

7. Prototípus koncepciója

68 – 00ff00

Konzulens:

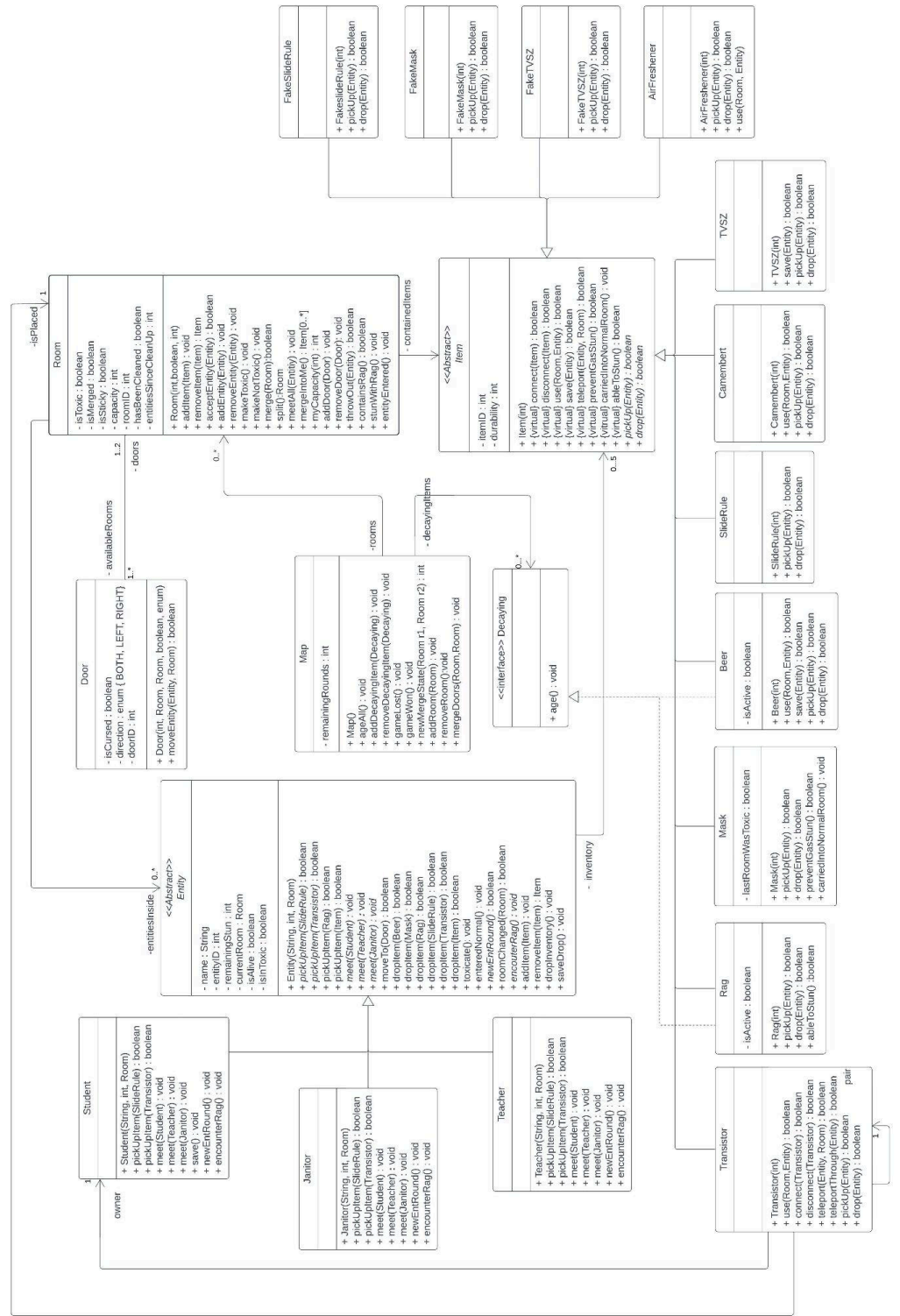
Szabó Bence Sándor

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 | miczkimark@gmail.com |
| Nagy László | BC7TB3 | nagy.laszlo1130@gmail.com |

7. Prototípus koncepciója

7.0.1 Módosult osztálydiagram



7.0.2 Új vagy megváltozó metódusok/függvények

7.0.2.1 Új metódusok/függvények

- **[Item] boolean ableToStun()** : A tárgy bénítási képességének logikai értékével tér vissza. Csak a rongy tárgynál van felül definiálva. Alap esetben "false" értéket ad vissza, hogyha nem "Rag" tárgy. Ha "Rag" tárgy ableToStun() függvénye hívódik meg, akkor megnézi, hogy az aktiválva van-e (`isActive == true`). Hogyha nincsen aktiválva, akkor szintén "false" értéket ad vissza. Ha aktiválva van, akkor megnézi, hogy a tárgynak van-e még tartóssága (`durability >= 0`). Hogyha a tartóssága nulla, akkor megint "false" értékkel tér vissza. Hogyha a tartóssága nagyobb mint nulla, és aktiválva is van, akkor "true" értékkel tér vissza.
- **[Room] boolean containsRag()** : "true" értéket ad vissza, ha szobában van aktív rongy. Végig hívja az összes szobában lévő tárgy ableToStun() függvényét ha valamelyik "true" értékkel tér vissza akkor maga a függvény is.
- **[Room] void stunWithRag():** Rongyal való bénítást megvalósító metódus. Amennyiben a containsRag() "true" értékkel tér (ellenkező esetben nem történik semmi) vissza a szobában lévő összes Entity-n végig hívja az encounterRag() függvényt.
- **[Room] void makeNotToxic()** : Átalakítja a szobát nem mérgezővé (`isToxic = false`), ezt a metódust használja a AirFreshener valamint Janitor osztály is.
- **[Room] boolean throwOut(Entity ent)** : Amennyiben lehetséges ki rakja a referált Entity-t a szobából, ha nem hamis értékkel tér vissza
- **[Entity] void saveDrop():** Új metódus a sör használat megváltozott követelményeinek teljesítésére. Meghívásakor az Entity hátizsákjából az utolsó tárgytól indulva az első olyan ami eldobható, eldobásra kerül. Ha hátizsákja üres vagy nincs nála olyan tárgy, ami eldobható lenne, a lefutás eredmény nélkül zajlik.
- **[Entity] canPickUp():** Ellenőrzi, hogy az Entity fel tud-e venni egy tárgyat a hátizsákjába, azaz kevesebb mint 5 tárgy van benne a metódushívás pillanatában. Visszatér a vizsgálat logikai értékével.
- **[Item] boolean preventGasStun():** 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.
- **[Mask] boolean preventGasStun():** A gáz által okozott bénítás kivédését végzi. Ellenőrzi, hogy a maszk tartalmaz-e még felhasználható töltetet. Ha igen a lastRoomWasToxic értékétől függően azt vizsgálja meg, hogy ez a fennmaradó töltet elegendő-e a bénítás kivédésére. Ugyanis, ha az előző szoba gázos volt (`lastRoomWasToxic == true`) akkor 2 töltet szükséges ehhez, egyébként 1. Ha van elegendő, a szükséges mennyiséget levonja, beállítja a lastRoomWasToxic értéket igazra és visszatér „true”-val. Ha nincs elegendő, a töltet szám nullázódik és a visszatérési érték „false”.

- **[Item] void carriedIntoNormalRoom():** Egy tárgy normál szobába bevitelével járó állapotváltozásokat kezeli.
- **[Mask] void carriedIntoNormalRoom():** Egy maszk tárgy normál szobába való bevitelével járó állapotváltozását kezeli, azaz beállítja a lastRoomWasToxic tagváltozó értékét „false” értékűre.
- **[Room] boolean acceptEntity(Janitor):** Janitor szobába lépésének következményeit kezeli. Elsősorban ellenőrzi, hogy a takarító befér-e a szobába. Ha igen, a szobában lévő összes Entity-re meghívódik a szoba „throwOut()” metódusa, mellyel kitessekeli őket. Ezt követően lezajlanak a takarító szoba váltásával kapcsolatos teendők. Végül a szobában megtörténik a takarítás, azaz a „setHasBeenCleaned” igaz értékűre állítódik, valamint a „entitiesSinceCleanup” értéke 0-ra változik. Meghívásra kerül a szoba „makeNotToxic” metódusa, amely a szobában lévő esetleges gáz kiszellőztetéséért felel.
- **[Room] entityEntered():**Új metódus a ragacsos padló kezelésére. 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). Ezt követően ha az így megnövelt érték legalább öt, a szoba padlóját ragacsossá állítja (az „isSticky” értékét „true”-ra állítja).

7.0.2.2 Megváltozott metódusok/függvények

- **Transistor.connect()** : A függvény először ellenőrzi, hogy van-e párja a tranzisztornak, amin meghívódott. Ha van, akkor hamissal tér vissza, ellenkező esetben pedig beállítja párjaként a paraméterként kapott tranzisztort, ownerként az adott entitást, majd meghívja rajta is a connect() függvényt. Ott is végbemegy a párosítás, majd amikor az is visszahívja a metódust az eredeti tranzisztoron, akkor annak már lesz párja, ezáltal hamissal tér vissza a függvény.
- **Transistor.disconnect()** : A függvény először ellenőrzi, hogy a paraméterként kapott másik tranzistor-e a párja a tranzisztornak, amelyen meg lett hívva. Ha nem, akkor hamissal tér vissza. Ebbe az esetben tartozik az is, ha nincs párja. Ha viszont az a párja, akkor mindkét tranzistor pair és owner attribútumát null-ra állítja, és igazzal tér vissza.
- **Item.save(Entity)**: A metódus megváltoztatásra került annak érdekében, hogy támogatni tudja az új követelményt, miszerint a hallgató egy aktív söröskorsót felhasználva lelke megmentésére olyan mellékhatással jár, hogy egy nála lévő tárgyat elejt. Ehhez a metódus kiegészült egy Entity argumentummal mely azt az Entity-t adja át a tárgy számára, aki éppen használja azt.
- **TVSZ.save(Entity)**: A metódus megváltoztatásra került annak érdekében, hogy támogatni tudja az új követelményt a sör lélekmentésével kapcsolatban. . Ehhez a metódus kiegészült egy Entity argumentummal mely azt az Entity-t adja át a tárgy számára, aki éppen használja azt. A metódus belső működésében változtatás nem történt.
- **Beer.save(Entity)**: A metódus megváltoztatásra került annak érdekében, hogy támogatni tudja az új követelményt a sör lélekmentésével kapcsolatban. . Ehhez a metódus kiegészült egy Entity argumentummal mely azt az Entity-t adja át a tárgy számára, aki éppen használja azt. Meghívásakor ellenőrzi, hogy a tárgy durability valamint isActive attribútuma együttesen igaz értékű-e. Ha igen, meghívja a paraméterül kapott Entity saveDrop() függvényét és visszatér „true”-val. Ellenkező esetben a visszatérési érték „false”.
- **Student.pickUpItem(Sliderule), Student.pickUpItem(Transistor)...** (minden Item-re külön verzió, mindegyikben megtörtént ez a változás): A függvény először lefuttatja a szobára a removeItem() függvényt. Ha ennek a visszatérési értéke null, akkor hamissal tér vissza, egyéb esetben megtörténik a felvétel.
- **Entity.enteredNormal()**: Entity egy normál szobába belépésével járó állapotváltozásokat kezeli. Beállítja az „isInToxic” tagváltozó értékét hamisra, valamint az összes hátizsákjában lévő tárgyra meghívja a „carriedIntoNormalRoom()” metódust.
- **Entity.toxicate()**: 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 a „remainingStun” értékét 3-ra, valamint az Entity hátizsákjából az összes tárgyat eldobja a “dropInventory()” meghívásával.

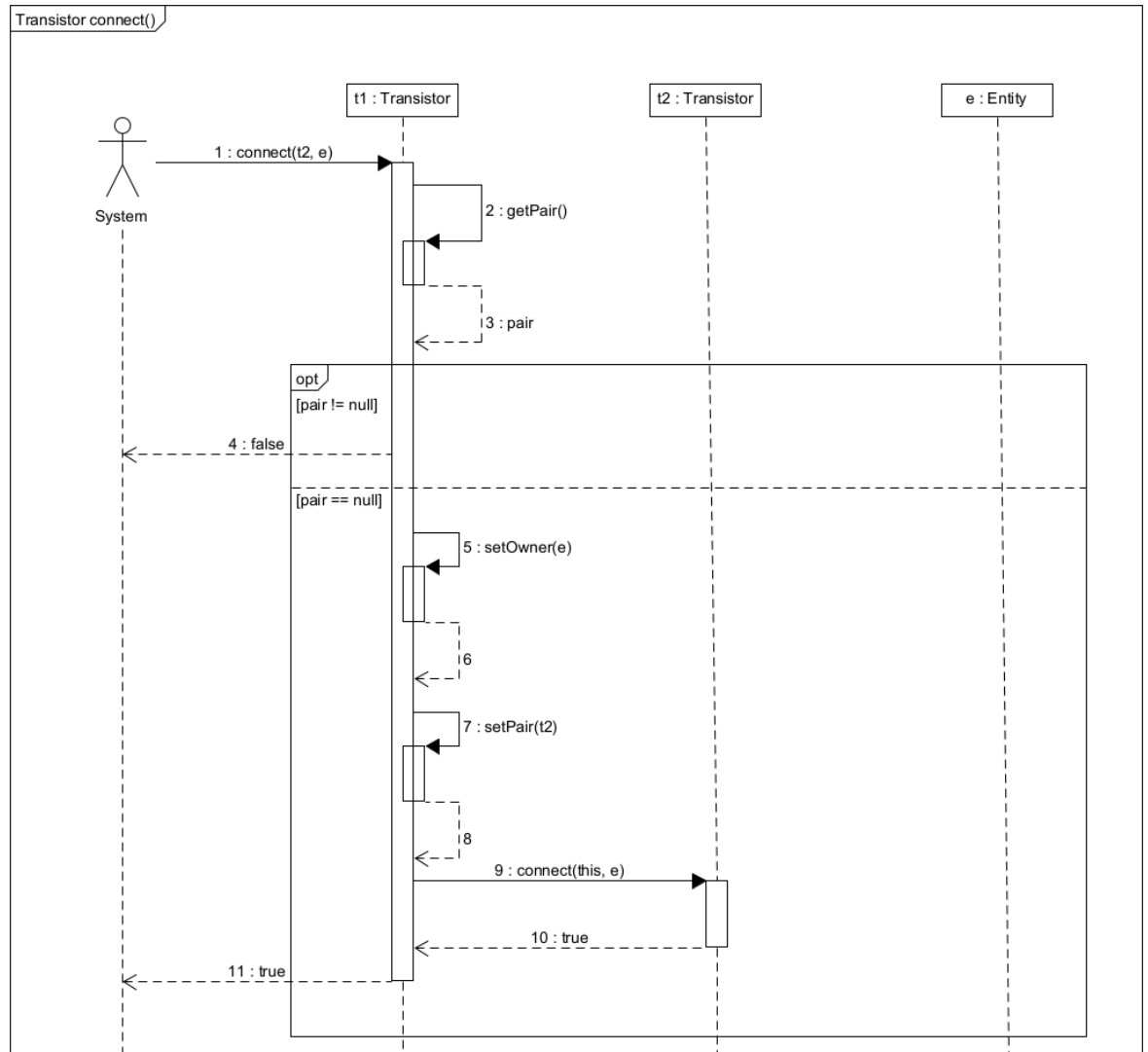
- **Entity.moveTo(Door):** A metódus belső működése megváltozott annak érdekében, hogy támogatni tudja azt az új követelményt, miszerint, ha egy takarító belép egy szobába minden mozogni képes embert kitessekel onnan. Ehhez kiegészült azzal, hogy meghívásakor ellenőrzi az adott Entity benuptási állapotát. Ha benupt, a mozgás nem történik meg, a visszatérési érték „false”. Ellenkező esetben a korábban dokumentált működés zajlik le.
- **Room.removeItem(Item):** A metódus belső működése megváltozott annak érdekében, hogy támogatni tudja azt az új követelményt, amely azt mondja hogy a szobák ragadóssá változhatnak. Ha egy szoba ragadóssá változik, az azt eredményezi, hogy a szoba tárgyai közül nem lehet egy tárgyat sem felvenni. Ennek kielégítése érdekében a metódus kiegészült azzal, hogy meghívásakor ellenőrzi a szoba ragacsosságát (`isSticky == true`). Hogyha ragacsos szobából próbálunk eltávolítani tárgyat, akkor a metódus 'null' értékkel tér vissza a megszokottal ellentétben. Ha a szoba melyből tárgyat próbálunk meg eltávolítani nem ragacsos, akkor a korábban dokumentált működés zajlik le.
- **Room.merge():** Ha az összeolvasztás sikeres, az új szoba ragacsosságának számítása csak a takarító belépéssel számolódik.
- **Room.split():** Ha a kettéválasztás sikeres, az új szobák ragacsosságának számítása csak a takarító belépéssel számolódik.
- **Room.acceptEntity(Entity):** A metódus belső működése megváltozott annak érdekében, hogy támogatni tudja azt az új követelményt, amely azt mondja hogy a szobák ragadóssá változhatnak. Ehhez, ha az Entity sikeresen regisztrálódott a szobában tartózkodók közé, utána meghívja az “entityEntered()” metódust. A működés többi részletében nem történt változás.

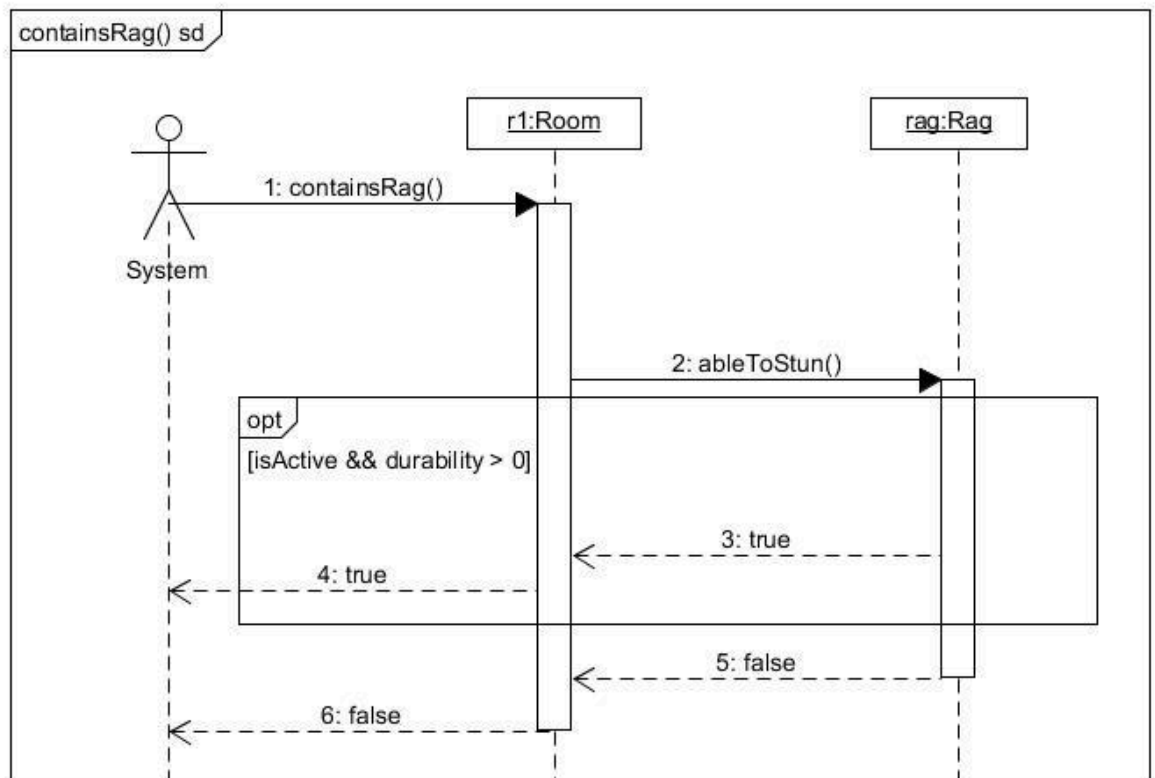
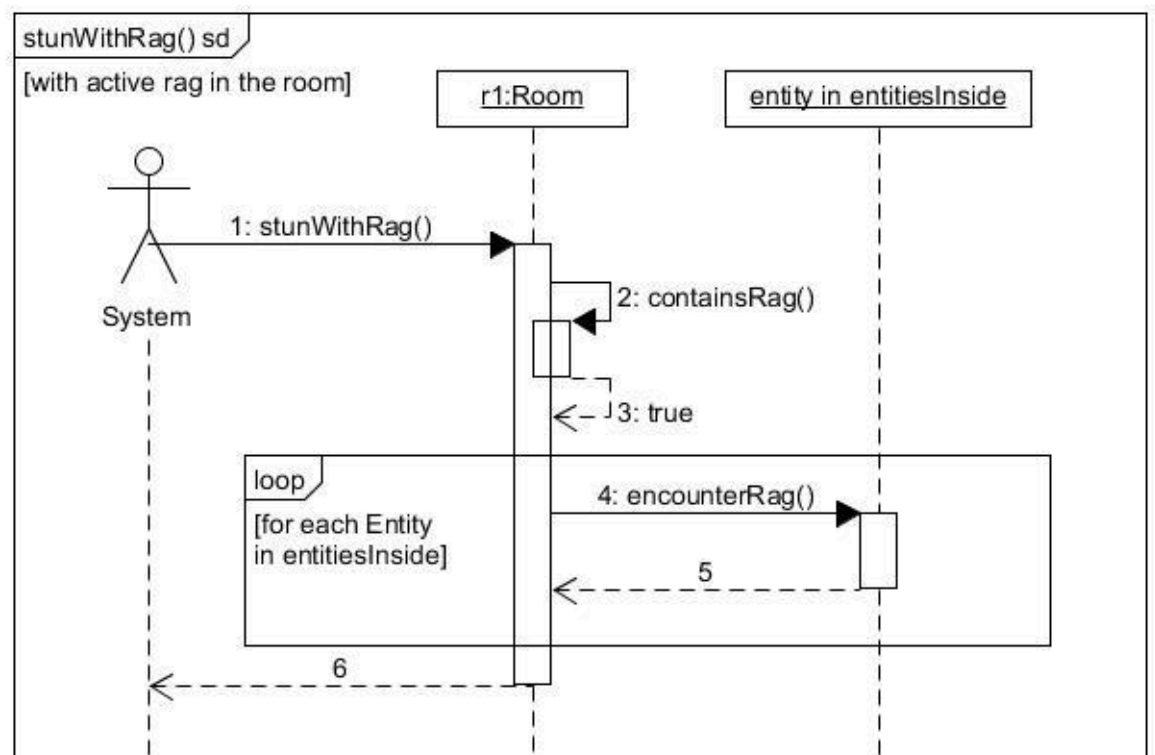
7.0.3 Új osztályok:

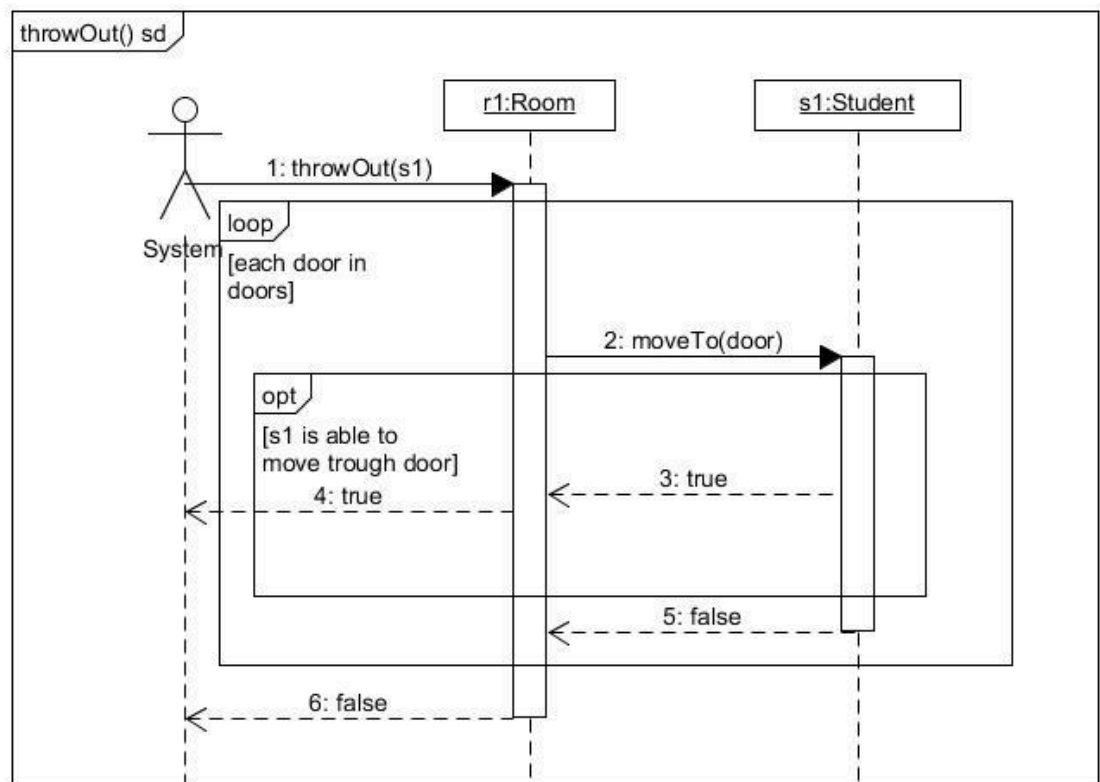
- **Janitor osztály:** az Entity osztályból származik le, a belép egy szobába, minden mozogni képes (nem benupt / ájupt) embert kitessekel onnan. Ha gázos szobába lép, kiszellőztet, megszüntetve a szoba gázosságát.
- **AirFreshener osztály:** az Item osztályból származik le, use() függvénye meghívja az Entity currentRoom-ján a makeNotToxic() metódust. Egy darab töltettel rendelkezik akárcsak a camembert, tulajdonképpen az ellentéte.
- **FakeMask osztály:** a játékos szempontjából maszknak látszó tárgy azonban, mérgező szobába kerülve hiába van nála ugyanúgy eszméletét fogja veszteni mintha nem lenne, a felvételen és eldobáson kívül nem lehet mást csinálni vele.
- **FakeSlideRule osztály:** a játékos szempontjából logarlécnek látszó tárgy azonban, a felvétel után játékos nem nyer. Csak felvenni és eldobni lehet ezt a tárgyat.
- **FakeTVSZ osztály:** a játékos szempontjából TVSZ-nek látszó tárgy azonban tanárral való találkozás esetén nem védi meg őt. Csak felvenni és eldobni lehet ezt a tárgyat.

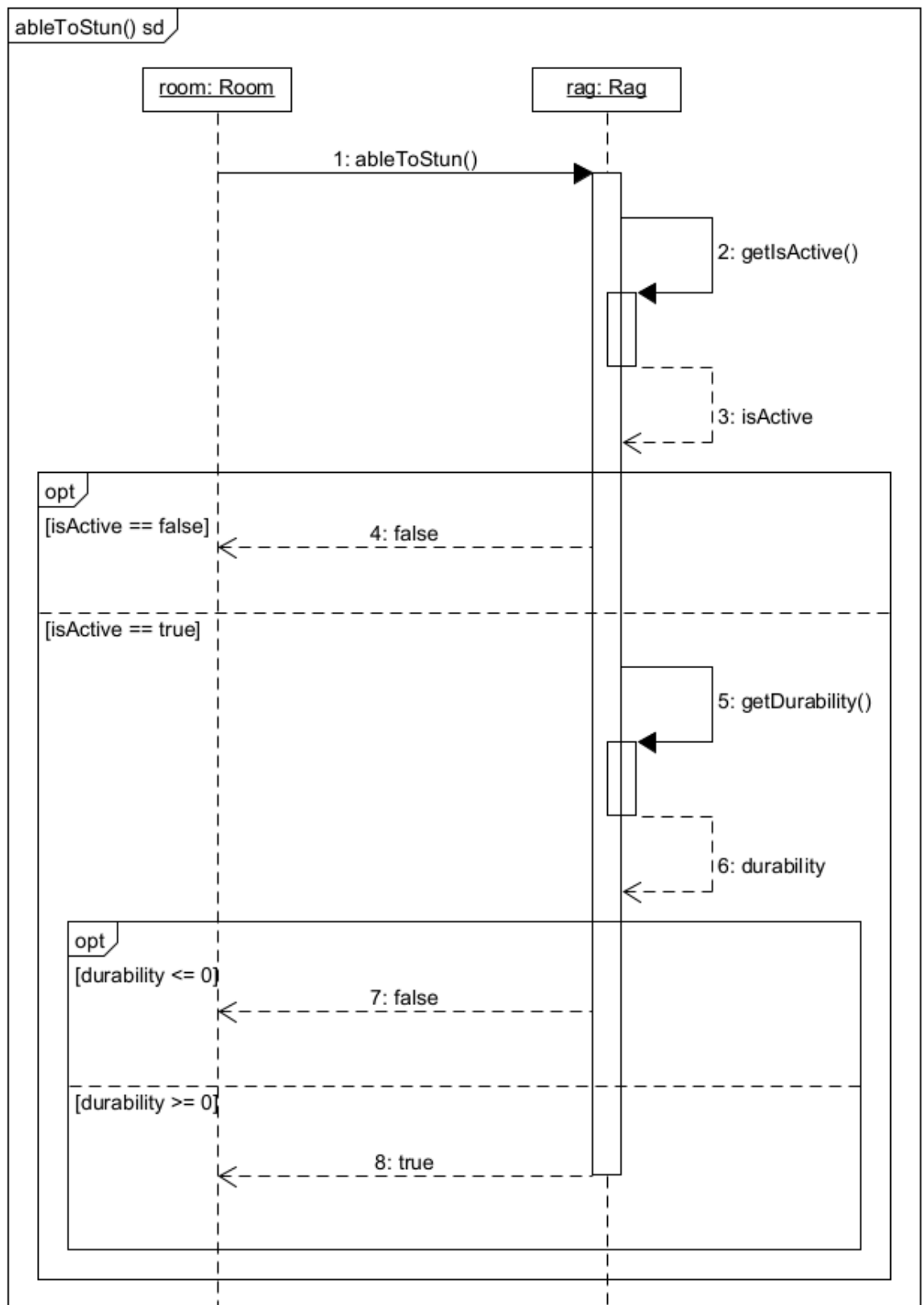
7.0.4 Szekvencia-diagramok

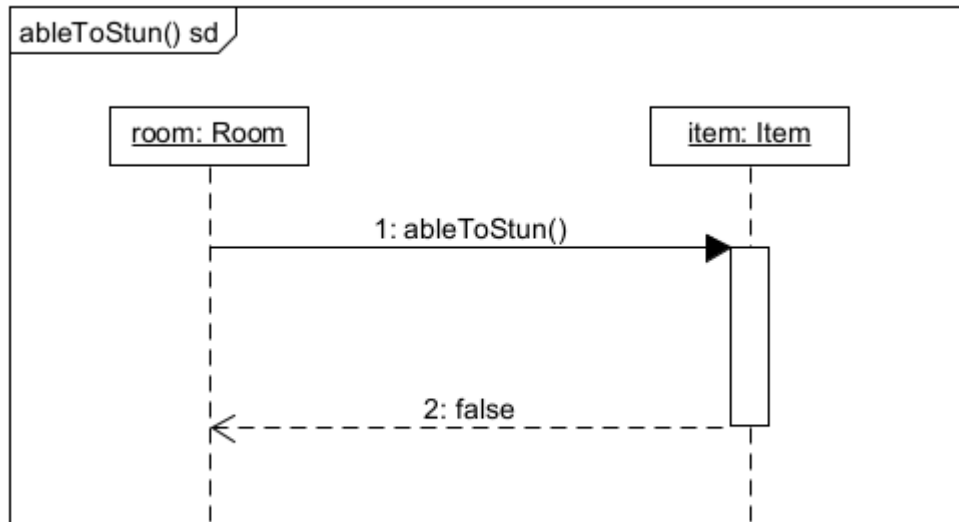
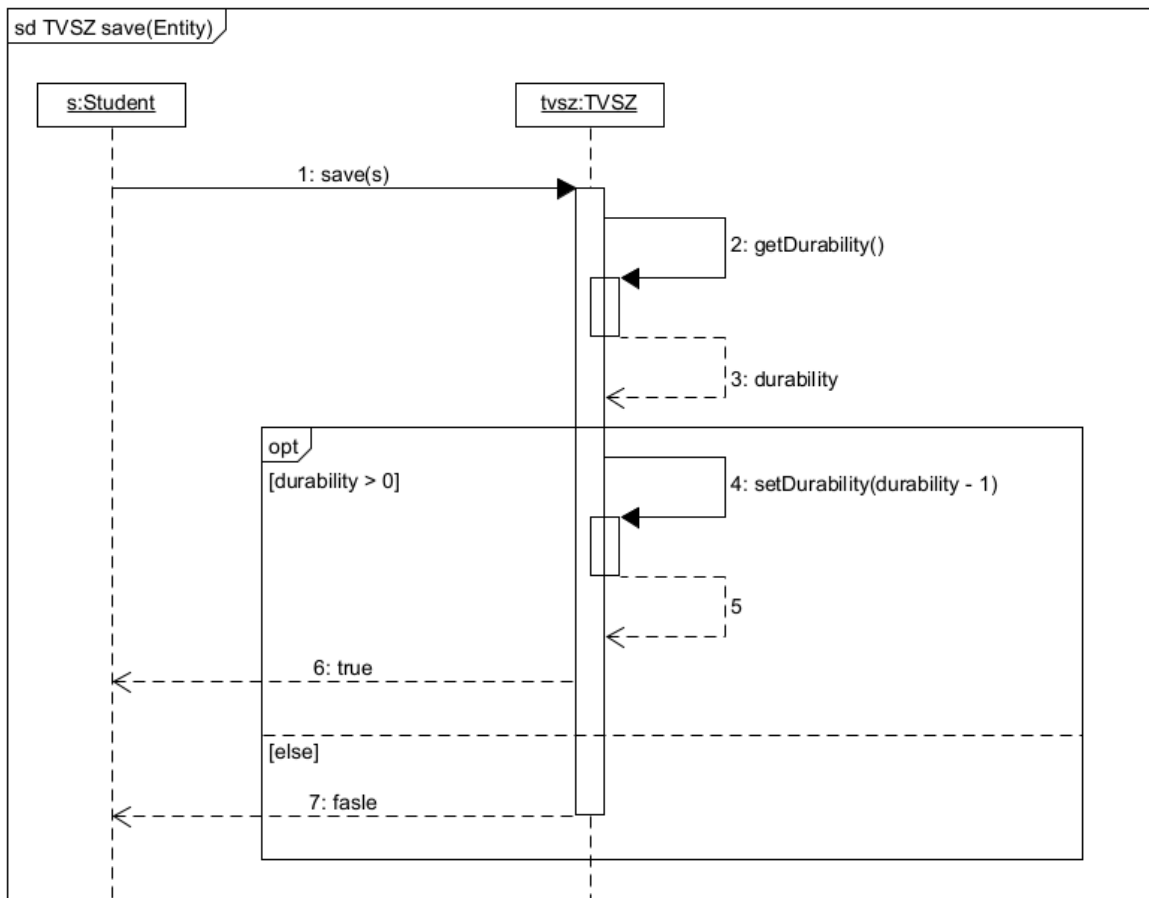
1. Transistor connect()



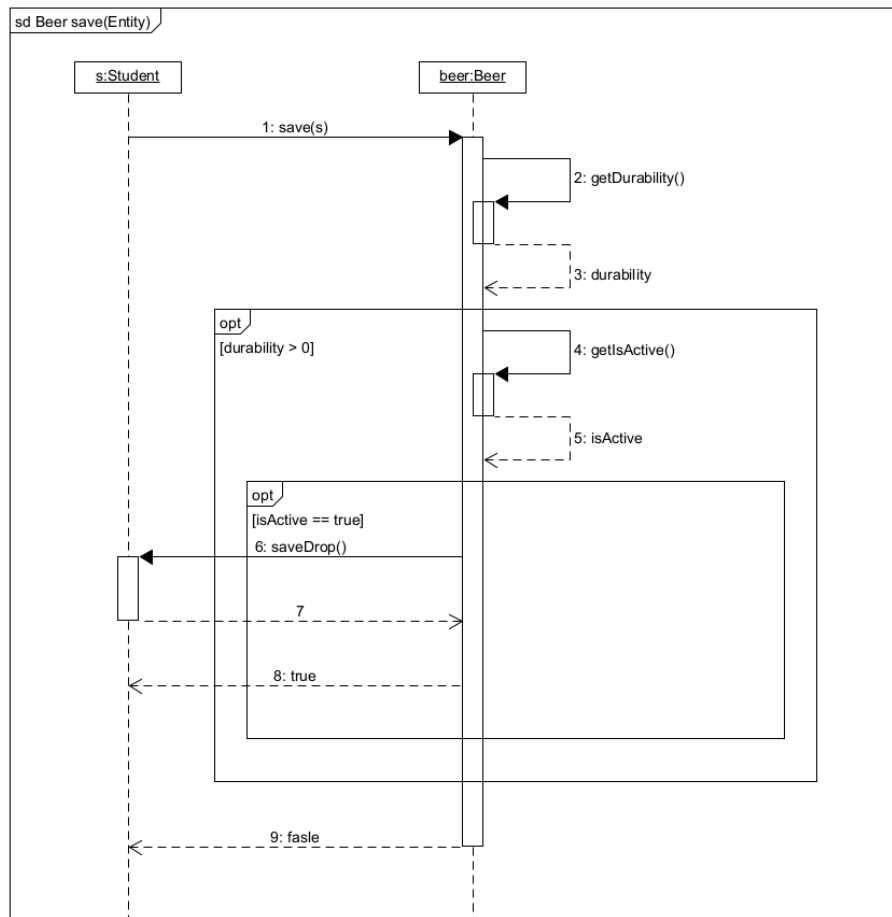
2. containsRag()**3. stunWithRag()**

4. throwOut()

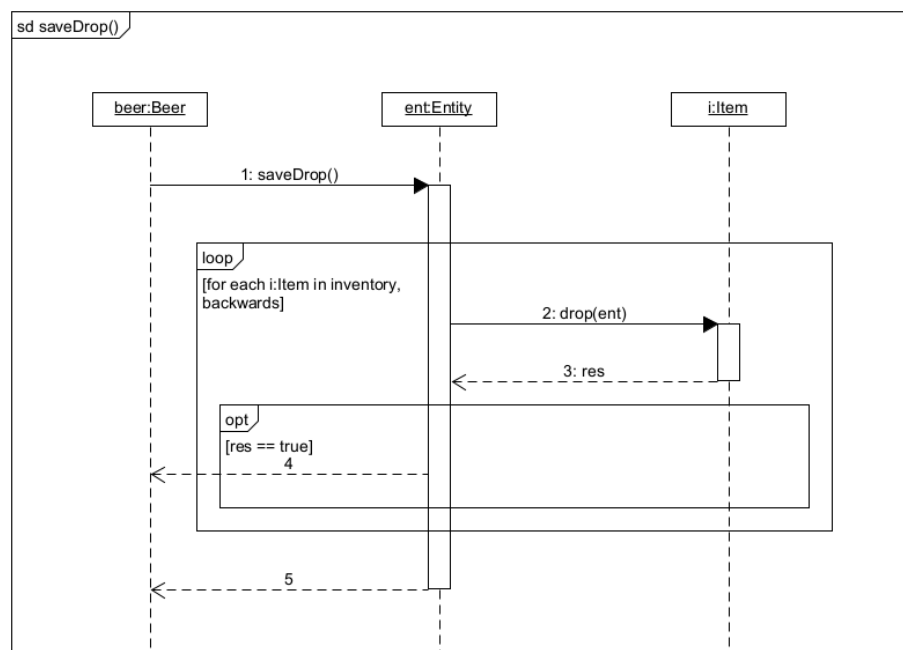
5. ableToStun() if Item is Rag

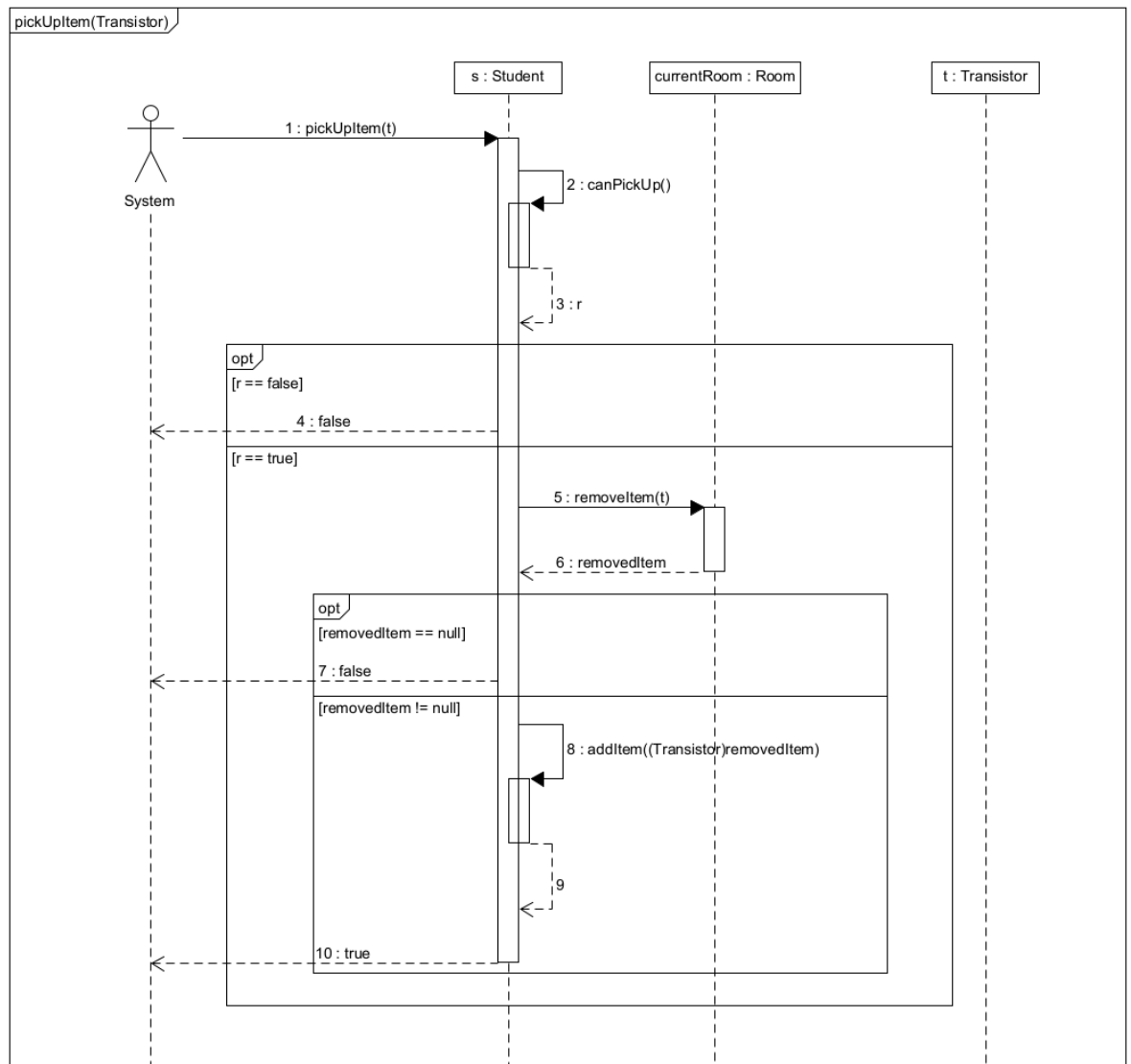
6. ableToStun() if Item is not Rag**7. TVSZ save(Entity)**

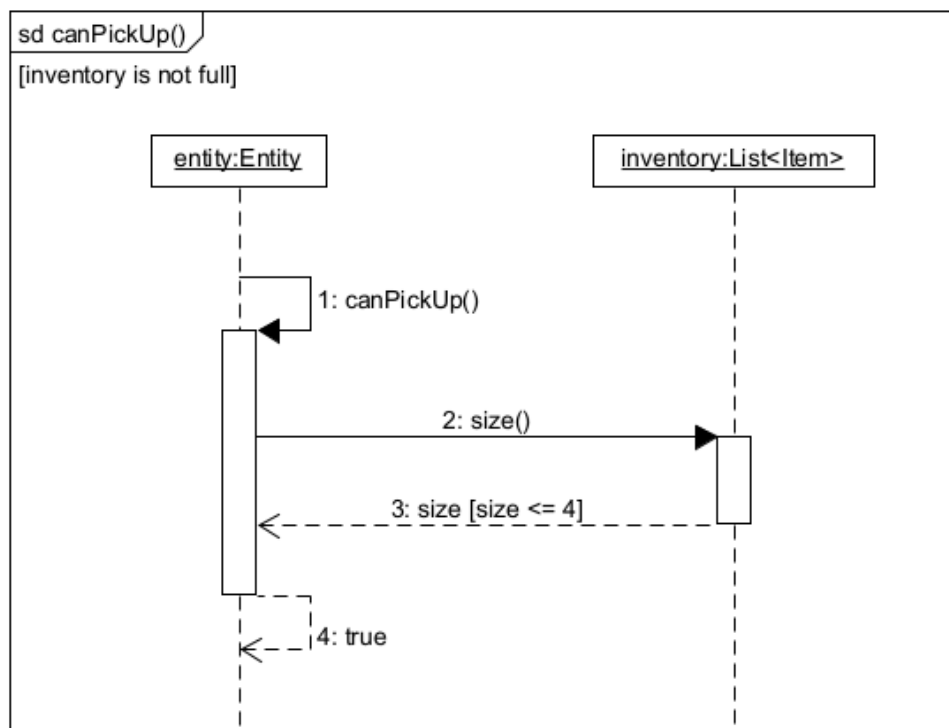
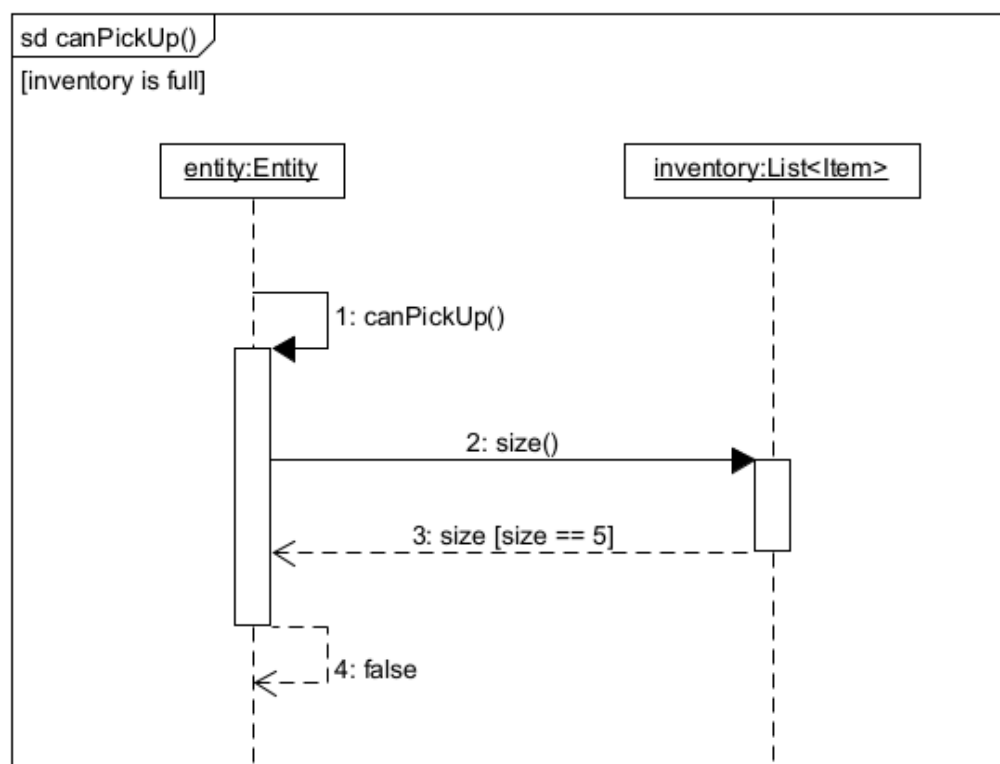
8. Beer save(Entity)



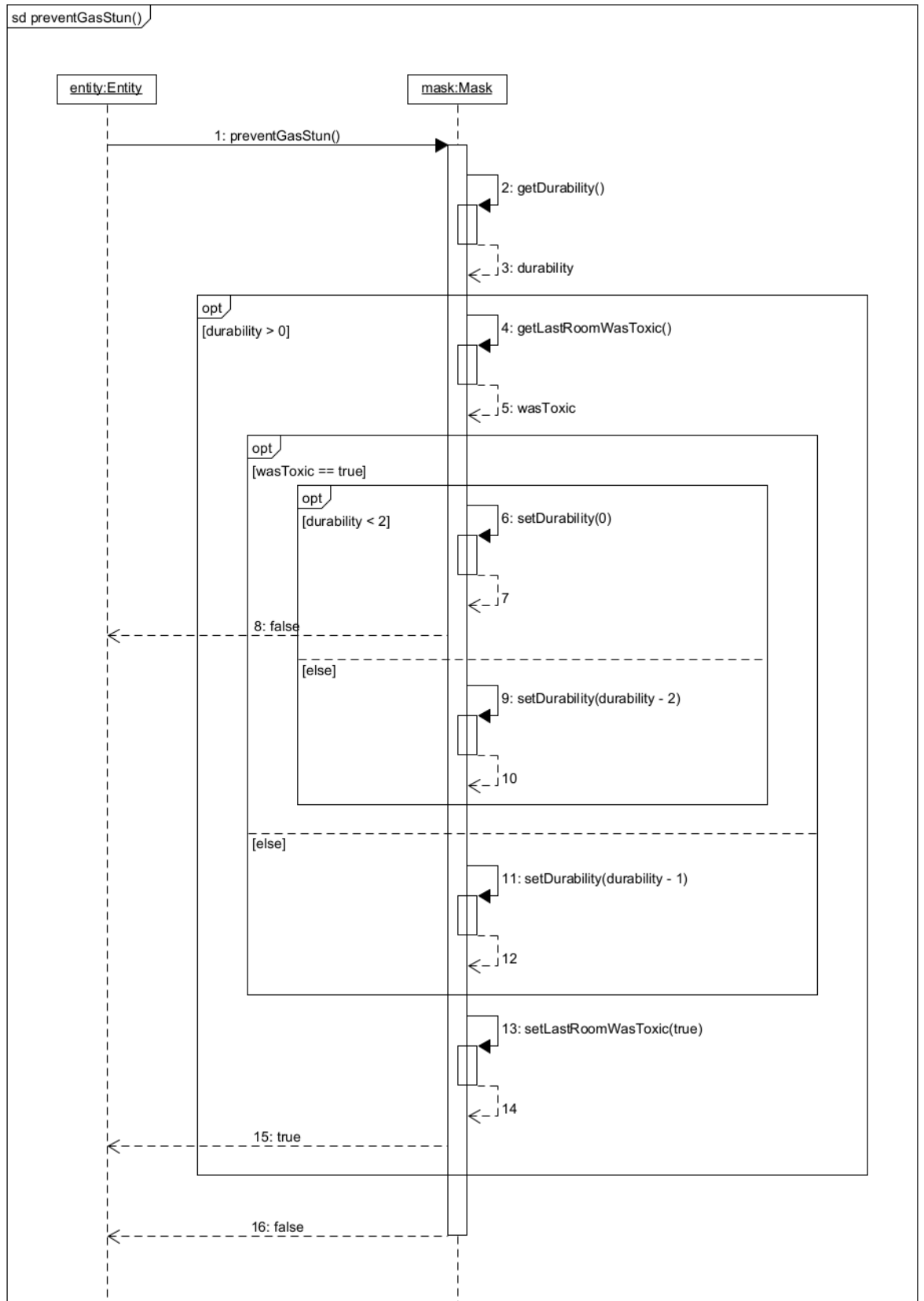
9. saveDrop()

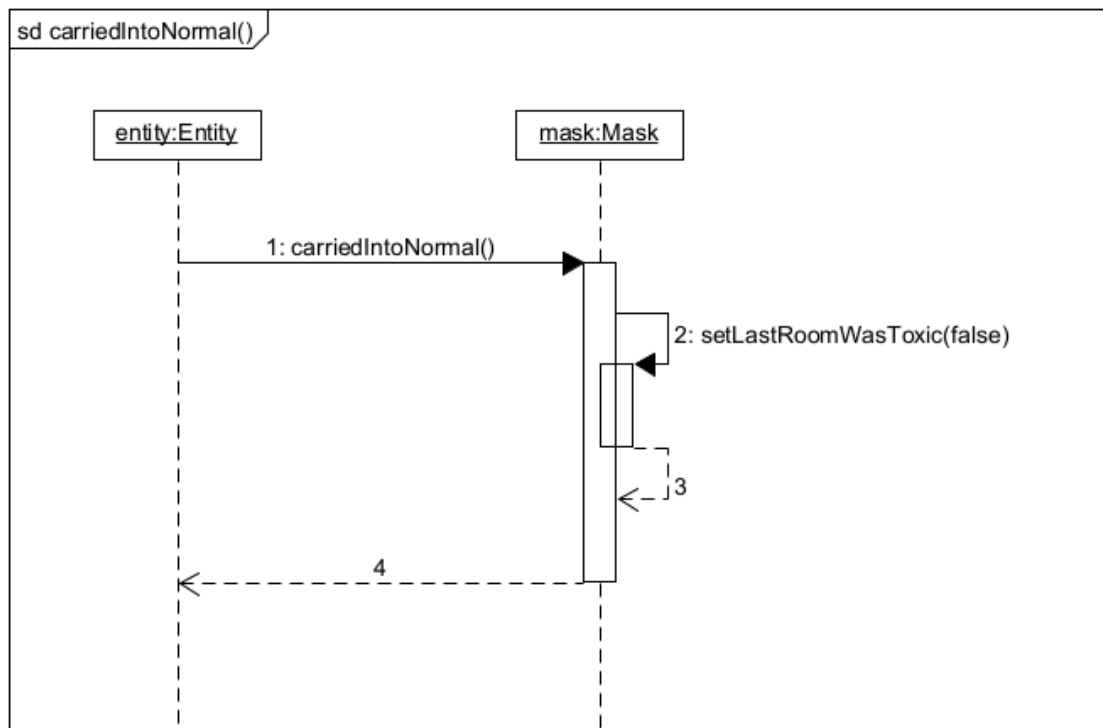
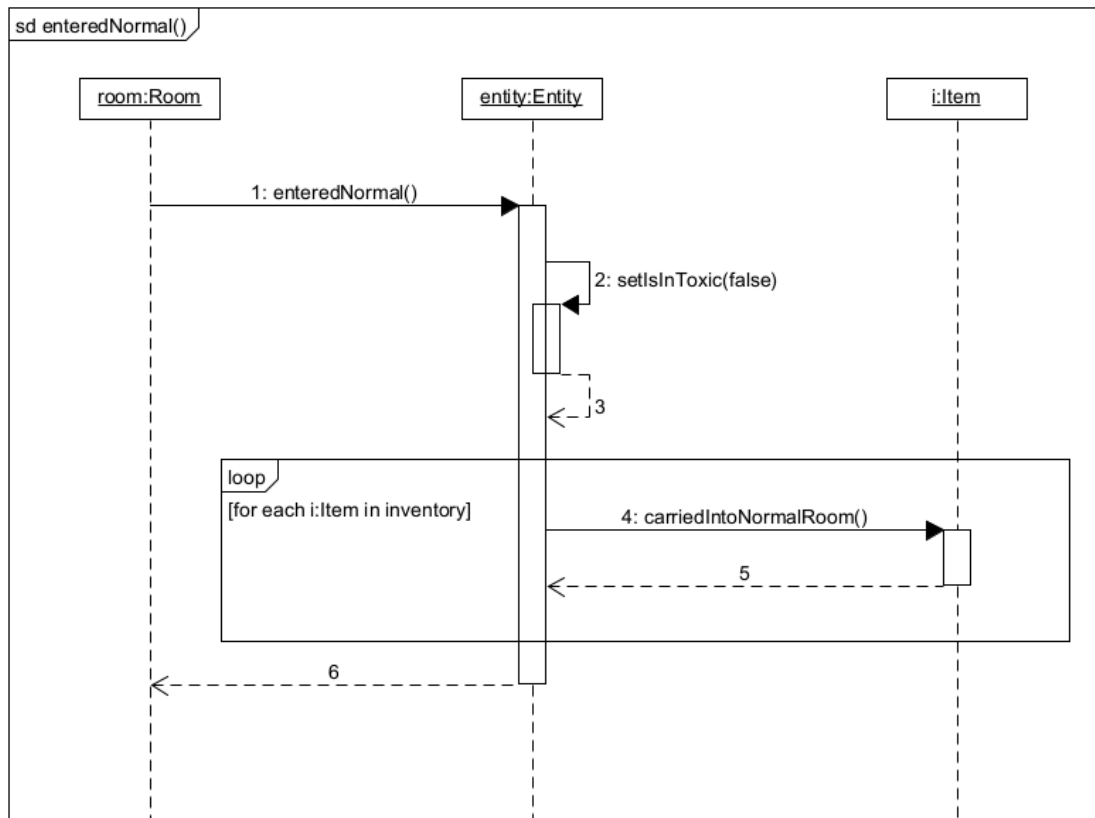


10. pickUpItem(Transistor)

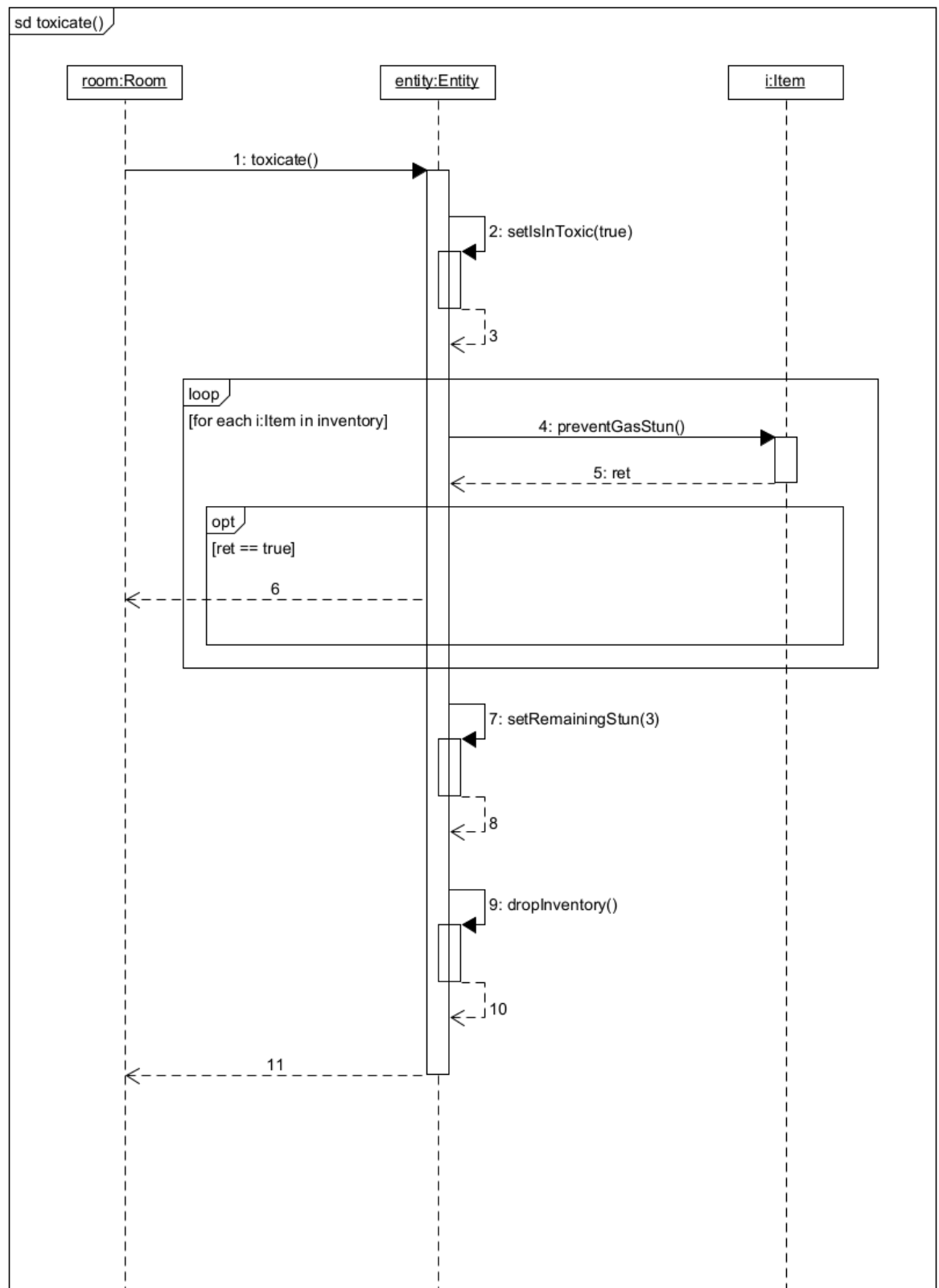
11. canPickUp() inventory is not full**12. canPickUp() inventory is full**

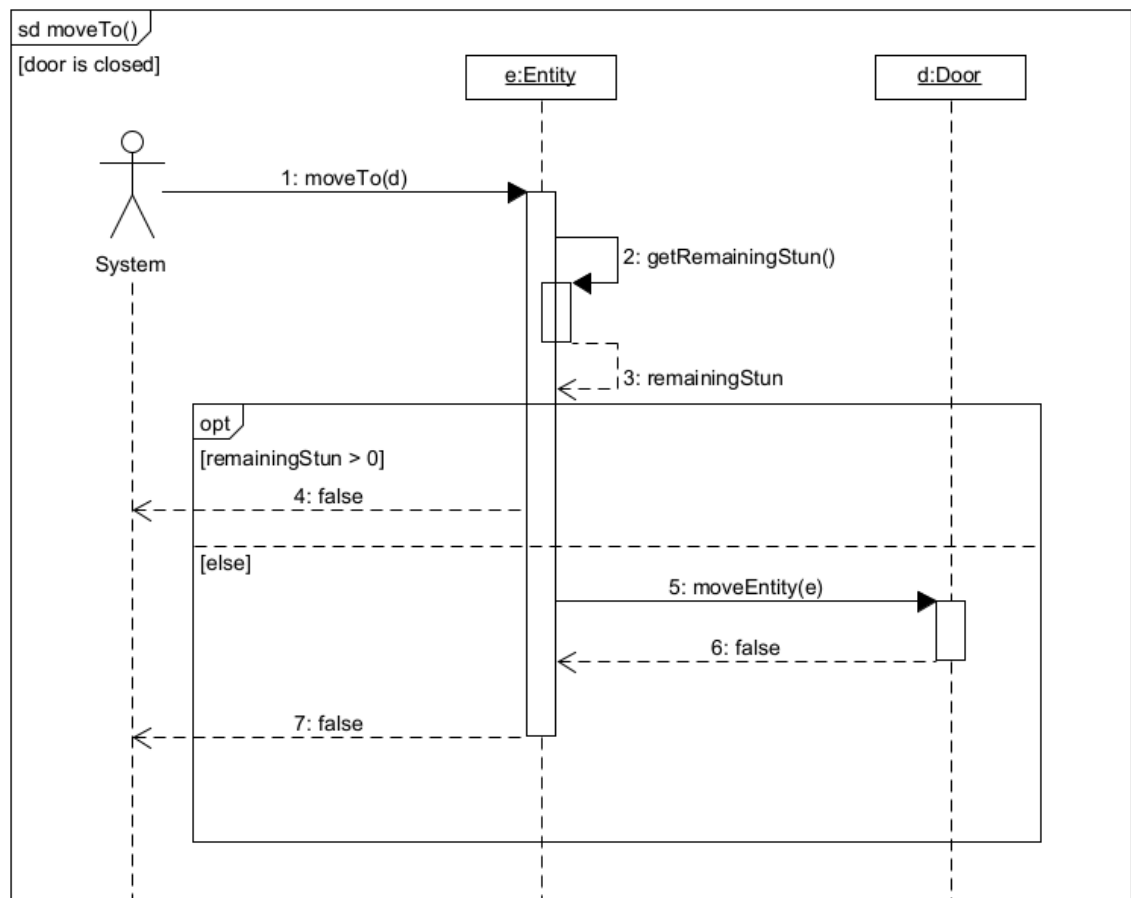
13. Mask preventGasStun()

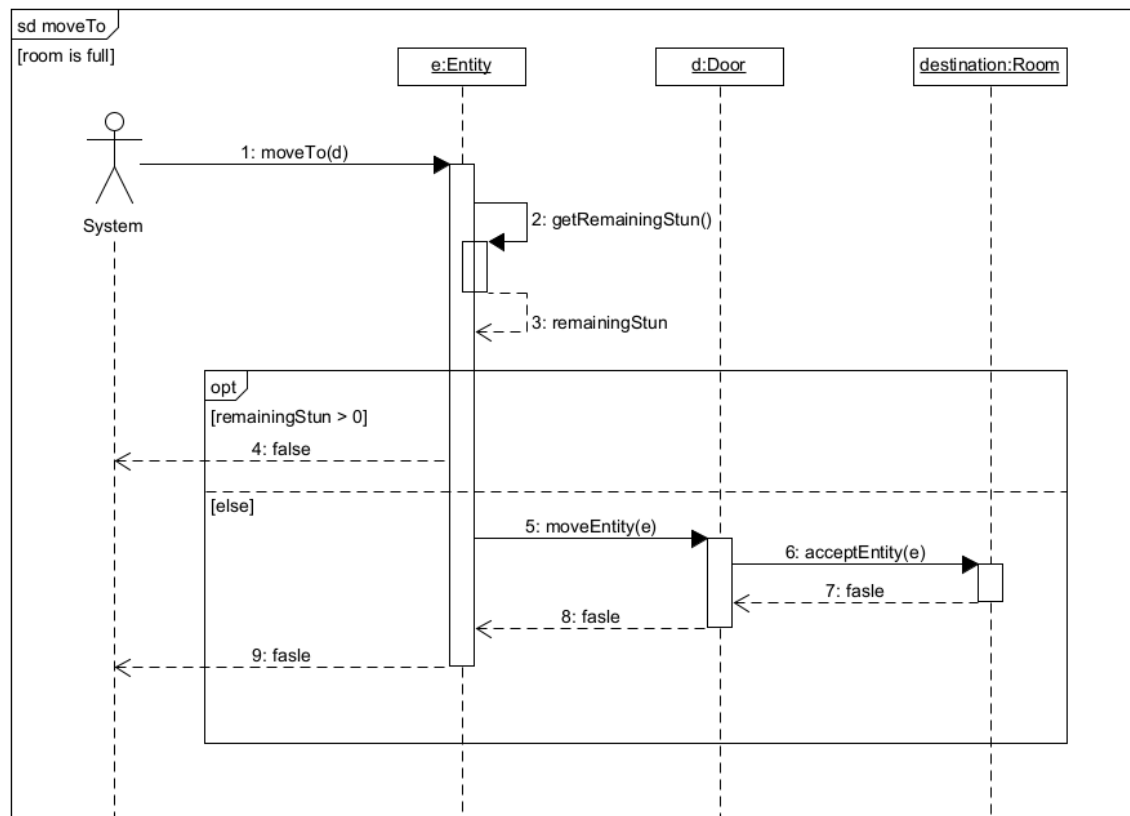


14. Mask carriedIntoNormalRoom()**15. enteredNormal()**

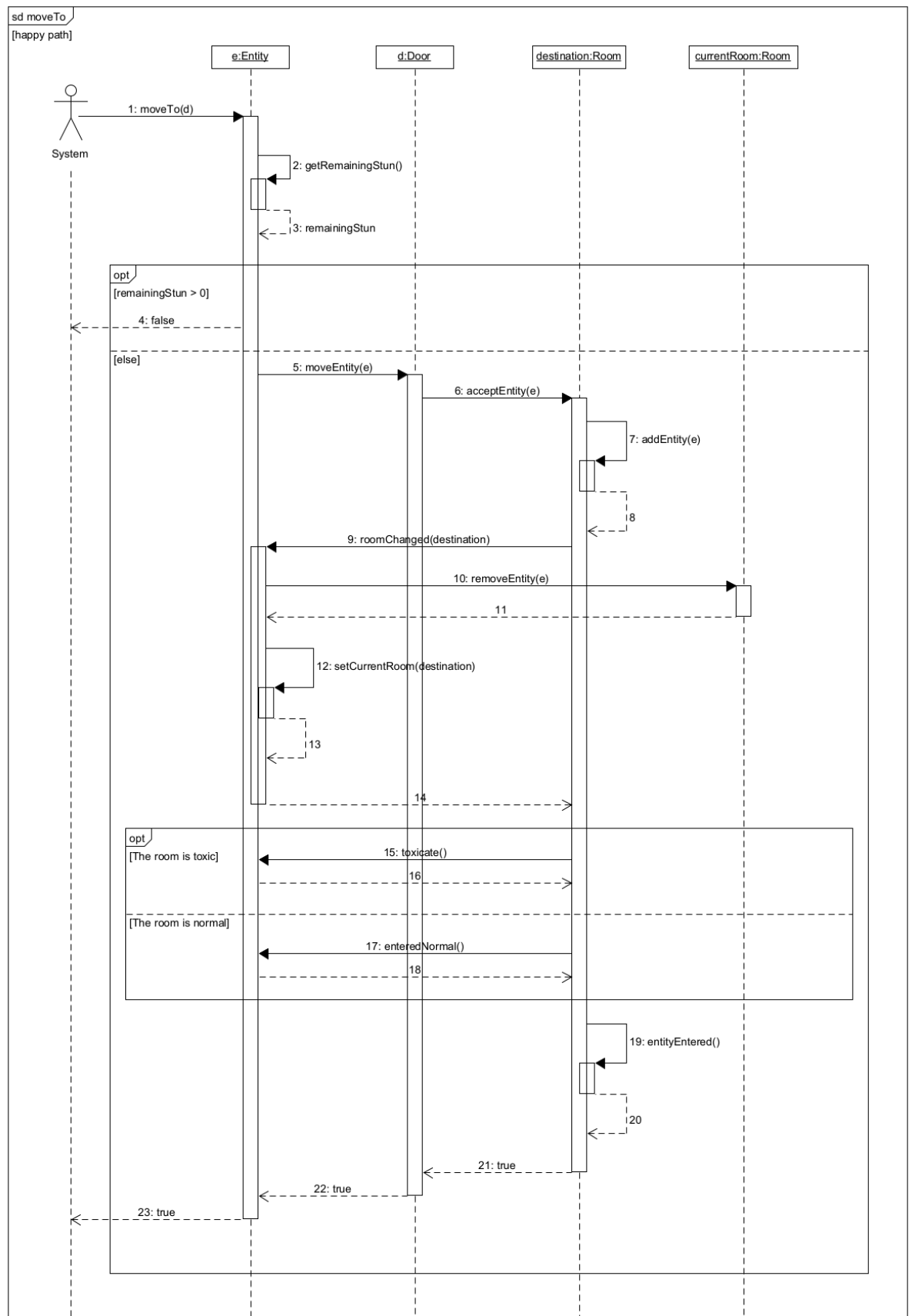
16. toxicate()

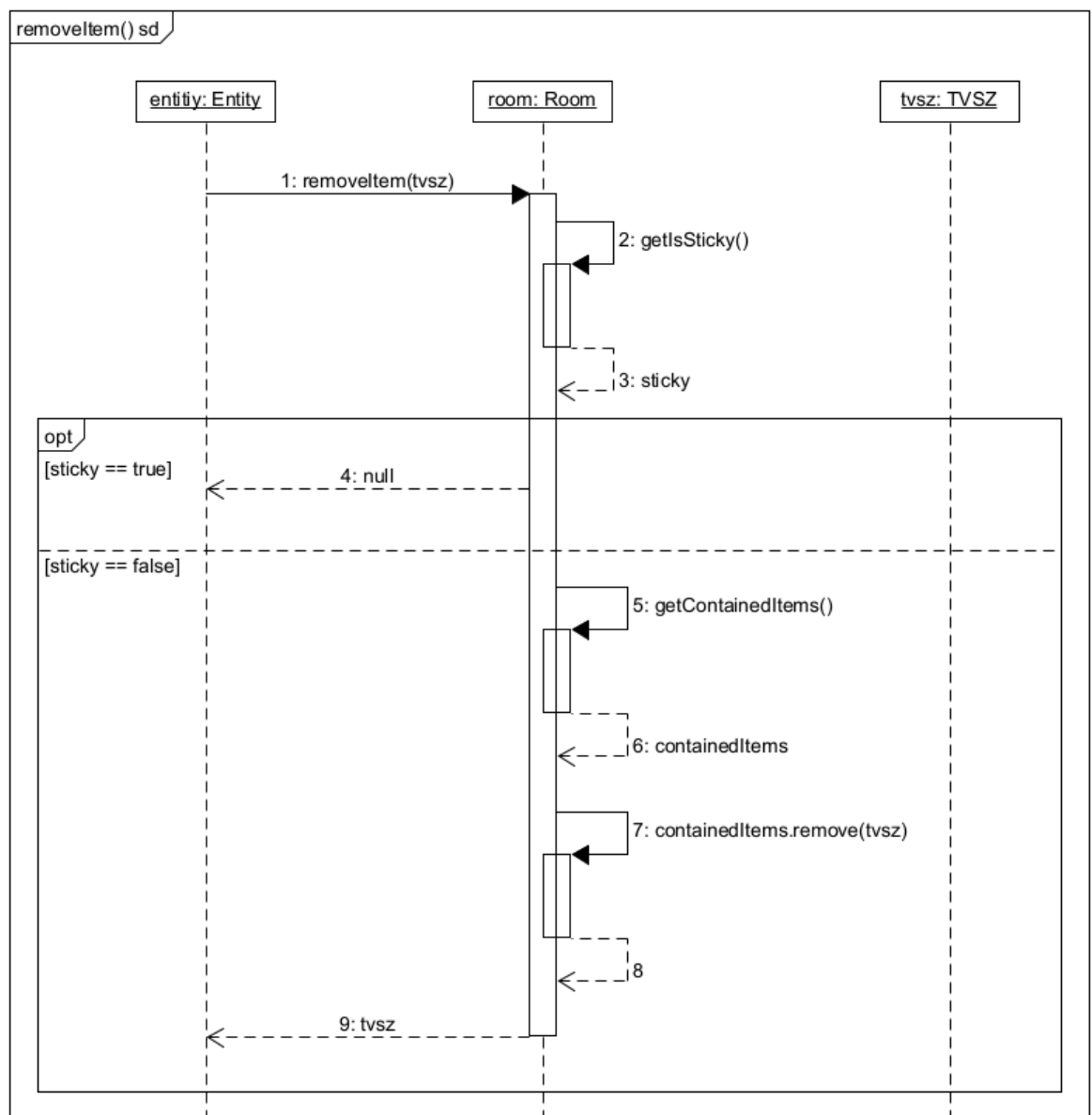


17. moveTo() door is closed

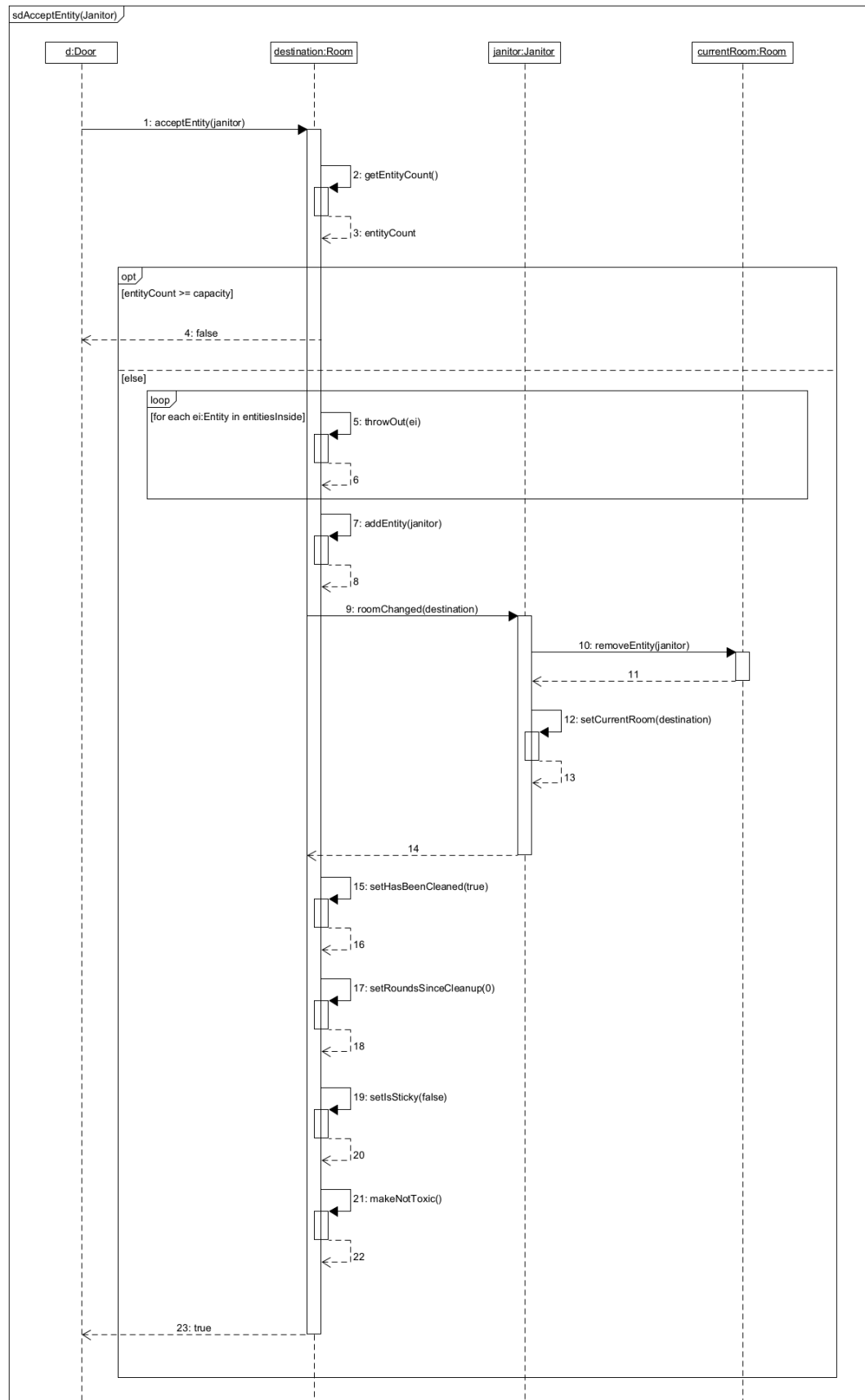
18. moveTo() room is full

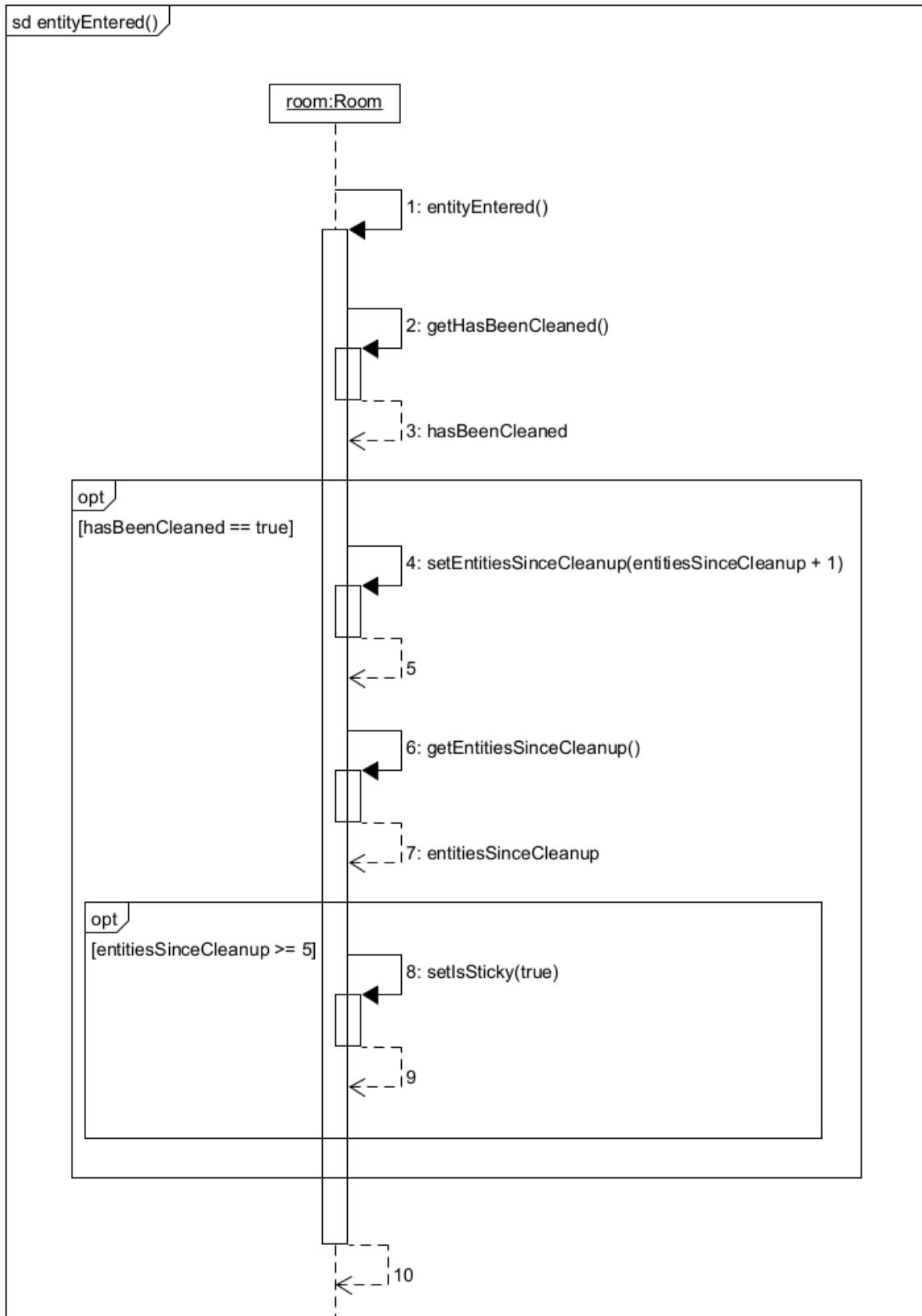
19. moveTo() happy path & acceptEntity(Entity)

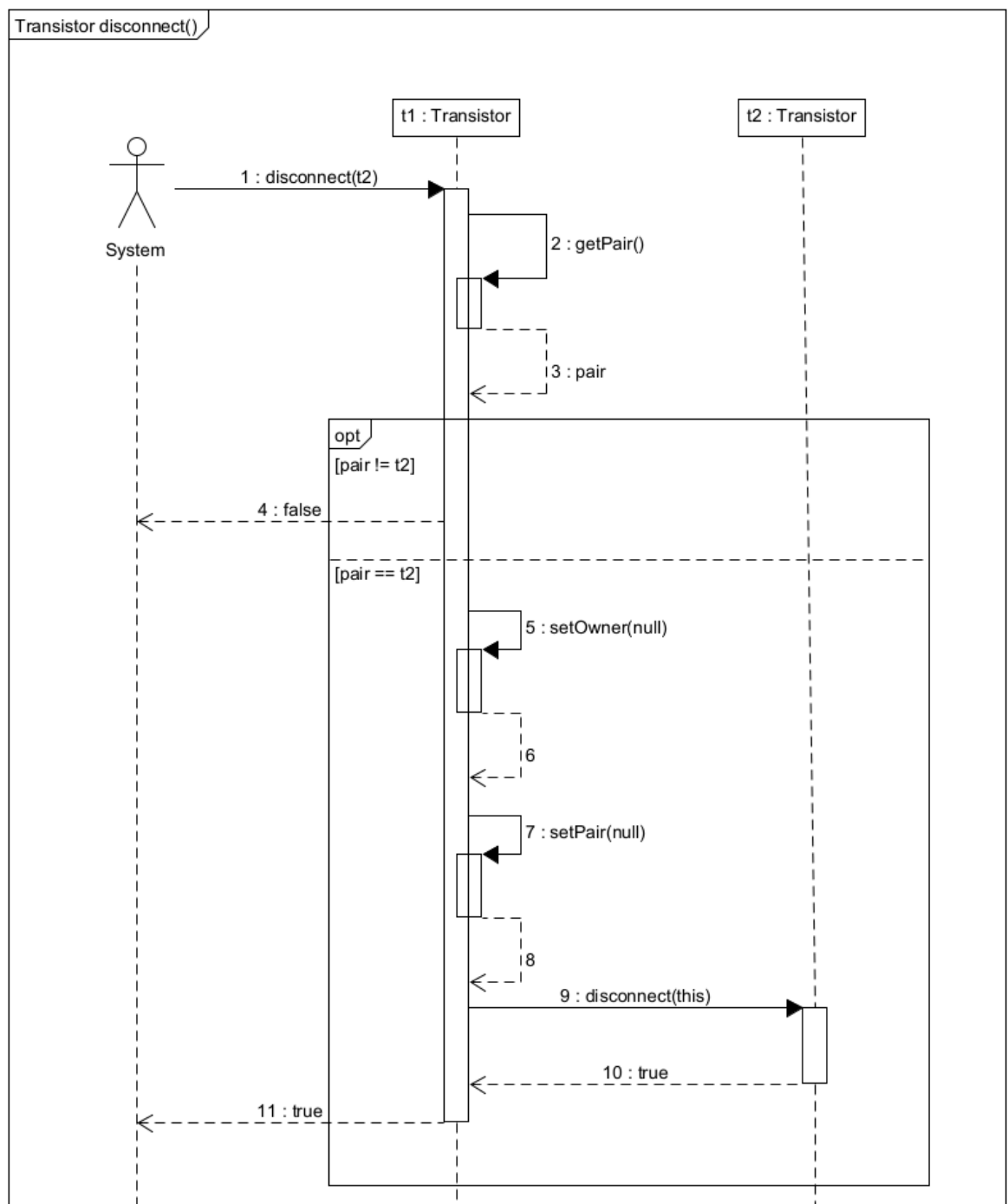


20. removeItem()

21. acceptEntity(Janitor)



22. Room entityEntered()

23. Transistor disconnect()

7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A bemenetet meg lehet adni egy txt fájlban, megegyező módon ahogy a parancsokat a szabványos bemenetre íránk. A parancsokat ENTER-el kell elválasztani(‘\n’).

7.1.2 Bemeneti nyelv

A játék elindítása előtt csak az addplayer, removeplayer, load, showplayers és start parancsok használhatók! Ez fordítva is igaz, az indítás után csak a többi parancs használható, az itt felsoroltak nem! Bármilyen hibás parancs kimenetet eredményez. A parancsokhoz egy időben csak egy kapcsoló adható meg.

debug

Leírás: A parancs arra használható, hogy kiírja egy szöveges file-ba egyes objektumok állapotát, tesztelés céljából. A parancs után kell írni az objektumok nevét amiket ki akarunk írni.

debug <obj1> <obj2> ...

load

Leírás: Használatával betölthetünk egy már elmentett játékmenetet. A játék betöltéséhez meg kell adnunk egy fájlnevet. A betöltendő fájl a játék mappájában kell hogy legyen. Ha sikeres a fájlbetöltés, akkor a szoftver ezt üzenettel közli. Ha sikertelen a fájlbetöltés, vagy helytelen fájlnevet adunk meg, akkor azt is szöveges üzenettel közli a felhasználóval a szoftver. A parancs csak a játék indítása előtt működik, hogyha a felhasználó a játék kezdete után szeretné használni, akkor hibaüzenettel válaszol a szoftver. A parancs formátuma:

load <filename>

Opciók: -

save

Leírás: Használatával elmenthetjük a játékmenetet. A játék lementéséhez megadhatunk egy filenevet is, de ez nem kötelező. A file a játék mappájában tárolódik el. Hogyha megadtunk egy nevet, akkor a játék állása a megadott nevű fileba fog elmentődni. Hogyha létezik már egy ilyen nevű file, akkor a játék azt felülírja. Ha nem adtunk meg nevet, akkor a játék egy alap fájlnévre menti el az állást. Ezt a parancsot csak a játék elkezdése után adhatjuk ki. A játék válaszol a mentés sikerességéről, vagy esetleges sikertelenségéről. A parancs formátuma:

save

Opciók: [filename] A mentési fájl neve.

execute

Leírás: Egy bemeneti nyelv parancsainak listáját végrehajtó parancs, amely a végrehajtandó parancsokat egy '.txt' fájlból olvassa be, amit kötelező név szerint megadnunk a parancs kiadásakor.

execute <filename>

Opciók: -

addplayer

Leírás: Használatával hozzáadunk egy új játékost a játék kezdete előtt. A játék elkezdéséhez legalább kettő játékosra van szükség. A játékos létrehozásához kötelező egy nevet adni a játékosnak, amelyen a játék során hívni fogják. Ez a név nem lehet megegyező egy már másik létező játékos nevével. A parancs csak akkor megy végbe, hogyha a játék még nem indult el. A parancs formátuma:

addplayer <name>

Opciók: -

removeplayer

Leírás: Használatával eltávolíthatunk egy játékost a játék kezdete előtt. A játék elkezdéséhez legalább kettő játékosra van szükség. Kötelező megadni az eltávolítandó játékos nevét, így választhatjuk ki, hogy melyik játékost szeretnénk törölni a játékból. A parancs csak akkor megy végbe, hogyha a játék még nem indult el. A parancs formátuma:

removeplayer <name>

Opciók: -

showplayers

Leírás: Ezzel a paranccsal a felhasználó kilistázhatja az eddig hozzáadott játékosokat. Segítségével ellenőrizni tudja, hogy mennyi, és milyen nevű játékosok vannak a játékban. A parancs csak a játék elindítása előtt működik. A parancs formátuma:

showplayers

Opciók: -

start

Leírás: Ezzel a paranccsal indíthatjuk el a játékmenetet. Csak akkor indul el a játék, hogyha létezik legalább kettő játékos, melyet az 'addplayer' paranccsal adtunk hozzá a játékhoz. A parancs csak akkor megy végbe, hogyha a játék még nem indult el. A parancs formátuma:

start

Opciók: -

show

Leírás: Ezzel a paranccsal kilistázzhatunk adatokat a játékosról, vagy a játékos tartózkodási szobájáról. A következő kapcsolók léteznek:

- **-i:** a játékos hátizsákjának tartalmát listázza ki.
- **-d:** a játékos tartózkodási szobájának ajtajait listázza ki.
- **-e:** a játékos tartózkodási szobájában lévő összes entitást listázza ki.
- **-ri:** a játékos tartózkodási szobájában lévő összes tárgyat listázza ki.
- **-t:** a játékos tranzisztorttal való teleportálási lehetőségeit listázza ki.

Továbbá, a parancs csak akkor megy végbe, hogyha a játék már el lett indítva. Ellenkező esetben a szoftver szöveges üzenetben jelzi a felhasználónak, hogy sikeres vagy sikertelen volt a parancs lefutása. A parancs formátuma:

show -i

Opciók:

- i:** Opció arra, hogy a játékos hátizsákjának tartalmát listázzuk ki.
- d:** Opció arra, hogy a játékos szobájának ajtajait listázzuk ki.
- e:** Opció arra, hogy a játékos.
- ri:** Opció arra, hogy a játékos szobájának tárgyait listázzuk ki.
- t:** Opció arra, hogy a játékos teleportálási lehetőségeit listázzuk ki.

pickup

Leírás: Ezzel a paranccsal a játékos felveheti az általa kijelölt tárgyat. A parancs után kötelező megadni a felvételre szánt tárgy ID-ját. Csak olyan tárgy vehető fel, amely megtalálható a játékos szobájában. A felvétel csak akkor lehetséges, ha még van üres hely a parancsot futtató entitás hátizsákjában. A parancs formátuma:

pickup <itemID>

Opciók:

- a:** Amennyiben van elegendő hely a játékos hátizsákjában, az összes tárgy felvételre kerül.

drop

Leírás: Ezzel a paranccsal a játékos eldobhatja az általa kijelölt tárgyat a hátizsákjából. A tárgy csak akkor dobható el, ha a játékos hátizsákjában található. A parancs után kötelező megadni az eldobásra szánt tárgy ID-ját. A parancs formátuma:

drop <itemID>

Opciók:

- a:** Üríti a teljes hátizsákot, minden tárgyat eldob.

use

Leírás: A parancs használatával használhat a játékos egy tárgyat. Ez csak akkor lehetséges, amennyiben a megadott ID-jú tárgy megtalálható a hátizsákban, és a tárgy használható. A parancs formátuma:

use <itemID>

Opciók: -

move

Leírás: A parancs használatával lehetséges mozogni szomszédos szobák között. A parancs mögé megadható egy szomszédos szobába vezető ajtó ID-je. Ha nem adunk meg ilyen azonosítót, akkor a szobában maradási szándékunkat jelezzük. Ha adunk meg egy ID-t, azonban egy olyat adunk meg ami nem megfelelő formátumú, vagy esetleg nem elérhető a szobából, akkor a parancs sikertelen. Ha helyes azonosítót adunk meg, akkor ezzel a mozgási szándékunkat jelezzük, és a parancs le fog futni. Ha a parancs sikeresen lefut, a játékos köre véget ér, további műveletet a fordulón belül nem hajthat végre. A parancs formátuma:

move <doorID>

Opciók: -

connect

Leírás: Ezzel a paranccsal köthető össze két tárgy, melyek a játékos hátizsákjában található, és még nincsenek párosítva. A parancs sikeres végrehajtásához meg kell adni kettő, tárgy ID-ját. Hatással csak két nem összekötött tranzisztorra van, minden más esetben hiba jelenik meg. A parancs formátuma:

connect <itemID1> <itemID2>

Opciók: -

disconnect

Leírás: A parancs használatával lehet szétkapcsolni két tárgyat, melyek a játékosnál található, és össze vannak párosítva. Hatással csak kettő összekapcsolt tranzisztorra van, bármilyen más tárgyak szétkapcsolása esetén hiba jelenik meg. A parancs formátuma:

disconnect <itemID>

Mivel egy összekapcsolt tranzisztoron keresztül elérhető a párja, ezért elég a pár egyik tagjának ID-ját megadni.

Opciók: -

teleport

Leírás: A parancs használatával képes a játékos teleportálni. Ha a teleportálásra használt tárgy nem megfelelő, vagy a cél szoba tele van, a parancs nem fut le helyesen. A parancs formátuma:

teleport <itemID>

Opciók: -

7.1.3 Kimeneti nyelv

debug

Üzenetet add a file névről, és leírja az objektumok nevét amik kiírásra kerültek.
Az egyes objektumok kiírási formátuma:

```
-- <változónév> :<osztálynév> --  
<változó1> - <változó1 érték>  
<változó2> - <változó2 érték>  
<...> - <...>  
--END <változónév> :<osztálynév> --
```

Egy lehetséges, sikeres kimenet:

```
Successfully wrote debug to "debug.txt"  
Objects dumped:  
- player12  
- room354  
- beer44
```

Egy lehetséges, hibás kimenet rossz objektummal paraméterezve:

```
Error: object player33 doesn't exist!
```

load

Kiírja, hogy a fájl betöltése sikeres volt-e vagy sem. Hiányzó fájlnev esetén kiírja, hogy a parancs lefutása a helytelen paraméterezés miatt nem sikerült. Hogyha ilyen nevű fájl nem létezik, akkor erről értesíti a felhasználót.

Egy lehetséges kimenet sikeres beolvasás esetén:

```
File 'jatekmenet2' successfully loaded.
```

Egy lehetséges kimenet hibás fájlnev esetén:

```
Error loading file 'jatekmenet53', file not found. Please try again.
```

Egy lehetséges kimenet fájlnev hiánya esetén:

```
Error, file name not specified. Please try again.
```

Egy lehetséges kimenet rossz helybeni híváskor:

```
Error, this command can not be called right now.
```

save

Kiírja, hogy a játékmenet mentése sikeres volt - e vagy sem. Ha lett megadva fájlnev, akkor a megadott néven menti el a menetet. Ha nem lett megadva név, akkor pedig tudatja a felhasználóval, hogy milyen nevű fájlként lett elnevezve haladása.

Egy lehetséges kimenet sikeres megadott neves mentés esetén:

```
Your progress has been successfully saved under the file named  
'jatekmenet3'.
```

Egy lehetséges kimenet sikeres megadott név nélküli mentés esetén:

```
Your progress has been successfully saved under the standard file named  
'mentes'.
```

Egy lehetséges kimenet hiba esetén:

```
Error while saving file. Please try again.
```

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

```
Error, this command can not be called right now.
```

execute

Kiírja, hogy a fájl megtalálható volt-e vagy sem.

Egy lehetséges kimenet sikeres parancs beolvasás esetén:

```
File 'parancsok' successfully loaded.
```

Egy lehetséges kimenet hibás fájlnev esetén:

```
Error loading file 'ezvalamirossz', file not found. Please try again.
```

Egy lehetséges kimenet fájlnev hiánya esetén:

```
Error, file name not specified. Please try again.
```

Egy lehetséges kimenet rossz helybeni híváskor:

```
Error, this command can not be called right now.
```

addplayer

Kiírja, hogy a játékos hozzáadása sikeres volt-e, vagy sem. Hiányzó név esetében kiírja, hogy a játékos hozzáadása a parancs helytelen paraméterezése miatt nem sikerült. Sikeres játékos hozzáadása esetén kiírja az újonnan hozzáadott játékos nevét. Hogyha a felhasználó a játék elindítása után próbálja meg a parancsot elsütni, akkor a szoftver szöveges üzenetben közli a felhasználóval, hogy ez nem lehetséges.

Egy lehetséges kimenet sikeres játékos hozzáadása esetén:

Player 'János' has been successfully added to the game.

Egy lehetséges hibás kimenet ha már ilyen nevű játékos létezik:

Error, a player named 'Pista' already exists in the game. Please try again.

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

Error, this command can not be called right now.

removeplayer

Kiírja, hogy sikeres volt-e a játékos eltávolítása. Hogyha sikeres volt kiírja az eltávolított játékos nevét. Hiányzó játékosnév esetén kiírja, hogy a játékos eltávolítása a parancs helytelen paraméterezése miatt nem sikerült. Helytelen játékosnév megadása esetén kiírja, hogy a játékos eltávolítása azért nem sikerült, mivel ilyen nevű játékos nem létezik. Hogyha a felhasználó a játék elindítása után próbálja meg a parancsot elsütni, akkor a szoftver szöveges üzenetben közli a felhasználóval, hogy ez nem lehetséges.

Egy lehetséges kimenet sikeres eltávolítás során:

Player named 'Pista' has been successfully removed from the game.

Egy lehetséges sikertelen kimenet, ha nincsen ilyen nevű játékos:

Error, there is no player named 'Imre' in the game. Please try again.

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

Error, this command can not be called right now.

showplayers

Kiírja, hogy melyik játékosok vannak hozzáadva a játékhoz, kilistázza őket a nevükkel. Mivel csak a játék elindítása előtt lehet a parancsot kiadni, ezért ilyen esetben is közli ezt a felhasználóval.

Egy lehetséges sikeres kimenet a játék elindítása előtt:

```
Listing all players who are added to the game:  
-János  
-Pista  
-Elemér
```

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

```
Error, this command can not be called right now.
```

start

Kiírja, hogy a játék elindítása sikeres volt-e. Hogyha sikeres, kilistázza a játékosok neveit. Hogyha kevesebb mint kettő hozzáadott játékos van, akkor kiírja, hogy a játék kezdetéhez legalább kettő játékosra van szükség. Hogyha a felhasználó a játék elindítása után próbálja meg a parancsot elsütni, akkor a szoftver szöveges üzenetben közli a felhasználóval, hogy ez nem lehetséges.

Egy lehetséges sikeres kimenet játék elindításakor legalább kettő játékosal:

```
The game has been successfully started with the following players:  
-János  
-Pista
```

Egy lehetséges kimenet játék elindításakor, ha nincs legalább kettő játékos:

```
Error while starting the game, there has to be at least two players to  
start the game. Please add more players, and try again.
```

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

```
Error, this command can not be called right now.
```

show

Kilistázza a kért adatokat. Hogyha helytelenül lett paraméterezve a parancs, akkor kiírja, hogy a parancs futása a helytelen paraméterek megadása miatt nem sikerült. Hogyha a listázás sikeres, akkor kiírja a választott módot, és hogy éppen melyik adatokat listázza ki. Hogyha a felhasználó a játék elindítása előtt próbálja meg a parancsot elsütni, akkor a szoftver szöveges üzenetben közli a felhasználóval, hogy ez nem lehetséges. Ezenfelül mivel egyszerre csak egy kapcsolót lehet használni, ezért ha valaki ilyennel próbálkozik, akkor a szoftver azt hibaüzenetben közli.

Egy lehetséges helyes kimenet a játékos hátizsákja tartalmának listázására:

Listing inventory contents for player "János"

Inventory:

- Beer: #195
- Rag: #392
- Mask: #942
- Transistor: #723

Egy lehetséges helyes kimenet a játékos szobájának ajtajainak listázására:

Listing all doors for current room(Room #123) of player "János"

Doors:

- Door #3: leading to Room #134
- Door #17: leading to Room #35

Egy lehetséges helyes kimenet a játékos szobájában tartózkodó entitások listázására:

Listing all Entities in current room(Room #123) of player "János"

Entities:

- János: Student
- Jóska: Student
- Béla: Teacher

Egy lehetséges helyes kimenet a játékos szobájában lévő tárgyak listázására:

Listing all Items in current room(Room#123) of player "János"

Items:

- Mask: #198
- Transistor: #341
- Beer: #832

Egy lehetséges helyes kimenet a játékos teleportálási lehetőségeinek kilistázására:

Listing all teleportation possibilities for player "János"

Rooms:

- Room #100

Egy lehetséges hibás kimenet ha több mint egy kapcsolót adunk meg:

Error, you can only use one switch per command.

Egy lehetséges hibás kimenet ha nem adunk meg kapcsolót:

Error, nonexistent switch for this command.

Egy lehetséges hibás kimenet ha nem létező kapcsolót adunk meg:

Error, nonexistent switch for this command.

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

Error, this command can not be called right now.

pickup

Amennyiben kijelölt tárgy sikeresen felvételre került, egy megerősítő üzenet jelenik meg. Minden más esetben a megfelelő hibaüzenet kerül kiírásra.

Egy lehetséges, helyes kimenet sikeres sör felvétel esetén:

```
Picked up Beer #211
```

Egy lehetséges kimenet hiba (tele hátizsák)esetén:

```
Error picking up TVSZ #127. Inventory is full.
```

Egy lehetséges kimenet hiba(ragacsos szoba) esetén:

```
Error picking up TVSZ #127. The room you are trying to pick this item up from is sticky.
```

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

```
Error, this command can not be called right now.
```

drop

Ha egy tárgy sikeresen eldobásra került, egy ezt megerősítő üzenet jelenik meg. Minden más esetben egy megfelelő hibaüzenetet kapunk.

Egy lehetséges, helyes kimenet sikeres sör eldobás esetén:

```
Dropped Beer #186.
```

Egy lehetséges, hibás kimenet ha rossz ID-t adunk meg:

```
Error dropping TVSZ #232. No such item in inventory.
```

Egy lehetséges, hibás kimenet, ha gázos szobában maszkot próbálunk eldobni:

```
Error dropping Mask #123. You are currently in a toxic room.
```

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

```
Error, this command can not be called right now.
```

use

Ha a kijelölt tárgy sikeresen használatra került, egy ezt megerősítő üzenet jelenik meg. Minden más esetben hibaüzenet íródik ki a konzolon.

Egy lehetséges, helyes kimenet sikeres sör használat esetén:

Used Beer #111.

Egy lehetséges, hibás kimenet ha már használt tárgyra hívjuk a parancsot:

Error using Beer #111. It is already being used.

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

Error, this command can not be called right now.

move

Ha a parancs sikeresen végrehajtásra kerül, egy ezt megerősítő üzenet jelenik meg a képernyőn. Minden más esetben hibaüzenet íródik ki a konzolra.

Egy lehetséges helyes kimenet helyben maradás esetén:

Successfully stayed in Room # 111

Egy lehetséges helyes kimenet mozgás esetén:

Successfully moved to room # 123.

Egy lehetséges, hibás kimenet ha tele szobába szeretnénk mozogni:

Error moving to room # 123. Room is already full.

Egy lehetséges, hibás kimenet, ha érvénytelen ajtón keresztül szeretnénk közlekedni:

Error moving to room # 123. Door #66 doesn't exist.

Egy lehetséges, hibás kimenet, ha rossz irányban szeretnénk közlekedni:

Error moving to room # 123. Door #66 cannot be used in this direction.

Egy lehetséges, hibás kimenet, ha egy elátkozott ajtón keresztül szeretnénk közlekedni:

Error moving to room # 123. Door #66 is cursed, and can not be used at this time.

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

Error, this command can not be called right now.

connect

Ha a két tranzisztor összekapcsolása sikeres, egy ezt megerősítő üzenet jelenik meg. Ellenkező esetben hiba íródik ki a konzolra.

Egy lehetséges, helyes kimenet két tranzisztor összekapcsolása esetén:

```
Connected Transistor #64 and Transistor #9
```

Egy hibás kimenet ha helytelen ID-t adunk meg:

```
Error connecting Transistor #53 and Transistor #42. Transistor #42 is not in the inventory.
```

Egy hibás kimenet, ha már összekapcsolt tranzisztort próbálunk összekapcsolni:

```
Error connecting Transistor #53 and Transistor #42. Transistor #44 is already connected.
```

Egy hibás kimenet, ha nem megfelelő tárgyakat próbálunk összekapcsolni:

```
Error connecting Transistor #53 and TVSZ #76. These items cannot be connected.
```

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

```
Error, this command can not be called right now.
```

disconnect

Ha a két tranzisztor szétválasztása sikeres, egy ezt megerősítő üzenet jelenik meg. Ellenkező esetben hiba íródik ki a konzolra.

Egy példa a helyes kimenetre két összekapcsolt tranzisztor szétválasztása esetén:

```
Successfully disconnected Transistor #32 and Transistor #27.
```

Egy példa hibás kimenetre, ha nem összekötött tárgyakat próbálunk szétválasztani:

```
Error disconnecting Transistor #6 and Beer #9. They are not connected.
```

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

```
Error, this command can not be called right now.
```

teleport

Ha a teleportálás sikeres, egy megerősítő üzenet jelenik meg a konzolon. Ellenkező esetben egy hiba jelenik meg.

Egy példa helyes kimenetre helyes teleporttal:

```
Successfully teleported using Transistor #219. Teleport destination: Room #888
```

Egy példa helytelen kimenetre ha tele szobába próbálunk teleportálni:

```
Error teleporting using Transistor #2. Remote location (Room #14) is full.
```

Egy példa helytelen kimenetre, ha más által elhelyezett tranzisztorral próbálunk teleportálni:

```
Error teleporting using Transistor #32. This transistor is placed by another player.
```

Egy példa helytelen kimenetre, ha a feltételeknek nem megfelelő tárggyal próbál a felhasználó teleportálni:

```
Error teleporting using Rag #32.This item cannot be used to teleport.
```

Egy lehetséges hibás kimenet rossz időpontbeli hívás esetén:

```
Error, this command can not be called right now.
```

7.2 Összes részletes use-case

| Use-case neve | Debug |
|-------------------------|---|
| Rövid leírás | A játék egyes változóinak értékeinek megtekintéshez a felhasználó megadja az objektumok nevét amiket ki akar írni. |
| Aktorok | Player |
| Forgatókönyv | A felhasználó úgy dönt, hogy a játék belső állapotára kíváncsi, ezért meghívja a <i>debug room11 beer2</i> parancsot, majd ezek az objektumok kiíródnak a debug.txt kimeneti file-ba. |
| Alternatív forgatókönyv | A felhasználó úgy dönt, hogy a játék belső állapotára kíváncsi, ezért meghívja a <i>debug room11 beer2</i> parancsot. A játékban épp nem létezik <i>beer2</i> változó, ezért hibaüzenetet kap, nem történik kiírás. |

| Use-case neve | Load |
|-------------------------|--|
| Rövid leírás | A játék betöltéséhez a felhasználó a 'load' parancsot használhatja. A parancs mellé meg kell adni egy fájlnevet is, mely megmondja hogy mi a betölteni kívánt fájl. |
| Aktorok | Player |
| Forgatókönyv | János a játék elindítása előtt úgy dönt, hogy szeretné betölteni egy régebbi játékmenetének állását. Ehhez a 'load mentes' parancsot adja ki. Ilyen fájl létezik, ezért a parancs sikeres, és János onnan folytathatja a játékot ahol előzőleg abbahagyta. |
| Alternatív forgatókönyv | János a játék elindítása előtt úgy dönt, hogy szeretné betölteni egy régebbi játékmenetének állását. Ehhez a 'load asd123' parancsot adja ki. Ilyen fájl nem létezik, ezért a parancs sikertelen. Ezt a szoftver szöveges üzenetben jelzi Jánosnak, aki újra megpróbálhatja egy régebbi játékmenet betöltését. |

| Use-case neve | Save |
|-------------------------|--|
| Rövid leírás | A játék elmentéséhez a felhasználó a 'save' parancsot használhatja. A parancs mellé meg lehet adni egy fájlnevet is, ami megadja hogy milyen neve lesz a mentési fájlnak. |
| Aktorok | Player |
| Forgatókönyv | János a játék közben úgy dönt, hogy egy ideig be szeretné fejezni a játékot, de nem szeretné elveszteni a jelenlegi állást. Ehhez kiadja a következő parancsot: 'save'. Mivel nem adott meg fájlnevet, ezért az alap 'mentes' nevű fájlba mentődik el. |
| Alternatív forgatókönyv | János a játék közben úgy dönt, hogy egy ideig be szeretné fejezni a játékot, de nem szeretné elveszteni a jelenlegi állást. Ehhez kiadja a következő parancsot: 'save jatekmenet2'. Mivel megadott egy fájlnevet, ezért a 'jatekmenet2' nevű fájlba mentődik el a jelenlegi állás. |

| Use-case neve | Add player |
|-------------------------|--|
| Rövid leírás | A felhasználó hozzá szeretné adni az egyedi névvel rendelkező diák szerepkörű karakterét a játékhoz. |
| Aktorok | Player |
| Forgatókönyv | János a játék indítása előtt hozzá szeretné adni a saját diák karakterét a játékhoz "János" névvel. Mivel még nincs ilyen regisztrált karakter a hozzáadás sikeres, János az alábbi üzenetet látja megjelenni a parancs beírása után: <i>Player 'János' has been successfully added to the game.</i> |
| Alternatív forgatókönyv | János a játék indítása előtt hozzá szeretné adni a saját diák karakterét a játékhoz "János" névvel. Azonban előtte egy másik felhasználó már létrehozott egy karaktert ezzel megegyező névvel, de ezt János nem tudta. Emiatt a hozzáadás sikertelen lesz és az alábbi üzenet jelenik meg: <i>Error, a player named 'János' already exists in the game. Please try again.</i> |
| Alternatív forgatókönyv | János későn érkezett meg, a társai már elkezdtek játszani a játékkal. János kicsit elszomorodott, de az az ötlete támadt, hogy miért is ne próbálná meg hozzáadni saját karakterét a játék közben. Próbálkozása azonban kudarcba fullad, játék közben nem adható hozzá új karakter, az alábbi üzenet jelenik meg a parancs kiadása után: <i>Error, this command can not be called right now.</i> |

| | |
|--------------------------------|---|
| Use-case neve | Remove player |
| Rövid leírás | A felhasználó el szeretne távolítani egy már hozzáadott karaktert a játékból. |
| Aktorok | Player |
| Forgatókönyv | János már hozzáadta saját karakterét a játékhoz. Még a játék elindítása előtt, miközben a többi társa adja hozzá saját karakterét a játékhoz hirtelen eszébe jut, hogy mégsem azzal a névvel/egyáltalán nem szeretne játszani a játékkal. Ezért törli azt a karaktert a játékból. Ez sikeresen megtörténik, ennek nyugtázására az alábbi üzenet íródik ki: <i>Player named 'János' has been successfully removed from the game.</i> |
| Alternatív forgatókönyv | János még a játék elindítása előtt szeretné eltávolítani a hozzáadott karakterét. Nem tudta azonban hogy társai az előző menetet már lejátszották miközben ő a Bolzano-Weierstrass tétel bizonyításával volt elfoglalva. Emiatt a karakter nem létezik, így az eltávolítása nem lehetséges. Az alábbi üzenet érkezik válaszul erre: <i>Error, there is no player named 'János' in the game. Please try again.</i> |
| Alternatív forgatókönyv | Jánost játék közben felhívta a nagymamája, hogy készített neki húslevest. János ennek nagyon megörült, indult is volna ebédelni, de előtte szeretné a saját karakterét törölni a játékból. Ezt megpróbálja, de nem sikerül neki, hiszen már futó játékból regisztrált karaktert törölni nem lehet. Próbálkozását az alábbi üzenet követi: <i>Error, this command can not be called right now.</i> |

| | |
|--------------------------------|---|
| Use-case neve | Játék elindítása |
| Rövid leírás | A játék indítása. Játékosok hozzáadása után a felhasználó megpróbálja elindítani a játékmenetet a 'start' parancs segítségével. |
| Aktorok | Player |
| Forgatókönyv | A játék elindítása előtt János csak saját magát adja hozzá a játékhoz az 'addplayer' parancssal. Ezután megpróbálja elindítani a játékot a 'start' parancs használatával. A játék nem indul el mivel a játék elindításához legalább kettő játékosra van szükség. A szoftver ezután szöveges üzenetben tájékoztatja Jánost, hogy a játék indítása sikertelen volt. |
| Alternatív forgatókönyv | A játék elindítása előtt János szól barátjának Pistának is, hogy játszanak egy menetet a kedvenc játékával. Ezután mind a kettőjüket, Jánost és Pistát is hozzáadják a játékhoz az 'addplayer' parancs segítségével. Miután ezt megtették megpróbálják elindítani a játékot a 'start' parancs segítségével. Mivel ketten is vannak akik játsszák a játékot, ezért sikeres a menet elindítása. |

| Use-case neve | Játékos információjának kilistázása |
|--------------------------------|--|
| Rövid leírás | <p>A felhasználó a 'show' parancs, és annak megfelelő kapcsolói segítségével szeretne különböző adatokat kilistázni. Ezt több különböző módon is megteheti. A kapcsolók a következők:</p> <ul style="list-style-type: none"> • -i: A játékosnál lévő összes tárgyat listázza ki. • -d: A játékos tartózkodási szobájának az összes létező ajtaját listázza ki. • -e: A játékos tartózkodási szobájában lévő összes entitást listázza ki. • -ri: A játékos tartózkodási szobájában lévő összes tárgyat listázza ki. • -t: A játékos összes teleportálási lehetőségét listázza ki. |
| Aktorok | Player |
| Forgatókönyv | <p>János a játék közben kíváncsi néhány információra. Ezért használja a show parancsot, hogy többet tudjon meg karakteréről. Az általa elsütött parancs: 'show -i': ez a parancs kilistázza az összes nála lévő tárgyat. A tárgyak mellé az is kiírásra kerül, hogy azoknak mi az egyedi azonosítójuk.</p> |
| Alternatív forgatókönyv | <p>János a játék közben kíváncsi néhány információra. Ezért használja a show parancsot, hogy többet tudjon meg karakteréről. Az általa elsütött parancs: 'show -d': ez a parancs kiírja az összes ajtót, melyen keresztül el tudja hagyni a szobát ahol éppen tartózkodik. Az ajtók mellett kiírja azt is, hogy azok melyik szobákba vezetnek.</p> |
| Alternatív forgatókönyv | <p>János a játék közben kíváncsi néhány információra. Ezért használja a show parancsot, hogy többet tudjon meg karakteréről. Az általa elsütött parancs: 'show -e': ez a parancs kiírja, hogy abban a szobában melyben János éppen tartózkodik melyik entitások vannak. Az entitások neve mellé kiírja azt is, hogy milyen fajták.</p> |
| Alternatív forgatókönyv | <p>János a játék közben kíváncsi néhány információra. Ezért használja a show parancsot, hogy többet tudjon meg karakteréről. Az általa elsütött parancs: 'show -ri': ez a parancs kiírja az összes tárgyat, melyek abban a szobában vannak ahol János éppen tartózkodik. A tárgyak neve mellé kiírja azok egyedi azonosítóját is.</p> |
| Alternatív forgatókönyv | <p>János a játék közben kíváncsi néhány információra. Ezért használja a show parancsot, hogy többet tudjon meg karakteréről. Az általa elsütött parancs: 'show -t': ez a parancs kiírja az összes lehetséges teleportálási lehetőséget, amivel János élhet. Hogyha nincsen egyetlen lehelyezett tranzisztora sem, akkor a szoftver ezt jelzi. Ha van lehelyezett tranzisztora, akkor kilistázásra kerül, hogy melyik szobába tud teleportálni</p> |

| | |
|--------------------------------|---|
| | János. A szobák mellé kiírásra kerül az is, hogy azt melyik egyedi azonosítójú tranzisztoron keresztül teheti ezt meg. |
| Alternatív forgatókönyv | János a játék közben kíváncsi nagyon sok információra. Ezért úgy gondolja, hogy a 'show' parancsa mögé több kapcsolót is ír, remélve hogy így sokmindent meg fog tudni. Sajnálatára ez nem lehetséges, és a parancs nem fog lefutni, helyette a következő hibaüzenetet kapja: <i>Error, you can only use one switch per command.</i> Ahhoz hogy a sok információ birtokába kerülhessen, egyesével kell a 'show' parancs mögé kapcsolókat írnia. |

| | |
|--------------------------------|--|
| Use-case neve | Pick up |
| Rövid leírás | A játékos fel szeretne venni egy, a szobában földön lévő tárgyat. |
| Aktorok | Player, System |
| Forgatókönyv | János a játék közben egy olyan szobában áll, melyben található a földön egy söröskorsó. János táskájában van még egy üres hely. János a következő parancsot üti be a konzolon: <i>"pickup 123"</i> . A parancs hatására a sör a szobából János hátizsákjába kerül, a konzolon pedig értesítés érkezik a sikeres felvételtől. |
| Alternatív forgatókönyv | János a játék közben egy olyan szobában áll, melyben található a földön egy söröskorsó. János táskájában már nincsen több üres hely. János a következő parancsot üti be a konzolon: <i>"pickup 123"</i> . A konzolon hiba jelenik meg utalva arra a tényre, hogy Jánosnál már nem férne el a sör. |
| Alternatív forgatókönyv | János a játék közben egy olyan szobában áll, melyben található a földön egy lehelyezett tranzisztor. János János a következő parancsot üti be a konzolon: <i>"pickup 123"</i> . Mivel a tranzisztor egy másik játékos által elhelyezett aktív példány, Jánosnak nincs hozzáférése, a rendszer ezt egy hibával jelzi. |
| Alternatív forgatókönyv | János a játék közben egy olyan szobában áll, melyben található a földön egy söröskorsó. János táskájában van még egy üres hely. János a következő parancsot üti be a konzolon: <i>"pickup 111"</i> . Mivel 111-es ID-val nem található tárgy a szobában, a rendszer hibaüzenettel értesíti Jánost a konzolon. |
| Alternatív forgatókönyv | János egy ragacsos szobában áll, ahonnan fel szeretne venni egy földön lévő sört. Ehhez beüti a pickup #9 parancsot, a rendszer pedig hibaüzenetet küld, mivel a tárgy nem vehető fel. |

| | |
|--------------------------------|---|
| Use-case neve | Drop |
| Rövid leírás | Tárgy eldobása hátizsákból. |
| Aktorok | Player, System |
| Forgatókönyv | János úgy dönt, hogy el szeretné dobni a 6-os ID-val rendelkező tárgyat a hátizsákjából. Beüti a <i>drop 6</i> parancsot, a tárgy a szobába kerül, a konzolon pedig az eldobás sikerességét jelző üzenet jelenik meg. |
| Alternatív forgatókönyv | János egy gázos szobában áll, egy maszkkal a hátizsákjában. Úgy dönt, hogy eldobja. Beüti a konzolon a megfelelő parancsot, viszont a rendszer hibaüzenetet küld, amely értesíti, hogy mérgező gázzal elárasztott szobában nem lehetséges a maszk eldobása, mivel az tartja eszméleténél a játékost. |
| Alternatív forgatókönyv | János hátizsákjában található egy pár már összekapcsolt tranzisztor, és úgy dönt hirtelen felindulásból, hogy az egyiket el szeretné dobni. Meghívja a <i>drop</i> parancsot megfelelő paraméterezéssel, viszont az eldobás sikertelen, mivel csak nem párosított tranzisztor dobható el. A rendszer egy erre utaló hibaüzenetet jelenít meg a képernyőn. |
| Alternatív forgatókönyv | János úgy dönt, hogy el szeretné dobni a 6-os ID-val rendelkező aktív sört a hátizsákjából. Beüti a <i>drop 6</i> parancsot, a tárgy eltűnik, mivel az aktív sör ha eldobásra kerül, összetörik. A konzolon megjelenik egy üzenet, ami erről a tényről értesíti Jánost. |

| Use-case neve | Tárgy használat |
|--------------------------------|---|
| Rövid leírás | A felhasználó use parancs és id segítségével használni akar egy tárgyat. |
| Aktorok | Player, System |
| Forgatókönyv | A Player beírja a use parancsot, és egy a táskájában lévő tárgynak az id-ját, azonban a tárgynak nincs aktív képessége, ezért <i>“Error using. This item cannot be used.”</i> választ kap a System-től. |
| Alternatív forgatókönyv | A Player beírja a use parancsot azonban olyan id-t ad meg ami nincs a táskájában ezért, <i>“Error using. No ID like this.”</i> választ kap a Systemtől. |
| Alternatív forgatókönyv | A Player beírja a use parancsot és megadja két, nem összekötött tranzisztor id-jait, ekkor <i>“Cannot be used, connect first.”</i> visszajelzést kap a rendszertől. |
| Alternatív forgatókönyv | A Player beírja a use parancsot és megadja két, összekötött tranzisztor id-jait, ekkor <i>“Transistor placed.”</i> visszajelzést kap a rendszertől. |
| Alternatív forgatókönyv | A Player beírja a use parancsot és megadja egy sör id-ját, ekkor <i>“Beer used #(valami id)”</i> visszajelzést kap a rendszertől. |
| Alternatív forgatókönyv | A Player beírja a use parancsot és megadja egy camembert id-ját, ekkor <i>“Camembert used #(valami id)”</i> visszajelzést kap a rendszertől. |
| Alternatív forgatókönyv | A Player beírja a use parancsot és megadja egy légfrissítő id-ját, ekkor <i>“Airfreshener used #(valami id)”</i> visszajelzést kap a rendszertől. |

| | |
|--------------------------------|--|
| Use-case neve | Move |
| Rövid leírás | A játékos a köre utolsó cselekvéseként a szoba változtatásáról dönt. |
| Aktorok | Player |
| Forgatókönyv | sikeres, nem sikeres (tele szoba, egy irányú ajtó, nem létező ajtó a szobában, elátkozott ajtó) János a köre végén úgy döntött, hogy tovább szeretne menni egy másik szobába. Valahonnan megszerezte annak az ajtónak az azonosítóját amelyen keresztül tovább szeretne haladni. Ennek jelen esetben semmi akadálya nem volt, a helyváltoztatás sikerességéről az alábbi üzenet tájékoztatja Jánost: <i>Successfully moved to room # 123.</i> |
| Alternatív forgatókönyv | János a köre végén úgy döntött, hogy tovább szeretne menni egy másik szobába. Valahonnan megszerezte annak az ajtónak az azonosítóját amelyen keresztül tovább szeretne haladni. Azonban valami félrecsúszott az információ megszerzésekor, hiszen olyan azonosítót adott meg, amelyhez nem tartozik a szobában található ajtó. Ebben az esetben a helyváltoztatás sikertelen és erről az alábbi üzenettel értesül János: <i>Error moving to room # 123. Door #66 doesn't exist.</i> |
| Alternatív forgatókönyv | János a köre végén úgy döntött, hogy tovább szeretne menni egy másik szobába. Valahonnan megszerezte annak az ajtónak az azonosítóját amelyen keresztül tovább szeretne haladni. Sajnos azonban miközben János próbálná kinyitni az ajtót az hirtelen eltűnik, elátkozott volt, ezért a szobaváltás nem sikeres. Ezt az eseményt az alábbi üzenet jelzi: <i>Error moving to room # 123. Door #66 is cursed, and can not be used at this time.</i> |
| Alternatív forgatókönyv | János a köre végén úgy döntött, hogy tovább szeretne menni egy másik szobába. Valahonnan megszerezte annak az ajtónak az azonosítóját amelyen keresztül tovább szeretne haladni. Sajnos azonban mikor János kinyitotta az ajtót csodálkozva látta, hogy a folyosó egy hatalmas csúszós rámpává változott. Ezen a másik irányból le tud csúszni, de visszafelé mászni nem tud, az ajtó áthaladás szempontjából egyirányú folyosóra vezet. Tehát erre nem tud tovább menni. Erről a játékos az alábbi üzenettel értesül: <i>Error moving to room # 123. Door #66 cannot be used in this direction.</i> |
| Alternatív forgatókönyv | János a köre végén úgy döntött, hogy tovább szeretne menni egy másik szobába. Valahonnan megszerezte annak az ajtónak az azonosítóját amelyen keresztül tovább szeretne haladni. Meglepődve tapasztalja, hogy az ajtó túloldalán lévő szoba tele van és már nem fér be. Szomorúan vissza kell forduljon. Erről az alábbi üzenet |

| | |
|--------------------------------|--|
| | értesíti: <i>Error moving to room # 123. Room is already full.</i> |
| Alternatív forgatókönyv | János annyira lelkes, hogy már a játék indítása előtt beírja a parancsot egy kitalált ajtó azonosítóval. Természetesen ekkor ennek nincs hatása, valamint erre az alábbi választ kapja: <i>Error, this command can not be called right now.</i> |
| Alternatív forgatókönyv | János a köre végén úgy döntött, hogy maradni szeretne a jelenlegi szobájában. Ennek semmi akadálya, úgy teheti ezt meg, hogy egyszerűen nem ír ajtó azonosítót a 'move' parancs mellé. Sikereségéről az alábbi üzenet tájékoztatja: <i>"Successfully stayed in Room # 111"</i> |

| | |
|--------------------------------|---|
| Use-case neve | Connect |
| Rövid leírás | A felhasználó a connect parancs, valamint 2 darab ID megadásával össze szeretne kapcsolni két tranzisztort. |
| Aktorok | Player |
| Forgatókönyv | János kiadja a connect parancsot és megadja a két nála levő, pár nélküli Transistor ID-ját. A művelet sikeres, a felhasználó ezt megerősítő üzenetet kap. |
| Alternatív forgatókönyv | János két olyan tranzisztort próbál összekapcsolni, amelyek már össze vannak kapcsolva, így a művelet sikertelen, és erről visszajelzést is kap a rendszertől. |
| Alternatív forgatókönyv | János megadja a két ID-t, azonban legalább az egyik Item nem Transistor. Ekkor az összekapcsolás sikertelen, és ezt igazoló hibaüzenetet is kap. |
| Alternatív forgatókönyv | János megadja a két ID-t, azonban legalább az egyik egy olyan item-é, ami nincs nála, vagy nem létezik. Ekkor sikertelen a művelet, erről a felhasználó üzenetet kap. |

| | |
|--------------------------------|---|
| Use-case neve | Disconnect |
| Rövid leírás | A felhasználó a disconnect parancs, illetve egy ID megadásával szétkapcsolhat két, összekapcsolt Transistort. |
| Aktorok | Player |
| Forgatókönyv | János kiadja a disconnect parancsot, és megadja az egyik, összekapcsolt Transistor ID-ját. Ekkor a szétkapcsolás sikeresen végbemegy. |
| Alternatív forgatókönyv | János egy olyan Transistor ID-ját adja meg, amely nincs összekapcsolva. Ekkor a művelet sikertelen. |
| Alternatív forgatókönyv | János egy olyan ID-t ad meg, amely olyan Item-hez tartozik, amely vagy nincs nála, vagy nem létezik. Ekkor sikertelen a művelet. |
| Alternatív forgatókönyv | János egy olyan Item ID-ját adja meg, amely nem Transistor. Ekkor sikertelen a művelet. |

| | |
|--------------------------------|---|
| Use-case neve | Teleportálás |
| Rövid leírás | A játékos a teleport parancs segítségével teleportálni próbál. |
| Aktorok | Player, System |
| Forgatókönyv | A játékos beírja a teleport parancsot és egy helyes itemID-t (tranzisztoré és az le is lett előtte helyezve rendesen). |
| Alternatív forgatókönyv | A játékos beírja a teleport parancsot azonban nem tranzisztor tárgy ID-ját adja meg. |
| Alternatív forgatókönyv | A játékos beírja a teleport parancsot és egy tranzisztor tárgy ID-ját adja meg, azonban ez a tranzisztor még nincs összekapcsolva másikkal. |

| | |
|--------------------------------|---|
| Use-case neve | Hozzáadott játékosok listázása |
| Rövid leírás | A felhasználó a játék kezdete előtt ellenőrizni szeretné a hozzáadott játékosok számát, és az ő nevüket. Ehhez a 'showplayers' commandot tudja használni. |
| Aktorok | Player |
| Forgatókönyv | János a játék elindítása előtt ellenőrizni szeretné, hogy minden barátját hozzáadta-e a játékhoz, és hogy nekik helyesen lett megadva a nevük. Ehhez kiadja a 'showplayers' parancsot. Mivel még nem lett elindítva a játék, ezért ez a parancs sikeresen lefut, és listába szedi az összes eddig hozzáadott játékost. Így János és barátai biztosra mehetnek, hogy helyes nevekkal tudják elindítani a játékot |
| Alternatív forgatókönyv | János annyira lelkes, hogy a játék elindítása után is le szeretné ellenőrizni, hogy megfelelő neveket adott barátainak. Ezért megpróbálja használni a 'showplayers' parancsot, azt remélve hogy ez működni fog. Szerencsétlenségére ezt a parancsot csak a játék elindítása előtt lehet kiadni, ezért ez nem fog lefutni, és ezt hibaüzenet segítségével közli is a szoftver. |

| | |
|----------------------|---|
| Use-case neve | Diák körváltás |
| Rövid leírás | Miután a egy játékos mozgott (amennyiben egy helyben akar maradni az is annak számít), a System kiválasztja következő játékos és ő kapja meg az irányítást. |
| Aktorok | Player, System |
| Forgatókönyv | Az 1. számú játékos tovább lép egy másik szobába sikeresen. Ezután a System átadja a 2. játékosnak az irányítást. |

| | |
|----------------------|---|
| Use-case neve | Tanárok és takarítók köre (botok) |
| Rövid leírás | Miután mozogtak a diákok, a tanár és a takarítók is "lejátsszák" saját körüket. |
| Aktorok | System |
| Forgatókönyv | Miután 2. játékos befejezte a körét (azaz lépett), nincs több diák így a tanárok jönnek akik ugyanúgy egyenként lépéssel fejezik be a körük. Ezután a takarítók ugyanilyen módon. |

| | |
|----------------------|--|
| Use-case neve | Forduló váltás |
| Rövid leírás | Minden diák, tanár és takarító lépett. Ezután jönnek a térképet változtató algoritmusok. |
| Aktorok | System |
| Forgatókönyv | Miután a takarítók is mind léptek a System kiválasztja melyik szobákat szeretné összeolvasztani/kettéválasztani vagy elátkozni. Majd ezeket lefolytatja és újra a diákok jönnek. |

| | |
|--------------------------------|---|
| Use-case neve | Execute |
| Rövid leírás | Az 'execute' parancs segítségével betöltünk egy fájlt, amelyben a bemeneti nyelvnek megfelelő parancsok vannak egymás után felsorolva, majd ezek betöltése után ezen parancsok összessége meghívódik egymás után. |
| Aktorok | System, Player |
| Forgatókönyv | János előre felírta magának, hogy milyen parancsokat szeretne majd beírni a játék során. Ezeket beírta egy szöveges dokumentumba. A parancs használatával és a fájlnev megadásával a játék beolvassa ezeket a parancsokat és sorban végrehajtja. |
| Alternatív forgatókönyv | János előre felírta magának, hogy milyen parancsokat szeretne majd beírni a játék során. Ezeket beírta egy szöveges dokumentumba. A parancs használatával és egy rossz fájlnev megadásával a játék megpróbálná beolvasni a fájlt, de ez nem létezik. Erről hibaüzenetben tájékoztatja Jánost. |

7.3 Tesztelési terv

1.

| | |
|------------------------|---|
| Teszt-eset neve | Player hozzáadása sikeres |
| Rövid leírás | A felhasználó felvesz a játékba egy új játékost az <i>addplayer</i> paranccsal. A beütött parancs: <i>addplayer testP</i> . Ezek után a <i>showplayers</i> parancs kimenetelén meg kell jelennie <i>testP</i> -nek. |
| Teszt célja | Ellenőrzi, hogy valóban felvehető-e új, egyedi nevű játékos a játékba. |

2.

| | |
|------------------------|--|
| Teszt-eset neve | Player hozzáadása, sikertelen |
| Rövid leírás | A felhasználó hozzáad egy új játékost az <i>addplayer</i> paranccsal. Ezek után még egyszer megpróbál hozzáadni egy ugyanolyan nevű játékost. Az első hozzáadás sikeres, a második sikertelen. A beütött parancsok: <i>-addplayer testP</i> <i>-addplayer testP</i> |
| Teszt célja | Ellenőrzi, hogy valóban nem lehet-e e kettő azonos nevű játékost hozzáadni a játékhoz |

3.

| | |
|------------------------|--|
| Teszt-eset neve | Player eltávolítása, sikeres |
| Rövid leírás | A felhasználó hozzáad egy játékost az <i>addplayer</i> paranccsal, majd eltávolítja a <i>removeplayer</i> -el. Ezek után ellenőrzi, hogy valóban nincs játékos a játékban. A beütött parancsok: - <i>addplayer testP</i> - <i>removeplayer testP</i> - <i>showplayers</i> |
| Teszt célja | Ellenőrzi, hogy a játékban levő játékosok valóban eltávolíthatók-e. |

4.

| | |
|------------------------|---|
| Teszt-eset neve | Player eltávolítása sikertelen |
| Rövid leírás | A felhasználó felvesz a játékba egy játékost az <i>addplayer</i> paranccsal, majd a <i>removeplayer</i> parancsot meghívja egy nem létező játékos nevére. A használt parancsok: - <i>addplayer testP</i> - <i>removeplayer wrongInputTest</i> |
| Teszt célja | Ellenőrzi, hogy valóban csak olyan játékosok távolíthatók el, akik léteznek. |

5.

| | |
|------------------------|---|
| Teszt-eset neve | Start sikeres. |
| Rövid leírás | A felhasználó felvesz a játékba kettő játékost, majd elindítja a játékot. Ez sikeresen megtörténik. A futtatott parancsok: <ul style="list-style-type: none"> - <i>addplayer p1</i> - <i>addplayer p2</i> - <i>start</i> |
| Teszt célja | Ellenőrzi, hogy a játék elindítható-e, amennyiben fennállnak a szükséges feltételek. |

6.

| | |
|------------------------|---|
| Teszt-eset neve | Start sikertelen (nincs elég játékos) |
| Rövid leírás | A felhasználó hozzáad egy játékost, majd elindítja a játékot. Mivel a játékhoz legalább kettő játékos kell, ezért ez sikertelen. A felhasznált parancsok: <ul style="list-style-type: none"> - <i>addPlayer p1</i> - <i>start</i> |
| Teszt célja | A teszt ellenőrzi azon szabály betartását, mely szerint legalább kettő hozzáadott játékos szükséges egy új játék indításához. |

7.

| | |
|------------------------|--|
| Teszt-eset neve | Load sikeres |
| Rövid leírás | A felhasználó elindít egy játékot, majd elmenti. Ezek után újraindítja a programot, és a load paranccsal visszatölti a mentett állást. A felhasznált parancsok: <ul style="list-style-type: none"> - <i>addplayer p1</i> - <i>addplayer p2</i> - <i>start</i> - <i>save testSave</i> <i>[program újraindítása]</i> - <i>load testSave</i> |
| Teszt célja | A teszt ellenőrzi, hogy a játékállás egy mentésből helyesen visszatölthető-e |

8.

| | |
|------------------------|--|
| Teszt-eset neve | Load sikertelen (Nem létezik ilyen file) |
| Rövid leírás | <p>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.</p> <p>A felhasznált parancsok:</p> <ul style="list-style-type: none"> - <i>addplayer p1</i> - <i>addplayer p2</i> - <i>start</i> - <i>save testSave</i> <i>[program újraindítása]</i> - <i>load wrongInputTest</i> |
| Teszt célja | A teszt ellenőrzi, hogy helytelenül megadott mentést nem lehet visszatölteni, a program helyesen kezeli a rossz utasítást. |

9.

| | |
|------------------------|--|
| Teszt-eset neve | Save sikeres |
| Rövid leírás | <p>A felhasználó elindít egy játékot, majd elmenti.</p> <p>A felhasznált parancsok:</p> <ul style="list-style-type: none"> - <i>addplayer p1</i> - <i>addplayer p2</i> - <i>start</i> - <i>save testSave</i> |
| Teszt célja | A teszt ellenőrzi, hogy a játékállás elmentésre kerül-e, és hogy ez helyesen történik-e meg. |

10.

| | |
|------------------------|--|
| Teszt-eset neve | Roszkor hívott command |
| Rövid leírás | <p>A felhasználó elindít egy játékot, majd megpróbál hozzáadni egy új játékost. Mivel a játék már fut, ez sikertelen. A felhasznált parancsok:</p> <ul style="list-style-type: none"> - <i>addplayer p1</i> - <i>addplayer p2</i> - <i>start</i> - <i>addplayer p3</i> |
| Teszt célja | A teszt ellenőrzi, hogy egyes parancsok csak a megfelelő helyzetben hívhatók-e. |

11.

| | |
|------------------------|--|
| Teszt-eset neve | Show -i sikeres |
| Rövid 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. A felhasznált parancsok: <i>addplayer p1</i> <i>addplayer p2</i> <i>start</i> <i>pickup 999</i> <i>show -i</i> |
| Teszt célja | A teszt ellenőrzi, hogy a felhasználó valóban képes-e lekérdezni a hátizsákja tartalmát. |

12.

| | |
|------------------------|--|
| Teszt-eset neve | Show -d sikeres |
| Rövid leírás | A felhasználó elindít egy játékot, majd lekérdezi a szobájából nyíló ajtókat. Ez sikeresen megtörténik. A felhasznált parancsok: <i>addplayer p1</i> <i>addplayer p2</i> <i>start</i> <i>show -d</i> |
| Teszt célja | A teszt ellenőrzi, hogy a felhasználó valóban képes-e megjeleníteni az ajtókat, amiken keresztül elhagyhatja a tartózkodási helyét, amennyiben járható a megfelelő irányban. |

13.

| | |
|------------------------|---|
| Teszt-eset neve | Show -e sikeres |
| Rövid leírás | A felhasználó elindít egy játékot, egy másik játékosal egy szobába mennek. Ezután a másik játékos meghívja a show -e parancsot. <i>addplayer p1</i> <i>addplayer p2</i> <i>start</i> <i>move 111</i> <i>[player 2]</i> <i>move 111</i> <i>[player 1]</i> <i>show -d</i> |
| Teszt célja | A teszt ellenőrzi, hogy a játékos valóban képes-e megjeleníteni a vele egy szobában tartózkodó entitásokat. |

14.

| | |
|------------------------|--|
| Teszt-eset neve | Show -ri sikeres |
| Rövid leírás | Egy diák tartózkodik egy olyan szobában, ahol a földön található egy Sör és egy TVSZ. A felhasználó beüti a <i>show -ri</i> parancsot, a tárgyak pedig megjelennek a konzolon. |
| Teszt célja | A teszt ellenőrzi, hogy a felhasználó képes-e megjeleníteni a tartózkodási helyén földön található tárgyakat. |

15.

| | |
|------------------------|---|
| Teszt-eset neve | Show -t sikeres |
| Rövid leírás | Egy diák tartózkodik egy szobában, ahol egy általa elhelyezett, aktív tranzistor is megtalálható. Ekkor meghívja a <i>show -t</i> parancsot, amely sikeresen kiírja a tranzistor által nyújtott teleportálási lehetőséget. |
| Teszt célja | A teszt ellenőrzi, hogy a játékos valóban képes az általa elhelyezett, teleportálásra alkalmas tranzistor különleges képességét használni. (Ehhez mindenekelőtt látnia kell, hogy a tárgy le van helyezve, és üzemkész, erre való a parancs.) |

16.

| | |
|------------------------|---|
| Teszt-eset neve | Command után rossz kapcsoló |
| Rövid leírás | A játékos meghívja a <i>show -x</i> parancsot. Az <i>x</i> egy érvénytelen kapcsoló, hibaüzenetet ad a rendszer. |
| Teszt célja | A teszt ellenőrizni, hogy helytelen kapcsolókkal ellátott parancsok esetén is helyesen fusson a program, és tájékoztassa a felhasználót a hibás kérésről. |

17.

| | |
|------------------------|---|
| Teszt-eset neve | Pick up teszt eset sikeres felvétel |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy a tárgyfelvétel üres hátizsák és nem ragacsos szoba esetén helyesen működik-e. |

18.

| | |
|------------------------|--|
| Teszt-eset neve | Pickup tesztetet sikertelen felvétel (teli hátizsák) |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy tele hátizsák és nem ragacsos szoba esetén a tárgyfelvétel nem megy végbe. |

19.

| | |
|------------------------|---|
| Teszt-eset neve | Pickup tesztetet sikertelen felvétel (nincs ilyen tárgy a szobában) |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy nem lehet olyan tárgyat felvenni, amely nincsen a szoba tárgyai között. |

20.

| | |
|------------------------|--|
| Teszt-eset neve | Pickup tesztetet sikertelen felvétel (a szoba ragacsos) |
| Rövid leírás | Egyetlen diák tartózkodik a pálya egyik szobájában. A szobában megtalálható egy TVSZ tárgy. A szoba ragacsos, így a benne lelhető tárgyakat nem lehet felvenni. A diák hátizsákja üres. Kiválasztja 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. |
| Teszt célja | Ellenőrizni, hogy ragacsos szobából nem lehet tárgyat felvenni. |

21.

| | |
|------------------------|---|
| Teszt-eset neve | Pickup teszt eset sikertelen felvétel (elhelyezett tranzisztort másé) |
| Rövid leírás | Egyetlen diák tartózkodik a pálya egyik szobájában. A szobában megtalálható egy Tranzisztort 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 Tranzisztort 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. |
| Teszt célja | Ellenőrizni, hogy egy játékos által lehelyezett tranzisztort más nem veheti fel. |

22.

| | |
|------------------------|---|
| Teszt-eset neve | Drop teszt eset sikeres |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy a tárgy eldobása helyesen működik. |

23.

| | |
|------------------------|--|
| Teszt-eset neve | Drop teszt eset sikertelen (nincsen ilyen tárgy az inventoryban) |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy olyan tárgyat nem lehet eldobni, amely nincs benne a hátizsákban. |

24.

| | |
|------------------------|---|
| Teszt-eset neve | Drop teszt eset sikertelen (maszk gázos szobában) |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy gázos szobában maszk eldobására nincs lehetőség. |

25.

| | |
|------------------------|--|
| Teszt-eset neve | Drop teszt eset sikeres (aktív sör) |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy aktív sör eldobása esetén a tárgy nem kerül be a szoba tárgyai közé. |

26.

| | |
|------------------------|---|
| Teszt-eset neve | Drop teszt eset sikertelen (összekapcsolt tranzisztor eldobása) |
| Rövid 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 szobában 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. |
| Teszt célja | Ellenőrizni, hogy összekapcsolt Tranzisztort nem lehet eldobni. |

27.

| | |
|------------------------|--|
| Teszt-eset neve | Use sikeres(Tranzisztor) |
| Rövid leírás | Egy diáknál van két, összekapcsolt tranzisztor. Az egyiket lerakja abban a szobában, ahol tartózkodik. |
| Teszt célja | Ellenőrizni, hogy a lerakott tranzisztor valóban lekerült-e a szobában. |

28.

| | |
|------------------------|--|
| Teszt-eset neve | Use sikeres(Camembert) |
| Rövid leírás | Egy diáknál van egy camembert. Felhasználva azt az adott szoba gázos szobává változik. |
| Teszt célja | Ellenőrizni, hogy valóban gázos lett-e a szoba. |

29.

| | |
|------------------------|---|
| Teszt-eset neve | Use sikeres(Sör) |
| Rövid leírás | Egy diáknál van egy nem aktív söröspohár. Ezt használva aktívvá változtatja az eszközt. |
| Teszt célja | Ellenőrizni, hogy aktívvá változott-e a söröspohár. |

30.

| | |
|------------------------|---|
| Teszt-eset neve | Use sikeres(Légfrissítő) |
| Rövid leírás | Egy diáknál van egy légfrissítő. Egy gázos szobában használva azt, a szoba normálissá válik. |
| Teszt célja | Ellenőrizni, hogy az eredetileg gázos szoba normális szobává változott-e, miután a diák használta a légfrissítőt. |

31.

| | |
|------------------------|---|
| Teszt-eset neve | Use sikertelen (nem use-olható tárgy / nincs ilyen) |
| Rövid leírás | Egy diáknál van egy olyan tárgy, amelyet nem lehet használni. Ekkor a tárgy használata hamis értéket ad vissza. |
| Teszt célja | Ellenőrizni, hogy a nem használható tárgyak használatakor valóban nem történik semmi. |

32.

| | |
|------------------------|---|
| Teszt-eset neve | Use sikertelen (tranzisztor not connected) |
| Rövid leírás | Egy diáknál van két tranzisztor, azonban ezek nincsenek összekapcsolva. Ekkor nem lehet egyiket sem lehelyezni. |
| Teszt célja | Ellenőrizni, hogy nem összekapcsolt tranzisztort ne lehessen lehelyezni. |

33.

| | |
|------------------------|---|
| Teszt-eset neve | Use sikertelen (már aktivált sör) |
| Rövid leírás | Egy diáknál van egy már eleve aktivált sör. Ekkor a use parancsra történő aktiválás nem mehet végbe, így sikertelen az aktiválás. |
| Teszt célja | Ellenőrizni, hogy már eleve aktivált sört ne lehessen még egyszer aktiválni. |

34.

| | |
|------------------------|--|
| Teszt-eset neve | Move sikeres |
| Rövid leírás | Egyetlen diák van a pálya egyik szobájában. Kiválaszt a szoba ajtajai közül egyet amin átmegy az annak túloldalán lévő másik szobába. A helyváltoztatás sikerül. |
| Teszt célja | Ellenőrizni hogy egy helyváltoztatás sikeresen meg tud történni. |

35.

| | |
|------------------------|--|
| Teszt-eset neve | Move sikertelen - a célszoba tele van |
| Rövid leírás | Egy diák van a pálya egyik szobájában. Kiválaszt a szoba ajtajai közül egyet amin 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 helyváltoztatás. |
| Teszt célja | Ellenőrizni, hogy tele szobába nem lehet átmenni. |

36.

| | |
|------------------------|--|
| Teszt-eset neve | Move sikertelen - a kiválasztott ajtó elátkozott |
| Rövid 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. |
| Teszt célja | Ellenőrizni hogy az elátkozott ajtón nem lehet keresztülmenni. |

37.

| | |
|------------------------|---|
| Teszt-eset neve | 37:Move sikertelen - az ajtó egyirányú, ebből az irányból nem átjárható |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy az egyirányú ajtó rossz irányból nem biztosít átjárást. |

38.

| | |
|------------------------|--|
| Teszt-eset neve | 38: Move sikertelen - kiválasztott ajtó nem része a szobának |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy csak a szobában lévő ajtókon keresztül lehet közlekedni. |

39.

| | |
|------------------------|--|
| Teszt-eset neve | 39: Move sikeres, player a szobában marad |
| Rövid leírás | Egyetlen diák van a pálya egyik szobájában. Kör végén maradni akar a szobában. A diák a szobában maradt. |
| Teszt célja | Ellenőrizni, hogy a kör végén szobában maradás helyesen hajtódik végre. |

40.

| | |
|------------------------|---|
| Teszt-eset neve | Connect sikeres (kettő még nem connected tranzisztor) |
| Rövid leírás | Egy diáknál van kettő, még nem összekapcsolt tranzisztor. A diák összekapcsolja ezt a két tranzisztort. |
| Teszt célja | Ellenőrizni, hogy a két tranzisztor összekapcsolódott. |

41.

| | |
|------------------------|---|
| Teszt-eset neve | Connect sikertelen (már kapcsolt) |
| Rövid leírás | Egy diáknál van kettő, már összekapcsolt tranzisztor. A diák megpróbálja összekapcsolni ezt a két tranzisztort. |
| Teszt célja | Ellenőrizni, hogy az összekapcsolás sikertelen. |

42.

| | |
|------------------------|--|
| Teszt-eset neve | Connect sikertelen (nem tranzisztor tárgy) |
| Rövid leírás | Egy diáknál van kettő, nem tranzisztor tárgy. A diák megpróbálja összekapcsolni ezt a két tárgyat. A művelet sikertelen, mert csak tranzisztorokat lehet összekapcsolni. |
| Teszt célja | Ellenőrizni, hogy az összekapcsolás sikertelen, hiszen csak tranzisztorokat lehet összekapcsolni. |

43.

| | |
|------------------------|---|
| Teszt-eset neve | Connect sikertelen (nem létező tárgy) |
| Rövid leírás | Egy diák olyan tárgyakat próbál összekapcsolni (ID megadásával), amelyek nem léteznek. A művelet sikertelen, hiszen csak létező tárgyat lehet összekapcsolni. |
| Teszt célja | Ellenőrizni, hogy nem létező tárgyakat nem lehet összekapcsolni. |

44.

| | |
|------------------------|---|
| Teszt-eset neve | Disconnect sikeres (kettő összekapcsolt tranzisztor) |
| Rövid leírás | Egy diáknál van kettő, már összekapcsolt tranzisztor. A diák szétkapcsolja őket. A művelet sikeresen végbemegy. |
| Teszt célja | Ellenőrizni, hogy sikeres volt-e a szétkapcsolás. |

45.

| | |
|------------------------|---|
| Teszt-eset neve | Disconnect sikertelen (kettő nem összekapcsolt tranzisztor) |
| Rövid leírás | Egy diáknál van kettő, nem összekapcsolt tranzisztor. A diák megpróbálja szétkapcsolni ezt a két tranzisztort. A művelet sikertelen, hiszen nincsenek összekapcsolva. |
| Teszt célja | Ellenőrizni, hogy sikertelen, hiszen csak összekapcsolt tranzisztorokat lehet szétkapcsolni. |

46.

| | |
|------------------------|---|
| Teszt-eset neve | Disconnect sikertelen (nem tranzisztor tárgy) |
| Rövid leírás | Egy diáknál van kettő, nem tranzisztor típusú tárgy. A diák megpróbálja szétkapcsolni ezt a két tárgyat. A művelet sikertelen, hiszen csak tranzisztorokat lehet szétkapcsolni. |
| Teszt célja | Ellenőrizni, hogy sikertelen a szétkapcsolás, hiszen csak tranzisztor típusú tárgyakat lehet szétkapcsolni. |

47.

| | |
|------------------------|--|
| Teszt-eset neve | Disconnect sikertelen (a OR b tárgy nem létezik) |
| Rövid leírás | Egy diák megpróbál két olyan tárgyat szétkapcsolni (ID megadásával), amelyek közül az egyik nem létezik. A művelet sikertelen, hiszen csak létező tranzisztorokat lehet szétkapcsolni. |
| Teszt célja | Ellenőrizni, hogy sikertelen a szétkapcsolás, hiszen nem létező tárgyat nem lehet szétkapcsolni. |

48.

| | |
|------------------------|---|
| Teszt-eset neve | Teleport sikeres |
| Rövid leírás | Egy diáknál van egy tranzisztor, amelynek a párja egy másik szobába van lerakva. A diák a nála levő tranzisztor segítségével teleportál a másik szobába. A művelet sikeres. |
| Teszt célja | Ellenőrizni, hogy a diák sikeresen átteleportált a másik szobába. |

49.

| | |
|------------------------|---|
| Teszt-eset neve | Teleport sikertelen (Rossz tárgyra hívódik) |
| Rövid leírás | Egy diáknál van egy nem tranzisztor típusú tárgy, amivel teleportálni próbál. A művelet sikertelen, mivel csak tranzisztorral lehet teleportálni. |
| Teszt célja | Ellenőrizni, hogy sikertelen a teleportálás, hiszen teleportálni csak tranzisztorral lehet. |

50.

| | |
|------------------------|---|
| Teszt-eset neve | Teleport sikertelen (Távoli szoba tele van) |
| Rövid leírás | Egy diáknál van egy tranzisztor, amelynek a párja egy másik szobába van lerakva, azonban ez a szoba tele van. A diák a nála levő tranzisztor segítségével teleportálna a másik szobába. A művelet sikertelen, mivel teli szobába nem lehet bemenni. |
| Teszt célja | Ellenőrizni, hogy sikertelen a teleportálás, hiszen teli szobába nem lehet bemenni, vagy beteleportálni. |

51.

| | |
|------------------------|---|
| Teszt-eset neve | Kettéválasztás sikeres |
| Rövid leírás | Létezik egy üres szoba, ezután a System megpróbálja azt kettéválasztani, két szobára. Sikerrel jár. |
| Teszt célja | Szoba kettéválasztásának helyes végrehajtása, ideális feltételek esetén. |

52.

| | |
|------------------------|--|
| Teszt-eset neve | Kettéválasztás sikertelen (nem üres a szoba) |
| Rövid leírás | Létezik egy üres szoba, benne tartózkodik a diák. A System úgy dönt váljon ketté a szoba, azonban ez megghiúsul. |
| Teszt célja | Szoba kettéválasztásának megghiúsulása, nem ideális feltétel miatt. |

53.

| | |
|------------------------|--|
| Teszt-eset neve | Összeolvasztás sikeres |
| Rövid 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. |
| Teszt célja | Szobák összeolvasztásának helyes lefolyása, ideális feltételek esetén. |

54.

| | |
|------------------------|--|
| Teszt-eset neve | Összeolvasztás sikertelen (nem üres az egyik szoba) |
| Rövid leírás | Létezik két darab üres szoba, az egyikben egy diák tartózkodik. A System úgy dönt összeolvasztja őket, azonban ez sikertelen lesz. |
| Teszt célja | Szobák összeolvasztásának megghiúsulása, nem ideális feltétel miatt. |

55.

| | |
|------------------------|---|
| Teszt-eset neve | Szoba el átkozódik |
| Rövid leírás | Létezik egy szoba, benne egy diák, a kör végén a szoba el átkozódik. A következő körben a diák nem tud kimenni. |
| Teszt célja | El átkozódás tesztelése |

56.

| | |
|------------------------|---|
| Teszt-eset neve | Szoba visszaváltozik az el átkozásból |
| Rövid 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. |
| Teszt célja | Visszaváltozás tesztelése |

57.

| | |
|------------------------|--|
| Teszt-eset neve | Szoba ragacsos lesz |
| Rövid 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. |
| Teszt célja | Ragacsosság működésének tesztelése. |

58.

| | |
|------------------------|--|
| Teszt-eset neve | Show parancs után több mint egy kapcsoló |
| Rövid leírás | A játék elindítása után, az egyik játékos a körében nagyon sok információra kíváncsi. Ezért a következő parancsot adja ki: 'show -i -e' Mivel egy parancs mögé maximum egyetlen kapcsolót lehet csak írni, ezért a következő hibaüzenetet kell, hogy kapja a felhasználó: <i>Error, you can only use one switch per command.</i> |
| Teszt célja | Ellenőrizni, hogy egy parancs mögé csak egyetlen kapcsolót lehet írni. |

59.

| | |
|------------------------|---|
| Teszt-eset neve | Takarító kitisztítja a szobát |
| Rövid leírás | Létezik egy gázos szoba, belép egy takarító és kitisztítja. |
| Teszt célja | Tisztítás sikerességének vizsgálata |

60.

| | |
|------------------------|--|
| Teszt-eset neve | Takarító kiküld a szobából egy diákot. |
| Rövid leírás | Létezik egy szoba, benne egy diákkal. Belép a takarító és kiküldi a szobából. A takarító sikerrel jár. |
| Teszt célja | Szobából kiküldés tesztelése, ideális feltételekkel. |

61.

| | |
|------------------------|--|
| Teszt-eset neve | Takarító kiküld a szobából egy diákot, de az nem tud kimenni. |
| Rövid leírás | Létezik egy szoba, benne egy diákkal. Belép a takarító és kiküldi a szobából. Azonban a diák nem tud másik szobába menni hiszen minden szomszédos tele van. Így a takarító nem jár sikerrel. |
| Teszt célja | Szobából kiküldés megghiúsulásának tesztelése, nem ideális feltétel miatt. |

62.

| | |
|------------------------|---|
| Teszt-eset neve | Takarító kiküld a szobából egy tanárt, de az nem tud kimenni mivel kábult. |
| Rövid leírás | Létezik egy szoba, benne egy diákkal. Belép a takarító és kiküldi a szobából. Azonban a diák nem tud másik szobába menni hiszen kábult. |
| Teszt célja | Szobából kiküldés megghiúsulásának tesztelése, nem ideális feltétel miatt. |

63.

| | |
|------------------------|--|
| Teszt-eset neve | Diák találkozik egy tanárral, megmenekül (TVSZ) |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy a TVSZ tárgy meg tudja menteni a diákot. |

64.

| | |
|------------------------|---|
| Teszt-eset neve | Diák találkozik egy tanárral, megmenekül (Sör) |
| Rövid 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 nem veszít a tartósságából. |
| Teszt célja | Ellenőrizni, hogy az aktív Sör tárgy meg tudja menteni a diákat. |

65.

| | |
|------------------------|---|
| Teszt-eset neve | Diák találkozik egy tanárral nem menekül meg |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy ha diáknál nincs lélekmentő tárgy, a tanár ki tudja szívni a lelkét. |

66.

| | |
|------------------------|--|
| Teszt-eset neve | Tanár megbénul egy rongy által |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy a Rongy tárgy hatása helyesen működik a tanárookra. |

67.

| | |
|------------------------|--|
| Teszt-eset neve | Tanár megbénul gáz által |
| Rövid 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. |
| Teszt célja | Ellenőrizni, hogy a gáz hatása helyesen működik a tanárookra. |

68.

| | |
|------------------------|--|
| Teszt-eset neve | Gáz által bénult tanár rongy által is megénul |
| Rövid 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 gázzosága 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 tárgyát a szobában. Ennek hatására a szobában lévő tanár elkábultsága növekszik. |
| Teszt célja | Ellenőrizni, hogy a tanárookra a gáz és Rongy tárgy által ható elkábulás egyszerre is működik. |

69.

| | |
|------------------------|--|
| Teszt-eset neve | Diák megbénul gáz által |
| Rövid 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 köre 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. |
| Teszt célja | Ellenőrizni, hogy a diákokra a gáz hatása helyesen működik. |

70.

| | |
|------------------------|---|
| Teszt-eset neve | Tanár nem bénul meg maszk miatt |
| Rövid leírás | Létezik egy gázzal teli szoba, belép egy tanár de nem bénul meg mivel van nála maszk. |
| Teszt célja | Maszk működés tesztje. |

71.

| | |
|------------------------|--|
| Teszt-eset neve | Diák nem bénul meg maszk miatt |
| Rövid leírás | Létezik egy gázzal teli szoba, belép egy diák de nem bénul meg mivel van nála maszk. |
| Teszt célja | Maszk működés tesztje. |

72.

| | |
|------------------------|---|
| Teszt-eset neve | Takarítóra nem hat a gázos szoba |
| Rövid leírás | Létezik egy gázzal teli szoba, belép egy takarító de nem bénul meg hanem kiszűti azt. |
| Teszt célja | A gáz hatástalansága a takarítóra. |

73.

| | |
|------------------------|---|
| Teszt-eset neve | Logarléc felvétele |
| Rövid leírás | Létezik egy szoba, benne egy diákkal és egy logarléc tárgyal. A diák felveszi logarlécet ezzel megnyerve a játékot. |
| Teszt célja | Győzelmi feltétel tesztelése. |

74.

| | |
|------------------------|---|
| Teszt-eset neve | Hamis TVSZ-es diák és tanár találkozás |
| Rövid leírás | Létezik egy szoba, benne egy diák, a diák hátizsákjában egy hamis TVSZ. A szobába belép egy tanár, a diákot nem védi meg a TVSZ így meghal. |
| Teszt célja | Hamis TVSZ működési tesztje |

75.

| | |
|------------------------|--|
| Teszt-eset neve | Hamis Maszk, gázos szobában nem véd |
| Rövid 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. |
| Teszt célja | Hamis maszk működési tesztje |

76.

| | |
|------------------------|---|
| Teszt-eset neve | Hamis Logarléc, nem nyeri meg a játékot |
| Rövid leírás | Létezik egy szoba, benne egy diák és egy hamis logarléc. A diák felveszi a hamis logarléctet azonban nem nyeri meg a meccset. |
| Teszt célja | Hamis logarléc működési tesztje |

77.

| | |
|------------------------|---|
| Teszt-eset neve | Minden játékos meghal |
| Rövid leírás | Létezik egy szoba, benne 2 diák (az egész térképen ennyi van). Semmi nincs a hátizsákjukban. Belép egy tanár és mind a kettőt megöli. A játék véget ér. |
| Teszt célja | Vesztési feltétel tesztelése. |

78.

| | |
|------------------------|---|
| Teszt-eset neve | Utolsó kör |
| Rövid leírás | Létezik egy szoba egy diákkal és egy szomszédos szoba. A diák átlép a szobába és nincs ott a logarléc, azonban ez az utolsó kör a játékban. Így a játékot elveszti a felhasználó. |
| Teszt célja | Vesztési feltétel tesztelése. |

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A tesztek végrehajtásához az “execute” parancsot valamint a Java JUnit tesztelő keretrendszert alkalmazzuk.

7.5 Napló

| Kezdet | Időtartam | Résztevők | Leírás |
|----------------------|-----------|---------------------------------------|--|
| 2024.03.28. 14:00 | 2,5 óra | Páter, Miczki, Nagy | Értekezlet. Alapvető változtatások eszközlése a modellen. Néhány részfeladat feljegyzése, uml osztálydiagram kiegészítése |
| 2024.03.29. 9:00 | 2 óra | Páter, Bereczky Miczki, Nagy | Értekezlet. Maszk és Rag működéséhez szükséges új függvények felvétele. Commandok vázának közös megbeszélése. Feladatok kiosztása: Páter: makeNotToxic dokumentáció, légfrissítő dok., throwOut implementáció és dok., rag találkozás működési logika, containsRag dok és imp. Bereczky: pickup,drop,use,move,connect,disconnect és teleport command Horváth: ableToStun dok. és implementáció, addplayer,removeplayer,start,show command Miczki: átdolgozott save dok., canPickUp dok, maszk működési logika, prevent toxic dok és imp., carriedIntoNormalRoom dok és imp. Nagy: átdolgozott pickUpItem dok., teleport/connect új szekvencia diagram Vasárnap újabb meeting |
| 2024.03.31. 9:00 | 1 óra | Páter, Horváth Miczki, | Értekezlet. Eddig elkészült dokumentációk átnézése. |
| 2024.04.01. 8:00 | 2 óra | Páter | makeNotToxic dokumentáció, légfrissítő dok., throwOut implementáció és dok., rag találkozás működési logika, containsRag dok és imp. elkészültek |
| 2024.04.01. 8:00 | 2 óra | Bereczky | pickup,drop,use,move,connect,disconnect és teleport command elkészültek. |
| 2024.04.01. 10:00 | 2 óra | Horváth | ableToStun dok. és implementáció, addplayer,removeplayer,start,show command elkészültek. |
| 2024.04.01. 11:00 | 2 óra | Miczki | átdolgozott save dok., canPickUp dok, maszk működési logika, prevent toxic dok és imp., carriedIntoNormalRoom dok és imp. elkészültek. |

| | | | |
|----------------------|-------|--|--|
| 2024.04.01. 15:00 | 2 óra | Nagy | átdolgozott pickUpItem dok., teleport/connect új szekvencia diagram elkészültek. |
| 2024.04.01. 17:00 | 2 óra | Páter, Bereczky, Horváth, Miczki, Nagy | Értekezlet. Eddig elkészült dokumentációk átnézése. Tesztek felosztása: Páter: 51-62, 71-78 tesztek, tick use-case Bereczky: 1-16 tesztek, szöveges szerializáció Hováth: 17-26, 63-70 tesztek, command egységesítés/kiegészítés Miczki: 34-39, Janitor osztály működési logika Nagy: 27-33, 40-50 teszt, tranzistor kiegészítése |
| 2024.04.02 7:00 | 2 óra | Páter | 51-62, 71-78 tesztek, tick use-case elkészültek |
| 2024.04.02 9:00 | 2 óra | Bereczky | 1-16 tesztek, szöveges szerializáció elkészültek |
| 2024.04.02 11:00 | 2 óra | Horváth | 17-26, 63-70 tesztek, command egységesítés/kiegészítés elkészültek |
| 2024.04.02 11:00 | 2 óra | Miczki | 34-39, Janitor osztály működési logika elkészültek |
| 2024.04.02 19:00 | 2 óra | Nagy | 27-33, 40-50 teszt, tranzistor kiegészítése elkészültek. |
| 2024.04.03 08:00 | 1 óra | Páter, Horváth, Miczki, Nagy | Értekezlet. Eddig elkészült dokumentációk átnézése. |