

5. Szkeleton tervezése

70 – Edgerunners

Konzulens:

Szabó Bence Sándor

Csapattagok

Csabuda Nóra
Gulybán Dániel
Nagy Gergely
Papp Levente
Sánta Dániel

GHDCTP
HUCHEX
HC1QJP
HV0CRH
OGLSI2

csabudan@gmail.com
gulyban.daniel@gmail.com
nagyg2002@gmail.com
papp.levente2003@gmail.com
daniel.santa0615@gmail.com

2024-03-18

5. Szkeleton tervezése

5.0 Javítások az analízis modellhez

Az előző laboralkalom végén megbeszélték szerint nincs szükség az *ITickable* interfészre, mert nem fogjuk felhasználni heterogén kollekcióra, csoportosításra. A *Game* osztály objektuma fog entitásokat és szobákat tárolni, amelyeknek megvan a megfelelő *tick* metódusuk.

Az *Entity*, *Student* és *Prof* osztályok *meetStudent(s: Student)* és *meetProf(p: Prof)* fejlécű metódusai átnevezésre kerültek *meet(s: Student)* és *meet(p: Prof)* fejlécű metódusokra, viselkedésükben nem történt változás.

A *Door* osztály *canPassThrough(r: Room): bool* metódusa törlésre került és kapott egy új, *letEntityThrough(e: Entity, dest: Room): bool* metódust. Az ajtó, a paraméterben kapott szobán meghívja az *acceptEntity* metódust, ha oda vezet. Ebben az esetben az *acceptEntity* visszatérési értékével tér vissza, különben hamissal. A *letEntityThrough* metódust az *Entity* osztály *move* metódusában fogjuk felhasználni. Ezenfelül a *Door* osztályba került egy új két paraméteres publikus metódus a *swapRoom(oldRoom: Room, newRoom: Room)*. Első paramétere ezen metódusnak azt adja meg, hogy az ajtó melyik tárolt szobáját állítsa át a második paraméterben kapott szobára.

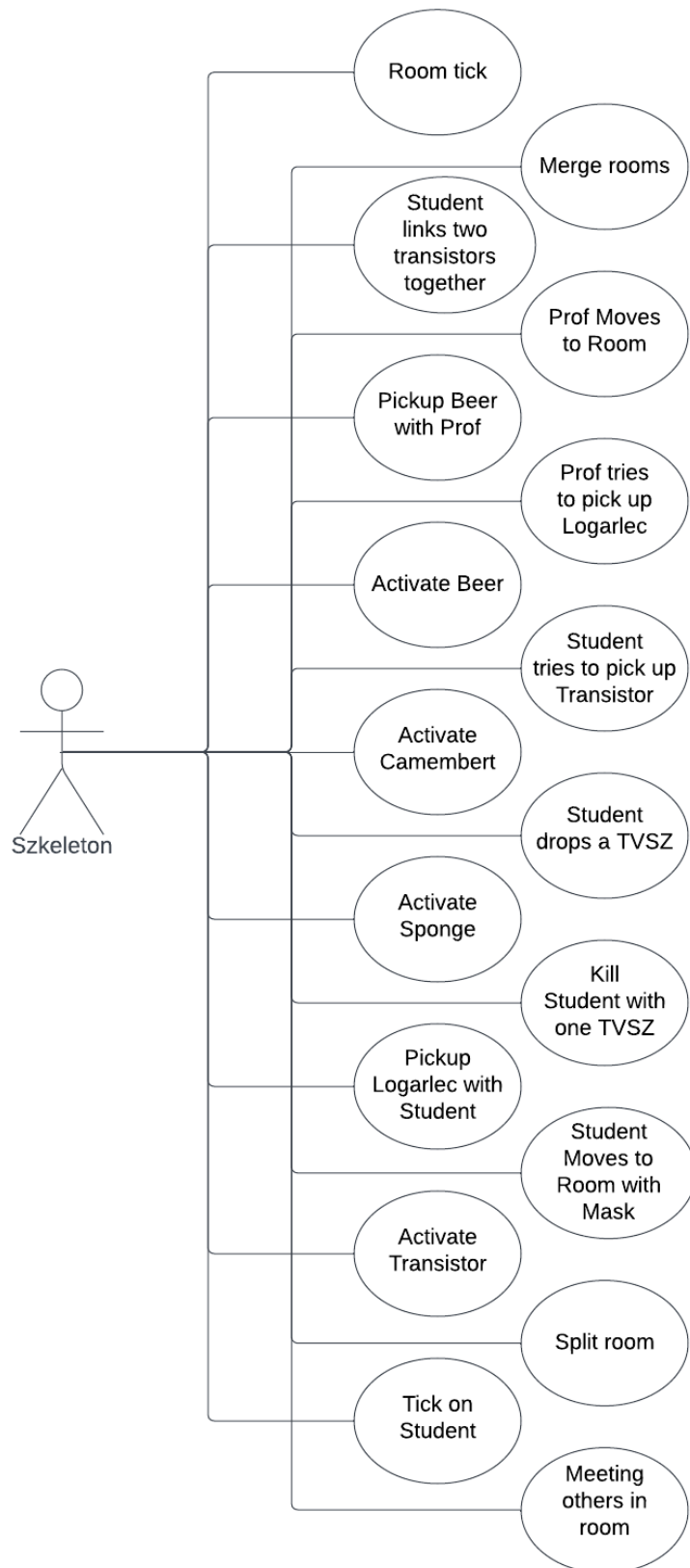
Az *acceptEntity* metódusból átkerült a *toxicate* és *immobilize* metódusokba a mérgezés, tárgyak elejtése és entitás bénításának logikája.

A *Room* osztályba felvételre került három új metódus. Ezek rendre a *makeRoomToxic()*, *makeRoomWet()* és *makeRoomCursed()*. A hívott szoba átállítja a megfelelő flageket. Hasonló elgondolásból a *Student* osztályba is felvételre került egy *makeStudentDrunk()* metódus, amelyben a hívott hallgató beállítja magában a megfelelő értéket.

A *Transistor*ba felvételre került még egy *teleport(s: Student): bool* metódus is, amely a tranzisztorba regisztrált szobába próbálja meg átküldeni az *s Student*-et az *acceptEntity* hívásával. A művelet sikerességét a visszatérési értékében jelzi.

5.1 A szkeleton modell valóságos use-case-ei

5.1.1 Use-case diagram



5.1.2 Use-case leírások

Use-case neve	Pickup Beer with Prof
Rövid leírás	A Szkeleton egy sör felvételét hajtja végre egy professzorral.
Aktorok	Skeleton
Forgatókönyv	Az éppen következő professzor megpróbál felvenni egy sört, ha van hely neki, akkor sikeresen felveszi és eltávolítja az a szobából.

Use-case neve	Activate Beer
Rövid leírás	A Szkeleton aktivál egy sört egy hallgatóval..
Aktorok	Skeleton
Forgatókönyv	Ha a sört még nem használták, akkor elhasználódik és a hallgató sörös hatás alá kerül.

Use-case neve	Activate Camembert
Rövid leírás	A Szkeleton aktivál egy camembertes dobozt egy hallgatóval.
Aktorok	Skeleton
Forgatókönyv	Ha a dobozt még nem használták, akkor a hallgató szobája gázos lesz.

Use-case neve	Activate Sponge
Rövid leírás	A Szkeleton aktivál egy szivacsot egy hallgatóval.
Aktorok	Skeleton
Forgatókönyv	Ha a szivacsot még nem használták, akkor a hallgató szobája vizes lesz.

Use-case neve	Student picks up Logarlec
Rövid leírás	A Szkeleton felvetet egy logarlécet egy hallgatóval.
Aktorok	Skeleton
Forgatókönyv	A hallgató, ha van nála hely, felveszi a logarlécet a szobájából és közben megnyeri a játékot a hallgatók javára.

Use-case neve	Prof tries to pick up Logarlec
Rövid leírás	A Szkeleton megpróbál felvetetni egy logarlécet egy oktatóval.
Aktorok	Skeleton
Forgatókönyv	Az oktató megpróbálja felvenni a logarlécet a szobából, azonban ez nem sikerül neki és nem tudja felvenni.

Use-case neve	Student tries to pick up Transistor
Rövid leírás	A Szkeleton egy hallgatóval próbál felvetetni egy tranzisztort.
Aktorok	Skeleton
Forgatókönyv	A hallgató, ha van még nála hely, akkor megpróbál felvenni a szobájából egy tranzisztort, ha a tranzisztor aktív, akkor sikertelenül jár, ha nem aktív, akkor pedig fel is veszi azt.

Use-case neve	Student drops a TVSZ
Rövid leírás	A Szkeleton eldobat egy tvsz-t a hallgatóval.
Aktorok	Szkeleton
Forgatókönyv	A hallgató eldobja a nála lévő tvsz-t és az a szobába kerül.

Use-case neve	Kill Student with one TVSZ
Rövid leírás	A Szkeleton megpróbálja megölni a hallgatót, akinél van egy tvsz.
Aktorok	Szkeleton
Forgatókönyv	A hallgató meghal, ha a hallgató nem sörös, vagy ha a nála lévő tvsz-nek nincs élete. Ha a tvsz-nek van élete, akkor eggyel csökkenti azt.

Use-case neve	Student links two transistors together
Rövid leírás	A Szkeleton egy hallgatóval kettő tranzisztort összekapcsol.
Aktorok	Szkeleton
Forgatókönyv	A hallgató összekapcsolja a nála lévő két tranzisztort, ezzel együtt beállítja mindkettő párját a másikkra.

Use-case neve	Activate Transistor
Rövid leírás	A Szkeleton aktivál egy tranzisztort a hallgatóval.
Aktorok	Szkeleton
Forgatókönyv	A hallgató aktiválja a nála lévő tranzisztort, ha van párja és nem aktív, akkor aktiválódik és eldobjuk azt a jelenlegi szobába. Ha van párja és aktív, akkor a hallgató teleportál a pár szobájába, ha befér abba és mindkét tranzisztor újra inaktív lesz.

Use-case neve	Prof Moves to Room
Rövid leírás	A Szkeleton egy újabb szobába mozgat egy professzort.
Aktorok	Szkeleton
Forgatókönyv	A professzor átmegy egy másik szobába, ha az nincs tele. Ha a szoba gázos, akkor a professzor nem tud belépni, elkábul egy körre és eldobja a tárgyait. Ha a szoba vizes, akkor nem tud belépni és szintén elkábul egy körre. Ha egyik sem, akkor sikeresen be tud lépni a szobába.

Use-case neve	Student Moves to Room with Mask
Rövid leírás	A Szkeleton egy maszkkal rendelkező hallgatót léptet egy szobába.
Aktorok	Szkeleton
Forgatókönyv	A hallgató sikeresen belép a szobába, ha abba még van hely. Ha a szoba gázos és a nála lévő maszknak már nincs élete, elkábul egy körre és eldobja a nála lévő tárgyakat és . Ha a maszknak van élete, akkor ilyenkor csökken egyet és következmény nélkül bekerül a hallgató.

Use-case neve	Merge rooms
Rövid leírás	A Szkeleton két entitás mentes szobát egyesít.
Aktorok	Szkeleton
Forgatókönyv	Az egyesítésből keletkező szoba tulajdonságai megegyezik a kezdeti két szoba tulajdonságainak uniójával. Tárgyaira és aajtajaira ugyanez elmondható. A kapacitás a nagyobb kapacitású szoba kapacitásával lesz egyenlő.

Use-case neve	Split room
Rövid leírás	A Szkeleton egy entitás mentes szobát ketté oszt.
Aktorok	Szkeleton
Forgatókönyv	A keletkező új szoba véletlenszerűen megkapja az eredeti szoba aajtajai, tárgyai egy részét. A keletkező szoba kapacitása megegyezik az előző kapacitásával.

Use-case neve	Tick on Student
Rövid leírás	A Szkeleton tick-el egy hallgatót.
Aktorok	Szkeleton
Forgatókönyv	A hallgató a sörös és kábult effektusból idejéből levon egyet, ha azoknak az értéke nem 0.

Use-case neve	Room tick
Rövid leírás	A Szkeleton tick-el egy szobát, ahol van egy professzor és két hallgató.
Aktorok	Szkeleton
Forgatókönyv	A szoba, ha gázos, megpróbálja elkábítani az összes benne lévő entitást, ha azoknál van tárgy, akkor meg próbálnak védekezni ellene. Akik nem tudják kivédeni a támadást, azok egy körre elkábulnak és eldobják a tárgyaikat. Ha a szoba vizes, akkor a professzort elkábítja egy körre. Ha a szoba elátkozott, akkor véletlen eséllyel elvarázsolja az aajtajait.

Use-case neve	Meet others in room
Rövid leírás	A Szkeleton egy tvsz-szel rendelkező hallgatót mozgat egy olyan szobába, ahol van egy professzor és egy másik hallgató.
Aktorok	Szkeleton
Forgatókönyv	A hallgató belép a szobába, ha ott van hely. Ha gázos a szoba, akkor belépéskor megtámadja a hallgatót, ha nem tud védekezni, akkor elkábul és eldobja a nála lévő tárgyakat. Ezután a szobában lévő professzor megtámadja a belépő hallgatót. Ha a hallgató sörös effekt alatt áll, akkor kivédi a támadást, ha nem, akkor pedig megpróbálja felhasználni a nála lévő tárgyakat. Ha nem sikerül így sem védekezni, akkor a hallgató meghal, ha sikeresen védekezett, akkor bent marad a szobában.

5.2 A szkeleton kezelői felületének terve, dialógusok

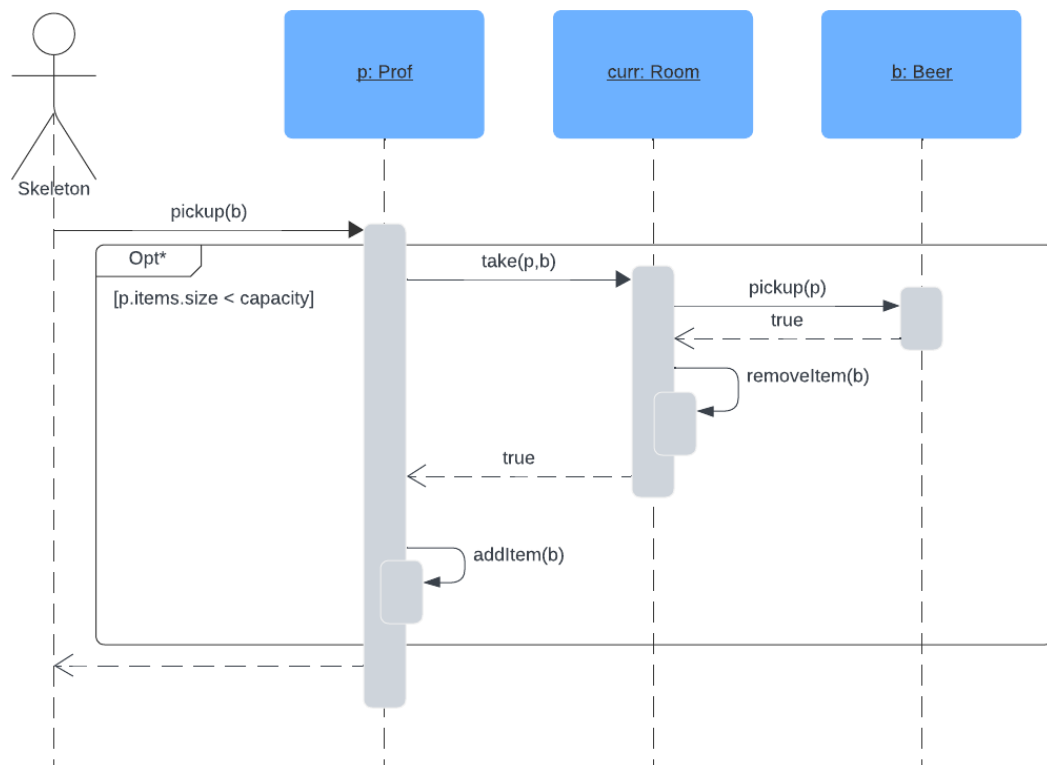
A szkeleton egy menüvezérelt parancssori alkalmazás lesz. Az egyes menüpontok kiválasztásával lehet majd a különböző test-case-eket lejátszatni. A menüpontok sorszámozva lesznek, számuk megadásával lehet kiválasztani őket. A szkeletonprogram nem tárolja még az objektumok állapotát, ezért, ha egy objektum a belső állapota alapján eltérően viselkedne, akkor a program egy eldöntendő kérdés formájában megkérdezi a felhasználót az objektum állapotáról. Az eldöntendő kérdésekre az i (igen) és n (nem) karakterek megadásával lehet válaszolni. Az egyes üzenetek (metódushívások), azok válaszaik (visszatérési érték) és kérdések új sorban, indentálva fognak megjelenni az alábbi példa szerint.

Példa:

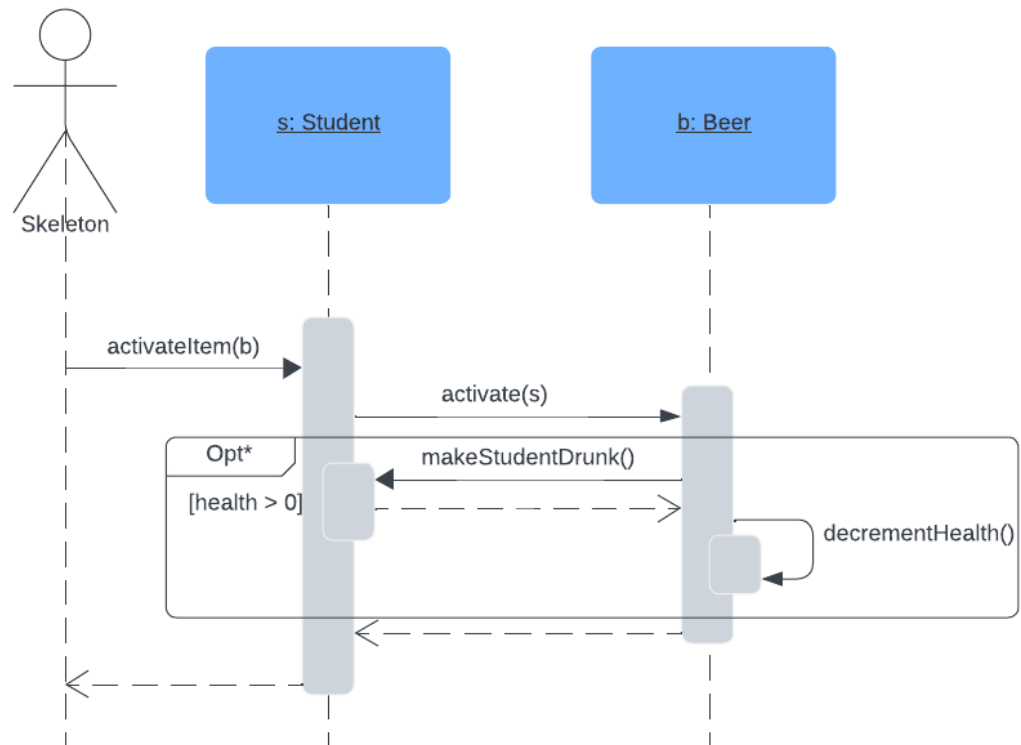
```
Entity.pickup(item: Item) called
  Van elég hely az entitásnál? i
  Room.take(entity: Entity, item: Item) called
    Item.pickup(e: Entity) called
    Item.pickup(e: Entity) returned true
  Room.take(entity: Entity, item: Item) returned true
Entity.pickup(item: Item) returned
```

5.3 Szekvencia diagramok a belső működésre

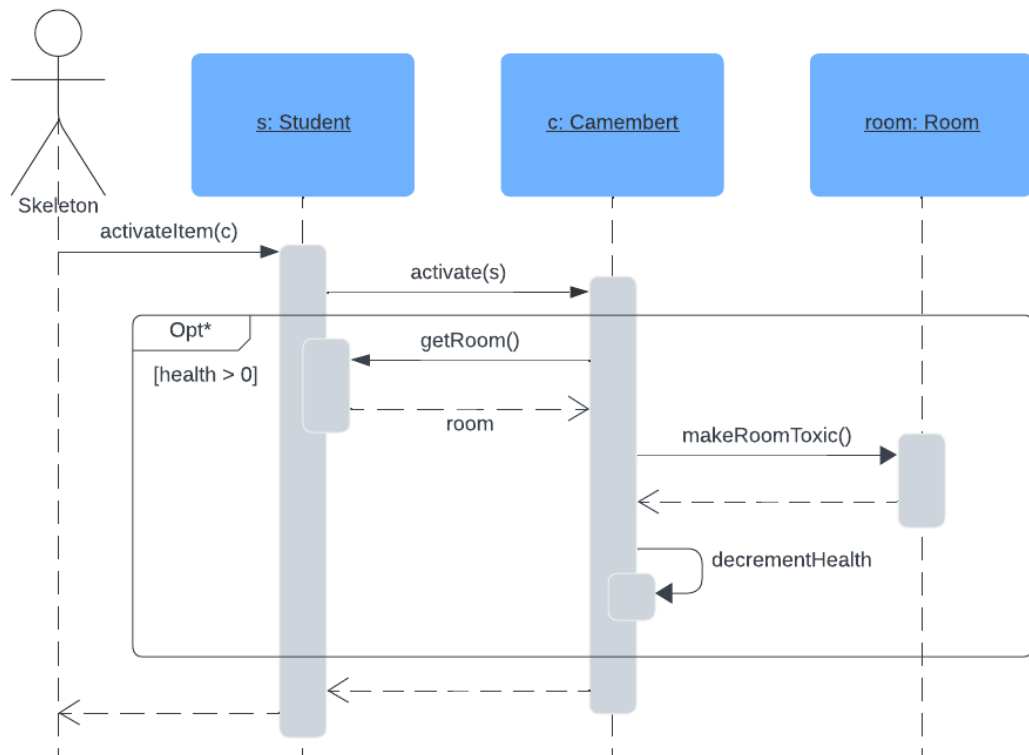
5.3.1 Professor picks up a Beer



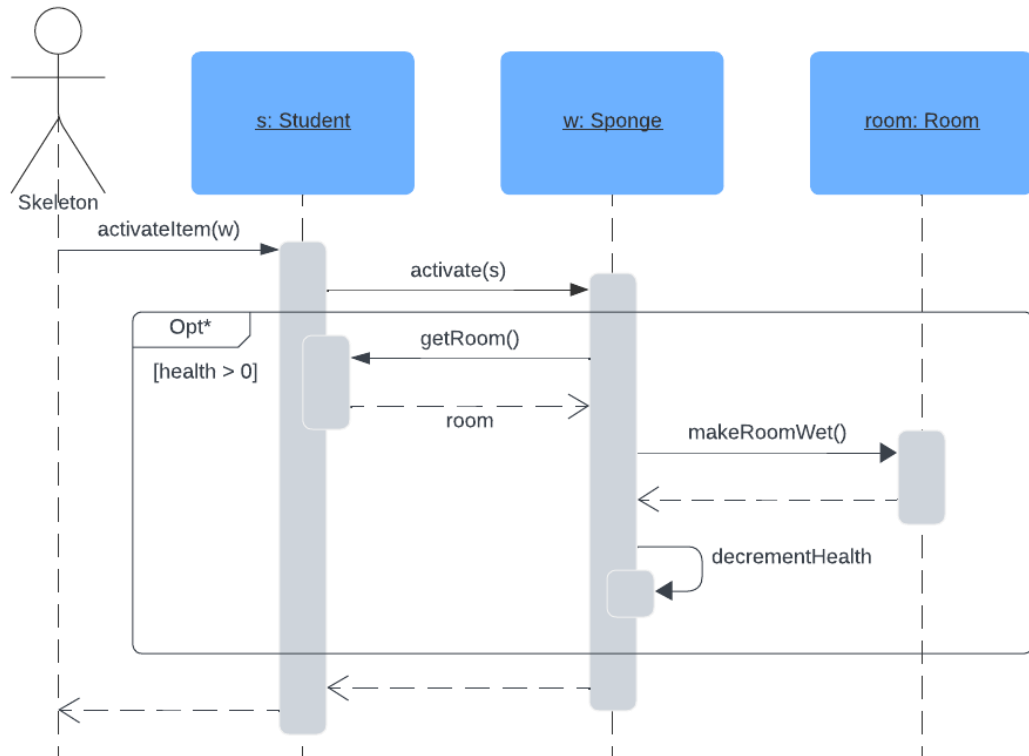
5.3.2 Student activates a Beer



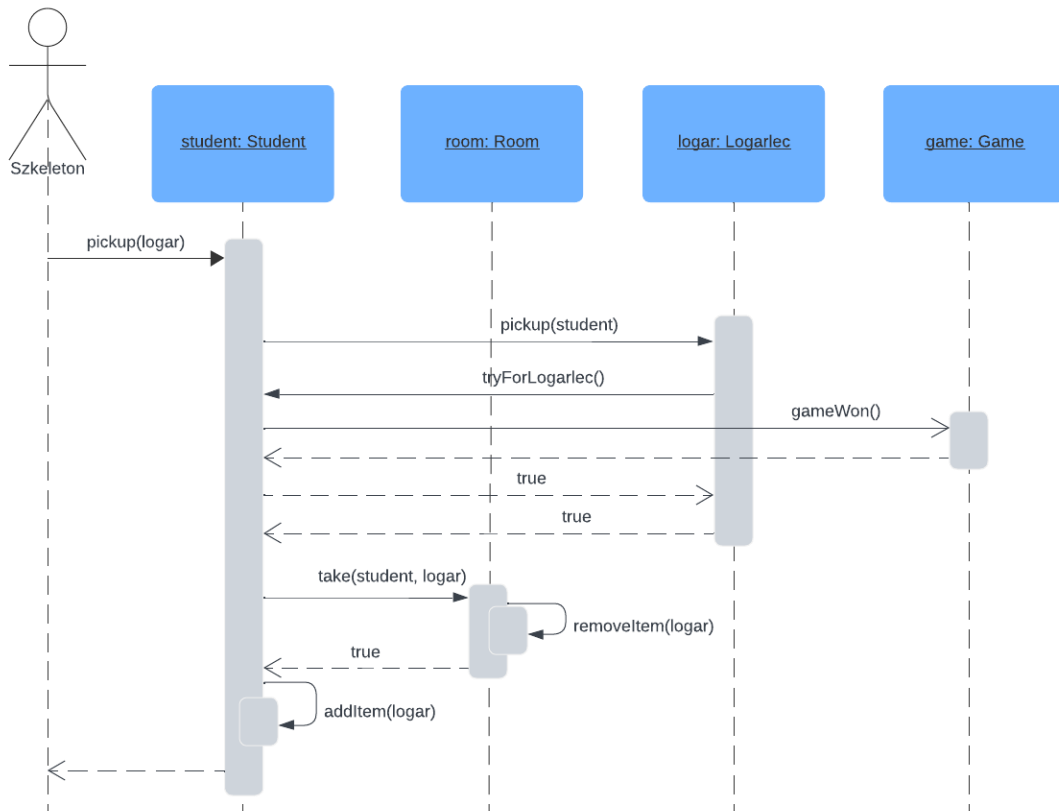
5.3.3 Student activates a Camembert



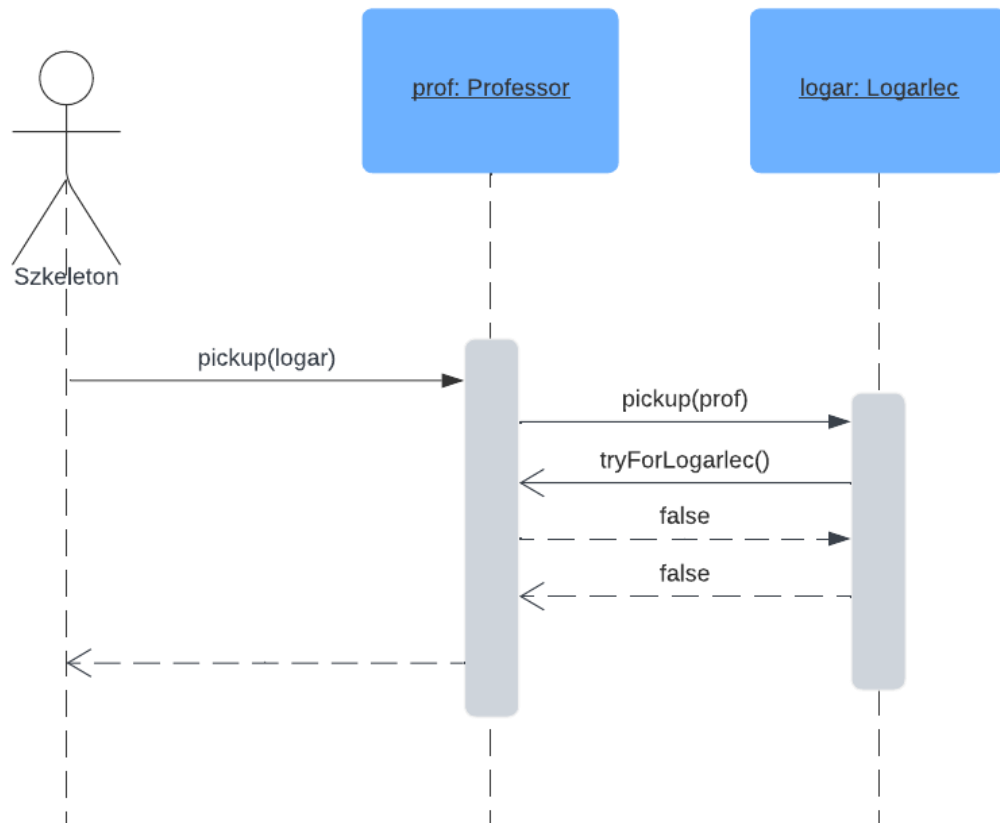
5.3.4 Student activates a WetSponge



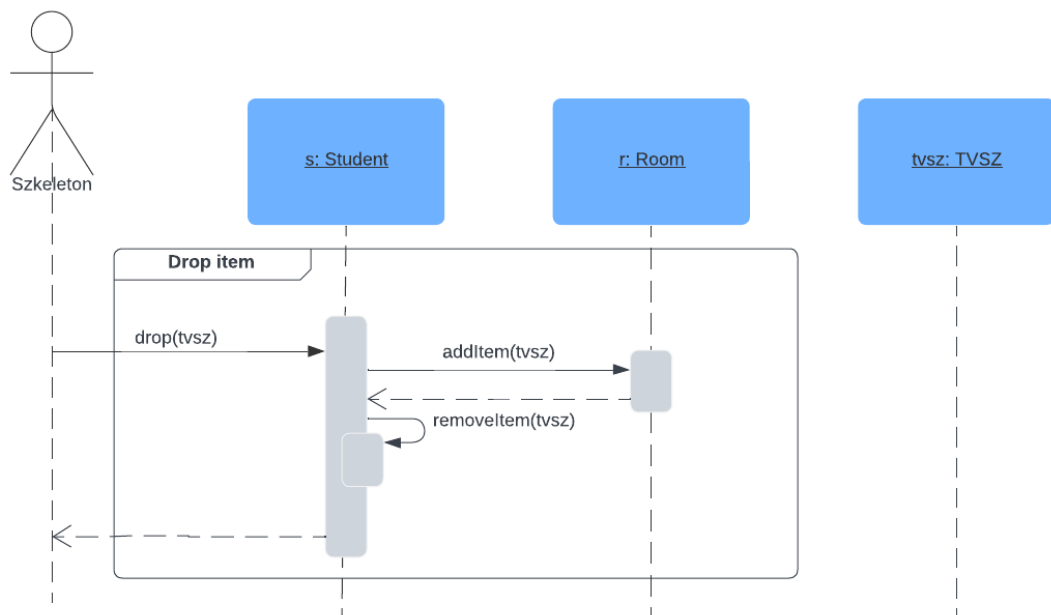
5.3.5 Student picks up Logarlec



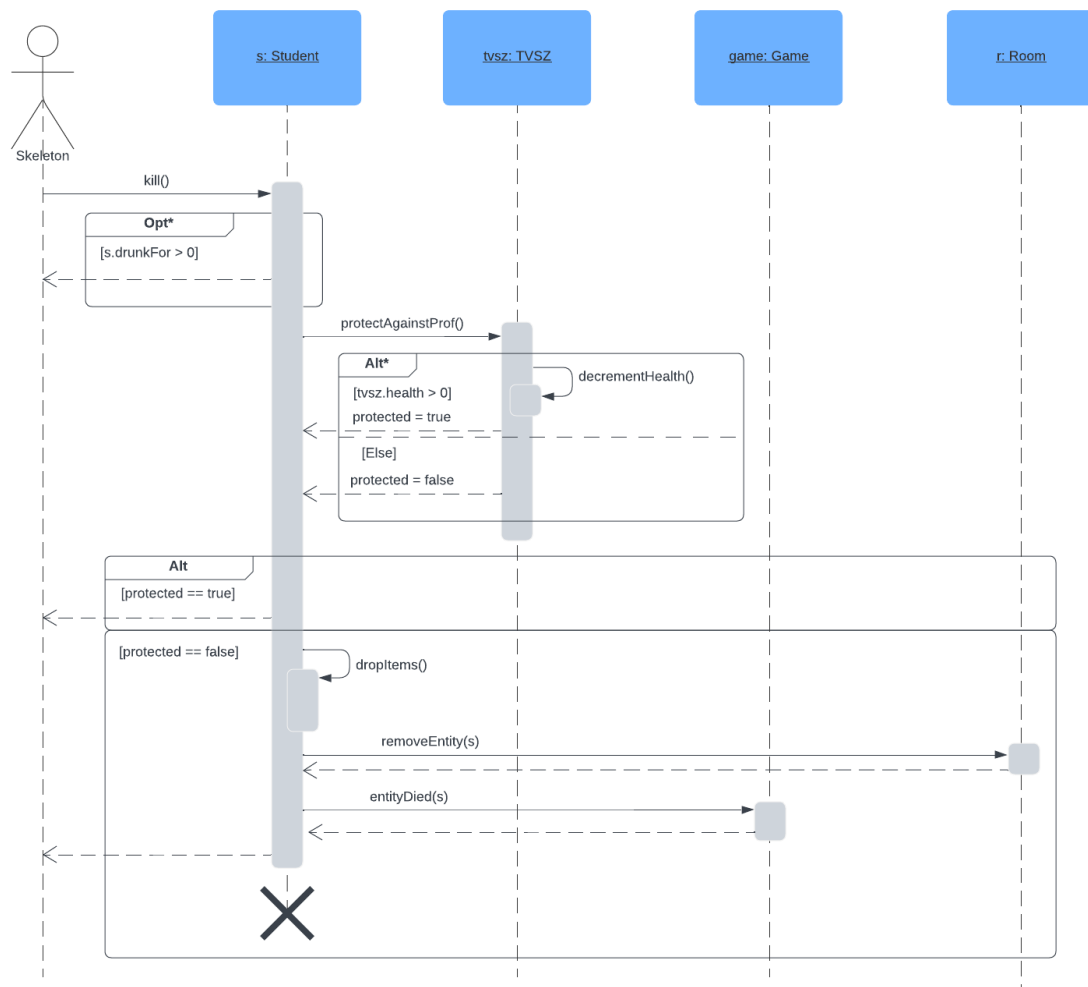
5.3.6 Professor tries to pick up Logarlec



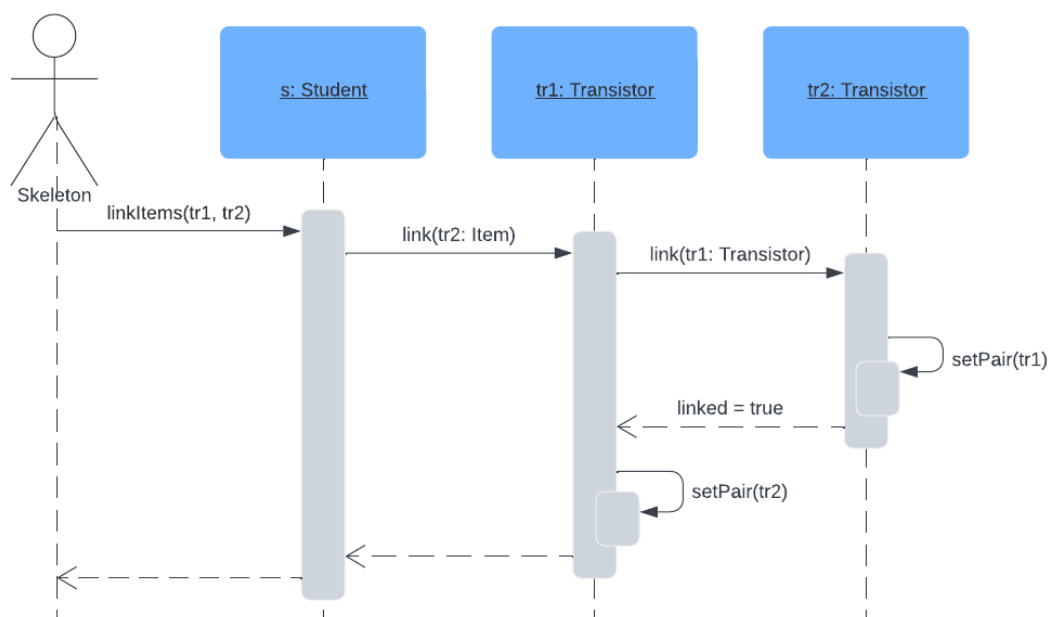
5.3.7 Student drops a TVSZ



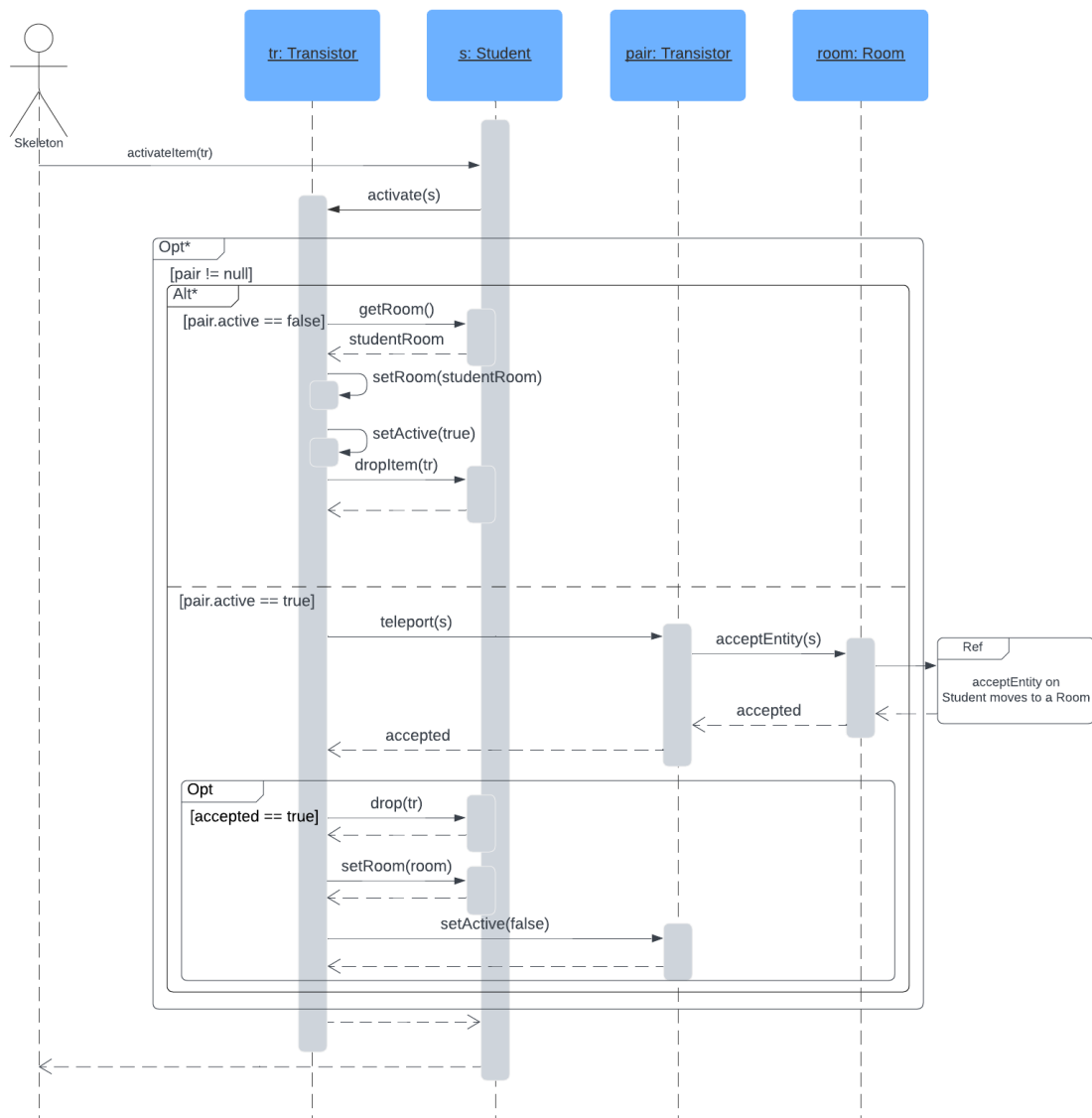
5.3.8 Student gets attacked



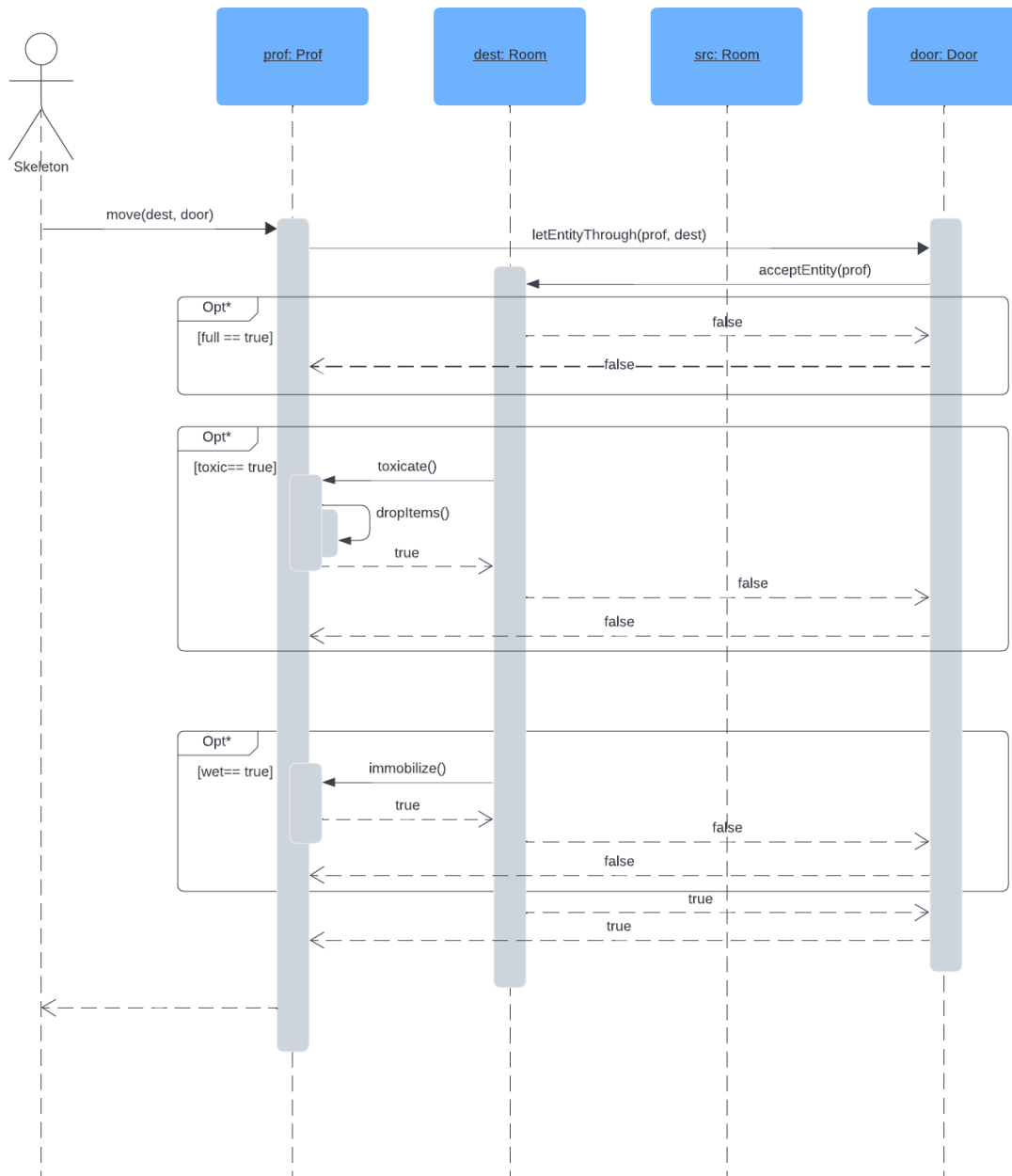
5.3.9 Student links two Transistors together



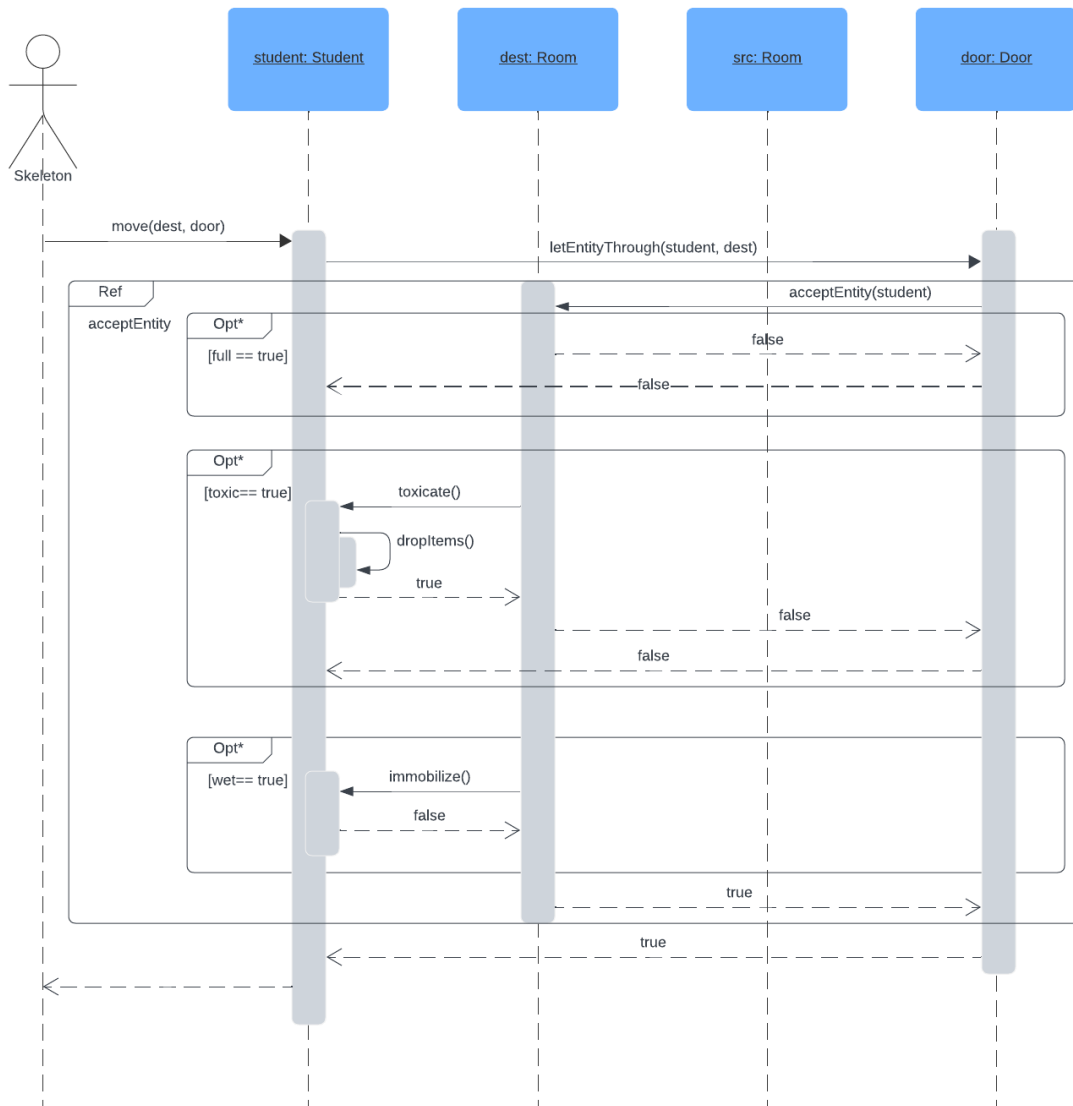
5.3.10 Student activates a Transistor



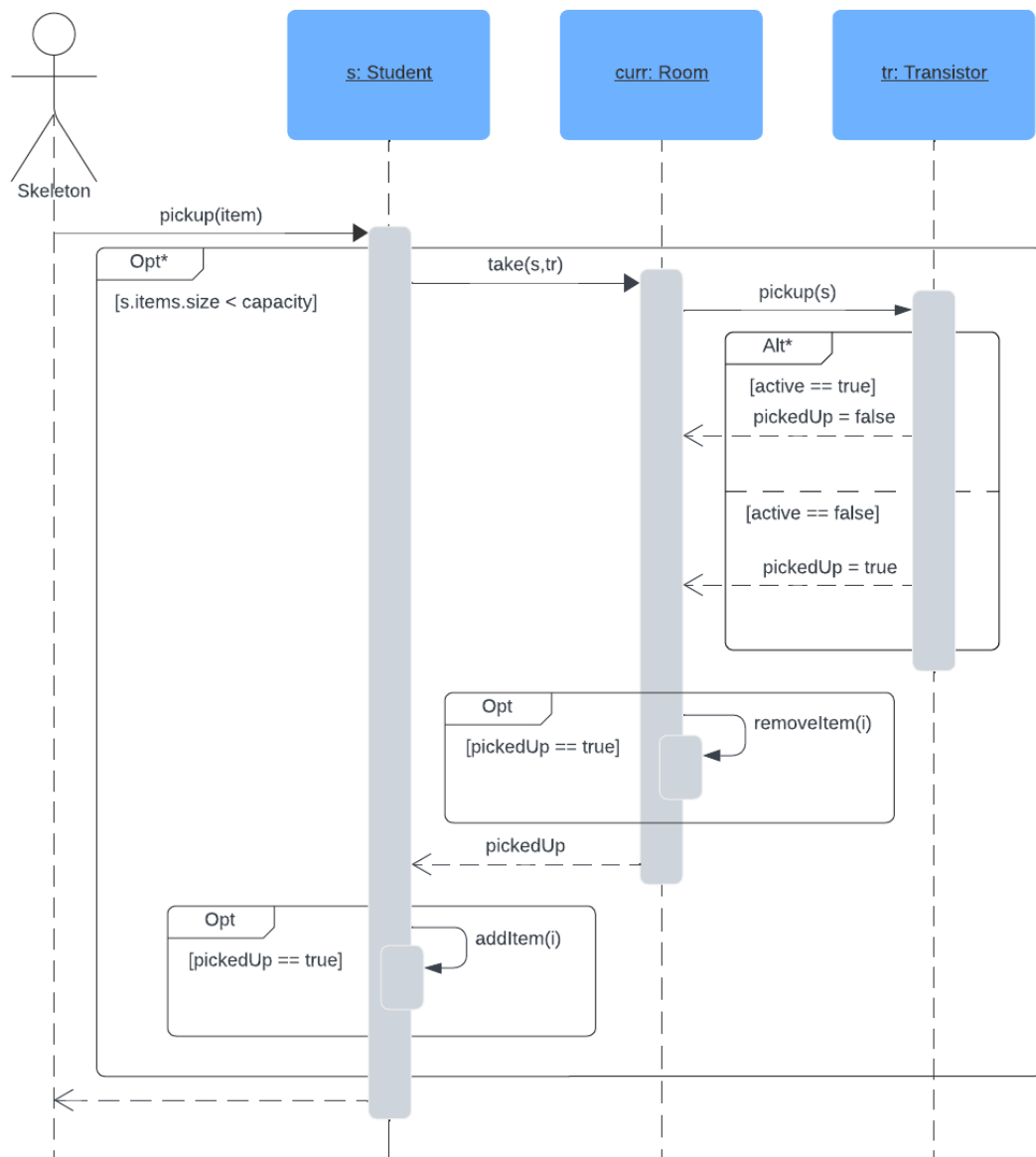
5.3.11 Professor moves to a Room



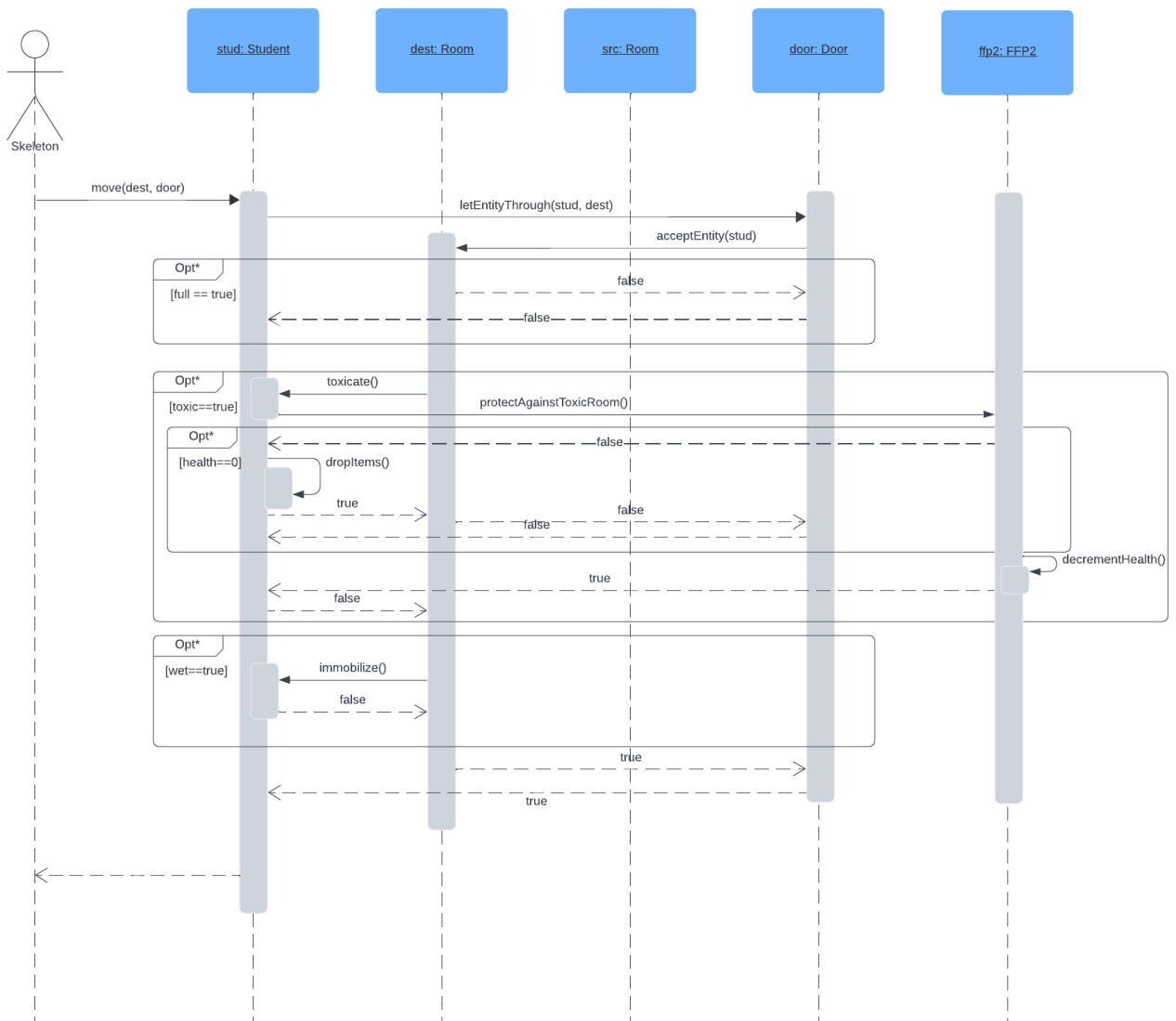
5.3.12 Student moves to a Room



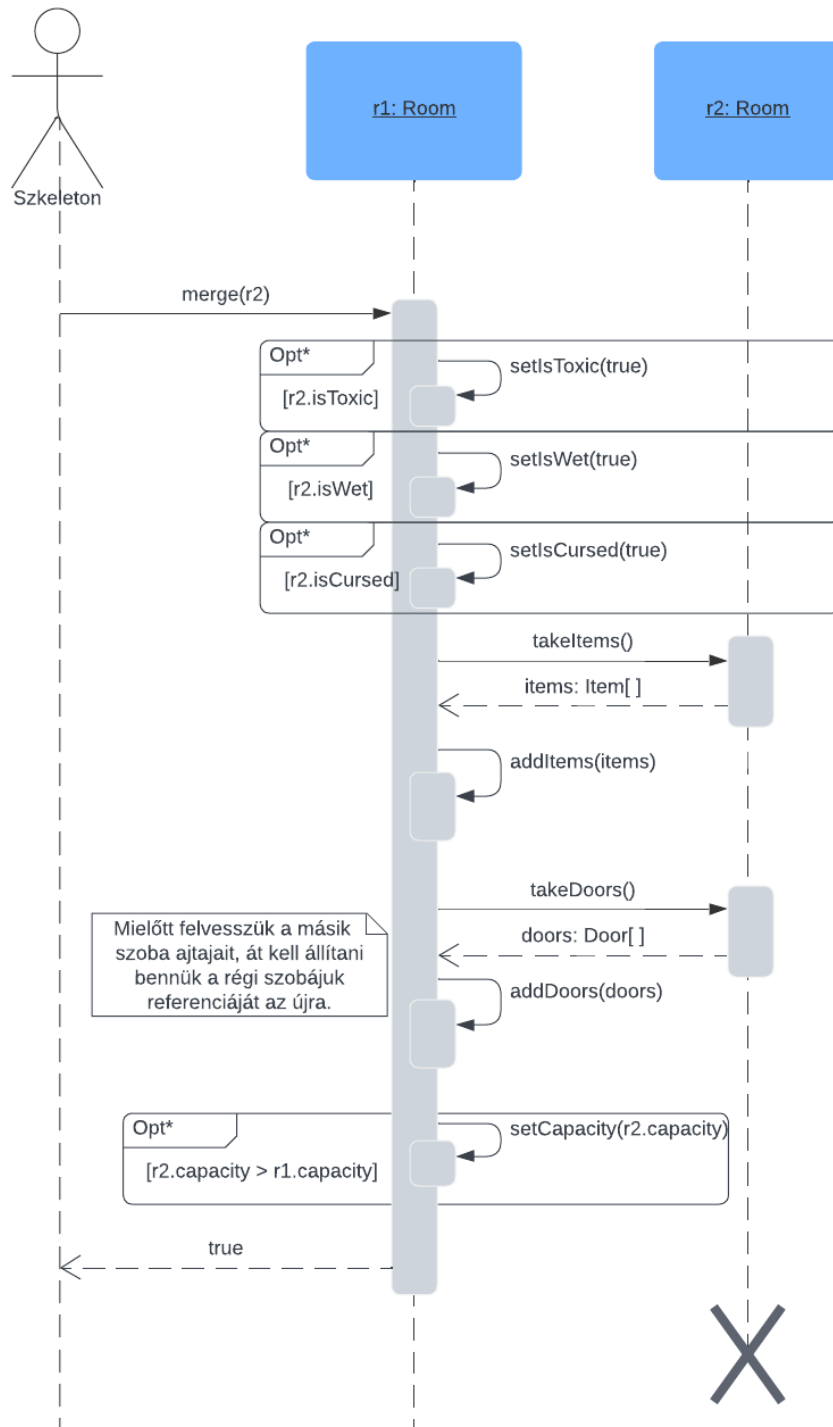
5.3.13 Student tries to pick up a Transistor



5.3.14 Student moves to Room with mask

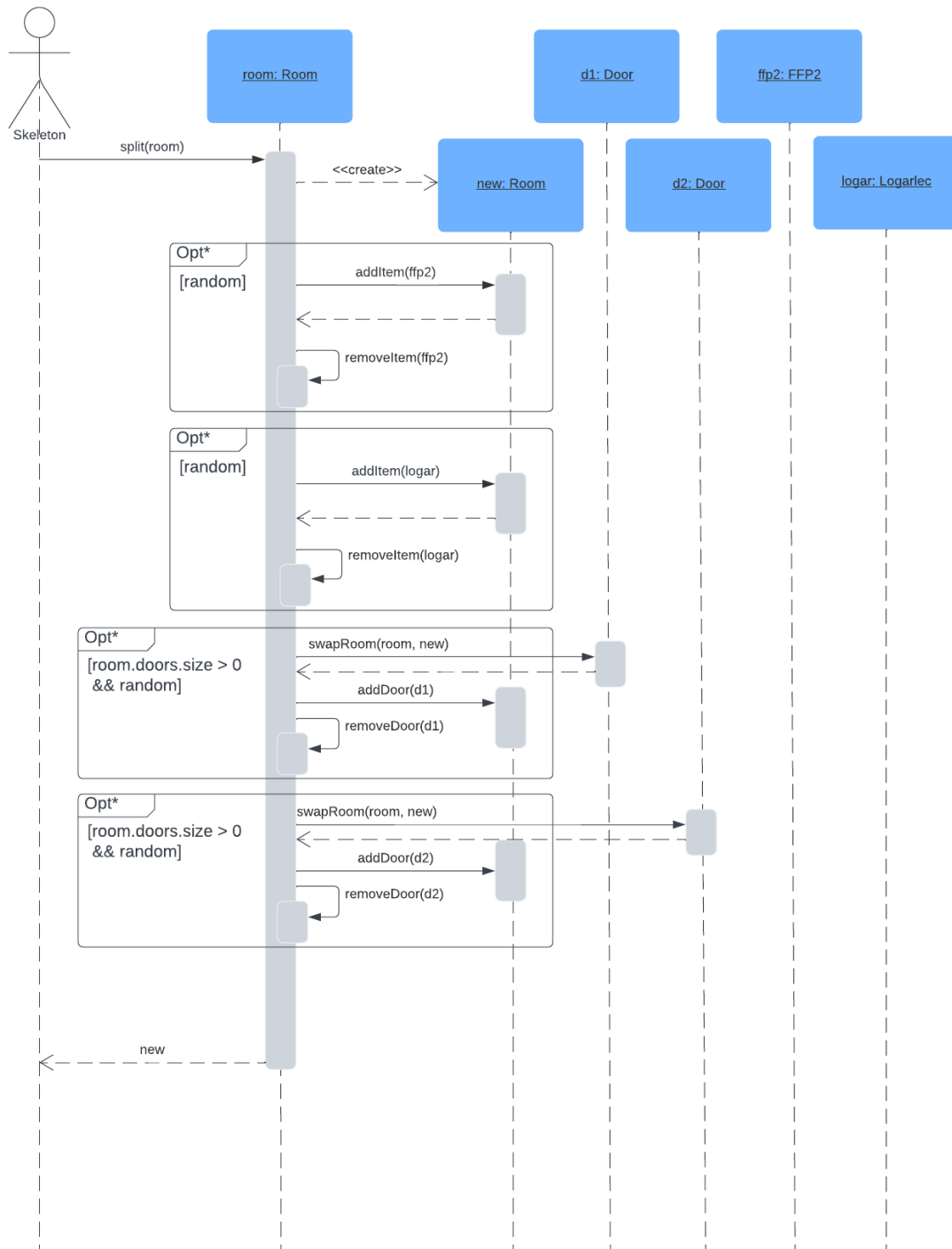


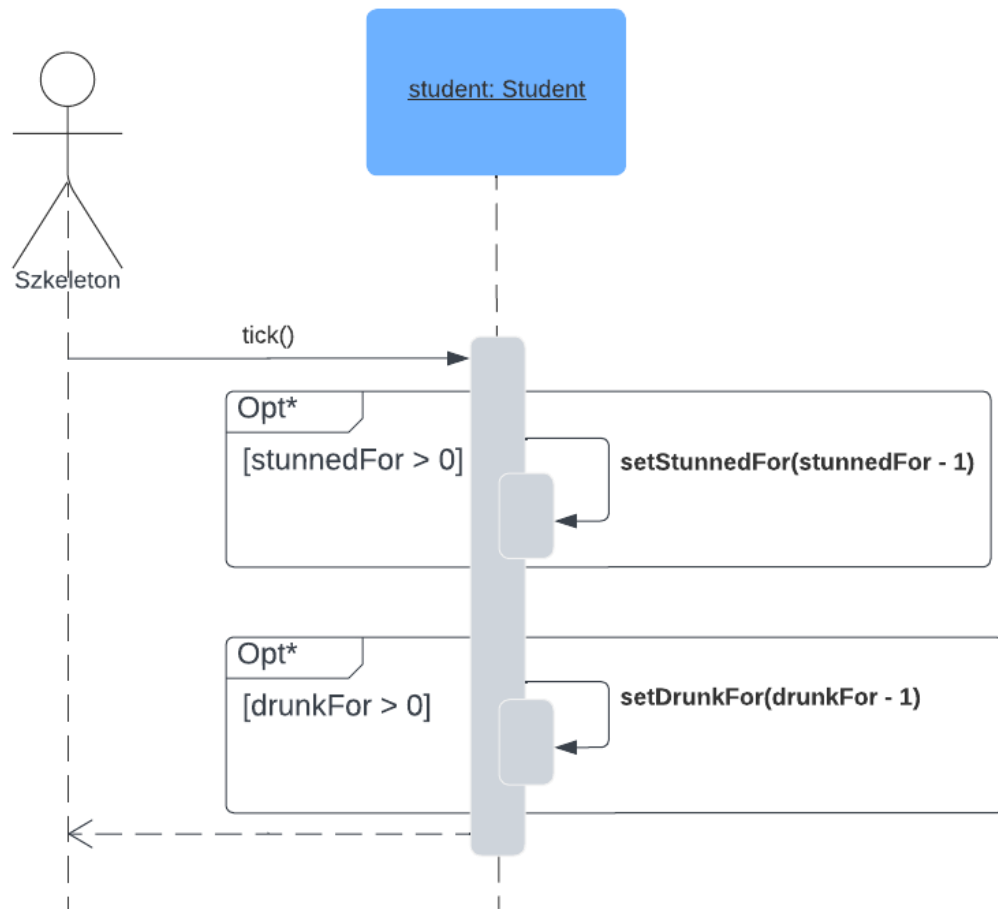
5.3.15 Merge Rooms



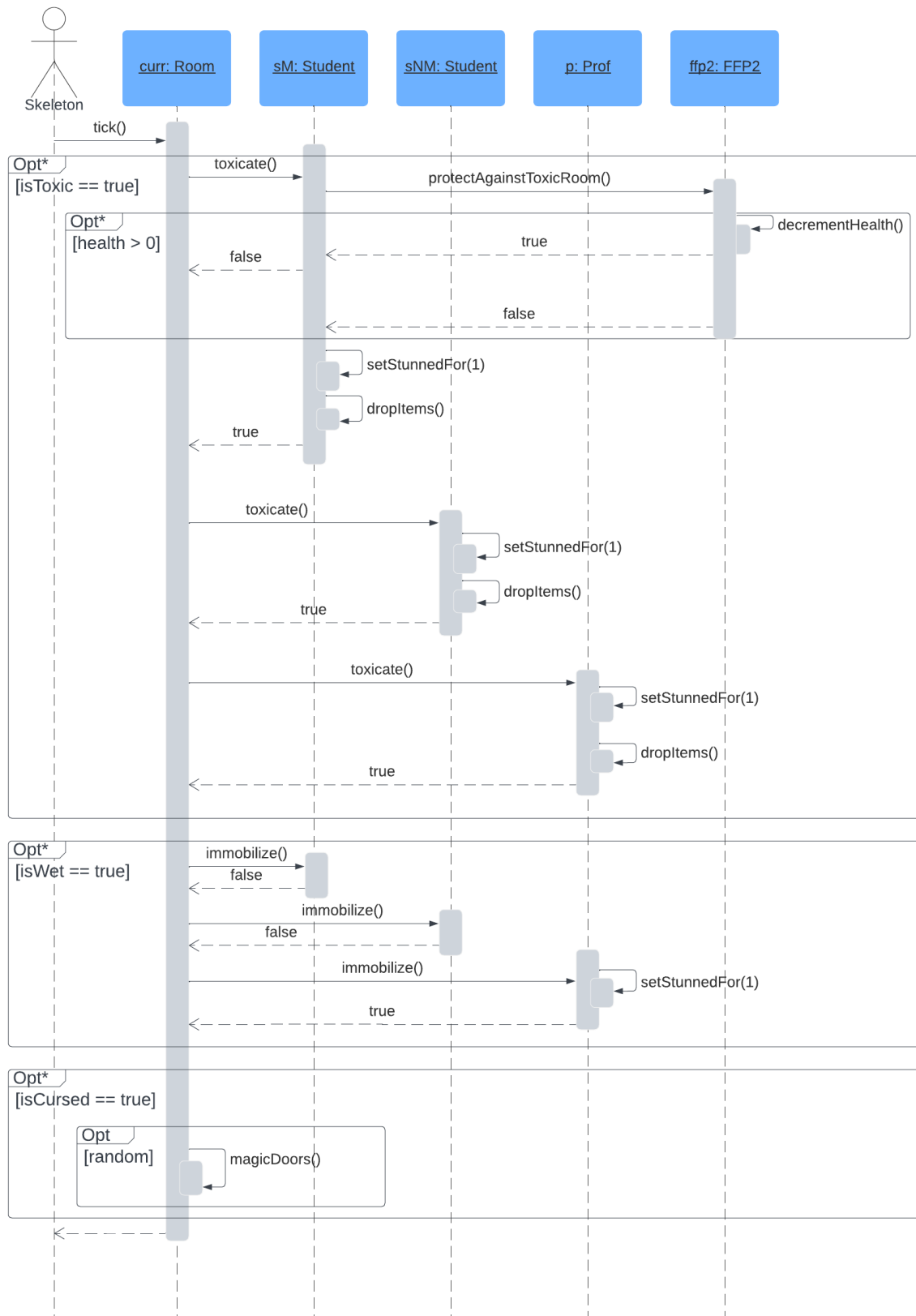
Valójában a `takeDoors()` és `takeItems()`-re sincs szükség, mert a szoba ismeri a másik szobának a privát adatait is, ezért le tudja akasztani a másik szobáról és fel tudja akasztani magára ezeket a gyűjteményeket. Tehát a metódusok csupán az értelmezést igyekeznek megkönnyíteni, ezért kerültek be.

5.3.16 Split Room

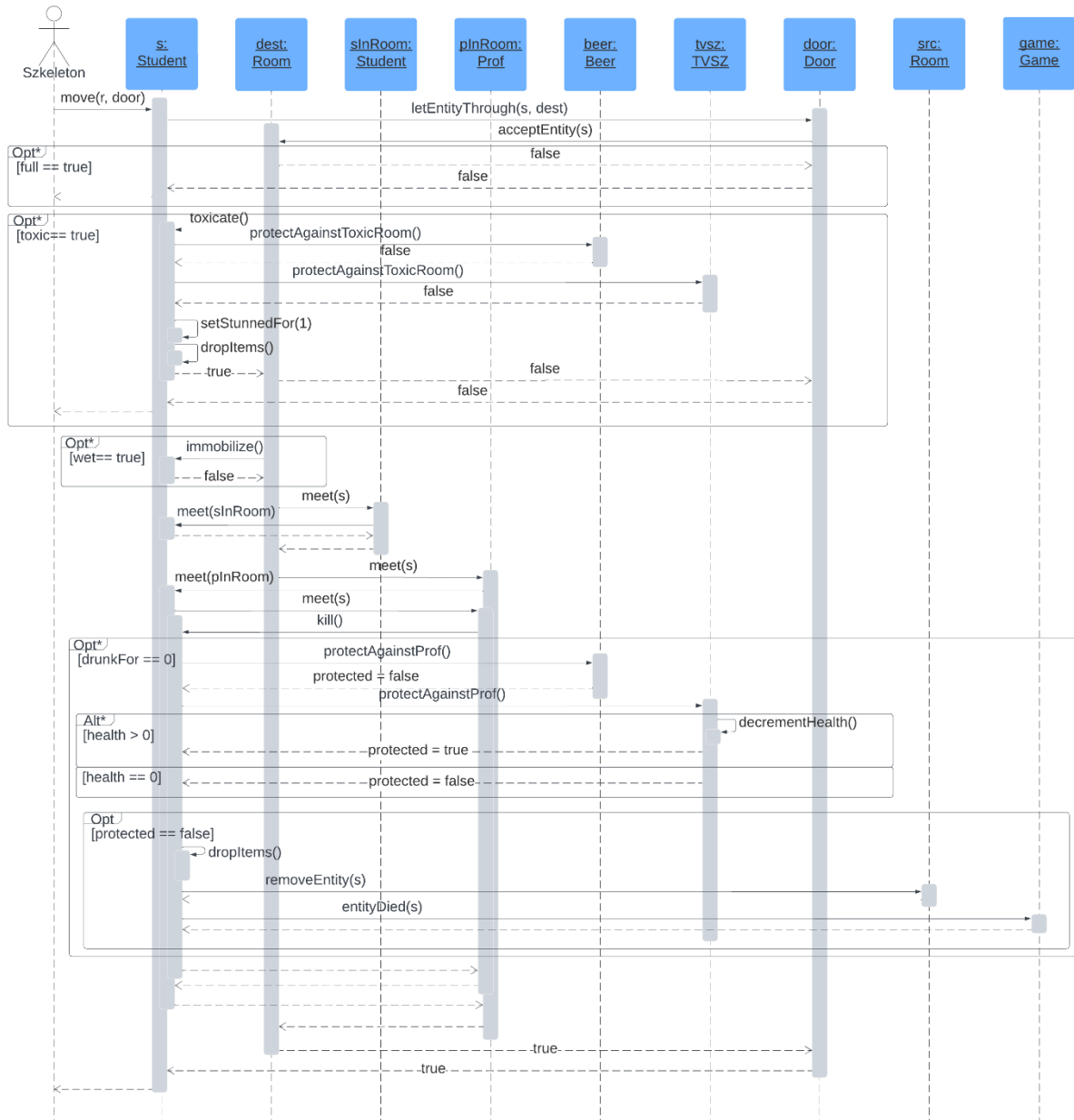


5.3.17 Student tick

5.3.18 Room tick



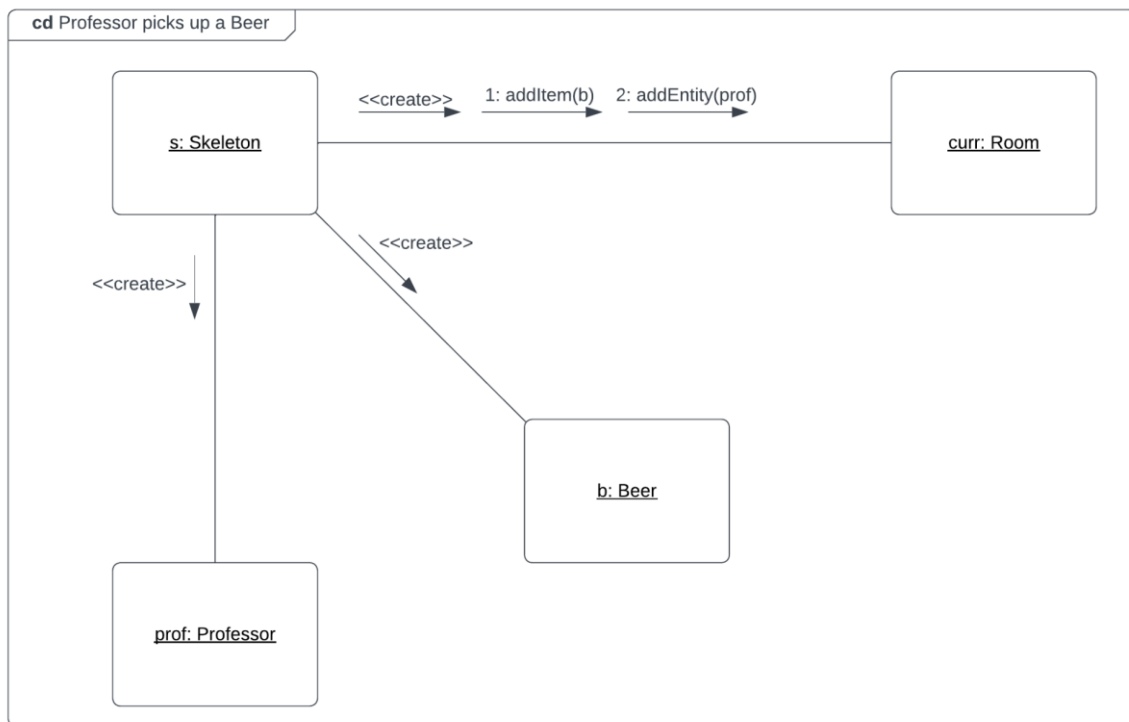
5.3.19 Meet others in Room

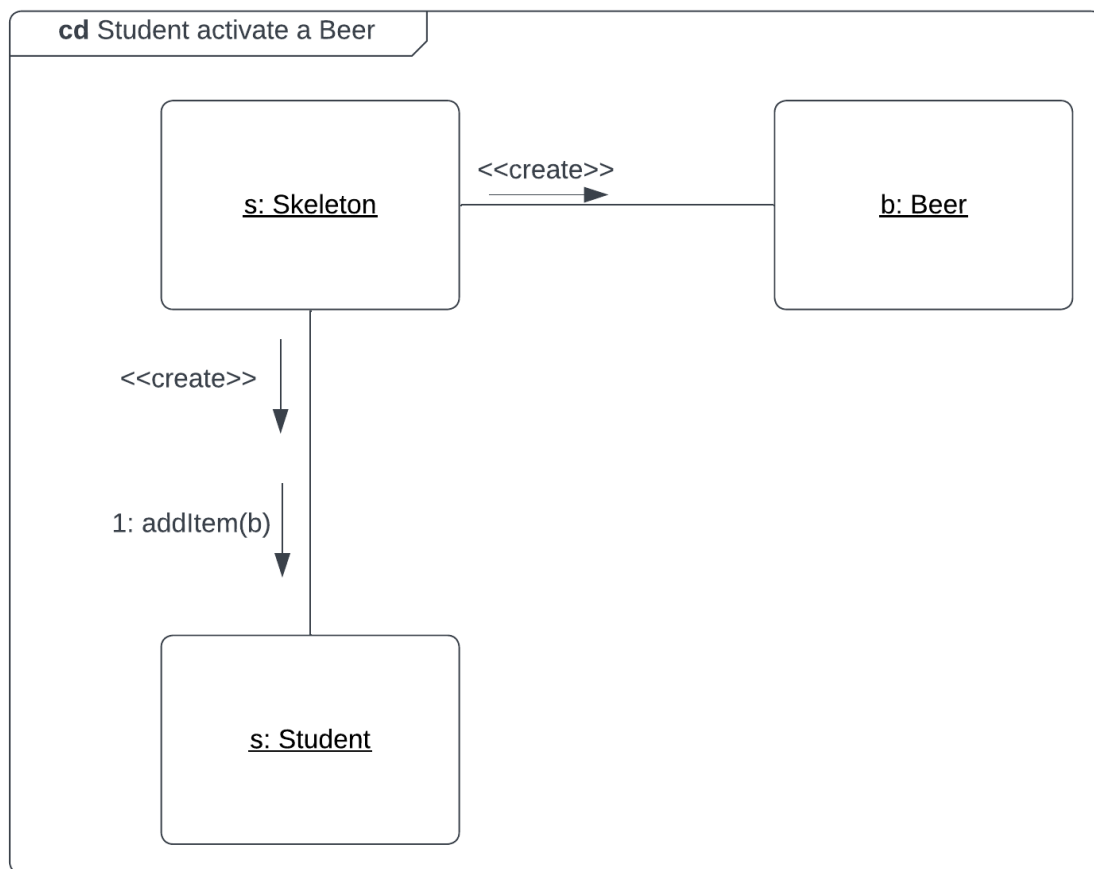


A * jelölés jelentése: az így megjelölt Opt és Alt elemek az objektumok belső állapota alapján döntenek, ezt a döntést a felhasználótól kérjük be a Szkeleton alkalmazásban. A többi, jelöletlen Opt és Alt elemek az előttük bekért döntések alapján működnek.

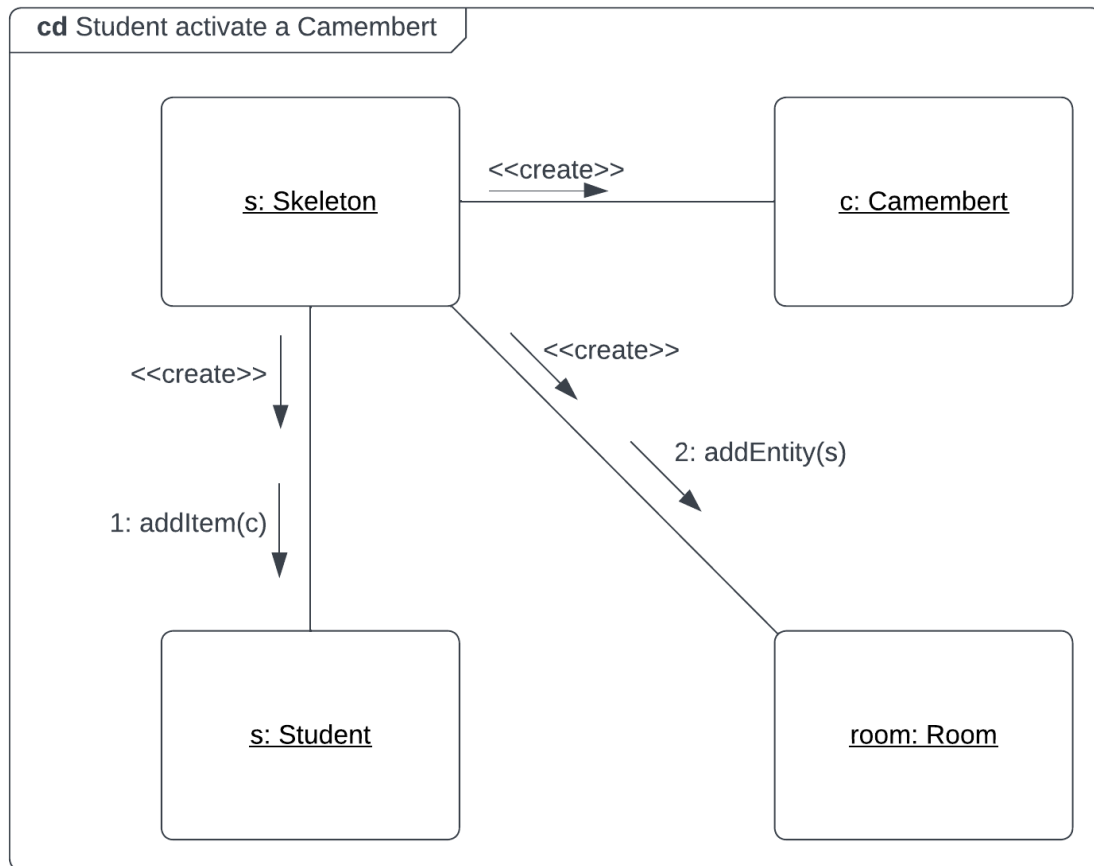
5.4 Kommunikációs diagramok

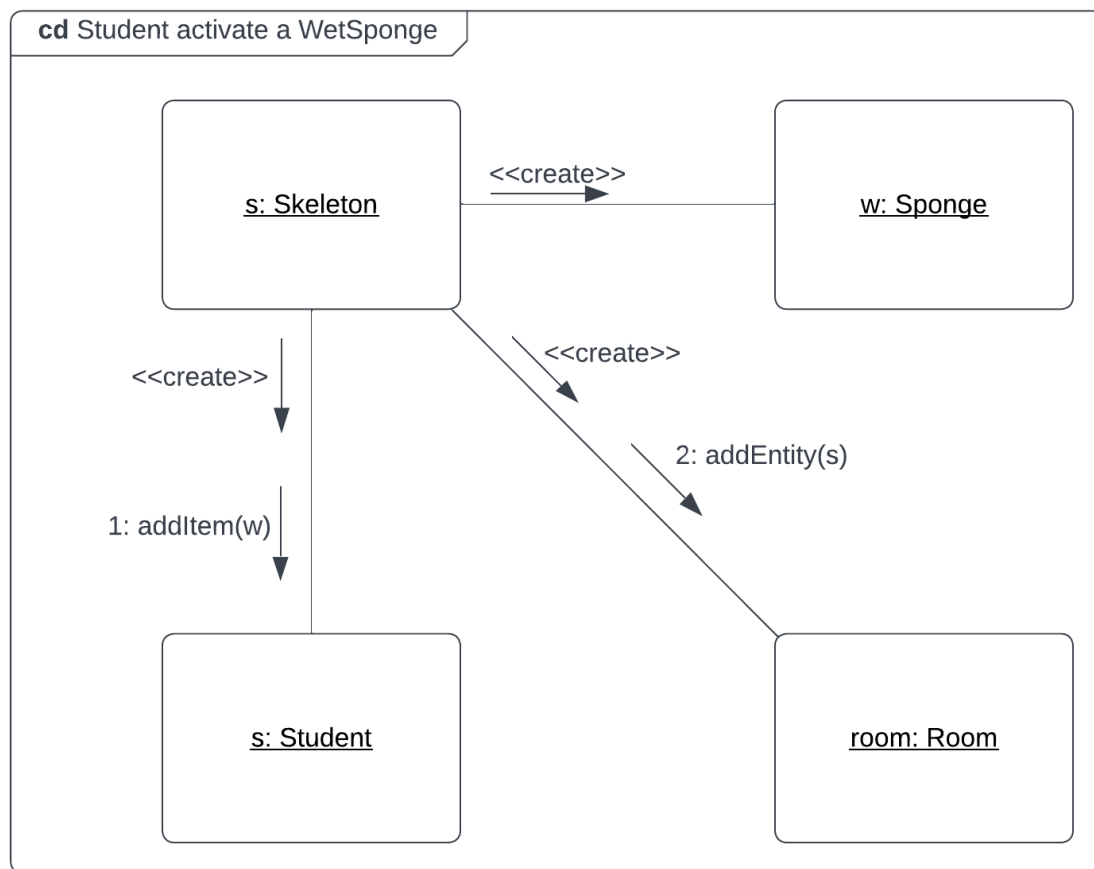
5.4.1 Professor picks up a Beer



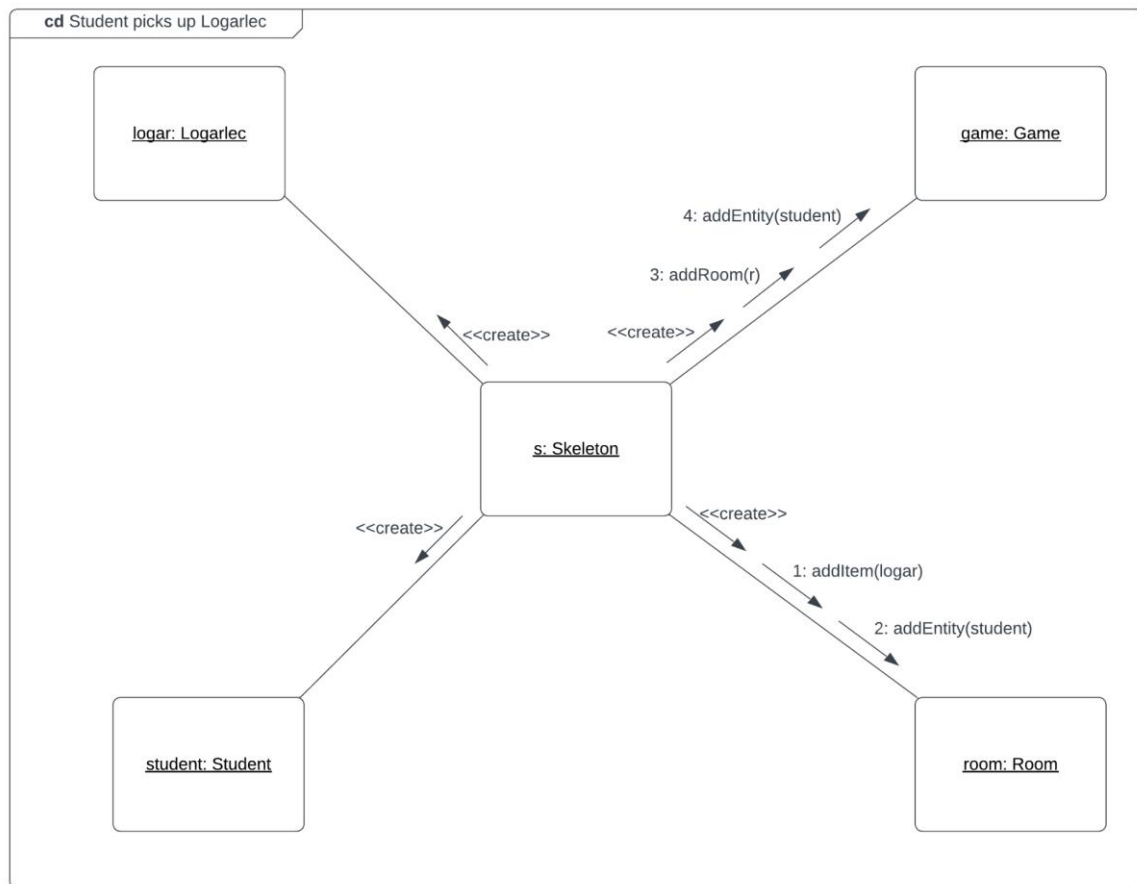
5.4.2 Student activates a Beer

5.4.3 Student activates a Camembert

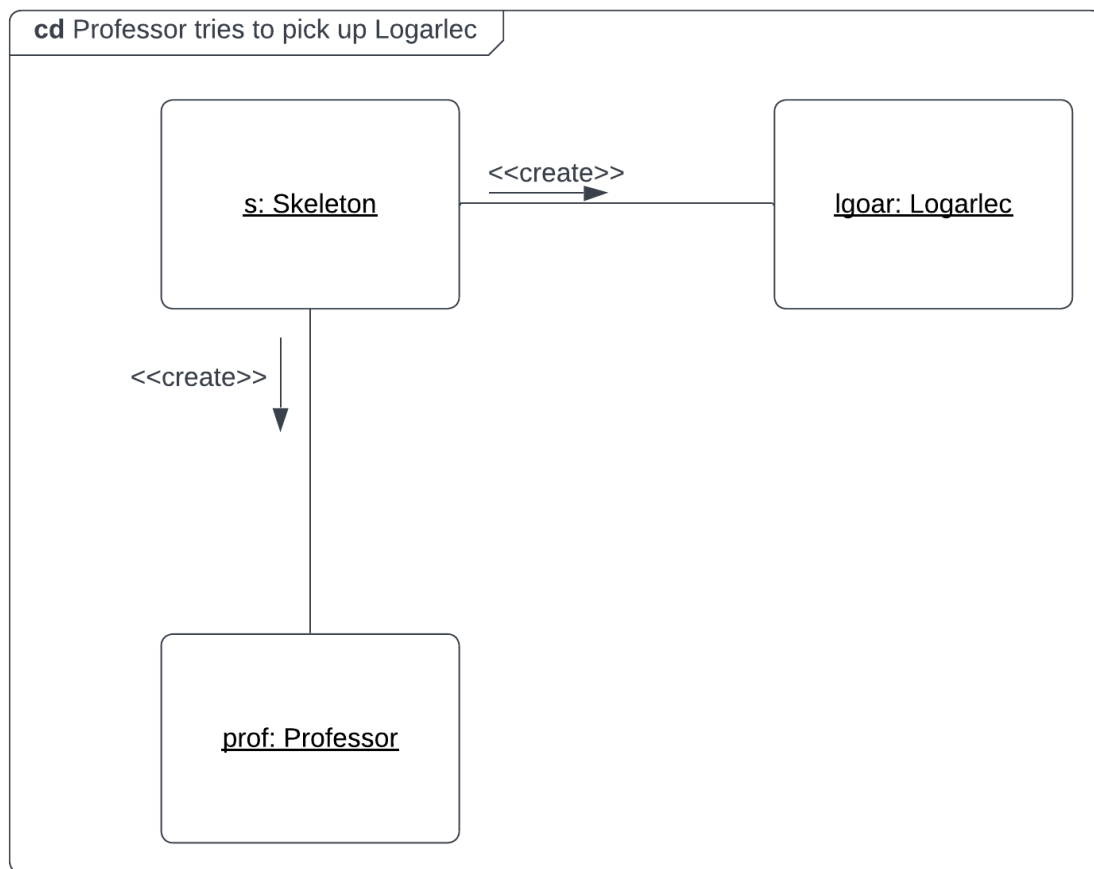


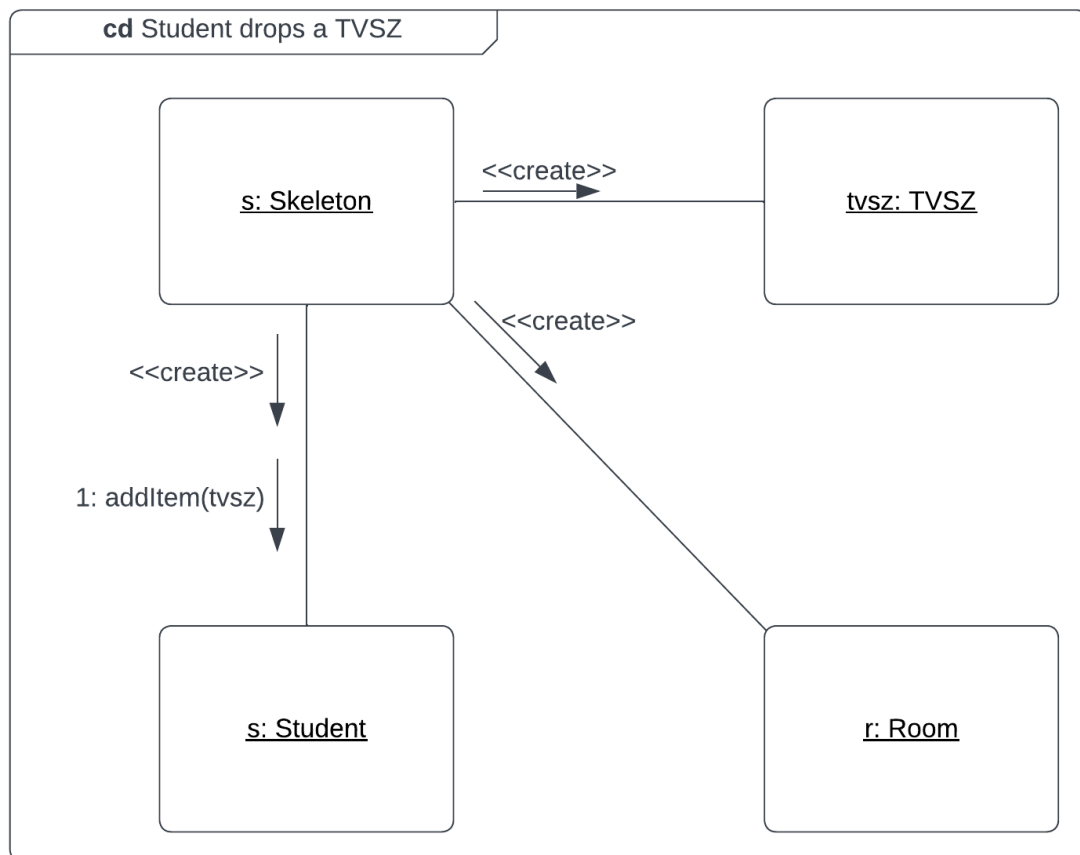
5.4.4 Student activates a WetSponge

5.4.5 Student picks up Logarlec

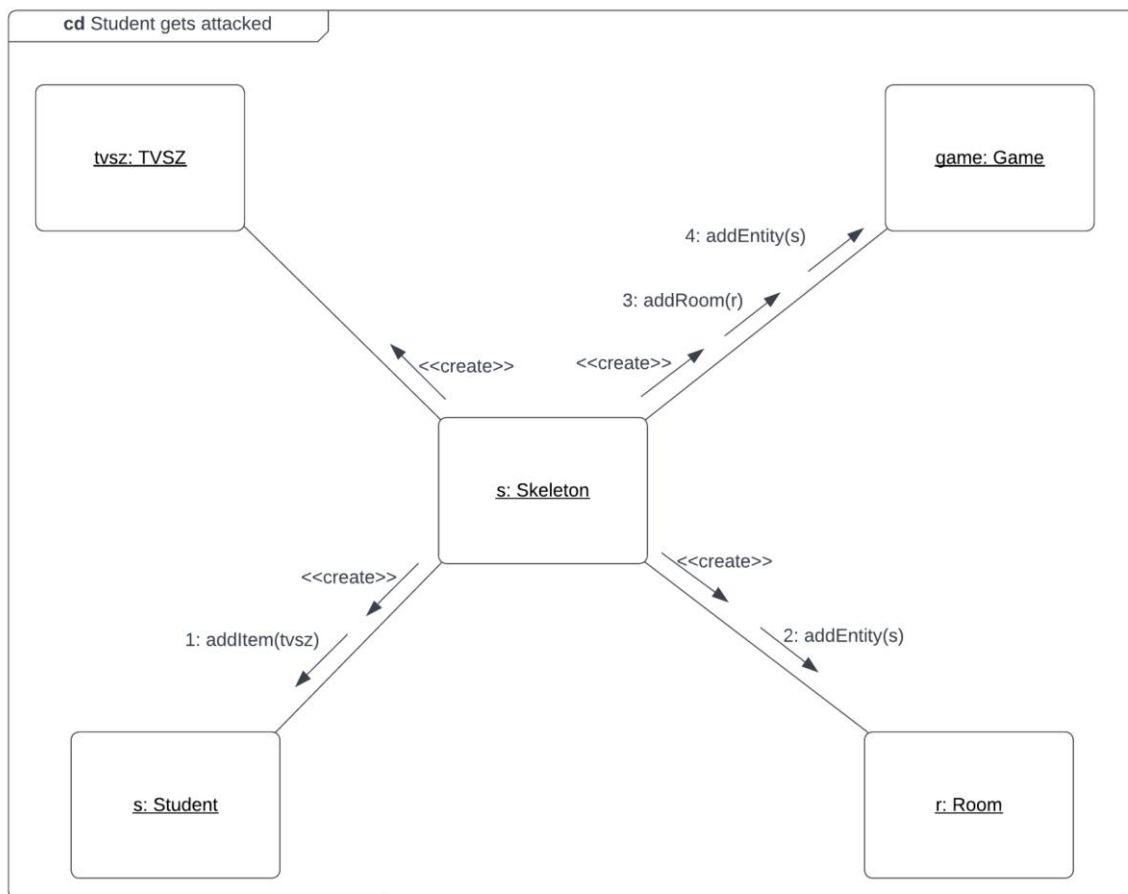


5.4.6 Professor tries to pick up Logarlec

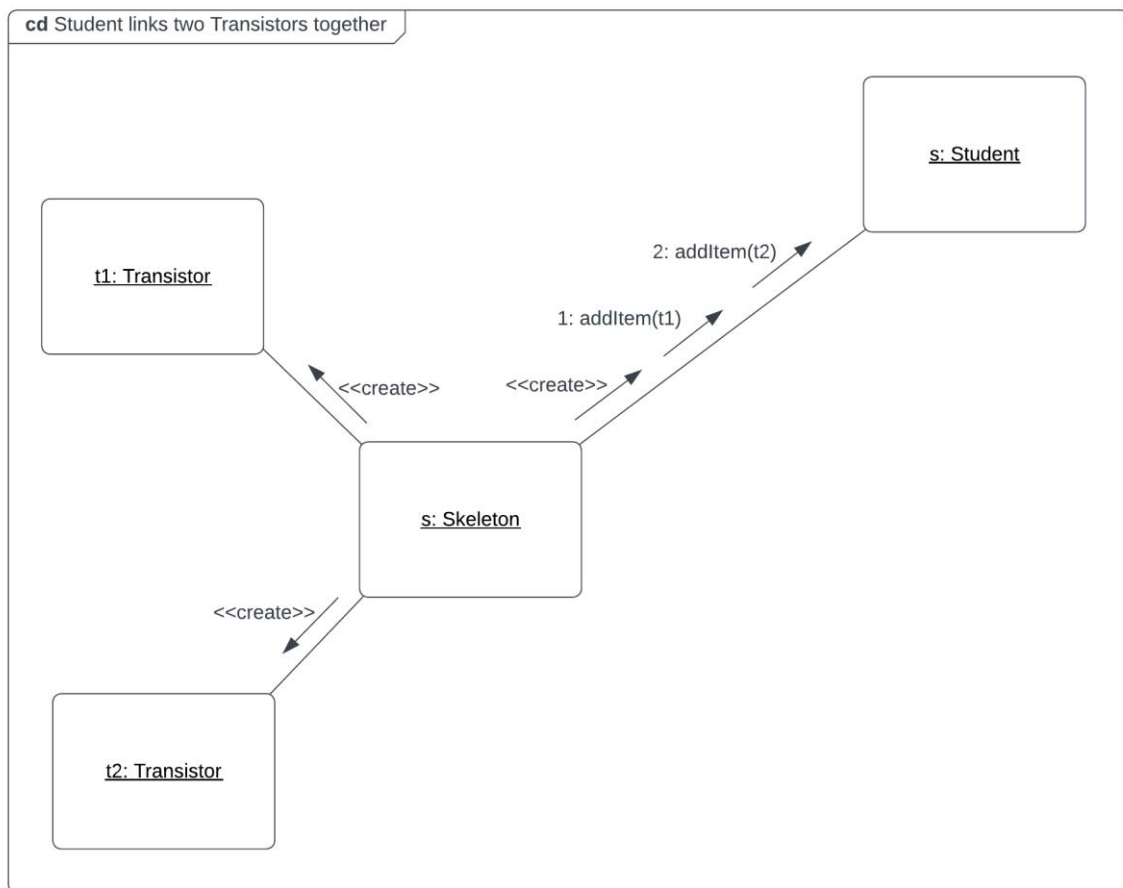


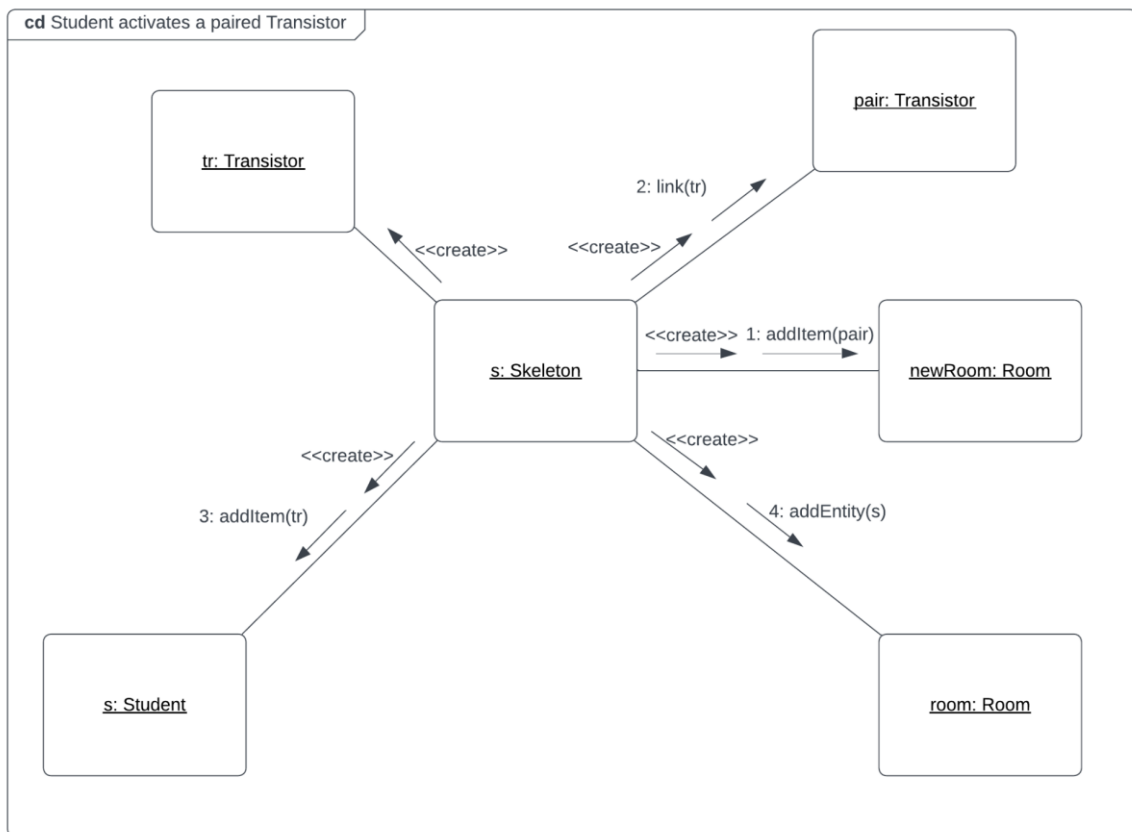
5.4.7 Student drops a TVSZ

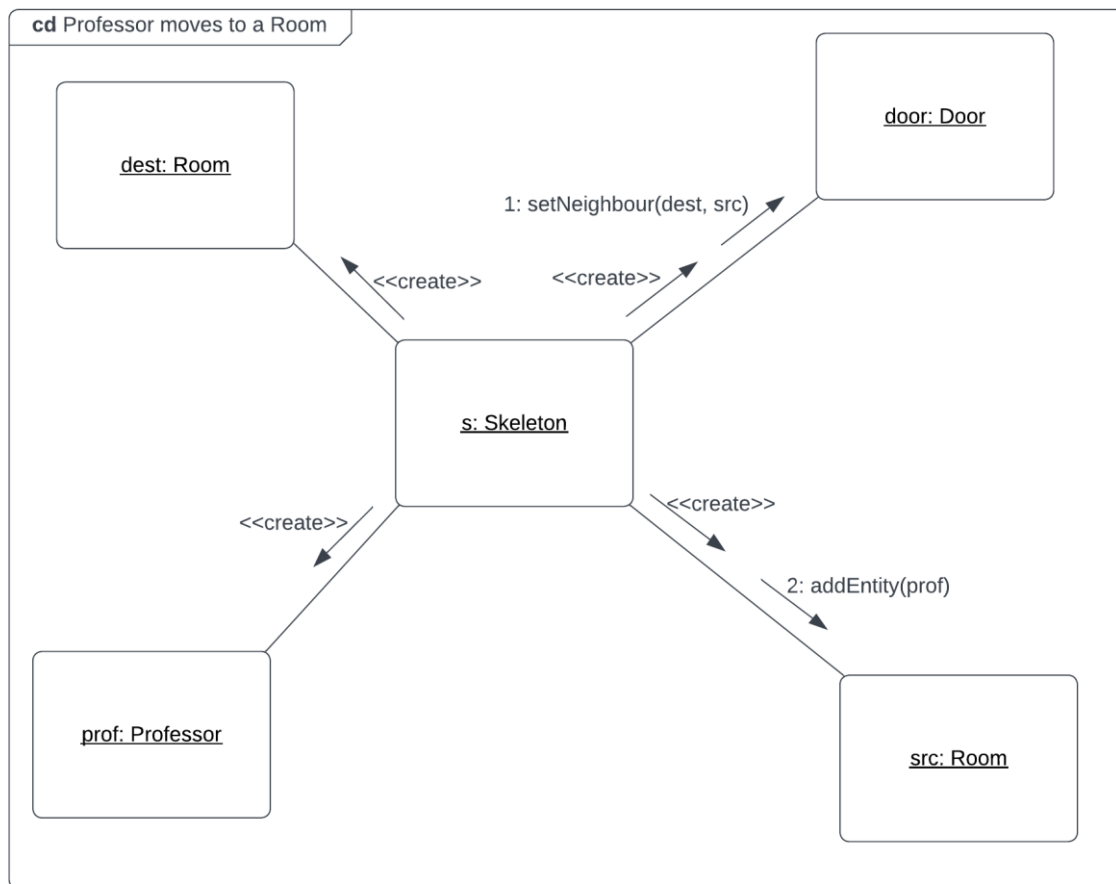
5.4.8 Student gets attacked

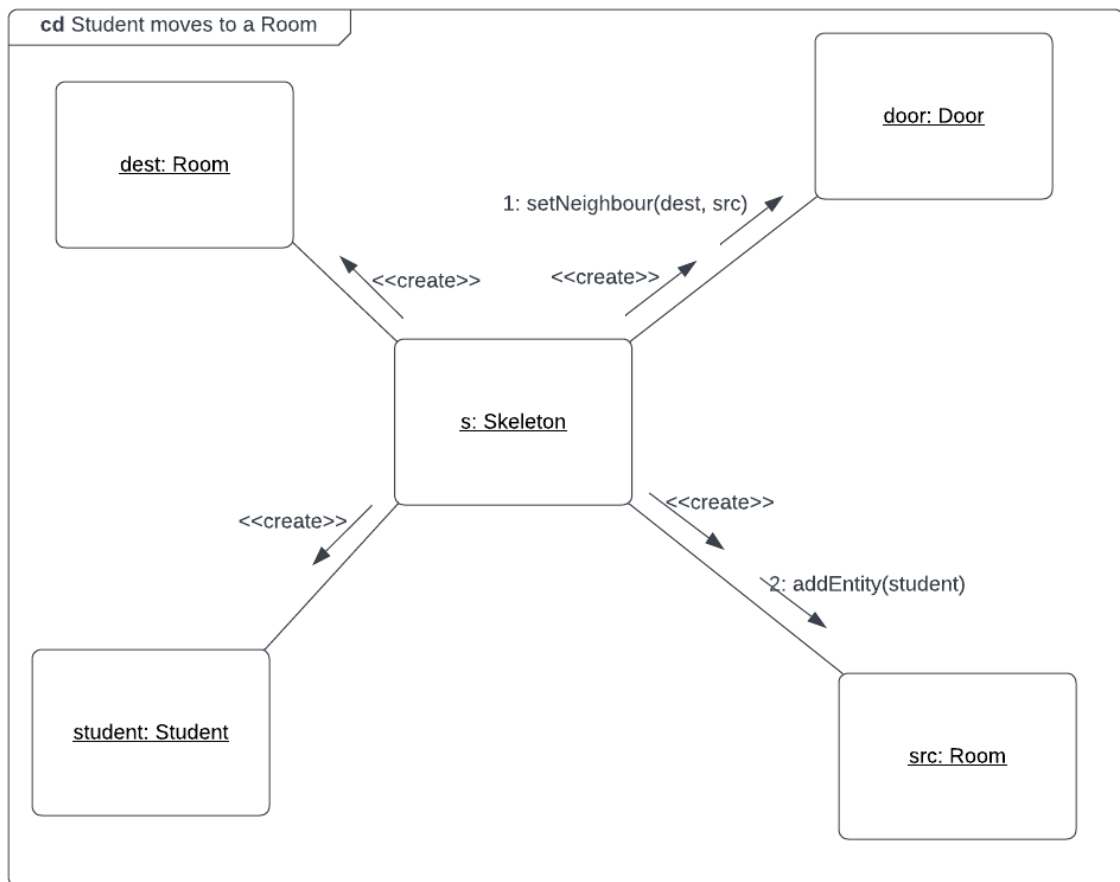


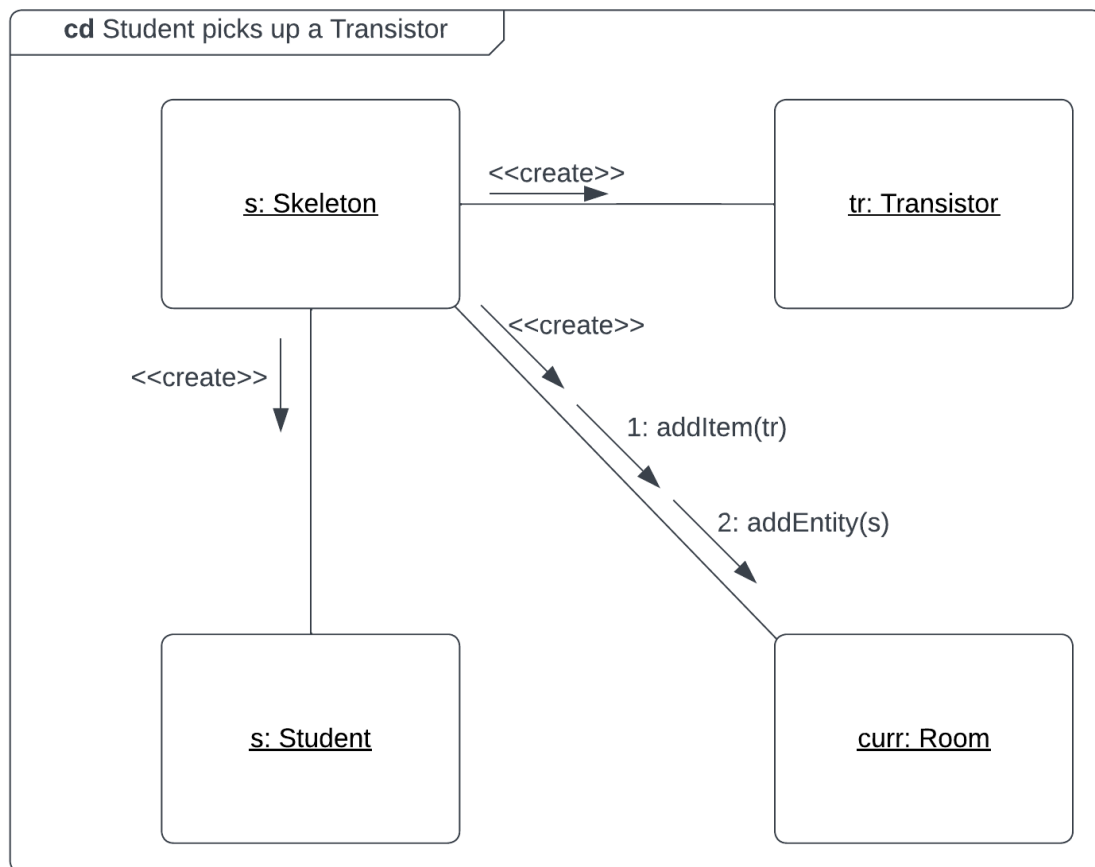
5.4.9 Student links two Transistors together

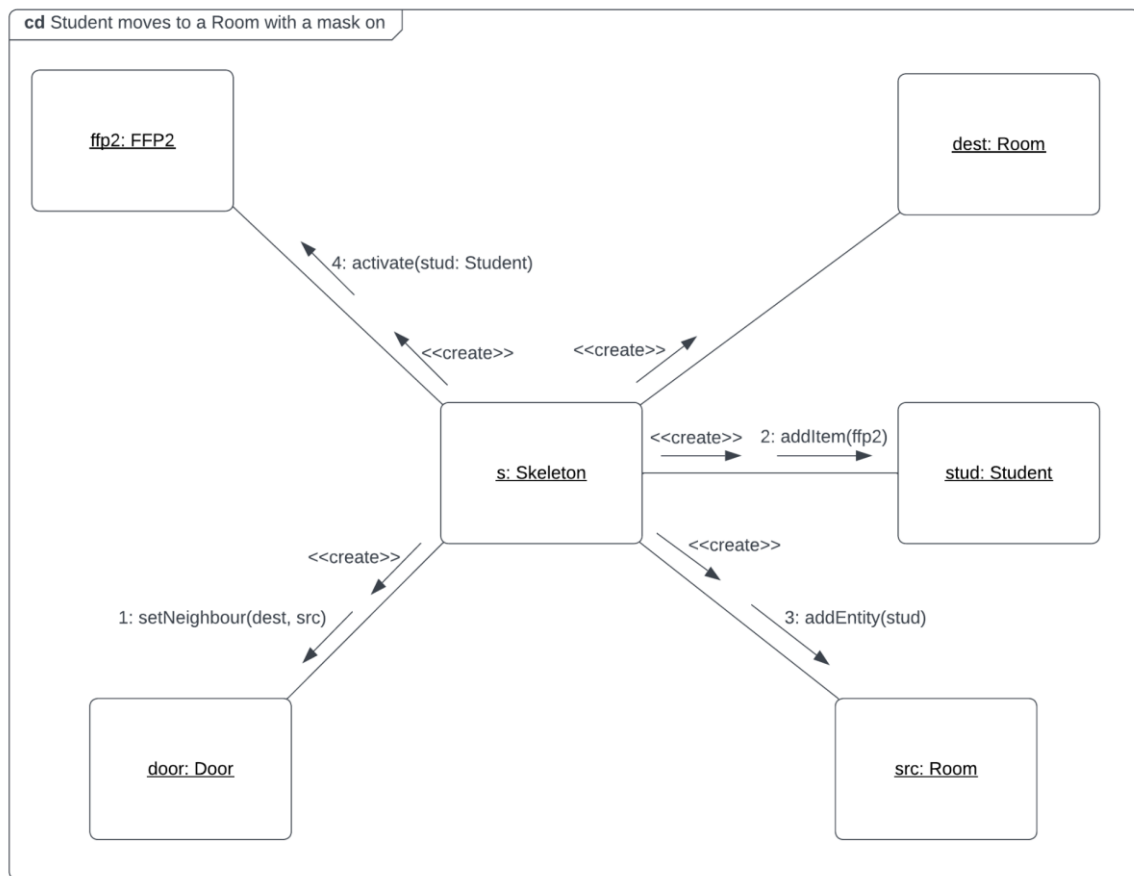


5.4.10 Student activates a Transistor

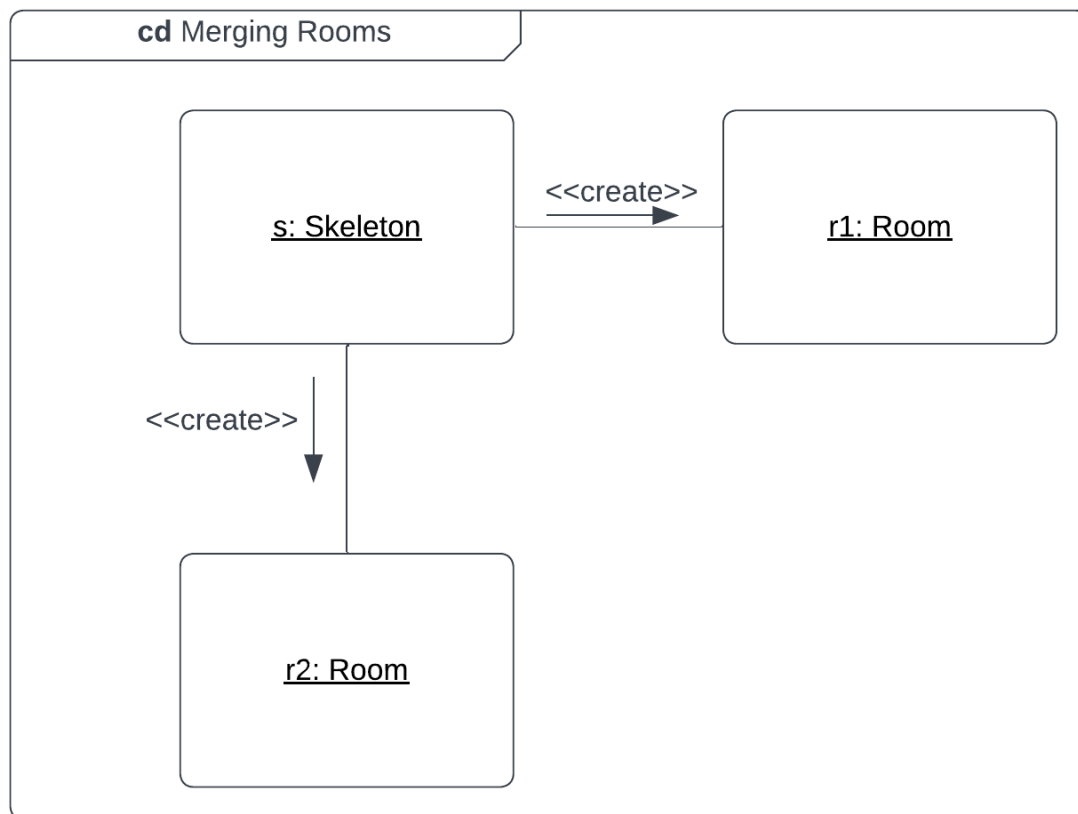
5.4.11 Professor moves to a Room

5.4.12 Student moves to a Room

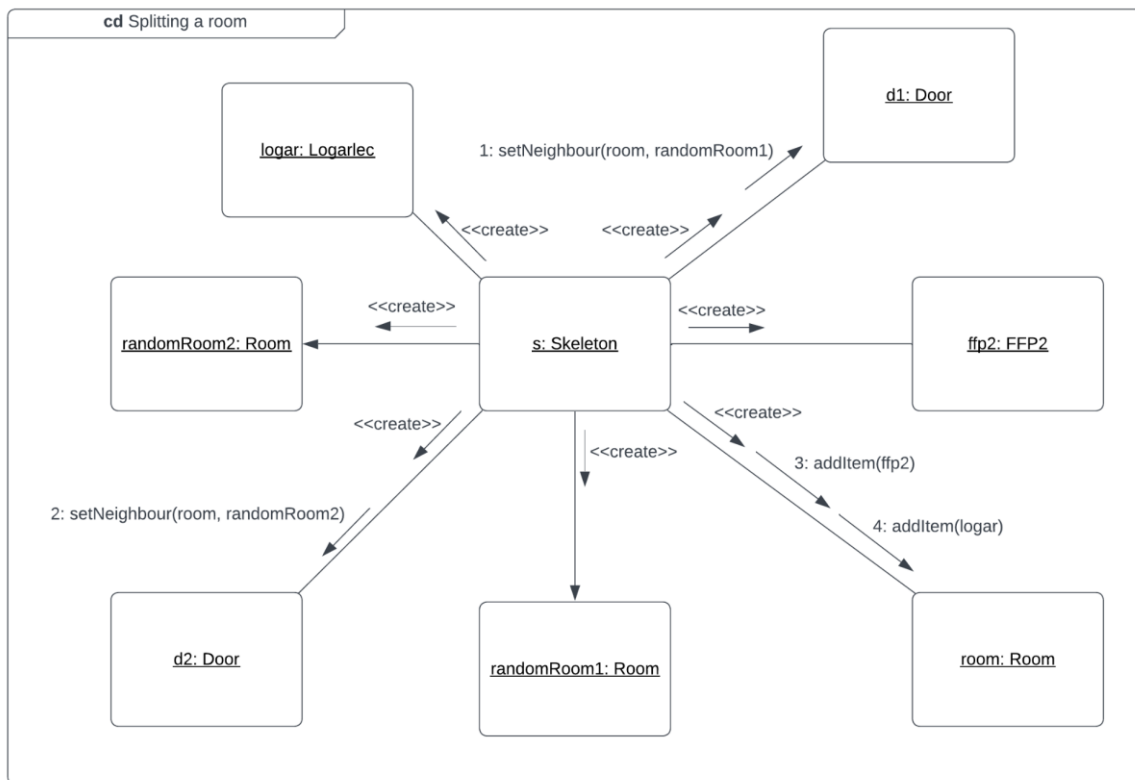
5.4.13 Student tries to pick up a Transistor

5.4.14 Student moves to a Room with a mask

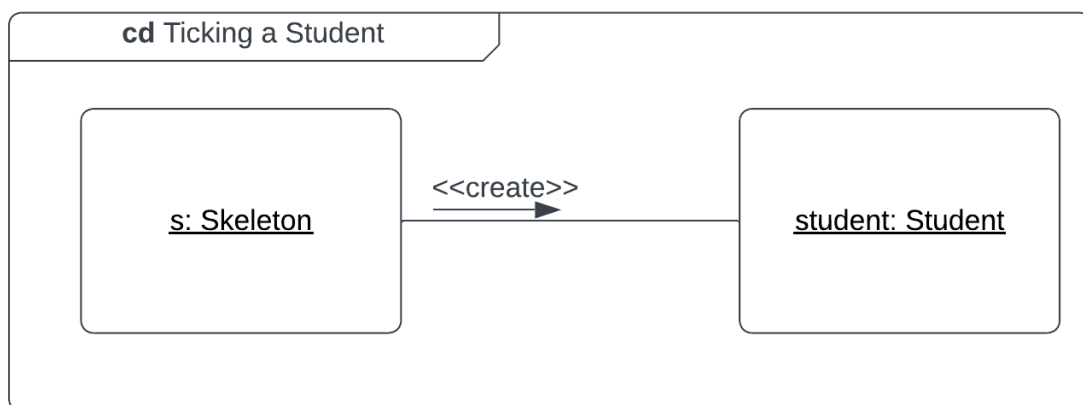
5.4.15 Merge Rooms



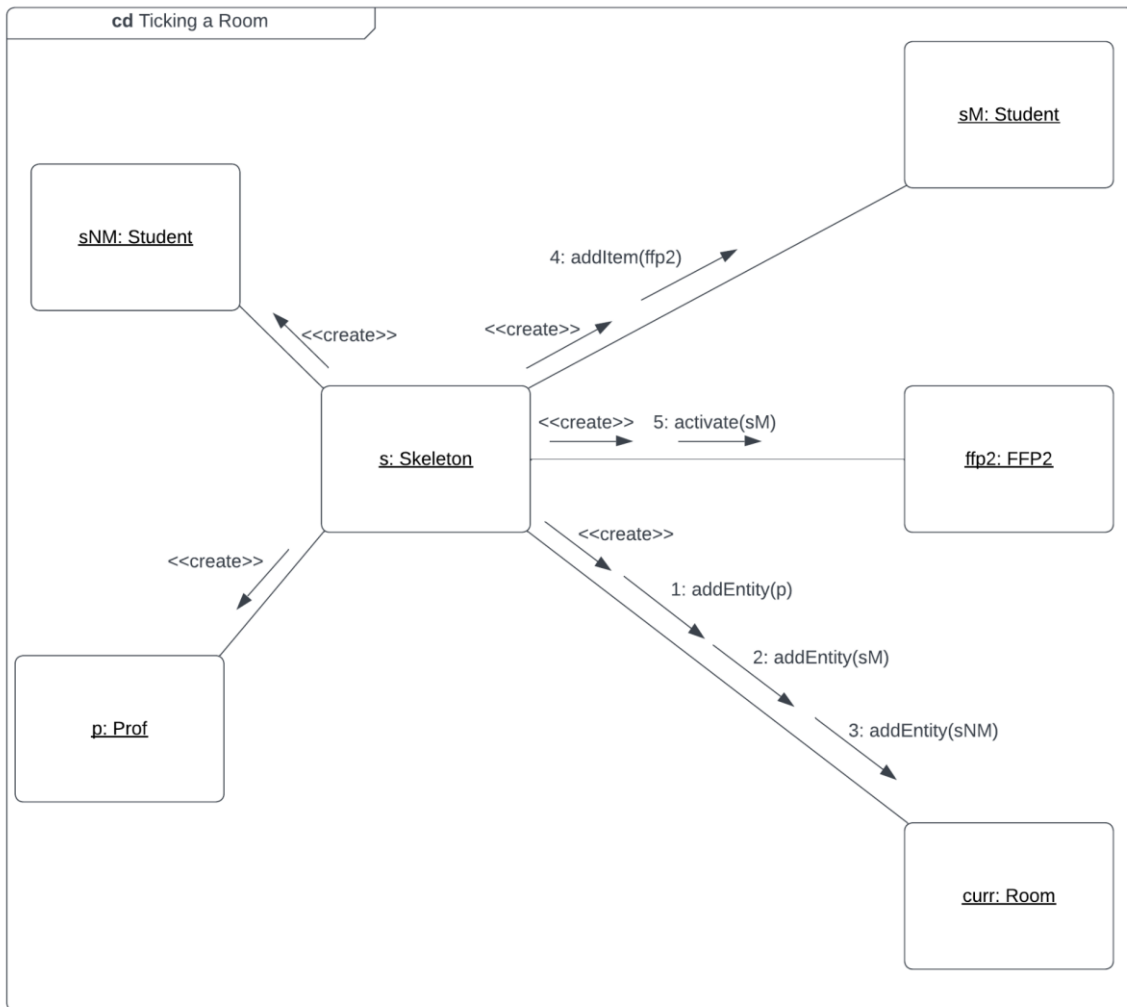
5.4.16 Split a Room



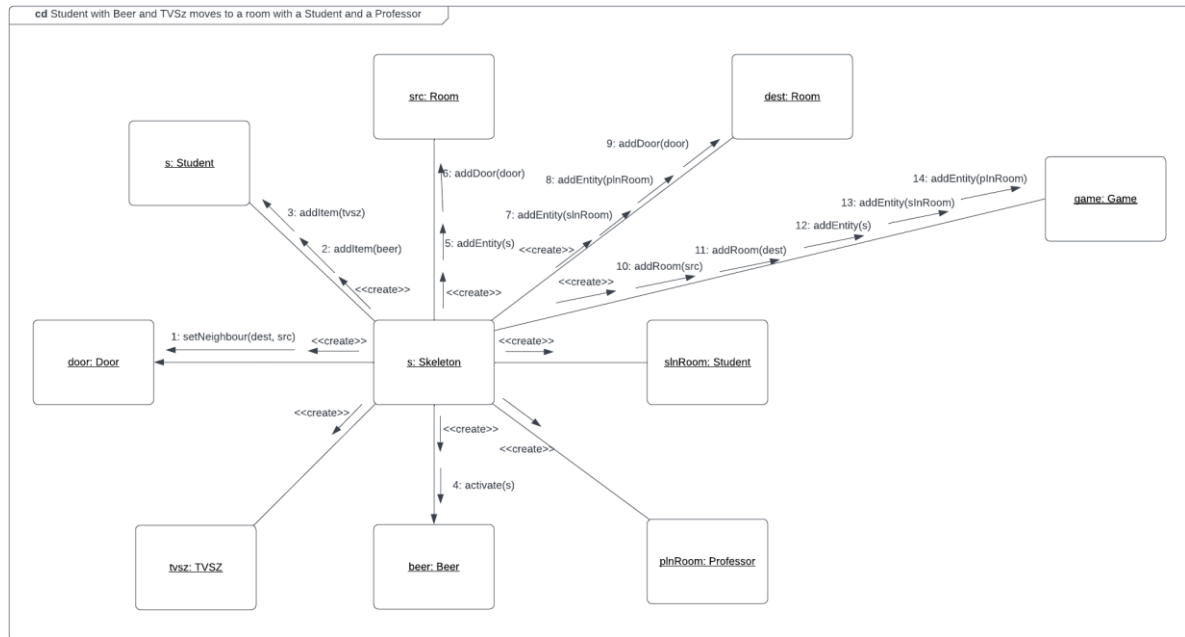
5.4.17 Student tick



5.4.18 Room tick



5.4.19 Meet others in room



5.5 Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.03.13. 14:30	0,1 óra	Csabuda Gulybán Nagy Papp Sánta	Értekezlet. Döntés: „Megcsináljuk!”
2024.03.14. 15:00	1 óra	Papp	Analízis modell hibajavításainak megírása.
2024.03.15. 17:00	2 óra	Papp Sánta	Tesztesetek kidolgozása.
2024.03.15. 20:00	0,5 óra	Nagy	Use-case keresés.
2024.03.16 14:00	1,5 óra	Papp Sánta	Szekvencia diagramok készítése.
2024.03.16 16:00	2 óra	Nagy Papp	Szekvenciák tervezése, modell újratervezése, hibajavítások.
2024.03.16. 18:00	1 óra	Nagy Sánta	Szekvencia diagramok megbeszélése/kidolgozása.
2024.03.16. 19:00	1 óra	Papp Sánta	Szekvencia diagramok megbeszélése/kidolgozása.
2024.03.17. 14:00	2 óra	Nagy	Szekvencia diagramok kidolgozás
2024.03.17. 16:00	4 óra	Papp	Szekvencia diagramok készítése
2024.03.17. 20:00	3 óra	Gulybán	Kommunikációs diagramok készítése
2024.03.17. 20:00	4 óra	Sánta	Use-case leírások, use-case diagram készítése, szekvencia diagram tervezése.
2024.03.17. 22:00	4,5 óra	Nagy	Szekvencia diagramok kidolgozás
2024.03.18. 8:00	2 óra	Nagy	Szekvencia diagram javítás
2024.03.18. 08:00	2 óra	Papp	Dokumentum szerkesztése