

# 7. Prototípus koncepciója

70 – Edgerunners

Konzulens:

Szabó Bence Sándor

## Csapattagok

Csabuda Nóra  
Gulybán Dániel  
Nagy Gergely  
Papp Levente  
Sánta Dániel

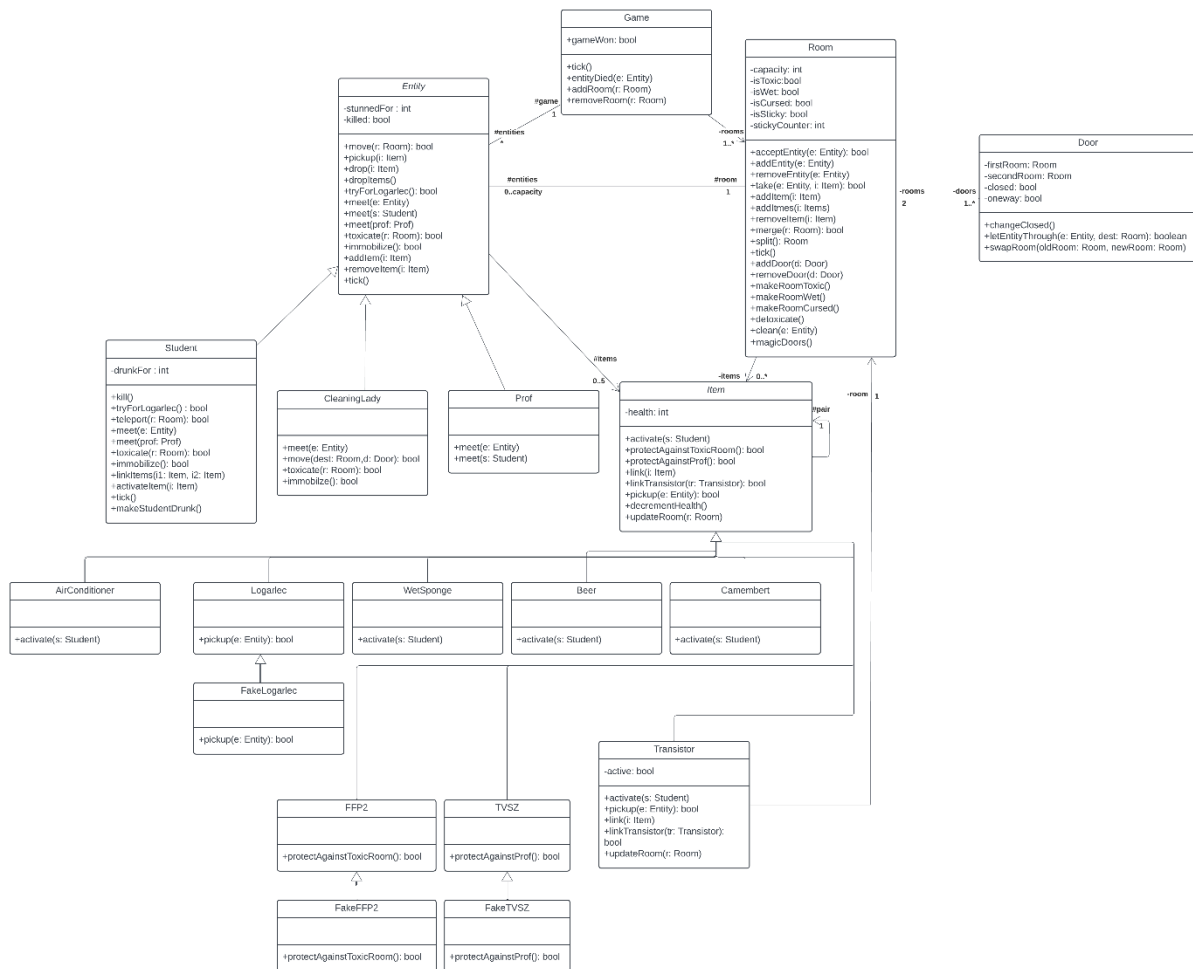
GHDCTP  
HUCHEX  
HC1QJP  
HV0CRH  
OGLSI2

csabudan@gmail.com  
gulyban.daniel@gmail.com  
nagy2002@gmail.com  
papp.levente2003@gmail.com  
daniel.santa0615@gmail.com

## 7. Prototípus koncepciója

### 7.0 Változás hatása a modellre

#### 7.0.1 Módosult osztálydiagram



#### 7.0.2 Új vagy megváltozó metódusok

##### AirConditioner

- **void activate(e: Entity):** Meghívja a szükséges függvényt a szoba megtisztításához a gáztól, és lecsökkenti a tárgy health-ét.

##### Room

##### Tagváltozók:

- **bool isSticky:** Ezen attribútum bool értékkel tárolja, hogy az adott szoba ragadós-e vagy még nem.
- **int stickyCounter:** Ezen attribútum alapvetően mínusz értékekkel számol(kezdetben mínusz egy) és folyton csökken, ha új látogató érkezik az adott szobába. Amennyiben takarító érkezik szobába megadott plusz értékre állítja ezen változó értékét és ha nulla értékre csökken vissza, a szoba ragadóssá válik.

Metódusok:

- **void addEntity():** A szoba entitásaihoz ad egy újabbat és elvégzi a látogatók számlálásához szükséges lépést.
- **void detoxicate():** Az adott szoba, nem gázos szobává változik.
- **void clean(e:Entity):** Az adott szobából „kitakarítja” a karaktereket, gáztalanítja is a szobát és beállítja a ragadós padló elődizéséhez szükséges dolgokat.

FakeFFP2

- **bool protectAgainstToxicRoom():** Ezen metódus felülírásának köszönhetően a hamis tárgy nem tud semmilyen hatást kiváltani

FakeLogarlec

- **bool pickup(e: entity):** Ezen metódus felülírásának köszönhetően a hamis tárgy nem tud semmilyen hatást kiváltani

FakeTVSZ

- **bool protectAgainstProf():** Ezen metódus felülírásának köszönhetően a hamis tárgy nem tud semmilyen hatást kiváltani

Student

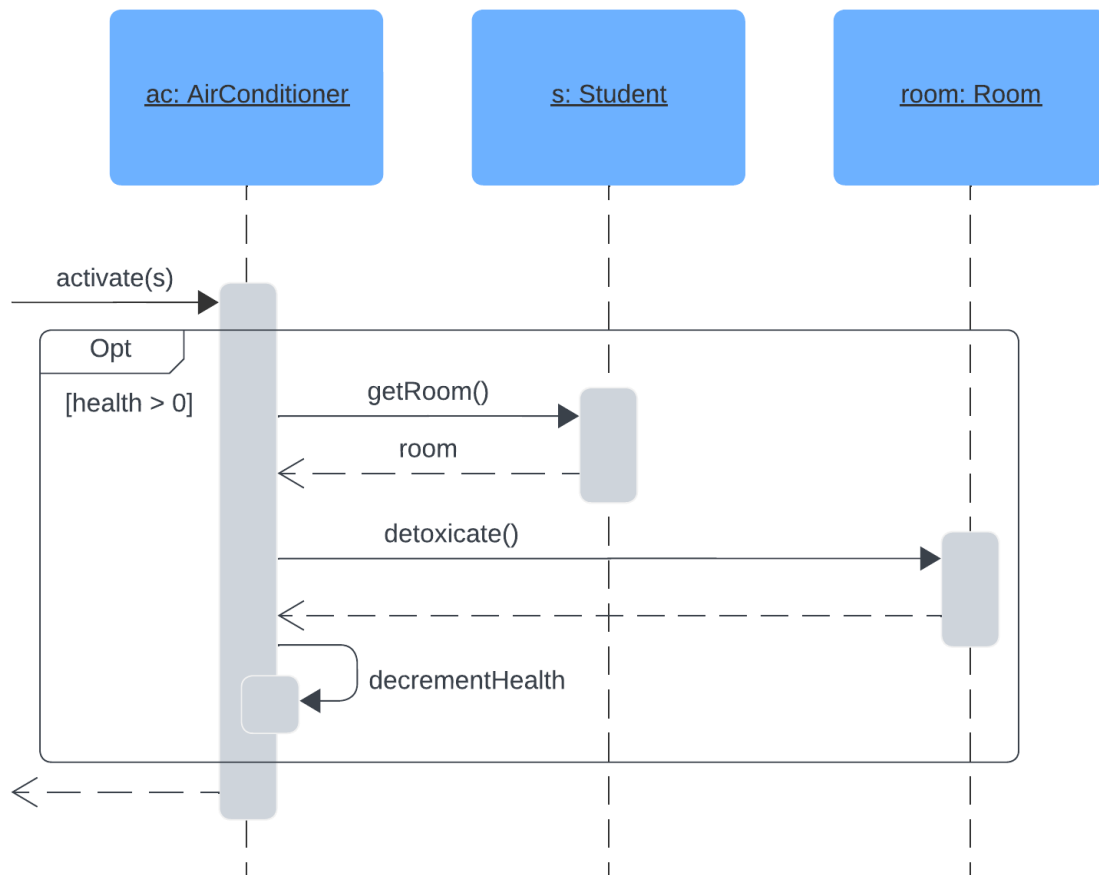
- **void kill():** A hívott karaktert megpróbálja megölni ezen metódus. Ha éppen még részeg a hallgató és ez megvédi ez, akkor elejtésre kerül egy tárgya.

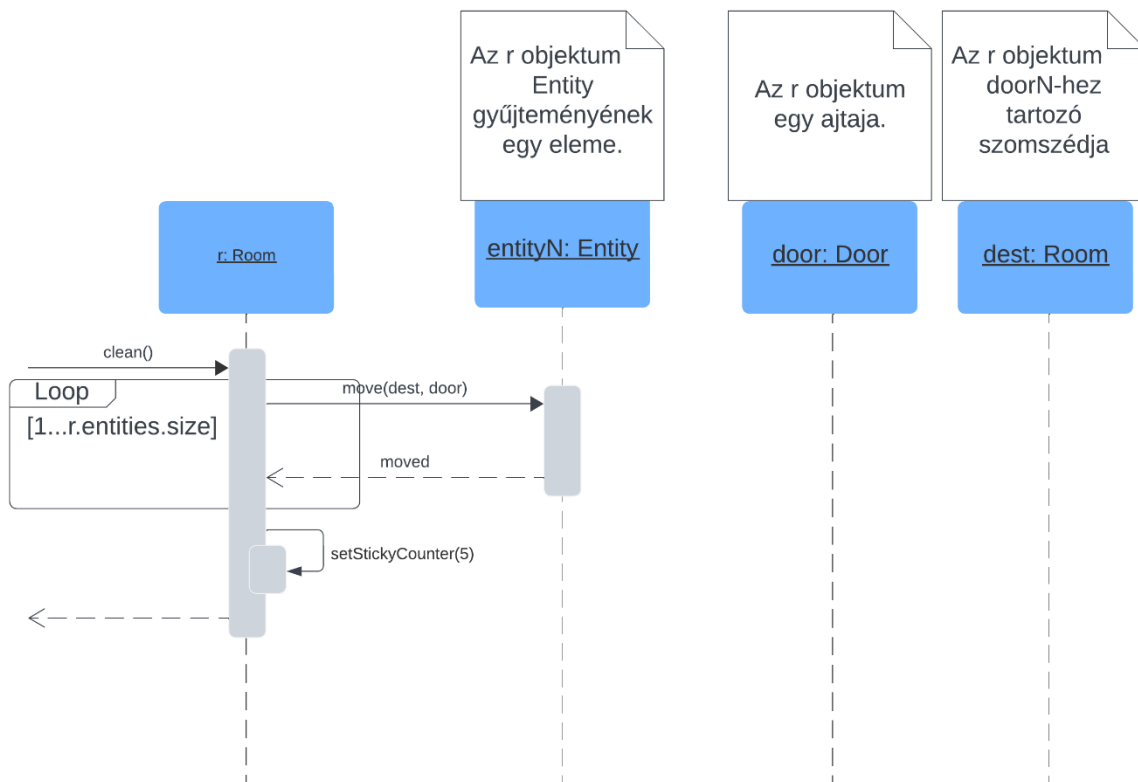
CleaningLady

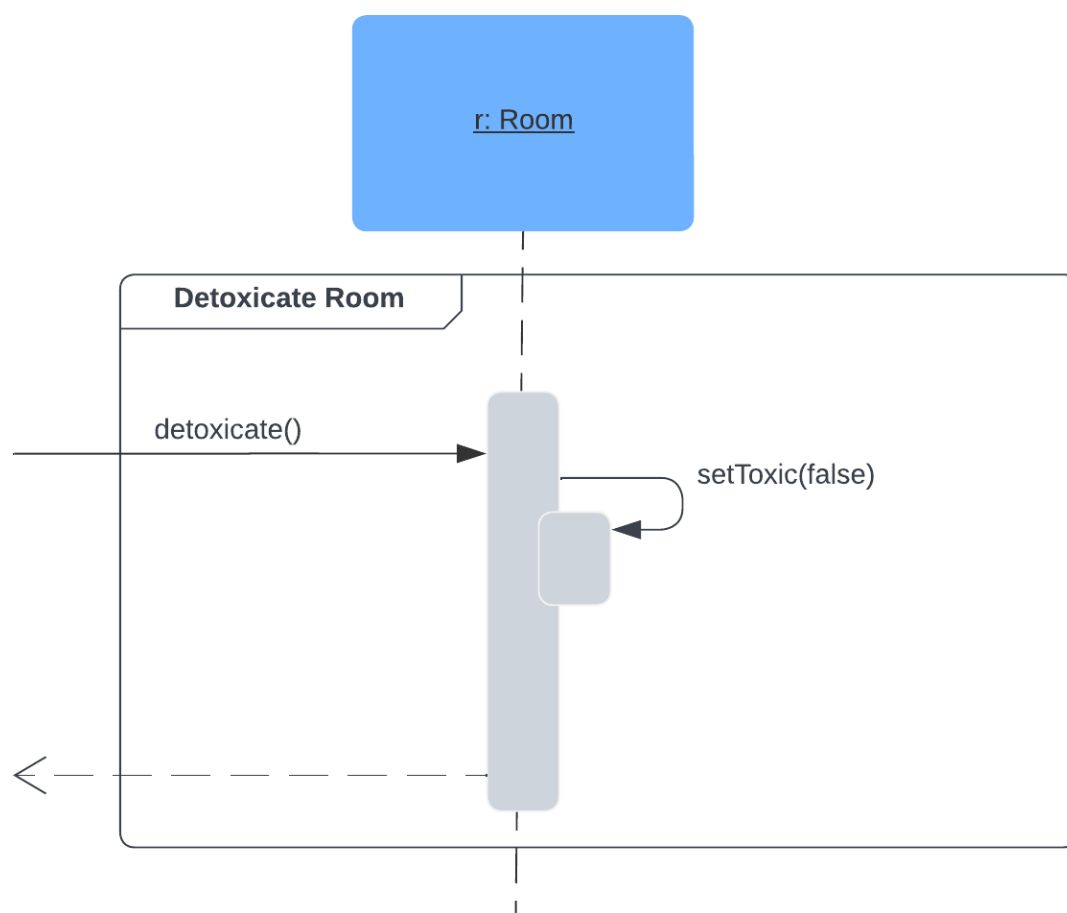
- **meet(e: Entity):** Ezen függvény felülírása, az eredeti Entity osztály függvényének rekurzióját védi ki.
- **void move(dest: Room,d: Door):** Az eredeti Entity move függvényéhez, képest a felülírásban, ha sikeres a belépés meghívódik a szobára a kitakarításhoz szükséges metódus.
- **bool toxicate():** A takarítónő esetén ezen metódus visszatérési értéke mindig hamis.
- **bool immobilize():** A takarítónő esetén ezen metódus visszatérési értéke mindig hamis.

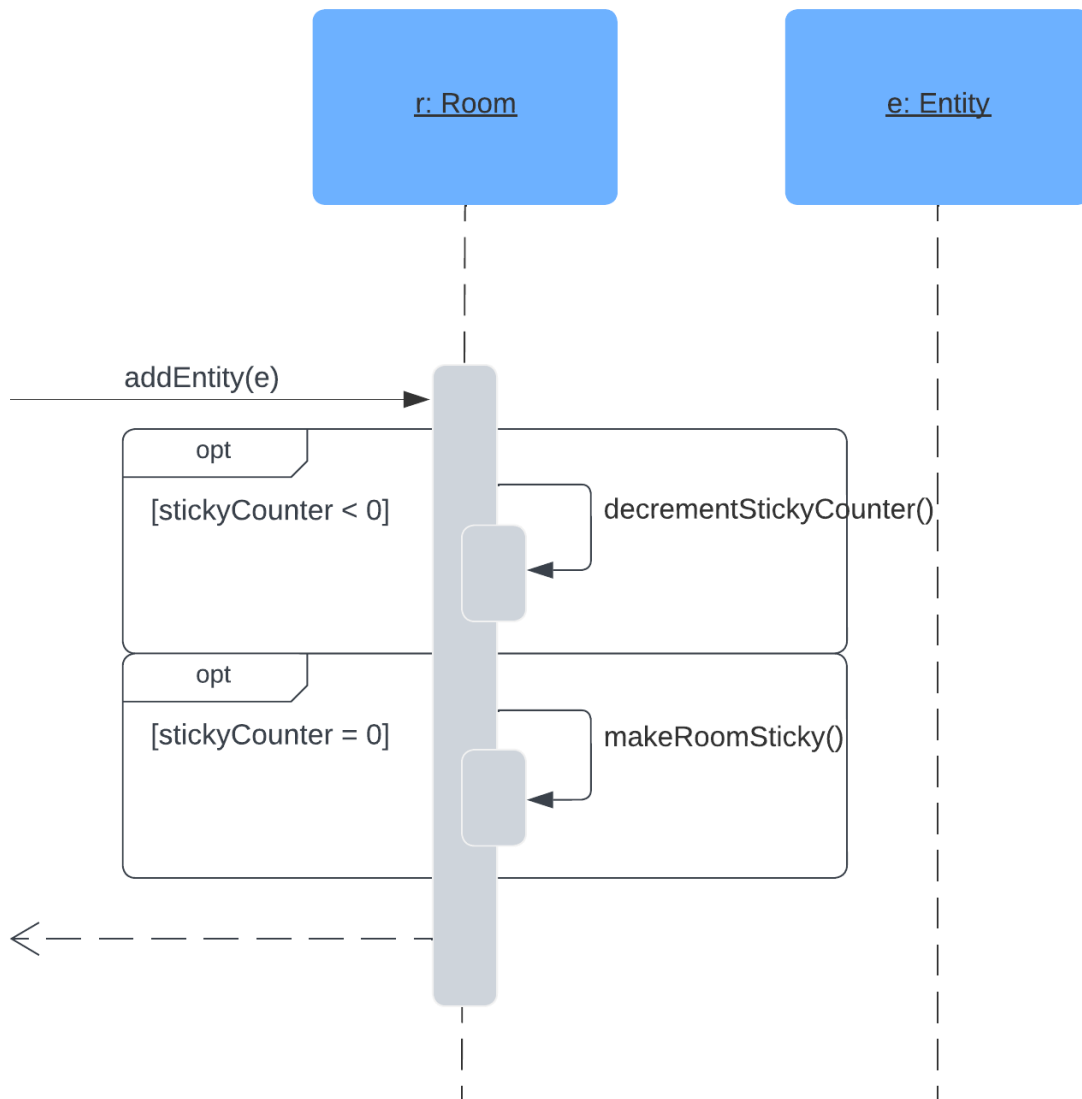
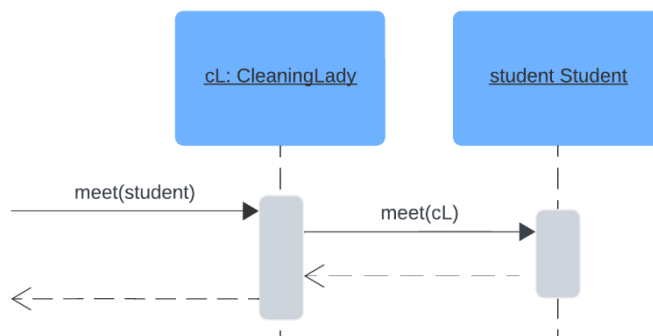
### 7.0.3 Szekvencia-diagramok

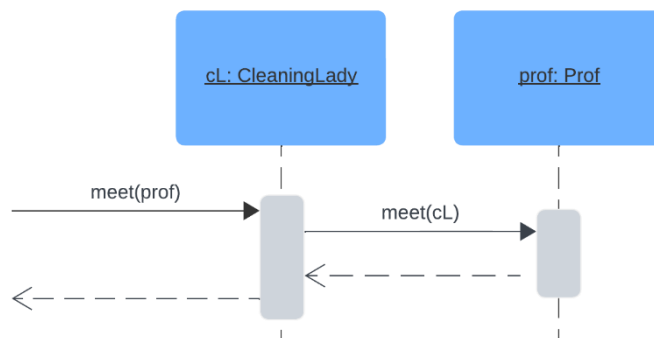
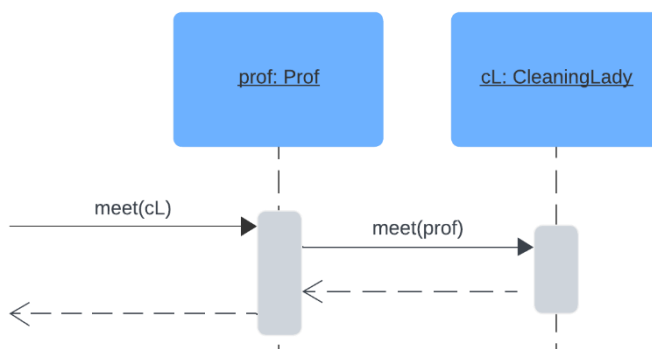
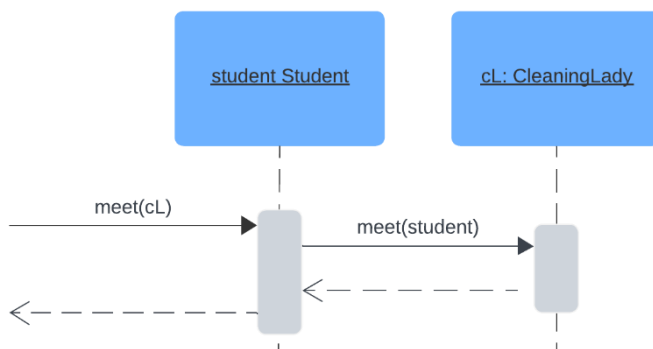
### 7.0.3.1 Léghűtő aktiválása



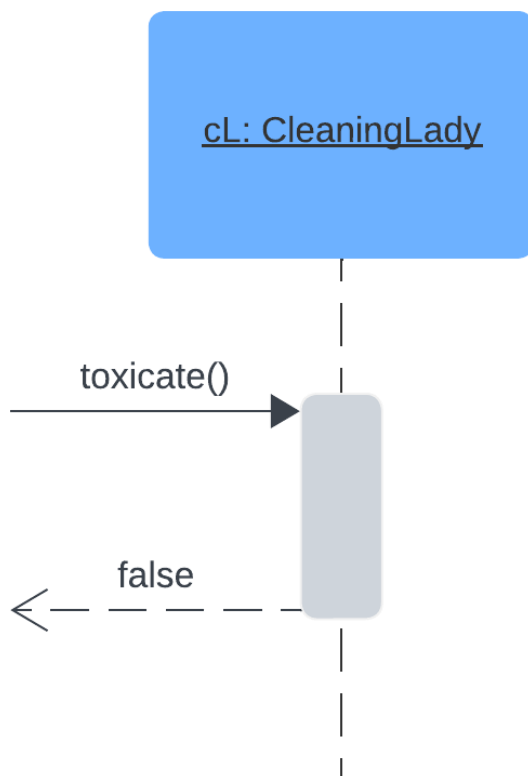
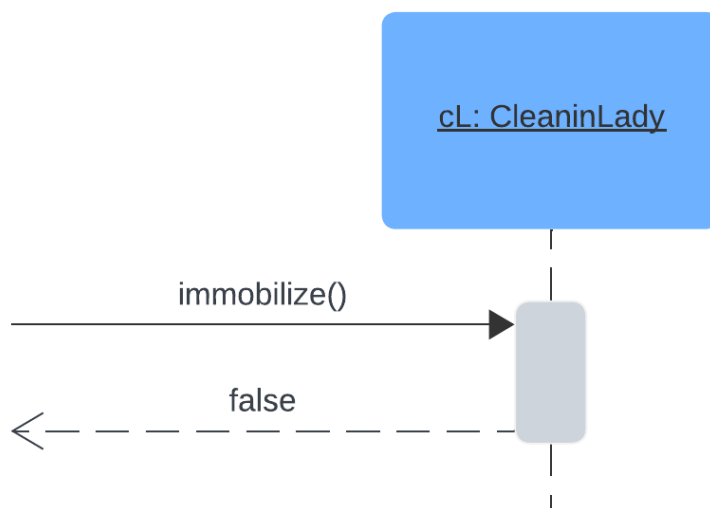
**7.0.3.2 Takarítás utáni látogatás**

**7.0.3.3 Méregző szoba méregtelenítése**

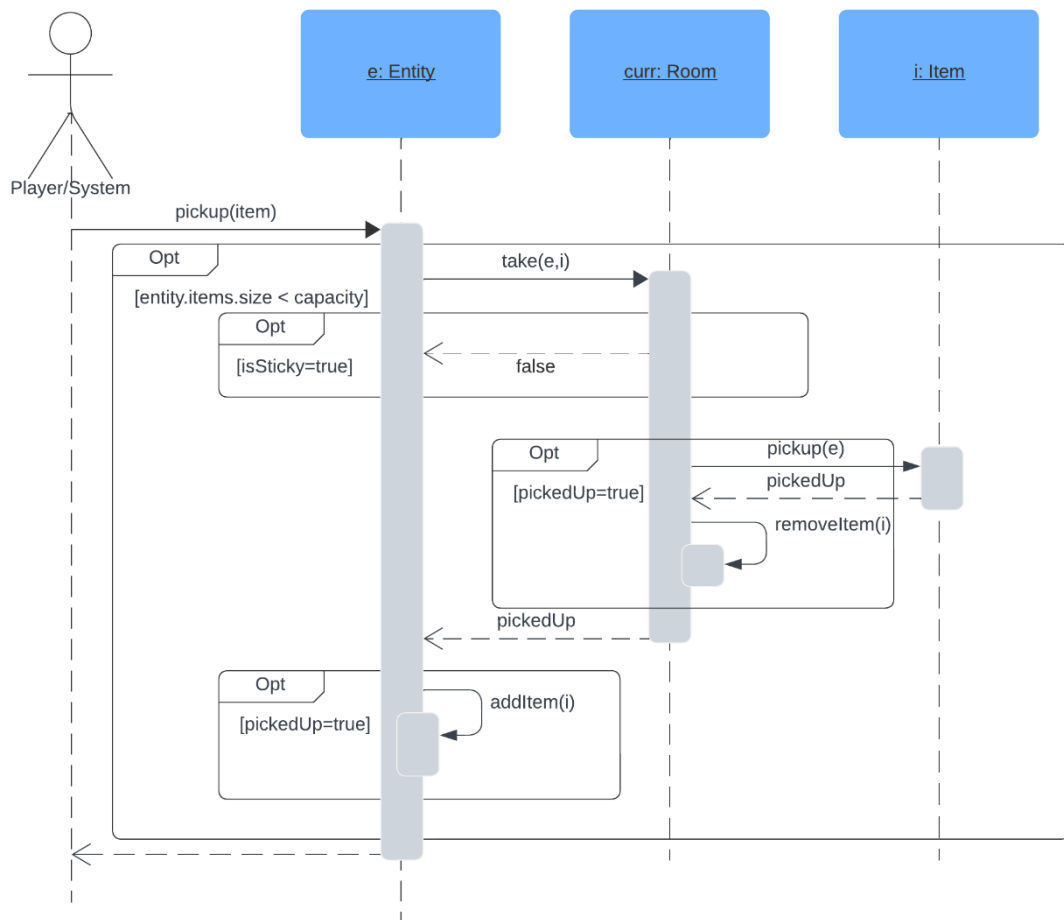
**7.0.3.4 Szoba ragadóssá válásának folyamata****7.0.3.5 Takarító Hallgatóval találkozik**

**7.0.3.6 Takarító Oktatóval találkozik****7.0.3.7 Oktató Takarítóval találkozik****7.0.3.8 Hallgató Takarítóval találkozik**

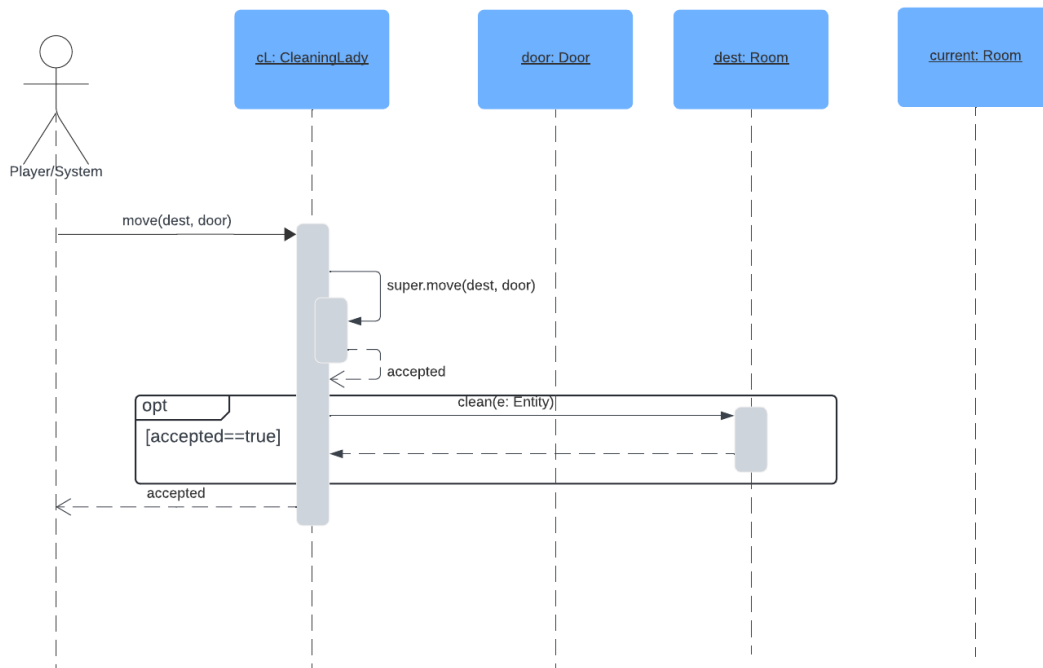


**7.0.3.9 Takarító mérgezése****7.0.3.10 Takarító mozgásképtelenítése**

## 7.0.3.11 Tárgyfelvétel



## 7.0.3.12 Takarító mozgása



## 7.1 Prototípus interface-definíciója

### 7.1.1 Az interfész általános leírása

A prototípus alkalmazást több módban is el lehet indítani attól függően, hogy játszani, pályát építeni vagy tesztelni szeretnénk. Ha a program indításánál nem adunk meg parancssori argumentumot, akkor játszani szeretnénk az alkalmazással. Ebben a módban játékállást lehet betölteni, kimenteni, elindítani, megállítani, valamint lehet parancsokkal kezelni a karaktereket. Ha az első parancssori argumentum *editor*, akkor egy világépítő alkalmazást indítunk, amelyben létre lehet hozni karaktereket, szobákat, ajtókat és tárgyakat. Ajánlott először létrehozni a szobákat, majd ezekben elhelyezni a karaktereket és tárgyakat, valamint létrehozni ajtókat. Ha *test* módban indítjuk az alkalmazást, akkor az összes tesztet lefut, amelyek előállnak egy-egy emberek számára olvasható eredménnyel (getinfo eredménye) egy szövegfájl formájában.

### 7.1.2 Bemeneti nyelv

A következő parancsokat lehet használni az egyes alkalmazásmódokban.

#### *startGame*

**Leírás:** Elindítja a játékot, ilyenkor nem lehet használni a *load* parancsot. Ahhoz, hogy el lehessen indítani a játékot követelmény, hogy betöltsünk egy játékállapotot a *load* parancssal.

**Opciók:** nincsenek

**Elérhető:** csak játékmódban

#### *pauseGame*

**Leírás:** Megállítja a játékot.

**Opciók:** nincsenek

**Elérhető:** csak játékmódban

#### *load*

**Leírás:** Betölti a játékállást a paraméterében megadott fájlból.

**Opciók:** <abszolút elérési útvonal>

**Elérhető:** játékmódban és világépítőmódban

#### *save*

**Leírás:** Kimentti a játékállást a paraméterben kapott fájlba.

**Opciók:** <abszolút elérési útvonal>

**Elérhető:** játékmódban és világépítőmódban

#### *random*

**Leírás:** Be- vagy kikapcsolja a randomicitást a paramétere szerint. Ha ki van kapcsolva a randomicitás, akkor a már szkeletonban is megszokott módon a program megkérdezi a felhasználót a program további futásáról. Ha be van kapcsolva, akkor nyilván a program véletlenszerűen viselkedik. Teszt módban mindig ki van kapcsolva a randomicitás.

**Opciók:** <randomicitás be- vagy kikapcsolva – true/false érték>

**Elérhető:** csak játékmódban

#### *getInfo*

**Leírás:** Segítségével lekérdezhető a karakterek és szobák állapota. A tárgyak állapota a tulajdonosuknál (karakter/szoba) jelenik meg. Ha az opcionális paramétere megvan

adva, akkor csak a paraméterben megkapott karakter vagy szoba állapotát írja ki. Paraméterek nélkül pedig minden karakter és minden szoba állapotát kiírja. A teszt módban ezt írja ki fájlba.

**Opciók:** [opt]<objektumazonosító – név vagy azonosító>

**Elérhető:** játékmódban és világépítőmódban

#### *list*

**Leírás:** Kilistázza a paraméterében lekérdezett objektumok azonosítóit.

**Opciók:** <entities/rooms - listázandó objektumok>

**Elérhető:** játékmódban és világépítőmódban

#### **A következő parancsok az éppen körét játszó játékos karakterével hajtódnak végre**

##### *move*

**Leírás:** Megpróbálja átmozgatni a játékost a paraméterében megkapott szobába a szintén paraméterében megkapott ajtón keresztül. Ha már egyszer megpróbált mozogni a játékos, akkor abban a körben többet már nem mozoghat.

**Opciók:** <szoba azonosítója> <ajtó azonosítója>

**Elérhető:** csak játékmódban

##### *pickup*

**Leírás:** Megpróbálja felvenni a paraméterként kapott azonosítóval rendelkező tárgyat.

**Opciók:** <tárgy azonosítója>

**Elérhető:** csak játékmódban

##### *activate*

**Leírás:** Megpróbálja aktiválni a paraméterként kapott azonosítóval rendelkező tárgyat.

**Opciók:** <tárgy azonosítója>

**Elérhető:** csak játékmódban

##### *drop*

**Leírás:** Leteszi a paraméterként kapott azonosítóval rendelkező tárgyat.

**Opciók:** <tárgy azonosítója>

**Elérhető:** csak játékmódban

##### *endTurn*

**Leírás:** Befejezi a játékos körét.

**Opciók:** nincsenek

**Elérhető:** csak játékmódban

#### **A következő parancsok a világépítésben segítenek.**

##### *addPlayer*

**Leírás:** Hozzáad a játékállapothoz egy játékost (hallgatót). A paraméterben megadott név lesz az objektum azonosítója. Két azonos név nem lehetséges.

**Opciók:** <név>

**Elérhető:** csak világépítő módban

##### *addNPC*

**Leírás:** Hozzáad a játékállapothoz egy nemjátszható karaktert (oktató vagy takarító). A paraméterben át kell adni a karakter típusát, valamint a szobát, amiben el szeretnénk helyezni. Azonosítója automatikusan generálódik.

**Opciók:** <nemjátszható karakter típusa – Prof, CleaningLady> <szoba azonosítója>

**Elérhető:** csak világépítő módban

### *addRoom*

**Leírás:** Hozzáad a játékállapothoz egy szobát. Egy kötelező paramétere van, a szoba kapacitása. Opcionális paraméterei vannak, amelyek megadják, hogy mérgező-e, valamint elátkozott-e. Azonosítója automatikusan generálódik.

**Opciók:** <szoba kapacitása> [opt]<szoba állapota>...

**Példa:** addRoom toxic cursed

**Elérhető:** csak világépítő módban

### *addDoor*

**Leírás:** Hozzáad a játékállapothoz egy ajtót a két különböző paraméterű szoba és az opcionális, egyirányúságot jelző utolsó paramétere alapján. Ha be van állítva a oneway flag, az azt jelenti, hogy az ajtó az első szobából vezet a másodikba, és fordítva nem. Itt a kapott szobákhoz megtörténik az ajtó hozzáadása. Azonosítója automatikusan generálódik.

**Opciók:** <első szoba azonosítója> <második szoba azonosítója> [opt]<oneway>

**Elérhető:** csak világépítő módban

### *addItem*

**Leírás:** Hozzáad a játékállapothoz egy tárgyat. Az első paramétere megadja a tárgy típusát, a második paramétere megadja a szobát, ahol el lesz helyezve a tárgy. Logarlécet nem lehet hozzáadni. Azonosítója automatikusan generálódik.

**Opciók:** <tárgy típusa – beer, camembert, ...> <szoba azonosítója>

**Elérhető:** csak világépítő módban

### *placeLogar*

**Leírás:** Elhelyezi a játékállapotban megtalálható egyetlen Logarlécet egy szobába. Azonosítója automatikusan generálódik.

**Opciók:** <szoba azonosítója>

**Elérhető:** csak világépítő módban

A teszt módban történő indítás során az összes tesztet egymás után sorban lefut és előállnak a tesztesetekhez tartozó eredményfájlok, amelyeket a továbbiakban a felhasználó feladata összehasonlítani az elvárt eredménnyel.

### 7.1.3 Kimeneti nyelv

#### *getinfo*

#### 1. Karakter állapotának kimenete:

ID: <Karakter azonosítója>

Stunned for: <még ennyi ideig bénult az entitás>

[opt](Drunk for: <még ennyi ideig részeg a hallgató>)

Items of entity:

**[szakasz ismétlése: 0..5]**

ID: <tárgy azonosítója>

Health: <a tárgy életpontjai - szokásos értékek: -1 (végtelen), 1 (egyszerhasználatos tárgy), 3 (egyéb)>

Pair: <tárgy párjának azonosítója>

[opt](Active: <true/false>)

[opt](Room: <szoba azonosítója>)

**[ismétlés vége]**

#### 2. Szoba állapotának kimenete:

ID: <Szoba azonosítója>

Capacity: <szoba kapacitása>

Toxic: <true/false>

Wet: <true/false>

Cursed: <true/false>

Sticky: <true/false>

Entities of room:

**[szakasz ismétlése: 0..Capacity]**

ID: <entitás azonosítója>

**[ismétlés vége]**

Items of room:

**[szakasz ismétlése: 0..\*]**

ID: <tárgy azonosítója>

Health: <a tárgy életpontjai - szokásos értékek: -1 (végtelen), 1 (egyszerhasználatos tárgy), 3 (egyéb)>

Pair: <tárgy párjának azonosítója>

[opt](Active: <true/false>)

[opt](Room: <szoba azonosítója>)

**[ismétlés vége]**

Doors of room:

**[szakasz ismétlése 0..\*]**

ID: <ajtó azonosítója>

Room #1: <szoba azonosítója>

Room #2: <szoba azonosítója>

Oneway: <true/false>

**[ismétlés vége]**

#### *list*

#### 1. Játék összes karakterének listázása

**[szakasz ismétlése: 0..\*]**

ID: <entitás azonosítója>

**[ismétlés vége]**

**2. Játék összes szobájának listázása**

[szakasz ismétlése: 1..\*]

ID: &lt;szoba azonosítója&gt;

[ismétlés vége]

Ha minden jól lement a többi parancs meghívása után, akkor a Linux rendszerekhez hasonlóan nem ír ki semmit a program. Ha valami nem ment jól, akkor valamilyen exception-nel fog visszatérni a programkód.

**7.2 Összes részletes use-case**

Use-case neve	startGame
Rövid leírás	Egy játék indítása.
Aktorok	Tesztelő
Forgatókönyv	Elindítja a játékot. Ezután nem lehet a <i>load</i> parancsot használni. Előkövetelmény a játék elindítása előtt, hogy be legyen töltve egy játékállapot a <i>load</i> paranccsal.

Use-case neve	pauseGame
Rövid leírás	A játék megállítása.
Aktorok	Tesztelő
Forgatókönyv	A játék megáll.

Use-case neve	load
Rövid leírás	Egy előzőleg mentett játékállást betölt.
Aktorok	Tesztelő
Forgatókönyv	A játékállás betöltődik a paraméterben megadott fájlból.

Use-case neve	save
Rövid leírás	Elmenti a jelenlegi játékállást egy fájlba.
Aktorok	Tesztelő
Forgatókönyv	Egy fájlba kiment a jelenlegi játékállást.

Use-case neve	random
Rövid leírás	A játék randomicitásának beállítása.
Aktorok	Tesztelő
Forgatókönyv	A paraméter szerint be- vagy kikapcsolja a randomicitást. Kikapcsolt állapotban a program a felhasználótól fogja megkérdezni a program a további futásáról. Bekapcsolt állapotban a program véletlenszerűen viselkedik.

Use-case neve	getinfo
Rövid leírás	Karakterek és szobák állapotának lekérdezése.
Aktorok	Tesztelő
Forgatókönyv	A szobák és karakterek állapotának lekérdezése megtörténik kimenetre. Opcionális paraméter esetén az ott megadott szoba vagy karakter állapotát írja csak ki.

<b>Use-case neve</b>	list
<b>Rövid leírás</b>	Lekérdezett objektumok azonosítóit listázza.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	A paraméterben megadott objektumok (karakterek, szobák) azonosítói kiíródnak a kimenetre.

<b>Use-case neve</b>	move
<b>Rövid leírás</b>	A karakter mozgatása.
<b>Aktorok</b>	Tesztelő / Game
<b>Forgatókönyv</b>	Megpróbálja a karaktert átmozgatni egy, a paraméterben megadott szobába átmozgatni. Ha sikerül, akkor át is kerül a másik szobába. Ha nem, akkor marad a szobájában a karakter. Egy mozgatási kísérlet után mozoghat a játékos.

<b>Use-case neve</b>	pickup
<b>Rövid leírás</b>	A karakter felvesz egy tárgyat.
<b>Aktorok</b>	Tesztelő / Game
<b>Forgatókönyv</b>	Megpróbálja felvenni a tárgyat a paraméterben kapott tárgyat a karakter. Ha sikerül, akkor bekerül a raktárjába a karakternek a tárgy. Ha nem, akkor marad a szoba raktárjában a tárgy.

<b>Use-case neve</b>	activate
<b>Rövid leírás</b>	A karakter aktivál egy tárgyat.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	Megpróbálja a Hallgató a paraméterben kapott tárgyat. Ha a raktárjában és használható állapotban van a tárgy, akkor a hatása teljesül. Ha bármelyik nem igaz, akkor nem történik semmi.

<b>Use-case neve</b>	drop
<b>Rövid leírás</b>	A karakter eldob egy tárgyat.
<b>Aktorok</b>	Tesztelő / Game
<b>Forgatókönyv</b>	A paraméterben kapott tárgyat eldobja a karakter abba a szobába, amelyben a karakter jelenleg van. Ekkor a karakter raktárjából átkerül a szoba raktárjába a tárgy.

<b>Use-case neve</b>	endturn
<b>Rövid leírás</b>	A karakter befejezi a körét.
<b>Aktorok</b>	Tesztelő / Game
<b>Forgatókönyv</b>	A karakternek a köre befejeződik.



<b>Use-case neve</b>	addPlayer
<b>Rövid leírás</b>	Hozzáad a játékhoz egy Hallgatót.
<b>Aktorok</b>	Tesztelő
<b>Forgatókönyv</b>	A játékállapotban egy új játszható karakter kerül hozzáadásra. A paraméterben megadott név lesz az azonosítója, ami egyedi.

<b>Use-case neve</b>	addNPC
<b>Rövid leírás</b>	Hozzáad a játékhoz egy nem játszható karaktert (Oktató vagy Takarítót)
<b>Aktorok</b>	Tesztelő / Game
<b>Forgatókönyv</b>	Hozzáad a játékállapothoz egy nem játszható karaktert. A paraméterben megadott típus szerint kerül létrehozásra a karakter a paraméterben megadott szobába. Ennél a karakternél az azonosító automatikusan generálódik.

<b>Use-case neve</b>	addRoom
<b>Rövid leírás</b>	Hozzáad a játékhoz egy szobát.
<b>Aktorok</b>	Tesztelő / Game
<b>Forgatókönyv</b>	A játékállapothoz hozzáadódik egy szobát. Paraméterében megadott kapacitással, és esetleges tulajdonsággal kerül létrehozásra ekkor. Azonosítója automatikusan generálódik.

<b>Use-case neve</b>	addDoor
<b>Rövid leírás</b>	Hozzáad a játékhoz egy ajtót.
<b>Aktorok</b>	Tesztelő / Game
<b>Forgatókönyv</b>	A játékállapothoz hozzáadódik egy új ajtót. Ekkor a paraméterben megadott két szoba között jön létre, illetve további paraméterben megadott jelző alapján egyirányúként is létrejöhet. Azonosítója automatikusan generálódik.

<b>Use-case neve</b>	addItem
<b>Rövid leírás</b>	A szobának a raktárjába ad egy új tárgyat.
<b>Aktorok</b>	Tesztelő / Game
<b>Forgatókönyv</b>	A játékállapophoz hozzáadódik egy új tárgy. A paraméterben megadott típus alapján a megadott szobába kerül létrehozásra a tárgy. Logarlécet nem lehet létrehozni. Azonosító automatikusan generálódik.

Use-case neve	placeLogar
Rövid leírás	Egy szobába lerak egy Logarlécet.
Aktorok	Tesztelő / Game
Forgatókönyv	A játékállapotba elhelyezi az egyetlen Logarlécet egy szobába. Az azonosító automatikusan generálódik.

### 7.3 Tesztelési terv

Teszt-eset neve	<b>Hallgató szomszédos szobába lép</b>
Rövid leírás	Hallgató mozgatása.
Teszt célja	A teszt célja az entitások mozgási képességének ellenőrzése.

Teszt-eset neve	<b>Hallgató nem szomszédos szobába lép</b>
Rövid leírás	Hallgató mozgatása.
Teszt célja	Teszteli, hogy a Hallgató nem tud egy nem szomszédos szobába továbblépni.

Teszt-eset neve	<b>Hallgató Sört vesz fel</b>
Rövid leírás	Hallgató felvesz egy sört a szobában.
Teszt célja	Teszteli, hogy a Hallgató képes tárgyat felvenni, amikor a tárolójában van még üres hely.

Teszt-eset neve	<b>Hallgató teli raktárral vesz fel Tranzisztort</b>
Rövid leírás	Hallgató megpróbál felvenni egy Tranzisztort tárgyat teli raktárral.
Teszt célja	Teszteli azt, hogy teli raktárral a Hallgató nem tud tárgyat felvenni.

Teszt-eset neve	<b>Hallgató felveszi a Logarlécet</b>
Rövid leírás	Hallgató felveszi a Logarléc tárgyat.
Teszt célja	Teszteli azt, hogy amikor a hallgató felveszi Logarlécet, akkor a játéknak vége lesz és a hallgatók nyertek.

Teszt-eset neve	<b>Hallgató ragacsos szobában felvesz egy Rongyot</b>
Rövid leírás	Hallgató ragacsos szobában próbál felvenni egy Rongy tárgyat.
Teszt célja	Teszteli azt, hogy a hallgató nem tud tárgyat felvenni a ragacsos szobában.

Teszt-eset neve	<b>Hallgató Tranzisztorokat párosít</b>
Rövid leírás	Hallgató két, már raktárban lévő, Tranzisztor tárgyat összepárosít.
Teszt célja	A teszt célja, hogy a Hallgató tud Tranzisztorokat párosítani, amik a raktárjában vannak.

Teszt-eset neve	<b>Hallgató eldob egy Légfrissítőt</b>
Rövid leírás	Hallgató raktárából eldob egy Légfrissítő tárgyat.

<b>Teszt célja</b>	Teszteli, hogy a Hallgató képes eldobni tárgyat a saját raktárjából.
--------------------	--

<b>Teszt-eset neve</b>	<b>Hallgató Légfrissítőt használ</b>
<b>Rövid leírás</b>	Hallgató gázos szobába lépve elhasznál egy Légfrissítő tárgyat.
<b>Teszt célja</b>	A teszt célja, a Légfrissítő tárgy használatának ellenőrzése. A Hallgató gázos szobában, elhasználva ezt a tárgyat, képes kitisztítani a szobát, szóval a szoba nem lesz gázos többé.

<b>Teszt-eset neve</b>	<b>Hallgató megiszik egy Sört</b>
<b>Rövid leírás</b>	Hallgató elhasznál egy Sör tárgyat.
<b>Teszt célja</b>	A teszt célja a Sör tárgy használatának ellenőrzése, illetve hogy a Hallgató sör hatása alá kerül.

<b>Teszt-eset neve</b>	<b>Hallgató teleportál</b>
<b>Rövid leírás</b>	Hallgató elteleportál egyik szobából a másikba Tranzisztorok segítségével.
<b>Teszt célja</b>	A teszt célja az, hogy az előre párosított, illetve másik szobában lévő tranzisztorra teleportálja a Hallgatót a nála lévő tranzisztor használatakor.

<b>Teszt-eset neve</b>	<b>Hallgató elhasznál egy Camembert</b>
<b>Rövid leírás</b>	Hallgató használ egy Camembert tárgyat.
<b>Teszt célja</b>	Teszteli a Camembert tárgy használatát és annak következményét. Camembert elhasználva a szoba gázos lesz.

<b>Teszt-eset neve</b>	<b>Hallgató felvesz egy hamis Logarléce</b>
<b>Rövid leírás</b>	Hallgató felvesz egy hamis Logarléc tárgyat.
<b>Teszt célja</b>	Teszteli, hogy a hamis tárgyat is fel tud venni a Hallgató.

<b>Teszt-eset neve</b>	<b>Hallgató elhasznál egy hamis TVSZ-t</b>
<b>Rövid leírás</b>	Hallgató elhasznál egy hamis TVSZ tárgyat.
<b>Teszt célja</b>	A teszt célja a hamis tárgy elhasználásának bemutatása. Hallgató hamis TVSZ-t próbál használni, aminek nincs hatása.

<b>Teszt-eset neve</b>	<b>Hallgató elhasznál egy Törlőrongyot</b>
<b>Rövid leírás</b>	Hallgató elhasznál egy Törlőrongy tárgyat.
<b>Teszt célja</b>	Teszteli a Törlőrongy használatát. Ennek használata után az Oktató megbénul szobába lépéskor.

<b>Teszt-eset neve</b>	<b>Hallgató már aktív sör hatással elhasznál egy sört</b>
<b>Rövid leírás</b>	Hallgató már Sör hatása alatt elhasznál egy Sör tárgyat.
<b>Teszt célja</b>	Teszteli, hogy a már Sör hatása alatt lévő Hallgató sört használva elejt egy tárgyat a raktárából.

<b>Teszt-eset neve</b>	<b>Hallgató belép gázos szobába</b>
<b>Rövid leírás</b>	Hallgató gázos szobába lép.
<b>Teszt célja</b>	Teszteli a gázos szoba hatásait a Hallgatóra.

<b>Teszt-eset neve</b>	<b>Hallgató belép gázos szobába ffp2-s maszkkal</b>
<b>Rövid leírás</b>	Hallgató gázos szobába lép FFP2 maszk tárggyal.
<b>Teszt célja</b>	A teszt célja, hogy ellenőrizze a Hallgató gázos szobába lépésekor ffp2-es maszk működését, ezzel kivédve a gázos szoba hatásait a Hallgatóra.

<b>Teszt-eset neve</b>	<b>Oktató megtámadja a Hallgatót</b>
<b>Rövid leírás</b>	Oktató megtámadja a Hallgatót.
<b>Teszt célja</b>	Teszteli, amikor az Oktató megtámadja a védtelen Hallgatót, aki a támadás hatására kibukik.

<b>Teszt-eset neve</b>	<b>Oktató megtámadja Sör hatása alatt lévő Hallgatót</b>
<b>Rövid leírás</b>	Oktató megtámadja a Hallgatót, aki épp Sör hatása alatt van.
<b>Teszt célja</b>	Teszteli a Sör hatás védelmét a Hallgatóra, amikor azt megtámadja egy Oktató.

<b>Teszt-eset neve</b>	<b>Oktató megtámadja TVSZ-szel rendelkező Hallgatót</b>
<b>Rövid leírás</b>	Oktató megtámadja a Hallgatót, akinek van TVSZ tárgya.
<b>Teszt célja</b>	Teszteli a TVSZ működését, hogy az Oktató megtámad egy Hallgatót, akinél megtalálható ez a tárgy.

<b>Teszt-eset neve</b>	<b>Oktató megtámadja hamis TVSZ-szel rendelkező Hallgatót</b>
<b>Rövid leírás</b>	Oktató megtámadja a hamis TVSZ tárggyal rendelkező Hallgatót.
<b>Teszt célja</b>	A teszt célja a hamis TVSZ használhatatlanságát bemutatni, azzal, hogy az Oktató megtámad egy Hallgatót, akinél ez a tárgy megtalálható.

<b>Teszt-eset neve</b>	<b>Takarító belép egy szobába amiben van Hallgatót</b>
<b>Rövid leírás</b>	Takarító belép egy szobába, ahol már van egy Hallgató.
<b>Teszt célja</b>	Teszteli a Takarító entitás képességét, azzal, hogy a Takarító szobába lépésekor a Hallgató átkerül egy másik szobába.

<b>Teszt-eset neve</b>	<b>Takarító gázos szobába lép</b>
<b>Rövid leírás</b>	Takarító gázos szobába lép.
<b>Teszt célja</b>	Teszteli azt, hogy a Takarító ekkor a gázos szobát kiszellőzteti és nem lesz már gázos szoba.

<b>Teszt-eset neve</b>	<b>Két szoba egyesül</b>
<b>Rövid leírás</b>	Két szoba egyesül.
<b>Teszt célja</b>	A teszt célja, hogy a két szoba egyesülésének sikerességét ellenőrizze.

<b>Teszt-eset neve</b>	<b>Egy szoba szétválík</b>
<b>Rövid leírás</b>	Egy szoba szétválík.
<b>Teszt célja</b>	A teszt célja, hogy egy szoba szétválásának sikerességét ellenőrizze.

<b>Teszt-eset neve</b>	<b>Gázos és elátkozott szoba egyesül</b>
<b>Rövid leírás</b>	Két különböző tulajdonságú, egy gázos és egy elátkozott, szoba egyesül.
<b>Teszt célja</b>	Teszteli, hogy megfelelően egyesül két különböző tulajdonságú szoba, egy gázos és egy elátkozott szoba.

<b>Teszt-eset neve</b>	<b>Szoba ragacsossá válik</b>
<b>Rövid leírás</b>	Szoba ragacsossá válik miután kitakarították.
<b>Teszt célja</b>	Teszteli, hogy a egy takarított szoba, megfelelő mennyiségű áthaladott entitás után ragacsossá válik.

<b>Teszt-eset neve</b>	<b>Minden Hallgató kibukik</b>
<b>Rövid leírás</b>	Minden Hallgató kibukik és elveszítik a játékot.
<b>Teszt célja</b>	A teszt célja, hogy ellenőrizze azt az állapotot, amikor minden Hallgató kibukott és nem teljesítették a küldetést, és a játék véget ér a Hallgatók vesztésével.

#### 7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A programnak van *test* módja. Ebben a módban az összes tesztet automatikusan lefut és előáll minden tesztetthez egy eredményfájl. A tesztekhez az elvárt eredményt is megadjuk. Az előre definiált és a kapott szövegfájlokat a felhasználó más alkalmazással tudja majd összehasonlítani. Pl. Windows alatt az *fc* parancssori alkalmazás jól használható erre.

## 7.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.04.05. 17:00	2 óra	Nagy Papp Sánta	Feladatok elosztása, osztálydiagram tervezés
2024.04.05. 19:00	2 óra	Papp	Prototípus interfészének definiálása
2024.04.06. 10:00	1 óra	Nagy Sánta	Osztálydiagram tervezés, metódusok tervezése
2024.04.06. 18:30	3 óra	Nagy Sánta	Szekvencia-diagramok készítése, hibajavítások
2024.04.06. 20:00	2 óra	Gulybán	Use-case esetek megtervezése és megírása.
2024.04.07. 11:00	1 óra	Nagy	Új funkciókhoz szekvencia készítés és hiba javítás.
2024.04.07. 11:00	2 óra	Csabuda	Tesztek összegyűjtése és tesztelése terv írása.