

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Sarkkutató

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

3.1.2 Eszkimó

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

3.1.3 Jégtábla

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

3.1.4 Lapát

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

3.1.5 Kötél

Egyike a jégből kiásható tárgyaknak. A játékosnak kötéltre van szüksége ahhoz, hogy kimenekítsen egy másik, vízbe esett játékost.

3.1.6 Búvárruha

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

3.1.7 Élelem

Egyike a jégből kiásható tárgyaknak. A felhasználásával a játékosok növelhetik a testhőjüket eggyel.

3.1.8 Iglu

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

3.1.9 Hóvihar

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

3.1.10 Jelzőrakéta

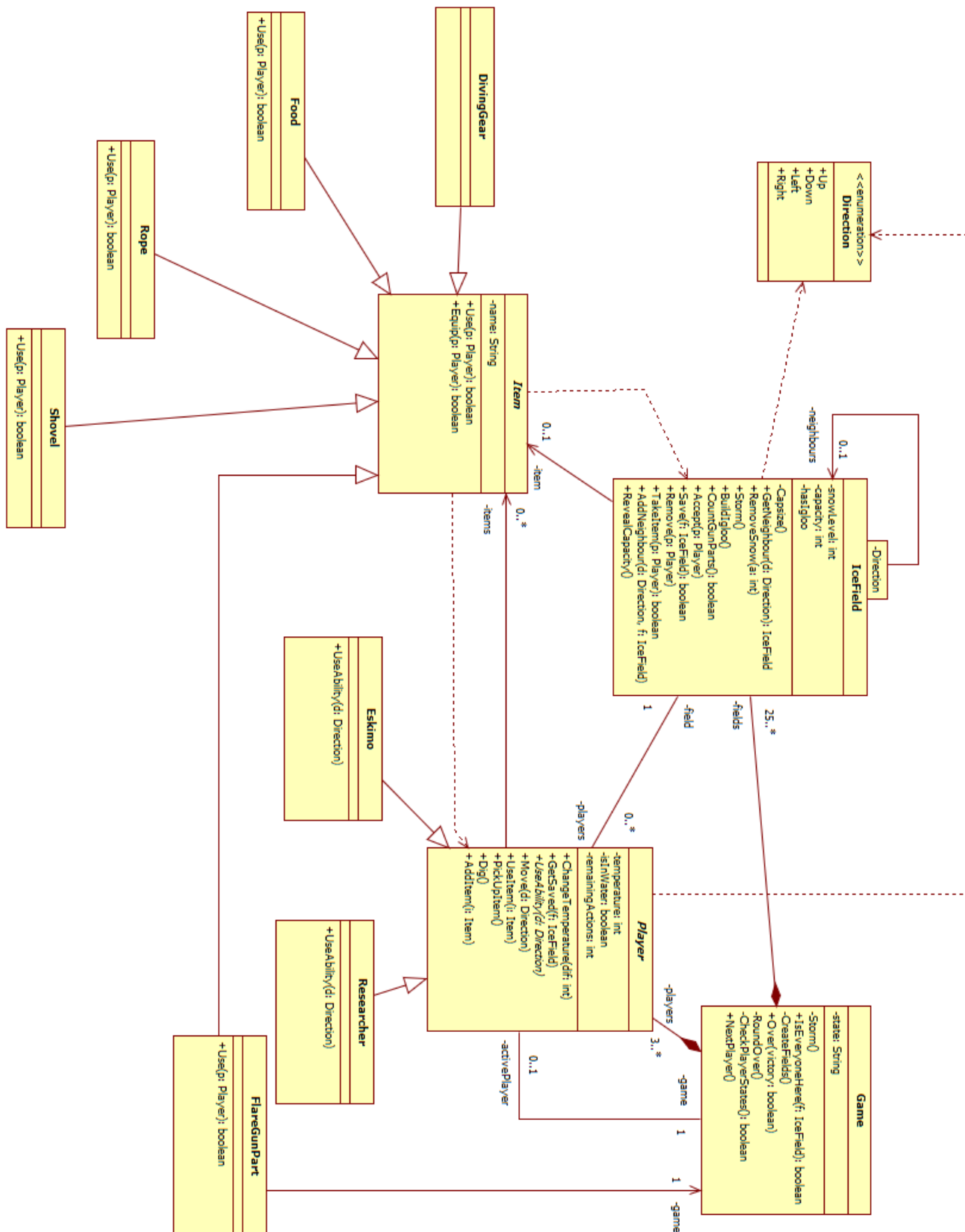
A játékot ennek az elsütésével lehet megnyerni, amikor minden játékos egy jégtáblán áll.

3.1.11 Jelzőfény, pisztoly, patron

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

3.2 Statikus struktúra diagramok

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



3.3 Osztályok leírása

3.3.1 DivingGear

- **Felelősség**

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

- **Ősosztályok**

Item → DivingGear

- **Metódusok**

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

3.3.2 Eskimo

- **Felelősség**

Képes Iglut építeni.

- **Ősosztályok**

Player → Eskimo

- **Metódusok**

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

3.3.3 FlareGunPart

- **Felelősség**

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

- **Ősosztályok**

Item → FlareGunPart

- **Metódusok**

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

3.3.4 Food

- **Felelősség**

Fel lehet használni a játékost testhőjének növelésére.

- **Ősosztályok**

Item → Food

- **Metódusok**

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

3.3.5 Game

- **Felelősség**

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

- **Attribútumok**

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

- **Metódusok**

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

3.3.6 IceField

- **Felelősség**

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

- **Attribútumok**

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

- **Metódusok**

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

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

3.3.7 Item

- **Felelősség**

A használható tárgyak heterogén kollekciója.

- **Attribútumok**

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

- **Metódusok**

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

3.3.8 Player

- **Felelősség**

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

- **Attribútumok**

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

- **Metódusok**

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

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

3.3.9 Researcher

- **Felelősség**

Meg tudja nézni egy IceField kapacitását.

- **Ősosztályok**

Player → Researcher

- **Metódusok**

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

3.3.10 Rope

- **Felelősség**

Ki lehet menteni a játékosokat a vízből a környező mezőkről.

- **Ősosztályok**

Item → Rope

- **Metódusok**

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

3.3.11 Shovel

- **Felelősség**

Havat lehet vele eltakarítani a jégmezőről.

- **Ősosztályok**

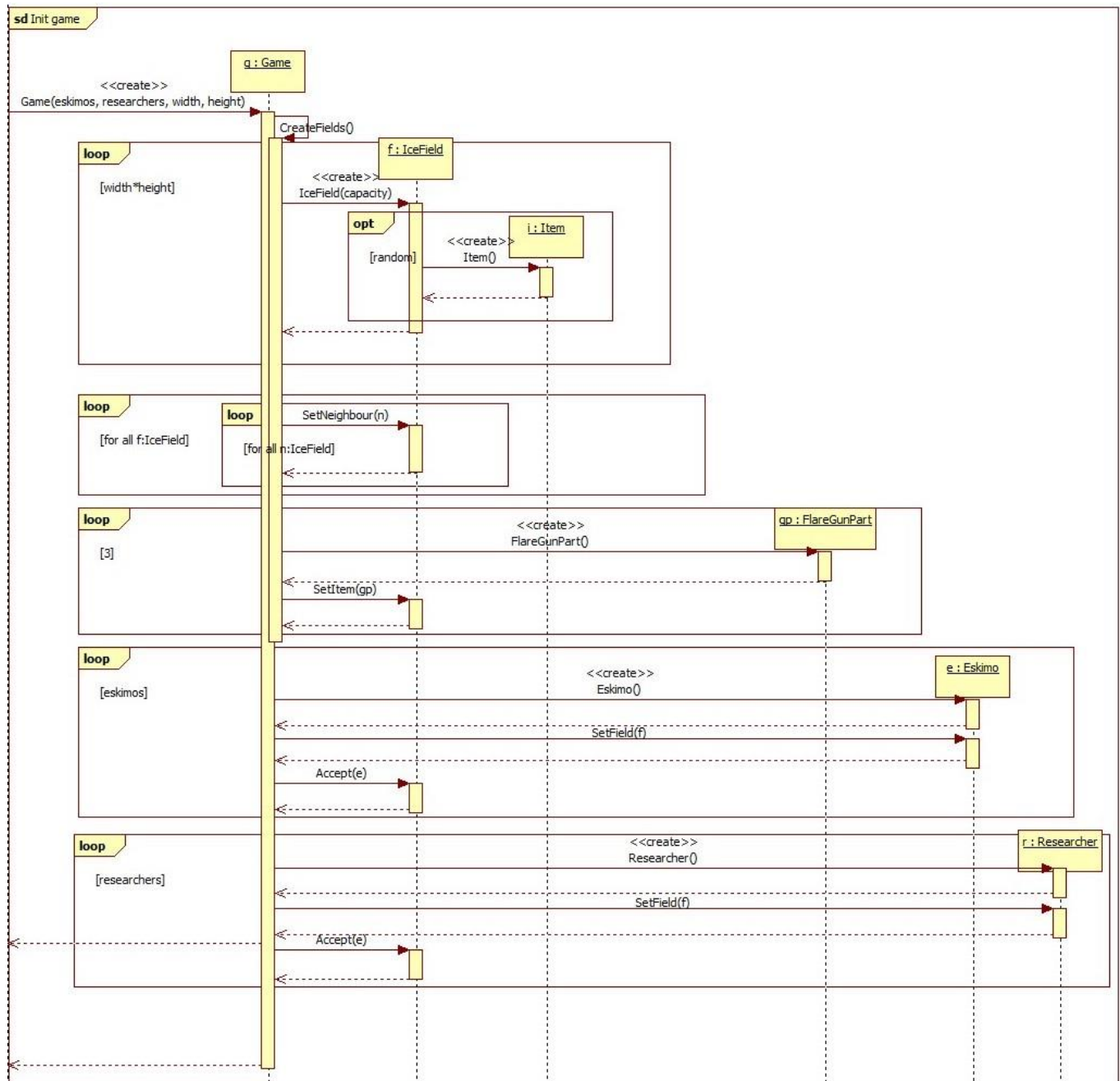
Item → Food

- **Metódusok**

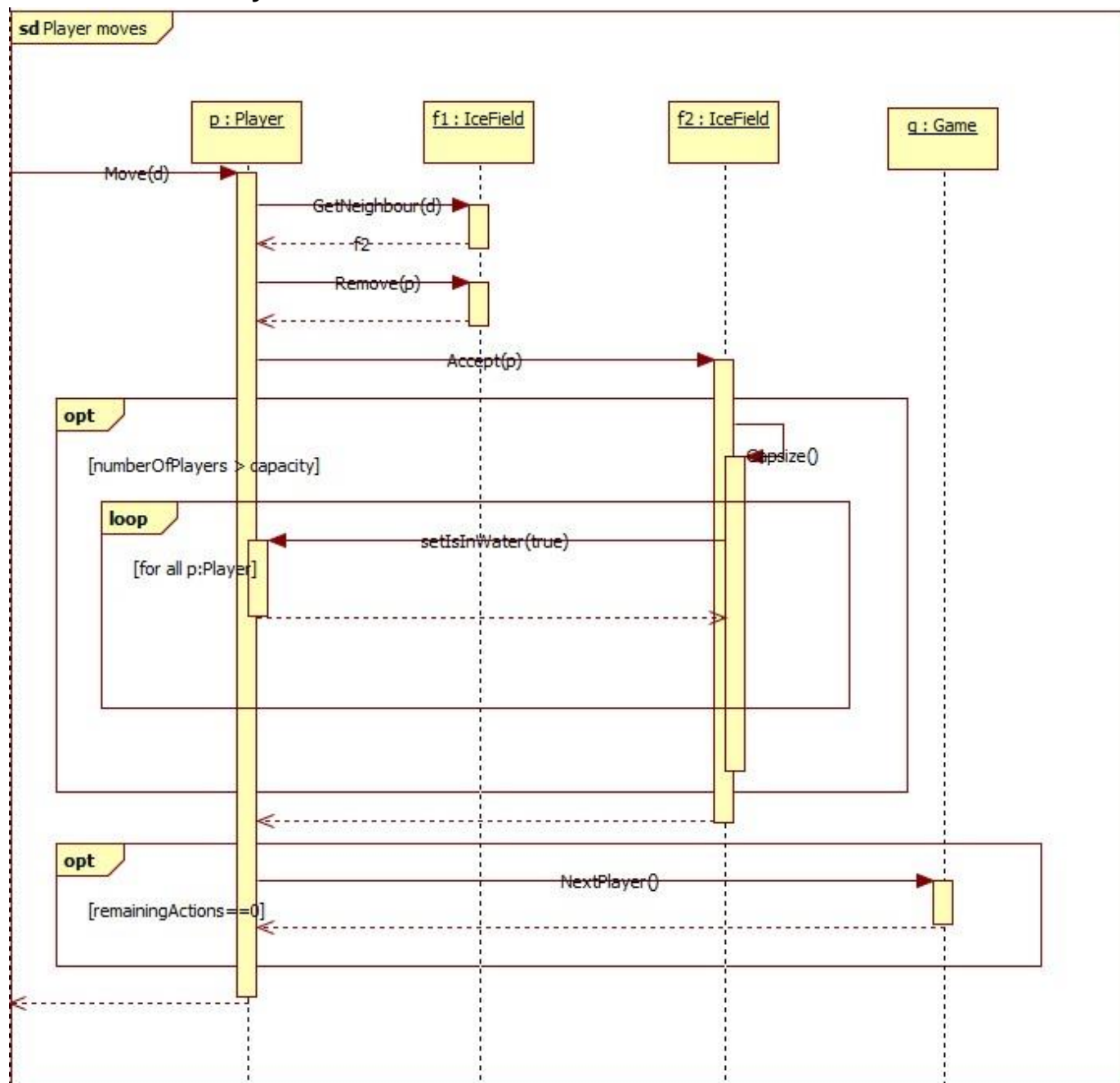
- **boolean Use(Player p):** Két egységnyi hóréteget takarít el a jégmezőről, ahol a használója áll.

3.4 Szekvencia diagramok

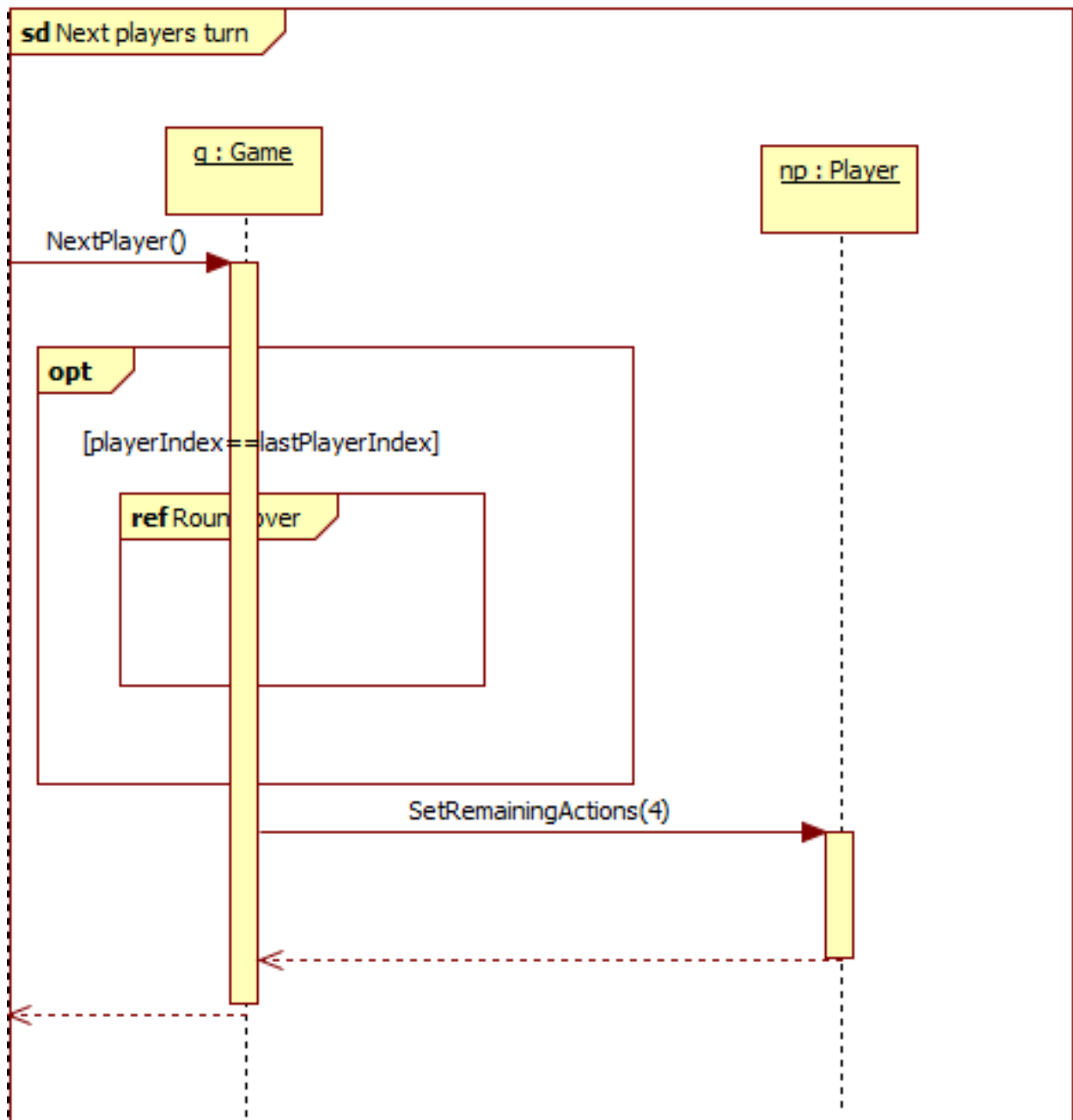
3.4.1 Init Game



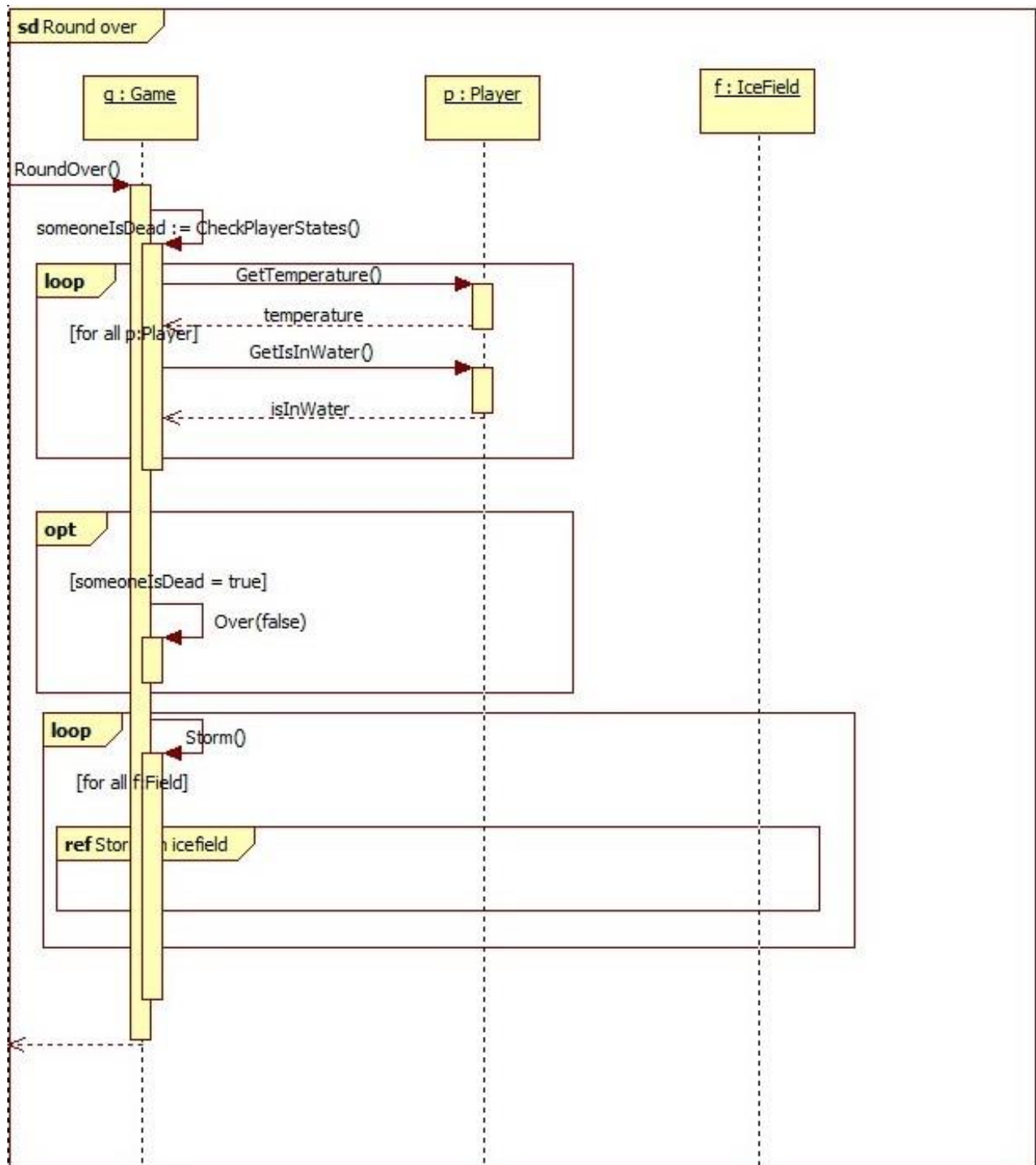
3.4.2 Player Moves



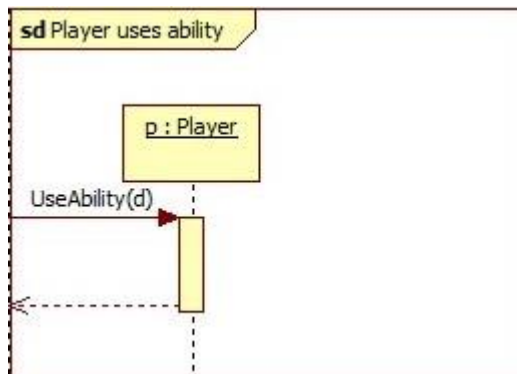
3.4.3 Next Player's turn



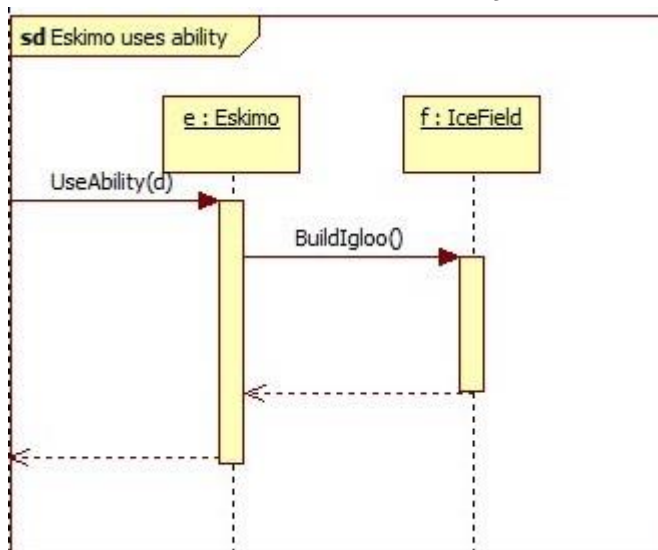
3.4.4 Round Over



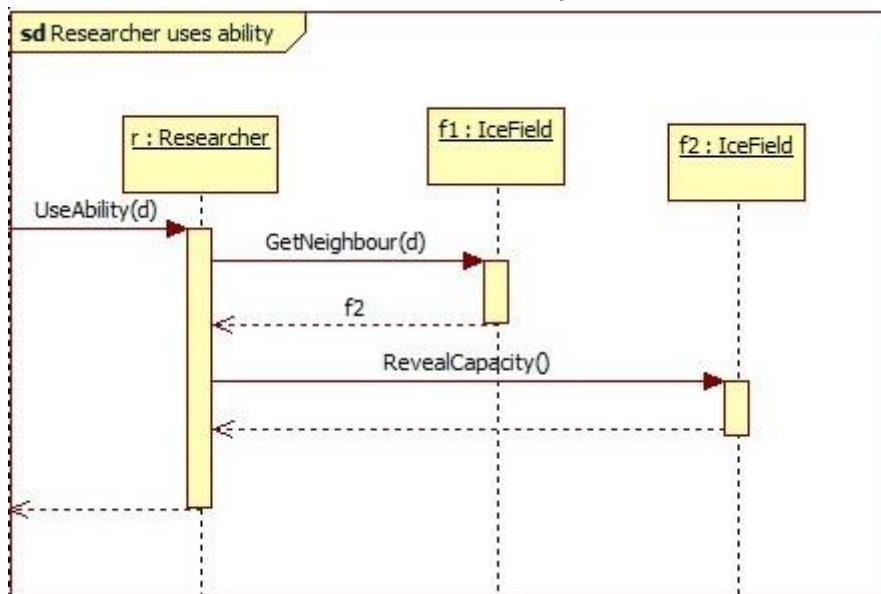
3.4.5 Player Uses Ability



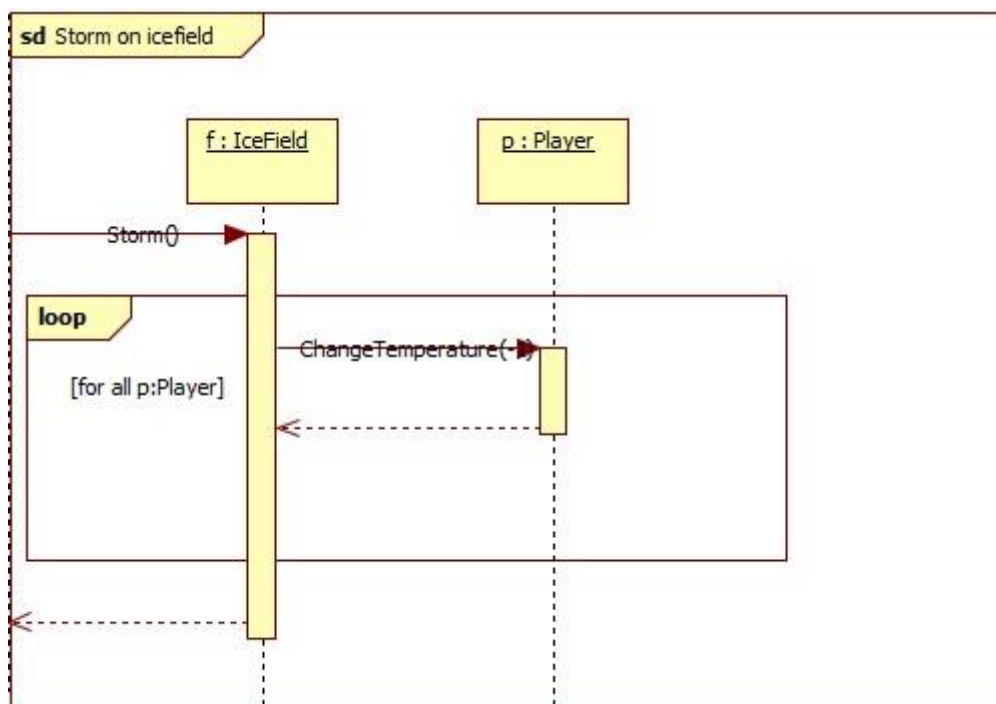
3.4.6 Eskimo Uses Ability



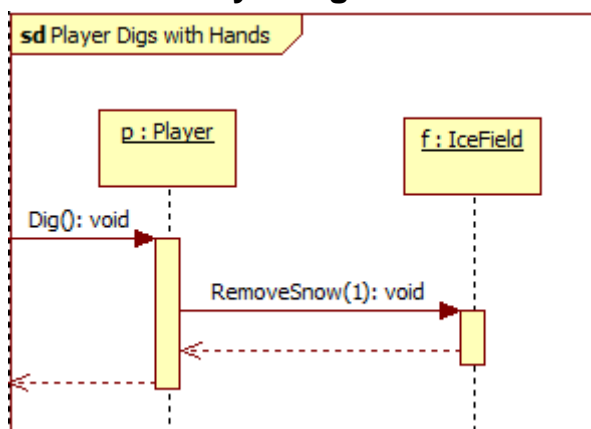
3.4.7 Researcher Uses Ability



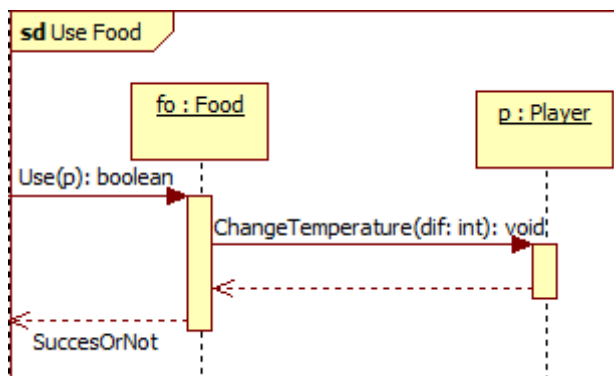
3.4.8 Storm On Icefield



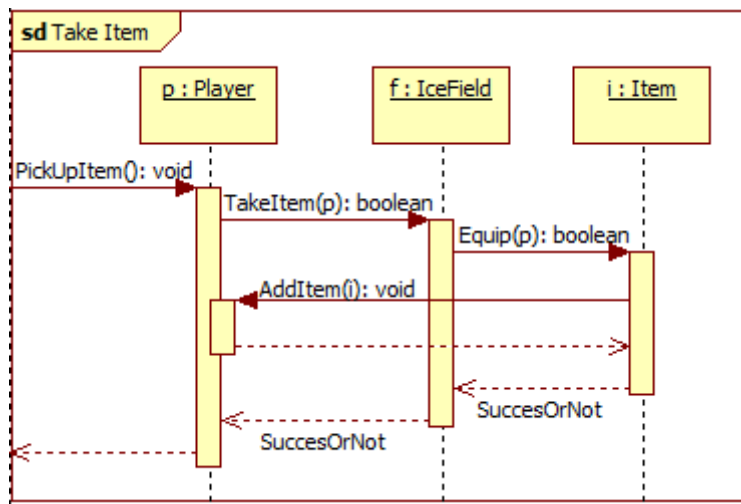
3.4.9 Player Digs with Hands



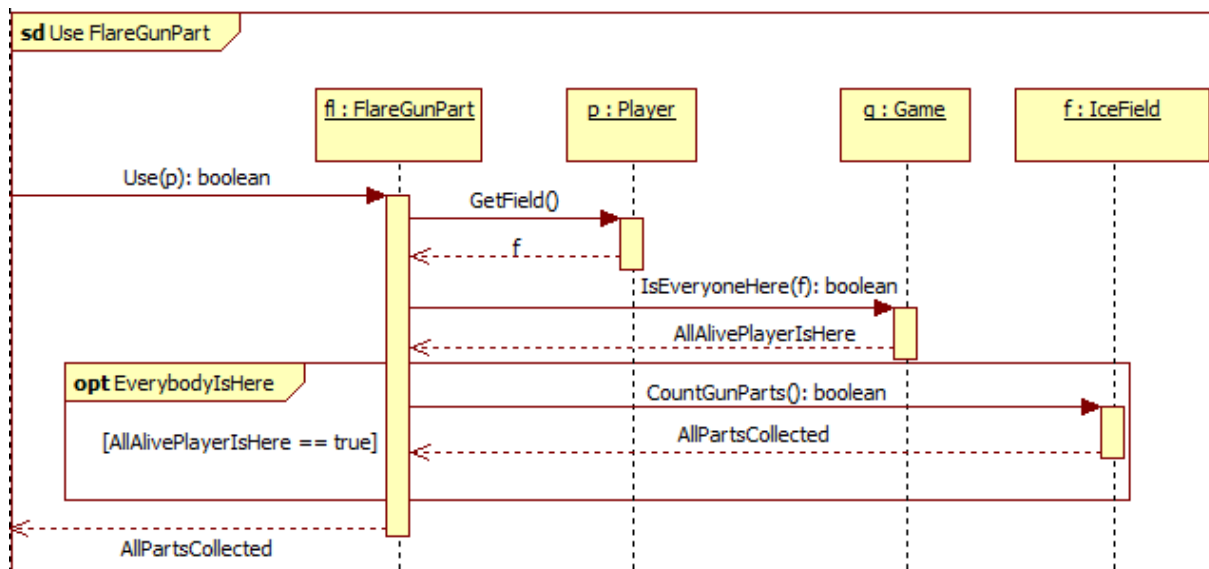
3.4.10 Use Food



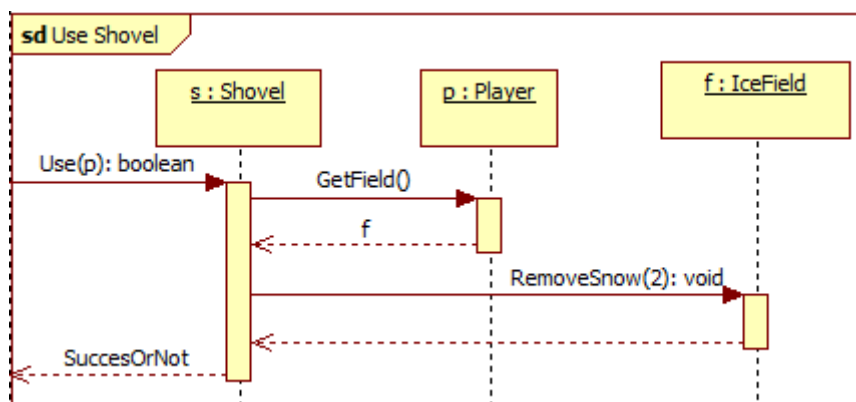
3.4.11 Take Item



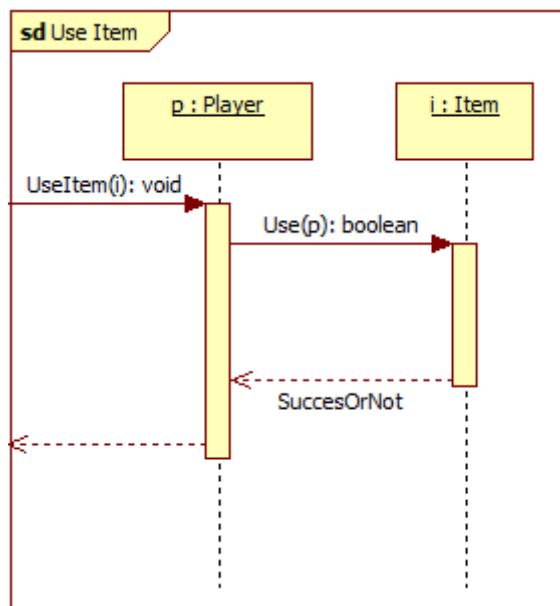
3.4.12 Use Flare Gun Part



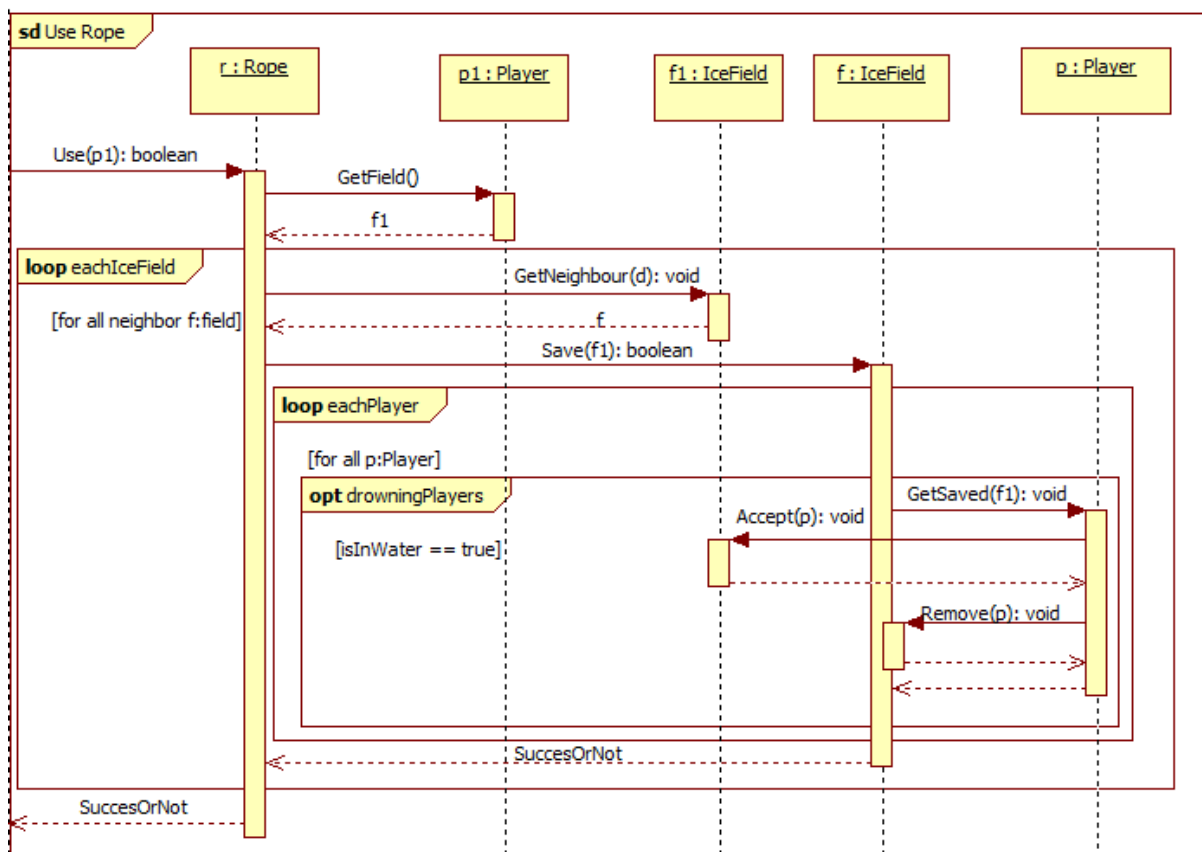
3.4.13 Use Shovel



3.4.14 Use Item



3.4.15 Use Rope



3.5 State-chartok

Állapotgépre nincs szükség a programterv bemutatásához.

3.6 Napló

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