

## 4. Analízis modell II.

### 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

2024.03.11

### 3. Analízis modell kidolgozása

#### 3.1 Objektum katalógus

##### 3.1.1 Sör

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 TVSZ

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 Camembert

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 Rongy

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 Ajtó

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

##### 3.1.6 Maszk

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

##### 3.1.7 Szoba

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.

##### 3.1.8 Logarléc

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.

##### 3.1.9 Tanuló

A hallgatót a játékos irányítja. A szobák között tud mozogni ajtók használatával vagy tranzisztorok 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 logarlécet akkor a játéknak vége.

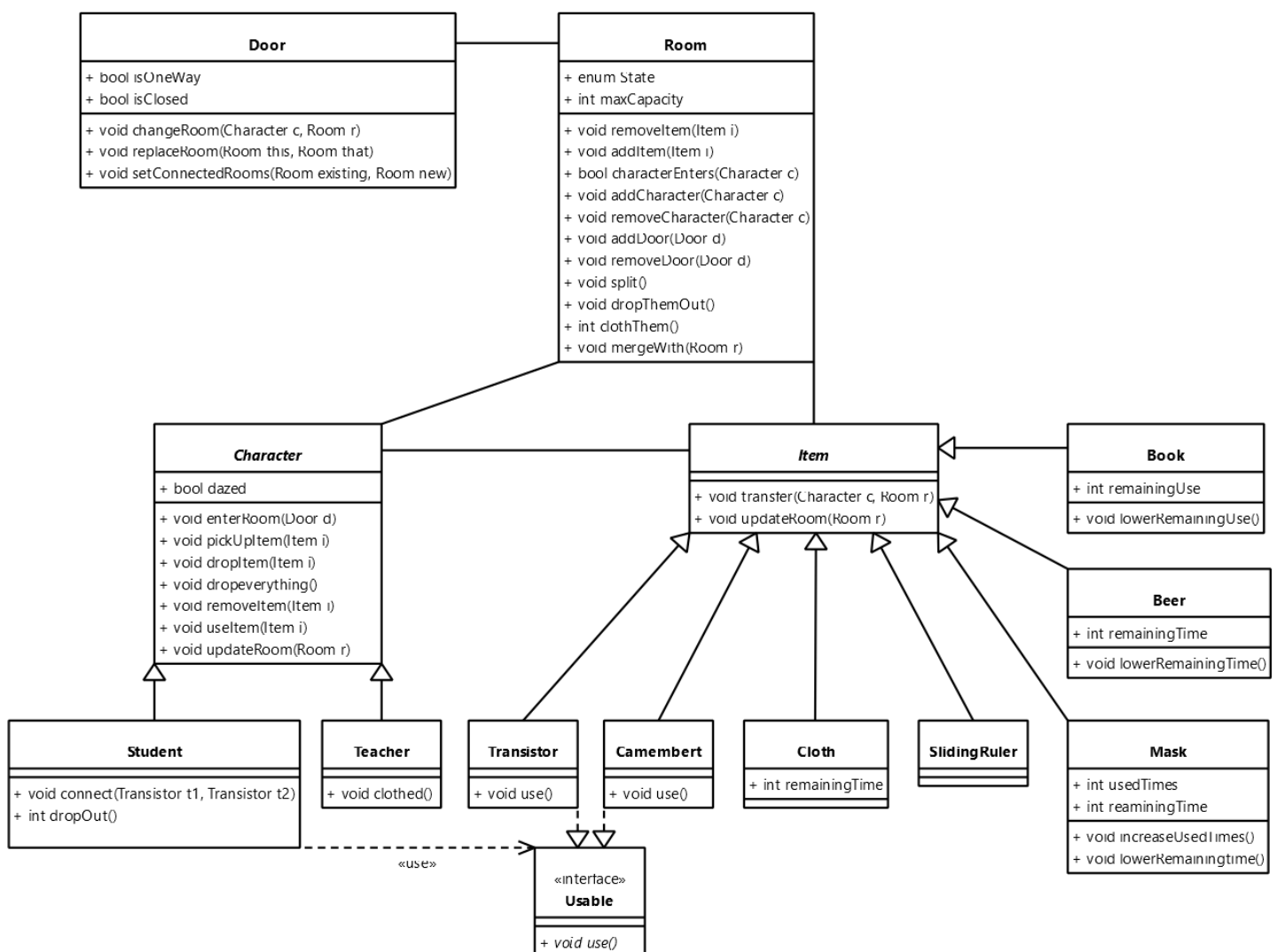
### 3.1.10 Oktató

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.

### 3.1.11 Tranzisztor

Felvehető tárgy. Ha 2 van egy hallgatónál akkor összekapcsolhatók. A hallgatónál lévő fél megjegyzi, hogy a letett fél melyik szobában van. Összekapcsolt és bekapcsolt tranzisztor 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 Camembert

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

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

### 3.3.4 Character

- **Felelősség**
  - A karaktereket megvalósító absztrakt osztály, (oktató és hallgató).
- **Ősosztályok**
  -
- **Interfészek**
  -
- **Asszociációk**
  - **Room currentRoom**: Minden karakternek tudnia kell, hogy melyik szobában van.
  - **Item inventory**: Egy karakternek tudnia kell, hogy milyen tárgyak vannak nála.
- **Attribútumok**
  - **bool dazed**: Számontartja, hogy a karakter el van-e kábulva.
- **Metódusok**
  - **void enterRoom(Door d)**: Belép egy új szobába.
  - **void pickUpItem(Item i)**: Felvesz egy tárgyat.
  - **void dropItem(Item i)**: Eldob egy tárgyat.
  - **void dropEverything()**: Eldob mindent.
  - **void removeItem(Item i)**: Eltávolítja az elhasznált tárgyat.
  - **void useItem(Item i)**: Ha usable az item akkor használja.
  - **void updateRoom(Room r)**: Átállítja a karakter szobáját.

### 3.3.5 Cloth

- **Felelősség**
  - A táblatörlő rongyot reprezentáló osztály.
- **Össztályok**
  - Item
- **Interfészek**
  -
- **Asszociációk**
  -
- **Attribútumok**
  - **int remainingTime**: A védelemből hátralévő idő.
- **Metódusok**
  -

### 3.3.6 Door

- **Felelősség**
  - Az ajtót reprezentáló osztály.
- **Össztályok**
  -
- **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 changeRoom(Character c, Room r)**: Átrakja a karaktert a szoba párjába.
  - **void replaceRoom(Room this, Room that)**: Kicseréli a this-t a that-re a nyilvántartásban.
  - **void setConnectedRooms(Room existing, Room new)**: Beállítja hogy melyik két szoba között van.

### 3.3.7 Item

- **Felelősség**
  - A tárgyak egységes kezelésére használjuk.
- **Ősosztályok**
  -
- **Interfészek**
  -
- **Asszociációk**
  - **Character heldBy**: Tudnia kell, hogy melyik karakternél van.
  - **Room containedBy**: Tudnia kell, melyik szobában található.
- **Attribútumok**
  -
- **Metódusok**
  - **void transfer(Character c, Room r)**: Setteli a pozícióját.
  - **void updateRoom(Room r)**: Átállítja a karakter szobáját.

### 3.3.8 Mask

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

### 3.3.9 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 State**: A szoba állapotát tárolja, (gázos, mágikus, cloth-ed).
  - **int maxCapacity**: A szoba befogadóképessége.
- **Metódusok**
  - **void removeItem(Item i)**: Kitörli a szoba nyilvántartásából az item-et.
  - **void addItem(Item i)**: Felveszi az item-et a szoba nyilvántartásába.
  - **bool characterEnters(Character c)**: A c karakter megpróbál belépni.
  - **void addCharacter(Character c)**: Felveszi az karaktert a szoba nyilvántartásába.
  - **void removeCharacter(Character c)**: Kitörli a szoba nyilvántartásából a karaktert.
  - **void addDoor(Door d)**: Felvesz egy új ajtót a szoba nyilvántartásába.
  - **void removeDoor(Door d)**: Kitöröl egy ajtót a szoba nyilvántartásából.
  - **void split()**: Kezdeményezi a szoba szétválását.
  - **void dropThemOut()**: Minden karaktert megpróbál kibuktatni.
  - **void clothThem()**: Minden karaktert megpróbál a nedves táblatörlő hatása alá vonni.
  - **void mergeWith(Room r)**: Kezdeményezi a szobák összeolvadását.

### 3.3.10 SlidingRuler

- **Felelősség**
  - A logarlécet reprezentáló osztály, aminek a megtalálása az egyetlen nyelési feltétel.
- **Össztályok**
  - Item
- **Interfészek**
  -
- **Asszociációk**
  -



- **Attribútumok**

- 

- **Metódusok**

- 

### 3.3.11 Student

- **Felelősség**

- Egy hallgatót reprezentáló osztály.

- **Ősosztályok**

- Character

- **Interfészek**

- 

- **Asszociációk**

- 

- **Attribútumok**

- 

- **Metódusok**

- **void connect(Transistor t1, Transistor t2):** Összeköt két tranzisztort.
- **int dropOut():** Megnézi, hogy a tanuló meg tudja-e védeni magát, a tárgyaktól függő számmal tér vissza.

### 3.3.12 Teacher

- **Felelősség**

- A hallgató kibukását okozza.

- **Ősosztályok**

- Character

- **Interfészek**

- 

- **Asszociációk**

- 

- **Attribútumok**

- 

- **Metódusok**

- **void clothed():** Megbénítja a tanárt.

### 3.3.13 Transistor

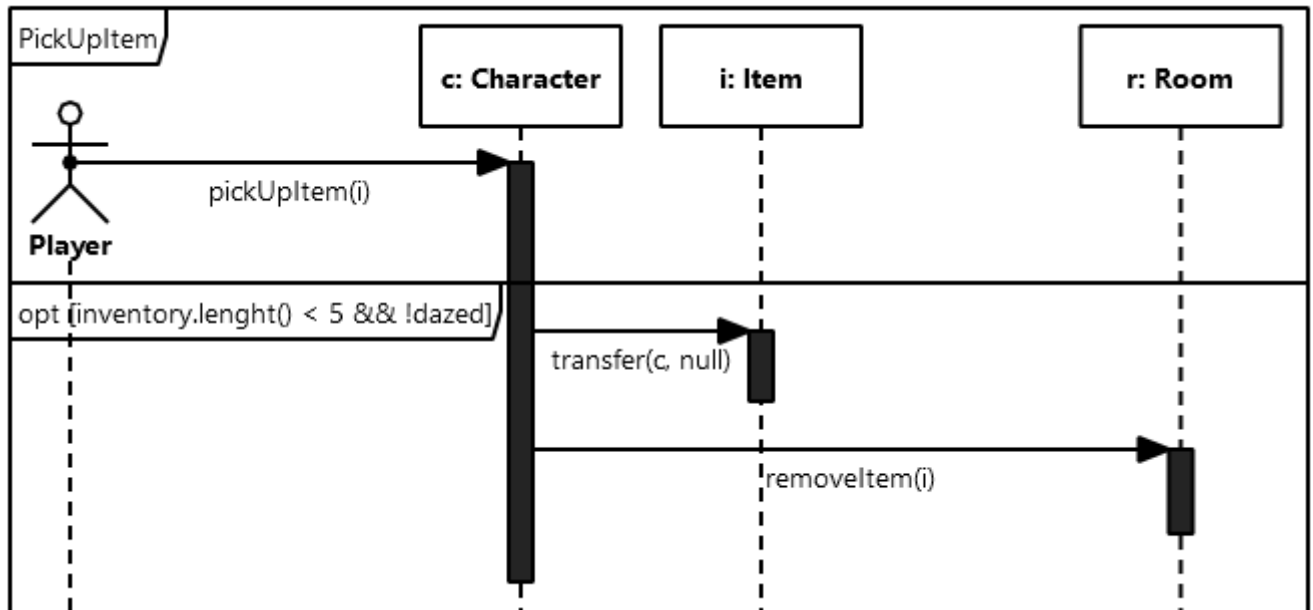
- **Felelősség**
  - A hallgató teleportálására képes.
- **Ősosztályok**
  - Item
- **Interfészek**
  - Usable
- **Asszociációk**
  - **Transistor pair:** Az összepárosított tranzisztort párja.
- **Attribútumok**
  -
- **Metódusok**
  - **void use():** Bekapcsol egy tranzisztort.

### 3.3.14 Usable

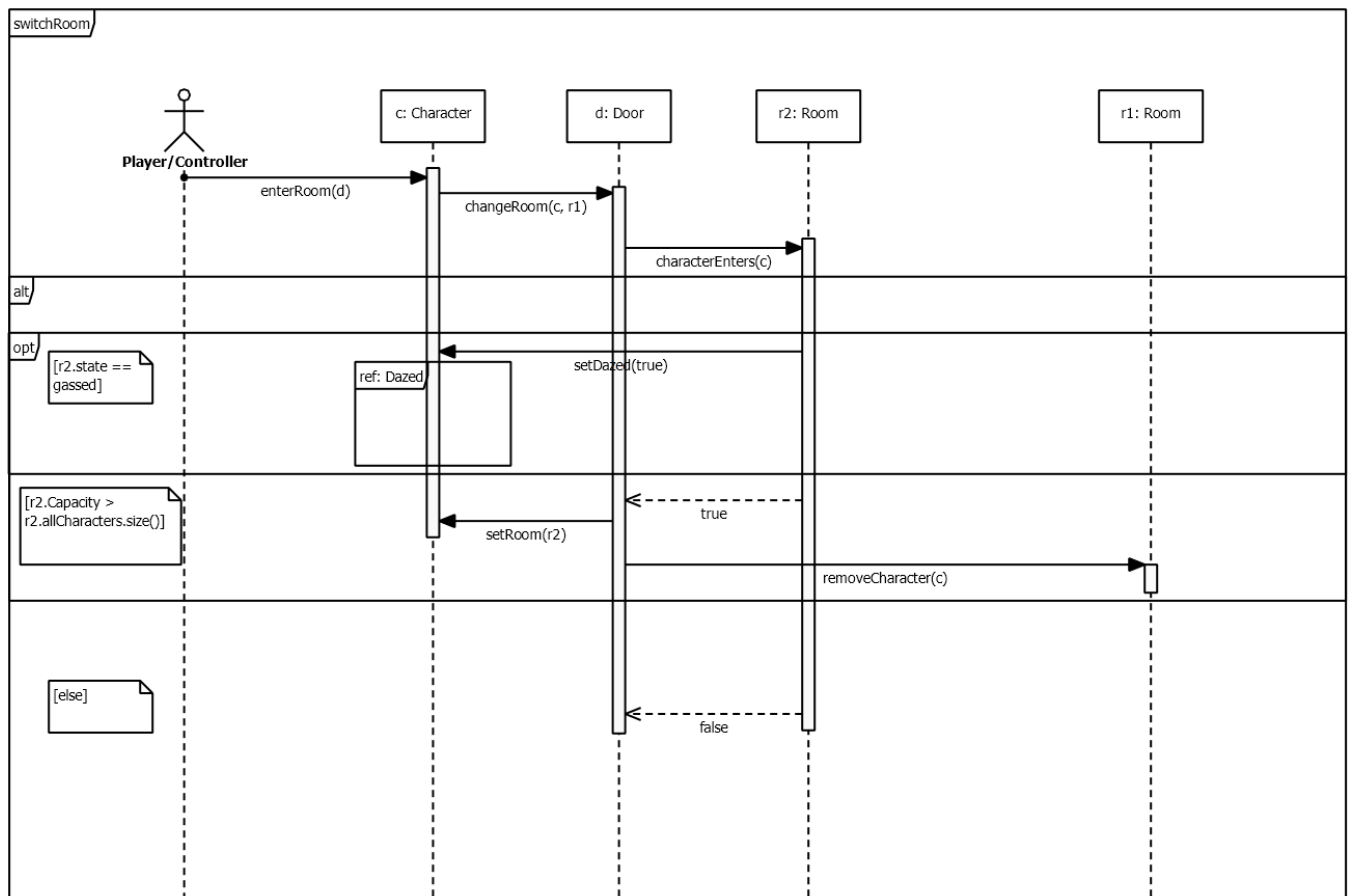
- **Felelősség**
  - A manuálisan felhasználható tárgyak lehet használni rajta keresztül.
- **Ősosztályok**
  -
- **Asszociációk**
  -
- **Metódusok**
  - **void use():** A tárgyhasználatot biztosítja.

### 3.4 Szekvencia diagramok

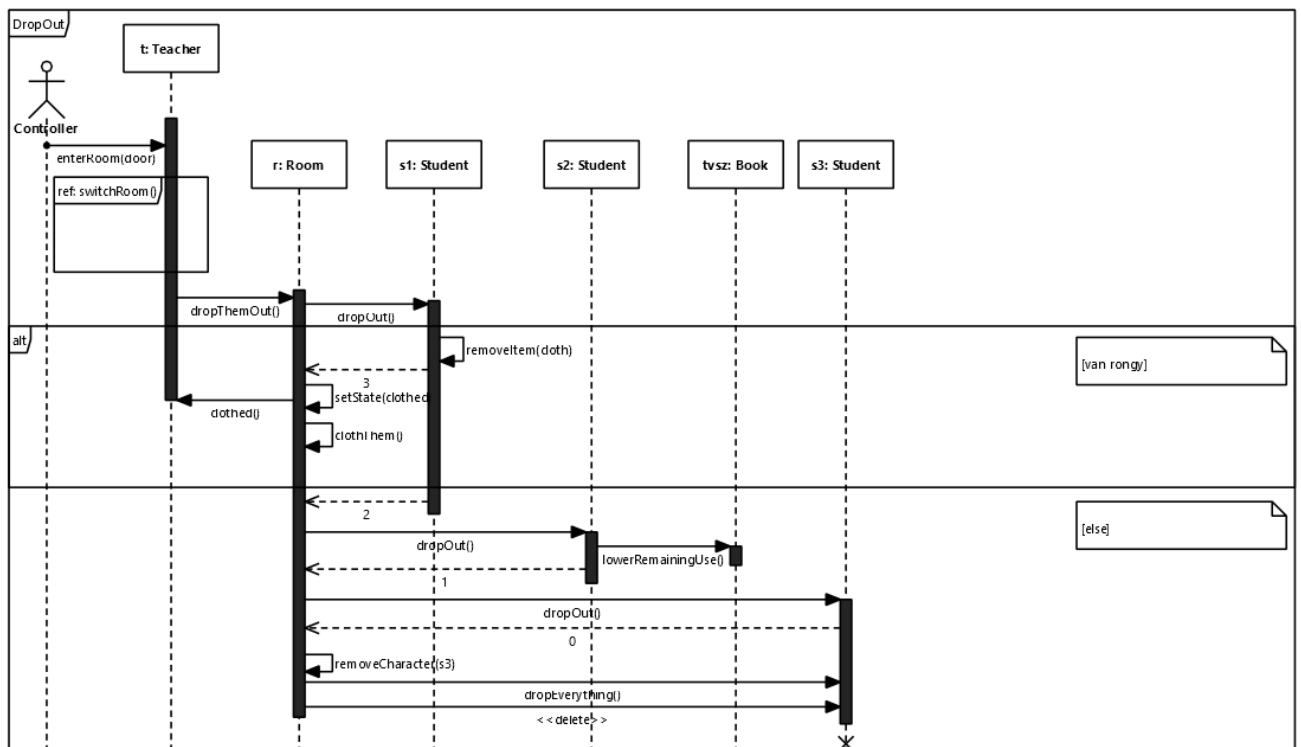
#### 3.4.1 Tárgy felvétele



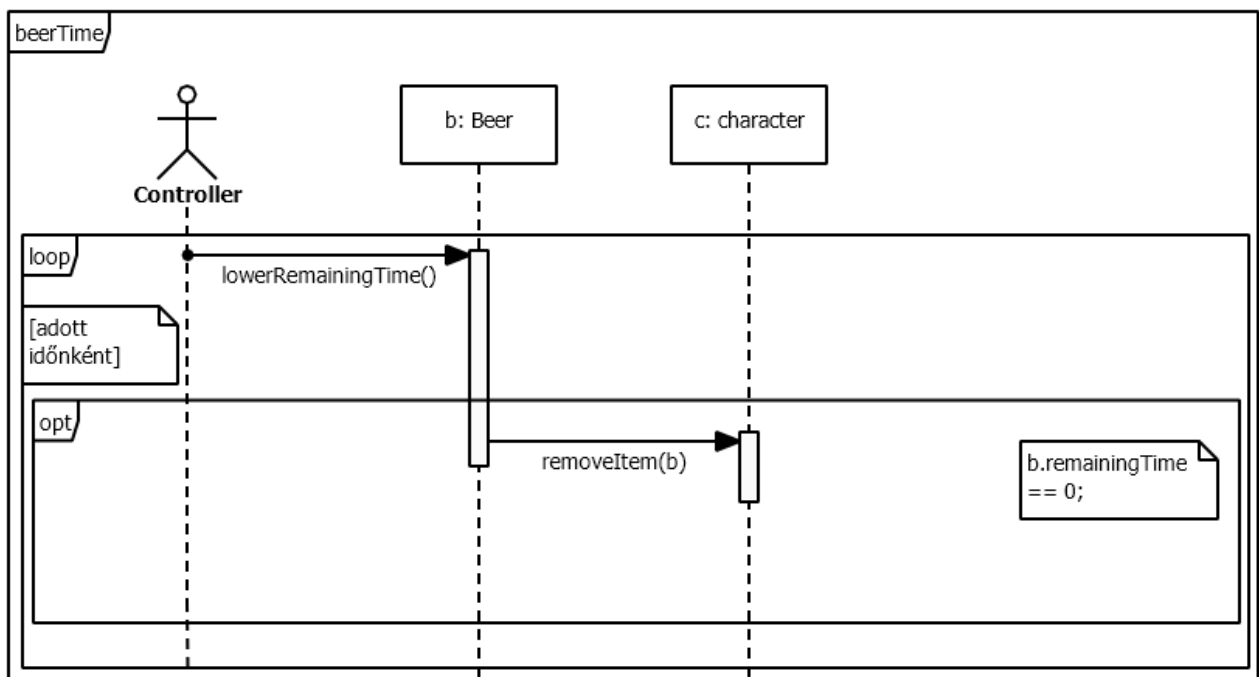
#### 3.4.2 Szobaváltás



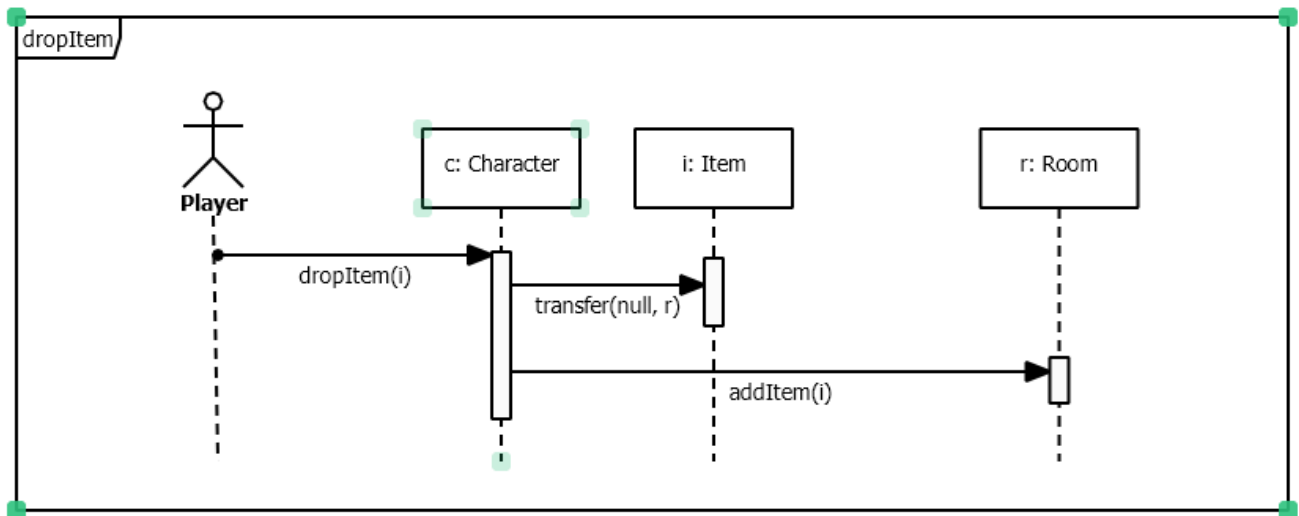
## 3.4.3 Kibukás



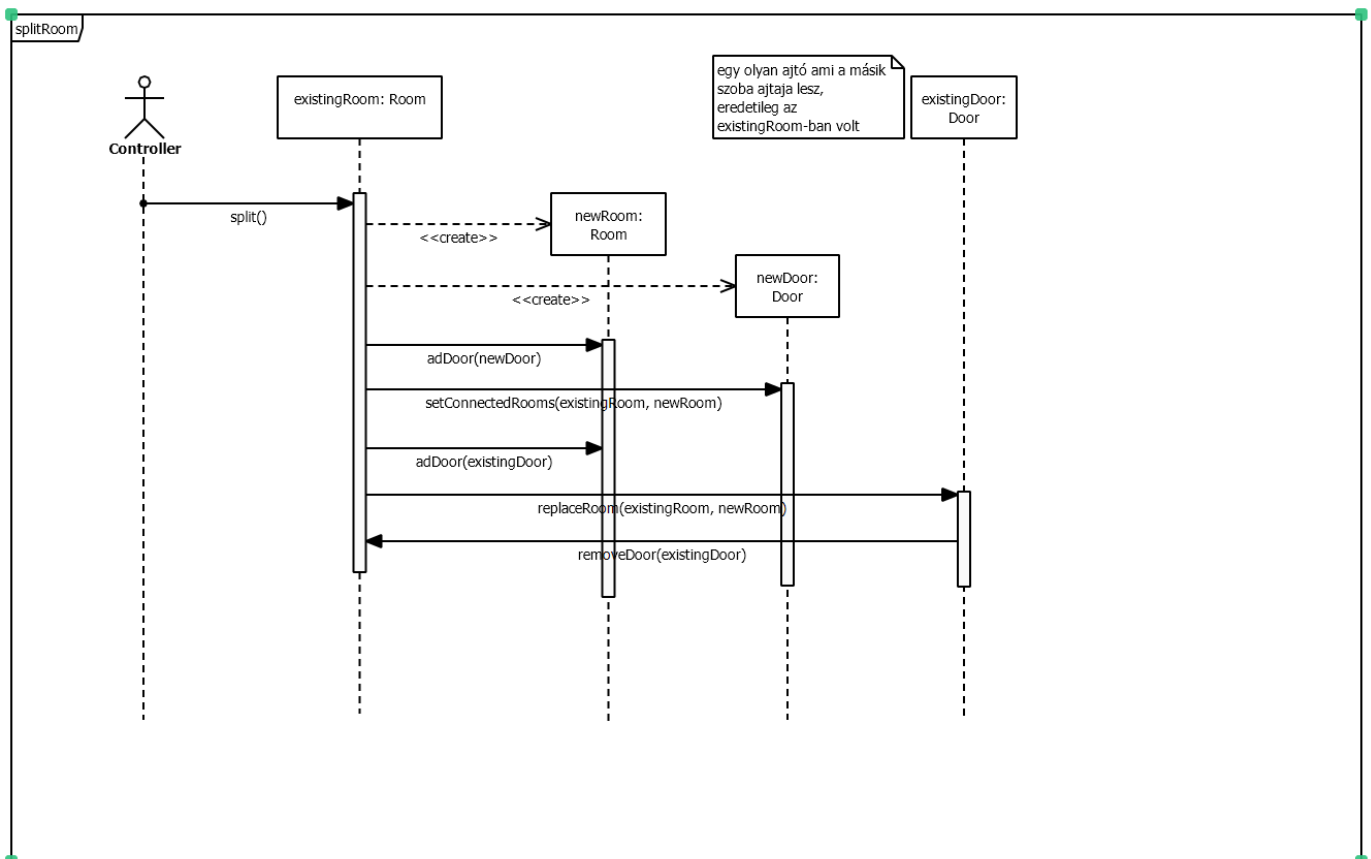
## 3.4.4 Sör



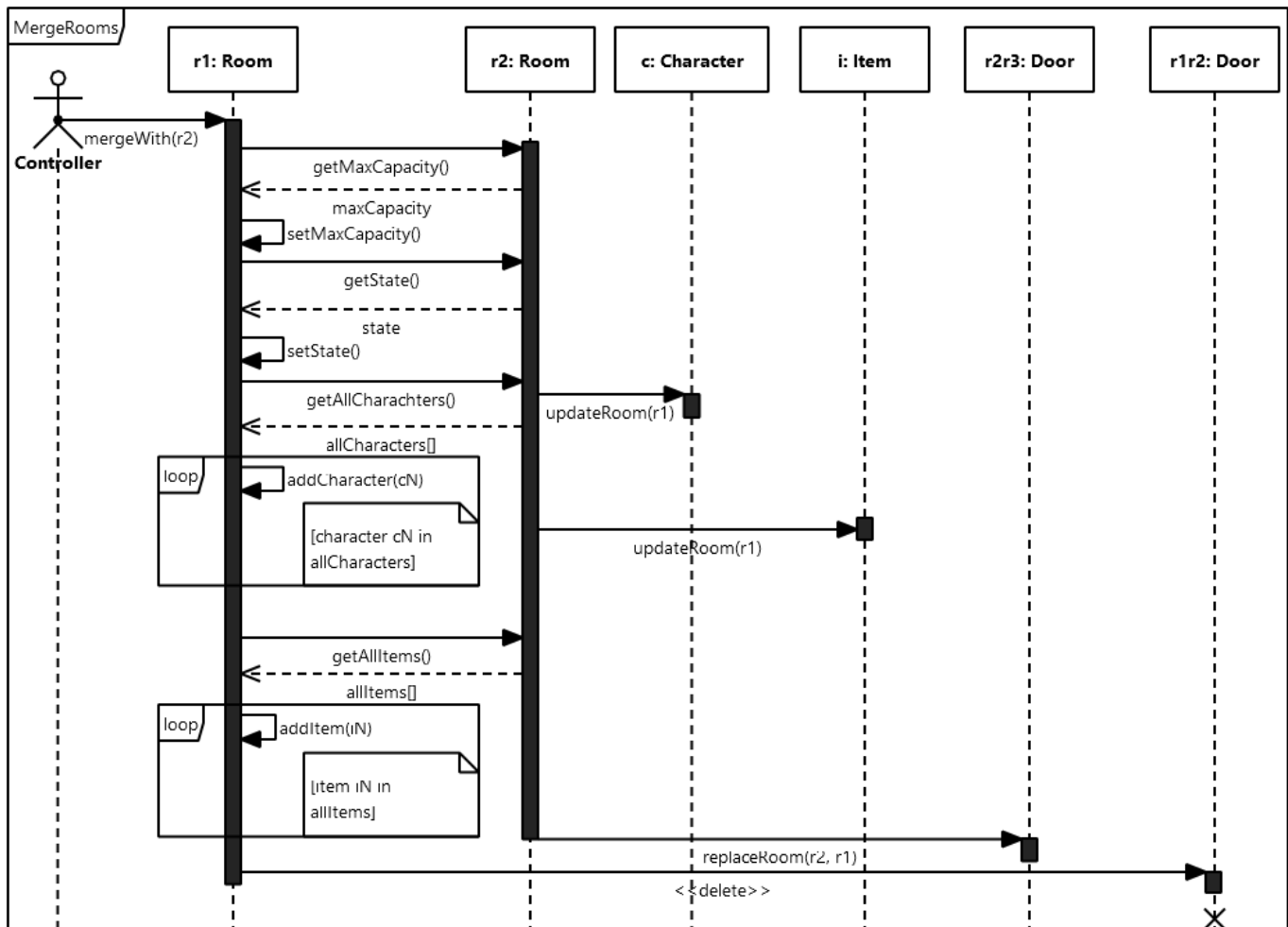
### 3.4.5 Tárgy letétele



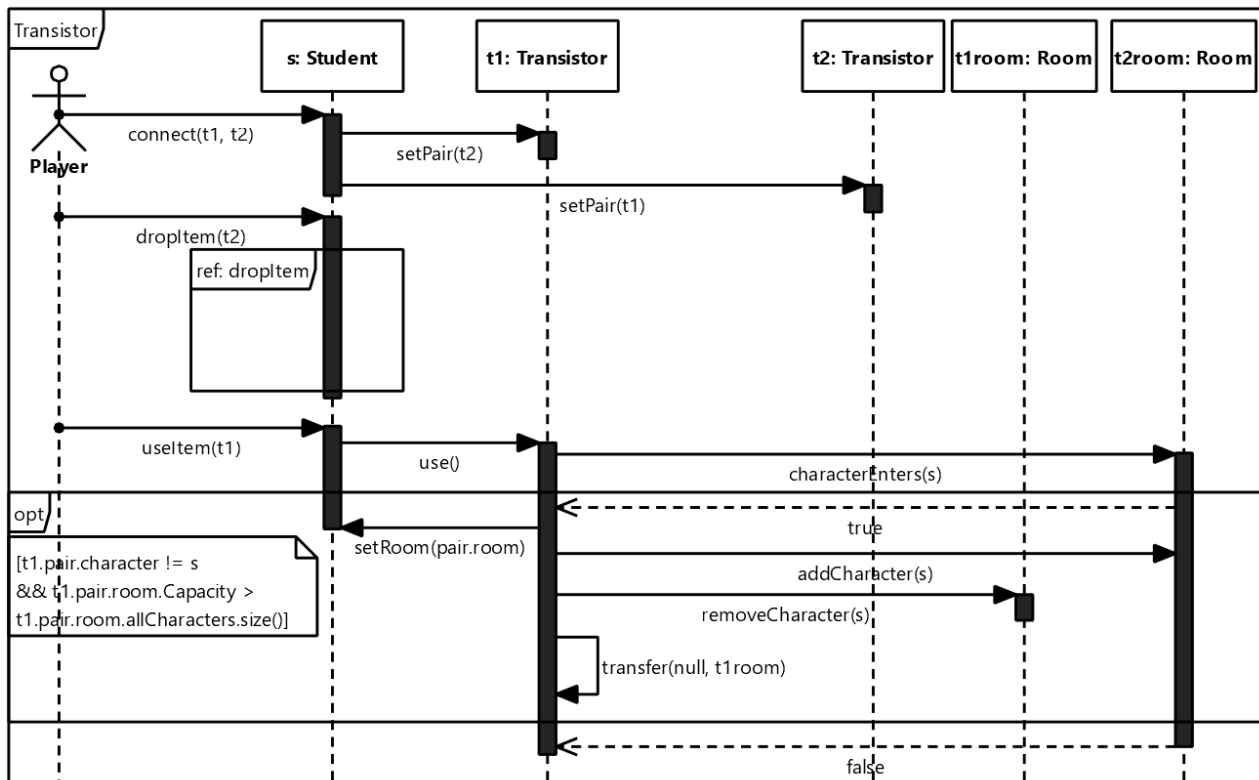
### 3.4.6 Szobák szétválása



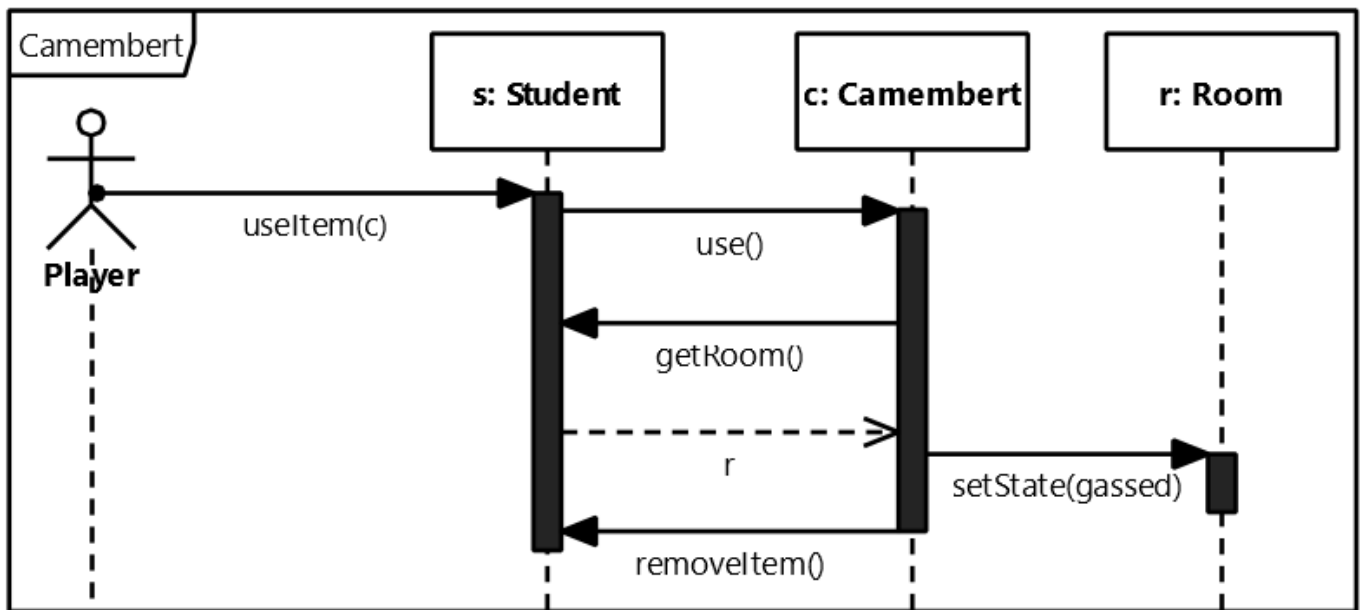
### 3.4.7 Szobák összevonása



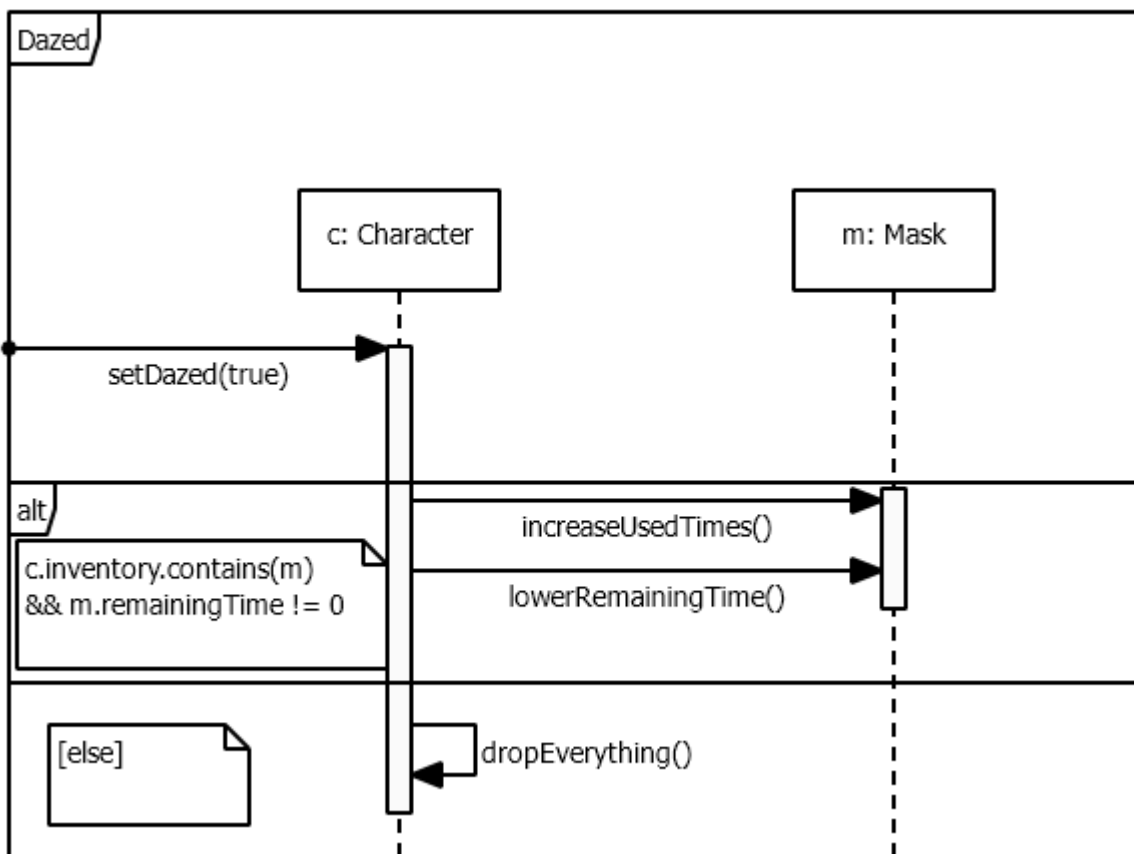
### 3.4.8 Tranzisztorok



## 3.4.9 Item használat: Camembert

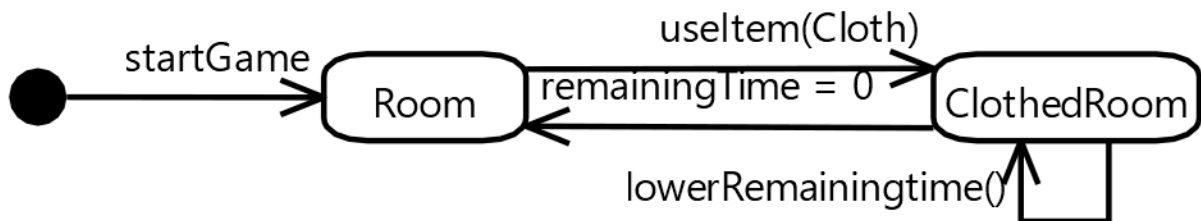
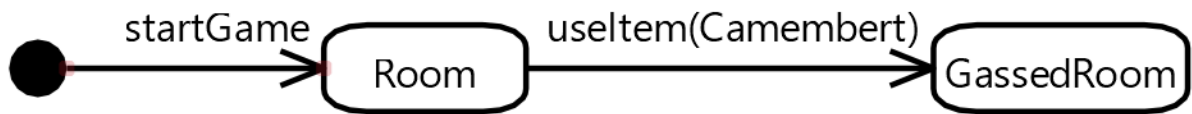


## 3.4.10 Item használata: maszk



### 3.5 State-chartok

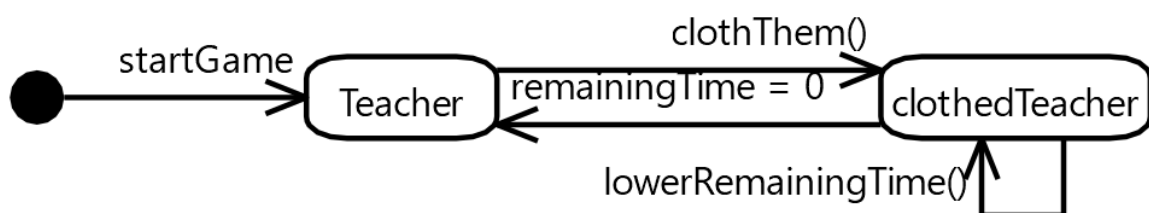
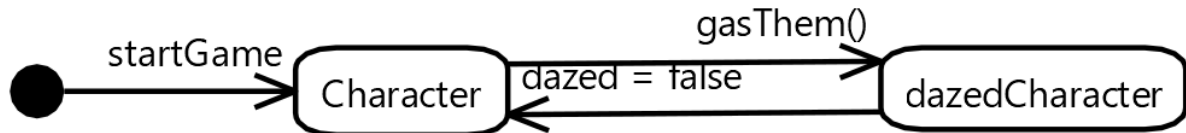
#### 3.5.1 Szoba státuszok



#### 3.5.2 Tranzisztor státuszok



#### 3.5.3 Karakter állapotok





### 3.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.03.06. 12:15	2 óra	Soós Kaposvári Hógye Havasi Reszegi	Értekezlet. Konzultáció
2024.03.10. 13:00	5 óra	Kaposvári Hógye Havasi	Tevékenység: Szekvencia diagrammok újratervezése
2024.03.10. 19:00	6 óra	Soós Kaposvári Havasi Reszegi Hógye	Tevékenység: Szekvencia diagrammok befejezése, state- chartok és statikus struktúra diagramm elkészítése
2024.03.11 01:00	1 óra	Havasi	Dokumentum összeszerkesztése