

## **4. Analízis modell kidolgozása II.**

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

2024.03.11.

### 3. Analízis modell kidolgozása

#### 3.1 Objektum katalógus

##### 3.1.1 Labirintus

A játék színtere, amely szobákból áll, ezeket számon is tartja.

Képes ajtók megjelenítésére valamint törlésére, emellett szobákat tud egyesíteni és összeolvasztani.

##### 3.1.2 Hallgató

A felhasználó által irányított karakter. Célja, hogy a számítógép által irányított oktatók ne kapják el (ezzel a halálukat okozva) azelőtt, mielőtt megszerzi a Logarlécet, mellyel megnyeri a játékot.

Van neve és számon tartja, hogy melyik szobában tartózkodik. Körönként egyszer akciót hajt végre, mely során egyszer vele szomszédos szobába léphet, ahova vezet ajtó és a létszám engedi. Ezenkívül az akciója során tud felvenni tárgyakat, ezeket szintén számontartja és a maximális számuk 5 lehet, a tárgyakat fel tudja használni, elégetni, valamint letenni.

A Hallgatókra a környezetük is hathat, amely különböző állapotokat idézhet elő, ezek a következők: mérges gáztól védett, oktatótól védett, bénult, eszméletét vesztt.

##### 3.1.3 Oktató

A számítógép által irányított karakter. Célja, hogy megakadályozza a felhasználó/játékos által irányított hallgatót a játék megnyerésében. A vele egy szobában védettséggel nem rendelkező hallgatót/hallgatókat megöli.

Van neve és számon tartja, hogy melyik szobában tartózkodik. Körönként egyszer akciót hajt végre, mely során először egyszer vele szomszédos szobába lép (ha van ilyen), ahova vezet ajtó és a létszám engedi. Ezt követően a szobában lévő összes tárgyat felveszi és egyből el is égeti. Ha a Logarlécet vette fel, azt nem égeti el, hanem néhány kör után a lépése előtt leteszi.

Az Oktatókra a környezetük is hathat, amely különböző állapotokat idézhet elő, ezek a következők: bénult, eszméletét vesztt.

##### 3.1.4 Szoba

A szoba az a színtér, ahol a játék játszódik. A játéktér (Labirintus) számtalan szoba alkotja, amelyeknek különböző befogadóképessége van.

Van szobaszáma és rendelkezik befogadóképességgel, ezenkívül számontartja a benne tartózkodó karaktereket.

Képes osztódásra és egyesülésre, valamint megmondani, hogy egy lépéssel meg szobákba lehet belőle eljutni.

Vannak szobák melyek különleges képességekkel rendelkeznek, ilyenek az elátkozott és a gázos szobák. Az elátkozott szoba az a szoba, amelynek időnként eltűnhet az ajtaja, amin keresztül egy másik szobába átjuthatna a karakter. Az elátkozott szoba esetében az is előfordulhat, hogy nem lehet oda bejutni, mivel az összes ajtaja eltűnt. A gázos szoba az a szoba, ahol a karakter a bent megtalálható mérges gáz miatt elájul.

Gázos szoba esetében, ha a karakterről lekerül a bénítás, addig élvez védeltséget a gázzal szemben, ameddig nem távozik a szobából.

A szobákban tárgyak vannak, amelyeket a karakterek fel tudnak venni.

### **3.1.5 Logarléc**

A logarléc az a tárgy, amelynek a megszerzésével a hallgató megnyeri a játékot. A tárgyat egyébként az oktató is fel tudja venni, azonban 1-2 lépés után megszabadul tőle, azaz visszarakja a játéktérre.

### **3.1.6 Tranzisztor**

Olyan tárgy, amelyből kettőt összekapcsolva a hallgatók az egyik szobából a másikba tudnak teleportálni.

A tranzisztorok számon tartják, hogy van-e párjuk. Ha a hallgató eszköztárába van két tranzisztor, amelynek nincs párja, akkor azok automatikusan összekapcsolódnak.

A teleportáláshoz fel kell használni az egyik összekapcsolt tranzisztort, ezzel az bekapcsolva letevéődik a szobában. Ilyenkor már a párját felhasználva automatikusan visszakerülünk abba a szobába, ahol az első tranzisztort otthagytuk. A teleportálás után a két tranzisztor a két szobában kikapcsolt és párosítatlan állapotban lesz. Ilyenkor ezek újra felhasználhatók válnak.

### **3.1.7 Nedves táblatörlő rongy**

Olyan tárgy, amit a hallgató tud hatékonyan használni arra, hogy a hallgatóval egy szobában levő oktatókat lebénítsa. Rendelkezik használati idővel, mivel a felvételétől számítva csak adott ideig használható. A használhatósága körönként mindig csökken, és ha már nem használható, magától elég a karakter eszköztárában.

### **3.1.8 FFP2-es maszk**

Olyan tárgy, amit a hallgató a mérges gázzal teli szobákban a gázzal szembeni védelemre tud használni. Rendelkezik használati idővel és számon tartja, hogy használatba lett-e már véve, mivel a legelső használatától számítva csak adott ideig működik. A használhatósága körönként mindig csökken, és ha már nem használható, magától elég a karakter eszköztárában.

### **3.1.9 TVSZ denevérbőrre nyomtatott példánya**

Olyan tárgy, amit a hallgató hatékonyan tud arra használni, hogy megmentse a lelkét attól, hogy azt az oktatók elvegyék. Rendelkezik használati idővel és számon tartja, hogy használatba lett-e már véve, mivel a legelső használatától számítva már csak két alkalommal használható. A használhatósága minden alkalommal csökken, amikor egy oktató megpróbálja elvenni a lelkét, és ha már nem használható, magától elég a karakter eszköztárában.

### **3.1.10 Dobozolt káposztás camembert**

Olyan tárgy, amelyet a karakter tud felhasználni arra, hogy segítségével mérges gázt bocsásson ki a szobában. Ennek hatására a szoba gázos szobává válik.

### **3.1.11 Szent söröspohár**

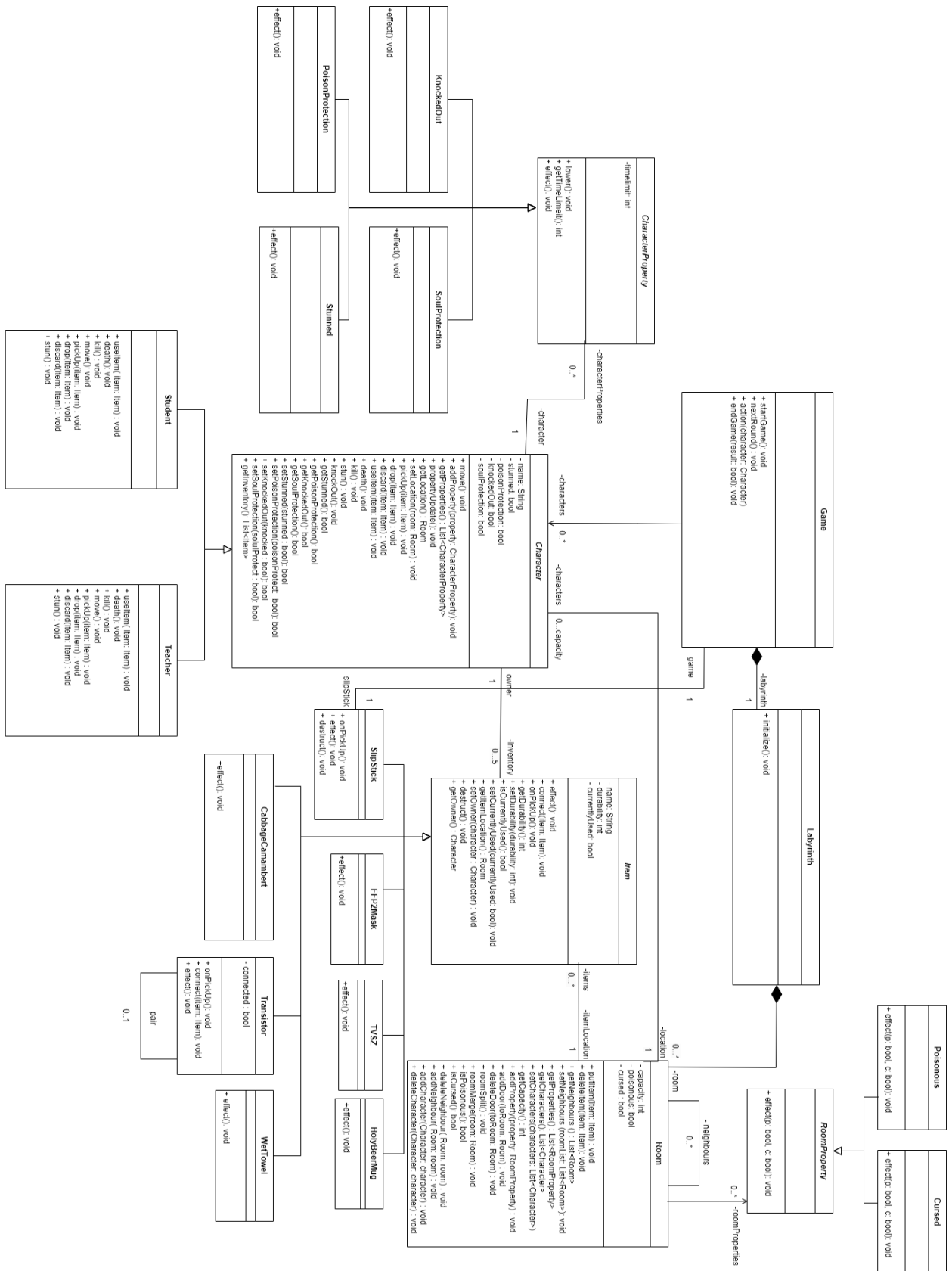
Olyan tárgy, amit a hallgató hatékonyan tud arra használni, hogy megmentse a lelkét attól, hogy azt az oktatók elvegyék. Rendelkezik használati idővel és számon tartja, hogy használatba lett-e már véve, mivel a legelső használatától számítva csak adott ideig működik. A használhatósága körönként mindig csökken, és ha már nem használható, magától elég a karakter eszköztárában.

### **3.1.12 Játék**

A játékmenetet kezeli, vezérli a köröket és vizsgálja, hogy a játék véget ért-e.

Számontartja a játékteret és karaktereket, valamint lehetővé teszi a karakterek számára az akciók végrehajtását.

### 3.2 Statikus struktúra diagramok



### 3.3 Osztályok leírása

#### 3.3.1 CabbageCamambert

- **Felelősség**

Az osztály a Dobozos káposztás camembert tárgy viselkedését valósítja meg. Felelőssége, hogy a hallgatók az osztály felhasználásával a szobának a típusát meg tudják változtatni “Poisonous”-re.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatának a számát jelez, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok**

- **+void effect() :** A metódus azt a szobát, amelyben a hallgató ezt a tárgyat használja, “Poisonous” típusúvá alakítja. Amennyiben már a szoba “Poisonous” volt, akkor nem történik semmi.

#### 3.3.2 Character

- **Felelősség**

Ez egy absztrakt osztály, amely a játékban szereplő karaktereknek az absztrakt osztálya.

- **Össosztályok**

Nem rendelkezik össosztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** A karakter neve.
- **-location:** Azon szoba amelyben a karakter tartózkodik.
- **-inventory:** A karakternél lévő tárgyak tárolódnak itt.
- **-characterProperties:** A karakterre érvényben lévő állapotokat tárolja.
- **-stunned:** A karakter bénított állapotát jelző érték.
- **-knockedOut:** A karakter eszméletvesztett állapotát jelző érték.
- **-poisonProtection:** A karakter mérge elleni védetségét jelző érték.
- **-soulProtection:** A karakter oktató lélek kiszívása elleni védetségét jelző érték.

- **Asszociációk**

- **egyirányú kapcsolat a Character osztályból a CharacterProperties osztály felé:** A Character és a CharacterProperties osztályok közötti kapcsolatot valósítja meg. A karakter tudja magáról milyen státuszai vannak, amelyekből legfeljebb 4 lehet neki.

- **kétirányú kapcsolat a Character osztályból az Item osztály felé:** A Character és az Item osztályok közötti kapcsolatot valósítja meg. A karakter tudja milyen tárgyak vannak az eszköztárában, de egy időben egy karakternél legfeljebb 5 Item lehet.

- **kétirányú kapcsolat a Room és a Character osztályok közt:** A karakterek tudják melyik szobában vannak, a szobák pedig azt hogy mely karakterek tartózkodnak bennük, és egy karakter egy időben csak egy szobában lehet. Egy szobában egy időben a szoba befogadóképességével megegyező számú karakter lehet.

- **egyirányú kapcsolat a Game osztályból a Character osztály felé:** a Game osztály számon tartja a játékban részt vevő karaktereket.

- **Metódusok**

- **+void move() :** A karakter mozgását valósítja meg(absztrakt).
- **+void addProperty(CharacterProperty property) :** A karakterre vonatkozó hatásokat lehet vele állítani.
- **+List<CharacterProperty> getProperties() :** A karakterre vonatkozó hatásokat adja vissza.
- **+Room getLocation() :** Visszaadja a karakter mely szobában helyezkedik el.
- **+void getInventory():** Visszaadja az adott karakternél lévő tárgyak listáját.
- **+void setLocation(Room room) :** A karakter tartózkodási helyét lehet vele állítani.
- **+void pickUp(Item item) :** A karakter felveszi a szobában lévő tárgyat, így a tárgy bekerül a karakter eszköztárába(absztrakt).
- **+void drop(Item item) :** A karakter az eszköztárából abba a szobába teszi le a kiválasztott tárgyat amiben éppen tartózkodik(absztrakt).
- **+void discard(Item item) :** A karakter az eszköztárából elégeti a kiválasztott tárgyat(absztrakt).
- **+void death():** Ha a hallgatónak a lelkét kiszívják, akkor ez a metódus felel a hallgató játékból való kieséséért, ha pedig ezt a függvényt egy oktató hívja meg akkor nem történik semmi(absztrakt).

- **+void kill():** Ezzel a metódussal képes egy oktató egy hallgató lelkét kiszívni, ha ugyanabban a szobában tartózkodnak, és ha a hallgató nem rendelkezik megfelelő védelemmel, ha egy hallgató próbálná ezt a metódust használni, akkor nem történik semmi(absztrakt).
- **+void useItem(Item item):** Ennek a függvénynek a segítségével képes egy hallgató, az eszköztárában szereplő tárgyak közül egyet használni, azonban ha egy oktató próbálná meg ezt a metódust használni, akkor nem történik semmi. (absztrakt).
- **+void propertyUpdate():** A karakterre érvényben lévő hatások értékeit frissíti
- **+void stun() :** A karakter bénult állapotában hívódik meg, kezeli a karakter bénult állapotában való tevékenységét.
- **+void knockOut() :** A karakter eszméletvesztett állapotában hívódik meg, kezeli a karakter eszméletvesztett állapotában lévő tevékenységét.
- **+bool getStunned():** visszaadja a “stunned” tagváltozó értékét.
- **+bool getPoisnProtection():** visszaadja a “poisonProtection” tagváltozó értékét.
- **+bool getKnockedOut():** visszaadja a “knockedOut” tagváltozó értékét.
- **+bool getSoulProtection():** visszaadja a “soulProtection” tagváltozó értékét.
- **+void setStunned():** beállítja a “stunned” tagváltozó értékét.
- **+void setPoisnProtection():** beállítja a “poisonProtection” tagváltozó értékét.
- **+void setKnockedOut():** beállítja a “knockedOut” tagváltozó értékét.
- **+void setSoulProtection():** beállítja a “soulProtection” tagváltozó értékét.

### 3.3.3 CharacterProperty

- **Felelősség**

Olyan absztrakt osztály, melynek felelőssége az, hogy a segítségével a karakterek státuszait nyilván tudjuk tartani.

- **Ősosztályok**

Nem rendelkezik őszosztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- Nincsen attribútuma

- **Asszociációk**

- **egyirányú kapcsolat a Character osztályból a CharacterProperty osztály felé:** A Character és a CharacterProperty osztályok közötti kapcsolatot valósítja meg. A karakter tudja magáról milyen státuszai vannak, amelyekből legfeljebb 4 lehet neki.

- **Metódusok:**

- **+ void lower():** Ezzel a metódussal csökkentjük a timelimit értékét.
- **+ int getTimeLimeit():** Visszaadja a timelimitet értékét.
- **+ void effect(bool st, bool p, bool k, bool so):** Beállítja a paraméterként kapott változókat, melyek a karakter tulajdonságait tükrözik.



### 3.3.4 Cursed

- **Felelősség**

Az elátkozott tulajdonság rendelhető segítségével a szobákhoz.

- **Össztályok**

RoomProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- Nincsen attribútuma.

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok:**

- + **void effect(bool p, bool c):** A szobához tartozó típusokat jelző értékeket beállítja.

### 3.3.5 FFP2Mask

- **Felelősség**

Az osztály az FFP2-es maszk tárgy viselkedését valósítja meg. felelőssége, hogy a hallgatók az osztály felhasználásával a védetséget kapjanak a mérges gáz hatása ellen. A hallgatók status attribútuma változik az osztály felhasználásával.

- **Össztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatának a számát jelzi, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**
- Nem vesz részt asszociációban.
- **Metódusok**
- **+void effect()** : A metódus a hallgató állapotát beállítja mérges gáz ellen védetté (PoisonProtection) így a gáz hatása nem érvényes a hallgatóra.

### 3.3.6 Game

- **Felelősség**

Az osztály a játék indításáért és a körök menedzseléséért felelős, illetve lehetővé teszi a karakterek számára, hogy akciót hajtsanak végre a körön belül (azaz “sorra kerüljenek”). A játék végének kezelése is az osztály felelőssége.

- **Ősosztályok**

Nem rendelkezik őssztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**
- **-characters:** a játékban részt vevő karakterek
- **-labyrinth:** a játétként szolgáló labirintus
- **Asszociációk**
- **egyirányú kapcsolat a Game osztályból a Character osztály felé:** a Game osztály számon tartja a játékban részt vevő karaktereket.
- **kompozíció a Labyrinth osztályból a Game osztály felé:** a Game osztály számon tartja a labirintust, ha a Game megszűnik, akkor az egész labirintus is.
- **Metódusok**
- **+void startGame():** játék indítása, kezdőállapot betöltése.
- **+void nextRound():** következő kör indításának megvalósítása.
- **+void endGame(bool result):** a játék végének kezelése attól függően, hogy ki nyert.
- **+void action(Character character):** a karakterek akciójának megvalósítása.

### 3.3.7 HolyBeerMug

- **Felelősség**

Az osztály a Szent söröspohár tárgy viselkedését valósítja meg. Felelőssége, hogy ha egy hallgató használja ezt tárgyat, akkor az megvédje őt egy adott ideig attól, hogy egy oktató kiszívja a lelkét.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**
- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatának a számát jelez, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**
- Nem vesz részt asszociációban.

- **Metódusok**
- **+void effect():** A metódus a hallgató állapotát beállítja oktató lélek kiszívása ellen védetté (SoulProtection) így az oktatók támadásai nem hatnak a hallgatóra.

### 3.3.8 Item

- **Felelősség**

A játékban szereplő tárgyaknak egy absztrakt osztályt biztosít. A tárgyak hatáskifejtéséért is felelős.

- **Ősosztályok**

Nem rendelkezik őssosztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**
- **-name:** a tárgy neve.
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)
- **-durability:** a tárgy használatának a számát jelez, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik.
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).
- **-owner:** A karakter aki felvette a tárgyat.

- **Asszociációk**
- **kétirányú kapcsolat az Character osztályból az Item osztály felé:** A karakter tudja milyen tárgyak vannak az eszköztárában, de egyidőben egy karakternél legfeljebb 5 Item lehet.

- **kétirányú kapcsolat a Room osztályból az Item osztály felé:** A szobában elérhetőek a benne lévő tárgyak. Egy szobában akármennyi tárgy lehet.

- **Metódusok**

- **+void effect() :** A tárgy hatását kifejtő függvény (absztrakt).
- **+ void connect(Item item):** Transistorok-nál ez a metódus felel azért, hogy összekapcsolódjanak, a többi Item esetében ez a metódus nem csinál semmit.
- **+void getOwner():** Visszaadja a tárgy tulajdonosát.
- **+ void onPickUp():** Tárgy felvételénél lefutó metódus, ami csak a “SlipStick”, illetve a “Transistor” nevű tárgyaknál fejt ki hatást.
- **+int getDurability():** a “durability” attribútum értékét kapjuk vissza ezen getter metódus által.
- **+void setDurability(int durability):** ezzel a setter metódussal vagyunk képesek a “durability” tagváltozó értéket beállítani.
- **+bool isCurrentlyUsed() :** a “currentlyUsed” attribútum értékét kapjuk vissza ezen getter metódus által.
- **+void setCurrentlyUsed(bool currentlyUsed) :** ezzel a setter metódussal vagyunk képesek a “currentlyUsed” tagváltozó kezdőértéket beállítani.
- **+Room getItemLocation():** Visszaadja, hogy a tárgy éppen hol van.
- **+void setItemLocation(Room room):** Beállítja a tárgy melyik szobában van.
- **+void setOwner(Character character):** a tárgy tulajdonosát állítja be.
- **+void destruct():** a tárgy elégtételt valósítja meg.

### 3.3.9 KnockedOut

- **Felelősség**

A karakterek eszméletvesztés állapotának beállítására szolgál.

- **Össztályok**

CharacterProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- Nincsen attribútuma.

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok**

- **+void effect(bool st, bool p, bool k, bool so):** A karakterhez tartozó állapotokat jelző értékeket beállítja.

### 3.3.10 Labyrinth

- **Felelősség**

A játéktér létrehozásáért és a szobák nyilvántartásáért felelős.

- **Őszosztályok**

Nem rendelkezik őszosztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-playground** : A szobákat és szomszédait tartja nyilván.

- **Asszociációk**

Nem vesz részt asszociációban.

- **Metódusok**

- **+void initialize()**: A labirintus felépítését, a kezdeti állapotának beállítását valósítja meg.

### 3.3.11 PoisonProtection

- **Felelősség**

A karakterek mérgezés elleni védettségének beállítására szolgál.

- **Őszosztályok**

CharachterProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- Nincsen attribútuma.

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok**

- **+void effect(bool st, bool p, bool k, bool so)**: A karakterhez tartozó állapotokat jelző értékeket beállítja.

### 3.3.12 Poisonous

- **Felelősség**

A mérgező tulajdonság rendelhető segítségével a szobákhoz.

- **Össztályok**

RoomProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- Nincsen attribútuma.

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok:**

- + **void effect(bool p, bool c):** A szobához tartozó típusokat jelző értékeket beállítja.

### 3.3.13 Room

- **Felelősség**

A labirintust felépítő szobák és azok viselkedésének megvalósításáért felelős.

- **Össztályok**

Nem rendelkezik össztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-capacity:** A szobában tartózkodó karakterek maximális számát adja meg.
- **-roomType:** A szoba képességet meghatározó típusát/típusait tárolja.
- **-characters:** A szobában lévő karaktereket tartalmazza.
- **-neighbours:** A szoba szomszédait tárolja
- **-items:** A szobában lévő tárgyakat tartalmazza.
- **-poisonous:** A szoba mérgezett állapotát tároló érték.
- **-cursed:** A szoba elátkozott állapotát tároló érték.

- **Asszociáció**

- **egyirányú kapcsolat a Room osztályból a RoomProperty osztály felé:** A szoba tudja magáról milyen képességet adó típusa/típusai vannak, amelyekből legfeljebb 2 van neki.

- **kétirányú kapcsolat a Room osztályból az Item osztály felé:** A szobában elérhetőek a benne lévő tárgyak. Egy szobában akár mennyi tárgy lehet.
- **kétirányú kapcsolat a Room és a Character osztályok közt:** A karakterek tudják melyik szobában vannak, a szobák pedig azt hogy mely karakterek tartózkodnak bennük. Egy karakter egy időben egy szobában tartózkodhat. Egy szobában egy időben egyszerre a szoba befogadóképességével megegyező karakterek lehet.
- **kompozíció a Room osztályból a Labyrinth osztály a felé:** A labirintus szobákból áll, egy szoba egy labirintushoz tartozhat. Ha a labirintus megsemmisül akkor a tartalmazott szobák is megszűnnek. Egy labirintus akárhány szobából állhat.
- **asszociáció a Room osztállyal:** A szobák ismerik a szomszédaikat.
- **Metódusok**
  - **+void putItem(Item item) :** A szobában új tárgy helyeződik el.
  - **+void deleteItem(Item item):** A szobából egy - a szobában lévő- tárgyat felvettek a karakterek.
  - **+List<Room> getNeighbours () :** Visszaadja a szoba szomszédait.
  - **+void setNeighbours(List<Room> roomList):** Visszaadja a szomszédos szobák listáját.
  - **+ List<RoomProperty> getProperties() :** A szoba állapota(i) kérdezhető le
  - **+ void addProperty(RoomProperty property):** A szoba állapotához lehet rendelni új tulajdonságot
  - **+void getCharacters():** Visszaadja a szobában tartózkodó karakterek listáját.
  - **+void setCharacters():** Beállítja a szoba karakterlistáját.
  - **+void getCapacity():** Visszaadja azt a számot, ami megmutatja, hogy az adott szobában legfeljebb hányan tartózkodhatnak egyidőben.
  - **+void addDoor(Room room) :** Megjelenik egy ajtó a paraméterként kapott szoba felé
  - **+void deleteDoor(Room room) :** Eltűnik egy ajtó a paraméterként kapott szoba felé
  - **+void roomSplit() :** Egy szoba osztódását valósítja meg.
  - **+void roomMerge(Room room) :** Két szoba egyesülését valósítja meg.
  - **+bool IsPoisonous() :** A szoba mérgezett állapotát adja vissza.
  - **+bool IsCursed() :** A szoba elátkozott állapotát adja vissza.
  - **+void deleteNeighbour(Room room):** Törli a szoba egyik ajtóját.
  - **+void addNeighbour(Room room):** A paraméterként kapott szoba a szoba szomszédja lesz.
  - **+void addCharacter(Character character):** A szoba karakterlistájához hozzáadódik a paraméterként kapott karakter.
  - **+void deleteCharacter(Character character):** A szoba karakterlistájából törlődik a paraméterként kapott karakter.

### 3.3.14 RoomProperty

- **Felelősség**

Olyan absztrakt osztály, melynek felelőssége az, hogy a segítségével a szobák típusait nyilván tudjuk tartani.

- **Ősosztályok**

Nem rendelkezik ősosztállyal.

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **Asszociáció**

- **egyirányú kapcsolat a Room osztályból a RoomProperty osztály felé:** A szoba tudja magáról milyen képességet adó típusa/típusai vannak. Egy szobának egy időben egyszerre több típusa is lehet.

- **Metódusok**

- **+ void effect(bool p, bool c):** Beállítja a paraméterként kapott változókat, melyek a szoba tulajdonságait tükrözik.

### 3.3.15 SlipStick

- **Felelősség**

Az osztály a Logarléc tárgy viselkedését valósítja meg. Felelőssége, hogy ha a hallgatók eszköztárába kerül a Logarléc nevű tárgy, akkor sikeresen teljesítették a küldetésüket és megnyerték a játékot.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-durability:** a tárgy használhatóságát a számát jelez, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).



- **Asszociációk**
- **kétirányú kapcsolat a SlipStick osztályból a Game osztály felé:** A Logarléc ismeri a játékot, és a játék tudja mi van a Logarléccel.
- **Metódusok**
- **+void effect() :** Ez a metódus akkor fut le, ha egy hallgató eszköztárába kerül a Logarléc nevezetű tárgy, és ennek hatására megnyerik a hallgatók a játékot.
- **+void onPickUp():** Ha ezt a tárgyat felveszi egy karakter, akkor lép működésbe ez a metódus. Ha egy hallgató veszi fel ezt a tárgyat, akkor győzelemmel zárul a játék, ha egy oktató, akkor pedig letételre kerül máshol és nem ér véget a játék.
- **+void destruct():** biztosítja, hogy a Logarléccet ne lehessen elégetni.

### 3.3.16 SoulProtection

- **Felelősség**  
A karakterek oktató lélek kiszívás elleni védettségének beállítására szolgál.

- **Ősosztályok**

CharacterProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**
- Nincsen attribútuma.

- **Asszociációk**
- Nem vesz részt asszociációban.

- **Metódusok**
- **+void effect(bool st, bool p, bool k, bool so):** A karakterhez tartozó állapotokat jelző értékeket beállítja.

### 3.3.17 Student

- **Felelősség**

Ez az osztály felel a hallgatók viselkedésének a megvalósításáért.

- **Össztályok**

Character

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a hallgató neve (örökölt attribútum).
- **-location:** Azon szoba amelyben a karakter tartózkodik (örökölt attribútum).
- **-inventory:** A karakternél lévő tárgyak tárolódnak itt (örökölt attribútum).
- **-stunned:** A karakter bénított hatását jelző érték (örökölt attribútum).
- **-knockedOut:** A karakter eszméletvesztett hatását jelző érték (örökölt attribútum).
- **-poisonProtection:** A karakter mérge elleni védetségét jelző érték (örökölt attribútum).
- **-soulProtection:** A karakter oktató lélek kiszívása elleni védetségét jelző érték (örökölt attribútum).
- **-characterPropterties:** A karakterre érvényes hatások tárolja (örökölt attribútum).

- **Asszociációk**

Nem vesz részt asszociációban.

- **Metódusok**

- **+void move() :** A hallgató mozgását megvalósító metódus.
- **+void pickUp(Item item) :** A függvény segítségével képes egy hallgató felvenni a szobában lévő tárgyat, így a tárgy bekerül az eszköztárába.
- **+void drop(Item item) :** A függvény segítségével képes egy hallgató, az eszköztárából a szobába letenni a kiválasztott tárgyat.
- **+void discard(Item item) :** A függvény segítségével képes egy hallgató, az eszköztárából elégetni a kiválasztott tárgyat.
- **+void death():** Ha a hallgatónak a lelkét kiszívják, akkor ez a metódus felel a hallgató játékból való kieséséért.
- **+void useItem(Item item):** Ennek a függvénynek a segítségével képes egy hallgató, az eszköztárában szereplő tárgyak közül egyet használni.
- **+void kill():** Mivel egy hallgató nem képes lelket kiszívni, emiatt ez a metódus nem csinál semmit.
- **+void stun():** Mivel egy hallgató nem képes megbénulni, emiatt ez a függvény nem csinál semmit.

### 3.3.18 Stunned

- **Felelősség**

A karakterek bénult állapotának beállítására szolgál.

- **Össztályok**

CharacterProperty

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- Nincsen attribútuma.

- **Asszociációk**

- Nem vesz részt asszociációban.

- **Metódusok**

- **+void effect(bool st, bool p, bool k, bool so):** A karakterhez tartozó állapotokat jelző értékeket beállítja.

### 3.3.19 Teacher

- **Felelősség**

Ez az osztály felel az oktatók viselkedésének a megvalósításáért.

- **Össztályok**

Character

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** az oktató neve (örökölt attribútum).
- **-location:** Azon szoba amelyben a karakter tartózkodik (örökölt attribútum).
- **-inventory:** A karakternél lévő tárgyak tárolódnak itt (örökölt attribútum).
- **-stunned:** A karakter bénított hatását jelző érték (örökölt attribútum).
- **-knockedOut:** A karakter eszméletvesztett hatását jelző érték (örökölt attribútum).
- **-poisonProtection:** A karakter mérge elleni védetségét jelző érték (örökölt attribútum).
- **-soulProtection:** A karakter oktató lélek kiszívása elleni védetségét jelző érték (örökölt attribútum).
- **-characterPropterties:** A karakterre érvényes hatások tárolja (örökölt attribútum).

- **Asszociációk**

Nem vesz részt asszociációban.

- **Metódusok**

- **+void move() :** A oktató mozgását megvalósító metódus.

- **+void pickUp(Item item)** : A függvény segítségével képes egy oktató felvenni a szobában lévő tárgyat, így a tárgy bekerül az eszköztárába.
- **+void drop(Item item)** : A függvény segítségével képes egy oktató, az eszköztárából a szobába letenni a kiválasztott tárgyat.
- **+void discard(Item item)** : A függvény segítségével képes egy oktató, az eszköztárából elégetni a kiválasztott tárgyat.
- **+void death()**: Mivel egy oktató nem tud meghalni, ezért a metódus nem csinál semmit.
- **+void useItem(Item item)**: Mivel egy oktató nem tud tárgyat használni a metódus nem csinál semmit.
- **+void kill()**: Az oktató a vele egy szobában lévő hallgatók lelkét kiszívja.
- **+void stun()**: Az oktató megbénításáért felelős függvény.

### 3.3.20 Transistor

- **Felelősség**

Az osztály a Tranzistor tárgy viselkedését valósítja meg. Felelőssége, hogy biztosítja azon hallgatók számára a teleportálást, akiknek van két összekötött tranzisztoruk és ezek közül az egyik már azon szobába le van helyezve, ahová a hallgató teleportálni szeretne és használja a másikat.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**
- **-connected**: ezzel az attribútummal jelezzük, hogy egy tranzisztor össze van-e már kötve egy másikkal.
- **-name**: a tárgy neve (örökölt attribútum).
- **-owner**: A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation**: A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability**: a tárgy használatának a számát jelez, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed**: ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).
- **-pair**: összekapcsolt tranzisztor esetén a tranzisztor párja (egy másik tranzisztor)
- **Asszociációk**
- **asszociáció a Transistor osztállyal**: összekapcsolt tranzisztorok esetén a tranzisztor párja egy másik tranzisztor. Ezt a kapcsolatot fejezi ki ez az asszociáció.

- **Metódusok**
- **+void connect(Item item):** ezzel a függvénnyel kapcsoljuk össze a tranzistorunkat.
- **+void OnPickup():** Ez a metódus felel azért, hogy ha egy hallgatónál már van egy Transistor, akkor egy másik felvételénél a párosodás teljesüljön.
- **+void effect() :** ezen metódussal történik meg a hallgató elteleportálása a már letett és összekapcsolt tranzistorhoz.

### 3.3.21 TVSZ

- **Felelősség**

Az osztály a “TVSZ denevérbőrre nyomtatott példánya” tárgy viselkedését valósítja meg. Felelőssége, hogy ha egy hallgató használja a “TVSZ denevérbőrre nyomtatott példánya” nevű tárgyat, akkor az megvédje őt 3-szor attól, hogy egy oktató kiszívja a lelkét.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**
- **-name:** a tárgy neve (örökölt attribútum).
- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).
- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).
- **-durability:** a tárgy használatainak számát jelzi, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).
- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**

Nem vesz részt asszociációban.

- **Metódusok**
- **+void effect() :** A metódus a hallgató állapotát beállítja oktató lélek kiszívása ellen védetté (SoulProtection) így az oktatók támadásai nem hatnak a hallgatóra.

### 3.3.22 WetTowel

- **Felelősség**

Az osztály a “Nedves táblatörlő rongy” tárgy viselkedését valósítja meg. Felelőssége, hogy a hallgató a használata során képes legyen egy oktatót megbénítani, és ez által kisebb előnyre tegyen szert a játék során.

- **Ősosztályok**

Item

- **Interfészek**

Az osztály nem valósít meg interfészt.

- **Attribútumok**

- **-name:** a tárgy neve (örökölt attribútum).

- **-owner:** A tárgyak birtokló játékos (örökölt attribútum).

- **-itemLocation:** A szoba ahol jelenleg a tárgy van (felvétel után a karakter szobájával egyezik meg)(örökölt attribútum).

- **-durability:** a tárgy használatának a számát jelez, amely hogyha eléri a 0-át, akkor a tárgy használhatatlanná válik (örökölt attribútum).

- **-currentlyUsed:** ezzel az attribútummal vizsgáljuk, hogy az adott tárgy használatban van-e (örökölt attribútum).

- **Asszociációk**

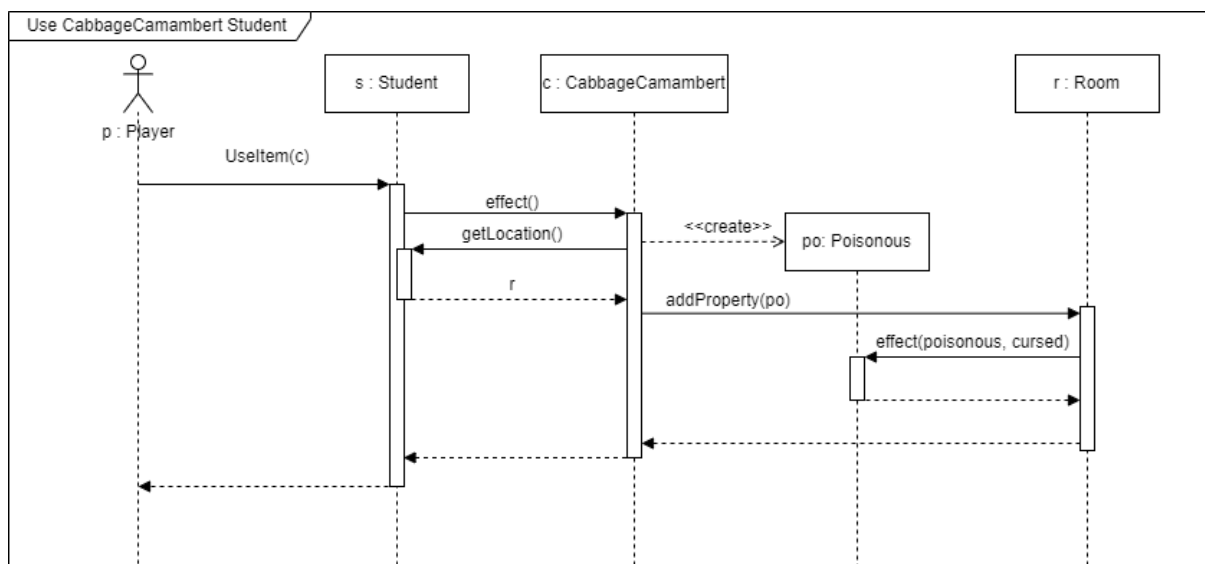
Nem vesz részt asszociációban.

- **Metódusok**

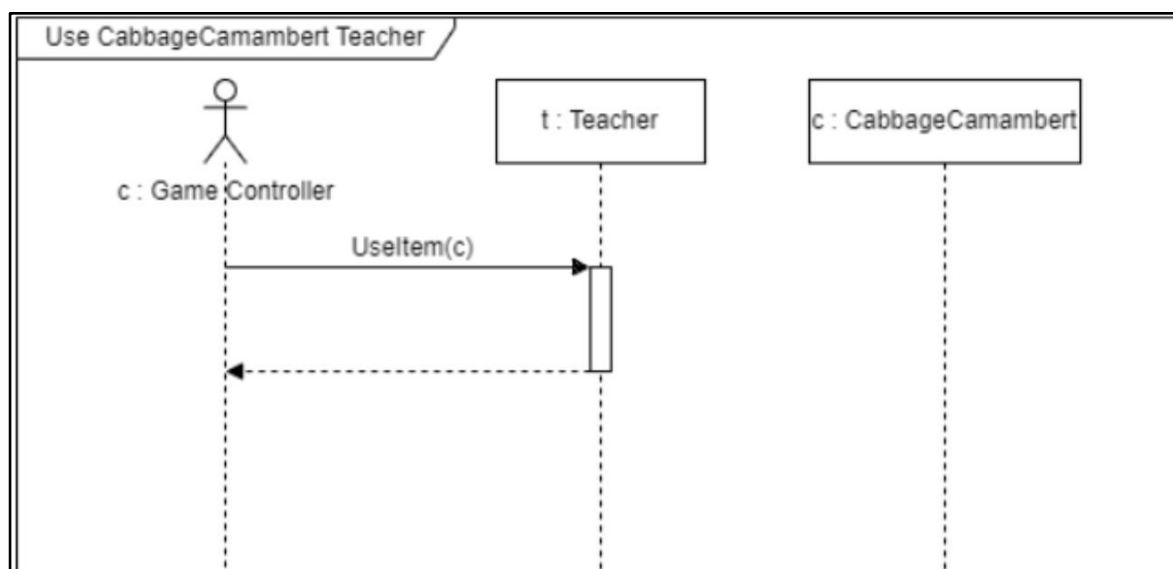
- **+void effect() :** A metódus az a szobában lévő oktatókat lebénítja de a hallgatókra nem hat.

### 3.4 Szekvencia diagramok

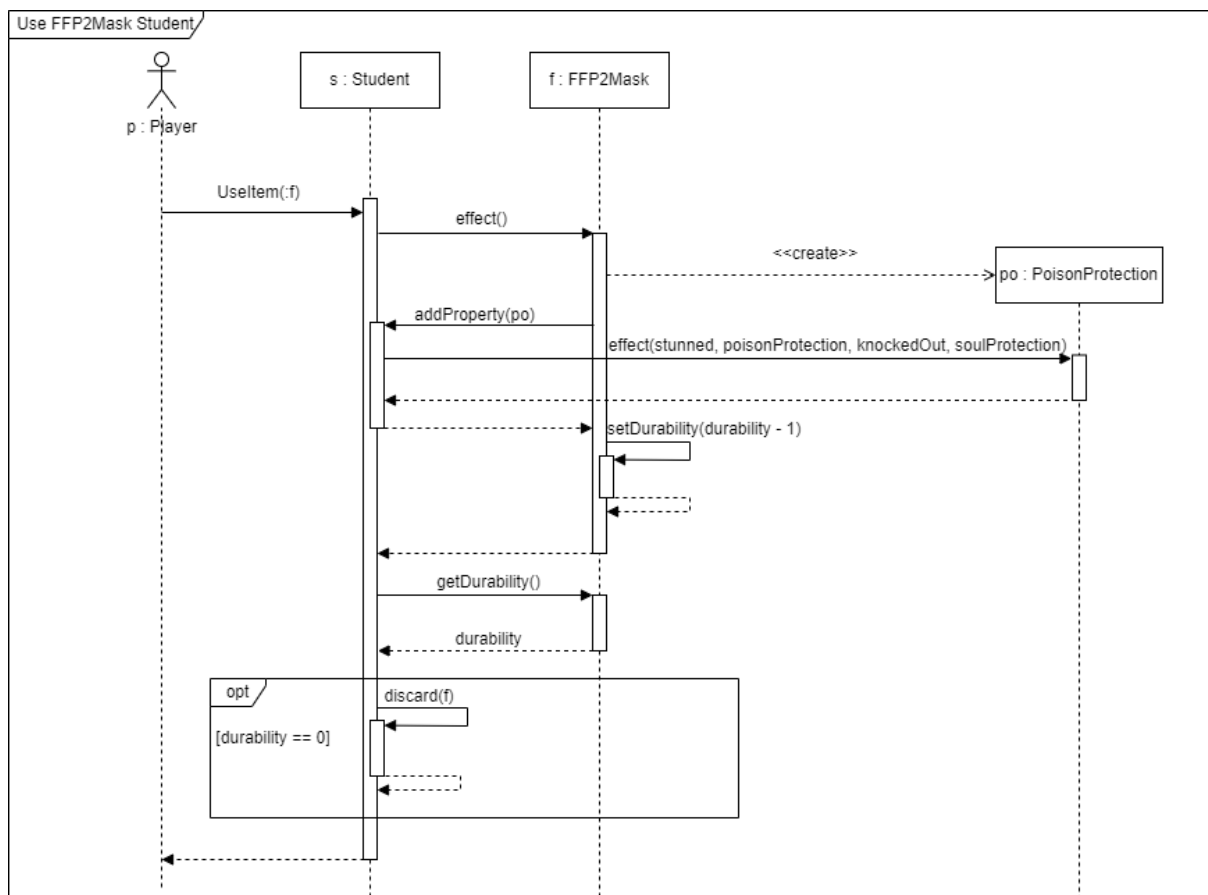
#### 3.4.1 Use CabbageCamambert Student



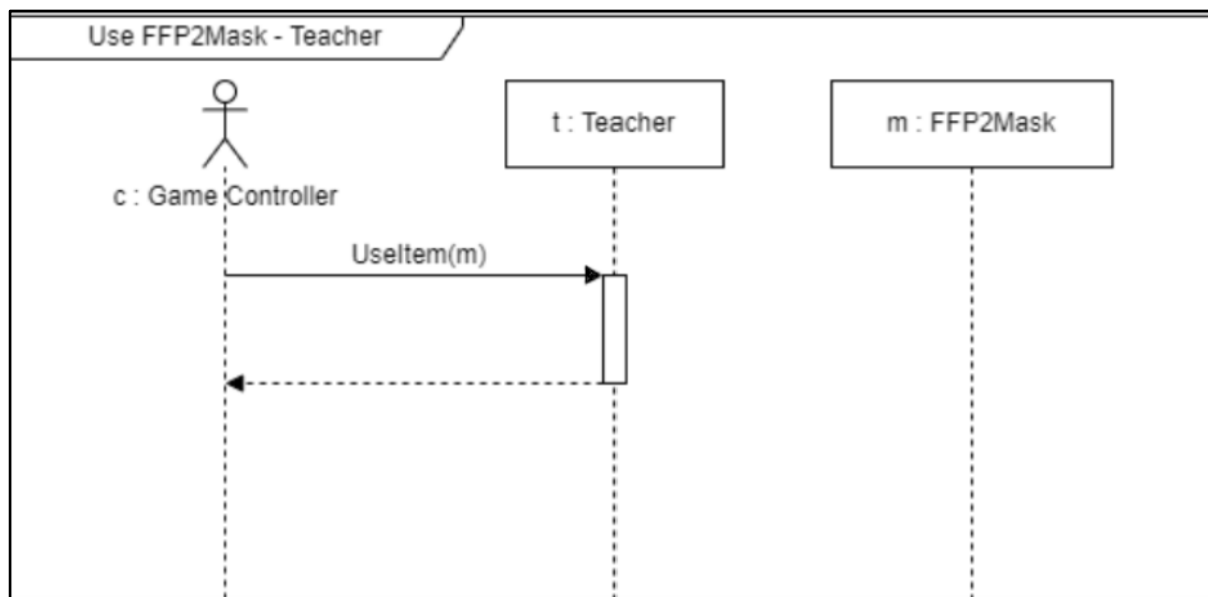
#### 3.4.2 Use CabbageCamambert Teacher



### 3.4.3 Use FFP2Mask Student

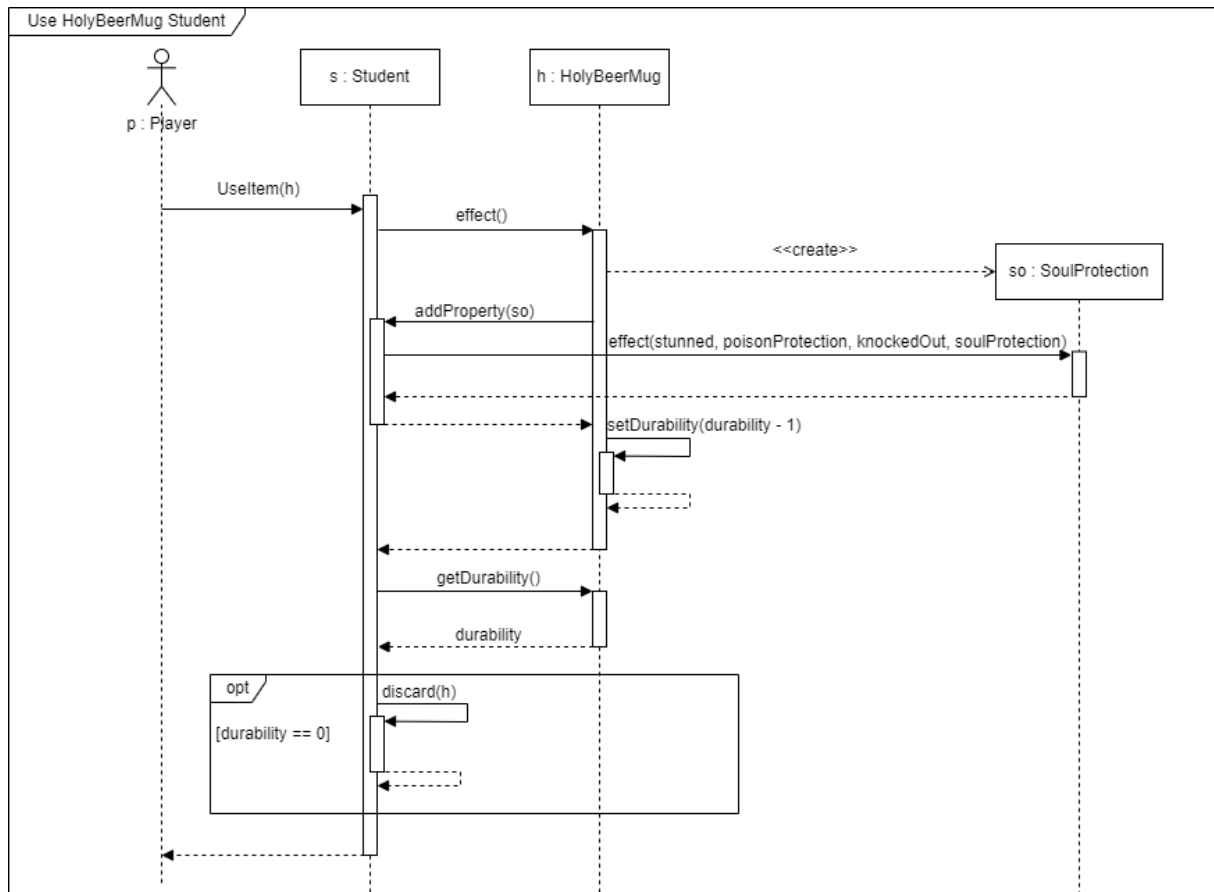


### 3.4.4 Use FFP2Mask Teacher

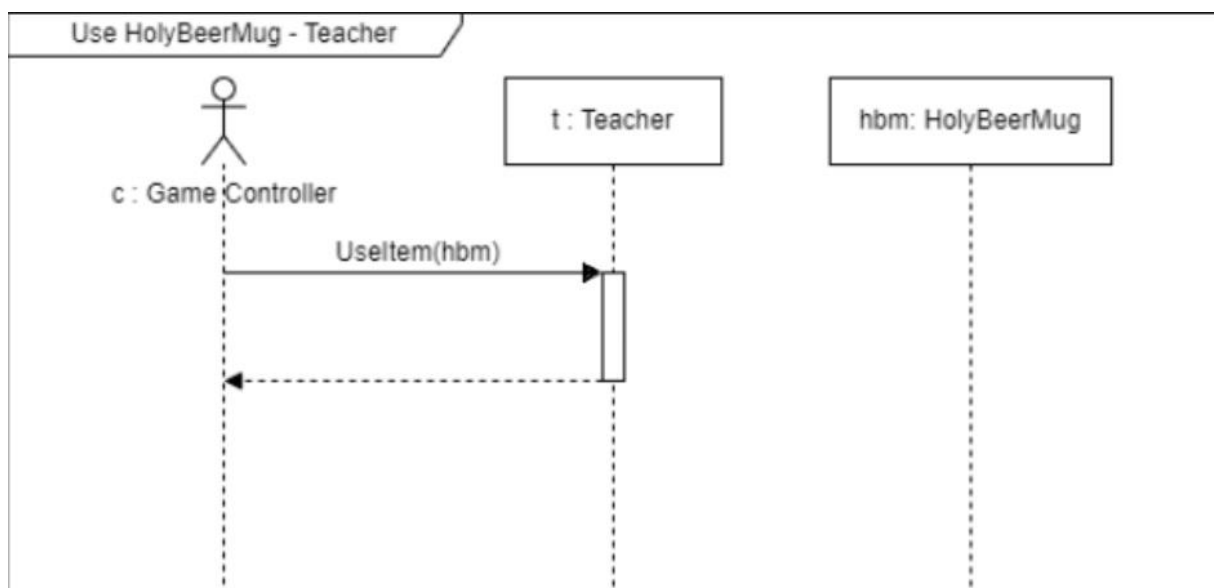




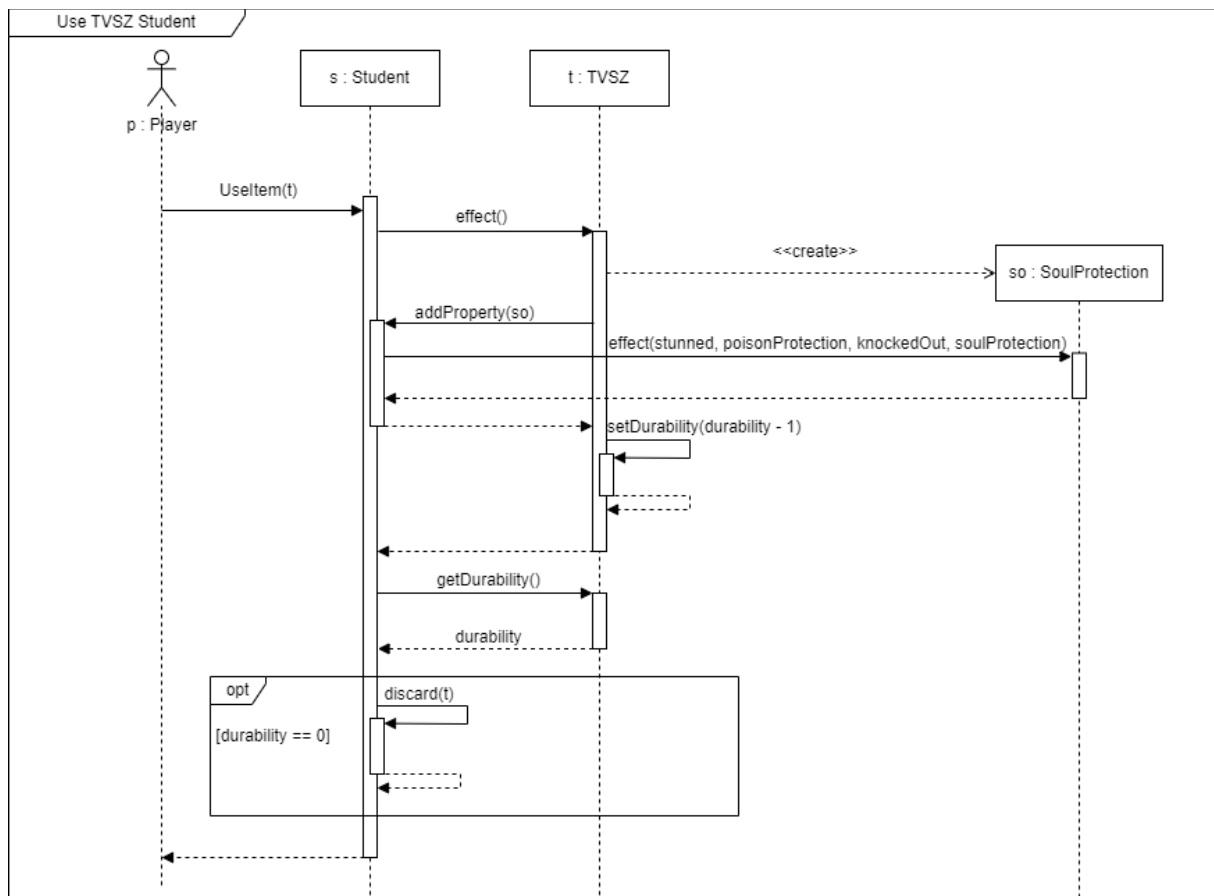
### 3.4.5 Use HolyBeerMug Student



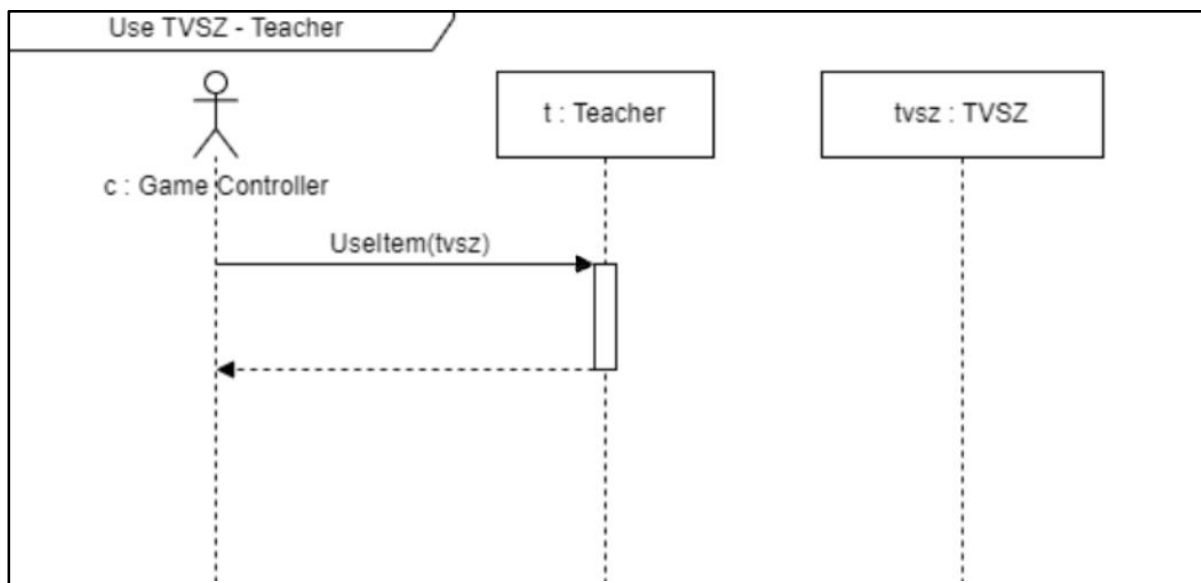
### 3.4.6 Use HolyBeerMug Teacher



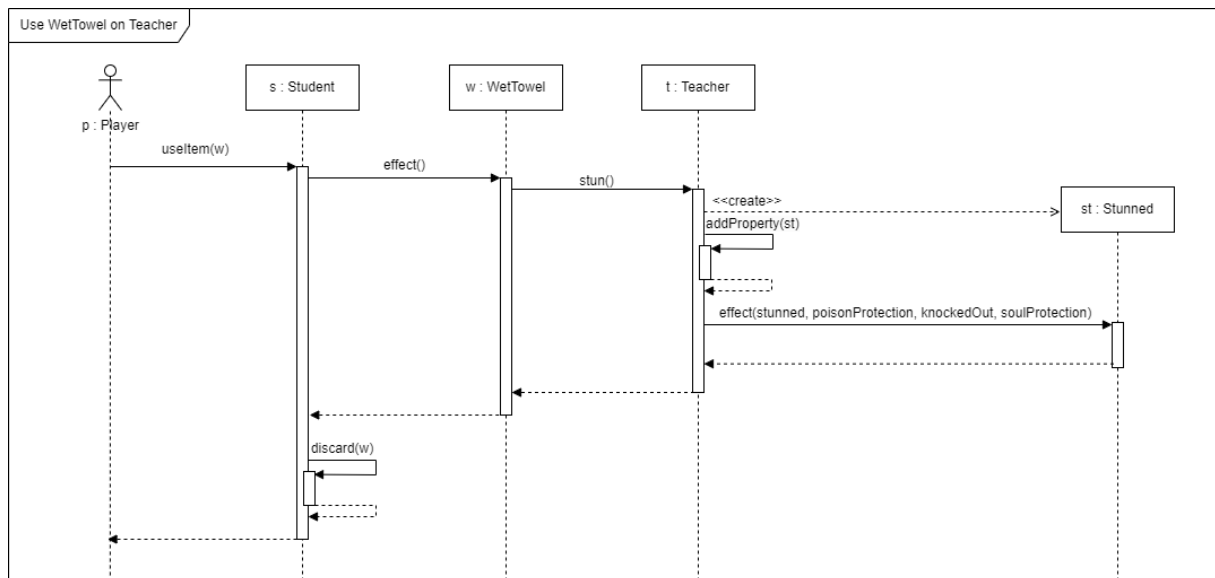
## 3.4.7 Use TVSZ Student



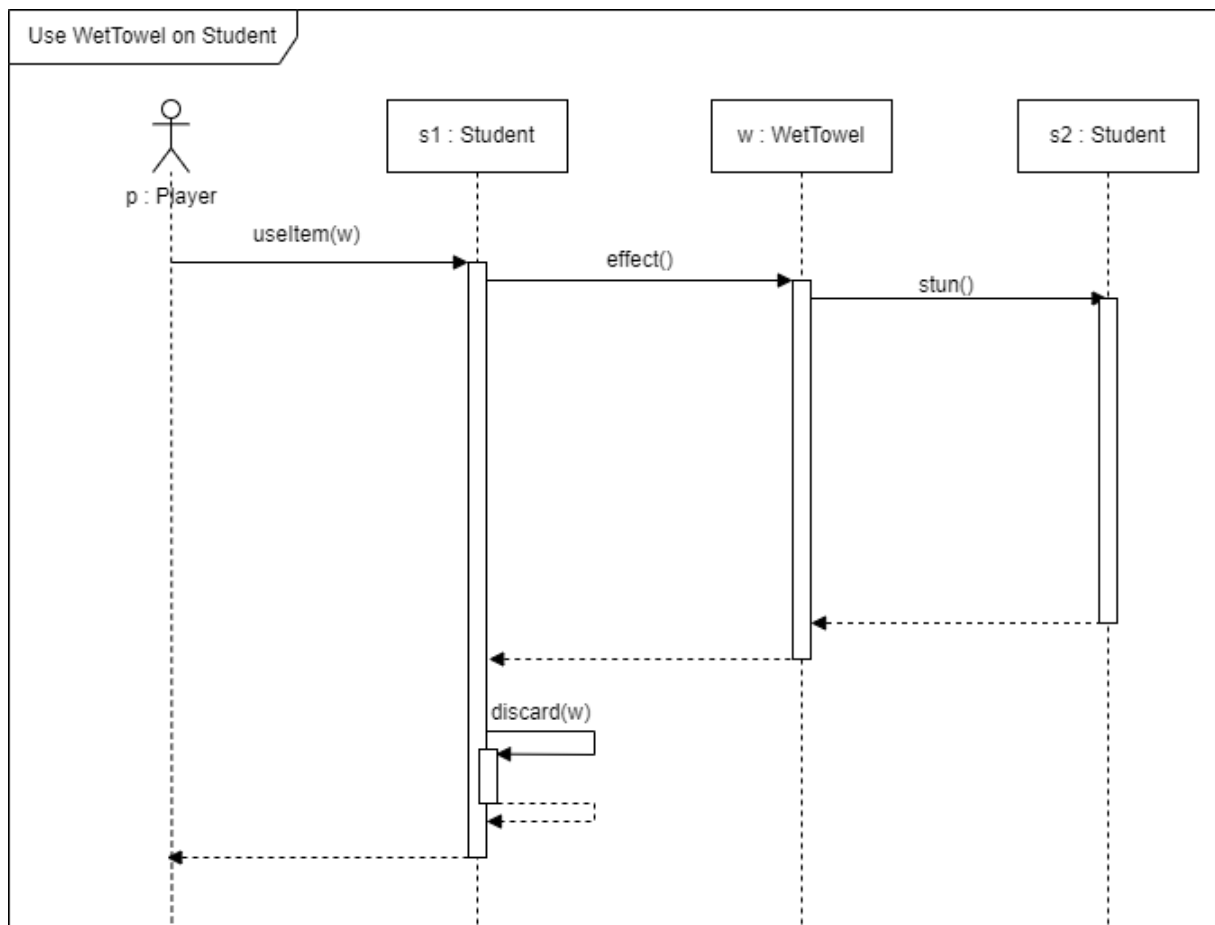
## 3.4.8 Use TVSZ Teacher



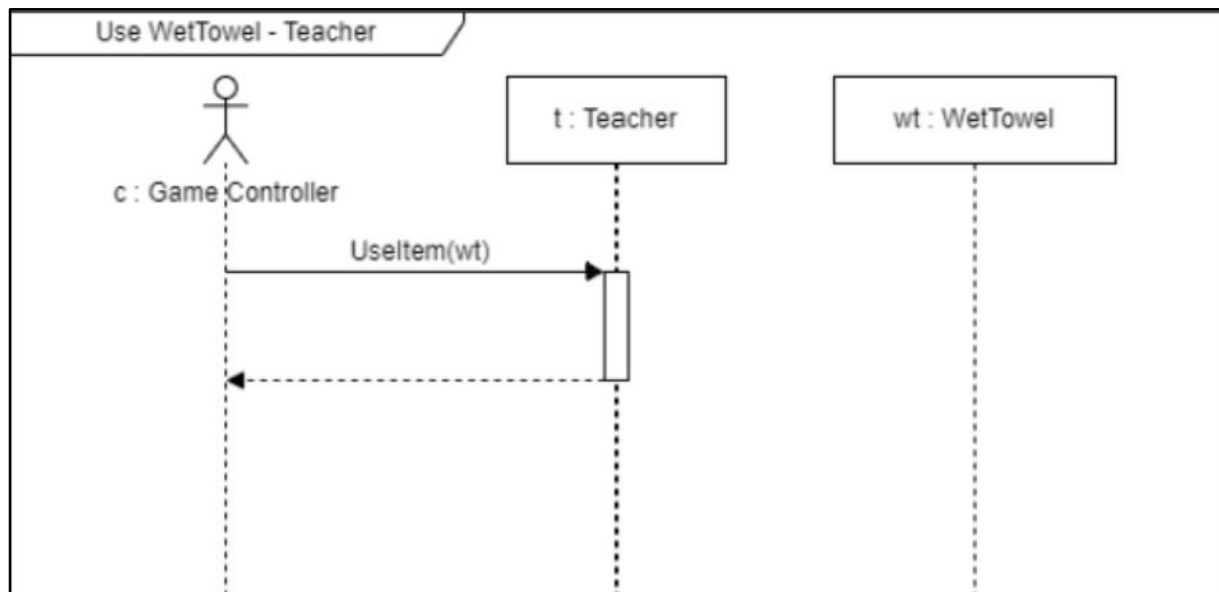
### 3.4.9 Use WetTowel by Student on Teacher



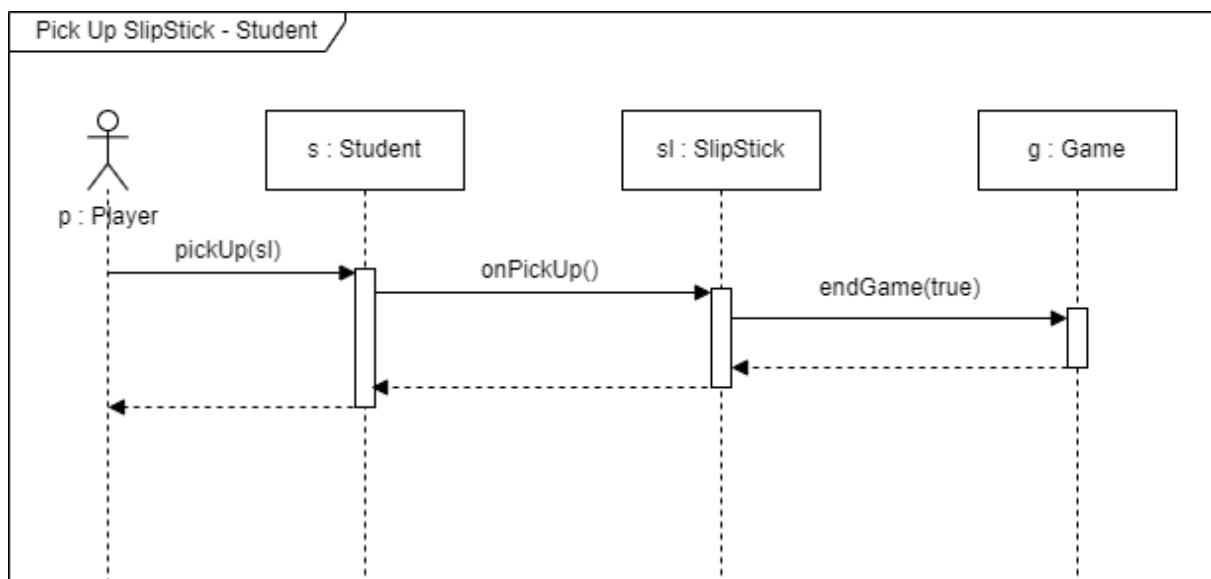
### 3.4.10 Use WetTowel by Student on Student



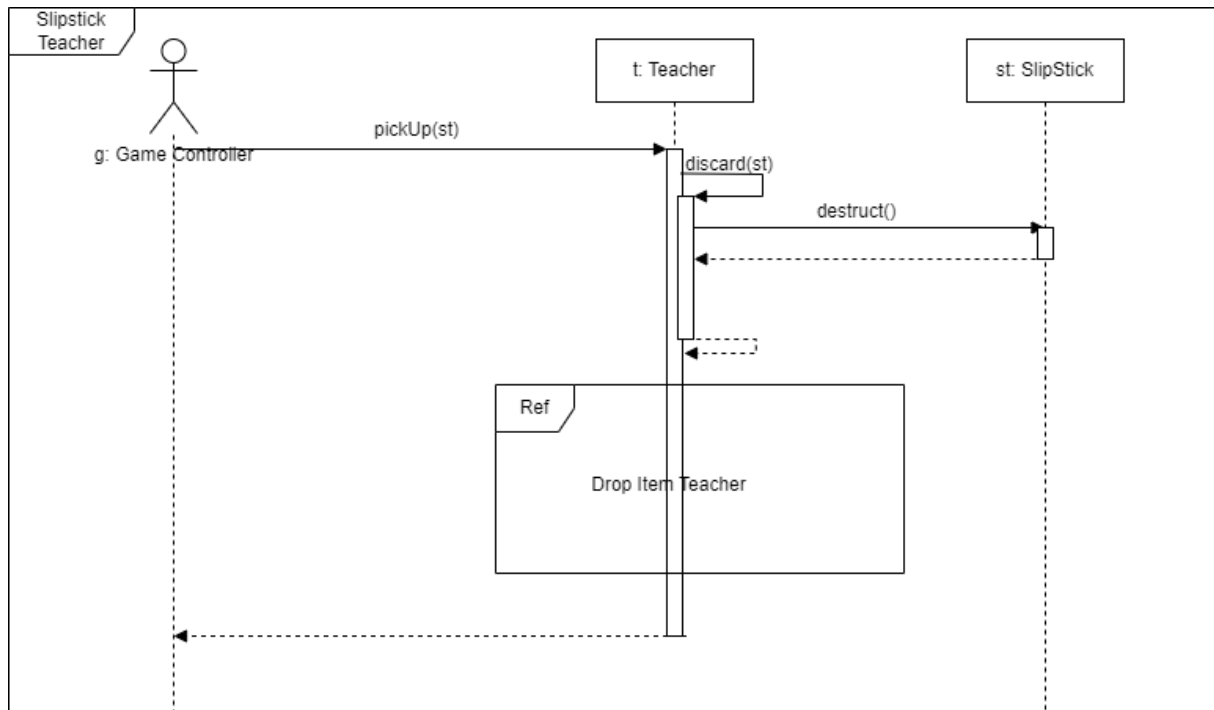
### 3.4.11 Use WetTowel by Teacher



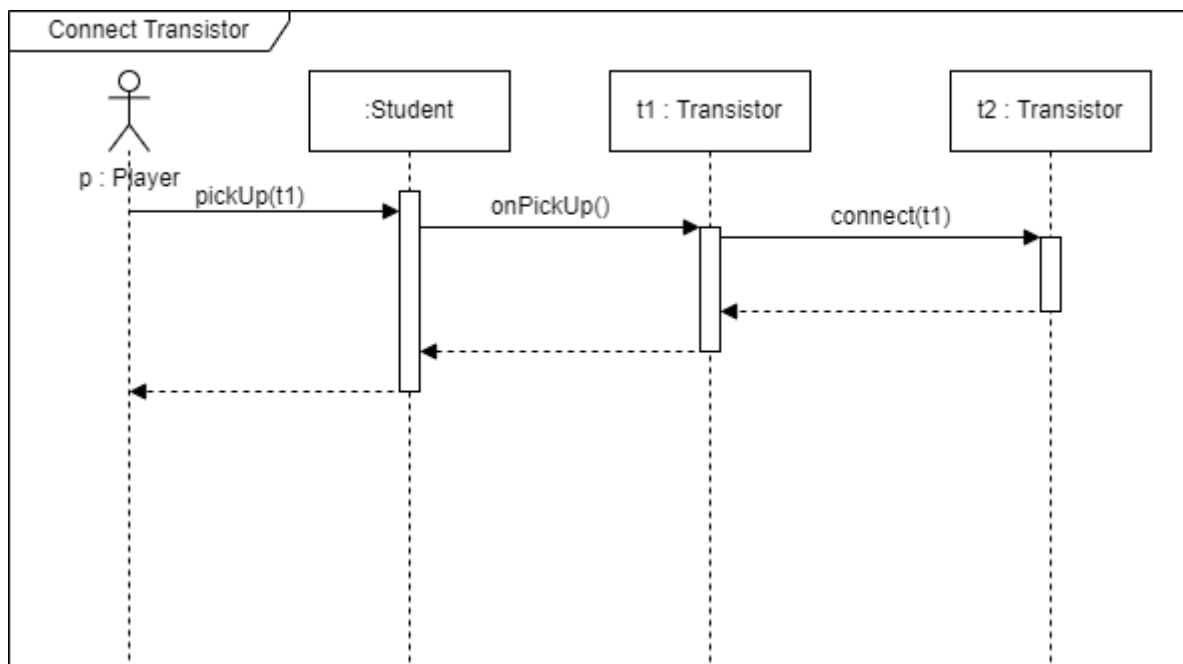
### 3.4.12 Pick Up SlipStick by Student



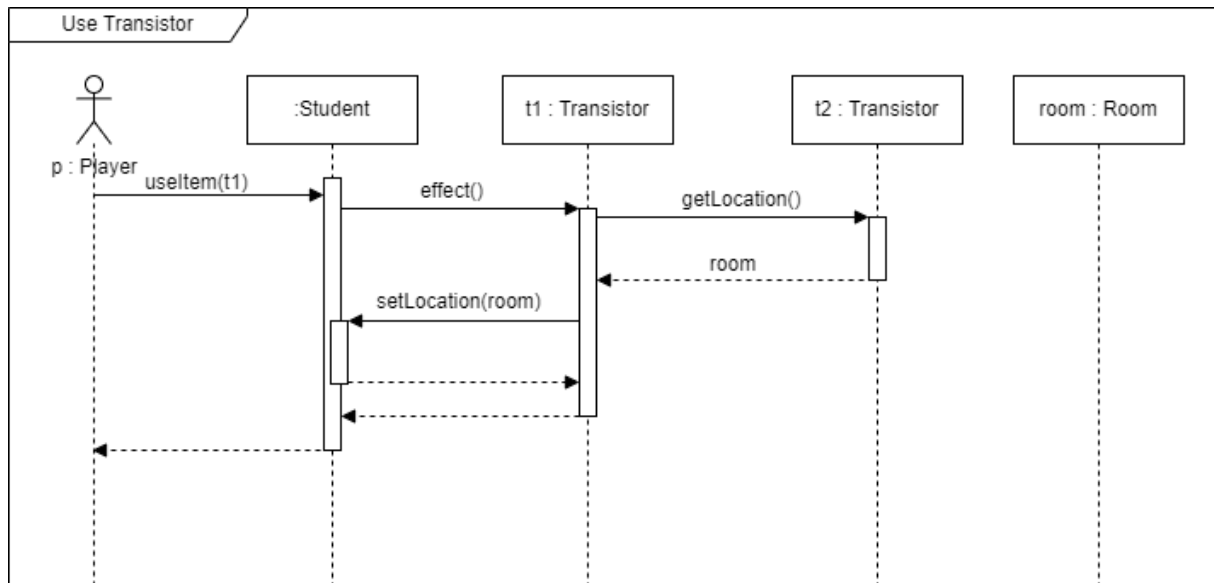
### 3.4.13 Pick Up SlipStick Teacher



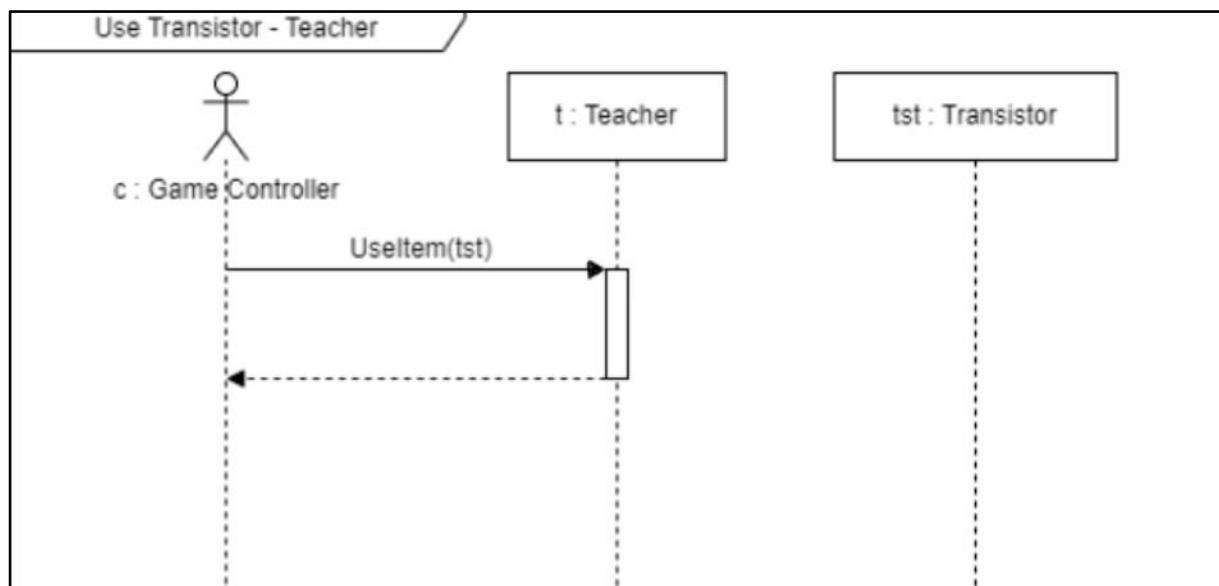
### 3.4.14 Connect Transistor



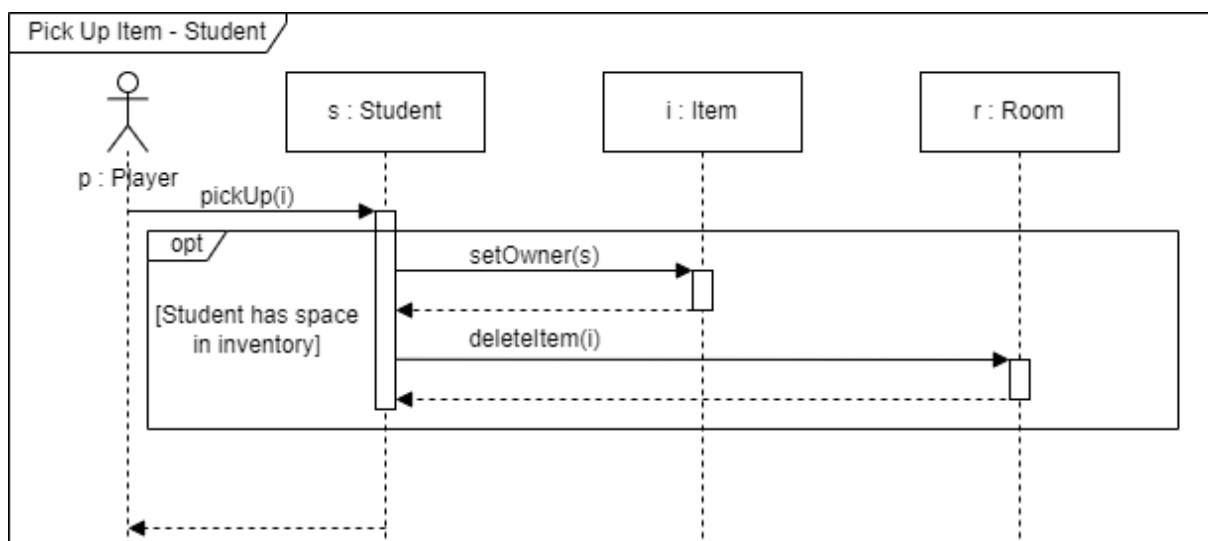
### 3.4.15 Use Transistor by Student



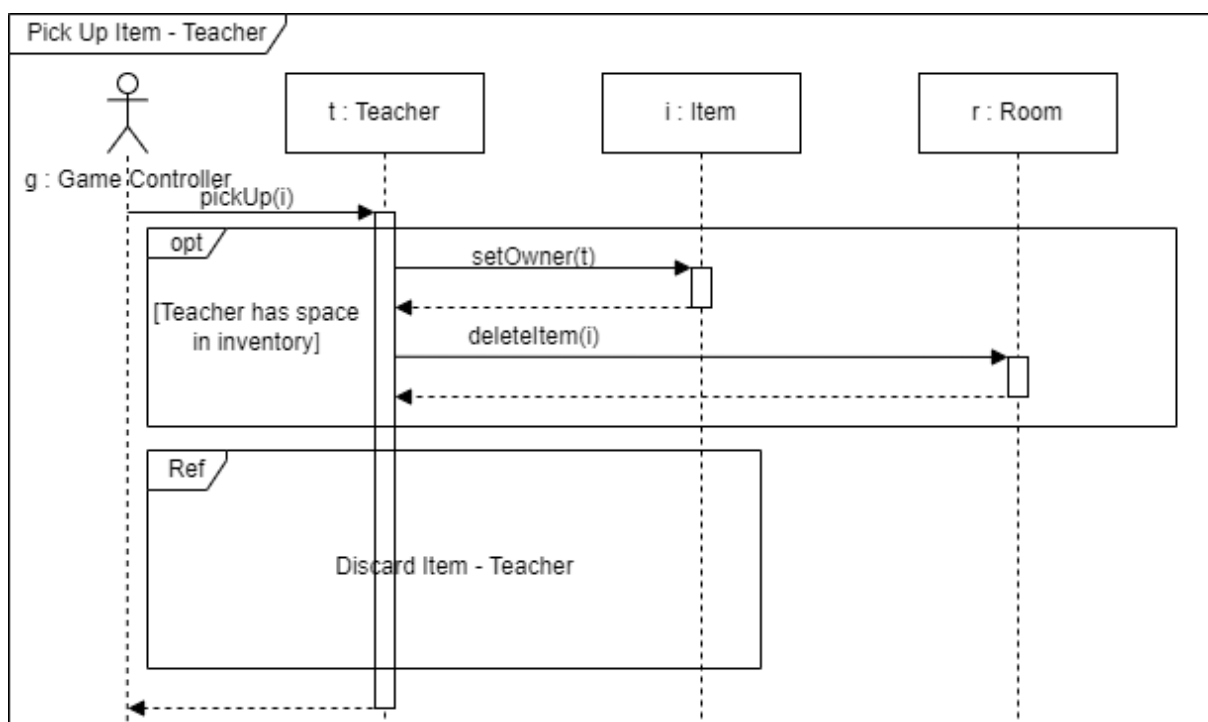
### 3.4.16 Use Transistor Teacher

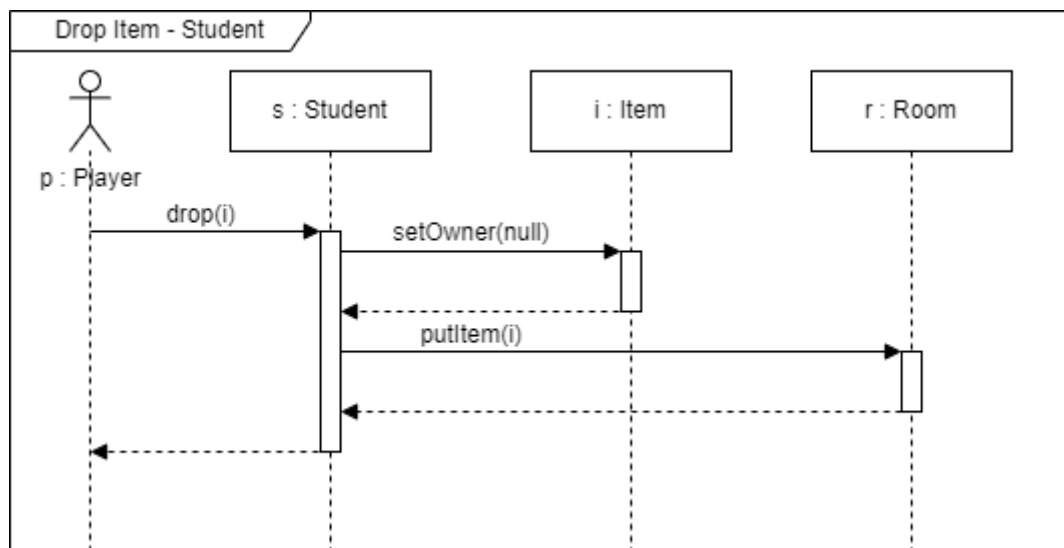
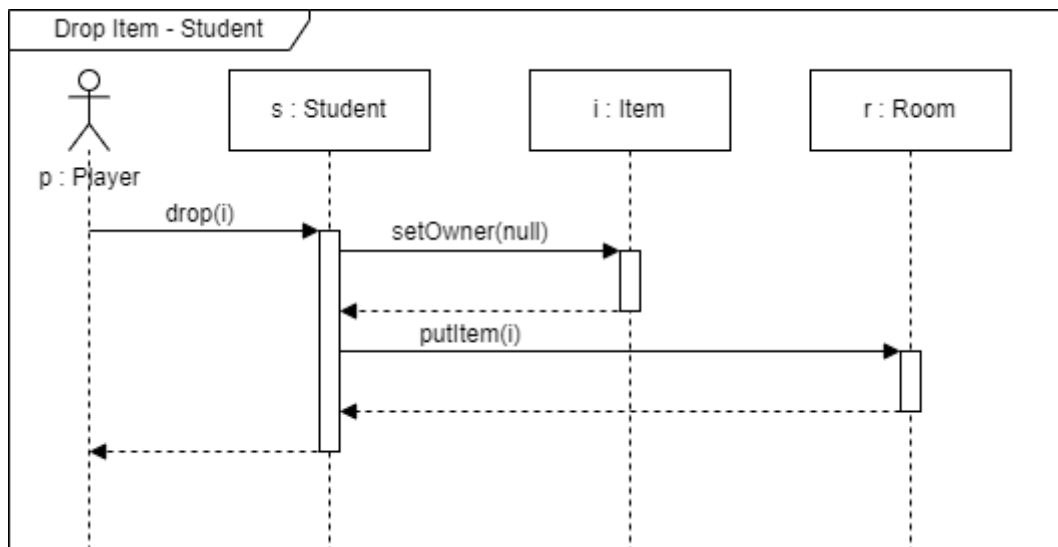


## 3.4.17 Pick Up Item Student

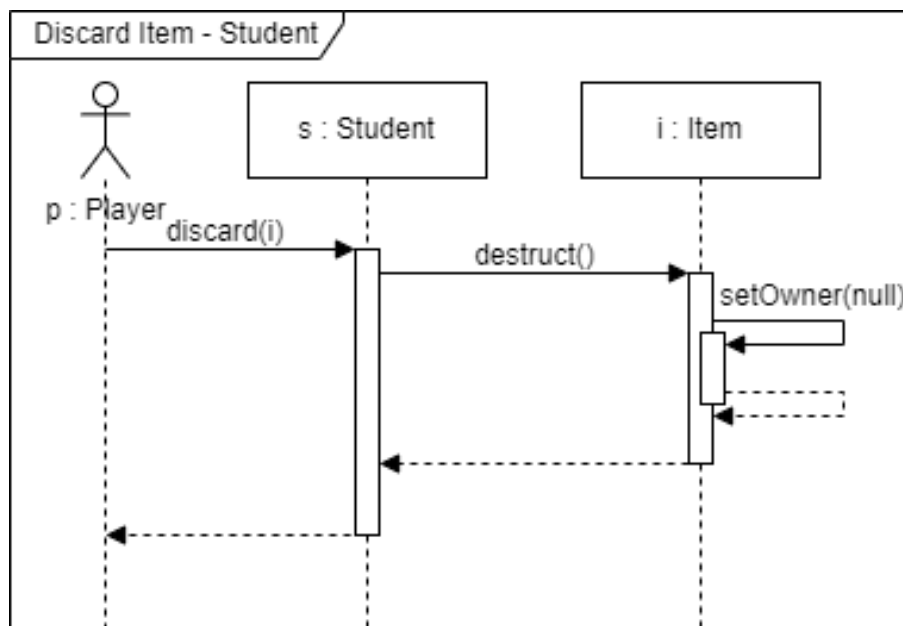
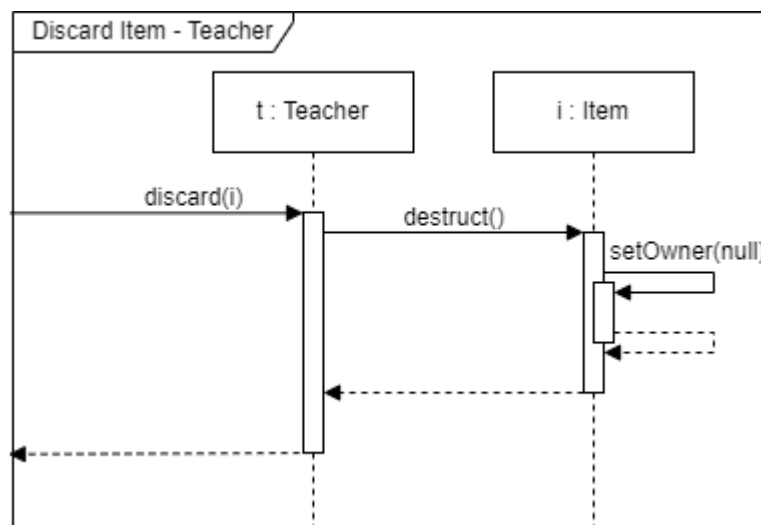


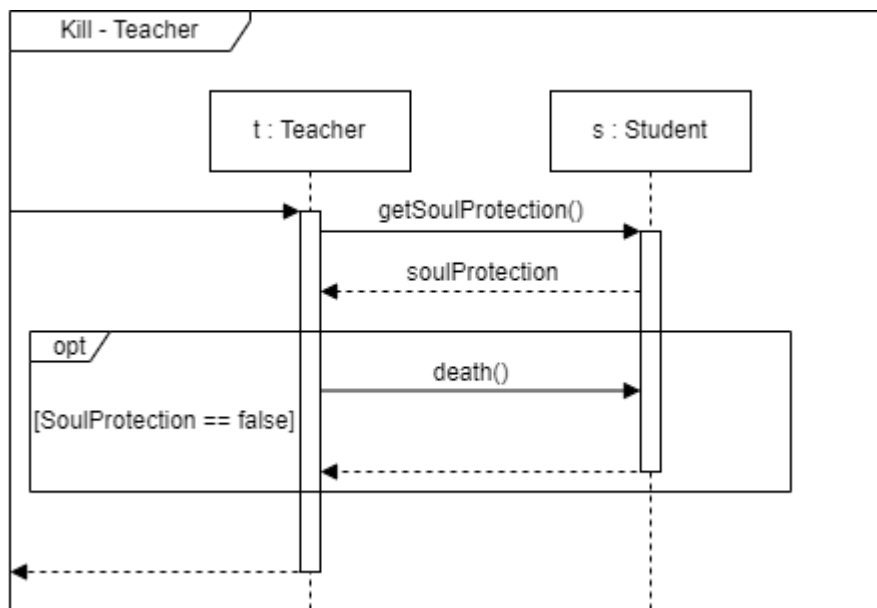
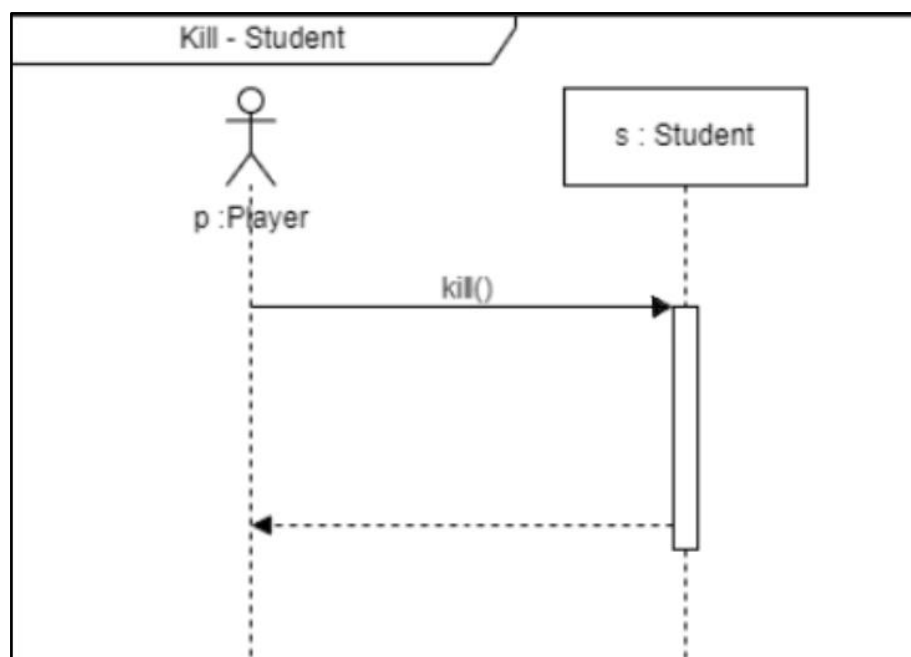
## 3.4.18 Pick Up Item Teacher



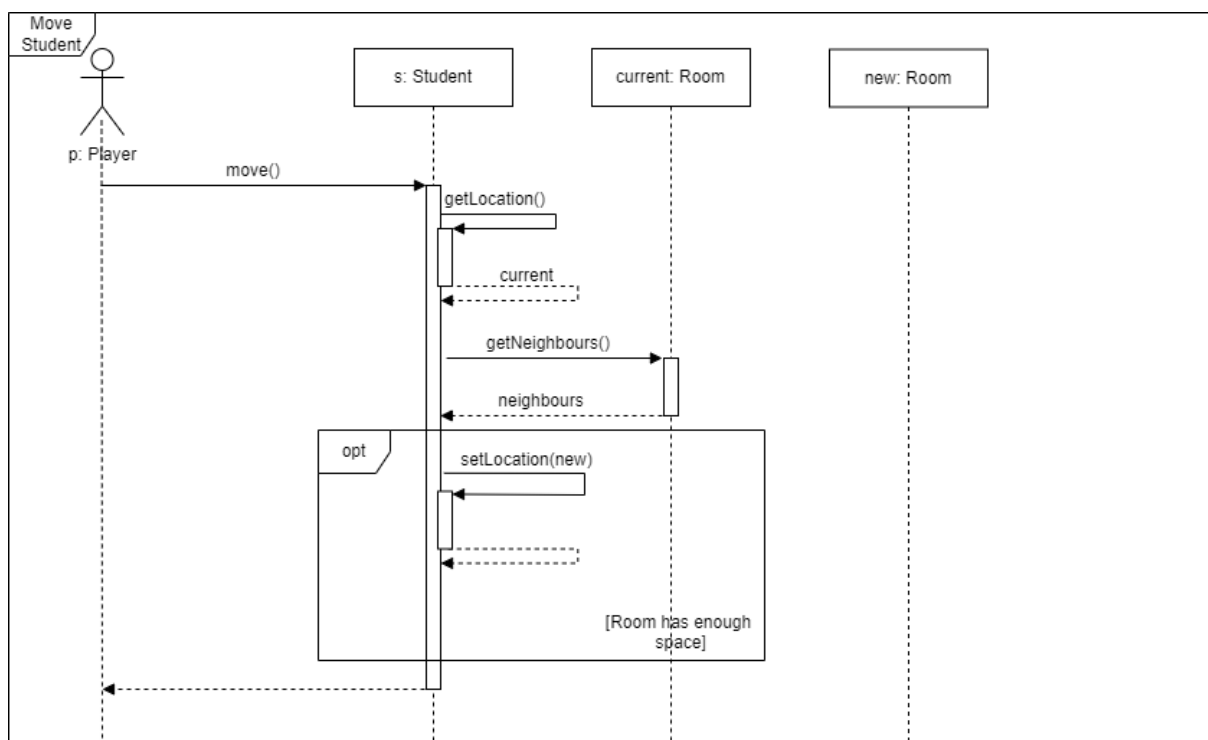
**3.4.19 Drop Item Student****3.4.20 Drop Item Teacher**



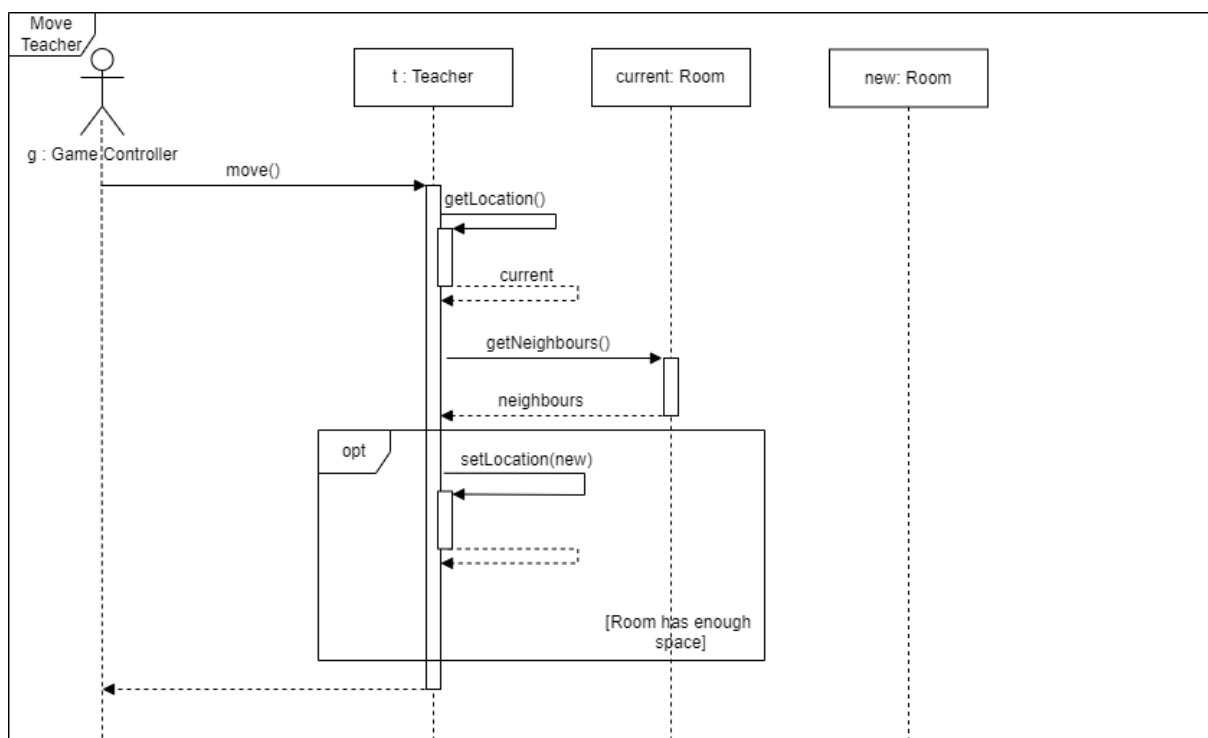
**3.4.21 Discard Item Student****3.4.22 Discard Item Teacher**

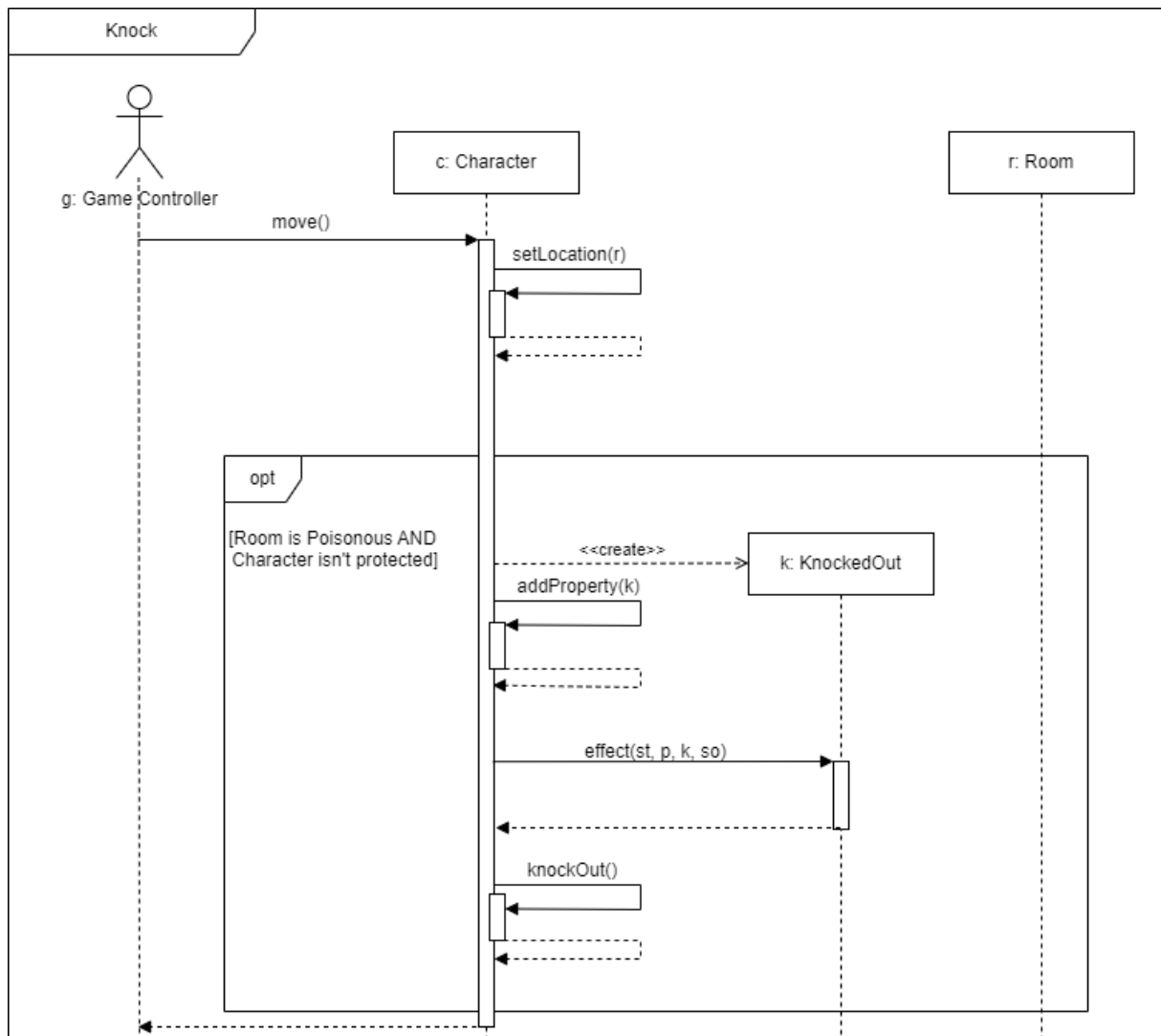
**3.4.23 Kill Student - Teacher****3.4.24 Kill Student - Student**

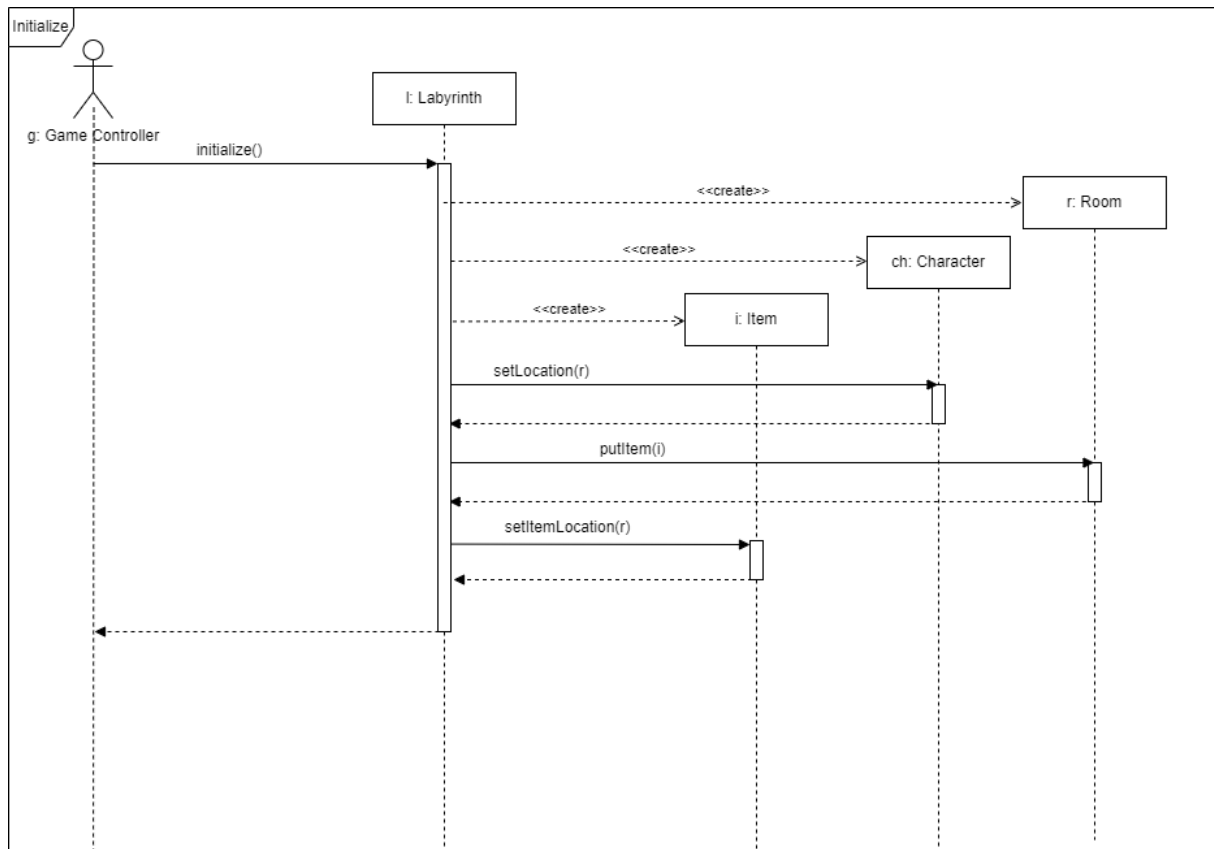
### 3.4.25 Move Student



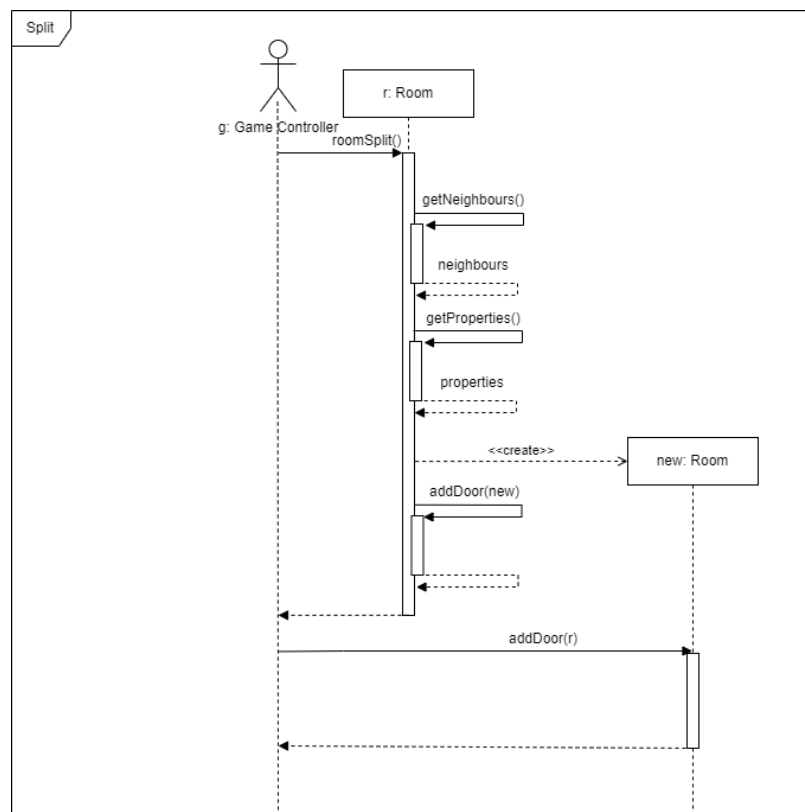
### 3.4.26 Move Teacher



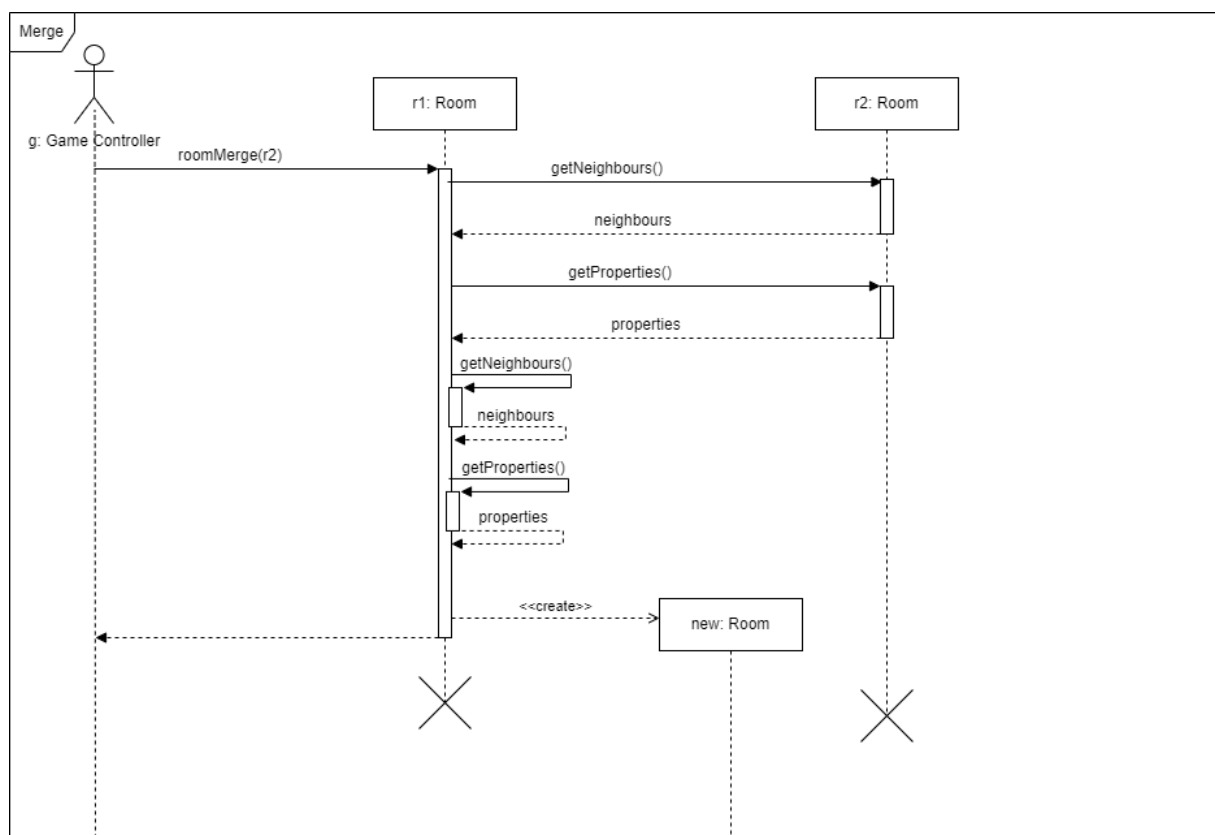
**3.4.27 Knock**

**3.4.28 Initialize**

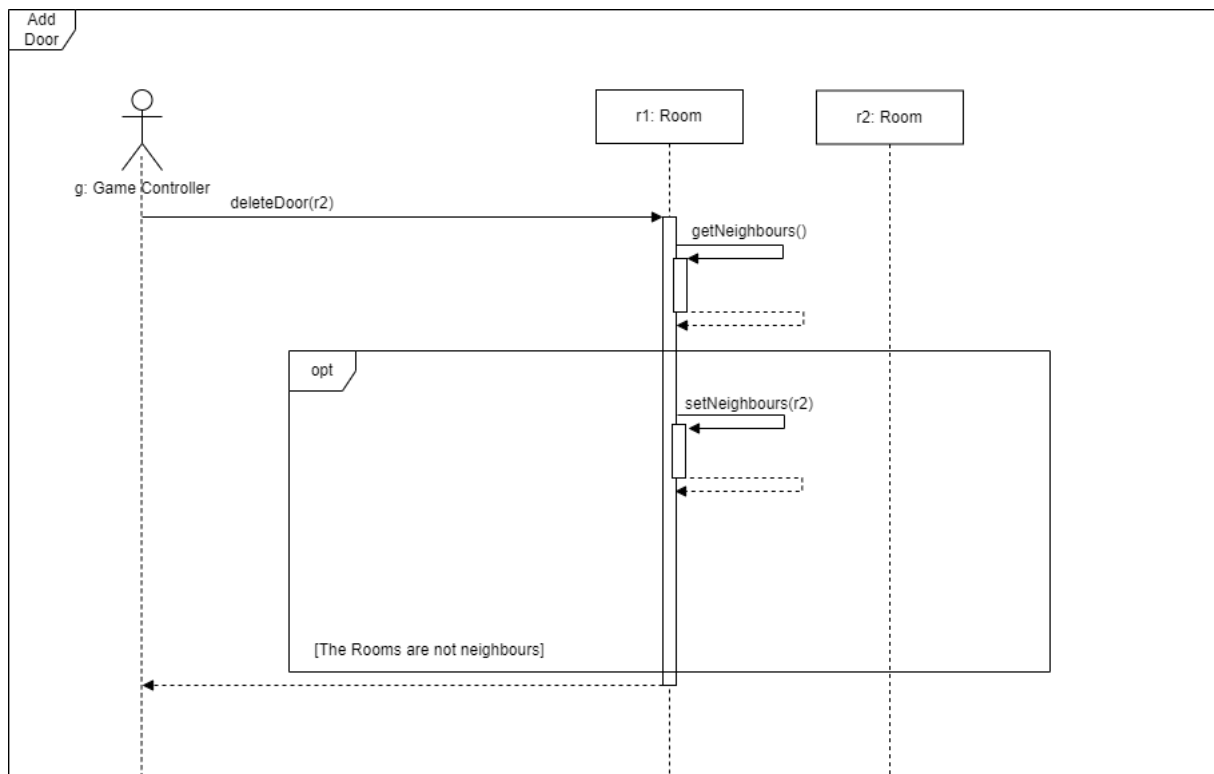
### 3.4.29 Split Rooms



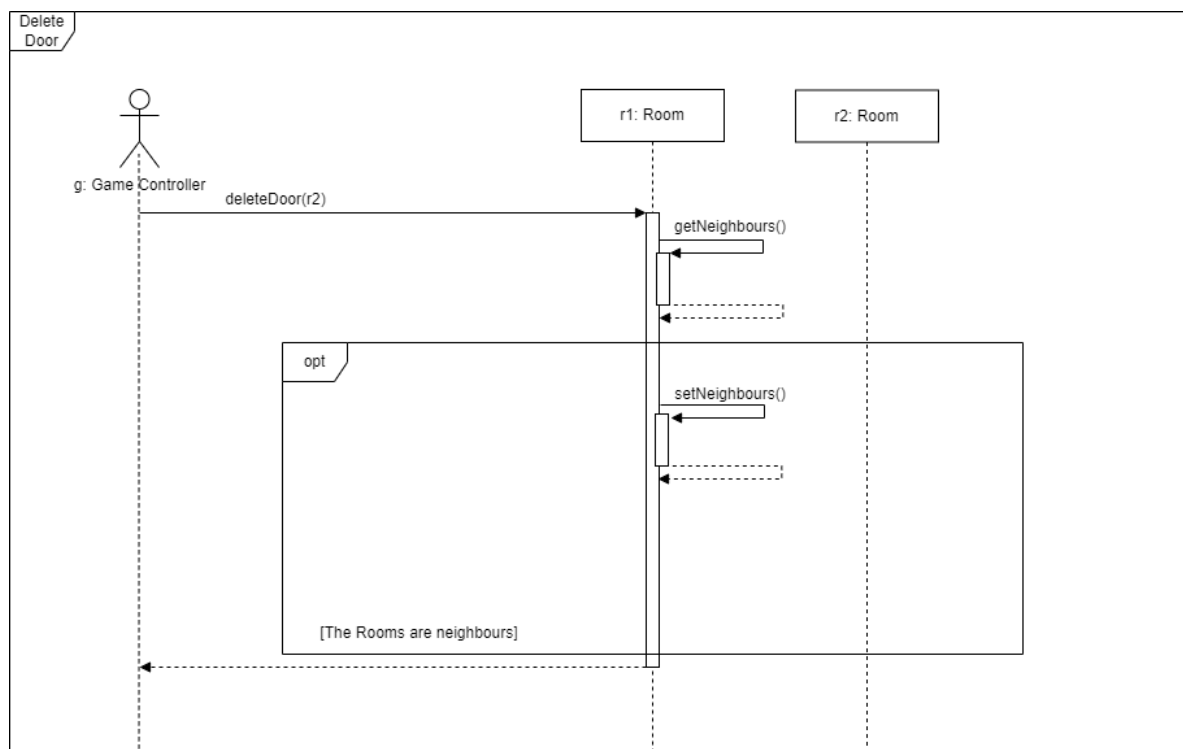
### 3.4.30 Merge Rooms



### 3.4.31 Add Door

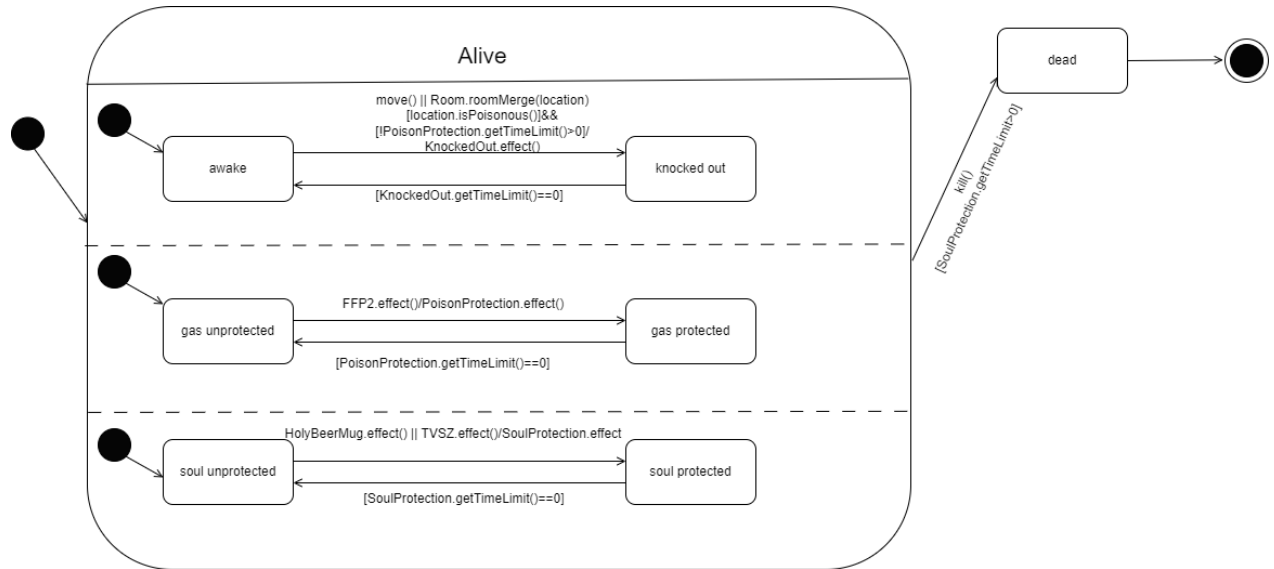


### 3.4.32 Delete Door

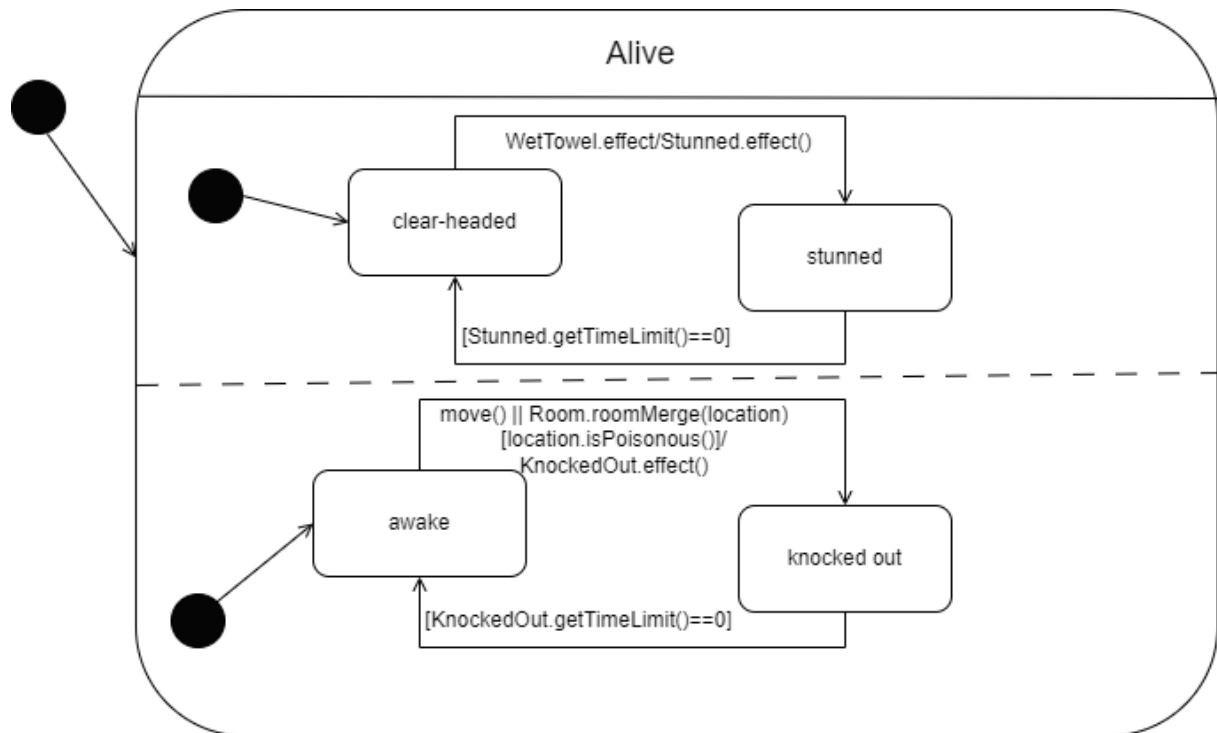


## 3.5 State-chartok

### 3.5.1 Student States

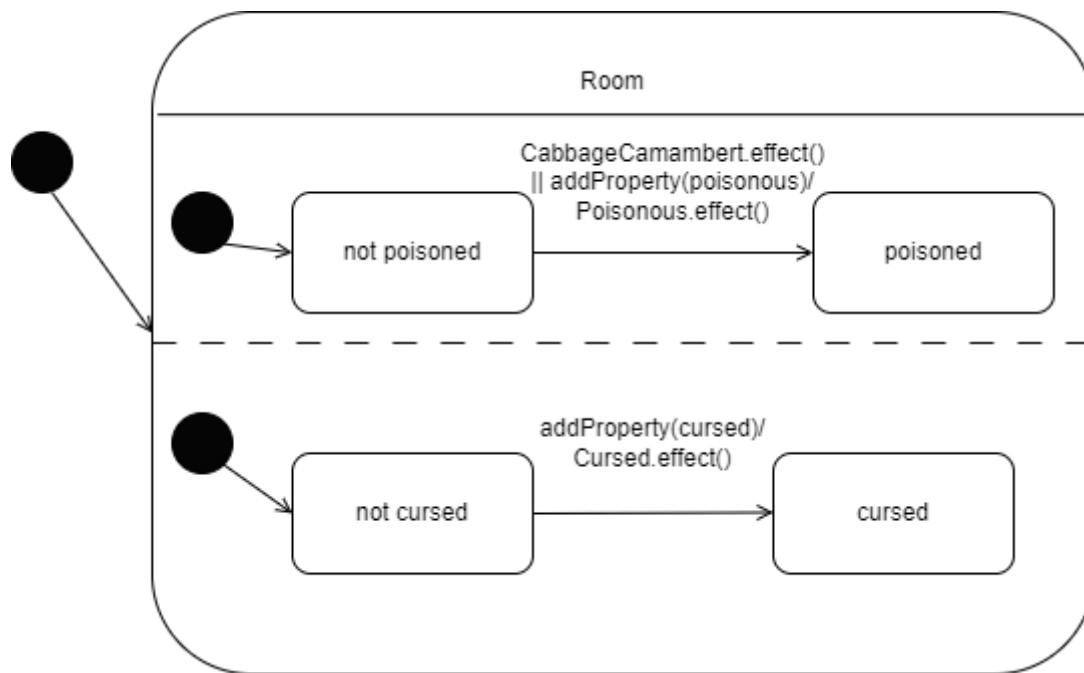


### 3.5.2 Teacher States



### 3.5.3 Room States





### 3.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.03.06 19:00	2 óra	Fazekas Molnár Simon Rimmel Juhász	Értekezlet: A konzultáción megjelent problémák átbeszélése, lehetséges javítások felvetése.
2024.03.08. 12:00	2 óra	Fazekas	Az értekezleten megbeszélte hibák javítása az osztálydiagrammon
2024.03.08 13:30	2,5 óra	Molnár	Az osztálydiagramnak megfelelően az osztályleírások javítása
2024.03.08 16.00	3 óra	Juhász	Szekvencia diagramok javítása
2024.03.08 20:30	3 óra	Rimmel	Szekvencia diagramok javítása

2024.03.09 13:00	6 óra	Juhász	Szekvenciadiagramok átdolgozása, osztálydiagrambeli esetleges módosítások feljegyzése
2024.03.09 13:00	6 óra	Simon	State Chartok átdolgozása, majd szekvenciadiagrammok javítása
2024.03.09 14:00	6 óra	Rimmel	Szekvenciadiagrammok készítése, osztálydiagrambeli esetleges módosítások feljegyzése
2024.03.09 15:00	5 óra	Fazekas	A szekvenciadiagrammok készítése során felmerült osztálydiagrambeli változtatások végrehajtása, osztályleírások frissítése, ellenőrzése
2024.03.09 20:00	5,5 óra	Molnár	A nap folyamán a csapattársak által készített diagramok, leírások, változtatások ellenőrzése.
2024.03.10 11:00	3 óra	Juhász	További szekvenciadiagramok készítése
2024.03.10 12:00	4 óra	Fazekas	Osztálydiagram, osztálykatalógus véglegesítése, dokumentumszerkesztés
2024.03.10 15:00	4 óra	Molnár	Szekvenciadiagramok készítése osztályleírások javítása.
2024.03.10 20:00	1 óra	Rimmel	Szekvenciadiagramok javítása
2024.03.10 21:00	4 óra	Simon	Szekvenciadiagramok ellenőrzése, dokumentum átnézése, fedőlap elkészítése

2024.03.11 10:00	1 óra	Juhász	Utólagos hibakeresés, megmaradt hibák javítása
------------------	-------	--------	--