

## 4. Analízis modell kidolgozása 2

66 – [\[simon\\_balazst\\_szeretnenk\\_konzulensnek\]](#)

Konzulens:

Simon Balázs

### Csapattagok:

Kiss Andor	TXC54G	kissandor4@gmail.com
Konrád Márk	JSPDME	konrad0816@gmail.com
Glávits Balázs Róbert	NMZC9G	glavits.balazs@gmail.com
Máté Botond	ELOYOV	m.botond7@gmail.com
Lant Gábor	P35E36	lant.gabor98@gmail.com

2020. március 8.

# Tartalomjegyzék

<b>4</b>	<b>Analízis modell kidolgozása 1</b>	<b>5</b>
4.1	Objektum katalógus . . . . .	5
4.1.1	Játékos . . . . .	5
4.1.2	Jégtábla . . . . .	5
4.1.3	Kötél . . . . .	5
4.1.4	Búvárruha . . . . .	5
4.1.5	Lapát . . . . .	5
4.1.6	Élelem . . . . .	5
4.1.7	Rakétapisztoly Alkatrész . . . . .	5
4.1.8	Iglu . . . . .	5
4.2	Osztályok leírása . . . . .	5
4.2.1	BareHands . . . . .	5
4.2.2	BareIce . . . . .	6
4.2.3	CantRescue . . . . .	6
4.2.4	ChillStormStrategy . . . . .	6
4.2.5	ChillWaterStrategy . . . . .	6
4.2.6	DigStrategy . . . . .	6
4.2.7	DryLand . . . . .	7
4.2.8	Empty . . . . .	7
4.2.9	Eskimo . . . . .	7
4.2.10	Food . . . . .	7
4.2.11	FoodStore . . . . .	7
4.2.12	Game . . . . .	8
4.2.13	Igloo . . . . .	8
4.2.14	Item . . . . .	8
4.2.15	Naked . . . . .	9
4.2.16	Part . . . . .	9
4.2.17	PartStore . . . . .	9
4.2.18	Player . . . . .	9
4.2.19	PolarExplorer . . . . .	10
4.2.20	RescueStrategy . . . . .	10
4.2.21	Rope . . . . .	10
4.2.22	RopeRescue . . . . .	11
4.2.23	ScubaGear . . . . .	11
4.2.24	ScubaWearing . . . . .	11
4.2.25	Sea . . . . .	11
4.2.26	Shovel . . . . .	12
4.2.27	ShovelDig . . . . .	12
4.2.28	Tile . . . . .	12
4.2.29	WaterResistanceStrategy . . . . .	13
4.3	Statikus struktúra diagramok . . . . .	14
4.4	Szekvencia diagramok . . . . .	16
4.5	Napló . . . . .	46

## Ábrák jegyzéke

4.1	Osztálydiagram 1. . . . .	14
4.2	Osztálydiagram 2. . . . .	15
4.3	Game.CreateEskimo(), Game.CreatePolarExplorer() . . . . .	16
4.4	Game.Turn() . . . . .	17
4.5	Game.CreateIce() . . . . .	18
4.6	Game.CreateUnstableIce() . . . . .	19
4.7	Game.CreateHole() . . . . .	20
4.8	Game.CreateSea() . . . . .	21
4.9	Game.CreatePolarExplorer() . . . . .	22
4.10	Game.CreateEskimo() . . . . .	22
4.11	Game.GenerateItem() . . . . .	23
4.12	Player.Step(direction: int) . . . . .	23
4.13	Player.Dig() . . . . .	24
4.14	Player.PickUp() . . . . .	25
4.15	Player.Equip(int) . . . . .	26
4.16	Player.PlaceOn(Tile) . . . . .	26
4.17	Player.EatFood() . . . . .	27
4.18	Player.RescueTeammate(direction: int) . . . . .	27
4.19	Player.Chill() . . . . .	28
4.20	Player.ResistWater() . . . . .	28
4.21	Player.AssembleFlare() . . . . .	29
4.22	Player.AssembleFlare() . . . . .	30
4.23	Eskimo.BuildIgloo() . . . . .	30
4.24	PolarExplorer.Examine(direction: int) . . . . .	31
4.25	Tile.StepOn(Player) . . . . .	32
4.26	Tile.StepOff(Player) . . . . .	33
4.27	Tile.TakeItem() . . . . .	34
4.28	Tile.ChillWater() . . . . .	35
4.29	Tile.ChillStorm() . . . . .	35
4.30	Naked.Chill(Player) . . . . .	36
4.31	ScubaWearing.Chill(Player) . . . . .	36
4.32	BareIce.Chill() . . . . .	37
4.33	DryLand.Chill(Tile) . . . . .	37
4.34	Igloo.Chill(Tile) . . . . .	38
4.35	Sea.Chill(Tile) . . . . .	38
4.36	Empty.GiveTo(Player) . . . . .	39
4.37	Food.GiveTo(Player) . . . . .	40
4.38	FoodStore.Feed(Player) . . . . .	41
4.39	ScubaGear.GiveTo(Player) . . . . .	41
4.40	Rope.GiveTo(Player) . . . . .	42
4.41	Part.GiveTo(Player) . . . . .	42
4.42	Shovel.GiveTo(Player) . . . . .	43
4.43	CantRescue.Rescue(Tile, Tile) . . . . .	43
4.44	RopeRescue.Rescue(Tile, Tile) . . . . .	43
4.45	PartStore.Gain(PartStore) . . . . .	44
4.46	BareHandsDig.Dig(Tile) . . . . .	44

4.47 ShovelDig.Dig(Tile)	45
--------------------------	----

## 4. Analízis modell kidolgozása 1

### 4.1. Objektum katalógus

#### 4.1.1. Játékos

Három vagy több van belőle. Körökre bontva teszik a dolgukat. Saját körükben tudnak mozogni, különböző tárgyakat használni vagy a speciális képességüket használni. A játék megnyeréséhez szükséges rakétapisztoly alkatrészek összegyűjtése a feladatuk. Ha vízbe esnek, vagy kihűlnek akkor a játéknak vége.

#### 4.1.2. Jégtábla

Ilyenek alkotják a játékos számára a játéktér, ezeken lehet mozogni. Jégtáblák tartalmazhatnak tárgyakat amelyeket ki lehet ásni. Az instabil jégtábla képes vízbe ejteni a rajta állókat, ha túl sokan vannak. A jégtáblán lehet hó. Néha lehet rajta hóvihár, mely csökkenti a rajta állók testhőjét

#### 4.1.3. Kötél

Ennek segítségével ki lehet húzni egy vízbe esett játékost.

#### 4.1.4. Búvárruha

A játékos képes a vízben is mozogni vele, illetve nem veszít testhőt ha vízben tartózkodik.

#### 4.1.5. Lapát

Segítségével 2 egységnyi hó takarítható el, egy egység munkával.

#### 4.1.6. Élelem

Ha a játékos elfogyasztja, a testhője 1-el megnő.

#### 4.1.7. Rakétapisztoly Alkatrész

A játékban 3 darab ilyen megtalálása vezet a játék sikeres befejezéséhez. Az összeszereléshez mindháromnak egy helyen kell lennie.

#### 4.1.8. Iglu

Eszkimó (Játékos) képes építeni, itt átvészélhetők a hóviharak.

### 4.2. Osztályok leírása

#### 4.2.1. BareHands

- A játékos így ás, ha nincs ásója. A kiválasztott cellán csökkennie kell a hó mennyiségnek ásáskor.
- Interfészek:
  - DigStrategy
- Metódusok:

- `bool Dig(Tile t)`: Csökkenti a tile-on található hó mennyiségét. Minden alkalommal fárasztó az ásás, ezért a visszatérési érték mindig `true`.

#### 4.2.2. BareIce

- Ilyen a jégtábla, ha nincs rajta iglu. A jégtáblán nincs védelem a vihar elől.
- Interfészek:
  - `ChillStormStrategy`
- Metódusok:
  - `void Chill(Tile t)`: A paraméterként kapott `t` Tilen álló játékosok testhője csökken.

#### 4.2.3. CantRescue

- A játékos nem tudja kihúzni a csapattársát. A játékos ilyen állapotban van, ha nincs nála kötél.
- Interfészek:
  - `RescueStrategy`
- Metódusok:
  - `void Rescue(Tile water, Tile land)`: Mivel a játékos ebben az állapotban nem tudja megmenteni a csapattársát, ez a fv nem csinál vele semmit.

#### 4.2.4. ChillStormStrategy

- A jégtábla így hűti viharban a játékosokat. Vihar esetén a játékos testhője csökken, a megvalósított stratégia alapján.
- Metódusok:
  - `abstract void Chill(Tile t)`: A stratégiát megvalósító elem dolga implementálni mi történik.

#### 4.2.5. ChillWaterStrategy

- A jégtábla így hűti a vízbe esett játékosokat. Vízben tartózkodás esetén a játékos testhője csökken, a megvalósított stratégia alapján.
- Metódusok:
  - `abstract void Chill(Tile t)`: A stratégiát megvalósító elem dolga implementálni mi történik.

#### 4.2.6. DigStrategy

- A játékos így ás. Ásáskor a cellán a hó mennyiség csökken.
- Metódusok:
  - `abstract bool Dig(Tile t)`: A stratégiát megvalósító elem dolga implementálni mi történik ásáskor. Visszaadja, hogy az ásás fárasztó-e.

#### 4.2.7. DryLand

- A szárazföld nem hűti a játékosokat. A játékos nincsen vízben.
- Interfészek:
  - ChillWaterStrategy
- Metódusok:
  - void Chill(Tile t): A stratégia megvalósítása miatt kér be egy t Tile paramétert, a rajta levő játékosal viszont nem csinál semmit, mert az nincs vízben, nem csökkenti testhőjét.

#### 4.2.8. Empty

- Nincs jégbe fagyott tárgy. Ez az üres eszköz típus, nem képes semmi extra tulajdonságot biztosítani a tulajdonosnak.
- Interfészek:
  - Item
- Metódusok
  - void GiveTo(Player p): A paraméterként kapott játékost nem ruházza fel extra tulajdonsággal, mivel épp nincs itt jégbe fagyott tárgy.

#### 4.2.9. Eskimo

- Játékos fajta. 5 egységnyi testhővel kezd. Képes iglut építeni. A játékos irányítja.
- Ősosztályok:
  - Player
- Metódusok:
  - void BuildIgloo(): Épít egy iglut a mezőre, amin áll. Az iglu megvéd majd a hóvihartól. Beállítja a mező hóvihar stratégiáját Iglusra.

#### 4.2.10. Food

- Élelem, amit a játékos meg tud enni, hogy növelje a testhőjét. Élelem a pályán lesz található.
- Interfészek:
  - Item
- Metódusok:
  - void GiveTo(Player p): A paraméterként kapott játékos kap egy élelmet, az bekerül az élelemtárolójába.

#### 4.2.11. FoodStore

- A játékos ebben a zsebben tárolja az élelmet.
- Attribútumok:
  - count: int: Hány élelem van a játékosnál.
- Metódusok:

- void feed(Player p): Játékos testhője megnő, az élelem mennyisége csökken, mivel a játékos meg-  
eszi azt.
- void DecrementCount(): Csökkenti a benne található elemek számát.
- void Gain(): növeli a benne található elemek számát.

#### 4.2.12. Game

- Interface a Model és a Controller között. A játékmesterhez tartozó működést valósítja meg. Felelős a játékban lévő objektumok tárolásáért és létrehozásáért.
- Attribútumok:
  - players: Player[3..\*]: Tárolja a játékosokat.
  - icefield: Tile[1..\*]: Tárolja a pályát alkotó elemeket.
- Metódusok:
  - void AddTile(t: Tile): Hozzáad egy cellát a játékhoz.
  - void AddPlayer(pl: Player): Hozzáad egy játékost a játékhoz.
  - Tile CreateIce(): Létrehoz egy jégtáblát. Ez a metódus az init szekvencia része.
  - Tile CreateUnstableIce(): Létrehoz egy instabil jégtáblát. Ez a metódus az init szekvencia része.
  - Tile CreateSea(): Létrehoz egy vizet. Ez a metódus az init szekvencia része.
  - Tile CreateHole(): Létrehoz egy lyukat: olyan vizet amit hó fed. Ez a metódus az init szekvencia része.
  - Player CreateEskimo(): Létrehoz egy eszkimó játékost. Ez a metódus az init szekvencia része.
  - Player CreatePolarExplorer(): Létrehoz egy sarkkutató játékost. Ez a metódus az init szekvencia része.
  - void GameOver(): Ha vége a játéknak, szól a Controllernek, hogy veszítettünk. Külső metódus.
  - void Turn(): Ezt a metódust a Controller hívja körönként, a körök vezénylezésére szolgál.
  - void Victory(): Ha vége a játéknak, szól a Controllernek, hogy nyertünk. Külső metódus.

#### 4.2.13. Igloo

- Ezen a jégtáblán iglu áll, a játékosok védve vannak a vihartól. Az ilyen táblán nem csökken a viharban a rajta állók testhője.
- Interfészek:
  - ChillStromStrategy
- Metódusok:
  - void Chill(Tile t): A paraméterként kapott cellán álló játékosok testhője nem csökken, mivel igluban vannak.

#### 4.2.14. Item

- Tárgy, a játékos képes ilyeneket felvenni a cellákról. A tárgyak képesek a játékosok képességeket adni. A tárgyak alapvetően jégbe fagyva vannak a pályán.
- Metódusok:
  - void GiveTo(p: Player): A játékos kap valamilyen tárgyat, az Item interfészt megvalósító tárgyak felüldefiniálják ezt.



## 4.2.15. Naked

- A játékos védtelen a hideg vízzel szemben. A játékos ha így esik vízbe és nem menekít ki megfullad.
- Interfészek:
  - WaterResistanceStrategy
- Metódusok:
  - void Chill(Player p): Játékosnak nincsen ereje a vízben úszni bűvárruha nélkül.

## 4.2.16. Part

- Jégbefagyott alkatrész. Csak akkor ásható ki, ha nincs rajta hó.
- Interfészek:
  - Item
- Metódusok:
  - void GiveTo(Player p): A játékos tárolójába kerül egy darab a rakétapisztolyból.

## 4.2.17. PartStore

- A játékos ebben a zsebben tárolja az alkatrészeket.
- Attribútumok:
  - count: int: Tárolja hány darab alkatrész van belőle a játékosnál.
- Metódusok:
  - void Gain(PartStore ps): Átveszi az alkatrészeket a paraméterként kapott alkatrésztárolóból.
  - void Gain(int n): Megnö az alkatrészek száma, ami a játékosnál van.

## 4.2.18. Player

- Játékos osztály, amit a felhasználó irányít a grafikus felületen keresztül. Ilyen típussal nem lehet játszani, csak a leszármazottakkal. Felelőssége a játékos által a controlleren keresztül kiadott műveletek elvégzése. Tárolja a játékos jelenlegi állapotát.
- Attribútumok:
  - bodyTemp: int: Jelzi a játékos jelenlegi hőmérsékletét, ha 0 akkor megfagy → játék vége.
  - currentTile: Tile: A játékos ismeri a mezőt amin éppen áll.
  - inventory: Item[\*]: Tárolja a játékos tárgyait, amik képességekkel tudjak felruházni őt.
  - digStrategy: DigStrategy: Eldönti hogyan képes ásni a játékos.
  - energy: int: Számlálja mennyit mozogott az adott körben a játékos.
  - foodStore: FoodStore: Tárolja a játékos ételeit.
  - game: Game: A játékos ismeri a játékot.
  - partStore: PartStore: Tárolja a játékos rakéta alkatrészeit.
  - rescueStrategy: RescueStrategy: Eldönti, hogy megmenthet egy játékos egy másikat a vízbeesés után.

- waterResistanceStrategy: WaterResistanceStrategy: Eldönti, hogy a játékos hogyan viselkedik víz-beesés esetén.
- Metódusok:
  - void AssembleFlare(): Összerakja a játék végéhez szükséges rakéta pisztolyt. 1 munkaegység
  - void Chill(): A testhő 1-el csökken, ha 0 alá megy → GameOver.
  - void DecrementEnergy(): Az energiát csökkentő helper metódus.
  - void Dig(): Ezt a metódust a Controller hívja. A játékos havat ás. 1 munkaegység
  - void EatFood(): Ezt a metódust a Controller hívja. A játékos eszik. A testhője megnő 1-el.
  - void Pickup(): Ezt a metódust a Controller hívja. A játékos felvesz egy tárgyat. 1 munkaegység
  - void Equip(inventorySlot: int): Ezt a metódust a Controller hívja. A játékos kiválaszt egy tárgyat használatra.
  - void PlaceOn(Tile t): Init szekvencia része. RopeRescue szekvencia része. Ráak egy játékost egy másik Tile-ra.
  - void RescueTeammate(direction: int): Ezt a metódust a Controller hívja. A játékos kiment egy másikat a vízből. 1 munkaegység
  - void ResistWater(): A játékos testhője a WaterResistance szerint változik.
  - void Step(direction: int): Ezt a metódust a Controller hívja. A játékos lép, ha van még hozzá elég energiája. 1 munkaegység
  - void ToFoodStore(): Élelem megtalálásához helper metódus.

#### 4.2.19. PolarExplorer

- Játékos fajta. 4 egységnyi testhővel kezd. Képes megnézni egy cella teherbíró képességét. A játékos irányítja.
- Ősosztályok:
  - Player
- Metódusok:
  - int Examine(direction: int): A játékos megnézheti, hogy egy adott irányban lévő Tile-nak mennyi a teherbírása.

#### 4.2.20. RescueStrategy

- A játékos így húzza ki csapattársát a vízből. A játékos így képes megmenteni a vízbe esett csapattársát a szomszédos celláról, a megvalósított stratégia alapján. Kötél szükséges a másik játékos megmentéséhez.
- Metódusok:
  - abstract void Rescue(Tile water, Tile land): A stratégiát megvalósító elem dolga implementálni mi történik.

#### 4.2.21. Rope

- Jégbe fagyott kötél. Ezzel lehet megmenteni a vízbe esett csapattársat a szomszédos celláról.
- Interfészek:
  - Item

- Metódusok
  - void GiveTo(Player p): A játékos kap egy kötelet. Az bekerül az inventoryjába és a megfelelő stratégiájához is a kötél által adott képesség.

#### 4.2.22. RopeRescue

- A játékos kihúzza csapattársát a vízből. A játékos így menti meg a szomszédos cellán vízbe esett csapattársát.
- Interfészek:
  - RescueStrategy
- Metódusok:
  - void Rescue(Tile water, Tile land): A vízben lévőek közül egyvalaki rákerül a kihúzó játékos cellájára.

#### 4.2.23. ScubaGear

- Jégbe fagyott búvárruha. Ezzel lehet életben maradni a vízben.
- Interfészek:
  - Item
- Metódusok:
  - void GiveTo(): A játékos búvárruhát kap. Az bekerül az inventoryjába és a megfelelő stratégiája helyére is a búvárruha által adott képesség.

#### 4.2.24. ScubaWearing

- A játékos testhője nem csökken a vízben. A játékos nem hal bele, ha a vízben marad.
- Interfészek:
  - WaterResistanceStrategy
- Metódusok:
  - void Chill(p: Player): A játékost nem hűti a víz, mivel búvárruhát visel.

#### 4.2.25. Sea

- Ez a cella tenger, hűti a játékosokat.
- Interfészek:
  - ChillWaterStrategy
- Metódusok:
  - void Chill(Tile t): Minden rajta álló testhője csökken a WaterResistanceStrategy szerint.

## 4.2.26. Shovel

- Jégbe fagyott ásó. Ezzel lehet több havat eltakarítani a celláról.
- Interfészek:
  - Item
- Metódusok:
  - void GiveTo(): A játékos ásót kap, ami bekerül az inventoryjába és a megfelelő stratégiájához is bekerül az ásó által adott képesség.

## 4.2.27. ShovelDig

- Egyszer lehet ásni vele fáradtság nélkül is.
- Interfészek:
  - DigStrategy
- Attribútumok:
  - lastUsed: bool: Volt-e már használva a körben.
- Metódusok:
  - bool Dig(Tile t): Csökkenti a tile-on található hó mennyiségét. Minden második alkalommal fárasztó.

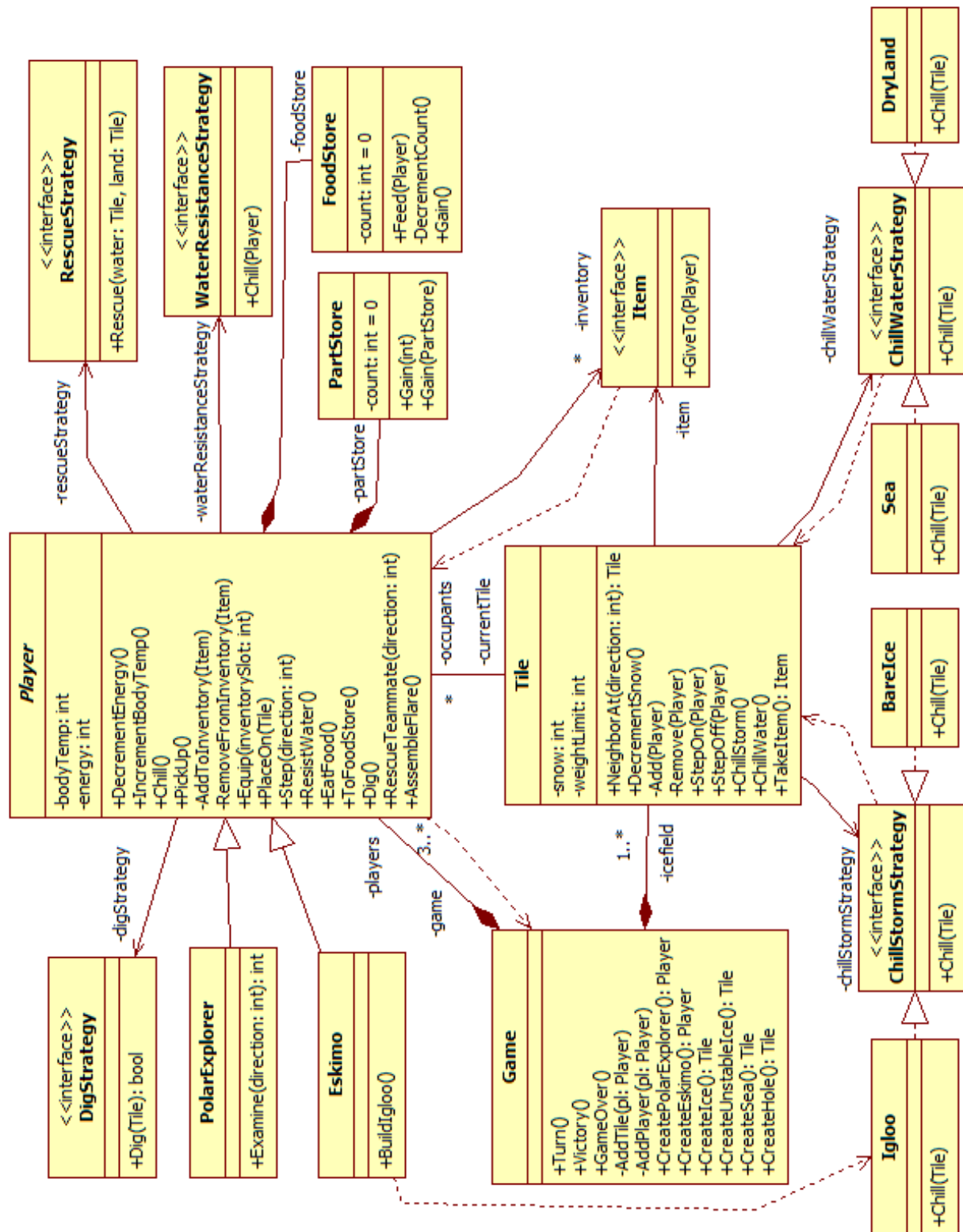
## 4.2.28. Tile

- Cella, ilyenekből áll a jégmező ahol a játékosok játszanak.
- Attribútumok:
  - chillStormStrategy: ChillStormStrategy: Eldönti, kinek változik a testhője vihar esetén.
  - chillWaterStrategy: ChillWaterStrategy: Eldönti, kinek változik a testhője víz esetén.
  - item: Item: Ezt a tárgyat lehet kiásni belőle.
  - neighborTiles: Tile[\*]: Szomszédos cellákat ismer.
  - occupants: Player[\*]: Rajta lévő játékosok.
  - snow: int: Rajta lévő hó mennyiség.
  - weightLimit: int: Rajta lévő játékosok számának maximuma.
- Metódusok:
  - void ChillStorm(): Ezt a metódust a Controller hívja viharban. Hűti a játékosokat, ha nincsenek igluban.
  - void ChillWater(): Ezt a metódust a Controller hívja körönként. Hűti a játékosokat, ha ez a cella víz.
  - void DecrementSnow(): A hó mennyiséget csökkentő helper függvény.
  - Item TakeItem(): A játékos megkapja a tartalmazott tárgyat.
  - Tile NeighborAt(direction): Visszaadja az adott irányban szomszédos cellát.
  - StepOn(Player): Játékos rálép a cellára, ha többen vannak mint a korlát, a jégtábla átfordul. A függvény futása során beállítja a megfelelő adattagokat az új értékekre.
  - StepOff(Player): Járékos lelép a celláról. A függvény futása során beállítja a megfelelő adattagokat az új értékekre.

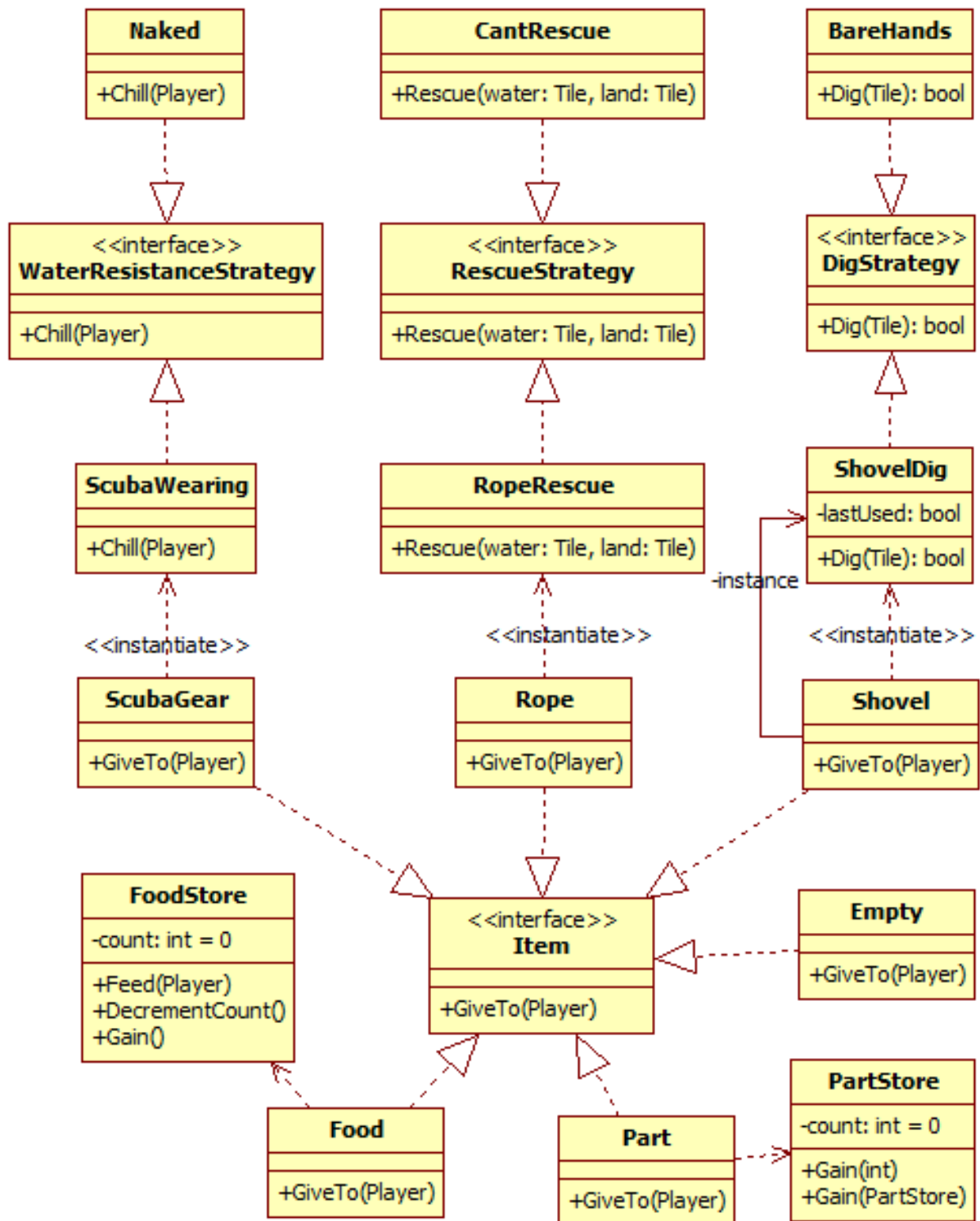
## 4.2.29. WaterResistanceStrategy

- Így reagál a játékos a hideg vízre. A vízben búvárruh nélkül nem lehet mozogni. A vízből ha búvárruha nélkül nem húznak ki, nem lehet életben maradni.
- Metódusok:
  - `abstract void Chill(Player p)`: A stratégiát megvalósító elem dolga implementálni mi történik.

### 4.3. Statikus struktúra diagramok

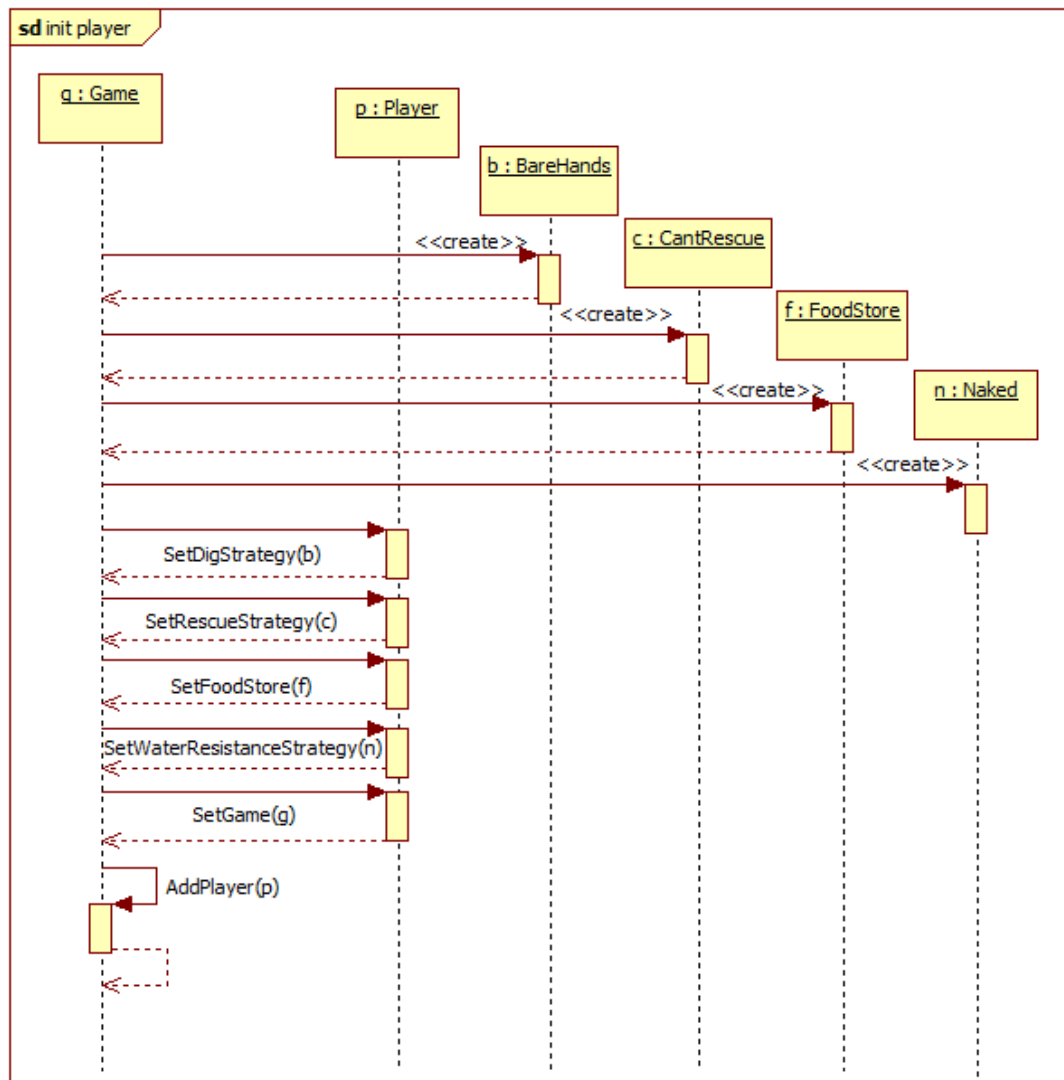


4.1. ábra. Osztálydiagram 1.



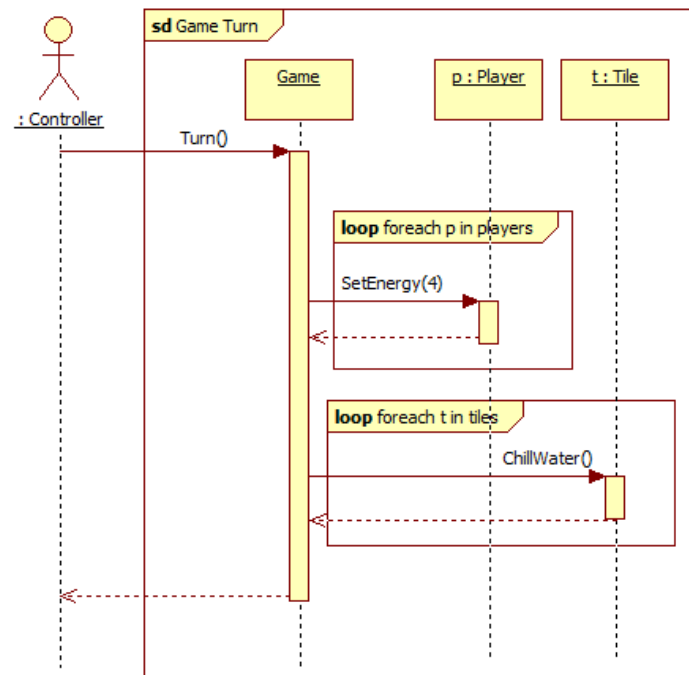
4.2. ábra. Osztálydiagram 2.

## 4.4. Szekvencia diagramok

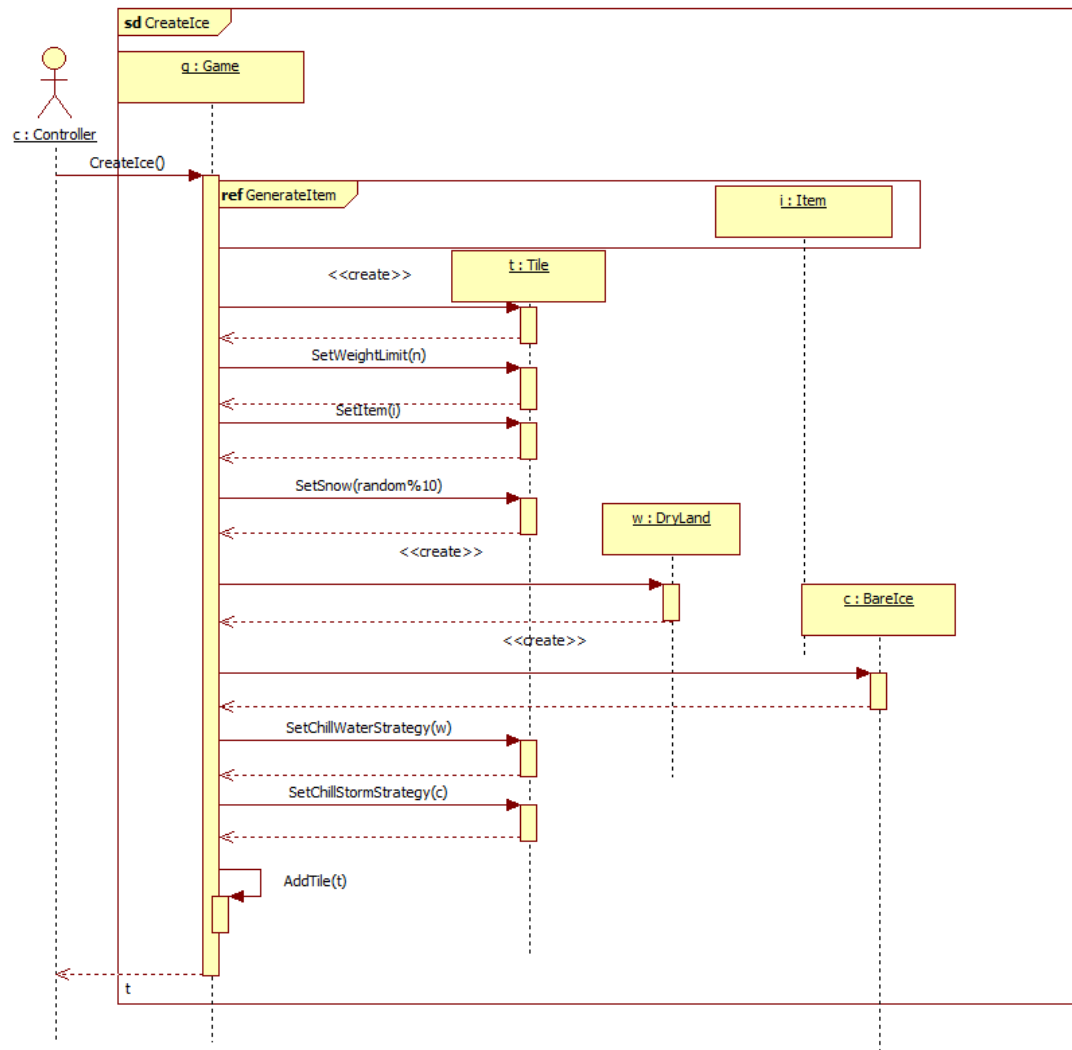


4.3. ábra. Game.CreateEskimo(), Game.CreatePolarExplorer()

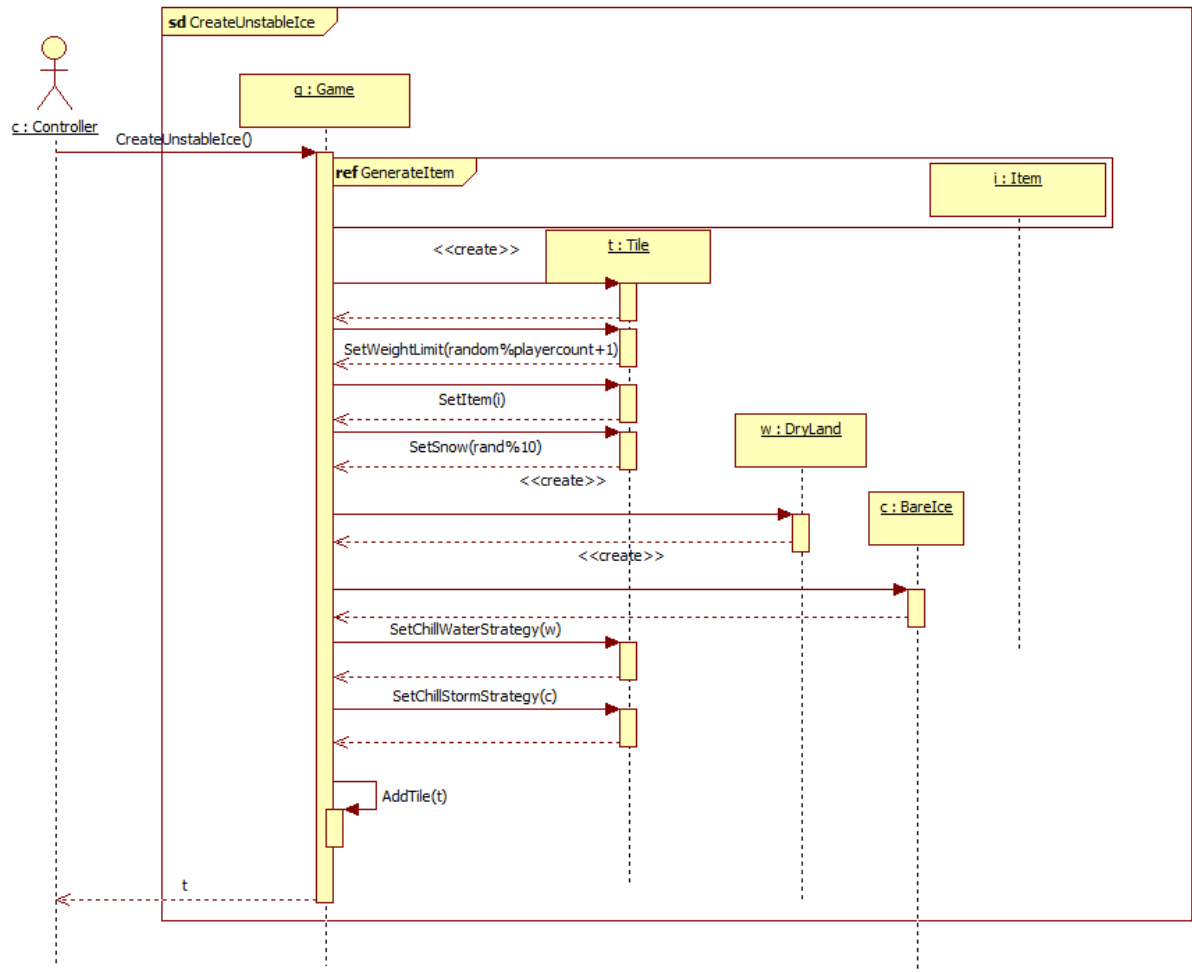




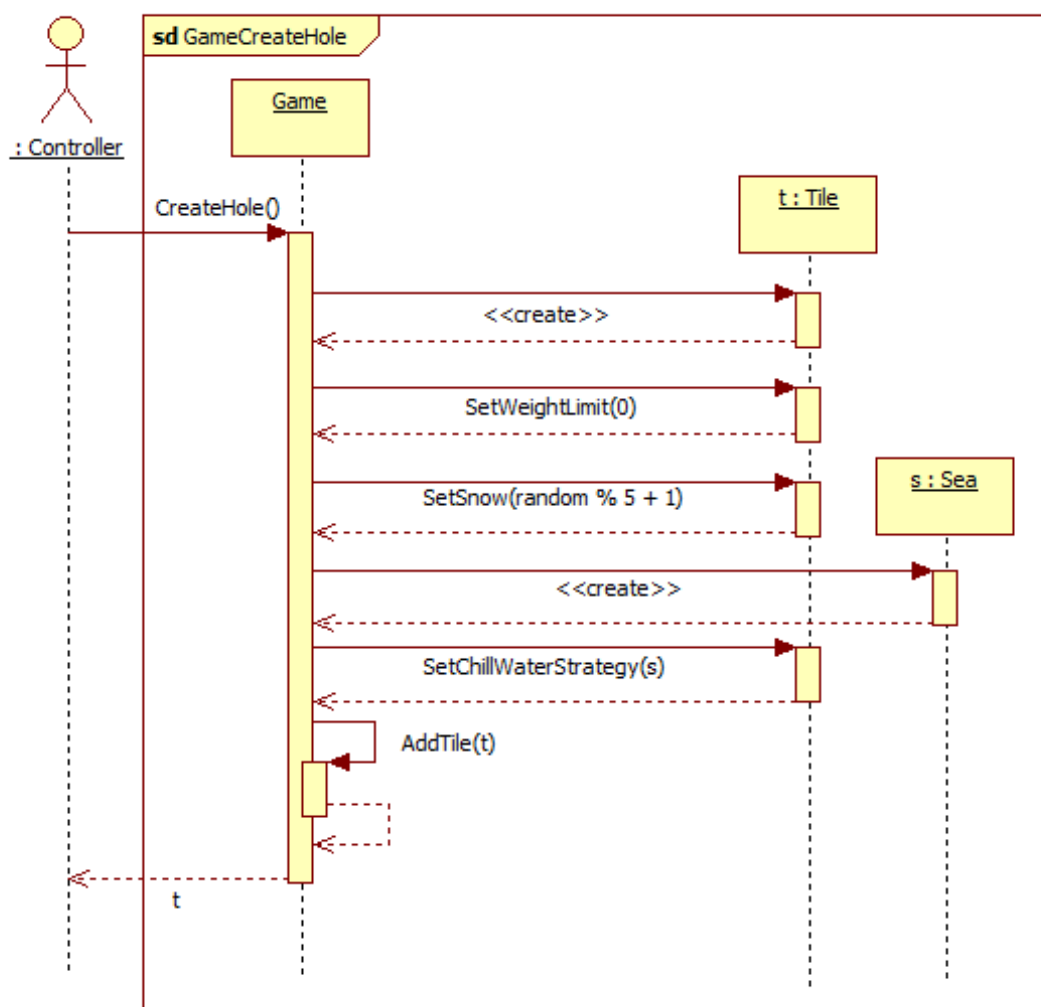
4.4. ábra. Game.Turn()



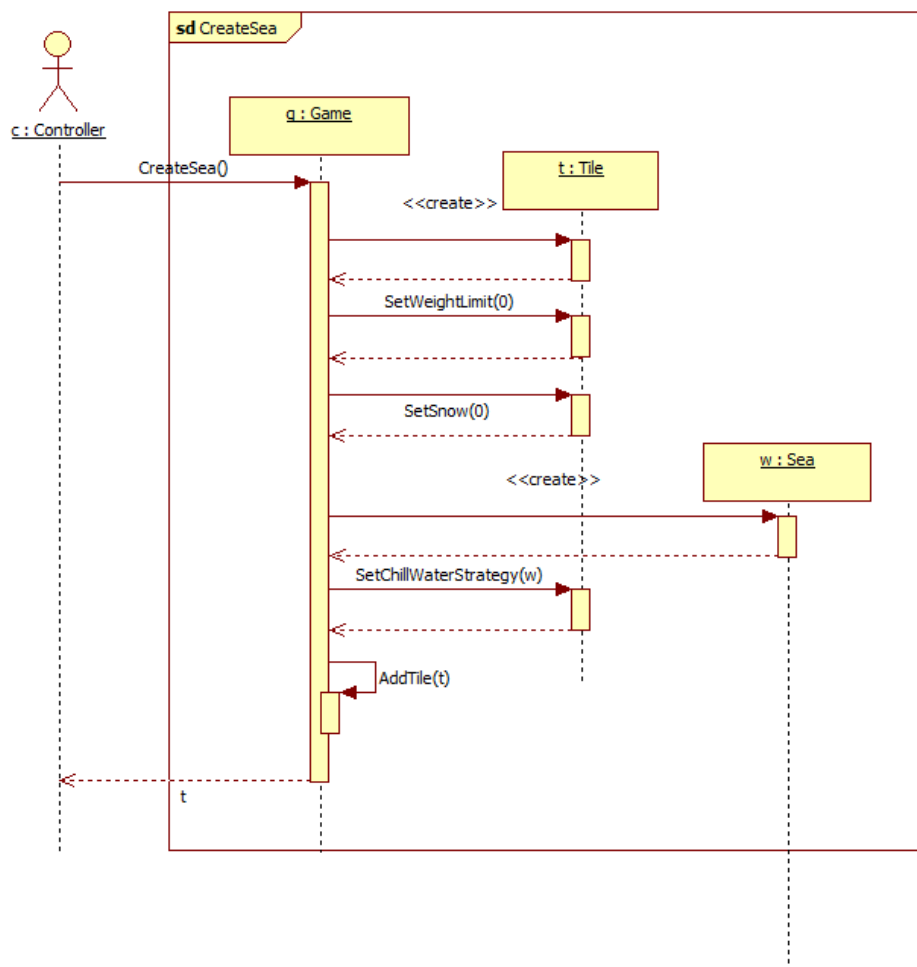
4.5. ábra. Game.CreateIce()

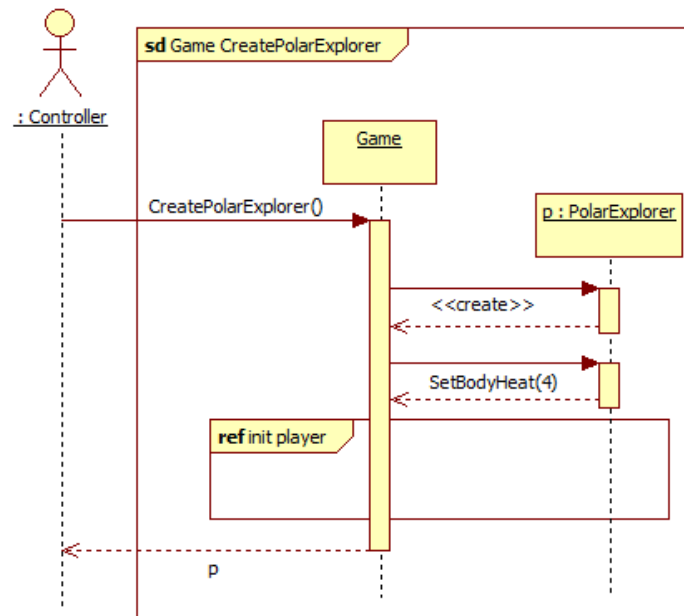


4.6. ábra. Game.CreateUnstableIce()

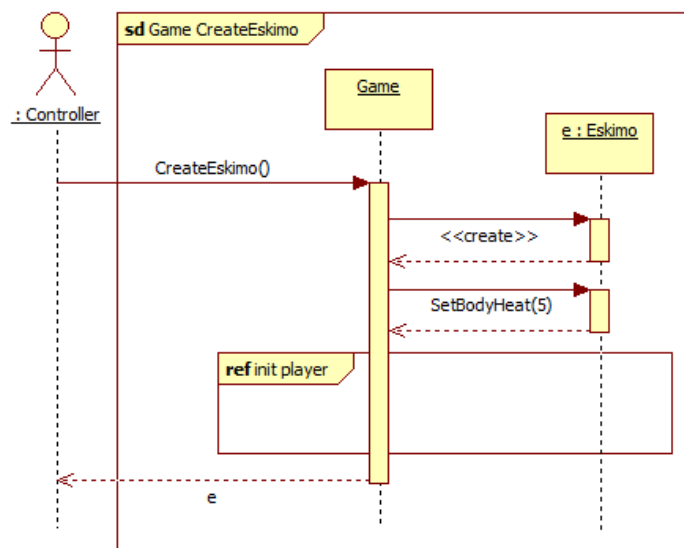


4.7. ábra. Game.CreateHole()

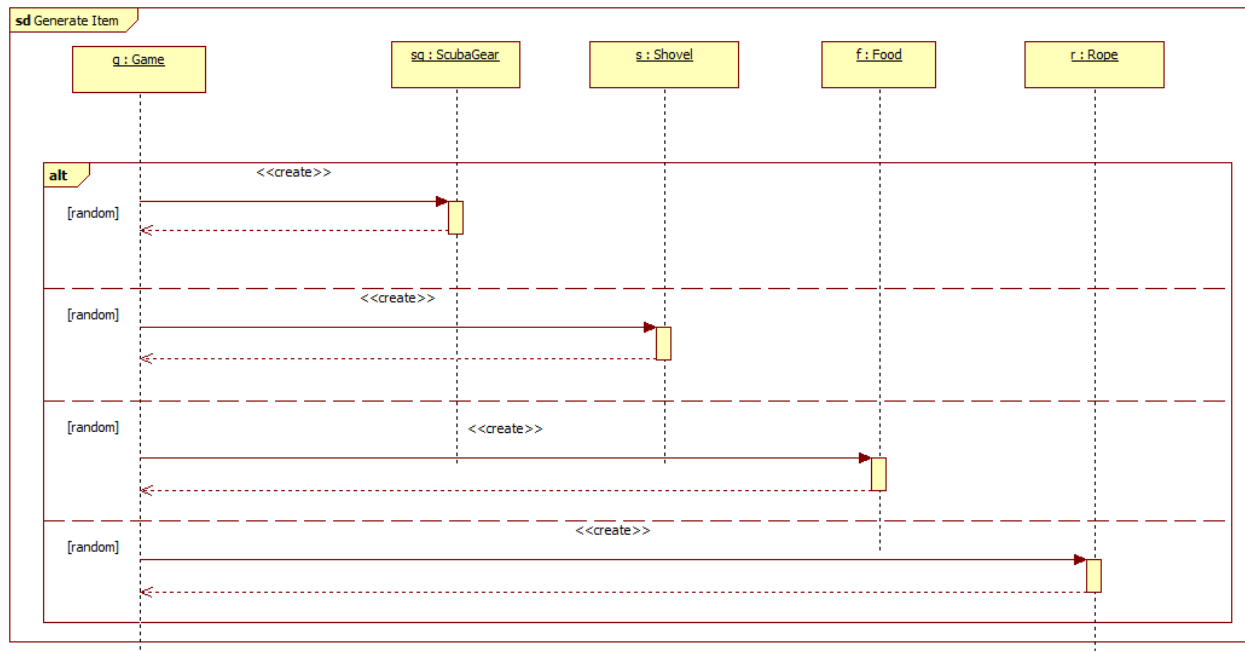
4.8. ábra. `Game.CreateSea()`



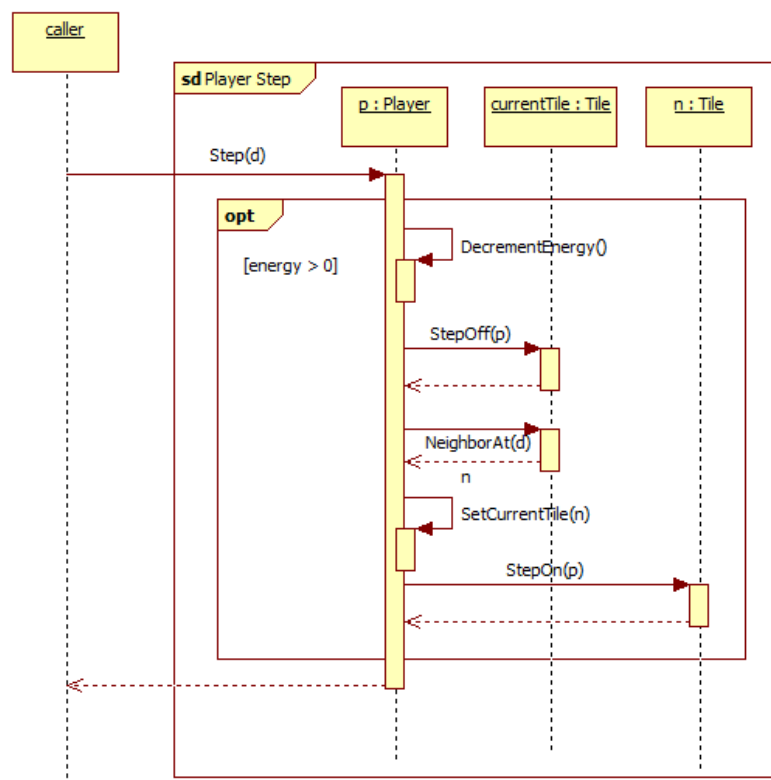
4.9. ábra. Game.CreatePolarExplorer()



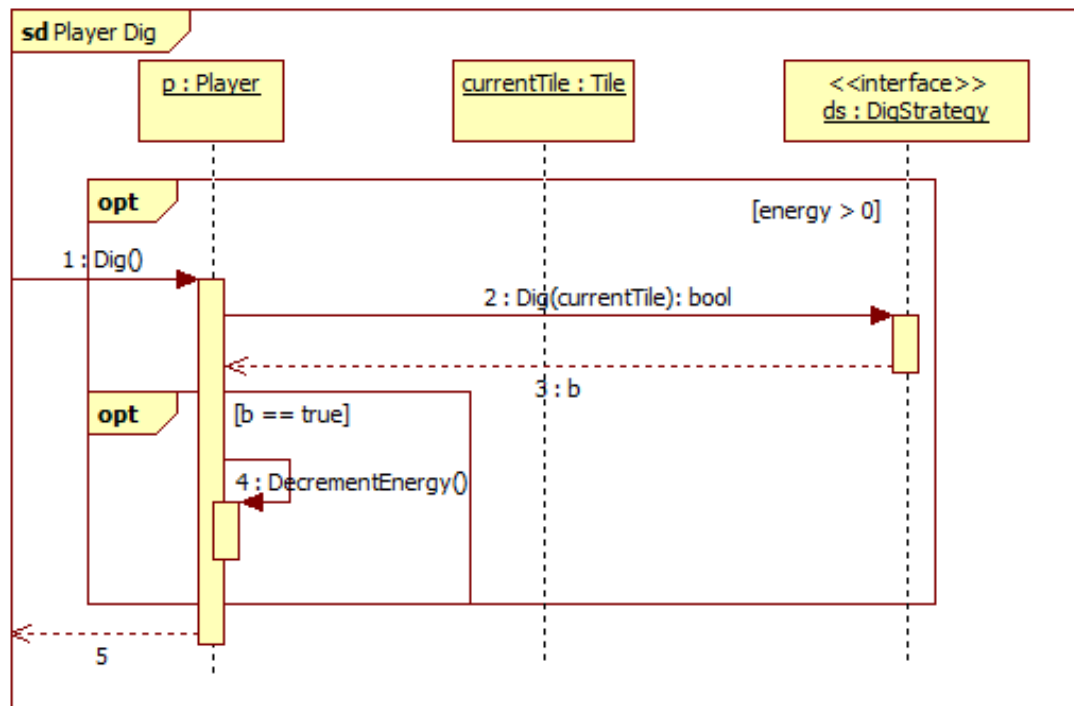
4.10. ábra. Game.CreateEskimo()



4.11. ábra. Game.GenerateItem()

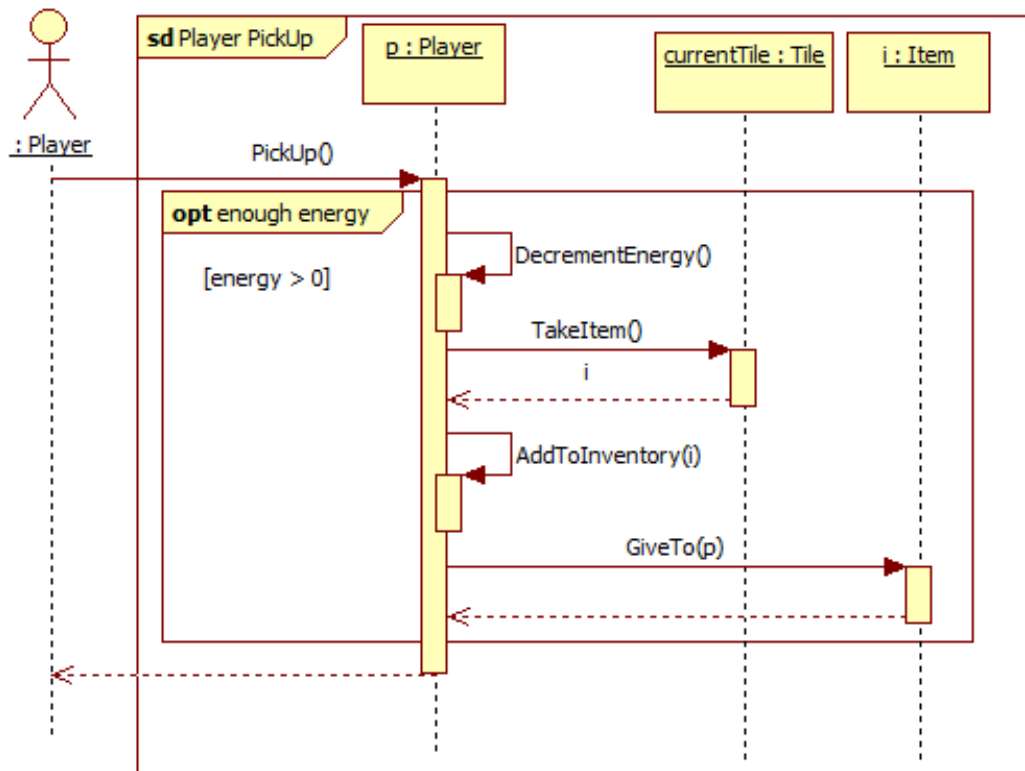


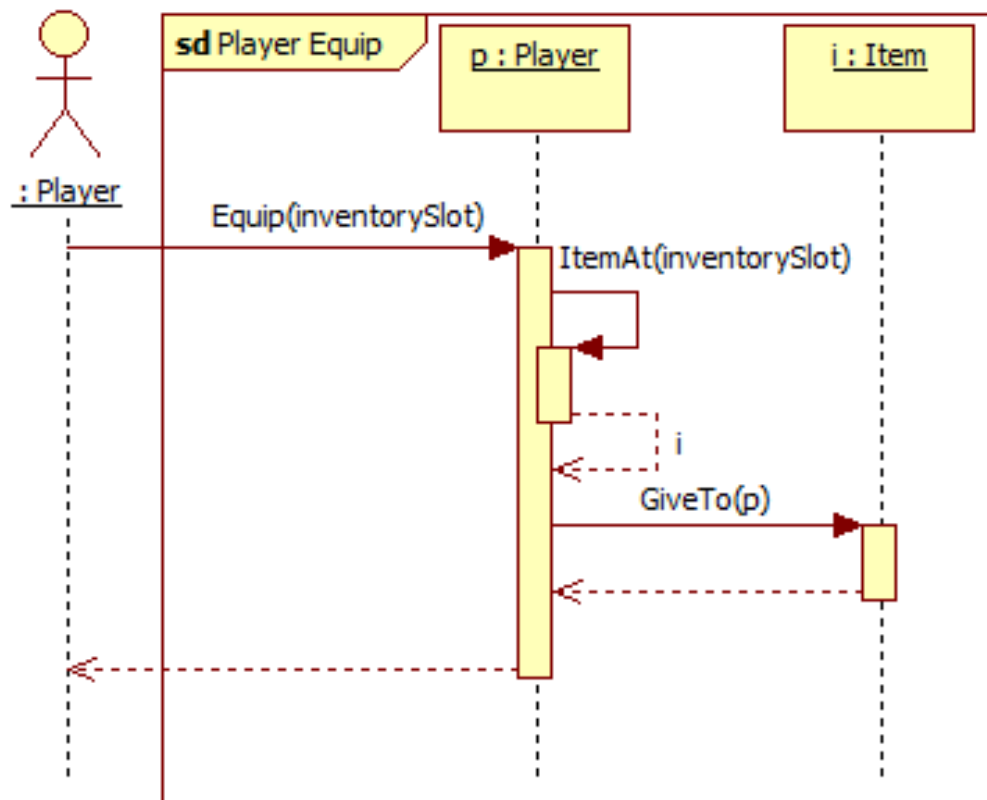
4.12. ábra. Player.Step(direction: int)



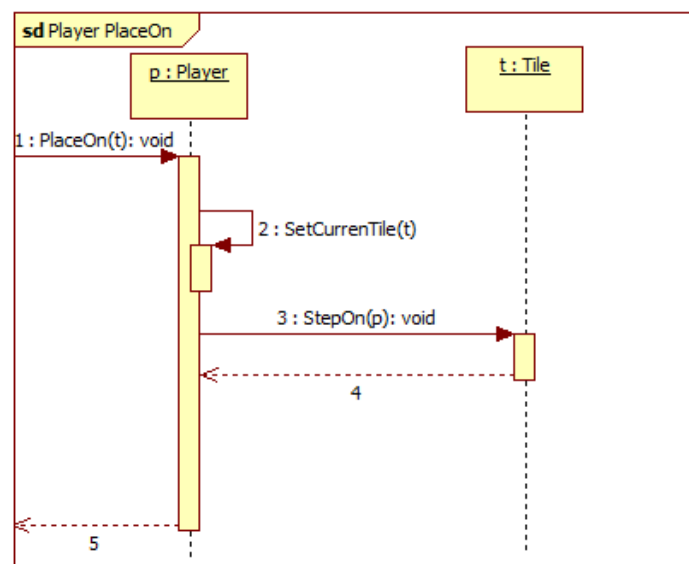
4.13. ábra. Player.Dig()



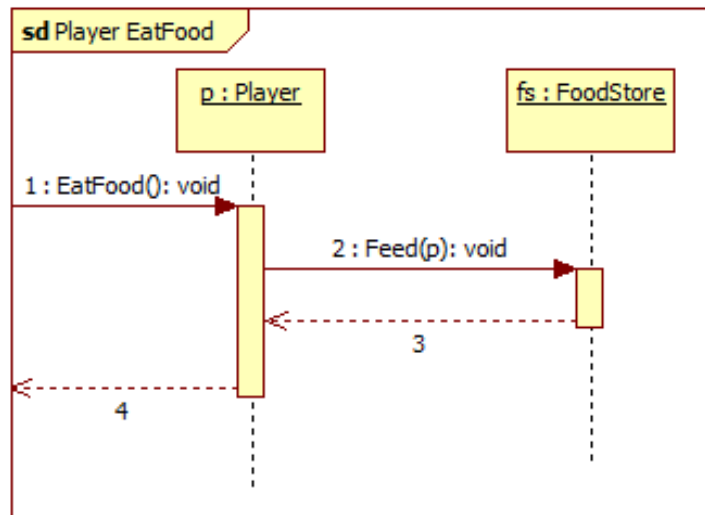
4.14. ábra. `Player.PickUp()`



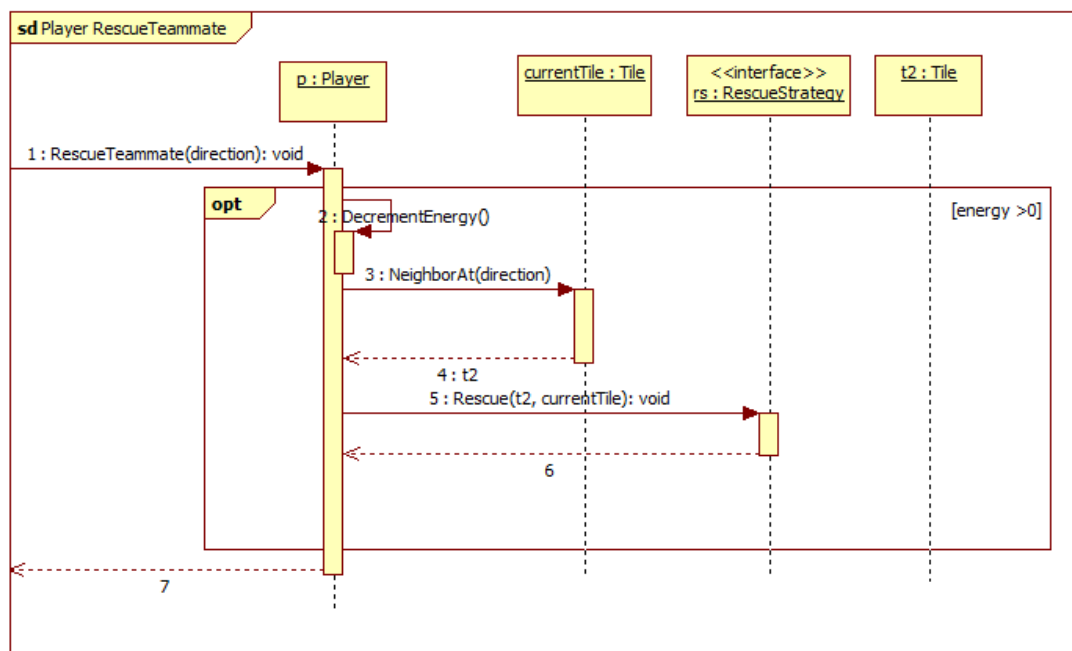
4.15. ábra. Player.Equip(int)



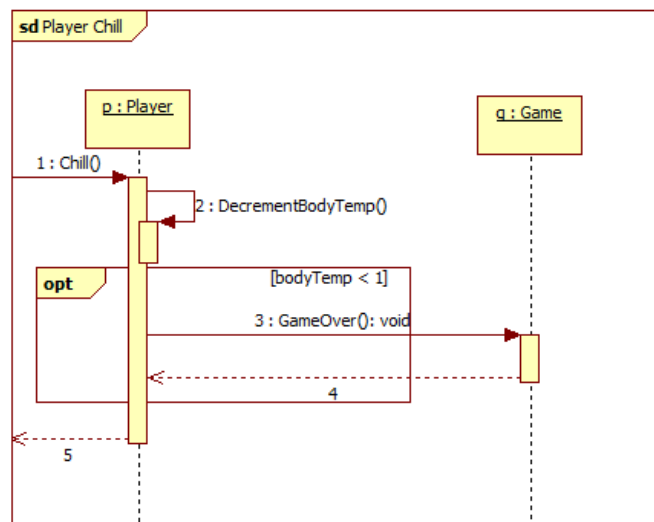
4.16. ábra. Player.PlaceOn(Tile)



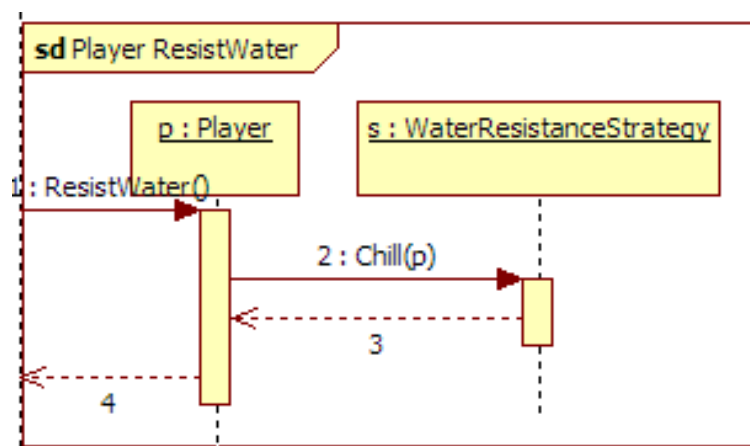
4.17. ábra. Player.EatFood()



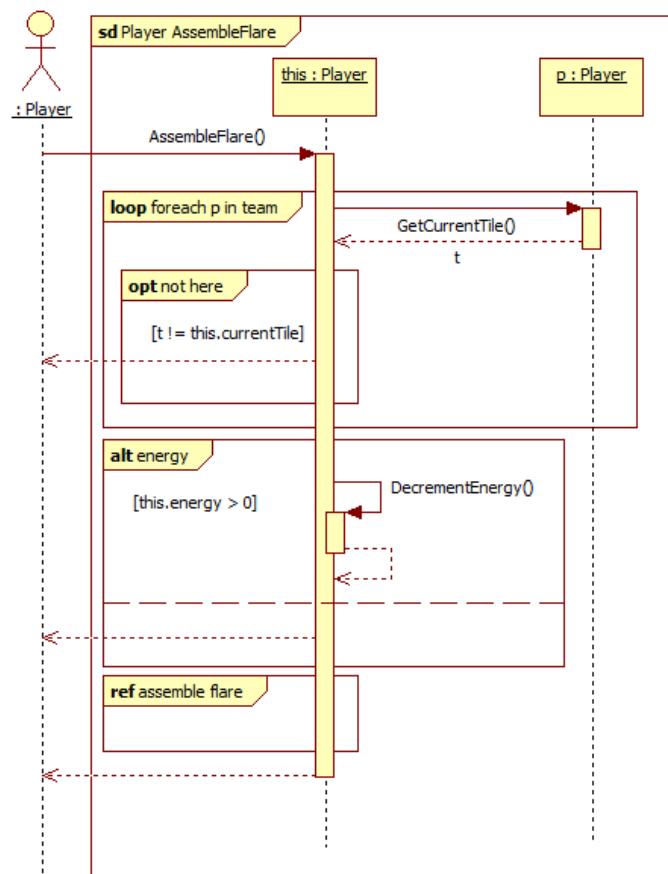
4.18. ábra. Player.RescueTeammate(direction: int)



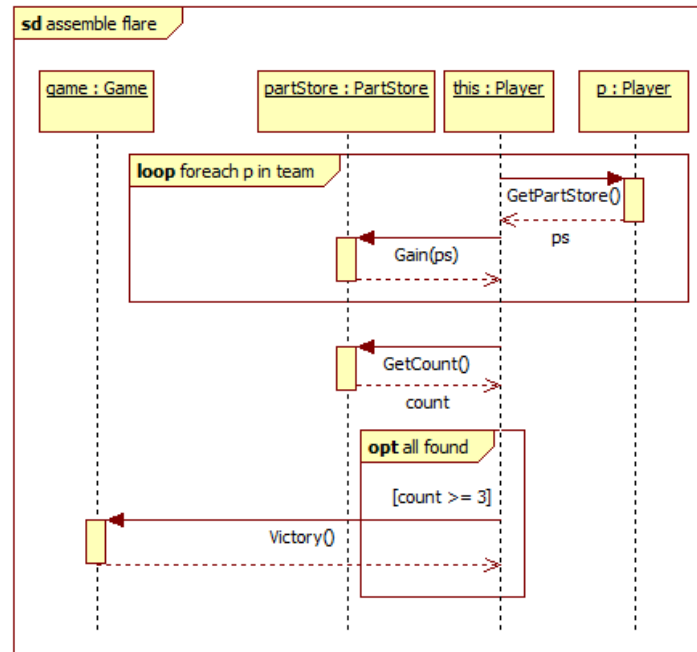
4.19. ábra. Player.Chill()



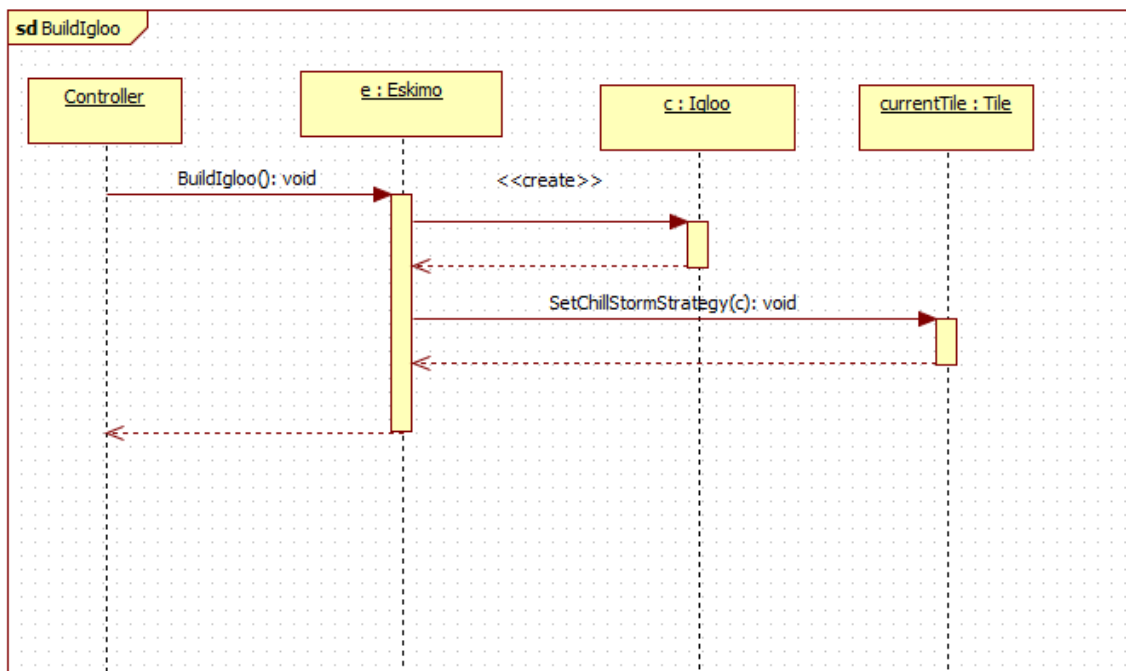
4.20. ábra. Player.ResistWater()



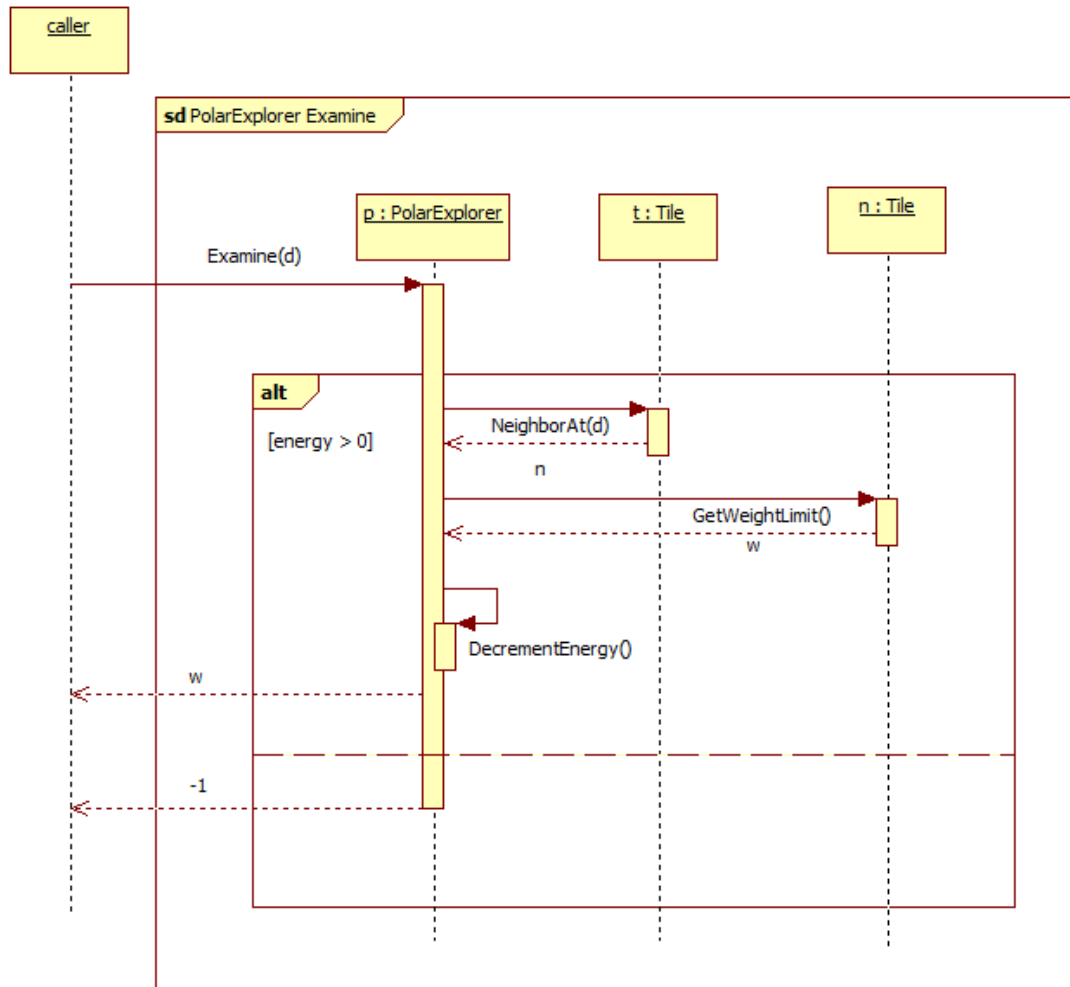
4.21. ábra. Player.AssembleFlare()

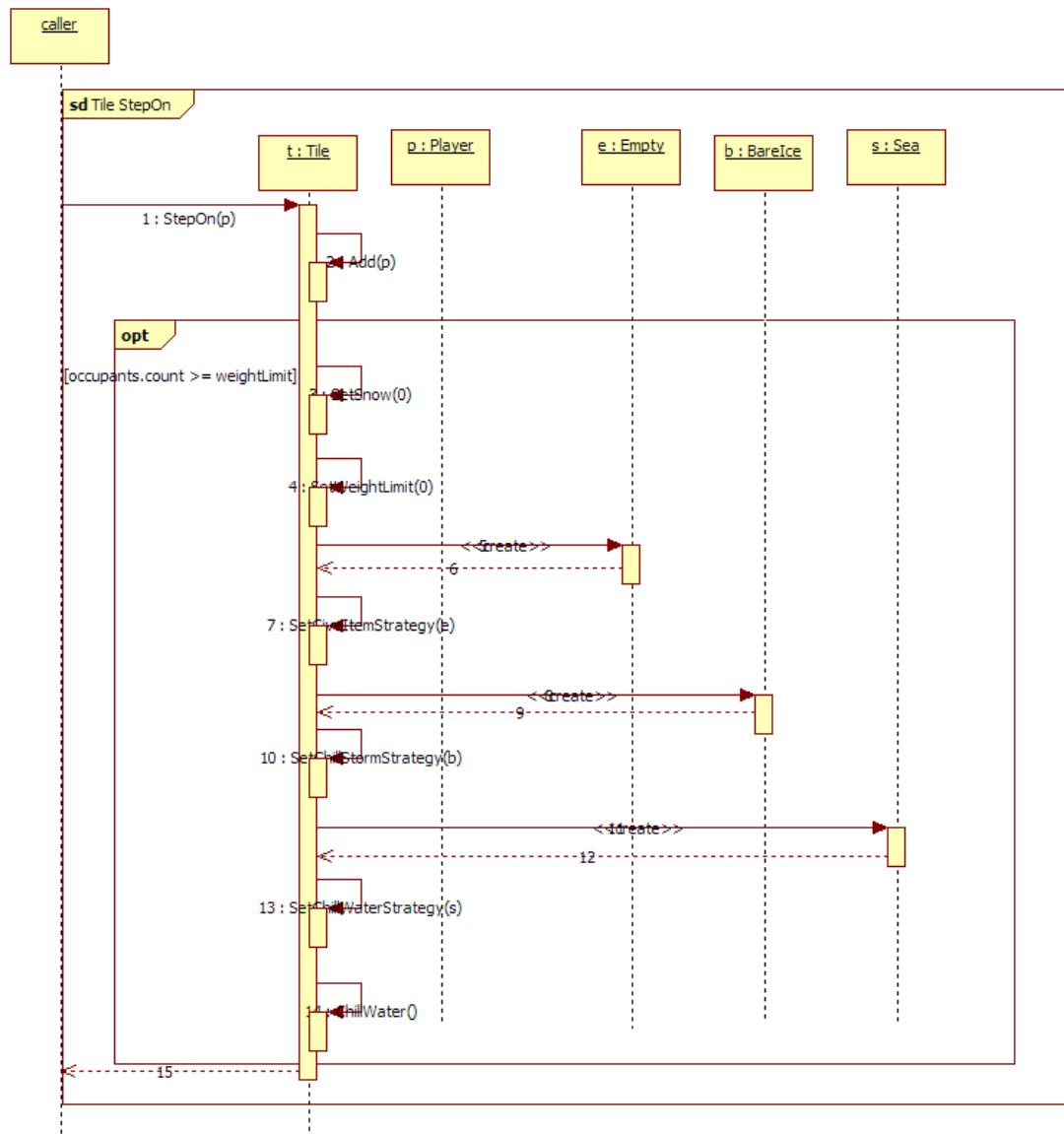


4.22. ábra. Player.AssembleFlare()



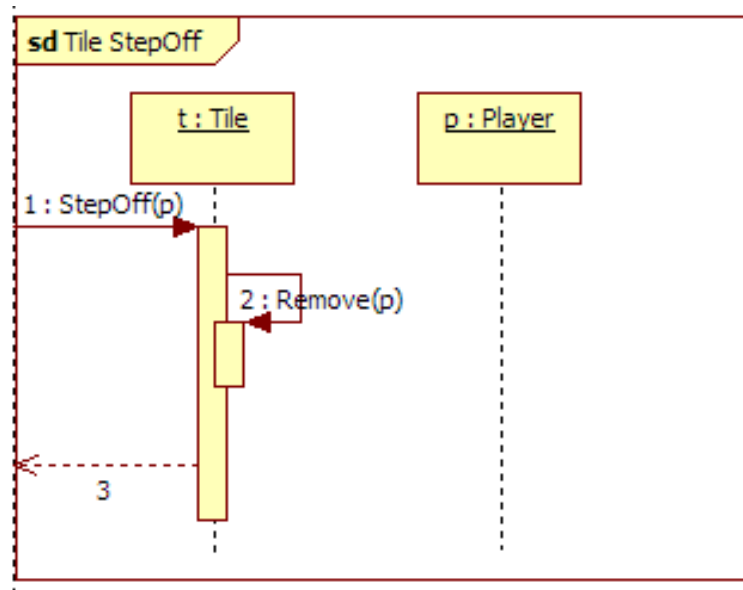
4.23. ábra. Eskimo.BuildIgloo()

4.24. ábra. `PolarExplorer.Examine(direction: int)`

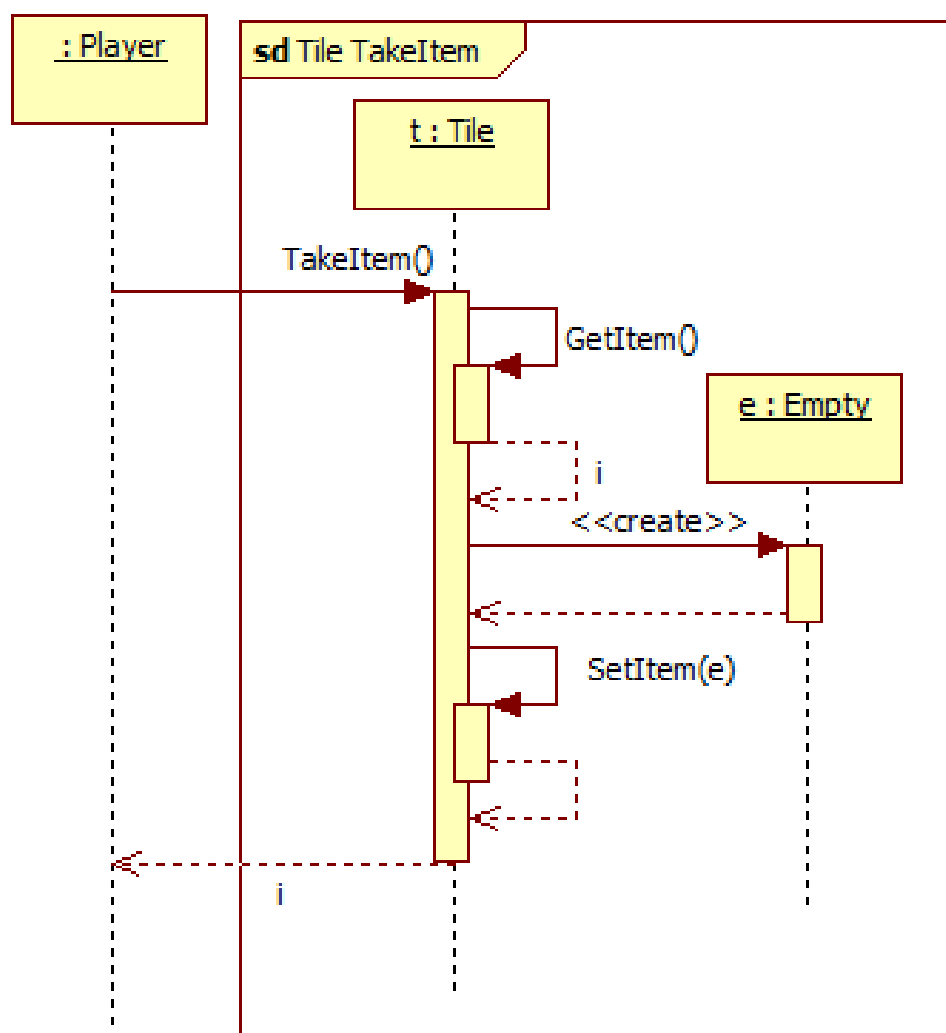


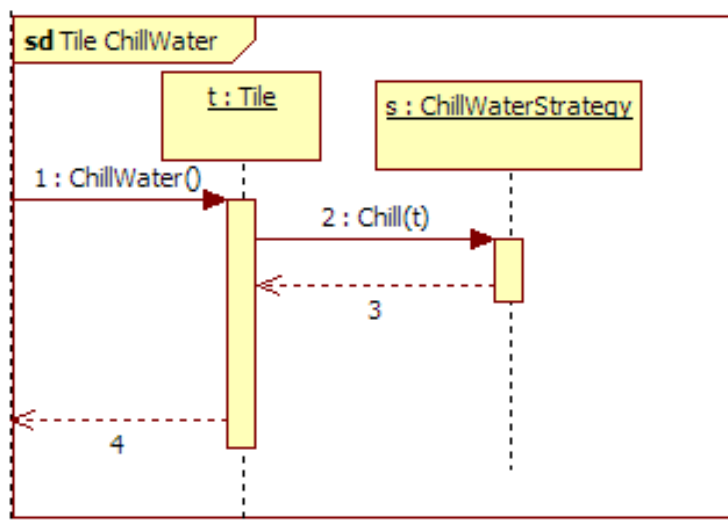
4.25. ábra. Tile.StepOn(Player)



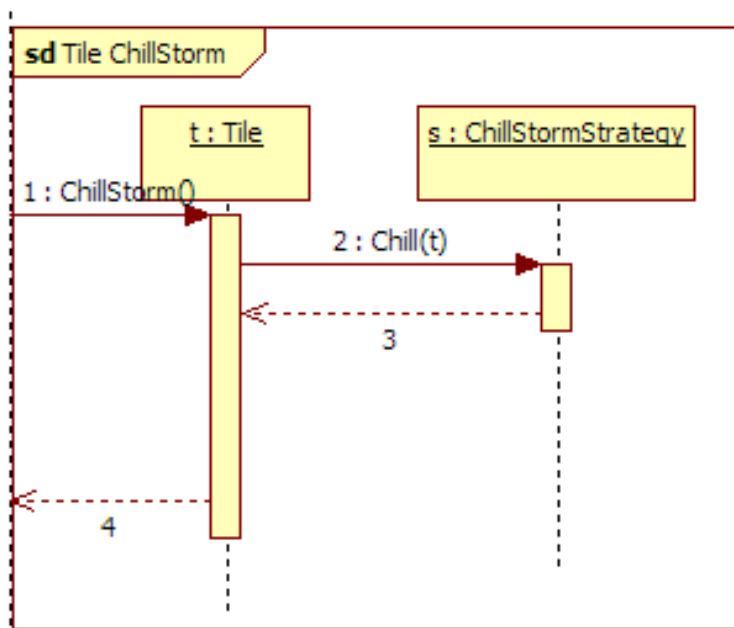


4.26. ábra. Tile.StepOff(Player)

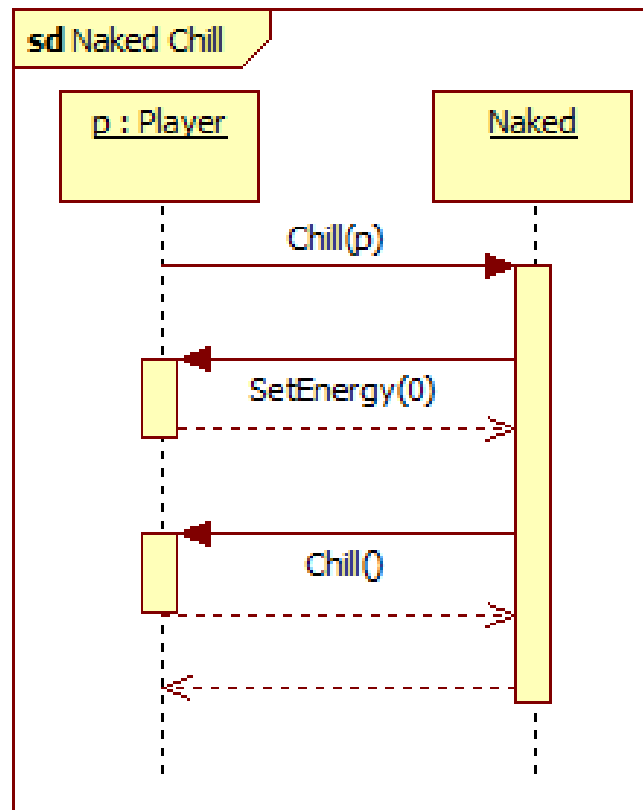
4.27. ábra. `Tile.TakeItem()`



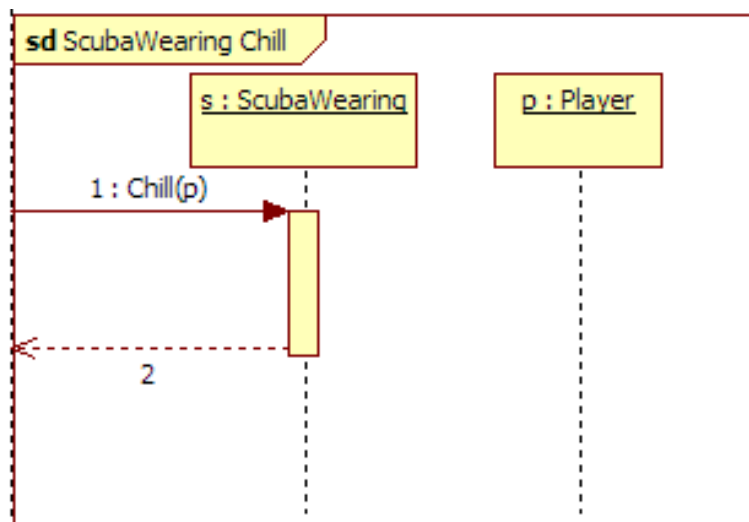
4.28. ábra. Tile.ChillWater()



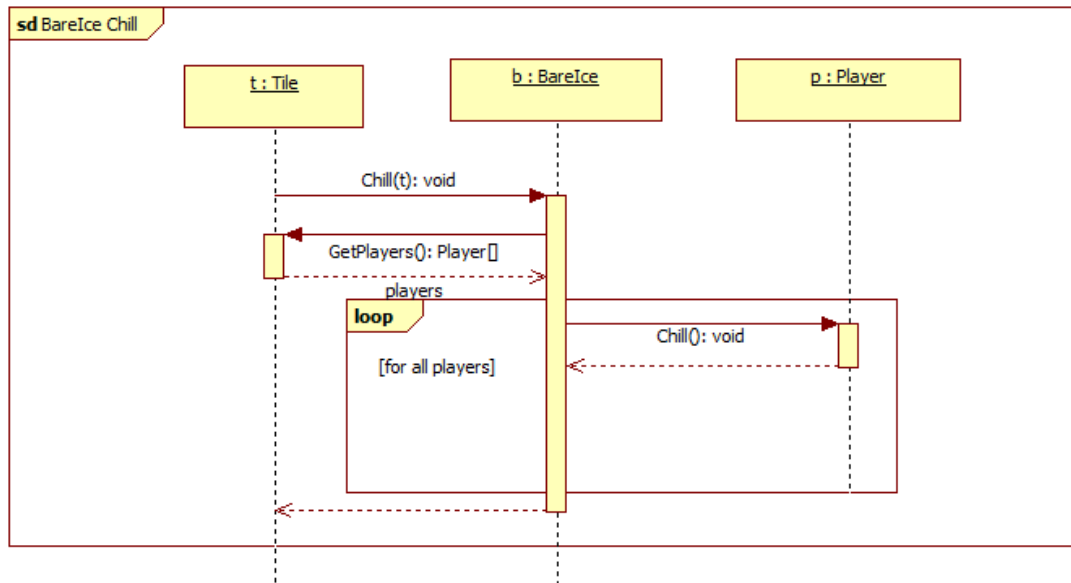
4.29. ábra. Tile.ChillStorm()



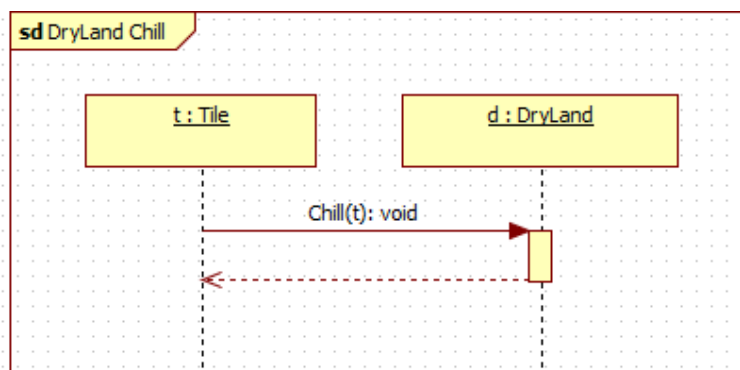
4.30. ábra. Naked.Chill(Player)



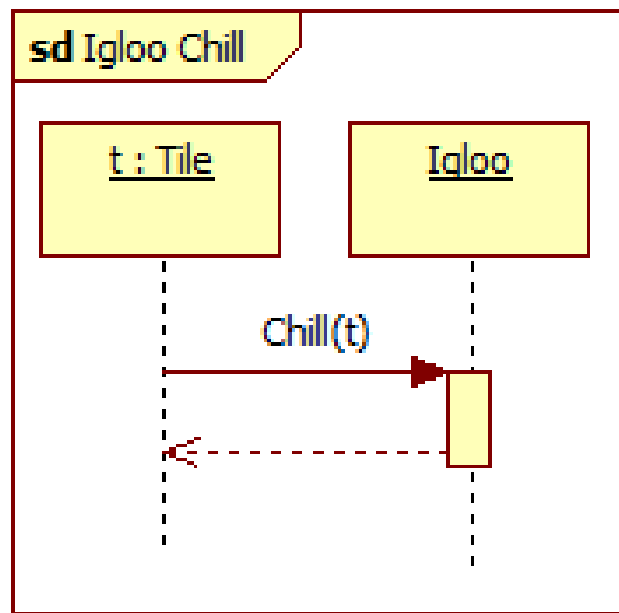
4.31. ábra. ScubaWearing.Chill(Player)



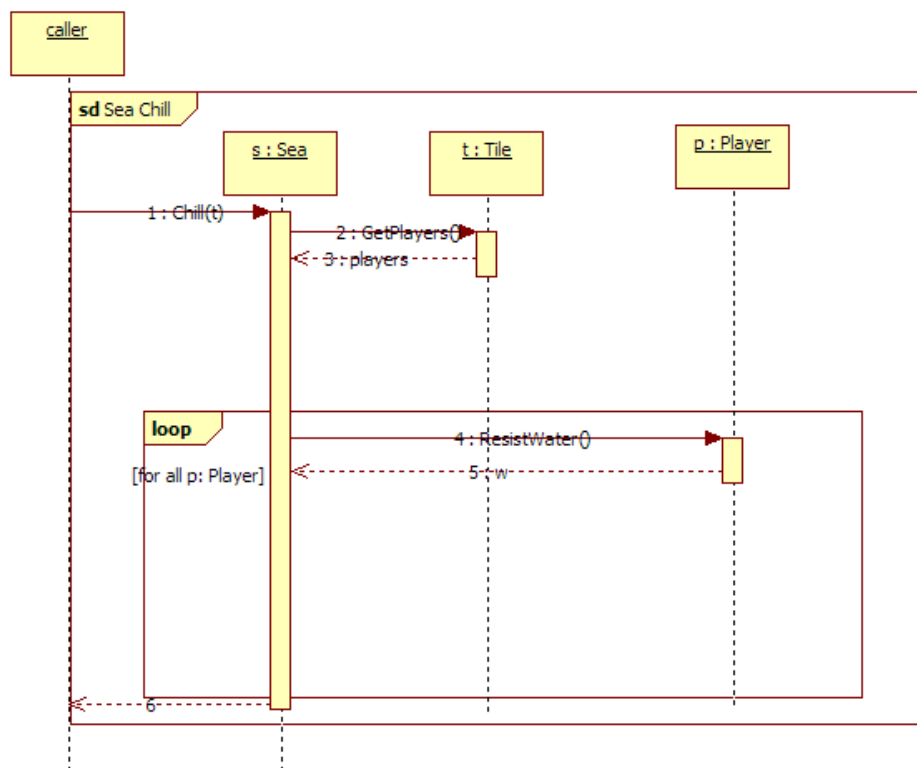
4.32. ábra. BareIce.Chill()



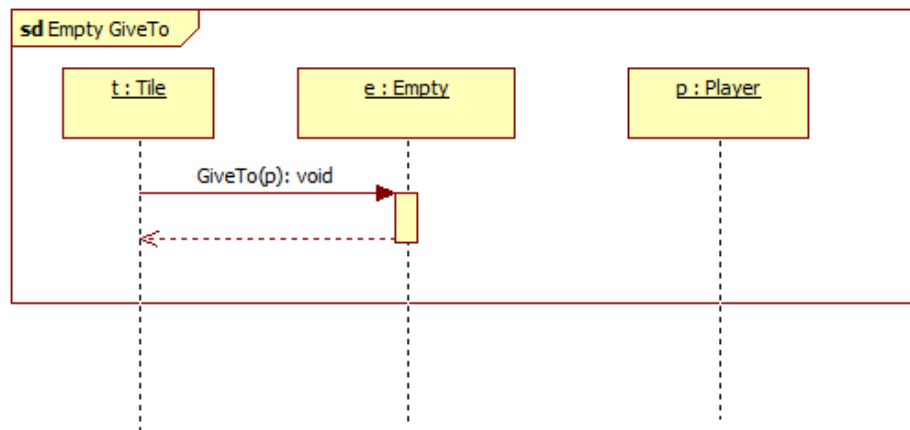
4.33. ábra. DryLand.Chill(Tile)



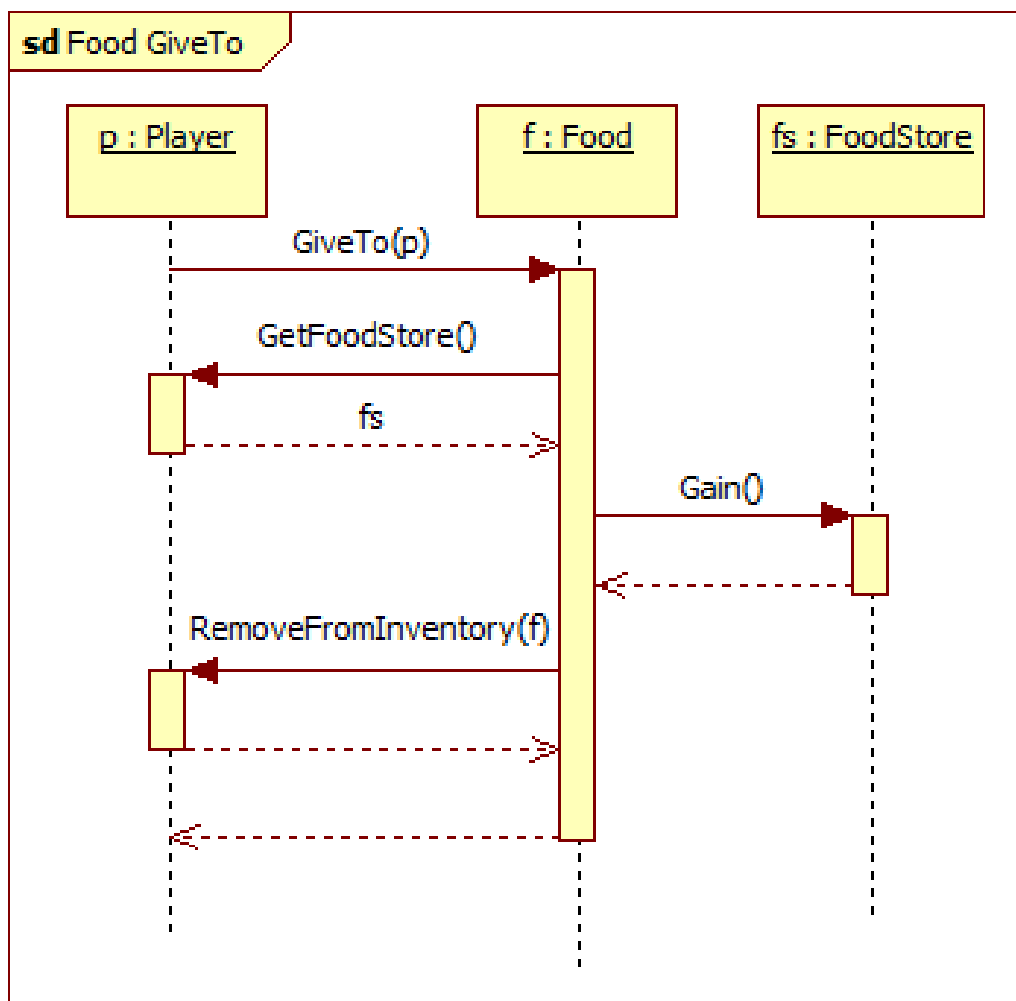
4.34. ábra. Igloo.Chill(Tile)



4.35. ábra. Sea.Chill(Tile)

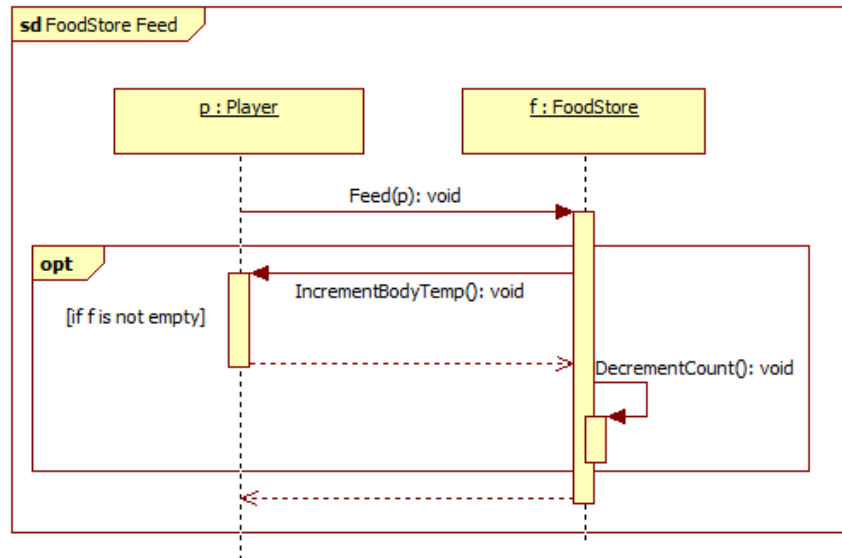


4.36. ábra. Empty.GiveTo(Player)

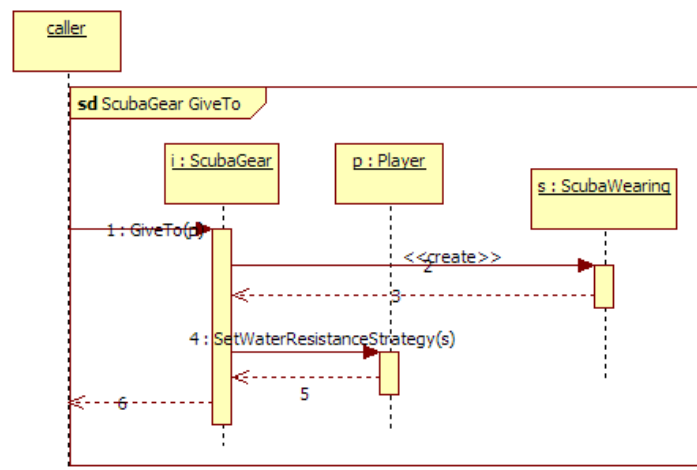


4.37. ábra. Food.GiveTo(Player)

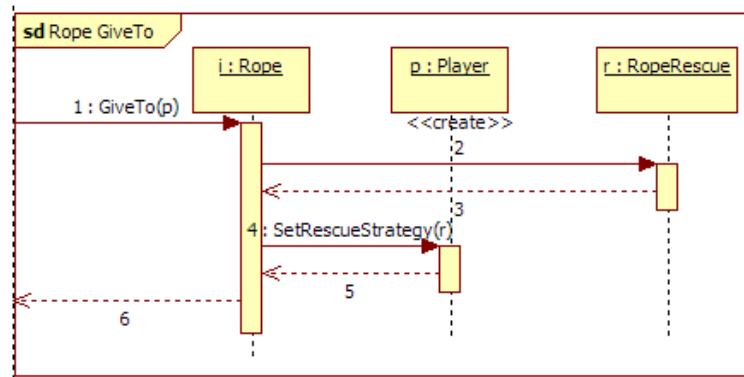




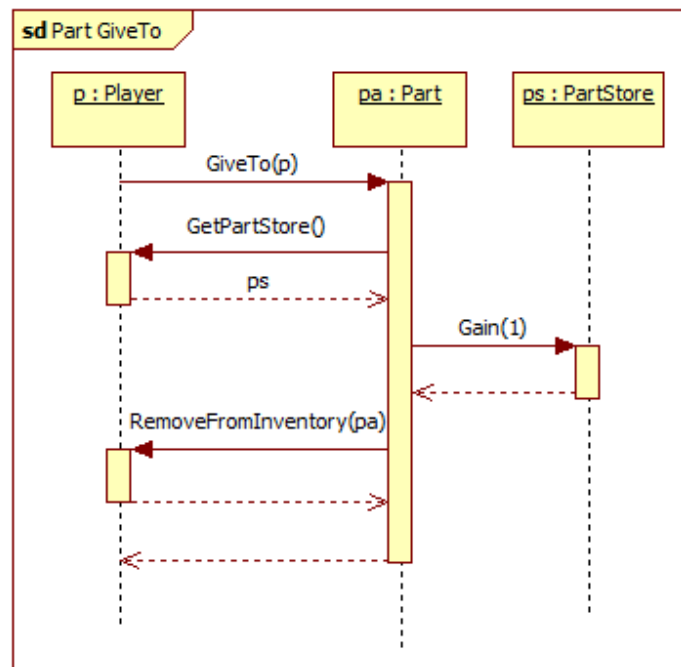
4.38. ábra. FoodStore.Feed(Player)



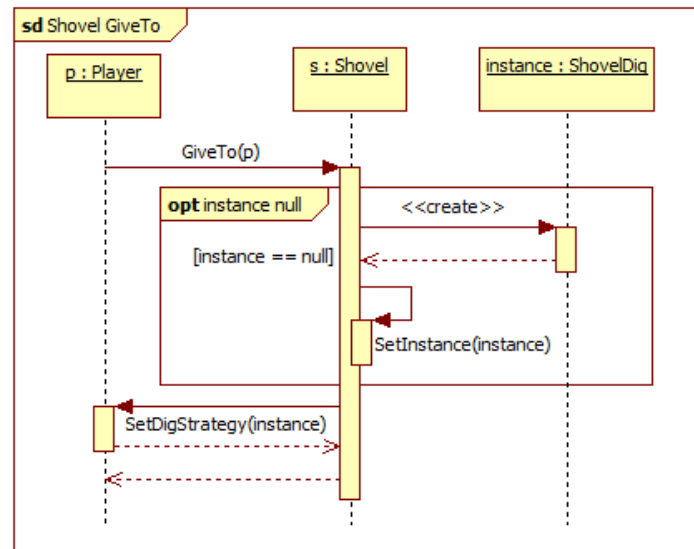
4.39. ábra. ScubaGear.GiveTo(Player)



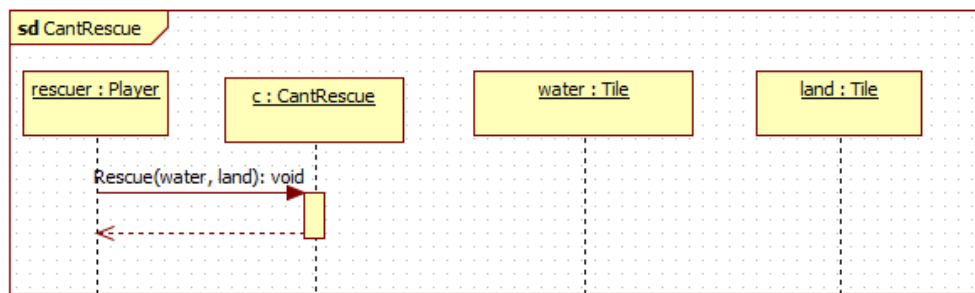
4.40. ábra. Rope.GiveTo(Player)



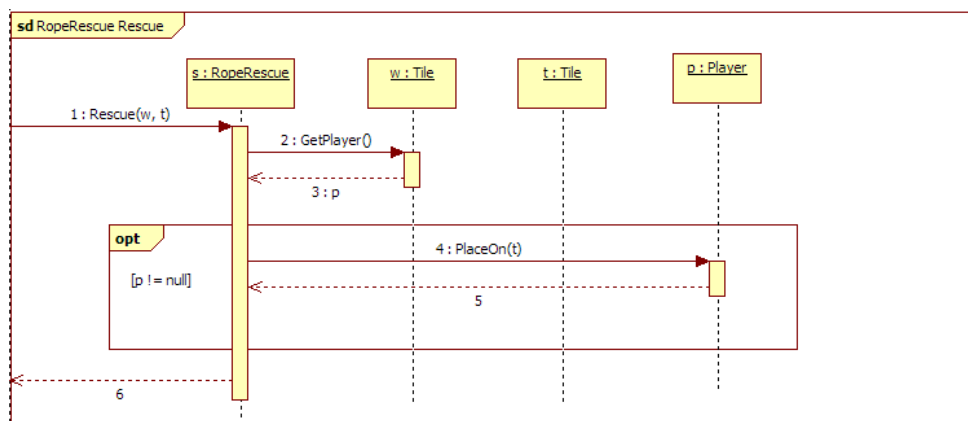
4.41. ábra. Part.GiveTo(Player)



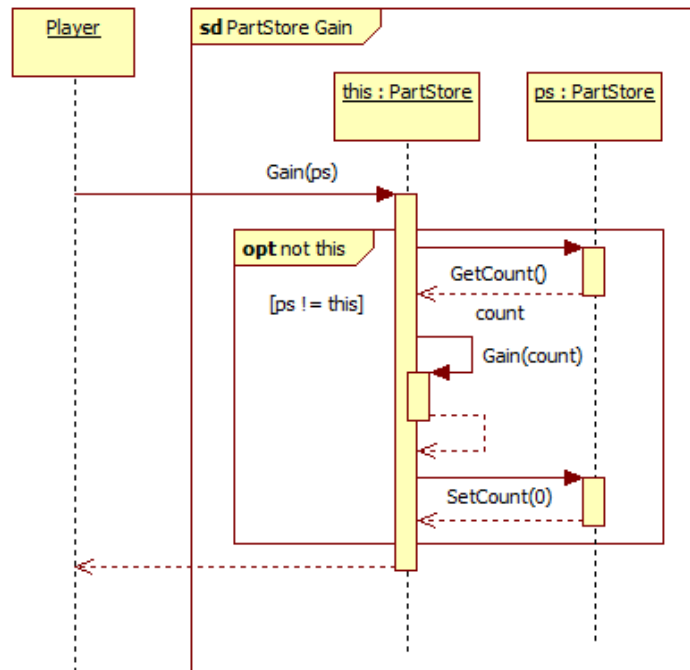
4.42. ábra. Shovel.GiveTo(Player)



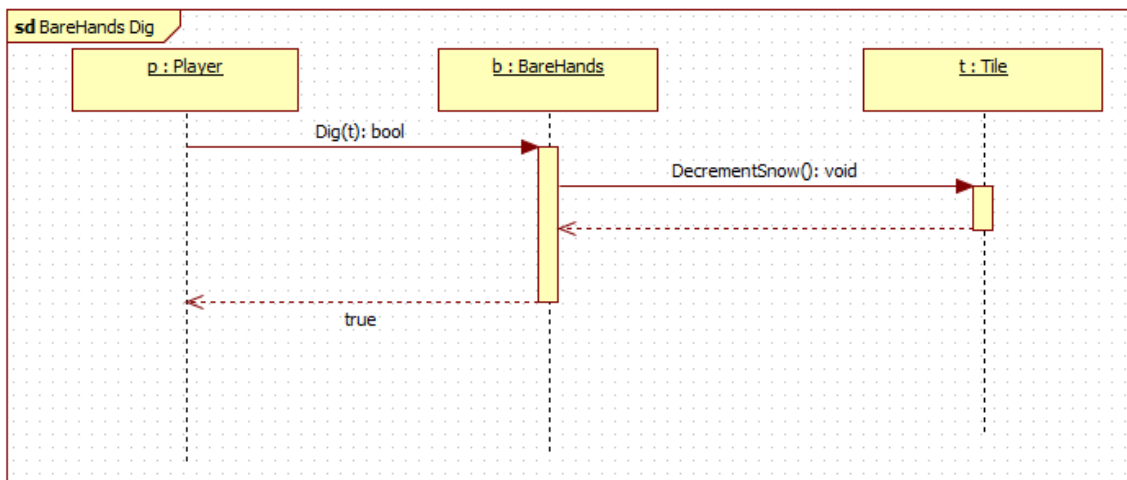
4.43. ábra. CantRescue.Rescue(Tile, Tile)



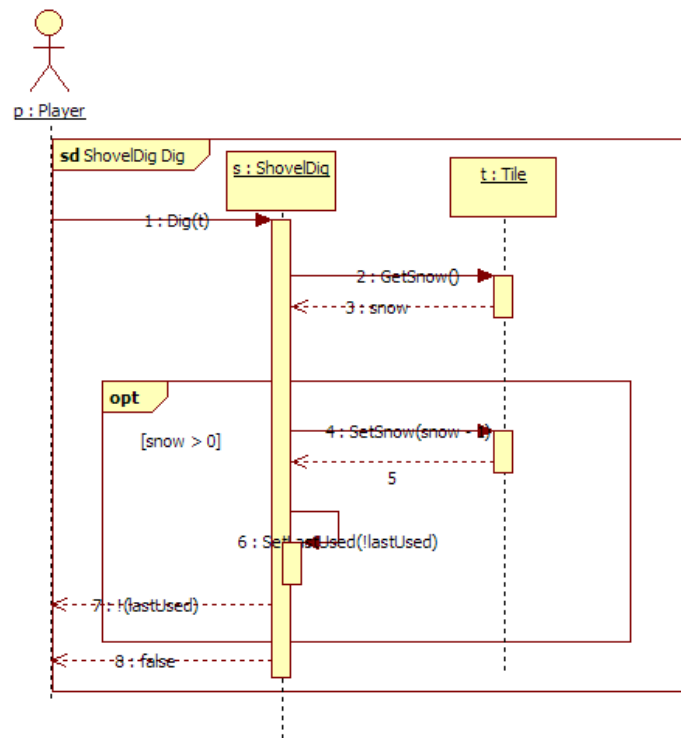
4.44. ábra. RopeRescue.Rescue(Tile, Tile)



4.45. ábra. PartStore.Gain(PartStore)



4.46. ábra. BareHandsDig.Dig(Tile)



4.47. ábra. ShovelDig.Dig(Tile)

**4.5. Napló**

<b>Kezdet</b>	<b>Időtartam</b>	<b>Résztevők</b>	<b>Leírás</b>
2020.03.05. 14:00	1 óra	<b>Kiss</b>	Ötletelés
2020.03.07. 15:00	1 óra	<b>Glávits</b>	Ötletelés
2020.03.07. 18:00	1 óra	<b>Kiss</b>	Glávits ötleteinek pontosítása
2020.03.07. 19:00	1,5 óra	<b>Glávits</b>	Szekvenciák
2020.03.07. 21:00	0,5 óra	<b>Glávits</b>	Dokumentáció
2020.03.07. 23:00	1,5 óra	<b>Lant</b>	Class diagram update, javítás, Class diagram hibák keresése
2020.03.08. 14.30	1 óra	<b>Kiss</b>	Dokumentáció javítgatás