

4. Analízis modell kidolgozása

4.1 Objektum katalógus

4.1.1 Labirintus

Szobák által alkotott elágazó utak hálózata a Műegyetem Központi épülete alatt. A hallgatóknak itt kell keresniük a Logarlécet.

4.1.2 Szoba

A labirintus egyik eleme, ahol tárgyak találhatók és ahol esetlegesen oktatók és hallgatók is tartózkodhatnak. A szobákat ajtók kapcsolják össze.

4.1.3 Ajtó

Az egyes szobákat összekötő elem a labirintusban. Áthaladva rajta lehetőség van egyik szobából a másikba jutni.

4.1.4 Logarléc

A hallgatóknak fel kell kutatniuk és magukhoz kell venniük ezt a tárgyat a labirintusban, hogy megnyerjék a játékot.

4.1.5 TVSZ denevérbőrre nyomtatott példánya

Ez a különleges dokumentum három alkalommal képes megmenteni egy hallgató életét az oktatók ellen. Ezután elvesztik varázserejüket.

4.1.6 Szent söröspohár

Olyan tárgy, amely adott ideig képes megvédeni a hallgatót az oktatók ellen. Ez az idő azonban korlátozott.

4.1.7 Nedves táblatörlő rongy

Ez a tárgy megbénítja az oktatókat, ha egy szobába kerülnek vele. Hatása addig tart, amíg ki nem szárad.

4.1.8 Dobozolt káposztás camembert

Ha ez a sajt felbontásra kerül, mérgező gázt bocsát ki a szobába, ami átmenetileg veszélyessé teszi a szobát.

4.1.9 FFP2-es maszk

Speciális maszk, amely védelmet nyújt a mérgező gázok ellen. Ez azonban csak korlátozott ideig tart.

4.1.10 Tranzisztor

Speciális tárgyak a labirintusban, amelyeket a hallgatók összekapcsolhatnak és használhatnak arra, hogy áthelyezzék magukat a labirintus különböző részeibe.

4.1.11 Oktató

Az ellenfelek a játékban, akik megpróbálják megakadályozni a hallgatókat abban, hogy sikeresen megtalálják a Logarléceket. Ha egy szobában vannak egy vagy több hallgatóval, akkor elveszik a hallgatók lelkét, és azok kibuknak az egyetemről.

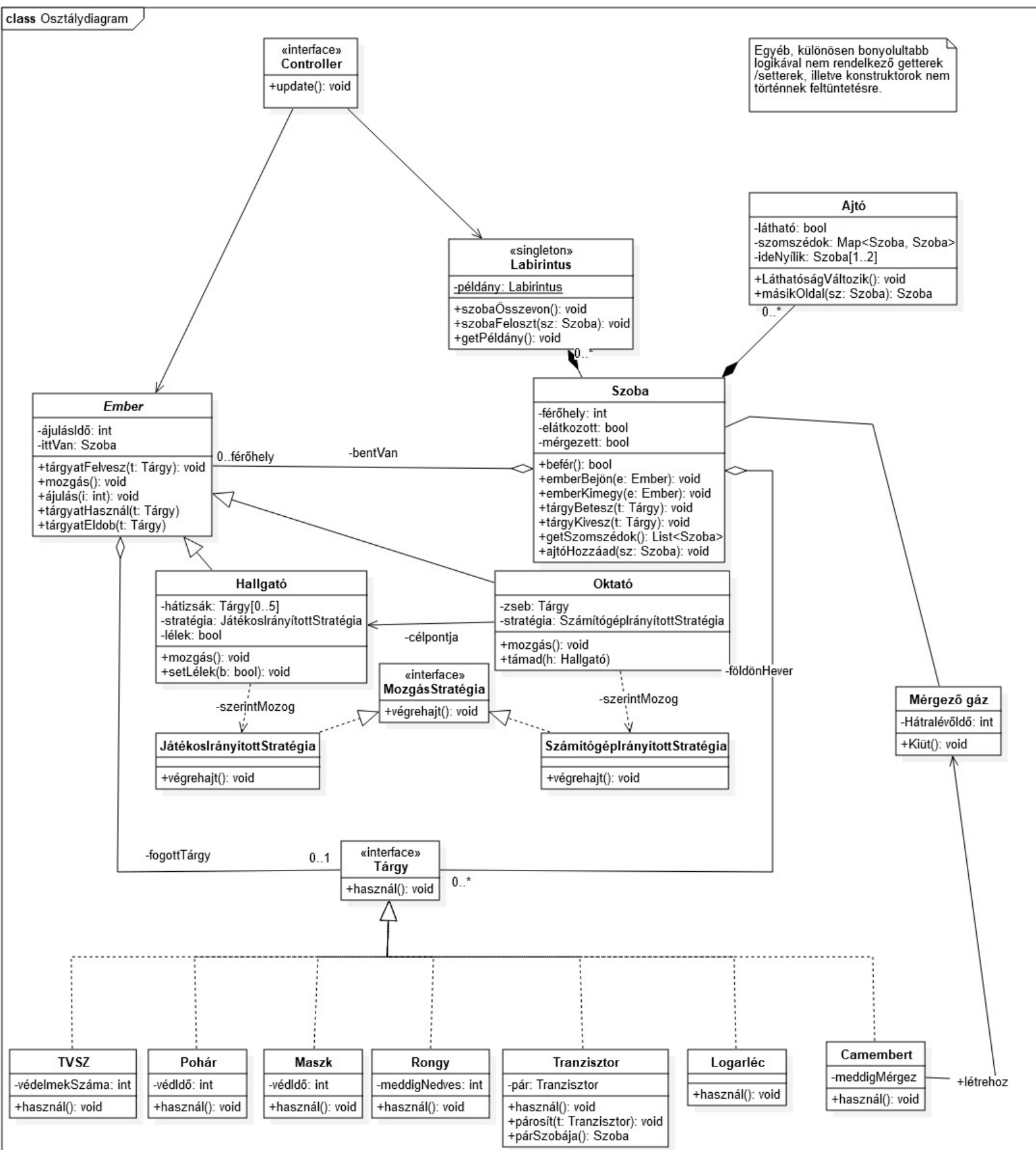
4.1.12 Mérgező gáz

Egy veszélyes elem a labirintus egyes szobáiban, amitől a benne tartózkodók eszméletüket veszítik. Például, ha egy dobozolt káposztás camembert felbontanak, ideiglenesen mérgező gázt bocsát ki.

4.1.13 Hallgató

A játékos által irányított karakter, aki tárgyak használatával szobáról-szobára járkálva keresi a Logarléceket, aminek segítségével nyerhet. Oktatók akadályozzák útja során, akiket bizonyos tárgyak segítségével megbéníthat, vagy támadásukat kivédekezheti.

4.2 Statikus struktúra diagramok



4.3 Osztályok leírása

4.3.1 Ajtó

Felelősség:

- Az Ajtó osztály felelős az egyes szobákat összekötő ajtók kezeléséért és működtetéséért a labirintusban. Feladata az ajtók állapotának nyomon követése, valamint annak ellenőrzése, hogy nyitható-e a kért irányba az ajtó. Az Ajtó osztály biztosítja a játékosok számára az átjutást a különböző szobák között azáltal, hogy lehetővé teszi az áthaladást az ajtókon keresztül.

Össztályok:

- Nincs specifikált össztály.

Interfészek:

- Nincs specifikált interfész.

Asszociációk:

- Kapcsolatban van a Szoba osztállyal, és így tudja melyik két szoba között helyezkedik el, illetve, hogy melyik szobából lehet kinyitni esetleg, ha csak egy irányba nyílik.

Attribútumok:

- szomszédok: Map<Szoba, Szoba>
- látható: bool
- ideNyílik: Szoba[1..2]

Metódusok:

- **void láthatóságVáltozik():** Ez a metódus eltávolítja vagy megjeleníti az ajtót a labirintusból. Amikor ezt a metódust meghívják, az ajtó eltűnik vagy megjelenik. Átállítja a látható attribútumot
- **Szoba másikOldal(Szoba):** Paraméterként megkap egy szobát, aminek a túloldalát keressük az ajtó mentén. Visszaadja a Szomszédok Map-ból azt a szobát, ami az ajtó túloldalán van.

4.3.2 Camembert

Felelősség:

- A Camembert osztály felelős a mérgező gáz ideiglenes kibocsátásáért a labirintus, egy a játékos által kiválasztott szobájába. Feladata számontartani mennyi ideje bocsájtották ki és ezáltal mennyi ideig érvényes még.

Össztályok:

- Nincs specifikált osztály.

Interfészek:

- Megvalósítja a Tárgy interfacet

Össztályok:

- Kapcsolatban van a Mérgező gáz osztállyal és ennek köszönhetően a Szoba osztály számon tudja tartani mely szobákban van mérges gáz.

Attribútumok:

- meddigMérgez: int

Metódusok:

- **void használ():** A metódus beállítja a mérgezett flaget a szobában, amiben a tárgyat használták.

4.3.3 Ember

Felelősség:

- Az Ember osztály azonosítja a karaktert, valamint tárolja az esetleges ájulási idő és az általuk birtokolt tárgyakat. Képesek tárgyak felvételére, eldobására és használatára.

Össztályok:

- Nincs specifikált össztály.

Interfészek:

- Nincs specifikált interfész.

Asszociációk:

- Kapcsolatban áll a Szoba osztállyal, mivel minden szoba tárolja az esetlegesen benne tartózkodó embereket
- Kapcsolatban áll a Tárgy interface-el is hiszen, tárolhat/felvehet/letehet tárgyakat
- Kapcsolatban áll a Controller interface-el, hiszen ennek hatására mozognak az emberek

Attribútumok

- ájulásIdő: int
- fogottTárgy: Tárgy

Metódusok

- **void tárgyatFelvesz(Tárgy t):** Ez a metódus lehetővé teszi, hogy felvegyen egy tárgyat a szobából, ha van elég helye. Paraméterként megkapja azt a tárgyat, amit felvett az ember, és ez bekerül a hátizsák/zseb Tárgyakat tároló heterogén kollekcióba. Továbbá meghívja a Szoba tárgyKivesz(Tárgy t) metódusát és ezzel eltűnik onnan is a Tárgy.
- **void mozgás():** A controller egy jelére indul meg a mozgás, amit a Hallgató és az oktató felül definiál az általuk specifikált mozgásformára.
- **void ájulás(int i):** Metódus megkapja mennyi ideig lesz kiütve az ember és ennek megfelelően addig kiüti.
- **void tárgyatHasznál(Tárgy t):** Ez a metódus lehetővé teszi egy tárgy kézbe vételét, majd használatát. Paraméterként megkapja azt a tárgyat, amit használni szeretne és meghívja rajta az adott tárgy használ() metódusát. Oktató csak a maszkot használhatja.
- **void tárgyatEldob(Tárgy t):** Ez a metódus lehetővé teszi egy tárgy zsebből/hátizsákából való eldobását. Paraméterként megkapja azt a tárgyat, amit el szeretne dobni. Ennek hatására a tárgy kikerül a hátizsák/zseb Tárgyakat tároló heterogén kollekcióból. Továbbá meghívja a Szoba tárgyBeteszt(Tárgy t) metódusát, amivel bekerül a szobában lévő tárgyak közé a Tárgy.

4.3.4 Hallgató

Felelősség:

- A Hallgató osztály felelős a játékos karakterének kezeléséért a labirintusban. Ez az osztály rögzíti és kezeli a hallgatóval kapcsolatos információkat, a felvett tárgyak listáját. Emellett felelős a hallgató cselekedeteinek végrehajtásáért, beleértve a tárgyak felvételét és letételét, valamint a szobák közötti mozgást. A Hallgató osztály biztosítja a felhasználói interakciót a játék során.

Ősosztályok

- Ember -> Hallgató

Interfészek

- Nincs specifikált interfész.

Asszociációk

- Amikor a Hallgatót mozgatják a labirintusban, az kapcsolatba léphet az Ajtó osztállyal, hogy áthaladhasson a különböző szobák között.
- A Hallgató osztály kapcsolatban áll az Oktató osztállyal a labirintusban való interakciók során hiszen, ha találkoznak akkor az Oktató kiszívja a lelkét.
- A Hallgató kapcsolatban áll a JátékosÍrányítottStratégia-val hiszen e szerint mozog a Hallgató a labirintusban

Attribútumok

- hátizsák: Tárgy[0..5]
- stratégia: JátékosÍrányítottStratégia
- lélek: bool

Metódusok

- **void Mozgás():** A controller egy jelére indul meg a mozgás, a felhasználó által kijelölt irányba a JátékosÍrányítottStratégia szerint.
- **void SetLélek(bool b):** Amennyiben az oktató elkapja a Hallgatót elveszti a lelkét. Itt paraméterként megkapja a hallgató aktuális lélekállapotát és beállítja azt. Ha a lélek el lett véve akkor a Hallgató kibukik és vége a játéknak.

4.3.5 JátékoslrányítottStratégia

Felelősség

- Az osztály felelős a játékos által bevitt adatok értelmezéséért, amiket a Hallgató mozgásához kell felhasználni

Össztályok

- Nincs specifikált össztály.

Interfészek

- Megvalósítja a MozgásStratégia interfacet

Asszociációk

- Kapcsolatban áll a Hallgató osztállyal

Attribútumok

-

Metódusok

- void végrehajt()

4.3.6 Labirintus

Felelősség:

- A Labirintus osztály felelős a játékterület létrehozásáért és kezeléséért. Ez az osztály tartalmazza a labirintusban található szobákat és irányítja azok osztódását és egyesülését. A Labirintus osztály feladata a szobák hálózatának kezelése. Emellett a Labirintus osztály gondoskodik a játékalapot frissítéséről és ellenőrzéséről, figyelembe véve az esetleges változásokat, például új szobák megjelenését.

Ősosztályok

- Nincs specifikált ősosztály.

Interfészek

- Nincs specifikált interfész.

Asszociációk

- A Labirintus osztály tartalmazza a Szoba objektumokat, mivel a labirintus fő elemei a szobák, amelyekből áll. Az asszociáció lehetővé teszi a labirintus osztálynak, hogy kezelje és irányítsa a szobákat a játék során.

Attribútumok

- Szobák: List<Szoba>
- példány: Labirintus

Metódusok

- **void szobaÖsszevon()** - Ez a metódus két véletlenszerűen kiválasztott szobát von össze a labirintusban. A metódus végrehajtásával a két szoba egyesül, és a labirintusban egy új, nagyobb szoba jön létre, amely megőröklí a két kiválasztott szoba tulajdonságát és szomszédjait. Az eredeti két szobában tartózkodó emberek és tárgyak az új szobába kerülnek be, a szobák törlésre kerülnek, illetve az újonnan létrejött nagyobb szoba bekerül a szobák listájába.
- **void szobaFeloszt()** - Ez a metódus egy véletlenszerűen kiválasztott szobát két szobára oszt a labirintusban. Az új szobák megosztóznak az eredeti szoba tulajdonságaiban és szomszédságaiban. Az eredeti szobában lévő embereket és tárgyakat szintén szétosztja, majd törli az eredeti szobát a labirintusból.
- **Labirintus getPéldány()** - Mivel az osztály a Singleton pattern-t követi, ezért ez a metódus létrehozza a Labirintust, ha még nem létezik, amit a példány változóba ment el. Futás végén a Labirintus egyetlen példányával tér vissza.

4.3.7 Logarléc

Felelősség

- A Logarléc osztály felelős a játék céljának reprezentálásáért. A játékosok feladata a Logarléc felkutatása és megszerzése a labirintusban annak érdekében, hogy nyerjenek.

Ősosztályok

- Nincs specifikált ősosztály.

Interfészek

- Megvalósítja a Tárgy interfacet

Asszociációk

-

Attribútumok

-

Metódusok

- **használ():** leállítja a játékot, meghívja a Controlleren a játék végét kezelő metódust

4.3.8 Maszk

Felelősség:

- A Maszk osztály felelős a játékosok védelméért a mérgező gázokkal teli szobákban. A maszk egy adott ideig tudja csak megakadályozni a mérges gázok belélegzését.

Ősosztályok**Interfészek:**

- Megvalósítja a Tárgy interfacet

Asszociációk**Attribútumok**

- védIdő: int

Metódusok

- **használ():** A metódus aktiválásakor a maszk elkezd a védelmi funkcióját, és az Embert megvédi, az elájulástól, vagyis az ájulásIdőt 0-ra állítja.

4.3.9 Mérgező gáz

Felelősség:

- A Mérgező gáz osztály felelős a mérgező gáz jelenlétének reprezentálásáért a labirintus egyes szobáiban. Hatására a benne tartózkodók eszméletüket veszthetik és az összes náluk található tárgy automatikusan eldobódik.

Össztályok

- Nincs specifikált össztály.

Interfészek

- Nincs specifikált interfész.

Asszociációk

- A Mérgező gáz kapcsolatban van a Szoba osztállyal, mivel annak jelenléte a szobában hatással van a szoba állapotára és tulajdonságaira. A szoba tartalmazza vagy kezeli a mérgező gáz állapotát és jelenlétét.
- Kapcsolatban van a Camembert osztállyal, hiszen az létrehozhatja egy szobában

Attribútumok

- HátralévőIdő: int

Metódusok

- **void Kiüt():** arra szolgál, hogy a gáz hatására azon személyeket, akik a szobában tartózkodnak és nincsen rajtuk maszk azokat elkábítsa. Meghívja az emberek Ájulás(int i) függvényét, egy meghatározott értékkel, valamint a tárgyatEldob(Tárgy t) függvényüket, az összes náluk lévő tárggyal.

4.3.10 MozgásStratégia

Felelősség

- Az interface felelős az Emberek mozgásának irányításáért.

Össztályok

- Nincs specifikált össztály.

Asszociációk

- Nincs specifikált interfész

Metódusok

- **void végrehajt():** az interface ezen metódusát implementálja a JátékosÍrányítottStratégia és a SzámítógépÍrányítottStratégia.

4.3.11 Oktató

Felelősség:

- Az Oktató osztály felelős a nem játékos karakterek kezeléséért a labirintusban. Ez az osztály felelős azért, hogy megnehezítse a játékos dolgát a játék során.

Ősosztályok

- Ember -> Oktató

Interfészek

- Nincs specifikált interfész.

Asszociációk

- Amikor az oktató mozog a labirintusban, az kapcsolatba léphet az Ajtó osztállyal, hogy áthaladhasson a különböző szobák között.
- Az Oktató osztály kapcsolatban áll a Hallgató osztállyal a labirintusban való interakciók során hiszen, ha találkoznak akkor az Oktató elveszi a Hallgató lelkét.

Attribútumok

- zseb: Tárgy
- stratégia: SzámítógépIrányítottStratégia

Metódusok

- **void támad(Hallgató h):** Megtámadja az argumentumként kapott hallgatót, amennyiben a hallgató nem rendelkezik védelmi tárggyal, akkor a lélek attribútuma hamisra állítódik a setLélek() metódussal és a hallgató számára a játéknak vége.
- **void mozgás():** A controller egy jelére indul meg a mozgás, a felhasználó által kijelölt irányba a SzámítógépIrányítottStratégia szerint.

4.3.12 Pohár

Felelősség

- A Pohár osztály felelős azért, hogy tárolja és kezelje a söröspohár tulajdonságait és funkcióit a labirintusban, mint például használható-e a tárgy védelem nyújtására az oktatókkal szemben.

Össztályok

- Nincs specifikált össztály.

Interfészek

- Megvalósítja a Tárgy interfacet

Asszociációk

- Kapcsolatban áll a Tárgy interface-el mivel implementálja azt

Attribútumok

- védIdő: int

Metódusok

- **void használ():** Ezt a metódust aktiválva a hallgató védelmet kap az oktatókkal szemben.

4.3.13 Rongy

Felelősség

- Az osztály felelős a nedves táblatörlő rongy kezeléséért a játék során. A tárgyat fel lehet venni szobákban, ahol található és el lehet őket használni védekezésre oktatók ellen.

Össztályok

- Nincs specifikált osztály.

Interfészek

- Megvalósítja a Tárgy interfacet

Asszociációk

-

Attribútumok

- meddigNedves: int

Metódusok

- **void használ():** Ez a metódus megvalósítja a tárgy hatását, amely megbénítja azokat az oktatókat, akik a szobában tartózkodnak vele. Meghívja az oktatók Ájulás(int i) függvényét, egy meghatározott értékkel.

4.3.14 SzámítógéplányítottStratégia

Felelősség

- Felelős az Oktatók mozgásának irányításáért

Össztályok

- Nincs specifikált osztály.

Interfészek

- Megvalósítja a MozgásStratégia interfacet

Asszociációk

- Kapcsolatban áll az Oktató osztállyal

Attribútumok

-

Metódusok

- **void végrehajt():** irányítja az oktató mozgását egy bizonyos stratégia szerint.

4.3.15 Szoba

Felelősség

- Számon tartja a benne tartózkodó hallgatókat, oktatókat és tárgyakat, valamint a szobának a tulajdonágait tárolja

Ősosztályok

- Nincs specifikált ősosztály.

Interfészek

- Nincs specifikált interfész.

Asszociációk

- Kapcsolatban az Ember osztállyal, hiszen tartózkodhatnak benne
- Kapcsolatban van a Tárgy osztállyal, a benne található tárgyakat tárolja
- Kapcsolatban van az Ajtó osztállyal is hiszen tárolja a sajátjait.
- Kapcsolatban van a Mérgező gázzal is hiszen a gáz hatására lesz a szoba mérgezett.

Attribútumok

- férőhely: int
- elátkozott: bool
- mérgezett: bool
- földönHever: List<Tárgy>
- bentVan: List<Ember>
- ajtók: List<Ajtó>

Metódusok

- **bool befér():** Visszaadja, hogy befér-e még egy ember a szobába, tehát hogy a bentlévők száma kisebb-e a férőhelyek számát.
- **void emberBejön(Ember e):** Az Embert, aki bemegy a szobába hozzáadja a bentVan listához, valamint átállítja az emberben azt a szobát, ahol éppen tartózkodik.
- **void emberKimegy(Ember e):** Az Embert, aki kimegy a szobából kitörli a bentVan listából.
- **void tárgyBeteszt(Tárgy t):** Ha egy Ember eldob egy tárgyat ezt a függvényt hívja, hogy az átkerüljön a szoba FöldönHever listájába.
- **void tárgyKiveszt(Tárgy t):** Ha egy Ember felvesz egy tárgyat, a szobában ezt a függvényt hívja, hogy az a Tárgy kikerüljön a Szoba FöldönHever listájából.
- **List<Szoba> getSomszédok():** Visszaadja a Szobával szomszédos Szobák listáját

4.3.16 Tárgy

Felelősség

- A Tárgy osztály felelős a tárgyak használatát biztosító interface megvalósításáért

Ősosztályok

- Nincs specifikált őszosztály.

Interfészek

- Nincs specifikált interfész

Asszociációk

- Kapcsolatban van a Szoba osztállyal, mivel ez tárolhat Tárgy típusú objektumokat
- Kapcsolatban van az Ember osztállyal, mivel ez is tárolhat ilyen objektumokat

Attribútumok

-

Metódusok

- **void használ():** ezt a metódust írják felül az interfacet megvalósító osztályok

4.3.17 Tranzisztor

Felelősség

- A Tranzisztor osztály felelős a játékosok általi mozgás segítéséért. Segítségével a játékosok képesek összekapcsolni két különböző szobát, hogy gyorsabban mozoghassanak a labirintusban.

Ősosztályok

- Nincs specifikált ősosztály.

Interfészek

- Megvalósítja a Tárgy interfacet

Asszociációk

- Kapcsolatban van még a Szoba osztállyal

Attribútumok

- pár: Tranzisztor
- ittEldobva : Szoba

Metódusok

- **void párosít():** Ez a metódus párosítja a tranzisztorokat, azaz összekapcsolja a tranzisztort egy másik tranzisztorral, tehát ha van egy tranzisztor a Hallgató kezében(fogottTárgy) és egy a Hátizsákjában, akkor azok össze lesznek párosítva, és a Pár attribútumban tárolják egymást.
- **void használ():** Ez a metódus lehetővé teszi a játékosok számára, hogy az egyik tranzisztort használva átteleportálják magukat egy másik tranzisztorral összekapcsolt szobába, ehhez a PárSzobája() függvényt használja.
- **Szoba párSzobája():** visszaadja a Szobát, ahol a tranzisztor párja el lett helyezve

4.3.18 TVSZ

Felelősség

- Ez az osztály felelős a TVSZ kezeléséért és tulajdonságaiért a játékban, védelem nyújtása és annak fentállásának ellenőrzésért

Össztályok

- Nincs specifikált ösztály

Interfészek

- Megvalósítja a Tárgy interfacet

Asszociációk

-

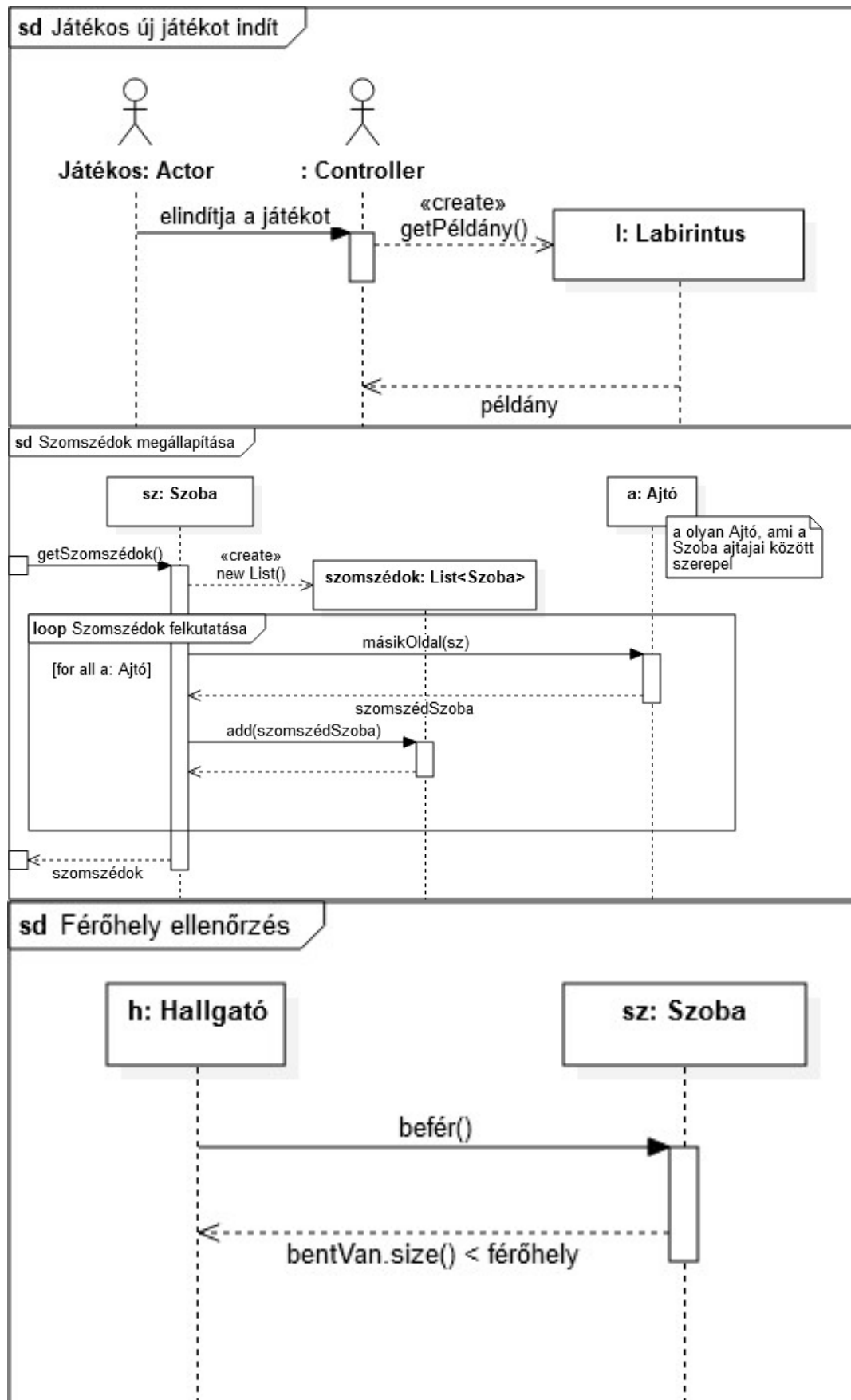
Attribútumok

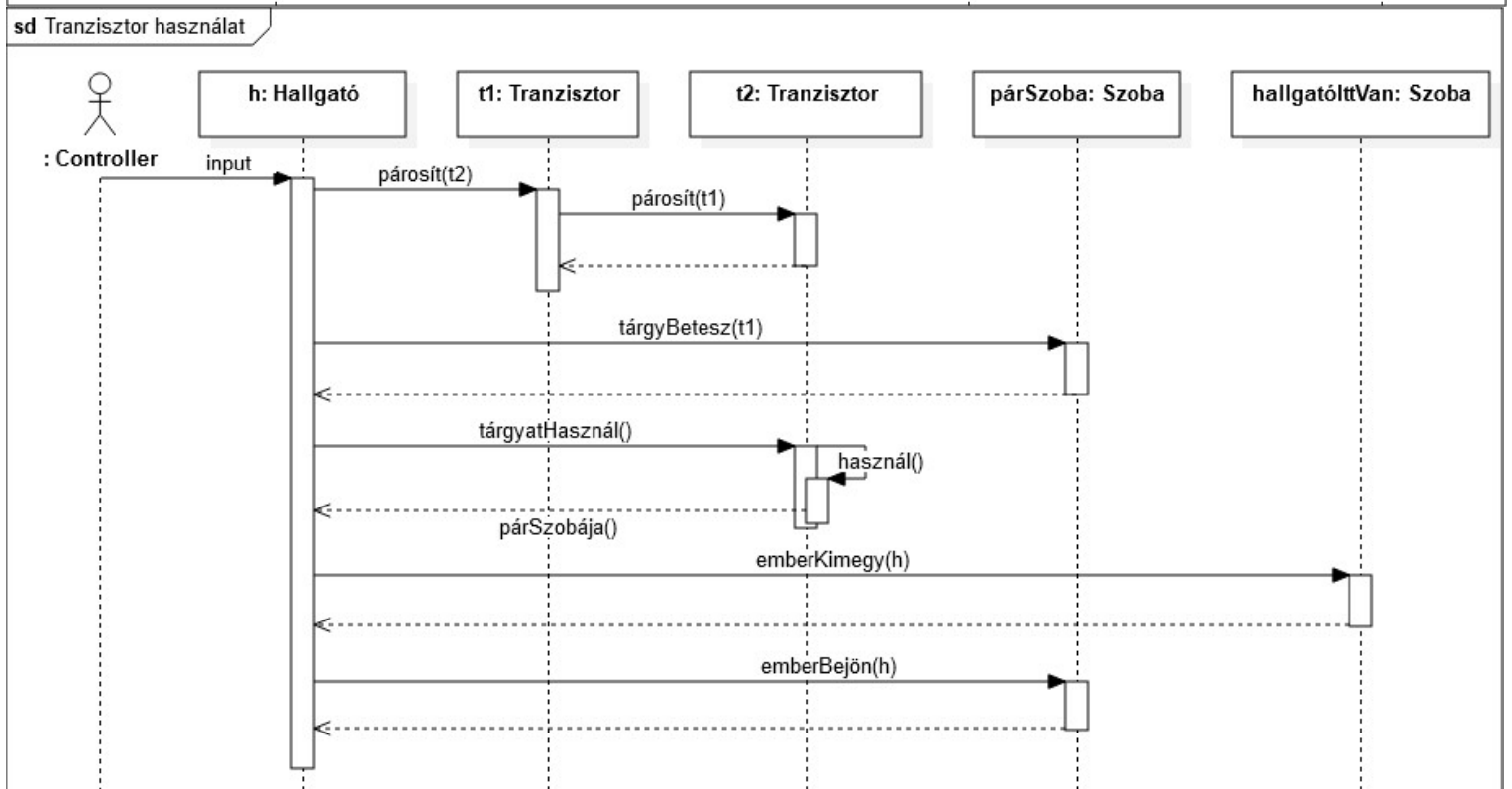
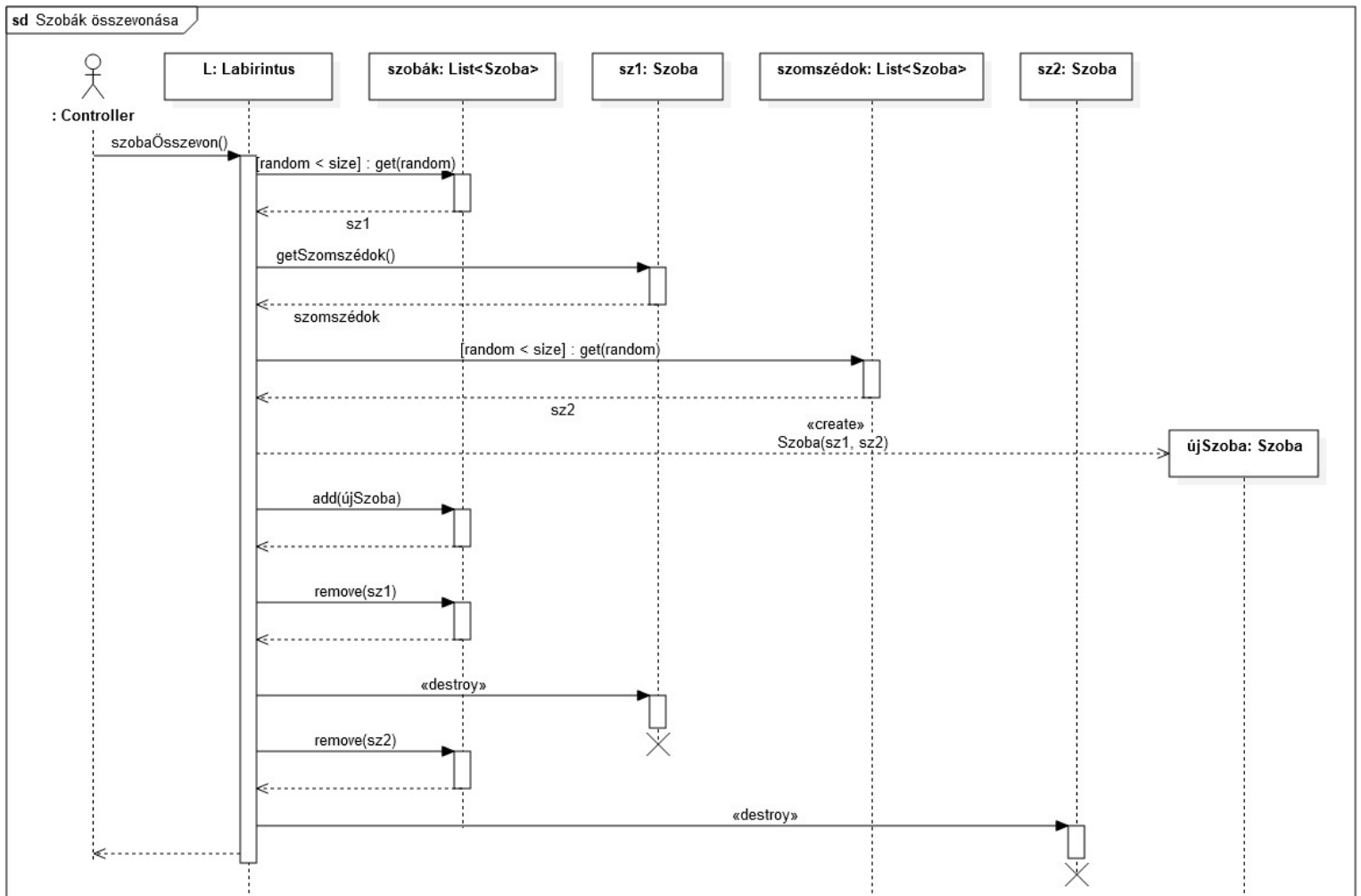
- védelmekSzama: int

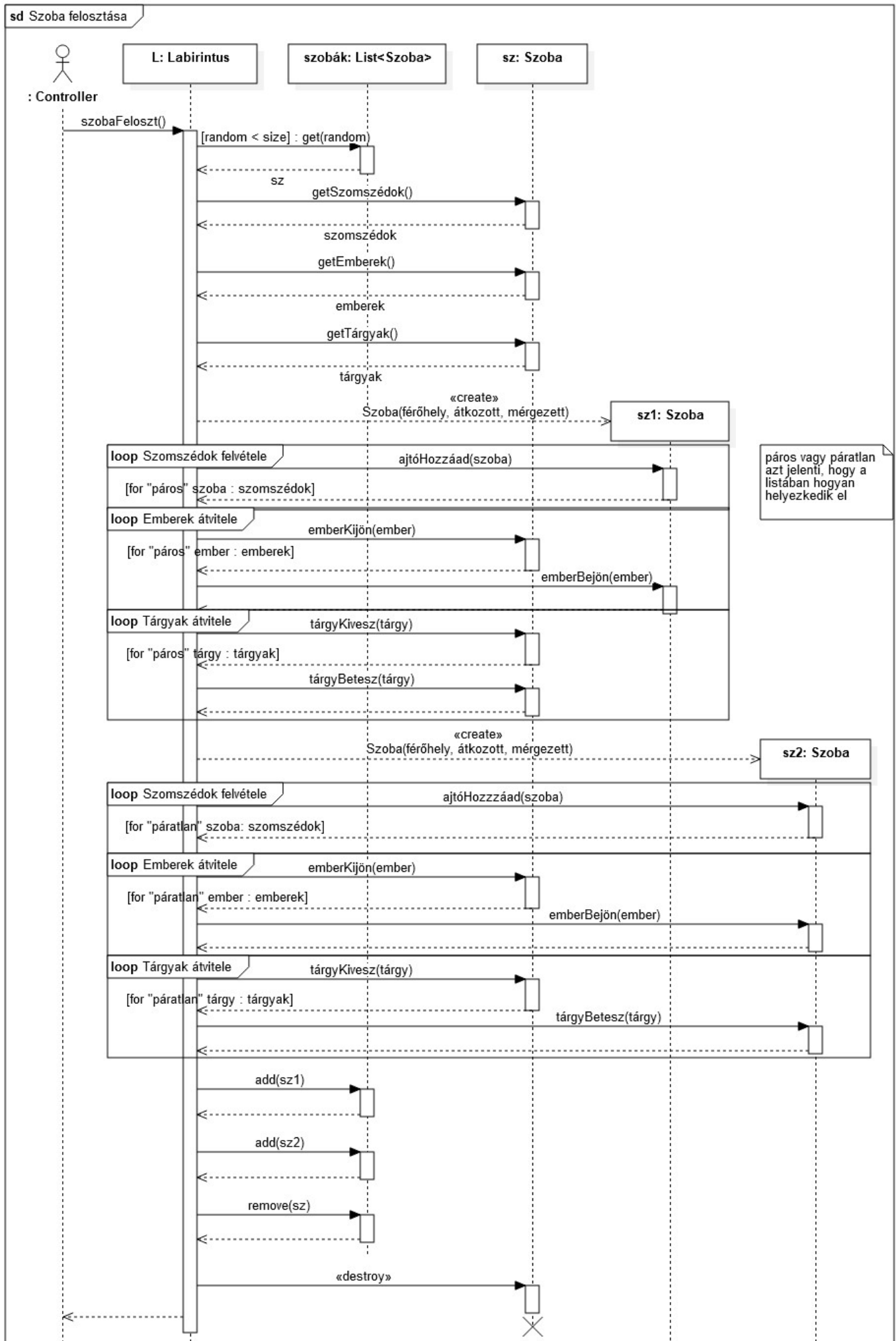
Metódusok

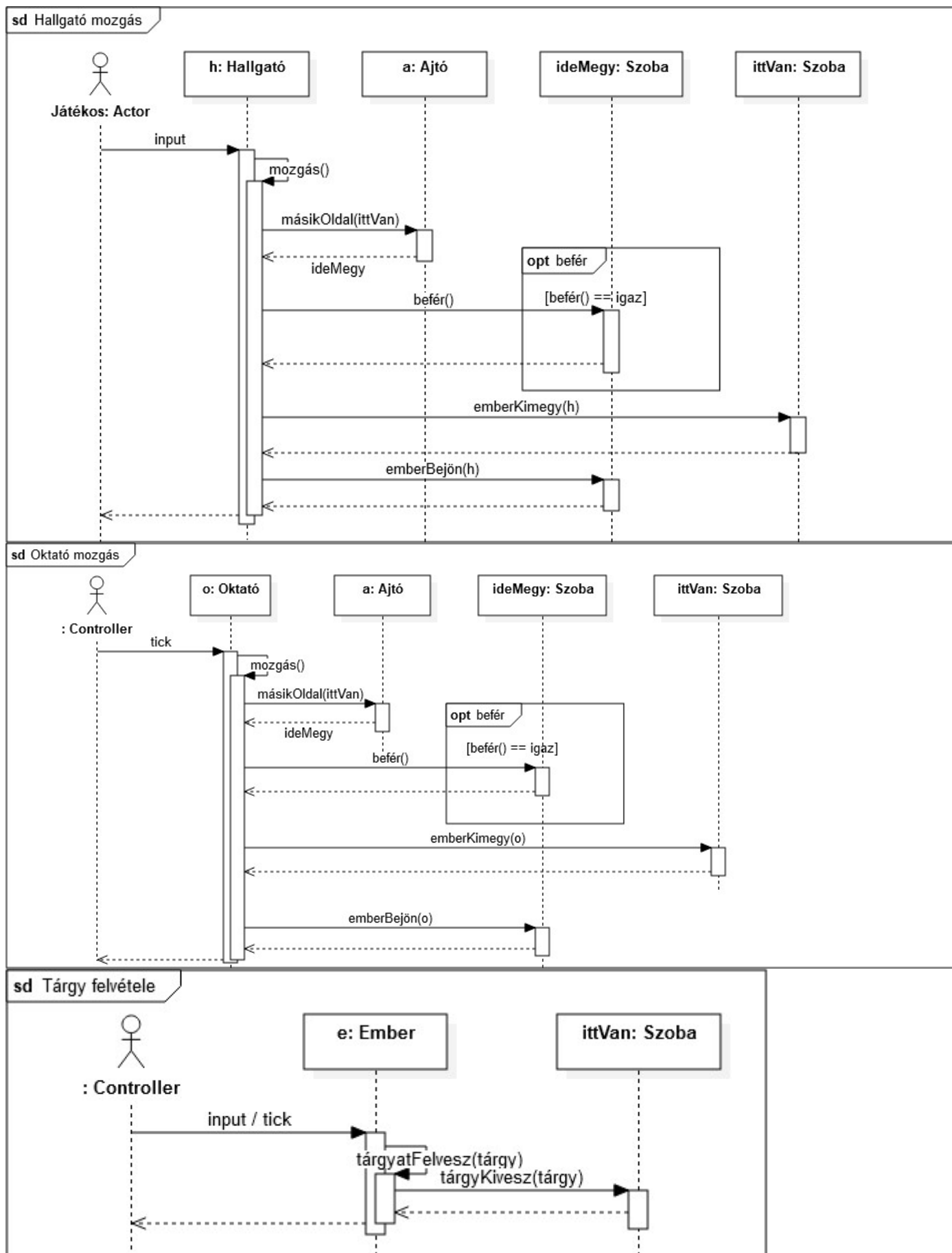
- **void használ():** Ezt a metódust aktiválva a hallgató védelmet kap az oktatókkal szemben, ezzel tud védekezni az Oktató Támad() metódusa ellen, ami ebben az esetben nem lesz sikeres és csökkenti a védelmekSzámát.

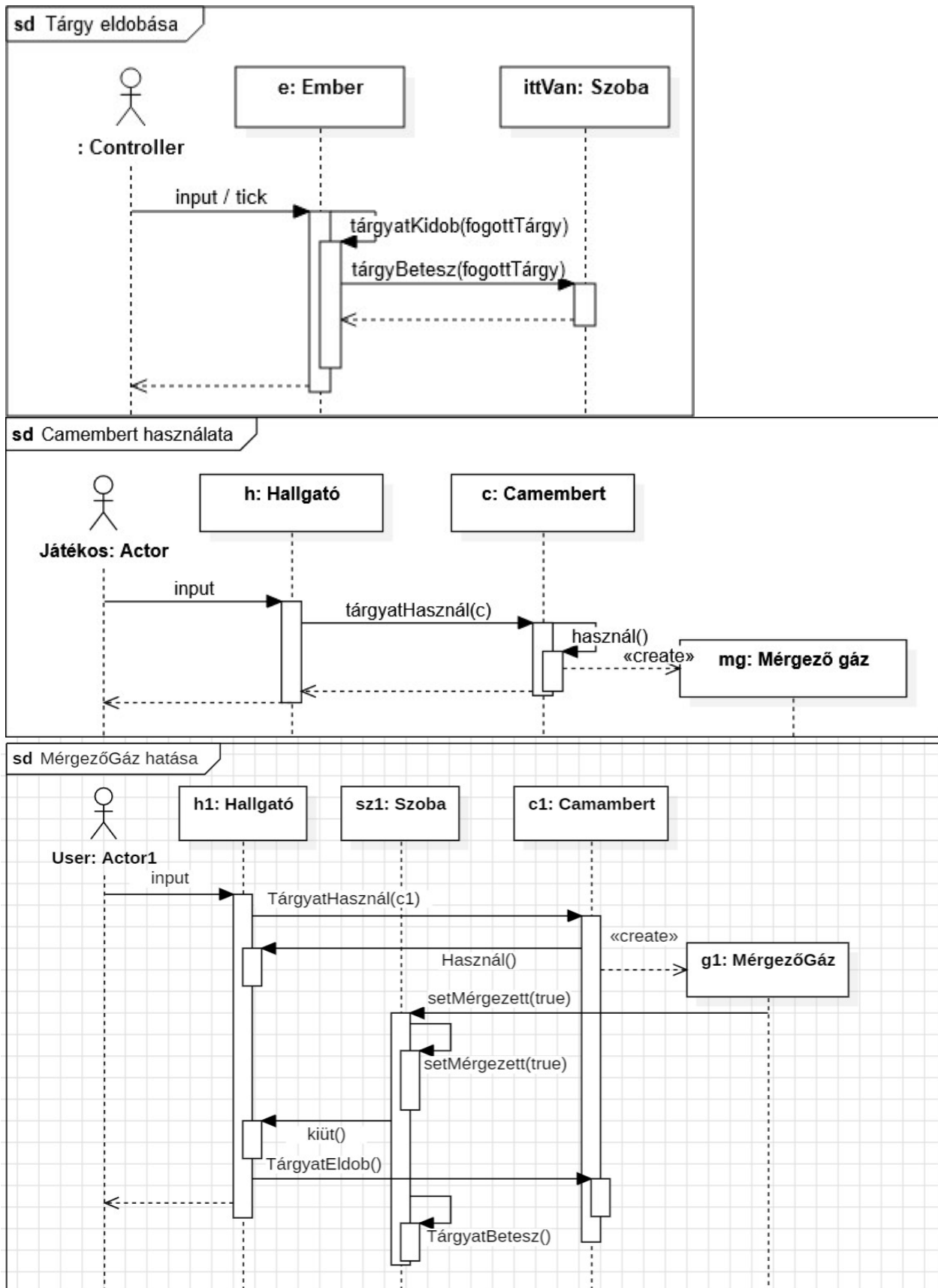
4.4 Szekvencia diagramok



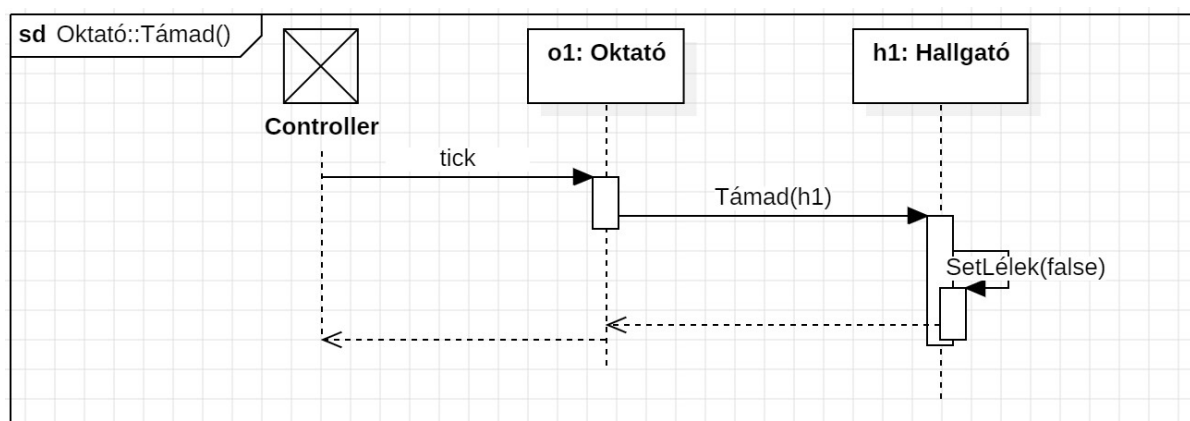
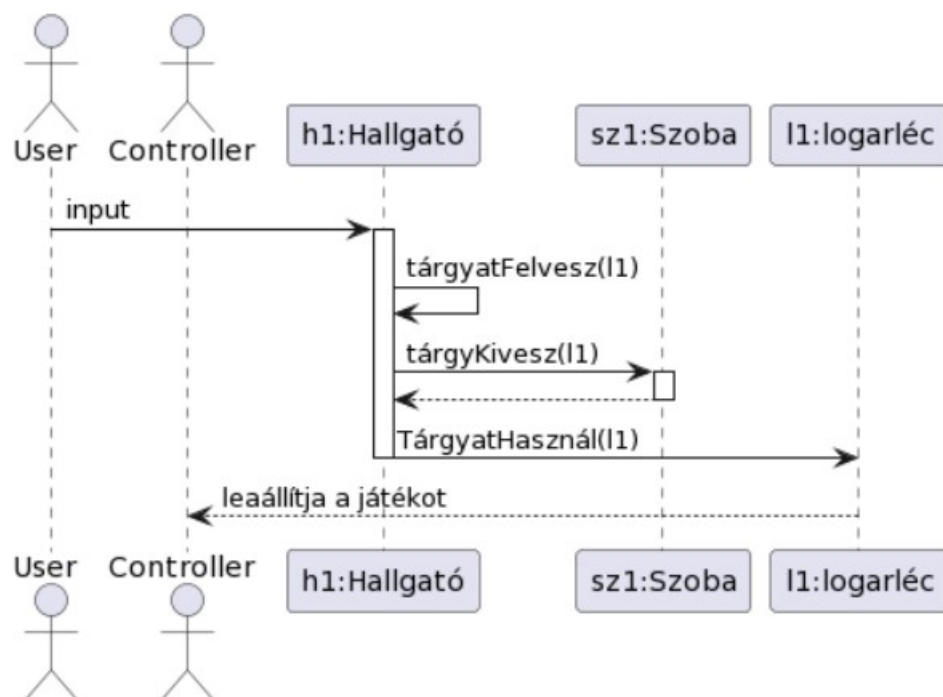




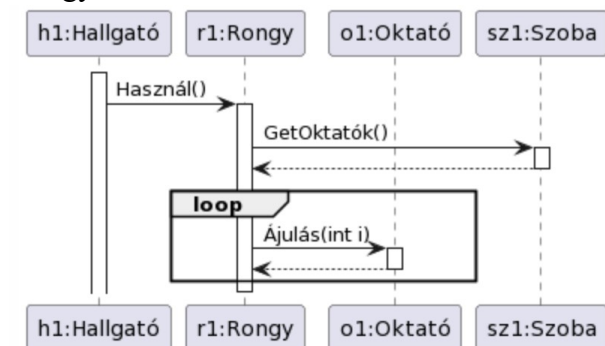




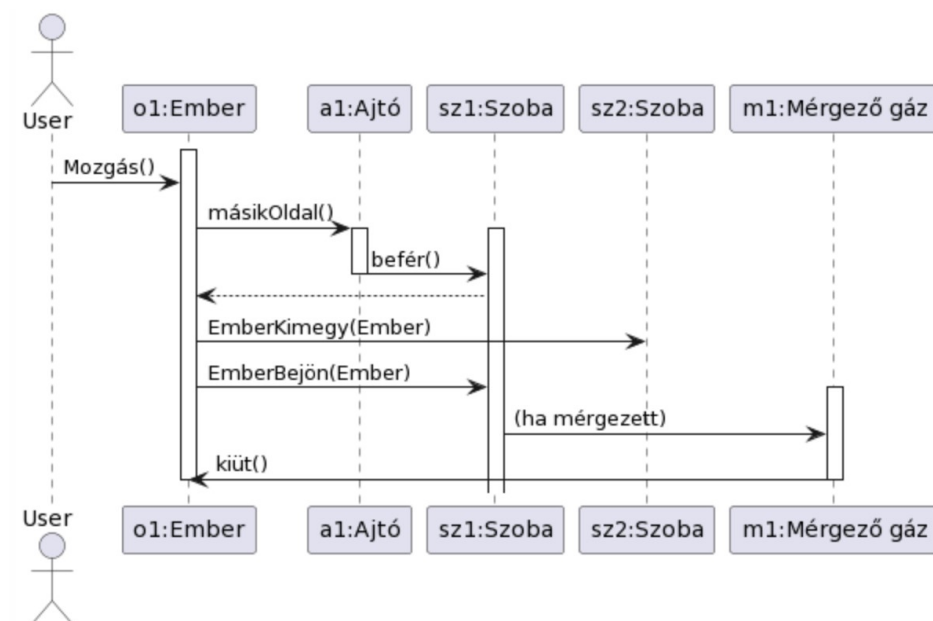
Logarléc működése:



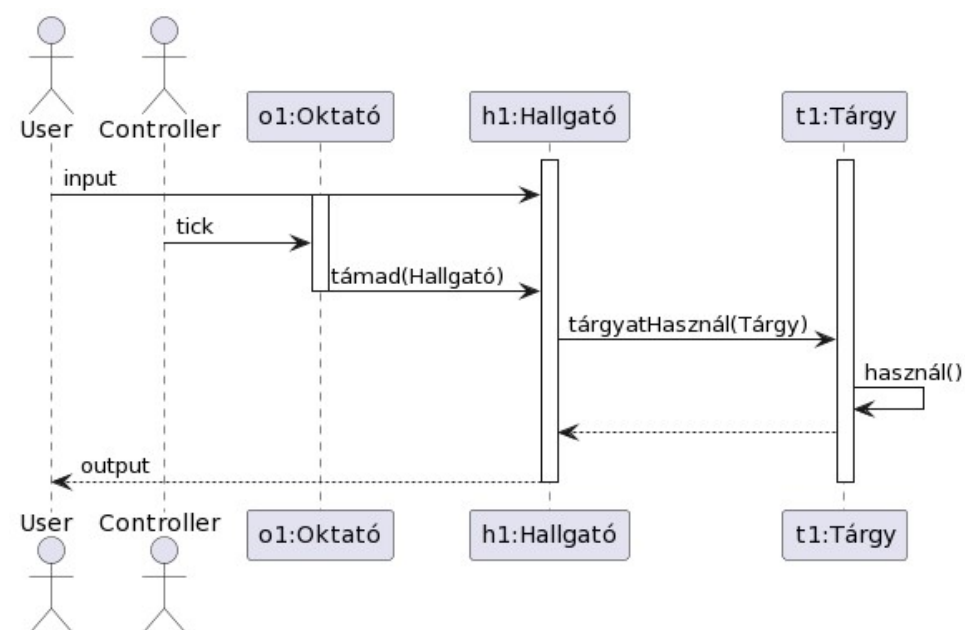
Rongy használ metódusa:



Mérgező gáz hatása:

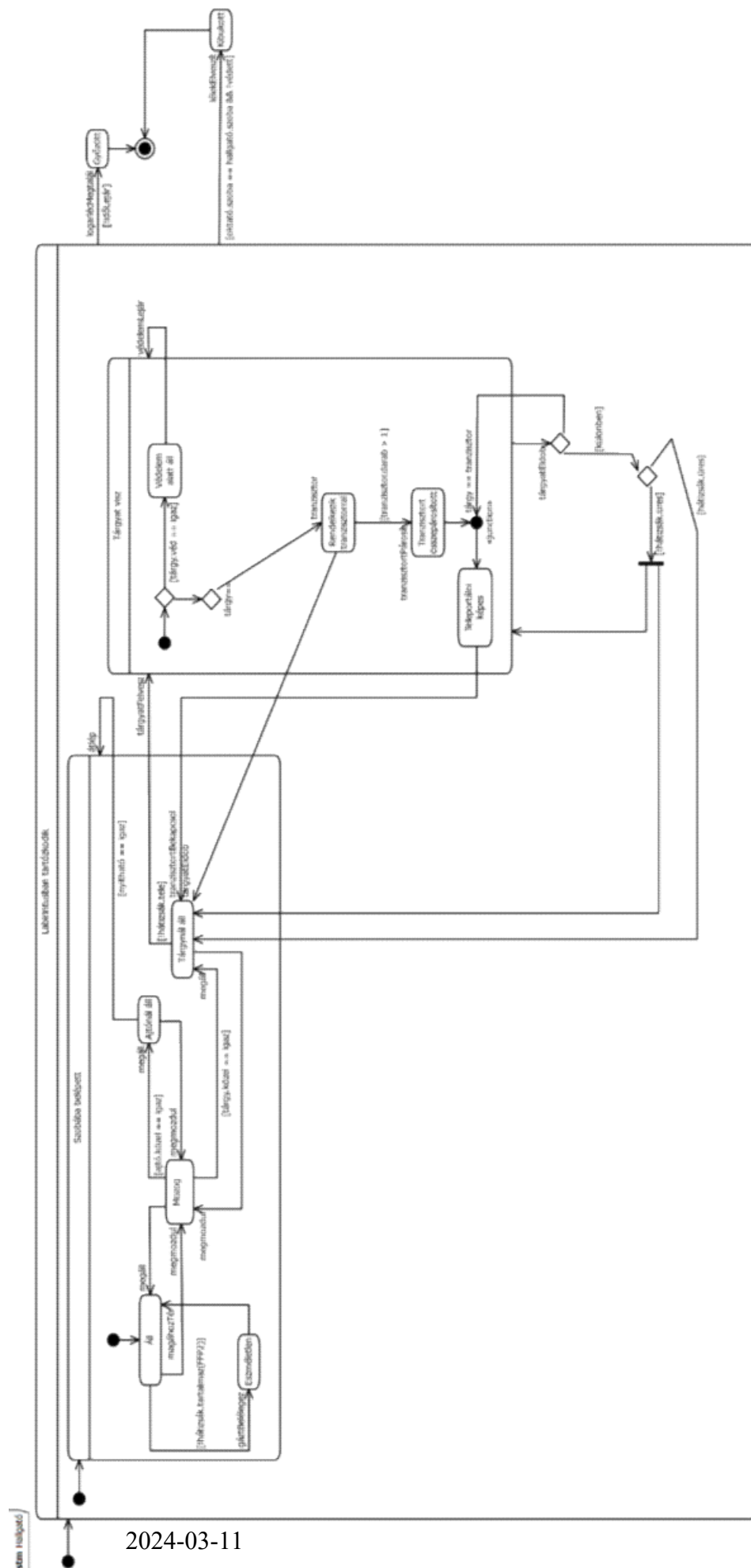


Oktatói támadás elleni védekezés:



~~2024-03-11~~





4.6 Napló

Kezdet	Időtartam	Résztevők	Leírás
2024.03.07. 16:00	2 óra	Ábel Bakó Csala Lévai	Tevékenység: UML osztálydiagram átdolgozása
2024.03.08. 16:00	2.5 óra	Ábel Bakó Csala Lévai	Tevékenység: UML osztálydiagram átdolgozása
2024.03.09. 20:00	2 óra	Csala	Szekvencia diagramok készítése
2024.03.08. 19:00	2 óra	Melegh	Változtatások átnézése, további változtatások megtervezése
2024.03.09. 11:00	12 óra	Bakó	Osztályleírások, szekvencia diagrammok, class diagram szerkesztés
2024.03.10. 12:00	9 óra	Lévai	Osztályleírások, szekvencia diagrammok
2024.03.10 11:00	14 óra	Ábel	UML szekvencia diagramok és osztálydiagram készítése, javítása
2024.03.10. 11:00	2,5 óra	Bakó Csala Lévai Melegh	Tevékenység: Hibák javítása
2024.03.10. 15:00	3 óra	Ábel Bakó Csala Lévai Melegh	Tevékenység: Hibák javítása
2024.03.10. 11:00	10 óra	Csala	Szekvencia diagramok
2024.03.11 9:00	2 óra	Ábel	Dokumentum átnézése, javítása, törlése