

8. Részletes tervek

70 – Edgerunners

Konzulens:

Szabó Bence Sándor

Csapattagok

Csabuda Nóra
Gulybán Dániel
Nagy Gergely
Papp Levente
Sánta Dániel

GHDCTP
HUCHEX
HC1QJP
HV0CRH
OGLSI2

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

2024-04-15

8. Részletes tervek

8.0 Hibajavítások

8.1.1 Prototípus interface-definíciója

8.1.1.1 Prototípus interface-definíciója

A program editor módját eltöröltük, ezt követően a *test* módban lehet minden parancsot használni, míg az alap (játék) módban csak *load/save*, *random*, *getinfo*, *list*, valamint az aktív karaktert irányító *move*, *pickup*, *activate*, *drop*, *link* és *endTurn* parancsok lesznek elérhetőek. A *test* mód futtatásával nem fut le automatikusan minden teszteset, mint ahogyan azt korábban definiáltuk, hanem sorszám megadásával lehet kiválasztani, hogy melyik tesztesetet szeretnénk indítani. A tesztek ugyan minden esetben a *getinfo* parancs kimenetével állnak elő, mint eredményfájl, azt érdemes tudni, hogy ehhez nem szükséges (és mi nem is fogjuk) kiadni a tesztek bemeneti fájljaiban ezt a parancsot.

8.1.1.2 Bemeneti nyelv

A parancsok közül úgy ítéltük meg, hogy nem lesz szükség *startGame*, valamint *pauseGame* parancsokra. A parancsok közé felvételre került még egy *setActiveCharacter* [karakterazonosító], valamint *link* [tárgyazonosító] [tárgyazonosító] parancs is. A *setActiveCharacter* csak test módban érhető el, segítségével megmondható, hogy éppen kit szeretnénk irányítani, legyen az játékos vagy akár nemjátszható karakter. A *link* parancs két tárgyat kapcsol össze. Továbbá az *addPlayer* parancs kapott egy új [szobaazonosító] paramétert, amellyel meg tudjuk adni, hogy hova szeretnénk elhelyezni a játékost. Az *addItem* parancs annyiban módosult, hogy második paramétere nem csak szobaazonosítót jelezhet, hanem entitásazonosítót is, így nem csak szobához, hanem játékoshoz is tudunk adni tárgyat tesztelés során. Továbbá eltöröltük a *placelogar* parancsot is, helyette az *addItem Logarlec* [szobaazonosító]-hoz hasonló parancsokat tudjuk használni. Felvételre került még egy *merge* és egy *split* nevű parancs is. Az előbbinek két paramétere, míg az utóbbinak egy paramétere van és ezek értelemszerűen mind szobaazonosítók.

8.1.1.3 Kimeneti nyelv

A *getinfo* paraméter nélküli változatának a kimenetét pontosítottuk. Mivel ez a változat a játékalapotról ad információt, ezért az korábbiakban definiált kimenet elejére fűztünk egy új sort: *Game state: lost/ongoing/won*. Továbbá a karakterek infóinak megjelenítése előtt *Characters of Game:*, valamint a szobák infóinak megjelenítése előtt *Rooms of Game:* sorokat vettünk fel. Ha ezekhez és más hasonló (Items of entity, Items of room, Doors of room, stb.) felsorolásokban nincs egyetlen listázandó objektum sem, akkor a kettőspont után egy szóközzel elválasztva csak annyi szerepel, hogy *No items.* / *No entites.* / *No doors.*, stb.

8.1 *Osztályok és metódusok tervei.*

8.1.1 AirConditioner

- **Felelősség**

Ez a tárgy gázos szobában alkalmazva elveszi a szoba gázosságát.

- **Ősosztályok**

Item

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

- **+void activate(s: Student):** Megnézi, hogy a szülőből származó health attribútum nagyobb-e, mint 0, ha nem nagyobb, akkor nem csinál semmit. Ha nagyobb, akkor a paraméterként kapott Student objektumtól a getRoom() metódussal lekéri a szobáját és ezen a szobán meghívja a detoxicate() metódust. Ezután a health attribútumot eggyel csökkenti.

8.1.2 Beer

- **Felelősség**

A sör egy aktiválással hasznosítható tárgy. Aktiválásával a hallgató védettségére tehát szert (részezzé válással).

- **Ősosztályok**

Item

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

- **+void activate(s: Student):** Megnézi, hogy a szülőből származó health attribútum nagyobb-e, mint 0, ha nem nagyobb, akkor nem csinál semmit. Ha nagyobb, akkor a paraméterként kapott Student objektumon meghívja a makeStudentDrunk() metódust. Ezután a health attribútumot eggyel csökkenti.

8.1.3 Camembert

- **Felelősség**

Ez a tárgy az őt felhasználó hallgató szobáját mérgezővé teszi, aktiválás hatására.

- **Ősosztályok**

Item

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

- **+void activate(s: Student):** Megnézi, hogy a szülőből származó health attribútum nagyobb-e, mint 0, ha nem nagyobb, akkor nem csinál semmit. Ha nagyobb, akkor a paraméterként kapott Student objektumtól a getRoom() metódussal lekéri a szobáját és ezen a szobán meghívja a makeRoomToxic() metódust. Ezután a health attribútumot eggyel csökkenti.

8.1.4 CleaningLady

- **Felelősség**

Ez az osztály egy takarítónőt reprezentál, ha belép egy szobába, minden mozogni képes (nem bénult / ájult) embert kitessékel onnan. Ha gázos szobába lép, kiszellőztet, megszüntetve a szoba gázosságát.

- **Ősosztályok**

Entity

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

- **+void meet(e: Entity):** Felülírja az őosztály meet(e: Entity) függvényét, a kapott entitáson meghívja a meet(e: Entity) függvényt és magát adja meg paraméterül.
- **+bool move(dest: Room, d: Door):** Meghívja az őosztály move(dest: Room, d: Door) függvényét, paraméterként megadja a saját paraméterül kapott objektumjait. Ha ennek a függvénynek a visszatérési értéke hamis, akkor hamissal tér vissza, ha igaz, akkor meghívja a dest szobán a clean() függvényt, majd igazzal visszatér.
- **+bool toxicate(r: Room):** Felülírja az őosztály metódusát, hamissal visszatér.
- **+bool immobilize():** Felülírja az őosztály metódusát, hamissal visszatér.

8.1.5 Controller

- **Felelősség**

Ez az osztály felel a játék játszataáért, a modell irányításáért. A prototípus beadásban a nézet lényegében a szabványos be- és kimenet, a kiíratásért és beolvasásért egyelőre nem láttuk célszerűnek egy *View* osztály bevezetését, így ezeket is jelenleg a *Controller* végzi.

- **Ősosztályok**

Nincsenek.

- **Interfészek**

Nincsenek.

- **Attribútumok**

- **-model: Game:** A játék modellje.

- **Metódusok**

- **+Controller(model: Game):** Ez egy konstruktor. A paraméterben kapott játékmodellt beállítja a *model* attribútumába és ezt fogja irányítani a következőekben.
- **+start():** Elindít egy játékot. A játékot körökre bontva lehet játszani. Egy körben a játékban megtalálható karaktereken lépked végig és hívja meg a *handlePlayerTurn* metódust, ha az aktuális karakter az egy játékos és a *handleProfTurn* és *handleCleaningLadyTurn* metódusokat, ha az egy nemjátszható karakter.
- **+handlePlayerTurn(entity: Entity):** Játékosok körének kezeléséért felel. Egy eseményhurokként funkcionál, várja a szabványos bemeneten a parancsokat és meghívja a megfelelő parancs metódusát, amely szintén a Controllerben van definiálva (ld. alább).
- **+handleProfTurn(entity: Entity):** Professzor karakterek körét kezeli. Egy körben átmozog egy szobába és felvesz a szobában egy tárgyat. A szoba és a tárgy kiválasztása véletlenszerűen működik.
- **+handleCleaningLadyTurn(entity: Entity):** Takarító karakterek körét kezeli. Egy körben átmozog egy szobába. A szoba kiválasztása véletlenszerűen történik.
- **+load(cmd: String[]):** Betölt egy, a paraméterében megkapott játékállást.
- **+save(cmd: String[]):** Kimentí az aktuális játékállást a paraméterében megkapott helyre.
- **+random(cmd: String[]):** Ki-/bekapcsolja a randomicitást.
- **+getinfo(cmd: String[]):** Információt ír ki az entitásokról, szobákról vagy az egész játékállapotról.
- **+list(cmd: String[]):** Kilistázza az entításokat vagy szobákat.
- **+move(cmd: String[]):** Átmozgatja az aktív karaktert a paraméterben megkapott szobába a szintén paraméterben megkapott ajtón keresztül.
- **+pickup(cmd: String[]):** Felveteti a paraméterben megkapott tárgyat az aktív karakterrel.
- **+activate(cmd: String[]):** Aktiváltatja a paraméterben megkapott tárgyat az aktív karakterrel.
- **+drop(cmd: String[]):** Elejteti a paraméterben megkapott tárgyat az aktív karakterrel.
- **+endTurn(cmd: String[]):** Befejezi az aktív karakter körét.

- **+addPlayer(cmd: String[]):** Hozzáad egy játszható karaktert a játékállapothoz a paraméter megkapott egyedi névvel a szintén paraméterben megkapott szobába elhelyezve.
- **+addNPC(cmd: String[]):** Hozzáad egy nemjátszható karaktert a játékállapothoz paraméterben megkapott típussal a szintén paraméterben megkapott szobába elhelyezve.
- **+addRoom(cmd: String[]):** Hozzáad egy szobát a játékállapothoz a paraméterben megkapott kapacitással és tulajdonságokkal.
- **+addDoor(cmd: String[]):** Hozzáad egy ajtót a játékállapothoz a paraméterben megkapott szobák között a szintén paraméterben megadott tulajdonságokkal.
- **+addItem(cmd: String[]):** Hozzáad egy tárgyat a paraméterben megadott típusból egy szobához vagy egy karakterhez.
- **+link(cmd: String[]):** A paraméterekben megkapott tárgyakat kapcsoltatja össze az aktív karakterrel.
- **+setActiveCharacter(cmd: String[]):** Beállítja a paraméterben megkapott karaktert aktívnak. Tehát a parancs kiadása után az előbb említett parancsok erre a karakterre lesznek hatással.
- **+merge(cmd: String[]):** A paraméterekben megkapott szobákat egyesíti.
- **+split(cmd: String[]):** A paraméterben megkapott szobát bontja.

8.1.6 Door

- **Felelősség**

Az ajtón át elérhető szobák és az ajtó tulajdonságainak tárolására szolgáló osztály.

- **Ősosztályok**

Nincsen.

- **Interfészek**

Nincsen.

- **Attribútumok**

- **-firstRoom: Room:** Az ajtó egyik oldalán lévő szoba.
- **-secondRoom: Room:** Az ajtó másik oldalán lévő szoba.
- **-close: bool:** Ezen attribútum bool értékkel tárolja, hogy az adott ajtót lehet-e használni.
- **-oneway: bool:** Ezen attribútum bool értékkel tárolja, hogy az adott ajtó egyirányú-e. Igaz érték esetén ezen ajtón keresztül csak a firstRoom által tárolt szobából lehet a secondRoom által tárolt szobába menni.

- **Metódusok**

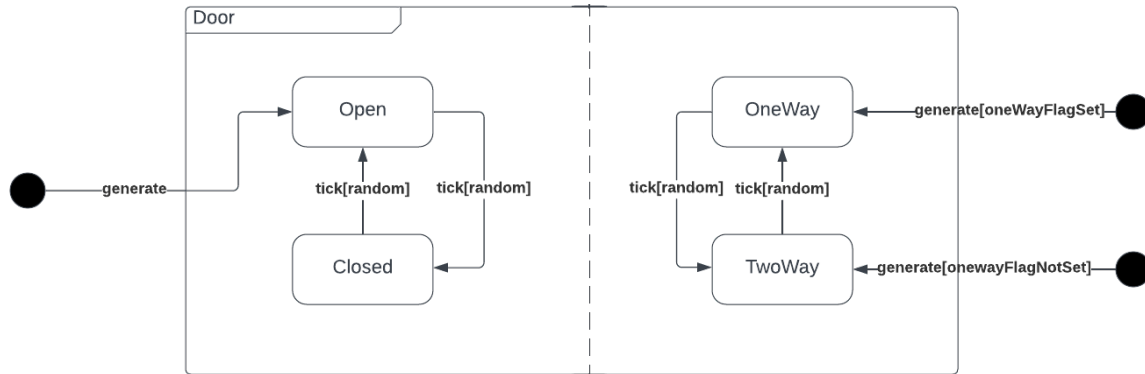
- **+void changeClosed():** A closed attribútum értékét az ellentettjére állítja.
- **+bool letEntityThrough(e: Entity, dest: Room):** Legelőször megnézi, hogy a closed attribútuma igaz-e, ha nem, akkor azonnal visszatér hamissal. Ha igaz, akkor megvizsgálja, hogy a paraméterül kapott dest szoba megegyezik-e valamelyik attribútumként tárolt szobával, ha nem, akkor nem csinál semmit és hamissal tér vissza. Ha megegyezik, akkor megnézi, hogy a oneway attribútum igaz-e, ha igaz, akkor megnézi, hogy a dest paraméter az a secondRoom-mal egyezik-e. Ha nem egyeznek, akkor visszatér hamissal, ha megegyeznek, akkor a függvény a oneway értékétől függetlenül a következőképp fut le, a dest szobán meghívja az acceptEntity(e: Entity) függvényt és paraméterének adja a kapott e entitást és a függvény visszatérési értékével megegyező értékével tér vissza.

Pseudo:

1. HA closed = igaz
2. AKKOR visszatér: HAMIS
3. HA dest = firstRoom vagy dest = secondRoom
4. AKKOR HA oneway = igaz
5. AKKOR HA dest != secondRoom
6. visszatér: HAMIS
7. visszatér: dest.acceptEntity(e: Entity) értékével

- **+void swapRoom(oldRoom: Room, newRoom: Room):** Ezen függvény a paraméterben kapott oldRoom-ot először megvizsgálja, hogy egyezik-e a firstRoom attribútumával, majd a secondRoom attribútumával és ha valamelyik attribútummal megegyezik, akkor azt az attribútumot a newRoom attribútumra cseréli.
- **+void setNeighbours(r1: Room, r2:Room):** Ezen függvény beállítja a paraméterben kapott, r1 objektumot a firstRoom attribútumnak, aztán az r2 paramétert pedig a secondRoom paraméternek.

- **+Room getNeighbour(r: Room):** Ezen függvény megvizsgálja, hogy a paraméterben kapott r Room megegyezik-e a firstRoom attribútummal. Ha megegyezik visszatér a secondRoom attribútummal, ha nem akkor a firstRoom attribútummal.
- **Állapotdiagram**



8.1.7 Entity

- **Felelősség**

Ez a tárgy az őt felhasználó hallgató szobáját mérgezővé teszi, aktiválás hatására.

- **Ősosztályok**

Nincsen.

- **Interfészek**

Nincsen.

- **Attribútumok**

- **#game: Game:** asd
- **#room: Room:** Ezen attribútumba tárolódik az Entity Room-ja, amelyben éppen van.
- **#items: List<Item>:** Azon Item-ek listája, amelyeket az Entity birtokol.
- **-itemCapacity: int:** Ezen attribútumba tárolódik, hogy hány Item-et birtokolhat ezen Entity.
- **-stunnedFor: int:** Ezen attribútumba tárolódik, hogy még hány tick-en erejéig kábult az adott entity.
- **-killed: bool:** Ezen attribútumba tárolódik, hogy az adott entitás meghalt-e már.

- **Metódusok**

- **+bool move(dest: Room, door: Door):** Ezen függvény először megvizsgálja, hogy a stunnedFor attribútum, nagyobb-e mint 0, ha igen, akkor azonnal hamis értékkel tér vissza. Ha nem akkor a paraméterben kapott door objektumnak meghívja a letEntityThrough(e: Entity, dest: Room) függvényét, amelyben első paraméternek önmagát adja, második paraméternek pedig a paraméterben kapott dest Room objektumot. Ha ezen függvény hívás igaz értékkel tér vissza, akkor az Entity room attribútumára meghívódik a removeEntity(e: Entity), és a dest paraméterre pedig az addEntity(e: Entity) függvény. A függvény végső visszatérési értéke a hívott letEntityThrough(e: Entity, dest: Room) függvény visszatérési értékével megegyezik.

Pseudo:

1. HA stunnedFor > 1
2. AKKOR visszatér: HAMIS
3. átenged= door.letEntityThrough(e: Entity, dest: Room)
4. HA átenged = igaz
5. AKKOR room.removeEntity(e: Entity)
6. dest.addEntity(e: Entity)
7. visszatér: átenged

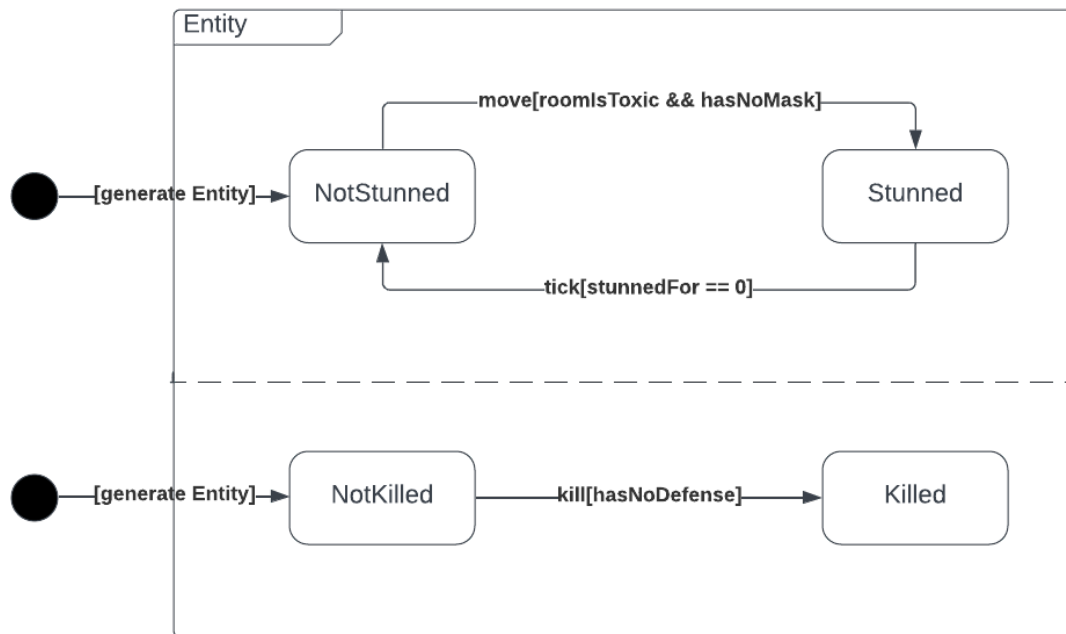
- **+void pickUp(item: Item):** Ezen függvény először megvizsgálja, hogy items attribútum elemeinek száma kisebb-e, mint az itemCapacity. Ha nem kisebb az elemeinek száma, vagy a stunnedFor attribútum értéke, nagyobb mint 0, azonnal visszatér. Ha kisebb, akkor meghívja ezen objektum room attribútumra a take(e: Entity, i: Item) függvényt, ahol az első paraméternek önmagát, a másodiknak a paraméterben kapott item-et adja meg. Ha a take(e: Entity, i: Item) függvény visszatérési értéke igaz, akkor az Entity meghívja magán az addItem(i: Item) függvényt melynek paraméterül a paraméterben kapott item-et adja.

Pseudo:

1. HA (items mérete) > itemCapacity VAGY stunnedFor > 1
2. AKKOR visszatér
3. HA room.take(e: Entity, i: Item) = igaz
4. AKKOR addItem(i: Item)
5. visszatér

- **+void drop(item: Item):** Ezen függvény először megvizsgálja, hogy a stunnedFor attribútum, nagyobb-e mint 0, ha igen, akkor azonnal visszatér. Ha nem a room attribútumnak meghívja az addItem(i: Item) függvényét és önmagán meghívja a removeItem(i: Item) függvényt. Mindkettő paraméterként a paraméterben kapott item-et kapja.
- **+bool tryForLogarLec():** Ez egy üres függvény. (Lényege, hogy a leszármazott osztályban ezen függvény definiálása megvalósítsa a szükséges logikát ezen általános esetben pedig ne történjen semmi.)
- **+meet(entity: Entity):** Ez egy üres függvény. (Lényege, hogy a leszármazott osztályban ezen függvény definiálása megvalósítsa a szükséges logikát ezen általános esetben pedig ne történjen semmi.)
- **+meet(prof: Prof):** Ez egy üres függvény. (Lényege, hogy a leszármazott osztályban ezen függvény definiálása megvalósítsa a szükséges logikát ezen általános esetben pedig ne történjen semmi.)
- **+meet(student: Student):** Ez egy üres függvény. (Lényege, hogy a leszármazott osztályban ezen függvény definiálása megvalósítsa a szükséges logikát ezen általános esetben pedig ne történjen semmi.)
- **+meet(cl: CleaningLady):** Ez egy üres függvény. (Lényege, hogy a leszármazott osztályban ezen függvény definiálása megvalósítsa a szükséges logikát ezen általános esetben pedig ne történjen semmi.)
- **+bool toxicate(destRoom: Room):** Ezen függvény meghívja az Entity osztályban található setStunnedFor(value: int) függvényt, 1 értéket adva paraméterben, majd a destRoom-ra az addItem(items: List<Item>) függvényét, a paraméterben az osztály items attribútumával. Ezután az items lista attribútumra hív a nyelvnek megfelelő függvényt, amely kitörli a lista elemeit. Majd a végén mindig igaz értékkel visszatér.
- **+bool immobilize():** Ezen függvény meghívja az Entity (azaz ebben az) osztályban található setStunnedFor(value: int) függvényt, 1 értéket adva paraméterben. Majd igaz értékkel tér vissza mindig.
- **+void addItem(item: Item):** Ezen függvény először megvizsgálja, hogy a item lista attribútumnak elemeinek száma kisebb-e mint az itemCapacity attribútum értéke. Ha kisebb akkor a paraméterben kapott item Item-et a lista elemihez adja.
- **+void removeItem(item: Item):** A paraméterben kapott item Item-et eltávolítja az osztály items lista attribútumából.
- **+void tick():** Ezen függvény megvizsgálja, hogy az osztály stunnedFor attribútuma, nagyobb-e, mint nulla és ha igen csökkenti ezen attribútum értékét eggyel.
- **+Room getRoom():** Ezen függvény visszatér a room attribútummal.
- **+void setStunnedFor(value: int):** A paraméterben kapott value értékre állítja az osztály stunnedFor attribútumát.
- **+void setRoom(room: Room):** A paraméterben kapott Room objektumra állítja az osztály room attribútumát.
- **+void setGame(game: Game):** A paraméterben kapott Game objektumra állítja az osztálygame attribútumát.

- **Állapotdiagram**



8.1.8 FakeFFP2

- **Felelősség**

Ez a tárgy nem rendelkezik semmilyen speciális képességgel.

- **Ősosztályok**

Item -> FFP2

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

- **+bool protectAgainstToxicRoom()**: Felülírja az FFP2 metódusát, hamissal tér vissza.

8.1.9 FakeLogarlec

- **Felelősség**

Ez a tárgy nem rendelkezik semmilyen speciális képességgel.

- **Ősosztályok**

Item -> Logarlec

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

- **+bool pickUp(e: Entity):** Felülírja az Logarlec metódusát, hamissal tér vissza.

8.1.10 FakeTVSZ

- **Felelősség**

Ez a tárgy nem rendelkezik semmilyen speciális képességgel.

- **Ősosztályok**

Item -> TVSZ

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

- **+bool protectAgainstProf()**: Felülírja az TVSZ metódusát, hamissal tér vissza.

8.1.11 FFP2

- **Felelősség**

Ha egy hallgatónál van ez a tárgy, akkor meg tudja őt védeni a gázos szobától.

- **Ősosztályok**

Item

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

- **+bool protectAgainstToxicRoom():** Felülírja az ősosztályban megírt metódust. Megvizsgálja, hogy a maszknak az Item osztályból örökölt health attribútuma nagyobb-e, mint 0. Ha nagyobb, akkor a health-ből levon egyet és igazzal tér vissza, ha nem nagyobb, akkor hamissal tér vissza.

8.1.12 Game

- **Felelősség**

Ez a tárgy az őt felhasználó hallgató szobáját mérgezővé teszi, aktiválás hatására.

- **Ősosztályok**

Nincsen.

- **Interfészek**

Nincsen.

- **Attribútumok**

- **-bool gameWon:** Ezen attribútum azt tárolja a játék véget ért-e már, játékosok által nyert álapottal.
- **-rooms: List<Room>:** Ezen listában a játékban szereplő szobák példányai találhatóak.
- **-entities: List<Entity>:** Ezen listában a játékban szereplő Entity-k(, karakterk) példányai találhatóak.

- **Metódusok**

- **+void gameWon():** Ezen függvény az osztály gameWon attribútum értékét igazra állítja.
- **+void addEntity(e: Entity):** Ezen függvény a paraméterben kapott e Entity-t az osztály entities lista attribútum elemeihez fűzi.
- **+void entityDied(e : Entity):** Ezen függvény a paraméterben kapott e Entity-t az osztály entities lista attribútum elemei közül törli.
- **+void addRoom(r: Room):** Ezen függvény a paraméterben kapott r Room-ot az osztály rooms lista attribútum elemeihez fűzi.
- **+void removeRoom(r: Room):** Ezen függvény a paraméterben kapott r Room-ot az osztály rooms lista attribútum elemei közül törli.
- **+void tick():** Ezen függvény végig megy az osztály entities lista attribútum elemein és minden elemre meghívja az Entity tick() függvényét. Majd végig megy az osztály rooms lista attribútum elemein és minden elemre meghívja a Room tick() függvényét. Ezután random számot generálva, a szoba indexei közül kisorsol két különböző indexet és a rooms lista attribútumból az első sorsolt indexű szobát lekérve, ezen szobára meghívjuk a merge(r: Room) függvényt melynek paraméterül a második sorsolt indexű szobát adjuk. Ha a merge függvény igaz értékkel tér vissza, akkor a második sorsolt indexű szobát a rooms lista elemei közül a removeRoom(r: Room) osztályban található függvénnyel eltávolítja a listából. Ezután generál még egy random indexű szobát, és ezen szobát a rooms lista attribútumból lekérve meghívja rá a split() függvényt. Ha a split függvény egy szobával tér vissza, akkor ezen osztály addRoom(r: Room) függvényével, hozzáadja a rooms lista attribútum elemeihez az új szobát.

Pseudo:

1. CIKLUS ELEJE e végigmegy entities elemein
2. e.tick()
3. CIKLUS VÉGE
4. CIKLUS ELEJE r végigmegy rooms elemein
5. r.tick()
6. CIKLUS VÉGE

```
7. elsoIndex = random szoba index generálás
8. masodikIndex = random másik indexű szoba generálás
9. HA rooms[elsoIndex].merge( rooms[masodikIndex] ) = igaz
10.     AKKOR removeRoom( rooms[masodikIndex] )
11. splitIndex = random szoba index generálás
12. splitSzoba = rooms[splitIndex].split()
13. HA splitSzoba != ÜRES ((null))
14.     AKKOR addRoom(splitSzoba)
```

- **+List<Room> getRoom():** Ezen függvény visszatérési értéként megadja az osztály rooms attribútumát.
- **+List<Entity> getEntity(): ():** Ezen függvény visszatérési értéként megadja az osztály entities attribútumát.

8.1.13 Item

- **Felelősség**

A játékban megtalálható tárgyakat reprezentáló osztály. A tárgyak felhasználásáért, összekapcsolásáért, valamint felvételéért felel.

- **Össztályok**

Nincsen.

- **Interfészek**

Nincsen.

- **Attribútumok**

- **#pair: Item:** Egy tárgyat tárol, ami ezzel a tárggyal van összekapcsolva, ha az érvényes objektum.
- **-health: int:** A tárgynak az éppen meglévő életpontjait tárolja.

- **Metódusok**

- **+bool pickUp(entity: Entity):** Ebben az implementációban igazzal tér vissza. Arra szolgál, hogy például a logarléc vagy a tranzisztor felüldefiniálja és új viselkedést vezessenek be.
- **+void activate(student: Student):** Paraméterként kapja azt a hallgatót, aki meghívta. Ez egy üres implementáció, arra szolgál, hogy a leszármazott tárgyak felüldefiniálják, ha parancsra fel lehet őket használni.
- **+bool protectAgainstToxicRoom():** Arra szolgál, hogy a tárgy megvédje a hallgatót a mérgező szoba támadásától. Ebben az implementációban hamissal tér vissza, a leszármazottak, amiknek van ilyen képességük felüldefiniálják.
- **+bool protectAgainstProf():** Arra szolgál, hogy a tárgy megvédje a hallgatót az oktatók támadásaitól. Ebben az implementációban hamissal tér vissza, a leszármazottak, amiknek van ilyen képességük felüldefiniálják.
- **+void link(item: Item):** Ebben az implementációban üres. Arra szolgál, hogy felüldefiniálják a leszármazottak, amiket össze lehet kapcsolni.
- **+bool link(transistor: Transistor):** Ebben az implementációban hamissal tér vissza, arra szolgál, hogy a tranzisztor felüldefiniálja. Használata azt biztosítja, hogy tranzisztort csak tranzisztorral lehessen összekapcsolni.
- **+void setPair(pair: Item):** A pair attribútumot beállítja a paraméterként kapott tárgyra.
- **+updateRoom(r: Room):** Üres implementáció, tranzisztor írja felül.

8.1.14 **Logarlec**

- **Felelősség**

Ez a tárgy az őt felhasználó hallgató szobáját mérgezővé teszi, aktiválás hatására.

- **Ősosztályok**

Item

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

- **+bool pickUp(entity: Entity):** A paraméterként kapott entitáson meghívja a tryForLogarLec függvényt és ennek a visszatérési bool értékével tér vissza ez a metódus is.

8.1.15 Prof

- **Felelősség**

Ez a tárgy az őt felhasználó hallgató szobáját mérgezővé teszi, aktiválás hatására.

- **Ősosztályok**

Entity

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

- **+void meet(entity: Entity):** A kapott entitásnak meghívja a meet(prof: Prof) függvényét, aminek átadja saját magát.
- **+void meet(student: Student):** A kapott tanulónak meghívja a meet(prof: Prof) függvényét, aminek átadja saját magát.

8.1.16 Room

- **Felelősség**

Ez a tárgy az őt felhasználó hallgató szobáját mérgezővé teszi, aktiválás hatására.

- **Ősosztályok**

Nincsen.

- **Interfészek**

Nincsen.

- **Attribútumok**

- **-doors: List<Door>:** A szobához tartozó ajtókat tárolja, amik más szobákhoz vezetnek.
- **-items: List<Item>:** A szobához tartozó tárgyakat tárolja, amelyek a szobában vannak lerakva.
- **-entities: List<Entity>:** A szobában lévő entitásokat tárolja.
- **-capacity: int:** A szobának a kapacitása, ettől több entitás nem tartózkodhat a szobában.
- **-isToxic: bool:** Ezen attribútum tárolja, a szoba gázos állapotát. Igaz, ha gázos, hamis ha nem.
- **-isWet: bool:** Ezen attribútum tárolja, a szoba vizes állapotát. Igaz, ha vizes, hamis ha nem.
- **-isCursed: bool:** Ezen attribútum tárolja, a szoba átkozott állapotát. Igaz, ha átkozott, hamis ha nem.
- **-isSticky: bool:** Ezen attribútum tárolja, a szoba ragacsos állapotát. Igaz, ha ragacsos, hamis ha nem.
- **-stickyCounter: int:** Ezen attribútum tárolja, hogy mennyi entitás lépett be a szobába takarítás után.

- **Metódusok**

- **+bool acceptEntity(e: Entity):** Ha a capacity attribútumtól nagyobb vagy egyenlő az entities lista mérete, akkor hamissal tér vissza. Megvizsgálja, hogy gázos-e a szoba, ha gázos, akkor meghívja a kapott entitáson a toxicate(r: Room) függvényt és ha igazzal tér vissza a hívott függvény, akkor az eredeti függvény hamissal tér vissza azonnal. Megvizsgálja, hogy nedves-e a szoba, ha nedves, akkor meghívja a kapott entitáson a immobilize() függvényt és ha igazzal tér vissza a hívott függvény, akkor az eredeti függvény hamissal tér vissza azonnal. Ezután ha még nem tért vissza függvény végigmegy az entites lista attribútum összes elemén és meghívja az összes elemre a meet(e: Entity) függvényt amelynek paraméterül, a paraméterben kapott e Entity-t adja. Majd ezután visszatér igaz értékkel.

Pseudo:

1. HA (entities méret) >= capacity
2. AKKOR visszatér: HAMIS
3. HA isToxic = igaz
4. AKKOR HA e.toxicate(r: Room) = igaz
5. AKKOR visszatér: HAMIS
6. HA isWet = igaz
7. AKKOR HA e.immobilize() = igaz

```

8.                AKKOR visszatér: HAMIS
9. CIKLUS ELEJE ent végig iterál entities elemein
10.    ent.meet(e: Entity)
11. CIKLUS VÉGE
12. visszatér: IGAZ

```

- **+void addEntity(e: Entity):** Ez a metódus a kapott entitást hozzáadja az entities listához. Megnézi, a stickyCounter értékét, ha ez 0, akkor meghívja magán a makeRoomSticky() függvényét, ha a stickyCounter értéke nagyobb, mint 0, akkor csökkenteni ennek az attribútumnak az értékét eggyel.
- **+void removeEntity(e: Entity):** A kapott entitást eltávolítja az entities listából.
- **+bool take(e: Entity, i: Item):** Ha az isSticky attribútum igaz, akkor hamissal tér vissza, ha hamis, akkor pedig a kapott itemen meghívja a pickUp(e: Entity) függvényt, paraméterül átadja a kapott e entitást, ha ez igazzal tér vissza, akkor kitörli meghívja magán a removeItem(i: Item)-et és megadja a kapott i tárgyat. Ezután visszatér a pickUp(e: Entity) értékével.
- **+void addDoor(door: Door):** A doors listához hozzáadja a kapott ajtót.
- **+void removeDoor(door: Door):** A doors listából eltávolítja a kapott ajtót.
- **+void addItem(i: Item):** Az items listához hozzáadja az i kapott tárgyat.
- **+void addItem(items: List<Item>):** Az items listából eltávolítja a listában kapott tárgyakat.
- **+void removeItem(i: Item):** Az items listából eltávolítja a paraméterként kapott i tárgyat.
- **+bool merge(r: Room):** A hívott szobába próbálja meg beolvasztani a paraméterben átadott szobát. Az egyesülés nem valósul meg, ha a szobákban vannak entitások. Ha meg tud valósulni az egyesülés, akkor a kapott szobára igaz tulajdonságokat beállítjuk a hívott szobában is. Ezt követően átvesszük a paraméterben kapott szoba minden tárgyát, azokon meghívjuk az updateRoom() metódust. (Ez tranzisztoroknál fontos.) Majd átvesszük az ajtókat is, egyúttal átállítva bennük a régi szobáról a hivatkozást az újra. Végezetül a hívott szobának megváltoztatjuk a kapacitását a paraméterben kapott szoba kapacitására, ha az nagyobb.

Pseudo:

```

1.    HA !üres(this.entities) VAGY !üres(r.entities)
2.        Visszatér: HAMIS
3.
4.    HA r.isToxic = IGAZ
5.        this.makeRoomToxic()
6.    HA r.isWet = IGAZ
7.        this.makeRoomWet()
8.    HA r.isCursed = IGAZ
9.        this.makeRoomCursed()
10.   HA r.isSticky = IGAZ
11.       this.makeRoomSticky()
12.
13.   CIKLUS ELEJE item végig iterál az r.items elemein
14.       item.updateRoom(this)
15.       items listához hozzáadja: item
16.   CIKLUS VÉGE
17.   r.items törlése
18.
19.   CIKLUS ELEJE door végigiterál az r.doors elemein:
20.       door.swapRoom(r, this)
21.       HA door.getNeighbour(this) != this

```

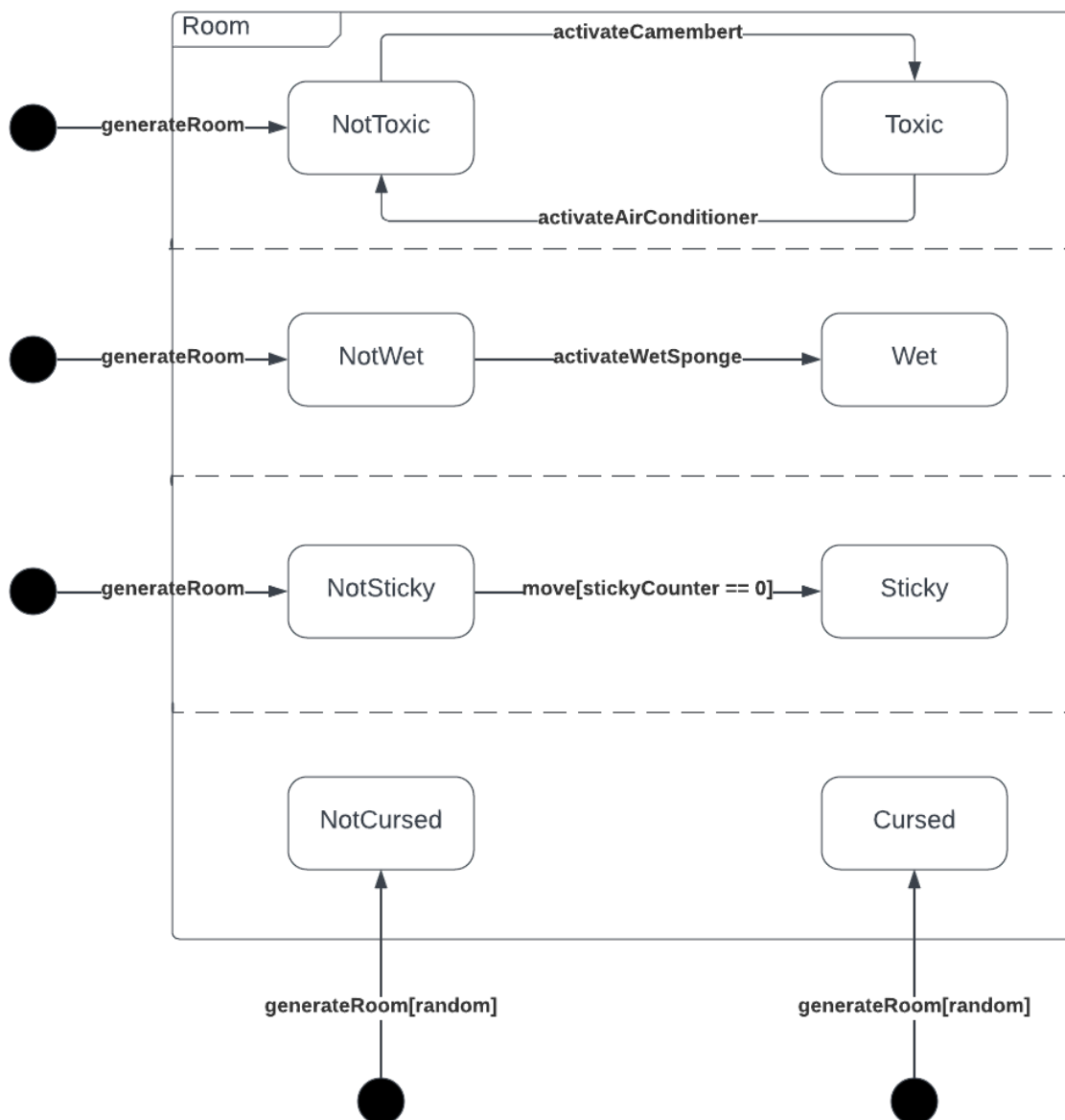

- 22. doors listához hozzáadja: door
- 23. CIKLUS VÉGE
- 24. r.doors törlése
- 25.
- 26. HA r.capacity > this.capacity
- 27. this.setCapacity beállítása: r.capacity
- 28.
- 29. Visszatér: IGAZ
- **+Room split():** A hívott szobát megpróbálja szétválasztani két szobára, ami egymás szomszédjai lesznek. Ez akkor lehetséges, ha a szobában nincsenek entitások. Ha lehetséges a művelet, akkor egy szobával, különben null értékkel tér vissza. Az új szoba néhány tulajdonságát átveszi annak a szobának, amiből ő kiválik, kapacitása pedig megegyezik a régiével. Ezt követően a tárgyak közül véletlenszerűen néhányat átmozgatunk az új szobába, itt is meg kell hívni az updateRoom() metódust az áthelyezett tárgyakon.

Pseudo:

1. HA !üres(this.entities):
2. Visszatér: null
- 3.
4. Új szoba létrehozása: newRoom
- 5.
6. Véletlenszerű IGAZ/HAMIS létrehozása: random
7. HA this.isToxic = IGAZ és random = IGAZ:
8. this.isToxic = HAMIS
9. newRoom.isToxic = IGAZ
10. Véletlenszerű IGAZ/HAMIS létrehozása: random
11. HA this.isWet = IGAZ és random = IGAZ:
12. this.isWet = HAMIS
13. newRoom.isWet = IGAZ
14. Véletlenszerű IGAZ/HAMIS létrehozása: random
15. HA this.isCursed = IGAZ és random = IGAZ:
16. this.isCursed = HAMIS
17. newRoom.isCursed = IGAZ
18. Véletlenszerű IGAZ/HAMIS létrehozása: random
19. HA this.isSticky = IGAZ és random = IGAZ:
20. this.isSticky = HAMIS
21. newRoom.isSticky = IGAZ
- 22.
23. newRoom.capacity beállítása: this.capacity
- 24.
25. Véletlenszerű egész választása [0..items mérete]: random
26. CIKLUS 0-tól random-ig:
27. Véletlenszerű index kiválasztása [0..items mérete]: randomIndex
28. items[randomIndex].updateRoom()
29. newRoomhoz hozzáadja az items[randomIndex] elemet
30. items listából eltávolítja az items[randomIndex] elemet
31. CIKLUS VÉGE
- 32.
33. Véletlenszerű egész választása [0..doors mérete]: random
34. CIKLUS 0-tól random-ig:
35. Véletlenszerű index kiválasztása: randomIndex a doors listából
36. doors[randomIndex].swapRoom(this, newRoom) meghívása
37. HA doors[randomIndex].getNeighbour(this) != this:
38. newRoomhoz hozzáadja a doors[randomIndex] elemet
39. doors listából eltávolítja a doors[randomIndex] elemet
40. CIKLUS VÉGE
- 41.
42. Visszatér: newRoom

- **+void tick():** Tulajdonságaitól függően minden benne tartózkodó entitást megpróbálhat megmérgezni, megbénítani. Ha átkozott a szoba, akkor az ajtajain meghívhatja a magicDoor() metódust.
- **+void magicDoor():** A doors listában tárolt összes ajtón meghívja a changeClosed() metódusát.
- **+void setCapacity(capacity: int):** A paraméterként kapott számra állítja a capacity attribútumot.
- **+void getCapacity():** Visszaadja a capacity-ben tárolt számot.
- **+void makeRoomToxic():** Az isToxic attribútum értékét igazra állítja.
- **+void makeRoomWet():** Az isWet attribútum értékét igazra állítja.
- **+void makeRoomCursed():** Az isCursed attribútum értékét igazra állítja.
- **+void makeRoomSticky():** Az isSticky attribútum értékét igazra állítja.
- **+void detoxicate():** Az isToxic attribútum értékét hamisra állítja.
- **+void clean(e: Entity):** Az entities lista összes entitásán meghívja a move(room: Room) függvényt, ahol a room egy véletlenszerűen kiválasztott szoba a rooms listából. Ezután pedig beállítja a stickyCounter értékét 3-ra.
- **+List<Item> getItems():** Ezen függvény visszatérési értékként megadja az osztály items lista attribútumát.
- **+List<Door> getDoors():** Ezen függvény visszatérési értékként megadja az osztály doors lista attribútumát.

- Állapotdiagram



8.1.17 Student

- **Felelősség**

Ez a tárgy az őt felhasználó hallgató szobáját mérgezővé teszi, aktiválás hatására.

- **Ősosztályok**

Entity

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

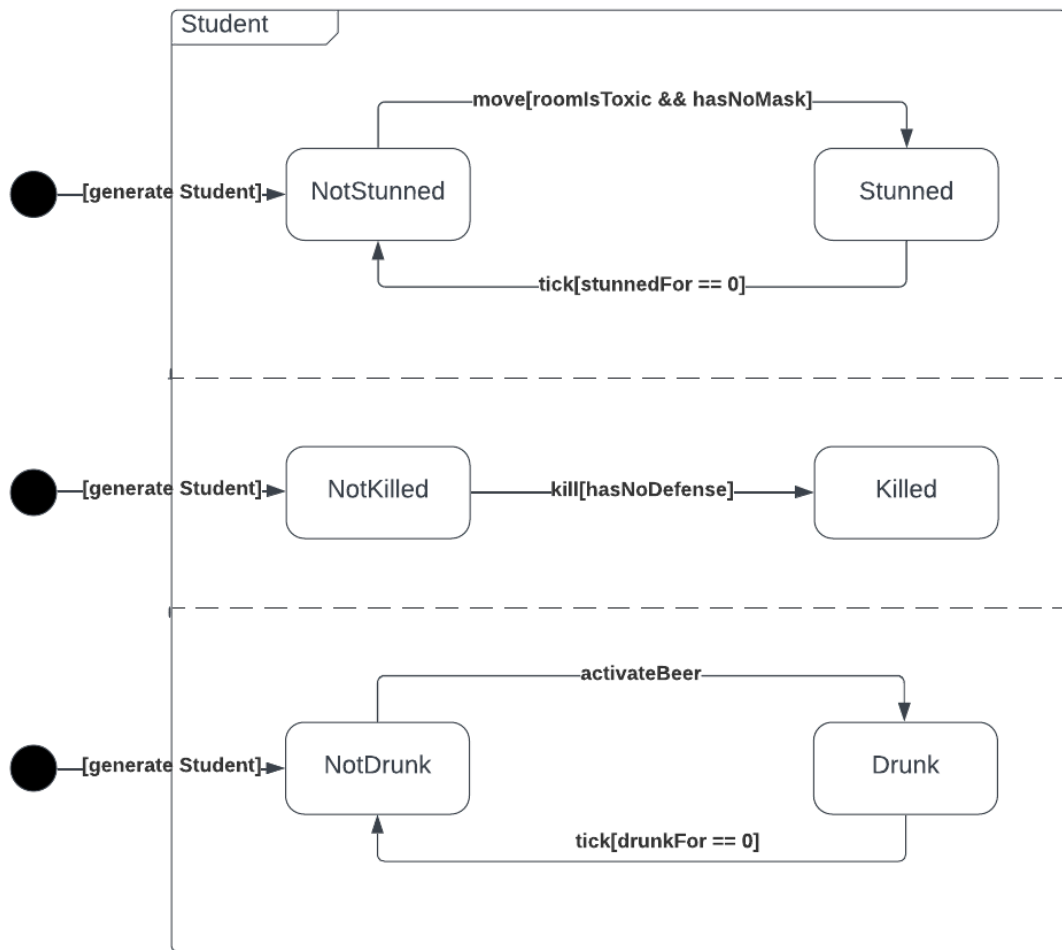
- **+void kill():** Ezen függvény először megvizsgálja, hogy a drunkFor attribútum értéke, nagyobb-e mint nulla, ha igen azonnal visszatér a függvény. Ha nem, akkor elkezd iterálni az items lista attribútum elemein, és meghívja az elemeken a protectAgainstProf() függvényt és ha valamelyik elem esetén igaz értékkel tér vissza ezen függvény akkor azonnal visszatér a kill függvény. Ha egyik elem esetén sem ad igaz értéket a protectAgainstProf() függvény, akkor miután teljesen végig iterált az items elemein, meghívja a room attribútumára az addItem(i: List<Item>) függvény paraméterben pedig az items lista attribútumát adja. Majd az items lista attribútumra hív egy a nyelvnek megfelelő függvényt, amely kitörli a lista elemeit. Legvégül a game attribútum entityDied(e: Entity) függvényét, amelynek paraméterül önmagát adja.

Pseudo:

1. HA drunkFor > 0
2. AKKOR visszatér
3. CIKLUS ELEJE i végig iterál items elemein
4. HA i.protectAgainstProf() = igaz
5. AKKOR: visszatér
6. CIKLUS VÉGE
7. room.addItem(i: Item)
8. items elemeinek törlése
9. room.removeEntity(e: Entity)
10. game.entityDied(e: Entity)
11. visszatér

- **+void activateItem(item: Item):** A paraméterként kapott item-en meghívja az active(s: Student) függvényt, paraméterként magát adja át.
- **+bool tryForLogarlec():** Ezen függvény meghívja a game attribútum gameWon() függvényét, majd visszatér igaz értékkel.
- **+void meet(entity: Entity):** Ezen függvény a paraméterben kapott Entity meet(entity: Entity) függvényét hívja meg melynek paraméterül, önmagát adja.
- **+void meet(prof: Professor):** Ezen függvény a paraméterben kapott Prof meet(entity: Entity) függvényét hívja meg melynek paraméterül, önmagát adja.
- **+bool toxicate(room: Room):** Az ősosztályból származott items listának az összes tárgyán meghívja a protectAgainstToxic() metódust, ha valamelyik igazzal tér vissza, akkor hamissal tér vissza, ha egyik sem tér vissza igazzal, akkor igazzal tér vissza.
- **+bool immobilize():** Ezen függvény mindig hamis értékkel tér vissza.

- **+void linkItems(item1: Item, item2: Item):** Ezen függvény a paraméterben kapott, i1 Item-re meghívja a link(i: Item) függvényét, paraméterben az i2 kapott Item paramétert használva
- **+void makeStudentDrunk():** Az osztály drunFor attributumát 3-ra állítja.
- **+void tick():** Ezen függvény meghívja a szülő osztályának tick() függvényét, majd megvizsgálja, hogy a drunkFor() attribútum, nagyobb-e mint nulla és ha igen csökkenti az értékét ezen attribútumnak eggyel.
- **Állapotdiagram**



8.1.18 Transistor

- **Felelősség**

Ez a tárgy az őt felhasználó hallgató szobáját mérgezővé teszi, aktiválás hatására.

- **Össztályok**

Item

- **Interfészek**

Nincsen.

- **Attribútumok**

- **-room: Room:** Ezen attribútumba tárolódik, azon Room példánya amelyikben a tranzistor helyezésre került.
- **-active: bool:** Ezen attribútumba az tárolódik bool értékben, hogy aktív-e ezen tranzistor.

- **Metódusok**

- **+void activate(student: Student):** Ezen függvény először megvizsgálja, hogy az osztály által örökölt pair attribútum, tárol-e már párt. Ha nem tárol, visszatér, azaz véget ér azonnal a függvény. Lekéri hogy a osztályból származó pair attribútumba tárolt tranzistor aktív-e, ha igen, akkor meghívja a pair attribútum teleport(student: Student) függvényét, melynek paraméterül a paraméterben kapott student-et adja. Ha a teleport függvény igaz értékkel tér vissza, akkor a pair attribútumot üres értékre állítja (null). Ha pair attribútumba tárolt tranzistor nem volt aktív, akkor az osztály setRoom(r: Room) függvényt, hívja paraméterül a paraméterben kapott student room attribútumát kapja (a student getRoom() függvényének segítségével) és még meghívja a paraméterben kapott student objektum drop(i: Item) függvényér paraméterben önmagát átadva és az osztály saját setActive(active: bool) függvényét paraméterben igaz értékkel.

Pseudo:

1. HA pair = ÜRES ((null))
2. AKKOR visszatér
3. HA pair.active = igaz
4. AKKOR HA pair.teleport(student)
5. pair=ÜRES (null)
6. EGYÉBKÉNT
7. setRoom(student.getRoom())
8. setActive(IGAZ)
9. student.Drop(önmaga ((this)))

- **+bool pickUp(entity: Entity):** Ezen függvény először megvizsgálja, hogy az active attribútuma az osztálynak, igaz értéket tárol-e. Ha igenhamis értékkel tér vissza a függvény. Egyébként igaz értékkel tér vissza a függvény.
- **+void link(item: Item):** Ezen függvény meghívja először a paraméterben kapott, item link(i: Item) függvényét és ha a hívott függvény igaz értékkel tér vissza, akkor meghívja az osztályból származó setPair(pair: Item) függvényt, amelynek paraméterül, a paraméterben kapott item objektumot állítja.

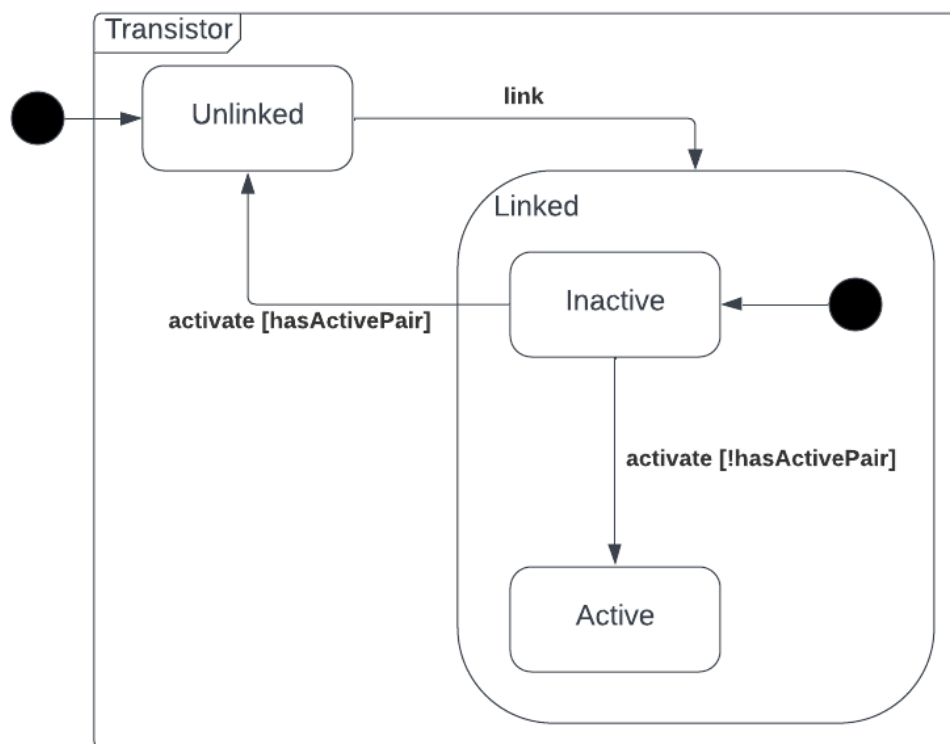
- **+bool link(transistor: Transistor):** Ezen függvény a meghívja az ősosztályból származó setPair(pair: Item) függvényt, amelynek paraméterül, a paraméterben kapott transistor objektumot állítja.
- **+bool teleport(student: Student):** Ezen függvény először meghívja az osztály room attribútumára, az acceptEntity(e: Entity) függvény paraméterül, a paraméterben kapott student objektumot adva. Amennyiben az előbbi függvény igaz értékkel tér vissza, a következő függvények kerülnek meghívásra:
A paraméterben kapott student-nek a drop(i: Item) függvénye, amelynek paraméterül a pair attribútumot adja. Majd lekéri a paraméterben kapott student-nek a room attribútumát, és ezen Room objektumra meghívja removeEntity(e: Entity) függvényt, amelynek paraméterül a paraméterben kapott student objektumot adja. És szintén meghívja a paraméterben kapott student-nek a setRoom(r: Room) függvényét, paraméterben az osztály room attribútumát adva. Az osztály room attribútumának meghívja addEntity(e: Entity) függvényét, paraméterül a paraméterben kapott student-et átadva. Legvégül ezen osztály setActive(active: bool) függvényét, amelynek paraméterül hamis értéket ad és az ősosztályból származó setPair(item: Item) függvényt oly módon, hogy a beállított pár attribútum üressé váljon (null értékkel). Majd legvégül az egész függvény, mindentől függetlenül az elején hívott acceptEntity(e: Entity) függvény visszatérési értékével tér vissza

Pseudo:

1. átenged = room.acceptEntity(student)
2. HA átenged
3. AKKOR:student.drop(pair)
4. student.getRoom().removeEntity(student)
5. room.addEntity(student)
6. student.setRoom(room)
7. setActive(HAMIS)
8. setPair(ÜRES ((null)))
- 9.visszatér: átenged

- **+void setRoom(room: Room):** Ezen függvény paraméterben kapott room Room-ra állítja az osztály room attribútumát.
- **+void setActive(active: bool):** Ezen függvény a paraméterben kapott, bool értékre állítja az osztály active attribútumát.

- **Állapotdiagram**



8.1.19 TVSZ

- **Felelősség**

Ez a tárgy az őt felhasználó hallgató szobáját mérgezővé teszi, aktiválás hatására.

- **Ősosztályok**

Item

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

- **+bool protectAgainstProf():** Ezen metódus megvizsgálja, hogy a TVSZ-nek az ősosztályból származó health attribútuma nagyobb-e, mint 0, ha nem, akkor hamissal visszatér. Ha nagyobb, akkor csökkenti az életét eggyel és igazzal tér vissza.

8.1.20 WetSponge

- **Felelősség**

Ez a tárgy az őt felhasználó hallgató szobáját mérgezővé teszi, aktiválás hatására.

- **Ősosztályok**

Item

- **Interfészek**

Nincsen.

- **Attribútumok**

Nincsen.

- **Metódusok**

- **+void activate(student: Student):** A hallgató szobáját gázossá változtatja.

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

8.2.1 Hallgató szomszédos szobába lép

- **Leírás**

A tesztet létrehoz két Szobát, közéjük egy ajtót, majd az első szobában elhelyez egy játékost. Ezt követően átmozgatja a játékost a második szobába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Játékos mozgatása. Nem várunk hibát.

- **Bemenet**

addRoom 5

addRoom 5

addDoor Room#1 Room#2

addPlayer Lacika Room#1

setActiveCharacter Lacika

move Room#2 Door#1

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

ID: Lacika

Stunned for: 0

Drunk for: 0

Killed: false

Items of entity: No items.

Room: Room#1

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room: No entities.

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false

Oneway: false

ID: Room#2

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room:

ID: Lacika

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1
Second Room: Room#2
Closed: false
Oneway: false

8.2.2 Hallgató nem szomszédos szobába lép

- **Leírás**

A tesztet létrehoz három Szobát és közéjük ajtókat, valamint elhelyez egy játékost az első szobában. Ezt követően megpróbálja áthelyezni a játékost a harmadik szobába. Ez nem történik meg, mert ez a szoba nem szomszédos a játékos szobájával.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Játékos mozgatus. Azt várjuk, hogy a játékos a szobájában maradjon.

- **Bemenet**

```
addRoom 5
addRoom 5
addRoom 5
addDoor Room#1 Room#2
addDoor Room#2 Room#3
addPlayer Joco Room#1
setActiveCharacter Joco
move Room#3 Door#2
```

- **Elvart kimenet**

Error: You gave a room, which is not the neighbour of your current room.

Game state: ongoing

Characters of Game:

```
ID: Joco
Stunned for: 0
Drunk for: 0
Killed: false
Items of entity: No items.
Room: Room#1
```

Rooms of Game:

```
ID: Room#1
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
  ID: Joco
Items of room: No items.
Doors of room:
  ID: Door#1
  First Room: Room#1
  Second Room: Room#2
  Closed: false
  Oneway: false
```

```
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room: No entites.
Items of room: No items.
```

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false

Oneway: false

ID: Door#2

First Room: Room#2

Second Room: Room#3

Closed: false

Oneway: false

ID: Room#3

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room: No entites.

Items of room: No items.

Doors of room:

ID: Door#2

First Room: Room#2

Second Room: Room#3

Closed: false

Oneway: false

8.2.3 Hallgató Sört vesz fel

- **Leírás**

A tesztet létrehoz egy Szobát, elhelyez abban egy játékost, valamint egy sört. Majd felveteti a játékos a sört.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tárgyak felvétele. Nem várunk hibát.

- **Bemenet**

addRoom 5

addPlayer Andris Room#1

addItem Beer Room#1

setActiveCharacter Andris

pickup Beer#1

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

ID: Andris

Stunned for: 0

Drunk for: 0

Killed: false

Items of entity:

ID: Beer#1

Health: 1

Pair: No pair.

Room: Room#1

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room:

ID: Andris

Items of room: No items.

Doors of room: No doors.

8.2.4 Hallgató teli raktárral vesz fel Tranzisztort

- **Leírás**

A tesztet létrehoz egy Szobát, elhelyez abban egy játékost, valamint egy tranzisztort. A játékosnál még elhelyez 5 db sört. Majd megpróbálja a játékosal felvetetni a tranzisztort.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tárgyak felvétele. Azt várjuk, hogy a tranzisztor a földön maradjon a szobában.

- **Bemenet**

```
addRoom 5
addPlayer Andris Room#1
addItem Beer Andris
addItem Beer Andris
addItem Beer Andris
addItem Beer Andris
addItem Beer Andris
addItem Beer Andris
addItem Transistor Room#1
setActiveCharacter Andris
pickup Transistor#1
```

- **Elvárt kimenet**

Error: Your character inventory is full.

Game state: ongoing

Characters of Game:

ID: Andris

Stunned for: 0

Drunk for: 0

Killed: false

Items of entity:

ID: Beer#1

Health: 1

Pair: No pair.

ID: Beer#2

Health: 1

Pair: No pair.

ID: Beer#3

Health: 1

Pair: No pair.

ID: Beer#4

Health: 1

Pair: No pair.

ID: Beer#5

Health: 1

Pair: No pair.

Room: Room#1

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room:

ID: Andris
Items of room:
ID: Transistor#1
Health: -1
Pair: No pair.
Active: false
Room: No room.
Doors of room: No doors.

8.2.5 Hallgató felveszi a Logarlécet

- **Leírás**

A tesztet létrehoz egy Szobát, elhelyez benne egy játékost, valamint egy logarlécet. Majd felveteti a játékkal a logarlécet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tárgyak felvétele, játék megnyerése. Nem várunk hibát.

- **Bemenet**

addRoom 5

addPlayer Joska Room#1

addItem Logarlec Room#1

setActiveCharacter Joska

pickup Logarlec#1

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

ID: Joska

Stunned for: 0

Drunk for: 0

Killed: false

Items of entity:

ID: Logarlec#1

Health: 1

Pair: No pair.

Room: Room#1

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room:

ID: Joska

Items of room: No items.

Doors of room: No doors.

8.2.6 Hallgató ragacsos szobában felvesz egy Rongyot

- **Leírás**

A tesztet létrehoz egy ragadós Szobát, elhelyez benne egy játékost, valamint egy rongyot. Ezt követően megpróbálja felvetetni a játékos a rongyot.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tárgyak felvétele. Azt várjuk, hogy a tárgy a szobában maradjon.

- **Bemenet**

```
addRoom 5 sticky
addPlayer Pistike Room#1
addItem WetSponge Room#1
setActiveCharacter Pistike
pickup WetSponge#1
```

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

```
ID: Pistike
Stunned for: 0
Drunk for: 0
Killed: false
Items of entity: No items.
Room: Room#1
```

Rooms of Game:

```
ID: Room#1
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: true
Entities of room:
    ID: Pistike
Items of room:
    ID: WetSponge#1
    Health: 1
    Pair: No pair.
Doors of room: No doors.
```

8.2.7 Hallgató Tranzisztorokat párosít

- **Leírás**

A tesztet létrehoz egy Szobát, elhelyez benne egy játékost. A játékosnak ad két tranzisztort. Ezt követően a játékosal összekapcsolatja a két tranzisztort.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tranzisztorok kapcsolása. Nem várunk hibát.

- **Bemenet**

```
addRoom 5
addPlayer Karoly Room#1
addItem Transistor Karoly
addItem Transistor Karoly
setActiveCharacter Karoly
link Transistor#1 Transistor#2
```

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

ID: Karoly

Stunned for: 0

Drunk for: 0

Killed: false

Items of entity:

ID: Transistor#1

Health: 1

Pair: Transistor#2

Active: false

Room: No room.

ID: Transistor#2

Health: 1

Pair: Transistor#1

Active: false

Room: No room.

Room: Room#1

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room:

ID: Karoly

Items of room: No items.

Doors of room: No doors.

8.2.8 Hallgató eldob egy Légfrissítőt

- **Leírás**

A tesztet létrehoz egy Szobát és elhelyez benne egy játékost (Hallgatót). A játékosnak (Hallgatónak) ad egy Légfrissítő tárgyat. Ezt követően a játékos (Hallgató) eldobja a Légfrissítőt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Légfrissítő, mint tárgy eldobása. Nem várunk hibát.

- **Bemenet**

```
addRoom 5
addPlayer Karoly Room#1
addItem AirConditioner Karoly
setActiveCharacter Karoly
drop AirConditioner#1
```

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

```
ID: Karoly
Stunned for: 0
Drunk for: 0
Killed: false
Items of entity: No items.
Room: Room#1
```

Rooms of Game:

```
ID: Room#1
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
    ID: Karoly
Items of room:
    ID: AirConditioner#1
    Health: 1
    Pair: No pair.
Doors of room: No doors.
```

8.2.9 Hallgató Légfrissítőt használ

- **Leírás**

A tesztet létrehoz egy gázos Szobát és elhelyez benne egy játékost (Hallgatót). A játékosnak (Hallgatónak) ad egy Légfrissítő tárgyat. Ezt követően a játékos (Hallgató) elhasználja a légfrissítőt a szobában.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Légfrissítő használata. Nem várunk hibát.

- **Bemenet**

```
addRoom 5 toxic
addPlayer Karoly Room#1
addItem AirConditioner Karoly
setActiveCharacter Karoly
getInfo
activate AirConditioner#1
```

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

```
ID: Karoly
Stunned for: 0
Drunk for: 0
Killed: false
Items of entity:
    ID: AirConditioner#1
    Health: 1
    Pair: No pair.
Room: Room#1
```

Rooms of Game:

```
ID: Room#1
Capacity: 5
Toxic: true
Wet: false
Cursed: false
Sticky: false
Entities of room:
    ID: Karoly
Items of room:
Doors of room: No doors.
```

Game state: ongoing

Characters of Game:

```
ID: Karoly
Stunned for: 0
Drunk for: 0
Killed: false
Items of entity:
    ID: AirConditioner#1
    Health: 0
    Pair: No pair.
Room: Room#1
```

Rooms of Game:

```
ID: Room#1
```

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room:

 ID: Karoly

Items of room: No items.

Doors of room: No doors.

8.2.10 Hallgató megiszik egy Sört

- **Leírás**

A tesztet létrehoz egy Szobát és elhelyez benne egy játékost (Hallgatót). A játékosnak (Hallgatónak) ad egy Sör tárgyat. Ezt követően a játékos elhasználja a Sör tárgyat.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Sör használata. Nem várunk hibát.

- **Bemenet**

```
addRoom 5
addPlayer Karoly Room#1
addItem Beer Karoly
setActiveCharacter Karoly
getInfo
activate Beer#1
```

- **Elvárt kimenet**

```
Game state: ongoing
Characters of Game:
  ID: Karoly
  Stunned for: 0
  Drunk for: 0
  Killed: false
  Items of entity:
    ID: Beer#1
    Health: 1
    Pair: No pair.
  Room: Room#1
Rooms of Game:
  ID: Room#1
  Capacity: 5
  Toxic: false
  Wet: false
  Cursed: false
  Sticky: false
  Entities of room:
    ID: Karoly
  Items of room:
  Doors of room: No doors.
```

```
Game state: ongoing
Characters of Game:
  ID: Karoly
  Stunned for: 0
  Drunk for: 3
  Killed: false
  Items of entity:
    ID: Beer#1
    Health: 0
    Pair: No pair.
  Room: Room#1
Rooms of Game:
  ID: Room#1
```


Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room:

 ID: Karoly

Items of room: No items.

Doors of room: No doors.

8.2.11 Hallgató teleportál

- **Leírás**

A tesztet létrehoz két Szobát, közöttük egy ajtót és elhelyez az egyik szobába egy játékost (Hallgató). A játékosnak ad két Tranzisztort, amiket összekapcsol. Miután az összekapcsolás megtörtént, az egyik Tranzisztort aktiváljuk, ezáltal aktívvá válnak a Tranzisztorok és elejti a szobában azt a Tranzisztort, amelyet aktiváltunk. Elejtés után átmegy a másik szobába, és ott aktiválja a nála lévő Tranzisztort.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tranzisztorok teleportáló képessége. Nem várunk hibát.

- **Bemenet**

```
addRoom 5
addRoom 5
addDoor Room#1 Room#2
addPlayer Karoly Room#1
addItem Transistor Karoly
addItem Transistor Karoly
setActiveCharacter Karoly
link Transistor#1 Transistor#2
activateTransistor#1
move Room#2 Door#1
getinfo
activate Transistor#2
```

- **Elvárt kimenet**

```
Game state: ongoing
Characters of Game:
  ID: Karoly
  Stunned for: 0
  Drunk for: 0
  Killed: false
  Items of entity:
    ID: Transistor#2
    Health: 1
    Pair: Transistor#1
    Active: true
    Room: No room.
  Room: Room#2
Rooms of Game:
  ID: Room#1
  Capacity: 5
  Toxic: false
  Wet: false
  Cursed: false
  Sticky: false
  Entities of room: No entities.
  Items of room:
    ID: Transistor#1
    Health: 1
    Pair: Transistor#2
    Active: true
  Doors of room:
```

ID: Door#1
First Room: Room#1
Second Room: Room#2
Closed: false
Oneway: false
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
 ID: Karoly
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#1
 Second Room: Room#2
 Closed: false
 Oneway: false

Game state: ongoing
Characters of Game:
 ID: Karoly
 Stunned for: 0
 Drunk for: 0
 Killed: false
 Items of entity:
 ID: Transistor#2
 Health: 1
 Pair: Transistor#1
 Active: false
 Room: No room.
 Room: Room#1
Rooms of Game:
 ID: Room#1
 Capacity: 5
 Toxic: false
 Wet: false
 Cursed: false
 Sticky: false
 Entities of room:
 ID: Karoly
 Items of room:
 ID: Transistor#1
 Health: 1
 Pair: Transistor#2
 Active: false
 Doors of room:
 ID: Door#1

First Room: Room#1
Second Room: Room#2
Closed: false
Oneway: false
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room: No entities.
Items of room: No items.
Doors of room:
ID: Door#1
First Room: Room#1
Second Room: Room#2
Closed: false
Oneway: false

8.2.12 Hallgató elhasznál egy Camembert

- **Leírás**

A tesztet létrehoz egy Szobát és benne egy játékost (Hallgatót). A játékosnak (Hallgatónak) ad egy Camember tárgyat. Ezt követően a játékos (Hallgató) kibontja a Camembert.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Camember használata. Nem várunk hibát.

- **Bemenet**

```
addRoom 5
addPlayer Karoly Room#1
addItem Camembert Karoly
setActiveCharacter Karoly
getInfo
activate Camembert
```

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

```
ID: Karoly
Stunned for: 0
Drunk for: 0
Killed: false
Items of entity:
    ID: Camembert#1
    Health: 1
    Pair: No pair.
```

Room: Room#1

Rooms of Game:

```
ID: Room#1
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
    ID: Karoly
Items of room: No items.
Doors of room: No doors.
```

Game state: ongoing

Characters of Game:

```
ID: Karoly
Stunned for: 0
Drunk for: 3
Killed: false
Items of entity:
    ID: Camembert#1
    Health: 0
    Pair: No pair.
```

Room: Room#1

Rooms of Game:

```
ID: Room#1
```

Capacity: 5

Toxic: true

Wet: false

Cursed: false

Sticky: false

Entities of room:

 ID: Karoly

Items of room: No items.

Doors of room: No doors.

8.2.13 Hallgató felvesz egy hamis Logarlécet

- **Leírás**

A teszteset létrehoz egy Szobát és egy játékost (Hallgató). A szobában elhelyez egy hamis Logarlécet, amit majd a játékos (Hallgató) felvesz.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Logarléc pozitív tulajdonságának hiánya Hamis tulajdonsága miatt. Nem várunk hibát.

- **Bemenet**

```
addRoom 5
addPlayer Karoly
addItem FakeLogarlec Room#1
setActiveCharacter Karoly
pickUp FakeLogarlec#1
```

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

ID: Karoly

Stunned for: 0

Drunk for: 0

Killed: false

Items of entity:

ID: FakeLogarlec#1

Health: 1

Pair: No pair.

Room: Room#1

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room:

ID: Karoly

Items of room: No items.

Doors of room: No doors.

8.2.14 Hallgató elhasznál egy Törlőrongyot

- **Leírás**

A tesztet létrehoz három Szobát és elhelyez az egyikben játékost Hallgatót, másikon pedig egy Oktatót. A játékos (Hallgató) elhasznál a szobájában egy törlőrongyot, majd átmegy abba a szobába, amiben nincs senki. Utána az Oktató átmegy a nedves szobába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A törlőrongy használata. Nem várunk hibát.

- **Bemenet**

```
addRoom 5
addRoom 5
addRoom 5
addDoor Room#1 Room#2
addDoor Room#2 Room#3
addPlayer Karoly Room#2
addItem WetSponge Karoly
addNPC Prof Room#3
getInfo
setActiveCharacter Karoly
activate WetSponge#1
move Room#1
setActiveCharacter Prof#1
move Room#2
```

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

ID: Karoly

Stunned for: 0

Drunk for: 0

Killed: false

Items of entity:

ID: WetSponge#1

Health: 1

Pair: No pair.

Room: Room#2

ID: Prof#1

Stunned for: 1

Killed: false

Items of entity: No items.

Room: Room#3

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room: No entites.

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1
Second Room: Room#2
Closed: false
Oneway: false
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
 ID: Karoly
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#1
 Second Room: Room#2
 Closed: false
 Oneway: false
 ID: Door#2
 First Room: Room#2
 Second Room: Room#3
 Closed: false
 Oneway: false
ID: Room#3
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
 ID: Prof#1
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#2
 Second Room: Room#3
 Closed: false
 Oneway: false

Game state: ongoing
Characters of Game:
 ID: Karoly
 Stunned for: 0
 Drunk for: 0
 Killed: false
 Items of entity:
 ID: WetSponge#1
 Health: 0
 Pair: No pair.

Room: Room#1
ID: Prof#1
Stunned for: 1
Killed: false
Items of entity: No items.
Room: Room#1
Rooms of Game:
ID: Room#1
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
ID: Karoly
Items of room: No items.
Doors of room:
ID: Door#1
First Room: Room#1
Second Room: Room#2
Closed: false
Oneway: false
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
ID: Prof#1
Items of room: No items.
Doors of room:
ID: Door#1
First Room: Room#1
Second Room: Room#2
Closed: false
Oneway: false
ID: Door#2
First Room: Room#2
Second Room: Room#3
Closed: false
Oneway: false
ID: Room#3
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room: No entites.
Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#2

Second Room: Room#3

Closed: false

Oneway: false

8.2.15 Hallgató már aktív Sör hatással elhasznál egy Sört

- **Leírás**

A tesztet létrehoz egy szobát és elhelyez benne egy játékost (Hallgató). A játékosnak (Hallgatónak) ad két Sör tárgyat és egy hamis FFP2-s maszkot. Mindezek után a söröket egymás után megissza a játékos (Hallgató).

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A játékos ha már sör hatása alatt áll és rá iszik még egy sört, akkor el dob egy tárgyat. Hibát nem várunk.

- **Bemenet**

```
addRoom 5
addPlayer Karoly Room#1
addItem Beer Karoly
addItem Beer Karoly
addItem FakeFFP2 Karoly
setActiveCharacter Karoly
activate Beer#1
activate Beer#2
```

- **Elvárt kimenet**

```
Game state: ongoing
Characters of Game:
  ID: Karoly
  Stunned for: 0
  Drunk for: 3
  Killed: false
  Items of entity:
    ID: Beer#1
    Health: 0
    Pair: No pair.
    ID: Beer#2
    Health: 0
    Pair: No pair.
  Room: Room#1
Rooms of Game:
  ID: Room#1
  Capacity: 5
  Toxic: false
  Wet: false
  Cursed: false
  Sticky: false
  Entities of room:
    ID: Karoly
  Items of room:
    ID: FakeFFP2#1
    Health: 1
    Pair: No pair.
  Doors of room: No doors.
```

8.2.16 Hallgató belép gázos szobába

- **Leírás**

A tesztet létrehoz egy gázos Szobát, egy sima Szobát és a simába elhelyez egy játékost (Hallgatót). A játékos (Hallgató) kap két darab Tranzisztort és egy Camembert. Ezután a sima Szobából átmozog a gázos Szobába a játékos (Hallgató).

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A gázos szoba negatív hatása a játékosra (Hallgatóra), mint entitásra. Hibát nem várunk.

- **Bemenet**

```
addRoom 5 toxic
addRoom 5
addDoor Room#1 Room#2
addPlayer Karoly Room#2
addItem Transistor Karoly
addItem Transistor Karoly
addItem Camembert Karoly
setActiveCharacter Karoly
move Room#1
```

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

```
ID: Karoly
Stunned for: 1
Drunk for: 0
Killed: false
Items of entity: No items.
Room: Room#1
```

Rooms of Game:

```
ID: Room#1
Capacity: 5
Toxic: true
Wet: false
Cursed: false
Sticky: false
Entities of room:
    ID: Karoly
Items of room: No items.
Doors of room:
    ID: Door#1
    First Room: Room#1
    Second Room: Room#2
    Closed: false
    Oneway: false
```

```
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
Items of room:
```

ID: Transistor#1

Health: 1

Pair: No pair.

Active: false

Room: No room.

ID: Transistor#2

Health: 1

Pair: No pair.

Active: false

Room: No room.

ID: Camembert

Health: 1

Pair: No pair.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false

Oneway: false

8.2.17 Hallgató belép gázos szobába FFP2-s maszkkal

- **Leírás**

A tesztet létrehoz egy gázos Szobát, egy sima Szobát és a simába elhelyez egy játékost (Hallgatót). A játékos (Hallgató) kap két darab Tranzisztort, egy FFP2-s maszkot és egy Camembert. Ezután a sima Szobából átmozog a gázos Szobába a játékos (Hallgató).

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- **Bemenet**

```
addRoom 5 toxic
addRoom 5
addDoor Room#1 Room#2
addPlayer Karoly Room#2
addItem FFP2 Karoly
addItem Transistor Karoly
addItem Transistor Karoly
addItem Camembert Karoly
setActiveCharacter Karoly
move Room#1
```

- **Elvárt kimenet**

```
Game state: ongoing
Characters of Game:
  ID: Karoly
  Stunned for: 0
  Drunk for: 0
  Killed: false
  Items of entity:
    ID: FFP2#1
    Health: 0
  Pair: No pair.
    ID: Transistor#1
    Health: 1
    Pair: No pair.
    Active: false
    Room: No room.
    ID: Transistor#2
    Health: 1
    Pair: No pair.
    Active: false
    Room: No room.
    ID: Camembert
    Health: 1
    Pair: No pair.
  Room: Room#1
Rooms of Game:
  ID: Room#1
  Capacity: 5
  Toxic: true
  Wet: false
  Cursed: false
```

Sticky: false
Entities of room:
 ID: Karoly
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#1
 Second Room: Room#2
 Closed: false
 Oneway: false
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room: No entities.
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#1
 Second Room: Room#2
 Closed: false
 Oneway: false

8.2.18 Oktató megtámadja a Hallgatót

- **Leírás**

A tesztet létrehozó Szobát, aztán az egyikben elhelyez egy játékost (Hallgatót) és elhelyez egy Oktatót a másikban. Utána az Oktató átmozog a játékost (Hallgató) Szobájába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az Oktató kibuktató képessége. Hibát nem várunk

- **Bemenet**

```
addRoom 5
addRoom 5
addPlayer Karoly Room#1
addNPC Prof Room#2
setActiveCharacter Prof#1
move Room#1
```

- **Elvárt kimenet**

Game state: lost

Characters of Game:

```
ID: Karoly
Stunned for: 0
Drunk for: 0
Killed: true
Items of entity: No items.
Room: Room#1
ID: Prof#1
Stunned for: 0
Killed: false
Items of entity: No items.
Room: Room#1
```

Rooms of Game:

```
ID: Room#1
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
    ID: Karoly
    ID: Prof#1
Items of room: No items.
Doors of room:
    ID: Door#1
    First Room: Room#1
    Second Room: Room#2
    Closed: false
    Oneway: false
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
```

Entities of room: No entities.

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false

Oneway: false

8.2.19 Oktató megtámadja Sör hatása alatt lévő Hallgatót

- **Leírás**

A tesztet létrehoz kettő Szobát és elhelyez az egyikben egy Oktatót és a másikban egy játékost (Hallgatót). A játékosnak (Hallgatónak) ad egy Sört, amit aztán meg is fog inni. Az Oktató aztán átmozog az játékos (Hallgató) szobájába

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A sör pozitív tulajdonságát, hatása alatt a játékost (Hallgatót) nem tudja kibuktatni az Oktató. Hibát nem várunk.

- **Bemenet**

```
addRoom 5
addRoom 5
addPlayer Karoly Room#1
addPlayer Beer Karoly
addNPC Prof Room#2
setActiveCharacter Karoly
activate Beer#1
setActiveCharacter Prof#1
move Room#1
```

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

ID: Karoly

Stunned for: 0

Drunk for: 3

Killed: false

Items of entity:

ID: Beer#1

Health: 0

Pair: No pair.

Room: Room#1

ID: Prof#1

Stunned for: 0

Killed: false

Items of entity: No items.

Room: Room#1

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room:

ID: Karoly

ID: Prof#1

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false
Oneway: false
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room: No entities.
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#1
 Second Room: Room#2
 Closed: false
 Oneway: false

8.2.20 Oktató megtámadja TVSZ-szel rendelkező Hallgatót

- **Leírás**

A tesztet létrehoz kettő Szobát és elhelyez az egyikben egy Oktatót és a másikban egy játékost (Hallgatót). A játékosnak (Hallgatónak) ad egy TVSZ-t. Az Oktató aztán átmegy az játékos (Hallgató) Szobájába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A TVSZ pozitív tulajdonságát teszteljük, amikor a játékosnál (Hallgatónál) van ilyen tárgy, akkor az Oktató nem tudja kibuktatni. Hibát nem várunk.

- **Bemenet**

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

ID: Karoly

Stunned for: 0

Drunk for: 0

Killed: false

Items of entity:

ID: TVSZ#1

Health: 0

Pair: No pair.

Room: Room#1

ID: Prof#1

Stunned for: 0

Killed: false

Items of entity: No items.

Room: Room#1

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room:

ID: Karoly

ID: Prof#1

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false

Oneway: false

ID: Room#2

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room: No entities.

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false

Oneway: false

8.2.21 Oktató megtámadja hamis TVSZ-szel rendelkező Hallgatót

- **Leírás**

A tesztet létrehoz kettő Szobát és elhelyez az egyikben egy Oktatót és a másikban egy játékos (Hallgatót). A játékosnak (Hallgatónak) ad egy hamis TVSZ-t. Az Oktató aztán átmegy az játékos (Hallgató) szobájába

- **Ellenőrzött funkcionalitás, várható hibahelyek**

TVSZ pozitív tulajdonságának hiánya Hamis tulajdonsága miatt. Nem várunk hibát.

- **Bemenet**

```
addRoom 5
addRoom 5
addNPC Prof Room#2
addPlayer Karoly Room#1
addItem FakeTVSZ Karoly
setActiveCharacter Prof#1
move Prof#1
```

- **Elvárt kimenet**

Game state: lost

Characters of Game:

```
ID: Karoly
Stunned for: 0
Drunk for: 0
Killed: true
Items of entity:
    ID: FakeTVSZ#1
    Health: 1
    Pair: No pair.
Room: Room#1
ID: Prof#1
Stunned for: 0
Killed: false
Items of entity: No items.
Room: Room#1
```

Rooms of Game:

```
ID: Room#1
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
    ID: Karoly
    ID: Prof#1
Items of room: No items.
Doors of room:
    ID: Door#1
    First Room: Room#1
    Second Room: Room#2
    Closed: false
    Oneway: false
ID: Room#2
```

Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room: No entities.
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#1
 Second Room: Room#2
 Closed: false
 Oneway: false

8.2.22 Takarító belép egy szobába, amiben van kettő Hallgató

- **Leírás**

A tesztet létrehoz két szobát, az egyikben elhelyez kettő játékost (Hallgatót), a másikban pedig egy Takarítót. A Takarító ezután átmegy a másik szobába, amelyben a játékosok (Hallgatók) tartózkodnak.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Takarító mikor egy olyan szobába lép, ahol van mozogni képes ember, akkor onnan kitesszékel mindenkit. Hibát nem várunk.

- **Bemenet**

```
addRoom 5
addRoom 5
addDoor Room#1 Room#2
addPlayer Karoly Room#1
addPlayer Bea Room#1
addNPC CleaningLady Room#2
setActiveCharacter CleaningLady
move Room#1
```

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

```
ID: Karoly
Stunned for: 0
Drunk for: 0
Killed: false
Items of entity: No items.
Room: Room#2
ID: Bea
Stunned for: 0
Drunk for: 0
Killed: false
Items of entity: No items.
Room: Room#2
ID: CleaningLady#1
Stunned for: 0
Killed: false
Items of entity: No items.
Room: Room#1
```

Rooms of Game:

```
ID: Room#1
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
    ID: CleaningLady#1
Items of room: No items.
Doors of room:
    ID: Door#1
    First Room: Room#1
```

Second Room: Room#2
Closed: false
Oneway: false
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
 ID: Karoly
 ID: Bea
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#1
 Second Room: Room#2
 Closed: false
 Oneway: false

8.2.23 Takarító belép egy nedves szobába, amiben van kettő Oktató

- **Leírás**

A tesztet létrehoz három szobát, az egyikben elhelyez kettő Oktatót, a másikban pedig egy Takarítót. A Takarító ezután átmozog a másik szobába, amelyben a Oktatók tartózkodnak.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Takarító mikor egy olyan szobába lép, ahol van mozogni képes ember, akkor onnan kitessekel mindenkit. Ebben az esetben az Oktatók meg vannak bénulva. Hibát nem várunk.

- **Bemenet**

```
addRoom 5
addRoom 5 wet
addRoom 5
addDoor Room#1 Room#2
addDoor Room#2 Room#3
addNPC Prof Room#1
addNPC Prof Room#1
addNPC CleaningLady Room#3
setActiveCharacter CleaningLady
move Room#1
```

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

```
ID: Prof#1
Stunned for: 1
Drunk for: 0
Killed: false
Items of entity: No items.
Room: Room#2
ID: Prof#2
Stunned for: 1
Drunk for: 0
Killed: false
Items of entity: No items.
Room: Room#2
ID: CleaningLady#1
Stunned for: 0
Killed: false
Items of entity: No items.
Room: Room#1
```

Rooms of Game:

```
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room: No entities.
Items of room: No items.
Doors of room:
  ID: Door#1
  First Room: Room#1
```

Second Room: Room#2
Closed: false
Oneway: false
ID: Room#2
Capacity: 5
Toxic: false
Wet: true
Cursed: false
Sticky: false
Entities of room:
 ID: CleaningLady#1
 ID: Prof#1
 ID: Prof#2
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#1
 Second Room: Room#2
 Closed: false
 Oneway: false
 ID: Door#2
 First Room: Room#2
 Second Room: Room#3
 Closed: false
 Oneway: false

ID: Room#3
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room: No entities.
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#2
 Second Room: Room#3
 Closed: false
 Oneway: false

8.2.24 Takarító gázos szobába lép

- **Leírás**

A tesztet létrehoz egy gázos és egy sima Szobát, az egyikben elhelyez egy Takarítót. A Takarító ezután átmozog a másik szobába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Takarító gázos Szobába lépve kiszellőztet, ezáltal nem lesz többé gázos Szoba. Hibát nem várunk.

- **Bemenet**

addRoom 5 toxic

addRoom 5

addDoor Room#1 Room#2

addNPC CleaningLady Room#2

setActiveCharacter CleaningLady

move Room#1

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

ID: CleaningLady#1

Stunned for: 0

Killed: false

Items of entity: No items.

Room: Room#1

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room:

ID: CleaningLady#1

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false

Oneway: false

ID: Room#2

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room: No entities.

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false
Oneway: false

8.2.25 Két szoba egyesül

- **Leírás**

A játék létrehoz három Szobát és kettő között egy Ajtót. Létrehozásuk után két szobát egyesít.

- **Ellenőrzött funkcionalitás, várható hibahelyet**

A két szoba tud egyesülni egymással, ekkor az ajtajaik egy jönnek a szobáknak. Hibát nem várunk.

- **Bemenet**

addRoom 5

addRoom 5

addRoom 5

addDoor Room#1 Room#3

merge Room#1 Room#2

- **Elvárt kimenet**

Game state: ongoing

Characters of Game: No characters.

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room: No entities.

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false

Oneway: false

ID: Room#2

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room: No entities.

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false

Oneway: false

8.2.26 Két szoba szétválik

- **Leírás**

Létrehoz két Szobát és kettejük között egy Ajtót. Létrehozásuk után az egyik szobát szétválasztja.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Egy Szoba szétválását, amikor is az egyik Szoba meg is kapja az egyik Ajtót. Hibát nem várunk.

- **Bemenet**

addRoom 5

addRoom 5

addDoor Room#1 Room#2

split Room#1

- **Elvárt kimenet**

Game state: ongoing

Characters of Game: No characters.

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room: No entities.

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false

Oneway: false

ID: Room#2

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room: No entities.

Items of room: No items.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false

Oneway: false

ID: Room#3

Capacity: 5

Toxic: false

Wet: false

Cursed: false

Sticky: false

Entities of room: No entities.

Items of room: No items.

Doors of room: No doors

8.2.27 Gázos és elátkozott szoba egyesül

- **Leírás**

Létrehoz 2 szobát. Az egyik szoba gázos, a másik szoba elátkozott állapotú. Létrehozásuk után egyesülnek.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrzi a két szoba egyesülésének sikerességét. A végkifejlet egy szoba a mindkét állapottal.

- **Bemenet**

addRoom 5 toxic

addRoom 5 cursed

merge

- **Elvárt kimenet**

Game state: ongoing

Characters of Game: No characters.

Rooms of Game:

ID: Room#1

Capacity: 5

Toxic: true

Wet: false

Cursed: true

Sticky: false

Entities of room: No entities.

Items of room: No items.

8.2.28 Szoba ragacsossá válik

- **Leírás**

A tesztet létrehoz három Szobát, amiből a második az mérgező Szoba. Az első és második, illetve második és harmadik szoba között egy Ajtót is létrehoz. Elhelyez az elsőben egy Takarítót, a harmadikban pedig három játékos (Hallgatót). A Takarító nő átmegy a gázos szobába, amit kitakarít. Utána a Takarító visszamegy az első szobába, és a 3 Hallgató egyenként átmegy a második szobába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Egy takarítás után 3 Entitás látogatására ragacsossá válik a szoba. Hibát nem várunk.

- **Bemenet**

```
addRoom 5
addRoom 5 toxic
addRoom 5
addDoor Room#1 Room#2
addDoor Room#2 Room#3
addNPC CleaningLady Room#1
addPlayer Balint Room#3
addPlayer Bela Room#3
addPlayer Ervin Room#3
setActiveCharacter CleaningLady#1
move Room#2
move Room#1
setActiveCharacter Balint
move Room#2
setActiveCharacter Bela
move Room#2
setActiveCharacter Ervin
move Room#2
```

- **Elvárt kimenet**

Game state: ongoing

Characters of Game:

```
ID: CleaningLady#1
Stunned for: 0
Killed: true
Items of entity: No items.
Room: Room#2
ID: Akos
Stunned for: 0
Drunk for: 0
Killed: true
Items of entity: No items.
Room: Room#2
ID: Prof#1
Stunned for: 0
Killed: false
Drunk for: 0
Items of entity: No items.
Room: Room#2
```

Rooms of Game:

```
ID: Room#1
```

Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
 ID: CleaningLady#1
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#1
 Second Room: Room#2
 Closed: false
 Oneway: false
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
 ID: Balint
 ID: Bela
 ID: Ervin
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#1
 Second Room: Room#2
 Closed: false
 Oneway: false
 ID: Door#2
 First Room: Room#2
 Second Room: Room#3
 Closed: false
 Oneway: false
ID: Room#3
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room: No entities.
Items of room: No items.
Doors of room:
 ID: Door#2
 First Room: Room#2
 Second Room: Room#3
 Closed: false
 Oneway: false

8.2.29 Minden Hallgató kibukik

- **Leírás**

Ez a tesztet létrehoz két szobát, az egyikbe elhelyez egy Oktatót, a másikba pedig 3 játékost (Hallgatót). Létrehozásuk után az Oktató átlép a másik szobába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Az oktató minden védtelen hallgatót kibuktat automatikusan, akivel egy szobába került. Ekkor a játék is elvesztődik. Hibát nem várunk.

- **Bemenet**

```
addRoom 5
addRoom 5
addDoor Room#1 Room#2
addNPC Prof Room#1
addPlayer Daniel Room#2
addPlayer Akos Room#2
setActiveCharacter Prof#1
move Room#2
```

- **Elvárt kimenet**

Game state: lost

Characters of Game:

```
ID: Daniel
Stunned for: 0
Drunk for: 0
Killed: true
Items of entity: No items.
Room: Room#2
ID: Akos
Stunned for: 0
Drunk for: 0
Killed: true
Items of entity: No items.
Room: Room#2
ID: Prof#1
Stunned for: 0
Killed: false
Items of entity: No items.
Room: Room#2
```

Rooms of Game:

```
ID: Room#1
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room: No entities.
Items of room: No items.
Doors of room:
  ID: Door#1
  First Room: Room#1
  Second Room: Room#2
  Closed: false
```

Oneway: false
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room:
 ID: Daniel
 ID: Akos
 ID: Prof#1
Items of room: No items.
Doors of room:
 ID: Door#1
 First Room: Room#1
 Second Room: Room#2
 Closed: false
 Oneway: false

8.2.30 Már megivott Sört újramegiszunk

- **Leírás**

Ez a teszt eset létrehoz egy szobát és benne egy játékost (Hallgatót). A játékosnak (Hallgatónak) ad egy darab Sört. Aztán elhasználja egymás után kétszer a Sört a játékos (Hallgató).

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrzi azt, hogy egy elhasznált tárgyat nem lehet még egyszer elhasználni. Hibát kap ekkor arra, hogy nem lehet ezt elvégezni.

- **Bemenet**

```
addRoom 5
addPlayer Adam Room#1
addItem Beer Adam
setActiveCharacter Adam
activate Beer#1
activate Beer#1
```

- **Elvárt kimenet**

Error: This item already been used.

Game state: ongoing

Characters of Game:

```
ID: Adam
Stunned for: 0
Drunk for: 3
Killed: false
Items of entity:
  ID: Beer#1
  Health: 0
  Pair: No pair.
Room: Room#1
```

Rooms of Game:

```
ID: Room#1
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room: No entities.
Items of room: No items.
Doors of room: No doors.
```

8.2.31 Ragacsos szobában tárgy felvétel

- **Leírás**

Létrehoz két szobát, amik között van ajtó. Az egyik szoba ragacsos állapotú és elhelyez egy tárgyat benne. A nem ragacsos szobához hozzáad egy játékos. A játékos átlép a ragacsos szobába és megpróbálja felvenni az ott lévő tárgyat.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ellenőrzi, hogy a tárgyat nem lehet felvenni ragacsos szobában. Elvárt kimenetként pedig hibaüzenetet kapunk: Item can not be picked up right now.

- **Bemenet**

```
addRoom 5
addRoom 5 sticky
addDoor Room#1 Room#2
addItem Camembert Room#1
addPlayer Otto Room#2
setActiveCharacter
move Room#1
pickup Item#1
```

- **Elvárt kimenet**

Error: You can't pick up an item in a sticky room.

Game state: ongoing

Characters of Game:

```
ID: Otto
Stunned for: 0
Drunk for: 0
Killed: true
Items of entity: No items.
Room: Room#1
```

Rooms of Game:

```
ID: Room#1
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: false
Entities of room: No entities.
Items of room: No items.
```

Doors of room:

```
ID: Door#1
First Room: Room#1
Second Room: Room#2
Closed: false
Oneway: false
```

```
ID: Room#2
Capacity: 5
Toxic: false
Wet: false
Cursed: false
Sticky: true
Entities of room:
ID: Otto
```

Items of room:

ID: Camembert#1

Health: 0

Pair: No pair.

Doors of room:

ID: Door#1

First Room: Room#1

Second Room: Room#2

Closed: false

Oneway: false

8.3 A tesztelést támogató programok tervei

A programot test argumentummal indítva tesztelő módban indul. Ezután a program kilistázza a teszteket és egy adott teszt sorszámának kiválasztásával és beírásával, majd az enter lenyomásával lefut az adott sorszámú teszt. A teszt lefutása során a program mappájában a test mappában a teszt sorszámának megfelelő mappában egy output.txt fájlban megjelenik a teszt kimenete. (pl 11-es teszt esetén program mappa/test/11/output.txt) Az elvárt kiment az adott teszthez ugyanebben a mappában expected.txt néven található. (pl. program mappa/test/11/expected.txt) Ezután a felhasználó az általa preferált programmal összehasonlíthatja.

Windows esetén a parancssoros fc paranccsal érdemes összehasonlítani a két fájlt. Az fc parancs két paraméterének az output és expected txt fájlokat, kell megadni majd futtatva megjelenik az összehasonlítás.

8.4 Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.04.13. 09:00	4 óra	Nagy Sánta	Osztályleírások
2024.04.13. 19:00	3 óra	Nagy Sánta	Osztályleírások
2024.04.13. 12:00	1 óra	Papp	Segítség osztályleírásokhoz
2024.04.13. 13:00	2,5 óra	Papp	Tesztesetek írása
2024.04.13. 15:30	0,5 óra	Papp	Hibajavítások írása
2024.04.14. 13:00	1 óra	Nagy	Pszeudokód írása
2024.04.14. 17:00	2 óra	Papp	Osztályleírás, hozzátartozó pszeudokód írása
2024.04.14. 19:00	3 óra	Nagy	Pszeudokód írása
2024.04.14. 19:00	4 óra	Gulybán	Tesztesetek írása
2024.04.14. 19:00	3 óra	Csabuda	Tesztesetek írása
2024.04.14. 23:00	1 óra	Nagy	Osztályleírás
2024.04.15. 09:00	1 óra	Nagy	Osztályleírás, Tesztelést támogató programok tervei
2024.04.15. 09:00	2 óra	Papp	Controller osztály megírása, Formázás, naplózás