

4. Analízis modell kidolgozása II.

18 – Mumbai IT Solution

Konzulens:
Goldschmidt Balázs

Csapattagok

Cardinael Jan Maerten	M1GVE2	cardinael.jan@gmail.com
Görömbey Lilla Zsófia	EBQG95	gorombeylilla@gmail.com
Riba Miklós Pál	J519A1	ribamiki@outlook.hu
Király Bálint	EQF1M0	kiraly.balint@edu.bme.hu
Szakov Máté Antal	DKQQXE	szakos.mate@gmail.com

2024.03.11.

4. Analízis modell kidolgozása

4.1 Objektum katalógus

4.1.1 Beer

Felvehető tárgy, ha a játékos birtokában van és aktiválja, megmenti az oktatóktól adott ideig.

4.1.2 Camembert

Felvehető tárgy, ha a játékos birtokában van és aktiválja, a szoba amiben van, elgázosodik.

4.1.3 Cloth

Felvehető tárgy, ha a játékos birtokában van és aktiválja, akkor a szobában lévő oktatókat megbénítja, majd kiszárad.

4.1.4 Door

A hallgatóknak és oktatóknak biztosít átjárást a szobák között. Vannak 1 illetve 2 irányú ajtók, azaz bizonyos ajtók csak egy irányban használhatók.

4.1.5 Mask

Ha egy hallgató elgázosított szobába lép és van nála FFP2-es maszk, akkor az adott ideig megvédi a gázmérgezéstől. Nem szükséges aktiválni, mivel passzív hatása van: minden körben amikor a hallgató elgázosított szobában tartózkodik védelmet biztosít.

4.1.6 Room

A szobákban vannak hallgatók, oktatók és tárgyak. Minden szobának van befogadóképessége. Ennél több személy a szobában nem tartózkodhat. Ezen kívül a szobáknak több fajtája is ismert. Vannak szobák, amikben mérgező gáz van. Az ide belépő hallgatók és oktatók egy rövid időre eszméletüket veszítik (2 körig nem csinálhatnak semmit) és a náluk lévő tárgyakat elejtik. 2 kör letelte után a hallgató lehetőséget kap a szoba elhagyására, ha ott marad újra eszméletét veszti. Vannak olyan elátkozott szobák, amiknek az ajtajai időnként (körönként véletlenszerűen) eltűnnek, majd később újra előtűnnek. A szobák egy korábbi (félresikerült) gráfelméleti tételbizonyítás eredményeként meghazudtolják a fizika törvényeit: képesek egyesülni és osztozni. Két szomszédos szoba egyesülésével létrejövő szoba a korábbi két szoba tulajdonságaival, szomszédjaival, tárgyaival és entitásaival rendelkezik, de a befogadóképessége a nagyobb szoba befogadóképességével lesz azonos. Két szoba csak akkor egyesülhet, ha a két szobában lévő entitások összege nem haladja meg a nagyobb szoba befogadóképességét. Az osztódó szoba két olyan szobára válik szét, amelyek egymás szomszédai lesznek, és megosztoznak a korábbi szoba tulajdonságain és szomszédain (a korábbi szomszédok, tárgyak és entitások vagy csak az egyik, vagy csak a másik "új" szobáé lesznek). Az osztódás után létrejövő két szoba között kétirányú ajtó jön létre. A két "új" szoba befogadóképessége az eredeti szobának a befogadóképességével fog megegyezni. Egy szobában bármennyi tárgy lehet.

4.1.7 SlideRule

Ebből a tárgyból csak egy darab van összesen, felvételével a hallgatók megnyerik a játékot. Az oktatók nem képesek megsemmisíteni.

4.1.8 Student

A játékos által mozgatott karakter. Felvehet tárgyakat és szobákból szobába léphet.

4.1.9 Teacher

Egy a program által irányított karakter, amelynek célja a hallgatók megakadályozása. Ha egy szobába kerül egy hallgatóval, megtámadja azt, ha a hallgató nem képes védekezni, akkor a hallgató számára vége a játéknak. Továbbá képes tárgyakat felvenni a játéktérről, ezzel is nehezítve a hallgatók dolgát.

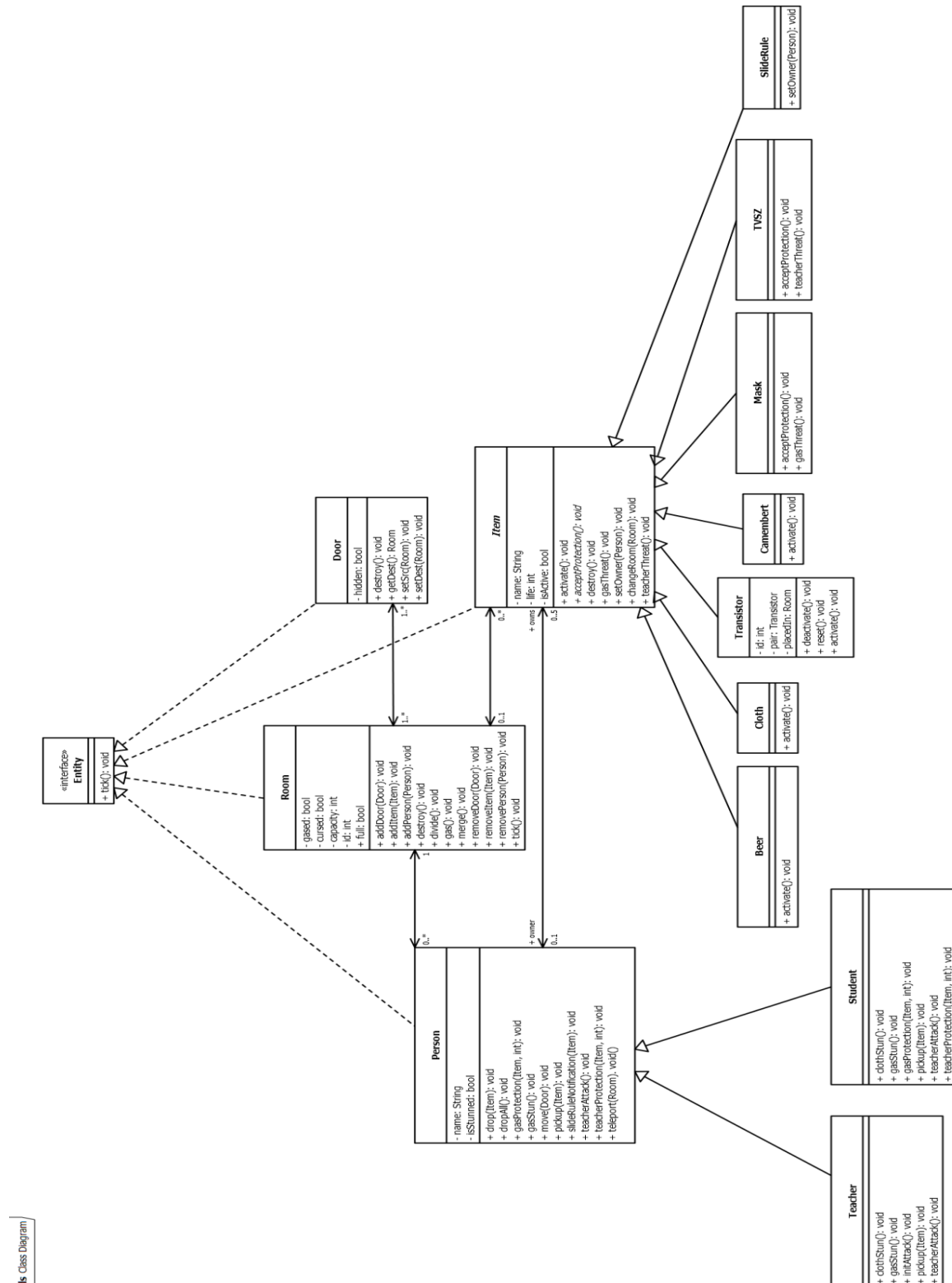
4.1.10 Transistor

A hallgatónál lévő tranzisztorokat páronként össze lehet kapcsolni. A tranzisztorok az aktiválásuk, azaz bekapcsolásuk sorrendjében alkotnak párokat. Az egyik szobában be kell kapcsolni és letenni az egyik tranzisztort majd a pár másik tagját menet közben egy másik szobában le lehet tenni. Az így összekapcsolt tranzisztorok varázserővel bírnak: ha a hallgató a pár második tagját bekapcsolja és leteszi, akkor az elsőként lehelyezett tranzisztor szobájába kerül és mindkét tranzisztor kikapcsol. Ez innentől kezdve akárhányszor lejátszható, bármelyik irányban. A tranzisztorok korlátlan ideig használhatóak, azonban élettartamukat limitálja az a tény, hogy az oktatók is felvehetik a tárgyakat, azaz a tranzisztorokat is, ekkor a felvett tranzisztor a hallgatók nagy sajnálatára megsemmisül. A tranzisztorok megőrzik párjaikat mindaddig, amíg azt egy oktató el nem pusztítja. A pár megmaradt tagját egy hallgató újra beüzemelheti, ha felveszi és bekapcsolva elhelyezi egy új pár részeként.

4.1.11 TVSZ

Három alkalommal menti meg a hallgató életét az oktatóval szemben, utána eltűnik.

4.2 Statikus struktúra diagramok



4.3 Osztályok leírása

*Az őszosztály asszociációit és metódusait nem minden esetben tüntetjük fel a leszármazottaknál.

4.3.1 Beer

- **Felelősség**
Ha a hallgató aktiválja, akkor védelmet nyújt neki az oktatók ellen. Számontartja, hogy meddig van érvényben.
- **Őszosztályok**
Item
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok:**
- **Metódusok:**
activate(): Az active attribútumot igazra állítja, innentől minden tick() hatására csökkenti a life értékét.

4.3.2 Camembert

- **Felelősség**
Ha aktiválódik, akkor elgázosítja azt a szobát, amelyikben van. Jelzi a szobának, hogy el van gázosítva.
- **Őszosztályok:** Item
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok:** nincs
- **Metódusok**
activate(): Meghívja a szoba gas függvényét majd megsemmisül.

4.3.3 Cloth

- **Felelősség**
Megpróbál mindenkit megbénítani, aki a szobában tartózkodik, de csak az oktatókra lesz hatással. Használat után megsemmisül.
- **Őszosztályok:** Item
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok:** nincs
- **Metódusok**
activate(): a Cloth meghívja a szoba összes Personjének a clothStun() függvényét majd megsemmisül.

4.3.4 Door

- **Felelősség**
Átjárást biztosít szobák között a személyek számára, csak az egyik vagy mindkét irányban. Tudja hogy melyik szobák között van.
- **Össosztályok:** nincs
- **Interfészek:** Entity
- **Asszociációk:**
Room source: Szomszédos szoba, innen nyílik.
Room destination: Szomszédos szoba, ide vezet.
- **Attribútumok:**
bool hidden: Használható-e az adott ajtó. (eltűnt-e)
- **Metódusok:**
destroy(): Megsemmisíti az ajtót.

4.3.5 Entity

- **Felelősség**
Egy alap interfész a játék objektumainak számára.
- **Össosztályok:** nincs
- **Interfészek:** nincs
- **Asszociációk:**
- **Attribútumok:**
- **Metódusok:**
void tick(): Egy időegység leteltét jelzi az objektum számára.

4.3.6 Item

- **Felelősség**
A tárgyak absztrakt össosztálya. Nyilvántartja a tárgy nevét, élettartamát, aktiválja a tárgyat és eltárolja, hogy aktív-e. Elpusztítja magát, ha szükséges. Ha kérnek tőle protection-t és van neki megfelelő, akkor szól a Studentnek.
- **Össosztályok:** nincs
- **Interfészek:** Entity
- **Asszociációk:**
Person owner: Kinél van az adott tárgy. Ha szobában van, null.
Room room: Melyik szobában van az adott tárgy. Ha személynél van, null.
- **Attribútumok**
string name: A tárgy neve
int life: Hátralévő élettartam, ennyiszer használható még a tárgy.
bool active: Jelzi, hogy aktív-e az item.
- **Metódusok**
void acceptProtection(): A Person ezzel a hívással jelez ha elfogadja a tárgy védelmi ajánlatát.
void activate: A tárgy aktiválása.
void changeRoom(Room room): Beállítja a tárgy szobáját.
void destroy(): A tárgy megsemmisítése.
void gasThreat(): A Person ezzel a hívással jelez az összes általa birtokolt Item felé ha gázos szobában tartózkodik.

void setOwner(Person owner): Beállítja a tárgy tulajdonosát és kezeli azt az eseményt, hogy a tárgyat felvették, utóbbi néhány implementációban lényeges.

void teacherThreat(): A Person ezzel a hívással jelez az összes általa birtokolt Item felé ha egy oktató megtámadta őt.

4.3.7 Mask

- **Felelősség**
Ha egy hallgatót megtámad egy elgázosított szoba, akkor aktiválódik, és megvédi a hallgató életét.
- **Össztályok**
Item
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok:** nincs
- **Metódusok:**
 - void acceptProtection():** A Person elfogadja a védelmi ajánlatot, az active attribútum igaz értéket kap, a life értéke eggyel csökken.
 - void gasThreat():** A Mask értesül az elgázosított szobáról és védelmet kínál tulajdonosának.

4.3.8 Person

- **Felelősség**
A személyek (hallgatók és oktatók) absztrakt osztálya. Tárolja a nevét, és felelős a személy mozgatásáért, tárgyakkal való interakciókért, valamint az állapotaiért is felel. Szól a szobának, hogy belépett.
- **Össztályok:** nincs
- **Interfészek:** Entity
- **Asszociációk**
Room: A Person tudja, hogy melyik szobában van.
Item: Minden személynek van egy Item listája.
- **Attribútumok**
string name: Személy neve.
bool stunned: Megbénult állapot.
- **Metódusok:**
 - void clothStun():** Ezzel a hívással jelezzük ha a személyre egy rongy próbál hatni.
 - void drop(Item):** Eldobja a tárgyat a szobában, ahol van.
 - void dropAll():** A személy eldobja az összes általa birtokolt tárgyat abban a szobában ahol épp tartózkodik.
 - void gasProtection(Item protectionProvider,int priority):** Ezzel a hívással jelez tulajdonosának a védelmet nyújtó tárgy a gasThreat() hatására ha védelmet kínál gáz ellen, ezáltal tulajdonképpen védelmi ajánlatot tesz.
 - void gasStun():** Ezzel a hívással jelezzük ha a személyre gáz próbál hatni.
 - void move(Door):** A személy átlép a megadott ajtón ha ezt megteheti.
 - void pickup(Item):** Felveszi a tárgyat a szobából, ahol van.

void slideRuleNotifcation(Item slideRule): Ezt a metódust a SlideRule hívja ha a Person felvette őt. Ez implementációtól függően kezeli a SlideRule megszerzés eseményét.

void teacherAttack(): Ezzel a hívással jelezzük, ha a személyt egy oktató megtámadja.

void teacherProtection(Item protectionProvider, int priority): Ezzel a hívással jelez tulajdonosának a védelmet nyújtó tárgy a teacherThreat hatására ha védelmet kínál oktató ellen, ezáltal tulajdonképpen védelmi ajánlatot tesz.

void teleport(Room roomTo): A személy átkerül a paraméterben kapott szobába.

4.3.9 Room

- **Felelősség**
A játék szobáit reprezentálja. Számon tartja az adott szoba állapotát (elgázosított, elátkozott), a benne lévő tárgyakat és entitásokat, illetve képes osztódni vagy összeolvadni másik szobával.
- **Össosztályok:** nincs
- **Interfészek:** Entity
- **Asszociációk:** nincs
 - Person:** Tárolja a benne lévő személyeket.
 - Door:** Tárolja a hozzá tartozó ajtókat.
 - Item:** Tárolja a benne lévő tárgyakat.
- Attribútumok**
 - int id:** Szoba azonosítója.
 - int capacity:** A szoba befogadóképessége.
 - bool gased:** El van-e gázosodva a szoba.
 - bool cursed:** El van-e átkozva a szoba.
 - bool full:** Tele van-e a szoba (entitásokkal)
- **Metódusok:**
 - void addDoor(Door):** A paraméterként kapott ajtót hozzáadja a szobához.
 - void addItem(Item):** A paraméterként kapott tárgyat hozzáadja a szobához.
 - void addPerson(Person person):** A paraméterként kapott személyt hozzáadja a szobához.
 - void destroy():** A szoba megszűnését jelentő függvény.
 - void divide():** A szoba két részre osztódik.
 - void gas():** A szoba elgázosodik.
 - void merge():** Összeolvad egy találmányra kiválasztott szomszédjával.
 - void removeDoor(Door):** A paraméterként kapott ajtót eltávolítja a szobából.
 - void removeItem(Item):** A paraméterként kapott tárgyat eltávolítja a szobából.
 - void removePerson(Person person):** A paraméterként kapott személyt kiveszi a szobából.
 - void tick():** Hatására a szoba bizonyos valószínűséggel osztódik vagy összeolvad egy szomszédjával.

4.3.10 SlideRule

- **Felelősség**
Szól a Person-nek aki felvette, hogy ez egy SlideRule. Az oktató vagy hallgató eszerint kezeli a tárgyat.
- **Ősosztályok**
Item
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok:**
- **Metódusok:**
void setOwner(Person): Ha egy Person felveszi, akkor slideRuleNotification-t küld a számára.

4.3.11 Student

- **Felelősség**
A hallgatót megvalósító osztály.
- **Ősosztályok**
Person
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok**
- **Metódusok:**
void clothStun(): Immunis a ronggyal szemben, ez egy üres implementáció ami nem csinál semmit.
void gasStun(): A gáz ellen megpróbál védekezni (tárgyak felé gasThreat() jelzés), ha nem kap védelmet akkor elejti tárgyait és megbénul 2 körig.
void gasProtection(Item protectionProvider,int priority): Ezzel a hívással a hallgató védelmi ajánlatot kap gázzal szemben, elfogadja a legnagyobb prioritású ajánlatot.
void pickup(Item): A felvett tárgyat eltárolja ha van nála hely.
void teacherAttack(): Az oktató támadása ellen megpróbál védekezni (tárgyak felé teacherThreat() jelzés), ha nem kap védelmet, meghal.
void teacherProtection(Item protectionProvider,int priority): Ezzel a hívással a hallgató védelmi ajánlatot kap oktatóval szemben, elfogadja a legnagyobb prioritású ajánlatot.

4.3.12 Teacher

- **Felelősség**
Az oktatót megvalósító osztály. Felelős a támadásaiért. Mivel nem tudja, hogy milyen típusú Person-ök vannak a szobában, ezért mindenkit megtámad, de csak a Studentekre hat valójában.
- **Ősosztályok**
Person
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok:** nincs
- **Metódusok**

void clothStun(): A rongy hatására megbénul 3 körig.
void gasStun(): A gáz hatására megbénul 2 körig.
void initAttack(): Megtámadja a vele egy szobában tartózkodó személyeket.
void pickup(Item): A felvett tárgyat megsemmisíti.
void teacherAttack(): Oktató támadására immunis, ez az implementáció semmit nem csinál.

4.3.13 Transistor

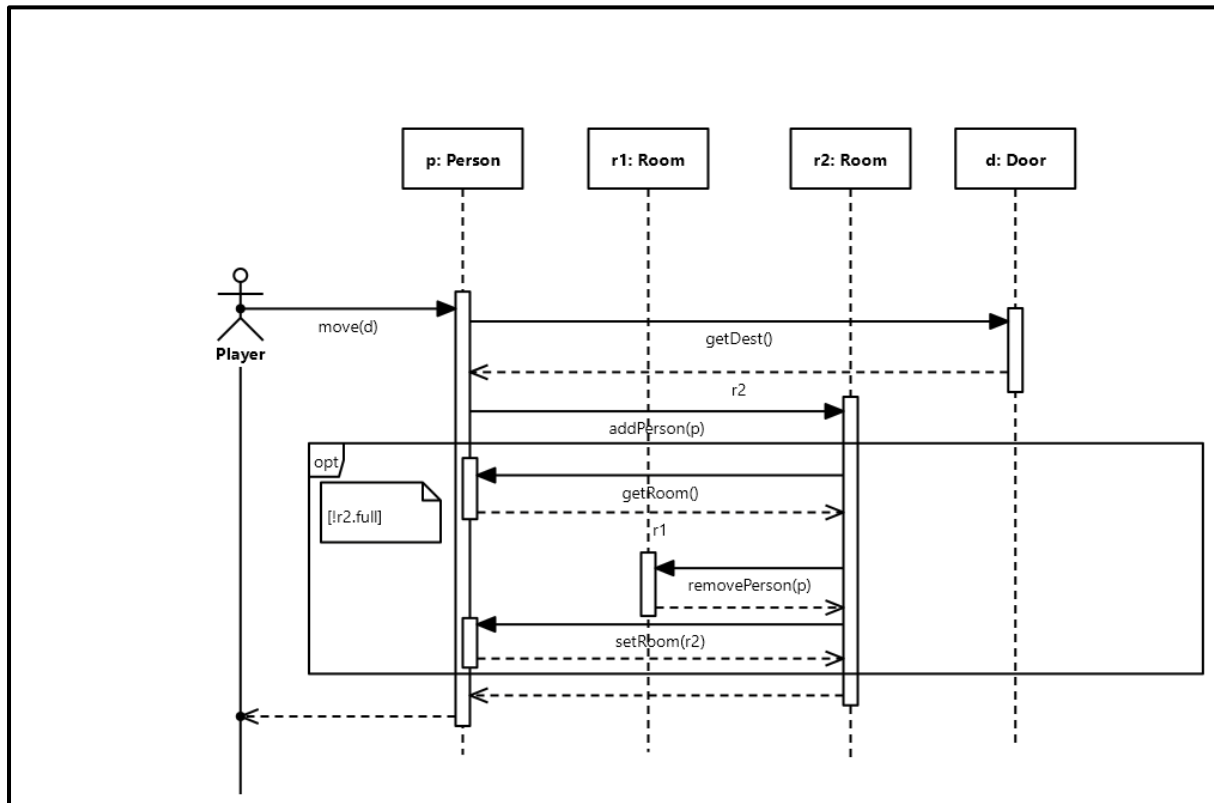
- **Felelősség**
A tranzisztort megvalósító osztály. Két tranzisztort össze lehet kötni, mindegyik számon tartja párját. Ha létrejött egy aktív pár, akkor elmozdítja a Studentet a másik tranzistor szobájába. Deaktiválja magát és a párját ezután. Ha egy Teacher veszi fel, akkor reseteli a másik tranzisztort.
- **Össztályok**
Item
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok**
int id: Ha felvesznek egy tranzisztort és aktiválják, kap egy id-t az aktiválásuk sorrendjében, és ezek szerint az id-k szerint fognak összekapcsolódni.
Transistor pair: Az összekapcsolt tranzistor párja.
- **Metódusok**
void activate(): A tranzistor bekapcsolása. Ha még nincs párja akkor aktiváláskor id-t kap.
void deactivate(): Ezt a függvényt hívja meg a tranzistor magán és a párján, amikor a hallgató elteleportál.
void reset(): A párja hívja meg ezt a függvényt, amikor elpusztul (oktató megsemmisíti). Ekkor a tranzistor deaktiválódik, amíg újra fel nem veszi egy hallgató és össze nem köti egy új tranzisztorttal.

4.3.14 TVSZ

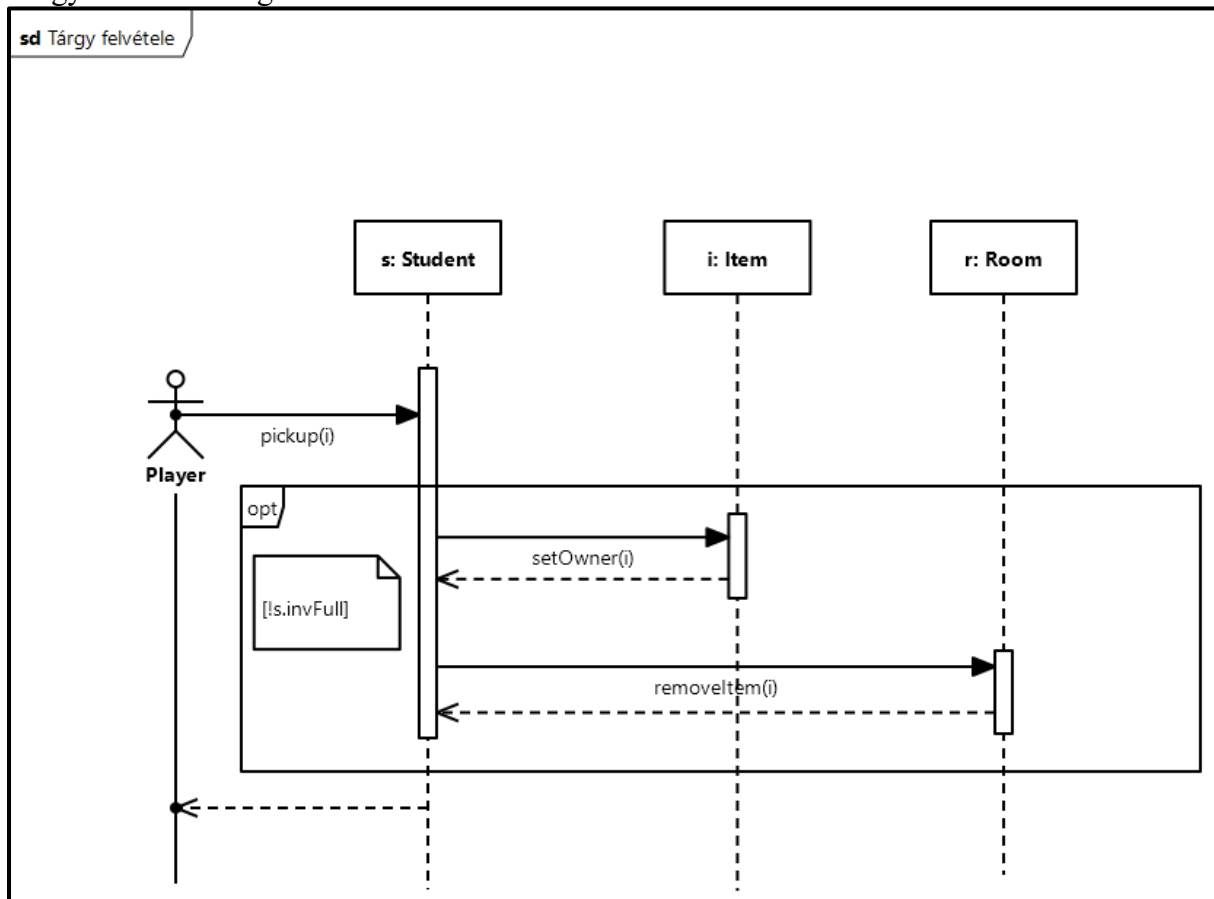
- **Felelősség**
Ha az őt birtokló hallgatót megtámadja egy oktató és nincs magasabb prioritású védelme (Beer), akkor aktiválódik, ezzel megvédve a hallgatót. Számontartja, hogy hány alkalommal képes még megvédeni a hallgatót, mielőtt elhasználná.
- **Össztályok**
Item
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok:** nincs
- **Metódusok:**
void teacherThreat(): Értesül az oktató támadásáról és védelmet kínál tulajdonosának.
void acceptProtection(): A Person elfogadja a védelmi ajánlatot, az active attribútum igaz értéket kap, a lifeTime értéke elkezd tick() hatására csökkenni.

4.4 Szekvencia diagramok

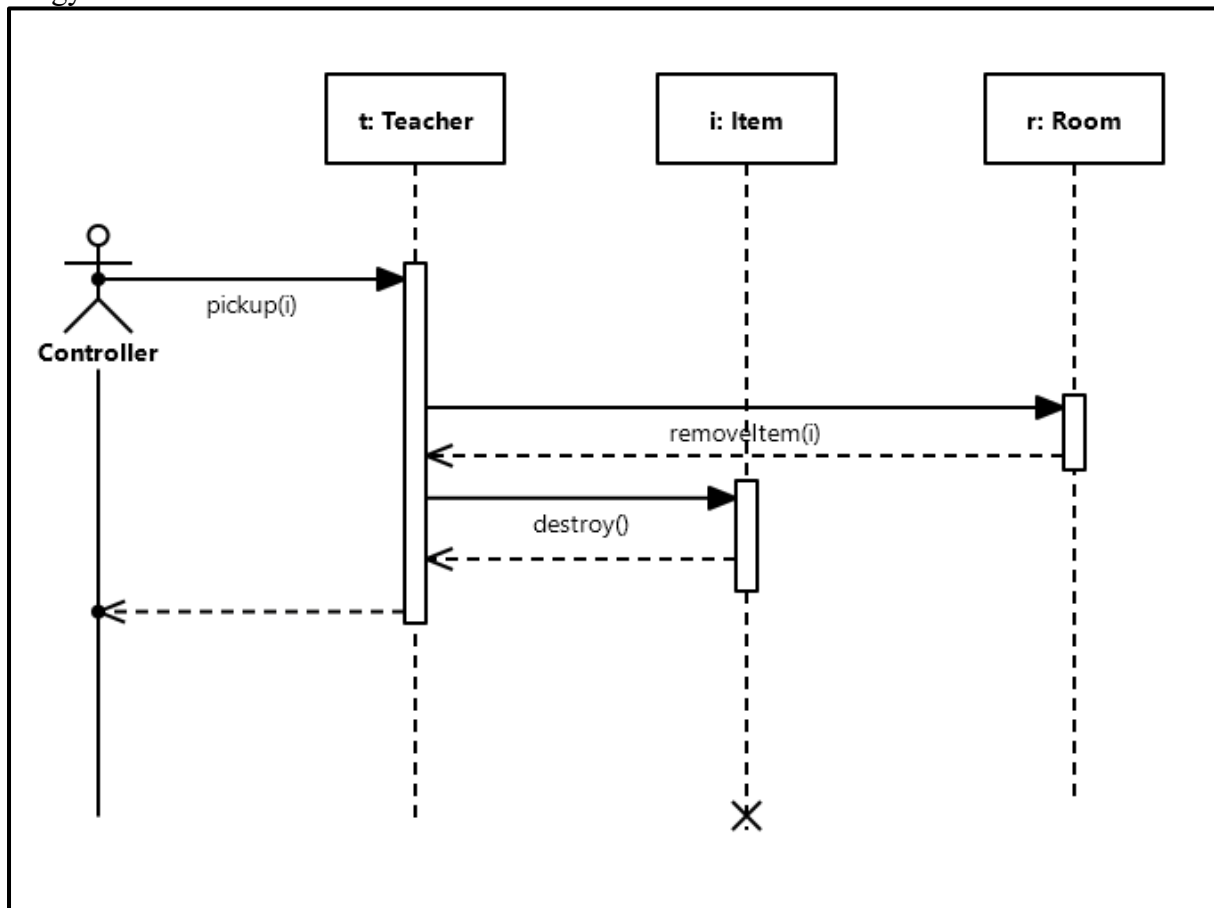
Person szobát vált:



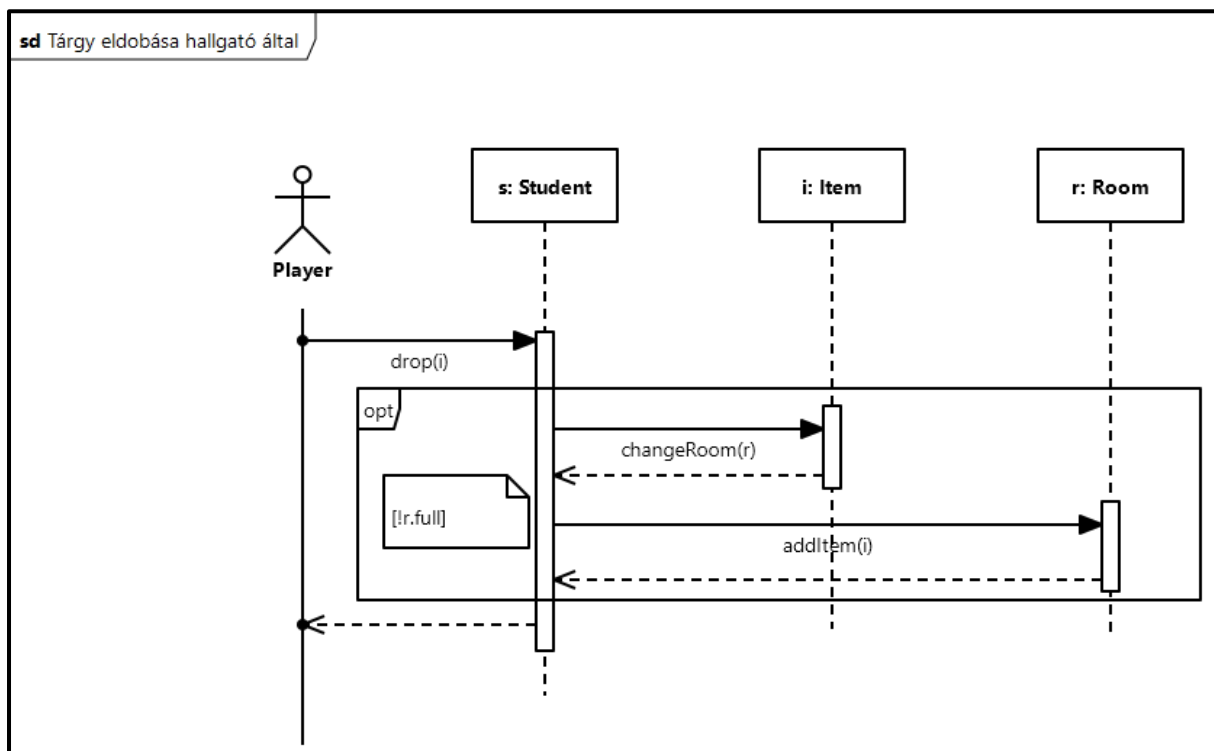
Tárgy felvétele hallgató által:



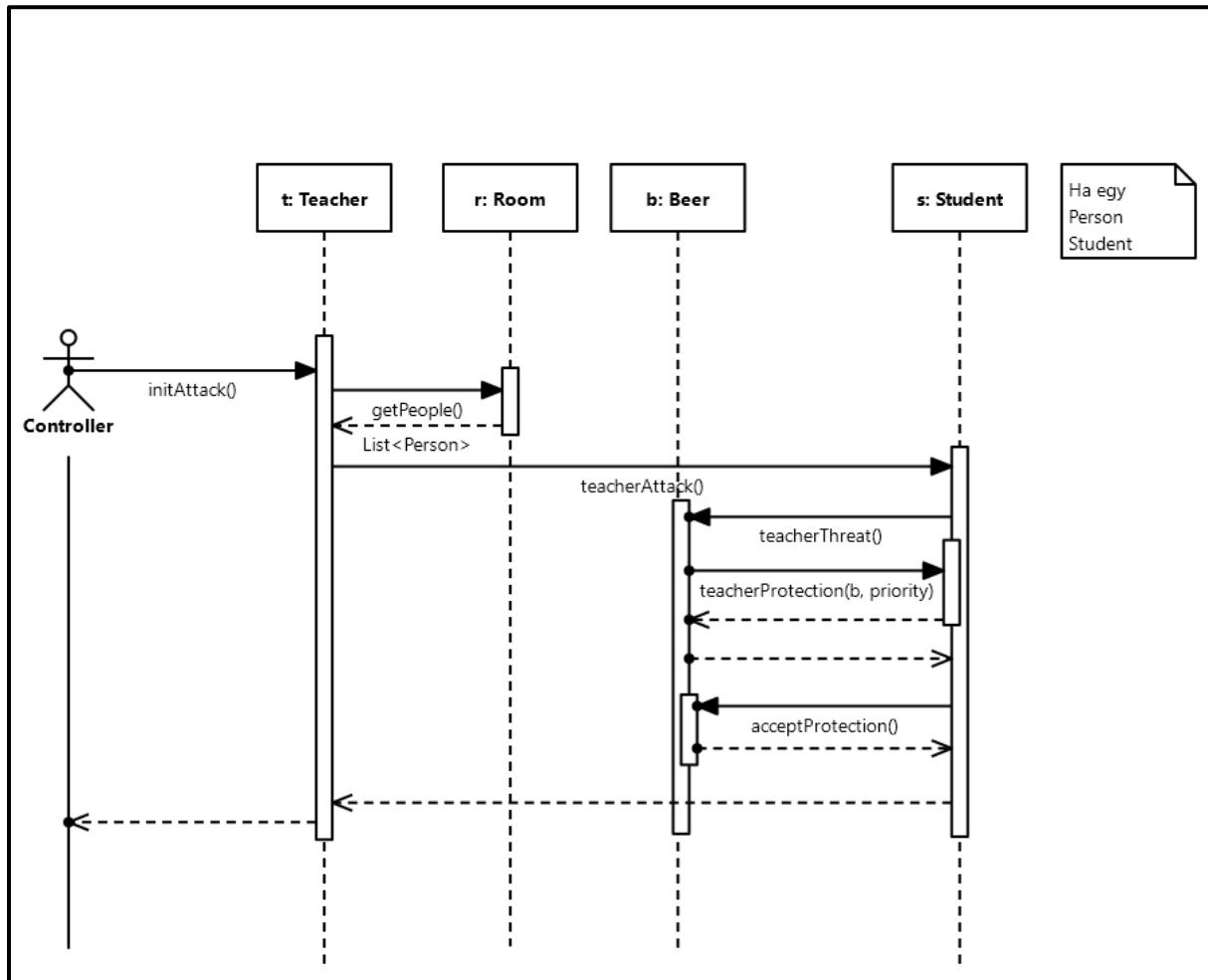
Tárgy felvétele oktató által:



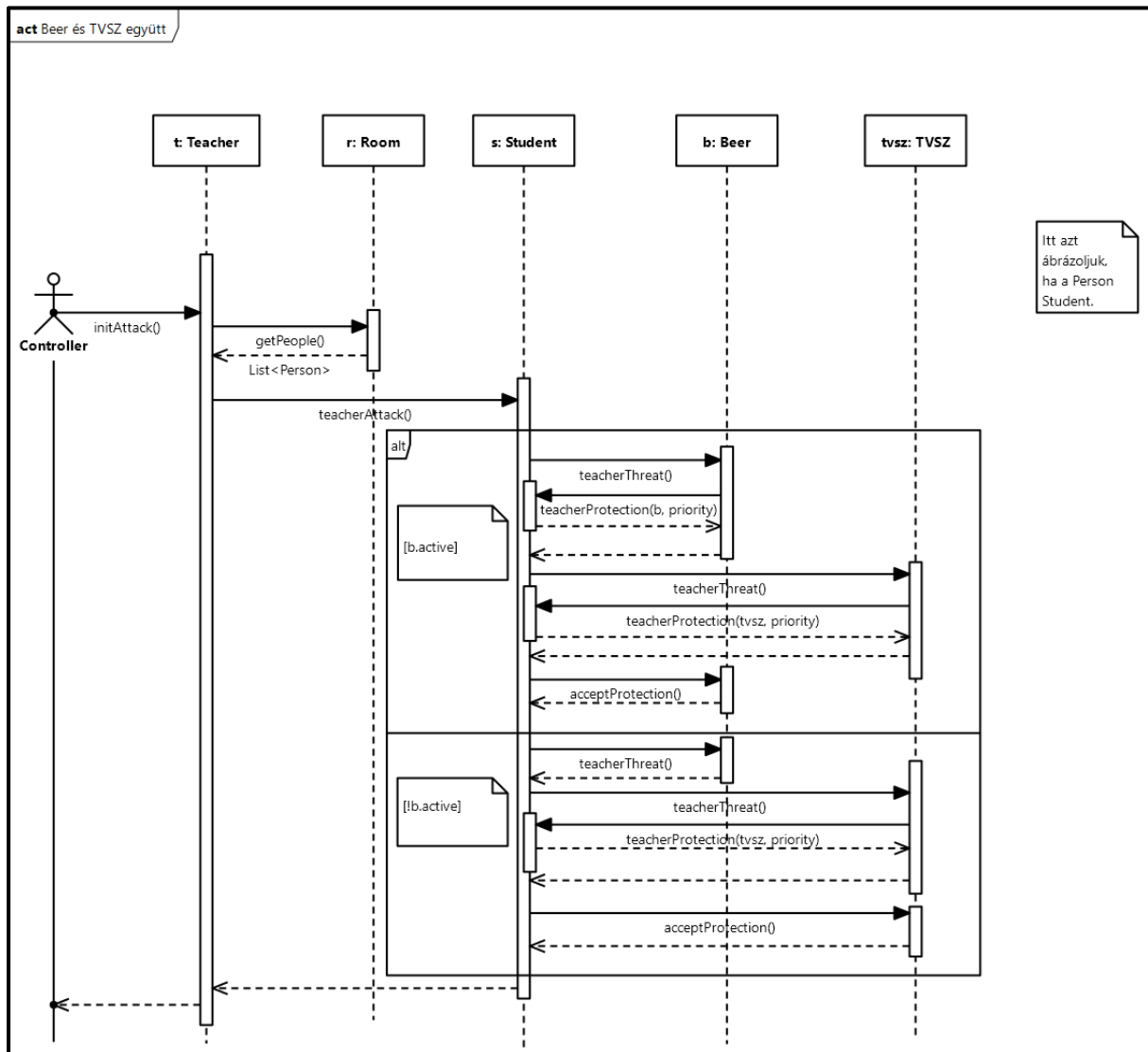
Tárgy eldobása hallgató által:



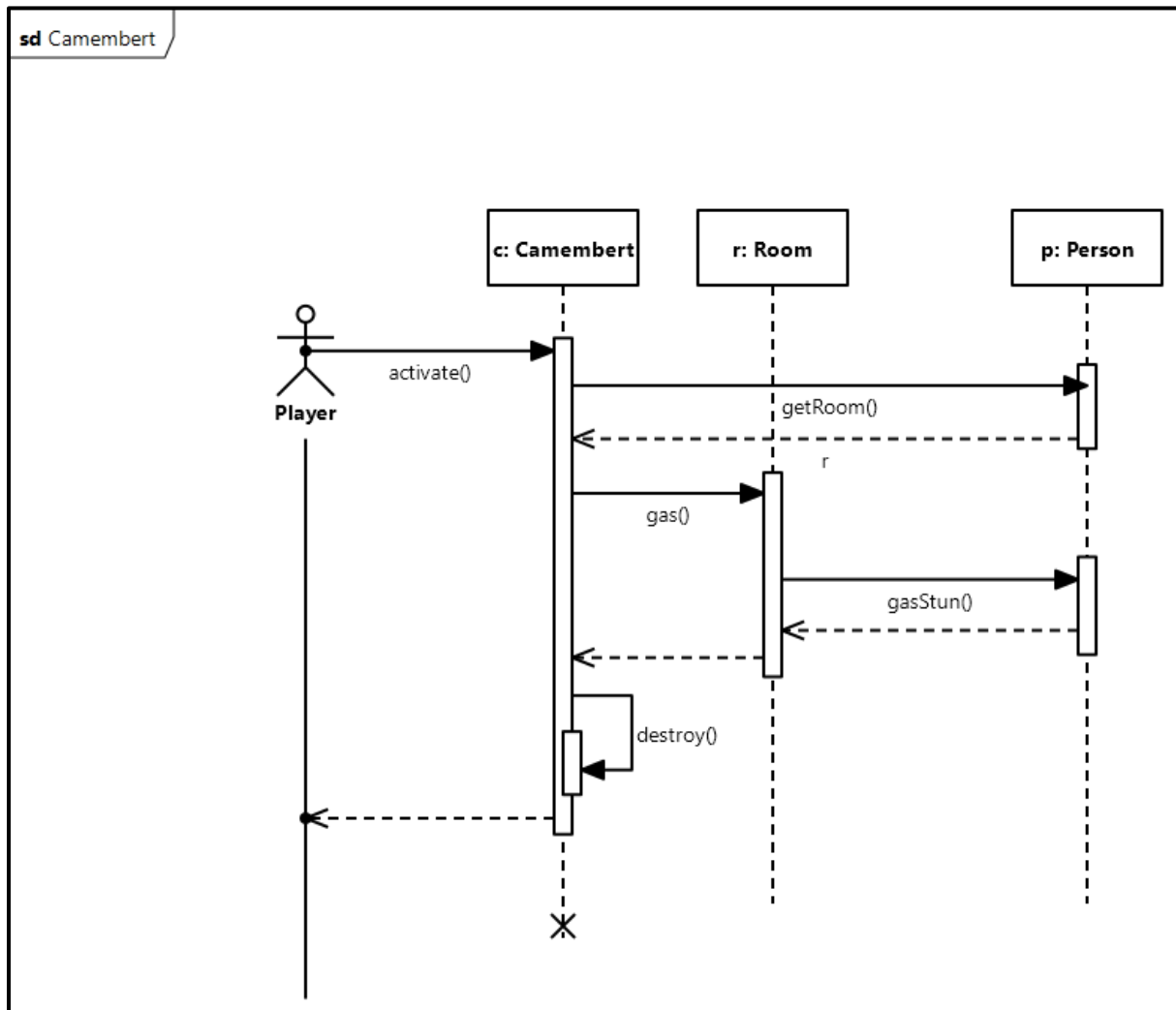
Beer használata:



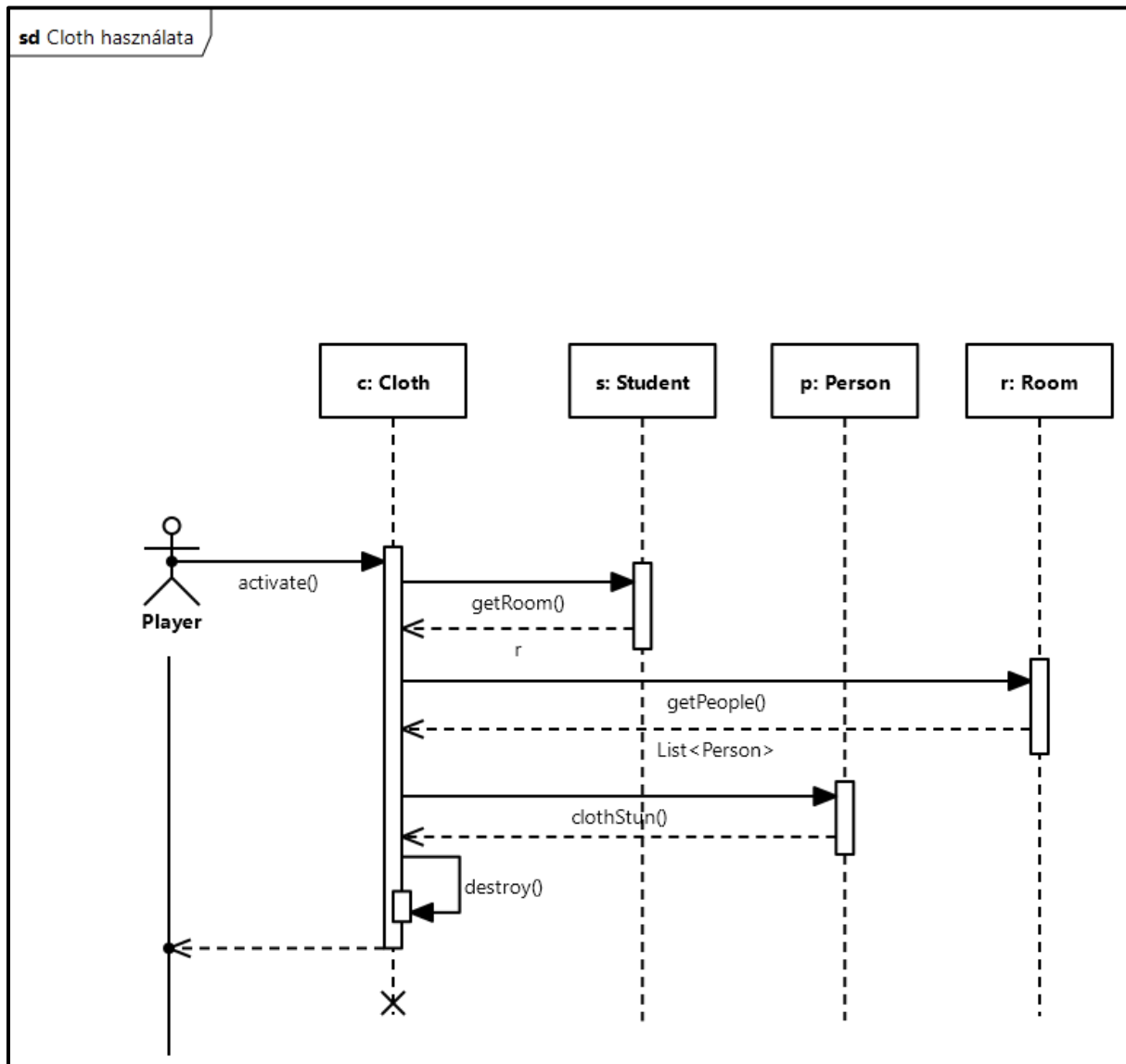
Beer és TVSZ is van:



Camembert használata:

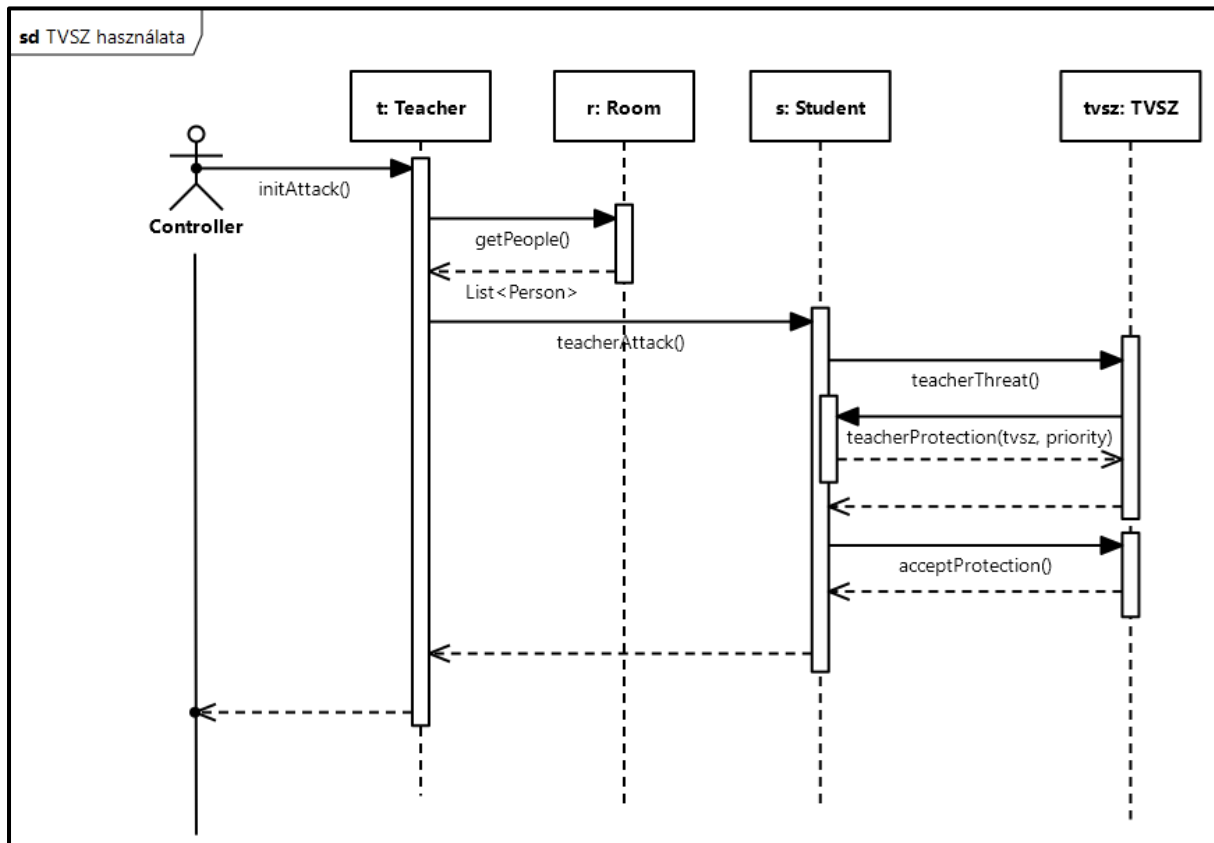


Cloth használata:

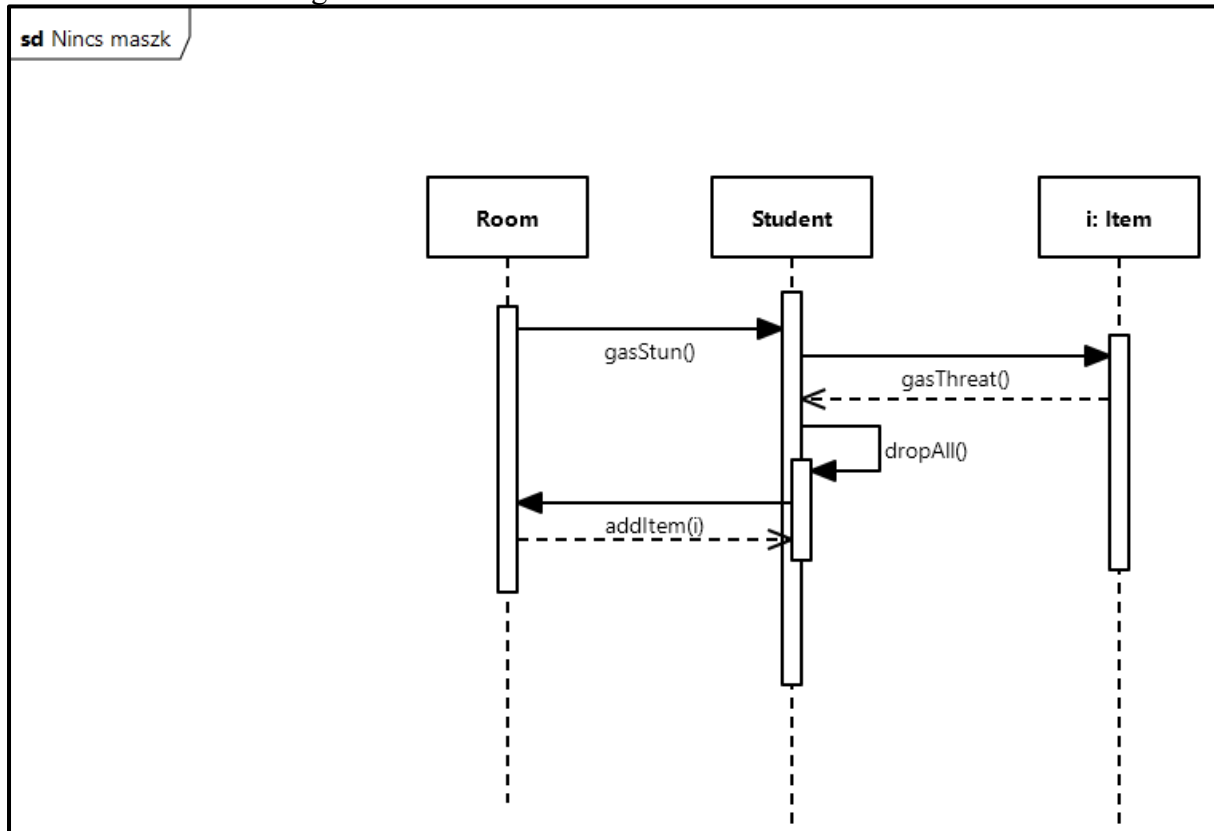


(Interakciók szempontjából nincs különbség a Teacher és a Student között a clothStun hatására csak a metóduson belül van különbség.)

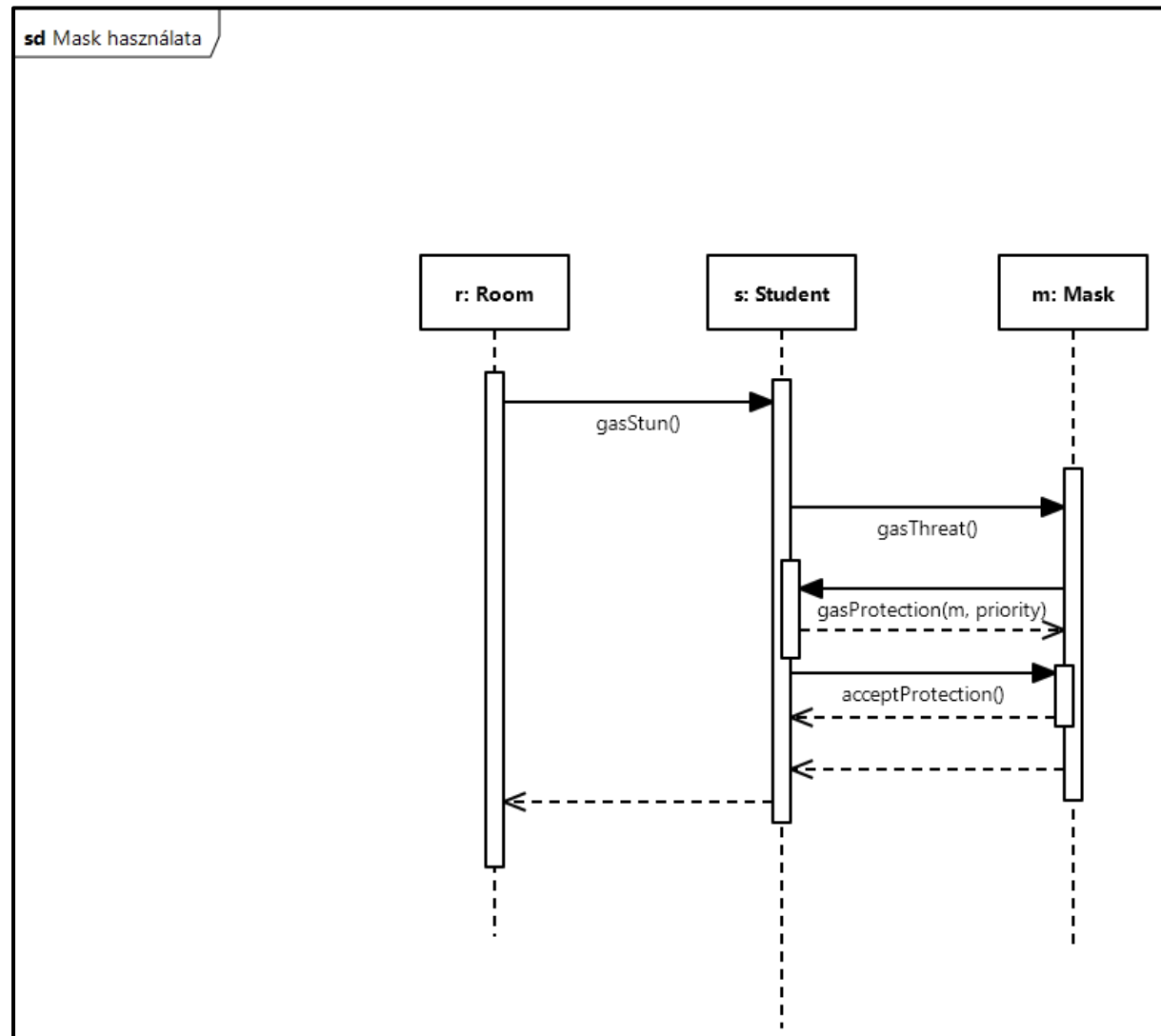
TVSZ használata:



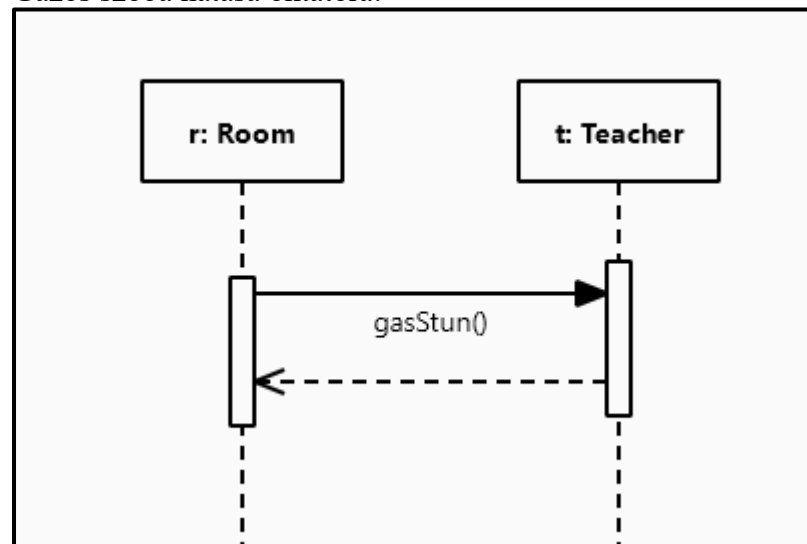
Gázos szoba hatása hallgatóra:



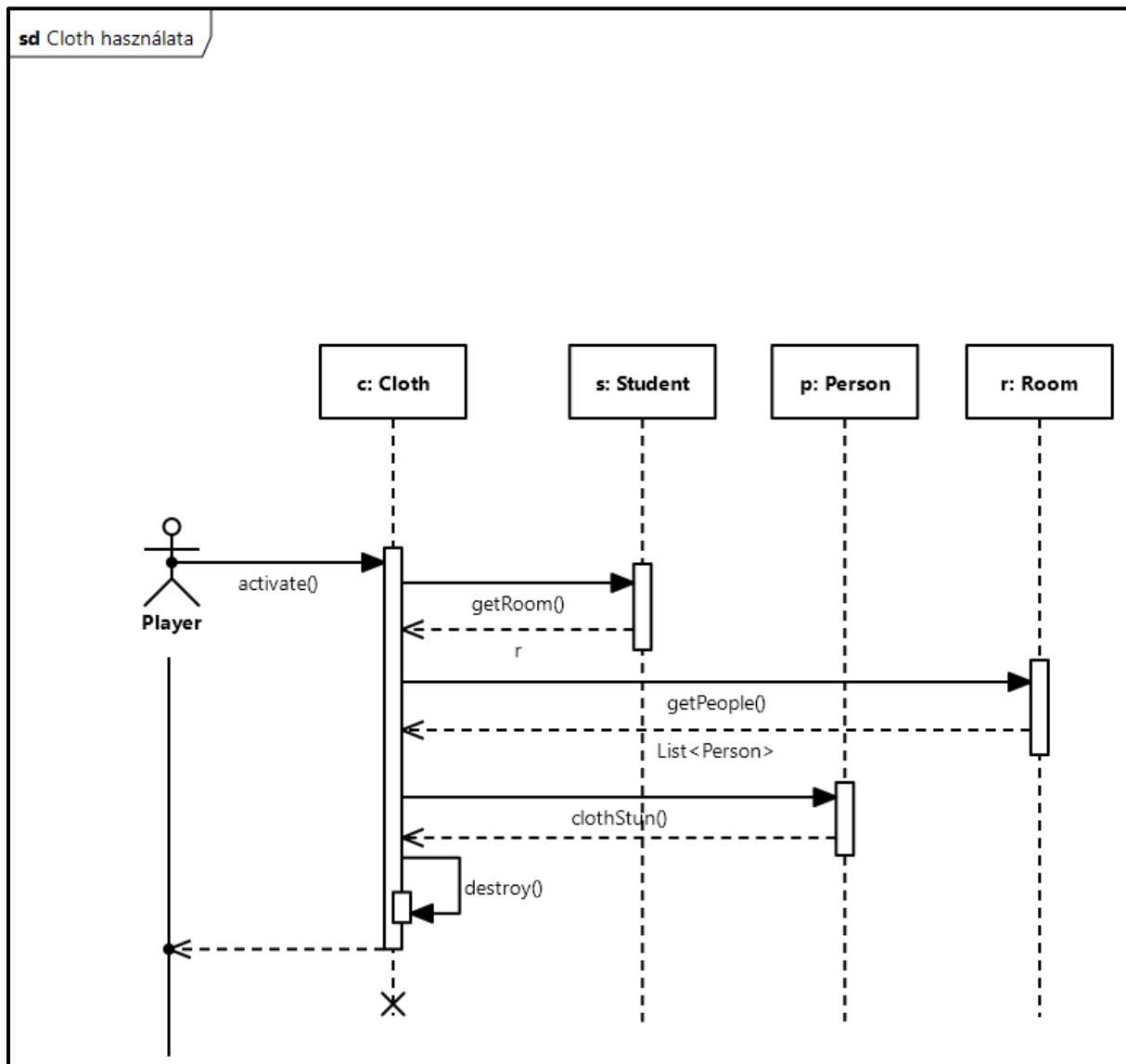
Mask használata:



Gázos szoba hatása oktatóra:

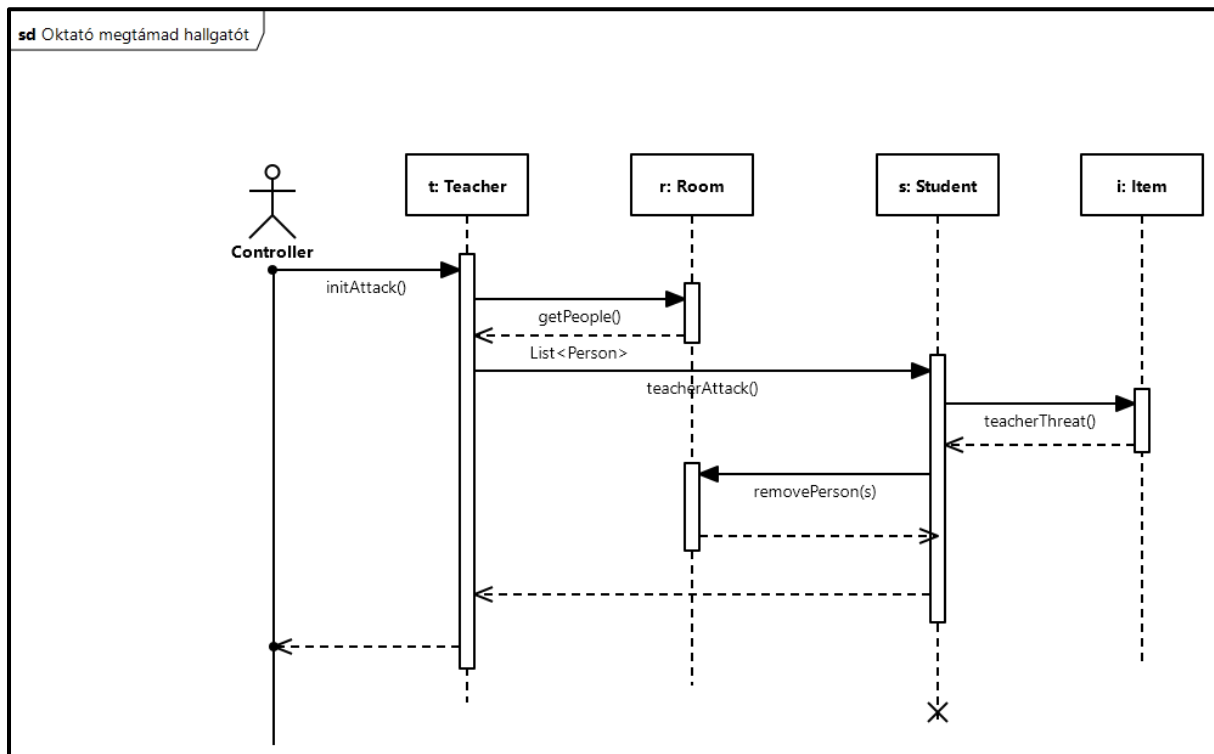


Rongy használata:

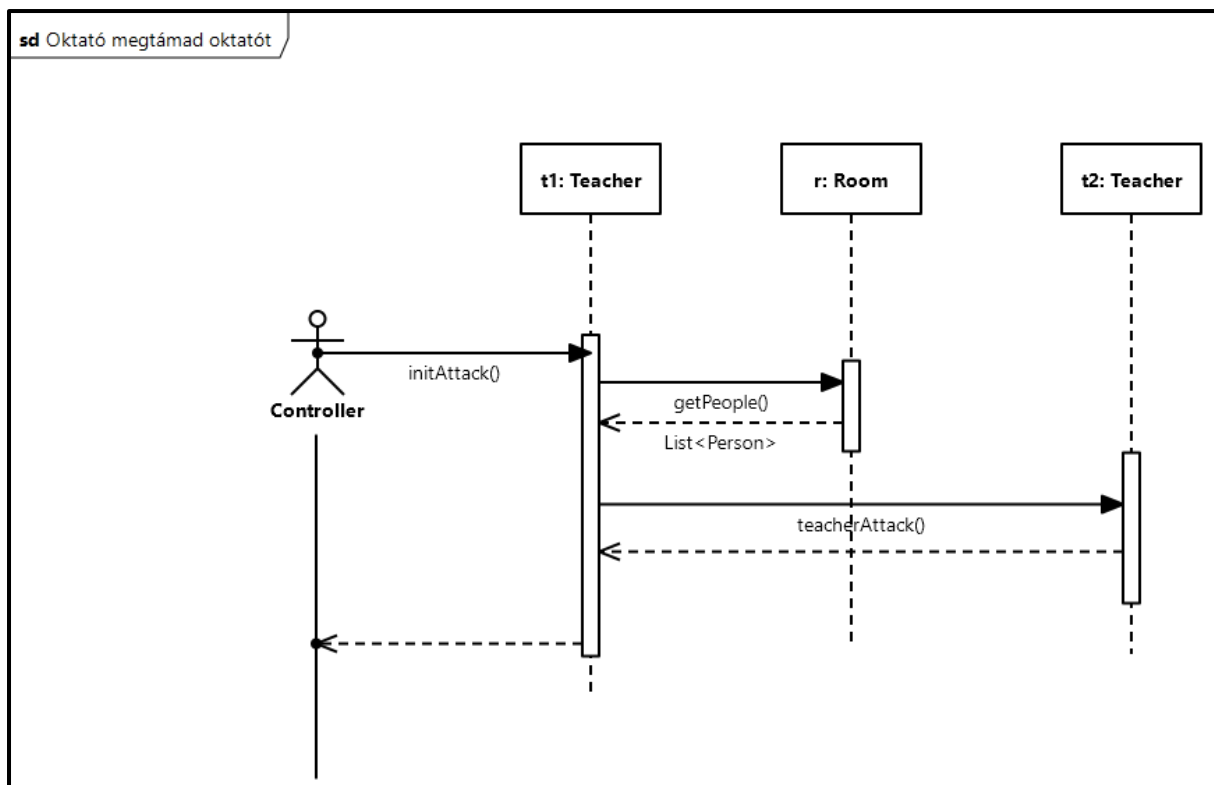


(A rongy hatásának interakciók szempontjából nincs különbség Teacher és Student között, csupán a függvény belsejében)

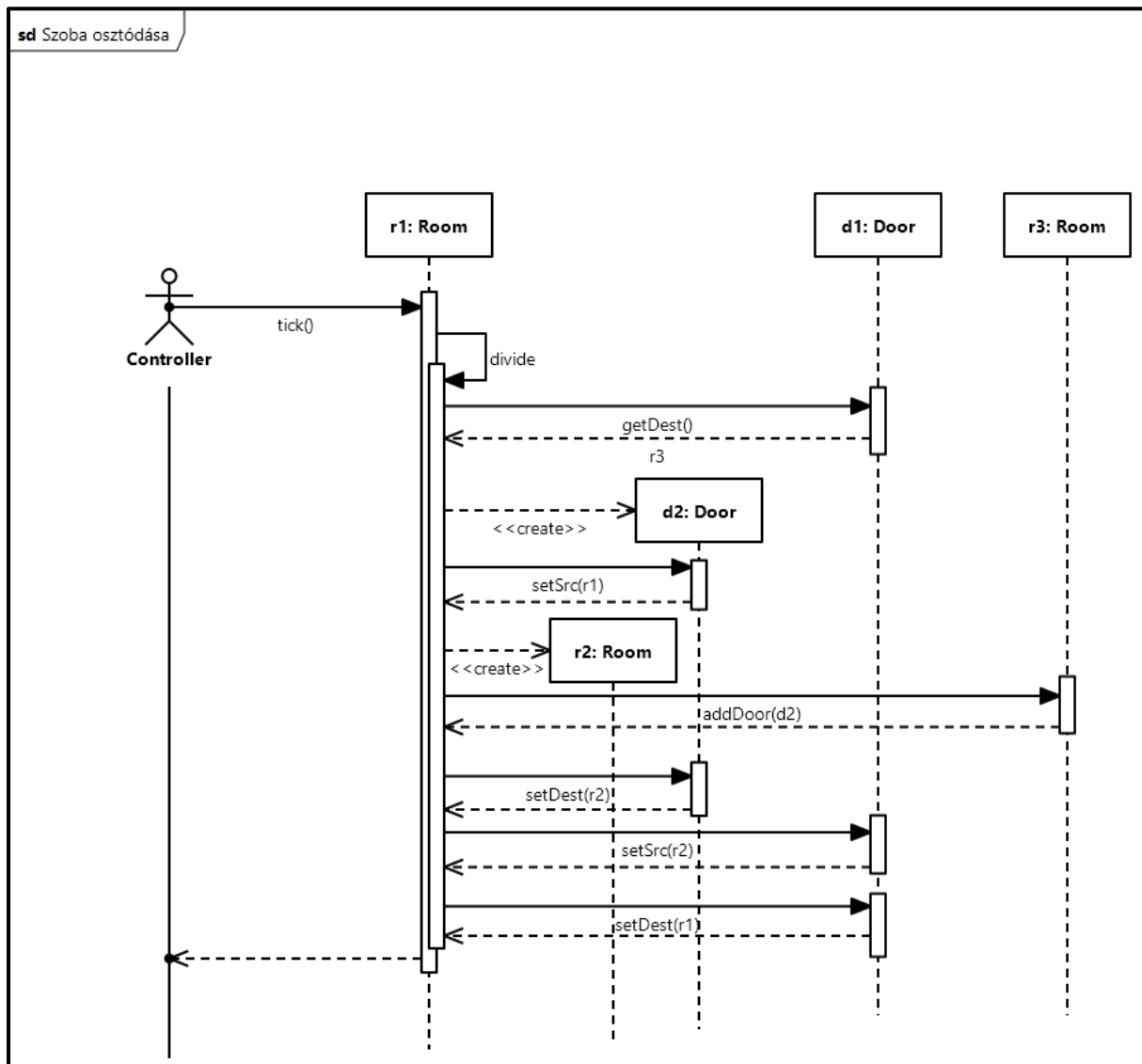
Oktató megtámadja hallgatót, akinek nincs védelme:



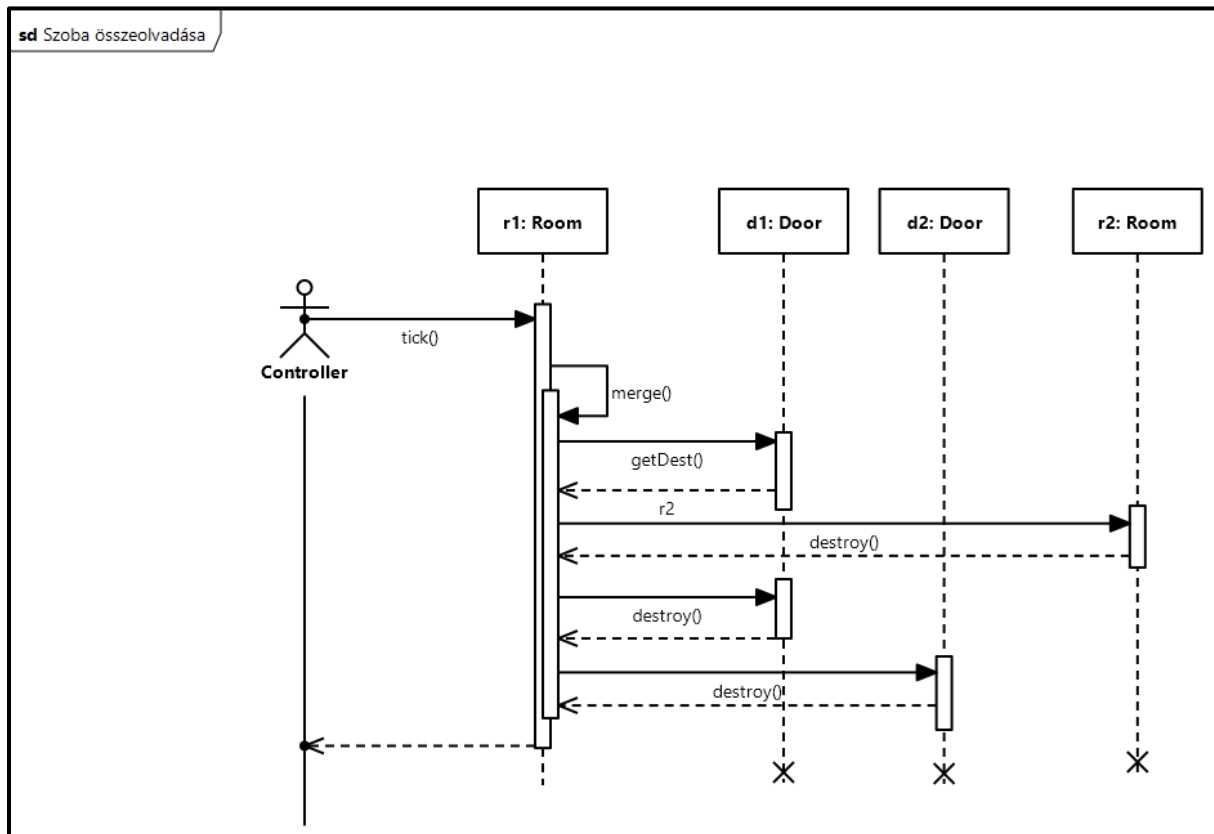
Oktató megtámadja oktatót:



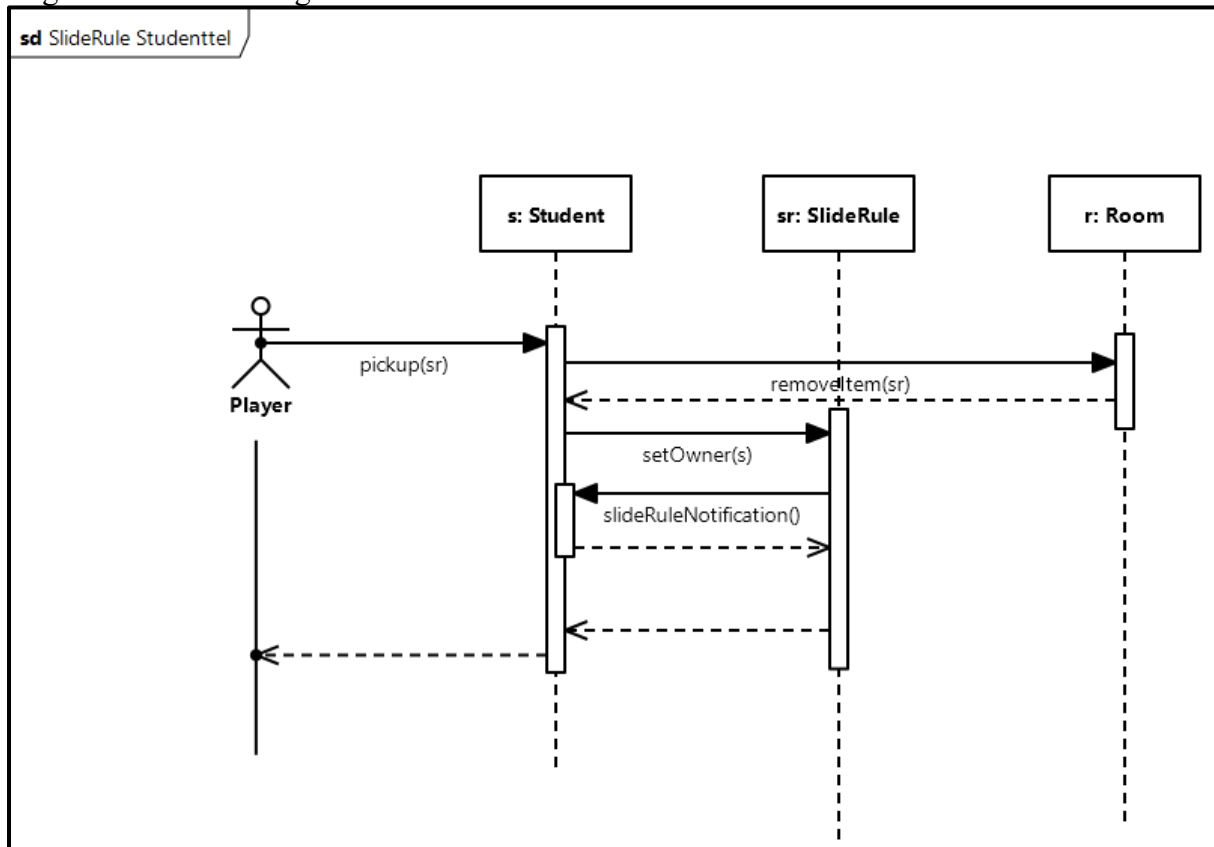
Szoba osztódás:



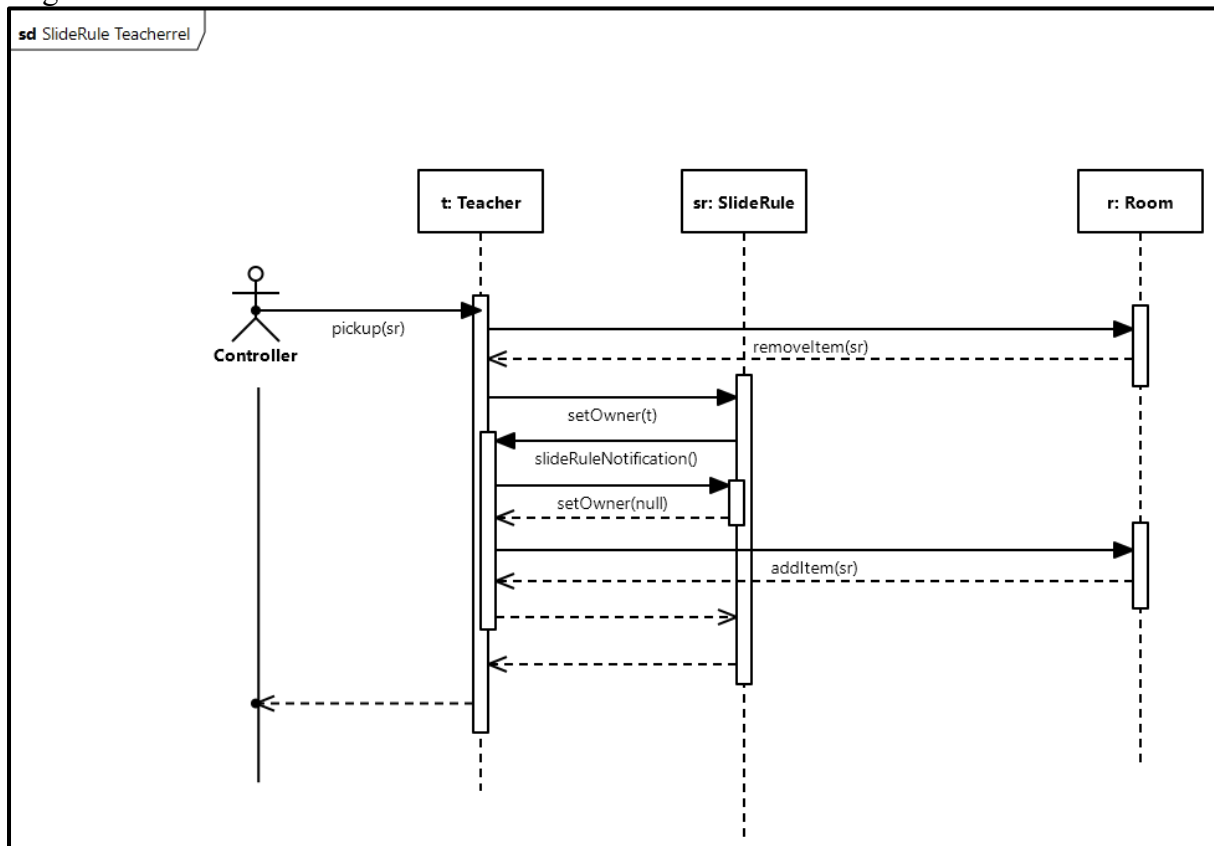
Szoba összeolvasás:



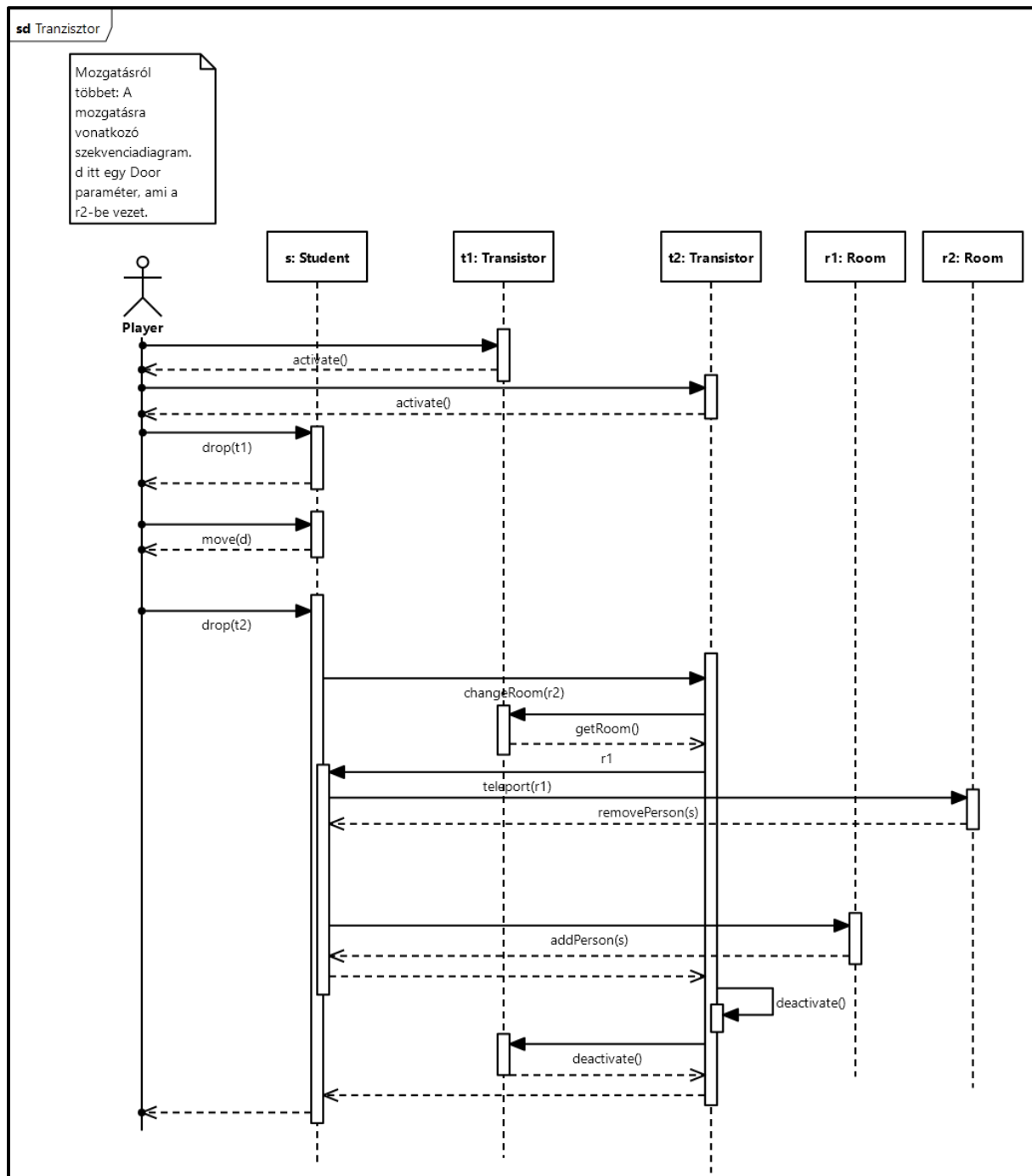
Logarléc felvétele hallgató által:



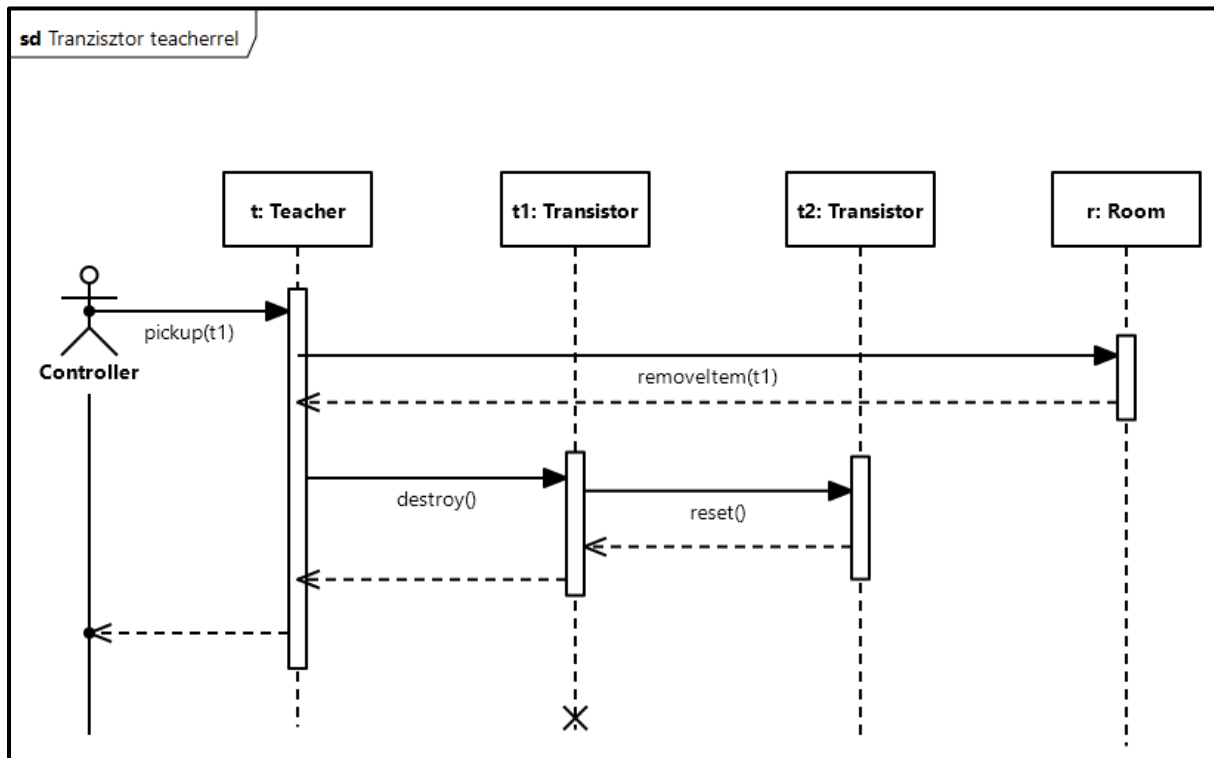
Logarléc felvétele oktató által:



Tranzisztor:

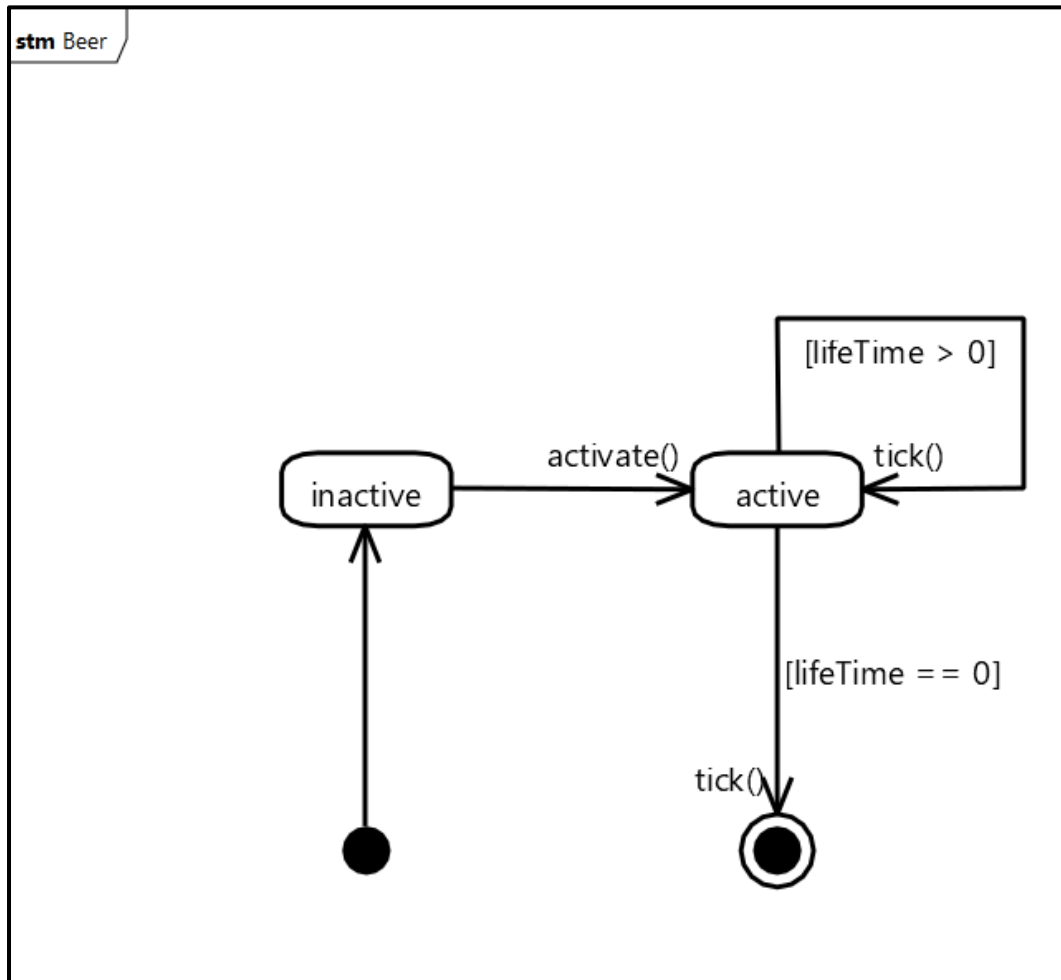


Tranzisztor teacherrel

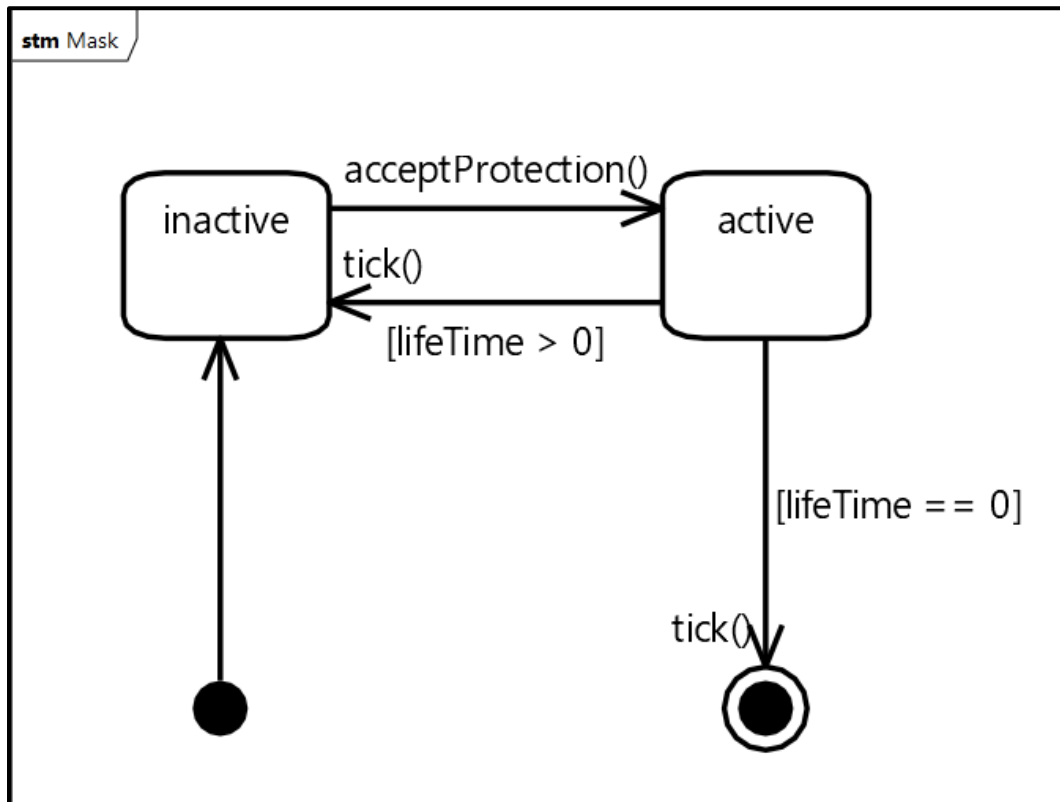


4.5 State-chartok

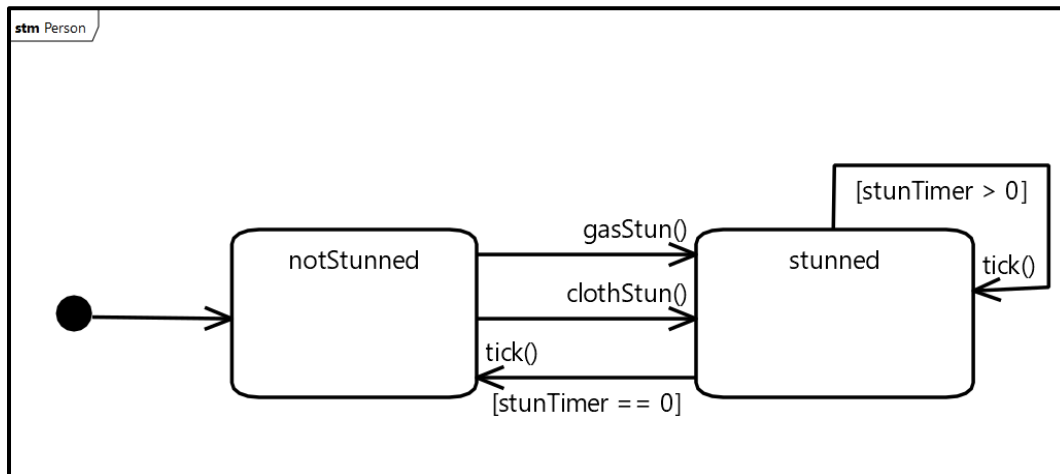
Beer állapotai:



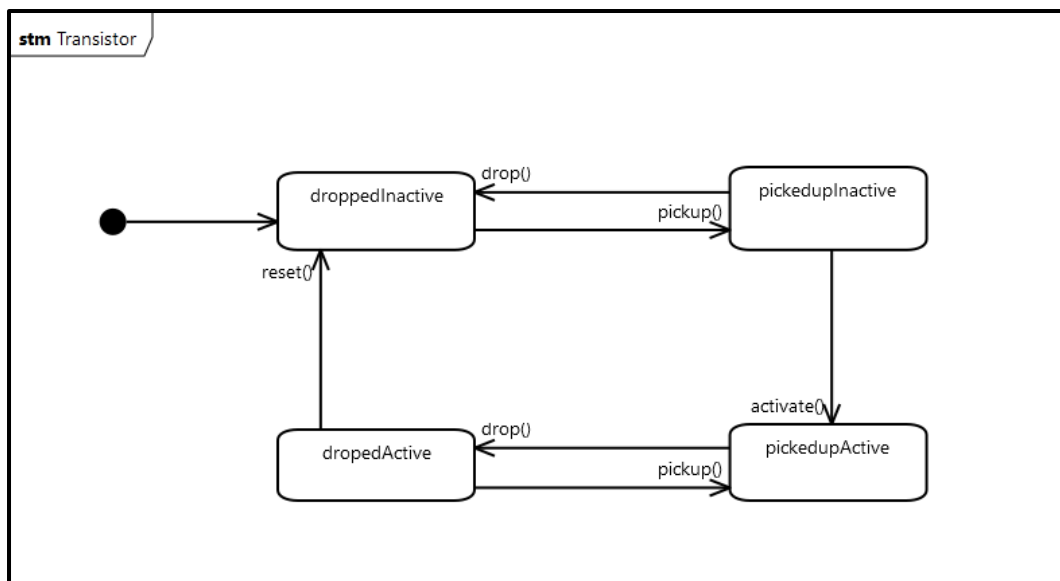
Mask állapotai:



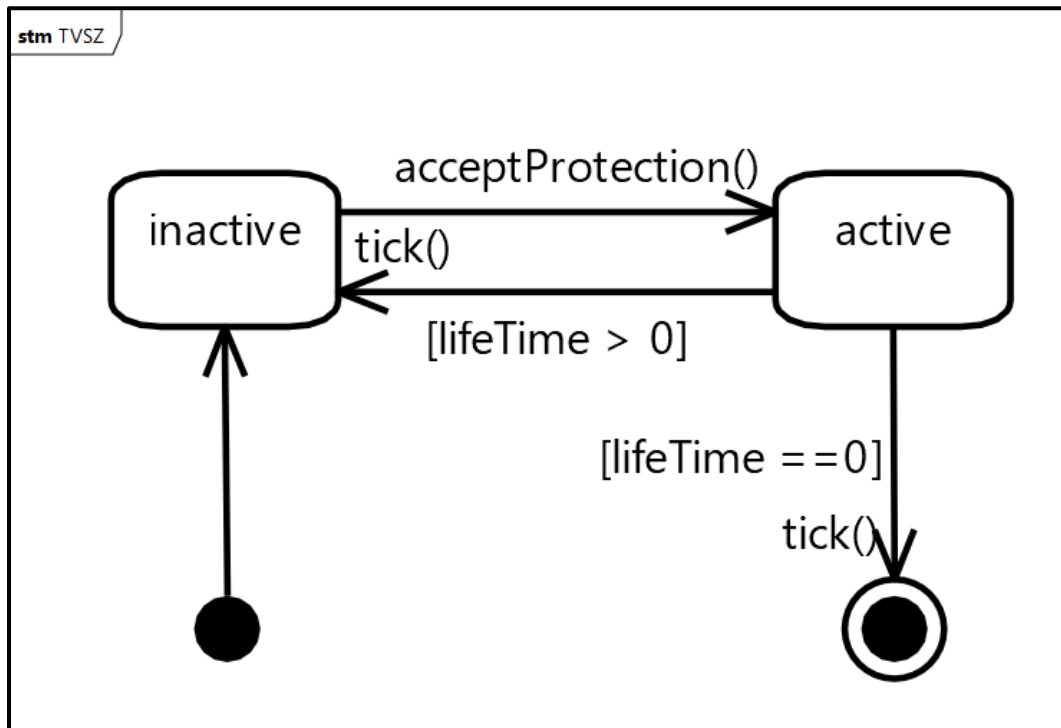
Person állapotai:



Tranzisztor állapotai:



TVSZ állapotai:



4.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.03.06. 15:00	30p	Görömbey Riba Szakos	Egyeztettek a laboralkalmon elhangzott visszajelzésekről.
2024.03.08. 13:00	2,5 óra	Cardinael Görömbey Riba	Szekvenciadiagramok észítése mozgáshoz és tárgyakhoz.
2024.03.09. 16:00	2,5 óra	Cardinael Görömbey Szakos	Szekvenciadiagramok készítése.
2024.03.09. 20:00	1,5 óra	Cardinael Görömbey Szakos	Szekvencia.
2024.03.09. 21:30	1,5 óra	Cardinael Görömbey	Szekvenciadiagramok folytatása.
2024.03.10. 11:00	1,5 óra	Cardinael Görömbey Riba	Szekvenciadiagramok folytatása.
2024.03.10. 14:00	1,5 óra	Cardinael Görömbey Riba	Szekvenciadiagramok folytatása és állapotgépek elkezdése.
2024.03.10. 19:00	1,5 óra	Riba	Állapotgépek.
2024.03.10. 21:00	3 óra	Cardinael Riba	Szekvenciadiagramok ban lévő inkozisztenciák megszüntetése.
2024.03.10. 20:00	4 óra	Szakos	Osztályok leírása, osztálydiagram.
2024.03.11 9:00	1,5 óra	Cardinael	Szekvenciák javítása, véglegesítése, nyomtatás