

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 1.

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	4
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	6
3.3.7	DryLand	6
3.3.8	Empty	6
3.3.9	Eskimo	6
3.3.10	Food	6
3.3.11	FoodStore	7
3.3.12	Game	7
3.3.13	Igloo	7
3.3.14	Naked	8
3.3.15	Part	8
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	10
3.3.22	ScubaGear	10
3.3.23	Sea	10
3.3.24	ShovelDig	10
3.3.25	Tile	11
3.3.26	WaterResistanceStrategy	11
3.4	Statikus struktúra diagramok	11
3.5	Szekvencia diagramok	12
3.6	State-chartok	31
3.7	Napló	32

Ábrák jegyzéke

. x	4
. Game.CreateHole()	12
. Game.CreatePolarExplorer()	12
. Player.PickUp()	13
. DryLand.Chill(Tile)	13
. ShovelDig.Dig(Tile)	14
. Empty.GiveTo(Player)	14
. ScubaWearing.Chill(Player)	15
. Player.AssembleFlare()	15
. Game.InitPlayer()	16
. Igloo.Chill(Tile)	16
. Game.Turn()	17
. Game.CreateEskimo()	17
. Tile.StepOff(Player)	18
. Tile.GiveItem(Player)	18
. Sea.Chill(Tile)	19
. Game.CreateIce()	19
. Player.RescueTeammate(direction)	20
. Player.Dig()	20
. CantRescue.Rescue(Tile, Tile)	20
. ScubaGear.GiveTo(Player)	21
. Player.Step(direction)	21
. Food.GiveTo(Player)	21
. Rope.GiveTo(Player)	22
. Naked.Chill(Player)	22
. PolarExplorer.Examine(direction)	23
. Part.GiveTo(Player)	23
. Game.GenerateItem()	24
. Player.AssembleFlare()	24
. Player.PlaceOn(Tile)	25
. RopeRescue.Rescue(Tile, Tile)	25
. Shovel.GiveTo(Player)	25
. Tile.StepOn(Player)	26
. Player.EatFood()	26
. PartStore.Gain(PartStore)	27
. Game.CreateUnstableIce()	27
. Tile.ChillWater()	28
. Game.CreateSea()	28
. Tile.ChillStorm()	29
. BareHandsDig.Dig(Tile)	29
. BareIce.Chill()	29
. Eskimo.BuildIgloo()	30
. FoodStore.Feed(Player)	30
. Player.Chill()	31
. Player.ResistWater()	31

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 a egy adott tábláról.

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 átvészélhetők a hóviharak

3.2. Statikus struktúra diagramok

[Az előző alfejezet osztályainak kapcsolatait és publikus metódusait bemutató osztálydiagram(ok). Tipikus hibalehetőségek: csillag-topológia, szigetek.]

3.1. ábra. x

3.3. Osztályok leírása

[Az előző alfejezetben tárgyalt objektumok felelősségének formalizálása attribútumokká, metódusokká. Csak publikus metódusok szerepelhetnek. Ebben az alfejezetben megjelennek az interfészek, az öröklés, az absztrakt osztályok. Segédosztályokra még mindig nincs szükség. Az osztályok ABC sorrendben kövessék egymást. Interfészek esetén az Interfészek, Attribútumok pontok kimaradnak.]

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 található játékos 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 Tile így hűti a viharban a játékosokat
- Metódusok
 - abstract void Chill(Tile t)

3.3.5. ChillWaterStrategy

- A Tile így hűti 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 típus, akivel valaki játszhat
- Ősosztályok
Player
- Metódusok
 - void BuildIgloo(): Épít egy iglut az adott mezőre

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 modell és a kontroller 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. 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.
 - 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 a ruha 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 Take(PartStore ps): Átveszi az alkatrészeket
 - void Gain(int n): Megnö az alkatrészek száma ami a játékosnál van
 - void Build(): Összerakja a rakétapisztolyt

3.3.17. Player

- Játékos osztály, amit a játékos 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 hogy viselkedik vízbeesés esetén
- Metódusok
 - void AssembleFlare(): Összerakja a játék végéhez szükséges rakéta pisztolyt.
 - void Chill(): A testhő 1-el csökken, ha 0 alá megy GameOver.
 - void Dig(): Ezt a metódust a controller hívja. A játékos havat ás.
 - void EatFood(): Ezt a metódust a controller hívja. A játékos eszik.
 - void PickUp(): Ezt a metódust a controller hívja. A játékos felvesz egy tárgyat.
 - 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.
 - 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.

3.3.18. PolarExplorer

- Játékos típus, akivel valaki játszhat
- Ősosztályok
Player
- Metódusok
 - void 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): Felruázza a játékos a megmentésre alkalmas eszközzel.

3.3.21. RopeRescue

- A játékos kihúzza csapattársát a vízből.
- Interfészek
RescueStrategy
- Metódusok
 - void Rescue(Tile water, Tile land): A játékos kihúzza a vízbe esett csapattársát a vízből, ha van kötele.

3.3.22. ScubaGear

- Jégbe fagyott búvárruha.
- Interfészek
GiveItemStrategy
- Metódusok
 - void GiveTo(): Felruházza a játékos a vízben maradásra alkalmas eszközzel.

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 (int)

3.3.25. Tile

- 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 eszközt vett fel valaki.
 - `neighborTiles`: `Tile[*]`: Szomszédos cellákat tárolja, ismeri.
 - `occupants`: `Player[*]`: Rajta lévő játékosok.
 - `snow`: `int`: Rajta lévő hó mennyiség
 - `weightLimit`: `int`: Rajta lévő játékosok számának maximuma.
- Metódusok
 - `void AddOccupant(Player p)`: Rátesz egy játékost a cellára.
 - `void RemoveOccupant(Player p)`: Levesz egy játékos a celláról.
 - `void ChillStorm()`: Ezt a metódust a Controller hívja. Hűti a játékosokat.
 - `void GiveItem(Player)`: A játékos megkapja a felvett tárgyat.
 - `Tile NeighborAt(direction)`: Visszaadja az adott irányba lévő cellát.
 - `StepOn(Player)`:
 - `StepOff(Player)`:

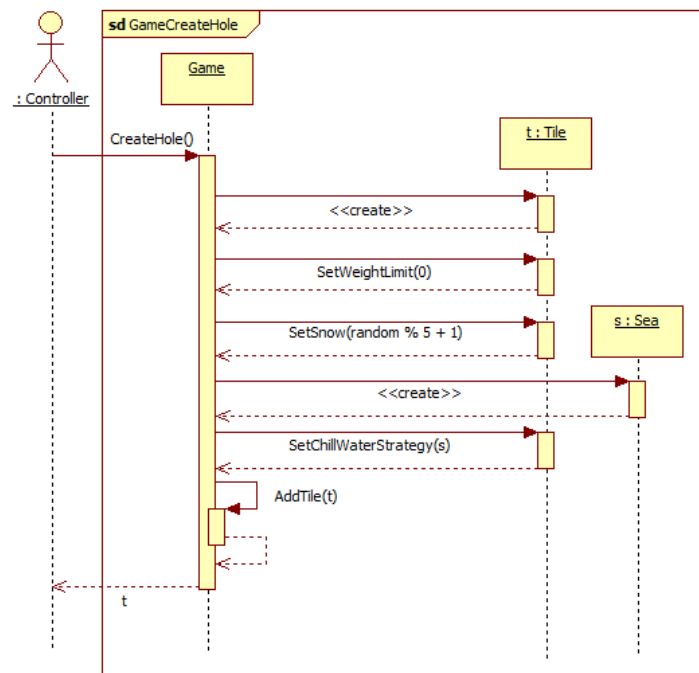
3.3.26. WaterResistanceStrategy

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

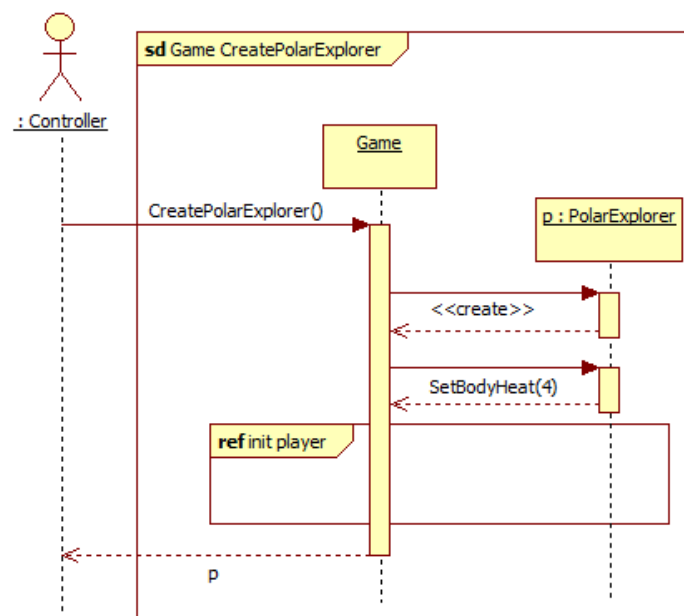
3.4. Statikus struktúra diagramok

[Az előző alfejezet osztályainak kapcsolatait és publikus metódusait bemutató osztálydiagram(ok). Tipikus hibalehetőségek: csillag-topológia, szigetek.]

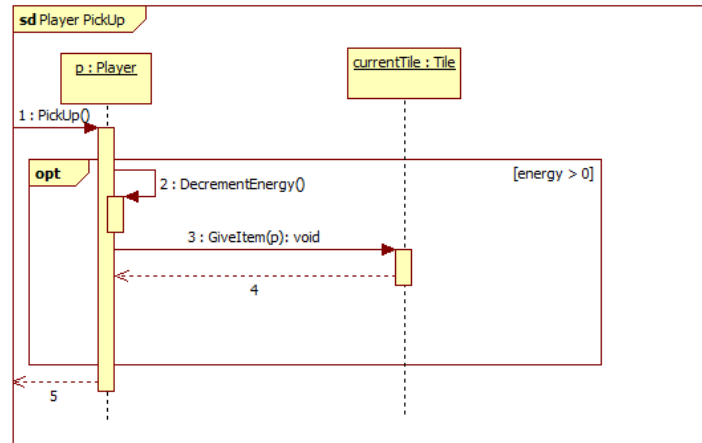
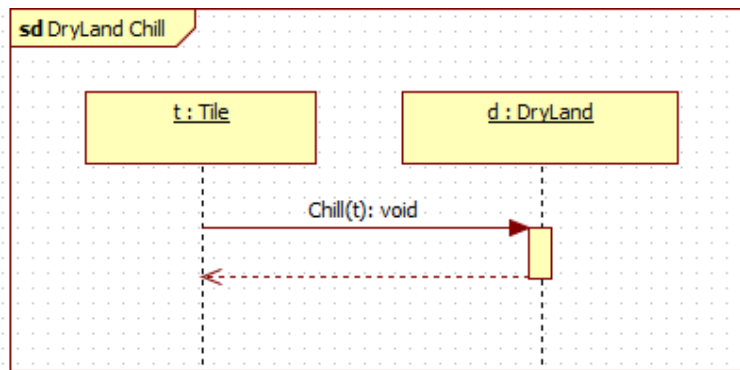
3.5. Szekvencia diagramok

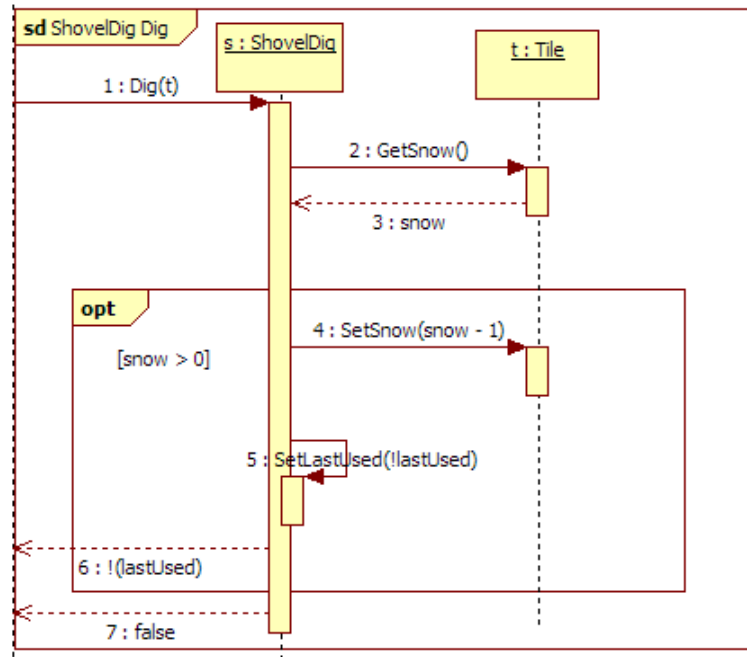


3.2. ábra. Game.CreateHole()

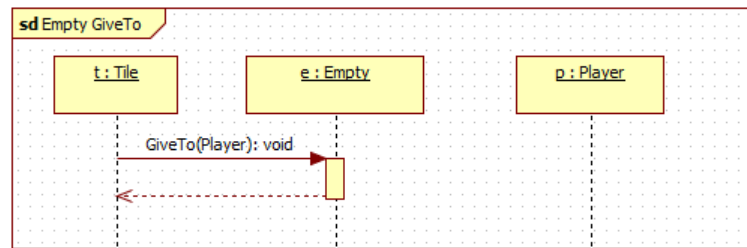


3.3. ábra. Game.CreatePolarExplorer()

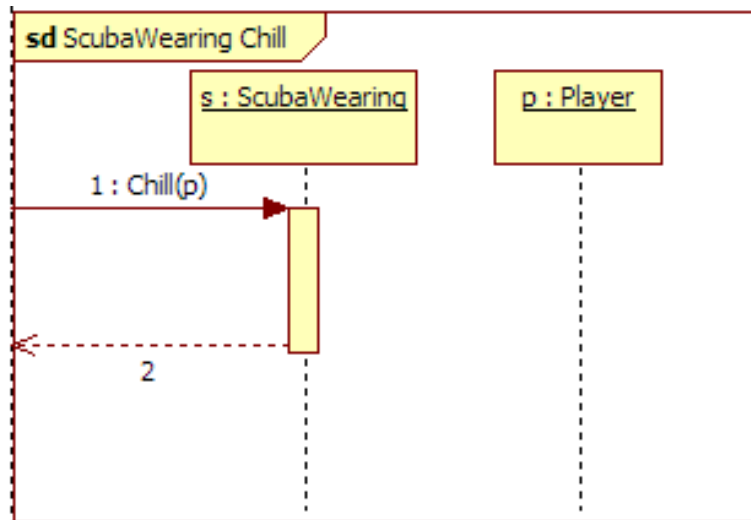
3.4. ábra. `Player.PickUp()`3.5. ábra. `DryLand.Chill(Tile)`



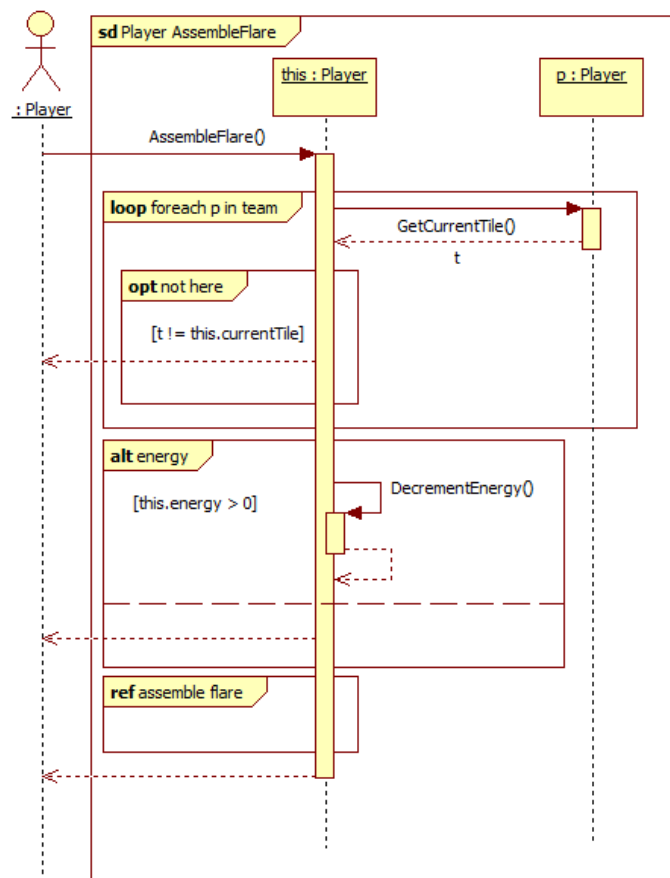
3.6. ábra. ShovelDig.Dig(Tile)



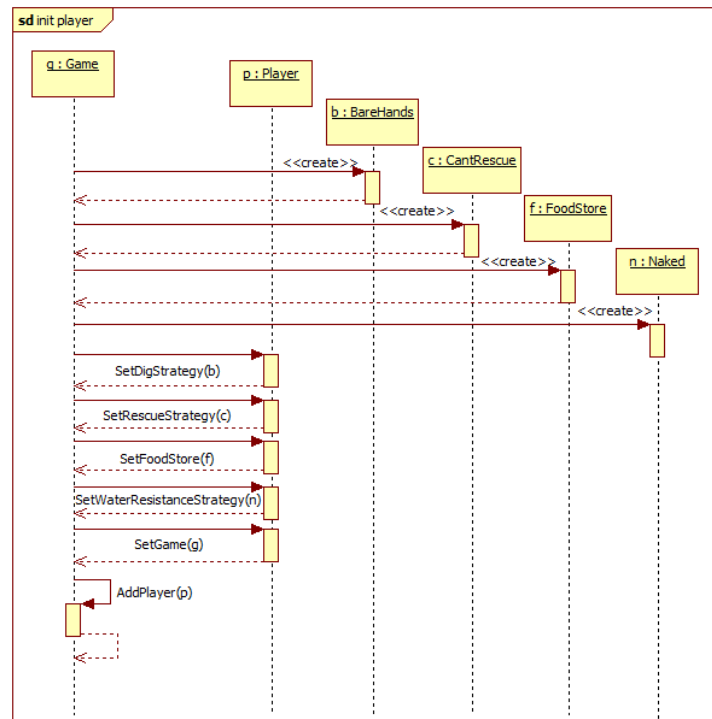
3.7. ábra. Empty.GiveTo(Player)



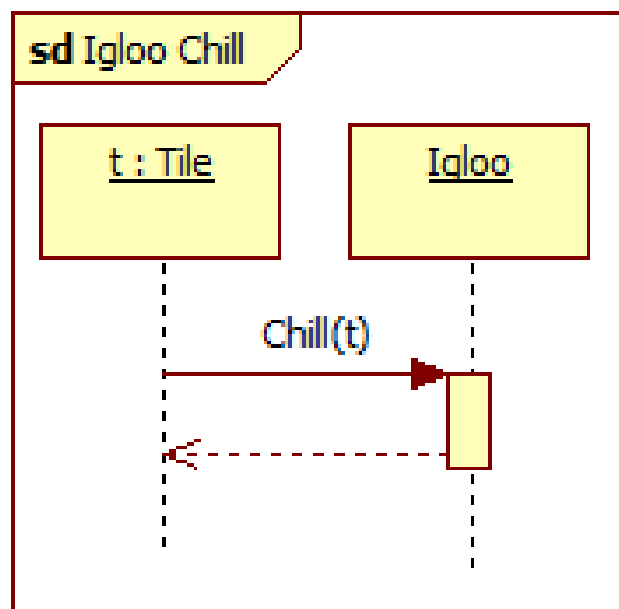
3.8. ábra. ScubaWearing.Chill(Player)



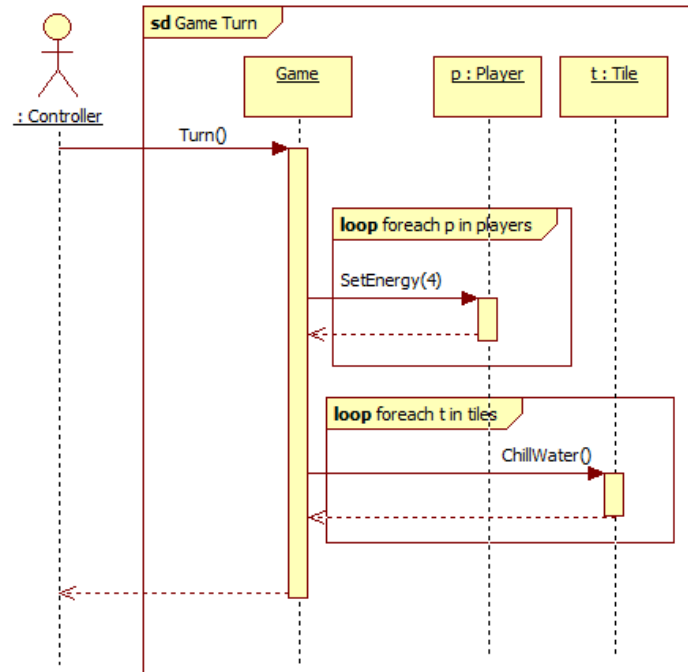
3.9. ábra. Player.AssembleFlare()



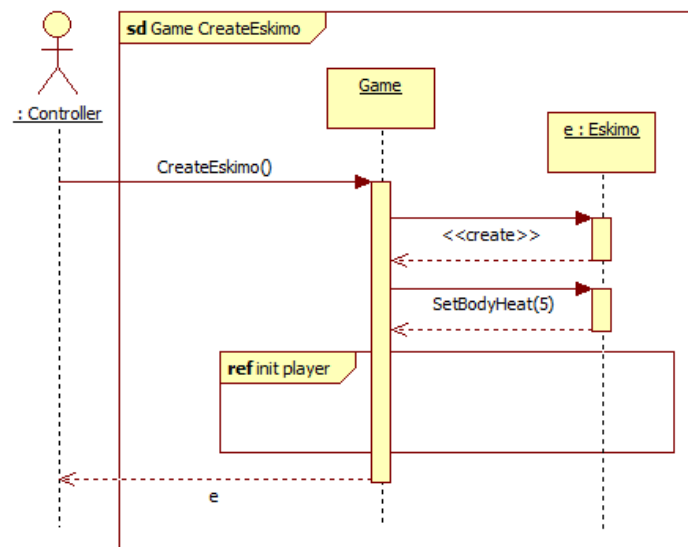
3.10. ábra. Game.InitPlayer()



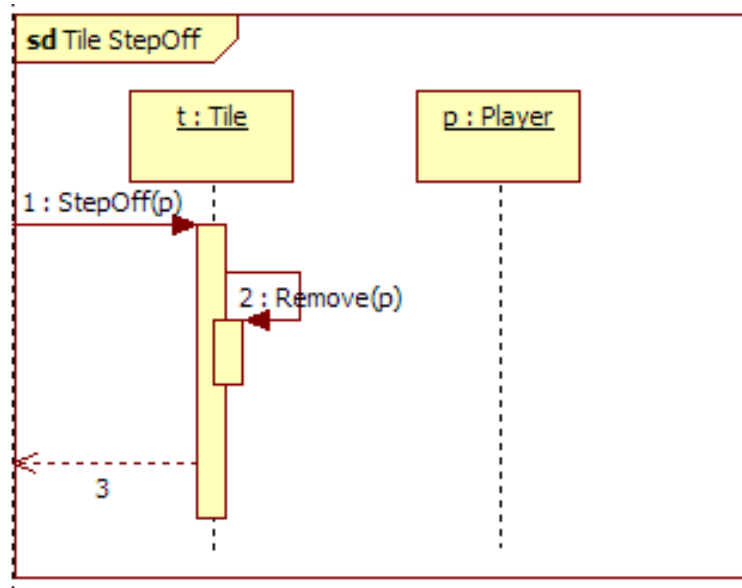
3.11. ábra. Igloo.Chill(Tile)



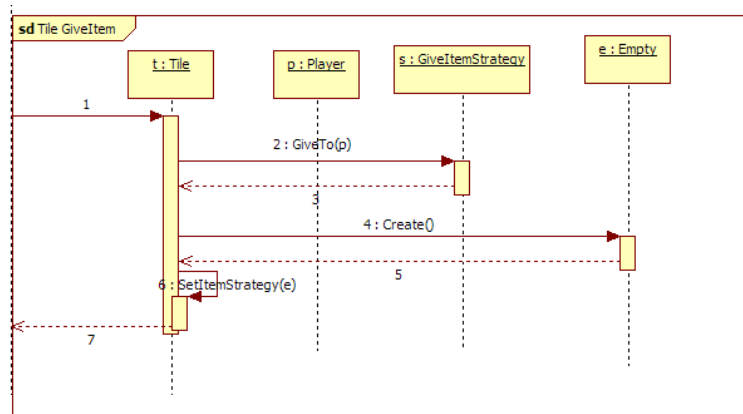
3.12. ábra. Game.Turn()



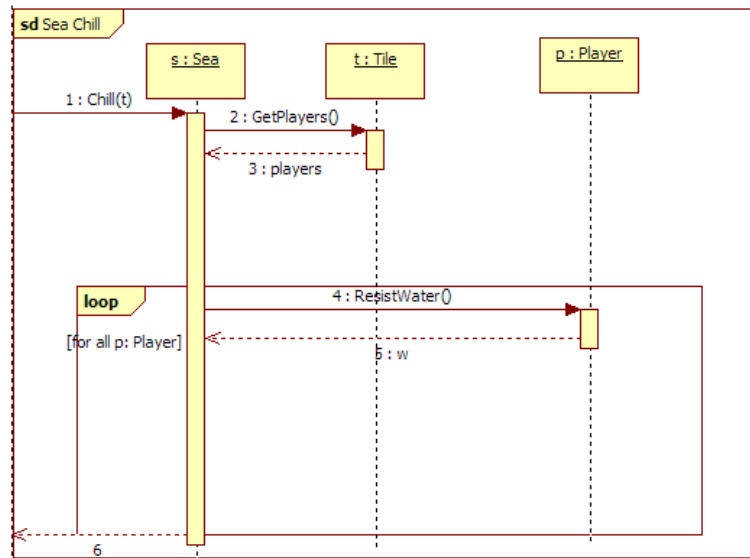
3.13. ábra. Game.CreateEskimo()



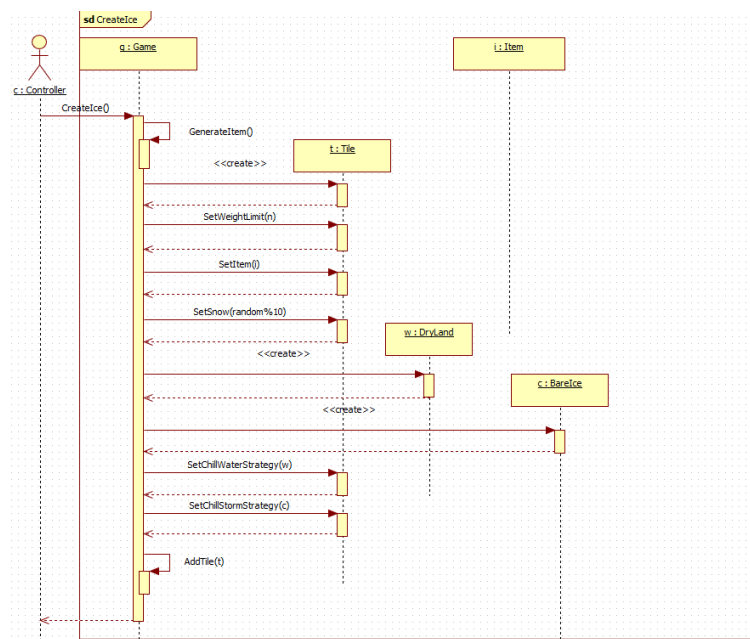
3.14. ábra. Tile.StepOff(Player)



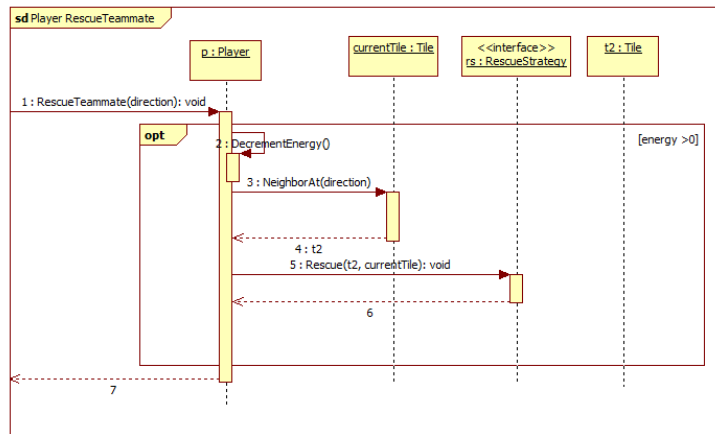
3.15. ábra. Tile.GiveItem(Player)



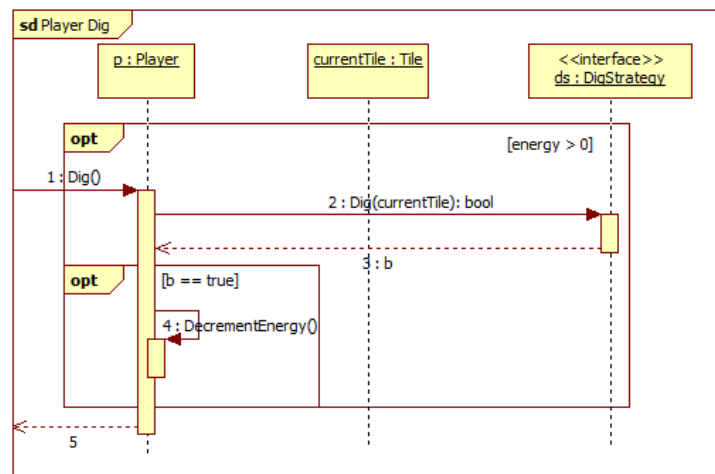
3.16. ábra. Sea.Chill(Tile)



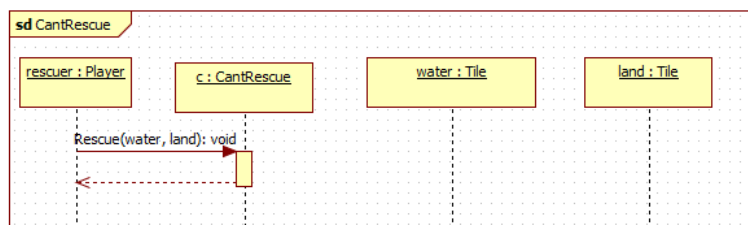
3.17. ábra. Game.CreateIce()



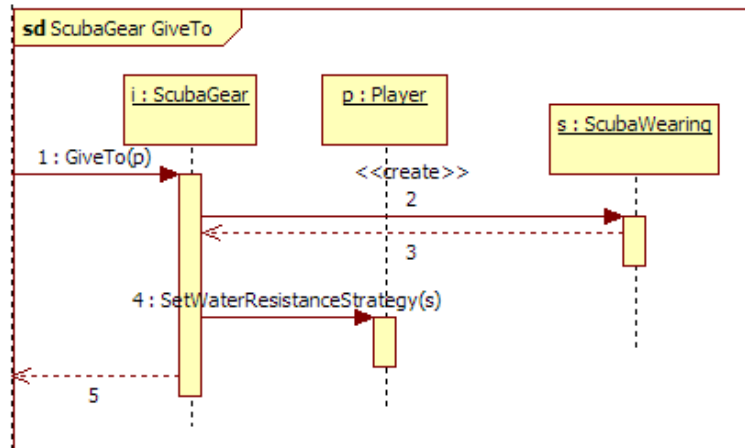
3.18. ábra. Player.RescueTeammate(direction)



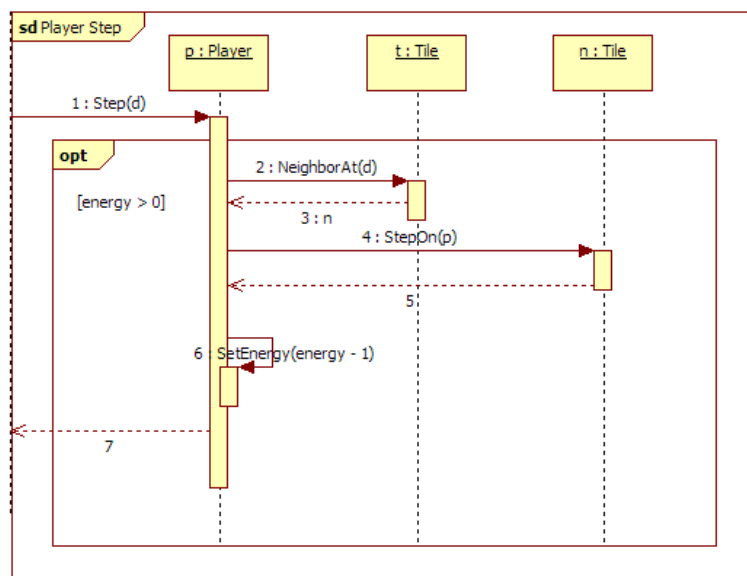
3.19. ábra. Player.Dig()



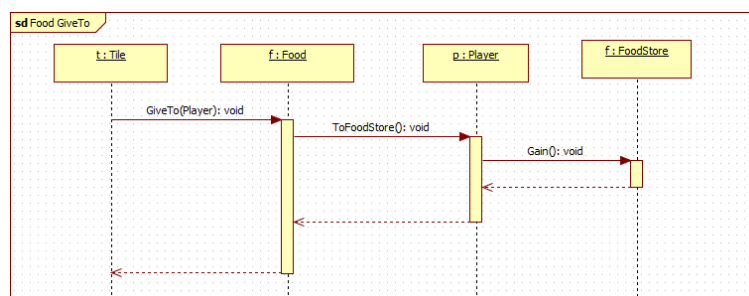
3.20. ábra. CantRescue.Rescue(Tile, Tile)



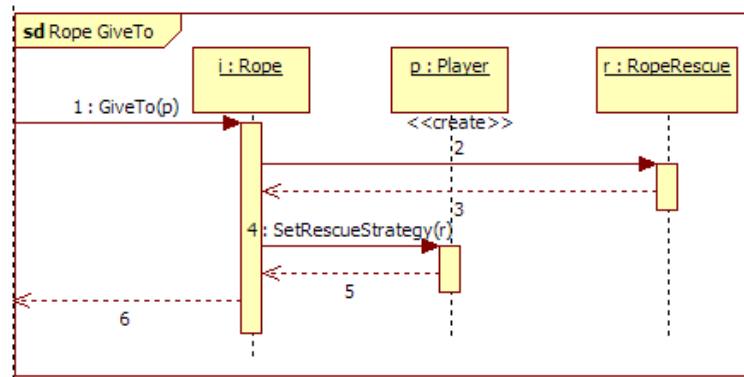
3.21. ábra. ScubaGear.GiveTo(Player)



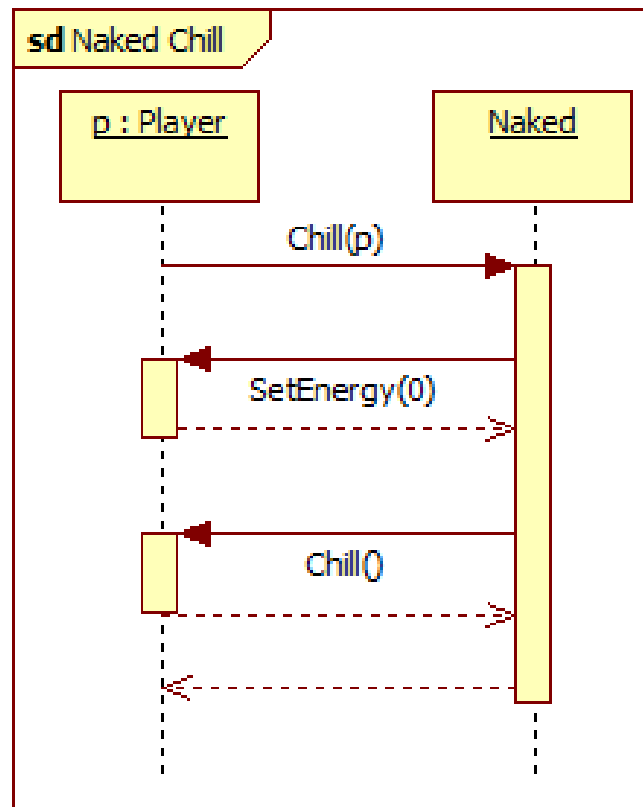
3.22. ábra. Player.Step(direction)



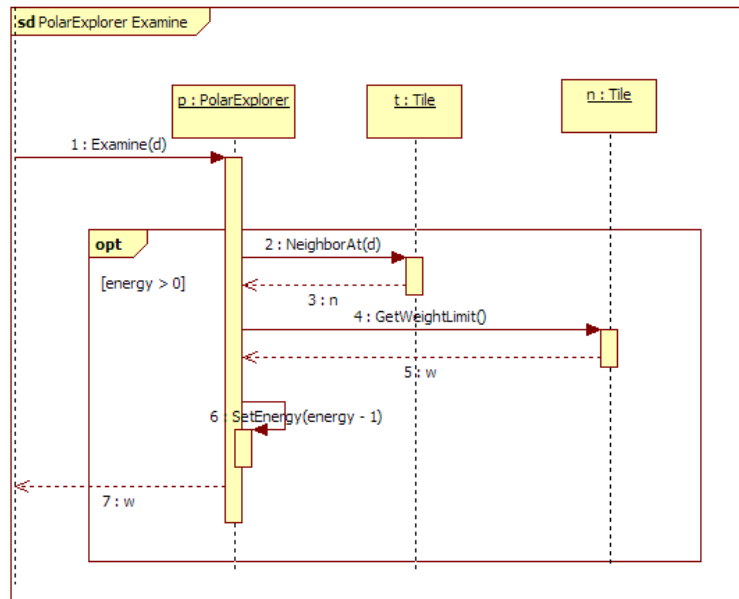
3.23. ábra. Food.GiveTo(Player)



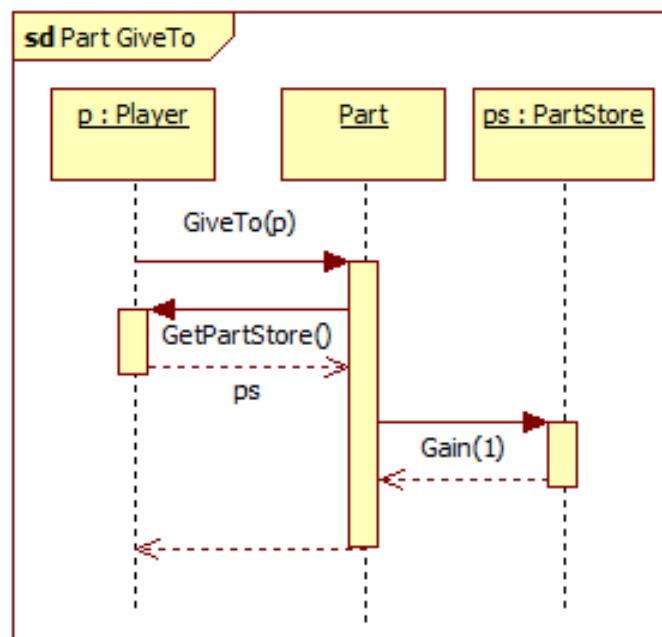
3.24. ábra. Rope.GiveTo(Player)



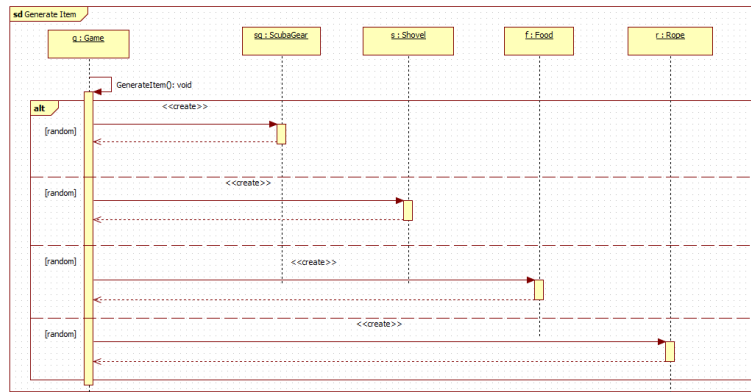
3.25. ábra. Naked.Chill(Player)



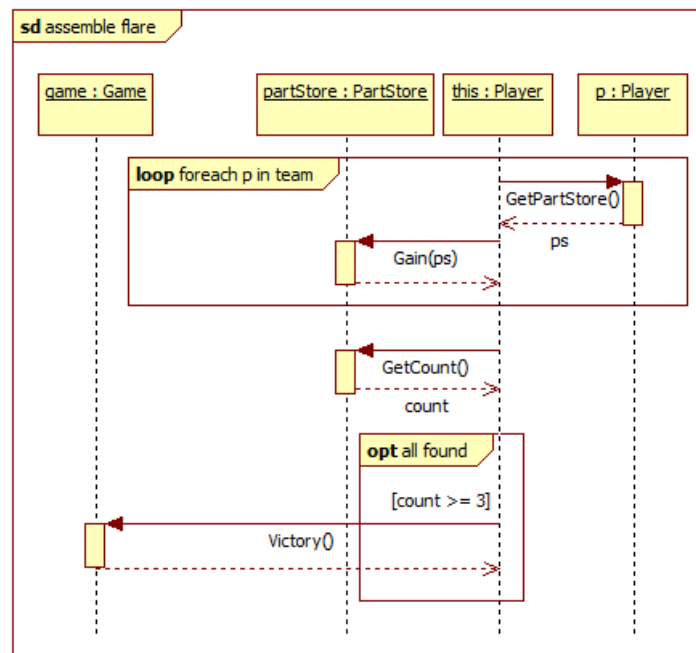
3.26. ábra. PolarExplorer.Examine(direction)



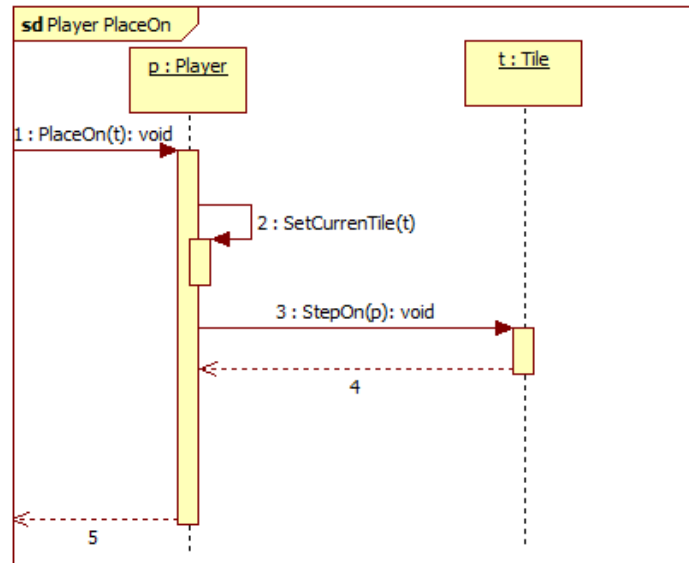
3.27. ábra. Part.GiveTo(Player)



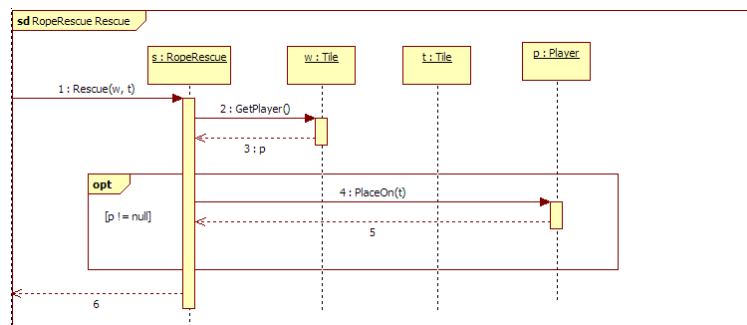
3.28. ábra. Game.GenerateItem()



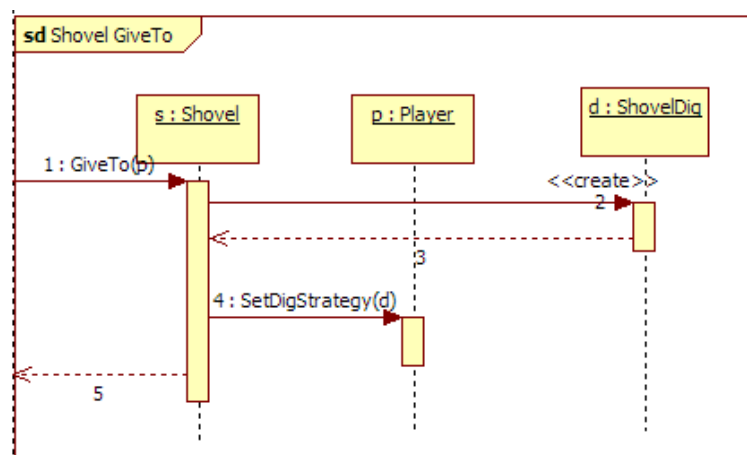
3.29. ábra. Player.AssembleFlare()



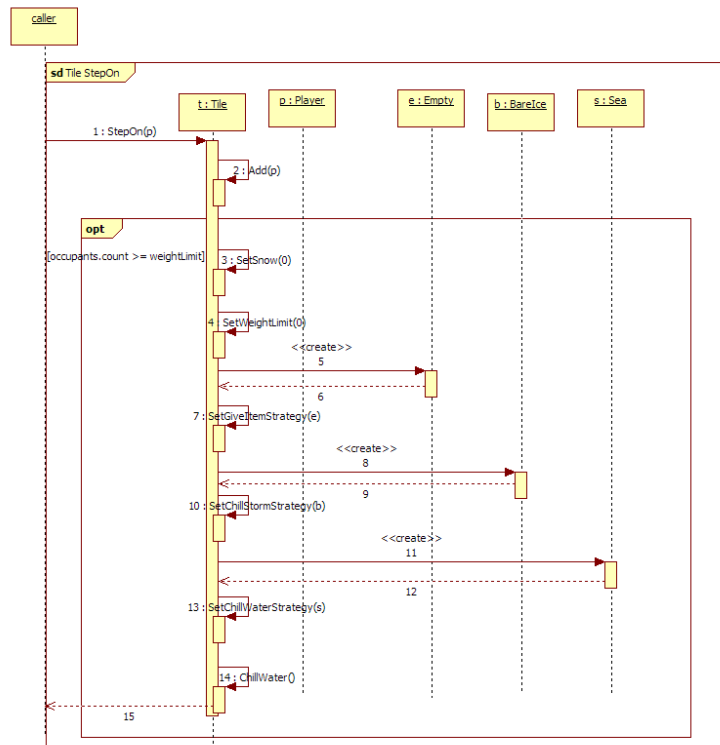
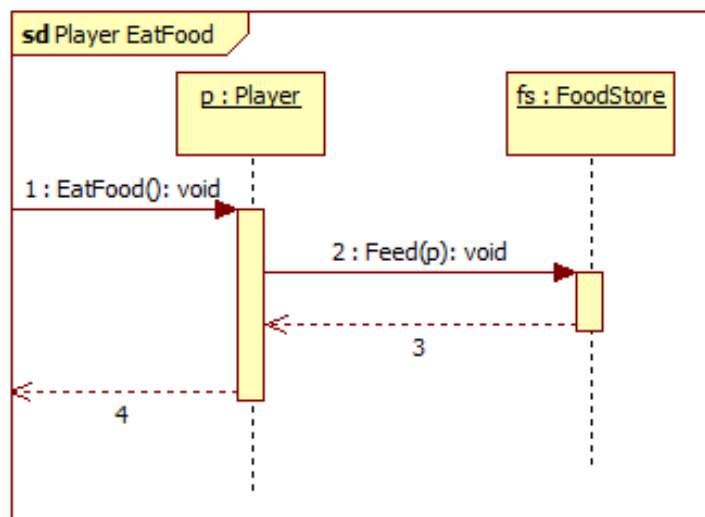
3.30. ábra. Player.PlaceOn(Tile)

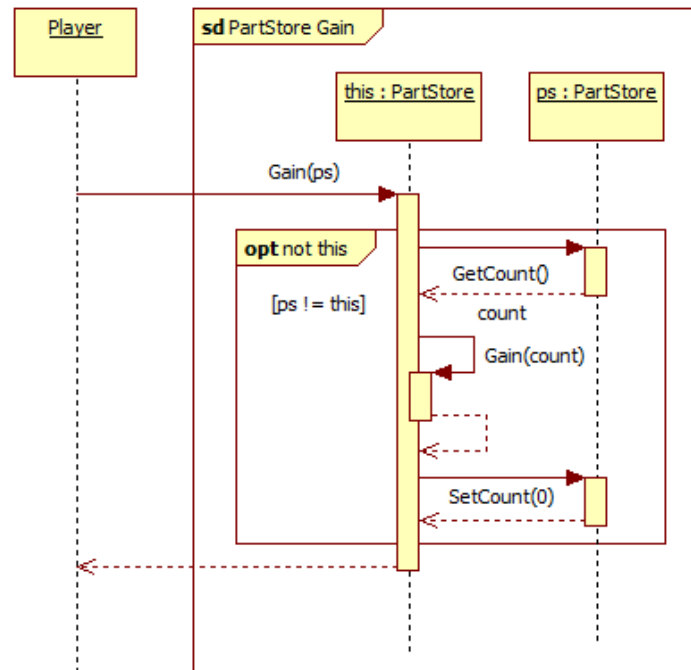


3.31. ábra. RopeRescue.Rescue(Tile, Tile)

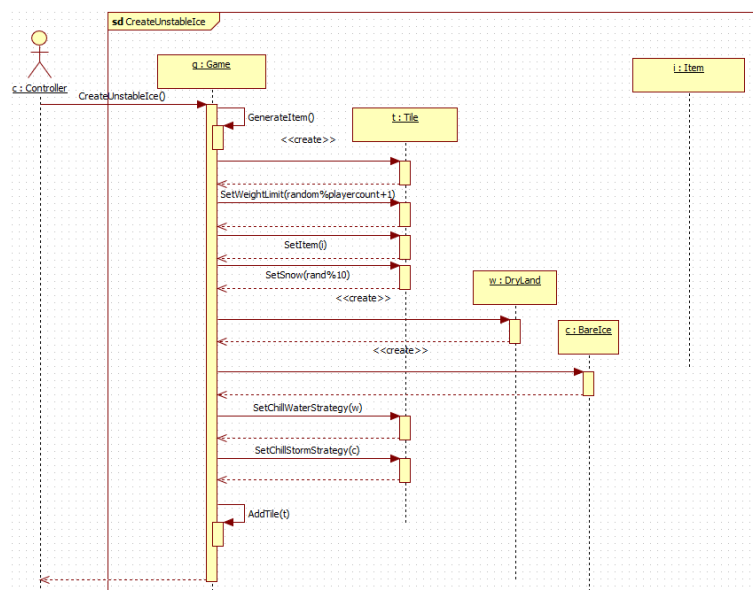


3.32. ábra. Shovel.GiveTo(Player)

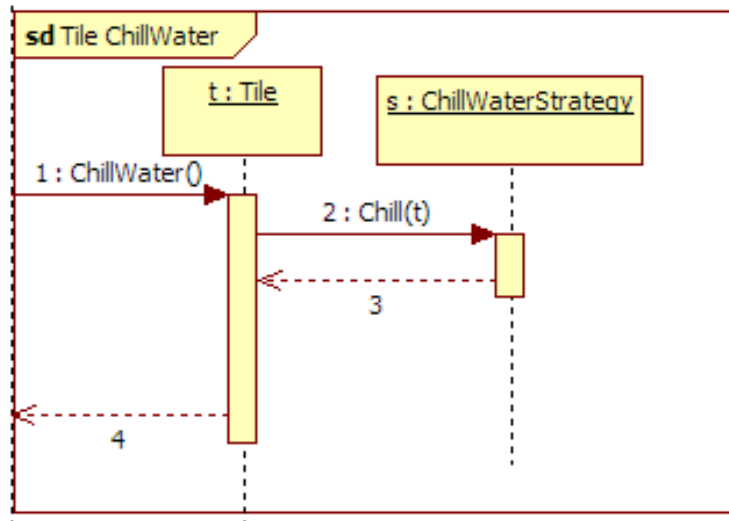
3.33. ábra. `Tile.StepOn(Player)`3.34. ábra. `Player.EatFood()`



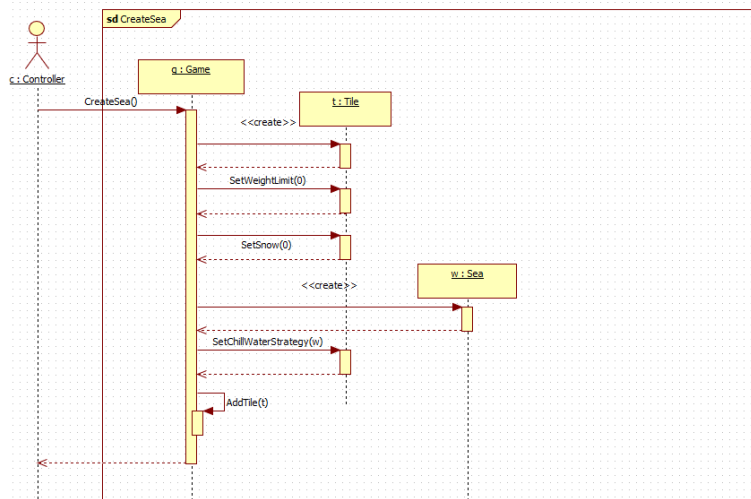
3.35. ábra. PartStore.Gain(PartStore)



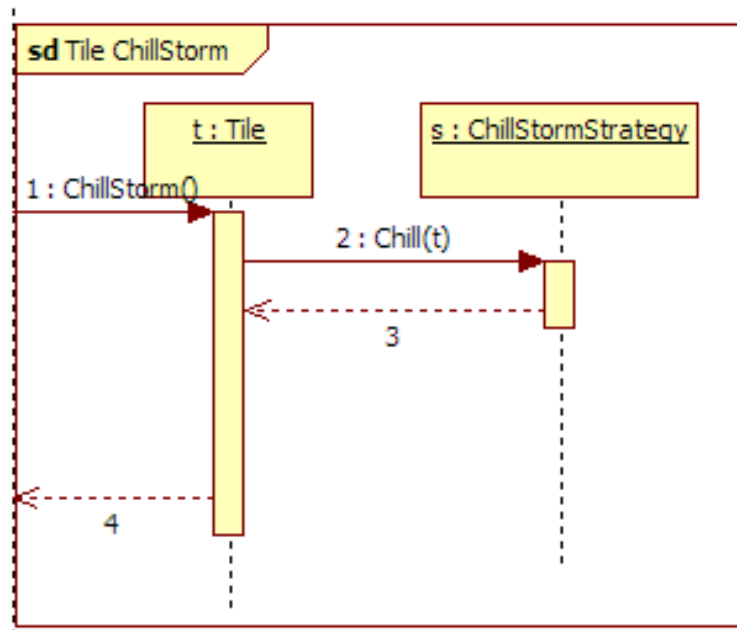
3.36. ábra. Game.CreateUnstableIce()



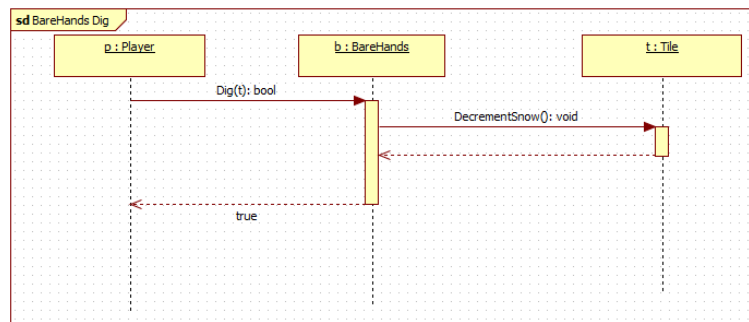
3.37. ábra. Tile.ChillWater()



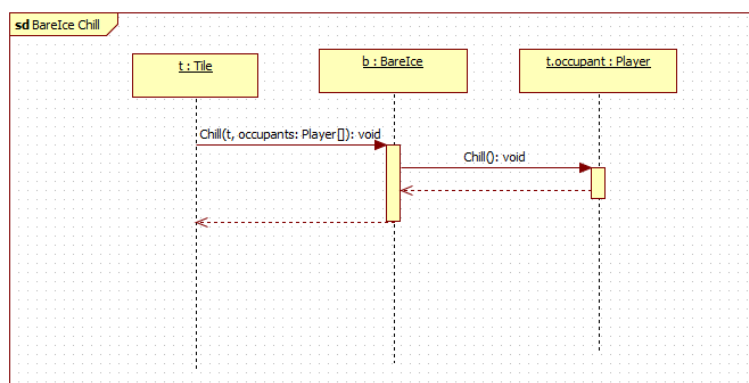
3.38. ábra. Game.CreateSea()



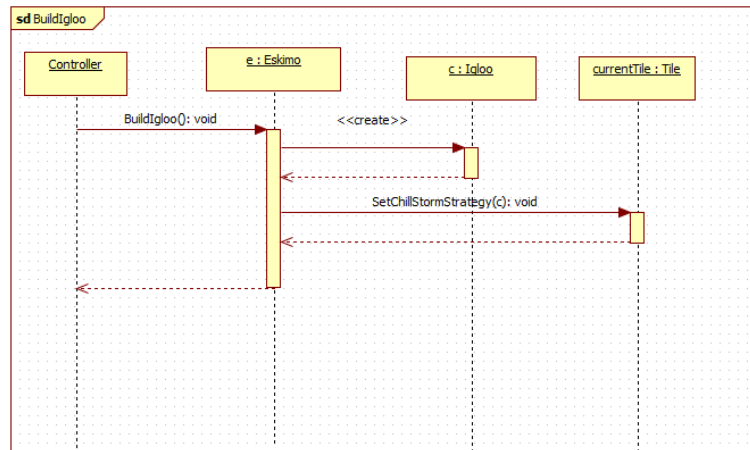
3.39. ábra. Tile.ChillStorm()



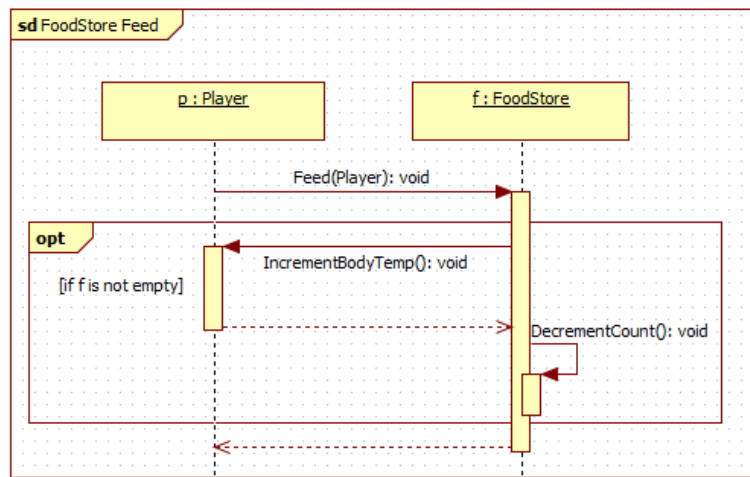
3.40. ábra. BareHandsDig.Dig(Tile)



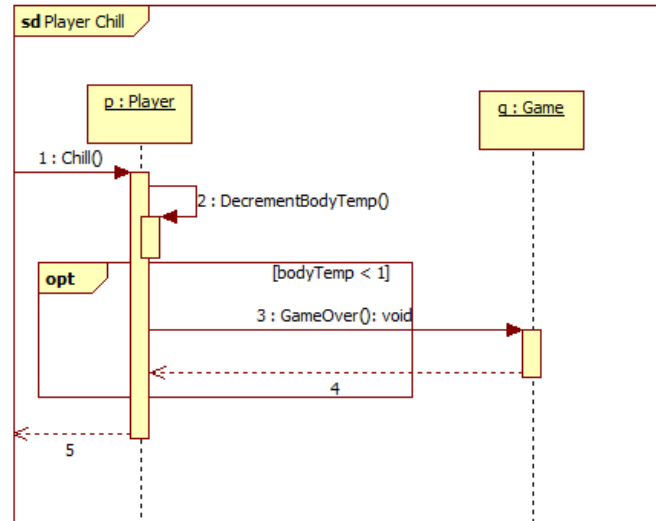
3.41. ábra. BareIce.Chill()



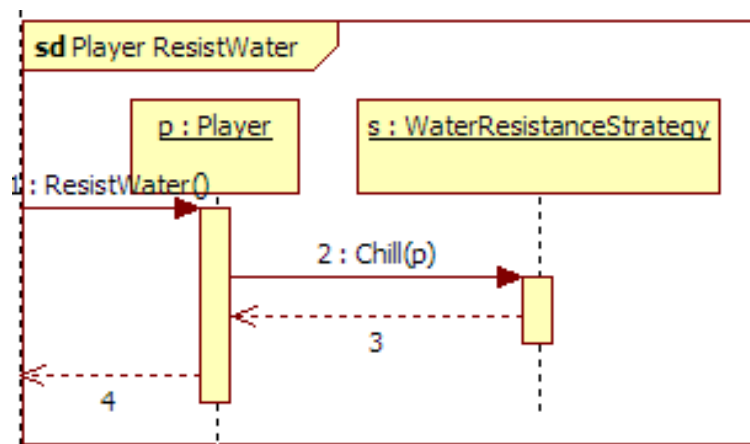
3.42. ábra. Eskimo.BuildIgloo()



3.43. ábra. FoodStore.Feed(Player)



3.44. ábra. Player.Chill()



3.45. ábra. Player.ResistWater()

3.6. State-chartok

[Csak azokhoz az osztályokhoz, ahol van értelme. Egyetlen állapotból álló state-chartok ne szerepeljenek. A játék működését bemutató state-chart-ot készíteni tilos.]

3.7. Napló

Kezdet	Időtartam	Résztevők	Leírás
2010.03.21. 18:00	2,5 óra	Horváth Németh Tóth Oláh	Értekezlet. Döntés: Horváth elkészíti az osztálydiagramot, Oláh a use-case leírásokat.
2010.03.23. 23:00	5 óra	Németh	Tevékenység: Németh implementálja a tesztelő programokat.
2020.03.01. 16:00	3,5 óra	Máté	Szekvencia diagram készítés
...