

8. Részletes tervek

53 – kedvenc_csapatom

Konzulens:

Potyók Csaba

Csapattagok

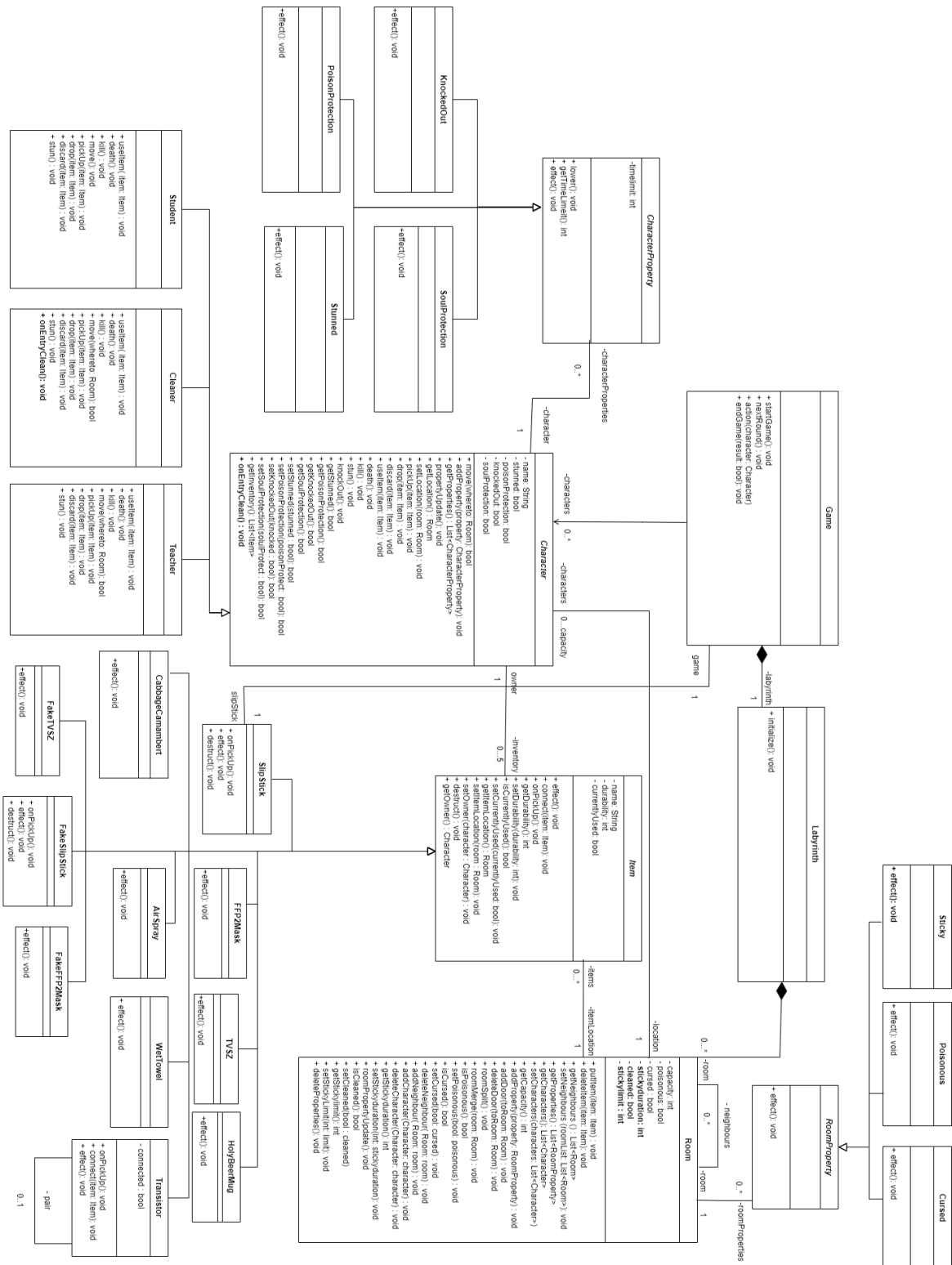
Fazekas Bence Mihály	W0TH54	bencemisi2003@gmail.com
Molnár Botond Kristóf	H1XYPA	molnarboti2003@gmail.com
Simon Tímea	SZFS1V	simontiti1109@gmail.com
Juhász Gábor	VA2469	ju.gabo14@gmail.com
Rimmel Botond	DMPMNZ	rimmelbotond@gmail.com

2024-04-15

8. Részletes tervek

Osztályok és metódusok terveí.

A függvények és attribútumok láthatóságát a nevük előtti szimbólum jelzi. Ezek:



- +: public**
- :private**
- # :protected**

8.1.1 AirSpray

- **Felelősség**

Az osztály a Légrfrissítő tárgy viselkedését valósítja meg. Felelőssége, hogy ha egy hallgató használja ezt tárgyat, akkor a szoba ne legyen többé mérges gázzal teli.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatának a számát jelez, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok**

- **+void effect():** Ez a metódus lekérdezi hogy az adott szoba “cursed” vagy “cleaned”-e, majd a “deleteProperties()” függvénnyel kitöröljük a szoba “RoomProperties” listáját, így nem lesz többé “poisonus”.Ezután létrehozunk új “Sticky” és “Cursed” típusú objektumokat és az “addProperty()” metódussal visszarakjuk azokat, amelyek a lekérdezésnél igazak voltak.

8.1.2 CabbageCamambert

- **Felelősség**

Az osztály a Dobozos káposztás camembert tárgy viselkedését valósítja meg. Felelőssége, hogy a hallgatók az osztály felhasználásával a szobának a típusát meg tudják változtatni mérgezőre.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatának a számát jelez, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok**

- **+void effect() :** A metódus meghívására létre jön, egy "Poisonous" típusú objektum, amely ezután hozzáadódik a szoba (amelyben a használó tartózkodik) "roomProperty" listájához, és meghívja ezen objektum "effect()" metódusát, ami beállítja a szoba poisonous tagváltozójának értékét "true"-ra.

8.1.3 Character

- **Felelősség**

Ez egy absztrakt osztály, amely a játékban szereplő karaktereknek az absztrakt osztálya.

- **Ősosztályok**

Nem rendelkezik ősosztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** A karakter neve.
- **-location:** Azon szoba amelyben a karakter tartózkodik.
- **-inventory:** A karakternél lévő tárgyak tárolódnak itt.
- **-characterProperties:** A karakterre érvényben lévő állapotokat tárolja.
- **-stunned:** A karakter bénított állapotát jelző érték.
- **-knockedOut:** A karakter eszméletvesztett állapotát jelző érték.
- **-poisonProtection:** A karakter mérge elleni védetségét jelző érték.
- **-soulProtection:** A karakter oktató lélek kiszívása elleni védetségét jelző érték.

- **Asszociációk**
- **kétirányú kapcsolat a Character osztályból a CharacterProperties osztály felé:** A Character és a CharacterProperties osztályok közötti kapcsolatot valósítja meg. A karakter tudja magáról milyen státuszai vannak.
- **kétirányú kapcsolat a Character osztályból az Item osztály felé:** A Character és az Item osztályok közötti kapcsolatot valósítja meg. A karakter tudja milyen tárgyak vannak az eszköztárában, de egyidőben egy karakternél legfeljebb 5 Item lehet.
- **kétirányú kapcsolat a Room és a Character osztályok közt:** A karakterek tudják melyik szobában vannak, a szobák pedig azt hogy mely karakterek tartózkodnak bennük, és egy karakter egy időben csak egy szobában lehet. Egy szobában egy időben a szoba befogadóképességével megegyező számú karakter lehet.
- **egyirányú kapcsolat a Game osztályból a Character osztály felé:** a Game osztály számon tartja a játékban részt vevő karaktereket.
- **Metódusok**
- **+bool move(Room whereto) :** A karakter mozgását valósítja meg, abba a szobába, amely paraméterként megvan adva a függvénynek. Amennyiben a paraméterként kapott szoba nem szomszédos a karakter tartózkodási helyével(“getNeighbours()” függvény segítségével) nem történik semmi. A szobába lépés után a régi szobának meghívódik a “deleteCharacter()”, míg az új szobának az “addCharacter()” nevű függvénye az éppen mozgó karakterrel mint paraméterrel. Ezután átállítja a karakter “location”-át az új szobára a “setLocation()” függvénnyel amit az új szobával, mint paraméterrel meghív. Végül meghívódik a karakter “OnEntryClean()” függvénye. (Ez csak a takarítónál csinál valamit)(absztrakt)
- **+void addProperty(CharacterProperty property) :** A karakterre vonatkozó hatásokat lehet vele állítani, oly módon, hogy a paraméterként megkapott “CharacterProperty”-t az adott Character “characterProperties” listájához adja, majd a hozzáadott objektumnak meghívja az “effect()” metódusát, mellyel beállítja a megfelelő bool tagváltozó (stunned, knockedOut, poisonProtection, soulProtection) értékét “true”-ra.
- **+List<CharacterProperty> getProperties() :** Ez egy Getter. Az osztály “characterProperties” nevű tagváltozóját adja vissza.
- **+void propertyUpdate() :** A karakter stunned, knockedOut, poisonProtection, soulProtection nevű tagváltozóinak beállítására szolgál, a “characterProperties” lista alapján. Azon végigmenve, meghívja minden tag “effect()” függvényét.
- **+Room getLocation() :** Ez egy Getter. Az osztály “location” nevű tagváltozóját adja vissza.
- **+void setLocation(Room room) :** Ez egy Setter. Az osztály “location” nevű tagváltozóját állítja be a paraméterként kapott szobára.
- **+void pickUp(Item item) :** A karakter felveszi a szobában a paraméterként kapott tárgyat, így a tárgy bekerül a karakter “inventory” tagváltozójába (absztrakt).
- **+void drop(Item item) :** A karakter az “inventory” tagváltozójából kitörli a paraméterként kapott tárgyat és hozzáadja a karakter “location” tagváltozójába tárolt szoba “items” listájába.(absztrakt).
- **+void discard(Item item) :** A karakter az “inventory” tagváltozójából elégeti a paraméterként megadott tárgyat, amely ennek hatására kikerül az “inventory”-jából(absztrakt).
- **+void death() :** Ha a hallgatónak a lelkét kiszívják, akkor ez a metódus felel a hallgató játékból való kieséséért, ha pedig ezt a függvényt egy oktató hívja meg akkor nem történik semmi. A hallgató esetében ez a függvény a “setLocation()” hívja “null”

paraméterrel, ezzel átállítva a hallgató helyét semmire. Így jelezzük hogy ez a hallgató már nincsen a labirintusban(és a játékban sem.)

- **+void kill():** Ezzel a metódussal képes egy oktató egy hallgató lelkét kiszívni- azaz meghívja a hallgató “death()” metódusát-, ha ugyanabban a szobában tartózkodnak, és ha a hallgató nem rendelkezik megfelelő védelemmel, ha ezt egy “Student” vagy egy “Cleaner” osztály hívja, akkor nem történik semmi(absztrakt).
- **+void useItem(Item item):** Ennek a függvénynek a segítségével képes egy hallgató, az “inventory” -jában szereplő tárgyak közül egyet használni, azonban ha egy oktató vagy egy takarító próbálna meg ezt a metódust használni, akkor nem történik semmi. Ez a metódus meghívja majd az adott tárgynak az “effect()” függvényét, hogy azok kifejtsék hatásukat(absztrakt).
- **+void propertyUpdate():** A karakterre érvényben lévő hatások értékeit frissíti. Végigmegy a “characterProperties” listán és minden elemének meghívja az “effect()” függvényt.
- **+void onEntryClean():** Ennek a metódusnak, csak a Cleaner osztályban van hatása, különben csak visszatér.
- **+void stun() :** A karakter bénult állapotában hívódik meg, kezeli a karakter bénult állapotában való tevékenységét.
- **+void knockOut() :** A karakter eszméletvesztett állapotában hívódik meg, kezeli a karakter eszméletvesztett állapotában lévő tevékenységét. Ez a függvény hívja meg a karakter “drop(Item item)” függvényét ahol az “item” paraméter a hallgatónál lévő tárgy. Ez annyiszor hívódik ahány tárgy van a hallgatónál.
- **+bool getStunned():** Ez egy Getter. Visszaadja a “stunned” tagváltozó értékét.
- **+bool getPoisnProtection():** Ez egy Getter. Visszaadja a “poisonProtection” tagváltozó értékét.
- **+bool getKnockedOut():** Ez egy Getter. Visszaadja a “knockedOut” tagváltozó értékét.
- **+bool getSoulProtection():** Ez egy Getter. Visszaadja a “soulProtection” tagváltozó értékét.
- **+void setStunned(bool stunned):** Ez egy Setter. Beállítja a “stunned” változó értékét a paraméterben kapott értékre.
- **+void setPoisnProtection(bool poisonProtect):** Ez egy Setter. Beállítja a “poisonProtection” változó értékét a paraméterben kapott értékre.
- **+void setKnockedOut(bool knockedOut):** Ez egy Setter. Beállítja a “knockedOut” változó értékét a paraméterben kapott értékre.
- **+void setSoulProtection(bool soulProtection):** Ez egy Setter. Beállítja a “soulProtection” változó értékét a paraméterben kapott értékre.
- **+String getName():** Visszaadja a karakter nevét.

8.1.4 CharacterProperty

- **Felelősség**

Olyan absztrakt osztály, melynek felelőssége az, hogy a segítségével a karakterek státuszait nyilván tudjuk tartani.

- **Ősosztályok**

Nem rendelkezik őssosztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-timelimit:** megadja, hogy az adott státusz mennyi időegységig érvényes
- **-character:** Az a Character akire a hatást kifejtik

- **Asszociációk**

- **kétirányú kapcsolat a Character osztályból a CharacterProperties osztály felé:**

A Character és a CharacterProperties osztályok közötti kapcsolatot valósítja meg. A karakter tudja magáról milyen státuszai vannak.

- **Metódusok:**

- **+ void lower():** Ezzel a metódussal csökkentjük a timelimit tagváltozó értékét, ha 0 akkor nem csökkenti tovább..
- **+ int getTimeLimit():** Ez egy Getter. Visszaadja a timelimit tagváltozó értékét.
- **+ void effect():** A CharacterProperty leszármazottaiban van megvalósítva, működése során a leszármazottnak megfelelő Character tagváltozó(stunned, knockedOut, poisonProtection, soulProtection) értékét "true"-ra állítja állítja a leszármazottakban leírt módon.(absztrakt függvény)

8.1.5 Cleaner

- **Felelősség**

Ez az osztály felel a takarító viselkedésének a megvalósításáért.

- **Ősosztályok**

Character

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a takarítóneve (örökölt attribútum).
- **-location:** Azon szoba amelyben a karakter tartózkodik (örökölt attribútum).
- **-inventory:** A karakternél lévő tárgyak tárolódnak itt (örökölt attribútum).
- **-stunned:** A karakter bénított hatását jelző érték (örökölt attribútum).
- **-knockedOut:** A karakter eszméletvesztett hatását jelző érték (örökölt attribútum).
- **-poisonProtection:** A karakter mérge elleni védetségét jelző érték (örökölt attribútum).
- **-soulProtection:** A karakter oktató lélek kiszívása elleni védetségét jelző érték (örökölt attribútum).
- **-characterProperties:** A karakterre érvényes hatások tárolja (örökölt attribútum).

- **Asszociációk**

Nem vesz részt asszociációban.

- **Metódusok**

- **+bool move(Room whereto) :** A takarító mozgását valósítja meg, abba a szobába, amely paraméterként megvan adva a függvénynek. Amennyiben a paraméterként kapott szoba nem szomszédos a karakter tartózkodási helyével ("getNeighbours()") függvény segítségével) nem történik semmi. A szobába lépés után a régi szobának meghívódik a "deleteCharacter()", míg az új szobának az "addCharacter()" nevű függvénye az éppen mozgó karakterrel mint paraméterrel. Ezután átállítja a takarító "location"-át az új szobára a "setLocation()" függvénnyel amit az új szobával, mint paraméterrel meghív. Végül meghívódik a karakter "OnEntryClean()" függvénye.
- **+void pickUp(Item item) :** Mivel egy takarító nem képes tárgyat felvenni, emiatt ez a metódus nem csinál semmit, csak visszatér.
- **+void drop(Item item) :** Mivel egy takarító nem képes tárgyat eldobni, emiatt ez a metódus nem csinál semmit, csak visszatér.
- **+void discard(Item item) :** Mivel egy takarító nem képes tárgyat elégetni, emiatt ez a metódus nem csinál semmit, csak visszatér.
- **+void death():** Mivel egy takarító nem tud meghalni, emiatt ez a metódus nem csinál semmit, csak visszatér.
- **+void useItem(Item item):** Mivel egy takarító nem tud tárgyat használni, emiatt ez a metódus nem csinál semmit, csak visszatér.
- **+void kill():** Mivel egy takarító nem képes lelket kiszívni, emiatt ez a metódus nem csinál semmit, csak visszatér.

- **+void stun():** Mivel egy takarító nem képes megbénulni, emiatt ez a függvény nem csinál semmit, csak visszatér.
- **+void onEntryClean():** Ennek a módszernek a segítségével érjük el azt, hogy ha egy Cleaner belép egy szobába, akkor a szobában lévőket akik nem ájultak vagy bénultak azokat átrakja egy szomszédos szobába, és ezzel egyidejűleg gáztalanítsa és ragadóssá tegye az adott szobát. Ha nem képes áttenni a karaktereket más szobába(mert tele vannak a szobák) akkor helyben hagyja őket.

A függvény hívásakor a szoba “characters” listáján végigmenve lekérdezzük az egyes karakterek állapotát a “GetKnockedOut()” és “GetStunned” getter függvényekkel. Ha mindkét getter “false” értékkel tér vissza akkor átállítjuk a karakter “location” változóját egy szomszédos szobára, amennyiben annak a kapacitása megengedi. Ezzel egy időben az eredeti szobának meghívjuk a “deleteCharacter()” függvényét az áthelyezni kívánt karakter paraméterrel. Amennyiben nincsen megfelelő szomszédos szoba akkor a karakterek maradnak az eredeti szobában.

Az összes karakter vizsgálata után a függvény, amennyiben a szoba “isCleaned()” metódusa “false” értékkel tér vissza létrehoz egy “Sticky” típusú objektumot és meghívja a szoba “addProperty()” függvényét a létrehozott objektummal, mint paraméterrel. Ezután a “setStickyDuration()”, “setCleaned()”, “setStickyLimit()” függvények hívódnak.

Sorrendben ezek paraméterei:

- egy szám, amely azt adja meg meddig nem lehet a szobából tárgyakat felvenni miután ragadós lett.
- “true” mivel a szobát kitakarították és ragadóssá válhat
- egy szám , amely azt adja meg hány karakter belépés után lesz ragacsos a szoba

Majd a függvény visszatér.

8.1.6 Cursed

- **Felelősség**

Az elátkozott tulajdonság rendelhető segítségével a szobákhoz.

- **Ósosztályok**

RoomProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**
- Nincsen attribútuma.

- **Asszociációk**
- Nem vesz részt asszociációban.

- **Metódusok:**

- **+ void effect():** Ez a metódus meghívja annak a szobának a “SetCursed(bool c)” metódusát amelyhez tartozik, és így beállítja a “cursed” tagváltozót a paraméterben kapott értékre, amely ez esetben “true” lesz.

8.1.7 FakeFFP2Mask

- **Felelősség**

Az osztály a hamis FFP2-es maszk tárgy viselkedését valósítja meg. felelőssége, hogy megtévessze a hallgatókat, abban, hogy védettséget szereznek a mérges gáz ellen.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatának a számát jelzi, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok**

- **+void effect() :** Ez a metódus felel azért, hogy ha egy hallgató használná a "FakeFFP2Mask" nevű tárgyat, az ne fejtse ki az eredeti "FFP2Mask" nevű tárgy hatását, hanem csak hívja meg a tárgy "discard()" metódusát, amely így törlésre kerül az "inventory"-ből.

8.1.8 FakeSlipStick

- **Felelősség**

Az osztály a hamis SlipStick tárgy viselkedését valósítja meg. felelőssége, hogy megtévessze a hallgatókat, abban, hogy megnyerik a játékot.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatának a számát jelzi, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok**

- **+void onPickUp():** Ha ezt a tárgyat felveszi egy karakter, akkor lép működésbe ez a metódus. Ha egy karakter fel veszi ezt a tárgyat, akkor meghívódik a tárgy “discard()” metódusa, amely így törlésre kerül.
- **+void destruct():** Ez a metódus biztosítja, hogy a tárgy törlődik a karakter “inventory” listájából, és beállítja az owner és a itemLocation tagváltozókat null-ra.

8.1.9 FakeTVSZ

- **Felelősség**

Az osztály a hamis TVSZ tárgy viselkedését valósítja meg. felelőssége, hogy megtévessze a hallgatókat, abban, hogy védettséget szereznek a hallgatók ellen.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatának a számát jelzi, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**
- Nem vesz részt asszociációban.
- **Metódusok**
- **+void effect() :** Ez a metódus felel azért, hogy ha egy hallgató használná a “FakeTVSZ” nevű tárgyat, az ne fejtse ki az eredeti “TVSZ” nevű tárgy hatását, hanem csak hívja meg a tárgy “discard()” metódusát, amely így törlésre kerül az “inventory”-ből.

8.1.10 FFP2Mask

- **Felelősség**
Az osztály az FFP2-es maszk tárgy viselkedését valósítja meg. felelőssége, hogy a hallgatók az osztály felhasználásával a védeltséget kapjanak a mérges gáz hatása ellen.

- **Össztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**
- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatának a számát jelzi, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**
- Nem vesz részt asszociációban.

- **Metódusok**
- **+void effect() :** Ez a metódus meghívja a tárgyat használó hallgató “SetPoisonProtection(bool p)” metódusát, amellyel beállítjuk a hallgató “poisonProtection” tagváltozóját a paraméterben megadottnak, amely ez esetben “true” lesz és így elérjük, hogy a hallgató védetté váljon a mérges gáztól.

8.1.11 Game

- **Felelősség**

Az osztály a játék indításáért és a körök menedzseléséért felelős, illetve lehetővé teszi a karakterek számára, hogy akciót hajtsanak végre a körön belül (azaz “sorra kerüljenek”). A játék végének kezelése is az osztály felelőssége.

- **Ősosztályok**

Nem rendelkezik ősosztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-characters:** a játékban részt vevő karakterek
- **-labyrinth:** a játétként szolgáló labirintus
- **-slipstick:** a célt jelentő Logarléc nevű tárgy

- **Asszociációk**

- **egyirányú kapcsolat a Game osztályból a Character osztály felé:** a Game osztály számon tartja a játékban részt vevő karaktereket.

- **kompozíció a Labyrinth osztályból a Game osztály felé:** a Game osztály számon tartja a labirintust, ha a Game megszűnik, akkor az egész labirintus is.

- **kétirányú kapcsolat a Game osztályból a SlipStick osztály felé:** A Game és a SlipStick osztályok közötti kapcsolatot valósítja meg. A slipstick tudja magáról milyen játékban van, és emiatt be tudja azt fejezni, ha egy hallgató felveszi.

- **Metódusok**

- **+void startGame():** játék indítása, kezdőállapot betöltése, megfelelő objektumok és állapotaik inicializálása.
- **+void nextRound():** következő kör indításának megvalósítása. Végigmegy a karakterek tömbjén és megvizsgálja azok helyzetét. Ha valahol “null” érték szerepel azt a

karaktert törli a listából egyébként meghívja az “action() függvényt” a karakterrel, mint paraméterrel.

- **+void endGame(bool result):** a játék végének kezelése attól függően, hogy ki nyert. A paraméter értéke “true” ha a hallgatók, “false” ha az oktatók. Kiír egy szöveget a konzolra, amit a nyerést szemlélteti.
- **+void action(Character character):** a karakterek akciójának megvalósítása. A bemeneti nyelven megadott parancsnak megfelelően a paraméterként kapott karakterrel végrehajt tevékenységet(pl.:mozgás, tárgyfelvétel stb...).

8.1.12 HolyBeerMug

- **Felelősség**

Az osztály a Szent söröspohár tárgy viselkedését valósítja meg. Felelőssége, hogy ha egy hallgató használja ezt tárgyat, akkor az megvédje őt egy adott ideig attól, hogy egy oktató kiszívja a lelkét.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**
- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatának a számát jelez, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok**

- **+void effect():** Ez a metódus meghívja a tárgyat használó hallgató “SetSoulProtection(bool p)” metódusát, amellyel beállítjuk a hallgató “soulProtection” tagváltozóját a paraméterben megadottnak, amely ez esetben “true” lesz, és így elérjük, hogy a hallgató védetté váljon az oktató támadásai ellen, illetve meghívódik a tárgyat használó karakternek a “dropItem(Item item)” függvénye, amelynek hatására eldob egy tárgyat, amennyiben a karakter “inventory” listája nem üres.

8.1.13 Item

- **Felelősség**

A játékban szereplő tárgyaknak egy közös absztrakt osztályt biztosít. A tárgyak hatáskifejtéséért is felelős.

- **Ősosztályok**

Nem rendelkezik ősosztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve.
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)
- **-durability:** a tárgy használhatóságát jelez, amely ha eléri a 0-át, akkor a tárgy használhatatlanná válik.
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).
- **-owner:** A karakter aki felvette a tárgyat.

- **Asszociációk**

- **kétirányú kapcsolat az Character osztályból az Item osztály felé:** A karakter tudja milyen tárgyak vannak az eszköztárában, de egyidőben egy karakternél legfeljebb 5 Item lehet.
- **kétirányú kapcsolat a Room osztályból az Item osztály felé:** A szobában elérhetőek a benne lévő tárgyak. Egy szobában akár mennyi tárgy lehet.

- **Metódusok**

- **+void effect() :** A tárgy hatását kifejtő függvény. Működést lásd a leszármazottaknál, ahol történik hatására valami.
- **+ void connect(Item item):** Transistorok-nál ez a metódus felel azért, hogy összekapcsolódjanak(részletesebb leírást lásd a Transistor osztálynál), a többi Item leszármazott esetében ez a metódus nem csinál semmit.
- **+ void onPickUp():** Tárgy felvételénél lefutó metódus, ami csak a "SlipStick", illetve a "Transistor" nevű tárgyaknál fejt ki hatást. Részletes leírást lásd az osztályoknál.
- **+int getDurability():** a "durability" attribútum értékét kapjuk vissza ezen getter metódus által.
- **+void setDurability(int durability):** ezzel a setter metódussal vagyunk képesek a "durability" tagváltozó értékét beállítani a paraméterben kapottra.
- **+bool isCurrentlyUsed() :** a "currentlyUsed" attribútum értékét kapjuk vissza ezen getter metódus által.
- **+void setCurrentlyUsed(bool currentlyUsed) :** ezzel a setter metódussal vagyunk képesek a "currentlyUsed" tagváltozó értékét beállítani a paraméterként kapottra.
- **+Room getItemLocation():** a "location" attribútum értékét kapjuk vissza ezen getter metódus által.
- **+void setItemLocation(Room room):** ezzel a setter metódussal vagyunk képesek a "location" tagváltozó értékét beállítani a paraméterként kapottra.
- **+void setOwner(Character character):** ezzel a setter metódussal vagyunk képesek a "Owner" tagváltozó értékét beállítani a paraméterként kapottra.

- **+void destruct():** a tárgy elégését valósítja meg. A tárgy törli magát a szoba “items” listájából, amennyiben egy karakter eszköztárában van akkor a karakter “inventory” listájából. Az Owner és a itemLocation tagváltozója pedig null lesz.
- **+void getOwner() :** az “Owner” attribútum értékét kapjuk vissza ezen getter metódus által.

8.1.14 KnockedOut

- **Felelősség**

A karakterek eszméletvesztés állapotának beállítására szolgál.

- **Ősosztályok**

CharachterProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**
- **-timelimit:** megadja, hogy az adott státusz mennyi időegységig érvényes(örökölt attribútum)
- **-character:** Az a Character akire a hatást kifejtik(örökölt attribútum)
- **Asszociációk**
- Nem vesz részt asszociációban.
- **Metódusok**
- **+void effect():** Ez a metódus meghívja annak a karakternek a “SetKnockedOut(bool k)” metódusát amelyik belépett egy mérges gázzal teli szobába és a “poisonProtection” tagváltozó értéke “false” volt, és így beállítja a “knockedOut” tagváltozót a paraméterben kapott értékre, amely ez esetben “true” lesz.

8.1.15 Labyrinth

- **Felelősség**

A játéktér létrehozásáért és a szobák nyilvántartásáért felelős.

- **Ősosztályok**

Nem rendelkezik ősosztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**
- **-room :** A játékban szereplő szobákat tartja nyilván.

Asszociációk

- **kompozíció a Room osztályból a Labyrinth osztály felé:** a Labyrinth osztály számon tartja a a játékban szereplő szobákat, ha a Labyrinth megszűnik, akkor az összes szoba is.
- **Metódusok**
- **+void initialize():** A labirintus felépítését, a kezdeti állapotának beállítását valósítja meg. Létrehozza a "Room" "Item" "Character" típusú objektumokat és elhelyezi őket a játék megkezdéséhez.
- **+void Getter():** A szobák kiíratásához használandó függvény.

8.1.16 PoisonProtection

- **Felelősség**
A karakterek mérgezés elleni védettségének beállítására szolgál.

- **Ősosztályok**

CharachterProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**
- **-timelimit:** megadja, hogy az adott státusz mennyi időegységig érvényes(örökölt attribútum)
- **-character:** Az a Character akire a hatást kifejtik(örökölt attribútum)

- **Asszociációk**
- Nem vesz részt asszociációban.

- **Metódusok**
- **+void effect():** Ez a metódus meghívja annak a karakternek a "SetPoisonProtection(bool p)" metódusát, aki épp elhasználta az "FFP2Mask" nevezetű tárgyat, és ez beállítja a "poisonProtection" tagváltozót a paraméterben kapott értékre, amely ez esetben "true" lesz.

8.1.17 Poisonous

- **Felelősség**

A mérgező tulajdonság rendelhető segítségével a szobákhoz.

- **Ősosztályok**

RoomProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- Nincsen attribútuma.

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok:**

- + **void effect()**: Ez a metódus meghívja annak a szobának a “SetPoisonous(bool c)” metódusát amelyhez tartozik, és így beállítja a “poisonous” tagváltozót a paraméterben kapott értékre, amely ez esetben “true” lesz.

8.1.18 Room

- **Felelősség**

A labirintust felépítő szobák és azok viselkedésének megvalósításáért felelős.

- **Ősosztályok**

Nem rendelkezik ősosztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-capacity**: A szobában tartózkodó karakterek maximális számát adja meg.
- **-roomProperties**: A szoba képességet meghatározó “RoomProperty”-eit tárolja.
- **-characters**: A szobában lévő karaktereket tartalmazza.
- **-neighbours**: A szoba szomszédait tárolja
- **-items**: A szobában lévő tárgyakat tartalmazza.
- **-poisonous**: A szoba mérgezett állapotát tároló érték.
- **-cursed**: A szoba elátkozott állapotát tároló érték.
- **-cleaned**: A szoba kitakarított állapotát tároló érték.
- **-stickylimit**: A szoba ragadóssá válásához szükséges karakter számot tartalmazza

- **-stickyduration:** Ez a tagváltozó mondja meg, meddig lesz ragadós az adott szoba.
- **Asszociáció**
- **egyirányú kapcsolat a Room osztályból a RoomProperty osztály felé:** A szoba tudja magáról milyen képességet adó típusa/típusai vannak
- **kétirányú kapcsolat a Room osztályból az Item osztály felé:** A szobában elérhetőek a benne lévő tárgyak. Egy szobában akár mennyi tárgy lehet.
- **kétirányú kapcsolat a Room és a Character osztályok közt:** A karakterek tudják melyik szobában vannak, a szobák pedig azt hogy mely karakterek tartózkodnak bennük. Egy karakter egy időben egy szobában tartózkodhat. Egy szobában egy időben egyszerre a szoba befogadóképességével megegyező karakterek lehet.
- **kompozíció a Room osztályból a Labyrinth osztály a felé:** A labirintus szobákból áll, egy szoba egy labirintushoz tartozhat. Ha a labirintus megsemmisül akkor a tartalmazott szobák is megszűnnek. Egy labirintus akárhány szobából állhat.
- **asszociáció a Room osztállyal:** A szobák ismerik a szomszédaikat.
- **Metódusok**
- **+void putItem(Item item) :** A szoba "items" nevezetű "Item" listájába belehelyeződik a paraméterként megadott "Item".
- **+void deleteItem(Item item):** Ennek a metódusnak a segítségével törlődik a szoba "items" listájából a paraméterként megadott "Item", ezzel érjük azt el, hogy ha felvesznek egy tárgyat a szobából, akkor az kitörlődjön.
- **+List<Room> getNeighbours():** Ez egy Getter, amely visszaadja a szoba szomszédaikat.
- **+void setNeighbours(List<Room> roomList):** Ez egy Setter, amellyel megadható paraméterként egy "Room-nak", hogy mely másik "Room"-okkal legyen összekötve.
- **+ List<RoomProperty> getProperties() :** Ez egy Getter, amellyel a "Room" "RoomProperties" listája kérdezhető le.
- **+ void addProperty(RoomProperty property):** Az adott "Room" "RoomProperties" listájához adhatunk hozzá még egy "RoomProperty"-t, amelyet paraméterként adtunk meg.
- **+void deleteProperties():** Ennek a metódusnak a hatására az adott szoba teljes "RoomProperties" listája törlődik, amely az "AirSpray", illetve a "Cleaner" osztályok működése miatt szükséges.
- **+void roomPropertyUpdate():** Ez a metódus felel azért, hogy minden körben megvizsgálja, hogy mennyi a "stickyduration" értéke, mivel ha az eléri a 0-át, akkor ki kell venni az adott "Room" "RoomProperties" listájából a "Sticky" "RoomProperty"-t
- **+void addOneWayDoor(Room room) :** Ezzel a metódussal tudjuk elérni, hogy az adott "Room" és a paraméterben megadott összekötődjenek.
- **+void deleteDoor(Room room) :** Ezzel a metódussal tudjuk elérni, hogy az adott "Room" és a paraméterben megadott között a kapcsolat megszakadjon.
- **+Room roomMerge(Room room) :** Az adott "Room" és a paraméterben kapott "Room" összemergelését valósítja meg, melynek hatására egy új "Room" keletkezik, amely rendelkezni fog az összemergelődő "Room"-ok tulajdonságaival.
- **+List<Room> roomSplit():** Ennek a metódusnak a hatására, az adott szoba ketté válik, olyan szobákra, melyek rendelkezni fognak ugyanazokkal a "RoomProperty"-kkel.
- **+bool IsPoisonous() :** Ez egy Getter, amely visszaadja a "poisonous" tagváltozó értékét.

- **+void SetPoisonous(bool p):** Ez egy Setter, amellyel betudjuk állítani a “poisonous” tagváltozó értékét a paraméterben megadottra
- **+bool IsCursed() :** Ez egy Getter, amely visszaadja a “cursed” tagváltozó értékét.
- **+void SetCursed(bool c):** Ez egy Setter, amellyel betudjuk állítani a “cursed” tagváltozó értékét a paraméterben megadottra
- **+bool IsCleaned():** Ez egy Getter, amely visszaadja a “cleaned” tagváltozó értékét.
- **+void SetCleaned(bool c):** Ez egy Setter, amellyel betudjuk állítani a “cleaned” tagváltozó értékét a paraméterben megadottra.
- **+List<Character> getCharacters():** Ez egy Getter, amely visszaadja a “characters” listában szereplő karaktereket.
- **+void setCharacters(List<Character> characters):** Ez egy Setter, amellyel betudjuk helyezni a paraméterben megadott karakter listát az adott szoba “characters” listájába, amely segítségével a kezdő állapot inicializálása elvégezhető.
- **+int getStickylimit():** Ez egy Getter, amely visszaadja a “stickylimit” tagváltozó értékét.
- **+void SetStickylimit(int limit):** Ez egy Setter, amellyel betudjuk állítani a “stickylimit” tagváltozó értékét a paraméterben megadottra.
- **+int getStickyduration():** Ez egy Getter, amely visszaadja a “stickyduration” tagváltozó értékét.
- **+void SetStickyduration(int duration):** Ez egy Setter, amellyel betudjuk állítani a “stickyduration” tagváltozó értékét a paraméterben megadottra.
- **+int getCapacity():** Ez egy Getter, amely visszaadja a “capacity” tagváltozó értékét.
- **+void addCharacter(Character character):** Ezzel a metódussal vagyunk képesek hozzáadni az adott szoba “characters” listájához a paraméterben megadott “Character”-t (például, ha valaki belép a szobába).
- **+void deleteCharacter(Character character):** Ezzel a metódussal vagyunk képesek törölni az adott szoba “characters” listájából a paraméterben megadott “Character”-t (például, ha valaki kilép a szobából).
- **+void deleteNeighbour(Room room):** Ezzel a metódussal vagyunk képesek kitörölni a paraméterben megadott “Room”-ot az adott szoba “neighbours” listájából.
- **+void addTwoWayDoor(Room room):** Ezzel a metódussal vagyunk képesek hozzáadni a paraméterben megadott “Room”-ot az adott szoba “neighbours” listájához.
- **+List<Item> GetItems():** Visszaadja a szobában tárolt tárgyak listáját.

8.1.19 RoomProperty

- **Felelősség**

Olyan absztrakt osztály, melynek felelőssége az, hogy a segítségével a szobák típusait nyilván tudjuk tartani.

- **Ősosztályok**

Nem rendelkezik ősosztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **Asszociáció**

- **kétirányú kapcsolat a Room osztályból a RoomProperty osztály felé:** A szoba tudja magáról milyen képességet adó típusa/típusai vannak. Egy szobának egy időben egyszerre több típusa is lehet. A működése hasonló mint a "Character" és "CharacterProperty" osztályok kapcsolata.

- **Metódusok**

- **+ void effect():** A RoomProperty leszármazottaiban van megvalósítva, működése során a leszármazottnak megfelelő Room tagváltozó (poisonous, cursed, vagy sticky) értékét "true"-ra állítja a leszármazottakban leírt módon. (absztrakt függvény)
- **+void Setter():** Room tagváltozó beállítására.

8.1.20 SlipStick

- **Felelősség**

Az osztály a Logarléc tárgy viselkedését valósítja meg. Felelőssége, hogy ha a hallgatók eszköztárába kerül a Logarléc nevű tárgy, akkor sikeresen teljesítették a küldetésüket és megnyerték a játékot.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-durability:** a tárgy használatának a számát jelez, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).

- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).

- **Asszociációk**

- **kétirányú kapcsolat a SlipStick osztályból a Game osztály felé:** A Logarléc ismeri a játékot, és a játék tudja mi van a Logarléccel.

- **Metódusok**

- **+void onPickUp():** Ha ezt a tárgyat felveszi egy karakter, akkor lép működésbe ez a metódus. Ha egy hallgató veszi fel ezt a tárgyat, akkor győzelemmel zárul a játék, oly módon hogy meghívódik a játék "endGame()" metódusa "true" paraméterrel ha egy oktató, akkor pedig letételre kerül máshol és nem ér véget a játék.
- **+void destruct():** biztosítja, hogy a Logarléccet ne tudja oktató elégetni. Nem fog a függvény semmit sem csinálni, csak visszatér.

8.1.21 SoulProtection

- **Felelősség**

A karakterek oktató lélek kiszívás elleni védettségének beállítására szolgál.

- **Ősosztályok**

CharacterProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-timelimit:** megadja, hogy az adott státusz mennyi időegységig érvényes(örökölt attribútum)
- **-character:** Az a Character akire a hatást kifejtik(örökölt attribútum)

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok**

- **+void effect():** Ez a metódus meghívja annak a karakternek a "SetSoulProtection(bool p)" metódusát, aki épp elhasználta az "TVSZ" avagy a HolyBeerMug típusú tárgyat, és ez beállítja a karakter "soulProtection" tagváltozóját a paraméterben kapott értékre, amely ez esetben "true" lesz.

8.1.22 Sticky

- **Felelősség**

A ragacsos tulajdonság rendelhető segítségével a szobákhoz.

- **Össztályok**

RoomProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- Nincsen attribútuma.

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok:**

- **+ void effect():** Ez a metódus meghívja annak a szobának a “SetStickyduration(int duration)” metódusát amelyhez tartozik, és így beállítja a “stickyduration” tagváltozót a paraméterben kapott értékre, és így addig, amíg ez az érték nem éri el a 0-át, ragadós lesz a szoba.

8.1.23 Student

- **Felelősség**

Ez az osztály felel a hallgatók viselkedésének a megvalósításáért.

- **Össztályok**

Character

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a hallgató neve (örökölt attribútum).
- **-location:** Azon szoba amelyben a karakter tartózkodik (örökölt attribútum).
- **-inventory:** A karakternél lévő tárgyak tárolódnak itt (örökölt attribútum).
- **-stunned:** A karakter bénított hatását jelző érték (örökölt attribútum).
- **-knockedOut:** A karakter eszméletvesztett hatását jelző érték (örökölt attribútum).
- **-poisonProtection:** A karakter mérge elleni védetségét jelző érték (örökölt attribútum).

- **-soulProtection:** A karakter oktató lélekkiszívása elleni védetségét jelző érték (örökölt attribútum).
- **-characterProperties:** A karakterre érvényes hatások tárolja (örökölt attribútum).

- **Asszociációk**

Nem vesz részt asszociációban.

- **Metódusok**

- **+bool move(Room whereto) :** A hallgató mozgását valósítja meg, abba a szobába, amely paraméterként megvan adva a függvénynek. Amennyiben a paraméterként kapott szoba nem szomszédos a karakter tartózkodási helyével ("getNeighbours()") függvény segítségével) nem történik semmi. A szobába lépés után a régi szobának meghívódik a "deleteCharacter()", míg az új szobának az "addCharacter()" nevű függvénye az éppen mozgó karakterrel mint paraméterrel. Ezután átállítja a hallgató "location"-át az új szobára a "setLocation()" függvénnyel amit az új szobával, mint paraméterrel meghív. Végül meghívódik a karakter "OnEntryClean()" függvénye. (Ez csak a takarítónál csinál valamit)
- **+void pickUp(Item item) :** A függvény segítségével képes egy hallgató felvenni a szobában lévő tárgyat, így a tárgy bekerül az eszköztárába. Az "inventory" listához hozzáadódik a paraméterben megadott tárgy, míg a szobának meghívódik a "deleteItem" nevű metódusa ugyanazzal a paraméterrel. Ez mind csak akkor történik meg ha a hallgató fel tudja venni a szobából a tárgyat. (Tehát a tárgy a szobában van, nem ragacsos a szoba és nincsen tele a hallgató eszköztára.)
- **+void drop(Item item) :** A függvény segítségével képes egy hallgató, az eszköztárából a szobába letenni a kiválasztott tárgyat. Ekkor a hallgató "inventory" listájából törlésre kerül a tárgy és hozzáadódik a hallgató szobájának "items" listájába a szoba "putItem" függvényének segítségével, melynek paramétere megegyezik a "drop" függvény paraméterével.
- **+void discard(Item item) :** A függvény segítségével képes egy hallgató, az eszköztárából elégetni a kiválasztott tárgyat. Ekkor a tárgy törlődik a hallgató "inventory" listájából. A tárgy "owner" és "location" tagváltozók értéke null lesz.
- **+void death():** Ha a hallgatónak a lelkét kiszívják, akkor ez a metódus felel a hallgató játékból való kieséséért.
- **+void useItem(Item item):** Ennek a függvénynek a segítségével képes egy hallgató, az eszköztárában szereplő tárgyak közül egyet használni. Ekkor meghívódik a tárgy "effect()" függvénye ami kifejti a tárgy hatását. Az "effect()" függvény hatásait lásd a tárgyaknál részletezve.
- **+void kill():** Mivel egy hallgató nem képes lelket kiszívni, emiatt ez a metódus nem csinál semmit, csak visszatér.
- **+void stun():** Mivel egy hallgató nem képes megbénulni, emiatt ez a függvény nem csinál semmit, csak visszatér.

8.1.24 Stunned

- **Felelősség**

A karakterek bénult állapotának beállítására szolgál.

- **Ősosztályok**

CharacterProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-timelimit:** megadja, hogy az adott státusz mennyi időegységig érvényes(örökölt attribútum)

- **-character:** Az a Character akire a hatást kifejtik(örökölt attribútum)

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok**

- **+void effect():** Ez a metódus meghívja annak a karakternek a “SetStunned(bool s)” metódusát, és ez beállítja a karakter “stunned” tagváltozóját a paraméterben kapott értékre.

8.1.25 Teacher

- **Felelősség**

Ez az osztály felel az oktatók viselkedésének a megvalósításáért.

- **Ősosztályok**

Character

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** az oktató neve (örökölt attribútum).

- **-location:** Azon szoba amelyben a karakter tartózkodik (örökölt attribútum).

- **-inventory:** A karakternél lévő tárgyak tárolódnak itt (örökölt attribútum).

- **-stunned:** A karakter bénított hatását jelző érték (örökölt attribútum).

- **-knockedOut:** A karakter eszméletvesztett hatását jelző érték (örökölt attribútum).

- **-poisonProtection:** A karakter mérge elleni védetségét jelző érték (örökölt attribútum).

- **-soulProtection:** A karakter oktató lélekkiszívása elleni védetségét jelző érték (örökölt attribútum).

- **-characterProperties:** A karakterre érvényes hatások tárolja (örökölt attribútum).

- **Asszociációk**

Nem vesz részt asszociációban.

- **Metódusok**
- **+bool move(Room whereto) :** A oktató mozgását valósítja meg, abba a szobába, amely paraméterként megvan adva a függvénynek. Amennyiben a paraméterként kapott szoba nem szomszédos a karakter tartózkodási helyével(“getNeighbours()” függvény segítségével) nem történik semmi. A szobába lépés után a régi szobának meghívódik a “deleteCharacter()”, míg az új szobának az “addCharacter()” nevű függvénye az éppen mozgó karakterrel mint paraméterrel. Ezután átállítja a oktató “location”-át az új szobára a “setLocation()” függvénnyel amit az új szobával, mint paraméterrel meghív. Végül meghívódik a karakter “OnEntryClean()” függvénye. (Ez csak a takarítónál csinál valamit)
- **+void pickUp(Item item) :** A függvény segítségével képes egy oktató felvenni a szobában lévő “Item”-ek közül a paraméterben megadottat, így a tárgy bekerül az oktató “inventory” listájába. Az “inventory” listához hozzáadódik a paraméterben megadott tárgy, míg a szobának meghívódik a “deleteItem” nevű metódusa ugyanazzal a paraméterrel. Ez mind csak akkor történik meg ha a hallgató fel tudja venni a szobából a tárgyat. (Tehát a tárgy a szobában van, nem ragacsos a szoba és nincsen tele a hallgató eszköztára.)
- **+void drop(Item item) :** A függvény segítségével képes egy oktató, az “inventory” -ból lerakni egy tárgyat az adott szobába. Ekkor az oktató “inventory” listájából törlésre kerül a tárgy és hozzáadódik a hallgató szobájának “items” listájába a szoba “putItem” függvényének segítségével, melynek paramétere megegyezik a “drop” függvény paraméterével.
- **+void discard(Item item) :** A függvény segítségével képes egy oktató, az “inventory” listájából elégetni a paraméterként megadott “Item”-et. Ekkor a tárgy törlődik a hallgató “inventory” listájából. A tárgy “owner” és “location” tagváltozók értéke null lesz.
- **+void death():** Mivel egy oktató nem tud meghalni, ezért a metódus nem csinál semmit.
- **+void useItem(Item item):** Mivel egy oktató nem tud tárgyat használni, emiatt a metódus meghívja a tárgy “destruct()” függvényét, amellyel az adott tárgy törlésre kerül.
- **+void kill():** Az oktató a vele egy szobában lévő hallgatók lelkét kiszívja, vagyis minden szobában tartozkodó karakternek meghívódik a death() függvénye (ez a függvény csak a hallgatóknál csinál valamit), amelynek hatására a “soulProtection”-nel nem rendelkező hallgatók kiesnek a játékból.
- **+void stun():** Az oktató megbénításáért felelős függvény, amelynek hatására hozzáadódik a “characterProperties” listájához egy “Stunned” “CharacterProperty”, melynek az “effect()” metódusának meghívása után, beállítódik a “stunned” tagváltozó, így bénultá válik az oktató.

8.1.26 Transistor

- **Felelősség**

Az osztály a Tranzisztor tárgy viselkedését valósítja meg. Felelőssége, hogy biztosítja azon hallgatók számára a teleportálást, akiknek van két összekötött tranzisztoruk és ezek közül az

egyik már azon szobába le van helyezve, ahová a hallgató teleportálni szeretne és használja a másikat.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-connected:** ezzel az attribútummal jelezzük, hogy egy tranzisztor össze van-e már kötve egy másikkal.
- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használhatóságának a számát jelez, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).
- **-pair:** összekapcsolt tranzisztor esetén a tranzisztor párja (egy másik tranzisztor)

- **Asszociációk**

- **asszociáció a Transistor osztállyal:** összekapcsolt tranzisztorok esetén a tranzisztor párja egy másik tranzisztor. Ezt a kapcsolatot fejezi ki ez az asszociáció.

- **Metódusok**

- **+void connect(Item item):** ezzel a függvénnyel kapcsoljuk össze a hallgató az "inventory"-ában lévő tranzisztorokat, úgy hogy beállítja a "pair" tagváltozókat egymásra.
- **+void OnPickup():** Ez a metódus felel azért, hogy ha egy hallgatónál már van egy Transistor, akkor egy másik felvételénél a párosodás teljesüljön. A tárgy felvételekor meghívódik ez a függvény, amennyiben a "connected" változó értéke false akkor meghívja a "connect" függvényt. Amennyiben a "connected" változó értéke true, akkor a hallgató visszateszi a szobába a tranzisztort. Amennyiben egy tanár próbálná meg felvenni a tárgyat és a "connected" változó értéke "true" akkor az oktató is visszateszi a szobába a tranzisztort.
- **+void effect() :** Ezzel a metódussal a hallgató képes lesz teleportálni. A függvény meghívása után a hallgató helyzete az elsőnek letett tranzisztor helyével fog megegyezni.

8.1.27 TVSZ

- **Felelősség**

Az osztály a “TVSZ denevérbőrre nyomtatott példánya” tárgy viselkedését valósítja meg. Felelőssége, hogy ha egy hallgató használja a “TVSZ denevérbőrre nyomtatott példánya” nevű tárgyat, akkor az megvédje őt 3-szor attól, hogy egy oktató kiszívja a lelkét.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatának számát jelzi, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**

Nem vesz részt asszociációban.

- **Metódusok**

- **+void effect() :** Ez a metódus meghívja a tárgyat használó hallgató “SetSoulProtection(bool p)” metódusát, amellyel beállítjuk a hallgató “soulProtection” tagváltozóját a paraméterben megadottnak, amely ez esetben “true” lesz, és így elérjük, hogy a hallgató védetté váljon az oktató támadásai ellen.

8.1.28 WetTowel

- **Felelősség**

Az osztály a “Nedves táblatörő rongy” tárgy viselkedését valósítja meg. Felelőssége, hogy a hallgató a használata során képes legyen egy oktatót megbénítani, és ez által kisebb előnyre tegyen szert a játék során.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatának a számát jelez, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**

Nem vesz részt asszociációban.

- **Metódusok**

- **+void effect():** Ez a metódus meghívja minden a szobában lévő karakter “stun()” metódusát. Amennyiben a karakter típusa “Teacher”, úgy létrejön egy “Stunned” típusú objektum, amit paraméterül kap az addProperty függvény, végül meghívódik az így létrejött “Stunned” osztály “effect()” függvénye.

8.1.29. Controller

- **Felelősség**

A Controller osztály kezeli a bemeneteket, egyszerű hibakezelést hajt végre, és a bemenettől függően meghívja azokat a függvényeket, melyek szükségesek. Az osztály <funkció neve>Handler függvényeket tartalmaz, amelyekben a konzolos működés van megvalósítva.

- **Ősosztályok**

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

A tesztek részletes tervei, leírásuk a teszt nyelvén

8.1.1 Delete Door

- **Leírás**

Két szoba között törlődik az ajtó, amelyen keresztül korábban a két szoba elérhető volt egymásból.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Az ajtó valóban törlődik-e
- Hiba lehet, hogy az ajtó törlése után egy karakter mégis képes átlépni közvetlen az egyik szobából a másikba.

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 unconnect 0 1 start game move 1	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a 0.room is not longer a neighbour of 1.room a 1.room is not longer a neighbour of 0.room game started it's Jomagam's turn! player Jomagam tried to move to 1.room, but could not do it

8.1.2 Add Door

- **Leírás**

Két szoba között nincs ajtó, így hiába próbál a játékos átmenni a másik szobába, nem sikerül neki. Ezt követően hozzáadásra kerül a két szoba közé egy ajtó, így a következő próbálkozásnál már végre tudja hajtani a műveletet.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Két olyan szoba között, melyek között nincs ajtó, nem tud átlépni közvetlenül a karakter
- Sikeres-e az ajtó hozzáadása

• Bemenet	• Elvárt kimenet
add room -1 add room 0	the first room has been added to the labyrinth a room was added as a neighbour of 0.room

add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 disconnect 0 1 start game move 1 connect 0 1 1 move 1	a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a 0.room is not longer a neighbour of 1.room a 1.room is not longer a neighbour of 0.room game started it's Jomagam's turn! player Jomagam tried to move to 1.room, but could not do it a 0.room became neighbour of 1.room a 1.room became neighbour of 0.room it's Jomagam's turn! player Jomagam moved to 1.room
--	--

8.2.3 Split Rooms

- Leírás**

Egy szoba kettéosztódik úgy, hogy a belőle keletkező két szoba megörökli az eredeti szoba tulajdonságait, emellett a tárgyak és a karakterek egyenlő arányban osztódnak szét a két szoba között.

- Ellenőrzött funkcionalitás, várható hibahelyek**

- A keletkező szobák megöröklik-e a tulajdonságokat
- Egyenlő arányban történik-e meg a szétosztás

<ul style="list-style-type: none"> Bemenet controller toggle random add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 add character student En 0 add item tvsz 0 add item ffp2 0 controller poisonous 0 controller cursed 0 start game controller split 0 print room 0 print room 1	<ul style="list-style-type: none"> Elvárt kimenet the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a student was added to 0.room a tvsz was added to 0.room a ffp2 was added to 0.room 0.room was poisoned by controller. 0.room was cursed by controller. game started it's Jomagam's turn!
--	---

	0.room was split by controller. details of room 0: cursed: true poisonous: true slippery: false cleaned: false capacity: 6 stickylimit:3 neighbours: 1 2 7 8 details of room 1: cursed: true poisonous: true slippery: false cleaned: false capacity: 6 stickylimit: 3 neighbours: 0 2 7 8
--	--

8.2.4 Merge Rooms

• Leírás

Két szoba összeolvad úgy, hogy az új szoba megörökli az eredeti szobák tulajdonságait, a bennük lévő tárgyak és karakterek az új szobába kerülnek. A megöröklés módja alatt azt értjük, hogy amennyiben valamelyik szoba rendelkezik egy tulajdonsággal (pl. gázos), akkor az új szoba is rendelkezni fog azzal.

• Ellenőrzött funkcionalitás, várható hibahelyek

- Az új szoba valóban megörökli-e a tulajdonságokat
- A karakterek valóban átkerülnek-e az új szobába
- A tárgyak valóban átkerülnek-e az új szobába
- Az új szobából elérhetőek-e azok a szobák, amelyek az előző szobákból is elérhetőek voltak

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 add character student En 1 add item tvsz 0 add item ffp2 1 controller poisonous 0 controller slippery 0 controller cursed 1 start game controller merge 0 1 print room 0	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a student was added to 1.room a tvsz was added to 0.room a ffp2 was added to 1.room 0.room was poisoned by controller. 0.room was made slippery by controller 1.room was cursed by controller. It's Jomagam's turn! 0. room was merge with 1 by controller details of room 0: cursed: true poisonous: true slippery: true cleaned: false capacity: 12

	stickylimit: 10 neighbours: 1, 5, 6
--	--

8.2.5 Room Gets Poisonous

- **Leírás**

A szoba gázossá válik, mivel egy benne tartózkodó hallgató felhasznál egy CabbageCamambert tárgyat.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Valóban gázossá válik-e a szoba.

Bemenet	Elvárt kimenet
<ul style="list-style-type: none"> controller toggle random add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 put item Jomagam cabbagecamambert start game use 0 print room 0 	<ul style="list-style-type: none"> the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a CabbageCamambert was added to Jomagam's inventory It's Jomagam's turn! player Jomagam made 0.room poisonous details of room 0: <ul style="list-style-type: none"> cursed: false poisonous: true slippery: false cleaned: false capacity: 6 stickylimit: 3 neighbours: 1, 6, 7

8.2.6 Room Gets Slippery

- **Leírás**

A szoba koszosná válik, mivel a szobába belépett hallgatók száma meghaladja a limitet, ami a tesztelésben 2. Ennek következtében a szobában lévő tárgyat nem fogják tudni a hallgatók felvenni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Valóban koszosná válik-e a szoba.

- Valóban nem tudják majd felvenni a szobában elhelyezett tárgyat a hallgatók, amennyiben a szoba koszos.

Bemenet	Elvárt kimenet
<ul style="list-style-type: none"> controller toggle random add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 print room 0 add character student Jomagam 1 add character student En 6 add character student ST 7 add item ffp2 0 start game move 0 move 0 move 0 pickup 0 	<ul style="list-style-type: none"> the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit: 2 neighbours: 1, 6, 7 a student was added to 1.room a student was added to 6.room a student was added to 7.room game started it's Jomagam's turn! player Jomagam moved to 0.room player En moved to 0.room player ST moved to 0.room limit exceeded - room 0 became slippery it's Jomagam's turn! Jomagam could not pick up item, because room is slippery

8.2.7 Student enters a Room, which is Poisonous and has not got any Item, which can protect him

Leírás

Egy hallgató belép egy szobába, amelyben mérges gáz van, de nincs nála olyan tárgy, amellyel védelmet tudna magának szerezni. Ennek következtében a hallgató elájul.

Ellenőrzött funkcionalitás, várható hibahelyek

- Valóban elájul-e a hallgató.

Bemenet	Elvárt kimenet
<ul style="list-style-type: none"> add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 controller poisonous 0 print room 0 add character student Jomagam 1 	<ul style="list-style-type: none"> the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room

start game move 0	a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room 0. room was poisoned by controller. details of room 0: cursed: false poisonous: true slippery: false cleaned: false capacity: 6 stickylimit: 2 neighbours: 1, 6, 7 a student was added to 1.room game started it's Jomagam's turn! player Jomagam moved to 0.room Jomagam knocked out
----------------------	---

8.2.8 Student enters a Room, which is Poisonous and an FFP2Mask, which can protect him

- Leírás**

Egy hallgató belép egy szobába, amelyben mérges gáz van, de van nála egy FFP2Mask tárgyat, amelyet elhasználva védelmet tud magának szerezni. Ennek következtében a hallgató nem ájul el.

- Ellenőrzött funkcionalitás, várható hibahelyek**

- Valóban képes-e a hallgatót az item megvédeni.
- Csökken-e az FFP2Mask tárgynak a felhasználhatósági ideje.

<ul style="list-style-type: none"> Bemenet <p>add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 controller poisonous 0 print room 0 add character student Jomagam 1 put item Jomagam ffp2 start game use 0 move 0</p>	<ul style="list-style-type: none"> Elvárt kimenet <p>the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room 0. room was poisoned by controller. details of room 0: cursed: false poisonous: true slippery: false cleaned: false capacity: 6 stickylimit: 2 neighbours: 1, 6, 7 a student was added to 1.room a FFP2Mask was added to Jomagam's inventory game started it's Jomagam's turn! player Jomagam get protected from poison</p>
--	--

	it's Jomagam's turn! player Jomagam moved to 0.room
--	--

8.2.9 Relationship between Students and Teachers

• Leírás

Ebben a tesztben a játék legfontosabb tényezőit mutatjuk be, még pedig azokat, hogy a hallgatók hogyan mozognak egyik szobából a másikba, hogyan tudják megvédeni magukat a tanárok támadásától (vagy hogyan ér véget számukra a játék), továbbá hogyan tudnak visszatámadni. Végül, de nem utolsó sorban tesztelésre kerül a játék vége metódus is és a gázos szoba kitakarítása is.

• Ellenőrzött funkcionalitás, várható hibahelyek

- Képesek-e a hallgatók szobát váltani
- Az általuk felhasznált tárgyak megvédik-e őket a tanárok támadásától
- Ha nem tudnak védekezni, valóban véget ér-e a számukra a játék.

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 controller poisonous 0 add item slipstick 1 add character student elso 2 add character student masodik 3 put item masodik holybeermug put item masodik airspray add character student harmadik 4 put item harmadik tvsz put item harmadik wettowel add character student negyedek 6 put item otodik tvsz add character teacher tanar 7 start game skip turn use item 0 use item 0 use item 0 skip turn move 7 move 7 move 7 kill move 0 use item 1 move 1 skip turn use item 1 skip turn pick up 0	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room 0.room was poisoned by controller a slipstick was added to 1.room a student was added to 2.room a student was added to 3.room a holybeermug was added to masodik's inventory a airspray was added to masodik's inventory a student was added to 4.room a tvsz was added to harmadik's inventory a wettowel was added to harmadik's inventory a student was added to 6.room a tvsz was added to negyedek's inventory a teacher was added to 7.room game started it's elso's turn! elso has skipped his turn player masodik get protected from teachers player harmadik get protected from teachers player negyedek get protected from teachers teacher has skipped his turn it's elso's turn! player elso moved to 7.room player second moved to 7.room player harmadik moved to 7.room player negyedek moved to 7.room teacher tried to kill the students in the room it's second's turn! player second moved to 0.room player harmadik stunned teachers in 7.room player negyedek moved to 1.room

	teacher is stunned it's second's turn! player used air spray in 0.room harmadik has skipped his turn player negyedik picked up Slipstick from 1.room Students won the game
--	---

8.2.10 Room is full, student cannot get in

- **Leírás**

Ebben a tesztben megvizsgálásra kerül, hogy amennyiben egy szoba tele van, tehát nem képes a kapacitásán felül több személyt befogadni, akkor egy újabb játékos ne tudjon belépni a szobába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Valóban be tudnak-e lépni a karakterek a szobába, amíg az nincs tele.
- Valóban nem tud-e belépni a karakter a szobába, amíg az tele van.

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 print room 0 add character student elso 6 add character student masodik 1 add character student harmadik 7 add character student negyedik 7 start game move 0 move 0 move 0 move 0	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 3 stickylimit:2 neighbours: 1,6,7 items: a student was added to 6.room a student was added to 1.room a student was added to 7.room a student was added to 7.room game started it's elso's turn! player elso moved to 0.room player masodik moved to 0.room player harmadik moved to 0.room player negyedik tried to move to 0.room, but could not do it

8.2.11 Transistors in action

- **Leírás**

Felépítjük a labirintust. Lehelyezünk benne egy tanulót, majd leteszünk egy transzistort a szobában és egyet a tanuló tárolójában. Ezt követően elkezdődik a játék a tanuló felveszi a szobában levő transzistort, ezáltal az kapcsolódik a már eredetileg nála levővel. Ezt követően lehelyezi a jelenlegi szobájában, majd átlép a vele szomszédos 1-es szobába és felhasználja a nála levő transzistort ezzel visszateleportálva a 0. szobába, ahol letette a párját. (Közben kiírások segítik az ellenőrzést, hogy megtörtént-e a kapcsolódás és a teleportálás. Kiírjuk a tanuló jelenlegi szobáját a tranzisztor felhasználása (teleportálás) előtt és után.)

• **Ellenőrzött funkcionalitás, várható hibahelyek**

A két tranzisztornak automatikusan kapcsolódnia kell, miután a 2. darab is bekerült a tárolóba.

A tanulónak sikeres teleportálódnia kell abba szobába, ahova a tranzisztor párja le lett helyezve.

Teleportálás után a tranzisztorok megsemmisülnek.

• Bemenet	• Elvárt kimenet
<pre> add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 add item transistor 0 put item Jomagam transistor start game print inventory Jomagam pickup 0 print inventory Jomagam drop 1 move 1 print currRoom Jomagam use 0 print currRoom Jomagam </pre>	<pre> the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a transistor was added to 0.room a transistor was added to Jomagam's inventory game started it's Jomagam's turn! Jomagam's inventory: details of item 0 name: transistor durability: 5 currentlyUsed: false connected: false player Jomagam picked up 0.item (transistor) from 0.room Jomagam's inventory: details of item 0 name: transistor durability: 5 currentlyUsed: false connected: true details of item 1 name: transistor durability: 5 currentlyUsed: false connected: true it's Jomagam's turn! player Jomagam dropped 1.item (transistor) from inventory to 0.room it's Jomagam's turn! player moved to 1.room details of room 1: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit: 4 neighbours: 0,7,2 items: </pre>

	it's Jomagam's turn! player Jomagam teleported from 1.room to 0.room details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:4 neighbours: 1,6,7 items:
--	--

8.2.12 Cleaner enter room

• Leírás

Felépítjük a labirintust, melynek 0. és 1. szobájában elhelyezünk egy-egy tanulót, egyiküknek a tárolójába elhelyezünk egy ffp2 maskot, másikéba pedig nem, így ő nem tud védett lenni a gáztól. A 7. szobát gázossá tesszük. Elhelyezünk egy takarítót is a labirintus 2. szobájában. Ezt követően elkezdődik a játék. A tanuló használja a maszkot, mielőtt a gázos szobába lépne, emiatt védetté válik (így nem számít kábultnak, ezért ki lehet majd tesséklni), a másik tanuló belép a 7. szobába és elájul. Ezt követően a takarító jön, aki skippel, majd belép a 7. szobába, kitessekeli a tanulót és kitisztítja a szobát. (Az 7. szoba adatait kiirattjuk a takarítás előtt és után is, így látjuk, hogyan változott.)

• Ellenőrzött funkcionalitás, várható hibahelyek

- A takarító sikeresen kitessekeli-e a nem kábult tanulót.
- A takarító nem tessékeli ki a kábult tanulót.
- A szoba kitakarítottá válik és már nem mérgező többé.

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 put item Jomagam ffp2mask add character student Ajult 1 controller poisonous 7 add character cleaner Takarito 2 start game print room 7 use 0 move 7 skip turn move 7 skip turn print room 7	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a ffp2 was added to Jomagam's inventory a student was added to 1.room 7.room was poisoned by controller a cleaner was added to 2.room game started it's Jomagam's turn! details of room 7: cursed: false poisonous: true slippery: false cleaned: false

	capacity: 5 stickylimit: 4 neighbours: 0 1 2 3 4 5 6 items: player Jomagam get protected from poison player Ajult moved to 7.room Ajult knocked out Takarito has skipped his turn it's Jomagam's turn! player Jomagam moved to 7.room Ajult skipped his turn Cleaner moved to 7.room details of room 7: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit: 4 neighbours: 0 1 2 3 4 5 6 items:
--	---

8.2.13 Pick up SlipstickTeacher

- **Leírás**

Felépítjük a labirintust és a 0. szobájában elhelyezünk egy tanárt. A szobába, ahol tartózkodik leteszünk egy logarlécet. A játék elkezdődik és amikor a tanár következik megpróbálja felvenni a logarlécet a szobából a tárolójába. (Közben kiíratjuk a tanár tárolójának adatait és a szoba adatait is, a felvétel előtt és után és látjuk, hogy a szobából eltűnt a tárgy, de a tárolóban megjelent.)

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Sikeresen megjelenik a tárgy a karakter tárolójában.
- Eltűntik a tárgy a szobából.
- A logarléc felvétele egy tanár által nem okozhatj a játék végét/megnyerését.

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character teacher Tanar 0 add item slipstick 0 start game print room 0 print inventory pickup 0 print room 0 print inventory 0	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a teacher was added to 0.room a slipstick was added to 0.room game started it's Tanar's turn! details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5

	neighbours: 1,6,7 items: details of item 0 name: slipstick durability: 5 currentlyUsed:false Tanar's inventory: player Tanar picked up 0.item(slipstick) from 0.room details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: Tanar's inventory: details of item 0 name: slipstick durability: 5 currentlyUsed:false
--	---

8.2.14 Pick up stg Student

Felépítjük a labirintust és a 0. szobájában elhelyezünk egy tanulór. A szobába, ahol tartózkodik leteszünk egy légrfrissítőt. A játék elkezdődik és amikor a tanuló következik megpróbálja felvenni a légrfrissítőt a szobából a tárolójába. (Közben kiiratjuk a tanuló tárolójának adatait és a szoba adatait is, a felvétel előtt és után és látjuk, hogy a szobából eltűnt a tárgy, de a tárolóban megjelent.)

- **Ellenőrzött funkcionalitás, várható hibahelyek**
- Sikeresen megjelenik a tárgy a karakter tárolójában.
- Eltűntik a tárgy a szobából.

Bemenet	Elvárt kimenet
• add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 add item airspray 0 start game print room 0 print inventory pickup 0 print room 0 print inventory 0	• the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a airspray was added to 0.room game started it's Jomagam's turn! details of room <0>: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: details of item <0>

	name: aerosol durability: 5 currentlyUsed:false Jomagam's inventory: player Jomagam picked up 0.item(aerosol) from 0.room details of room <0>: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: Jomagam's inventory: details of item <0> name: aerosol durability: 5 currentlyUsed:false
--	--

8.2.15 Student tries to pick up an item, but inventory is full

Ennek a tesztnek a célja, hogy megvizsgáljuk azt, hogy amennyiben egy hallgató eszköztára tele van (5 tárgy van nála), akkor hiába próbál meg felvenni egy tárgyat a szobában, nem fog neki sikerülni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**
- Ha tele van az eszköztára, fel tud-e venni további tárgyat.

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student elso 0 put item elso tvsz put item elso transistor put item elso ffp2 put item elso aerosol put item elso fakeslipstick add item holybeermug 0 start game pickup 0	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a tvsz was added to elso's inventory a transistor was added to elso's inventory a ffp2 was added to elso's inventory a aerosol was added to elso's inventory a fakeslipstick was added to elso's inventory a holybeermug was added to 0.room game started it's elso's turn! elso tried to pickup item, but his inventory is full

8.2.16 Pick up stg Teacher

Felépítjük a labirintust és a 0. szobájában elhelyezünk egy tanár. A szobába, ahol tartózkodik letesszünk egy légfrissítőt. A játék elkezdődik és amikor a tanár következik megpróbálja felvenni a légfrissítőt a szobából a tárolójába. (Közben kiiratjuk a tanuló

tárolójának adatait és a szoba adatait is, a felvétel előtt és után és látjuk, hogy a szobából eltűnt a tárgy, de a tárolóban megjelent.)

- **Ellenőrzött funkcionalitás, várható hibahelyek**
- Sikeresen megjelenik a tárgy a karakter tárolójában.
- Eltűntik a tárgy a szobából.

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character teacher Tanar 0 add item airspray 0 start game print room 0 print inventory Tanar pickup 0 print room 0 print inventory Tanar	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a teacher was added to 0.room a airspray was added to 0.room game started it's Tanar's turn! details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: details of item 0 name: airspray durability: 5 currentlyUsed:false Tanar's inventory: player Tanar picked up 0.item(airspray) from 0.room details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: Tanar's inventory: details of item 0 name: airspray durability: 5 currentlyUsed:false

8.2.17 Pick up Slipstick Student - winning alg test

- **Leírás**
 Felépítjük a labirintust és a 0. szobájában elhelyezünk egy tanulót. A szobába, ahol tartózkodik leteszünk egy logarlécet. A játék elkezdődik és amikor a tanuló következik megpróbálja felvenni a logarlécet a szobából, ezzel megnyerve a játékot.

- **Ellenőrzött funkcionalitás, várható hibahelyek**
- A logarléc felvétele automatikusan a játék megnyerését, ezzel a végét eredményezte..

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 add item slipstick 0 start game pickup 0	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a slipstick was added to 0.room game started it's Jomagam's turn! Students won the game

8.2.18 Drop Slipstick Teacher

- **Leírás**

Felépítjük a labirintust, a 0. szobában elhelyezünk egy tanárt, akinek a tárolójába felvesszük a logarléceket. Ezt követően elkezdődik a játék. A tanár megpróbálja lehelyezni a logarléceket a tárolójából a szobába, ahol tartózkodik. (Közben kiírtajuk a szobát és a tanár tárolójának tartalmát a tárgy letevése előtt. Azt kell látnunk, hogy a tárgy megjelent a szobában és eltűnt a tárolójából.

- **Ellenőrzött funkcionalitás, várható hibahelyek**
- Sikeresen meg kell jelennie a tárgynak a szobában ahol tartózkodik.
- El kell tűnnie a tárgynak a tárolójából

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character teacher Tanar 0 put item Tanar slipstick start game print room 0 print inventory Tanar	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room

drop 0 print room 0 print inventory Tanar	a 7.room became neighbour of 4.room a student was added to 0.room a airspray was added to 0.room game started it's Tanar's turn! details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: Tanar's inventory: details of item <0> name: slipstick durability: 5 currentlyUsed:false player Tanar dropped 0.item(slipstick) from inventory to 0.room details of room <0>: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: details of item <0> name: slipstick durability: 5 currentlyUsed:false Tanar's inventory:
---	--

8.2.19 Drop Item Student

- Leírás**

Felépítjük a labirintust, a 0. szobában elhelyezünk egy tanulót, akinek a tárolójába felvesszük egy léghfrissítőt. Ezt követően elkezdődik a játék. A tanuló megpróbálja lehelyezni a léghfrissítőt a tárolójából a szobába, ahol tartózkodik. (Közben kiírtajuk a szobát és a tanuló tárolójának tartalmát a tárgy letevése előtt. Azt kell látnunk, hogy a tárgy megjelent a szobában és eltűnt a tárolóból.

- Ellenőrzött funkcionalitás, várható hibahelyek**

- Sikeresen meg kell jelennie a tárgynak a szobában ahol tartózkodik.
- El kell tűnnie a tárgynak a tárolójából

<ul style="list-style-type: none"> Bemenet add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 put item Jomagam airspray start game print room 0 print inventory Jomagam drop 0 print room 0	<ul style="list-style-type: none"> Elvart kimenet the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room
---	--

print inventory Jomagam	a aerospray was added to Jomagam's inventory game started it's Jomagam's turn! details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: Jomagam's inventory: details of item 0 name: aerospray durability: 5 currentlyUsed:false player Jomagam dropped 0.item(aerospray) from inventory to 0.room details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: details of item 0 name: aerospray durability: 5 currentlyUsed:false Jomagam's inventory:
-------------------------	--

8.2.20 Discard Item Teacher

- Leírás**

- Felépítjük a labirintust, a 0. szobában elhelyezünk egy tanárt, akinek a tárolójába felvesszük egy légfrissítőt. Ezt követően elkezdődik a játék. A tanár megpróbálja elégetni a légfrissítőt a tárolójából. (Közben kiírtajuk a szobát és a tanár tárolójának tartalmát a tárgy letevése előtt. Azt kell látnunk, hogy a tárgy eltűnt a tárolóból és a szobában sem jelent meg.

- Ellenőrzött funkcionalitás, várható hibahelyek**

- Nem szabad, hogy a tárgy égetés után megjelenjen a szobában.
- El kell tűnnie a tárgynak a tárolójából a karakternek.

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character teacher Tanar 0 put item Tanar aerospray start game print room 0 print inventory Tanar discard 0 print room 0 print inventory Tanar	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a teacher was added to 0.room a aerospray was added to 0.room game started it's Tanar's turn! details of room 0:

	cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: Tanar's inventory: details of item 0 name: aerospray durability: 5 currentlyUsed:false player Tanar discarded 0.item(aerospray) from inventory details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: Tanar's inventory:
--	---

8.2.21 Discard Item Student

- Leírás

Felépítjük a labirintust, a 0. szobában elhelyezünk egy tanulót, akinek a tárolójába felvesszük egy légfrissítőt. Ezt követően elkezdődik a játék. A tanuló megpróbálja elégetni a légfrissítőt a tárolójából. (Közben kiírtajuk a szobát és a tanuló tárolójának tartalmát a tárgy letevése előtt. Azt kell látnunk, hogy a tárgy eltűnt a tárolóból és a szobában sem jelent meg.

- Ellenőrzött funkcionalitás, várható hibahelyek

- Nem szabad, hogy a tárgy égetés után megjelenjen a szobában.
- El kell tűnnie a tárgynak a tárolójából a karakternek.

<ul style="list-style-type: none"> • Bemenet add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 put item Jomagam aerospray start game print room 0 print inventory Jomagam discard 0 print room 0 print inventory Jomagam 	<ul style="list-style-type: none"> • Elvárt kimenet the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a aerospray was added to 0.room game started it's Jomagam's turn! details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items:
---	---

	Jomagam's inventory: details of item 0 name: aerospray durability: 5 currentlyUsed:false player Jomagam discarded 0.item(aerospray) from inventory details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: Jomagam's inventory:
--	--

8.2.22 Use CabbageCamambert Student

• Leírás

Felépítjük a labirintust és helyezzünk két tanulót a 0.szobában. Az elsőnek helyezett tanuló tárolójába helyezzünk egy káposztás camambert. Ezt követően elkezdődik a játék és tanuló felhasználja a káposztást camambert. Ez a szobát, ahol tartózkodik gázossá teszi, de egyik karakter sem belül le, aki alpból a szobában tartózkodott (Közben a használat előtt és után kiiradjuk az elsőnek helyezett tanuló tárolóját és a szobát, ahol tartózkodik. Azt kell látnunk, hogy a tárgy eltűnt a tárolójából, nem jelent meg a szobában és kifejtette a hatását, tehát a szobát mérgezővé tette.)

• Ellenőrzött funkcionalitás, várható hibahelyek

- Sikeresen eltűnt a tárgy a tanuló tárolójából.
- A tárgy nem jelent meg a szobában.
- A szoba, ahol tartózkodott gázossá vált.
- A gáz hatására egyik karakter sem ájult el, aki már alpból a szobában tartózkodott.

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 add character student Jomagam2 0 put item Jomagam cabbagecamambert start game print room 0 print inventory Jomagam use 0 print room 0 print inventory Jomagam	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a student was added to 0.room game started

	it's Jomagam's turn! details of room 0: cursed: false poisonous: false slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items: Jomagam's inventory: details of item 0 name: cabbagemambert durability: 5 currentlyUsed:false player Jomagam made 0.room poisonous player Jomagam get protected from poison player Jomagam2 knocked out Jomagam's inventory details of room 0: cursed: false poisonous: true slippery: false cleaned: false capacity: 5 stickylimit:5 neighbours: 1,6,7 items:
--	--

8.2.23 Use FFP2 Student

- **Leírás**

Felépítjük a labirintust, elhelyezünk benne egy tanulót, akinek a tárolójába egy ffp2-es maszkot teszünk. És mérges gázossá tesszük az 1. szobát. Ezt követően elkezdődik a játék és a tanuló felhasználja a maszkot, majd belép egy olyan szobába, ami gázos. (A használat előtt és után is kiiratjuk a tárolóját a tanulónak, azt kell látnunk, hogy a maszk nem tűnt el, de látszik, hogy használatba került.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Gáztól védetté vált-e a karakter, tehát nem ájul el a gázos szobában
- A tárgynak nem szabad még eltűnnie a tárolójából, csak használatba kerülni

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 put item Jomagam ffp2 controller poisonous 1 start game print inventory Jomagam use 0 move 1 print inventory Jomagam	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a ffp2 was added to Jomagam's inventory 1.room was poisoned by controller game started it's Jomagam's turn! Jomagam's inventory:

	details of item 0 name: ffp2 durability: 5 currentlyUsed:false player Jomagam get protected from poison it's Jomagam's turn! player Jomagam moved to 1.room Jomagam's inventory: details of item 0 name: ffp2 durability: 5 currentlyUsed:true
--	---

8.2.24 Use TVSZ Student

• Leírás

Felépítjük a labirintust, elhelyezünk benne egy tanulót, akinek a tárolójába egy TVSZ-t teszünk. És az 1. szobában elhelyezünk egy tanárt. Ezt követően elkezdődik a játék és a tanár belép abba a szobába, ahol a tanuló van és megpróbálja megölni. (A használat előtt és után is kiírjuk a tárolóját a tanulónak, azt kell látnunk, hogy a tvsz nem tűnt el, de látszik, hogy használatba került.

• Ellenőrzött funkcionalitás, várható hibahelyek

- Oktatóktól védetté vált-e a karakter vagyis, ha az oktató megpróbálja megölni nem sikerül neki
- Nem szabad eltűnnie a tárolójából

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 add character teacher Tanar 1 put item Jomagam tvsz start game print inventory Jomagam use 0 print inventory Jomagam move 0 skip turn kill	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a teacher was added to 0.room a tvsz was added to Jomagam's inventory game started it's Jomagam's turn! Jomagam's inventory: details of item 0 name: tvsz durability: 5 currentlyUsed:false player Jomagam get protected from teachers Jomagam's inventory: details of item 0 name: tvsz durability: 5 currentlyUsed:true player Tanar moved to 0.room it's Jomagam's turn! Jomagam has skipped his turn Tanar tried to kill the students

8.2.25 Use HolyBeerMug Student

- **Leírás**

Felépítjük a labirintust, elhelyezünk benne egy tanulót, akinek a tárolójába egy szent söröskorsót teszünk. És az 1. szobában elhelyezünk egy tanárt. Ezt követően elkezdődik a játék és a tanár belép abba a szobába, ahol a tanuló van és megpróbálja megölni. (A használat előtt és után is kiirattjuk a tárolóját a tanulónak, azt kell látnunk, hogy a szent sörös korsó nem tűnt el, de látszik, hogy használatba került.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Oktatóktól védetté vált-e a karakter vagyis, ha az oktató megpróbálja megölni nem sikerül neki
- Nem szabad eltűnnie a tárolójából

Bemenet	Elvárt kimenet
<ul style="list-style-type: none"> add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 add character teacher Tanar 1 put item Jomagam holybeermug put item Jomagam ffp2 start game print inventory Jomagam use 0 print inventory Jomagam move 0 skip turn kill 	<ul style="list-style-type: none"> the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a teacher was added to 0.room a holybeermug was added to Jomagam's inventory a ffp2 was added to Jomagam's inventory game started it's Jomagam's turn! Jomagam's inventory: details of item 0 name: holybeermug durability: 5 currentlyUsed:false details of item 1 name: ffp2 durability: 3 currentlyUsed:false player Jomagam get protected from teachers Jomagam's inventory: player Tanar moved to 0.room it's Jomagam's turn! Jomagam has skipped his turn teacher tried to kill the students in the room

8.2.26 Use AirSpray Student

- **Leírás**

Felépítjük a labirintust, elhelyezünk benne egy tanulót, akinek a tárolójába egy légfrissítőt valamint egy ffp2-es maszkot teszünk. Az 1. szobát gázossá tesszük, majd elkezdődik a játék. a tanuló először felhasználja az ffp2-es maszkot ezáltal védetté válik a gáztól, majd belép a gázos szobába és felhasználja a légfrissítőt. (A használat előtt és után is kiíratjuk a szoba adatait, azt kell látnunk, hogy már nem mérgező, valamint a tanuló tárolóját is kiíratjuk, ahol azt kell látni, hogy a tárgy eltűnt)

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- A tárgy eltűnt a tárolójából
- A szobának már nem szabad mérgezőnek lennie.

Bemenet	Elvárt kimenet
<ul style="list-style-type: none"> add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 put item Jomagam ffp2 put item Jomagam airspray controller poisonous 1 start game print inventory use 0 print room 1 move 1 use 1 print inventory 0 print room 1 	<p>the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a ffp2 was added to Jomagam's inventory a airspray was added to Jomagam's inventory 1.room was poisoned by controller game started it's Jomagam's turn! Jomagam's inventory: details of item 0 name: ffp2 durability: 5 currentlyUsed:false details of item 1 name: airspray durability: 5 currentlyUsed:false player Jomagam get protected from poison details of room 1: cursed: false poisonous: true slippery: false cleaned: false capacity: 6 stickylimit: 3 neighbours: 0 2 7 it's Jomagam's turn! player Jomagam moved to 1.room it's Jomagam's turn! player Jomagam used air spray in 1.room Jomagam's inventory: details of item 0 name: ffp2 durability: 5 currentlyUsed:true details of room 1: cursed: false poisonous: false slippery: false cleaned: false capacity: 6 stickylimit: 3 neighbours: 0 2 7</p>

8.2.27 Use WetTowel Student

- **Leírás**

Felépítjük a labirintust, elhelyezünk benne egy tanulót, akinek a tárolójába egy nedves táblatörölt teszünk. És az 1. szobában lehelyezünk egy tanárt. Ezt követően elkezdődik a játéka tanuló felhasználja a nedves táblatörölt, majd belép a szobába, ahol a tanár tartózkodik, ezzel megbénítva a tanárt. (A használat előtt és után is kiíratjuk a tárolóját a tanulónak, azt kell látnunk, hogy a szent nedves táblatörölt nem tűnt el, de látszik, hogy használatba került.)

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- Megbénította az oktatót, aki a szobában tartózkodott, ahova lépett
- Nem szabad eltűnnie a tárolójából a tárgynak

• Bemenet	• Elvárt kimenet
<pre> add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 add character teacher Tanar 1 put item Jomagam wettowel start game print inventory Jomagam use 0 skip turn print inventory Jomagam move 1 </pre>	<pre> the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a teacher was added to 1.room a wettowel was added to Jomagam's inventory game started it's Jomagam's turn! Jomagam's inventory: details of item 0 name: wettowel durability: 5 currentlyUsed:false player Jomagam get protected from teachers Tanar has skipped his turn it's Jomagam's turn! Jomagam's inventory: details of item 0 name: wettowel durability: 5 currentlyUsed:true it's Tanar's turn! Jomagam move to room 1 player Jomagam stunned teachers in 1.room </pre>

8.2.28 Use fakeFFP2 Student

- **Leírás**

Felépítjük a labirintust, elhelyezünk benne egy tanulót, akinek a tárolójába egy fakeffp2-es maszkot teszünk. A tanuló megpróbálja felhasználni, de nem történik semmi (A használat előtt és után is kiiratjuk a tárolóját a tanulónak, azt kell látnunk, hogy a tárgy eltűnt onnan)

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- A tárgy eltűnt a tárolójából

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 put item Jomagam fakeffp2 start game print inventory Jomagam use 0 print inventory Jomagam	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a fakeffp2 was added to Jomagam's inventory game started it's Jomagam's turn! Jomagam's inventory: details of item 0 name: fakeffp2 durability: 5 currentlyUsed:false it was fake Jomagam's inventory:

8.2.29 Use fakeTVSZ Student

- **Leírás**

Felépítjük a labirintust, elhelyezünk benne egy tanulót, akinek a tárolójába egy faketvszt teszünk. A tanuló megpróbálja felhasználni, de nem történik semmi (A használat előtt és után is kiiratjuk a tárolóját a tanulónak, azt kell látnunk, hogy a tárgy eltűnt onnan)

- **Ellenőrzött funkcionalitás, várható hibahelyek**

- A tárgy eltűnt a tárolójából

• Bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room

add character student Jomagam 0 put item Jomagam faketsz start game print inventory Jomagam use 0 print inventory Jomagam	a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a faketsz was added to Jomagam's inventory game started it's Jomagam's turn! Jomagam's inventory: details of item 0 name: faketsz durability: 5 currentlyUsed:false player Jomagam player Jomagam picked up 0.item (faketsz) from 0.room it was fake Jomagam's inventory:
--	--

8.2.30 Pickup fakeSlipStick Student

• Leírás

Felépítjük a labirintust, elhelyezünk benne egy tanulót, és egy szomszédos szobába egy fakeslipsticket teszünk. A tanuló megpróbálja felvenni, de nem történik semmi (A használat előtt és után is kiiratjuk a tárolóját a tanulónak, azt kell látnunk, hogy a tárgy eltűnt onnan)

• Ellenőrzött funkcionalitás, várható hibahelyek

- A tárgy eltűnt a tárolójából
- A tárolóba kerülése nem jelenti a játék végét/megnyerését

Bemenet	Elvárt kimenet
• add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 5 add room 6 connect 0 6 1 connect 0 7 1 connect 1 7 1 connect 2 7 1 connect 3 7 1 connect 4 7 1 add character student Jomagam 0 add item fakeslipstick 1 start game move 1 pickup 0	• the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 5.room a room was added as a neighbour of 6.room a 0.room became neighbour of 6.room a 6.room became neighbour of 0.room a 0.room became neighbour of 7.room a 7.room became neighbour of 0.room a 1.room became neighbour of 7.room a 7.room became neighbour of 1.room a 2.room became neighbour of 7.room a 7.room became neighbour of 2.room a 3.room became neighbour of 7.room a 7.room became neighbour of 3.room a 4.room became neighbour of 7.room a 7.room became neighbour of 4.room a student was added to 0.room a fakeslipstick was added to 1.room game started it's Jomagam's turn! player Jomagam moved to 1.room it's Jomagam's turn! player Jomagam picked up 0.item (fakeslipstick) from 1.room it was fake

8.2.31 Órai feladat megoldása

• Leírás

Felépítjük a labirintust, elhelyezünk benne egy oktatót és egy tanulót, akinek a tárolójába egy TVSZ-t teszünk. A tanuló elhasználja a tvsz, ami így megvédi az oktató támadása ellen és kilép az oktató szobájából.

- **Ellenőrzött funkcionalitás, várható hibahelyek**
- Tvsz megvédi a hallgatót
- A tárgy eltűnt a tárolójából

• Elvárt bemenet	• Elvárt kimenet
add room -1 add room 0 add room 1 add room 2 add room 3 add room 4 add room 1 add room 6 connect 1 4 1 connect 3 5 1 connect 1 5 1 add character teacher Gajdos 0 add character student En 6 add item slipstick 2 add item ffp2 1 put item En tvsz start game move 1 skip turn pickup 0 use 0 skip turn move 1 kill move 2 skip turn pickup 0	the first room has been added to the labyrinth a room was added as a neighbour of 0.room a room was added as a neighbour of 1.room a room was added as a neighbour of 2.room a room was added as a neighbour of 3.room a room was added as a neighbour of 4.room a room was added as a neighbour of 1.room a room was added as a neighbour of 6.room a 1.room became neighbour of 4.room a 4.room became neighbour of 1.room a 3.room became neighbour of 5.room a 5.room became neighbour of 3.room a 1.room became neighbour of 5.room a 5.room became neighbour of 1.room a teacher was added to 0.room a student was added to 6.room a slipstick was added to 2.room a ffp2mask was added to 1.room a tvsz was added to En's inventory game started it's Gajdos's turn! player Gajdos moved to 1.room En has skipped his turn it's Gajdos's turn! player Gajdos picked up 0.item (ffp2) from 1.room player En get protected from teachers it's Gajdos's turn! Gajdos has skipped his turn player En moved to 1.room it's Gajdos's turn! teacher tried to kill the students in the room player En moved to 2.room it's Gajdos's turn! Gajdos has skipped his turn player En picked up 0.item (slipstick) from 2.room Students won the game

A tesztelést támogató programok tervei

A manuálisan előállított, és ellenőrzött kimeneteket a prototípus által előállított kimenetekkel a Windows parancssor file compare (fc) parancsával fogjuk ellenőrizni. Előre el lesznek készítve az elvárt kimeneteket tartalmazó fájlok, és a program futása után az előállt kimenetekkel össze lehet majd hasonlítani őket a parancsok futtatásával.

A teszteléshez tartozó kimenet fájlban az eredmény ugyanúgy jelenik meg, mint ahogy a tesztelésnél az **“Elvárt kimenet”** alpontban szerepel.

Ezeket a Windows fc parancs segítségével össze tudjuk hasonlítani az általunk előre elkészített elvárt kimenettel. Ha a két kimenet megegyezik a következő kimenetet kell kapjuk:

**Comparing files <teszt_által_előállt_fájl_neve>.txt and <előre_elkészített_fájl_neve>.TXT
FC: no differences encountered**

Amennyiben ez az üzenet akkor a teszt eredménye sikeresnek tekinthető. Bármilyen más esetben a teszt eredménye sikertelen.

Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.04.11 19:00	4 óra	Fazekas Molnár Rimmel	Értekezlet: Új parancsok, tesztek meghatározása, feladatok szétosztása
2024.04.12 12:00	6 óra	Fazekas	Osztályleírások készítése
2024.04.12 12:00	7 óra	Molnár	Osztályleírások készítése
2024.04.13 9:00	3 óra	Rimmel Simon Juhász	Értekezlet: Teszt formátumának megbeszélése, tesztek felosztása, egy mintateszt elkészítése
2024.04.13 15:00	3 óra	Rimmel	Tesztesetek készítése
2024.04.13 17:00	3 óra	Simon	Tesztesetek készítése

2024.04.13 18:00	3 óra	Juhász	Tesztesetek készítése
2024.04.14 17:00	4 óra	Simon	Tesztesetek ellenőrzése
2024.04.14 19:00	3 óra	Juhász	Osztálydiagram ellenőrzése