

3. Analízis modell I.

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

2024.03.03

3. Analízis modell kidolgozása

3.1 *Objektum katalógus*

3.1.1 Beer

A felvehető tárgyak egyik fajtája. Felvétel után 1 helyet foglal a tanuló tárhelyében. Tárolja, hogy még meddig biztosít védelmet oktatók ellen. Ez az idő az első felvételtől számol vissza. Ha lejár az idő a tárgy eltűnik, nem biztosít tovább védelmet. A tárgyat oktató is fel tudja venni, de ez esetben nem történik semmi különös. A tanár tárhelyében is 1 helyet foglal.

3.1.2 Book

A TVSZ denevérbőrre nyomtatott példányai (Továbbiakban csak TVSZ) egy tárgy, amit fel lehet venni. Ha egy hallgató birtokába kerül akkor a TVSZ 3 alkalommal megvédi az őt birtokló hallgatót az oktatóktól. A 3. alkalom után megsemmisül.

3.1.3 Camambert

Felvehető tárgy. Gázos szoba lesz abból a szobából, ahol használják. Egyszer használható, illetve használat után eltűnik a hallgató leltárából. Használat után nem felvehető. Csak hallgatók használhatják.

3.1.4 Cloth

Megbénítja a szobában tartózkodó oktatókat. Hatása adott ideig tart. Az ez idő alatt érkező új oktatókra is hatással van. Automatikusan aktiválódik, ha hallgató tárhelyében van és a hallgató találkozik oktatóval.

3.1.5 Door

A szobák közti átjárhatóságot biztosítja. Lehet egy irányból és két irányból bejárható.

3.1.6 Map

A labirintusban tárolódnak a szobák.

3.1.7 Mask

Felvehető tárgy. Használat után bizonyos ideig védettséget ad az eszmélet vesztéstől a gázos szobákban. Használat után eltűnik a hallgató leltárából. Csak hallgatók használhatják.

3.1.8 Room

A szoba tárolja a szobában tartózkodó hallgatókat és oktatókat. Ezen felül a szobában vannak felvehető tárgyak, amiket a hallgatók és az oktatók fel tudnak venni, egyet egyszerre egy valaki. A szobának van kapacitása, vagyis, hogy hány hallgató fér el benne. A szobának lehet fajtája: gázos és elátkozott.

3.1.9 SlidingRuler

Az egyetlen olyan tárgy, amit csak hallgató vehet fel, oktató nem. Az egész pályán csak 1 darab van belőle. Ahogy fel lett véve, a játék véget ér, a hallgatók nyernek.

3.1.10 Student

A hallgatót a játékos irányítja. A szobák között tud mozogni ajtók használatával vagy tranzistorok segítségével. A szobákban tárgyakat tud felvenni, illetve azokat tudja használni. A tárgyakat le is tudja rakni. A hallgató tárolja, hogy milyen tárgyak vannak nála (max 5). Ha védelmet biztosító tárgy nélkül találkozik egy oktatóval akkor meghal. Bármelyik tanuló megtalálja a logarléccet akkor a játéknak vége.

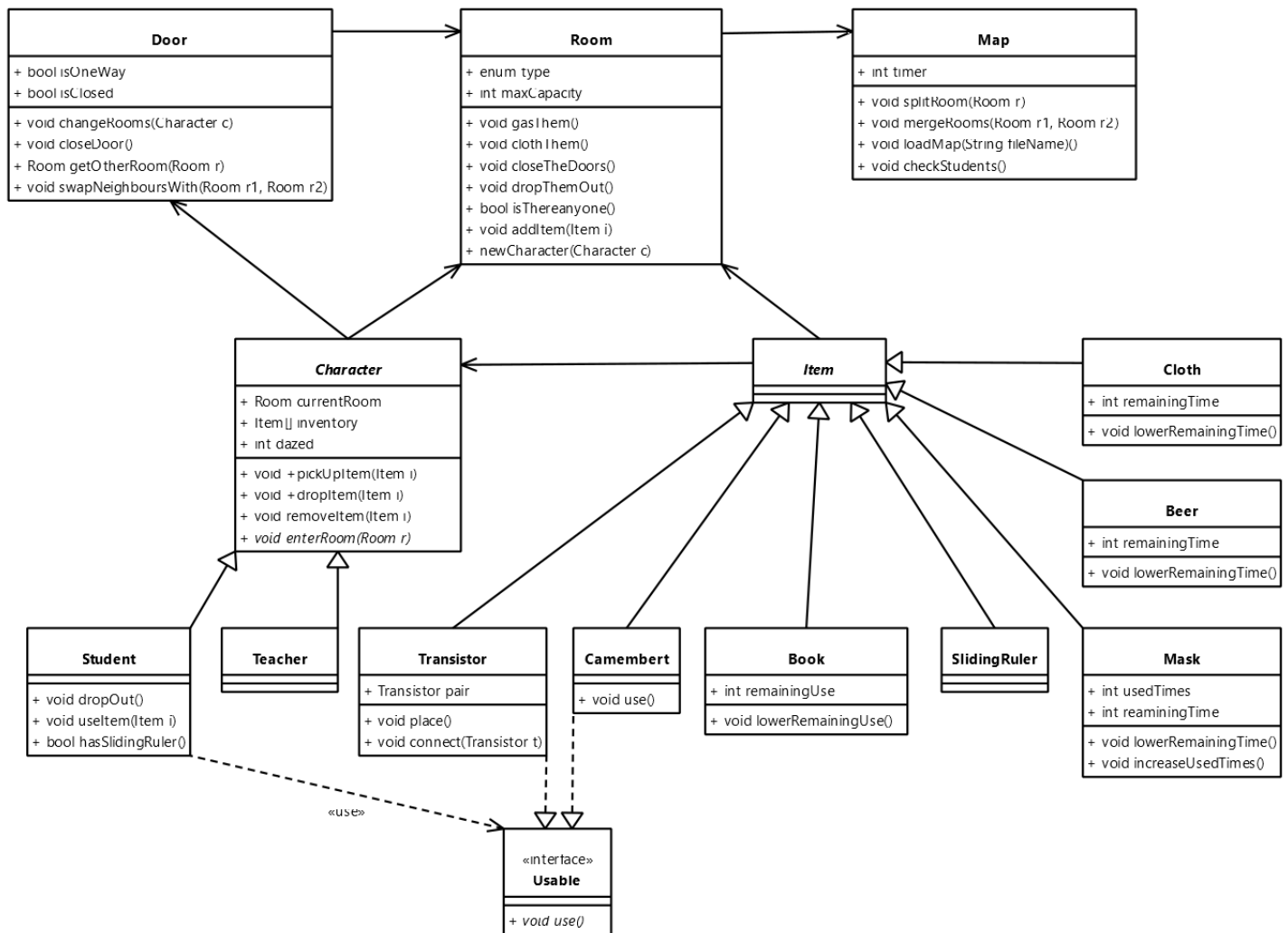
3.1.11 Teacher

Az oktató egy a játék által irányított karakter. Célja, hogy elvegye a hallgatók lelkét. Az oktató képes a szomszédos szobák között mozogni és a vele egy szobában lévő tárgyakat felvenni, azonban a felvett tárgyakat képtelen használni.

3.1.12 Transistor

Felvehető tárgy. Ha 2 van egy hallgatónál akkor összekapcsolhatók. A hallgatónál lévő fél megjegyzi, hogy a letett fél melyik szobában van. Összekapcsolt és bekapcsolt tranzistor nem vehető fel.

3.2 Statikus struktúra diagramok



3.3 Osztályok leírása

3.3.1 Beer

- **Felelősség**
 - A sört reprezentáló osztály.
- **Ősosztályok**
 - Item
- **Interfészek**
 -
- **Asszociációk**
 -
- **Attribútumok**
 - **int remainingTime**: A védelemből hátralévő idő.
- **Metódusok**
 - **void lowerRemainingTime()**: Csökkenti a hátralévő időt.

3.3.2 Book

- **Felelősség**
 - A TVSZ-t reprezentáló osztály megvalósítása.
- **Ősosztályok**
 - Item
- **Interfészek**
 -
- **Asszociációk**
 -
- **Attribútumok**
 - **int remainingUse**: Hány felhasználás maradt.
- **Metódusok**
 - **void lowerRemainingUse()**: Csökkenti a hátralévő felhasználások számát.

3.3.3 Camambert

- **Felelősség**
 - A dobozott camambertet reprezentáló osztály.

- **Össztályok**
 - Item
- **Interfészek**
 - Usable
- **Asszociációk**
 -
- **Attribútumok**
 -
- **Metódusok**
 - `void use()`: Elgázosítja a szobát.

3.3.4 Character

- **Felelősség**
 - A karaktereket megvalósító absztrakt osztály, (oktató és hallgató).
- **Össztályok**
 -
- **Interfészek**
 -
- **Asszociációk**
 - **Room currentRoom**: Minden karakternek tudnia kell, hogy melyik szobában van.
 - **Item inventory**: Egy karakternek tudnia kell, hogy milyen tárgyak vannak nála.
- **Attribútumok**
 - **int dazed**: számontartja, hogy a karakter meddig van még elkábulva
- **Metódusok**
 - `void inANewRoom()`: Belép egy új szobába.
 - `void pickUpItem(Item i)`: Felvesz egy tárgyat.
 - `void dropItem(Item i)`: Eldob egy tárgyat.
 - `void removeItem(Item i)`: Eltávolítja az elhasznált tárgyat.

3.3.5 Cloth

- **Felelősség**
 - A táblatörlő rongyot reprezentáló osztály.

- **Ősosztályok**
 - Item
- **Interfészek**
 -
- **Asszociációk**
 -
- **Attribútumok**
 - **int remainingTime**: A védelemből hátralévő idő.
- **Metódusok**
 - **void blindTeacher(Room r)**: Elkábítja a szoba oktatóit.
 - **void lowerRemainingTime()**: Csökkenti a hátralévő időt.

3.3.6 Door

- **Felelősség**
 - Az ajtót reprezentáló osztály.
- **Ősosztályok**
 -
- **Interfészek**
 -
- **Asszociációk**
 - **Room connectedRooms**: Az ajtónak tudnia kell melyik két szobát köti össze.
- **Attribútumok**
 - **bool isOneWay**: Meghatározza, hogy az adott ajtó egy- vagy kétirányú. Ha egyirányú, akkor a tömb 0. indexű eleméből megy az 1. indexű elem felé.
 - **bool isClosed**: Eltűnő ajtós szoba esetén jelzi, ha az ajtók nem használhatók.
- **Metódusok**
 - **void closeDoors()**: Mágikus szoba bezárhatja az ajtókat.
 - **void changeRoom(Character c)**: Áthelyezi a kapott karaktert az egyik szobájából a másikba.
 - **Room getOtherRoom(Room r)**: Visszatér az ajtó másik oldalán lévő szobát.
 - **void swapNeighboursWith(Room r1, Room r2)**: Összeolvadás vagy szétválás után a szomszédokat beállítja helyesen. r1 jelöli a frissítendő szobát r2 pedig az új szobát.

3.3.7 Item

- **Felelősség**
 - A tárgyak egységes kezelésére használjuk.
- **Össztályok**
 -
- **Interfészek**
 -
- **Asszociációk**
 - **Character heldBy**: Tudnia kell, hogy melyik karakternél van.
 - **Room containedBy**: Tudnia kell, melyik szobában található.
- **Attribútumok**
 -
- **Metódusok**
 -

3.3.8 Map

- **Felelősség**
 - A pályát reprezentáló osztály.
- **Össztályok**
 -
- **Interfészek**
 -
- **Asszociációk**
 - **Room allRooms**: Tudnia kell az összes szobáról.
- **Attribútumok**
 - **int timer**: Hátra lévő időt tárolja.
- **Metódusok**
 - **void splitRoom(Room r)**: A kapott szobát felbontja kettő szobára és elosztja a kettő között az eredeti szoba szomszédait és tárgyait.
 - **void mergeRooms(Room r1, Room r2)**: A kapott szobákat összevonja.
 - **void loadMap(String fileName)**: Betölti pályát, mivel a pálya mindig ugyan az.
 - **void checkStudents ()**: Megnézi, hogy van-e még hallgató.

3.3.9 Mask

- **Felelősség**
 - A maszkot reprezentáló osztály.
- **Össztályok**
 - Item
- **Interfészek**
 -
- **Asszociációk**
 -
- **Attribútumok**
 - **int remainingTime**: A védelemből hátralévő idő.
 - **int usedTimes**: Hányszor használták már a maszkot.
- **Metódusok**
 - **void lowerRemainingTime()**: Csökkenti a hátralévő időt.
 - **void increaseUsedTimes()**: Növeli a felhasznált alkalmak számát.

3.3.10 Room

- **Felelősség**
 - A szobában lévő karakterekért és tárgyakért felelős.
- **Össztályok**
 -
- **Interfészek**
 -
- **Asszociációk**
 - **Door allDoors**: Tudnia kell az összes ajtójáról.
 - **Character allCharacters**: Tudnia kell a szobában lévő karakterekről.
 - **Item allItems**: Tudnia kell a szobában heverő tárgyokról.
- **Attribútumok**
 - **enum Type()**: A szoba típusát tárolja, (gázos, mágikus, cloth-ed).
 - **int maxCapacity**: a szoba befogadóképessége
- **Metódusok**
 - **void gasThem()**: A szobában lévő karakterek gázzal elkábítja.
 - **void clothThem()**: A szobában lévő oktatókat táblatörlővel elkábítja.
 - **void closeTheDoors()**: Mágikus szobák becsukhatják az ajtajaikat.

- **void dropThemOut():** Kibuktatja a védelemmel nem rendelkező hallgatókat.
- **bool areThereCharactersInTheRoom():** Megnézi, hogy van-e még karakter a szobában.
- **void newCharacter(Character c):** a belépni akaró karakter beléphet, ha van hely a szobában.

3.3.11 SlidingRuler

- **Felelősség**
 - A logarlécet reprezentáló osztály, aminek a megtalálása az egyetlen nyerési feltétel.
- **Ősosztályok**
 - Item
- **Interfészek**
 -
- **Asszociációk**
 -
- **Attribútumok**
 -
- **Metódusok**
 -

3.3.12 Student

- **Felelősség**
 - Egy hallgatót reprezentáló osztály.
- **Ősosztályok**
 - Character
- **Interfészek**
 -
- **Asszociációk**
 -
- **Attribútumok**
 -
- **Metódusok**
 - **void dropOut():** Kibukik a hallgató az egyetemről / meghal.
 - **void useItem(Item i):** Használ egy használható tárgyat.

- **bool hasSlidingRuler()**: Megnézi, hogy nála van-e a logarléc.

3.3.13 Teacher

- **Felelősség**
 - A hallgató kibukását okozza.
- **Ősosztályok**
 - Character
- **Interfészek**
 -
- **Asszociációk**
 -
- **Attribútumok**
 -
- **Metódusok**
 -

3.3.14 Transistor

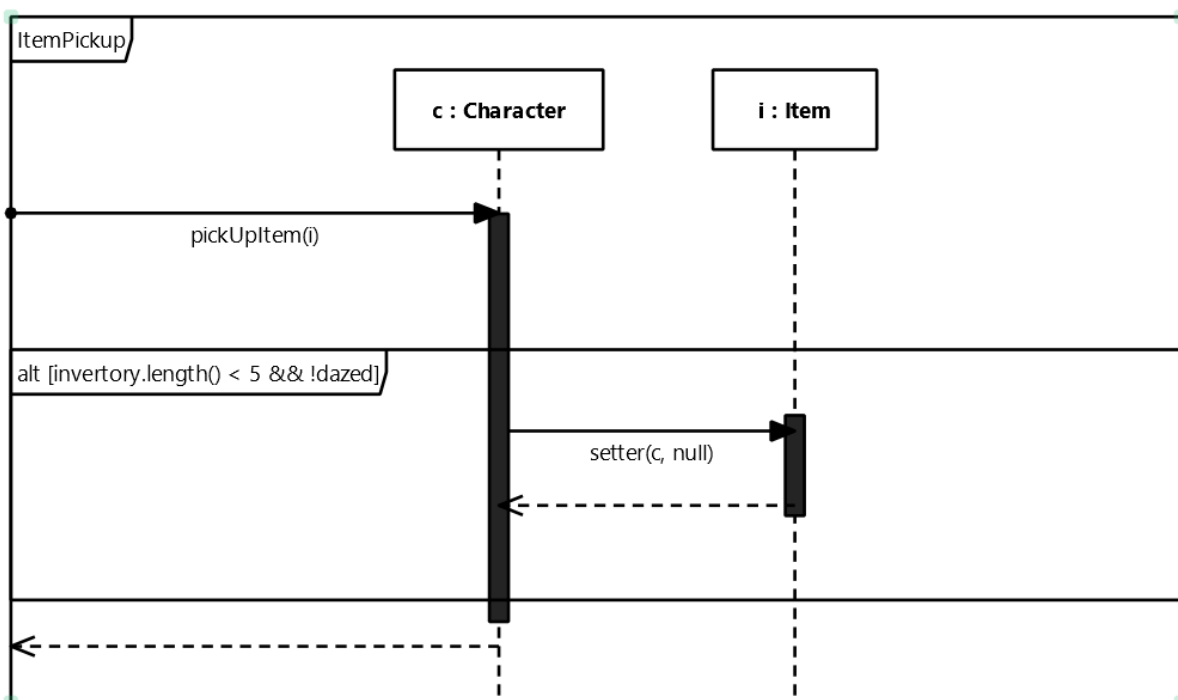
- **Felelősség**
 - A hallgató teleportálására képes.
- **Ősosztályok**
 - Item
- **Interfészek**
 - Usable
- **Asszociációk**
 - **Transistor pair**: Az összepárosított tranzisztor párja.
- **Attribútumok**
 -
- **Metódusok**
 - **void place()**: Az összekötött tranzisztor párból az egyiket leteszi.
 - **void connect(Transistor t)**: Összeköt egy tranzisztort egy másikkal.

3.3.15 Usable

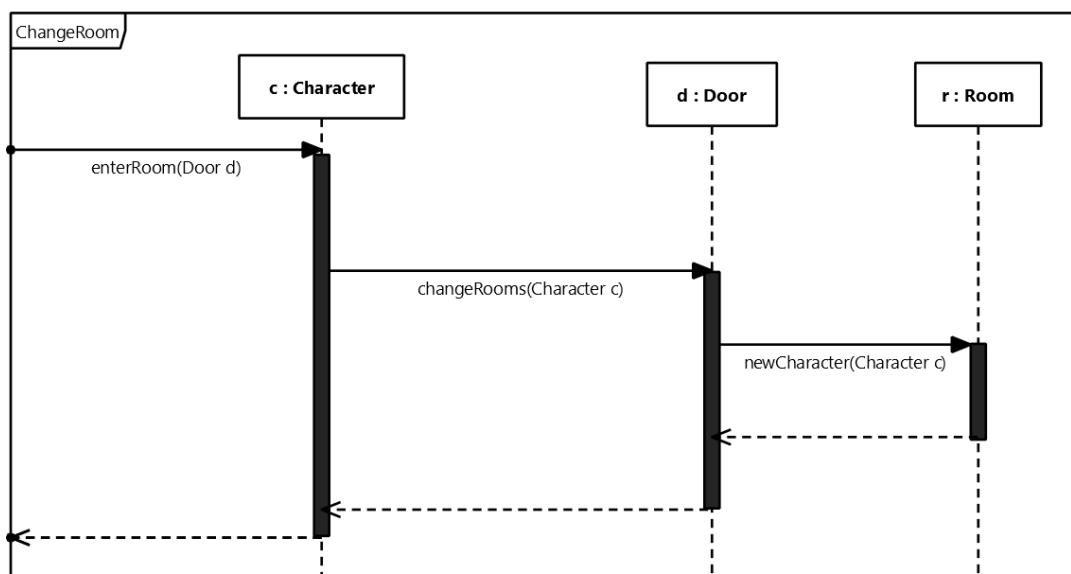
- **Felelősség**
 - A manuálisan felhasználható tárgyak lehet használni rajta keresztül.
- **Össztályok**
 -
- **Asszociációk**
 -
- **Metódusok**
 - **void use()**: A tárgyhasználatot biztosítja.

3.4 Szekvencia diagramok

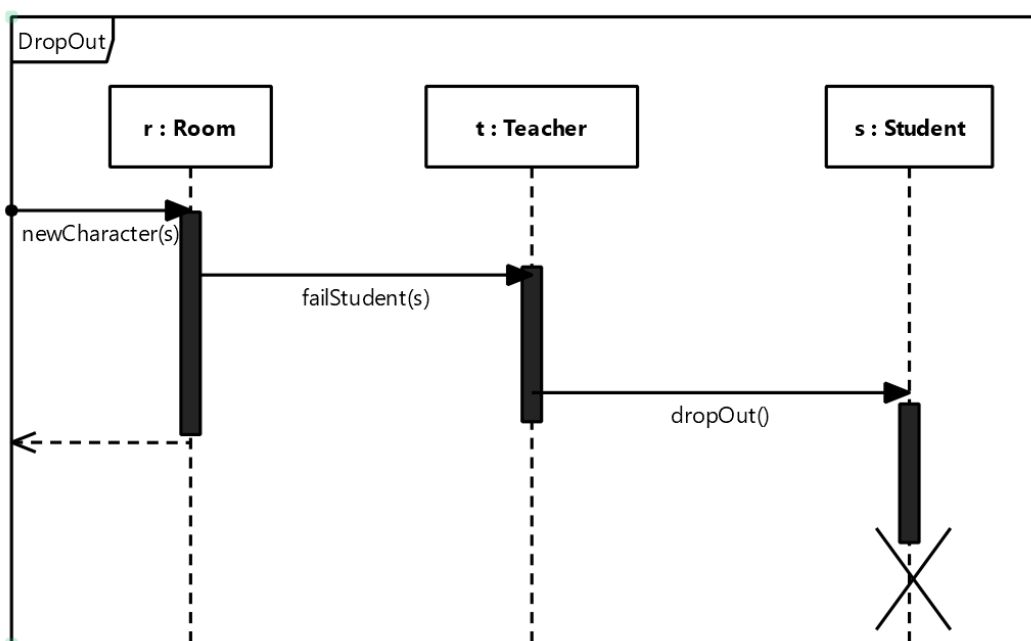
3.4.1 Tárgy felvétele



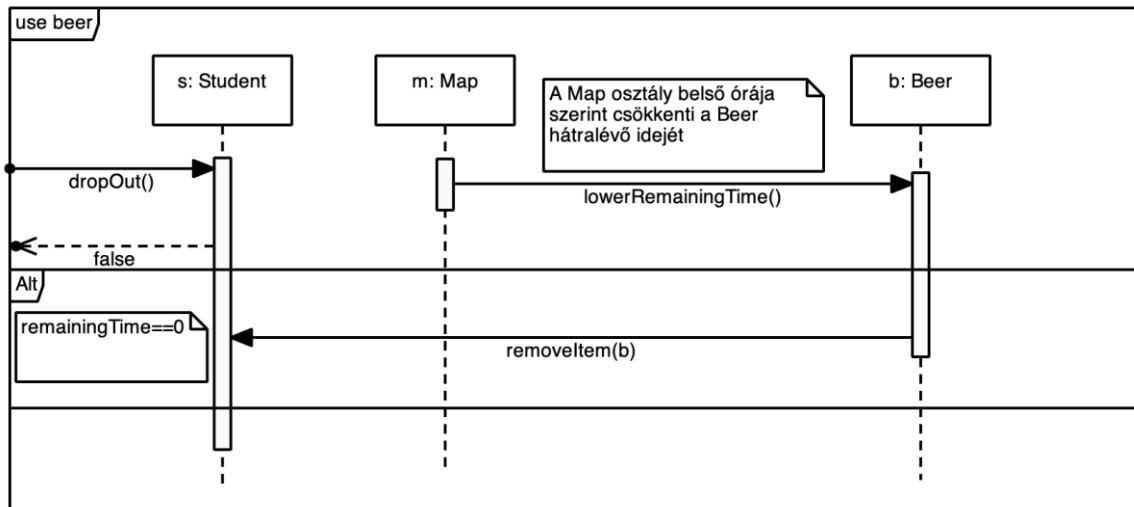
3.4.2 Szobaváltás



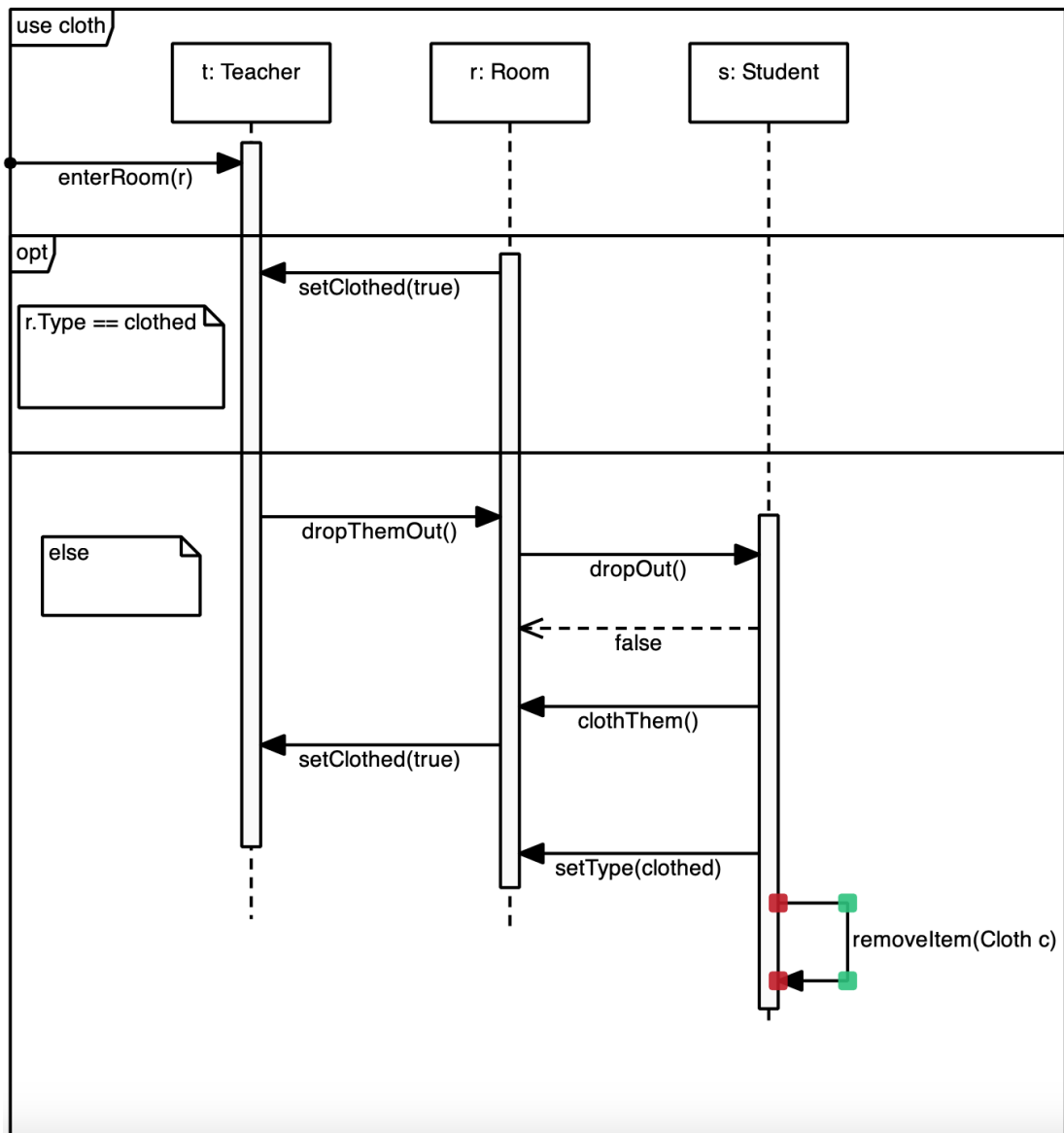
3.4.3 Kibukás

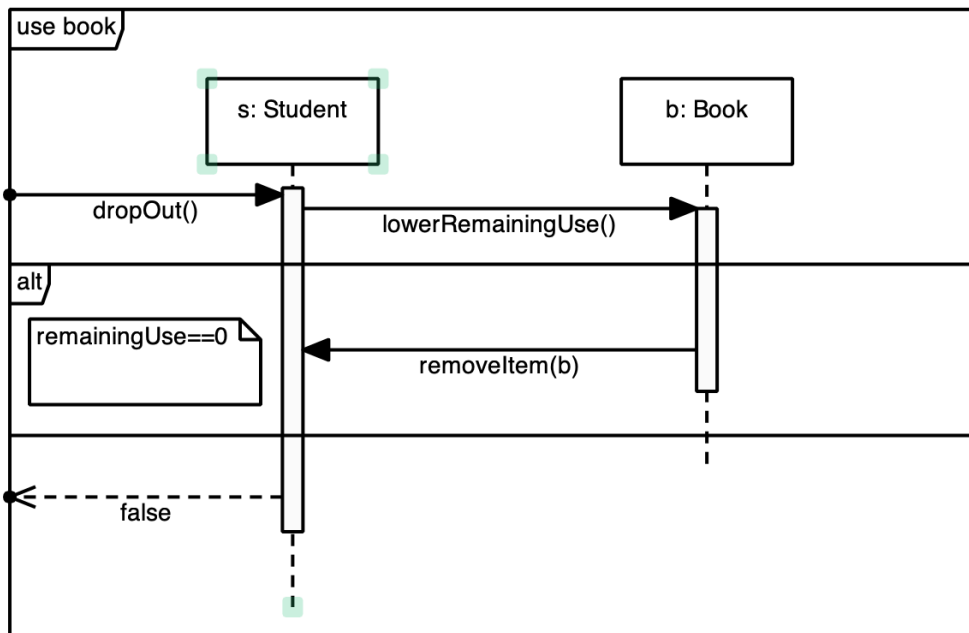
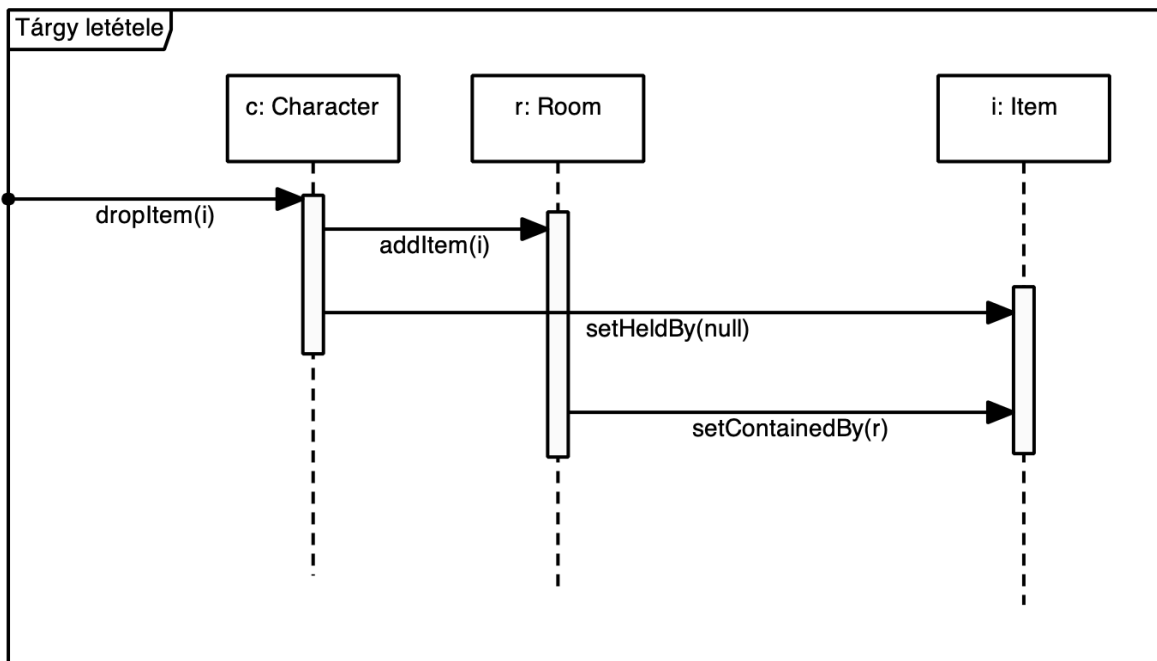


3.4.4 Item használat: sör

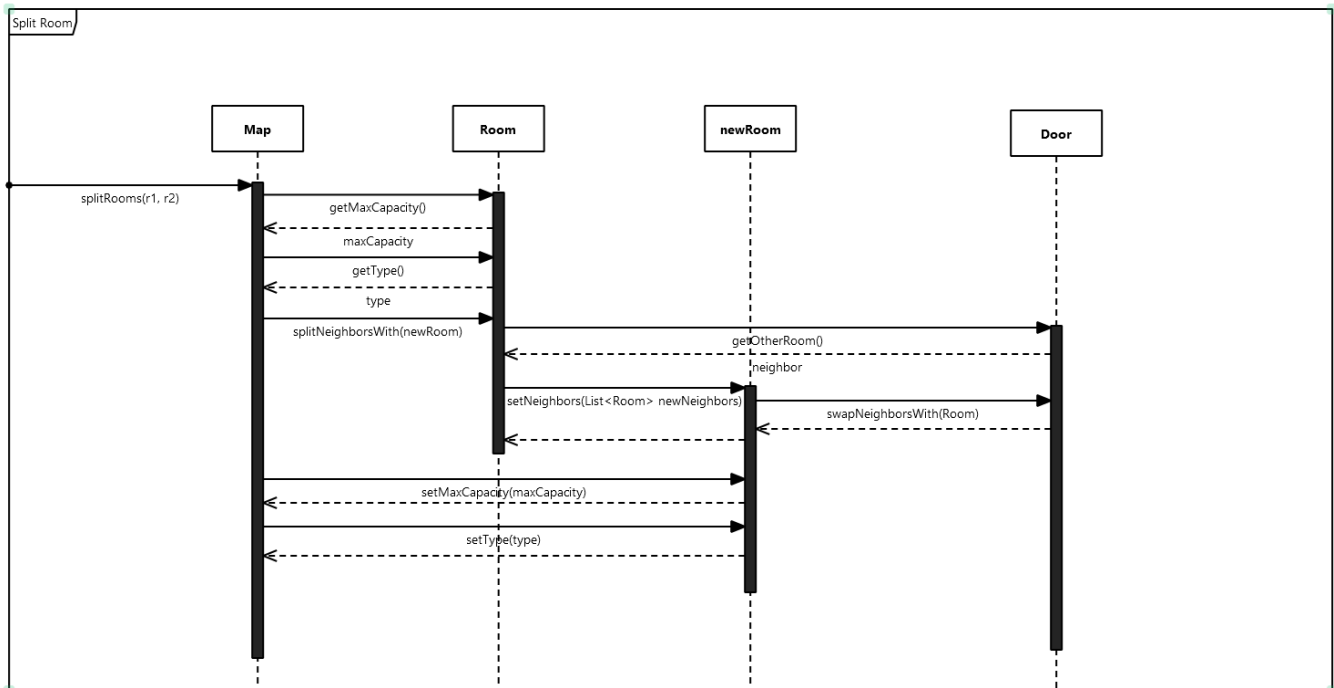


3.4.5 Item használat: rongy

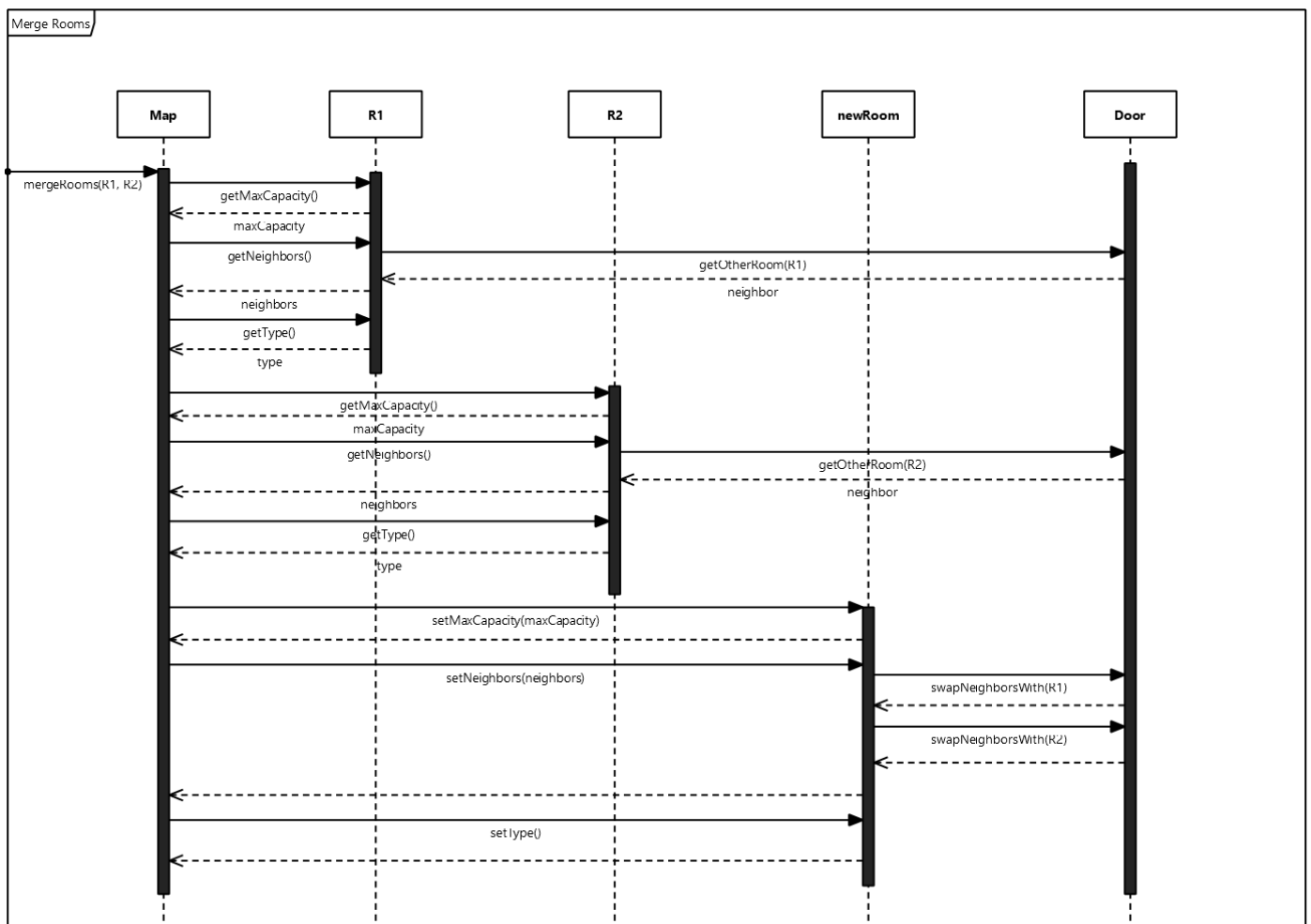


3.4.6 Item használat: tvsz**3.4.7 Tárgy letétele**

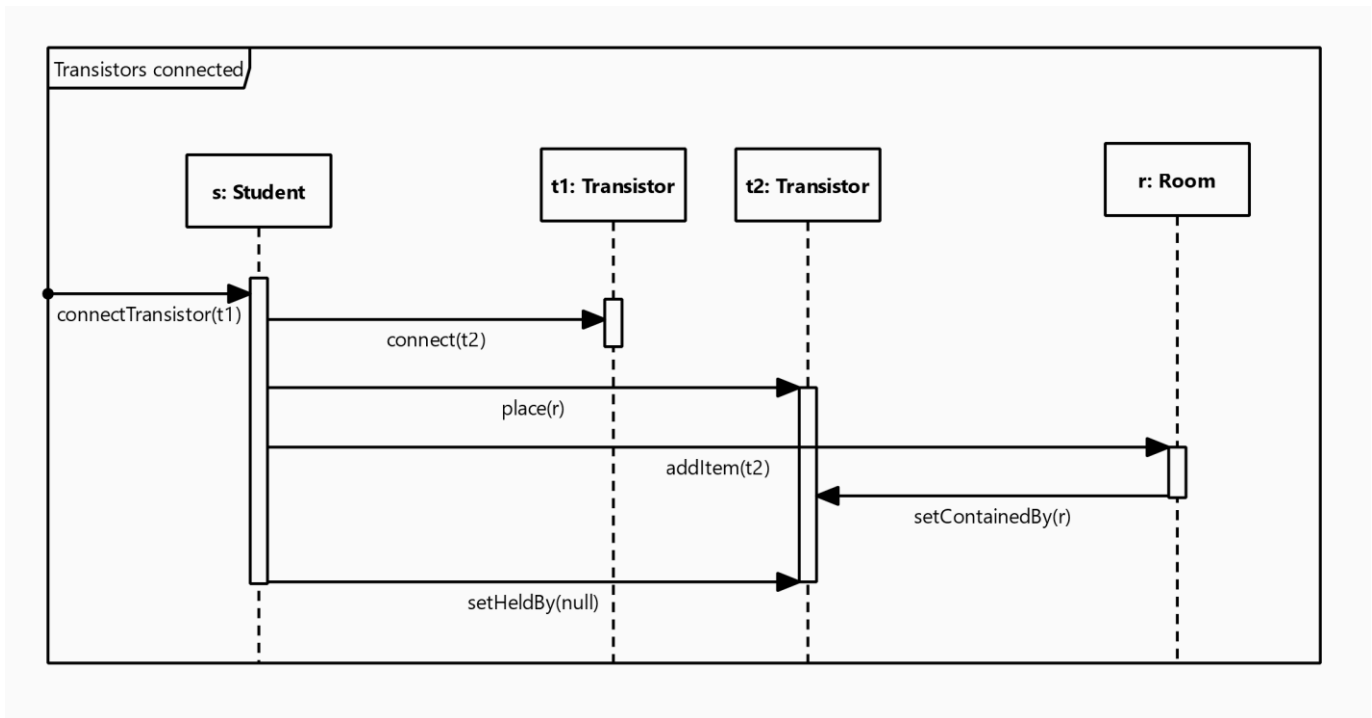
3.4.8 Szobák szétválása



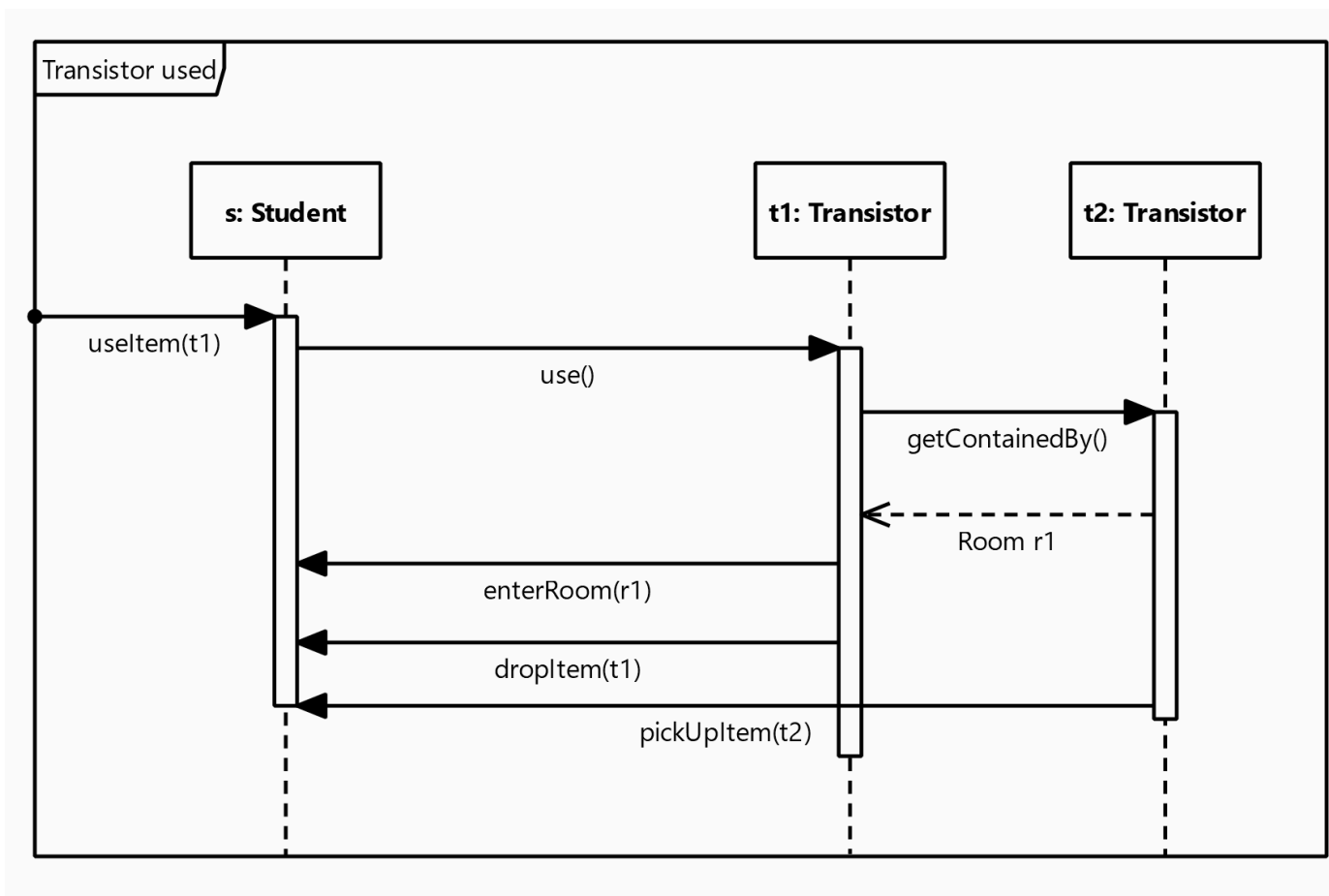
3.4.9 Szobák összevonása



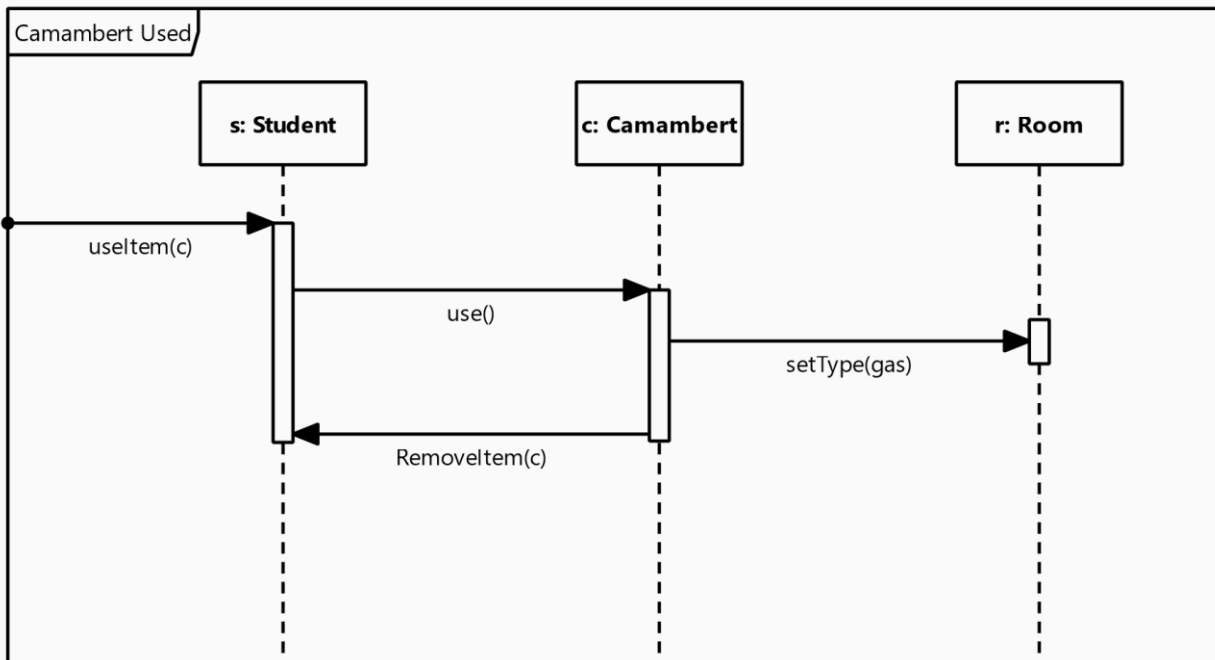
3.4.10 Tranzisztorok összekapcsolása



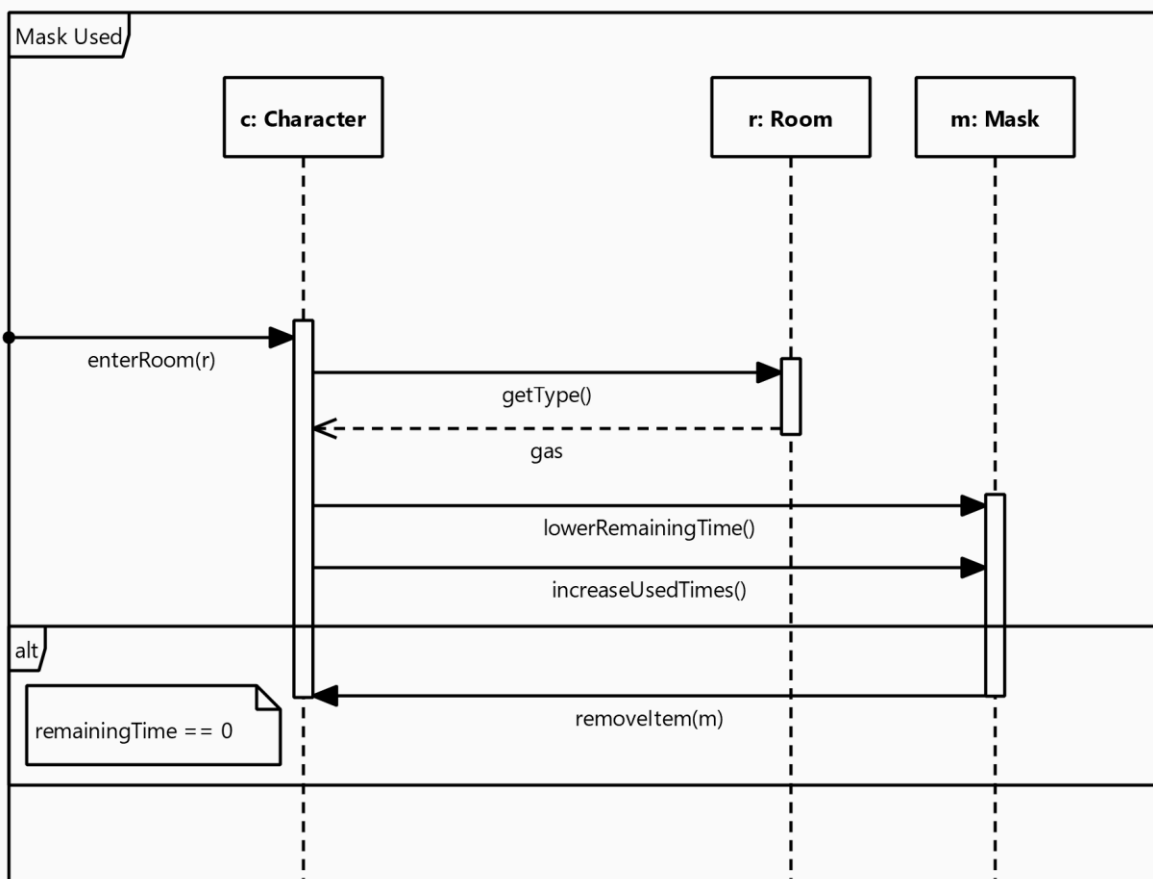
3.4.11 Tranzisztorok használata



3.4.12 Ítem használat: Camembert

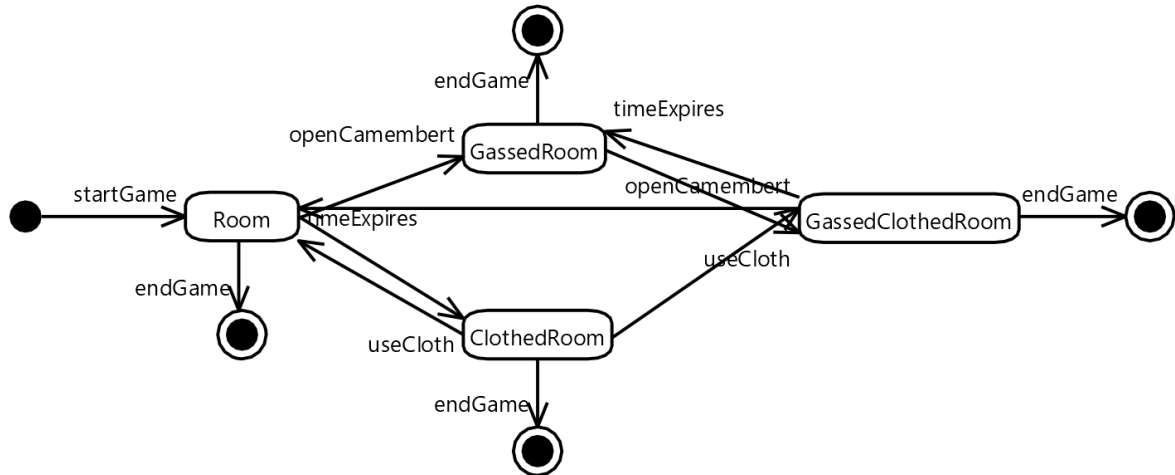


3.4.13 Ítem használata: maszk

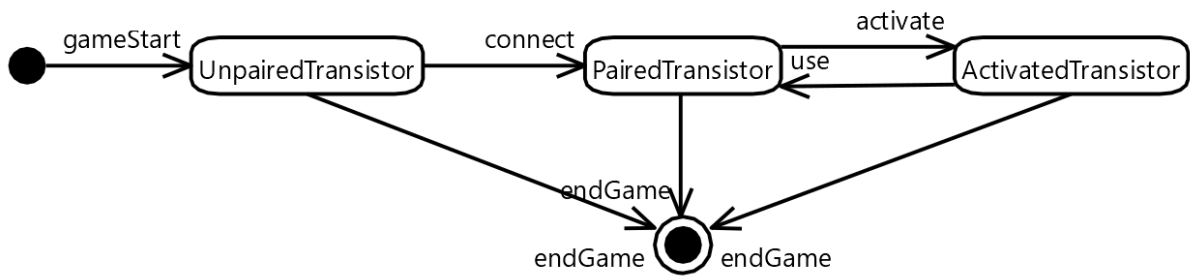


3.5 State-chartok

3.5.1 Szoba státuszok



3.5.2 Tranzisztor státuszok



3.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.02.28. 12:15	1 óra	Soós Kaposvári Hőgye	Értekezlet. Konzultáció
2024.02.29. 22:00	1 óra	Soós Kaposvári Hőgye Reszegi Havasi	Tevékenység: 3.2 megtervezése
2024.03.01. 20:00	2 óra	Soós Kaposvári	Tevékenység: 3.3 Osztályleírás megtervezése
2024.03.02. 20:00	2,5 óra	Soós Kaposvári Hőgye Reszegi Havasi	Tevékenység: 3.3 Osztályleírás véglegesítése
2024.03.03 15:00	2 óra	Havasi	3.2 és 3.5 alpontok elkészítése
2024.03.03 16:00	3 óra	Soós	A maszk, camembert, tranzisztor működését bemutató szekvencia diagramok
2024.03.03 16:00	3 óra	Kaposvári	Sör, rongy, TVSZ használatának és tárgyak eldobásának szekvencia diagramok
2024.03.03 17:00	2 óra	Hőgye	Tárgyak felvételének, átlépés másik szobába és a hallgató kibukásának szekvencia diagramok
2024.03.03 15:00	4 óra	Reszegi	Szobák szétválásának és összeolvadásának szekvencia diagramja
2024.03.03 20:00	3 óra	Havasi	Dokumentumok összesítése és nyomtatás