a maximumon. Ezután az étel eltűnik, mivel el lett fogyasztva.
- **public String GetName()** : Az étel nevével tér vissza (food) . Innen tudja a szereplő, hogy étel van nála.

8.1.8 Gun

Felelősség

A pisztolyt megvalósító osztály. Ez a rakéta egyik alkatrésze, amíg nincs ugyanazon a mezőn mint a többi alkatrész, addig nem lehet összeszerelni a rakétát.

Ősosztályok

Item

Interfészek

Nem valósít meg interfészt.

Attribútumok

Metódusok

- **public void Use(Player player)** : A pisztoly használatakor ez a függvény hívódik. Mivel a pisztolyt önmagában nem lehet használni, ezért ekkor csak egy üzenet íródik ki a képernyőre: "gun cannot be used individually".
- **public String GetName()** : A pisztoly nevével tér vissza (gun) . Innen tudja a szereplő, hogy pisztoly van nála.

8.1.9 Item

Felelősség

A tárgyat megvalósító osztály, azoknak közös absztrakt alaposztálya. A szereplő ilyeneket tud tárolni az eszköztárában.

Ősosztályok

Nem leszármazott osztály.

Interfészek

Nem valósít meg interfést.

Attribútumok

Metódusok

- **public abstract void Use(Player player)** : A tárgy használatakor ez a függvény hívódik. Ez minden tárgynál másféleképpen van megvalósítva, ezért ebben az osztályban nincs implementációja, absztrakt.
- **public abstract String GetName()** : A tárgy nevével tér vissza. Innen tudja a szereplő, hogy milyen van nála. Absztrakt, mivel minden tárgynak más neve van.

8.1.10 Menu

• Felelősség

A játék menüt megvalósító osztály.

• Ősosztályok

Nem leszármazott osztály.

• Interfészek

Nem valósít meg interfést.

• Attribútumok

- **Options options:** Az Options osztály egy példánya a beállításokhoz. Láthatósága:-
- **Vector<ScoreData> highscores:** A toplistát megvalósító tároló. Láthatósága:-

- **Metódusok**

- **void ShowMenuItems()**: A menüben található alpontokat írja ki. Láthatósága: +
- **boolean ChooseMenuItem(MenuItem item)**: A paraméternek megfelelő alpontot kiválasztva a megadott menüponthoz tartozó tevékenységet hajtjuk végre.
Láthatósága: +
- **void NewGame()**: Új játék kezdésére szolgáló függvény. Láthatósága: +
- **void Settings()**: A beállításokkal foglalkozó függvény. Láthatósága: +
- **void ShowBestScores()** : A toplista megjelenítését végző függvény. Láthatósága: +
- **void Exit()** : A játékból való kilépésre szolgáló függvény. Láthatósága: +
- **void AddHighscore**: A toplistához adja a jelenlegi felhasználó által elért eredményt.
Láthatósága: +

8.1.11 Options

- **Felelősség**

A játékkal kapcsolatos beállításokat valósítja meg.

- **Ősosztályok**

Nem leszármazott osztály.

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **String playerName**: A felhasználó nevét tároló attribútum. Láthatósága: -
- **int playerNum**: A játékosok számát tároló attribútum. Láthatósága: -
- **boolean music**: A zene engedélyezésének állapotát tároló attribútum. Láthatósága: -
- **boolean determinism**: A determinizmus engedélyezésének állapotát tartalmazó attribútum. Láthatósága: -

- **Metódusok**

- **void ShowOptionItems()**: A beállításokban található alpontokat írja ki. Láthatósága: +
- **void ChooseOptionsItem(OptionsItem item)**: A paraméternek megfelelő alpontot kiválasztva módosítást végezünk a beállításokban szereplő egyik attribútumon.
Láthatósága: +
- **void PlayerNameOption()**: A függvény bekéri a felhasználó nevét és eltárolja azt.
Láthatósága: +
- **void SetPlayerName(String newName)**: A paraméterben megadott Stringre állítja a felhasználó nevét. Láthatósága: +
- **String GetPlayerName()**: Visszaadja a felhasználó nevét. Láthatósága: +
- **void PlayerNumberOption()**: A függvény a felhasználótól bekéri a játékosok számát és eltárolja azt. A játékosok száma 3 és 8 között mozoghat. Láthatósága: +
- **void SetPlayerNumber(int newValue)**: A paraméterben megadott intre állítja a játékosok számát. Láthatósága: +
- **int GetPlayerNumber()**: Visszaadja a játékosok számát. Láthatósága: +

- **void MusicOption():** A felhasználó on/off parancssal megadja, hogy szeretné-e hallani a játék zenéjét vagy sem. Láthatósága: +
- **void SetMusic(boolean newValue):** A paraméterben megadott értékre állítja a zene engedélyezésének értékét. Láthatósága: +
- **boolean GetMusic():** Visszaadja, hogy a zene engedélyezve van-e vagy sem. Láthatósága: +
- **void DeterminismOption():** A felhasználó on/off parancssal megadja, hogy szeretné-e bekapcsolni a véletlenszerűséget vagy sem. Láthatósága: +
- **void SetDeterminism(boolean newValue):** A paraméterben megadott értékre állítja a determinizmus értékét. Láthatósága: +
- **boolean GetDeterminism():** Visszaadja a determinizmus értékét. Láthatósága: +

8.1.12 Player

- **Felelősség**

A Player osztály reprezentál egy játékbeli szereplőt. Ez az objektum reagál a felhasználó interakcióira, azaz ő a játék aktív szereplője. Van egy eszköztára, amiben a kiásott tárgyakat tárolhatja. Tud mozogni, és dolgozni, azaz a tárgyait használni. Körökre osztva tevékenykedik, és minden körben adott számú energiája van, amely a munkáját korlátozza. Élet pontjai is vannak, amiket ha elveszít, meghal.

- **Ősosztályok**

Nem leszármazott osztály.

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

- **int health:** A szereplő élet pontjainak száma. Élelem megevésével lehet növelni. Ha elfogy, a játékos meghal. Láthatósága: #
- **int maxHealth:** A szereplő maximális élet pontjainak száma. Ennél több élet pontja nem lehet a szereplőnek. Láthatósága: #
- **int energy:** A szereplő energiapontjainak száma az adott körben. Ez az attribútum a munkavégző-képességét számosítja. minden körben 4-et használhat belőle, ha elfogy, cselekvőképtelen lesz. Ha nem használja fel az összeset egy körben, akkor a megmaradt energiát nem viheti tovább a következő körre. Láthatósága: #
- **boolean wears_suit:** Megmondja, hogy a szereplő rendelkezik-e bűvárruhával. Ha igen, akkor minden esetben túléli a vízbe zuhanást. Láthatósága: #
- **ArrayList<Item> items:** A karakter eszköztára, maximum 5 tárgyat tud ebben tárolni. Láthatósága: #
- **Field currentField:** Az a mező, amin a karakter aktuálisan áll. Láthatósága: #
- **Controller controller:** A játék irányítóegysége. Láthatósága: #

- **Metódusok**

- **void Move(int nextField):** A paraméterben megadott irányban lévő szomszédos mezőre próbálja léptetni a lényt, a currentField PassPlayer függvényét hívja meg (ha van elég energiája), ami true-t térít vissza, ha sikeres a mozgás, ekkor pedig meghívja a DecrementEnergy() függvényt. Láthatósága: +

- **void Turn()**: Ez a függvény a szereplő egy turnusát reprezentálja. A soron levő szereplőnek kiadott felhasználói parancsokat dolgozza fel és meghívja a megfelelő függvényeket. Mindaddig ciklusban marad, amíg a felhasználó a turnus végét nem jelzi. Láthatósága: +
- **void EndTurn()**: Ez a függvény akkor hívódik, amikor a felhasználó turnus végét jelez. Csökkenti a játékos életpontjait, ha vízben van.
- **void PickItemUp()**: A szereplő ennek hatására veszi fel az aktuális mezőn lévő legelső tárgyat, amennyiben az ki van ásva, a szereplőnek van elég energiája. A **Field** osztály **PopItem** segítségével kiveszi a tárgyat a currentField tárolójából és az **AddItem** segítségével hozzáadja a saját eszköztárához. Csökkenti az energiát eggyel. Láthatóság: +
- **void AddItem(Item item)**: A paraméterben megadott tárgyat hozzáadja az szereplő eszköztárához, ha maradt még hely.
- **int SearchItem(String input)**: Megkeresi, hogy van-e a paraméterben megadott tárgy a szereplő eszköztárában és ha igen visszatér az indexével, egyébként -1-el tér vissza. Láthatóság: +
- **void DropItem(String input)**: Először megnézi a SearchItem segítségével, hogy a paraméterben kapott tárgynév megfeleltethető-e egy az eszköztárban található tárgyal, és ha talált ilyet akkor átadja ezt a vele megegyező nevű, de más paraméter listájú függvénynek. Láthatóság: +
- **void DropItem(Item item)**: A szereplő kirak egy a paraméterben megadott tárgyat, lerakja a mezőre és ha a mezőn található legelső tárgy megegyezik a lerakottal, akkor kiveszi az eszköztárából is. Itt is ellenőrzi, hogy van-e elég energiája a tevékenységehez. Csökkenti az energiát eggyel. Láthatóság: +
- **void RemoveItem(Item item)**: Az paraméterben megadott tárgyat távolítja el az eszköztárból. Láthatóság: +
- **void UseItem(String input)**: Először megnézi a SearchItem segítségével, hogy a paraméterben kapott tárgynév megfeleltethető-e egy az eszköztárban található tárgyal, és ha igen akkor átadja a kapott tárgyat az ugyanolyan nevű, de más paraméterlistájú UseItem függvénynek. Láthatóság: +
- **void RemoveSnow()**: Eltávolít egy egységnyi a havat az *currentField*-ről és csökkenti az energiát eggyel. Láthatóság: +
- **void ListItems()**: Kilistázza a szereplő eszköztárát. Láthatóság: +
- **boolean GetWearsSuit()**: A *wears_suit* attribútum getter függvénye. Láthatóság: +
- **void SetWearsSuit(boolean value)**: A *wears_suit* attribútum setter függvénye. Láthatóság: +
- **Field GetCurrentField()**: A *currentField* attribútum getter függvénye. Láthatóság: +
- **void SetCurrentField(Field nextField)**: A *currentField* attribútum setter függvénye. Láthatóság: +
- **int GetHealth()**: A *health* attribútum getter függvénye. Láthatóság: +
- **void SetHealth(int value)**: A *health* attribútum setter függvénye. Láthatóság: +
- **void DecrementHealth()**: Csökkent a *health* attribútum értékét 1-el. Láthatóság: +
- **void IncrementHealth()**: Növeli a *health* attribútum értékét 1-el. Láthatóság: +
- **int GetMaxHealth()**: A *maxHealth* attribútum getter függvénye. Láthatóság: +
- **int GetEnergy()**: Az *energy* attribútum getter függvénye. Láthatóság: +
- **void SetEnergy(int value)**: Az *energy* attribútum setter függvénye. Láthatóság: +
- **void DecrementEnergy()**: Csökkenti az *energy* attribútum értékét 1-el. Láthatóság: +

8.1.13 Rocket

- **Felelősség**

A rakétát megvalósító osztály. Ez egy olyan objektum, ami tartalmaz (kompozíció) egy pisztolyt (Gun), jelzőfényt (Flare), és egy patron (Charge). Ezt kell a játékosoknak összeszerelni és elsütni a gyözelemhez. Csak akkor szerelhetik össze, ha a játékosok és az alkatrészek is ugyanazon a mezőn vannak. A rakéta felelőssége ezt megvizsgálni, valamint elsütés után kiváltani a játék befejeződését.

- **Ősosztályok**

Nem leszármazott osztály.

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

-isOnOneField : boolean : A rakéta osztály ebben a logikai változóban tartja nyilván, hogy az alkatrészei egy mezőn helyezkednek-e el.

-isAssembled: boolean : Azt jelzi, hogy a rakéta össze van-e szerelve. Ha igen, akkor el lehet sütni. Ha össze van szerelve, akkor már nem lehet mozgatni az alkatrészeit.

-gun: Gun : A pisztoly.

-flare: Flare : A jelzőfény.

-charge: Charge : A patron.

- **Metódusok**

- **public void Assemble()**: Ennek a meghívására összeszerelődik a rakéta, de előtte megvizsgálja, hogy az alkatrészei egy mezőn vannak-e, valamint lekérdezi a kontrollertől, hogy a szereplők is ugyanazon a mezőn vannak-e mindenki. Azt is megvizsgálja, hogy a meghívó szereplőnek van-e elég energiája az összeszereléshez.
- **public void Fire()**: Ez a függvény süti el a rakétát, amennyiben a meghívójának van elég energiája, és a rakéta össze van szerelve. A játékosoknak ekkor is ugyanazon a mezőn kell lenniük.

8.1.14 Rope

Felelősség

A kötelet megvalósító osztály. Ez egy tárgy, amit a szereplő megtalálhat és elhelyezheti az eszköztárában. Felelőssége, hogy használatkor a szomszédos mezőn lévő vízbe esett szereplőket kimentse a vízből, a használója jeigtáblájára.

Ősosztályok

Item

Interfészek

Nem valósít meg interfézst.

Attribútumok

-

Metódusok

- **public void Use(Player player)** : A kötél használatakor ez a függvény hívódik. Hatására a kötél használójának mezőjével szomszédos mezőkön lévő vízbe esett szereplők átkerülnek a használó szereplő mezőjére. A kötél használata egy munkába kerül.
- **public String GetName()** : A kötél nevével tér vissza (rope) . Innen tudja a szereplő, hogy kötél van nála.

8.1.15 Scientist

• Felelősség

A sarkkutató egy játékbeli szereplő. Különleges képessége, hogy meg tudja nézni, hogy az a jágtábla amire lépne, hány embert bír el. Gyengesége, hogy csak 4 életpontja van.

• Ősosztályok

Player

• Interfészek

Nem valósít meg interfézst.

• Attribútumok

Megegyeznek az Ősosztályéval.

• Metódusok

Ugyanazokkal a metódusokkal rendelkezik, mint az Ős Player osztály, egy kivételével.

- **void InspectField(Field field)**: Megállapítja egy szomszédos mező stabilitását, vagyis hogy hány szereplőt bír el. Ezt az értéket kiírja a kimenetre.

8.1.16 ScoreData

• Felelősség

A toplista egy elemét megvalósító osztály.

• Ősosztályok

Nem leszármazott osztály.

• Interfészek

Nem valósít meg interfézst.

• Attribútumok

- **String name**: A felhasználó nevét tároló attribútum. Láthatósága:-

- **int score:** A felhasználó pontszámát tároló attribútum. Láthatósága:-

8.1.17 Shovel

- **Felelősség**

Az lapátot megvalósító osztály. Ez egy tárgy, aminek használatával egy energiával két egység havat takaríthatunk el a mezőről. Lehet lapát, vagy törékeny ásó.

- **Ősosztályok**

Item

- **Interfészek**

Nem valósít meg interfészt.

- **Attribútumok**

-

- **Metódusok**

- **public void Use(Player player) :** Az ásó használatakor ez a függvény hívódik meg. Amennyiben a szereplőnek van legalább egy energiája, akkor a mezőről két egység hó eltűnik, feltéve hogy legalább két egység hó volt a mezőn, ha nem akkor értelemszerűen kevesebb tűnik el.
- **public String GetName():** Visszaadja az lapát nevét (shovel), innen tudja a szereplő, hogy lapát van nála.

8.1.18 Spade

Felelősség

A törékeny ásót megvalósító osztály. Ez is egy eszköz, ugyanúgy működik, mint a lapát, csak nem tartós, három használat után eltörök, azaz eltűnik a játékos eszköztárából.

Ősosztályok

Item, Shovel

Interfészek

Nem valósít meg interfészt.

Attribútumok

-health : int : Az ásó állapota, kezdetben ez az érték 3. Ha 0-ra csökken, az ásó eltörök, azaz megszűnik létezni.

Metódusok

- **public void Use(Player player) :** A törékeny ásó használatakor ez a függvény hívódik. Hatására a szereplő az adott mezőről 2 havat tud egy munkával eltakarítani, vagy kevesebbet, ha a mezőn nincs 2 egység hó. Használatkor az életpont eggyel csökken. Ha egy életpontja van, és használjuk, lecsökken 0-ra az életpont, ekkor az ásó eltűnik.

- **public String GetName()** : A törékeny ásó nevével tér vissza (spade) . Innen tudja a szereplő, hogy törékeny ásó van nála.

8.1.19 Tent

Felelősség

A sátor megvalósító osztály. Ez is egy tárgynak számít, amit ki lehet ásni. Hasonlóan viselkedik, mint az iglu, azaz véd a hóvihartól, viszont még az igluba nem tud bemenni a jégesmedve, a sátorba igen. Különbség még, hogy a sátor a kör végén eltűnik.

Ősosztályok

Item

Interfészek

Nem valósít meg interfészt.

Attribútumok

-

Metódusok

- **public void Use(Player player)** : A sátor használatakor ez a függvény hívódik. A sátor használata a sátor felállítását jelenti egy munka árán. Csak stabil, hómentes mezőre lehet sárat rakni.
- **public String GetName()** : A sátor nevével tér vissza (tent) . Innen tudja a szereplő, hogy sátor van nála.

8.1.20 EndOfGame

• Felelősség

A játék Finish() függvényét tartalmazó interface. Feladata, hogy a Finish függvényt egy interfacen keresztül érhesse el a többi osztály.

• Ősosztályok

Nem leszármazott osztály.

• Interfészek

Nem valósít meg interfészt.

• Metódusok

- **void Finish()**: Átállítja a játék végét jelző változót, aminek a segítségével kilépünk a GameLoopból. Láthatósága:+

8.1.21 Eskimo

• Felelősség

Az Eskimo osztály egy játékbeli szereplő, az eszkimó. Különleges képessége, hogy tud iglut építeni, amelyben átvésselhetőek a hóviharok. A játékot 5 életponttal kezdi.

- **Ősosztályok**

Player

- **Interfészek**

Nem valósít meg interfézszt.

- **Attribútumok**

Megegyeznek az ősosztályéval.

- **Metódusok**

Ugyanazokkal a metódusokkal rendelkezik, mint az ős Player osztály, egy kivételével.

- **void BuildIgloo():** Az eszkimó iglут épít, ehhez egy munkát fel kell használnia. Csak stabil mezőre tud iglut építeni. Láthatósága: +

8.1.22 Field

- **Felelősség**

A Field egy mezőt jelképes amire a szereplők léphetnek, szerepe van a pálya felépítésében illetve a szereplők mozgásában. A mezőkön találhatóak egyes felvehető tárgyak vagy például építhető iglu vagy sátor is.

- **Ősosztályok**

Nem leszármazott osztály.

- **Interfészek**

Nem valósít meg interfézszt.

- **Attribútumok**

- **int maxWeight:** A mező teherbírása. Ha ez az érték nagyobb mint a szereplők száma plusz egy, akkor a mező stabilnak számít. Láthatóság: -
- **int snowCount:** A mezőn levő hóegységek száma. Láthatóság: -
- **boolean hasIgloo:** Tárolja, hogy az adott mezőn épült-e iglu. Láthatóság: -
- **boolean hasTent:** Tárolja, hogy az adott mezőn épült-e sátor. Láthatóság: -
- **boolean isUpsideDown:** Értéke true, ha a mező átfordult mert a szereplők száma meghaladta a maximális teherbírást. Láthatóság: -
- **ArrayList<Item> items:** A mezőn található eszközök listája. Láthatóság: -
- **ArrayList<Player> players:** A mezőn álló játékosok listája. Láthatóság: -
- **Creature polarBear:** Ha értéke nem null, akkor jelenleg egy jegesmedve (is) áll a mezőn. Láthatóság: -
- **ArrayList<Field> neighbors:** A mező szomszédait tároló lista. Láthatóság: -

- **Metódusok**

- **void AddNeighbor(Field neighbor):** Hozzáad egy mezőt a szomszédok listájához. Láthatóság: +
- **Field GetNeighbor(int neighborIndex):** Visszaadja a paraméterként kapott indexsel rendelkező szomszéd mezőt vagy null-t. Láthatóság: +

- **boolean PassPlayer(int directionIndex, Player player):** A szereplők mozgásában van szerepe, átrakja a directionIndex paraméter irányában levő mezőre player paramétert, ekkor visszatérési értéke true, ellenkező esetben, ha például nincs ilyen szomszéd akkor false. Láthatóság: +
- **Creature GetPolarBear():** Visszaadja a mezőn levő jegesmedvét képviselő Creature objektumot, vagy null-t ha jelenleg nincs jegesmedve a mezőn. Láthatóság: +
- **void SetPolarBear(Creature pb):** A polarBear attribútum setttere. Láthatóság: +
- **boolean Accept(Player player):** Értéke true, ha a paraméterként kapott játékos a mezőre került (sikeresen hozzáadtuk a players listához) illetve false ellenkező esetben. Láthatóság: +
- **void Remove(Player player):** Eltávolítja a paraméterként kapott játékost a mezőről. Láthatóság: +
- **void PushItem(Item item):** A mezőre kerül a paraméterként kapott eszköz. Láthatóság: +
- **Item PopItem():** Visszaadja a legfelső elemet az items listából és törli. Láthatóság: +
- **Item GetItem():** Visszaadja a legfelső elemet az items listából. Láthatóság: +
- **boolean IsOverWeight():** Ha túl sok játékos van a mezőn, beállítja az isUpsideDown értékét true-ra. Visszatérési értéke true ha a mező átfordult. Láthatóság: +
- **ArrayList<Player> GetPlayers():** A players attribútum gettere. Láthatóság: +
- **ArrayList<Field> GetNeighbors():** A neighbors attribútum gettere. Láthatóság: +
- **int GetMaxWeight():** A maxWeight attribútum gettere. Láthatóság: +
- **int GetSnow():** A snowCount attribútum gettere. Láthatóság: +
- **void SetSnow(int value):** A snowCount attribútum setttere. Láthatóság: +
- **void DecrementSnow():** Növeli a snowCount értékét egy egységgel. Láthatóság: +
- **void IncrementSnow():** Csökkenti a snowCount értékét egy egységgel. Láthatóság: +
- **void SetHasIgloo(boolean value):** A hasIgloo attribútum gettere. Láthatóság: +
- **boolean GetUpsideDown():** Az isUpsideDown paraméter gettere. Láthatóság: +
- **void ListItems():** Kiírja a szabványos kimenetre a mezőn levő tárgyak nevét. Láthatóság: +

8.2 A tesztek részletes tervez, leírásuk a teszt nyelvén

8.2.1 Player steps on Field

- Leírás

A játékos egy szomszédos mezőre lép.

- Ellenőrzött funkcionalitás, várható hibahelyek

Hibalehetőség lehet ha átfordul egy mező annak ellenére hogy el kellene bírnia a szereplőt, ha nem kerül át a jelenlegi (currentField) mezőről a következőre (nextField) vagy ha átkerül, de a régről nem távolítja el a Remove() függvény és a szereplő két mezőn egyszerre jelenik meg. Hiba lehet az is, ha a szereplő energiája nem csökken.

- Bemenet

move 1

- Elvárt kimenet

remaining energy: 2

8.2.2 Player steps on hole

- **Leírás**

A játékos egy lyukba esik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hiba lehet, ha a szereplő nem esik a lyukba vagy ha nem csökken az energiája.

- **Bemenet**

move 1

- **Elvárt kimenet**

player fell into hole

remaining energy: <N>

8.2.3 Player steps on side of map

- **Leírás**

A játékos a pálya szélén túlra próbálja irányítani a szereplőt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A szereplő átkerül a nem létező szomszédra.

- **Bemenet**

move 1

- **Elvárt kimenet**

no neighbor in direction <directionIndex>

8.2.4 Player steps on unstable Field that flips

- **Leírás**

A szereplő olyan mezőre lép, ami instabil és a teherbírása kisebb mint a jelenleg rajta álló szereplők száma. Ekkor a mező átfordul, a szereplő meghal és vége a játéknak.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hibalehetőség lehet ha a jégtábla nem fordul át a szereplők súlya alatt vagy ha a szereplő(k) nem hal(nak) meg és nem lesz vége a játéknak.

- **Bemenet**

move 1

- **Elvárt kimenet**

Field flipped

Game over

8.2.5 Player steps in Igloo

- **Leírás**

Lemodellezük azt a folyamatot, ahogy a játékos olyan instabil mezőre lép, amin iglu van.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos instabil iglut tartalmazó mezőre lép ami iglu nélkül már felborulna, ellenőrizzük, hogy a mező tud-e a rajta található igluról, megvédi-e a játékost a vízbeborulástól.

Hiba lehetőségek: Az Eskimo BuildIgloo() függvényében a mező nem értesül a ráépült igluról, a program nem ellenőrzi le a mező állapotát amikor borul.

- **Bemenet**

move 1

- **Elvárt kimenet**

the field has an igloo
remaining energy: 3

8.2.6 Player steps in Tent

- **Leírás**

Lemodellezük azt a folyamatot, ahogy a játékos olyan mezőre lép, amin sátor van.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos instabil, sátrat tartalmazó mezőre lép ami sátor nélkül már felborulna, ellenőrizzük, hogy a mező tud-e a rajta található sátorról, megvédi-e a játékost a vízbeborulástól.

Hiba lehetőségek: Az Player BuildTent() függvényében a mező nem értesül a ráépült sátorról, a program nem ellenörzi le a mező állapotát amikor borul.

- **Bemenet**

move 1

- **Elvárt kimenet**

the field has a tent
remaining energy: 3

8.2.7 Player steps on Field with polar bear on it

- **Leírás**

A szereplő egy olyan mezőre lép, amin jegesmedve van.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Leellenőrizzük azt a folyamatot, amikor a szereplő rálép egy olyan mezőre, ahol jegesmedve van, azaz a polarbear változó nem null. Ekkor a szereplő meghal és a játék véget ér.

- **Bemenet**

move 2

- **Elvárt kimenet**

remaining energy: 3
polarbear on field
game over

8.2.8 Polar bear steps on Field with Player on it

- **Leírás**

A jegesmedve egy olyan mezőre lép, amin egy szereplő van, és a mezőn nincs iglu.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Leteszteljük azt a folyamatot, ahogy a jegesmedve olyan mezőre lép, amin szereplő van, de a szereplőt nem védi iglu. Az a cél, hogy ilyenkor a szereplő meghaljon, és a csapat veszítsen, azaz a Finish() meghívódjon. Lehetséges hibalehetőség, ha ilyenkor a szereplő túléli, vagy a jegesmedve nem mozdul.

- **Bemenet**

move 3

endturn

- **Elvárt kimenet**

polarbear moved to <field>
polarbear on field

game over

8.2.9 Polar bear steps on Field with Igloo on it

- **Leírás**

A jegesmedve egy olyan mezőre lép, amin iglu van, és az igluban egy szereplő van.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Leteszteljük azt a folyamatot, ahogy a jegesmedve olyan mezőre lép, amin iglu van, és az igluban szereplő. Az a cél, hogy ilyenkor a szereplő ne haljon meg, és a csapat ne veszítsen. Lehetséges hibalehetőség, ha ilyenkor a szereplő meghal, vagy a jegesmedve nem mozdul.

- **Bemenet**

use tent

endturn

- **Elvárt kimenet**

tent set

polarbear moved to <field>

polarbear on field

game over

8.2.10 Polar bear steps on Field with Tent on it

- **Leírás**

A jegesmedve egy olyan mezőre lép, amin sátor van, és a sátorban a mezőn egy szereplő van. Fontos, hogy a mezőn nem lehet iglu.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Leteszteljük azt a folyamatot, ahogy a jegesmedve olyan mezőre lép, amin sátor van, és a sátorban szereplő. Az a cél, hogy ilyenkor a szereplő meghaljon, és a csapat veszítsen. Lehetséges hibalehetőség, ha ilyenkor a szereplő nem hal meg, vagy a jegesmedve nem mozdul.

- **Bemenet**

move 3

endturn

- **Elvárt kimenet**

the field has an igloo

polarbear moved to <field>

<player n+1>'s turn

8.2.11 Player saves Player

- **Leírás**

Egy játékos kihúz egy másik játékost a vízből.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Leellenőrizzük azt a folyamatot, ahogy egy játékos kiment egy másik játékost vízből kötél segítségével. Hibás lehet: a kötél használatához kapcsolódó függvények: UseItem (lásd korábbi szekvenciadiagramok).

- **Bemenet**

use rope

- **Elvárt kimenet**

1 players saved

remaining energy: 3

8.2.12 Snowstorm hits Field

- **Leírás**

A mező hómennyisége egygyel nő hóvihar után.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Leellenőrzük azt a folyamatot, ahogy a hóvihar lesújt egy mezőre és annak hómennyisége egygyel nő. Hibás lehet: a Controller SnowStorm() függvénye vagy Field osztály IncrementSnow() metódusa.

- **Bemenet**

Megj. utolsó játékos van soron
endturn

- **Elvárt kimenet**

snowstorm happened
player 1's health = 4
player 2's health = 4
player 1's turn

8.2.13 Snowstorm hits player

- **Leírás**

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

Megj. utolsó játékos van soron
endturn

- **Elvárt kimenet**

snowstorm happened
player 1's health = 3
player 2's health = 4
player 1's turn

8.2.14 Eskimo builds an igloo

- **Leírás**

Lemodellezük azt a folyamatot, ahogy egy eszkimó iglut épít.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az Eskimo létrehoz egy iglut, azon a mezőn, amelyiken éppen tartózkodik, ellenőrzük, hogy az iglu megépül-e és a field értesül-e róla.

Hibás lehet: Az Eskimo BuildIgloo() függvénye.

- **Bemenet**

igloo

- **Elvárt kimenet**

igloo built
remaining energy: 3

8.2.15 Scientist inspects Ice Float

- **Leírás**

A kutató megállapítja egy szomszédos mező kapacitását.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Leellenőrizzük azt a folyamatot, ahogy egy kutató megállapítja egy szomszédos mező kapacitását. Hibás lehet: Scientist InspectField függvénye, Field osztály GetMaxWeight függvénye.

- **Bemenet**

inspect 1

- **Elvárt kimenet**

stability: 3

remaining energy: 3

8.2.16 Player Picks Item Up

- **Leírás**

A játékos felveszi a jelenlegi mezőn lévő legfelső tárgyat amennyiben még van hely az eszköztárában.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a játékos felveszi a jelenlegi mezőn lévő legfelső tárgyat amennyiben még van hely az eszköztárában..Hibás lehet: Player osztály PickItemUp(), AddItem(Item item) függvényei és a Field osztály GetItem(), PopItem() függvényei.

- **Bemenet**

pickup

- **Elvárt kimenet**

<item> picked up

vagy

inventory full

8.2.17 Player Drops Item

- **Leírás**

A játékos letesz egy tárgyat a jelenlegi mezőre, amennyiben a mező teljesen letakarított.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a játékos letesz egy tárgyat a jelenlegi mezőre, amennyiben a mező teljesen letakarított. Hibás lehet: Player DropItem(Item item), RemoveItem(Item item)függvényei és a Field GetItem(), PushItem(Item item) függvényei.

- **Bemenet**

drop <item>

- **Elvárt kimenet**

<item> dropped

vagy

field not clear

8.2.18 Player Eats

- **Leírás**

A játékos megeszik egy ételt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Leellenőrizzük, hogy a játékos megeszik egy ételt, ezzel egy plusz életerőre tesz szert, amennyiben az életereje még nem érte el a maximum szintet. Hibás lehet: Player osztály UseItem(Item item) függvénye, és a Food osztály Use() függvénye.

- **Bemenet**

- use food
- **Elvárt kimenet**
health = 4
remaining energy: 3

8.2.19 Player Digs

- **Leírás**
A játékos eltakarít egy egységnyi havat arról a mezőről amin éppen rajta áll.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizzük, hogy a játékos eltakarít egységnyi havat arról a mezőről amin éppen rajta áll. Hibás lehet Player osztály RemoveSnow() függvénye, Field osztály DecrementSnow() függvénye.
- **Bemenet**
dig
- **Elvárt kimenet**
remaining energy: 3
1 snow removed
vagy
0 snow removed

8.2.20 Player Removes Snow With Shovel

- **Leírás**
A játékos eltakarít két egységnyi havat arról a mezőről amin éppen rajta áll lapátot használva.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizzük, hogy a játékos két eltakarít egységnyi havat arról a mezőről amin éppen rajta áll lapátot használva. Hibás lehet Player osztály UseItem(Item item) függvénye, Shovel osztály Use() függvénye.
- **Bemenet**
use shovel
- **Elvárt kimenet**
remaining energy: 3
2/1 snow removed
vagy
0 snow removed

8.2.21 Player Removes Snow With Spade

- **Leírás**
A játékos eltakarít két egységnyi havat arról a mezőről amin éppen rajta áll ásót használva.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizzük, hogy a játékos két eltakarít egységnyi havat arról a mezőről amin éppen rajta áll ásót használva. Hibás lehet Player osztály UseItem(Item item) függvénye, Spade osztály Use() függvénye.
- **Bemenet**
use spade
- **Elvárt kimenet**
remaining energy: 3

spade condition: 2
 2/1 snow removed
 vagy
 0 snow removed

8.2.22 Player wears diving suit

- **Leírás**
 A játékos felvesz egy búvárruhát.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
 Leellenőrzük azt a folyamatot, ahogy a játékos felvesz egy búvárruhát. Hibás lehet: Player UseItem() függvénye, wears_suit attribútum setttere.
- **Bemenet**
 use divingsuit
- **Elvárt kimenet**
 divingsuit on
 remaining energy: 3

8.2.23 Player takes off diving suit

- **Leírás**
 A játékos leveszi magáról a búvárruhát.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
 Leellenőrzük azt a folyamatot, ahogy a játékos leveszi magáról a búvárruhát. Hibás lehet: Player UseItem() függvénye, wears_suit attribútum setttere.
- **Bemenet**
 use divingsuit
- **Elvárt kimenet**
 divingsuit off
 remaining energy: 3

8.2.24 Not enough energy for moving

- **Leírás**
 Leteszteljük azt a folyamatot, amikor a szereplő lépni próbál, de nincs elég energiája hozzá.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
 Feltezzük, hogy a szereplők elfogyott az adott turnusra az energiája, és ezután lépni próbálunk vele. Az a cél, hogy ilyenkor ne tudjon mozogni a szereplő. Hibalehetőség, ha a szereplő mégis mozog, vagy olyan, mozgáshoz kapcsolódó események történnek, amik mozgás nélkül nem történhetnének meg, pl. vízbe esés, igluba bemenés, jegesmedvével találkozás.
- **Bemenet**
 move 4
- **Elvárt kimenet**
 not enough energy

8.2.25 Not enough energy for using item

- **Leírás**
 Leteszteljük azt a folyamatot, amikor a szereplő egy tárgyat próbál használni, de nincs elég energiája hozzá.
- **Ellenőrzött funkcionalitás, várható hibahelyek**

Feltesszük, hogy a szereplőnek elfogyott az adott turnusra az energiája, és ezután használni próbálunk egy tárgyat. Az a cél, hogy ilyenkor ne tudja ezt a tárgyat a szereplő használni. Hibalehetőség, ha a játékos mégis tudja használni a tárgyat.

- **Bemenet**
use shovel
- **Elvárt kimenet**
not enough energy

8.2.26 Player signals end turn

- **Leírás**
Lemodellezük azt a folyamatot, ahogy a játékos kör végét jelez.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A játékos turnus végét jelez, a következő játékos jön. Hiba a Player osztály EndOfTurn() függvényében lehet.
- **Bemenet**
endturn
- **Elvárt kimenet**
player<n+1>'s turn (<eskimo/scientist>)

8.2.27 Spade breaks after using it

- **Leírás**
Lemodellezük azt a folyamatot, amikor is a törékeny ásó használat után eltörök, mivel már háromszor volt használva.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A tesztben háromszor használunk egy törékeny ásót, majd megnézzük, hogy eltörök-e. Lehetséges hiba, ha nem törik el, vagy nem tűnik el a szereplő eszköztárából, miután eltört.
- **Bemenet**
use spade
use spade
use spade
use spade
- **Elvárt kimenet**
2 snow removed
remaining energy 3
spade condition : 2
2 snow removed
remaining energy 2
spade condition : 1
2 snow removed
remaining energy 2
spade condition : 1
spade broke
player doesn't have this item

8.2.28 Tent disappears at the start of a new turn

- **Leírás**

Tent disappears at the start of a new turn.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Lemodellezük azt a folyamatot, amikor is az előző körben egy mezőre felhúzott sátor új kör elején eltűnik. Hiba lehet a Player UseItem(Item item) és Turn() függvényeiben.

- **Bemenet**

use tent

- **Elvárt kimenet**

tent set

remaining energy: 3

8.2.29 Players assemble and fire Rocket

- **Leírás**

A szereplők összeszerelik, és elsütik a rakétát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Leteszteljük, ahogyan a szereplők egy helyre viszik a rakéta alkatrészeit, majd összeszerelik, majd elsütik, és a játék győzelemmel véget ér. Hibalehetőségek: a rakéta összeszerelődik úgy, hogy az alkatrészei nincsenek egy mezőre lerakva, esetleg a szereplő eszköztárában van valamelyik ahelyett, hogy a mezőn lenne letéve. Vagy rakéta úgy szerelődik össze, hogy a játékosok nincsenek ugyanazon a mezőn, és az is hiba lehet, ha az összeszerelőnek nincs elég energiája, de mégis összeszereli a rakétát. További hiba, ha az elsütés után nem ér véget a játék.

- **Bemenet**

(kezdő állapot: 3 játékos 1-1-1 alkatréssel a lerakási mezővel szomszédos mezőkön helyezkednek el)

move 3

drop charge

endturn

move 1

drop gun

endturn

move 3

drop flare

endturn

assemble

fire

- **Elvárt kimenet**

remaining energy: 3

charge dropped

player2's turn

remaining energy: 3

gun dropped

player3's turn

remaining energy: 3

flare dropped

player1's turn

rocket assembled

rocket fired

victory

8.2.30 Snowstorm doesn't affect Players in Igloo

- Leírás

A játékban kialakuló hóvihar lesújt a mezőre, de a mezőn levő szereplők igluban vannak. Ekkor a testhőjük nem csökken. A teszben iglut építünk az Eskimo szereplővel. Bemegy a szereplő, bekövetkezik a hóvihar és azt várjuk, hogy nem csökken az eszkimó testhője.

- Ellenőrzött funkcionalitás, várható hibahelyek

Hiba, ha a szereplő testhője csökken, vagy ha a nem stabil mezőre iglu épül.

- Bemenet

igloo

- Elvárt kimenet

igloo built

snowstorm happened

Ahol N a maradék energia és a hóvihar után azt várjuk hogy nem jelenik meg a “remaining health of Eskimo: <N>” kimenet.

8.2.31 Snowstorm doesn't affect Players in Tent

- Leírás

A játékban kialakuló hóvihar lesújt a mezőre, de a mezőn levő szereplők sátorban vannak. Ekkor a testhőjük nem csökken. A teszben sátrat építünk egy Player szereplővel. Bemegy a szereplő, bekövetkezik a hóvihar és azt várjuk, hogy nem csökken az eszkimó testhője.

- Ellenőrzött funkcionalitás, várható hibahelyek

Hiba, ha a szereplők testhője csökken.

- Bemenet

use tent

- Elvárt kimenet

tent set

snowstorm happened

8.2.32 ELŐÍRT TESZTESET: Mozog a medve és megeszik valakit, a játékosok csak fel-le mászkálnak

- Leírás

A tárgyfelelősök által adott kezdeti konfigurációval lemodellezük és teszteljük azt a folyamatot, hogy a medve mozog és megeszik valakit, miközben a szereplők csak fel-le mászkálnak.

- Ellenőrzött funkcionalitás, várható hibahelyek

Azt vizsgáljuk, hogy minden a játék funkcionalitása és logikai rendje szerint halad abban az esetben, ha innen indítva előidézünk egy olyan esetet, hogy a jegesmedve megeszik valakit. Hibalehetőség lehet, ha a játékosok mozgásán kívül bármilyen más történik velük (hiszen ha például felvesznek egy tárgyat, azzal megsérlik a teszt követelményét), illetve ha a jegesmedve nem mozdul, vagy ha egy mezőre kerül egy szereplővel, akkor nem ér véget a játék.

- Bemenet

(ebben a teszben az ábrán fentebb elhelyezkedő eszkimó kezd)

move 3 (ezzel a 2-es rakétaalkatrész mezőjére lép)

endturn

- **Elvárt kimenet**

remaining energy: 3

polarbear moved to <field>

polarbear on field

game over

8.2.33 ELŐÍRT TESZTESET: A játékosok törékeny táblára mennek és az eltörök

- **Leírás**

A tárgyfelelősök által adott kezdeti konfigurációval lemodellezük és teszteljük azt a folyamatot, hogy a szereplők egy törékeny mezőre annyian mennek, hogy az már nem bírja el őket, ezért átfordul, és ők a vízbe esnek.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt vizsgáljuk, hogy minden a játék funkcionalitása és logikai rendje szerint halad abban az esetben, ha innen indítva előidézünk egy olyan esetet, hogy a szereplők belesnek a vízbe az instabil mező átfordulása miatt. Hibalehetőség, ha valamelyik szereplő idő előtt meghal, vagy a mező esetleg nem fordul át, vagy nem az összes szereplő esik bele a vízbe.

- **Bemenet**

(ebben a tesztnél az ábrán fentebb elhelyezkedő eszkimó kezd)

move 1 (ezzel a lenti kutató mezőjére mezőjére lép)

move 2 (ezzel a következő kutató instabil mezőjére lép)

endturn (most a fentebbi kutató jön)

move 2

- **Elvárt kimenet**

remaining energy: 3

remaining energy: 2

player2's turn (scientist)

remaining energy: 3

field flipped

game over

8.2.34 ELŐÍRT TESZTESET: Eszkimó iglut épít

- **Leírás**

A tárgyfelelősök által adott kezdeti konfigurációval lemodellezük és teszteljük azt a folyamatot, ahogyan egy eszkimó felépít egy iglut.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Azt vizsgáljuk, hogy a kiadott parancsokra pontosan az elvárt kimenetek történnek-e, amik jól tükrözik a belső mechanizmust. Ellenőrizzük, hogy az iglu valóban felépült-e. Hibalehetőség, ha az iglu nem épül fel.

- **Bemenet**

(ebben a tesztnél az ábrán fentebb elhelyezkedő eszkimó kezd)

move 3 (ezzel a 2-es rakétaalkatrész mezőjére lép)

igloo (stabil a mező, ide tud építeni)

move 1 (visszalép az előbbi mezőre)

move 3

- **Elvárt kimenet**

remaining energy: 3 (rálépünk a mezőre)

igloo built

remaining energy: 2 (igluépítés energiába kerül)

remaining energy: 1 (visszalépünk az eredeti mezőre)

remaining energy: 0 (belépünk az igluba)

player stepped into igloo

8.3 A tesztelést támogató programok tervei

A tesztelés JUnit-tal valósítjuk meg, ami az erre a célra létrehozott fájlok ból érhető el.

A program tesztesetei által megadott bemeneteire generált kimeneteket a parancssor segítségével vetjük össze a tesztesetekben megadott kimenetekkel.

8.4 Napló

Kezdet	Időtartam	Résznevők	Leírás
2020.04.08	3 óra	Puskás	Tevékenység: Creature, Player, Eskimo, Scientist osztályok terveinek megírása
2020.04.09 16:00	2 Óra	Kuzsner	Tevékenység: A Controller és EndOfGame osztályokhoz tartozó részletek felmérése és megírása.
2020.04.11	1 óra	Jakabos	Tevékenység: hibajavítás a Field osztályban
2020.04.11	1 óra	Jakabos	Tevékenység: Field osztály terveinek megírása.
2020.04.11 18:00	3 óra	Filep	Tevékenység: Menu, Options és ScoreData osztály terveinek megírása.
2020.04.11 23:00	1 óra	Filep	Tevékenység: Player Eats, Player Drops Item, Player Picks Item Up, Player Digs, Player Removes Snow With Shovel, Player Removes Snow With Spade

			tesztesetek megírása.
2020.04.12 10:00	3 óra	Kuzsner	Tevékenység: 5-ös, 6-os és 14es számú tesztesetek megírása.
2020.04.12 11:00	3 óra	Jakabos	Tevékenység: 1, 2, 3, 4 és 30, 31-es számú tesztesetek megírása.
2020.04.12 14:00	2 óra	Kálmán	Tevékenység: Charge, Flare, Gun, Rocket osztályok leírásának elkészítése.
2020.04.13 11:00	2 óra	Kuzsner	Tevékenység: 26-os, 27-es és 28-as tesztesetek megírása.
2020.04.13 13:00	3 óra	Kálmán	Tevékenység: DivingSuit, Food, Rope, Shovel, Spade, Tent osztályok leírásának elkészítése.
2020.04.13 18:00	1 óra	Kálmán	Tevékenység: 7., 8., 9. és 10. tesztesetek elkészítése.
2020.04.13 19:00	2 óra	Filep Puskás Kálmán Jakabos	Értekezés: 8.3-as fejezet megbeszélése, dokumentum átnézése. Döntés: Kálmán megírja a 8.3-as fejezetet, hibák javítása, változások megbeszélése.
2020.04.13 23:00	1 óra	Kálmán	Tevékenység: 10., 24., 25., 29. számú tesztesetek elkészítése.
2020.04.14. 05:00	1,5 óra	Kálmán	Tevékenység: Tárgyfelelősök által előírt tesztesetek definiálása.
2020.04.14. 9:00	0,5 óra	Jakabos	Tevékenység:

			Moveable interfész megírása.
2020.04.14 10:00	0,5 óra	Puskás	Tevékenység: Teszt-esetek befejezése
2020.04.14. 12:00	0,5 óra	Kálmán	Dokumentum átnézése, formázás, hibák javítása

10-Prototípus beadása

47 – cereal_killers

Konzulens:

**Micskei Zoltán
Vörös András**

Csapattagok

Filep Leonárdó Márk	F6JQ0X	filep.leonardo@gmail.com
Kálmán Bendegúz Bence	PTW6BD	kalmanbendeguz@gmail.com
Jakabos Csenge Apolka	KO6MG0	jakabos11@gmail.com
Puskás Levente	SN8YG9	puskas.levente99@gmail.com
Kuzsner Sára	HVE3FV	kuzsnersara@gmail.com

2020.04.26.

10. Prototípus beadása

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Field.java	7,941 byte	2020.03.26.	Jégtáblát reprezentáló Field osztály.
Controller.java	12,049 byte	2020.03.23.	A játékot irányító Controller osztály.
Charge.java	638 byte	2020.03.25.	A töltény osztályt reprezentáló Charge osztály.
DivingSuit.java	674 byte	2020.03.25.	Búvárruhát reprezentáló DivingSuit osztály.
Flare.java	642 byte	2020.03.25.	Jelzőfényt reprezentáló Flare osztály.
Food.java	634 byte	2020.03.25.	Ételt reprezentáló Food osztály.
Gun.java	629 byte	2020.03.25.	Pisztolyt reprezentáló Gun osztály.
Item.java	582 byte	2020.03.25.	Eszközöket reprezentáló Item absztrakt osztály.
Rope.java	1,837 byte	2020.03.25.	Kötelet reprezentáló Rope osztály.
Shovel.java	1,499 byte	2020.03.25.	Ásót reprezentáló Shovel osztály.
Spade.java	1,175 byte	2020.04.25.	Törékeny ásót reprezentáló Spade osztály.
Tent.java	995 byte	2020.04.25.	Sárat reprezentáló Tent osztály.
AudioPlayer.java	1,647 byte	2020.04.25.	Háttérzenéért felelős AudioPlayer osztály.
Main.java	519 byte	2020.04.26.	A játék bemeneteiért felelős Main osztály.
Menu.java	4,456 byte	2020.03.26.	Menu osztály.
MenuItem.java	142 byte	2020.03.26.	MenuItem felsorolt típus, a menüelemek kiválasztásában van szerepe.
Options.java	2,799 byte	2020.04.11.	A menü opcióinak kezeléséért felelős Options osztály.
OptionsItem.java	110 byte	2020.04.11.	OptionsItem felsorolt típus, a menü alelemeinek kiválasztásában van szerepe.
ScoreData.java	996 byte	2020.04.11.	A játék ranglistájáért felelős ScoreData osztály.

main.wav	11,165,618 byte	2020.04.25.	A játékhoz tartozó zenefájl.
menu.wav	18,237,092 byte	2020.04.25.	A menühöz tartozó zenefájl.
Eskimo.java	3,375 byte	2020.03.25.	Eskimo osztály.
Scientist.java	3,544 byte	2020.03.25.	Scientist (tudós) osztály.
Moveable.java	644 byte	2020.04.22.	A szereplők és jegesmedve mozgatását végző Moveable absztrakt osztály.
Player.java	9,154 byte	2020.04.26.	A szereplők általános osztálya.
PolarBear.java	1,609 byte	2020.04.22.	Jegesmedvét reprezentáló PolarBear osztály.
TestFunctions.java	9,910 byte	2020.04.26.	A játék teszteléséhez szükséges függvényeket tartalmazó TestFunctions osztály.
focilabda.xml	16,692 byte	2020.04.26.	Teszteleléshez kért pályát tartalmazó fájl.
highscores.xml	137 byte	2020.04.26.	Toplista vezetéséhez szükséges fájl.
nagypalya.xml	33,171 byte	2020.04.26.	A játékhoz tartozó pályát tartalmazó fájl.
tesztpalya.xml	11,821 byte	2020.04.26.	Belső teszteléshez használt pályát tartalmazó fájl.

10.1.2 Fordítás

A: Eclipse

Hozunk létre egy új projektet az Eclipse fejlesztőkörnyezet segítségével, majd importáljuk a forrásfájlokat a File > Import... > File System menüpontokat választva, a megfelelő könyvtárból, megtartva a hierarchiát. Crtl + B-vel fordítunk.

B.: Parancssor

Ha a géünkre telepítve van a JDK és a környezeti változók be vannak állítva, lépjünk a projekt mappájába. Ezen belül navigálunk az src mappába, és adjuk ki a következő parancsot:

```
javac -encoding ISO-8859-1 -release 11 field\*.java icefield\*.java item\*.java menu\*.java
player\*.java test\*.java
```

Ekkor a projekt lefordítva készen áll a futtatásra.

10.1.3 Futtatás

A.: Eclipse

Állítsuk be a “Run configuration”-ben a menu package Main osztályát a main osztálynak, majd ezután futtatjuk.

B.: Parancssor

Ha a JRE a gépünkre telepítve van, miután a projektet lefordítottuk, ugyanebből a mappából adjuk ki a következő parancsot:
java menu.Main

10.2 Tesztek jegyzőkönyvei

Megjegyzés: A tesztfájlokban a bemenetek és az erre elvárt kimenetek szerepelnek. Ha az adott sorban bemenet van jelölve, akkor egy tabulátor van a sor elején, ha kimenet, akkor nem. Tehát a tabulátorral kezdődő sorokat kell a teszt során a felhasználónak beírnia, amire az ez utáni nem behúzott sorokat kell kimenetül kapja.

10.2.1 Player steps on Field

Teszteleő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 15:45

10.2.2 Player steps on hole

Teszteleő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 17:40

10.2.3 Player steps on unstable Field that flips

Teszteleő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 17:45

10.2.4 Player steps in Igloo

Teszteleő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 17:50

10.2.5 Player steps in Tent

Teszteleő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 18:00

10.2.6 Player steps on Field with polar bear on it

Teszteleő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 18:05

10.2.7 Polar bear steps on Field with Player on it

Teszteleő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 19:30

10.2.8 Polar bear steps on Field with Igloo on it

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 19:40

10.2.9 Polar bear steps on Field with Tent on it

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 19:45

10.2.10 Player saves Player

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 19:50

10.2.11 Snowstorm hits Field

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 19:55

10.2.12 Snowstorm hits Player

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 19:55

10.2.13 Scientist inspects Ice Float

Tesztelő neve	Puskás Levente
Teszt időpontja	2020.04.26 21:05

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 20:00
Teszt eredménye	InspectField(field) Field can hold: 0players energy level: 3
Lehetséges hibaok	-felegleges kiírás: InspectField(field) -0players egybeírva
Változtatások	-space írása az érték és a players közé -kiírás törlése

10.2.14 Player Picks Item Up

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 20:10

10.2.15 Player Eats

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 20:17

10.2.16 Player Removes Snow

Tesztelő neve	Kuzsner Sára
Teszt időpontja	2020.04.27 10:32

10.2.17 Player Removes Snow With Shovel

Tesztelő neve	Kuzsner Sára
Teszt időpontja	2020.04.27 10:44

10.2.18 Player Removes Snow With Spade

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 20:23

10.2.19 Not enough energy for moving

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 20:25

10.2.20 Not enough energy for using item

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 20:26

10.2.21 Player signals end turn

Tesztelő neve	Filep Leonárdó Márk
Teszt időpontja	2020.04.25 21:00

10.2.22 Spade breaks after using it

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 20:30

10.2.23 Tent disappears at the start of a new turn

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 20:35

10.2.24 Players assemble and fire Rocket

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 20:45

10.2.25 Snowstorm doesn't affect Players in Igloo

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 20:50

10.2.26 Snowstorm doesn't affect Players in Tent

Tesztelő neve	Filep Leonardó Márk
Teszt időpontja	2020.04.26 21:10

Tesztelő neve	Filep Leonardó Márk
Teszt időpontja	2020.04.26 21:00
Teszt eredménye	Player is in hole, can't move! energy level: 3 1 player(s) saved energy level: 3
Lehetséges hibaok	Player is in hole, can't move! felesleges kiírás
Változtatások	felesleges kiírás törlése

10.2.27 Food can't overheat Player

Tesztelő neve	Puskás Levente
Teszt időpontja	2020.04.26 22:30

10.2.28 Eskimo tries to build igloo on unstable field

Tesztelő neve	Puskás Levente
Teszt időpontja	2020.04.26 22:50

10.2.29 ELŐÍRT TESZTESET: Mozog a medve és megeszik valakit, a játékosok csak fel-le mászkálnak

Tesztelő neve	Jakabos Csenge Apolka
Teszt időpontja	2020.04.26 20:55

10.2.30 ELŐÍRT TESZTESET: A játékosok törékeny táblára mennek és az eltörök

Tesztelő neve	Kálmán Bendegúz Bence
Teszt időpontja	2020.04.26 20:35

10.2.31 ELŐÍRT TESZTESET: Eszkimó iglut épít

Tesztelő neve	Kálmán Bendegúz Bence
Teszt időpontja	2020.04.26 20:50

10.3 Értékelés

Tag neve	Tag neptun	Munka százalékban
Filep Leonardó Márk	F6JQ0X	20
Kálmán Bendegúz Bence	PTW6BD	20
Jakabos Csenge Apolka	KO6MG0	20
Puskás Levente	SN8YG9	20
Kuzsner Sára	HVE3FV	20

10.4 Napló

Kezdet	Időtartam	Résznevők	Leírás
2020.04.14	1 óra	Puskás Filep Jakabos Kuzsner Kálmán	Értekezés: Feladat és munkamegosztás megbeszélése, döntés: 10 heti feladat részfeladatokra bontása, részfeladatok kiosztása.
2020.04.15. 13:00	40 perc	Puskás Jakabos Kuzsner Kálmán	Konzultáció: kérdések megbeszélése.
2020.04.15. 14:00	30 perc	Puskás Jakabos Kuzsner Kálmán	Döntés: A megoldás letisztázása, változtatások behozása.
2020.04.15. 15:00	1 óra	Kuzsner	Tevékenység: A Közös dokumentumok rendezése, a logok tisztázása, lényeg összegyűjtése.
2020.04.20. 14:00	1 óra 20 perc	Puskás Jakabos Kuzsner	Értekezés: Elvégzett feladatok megbeszélése, hibajavítás.

		Filep	Döntés: A rakétaosztályt egyszerűsíteni kell.
2020.04.23. 12:00	7 óra	Kuzsner	Tevékenység: numOfTurns létrehozása és beállítása a Start() és GameLoop() függvényekben. FociLabdás gráf pálya elkészítése. Nagy pálya megtervezése, elkészítése. Osztályleírások dokumentálása és átnézése, saját osztályaim kommentezése.
2020.04.24. 18:00	3 óra	Kuzsner	Tevékenység: Tesztpálya létrehozása, Pályák megrajzolása papíron, dokumentálásuk a tesztelésnél pont alatt, Gameloop módosítása.
2020.04.25. 17:00	3 óra 30 perc	Puskás Jakabos Kuzsner Filep Kálmán	Közös kódolás. Értekezés: Hibajavítások, pontosítások. Döntés: Kivesszük a rakéta osztályt, a helyettesítő függvényeket Kuzsner és Puskás írja meg.
2020.04.25. 17:00	5 óra	Kuzsner	Tevékenység: A SnowStorm függvény megírása, getterek és setterek elkészítése a kontroller osztályban, a Tryfire() és AreParts Together függvények megírása és dokumentálása, A GameLoop módosítása a vízbeesettek körönkénti ellenőrzése végett. Tesztfüggvény mappa létrehozása, azokhoz bemeneti parancs rendelése, deklarálásuk a programkódban egy új, tesztelő függvényeknek szánt osztályban. SpawnEskimo, SpawnScientist, SpawnPolarBear, SpwnTent, SpawnIgloo, SpawnItem, SpawnSnowStorm függvények megírása. A Gameloop módosítása úgy, hogy vereség esetén -1-et adjon vissza.
2020.04.25. 18:00	2 óra	Jakabos	Tevékenység: Hibajavítás a Field osztályban.
2020.04.25. 20:30	5 óra	Kálmán Puskás	Értekezés: Tesztesetek vizsgálata, hibajavítások, funkcionálisok ellenőrzése.
2020.04.26. 15:45	5 perc	Jakabos	Tevékenység: 10.2.1-es teszteset tesztelése
2020.04.26 17:40	3 óra 30 perc	Jakabos	Tevékenység: 10.2.1-10.2.25-ig levő tesztesetek tesztelése.

			10.2.28-as teszteset tesztelése.
2020.04.26. 16:00	2 óra	Jakabos	Tevékenység: Dokumentum szerkesztése
2020.04.26. 12:00	10 óra 10 perc	Puskás Jakabos Filep Kálmán	Közös kódolás. Értekezés: Hibajavítások, pontosítások. Döntés: Eldöntöttük, hogy a játékunk hibamentes :)
2020.04.27. 10:00	2 óra	Kuzsner	Tevékenység: 16-os, és 17-es számú tesztesetek elvégzése és a hozzájuk tartozó txt file elkészítése. Naplázás és a napló rendezése. A Tesztpálya mezőin lévő hómennyiség és a teherbírási specifikálása.

11-Grafikus felület specifikációja

47 – cereal_killers

Konzulens:

**Micskei Zoltán
Vörös András**

Csapattagok

Filep Leonárdó Márk	F6JQ0X	filep.leonardo@gmail.com
Kálmán Bendegúz Bence	PTW6BD	kalmanbendeguz@gmail.com
Jakabos Csenge Apolka	KO6MG0	jakabos11@gmail.com
Puskás Levente	SN8YG9	puskas.levente99@gmail.com
Kuzsner Sára	HVE3FV	kuzsnersara@gmail.com

11. Grafikus felület specifikációja

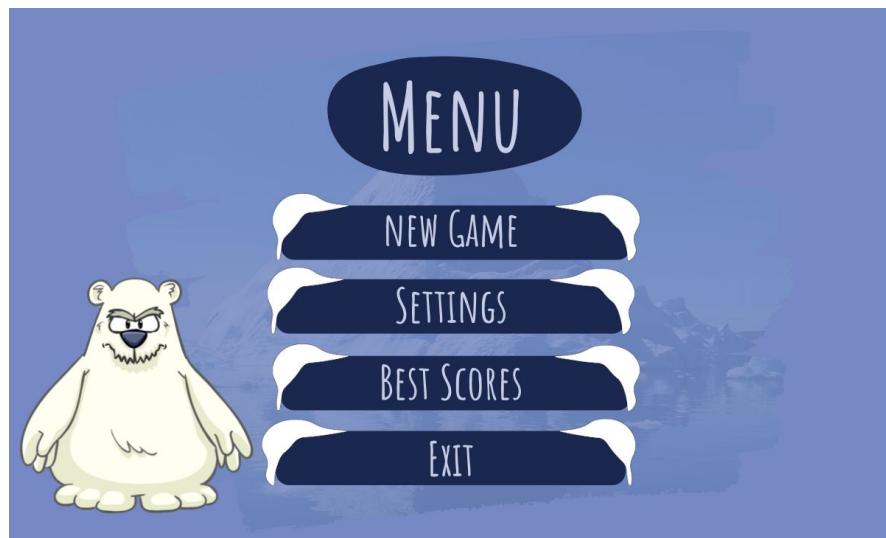
11.1 A grafikus interfész

A játékban a következő képekkel ábrázoljuk az egyes szereplőket, eszközöket:

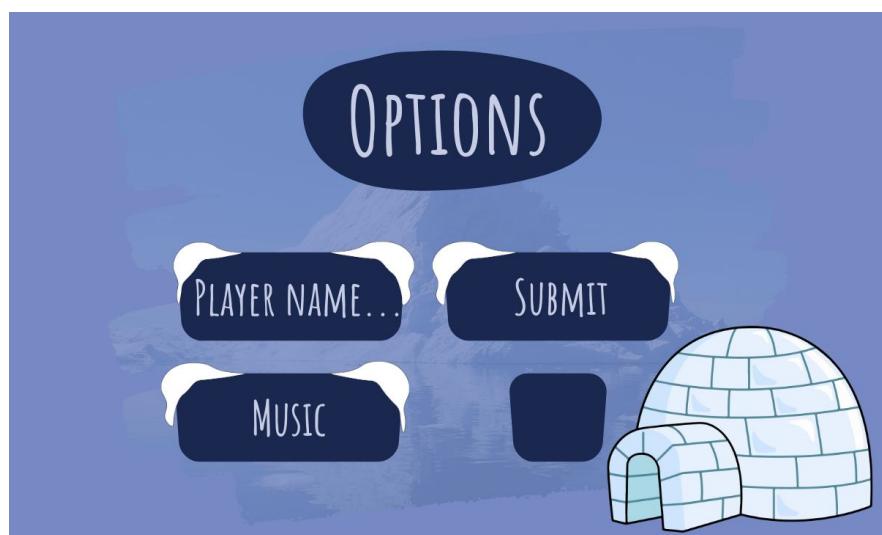


<u>Pisztoly:</u>	<u>Zene bekapcsolva:</u>	<u>Zene kikapcsolva:</u>	<u>Hó:</u>			
						
<u>Lyuk:</u>	<u>Lyukba esett eszkimó:</u>	<u>Lyukba esett játékos aki épp soron van:</u>	<u>Eszkimó, aki épp soron van:</u>			
						
<u>Lyukba esett sarkkutató:</u>	<u>Lyukba esett soron levő sarkkutató:</u>	<u>Sarkkutató, aki épp soron van:</u>	<u>Eszköztár:</u>			
						

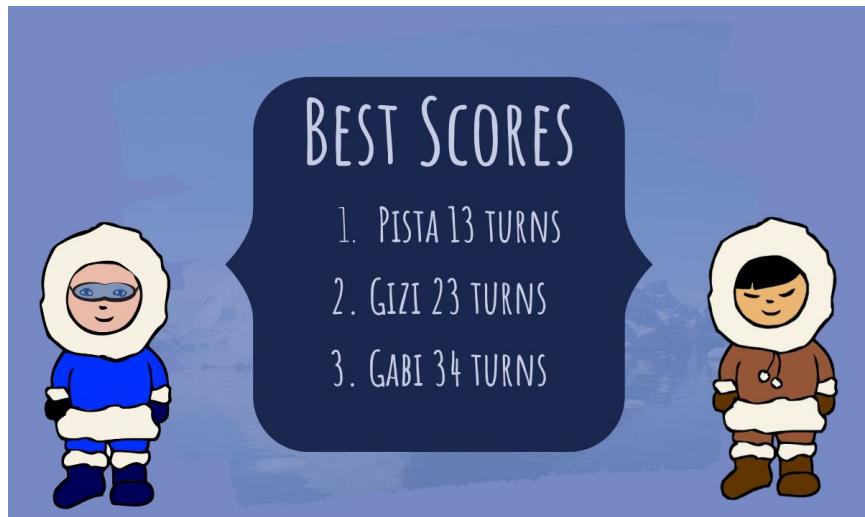
A menü terve:



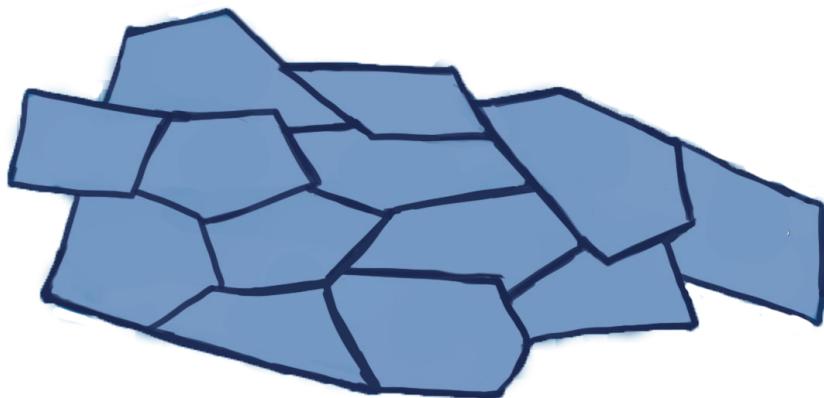
A “Settings” menüpont választása után megjelenő “Options” terve:



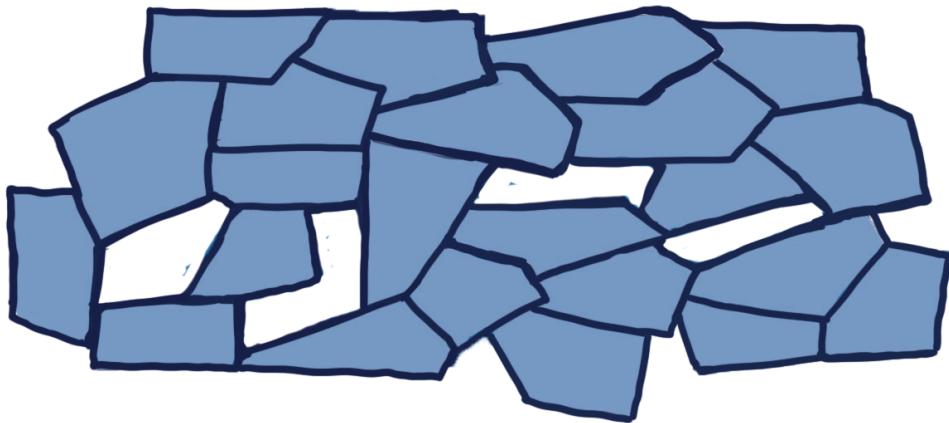
Dicsőséglista terve:



Játék (foci labda pálya):



Játék (nagypalya pálya):



11.2 A grafikus rendszer architektúrája

A grafikus felület tervezése során törekedtünk az MVC megvalósításra.

A felhasználó billentyű lenyomással és egérkattintással képes irányítani a játékot, ezeket egy Listener figyeli és továbbítja a játék modellje felé. A modell megváltozik a felhasználói input hatására, ami ezek után minden egyes grafikus osztályt újraraajzolásra készít (push).

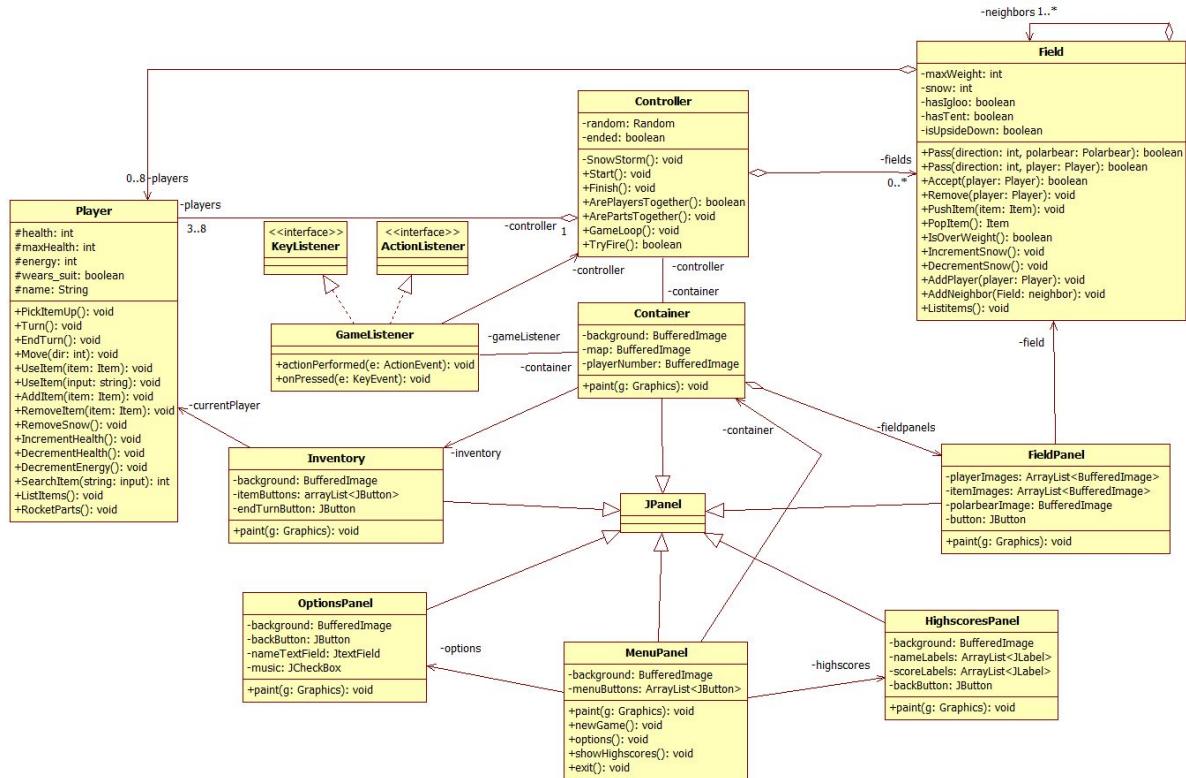
A modellen a Controller osztály is változást generálhat (pl. snowstorm) amely szintén újraraajzolásra készíti a grafikus osztályokat.

11.2.1 A felület működési elve

A grafikus megvalósításhou a Java Swing GUI elemeit használjuk fel. Az működési elvet illetve push alapú, tehát a modell értesíti a felületet, hogy az megváltozott. Az MVC megvalósítás úgy jelenik meg az alkalmazásban, hogy a modell és a nézet jól elkülönülnek egymástól, a modell osztályain a lehető legkevesebb módosítást visszük véghez. A kettő

közötti kapcsolat a Controller osztályban történik, ezáltal a modell képes értesíteni a grafikus felületet.

11.2.2 A felület osztály-struktúrája



11.3A grafikus objektumok felsorolása

Megjegyzés: minden alább felsorolt osztály összes attribútumához létezni fog getter-setter páros, viszont ezeket a könnyebb olvashatóság érdekében itt nem soroljuk fel.

11.3.1 Controller

- Felelősség**

Feladata a játék menetének és folyamatainak irányítása. Idetartozik a pálya elemeinek és a játékosoknak a létrehozása és elhelyezése, a GameLoop, valamint a hóvihar indítása és a jegesmedve mozgatása.

- Ősosztályok**

Nem leszármazott osztály.

- Interfészek**

Nem valósít meg interfész.

- Attribútumok**

- Container container:** A container osztály egy példánya, célja, hogy a modell kapcsolatot létesítsen a nézzettel. Láthatósága:-

- **Metódusok**

- **void GameLoop():** A játék menetéért felelős függvény. A függvény hatással lesz a játék nézetére, frissíti azt (pl snowstorm esetén), illetve szerkezetileg átalakul a grafikus nézet miatt. Láthatósága:+

11.3.2 Container

- **Felelősség**

A grafikus felület gameplay nézetét összefogó panel.

- **Ősosztályok**

JPanel.

- **Interfészek**

Nem valósít meg interfészrt.

- **Attribútumok**

- **BufferedImage background:** A gameplay nézet hátteréül szolgáló kép. Láthatósága:-
- **BufferedImage map:** A gameplay nézet betöltött pályáját szimbolizáló kép. Láthatósága:-
- **BufferedImage playerNumber:** A gameplay nézet jelenlegi játékosának sorszámát mutató kép.
- **GameListener gameListener:** A felhasználó inputjait figyelő listener. Kapcsolatot biztosít a felhasználó és a grafikus felület között. Láthatósága:-
- **Controller controller:** A Controller osztály egy példánya. Kapcsolatot biztosít a modell és grafikus felület között. Láthatósága:-
- **ArrayList<FieldPanel> fieldPanels:** A fieldeket reprezentáló osztály tömbje. Láthatósága:-

- **Metódusok**

- **void paint(Graphics g):** Újraraajzolja ezt, és az ebben található grafikus elemeket. Láthatósága:+

11.3.3 Inventory

- **Felelősség**

A gameplay nézet eszköztáraként szolgáló panel. A jelenlegi játékos életerejét, energiáját és eszköztárában lévő tárgyat mutatja meg.

- **Ősosztályok**

JPanel.

- **Interfészek**

Nem valósít meg interfészrt.

- **Attribútumok**

- **BufferedImage background:** Az inventory hátteréül szolgáló kép. Láthatósága:-
- **JButton endTurnButton:** A játékos egy körének befejezését jelző gomb. Láthatósága:-
- **ArrayList<JButton> itemButtons:** Az eszköztárban lévő tárgyak használatához szükséges gombok. Láthatósága:-
- **Player currentPlayer:** A jelenlegi játékos példánya. Kapcsolatot teremt a grafikus felület és a modell között. Láthatósága:-

- **Metódusok**

- **void paint(Graphics g):** Újrarajzolja ezt, és az ebben található grafikus elemeket. Láthatósága:+

11.3.4 FieldPanel

- **Felelősség**

A grafikus felület gameplay nézetében lévő fieldek grafikus reprezentációjaként szolgáló panel.

- **Ősosztályok**

JPanel.

- **Interfészek**

Nem valósít meg interfacet.

- **Attribútumok**

- **ArrayList<BufferedImage> itemImages:** Az adott fielden lévő tárgyak képei. Láthatósága:-
- **ArrayList<BufferedImage> playerImages:** Az adott fielden lévő playerek képei. Láthatósága:-
- **BufferedImage polarbearImage:** Az adott fielden lévő polarbear képe. Láthatósága:-
- **Field field:** A modell egy field eleme, amelyet grafikusan megvalósít az osztály. Kapcsolat a modell és a grafikus felület között. Láthatósága:-
- **JButton button:** A playernek a fielddel történő interakciójára létrehozott gomb. Láthatósága:-

- **Metódusok**

- **void paint(Graphics g):** Újrarajzolja ezt, és az ebben található grafikus elemeket. Láthatósága:+

11.3.5 MenuPanel

- **Felelősség**

A grafikus felület menüt megvalósító panel.

- **Ősosztályok**

JPanel.

- **Interfész**

Nem valósít meg interfacet.

- **Attribútumok**

- **ArrayList<JButton> menuButtons:** A menü opciók kiválasztásához szükséges gombok. Láthatósága:-
- **BufferedImage background:** A menü hátterének képe. Láthatósága:-
- **Container container:** A Container osztály egy példánya, így tudunk navigálni a gameplay nézetbe. Láthatósága:-
- **OptionsPanel options:** Az options nézetbe történő navilágáshoz szükséges OptionsPanel példánya. Láthatósága:-
- **HighscoresPanel highscores:** A highscores nézetbe történő navigáláshoz szükséges HighscoresPanel egy példánya. Láthatósága:-

- **Metódusok**

- **void paint(Graphics g):** Újrarajzolja ezt, és az ebben található grafikus elemeket. Láthatósága:+

- **void newGame()**: Elindít egy új játéket. Láthatósága:+
- **void options()**: Navigál a beállítások panelra. Láthatósága:+
- **void showHighscores()**: Navigál a ranglista paneljára. Láthatósága:+
- **void exit()**: kilép a játékból. Láthatósága: +

11.3.6 OptionsPanel

- **Felelősség**

A grafikus felület beállítások nézetét megvalósító panel.

- **Ősosztályok**

JPanel.

- **Interfészek**

Nem valósít meg interfacet.

- **Attribútumok**

- **JTextField nameTextField**: A player a ranglistában szereplő nevének megadásához szükséges textfield. Láthatósága:-
- **JCheckBox music**: A zene beállításához szükséges checkbox. Láthatósága:-
- **JButton backButton**: A menübe történő navigáláshoz szükséges gomb. Láthatósága:-
- **BufferedImage background**: A beállítások panel háttérképe. Láthatósága:-

- **Metódusok**

- **void paint(Graphics g)**: Újrarajzolja ezt, és az ebben található grafikus elemeket. Láthatósága:+

11.3.7 HighscoresPanel

- **Felelősség**

A grafikus felület toplistáját megvalósító panel.

- **Ősosztályok**

JPanel.

- **Interfészek**

Nem valósít meg interfacet.

- **Attribútumok**

- **ArrayList<JLabel> nameLabel**: A toplistában szereplő nevek. Láthatósága:-
- **ArrayList<JLabel> scoreLabels**: A toplistában szereplő pontszámok. Láthatósága:-
- **JButton backButton**: A menübe történő navilágáshoz szükséges gomb. Láthatósága:-
- **BufferedImage background**: A toplista hátterének képe. Láthatósága:-

- **Metódusok**

- **void paint(Graphics g)**: Újrarajzolja ezt, és az ebben található grafikus elemeket. Láthatósága:+

11.3.8 GameListener

- **Felelősség**

A felhasználó által kiváltott inputokat kezeli le és továbbítja a modell felé a grafikus felületen keresztül.

- **Ősosztályok**

Nem leszármazott osztály.

- **Interfészek**

ActionListener, KeyListener.

- **Attribútumok**

- **Container container:** A container egy példánya, kapcsolatot teremt a felhasználó és a grafikus felület között. Láthatósága:-
- **Controller controller:** A Controller osztály példánya. Kapcsolatot teremt a felhasználó és a modell között. Láthatósága:+

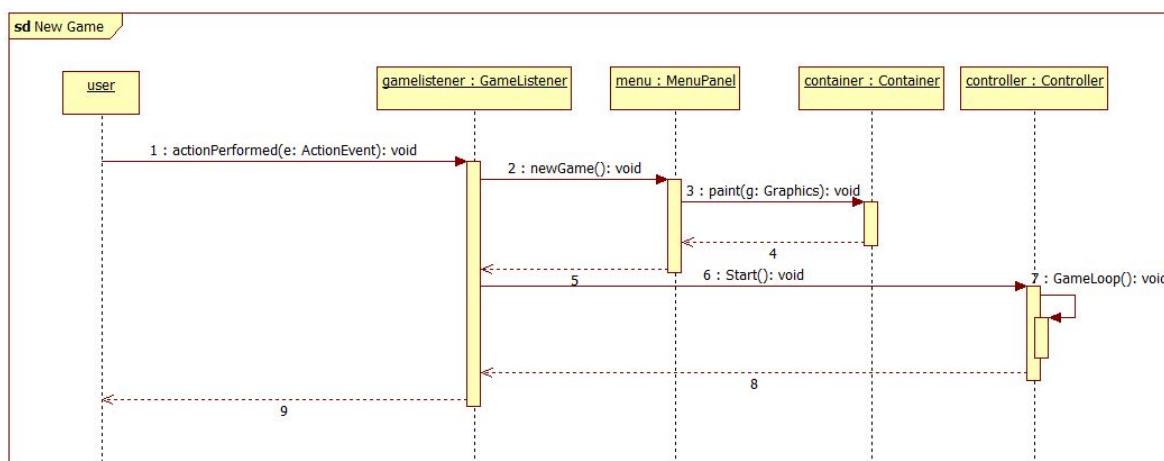
- **Metódusok**

- **void actionPerformed(ActionEvent e):** A grafikus felületeken lévő gombok lenyomásakor keletkező eventketeket kezelő függvény. Láthatósága:+
- **void onPressed(KeyEvent e):** A billentyűzet egy gombjának megnyomásakor keletkező eventketeket kezelő függvény. Láthatósága:+

11.4 Kapcsolat az alkalmazói rendszerrel

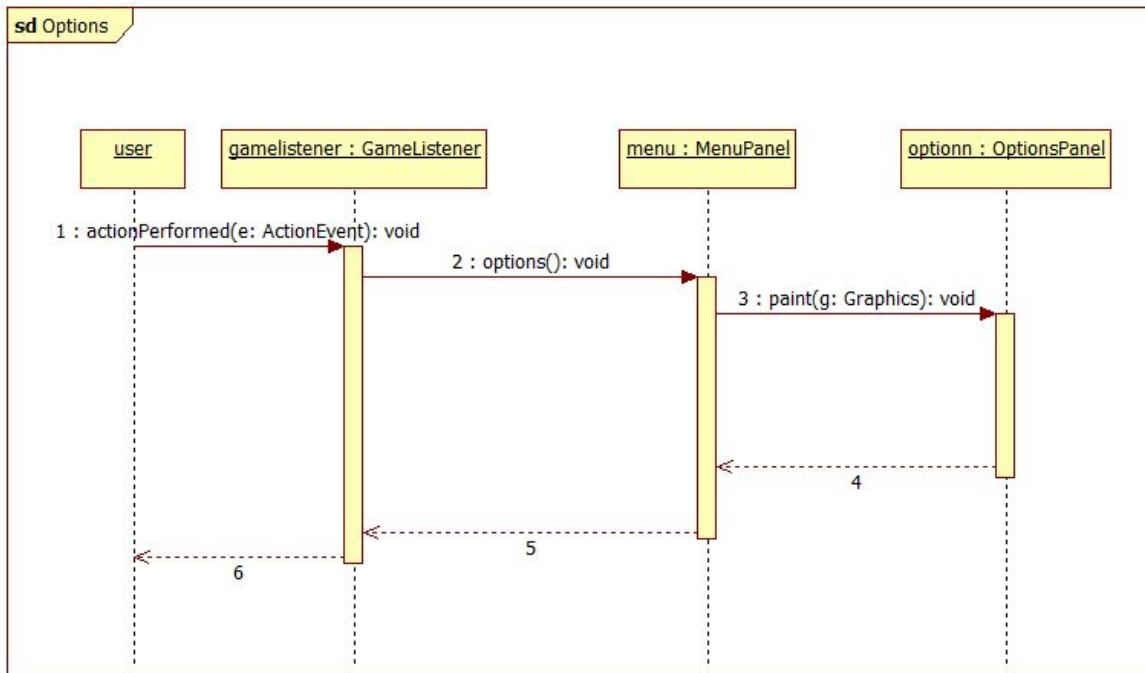
New Game

A felhasználó új játékot indít.



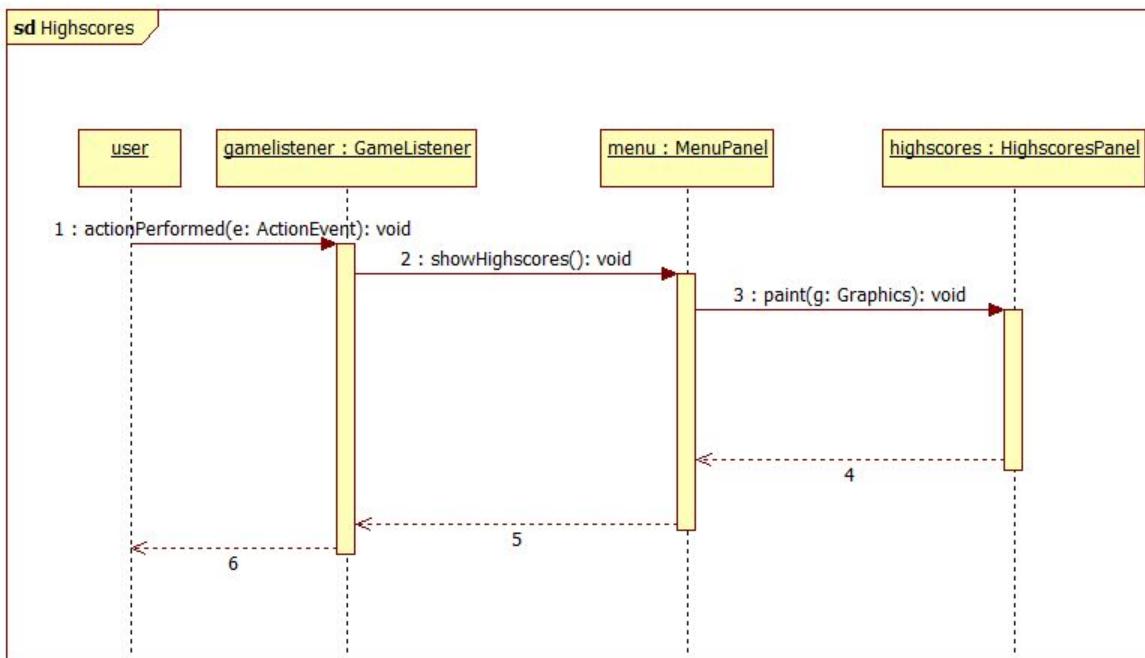
Options

A felhasználó belép a beállításokba.



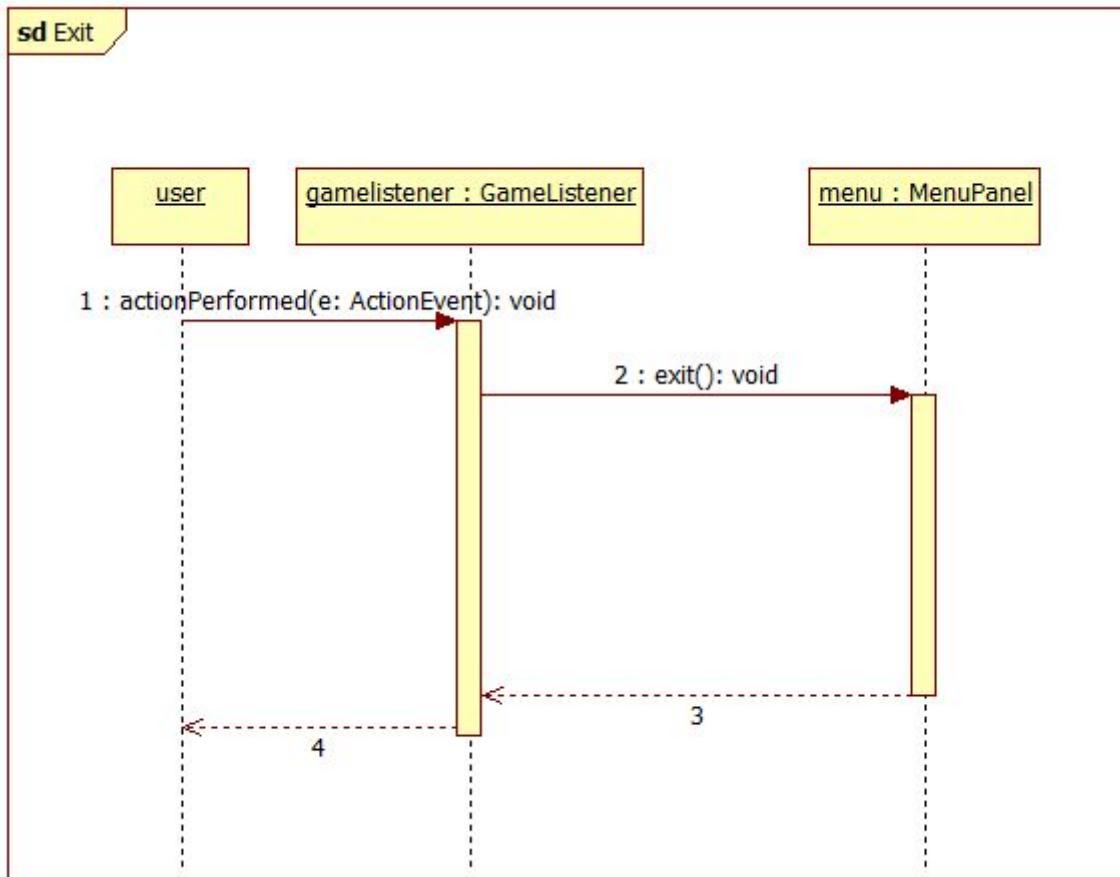
Highscores

A felhasználó megtékinti a toplistát.



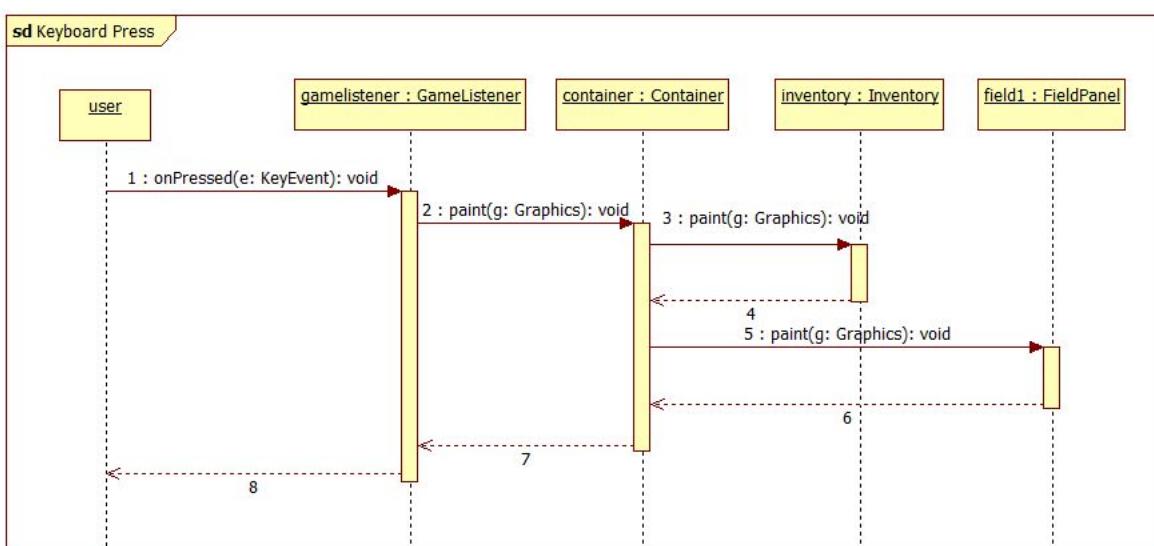
Exit

A felhasználó kilép a játékból.



Keyboard Press

A felhasználó egy billentyűt lenyomva változtatást végez a modellen és a grafikus felület frissül.



11.5 Napló

Kezdet	Időtartam	Részttvevők	Leírás
2020.04.28. 16:00	2 óra	Jakabos Puskás Kuzsner Kálmán	Értekezés: A Grafikus elemekről, szereplők rajzainak kiválasztása, és szükséges képek összegyűjtése. Döntés: Az egységesség miatt az eszkimóhoz hasonló kutatót készítünk.
2020.04.29. 13:00	2 óra	Filep Jakabos Puskás Kuzsner Kálmán	Értekezés: Feladatok megbeszélése, grafikus felület alapelveinek egyeztetése. Döntés: Filep kidolgozza és dokumentálja a grafikus program szerkezetét, valamint Jakabossal együtt létrehozzák a grafikus elemeket.
2020.04.30. 15:00	5 óra	Filep	Tevékenység: Filep elkészíti az osztálydiagramot és megrajzolja a mapokat illetve inventoryt.
2020.04.30. 17:00	1 óra	Jakabos	Tevékenység: képernyőképek elkészítése.
2020.05.01. 19:00	2 óra	Jakabos	Tevékenység: 11.1-es alpont elkészítése.
2020.05.02 10:00	1,5 óra	Puskás	Tevékenység: Controller, Container és Inventory osztályok megírása.
2020.05.02. 12:00	30 perc	Kálmán	Tevékenyég: GameListener

			osztály leírásának elkészítése.
2020.05.03. 18:00	2 óra	Kuzsner	Tevékenység: A Field panel, Menu Panel, Options panel és HighScores Panel osztályok leírása.
2020.05.03. 20:00	30 perc	Kálmán	Tevékenység: Dokumentum formázása, ellenőrzése.

13-Grafikus változat

47 – cereal_killers

Konzulens:
Micskei Zoltán
Vörös András

Csapattagok

Filep Leonárdó Márk	F6JQ0X	filep.leonardo@gmail.com
Kálmán Bendegúz Bence	PTW6BD	kalmanbendeguz@gmail.com
Jakabos Csenge Apolka	KO6MG0	jakabos11@gmail.com
Puskás Levente	SN8YG9	puskas.levente99@gmail.com
Kuzsner Sára	HVE3FV	kuzsnersara@gmail.com

13. Grafikus változat beadása

13.1 Fordítási és futtatási útmutató

13.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Field.java	8,602 byte	2020.03.26.	Jégtáblát reprezentáló Field osztály.
Controller.java	17,833 byte	2020.03.23.	A játékot irányító Controller osztály.
Charge.java	638 byte	2020.03.25.	A töltény osztályt reprezentáló Charge osztály.
DivingSuit.java	704 byte	2020.03.25.	Búvárruhát reprezentáló DivingSuit osztály.
Flare.java	642 byte	2020.03.25.	Jelzőfényt reprezentáló Flare osztály.
Food.java	634 byte	2020.03.25.	Ételt reprezentáló Food osztály.
Gun.java	629 byte	2020.03.25.	Pisztolyt reprezentáló Gun osztály.
Item.java	582 byte	2020.03.25.	Eszköözöket reprezentáló Item absztrakt osztály.
Rope.java	1,882 byte	2020.03.25.	Kötelet reprezentáló Rope osztály.
Shovel.java	1,499 byte	2020.03.25.	Ásót reprezentáló Shovel osztály.
Spade.java	1,277 byte	2020.04.25.	Törékeny ásót reprezentáló Spade osztály.
Tent.java	995 byte	2020.04.25.	Sátrat reprezentáló Tent osztály.
AudioPlayer.java	1,647 byte	2020.04.25.	Háttérzenéért felelős AudioPlayer osztály.
Main.java	899 byte	2020.04.26.	A játék bemeneteiért felelős Main osztály.
Menu.java	5,566 byte	2020.03.26.	Menu osztály.
MenuItem.java	142 byte	2020.03.26.	MenuItem felsorolt típus, a menüelemek kiválasztásában van szerepe.
Options.java	3,508 byte	2020.04.11.	A menü opcióinak kezeléséért felelős Options osztály.
OptionsItem.java	118 byte	2020.04.11.	OptionsItem felsorolt típus, a menü aleleméinek kiválasztásában van szerepe.
ScoreData.java	1,010 byte	2020.04.11.	A játék ranglistájáért felelős ScoreData osztály.
main.wav	11,165,618 byte	2020.04.25.	A játékhöz tartozó zenefájl.

menu.wav	18,237,092 byte	2020.04.25.	A menühöz tartozó zenefájl.
Eskimo.java	3,813 byte	2020.03.25.	Eskimo osztály.
Scientist.java	4,014 byte	2020.03.25.	Scientist (tudós) osztály.
Moveable.java	683 byte	2020.04.22.	A szereplők és jegesmedve mozgatását végző Moveable absztrakt osztály.
Player.java	9,501 byte	2020.04.26.	A szereplők általános osztálya.
PolarBear.java	1,710 byte	2020.04.22.	Jegesmedvét reprezentáló PolarBear osztály.
TestFunctions.java	10,195 byte	2020.04.26.	A játék teszteléséhez szükséges függvényeket tartalmazó TestFunctions osztály.
focilabda.xml	17,112 byte	2020.04.26.	Teszteleshoz kért pályát tartalmazó fájl.
highscores.xml	803 byte	2020.04.26.	Toplista vezetéséhez szükséges fájl.
nagypalya.xml	33,811 byte	2020.04.26.	A játékhoz tartozó pályát tartalmazó fájl.
tesztpalya.xml	12,125 byte	2020.04.26.	Belső teszteléshez használt pályát tartalmazó fájl.
Container.java	4,208 byte	2020.05.14	A játékkablonkat megvalósító osztály, navigál a panelek között. Ez a játék View részéhez tartozik
EndPanel.java	1,072 byte	2020.05.14	A játék végén megjelenő grafikus felület.
FieldPanel.java	8,732 byte	2020.05.13	A mezők grafikus megjelenítéséhez szükséges osztály.
GameListener.java	12,765 byte	2020.05.13	A játék grafikus inputjait általánosan kezelő osztály.
GamePanel.java	7,279 byte	2020.05.14	A pályák grafikus felépítését végző osztály.
HighscoresPanel.java	3,314 byte	2020.05.14	Dicsőséglista grafikus megjelenítését végző osztály.
Inventory.java	5,277 byte	2020.05.14	A játékos eszköztárának grafikus megjelenését biztosító osztály.
MapPanel.java	2,230 byte	2020.05.14	A játék pályáját kiválasztó panel.
MenuPanel.java	3,601 byte	2020.05.10	A menüt megvalósító panel
OptionsPanel.java	5,403 byte	2020.05.10	A beállításokat megvalósító panel.
amaticsc.ttf	146,348 byte	2020.05.16	A játékhoz tartozó betűtípus tartalmazó fájl.

background.png	54,670 byte	2020.05.14	A játék végén használt háttér.
buttonoff.png	22,372 byte	2020.05.14	Menüben használt kikapcsológomb (zene).
buttonon.png	20,871 byte	2020.05.14	Menüben használt bekapcsológomb (zene).
charge.png	16,566 byte	2020.05.14	Felvehető eszköz grafikus megjelenítéséhez használt kép (töltény).
divingsuit.png	20,159 byte	2020.05.14	Felvehető eszköz grafikus megjelenítéséhez használt kép (búvárruha).
eskimo.png	23,152 byte	2020.05.14	Eszkimó szereplő grafikus megjelenítéséhez használt kép.
eskimo_drowned.png	20,402 byte	2020.05.14	Eszkimó szereplő grafikus megjelenítéséhez használt kép. (lyukba esve).
eskimo_selected.png	26,769 byte	2020.05.14	Eszkimó szereplő grafikus megjelenítéséhez használt kép. (Ha épp ő az aktuális szereplő).
eskimo_selected_drowned.png	21,136 byte	2020.05.14	Eszkimó szereplő grafikus megjelenítéséhez használt kép. (Ha épp ő az aktuális szereplő és lyukba esett).
scientist.png	2,891 byte	2020.05.14	Sarkkutató szereplő grafikus megjelenítéséhez használt kép.
scientist_drowned.png	20,630 byte	2020.05.14	Sarkkutató szereplő grafikus megjelenítéséhez használt kép. (lyukba esve).
scientist_selected.png	25,962 byte	2020.05.14	Sarkkutató szereplő grafikus megjelenítéséhez használt kép. (Ha épp ő az aktuális szereplő).
scientist_selected_drowned.png	21,502 byte	2020.05.14	Sarkkutató szereplő grafikus megjelenítéséhez használt kép. (Ha épp ő az aktuális szereplő és lyukba esett).
flare.png	8,310 byte	2020.05.15	Felvehető eszköz grafikus megjelenítéséhez használt kép (jelzőfény).
foci.png	306,966 byte	2020.05.15	Focilabda pálya megjelenítéséhez használt kép.
food.png	19,968 byte	2020.05.14	Felvehető eszköz grafikus megjelenítéséhez használt kép (étel).

gun.png	19,341 byte	2020.05.14	Felvehető eszköz grafikus megjelenítéséhez használt kép (pisztoly).
highscores.png	311,701 byte	2020.05.14	Dicsőséglista háttér.
hole.png	21,650 byte	2020.05.14	Lyukak megjelenítéséhez használt kép.
igloo.png	28,757 byte	2020.05.14	Igluk megjelenítéséhez használt kép.
inventory.png	127,379 byte	2020.05.14	Eszköztárhoz használt kép.
map.png	386,369 byte	2020.05.14	Pályaválasztáshoz használt kép.
lose.png	70,836 byte	2020.05.15	A játék elvesztésekor megjelenő kép.
menu.png	386,846 byte	2020.05.14	Menü grafikus megjelenítéséhez használt kép.
nagy.png	289,292 byte	2020.05.15	Nagypalya pálya megjelenítéséhez használt kép.
options.png	367,672 byte	2020.05.14	Játék opcióinak grafikus megjelenítéséhez használt kép.
polarbear.png	28,003 byte	2020.05.14	Jegesmedve szereplő megjelenítéséhez használt kép.
rope.png	23,008 byte	2020.05.14	Felvehető eszköz grafikus megjelenítéséhez használt kép (kötél).
shovel.png	16,798 byte	2020.05.14	Felvehető eszköz grafikus megjelenítéséhez használt kép (ásó).
snow.png	24,497 byte	2020.05.14	Hó grafikus megjelenítéséhez használt kép.
spade.png	16,777 byte	2020.05.14	Felvehető eszköz grafikus megjelenítéséhez használt kép (törékeny ásó).
tent.png	21,906 byte	2020.05.14	Felvehető eszköz grafikus megjelenítéséhez használt kép (sátor).
teszt.png	306,966 byte	2020.05.15	Tesztpalya pálya megjelenítéséhez használt kép.
win.png	72,325 byte	2020.05.15	A játék megnyerésekor megjelenő kép.

13.1.2 Fordítás és telepítés

A.: Eclipse

Hozzunk létre egy új projektet az Eclipse fejlesztőkörnyezet segítségével, majd importáljuk a forrásfájlokat a File > Import... > File System menüpontokat választva, a megfelelő könyvtárból, megtartva a hierarchiát. Crtl + B-vel fordítunk.

B.: Parancssor

Az elérési útvonalak miatt kicsit problémásabb parancssorból futtatni.

Ha a gépünkre telepítve van a JDK és a környezeti változók be vannak állítva, lépjünk a projekt mappájába. Az src mappán belül létre kell hoznunk még egy src mappát, ebbe bele kell másoljuk az images és menu mappát valamint az .xml fájlokat. Navigálunk a külső src mappába, és adjuk ki a következő parancsot:

```
javac -encoding UTF-8 field*.java icefield*.java item*.java menu*.java player*.java
test*.java graphics*.java
```

Ekkor a projekt lefordítva készen áll a futtatásra.

Fontos: Ha ezek után Eclipseben is akarjuk fordítani, akkor töröljük a legfelső src mappában lévő src mappát.

13.1.3 Futtatás

A.: Eclipse

Állítsuk be a “Run configuration”-ben a menu package Main osztályát a main osztálynak, majd ezután futtatjuk.

B.: Parancssor

Ha a JRE a gépünkre telepítve van, miután a projektet lefordítottuk, ugyanebből a mappából adjuk ki a következő parancsot:

```
java menu.Main
```

13.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Filep Leonardó Márk	F6JQ0X	20
Kálmán Bendegúz Bence	PTW6BD	20
Jakabos Csenge Apolka	KO6MG0	20
Puskás Levente	SN8YG9	20
Kuzsner Sára	HVE3FV	20

13.3 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2020.05.06. 17:00	2 óra	Filep Kuzsner Jakabos Kálmán Puskás	Értekezés: A grafikus felület elkészítésének módjáról. Döntés: Osztályonként osztjuk el a feladatot
2020.05.10. 11:00	4 óra	Kuzsner	Tevékenység: MenuPanel, Highscores panel és options panel osztályok létrehozása, tagváltozók megírása és függvények deklarálása. MenuPanel osztály megírása, a függvényeinek elkészítése.
2020.05.14. 11:00	4 óra	Kuzsner	Tevékenység: MenuPanel, HighScoresPanel és Optionspanel osztályok módosítása a megújult osztálydiagram szerint, a newGame, options ,showHighscores és exit függvények törlése, új, megvalósítás megírása. MenuPanel, HighscoresPanel es OptionsPanel osztályok kódjának megírása, és a szükséges getterek, setterek elkészítése.
2020.05.15. 15:00	4 óra	Kálmán	Tevékenység: GameListener osztály megtervezése, implementálása. Felhasználói interakciók összekötése a modellel, és a modellen történő változások kiváltása.
2020.05.16 14:00	3 óra	Jakabos	Tevékenység: Munkaórák számolása, FieldPanel implementálása.
2020.05.16. 14:00	5 óra	Kálmán	Tevékenység: Szálkezelés megvalósítása a szinkronizált működéshez. Controller osztályval való kommunikáció és a vele való szálbiztoság megvalósítása.
2020.05.12 14:00	6 óra	Filep	Tevékenység: Container, GamePanel, MapPanel implementálása
2020.05.12 16:00	2 óra	Puskás	Tevékenység: Inventory osztály implemantálása
2020.05.14 12:00	6 óra	Filep	Tevékenység: Container módosítása, MenuPanel módosítása.
2020.05.14 14:00	3 óra	Puskás	Tevékenység: Inventory osztály módosítása, bővítése, finomítása
2020.05.15 11:00	6 óra	Filep	Tevékenység: Feladatok kiosztása, TODO-k írása, egyeztetés a többiekkel, felmerülő grafikus problémák kijavítása.

2020.05.16 11:00	3 óra	Puskás	Tevékenység: HighscoresPanel osztály módosítása, javítása
2020.05.17. 12:00	5 óra	Kálmán	Tevékenység: Tesztpálya elérésének és tesztparancsok bevitelének megvalósítása. Formázás, kommentezés, felesleges kiírások kivevése, hibajavítások.
2020.05.17. 14:00	3 óra	Puskás	Tevékenység: Inventory, HighscoresPanel osztály kommentezése. Player, Scientist, Eskimo osztályok javítása.
2020.05.17 18:00	6 óra	Filep	Tevékenység: Végeleges osztálydiagram elkészítése, kommentelés véghezvitele.
2020.05.17 15:00	6 óra	Jakabos	Tevékenység: Dokumentumok szerkesztése, kódsorok összeszámolása.
2020.05.17 21:00	3 óra	Puskás	Tevékenység: Tesztelés a tesztesetek szerint, apróbb finomítások.
2020.05.18 10:00	30 perc	Jakabos	Tevékenység: Dokumentumokból kimaradó dolgok hozzáadása, dokumentum átolvasása.
2020.05.18 12:00	2 óra	Puskás	Tevékenység: Dokumentumok formázása, összefűzése, apróbb korrigálások, javítások.

13-Összefoglalás

47 – cereal_killers

Konzulens:
Micskei Zoltán
Vörös András

Csapattagok

Filep Leonárdó Márk	F6JQ0X	filep.leonardo@gmail.com
Kálmán Bendegúz Bence	PTW6BD	kalmanbendeguz@gmail.com
Jakabos Csenge Apolka	KO6MG0	jakabos11@gmail.com
Puskás Levente	SN8YG9	puskas.levente99@gmail.com
Kuzsner Sára	HVE3FV	kuzsnersara@gmail.com

14. Összefoglalás

14.1 A projektre fordított összes munkaidő

Tag neve	Munkaidő (óra)
Filep Leonárdó Márk	91
Kálmán Bendegúz Bence	92
Jakabos Csenge Apolka	91
Puskás Levente	85
Kuzsner Sára	95
Összesen	455

- A feltöltött programok forrássorainak száma**

Fázis	KódSOROK száma
Szkeleton	781
Prototípus	1459
Grafikus változat	2837
Összesen	5077

14.2 • Projekt összegzés

14.2.1 Mit tanultak a projektből konkrétan és általában?

A projekt során megtanultuk, hogyan osszuk be az időket, tervezzünk meg és valósítsunk meg egy megoldást illetve hogy hogyan kommunikáljuk az ötleteinket munkatársaink felé. A közös munka nem csak a nehézségekre világított rá, hanem a munkamegosztással járó pozitívumokra is: minden volt valaki aki szívesen elvégzett egy feladatot, amit más esetleg nem kedvelt annyira vagy segített ha elakadtunk. Láthattuk hogy milyen egy nem általunk kitalált házi feladaton dolgozni, hanem egy “megrendelésen” dolgozni, határidőre, együttműködve másokkal.

14.2.2 Mi volt a legnehezebb és a legkönnyebb?

A csapatban összesen öten dolgoztunk, így talán a legnehezebb a kommunikáció volt, hogy összehangoljuk az időket és minden meg tudjunk beszélni határidő előtt.

A legkevesebb gondunk a feladatok kiosztásával volt, mindig volt aki szivesen és gyorsan elvállalta az adott dolgot.

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

Igen.

14.2.4 Ha nem, akkor hol okozott ez nehézséget?

-

14.2.5 Milyen változtatási javaslatuk van?

-

14.2.6 Milyen feladatot ajánlanának a projektre?

A "Bomb it" nevű játékhoz hasonló de egyszerűsített játékot.

14.2.7 Egyéb kritika és javaslat

-