

7. Prototípus koncepciója

53 – kedvenc_csapatom

Konzulens:

Potyók Csaba

Csapattagok

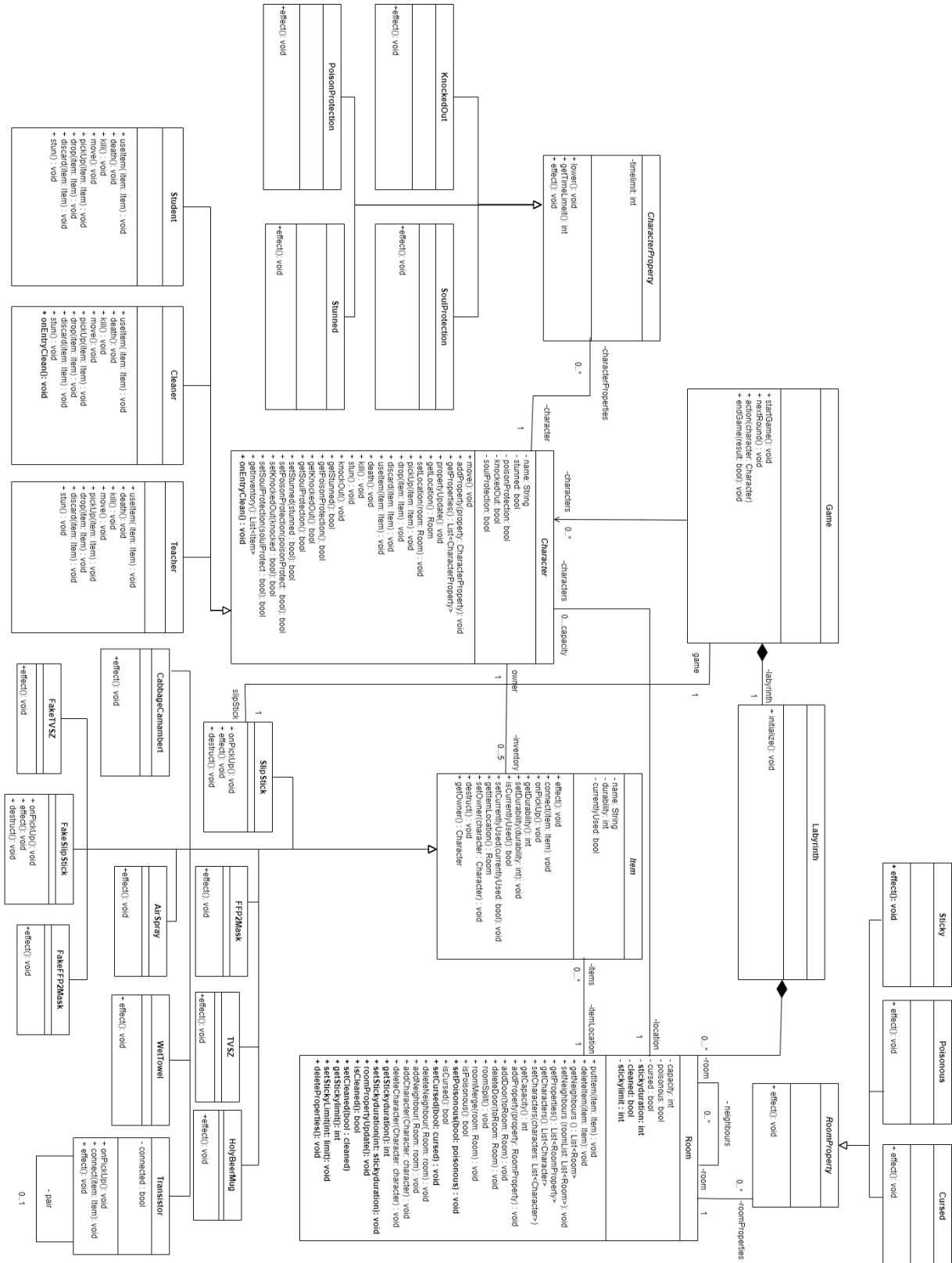
Fazekas Bence Mihály	W0TH54	bencemisi2003@gmail.com
Molnár Botond Kristóf	H1XYPA	molnarboti2003@gmail.com
Simon Tímea	SZFS1V	simontiti1109@gmail.com
Juhász Gábor	VA2469	ju.gabo14@gmail.com
Rimmel Botond	DMPMNZ	rimmelbotond@gmail.com

2024-04-07

7. Prototípus koncepciója

7.0 Változás hatása a modellre

7.0.1 Módosult osztálydiagram



7.0.2 Új vagy megváltozó metódusok

HolyBeerMug osztályban:

- **+void effect():** A metódus a hallgató állapotát beállítja oktató lélek kiszívása ellen védetté (SoulProtection) így az oktatók támadásai nem hatnak a hallgatóra, emellett véletlenszerűen elejt egy tárgyat a használó.

AirSpray osztályban:

- **+void effect():** Ez a metódus felel azért, amikor egy hallgató használja az “AirSpray” nevű tárgyat, hogy a szoba gáztalanná váljon.

RoomProperty osztályban:

- A room osztállyal mostmár két irányú asszociációban van, hogy hasonlóan működjön mint a karakterek állapotainak esetében. Ennek következtében az effect() függvények paramétere eltűnt. Továbbá változtatásokat okozott a Room osztályban és a leszármazottak között. Előbbiben megjelentek új setterek, míg az utóbbiban megjelent egy új leszármazott.

Sticky osztályban:

- A megrendelő általi módosítások miatt lett bevezetve, ennek a segítségével tudjuk megmondani egy szobáról hogy ragacsos-e. A RoomProperty osztályból származik le.
- Egyetlen függvénye van:
 - **+void effect():** A metódus a szobát, “Sticky” típusúvá alakítja. Amennyiben már a szoba “Sticky” volt, akkor nem történik semmi.

Room osztályban:

- A RoomProperty osztályban említett kétirányú asszociáción kívül lett:
 - 3 új tagváltozója:
 - stickyduration: Egy egész szám mely azt adja meg, hogy mennyi időegységig marad ragacsos a szoba.
 - cleaned: Egy flag, amely azt mondja meg hogy az adott szoba volt-e már takarítva, ebből kifolyólag lehet-e ragacsos.
 - stickylimit: A takarítás után az a szám, amely darabszámú ember belépésekor a szoba ragacsossá válik.
 - 10 új függvénye:
 - melyből 6 az új tagváltozókhoz bevezetett getter és setter(ezek névszerint: getStickyduration, setStickyduration, isCleaned, setCleaned setStickyLimit, setStickyLimit)
 - a maradék 4 pedig:
 - roomPropertyUpdate : A szobára érvényben lévő hatások értékeit frissíti.
 - setPoisonous: A szoba mérgező hatását tároló flag értékét állítja.
 - setCursed: A szoba elátkozott hatását tároló flag értékét állítja.
 - deleteProperties: A szoba tulajdonság listáját törlő függvény.

Character osztályban:

- **+onEntryClean():** Ennek a metódusnak, csak a Cleaner osztályban van hatása, különben csak visszatér.

Cleaner osztályban:

- **+onEntryClean():** Ennek a metódusnak 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.
- **+void move() :** A takarító mozgását megvalósító metódus.
- **+void pickUp(Item item) :** Mivel egy takarító nem tud tárgyat felvenni, ezért a metódus nem csinál semmit.
- **+void drop(Item item) :** Mivel egy takarító nem tud tárgyat eldobni, ezért a metódus nem csinál semmit..
- **+void discard(Item item) :** Mivel egy takarító nem tud égetni tárgyat, ezért a metódus nem csinál semmit.
- **+void death():** Mivel egy takarító nem tud meghalni, ezért a metódus nem csinál semmit.
- **+void useItem(Item item):** Mivel egy takarító nem tud tárgyat használni a metódus nem csinál semmit.
- **+void kill():** Mivel egy takarító nem tud lelket kiszívni, ezért a metódus nem csinál semmit.
- **+void stun():** Mivel egy takarító nem tud lebénulni, ezért a metódus nem csinál semmit.

FakeTVSZ osztályban:

- **+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 égjen el a hallgató eszköztárából.

FakeSlipStick osztályban:

- **+void effect() :** Ez a metódus akkor fut le, ha egy hallgató eszköztárába kerül a "FakeSlipStick" nevezetű tárgy, és ennek hatására nem nyerik meg a hallgatók a játékot, hanem csak elég ez a tárgy.
- **+void onPickUp():** Ha ezt a tárgyat felveszi egy karakter, akkor lép működésbe ez a metódus. Ha egy hallgató vagy egy oktató veszi fel ezt a tárgyat, akkor elég ez a tárgy.
- **+void destruct():** ez a metódus biztosítja, hogy a "FakeSlipStick"-et el lehessen égetni.

FakeFFP2Mask osztályban:

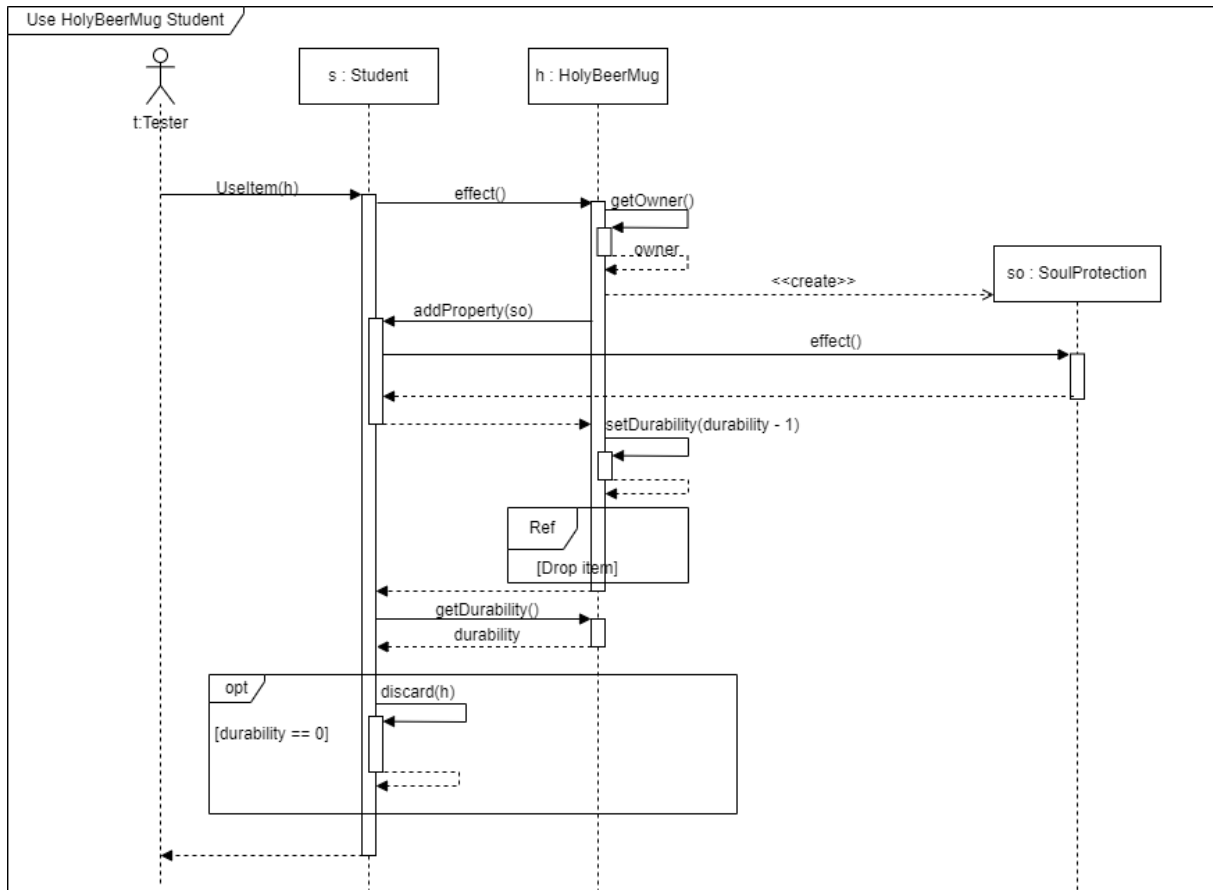
- **+void effect():** Ez a metódus felel azért, hogy ha egy hallgató használná a "FakeFFP2Mask" nevű tárgyat, az ne fejtse ki az eredeti "FFP2Mask" nevű tárgy hatását, hanem csak égjen el a hallgató eszköztárából.

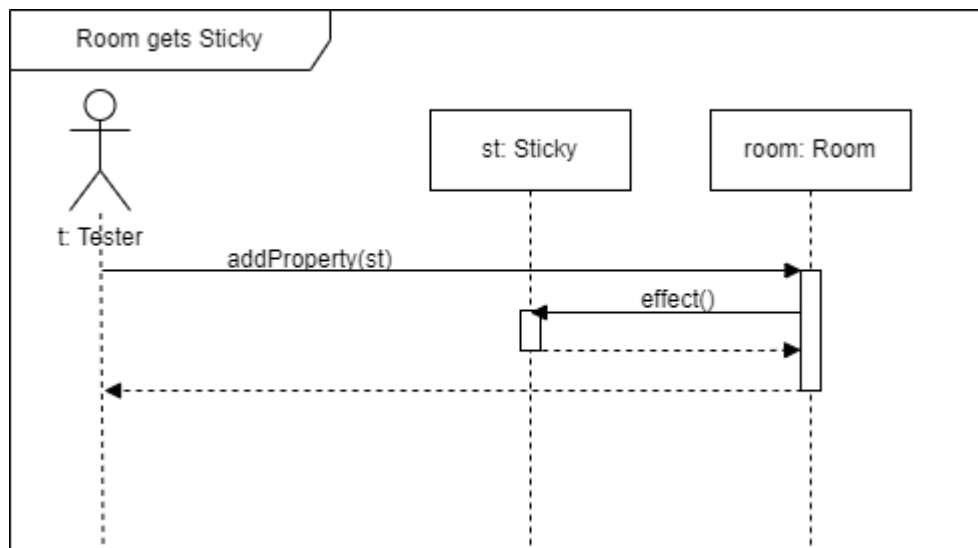
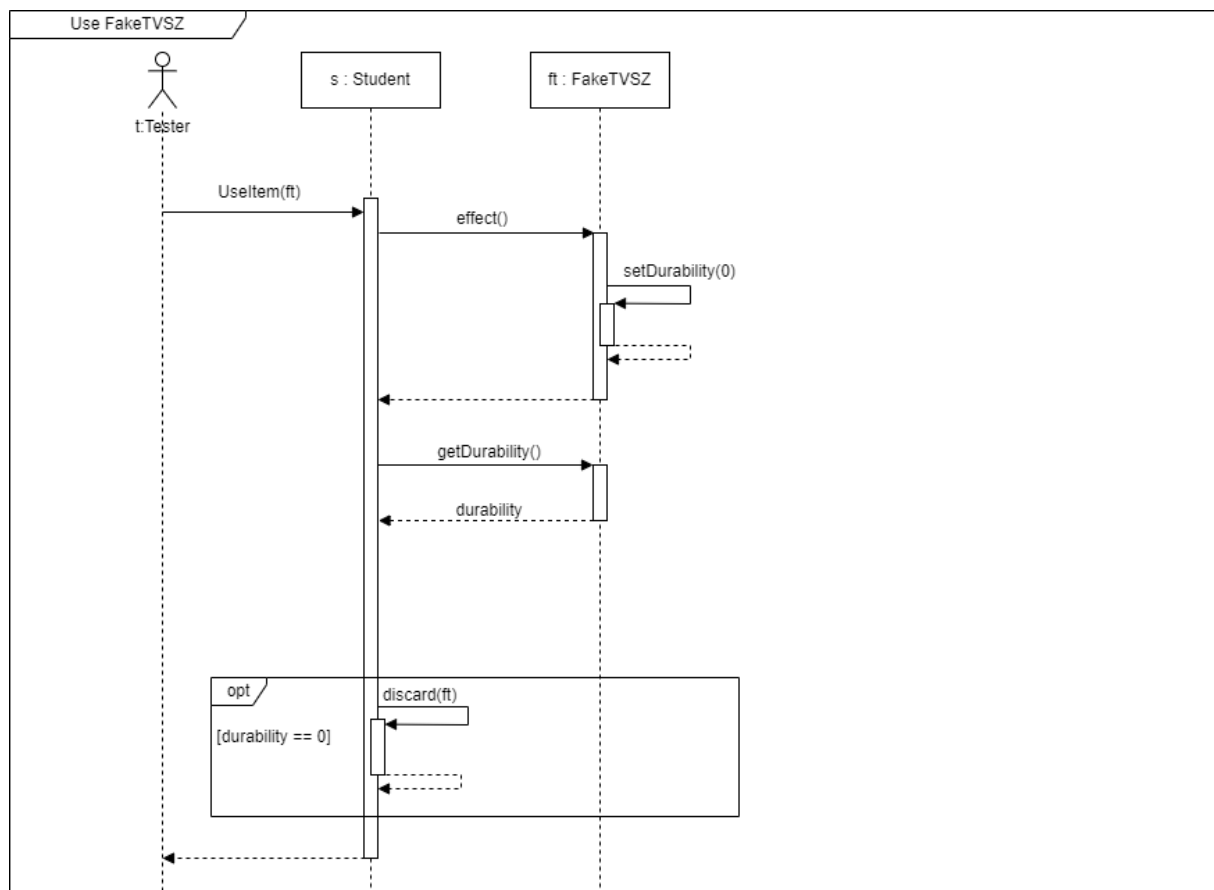
7.0.3 Szekvencia-diagramok

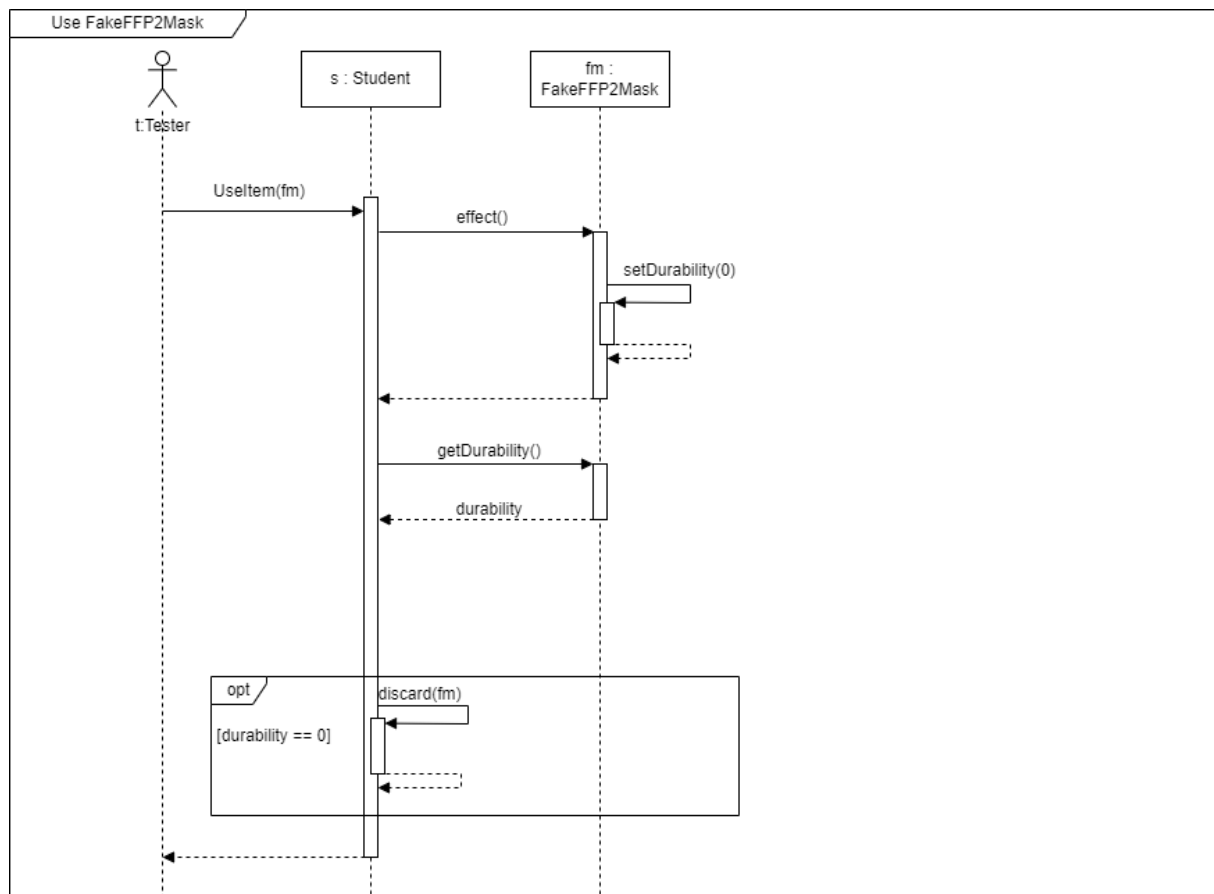
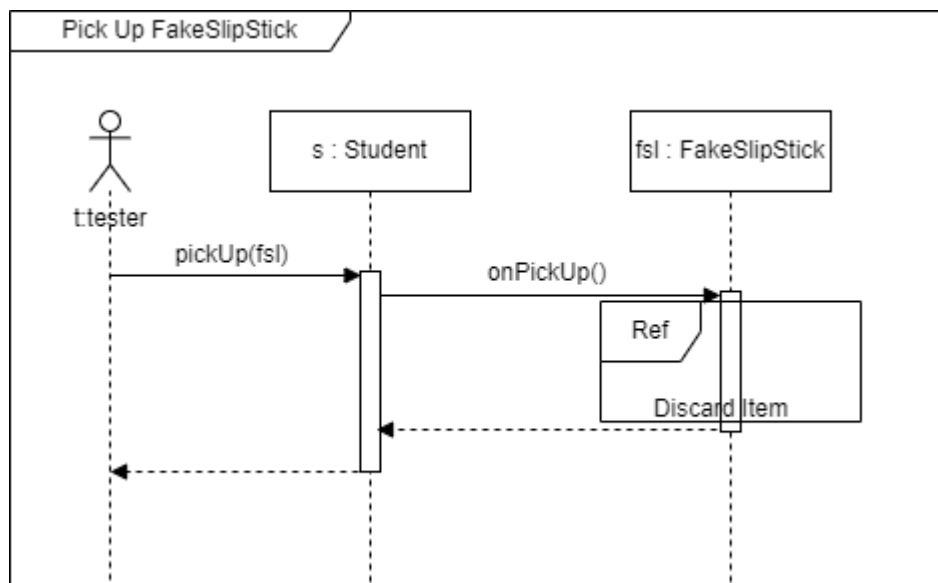
Azon szekvencia diagramok szerepelnek itt, amelyekben változások történtek. Amennyiben egy diagram nem szerepel az előző beadásokhoz képest az azt jelenti, hogy azon nem változott semmi.

Megváltozott diagrammok:

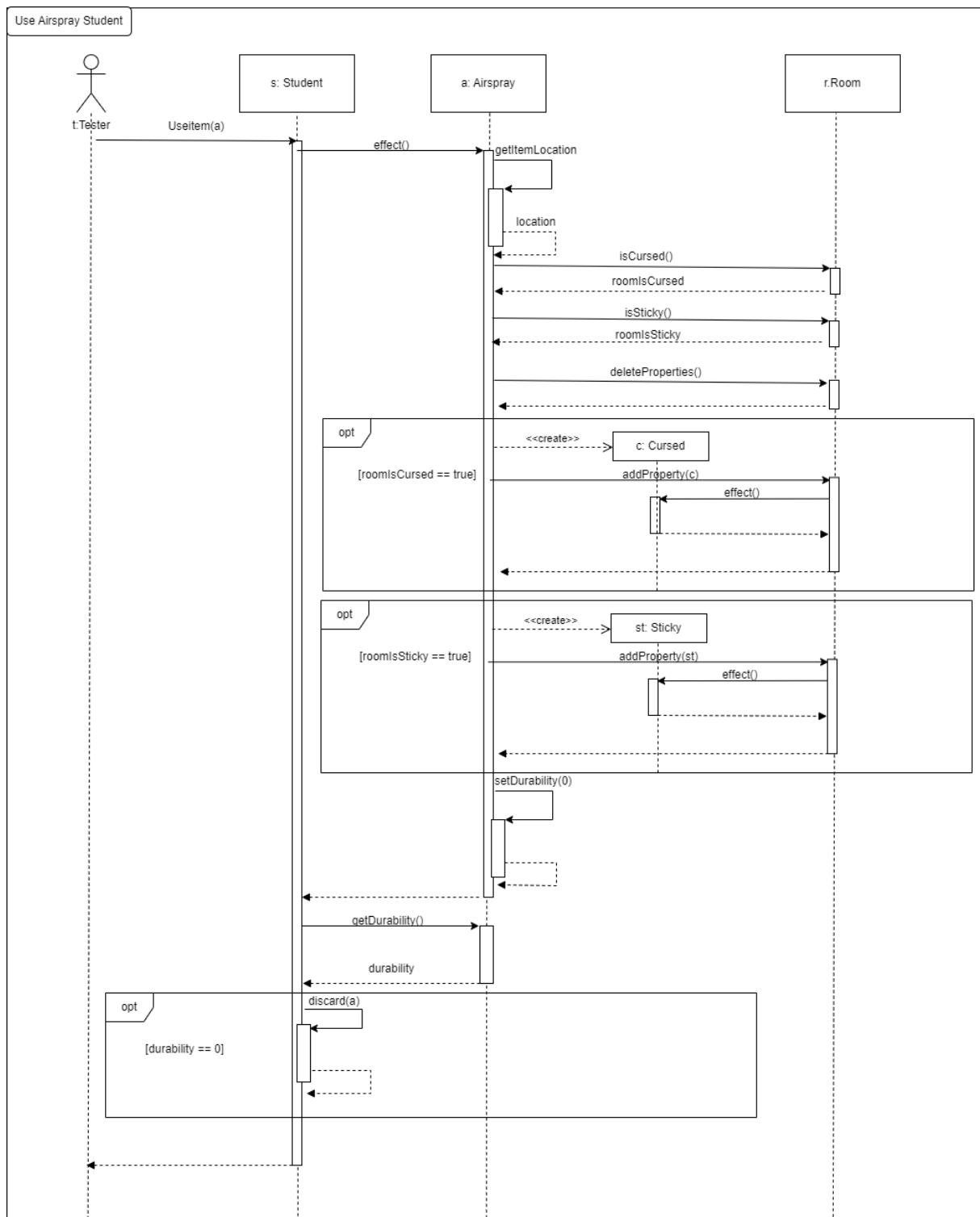
Use HolyBeerMug Student



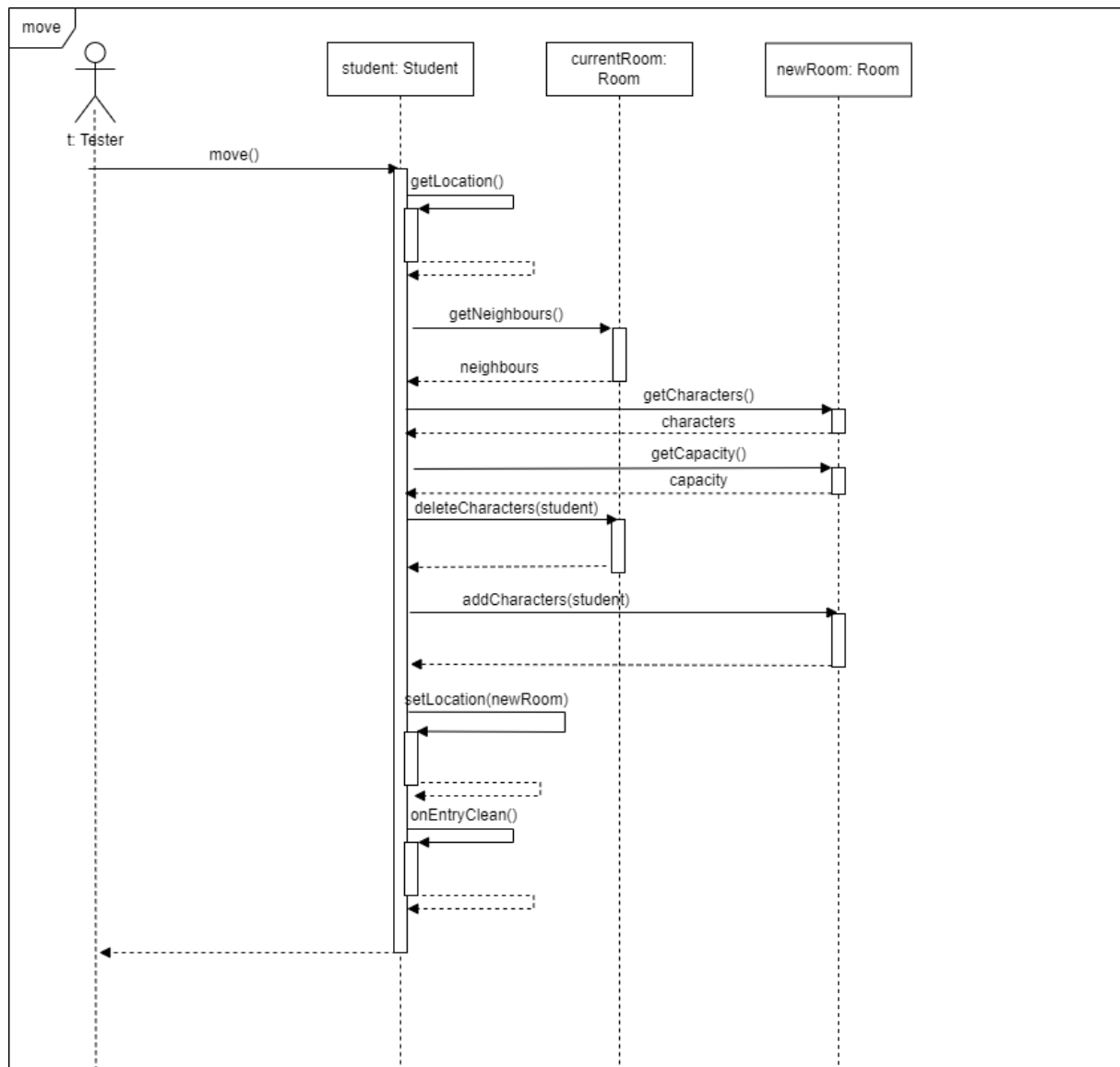
Room gets Sticky**Use FakeTVSZ**

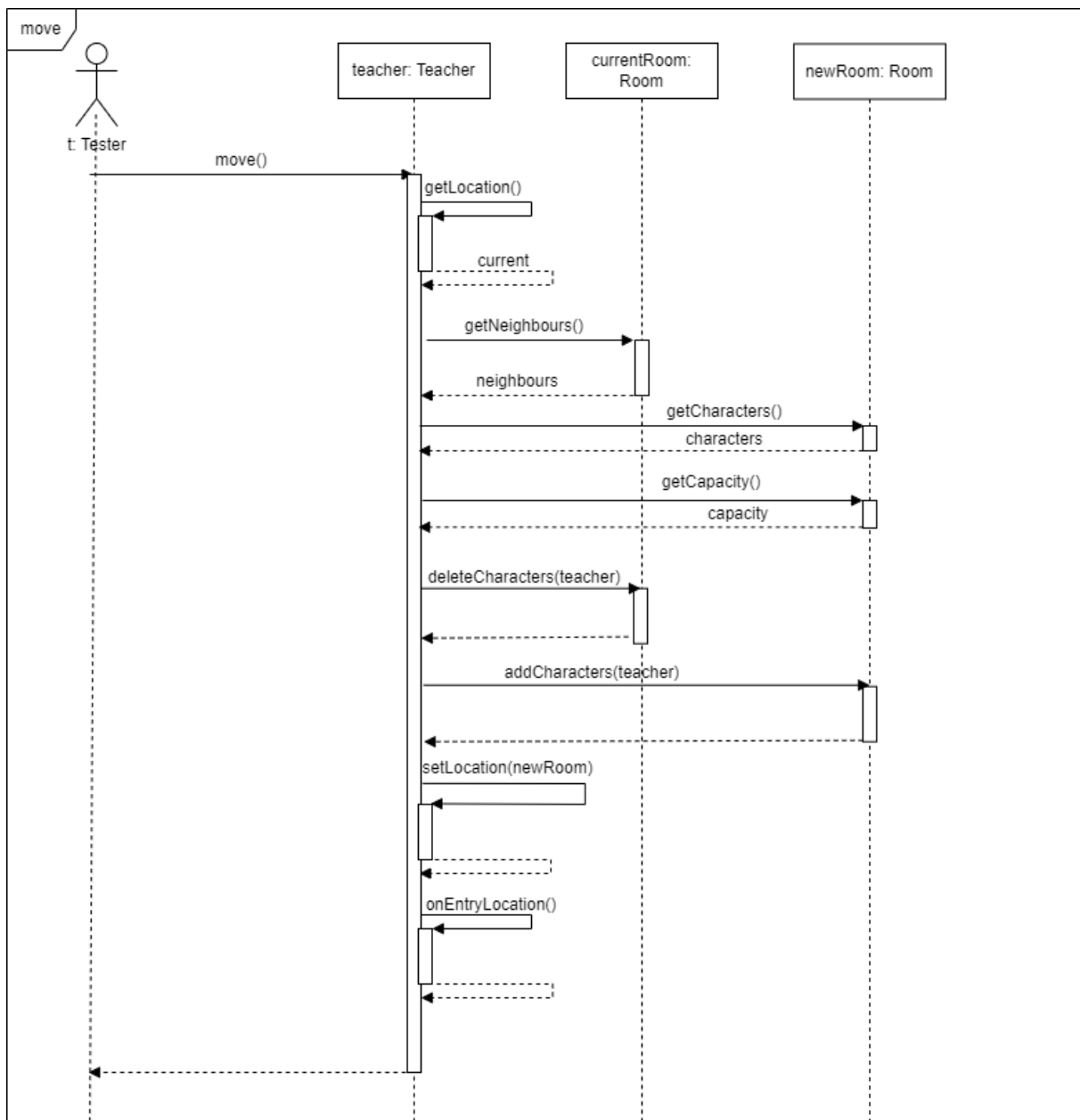
Use FakeFFP2Mask**Pick Up FakeSlipStick**

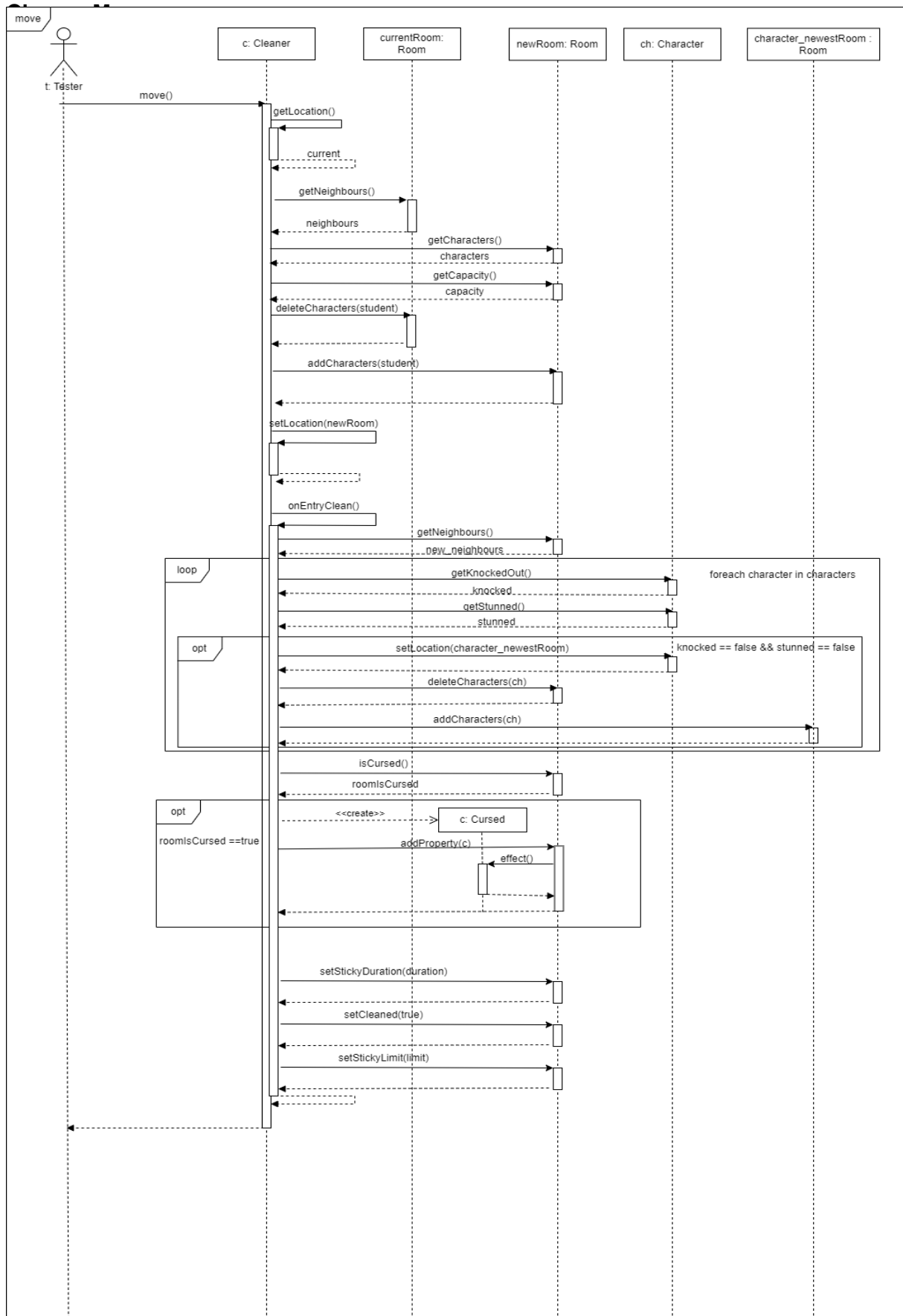
Use Airspray Student



Student move



Teacher move



Prototípus interface-definíciója

7.0.4 Az interfész általános leírása

A parancsok beolvasása történhet konzolról és bemeneti fájlokból is. A parancsok kimenete szintén. Ezek megadására két szöveges fájl szolgál:

- **input.txt**: ha üres, konzolról történik a parancsok beolvasása. Ha meg vannak adva benne további relatív elérési utak pl. test/test1.txt, test/test2.txt soronként, akkor az azoknak megfelelő tesztesetek parancsait is beolvassa, és végrehajtja.
- **output.txt**: ha üres, a konzolra kerül a parancsok kimenete. Ha meg vannak adva benne további relatív elérési utak pl. output/output1.txt, output/output2.txt soronként, akkor a megfelelő input fájlok parancsai a hozzájuk tartozó sorban lévő output fájlba lesznek írva. Fontos tehát, hogyha fájlokat adunk meg input.txt-ben, akkor mindegyikhez megadjuk egy kimeneti fájlt is output.txt-ben (ugyanannyi sorból kell állniuk!).

7.0.5 Bemeneti nyelv

A felhasználó által kiadott parancsok akkor érvényesek, ha Entert nyom a parancs leírása után. Először mindig egy műveletet kell megadni, ami esetünkben lehet move, add room, add item, pickup, use stb. A művelet nevének beírása után a lentebb taglaltak szerint kell követni a parancs kifejtését annak érdekében, hogy a program megfelelően működjön.

A parancsok általános felépítése a következőképpen vizualizálható:

<művelet neve> <művelet 1. paramétere> <művelet 2. paramétere>

Amennyiben több paraméter van az ábrázoltnál, azokat is szóközzel elválasztva kell felsorolni.

7.1.2.1 Parancsok a játék (pálya) előkészítéséhez

Rövid ismertető: Minden elem indexekkel van azonosítva. Az indexeket a játék rendeli az elemekhez a hozzáadás sorrendjében. Ezekkel az indexekkel lehet azonosítani az elemeket. A hozzáadó parancsok kimenetei kiírják ezeket az azonosítókat, de a **print labyrinth** paranccsal később is megtekinthetők az elemek azonosítói.

controller toggle random

Leírás: Ki-/bekapcsol minden véletlenszerű viselkedést a modellben, és helyettük a definiált (és módosítható) értékeket használja, hogy determinisztikus teszteseteket állíthassunk elő. Az alábbiakra van hatással:

- Szobák gázossá tétele a körök elején (manuálisan a **controller poisonous** paranccsal kell megadni majd a controller általi gázosítást helyette, ha szeretnénk ilyet).
- Szobák összeolvasztása a körök elején (manuálisan a **controller merge** paranccsal kell megadni majd a controller általi szoba összeolvasztást helyette, ha szeretnénk ilyet).
- Szobák szétválasztása a körök elején (manuálisan a **controller split** paranccsal kell megadni majd a controller általi szoba szétválasztást helyette, ha szeretnénk ilyet).
- Elátkozott szobák ajtóinak eltűnése a körök elején

- Elátkozott szobák ajtóinak megjelenése a körök elején

add room <room_nb_idx>

Leírás: A megadott indexű elem szomszédjának vesz fel egy szobát.

figyelem: ilyenkor a kölcsönös szomszédosság még nincs biztosítva, csupán a megadott szobából lehet eljutni az új szobába. A kölcsönösség a **connect** paranccsal érhető el.

Opciók: **<room_nb_idx>**: az ezzel megadott indexű elem mellé lesz lehelyezve.

figyelem: az első elem felvételéhez -1-gyel kell jelölni, hogy nincs szomszédja

connect <room1_idx> <room2_idx> <mutual>

Leírás: Szomszédok kialakítására használt parancs. Új elemet nem hoz létre, hanem két szoba között biztosít átjárhatóságot.

Opciók: **<room1_idx>**: a szomszédosság első tagja.

<room2_idx>: a szomszédosság második tagja.

<mutual> 0: az első tagból el lehet jutni a második tagba, de a másodiktól nem az elsőbe.

1: kölcsönösen oda vissza el lehet jutni.

- **disconnect <room1_idx> <room2_idx>**

Leírás: Szomszédok szétválasztására használt parancs. Új elemet nem hoz létre, hanem két szoba között lévő átjárhatóságot megszünteti..

Opciók: **<room1_idx>**: a szomszédosság első tagja.

<room2_idx>: a szomszédosság második tagja.

add character <character_type> <name> <room_idx>

Leírás: A megadott szobában lehelyezi a megadott típusú karaktert, ha a szoba kapacitása ezt megengedi, ellenkező esetben nem történik semmi.

Opciók: **<character_type>**: teacher, student, cleaner

name: a karakter neve. Egyes parancsok esetén és a kimenetben ezzel lehet majd azonosítani.

room_idx: a szoba indexe, ahova a karakter le szeretnénk tenni.

add item <item_type> <room_idx>

Leírás: A megadott szobában lehelyezi a megadott típusú tárgyat.

Opciók: **<item_type>**: tvsz, holybeermug, ffp2, slipstick, wettowel, cabbagecamambert, airspray, faketvsz, fakeffp2, fakeslipstick, transistor

<room_idx>: a szoba indexe, ahova a tárgyat le szeretnénk tenni

- **put item <character_name> <item_type>**

Leírás: A megadott indexű karakter eszköztárába elhelyez egy megadott típusú tárgyat

Opciók:

<character_name> : a karakter neve,, akinek az eszköztárába el szeretnénk helyezni a tárgyat

<item_type>: tvsz, holybeermug, ffp2, slipstick, wettowel, cabbagecamambert, airspray, faketvsz, fakeffp2, fakeslipstick, transistor

Hozzá tartozó kimeneti nyelv

controller poisonous <room_idx>

Leírás: Egy controller általi szoba mérgezővé tétele, azaz olyan, ami nem eredményezi a jelenlegi játékos körének végét, hacsak nem abban a szobában tartózkodik. Ha a szoba már alaptól mérgező volt, nem történik semmi.

Opciók: **<room_idx>**: a mérgezővé tenni kívánt szoba indexe.

controller cursed<room_idx>

Leírás: Egy controller általi szoba átkozottá tétele, azaz olyan, ami nem eredményezi a jelenlegi játékos körének végét. Ha a szoba már alaptól átkozott volt, nem történik semmi.

Opciók: **<room_idx>**: a mérgezővé tenni kívánt szoba indexe.

controller slippery <room_idx>

Leírás: Egy controller általi szoba csúszóssá tétele, azaz olyan, ami nem eredményezi a jelenlegi játékos körének végét. Ha a szoba már alaptól csúszós volt, nem történik semmi.

Opciók: **<room_idx>**: a mérgezővé tenni kívánt szoba indexe.

controller merge <room_idx1> <room_idx2>

Leírás: Két szoba összeolvasztása a controller által. Ha nincs valamely indexű szoba, illetve, ha a két szoba nem szomszédos, akkor nem történik semmi. Nem eredményezi a játékos körének végét, hacsak az összeolvasztás miatt, nem lett mérgező az a szoba, ahol tartózkodik.

Opciók: **<room_idx1>**: az összeolvasztandó szobák első tagjának indexe **<room_idx2>**: az összeolvasztandó szobák második tagjának indexe.

controller split <room_idx>

Leírás: Egy szoba kettővé szétválasztása a controller által. Nem eredményezi a jelenlegi játékos körének végét.

Opciók: <room_idx>: a szoba indexe, amelyet szét szeretnénk választani.

load state <rel_file_name>

Leírás: A megadott fájlból beolvassa, és végrehajtja a parancsokat.

Használható csak a játék előkészítésére, de komplett tesztelés ellenőrzésére is.

Opciók: **rel_file_name:** a fájl relatív elérési útvonala, amelyikből szeretnénk beolvasni az új állapotot.

figyelem: a tesztelő felelőssége, hogy létező útvonalat adjon meg, amelyen egy megfelelő parancsokat tartalmazó és formátumú fájl van.

reset state

Leírás: Alapállapotba állítja az egész játékot (üres pálya, nincsenek játékosok, stb.). A load state parancs előtt nincs szükség kiadni ezt a parancsot, mert alapjáraton megteszi ezt.

start game

Leírás: Elkezd a játékot (A játékosok felváltva kerülnek majd sorra, minden kör elején folyik a víz, stb.). figyelem: a tesztelő felelőssége, hogy a parancs kiadása előtt érvényes kezdeti feltételekkel rendelkező pályát alkosson (pl. megfelelő számú, és sorrendben hozzáadott játékosok legyenek).

A parancs kiadása után minden kör (turn) elején kiírja a következőt: **it's <name>'s turn!**

7.1.2.2 Parancsok a játék játszásához

A **start game** parancs kiadása után mindig a soron lévő játékost lehet irányítani. A következő parancsokat mindig a soron lévő játékos hajtja végre.

move <room_idx>

Leírás: A megadott indexű szobába próbál lépni a jelenlegi szobájából a játékos a hatására.

Ha az adott szoba nem szomszédos elem, vagy olyan szoba, amelyben már a kapacitásával megegyező számú karakter tartózkodik, akkor nem teljesül a parancs.

Ha gázos szobába lép, elkábul és ezzel a köre véget ér.

Opciók: **room_idx:** a szomszédos szoba indexe, amelyre lépni szeretne.

pickup <item_idx>

Leírás: A megadott indexű tárgyat próbálja meg felvenni abban a szobában, ahol tartózkodik a tanuló a hatására. Ha nincs ilyen indexű tárgy a szobában, akkor nem történik semmi. Ha a tanuló ragadós szobában van, akkor semmilyen index hatására nem történik semmi.

Opciók: **item_idx:** a tartózkodási szobájában levő tárgy indexe, amelyet fel szeretne venni.

discard <item_idx>

Leírás: A megadott indexű tárgyat próbálja meg elégetni a tárolójából. Ha nincs ilyen indexű tárgy a tárolójában, akkor nem történik semmi.

Opciók: **item_idx:** a tárolójában levő tárgy indexe, amelyet el szeretne égetni.

drop <item_idx>

Leírás: A megadott indexű tárgyat próbálja letenni abban a szobában, ahol tartózkodik a tanuló a hatására. Ha nincs ilyen indexű tárgy a tárolójában, akkor nem történik semmi.

Opciók: **item_idx:** a tárolójában levő tárgy indexe, amelyet le szeretne tenni.

use <item_idx>

Leírás: A megadott indexű tárgyat próbálja felhasználni abban a szobában, ahol tartózkodik a játékos a hatására. Ha nincs ilyen indexű tárgy a tárolójában, akkor nem történik semmi. (Ha sikerült felhasználnia a tárgyat elég a tárolójából)

tanuló	TVSZ	A játékos védetté válik az oktatóktól.
	HolyBeerMug	A játékos védetté válik az oktatóktól.
	FFP2 Mask	A játékos védetté válik a mérges gáztól.
	AirSpray	Eltávolítja a mérgező gázt a szobából, ahol tartózkodik. Ha eleve tiszta szobában használja fel, nem történik semmi.
	WetTowel	Lebénítja a szobában levő összes oktatót, ahol tartózkodik.

	Transistor	Elteleportál abba a szobába, ahova a másik tranzisztor lett téve. Ha olyan traszisztort próbál meg felhasználni, amelynek nincs párja vagy a párja nincs letéve, akkor nem történik semmi.
	SlipStick	- (!nem lehet felhasználni, felvétele automatikusan a játék megnyeréséhez vezet)
	CabbageCamambert	Mérgezővé teszi a szobát, ahol tartózkodik.
	Fake TVSZ	-
	Fake Slipstick	-
	Fake FFP2 Mask	-

Opciók: `room_idx`: a tárolójában levő tárgy indexe, amelyet fel szeretne használni.

skip turn

Leírás: A kör passzolása: véget vet a jelenlegi játékos körének (turn), anélkül, hogy olyan interakciót tenne, amely véget vetne neki.

kill

Leírás: Az oktató megpróbálja megölni a vele egy szobába került hallgatókat

7.1.2.3 Parancsok az objektumok ellenőrzéséhez

print inventory <character_name>

Leírás: Kiírja a megadott nevű karakter tárolójának tartalmát és annak jellemzőit.

Opciók: `character_name`: a karakter neve, akinek ki szeretnénk írni, hogy mi van a tárolójában.

print currRoom <character name>

Leírás: Kiírja a megadott nevű játékos jelenlegi szobájának(ahol tartózkodik) jellemzőit.

Opciók: **character_name:** a játékos neve

print room <room_idx>

Leírás: A megadott indexű szoba jellemzőit írja ki.

Opciók: **room_idx:** a szoba indexe, amelynek ki szeretnénk írni a jellemzőit.

print labyrinth

Leírás: A pálya összes szobájának kiírja a jellemzőit.

7.0.6 Kimeneti nyelv

A kimenetek a következő módon jelennek meg:

Miután a felhasználó beírta azt a bemeneti parancsot, amelyet meg akart valósítani, azt követően a bemeneti parancsot követő sorban megjelenik a hozzá tartozó kimeneti parancs.

A kimeneti parancs a következőképpen vizualizálható:

Kimenet: <bemeneti parancs hatásának leírása>

controller toggle random

Kimenet: random behaviour was turned off/on

figyelem: a random viselkedés bekapcsolásánál, a random bekövetkezett műveleteknél is megjelenik a kimeneten, hogy mi történt. Azoknál a műveleteknél, amelyek manuálisan is elvégezhetők, a kiemenet megegyezik.

Továbbiak: ajtók eltűnése: **door was deleted between <room1_idx>.room and <room2_idx>.room**

ajtók megjelenése: **door was added between <room1_idx>.room and <room2_idx>.room**

add room <room_nb_idx>

Kimenet: a room was added as a neighbour of <room_nb_idx>.room

-Abban az esetben ha a <room__idx> paraméter értéke -1 a kimenet:
the first room has been added to the labyrinth

connect <room1_idx> <room2_idx> <mutual>

Kimenet: a mutual parancstól függően értelemszerűen az első vagy mindkettő

a <room1_idx>.room became neighbour of <room2_idx>.room

a <room2_idx>.room became neighbour of <room1_idx>.room

disconnect <room1_idx> <room2_idx>

Kimenete: a <room1_idx>.room is not longer a neighbour of <room1_idx>.room

add character <character_type> <name> <room_idx>

Kimenet: a teacher/student/cleaner was added to <room_idx>.room

add item <item_type> <room_idx>

Kimenet: a tvsz/holybeermug/ffp2/slipstick/wettowel/cabbagecamambert/airspray/faketvsz/fakeffp2/fakesplistick/transistor was added to <room_idx>.room

controller poisonous <room_idx>

Kimenet: <room_idx>.room was poisoned by controller.

controller cursed <room_idx>

Kimenet: <room_idx>.room was cursed by controller.

controller slippery <room_idx>

Kimenet: <room_idx>.room was made slippery by controller.

controller merge <room_idx1> <room_idx2>

Kimenet: <room_idx1>.room was merge with <room_idx2> by controller

-Abban az esetben ha a <room_idx1> és <room_idx2> nem szomszédosak a kimenet:

<room_idx1>.room and <room_idx2>.room couldn't merge, because they aren't neighbours

controller split <room_idx>

Kimenet: <room_idx>.room was split by controller

load state <rel_file_name>

Kimenet: loading state from <rel_file_name>

<a fájlban lévő parancsoknak megfelelő sorok...

put item <character_name> <item_type>

Kimenet: a <item_type> was added to <character_name>'s inventory

reset state

Kimenet: game state was reset

start game

Kimenet: game started

A parancs kiadása után minden kör (turn) elején kiírja a következőt: **it's <name>'s turn!**

7.1.2.2 Parancsok a játék játszásához

move <room_idx>

Kimenet eddig: player <name> moved to <room_idx>.room

-Amennyiben a karakter nem tudja a lépést végrehajtani:

player <character_name> tried to move to <room_idx>.room, but could not do it

-Amennyiben a szoba ragadóssá vált:

limit exceeded - room 0 became slippery

-Amennyiben a szoba mérgező és a hallgató nincs védve a mérreg ellen:
<character_name> knocked out

-Amennyiben a tanár lebénult a lépés közben:
teacher is stunned

pickup <item_idx>

Kimenet:

igazi Logarléc esetében	fake Logarléc esetén	minden más tárgy esetében
player <name> picked up <item_idx>.item (<item_name>) from <room_idx>.room	player <name> picked up <item_idx>.item (<item_name>) from <room_idx>.room	player <name> picked up <item_idx>.item (<item_name>) from <room_idx>.room

Students won the game	it was fake	
-----------------------	-------------	--

Abban az esetben ha a hallgató nem tudja felvenni a tárgyakat mert a szoba ragacsos:

<character_name> could not pick up item, because room is slippery

discard <item_idx>

Kimenet: player <name> discarded <item_idx>.item (<item_name>) from inventory

drop <item_idx>

Kimenet: player <name> dropped <item_idx>.item (<item_name>) from inventory to <room_idx>.room

use <item_idx>

Kimenet: Értelemszerűen az alábbiak közül a megfelelő.

player <name> get protected from teachers

player <name> get protected from poison

player <name> used air spray in <room_idx>.room

player <name> stunned teachers in <room_idx>.room

player <name> teleported from <room_idx>.room to <newroom_idx>.room

player <name> made <room_idx>.room poisonous

it was fake

skip turn

Kimenet: <character_name> has skipped a turn

Leírás: A kör passzolása: véget vet a jelenlegi játékos körének (turn), anélkül, hogy olyan interakciót tenne, amely véget vetne neki.

kill

Kimenet: teacher tried to kill the students in the room

7.1.2.3 Parancsok az objektumok ellenőrzéséhez

print inventory <character_name>

Kimenet:

- -Inventory kiírás előtt megjelenik az alábbi szöveg az olvashatóság miatt:

<character_name>'s inventory:

Tárgyak a tranzisztor kivételével	Transistor
details of item <i> name: <iname> durability: <value> currentlyUsed: true/false	details of item <i> name: <iname> durability: <value> currentlyUsed: true/false connected: true/false

print currRoom <charahter name>

Kimenet: lásd a **print room <room_idx>** parancsot

print room <room_idx>

Kimenet: details of room <i>:

cursed: true/false

poisonous: true/false

slippery: true/false

cleaned: true/false

capacity: <value>

stickylimit:<value>

neighbours: <nb_idx1> <nb_idx2> ...

items: lásd, mint az inventory kimeneténél

print labyrinth

Kimenet: A pálya összes szobájára **print room <room_idx>** parancs az indexek szerint növekvő sorrendben.

7.1 Összes részletes use-case

Use-case neve	add room <room_nb_idx> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására hozzáadásra kerül egy szoba.</i>
Aktorok	Tester
Forgatókönyv	<ol style="list-style-type: none"> 1. A program ellenőrzi, hogy került-e már szoba hozzáadásra. Amennyiben nem, és a parancsban megadott szám nem -1, akkor nem történik semmi. 2. Amennyiben nem került még szoba hozzáadásra és a parancsban szereplő szám -1, akkor teljesül a parancs. 3. Amennyiben létezik az a szoba, aminek az indexe megegyezik a parancsban szereplő számmal, akkor egy új, vele szomszédos szoba kerül hozzáadásra. Ha nem létezik, akkor nem teljesül a parancs.

Use-case neve	connect <room1_idx> <room2_idx> <mutual> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására a megadott 2 szoba szomszédossá válik a megadott módon.</i>
Aktorok	Tester
Forgatókönyv	<ol style="list-style-type: none"> 1. A program ellenőrzi, hogy létezik-e <room1_idx> és <room2_idx> azonosítójú szoba a labirintusban, illetve, hogy a <mutual> paraméter értéke érvényes. <p>Ha teljesülnek a feltételek:</p> <ol style="list-style-type: none"> 2. A két szoba szomszédossá válik a parancs leírásában definiált módon. <p>Ha nem teljesülnek a feltételek, nem történik semmi.</p>

Use-case neve	add character <character_type> <name> <room_idx> parancshoz tartozó use-case.
---------------	--

Rövid leírás	<i>A parancs hatására hozzáadásra kerül egy megadott típusú karakter a megadott névvel a megadott szobába.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program ellenőrzi, hogy a <character_type> és a <room_idx> paraméter értéke érvényes.</p> <p>Ha érvényesek:</p> <p>2. A program ellenőrzi, hogy a <room_idx> azonosítójú szoba nincs-e tele.</p> <p>3. Ha nincs tele, akkor a <room_idx> azonosítójú szobába hozzáadásra kerül egy <character_type> típusú karakter <name> néven.</p> <p>Ha nem érvényesek, nem történik semmi.</p>

Use-case neve	add item <item_type> <room_idx> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására egy megadott típusú tárgy kerül a megadott szobába.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program ellenőrzi, hogy létezik-e <room_idx> azonosítójú szoba a labirintusban, illetve, hogy a <item_type> paraméter értéke érvényes.</p> <p>Ha teljesülnek a feltételek:</p> <p>2. Hozzáadásra kerül egy <item_type> típusú tárgy a <room_idx> azonosítójú szobába.</p> <p>Ha nem teljesülnek a feltételek, nem történik semmi.</p>

Use-case neve	controller poisonous <room_idx> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására a megadott szoba mérgezővé válik.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program ellenőrzi, hogy létezik-e a <room_idx> azonosítójú szoba.</p> <p>Ha teljesülnek a feltételek:</p> <p>2. A megadott szoba mérgezővé válik.</p> <p>Ha nem teljesülnek a feltételek, nem történik semmi.</p>

Use-case neve	controller cursed <room_idx> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására a megadott szoba elátkozottá válik.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program ellenőrzi, hogy létezik-e a <room_idx> azonosítójú szoba.</p> <p>Ha teljesülnek a feltételek:</p> <p>2. A megadott szoba elátkozottá válik.</p> <p>Ha nem teljesülnek a feltételek, nem történik semmi.</p>

Use-case neve	controller slippery <room_idx> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására a megadott szoba csúszóssá válik.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program ellenőrzi, hogy létezik-e a <room_idx> azonosítójú szoba.</p> <p>Ha teljesülnek a feltételek:</p> <p>2. A megadott szoba csúszóssá válik.</p> <p>Ha nem teljesülnek a feltételek, nem történik semmi.</p>

Use-case neve	controller merge <room_idx1> <room_idx2> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására a megadott szobák egyesülnek.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program ellenőrzi, hogy létezik-e a <room_idx1> és <room_idx2> azonosítójú szoba.</p> <p>Ha teljesülnek a feltételek:</p> <p>2. A két szoba egyesül.</p> <p>Ha nem teljesülnek a feltételek, nem történik semmi.</p>

Use-case neve	controller split <room_idx> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására a megadott szoba osztódik.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program ellenőrzi, hogy létezik-e a <room_idx> azonosítójú szoba.</p>

	<p>Ha teljesülnek a feltételek:</p> <ol style="list-style-type: none"> 2. A szoba kettéosztódik, két új szobát létrehozva ezzel. <p>Ha nem teljesülnek a feltételek, nem történik semmi.</p>
--	---

Use-case neve	load state <rel_file_name> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására betöltődnek és végrehajtódnak a megadott fájlban lévő parancsok.</i>
Aktorok	Tester
Forgatókönyv	<ol style="list-style-type: none"> 1. A program ellenőrzi, hogy létezik-e a <rel_file_name> fájl. <p>Ha teljesülnek a feltételek:</p> <ol style="list-style-type: none"> 2. Betöltődnek és végrehajtódnak a fájlban tárolt parancsok. <p>Ha nem teljesülnek a feltételek, nem történik semmi.</p>

Use-case neve	reset state parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására a játék alapállapotba áll.</i>
Aktorok	Tester
Forgatókönyv	<ol style="list-style-type: none"> 1. A játék alapállapotba áll a parancs leírásánál definiált módon.

Use-case neve	start game parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására a játék elindul.</i>
Aktorok	Tester
Forgatókönyv	<ol style="list-style-type: none"> 1. A játék elindul, az első játékos sorra kerül.

Use-case neve	move <room_idx> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására az éppen körön lévő karakter a megadott szobába lép.</i>
Aktorok	Tester
Forgatókönyv	<ol style="list-style-type: none"> 1. A program megvizsgálja, hogy van-e a karakter jelenlegi szomszédai között <room_idx> azonosítójú szoba. <p>Ha van <room_idx> azonosítójú szoba:</p> <ol style="list-style-type: none"> 2. A program megvizsgálja, hogy a megadott szobában van-e még hely.

	<p>3. Ha igen, a karakter a megadott szobába lép.</p> <p>Ha nincs <room_idx> azonosítójú szoba, nem történik semmi.</p>
--	--

Use-case neve	pickup <item_idx> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására az éppen körön lévő karakter felveszi a megadott tárgyat.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program megvizsgálja, hogy van-e a karakter szobájában <item_idx> azonosítójú tárgy.</p> <p>Ha van <item_idx> azonosítójú tárgy:</p> <p>2. A program megvizsgálja, hogy van-e hely a karakter eszköztárában.</p> <p>3. Ha van, a karakter felveszi a megadott tárgyat.</p> <p>Ha nincs <item_idx> azonosítójú tárgy, nem történik semmi.</p>

Use-case neve	discard <item_idx> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására az éppen körön lévő karakter elégeti a megadott tárgyat.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program megvizsgálja, hogy van-e a karakter eszköztárában <item_idx> azonosítójú tárgy.</p> <p>Ha van <item_idx> azonosítójú tárgy:</p> <p>2. A karakter elégeti a megadott tárgyat.</p> <p>Ha nincs <item_idx> azonosítójú tárgy, nem történik semmi.</p>

Use-case neve	drop <item_idx> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására az éppen körön lévő karakter leteszi a megadott tárgyat.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program megvizsgálja, hogy van-e a karakter eszköztárában <item_idx> azonosítójú tárgy.</p> <p>Ha van <item_idx> azonosítójú tárgy:</p> <p>2. A karakter leteszi a megadott tárgyat.</p>

	Ha nincs <item_idx> azonosítójú tárgy, nem történik semmi.
--	---

Use-case neve	use <item_idx> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására az éppen körön lévő karakter használja a megadott tárgyat.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program megvizsgálja, hogy van-e a karakter eszköztárában azonosítójú <item_idx> tárgy.</p> <p>Ha van <item_idx> azonosítójú tárgy:</p> <p>2. A karakter használja a megadott tárgyat, így az kifejti hatását.</p> <p>Ha nincs <item_idx> azonosítójú tárgy, nem történik semmi.</p>

Use-case neve	next turn parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására az éppen körön lévő karakter befejezi a saját körét.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A karakter köre véget ér.</p> <p>2. A soron következő karakter köre elkezdődik.</p>

Use-case neve	print inventory <character_name> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására kiíródik a megadott karakter eszköztárának tartalma.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program megvizsgálja, hogy létezik-e a <charahter name> nevű karakter.</p> <p>Ha létezik <charahter name> nevű karakter:</p> <p>2. Kiíródik a szabványos kimenetre a karakter eszköztárának tartalma a parancs meghatározott kimenetének megfelelően.</p> <p>Ha nem létezik <charahter name> nevű karakter, nem történik semmi.</p>

Use-case neve	print currRoom <character name> parancshoz tartozó use-case.
----------------------	---

Rövid leírás	<i>A parancs hatására kiíródnak a megadott karakter jelenlegi szobájának jellemzői.</i>
Aktorok	Tester
Forgatókönyv	<p>3. A program megvizsgálja, hogy létezik-e a <character name> nevű karakter.</p> <p>Ha létezik <character name> nevű karakter:</p> <p>4. Kiíródnak a szabványos kimenetre a karakter jelenlegi szobájának jellemzői a parancs meghatározott kimenetének megfelelően.</p> <p>Ha nem létezik <character name> nevű karakter, nem történik semmi.</p>

Use-case neve	print room <room_idx> parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására kiíródnak a megadott szoba jellemzői.</i>
Aktorok	Tester
Forgatókönyv	<p>1. A program megvizsgálja, hogy létezik-e a <room_idx> azonosítójú szoba.</p> <p>Ha létezik <room_idx> azonosítójú szoba:</p> <p>2. Kiíródnak a szabványos kimenetre szoba jellemzői a parancs meghatározott kimenetének megfelelő formátumban.</p> <p>Ha nem létezik <room_idx> azonosítójú szoba, nem történik semmi.</p>

Use-case neve	print labyrinth parancshoz tartozó use-case.
Rövid leírás	<i>A parancs hatására kiíródnak a pálya összes szobájának jellemzői.</i>
Aktorok	Tester
Forgatókönyv	<p>1. Kiíródnak a szabványos kimenetre labirintus összes szobájának jellemzői a parancs meghatározott kimenetének megfelelő formátumban.</p>

Use-case neve	controller toggle random parancshoz tartozó use case.
Rövid leírás	A parancs hatására bekapcsol, illetve kikapcsol a programban szereplő, nem a felhasználó által előidézett determinisztikus változások.

Aktorok	Tester
Forgatókönyv	<ol style="list-style-type: none">1. Amennyiben a parancs páratlan sokszor került megadásra, akkor újbóli megadáskor kikapcsol.2. Amennyiben a parancs páros sokszor került megadásra, akkor újbóli megadáskor bekapcsol.

7.2 Tesztelési terv

Teszt-eset neve	Teacher és Student kapcsolata
Rövid leírás	Teacher megpróbálja megölni a vele egy szobában lévő hallgatókat, de azok a náluk lévő tárgyakkal megvédik magukat
Teszt célja	Letesztelni, hogy megfelelően működnek a játék legfontosabb aspektusai: 1. Teacher hatása az egy szobában lévő hallgatókra 2. Hallgatók a náluk lévő tárgyakat felhasználják 3. Tárgyak megfelelően fejtik ki hatásukat

Teszt-eset neve	Cleaner hatása
Rövid leírás	Cleaner belép egy olyan szobába, ami gázos és vannak benne emberek, akik nincsenek elájulva és lebénulva
Teszt célja	Letesztelni, hogy megfelelően működnek a játék fontosabb aspektusai: 1. A Cleaner a gázos szobát gáztalanítja 2. A Cleaner áthelyezi a nem elájult/lebénult hallgatókat 3. A hallgatóknak a labirintusban elfoglalt pozíciója valóban megváltozik

Teszt-eset neve	Szoba kettéoszlik
Rövid leírás	Az egyik szoba két részre szakad
Teszt célja	Letesztelni, hogy megfelelően működnek a játék fontosabb aspektusai: 1. Az új szobák megfelelően öröklik meg az eredeti szoba tulajdonságait 2. Az új szobák között megfelelően oszlanak el a tárgyak 3. Az új szobák között megfelelően oszlanak el a tárgyak

Teszt-eset neve	Két szoba egyesül
Rövid leírás	Két szoba egy szobává egyesül
Teszt célja	Letesztelni, hogy megfelelően működnek a játék fontosabb aspektusai: 1. Az új szoba megfelelően öröklik meg az eredeti szoba tulajdonságait 2. Az új szoba között megfelelően oszlanak el a tárgyak 3. Az új szoba között megfelelően oszlanak el a tárgyak

megjegyzés: a tesztek helyességének ellenőrzése „manuális” megfontolással, a kimenetek és a print-ek várt és kapott formájának összevetésével fog történni.

7.3 Tesztelést támogató segéd- és fordítóprogramok specifikálása

Nincs szükség egyéb segéd- és fordítóprogramokra.

7.4 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2024.03.28. 10:00	5 óra	Fazekas	Osztálydiagram kiegészítése
2024.03.28 15:30	5 óra	Molnár	Osztálydiagram további kiegészítése
2024.03.29 21:00	3 óra	Rimmel	Korábbi Szkeleton kiegészítése az új osztálydiagram alapján
2024.03.31 21:00	2 óra	Fazekas Molnár Simon Juhász Rimmel	Értekezlet: Döntés a további feladatok kiosztásáról
2024.04.01 21:00	3 óra	Simon	Bemeneti parancsok megírása
2024.04.02 12:00	2 óra	Simon	Kimeneti parancsok megírása
2024.04.03 15:00	5 óra	Juhász	Use-casek kidolgozása.
2024.04.06 19:00	2 óra	Rimmel	Tesztelési tervek kidolgozása