

8. Részletes tervek

15 – Csinibaba

Konzulens:
Goldschmidt Balázs

Csapattagok

Havasi Benedek Solt
Kaposvári Gergő János
Reszegi Áron
Hőgye Dominik
Soós Márton

BOFMK5
CTL2L7
WLN2SO
QMVVUP
LWZD5C

havasib@edu.bme.hu
kaposvari.gergo.2003.23@gmail.com
aronreszegi2@gmail.com
hogye.dominik@gmail.com
martonxyz@gmail.com

2024.04.14

8. Részletes tervek

8.0 Interface javítása

Az objektumok létrehozásukkor automatikusan kapnak nevet. A számozás minden különböző típusú objektum esetén 0-tól kezdődik és növekvő sorrendben egész számokkal történik. Ahol bool típusú változókat kell beállítani, ott a – és + jelekkel kell jelezni a kívánt állapotot. Bemeneti fájlban ugyanazok a parancsok érvényesek, mint amik terminálból elérhetők. Minden sor pontosan egy parancsot tartalmazhat, a fájl utolsó sora üres sor. Mentéskor a program bemeneti parancsokat ír ki a kimeneti fájlba, hogy betöltéskor ebből visszaálljon a kívánt játékmenet. Az objektumok kiírási sorrendje: szobák, ajtók, karakterek, tárgyak.

Bemeneti nyelv

NewRoom

Leírás: Új szoba létrehozása, opciók megadásával

Opciók: gázos, ruhás, férőhely, mágikus, ragacsos, belépők takarítás óta

NewStudent

Leírás: Új tanuló létrehozása, opciók megadásával

Opciók: dazed, szoba

NewTeacher

Leírás: Új tanár létrehozása, opciók megadásával

Opciók: dazed, clothed, szoba

NewCleaner

Leírás: Új takarító létrehozása, opciók megadásával

Opciók: dazed, szoba

NewItem

Leírás: Új tárgy létrehozása, opciók megadásával

Opciók: item típusa (1- transistor, 2 – cloth, 3 – mask, 4 – camembert, 5 – book, 6 – beer, 7 – air freshener, 8 – sliding ruler), szoba, karakter, (egyes esetekben fennmaradó használat), fake

NewDoor

Leírás: Új ajtó létrehozása, két szoba között. A neve dxy ahol x a kisebb sorszámú szoba száma, y a nagyobbé.

Opciók: kisebb sorszámú szoba, nagyobb sorszámú szoba, egyirányú, eltűnt

SetTime

Leírás: A hátralévő idő beállítása.

Opciók: kívánt idő

Split

Leírás: Egy szoba szétválasztása.

Opciók: szoba

Merge

Leírás: Két szoba egyesítése.

Opciók: egyik szoba, másik szoba

Kimeneti nyelv

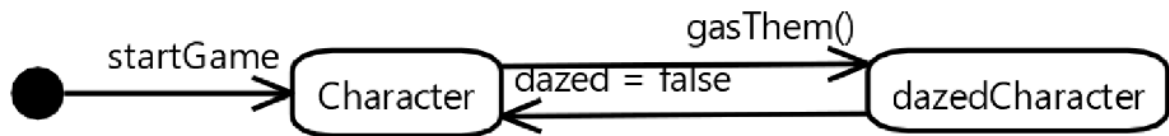
Az objektum létrejöttékor a program kiírja az új objektum nevét. (Példa: NewStudent – r1 parancsra a kimenet: s0 created in r1).

8.1 Osztályok és metódusok tervei.

8.1.1 Character

- **Felelősség**

Ez az osztály a számítógép és játékosok által irányított emberek ősosztálya. Minden karakter vehet fel tárgyakat és mozoghat a szobák között. Minden karakterre vonatkozik a szobák férőhelye. Elkábulhatnak a gáztól.



- **Ősosztályok**

-

- **Interfészek**

-

- **Attribútumok**

- - **bool dazed**: Igaz, ha a karakter gáztól kábult állapotban van, hamis, ha nincs.
- - **Room currentRoom**: Az a szoba amelyikben jelenleg a karakter tartózkodik
- - **Item[5] inventory**: Itt tárolódnak a karakternél fellelhető tárgyak.

- **Metódusok**

- + **void enterRoom(door D)**: belép az ajtó túloldalán található (azaz az ajtó által eltárolt két szoba közül abba amiben nem tartózkodik jelenleg a karakter)
- + **void pickUpItem(Item i)**: A karakter megpróbálja felvenni a paraméterként kapott tárgyat
- + **void dropItem(Item i)**: A paraméterként kapott tárgy az inventory-ból a jelenlegi szobába a „földre” kerül.
- + **void dropEverything()**: A karakter eldobja az összes nála lévő tárgyat, ez akkor következik be, ha elkábult és nincs maszkja.
- + **void removeItem(Item i)**: A paraméterként kapott tárgy kitörlődik a játékból, például ha elhasználódik.
- + **void useItem()**: a paraméterként kapott tárgyat megpróbálja használni, ilyen tárgyak: légfrissítő, camembert, transistor.
- + **void updateRoom(Room r)**: Beállítja a jelenlegi szobát a paraméterként kapott szobára.
- + **void teacherDuty()**: Ha egy karakter tanár, akkor megpróbál mindenkit kicsapni a szobában, ha nem akkor nem csinál semmit.
- + **void setDazed(boolean dazed)**: beállítja a Dazed változót a paraméterként átvett értékre.
- + **void getDazed()**: Visszaadja a
- + **Room getCurrentRoom()**: visszaadja a szobát amiben a karakter tartózkodik
- + **void setClothed(int i)**: csak definíció, nem valósít meg lényegi működést a karakter osztályban.

- **+int dropOut():** (-1)-el tér vissza, ezzel jelezve ha a karakter nem hallgató, a szobának jelez hogy ne buktassa ki az egyetemről.

8.1.2 Student

- **Felelősség**

A hallgatók a játékosok által irányítható karakterek. Rendelkeznak a karakterek összes tulajdonságával, ezen kívül az összes tárgyat képesek használni is. Céljuk a logarléc megtalálása a megadott időn belül, ezzel megnyerve a játékot.

- **Ósosztályok**

Character → Student

- **Interfészek**

-

- **Attribútumok**

-

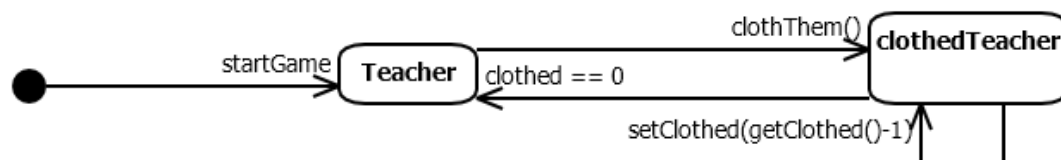
- **Metódusok**

- **+void connect(Transistor t1, Transistor t2):** A két paraméterként kapott tranzisztort megpróbálja összekapcsolni. Ez akkor lehet sikertelen, ha bármelyik tranzisztornak van már párja.
- **+int dropOut():** Ellenőrzi hogy van-e a hallgatónak védelme, és ezt visszatérési értékben jelzi a szobának. 0: nincs védelem, 1: TVSZ (Book), 2: sor (Beer), 3: rongy (Cloth).
- **+void pickUpItem(Item i):** A karakter megpróbálja felvenni a paraméterként kapott tárgyat, annyiban különbözik a tanár azonos nevű függvényétől, hogy a hallgatónak meg van engedve a logarléc felvétele.

8.1.3 Teacher

- **Felelősség**

A gép/controller által irányított oktatókat reprezentáló osztály. Mindenre képes amire egy karakter is. Ezen felül, ha bekerül egy szobába, megpróbálja kibuktatni az ott lévő hallgatókat. A logarlécen kívül képes felvenni a tárgyakat. Akkor nyernek az oktatók, ha lejár az idő vagy az összes hallgató kibukott.



- **Ósosztályok**

Character → Student

- **Interfészek**

-

- **Attribútumok**

- - **int clothed**: Azt tartja számon, hogy még meddig van rongy hatása alatt az oktató.

- **Metódusok**

- +**int getClothed()**: Visszaadja a clothed attribútum értékét.
- +**int setClothed(int i)**: Beállítja a clothed attribútum értékét.
- +**void pickUpItem(Item i)**: Megpróbálja felvenni a paraméterként kapott tárgyat, a hallgató azonos nevű metódusától annyiban különbözik, hogy ha a tárgy logarléc, akkor nem veszi fel.
- +**void teacherDuty()**: Megpróbálja kibuktatni az összes hallgatót aki a szobában tartózkodik.

8.1.4 Cleaner

- **Felelősség**

A gép/controller által irányított takarítót reprezentáló osztály. „Butább” az oktatónál, csak kevés specifikus dologra képes.

- **Ősosztályok**

Character → Student

- **Interfészek**

-

- **Attribútumok**

-

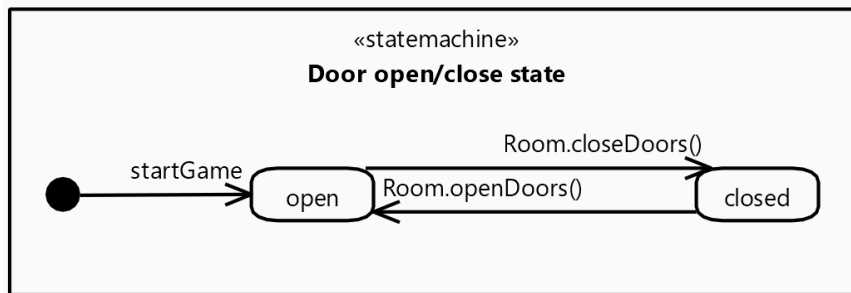
- **Metódusok**

- + **void setCurrentRoom(Room r)**: A takarító legfontosabb metódusa, amellelt hogy beállítja a szobát amiben jelenleg tartózkodik, kiszellőzteti azt (az a szoba nem lesz gázos) és kitessekel minden más karaktert a szobából.

8.1.5 Door

- **Felelősség**

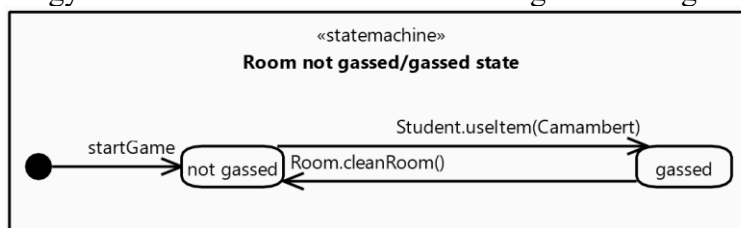
Az ajtót reprezentáló osztály, amely lehetővé teszi a karakterek számára, hogy szobák között tudjanak közlekedni. Mágikus szobák ajtóit véletlen időközönként bezáródnak majd újra kinyílnak.

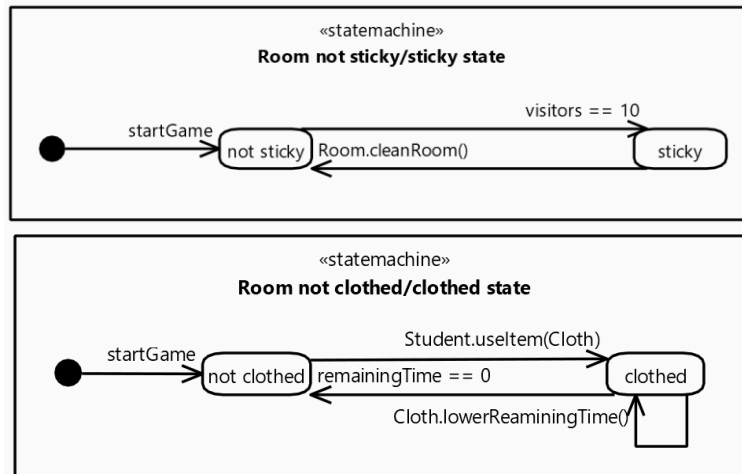


- **Össztályok**
 -
- **Interfészek**
 -
- **Attribútumok**
 - - **bool isOneWay**: Meghatározza, hogy egy ajtót csak az egyik irányba lehet használni vagy mindkét irányba. Ha az értéke igaz akkor egyirányú, ha pedig hamis akkor kétirányú.
 - - **bool isClosed**: Meghatározza, hogy egy ajtó csukva van vagy sem. Csukott ajtót nem tudnak a karakterek használni. Ha az értéke igaz akkor csukva van, ha pedig hamis akkor nyitva van.
 - - **List<Room> rooms**: Meghatározza, hogy melyik két szobát köti össze az ajtó.
- **Metódusok**
 - + **void changeRoom(Character c, Room r)**: A megadott c karaktert átlépteti az r szobából annak a párjába. Amennyiben ez sikeres akkor beállítja a c karakternek az új szobát, törli a c karaktert az r szobából és hozzáadja a c karaktert az új szobához.
 - + **void replaceRoom(Room x, Room y)**: Kicseréli az x szobát y szobára, azaz megváltoztatja, hogy melyik két szobát köti össze az ajtó.
 - + **void setConnectedRooms(Room existing, Room _new)**: Beállítja, hogy az ajtó melyik két szobát kösse össze. Az ajtó existing és a _new szobát fogja összekötni.

8.1.6 Room

- **Felelősség**
A szobát reprezentáló osztály, amely tárolja a benne található karaktereket, ajtókat, illetve tárgyakat. A szobát be lehet gázosítani, illetve ezt meg lehet szüntetni. A szobát egy időre rongyossá lehet tenni és adott számú látogató után ragacsos lesz.





- **Össosztályok**
 -
- **Interfészek**
 -
- **Attribútumok**
 - - **bool gassedRoom**: Meghatározza, hogy egy szoba gázos vagy sem. Ha értéke igaz akkor gázos, ha pedig hamis akkor nem gázos.
 - - **int clothedRoom**: Meghatározza, hogy egy szoba rongyosságából mennyi idő van hátra. Ha az értéke 0 akkor a szoba nem rongyos.
 - - **int visitors**: Meghatározza, hogy a szoba tisztítása óta mennyi karaktert járt a szobában. Takarítás után az értéke 0.
 - - **int maxCapacity**: Meghatározza, hogy egy szobában egyszerre maximum mennyi karakter tartózkodhat. Ha ez az érték megegyezik a szobában lévő karakterek számával akkor a szoba tele van, nem léphet be új karakter.
 - - **bool magical**: Meghatározza, hogy egy szoba mágikus vagy sem. Ha értéke igaz akkor mágikus, ha pedig hamis akkor nem. Mágikus szobának az ajtóit véletlen időközönként bezáródnak.
 - - **bool sticky**: Meghatározza, hogy egy szoba ragacsos vagy sem. Ha értéke igaz akkor ragacsos, ha hamis akkor nem. Ragacsos szobában nem tudnak a karakterek tárgyakat felvenni.
 - - **List<Door> doors**: Meghatározza, hogy melyik ajtók vannak a szobában. Ebből fogjuk tudni megkapni a szomszédos szobákat.
 - - **List<Item> items**: Meghatározza a szobában lévő tárgyakat.
 - - **List<Character> characters**: Meghatározza a szobában lévő karaktereket. Ennek a méretének nem szabad nagyobb lennie, mint a szoba kapacitása.
- **Metódusok**
 - + **int getMaxCapacity()**: Visszaadja a maxCapacity értékét, azaz azt, hogy mekkora a szoba befogadó képessége.
 - + **int getclothedRoom()**: Visszaadja a clothedRoom értékét, azaz azt, hogy ha a szoba rongyos akkor még mennyi ideig.
 - + **bool getgassedRoom()**: Visszaadja a gassedRoom értékét, azaz azt, hogy a szoba gázos vagy sem.

- + **void setgassedRoom(bool b)**: A paraméterül kapott értékre beállítja a gassedRoom értékét.
- + **List<Character> getCharacters()**: Visszaadja a characters listát, azaz visszaadja a szobában tartózkodó karaktereket egy listában.
- + **List<Item> getItems()**: Visszaadja az items listát, azaz visszaadja a szobában lévő tárgyak listáját.
- + **List<Door> getDoors()**: Visszaadja a doors listát, azaz a szobában lévő ajtók listáját.
- + **void removeItem(Item i)**: Törli az items listából a paraméterül kapott Item-et amennyiben az benne volt a listában.
- + **void addItem(Item i)**: Hozzáadja az items listához a paraméterül kapott Item-et amennyiben annak az értéke nem null.
- + **bool characterEnters(Character c)**: Megnézi, hogy a szobába befér egy új karakter vagy sem, azaz a szoba kapacitása nagyobb-e mint a szobában tartózkodó karakterek száma. Ha befér true-val tér vissza, ha nem akkor false-al.
- + **void addCharacter(Character c)**: Hozzáadja a characters listához a paraméterül kapott Character-t amennyiben annak az értéke nem null. Ha a szoba gázos akkor meghívja a karakter megfelelő függvényét, illetve, ha van a karakternek valami egyéb kötelessége akkor azokat.
- + **void removeCharacter(Character c)**: Törli a characters listából a paraméterül kapott Character-t amennyiben az benne volt a listában.
- + **void addDoor(Door d)**: Hozzáadja az doors listához a paraméterül kapott Door-t amennyiben annak az értéke nem null.
- + **void removeDoor(Door d)**: Törli a doors listából a paraméterül kapott Door-t amennyiben az benne volt a listában.
- + **void split()**: Egy szobát ketté választ, azaz létrehoz egy új szobát aminek a kapacitása és állapotai(gassedRoom, clothedRoom, visitors, magical, sticky) megegyeznek az eredeti szoba állapotaival. Az eredeti szoba ajtóiból minden második az új szobának lesz az ajtaja. A tárgyak és a karakterek maradnak az eredeti szobában.
- + **void dropThemOut()**: Megnézi, hogy a szobában lévő diákok meg tudják védeni magukat vagy sem. Ha ronggyal védekezik valaki akkor a szoba clothedRoom attribútuma nő. Akik nem tudták magukat megvédeni azokat törli a characters listából és eldobatja velük a náluk lévő tárgyakat.
- + **void clothThem()**: A szobában lévő oktatók clothed állapotát beállítja a szoba clothedRoom értékére.
- + **void mergeWithRoom(Room r)**: A szobát összeolvasztja a paraméterül kapott szobával úgy, hogy az int típusú attribútumokból a nagyobbat veszi alapul, a bool típusúakból pedig ha valamelyik igaz akkor azt. Az összeolvasztást csak akkor hajtja végre, ha a két szoba minden karaktere belefér az összevont szobába. Az összes karakter, tárgy és ajtó hozzáadódik a szoba megfelelő listájához.
- + **void closeDoors()**: Bezárja a szoba összes ajtaját, azaz az összes ajtó isClosed attribútumát true-ra állítja.
- + **void openDoors()**: Kinyitja a szoba összes ajtaját, azaz az összes ajtó isClosed attribútumát false-ra állítja.
- + **void setSticky(bool b)**: A paraméterül kapott értékre beállítja a sticky értékét.
- + **bool getSticky()**: Visszaadja a sticky értékét, azaz azt, hogy a szoba ragacsos vagy sem.

- + **void cleanRoom()**: Minden olyan karaktert aki nem bénult/ájult kidob a szobából, beállítja a visitors értékét 0-ra, illetve a gassedRoom és a sticky értékét pedig false-ra állítja.
- + **void increaseVisitors()**: A szoba visitors attribútumát növeli eggyel. Ez akkor fog bekövetkezni, ha egy karakter belép a szobába.

8.1.7 Item

- **Felelősség**
A tárgyak egységes kezelése ebben az absztrakt osztályban valósul meg.
- **Interfészek**
-
- **Ősosztályok**
-
- **Attribútumok**
 - **heldBy**: Tudnia kell, hogy melyik karakternél van, láthatóság: #, Character
 - **containedBy**: Tudnia kell, melyik szobában található, láthatóság: #, Room
- **Metódusok**
 - **boolean transfer(Character c, Room r)**: Beállítja a heldBy attribútumot a paraméterbe kapott c-re, beállítja a containedBy attribútumot a paraméterben kapott r-re. Falseal tér vissza, láthatóság: +
 - **void updateRoom(Room r)**: Beállítja a containedBy attribútumot a paraméterben kapott r-re., láthatóság: +
 - **int saveMe()**: 0-val tér vissza, (nem védi meg hallgatót a tanártól), láthatóság: +
 - **boolean protectMe()**: False-al tér vissza, (nem véd a gázos szobától), láthatóság: +
 - **Use()**: Függvény definíció, nem tud semmit.
 - **LowerRemainingTime()**: Absztrakt függvény, a leszármazó osztályok maguk implementálják a működést.

8.1.8 AirFreshener

- **Felelősség**
Egy olyan tárgy, amely használatával a használt gázos szobában a szoba elveszti gázos tulajdonságát.
- **Ősosztályok**
Item
- **Interfészek**
 - Useable
- **Attribútumok**
-

- **Metódusok**

- **void use():**

- Ha containedBy != null, akkor azt a szobát, amiben a légfrissítő van, annak a gázos tulajdonsága legyen hamis
 - egyébként. akkor az a karakter, akinél van a légfrissítő, a szoba, ahol ez a karakter van, annak a gázos tulajdonsága legyen hamis

, láthatóság: +

8.1.9 Beer

- **Felelősség**

Egy olyan tárgy, amely képes meg védeni, azt a hallgatót, akinél van egy tanártól, aki a hallgatót ki akarja buktatni.

- **Ősosztályok**

Item

- **Interfészek**

-

- **Attribútumok**

- **remainingUse:** Megadja, hogy hányszor lehet még felhasználni, a hallgató védésére, láthatóság: -, int

- **Metódusok**

- **void lowerRemainingUse():**

- Ha a remainingUse nagyobb, mint 0, akkor csökkentse eggyel a remainingUse értékét és használójának egyik tárgyát véletlenszerűen kiválasztja, amit eldobat a használóval (beleértve a sört saját magát is eldobhatja).
 - Egyébként, akkor az a karakter, akinél van a TVSZ, az a karakter távolítsa el a TVSZ-t.

, láthatóság: +

- **int saveMe():** Térjen vissza 1-gyel (Megvédi a hallgatót a tanártól), láthatóság: +

8.1.10 Book

- **Felelősség**

Egy olyan tárgy, amely képes meg védeni, azt a hallgatót, akinél van egy tanártól, aki a hallgatót ki akarja buktatni.

- **Ősosztályok**

Item

- **Interfészek**

-

- **Attribútumok**

- **remainingUse:** Megadja, hogy hányszor lehet még felhasználni, a hallgató védésére, láthatóság: -, int

- **fake**: Azt jelöli, hogy a tárgy hamis verziója-e, láthatóság: -, boolean
- **Metódusok**
 - **void lowerRemainingUse()**:
 - Ha hamis TVSZ térjen vissza
 - Ha a remainingUse nagyobb, mint 0, akkor csökkentse eggyel a remainingUse értékét.
 - Egyébként, akkor az a karakter, akinél van a TVSZ, az a karakter távolítsa el a TVSZ-t.
 - , láthatóság: +
 - **int saveMe()**: Térjen vissza 1-gyel (Megvédi a hallgatót a tanártól), láthatóság: +

8.1.11 Camembert

- **Felelősség**
Egy olyan tárgy, amely használatával a használt szobában a szoba gázos lesz.
- **Ősosztályok**
Item
- **Interfészek**
○ Useable
- **Attribútumok**
-
- **Metódusok**
 - **void use()**:
 - Ha containedBy != null, akkor azt a szobát, amiben a légfrissítő van, annak a gázos tulajdonsága legyen igaz
 - egyébként. akkor az a karakter, akinél van a légfrissítő, a szoba, ahol ez a karakter van, annak a gázos tulajdonsága legyen igaz, láthatóság: +

8.1.12 Cloth

- **Felelősség**
Egy olyan tárgy, amely képes meg védeni, a hallgatókat azáltal, hogy azt a szobát, ahol használták, egy ideig a benne tartózkodó tanárok nem képesek kibuktatni hallgatókat.
- **Ősosztályok**
Item
- **Interfészek**
-
- **Attribútumok**
 - **remainingTime**: Meghatározza, hogy a védelemből mennyi idő van még hátra a védelméből, láthatósága: -, int

- **Metódusok**

- **int saveMe():** Térjen vissza 3-mal (Megvédi a hallgatót a tanártól), láthatóság: +

8.1.13 Mask

- **Felelősség**

Egy olyan tárgy, amely megvédi a használóját a gázos szoba káros hatásaitól.

- **Ősosztályok**

Item

- **Interfészek**

-

- **Attribútumok**

- **remainingTime:** Meghatározza, hogy a védelemből mennyi idő van még hátra, láthatósága: -, int
- **useTimes:** Meghatározza, hogy hányszor volt használva, láthatósága: -, int
- **fake:** Azt jelöli, hogy a tárgy hamis verziója-e, láthatóság: -, boolean

- **Metódusok**

- **void increaseUsedTimes():**
 - Ha hamis maszk térjen vissza
 - A usedTimes értékét növelje eggyel.
, láthatósága: +
- **void lowerRemainingTime():**
 - Ha hamis maszk térjen vissza
 - Ha a remainingTime nagyobb, mint 0, akkor csökkentse eggyel a remainingTime értékét.
 - Egyébként, akkor az a karakter, akinél van a Maszk, az a karakter távolítsa el a TVSZ-t.
, láthatóság: +
- **boolean protectMe():** Térjen vissza igaz értékkel, láthatósága: +

8.1.14 SlidingRuler

- **Felelősség**

Egy olyan tárgy, amit, ha egy hallgató felvesz, akkor a hallgatók nyerik a játékot.

- **Ősosztályok**

Item

- **Interfészek**

-

- **Attribútumok**

- **fake:** Azt jelöli, hogy a tárgy hamis verziója-e, láthatóság: -, boolean

- **Metódusok**
 - **boolean transfer(Character c, Room r):** Beállítja a heldBy attribútumot a paraméterbe kapott c-re, beállítja a containedBy attribútumot a paraméterben kapott r-re. True-val tér vissza, láthatóság: +

8.1.15 Transistor

- **Felelősség**
Egy olyan tárgy, amellyel a hallgatók egyik szobából a másikba képesek teleportálni.
- **Össztályok**
Item
- **Interfészek**
 - Useable
- **Attribútumok**
 - **pair:** Megadja, hogy mi a párja a tranzisztornak
- **Metódusok**
 - **Transistor getPair():** Térjen vissza a pair-rel, láthatósága: +
 - **void setPair(Transistor):** Beállítja a pair-t a paraméterben kapott tranzisztorra, láthatósága: +
 - **Room getRoom():** Térjen vissza, azzal, hogy melyik szobában van a tranzisztor, láthatósága: +
 - **void use():**
 - Ha az a karakter, akinél van a tranzisztor képes a tranzisztor párjának a szobájába lépni, akkor:
 - dobja el a tranzisztort a felhasználója
 - felhasználónak a szobáját állítsuk a pár szobáját
 - pár szobájába lépjen a felhasználó
 - az eredeti szobából töröljük ki a felhasználó
 - az eredeti szobába tegyük a tranzisztort
 - a tranzisztoron állítsuk be, hogy az eredeti szobába legyen

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

8.2.1 Hallgató összekapcsol két tranzisztort

- **Leírás**
Egy szobában van egy hallgató és két tranzisztor van nála, ezeket összekapcsolja.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
A tranzisztorok összekapcsolása
- **Bemenet**
NewRoom - - 10 - - 0
NewStudent – r0
NewItem 1 – r0 s0
NewItem 1 – r0 s0
Connect i0 i1

- **Elvárt kimenet**

r0 created
s0 created in r0
i0 created in s0 r0
i1 created in s0 r0
i0 i1 connected

8.2.2 Földön lévő tranzisztorhoz teleportálás

- **Leírás**

Van két szoba, az egyikben a hallgató, akinél van egy tranzisztor, ami össze van kapcsolva egy másik tranzisztorral. A másik tranzisztor a másik szobában a földön van. A hallgató bekapcsolja a nála lévő tranzisztor és eldobja a földre. A hallgató a másik szobába kerül és az eredeti szobában lévő tranzisztor kikapcsol.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A Földön lévő tranzisztorhoz teleportálás

- **Bemenet**

NewRoom - - 10 - - 0
NewRoom - - 10 - - 0
NewDoor r0 r1
NewStudent – r0
NewItem 1 – r0 s0
NewItem 1 – r0 s0
Connect i0 i1
Drop s0 i0
Enter s0 d01
Use s0 i1

- **Elvárt kimenet**

r0 created
r1 created
d01 created
s0 created in r0
i0 created in s0 r0
i1 created in s0 r0
i0 i1 connected
i0 added to r0
s0 entered r1
s0 used i1
s0 entered r0

8.2.3 Másik hallgatónál/tanárnál lévő tranzisztorhoz teleportálás

- **Leírás**

Van két szoba, az egyikben a hallgató, akinél van egy tranzisztor, ami össze van kapcsolva egy másik karakternél lévő tranzisztorral. A másik karakter a másik szobában van. A hallgató bekapcsolja a nála lévő tranzisztor és eldobja a földre. A hallgató a másik szobába kerül és az eredeti szobában lévő tranzisztor kikapcsol.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Egy hallgatónál/tanárnál lévő tranzisztorhoz teleportálás ellenőrzése

- **Bemenet**

NewRoom - - 10 - - 0

NewRoom - - 10 - - 0
 NewDoor r0 r1
 NewStudent – r0
 NewStudent – r0
 NewItem 1 – r0 s0
 NewItem 1 – r0 s0
 Connect i0 i1
 Drop s0 i0
 PickUp s1 i0
 Enter s0 d01
 Use s0 i1

- **Elvárt kimenet**

r0 created
 r1 created
 d01 created
 s0 created in r0
 s1 created in r0
 i0 created in s0 r0
 i1 created in s0 r0
 i0 i1 connected
 i0 added to r0
 i0 added to s1
 s0 entered r1
 s0 used i1
 s0 entered r0

8.2.4 Tárgy eldobása

- **Leírás**

Egy szobában van egy hallgató/tanár nála egy tárgy. A hallgató/tanár eldobja a nála lévő tárgyat. A tárgy a földön lesz.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A tárgy eldobás funkció ellenőrzése

- **Bemenet**

NewRoom - - 10 - - 0
 NewStudent – r0
 NewItem 1 – r0 s0
 Drop s0 i0

- **Elvárt kimenet**

r0 created
 s0 created in r0
 i0 created in s0 r0
 i0 added to r0

8.2.5 Tárgy sikeres felvétele

- **Leírás**

Egy szobában van egy hallgató/oktató, egy tárgy, ami a szobában a földön van. A hallgató/tanár megpróbálja felvenni tárgyat. A tárgy sikeresen átkerül a hallgatóhoz/tanárhoz.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tárgyfelvétel

- **Bemenet**

NewRoom - - 10 - - 0

NewStudent – r0

NewItem 1 – r0

Pickup s0 i0

- **Elvárt kimenet**

r0 created

s0 created in r0

i0 created in r0

i0 added to s0

8.2.6 Tárgy sikertelen felvétele 1.

- **Leírás**

Egy szobában van egy hallgató/oktató, egy tárgy, ami a szobában a földön van. A hallgató/tanár megpróbálja felvenni tárgyat. De nem sikerül venni, mert már a hallgatónál/tanárnál van 5 tárgy.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tárgyfelvétel

- **Bemenet**

NewRoom - - 10 - - 0

NewStudent – r0

NewItem 1 – r0 s0

NewItem 2 – r0 s0

NewItem 3 – r0 s0

NewItem 4 – r0 s0

NewItem 5 – r0 s0

NewItem 6 – r0

PickUp s0 i5

- **Elvárt kimenet**

r0 created

s0 created in r0

i0 created in s0 r0

i1 created in s0 r0

i2 created in s0 r0

i3 created in s0 r0

i4 created in s0 r0

i5 created in r0

s0 cant pick up i5

8.2.7 Tárgy sikertelen felvétele 2.

- **Leírás**

Egy ragacsos szobában van egy hallgató/oktató, egy tárgy, ami a szobában a földön van. A hallgató/tanár megpróbálja felvenni tárgyat. De nem sikerül venni, mert a szoba ragacsos.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tárgyfelvétel

- **Bemenet**

NewRoom - - 10 - + 5

NewStudent – r0

NewItem 1 – r0

PickUp s0 i0

- **Elvárt kimenet**

r0 created

s0 created in r0

i0 created in r0

s0 cant pick up i0

8.2.8 Szoba szétválása

- **Leírás**

Van egy gázos szoba, benne két hallgató és a földön két tárgy. A szoba szétválík. A két új szoba között van egy ajtó, mindkettő új szoba gázos. A szobákban van 1-1 hallgató és 1-1 tárgy.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szoba szétválása

- **Bemenet**

NewRoom + - 10 - - 0

NewStudent – r0

NewStudent – r0

NewItem 1 – r0

NewItem 1 – r0

Split r0

- **Elvárt kimenet**

r0 created

s0 created in r0

s1 created in r0

i0 created in r0

i1 created in r0

r0 split into r0.0 and r0.1

8.2.9 Szoba sikeres összeolvadása

- **Leírás**

Van két szoba, egyik szoba gázos és 3 kapacitású és van benne egy tárgy, a másik szoba ragacsos és 4 kapacitású és van benne egy karakter. A két szoba szomszédos. A két szoba megpróbál összeolvadni. Az új szoba gázos és ragacsos és 4 kapacitású. A szobában van a karakter és a tárgy.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szobák összeolvadása

- **Bemenet**

NewRoom + - 3 - - 0

NewRoom - - 4 - + 5

NewDoor r0 r1 - -

NewStudent – r1

NewItem 1 – r0

Merge r0 r1

- **Elvárt kimenet**

r0 created

r1 created

d01 created
 s0 created in r1
 i0 created in r0
 r0 and r1 merged into r0

8.2.10 Szoba sikertelen összeolvadása

- **Leírás**

Van két szoba, 1 kapacitású és van benne egy karakter, a másik szoba 1 kapacitású és van benne egy karakter. A két szoba szomszédos. A két szoba megpróbál összeolvadni, de nem tudnak, kapacitás hiány miatt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szobák összeolvadása

- **Bemenet**

NewRoom - - 1 - - 0
 NewRoom - - 1 - - 0
 NewDoor r0 r1 - -
 NewStudent – r0
 NewStudent – r1
 Merge r0 r1

- **Elvárt kimenet**

r0 created
 r1 created
 d01 created
 s0 created in r0
 s1 created in r1
 r0 and r1 failed to merge

8.2.11 Sikeres szobaváltás

- **Leírás**

Van két szoba, melyek szomszédosok. Az egyik szobában van egy karakter. A karakter át akar lépni a másik szobába. Ez sikeresen megtörténik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szobaváltás

- **Bemenet**

NewRoom - - 1 - - 0
 NewRoom - - 1 - - 0
 NewDoor r0 r1 - -
 NewStudent – r0
 Enter s0 d01

- **Elvárt kimenet**

r0 created
 r1 created
 d01 created
 s0 created in r0
 s0 entered r1

8.2.12 Sikertelen szobaváltás

- **Leírás**

Van két szoba, melyek szomszédosok, mindkettő kapacitása 1. Mindkét szobában van 1-1 karakter. Az egyik szobában lévő karakter megpróbál átlépni a másik szobába. Ez nem sikerül, mert nincs a másik szobának elég kapacitása.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Szobaváltás

- **Bemenet**

NewRoom - - 1 - - 0

NewRoom - - 1 - - 0

NewDoor r0 r1 - -

NewStudent – r0

NewStudent – r1

Enter s0 d01

- **Elvárt kimenet**

r0 created

r1 created

d01 created

s0 created in r0

s1 created in r1

s0 failed to enter r1

8.2.13 Camembert használata

- **Leírás**

Van egy szoba benne egy hallgató, nála egy Camembert. A hallgató használja a camembert és a szoba gázos lesz.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Camembert

- **Bemenet**

NewRoom - - 1 - - 0

NewStudent – r0

NewItem 4 r0 s0

Use s0 i0

- **Elvárt kimenet**

r0 created

s0 created in r0

i0 created in s0 r0

s0 gassed r0

8.2.14 Mágikus szoba viselkedése

- **Leírás**

Van egy mágikus szoba és egy sima szoba, a két szoba szomszédos. A mágikus szobában van egy karakter. A mágikus szoba eltünteti az ajtajait. Így a karakter nem képes a másik szobába átlépni.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Mágikus szoba

- **Bemenet**

NewRoom - - 1 + - 0

NewRoom - - 1 - - 0

NewDoor r0 r1 - +

NewStudent – r0

Enter s0 d01

- **Elvárt kimenet**

r0 created
r1 created
d01 created
s0 created in r0
s0 failed to enter r1

8.2.15 Légyfrissítő használata

- **Leírás**

Van egy gázos szoba, benne egy hallgató, akinél van egy maszk és egy légyfrissítő. A hallgató használja a légyfrissítőt. A szoba elveszti a gázos tulajdonságát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Légyfrissítő

- **Bemenet**

NewRoom + - 1 - - 0
NewStudent – r0
NewItem 7 r0 s0
Use s0 i0

- **Elvárt kimenet**

r0 created
s0 created in r0
i0 created in s0 r0
s0 degassed r0

8.2.16 Kibukás

- **Leírás**

Van két szomszédos szoba, az egyikben egy hallgató, a másikban egy tanár. A hallgató átlép a másik szobába. A hallgató kibukik, amely során a hallgató a földre dobja a nála lévő tárgyat.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Kibukás

- **Bemenet**

NewRoom - - 10 - - 0
NewRoom - - 10 - - 0
NewDoor r0 r1 - -
NewStudent – r0
NewTeacher - - r1
Enter s0 d01

- **Elvárt kimenet**

r0 created
r1 created
d01 created
s0 created in r0
t0 created in r1
s0 entered r1
s0 failed

8.2.17 Rongy használata

- **Leírás**

Van két szomszédos szoba, az egyikben egy hallgató nála egy törlő rongy, a másikban egy tanár. A hallgató átlép a másik szobába. A hallgató felhasználja a rongyot így egy ideig a szobában lévő tanár egy ideig nem tudja kibuktatni a hallgatót.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Rongy

- **Bemenet**

NewRoom - - 10 - - 0

NewRoom - - 10 - - 0

NewDoor r0 r1 - -

NewStudent – r0

NewTeacher - - r1

NewItem 2 r0 s0

Enter s0 d01

- **Elvárt kimenet**

r0 created

r1 created

d01 created

s0 created in r0

t0 created in r1

i0 created in r0 s0

s0 entered r1

s0 used i0

8.2.18 TVSZ használata

- **Leírás**

Van két szomszédos szoba, az egyikben egy hallgató nála egy tvsz, a másikban egy tanár. A hallgató átlép a másik szobába. A hallgató felhasználja a tvsz-t így a szobában lévő tanár nem tudja kibuktatni a hallgatót.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

TVSZ

- **Bemenet**

NewRoom - - 10 - - 0

NewRoom - - 10 - - 0

NewDoor r0 r1 - -

NewStudent – r0

NewTeacher - - r1

NewItem 5 r0 s0

Enter s0 d01

- **Elvárt kimenet**

r0 created

r1 created

d01 created

s0 created in r0

t0 created in r1

i0 created in r0 s0

s0 entered r1

s0 used i0

8.2.19 Hamis tárgy használata

- **Leírás**

Van két szomszédos szoba, az egyikben egy hallgató nála egy hamis tvsz, a másikban egy tanár. A hallgató átlép a másik szobába. A hallgató felhasználja a hamis tvsz-t ami nem védi meg.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hamis tárgy (TVSZ)

- **Bemenet**

NewRoom - - 10 - - 0

NewRoom - - 10 - - 0

NewDoor r0 r1 - -

NewStudent – r0

NewTeacher - - r1

NewItem 5 r0 s0 +

Enter s0 d01

- **Elvárt kimenet**

r0 created

r1 created

d01 created

s0 created in r0

t0 created in r1

i0 created in r0 s0

s0 entered r1

s0 used i0

s0 failed

8.2.20 Sör használata

- **Leírás**

Van két szomszédos szoba, az egyikben egy hallgató nála egy sör és egy tvsz, a másikban egy tanár. A hallgató átlép a másik szobába. A hallgató felhasználja a sört, ami megvédi, de elejti a tvsz-t.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Sör

- **Bemenet**

NewRoom - - 10 - - 0

NewRoom - - 10 - - 0

NewDoor r0 r1 - -

NewStudent – r0

NewTeacher - - r1

NewItem 6 r0 s0

NewItem 5 r0 s0

Enter s0 d01

- **Elvárt kimenet**

r0 created

r1 created

d01 created

s0 created in r0

t0 created in r1

i0 created in r0 s0

i1 created in r0 s0
 s0 entered r1
 s0 used i0
 i1 added to r1

8.2.21 Az összes hallgató megbukott

- **Leírás**

Van két szoba, egyikben két hallgató másikban egy tanár. A két szoba szomszéd. A tanár átlép a hallgatóval teli szobába. A két hallgató kibukik. hallgatók elvesztik a játékot.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Játék végének tesztelése

- **Bemenet**

NewRoom - - 3 - -
 NewRoom - - 1 - -
 NewDoor r0 r1 - -
 NewStudent - r0
 NewStudent - r0
 NewTeacher - - r1
 Enter t0 d01

- **Elvárt kimenet**

r0 created
 r1 created
 d01 created
 s0 created in r0
 s1 created in r0
 t0 created in r1
 t0 entered r0
 s0 failed
 s1 failed

8.2.22 Idő lejárt

- **Leírás**

Van egy hallgató egy szobában. Letelik a rendelkezésre álló játék idő. A játékot elvesztik a hallgatók.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Játék végének tesztelése

- **Bemenet**

NewRoom - - 1 - -
 NewStudent - r0
 SetTime 0

- **Elvárt kimenet**

r0 created
 s0 created in r0
 Game over

8.2.23 Hamis logarlécet felveszi egy hallgató

- **Leírás**

Van egy szoba, benne egy hallgató és a földön egy logarléc. A hallgató felveszi a logarlécet. A hamis logarléc a hallgatóhoz kerül.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hamis logarléc használhatatlanságának ellenőrzése

- **Bemenet**
NewRoom - - 1 - - -
NewStudent - r0
NewItem 8 + r0 –
PickUp s0 i0
- **Elvárt kimenet**
r0 created
s0 created
i0 created
s0 picked up i0

8.2.24 Logarlécet felveszi egy hallgató

- **Leírás**

Van egy szoba, benne egy hallgató és a földön egy logarléc. A hallgató felveszi a logarlécet. A játékot megnyerik a hallgatók.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Játék végének tesztelése

- **Bemenet**
NewRoom - - 1 - - -
NewStudent r0 –
NewItem 8 – r0 –
PickUp s0 i0
- **Elvárt kimenet**
r0 created
s0 created in r0
i0 created in r0
s0 picked up i0
YOU WIN

8.2.25 Ragacsos szoba keletkezése

- **Leírás**

Van egy kitakarított szoba és már csak egy karakternek kell belépnie, hogy ragacsos legyen és egy karakter be is lép. A szoba ragacsossá válik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A ragacsossá válás tesztelése.

- **Bemenet**
NewRoom - - 1 - - -
NewRoom - - 1 - - 4
NewDoor r0 r1 - -
NewStudent - r0
Enter s0 d01
- **Elvárt kimenet**
r0 created
r1 created
d01 created
s0 created in r0
s0 entered r1

r1 is now sticky

8.2.26 Gázos szoba szellőztetése

- **Leírás**

Van két szoba, egyik gázos. A nem gázos szobában van egy takarító. A takarító átlép a gázos szobába. A gázos szoba elveszti a gázos tulajdonságát.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Takarító szellőztetési képességének tesztelése

- **Bemenet**

NewRoom - - 1 - - -
NewRoom + - 1 - - -
NewDoor r0 r1 - -
NewCleaner - r0
Enter c0 d01

- **Elvárt kimenet**

r0 created
r1 created
d01 created
c0 created
c0 entered r1
r1 is now clean

8.2.27 Takarító kitessékeltetési képessége

- **Leírás**

Van két szoba egy sima és egy gázos. A gázos szobában van egy hallgató, aki a maszkja hatása alatt van. A sima szobában van egy takarító. A takarító átlép a gázos szobába. A takarító kitessékeli a hallgatót. A hallgató átkerül a sima szobába.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Takarító kitessékeltetési képességének tesztelése

- **Bemenet**

NewRoom - - 2 - - -
NewRoom + - 2 - - -
NewDoor r0 r1 - -
NewStudent r1 -
NewItem 3 - s0
NewCleaner - r0
Enter c0 d01

- **Elvárt kimenet**

r0 created
r1 created
d01 created
s0 created in r1
i0 created in s0
c0 created in r0
c0 entered r1
s0 entered r0
r1 is now clean

8.2.28 Hamis maszk nem létező védelme gázos szoba ellen

- **Leírás**

Van két szoba egy sima és egy gázos. Van egy hallgató a sima szobában, a hallgatónál egy hamis maszk. A hallgató átlép a másik szobába. A hallgató elkábul, ezért a földre dobja nála lévő hamis maszkot

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Maszk használhatatlanságának ellenőrzése

- **Bemenet**

NewRoom - - 1 - - -
NewRoom + - 1 - - -
NewDoor r0 r1 - -
NewStudent r1 -
NewItem 3 + s0 -
Enter s0 d01

- **Elvárt kimenet**

r0 created
r1 created
d01 created
s0 created in r1
i0 created in s0
s0 entered r0
i0 is added to r0

8.2.29 Maszk védelme gázos szoba ellen

- **Leírás**

Van két szoba egy sima és egy gázos. Van egy hallgató a sima szobában, a hallgatónál egy maszk. A hallgató átlép a másik szobába. A hallgató felveszi a maszkot és így nem éri a gázos szoba káros hatása.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Maszk használatának ellenőrzése

- **Bemenet**

NewRoom - - 1 - - -
NewRoom + - 1 - - -
NewDoor r0 r1 - -
NewStudent r0 -
NewItem 3 - s0 -
Enter s0 d01

- **Elvárt kimenet**

r0 created
r1 created
d01 created
s0 created in r0
i0 created in s0
s0 entered r1

8.2.30 Gázos szoba hatása a hallgatón/tanáron

- **Leírás**

Van két szoba egy sima és egy gázos. Van egy hallgató a sima szobában, a hallgatónál egy TVSZ és egy söröskorsó. A hallgató átlép a gázos szobába. A hallgató elkábul, ezért a földre dobja nála lévő söröskorsót és TVSZ-t.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Gázos szoba hatásának tesztelése

- **Bemenet**

NewRoom - - 1 - - -
 NewRoom + - 1 - - -
 NewDoor r0 r1 - -
 NewStudent - r0
 NewItem 5 - s0 -
 NewItem 6 - s0 -
 Enter s0 d01

- **Elvárt kimenet**

r0 created
 r1 created
 d01 created
 s0 created in r0
 i0 created in s0
 i1 created in s0
 s0 entered r1
 i0 added to r1
 i1 added to r1

8.3 A tesztelést támogató programok tervei

A program tesztelése úgy történik, hogy a program teszt eseteket olvas be (az előző pontban specifikált bemeneti nyelv szerint), ezután a programot futtatva végrehajtja az adott teszt esetet és ad egy kimenetet. Minden teszt esethez tartozik egy elvárt kimenet (az előző pontban specifikált kimeneti nyelv szerint). A kapott kimenetet összehasonlítjuk az elvárt kimenettel és ha ez a kettő nem egyezik meg akkor a teszt lefutását sikertelennek tekintjük. Ha viszont a kapott kimenet és az elvárt kimenet megegyezik akkor a tesztet sikeresnek tekintjük. Az összehasonlítást kézzel végezzük, mint felhasználó/tesztelő személy.

8.4 Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.04.10. 12:00	2 óra	Havasi Hőgye	Konzultáció
2024.04.14. 18:00	2 óra	Kaposvári	Karakter, hallgató, oktató, takarító osztályok részletes leírása
2024.04.14. 13:00	2,5 óra	Soós	8.1-ben a Door és Room osztályok leírása + 8.3
2024.04.14 18:00	3 óra	Havasi	Interface javítása, 8.2 első fele
2024.04.14 19:00	1,5 óra	Reszegi	8.2 második fele
2024.04.14 20:00	2,5	Hőgye	8.1-ben az Item, Book, Beer, Mask, SlidingRuler, Cloth, AirFreshener, camembert és Tarnsistor osztályok