

7. Prototípus koncepciója

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. április 12.

Tartalomjegyzék

8	Részletes tervek	4
8.1	Osztályok és metódusok tervei	4
8.2	Osztályok leírása	4
8.2.1	BareHands	4
8.2.2	BareIce	4
8.2.3	BuildStrategy	4
8.2.4	BreakingShovel	4
8.2.5	BreakingShovelDig	5
8.2.6	CantRescue	5
8.2.7	ChillWaterStrategy	5
8.2.8	DigStrategy	5
8.2.9	DryLand	5
8.2.10	Empty	6
8.2.11	Entity	6
8.2.12	Eskimo	6
8.2.13	Food	6
8.2.14	FoodStore	7
8.2.15	Game	7
8.2.16	Igloo	7
8.2.17	Item	8
8.2.18	Naked	8
8.2.19	Part	8
8.2.20	PartStore	8
8.2.21	Player	8
8.2.22	PolarBear	9
8.2.23	PolarExplorer	10
8.2.24	RescueStrategy	10
8.2.25	Rope	10
8.2.26	RopeRescue	10
8.2.27	ScubaGear	10
8.2.28	ScubaWearing	11
8.2.29	Sea	11
8.2.30	Shovel	11
8.2.31	ShovelDig	11
8.2.32	TentKit	12
8.2.33	Tile	12
8.2.34	WaterResistanceStrategy	12
8.2.35	Osztály1	13
8.2.36	Osztály2	13
8.3	A tesztek részletes tervei, leírásuk a teszt nyelven	13
8.3.1	Teszteset1	14
8.3.2	Teszteset2	14
8.4	A tesztelést támogató programok tervei	14
8.5	Napló	14

Ábrák jegyzéke

8. Részletes tervek

8.1. Osztályok és metódusok tervei

8.2. Osztályok leírása

8.2.1. 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.

8.2.2. BareIce

- Ilyen a jégtábla, ha nincs rajta iglu. A jégtáblán nincs védelem a vihar elől.
- Ősosztályok:
 - Shelter
- Metódusok:
 - void ChillStorm(Tile t): A paraméterként kapott t Tile-n álló játékosok testhője csökken.
 - void BearAttack(Tile t):
 - void Break(Tile t):

8.2.3. BuildStrategy

- A játékos így képes építeni. Iglut vagy sátrat.
- Attribútumok:
 - count: int:
- Metódusok:
 - void Build(): Épít valamit a táblára.
 - void Gain(): Kap egy sátrat.

8.2.4. BreakingShovel

- Törhető ásó osztály.
- Interfészek:
 - Item
- Metódusok:
 - void GiveTo(Player p): A játékos így kap ásót.

8.2.5. BreakingShovelDig

- A játékos így ás, ha törhető ásó van nála.
- Interfészek:
 - DigStrategy
- Attribútumok:
 - lastUsed: bool: Volt-e használva a körben.
 - durability: int: Mennyiszer lehet még ásni vele.
- Metódusok:
 - bool Dig(Tile t): Csökkenti a tile-on található hó mennyiségét.

8.2.6. 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.

8.2.7. 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.

8.2.8. 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.

8.2.9. 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.

8.2.10. 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.

8.2.11. Entity

- Entitás osztály ami a pályát tartózkodhat.
- Metódusok:
 - Step(Direction d): Lép valamilyen irányba.
 - void PlaceOn(Tile t): Ráteszi az entitást egy másik táblára. A kötél használatakor használatos.
 - void Chill(): Hűti az entitést. A testhője csökken.
 - ResistWater(): Így viselkedik vízben.
 - BearAttack(): Így viselkedik, ha megtámadja a medve.

8.2.12. 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.

8.2.13. 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.

8.2.14. 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.

8.2.15. 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.
 - bears: PolarBear[*]: Tárolja a medvé(ke)t, ha több van akkor is.
- 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.
 - PolarBear CreatePolarBear(): Létrehoz egy medvét. 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 vezénylésére szolgál.
 - void Victory(): Ha vége a játéknak, szól a Controllernek, hogy nyertünk. Külső metódus.

8.2.16. Igloo

- Ezen a jégtáblán iglu áll, a játékosok védve vannak a vihartól. Az ilyen táblán nem csökken a viharban a rajta állók testhője.
- Ősosztályok:
 - Shelter

- Metódusok:
 - void ChillStorm(Tile t): A paraméterként kapott cellán álló játékosok testhője nem csökken, mivel igluban vannak.
 - void BearAttack(Tile t): Így viselkedik a mező ha valaki igluban van és megtámadja a medve.

8.2.17. Item

- Tárgy, a játékos képes ilyeneket felvenni a cellákról. A tárgyak képesek a játékosak 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.

8.2.18. Naked

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

8.2.19. 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.

8.2.20. 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.

8.2.21. 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.
- Ősosztályok:
 - Entity
- 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 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. 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. Rárak 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.

8.2.22. PolarBear

- Jegesmedve osztály. Random lépeget a táblán és ha playert talál megtámadja azt.
- Ősosztályok:
 - Entity
- Metódusok:
 - Step(Directon d): Lép az adott irányba.

8.2.23. 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.

8.2.24. 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.

8.2.25. 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.

8.2.26. 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 lévők közül egyvalaki rákerül a kihúzó játékos cellájára.

8.2.27. 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.

8.2.28. 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.

8.2.29. Sea

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

8.2.30. 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.

8.2.31. 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.

8.2.32. Tent

- Sátor osztály. Le lehet rakni táblára.
- Ősosztályok:
 - Shelter
- Metódusok:
 - void ChillStorm(Tile t): Így viselkedik a tábla, ha sátor van rajta hóviharban.
 - void Break(Tile t): Így viselkedik a tábla, ha eltörik miközben valaki a sátorban van.

8.2.33. TentKit

- Sátor építését lehetővé teszi.
- Interfészek:
 - Item
- Metódusok:
 - void GiveTo(Player p): A játékos így kap sátor alapanyagot.

8.2.34. 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: Entity[*]: Rajta lévő entitások.
 - snow: int: Rajta lévő hómennyiség.
 - weightLimit: int: Rajta lévő játékosok számának maximuma.
- Metódusok:
 - void ChillStorm(): Ezt a metódust a Controller hívja viharban. Hűti a játékosokat, ha nincsenek igluban vagy sátorban.
 - void ChillWater(): Ezt a metódust a Controller hívja körönként. Hűti a játékosokat, ha ez a cella víz.
 - void DecrementSnow(): A hómennyiséget 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): Játékos lelép a celláról. A függvény futása során beállítja a megfelelő adattagokat az új értékekre.

8.2.35. 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.

8.2.36. Osztály1

- Felelősség
[Mi az osztály felelőssége. Kb 1 bekezdés. Ha szükséges, akkor state-chart is.]
- Ősosztályok
*[Mely osztályokból származik (öröklési hierarchia)
Legősebb osztály → Ősosztály2 → Ősosztály3...]*
- Interfészek
[Mely interfészeket valósítja meg.]
- Attribútumok
[Milyen attribútumai vannak]
 - `attribútum1`: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
 - `attribútum2`: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
- Metódusok
[Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról.]
 - `int foo(Osztály3 o1, Osztály4 o2)`: metódus leírása, láthatósága (UML jelöléssel)
 - `int bar(Osztály5 o1)`: metódus leírása, láthatósága (UML jelöléssel)

8.2.37. Osztály2

- Felelősség
[Mi az osztály felelőssége. Kb 1 bekezdés. Ha szükséges, akkor state-chart is.]
- Ősosztályok
*[Mely osztályokból származik (öröklési hierarchia)
Legősebb osztály → Ősosztály2 → Ősosztály3...]*
- Interfészek
[Mely interfészeket valósítja meg.]
- Attribútumok
[Milyen attribútumai vannak]
 - `attribútum1`: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
 - `attribútum2`: attribútum jellemzése: mire való, láthatósága (UML jelöléssel), típusa
- Metódusok
[Milyen publikus, protected és privát metódusokkal rendelkezik. Metódusonként precíz leírás, ha szükséges, activity diagram is a metódusban megvalósítandó algoritmusról.]
 - `int foo(Osztály3 o1, Osztály4 o2)`: metódus leírása, láthatósága (UML jelöléssel)
 - `int bar(Osztály5 o1)`: metódus leírása, láthatósága (UML jelöléssel)

8.3. A tesztek részletes tervei, leírásuk a teszt nyelvén

[A tesztek részletes tervei alatt meg kell adni azokat a bemeneti adatsorozatokot, amelyekkel a program működése ellenőrizhető. Minden bemenő adatsorozathoz definiálni kell, hogy az adatsorozat végrehajtásától a program mely részeinek, funkcióinak ellenőrzését várjuk és konkrétan milyen eredményekre számítunk, ezek az eredmények hogyan vethetők össze a bemenetekkel.]

8.3.1. Teszteset1

- Leírás
[szöveges leírás, kb. 1-5 mondat.]
- Ellenőrzött funkcionalitás, várható hibahelyek
- Bemenet
[a proto bemeneti nyelvén megadva (lásd előző anyag)]
- Elvárt kimenet
[a proto kimeneti nyelvén megadva (lásd előző anyag)]

8.3.2. Teszteset2

- Leírás
[szöveges leírás, kb. 1-5 mondat.]
- Ellenőrzött funkcionalitás, várható hibahelyek
- Bemenet
[a proto bemeneti nyelvén megadva (lásd előző anyag)]
- Elvárt kimenet
[a proto kimeneti nyelvén megadva (lásd előző anyag)]

8.4. A tesztelést támogató programok tervei

[A tesztadatok előállítására, a tesztek eredményeinek kiértékelésére szolgáló segédprogramok részletes terveit kell elkészíteni.]

8.5. Napló

Kezdet	Időtartam	Résztevők	Leírás
2020.04.09. 16:00	1 óra	Glávits	Teszt nyelv fejlesztése
2020.04.09. 19:00	1 óra	Glávits	Parser tervezése
2020.04.10. 18:00	3 óra	Glávits	Parser tervezése