

## 8. Részletes tervek

### 63 – Lehozook

Konzulens:

Szabó András

#### Csapattagok

Osztrogonác Dóra	UHI5AX	osztrogonaczdora@gmail.com
Kiss Tamás	C88IEG	kisstomi03@gmail.com
Halász Adrián	CZT1LC	virtualriot09@gmail.com
Schmidt Gergő	S2Q2D4	s.gergo.az@gmail.com
András		
Gáspár András	I6BST3	gaspar.andras.03@gmail.com

2024.04.15.

## 8. Részletes tervek

### 8.0 Módosítások a korábbi dokumentumokhoz képest

#### 7.1 Prototípus interface definíciója

**Megjegyzés:** a program/játék nyelve angol, a korábban meghatározott kimeneti nyelv tekintettel a dokumentum magyar nyelvére készült, viszont a tesztesetek részletezése során már a tesztek futtatásakor látható angol kifejezések fognak szerepelni.

#### Módosult parancs:

##### **AddCharacter <Character\_id><Type> <Room\_id>**

Bemenet: új paraméter hozzáadva a parancshoz

Leírás: Adott típusú karaktert belerak egy megadott id-jű szobába.

Opciók: Type lehet "Student" "Professor" "Cleaner", a Room\_id amibe kerül.

Kimenet: <Character\_id>-es karakter hozzáadva a <Room\_id>-es szobához.

##### **EnterRoom <Character\_id> <Room\_id>**

Kimenet:

Ha nincs tele a szoba:

<Character\_id> karakter áthelyezve a <Room\_id> szobába.

Ha tele van:

<Character\_id> karakter nem tudott belépni a <Room\_id> szobába.

##### **AddItem <Character\_id><name><Item\_id><real>**

Bemenet: új paraméter hozzáadva a parancshoz

Leírás: Megadott karakterhez hozzáad egy tárgyat

Opciók: A Character\_id a kiválasztott karakter és a name pedig a tárgy neve ami hozzá kerül, a real attribútum pedig megadja, hogy "true" valódi item lesz, vagy "false" hamis item.

Kimenet: A <name> <Item\_id> tárgy <real> verziója hozzáadva a <Character\_id> karakterhez.

##### **Init<file\_name>**

Bemenet: új paraméter hozzáadva a parancshoz

Leírás: A pálya betöltése

Opciók: -

Kimenet: A <file\_name> pálya betöltve.

##### **GooRoom <Room\_id><glue>**

Bemenet: új paraméter hozzáadva a parancshoz

Leírás: Aktiválja a szoba ragacsát.

Opciók: Ha a szoba már elkezdett ragacsosodni (goo nem null), akkor a glue paraméter szerint jár el, ellenkező esetben létrehozza a ragacsot a szobában.

Room\_id a kiválasztott szoba, a glued kapcsolóval beállíthatjuk, hogy a ragacs leragassza a tárgyakat a szobában, elkerülve azt, hogy meg kelljen várni, amíg belép annyi számú karakter a szobába. Az unglued pedig feloldja a ragacsot.

Kimenet:

- A szoba már ragacsos
  - A <Room\_id> szoba alaphelyzetbe állt, tárgyak leragasztva: <glue>
- A szoba még nem ragacsos
  - A <Room\_id> szoba ragacsa aktiválva, tárgyak leragasztva: <glue>

**Új parancs:****AddRoom <Room\_id><Type>**

Bemenet: pályaeépítéshez szükséges új parancs

Leírás: Szoba hozzáadása a pályához

Opciók: Típus lehet "Cursed", alapértelmezetten normál

Kimenet: Szoba <Room\_id> hozzáadva a pályához

**SetNeighbour<Room\_id1><Room\_id2>**

Bemenet: pályaeépítéshez szükséges új parancs

Leírás: Az első elemnek beállítja a másodikat szomszédjaként

(1-es szomszédos a 2-essel, visszafelé csak akkor igaz, ha a paramétereket felcserélve is kiadjuk a parancsot)

Opciók: -

Kimenet: <Room\_id2> szoba hozzáadva szomszédként a <Room\_id1> szobához.

**SpawnItem <Room\_id><name><Item\_id><real>**

Bemenet: pályaeépítéshez szükséges új parancs

Leírás: Megadott szobához hozzáad egy tárgyat

Opciók: Room\_id a kiválasztott szoba, a name pedig a tárgy neve és az Item\_id az azonosítója ami alapján bele kerül a terembe, a real attribútum pedig megadja, hogy "true" valódi item lesz, vagy "false" hamis item.

Kimenet: <Character\_id>-es karakter hozzáadva a <Room\_id>-es szobához.

**SetPair<Item\_id1><Item\_id2>**

Bemenet: tranzisztor párosításához kell

Leírás: A megadott két tranzisztort összepárosítja

Opciók: Az item\_Id1 az első az Item\_id2 a második tranzisztor.

Kimenet: <Item\_id1> tranzisztor párosítva <Item\_id2> tranzisztorral.

**ActivateTransistor<Item\_id>**

Bemenet: tranzisztor használatához kell

Leírás: Használja a teleportálás funkciót

Opciók: Az item\_Id egy párosított tranzisztor kézben tartott párja.

Kimenet: <Item\_id> tranzisztor aktiválva.

**SetRoomFull<Room\_id>**

Bemenet: Pályaeépítéshez szükséges

Leírás: Beállítja a szoba kapacitását maximumra

Opciók: -

Kimenet: A szoba <Room\_id> tele van

**Help**

Bemenet: felhasználót segítő új parancs

Leírás: Kiírja rövid magyarázattal a használható parancsok listáját

Opciók: -

Kimenet: parancsok nevei és attribútumai felsorolva

Init<file\_name> pályabetöltése

...

UseItem<Character\_id><Item\_id> a karakter használ egy tárgyat

...

**7.3 Tesztelési terv****teszteset átfogalmazása**

<b>Teszt-eset neve</b>	Professzor-hallgató interakció
<b>Rövid leírás</b>	A hallgató belép egy szobába. Az adott szobába már éppen tartózkodik egy nem kábult oktató. A következő kör combat lesz.
<b>Teszt célja</b>	A hallgató oktató közös szobában tartózkodásakor harchelyzet alakul ki.

**új teszteset hozzáadása:**

<b>Teszt-eset neve</b>	Professzor aktívan ragacsos szobába lép
<b>Rövid leírás</b>	A professzor belép egy szobába. A szoba éppen aktív ragacsos állapotban van, ami miatt a professzor nem tudja felemészteni a tárgyakat.
<b>Teszt célja</b>	A ragacs hatásának tesztelése a professzorokra nézve.

<b>Teszt-eset neve</b>	Kábult professzor-hallgató interakció
<b>Rövid leírás</b>	A hallgató belép egy szobába. Az adott szobába már éppen tartózkodik egy kábult oktató. Nem lesz combat kör, a hallgató folytatja a körét, itemeket vehet fel, használhat stb.
<b>Teszt célja</b>	A hallgató és kábult oktató közös szobában tartózkodásakor nem alakul ki harchelyzet.

<b>Teszt-eset neve</b>	Hamis tárgy felvétele
<b>Rövid leírás</b>	A hallgató egy hamis tárgyat vesz fel, ami azonnal kifejti a hatását, egy szomszédos szobába helyezi át.
<b>Teszt célja</b>	A hallgató átkerül egy másik szobába.

<b>Teszt-eset neve</b>	StBeerCups használata
<b>Rövid leírás</b>	A hallgató és oktató harchelyzetben van. A hallgató használja a StBeerCups tárgyat és véletlenszerűen elejt egy másikat. Megvédi magát az oktató ellen.
<b>Teszt célja</b>	A hallgató elejti egy tárgyát.

## 8.1 Osztályok és metódusok tervei.

### 8.1.1 Scheduler

- **Felelősség**

Ez az osztály valósítja meg a körök lebonyolítását és számon tartja a játékból hátralévő körök számát. A Scheduler kéri fel a külső szereplőket (Chart, Student, Professor, Cleaner), hogy hajtsák végre a lépésüket az adott körben.

- **Attribútumok**

- - **chart** : **Chart**: A játékban aktív pálya.
- - **characters** : **List<Character>**: A játékban aktív karakterek listája.

- **Metódusok**

- - **Init()** : **void**: Létrehozza a játékban résztvevő karaktereket és chart-ot.
- + **Play()** : **void**: Elindítja a játékot és körönként meghívja a következő karakter/chart megfelelő metódusát.

### 8.1.2 Chart

- **Felelősség**

A Chart osztály tartalmazza a pályán lévő szobákat. A Scheduler ezen keresztül érheti el őket.

- **Attribútumok**

- - **rooms** : **List<Room>**: A pályán lévő szobák listája.

- **Metódusok**

- - **Init()** : **void**: A pályán lévő szobákat hozza létre és építi fel a szerkezetét.
- + **IterateForRoomChanges()** : **void**: A pályán lévő szobák **Change()** metódusát hívja meg egyesével.
- + **GetAllRooms()** : **List<Room>**: A rooms tartalmát adja vissza.

### 8.1.3 Room

- **Felelősség**

A Room osztály felelős a szobák állapotáért. Ellenőrzi, hogy a szoba átkozott/gázos/ragacsos állapotát. A szobák változásáért is ő felel.

- **Attribútumok**

- - **maximumcapacity** : **int**: A szoba befogadóképességét tárolja el.
- - **characterCount** : **int**: A szobában tartózkodó karakterek számát adja meg.
- - **neighbours** : **List<Room>**: A szomszédos szobákat tartalmazza.
- - **isCursed** : **boolean**: Megadja, hogy a szoba elátkozott-e.
- - **items** : **List<Item>**: A szobában található tárgyakat tárolja.
- - **professors** : **List<Professor>**: A szobában lévő professzorok listája.
- - **students** : **List<Student>**: A szobában lévő hallgatók listája.
- - **cleaners** : **List<Cleaner>**: A szobában lévő takarítók listája.

- **-gas : Gas:** A szobában lévő gázt tartalmazó attribútum.
- **-goo : Goo:** A szobában lévő ragacsot tartalmazó attribútum.
- **+isFull: boolean:** Publikusan elérhető attribútum a szoba befogadóképességére vonatkozóan.
- **Metódusok**
  - **-Curse() : void:** Véletlenszerűen hozzáad egy új kétirányú átjárót egy másik szobához, valamint egyet a sajátjai közül eltüntet.
  - **-Split() : void:** A szoba létrehoz egy újat, amelynek a befogadóképessége az eredeti felének a felső egész része, az eredeti szoba megőrzi a befogadóképességét. Szomszédjait beállítja oly módon, hogy a saját szomszédjainak első felét megtartja, a másodikat pedig átadja az újonnan létrehozott szobának, valamint ez a két szoba is szomszédos lesz egymással. Az átjárók irányai nem változnak (egyirányú, kétirányú). A szobában lévő tárgyak hasonló módon oszlanak meg. Olyan szobában amiben tartózkodik karakter, nem osztható meg.
  - **-Merge(Room r) : void:** Két szomszédos szoba egyesülésekor a kisebb befogadóképességű szoba törlődik, a nagyobb befogadóképességű pedig megkap minden az eltüntetett szobában lévő tárgyat, valamint a szomszédjait is. Az átjárók irányai nem változnak.
  - **+GetNeighbours() : List<Room>:** Visszaadja a szobának a szomszédait, lekérdezi a neighbours attribútum tartalmát.
  - **+SetNeighbours(Room r2) : void:** Hozzáadja a neighbours listához a paraméterként kapott szobát, vagy ha már a szomszédja, akkor eltávolítja onnan, vagyis beállítja/eltünteti a szomszédját.
  - **+Change() : void:** Amennyiben a szoba elátkozott, adott valószínűséggel meghívja a Curse() metódust, majd ezzel megegyező módon meghívja a Split() vagy Merge() metódusokat.
  - **+SearchItem() : List<Item>:** visszaadja az items tartalmát, vagyis a szobában felvehető itemeket.
  - **+AddItem(Item i) : void:** A szoba items listájához hozzáadja a paraméterként kapott i Itemet.
  - **+PopItem(Item i) : Item:** A szobában lévő items lista közül visszaadja a kiválasztottat (i) és a saját listájából törli azt.
  - **+GetProfessors() : List<Professor>:** Visszaadja a professors lista tartalmát, vagyis az itt tartózkodó professzorok listáját.
  - **+GetStudents() : List<Student>:** Visszaadja a students lista tartalmát, vagyis az itt tartózkodó hallgatók listáját.
  - **+GetCleaners() : List<Cleaner>:** Visszaadja a cleaners lista tartalmát, vagyis az itt tartózkodó takarítók listáját.
  - **+CharacterEntered(Character currentCharacter, string characterName) : void:** Növeli eggyel a characterCount értékét, az újonnan érkező miatt. Ellenőrzi, hogy a szoba elérte-e a maximum kapacitását, vagyis a characterCount értéke egyenlő-e a maximumcapacity-val, ha igen, akkor az isFull igaz lesz, egyébként marad hamis. Ha a szobában van gáz (gas), akkor meghívja a gas objektum Gasify metódusát. Ha a szobában van ragacs(goo), akkor meghívja a goo objektum UpVisitorCount metódusát. Belerakja a name alapján a megfelelő listájába a kapott karakter objektumot, vagyis eltárolja a belépett karaktert.

- **+CharacterLeft(Character currentCharacter): void:**  
Csökkenti eggyel a characterCount értékét és beállítja az isFull értékét hamisra. Megkeresi a kapott objektumot valamelyik listában és kiveszi abból.
- **+AddGas() : void:** meghívja a Gas konstruktorát és inicializálja a gas attribútumot.
- **+GasExpired() : void:** gas attribútum null értékre állítása, eltünteti a szobából.
- **+HasGas() : boolean:** ha a gas null, akkor hamis, egyébként igaz értéket ad vissza.
- **+Clean() : void:** ha goo null, akkor meghívja a ragacs konstruktorát és inicializálja a goo attribútumot, egyébként meghívja a goo ResetVisitorCount metódusát.

### 8.1.4 Character

- **Felelősség**

A Character osztály felel a játékban szereplő karakterek nyilvántartásáért, léptetéséért illetve egyéb cselekvésekért. Absztrakt osztály.

- **Attribútumok**

- - **name: String:** A karakterek neve, az azonosításáért felelős.
- - **location: Room:** A karakter helyzetét tárolja.

- **Metódusok**

- **+ Turn() : void:** A játékban szereplő professzorok, diákok és takarítók körönkénti léptetését, illetve akcióit valósítja meg.
- **+ EnterRoom() : void** A karakter beállítja magának azt a szobát amibe belépett.

### 8.1.5 Professor

- **Felelősség**

A játék professzor szereplőinek mozgását, akcióit kezeli.

- **Össztályok**

Character

- **Interfészek**

IFighter

- **Attribútumok**

- -**name: String:** A karakterek neve, örökölt tulajdonság, azonosításra szolgál.
- -**location:Room:** A karakter helyzete, örökölt tulajdonság.
- -**isStunned: int:** tárolja, hogy a karakter hány körig kábult.

- **Metódusok**

- **+Turn() : void:** A professzor átlép egy random szomszédos szobába, ahol megeszi a benne lévő tárgyakat, ha vannak. Implementált metódus.
- **+EnterRoom() : void** A karakter beállítja magának azt a szobát amibe belépett. Implementált metódus.
- **+IsStunned() : void:** Visszaadja, hogy hány körig kábult. Implementált metódus.

- **+Stun() : boolean:** Átállítja az isStunned attribútumot True-ra és visszajelzést küld a sikerességről. Implementált metódus.
- **+Combat() : void:** A szobában tartózkodó hallgatóknak meghívja a Death() metódusát, ha nem kábult. Implementált metódus.

### 8.1.6 Student

- **Felelősség**

A játékos által vezérelt karaktert tartja nyilván, kezeli.

- **Ősosztályok**

Character

- **Interfészek**

IFighter

- **Attribútumok**

- **-name : String:** A karakterek neve, örökölt tulajdonság, azonosításra szolgál.
- **-location : Room:** A karakter helyzete, örökölt tulajdonság.
- **-items : List<Item>:** A hallgató által felvett tárgyak, maximum 5 db lehet
- **-inCombat : boolean:** Visszaadja, hogy az adott tanuló combat körben van-e
- **-isStunned : int:** tárolja, hogy a karakter kábult hány körig van kábult állapotban.
- **+isProtected : int:** Visszaadja, hogy az adott tanuló mennyi ideig védett.

- **Metódusok**

- **+Turn() : void:** A játékban szereplő professzorok és diákok körönkénti léptetését, illetve akcióit valósítja meg. Implementált metódus.
- **+EnterRoom() : void** A karakter beállítja magának azt a szobát amibe belépett. Implementált metódus
- **+Death() : void:** Amennyiben nem védett a hallgat, kidobja a szobába az összes tárgyát és törli magát belőle.
- **+GetLocation() : Room:** visszaadja a hallgató helyzetét.
- **+Drunk() : void:** Eldobhatja a hallgató egy véletlenszerűen kiválasztott tárgyát.
- **+IsStunned() : int:** Visszaadja, hogy hány körig kábult. Implementált metódus.
- **+Stun() : boolean:**Ha a hallgatónál nincsen maszk, akkor az isStunned értékét true-ra állítja, ellenkező esetben nem és visszajelzést küld a sikerességről. Implementált metódus.
- **+Combat() : void:** Választhat a tárgyai közül, amit használni szeretne a csata közben. Implementált metódus.



### 8.1.7 Cleaner

- **Felelősség**

A játék takarító szereplőinek mozgását és akcióit kezeli.

- **Ősosztályok**

Character

- **Attribútumok**

- **-name : String**: A karakterek neve, örökölt tulajdonság, azonosításra szolgál.
- **-location : Room**: A karakter helyzete, örökölt tulajdonság.

- **Metódusok**

- **+Turn() : void**: Belép egy random szomszédos szobába, ahonnan kiküldi a benne lévő karaktereket random szomszédos szobákba, ezek után meghívja a szoba Clean() metódusát. Implementált metódus.
- **+ EnterRoom() : void**: A karakter beállítja magának azt a szobát amibe belépett. Implementált metódus.

### 8.1.8 Goo

- **Felelősség**

A szobák ragacsosságáért felelős osztály.

- **Attribútumok**

- **-location : Room**: A ragacs tartózkodási helye.
- **-visitorCount : int**: A látogatók számát tárolja.

- **Metódusok**

- **+UpVisitorCount() : void**: Megemeli a látogatók számát 1-el, ha eléri az ötöt, akkor a szobában lévő tárgyak Glue() metódusát hívja meg.
- **+ResetVisitorCount() : void**: Nullára állítja a látogatók számát. Ha az addigra már elérte a küszöböt, akkor ezáltal feloldja a tárgyakra kifejtett hatását is, vagyis meghívja a tárgyak Glue() metódusát.

### 8.1.9 Gas

- **Felelősség**

Az adott szobák elgázosításáért felelős.

- **Attribútumok**

- **-location : Room**: A ragacs tartózkodási helye.

- **Metódusok**

- **+Gasify() : void**: Meghívja a szobában tartózkodó karakterek Stun() függvényét, ha valakit sikeresen elkábít, akkor kiveszi magát a játékból.

### 8.1.10 Item

- **Felelősség**

A különböző tárgyak realizációja. Absztrakt osztály.

- **Attribútumok**

- **-owner : Student:** A tárgy tulajdonosa, ha valakinek a hátizsákjában van.
- **-name : String:** A tárgy neve.

- **Metódusok**

- **+GetName() : String:** Visszaadja a tárgy nevét.
- **+ApplyEffect : void:** Az adott tárgy kifejti a hatását.
- **+Glue() : void:** Leragasztja vagy feloldja a ragasztást az adott tárgyon.
- **+IsGlued() : boolean:** Visszaadja, hogy a tárgy felvehető-e vagy sem.

### 8.1.11 StBeerCups

- **Felelősség**

A Hallgatónak, aki ezt megissza adott körön át immunitást biztosít a támadások ellen. Továbbá a még aktív söröskorsót használva a hallgatók elejtik az egyik náluk lévő tárgyat.

- **Összostályok**

Item

- **Attribútumok**

- **-owner : Student:** A tárgy tulajdonosa, ha valakinek a hátizsákjában van. Örökölt tulajdonság.
- **-name : String:** A tárgy neve. Örökölt tulajdonság.

- **Metódusok**

- **+GetName() : String:** Visszaadja a tárgy nevét. Implementált.
- **+ApplyEffect() : void:** A hallgató isProtected értékét megadott értékre állítja és meghívja a Drunk() metódusát. Implementált.
- **+Glue() : void:** Leragasztja vagy feloldja a ragasztást az adott tárgyon. Implementált metódus.
- **+IsGlued() : boolean:** Visszaadja, hogy a tárgy felvehető-e vagy sem. Implementált metódus

### 8.1.12 Camembert

- **Felelősség**

A szobákat el lehet gázosítani a tárgy használatával.

- **Ősosztályok**

Item

- **Attribútumok**

- **-owner : Student:** A tárgy tulajdonosa, ha valakinek a hátizsákjában van. Örökölt tulajdonság.
- **-name : String:** A tárgy neve. Örökölt tulajdonság.

- **Metódusok**

- **+GetName() : String:** Visszaadja a tárgy nevét. Implementált.
- **+ApplyEffect : void:** Meghívja a szoba AddGas() metódusát. Implementált.
- **+Glue() : void:** Leragasztja vagy feloldja a ragasztást az adott tárgyon. Implementált metódus.
- **+IsGlued() : boolean:** Visszaadja, hogy a tárgy felvehető-e vagy sem. Implementált metódus.

### 8.1.13 SlideRule

- **Felelősség**

A játék megnyeréséhez ezt a tárgyat kell birtokolni.

- **Ősosztályok**

Item

- **Attribútumok**

- **-owner : Student:** A tárgy tulajdonosa, ha valakinek a hátizsákjában van. Örökölt tulajdonság
- **-name : String:** A tárgy neve. Örökölt tulajdonság.

- **Metódusok**

- **+GetName():String:** Visszaadja a tárgy nevét. Implementált.
- **+ApplyEffect : void:** Amikor a hallgató felveszi, véget ér a játék. Nyertek a hallgatók. Implementált.
- **+Glue() : void:** Leragasztja vagy feloldja a ragasztást az adott tárgyon. Implementált metódus.
- **+IsGlued() : boolean:** Visszaadja, hogy a tárgy felvehető-e vagy sem. Implementált metódus.

### 8.1.14 Fakeltem

- **Felelősség**

A hamis tárgyakat valósítja meg, a felhasználó számára egy másik valós tárgy tulajdonságait mutatja, viszont annak képességeivel nem rendelkezik.

- **Ősosztályok**

Item

- **Attribútumok**

- **-owner : Student:** A tárgy tulajdonosa, ha valakinek a hátizsákjában van. Örökölt tulajdonság
- **-name : String:** A tárgy neve. Örökölt tulajdonság.

- **Metódusok**

- **+GetName() : String:** Visszaad egy véletlenszerű item nevet egy listából. Implementált.
- **+ApplyEffect : void:** Felvételnél egy szomszédos szobába helyezi a hallgatót. Implementált.
- **+Glue() : void:** Leragasztja vagy feloldja a ragasztást az adott tárgyon. Implementált metódus.
- **+IsGlued() : boolean:** Visszaadja, hogy a tárgy felvehető-e vagy sem. Implementált metódus.

### 8.1.15 FFP2Mask

- **Felelősség**

Megvédi a hallgatót a gázos szobáktól, ameddig a hátizsákjukban van.

- **Ősosztályok**

Item

- **Attribútumok**

- **-owner : Student:** A tárgy tulajdonosa, ha valakinek a hátizsákjában van. Örökölt tulajdonság
- **-name : String:** A tárgy neve. Örökölt tulajdonság.
- **-counter : int:** Számon tartja, hogy egymás után hány körig tud védeni a maszk.(minden használat után csökken).
- **+durability : int:** Mutatja, hogy hány kör maradt hátra, ameddig a gázban tartózkodhat.

- **Metódusok**

- **+GetName() : String:** Visszaadja a tárgy nevét. Implementált.
- **+ApplyEffect : void:** Immunissá teszi a hallgatót a gázra. Implementált.
- **+Glue() : void:** Leragasztja vagy feloldja a ragasztást az adott tárgyon. Implementált metódus.
- **+IsGlued() : boolean:** Visszaadja, hogy a tárgy felvehető-e vagy sem. Implementált metódus.

### 8.1.16 WetRag

- **Felelősség**

Oktatók elkábítására képes.

- **Ősosztályok**

Item

- **Attribútumok**

- **-owner : Student:** A tárgy tulajdonosa, ha valakinek a hátizsákjában van. Örökölt tulajdonság
- **-name : String:** A tárgy neve. Örökölt tulajdonság.
- **-moistness : int:** A WetRag nedvességét tartja számon.

- **Metódusok**

- **+GetName() : String:** Visszaadja a tárgy nevét. Implementált.
- **+ApplyEffect() : void:** Elkábítja a szobában tartózkodó oktatókat a nedvességével arányos ideig. Implementált.
- **+Glue() : void:** Leragasztja vagy feloldja a ragasztást az adott tárgyon. Implementált.
- **+IsGlued() : boolean:** Visszaadja, hogy a tárgy felvehető-e vagy sem. Implementált.

### 8.1.17 AirFreshener

- **Felelősség**

Ha gázos szobában vagyunk, semlegesíti a gáz hatását.

- **Ősosztályok**

Item

- **Attribútumok**

- **-owner : Student:** A tárgy tulajdonosa, ha valakinek a hátizsákjában van. Örökölt tulajdonság
- **-name : String:** A tárgy neve. Örökölt tulajdonság.

- **Metódusok**

- **+GetName() : String:** Visszaadja a tárgy nevét. Implementált.
- **+ApplyEffect() : void:** használata meghívja a szoba GasExpired() metódusát. Implementált.
- **+Glue() : void:** Leragasztja vagy feloldja a ragasztást az adott tárgyon. Implementált metódus.
- **+IsGlued() : boolean:** Visszaadja, hogy a tárgy felvehető-e vagy sem. Implementált metódus.

### 8.1.18 TVSZ

- **Felelősség**

A Hallgató háromszor tudja ezt a tárgyat használni arra, hogy a szobában lévő összes Tanárt elkábítása.

- **Ősosztályok**

Item

- **Attribútumok**

- **-owner : Student:** A tárgy tulajdonosa, ha valakinek a hátizsákjában van. Örökölt tulajdonság
- **-name : String:** A tárgy neve. Örökölt tulajdonság.
- **-uses : int:** Tárolja, hogy hány használat maradt benne

- **Metódusok**

- **+GetName() : String:** Visszaadja a tárgy nevét. Implementált.
- **+ApplyEffect() : void:** A hallgatót és a vele egy szobában lévő hallgatók isProtected értékét 1-re állítja és csökkenti a uses értékét. Implementált.
- **+Glue() : void:** Leragasztja vagy feloldja a ragasztást az adott tárgyon. Implementált metódus.
- **+IsGlued() : Boolean:** Visszaadja, hogy a tárgy felvehető-e vagy sem. Implementált metódus.

### 8.1.19 Transistor

- **Felelősség**

A Tranzisztorokat össze lehet kapcsolni páronként, amiket azután átjáróként lehet használni a szabályok szerint.

- **Ősosztályok**

Item

- **Attribútumok**

- **-owner : Student:** A tárgy tulajdonosa, ha valakinek a hátizsákjában van. Örökölt tulajdonság
- **-name : String:** A tárgy neve. Örökölt tulajdonság.
- **-pair : Transistor:** Az összepárosított tranzisztor másik párja.
- **+location : Room:** A tárgy helyzete, ha le van rakva.

- **Metódusok**

- **+GetName() : String:** Visszaadja a tárgy nevét. Implementált.
- **+ApplyEffect() : void:** A tárgyat bekapcsolja és a földre rakja a Hallgatóval azonos szobában, ha van párja. Implementált.
- **+Glue() : void:** Leragasztja vagy feloldja a ragasztást az adott tárgyon. Implementált.
- **+IsGlued() : boolean:** Visszaadja, hogy a tárgy felvehető-e vagy sem. Implementált.
- **+Activate() : void:** A hallgatót a tranzisztor párjához teleportálja.
- **+SetPair() : void:** Beállítja a tranzisztor párját.
- **+HasPair() : boolean:** Lekérdezi, hogy van-e párja a tranzisztornak.

## 8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

### 8.2.1 Oktatók-tárgyak interakciók

- **Leírás**

Az oktató belép a szobába. A szobában vannak tárgyak (igaziak és hamisak is). Az oktató megemészti a tárgyakat, ezáltal a szobából mindegyik eltűnik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az oktatók tárgy eltüntető-képességét teszteljük

- **Bemenet**

```
init_profItem.txt:
    AddRoom r1
    AddRoom r2
    SetNeighbour r1 r2
    AddCharacter p1 Professor r1
    SpawnItem r2 i1 Camembert true
    SpawnItem r2 i2 TVSZ false
```

```
test_profItem.txt:
    Init init_profitem.txt
    EnterRoom p1 r2
    Info r2
```

- **Elvárt kimenet**

```
result_profItem.txt:
    Room r1 added to chart
    Room r2 added to chart
    Room r2 set as neighbour of Room r1
    Character p1 added to Room r1
    Normal Item Camembert i1 has been added to Room r2
    Fake Item i2 TVSZ has been added to Room r2

    Chart loaded
    Character p1 has entered Room r2
    ...
    r2.items 0
    ...
```

### 8.2.2 Professzor gázos szobába lép

- **Leírás**

A professzor belép egy szobába. A szoba éppen gázos állapotban van, aminek hatására professzor elkábul és a szobából eltűnik a gáz.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A gáz hatásának tesztelése a professzorokra nézve.

- **Bemenet**

```
init_profGas.txt:
    AddRoom r1
    AddRoom r2
    SetNeighbour r1 r2
    AddCharacter p1 Professor r1
    GasRoom r2
```

```
test_profGas.txt:
    Init init_profgas.txt
    EnterRoom p1 r2
    Info r2
    Info p1
```

- **Elvárt kimenet**

```
result_profGas.txt:
    Room r1 added to chart
    Room r2 added to chart
    Room r2 set as neighbour of Room r1
    Character p1 added to Room r1
    Room r2 gasified

    Chart loaded
    Character p1 has entered Room r2
    ...
    r2.gas: null
    ...
    p1.isStunned: positive
    ...
```

### 8.2.3 Professzor aktívan ragacsos szobába lép

- **Leírás**

A professzor belép egy szobába. A szoba éppen aktív ragacsos állapotban van, ami miatt a professzor nem tudja felemészteni a tárgyakat.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A ragacs hatásának tesztelése a professzorokra és tárgyakra nézve.

- **Bemenet**

```
init_profGoo.txt:
    AddRoom r1
    AddRoom r2
    SetNeighbour r1 r2
    AddCharacter p1 Professor r1
    SpawnItem r2 i1 Camembert true
    SpawnItem r2 i2 TVSZ false
    GooRoom r2 glued
```

```
test_profGoo.txt:
    Init init_profgoo.txt
    EnterRoom p1 r2
    Info r2
```



- **Elvárt kimenet**

```
result_profGoo.txt:
    Room r1 added to chart
    Room r2 added to chart
    Room r2 set as neighbour of Room r1
    Character p1 added to Room r2
    Normal Item Camembert i1 has been added to Room r2
    Fake Item TVSZ i2 has been added to Room r2
    Goo activated in Room r2, items glued

    Chart loaded
    Character p1 has entered Room r2
    ...
    r2.items 2
    ...
```

#### 8.2.4 Professzor-hallgató interakció

- **Leírás**

A hallgató belép egy szobába. Az adott szobába már éppen tartózkodik egy nem kábult oktató. A következő kör combat lesz.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A hallgató oktató közös szobában tartózkodásakor harchelyzet alakul ki.

- **Bemenet**

```
init_studentProf.txt:
    AddRoom r1
    AddRoom r2
    SetNeighbour r1 r2
    AddCharacter p1 Professor r2
    AddCharacter s1 Professor r1
```

```
test_studentProf.txt:
    Init init_studentprof.txt
    EnterRoom s1 r2
    Info s1
```

- **Elvárt kimenet**

```
result_studentProf.txt
    Room r1 added to chart
    Room r2 added to chart
    Room r2 set as neighbour of Room r1
    Character p1 added to Room r2
    Character s1 added to Room r1

    Chart loaded
    Character s1 has entered Room r2
    ...
    s1.inCombat: true
```

### 8.2.5 Kábult professzor-hallgató interakció

- **Leírás**  
A hallgató belép egy szobába. Az adott szobába már éppen tartózkodik egy kábult oktató. Nem lesz combat kör, a hallgató folytatja a körét, itemeket vehet fel, használhat stb.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A hallgató és kábult oktató közös szobában tartózkodásakor nem alakul ki harchelyzet.

- **Bemenet**

```
init_studentStunnedProf.txt:
    Init init_profgas.txt
    AddCharacter s1 Student r1
```

```
test_studentStunnedProf.txt:
    Init init_studentprof.txt
    EnterRoom s1 r2
    Info s1
```

- **Elvárt kimenet**

```
result_studentStunnedProf.txt:
    Room r1 added to chart
    Room r2 added to chart
    Room r2 set as neighbour of Room r1
    Character p1 added to Room r1
    Room r2 gasified

    Chart loaded
    Character p1 has entered Room r2
    ...
    r2.gas: null
    ...
    p1.isStunned: positive
    ...

    Character s1 added to Room r1

    Chart loaded
    Character s1 has entered Room r2
    ...
    s1.inCombat: false
    ...
```

### 8.2.6 Hallgató-tárgy interakció

- **Leírás**  
A hallgató belép a szobába, ahol található egy vagy több tárgy. A hallgatónak még nincs tele a hátizsákja, ezért felvesz egyet. A szoba tárgyai közül a hallgató által felvett törlődik.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A tárgyfelvétel tesztelése a hallgatók esetében, illetve ellenőrzés a szoba helyes tárgy törléséről.

- **Bemenet**

```
init_studentItemPick.txt:
    AddRoom r1
    AddRoom r2
    SetNeighbour r1 r2
    AddCharacter s1 Student r1
    SpawnItem r2 TVSZ i1 true

test_studentItemPick.txt
    Init init_StudentItemPick.txt
    EnterRoom s1 r2
    PickupItem s1 i1
    Info r2
    Info s1
```

- **Elvárt kimenet**

```
result_studentItemPick.txt
    Room r1 added to chart
    Room r2 added to chart
    Room r2 set as neighbour of Room r1
    Character s1 added to Room r1

Normal Item TVSZ i1 has been added to Room r2

Chart loaded
Character s1 has entered to Room r2
Character s1 picked up Item i1
...
r2.items 0
...
s1.items 1
```

## 8.2.7 Hallgató elkábulásának hiánya

- **Leírás**

A hallgató belép a szobába, ahol felvesz egy FFP2 maszkot. Ezután átlép egy másik szobába, ami gázos állapotban van. A hallgató nem kábul el, ugyanis a maszk megvédi őt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A maszk védelmének tesztelése

- **Bemenet**

```
init_studentImmuneToGas.txt
    AddRoom r1
    AddRoom r2
    AddRoom r3
    SetNeighbour r1 r2
    SetNeighbour r2 r3
    GasRoom r3
    AddCharacter s1 Student r1
    SpawnItem r2 FFP2Mask i1 true
```

```
test_studentImmuneToGas.txt
  Init init_studentImmuneToGas.txt
  EnterRoom s1 r2
  PickupItem s1 i1
  EnterRoom s1 r3
  Info s1
```

- **Elvárt kimenet**

```
result_studentImmuneToGas.txt
  Room r1 added to chart
  Room r2 added to chart
  Room r3 added to chart
  Room r2 set as neighbour of Room r1
  Room r3 set as neighbour of Room r2
  Room r3 gasified
  Character s1 added to Room r1
  Normal Item FFP2MASK i1 has been added to Room r2

  Chart loaded
  Character s1 has entered to Room r2
  Character s1 picked up Item i1
  Character s1 has entered to Room r3
  ...
  s1.isStunned : 0
```

### 8.2.8 Hallgató halála

- **Leírás**

A hallgató combat körben van éppen egy vagy több tanárral. Nem tudja megvédeni magát, a professzor megöli. Az elhunyt hallgató kikerül a szobából és a játékból is, a combat körnek vége.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A professzor hatásának tesztelése a védtelen hallgatóra, a hallgató eltűnésének tesztelése.

- **Bemenet**

```
init_studentDies.txt
  AddRoom r1
  AddCharacter s1 Student r1
  AddCharacter p1 Professor r1
test_studentDies.txt
  Init init_StudentDies.txt
  CombatRoom r1
  Info r1
```

- **Elvárt kimenet**

```
result_studentDies.txt
    Room r1 added to chart
    Character s1 added to Room r1
    Character p1 added to Room r1

    Chart loaded
    s1, p1 are in combat
    ...
    r1.students 0
```

### 8.2.9 Hallgató elkábulása

- **Leírás**

A hallgató belép a szobába, ami éppen gázos állapotban van. A diákon nincsen semmiféle védőeszköz a gázossággal szemben. A diák elkábul. Ezután a gáz eltűnik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A hallgató elkábul a gáz hatására. A gáz eltűnik a szobából.

- **Bemenet**

```
init_studentNotImmuneToGas.txt
    AddRoom r1
    AddRoom r2
    SetNeighbour r1 r2
    GasRoom r2
    AddCharacter s1 Student r1

test_studentNotImmuneToGas.txt
    Init init_studentNotImmuneToGas.txt
    EnterRoom s1 r2
    Info r2
    Info s1
```

- **Elvárt kimenet**

```
result_studentNotImmuneToGas.txt
    Room r1 added to chart
    Room r2 added to chart
    Room r2 set as neighbour of Room r1
    Room r2 gasified
    Character s1 added to Room r1

    Chart loaded
    Character s1 has entered Room r2
    ...
    r2.gas : null
    ...
    s1.isStunned : positive
```

### 8.2.10 Hallgató használja a Camembertet

- **Leírás**  
A hallgató belép a szobába, és kinyitja a Camembertet. Ettől a szoba gázossá válik.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A szoba gázossá válik.

- **Bemenet**

```
init_studentUseCamembert.txt
    AddRoom r1
    AddCharacter s1 Student r1
    AddItem s1 Camembert i1 true

test_studentUseCamembert.txt
    Init init_studentUseCamembert.txt
    UseItem s1 i1
    Info r1
```

- **Elvárt kimenet**

```
result_studentUseCamembert.txt
    Room r1 added to chart
    Character s1 added to Room r1
    Normal Item Camembert i1 has been added to Student
    s1

    Chart loaded
    Character s1 used Item i1
    ...
    r1.gas : notnull
```

### 8.2.11 Tranzisztorok használata

- **Leírás**  
A hallgató belép egy szobába, ahol lerakja a már előre összepárosított tranzisztorok egyikét. A másikat már egy korábbi szobában elhelyezte. Ezzel vissza teleportál abba a szobába, ahol az előzőt elhelyezte.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A hallgató belép egy szobába, ahol lerakja a már előre összepárosított tranzisztorok egyikét. A másikat már egy korábbi szobában elhelyezte. Ezzel vissza teleportál abba a szobába, ahol az előzőt elhelyezte.

- **Bemenet**

```
init_studentUseTransistor.txt
    AddRoom r1
    AddRoom r2
    SetNeighbours r1 r2
    AddCharacter s1 Student r1
    AddItem s1 Transistor i1 true
    AddItem s1 Transistor i2 true
    SetPair i1 i2
```

```
test_studentUseTransistor.txt
  init_studentUseTransistor.txt
  UseItem s1 i1
  EnterRoom s1 r2
  ActivateTransistor i2
  Info s1
  Info r2
```

- **Elvárt kimenet**

```
result_studentUseTransistor.txt
  Room r1 added to chart
  Room r2 added to chart
  Room r2 set as neighbour of Room r1
  Character s1 added to Room r1
  Normal Item Transistor i1 has been added to Student
  s1
  Normal Item Transistor i2 has been added to Student
  s2

  Chart loaded
  Character s1 used Item i1
  Character s1 has entered to Room r2
  Character s1 activated Item i2
  ...
  s1.location : r1
  ...
  r2.items 1
  Transistor i1
```

## 8.2.12 TVSZ használata

- **Leírás**

Egy hallgató egy másik hallgatóval együtt van combat körben, egy szobában. Az 1-es számú hallgató nem rendelkezik tárgyakkal, viszont a 2-es számúnál van egy TVSZ. A 2-es számú hallgató ezt az eszközt aktiválva megmenti a társát és saját magát is. Combat kör véget ér.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

TVSZ működése, hiba lehet, ha a tanár véghezviszi a gyilkolást így is

- **Bemenet**

```
init_studentUseTVSZ.txt
  AddRoom r1
  AddCharacter s1 Student r1
  AddCharacter s2 Student r1
  AddItem s2 TVSZ i1 true
  AddCharacter p1 Professor r1

test_studentUseTVSZ.txt
  UseItem s2 i1
  CombatRoom r1
  Info r1
```

- **Elvárt kimenet**

```
result_studentUseTVSZ.txt
Room r1 added to chart
Character s1 added to Room r1
Character s2 added to Room r2
Normal Item TVSZ i1 has been added to Student s1
Character p1 added to Room r1

Chart loaded
Character s1 used Item i1
s1, s2, p1 are in combat
...
r1.students 2
```

### 8.2.13 WetRag használata

- **Leírás**

Két hallgató is combat körben van egy szobában. Az első hallgató nem tudja megvédeni magát, ő így meghalna. A második hallgató használja a vizes táblatörő rongy tárgyat. Ekkor a tárgy jelzést ad, az összes szobában lévő oktatót lebénítja. Mindkettő hallgató megmenekül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A WetRag működésének tesztelése, professzorok elkábulása

- **Bemenet**

```
init_WetRag.txt
AddRoom r1
AddCharacter p1 Professor r1
AddCharacter s1 Student r1
AddCharacter s2 Student r1
SpawnItem r1 WetRag i1 true
AddItem s2 WetRag i1 true
```

```
test_WetRag.txt
Init init_WetRag.txt
UseItem s2 i1
CombatRoom r1
Info r1
Info p1
Info p2
```

- **Elvárt kimenet**

```
result_WetRag.txt
Room r1 added to chart
Character p1 added to Room r1
Character p2 added to Room r1
Character s1 added to Room r1
Character s2 added to Room r1
Normal Item WetRag i1 has been added to Room r1
Character s2 picked up Item i1

Chart loaded
Character s2 used Item i1
```



```
s1, s2, p1, p2 are in combat
...
r1.students 2
...
p1.isStunned: positive
...
p2.isStunned: positive
```

#### 8.2.14 Hallgatók nyernek

- **Leírás**

A hallgató bemegy a szobába, és felveszi a Logarlécet, ugyanis az a szobában található. A hallgatók ezzel megnyerték a játékot. A Scheduler leállítja a játékot.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Logarléc működése

- **Bemenet**

```
init_studentsWin.txt
  AddRoom r1
  AddCharacter s1 Student r1
```

```
test_studentsWin.txt
  Init init_studentsWin.txt
  PickupItem r1 i1
```

- **Elvárt kimenet**

```
result_studentsWin.txt
  Room r1 added to chart
  Character s1 added to Room r1

  Chart loaded
  Character s1 picked up Item i1

  Students have won
```

#### 8.2.15 Oktatók nyernek

- **Leírás**

Ha az oktatók az utolsó életben lévő hallgatót is megölik, akkor a Scheduler jelzést kap, hogy nincs több hallgató. Az oktatók nyertek, a játék leáll.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A hallgatók hiányának ellenőrzése

- **Bemenet**

```
init_professorsWin.tx
  AddRoom r1
  AddCharacter s1 Student r1
  AddCharacter p1 Professor r1
```

```
test_professorsWin.txt
  Init init_professorsWin.txt
  CombatRoom r1
```

- **Elvárt kimenet**

```
result_professorsWin.txt
    Room r1 added to chart
    Character s1 added to Room r1

    Chart loaded
    s1, p1 are in combat
    Professors have won
```

### 8.2.16 Takarítónő kitesszékelés

- **Leírás**

A takarítónő belép egy szobába, ahol mások is tartózkodnak. Minden más karakter elhagyja a szobát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A takarítónő kitesszékelő felhívására való reagálás

- **Bemenet**

```
init_cleanerThrowOut.txt
    AddRoom r1
    AddRoom r2
    SetNeighbours r1 r2
    AddCharacter s1 Student r2
    AddCharacter s2 Student r2
    AddCharacter c1 Cleaner r1
```

```
test_cleanerThrowOut.txt
    EnterRoom c1 r2
    Info r2
```

- **Elvárt kimenet**

```
result_cleanerThrowOut.txt
    Room r1 added to chart
    Room r2 added to chart
    Room r2 set as neighbour of Room r1
    Character s1 added to Room r2
    Character s2 added to Room r2
    Character c1 added to Room r1

    Chart loaded
    Character c1 has entered to Room r2
    ...
    r2.students 0
    ...
    r2.professors 0
    ...
    r2.cleaners 1
```

**8.2.17 Takarítónő kábultak kitessékelése**

- **Leírás**

A takarítónő belép egy szobába, ahol mások is tartózkodnak, de el vannak kábulva. Ekkor nem hagyja el a szobát senki.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A takarítónő kitessékelő felhívására való reagálás

- **Bemenet**

```
init_cleanerStunned.txt
  AddRoom r1
  AddRoom r2
  SetNeighbours r1 r2
  AddCharacter s1 Student r2
  AddCharacter s2 Student r2
  AddCharacter c1 Cleaner r1
  AddItem s1 Camembert i1 true
  UseItem s1 i1
  Info s1
  Info s2
```

```
test_cleanerStunned.txt
  EnterRoom c1 r1
  Info r2
```

- **Elvárt kimenet**

```
result_cleanerStunned
  Room r1 added to chart
  Room r2 added to chart
  Room r2 set as neighbour of Room r1
  Character s1 added to Room r2
  Character s2 added to Room r2
  Character c1 added to Room r1
  Normal Item Camembert i1 has been added to Student
  s1
  Character s1 used Item i1
  ...
  s1.isStunned : positive
  ...
  s2.isStunned : positive
  ...

  Chart loaded
  Character c1 has entered to Room r2
  ...
  r2.students 2
```

### 8.2.18 Takarítónő takarítás

- **Leírás**  
A takarítónő belép egy szobába. Ekkor minden, ott lévő tárgy ragacsossága nullára változik, ezáltal felvehetők lesznek.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A takarítónő takarításának tesztelése.

- **Bemenet**

```
init_clean.txt
  AddRoom r1
  AddRoom r2
  SetNeighbours r1 r2
  AddCharacter s1 Student r1
  AddCharacter c1 Cleaner r1
  SpawnItem r2 i1 Camembert true
  GooRoom r2 glued
```

```
test_clean.txt
  Init init_clean.txt
  EnterRoom c1 r2
  EnterRoom s1 r2
  PickupItem s1 i1
```

- **Elvárt kimenet**

```
result_clean.txt
  Room r1 added to chart
  Room r2 added to chart
  Room r2 set as neighbour of Room r1
  Character s1 added to Room r1
  Character c1 added to Room r1
  Normal Item Camembert i1 has been added to Room r2
  Goo activated in Room r1, items glued

  Chart loaded
  Character c1 has entered to Room r2
  Character s1 has entered to Room r2
  Character s1 picked up Item i1
```

### 8.2.19 Takarítónő szellőztet

- **Leírás**  
A takarítónő belép egy gázos szobába. A takarítónő kiszellőztet, ezáltal a szoba megszűnik gázosnak lenni.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A gázosság eltűnése

- **Bemenet**

```
init_cleanerVentilates.txt
  AddRoom r1
  AddRoom r2
  SetNeighbours r1 r2
  AddCharacter c1 Cleaner r1
  GasRoom r2
```

```
test_cleanerVentilates.txt
  EnterRoom c1 r2
  Info r2
```

- **Elvárt kimenet**

```
result_cleanerVentilates.txt
  Room r1 added to chart
  Room r2 added to chart
  Room r2 set as neighbour of Room r1
  Character c1 added to Room r1
  Room r2 gasified

  Chart loaded
  Character c1 has entered to Room r1
  r2.gas : null
```

### 8.2.20 Takarítónő telített szobába

- **Leírás**

A takarítónő olyan szobába próbál belépni, ami már telített. Ez nem sikerül neki, ezért egy másik szobába lép.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Telítettség-ellenőrzés

- **Bemenet**

```
init_cleanerFull.txt
  AddRoom r1
  AddRoom r2
  SetNeighbours r1 r2
  SetRoomFull r2
  AddCharacter c1 Cleaner r1
```

```
test_cleanerFull.txt
  Init init_cleanerFull.txt
  EnterRoom c1 r2
```

- **Elvárt kimenet**

```
result_cleanerFull.txt
  Room r1 added to chart
  Room r2 added to chart
  Room r2 set as neighbour of Room r1
  Room r2 is full
  Character c1 added to Room r1

  Chart loaded
  Character c1 could not enter Room r2
```

### 8.2.21 Ragacsos tárgyak felvétele

- **Leírás**  
A diák egy szobába megpróbál felvenni tárgyakat. A szobában azonban már elérte a küszöbértéket a ragacsosság. Ekkor a hallgató nem tud felvenni egyetlen tárgyat sem.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A ragacsosság hatásának ellenőrzésre

- **Bemenet**

```
init_gluedItem.txt
  AddRoom r1
  AddCharacter s1 Student r1
  SpawnItem r1 i1 Camembert true
  GooRoom r1 glued
```

```
test_gluedItem.txt
  Init gluedItem.txt
  PickupItem s1 i1
```

- **Elvárt kimenet**

```
result_gluedItem.txt
  Room r1 added to chart
  Character s1 added to Room r1
  Normal Item Camembert i1 has been added to Room r1
  Goo activated in Room r1, items glued
```

```
Chart loaded
Character s1 could not pick up Item i1
```

### 8.2.22 Légfrissítő használata

- **Leírás**  
Egy hallgató használja a légfrissítőt egy gázos szobában. A gázosság megszűnik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A gázosság eltűnése

#### Bemenet

```
init_airFreshener.txt
  AddRoom r1
  AddRoom r2
  SetNeighbours r1 r2
  AddCharacter s1 Student r1
  AddItem s1 FFP2Mask i1 true
  AddItem s1 AirFreshener i2 true
  GasRoom r2
```

```
test_airFreshener.txt
  Init init_airFreshener.txt
  EnterRoom s1 r2
  UseItem s1 i2
  Info r2
```

- **Elvárt kimenet**

```
result_cleanerVentilates.txt
Room r1 added to chart
Room r2 added to chart
Room r2 set as neighbour of Room r1
Character s1 added to Room r1
Normal Item FFP2Mask i1 has been added to Student s1
Normal Item AirFreshener i2 has been added to
Student s1
Room r2 gasified

Chart loaded
Character s1 has entered Room r2
Character s1 used Item i2
...
r2.Gas : null
```

### 8.2.23 Hamis tárgy felvétele

- **Leírás**

A hallgató egy hamis tárgyat vesz fel, ami azonnal kifejti a hatását, egy szomszédos szobába helyezi át.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A hallgató átkerül egy másik szobába.

- **Bemenet**

```
init_fakeItem.txt
AddRoom r1
AddRoom r2
SetNeighbours r1 r2
AddCharacter s1 Student r1
SpawnItem r1 FFP2Mask i1 false
```

```
test_fakeItem.txt
Init init_fakeItem.txt
PickupItem s1 i1
Info r1
Info r2
```

- **Elvárt kimenet**

```
result_fakeItem.txt
Room r1 added to chart
Room r2 added to chart
Room r2 set as neighbour of Room r1
Character s1 added to Room r1
Fake Item FFP2Mask i1 has been added to Room r1

Chart loaded
Character s1 picked up Item i1
...
r1.students 0
...
r2.students 1
```

### 8.2.24 StBeerCups használata

- **Leírás**  
A hallgató és oktató harchelyzetben van. A hallgató használja a StBeerCups tárgyat és véletlenszerűen elejt egy másikat. Megvédi magát az oktató ellen.
- **Ellenőrzött funkcionalitás, várható hibahelyek**  
A hallgató elejti egy tárgyat.

- **Bemenet**

```
init_useStBeerCups.txt
  AddRoom r1
  AddCharacter s1 Student r1
  AddCharacter p1 Professor r1
  AddItem s1 StBeerCups i1 true
  AddItem s1 Camembert i2 true
```

```
test_useStBeerCups.txt
  Init init_useStBeerCups.txt
  UseItem s1 i1
  CombatRoom r1
  Info s1
  Info r1
```

- **Elvárt kimenet**

```
result_useStBeerCups.txt
  Room r1 added to chart
  Character s1 added to Room r1
  Character p1 added to Room r1
  Character s1 picked up Item i1
  Character s1 picked up Item i1

  Chart loaded
  Character s1 used Item i1
  s1, p1 are in combat
  ...
  s1.items 0
  ...
  r1.items 1
  ...
  r1.students 1
  ...
```



### 8.3 A tesztelést támogató programok tervei

Minden tesztesethez 4 + 1 fájl tartozik:

- **init\_[tesztnév].txt**
- **test\_[tesztnév].txt**
- **result\_[tesztnév].txt**
- **tester\_[tesztnév].sh**
- **proto.jar**

**init:** A pálya építéséhez szükséges parancsokat tartalmazza.

**test:** A tesztelt funkcionalitást lefolytató parancsok vannak benne. Tartalmazza az Init parancsot is, ami betölti az init fájlban található pályát.

**result:** A teszt lefutása során keletkezett kimeneteket magába foglaló txt fájl.

**tester:** Bash-ben írt futtatható állomány, analizálja a result fájlt, majd kiértékeli azt.

**proto.jar:** A prototípus program egy fájlban megadott, futtatható formátuma. Paraméterként megadhatunk neki egy txt fájlt amit feldolgoz és a benne található parancsokat lefuttatja.

A teszt elindításához csak futtatni kell a tester állományt. Létezik olyan teszt ami az összes tesztet lefuttatja, ehhez egyesével lefuttatja a tester fájlokat.

#### Fájl fastruktúra:

```
root_folder
->proto.jar
->test_all.sh
->[testname1]
  ->init_[testname1].txt
  ->test_[testname1].txt
  ->result_[testname1].txt
  ->tester_[testname1].sh
->[testname2]
  ...
->[testname3]
  ...
```

**Kimenet:** Minden teszt kiértékelése egy PASSED vagy FAIL szöveggel ér véget, így jelezve, hogy sikeres, vagy sikertelen volt a teszt.

**8.4 Napló**

Kezdet	Időtartam	Résztevők	Leírás
			Értekezlet.
2024.04.13 20:00	2,5 óra	Kiss	Korábbi bemeneti és kimeneti nyelv módosításai, tesztesetek
2024.04.13. 20:00	2,5 óra	Osztrogonác	Objektum és metódus tervek, Room osztály
2024.04.13 23:00	1,25 óra	Halász	Osztályok leírása
2024.04.14. 10:15	1,75 óra	Osztrogonác Kiss	Tesztek írása és osztályok
2024.04.14 17:05	1,5 óra	Halász	Tesztek
2024.04.14 18:30	3 óra	Schmidt	Osztályok és tesztek
2024.04.14. 21:00	1 óra	Osztrogonác	Dokumentum szerkesztés, konzisztencia visszaállítása
2024.04.14 19:30	2,25 óra	Gáspár	Tesztek
2024.04.15. 9:00	3 óra	Osztrogonác Kiss	Dokumentum renderakása, utolsó tesztek hozzáadása, összes teszt, osztályok egységessé tétele
2024.04.15 9:35	25 perc	Gáspár	Teszt