

3. Analízis modell kidolgozása I.

18 – Mumbai IT Solution

Konzulens:
Goldschmidt Balázs

Csapattagok

Cardinael Jan Maerten	M1GVE2	cardinael.jan@gmail.com
Görömbey Lilla Zsófia	EBQG85	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

2024.03.04.

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Beer

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

3.1.2 Camembert

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

3.1.3 Cloth

Felvehető tárgy, ha a játékos birtokában van és aktiválja, akkor a szobában lévő oktatókat megbénítja. Adott ideig működik a hatása.

3.1.4 Door

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

3.1.5 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.

3.1.6 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 osztódni. Két szomszédos szoba egyesülésével létrejövő szoba a korábbi két szoba tulajdonságaival, szomszédaival, 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 megosztóznak a korábbi szoba tulajdonságain és szomszédain. 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.

3.1.7 SlideRule

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

3.1.8 Student

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

3.1.9 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 játéktérrel, ezzel is nehezítve a hallgatók dolgát.

3.1.10 Transistor

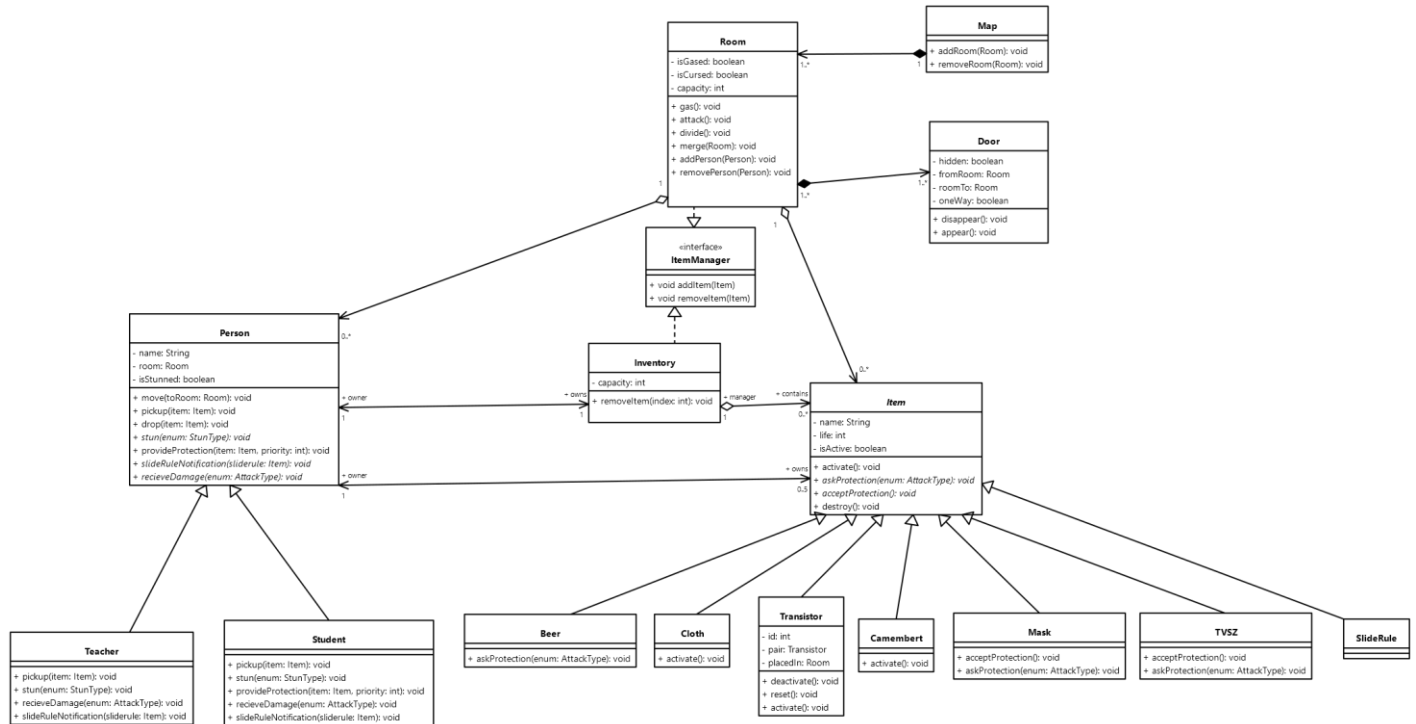
A hallgatónál lévő tranzisztorokat páronként össze lehet kapcsolni. A tranzisztorok az aktiválásuk, azaz bekapcsolásuk sorrendjében alkotnak párokat. Az egyik szobában be kell kapcsolni és letenni az egyik tranzisztort majd a pár másik tagját menet közben egy másik szobában le lehet tenni. Az így összekapcsolt 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.

3.1.11 TVSZ

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

3.2 Statikus struktúra diagramok

cls Class Diagram



3.3 Osztályok leírása

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

3.3.1 Beer

- Felelősség**

Ha a hallgató aktiválja, akkor védelmet nyújt neki az oktatók ellen. Számontartja, hogy meddig van érvényben.

- Ősz osztályok**

Item

- Interfészek:** nincs
- Asszociációk:** nincs
- Attribútumok:** nincs
- Metódusok:** nincs

3.3.2 Camembert

- Felelősség**

Ha aktiválódik, akkor elgázosítja azt a szobát, amelyikben van. Jelzi a szobának, hogy el van gázosítva.

- **Ősosztályok:** Item
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok:** nincs
- **Metódusok**
activate(): Meghívja a szoba gas függvényét.

3.3.3 Cloth

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

3.3.4 Door

- **Felelősség**
Átjárást biztosít szobák között a személyek számára, csak az egyik vagy mindkét irányban. Tudja hogy melyik szobák között van.
- **Ősosztályok:** nincs
- **Interfészek:** nincs
- **Asszociációk:**
Room roomFrom: Szomszédos szoba. Egyirányúság esetén innen nyílik.
Room roomTo: Szomszédos szoba. Egyirányúság esetén ide vezet.
- **Attribútumok:**
bool hidden: Használható-e az adott ajtó. (eltűnt-e)
bool oneWay: Egyirányú az ajtó.
- **Metódusok:**

3.3.5 Inventory

- **Felelősség**
Egy adott személynél lévő Item-eket tárolja, maximum 5 darabot.
- **Ősosztályok:** nincs
- **Interfészek:** ItemManager
- **Asszociációk**
Person owner: Ehhez a személyhez tartozik.
Item[] items: ebben tárolja a személynél lévő Item-eket.
- **Attribútumok**
int capacity: Maximum tárolási kapacitás.
- **Metódusok**
void removeItem(int index): Adott indexű tárgy eltávolítása.

3.3.6 Item

- **Felelősség**

A tárgyak absztrakt őosztálya. 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.

- **Őosztályok:**nincs
- **Interfészek:** nincs
- **Asszociációk:**
 - Person owner:** Kinél van az adott tárgy. Ha szobában van, null.
 - ItemManager manager:** Az ItemManager ami a tárgyat tárolja.
- Attribútumok**
 - string name:** A tárgy neve
 - int life:** Hátralévő élettartam, ennyiszer használható még a tárgy.
 - bool isActive:** Jelzi, hogy aktív-e az item.
- **Metódusok**
 - void activate:** A tárgy aktiválása
 - void askProtection(enum:AttackType):** A Person ezt a függvényt hívja meg veszély esetén minden tárgyon melyet birtokol és a megfelelő tárgyak válaszolnak a kérésre.
 - void acceptProtection:** A Person ezt a függvényt hívja meg, ha elfogadja egy tárgy védelmi ajánlatát.
 - void destroy:** A tárgy megsemmisítése.

3.3.7 ItemManager

- **Felelősség**

Ez egy interfész. Az ezt implementáló osztályok képesek tárgyakat tárolni, tehát rendelkeznek kettő művelettel: tárgy eltárolása és tárgy eltávolítása.
- **Őosztályok**

Item
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok:** nincs
- **Metódusok:**
 - void addItem(Item item):** Tárgy eltárolása.
 - void removeItem(Item item):** Tárgy eltávolítása.

3.3.8 Map

- **Felelősség**

Ő tudja a szobák elrendezését.
- **Őosztályok:** nincs
- **Interfészek:** nincs
- **Asszociációk**
 - Room:** Tárolja a Room-okat.
 - Door:** Ajtók a szobák közt.
- **Attribútumok:**
- **Metódusok:**
 - void addRoom(Room room):**
 - void removeRoom(Room room):**

3.3.9 Mask

- **Felelősség**

Ha egy hallgatót megtámad egy elgázósított szoba, akkor aktiválódik, és megvédi a hallgató életét.

- **Ősosztályok**
Item
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok:** nincs
- **Metódusok:** nincs

3.3.10 Person

- **Felelősség**
A személyek (hallgatók és oktatók) absztrakt ősosztálya. Tárolja a nevét, és felelős a személy mozgatásáért, tárgyakkal való interakciókért, valamint az állapotaiért is felel. Szól a szobának, hogy belépett.
- **Ősosztályok:** nincs
- **Interfészek:** nincs
- **Asszociációk**
Room: A Person tudja, hogy melyik szobában van.
Inventory: Minden személynek van saját inventory-ja.
- **Attribútumok**
string name: Személy neve.
bool isStunned: Megbénult állapot.
- **Metódusok:**
void receiveDamage: Amikor egy személyt megtámadnak ez a függvény meghívódik.
void drop: Eldobja a tárgyat a szobában, ahol van.
virtual void stun: Megbénítja a személyt. Mind a Student, mind a Teacher implementálja, különböző módokon.
void provideProtection: Ezt a függvényt hívja meg a védelmet nyújtó tárgy az askProtection hatására. Tulajdonképpen egy védelmi ajánlatként fogható fel.
virtual void slideRuleNotification: Ezt a metódust a SlideRule hívja ha a Person felvette őt. Ez implementációtól függően kezeli a SlideRule megszerzés eseményét: Teacher esetben instant eldobja azt, Student esetben pedig a játékot befejezi.

3.3.11 Room

- **Felelősség**
A játék szobáit reprezentálja. Számon tartja az adott szoba állapotát (elgázósított, elátkozott), a benne lévő tárgyakat és entitásokat, illetve képes osztódni vagy összeolvadni másik szobával.
- **Ősosztályok:** nincs
- **Interfészek:** ItemManager
- **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 isGased: El van-e gázosodva a szoba.

bool cursed: El van-e átkozva a szoba.

- **Metódusok**

void gas: A szoba elgázosodik.

void divide: A szoba osztódó függvénye.

void merge(Room other): A szoba összeolvadó függvénye.

void destroy(): A szoba megszűnését jelentő függvény.

void addPerson(Person person): A paraméterként kapott Person-t hozzáadja a szobához.

void removePerson(Person person): A paraméterként kapott Person-t kiveszi a szobából.

3.3.12 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:** nincs

- **Metódusok:** nincs

3.3.13 Student

- **Felelősség**

A hallgatót megvalósító osztály.

- **Ősosztályok**

Person

- **Interfészek:** nincs

- **Asszociációk:** nincs

- **Attribútumok**

Inventory inventory: A Student által felvett tárgyakat tároló Inventory.

- **Metódusok**

void stun(enum: StunType): Gáz hatására elejti a tárgyait (ha vannak) és megbénul adott időre.

3.3.14 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 stun(enum: StunType): Gáz vagy Cloth hatására megbénul adott időre.

3.3.15 Transistor

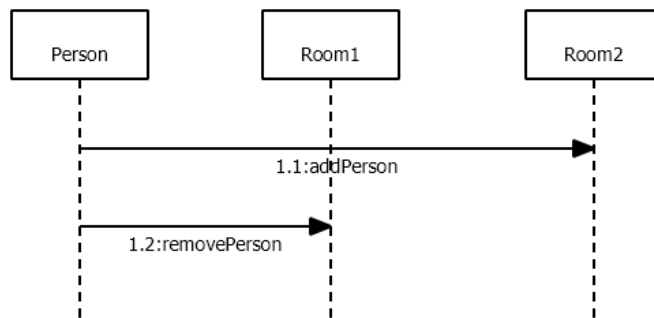
- **Felelősség**
A tranzisztort megvalósító osztály. Két tranzisztort össze lehet kötni, mindegyik számon tartja párját. Ha létrejött egy aktív pár, akkor elmozdítja a Studentet a másik tranzisztor szobájába. Deaktiválja magát és a párját ezután. Ha egy Teacher veszi fel, akkor reseteli a másik tranzisztort.
- **Ősosztályok**
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 tranzisztor párja.
- **Metódusok**
void deactivate: Ezt a függvényt hívja meg a tranzisztor magán és a párján, amikor a hallgató elteleportál.
void reset: A párja hívja meg ezt a függvényt, amikor elpusztul (oktató megsemmisíti). Ekkor a tranzisztor deaktiválódik, amíg újra fel nem veszi egy hallgató és össze nem köti egy új tranzisztorral.

3.3.16 TVSZ

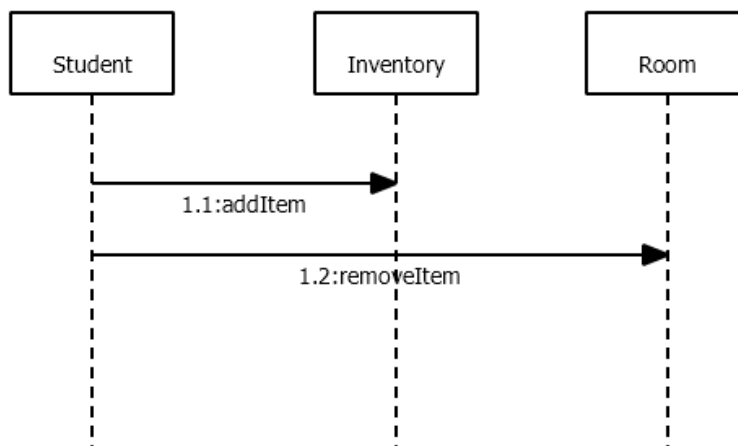
- **Felelősség**
Ha az öt birtokló hallgatót megtámadja egy oktató és nincs magasabb prioritású védelme (Beer), akkor aktiválódik, ezzel megvédve a hallgatót. Számontartja, hogy hány alkalommal képes még megvédeni a hallgatót, mielőtt elhasználná.
- **Ősosztályok**
Item
- **Interfészek:** nincs
- **Asszociációk:** nincs
- **Attribútumok:** nincs
- **Metódusok:** nincs

3.4 Szekvencia diagramok

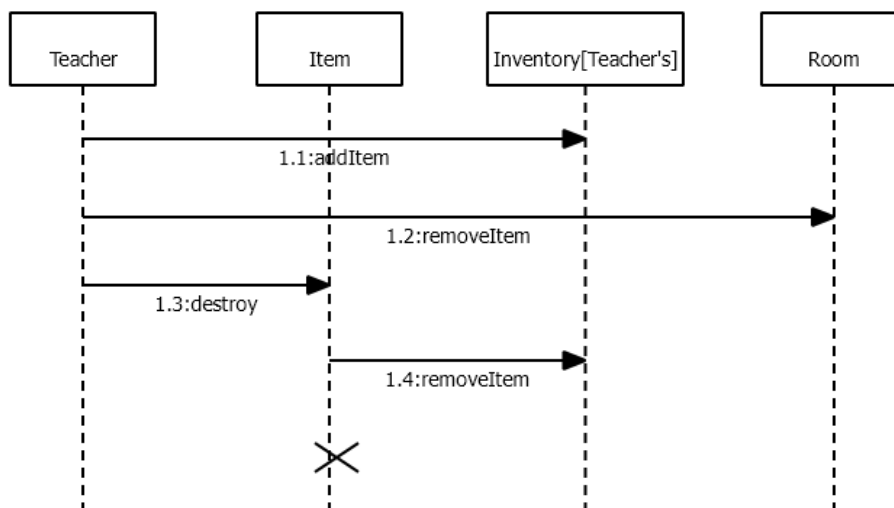
Person szobát vált:



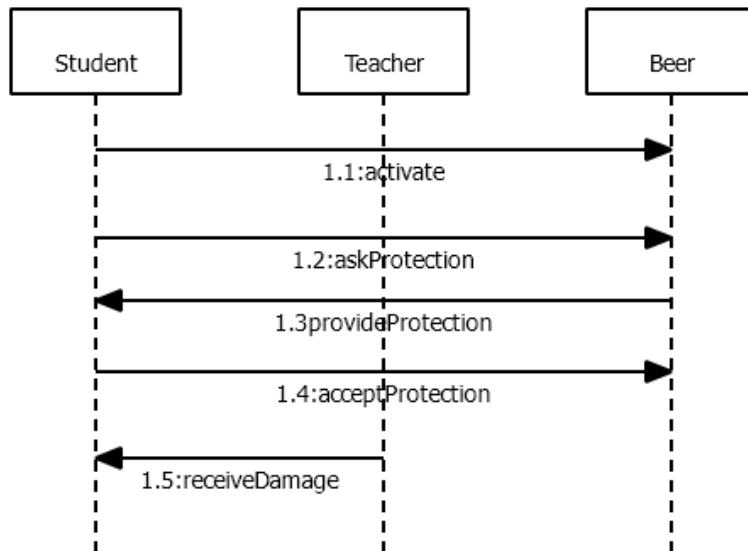
Tárgy felvétele hallgató által:



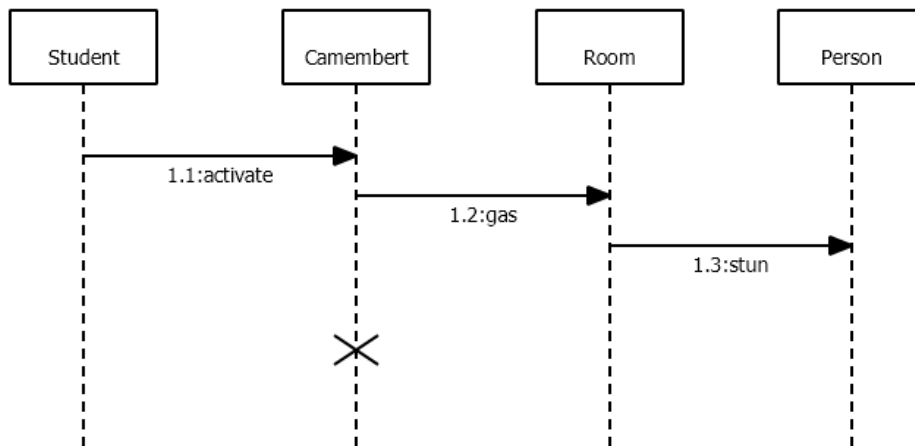
Tárgy felvétele oktató által:



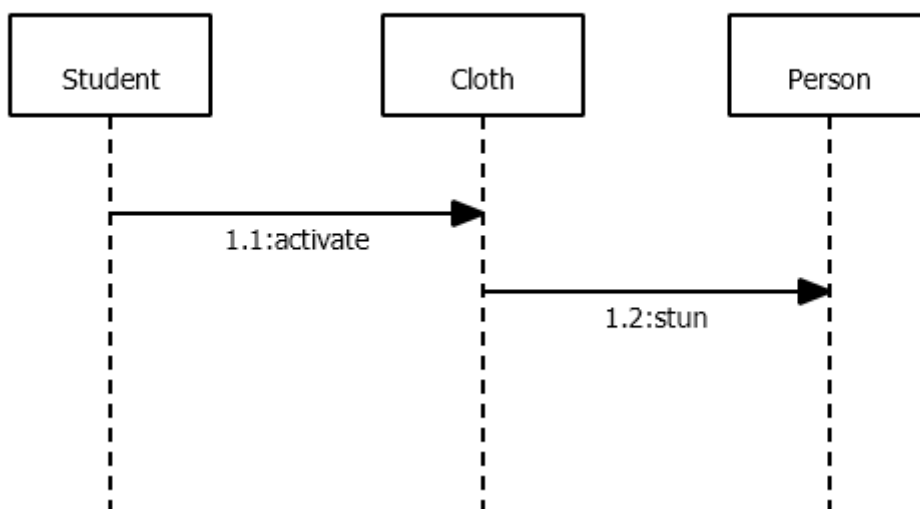
Beer használata:



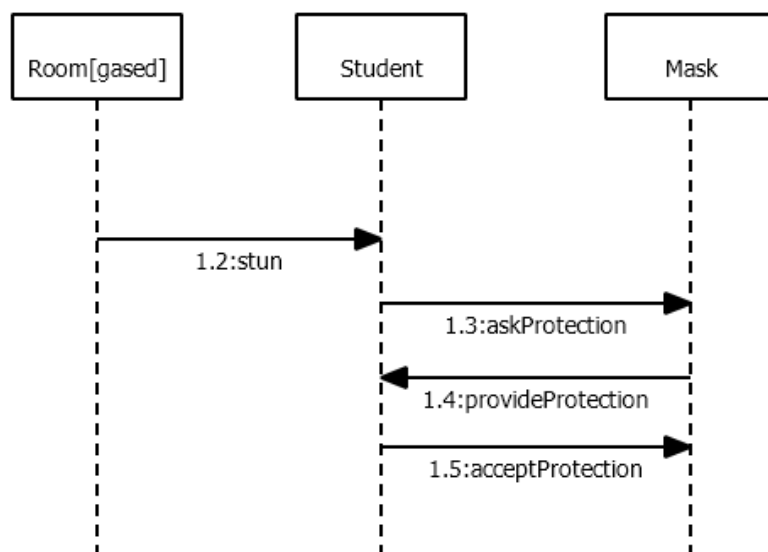
Camambert használata:



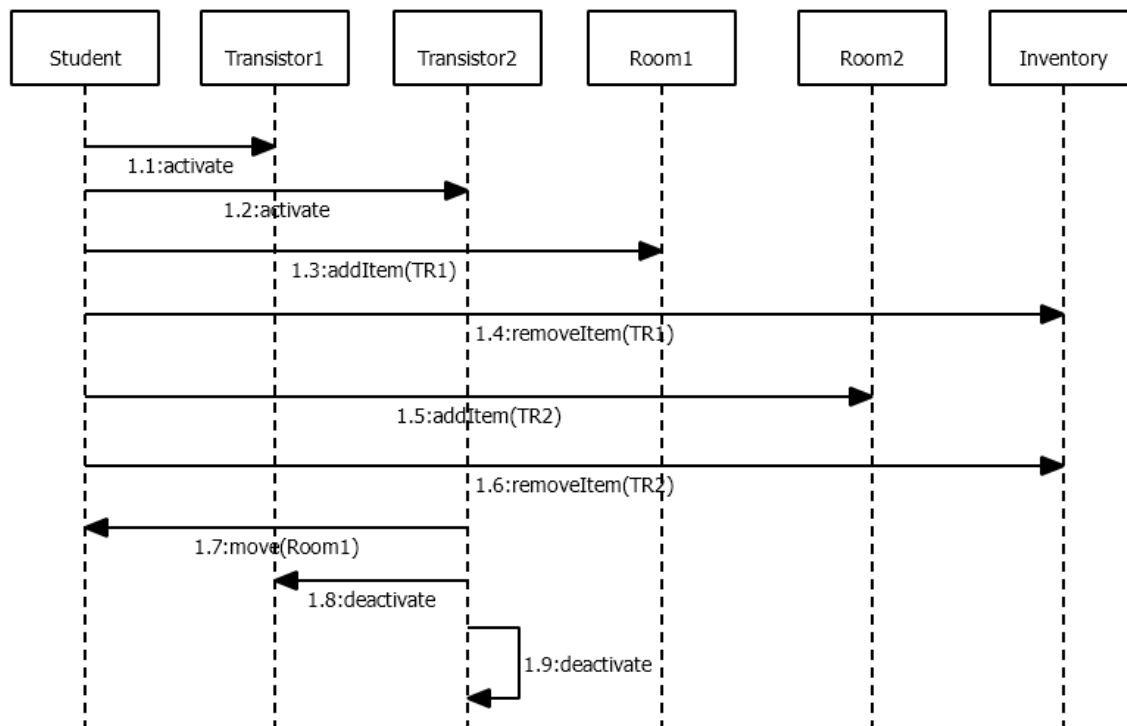
Cloth használata:(Student esetén nem történik semmi, Teacher lebénul)



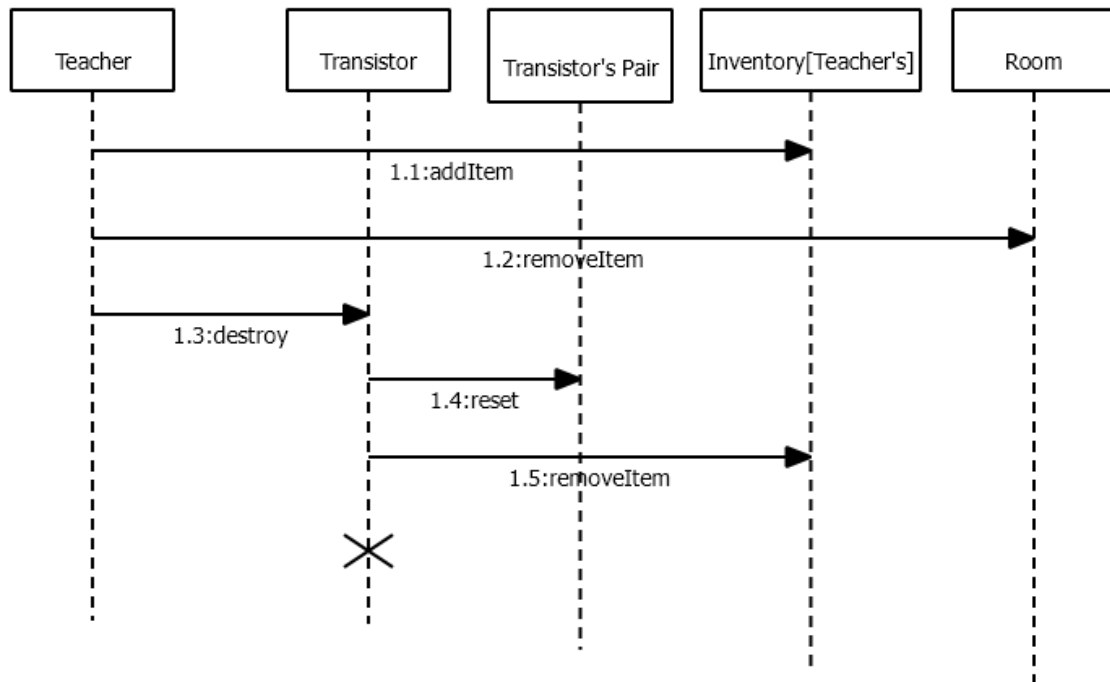
Mask használata:



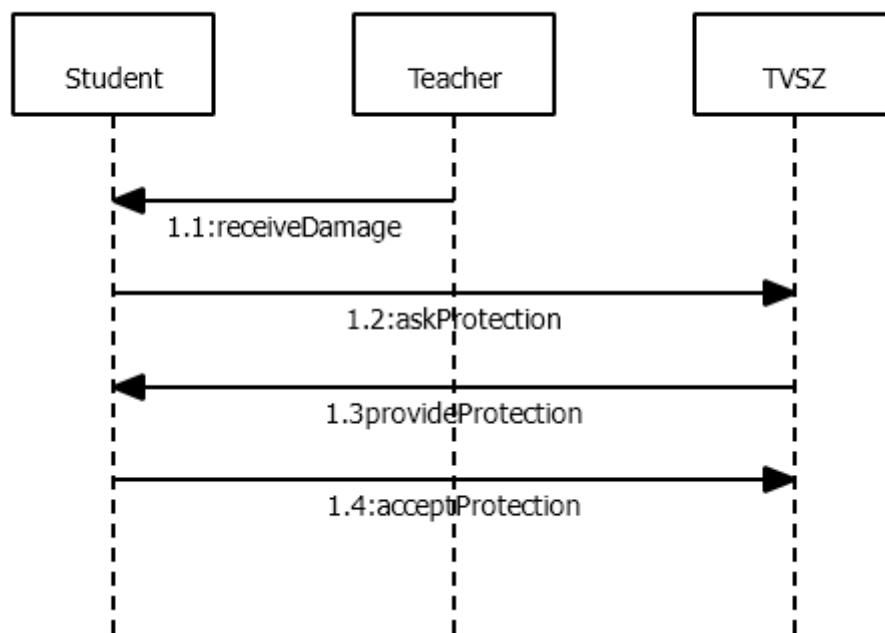
Transistor használata:



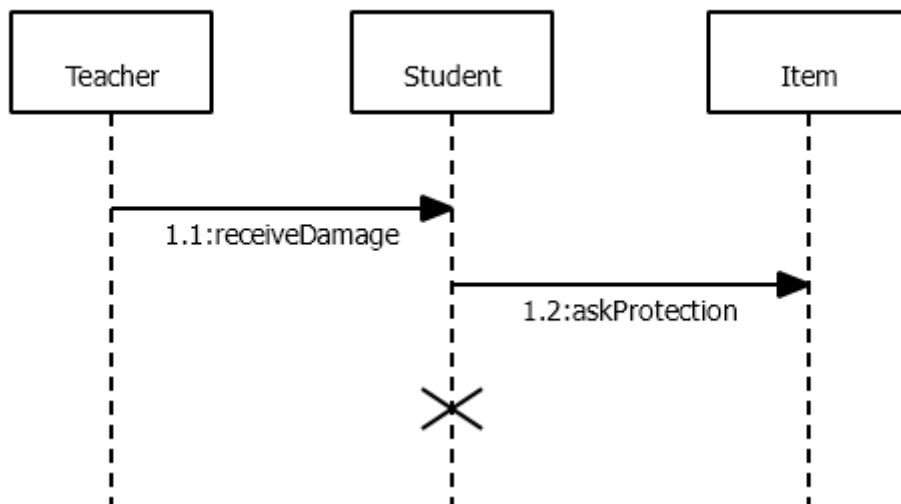
Oktató felvesz egy már használt tranzisztort:



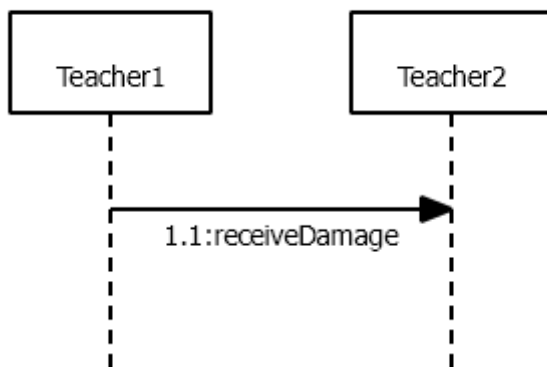
TVSZ használata:



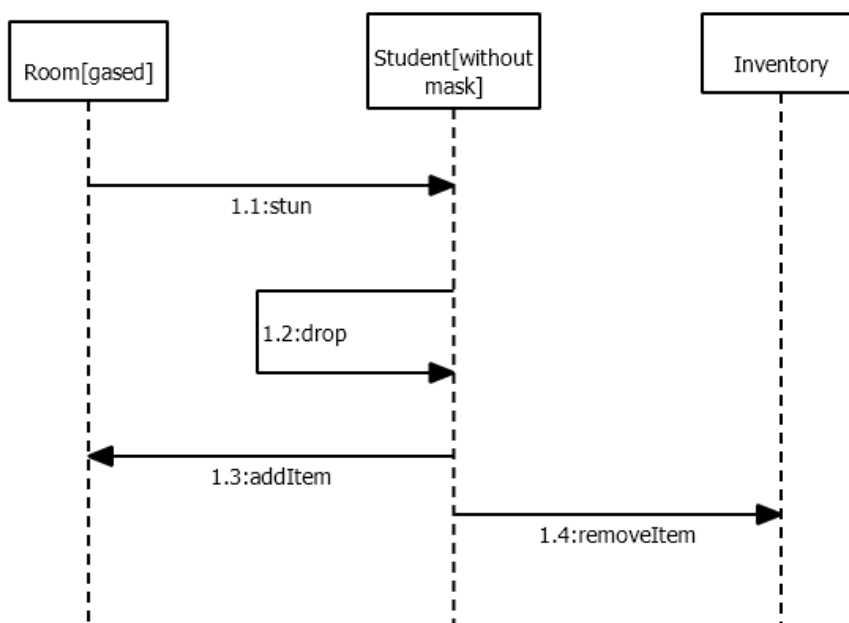
Oktató megtámadja a hallgatót, akinek nincs védelme:



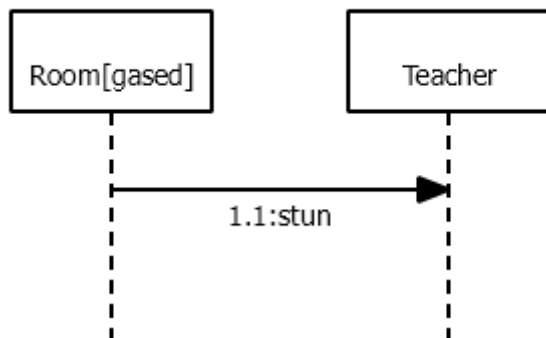
Oktató megtámadja oktatót:(mivel egy szoba összes Person-ét megtámadja, ezért ez is előfordulhat)



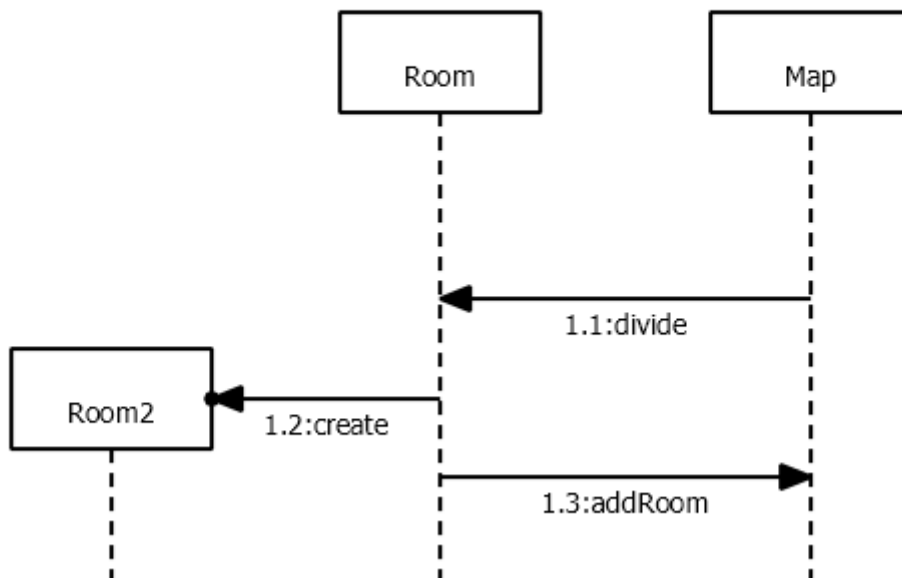
Gázos szoba hatása maszk nélküli hallgatóra:



Gázos szoba hatása oktatóra:

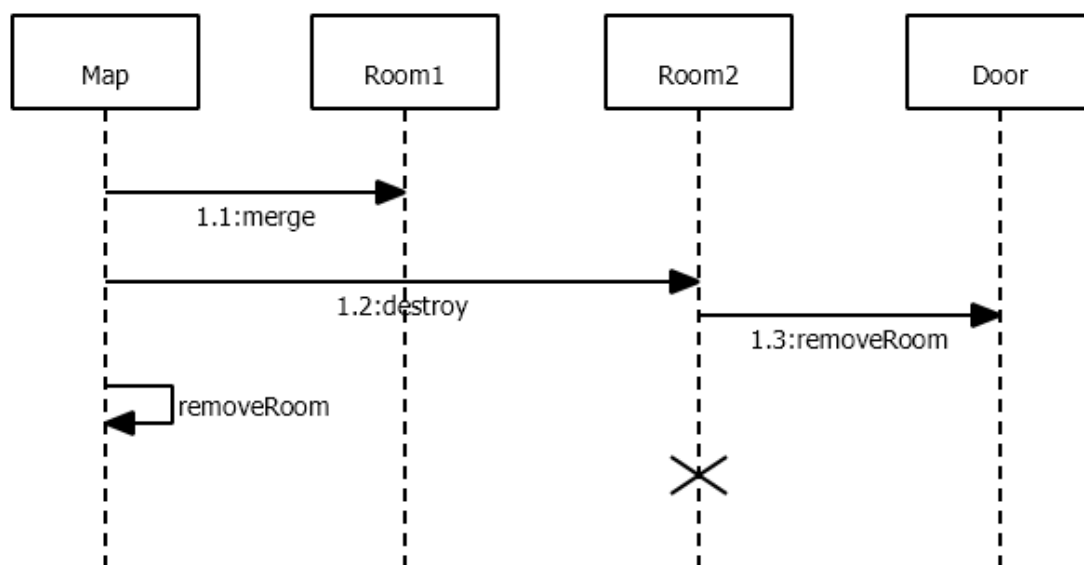


Szoba osztódás:

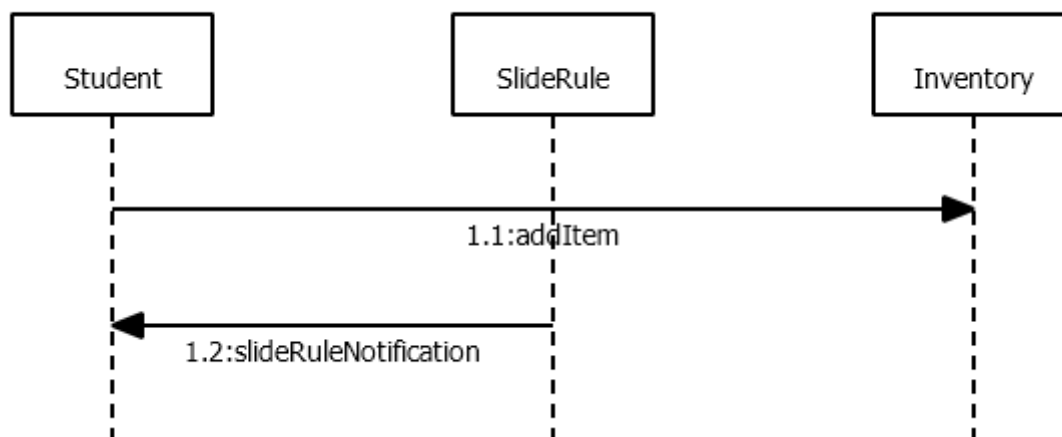


create: konstruktorhívás

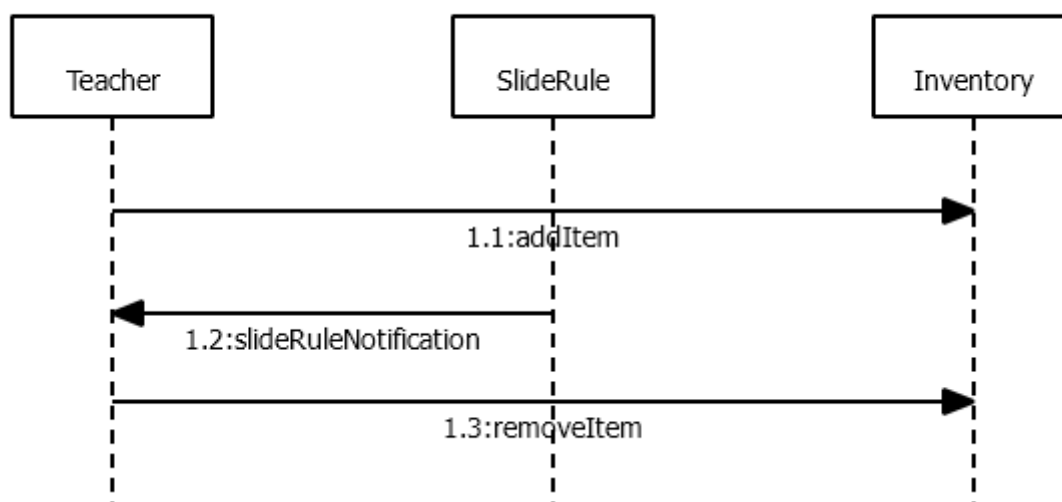
Szoba összeolvadás:



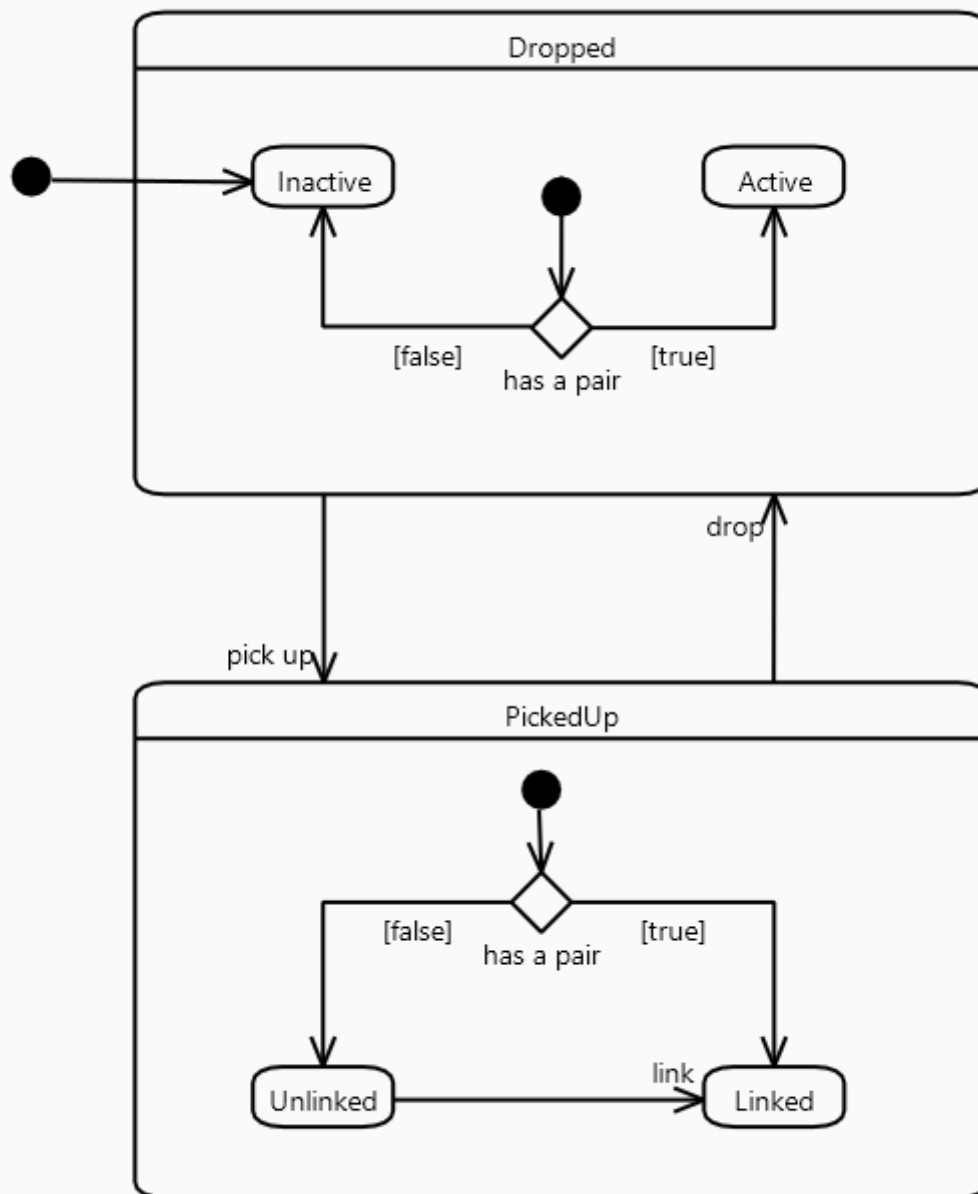
Logarléc felvétele hallgató által:



Logarléc felvétele oktató által:



3.5 State-chartok



3.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.02.27. 20:00	1 óra	Cardinael Görömbey Király Szakos	Értekezlet. Feladat áttekintése a csapattal. A program kódbeli struktúrájának felvázolása. Feladatok kiosztása.
2024.03.01. 12:00	3 óra	Cardinael Görömbey Riba Szakos	Értekezlet. Az osztályok összeírása és a felelőségek meghatározása. Szélesebb körű tervezési döntések egyeztetése.
2024.03.02. 12:00	2 óra	Cardinael Görömbey	Objektumkatalógus megcsinálása, osztályok leírásának bővítése. Szekvenciadiagram ok elkezdése.
2024.03.03. 14:30	4 óra	Cardinael Görömbey Király Riba Szakos	Osztályok működésének és kapcsolatainak átbeszélése és fixálása, diagramok megcsinálásának kiosztása.
2024.03.03 20:00	1.5 óra	Király	3.3 fejezet lektorálása, javítások ajánlása.
2024.03.03. 19:00	2 óra	Cardinael Görömbey	Szekvenciadiagram k.
2024.03.03. 22:00	3 óra	Cardinael Görömbey Király Szakos	Végső simítások.
2024.03.04. 10:00	1 óra	Cardinael	Konzisztenciai javítások, nyomtatás