



# Szoftver projekt laboratórium

## ANALÍZIS MODEL I.

Csapat

**61 - Zeusz**

Konzulens

Dr. Simon Balázs

### Csapattagok

---

Balla Gergely	[NEPTUN]	[email]
Joób Zalán Miklós	[NEPTUN]	[email]
Nagy Alexandra	[NEPTUN]	[email]
Tóth Boldizsár	[NEPTUN]	[email]
Zelei Máttyás	[NEPTUN]	[email]

2024. március 4.



## 3. fejezet

# Analízis modell kidolgozása 1.

### 3.1. Objektum katalógus

#### 3.1.1. Hallgató

A játékos irányítja. Szobáról szobára haladnak ajtókon keresztül. Tudják, melyik szobában vannak, milyen tárgyak vannak náluk, és tudnak velük interaktálni.

#### 3.1.2. Oktató

Szobáról szobára haladnak. Ha egy szobába kerülnek egy hallgatóval, kiejtik a játékból.

#### 3.1.3. Logarléc

A játék célja ezt megtalálni. Amint egy hallgató felveszi, a játék véget ér.

#### 3.1.4. Tranzisztor

Teleportálásra lehet használni. Össze lehet kapcsolni egy másik tranzisztorral, ami letéves után tudja, melyik szobában van. A kézben lévő tranzisztor a használat után deaktiválódik. Tudja, hogy mely szobákat kapcsolja össze.

#### 3.1.5. TVSZ denevérbőrön

Védelmet nyújt az oktatókkal szemben, oktatónként használódik. Tudja, hogy hány használat van hátra.

#### 3.1.6. Szent söröspohár

Aktiválás után 5 másodpercig nyújt védelmet az oktatókkal szemben, aztán elveszti a képességét. Tudja, hogy még mennyi idő van hátra a hatásából.

#### 3.1.7. Nedves táblatörlő rongy

A vele egy szobában lévő oktatókat megbénítja 15 másodpercre. Tudja, hogy mennyi idő van hátra a hatásának időtartamából.

### 3.1.8. FFP2-es maszk

A mérgesgáz ellen véd. Tudja, hogy hány használata van hátra, és hogy az adott használatból mennyi van hátra.

### 3.1.9. Dobozolt káposztás Camambert

Felbontáskor mézgező gázt bocsát ki. Használatkor a birtokló játékos elmondja a szobájának, hogy legyen adott ideig mérgező.

### 3.1.10. Szoba

Egy szoba a játékban. Ő birtokolja a tárgyakat, oktatókat, hallgatókat, amik megtalálhatók benne, illetve az ajtókat, amik belőle nyílnak. Tudja, hogy milyen különleges képessége van. Ismeri az ajtóit, hogy hány ember van benne

### 3.1.11. Ajtó

Tudja melyik két szoba között megy, azt, hogy egy- vagy kétirányú. Ha egyirányú, az irányt.

## 3.2. Osztályok leírása

### 3.2.1. Beer

#### ■ Felelősség

A játékbeli tárgy reprezentálása a modellben.

#### ■ Interfészek

Item

### 3.2.2. CodeOfStudies

#### ■ Felelősség

A játékbeli tárgy reprezentálása a modellben.

#### ■ Interfészek

Item

#### ■ Attribútumok

◇ - *uses : int* - Megadja, hogy hányszor használható még a tárgy.

### 3.2.3. Camambert

#### ■ Felelősség

A játékbeli tárgy reprezentálása a modellben.

#### ■ Interfészek

Item

### 3.2.4. Door

#### ■ Felelősség

Tudja melyik két szobát köti össze.

#### ■ Metódusok

◇ *use(Person person, Room from) : void* - A személy át kíván lépni a másik szobába. Ha abban van kapacitás a szobák értesülnek az átlépésről. Különben nem történik semmi.

### 3.2.5. Item

#### ■ Felelősség

Interfész biztosítása a különböző tárgyat reprezentáló osztályok számára.

#### ■ Metódusok

◇ *accept(visitor : ItemVisitor) : void* - Visitor befogadása.

◇ *use(target : GameObject) : void* - Felhasználási kérelem befogadása.

### 3.2.6. Mask

#### ■ Felelősség

A játékbeli tárgy reprezentálása a modellben.

#### ■ Interfészek

Item

#### ■ Attribútumok

◇ *effect : RagEffect* - A maszkhoz tartozó effekt.

◇ *uses : int* - A maszk felhasználásainak száma

### 3.2.7. Person

#### ■ Felelősség

Alaposztály szolgáltatása a mozgó játék beli entitások számára.

#### ■ Attribútumok

- ◇ - *name* : *String* - Az ember neve.
- ◇ - *numItems* : *int* - Az embernél lévő tárgyak száma.

#### ■ Metódusok

- ◇ *+dropItem(Item item)* : *void* - Az ember eldobja/elejti a tárgyat.
- ◇ *+knockOut()* : *void* - Az ember adott időre elájul.
- ◇ *+enterRoom(room : Room)* *void* - A szoba értesítése arról, hogy a játékos belépett.

### 3.2.8. Room

#### ■ Felelősség

Ő birtokolja a benne lévő Door, Person és (nem Personnál lévő) Item objektumokat. Ő engedélyezi vagy tagadja meg a Personok belépését. Ő értesíti a benne lévő Personokat új Person belépéséről, mérges gázzal, táblatörlő rongy használatáról.

#### ■ Attribútumok

- ◇ - *capacity* : *int* - Megadja maximum hány ember fér el a szobában.
- ◇ - *numOfPeople* : *int* - A szobába jelenleg tartózkodó emberek száma.

#### ■ Metódusok

- ◇ *visitEffects(visitor: EffectVisitor)* : *void* - Meghívja az összes szobára hatást gyakorló visitort.
- ◇ *visitPeople(visitor: PeopleVisitor)* : *void* - Meghívja az összes szobában lévő emberre a visitort.
- ◇ *enter(Person person)* : *void* - A személy belép a szobába és a többi személy értesül róla, és reagál, ha kell
- ◇ *leave(Person person)* : *void* - A szobát elhagyna egy személy, a szobában lévő emberek száma csökken.
- ◇ *merge(Room room)* : *void* - Két szoba összeolvad.
- ◇ *split()* : *void* - A szoba a szabályoknak megfelelően osztódik.

### 3.2.9. Student

#### ■ Felelősség

A játékos reprezentálása a modellben.

#### ■ Össosztály

Person

### ■ Metódusok

- ◇ *+beingAttacked()* : *void* - Megtámadja egy oktató. Ha van védőeszköze, azt használja. Ha nem, eltávolítja magát a szobából.
- ◇ *+ignoreTeacher(Teacher teacher)* : *void* - Atott tanárral szemben védi a hallgatót, amíg ugyanabban a szobában vannak.
- ◇ *+eliminate()* : *void* - A hallgatót értesíti arról, hogy meghalt.

## 3.2.10. Teacher

### ■ Felelősség

Az ellenségek reprezentálása a modellben.

### ■ Ősosztály

Person

## 3.2.11. Transistor

### ■ Felelősség

A játékbeli tárgy reprezentálása a modellben.

### ■ Interfészek

Item

### ■ Metódusok

- ◇ *use(person : Person)* : *void* -
- ◇ *use(item : Item)* : *void* -
- ◇ *use(transistor : Transistor)* : *void* -

A három use függvény a Use Transistors szekvencia-diagramban van magyarázva.

- ◇ *link(other: Transistor)* - A tranzisztor összekapcsolása a párjával.
- ◇ *setTarget(target : Room)* : *void* - A tranzisztor-teleportáció célszobájának beállítása

## 3.2.12. WetRag

### ■ Felelősség

A játékbeli tárgy reprezentálása a modellben.

### ■ Interfészek

Item

### ■ Attribútumok

- ◇ - *effect : RagEffect* - A tárgyhoz hozzárendelt effekt.

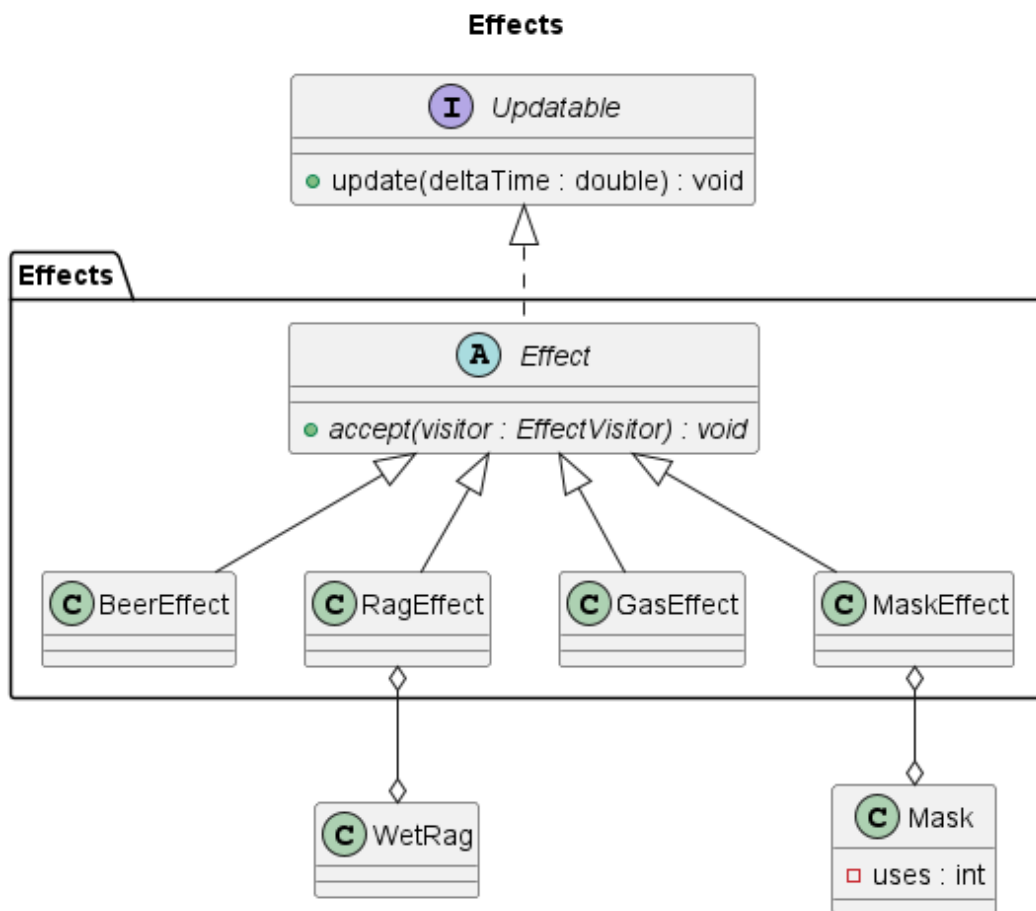
### 3.3. Statikus struktúra diagramok

Az osztálydiagram létrehozásakor egy szoba alapú architektúrát modelleztünk. Architektúránkban a szobák kezelik a területükön végbemenő folyamatokat.

A szobába való belépés egy ajtón keresztül történik. Egy ajtót a Door osztály reprezentál. Egy játékos(Student) az ajtó(Door) objektumhoz fordulva léphet be egy szobába. A belépési kérelmet az ajtó összeveti a saját irányával, majd ha a kérés kompatibilis az átengedési követelménnyel, akkor a kérést továbbítja a hozzá rendelt Room objektumnak, ami a kérésről a végső döntést meghozza.

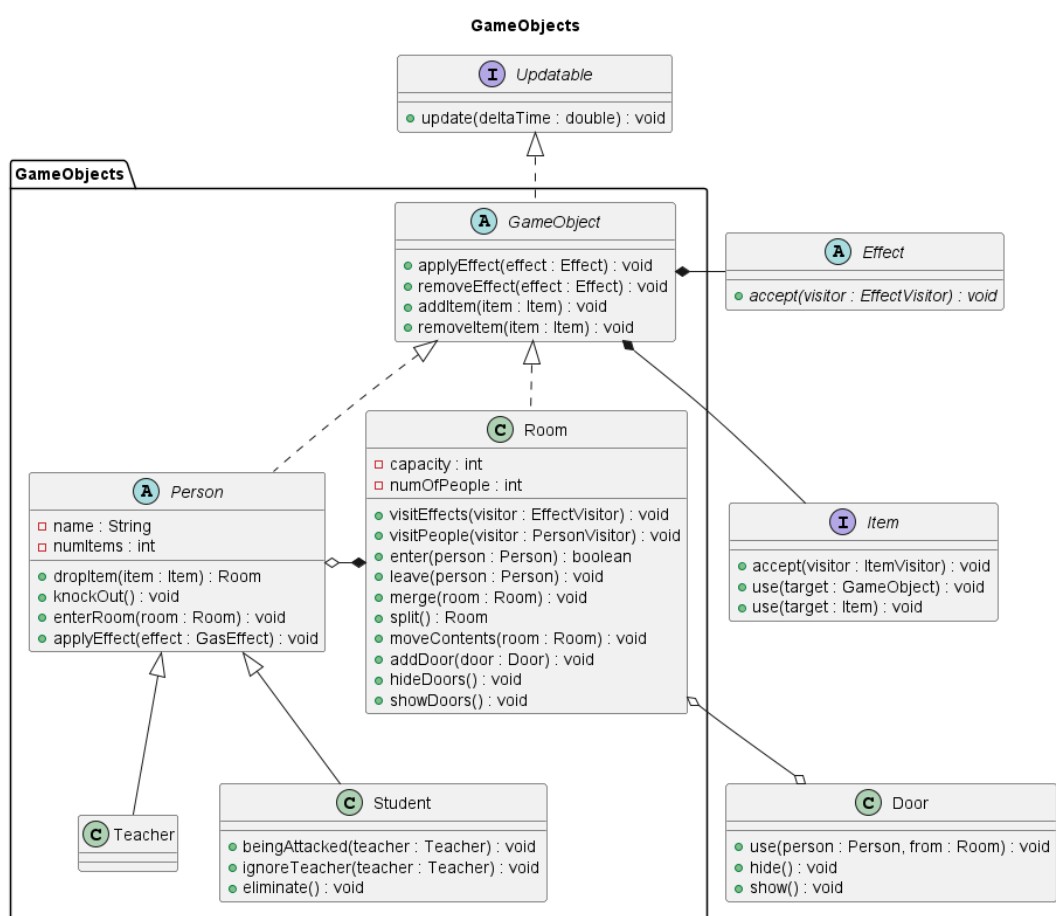
A játékban két mozgásra képes entitás létezik: a tanárok(Teacher) és a játékosok(Student). Ezeket egy ősosztályból származtatjuk(Person).

A tárgyak(Item) kezelését Visitor tervezési minta használatával oldottuk meg, mivel így a kezelési algoritmusok teljesen elválaszthatóak az objektumoktól. Így későbbiekben a kezelés viselkedése megváltoztatható az adott osztály belső működésének megváltoztatása nélkül. A különböző tárgyak egy Item interfész megvalósításai, így a tárgyak körében biztosított az encapsulation elve. Ezeket az Item-eket Különböző visitor-ok kezelik, melyek az ItemVisitor, EffectVisitor és PersonVisitor interfészeket valósítják meg.

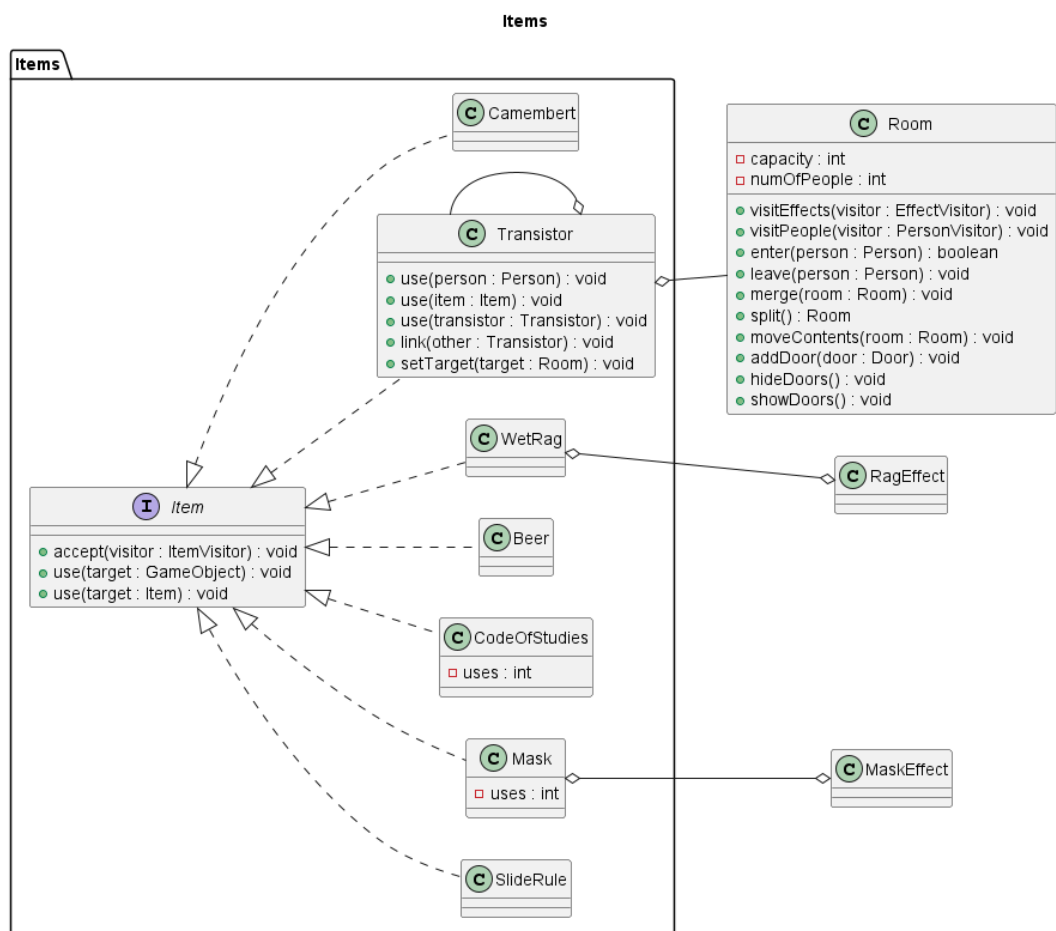


3.1. ábra. Hatások

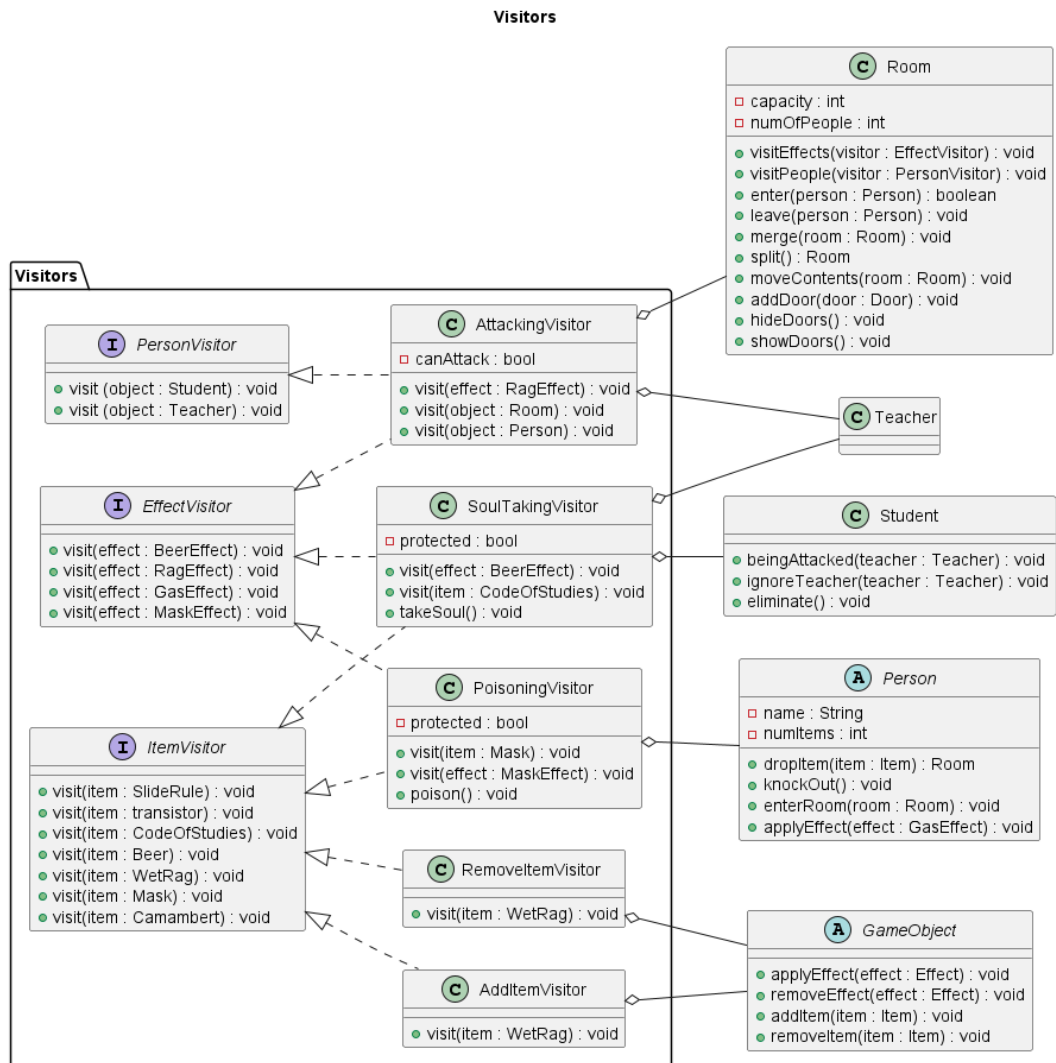




3.2. ábra. Játék objektumok

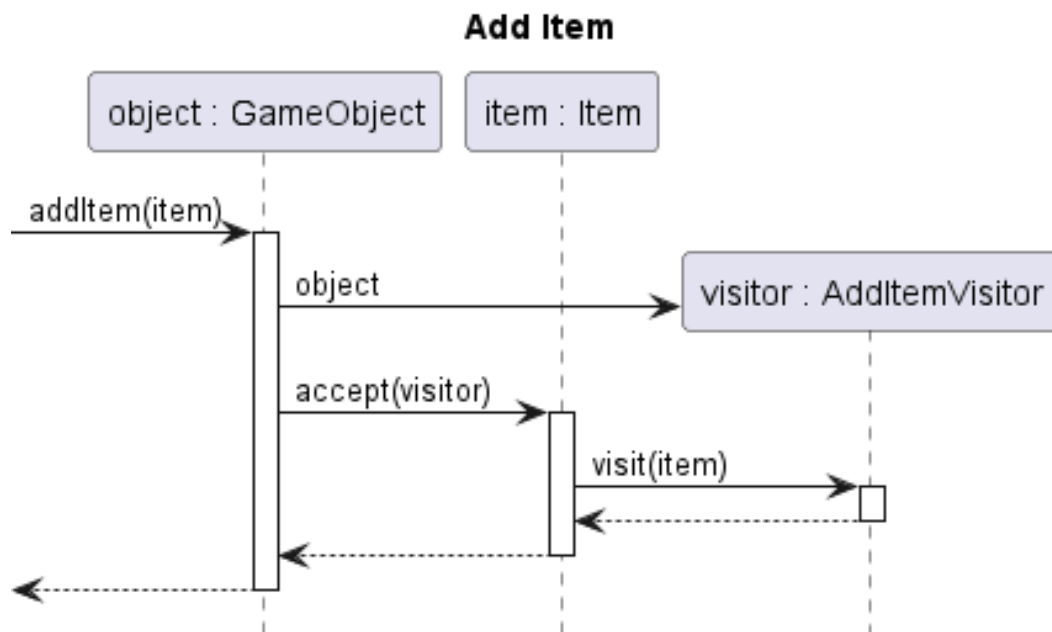


3.3. ábra. Tárgyak

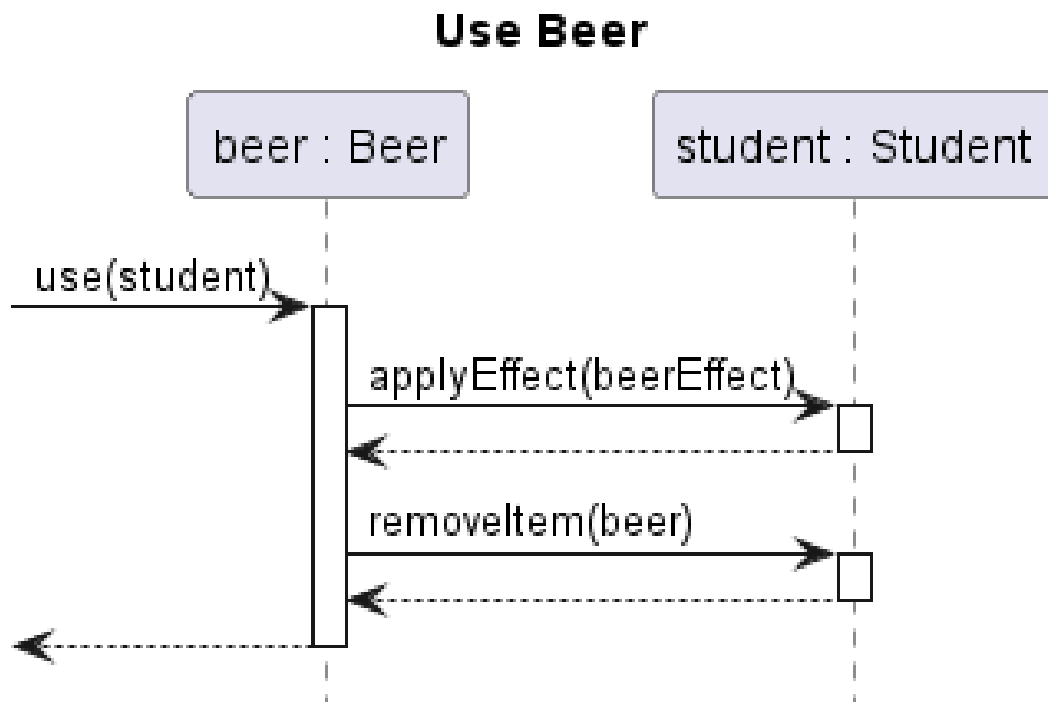


3.4. ábra. Látogató minta megvalósítása

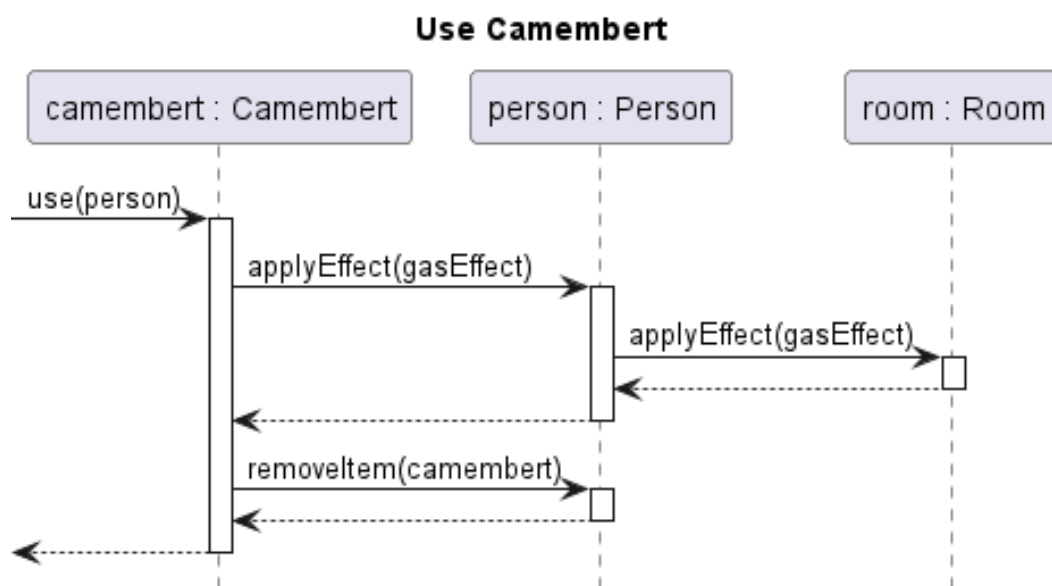
### 3.4. Szekvencia diagramok



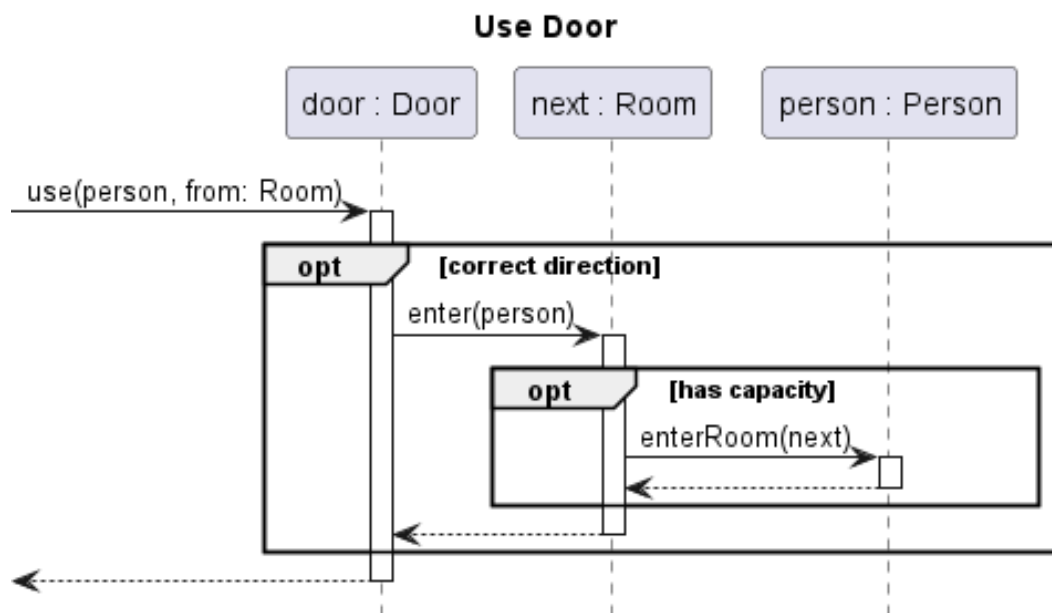
3.5. ábra. Tárgyfelvétel diagram



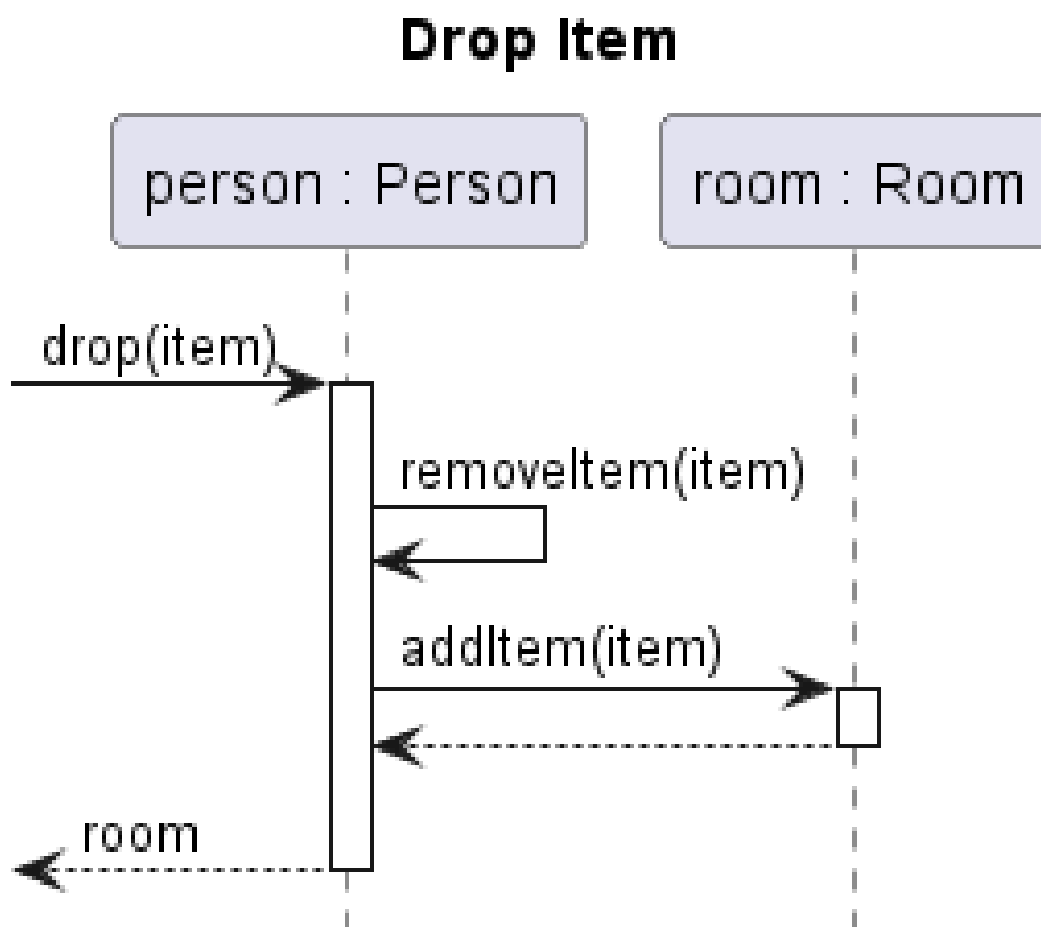
3.6. ábra. Söröspohár használata diagram



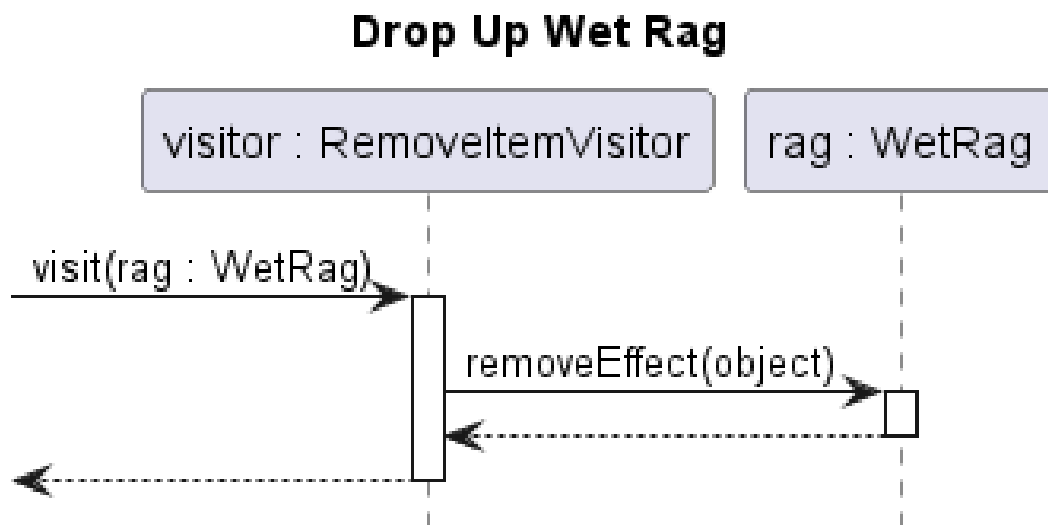
3.7. ábra. Camembert diagram



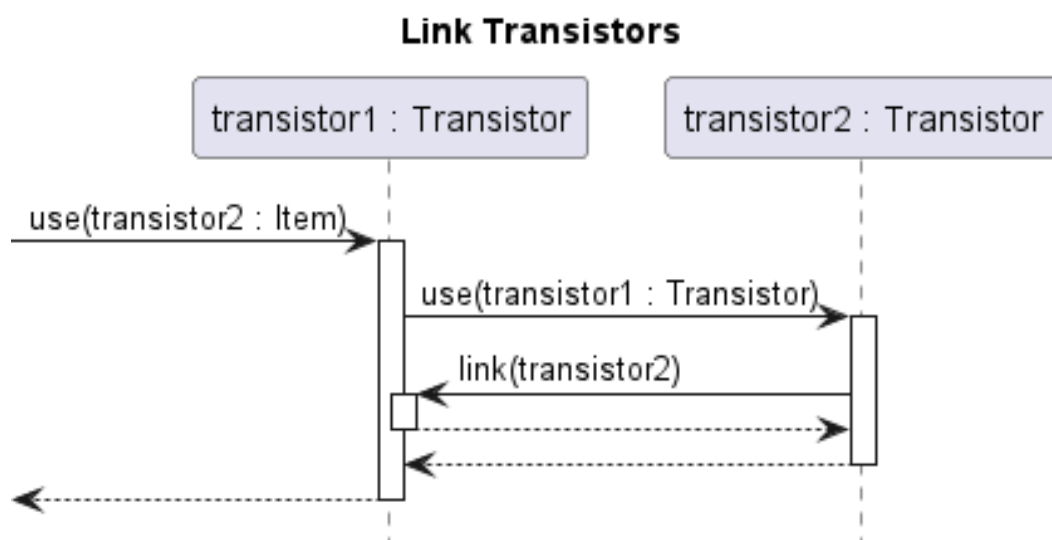
3.8. ábra. Ajtó használat diagram



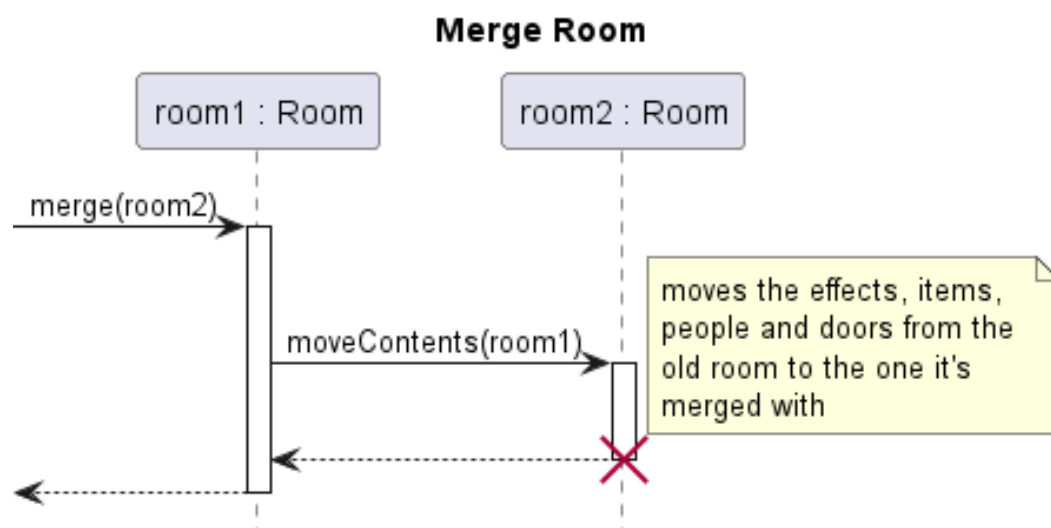
3.9. ábra. Tárgyelejtés diagram



3.10. ábra. Nedves táblatörő rongy elejtése diagram

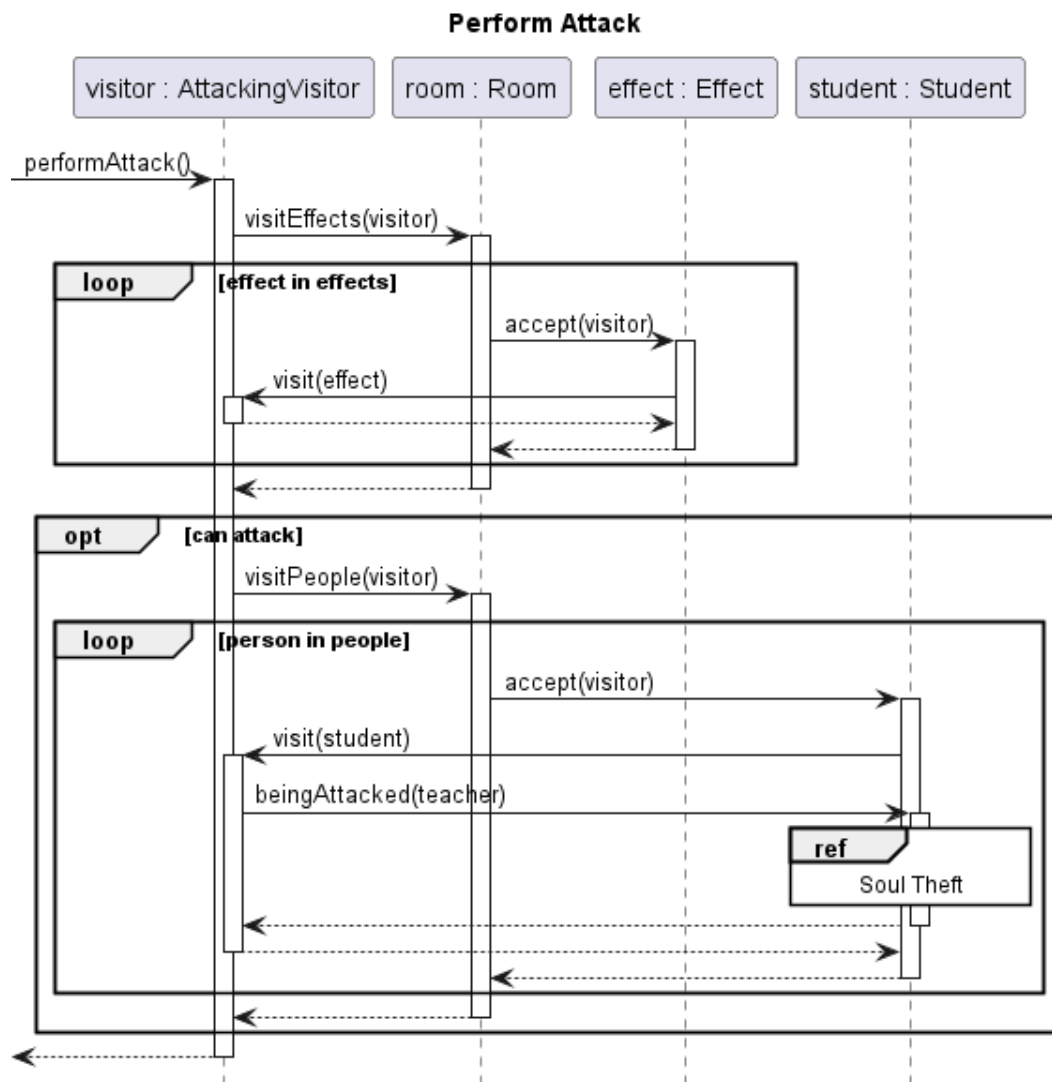


3.11. ábra. Tranzisztor összekapcsolása diagram

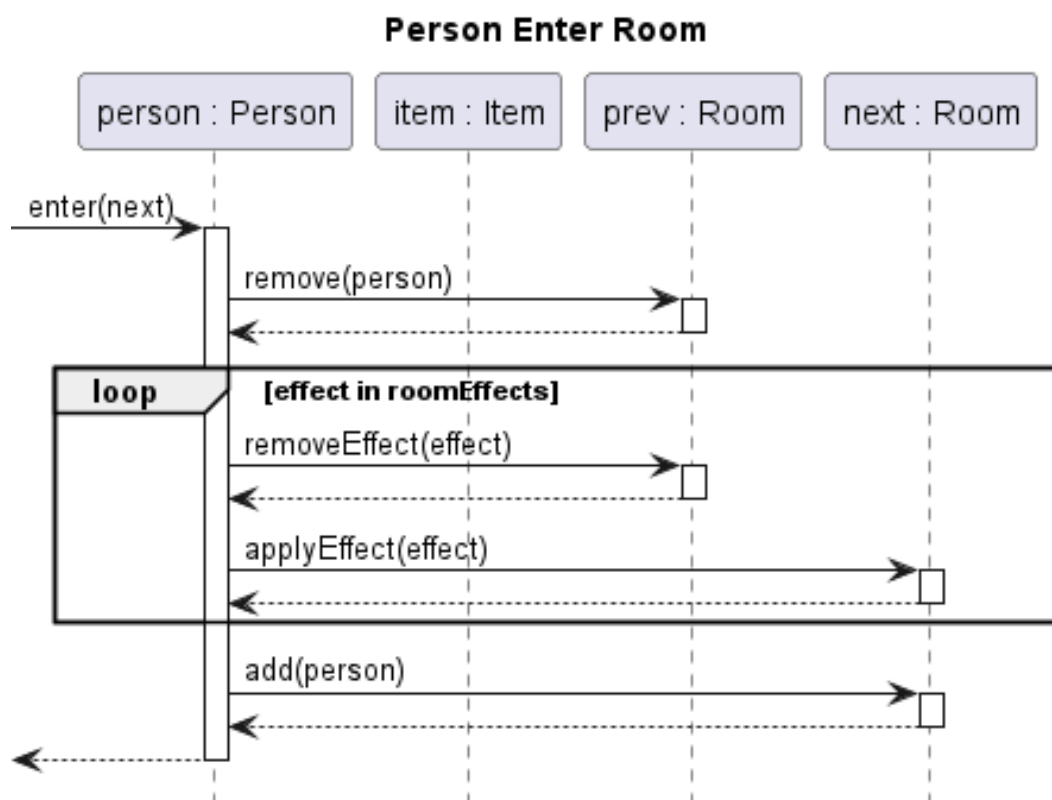


3.12. ábra. Szobák egyesülése diagram

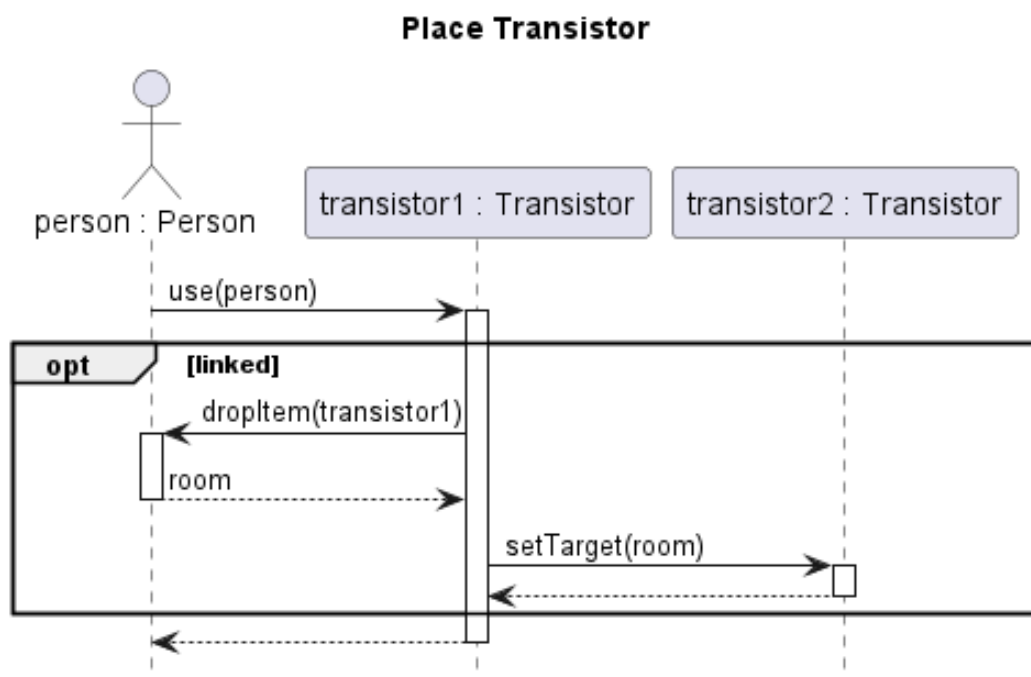




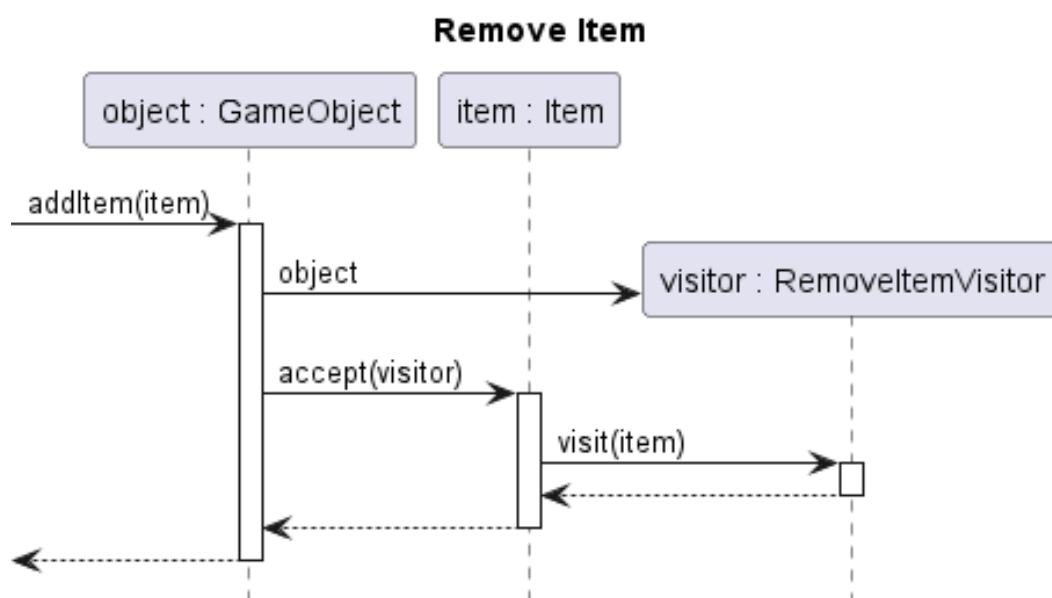
3.13. ábra. Támadás diagram



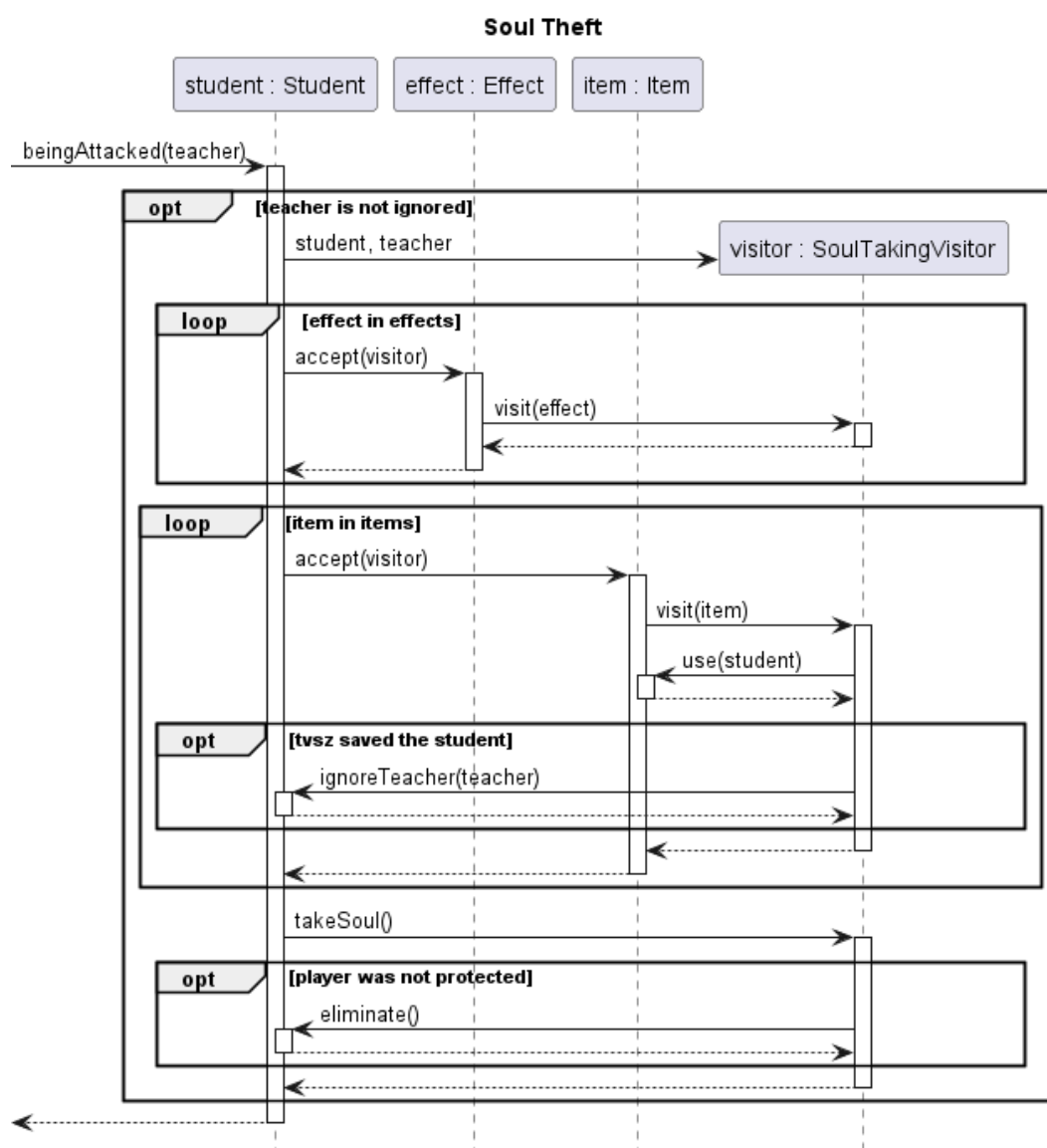
3.14. ábra. Személy belépése egy szobába diagram



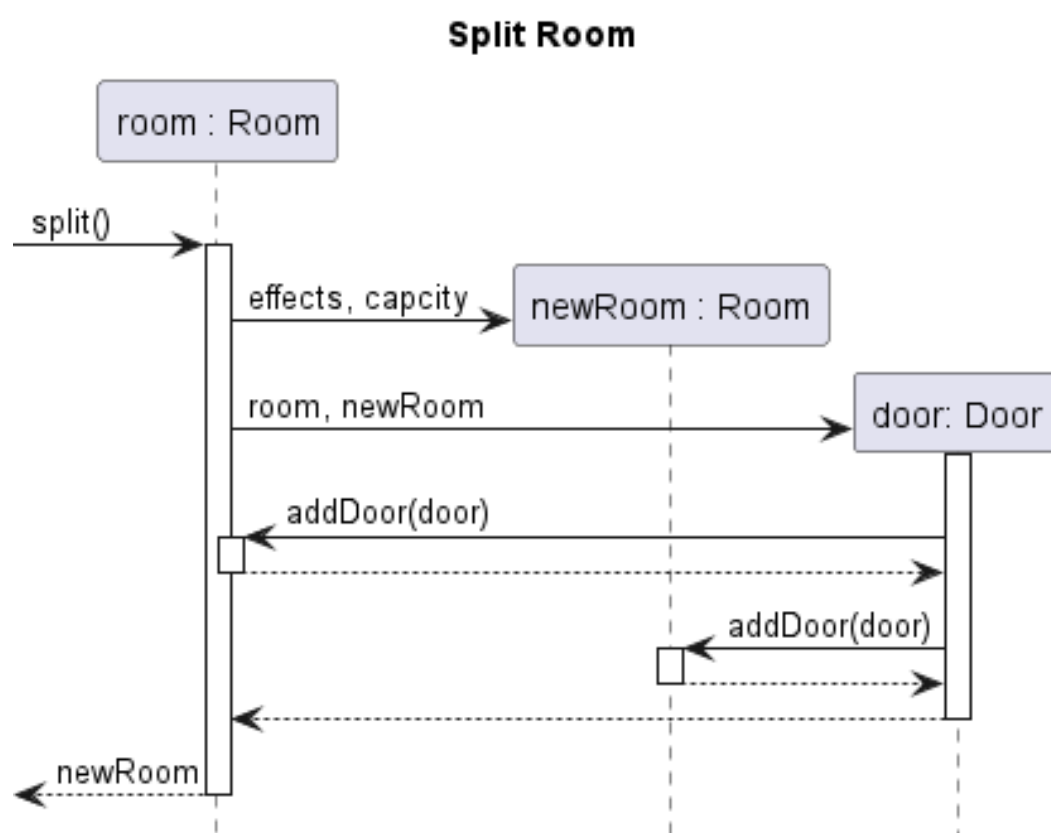
3.15. ábra. Tranzisztor lehelyezése diagram



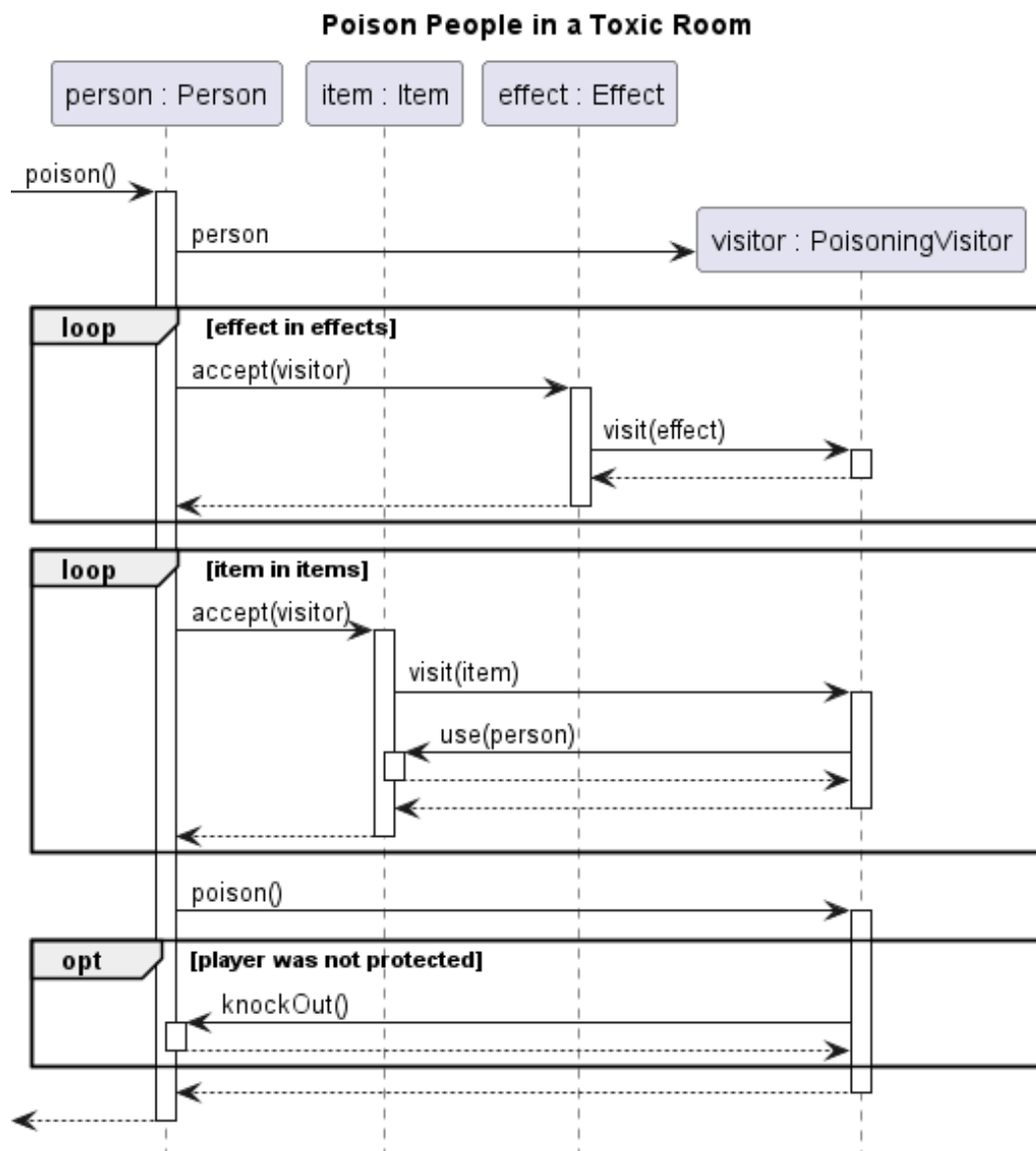
3.16. ábra. Tárgy eltávolítása diagram



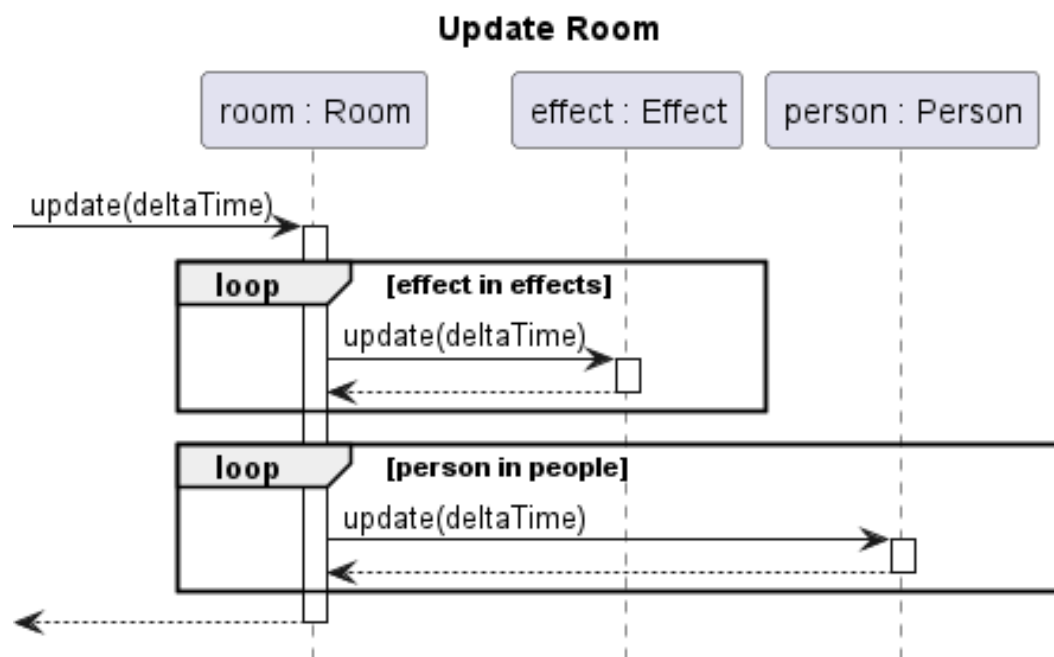
3.17. ábra. Lélek elvétel diagram



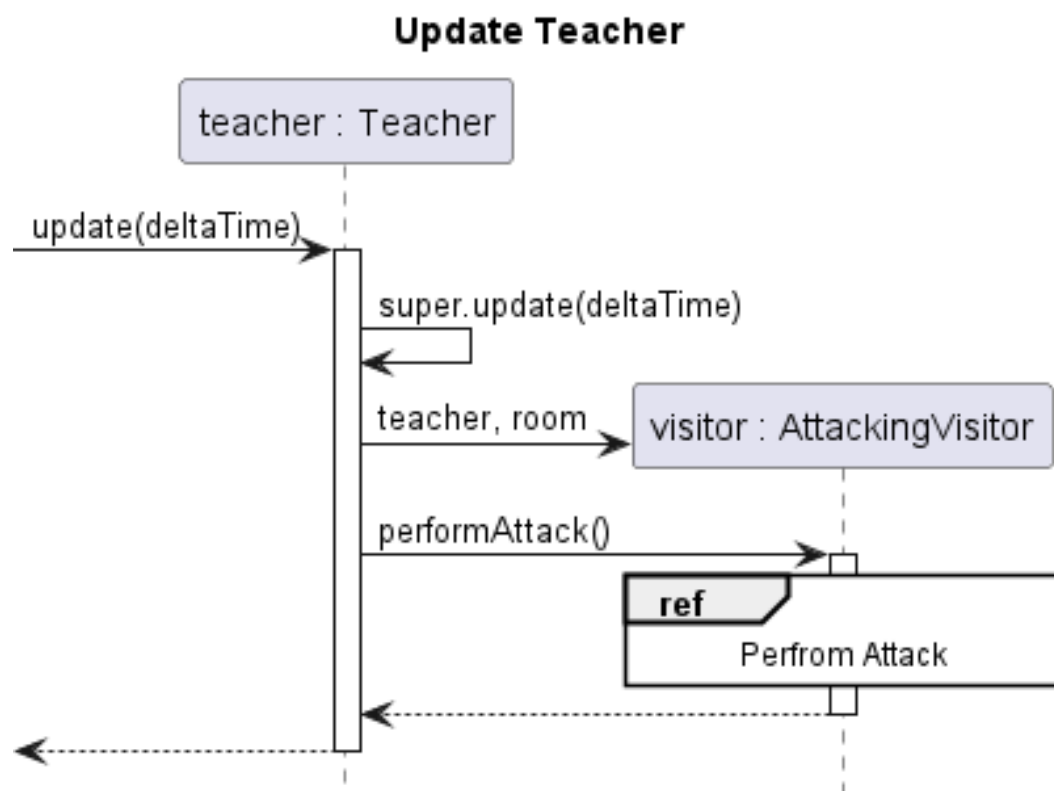
3.18. ábra. Szoba osztódás diagram



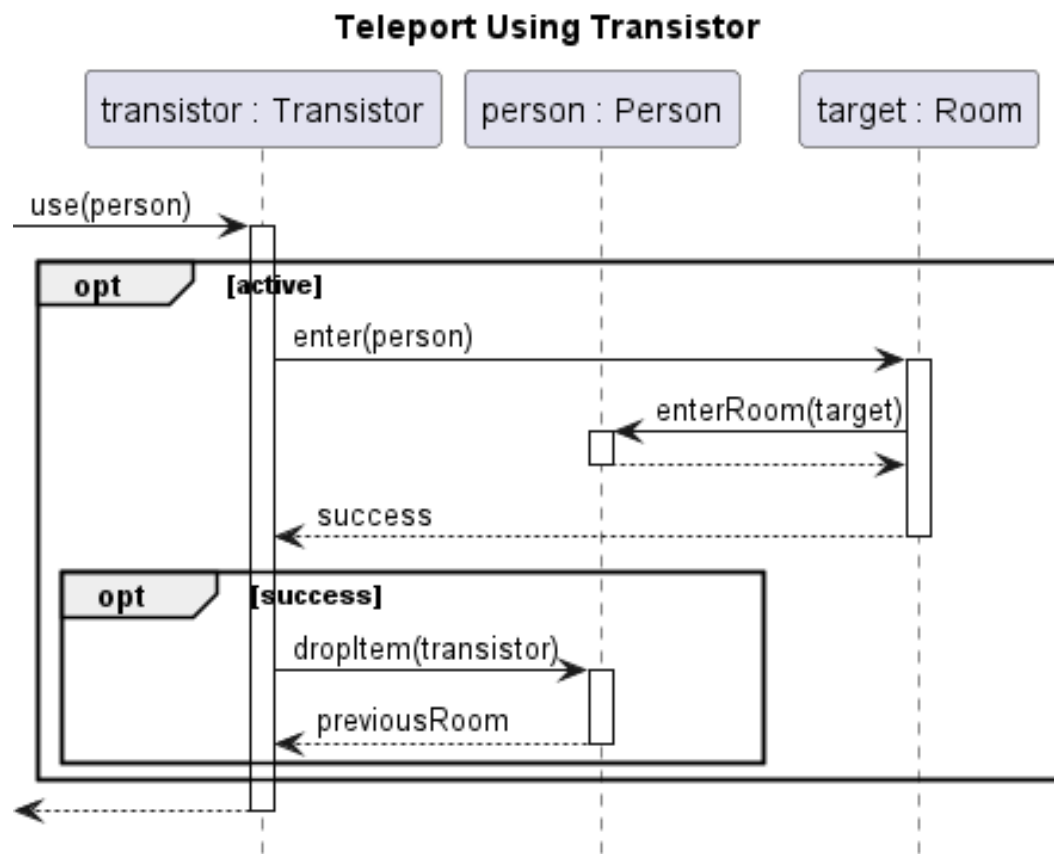
3.19. ábra. Mérgező szoba diagram



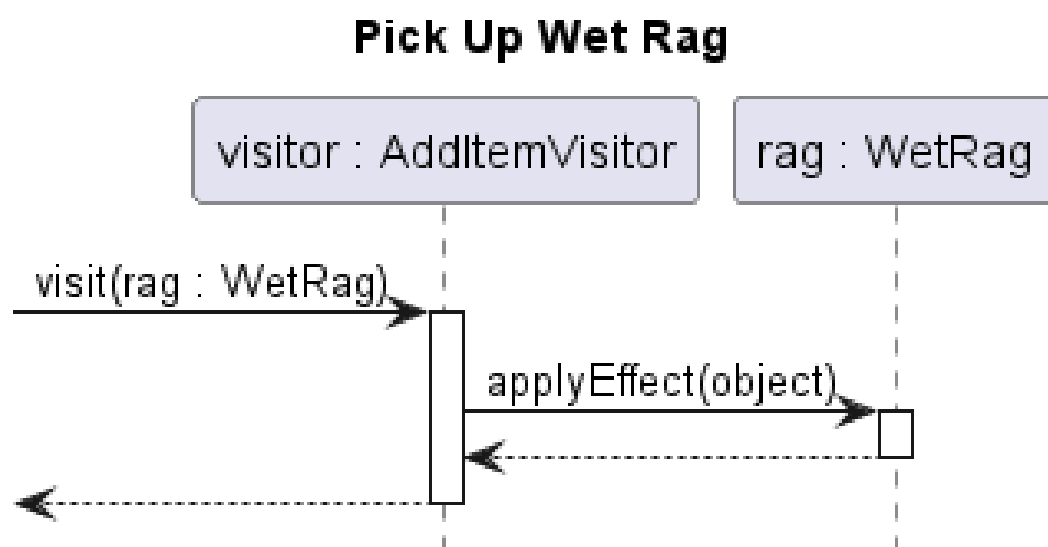
3.20. ábra. Szoba frissítése diagram



3.21. ábra. Oktató frissítése diagram



3.22. ábra. Tranzisztor használat diagram



3.23. ábra. Nedves táblatörő rongy felvétele diagram



### 3.5. Napló

Kezdet	Időtartam	Résztevők	Leírás
feb. 28. 10:15	1,5 óra	Balla Joób Nagy Tóth Zelei	Labor
feb. 28. 11:45	4 óra	Balla Joób Nagy Tóth Zelei	Értekezlet, a feladatok közös megoldása
feb. 29. 12:00	2 óra	Tóth	Modell megvalósítási lehetőségeinek felderítése
márc. 1. 12:00	2 óra	Nagy	Osztályok leírásának készítése
márc. 1. 10:00	2 óra	Joób	Megoldás keresés a modellben felmerülő problémákra
márc. 1. 13:00	1 óra	Tóth	Modell megvalósítási lehetőségeinek felderítése
márc. 1. 21:00	2 óra	Tóth	Item és ItemVisitor diagrammok elkészítése
márc. 2. 15:00	1 óra	Zelei	Szekvenciadiagramok készítése
márc. 2. 20:00	6 óra	Tóth	Tárgyak és hatásaik kezelésének modellezése
márc. 3. 18:00	0,5 óra	Nagy	Szekvencia és osztálydiagram konzisztencia ellenőrzése
márc. 3. 19:00	1 óra	Balla	Szekvencia diagramok készítése, dokumentáció javítása
márc. 3. 19:00	8 óra	Tóth	Szekvencia diagrammok elkészítése, konzisztencia javítása
márc. 3. 20:00	1 óra	Joób	Szekvencia diagrammok készítése
márc. 4. 8:00	0,5 óra	Nagy	Szekvencia diagrammok javítása
márc. 4. 8:30	3 óra	Tóth	Diagrammok véglegesítése, hibák javítása
márc. 4. 9:00	2 óra	Joób	Dokumentáció készítése