

8. Részletes tervek

68 – 00ff00

Konzulens:

Szabó Bence Sándor

Csapattagok

| | | |
|-----------------------------|--------|-------------------------------------------------------------------------------------|
| Páter Balázs Ábrahám | IW6H6X | pater.balazs@gmail.com (kapcsolattartó) |
| Bereczky Dániel | WKMTM2 | dani.bereczky@gmail.com |
| Horváth Gergő Zsombor | QJKEKQ | horvgergi@gmail.com |
| Miczki Márk | DO9MAO | miczki-mark@gmail.com |
| Nagy László | BC7TB3 | nagy.laszlo1130@gmail.com |

2024.04.15.

8. Részletes tervek

8.0 Módosítások

A 3 fake item osztályának átdolgozása. A FakeMask a Mask osztályból, a FakeSlideRule a SlideRule, a FakeTVSZ pedig a TVSZ osztályból származik, és felülírja az őssztály függvényeit.

8.0.1 Bemeneti nyelv módosítások

A játék elindulásakor megkérdezi a program a felhasználót, hogy teszt módban szeretné-e futtatni a játékot, erre egy 'y'-t vár ha a válasza igen, ellentétes esetben pedig egy 'n' karaktert. Ha a felhasználó teszt módban szeretné használni a programot, akkor képes kézzel, parancsok megadásával definiálni mindent, emellett tesztek futtatni. A következő parancsok kizárólag teszt módban használhatóak: 'addroom', 'adddoor', 'additem', 'addentity', 'executetest'. A következő parancsok pedig kizárólag normál módú indításkor használhatóak: 'addplayer', 'removeplayer', 'showplayers'. A többi parancs mindkettő módú futtatás esetén használható. Léteznek olyan parancsok, melyeket kizárólag a játék elindítása előtt lehet csak használni, ezek a következők: 'load', 'addroom', 'adddoor', 'additem', 'addentity', 'addplayer', 'removeplayer', 'showplayers', 'start'. Van néhány parancs, mely elindítás előtt és után is használható, ezek a következők: 'debug', 'execute', 'executetest'. A többi parancs szigorúan csak a játék elindítását követően használható.

Az új bemeneti parancsok:

taddroom

Leírás: Használatával a játék kezdete előtt hozzáadhatunk egy új szobát a már meglévő szobák listájához. A szoba létrehozásához kötelező megadni a következő paramétereket:

- 't' vagy 'f' karakter, attól függően hogy azt szeretnénk-e, hogy a szoba gázos legyen-e vagy sem. 't', ha azt szeretnénk hogy gázos legyen, 'f' hogyha nem.
- Egy egész szám, a szoba befogadóképességét reprezentálva.
- Egy egész szám, a szoba egyedi azonosítóját reprezentálva.

A parancs formátuma:

taddroom <'t'/'f'> <capacity> <roomID>

tadddoor

Leírás: Használatával a játék kezdetet előtt hozzáadhatunk egy új ajtót, mellyel kettő már létező szobát összeköthetünk. Az ajtó létrehozásához kötelező megadni a következő paramétereket:

- Egy egész szám, az első szoba egyedi azonosítója.
- Egy egész szám a második szoba egyedi azonosítója.
- 't' vagy 'f' karakter, attól függően, hogy azt szeretnénk-e , hogy az ajtó elátkozott legyen, 't' ha igen, 'f' ha nem.
- 'r' vagy 'l' vagy 'b' karakter, a szoba átjárhatóságát reprezentálva. 'r', ha azt szeretnénk, hogy az ajtó csak jobbról legyen átjárható (az első szobából a második irányába), 'l', ha azt szeretnénk hogy az ajtó csak balról legyen átjárható (a második szobából az elsőbe), 'b', ha azt szeretnénk, hogy mindkettő irányból át lehessen menni rajta.
- Egy egész szám, az ajtó egyedi azonosítóját reprezentálva.

A parancs formátuma:

tadddoor <roomID1> <roomID2> <'t'/'f'> <'r'/'l'/'b'> <doorID>

tadditem

Leírás: Használatával a játék kezdete előtt hozzáadhatunk egy új tárgyat a már meglévő szobák egyikéhez. A tárgy létrehozásához kötelező megadni a következő paramétereket:

- 'airfreshner'/'beer'/'camembert'/'fakemask'/'fakesliderule'/'faketvsz'/'mask'/'rag'/'sliderule'/'transistor'/'tvsz', melyek mind a hozzáadandó tárgyat specifikálják.
- Egy egész szám, a tárgy egyedi azonosítója.
- Egy egész szám, a szoba egyedi azonosítója, melybe a tárgy kerülni fog.

A parancs formátuma:

```
tadditem <'airfreshner'/'beer'/'camembert'/'fakemask'/'fakesliderule'/'faketvsz'/'mask'/'rag'/'sliderule'/'transistor'/'tvsz'> <itemID> <roomID>
```

taddentity

Leírás: Használatával a játék kezdete előtt hozzáadhatunk egy új entitást a már meglévő entitások listájához. Az entitás létrehozásához kötelező megadni a következő paramétereket:

- 's' vagy 't' vagy 'j'. 's', ha egy diákot szeretnénk hozzáadni, 't', ha egy tanárt szeretnénk hozzáadni, 'j', ha pedig egy takarítót szeretnénk hozzáadni.
- Az entitás neve.
- Egy egész szám, az entitás egyedi azonosítója.
- Egy egész szám, a szoba egyedi azonosítója, melybe az entitás kerülni fog.

A parancs formátuma:

```
taddentity <'s'/'t'/'j'> <name> <entityID> <roomID>
```

tmerge

Leírás: Két szoba össze olvasztásához használható parancs, csak teszt módban működik. A parancs neve után a két összeolvasztandó szoba ID-jét várja whitespace-el elválasztva.:

A parancs formátuma:

```
tmerge <roomID> <roomID>
```

tsplit

Leírás: Egy szoba ketté választásához használható parancs, csak teszt módban működik. A parancs neve után a ketté választandó szoba ID-jét várja:

A parancs formátuma:

```
tsplit <roomID>
```

executetest

Leírás: Használatával egy előre megírt tesztet futtathatunk le, és hasonlíthatjuk össze kimenetelét egy előre megírt elvárt működést reprezentáló file-al. A parancs futtatásához kötelező megadni a következő paramétereket:

- A bemeneti fájl neve, melyből a parancsokat kiolvassa.
- Az elvárt működést leíró fájl neve.

A parancs formátuma:

```
texecutetest <inputfile> <expectedfile>
```

start

Leírás: A játék elindításához használt parancs is változott néhány dologban. Attól függően, hogy a játékot teszt módban futtatjuk különböző módon várja a felhasználótól a parancs paraméterezését. Ha normál módban fut a játék, akkor a használata megegyezik az eddigi használattal, azaz nem kér semmilyen más paramétert a felhasználótól. Ha viszont teszt módban futtatjuk a játékot, akkor a következő paramétereket kötelező megadni:

- Egy egész szám, mely a játék hátralévő köreit reprezentálja.
- 't' vagy 'm' attól függően, hogy kézzel szeretnénk-e mozgatni az entitásokat az elindítás után, vagy egy fileből olvassuk be az akcióinkat. 't', hogyha egy fileből szeretnénk ezeket olvasni, vagy 'm' hogyha manuális módon szeretnénk parancsokat megadni.

A parancs formátuma normál módban:

start

A parancs formátuma teszt módban:

start <remainingRounds> <'t'/'m'>

8.0.1 Kimeneti nyelv módosítások

taddroom

Kiírja, hogy az új szoba hozzáadása sikeres volt-e vagy sem. Hogyha helytelenül lett paraméterezve, akkor hibaüzenet segítségével jelez a felhasználónak.

Egy lehetséges sikeres kimenet szoba hozzáadása esetén:

Room successfully added!

Egy lehetséges hibás kimenet rossz paraméterezés esetén:

ERROR -> Wrong syntax!

tadddoor

Kiírja, hogy az új ajtó hozzáadása sikeres volt-e vagy sem. Hogyha helytelenül lett paraméterezve, akkor hibaüzenet segítségével jelez a felhasználónak. Ha olyan szobát próbálunk megadni az ajtónak, amely nem létezik, akkor speciális hibaüzenet segítségével jelzi azt.

Egy lehetséges sikeres kimenet ajtó hozzáadása esetén:

Door successfully added!

Egy lehetséges hibás kimenet rossz paraméterezés esetén:

ERROR -> Wrong syntax!

Egy lehetséges hibás kimenet nem létező szobával való paraméterezés esetén:

ERROR -> There is no room with ID #13.

tadditem

Kiírja, hogy az új tárgy hozzáadása sikeres volt-e vagy sem. Hogyha helytelenül lett paraméterezve, akkor hibaüzenet segítségével jelez a felhasználónak. Ha olyan szobát próbálunk megadni a tárgynak, amely nem létezik, akkor speciális hibaüzenet segítségével jelzi azt.

Egy lehetséges sikeres kimenet szoba hozzáadása esetén:
Item successfully added!

Egy lehetséges hibás kimenet rossz paraméterezés esetén:
ERROR -> Wrong syntax!

Egy lehetséges hibás kimenet nem létező szobával való paraméterezés esetén:
ERROR -> There is no room with ID #13.

taddentity

Kiírja, hogy az új entitás hozzáadása sikeres volt-e vagy sem. Hogyha helytelenül lett paraméterezve, akkor hibaüzenet segítségével jelez a felhasználónak. Ha olyan szobát próbálunk megadni az entitásnak, amely nem létezik, akkor speciális hibaüzenet segítségével jelzi azt.

Egy lehetséges sikeres kimenet diák hozzáadása esetén:
Student successfully added!

Egy lehetséges sikeres kimenet tanár hozzáadása esetén:
Teacher successfully added!

Egy lehetséges sikeres kimenet takarító hozzáadása esetén:
Janitor successfully added!

Egy lehetséges hibás kimenet rossz paraméterezés esetén:
ERROR -> Wrong syntax!

Egy lehetséges hibás kimenet nem létező szobával való paraméterezés esetén:
ERROR -> There is no room with ID #13.

tmerge

Jelzi ha a parancs sikeresen futott le, esetleg ha hibába ütközött akkor milyenbe a következő módokon:

Egy lehetséges kimenet sikeres összeolvasztás esetén:
Merge successful.

Egy lehetséges hibás kimenet rossz paraméterezés esetén:
ERROR -> Wrong syntax!

Egy lehetséges hibás kimenet nem létező szobával való paraméterezés esetén:
ERROR -> There is no room with ID 13 or 14.

tsplit

Jelzi ha a parancs sikeresen futott le, esetleg ha hibába ütközött akkor milyenbe a következő módokon:

Egy lehetséges kimenet sikeres kettéválasztás esetén esetén:
Split successful.

Egy lehetséges hibás kimenet rossz paraméterezés esetén:
ERROR -> Wrong syntax!

Egy lehetséges hibás kimenet nem létező szobával való paraméterezés esetén:
ERROR -> There is no room with ID 13.

executetest

Kiírja, hogy a teszt lefuttatása sikeres volt-e vagy sem. Hogyha helytelenül lett paraméterezve, akkor hibaüzenet segítségével jelez a felhasználónak. Ha nem egyezik meg az aktuális és az elvárt fájl egyik sora, akkor azt jelzi.

Egy lehetséges sikeres kimenet teszt futtatása esetén:
Test successful!
The map has been cleared, ready for new command!

Egy lehetséges hibás kimenet rossz paraméterezés esetén:
ERROR -> Wrong syntax!

Egy lehetséges hibás kimenet ha az elvárt és az aktuális fileok hossza nem egyezik:

Input files does not match in length aborted.
The map has been cleared, ready for new command!

Egy lehetséges hibás kimenet ha az elvárt és az aktuális fileok egy sorban eltérnek egymástól:

ERROR -> Errors in test in the following line(s):
-> line 1:
hello
bello
The map has been cleared, ready for new command!

8.1 Osztályok és metódusok tervei.

A pszeudokódban a változó neveket nem változtattuk meg így angolul szerepelnek. Ha egy objektum adott függvényére vagy metódusára hivatkozunk akkor az következő jelöléssel jelöltük: `objectInstance.method!`

8.1.1 AirFreshner

- **Felelősség**

Az osztály a légfrissítő tárgyat reprezentálja.

- **Ősosztályok**

Item

- **Interfészek**

Debuggable

- **Attribútumok**

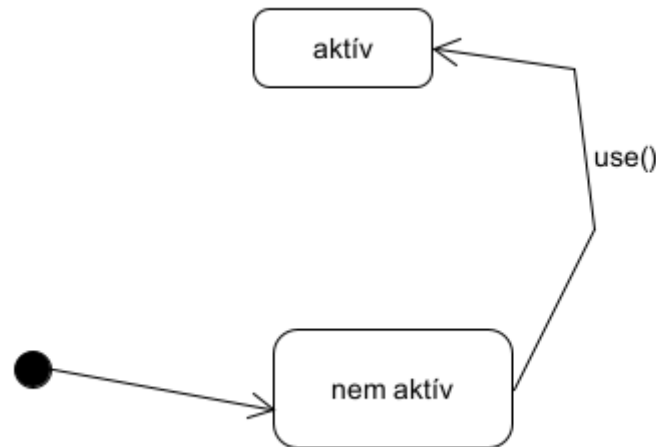
- **Metódusok**

- **+ AirFreshner(int AFID) : void** - Konstruktor, létrehoz egy AirFreshner objektumot a megadott ID-val. Beállítja a tartósságot 1-re. Elhelyezi a frissítőt a debugolható tárgyak közé.
- **+ use(Room r, Entity e) : boolean** - Ez a függvény a tárgy használatára való. Kitisztítja a szobát, ahonnan hívják. Amennyiben nincs felüldefiniálva a metódus az adott örökös tárgy osztályban akkor "false" értékkel tér vissza.
- **+ pickUp(Entity e) : boolean** - Visitor pattern alapján működik, meghívja a paraméter entitás felvevő függvényét a légfrissítővel. Igazzal tér vissza, ha sikeres a felvétel, hamissal ha sikertelen.
- **+ drop(Entity e) : boolean** - Visitor pattern alapján működik, meghívja a paraméter entitás eldobó függvényét a légfrissítővel. Igazzal tér vissza, ha sikeres az eldobás, hamissal ha sikertelen.
- **+ toString() : String** - Visszatér egy Stringgel ami a légfrissítő nevéből és az azonosítójából áll
- **+ debug() : String** - Visszaadja az objektum szöveges reprezentációját, debug célra használandó.

8.1.2 Beer

- **Felelősség**

Az osztály a sör tárgyat reprezentálja.



- **Össztályok**

Item

- **Interfészek**

Debuggable, Decaying

- **Attribútumok**

-isActive : boolean - A sör állapotát tárolja. Logikai igaz ha aktív, hamis ha inaktív.

- **Metódusok**

- **+Beer(int aID)** - Konstruktor, a megadott ID-val létrehoz egy új sört, majd hozzáadja a debug-olható objektumok gyűjteményéhez.
- **+getIsActive() : boolean** - Visszaadja az isActive tagváltozó értékét.
- **+setIsActive(boolean b)** - Beállítja az isActive tagváltozó értékét b paraméter- re.
- **+use(Room r, Entity b) : boolean** - A függvény célja, hogy aktivizálja a sört. r - A szoba, ahol használva lesz a tárgy. e - Az Entity, aki használja. Ha a sör már aktív hamisat ad vissza, amennyiben nem, aktiválja, és ezután logikai igazzal tér vissza.

Pszeudokód:

```

Függvény use(Room r, Entity b)
    ha isActive == igaz
        visszatérés hamis
    vége
    setIsActive(igaz)
    visszatérés igaz
    
```

Függvény vége

- **+pickUp(Entity e) : boolean** - Visitor pattern alapján működik, meghívja a paraméter entitás felvevő függvényét a sörrel. Igazzal tér vissza, ha sikeres a felvétel, hamissal ha sikertelen.
- **+drop(Entity e)** - Visitor pattern alapján működik, meghívja a paraméter entitás eldobó függvényét a sörrel. Igazzal tér vissza, ha sikeres a dobás, hamissal ha

sikertelen, vagy aktív sört dobunk el. Pszeudokód:

```
Függvény drop(Entity e)
    boolean dropSuccess = hamis
    ha getIsActive() == igaz
        e.removeItem(sör példány)
        GameController.getInstance().removeDecayingItems(sör példány)
    vége
    különben
        dropSuccess = e.dropItem(sör példány)
    vége
    visszatérés dropSuccess
```

Függvény vége

- **+age()** - Ez a függvény öregíti a tárgyat, ha aktív és meghívják. Az öregítés a következő módon történik: a durability tagváltozó értéke 1-el csökken.
- **+save(Entity ent) : boolean** - *Megpróbálja megmenteni a diák életét(megis menti ha aktív és van benne töltet). ent: Megmentendő diák. Ha sikerül "true" értékkel tér vissza, ellenkező esetben "false"-al. Pszeudokód:*

```
Függvény save(Entity ent)
    ha isActive és getDurability() > 0
        ent.saveDrop()
        visszatérés igaz
    vége
    visszatérés hamis
```

Függvény vége

- **+ toString() : String** - Visszatér egy Stringgel ami a sör nevéből és az azonosítójából áll
- **+ debug() : String** - Visszaadja az objektum szöveges reprezentációját, debug célra használandó.

8.1.3 Camembert

- **Felelősség**

A Camembert Item-mel végzett műveleteket kezelő osztály.

- **Ősosztályok**

Item osztály

- **Interfészek**

Debuggable.

- **Attribútumok**

Nincs.

- **Metódusok**

- **+Camembert(int aID):** Az osztály konstruktora, beállítja a tárgy ID-ját.

pszeudokód:

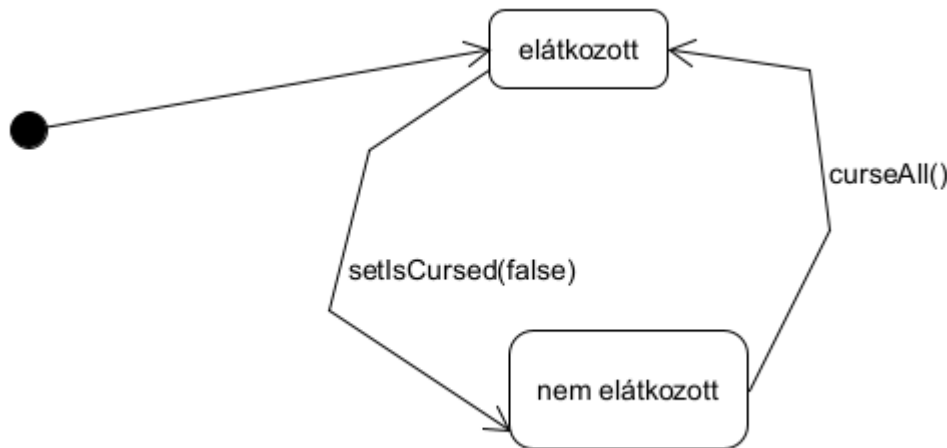
```
Függvény Camembert(int aID)
    setDurability(1)
    GameController.getInstance().debuggableObject.put(aID, Camembert
példány)
Függvény vége
```

- **+boolean use(Room r, Entity e):** A függvény célja, hogy a Camembertet használva elgázosítja a szobát.
 pszeudokód:
 Függvény use(Room r, Entity e)
 r.makeToxic()
 e.removeItem(Camembert példány)
 visszatérés igaz
 Függvény vége
- **+boolean pickUp(Entity e):** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity “pickUpItem” rá vonatkozó változtatát. (visitor pattern)
 pszeudokód:
 Függvény pickup(Entity e)
 visszatérés e.pickupItem(Camembert példány)
 Függvény vége
- **+boolean drop(Entity e):** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity “dropItem” rá vonatkozó változtatát. (visitor pattern)
 pszeudokód:
 Függvény drop(Entity e)
 visszatérés e.dropItem(Camembert példány)
 Függvény vége
- **+String toString():** Visszatér egy Stringgel ami a maga névből és az azonosítójából áll
 pszeudokód:
 Függvény toString()
 visszatérés “Camembert: #” + Integer.toString(getID())
 Függvény vége
- **+String debug():** Debug szöveg generálása.
 pszeudokód:
 Függvény debug()
 visszatérés “---- Camembert “ + Camembert példány.getID() + “
 --\ndurability : “ + getDurability() + “\n-- Camembert “ + Camembert
 példány.getID + “ -- \n”
 Függvény vége

8.1.4 Door

- Felelősség

Az osztály a játékbeli ajtót reprezentálja



- Össztályok

- Interfészek

Debuggable
Serializable

- Attribútumok

- **+ID : int** - Az ajtó egyedi azonosítója
- **+Direction : enum[LEFT,RIGHT,BOTH]** - Az ajtó irányát tároló enum értékek lehetséges változatai.
- **- isCursed : boolean** - Megadja hogy az ajtó elátkozott-e.
- **- availableRooms : Room[]** - Az ajtón keresztül elérhető szobák
- **- direction : Direction** - Direction enum szerinti irány

- Metódusok

- **+getID() : int** - Az ajtó egyedi azonosítóját adja vissza.
- **+setID(ind id)** - Az ajtó egyedi azonosítóját állítja be a paraméter id értékére.
- **+getIsCursed() : boolean** - Visszaadja hogy az ajtó elátkozott-e.
- **+setIsCursed(boolean c)** - Az isCursed setter függvénye. c - erre az értékre állítja az isCursed értéket.
- **+getRoomOne()** - Az ajtó 1-es számú szobájának getter függvénye, visszatér az első ajtót.
- **+getRoomTwo()** - Az ajtó 2-es számú szobájának getter függvénye, visszatér a második ajtót.
- **+setRoomOne(Room newR)** - Az ajtó 1-es számú szobájának setter függvénye. newR - ezt a szobát állítja be az egyes számú szobának.
- **+setRoomTwo(Room newR)** - Az ajtó 2-es számú szobájának setter függvénye. newR - ezt a szobát állítja be az kettes számú szobának.
- **+getDirection() : Direction** - Visszaadja az ajtó irányát.

- **+setDirection(Direction d)** - Beállítja az ajtó irányát a d értékre.
- **+Door(Room r1, Room r2, boolean cursed, Direction d, int dID)** - létrehoz egy új ajtót a megadott paraméterekkel. Ezután a debugolható objektumok közé felveszi az új ajtót. Pszeudokód:

```
Függvény Door(Room r1, Room r2, boolean cursed, Direction d, int dID)
    availableRooms = új Room[2]
    availableRooms[0] = r1
    availableRooms[1] = r2
    isCursed = cursed
    direction = d
    ID = dID
    GameController.getInstance.debuggableObjects-be belerakása (ID, ajtó
példány)
Függvény vége
```

- **+moveEntity(Entity e, Room curr) : boolean** - Az Entity tovább haladását bonyolítja le az ajtón keresztül. ent - továbbhaladni akaró Entity. curr - Entity jelenlegi szobája. A visszatérés értéke az ajtón való áthaladás sikeressége. Ha az ajtó elátkozott, a függvény hamis értékkel tér vissza. Ellenkező esetben: Pszeudokód:

```
Függvény moveEntity(Entity ent, Room curr)
    boolean res = hamis
    ha isCursed != igaz
        ha curr == availableRooms[0]
            ha direction == Direction.RIGHT vagy direction ==
Direction.BOTH
                res = availableRooms[1].acceptEntity(ent)
            vége
        különben
            GameController.getInstance().errorMsg(303, ID)
        vége
    vége
    különben ha curr == availableRooms[1]
        ha direction == Direction.LEFT vagy direction ==
direction.BOTH
            res = availableRooms[0].acceptEntity(ent)
        vége
        különben
            GameController.getInstance().errorMsg(303, ID)
        vége
    vége
    különben
        visszatérés hamis
    vége
    különben GameController.getInstance().errorMsg(303, ID)
    vége
    visszatérés res
```

Függvény vége

- **+ toString() : String** - Visszatér egy Stringgel ami a az ajtó nevéből és az azonosítójából áll
- **+ getCurrentDirection(Room r) : String** - Visszatér egy Stringgel ami azt tartalmazza, hogy melyik szobákat köti össze. Megadunk neki egy szobát, ez alapján tudja hogy a másik szobát kell visszaadnia. room - a megadott szoba. Visszatérés: a másik szoba Stringként, ID-vel együtt.

Pszeudokód:

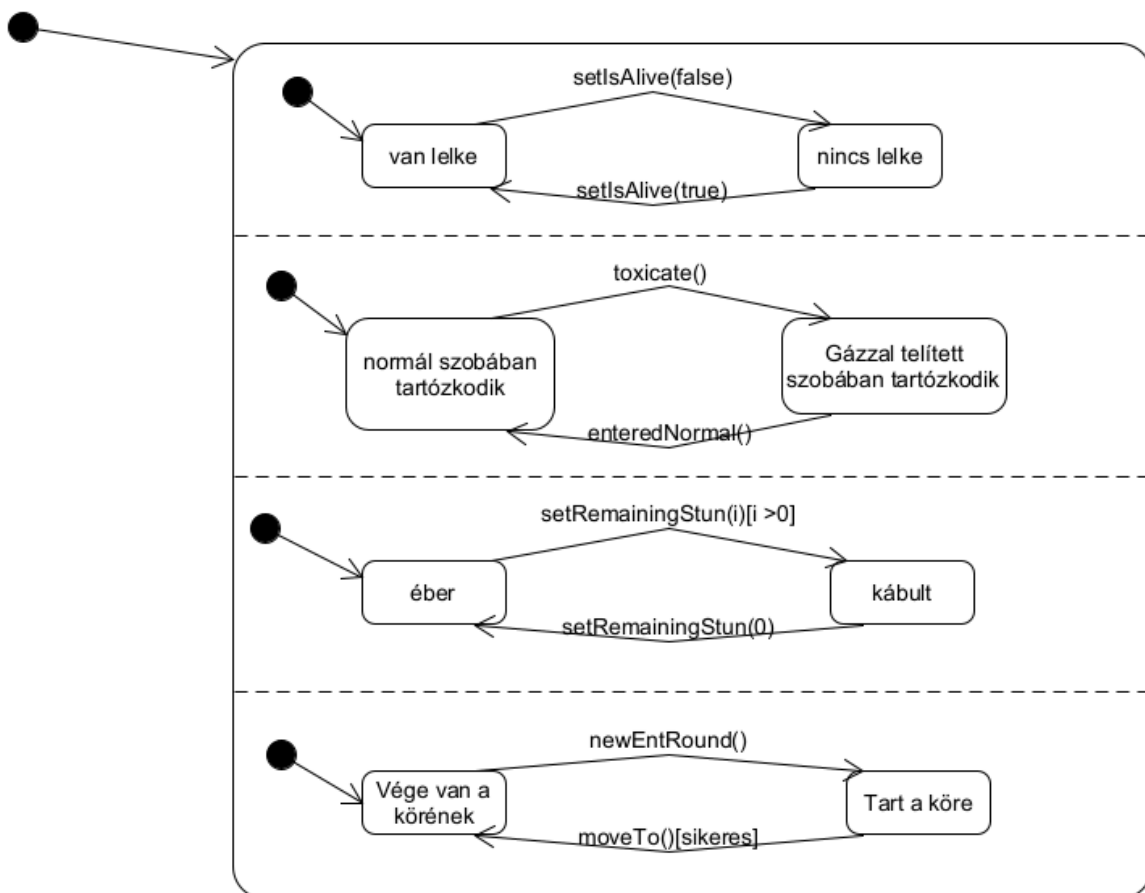
```
Függvény getCurrentDirection(Room room)
    Room leadingTo
    ha availableRooms[0] == room
        leadingTo = availableRooms[1]
    vége
    különben
        leadingTo = availableRooms[0]
    vége
    visszatérés "Room#" + Sztring(leadingTo.getID())
Függvény vége
```

- **+containsRoom(Room r1) : boolean** - Ajtó tartalmazását ellenőrző függvény. r1 - az ajtó amit keresünk. Ha az ajtó utat biztosít az adott ajtó felé akkor "true" értékkel tér vissza. ellenkező esetben hamis.
- **+neighbourRoom(Room curr) : Room** - Megnézi hogy melyik szobája a curr és vissza adja a másikat. curr - szoba aminek a párját keressük. Visszatérés: pár.
- **+String debug()**: Debug szöveg generálása.

8.1.5 Entity

- **Felelősség**

A szobák között mozogni, tárgyakat felvenni, illetve egyéb interakciókat mind a játéktér mind egymás között végezni képes entitások absztrakt őssztálya.



- **Őssztályok**

Nem rendelkezik őssztállyal.

- **Interfészek**

Megvalósítja a Serializable valamint Debuggable interfészt.

- **Attribútumok**

- **#name: String** Az Entity neve.
- **#ID: int** Az Entity egyedi azonosítója.

- **#remainingStun: int** Számontartja, hogy az Entity hány körből marad még ki.
- **#currentRoom: Room** A szoba ahol az Entity jelenleg tartózkodik.
- **#isAlive: boolean** Entity lélekkel rendelkezésének logikai értéke.
- **#isInToxic: boolean** Az Entity gázzal telített szobában tartózkodásának logikai értéke
- **#inventory: ArrayList<Item>** Az Entity hátizsákja
- **#roundIsOver: boolean** Logikai értéke annak, hogy az Entity saját köre befejeződött-e
- **Metódusok**
 - **+Entity(String, int, Room)** Az Entity konstruktora. Argumentumként vár egy stringet, amely az Entity neveként, egy egész számot, amely az Entity egyedi azonosítójaként, valamint egy Room referenciát, amely az Entity jelenlegi tartózkodási pozíciójaként kerül beállításra.
 - **+getName(): String** Az Entity nevét teszi kívülről lekérdezhetővé, meghívásakor visszatér annak értékével.
 - **+getID(): int** Az Entity egyedi azonosítóját teszi kívülről lekérdezhetővé, meghívásakor visszatér annak értékével.
 - **+getRemainingStun(): int** Az Entity még letöltendő körkimaradásainak számát teszi kívülről lekérdezhetővé, meghívásakor visszatér annak értékével.
 - **+getCurrentRoom(): Room** Az Entity jelenlegi tartózkodási szobáját teszi kívülről lekérdezhetővé, meghívásakor visszatér annak értékével.
 - **+getIsAlive(): boolean** Az Entity lélekkel rendelkezésének logikai értékét teszi kívülről lekérdezhetővé, meghívásakor visszatér annak értékével.
 - **+getIsInToxic(): boolean** Az Entity gázzal telített szobában tartózkodásának logikai értékét teszi kívülről lekérdezhetővé, meghívásakor visszatér annak értékével.
 - **+getInventory(): ArrayList<Item>** Az Entity hátizsákját teszi kívülről lekérdezhetővé, meghívásakor visszatér annak értékével.
 - **+setName(String): void** Az Entity nevét teszi kívülről beállíthatóvá. A metódus argumentuma az Entity új neve, ami a meghívást követően beállításra kerül.
 - **+setID(int): void** Az Entity egyedi azonosítóját teszi kívülről beállíthatóvá. A metódus argumentuma az Entity új azonosítója, ami a meghívást követően beállításra kerül.
 - **+setRemainingStun(int): void** Az Entity még letöltendő körkimaradásainak számát teszi kívülről beállíthatóvá. A metódus argumentuma az Entity kimaradásainak száma, ami a meghívást követően beállításra kerül.
 - **+setCurrentRoom(Room): void** Az Entity jelenlegi tartózkodási szobáját teszi kívülről beállíthatóvá. A metódus argumentuma az Entity tartózkodási szobájának referenciája, ami a meghívást követően beállításra kerül.
 - **+setIsAlive(boolean): void** Az Entity lélekkel rendelkezését teszi kívülről beállíthatóvá. A metódus argumentuma az Entity lélekkel rendelkezésének új logikai értéke, ami a meghívást követően beállításra kerül.
 - **+setIsInToxic(boolean): void** Az Entity gázzal telített szobában tartózkodásának logikai értékét teszi kívülről beállíthatóvá. A metódus argumentuma az Entity gázzal telített szobában tartózkodásának új logikai értéke, ami a meghívást követően beállításra kerül.

- **+setInventory(List<Item>): void** z Entity hátizsákját teszi kívülről beállíthatóvá. A metódus argumentuma az Entity új hátizsákja egy listával reprezentálva, ami a meghívást követően beállításra kerül.
- **+{abstract} pickUpItem(SlideRule): boolean** Logarléc felvételét kezelő metódus. Absztrakt, mivel a különböző karakterek a tárgy felvételét különböző módon bonyolítják le. Argumentuma a felvétel alanyául szolgáló Logarléc referenciája. Visszatérése a felvétel sikerességének logikai értéke.
- **+{abstract} pickUpItem(Transistor): boolean** Tranzisztor felvételét kezelő metódus. Absztrakt, mivel a különböző karakterek a tárgy felvételét különböző módon bonyolítják le. Argumentuma a felvétel alanyául szolgáló Tranzisztor referenciája. Visszatérése a felvétel sikerességének logikai értéke.
- **+pickUpItem(Rag): boolean** Rongy tárgy felvételét kezelő metódus. Argumentuma a felvétel alanyául szolgáló rongy referenciája. Visszatérése a felvétel sikerességének logikai értéke. Ellenőrzi, hogy a rongy befér-e az Entity hátizsákjába. Ha nem, visszatér hamis értékkel. Ha igen meghívja a tartózkodási szobájának removeItem metódusát a rongy referenciájával. Ha ez egy null referenciával tért vissza akkor a tárgy nem vehető fel a szobából tehát a visszatérési érték ismét hamis. Ellenkező esetben a rongy hozzáadódik az Entity hátizsákjának tartalmához, továbbá a GameController decayingItem listájához, hiszen a rongy körönként veszít tartósságából felvétele után. Végül aktív állapotúvá változtatja a rongyot. Ezek után a felvétel sikeresen megtörtént, tehát a visszatérési érték igaz. Pszeudokód:

```
Függvény pickupItem(Rag rag):
    Ha canPickup() == hamis akkor visszatérés hamis
    Item it = currentRoom.removeItem(rag)
    Ha it == null akkor visszatérés hamis
    addItem(it)
    GameController.getInstance().addDecayingItem(it)
    it.setActive(true)
    Visszatérés igaz
```

Függvény vége

- **+pickUpItem(Beer): boolean** Sör tárgy felvételét kezelő metódus. Argumentuma a felvétel alanyául szolgáló sör referenciája. Visszatérése a felvétel sikerességének logikai értéke. Ellenőrzi, hogy a sör befér-e az Entity hátizsákjába. Ha nem, visszatér hamis értékkel. Ha igen meghívja a tartózkodási szobájának removeItem metódusát a rongy referenciájával. Ha ez egy null referenciával tért vissza akkor a tárgy nem vehető fel a szobából tehát a visszatérési érték ismét hamis. Ellenkező esetben a sör hozzáadódik az Entity hátizsákjának tartalmához úgy, hogy az az első helyre kerüljön. Ezek után a felvétel sikeresen megtörtént, tehát a visszatérési érték igaz. Pszeudokód:

```
Függvény pickupItem(Beer beer):
    Ha canPickup() == hamis akkor visszatérés hamis
    Item beerRemoved = currentRoom.removeItem(beer)
    Ha beerRemoved == null akkor visszatérés hamis
    inventory[0] = beerRemoved
    Visszatérés igaz
```

Függvény vége

- **+pickUpItem(Item): boolean** Általános tárgy felvételt kezelő metódus. Argumentuma a felvétel alanyául szolgáló tárgy referenciája. Visszatérése a felvétel sikerességének logikai értéke. Ellenőrzi, hogy a tárgy befér-e az Entity hátizsákjába. Ha nem, visszatér hamis értékkel. Ha igen meghívja a tartózkodási szobájának

removeItem metódusát a tárgy referenciájával. Ha ez egy null referenciával tért vissza akkor a tárgy nem vehető fel a szobából tehát a visszatérési érték ismét hamis. Ellenkező esetben a tárgy hozzáadódik az Entity hátizsákjának tartalmához. Ezek után a felvétel sikeresen megtörtént, tehát a visszatérési érték igaz. Pszeudokód:

```
Függvény pickupItem(Item item):
    Ha canPickup() == hamis akkor visszatérés hamis
    Item it = currentRoom.removeItem(item)
    Ha it == null akkor visszatérés hamis
    additem(it)
    Visszatérés igaz
Függvény vége
```

- **+meet(Student): void** Egy Entity és egy diák találkozását kezelő virtuális függvény. Mivel a találkozásnak az entityből leszármazott osztály típusától függően különböző műveletei vannak, így ott felül kell definiálni. Ha ez nem történik meg a találkozás művelet nélkül zajlik. A függvény törzse üres.
- **+meet(Teacher): void** Egy Entity és egy tanár találkozását kezelő virtuális függvény. Mivel a találkozásnak az entityből leszármazott osztály típusától függően különböző műveletei vannak, így ott felül kell definiálni. Ha ez nem történik meg a találkozás művelet nélkül zajlik. A függvény törzse üres.
- **+meet(Janitor): void** Egy Entity és egy takarító találkozását kezelő virtuális függvény. Mivel a találkozásnak az entityből leszármazott osztály típusától függően különböző műveletei vannak, így ott felül kell definiálni. Ha ez nem történik meg a találkozás művelet nélkül zajlik. A függvény törzse üres.
- **+meetMe(Entity): void** Két Entity találkozását kezelő függvény. Célja, hogy a találkozásnak megfelelő függvény hívódjon a származtatott osztályokban.(Visitor) Ehhez a paraméterül kapott entity-re meghívja a saját maga referenciájával a meet metódust. A metódus argumentuma az entity referenciája akivel a találkozás történik.
- **+moveTo(Door): boolean** Kezeli az Entity pozícióváltását a szobák között ajtókon keresztül. Ha a kapott ajtó referencia null értékű, akkor az Entity a jelenlegi tartózkodási szobájában szeretne maradni a következő körben is. Ebben az esetben a saját szobája acceptEntity metódusát hívja saját maga referenciájával. Ennek visszatérési értékét állítja be a roundIsOver értékének. Ha egy tényleges ajtó referencia érkezett paraméterül akkor ellenőrzi, hogy az Entity képes-e a helyváltásra, azaz ellenőrzi, hogy a remainingStun értéke egyezik-e nullával. Ha igen, meghívja annak moveEntity() függvényét, és visszatérési értékét beállítja a roundIsOver értékének majd ebből a metódussal is ezzel az értékkel tér vissza. Ellenkező esetben a visszatérési érték hamis. Pszeudokód:

```
Függvény moveTo(Door door):
    Ha door == null akkor:
        roundIsOver = currentRoom.acceptEntity(this)
    Különben:
        Ha remainingStun == 0 akkor:
            roundIsOver = door.moveEntity(this, currentRoom)
        Különben visszatérés hamis
        ha vége
    ha vége

    Visszatérés roundIsOver
Függvény vége
```

- **+dropItem(Item): boolean** Tárgyak eldobását kezelő metódus. A paraméterül kapott tárgyat először eltávolítja az Entity hátizsákjából, majd hozzáadja a jelenlegi tartózkodási szobájának tárgy listájához. A visszatérési érték ezen a tranzakció sikerességének logikai értéke. Pszeudokód:

```
Függvény dropItem(Item item):
    removeItem(item)
    currentRoom.addItem(item)
    Visszatérés igaz
Függvény vége
```

- **+dropItem(Mask): boolean** Maszk eldobását kezelő metódus. Elsősorban ellenőrzi, hogy az Entity gázzal telített szobában tartózkodik-e. Ha igen az eldobás nem lehetséges ezért a visszatérési érték hamis. Ellenkező esetben a paraméterül maszkot először eltávolítja az Entity hátizsákjából, majd hozzáadja a jelenlegi tartózkodási szobájának tárgy listájához. A visszatérési érték ezen a tranzakció sikerességének logikai értéke. Pszeudokód:

```
Függvény dropItem(Mask mask):
    Ha isInToxic == igaz akkor:
        visszatérés hamis
    Különben:
        removeItem(mask)
        currentRoom.addItem(mask)
    ha vége

    Visszatérés igaz
Függvény vége
```

- **+toxicate(): void** Entity egy gázzal telített szobába lépésével járó állapotváltozásait kezeli. Ehhez az összes hátizsákjában lévő tárgyon meghívja a „preventGasStun()” metódust egészen addig ameddig az első „true” értékkel nem tér vissza, ekkor további művelet nélkül befejeződik a metódus futása. Ha egyetlen ilyen tárgy sincs akkor beállítja „remainingStun” értékét 3-ra, valamint az Entity hátizsákjából az összes tárgyat eldobja a „dropInventory()” meghívásával. Pszeudokód:

```
Függvény toxicate():
    Ciklus végig minden item-en az inventory-ban:
        res = item.preventGasStun()
        Ha res == igaz akkor:
            visszatérés igaz
    Ha vége
    Ciklus vége

    setIsIntoxic(true)
    dropInventory()
    setRemainingStun(3)
Függvény vége
```

- **+enteredNormal(): void** Entity egy normál szobába belépésével járó állapotváltozásokat kezeli. Beállítja az „isInToxic” tagváltozó értékét hamisra,

valamint az összes hátizsákjában lévő tárgyra meghívja a „carriedIntoNormalRoom()” metódust. Pszeudokód:

```
Függvény enteredNormal():
    setIsIntoxic(false)
    Ciklus végig minden item-en az inventory-ban:
        item.carriedIntoNormal()
    Ciklus vége
Függvény vége
```

- **+{abstract} newEntRound(): boolean** Absztrakt metódus, amely kezeli az adott Entity saját körében az általa végrehajtani kívánt cselekvéseket.
- **+roomChanged(Room): boolean** Az Entity jelenlegi pozíciójához tartozó szobából eltávolítja magát annak removeEntity() metódusának meghívásával. Ezt követően a pozícióját nyilvántartó currentRoom változó értéket a paraméterben kapott szoba értékére változtatja. Visszatérési értéke a tranzakció sikerességének logikai értéke. Pszeudokód:

```
Függvény roomChanged(Room room):
    currentRoom.removeEntity(this)
    setCurrentRoom(room)
    visszatérés igaz
Függvény vége
```

- **+encounterRag(): void** Az Entity egy rongyal találkozását kezelő virtuális metódus. Alapesetben a találkozás nem jár következménnyel ezért a metódus törzse üres. Ahol en nem így van, a leszármazottban felül kell definiálni.
- **+addItem(Item): void** A paraméterül kapott tárgyat az Entity hátizsákjában az utolsó szabad helyére teszi.
- **+removeItem(Item): void** A paraméterül kapott tárgyat eltávolítja az Entity hátizsákjából majd visszatér az eltávolított tárgy referenciájával.
- **+dropInventory(): void** Az Entity hátizsákjának teljes tartalmát megpróbálja eldobni úgy, hogy mindegyik tárgyon meghívja annak drop() metódusát az azt eldobó Entity referenciájával.

```
Függvény dropInventory():
    copy[] = inventory
    Ciklus végig minden item-en a copy-ban:
        item.drop(this)
    Ciklus vége
Függvény vége
```

- **+canPickUp(): boolean** Az Entity hátizsákját ellenőrzi, hogy az képes-e egy újabb tárgy befogadására. Ennek logikai értékével tér vissza.
- **+saveDrop(): void** Virtuális metódus annak támogatására, hogy ha egy Entity sörrel mentette meg a lelkét eldobja egy a hátizsákjában található tárgyat. Mivel nem minden Entityre vonatkozik, ezért a törzse üres. Ahol ez funkcionalitással rendelkezik felül kell definiálni.
- **+listItems(): String** Visszatérési Stringben az Entity összes tárgyat kilistázza. Pszeudokód:

```

Függvény listItems():
    message = ""
    section = ""
    Ciklus végig minden item-en az inventory-ban:
        section = item.toString()
        message += "\t-" + section + "\n"
    Ciklus vége
    visszatérés message
Függvény vége

```

- **+listTeleport(): String** Listába szedi az entity teleportálási lehetőségeit. Végignézi, hogy van-e tranzisztora, ha van akkor azon meghívja a getTeleport() függvényt. Ha nem üres a String amit az visszaad, akkor hozzáadja a listához. Ha üres nem történik semmi. Pszeudokód:

```

Függvény listTeleport():
    message = ""
    section = ""
    Ciklus végig minden item-en az inventory-ban:
        section = item.getTeleport()
        Ha section != "" akkor:
            message += "\t-" + section + "\n"
        Ha vége
    Ciklus vége
    visszatérés message
Függvény vége

```

- **+initiateUse(String[]): boolean** Az item terminálból hívott use parancsát támogató metódus. Az átadott szöveges tárgy ID alapján kikeresi az Entity hátizsákjából a tárgyat ezzel elvégezve az ID és referencia közötti feloldást. Ha ez sikeres meghívja a tárgyra annak use metódusát. Ha nincs ilyen ID-val rendelkező tárgy az Entitynél akkor meghívja a GameController megfelelő hiba kiíró metódusát. Visszatérési értéke a use lefutásának sikerességének logikai értéke. Pszeudokód:

```

Függvény initiateUse(String[] input):
    grabbedItem = null
    targetItemId = input[1]
    Ciklus végig minden item-en az inventory-ban:
        Ha item.checkID(targetItemId) akkor:
            grabbedItem = item
        Ha vége
    Ciklus vége
    Ha grabbedItem == null akkor:
        GameController.getInstance().errorMsg(208, input[1])
        visszatérés hamis
    Ha vége
    GameController.getInstance().statusMsg(213, targetItemId)
    visszatérés grabbedItem.Use(currentRoom, this)

```

Függvény vége

- **+initiateDrop(String[]): boolean** Az item terminálból hívott drop parancsát támogató metódus. Az átadott szöveges tárgy ID alapján kikeresi az Entity hátizsákjából a tárgyat ezzel elvégezve az Id és referencia közötti feloldást. Ha ez sikeres meghívja a tárgyra annak drop metódusát. Ha nincs ilyen id-val rendelkező tárgy az Entitynél akkor meghívja a GameController megfelelő hibakiíró metódusát. Visszatérési értéke a drop lefutásának sikerességének logikai értéke. Pszeudokód:

Függvény initiateDrop(String[] inp)

Ha inp.length == 3 és inp[2] == "-a" akkor:

dropInventory()

visszatérés igaz

Ha vége

grabbedItem = null

argetItemID = inp[1]

Ciklus végig minden itemen az inventory-ban):

Ha i.checkID(targetItemID) == igaz akkor:

grabbedItem = i

Ha vége

Ciklus vége

Ha grabbedItem == null akkor:

GameController.getInstance().errorMsg(204,inp[1])

Visszatérés hamis

Ha vége

GameController.getInstance().statusMsg(212, targetItemID)

Visszatérés grabbedItem.drop(this)

Függvény vége

- **+initiateConnect(String[]): boolean** Az itemek terminálból hívott connect parancsát támogató metódus. Az átadott szöveges tárgy ID alapján kikeresi az Entity hátizsákjából a tárgyakat ezzel elvégezve az Id és referencia közötti feloldást. Ha ez sikeres meghívja a tárgyra annak connect metódusát a másik referenciára feloldott tárggyal. Ha nincs ilyen id-val rendelkező tárgy az Entitynél akkor meghívja a GameController megfelelő hibakiíró metódusát. Visszatérési értéke a connect lefutásának sikerességének logikai értéke. Pszeudokód:

```
Függvény initiateConnect(String[] split)
    Ha split.length < 3 akkor:
        GameController.getInstance().errorMsg(0, "")
        visszatérés hamis
    Ha vége

    grabbedItem1 = null
    grabbedItem2 = null
    targetItemID1 = split[1]
    targetItemID2 = split[2]

    //Tárgy keresése inventory-ban
    Ciklus i 0-tól az inventory méretéig:
        Ha inventory.get(i).checkID(targetItemID1) == igaz akkor:
            grabbedItem1 = inventory.get(i)
            különben Ha inventory.get(i).checkID(targetItemID2) == igaz
akkor:
            grabbedItem2 = inventory.get(i)
        Ha vége
    Ciklus vége

    //ha a tárgy nem található a zsákban:
    Ha grabbedItem1 == null || grabbedItem2 == null akkor:
        GameController.getInstance().errorMsg(204, split[1] + ", " +
split[2]);
        Visszatérés hamis
    Ha vége

    GameController.getInstance().statusMsg(511, targetItemID1 + ", " +
targetItemID2)

    visszatérés grabbedItem1.connect(grabbedItem2, this);
Függvény vége
```

- **+initiateDisconnect(String[]): boolean** Az item terminálból hívott disconnect parancsát támogató metódus. Az átadott szöveges tárgy ID alapján kikeresi az Entity hátizsákjából a tárgyat ezzel elvégezve az Id és referencia közötti feloldást. Ha ez sikeres meghívja a tárgyra annak disconnect metódusát. Ha nincs ilyen id-val rendelkező tárgy az Entitynél akkor meghívja a GameController megfelelő hibakiíró

metódusát. Visszatérési értéke a disconnect lefutásának sikerességének logikai értéke.
Pszedokód:

Függvény initiateDisconnect(String[] split)

```

Ha split.length < 3 akkor:
    GameController.getInstance().errorMsg(0, "")
    visszatérés hamis
Ha vége

grabbedItem1 = null
targetItemID1 = split[1]

//Tárgy keresése inventory-ban
Ciklus i 0-tól az inventory méretéig:
    Ha inventory.get(i).checkID(targetItemID1) akkor:
        grabbedItem1 = inventory[i]
    Ha vége
Ciklus vége

//ha a tárgy nem található a zsákban:
Ha grabbedItem1 == null akkor:
    GameController.getInstance().errorMsg(204, split[1])
    visszatérés hamis
Ha vége

GameController.getInstance().statusMsg(512, "" + targetItemID1)
visszatérés grabbedItem1.disconnect(grabbedItem1.getPair())
Függvény vége

```

- **+debug(): String** Debug szöveg generálása formázva. A formázott szöveg a visszatérési érték. Pszeudokód:

Függvény debug()

```

retVal = "---- " + getName() + " " + getID() + " ----\nname : " +
name + "\nremainingStun : " + remainingStun + "\ncurrentRoom : " +
currentRoom.getID() + "\nisAlive : " + isAlive + "\nisInToxic : " +
isInToxic + "\nroundIsOver : " + roundIsOver + "\ninventory : \n"

```

Ciklus végig minden i tárgyon az inventory-ban):

```
retVal += i.debug()
```

Ciklus vége

```
retVal += "---- " + getName() + " " + getID() + " ----\n";
```

```
visszatérés retVal
```

Függvény vége

8.1.6 FakeMask

- **Felelősség**

A Mask Item hamis verzióját kezelő osztály.

- **Ősosztályok**

Item→Mask osztály

- **Interfészek**

Debuggable.

- **Attribútumok**

Nincs.

- **Metódusok**

- **+FakeMask(int fMID):** Az osztály konstruktora.
 pszeudokód:
 Függvény FakeMask(int fMID)
 GameController.getInstance().debuggableObjects.put(fMid, FakeMask
 példány)
 Függvény vége
- **+boolean pickUp(Entity e):** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity “pickUpItem” rá vonatkozó változtatát. (visitor pattern)
 pszeudokód:
 Függvény pickUp(Entity e)
 visszatérés e.pickupItem(FakeMask példány)
 Függvény vége
- **+boolean drop(Entity e):** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity “dropItem” rá vonatkozó változtatát. (visitor pattern)
 pszeudokód:
 Függvény drop(Entity e)
 visszatérés e.dropItem(FakeMask példány)
 Függvény vége
- **+boolean getLastRoomWasToxic():** Az eredeti függvény felüldefiniálása false visszatérésűre.
 pszeudokód
 Függvény getLastRoomWasToxic()
 visszatérés hamis
 Függvény vége
- **+boolean preventGasStun():** Az eredeti függvény felüldefiniálása false visszatérésűre.
 pszeudokód:
 Függvény preventGasStun()
 visszatérés hamis
 Függvény vége

- **+void carriedIntoNormalRoom():** Az eredeti függvény felüldefiniálása üres függvényre.
pszeudokód:
Függvény carriedIntoNormalRoom()

Függvény vége
- **+String toString():** Visszatér egy Stringgel ami a maga nevéből és az azonosítójából áll
pszeudokód
Függvény toString()
 visszatérés "Mask: #" + Integer.toString(getID())
Függvény vége
- **+String debug():** Debug szöveg generálása.
pszeudokód:
Függvény debug()
 visszatérés "---- FakeMask" + Fakemask példány.getID() + "
--\ndurability : " + getDurability() + "\n -- FakeMask " + FakeMask
példány.getID() + "----\n"
Függvény vége

8.1.7 FakeSlideRule

- **Felelősség**

Az osztály a hamis logarlécet modellezi.

- **Ősosztályok**

Item -> SlideRule -> FakeSlideRule

- **Interfészek**

Debuggable

- **Attribútumok**

- **Metódusok**

- **+ FakeSlideRule(int fsID)** - Konstruktor, beállítja az ID változót a kapott paraméterre.
- **+pickUp(Entity e) :boolean** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity "pickUpItem" rá vonatkozó változtatát. (visitor pattern)
- **+drop(Entity e) : boolean** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity "dropItem" rá vonatkozó változtatát. (visitor pattern)
- **+String toString():** Visszatér egy Stringgel ami a hamis logarléc nevéből és az azonosítójából áll.
- **+String debug():** Debug szöveg generálása.

8.1.8 FakeTVSZ

- **Felelősség**

Az osztály a hamis TVSZ-t modellezi.

- **Ősosztályok**

Item -> TVSZ -> FakeTVSZ

- **Interfészek**

Debuggable

- **Attribútumok**

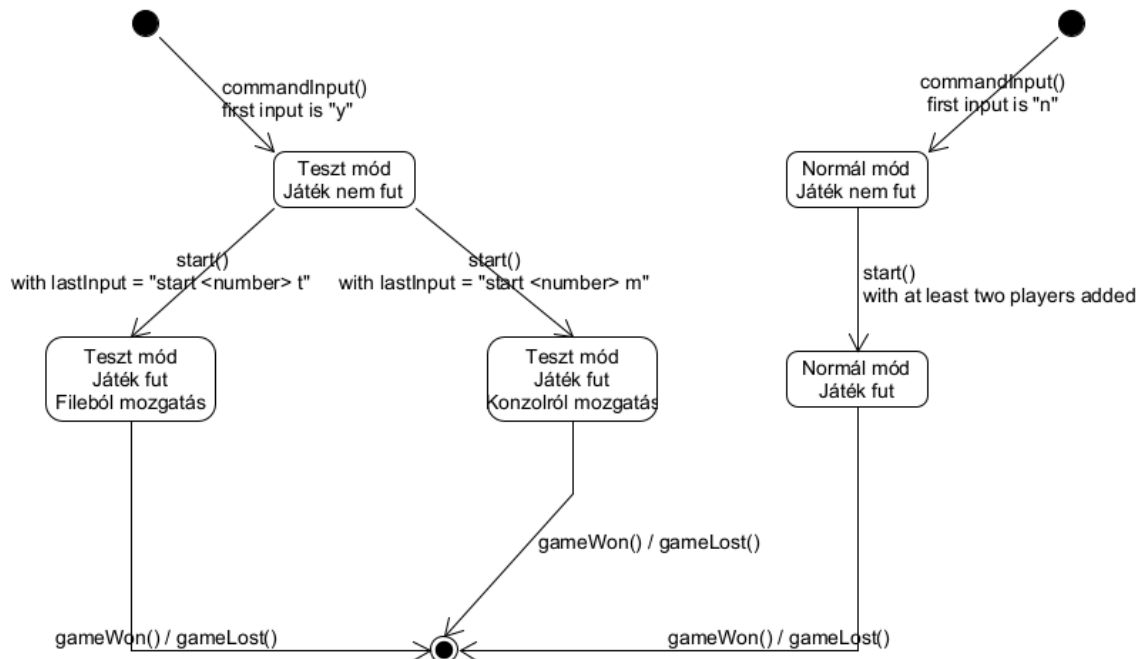
- **Metódusok**

- **+ FakeTVSZ(int aID)** - Konstruktor; beállítja az ID változót a kapott paraméterre.
- **+pickUp(Entity e) :boolean** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity "pickUpItem" rá vonatkozó változtatát. (visitor pattern)
- **+drop(Entity e) : boolean** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity "dropItem" rá vonatkozó változtatát. (visitor pattern)
- **+save(Entity ent) : boolean** - Az eredeti függvény felüldefiniálása false visszatérésűre, mivel hamis tárgyról van szó
- **+String toString():** Visszatér egy Stringgel ami a hamis TVSZ nevéből és az azonosítójából áll.
- **+String debug():** Debug szöveg generálása.

8.1.9 GameController

- **Felelősség**

Ez az osztály felel a bemeneti nyelv parancsainak megvalósításáért, a játék futtatásáért és a nyereségi, veszteségi kondíciók kezeléséért. Felelős ezen felül a hiba - és státusz üzenetek kiírásáért is.



- **Ősosztályok**

-

- **Interfészek**

Serializable

- **Attribútumok**

- **-instance: GameController:** Ez az attribútum felelős a GameController egyetlen megvalósításáért, és a Singleton design pattern megvalósításáért. Privát láthatóságú, így csak az osztályon belül, vagy a getInstance() függvény segítségével elérhető.
- **+debuggableObjects: HashMap<Integer, Debuggable>:** Az összes debuggolható objektumot, és annak az ID-jét tárolja egymás mellett.
- **+entities: ArrayList<Entity>:** Az összes a játékban résztvevő entitást tárolja.
- **+countOfPlayers: int:** Az összes a játékban résztvevő diák száma.
- **-playersWithoutSoul: int:** Az összes a játékban résztvevő, már halott diák száma.
- **+currentEntity: Entity:** Az éppen sorban lévő entitást tárolja el.
- **-decayingItems: ArrayList<Decaying>:** Az összes a játékban lévő, a Decaying interfészt megvalósító, és már romlásban lévő tárgyat tárolja.
- **-map: Map:** Az aktuális térképet tárolja el.

- **+menuCommands: HashMap<String, Runnable>:** A csak menüből meghívható parancsok nevét, és a parancs függvényét tárolja.
- **+gameCommands: HashMap<String, Runnable>:** A játékmenet során meghívható parancsok nevét, és a parancs függvényét tárolja.
- **+initCommands: HashMap<String, Runnable>:** A csak teszt inicializálása során meghívható parancsok nevét, és a parancs függvényét tárolja.
- **+lastInput: String:** A legutóbb beolvasott parancsot tárolja el.
- **+gameStarted: boolean:** Alap esetben 'false'. A játék elindítását figyeli, ha a játék elindult 'true'-ra vált.
- **+testMode: boolean:** Logikai érték arra, hogy a játék tesztelési módban fut-e.
- **+testInputs: ArrayList<String>:** Arra való, hogy amikor egy tesztet futtatunk, akkor a játék elindítása utáni parancsokat el tudjuk tárolni.
- **+nextTestCommand: int:** A következő tesztet parancsának a száma.
- **+manualInputMode: boolean:** Logikai érték arra, hogy a játékban kezelt entitásokat kézzel beírt, vagy fileből beolvasott parancsokkal akarjuk irányítani.
- **+testEntityPickupCount: int:** Ha a játék tesztelési módban fut, meghatározza, hogy egy körben egy a játék által irányított entitás hányszor próbál meg felvenni egy tárgyat a földről.
- **+testEntityMovementCount: int:** Ha a játék tesztelési módban fut, meghatározza, hogy egy körben egy a játék által irányított entitás hány ajtón próbál meg továbbhaladni.
- **+random: Random:** A játék véletlen elemeinek forrásaként szolgáló Random példány.
- **+scanner: Scanner:** A felhasználó általi input bekéréséért felelős Scanner objektum.
- **-remainingRounds: int:** A játékban hátralévő körök száma.
- **-globalID: int:** Private láthatóságú, a következő azonosító érték, amit egy diák a hozzáadásakor kaphat.
- **-manualMap: boolean:** Logikai érték annak eldöntésére, hogy a térképet a konzolon megadható parancsok segítségével készítjük-e el.
- **Metódusok**
 - **+addEntity(Entity nEntity): void:** Hozzáad egy entitást a GameController 'entities' listájához. Argumentuma a hozzáadandó entitás
 - **+getNextGlobalID(): int:** Visszatér egy integer számmal, ami épp a következő globalID.
 - **+addALostSoul(): void:** Eggyel növeli a GameController 'playersWithoutSoul' változójának értékét.
 - **+getRemainingRounds: int:** Visszatér a 'remainingRounds' változó értékével.
 - **+setRemainingRounds(int newRemainingVal): void:** Módosítja a kapott értékre a 'remainingRounds' változót. Argumentuma az új érték.

- **+getInsance(): GameController:** A GameController instance elérésért felel, a Singleton design pattern miatt. Ha nincs még instance, akkor készít egy újat.
Pszudokód:
Függvény getInstance()
 Ha instance == null
 instance = Új GameController()
 Visszatérés instance
Vége
- **+ageAll(): void:** Minden, a 'decayingItems' listában lévő tárgy 'age()' függvényét hívja meg, ezzel csökkentve eggyel azok tartósságát.
- **+addDecayingItem(Decaying newItem): void:** Hozzáad egy a 'Decaying' interfészt megvalósító tárgyat a 'decayingItems' listához. Argumentuma a hozzáadandó tárgy.
- **+removeDecayingItem(Decaying itemToRemove): void:** Kitörli a kapott 'Decaying' interfészt megvalósító tárgyat a 'decayingItems' listából. Argumentuma a kitörendő tárgy.
- **+clearMap(): void:** Visszaállítja a térkép, és a GameController változóit az indításkori állapotba, és létrehoz egy új Map objektumot.
Pszudokód:
Függvény clearMap()
 Törlés entities
 Törlés map.rooms
 countOfPlayers = 0
 Törlés decayingItems
 Törlés debuggableObjects
 remainingRounds = 0
 playersWithoutSoul = 0
 currentEntity = null
 map = Új Map(0)
Vége
- **+gameLost(int type): void:** Ha a játékosok elvesztették a játékot, akkor kiírja, hogy miért történt ez, és kilép a futtatásból. Argumentuma egy egész szám, amely a játék elvesztésének a fajtáját tükrözi.
Pszudokód:
Függvény gameLost(type)
 Kiír "Game Over!"
 message = (Ha type == 0) ? "[A tanárok elérték céljukhoz, minden diák elvesztette a lelkét!]" : "[A diákok nem találták meg időben a csúszkát.]"
 Kiír message
 Kilépés 0
Vége
- **+gameWon(): void:** A játék megnyerését kezelő metódus.
- **+newMergeState(Room r1, Room r2): int:** Kettő szoba összeolvadásakor hívódik meg. Visszatért egy számmal, amely az új szoba gázosságának felel meg. Argumentumai a kettő összeolvadó szoba.

- **+mergeDoors(Room r1, Room r2): void:** Kettő szoba ajtajainak a tömbjeit olvasztja össze az első szoba ajtajainak tömbjébe. Argumentumai a kettő összeolvadó szoba.

Pszudokód:

```
Függvény mergeDoors(r1, r2)
    Ciklus i = 0-tól r2.getDoors() méretéig
        tmpDoor = r2.getDoors()[i]
        Ha (tmpDoor.getRoomOne() == r1 és tmpDoor.getRoomTwo() == r2) vagy
        (tmpDoor.getRoomOne() == r2 és tmpDoor.getRoomTwo() == r1)
            r1.removeDoor(tmpDoor)
            r2.removeDoor(tmpDoor)
        Különben
            r1.addDoor(tmpDoor)
            r2.removeDoor(tmpDoor)
    Ciklus vége
Vége
```

- **+newRound(): void:** A játék 'nagy' köreinek kezeléséért felelős metódus. Kezeli a szobák elátkozódását, kettéválását és összeolvadását. Meghívja minden játékban lévő entitás newEntRound() függvényét.

Pszudokód:

```
Függvény newRound()
    countOfRooms = map.rooms.size()
    Ha countOfRooms >= 4
        map.curseMany(Véletlen szám 0 és (countOfRooms / 4) között)
        map.splitMany(Véletlen szám 0 és (countOfRooms / 4) között)
        map.mergeMany(Véletlen szám 0 és (countOfRooms / 4) között)
    map.tickRooms()
    Ciklus i = 0-tól entities méretéig
        currentEntity = entities[i]
        entities[i].newEntRound()
    Ciklus vége
    Ha countOfPlayers == playersWithoutSoul
        gameLost(0)
Vége
```

- **+runGame(): void:** A játék elindításáért felelős metódus, amely a 'start()' függvényen belül kell, hogy megtörténjen. Addig hívja körönként a 'newRound()' függvényt, ameddig a játékosok hátralévő köreinek száma el nem éri a nullát.

Pszudokód:

```
Függvény runGame()
    remainingRounds = 20
    Ciklus i = 0-tól remainingRounds-ig
        newRound()
    Ciklus vége
    gameLost(1)
Vége
```

- **+commandInput(): void:** A főmenütől egészen a játék indításáig kezeli a felhasználói parancsokat.

Pszudokód:

```
Függvény commandInput()
    mainMenuInput = null
    gameStarted = Hamis
    Kiír "Do you wish to run the game in test mode [y/n]?"
    flag = Hamis
    Ciklus amíg nem flag
        Kiír "[y/n]>"
        mainMenuInput = Új sor beolvasása a bemenetről
        Ha mainMenuInput egyenlő "y"
            testmode = Igaz
            manualMap = Igaz
            map = Új Map(0)
            flag = Igaz
        Ha mainMenuInput egyenlő "n"
            flag = Igaz
    Ciklus vége
    Ciklus amíg mainMenuInput nem egyenlő "exit"
        Kiír "Main menu>"
        mainMenuInput = Új sor beolvasása a bemenetről
        terminal(mainMenuInput)
    Ciklus vége
```

Vége

- **+testRun(Map map, ArrayList<Entity> entities, int countOfPlayers, int remainingRounds, int entityMovementTry, int entityPickupTry): void:** Lehetővé teszi a játék futtatását előre meghatározott körülmények mellett. Argumentumai a következők: Az előre elkészített térkép, a hozzáadott entitások listája, a játékosok száma, a hátralévő körök száma, a nem általunk irányított entitások mozgásra vonatkozó próbálkozásainak száma és a nem általunk irányított entitások tárgyfelvételre vonatkozó próbálkozásainak száma.

Pszudokód:

```
Függvény testRun(map, entities, countOfPlayers, remainingRounds,
entityMovementTry, entityPickupTry)
    GameController.getInstance().map = map
    GameController.getInstance().entities = entities
    GameController.getInstance().remainingRounds = remainingRounds
    testEntityMovementCount = entityMovementTry
    testEntityPickupCount = entityPickupTry
    testmode = Igaz
    gameStarted = Igaz
    GameController.getInstance().countOfPlayers = countOfPlayers
    playersWithoutSoul = 0
    runGame()
```

Vége

- **+terminal(String input): void:** Kezeli a játék összes inputjának végrehajtását. Argumentuma a végrehajtandó parancs teljes sora.

Pszeudokód:

```
Függvény terminal(input)
    tokenizedInput = input szétválasztása
    commandsToUse = null
    Ha gameStarted akkor
        commandsToUse = gameCommands
    Egyébként Ha testmode akkor
        commandsToUse = initCommands
    Egyébként
        commandsToUse = menuCommands
    Ha tokenizedInput hossza nagyobb mint 0
        isExistentCommand = tokenizedInput[0] a gameCommands-ben vagy
a menuCommands-ben vagy az initCommands-ben
        isUsableCommand = tokenizedInput[0] a commandsToUse-ban
    Ha isExistentCommand és isUsableCommand
        lastInput = input
        tokenizedInput[0] futtatása
    Különben Ha isExistentCommand és nem isUsableCommand
        errorMsg(1, "")
    Egyébként Ha nem isExistentCommand
        errorMsg(0, "")
```

Vége

- **+exit(): void:** A játékból való kilépést kezelő függvény.
- **+debug(): void:** A játék állásának ellenőrzését teszi lehetővé egy szöveges kimeneten keresztül. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor a 'debug.txt' fileba kiírja az összes debugolandó objektum állapotát.

Pszeudokód:

```
Függvény debug()
    splitInput = lastInput szétválasztva
    outMessage = ""
    Ciklus i = 1-től splitInput hossz-ig
        Ha debuggableObjects nem tartalmazza splitInput[i]-t
            errorMsg(405, splitInput[i])
        Visszatérés
    Egyébként
        outMessage += debuggableObjects[splitInput[i]].debug()]
    Ciklus vége
    outMessage Fájlbba írása
    Ha nem sikerült
        errorMsg(406, "")
```

Vége

- **+load(): void:** Egy játékmenet betöltését megvalósító metódus. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor beállítja az összes fontos objektum állapotát a fileből olvasottra, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszudokód:

Függvény load()

```

    splitInput = lastInput szétválasztása
    Ha splitInput hossza kisebb mint 2
        errorMsg(402, "")
        Visszatérés
    Ha splitInput hossza nagyobb mint 2
        errorMsg(0, "")
        Visszatérés
    Ha gameStarted
        errorMsg(1, "")
        Visszatérés
    fileName = splitInput[1] + ".ser"
    in = fileName Fájl beolvasása
    entities = in Következő objektum olvasása
    countOfPlayers = in Következő objektum olvasása
    playersWithoutSoul = in Következő objektum olvasása
    currentEntity = in Következő objektum olvasása
    decayingItems = in Következő objektum olvasása
    map = in Következő objektum olvasása
    gameStarted = Igaz
    statusMsg(412, splitInput[1])
    runGame()
    Ha hiba olvasás közben
        errorMsg(404, splitInput[1])

```

Vége

- **+save(): void:** A játék aktuális állásának elmentését teszi lehetővé. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor szerializáció segítségével elmenti a játék szempontjából fontos objektumokat, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszudokód:

Függvény save()

```

    fileName = "mentes.ser"
    splitInput = lastInput szétválasztása
    Ha splitInput hossza nagyobb mint 2
        errorMsg(0, "")
        Visszatérés
    Ha nem gameStarted
        errorMsg(1, "")
        Visszatérés
    Ha splitInput hossza egyenlő 2
        fileName = splitInput[1] + ".ser"
    out = fileName Fájlba kiírása
    entities Fájlba kiírása
    countOfPlayers Fájlba kiírása
    playersWithoutSoul Fájlba kiírása
    currentEntity Fájlba kiírása
    decayingItems Fájlba kiírása
    map Fájlba kiírása

```

```

out.close()
statusMsg(414, "")
Ha hiba fájlkezelés közben
    errorMsg(403, "")

```

Vége

- **+execute(): void:** Lehetőséget ad egy '.txt' file betöltésére, melyből sorokat olvas be. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor soronként beolvassa a file-t, és minden sort átad a 'terminal()' függvénynek, amely elvégzi a kapott parancsot, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```

Függvény execute()
    splitInput = lastInput szétválasztása
    Ha splitInput hossza kisebb mint 2
        errorMsg(402, "")
        Visszatérés
    Ha splitInput hossza nagyobb mint 2
        errorMsg(0, "")
        Visszatérés
    Ha gameStarted
        errorMsg(1, "")
        Visszatérés
    fileName = splitInput[1] + ".txt"
    reader = Új olvasó a fileName fájlban
    line = reader Új sor olvasása
    statusMsg(412, splitInput[1])
    Amíg line nem egyenlő null
        terminal(line)
        line = reader.readLine()
    reader bezárása
    Ha hiba beolvasás közben
        errorMsg(404, splitInput[1])

```

Vége

- **+taddroom(): void:** Teszt módban futtatott játékmenet esetén lehetőséget ad egy szoba hozzáadására. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor egy szobát ad a térképhez, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```

Függvény taddroom()
    splitInput = lastInput szétválasztása
    Ha nem manualMap
        errorMsg(5, "")
        Visszatérés
    Ha splitInput hossza nem egyenlő 4
        errorMsg(0, "")
        Visszatérés
    Ha gameStarted
        errorMsg(1, "")
        Visszatérés
    Ha splitInput[1] nem egyenlő "t" és nem egyenlő "f"
        errorMsg(0, "")
        Visszatérés
    Különben
        newRoom = Új Room((splitInput[1] egyenlő "t"), splitInput[2],
splitInput[3])
        map.addRoom(newRoom)
        statusMsg(601, "")

```

```

Ha hiba
    errorMsg(0, "")
Visszatérés

```

Vége

- **+tmerge(): void:** Teszt módban futtatott játékmenet esetén lehetőséget ad kettő szoba összeolvasztására. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor összeolvasztja kettő specifikált szobát, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```

Függvény tmerge()
    splitInput = lastInput szétválasztása
    Ha nem manualMap
        errorMsg(5, "")
        Visszatérés
    Különben ha splitInput hossza nem egyenlő 3
        errorMsg(0, "")
        Visszatérés
    r1 = map.roomById(splitInput[1])
    r2 = map.roomById(splitInput[2])
    Ha r1 egyenlő null vagy r2 egyenlő null
        errorMsg(606, splitInput[1] + " or " + splitInput[2])
        Visszatérés
    Különben
        canMerge = r1.merge(r2)
        Ha nem lehet canMerge
            errorMsg(602, "")
            Visszatérés
        map.removeRoom(r2)
        r2 eltávolítása debuggableObjects-ből
        statusMsg(608, "")

```

Vége

- **+tsplit(): void:** Teszt módban futtatott játékmenet esetén lehetőséget ad egy szoba kettéválasztására. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor kettéválasztja a specifikált szobát, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```

Függvény tsplit()
    splitInput = lastInput szétválasztása
    Ha nem manualMap
        errorMsg(5, "")
        Visszatérés
    Különben ha splitInput hossza nem egyenlő 2
        errorMsg(0, "")
        Visszatérés
    r1 = map.roomById(splitInput[1])
    Ha r1 egyenlő null
        errorMsg(603, splitInput[1])
    Különben
        newR = r1.split()
        Ha newR egyenlő null
            errorMsg(603, newR.getID())
        newR hozzáadása map.rooms-hoz
        newR hozzáadása map.rooms-hoz
        statusMsg(609, "")

```

Vége

- **+tadddoor(): void:** Teszt módban futtatott játékmenet esetén lehetőséget ad egy ajtó létrehozására. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor létrehoz egy új szobát a megadott paramétereknek megfelelően, a kettő kapott szobát így összeköti, ezután pedig válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```
Függvény tadddoor()
    splitInput = lastInput szétválsztása
    Ha nem manualMap
        errorMsg(5, "")
        Visszatérés
    Különben ha splitInput hossza nem egyenlő 6
        errorMsg(0, "")
        Visszatérés
    Ha gameStarted
        errorMsg(1, "")
        Visszatérés
    Ha splitInput[3] egyenlő "t" vagy splitInput[3] egyenlő "f"
        Ha splitInput[4] egyenlő "r" vagy splitInput[4] egyenlő "l" vagy
        splitInput[4] egyenlő "b"
            r1 = map.roomById(splitInput[1])
            r2 = map.roomById(splitInput[2])
            Ha r1 egyenlő null
                errorMsg(601, splitInput[1])
                Visszatérés
            Különben ha r2 egyenlő null
                errorMsg(601, splitInput[2])
                Visszatérés
            dir
            Kapcsoló(splitInput[4])
                eset "r" :
                    dir = Door.Direction.RIGHT;
                    Kilép
                eset "l" :
                    dir = Door.Direction.LEFT;
                    Kilép
            Alapértelmezett :
                dir = Door.Direction.BOTH;
                Kilép
            Kapcsoló Vége
            newDoor = Új Door(r1, r2, splitInput[3] egyenlő "t", dir,
splitInput[5])
            r1.addDoor(newDoor)
            r2.addDoor(newDoor)
            statusMsg(606, "")
            Ha hiba a paraméterezésben
                errorMsg(0, "")
                Visszatérés
            Különben
                errorMsg(0, "")
                Visszatérés
            Különben
                errorMsg(0, "")
                Visszatérés
```

Vége

- **+tadditem(): void:** Teszt módban futtatott játékmenet esetén lehetőséget ad egy tárgy hozzáadására. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor létrehoz egy új a paramétereknek megfelelő tárgyat, és azt be is teszi a megkapott ID-ú szobába, ezután pedig válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```
Függvény tadditem()
    splitInput = lastInput szétválasztása
    Ha nem manualMap
        errorMsg(5, "")
        Visszatérés
    Különbön ha splitInput hossza nem egyenlő 4
        errorMsg(0, "")
        Visszatérés
    Ha gameStarted
        errorMsg(1, "")
        Visszatérés
    room = map.roomById(splitInput[3])
    Ha room egyenlő null
        errorMsg(601, splitInput[3])
        Visszatérés
    Kapcsoló(splitInput[1])
        eset "airfreshner" :
            item = Új AirFreshner(splitInput[2])
            room.addItem(item)
            statusMsg(602, "")
            Kilépés
        eset "beer" :
            item = Új Beer(splitInput[2])
            room.addItem(item)
            statusMsg(602, "")
            Kilépés
        eset "camembert" :
            item = Új Camembert(splitInput[2])
            room.addItem(item)
            statusMsg(602, "")
            Kilépés
        eset "fakemask" :
            item = Új FakeMask(splitInput[2])
            room.addItem(item)
            statusMsg(602, "")
            Kilépés
        eset "fakesliderule" :
            item = Új FakeSlideRule(splitInput[2])
            room.addItem(item)
            statusMsg(602, "")
            Kilépés
        eset "faketvsz" :
            item = Új FakeTVSZ(splitInput[2])
            room.addItem(item)
            statusMsg(602, "")
            Kilépés
        eset "mask" :
            item = Új Mask(splitInput[2])
            room.addItem(item)
            statusMsg(602, "")
            Kilépés
        eset "rag" :
```

```

        item = Új Rag(splitInput[2])
        room.addItem(item)
        statusMsg(602, "")
        Kilépés
    eset "sliderule" :
        item = Új SlideRule(splitInput[2])
        room.addItem(item)
        statusMsg(602, "")
        Kilépés
    eset "transistor" :
        item = Új Transistor(splitInput[2])
        room.addItem(item)
        statusMsg(602, "")
        Kilépés
    eset "tvosz" :
        item = Új TVSZ(splitInput[2])
        room.addItem(item)
        statusMsg(602, "")
        Kilépés
    Alapértelmezett :
        errorMsg(0, "")
        Visszatérés
Kapcsoló vége
Ha hiba a paraméterezéssel
    hibaÜzenet(0, "")
    Visszatérés

```

Vége

- **+taddentity(): void:** Teszt módban futtatott játékmenet esetén lehetőséget ad egy új entitás hozzáadására. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor létrehoz egy a kapott paramétereknek megfelelő új entitást, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```

Függvény taddentity()
    splitInput = lastInput szétválasztása
    Ha nem manualMap
        errorMsg(5, "")
        Visszatérés
    Különben ha splitInput hossza nem egyenlő 5
        errorMsg(0, "")
        Visszatérés
    Ha gameStarted
        errorMsg(1, "")
        Visszatérés
    room = map.roomById(splitInput[4])
    Ha room egyenlő null
        errorMsg(601, splitInput[3])
        Visszatérés
    Kapcsoló(splitInput[1])
        eset "t" :
            teacher = Új Teacher(splitInput[2],
splitInput[3], room)
            teacher hozzáadása az entites tömbhöz
            room.addEntity(teacher)
            statusMsg(604, "")
            Kilépés
        eset "j" : {

```

```

splitInput[3], room)      janitor      =      Új      Janitor(splitInput[2],
                           janitor hozzáadása az entites tömbhöz
                           room.addEntity(janitor)
                           statusMsg(605, "")
                           Kilépés
                           Alapértelmezett :
splitInput[3], room)      student      =      Új      Student(splitInput[2],
                           student hozzáadása az entites tömbhöz
                           room.addEntity(student)
                           countOfPlayers++
                           statusMsg(603, "")

Kapcsoló vége
Ha hiba a paraméterezéssel
    errorMsg(0, "")
    Visszatérés
Vége

```

- **+executetest(): void:** Teszt módban futtatott játékmenet esetén lehetőséget ad egy tesztet lefuttatására. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor beolvas inputokat egy megadott fileból, és azokat el is végezteti a programmal. Ezután ha minden parancsnak vége, a 'debug' parancs használatával kiírja a játék aktuális állapotát, és ezt az új filet összehasonlítja a kapott elvárt működést leíró fileal a 'fileCompare()' függvényt használva, és válaszol a felhasználónak a 'statusMsg()' függvény használatával. Mindezek után meghívja a 'clearMap()' függvényt, és lehetőséget ad újabb parancs fogadására.

Pszeudokód:

```

Függvény executetest()
    splitInput = lastInput szétválasztása
    Ha nem manualMap
        errorMsg(5, "")
        Visszatérés
    Különben ha splitInput hossza nem egyenlő 3
        errorMsg(0, "")
        Visszatérés
    Ha gameStarted
        errorMsg(1, "")
        Visszatérés
    clearMap()
    nextTestCommand = 0
    isStarted = Hamis
    fileName = splitInput[1] + ".txt"
    reader = Új olvasó a fileName fájlban
    line = reader Új sor olvasása
    lines[]
    statusMsg(412, splitInput[1])
    Amíg line nem egyenlő null
        Ha nem isStarted
            line hozzáadása a lines tömbhöz
        Különben
            line hozzáadása a testInputs tömbhöz
        Ha line tartalmazza "start"-ot
            isStarted = Igaz
        line = reader Új sor olvasása
    reader bezárása
    Ciklus i = 0-tól lines[] nagyságáig

```

```

        terminal(lines[i])
    Ciklus vége
    Ha fileCompare("debug.txt", splitInput[2] + ".txt")
        statusMsg(610, "")
    Különben
        reader = Új olvasó a "compareOutput.txt" fájlban
        line = reader Új sor olvasása
        Ha line egyenlő "Input files does not match in length
aborted."
            statusMsg(605, "")
        Különben
            statusMsg(610, "")
            Amíg line nem egyenlő null
                Kiírás(line)
                line = reader Új sor olvasása
    clearMap()
    statusMsg(607, "")
    Ha hiba van a fájlkezeléssel
        errorMsg(404, splitInput[1])
    Visszatérés
Vége

```

- **+nextTestLine(): String:** Visszatér a 'testInputs' listában lévő, soron következő paranccsal.
- **+addplayer(): void:** Normál módban futtatott játékmenet esetén lehetőséget ad egy új játékos hozzáadására. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor egy új játékos kerül bele a játékba. Ezután válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszudokód:

```

Függvény addplayer()
    split = lastInput szétválasztása
    Ha split hossza egyenlő 1
        errorMsg(0, "")
        Visszatérés
    Ha manualMap
        errorMsg(6, "")
        Visszatérés
    Ha gameStarted
        errorMsg(1, "")
        Visszatérés
    id = nextGlobalID()
    n = split[1]
    Ciklus i = 0-tól entities mérete - 1-ig
        Ha entities[i].getName() egyenlő n
            errorMsg(101, n)
            Visszatérés
    Ciklus vége
    student = Új Student(n, id, null)
    student hozzáadása az entities tömbhöz
    countOfPlayers++
    statusMsg(111, split[1])
Vége

```


- **+removeplayer(): void:** Normál módban futtatott játékmenet esetén lehetőséget ad egy játékos eltávolítására. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatakor eltávolítja a név alapján kapott játékost, majd válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```
Függvény removeplayer()
    split = lastInput szétválasztása
    Ha split hossza egyenlő 1
        errorMsg(0, "")
        Visszatérés
    Ha manualMap
        errorMsg(6, "")
        Visszatérés
    Ha gameStarted
        errorMsg(1, "")
        Visszatérés
    Ciklus i = 0-tól entities mérete - 1-ig
        Ha entities[i].getName() egyenlő split[1]
            entities.törlése(i)
            countOfPlayers--;
            statusMsg(112, split[1])
            Visszatérés
    Ciklus vége
    errorMsg(102, split[1])
```

Vége

- **+showplayers(): void:** Normál módban futtatott játékmenet esetén lehetőséget ad a már hozzáadott játékosok kilistázására. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor kiírja az összes hozzáadott játékost a 'statusMsg()' függvény használatával.

Pszeudokód:

```
Függvény showplayers()
    splitInput = lastInput szétválasztása
    Ha splitInput hossza nagyobb, mint 1
        errorMsg(0, "")
        Visszatérés
    Ha manualMap
        errorMsg(6, "")
        Visszatérés
    Ha gameStarted
        errorMsg(1, "")
        Visszatérés
    message = ""
    Ciklus i = 0-tól entities mérete - 1-ig
        section = entites[i].getName()
        message += "-" + section + "\n"
    Ciklus vége
    statusMsg(113, message)
```

Vége

- **+start(): void:** Megnézi, hogy teszt vagy normál módban fut a játék. Mindkettő esetben ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor az 'errorMsg()' függvény segítségével hibaüzenetet küld a felhasználónak. Ha normál módban fut a játék, akkor elkészíti a térképet, a 'buildMap()', 'fillWithItems()', és 'entityPlacer()' függvények segítségével, üzenetet küld a felhasználónak a 'statusMsg()' segítségével, és végül

elindítja a játékot a 'runGame()' függvénnyel. Ha teszt módban fut a játék, akkor erről küld üzenetet a felhasználónak a 'statusMsg()' használatával, és elindítja a játékot a 'testRun()' függvénnyel, annak átadva a megfelelő argumentumokat.

Pszeudokód:

```
Függvény start()
    Ha gameStarted
        errorMsg(1, "")
        Visszatérés
    Ha nem manualMap
        splitInput = lastInput szétválasztása
        Ha splitInput hossza nagyobb, mint 1
            errorMsg(0, "")
            Visszatérés
        Ha entities hossza kisebb, mint 2
            errorMsg(103, "")
            Visszatérés
        message = ""
        Ciklus i = 0-tól entities mérete - 1-ig
            section = entites[i].getName()
            message += "-" + section + "\n"
        Ciklus vége
        map = Új Map(0)
        map.buildMap(3, 4)
        map.fillWithItems(4, 2)
        map.entityPlacer(entities)
        gameStarted = Igaz
        playersWithoutSoul = 0
        testEntityMovementCount = 0
        testEntityPickupCount = 0
        testmode = Hamis
        statusMsg(114, message)
        runGame()
    Különben
        splitInput = lastInput szétválasztása
        Ha splitInput hossza nem egyenlő 3-mal
            errorMsg(0, "")
            Visszatérés
        Ha splitInput[2] egyenlő "t"-vel
            manualInputMode = Hamis
        Különben Ha splitInput[2] egyenlő "m"-mel
            manualInputMode = Igaz
        Különben
            errorMsg(0, "")
            Visszatérés
            gameStarted = Igaz
            playersWithoutSoul = 0
            testEntityMovementCount = 0
            testEntityPickupCount = 0
            testmode = Igaz
            statusMsg(115, "")
            testRun(map, entities, countOfPlayers,
szám(splitInput[1]), testEntityMovementCount, testEntityPickupCount)
        Ha paraméterezési hiba van
            errorMsg(0, "")
            Visszatérés
    Vége
```

- **+show(): void:** Lehetőséget ad az éppen soron következő entitás, és annak környezetéről információ kilistázásra. Ellenőrzi, hogy helyesen lett-e hívva, ha nem

akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor a megadott kapcsolónak megfelelő információkat összegyűjti, és a 'statusMsg()' függvény használatával azokat a felhasználónak meg is jeleníti..

Pszudokód:

```
Függvény show()
    splitInput = lastInput szétválasztása
    Ha splitInput hossza kisebb, mint 2
        errorMsg(0, "")
        Visszatérés
    Ha splitInput hossza nagyobb, mint 2
        errorMsg(3, "")
        Visszatérés
    Ha nem gameStarted
        errorMsg(1, "")
        Visszatérés
    message
    Kapcsoló(splitInput[1])
        eset "-i" :
            message = currentEntity.listItems()
            statusMsg(220, message)
            Kilépés
        eset "-d" :
            message = currentEntity.currentRoom.listDoors()
            statusMsg(221, message)
            Kilépés
        eset "-e" :
            message =
currentEntity.currentRoom.listEntities()
            statusMsg(222, message)
            Kilépés
        eset "-ri" :
            message = currentEntity.currentRoom.listItems()
            statusMsg(223, message)
            Kilépés
        eset "-t" :
            message = currentEntity.listTeleport()
            statusMsg(224, message)
            Kilépés
    alapértelmezett :
        errorMsg(4, "")
        Visszatérés
```

Vége

- **+pickup(): void:** Az éppen soron következő entitás tárgyfelvételi lehetőségét valósítja meg. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor meghívja az 'initiatePickup()' függvényt ami a tárgyfelvételt kezeli, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```
Függvény pickup()
    pickupAll = Hamis
    Ha nem gameStarted
        errorMsg(1, "")
        Visszatérés
    splitInput = lastInput szétválasztása
    Ha splitInput hossza nagyobb, mint 3 vagy kisebb, mint 2
        errorMsg(0, "")
        Visszatérés
    Ha splitInput hossza egyenlő 3-mal és splitInput[2] nem egyenlő
"-a"-val
        errorMsg(4, "")
        Visszatérés
    Különben ha splitInput hossza egyenlő 3-mal és splitInput[2] egyenlő
"-a"-val
        pickupAll = Igaz
        currentEntity.currentRoom.initiatePickup(pickupAll,    currentEntity,
splitInput, currentEntity inventory-ának mérete)
Vége
```

- **+drop(): void:** Lehetőséget ad egy tárgy eldobására a soron következő entitás hátizsákjából. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor meghívja az 'initiateDrop()' függvényt, amely elvégzi az eldobást, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```
Függvény drop()
    Ha nem gameStarted
        errorMsg(1, "")
        Visszatérés
    splitInput = lastInput szétválasztása
    Ha splitInput hossza nagyobb, mint 3 vagy kisebb, mint 2
        errorMsg(0, "")
        Visszatérés
    Különben ha splitInput hossza egyenlő 3-mal és splitInput[2] nem
egyenlő "-a"-val
        errorMsg(4, "")
        Visszatérés
    currentEntity.intiateDrop(splitInput)
Vége
```

- **+use(): void:** Lehetőséget ad egy tárgy használatára. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor meghívja az 'initiateUse()' függvényt, ami kezeli a tárgyhasználatot, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```
Függvény use()
    Ha nem gameStarted
        errorMsg(1, "")
        Visszatérés
    splitInput = lastInput szétválasztása
    Ha splitInput hossza nem egyenlő 2-vel
        errorMsg(0, "")
    Különben
        currentEntity.intiateUse(splitInput)
```

Vége

- **+move(): void:** Lehetőséget ad az éppen soron következő entitás mozgására/helybenmaradásra. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor meghívja az 'initiateMove()' függvényt a kiválasztott ajtóra. Egyhelyben maradáskor szintén az 'initiateMove()' függvényt hívja csak ebben az esetben az ajtó azonosító helyére '-1' értéket ad át. Az 'initiateMove()' függvény kezeli a mozgást, majd válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```
Függvény move()
    tokenizedInput = lastInput szétválasztása
    Ha tokenizedInput hossza egyenlő 1-gyel
        currentEntity.getCurrentRoom().initiateMove(-1, currentEntity)

    Különben ha tokenizedInput hossza nagyobb mint 2
        errorMsg(0, "")
    Különben
        roomId = tokenizedInput[1]
        currentEntity.getCurrentRoom().initiateMove(roomId,
currentEntity)
    Ha hiba a mozgással
        errorMsg(302, "DoorId: " + tokenizedInput[1])
```

Vége

- **+connect(): void:** Lehetőséget ad kettő, az éppen soron következő játékos hátizsákjában lévő tárgy összekapcsolására. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor meghívja az 'initiateConnect()' függvényt ami kezeli a tárgyak összeköttetését, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```
Függvény connect()
    split = lastInput szétválasztása
    Ha split hossza kisebb mint 3
        errorMsg(0, "")
        Visszatérés
    currentEntity.initiateConnect(split)
```

Vége

- **+disconnect(): void:** Lehetőséget ad kettő, az éppen soron következő játékos hátizsákjában lévő összekapcsolt tárgy szétkapcsolására. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor meghívja az 'initiateDisconnect()' függvényt ami kezeli a

tárgyak összeköttetését, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```
Függvény disconnect()
    split = lastInput szétválasztása
    Ha split hossza kisebb mint 3
        errorMsg(0, "")
        Visszatérés
    currentEntity.initiateDisconnect(split)
```

Vége

- **+teleport(): void:** Lehetőséget ad az éppen soron következő játékos tranzisztortal történő teleportálásához. Ellenőrzi, hogy helyesen lett-e hívva, ha nem akkor hibaüzenetet dob az 'errorMsg()' függvény segítségével. Helyes használatkor meghívja a teleportálásra használható tranzisztor 'teleport()' függvényét, amely a teleportálást kezeli, és válaszol a felhasználónak a 'statusMsg()' függvény használatával.

Pszeudokód:

```
Függvény teleport()
    split = lastInput szétválasztása
    Ha split hossza kisebb mint 2
        errorMsg(0, Nulla)
        Visszatérés
    id = split[1]
    t = -1
    Ciklus i = 0-tól i < currentEntity.inventory méretéig
        Ha currentEntity.inventory[i].getID() egyenlő id-vel
            t = i
    Ciklus vége
    s = "" + id
    Ha t nem egyenlő -1-gyel
        b = currentEntity.inventory[t].teleport(currentEntity,
currentEntity.currentRoom)
        Ha b igaz
            currentEntity.roundIsOver = igaz
            statusMsg(513, s)
            Visszatérés
```

Vége

- **+errorMsg(int errorType, String param): void:** Lehetőséget ad hibaüzenetek kiírásához. Egy nagy switch-case függvényen belül az 'errorType' értéknek megfelelő hibaüzenetet írja ki. Ha az üzenet mellé társul bővebb információ, akkor azt a 'param' változó segítségével lehet átadni, amely egyes üzeneteket kiegészít bővebb információkkal. Mivel csak az üzenetek kiírásával foglalkozik, és nincsen benne logika, ezért a pszeudokód leírásával nem bajlódunk.
- **+statusMsg(int statMsg, String param): void:** Lehetőséget ad státusz üzenetek kiírásához. Egy nagy switch-case függvényen belül a 'statMsg' értéknek megfelelő hibaüzenetet írja ki. Ha az üzenet mellé társul bővebb információ, akkor azt a 'param' változó segítségével lehet átadni, amely egyes üzeneteket kiegészít bővebb információkkal. Mivel csak az üzenetek kiírásával foglalkozik, és nincsen benne logika, ezért a pszeudokód leírásával nem bajlódunk.
- **+fileCompare(String filePath1, String filePath2): boolean:** Lehetőséget ad kettő file összehasonlításához. A kettő kapott fület soronként beolvassa, és azokat összehasonlítja egymással. Ha eltérést talál, akkor a 'compareOutput.txt' nevű fileba kiírja, hogy melyik sorban van az eltérés, ezek után pedig a kettő eltérő sort egymás

alá írja. Hogyha a kettő beolvasott file hossza eltér egymástól, akkor a 'compareOutput.txt' fileba a következő sor kerül: "Input files does not match in length aborted". Ha a kettő file megegyezik, akkor a kimeneti fileba nem kerül semmi.

Pszeudokód:

```
Függvény fileCompare(filePath1, filePath2)
    correct = igaz
    file1 = Új Fájl
    file2 = Új Fájl
    writer = Új Fájl író a "compareOutput.txt" fájlhoz
    Ha file1 hossza nem egyenlő file2 hosszával
        writer Fájlba írás: "Input files does not match in
length aborted."
        writer Fájlíró bezárása
        ViSSzatérés hamis
    reader1 = Új Olvasó a file1 fájlhoz
    reader2 = Új Olvasó a file2 fájlhoz
    lineCount = 0
    Amíg reader1 tud sort olvasni
        lineCount++
        fileLine1 = reader1 Következő sor olvasása
        fileLine2 = reader2 Következő sor olvasása
        Ciklus i = 0-tól i < fileLine1 hosszáig
            Ha fileLine1[i] nem egyenlő fileLine2[i]-vel
                writer Fájlba írás: "--> line " +
lineCount + ":\n" + fileLine1 + "\n" + fileLine2 + "\n")
                correct = hamis
            Kilépés ciklusból
        Ciklus vége
    reader1 Olvasó bezárása
    reader2 Olvasó bezárása
    writer Fájlíró bezárása
    Fájl hiba esetén
        errorMsg(404, filePath1 + " or" + filePath2)
    ViSSzatérés correct
Vége
```

8.1.10 Item

- Felelősség

Ez az osztály képi a tárgyak alapját.

- Ősosztályok

- Interfészek

Serializable, Debuggable

- Attribútumok

- - **ID : int** - *Az Item egyedi azonosítója*
- - **durability : int** - *Az Item fennmaradó élettartama*

- Metódusok

- + **abstract debug() : string** - *lehetővé teszi, hogy minden gyerekosztály definiáljon egy debugot.*
- + **item(int aID)** - konstruktor, beállítja az ID változót
- + **getID() : int** - visszatér az item ID-ját.
- + **setID(int aID)** - beállítja az ID-t a megadott id-ra.
- + **checkID(int inID)** - ID komparátor. inID - az ellenőrizendő ID. Visszatérés: egyenlőség az ellenőrizendő, és az item példány IDja között.
- + **getDurability() : int** - Visszatérés: tárgy tartóssága
- + **setDurability(int aDurability)** - Beállítja a tárgy tartósságát a megadott paraméterre.
- + **getPair() : Transistor** - A Transistorhoz pár érték getter.
- + **connect(Item i, Entity ent)** - *ét tárgy összekapcsolására szolgáló függvény. További bővíthetőség érdekében felül lehet definiálni hogy más tárgyak is tudjanak kapcsolódni. Alapesetben "false" értékkel tér vissza ha a két adott tárgy nem tud kapcsolódni. i - A másik tárgy, amivel összekapcsolódik. Visszatérés: False, mivel csak a Transistornál lesz értelme, ott lesz override-olva.*
- + **disconnect(Item i)** - *Két tárgy szétkapcsolására szolgáló függvény. Alapesetben "false" értékkel tér vissza ha a két adott tárgy nem tud szétkapcsolódni. i - A tárgy párja, amit szét szeretnénk kapcsolni a tárgytól. Visszatérés: False, mivel csak a Transistornál lesz értelme, ott lesz override-olva.*
- + **use(Room r, Entity e) : boolean** - *Amennyiben nincs felüldefiniálva a metódus az adott örökös tárgy osztályban akkor "false" értékkel tér vissza. Beer, Transistor és Camembert osztályokban van felüldefiniálva ez a függvény, mely egyedi használati működésüket valósítja meg. r - A szoba, ahol a tárgyat használó Entity tartózkodik. e - A tárgyat használó Entity. Visszatérés: False, a használható tárgyaknál felül kell definiálni.*
- + **save(Entity ent) : boolean** - *Amennyiben az adott tárgy életmentő, akkor ez a függvény felül van definiálva (és az adott osztály leírásban részletezve). Alapesetben visszatér egy "false"-al.*

- **+ teleport(Entity e, Room r) : boolean** - A játékos szobák közötti teleportálását teszi lehetővé. Ha egy teleportálást nem megvalósító tárgyon hívódik, művelet nélkül hamis értékkel tér vissza. e - Az Entity, aki teleportál. Visszatérés: False, mivel csak a Teleport tárgynál van értelme.
- **+ableToStun() : boolean** - A tárgy eszméletvesztés okozásának logikai értékét adja meg. Alap esetben "false" értéket ad vissza.
- **+abstract pickUp(Entity e)** - Minden tárgyhöz külön definíciója van a felvételnek (visitor pattern), ezáltal minden tárgynál külön lesz felüldefiniálva ez az absztrakt függvény. e - Az Entity, aki felveszi a tárgyat. Visszatérés: Igaz, ha sikeres a felvétel, egyéb esetben hamis.
- **+abstract drop(Entity e)** - Minden tárgyhöz külön definíciója van az eldobáshoz (visitor pattern), ezáltal minden tárgynál külön lesz felüldefiniálva ez az absztrakt függvény. e - Az Entity, aki lerakja a tárgyat. Visszatérés: Igaz, ha sikeres a lerakás, egyéb esetben hamis.
- **+preventGasStun() : boolean** - A gáz által okozott bénítás kivédését végzi. Virtuális metódus, ami hamis értékkel tér vissza. Ha egy tárgy véd a gáz ellen, ott felül kell definiálni. Visszatérés: A gáz elleni védekezés logikai értéke.
- **+ carriedIntoNormalRoom()** - Egy tárgy normál szobába bevitelével járó állapotváltozásokat kezeli. Virtuális metódus ami üres, ha egy tárgy a működést megvalósítja, ott felül kell definiálni.

8.1.11 Janitor

- **Felelősség**
A játékban szereplő egyik karakter amely a szobák között járkálva próbál rendet rakni a termekben.
- **Össztályok**
Az Entity osztályból származik.
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Ennek az osztálynak nincsenek attribútumai.
- **Metódusok**
 - **+Janitor(String, int, Room)** A takarító osztály konstruktora. Argumentumként vár egy stringet, amely a takarító nevéként, egy egész számot, amely a takarító egyedi azonosítójaként, valamint egy Room referenciát, amely a takarító jelenlegi tartózkodási pozíciójaként kerül beállításra. Mindezt teszi úgy, hogy meghívja az osztály konstruktorát az adott értékekkel. Továbbá hozzáadja a referenciáját a GameController debuggableObjects listájához.
 - **+moveTo(Door): boolean** Az osztályban lévő metódus felüldefiniálása. Működése megegyezik az osztályban található metóduséval, egyedüli különbség, hogy sikeres szobaváltás esetén meghívja az újonnan beállított tartózkodási szobájára a janitorEntered() metódust. Pszeudokód:

```

Függvény moveTo(Door door):

//ha a kapott ajtó referencia null, helyben akar maradni az Entity
Ha door == null akkor:
    roundIsOver = currentRoom.acceptEntity(this)
Különben:
    Ha remainingStun == 0:
        ret = door.moveEntity(this, currentRoom)

        //Ha sikerült a szoba váltás, ki is kell takarítani.
        Ha ret == igaz akkor
            currentRoom.janitorEntered()
        Ha vége

        //Ha sikerül a szoba változtatás az Entity körének vége
        roundIsOver = ret
    Különben:
        visszatérés hamis
    Ha vége
Ha vége
visszatérés roundIsOver

Függvény vége

```

- **+pickUpItem(SlideRule): boolean** Logarléc felvételének logikája, mivel egy takarító nem vehet fel logarléceket ezért alapértelmezetten hamis értékkel tér vissza.
- **+pickUpItem(Transistor): boolean** Tranzisztor felvételének logikája, mivel egy takarító nem vehet fel tranzisztort ezért alapértelmezetten hamis értékkel tér vissza.
- **+meetMe(Entity): void** Az őssosztályban lévő metódus felüldefiniálása. A paraméterül kapott Entityre meghívja annak meet metódusát a saját referenciájával.
- **+newEntRound(): boolean** Működése azonos a tanár azonos nevű metódusával.
- **+toString() : String** Visszatér az alábbi kimenettel: "Janitor: {takarító neve}"

8.1.12 Map

- **Felelősség**

Tárolja a játék térképét (szobáit), itt valósul meg a map generálás logikája, ebbe való entitások és tárgyak betételének logikája is.

- **Őssosztályok**

Nincs.

- **Interfészek**

Debuggable és Serializeable

- **Attribútumok**

- **+ int ID:** Egyedi azonosító
- **+ ArrayList<Room> rooms :** A térképen található szobák listája

- **Metódusok**

- **+ Map(int id):** Konstruktor

- + **void addRoom(Room newRoom):** Hozzá ad egy szobát a térképhez
- + **void removeRoom(Room roomToRemove) :** Eltávolít egy szobát a térképről
- + **void buildMap(int levels, int extraEdges) :** Felépíti a játéktérteret (térképet) először egy bináris fát (szoba -> csúcs, ajtó -> él). Második lépés képpen hozzá ad adott mennyiségű ajtót figyelve arra hogy nincs-e még ajtó a két szoba között vagy nem 1 szobába tesz ajtót önmagához.

Pszeudokód:

```
Függvény buildMap(int levels, int extraEdges):
    lista waitingForDoors
    lista waitingForWaitingForDoors
    waitinForDoors( Room(false, 5, GameController.getNextGlobalID()))
    Ciklus 0-tól levels-ig
        Ciklus 0-tól waitinForDoors méretéig
            Room newR1
            Room newR2
            Door newD1
            Door newD2
            newR1.addDoor(newD1)
            newR2.addDoor(newD2)
            waitingForDoors[j].addDoor(newD1)
            waitingForDoors[j].addDoor(newD2)
        Ciklus vége
    rooms.addAll(waitingForDoors)
    waitingForDoors.addAll(waitingForWaitingForDoors)
    waitingForWaitingForDoors.clear()
    Ciklus vége
    int room1Index = -1
    int room2Index = -1
    boolean canAddDoor = hamis
    Ciklus 0-tól extraEdges méretéig
        Ciklus amíg(room1Index == room2Index vagy room1Index < 3 vagy
room2Index < 3)
            room1Index = randomszám 0 és room méret -1 közt
            room2Index = randomszám 0 és room méret -1 közt
            Ha hasDoor(room1Index, room2Index) igaz akkor
                Door newD
                rooms[room1Index].addDoor(newD)
                rooms[room2Index].addDoor(newD)
                canAddDoor = igaz
            Ciklus vége
        Ciklus vége
    Függvény vége
```

- **+ void fillWithItems(int normalCount, int fakeCount):** Térképet tárgyakkal feltöltő metódus. Először egy listába rak meghatározott mennyiségű tárgyat, majd ezeket a Random osztály segítségével szétszórja a térképen.

Pszudokód:

```
Függvény fillWithItems(int normalCount, int fakeCount):
    lista itemsToAdd
    itemsToAdd-hozz hozzá adunk egy SlideRulet
    Ciklus 0-tól normalCount-ig
        itemsToAdd-hozz hozzá adunk egy AirFreshnert
        itemsToAdd-hozz hozzá adunk egy Beert
        itemsToAdd-hozz hozzá adunk egy Camembertet
        itemsToAdd-hozz hozzá adunk egy Maskot
        itemsToAdd-hozz hozzá adunk egy Raget
        itemsToAdd-hozz hozzá adunk egy TVSZt
        itemsToAdd-hozz hozzá adunk egy Tranzisztort
    Ciklus vége
    itemsToAdd-hozz hozzá adunk egy FakeSlideRulet
    Ciklus 0-tól fakeCount-ig
        itemsToAdd-hozz hozzá adunk egy FakeMaskot
        itemsToAdd-hozz hozzá adunk egy FakeTVSZt
    Ciklus vége
    int lastID = -1
    int randID = -1
    Ciklus 0-tól itemsToAdd méretéig
        Ciklus amíg randID == lastID vagy (i == 0 és randID < 6)
            randID = random szám 0 és rooms mérete közt
        Ciklus vége
        rooms[randID].addItem(itemsToAdd[i])
    Ciklus vége
Függvény vége
```

- **+ void entityPlacer(ArrayList<Entity> students) :** Hozzá ad a térképhez adott mennyiségű tanárt és takarítót figyelembe véve, hogy ne tegyen 2x ugyanoda egymás után entitást és az ID egyedi legyen az összes entitásra (ide értve a diákokat is).

- **+ boolean hasDoor(int r1Index, int r2Index) :** Vissza adja hogy a 2 szoba (index alapján) között van-e ajtó

Pszudokód:

```
Függvény hasDoor(int r1Index, int r2Index):
    lista doors
    Ciklus 0-tól doors méretéig
        Ha doors[i].containsRoom(rooms[r2Index])
            visszatérés true
    Ciklus vége
Függvény vége
```

- **- int[] getRandDiffIndexes(int count, int range) :** Beállítható mennyiségű random különböző ID-t hoz létre a range-ig

- + **boolean mergeMany(int count)** : Összeolvaszt megadható mennyiségű szobát, ha több a mennyiség mint a szobák száma összesen a térképen false-t ad vissza.

Pszeudokód:

```
Függvény mergeMany(int count)
    Ha rooms mérete < count akkor
        visszatérés hamis
    Különben
        int[] roomIDs = getRandDiffIndexes(count, rooms méret -1)
        Ciklus 0-tól count-ig
            mergee = rooms[roomIDs]
            Room neighbour = mergee.getFirstNeighbour
        Ciklus vége
        visszatérés hamis
Függvény vége
```

- + **boolean splitMany(int count)** : Megadható mennyiségű random szobát választ ketté, ha lehetséges.

Pszeudokód:

```
Függvény splitMany(int count)
    Ha rooms mérete < count akkor
        visszatérés hamis
    Különben
        int[] roomIDs = getRandDiffIndexes(count, rooms méret -1)
        Ciklus 0-tól count-ig
            Room newR = roomById(roomIDs[i]).split()
            Ha newR != null akkor
                rooms.add(newR)
        Ciklus amíg
        visszatérés igaz
Függvény vége
```

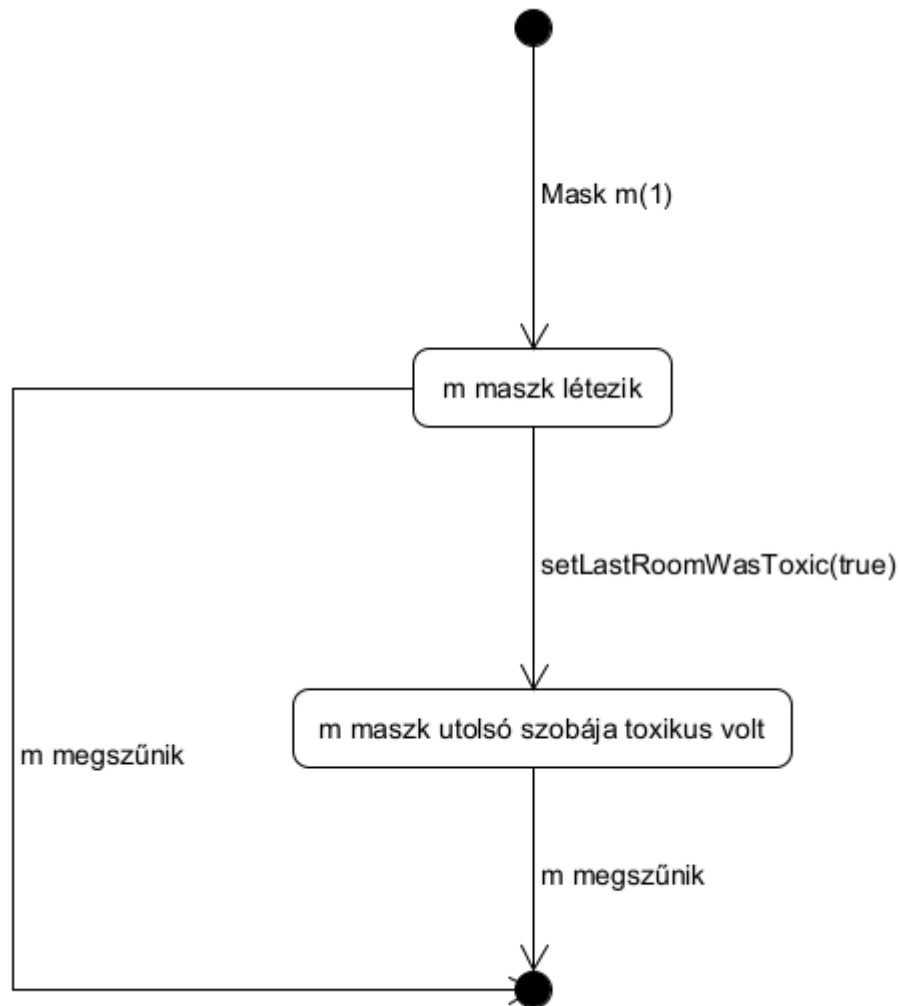
- + **boolean curseMany(int count)** : Megadható mennyiségű, különböző szoba összes ajtaját elátkozottá teszi

Pszeudokód:

```
Függvény mergeMany(int count)
    Ha rooms mérete < count akkor
        visszatérés hamis
    Különben
        int[] roomIDs = getRandDiffIndexes(count, rooms méret -1)
        Ciklus 0-tól count-ig
            roomById(roomIDs[i]).curseAllDoors()
        Ciklus vége
        visszatérés igaz
Függvény vége
```

- + **void tickRooms()** : A térkép összes szobájának jelzi, hogy új kör kezdődött.
- + **Room roomById(int id)** : Visszatér a paraméterként megadott ID-vel rendelkező szoba referenciájával. Ha nem talál szobát a kapott ID-hez, akkor NullPointerException-t dob, annak megadva a kapott ID-t
- + **String debug()** : Debug szöveg generálása

8.1.13 Mask



- **Felelősség**

A Mask Item-mel végzett műveleteket kezelő osztály.

- **Össztályok**

Item osztály

- **Interfészek**

Debuggable

- **Attribútumok**

- **-boolean lastRoomWasToxic:** Megadja, hogy az előző szoba mérgező volt-e.

- **Metódusok**

- **+Mask(int aID):** Az osztály konstruktora, beállítja a tárgy ID-ját.

pszeudokód:

Függvény Maszk(int aID)

```
        setDurability(5)
```

```
        GameController.getInstance().debuggableObjects.put(aID, Mask példány)
```

Függvény vége

- **+boolean getLastRoomWasToxic():** A lastRoomWasToxic attribútum gettere.

pszeudokód:

```
Függvény getLastRoomWasToxic()
```

```
    visszatérés lastRoomWasToxic
```

Függvény vége

- **+void setLastRoomWasToxic(boolean b):** A lastRoomWasToxic attribútum settere.

pszeudokód:

```
Függvény setLastRoomWasToxic(boolean b)
```

```
    lastRoomWasToxic = b
```

Függvény vége

- **+boolean pickUp(Entity e):** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity “pickUpItem” rá vonatkozó változtatát. (visitor pattern)

pszeudokód:

```
Függvény pickUp(Entity e)
```

```
    visszatérés e.pickUpItem(Mask példány)
```

Függvény vége

- **+boolean drop(Entity e):** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity “dropItem” rá vonatkozó változtatát. (visitor pattern)

pszeudokód:

```
Függvény drop(Entity e)
```

```
    visszatérés e.dropItem(Mask példány)
```

Függvény vége

- **+boolean preventGasStun():** A gáz által okozott bénítás kivédését végzi. Ellenőrzi, hogy a maszk tartalmaz-e még felhasználható töltetet.

pszeudokód:

```
Függvény preventGasStun()
```

```
    Ha getDurability() > 0
```

```
        Ha getLastRoomWasToxic()
```

```
            Ha getDurability < 2
```

```
                visszatérés hamis
```

```
            setDurability(getDurability() - 2)
```

```
        különben
```

```
            setDurability(getDurability() - 1)
```

```
            setLastRoomWasToxic(igaz)
```

```
        visszatérés igaz
```

```
    visszatérés hamis
```

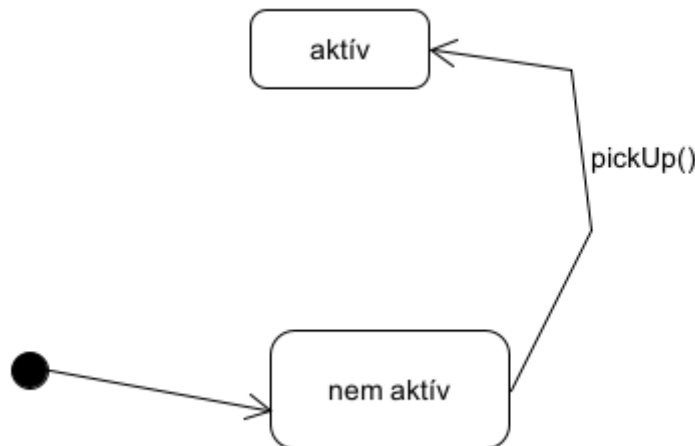
Függvény vége

- **+void carriedIntoNormalRoom:** Egy maszk tárgy normál szobába való bevitelével járó állapotváltozását kezeli.
 pszeudokód:
 Függvény carriedIntoNormalRoom()
 setLastRoomWasToxic(hamis)
 Függvény vége
- **+String toString():** Visszatér egy Stringgel ami a maga nevéből és az azonosítójából áll.
 pszeudokód:
 Függvény toString()
 visszatérés "Mask: #" + Integer.toString(getID())
 Függvény vége
- **+String debug():** Debug szöveg generálása.
 pszeudokód:
 Függvény debug()
 visszatérés "---- Mask" + Mask példány.getID() + "--\ndurability : "
 + getDurability() + "\nlastRoomWasToxic : " + lastRoomWasToxic +
 "\n-- Mask " + Mask példány.getID() + "-----\n"
 Függvény vége

8.1.14 Rag

- Felelősség

A Rag Item-mel végzett műveleteket kezelő osztály.



- Ősosztályok

Item

- Interfészek

Decaying, Debuggable

- Attribútumok

- - isActive : boolean

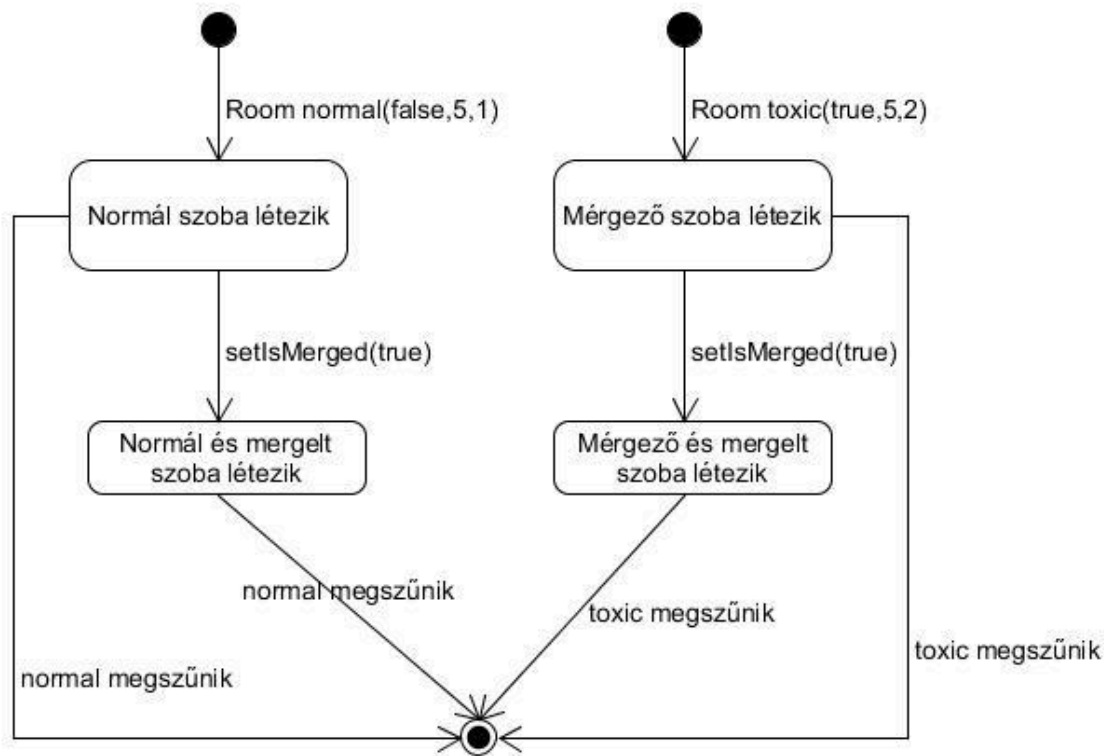
- Metódusok

- + **Rag(int aID)** - Az osztály konstruktora, beállítja a tárgy ID-ját.
- + **getIsActive() : boolean** - Visszaadja az isActive attribútum értékét.
- - **setIsActive(boolean b)** - Az isActive attribútum settere.
- + **boolean pickUp(Entity e)**: A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity "pickUpItem" rá vonatkozó változtatát. (visitor pattern)
- + **boolean drop(Entity e)**: A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity "dropItem" rá vonatkozó változtatát. (visitor pattern)
- + **age()** - Ez a függvény a tárgy korának állításáért felel.
- + **ableToStun() : boolean** - A rongy tanárra gyakorolt eszmélet vesztesi képességének logikai értéke ha a rongy aktív és van benne még töltet akkor "true" értékkel tér vissza ellenkező esetben "false"
- + **String toString()**: Visszatér egy Stringgel ami a maga nevéből és az azonosítójából áll.
- + **String debug()**: Debug szöveg generálása.

8.1.15 Room

- **Felelősség**

Ez az osztály képi le a játékbeli szobákat, felelős egy adott szoba ajtajainak tárolásáért valamint itt vannak definiálva az összeolvasztás, kettéválasztás mechanizmusai is.



- **Össztályok**

Nincs

- **Interfészek**

Debuggable és Serializeable

- **Attribútumok**

- - **in ID**: Egyedi azonosító
- - **boolean isToxic**: A szoba toxicitásának logikai értéke
- - **boolean isMerged**: A szoba összeolvasztás szüziességének logikai értéke
- - **boolean isSticky**: A szoba ragadósságának logikai értéke
- - **int capacity**: A szoba befogadó képességének száma
- - **boolean hasBeenCleaned**: A szoba (legalább egyszer) kitakarított mivoltának logikai értéke
- - **int entitiesSinceCleanup**: Az utolsó takarítás óta a szobába lépett Entity-k száma, csak akkor érvényes ha a hasBeenCleaned igaz értékű
- - **ArrayList<Item> containedItems**: A szobában tárolt tárgyak listája
- - **ArrayList<Entity> entitiesInside**: A szobában lévő entitások listája

- - **ArrayList<Entity> waitingToEnter**: A szobába belépni várakozó entitások listája
- - **ArrayList<Door> doors**: A szoba ajtajainak listája
- **Metódusok**
 - + **Room(boolean toxic, int cap, int id)**: Konstruktor
Pszudokód:
Függvény Room(boolean toxic, int cap, int id):

```

        isToxic = toxic
        isMerged = Hamis
        capacity = cap
        hasBeenCleaned =: Hamis
        entitiesSinceCleanup = 0
        ID =: id
        debuggableObjects-hez hozzá adjuk a kulcspárt(ID, Room példány)

```

Függvény vége
 - + **int getID()**: Az egyedi azonosítót adja vissza
 - + **void setID(int id)**: Az egyedi azonosítót állítja
 - + **boolean getIsToxic()**: isToxic logikai érték getter függvénye
 - + **void setIsToxic(boolean t)**: isToxic alapvető setter függvénye
 - + **void setIsToxic(int state)**: Beállítja az isToxic logikai értéket a kapott int alapján
 - + **boolean getIsMerged()**: A szoba összeolvasztás szüziességének logikai értékét visszaadó getter függvény
 - + **void setIsMerged(boolean m)**: A szoba összeolvasztás szüziességének logikai értékét beállító setter függvény
 - + **void setIsSticky(boolean sticky)**: Az isSticky logikai változó setter függvénye
 - + **int getCapacity()**: A szoba befogadóképességét vissza adó getter függvény
 - + **void setCapacity(int c)**: A szoba befogadóképességét beállító setter függvény
 - + **boolean getHasBeenCleaned()**: hasBeenCleaned tagváltozó getter metódusa
 - + **void setHasBeenCleaned(boolean cleaned)**: Beállítja a szoba (legalább egyszer megtörtént) kitisztítottságának logikai értékét a kapott paraméter alapján
 - + **int getEntitiesSinceCleanup()**: entitiesSinceCleanup tagváltozó getter metódusa
 - + **void setEntitiesSinceCleanup(int rounds)**: Beállítja az utolsó takarítás óta a szobába belépett Entity-k számát a paraméterben kapott értékre
 - + **ArrayList<Door> getDoors()**: A szoba ajtajainak listáját vissza adó getter függvény
 - + **int getEntityCount()**: Visszaadja szobában lévő entitások számát
 - + **ArrayList<Entity> getEntitesInside()**: A entitiesInside lista getter függvénye
 - + **ArrayList<Item> getContainedItems()**: Tárolt tárgyak getter függvénye
 - + **void addItem(Item i)**: Hozzá ad egy tárgyat a szobában lévő tárgyak listájához
 - + **Item removeItem(Item i)**: Kiveszi az adott tárgyat a szobában lévő tárgyak listájából

- **+ Item giveAnItem():** Ad egyet a szobában található tárgyak közül
 Pszeudokód:
 Függvény giveAnItem():
 numOfItemsInRoom = containedItems mérete
 Ha numOfItemsInRoom == 0
 akkor visszatérés null értékkel
 selectedItemId = random generált szám 0 és numOfItemsInRoom közt
 visszatérés containedItems[selectedItemId]
 Függvény vége
- **+ boolean initiateMove(int doorId, Entity ent):** Egy helyváltoztatás végrehajtását kezdeményezi. Ellenőrzi, hogy a megadott ajtó tényleg a szobában található-e. Valamint elvégzi az ajtóId és referencia közötti feloldást.
 Pszeudokód:
 Függvény initiateMove(int doorId, Entity ent):
 Ha doorId == -1
 visszatérés ent.moveTo(null)
 Ciklus 0-tól doors méretéig
 Ha doors[i].ID == doorId
 visszatérés ent.moveTo(doors[i])
 Ciklus vége
 GameController.errorMessage(302, doorId)
 visszatérés hamis
 Függvény vége
- **+ boolean acceptEntity(Entity ent):** Kezeli azt, hogy az entitás aki be szeretne menni a szobába befér - e. Hogy ha igen, hozzáadja a kapott entitást (e) az entitások listájához (entitiesInside), meghívja az Entity roomChanged() metódusát, valamint ha a szoba gázzal telített, az Entity toxicate() metódusát is. Ezek után visszatér "true"-val. Ha a szoba férőhelye megtelt, akkor pedig visszatér "false"-al.
 Pszeudokód:
 Függvény acceptEntity(Entity ent):
 Ha(entitiesInside tartalmazza ent-et)
 Ha isToxic igaz
 ent.toxicate()
 Különben
 ent.enteredNormal()
 GameController.statusMsg(311)
 visszatérés igaz
 Különben ha(capacity > (entitiesInside mérete + waitingToEnter mérete))
 addToQueue(ent)
 ent.roomChanged(Room példány)
 Ha isToxic igaz
 ent.toxicate()
 Különben
 ent.enteredNormal()
 entityEntered()
 GameController(312)
 visszatérés igaz
 GameController.errorMessage(301)
 Függvény vége
- **+ void janitorEntered():** Janitor szobába lépésének következményeit kezeli. Akkor hívandó, ha már a takarító biztosan be tudott lépni a szobába. A szobában lévő összes

Entity-re meghívódik a szoba „throwOut()” metódusa, mellyel kitessékeli őket. Végül a szobában megtörténik a takarítás, azaz a „setHasBeenCleaned” igaz értékűre állítódik, valamint a „roundsSinceCleanup” értéke 0-ra változik. Meghívásra kerül a szoba „makeNotToxic” metódusa, amely a szobában lévő esetleges gáz kiszellőztetésért felel.

Pszeudokód:

Függvény janitorEntered():

```
lista entitiesToThrowOut = entitiesInside
Ciklus 0-tól entitiesToThrowOut méretéig
    throwOut(entitiesToThrowOut[i])
Ciklus vége
hasBeenCleaned = igaz
entitiesSinceCleanup =: 0
isSticky =: hamis
makeNotToxic()
```

Függvény vége

- + **void addEntity(Entity ent)**: Hozzáad egy entitást a szobában lévő entitások listájához
- + **void removeEntity(Entity ent)**: eltávolít egy entitást a szobában lévő entitások listájából
- + **void addToQueue(Entity ent)**: Hozzáad egy Entityt a szobába belépésre várakozók listájába
- + **void makeToxic()**: Mérgezővé teszi a szobát
- + **void makeNotToxic()**: Megtisztítja a szobát a mérgezőségtől
- + **boolean merge(Room r2)**: A szobák összeolvasztását kezelő függvény

Pszeudokód:

Függvény merge(Room r2):

```
Ha entityCount == 0 akkor
    lista tmpItemList = r2.mergeIntoMe()
    Ha tmpItemList != null akkor
        int state = GameController.newMergeState(Room példány,
r2)

        isToxic =: setIsToxic(state)
        GameController.mergeDoors(Room példány, r2)
        capacity =: r2.myCapacity(capacity)
        Ciklus 0-tól tmpItemList méretéig
            addItem(tmpItemList[i])
        Ciklus vége
        visszatérés igaz
    Különben
        visszatérés hamis
    Különben
        visszatérés hamis
```

Függvény vége

- + **ArrayList<Item> mergeIntoMe()**: A merge() által hívott függvény, amely jelzi a cél szobának hogy olvadjon vele össze, ha üres visszaadja benne tárolt tárgyak listáját amúgy null értéket

- **+ Room split():** Egy szoba kettéválasztását kezelő függvény

Pszeudokód:

Függvény split():

Ha entityCount == 0 akkor

Room newR

Ciklus 0-tól containedItems méretének feléig

newR.addItem(containedItems[i])

Ciklus vége

Ciklus 0-tól doors méretének feléig

newR.addDoor(doors[i])

Ciklus vége

Door conn

addDoor(conn)

newR.addDoor(conn)

visszatérés newR

Különben

visszatérés null

Függvény vége

- **+ void curseAllDoors():** Elátkozza egy szoba összes ajtaját
- **+ Room getFirstNeighbour():** Visszaadja egy szoba első szomszédos szobáját
- **+ int myCapacity(int myCap):** Átadja egy szoba a saját kapacitását, annak a szobának amelyen meghívta hogy összehasonlítsa saját kapacitásával
- **+ void addDoor(Door d1):** Hozzáad egy ajtót a szobában tárolt ajtók listájához
- **+ void removeDoor(Door d1):** Eltávolít egy ajtót a szobában tárolt ajtók listájáról
- **+ boolean tryToMove(int noOfDoors, Entity entity, Room movefrom):** Adott számú ajtón megpróbálja az adott Entityt átmozgatni. Ha a megadott szám nagyobb mint ahány ajtó van a szobában az összesen megpróbálja átmozgatni. Az első sikeres mozgatsátnál a próbálkozás leáll.

Pszeudokód:

Függvény tryToMove(int noOfDoors, Entity entity, Room movefrom):

Ha entitiesInside nem tartalmazza entity-t vagy noOfDoors == 0 akkor
visszatérés hamis

Set doorIndexes

Ciklus amíg doorIndexes mérete < noOfDoors és doorIndexes mérete < doors mérete

doorIndexes-be beteszünk egy random generált szám 0-tól doors méretéig

Ciklus vége

Ciklus 0-tól doorIndexes méretéig

boolean isMoveSuccessful =: doors[i].moveEntity(entity, movefrom)

Ha isMoveSuccessful igaz akkor

visszatérés igaz

Ciklus vége

visszatérés hamis

Függvény vége

- **+ boolean throwOut(Entity entity):** Kidob egy Entity-t a szobából amennyiben lehetséges
 Pszeudokód:
 Függvény throwOut(Entity entity):
 boolean canBeMoved =: hamis
 Ciklus 0-tól doors méretéig
 canBeMoved =: entity.moveTo(doors[i])
 Ha canBeMoved igaz akkor
 visszatérés igaz
 Ciklus vége
 visszatérés hamis
 Függvény vége
- **+ boolean containsRag():** Végig megy a szoba összes tárgyán amennyiben van benne olyan amit eszméletlenné tud tenni, "true" értékkel tér vissza ellenkező esetben false
 Pszeudokód:
 Függvény containsRag():
 boolean isActRag =: hamis
 Ciklus 0-tól containedItems méretéig
 isActRag =: containedItems[i].ableToStun()
 Ha isActRag igaz akkor
 visszatérés igaz
 Ciklus vége
 Függvény vége
- **+ void entityEntered():** Egy Entity szobába belépésének a szobára gyakorolt hatásait érvényesíti. Ellenőrzi, hogy a szoba ki lett-e már takarítva. Ha igen, megnöveli a takarítás óta a szobába belépett Entity-k számát eggyel (entitiesSinceCleanup).
 Pszeudokód:
 Függvény entityEntered():
 Ha hasBeenCleaned igaz akkor
 entitiesSinceCleanup =: entitiesSinceCleanup + 1;
 Ha entitiesSinceCleanup >= 5 akkor
 isSticky =: igaz
 Függvény vége
- **+ void tick():** Szoba egy új kör kezdetekor esedékes műveleteit végzi el. A szobába belépésre várakozó entity-ket a szobában tartózkodók listájába rakja, valamint végbemennek a találkozók is. Mindenki mindenki mással pontosan egyszer találkozik.
 Pszeudokód:
 Függvény tick():
 entitiesInside tömbhöz hozzá adjuk waitingToEnter tömböt
 int countOfEntitiesInRoom = entitiesInside mérete
 Ciklus 0-tól szobában lévő entitások számáig
 Ciklus 0-tól szobában lévő entitások számáig
 Entity entToMeet = entitiesInside[j]
 entitiesInside[i].meetMe(entToMeet)
 Ciklus vége
 Ciklus vége
 Függvény vége

- + **String listDoors()**: Készít egy listát a szoba ajtajaival, és azzal, hogy azok melyik szobába tartanak.

Pszudokód:

```
Függvény listDoors()
    string message
    string section
    string roomSection
    Ciklus 0-tól doors méretéig
        section = doors[i].toString()
        roomSection = doors[i].getCurrentDirection(Room példány)
        message = message + section + roomSection
    Ciklus vége
    visszatérés message
Függvény vége
```

- + **String listEntities()**: Listába szedi a szobában lévő Entityket, és egy ebből álló stringgel tér vissza

Pszudokód:

```
Függvény listEntities():
    string message
    string section
    Ciklus 0-tól entitiesInside méretéig
        section = entitiesInside[i].toString()
        message = section
    Ciklus vége
    visszatérés message
Függvény vége
```

- + **String listItems()**: Listába szedi a szobában lévő Itemeket, és ebből álló stringgel tér vissza

Pszudokód:

```
Függvény listItems():
    string message
    string section
    Ciklus 0-tól containedItems méretéig
        section = containedItems[i].toString()
        message = section
    Ciklus vége
    visszatérés message
Függvény vége
```

- + **boolean initiatePickup(boolean pickupAll, Entity e, String[] input, int freeSpace)**: Item pickup parancsának kezelése

Pszudokód:

```
Függvény initiatePickup(boolean pickupAll, Entity e, String[] input, int freeSpace):
    Ha isSticky igaz akkor
        GameController.errorMsg(203)
    Ha pickupAll igaz és containedItems mérete <= freespace akkor
        Ciklus 0-tól containedItems méretéig
            containedItems[i].pickUp(e)
        Ciklus vége
        GameContoroller.statusMsg(214)
        visszatérés igaz
    Különben ha pickupAll igaz és containedItems mérete > freespace akkor
        GameController.errorMsg(209)
        visszatérés hamis
    Item grabbedItem
```



```

    int tragetItemID = input[1]
    Ciklus 0-tól containedItemes méretéig
        Ha containedItems[i].checkID(targetID) akkor
            grabbedItemID = i
            kitörés
    Ciklus vége
    GameController.statusMsg(211)
    visszatérés grabbedItemID(e)
Függvény vége

```

- **+ String debug():** Debug szöveg generálása
Pszudokód:

```

Függvény debug():
    string retVal
    Ciklus 0-tól containedItems mérete
        retVal = retVal + containedItems[i]
    Ciklus vége
    Ciklus 0-tól entititesInside mérete
        retVal = retVal + entititesInside[i]
    Ciklus vége
    Ciklus 0-tól doors mérete
        retVal = retVal + doors[i]
    Ciklus vége
    visszatérés retVal
Függvény vége

```

8.1.16 SlideRule

- **Felelősség**

Az osztály reprezentálja a logarléc objektumot.

- **Ősosztályok**

Item

- **Interfészek**

Debuggable

- **Attribútumok**

- **Metódusok**

- **+ SlideRule(in aID)** - Az osztály konstruktora, beállítja a tárgy ID-ját. aID - A tárgy ID-ja.
- **+ pickUp(Entity e) : boolean** - függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity "pickUpItem" rá vonatkozó változtatát. (visitor pattern). e - Az Entity, aki felveszi a Logarlécet. Visszatérés: Az Entity erre a tárgyra használt pickUpItem függvényének visszatérési értéke.
- **+ drop(Entity e) : boolean** - A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity "dropItem" rá vonatkozó változtatát. (visitor pattern) e - Az Entity, aki lerakja a Logarlécet. Visszatérés: False, mivel a Logarléc felvétele a játék végét jelenti, ezért nincs értelme az eldobásnak.
- **+String toString():** Visszatér egy Stringgel ami a maga nevéből és az azonosítójából áll.
- **+String debug():** Debug szöveg generálása.

8.1.17 Student

- **Felelősség**
A játékos által irányított Entity karaktert reprezentáló osztály.
- **Ősosztályok**
Az Entity osztályból származik.
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Az osztálynak nincsenek attribútumai.
- **Metódusok**
 - **+Student(String, int, Room):** A diák osztály konstruktora. Argumentumként vár egy stringet, amely a diák nevéként, egy egész számot, amely a diák egyedi azonosítójaként, valamint egy Room referenciát, amely a diák jelenlegi tartózkodási pozíciójaként kerül beállításra. Mindezt teszi úgy, hogy meghívja az őosztály konstruktorát az adott értékekkel. Továbbá hozzáadja a referenciáját a GameController debuggableObjects listájához.
 - **+pickUpItem(SlideRule): boolean** Kezeli a logarléc felvevési logikáját a diák karakter számára. Elsősorban ellenőrzi, hogy a diák hátizsákjában elfér-e a logarléc. Ha nem hamissal tér vissza. Ellenkező esetben meghívja az Entity tartózkodási szobáján a removeItem metódust a logarléc referenciájával. Ha ennek a visszatérése nem egy null referencia akkor a logarlécezt hozzáadja saját hátizsákjához, és meghívja a GameController gameWon() metódusát ezzel jelezve hogy a játéknak vége a diákok győzelmével. Ellenkező esetben ismét hamissal tér vissza.

```
Függvény pickUpItem(SlideRule slideRule)
    Ha canPickUp() == hamis akkor:
        visszatérés hamis
    Ha vége

    slideRuleRemoved = currentRoom.removeItem(slideRule)
    Ha slideRuleRemoved == null akkor:
        Visszatérés hamis
    Ha vége

    addItem(slideRuleRemoved)
    GameController.getInstance().gameWon()
    visszatérés igaz
Függvény vége
```

- **+pickUpItem(Transistor): boolean** Tranzisztor felvételének logikája, megegyezik egy általános tárgy felvételének logikájával, hiszen a diák számára megengedett a tranzisztor felvétele.

Függvény pickUpItem(Transistor transistor):

Ha canPickUp() == hamis akkor:

 visszatérés hamis

Ha vége

tranRemoved = currentRoom.removeItem(transistor);

Ha tranRemoved == null akkor:

 visszatérés hamis

Ha vége

addItem(tranRemoved)

 visszatérés igaz

Függvény vége

- **+meet(Teacher):void** A diák egy tanárral való találkozását kezeli. A találkozó következményekkel jár, a diák megpróbálja megmenteni a lelkét a nála lévő tárgyak segítségével. Ehhez meghívja a save() metódusát.
- **+meetMe(Entity):** Az ösosztályban lévő metódus felüldefiniálása. A paraméterül kapott Entityre meghívja annak meet metódusát a saját referenciájával.
- **+save(): void** A diák lelkét hivatott megmenteni. Végignézi a diák inventoryját, hogy van-e nála bármilyen lélekmentő ami használható. Ehhez a játékos inventoryjában lévő összes tárgyon végig meghívja azoknak a save() függvényét. Ha ezek közül valamelyik true értékkel tér vissza a diák lelke megmenekült. Ha a diák egy ilyen tárggyal, vagy egyetlen tárggyal sem rendelkezik, az isAlive attribútum értéke false lesz, valamint meghívja a GameController addALostSoul() metódusát ezzel jelezve hogy a diák elvesztette lelkét. Pszeudokód:

```
public void save(){
```

```
    Ciklus végig minen item-en az inventory-ban:
```

```
        Ha item.save(this) == igaz akkor:
```

```
            visszatérés
```

```
        Ha vége
```

```
    Ciklus vége
```

```
    setIsAlive(false);
```

```
    GameController.getInstance().addALostSoul();
```

```
    Függvény vége
```

- **+saveDrop(): void** Az ösosztályban található metódus felüldefiniálása. Akkor hívandó ha a diák életét egy sör mentette meg. A diák hátizsákjában található tárgyakon kezdve az utolsótól haladva visszafele az elsőig megpróbál eldobni egyet. Az első sikeres eldobásnál véget ér a metódus futása. Pszeudokód:

Függvény saveDrop():

```

    Ciklus i az inventory méretétől csönnekően 1 ig:
        dropWasSuccesful = inventory.get(i).drop(this);
        Ha dropWasSuccesful == igaz akkor:
            visszatérés
        Ha vége
    Ciklis vége
Függvény vége

```

- **+newEntRound(): void** Kezeli egy diák játékos általi irányítását. Kezdetben a roundIsOver értékét beállítja hamisra. Továbbá a GameController currentEntity tagváltozójának értékét saját maga referenciájára. Kiírja a standard kimenetre, hogy éppen vele történik az interakció. Ezt követően ellenőrzi, hogy a diák életben van-e még. Ha nincs ezt a standard kimeneten tudtára adja a játékosnak és vár tőle egy enter billentyű lenyomást a továbblépéshez. Ha megnyomta a játékos az roundIsOver értéke igazra változik. Másodsorban ha a diák még életben van de a remainingStun értéke nagyobb mint 0, akkor ebben a körben az adott diák nem cselekedhet. Az előzőhöz hasonlóan ezt tudtára adja a felhasználónak és vár egy entert. Ha ezek közül egyik sem áll fenn, akkor a GameController scannerjét használva inputot kér a játékostól, majd ezt az inputot továbbadja a GameController terminal metódusának. Ezt egészen addig csinálja amíg a felhasználó egy sikeres mozgás műveletet nem végez, melynek során a roundIsOver igaz értékre változik. Végén ha a remainingStun értéke nem nulla, akkor annak értékét csökkenti eggyel. Visszatér a diák lelkének meglételének logikai értékével. Pszeudokód:

```

Függvény newEntRound() {
    roundIsOver = false
    GameController.getInstance().currentEntity = this
    playerInputString = null

    kiírás "You're currently playing as: " + name + " in room : "
+ currentRoom.getID()

    ciklus ameddig roundIsOver == hamis:
        Ha isAlive == hamis akkor:
            Kírás "Your character does not have a soul
anymore.\nPres enter to continue:"
            Enter bemenetre várunk
            roundIsOver = true
        Különben Ha remainingStun > 0 akkor:
            kiírás "Your character is currently stunned.
Remaining stunned rounds: " + remainingStun + "\nPres enter to continue:"
            Enter bemenetre várunk
            roundIsOver = true
        Különben
            Kiírás name + ">"
            playerInputString = Következő sor a felhasználói
bemenetről

```

```

    GameController.getInstance().terminal(playerInputString);
        Ha vége
    Ciklus vége

    Ha remainingStun > 0 akkor:
        remainingStun--
    Ha vége

    visszatérés getIsAlive();
Függvény vége

```

- **+encounterRag(): void** Egy diák és egy ledobott aktív rongy interakcióját kezeli. Üres, hiszen a diákra nincs hatással a ledobott rongy.
- **+toString(): String** Visszatér az alábbi kimenettel: "Student: {diák neve}"

8.1.18 Teacher

- **Felelősség**
A játék egy nem játszható karaktere, célja hogy megakadályozza a diákokat a logarléc megszerzésében
- **Ősosztályok**
Az Entity osztályból származik
- **Interfészek**
Nem valósít meg interfészt.
- **Attribútumok**
Az osztálynak nincsenek attribútumai
- **Metódusok**
 - **+Teacher(String, int, Room)** A diák osztály konstruktora. Argumentumként vár egy stringet, amely a diák neveként, egy egész számot, amely a diák egyedi azonosítójaként, valamint egy Room referenciát, amely a diák jelenlegi tartózkodási pozíciójaként kerül beállításra. Mindezt teszi úgy, hogy meghívja az ősosztály konstruktort az adott értékekkel. Továbbá hozzáadja a referenciáját a GameController debuggableObjects listájához.
 - **+pickUpItem(SlideRule): boolean** Logarléc felvételének logikája, mivel egy tanár nem vehet fel logarlécet ezért alapértelmezetten hamis értékkel tér vissza.
 - **+pickUpItem(Transistor): boolean** Tranzisztor felvételének logikája, mivel egy tanár nem vehet fel tranzisztort ezért alapértelmezetten hamis értékkel tér vissza.
 - **+meet(Student): void** Diák tanárral való találkozását kezeli oly módon, ha a tanár nincs elkábulva meghívja a paraméterül kapott diákon annak meet metódusát a saját referenciájával.

- **+meetMe(Entity): void** Az ősosztályban lévő metódus felüldefiniálása. A paraméterül kapott Entityre meghívja annak meet metódusát a saját referenciájával.
- **+newEntRound(): boolean** Két féle viselkedése lehetséges. Ha a játék tesztelési módban fut, akkor működése megegyezik a diák azonos nevű metódusában leírtakkal. Ha nem akkor véletlen számú itemet megpróbál felvenni a szoba tárgyai közül, a tárgy kiválasztásához meghívja a tartrózkodási szobáján a giveAnitem metódust. Ezt követően megpróbál néhányszor a szoba ajtajai közül valamelyikeken átmenni. Ha sikeresen szobát változtatott a körének vége. Ha nem akkor a következő körét is a jelenlegi szobában folytatja. Pszeudokód:

```

public boolean newEntRound():

    roundIsOver = false;
    GameController.getInstance().currentEntity = this

//Ha a játék teszt módban van a tanárokkat is lehet irányítani.
Ha GameController.getInstance().testmode == igaz akkor:
    playerInputString = null

Kiírás "You're currently playing as Teacher: " + name + " in
room : " + currentRoom.getID()

    ciklus ameddig roundIsOver == hamis:
        Ha isAlive == hamis akkor:
            Kírás "Your character does not have a soul anymore.\nPres
enter to continue:"
                Enter bemenetre várunk
                roundIsOver = true
            Különben Ha remainingStun > 0 akkor:
                kiírás "Your character is currently stunned. Remaining
stunned rounds: " + remainingStun + "\nPres enter to
continue:"
                    Enter bemenetre várunk
                    roundIsOver = true
            Különben
                Kiírás name + ">"
                    playerInputString = Következő sor a
felhasználói bemenetről

GameController.getInstance().terminal(playerInputString);
    Ha vége
Ciklus vége

Különben:
    pickupTry = (GameController.getInstance().testmode) ?
GameController.getInstance().testEntityPickupCount :
GameController.getInstance().random.Következő véletlen szám 0 és 3 között

```

```

        moveTry = (GameController.getInstance().testmode) ?
GameController.getInstance().testEntityMovementCount :
GameController.getInstance().random.Következő véletlen szám 0 és 3 között

```

```

    Ha remainingStun <= 0 akkor:

```

```

        //Random próbálkozik a felvétellel

```

```

        ciklus i 0-tól a puckuptry-ig:

```

```

            itemToPickUp = currentRoom.giveAnItem()

```

```

            Ha itemToPickUp != null akkor:

```

```

                itemToPickUp.pickUp(this)

```

```

            Ha vége

```

```

        Ciklus vége

```

```

        //Random megpróbál továbbmenni

```

```

        moveWasSuccesful = currentRoom.tryToMove(moveTry, this,
currentRoom)

```

```

        //Ha nem sikerült semelyik ajtón átmenni maradunk
itt.

```

```

        Ha moveWasSuccesful == hamis akkor:

```

```

            currentRoom.acceptEntity(this)

```

```

        Ha vége

```

```

    Ha vége

```

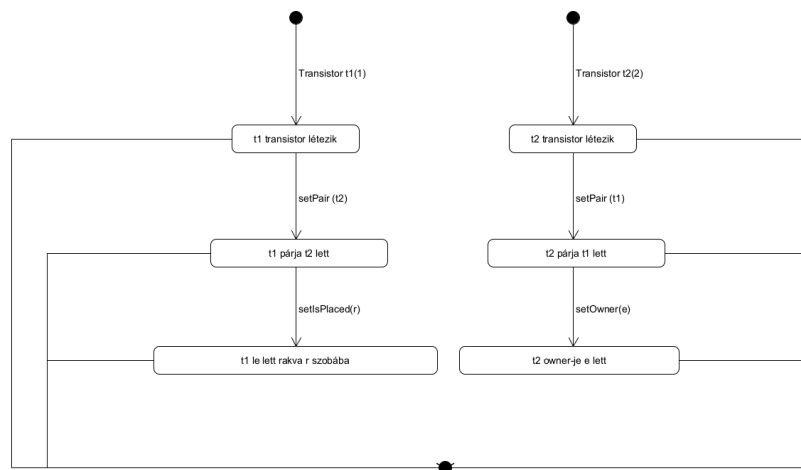
```

Függvény vége

```

- **+encounterRag(): void** Egy tanár és egy ledobott aktív rongy interakcióját kezeli. Következménye ennek az interakciónak, hogy a tanár remainingStun értékét eggyel növeli.
- **+toString(): String** Visszatér az alábbi kimenettel: "Teacher: {tanár neve}"

8.1.19 Transistor



- **Felelősség**

A Transistor Item-mel végzett műveleteket kezelő osztály.

- **Őosztályok**

Item osztály

- **Interfészek**

Debuggable.

- **Attribútumok**

- **-Transistor pair:** Az adott Transistor párját tároló attribútum
- **-Room isPlaced:** A szoba referenciája, ha a Tranzisztor le van helyezve.
- **-Entity owner:** A Tranzisztort használó Entity.

- **Metódusok**

- **+Transistor(int aID):** Az osztály konstruktora, beállítja a tárgy ID-ját.
 pszeudokód:
 Függvény Transistor(int aID)
 setDurability(1)
 GameController.getInstance().debuggableObject.put(Transistor példány)
 Függvény vége
- **+Transistor getPair():** A pair attribútum gettere.
 pszeudokód:
 Függvény getPair()
 visszatérés pair
 Függvény vége

- **+void setPair(Transistor t):** A pair attribútum settere.
pszeudokód:
Függvény setPair(Transistor t)
 pair = t
Függvény vége
- **+Room getIsPlaced():** Az isPlaced attribútum gettere.
pszeudokód:
Függvény getIsPlaced()
 visszatérés isPlaced
Függvény vége
- **+Room setIsPlaced(Room r):** Az isPlaced attribútum settere.
pszeudokód:
Függvény setIsPlaced(Room r)
 isPlaced = r
 visszatérés r
Függvény vége
- **+Entity getOwner():** Az owner attribútum gettere.
pszeudokód:
Függvény getOwner()
 visszatérés owner
Függvény vége
- **+void setOwner(Entity e):** Az owner attribútum settere.
pszeudokód:
Függvény setOwner(Entity e)
 owner = e
Függvény vége
- **+boolean use(Room r, Entity e):** Ez a függvény egy összekapcsolt tranzisztort helyez el a paraméterben megadott szobában, ezáltal lehetővé teszi a későbbi oda teleportálást.
pszeudokód:
Függvény use(Room r, Entity e)
 Ha pair == null vagy isPlaced != null vagy pair.getIsPlaced != null
 visszatérés hamis
 setIsPlaced(r)
 e.removeItem(Transistor példány)
 r.addItem(Transistor példány)
 visszatérés igaz

Függvény vége

- **+boolean connect(Transistor t, Entity e):** Ez a függvény összekapcsolja a tranzisztort egy másik Transistor típusú objektummal.

pszeudokód:

```
Függvény connect(Transistor t, Entity e)
    Ha pair != null
        visszatérés hamis
    Transistor példány.owner = ent
    Transistor példány.pair = t
    t.connect(Transistor példány, ent)
    visszatérés igaz
```

Függvény vége

- **+boolean disconnect(Tranistor t):** Ez a függvény szétkapcsolja az adott tranzisztort a párjától.

pszeudokód:

```
Függvény disconnect(Transistor t)
    Ha t == null vagy pair == null
        visszatérés hamis
    Ha !(pair.equals(t))
        GameController.getInstance().errorMsg(503, "" + this.getID() +
        ", " + t.getID())
        visszatérés hamis
    Transistor példány.pair = null
    Transistor példány.owner = null
    t.disconnect(Transistor példány)
    visszatérés igaz
```

Függvény vége

- **boolean teleport(Entity e, Room r):** A játékos szobaváltoztatásáért felelős tranzistorok használata segítségével.

pszeudokód:

```
Függvény teleport(Entity e, Room r)
    Ha pair == null vagy e != Transistor példány.owner
        visszatérés hamis
    Egyébként Ha Transistor példány.getIsPlaced() == null
        boolean rep = pair.teleportThrough(e)
        Ha rep
            Transistor példány.use(r, e)
        visszatérés rep
    Egyébként
        visszatérés hamis
```

Függvény vége

- **boolean teleportThrough(Entity ent):** Megnézi hogy leván-e a rakva az adott tranzisztor (ha nem akkor false-al tér vissza), ezután meghívja annak a szobának az acceptEntity függvényét amelyben leván téve majd elindít egy pickup folyamatot.

pszeudokód:

```
Függvény teleportThrough(Entity ent)
    Ha Transistor példány.isPlaced != null
        isPlaced.acceptEntity(ent)
        Transistor példány.pickup(ent)
        visszatérés igaz
    Egyébként
        visszatérés hamis
Függvény vége
```

- **boolean pickUp(Entity e):** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity “pickUpItem” rá vonatkozó változtatát. (visitor pattern)

pszeudokód:

```
Függvény pickUp(Entity e)
    Ha Transistor példány.owner != null és !Transistor
    példány.owner.equals(e)
        visszatérés hamis
    Egyébként
        Transistor példány.isPlaced = null
        boolean res = e.pickupItem(Transistor példány)
        visszatérés res
Függvény vége
```

- **boolean drop(Entity e):** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity “dropItem” rá vonatkozó változtatát. (visitor pattern)

pszeudokód:

```
Függvény drop(Entity e)
    Ha Transistor példány.getPair() != null
        visszatérés hamis
    visszatérés e.dropItem(Transistor példány)
Függvény vége
```

- **String toString():** Visszatér egy Stringgel, az azonosítójával kiegészítve.

pszeudokód:

```
Függvény toString()
    visszatérés “Transistor: #” + Integer.toString(Transistor
    példány.getID())
Függvény vége
```

- **String getTeleport():** Visszatér azzal, hogy ha a párja le van helyezve valahol, akkor az hol van. Ha nincsen lehelyezve vagy összekapcsolva egy üres Stringet ad vissza.

pszeudokód:

Függvény getTeleport()

```
Ha Transistor példány.pair != null és Transistor
példány.pair.isPlaced != null
    visszatérés Transistor példány.pair.isPlaced.toString()
visszatérés ""
```

Függvény vége

- **String debug():** Debug szöveg generálása

pszeudokód:

Függvény debug()

```
Ha Object.isNull(pair)
    Transistor példány.pairID = "No pair"
Egyébként
    Transistor példány.pairID = Integer.toString(Transistor
példány.pair.getID())
visszatérés "---- Transistor " + Transistor példány.getID() + "
--\ndurability : " + getDurability() + "\npair : " + pair.ŰID + "\n
--- Transistor " + Transistor példány.getID() + "----\n"
```

Függvény vége

8.1.20 TVSZ

- **Felelősség**

A TVSZ Item-mel végzett műveleteket kezelő osztály.

- **Ősosztályok**

Item osztály

- **Interfészek**

Nincs.

- **Attribútumok**

Nincs.

- **Metódusok**

- **+TVSZ(int aID):** Az osztály konstruktora, beállítja a tárgy ID-ját.

pszeudokód:

Függvény tvsz(int aID)

```
durability = 3
GameController.getInstance.debuggableObjects-be belerakása (ID, tvsz
példány)
```

Függvény vége

- **+boolean pickUp(Entity e):** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity “pickUpItem” rá vonatkozó változtatát. (visitor pattern)

pszeudokód:

Függvény pickUp(Entity e)

 visszatérés e.pickup(tvész példány)

Függvény vége

- **+boolean drop(Entity e):** A függvény célja, hogy a saját referenciájával tudja meghívni a paraméterként kapott Entity “dropItem” rá vonatkozó változtatát. (visitor pattern)

pszeudokód:

Függvény drop(Entity e)

 visszatérés e.dropItem(tvész példány)

Függvény vége

- **+boolean save(Entity ent):** A függvény célja, hogy ha a durability nagyobb mint 0, akkor “true” értékkel tér vissza és levon egyet a durabilityból. Ellenkező esetben a visszatérési érték “false”.

pszeudokód:

Függvény save(Entity ent)

 boolean ret = hamis

 int d = getDurability()

 ha d > 0

 setDurability(d - 1)

 ret = igaz

 visszatérés ret

Függvény vége

- **+String toString():** Visszatér egy Stringgel, az azonosítójával kiegészítve.

pszeudokód:

Függvény toString()

 visszatérés “TVSZ: #” + Integer.toString(getID())

Függvény vége

- **+String debug():** Debug szöveg generálása.

pszeudokód:

Függvény debug()

 visszatérés “---- TVSZ “ + tvész példány.getID() + “ --\ndurability :
 ” + getDurability() + “\n-- TVSZ “ + tvész példány.getID() + “ --\n”

Függvény vége

8.2 Interfacek

- *Debuggable*

Az Objektum kiírásának kezeléséért felelős interface. Egyetlen függvénye a `String debug()`. Visszatérési értéke a `String` reprezentációja. A kiírás formátuma:

```

----<ITEM TYPE> <ITEM ID>----

    <var1> : <value1>

    <varN> : <valueN>

    <List1> :

    - ListItem1

    - ListItem2

    - ListItemN

    <ListN> :

    ...

----<ITEM TYPE> <ITEM ID>----
```

- *Decaying*

A sör és rongy tárgyak közös tulajdonsága, hogy aktiválás után egy adott számú körig érvényes a hatásuk. Ennek az interfésznek a megvalósításával tudjuk ezt a folyamatot jól nyilvántartani és kezelni.

Egyetlen függvénye a `void age()`. Ez a függvény a tárgy korának állításáért felel.

8.3 A tesztek részletes tervei, leírásuk a teszt nyelven

Megjegyzés: A tesztesetekhez tartozó elvárt fájlok tartalma (hosszabb teszteknel a bemenet is) a leírások során oszlopokba lett tördelve a felesleges papírhulladék csökkentése érdekében, azonban éles környezetben, használat közben ezek egymás után külön sorban jelennek meg tördelés nélkül.

8.3.1 Player hozzáadása sikeres

- **Leírás**

A felhasználó felvesz a játékba egy új játékost, majd lekéri a játékosok listáját.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Ellenőrzi, hogy valóban felvehető-e új, egyedi nevű játékos a játékba.
Hibahely lehet a játékos hozzáadása és a kiírása

- **Bemenet**

```

addplayer p1
showplayers
```

- **Elvárt kimenet**

```
Main menu>
(111) Player 'p1' added to the game!
Main menu>
(113) Listing all players who are added to the game:
-p1
```

8.3.2 Player hozzáadása sikertelen

- **Leírás**

A felhasználó hozzáad egy új játékost az *addplayer* paranccsal. Ezek után még egyszer megpróbál hozzáadni egy ugyanolyan nevű játékost. Az első hozzáadás sikeres, a második sikertelen.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrzi, hogy valóban nem lehet-e e kettő azonos nevű játékost hozzáadni a játékhoz. Hibahelyek: játékos hozzáadás, egyediség biztosítása, játékosok kiírása

- **Bemenet**

```
addPlayer testP
addPlayer testP
```

- **Elvárt kimenet**

```
Main menu>
(111) Player 'testP' added to the game!
Main menu>
ERROR -> (101) Player named 'testP' already exists!
```

8.3.3 Player eltávolítása sikeres

- **Leírás**

A felhasználó hozzáad egy játékost az *addplayer* paranccsal, majd eltávolítja a *removeplayer*-el. Ezek után ellenőrzi, hogy valóban nincs játékos a játékban.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrzi, hogy valóban eltávolítható-e egy, már korábban hozzáadott játékos

- **Bemenet**

```
addplayer testP
removeplayer testP
showplayers
```

- **Elvárt kimenet**

```
Main menu>
(111) Player 'testP' added to the game!
Main menu>
(112) Player named ?testP? removed from the game!
Main menu>
(113) Listing all players who are added to the game:
```

8.3.4 Player eltávolítása sikertelen

- **Leírás**

A felhasználó felvesz a játékba egy játékost az *addplayer* paranccsal, majd a *removeplayer* parancsot meghívja egy nem létező játékos nevére.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrzi, hogy valóban csak olyan játékosok távolíthatók el, akik léteznek. Hibahelyek : hozzáadás, eltávolítás

- **Bemenet**
addplayer testP
removeplayer wrongInputTest
- **Elvárt kimenet**
Main menu>
(111) Player 'testP' added to the game!
Main menu>
ERROR -> (102) There is no player named wrongInputTest
Main menu>

8.3.5 Start sikeres

- **Leírás**
A felhasználó felvesz a játékba kettő játékost, majd elindítja a játékot. Ez sikeresen megtörténik. Hibás lehet a felvétel és az indítás
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrzi, hogy a játék elindítható-e, amennyiben fennállnak a szükséges feltételek.
- **Bemenet**
addplayer p1
addplayer p2
start 10 t
- **Elvárt kimenet**
Main menu>
(111) Player 'p1' added to the game!
Main menu>
(111) Player 'p2' added to the game!
Main menu>
The game has been successfully started with the following players:
-p1
-p2

8.3.6 Start sikertelen (nincs elég játékos)

- **Leírás**
A felhasználó hozzáad egy játékost, majd elindítja a játékot. Mivel a játékhoz legalább kettő játékos kell, ezért ez sikertelen.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A teszt ellenőrzi azon szabály betartását, mely szerint legalább kettő hozzáadott játékos szükséges egy új játék indításához.
- **Bemenet**
addplayer p1
start 10 t
- **Elvárt kimenet**
Main menu>
(111) Player 'p1' added to the game!
Main menu>
ERROR -> (103) Not enough players(2) to start!
Main menu>

8.3.7 Load sikeres

- **Leírás**
A felhasználó elindít egy játékot, majd elmenti. Ezek után újraindítja a programot, és a *load* paranccsal visszatölti a mentett állást.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A teszt ellenőrzi, hogy a játékállás egy mentésből helyesen visszatölthető-e
- **Bemenet**
addplayer p1
addplayer p2
start 10 t
save testSave
[újraindítás]
load testSave
- **Elvárt kimenet**
Main menu>
(111) Player 'p1' added to the game!
Main menu>
(111) Player 'p2' added to the game!
Main menu>
The game has been successfully started with the following players:
 -p1
 -p2
Your progress has been successfully saved under the file named 'testSave'.
[újraindítás]
Main menu>
File 'testSave' successfully loaded.

8.3.8 Load sikertelen (Nem létezik ilyen file)

- **Leírás**
A felhasználó elindít egy játékot, majd elmenti. Ezek után újraindítja a programot, és a *load* paranccsal visszatölti a mentett állást egy másik file névre hivatkozva. A parancs sikertelen, mivel nem létező mentés nevet adott meg.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A teszt ellenőrzi, hogy helytelenül megadott mentést nem lehet visszatölteni, a program helyesen kezeli a rossz utasítást. Hibás lehet a mentés kezelése
- **Bemenet**
addplayer p1
addplayer p2
start 10 t
save testSave
[újraindítás]
load wrongInputTest

- **Elvárt kimenet**

```

Main menu>
(111) Player 'p1' added to the game!
Main menu>
(111) Player 'p2' added to the game!
Main menu>
The game has been successfully started with the following players:
    -p1
    -p2
Your progress has been successfully saved under the file named 'testSave'.
[újraindítás]
Main menu>
Error loading file 'wrongInputTest', file not found. Please try again.
```

8.3.9 Save sikeres

- **Leírás**

A felhasználó elindít egy játékot, majd elmenti.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt ellenőrzi, hogy a játékállás elmentésre kerül-e, és hogy ez helyesen történik-e meg. Hibás lehet a játékállás mentése.

- **Bemenet**

```

addplayer p1
addplayer p2
start 10 t
save testSave
```

- **Elvárt kimenet**

```

Main menu>
(111) Player 'p1' added to the game!
Main menu>
(111) Player 'p2' added to the game!
Main menu>
The game has been successfully started with the following players:
    -p1
    -p2
Your progress has been successfully saved under the file named 'testSave'.
```

8.3.10 Rosszkor hívott command

- **Leírás**

A felhasználó elindít egy játékot, majd megpróbál hozzáadni egy új játékost. Mivel a játék már fut, ez sikertelen.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt ellenőrzi, hogy egyes parancsok csak a megfelelő helyzetben hívhatók-e

- **Bemenet**

```

addplayer p1
addplayer p2
start 10 t
addplayer p3
```

- **Elvárt kimenet**

```
Main menu>
(111) Player 'p1' added to the game!
Main menu>
(111) Player 'p2' added to the game!
Main menu>
The game has been successfully started with the following players:
    -p1
    -p2
p1>
Error, this command can not be called right now.
```

8.3.11 Show -i sikeres

- **Leírás**

A felhasználó elindít egy játékot, majd felvesz egy TVSZt a földről. Ez sikeresen megtörténik, mivel üres a hátitáskája, és így veszi fel a tárgyat.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt ellenőrzi, hogy a felhasználó valóban képes-e lekérdezni a hátizsákja tartalmát. Hibahelyek: tárgy felvétele, show parancs kiírása

- **Bemenet**

```
taddroom f 1 1
taddentity s "Jancsi" 10 1
tadditem tvsz 999 1
start 10 t
pickup 999
show -i
```

- **Elvárt kimenet**

```
Main menu>
Room successfully added!
Main menu>
Student successfully added!
Main menu>
Item successfully added!
Main menu>
The game has been successfully started with the following players:
    -Jancsi
Jancsi>
(211) Picked up 999!
Jancsi>
(220) Listing player's items!
Inventory:
    -TVSZ: #999
```

8.3.12 Show -d sikeres

- **Leírás**
A felhasználó elindít egy játékot, majd lekérdezi a szobájából nyíló ajtókat. Ez sikeresen megtörténik.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A teszt ellenőrzi, hogy a felhasználó valóban képes-e megjeleníteni az ajtókat, amiken keresztül elhagyhatja a tartózkodási helyét, amennyiben járható a megfelelő irányban.
- **Bemenet**
taddroom f 1 1
taddroom f 1 2
tadddoor 1 2 f b 20
taddentity s "Jancsi" 10 1
start 10 t
show -d
- **Elvárt kimenet**
Main menu>
Room successfully added!
Main menu>
Room successfully added!
Main menu>
Door successfully added!
Main menu>
Student successfully added!
Main menu>
The game has been successfully started with the following players:
-Jancsi
Jancsi>
(221) Listing all doors in current room!
Doors:
-Door: #20 leading to Room #2

8.3.13 Show -e sikeres

- **Leírás**
A felhasználó elindít egy játékot, egy másik játékosal egy szobába mennek. Ezután a másik játékos meghívja a show -e parancsot.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A teszt ellenőrzi, hogy a játékos valóban képes-e megjeleníteni a vele egy szobában tartózkodó entitásokat.
- **Bemenet**
taddroom f 5 1
taddroom f 5 2
tadddoor 1 2 f b 111
taddentity s "Jancsi" 6 1
taddentity s "Erik" 7 1
start 10 t
move 111
move 111
show -e

- **Elvárt kimenet**

```

Main menu>
Room successfully added!
Main menu>
Room successfully added!
Main menu>
Door successfully added!
Main menu>
Student successfully added!
Main menu>
Student successfully added!
Main menu>
The game has been successfully started with the following players:
    -Jancsi
    -Erik
Jancsi>
Successfully moved to Room:2.
Erik>
Successfully moved to Room:2
(222) Listing entities in current room!
Entities:
    -Student: Jancsi
    -Student: Erik

```

8.3.14 Show -ri sikeres

- **Leírás**

Egy diák tartózkodik egy olyan szobában, ahol a földön található egy Sör és egy TVSZ. A felhasználó beüti a *show -ri* parancsot, a tárgyak pedig megjelennek a konzolon.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A teszt ellenőrzi, hogy a felhasználó képes-e megjeleníteni a tartózkodási helyén földön található tárgyakat.

- **Bemenet**

```

taddrroom f 5 1
taddentity s "Jancsi" 6 1
tadditem beer 100 1
tadditem tvsz 123 1
start 10 t
show -ri

```

- **Elvárt kimenet**

```

Main menu>
Room successfully added!
Main menu>
Student successfully added!
Main menu>
Item successfully added!
Main menu>
Item successfully added!
Main menu>
The game has been successfully started with the following players:
    -Jancsi
(223) Listing items in current room!
Items:
    -Beer: #100
    -TVSZ: #123

```

8.3.15 Command után rossz kapcsoló

- **Leírás**
A játékos meghívja a `show -x` parancsot. Az `x` egy érvénytelen kapcsoló, hibaüzenetet ad a rendszer.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A teszt ellenőrizni, hogy helytelen kapcsolókkal ellátott parancsok esetén is helyesen fusson a program, és tájékoztassa a felhasználót a hibás kérésről.
- **Bemenet**

```
addplayer p1
addplayer p2
start 10 t
show -x
```
- **Elvárt kimenet**

```
Main menu>
(111) Player 'p1' added to the game!
Main menu>
(111) Player 'p2' added to the game!
Main menu>
The game has been successfully started with the following players:
    -p1
    -p2
Error, nonexistent switch for this command.
```

8.3.16 Pick up teszteteset sikeres felvétel

- **Leírás**
Egyetlen diák tartózkodik a pálya egyik szobájában. A szobában megtalálható egy TVSZ tárgy. A szoba nem ragacsos, így a benne lelhető tárgyakat fel lehet venni. A diák hátizsákja üres. Kiválasztja a szoba tárgyai közül a TVSZ-t. Mivel a szoba nem ragacsos, és a diák hátizsákja sincs tele, így a tárgy kikerül a szoba tárgyai közül, és belekerül a diák hátizsákjába. A tárgyfelvétel sikeres.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A tárgyfelvétel üres hátizsák és nem ragacsos szoba esetén helyesen működik
- **Bemenet**

```
taddroom f 5 1
taddentity s Jancsi 2 1
tadditem tvsz 3 1
start t
pickup 3
debug 0
```
- **Elvárt kimenet**

| | | |
|------------------|---------------------|-------------------|
| ---- Map ---- | capacity : 5 | entitiesInside : |
| rooms : | hasBeenCleaned : | ---- Student 2 |
| ---- Room 1 ---- | false | ---- |
| isToxic : false | entitiesSinceCleanu | name : Jancsi |
| isMerged : false | p : 0 | remainingStun : 0 |
| isSticky : false | containedItems : | currentRoom : 1 |

```

isAlive : true          ---- TVSZ 3 ----      doors :
isInToxic : false       durability : 3         ---- Room 1 ----
roundIsOver : false     ---- TVSZ 3 ----      ---- Map ----
inventory :             ---- Student 2 ----

```

8.3.17 Pickup tesztet sikertelen felvétel (teli hátizsák)

- **Leírás**

Egyetlen diák tartózkodik a pálya egyik szobájában. A szobában megtalálható egy TVSZ tárgy. A szoba nem ragacsos, így a benne lelhető tárgyakat fel lehet venni. A diák hátizsákja tele van, azaz pontosan öt tárgy van nála. Kiválasztja a szoba tárgyai közül a TVSZ-t. Mivel a szoba nem ragacsos, viszont a diák hátizsákja tele van, így a tárgy nem kerül ki a szoba tárgyai közül, és nem kerül bele a diák hátizsákjába. A tárgyfelvétel sikertelen

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tele hátizsák és nem ragacsos szoba esetén a tárgyfelvétel nem megy végbe.

- **Bemenet**

```

taddroom f 1                start 20 t
taddentity s Jancsi 2 1     pickup 3
tadditem tvsz 3 1           pickup 4
tadditem mask 4 1           pickup 5
tadditem beer 5 1           pickup 7
tadditem tvsz 6 1           pickup 8
tadditem transistor 7 1     pickup 6
tadditem transistor 8 1     debug 0

```

- **Elvárt kimenet**

```

---- Map ----              name : Jancsi              ---- Mask 4 ----
rooms :                    remainingStun : 0             ---- Transistor 7
---- Room 1 ----          currentRoom : 1              ----
isToxic : false           isAlive : true              durability : 1
isMerged : false          isInToxic : false            pair : No pair
isSticky : false          roundIsOver : false          ---- Transistor 7
capacity : 5              inventory :              ----
hasBeenCleaned :         ---- Beer 5 ----             ---- Transistor 8
false                    durability : 5              ----
entitiesSinceCleanu       isActive : false            durability : 1
p : 0                    ---- Beer 5 ----            pair : No pair
containedItems :          ---- TVSZ 3 ----             ---- Transistor 8
---- TVSZ 6 ----          durability : 3              ----
durability : 3            ---- TVSZ 3 ----             ---- Student 2 ----
---- TVSZ 6 ----          ---- Mask 4 ----             doors :
entitiesInside :          durability : 5              ---- Room 1 ----
---- Student 2            lastRoomWasToxic :    ---- Map ----
----                      false

```

8.3.18 Pickup tesztet sikertelen felvétel (nincs ilyen tárgy a szobában)

- **Leírás**

Egyetlen diák tartózkodik a pálya egyik szobájában. A szobában nem található meg egyetlen tárgy sem. A szoba nem ragacsos, így a benne lelhető tárgyakat fel lehet venni. A diák hátizsákja üres. A 'pickup' parancs segítségével megpróbál felvenni egy tárgyat, amelynek tudja az egyedi azonosítóját, viszont az a tárgy nincsen a szoba tárgyai között. Mivel a szoba tárgyai között nincs a kiválasztott tárgy, ezért a tárgyfelvétel sikertelen lesz. A tárgy nem kerül be a szoba tárgyai közé, és nem kerül bele a diák hátizsákjába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Nem lehet olyan tárgyat felvenni, amely nincsen a szoba tárgyai között.

- **Bemenet**

```
taddroom f 5 1
taddroom f 5 2
tadddoor 1 2 f b 3
taddentity s Jancsi 4 1
tadditem tvsz 5 2
start 20 t
pickup 5
debug 0
```

- **Elvárt kimenet**

```
---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 5
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 4 ----
doors :
---- Room 1 ----
---- Room 2 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 5
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
---- TVSZ 5 ----
durability : 3
---- TVSZ 5 ----
entitiesInside :
doors :
---- Room 2 ----
---- Map ----
```

8.3.19 Pickup tesztet sikertelen felvétel (a szoba ragacsos)

- **Leírás**

Hat diák tartózkodik a pályán és egy takarító. A takarító átlép egy másik szobába, majd utána mind a 6 diák is. A szobában megtalálható egy TVSZ tárgy. A szoba ragacsos, így a benne lelhető tárgyakat nem lehet felvenni. A diákok hátizsákja üres. Kiválasztja valamelyik diák a szoba tárgyai közül a TVSZ-t. Mivel a diák hátizsákja sincs tele, viszont a szoba ragacsos, így a tárgy nem kerül ki a szoba tárgyai közül, és nem kerül be a diák hátizsákjába. A tárgyfelvétel sikertelen.

- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ragacsos szobából nem lehet tárgyat felvenni.
- **Bemenet**

| | | |
|---------------------|--------------------|-----------|
| taddroom f 15 1 | taddentity s | move |
| taddroom f 15 2 | Jancsi6 9 1 | move |
| tadddoor 1 2 f b 3 | taddentity s | move 3 |
| taddentity s Jancsi | Jancsi7 10 1 | move 3 |
| 4 1 | taddentity j Bela4 | move 3 |
| taddentity s | 11 1 | move 3 |
| Jancsi2 5 1 | tadditem tvsz 12 2 | move 3 |
| taddentity s | start 20 t | move 3 |
| Jancsi3 6 1 | move | move 3 |
| taddentity s | move | move 3 |
| Jancsi4 7 1 | move | move 3 |
| taddentity s | move | pickup 12 |
| Jancsi5 8 1 | move | debug 0 |

- **Elvárt kimenet**

| | | |
|---------------------|---------------------|---------------------|
| ---- Map ---- | - 1 | name : Jancsi2 |
| rooms : | - 2 | remainingStun : 0 |
| ---- Room 1 ---- | ---- Door 3 ---- | currentRoom : 2 |
| isToxic : false | ---- Room 1 ---- | isAlive : true |
| isMerged : false | ---- Room 2 ---- | isInToxic : false |
| isSticky : false | isToxic : false | roundIsOver : true |
| capacity : 15 | isMerged : false | inventory : |
| hasBeenCleaned : | isSticky : true | ---- Student 5 ---- |
| true | capacity : 15 | ---- Student 6 ---- |
| entitiesSinceCleanu | hasBeenCleaned : | name : Jancsi3 |
| p : 0 | true | remainingStun : 0 |
| containedItems : | entitiesSinceCleanu | currentRoom : 2 |
| entitiesInside : | p : 7 | isAlive : true |
| ---- Janitor 11 | containedItems : | isInToxic : false |
| ---- | ---- TVSZ 12 ---- | roundIsOver : true |
| name : Béla | durability : 3 | inventory : |
| remainingStun : 0 | ---- TVSZ 12 ---- | ---- Student 6 ---- |
| currentRoom : 1 | entitiesInside : | ---- Student 7 ---- |
| isAlive : true | ---- Student 4 | name : Jancsi4 |
| isInToxic : false | ---- | remainingStun : 0 |
| roundIsOver : true | name : Jancsi | currentRoom : 2 |
| inventory : | remainingStun : 0 | isAlive : true |
| ---- Janitor 11 | currentRoom : 2 | isInToxic : false |
| ---- | isAlive : true | roundIsOver : true |
| doors : | isInToxic : false | inventory : |
| ---- Door 3 ---- | roundIsOver : false | ---- Student 7 ---- |
| isCursed : false | inventory : | ---- Student 8 ---- |
| direction : BOTH | ---- Student 4 ---- | name : Jancsi5 |
| AvailableRooms : | ---- Student 5 ---- | remainingStun : 0 |

```

currentRoom : 2      roundIsOver : true      ---- Student 10
isAlive : true       inventory :            ----
isInToxic : false    ---- Student 9 ----    doors :
roundIsOver : true    ---- Student 10      ---- Door 3 ----
inventory :          ----                  isCursed : false
---- Student 8 ----   name : Jancsi7        direction : BOTH
---- Student 9 ----   remainingStun : 0     AvailableRooms :
name : Jancsi6         currentRoom : 2      - 1
remainingStun : 0      isAlive : true        - 2
currentRoom : 2        isInToxic : false     ---- Door 3 ----
isAlive : true         roundIsOver : true     ---- Room 2 ----
isInToxic : false      inventory :          ---- Map ----

```

8.3.20 Pickup tesztet sikertelen felvétel (elhelyezett tranzisztor másé)

- **Leírás**

Két diák tartózkodik a pálya egyik szobájában. Az első szobában megtalálható egy Tranzisztor tárgy, melyet egy másik játékos egy korábbi körben helyezett le. A szoba nem ragacsos, így a benne lelhető tárgyakat fel lehet venni. A diák hátizsákja üres. Kiválasztja a szoba tárgyai közül a Tranzisztort-t. Mivel a diák hátizsákja nincs tele és a szoba sem ragacsos, viszont a Tranzisztor tárgy nem az ő tárgya, így a tárgy nem kerül ki a szoba tárgyai közül, és nem kerül be a diák hátizsákjába. A tárgyfelvétel sikertelen.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Egy játékos által lehelyezett tranzisztort más nem veheti fel.

- **Bemenet**

```

taddroom f 15 1
taddroom f 15 2
tadddoor 1 2 f b 3
taddentity s Jancsi 4 1
taddentity s Jancsi2 5 1
tadditem transistor 6 1
tadditem transistor 7 1
start 20 t
pickup 6
pickup 7
connect 6 7
use 6
move
pickup 6
debug 0

```

- **Elvárt kimenet**

```

---- Map ----      ---- Room 1 ----      isMerged : false
rooms :            isToxic : false        isSticky : false

```

```

capacity : 15          ---- Transistor 7          ---- Door 3 ----
hasBeenCleaned :      ----                      ---- Room 1 ----
false                durability : 1              ---- Room 2 ----
entitiesSinceCleanu  pair : 6                    isToxic : false
p : 0                ---- Transistor 7          isMerged : false
containedItems :      ----                      isSticky : false
---- Transistor 6     ---- Student 4 ----      capacity : 15
----                ---- Student 5 ----      hasBeenCleaned :
durability : 1        name : Jancsi2             false
pair : 7              remainingStun : 0          entitiesSinceCleanu
---- Transistor 6     currentRoom : 1            p : 0
----                isAlive : true              containedItems :
entitiesInside :      isInToxic : false          entitiesInside :
---- Student 4        roundIsOver : false        doors :
----                inventory :                  ---- Door 3 ----
name : Jancsi         ---- Student 5 ----      isCursed : false
remainingStun : 0     doors :                    direction : BOTH
currentRoom : 1       ---- Door 3 ----          AvailableRooms :
isAlive : true        isCursed : false           - 1
isInToxic : false     direction : BOTH           - 2
roundIsOver : true    AvailableRooms :          ---- Door 3 ----
inventory :           - 1                      ---- Room 2 ----
                    - 2                      ---- Map ----

```

8.3.21 Drop tesztelés sikeres

- **Leírás**

Egyetlen diák tartózkodik a pálya egyik szobájában. A szobában nem található meg egyetlen tárgy sem. A diák hátizsákjában van egy TVSZ tárgy. Kiválasztja a hátizsákjában lévő tárgyak közül a TVSZ-t. Mivel a diák hátizsákja nem üres, és megtalálható benne az eldobni kívánt TVSZ tárgy, így az eldobás sikeres. A tárgy kikerül a diák hátizsákjából, és bekerül a szoba tárgyai közé.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tárgy eldobása helyesen működik.

- **Bemenet**

```

taddroom f 15 1
taddentity s Jancsi 2 1
tadditem tvsz 3 1
start 20 t
pickup 3
drop 3
debug 0

```

- **Elvárt kimenet**

```

---- Map ----          isMerged : false          hasBeenCleaned :
rooms :                isSticky : false          false
---- Room 1 ----      capacity : 15              entitiesSinceCleanu
isToxic : false        p : 0

```

```

containedItems :      name : Jancsi      ---- Student 2 ----
---- TVSZ 3 ----      remainingStun : 0      doors :
durability : 3         currentRoom : 1      ---- Room 1 ----
---- TVSZ 3 ----      isAlive : true      ---- Map ----
entitiesInside :      isInToxic : false
  ---- Student 2      roundIsOver : false
  ----              inventory :

```

8.3.22 Drop tesztet sikertelen (nincsen ilyen tárgy az inventoryban)

- **Leírás**

Egyetlen diák tartózkodik a pálya egyik szobájában. A szobában nem található meg egyetlen tárgy sem. A diák hátizsákja üres. Valahonnan megszerzi egy TVSZ tárgy azonosítóját, ami nincs benne a hátizsákjában. A 'drop' parancs segítségével megpróbálja eldobni azt a TVSZ tárgyat, amely nincs nála. Mivel a tárgy nem található meg a diák hátizsákjában, így az eldobás sikertelen. A tárgy nem kerül bele a diák hátizsákjába, és nem kerül be a szoba tárgyai közé sem

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Olyan tárgyat nem lehet eldobni, amely nincs benne a hátizsákban.

- **Bemenet**

```

taddroom f 15 1
taddroom f 15 2
taddentity s Jancsi 3 1
tadditem tvsz 4 1
start 20 t
drop 4
debug 0

```

- **Elvárt kimenet**

```

---- Map ----      name : Jancsi      hasBeenCleaned :
rooms :             remainingStun : 0      false
---- Room 1 ----      currentRoom : 1      entitiesSinceCleanu
isToxic : false      isAlive : true      p : 0
isMerged : false      isInToxic : false      containedItems :
isSticky : false      roundIsOver : false      ---- TVSZ 4 ----
capacity : 15          inventory :      ---- TVSZ 4 ----
hasBeenCleaned :      ---- Student 3 ----      entitiesInside :
false                doors :      doors :
entitiesSinceCleanu  ---- Room 1 ----      ---- Room 2 ----
p : 0                ---- Room 2 ----      ---- Map ----
containedItems :      isToxic : false
entitiesInside :      isMerged : false
  ---- Student 3      isSticky : false
  ----              capacity : 15

```

8.3.23 Drop tesztet sikertelen (maszk gázos szobában)

- **Leírás**

Egyetlen diák tartózkodik a pálya egyik szobájában. A szobában nem található meg egyetlen tárgy sem, viszont egy gázos szoba. A diák hátizsákjában van egy nem nulla tartósságú Maszk tárgy. Kiválasztja a hátizsákjában lévő tárgyak közül a Maszkot. Mivel a diák hátizsákja nem üres és megtalálható benne az eldobni kívánt Maszk, viszont a szoba amiben éppen egy maszkot szeretné eldobni gázos, így az eldobás sikertelen. A tárgy nem kerül ki a diák hátizsákjából, és nem kerül be a szoba tárgyai közé.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Gázos szobában maszk eldobására nincs lehetőség.

- **Bemenet**

```
taddroom f 15 1
taddroom f 15 2
tadddoor 1 2 f b 3
taddentity s Jancsi 4 1
tadditem mask 5 1
start 20 t
pickup 5
move 3
drop 5
debug 0
```

- **Elvárt kimenet**

| | | |
|---------------------|---------------------|---------------------|
| ---- Map ---- | ---- Door 3 ---- | roundIsOver : false |
| rooms : | ---- Room 1 ---- | inventory : |
| ---- Room 1 ---- | ---- Room 2 ---- | ---- Mask 5 ---- |
| isToxic : false | isToxic : true | durability : 4 |
| isMerged : false | isMerged : false | lastRoomWasToxic : |
| isSticky : false | isSticky : false | true |
| capacity : 15 | capacity : 15 | ---- Mask 5 ---- |
| hasBeenCleaned : | hasBeenCleaned : | ---- Student 4 ---- |
| false | false | doors : |
| entitiesSinceCleanu | entitiesSinceCleanu | ---- Door 3 ---- |
| p : 0 | p : 0 | isCursed : false |
| containedItems : | containedItems : | direction : BOTH |
| entitiesInside : | entitiesInside : | AvailableRooms : |
| doors : | ---- Student 4 | - 1 |
| ---- Door 3 ---- | ---- | - 2 |
| isCursed : false | name : Jancsi | ---- Door 3 ---- |
| direction : BOTH | remainingStun : 0 | ---- Room 2 ---- |
| AvailableRooms : | currentRoom : 2 | ---- Map ---- |
| - 1 | isAlive : true | |
| - 2 | isInToxic : false | |

8.3.24 Drop tesztet sikeres (aktív sör)

- **Leírás**

Egyetlen diák tartózkodik a pálya egyik szobájában. A szobában nem található meg egyetlen tárgy sem. A diák hátizsákjában van egy aktív Sör tárgy. Kiválasztja a hátizsákjában lévő tárgyak közül a Sört. Mivel a diák hátizsákja nem üres, és megtalálható benne az eldobni kívánt Sör tárgy, így az eldobás sikeres. A tárgy kikerül a diák hátizsákjából, és nem kerül be a szoba tárgyai közé.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Aktív sör eldobása esetén a tárgy nem kerül be a szoba tárgyai közé.

- **Bemenet**

```
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem beer 5 1
start 20 t
pickup 5
use 5
drop 5
debug 0
```

- **Elvárt kimenet**

```
---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 4 ----
doors :
---- Room 1 ----
---- Map ----
```

8.3.25 Drop tesztet sikertelen (összekapcsolt tranzisztor eldobása)

- **Leírás**

Egyetlen diák tartózkodik a pálya egyik szobájában. A szobában nem található meg egyetlen tárgy sem.. A diák hátizsákjában van egy összekapcsolt Tranzisztor tárgy. Kiválasztja a hátizsákjában lévő tárgyak közül a Tranzisztort. Mivel a diák hátizsákja nem üres és megtalálható benne az eldobni kívánt Tranzisztor, viszont a szóban forgó Tranzisztor össze van kapcsolva egy másik Tranzisztor tárggyal, így az eldobás sikertelen. A tárgy nem kerül ki a diák hátizsákjából, és nem kerül be a szoba tárgyai közé

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Összekapcsolt Tranzisztort nem lehet eldobni.

- **Bemenet**

```
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem transistor 6 1
tadditem transistor 7 1
```

```

start 20 t
pickup 6
pickup 7
connect 6 7
drop 6
debug 0

```

- **Elvárt kimenet**

```

--- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :

---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Transistor 6
----
durability : 1
pair : 7

---- Transistor 6
----
Transistor 7
durability : 1
pair : 6
---- Transistor 7
----
Student 4 ----
doors :
---- Room 1 ----
---- Map ----

```

8.3.26 Tárgy használat sikeres(Tranzisztor)

- **Leírás**
Egy diáknál van két, összekapcsolt tranzisztor. Az egyiket lerakja abban a szobában, ahol tartózkodik.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Sikeresen lehelyezi az egyik Tranzisztort.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem transistor 6 1
tadditem transistor 7 1
start 20 t
pickup 6
pickup 7
connect 6 7
drop 6
debug 0

- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :

---- Transistor 6
----
durability : 1
pair : 7
---- Transistor 6
----
---- Transistor 7
----
durability : 1
pair : 6
---- Transistor 7
----
entitiesInside :

---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 4 ----
doors :
---- Room 1 ----
---- Map ----

```

8.3.27 Tárgy használat sikeres(camembert)

- **Leírás**
Egy diáknál van egy camembert. Felhasználva azt az adott szoba gázos szobává változik.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizni, hogy valóban gázos lett-e a szoba.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem camembert 6 1
start 20 t
pickup 6
use 6
debug 0
- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : true
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false

entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1

isAlive : true
isInToxic : true
roundIsOver : false
inventory :
---- Student 4 ----
doors :
---- Room 1 ----
---- Map --

```


8.3.28 Tárgy használat sikeres(sör)

- **Leírás**
Egy diáknál van egy nem aktív söröspohár. Ezt használva aktívvá változtatja az eszközt.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizni, hogy aktívvá változott-e a söröspohár.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem beer 6 1
start 5 t
pickup 6
use 6
debug 0
- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
---- Beer 6 ----
durability : 5
isActive : true
---- Beer 6 ----
entitiesInside :
---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isActive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 4 ----
doors :
---- Room 1 ----
---- Map --

```

8.3.29 Use sikeres(Légfrissítő)

- **Leírás**
Egy diáknál van egy légfrissítő. Egy gázos szobában használva azt, a szoba normálissá válik.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Gázos szoba nem gázos lesz.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem camembert 6 1
tadditem airfreshner 7 1
start t
pickup 6
use 6
debug 0

- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 4 ----
doors :
---- Room 1 ----
---- Map ----

```

8.3.30 Használat sikertelen (nem használható tárgy/nincs ilyen)

- **Leírás**

Egy diáknál van egy olyan tárgy, amelyet nem lehet használni. Ekkor a tárgy használata hamis értéket ad vissza.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrizni, hogy a nem használható tárgyak használatakor valóban nem történik semmi.

- **Bemenet**

```

taddroom f 15 1
taddentity s Jancsi 4 1
tadditem beer 6 1
start t
pickup 6
use 7
debug 0

```

- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
---- Beer 6 ----
durability : 5
isActive : false
---- Beer 6 ----
entitiesInside :
---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 4 ----
doors :
---- Room 1 ----
---- Map ----

```

8.3.31 Használat sikertelen (tranzisztor összekapcsolt állapot)

- **Leírás**
Egy diáknál van két tranzisztor, azonban ezek nincsenek összekapcsolva. Ekkor nem lehet egyiket sem lehelyezni.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizni, hogy nem összekapcsolt tranzisztort ne lehessen lehelyezni.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem transistor 6 1
tadditem transistor 7 1
start t
pickup 6
use 7
debug 0
- **Elvárt kimenet**

| | | |
|---------------------|--------------------------|-----------------------|
| ----- Map ----- | ----- Transistor 6 ----- | ----- Student 4 ----- |
| rooms : | ----- | ----- |
| ----- Room 1 ----- | durability : 1 | name : Jancsi |
| isToxic : false | pair : No pair | remainingStun : 0 |
| isMerged : false | ----- Transistor 6 ----- | currentRoom : 1 |
| isSticky : false | ----- | isAlive : true |
| capacity : 15 | ----- Transistor 7 ----- | isInToxic : false |
| hasBeenCleaned : | ----- | roundIsOver : false |
| false | durability : 1 | inventory : |
| entitiesSinceCleanu | pair : No pair | ----- Student 4 ----- |
| p : 0 | ----- Transistor 7 ----- | doors : |
| containedItems : | ----- | ----- Room 1 ----- |
| | entitiesInside : | ----- Map ----- |

8.3.32 Használat sikertelen (már aktivált sör)

- **Leírás**
Egy diáknál van egy már eleve aktivált sör. Ekkor a use parancsra történő aktiválás nem mehet végbe, így sikertelen az aktiválás.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizni, hogy már eleve aktivált sört ne lehessen még egyszer aktiválni.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem beer 6 1
start t
pickup 6
use 6
debug 0

- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
---- Beer 6 ----
durability : 5
isActive : true
---- Beer 6 ----
entitiesInside :

---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false

```

8.3.33 Move sikeres

- **Leírás**

Egyetlen diák van a pálya egyik szobájában. Kiválaszt a szoba ajtajai közül egyet amin átmegy az annak túloldalán lévő másik szobába. A helyváltztatás sikerül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Entity helyváltztatása

- **Bemenet**

```

taddroom f 15 1
taddroom f 15 2
tadddoor 1 2 f b 3
taddentity s Jancsi 4 1
start t
move 3
debug 0

```

- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
  doors :
---- Door 3 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35
- 36
- 37
- 38
- 39
- 40
- 41
- 42
- 43
- 44
- 45
- 46
- 47
- 48
- 49
- 50
- 51
- 52
- 53
- 54
- 55
- 56
- 57
- 58
- 59
- 60
- 61
- 62
- 63
- 64
- 65
- 66
- 67
- 68
- 69
- 70
- 71
- 72
- 73
- 74
- 75
- 76
- 77
- 78
- 79
- 80
- 81
- 82
- 83
- 84
- 85
- 86
- 87
- 88
- 89
- 90
- 91
- 92
- 93
- 94
- 95
- 96
- 97
- 98
- 99
- 100
- 101
- 102
- 103
- 104
- 105
- 106
- 107
- 108
- 109
- 110
- 111
- 112
- 113
- 114
- 115
- 116
- 117
- 118
- 119
- 120
- 121
- 122
- 123
- 124
- 125
- 126
- 127
- 128
- 129
- 130
- 131
- 132
- 133
- 134
- 135
- 136
- 137
- 138
- 139
- 140
- 141
- 142
- 143
- 144
- 145
- 146
- 147
- 148
- 149
- 150
- 151
- 152
- 153
- 154
- 155
- 156
- 157
- 158
- 159
- 160
- 161
- 162
- 163
- 164
- 165
- 166
- 167
- 168
- 169
- 170
- 171
- 172
- 173
- 174
- 175
- 176
- 177
- 178
- 179
- 180
- 181
- 182
- 183
- 184
- 185
- 186
- 187
- 188
- 189
- 190
- 191
- 192
- 193
- 194
- 195
- 196
- 197
- 198
- 199
- 200
- 201
- 202
- 203
- 204
- 205
- 206
- 207
- 208
- 209
- 210
- 211
- 212
- 213
- 214
- 215
- 216
- 217
- 218
- 219
- 220
- 221
- 222
- 223
- 224
- 225
- 226
- 227
- 228
- 229
- 230
- 231
- 232
- 233
- 234
- 235
- 236
- 237
- 238
- 239
- 240
- 241
- 242
- 243
- 244
- 245
- 246
- 247
- 248
- 249
- 250
- 251
- 252
- 253
- 254
- 255
- 256
- 257
- 258
- 259
- 260
- 261
- 262
- 263
- 264
- 265
- 266
- 267
- 268
- 269
- 270
- 271
- 272
- 273
- 274
- 275
- 276
- 277
- 278
- 279
- 280
- 281
- 282
- 283
- 284
- 285
- 286
- 287
- 288
- 289
- 290
- 291
- 292
- 293
- 294
- 295
- 296
- 297
- 298
- 299
- 300
- 301
- 302
- 303
- 304
- 305
- 306
- 307
- 308
- 309
- 310
- 311
- 312
- 313
- 314
- 315
- 316
- 317
- 318
- 319
- 320
- 321
- 322
- 323
- 324
- 325
- 326
- 327
- 328
- 329
- 330
- 331
- 332
- 333
- 334
- 335
- 336
- 337
- 338
- 339
- 340
- 341
- 342
- 343
- 344
- 345
- 346
- 347
- 348
- 349
- 350
- 351
- 352
- 353
- 354
- 355
- 356
- 357
- 358
- 359
- 360
- 361
- 362
- 363
- 364
- 365
- 366
- 367
- 368
- 369
- 370
- 371
- 372
- 373
- 374
- 375
- 376
- 377
- 378
- 379
- 380
- 381
- 382
- 383
- 384
- 385
- 386
- 387
- 388
- 389
- 390
- 391
- 392
- 393
- 394
- 395
- 396
- 397
- 398
- 399
- 400
- 401
- 402
- 403
- 404
- 405
- 406
- 407
- 408
- 409
- 410
- 411
- 412
- 413
- 414
- 415
- 416
- 417
- 418
- 419
- 420
- 421
- 422
- 423
- 424
- 425
- 426
- 427
- 428
- 429
- 430
- 431
- 432
- 433
- 434
- 435
- 436
- 437
- 438
- 439
- 440
- 441
- 442
- 443
- 444
- 445
- 446
- 447
- 448
- 449
- 450
- 451
- 452
- 453
- 454
- 455
- 456
- 457
- 458
- 459
- 460
- 461
- 462
- 463
- 464
- 465
- 466
- 467
- 468
- 469
- 470
- 471
- 472
- 473
- 474
- 475
- 476
- 477
- 478
- 479
- 480
- 481
- 482
- 483
- 484
- 485
- 486
- 487
- 488
- 489
- 490
- 491
- 492
- 493
- 494
- 495
- 496
- 497
- 498
- 499
- 500
- 501
- 502
- 503
- 504
- 505
- 506
- 507
- 508
- 509
- 510
- 511
- 512
- 513
- 514
- 515
- 516
- 517
- 518
- 519
- 520
- 521
- 522
- 523
- 524
- 525
- 526
- 527
- 528
- 529
- 530
- 531
- 532
- 533
- 534
- 535
- 536
- 537
- 538
- 539
- 540
- 541
- 542
- 543
- 544
- 545
- 546
- 547
- 548
- 549
- 550
- 551
- 552
- 553
- 554
- 555
- 556
- 557
- 558
- 559
- 560
- 561
- 562
- 563
- 564
- 565
- 566
- 567
- 568
- 569
- 570
- 571
- 572
- 573
- 574
- 575
- 576
- 577
- 578
- 579
- 580
- 581
- 582
- 583
- 584
- 585
- 586
- 587
- 588
- 589
- 590
- 591
- 592
- 593
- 594
- 595
- 596
- 597
- 598
- 599
- 600
- 601
- 602
- 603
- 604
- 605
- 606
- 607
- 608
- 609
- 610
- 611
- 612
- 613
- 614
- 615
- 616
- 617
- 618
- 619
- 620
- 621
- 622
- 623
- 624
- 625
- 626
- 627
- 628
- 629
- 630
- 631
- 632
- 633
- 634
- 635
- 636
- 637
- 638
- 639
- 640
- 641
- 642
- 643
- 644
- 645
- 646
- 647
- 648
- 649
- 650
- 651
- 652
- 653
- 654
- 655
- 656
- 657
- 658
- 659
- 660
- 661
- 662
- 663
- 664
- 665
- 666
- 667
- 668
- 669
- 670
- 671
- 672
- 673
- 674
- 675
- 676
- 677
- 678
- 679
- 680
- 681
- 682
- 683
- 684
- 685
- 686
- 687
- 688
- 689
- 690
- 691
- 692
- 693
- 694
- 695
- 696
- 697
- 698
- 699
- 700
- 701
- 702
- 703
- 704
- 705
- 706
- 707
- 708
- 709
- 710
- 711
- 712
- 713
- 714
- 715
- 716
- 717
- 718
- 719
- 720
- 721
- 722
- 723
- 724
- 725
- 726
- 727
- 728
- 729
- 730
- 731
- 732
- 733
- 734
- 735
- 736
- 737
- 738
- 739
- 740
- 741
- 742
- 743
- 744
- 745
- 746
- 747
- 748
- 749
- 750
- 751
- 752
- 753
- 754
- 755
- 756
- 757
- 758
- 759
- 760
- 761
- 762
- 763
- 764
- 765
- 766
- 767
- 768
- 769
- 770
- 771
- 772
- 773
- 774
- 775
- 776
- 777
- 778
- 779
- 780
- 781
- 782
- 783
- 784
- 785
- 786
- 787
- 788
- 789
- 790
- 791
- 792
- 793
- 794
- 795
- 796
- 797
- 798
- 799
- 800
- 801
- 802
- 803
- 804
- 805
- 806
- 807
- 808
- 809
- 810
- 811
- 812
- 813
- 814
- 815
- 816
- 817
- 818
- 819
- 820
- 821
- 822
- 823
- 824
- 825
- 826
- 827
- 828
- 829
- 830
- 831
- 832
- 833
- 834
- 835
- 836
- 837
- 838
- 839
- 840
- 841
- 842
- 843
- 844
- 845
- 846
- 847
- 848
- 849
- 850
- 851
- 852
- 853
- 854
- 855
- 856
- 857
- 858
- 859
- 860
- 861
- 862
- 863
- 864
- 865
- 866
- 867
- 868
- 869
- 870
- 871
- 872
- 873
- 874
- 875
- 876
- 877
- 878
- 879
- 880
- 881
- 882
- 883
- 884
- 885
- 886
- 887
- 888
- 889
- 890
- 891
- 892
- 893
- 894
- 895
- 896
- 897
- 898
- 899
- 900
- 901
- 902
- 903
- 904
- 905
- 906
- 907
- 908
- 909
- 910
- 911
- 912
- 913
- 914
- 915
- 916
- 917
- 918
- 919
- 920
- 921
- 922
- 923
- 924
- 925
- 926
- 927
- 928
- 929
- 930
- 931
- 932
- 933
- 934
- 935
- 936
- 937
- 938
- 939
- 940
- 941
- 942
- 943
- 944
- 945
- 946
- 947
- 948
- 949
- 950
- 951
- 952
- 953
- 954
- 955
- 956
- 957
- 958
- 959
- 960
- 961
- 962
- 963
- 964
- 965
- 966
- 967
- 968
- 969
- 970
- 971
- 972
- 973
- 974
- 975
- 976
- 977
- 978
- 979
- 980
- 981
- 982
- 983
- 984
- 985
- 986
- 987
- 988
- 989
- 990
- 991
- 992
- 993
- 994
- 995
- 996
- 997
- 998
- 999
- 1000
- 1001
- 1002
- 1003
- 1004
- 1005
- 1006
- 1007
- 1008
- 1009
- 1010
- 1011
- 1012
- 1013
- 1014
- 1015
- 1016
- 1017
- 1018
- 1019
- 1020
- 1021
- 1022
- 1023
- 1024
- 1025
- 1026
- 1027
- 1028
- 1029
- 1030
- 1031
- 1032
- 1033
- 1034
- 1035
- 1036
- 1037
- 1038
- 1039
- 1040
- 1041
- 1042
- 1043
- 1044
- 1045
- 1046
- 1047
- 1048
- 1049
- 1050
- 1051
- 1052
- 1053
- 1054
- 1055
- 1056
- 1057
- 1058
- 1059
- 1060
- 1061
- 1062
- 1063
- 1064
- 1065
- 1066
- 1067
- 1068
- 1069
- 1070
- 1071
- 1072
- 1073
- 1074
- 1075
- 1076
- 1077
- 1078
- 1079
- 1080
- 1081
- 1082
- 1083
- 1084
- 1085
- 1086
- 1087
- 1088
- 1089
- 1090
- 1091
- 1092
- 1093
- 1094
- 1095
- 1096
- 1097
- 1098
- 1099
- 1100
- 1101
- 1102
- 1103
- 1104
- 1105
- 1106
- 1107
- 1108
- 1109
- 1110
- 1111
- 1112
- 1113
- 1114
- 1115
- 1116
- 1117
- 1118
- 1119
- 1120
- 1121
- 1122
- 1123
- 1124
- 1125
- 1126
- 1127
- 1128
- 1129
- 1130
- 1131
- 1132
- 1133
- 1134
- 1135
- 1136
- 1137
- 1138
- 1139
- 1140
- 1141
- 1142
- 1143
- 1144
- 1145
- 1146
- 1147
- 1148
- 1149
- 1150
- 1151
- 1152
- 1153
- 1154
- 1155
- 1156
- 1157
- 1158
- 1159
- 1160
- 1161
- 1162
- 1163
- 1164
- 1165
- 1166
- 1167
- 1168
- 1169
- 1170
- 1171
- 1172
- 1173
- 1174
- 1175
- 1176
- 1177
- 1178
- 1179
- 1180
- 1181
- 1182
- 1183
- 1184
- 1185
- 1186
- 1187
- 1188
- 1189
- 1190
- 1191
- 1192
- 1193
- 1194
- 1195
- 1196
- 1197
- 1198
- 1199
- 1200
- 1201
- 1202
- 1203
- 1204
- 1205
- 1206
- 1207
- 1208
- 1209
- 1210
- 1211
- 1212
- 1213
- 1214
- 1215
- 1216
- 1217
- 1218
- 1219
- 1220
- 1221
- 1222
- 1223
- 1224
- 1225
- 1226
- 1227
- 1228
- 1229
- 1230
- 1231
- 1232
- 1233
- 1234
- 1235
- 1236
- 1237
- 1238
- 1239
- 1240
- 1241
- 1242
- 1243
- 1244
- 1245
- 1246
- 1247
- 1248
- 1249
- 1250
- 1251
- 1252
- 1253
- 1254
- 1255
- 1256
- 1257
- 1258
- 1259
- 1260
- 1261
- 1262
- 1263
- 1264
- 1265
- 1266
- 1267
- 1268
- 1269
- 1270
- 1271
- 1272
- 1273
- 1274
- 1275
- 1276
- 1277
- 1278
- 1279
- 1280
- 1281
- 1282
- 1283
- 1284
- 1285
- 1286
- 1287
- 1288
- 1289
- 1290
- 1291
- 1292
- 1293
- 1294
- 1295
- 1296
- 1297
- 1298
- 1299
- 1300
- 1301
- 1302
- 1303
- 1304
- 1305
- 1306
- 1307
- 1308
- 1309
- 1310
- 1311
- 1312
- 1313
- 1314
- 1315
- 1316
- 1317
- 1318
- 1319
- 1320
- 1321
- 1322
- 1323
- 1324
- 1325
- 1326
- 1327
- 1328
- 1329
- 1330
- 1331
- 1332
- 1333
- 1334
- 1335
- 1336
- 1337
- 1338
- 1339
- 1340
- 1341
- 1342
- 1343
- 1344
- 1345
- 1346
- 1347
- 1348
- 1349
- 1350
- 1351
- 1352
- 1353
- 1354
- 1355
- 1356
- 1357
- 1358
- 1359
- 1360
- 1361
- 1362
- 1363
- 1364
- 1365
- 1366
- 1367
- 1368
- 1369
- 1370
- 1371
- 1372
- 1373
- 1374
- 1375
- 1376
- 1377
- 1378
- 1379
- 1380
- 1381
- 1382
- 1383
- 1384
- 1385
- 1386
- 1387
- 1388
- 1389
- 1390
- 1391
- 1392
- 1393
- 1394
- 1395
- 1396
- 1397
- 1398
- 1399
- 1400
- 1401
- 1402
- 1403
- 1404
- 1405
- 1406
- 1407
- 1408
- 1409
- 1410
- 1411
- 1412
- 1413
- 1414
- 1415
- 1416
- 1417
- 1418
- 1419
- 1420
- 1421
- 1422
- 1423
- 1424
- 1425
- 1426
- 1427
- 1428
- 1429
- 1430
- 1431
- 1432
- 1433
- 1434
- 1435
- 1436
- 1437
- 1438
- 1439
- 1440
- 1441
- 1442
- 1443
- 1444
- 1445
- 1446
- 1447
- 1448
- 1449
- 1450
- 1451
- 1452
- 1453
- 1454
- 1455
- 1456
- 1457
- 1458
- 1459
- 1460
- 1461
- 1462
- 1463
- 1464
- 1465
- 1466
- 1467
- 1468
- 1469
- 1470
- 1471
- 1472
- 1473
- 1474
- 1475
- 1476
- 1477
- 1478
- 1479
- 1480
- 1481
- 1482
- 1483
- 1484
- 1485
- 1486
- 1487
- 1488
- 1489
- 1490
- 1491
- 1492
- 1493
- 1494
- 1495
- 1496
- 1497
- 1498
- 1499
- 1500
- 1501
- 1502
- 1503
- 1504
- 1505
- 1506
- 1507
- 1508
- 1509
- 1510
- 1511
- 1512
- 1513
- 1514
- 1515
- 1516
- 1517
- 1518
- 1519
- 1520
- 1521
- 1522
- 1523
- 1524
- 1525
- 1526
- 1527
- 1528
- 1529
- 1530
- 1531
- 1532
- 1533
- 1534
- 1535
- 1536
- 1537
- 1538
- 1539
- 1540
- 1541
- 1542
- 1543
- 1544
- 1545
- 1546
- 1547
- 1548
- 1549
- 1550
- 1551
- 1552
- 1553
- 1554
- 1555
- 1556
- 1557
- 1558
- 1559
- 1560
- 1561
- 1562
- 1563
- 1564
- 1565
- 1566
- 1567
- 1568
- 1569
- 1570
- 1571
- 1572
- 1573
- 1574
- 1575
- 1576
- 1577
- 1578
- 1579
- 1580
- 1581
- 1582
- 1583
- 1584
- 1585
- 1586
- 1587
- 1588
- 1589
- 1590
- 1591
- 1592
- 1593
- 1594
- 1595
- 1596
- 1597
- 1598
- 1599
- 1600
- 1601
- 1602
- 1603
- 1604
- 1605
- 1606
- 1607
- 1608
- 1609
- 1610
- 1611
- 1612
- 1613
- 1614
- 1615
- 1616
- 1617
- 1618
- 1619
- 1620
- 1621
- 1622
- 1623
- 1624
- 1625
- 1626
- 1627
- 1628
- 1629
- 1630
- 1631
- 1632
- 1633
- 1634
- 1635
- 1636
- 1637
- 1638
- 1639
- 1640
- 1641
- 1642
- 1643
- 1644
- 1645
- 1646
- 1647
- 1648
- 1649
- 1650
- 1651
- 1652
- 1653
- 1654
- 1655
- 1656
- 1657
- 1658
- 1659
- 1660
- 1661
- 1662
- 1663
- 1664
- 1665
- 1666
- 1667
- 1668
- 1669
- 1670
- 1671
- 1672
- 1673
- 1674
- 1675
- 1676
- 1677
- 1678
- 1679
- 1680
- 1681
- 1682
- 1683
- 1684
- 1685
- 1686
- 1687
- 1688
- 1689
- 1690
- 1691
- 1692
- 1693
- 1694
- 1695
- 1696
- 1697
- 1698
- 1699
- 1700
- 1701
- 1702
- 1703
- 1704
- 1705
- 1706
- 1707
- 1708
- 1709
- 1710
- 1711
- 1712
- 1713
- 1714
- 1715
- 1716
- 1717
- 1718
- 1719
- 1720
- 1721
- 1722
- 1723
- 1724
- 1725
- 1726
- 1727
- 1728
- 1729
- 1730
- 1731
- 1732
- 1733
- 1734
- 1735
- 1736
- 1737
- 1738
- 1739
- 1740
- 1741
- 1742
- 1743
- 1744
- 1745
- 1746
- 1747
- 1748
- 1749
- 1750
- 1751
- 1752
- 1753
- 1754
- 1755
- 1756
- 1757
- 1758
- 1759
- 1760
- 1761
- 1762
- 1763
- 1764
- 1765
- 1766
- 1767
- 1768
- 1769
- 1770
- 1771
- 1772
- 1773
- 1774
- 1775
- 1776
- 1777
- 1778
- 1779
- 1780
- 1781
- 1782
- 1783
- 1784
- 1785
- 1786
- 1787
- 1788
- 1789
- 1790
- 1791
- 1792
- 1793
- 1794
- 1795
- 1796
- 1
```

8.3.34 Move sikertelen - a célszoba tele van

- **Leírás**

Egy diák van a pálya egyik szobájában. Kiválaszt a szoba ajtajai közül egyet min át akar menni az annak túloldalán lévő szobába, amelyben a kapacitásával megegyező számú diák van. Az a szoba tele van így nem tud oda átmenni, eredménytelen a helyváltztatás.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tele szobába nem lehet átmenni.

- **Bemenet**

```
taddrroom f 15 1
taddrroom f 1 2
taddrdoor 1 2 f b 3
taddentity s Jancsi 4 1
taddentity s Jancsi2 5 2
start 20 t
move 3
debug 0
```

- **Elvárt kimenet**

| | | |
|---------------------|---------------------|---------------------|
| ---- Map ---- | inventory : | entitiesInside : |
| rooms : | ---- Student 4 ---- | ---- Student 5 |
| ---- Room 1 ---- | doors : | ---- |
| isToxic : false | ---- Door 3 ---- | name : Jancsi2 |
| isMerged : false | isCursed : false | remainingStun : 0 |
| isSticky : false | direction : BOTH | currentRoom : 1 |
| capacity : 15 | AvailableRooms : | isAlive : true |
| hasBeenCleaned : | - 1 | isInToxic : false |
| false | - 2 | roundIsOver : false |
| entitiesSinceCleanu | ---- Door 3 ---- | inventory : |
| p : 0 | ---- Room 1 ---- | ---- Student 5 ---- |
| containedItems : | ---- Room 2 ---- | doors : |
| entitiesInside : | isToxic : false | ---- Door 3 ---- |
| ---- Student 4 | isMerged : false | isCursed : false |
| ---- | isSticky : false | direction : BOTH |
| name : Jancsi | capacity : 1 | AvailableRooms : |
| remainingStun : 0 | hasBeenCleaned : | - 1 |
| currentRoom : 1 | false | - 2 |
| isAlive : true | entitiesSinceCleanu | ---- Door 3 ---- |
| isInToxic : false | p : 0 | ---- Room 2 ---- |
| roundIsOver : false | containedItems : | ---- Map ---- |

8.3.35 Move sikertelen - a kiválasztott ajtó elátkozot

- **Leírás**

Egyetlen diák van a pálya egyik szobájában. Kiválaszt a szoba ajtajai közül egyet amin át akar menni az annak túloldalán lévő szobába. Az ajtó elátkozott így azon keresztül nem lehet továbbmenni, eredménytelen a helyváltoztatás.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Elátkozott ajtón nem lehet továbbhaladni.

- **Bemenet**

```
taddrroom f 15 1
taddrroom f 1 2
tadddoor 1 2 t b 3
taddentity s Jancsi 4 1
start 20 t
move 3
debug 0
```

- **Elvárt kimenet**

| | | |
|---------------------|---------------------|---------------------|
| ---- Map ---- | isAlive : true | capacity : 1 |
| rooms : | isInToxic : false | hasBeenCleaned : |
| ---- Room 1 ---- | roundIsOver : false | false |
| isToxic : false | inventory : | entitiesSinceCleanu |
| isMerged : false | ---- Student 4 ---- | p : 0 |
| isSticky : false | doors : | containedItems : |
| capacity : 15 | ---- Door 3 ---- | entitiesInside : |
| hasBeenCleaned : | isCursed : true | doors : |
| false | direction : BOTH | ---- Door 3 ---- |
| entitiesSinceCleanu | AvailableRooms : | isCursed : true |
| p : 0 | - 1 | direction : BOTH |
| containedItems : | - 2 | AvailableRooms : |
| entitiesInside : | ---- Door 3 ---- | - 1 |
| ---- Student 4 | ---- Room 1 ---- | - 2 |
| ---- | ---- Room 2 ---- | ---- Door 3 ---- |
| name : Jancsi | isToxic : false | ---- Room 2 ---- |
| remainingStun : 0 | isMerged : false | ---- Map ---- |
| currentRoom : 1 | isSticky : false | |

8.3.36 Move sikertelen - az ajtó egyirányú, ebből az irányból nem átjárható

- **Leírás**

Egyetlen diák van a pálya egyik szobájában. Kiválaszt a szoba ajtajai közül egyet amin át akar menni az annak túloldalán lévő szobába. A kiválasztott ajtó egyirányú, úgy, hogy a jelenlegi szobából nem biztosít átjárást. Ennek következtében a helyváltoztatás sikertelen.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Egyirányú ajtó csak a helyes irányban járható át.

- **Bemenet**

```
taddrroom f 15 1
taddrroom f 1 2
taddrdoor 1 2 t 1 3
taddentity s Jancsi 4 1
start 20 t
move 3
debug 0
```

- **Elvárt kimenet**

| | | |
|---------------------|---------------------|---------------------|
| ---- Map ---- | isAlive : true | capacity : 1 |
| rooms : | isInToxic : false | hasBeenCleaned : |
| ---- Room 1 ---- | roundIsOver : false | false |
| isToxic : false | inventory : | entitiesSinceCleanu |
| isMerged : false | ---- Student 4 ---- | p : 0 |
| isSticky : false | doors : | containedItems : |
| capacity : 15 | ---- Door 3 ---- | entitiesInside : |
| hasBeenCleaned : | isCursed : false | doors : |
| false | direction : LEFT | ---- Door 3 ---- |
| entitiesSinceCleanu | AvailableRooms : | isCursed : false |
| p : 0 | - 1 | direction : LEFT |
| containedItems : | - 2 | AvailableRooms : |
| entitiesInside : | ---- Door 3 ---- | - 1 |
| ---- Student 4 | ---- Room 1 ---- | - 2 |
| ---- | ---- Room 2 ---- | ---- Door 3 ---- |
| name : Jancsi | isToxic : false | ---- Room 2 ---- |
| remainingStun : 0 | isMerged : false | ---- Map ---- |
| currentRoom : 1 | isSticky : false | |

8.3.37 Move sikertelen - kiválasztott ajtó nem része a szobának

- **Leírás**

Egyetlen diák van a pálya egyik szobájában. Kiválaszt egy ajtót amin át akar menni az annak túloldalán lévő szobába. A kiválasztott ajtó azonban nem része a szobának, a helyváltoztatás sikertelen.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Nem a szobában lévő ajtón nem lehet áthaladni

- **Bemenet**

```
taddrroom f 15 1
taddrroom f 1 2
taddrroom f 5 3
taddrdoor 1 3 t b 4
taddrdoor 1 2 f b 5
taddrdoor 2 3 f b 6
taddentity s Jancsi 7 1
start 20 t
move 6
debug 0
```

- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
---- Student 7
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 7 ----
doors :
---- Door 4 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 3
---- Door 4 ----
---- Door 5 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 2
---- Door 5 ----
---- Room 1 ----
---- Room 2 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 1
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
doors :
---- Door 5 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 2
---- Door 5 ----
---- Door 6 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 2
- 3
---- Door 6 ----
---- Room 2 ----
---- Room 3 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 5
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
doors :
---- Door 6 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 2
- 3
---- Door 6 ----
---- Door 4 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 3
---- Door 4 ----
---- Room 3 ----
---- Map ----

```

8.3.38 Move sikeres player szobában marad

- **Leírás**
Egyetlen diák van a pálya egyik szobájában. Kör végén maradni akar a szobában. A diák a szobában maradt
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizni, hogy a kör végén szobában maradás helyesen hajtódik végre.

- **Bemenet**
taddroom f 10 1
taddroom f 10 2
tadddoor 1 2 f b 3
taddentity s s1 15 1
start 5 t
move
debug 1
- **Elvárt kimenet**

```

---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned : false
entitiesSinceCleanup : 0
containedItems :
entitiesInside :
---- Student 15 ----
name : s1
remainingStun : 0
currentRoom : 2
isAlive : true

isInToxic : false
roundIsOver : false
inventory :
---- Student 15 ----
doors :
---- Door 3 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 2
---- Door 3 ----
---- Room 1 ----

```

8.3.39 Connect sikeres (kettő még nem connected tranzisztor)

- **Leírás**
Egy diáknál van kettő, még nem összekapcsolott tranzisztor. A diák összekapcsolja ezt a két tranzisztort.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Ellenőrizni, hogy a két tranzisztor összekapcsolódott.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem transistor 6 1
tadditem transistor 7 1
start t
pickup 6
pickup 7
connect 6 7
debug 0

- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :

---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 4 ----
---- Transistor 6
----
durability : 1

pair : 7
---- Transistor 6
----
durability : 1
pair : 6
---- Transistor 7
----
doors :
---- Room 1 ----
---- Map ---

```

8.3.40 Connect sikertelen (már kapcsolt)

- **Leírás**

Már összekapcsolt tranzisztorok sikertelen connectje.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Két, már összekapcsolt tranzisztort nem lehet újra összekapcsolni.

- **Bemenet**

```

taddrroom f 15 1
taddentity s Jancsi 4 1
tadditem transistor 6 1
tadditem transistor 7 1
start t
pickup 6
pickup 7
connect 6 7
connect 6 7
debug 0

```

- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :

---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Transistor 6
----
durability : 1
pair : No pair

---- Transistor 6
----
durability : 1
pair : No pair

---- Transistor 7
----
durability : 1
pair : No pair
---- Transistor 7
----
Student 4 ----
doors :
---- Room 1 ----
---- Map ----

```

8.3.41 Connect sikertelen (nem tranzisztor tárgy)

- **Leírás**
Csak tranzisztort lehet összekapcsolni.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Nem tranzisztor típusú tárgyat nem lehet összekapcsolni.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem beer 6 1
tadditem transistor 7 1
start t
pickup 6
pickup 7
connect 6
debug 0
- **Elvárt kimenet**

| | | |
|---------------------|---------------------|---------------------|
| ---- Map ---- | entitiesInside : | isActive : false |
| rooms : | ---- Student 4 | ---- Beer 6 ---- |
| ---- Room 1 ---- | ---- | ---- Transistor 7 |
| isToxic : false | name : Jancsi | ---- |
| isMerged : false | remainingStun : 0 | durability : 1 |
| isSticky : false | currentRoom : 1 | pair : No pair |
| capacity : 15 | isAlive : true | ---- Transistor 7 |
| hasBeenCleaned : | isInToxic : false | ---- |
| false | roundIsOver : false | ---- Student 4 ---- |
| entitiesSinceCleanu | inventory : | doors : |
| p : 0 | ---- Beer 6 ---- | ---- Room 1 ---- |
| containedItems : | durability : 5 | ---- Map ---- |

8.3.42 Connect sikertelen (nem létező tárgy)

- **Leírás**
Csak létező tárgyat lehet összekapcsolni.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Nem létező tárgy nem lesz összekapcsolva.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem beer 6 1
tadditem transistor 7 1
start t
pickup 6
pickup 7
connect 8 7
debug 0

- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Beer 6 ----
durability : 5
isActive : false
---- Beer 6 ----
---- Transistor 7
----
durability : 1
pair : No pair
---- Transistor 7
----
---- Student 4 ----
doors :
---- Room 1 ----
---- Map ----

```

8.3.43 Disconnect sikeres (kettő összekapcsolt tranzisztor)

- **Leírás**
Két, összekapcsolt tranzisztor szétkapcsolása.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Összekapcsolt tranzisztorok sikeres szétkapcsolása.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem transistor 6 1
tadditem transistor 7 1
start t
pickup 6
pickup 7
connect 6 7
disconnect 6 7
debug 0
- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 4 ----
---- Transistor 6
----
durability : 1
pair : no pair
---- Transistor 6
----
---- Transistor 7
----
durability : 1
pair : no pair
---- Transistor 7
----
pair : no pair
---- Transistor 7
----
doors :
---- Room 1 ----
---- Map ----

```

8.3.44 Disconnect sikertelen (kettő nem összekapcsolt tranzisztor)

- **Leírás**
Csak összekapcsolt tranzisztorokat lehet szétkapcsolni.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Nem összekapcsolt tranzisztorok szétkapcsolása sikertelen.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem transistor 6 1
tadditem transistor 7 1
start t
pickup 6
pickup 7
disconnect 6 7
debug 0
- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :

---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 4 ----
---- Transistor 6
----
durability : 1

pair : no pair
---- Transistor 6
---- Transistor 7
----
durability : 1
pair : no pair
---- Transistor 7
----
doors :
---- Room 1 ----
---- Map ----
durability : 1

```

8.3.45 Disconnect sikertelen(nem tranzisztor tárgy)

- **Leírás**
Csak tranzisztorokat lehet szétkapcsolni.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Nem tranzisztor tárggyal szétkapcsolás sikertelen.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem beer 6 1
tadditem transistor 7 1
start t
pickup 6
pickup 7
disconnect 6 7
debug 0
- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Beer 6 ----
durability : 5
isActive : false
---- Beer 6 ----
---- Transistor 7
----
durability : 1
pair : No pair
---- Transistor 7
----
---- Student 4 ----
doors :
---- Room 1 ----
---- Map ----

```

8.3.46 Disconnect sikertelen (a OR b tárgy nem létezik)

- **Leírás**
Csak létező tárgyat lehet szétkapcsolni.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Nem létező tárggyal szétkapcsolás sikertelen.
- **Bemenet**
taddroom f 15 1
taddentity s Jancsi 4 1
tadditem beer 6 1
tadditem transistor 7 1
start t
pickup 6
pickup 7
disconnect 6 8
debug 0
- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Beer 6 ----
durability : 5
isActive : false
---- Beer 6 ----
---- Transistor 7
----
durability : 1
pair : No pair
---- Transistor 7
----
---- Student 4 ----
doors :
---- Room 1 ----
---- Map ----

```

8.3.47 Teleport sikeres

- **Leírás**
Sikeres teleport.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Sikeres teleport.
- **Bemenet**
taddroom f 15 1
taddroom f 1 2
taddentity s Jancsi 4 1
tadditem transistor 6 1
tadditem transistor 7 1
tadddoor 1 2 f b 30
start t
pickup 6
pickup 7
connect 6 7
use 6
move 30
teleport 7
debug 0
- **Elvárt kimenet**

| | | |
|---------------------|---------------------|---------------------|
| ---- Map ---- | ---- Student 4 | isToxic : false |
| rooms : | ---- | isMerged : false |
| ---- Room 1 ---- | name : Jancsi | isSticky : false |
| isToxic : false | remainingStun : 0 | capacity : 1 |
| isMerged : false | currentRoom : 1 | hasBeenCleaned : |
| isSticky : false | isAlive : true | false |
| capacity : 15 | isInToxic : false | entitiesSinceCleanu |
| hasBeenCleaned : | roundIsOver : false | p : 0 |
| false | inventory : | containedItems : |
| entitiesSinceCleanu | ---- Student 4 ---- | entitiesInside : |
| p : 0 | doors : | doors : |
| containedItems : | ---- Door 30 ---- | ---- Door 30 ---- |
| ---- Transistor 6 | isCursed : false | isCursed : false |
| ---- | direction : BOTH | direction : BOTH |
| durability : 1 | AvailableRooms : | AvailableRooms : |
| pair : No pair | - 1 | - 1 |
| ---- Transistor 6 | - 2 | - 2 |
| ---- | ---- Door 30 ---- | ---- Door 30 ---- |
| entitiesInside : | ---- Room 1 ---- | ---- Room 2 ---- |
| | ---- Room 2 ---- | ---- Map ---- |

8.3.48 Teleport sikertelen (Rossz tárgyra hívódik)

- **Leírás**
Csak tranzisztorral lehet teleportálni
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Nem tranzisztor tárggyal sikertelen a teleport.
- **Bemenet**

```
taddroom f 15 1
taddroom f 1 2
taddentity s Jancsi 4 1
taddentity s Bela 5 2
tadditem transistor 6 1
tadditem transistor 7 1
tadddoor 1 2 f b 30
start t
```

```
pickup 6
pickup 7
connect 6 7
use 6
move 30
teleport 9
debug 0
```

- **Elvárt kimenet**

```
---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
---- Transistor 6
----
durability : 1
pair : No pair
---- Transistor 6
----
---- Transistor 7
----
durability : 1
pair : No pair
---- Transistor 7
----
entitiesInside :
---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 4 ----
doors :
---- Door 30 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 2
---- Door 30 ----
---- Room 1 ----
---- Room 2 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 1
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
---- Student 5
----
name : Bela
remainingStun : 0
currentRoom : 2
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 5 ----
doors :
---- Door 30 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 2
---- Door 30 ----
---- Room 2 ----
---- Map ----
```


8.3.49 Teleport sikertelen (Távoli szoba tele van)

- **Leírás**
Csak nem teli szobába lehet teleportálni.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Teli szobába sikertelen a teleport.
- **Bemenet**

```

taddroom f 15 1
taddroom f 1 2
taddentity s Jancsi 4 1
taddentity s Bela 5 2
tadditem transistor 6 1
tadditem transistor 7 1
tadddoor 1 2 f b 30
start t

```

```

pickup 6
pickup 7
connect 6 7
use 6
move 30
teleport 7
debug 0

```

- **Elvárt kimenet**

```

---- Map ----
rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 15
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
---- Transistor 6
----
durability : 1
pair : 7
---- Transistor 6
----
entitiesInside :
---- Student 4
----
name : Jancsi
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false

roundIsOver : false
inventory :
---- Student 4 ----
doors :
---- Door 30 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 2
---- Door 30 ----
---- Room 1 ----
---- Room 2 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 1
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
---- Student 5
----
name : Bela

remainingStun : 0
currentRoom : 2
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 5 ----
doors :
---- Door 30 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 2
---- Door 30 ----
---- Room 2 ----
---- Transistor 7
----
durability : 1
pair : 6
---- Transistor 7
----
---- Map ---

```

8.3.50 Sikeres kettéválasztás

- **Leírás**
Létezik egy üres szoba, ezután a System megpróbálja azt kettéválasztani, két szobára. Sikeresen jár.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Szoba kettéválasztásának helyes végrehajtása, ideális feltételek esetén.
- **Bemenet**
taddroom f 10 1
tsplit 1
debug 2
- **Elvárt kimenet**
---- Room 2 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 10
hasBeenCleaned : false
entitiesSinceCleanup : 0
containedItems :
entitiesInside :
doors :
---- Room 2 ----

8.3.51 Sikertelen kettéválasztás

- **Leírás**
Létezik egy üres szoba, benne tartózkodik a diák. A System úgy dönt váljon ketté a szoba, azonban ez meghiúsul.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Szoba kettéválasztásának meghiúsulása, nem ideális feltétel miatt.
- **Bemenet**
taddroom f 10 1
taddentity s s1 20 1
tsplit 1
debug 0
- **Elvárt kimenet**

● ---- Map ----

rooms :
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 10
hasBeenCleaned : false
entitiesSinceCleanup : 0
containedItems :
entitiesInside :
doors :
---- Room 1 ----

8.3.52 Összeolvasztás sikeres

- **Leírás**
Létezik két darab üres szoba, köztük egy ajtó. A System úgy dönt összeolvasztja őket. A két szobából egy darab lesz, a köztük lévő ajtó megszűnik.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Szobák összeolvasztásának helyes lefolyása, ideális feltételek esetén.
- **Bemenet**
taddroom f 10 1
taddroom f 10 2
tmerge 1 2
debug 0
- **Elvárt kimenet**
---- Room 1 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 10
hasBeenCleaned : false
entitiesSinceCleanup : 0
containedItems :
entitiesInside :
doors :
---- Room 1 ----
---- Map ----

8.3.53 Összeolvasztás sikertelen

- **Leírás**
Létezik két darab üres szoba, az egyikben egy diák tartózkodik. A System úgy dönt összeolvasztja őket, azonban ez sikertelen lesz.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Szobák összeolvasztásának megghiúsulása, nem ideális feltétel miatt.
- **Bemenet**
taddroom f 10 1
taddroom f 10 2
taddentity s s1 20 1
tmerge 1 2
debug 2
- **Elvárt kimenet**
---- Room 2 ----

isToxic : false
isMerged : false
isSticky : false
capacity : 10
hasBeenCleaned : false
entitiesSinceCleanup : 0
containedItems :
entitiesInside :
doors :
---- Room 2 ----

8.3.54 Szoba elátkozódik

- **Leírás**
Létezik egy szoba, benne egy diák, a kör végén a szoba el átkozódik (azaz minden ajtaja elátkozódik). A következő körben a diák nem tud kimenni.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Elátkozódás tesztelése
- **Bemenet**
taddroom f 10 1
taddroom f 10 2
taddentity s s1 20 1
tadddoor 1 2 f b 3
tcurse 1
start
move 3
debug 2
- **Elvárt kimenet**

| | |
|--------------------------|------------------|
| ---- Room 2 ---- | doors : |
| isToxic : false | ---- Door 3 ---- |
| isMerged : false | isCursed : false |
| isSticky : false | direction : BOTH |
| capacity : 10 | AvailableRooms : |
| hasBeenCleaned : false | - 1 |
| entitiesSinceCleanup : 0 | - 2 |
| containedItems : | ---- Door 3 ---- |
| entitiesInside : | ---- Room 2 ---- |

8.3.55 Szoba visszaváltozik a elátkozásból

- **Leírás**
Létezik egy szoba amelynek összes ajtaja el van átkozva benne egy diákkal. A kör végén a szoba “visszaváltozik. A diák el tudja hagyni a szobát a következő körben.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Visszaváltozás tesztelése
- **Bemenet**
taddroom f 10 1
taddroom f 10 2
taddentity s s1 20 1
tadddoor 1 2 t b 3
tuncurse 1
start
move 3
debug 2

- **Elvárt kimenet**

| | | |
|------------------|---------------|---------------|
| ---- Room 2 ---- | ---- Student | doors : |
| isToxic : | 20 ---- | ---- Door 3 |
| false | name : s1 | ---- |
| isMerged : | remainingStun | isCursed : |
| false | : 0 | false |
| isSticky : | currentRoom : | direction : |
| false | 1 | BOTH |
| capacity : 10 | isAlive : | AvailableRoom |
| hasBeenCleaned : | true | s : |
| false | isInToxic : | - 1 |
| entitiesSince | false | - 2 |
| Cleanup : 0 | roundIsOver : | ---- Door 3 |
| containedItems : | false | ---- |
| entitiesInside : | inventory : | ---- Room 2 |
| | ---- Student | ---- |
| | 20 ---- | |

8.3.56 Ragacsosság működésének tesztelése

- **Leírás**

Létezik egy 6 férőhelyes szoba, belép egy Janitor majd kitisztítja, ezután belép 5 diák. A szoba ragacsos lesz így az 5. már nem tud felvenni tárgyat.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ragacsosság működésének tesztelése.

- **Bemenet**

| | | |
|-------------------|----------------------|---------|
| taddrum f 10 1 | tadentity j j1 25 | move |
| taddrum f 6 2 | 3 | move |
| taddrum f 10 3 | taddrdoor 1 2 f b 10 | move |
| tadentity s s1 20 | taddrdoor 3 2 f b 11 | move 11 |
| 1 | start | move 10 |
| tadentity s s2 21 | move | move 10 |
| 1 | move | move 10 |
| tadentity s s3 22 | move | move 10 |
| 1 | move | move 10 |
| tadentity s s4 23 | move | debug 2 |
| 1 | move 10 | |
| tadentity s s5 24 | move | |
| 1 | move | |

- **Elvárt kimenet**

```

---- Room 2 ----
  isToxic : false
  isMerged : false
  isSticky : true
  capacity : 6
  hasBeenCleaned :
  false
  entitiesSinceCle
  anup : 0
  containedItems :
  entitiesInside :
    ---- Student 20
    ----
    name : s1
    remainingStun :
    0
    currentRoom : 2
    isAlive : true
    isInToxic :
    false
    roundIsOver :
    false
    inventory :
    ---- Student 20
    ----
    ---- Student 21
    ----
    name : s2
    remainingStun :
    0
    currentRoom : 2
    isAlive : true
    isInToxic :
    false
    roundIsOver :
    false
    inventory :
    ---- Student 22
    ----
    ---- Student 23
    ----
    name : s3
    remainingStun :
    0
    currentRoom : 2
    isAlive : true
    isInToxic :
    false
    roundIsOver :
    false
    inventory :
    ---- Student 22
    ----
    ---- Student 23
    ----
    name : s4
    remainingStun :
    0
    currentRoom : 2
    isAlive : true
    isInToxic :
    false
    roundIsOver :
    false
    inventory :
    ---- Student 21
    ----
    ---- Student 22
    ----
    name : s5
    remainingStun :
    0
    currentRoom : 2
    isAlive : true
    isInToxic :
    false
    roundIsOver :
    false
    inventory :
    ---- Student 24
    ----
    doors :
    ---- Door 10
    ----
    isCursed : false
    direction : BOTH
    AvailableRooms :
    - 1
    - 2
    ---- Door 10
    ----
    ---- Room 2 ----
  
```

8.3.57 Takarító kitisztítja a szobát

- **Leírás**
Létezik egy gázos szoba, belép egy takarító és kitisztítja
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Tisztítás sikerességének vizsgálata.
- **Bemenet**
taddrroom t 10 1
taddrroom f 10 2
taddrdoor 1 2 t b 10
taddentity j j1 25 2
start
move 10
debug 1

- **Elvárt kimenet**

```

---- Room 1
----
isToxic :
false
isMerged :
false
isSticky :
false
capacity : 10
hasBeenCleaned : false
entitiesSinceCleanup : 0
containedItems :

entitiesInside :
---- Janitor
25 ----
doors :
---- Door 10
----
isCursed :
false
direction :
BOTH
AvailableRooms :
- 1
- 2
---- Room 1
----
Map ----

```

8.3.58 Takarító kiküld a szobából egy diákot

- **Leírás**

Létezik egy szoba, benne egy diákkal. Belép a takarító és kiküldi a szobából. A takarító sikerrel jár.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szobából kiküldés tesztelése, ideális feltételekkel.

- **Bemenet**

```

taddrroom f 10 1
taddrroom f 10 2
tadddoor 1 2 f b 10
taddentity s s1 20 1
taddentity j j1 25 2
start
move
move 10
debug 2

```

- **Elvárt kimenet**

```

---- Room 2 ----
isToxic : false
isMerged : false
isSticky : false
capacity : 10
hasBeenCleaned :
false
entitiesSinceCleanup : 0
containedItems :
entitiesInside :

---- Student 20
----
name : s1
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : false
inventory :
---- Student 20
----

doors :
---- Door 10 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 2
---- Door 3 ----
---- Room 2 ---

```

8.3.59 Diák találkozik egy tanárral, megmenekül (TVSZ)

- **Leírás**
Egyetlen diák tartózkodik a pálya egyik szobájában, melynek egyetlen szomszédja van, amely nincsen tele. Ebben az egyetlen szomszédos szobában tartózkodik egy tanár. A tanár nincs elkábult állapotban. A diák hátizsákjában van egy nem nulla tartósságú TVSZ tárgy. A diák a köre végén úgy dönt, hogy elhagyja jelenlegi szobáját, és belép a szobába, melyben a tanár tartózkodik. Találkozásukkor a tanár megpróbálja kiszívni a diák lelkét, viszont mivel a diák hátizsákjában van egy nem nulla tartósságú TVSZ tárgy, így az megmenti a lélekvesztéstől. A diák megmenekül, és a hátizsákjában lévő TVSZ tárgy veszít egyet a tartósságából.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Tanár-diák találkozás tesztelése úgy, hogy a diák megmenekül a nála lévő TVSZ által
- **Bemenet**
taddroom f 10 1
taddroom f 10 2
tadddoor 1 2 f b 3
tadditem tvsz 4 1
taddentity s Jani 5 1
taddentity t Bela 6 2
start 5 t
pickup 4
move 3
move
debug 5 6
- **Elvárt kimenet**

| | | |
|---------------------|---------------------|---------------------|
| ---- Student 5 ---- | inventory : | remainingStun : 0 |
| name : Jani | ---- TVSZ 4 ---- | currentRoom : 2 |
| remainingStun : 0 | durability : 2 | isAlive : true |
| currentRoom : 2 | ---- TVSZ 4 ---- | isInToxic : false |
| isAlive : true | ---- Student 5 ---- | roundIsOver : true |
| isInToxic : false | ---- Teacher 6 ---- | inventory : |
| roundIsOver : false | name : Bela | ---- Teacher 6 ---- |

8.3.60 Diák találkozik egy tanárral, megmenekül (sör)

- **Leírás**
Egyetlen diák tartózkodik a pálya egyik szobájában, melynek egyetlen szomszédja van, amely nincsen tele. Ebben az egyetlen szomszédos szobában tartózkodik egy tanár. A tanár nincs elkábult állapotban. A diák hátizsákjában van egy nem nulla tartósságú, aktív Sör tárgy. A diák a köre végén úgy dönt, hogy elhagyja jelenlegi szobáját, és belép a szobába, melyben a tanár tartózkodik. Találkozásukkor a tanár megpróbálja kiszívni a diák lelkét, viszont mivel a diák hátizsákjában van egy nem nulla tartósságú, aktív Sör tárgy, így az megmenti a lélekvesztéstől. A diák megmenekül, és a hátizsákjában lévő Sör tárgy eltűnik.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Tanár-diák találkozás tesztelése úgy, hogy a diák megmenekül a nála lévő sör által

- **Bemenet**
taddroom f 10 1
taddroom f 10 2
tadddoor 1 2 f b 3
tadditem beer 4 1
taddentity s Jani 5 1
taddentity t Bela 6 2
start 5 t
pickup 4
use 4
move 3
move
debug 5 6
- **Elvárt kimenet**

| | | |
|---------------------|---------------------|---------------------|
| ---- Student 5 ---- | roundIsOver : false | currentRoom : 2 |
| name : Jani | inventory : | isAlive : true |
| remainingStun : 0 | ---- Student 5 ---- | isInToxic : false |
| currentRoom : 2 | ---- Teacher 6 ---- | roundIsOver : true |
| isAlive : true | name : Bela | inventory : |
| isInToxic : false | remainingStun : 0 | ---- Teacher 6 ---- |

8.3.61 Diák találkozik egy tanárral, nem menekül meg

- **Leírás**
Egyetlen diák tartózkodik a pálya egyik szobájában, melynek egyetlen szomszédja van, amely nincsen tele. Ebben az egyetlen szomszédos szobában tartózkodik egy tanár. A tanár nincs elkábult állapotban. A diák hátizsákja üres. A diák a köre végén úgy dönt, hogy elhagyja jelenlegi szobáját, és belép a szobába, melyben a tanár tartózkodik. Találkozásukkor a tanár megpróbálja kiszívni a diák lelkét, és mivel a diák hátizsákja üres, így nincs mi megmentse a lélekvesztéstől. A diák nem menekül meg, számára a játék véget ért.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Tanár-diák találkozás, úgy hogy a diák meghal
- **Bemenet**
taddroom f 10 1
taddroom f 10 2
tadddoor 1 2 f b 3
taddentity s Jani 5 1
taddentity t Bela 6 2
start 5 t
move 3
move
debug 5 6

- **Elvárt kimenet**

```

---- Student 5 ----      roundIsOver : false      currentRoom : 2
name : Jani              inventory :      isAlive : true
remainingStun : 0        ---- Student 5 ----      isInToxic : false
currentRoom : 2          ---- Teacher 6 ----      roundIsOver : true
isAlive : true           name : Bela          inventory :
isInToxic : false        remainingStun : 0      ---- Teacher 6 ----

```

8.3.62 Tanár megbénul egy rongy által

- **Leírás**

Egyetlen tanár tartózkodik a pálya egyik szobájában, melynek egyetlen szomszédja van, amely nincsen tele. Ebben az egyetlen szomszédos szobában van egy nem nulla tartósságú, aktív Rongy tárgy. A tanár nincs elkábult állapotban. A tanár a köre végén úgy dönt, hogy elhagyja jelenlegi szobáját, és belép a szobába, melyben a rongy található. Belépéskor a tanár elkábult állapotba kerül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Rongy tárgy képességének ellenőrzése

- **Bemenet**

```

taddroom f 10 1          start 5 t
taddroom f 10 2          pickup 6
taddroom f 10 3          drop 6
tadddoor 1 2 f b 4       move 5
tadddoor 1 3 f b 5       move 4
tadditem rag 6 1         move
taddentity s Jani 7 1    move
taddentity t Bela 8 2    debug 1

```

- **Elvárt kimenet**

```

---- Room 1 ----      entitiesInside :      direction : BOTH
isToxic : false        ---- Teacher 8      AvailableRooms :
isMerged : false      ----              - 1
isSticky : false       name : Bela          - 2
capacity : 10          remainingStun : 1    ---- Door 4 ----
hasBeenCleaned :      currentRoom : 1    ---- Door 5 ----
false                 isAlive : true      isCursed : false
entitiesSinceCleanu   isInToxic : false    direction : BOTH
p : 0                 roundIsOver : false  AvailableRooms :
containedItems :      inventory :          - 1
---- Rag 6 ----      ---- Teacher 8 ----      - 3
durability : 5        doors :              ---- Door 5 ----
isActive : true       ---- Door 4 ----      ---- Room 1 ----
---- Rag 6 ----      isCursed : false

```

8.3.63 Tanár megbénul gáz által

- **Leírás**
Egyetlen tanár tartózkodik a pálya egyik szobájában, melynek egyetlen szomszédja van, amely nincsen tele. Ez az egyetlen szomszédos szoba gázos állapotban van. A tanár nincs elkábult állapotban, és a hátizsákja üres. A tanár a köre végén úgy dönt, hogy elhagyja jelenlegi szobáját, és belép a szomszédos szobába. Belépéskor, mivel a hátizsákja üres, azaz nincs nála Maszk tárgy ami megmentené, elkábult állapotba kerül és eldobja a nála lévő összes tárgyat. Mivel hátizsákja üres, ezért nincs mit eldobnia.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Gázos szoba tanárra való hatása
- **Bemenet**
taddroom f 10 1
taddroom t 10 2
tadddoor 1 2 f b 3
taddentity t Bela 4 1
start 5 t
move 3
move
debug 2
- **Elvárt kimenet**

| | | |
|---------------------|--------------------|---------------------|
| ---- Room 2 ---- | entitiesInside : | ---- Teacher 4 ---- |
| isToxic : true | ---- Teacher 4 | doors : |
| isMerged : false | ---- | ---- Door 3 ---- |
| isSticky : false | name : Bela | isCursed : false |
| capacity : 10 | remainingStun : 1 | direction : BOTH |
| hasBeenCleaned : | currentRoom : 2 | AvailableRooms : |
| false | isAlive : true | - 1 |
| entitiesSinceCleanu | isInToxic : true | - 2 |
| p : 0 | roundIsOver : true | ---- Door 3 ---- |
| containedItems : | inventory : | ---- Room 2 ---- |

8.3.64 Gáz által bénult tanárt rongy is bénítja

- **Leírás**
Egyetlen tanár tartózkodik a pálya egyik szobájában mely gázos, és egyetlen szomszédja van, amely nincsen tele. Ebben az egyetlen szomszédos szobában van egy diák. A tanár elkábult állapotban van a gázos szoba miatt. A diák hátizsákjában van egy nem nulla tartósságú Maszk, és egy nem nulla tartósságú, aktív Rongy. A diák a köre végén úgy dönt, hogy elhagyja jelenlegi szobáját, és belép a szobába, melyben a tanár tartózkodik. Belépéskor a tanár, mivel kábult állapotban van, nem támadja meg a diákot, ezen felül a diákra a szoba gázosága sem hat, mivel van nála nem nulla tartósságú Maszk tárgy. Mikor újra a diák kerül sorra, körében úgy dönt, hogy eldobja a nem nulla tartósságú, aktív Rongy tárgyat a szobában. Ennek hatására a szobában lévő tanár elkábultsága növekszik.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Gázos szoba, és rongy tárgy hatása a tanárra

- **Bemenet**

| | | |
|---------------------|---------------------|---------|
| taddroom f 10 1 | taddentity t Bela 7 | move 3 |
| taddroom t 10 2 | 2 | move |
| tadddoor 1 2 f b 3 | start 5 t | drop 4 |
| tadditem rag 4 1 | pickup 4 | move 3 |
| tadditem mask 5 1 | pickup 5 | move |
| taddentity s Jani 6 | move | debug 2 |
| 1 | move | |

- **Elvárt kimenet**

| | | |
|---------------------|--------------------|---------------------|
| ---- Room 2 ---- | isActive : true | ---- Teacher 7 ---- |
| isToxic : true | ---- Rag 4 ---- | doors : |
| isMerged : false | entitiesInside : | ---- Door 3 ---- |
| isSticky : false | ---- Teacher 7 | isCursed : false |
| capacity : 10 | ---- | direction : BOTH |
| hasBeenCleaned : | name : Bela | AvailableRooms : |
| false | remainingStun : 2 | - 1 |
| entitiesSinceCleanu | currentRoom : 2 | - 2 |
| p : 0 | isAlive : true | ---- Door 3 ---- |
| containedItems : | isInToxic : true | ---- Room 2 ---- |
| ---- Rag 4 ---- | roundIsOver : true | |
| durability : 5 | inventory : | |

8.3.65 Diák megbénul gáz által

- **Leírás**

Egyetlen diák tartózkodik a pálya egyik szobájában, melynek egyetlen szomszédja van, amely nincsen tele. Ez az egyetlen szomszédos szoba gázos állapotban van. A diák nincs elkábult állapotban, és a hátizsákjában van egy TVSZ tárgy. A diák a köré végén úgy dönt, hogy elhagyja jelenlegi szobáját, és belép a szomszédos szobába. Belépéskor, mivel a hátizsákja nem üres, viszont nincs nála Maszk tárgy ami megmentené, elkábult állapotba kerül és eldobja a nála lévő összes tárgyat. A diák hátizsákjából kikerül a TVSZ tárgy, és bekerül a szoba tárgyai közé.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Gázos szoba hatása diákokra

- **Bemenet**

```
taddroom f 10 1
taddroom t 10 2
tadddoor 1 2 f b 3
tadditem tvsz 4 1
taddentity s Jani 5 1
start 5 t
pickup 4
move 3
debug 2
```

- **Elvárt kimenet**

```

---- Room 2 ----
isToxic : true
isMerged : false
isSticky : false
capacity : 10
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
---- TVSZ 4 ----
durability : 3
---- TVSZ 4 ----
entitiesInside :
---- Student 5
----
name : Jani
remainingStun : 2
currentRoom : 2
isAlive : true
isInToxic : true
roundIsOver : false
inventory :
---- Student 5 ----
doors :
---- Door 3 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 2
---- Door 3 ----
---- Room 2 ----

```

8.3.66 Tanár nem bénul meg maszk miatt

- **Leírás**

Létezik egy gázzal teli szoba, belép egy tanár de nem bénul meg mivel van nála maszk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Gázos szoba hatása tanárra, ha van nála maszk

- **Bemenet**

```

taddroom t 10 1
taddroom t 10 2
tadddoor 1 2 f b 3
tadditem mask 4 1
taddentity t Bela 5 1
start 5 t
pickup 4
move 3
move
debug 2

```

- **Elvárt kimenet**

```

---- Room 2 ----
isToxic : true
isMerged : false
isSticky : false
capacity : 10
hasBeenCleaned :
false
entitiesSinceCleanu
p : 0
containedItems :
entitiesInside :
---- Teacher 5
----
name : Bela
remainingStun : 0
currentRoom : 2
isAlive : true
isInToxic : false
roundIsOver : true
inventory :
---- Mask 4 ----
durability : 2
lastRoomWasToxic :
true
---- Mask 4 ----
---- Teacher 5 ----
doors :
---- Door 3 ----
isCursed : false
direction : BOTH
AvailableRooms :
- 1
- 2
---- Door 3 ----
---- Room 2 ----

```

8.3.67 Diák nem bénul meg a maszk miatt

- **Leírás**
Létezik egy gázzal teli szoba, belép egy diák de nem bénul meg mivel van nála maszk.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Maszk működés tesztje.

- **Bemenet**
taddrroom t 10 1
taddrroom t 10 2
tadddoor 1 2 f b 3
tadditem mask 4 1
taddentity s Bela 5 1
start 5 t
pickup 4
move 3
move
debug 2

- **Elvárt kimenet**

| | | |
|---------------------|--------------------|---------------------|
| ---- Room 2 ---- | name : Bela | ---- Student 5 ---- |
| isToxic : true | remainingStun : 0 | doors : |
| isMerged : false | currentRoom : 2 | ---- Door 3 ---- |
| isSticky : false | isAlive : true | isCursed : false |
| capacity : 10 | isInToxic : false | direction : BOTH |
| hasBeenCleaned : | roundIsOver : true | AvailableRooms : |
| false | inventory : | - 1 |
| entitiesSinceCleanu | ---- Mask 4 ---- | - 2 |
| p : 0 | durability : 2 | ---- Door 3 ---- |
| containedItems : | lastRoomWasToxic : | ---- Room 2 ---- |
| entitiesInside : | true | |
| ---- Student 5 ---- | ---- Mask 4 ---- | |

8.3.68 Takarítóra nem hat a gázos szoba

- **Leírás**
Létezik egy gázzal teli szoba, belép egy takarító de nem bénul meg.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A gáz hatástalansága a takarítóra.

- **Bemenet**
taddrroom f 10 1
taddrroom t 10 2
tadddoor 1 2 f b 3
taddentity j Bela 5 1
start 5 t
move 3
debug 5

- **Elvárt kimenet**

| | |
|---------------------|---------------------|
| ---- Janitor 5 ---- | isInToxic : false |
| name : j1 | roundIsOver : false |
| remainingStun : 0 | inventory : |
| currentRoom : 1 | ---- Janitor 5 ---- |
| isAlive : true | |

8.3.69 Logarléc felvétele

- **Leírás**
Létezik egy szoba, benne egy diákkal és egy logarléc tárgyal. A diák felveszi logarlécet ezzel megnyerve a játékot.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Győzelmi feltétel tesztelése.
- **Bemenet**
taddroom f 10 1
taddentity s Bela 5 1
tadditem sliderule 10 1
start 5 t
pickup 10
debug 5
- **Elvárt kimenet**
---- Student 5 ----
name : Bela
remainingStun : 0
currentRoom : 1
isAlive : true
isInToxic : false
roundIsOver : true
inventory :
---- SlideRule 10 ----
durability : 2
---- SlideRule 10 ----
---- Student 5 ----

8.3.70 Hamis Maszk, gázos szobában nem véd

- **Leírás**
Létezik egy szoba, benne egy diák, a diák hátizsákjában egy hamis maszk. Tovább lép egy gázos szobába ahol a maszk nem fogja megvédeni így elájul.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Hamis maszk működési tesztje
- **Bemenet**
taddroom f 10 1
taddroom t 10 2
tadddoor 1 2 f b 3
taddentity s Bela 5 1
tadditem fakemask 10 1
start 5 t
pickup 10
move 3
debug 5

- **Elvárt kimenet**

```

---- Student 5 ----
name : Bela
remainingStun : 3
currentRoom : 2
isAlive : true
isInToxic : false
roundIsOver : true

```

```

inventory :
---- FakeMask 10 ----
durability : 2
lastRoomWasToxic : false
---- FakeMask 10 ----
---- Student 5 ----

```

8.4 A tesztelést támogató programok tervei

A tesztelést segítő program

boolean fileCompare(String filePath1, String filePath2),
 funkcióval valósul meg.

- A függvény vár 2 db fájlnevet, ezeknek először megnézi hosszát.
- Amennyiben a két fájl nem ugyanolyan hosszú, visszatér egy false értékkel és a kimeneti fájlban következő jelenik meg:
 Input files does not match in length aborted.
- Ha a 2 fájl ugyanolyan hosszú, akkor összehasonlítja a soraikat (1-1 sort). Ha talál különbséget akkor a sorszámmal együtt a két sort egymás fölé téve, ki írja őket a kimeneti fájlba. Ezután folytatja az tesztelést többi sorra. Valamint a végén visszatér egy false értékkel.

Ha a bemeneti fájlok:

| | |
|-------------|-------------|
| test1.txt : | test2.txt : |
| test1 | 1test1 |
| test2 | test2 |
| test3 | 3test3 |

Kimeneti fájl:

```

compareOut.txt:
--> line 1:
test1
1test1
--> line 3:
test3
3test3

```

- Ha teljesen megegyezik a két fájl akkor a kimeneti fájl üres és true a visszatérési érték.

8.5 Napló

| Kezdet | Időtartam | Résztevők | Leírás |
|-------------------|-----------|------------------------------------------------|----------------------------------------------------------------------------------|
| 2024.04.10 14:00 | 2,5 óra | Páter Bereczky Horváth Miczki Nagy | Értekezlet. Alapvető logikai részletek átbeszélése, cél megteremtése |
| 2024.04.11. 15:00 | 5 óra | Páter Bereczky Horváth Miczki Nagy | Értekezlet. GameController osztály felépítésének tervezése. |
| 2024.04.12. 12:00 | 5 óra | Páter Bereczky Horváth Miczki Nagy | Értekezlet. GameController osztály metódusainak kidolgozása |
| 2024.04.13. 15:00 | 6 óra | Páter Bereczky Horváth Miczki Nagy | Értekezlet. Osztályok dokumentálása, pszeudokód elkészítése. |
| 2024.04.14. 15:00 | 4 óra | Páter Bereczky Horváth Miczki Nagy | Értekezlet tesztek megírása, ellenőrzése átbeszélése (pszeudo) |
| 2024.04.15. 7:00 | 5 óra | Páter Bereczky Horváth Miczki Nagy | Értekezlet. Dokumentáció véglegesítése. |