

8. Részletes tervek

48 – asd

Konzulens:

Ludmány Balázs

Csapattagok

Bödőcs Richárd	FG7VG1	bodocsr@gmail.com
Halász Bence Botond	Q7RW4E	worldofsteves@gmail.com
Hámori Péter Sándor	SGLJSB	hampet97@gmail.com
Lauber Anna	H9GBKF	lauber.anna@gmail.com
<u>Pintér Tamás</u>	JY4D5L	pintertamas99@gmail.com

2021.03.31.

7. Prototípus koncepciója - javítás

7.2.2. Bemeneti nyelv

A teszteléshez használt bemeneti nyelv koncepciója változatlan. A könnyebb értelmezhetőség és átláthatóság szempontjából két technikai módosítást eszközölünk.

Egyrészt, a bemeneti értékek nem számok, hanem szöveges információk lesznek, amelyek az emberek számára könnyebben értelmezhetőek, dekódolhatóak. Ez nagyban segíti a tesztesetekhez szükséges bemenetek elkészítését, valamint az ellenőrzésüket is.

Másrészt, a logikailag egy utasításhoz tartozó bemeneteket egy sorban adjuk meg (nem több egymás utáni sorban), amely szintén az átláthatóságot növeli.

Az új formátumban így néz ki egy példa bemenet a játék inicializálására:

```
number of players
number of asteroids
numberOfLayers0;material0;nearSunCount0;nearSun0
numberOfLayers1;material1;nearSunCount1;nearSun1
numberOfLayers2;material2;nearSunCount2;nearSun2
numberOfLayers3;material3;nearSunCount3;nearSun3
numberOfLayers4;material4;nearSunCount4;nearSun4
number of robots
number of ufos
number of pairs to be neighbors
asteroid0;asteroid1
asteroid0;asteroid2
asteroid1;asteroid2
asteroid1;asteroid3
asteroid2;asteroid3
asteroid2;asteroid4
asteroid3;asteroid4
number of portal pairs
portal0Asteroid;portal1Asteroid
robot0Asteroid
robot1Asteroid
ufo0Asteroid
player0Asteroid
player0uranium; players0ice; player0coal; player0iron; player0portals
player1Asteroid
player1uranium; players1ice; player1coal; player1iron; player1portals
player2Asteroid
player2uranium; players1ice; player2coal; player2iron; player2portals
```

A játéknak egy valid állapotát tartalmazza a következő példa (a példa a fenti sémának egy konkrét kitöltése):

```
4
5
3;ice;0;notNearSun
2;uranium;2;nearSun
5;iron;3;notNearSun
0;uranium;0;nearSun
1;coal;1;nearSun
0
0
7
1;2
1;3
2;3
2;4
3;4
3;5
4;5
2
0;0
1;4
2
2;2;2;1
4
3;0;0;1
0
0;1;2;0
1
1;3;0;0
```

Az új formátumban egy példa bemenet a játék menetének vezérlésére:

```
solarStorm;centralAsteroid
player1activity1
player2activity1;player2activity1additionalInfo
player3activity1
solarStorm
player1activity2
player2activity2
player3activity2;player3activity2additionalInfo
solarStorm;centralAsteroid
player1activity3
player2activity3;player2activity3additionalInfo
player3activity3
```

7.2.3. Kimeneti nyelv

A nyelv kimenete az adott szabály alapján generálódik:

Egy bizonyos "akció" végrehajtásának sikeressége: `successful`, `unsuccessful`

Az "akció" megnevezése: `buildRobot`; `move`; `moveThroughPortal`....

Az "akció" egyéb paraméterei: `move`;2 (a második szomszédos aszteroidára mozgunk) vagy `putMaterialBack`;iron (Egy iron típusú nyersanyagot akarunk visszatenni)

Az "akció" egyéb eseményt hordozhat magával (uránium felrobban ha napközelben bányásszák, jég elolvad ha napközelben bányásszák) - `successful`;mine;ice melted vagy `successful`;mine;uranium exploded

A fenti sémának megfelelően:

bemenet:

`move`;0

`drill`

`build`;robot

kimenet:

→ sikeres mozgás esetén: `successful`;move;0

→ sikertelen drill esetén: `unsuccessful`;drill

→ sikeres robot építés esetén: `successful`;build;robot

(A kimeneti nyelv a `result/` mappába generált `.txt` fájlok tartalmán értendő, konzolon nem így fognak megjelenni a dolgok)

7.3. Use-case-ek javítása

Use-case neve	SolarStorm
Rövid leírás	A pálya egy bizonyos területén megöli azokat a figurákat akik nem tudnak elbújni egy üres aszteroida belsejében.
Aktorok	Felhasználó
Forgatókönyv	<p>1: A felhasználó úgy állítja be, hogy ne legyen napvihar a körben.</p> <p>1.a: A felhasználó azt állítja be, hogy legyen napvihar és annak mi legyen az epicentruma.</p> <p>2. A napvihar megöli azokat a figurákat akik nem tudtak elbújni.</p>

Use-case neve	checkGameEnd
Rövid leírás	A játék megnyerhetőségét ellenőrzi.
Aktorok	Felhasználó
Forgatókönyv	<p>1: A felhasználó úgy állítja be, hogy ne legyen ellenőrizve a körben.</p> <p>1.a: A felhasználó azt állítja be, hogy legyen ellenőrizve a körben a játék nyerhetősége.</p> <p>2.a A játék folytatódik tovább amennyiben megnyerhető (van elég nyersanyag az összes telepesnél és az aszteroidák belsejében).</p> <p>2.b A játék véget ér, mert nincs elég nyersanyag, hogy a bázist fel lehessen építeni.</p>

8. Részletes tervek

8.1 Osztályok és metódusok tervei.

8.1.1 Map

- **Felelősség**

A map osztály a játék alapvető logikáját biztosítja, számon tartja az összes aszteroidát, a játék aktuális állapotát, továbbá a körök lebonyolítását (lesz egy napvihar a következő körben, ki lép), továbbá, hogy nyerhető-e a játék vagy esetleg nyertek-e már a játékosok.

- **Ösosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **gameState: GameState:** A játék aktuális állapotát tárolja el
- - **asteroids: ArrayList<Asteroid>:** Eltárolja az összes aszteroidát, amely a pályát alkotja

- **Metódusok**

- **+initGame(numberOfPlayers: int): void:** A játékot inicializálja, létrehozza az aszteroidákat és a játékosokat. Ez a program elején eldönthető, hogy manuálisan vagy automatikusan történjen. Manuális választásnál minden egyes a pályával kapcsolatos dolgot megkérdez a felhasználtól. Mennyi játékos legyen, mennyi aszteroida legyen, az aszteroidák belső tulajdonságát, a szomszédságokat. Amennyiben automatikus az inicializálás úgy random szám generátor segítségével döntjük el a tulajdonságokat és a számokat amiket bizonyos határok közé helyezhetünk ezt a program kódjában kell átírni.
- **+ addAsteroid(a: Asteroid): void:** Hozzáad egy aszteroidát a térképhez (aszteroidaövhöz) a lista . add() függvényével (aszteroidaövhöz).
- **+ removeAsteroid(a: Asteroid): void:** Leszed egy aszteroidát a térképről, a lista .remove() függvényével.
- **+ hasAllMaterials(): boolean:** Leellenőrzi, hogy megvan-e a játék megnyeréséhez szükséges nyersanyag mennyiség az aszteroidaövben.
- **+ solarStorm(): void:** Generál egy napvihart ami végigsöpör a pálya egy részén és megöli azokat a figurákat akik nem tudtak elbújni egy üres aszteroidában. Ezt is lehet manuális és automatikusan futtatni. Manuális a játékos dönti el, hogy legyen-e vihar

továbbá, hogy annak hol legyen az epicentruma. A vihar úgy működik, hogy kiválaszt egy aszteroidát és azon továbbá annak az összes szomszédján következik majd be a vihar. Amennyiben ez automatikus úgy egy kis eséllyel söpör végig vihar a pályán és ennek az epicentrumát sorsolja a gép egy random szám generátorral és ezeken az aszteroidákon meghívódik a `handleFigures()` függvény (ez kezeli a halálokat).

- **+ `checkGameEnd(): boolean`:** Lellenőrzi, hogy vége lett-e már a játéknak vagy sem. Meghívja a `checkIfWinnable()` függvényt és az alapján, hogy hamis vagy igaz a visszatérési érték, véget vet a játéknak és kiírja, hogy a játékosok vesztek amennyiben nem nyerhető.
- **+ `checkIfWinnable(): boolean`:** Megnézi hogy nyerhető-e a játék vagy sem. Ezt úgy csinálja, hogy meghívja a `hasAllMaterial()` és a `hasAllFigure()` függvényeket. Utóbbi megnézi, hogy van-e még élő játékos, amennyiben nincs hamis értékkel tér vissza és a `checkIfWinnable` visszatér hamissal. A `hasAllMaterial()` összegzi az összes játékban lévő materialok számát, legyen ez telepesnél vagy még aszteroida belsejében és ezt összehasonlítja a `buildBase` feltételével és amennyiben ez nem elegendő hamis értékkel tér vissza és ekkor a `checkIfWinnable` is hamis értékkel tér vissza.
- **+ `setupRound(): void`:** Ez a függvény vezérli a teljes kört, az elején kisorsolja, hogy lesz-e vihar a körben. Amennyiben igen úgy meghívja a `solarStorm()` függvényt amennyiben viszont nem végigmegy az összes asteroidán és és meghívja rajtuk az `invokeFigures()` függvényt.
- **+ `gameEnd(boolean b): boolean`:** Kezeli a játék végét. Amennyiben nyertek a játékosok annyiban kiírja: `Settlers won the game`. Ha veszítettek a következőt: `Settlers lost the game`.
- **+ `stormComing(): boolean`:** Random kisorosolja, hogy az adott körben lesz-e napvihar vagy sem, előre beállított érték alapján.
- **+ `reset(): void`:** Meghívja a játékban lévő összes asteroidán a `resetStep()` függvényt. Ez vezérli azt, hogy a kör végén frissüljenek a `roundFinished` változók a `settler` a robot és az ufónál.

8.1.2 Asteroid

- **Felelősség**

Az aszteroida(`Asteroid`) osztály valósítja meg a `map`-et építő kis elemeket. Számon tartja és frissíti az építőelem(`aszteroida`) tulajdonságait a játék fokozatos haladásával egyidejűleg.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **layers: int** : Ez az attribútum tárolja, hogy milyen vastag az aszteroida köpenye.
- - **isNearSun: boolean**: Ez azt jelöli, hogy az aszteroida napközelen van-e, vagy sem.
- - **+isHollow: boolean**: Ez jelöli, hogy az aszteroida üreges vagy sem.
- - **portal: Portal**: Tárolja az aszteroidán lévő portált.
- - **figures: ArrayList<Figure>**: Tárolja az aszteroidán lévő figurákat.
- - **material: Material**: Tárolja az aszteroida belsejében lévő nyersanyagot.
- - **neighbors: ArrayList<Asteroid>**: Az aszteroida szomszédjait tárolja.

- **Metódusok**

- **+ explode(): void**: Az aszteroida törlését végzi, amennyiben az felrobbant.
- **+ addFigure(f: Figure): void**: Hozzáad egy figurát az aszteroidához.
- **+ removeFigure(f: Figure): void**: Eltávolít egy figurát az aszteroidáról.
- **+ setMaterial(m: Material): void**: Beállítja az aszteroida magjában található nyersanyag típusát. (Ebbe beletartozik az üreges is.)
- **+ decreaseLayers(): void**: Csökkenti eggyel az aszteroida köpenyét.
- **+ addPortal(p: Portal): void**: Hozzáad egy portált az aszteroidához.
- **+ addNeighbor (a: Asteroid): void**: Hozzáad egy aszteroidát az aszteroida szomszédjaihoz.
- **+ summarizeMaterials(): Material[]**: Megszámolja az adott aszteroidán lévő nyersanyagokat, amik a telepések inventoryjában.
- **+ invokeFigures(): void**: Minden rajta álló figurát léptet vagyis meghívja a step metódusukat.
- **+ resetStep(): void**: Meghívja minden az aszteroidán álló figurán a setRoundFinished függvényt false értékkel, vagyis kinullázza azt, hogy a karakterek léptek-e már.
- **+ drilled(): boolean**: Amikor egy figura drillel ez a függvény hívodik meg az aszteroidán és különböző értékek alapján vagy sikeres lesz vagy nem.
- **+ mined(): boolean**: Hasonlóan működik mint a drilled csak a settler mine() függvénye hívja meg.

8.1.3 Portal

- **Felelősség**

A portál osztály valósítja meg azt a tárgyat amelyet a játékosok építeni tudnak és lehelyezni továbbá azok között teleportálni így a telepések nem csak szomszédos mezőre lesznek képesek lépni. Mindig 1-1 teleportkapu áll párban, ezek között lehet utazni. A kapuk mindig tudják ki a párjuk ezt számon tartják egy változóban.

- **Össztályok**

-

- **Interfészek**

IMove

- **Attribútumok**

- - **pair: Portal:** Innen fogja tudni a portál, hogy melyik a párja.
- - **asteroid: Asteroid:** Ez a portált tartalmazó aszteroida.

- **Metódusok**

+**move(): void:** A portál egy másik szomszédos aszteroidára mozog és ezt saját változójában beállítja továbbá a régi aszteroida listájáról leveszi magát és az új aszteroida listájára pedig felveszi magát.

8.1.4 Figure

- **Felelősség**

A figura(Figure) absztrakt osztály reprezentálja az összes telepes és robot összességét. Tudják fújni az aszteroida köpenyét, és meg tudnak halni.

- **Ősosztályok**

-

- **Interfészek**

- -**ISteppable:** Steppable interfész, segítségével tudjuk kezelni a figurák lépéseit.
- -**IPortalMove:** Portál interfész, figurák teleportkapun való átlépéséért felel.
- -**IMove:** Move interfész, a figurák másik aszteroidára való mozgásáért felel.

- **Attribútumok**

- - **asteroid: Asteroid:** Ez az az aszteroida amina Figure áll.
- **# roundFinished: boolean:** Azt nézi, hogy a figura lépett-e már a körben.

- **Metódusok**

- + **move(): void:** Absztrakt függvény a mozgás megvalósításához.
- + **onExplosion(): void:** Robbanás alkalmával hívódik meg ez a függvény, ami a Figure típusától függően fog műveleteket végrehajtani.
- + **die(): void:** Ez a függvény kezeli a figurák halálát.
- + **step(): void:** Eldönti a figura következő lépését.
- + **moveThroughPortal(): boolean:** A figurát átlépteti a teleportkapu párjának az aszteroidájára.
- +**getInventory(): Inventory:** Inventory lekérdezése.

8.1.5 Robot

- **Felelősség**

Ez az osztály reprezentálja a robotokat. Mozgásuk egy mesterséges intelligencia döntése alapján történik. A aszteroida robbanása után egy szomszédos aszteroidán landolnak

- **Össztályok**

Figure

- **Interfészek**

-IDrill: A fúrás interfésze

- **Attribútumok**

-

- **Metódusok**

- **+ onExplosion(): void:** Híváskor átrakja egy véletlenszerűen választott szomszédos aszteroidára a robotot, ezt is random szám generátor sorsolja.
- **+ move(): void:** Ez a függvény hívódik meg, amikor a robot mozogni akar. A chooseNextDestination() függvény visszaadja hova mozogjon a robot, a robot leveszi magát az aszteroida figurájának a listájához hozzáadja magát az aszteroida figure listájához amire lép és frissíti az aszteroida belső tulajdonságát az új aszteroidára.
- **+ step(): void:** Eldönti, hogy a robot lép-e vagy fúr. Amennyiben tud fúrni fúr, amúgy pedig lép
- **+ chooseNextDestination(): Asteroid:** Kiválasztja a következő aszteroidát, amelyre lépni fog. Sorsol egy indexet 0 és a szomszédok száma között, a 0-át beleértve és visszaadja a sorsolt indexű szomszédot.
- **+ drill(): boolean:** A robot fúrja az aszteroidát ha annak köpenye még nem 0. Ilyenkor a köpeny réteg 1-gyel csökken.

8.1.6 Settler

- **Felelősség**

Ez az osztály reprezentálja a telepeseket. Mozgáskor a játékos választhat aszteroidát, amire tovább akarja léptetni a telepest. Bányászhat, építhet robotot, portált, ezeket pedig le is tudja rakni. A kibányászott nyersanyagokat vissza tudja tenni egy üreges aszteroida belsejébe. Ha megfelelnek a kritériumoknak az aszteroidán álló telepések, akkor építhetnek egy bázist is, ami a játék megnyeréséhez vezeti őket. Robbanáskor meghalnak.

- **Össztályok**

Figure

- **Interfészek**

-**IDrill**: A fúrás interfésze

-**IMine**: a bányászat interfésze

· **Attribútumok**

- - **inventory**: **Inventory**: Ez a telepes táskája, amiben a nyersanyagokat és portálokat tárolja.

· **Metódusok**

- + **move()**: **void**: Ez a metódus kezeli a játékos által választott mezőre átléptetését a telepesnek. A szokásos módon leveszi magát a régi aszteroidáról hozzáadja magát az újhoz és frissíti a belső változóját, hogy melyiken áll.
- + **mine()**: **boolean**: A telepes kibányászik egy nyersanyagot. Amennyiben nem tud a függvény hamis értékkel tud, amúgy pedig az inventoryba beteszi a kiválasztott nyersanyagot.
- + **buildPortal()**: **void**: Megépít egy portál párt a megfelelő nyersanyagokért cserébe. Ezt a `billOfMaterials` osztály `hasEnoughMaterials()` függvény segítségével dönti el. Amennyiben van elég megépíti és a `pay()` függvénnyel levonja az árát továbbá visszatér igaz értékkel és beállítja `roundFinished` értéket `true`-ra, viszont ha nincs elég nyersanyag a `hasEnoughMaterials()` visszatér hamis értékkel majd a `buildPortal()` is visszatér hamis értékkel. Továbbá hozzáadja az inventoryhoz a megépített kettő portált, ez is csak akkor lehetséges ha a telepes inventoryjában maximum egy portál van amúgy a `buildPortal()` visszatér hamis értékkel.
- + **buildRobot()**: **void**: Megépít egy robotot a megfelelő nyersanyagokért cserébe. Csak úgy mint a `buildPortal` itt is ellenőrzi a `hasEnoughMaterials()`-al és egy az egyben azonos dolgok történnek csak itt nem megnyerik a játékot a telepesek hanem az adott aszteroidára leépítenek egy robotot és nem ad az inventoryhoz semmit.
- + **putPortalDown()**: **void**: Letesz egy portált az adott aszteroidára.
- + **putMaterialBack()**: **boolean**: Visszatesz egy nyersanyagot az aszteroida belsejébe, ha az már üres.
- + **chooseMaterial()**: **Material**: Kiválasztja a nyersanyagot.
- + **onExplosion()**: **void**: Robbanáskor a telepes meghal.
- + **buildBase()**: **void**: Felépíti a bázist a játékos, ezzel megnyerve a játékot. Ezt is a `hasEnoughMaterial`-al teszi csak annak paraméterében nem a saját inventoryját adja át hanem a `summarizeMaterial()` függvényt hívja meg ami visszaadja az aszteroidán álló összes telepes inventoryjában lévő nyersanyagokat. Amennyiben meg tudják építeni nyernek és a függvény igaz értékkel tér vissza, amúgy pedig hamissal.
- + **step()**: **void**: A játékos eldönti, hogy lépni, fújni vagy ásni szeretne `playerinput` hatására.

8.1.7 Inventory

· **Felelősség**

Ennek az osztálynak a felelőssége a játékosoknál lévő nyersanyagok és portálok tárolása. Egyszerre maximum 2 portál és 10 nyersanyag lehet egy játékosnál.

- **Össosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **materials: material[]**: Itt tároljuk a nyersanyagokat (maximum 10)
- - **portals: ArrayList<Portal>**: Itt tároljuk a portálokat (maximum 2).
- - **portalCapacity: int**: portálok maximális száma.
- - **materialCapacity: int**: materiálok maximális száma.

- **Metódusok**

- + **addMaterial(m: Material): void**: Hozzáad egy nyersanyagot az inventoryhoz a listák .add() metódusával
- + **removeMaterial(m: Material[]): void**: Kiveszi a listában lévő számú és típusú nyersanyagot az inventoryból a listák .remove() metódusával.
- + **addPortal(p: Portal): void**: Hozzáad egy portál-párt az inventoryhoz itt is a .add() metódussal.
- + **removePortal(p: Portal): void**: Kivesz egy portált az inventoryból, itt is a .remove() metódussal.
- + **isPortalInventoryFull(): boolean**: Ellenőrzi, hogy a portálok listája maxon van-e a portál list méretét összehasonlítja a portalCapacity-vel.
- **isInventoryFull(): boolean**: Ellenőrzi, hogy az material lista maxon van-e, az inventory méretét összehasonlítja a capacity értékével.

8.1.8 Material

- **Felelősség**

Ebből az osztályból származnak le a nyersanyagok. Két függvénye van, ami a mined() és az addToInventory(). A mined() akkor fog hívódni, amikor az aszteroida kéréget kifűrták, az addToInventory pedig, amikor kibányászták a magot.

- **Össosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **asteroid: Asteroid**: Itt tároljuk a nyersanyaghoz tartozó aszteroidát.

- **Metódusok**

- **# readyToMine(): void:** Ez a függvény akkor hívódik meg, ha a nyersanyagot fedő köpenyt lefúrták. Ekkor a nyersanyagnak jelzi az aszteroida hogy készen áll a kibányászásra, a mag pedig reagálni tud erre, például azzal, hogy felrobban, vagy szublimál. Ezt a függvényt valósítja meg az Ice és az Uránium, amely nyersanyagokkal egyes helyzetekben speciálisan kell bánni.
- **+ addToInventory(s: Settler): void:** Hozzáadja a Settler inventoryjához az adott nyersanyagot. Meghívja a paraméterben kapott settler inventoryján az addMaterial függvényt az adott materiallal.

8.1.9 Iron

- **Felelősség**

Ez az osztály reprezentálja a vasat.

- **Ősosztályok**

Material

Interfészek

-

- **Attribútumok**

-

- **Metódusok**

-

8.1.10 Ice

- **Felelősség**

Ez az osztály reprezentálja a vízjeget. Ha napközben bányásszák ki, akkor szublimál.

- **Ősosztályok**

Material

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **+ readyToMine(): void:** Meghívásakor átállítja az aszteroida belsejét Hollow-ra, ezzel jelezve, hogy szublimált.

8.1.11 Coal

- **Felelősség**

Ez az osztály reprezentálja a szenet.

- **Ősosztályok**

Material

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

-

8.1.12 Uranium

- **Felelősség**

Ez az osztály reprezentálja az urániumot. Ha napközben bányásszák ki, akkor az őt tartalmazó aszteroida felrobban.

- **Ősosztályok**

Material

- **Interfészek**

-

- **Attribútumok**

-

- **Metódusok**

- **+ readyToMine(): void** Meghívásakor felrobbantja az aszteroidát. Meghívja rajta az Explode() függvényét.

8.1.13 GameState

- **Felelősség**

A GameState egy enumeration valósítja meg a játék lehetséges állapotait.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **+ IN_PROGRESS:** Ez az attribútum reprezentálja azt az állapotot, amikor játék közben vagyunk.
- **+ WIN:** Ez az állapot reprezentálja azt az állapotot, amikor a játékosok nyertek.
- **+ LOST:** Ez az állapot reprezentálja azt az állapotot, amikor a játékosok vesztek.

- **Metódusok**

-

8.1.14 BillOfMaterials

- **Felelősség**

Az osztály teszi lehetővé, hogy tesztelni lehessen a különböző építési feltételeket.

- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **bill: HashMap<Material>**: Tárolja egy listában egy adott dolog építéséhez szükséges materialokat. pl.: Portál építése 2 egység vas 1 egység víz 1 egység urán akkor a tároló tartalma 2 vas objektum 1 víz objektum 1 urán objektum
- **Metódusok**
- **countMaterials(materials: ArrayList<Material>): HashMap<Class<?>, Integer>**: Egy HashMapben visszaadja, hogy adott materialból mennyi található a paraméterben kapott Materialok között.
- **+ pay(m: ArrayList<Material>): void**: Levonja a recept összetevőit a paraméterben kapott materialok közül.
- **+ hasEnoughMaterial(m: ArrayList<Material>): boolean**: Megnézi, hogy elég nyersanyag áll-e rendelkezésre a paraméterben kapott nyersanyagok között az adott recept elkészítéséhez.

8.1.15 Ufo

• Felelősség

Az ufo entitások működését valósítja meg, amelyek megpróbálják hátráltatni a játékosokat és kibányászni előlük a nyersanyagokat.

• Ősosztályok

Figure

• Interfészek

IMine

• Attribútumok

- **-materials: Material[]**: Az ufo ebben a tömb-ben tárolja az általa kibányászott nyersanyagokat, ennek a játék során nincsen jelentősége jelenleg.

Metódusok

- **+mine(): boolean**: Az ufó kibányássza az aszteroida belsejét amin áll és a nyersanyagot az inventoryjába rakja, ezt az akciót akkor csinálja, ha az aszteroida sziklarétege 0 és a belseje nem üres, amúgy pedig mozog.
- **+moveThroughPortal(): void**: Az ufó akkor választja ezt az akciót, hogy ha a mozgásra kerül a sor, és van teleport kapu. Ilyenkor is csak 50%-os eséllyel megy portálon keresztül és nem szomszédos aszteroidára. Amennyiben ez történik, a getNextDestinationIfPortal() függvénnyel lekéri, melyik portál párjának a szomszédjának az aszteroidájára menjen át és átmegy oda. Leveszi magát a

mostani aszteroidára felteszi magát a kisorsolt aszteroidára és ezt beállítja magának és az aszteroidának is.

- **+getNextDestination(): Asteroid:** Megadja, hogy egyszerű mozgásnál melyik szomszédos aszteroidára menjen át az ufó és ezt visszaadja. 0-tól sorsol egy indexet a szomszédos aszteroidák mérete-1-ig.
- **+getNextDestinationIfPortal(): Asteroid:** Megadja, hogy ha portálon keresztül mozogna az ufó akkor melyik portált válassza. Ezt úgy csinálja, hogy megnézi mely portáloknak nincs párja ezeket összegyűjti egy tiltott listában, utána pedig addig sorsol számokat ameddig olyan indexet kap ami nincs rajta a tiltott listán. Ezután megkeresi ennek a portálnak a párjának az aszteroidáját és returnolja.
- **+step(): void:** Ez a függvény lépteti az ufót egy körben. Amennyiben az aszteroida köpenye 0 és a magja nem üres a mine() metódusa hívódik meg, amúgy pedig sorsol egy számot ami vagy 0 vagy 1. Amennyiben 0 akkor a sima move() hívódik meg amennyiben 1 a moveThroughPortal() függvénye hívódik meg.

8.1.16 UserIO

- **Felelősség**

A bemenet és kimenet kezelése.

- **Össosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- **-static Scanner scanner:** Az adatok beolvasásához használt Scanner
- **-static boolean showInput:** Azt adja meg hogy vissza akarjuk-e jelezni a usernek hogy milyen inputot kaptunk tőle
- **-static boolean readFromFile:** azt adja meg, hogy fájlból olvasunk-e
- **-static boolean checkIfWinnable:** azt adja meg hogy a program ellenőrzi-e a játék megnyerhetőségét
- **-static boolean isAutomatic:** azt adja meg hogy a játék automatikusan folyik-e, vagy a user manuálisan fogja beírni a játék történéseit
- **-static String path:** az olvasandó file elérési útját adja meg
- **-static final ArrayList<String> currentLine:** a legutoljára beolvasott sort tárolja, ezzel tudunk referálni paraméterekre (pl move;1 esetén ezzel érjük el az 1-et később)
- **-static final ArrayList<String> temporaryInput:** itt tároljuk azokat a bemeneteket, amiket még nem írunk bele egy sorba. (pl a move;1 esetén a move-ot még nem írjuk, csak ha minden paramétert kiértékelünk már)
- **-static final ArrayList<String> customInput:** ez az a lista amibe kerülnek a végén kiírásra szánt sorok

· **Metódusok**

- **public static void openFile():** megnyitja a path-ben talált fájlt. ha üres, akkor console inputról olvas.
- **public static void closeFile():** bezárja az eddig olvasott file-t
- **private static String readNextLine():** beolvassa a következő sort az inputon, szükség esetén átvált consolerre
- **public static ArrayList<String> readLine():** feldarabolja a bemenetet a pontosvesszők alapján
- **public static int readInt():** beolvas egy integert
- **public static String readString():** beolvas egy stringet
- **public static void addToTemporaryInput(String tmp):** hozzáad egy stringet a temporaryinputhoz
- **public static void addToTemporaryOutput(String tmp):** hozzáad egy stringet a temporaryoutputhoz
- **public static void clear():** kitörli a temporaryinput és a custominput tartalmát
- **public static void addToCustomInput():** hozzáadja a temporaryinput tartalmát a custominput-hoz
- **public static void addToResultOutput():** hozzáadja a temporaryinput tartalmát a custominput-hoz
- **public static void clearTemporaryInput():** kitörli a temporaryinput tartalmát
- **public static void choosePath(Phase phase):** átváltja a Phase-t a paraméterben kapottra
- **public static void saveCustomIO(Phase phase, String filename):** elmenti a custominputon lévő sorokat a kért fileba

8.1.15 GameState

· **Felelősség**

A Phase határozza meg a UserIO számára, hogy inicializálási, vagy teszt fázisban vagyunk-e.

· **Össztályok**

-

· **Interfészek**

-

· **Attribútumok**

- **+ INIT:** Inicializálási fázis
- **+ TEST:** Teszt fázis
- **+ RESULT:** A futás eredmény kiírásának a fázisa

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

8.2.1 Move

- **Leírás**

Egy telepes, robot vagy ufo átlép egy szomszédos aszteroidára. Az egyszerűség kedvéért csak telepesre írtuk meg, mindegyikre ugyanúgy működik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes, robot és ufo szomszédos aszteroidára való lépésének tesztelése.

Hibalehetőség: az entitás nem kerül át a másik aszteroidára, az entitás eltűnik (az egyik aszteroida eltávolítja, viszont a másik nem adja hozzá).

- **Bemenet**

a pálya inicializálásához: (moveThroughPortallnit-InventoryFull-hasTwoPortals.txt)

```
1
3
1;hollow;1;nearsun
1;uranium;1;notnearsun
3;ice;1;notnearsun
0
0
2
0;1
1;2
1
0;1
0
2;2;2;4;1
```

a tesztet indításához:

```
move;0
```

- **Elvárt kimenet**

```
successful;move;0
```

8.2.2 Drill Asteroid

- **Leírás**

Egy telepes vagy robot aszteroidát fúr és csökkenti az aszteroida rétegeinek számát eggyel. Az egyszerűség kedvéért csak telepesre írtuk meg, mindegyikre ugyanúgy működik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes vagy robot fúrás tevékenységének tesztelése.

Hibalehetőség: az aszteroida rétegei nem csökkennek eggyel.

- **Bemenet**

a pálya inicializálásához: (drill-1asteroid-nothollow-hasOneLayer.txt)

```
1
1
1;ice;1
0
0
0
0
0
0
2;2;2;2;2
```

a tesztet indításához:

```
drill
```

- **Elvárt kimenet**

```
successful;drill
```

8.2.3 Can't Drill Asteroid, Layers = 0

- **Leírás**

Egy telepes vagy robot megpróbál fúrni egy aszteroidát, de annak nincs több rétege. Az egyszerűség kedvéért csak telepesre írtuk meg, mindegyikre ugyanúgy működik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes vagy robot fúrás tevékenységének tesztelése.

Hibalehetőség: az aszteroida rétegei csökkennek eggyel, megtörténik a fúrás és a következő játékos léphet.

- **Bemenet**

a pálya inicializálása: (1Player-1Asteroid(0 layers), Ice.txt):

```
1
1
0;iron;1;0
0
0
```

```
0
0
0
2;2;2;2;0
```

a tesztet indításához:

```
drill
```

- **Elvárt kimenet**

```
unsuccessful;drill
```

8.2.4 Drill Last Layer of Radioactive Asteroid Near Sun

- **Leírás**

Egy teleses vagy robot egy radioaktív (uránt tartalmazó) aszteroida utolsó rétegét fúrja át napközben. Az egyszerűség kedvéért csak telesesre írtuk meg, mindegyikre ugyanúgy működik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teleses vagy robot fúrás tevékenységének tesztelése, az aszteroida felrobbanásának tesztelése.

Hibalehetőség: az aszteroida rétegei nem csökkennek eggyel, az aszteroida nem robban fel, a teleses nem hal meg, a robot nem kerül át szomszédos aszteroidára.

- **Bemenet**

a pálya inicializálásához:

(2Players-2Asteroids-CanBuildAnything-FirstAsteroidNearSun(0,1).txt)

```
2
2
1;uranium;1;1
1;ice;1
0
0
1
0;1
0
0
2;2;2;2;0
0
2;2;2;2;0
```

a teszteset indításához:

```
drill
```

- **Elvárt kimenet**

```
successful;drill;asteroidExploded
```

8.2.5 Drill Last Layer of Asteroid Containing Ice Near Sun

- **Leírás**

Egy telepes vagy robot egy vízjeget tartalmazó aszteroida utolsó rétegét fúrja át napközelsben. Az egyszerűség kedvéért csak telepesre írtuk meg, mindegyikre ugyanúgy működik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes vagy robot fúrás tevékenységének tesztelése. A jég elolvadásának tesztelése. Hibalehetőség: az aszteroida rétegei nem csökkennek eggyel, a jég nem olvad el.

- **Bemenet**

a pálya inicializálásához: (drill-1asteroid-nothollow-hasOneLayer.txt)

```
1
1
1;ice;1
0
0
0
0
0
0
2;2;2;2;2
```

a teszteset indításához:

```
drill
```

- **Elvárt kimenet**

```
successful;drill;iceMelted
```

8.2.6 Settler Mines

- **Leírás**

Egy telepes kibányászik egy aszteroidát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes bányászat tevékenységének tesztelése.

Hibalehetőség: a nyersanyag nem kerül az inventory-jába, az aszteroidából nem tűnik el a nyersanyag, nem lesz üreges.

- **Bemenet**

a pálya inicializálása: (1Player-1Asteroid(0 layers), Ice.txt):

```
1
1
0;iron;1;0
0
0
0
0
0
2;2;2;2;0
```

a tesztet indításához:

```
mine
```

- **Elvárt kimenet**

```
successful;mine
```

8.2.7 Ufo Mines

- **Leírás**

Egy ufo kibányászik egy aszteroidát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az ufo bányászat tevékenységének tesztelése.

Hibalehetőség: a nyersanyag nem kerül az inventory-jába, az aszteroidából nem tűnik el a nyersanyag, nem lesz üreges.

- **Bemenet**

a pálya inicializálása: (1Ufo-1UraniumAsteroid.txt):

```
0
```

```
1
0;uranium;1;0
0
1
0
0
0
```

a teszteset indításához:

```
mine
```

- **Elvárt kimenet**

```
successful;mine
```

8.2.8 Mine Hollow Asteroid

- **Leírás**

Egy telepes vagy ufo megpróbál kibányászni egy aszteroidát, de az üreges. Az egyszerűség kedvéért csak telepesre írtuk meg, mindegyikre ugyanúgy működik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes vagy ufo bányászat tevékenységének tesztelése.

Hibalehetőség: megtörténik a bányászás és a következő játékos köre következik.

- **Bemenet**

a pálya inicializálásához: (1Player-1HollowAsteroid-CanNotBuildAnything.txt)

```
1
1
0;hollow;1;0;
0
0
0
0
0
0;0;0;1;0
```

a teszteset indításához:

```
mine
```

- **Elvárt kimenet**


```
successful;mine
```

8.2.9 Can't Mine, Layers > 0

- **Leírás**

Egy telepes vagy ufo megpróbál kibányászni egy aszteroidát, de még nincs lefúrva minden réteg az aszteroidán. Az egyszerűség kedvéért csak telepesre írtuk meg, mindegyikre ugyanúgy működik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes vagy ufo bányászat tevékenységének tesztelése.

Hibalehetőség: megtörténik a bányászás és a következő játékos köre következik.

- **Bemenet**

a pálya inicializálásához: (drill-1asteroid-nothollow-hasOneLayer.txt)

```
1
1
1;ice;1
0
0
0
0
0
2;2;2;2;2
```

a tesztet indításához:

```
mine
```

- **Elvárt kimenet**

```
unsuccessful;mine
```

8.2.10 Can't Mine, Inventory Is Full

- **Leírás**

Egy telepes vagy ufo megpróbál kibányászni egy aszteroidát, de az Inventory-ja tele van. Az egyszerűség kedvéért csak telepesre írtuk meg, mindegyikre ugyanúgy működik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes vagy ufo bányászat tevékenységének tesztelése.

Hibalehetőség: ki tudja bányászni és az Inventory-jába kerül a nyersanyag vagy megtörténik a bányászás, nem kerül az Inventory-ba és a következő játékos lép.

- **Bemenet**

a pálya inicializálásához: (moveThroughPortallnit-InventoryFull-hasTwoPortals.txt)

```
1
3
1;hollow;1;nearsun
1;uranium;1;notnearsun
3;ice;1;notnearsun
0
0
2
0;1
1;2
1
0;1
0
2;2;2;4;1
```

a teszteset indításához:

```
mine
```

- **Elvárt kimenet**

```
unsuccessful;mine
```

8.2.11 Move Through Portal

- **Leírás**

Egy telepes, robot vagy ufo teleportkapu-páron keresztül való mozgása. Az egyszerűség kedvéért csak telepesre írtuk meg, mindegyikre ugyanúgy működik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes, robot és ufo teleportkapu-páron keresztül való mozgásának tesztelése.

Hibalehetőség: nem sikerül átlépni a teleportkapu párjához, a portálnak nincs párja, a portál nem ahhoz az aszteroidához tartozik.

- **Bemenet**

a pálya inicializálásához: (moveThroughPortallnit-InventoryFull-hasTwoPortals.txt)

```
1
```

```
3
1;hollow;1;nearsun
1;uranium;1;notnearsun
3;ice;1;notnearsun
0
0
2
0;1
1;2
1
0;1
0
2;2;2;4;1
```

a teszteset indításához:

```
moveThroughPortal;0
```

- **Elvárt kimenet**

```
successful;moveThroughPortal;0
```

8.2.12 Build Teleport Portal

- **Leírás**

Egy telepes teleportkapu-párt épít.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes teleportkapu-pár építésének tesztelése.

Hibalehetőség: nem épül meg a teleportkapu-pár, nem kerülnek be a telepes Inventory-jába, csak az egyik teleportkapu épül meg, nem vonja le a nyersanyagokat a telepes inventory-jából.

- **Bemenet**

a pálya inicializálásához: (moveThroughPortalInit-InventoryFull-hasTwoPortals.txt)

```
1
3
1;hollow;1;nearsun
1;uranium;1;notnearsun
3;ice;1;notnearsun
0
0
2
0;1
1;2
1
```

```
0;1
0
2;2;2;4;1
```

a teszteset indításához:

```
build;portal
```

- **Elvárt kimenet**

```
successful;build;portal
```

8.2.13 Can't Build Teleport Portal, Not Enough Material

- **Leírás**

Egy telepes teleportkapu-párt próbál építeni, de nincs elegendő nyersanyaga.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes teleportkapu-pár építésének tesztelése.

Hibalehetőség: megépül a teleportkapu-pár, levonja a nyersanyagokat az inventory-ból.

- **Bemenet**

a pálya inicializálásához: (1Player-1HollowAsteroid-CanNotBuildAnything.txt)

```
1
1
0;hollow;1;0;
0
0
0
0
0
0;0;0;1;0
```

a teszteset indításához:

```
build;portal
```

- **Elvárt kimenet**

```
unsuccessful;build;portal
```

8.2.14 Can't Build Teleport Portal, Inventory Is Full

- **Leírás**

Egy telepes teleportkapu-párt próbál építeni, de nincs elég hely az Inventory-jában.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes teleportkapu-pár építésének tesztelése.

Hibalehetőség: megépül a teleportkapu-pár, bekerülnek az inventory-ba, levonja a nyersanyagokat.

- **Bemenet**

a pálya inicializálásához: (moveThroughPortalInit-InventoryFull-hasTwoPortals.txt)

```
1
3
1;hollow;1;nearsun
1;uranium;1;notnearsun
3;ice;1;notnearsun
0
0
2
0;1
1;2
1
0;1
0
2;2;2;4;1
```

a tesztet indításához:

```
build;portal
```

- **Elvárt kimenet**

```
unsuccessful;build;portal
```

8.2.15 Build Robot

- **Leírás**

Egy telepes robotot épít.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes robot építésének tesztelése.

Hibalehetőség: nem készül el a robot, nem vonja le a nyersanyagokat az inventory-ból.

- **Bemenet**

a pálya inicializálásához: (moveThroughPortallInit-InventoryFull-hasTwoPortals.txt)

```
1
3
1;hollow;1;nearsun
1;uranium;1;notnearsun
3;ice;1;notnearsun
0
0
2
0;1
1;2
1
0;1
0
2;2;2;4;1
```

a teszteset indításához:

```
build;robot
```

- **Elvárt kimenet**

```
successful;build;robot
```

8.2.16 Can't Build Robot, Not Enough Material

- **Leírás**

Egy telepes robotot próbál építeni, de nincs elegendő nyersanyaga.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes robot építésének tesztelése.

Hibalehetőség: a robot megépül, a nyersanyagok levonódnak az inventory-ból.

- **Bemenet**

a pálya inicializálásához: (1Player-1HollowAsteroid-CanNotBuildAnything.txt)

```
1
1
0;hollow;1;0;
0
0
0
0
0
0
```

```
0;0;0;1;0
```

a teszteset indításához:

```
build;robot
```

- **Elvárt kimenet**

```
unsuccessful;build;robot
```

8.2.17 Build Base

- **Leírás**

A telepések bázist építenek és megnyerik a játékot.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepések bázis építésének és játék megnyerésének tesztelése.

Hibalehetőség: nem épül meg a bázis, nem vonódnak le a nyersanyagok, nem nyerik meg a játékot.

- **Bemenet**

a pálya inicializálásához:

(2Players-2Asteroids-CanBuildAnything-FirstAsteroidNearSun(0,1).txt)

```
2
2
1;uranium;1;1
1;ice;1
0
0
1
0;1
0
0
2;2;2;2;0
0
2;2;2;2;0
```

a teszteset indításához:

```
build;base
```

- **Elvárt kimenet**

```
successful;build;base
```

8.2.18 Can't Build Base, Not Enough Material

- **Leírás**

A telepések bázist próbálnak építeni, de nincs elegendő nyersanyaguk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepések bázis építésének és játék megnyerésének tesztelése.

Hibalehetőség: a bázis megépül, levonódnak a nyersanyagok az inventory-ból, megnyerik a játékot.

- **Bemenet**

a pálya inicializálásához: (1Player-1HollowAsteroid-CanNotBuildAnything.txt)

```
1
1
0;hollow;1;0;
0
0
0
0
0
0
0;0;0;1;0
```

a tesztet indításához:

```
build;base
```

- **Elvárt kimenet**

```
unsuccessful;build;robot
```

8.2.19 Storm Kills Player

- **Leírás**

Egy telepés vagy robot nem üreges aszteroidán áll és meghal a napviharban. Az egyszerűség kedvéért csak telepésre írtuk meg, mindegyikre ugyanúgy működik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepések és robotok meghalásának tesztelése napviharban.

Hibalehetőség: nem halnak meg.

- **Bemenet**

a pálya inicializálásához: (drill-1asteroid-nothollow-hasOneLayer.txt)

```
1
1
1;ice;1
0
0
0
0
0
2;2;2;2;2
```

a teszteset indításához:

```
solarStorm;0
```

- **Elvárt kimenet**

```
solarStorm;0;player0Dead
```

8.2.20 Storm Does Not Kill Player

- **Leírás**

Egy telepes vagy robot üreges aszteroidán áll és túléli a napvihart. Az egyszerűség kedvéért csak telepesre írtuk meg, mindegyikre ugyanúgy működik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepesek és robotok meghalásának tesztelése napviharban.
Hibalehetőség: meghalnak.

- **Bemenet**

a pálya inicializálásához: (1Player-1HollowAsteroid-CanNotBuildAnything.txt)

```
1
1
0;hollow;1;0;
0
0
0
0
0
0
0;0;0;1;0
```

a teszteset indításához:

```
solarStorm;0
```

- **Elvárt kimenet**

```
solarStorm;0
```

8.2.21 Put Down First Portal

- **Leírás**

Egy telepes lerak egy teleportkaput.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes teleportkapu lerakás tevékenységének tesztelése.

Hibalehetőség: nem kerül az aszteroidára a teleportkapu, nem vonódik le az inventory-jából.

- **Bemenet**

a pálya inicializálásához: (moveThroughPortalInit-InventoryFull-hasTwoPortals.txt)

```
1
3
1;hollow;1;nearsun
1;uranium;1;notnearsun
3;ice;1;notnearsun
0
0
2
0;1
1;2
1
0;1
0
2;2;2;4;1
```

a tesztet indításához:

```
putPortalDown
```

- **Elvárt kimenet**

```
successful;putPortalDown
```

8.2.22 Put Down Second Portal

- **Leírás**

Egy telepes lerakja a teleportkapu párját is

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes teleportkapu lerakás tevékenységének tesztelése.

Hibalehetőség: nem kerül az aszteroidára a teleportkapu, nem vonódik le az inventory-jából, nem állítja be a párját.

- **Bemenet**

a pálya inicializálásához: (moveThroughPortalInit-InventoryFull-hasTwoPortals.txt)

```
1
3
1;hollow;1;nearsun
1;uranium;1;notnearsun
3;ice;1;notnearsun
0
0
2
0;1
1;2
1
0;1
0
2;2;2;4;1
```

a tesztet indításához:

```
putPortalDown
```

- **Elvárt kimenet**

```
successful;putPortalDown
```

8.2.23 Put Back Material

- **Leírás**

Egy telepes visszarak egy nyersanyagot egy üreges aszteroidába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes nyersanyag visszarakásának tesztelése

Hibalehetőség: nem kerül az aszteroidába a nyersanyag, nem vonódik le az inventory-jából.

- **Bemenet**

a pálya inicializálásához: (1Player-1HollowAsteroid-CanNotBuildAnything.txt)

```
1
1
0;hollow;1;0;
0
0
0
0
0
0;0;0;1;0
```

a teszteset indításához:

```
putMaterialBack;iron
```

- **Elvárt kimenet**

```
successful;putMaterialBack;iron
```

8.2.24 Can't Put Back Material, Not Hollow

- **Leírás**

Egy telepes megpróbál visszarakni egy nyersanyagot, de az aszteroida nem üreges.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes nyersanyag visszarakásának tesztelése.

Hibalehetőség: sikerül behelyezni az aszteroidába a nyersanyagot és a következő játékos lép, levonódik az inventory-jából a nyersanyag.

- **Bemenet**

a pálya inicializálása: (1Player-1Asteroid(0 layers), lce.txt):

```
1
1
0;iron;1;0
0
0
0
0
0
0
2;2;2;2;0
```

a teszteset indításához:

```
putMaterialBack;iron
```

- **Elvárt kimenet**

```
unsuccessful;putMaterialBack;iron
```

8.2.25 Can't Put Back Material, Layers > 0

- **Leírás**

Egy telepes megpróbál visszarakni egy nyersanyagot, de az aszteroida rétegei még nincsenek teljesen lefűrva.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A telepes nyersanyag visszarakásának tesztelése.

Hibalehetőség: sikerül behelyezni az aszteroidába a nyersanyagot és a következő játékos lép, levonódik az inventory-jából a nyersanyag.

- **Bemenet**

a pálya inicializálásához: (drill-1asteroid-nothollow-hasOneLayer.txt)

```
1
1
1;ice;1
0
0
0
0
0
0
2;2;2;2;2
```

a tesztet indításához:

```
putMaterialBack;iron
```

- **Elvárt kimenet**

```
unsuccessful;putMaterialBack;iron
```

8.3 A tesztelést támogató programok tervei

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

A teszteléshez nagy részben a UserIO osztály funkcióit fogjuk felhasználni, aminek a részletes terve feljebb olvasható az "Osztályok és metódusok tervei" részénél, ahol az osztályokat fejtettük ki. Ezen kívül a tesztek egyszerű lefuttatásához egy .txt fájlt fogunk használni (testname.txt), amivel könnyedén le lehet futtatni az adott tesztet és el lehet dönteni, hogy minden rendeltetése szerűen működik-e. Ahhoz hogy ezt eldöntsük, lesznek elvárt kimenetek is. A fájlok kiértékelésére a command promptból elérhető (`FC [pathname1] [pathname2]`) felel amivel két fájl tartalmának egyezését tudjuk ellenőrizni. Esetünkben két .txt fájlt fogunk összehasonlítani melyek minden sorának egyezőnek kell lenni. Amennyiben nem sikeres a lefutás, a filecompare parancs jelzi az összes különbséget a két file között.

8.4 Napló

Kezdet	Időtartam	Résztevők	Leírás
2021.03.31. 11:00	1 óra	Bödöcs Halász Lauber Pintér	Megbeszélés
2021.03.31 12:00	4.5 óra	Halász Hátori Pintér	Osztályok leírása, fájlból beolvasás, fájl mentése
2021.03.31 16:30	1.5 óra	Halász Pintér	fájlból beolvasás, fájl mentése
2021.03.31. 18:20	1 óra	Pintér	UserIO fájl bővítése settler step függvénye fájlból olvasással
2021.03.31. 20:30	1 óra	Pintér	settler step függvénye, fájlból olvasással
2021.04.01. 8:00	1 óra	Pintér	függvények javítása tesztesetek segítségével

2021.04.01. 11:00	0.5 óra	Bödőcs	test txt-k írása
2021.04.02. 14:30	2 óra	Pintér	egyedi tesztesetek menthetőségének készítése
2021.04.06. 9:00	2 óra	Bödőcs	initGame szerkesztése bemeneti nyelv alapján
2021.04.06. 11:30	0.5 óra	Bödőcs	test txt nevek szerkesztése
2021.04.07. 16:15	1.5 óra	Halász Hámori Lauber Pintér	fájlba írhatóság megvalósítása tesztesetek írása megbeszélés
2021.04.07 17:45	3.5 óra	Bödőcs Halász Lauber Pintér	fájlba írhatóság megvalósítása tesztesetek írása initGame javítása
2021.04.08 16:30	1 óra 10p	Halász	tesztesetek javítása, bemeneti nyelv frissítése
2021.04.09. 15:00	1.5 óra	Pintér	custom tesztesetek meg inicializálások

			menthetőségének megvalósítása
2021.04.09. 16:30	0.5 óra	Lauber Pintér	nearSunCount initben menthetősége
2021.04.11 21:45	2 óra	Hámori	dokumentáció frissítése
2021.04.12. 11:30	1.5 óra	Bödőcs Halász Lauber Pintér	dokumentáció javítása