

3. Analízis modell kidolgozása 1

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 2.

Tartalomjegyzék

3	Analízis modell kidolgozása 1	4
3.1	Objektum katalógus	4
3.1.1	Játékos	4
3.1.2	Jégtábla	4
3.1.3	Kötél	4
3.1.4	Búváruha	4
3.1.5	Lapát	4
3.1.6	Élelem	4
3.1.7	Rakétapisztoly Alkatrész	4
3.1.8	Iglu	4
3.2	Statikus struktúra diagramok	5
3.3	Osztályok leírása	5
3.3.1	BareHands	5
3.3.2	BareIce	5
3.3.3	CantRescue	5
3.3.4	ChillStormStrategy	5
3.3.5	ChillWaterStrategy	5
3.3.6	DigStrategy	5
3.3.7	DryLand	6
3.3.8	Empty	6
3.3.9	Eskimo	6
3.3.10	Food	6
3.3.11	FoodStore	6
3.3.12	Game	7
3.3.13	Igloo	7
3.3.14	Naked	7
3.3.15	Part	7
3.3.16	PartStore	8
3.3.17	Player	8
3.3.18	PolarExplorer	9
3.3.19	RescueStrategy	9
3.3.20	Rope	9
3.3.21	RopeRescue	9
3.3.22	ScubaGear	9
3.3.23	Sea	10
3.3.24	ShovelDig	10
3.3.25	Tile	10
3.3.26	WaterResistanceStrategy	11
3.4	Statikus struktúra diagramok	12
3.5	Szekvencia diagramok	14
3.6	Napló	40

Ábrák jegyzéke

. Osztálydiagram 1.	12
. Osztálydiagram 2.	13
. Game.InitPlayer()	14
. Game.Turn()	15
. Game.CreateIce()	16
. Game.CreateUnstableIce()	17
. Game.CreateHole()	18
. Game.CreateSea()	19
. Game.CreatePolarExplorer()	20
. Game.CreateEskimo()	20
. Game.GenerateItem()	21
. Player.Step(direction: int)	21
. Player.Dig()	22
. Player.PickUp()	22
. Player.PlaceOn(Tile)	23
. Player.EatFood()	23
. Player.RescueTeammate(direction: int)	24
. Player.Chill()	24
. Player.ResistWater()	25
. Player.AssembleFlare()	25
. Player.AssembleFlare()	26
. Eskimo.BuildIgloo()	26
. PolarExplorer.Examine(direction: int)	27
. Tile.StepOn(Player)	28
. Tile.StepOff(Player)	29
. Tile.GiveItem(Player)	29
. Tile.ChillWater()	30
. Tile.ChillStorm()	30
. Naked.Chill(Player)	31
. ScubaWearing.Chill(Player)	31
. BareIce.Chill()	32
. DryLand.Chill(Tile)	32
. Igloo.Chill(Tile)	33
. Sea.Chill(Tile)	33
. Empty.GiveTo(Player)	34
. Food.GiveTo(Player)	34
. FoodStore.Feed(Player)	35
. ScubaGear.GiveTo(Player)	35
. Rope.GiveTo(Player)	36
. Part.GiveTo(Player)	36
. Shovel.GiveTo(Player)	37
. CantRescue.Rescue(Tile, Tile)	37
. RopeRescue.Rescue(Tile, Tile)	37
. PartStore.Gain(PartStore)	38
. BareHandsDig.Dig(Tile)	38
. ShovelDig.Dig(Tile)	39

3. Analízis modell kidolgozása 1

3.1. Objektum katalógus

3.1.1. 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.

3.1.2. 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

3.1.3. Kötél

Ennek segítségével ki lehet húzni egy vízbe esett játékost.

3.1.4. Búvárruha

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

3.1.5. Lapát

Segítségével 2 egységnyi hó takarítható el, egy egység munkával.

3.1.6. Élelem

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

3.1.7. Rakétapisztoly Alkatrész

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

3.1.8. Iglu

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

3.2. Statikus struktúra diagramok

3.3. Osztályok leírása

3.3.1. BareHands

- A játékos így ás, ha nincs ásója.
- Interfészek:
 - DigStrategy
- Metódusok:
 - bool Dig(Tile t): Csökkenti a tile-on található hó mennyiségét (int)

3.3.2. BareIce

- A jég táblán nincs védelem a vihar elől.
- Interfészek:
 - ChillStormStrategy
- Metódusok:
 - void Chill(Tile t): Táblán álló játékosok testhője csökken.

3.3.3. CantRescue

- A játékos nem tudja kihúzni a csapattársát.
- Interfészek:
 - RescueStrategy
- Metódusok:
 - void Rescue(Tile water, Tile land): üres

3.3.4. ChillStormStrategy

- A jég tábla így húti viharban a játékosokat.
- Metódusok:
 - abstract void Chill(Tile t)

3.3.5. ChillWaterStrategy

- A jég tábla így húti a vízbe esett játékosokat.
- Metódusok:
 - abstract void Chill(Tile t)

3.3.6. DigStrategy

- A játékos így ás.
- Metódusok:
 - abstract bool Dig(Tile t)

3.3.7. DryLand

- A szárazföld nem hűti a játékosokat.
- Interfészek:
 - ChillWaterStrategy
- Metódusok:
 - void Chill(Tile t): üres

3.3.8. Empty

- Nincs jégbe fagyott tárgy.
- Interfészek:
 - GiveItemStrategy
- Metódusok
 - void GiveTo(Player p): üres

3.3.9. Eskimo

- Játékos osztály.
- Ősosztályok:
 - Player
- Metódusok:
 - void BuildIgloo(): Épít egy iglut a mezőre, amin áll.

3.3.10. Food

- Élelem, amit a játékos meg tud enni, hogy növelje a testhőjét.
- Interfészek:
 - GiveItemStrategy
- Metódusok:
 - void GiveTo(Player p): A játékos kap egy élelmet.

3.3.11. 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.

3.3.12. Game

- Interface a Model és a Controller között. A játékmesterhez tartozó működést valósítja meg.
- Attribútumok:
 - players: Player[3..*]: Tárolja a játékosokat
 - icefield: Tile[1..*]: Tárolja a pályát alkotó elemeket
- Metódusok:
 - 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.
 - void Victory(): Ha vége a játéknak, szól a Controllernek, hogy nyertünk. Külső metódus.

3.3.13. Igloo

- Ezen a jégtáblán iglu áll, a játékosok védve vannak a vihartól.
- Interfészek:
 - ChillStromStrategy
- Metódusok:
 - void Chill(Tile t): üres

3.3.14. Naked

- A játékos védtelen a hideg vízzel szemben.
- Interfészek:
 - WaterResistanceStrategy
- Metódusok:
 - void Chill(Player p): Játékosnak nincsen ereje a vízben úszni bűvárruha nélkül.

3.3.15. Part

- Jégbefagyott alkatrész.
- Interfészek:
 - GiveItemStrategy
- Metódusok:
 - void GiveTo(Player p): A játékos tárolójába kerül egy darab a rakétapisztolyból.

3.3.16. PartStore

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

3.3.17. Player

- Játékos osztály, amit a felhasználó irányít a grafikus felületen keresztül.
- Attribútumok:
 - bodyTemp: int: Jelzi a játékos jelenlegi hőmérsékletét, ha 0 akkor megfagy → játék vége.
 - currentTile: Tile: A játékos ismeri a mezőt amin éppen áll.
 - digStrategy: DigStrategy: Eldönti hogyan képes ásni a játékos.
 - energy: int: Számlálja mennyit mozogott az adott körben a játékos.
 - foodStore: FoodStore: Tárolja a játékos ételeit.
 - 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 vízbeesés 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.
 - void PickUp(): Ezt a metódust a Controller hívja. A játékos felvesz egy tárgyat. 1 munkaegység
 - void PlaceOn(Tile t): Init szekvencia része. RopeRescue szekvencia része. Rárak egy játékost egy másik Tile-ra.
 - void RescueTeammate(direction d): 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(): Ezt a metódust a Controller hívja. A játékos lép, ha van még hozzá elég energiája. 1 munkaegység
 - void ToFoodStore(): Élelem megtalálásához helper metódus.

3.3.18. PolarExplorer

- Játékos típus, akivel valaki játszhat
- Ősosztályok:
 - Player
- Metódusok:
 - int Examine(direction d): A játékos megnézheti, hogy egy adott Tile-nak mennyi a teherbírása.

3.3.19. RescueStrategy

- A játékos így húzza ki csapattársát a vízből.
- Metódusok:
 - abstract void Rescue(Tile water, Tile land): üres

3.3.20. Rope

- Jégbe fagyott kötél.
- Interfészek:
 - GiveItemStrategy
- Metódusok
 - void GiveTo(Player p): Felruházza a játékost kötéllal.

3.3.21. RopeRescue

- A játékos kihúzza csapattársát a vízből.
- Interfészek:
 - RescueStrategy
- Metódusok:
 - void Rescue(Tile water, Tile land): A vízben lévők közül egyvalaki rákerül a kihúzó játékos cellájára.

3.3.22. ScubaGear

- Jégbe fagyott búvárruha.
- Interfészek:
 - GiveItemStrategy
- Metódusok:
 - void GiveTo(): Felruházza a játékost búvárruhával.

3.3.23. Sea

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

3.3.24. ShovelDig

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

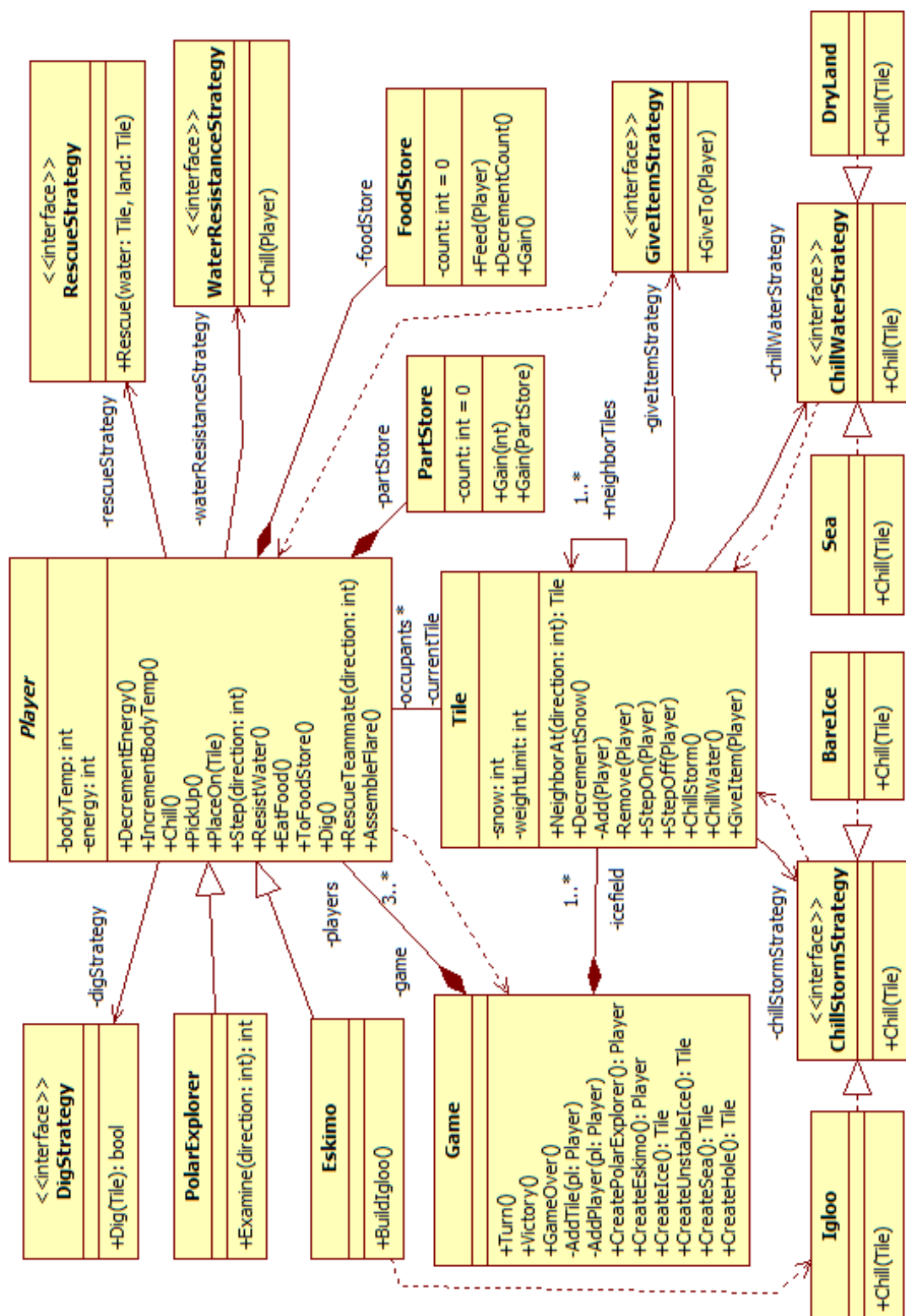
3.3.25. Tile

- 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.
 - giveItemStrategy: GiveItemStrategy: Eldönti, milyen tárgyat vesz fel a találó.
 - 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.
 - void GiveItem(Player): 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.
 - StepOff(Player): Járékos lelép a celláról.

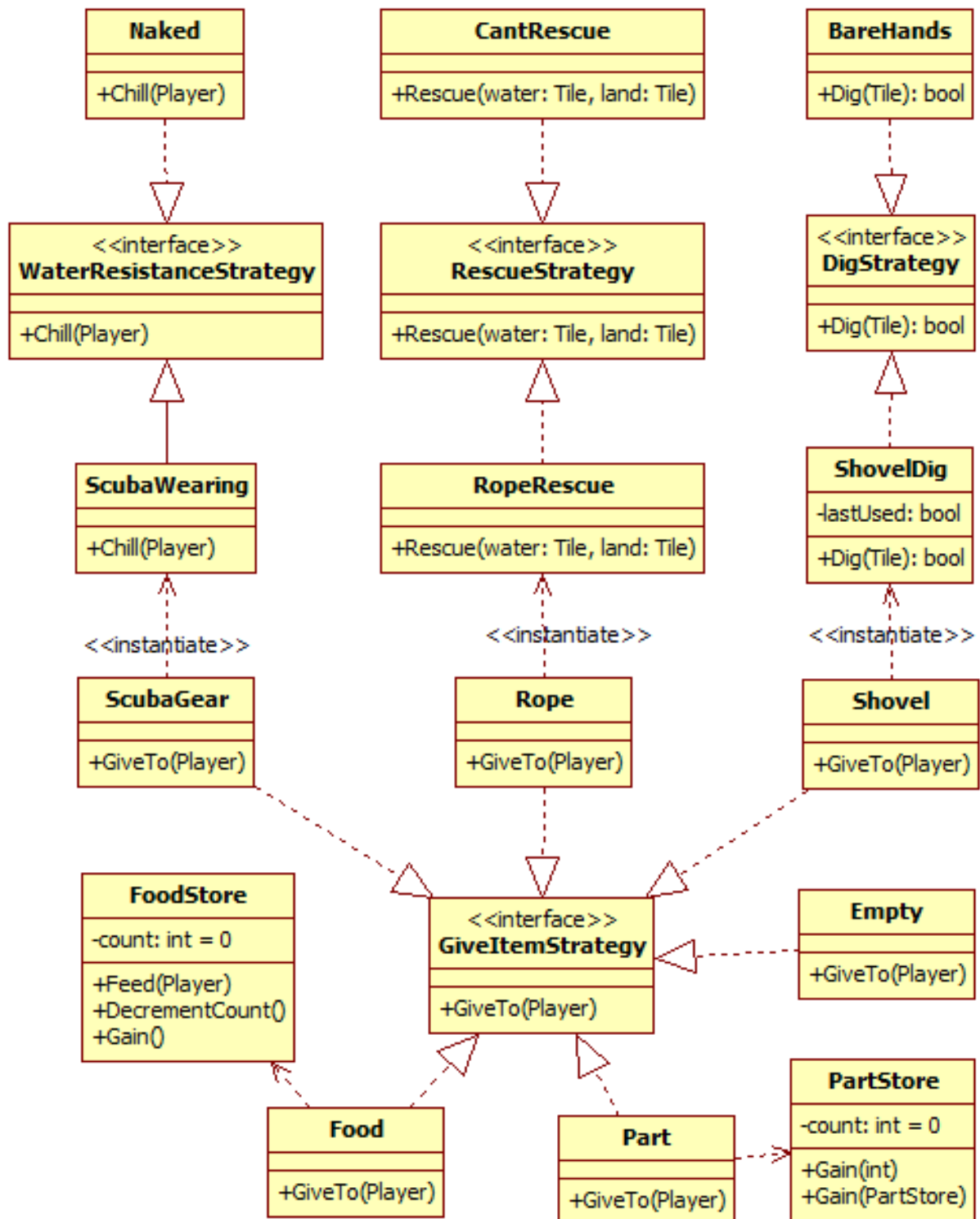
3.3.26. WaterResistanceStrategy

- Így reagál a játékos a hideg vízre.
- Metódusok:
 - `abstract void Chill(Player p):` üres

3.4. Statikus struktúra diagramok

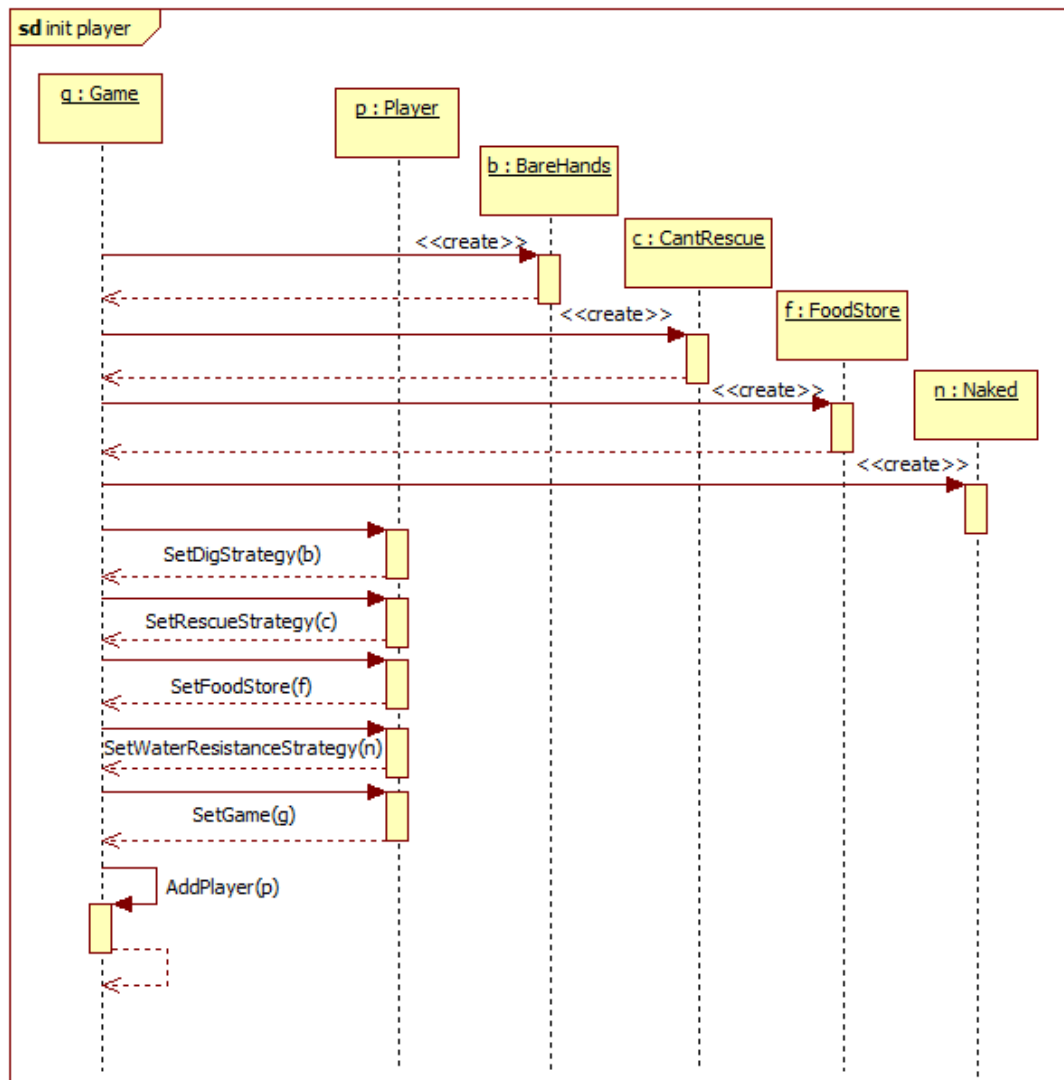


3.1. ábra. Osztálydiagram 1.

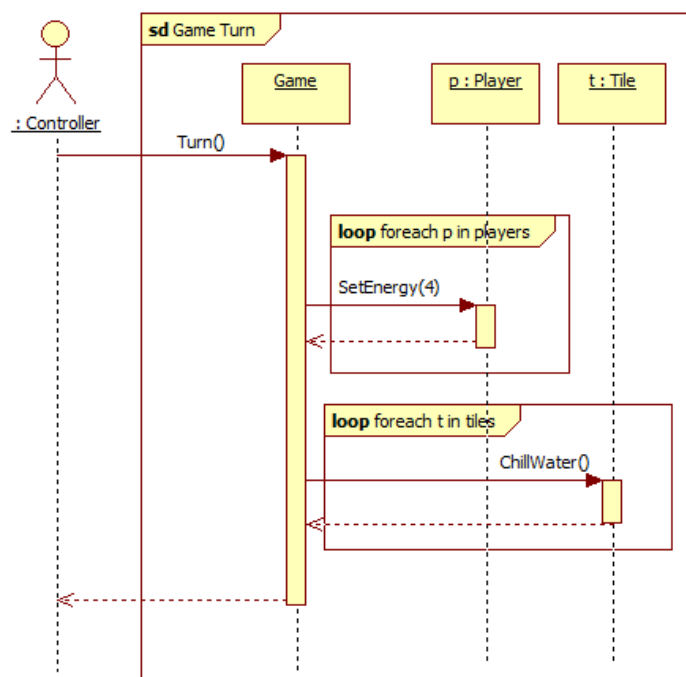


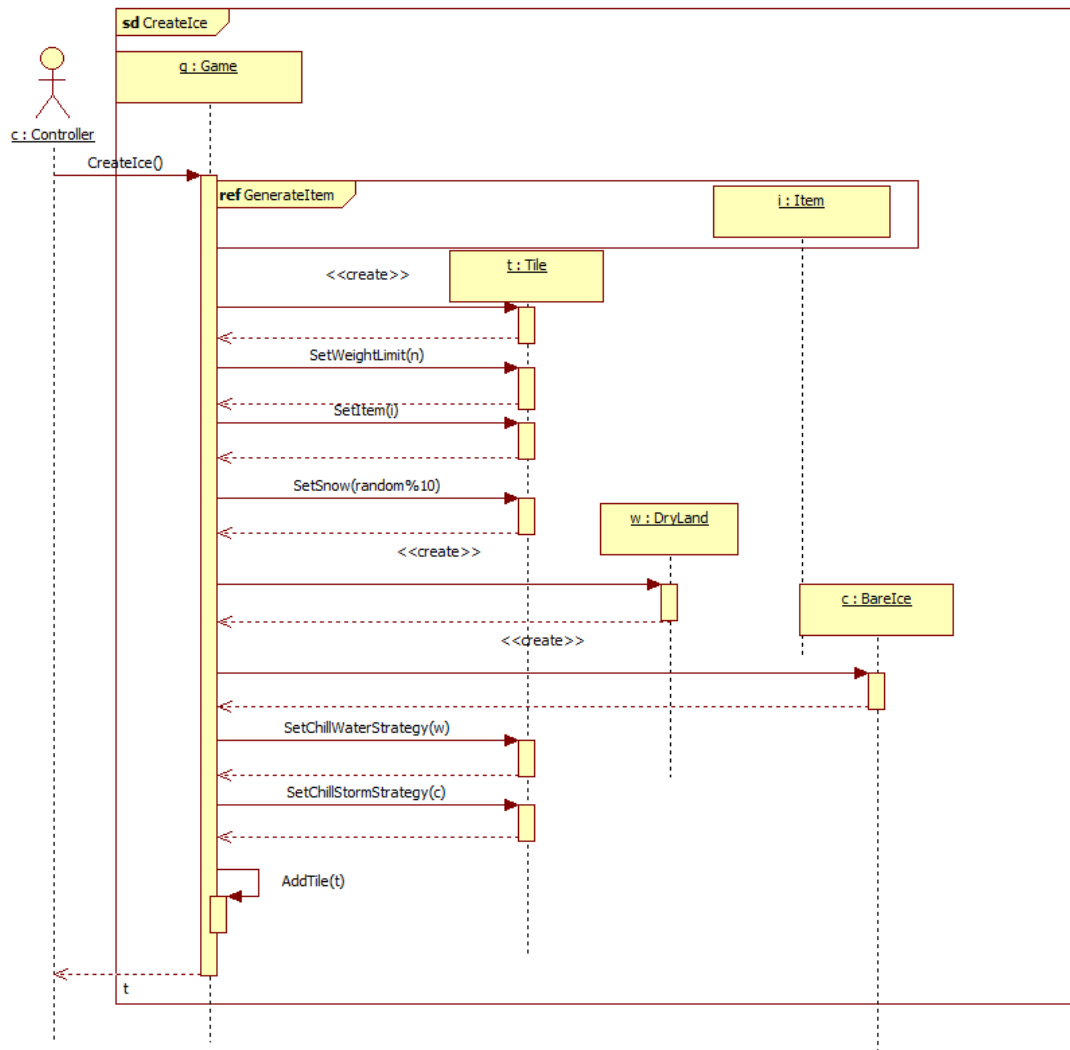
3.2. ábra. Osztálydiagram 2.

3.5. Szekvencia diagramok

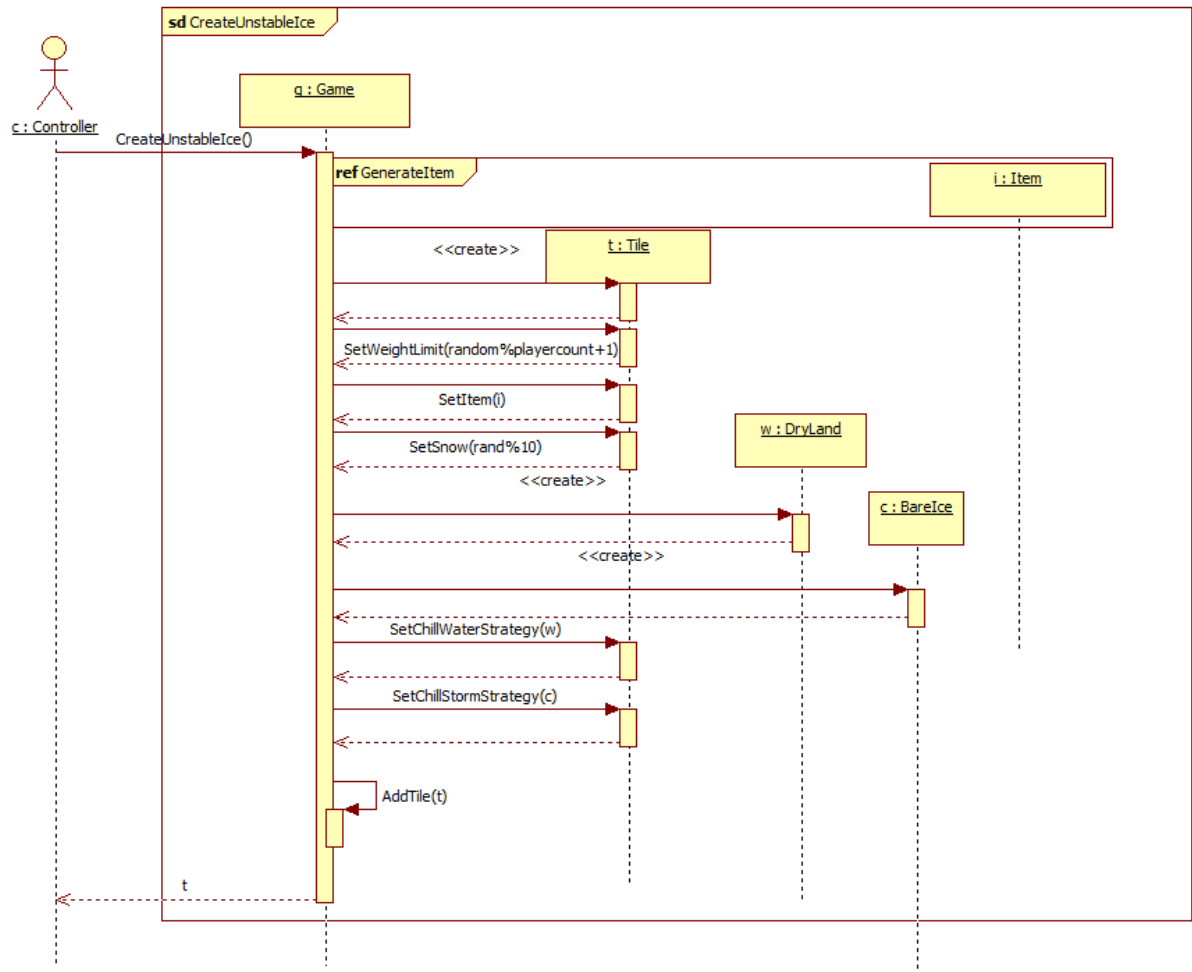


3.3. ábra. Game.InitPlayer()

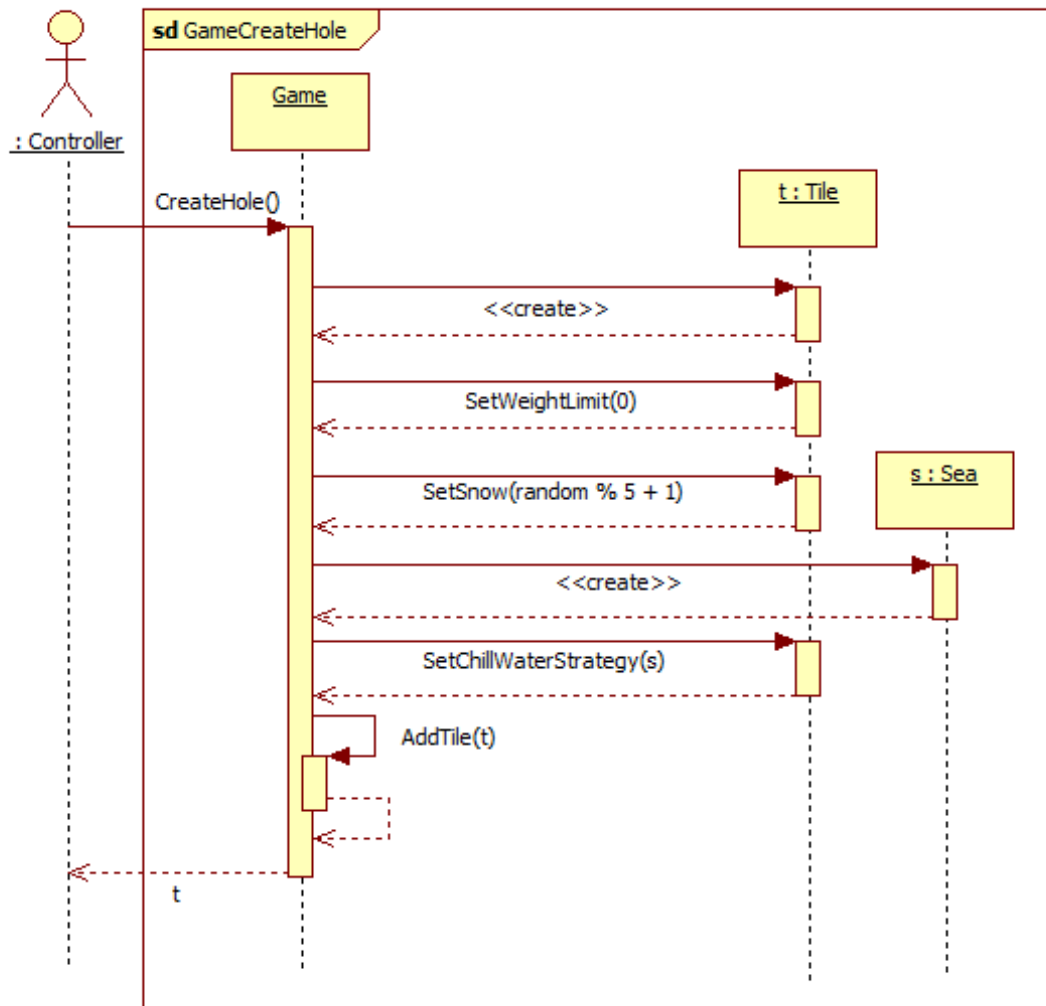
3.4. ábra. `Game.Turn()`



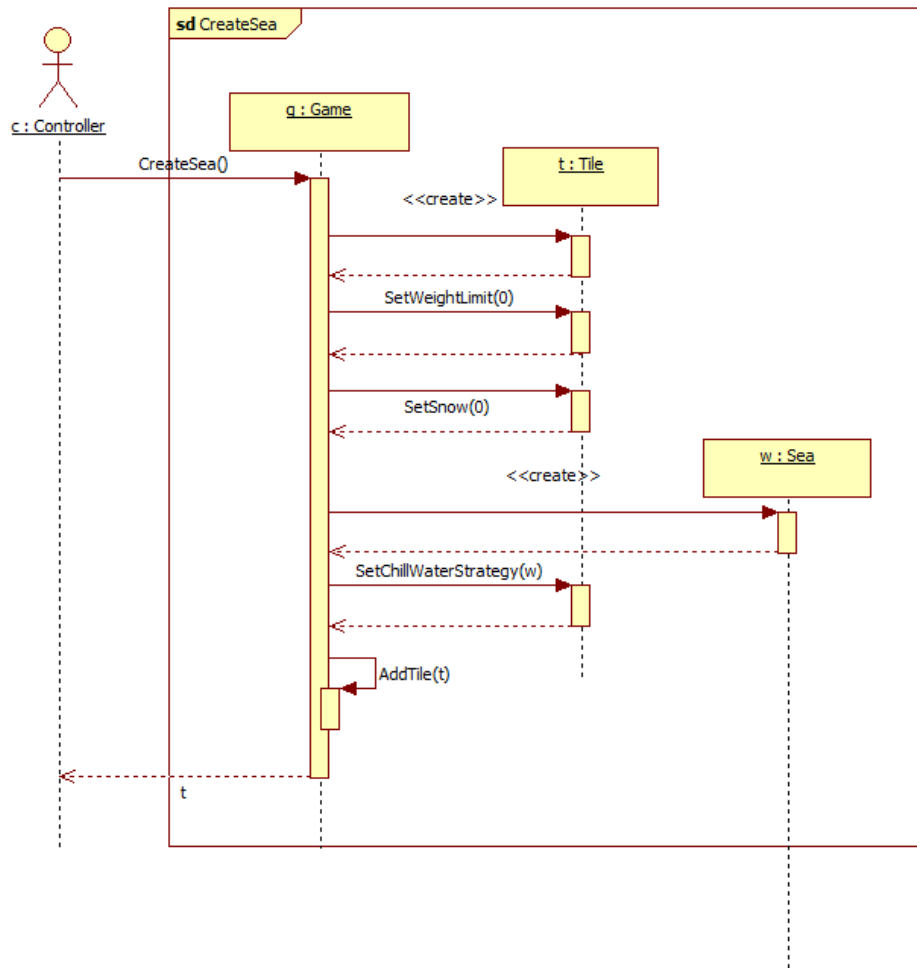
3.5. ábra. Game.CreateIce()

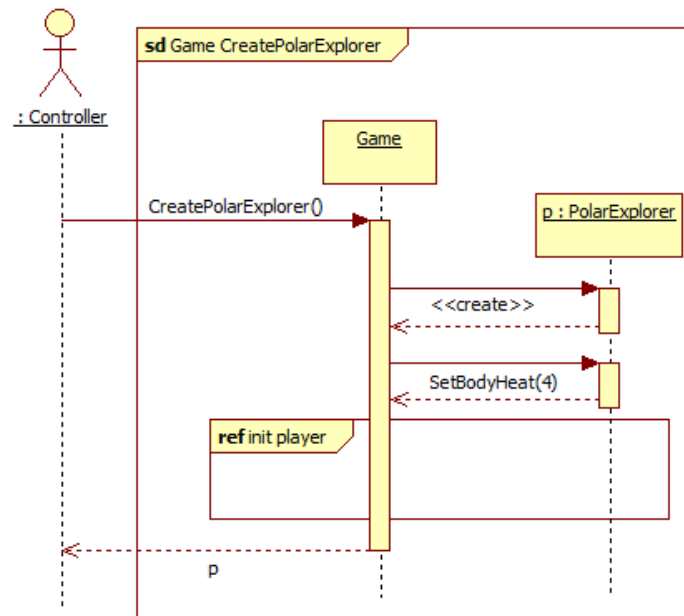


3.6. ábra. Game.CreateUnstableIce()

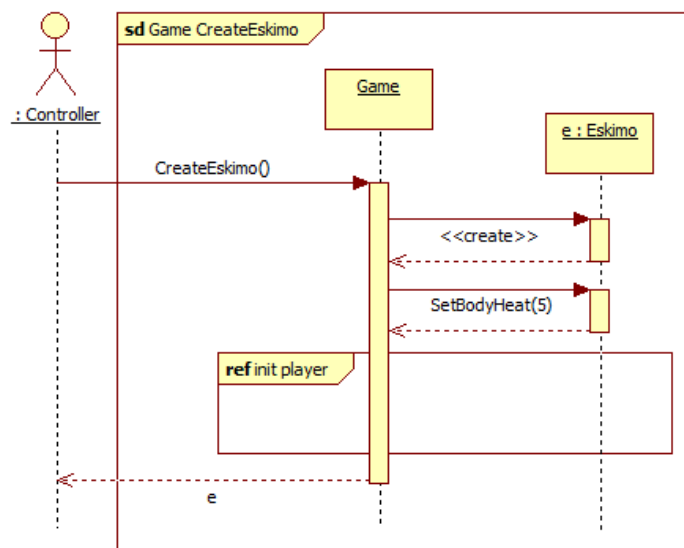


3.7. ábra. Game.CreateHole()

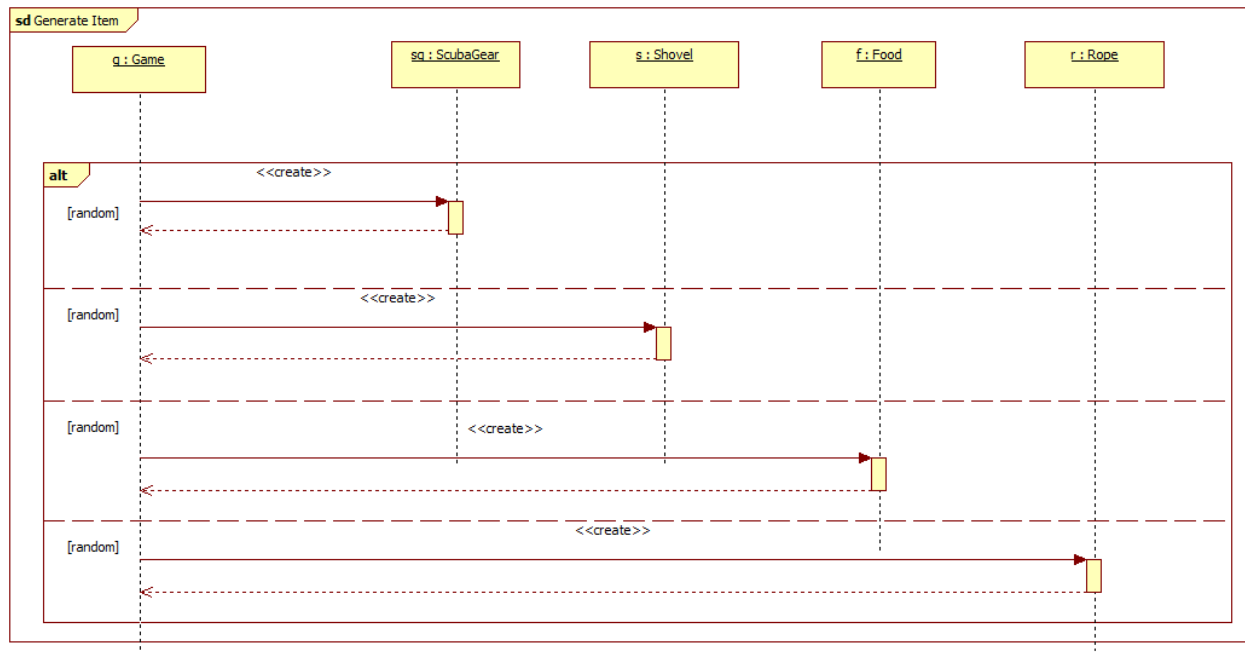
3.8. ábra. `Game.CreateSea()`



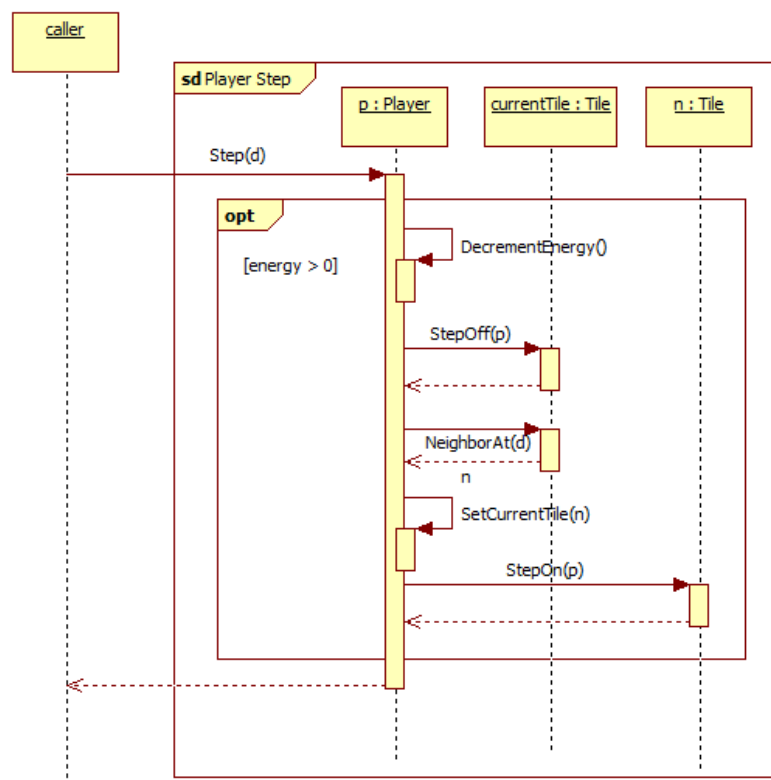
3.9. ábra. Game.CreatePolarExplorer()



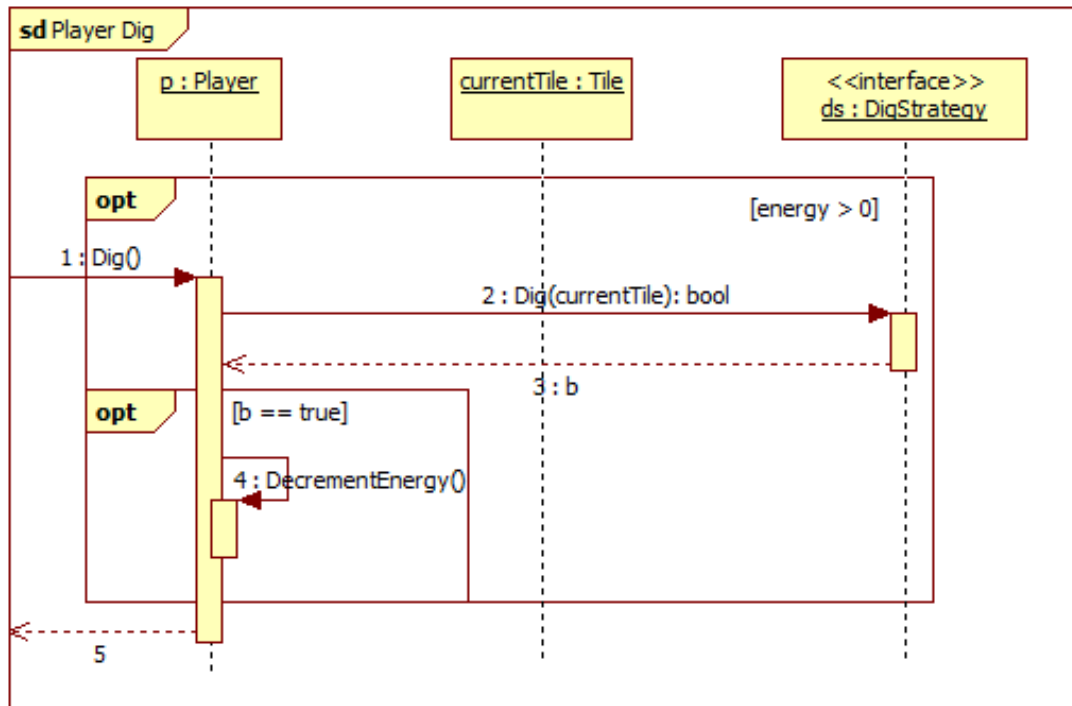
3.10. ábra. Game.CreateEskimo()



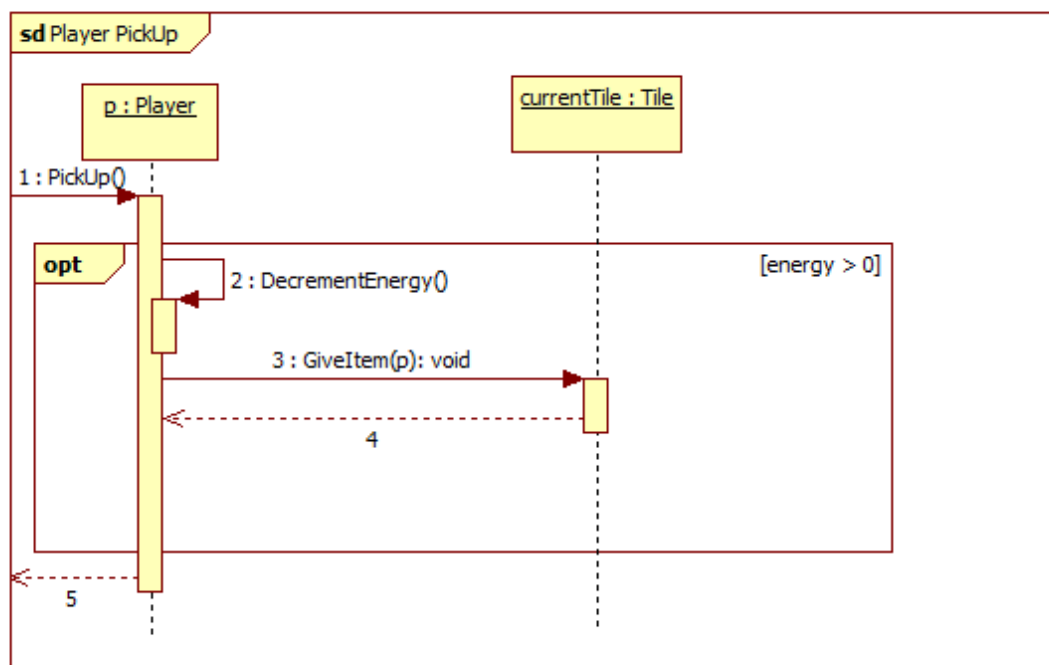
3.11. ábra. Game.GenerateItem()



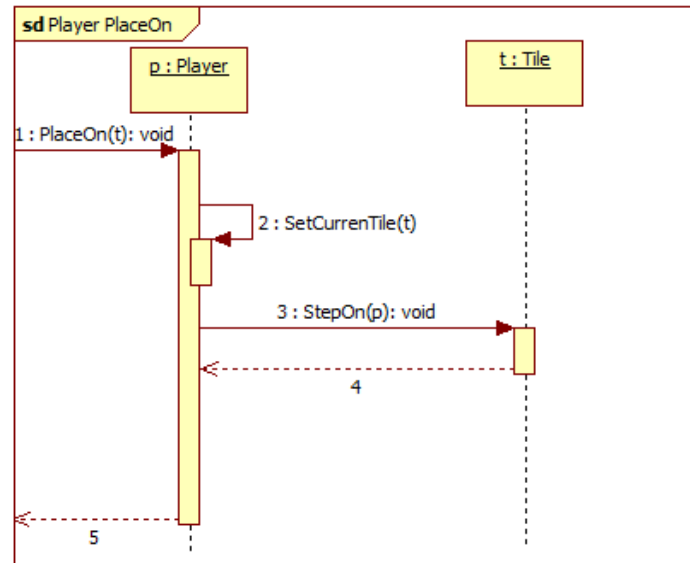
3.12. ábra. Player.Step(direction: int)



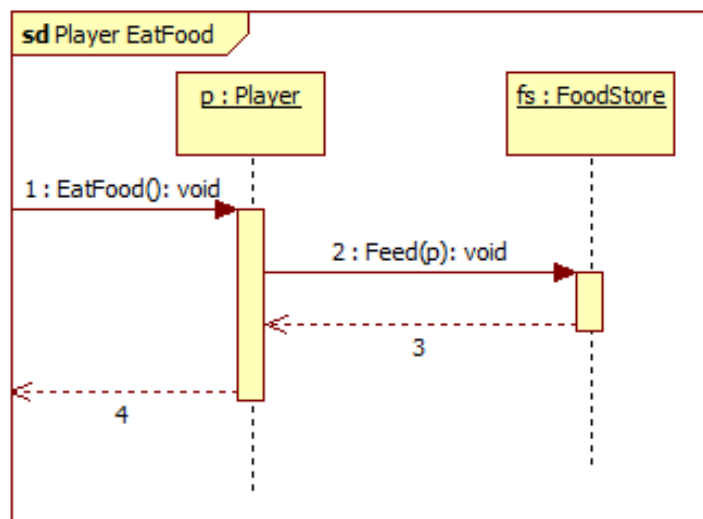
3.13. ábra. Player.Dig()



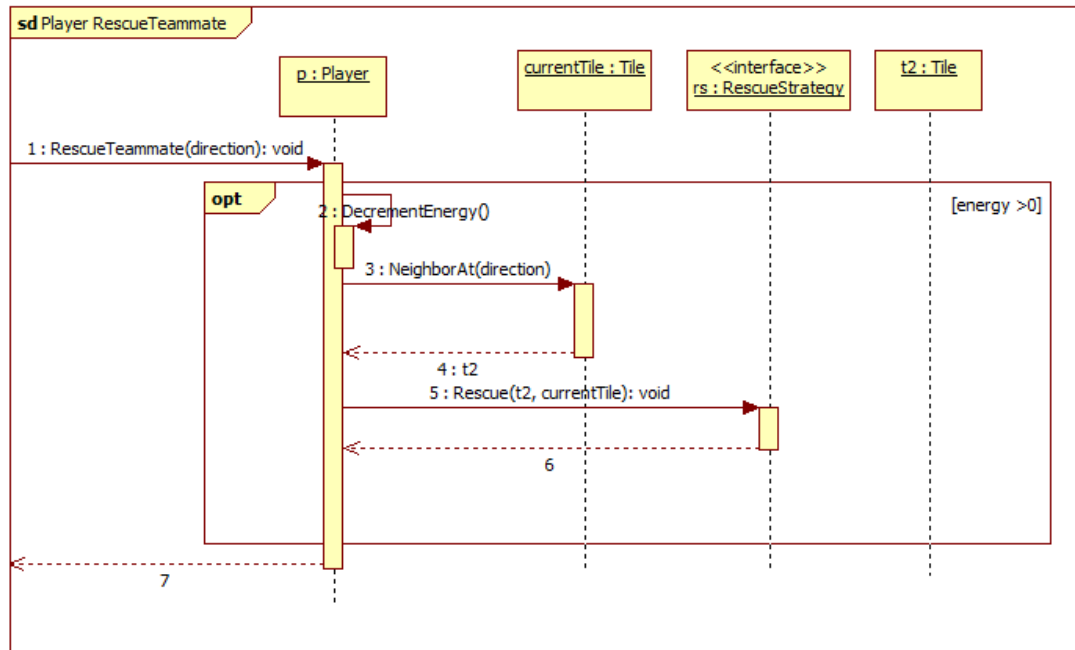
3.14. ábra. Player.PickUp()



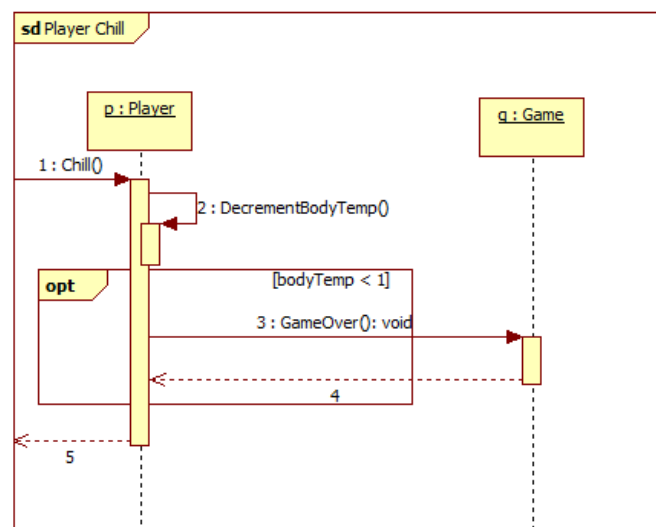
3.15. ábra. Player.PlaceOn(Tile)



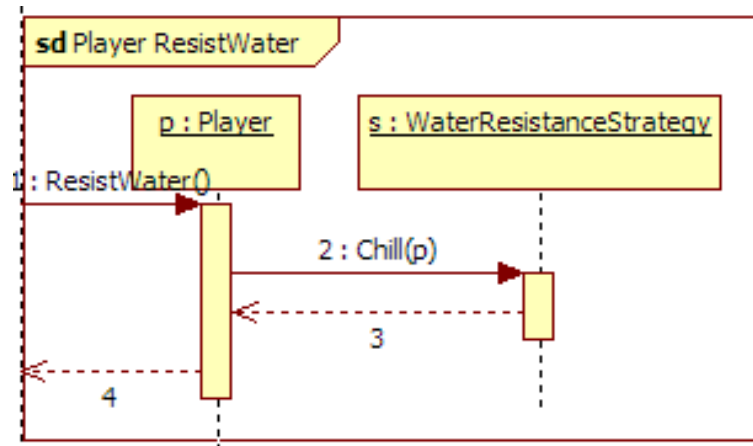
3.16. ábra. Player.EatFood()



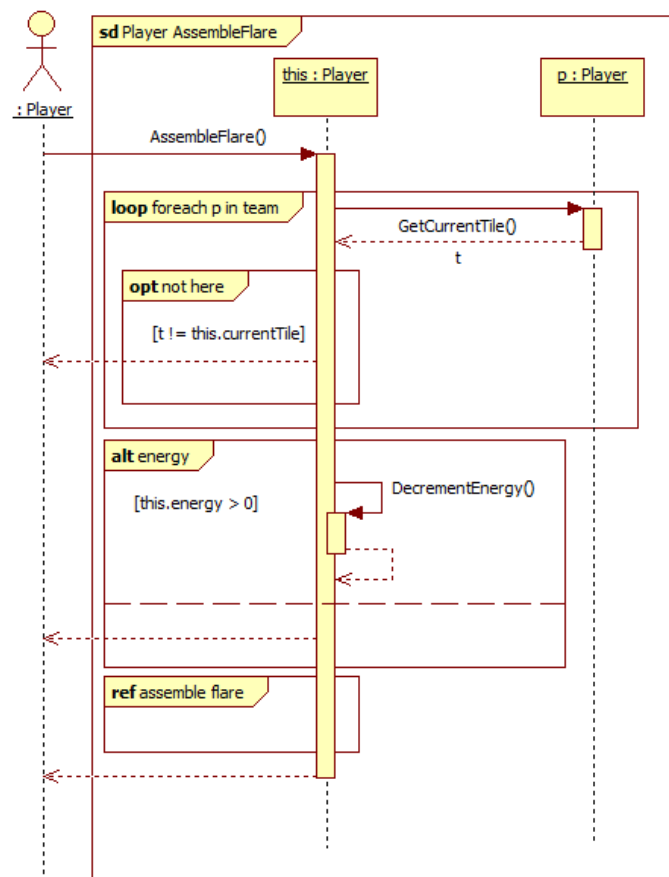
3.17. ábra. Player.RescueTeammate(direction: int)



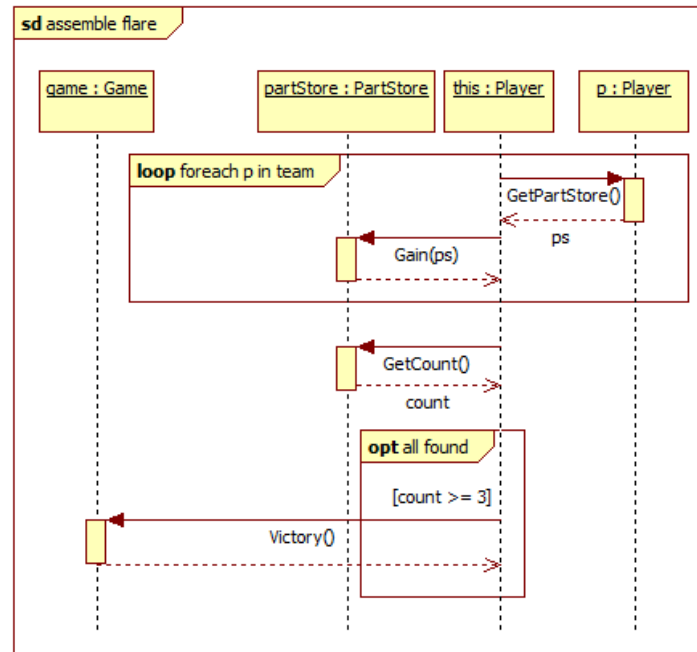
3.18. ábra. Player.Chill()



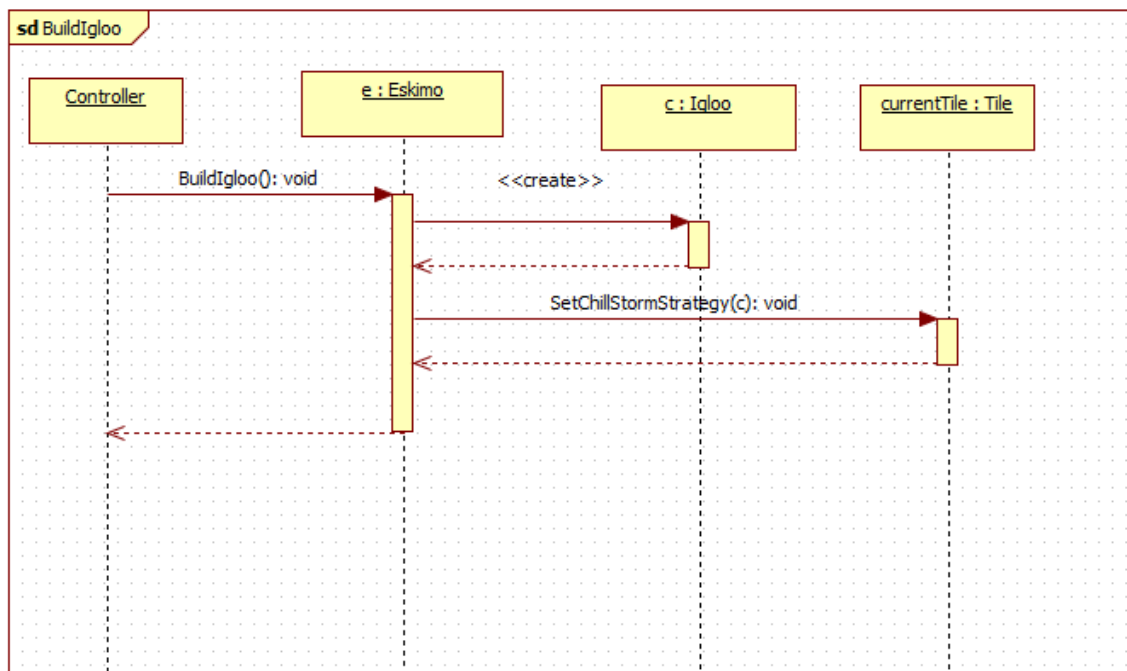
3.19. ábra. Player.ResistWater()



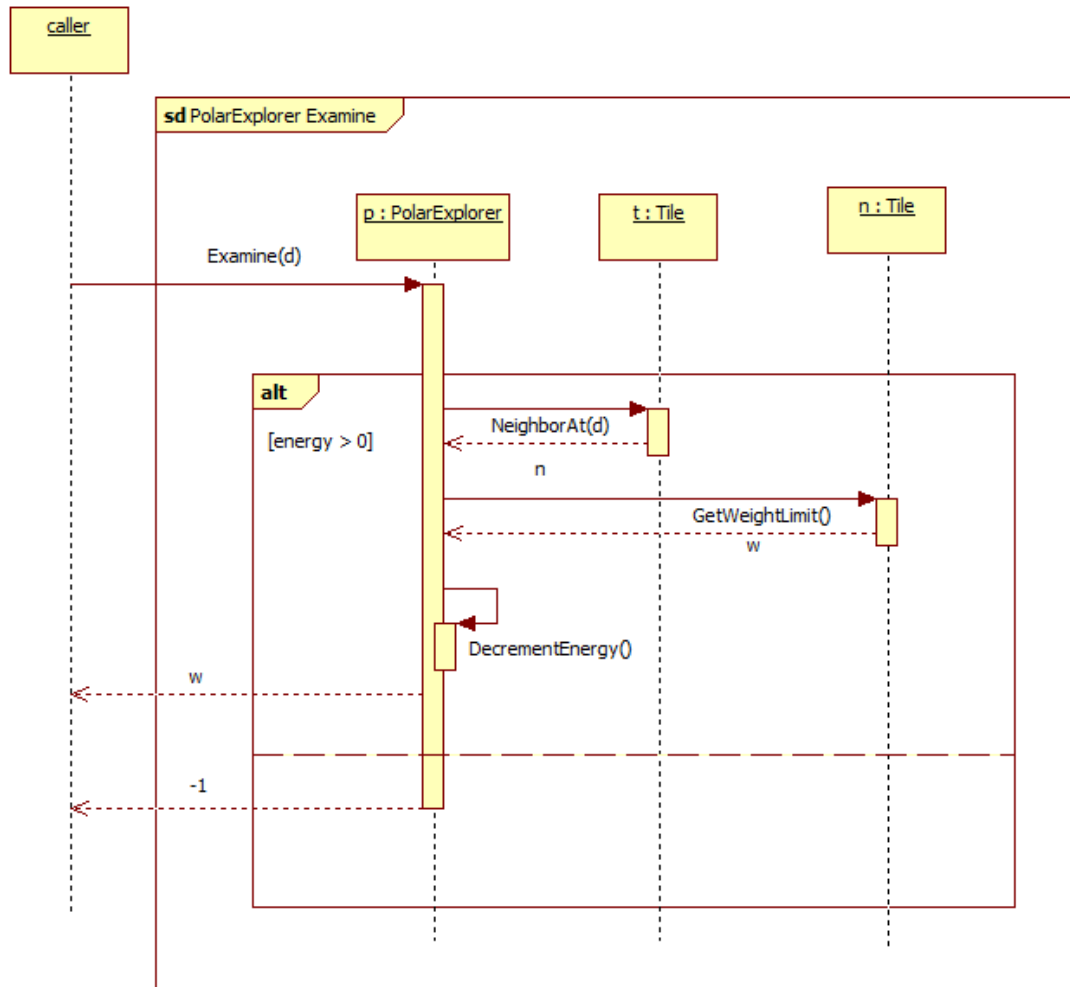
3.20. ábra. Player.AssembleFlare()

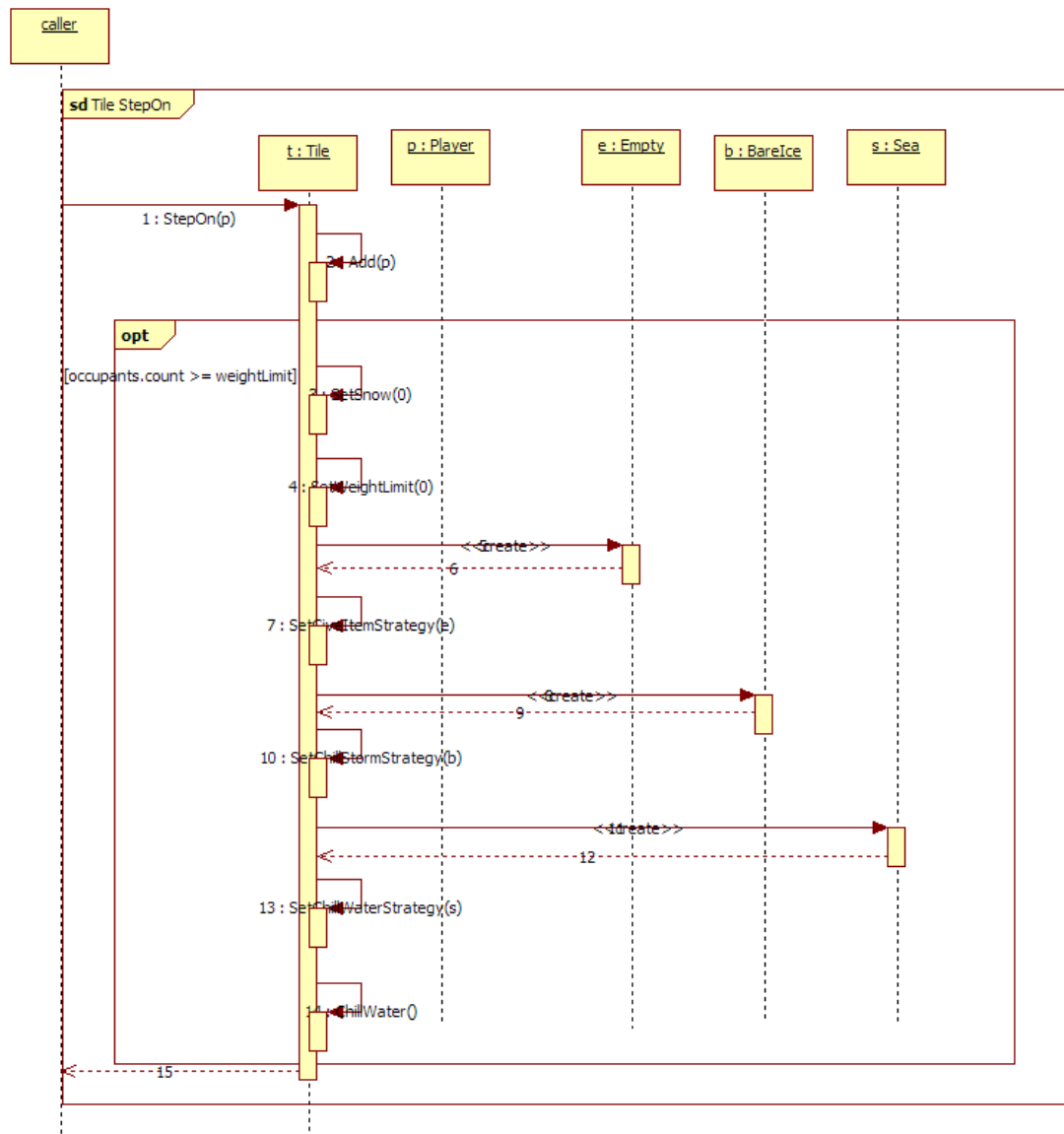


3.21. ábra. Player.AssembleFlare()

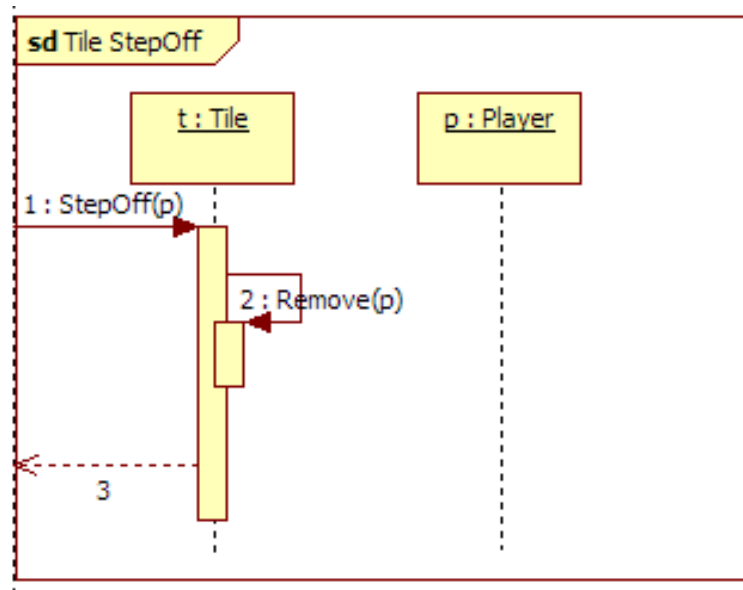
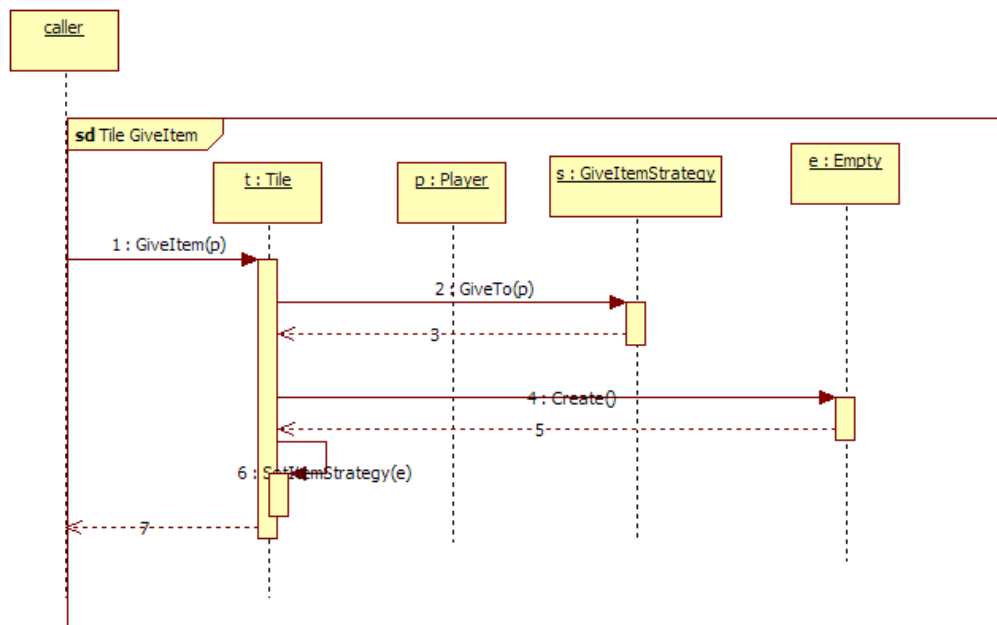


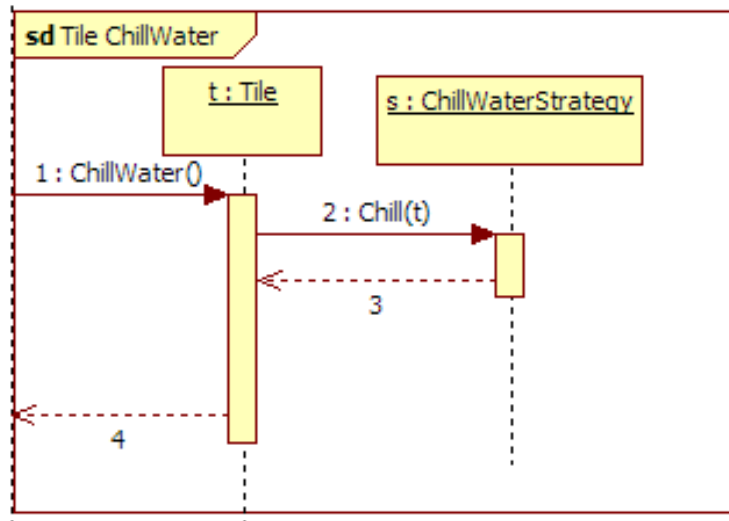
3.22. ábra. Eskimo.BuildIgloo()

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

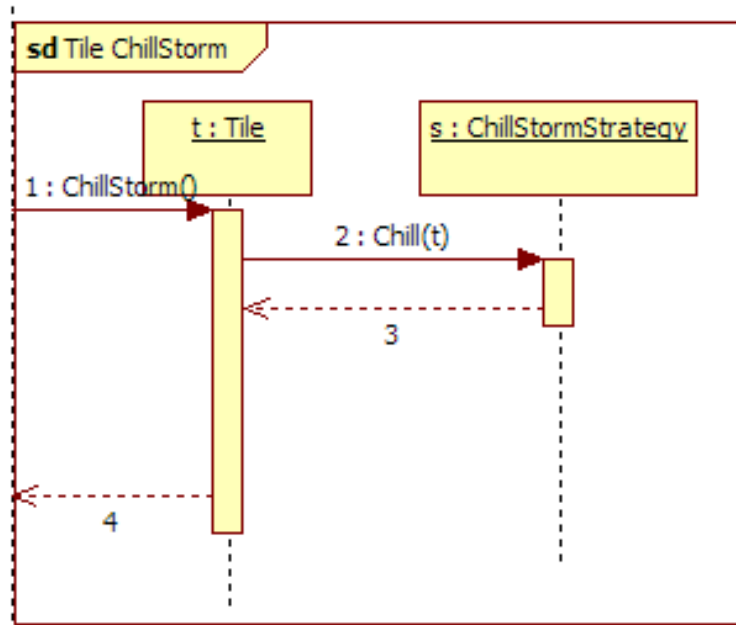


3.24. ábra. Tile.StepOn(Player)

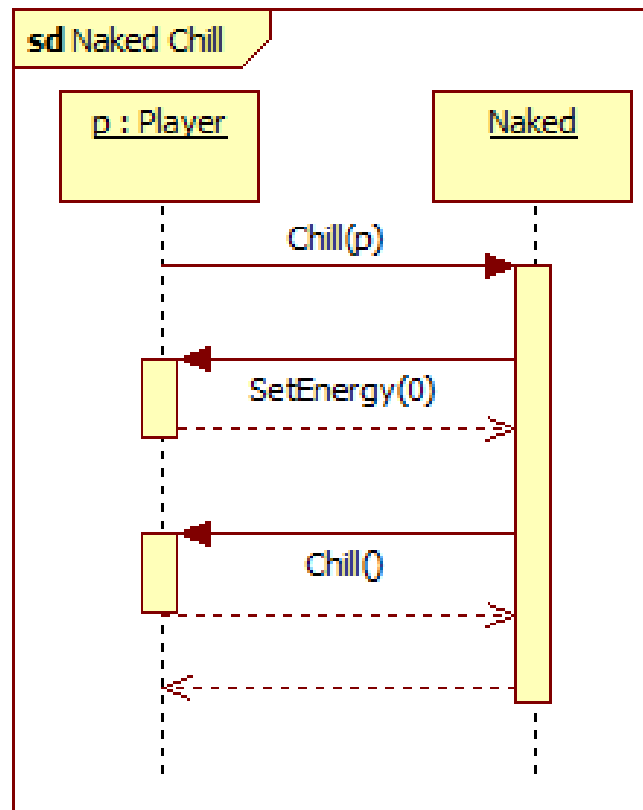
3.25. ábra. `Tile.StepOff(Player)`3.26. ábra. `Tile.GiveItem(Player)`



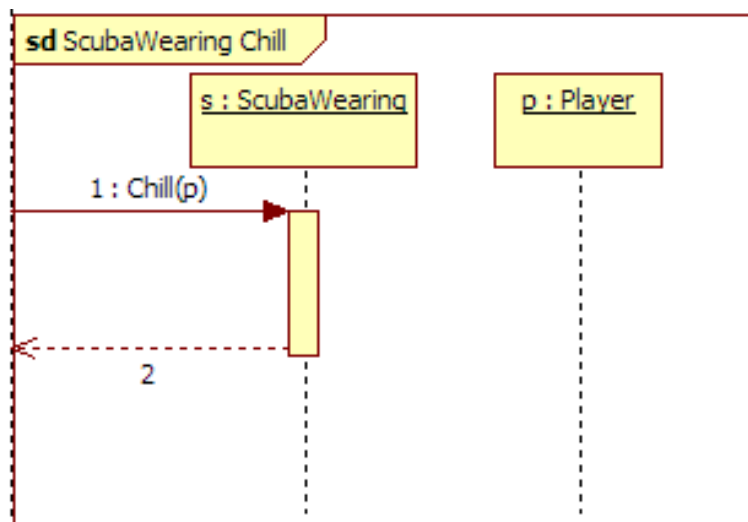
3.27. ábra. Tile.ChillWater()



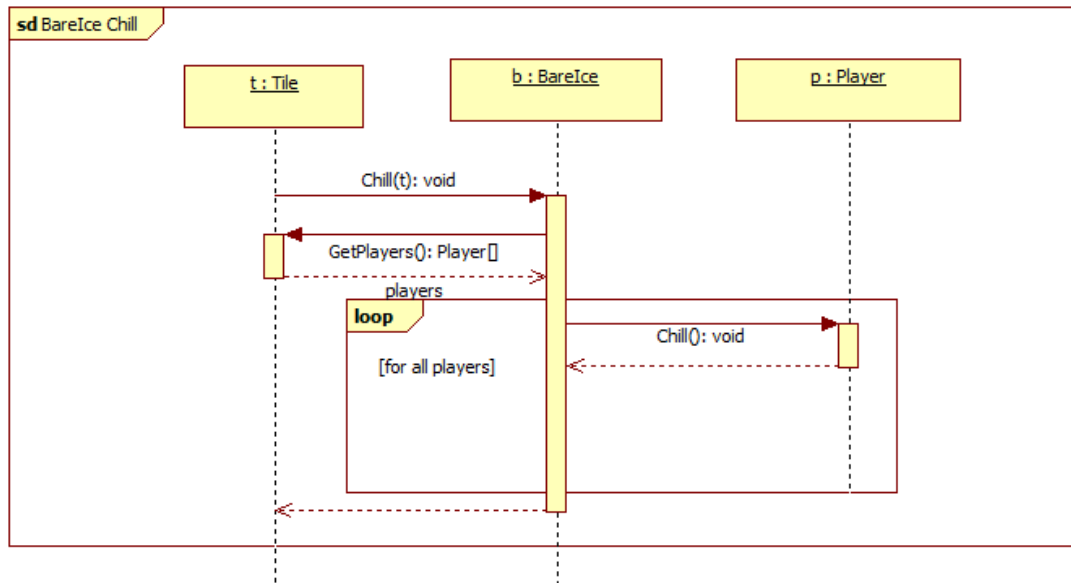
3.28. ábra. Tile.ChillStorm()



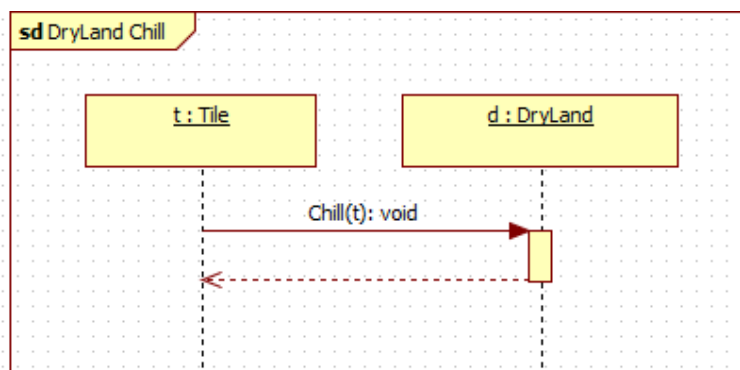
3.29. ábra. Naked.Chill(Player)



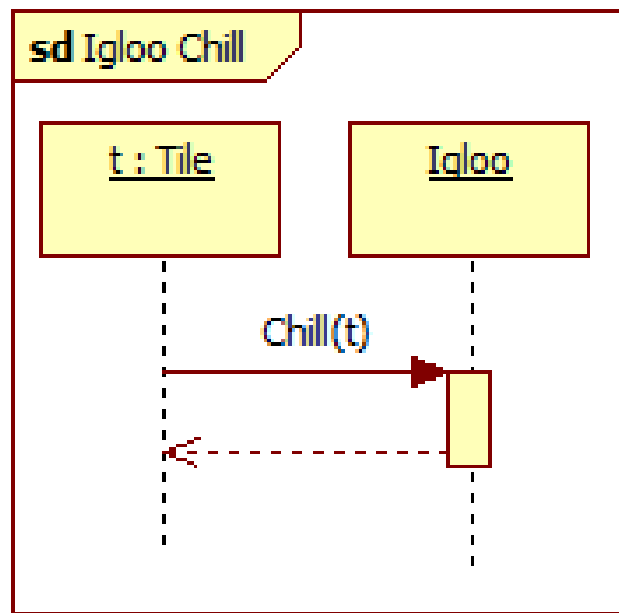
3.30. ábra. ScubaWearing.Chill(Player)



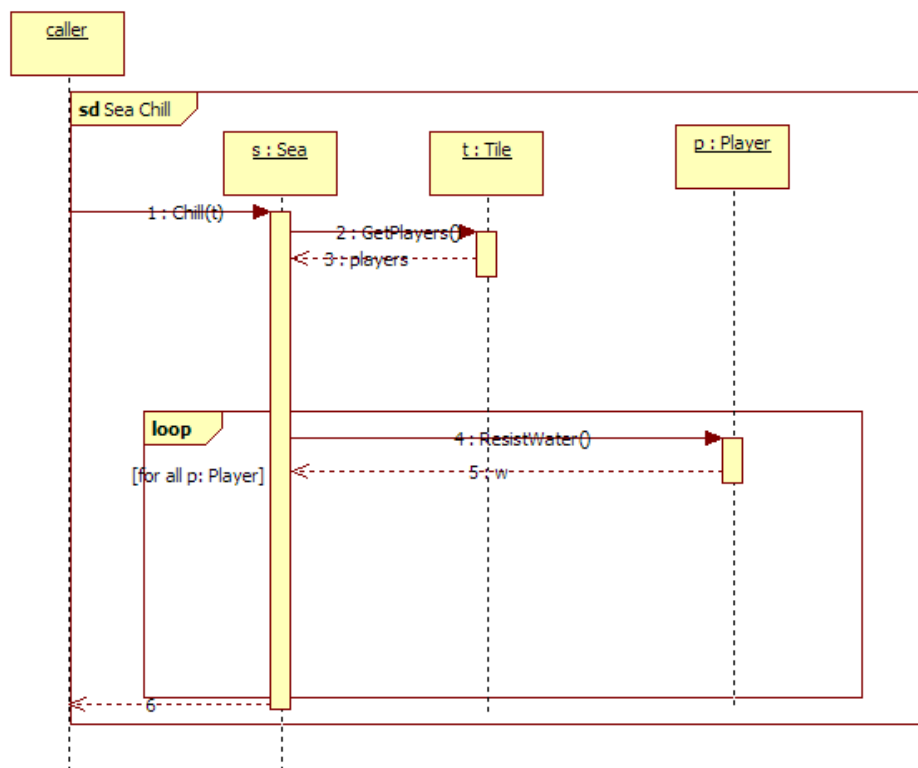
3.31. ábra. BareIce.Chill()



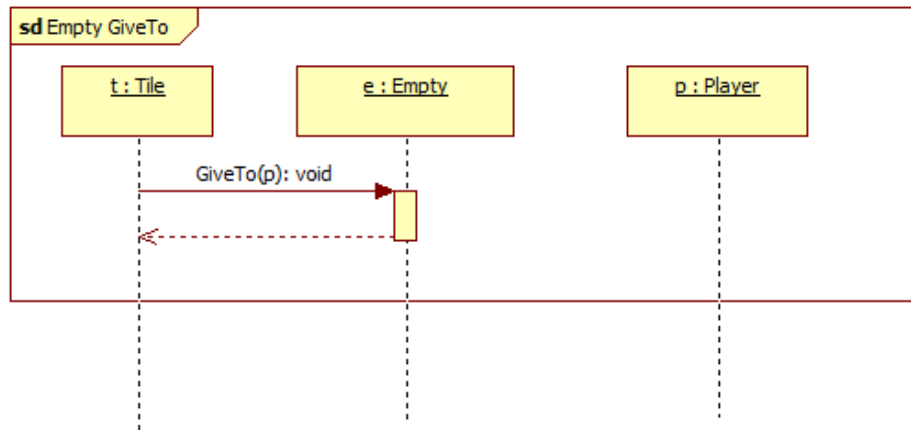
3.32. ábra. DryLand.Chill(Tile)



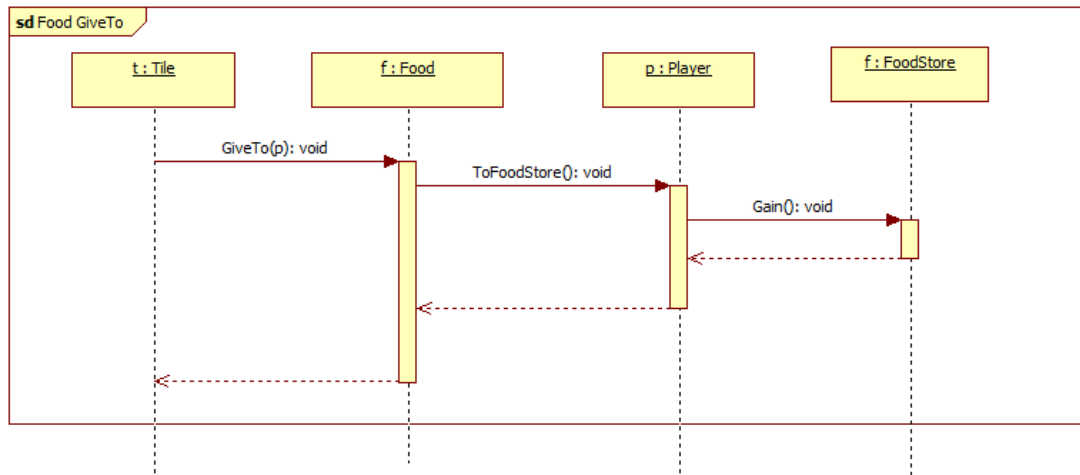
3.33. ábra. Igloo.Chill(Tile)



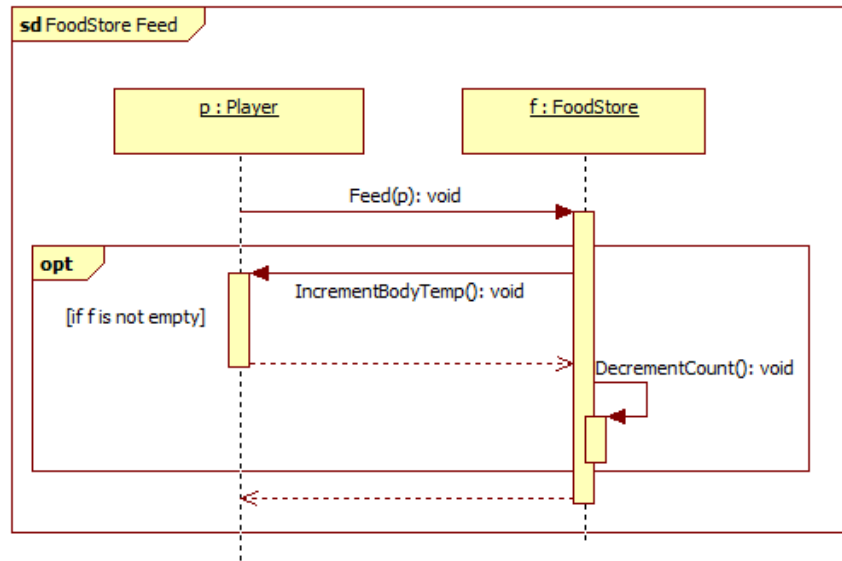
3.34. ábra. Sea.Chill(Tile)



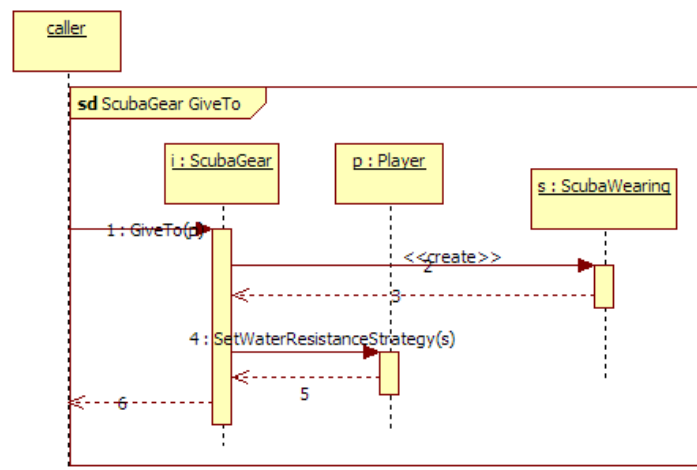
3.35. ábra. Empty.GiveTo(Player)



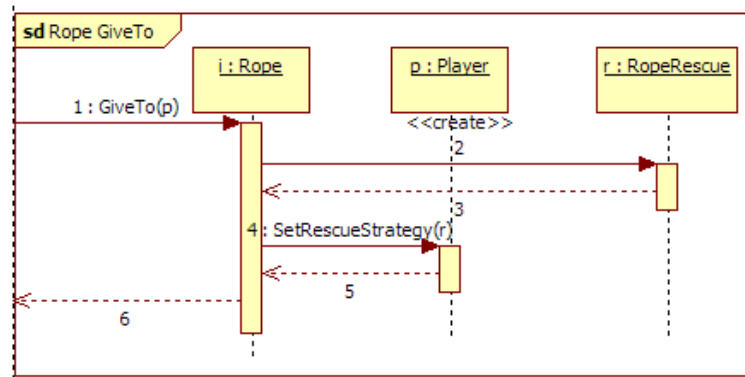
3.36. ábra. Food.GiveTo(Player)



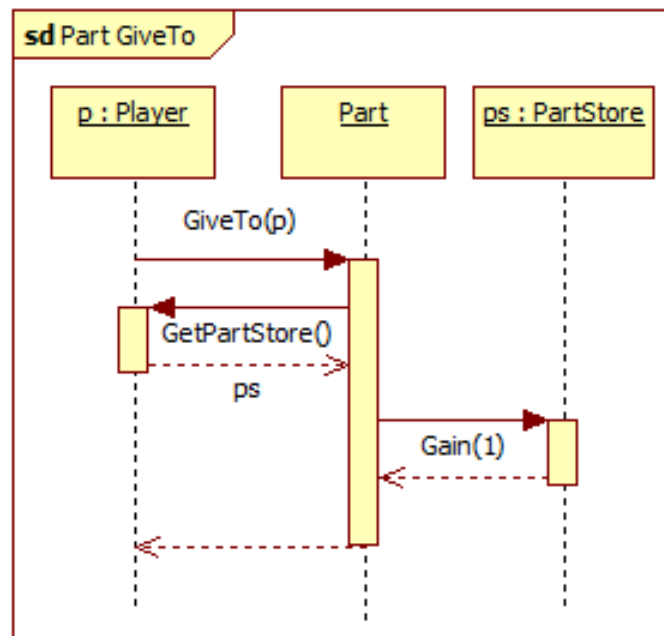
3.37. ábra. FoodStore.Feed(Player)



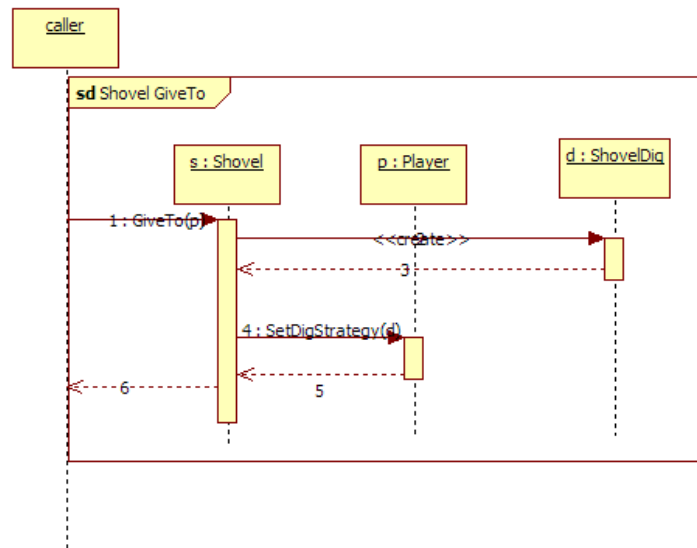
3.38. ábra. ScubaGear.GiveTo(Player)



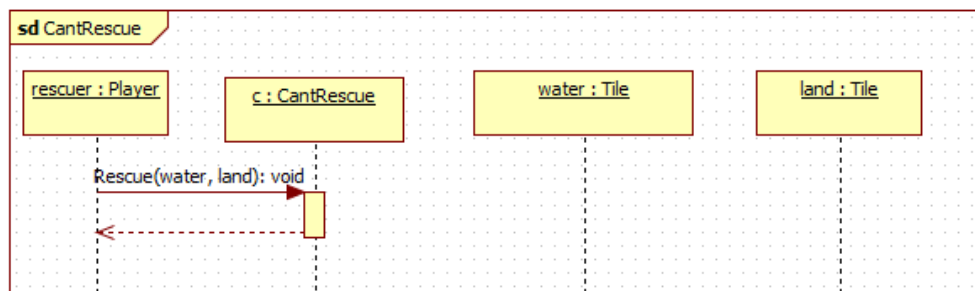
3.39. ábra. Rope.GiveTo(Player)



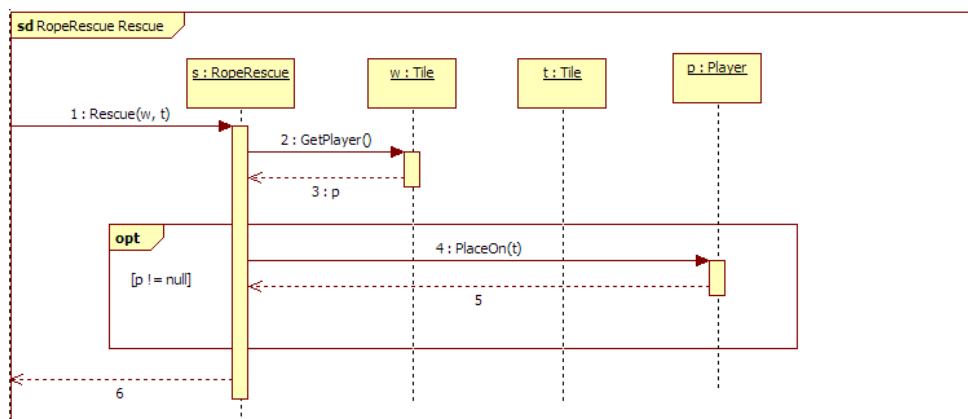
3.40. ábra. Part.GiveTo(Player)



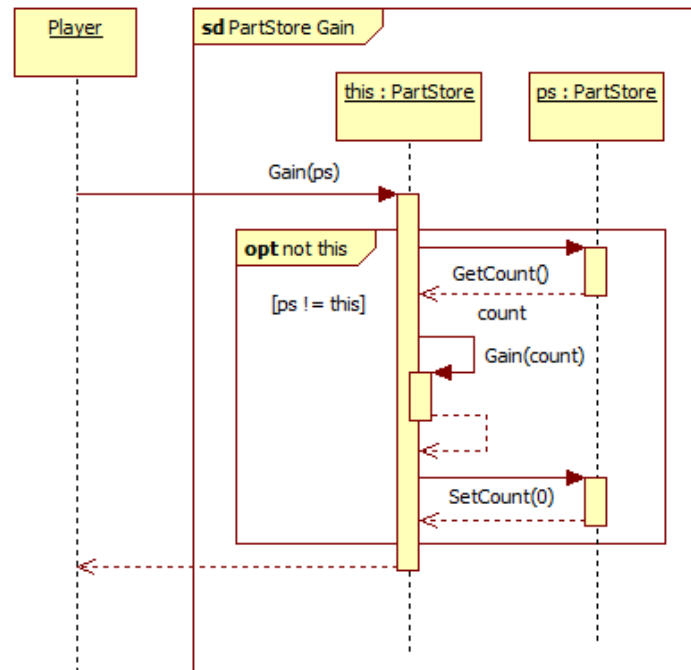
3.41. ábra. Shovel.GiveTo(Player)



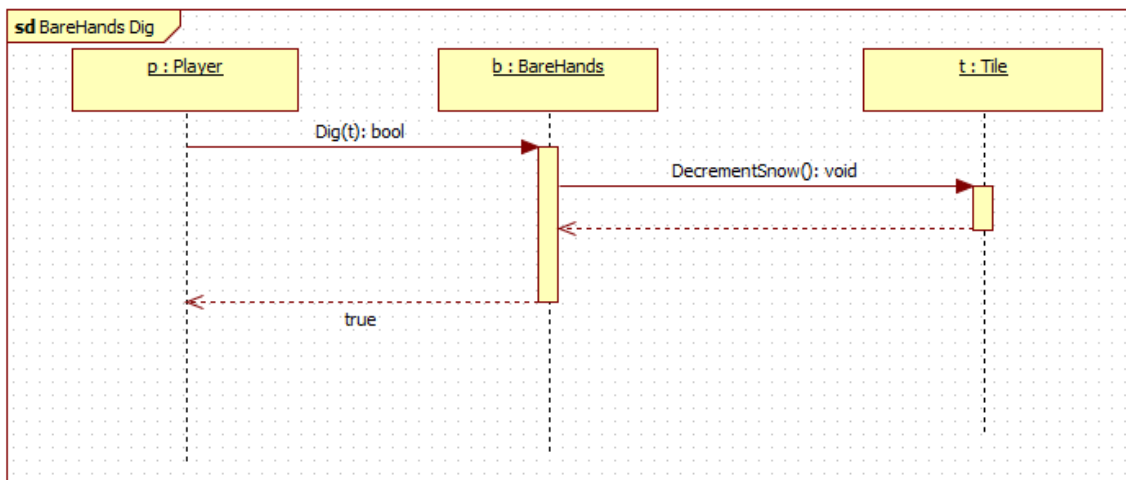
3.42. ábra. CantRescue.Rescue(Tile, Tile)



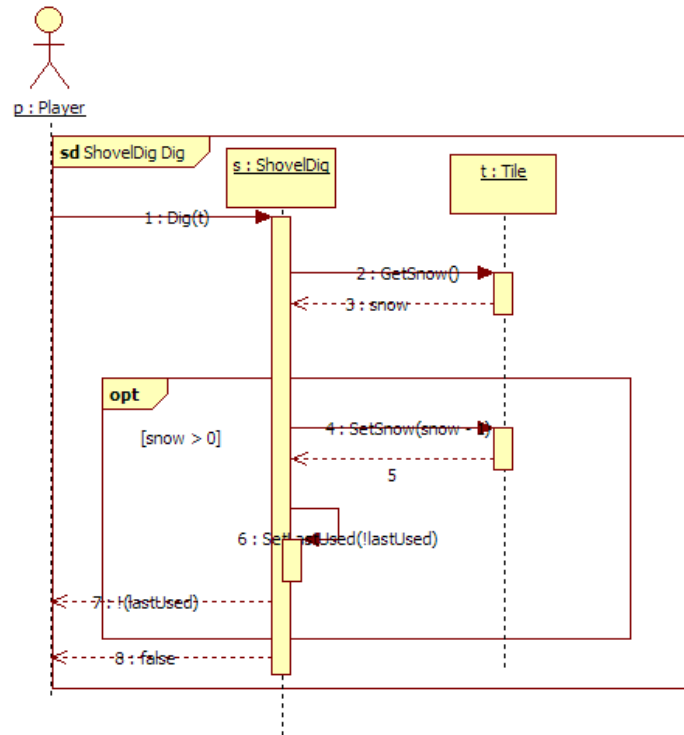
3.43. ábra. RopeRescue.Rescue(Tile, Tile)



3.44. ábra. PartStore.Gain(PartStore)



3.45. ábra. BareHandsDig.Dig(Tile)



3.46. ábra. ShovelDig.Dig(Tile)

3.6. Napló

Kezdet	Időtartam	Résztevők	Leírás
2020.02.24. 08:30	1 óra	Kiss	Class diagram rajzolás
2020.02.24. 14:00	1,5 óra	Lant	Ami kimaradt az első leadásból, use-caseknek
2020.02.24. 13:00	2 óra	Kiss	Szekvencia diagram rajzolás
2020.02.25. 09:30	1 óra	Kiss	Szekvencia diagram rajzolás
2020.02.25. 15:00	0,5 óra	Kiss	Tervezés ötletelés TODO megírása.
2020.02.23. 16:00	1 óra	Glávits	Class diagram ellenőrzés.
2020.02.25. 18.25	1 óra	Lant	Class diagram ellenőrzés.
2020.02.23. 22:45	2 óra	Glávits	Class diagram javítás
2020.02.20. 19:00	2 óra	Kiss	ötletelés
2020.02.20. 19:00	2 óra	Glávits	ötletelés
2020.02.20. 20:45	2 óra	Glávits	class diagram
2020.02.29.	reggeltől estig	Lant Glávits Kiss	ötletelés a class diagramról
2020.02.29. 22.00	20 perc	Lant	objektum jegyzék
2020.03.01. 11:00	1 óra	Glávits	osztály jegyzék írása
2020.03.01. 14:00	2 óra	Glávits	szekvencia
2020.03.01. 14:00	2 óra	Kiss	szekvencia
2020.03.01. 16:00	4 óra	Máté	Szekvencia diagram készítés
2020.03.01. 18:00	3 óra	Konrád	szekvencia
2020.03.01. 18:00	2 óra	Lant	class diagram leírások
2020.03.01. 19:00	5 óra	Glávits Kiss	minden szart csinál
2020.03.01. 20:30	1 óra	Máté	Szekvencia diagram javítás
2020.03.01. 22:00	2 óra	Konrád	Dokumentum szerkesztés
2020.03.01. 23:30	2,5 óra	Máté	Dokumentum formázás
2020.03.02. 00:00	45 perc	Glávits	Typo-k javítása
2020.03.02. 00:30	2,5 óra	Konrád	Dokumentum formázás
2020.03.02. 03:00	0,5 óra	Máté	Utolsó simítások
2020.03.02. 03:00	0,5 óra	Konrád	Utolsó simítások ellenőrzése