

3. Analízis modell kidolgozása

7 – bajaok

Konzulens:

Dobos-Kovács Mihály

Csapattagok

Tóth Gergő	IZN26S	tothgergo001@gmail.com
Bereczki Máté	M975OK	mate.bereczki0129@gmail.com
Schweitzer András Attila	TLEIB5	schweitzeraa16@gmail.com
Lenner András Tibor	L6KOH4	lenner.andras@edu.bme.hu
Hubai Rajmund Szilveszter	CQKFI2	hubairajmund0507@gmail.com

2024.03.04.

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Szoba

Ezek az objektumok alkotják a labirintust, a játékkeret. Ezen tudnak mozogni a hallgatók és az oktatók, és különböző interakciókat tudnak végrehajtani. A szobáknak van kapacitása, ami meghatározza, hogy hány karakter (hallgató/oktató) tartózkodhat a szobában. Ajtókat tartalmaznak, amelyeken keresztül közlekedhetnek a játékosok a szomszédos szobákba. Emellett a szobákban tárgyak vannak elhelyezve, amelyeket a hallgatók felvehetnek. Néhány szoba el lehet gázosítva. Az elgázosított tulajdonság a hallgatóra és oktatóra is hatással van.

3.1.2 Elátkozott szoba

Olyan szoba, amelynek bizonyos ajtajai eltűnhetnek, illetve megjelenhetnek, ezáltal az átjárást ellehetetlenítik.

3.1.3 Labirintus

Felelős a szobák közötti interakciók kezeléséért, mint szobák közötti összeolvadás és szétválás.. A labirintus felel azért, hogy számon tartsa az élő hallgatókat és jelezze a játéknak ha mind meghalt.

3.1.4 Hallgató

Felvehetnek, lerakhatnak és használhatnak tárgyakat, amelyek segítenek a túlélésben és az oktatók elleni védekezésben. Korlátozott számú tárggyal (maximum 5) rendelkeznek, és két tranzisztort tudnak egyszerre birtokolni (amik szintén tárgynak számítanak). Emellett a hallgató felel azért, hogy az életciklussal rendelkező/többször használatos tárgyait ellenőrizze/értesítse. Ha a szoba értesíti hogy egy oktatóval van egy szobában, akkor ellenőrzi hogy van-e védelme, és amennyiben nincs, akkor jelzi a halálát a labirintusnak. Ellenőrzi, hogy van-e nála Logarléc, ha igen, jelzi ezt a játéknak. Ha szobát vált a hallgató, akkor megkérdezi, hogy gázos-e, amennyiben az, akkor megbénul és eldobja tárgyait.

3.1.5 Oktató

Képesek felvenni tárgyakat (bizonyos valószínűséggel), a Logarlécet leszámítva, de nem használhatják azokat. Szobák közötti mozgásuk véletlenszerű, bizonyos időközönként mozognak. Ha szobát vált az oktató, akkor megkérdezi, hogy gázos-e, vagy van-e aktivált nedves táblatörő rongy a szobában, amennyiben valamelyik igaz, akkor megbénul és eldobja tárgyait.

3.1.6 Logarléc

Ez az objektum a hallgatók számára mindig látható, bármely szobában tartózkodnak.

3.1.7 TVSZ denevérbőrre nyomtatva

Három alkalommal védi meg a hallgatót az oktatótól- ezáltal nem hal meg a hallgató. Ezután eltűnik.

3.1.8 Szent Söröspoharak

Védőeszköz, amely aktiválása után bizonyos időkorlát lejártáig védelmet nyújt az oktatók ellen. Használat után eltűnik a játékos eszköztárából.

3.1.9 Nedves táblatörlő rongy

Támadó eszköz, amely a hallgató által aktiválva (ezzel automatikusan lerakva) bizonyos ideig megbénítja a szobájában lévő oktatókat. Az idő lejártá után a rongy kiszárad és eltűnik.

3.1.10 Dobozolt káposztás camembert

Támadó eszköz, ami elgázosítja a szobát használatkor. Használat után a camembert eltűnik a játékos eszköztárából.

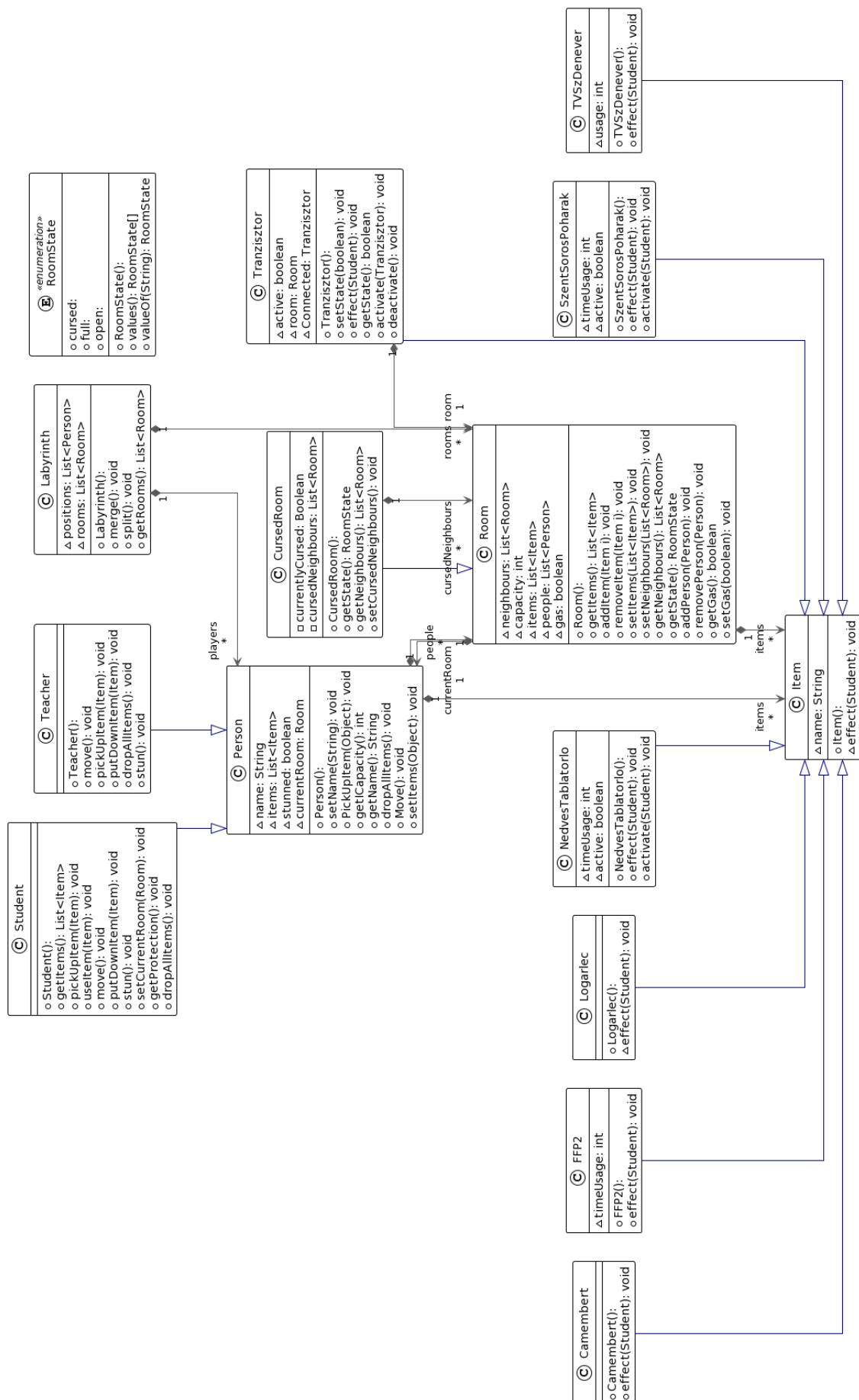
3.1.11 FFP2 Maszk

Védőeszköz, amely megvéd a gázosított szobák hatásaitól. Csak adott ideig nyújt védelmet, élettartam lejártá után eltűnik a játékos eszköztárából.

3.1.12 Tranzisztor

Tárgy, amelyből ha 2 van egy hallgatónál, akkor összekapcsolódnak. A kapcsolódott tranzisztorok lehetővé teszik a szobák közötti teleportálást a következő módon: Az egyik tranzisztort a hallgató lerakja (ez ekkor nem felvehető), majd a másikat bárhol lerakva az első tranzisztor szobájába teleportál. A teleportálás után tranzisztorok szétkapcsolódnak, és újra felvehetőek lesznek. Egy játékos egyszerre csak két tranzisztorral rendelkezhet. A tranzisztor felelőssége áthelyezni a hallgatót a párosított tranzisztor szobájába.

3.2 Statikus struktúra diagramok



3.3 Osztályok leírása

3.3.1 Room

- **Felelősség**

A Room osztály felelős a játéktér egységeit alkotó szobák kezeléséért. Tartalmazza a szobához kapcsolódó információkat, többi szobához vezető “ajtókat”, tárgyakat, hallgatókat és oktatókat.

- **Attribútumok**

- **capacity (int):** Hány karakter (hallgató/oktató) tartózkodhat a szobában.
- **neighbours(List<Room>):** Tárolja, mely szobák a szomszédai, merre tudnak menni a karakterek (ajtók).
- **items (List<Item>):** Tárolja, milyen tárgyak találhatók a szobában.
- **gas (bool):** A szoba elgázosított állapotát jelzi.
- **people (List<Person>):** Tárolja a benne tartózkodó karaktereket (hallgató/oktató).

- **Metódusok**

- Getter és setter metódusok
 - **public List<Item> getItems();**
 - **public void addItem(Item i); public void removeItem(Item i);**
 - **public List<Room> getNeighbours(); public void setNeighbours();**
 - **public bool getGas(); public void setGas(bool b);**
 - **public void addPerson(Person p); public void removePerson(Person p);**
- **public RoomState getState():** Visszaadja egy szoba állapotát, lehet ez open vagy full.

3.3.2 CursedRoom

- **Felelősség**

A CursedRoom osztály valósítja meg az elátkozott szobák eltérő viselkedését.

- **Ősosztályok**

Room

- **Attribútumok**

- **currentlyCursed (bool):** Eltárolja, hogy éppen el van-e átkozva a szoba, tehát, hogy minden ajtaja használható-e.
- **cursedNeighbours(List<Room>):** Eltárolja, hogy azon “ajtókat”, tehát szomszédokat, akikhez éppen nem lehet átmenni.

- **Metódusok**

- **public RoomState getState():** Visszaadja egy szoba állapotát, lehet ez open vagy full, vagy cursed.
- **public void curse():** Módosítja az elérhető ajtókat.
- Getter és setter metódusok

- **public List<Room> getNeighbours()**
- **public void setCursedNeighbours()**

3.3.3 Student

- **Felelősség**

A Student osztály reprezentálja a játékban irányított karaktereket. A hallgatók felvehetnek, lerakhatnak és használhatnak tárgyakat, és a céljuk a Logarléc megszerzése.

- **Össztályok**

Person

- **Attribútumok**

- **Name (string):** A hallgató nevét tárolja.
- **items (List<Item>):** A hallgatónál lévő tárgyakat tárolja.
- **currentRoom(Room):** A hallgató jelenlegi szobáját tárolja el.
- **stunned(bool):** Eltárolja, ha éppen meg van bénulva.
- **protected(bool):** Van az oktató ellen használható védelme.

- **Metódusok**

- **public void pickUpItem(Item i):** A hallgató kezébe vesz egy tárgyat.
- **public void putDownItem(Item i):** A hallgató lerak a kezéből egy tárgyat.
- **public void move():** A hallgató a szobák közötti átjárását teszi lehetővé.
- **public stun():** Megbénítja a hallgatót.
- **public void useItem(Item i):** A tárgyak használatát megvalósító metódus.
- Getter és setter metódusok
 - **public void setCurrentRoom(Room r);**
 - **public List <Item> getItems();**
 - **public getProtection();**
- **public void dropAllItems():** Megbénuláskor elejti az összes tárgyat.

3.3.4 Teacher

- **Felelősség**

A Teacher osztály reprezentálja a játékban irányítatlan karaktereket, azaz az oktatókat. Az oktatók felvehetnek bizonyos tárgyakat, de nem használhatják azokat.

- **Össztályok**

Person

- **Attribútumok**

- **Name (string):** Az oktató nevét tárolja.
- **items (List<Item>):** Az oktatónál lévő tárgyakat tárolja.
- **currentRoom(Room):** Az oktató jelenlegi szobáját tárolja el.
- **stunned(bool):** Eltárolja, ha éppen meg van bénulva.

- **Metódusok**

- **public void pickUpItem(Item i):** Az oktató kezébe vesz egy tárgyat.

- **public void putDownItem(Item i):** Az oktató lerak a kezéből egy tárgyat.
- **public void dropAllItems():** Megbénuláskor elejti az összes tárgyat.
- **public void move() :** A hallgató a szobák közötti átjárását teszi lehetővé.
- **public stun():** Megbénítja az oktatót.

3.3.5 Labyrinth

- **Felelősség**

Tárolja a szobákat, a játékban szereplő hallgatókat és oktatókat, valamint az összeolvasztott szobákat. Itt hajtódnak végre a szobák szétválása, illetve összeolvadása.

- **Attribútumok**

- **rooms(List<Room>):** Tárolja a játéktéren lévő szobákat.
- **positions(List<Person>):** Tárolja a hallgatókat és az oktatókat.

- **Metódusok**

- **public void Merge() :** Véletlenszerűen két szomszédos szobát összeolvaszt, és az új szoba megosztozik képességeiken és szomszédjaikon.
- **public void Split() :** Véletlenszerűen egy szobát kettéválaszt.
- Getter és setter metódusok
 - **public List <Room> getRooms();**

3.3.6 Logarlec

- **Felelősség**

Tárolja a szobákat, a játékban szereplő hallgatókat és oktatókat, valamint az összeolvasztott szobákat. Itt hajtódnak végre a szobák szétválása, illetve összeolvadása.

- **Ősosztályok**

Item

- **Attribútumok**

- **name (string):** A tárgy nevét tárolja.

- **Metódusok**

- **void effect(Student s) :** A tárggyal járó hatásokat megvalósító metódus. A tárgy felvétele után jelez a játéknak és nyernek a hallgatók. A hallgatót megkapja a függvény, hogy a megfelelő hatást ki tudja váltani.

3.3.7 TVSZDenever

- **Felelősség**

Ez az osztály valósítja meg a denevérbőrre nyomtatott TVSZ-t, ami 3-szor menti meg a hallgatót az oktatóktól.

- **Ősosztályok**

Item

- **Attribútumok**
 - **name (String):** A tárgy nevét tárolja.
 - **usage (int):** A tárgy számszori felhasználhatóságát tárolja.
- **Metódusok**
 - **void effect(Student s) :** A tárggyal járó hatásokat megvalósító metódus. Amint egy szobába kerül a hallgató egy oktatóval csökken egyet a használhatósága.

3.3.8 SzentSorosPoharak

- **Felelősség**

A Szent Söröspoharak osztály valósítja meg azt az oktatók ellen védekező tárgyat, amely egy adott ideig a hallgató kezében, majd lejárát után eltűnik onnan.

- **Ősosztályok**

Item

- **Attribútumok**
 - **name (String):** A tárgy nevét tárolja.
 - **timeUsage (int):** A tárgy használatának idejét tárolja.
 - **active(bool):** A tárgy használat alatt van.
- **Metódusok**
 - **void effect(Student s):** A tárggyal járó hatásokat megvalósító metódus. A hallgató kezében, ha beaktiválja immunitást élvez a tárgy idejének lejártáig.
 - **void activate(Student s):** Aktiválja az eszközt.

3.3.9 NedvesTablatorlo

- **Felelősség**

Ez az osztály valósítja meg a Nedves Táblatörlő rongy valósítja meg azt a támadó eszközt, amelyet, ha egy hallgató beaktivál, akkor az adott szobában egy ideig megbénítja az oktatókat.

- **Ősosztályok**

Item

- **Attribútumok**
 - **name (String):** A tárgy nevét tárolja.
 - **timeUsage (int):** A tárgy használatának idejét tárolja.
 - **active(bool):** A tárgy használat alatt van.
- **Metódusok**
 - **void effect(Student s):** A tárggyal járó hatásokat megvalósító metódus. Amint egy szobában aktiválja a hallgató, akkor a rongy kiszáradásáig (Az ideje lejártáig) a szobában lévő oktatókat megbénítja.
 - **void activate(Student s):** Aktiválja az eszközt.

3.3.10 Camambert

- **Felelősség**

Ez az osztály valósítja meg a Dobozos káposztás camembert tárgyat, amely használat után elgázosítja az adott szobát, amelyben felhasználták.

- **Ősosztályok**

Item

- **Attribútumok**

- **name (String):** A tárgy nevét tárolja.

- **Metódusok**

- **void effect(Student s):** A tárggyal járó hatásokat megvalósító metódus. Amint egy szobában aktiválja a hallgató, akkor egy adott idő után elgázosítja teljesen a szobát.

3.3.11 FFP2

- **Felelősség**

Az FFP2 Mask osztály valósítja meg azt, az FFP2 maszk tárgyat a játékban, amely a gázos szobák ellen immunitást ad.

- **Ősosztályok**

Item

- **Attribútumok**

- **name (String):** A tárgy nevét tárolja.
- **timeUsage(int):** A tárgy felhasználhatóságának idejét tárolja.

- **Metódusok**

- **void effect(Student s):** A tárggyal járó hatásokat megvalósító metódus. A hallgató immunitást élvez a gázok ellen és folyamatosan csökken az élettartama.

3.3.12 Tranzisztor

- **Felelősség**

Ez az osztály valósítja meg a Tranzisztor tárgyat a játékban, amelyekkel lehet teleportálni két szoba között, amik nem feltétlenül szomszédosak.

- **Ősosztályok**

Item

- **Attribútumok**

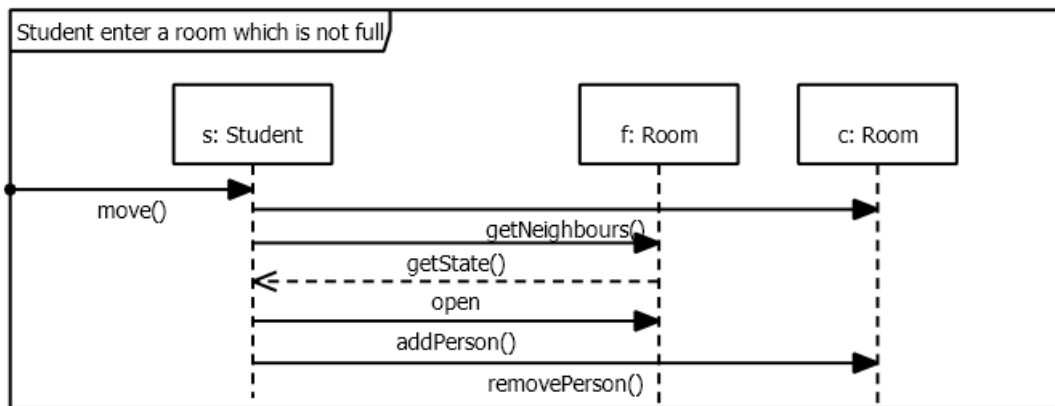
- **name (String):** A tárgy nevét tárolja.
- **active (bool):** a tranzisztor éppen használatban van-e, ekkor
- **connected (Tranzisztor):** A vele összekapcsolt és beaktivált tranzisztor helyét tárolja.
- **room(Room):** lerakott és aktivált tranzisztor szobája.

- **Metódusok**

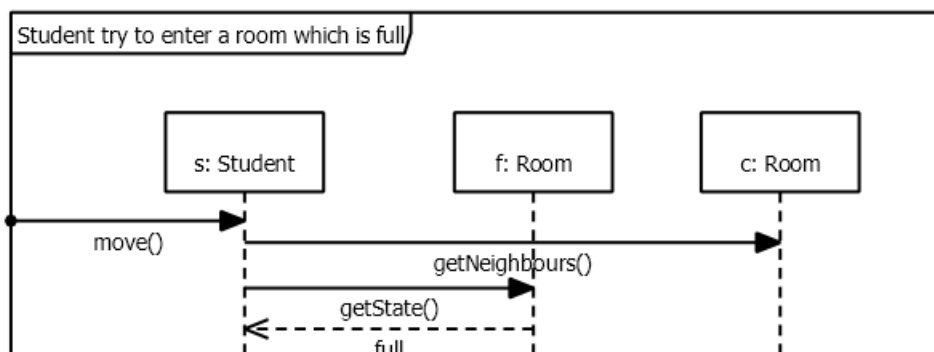
- **void effect(Student s):** Elteleportálja az egyik tranzisztortól a másikba a hallgatót, közben deaktiválja mind a kettőt és újra felvehetővé teszi.
- **void deactivate():** Ha meg akarunk szabadulni az aktivált tárgytól, anélkül, hogy használnánk.
- **void activate(Tranzisztor t):** Aktiválja és párosítja a kapott tranzisztort.
- **bool getState():** Visszaadja az adott tranzisztor állapotát.

3.4 Szekvencia diagramok

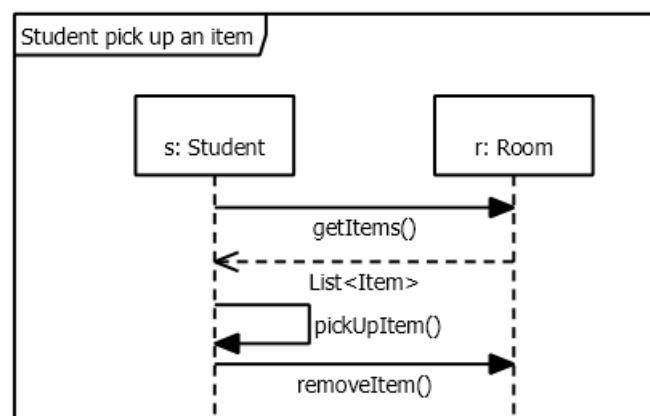
3.4.1 Student enter a room which is not full

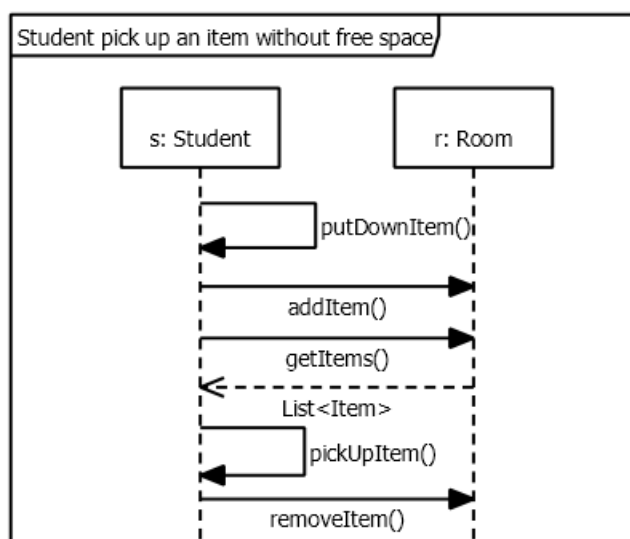
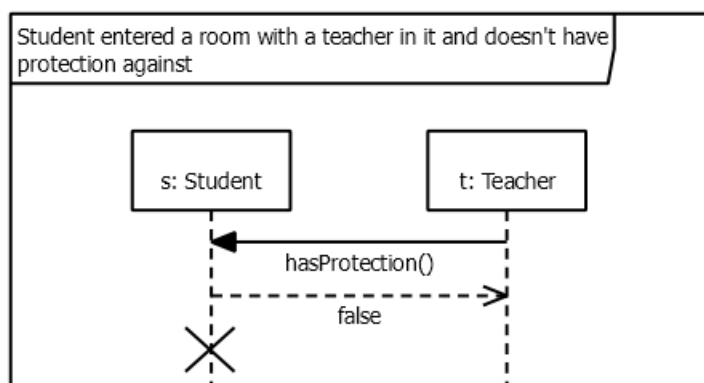


3.4.2 Student try to enter a room which is full

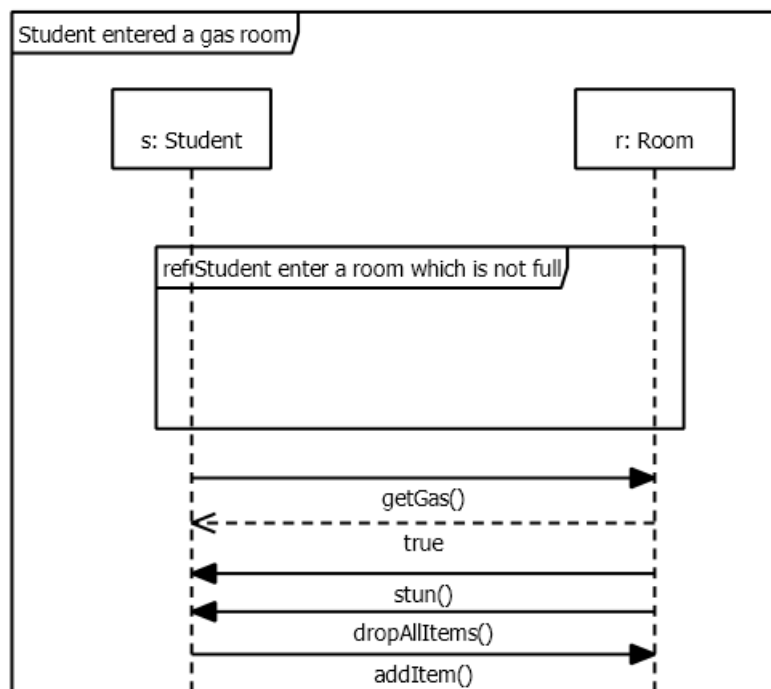


3.4.3 Student pick up an item

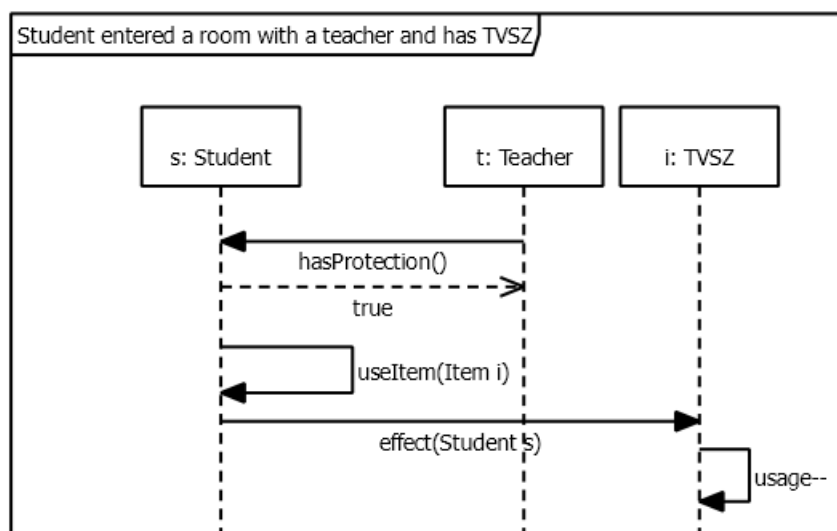


3.4.4 Student pick up an item without free space**3.4.5 Student entered a room with a teacher in it and doesn't have protection against**

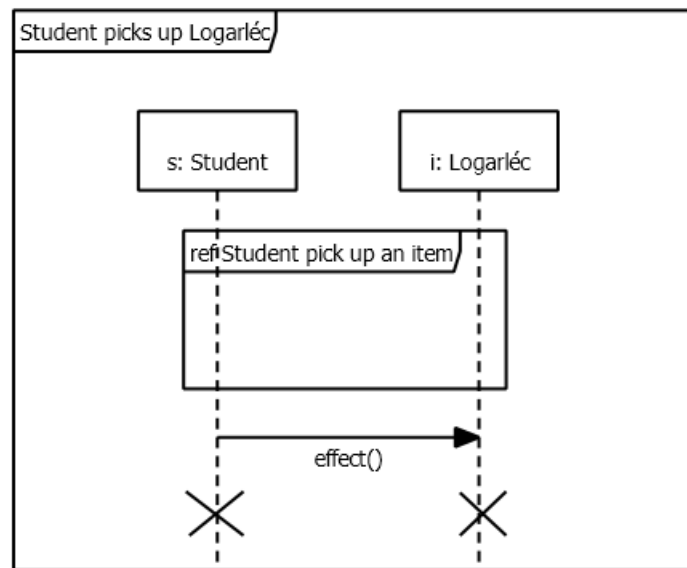
3.4.6 Student entered a gas room



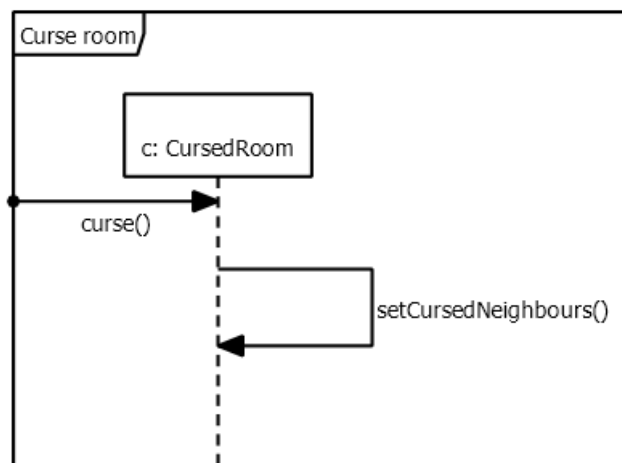
3.4.7 Student entered a room with a teacher and has TVSZ



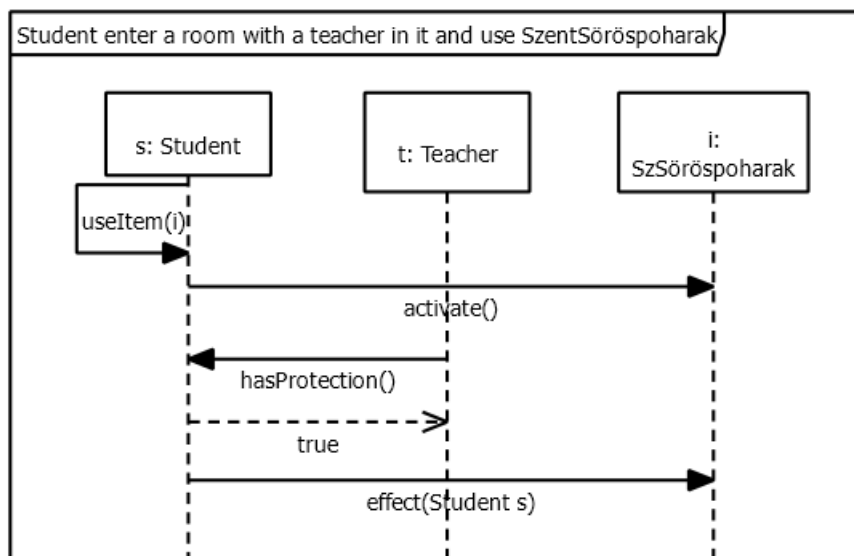
3.4.8 Student picks up Logarléc



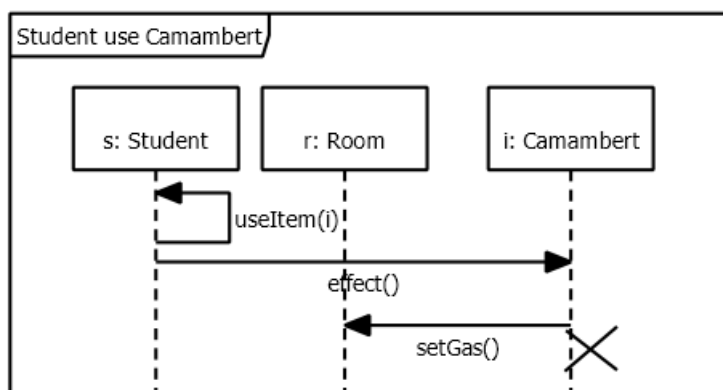
3.4.9 Curse Room



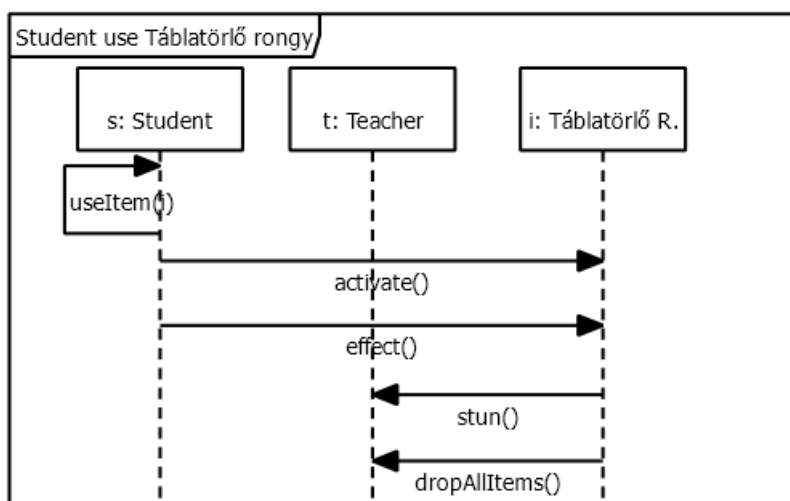
3.4.10 Student enter a room with a teacher in it and use SzentSöröspoharak



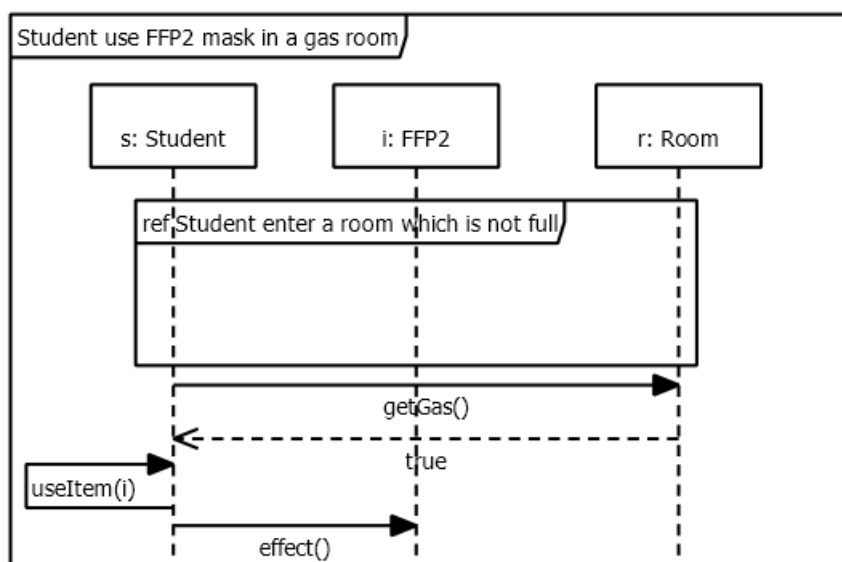
3.4.11 Student use Camambert

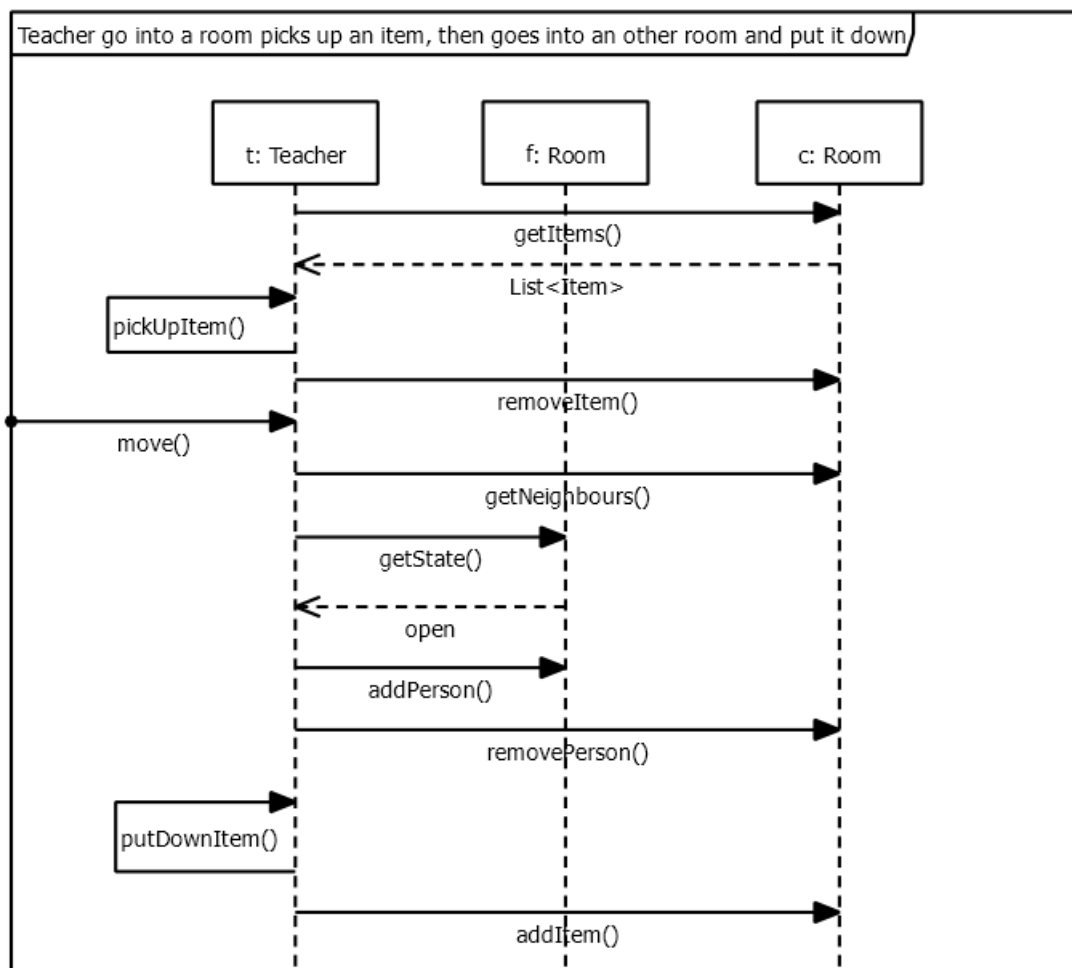


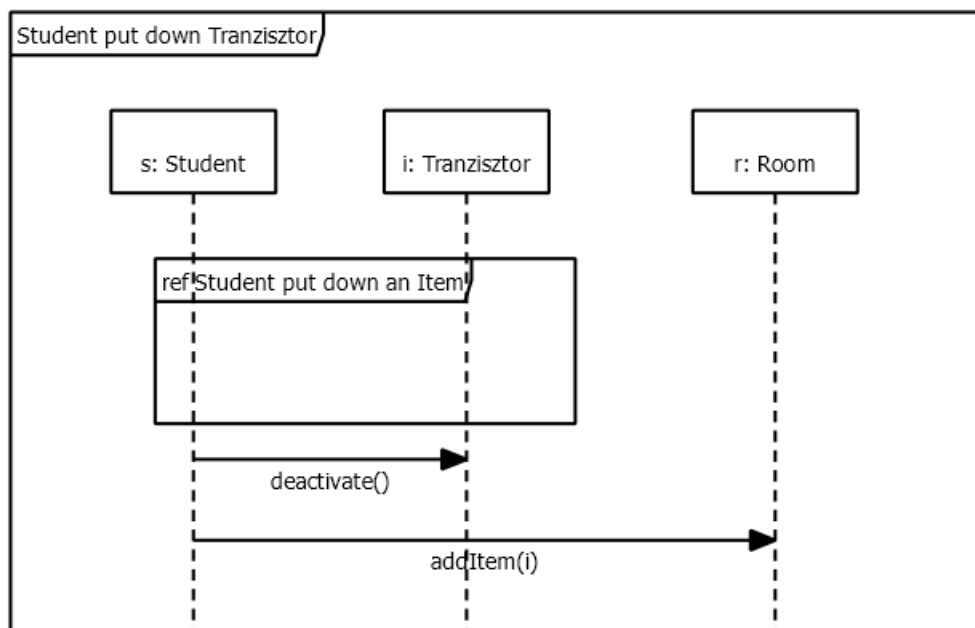
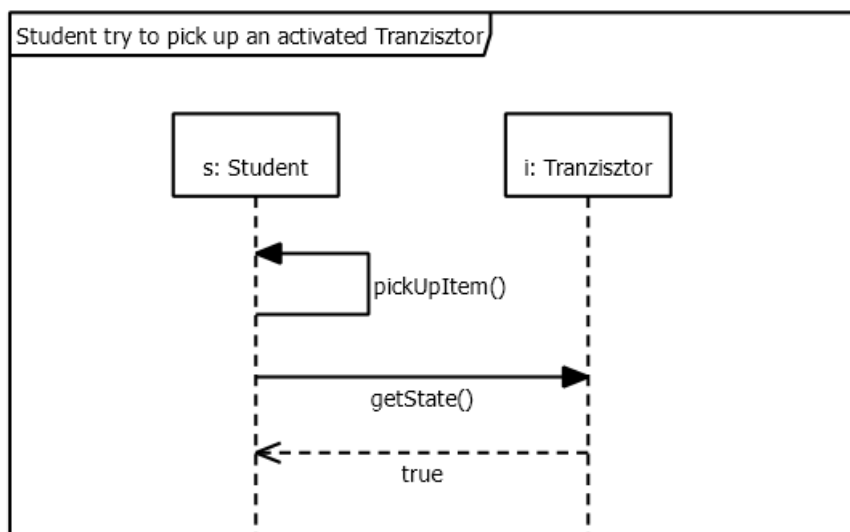
3.4.12 Student use Táblatörlő rongy

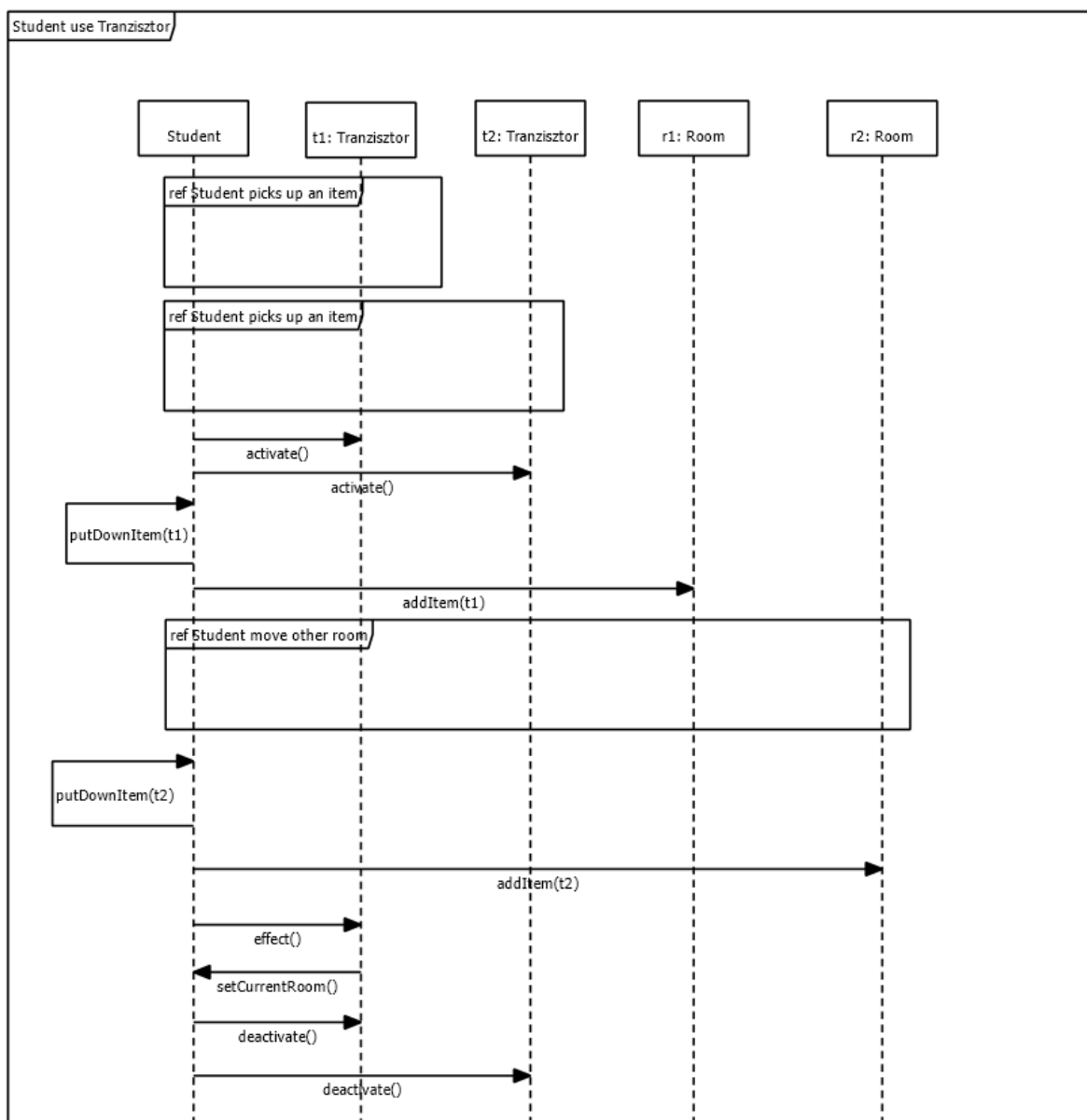


3.4.13 Student use FFP2 mask in a gas room

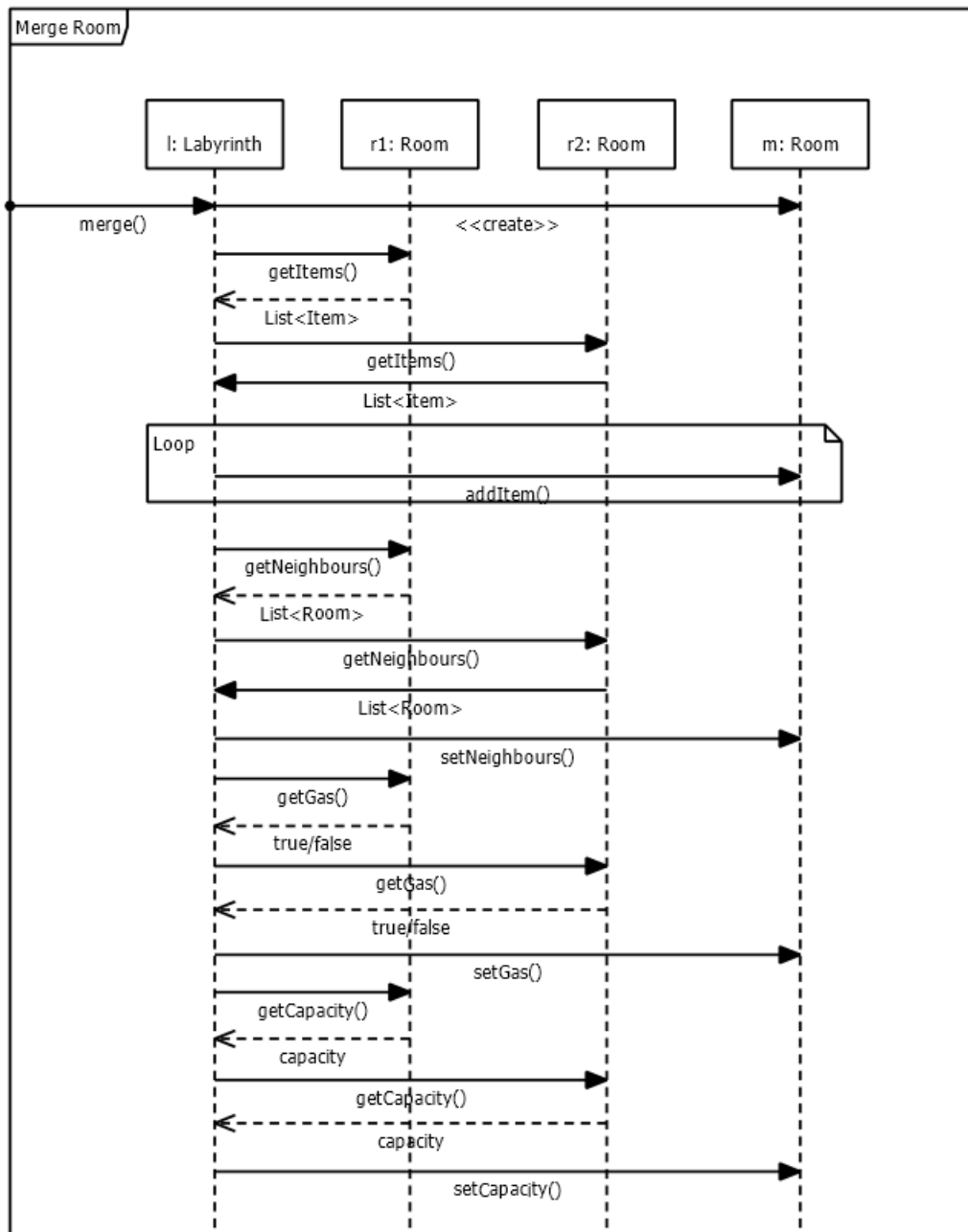


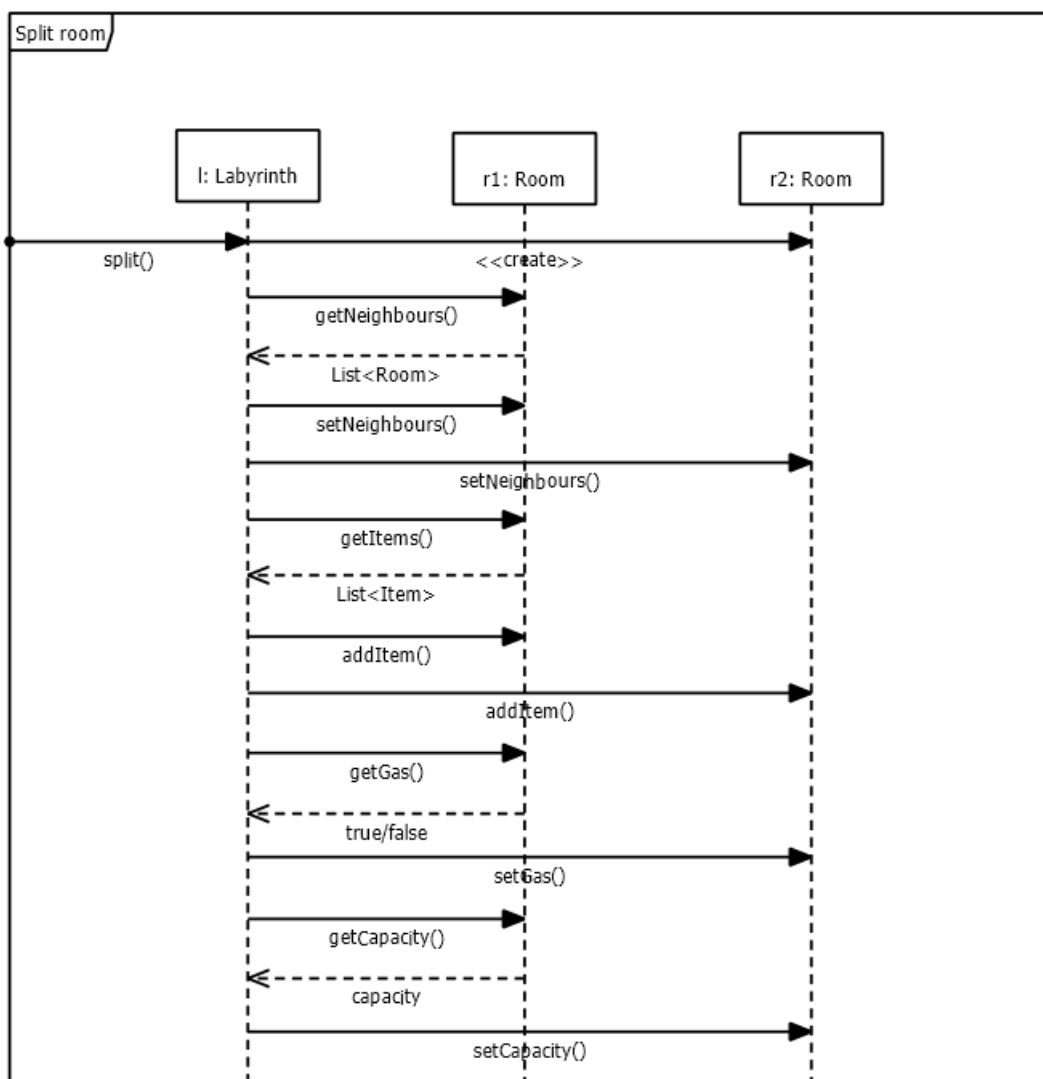
3.4.14 Teacher go into a room picks up an item, then goes into an other room and put it down

3.4.15 Student put down Tranzisztor**3.4.16 Student try to pick up an activated Tranzisztor**

3.4.17 Student use Tranzisztor

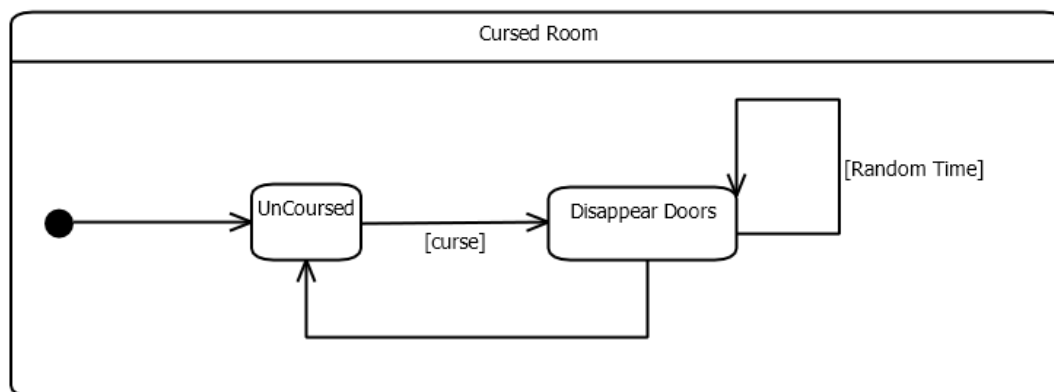
3.4.18 Merge room



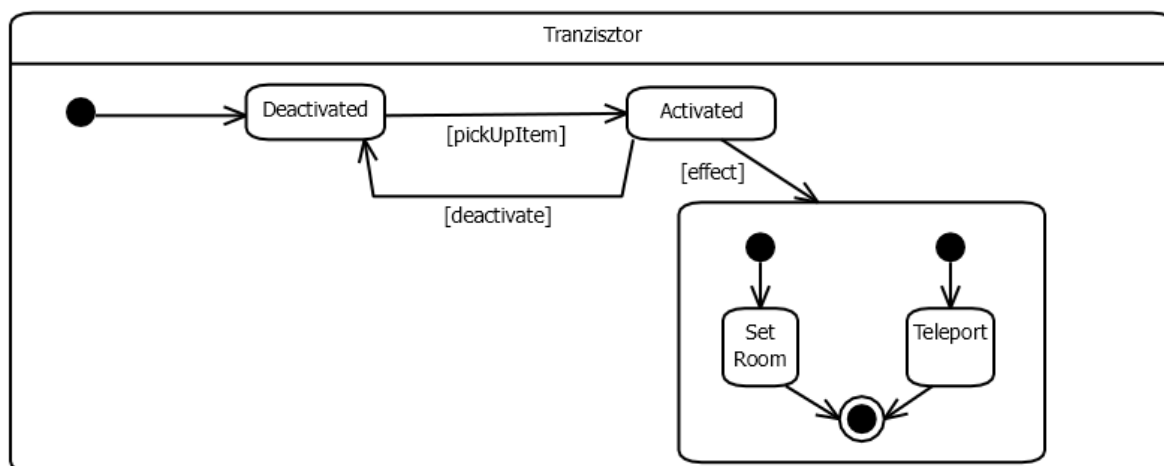
3.4.19 Split room

3.5 State-chartok

3.5.1 Egy CursedRoom élete



3.5.2 Egy Tranzisztor élete



Napló

Kezdet	Időtartam	Résztvevők	Leírás
2024.02.27. 18:00	2 óra	Bereczki	Objektum katalógus megírása és a dokumentum szerkesztése
2024.02.29. 13:00	1 óra	Bereczki Hubai Lenner Tóth Schweitzer	Értekezlet: Átbeszéltük a feladatok kiosztását. Döntés: Hubai, Bereczki, Lenner 29-e délután elkezdik kidolgozni a részletes tervet a komplexebb osztályokhoz. Tóth és Schweitzer péntek délután ellenőrzik ezeket és folytatják a maradék osztályok kidolgozását. 3-a délelőtt összegzünk és átnézünk.
2024.02.29. 15:00	3 óra	Bereczki Hubai Lenner	Objektum katalógus tervezése, kezdetleges osztály leírások átbeszélése
2024.03.01. 12:00	2 óra	Schweitzer Tóth	Objektum katalógus konzisztens kidolgozása
2024.03.01. 14:00	3 óra	Lenner Schweitzer	Objektum katalógus konzisztens kidolgozása
2024.03.02. 10:00	3,5 óra	Bereczki Hubai	Osztályok leírásának kidolgozása és strukturális diagramok tervezése és kidolgozása
2024.03.02. 15:00	4 óra	Hubai Lenner	Szekvencia diagramok és osztálydiagram kidolgozása
2024.03.03.	4,5 óra	Schweitzer	Osztálydiagram és osztályok leírásának végleges

11:00		Tóth	kidolgozása
2024.03.05. 00:00	2,5 óra	Tóth Bereczki	Szekvencia diagramok tervezése és készítése
2024.03.04. 9:30	4 óra	Tóth Bereczki	Szekvencia diagramok tervezése és készítése, state-chartok elkészítése, dokumentum formázása és befejezése