

8. Részletes tervek

8.1 Osztályok és metódusok tervei.

8.1.1 Backpack

- **Felelősség**

Egy karakter táskáját jelképezi. Itt vannak tárolva az adott játékos birtokolt tárgyai. Ezeket lehet kezelni, tehát eltávolítani, hozzáadni, stb.

- **Attribútumok**

- **-obtainedItems<PlayerActions, ArrayList<Items>>**: A birtokolt tárgyak egy HashMapben vannak tárolva. A tárgyak kapnak egy PlayerAction jelzőt, így megkülönböztethetővé válnak.

- **Metódusok**

- **+hasItem(it: PlayerActions): Items**: Visszaadja, hogy a paraméterként kapott PlayerAction-höz (Tárgy jelölő) tartozó tárgyat a táskából, ha nem létezik ilyen akkor null-t ad vissza.
- **+addItem(it: Items, ia: PlayerActions): boolean**: Mivel minden tárgyból 1 darab lehet a táskában, kivéve az essential és az étel, ezért az elején leellenőrzi, hogyha tartalmazza és nem ételről vagy essentialról van szó akkor visszatér egy falseval. Ha erről a két tárgyról van szó akkor azt hozzáadja az adott PlayerActionhöz tartozó HashMap Listájához. (HashMap<PlayerActions, ArrayList<Items>>)
- **+useFood(): Food**: Ha tartalmaz a táskánk ételt, akkor visszaadja a Lista utolsó elemét, és letörli a végéről. Tehát marad egy eggyel kisebb listánk, mert csökkent egy kajával.
- **+getEssentialItemNumber(): int**: Visszaadja a lista méretét, hogy mennyi essentialitem van nála.
- **+removeItem(pa: PlayerActions): void**: Eltávolítja a paraméterként kapott elemet a HashMapből.

8.1.2 Bear

- **Felelősség**

A pályán mozgó jegesmedvét valósítja meg, amivel ha a játékosok egy mezőre kerülnek, akkor megeszi őket.

- **Attribútumok**

- **-IceCell ownCell**: A medve saját mezőjét ebben az attribútumban tároljuk.

- **Metódusok**

- **+move(): void**: A függvény generál egy random számot 1 és 4 között, ami alapján kiválasztja a medve a következő mezejét. Ha a kiválasztott mező nem létezne, vagy

pedig vizes cellát választana ki, akkor a függvény egy másik cellát jelöl ki, amire szintén ellenőrzi ezeket a feltételeket.

8.1.3 Character

- **Felelősség**

Absztrakt osztály, ami a játékban választható karakterek állapotát, tulajdonságait és képességeit összesíti.

- **Attribútumok**

- **-maxBodyHeat: int:** Megadja a karakter max szerezhető testhőjét, és a kezdő testhőjét.
- **-bodyHeat: int:** Megadja a karakter jelenlegi testhőjét, ami kezdetben max. Ha 0-ra csökken akkor meghal.
- **-turnsInWater: int = 0:** Tárolja, hogy a karakter hány kör óta van vízben. Ha ez a szám eléri a játékosok számát (tehát mindenki lelépte a körét), és nincs a karakteren búvárruha, akkor megfullad és vége a játéknak.
- **-wearingDivingSuit: boolean = false:** Megmondja, hogy adott játékos visel-e búvárruhát.
- **-backpack: Backpack:** A játékos hátizsákja, amiben a felvett tárgyait tárolja.
- **-ownCell: IceCell:** Az a mező, ahol az adott játékos éppen áll.
- **-facingWay: Way = up:** Tárolja, hogy az adott karakter éppen milyen irányba néz.
- **-maxActions: int = 4:** A karakter maximum ennyi akciót hajthat végre.
- **-actionsLeft: int:** Karakternek hátralévő használható a akciói a körben, ami ha 0-ra csökken, akkor a következő játékos jön.

- **Metódusok**

- **+ability(): void:** Játékosok képességeinek absztrakt függvénye, amit a leszármazottak implementálnak.
- **+move(): void:** Karakter mozgatása egy mezővel amerre néz, ha található arra mező.
- **+dig(withShovel: boolean): void:** Játékos eltakarít egy vagy kettő hóréteget a saját mezején, a paramétertől függően, amíg van rajta hó.
- **+mine(): void:** Tárgy kibányászása a saját cellán, ha nincs rajta hóréteg és található ott tárgy. Emellett nem vesz bányászunk ki olyan tárgyat, ami nála van, ha az nem Food vagy EssentialItem.
- **+gainOneHeat(): boolean:** Karakterhez ad egy testhőt, ha az nem nőne meg vele a maximum fölé. Igazzal tér vissza, ha hozzáadott egyet.
- **+void loseOneHeat():** Karakter elveszít egy testhőt.
- **+addOneTurnInWater(): void:** Hozzáad egyet a turnsInWaterhez, ami tárolja, hogy a játékos hány köre van vízalatt.
- **+resetTurnsInWater(): void:** turnsInWater visszaállítása 0-ra.
- **+putItemtoBackPack(it: Items, ia: PlayerActions): boolean:** Egy Itemet belerak a Backpackbe a hozzá illő PlayerActionnel együtt, ha a táskában még nincs belőle, kivéve a Food és EssentialItem. Visszatér azzal, hogy sikerült-e belerakni az adott Itemet.
- **+wearDivingSuit(): boolean:** Divingsuit felvétele a wearingDivingSuit beállításával. Visszatér azzal, hogy felvette-e, és nem volt már alapból rajta egy.

- **+useItem(ia: PlayerActions): void:** A megadott kulcsú Item használata. Ha tudta használni az Itemet, akkor elveszít egy munkát vele.
- **+useEssentials(): void:** Jelzőpisztoly összeszerelésének próbálkozása. Ha össze tudták szerelni, akkor elveszít egy munkát, de megnyerik a játékot.
- **+loseOneAction(): void:** Egy felhasználható munka elvesztése, amit a körben használhat.
- **+resetActionLeft(): void:** Felhasználható munkát visszaállítja a maximumra.

8.1.4 Divingsuit

- **Felelősség**

A bűvárruha tárgyat reprezentálja, amit használva az adott játékos nem fullad bele a vízbe.

- **Interfészek**

Items

- **Metódusok**

- **+use(actualch: Character): void:** A játékos egy munkát felhasználva felveszi a bűvárruhát, ha még nincsen rajta. Ha sikeresen felvette, akkor a játékos veszít egy munkát. Ha a játékos úgy akarná felvenni a bűvárruhát, hogy már van rajta, akkor nem vonódik le emiatt munka.
- **+equip(ch: Character): boolean:** Berakja a bűvárruhát a táskába, amennyiben az még nem szerepel a karakternél. A visszatérési értékből derül ki, hogy a hátizsákba rakás sikeres volt-e.

8.1.5 Eskimo

- **Felelősség**

Az eskimo karaktert jelképezi. A két játszható karakter közül az egyik.

- **Ósosztályok**

Character

- **Metódusok**

- **+ability(): void:** Beállítja a saját cellájára az iglut, ha már létezik rajta iglu akkor nem von le munka használatot.

8.1.6 EssentialItem

- **Felelősség**

A jelzőrakéta darabjait reprezentálja, amiket a játékosoknak össze kell gyűjteni, hogy megnyerjék a játékot.

- **Interfészek**

Items

- **Attribútumok**

- **-int ID:** Ez az attribútum tárolja, hogy a jelzőrakéta melyik darabját reprezentálja az adott objektum. Három darabot kell összeszedni, ennek megfelelően az értéke 0-2 között lehet.

- **Metódusok**

- **+use(actualch: Character): void:** A függvény akkor hívódik meg, ha a játékosok kezdeményezik a pisztoly összerakását. Hozzáadja a wincheckerhez az adott karakternél lévő alkatrészek számát.
- **+equip(ch: Character): boolean:** Berakja az adott jelzőpisztoly alkatrészt a paraméterként kapott karakter hátizsákjába. A karaktereknél bármennyi alkatrész lehet, ezért ez a függvény mindig true-val fog visszatérni.

8.1.7 Explorer

- **Felelősség**

A felfedező karaktert jelképezi. A két játszható karakter közül az egyik.

- **Össosztályok**

Character

- **Metódusok**

- **+ability(): void:** Az adott karakter nézési irányába lévő cellára beállítja az ismeretséget. Ha az adott cella létezik és még nincs beállítva rá az ismeretség akkor levon egy munkát.

8.1.8 Food

- **Felelősség**

Az élelem tárgyat reprezentálja, amit használva az adott játékos növelni tudja a testhőmérsékletét

- **Interfészek**

Items

- **Metódusok**

- **+use(actualch: Character): void:** A játékos egy munkát felhasználva növeli a testhőmérsékletet és csökkenti a nála lévő étel mennyiséget eggyel. Ha a testhőmérséklete maximális, akkor nem von le munkát érte és az étel mennyiségét se csökkenti.
- **+equip(ch: Character): boolean:** Berakja az ételt a táskába.

8.1.9 FragileShovel

- **Felelősség**

A törékeny ásó tárgyat reprezentálja, amit használva az adott játékos el tudja takarítani az alatta lévő hóréteget, amíg az ásó el nem törik.

- **Interfészek**

Items

- **Attribútumok**

- **-durabilityMax: int = 3:** Megmondja, hogy az ásót hányszor lehet maximum használni, míg el nem törik.
- **-durability: int = durabilityMax:** Megadja, hogy még hányszor lehet használni az ásót, míg el nem törik.

- **Metódusok**

- **+use(actualch: Character): void:** A játékos egy munkát felhasználva a saját celláján ás egyet, ha van rajta hóréteg. Ha sikerült ásnia, akkor veszít egyet a durability-ből, ami ha 0-ra csökken, akkor az ásó törlődik a karakter táskájából, és nem lehet már használni.
- **+equip(ch: Character): boolean:** Berakja a törékeny ásót a táskába, amennyiben az még nem szerepel a karakternél. A visszatérési értékből derül ki, hogy a hátizsákba rakás sikeres volt-e.

8.1.10 IceField

- **Felelősség**

Ez az osztály képviseli az egész táblát, tárolja a mezőket, karaktereket és a játék menetéhez kapcsolódó fontosabb információkat. (játék végkimenetele, játékosszám, aktuális játékos)

- **Attribútumok**

- **-maxPlayer: int:** 3-6 és közötti értéket vehet fel, a játékosok számát jelöli.
- **-fieldLengths: int = maxPlayer + 4:** Pálya mérete, mivel négyzet alakú és a játékosok számától függ. Pl: 3 játékosnál a pálya mérete 7*7.
- **-currentPlayer: int:** Aktuális játékos sorszámát jelöli.
- **-gameWon: boolean = false:** A játék sikeres kimenetele esetén az értéke true-ra változik, alaphelyzetben false.
- **-gameLost: boolean = false:** A játék sikertelen kimenetele esetén az értéke true-ra változik, alaphelyzetben false.
- **-chosenToSave: int = -1:** Amikor egy kötéllel kimentünk egy embert, akkor ez az érték fog átváltozni a kimentendő karakter sorszámára.
- **-bear: Bear:** A medvét jelképezi.
- **-wc: WinChecker:** Ellenőrzi a nyerési feltételt, amennyiben egy játékos kezdeményezi a pisztoly összeszerelését.
- **-characters[3..maxPlayer]: Character:** A karakterek listája. aminek maximális mérete 6 lehet.
- **-field[7..10][7..10]: IceCell:** A pályát jelöli ami IceCellegekből épül fel. Lehet 7*7, 8*8, 9*9 és 10*10-es a pálya.

- **Metódusok**

- **+nextPlayer(): void:** Az elején, visszarakja a még aktuális játékos munkáját 4 re, és mozgatja a medvét. Aztán meghívja a hóvihart 20% os eséllyel. Majd minden vízben lévő játékos vízben töltött körét növeli eggyel és minden játékosra megnézi, hogy a testhője 0-e vagy a vízben töltött ideje búvárruha nélkül meghaladja-e a maximálisat, mert akkor vége a játéknak. Majd törli az összes sátozt a pályáról. Végül leellenőrzi, hogy mindenki vízben van-e és ha nem akkor növeli az aktuális játékos számát eggyel.
- **-snowStorm(): void:** Egy random középponttal a pályán, ezután választ egy véletlenszerű sugarat, 2 és a pálya szélessége felfele kerekítve között. Ezen a területen növeli a hóréteget eggyel, vagy ha van rajta igloo akkor azt törli.
- **+movePlayer(w: Way): void:** Az adott irányba beállítja az aktuális játékos nézési irányát. Majd meghívja a karakteren a move() függvényt.
- **+usePlayerItem(ia: PlayerActions): void:** Ha a paraméter egy essentialItem, meghívja a privét useEssentialItems() függvényt. Egyébként a useItem() függvényt az aktuális karakteren.
- **+useAbility(): void:** Meghívja a képesség használatát az aktuális játékosra.
- **-useEssentialItems(): void:** Először megnézi, hogy egy cellán állnak-e, ha nem akkor visszatér simán. Ha egy cellán vannak összeadja az összes karakter essentialItemjét, ha megvan mind akkor nyertek.

- **+addIceCell(ic: IceCell, removed: IceCell): void:** Ha tartalmazza a pálya az adott törlendő cellát akkor átalakítja azzá amit első paraméterként megkap.
- **+mineActualCell(): void:** Meghívja az adott karakter mine() függvényét.
- **+setPlayerWay(w: Way): void:** Beállítja az adott játékos nézési irányát.
- **+gameLost(): void:** A gameLost attribútumot truevá rakja.
- **+gameWon(): void:** A gameWon attribútumot truevá rakja.
- **+actionHandler(): void:** Ha 0 munkája van hátra vagy több mint 0 kör óta vízben van akkor meghívja a nextPlayert. Ez a függvény minden munkát végző IceFieldben lévő függvény végén hívódik meg.
- **+setChosenToSave(i: int): void:** Ha nagyobb az i mint 0 és kisebb mint maxPlayer akkor beállítja a chosenToSave értékét i re.
- **+getChosenToSave(): Character:** Ha az alapértékkel egyenlő (-1) akkor null-t ad vissza, egyébként az adott indexű játékost. A végén visszarakja az attribútum értékét -1 re.

8.1.11 IceCell

- **Felelősség**

A játéktábla egy celláját reprezentálja. Tárolja a szomszédait, a hórétteget, a rajta álló karaktereket, az IceField objektumot, amin a játék épp játszódik.

- **Attribútumok**

- **#int capacity:** Az adott cella kapacitása. Ez jelzi, hogy maximálisan mennyi ember léphet rá, mielőtt beszakadna.
- **-boolean capacityKnown = false:** Megmondja, hogy a játékosok tudják-e ennek a cellának a kapacitását.
- **#int snow:** Az aktuálisan cellán lévő hóréttek száma.
- **#int maxSnow = 5:** A maximális hórétteg, ami egy cellán szerepelhet.
- **#Bear bear = null:** Ha a mezőn rajta áll a medve, akkor itt tárolódik el, egyébként értéke null.
- **-neighbours<Way, IceCell>:** A cellával szomszédos mezőket tároljuk itt irányuk szerint.
- **#Character standingPlayers[0..*]:** A mezőn éppen álló karakter objektumok.
- **#IceField ownField:** A játéktábla, ahol a játék épp játszódik.

- **Metódusok**

- **+loseSnow(withShovel: boolean): boolean:** Eltávolít hórétteget az adott celláról. Ha a bool paraméter false, akkor kézzel takarít a játékos, tehát egy réteg tűnik el, ha pedig true, akkor valamilyen ásóval, tehát két réteg hó tűnik el.
- **+gainOneSnow(): void:** Megnöveli a cella hóréttegét eggyel. Ha ezzel együtt a hórétteg öt fölé lenne, akkor marad a cellán az öt réteg hó.
- **+getNeighbour(w: Way): IceCell:** Visszaadja a cella adott irányú szomszédját. Ha az nem létezik, akkor null-t ad vissza.
- **+addNeighbour(w: Way, ic: IceCell): void:** Hozzáadja a paraméterként kapott cellát a szomszédok közé. Annak irányát a w paraméter határozza meg.
- **+removeCharacter(ch: Character): void:** Eltávolítja az adott karaktert a standingPlayers tömbből.

- **+addCharacter(ch: Character): void:** Hozzáadja az adott karaktert a `standingPlayers` tömbhöz.
- **+hasBear(): boolean:** Megmondja, hogy az adott cellán áll-e medve.
- **+removeBear(): void:** A medvét eltávolítja az adott cella tagváltozójából.

8.1.12 Items

- **Felelősség**

Olyan Interface, amit az összes a táblán található tárgy implementál.

- **Metódusok**

- **+use(actualch: Character): void:** Az implementált tárgyat használja az a karakter, amit paraméterül kap.
- **+equip(ch: Character): boolean:** Tárgy hozzáadása a paraméterül kapott karakter táskájához. Ha nem tudta hozzáadni, akkor hamissal tér vissza.

8.1.13 PlayerActions

- **Felelősség**

Játékosok által végezhető cselekvések.

- **Attribútumok**

- **+shoveling**
- **+savingWithRope**
- **+eating**
- **+wearingSuit**
- **+assemblingEssentials**
- **+setUpTent**
- **+fragileshoveling**

8.1.14 Rope

- **Felelősség**

A kötel tárgyat reprezentálja, amit használva a játékos ki tud húzni egy választott játékost a mellette álló vízcellából.

- **Interfészek**

Items

- **Metódusok**

- **+use(actualch: Character): void:** A karakter megpróbál kimenteni egy választott játékost egy mellette lévő vízcellából. Ha sikeres volt, akkor azt a játékost a saját cellájára lépteti, és veszít egy munkát.

- **+equip(ch: Character): boolean:** Berakja a kötelet a táskába, amennyiben az még nem szerepel a karakternél. A visszatérési értékből derül ki, hogy a hátizsákba rakás sikeres volt-e.

8.1.15 Shovel

- **Felelősség**

Az ásó tárgyat reprezentálja, amit használva az adott játékos el tudja takarítani az alatta lévő hóréteget.

- **Interfészek**

Items

- **Metódusok**

- **+use(actualch: Character): void:** A játékos egy munkát felhasználva a saját celláján és egyet, ha van rajta hóréteg. Ha sikerült ásnia, akkor a játékosnak csökken eggyel a használható akciók száma.
- **+equip(ch: Character): boolean:** Berakja az ásót a táskába, amennyiben az még nem szerepel a karakternél. A visszatérési értékből derül ki, hogy a hátizsákba rakás sikeres volt-e.

8.1.16 StableIceCell

- **Felelősség**

Olyan jégmező, amin bármennyi karakter állhat, és lehetnek benne eltemetve tárgyak, amit kiáshatnak.

- **Ősosztályok**

IceCell

- **Attribútumok**

- **hasIgloo: boolean = false:** Megadja, hogy a cellán van-e építve iglu
- **tentTurnsLeft: int = 0:** Megadja, hogy a cellán található-e sátor, és hogy hány kör múlva tűnik el. Ha 0, akkor nincs rajta sátor.
- **-item: Items:** Cellába fagyott tárgy, amit ki tudnak ásni, ha nincs felette hóréteg.

- **Metódusok**

- **-removeItem(): void:** Eltávolítja a mezőn található tárgyat.
- **+mine(actual: Character):** A cellán található tárgy odaadása a paraméterül adott karakternek, ha nincs felette hó.
- **+setIgloo(b: boolean): boolean:** Iglu építése, visszaadja, hogy kellett-e építeni, és nem volt rajta már egy.
- **+snowing(): void:** Egy hóréteg hozzáadása, és ha nincs rajta iglu vagy sátor, akkor a rajta álló játékosok elvesztenek egy testhőt.
- **+accept(ch: Character): void:** Paraméterül kapott játékos befogadása a cellára. Ha volt rajta medve, akkor vége a játéknak.

- **+setUpTent(): boolean:** Sátor telepítése a cellán. Hamissal tér vissza, ha már ugyanez a játékos rakott már ebben a körben le sátrat.
- **+acceptBear(b: Bear): boolean:** Paraméterül kapott medve befogadása a cellára. Ha voltak a cellán játékosok, akkor vége a játéknak. StableIceCellnél mindig igazsal tér vissza.
- **+loseOneTentTurn(): void:** Rajta lévő sátor hátralévő köreinek csökkentése eggyel.
- **+resetTentTurnsLeft(): void:** Rajta lévő sátor hátra lévő köreinek számát visszaállítja 0-ra.

8.1.17 Tent

- **Felelősség**

Játékos lerakhatja a saját mezejére, ahol megvédi őket egy teljes körig a hóvihartól.

- **Interfészek**

Items

- **Metódusok**

- **+use(actualch: Character): void:** A játékos egy munkát felhasználva a saját celláján felépíthet egy sátrat, ha még nem rakott le egyet ugyanebben a körben. A sátor megvédi a rajta álló karaktereket a hóvihartól, de a medvétől nem
- **+equip(ch: Character): boolean:** Berakja a sátrat a táskába, amennyiben az még nem szerepel a karakternél. A visszatérési értékből derül ki, hogy a hátizsákba rakás sikeres volt-e.

8.1.18 UnstableIceCell

- **Felelősség**

Instabil típusú cellát valósít meg. Ha ilyen típusú cellára a kapacitásánál több karakter lép, akkor beszakad, és a rajta lévő karakterek vízbe kerülnek.

- **Ősosztályok**

IceCell

- **Attribútumok**

- **hasIgloo: boolean = false:** Megadja, hogy a cellán van-e építve iglu
- **tentTurnsLeft: int = 0:** Megadja, hogy a cellán található-e sátor, és hogy hány kör múlva tűnik el. Ha 0, akkor nincs rajta sátor.

- **Metódusok**

- **+setIgloo(b: boolean): boolean:** Iglu építése, visszaadja, hogy kellett-e építeni, és nem volt rajta már egy.
- **+snowing(): void:** Egy hóréteg hozzáadása, és ha nincs rajta iglu vagy sátor, akkor a rajta álló játékosok elvesztenek egy testhőt.
- **+setUpTent(): boolean:** Sátor telepítése a cellán. Hamissal tér vissza, ha már ugyanez a játékos rakott már ebben a körben le sátrat.

- **+acceptBear(b: Bear): boolean:** Paraméterül kapott medve befogadása a cellára. Ha voltak a cellán játékosok, akkor vége a játéknak. StableIceCellnél mindig igazzal tér vissza.
- **+loseOneTentTurn(): void:** Rajta lévő sátor hátralévő köreinek csökkentése eggyel.
- **+resetTentTurnsLeft(): void:** Rajta lévő sátor hátra lévő köreinek számát visszaállítja 0-ra.
- **+accept(ch: Character): void:** Paraméterül kapott játékos befogadása a cellára. Ha volt rajta medve, akkor vége a játéknak.

8.1.19 WaterCell

- **Felelősség**

Olyan cella, amiről a játékosok nem tudnak elmozdulni, és ha nem viselnek bűvárruhát, akkor egy körön belül meghalnak.

- **Ősosztályok**

IceCell

- **Attribútumok**

- **-broken: boolean = false:** Tárolja, hogy volt-e olyan karakter, aki már beleesett ebbe a vízcellába. Ha ez igaz, akkor nem tud hóréteg lenni a cellán ezután.

- **Metódusok**

- **+snowing(): void:** Ha még nincs betörve, akkor egy hóréteg hozzáadása.
- **+movePlayerOut(from: Way): boolean:** Ha megtalálható a cellán az a játékos, akit az IceField a chosenToSave-ben tárol, akkor őt kilépteti a paraméterül adott irányba. Ha nem található, akkor hamissal tér vissza.
- **+accept(ch: Character): void:** Paraméterül kapott karakter befogadása, és a broken igazzá állítása.

8.1.20 Way

- **Felelősség**

A lehetséges irányok a játékban.

- **Attribútumok**

- **+up:**
- **+right:**
- **+down**
- **+left**

- **Metódusok**

- **+opposite(): Way:** Visszaadja az ellenkező irányt annak, amin meghívták.
- **rotate(toRight: boolean): Way:** Elforgatja 90 fokkal az irányt amin meghívták, a paraméter szerinti irányba.

8.1.21 WinChecker

- **Felelősség**

Ez az osztály ellenőrzi, hogy a játékosok össze tudják-e szerelni a jelzőpisztolyt, ezzel megnyerve a játékot.

- **Attribútumok**

- **-int assemblingEssentials = 0:** Számláló, ami azt tartja nyilván, hogy az egy mezőn álló karaktereknél (azon karakterek, akik az összerakást kezdeményező játékoskal egy mezőn állnak) összesen hány EssentialItem van.
- **-int maxEssentials = 3:** Statikus attribútum, ami az összegyűjtendő tárgyak mennyiségét tárolja.

- **Metódusok**

- **+addEssentials(i: int): void:** Megnöveli az assemblingEssentials attribútum értékét a kapott i paraméterrel.
- **+resetAssembledItems(): void** Visszaállítja nullára az assemblingEssentials számláló értékét.
- **+isAssembled(): boolean:** Visszaadja az assemblingEssentials paraméter alapján, hogy sikerült-e összeszerelni a jelzőpisztolyt.

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

8.2.1 Kötél használata

- **Leírás**

A kötélt használatának tesztelése. Az első játékoskal rálépünk egy stabil cellára, passzolunk. A második játékoskal vizes cellába lépünk, a harmadik játékos fogja használni a kötelet. Először annak a játékosnak a számát adja meg, aki a stabil cellára lépett, bemutatva, hogy rajta nem működik a kötélt. Másodjára már a helyes sorszámot adjuk meg, aminek hatására a vízben lévő játékos rákerül az őt kimentő játékos cellájára.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A kötélt használatának helyessége: a kimentő játékos kiválaszthatja, hogy melyik játékoskal szeretné megmenteni, ill. a kimentett játékosnak ugyanarra a mezőre kell érkeznie, ahonnan ki lett mentve. Ha olyan játékos számát adná meg, akit nem tud kimenteni, akkor nem történik semmi.

- **Bemenet**

-moveright
-pass
-movedown
-save
-0
-save

-1

- **Elvárt kimenet**

Típus 0:St 1:Víz

2:Inst 3:Item	Karakterek	Kapacitás
0 0 0 0 0 0 0	2 1 0 0 0 0 0	3 3 3 3 3 3 3
1 0 0 0 0 0 0	0 0 0 0 0 0 0	0 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3

A képen látszódik a vízes cella helye, ugyanis annak kapacitása nulla. És látszódik, hogy a végén sikeresen kimentette a vízbe esett játékost. (2 en állnak egy helyen)

8.2.2 Instabil beborulása

- **Leírás**

A teszt az instabil cella beszakadását szimulálja. Ilyenkor az instabil cellát kivesszük a pálya tömbből, hozzáadunk egy vízeset, aminek beállítjuk a szomszédait, és rárakjuk a vízbe került karaktereket.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Egyrészt ellenőrizzük, hogy valóban átvette-e az instabil cella helyét egy vízes a megfelelő helyen. Másrészt ebben a tesztben az összes létrehozott karakter beesik a vízbe, ezért a játékosoknak automatikusan el kell veszteniük a játékot.

- **Bemenet**

-movedown

-pass

-movedown

-pass

-movedown

- **Elvárt kimenet**

Típus 0:St 1:Víz

2:Inst 3:Item	Karakterek	Kapacitás
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
2 0 0 0 0 0 0	3 0 0 0 0 0 0	0 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3

Vesztettetek!

8.2.3 Jelzőpisztoly összeszerelése

- **Leírás**

A játékosok kibányásszák és összeszerelik a három alkatrészt, ezáltal megnyerik a játékot. Először úgy próbálják meg összeszerelni, hogy nem áll egy mezőn a három darabot birtokló játékos, ezért nem fog történni semmi, és munka se vonódik le. Másodjára már egy mezőn állnak, ezért egy munka árán összeszerelik a pisztolyt és megnyerik a játékot.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Abban az esetben, amikor a pisztolydarabokat birtokló játékosok nem állnak egy helyen, akkor munkának nem szabad csökkenni, és nem nyerik meg a játékot. A második esetben az összeszerelés sikeres, egy munkába kerül, és a játék véget ér.

- **Bemenet**

```
-movedown
-mine
-pass
-movedown
-moveright
-mine
-pass
-movedown
-moveright
-moveright
-mine
-moveright
-pass
-pass
-assemble
-moveleft
-assemble
```

- **Elvárt kimenet**

```
Típus 0:St 1:Víz
2:Inst 3:Item      Karakterek
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
3 3 3 0 0 0 0 0    0 3 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0

Nyertetek!
```

8.2.4 Hóvihar miatt játék vége

- **Leírás**

Egy játékosnak lecsökken 0-ra a testhője a folyamatos hóviharok miatt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizzük, hogy a játék véget ér-e, ha valamelyik játékos testhője nullára esik. A pass bemenet azért szükséges, mert a játék végét ellenőrző függvények azután hívódnak, miután egy játékos befejezte a körét.

- **Bemenet**

-snow

-snow

-snow

-snow

-pass

- **Elvárt kimenet**

2. játékos hátralévő munkája: 4, testhője: 0 és v

Ásó:0 Kötél:0 Ruha:0 Étel:0 Plusz:0 Törékeny

Típus 0:St 1:Víz

2:Inst	3:Item	Karakterek	Kapacitás
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3
0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3

Vesztettetek!

8.2.5 Játékos vízbe fullad

- **Leírás**

Egy játékos vízbe esett, és a többi játékos nem menti őt ki.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Miután egy játékos beleesett a vízbe, és nem mentik ki őt csapattársai egy teljes körön belül, a játékot elvesztették a játékosok.

- **Bemenet**

-movedown

-pass

-pass

- **Elvárt kimenet**


```

Típus 0:St 1:Víz
2:Inst 3:Item      Karakterek
0 0 0 0 0 0 0 0    2 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0    1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0

Vesztettetek!

```

8.2.6 Medve megeszik egy játékost

- **Leírás**
A medve egy mezőre kerül egy játékosal, aki ennek következtében meghal, és a játékot elvesztik.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Miután a medve megeszi a játékost, a játéknak véget kell érnie.
- **Bemenet**
 - bearup
 - bearup
 - bearup
 - bearleft
 - bearleft
 - bearleft

- **Elvárt kimenet**

```

Típus 0:St 1:Víz
2:Inst 3:Item      Karakterek
0 0 0 0 0 0 0 0    3 B 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0    0 0 0 0 0 0 0 0

Vesztettetek!

```

8.2.7 Tárgy kiásása

- **Leírás**
Kiásunk egy tárgyat a stabil cellából.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A játékosok csak akkor bányászhatnak ki egy tárgyat a cellából, ha a hórétteg nulla, ilyenkor ez egy munkájába kerül. Amennyiben úgy akarna bányászni, hogy a hórétteg nem nulla, akkor nem történik semmi, és munka sem vonódik le.

- **Bemenet**
-movedown
-mine
-dig
-dig
-mine
-pass
-pass

- **Elvárt kimenet**

```

1. játékos hátralévő munkája: 4,
Ásó:1 Kötél:1 Ruha:0 Étel:0
Típus 0:St 1:Viz
2:Inst 3:Item      Karakterek
0 0 0 0 0 0 0      2 0 0 0 0 0 0
3 0 0 0 0 0 0      1 0 0 0 0 0 0
0 0 0 0 0 0 0      0 0 0 0 0 0 0
0 0 0 0 0 0 0      0 0 0 0 0 0 0
0 0 0 0 0 0 0      0 0 0 0 0 0 0
0 0 0 0 0 0 0      0 0 0 0 0 0 0
0 0 0 0 0 0 0      0 0 0 0 0 0 0
0 0 0 0 0 0 0      0 0 0 0 0 0 0

```

8.2.8 Iglu és sátor reagál a hóviharra

- **Leírás**
Az iglu és sátor viselkedése hóviharban.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Hóvihar lefolyása után az érintett iglu és sátor eltűnik a pályáról.
- **Bemenet**
-pass
-pass
-ability
-moveright
-setuptent
-snow

- **Elvárt kimenet**

Karakterek	Kapacitás	Hórétég	Kap. Known	Iglu	Sátor
2 1 0 0 0 0 0	3 3 3 3 3 3 3	1 1 1 1 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	3 3 3 3 3 3 3	1 1 1 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	3 3 3 3 3 3 3	1 1 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	3 3 3 3 3 3 3	1 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0

8.2.9 Búvárruha működése

- **Leírás**
Búvárruha helyes működésének ellenőrzése.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A búvárruhát viselő játékosok nem halnak meg soha vízbe esés után. A lenti kimenet azért felel meg a helyes kimenetnek, mert a passzok száma meghaladja a játékosét, és még mindig nem írta ki a program, hogy meghaltak volna.
- **Bemenet**
-wearsuit
-movedown
-pass
-pass
-pass
-pass
- **Elvárt kimenet**

```
Típus 0:St 1:Víz
2:Inst 3:Item   Karakterek
0 0 0 0 0 0 0   2 0 0 0 0 0 0
1 0 0 0 0 0 0   1 0 0 0 0 0 0
0 0 0 0 0 0 0   0 0 0 0 0 0 0
0 0 0 0 0 0 0   0 0 0 0 0 0 0
0 0 0 0 0 0 0   0 0 0 0 0 0 0
0 0 0 0 0 0 0   0 0 0 0 0 0 0
0 0 0 0 0 0 0   0 0 0 0 0 0 0
```

8.2.10 “Hibás” munkavégzés

- **Leírás**
Leellenőrizzük az összes olyan munkavégzést, amit lehet “hibásan” végezni. Ez alatt azt értjük, hogy úgy végzi el az adott tevékenységet, ahogy annak nincs értelme a játék szabályai alapján, ennek következtében munkát sem veszít (olyan irányba akar mozogni, amerre már nincs mező, hóréteggel borított mezőn akar bányászni, stb.).
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Az ilyen “hibás” munkavégzéseknek nincs eredménye, és munkát sem veszítenek a karakterek miattuk. A tesztelt munkavégzések:
 - karakter ki akar menni a pálya széléről
 - úgy akar tárgyat bányászni, hogy még van a cellán hórétég
 - úgy akar hórétéget eltüntetni, hogy már nincs rajta semennyi

- olyan tárgyat akar kibányászni, ami már van nála, és amiből nem lehet több egy karakternél
- egy játékos a képességét akarja használni, irányként pedig egy pályán kívüli mezőt ad meg
- egy olyan mezőre akar képességet használni, aminek nem lenne hatása (pl. felfedező újra meg akarja tudni egy már ismert kapacitású mező kapacitását)
- **Bemenet**
 - moveup
 - movedown
 - mine
 - dig
 - dig
 - mine
 - moveright
 - pass
 - pass
 - mine
 - moveleft
 - setwayleft
 - ability
 - setwayup
 - ability
 - ability

- **Elvárt kimenet**

1. játékos hátralévő munkája: 2, testhője: 4 és vízben töltött köre: 0

Ásó:1 Kötél:1 Ruha:0 Étel:0 Plusz:0 Törékeny ásó:0 Sátor:0

Típus 0:St 1:Víz

2:Inst 3:Item	Karakterek	Kapacitás	Hórégteg	Kap. Known	Iglu	Sátor
0 0 0 0 0 0 0	2 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	1 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
3 3 0 0 0 0 0	1 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0

8.2.11 Hóesés hatása vizes cellákra

- **Leírás**
A hóesés máshogy viselkedik attól függően, hogy a vizes mezőn van-e hóréteg.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Az olyan vizes cellákra, amelyeken még található hóréteg, a hóvihar ráhelyez plusz egyet. Amennyiben a vizes mezőn már nincs hóréteg (tehát már egyszer beleesett egy játékos), akkor nem kerül új hóréteg a cellára.
- **Bemenet**
 - snow
 - movedown
 - snow
- **Elvárt kimenet**

Típus 0:St 1:Víz

2:Inst 3:Item	Karakterek	Kapacitás	Hórétteg
0 0 0 0 0 0 0	2 0 0 0 0 0 0	3 3 3 3 3 3 3	2 2 2 2 0 0 0
1 0 0 0 0 0 0	1 0 0 0 0 0 0	0 3 3 3 3 3 3	0 2 2 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	2 2 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	2 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0

8.2.12 Iglu és a sátor a medvére reagál

- **Leírás**
Az iglu megvéd a medvétől, míg a sátor nem.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ha a medve olyan mezőre lépne, ahol az adott játékos egy sátorban tartózkodik, akkor ugyanúgy meghal a játékos, mintha nem lenne ott semmi. Ezzel ellentétben ha a játékos iglut tartalmazó mezőn találkozik a medvével, akkor a medve egyszerűen továbbmegy.
- **Bemenet**
 - pass
 - pass
 - setuptent
 - moveright
 - ability
 - bearup
 - bearup
 - bearup
 - bearleft
 - bearleft
 - bearleft
- **Elvárt kimenet**

Típus 0:St 1:Víz

2:Inst 3:Item	Karakterek	Kapacitás	Hórétteg	Kap. Known	Iglu	Sátor
0 0 0 0 0 0 0	2 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 1 0 0 0 0 0	3 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0
0 0 0 0 0 0 0	0 0 0 0 0 0 0	3 3 3 3 3 3 3	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0	0 0 0 0 0 0 0

Vesztettetek!

A medve mozgásának kirajzolását csak egy körrel lemaradva tudtuk megoldani. Szóval ami a képen látszódik az még az előző körben lévő medve állapota, viszont a tényleges állapota a bal felső sarok, ami a sátor helye. Szóval az iglun sikeresen továbbment.

8.2.13 Goldschmidt teszt

1:

- **Leírás**
Mozog a medve és megeszik valakit, a játékosok csak fel-le mászkálnak.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Miután a medve megeszi a játékost, a játéknak véget kell érnie.
- **Bemenet**
-moveleft
-beardown

Az alap down mozgás lett a downright és létezik downleft irány is ami a lenti irány bal oldalát jelöli. Ugyanez létezik az up mozgás esetében is. Ott is az alap up a felfele jobb iránya.

- **Elvárt kimenet**
Vesztettetek!

(mivel nem írjuk meg a programot mint a saját teszteseteinknél ezért nem tudunk olyan fajta képet bemásolni, de ugyanúgy fel lesz építve a pálya és látszódik, hogy azonos cellán vannak, ez vonatkozik a többi megadott tesztesetre is)

2:

- **Leírás**
Játékosok törékeny táblára mennek és az eltörrik.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizzük, hogy valóban átvette-e az instabil cella helyét egy vizes a megfelelő helyen. Másrészt ebben a tesztben az összes létrehozott karakter beesik a vízbe, ezért a játékosoknak automatikusan el kell veszteniük a játékot.
- **Bemenet**
-movedown
-pass
-pass
-moveright
- **Elvárt kimenet**

Vesztettetek!

3:

- **Leírás**
Eszkimó iglut épít.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Az eszkimó saját mezején meg kell jelennie egy iglunak.
- **Bemenet**
-ability
- **Elvárt kimenet**
Az iglu tömbjében meg kell jelennie az adott cellán egy 1 esnek.

8.2.14 Egyéni teszt

- **Leírás**
Ennél a tesztnél el lehet készíteni egy saját pályát amin lehet tesztek végrehajtani.

Lehet cellákat lerakni, itt a módosítási lehetőségek, hogy milyen fajta cellát rakjon le, és melyik x és y koordinátára rakja le. Az instabilnál lehet a kapacitásán választani. A stabil cellába lehet berakni itemet. A celláknak lehet növelni a hórétégét (alapértelmezetten nulla)

Lehet karaktereket lerakni. Először megkérdezi, hogy a 3 karaktert melyik cellára akarod lerakni. Ezután ki lehet választani a 3 karakter típusát. A karakterekhez, lehet hozzáadni több tárgyat is.

Lehet medvét is hozzáadni. Itt csak a helyzetét lehet kiválasztani.

8.3 Napló

Kezdet	Időtartam	Résztevők	Leírás
2020.04.10 18:00	4 óra	Nagy Gellért	A dokumentáció ⅓ -ének elkészítése.
2020.04.10 18:00	4 óra	Nagy Viktor	A dokumentáció ⅓ -ének elkészítése.
2020.04.10 18:00	4 óra	Markovics Gergely	A dokumentáció ⅓ -ének elkészítése.
2020.04.10 18:00	4 óra	Vermes Balázs	A dokumentáció ⅓ -ének elkészítése.
2020.04.10 18:00	4 óra	Zselinszky Norman	A dokumentáció ⅓ -ének elkészítése.