

Egyesített dokumentáció

35 – pupakok

Konzulens:
Koczó Attila

Csapattagok:

Tarsoly Levente	E1IK75	tarsoly1@edu.bme.hu
Németh Gergely	TA9G0R	gergo.h.nemeth@gmail.com
Schulcz Gábor	EP3I9R	schulcz.gabor6@gmail.com
Czotter Benedek	TFB4FY	czottibeni@gmail.com
Hermann Máté Béla	T83K5I	mate3829@gmail.com

2024.05.24

Követelmények, projekt funkcionalitás

35 – pupakok

Konzulens:
Koczó Attila

Csapattagok:

Tarsoly Levente	E1IK75	tarsolyl@edu.bme.hu
Németh Gergely	TA9G0R	gergo.h.nemeth@gmail.com
Schulcz Gábor	EP3I9R	schulcz.gabor6@gmail.com
Czotter Benedek	TFB4FY	czottibeni@gmail.com
Hermann Máté Béla	T83K5I	mate3829@gmail.com

2024.03.03.

2. Követelmény, projekt, funkcionalitás

2.1 Bevezetés

2.1.1 Cél

A dokumentum célja, hogy bemutassa és összefoglalja a “pupakok” csapat által készített implementációját a “Logarléc” feladatnak. A dokumentum betekintést nyújt továbbá a csapatunk gondolat menetébe, hogy hogyan is értelmeztük a feladatunkat és hogyan próbáltuk a lehető legjobban megvalósítani azt.

2.1.2 Szakterület

Egy szórakoztató célú számítógépes játék elkészítése, amely tökéletes a szabadidő eltöltésére és amely a lehető legjobban megfelel a megadott specifikációknak.

2.1.3 Definíciók, rövidítések

TVSZ - Tanulmányi és Vizsgaszabályzat (Budapesti Műszaki és Gazdaságtudományi Egyetem)

Műegyetem - Budapesti Műszaki és Gazdaságtudományi Egyetem

PL. - például

FFP - Filtering Face Piece

2.1.4 Hivatkozások

<https://www.iit.bme.hu/file/11582/feladat>

<https://www.iit.bme.hu/file/13619/2-k%C3%B6vetelm%C3%A9ny-projekt-funkcionalit%C3%A1s>

<https://gaphor.org/> (use case diagram elkészítéséhez)

<https://github.com/>

<https://docs.google.com/>

2.1.5 Összefoglalás

Az előző pontokban az elkészítendő feladatunknak az általános tudnivalóit ismertettük. A továbbiakban kicsit mélyebben, nagyobb részletességgel mutatjuk meg a feladattal járó sajátosságokat és egyéb más részleteket.

2.2 Áttekintés

2.2.1 Általános áttekintés

A megvalósítandó feladatunk lényege egy társasjátékhoz hasonló, számítógépes játék elkészítése. A játékhoz elegendő már csupán 1 játékos, de többen is játszhatnak egyszerre. Továbbá, a játék alapját különböző helyiségek, szobák fogják alkotni, amelyeken keresztül navigálva különböző interaktív tárgyakkal, vagy éppen ellenségesen viselkedő egyéb karakterekkel találhatjuk szembe magunkat. minden játékos, függetlenül a számuktól, egy csapatban, együttes dolgozik a végső győzelem elérésében. A játéknak vége lesz és a játékosok nyernek, ha megszerzik a "Logarléc" tárgyat, ellenben veszítenek, ha minden játékos egy ellenséges karakter áldozata lesz. Egy felhasználó mindenféleképpen csak egy játékbeli karakter irányításáról gondoskodik a játék során.

2.2.2 Funkciók

A Műegyetem Központi épületének alagsora alatt egy elátkozott labirintus rejtozik. A mérnökhallgatók dolga fellelni a Logarléc nevű mágikus képességű ereklyét. A labirintus szobáit ajtók választják el egymástól, ezeken átlépve lehet az egyik szobából a másikba átutni. Egy-egy szobából legalább egy, de esetenként sok másik szobába is nyílhat ajtó. Vannak ráadásul ajtók, amelyek csak egy irányban használhatók.

A szobákban különféle tárgyak lehetnek (ilyen a Logarléc is), amiket a hallgatók magukhoz vehetnek, de egy hallgatónál egy időben legfeljebb öt tárgy lehet. A tárgyakat a hallgatók le is tudják tenni.

A labirintusban oktatók próbálják megakadályozni a hallgatókat abban, hogy sikkerrel járjanak. Ha egy oktató egy szobába kerül egy vagy több hallgatóval, akkor elveszi a lelkét és a hallgató kibukik az egyetemről.

Abban az esetben, ha az adott hallgató "kibukik az egyetemről", számára a játék véget ér, valamint minden addig megszerzett tárgya megszűnik létezni. A tárgyak között azonban vannak olyanok, amik adott ideig védettséget nyújtanak az oktatók ellen (pl. a TVSZ denevérbőrre nyomtatott példányai három alkalommal mentik meg a hallgató életét, utána elveszítik a varázserejüket, a szent söröspoharak pedig csak adott ideig hatnak). Van olyan tárgy is, a nedves táblatörő rongy, amely adott ideig működik (amíg ki nem szárad), és a vele egy szobában lévő oktatókat megbénítja. A dobozolt káposztás camembert felbontáskor mérges gázt bocsát ki (lásd lejjebb a gázos szobákat). A tárgyakat az oktatók is fel tudják venni! **Az oktatók bármennyi tárgyat fel tudnak venni, a Logarlécen kívül. Az oktatók ezen kívül nem dobhatnak el tárgyakat és nem lehetnek le tárgyat (PL: A tranzisztor esetében).**

A szobákban elvétve tranzisztorok is találhatók. A hallgatónál levő tranzisztorokat páronként össze lehet kapcsolni, majd a pár egyik tagját menet közben egy másik szobában le lehet tenni. Az így összekapcsolt tranzisztorok varázserővel bírnak: ha a hallgató a nála maradó tranzisztort bekapsolja és leteszi, akkor a másik tranzisztor szobájába kerül, a bekapsolt tranzisztor pedig kikapcsol. A tranzisztorok korlátlan ideig használhatók. **Egy hallgatónál egyidejűleg maximum két tranzisztor lehet. Mindezek mellett, ha a hallgató már birtokol egy aktivált tranzisztort, akkor nem vehet magához többet.**

Minden szobának van egy (a szobára jellemző) befogadóképessége. Ennél több hallgató és oktató 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 vesztik (nem játszhat egy ideig)** és a náluk lévő tárgyakat **elejtik**. Ha valakinél van FFP2-es maszk, akkor ezekben a szobákban adott időre (**játszhat tovább, de egy maszk maximum kétszer nyújt védelmet**) védettséget kap, de a maszk egyre rövidebb ideig képes a védelem nyújtására. Vannak olyan elátkozott szobák, amiknek az ajtajai időnként eltűnnék, majd később újra előtűnnék.

A szobák egy korábbi (félresikerült) gráfelméleti tételelbizonyítás eredményeként meghazudtolják a fizika törvényeit: képesek egyesülni és osztódni. Két szomszédos szoba egyesülésével létrejövő szoba a korábbi két szoba tulajdonságaival és szomszédaival rendeletelezik, de a befogadóképessége a nagyobb szoba befogadóképességevel lesz azonos. Az osztódó szoba két olyan szobára válik szét, amelyek egymás

szomszédai lesznek, és megosztognak a korábbi szoba képességein és szomszédain (a korábbi szomszédon vagy csak az egyik, vagy csak a másik “új” szobának lesznek szomszédai). ***Osztódnak csak előtte már egyesült szobák osztódhannak.***

A játékot egyszerre több játékos játssza (***felváltva, egymás után lépkedve***), akik a hallgatókat irányítják, és akkor nyernek, ha megadott időn belül megtalálták és magukhoz vették a Logarlécet.

2.2.3 Felhasználók

A felhasználónak semmilyen előzetes tudásra nincsen szüksége a játékhoz, csupán annak megfelelő futását kell tudnia biztosítani. A felhasználók a játék során csak a “hallgató” karaktereket irányíthatják.

2.2.4 Korlátozások

A játék fő korlátozása, hogy a kiadott feladatban foglaltak szerint működjön, adottak legyenek a futtatásához szükséges feltételek és minden esetben stabilan fusson a játék.

2.2.5 Feltételezések, kapcsolatok

A feladatunk leírásában olykor előfordult, hogy nem minden részlet volt teljes mértékben meghatározva, ezeket saját magunktól, az elképzéléseink alapján kiegészítettük, illetve kifejtettük. A csapat a kapcsolattartásra különböző közösségi média felületeket használ a személyes találkozások mellett, valamint a projekt későbbi fázisaira létrehoztunk egy Github felületet is, amely segítségével többek között egyszerűbb lesz a csapattagok munkáját összehangolni. Továbbá a későbbiekben látható Use-case diagramot a Graphor segítségével állítottuk össze, valamint a Google saját szövegszerkesztőjét, a Google Docs-ot használtuk, hogy akár többen is tudjunk egyszerre dolgozni a projektünk dokumentálásán.

2.3 Követelmények

2.3.1 Funkcionális követelmények

Azono-sító	Leírás	Ellenőrzés	Priori-tás	Forrás	Use-case	Kom-ment
F01	Létezik a pálya	Létrejön a pálya egésze.	alapvető	feladatkiírás	Pálya generálása	
F02	A pálya kisebb egységekből, szobákból áll	Létrejönnek a szobák	alapvető	feladatkiírás	Mozgás, Szoba oszódása,	

Azono-sító	Leírás	Ellenőrzés	Priori-tás	Forrás	Use-case	Kom-ment
					Szoba egyesülése	
F03	A játékban vannak játékosok	A játékosok az éppen kiválasztott mennyiségben létrejönnek.	alapvető	feladatkiírás	Mozgás, Tárgykezelés,	
F04	A játékban vannak oktatók	Az oktatók létrejönnek, a játékosok számától függően	alapvető	feladatkiírás	Mozgás, Tárgykezelés, Hallgatók kibuktatása	
F05	A logarléc felvétele jelenti a játék végét	A játék befejeződik ilyenkor	alapvető	feladatkiírás		
F06	Ajtón átlépéskor másik szobába kerülünk.	Másik szobába került a játékos.	alapvető	feladatkiírás	Tárgyat felvenni	
F07	Létezik és használható a TVSZ denevérbőrre nyomtatott példánya	Védettséget nyújt.	alapvető	feladatkiírás	Tárgyat felvenni, Tárgyat használni	
F08	Létezik és használható a szent söröspohár	Védettséget nyújt, egy ideig.	alapvető	feladatkiírás	Tárgyat felvenni, Tárgyat használni	
F09	Létezik és használható a dobozolt káposztás camembert.	Gázosítja a szobát.	alapvető	feladatkiírás	Tárgyat felvenni, Tárgyat használni	

Azono-sító	Leírás	Ellenőrzés	Priori-tás	Forrás	Use-case	Kom-ment
F10	Létezik és használható a tranzisztor párban.	Tud teleportálni.	alapvető	feladatkiírás	Tárgyat felvenni, Tárgyat használni	
F11	Létezik és használható az FFP2-es maszk.	Védettséget nyújt a gázzal szemben egy ideig.	alapvető	feladatkiírás	Tárgyat felvenni, Tárgyat használni	
F12	A szobában nem lehet több ember mint amennyi a férőhelye.	Nincs ott több ember.	alapvető	feladatkiírás	Pálya generálása, Mozgás	
F13	Gázos szobák létrejönnek.	Van olyan szoba ahol gáz van.	alapvető	feladatkiírás	Pálya generálása	
F14	Elátkozott szobák létrejönnek	Van olyan szoba ahol eltűnnek majd előtűnnek az ajtók.	alapvető	feladatkiírás	Pálya generálása	
F15	Osztódó és egybeolvadó szobák létrejönnek	Vannak olyan szobák amik először összeolvadnak majd szétválnak.	alapvető	feladatkiírás	Pálya generálása	

2.3.2 Erőforrásokkal kapcsolatos követelmények

[A szoftver fejlesztésével és használatával kapcsolatos számítógépes, hardveres, alapszoftveres és egyéb architekturális és logisztikai követelmények]

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
R01	A program Java nyelven készül.	Futtatható környezet léte.	alapvető	megrendelő, csapat	
R02	Futtatható legyen a kari felhőn	Lefut a kari felhőn	alapvető	megrendelő	
R03	A használathoz Windows 10 vagy Windows 11 ajánlott	Van operációs rendszer a gépen	opcionális	csapat	
R04	Fájlkezelési jogok megléte	Felhatalmazás megléte	alapvető	csapat	
R05	Szükséges eszközök: számítógép, billentyűzet, egér	Szükséges perifériák, és eszközök megléte	alapvető	csapat	

2.3.3 Átadással kapcsolatos követelmények

[A szoftver átadásával, telepítésével, üzembe helyezésével kapcsolatos követelmények]

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
D01	Az összes leírt követelmény teljesül.	Utolsó ellenőrzés sikeressége	alapvető	megrendelő	
D02	Megfelelő futtatható környezet megléte	Rendelkezünk futtatható környezettel.	alapvető	megrendelő	

2.3.4 Egyéb nem funkcionális követelmények

[A biztonsággal, hordozhatósággal, megbízhatósággal, tesztelhetőséggel, a felhasználóval kapcsolatos követelmények]

Azonosító	Leírás	Ellenőrzés	Prioritás	Forrás	Komment
N01	A rendszer más gépen is működik	A program fordítható és futtatható más eszközein is	alapvető	megrendelő	
N02	A program stabilan és konzisztensen fut.	Ismételt elindításkor is az elvárt viselkedést hozza.	alapvető	megrendelő	
N03	A felhasználónak nem szükséges előzetes tudás a játékszabályon kívül	A belső működés takart.	alapvető	csapat	

2.4 Lényeges use-case-ek

[A 2.3.1-ben felsorolt követelmények közül az alapvető és fontos követelményekhez tartozó használati esetek megadása az alábbi táblázatos formában.]

2.4.1 Use-case leírások

[Minden use-case-hez külön]

Use-case neve	Másik szobába lépni
Rövid leírás	A karakter átlép egy másik szobába.
Aktorok	Karakter
Forgatókönyv	<ol style="list-style-type: none"> A karakter egy olyan szobába próbál átlépni, amely nincs teli, akkor átlép. <ol style="list-style-type: none"> Ha így egy szobába lesz hallgató, olyan oktatóval, aki nem vesztette eszméltetét és nincs nála olyan tárgy, ami ezt megakadályozná akkor a hallgató kibukik.

	<p>Ha van nála akkor csak a tárgy használodik, kivéve ha az nedves táblatörő rongy, akkor a hallgató meg is bénul.</p> <p>B. Ha a karakter gázos szobábá lép, akkor ha van nála maszk, akkor használodik, ha nincs akkor eszméletét veszti.</p> <p>2. A karakter egy olyan szobába próbál átlépni, amely tele van, akkor nem fog átlépni.</p>
--	---

Use-case neve	Tárgyat elejteni, eszméletvesztés.
Rövid leírás	Az összes tárgy amit a karakter birtokol az a szobába lesz és nem fogja birtokolni a karakter és eszméletét veszti.
Aktorok	Karakter
Forgatókönyv	Amikor a karakter eszméletüket vesztik akkor az összes tárgy amit a karakter birtokol az a szobába lesz és nem fogja tovább birtokolni.

Use-case neve	Tárgyat felvenni
Rövid leírás	A karakter felvesz egy vele egy szobában lévő használható tárgyat.
Aktorok	Karakter
Forgatókönyv	<p>1. Ha a karakter tanár felveszi a tárgyat, kivéve ha logarléc, a tárgyat birtokolni fogja és a maszkot tudja majd használni.</p> <p>2. Ha a karakter hallgató:</p> <p>1. A hallgató megpróbálja felvenni a tárgyat</p> <p>2.A Ha a hallgató tranzisztor akar felvenni, de van nála már két tranzisztor vagy birtokol már egy aktív tranzisztort vagy már birtokol 5 tárgyat akkor nem sikerül neki felvenni.</p> <p>2.B Ha a hallgató tranzisztor akar felvenni és nincs nála már két tranzisztor és nem birtokol már egy aktív tranzisztort sem és még nem birtokol 5 tárgyat akkor sikerül neki felvenni a tranzisztor ami eltűnik a pályáról és innentől a játékosnál lesz.</p> <p>2.C Ha a hallgató nem tranzisztor akar felvenni, de birtokol 5 már tárgyat akkor nem sikerül neki felvenni.</p> <p>2.D Ha a hallgató nem tranzisztor akar felvenni és még nem birtokol 5 tárgyat akkor sikerül neki felvenni a tárgyat ami eltűnik a pályáról és innentől a játékosnál lesz.</p> <p>3. Ha a felvette a tárgyat és az a tárgy logarléc akkor megnyerték a játékosok a játékot.</p>

Use-case neve	Hallgató kibuktatása
Rövid leírás	Amikor egy játékos egy szobába tartózkodik egy oktatóval, akkor az oktató kibuktatja a játékosat.
Aktorok	Oktató
Forgatókönyv	Amikor egy játékos egy szobába tartózkodik egy oktatóval, akkor az oktató kibuktatja a játékosat, így a játékos már nem vesz részt a játékban és a karaktere és a nála lévő tárgyak eltűnnek a pályáról, a hallgató nem fog többet lépni a játékban. Ha a hallgató az utolsó hallgató volt, akkor a játék véget ér és a játékosok elveszítik a játékot.

Use-case neve	Tárgyak használata
Rövid leírás	A karakter használja a nála lévő tárgyat.
Aktorok	Karakter
Forgatókönyv	A karakter használja a nála lévő tárgyat, annak használatának megfelelően. A: Ha egy szobába kerül a hallgató az oktatóval, de van nála TVSZ denevérbőrre nyomtatott példánya vagy szent söröspohár, akkor az megmenti a hallgatót, viszont használódik. B: Ha egy szobába kerül a hallgató az oktatóval, de van nála van a nedves táblatörő rongy, akkor az megmenti a hallgatót és megbénítja az oktatót. C: A dobozolt káposztás camembert felbontáskor mérges gázat bocsát ki és elgázosítja a szobát. D: A hallgatonál levő tranzisztorokat páronként össze lehet kapcsolni, majd a pár egyik tagját menet közben egy másik szobában le lehet tenni. Az így összekapcsolt tranzisztorok varázserővel bírnak: ha a hallgató a nála maradó tranzisztor bekapcsolja és leteszi, akkor a másik tranzisztor szobájába kerül, a bekapcsolt tranzisztor pedig kikapcsol. E: Ha van a karakternél FFP2-es maszk és úgy lé, hogy egy gázzal teli szobába lesz akkor nem veszti eszméletét, hanem csak az FFP2-es maszk használódik egyet. Egy maszk maximum kétszer nyújt védelmet.

Use-case neve	Tárgyat eldobni
----------------------	-----------------

Rövid leírás	Eldobja a kiválasztott tárgyat.
Aktorok	Hallgató
Forgatókönyv	Eldobja a kiválasztott tárgyat, amit így nem birtokol tovább a hallgató és a szobába jelenik meg.

Use-case neve	idő számlálás
Rövid leírás	Időt számolja.
Aktorok	Pálya
Forgatókönyv	A lejátszott időt számolja.

Use-case neve	Szobák osztódása
Rövid leírás	Szobák osztódása
Aktorok	Pálya
Forgatókönyv	Az osztódó szoba két olyan szobára válik szét, amelyek egymás szomszédai lesznek, és megosztznak a korábbi szoba képességein és szomszédain (a korábbi szomszékok vagy csak az egyik, vagy csak a másik "új" szobának lesznek szomszédaik). Osztódni csak előtte már egyesült szobák osztódhatnak. Emellett olyan szobák sem osztódhatnak amelyben karakter van.

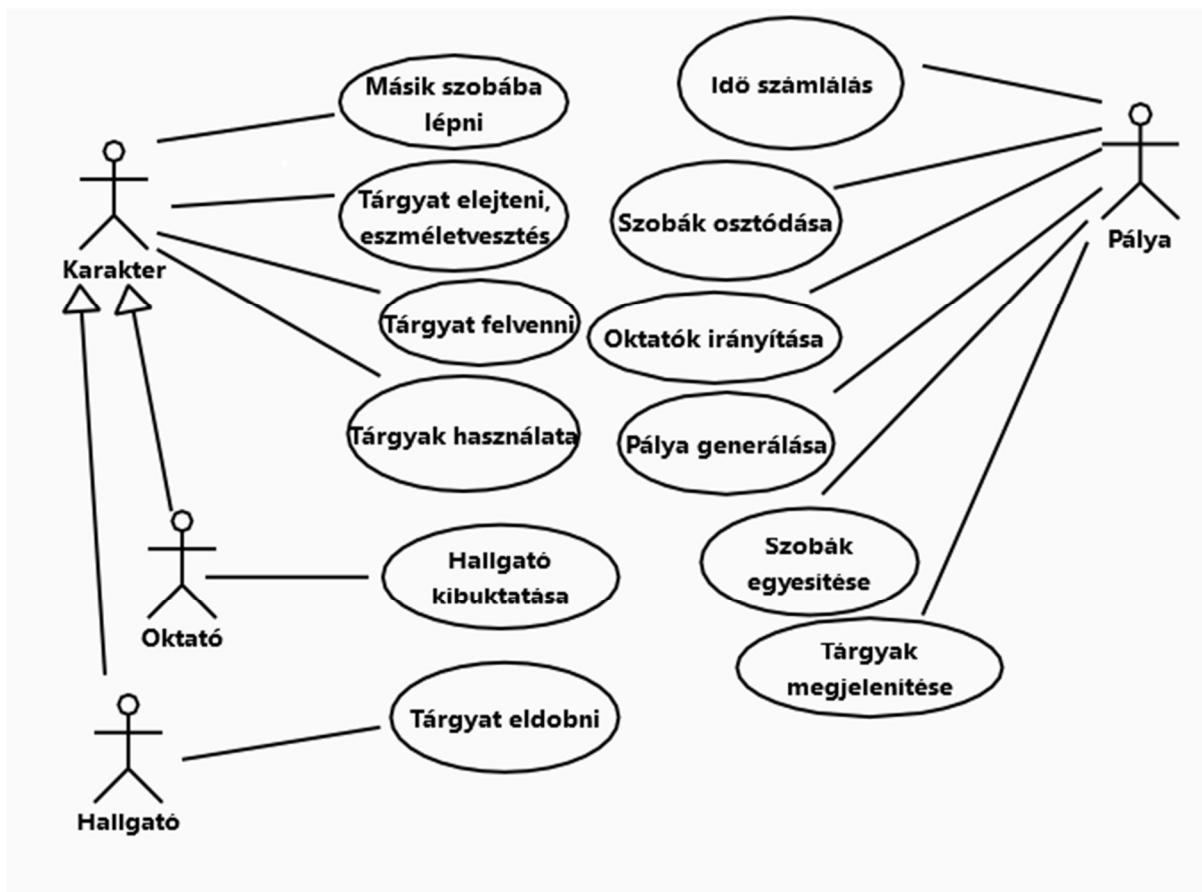
Use-case neve	Szobák egyesítése
Rövid leírás	Két szoba egy szobává egyesül.
Aktorok	Pálya
Forgatókönyv	Két szoba egy szobává egyesül. Két szomszédos szoba egyesülésével létrejövő szoba a korábbi két szoba tulajdonságaival és szomszédaival rendeletezik, de a befogadóképessége a nagyobb szoba befogadóképességével lesz azonos. Emellett olyan szobák sem egyesülhetnek amelyben karakter van.

Use-case neve	Oktatók irányítása
Rövid leírás	Oktatókat irányítja.
Aktorok	Pálya
Forgatókönyv	Oktatókat irányítja, lépteti.

Use-case neve	Pálya generálása
Rövid leírás	Pálya generálása
Aktorok	Pálya
Forgatókönyv	Létrehozza a pályát.

Use-case neve	Tárgyak megjelenítése
Rövid leírás	Megjeleníti a tárgyakat.
Aktorok	Pálya
Forgatókönyv	Megjeleníti a tárgyakat a pályán.

2.4.2 Use-case diagram



2.5 Szótár

Ajtó - Két szobát összekötő járat, amin hallgatók és oktatók áthaladhatnak egy másik szobába

Ajtó előtűnik - Ha egy ajtó eltűnt, képes egy időnként előtűnni elátkozott szobák esetében, ez után használhatják az oktatók és hallgatók

Ajtó eltűnik - Egy elátkozott szobában az ajtók képesek a hallgatók és oktatók számára eltűnni, ekkor számukra nem látható az ajtó, és emiatt nem használhatják, amíg elő nem tűnik

Befogadóképesség - Azt adja meg, hogy egy szobában hány karakter lehet

Dobozolt káposztás camembert - Ez egy tárgy, melynek használatával a szobából gázos szoba lesz

Egyirányú ajtó - Olyan két szobát összekötő járat, amely csak egyik szobából a másikba működik, a másik irányba nem

Elátkozott szoba - Ez egy speciális szoba, amely ajtajai képesek időnként eltűnni és előtűnni

Eldob - Egy hallgató szándékosa eldobhat tárgyakat, ekkor ezek abba a szobába kerülnek, amiben a hallgató van

Elejt - Egy karakter elejt egy tárgyat, ha valami arra kényszeríti (pl: gázos szoba), ekkor mint a tárgy eldobásánál, ez ekkor abba a szobába, amelyben a karakter tartózkodik, oda kerül

Eszméletvesztés - Egy karakter akkor veszíti el az eszméletét, ha egy gázos szobába kerül. Ekkor egy ideig nem csinálhat semmit, és elejti minden tárgyat. Ha letelt az eszméletvesztés ideje, akkor kénytelen elhagyni a szobát az elejtett tárgyai nélkül.

FFP2 maszk - Ez a tárgy véd meg az eszméletvesztéstől. Kétszer használható.

Gázos szoba - Ebbe a szobába belépéskor, ha nincs FFP2es maszkja, akkor eszméletét veszti. Ha van, akkor nem veszti el az eszméletét és használódik a maszkja.

Hallgató - A játékos által irányított karakter, aminek célja a Logarléc megtalálása és felvétele.

Játékos - A számítógép előtt ülő személy (felhasználó), aki egy hallgatót irányít a játékban

Karakter- Az oktatók és hallgatók összefoglaló neve

Kibukik az egyetemről - Ha egy hallgató találkozik egy oktatóval és nincs nála TVSZ denevérborre nyomtatott példánya, vagy szent söröspohár, akkor kibukik az egyetemről. Ebben az esetben az a hallgató kikerül a játékból és az általa hordozott tárgyak törlődnek.

Letesz - Egy hallgató letehet egy tranzisztor használásának érdekében

Logarléc - Egy tárgy, amelynek felvételével a hallgatók győznek. Ezt a tárgyat oktatók nem vehetik fel

Mérgező gáz - Ettől lesz egy szobából gázos szoba, a dobozolt káposztás camembert használatával jön létre

Nedves táblatörlő rongy - Egy tárgy, amellyel a használó hallgatóval egy szobában tartózkodó oktatók megbéníthatók egy időre

Oktató - Egy nem játékos által irányított karakter, amely célja a hallgatók hátráltatása a Logarléc megszerzésében.

Pálya - A teljes térképe a szobáknak, az azok közti ajtóknak és a szobákban lévő tárgyaknak

Szent söröspohár - Ennek viselője egy ideig immunis az oktatók egyetemről való kibuktatásáról. Ennek letelte után a tárgy törlődik.

Szoba - A pálya egy része, amelyben meghatározott számú karakter kerülhet be ajtókon keresztül, illetve tárgyak jelenhetnek meg és pihenhetnek benne

Szoba egyesül - Két szoba egyesülésekor a két külön szoba tárgyai az egyesültbe kerülnek, a befogadóképessége a nagyobb szoba befogadóképességével lesz azonos, illetve a szomszédjait mindenkorábbi szobának megkapja

Szoba osztódik - Két már egyesült szoba osztódásakor visszakerülnek eredeti állapotukba

Szoba tulajdonsága - A szoba tulajdonsága utal arra, ha egy speciális szobáról van szó (gázos, elátkozott), illetve az abban rejlő tárgyak összességét írja le

Szobába lép - Egy karakter akkor lép egy szobába, ha egy másik szobából a felé vezető ajtón távozik, ezáltal a következő körben oda kerül

Szomszéd - Két szoba akkor szomszédos, ha van köztük ajtó

Tárgy - Karakterek által felvehető eszközök, amelyek vagy passzív vagy aktív képességekkel ruházzák fel használóikat

Tranzisztor - Egy tárgy, amely páronként használható. Ha egy hallgatónál van két tranzisztor és az egyiket letesz, akkor aktiválódnak a tranzisztorok. Ha egy másik szobában a másikat is letesz, akkor az első szobájába kerül, és innentől kezdve minden az lesz bekapcsolt állapotban, amelyhez utaztak a másiktól. Egy hallgatónál legfeljebb két tranzisztor lehet, de ha van nála egy aktivált, akkor nem vehet fel újat.

TVSZ denevérbőrre nyomtatott példánya - Ennek viselője 3 alkalommal megvédi az oktatók egyetemről való kibuktatásáról. Ez után a tárgy törlődik

Védeeltség - A gázos szobákban ha egy karakter védt, nem veszítíti eszméletét

2.6 Projekt terv

Hét	Feladat	Határidő
2	Követelmény, projekt, funkcionális	feb. 26.
3	Analízis modell (I. változat)	márc. 4.
4	Analízis modell (II. változat)	márc. 11.
5	Szkeleton tervezése	márc. 18.
6	Szkeleton elkészítése	márc. 25.
Tavaszi szünet	(Tavaszi szünet)	-
7	Prototípus koncepciója, Szkeleton bemutatása	ápr. 8.
8	Részletes tervek	ápr. 15.
9	Prototípus elkészítése	ápr. 29.
10	(prototípus készítése)	-
11	Grafikus változat tervei	máj. 6.
12	(graf. vált. készítése)	-
13	Grafikus változat elkészítése, Prototípus bemutatása	máj. 22.
14	Egyesített dokumentáció, Grafikus verzió bemutatása	máj 24.

Dokumentálásra használt eszközök: Microsoft Word, Google Docs

Kommunikáció: Személyesen, Messenger, Discord

Modellező eszköz: Gaphor

Fejlesztőeszköz: Eclipse, IntelliJ, Visual Studio Code

Dokumentumok megosztása: Google Docs

Forráskód megosztása: GitHub

2.7 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2024.02.20. 18:00	1,5 óra	Teljes csapat.	Értekezlet. Döntések: Előzetes megbeszélés, terv előkészítése.
2024.02.21. 20:30	1 óra	Teljes csapat.	Értekezlet. Döntések: Feladatok szétosztása, funkciók közös leírása.
2024.02.21. 23:00	1 óra	Németh	Bevezetés és Áttekintés részbeli elkészítése
2024. 02. 22. 10:00	1 óra	Czotter	Use case diagram elkészítése
2024. 02. 23. 15:00	1 óra	Schulcz	Use-case leírások elkészítése
2024.02.24. 7:45	1 óra	Hermann	Követelményrendszer megírása
2024.02.24. 10:00	1,5 óra	Tarsoly	Szótár, Projekt terv megírása
2024.02.24. 20:45	45 perc	Csapat	Értekezlet. Döntések: Heti feladat lezárása, áttekintése, kisebb hibák javítása.

Analízis modell kidolgozása

I.

35 – pupakok

Konzulens:
Koczó Attila

Csapattagok:

Tarsoly Levente	E1IK75	tarsolyl@edu.bme.hu
Németh Gergely	TA9G0R	gergo.h.nemeth@gmail.com
Schulcz Gábor	EP3I9R	schulcz.gabor6@gmail.com
Czotter Benedek	TFB4FY	czottibeni@gmail.com
Hermann Máté Béla	T83K5I	mate3829@gmail.com

2024.03.04.

3. Analízis modell kidolgozása

3.1 Objektum katalógus

3.1.1 Ajtó

A szoba része, ezen keresztül lehet a szobák között mozogni.

3.1.2 Camembert

Egy tárgy, ami szobákból felvehető. Használatával elgázosítható egy szoba.

3.1.3 Egyirányú ajtó

Az ajtó egy speciális fajtája, ami csak egyik irányba engedi a közlekedést.

3.1.4 Elátkozott szoba

Az szoba egy olyan fajtája aminek vannak eltűnő, majd előtűnő ajtói.

3.1.5 Gázos Szoba

A szoba egy olyan fajtája, ami gázzal van megtelítve ami elkábítja a karaktereket belépéskor.

3.1.6 Hallgató

Egy irányítható karakter. Célja a logarléc nevű tárgy megszerzése.

3.1.7 Logarléc

Egy tárgy, ami szobákból felvehető. Ennek felvételével a hallgatók megnyerik a játékot.

3.1.8 Maszk

Egy tárgy, ami szobákból felvehető. Ennek segítségével nem vesztik eszméletüket a játékosok

3.1.9 Oktató

A játék által irányított karakter. Célja a hallgatók akadályozása.

3.1.10 Rongy

Egy tárgy, ami szobákból felvehető. Ennek használatával megbénulnak az oktatók.

3.1.11 Söröspohár

Egy tárgy, ami szobákból felvehető. Ez védelmet nyújt az oktatók ellen.

3.1.12 Szoba

A pálya része, ahol a játék történik. Ebben lehet felvenni a tárgyakat, és itt mozognak a karakterek.

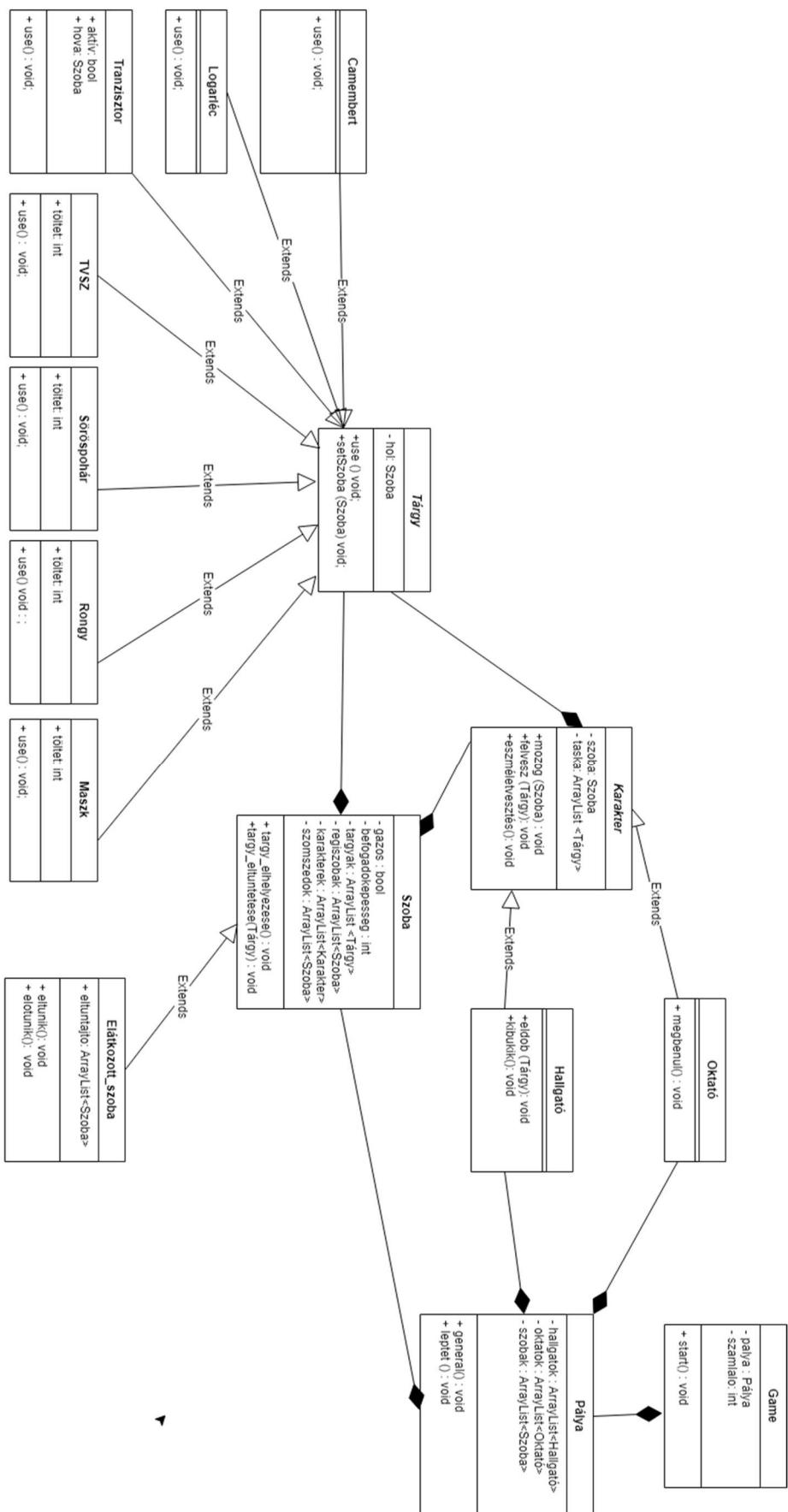
3.1.13 Tranzisztor

Egy tárgy, ami szobákból felvehető. Ennek használatával képesek hallgatók szobák közt ajtó nélkül közlekedni

3.1.14 TVSZ

Egy tárgy, ami szobákból felvehető. Ez védelmet nyújt az oktatók ellen

3.2 Statikus struktúra diagramok



3.3 Osztályok leírása

3.3.1 Camembert

- **Felelősség**

A feladat leírásában lévő Dobozolt káposztás camembert tárgyat reprezentálja. Ennek használatával egy szoba átalakítható gázos szobává.

- **Ősosztályok**

- Tárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- **Metódusok**

- void use(): A paraméterben átadott szobában elhasználódik, ennek köszönhetően az a szoba gázos lesz. Ez után a tárgy eltűnik.

3.3.2 Elátkozott_szoba

- **Felelősség**

A szobának egy olyan fajtájáért felel aminek vannak a szobán túlmutató tulajdnságai, ugyanis lehetnek eltűnő és előtűnő ajtói.

- **Ősosztályok**

Szoba

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- ArrayList<Szoba> eltuntajto : eltároljuk az eltűnt ajtókat, hogy később vissza tudjuk állítani.

- **Metódusok**

- void eltunik() : eltüntet egy ajtót, így azon nem lehet átmenni amíg nem kerül elő újra.
- void elotunik() : előtűntet egy korábban eltűnt ajtót, így azon újra át lehet menni.

3.3.3 Game

- **Felelősség**

Ez felel a program indulásáért és megállásáért.

- **Ősosztályok**

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- Pálya palya : a kezdeti pálya
- int szamlalo : ez adja meg milyen hosszú a játék, ha a leteltéig nem sikerül megtalálni a logarlécet akkor a hallgatók vesztettek.

- **Metódusok**

- void start() : a függvény ami indítja és zárja a játékot.

3.3.4 Hallgató

- **Felelősség**

Ez az osztály felel az irányítható karakterért a hallgató tulajdonságaiért.

- **Ősosztályok**

Karakter

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- **Metódusok**

- void eldob(Tárgy) : a hallgató saját akaratból is dobhat el tárgyat, ennek a függvénynek a segítségével.
- void kibukik() : ha a hallgató kibukik, akkor megszűnik létezni és a tárgyai megsemmisülnek.

3.3.5 Karakter

- **Felelősség**

Ez az absztrakt osztály felel a karakterek a működéséért. Ez kezeli azokat az eseményeket amik egy karakterrel történhetnek, a tárgyfelvétel és letétel, mozgás és eszméletvesztés.

- **Ősosztályok**

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- Szoba szoba : a szoba ahol a karakter tartózkodik.
- ArrayList<Tárgy> taska : tárgyak listája ami a karakternél van.

- **Metódusok**

- void mozog(Szoba) : a karakter mozgását végzi
- void felvesz(Tárgy) : felvesz egy tárgyat és eltárolja a táskahában.
- void eszméletvesztés() : ez végzi az eszméletvesztéskor beteljesülő eseményeket, a tárgy és kimerítést.

3.3.6 Logarléc

- **Felelősség**

Ennek a tárgynak felvételével a hallgatók megnyerik a játékot.

- **Ősosztályok**

Tárgy

- **Interfész**

- **Asszociáció**

- **Attribútumok**

- **Metódusok**

- void use(): Ez automatikusan lefut, amikor a hallgató felveszi ezt a tárgyat, és ezzel megnyerik a játékot.

3.3.7 Maszk

- **Felelősség**

A feladat leírásában lévő FFP2es maszk tárgyat reprezentálja. Megvédi ezt a tárgyat birtokló karaktereket az eszméletvesztéstől a gázos szobákban.

- **Ősosztályok**

Tárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- int töltet: Ez számlálja azt, hogy az adott maszk hányszor használható még. Ha ez letelik, akkor a tárgy már nem funkcionál.

- **Metódusok**

- void use(): Ha a játékos gázos szobába lép, akkor használja ezt a tárgyat, ezzel a töltete eggyel csökken, viszont nem veszti eszméletét és folytathatja a játékot.

3.3.8 Oktató

- **Felelősség**

Ez az osztály felel a játék által irányított karakter, az oktató tulajdonságaiért.

- **Ősosztályok**

Karakter

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- **Metódusok**

- void megbenul(): ha van megfelelő tárgy a hallgatónál és egy szobába kerülnek akkor az oktató megbénul és nem tud kibuktatni hallgatót.

3.3.9 Pálya

- **Felelősség**

Ez az osztály felel a játék generálásáért, valamint az oktatók mozgatásáért.

- **Ősosztályok**

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- ArrayList<Hallgató> hallgatok: összes hallgató listája
- ArrayList<Oktató> oktatok: összes oktató listája
- ArrayList<Szoba> szobák: összes szoba listája

- **Metódusok**

- void general() : legenerálja a pályát, vagyis a szobákat, azoknak a különböző típusait, valamint a játékosokat és tárgyakat.
- void leptet() : lép az oktatókkal, valamint a véghezviszi a pályabeli változtatásokat.

3.3.10 Rongy

- **Felelősség**

A feladat leírásában lévő Nedves táblatörlő rongy tárgyat reprezentálja. Ennek segítségével megbéníthatók az oktatók, ha egy szobában tartózkodnak egy olyan hallgatóval, amelynél van egy ilyen tárgy.

- **Ősosztályok**

Tárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- int töltet: Ez számlálja azt, hogy az adott rongy mennyi ideig használható még. Ha ez letelik, akkor a tárgy már nem funkcionál.

- **Metódusok**

- void use(): Ha egy hallgató egy szobába kerül egy oktatóval, akkor azt megbénítja egy adott időre.

3.3.11 Söröspohár

- **Felelősség**

A feladat leírásában lévő Szent söröspohár tárgyat reprezentálja. Ez egy ideig képes védelmet nyújtani az oktatók ellen a hallgatóknak.

- **Ősosztályok**

Tárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- int töltet: Ez számlálja azt, hogy az adott söröspohár mennyi ideig használható még. Ha ez letelik, akkor a tárgy már nem funkcionál.

- **Metódusok**

- void use(): Amíg egy hallgatónál van ez a tárgy, ez fogja a töltetet csökkenteni

3.3.12 Szoba

- **Felelősség**

Ez az osztály felel a szobának a működéséért. Itt fog történni a játék, itt fognak mozogni a karakterek.

- **Ősosztályok**

- **Interfész**

- **Asszociációk**

- **Attribútumok**

- bool gázos : megmutatja, hogy gázos-e a szoba
- int befogadokepesseg : hány fő lehet a szobában
- ArrayList<Tárgy> targyak : a szobában található tárgyak listája
- ArrayList<Szoba> regisztabak : tárolja a szobákat amiből összeolvadt egy szoba, hogy az alapján szét is tudjon osztódni.
- ArrayList<Karakter> karakterek : a szobában tartózkodó karakterek
- ArrayList<Szoba> szomszedok : szomszédos szobák listája

- **Metódusok**

- void targy_elhelyezese() : elhelyez egy tárgyat a szobában amit fel lehet venni.
- void targy_eltuntetese(Tárgy) : eltüntet egy tárgyat, hogy ne lehessen felvenni többször, ha egy játékos felveszi.

3.3.13 Tárgy

- **Felelősség**

Egy absztrakt osztály, amely a tárgyat reprezentálja.

- **Ősosztályok**

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- Szoba hol: Ez azt írja, le, hogy a tárgy melyik szobában van, ha ez egy karakternél van ez a tárgy, akkor is változik a karaktertől függően.

- **Metódusok**

- void use(): A tárgyat használja a karakter.

3.3.14 Tranzisztor

- **Felelősség**

A feladat leírásában lévő Tranzisztor tárgyat reprezentálja. Ennek segítségével képesek a hallgatók két tranzisztor összekapcsolása után egyik nem szomszédos szobából átmenni a másikba, ajtók használata nélkül.

- **Ősosztályok**

Tárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- bool aktív: Ez azt írja le, hogy a tranzisztor aktív-e
- Szoba hova: Ez azt tárolja, hogy ha össze van kötve két tranzisztor, akkor melyikbe visz át a használatakor

- **Metódusok**

- void use(): Első használatkor aktívvá teszi a tranzisztorokat, és az elsőt leteszi. Második használatkor leteszi a másikat a jelenlegi szobába és átmegy az első letett tranzisztorhoz, és a második aktivitását kikapcsolja. Ez után további használatokkor ha aktív a tranzisztor átmegy a karakter a párjához, és a kiindulót be, a másikat pedig beaktiválja.

3.3.15 TVSZ

- **Felelősség**

A feladat leírásában levő TVSZ denevérbőrre nyomtatott példányát reprezentálja. Megvédi a hallgatót a vele egy szobában lévő oktató általi kibuktatástól.

- **Ősosztályok**

Tárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

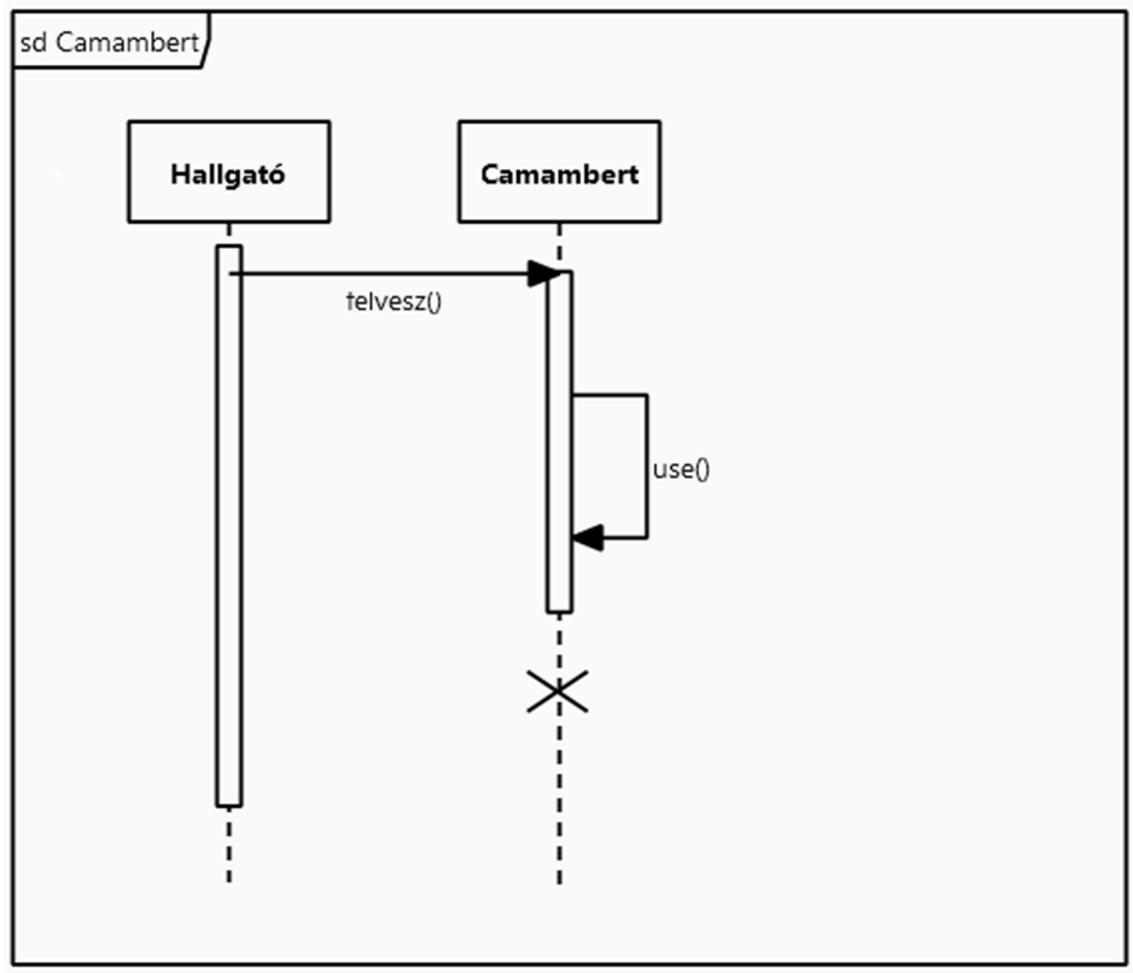
- int töltet: Ez számlálja azt, hogy az adott TVSZ hányszor használható még.

- **Metódusok**

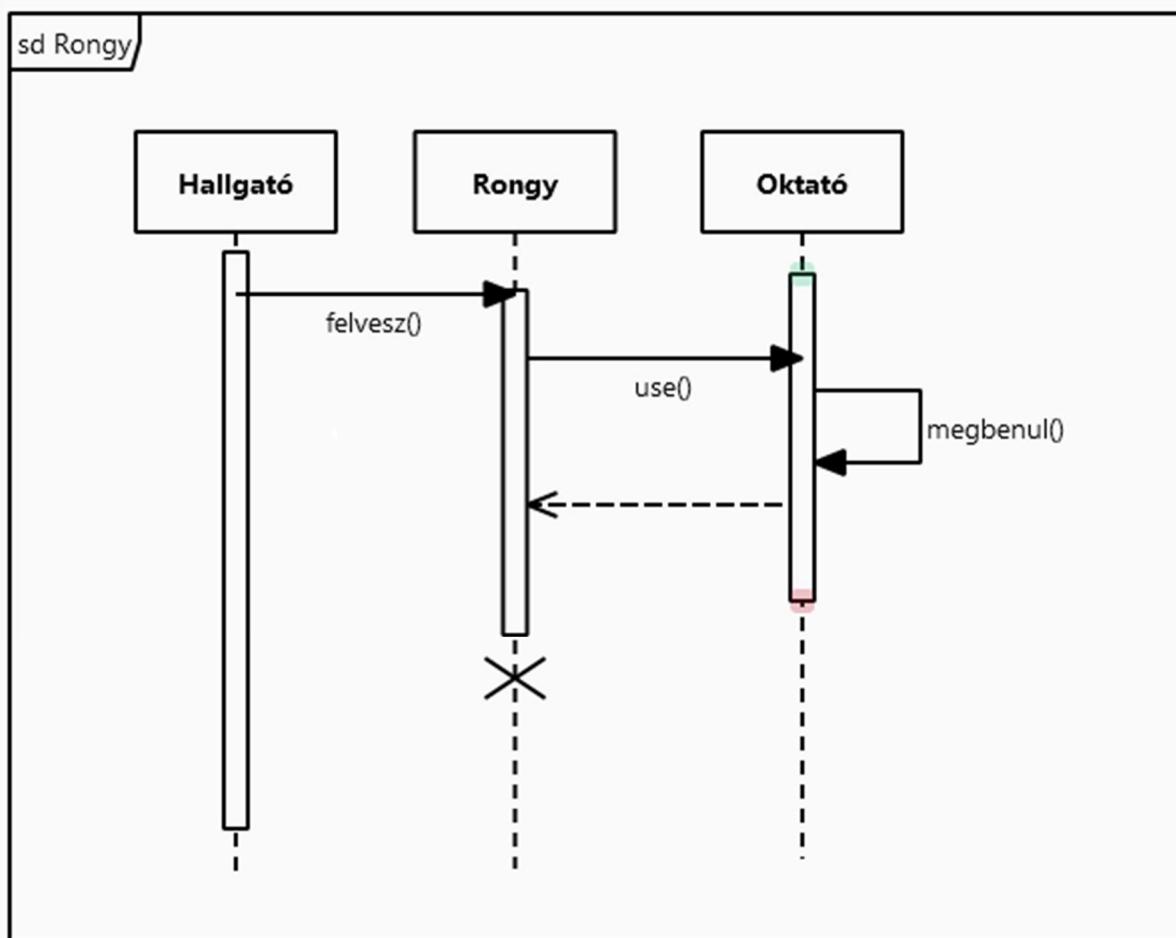
- void use(): Ha a játékos egy szobába lép, amelyben oktató is tartózkodik, akkor használja ezt a tárgyat, ezzel a töltete eggyel csökken, viszont folytathatja a játékot.

3.4 Szekvencia diagramok

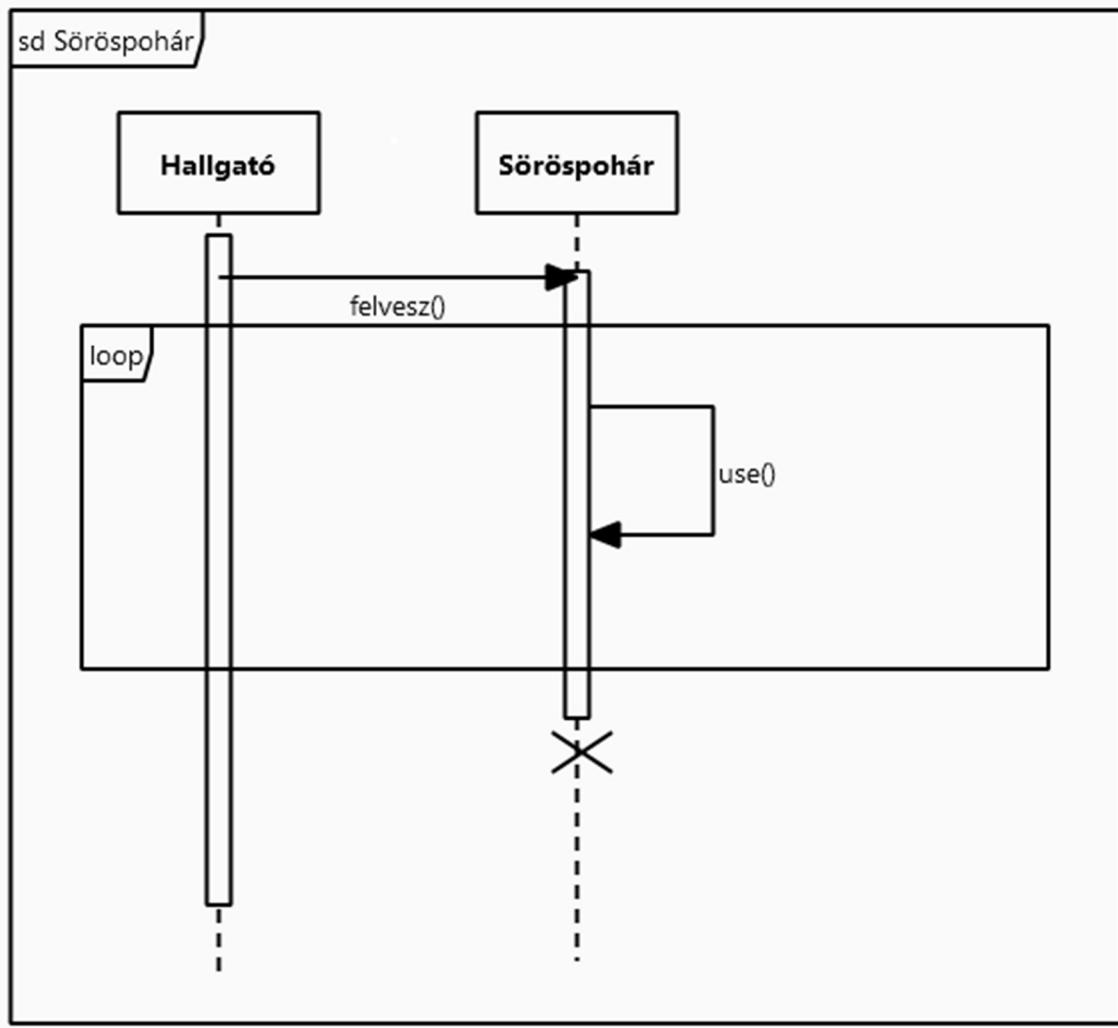
3.4.1 Káposztás camembert használata



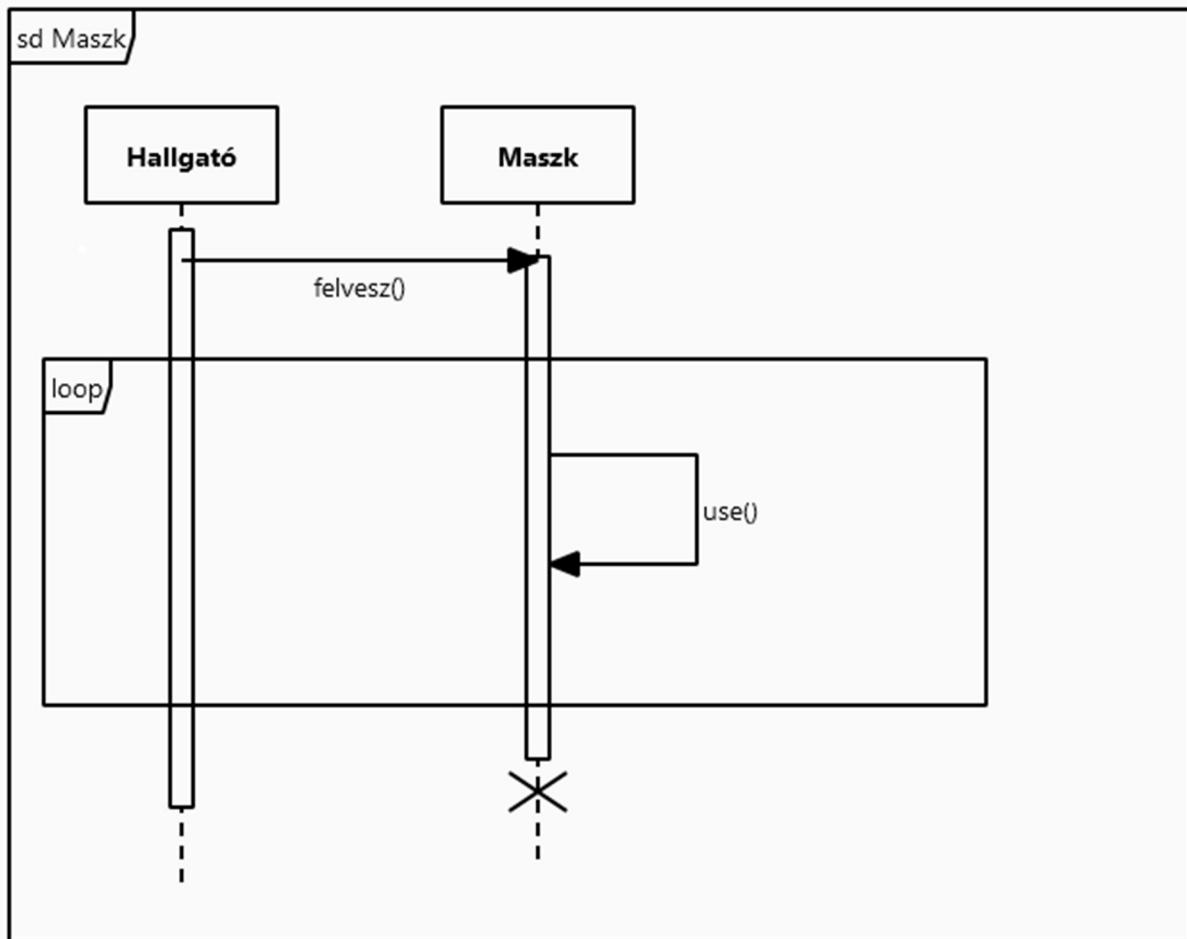
3.4.2 Rongy használata



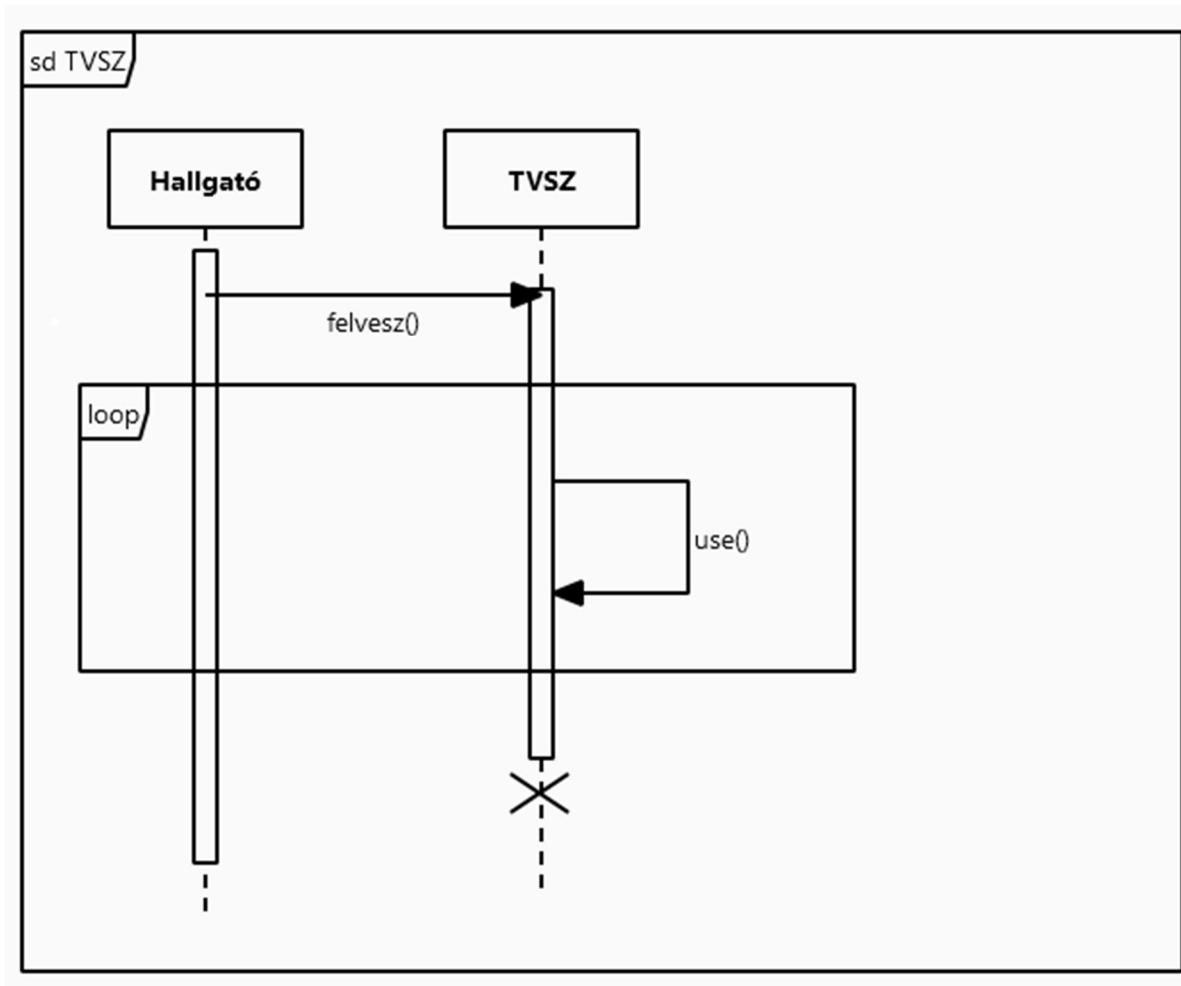
3.4.3 Söröspohár használata



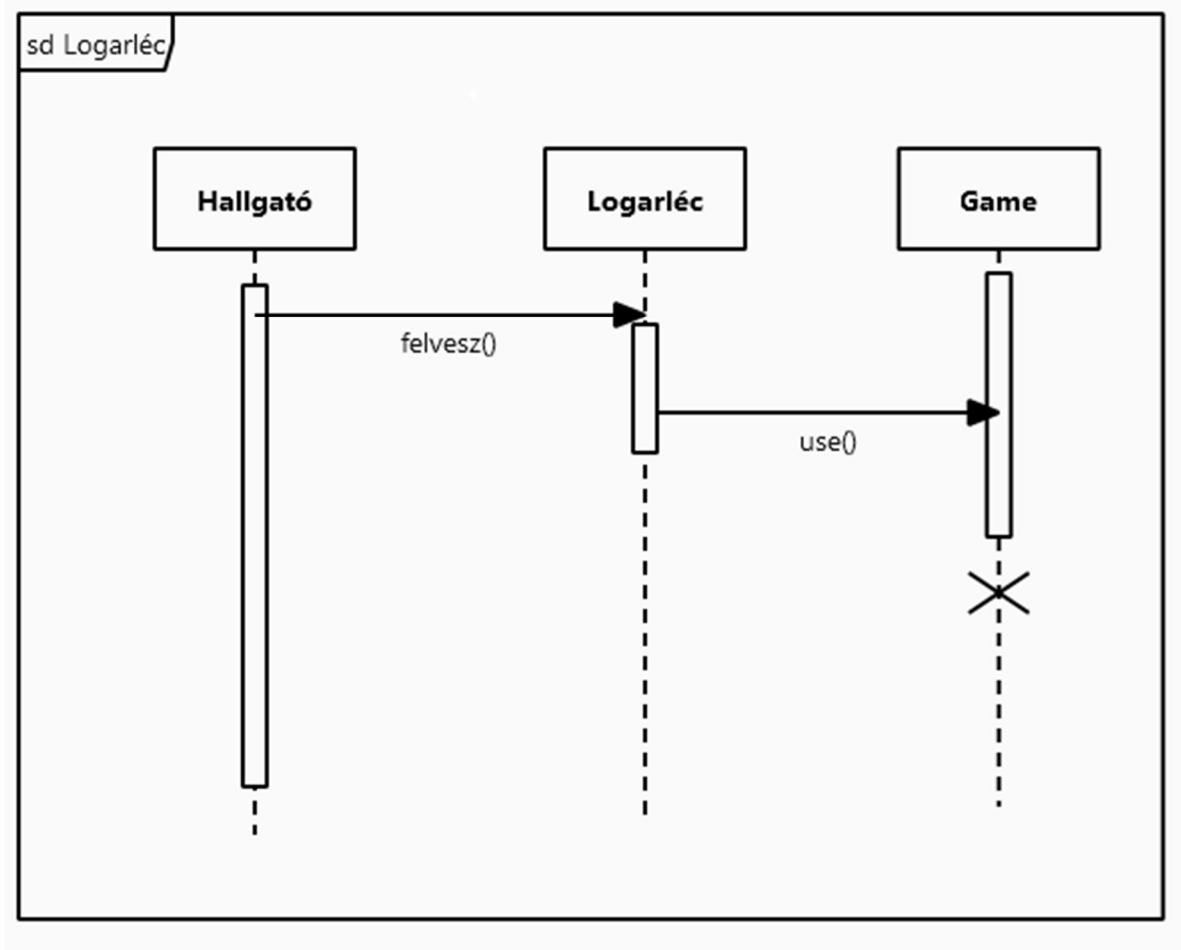
3.4.4 FFP2-es maszk használata



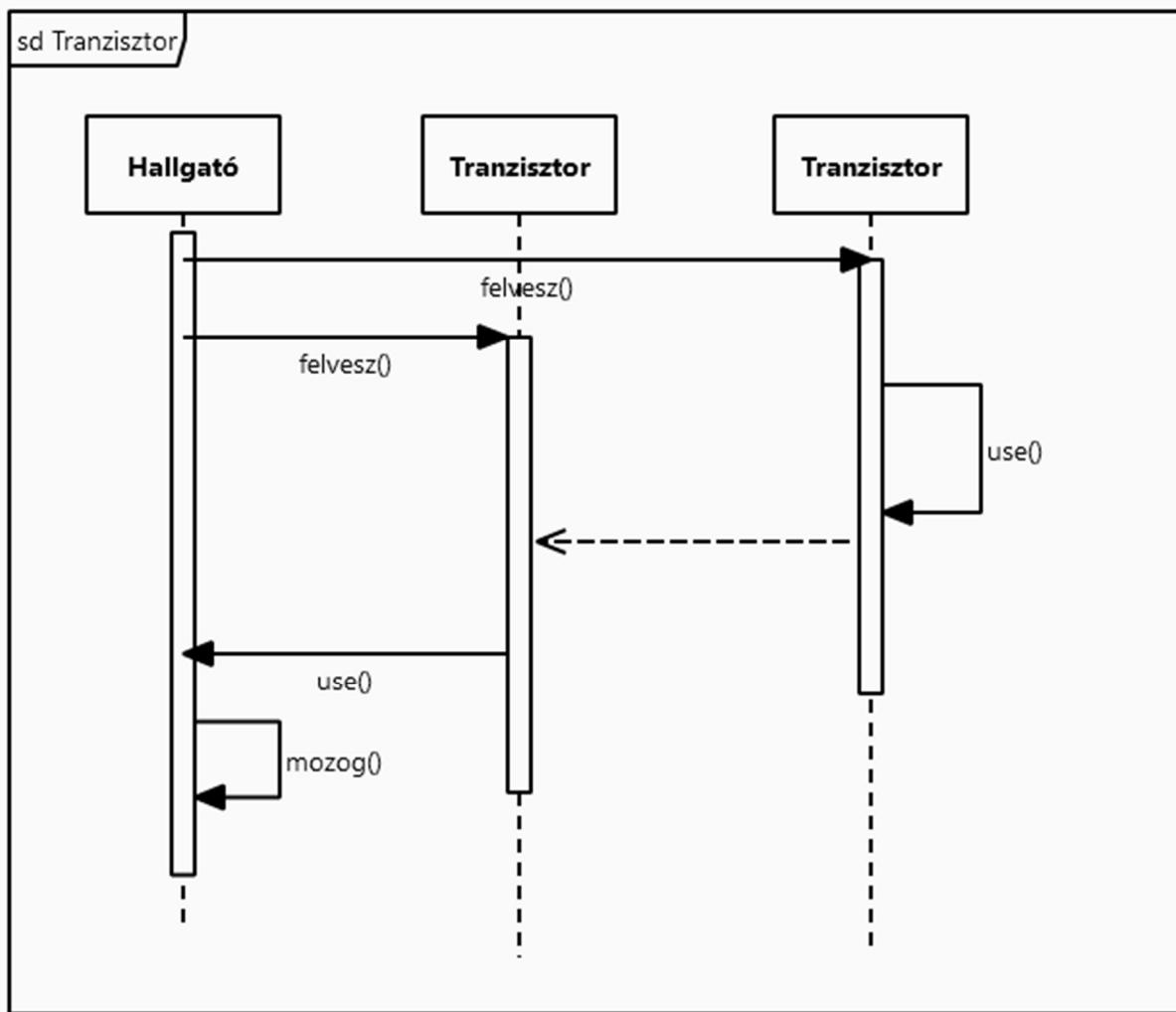
3.4.5 TVSZ használata



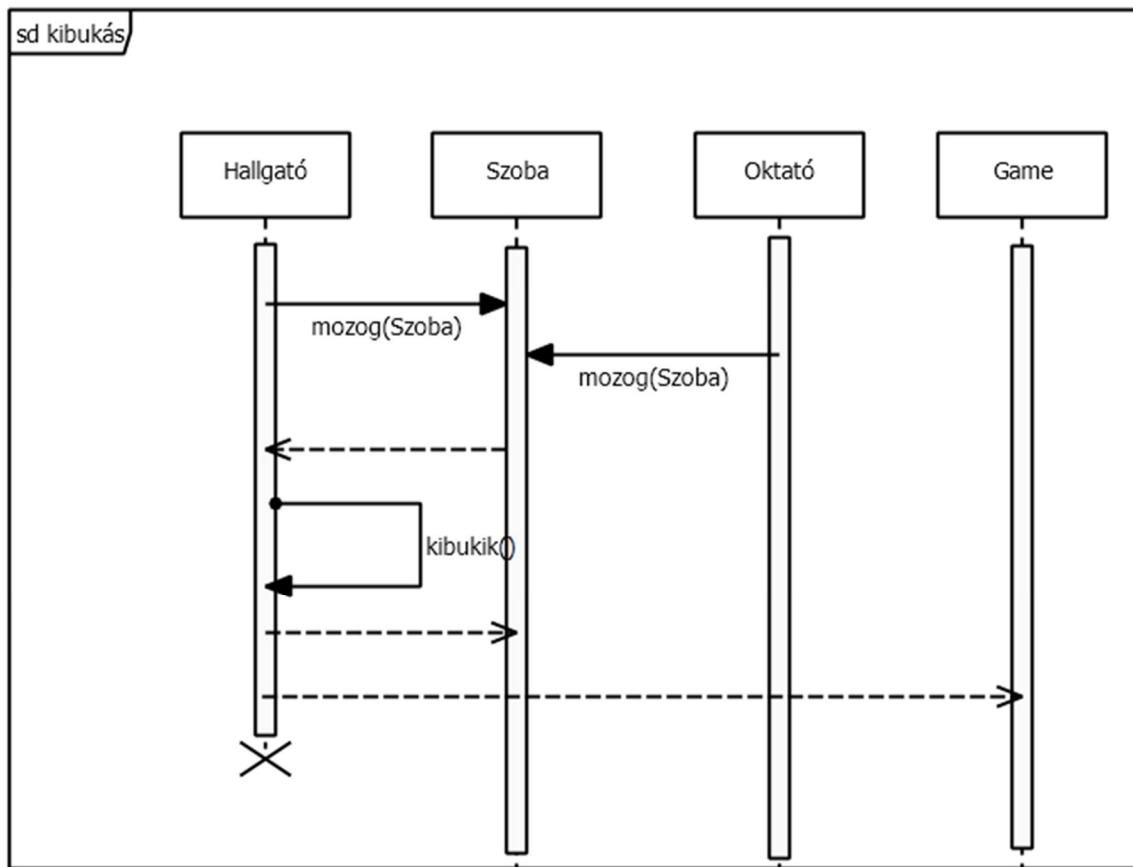
3.4.6 Logarléc



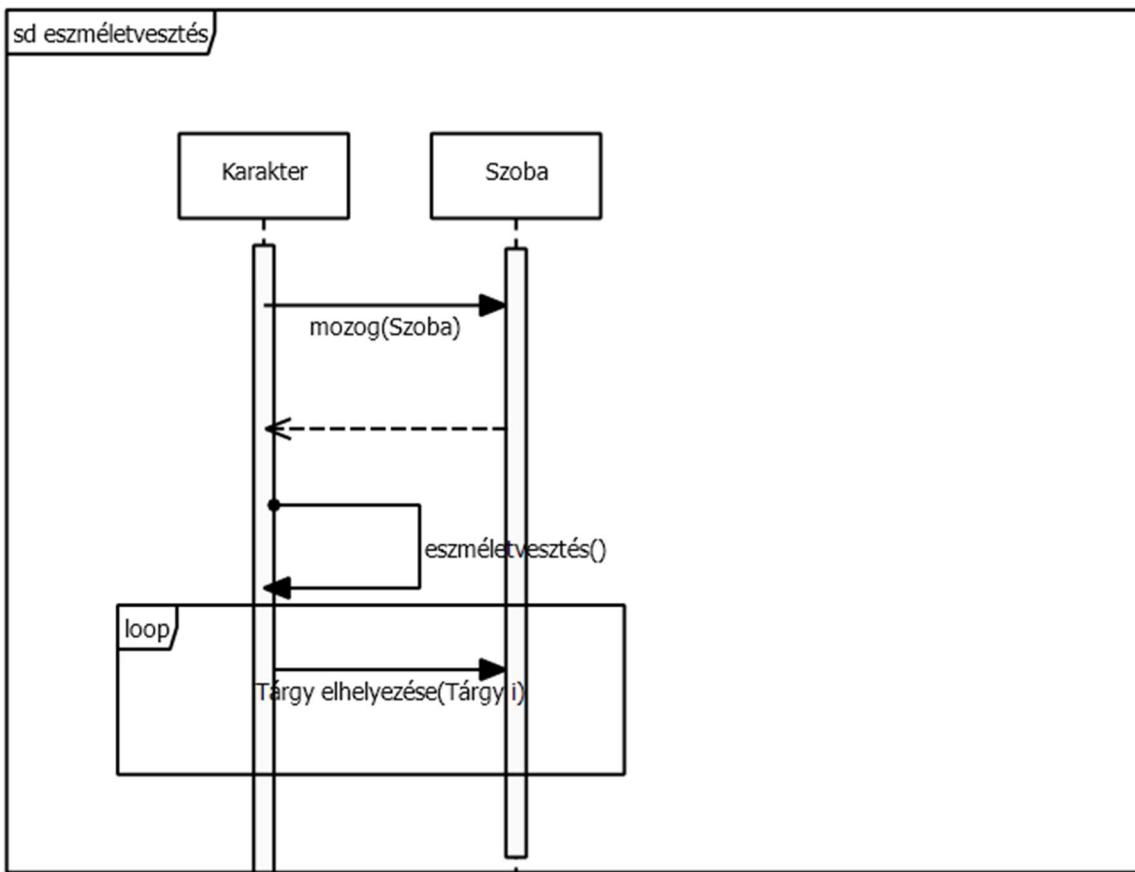
3.4.7 Tranzisztor használata



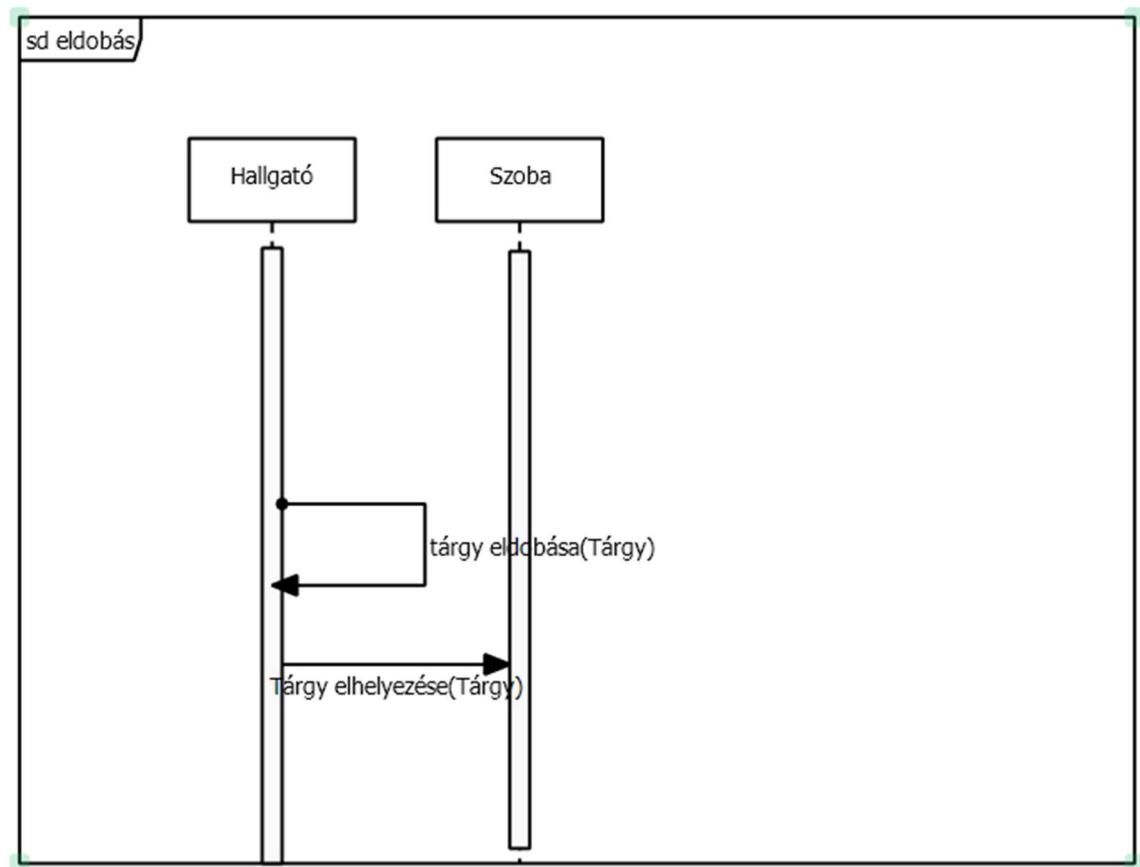
3.4.8 Hallgató kibukása



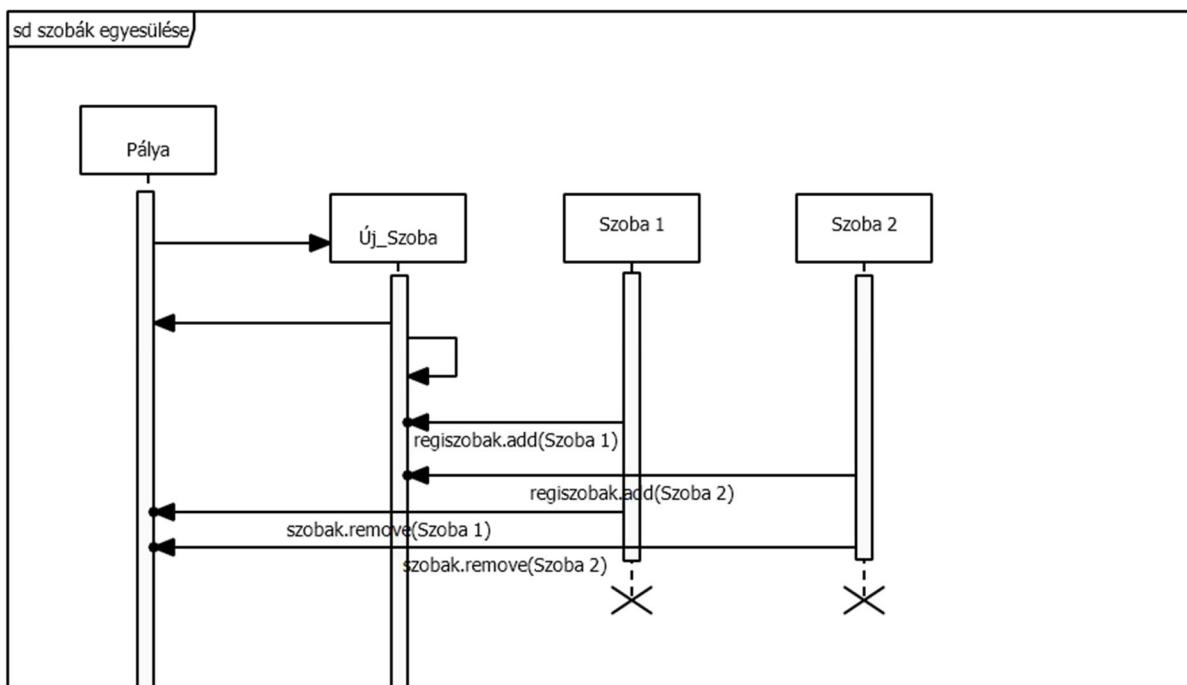
3.4.9 Karakterek eszméletvesztése



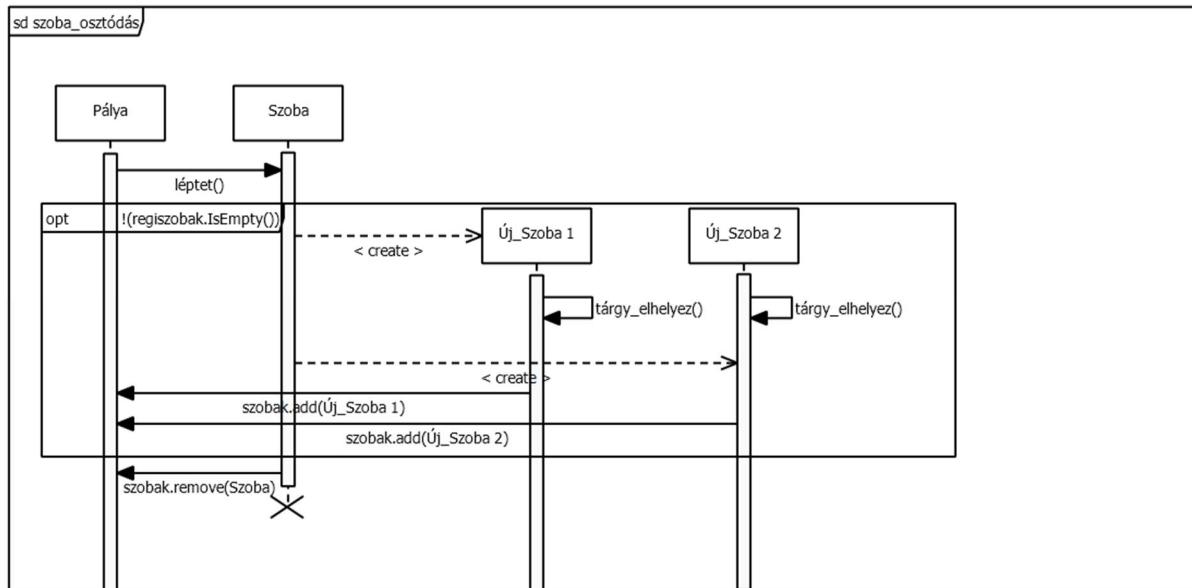
3.4.10 Eldobás



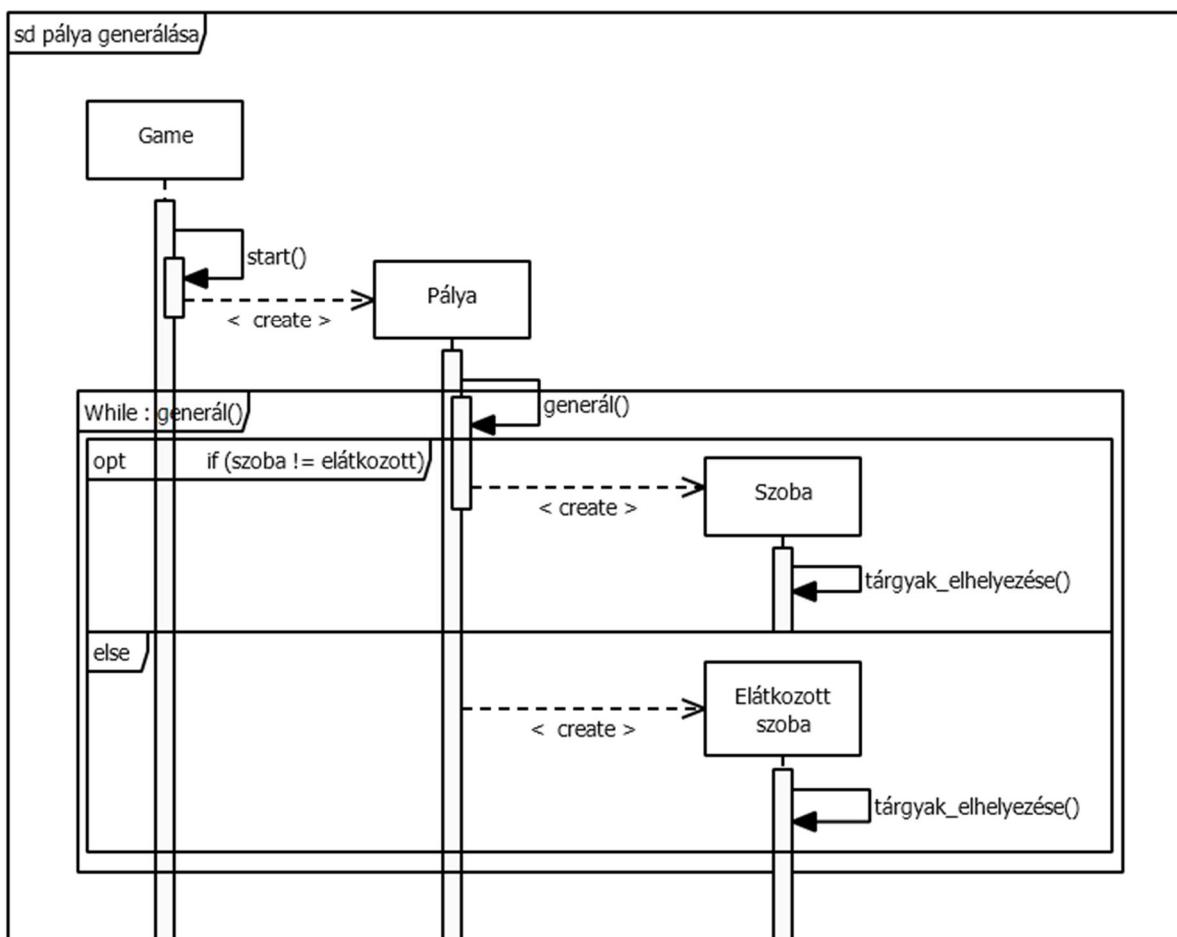
3.4.11 Szobák egyesülése



3.4.12 Szobák osztódása

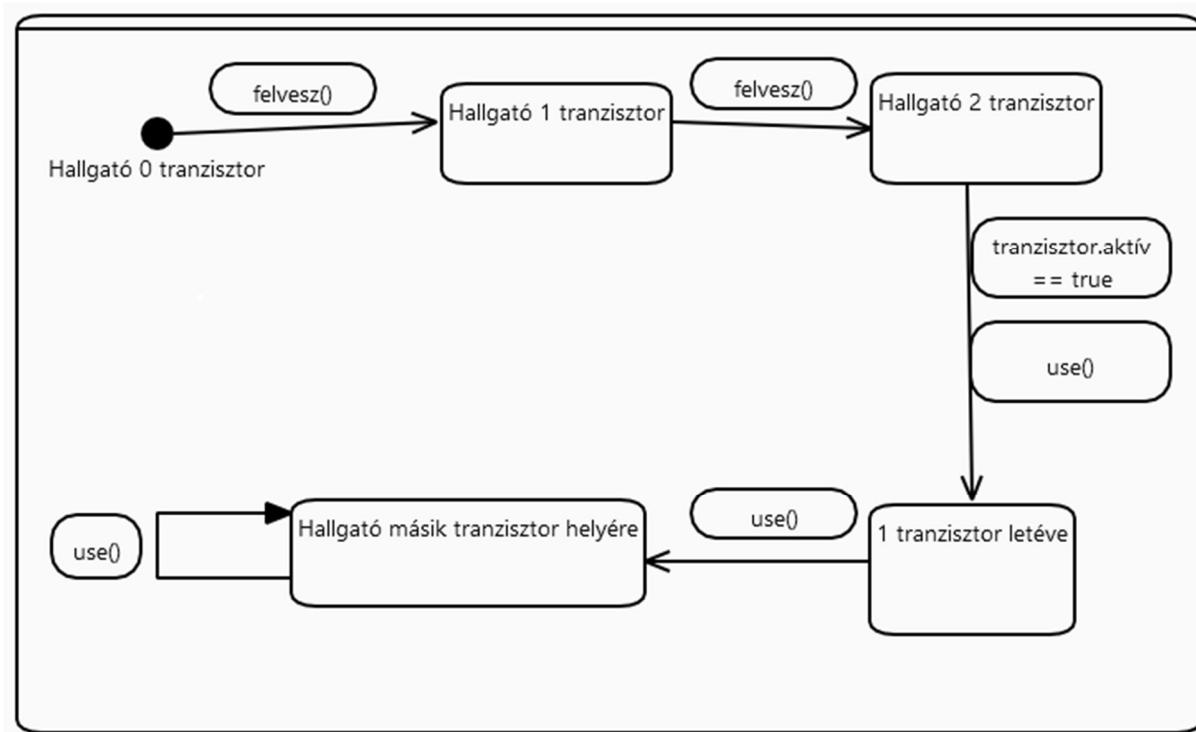


3.4.13 Pálya generálása



3.5 State-chartok

3.5.1 Tranzisztor használata



3.6 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2024.02.28. 20:15	2 óra	Csapat	Értekezlet. Döntés: Osztálydiagram elkezdése.
2024.02.29. 20:15	2 óra	Csapat	Értekezlet. Döntés: Osztálydiagram befejezése, feladatok kiosztása.
2024.03.01.9: 00	2 óra	Czotter	Szekvencia diagramok, állapotgép készítése
2024.03.02.14 :00	2 óra	Schulcz	Szekvencia diagramok
2024.03.03. 18:00	2 óra	Németh	Szekvencia diagramok

2024.03.03. 20:00	2 óra	Csapat	Értekezlet. Döntések: Heti feladatok megbeszélése és lezárása/javítása.
2024.03.03. 19:00	1,5 óra	Tarsoly	Objektum-katalógus és osztályok leírása
2024.03.04. 11:00	2 óra	Hermann, Tarsoly	Objektum-katalógus és osztályok leírása, dokumentum megformázása, végső ellenőrzés.

Analízis modell kidolgozása

II.

35 – pupakok

Konzulens:
Koczó Attila

Csapattagok:

Tarsoly Levente	E1IK75	tarsolyl@edu.bme.hu
Németh Gergely	TA9G0R	gergo.h.nemeth@gmail.com
Schulcz Gábor	EP3I9R	schulcz.gabor6@gmail.com
Czotter Benedek	TFB4FY	czottibeni@gmail.com
Hermann Máté Béla	T83K5I	mate3829@gmail.com

2024.03.11

4. Analízis modell kidolgozása

4.1 Objektum katalógus

4.1.1 Ajtó

A szoba része, ezen keresztül lehet a szobák között mozogni.

4.1.2 Camembert

Egy tárgy, ami szobákból felvehető. Használatával elgázosítható egy szoba.

4.1.3 Egyirányú ajtó

Az ajtó egy speciális fajtája, ami csak egyik irányba engedi a közlekedést.

4.1.4 Elátkozott szoba

Az szoba egy olyan fajtája aminek vannak eltűnő, majd előtűnő ajtói.

4.1.5 Gázos Szoba

A szoba egy olyan fajtája, ami gázzal van megtelítve ami elkábítja a karaktereket belépéskor.

4.1.6 Hallgató

Egy irányítható karakter. Célja a logarléc nevű tárgy megszerzése.

4.1.7 Logarléc

Egy tárgy, ami szobákból felvehető. Ennek felvételével a hallgatók megnyerik a játékot.

4.1.8 Maszk

Egy tárgy, ami szobákból felvehető. Ennek segítségével nem vesztik eszméletüket a játékosok

4.1.9 Oktató

A játék által irányított karakter. Célja a hallgatók akadályozása.

4.1.10 Rongy

Egy tárgy, ami szobákból felvehető. Ennek használatával megbénulnak az oktatók.

4.1.11 Söröspohár

Egy tárgy, ami szobákból felvehető. Ez védelmet nyújt az oktatók ellen

4.1.12 Szoba

A pálya része, ahol a játék történik. Ebben lehet felvenni a tárgyakat, és itt mozognak a karakterek.

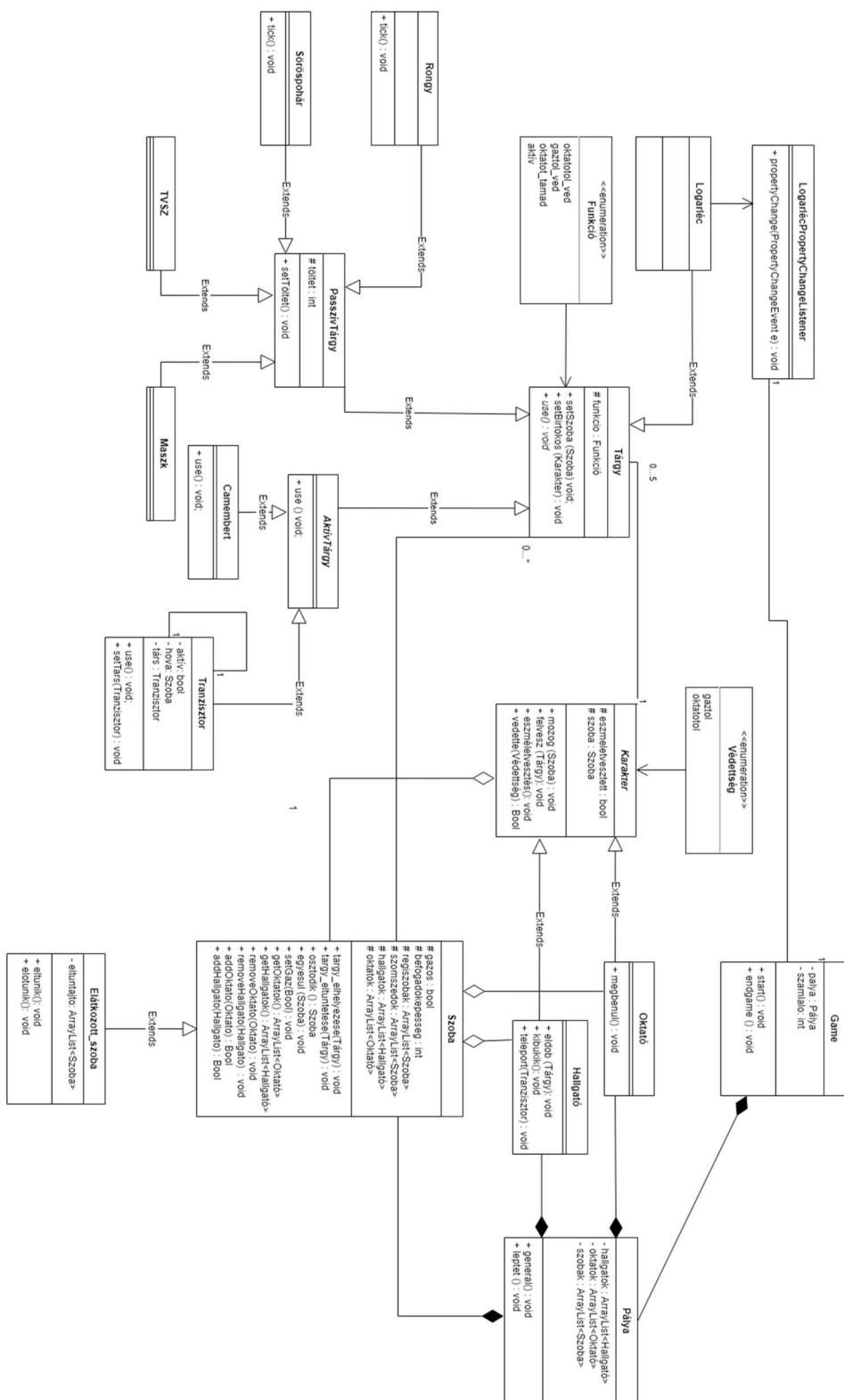
4.1.13 Tranzisztor

Egy tárgy, ami szobákból felvehető. Ennek használatával képesek hallgatók szobák közt ajtó nélkül közlekedni

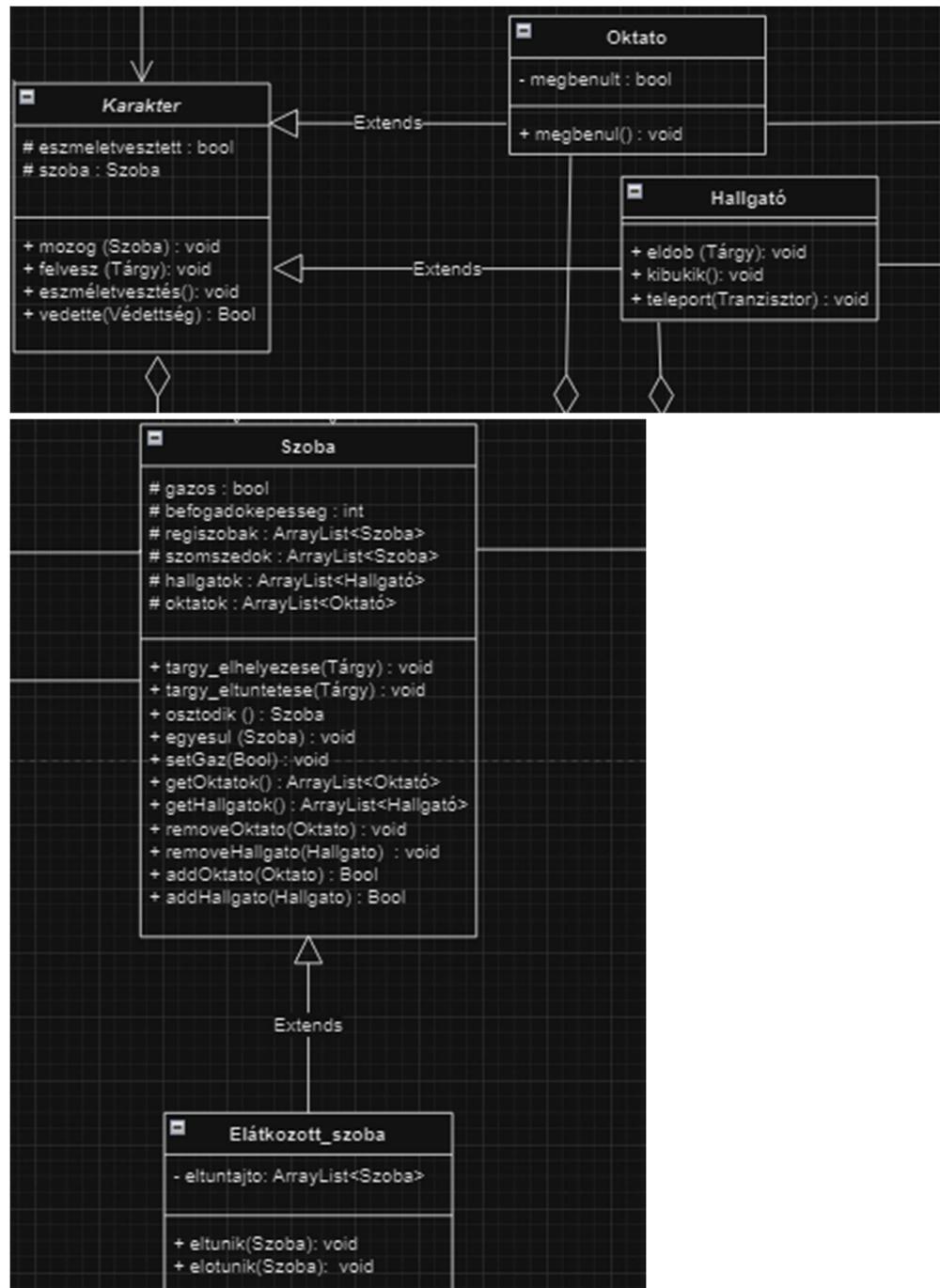
4.1.14 TVSZ

Egy tárgy, ami szobákból felvehető. Ez védelmet nyújt az oktatók ellen

4.2 Statikus struktúra diagramok



változtatás:



4.3 Osztályok leírása

4.3.1 AktívTárgy

- **Felelősség**

Egy absztrakt osztály, amely az aktív tárgyat reprezentálja. Ezek azok a tárgyak, amelyeket egyes esetekben használnak a karakterek.

- **Ősosztályok**

- Tárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- **Metódusok**

- void use(): Ez az a függvény, amelyet minden aktív tárgynak meg kell valósítania, hogy pontosan mit csinál, amikor használják

4.3.2 Camembert

- **Felelősség**

A feladat leírásában lévő Dobozolt káposztás camembert tárgyat reprezentálja. Ennek használatával egy szoba átalakítható gázos szobává.

- **Ősosztályok**

- AktívTárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- **Metódusok**

- void use(): A paraméterben átadott szobában elhasználódik, ennek köszönhetően az a szoba gázos lesz. Ez után a tárgy eltűnik.

4.3.3 Elátkozott _ szoba

- **Felelősség**

A szobának egy olyan fajtájáért felel aminek vannak a szobán túlmutató tulajdnságai, ugyanis lehetnek eltűnő és előtűnő ajtói.

- **Ősosztályok**

Szoba

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- ArrayList<Szoba> eltuntajto : eltároljuk az eltűnt ajtókat, hogy később vissza tudjuk állítani.

- **Metódusok**

- void eltunik() : eltüntet egy ajtót, így azon nem lehet átmenni amíg nem kerül elő újra.
- void elotunik() : előtűntet egy korábban eltűnt ajtót, így azon újra át lehet menni.

4.3.4 Game

- **Felelősség**

Ez felel a program indulásáért és megállásáért.

- **Ősosztályok**

- **Interfészek**

- **Asszociációk**

LogarlécPropertyChangeListener : ez fog szólni, hogy a logarléc fel lett véve és vége a játéknak.

- **Attribútumok**

- Pálya palya : a kezdeti pálya
- int szamlalo : ez adja meg milyen hosszú a játék, ha a leteltéig nem sikerül megtalálni a logarlécet akkor a hallgatók vesztettek.

- **Metódusok**

- void start() : a függvény ami indítja és zárja a játékot.
- void endgame() : befejezi a játékot

4.3.5 Hallgató

- **Felelősség**

Ez az osztály felel az irányítható karakterért a hallgató tulajdonságaiért.

- **Ősosztályok**

Karakter

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- **Metódusok**

- void eldob(Tárgy) : a hallgató saját akaratból is dobhat el tárgyat, ennek a függvénynek a segítségével.
- void kibukik() : ha a hallgató kibukik, akkor megszűnik létezni és a tárgyai megsemmisülnek.

4.3.6 Karakter

- **Felelősség**

Ez az absztrakt osztály felel a karakterek a működéséért. Ez kezeli azokat az eseményeket amik egy karakterrel történhetnek, a tárgyfelvétel és letétel, mozgás és eszméletvesztés.

- **Ősosztályok**

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- Szoba szoba : a szoba ahol a karakter tartózkodik.
- bool eszméletvesztett : a karakter állapotát írja le, hogy épp cselekvőképes-e

- **Metódusok**

- void mozog(Szoba) : a karakter mozgását végzi
- void felvesz(Tárgy) : felvesz egy tárgyat és eltárolja a táskában.
- void eszméletvesztés() : ez végzi az eszméletvesztéskor beteljesülő eseményeket, a tárgy elvesztését és a kimaradást.
- bool vedette(Védettség) : visszaadja , hogy a játékos védett-e éppen a paraméterben megadott veszély ellen
- void teleport(Tranzisztor) : a paraméterben megadott tranzisztorral a párjához teleportálunk

4.3.7 Logarléc

- **Felelősség**

Ennek a tárgynak felvételével a hallgatók megnyerik a játékot.

- **Ősosztályok**

- Tárgy

- **Interfészek**

- **Asszociációk**

- LogarlécPropertyChangeListener osztállyal azért, hogy lássa a Listener, hogy valaki felvette a Logarlécet.

- **Attribútumok**

- **Metódusok**

4.3.8 LogarlécPropertyChangeListener

- **Felelősség**

Ez az osztály figyeli azt, hogy felveszi-e valamelyik hallgató a Logarlécet.

- **Ősosztályok**

- **Interfészek**

- **Asszociációk**

- Game osztállyal azért, hogy ha valaki felveszi a Logarlécet, meghívassa az endgame() függvényét.

- **Attribútumok**

- **Metódusok**

void propertyChange(ChangeEvent e): Ha egy hallgató felveszi a Logarlécet (megváltozik a birtokos attribútum), akkor meghívja a Game endgame() függvényét.

4.3.9 Maszk

- **Felelősség**

A feladat leírásában lévő FFP2es maszk tárgyat reprezentálja. Megvédi ezt a tárgyat birtokló karaktereket az eszméletvesztéstől a gázos szobákban.

- **Ősosztályok**

- PasszívTárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- int töltet: Ez számlálja azt, hogy az adott maszk hányszor használható még. Ha ez letelik, akkor a tárgy már nem funkcionál.

- **Metódusok**

- void use(): Amikor a játékos gázos szobába lép, akkor használja ezt a tárgyat, ezzel a töltete eggyel csökken, viszont nem veszti eszméletét és folytathatja a játékot.

4.3.10 Oktató

- **Felelősség**

Ez az osztály felel a játék által irányított karakter, az oktató tulajdonságaiért.

- **Ősosztályok**

Karakter

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- **Metódusok**

- void megbénul(): ha van megfelelő tárgy a hallgatónál és egy szobába kerülnek akkor az oktató megbénul és nem tud kibuktatni hallgatót.

4.3.11 PasszívTárgy

- **Felelősség**

Egy osztály, amely azokat tárgyakat reprezentálja, amelyek egyes szituációkban hatnak és emiatt nem “aktiválhatóak”.

- **Ősosztályok**

- Tárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- int töltet: Ez azt adja meg, hogy mennyi ideig használható még a tárgy

- **Metódusok**

- void setTöltet(): töltet attribútum csökkentő függvénye

4.3.12 Pálya

- **Felelősség**

Ez az osztály felel a játék generálásáért, valamint az oktatók mozgatásáért.

- **Ősosztályok**

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- ArrayList<Hallgató> hallgatok: összes hallgató listája
- ArrayList<Oktató> oktatok: összes oktató listája
- ArrayList<Szoba> szobák: összes szoba listája

- **Metódusok**

- void general() : legenerálja a pályát, vagyis a szobákat, azoknak a különböző típusait, valamint a játékosokat és tárgyakat.
- void leptet() : lép az oktatókkal, valamint a meghívja a változtatásokat intéző függvényeket.

4.3.13 Rongy

- **Felelősség**

A feladat leírásában lévő Nedves táblatörlő rongy tárgyat reprezentálja. Ennek segítségével megbéníthatók az oktatók, ha egy szobában tartózkodnak egy olyan hallgatóval, amelynél van egy ilyen tárgy.

- **Ősosztályok**

- PasszívTárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- **Metódusok**

- void tick() : Ez a függvény számolja az időt egy timer segítségével, amivel lehet majd csökkenteni a tárgy töltetét.

4.3.14 Söröspohár

- **Felelősség**

A feladat leírásában lévő Szent söröspohár tárgyat reprezentálja. Ez egy ideig képes védelmet nyújtani az oktatók ellen a hallgatóknak.

- **Ősosztályok**

- PasszívTárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- **Metódusok**

- void tick() : Ez a függvény számolja az időt egy timer segítségével, amivel lehet majd csökkenteni a tárgy töltetét.

4.3.15 Szoba

- **Felelősség**

Ez az osztály felel a szobának a működéséért. Itt fog történni a játék, itt fognak mozogni a karakterek.

- **Ősosztályok**

- **Interfészek**

- **Asszociációk**

Tárgy : a szobában található tárgyak.

- **Attribútumok**

- bool gazos : megmutatja, hogy gázos-e a szoba
- int befogadokepesseg : hány fő lehet a szobában
- ArrayList<Tárgy> targyak : a szobában található tárgyak listája
- ArrayList<Szoba> regiszerek : tárolja a szobákat amiből összeolvadt egy szoba, hogy az alapján szét is tudjon osztódni.
- ArrayList<Szoba> szomszedok : szomszédos szobák listája
- ArrayList<Hallgató> hallgatok: a szobában tartózkodó hallgatók
- ArrayList<Oktatók> oktatok: a szobában tartózkodó oktatók

- **Metódusok**

- void targy_elhelyezese() : elhelyez egy tárgyat a szobában amit fel lehet venni.
- void targy_eltuntetese(Tárgy) : eltüntet egy tárgyat, hogy ne lehessen felvenni többször, ha egy játékos felveszi.
- Szoba osztodik() : visszaad egy új szobát ami kiosztódik a szobából
- void egyesul(Szoba) : egyesíti a paraméterben levő szobával a jelenlegi szobát
- void setGaz(Bool) : beállítja a gazos attribútum értékét, vagyis, hogy egy szoba gázos-e vagy nem
- ArrayList<Oktatók> getOktatók() : visszaadja a szobában tartózkodó oktatók listáját
- ArrayList<Hallgató> getHallgatók() : visszaadja a szobában tartózkodó hallgatók listáját
- void removeOktato() : kitöröl egy oktatót a listából
- void removeHallgato() : kitöröl egy hallgatót a listából
- bool addOktato(Oktató) : hozzáad egy oktatót a listához
- bool addHallgato(Hallgató) : hozzáad egy hallgatót a listához

4.3.16 Tárgy

- **Felelősség**

Egy absztrakt osztály, amely a tárgyakat reprezentálja.

- **Ősosztályok**

- **Interfészek**

- **Asszociációk**

- Karakter osztályal azért, hogy tudja, melyik karakter birtokolja ezt a tárgyat.
- Szoba osztályal azért, hogy tudja, melyik szobában van a tárgy

- **Attribútumok**

- Szoba hol: Ez azt írja le, hogy a tárgy melyik szobában van, ha ez egy karakternél van ez a tárgy, akkor is változik a karaktertől függően.
- Karakter birtokos: Ez azt adja meg, hogy melyik karakternél van a tárgy, ha szobában van, null
- Funkció funkció: Ez tárolja el hogy a tárgy milyen funkciót tölt be

- **Metódusok**

- void setSzoba(Szoba): szoba attribútum setttere
- void setBirtokos(Karakter): tárgy birtokosának setttere
- void use(): absztrakt függvény, amely a tárgyak használatát írja le minden tárgynak

4.3.17 Tranzisztor

- **Felelősség**

A feladat leírásában lévő Tranzisztor tárgyat reprezentálja. Ennek segítségével képesek a hallgatók két tranzisztor összekapcsolása után egyik nem szomszédos szobából átmenni a másikba, ajtók használata nélkül.

- **Ősosztályok**

- AktívTárgy

- **Interfészek**

- **Asszociációk**

- Saját magával, tudja, melyik tranzisztorral van összekapcsolva

- **Attribútumok**

- bool aktív: Ez azt írja le, hogy a tranzisztor aktív-e
- Szoba hova: Ez azt tárolja, hogy ha össze van kötve két tranzisztor, akkor melyikbe visz át a használatakor

- **Metódusok**

- void use(): Első használatkor aktívvá teszi a tranzisztorokat, és az elsőt leteszi. Második használatkor leteszi a másikat a jelenlegi szobába és átmegy az első letett tranzisztorhoz, és a második aktivitását kikapcsolja. Ez után további használatokkor ha aktív a tranzisztor átmegy a karakter a párjához, és a kiindulót be, a másikat pedig beaktiválja.
- void setTars(Tranzisztor): beállítja a tranzisztort, amellyel össze van kötve

4.3.18 TVSZ

- **Felelősség**

A feladat leírásában levő TVSZ denevérbőrre nyomtatott példányát reprezentálja. Megvédi a hallgatót a vele egy szobában lévő oktató általi kibuktatástól.

- **Ősosztályok**

Tárgy

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

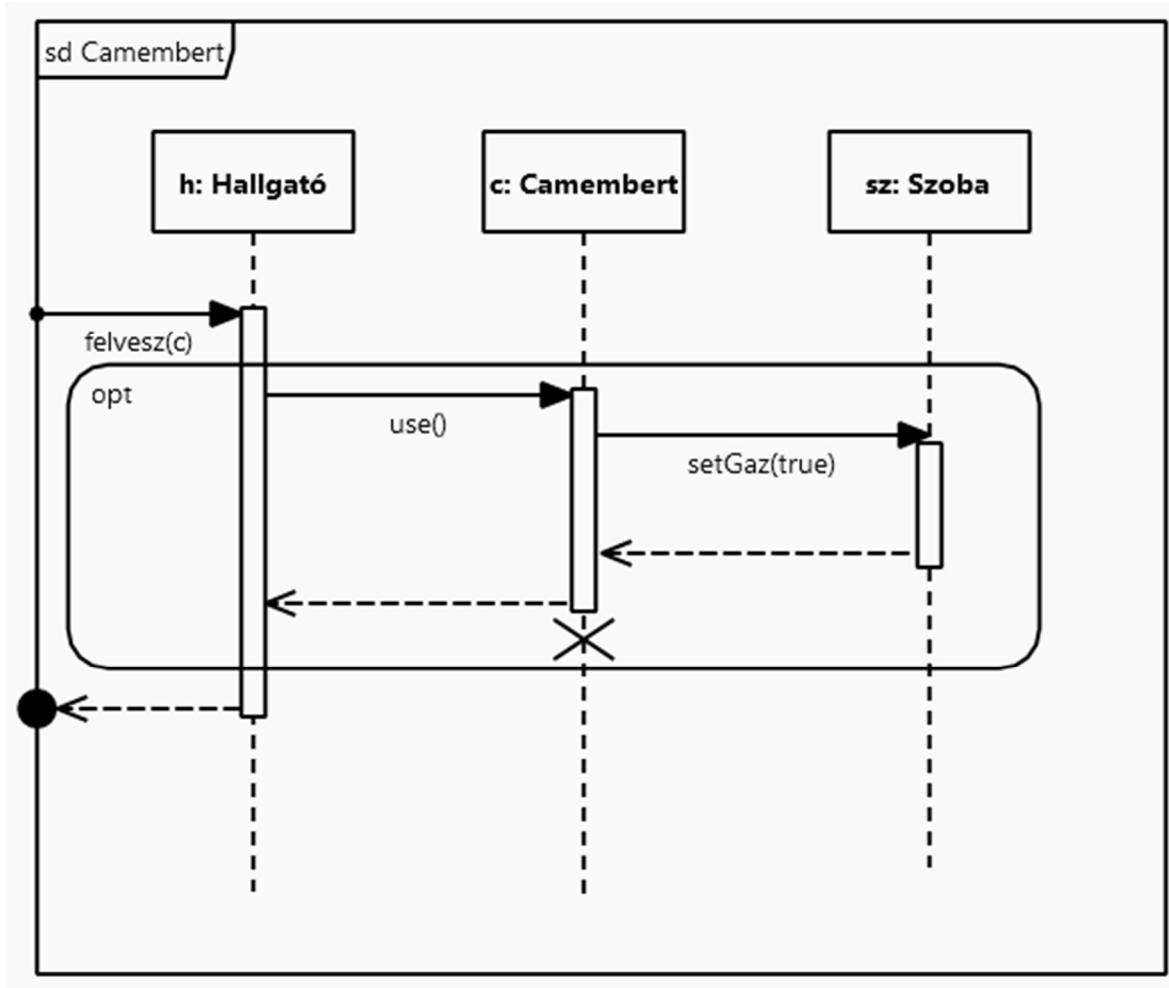
- int töltet: Ez számlálja azt, hogy az adott TVSZ hányszor használható még.

- **Metódusok**

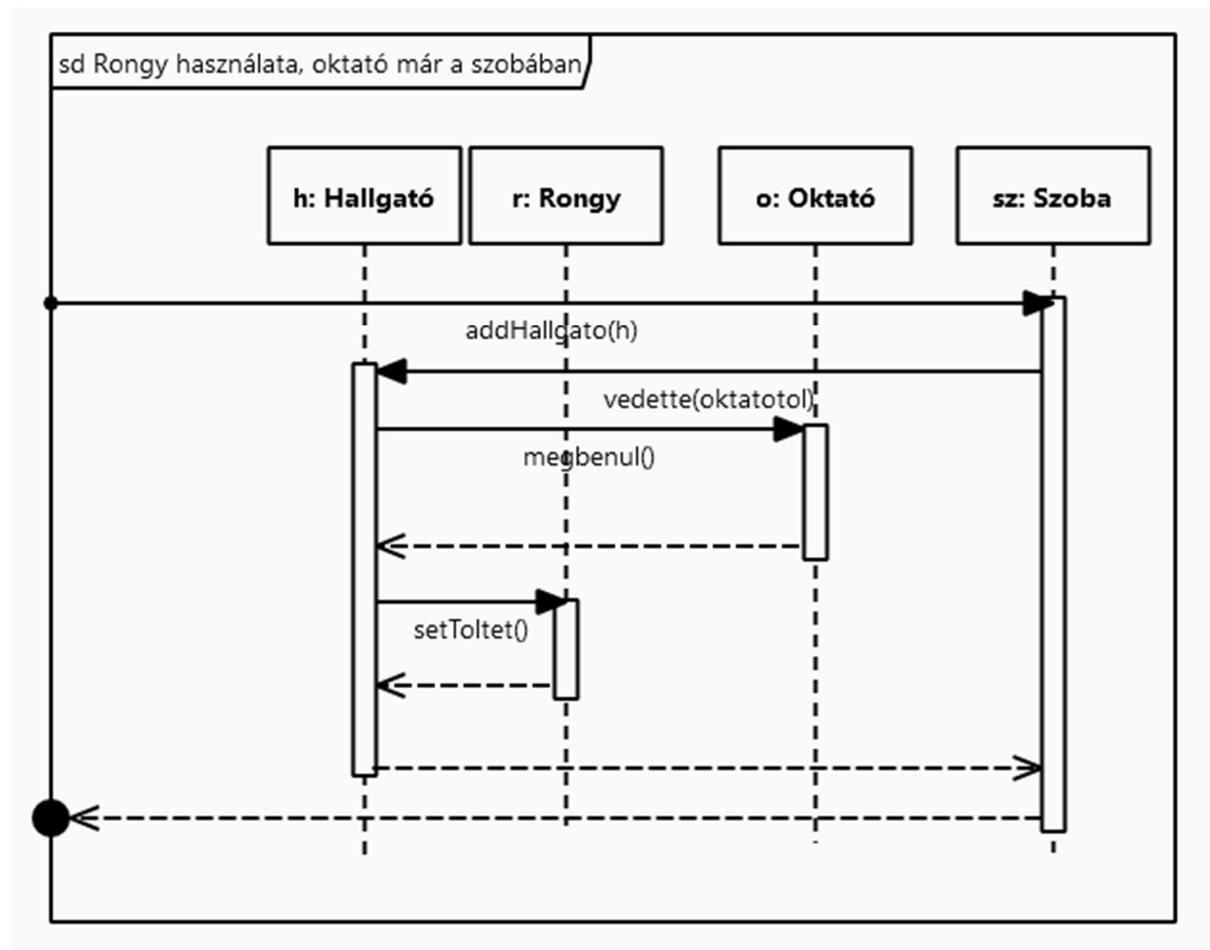
- void use(): Amikor a játékos egy szobába lép, amelyben oktató is tartózkodik, akkor használja ezt a tárgyat, ezzel a töltete eggyel csökken, viszont folytathatja a játékot.

4.4 Szekvencia diagramok

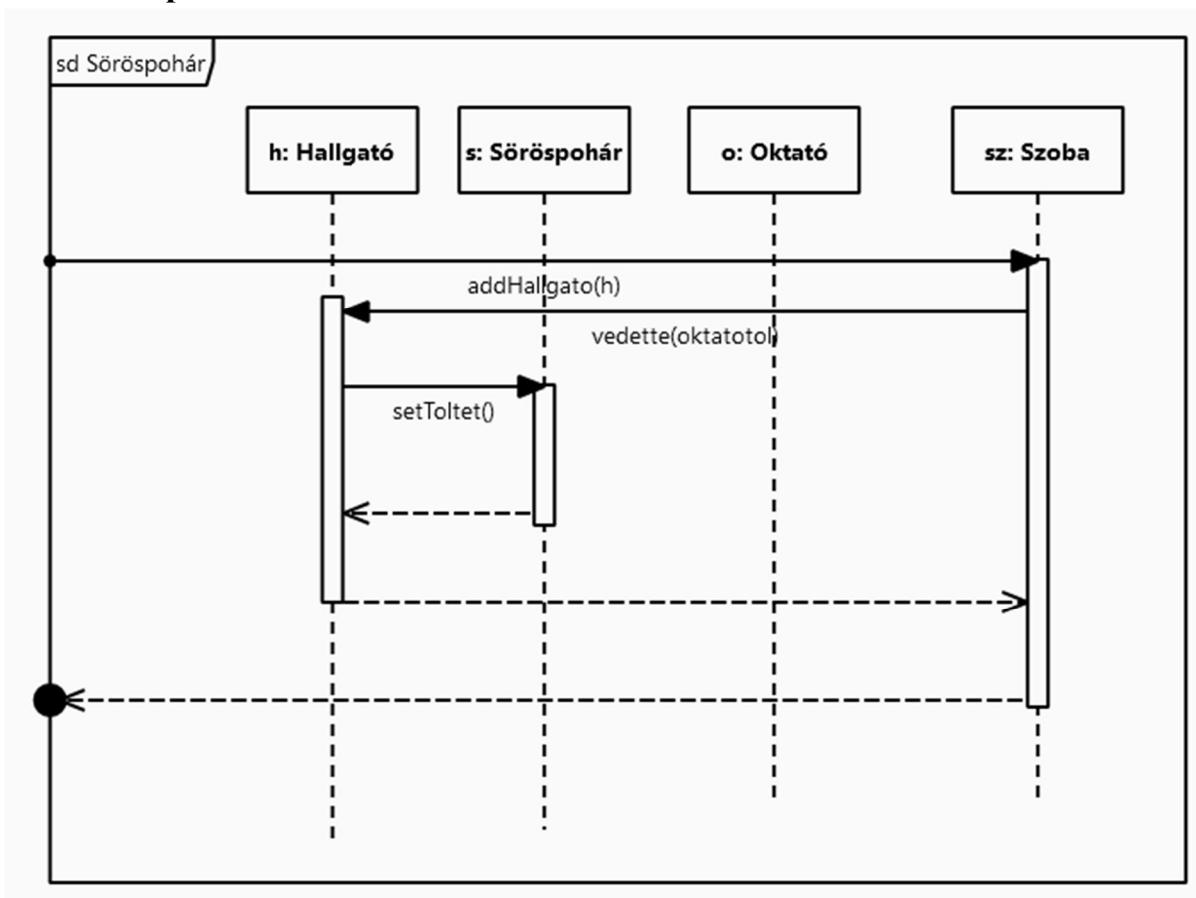
4.4.1 Káposztás camembert használata



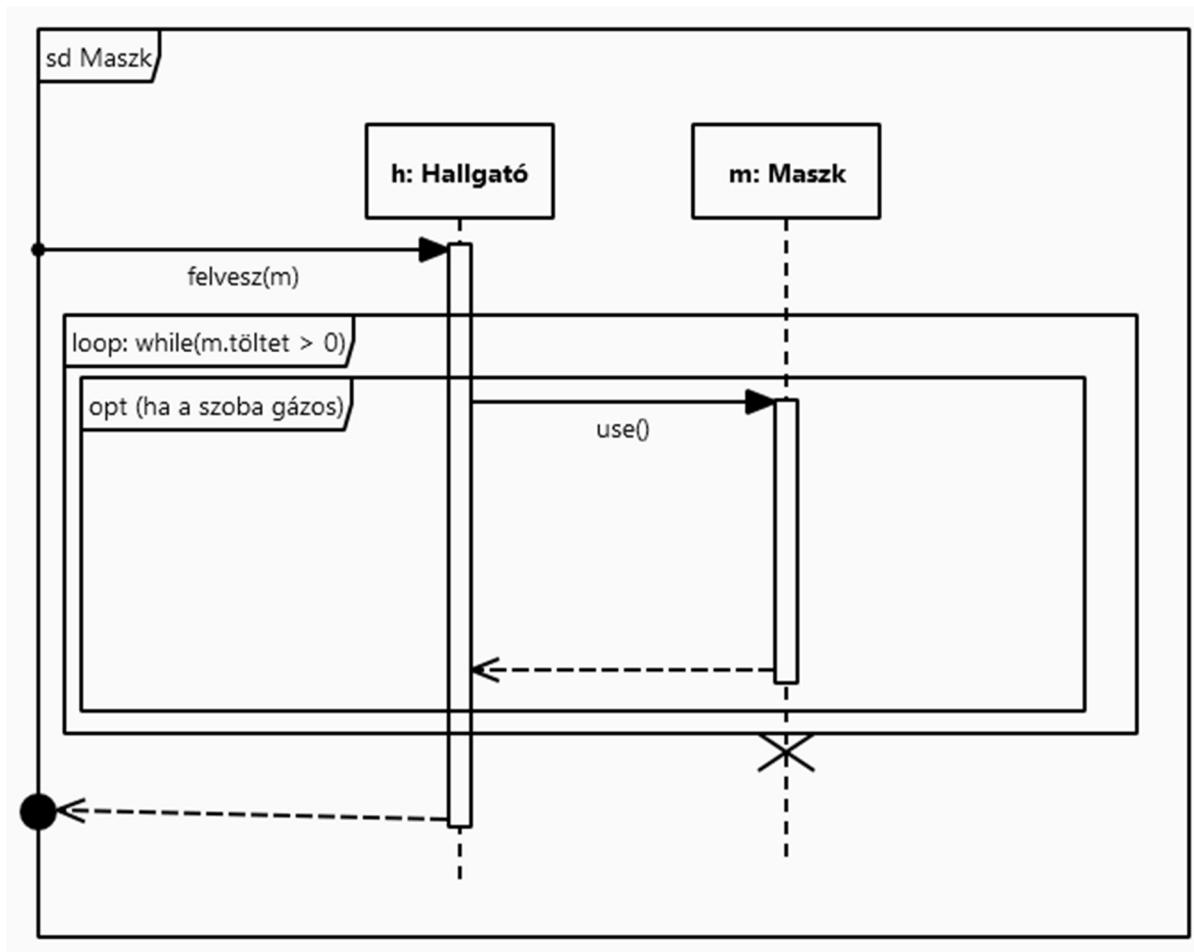
4.4.2 Rongy használata



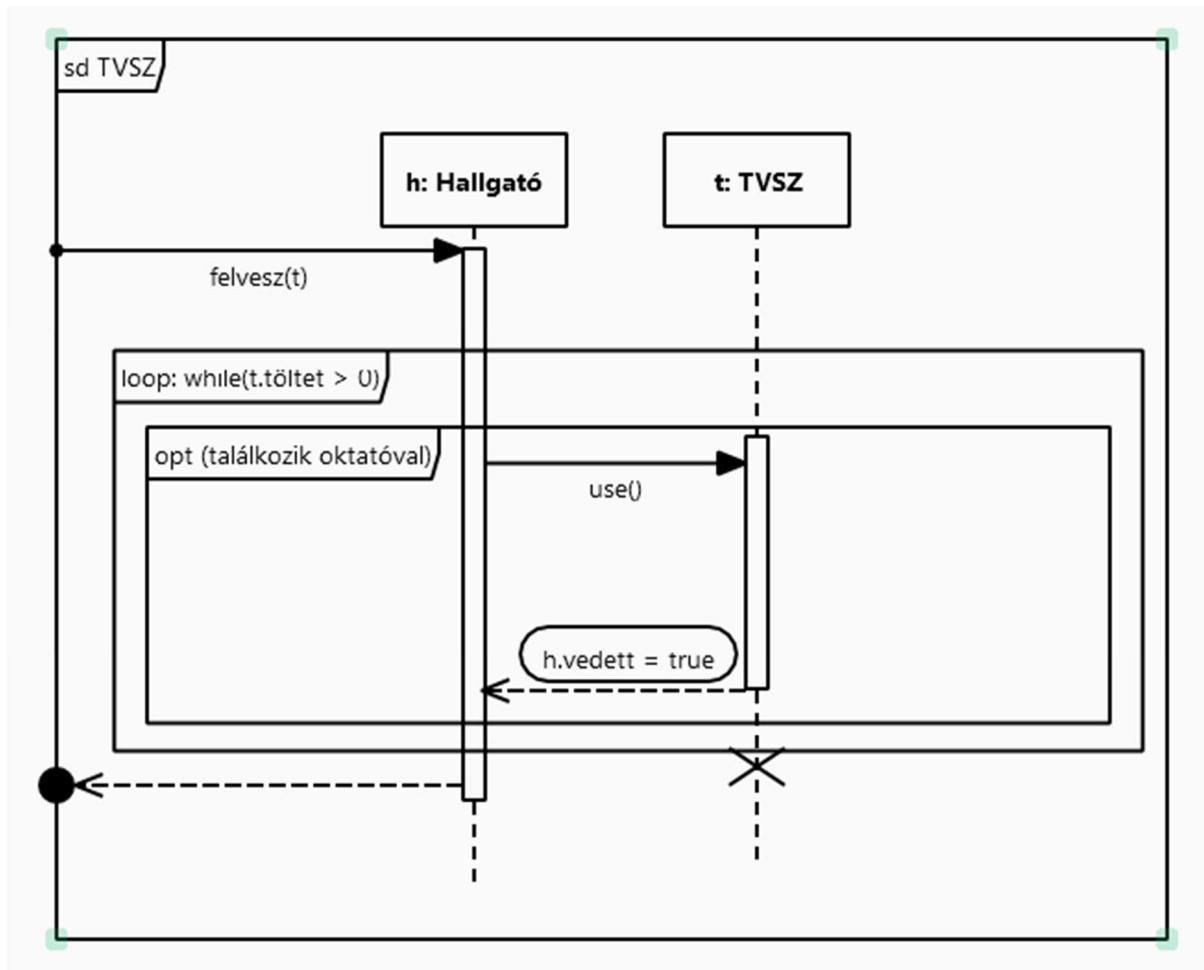
4.4.3 Söröspohár használata



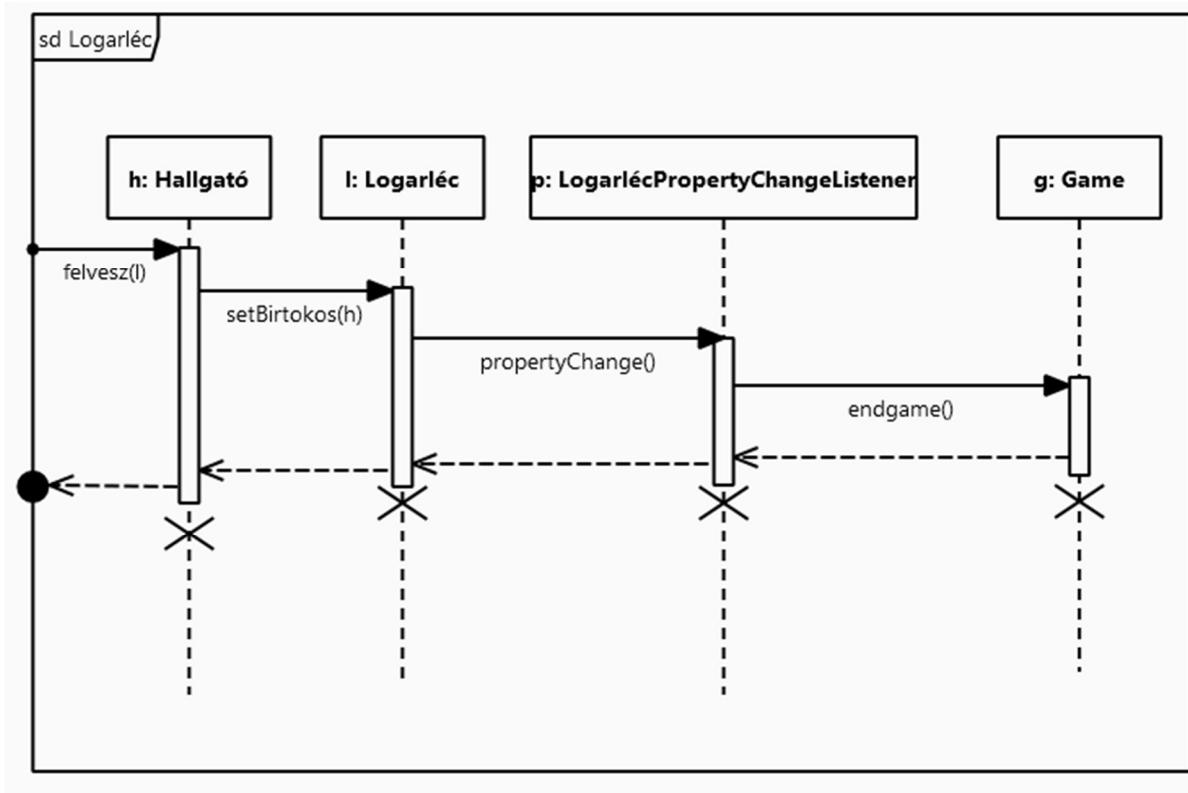
4.4.4 FFP2-es maszk használata



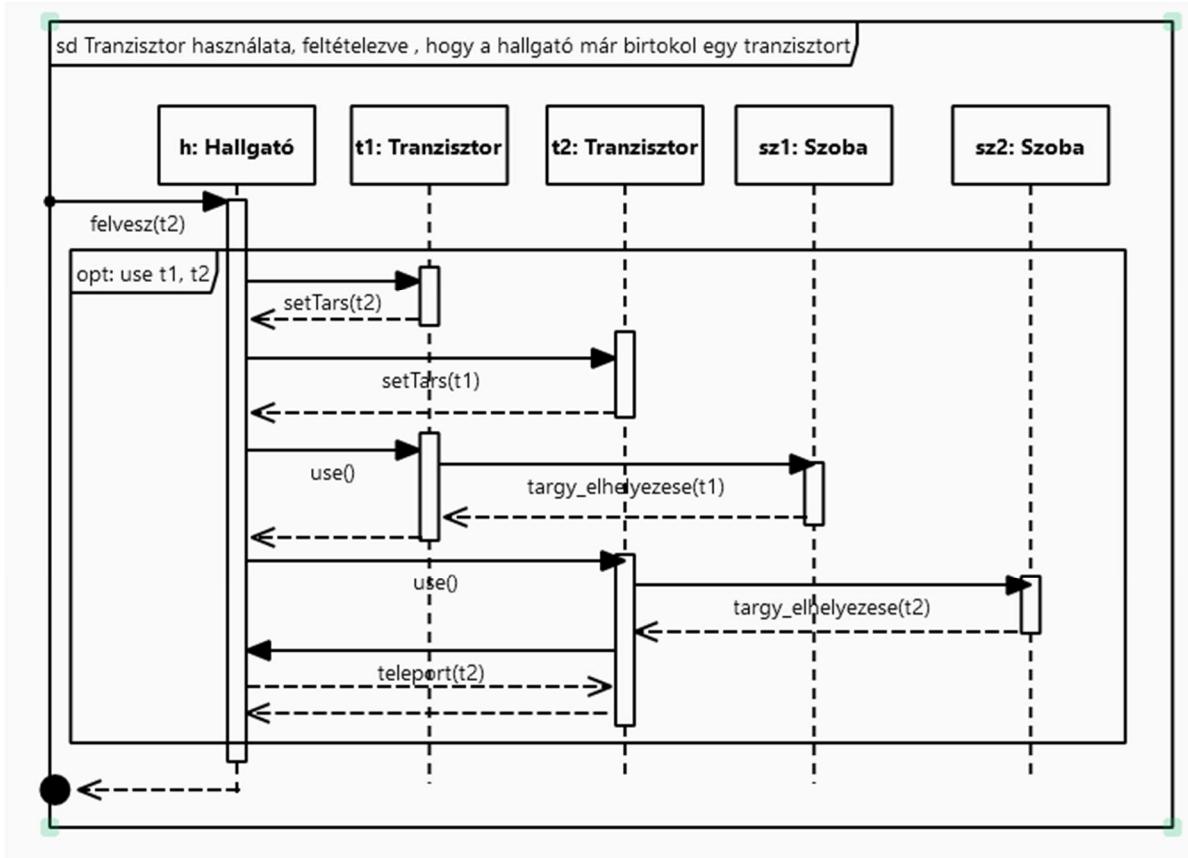
4.4.5 TVSZ használata



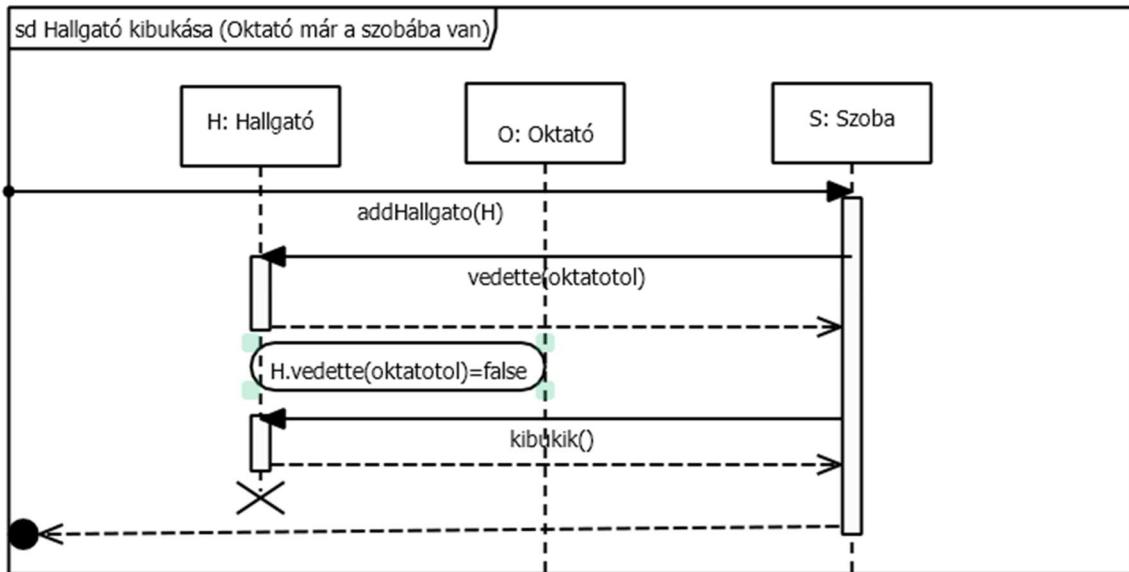
4.4.6 Logarléc



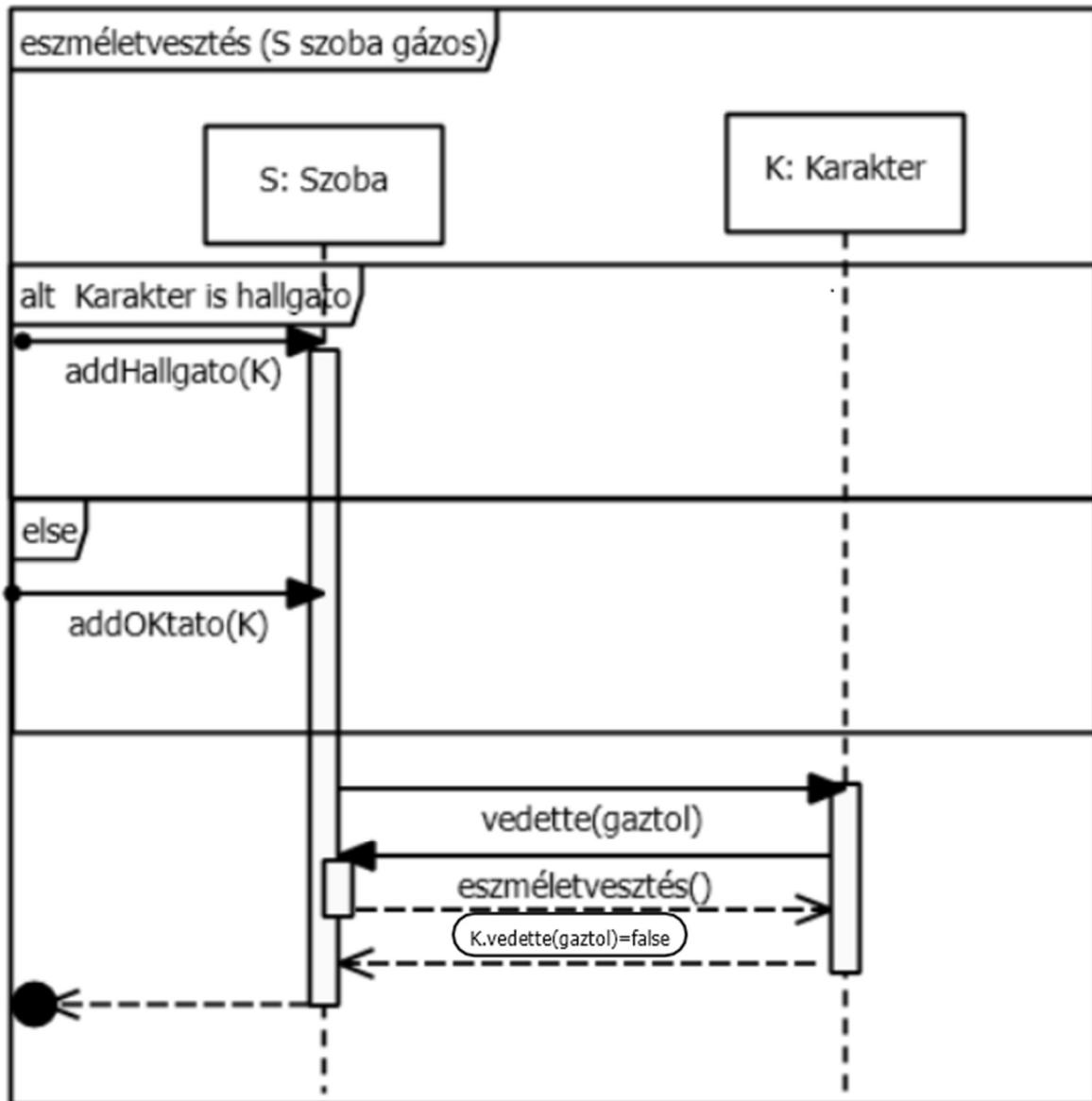
4.4.7 Tranzisztor használata



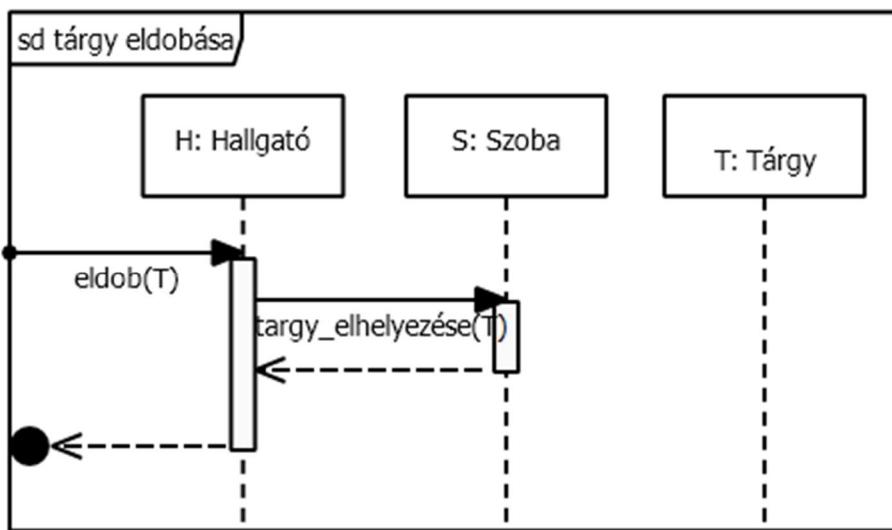
4.4.8 Hallgató kibukása



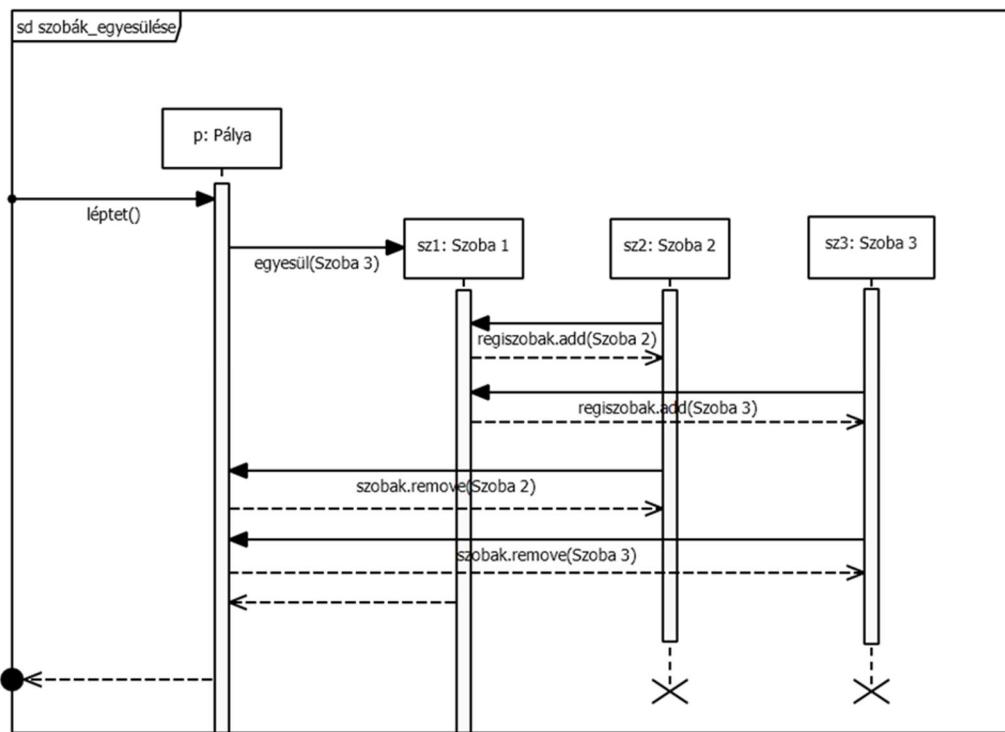
4.4.9 Karakterek eszméletvesztése



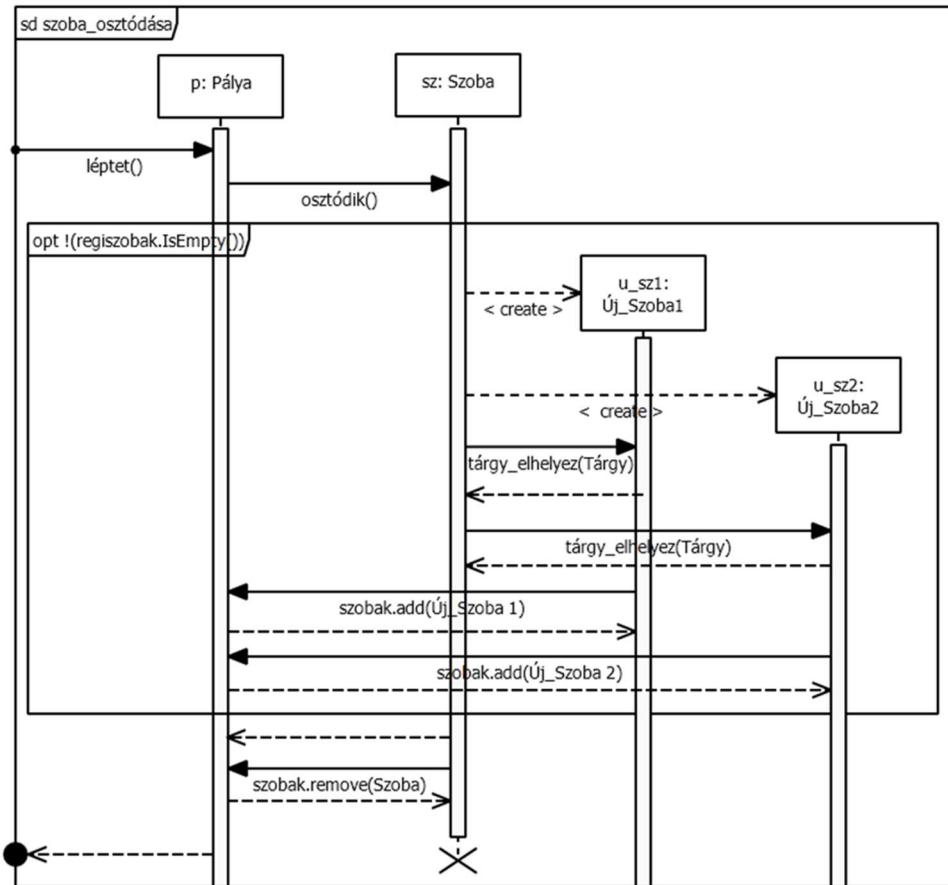
4.4.10 Eldobás



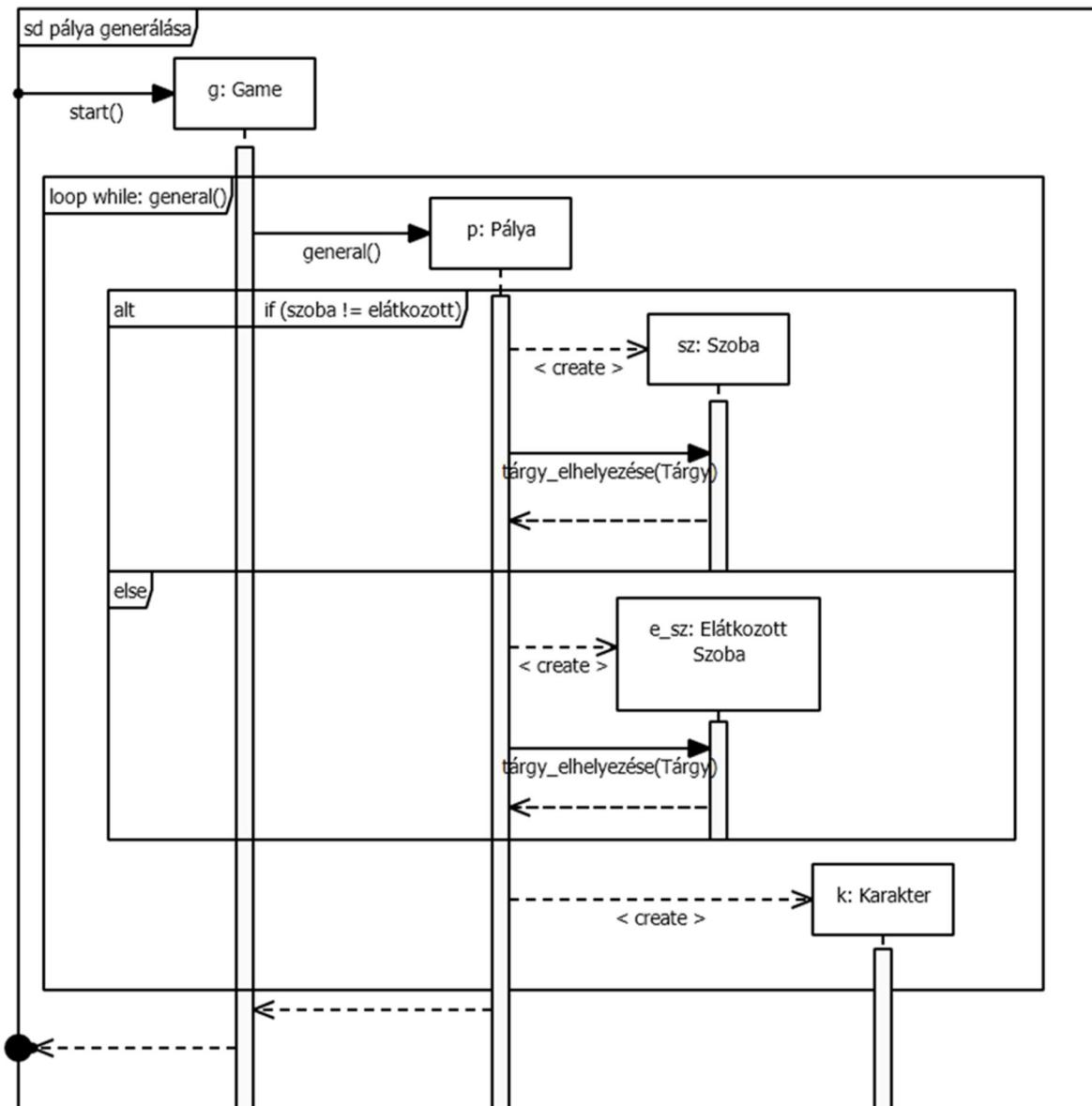
4.4.11 Szobák egyesülése



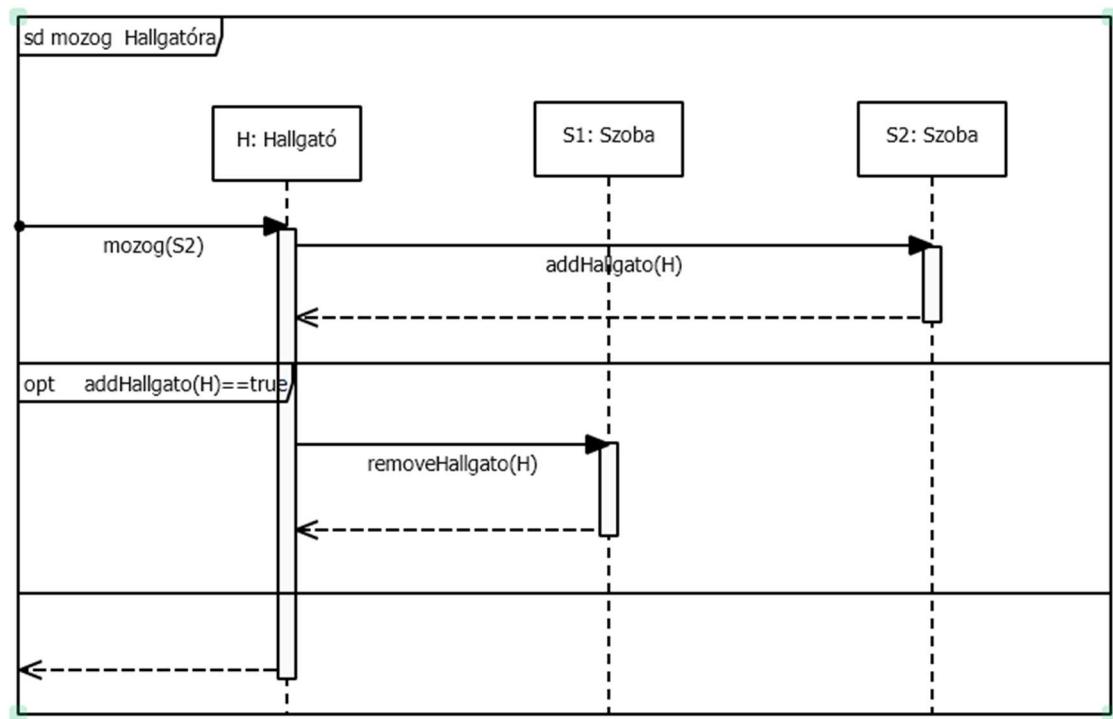
4.4.12 Szobák osztódása



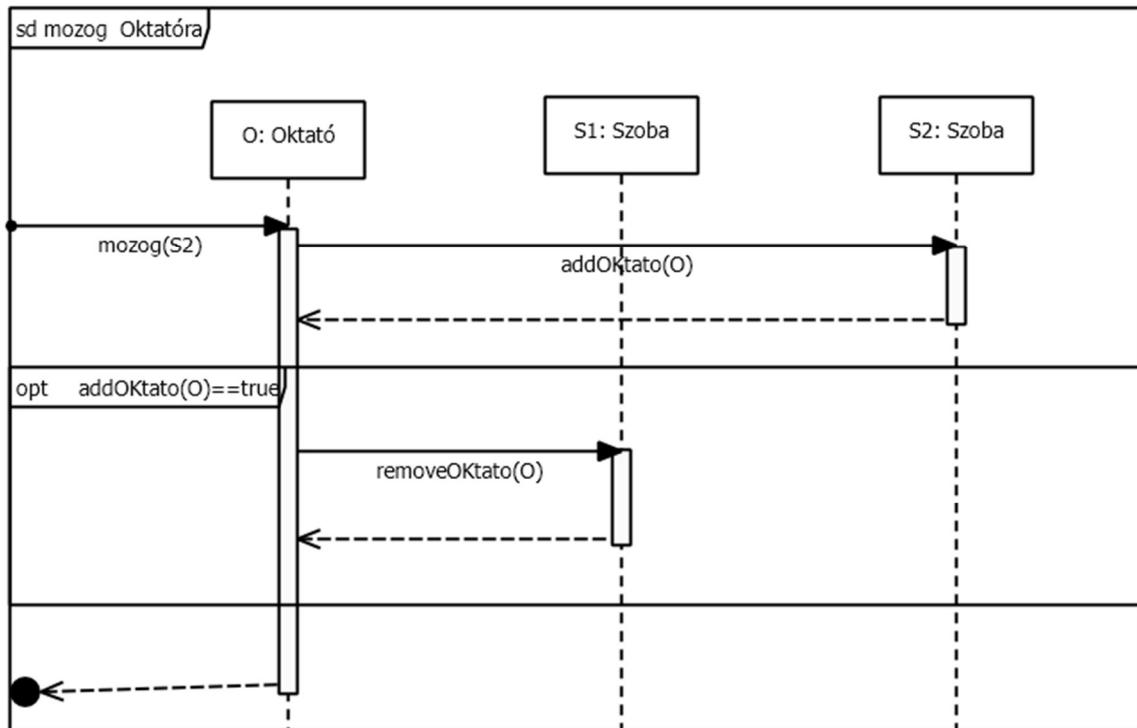
4.4.13 Pálya generálása



4.4.14 Mozgás hallgatóra

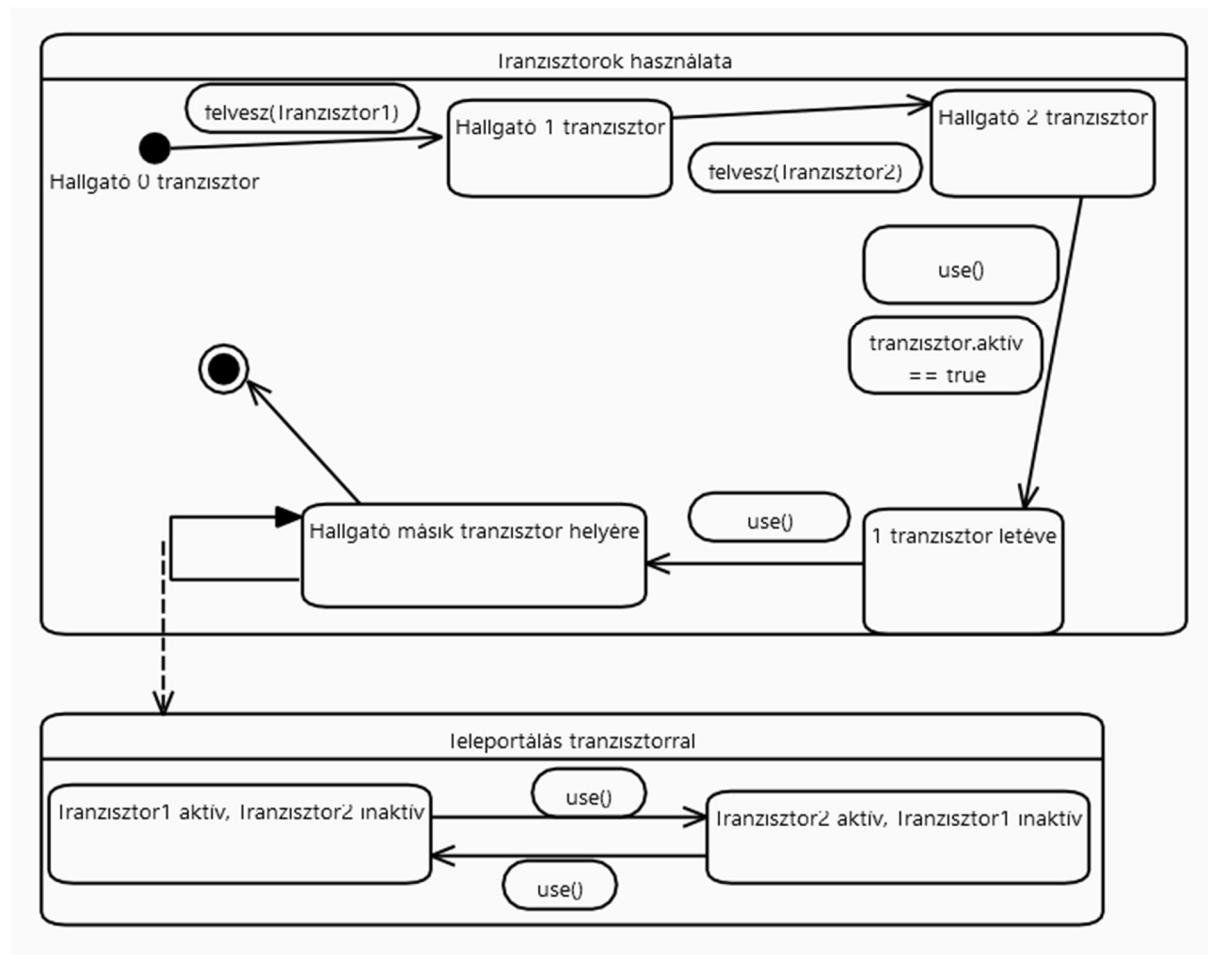


4.4.15 Mozgás oktatóra

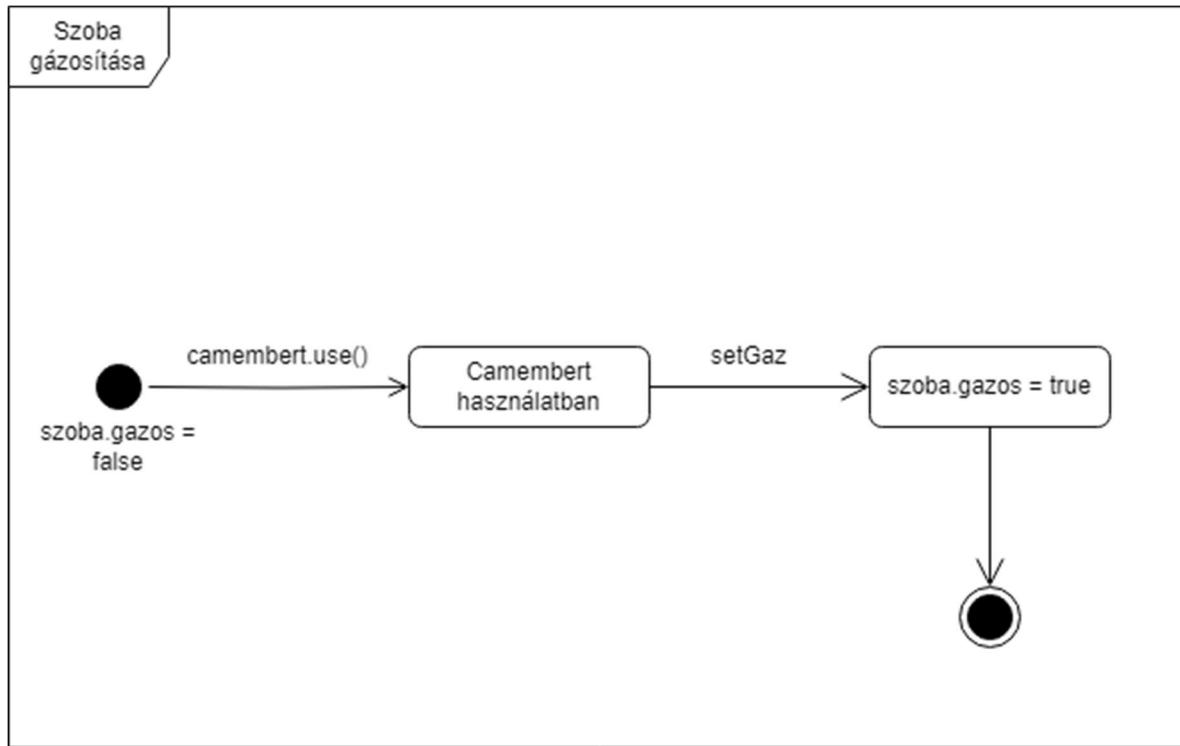


4.5 State-chartok

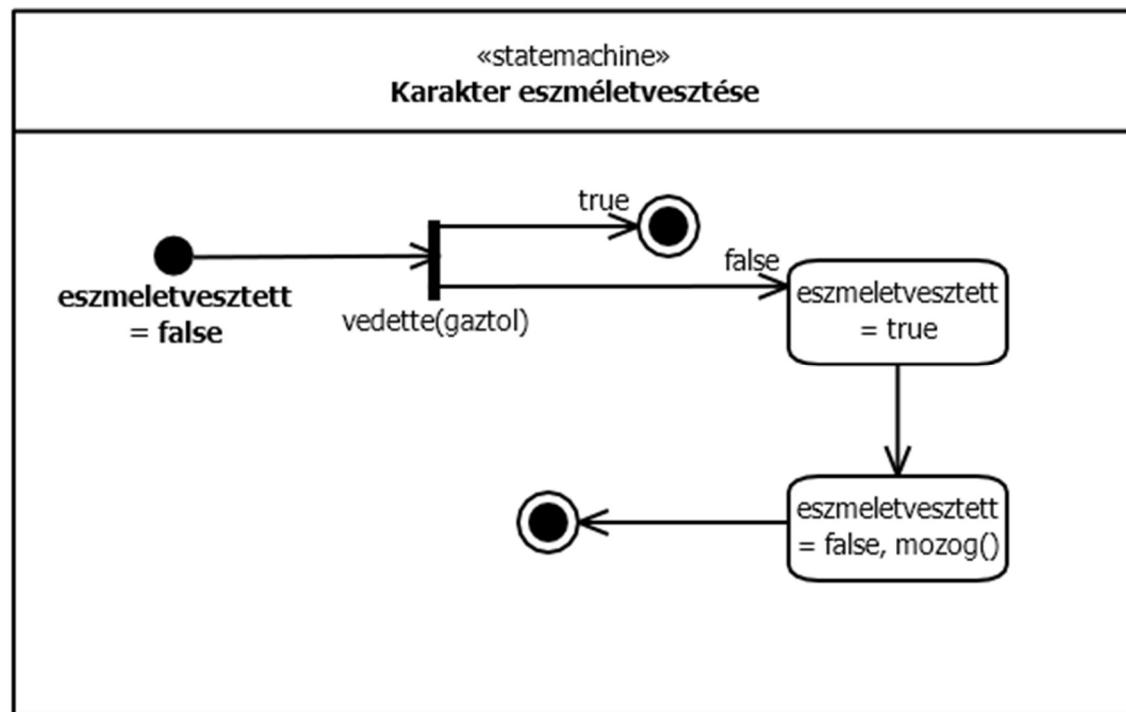
4.5.1 Tranzisztor használata



4.5.2 Szoba gázosítása



4.5.3 Karakter eszméletvesztése



4.6 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2024.03.06 21:00	1 óra	Csapat	Értekezlet. Döntések: heti teendők megbeszélése, felosztása.
2024.03.07. 20:00	3 óra	Csapat	Értekezlet. Döntések: Class diagram javítása
2024.03.09.13:00	1 óra	Czotter	Szekvencia diagramok javítása
2024.03.10.20:00	1,5 óra	Németh	Szekvencia diagramok javítása
2024.03.10. 22:00	2 óra	Csapat	Értekezlet. Döntések: Class diagram javítása
2024.03.10 9:00	2 óra	Csapat	Értekezlet. Döntések: Class diagram javítása, és végső megbeszélés
2024.03.11. 11:00	1 óra	Schulcz	Szekvencia diagramok javítása
2024.03.11. 11:00	1 óra	Hermann	Osztályok leírásának javítása, anyag formázása
2024.03.11. 11:00	1,5 óra	Tarsoly	Osztályok leírásának javítása, karakter state-chart

Szkeleton tervezése

35 – pupakok

Konzulens:
Koczó Attila

Csapattagok:

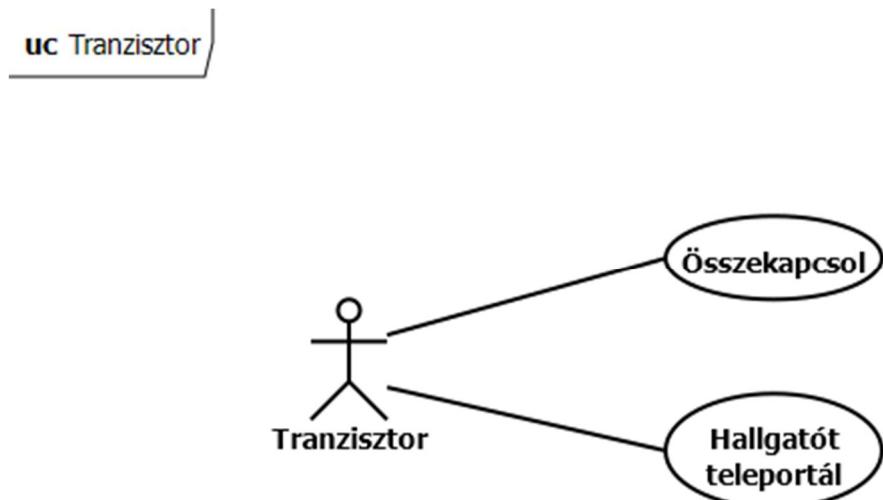
Tarsoly Levente	E1IK75	tarsolyl@edu.bme.hu
Németh Gergely	TA9G0R	gergo.h.nemeth@gmail.com
Schulcz Gábor	EP3I9R	schulcz.gabor6@gmail.com
Czotter Benedek	TFB4FY	czottibeni@gmail.com
Hermann Máté Béla	T83K5I	mate3829@gmail.com

2024.03.18

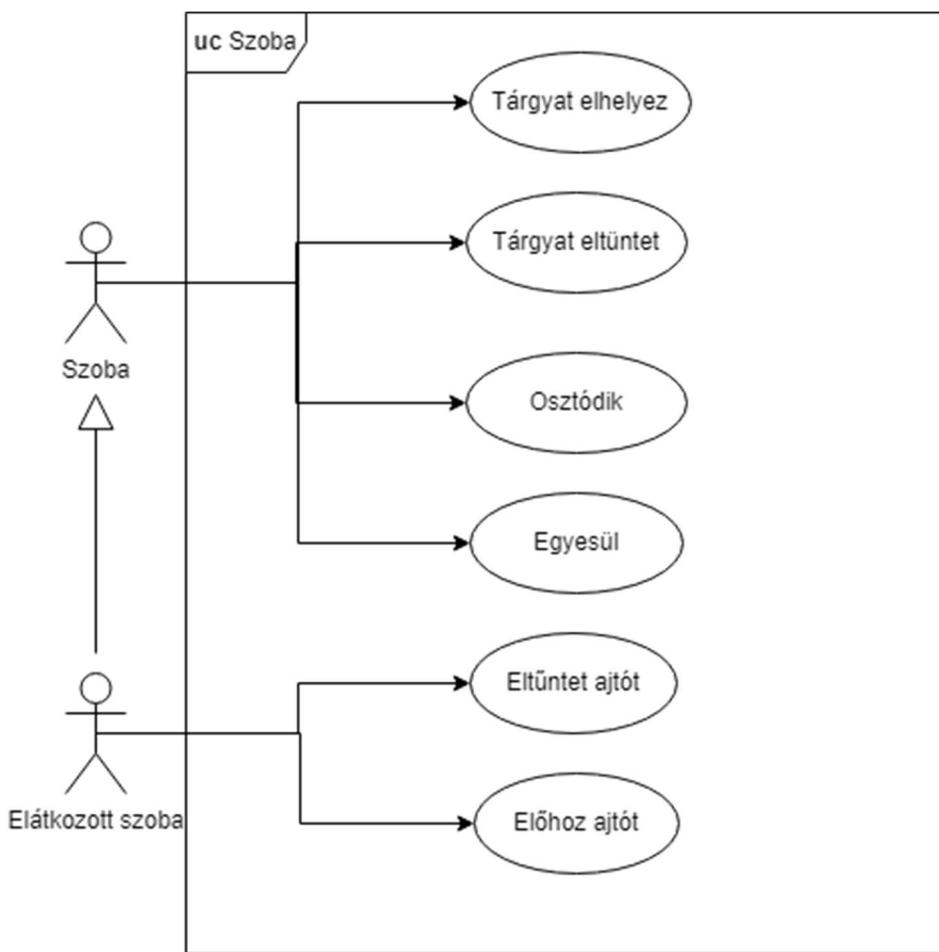
5.1 A szkeleton modell valóságos use-case-ei

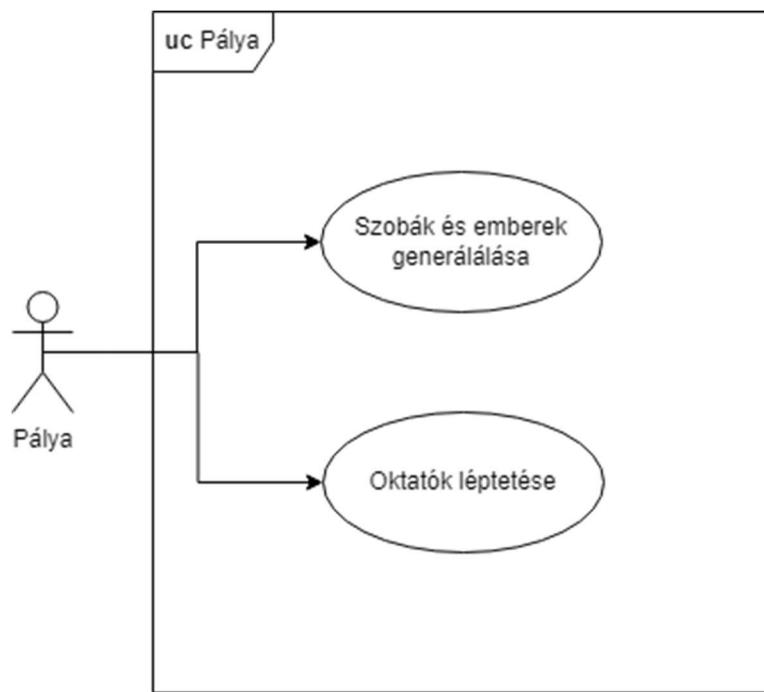
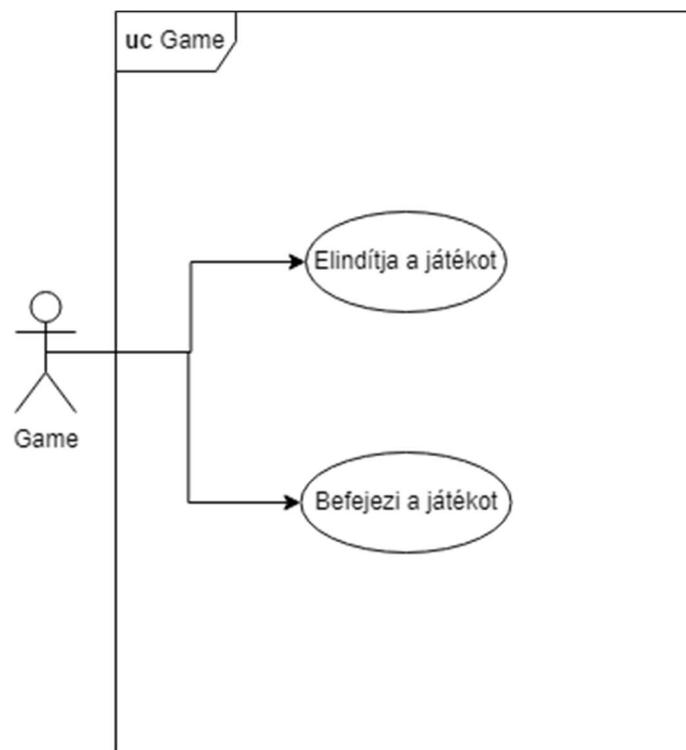
5.1.1 Use-case diagram

5.1.1.1 Tranzisztor

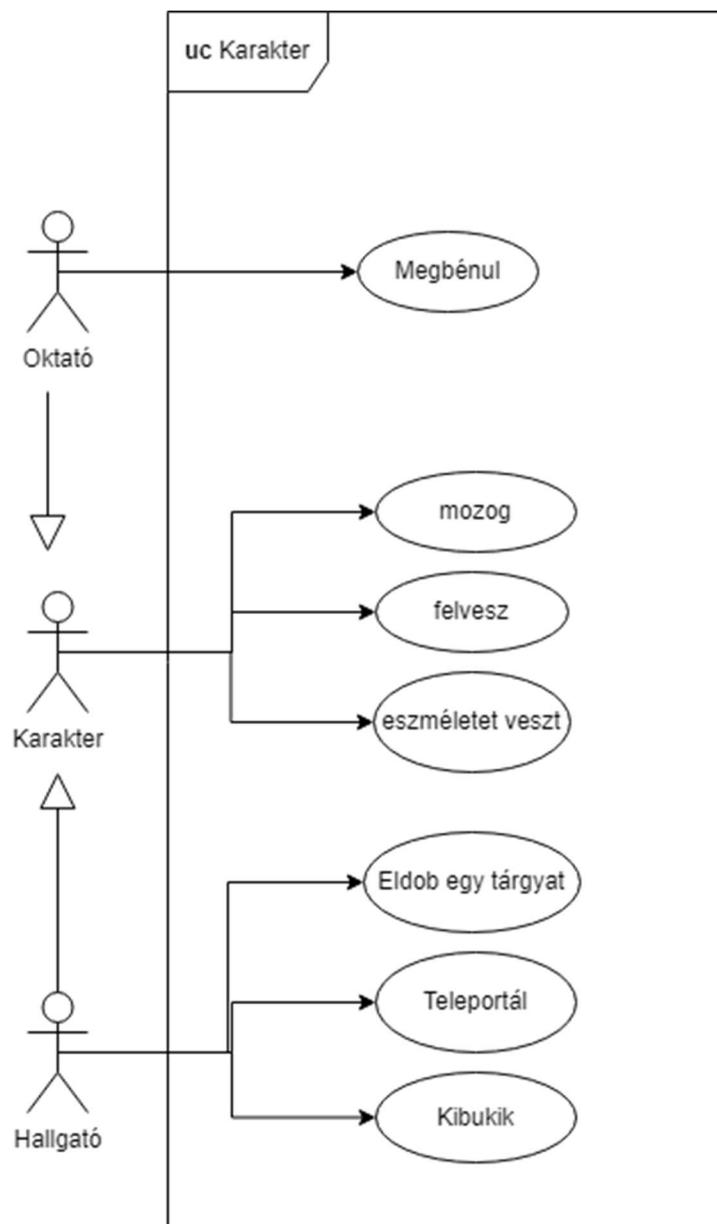


5.1.1.2 Szoba



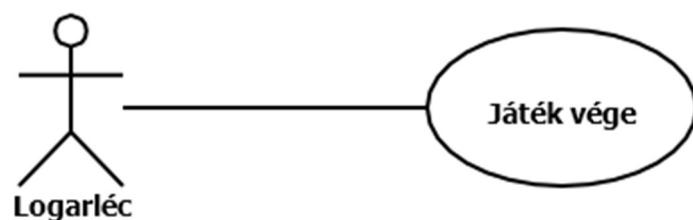
5.1.1.3 Pálya**5.1.1.4 Game**

5.1.1.5 Karakterek



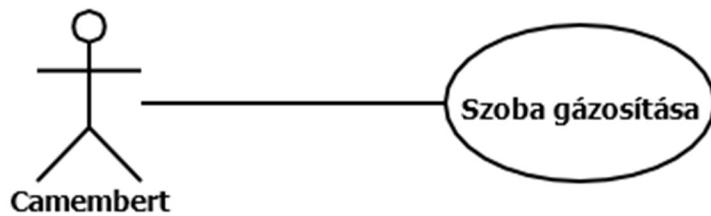
5.1.1.6 Logarléc

uc Logarléc



5.1.1.7 Camembert

uc Camembert



5.1.2 Use-case leírások

Use-case neve	TVSZ felvétele
Rövid leírás	A karakter felvesz egy TVSZ-t
Aktorok	Karakter
Forgatókönyv	<ol style="list-style-type: none"> 1. A karakter felveszi a TVSZ-t. 2. A TVSZ a karakter táskjába kerül. 3. A TVSZ kikerül a szoba nyilvántartásából.

Use-case neve	Camembert felvétele
Rövid leírás	A karakter felvesz egy camembert
Aktorok	Karakter
Forgatókönyv	<ol style="list-style-type: none"> 1. A karakter felveszi a camembert. 2. A camembert a karakter táskjába kerül. 3. A camembert kikerül a szoba nyilvántartásából.

Use-case neve	Logarléc felvétele
Rövid leírás	A karakter felvesz egy logarlécet.
Aktorok	Karakter
Forgatókönyv	<ol style="list-style-type: none"> 1. A karakter felveszi a logarlécet. 2. A logarléc a karakter táskjába kerül. 3. A logarléc kikerül a szoba nyilvántartásából.

Use-case neve	Tranzisztor felvétele
Rövid leírás	A karakter felvesz egy tranzisztort
Aktorok	Karakter
Forgatókönyv	<ol style="list-style-type: none"> 1. A karakter felveszi a tranzisztort. 2. A tranzisztort a karakter táskjába kerül. 3. A tranzisztort kikerül a szoba nyilvántartásából.

Use-case neve	Söröspohár felvétele
Rövid leírás	A karakter felvesz egy söröspoharat
Aktorok	Karakter
Forgatókönyv	<ol style="list-style-type: none"> 1. A karakter felveszi a söröspoharat. 2. A söröspohár a karakter táskjába kerül. 3. A söröspohár kikerül a szoba nyilvántartásából.

Use-case neve	Rongy felvétele
Rövid leírás	A karakter felvesz egy rongyot
Aktorok	Karakter
Forgatókönyv	<ol style="list-style-type: none"> 1. A karakter felveszi a rongyot.. 2. A rongy a karakter táskjába kerül. 3. A rongy kikerül a szoba nyilvántartásából.

Use-case neve	Maszk felvétele
Rövid leírás	A karakter felvesz egy maszkot
Aktorok	Karakter
Forgatókönyv	<ol style="list-style-type: none"> 1. A karakter felveszi a maszkot. 2. A maszk a karakter táskjába kerül. 3. A maszk kikerül a szoba nyilvántartásából.

Use-case neve	Camembert használata
Rövid leírás	A hallgató használja a camembertet, ezzel elgázosítja a szobát.
Aktorok	Hallgató
Forgatókönyv	<ol style="list-style-type: none"> 1. Hallgató használja a camembertet. 2. Camembert eltűnik a hallgató táskjából. 3. A szoba gázos lesz.

Use-case neve	Tranzisztorok összekapcsolása
Rövid leírás	Két tranzisztor összekapcsolása, későbbi teleportáláshoz.
Aktorok	Hallgató
Forgatókönyv	<ol style="list-style-type: none"> 1. Mindkét tranzisztornak a másik lesz a társa.

Use-case neve	Tranzisztor használata
Rövid leírás	A hallgató a már összekapcsolt tranzisztorokat lehelyezi.
Aktorok	Hallgató
Forgatókönyv	<ol style="list-style-type: none"> 1. A hallgató lehelyezi a tranzisztorokat.

Use-case neve	Teleportálás tranzisztorral
Rövid leírás	A hallgató a már lehelyezett tranzisztorokkal teleportál
Aktorok	Hallgató
Forgatókönyv	<ol style="list-style-type: none"> 1. Hallgató egyik tranzisztor szobájából a másikba kerül. 2. Amelyik tranzisztortól megy az inaktív lesz, ahova megy az pedig aktív.

Use-case neve	Játék vége logarléc felvétele után
Rövid leírás	Hallgató felveszi a logarlécet, ezzel megnyerve a játékot.
Aktorok	Hallgató, Game
Forgatókönyv	<ol style="list-style-type: none"> 1. Hallgató felveszi a logarlécet. 2. Logarléc birtokosa megváltozik. 3. Game osztály értesül a játék befejezéséről.

Use-case neve	Ajtó eltüntetése
Rövid leírás	Egy elátkozott szoba képes ajtókat eltüntetni
Aktorok	Elátkozott szoba
Forgatókönyv	<ol style="list-style-type: none"> 1. Az elátkozott szoba eltűnt ajtajai közé adódik a szoba 2. Az elátkozott szoba törli a szomszédai közül a szobát, és a szoba szomszédai közül is törlődik az elátkozott szoba

Use-case neve	Ajtó előtűnése
Rövid leírás	Egy elátkozott szoba képes eltűnt ajtókat újra megjeleníteni a karaktereknek
Aktorok	Elátkozott szoba
Forgatókönyv	<ol style="list-style-type: none"> 1. Az elátkozott szoba hozzáadja a szomszédai közé a szobát, és a szoba szomszédai közé is adódik az elátkozott szoba 2. Az elátkozott szoba eltűnt ajtajai közül törlődik a szoba

Use-case neve	Hallgató kibuktatása
Rövid leírás	Az oktató kibuktatja a hallgatót az egyetemről
Aktorok	Oktató
Forgatókönyv	<ol style="list-style-type: none"> 1. A hallgató egy szobába kerül a hallgatóval <ol style="list-style-type: none"> a. Ha van védelmet adó tárgya, nem történik semmi b. Ha nincs, kiesik a játékból

Use-case neve	Oktató megbénítása
Rövid leírás	Megbénít adott időre egy oktatót a hallgató
Aktorok	Hallgató
Forgatókönyv	<ol style="list-style-type: none"> 1. A hallgató megbénítja az oktatót 2. Az oktató egy ideig nem mozoghat/tehet semmit

Use-case neve	Eszméletvesztés
Rövid leírás	A gázos szobákban a karakterek eszméletüket veszik
Aktorok	Gázos szoba
Forgatókönyv	<ol style="list-style-type: none"> 1. Karakter gázos szobába lép <ol style="list-style-type: none"> a. Ha van maszkja, nem történik semmi b. Ha nincs eszméletét veszti, eleji tárgyait és nem mozoghat egy ideig

Use-case neve	Játék indítása
Rövid leírás	Elindul a játék
Aktorok	Game
Forgatókönyv	<ol style="list-style-type: none"> 1. A játék létrehozza a pályát 2. Elhelyeződnek a hallgatók, az oktatók és a tárgyak 3. Elkezdődik a játék

Use-case neve	Szoba osztódása
Rövid leírás	A korábban egyesült szobák szétválnak
Aktorok	Pálya, Szoba
Forgatókönyv	<ol style="list-style-type: none"> 1. A pálya szobái kibővülnek a régebbi, korábban egyesült szobákkal 2. A jelenleg megtalálható, egyesült szoba kikrül a pálya szobái közül 3. Elhelyeződnek a megfelelő tárgyak az osztódott szobákban

Use-case neve	Szobák egyesülése
Rövid leírás	A kiválasztott szobák egyesülnek egy új szobába
Aktorok	Pálya, Szoba
Forgatókönyv	<ol style="list-style-type: none"> 1. A pálya szobái közül kikerülnek az egyesülő szobák 2. Az egyesült szoba bekerül a pálya szobái közé 3. Elhelyeződnek a tárgyak az egyesült, új szobában

Use-case neve	Tárgy elhelyezése
Rövid leírás	Az adott tárgy elhelyezésre kerül a pálya egy szobájában
Aktorok	Szoba, Tárgy
Forgatókönyv	<ol style="list-style-type: none"> 1. Beállításra kerül az adott tárgyhoz tartozó szoba, így elhelyezésre kerül a tárgy a szobában

Use-case neve	Tárgy eltűnése
Rövid leírás	A tárgy nem lesz többé megtalálható a szobában, ahol eddig volt
Aktorok	Szoba
Forgatókönyv	<ol style="list-style-type: none"> 1. A szoba nyilvántartásában nem lesz többé megtalálható a tárgy

Use-case neve	Hallgató mozgása
Rövid leírás	Hallgató egyik szobából másikba való lépése
Aktorok	Hallgató, Szoba
Forgatókönyv	<ol style="list-style-type: none"> 1. A Hallgató belép egy másik szobába, ha az nincs teli, ha tele van, akkor nem történik semmi 2. Ha átlépett akkor másik szobából kilép 3. Ha a szoba gázos, akkor annak megfelelően megnézi hogy védette, ha nem akkor elkábul. 4. Megnézi hogy van-e a szobába, olyan oktató, aki nem eszméletvesztett és nem megbénult, ha van akkor megnézi, védette oktatotol és annak megfelelően kibukik vagy nem.

Use-case neve	Oktató mozgása
Rövid leírás	Oktató egyik szobából másikba való lépése
Aktorok	Oktató, Szoba
Forgatókönyv	<ol style="list-style-type: none"> 1. A Oktatóbelép egy másik szobába, ha az nincs teli, ha tele van, akkor nem történik semmi 2. Ha átlépett akkor másik szobából kilép 3. Ha a szoba gázos, akkor annak megfelelően megnézi hogy védette, ha nem akkor elkábul. 4. Ha nem kábul el, akkor megnézi hogy van-e a szobába hallgató, ha van akkor megnézi, védette oktatotol és annak megfelelően kibuktatja vagy nem.

Use-case neve	Oktatók léptetése
Rövid leírás	Oktató mozgatása egyik szobából a másikba
Aktorok	Oktató, Szoba, Pálya
Forgatókönyv	Az oktatóknál megnézi a mely szomszédos szobákba léphet és oda mozgatja.

Use-case neve	Tárgy eldobása
Rövid leírás	A hallgató eldob egy tárgyat saját akaratból
Aktorok	Hallgató, Szoba
Forgatókönyv	<ol style="list-style-type: none"> 1. A hallgató eldob egy nála lévő tárgyat. 2. A tárgy kikerül a nála lévő tárgyak közül. 3. A tárgy megjelenik a szobában.

Use-case neve	Utolsó hallgató is kibukik
Rövid leírás	Az utolsó hallgató is kiesik, ekkor a játéknak vége
Aktorok	Hallgató, Game
Forgatókönyv	<ol style="list-style-type: none"> 1. Az utolsó játékban lévő hallgató is kibukik. 2. A játék véget ér.

Use-case neve	Lejár az idő
Rövid leírás	Ha letelik az idő mielőtt megtalálják a logarlécet akkor a játék véget ér.
Aktorok	Game, Pálya
Forgatókönyv	<ol style="list-style-type: none"> 1. Letelik a játékidő 2. A játék véget ér.

5.2 A szkeleton kezelői felületének terve, dialógusok

A program indításakor kilistázza a végrehajtható eseményeket. A szkeleton bemeneti számok lesznek (ld. 5.2.1), melyek mindegyike egy adott tesztesetet futtat le. A kimeneten szerepelni fog, hogy melyik osztály mely függvénye hívódik meg.

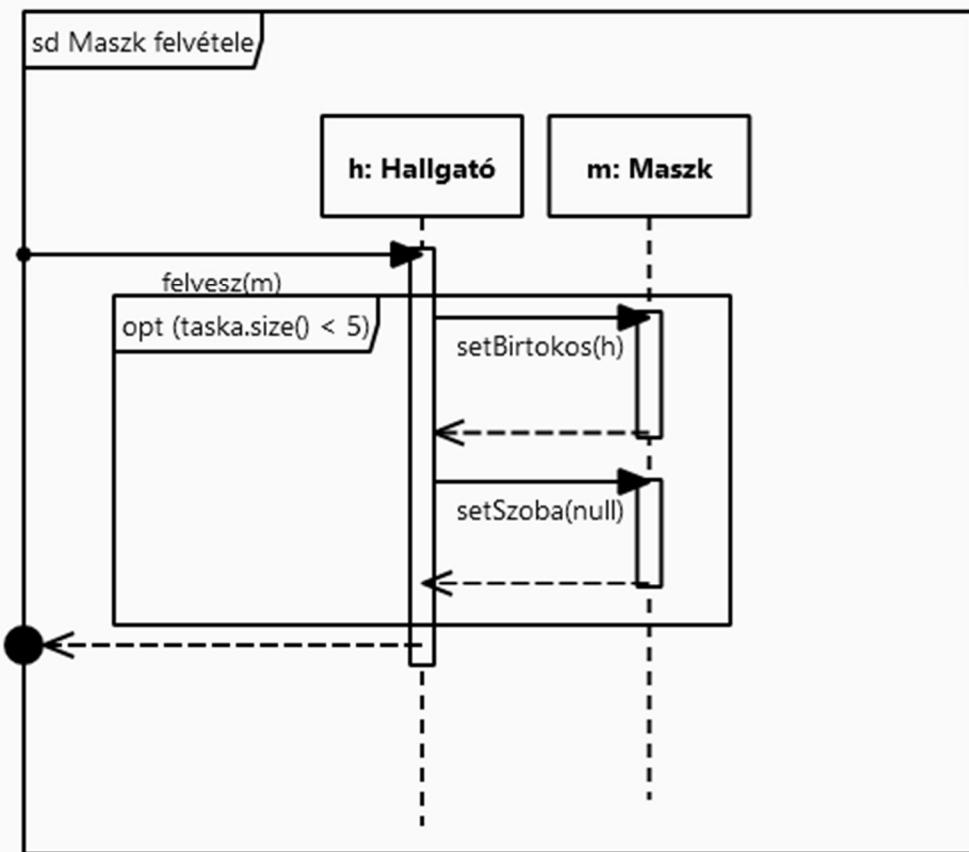
Input	Use-case
0	Játék indítása
1	Oktatók léptetése
2	Szoba osztódása
3	Szoba egyesülése
4	Camembert elhelyezése
5	Camembert eltüntetése
6	Logarléc elhelyezése
7	Logarléc eltüntetése
8	Tranzisztor elhelyezése
9	Tranzisztor eltüntetése
10	TVSZ elhelyezése
11	TVSZ eltüntetése
12	Söröspohár elhelyezése
13	Söröspohár eltüntetése
14	Rongy elhelyezése
15	Rongy eltüntetése
16	Maszk elhelyezése
17	Maszk eltüntetése
18	Hallgató elhelyezése
19	Hallgató eltüntetése
20	Oktató elhelyezése

21	Oktató eltüntetése
22	Elátkozott szobában ajtó eltüntetése
23	Elátkozott szobában ajtó előtűnése
24	Eszméletvesztés
25	Hallgató kibuktatása
26	Oktató megbénítása
27	Oktató mozgása szobába
28	Hallgató mozgása szobába
29	Hallgató camembert felvétele
30	Hallgató Logarléc felvétele
31	Hallgató tranzisztor felvétele
32	Hallgató TVSZ felvétele
33	Hallgató söröspohár felvétele
34	Hallgató rongy felvétele
35	Hallgató maszk felvétele
36	Camembert eldobása
38	Tranzisztor eldobása
39	TVSZ eldobása
40	Söröspohár eldobása
41	Rongy eldobása
42	Maszk eldobása
44	Oktató camembert felvétele
46	Oktató tranzisztor felvétele
47	Oktató TVSZ felvétele
48	Oktató söröspohár felvétele

49	Oktató rongy felvétele
50	Oktató maszk felvétele
51	Camembert használata
52	Tranzisztor összekapcsolása
53	Tranzisztor használata
54	Teleportálás tranzisztor segítségével
55	Logarléc felvétele à játék vége
56	Hallgatók kibukása à játék vége
57	Idő lejár à játék vége

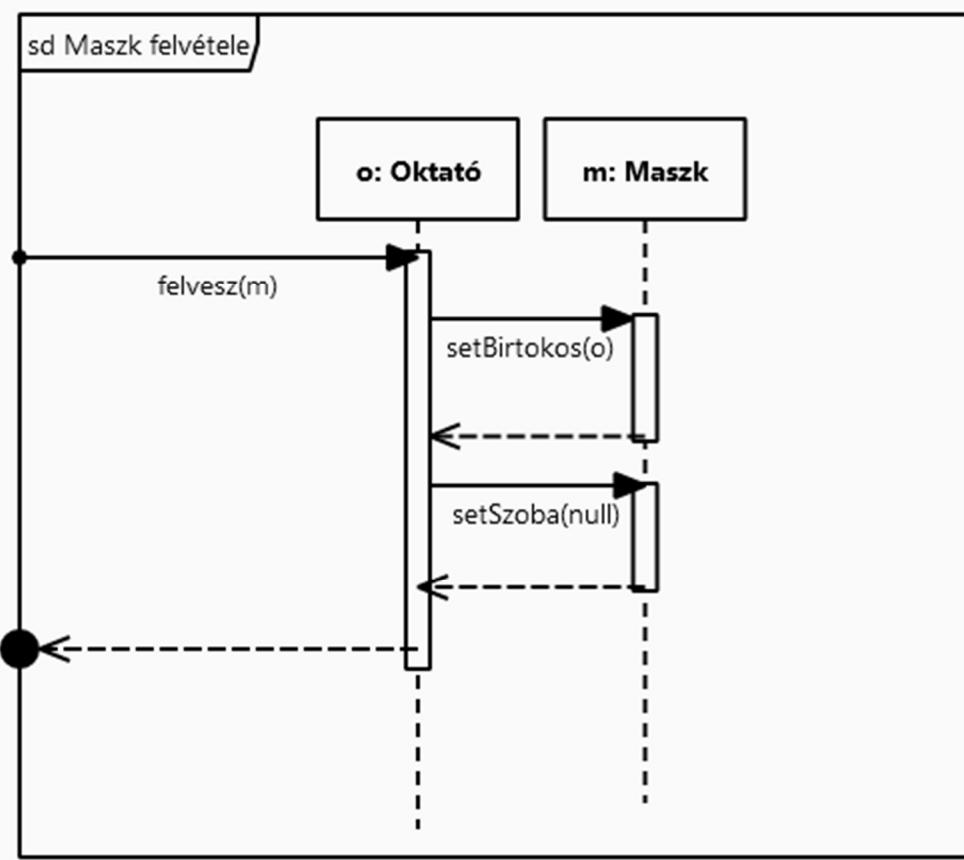
5.3 Szekvencia diagramok a belső működésre

5.3.1 Hallgató maszk felvétele



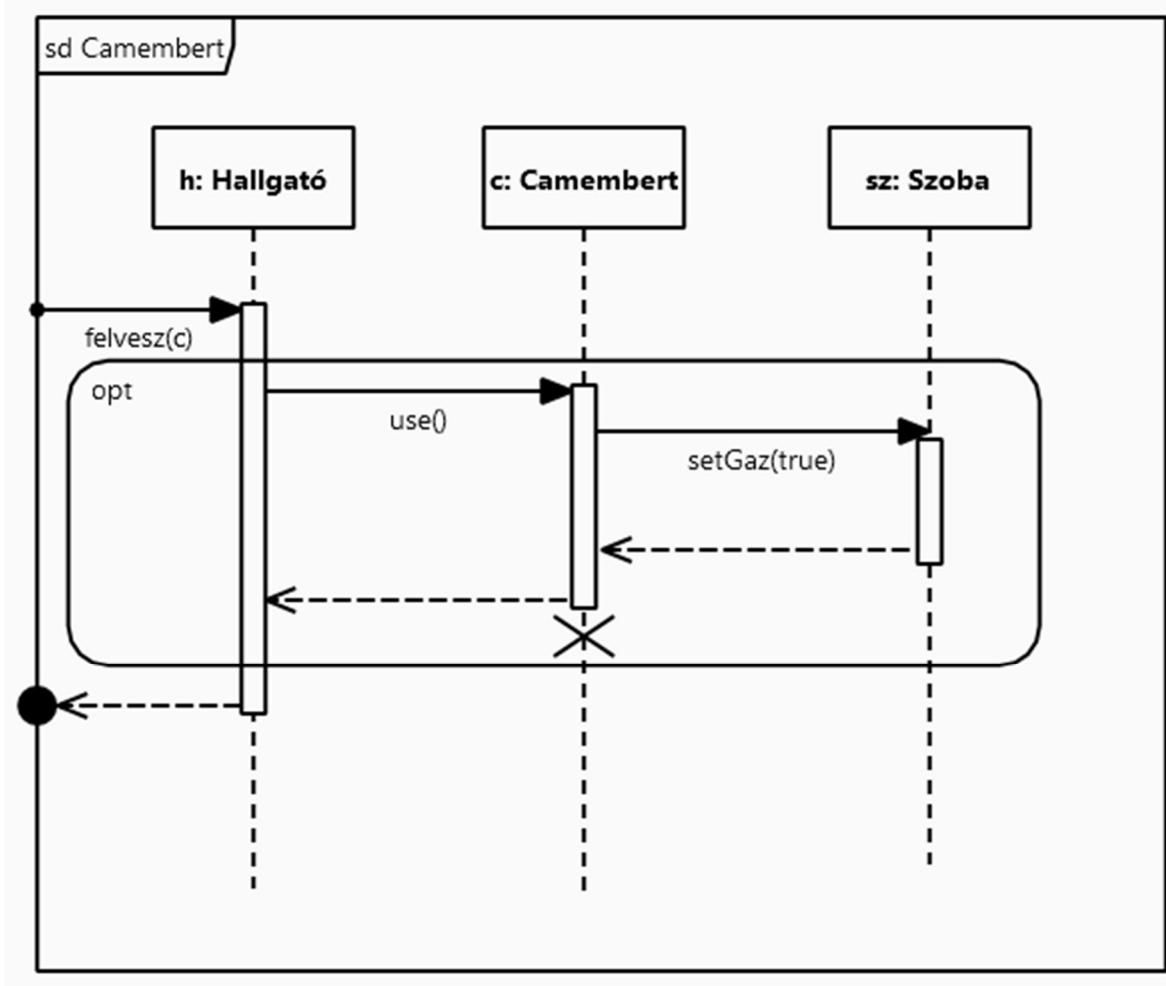
Az összes többi tárgy esetében a tárgyfelvétel teljesen azonos módon történik. Csupán annyiban különbözik, hogy a Maszk lifeline helyett az adott tárgy lifeline-ja van.

5.3.2 Oktató maszk felvétele

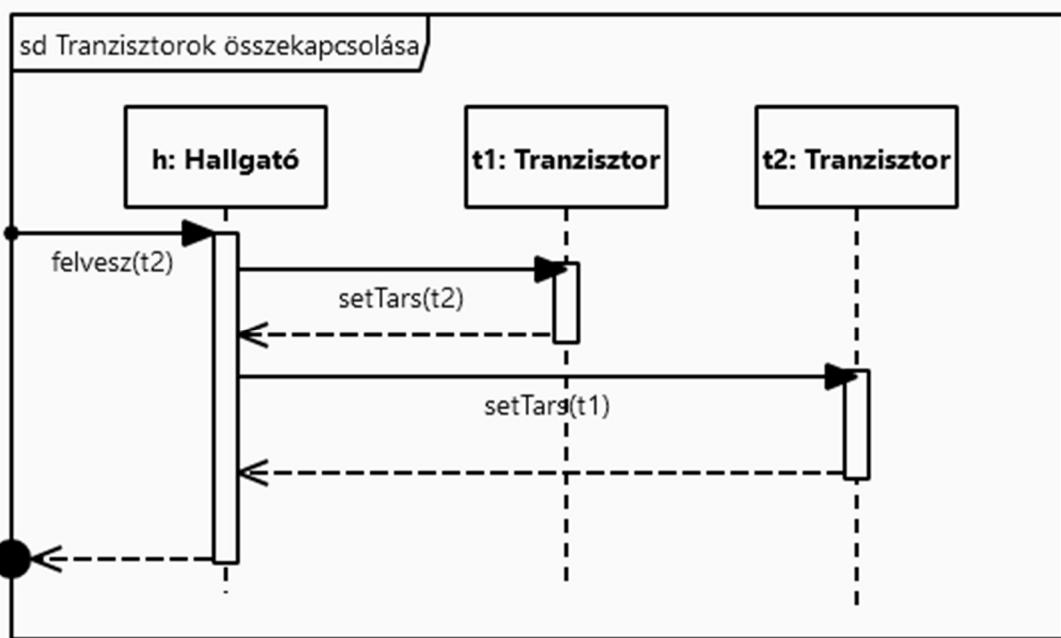


Az összes többi tárgy esetében a tárgyfelvétel teljesen azonos módon történik. Csupán annyiban különbözik, hogy a Maszk lifeline helyett az adott tárgy lifeline-ja van.

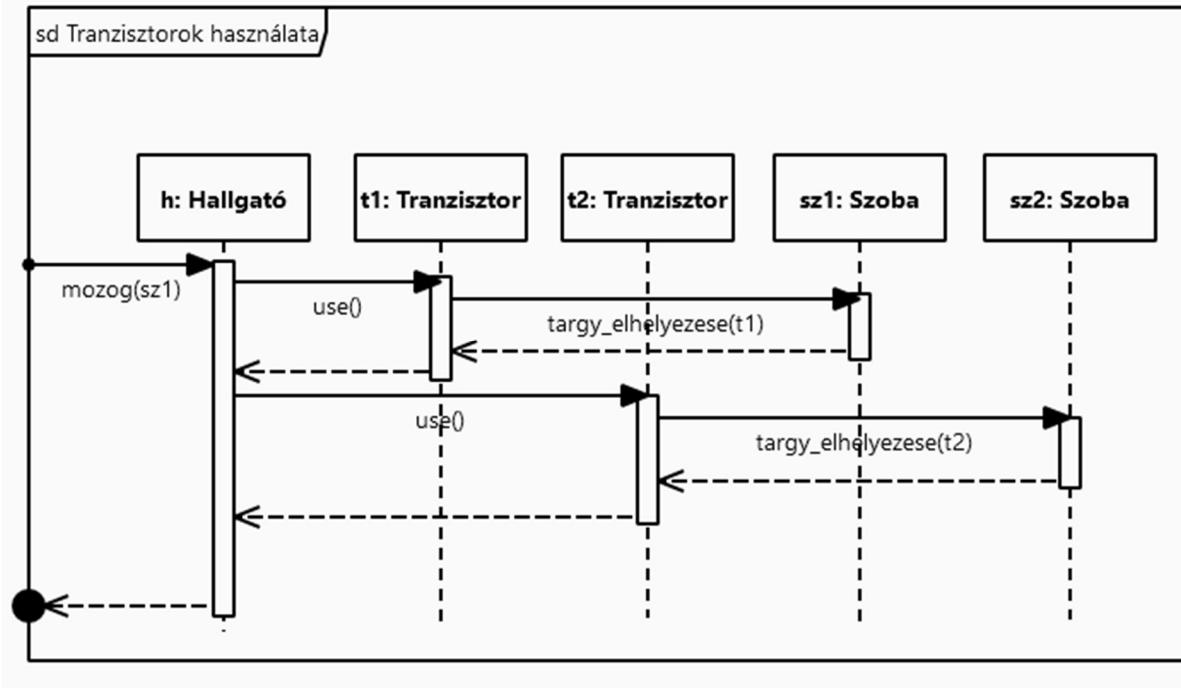
5.3.3 Camembert használata



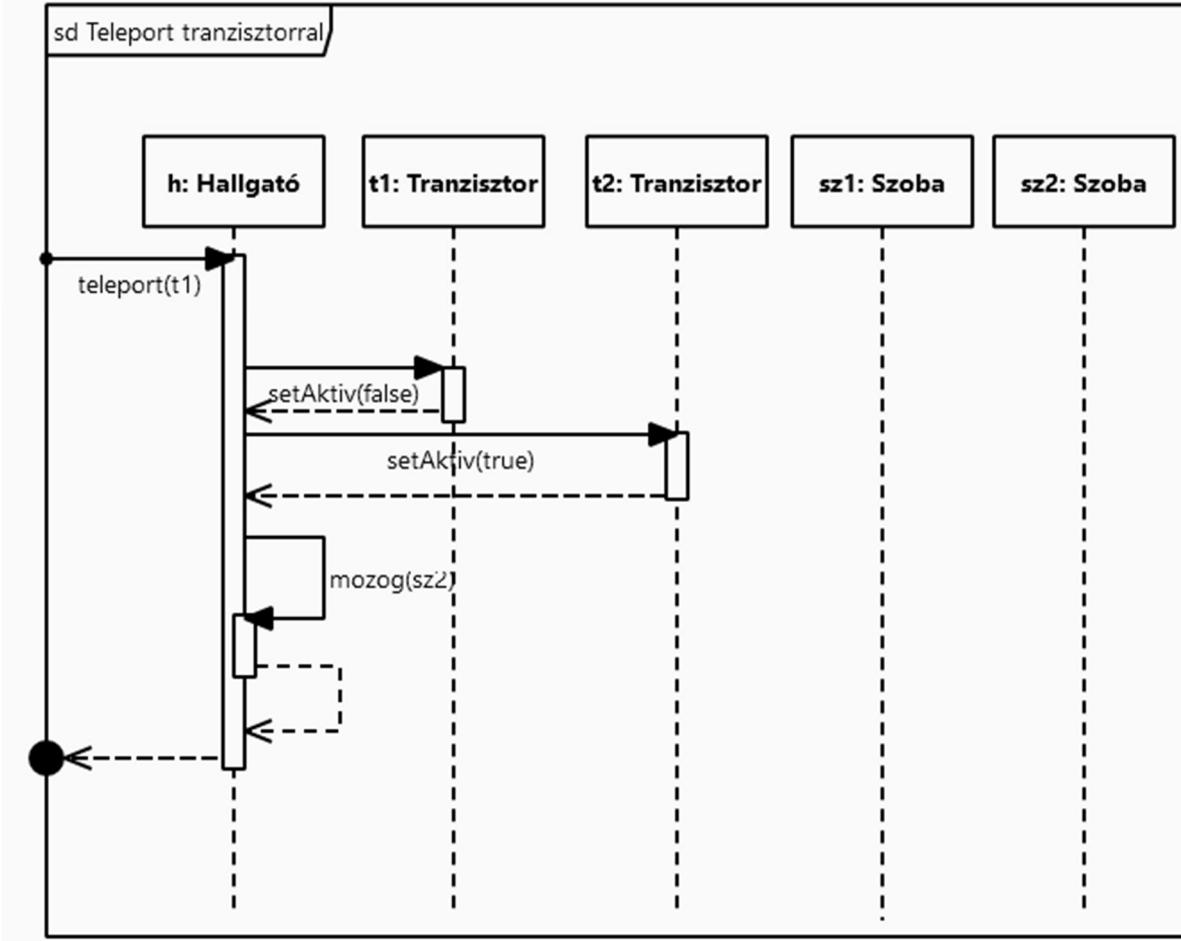
5.3.4 Tranzisztorok összekapcsolása



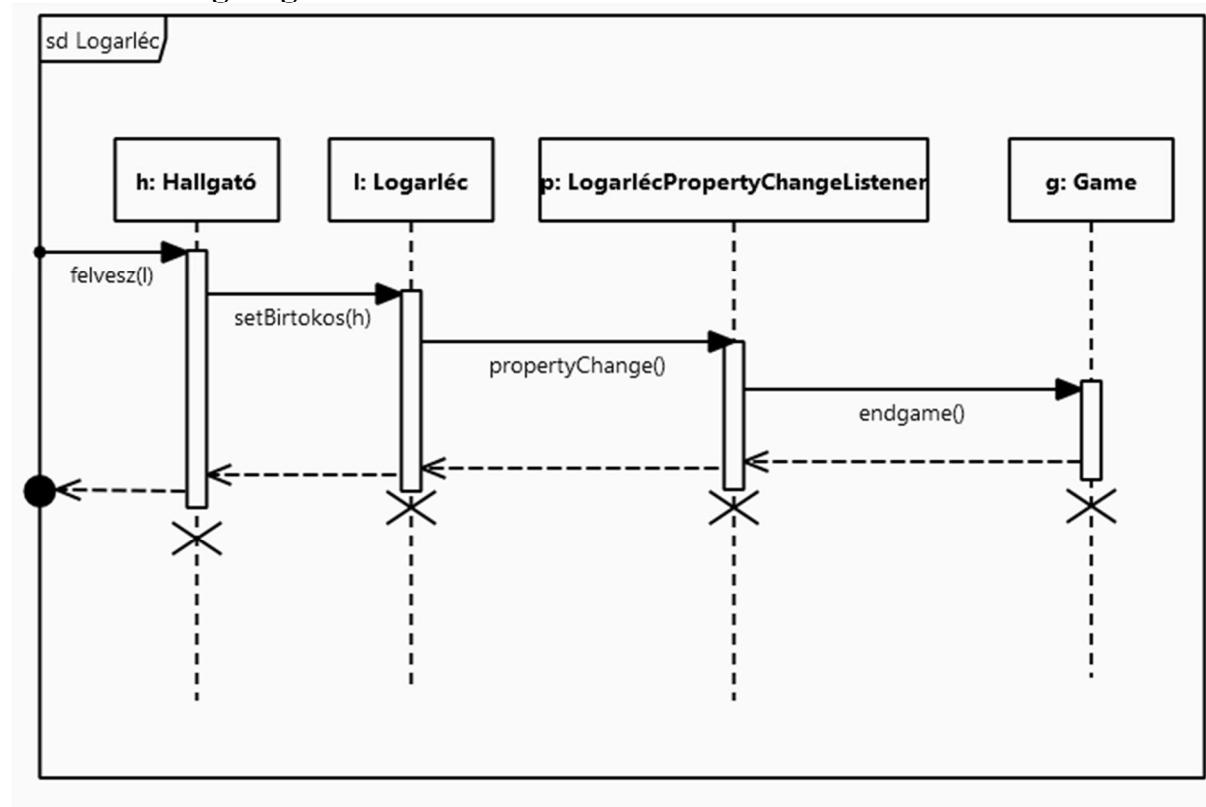
5.3.5 Tranzisztorok használata



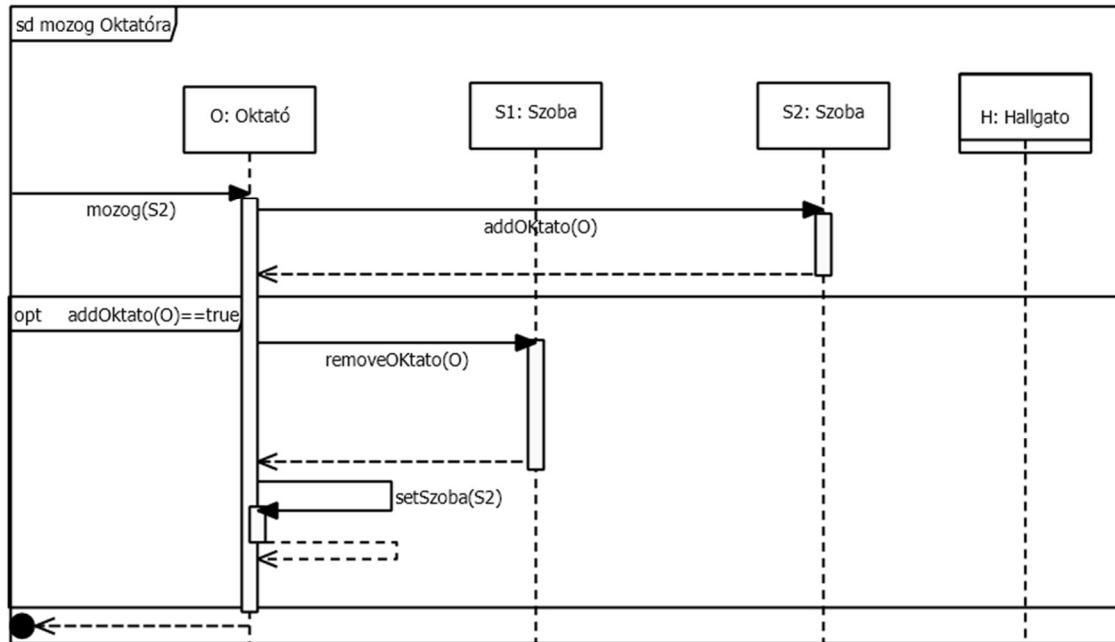
5.3.6 Teleportálás tranzisztorral



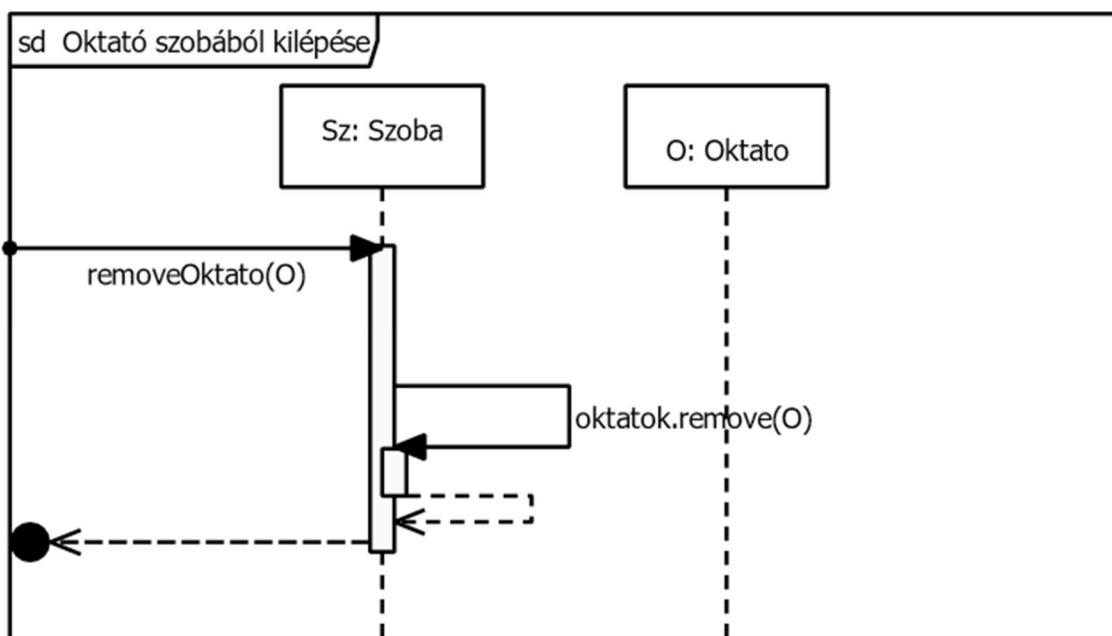
5.3.7 Játék vége logarléc felvétele után



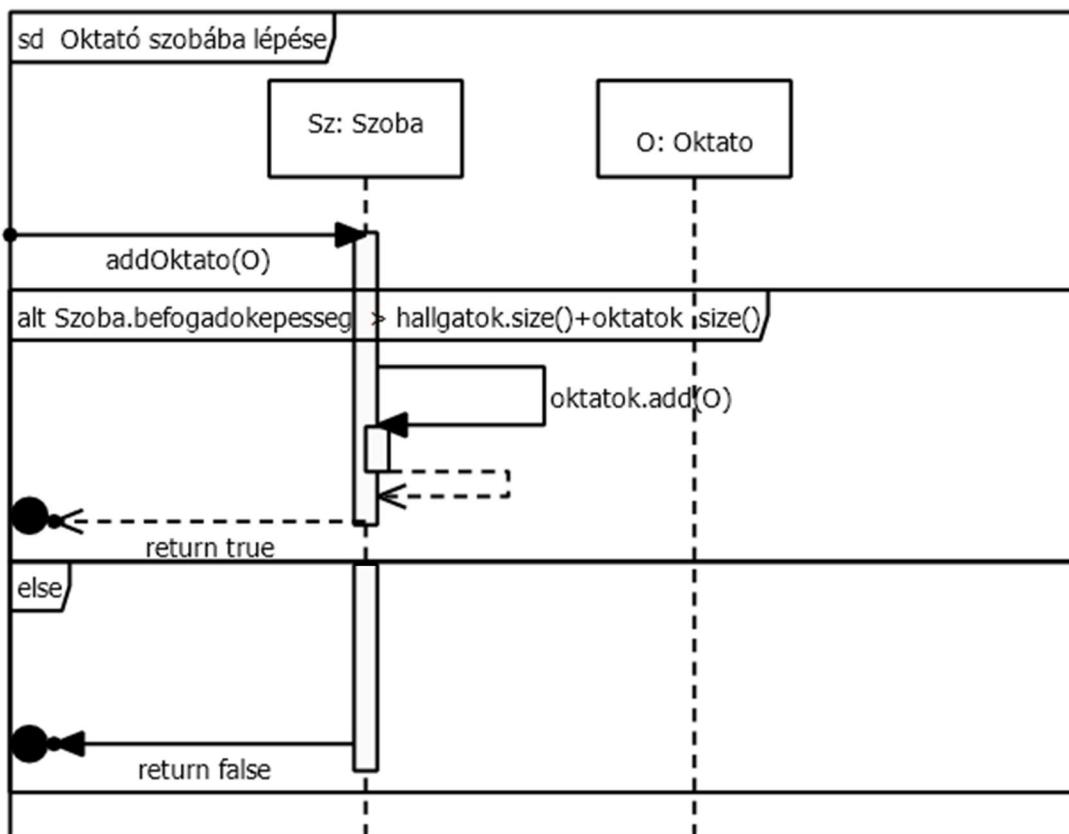
5.3.8 Oktató mozgása



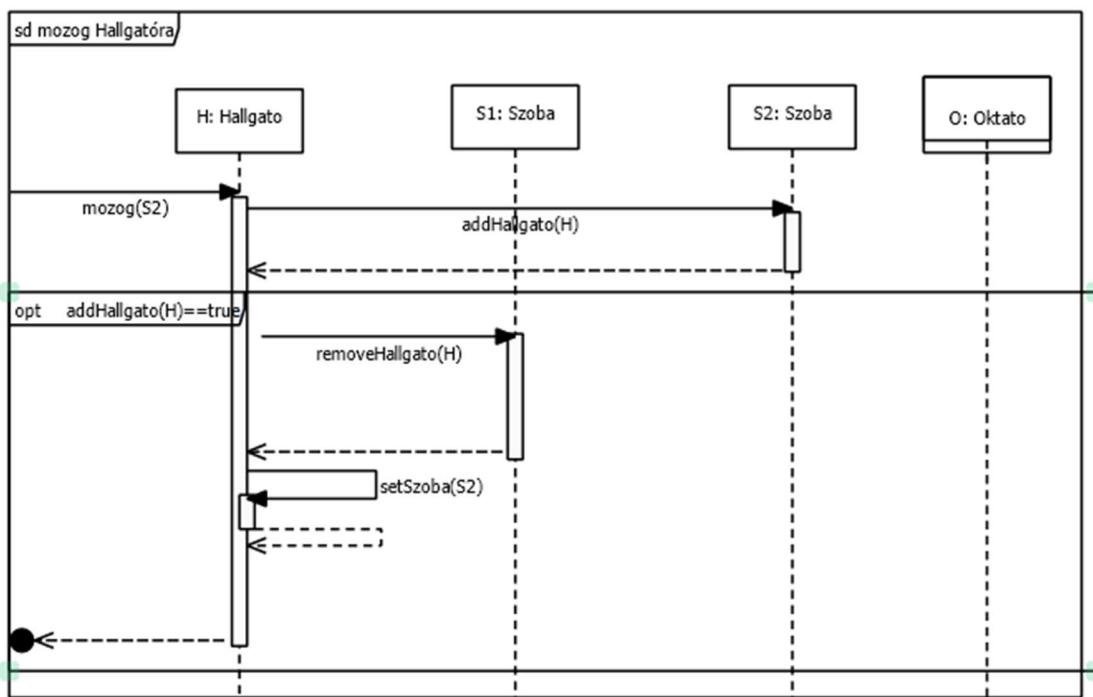
5.3.9 Oktató szobából kilépése



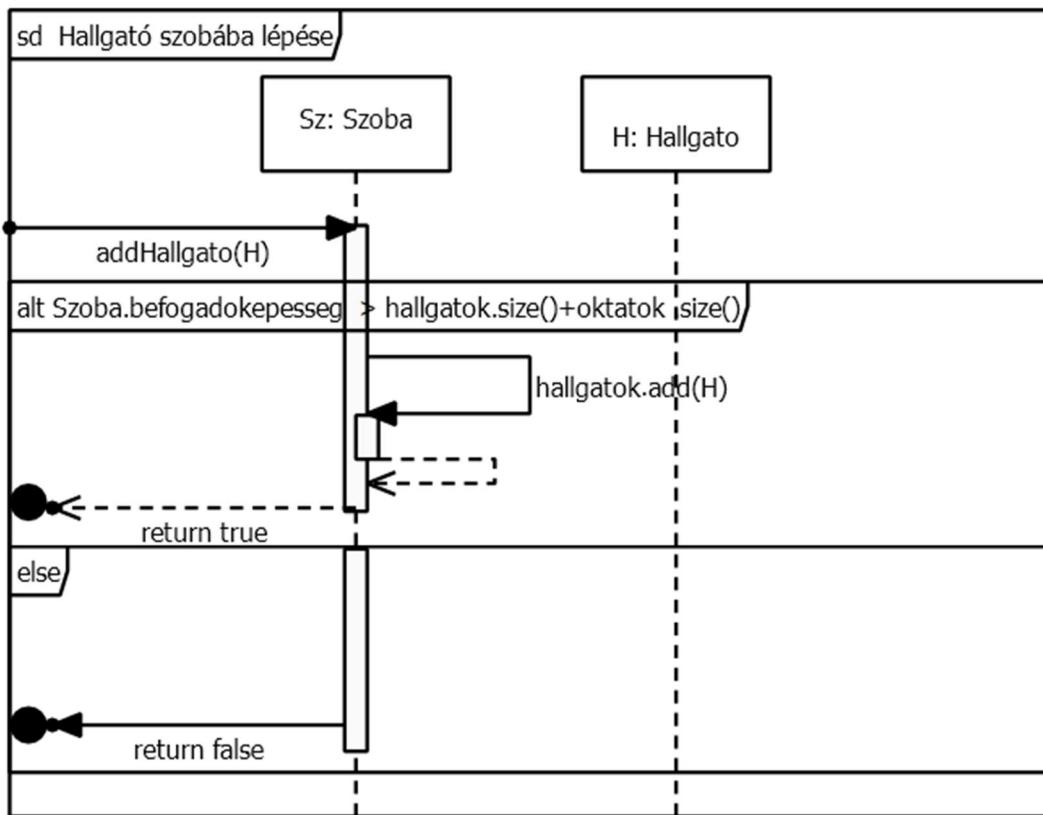
5.3.10 Oktató szobába lépése



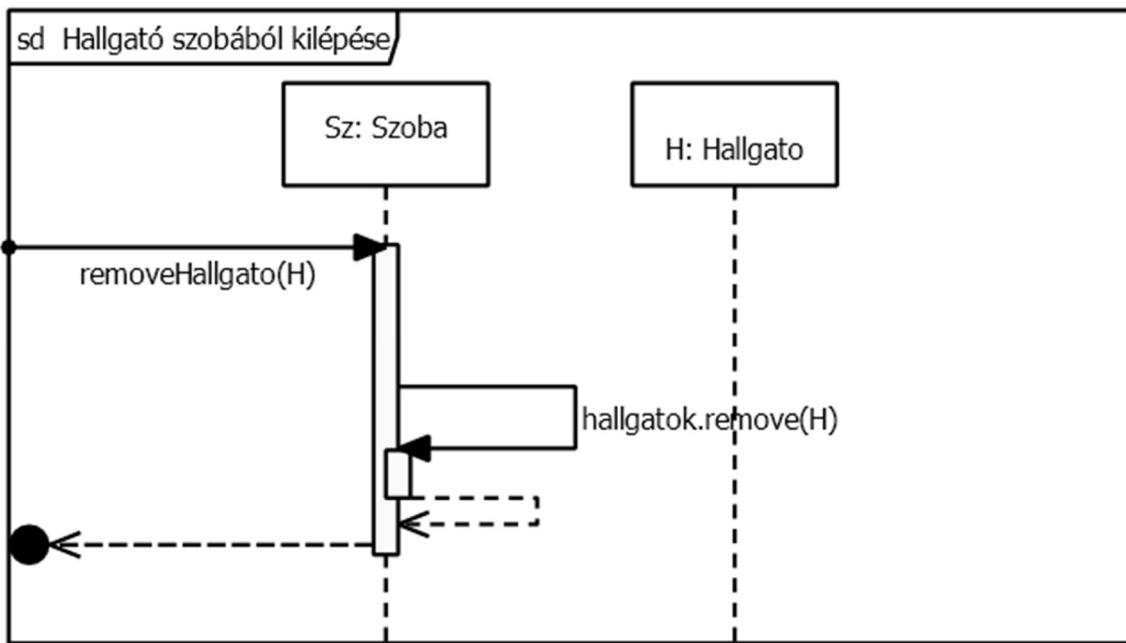
5.3.11 Hallgató mozgása



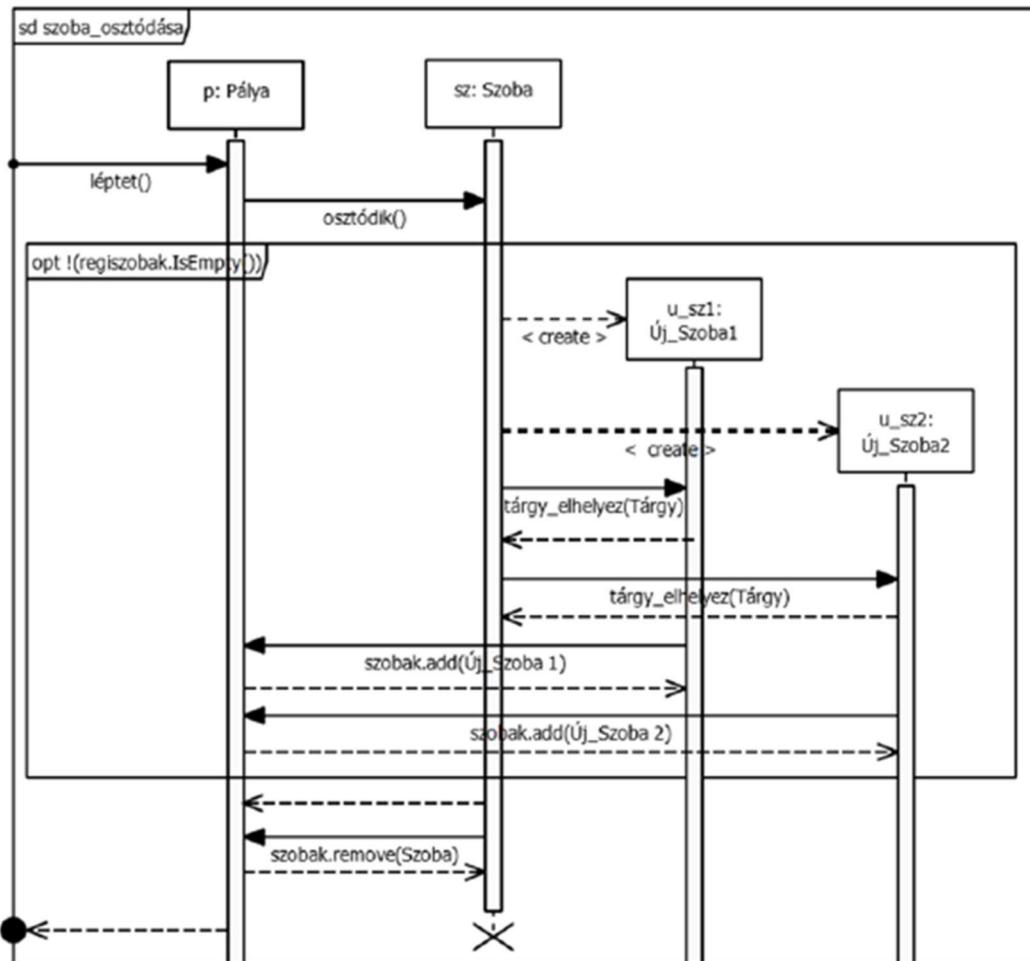
5.3.12 Hallgató szobába lépése



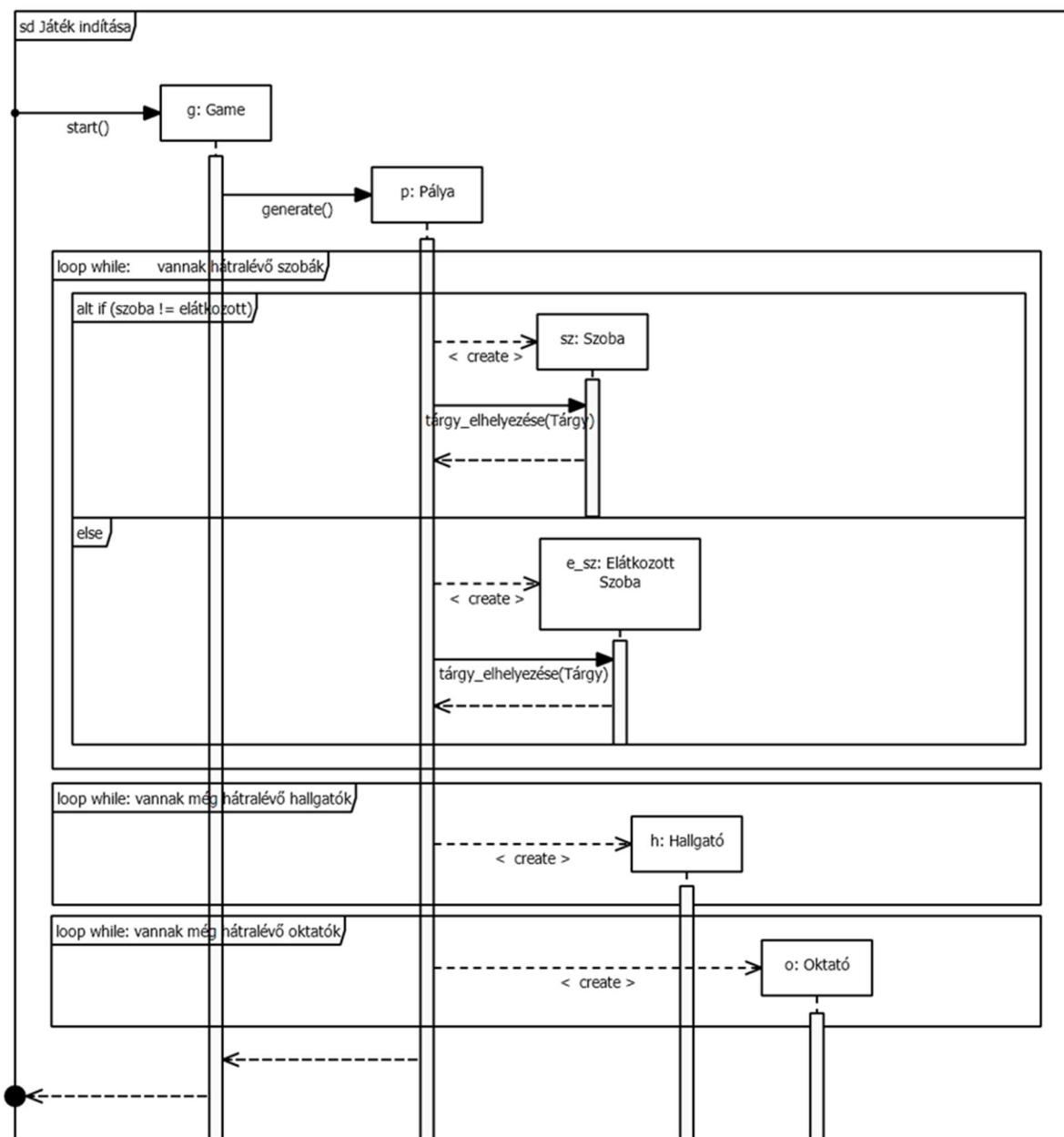
5.3.13 Hallgató szobából kilépése



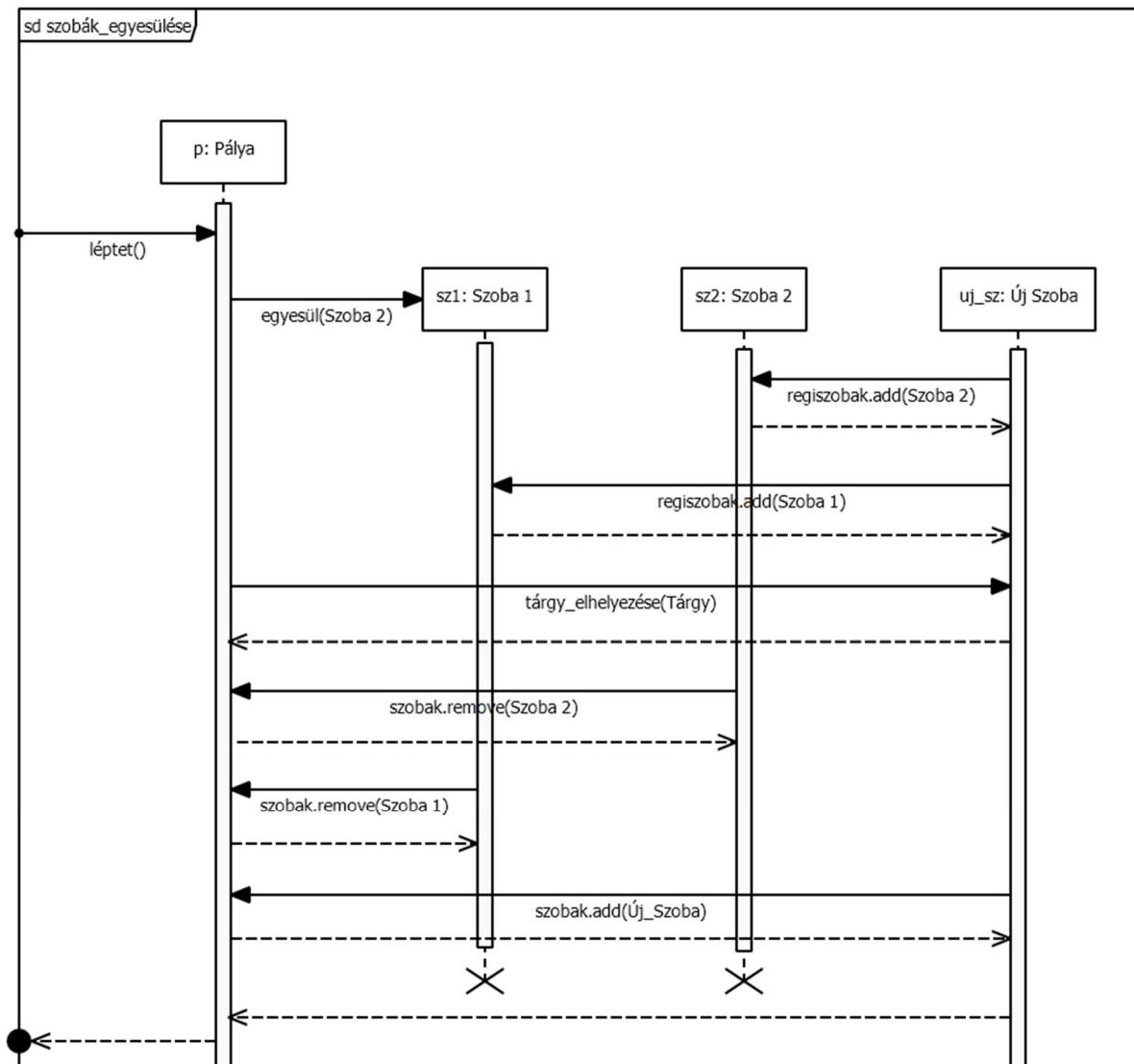
5.3.14 Szoba osztódása

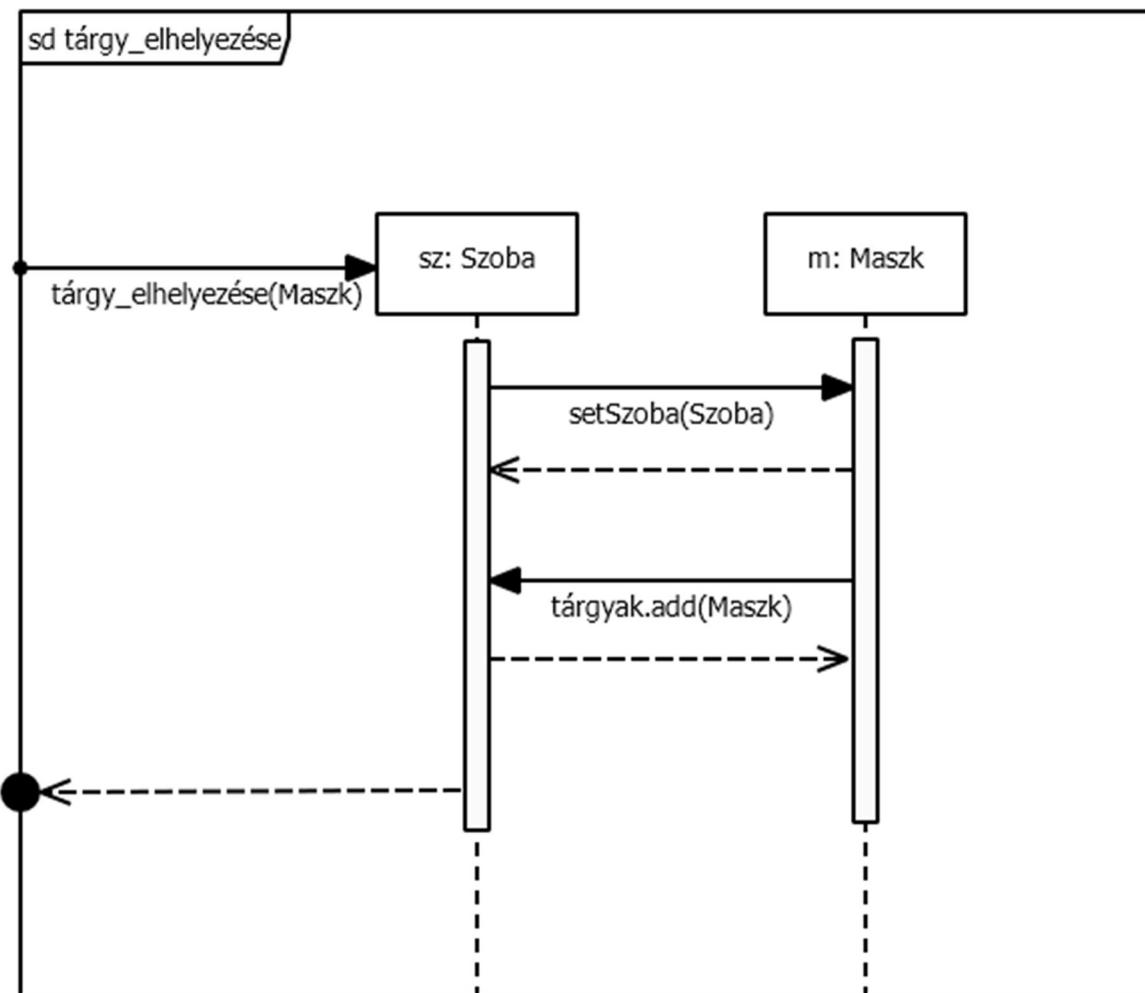


5.3.15 Játék indítása



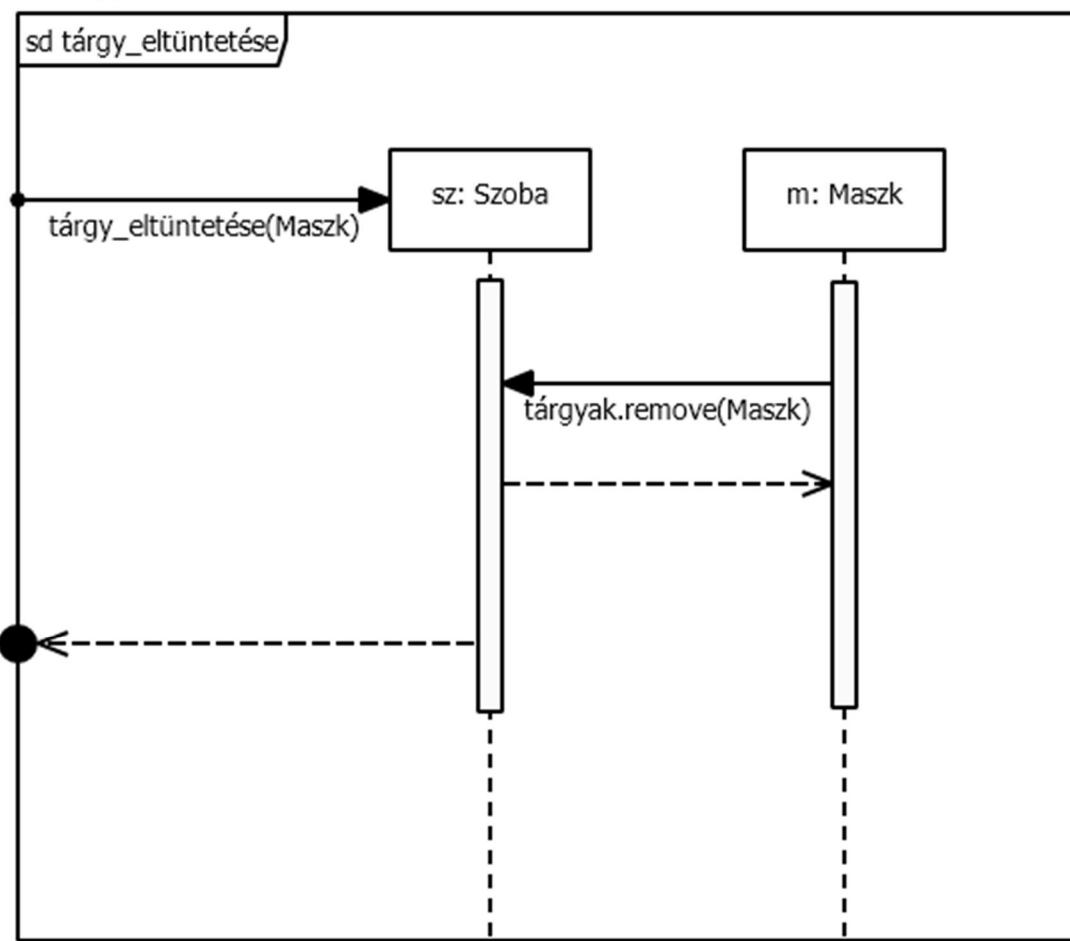
5.3.16 Szobák egyesülése



5.3.17 Tárgy elhelyezése

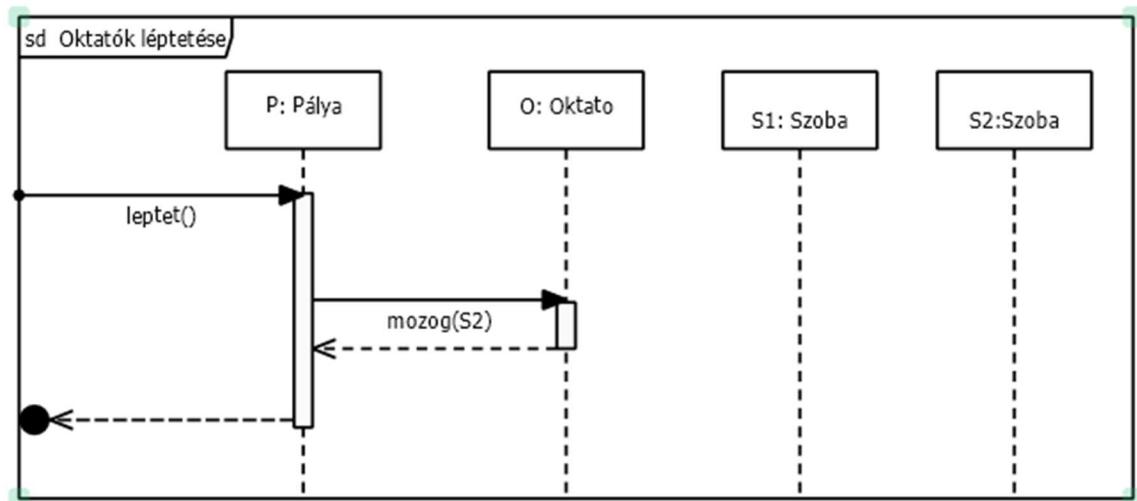
Az összes többi tárgy elhelyezése azonos a fenti példában szereplő Maszk tárgy elhelyezésével.

5.3.18 Tárgy eltüntetése

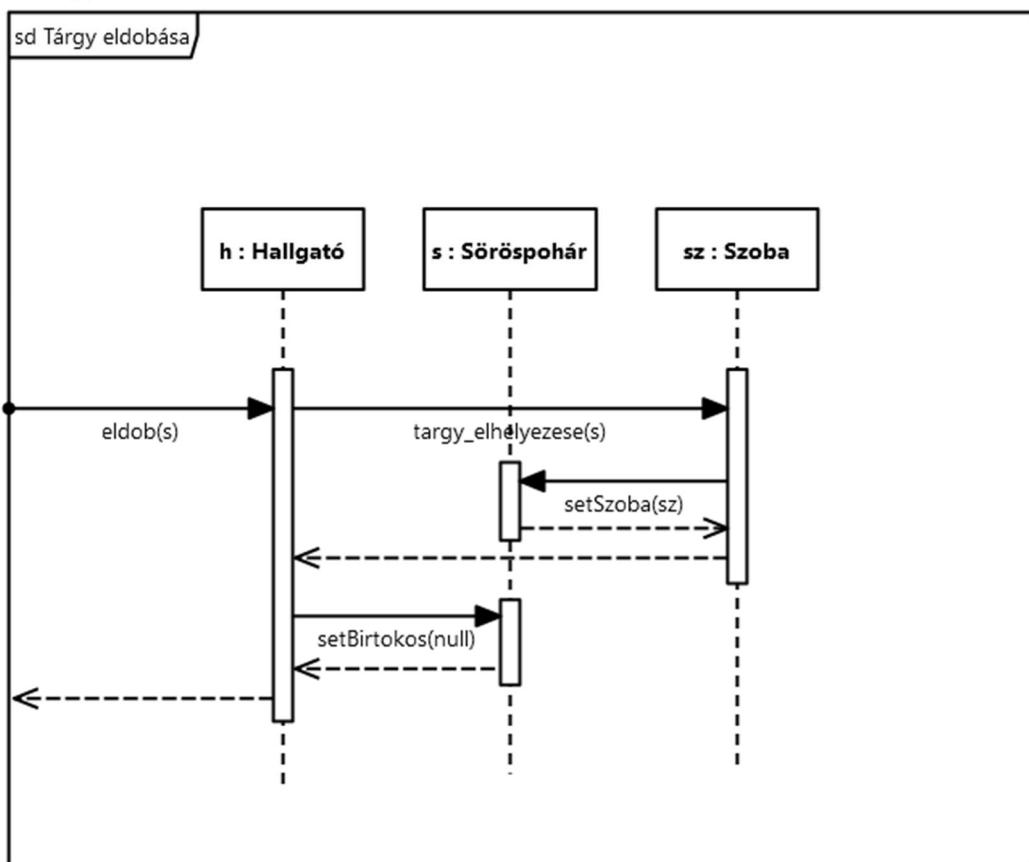


Az összes többi tárgy eltüntetése azonos a fenti példában szereplő `Maszk` tárgy eltüntetésével.

5.3.19 Oktatók léptetése

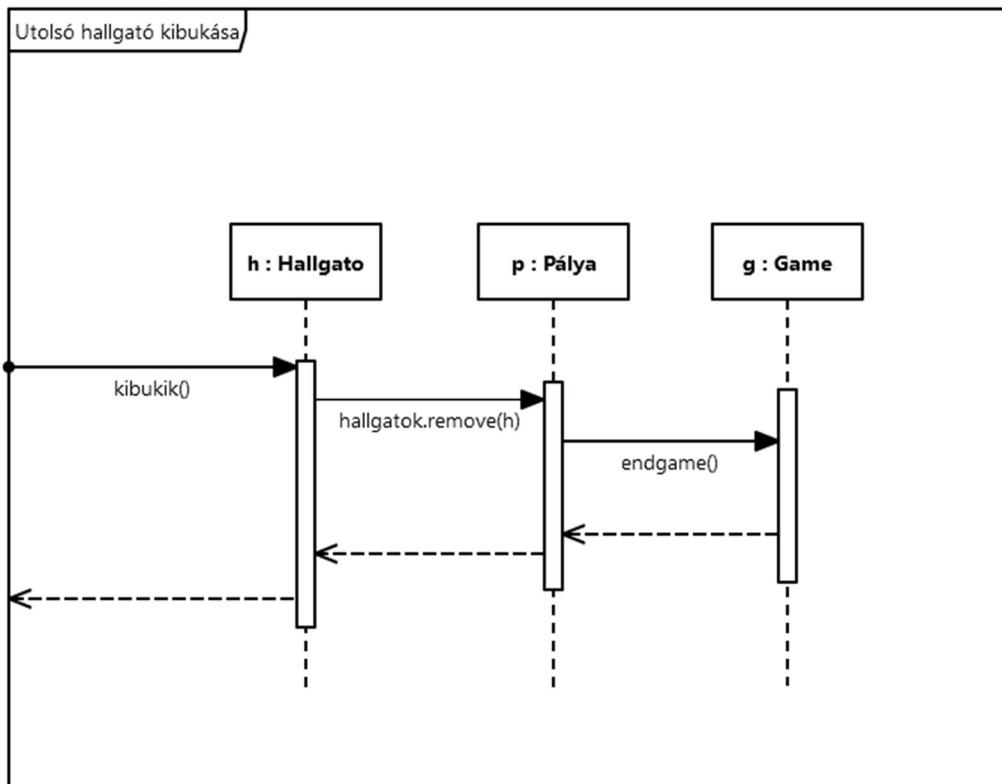


5.3.20 Tárgy eldobása

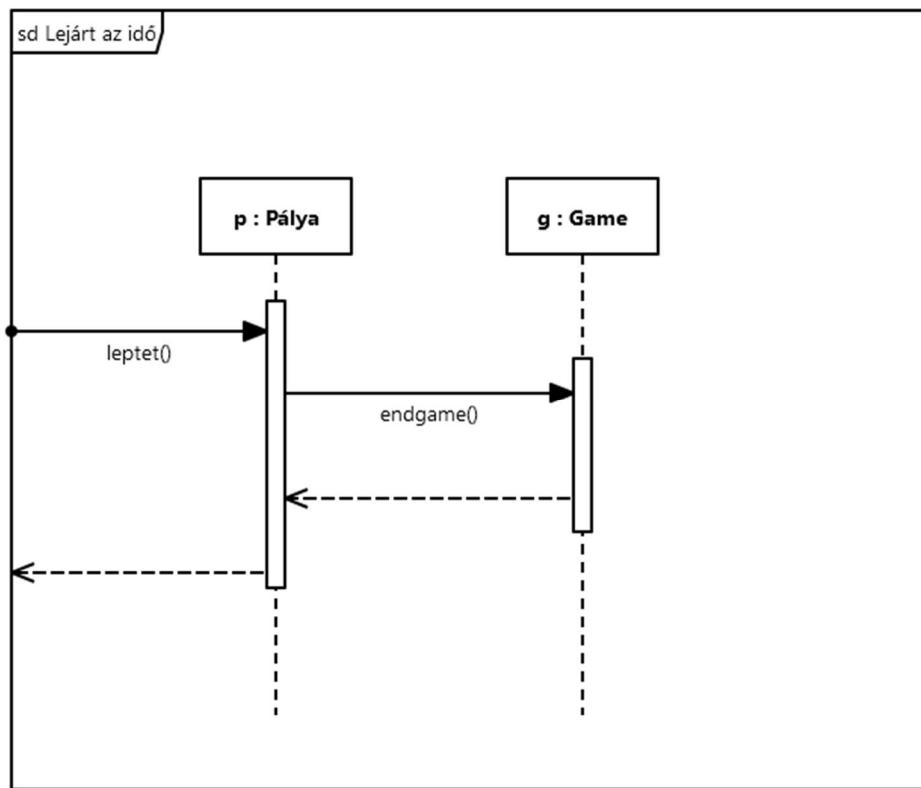


Összes tárgynál ugyanez a helyzet.

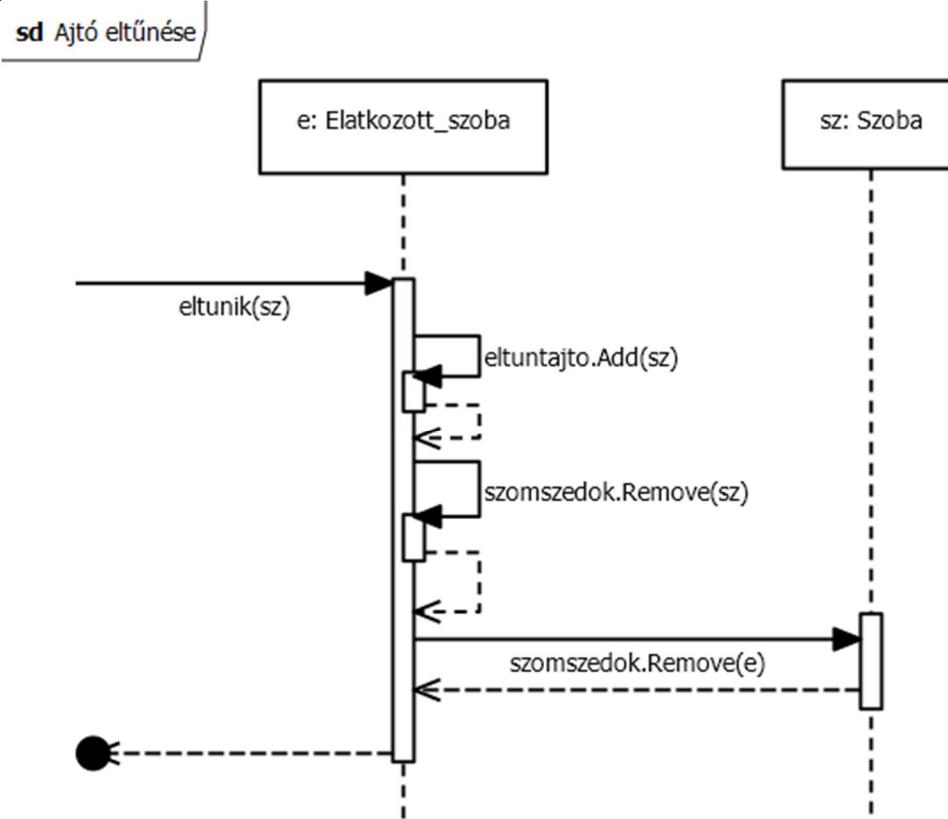
5.3.21 Utolsó hallgató kibukik

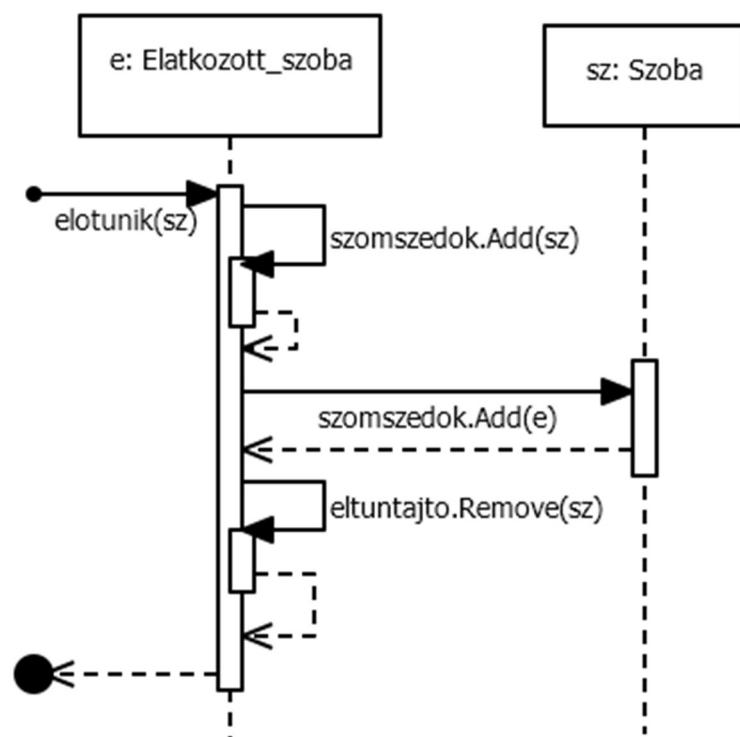


5.3.22 Lejár az idő



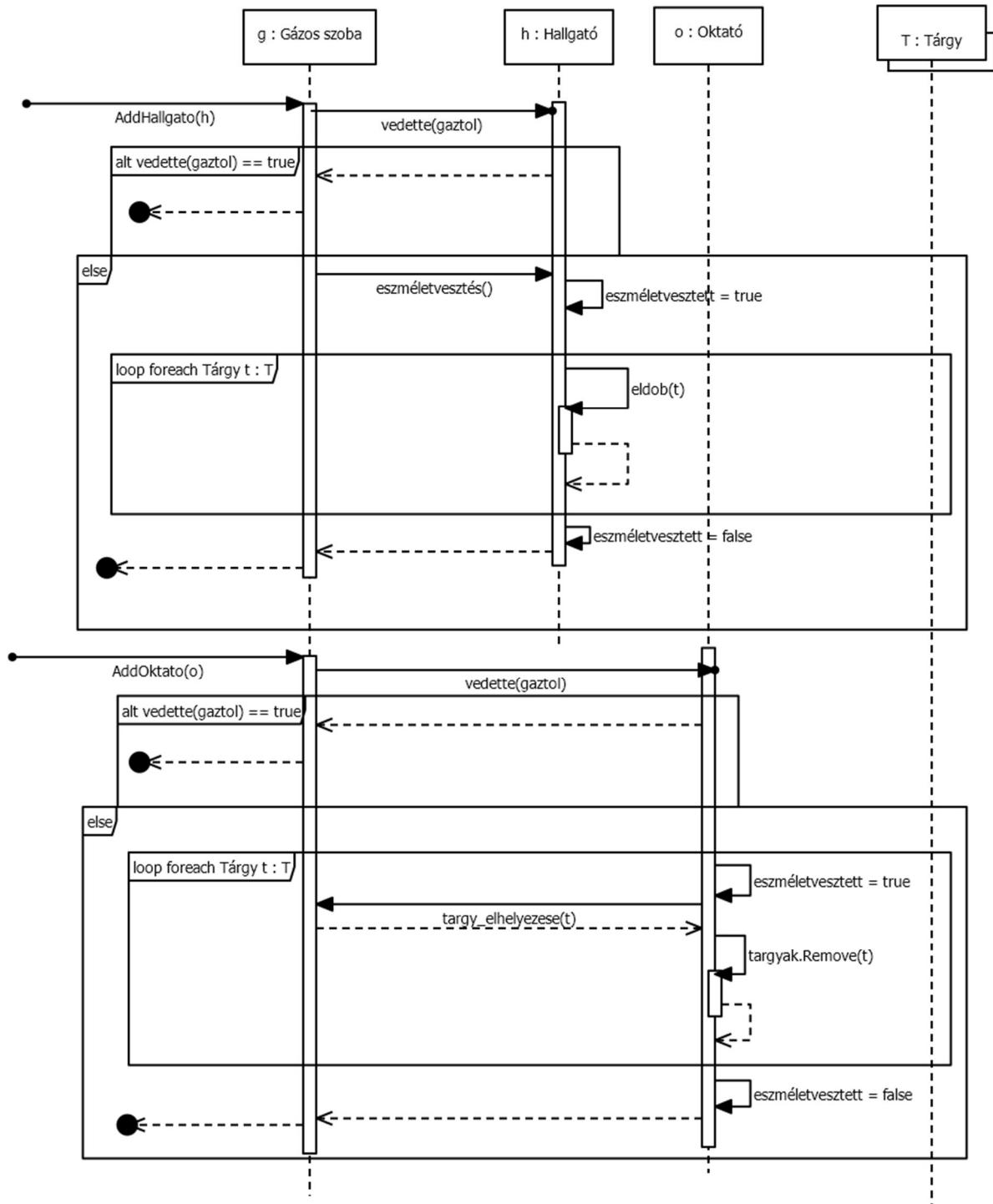
5.3.23 Ajtó eltűnése



5.3.24 Ajtó előtűnésesd Ajtó előtűnése

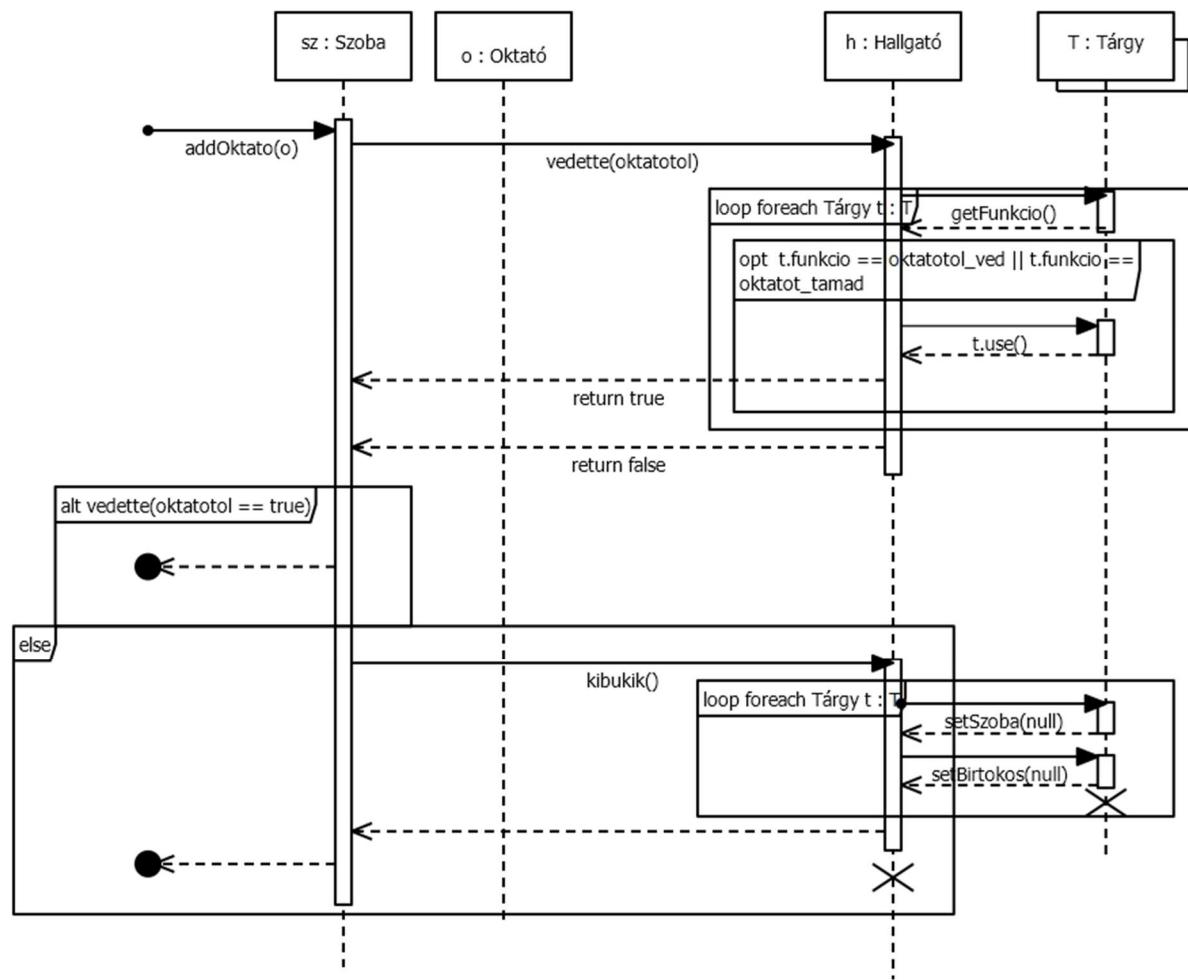
5.3.25 Eszméletvesztés

sd Eszméletvesztés

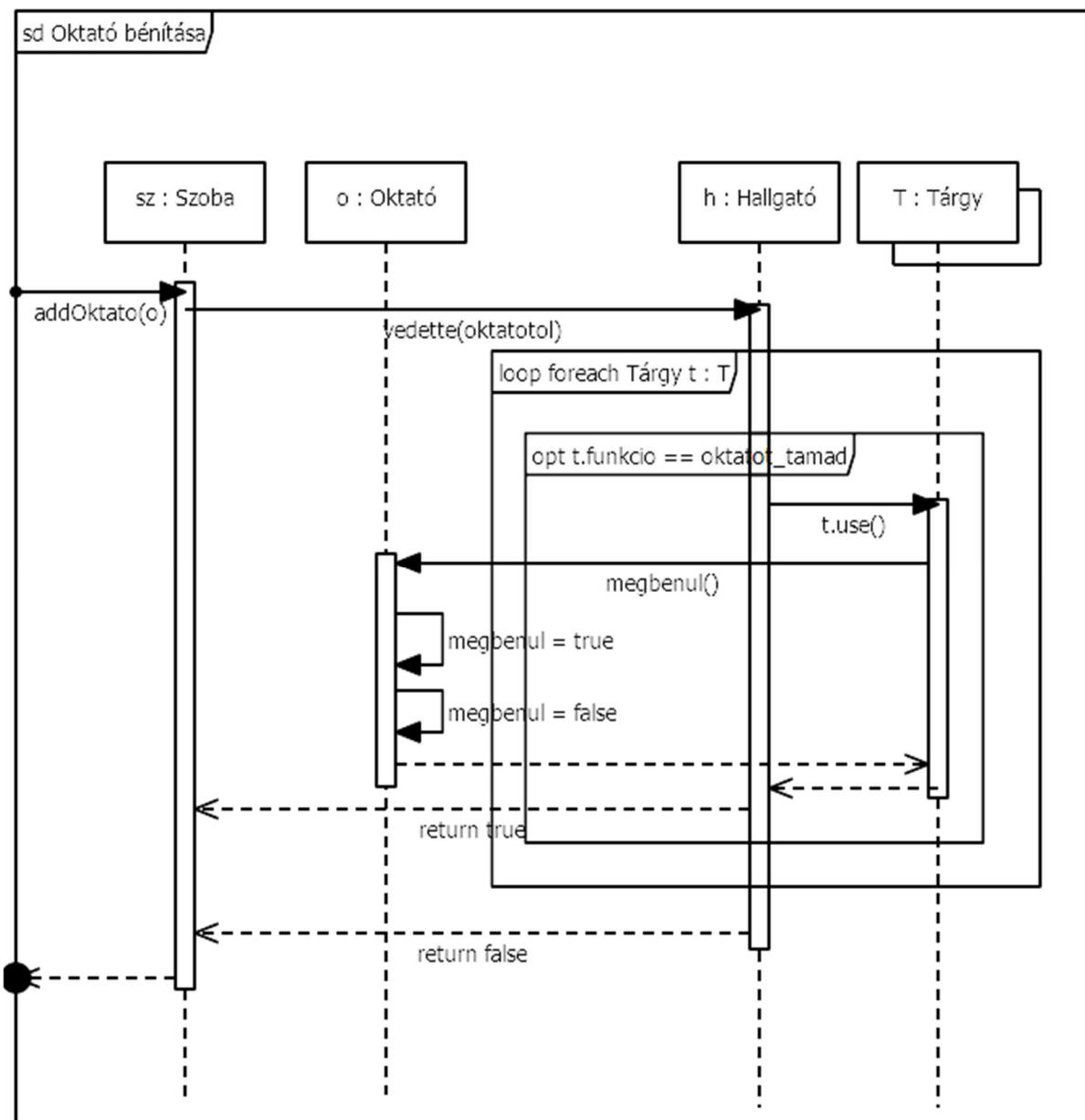


5.3.26 Hallgató kibuktatása

sd Hallgató kibuktatása

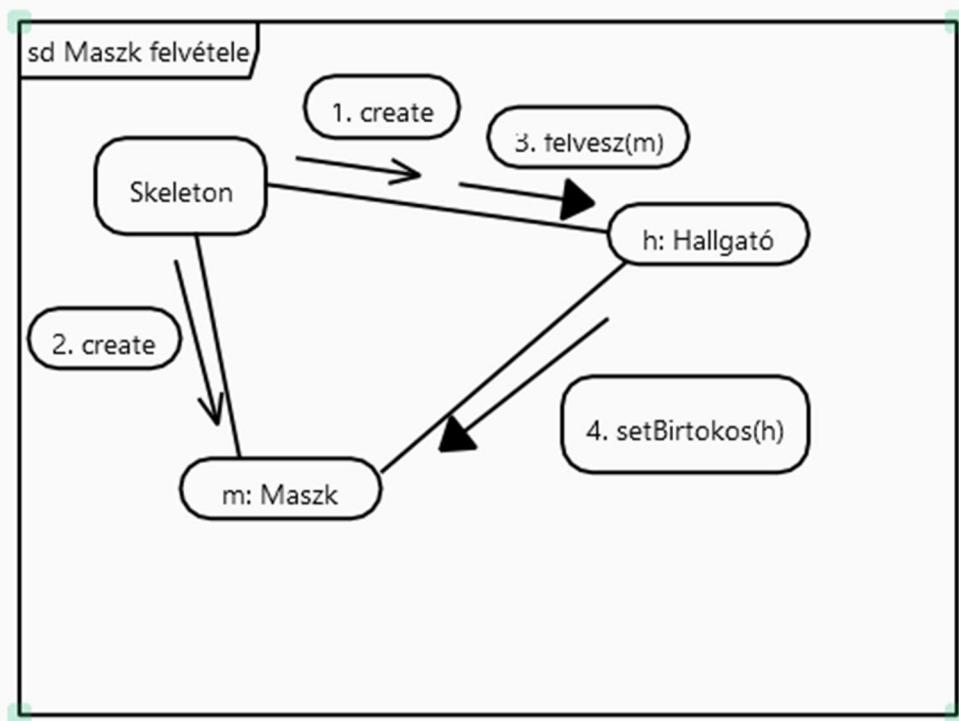


5.3.27 Oktató megbénítása



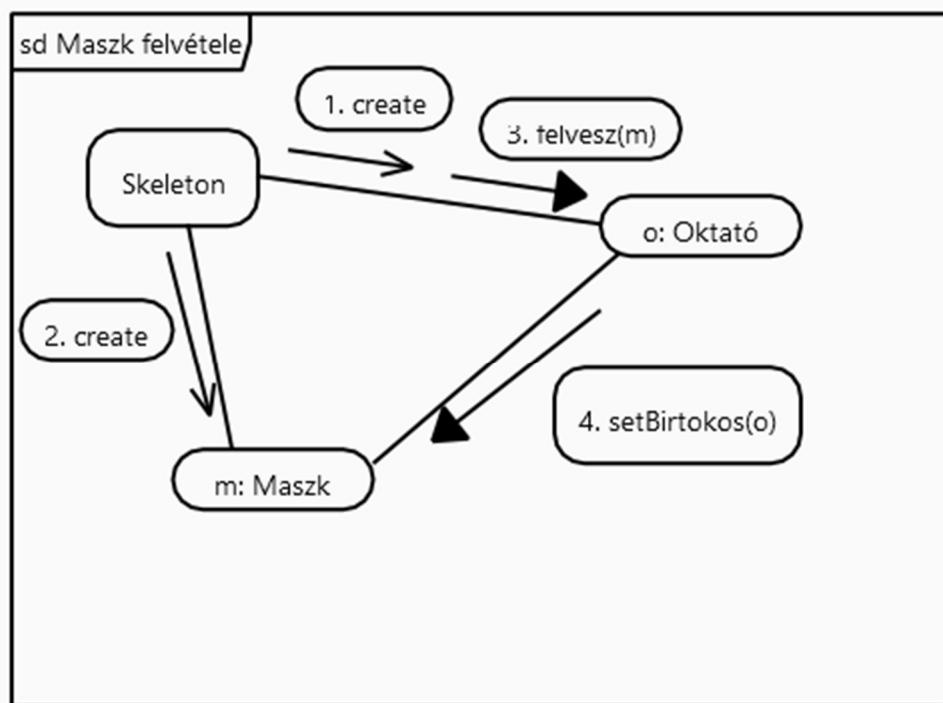
5.4 Kommunikációs diagramok

5.4.1 Hallgató maszk felvétele



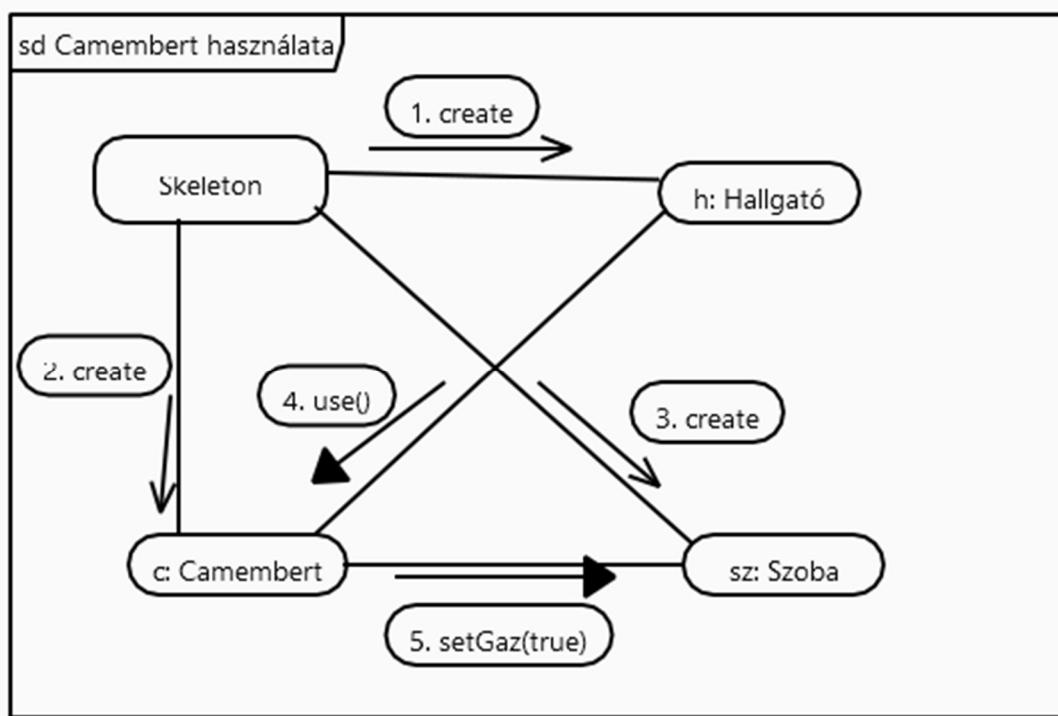
Az összes többi tárgy esetében a tárgyfelvétel teljesen azonos módon történik. Csupán annyiban különbözik, hogy a Maszk helyett az adott tárgy van.

5.4.2 Oktató maszk felvétele

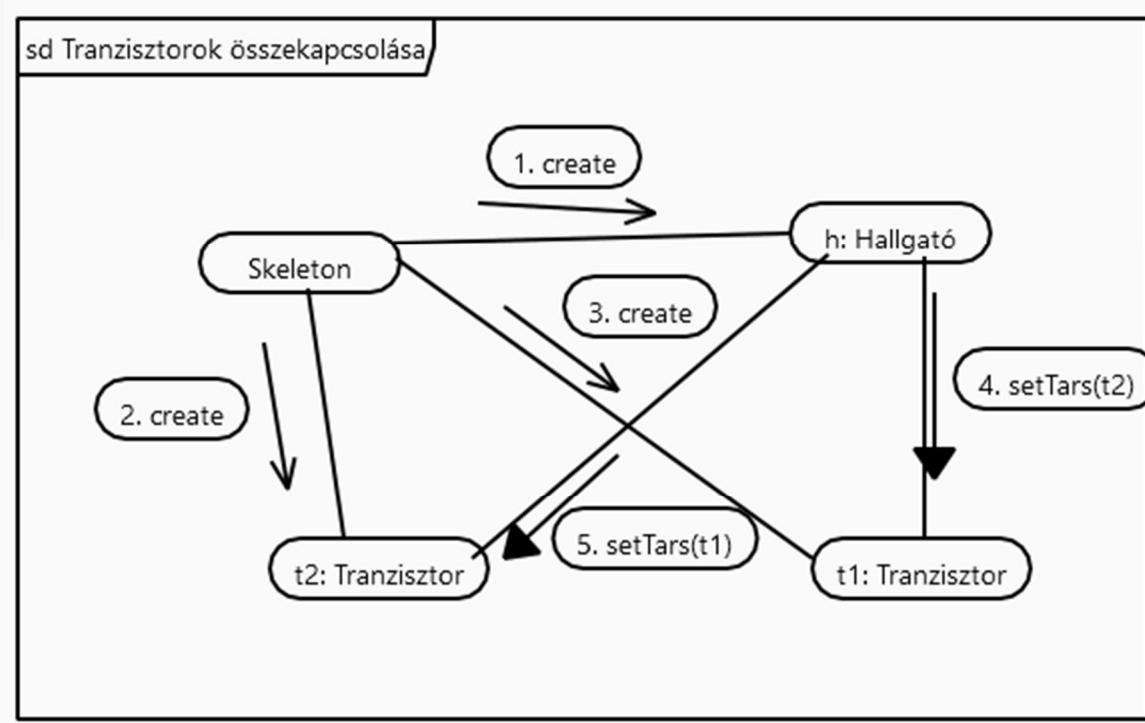


Az összes többi tárgy esetében a tárgyfelvétel teljesen azonos módon történik. Csupán annyiban különbözik, hogy a Maszk helyett az adott tárgy van.

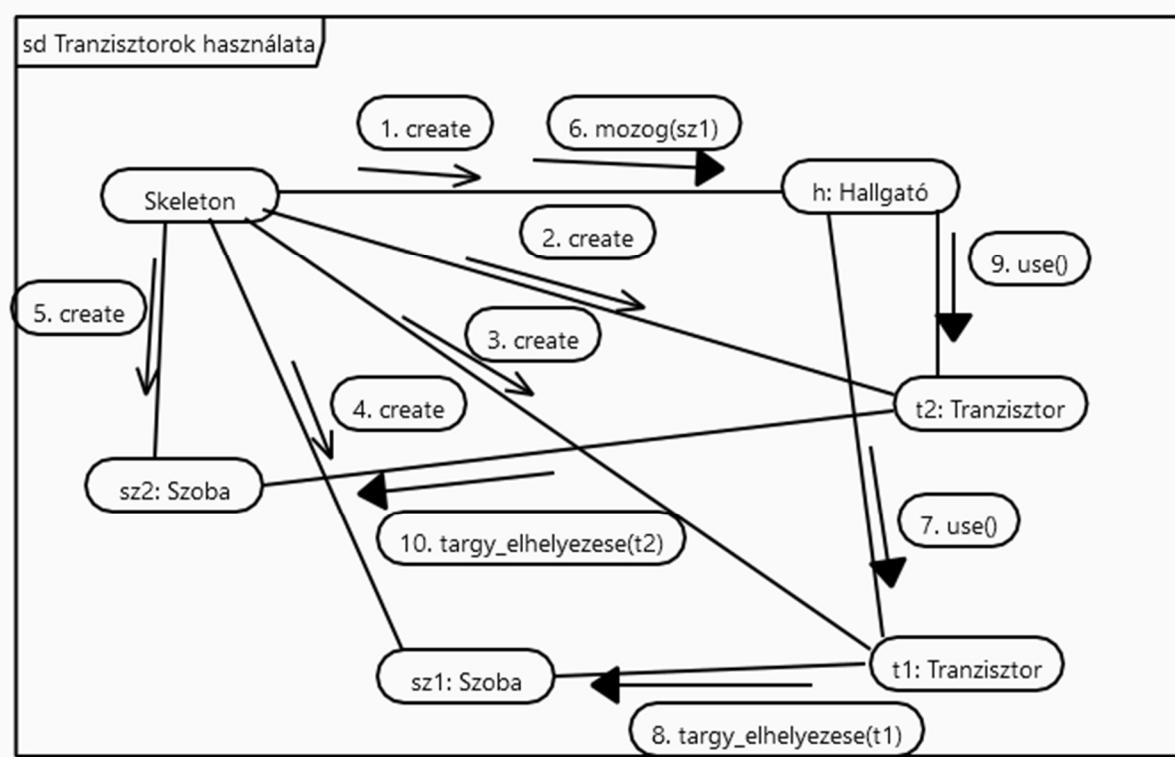
5.4.3 Camembert használata



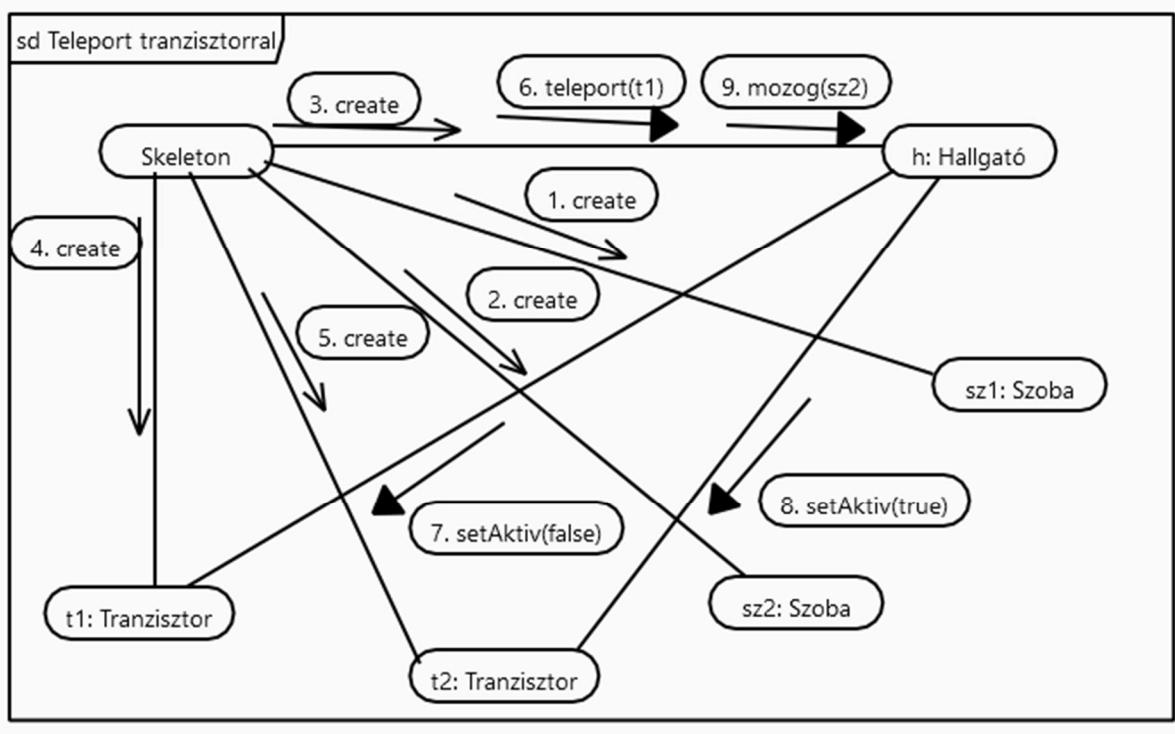
5.4.4 Tranzisztorok összekapcsolása



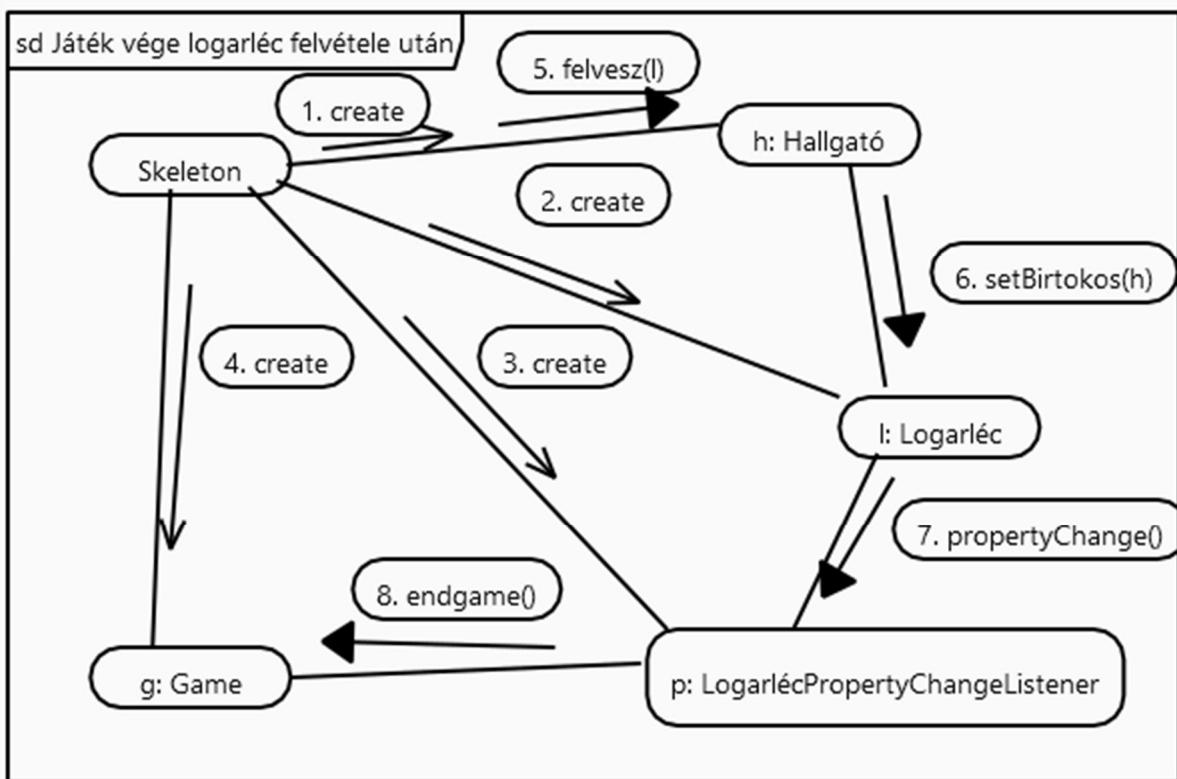
5.4.5 Tranzisztorok használata



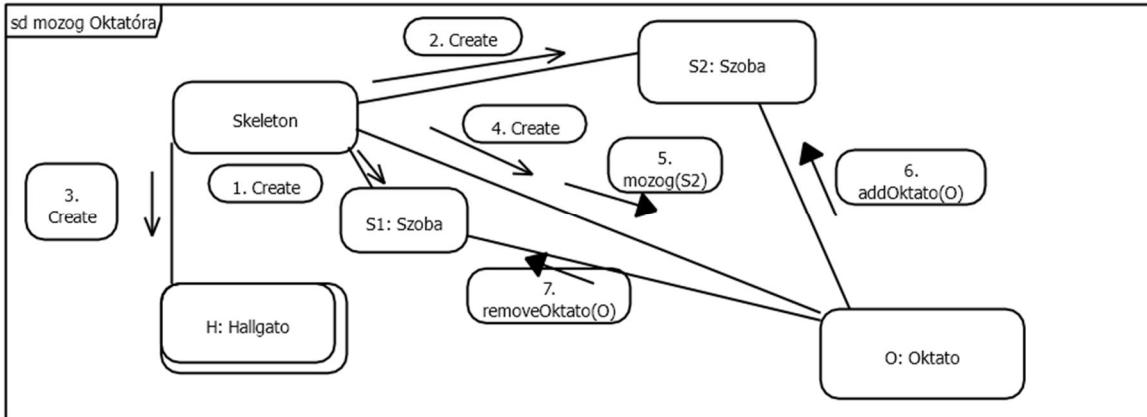
5.4.6 Teleportálás tranzisztorral

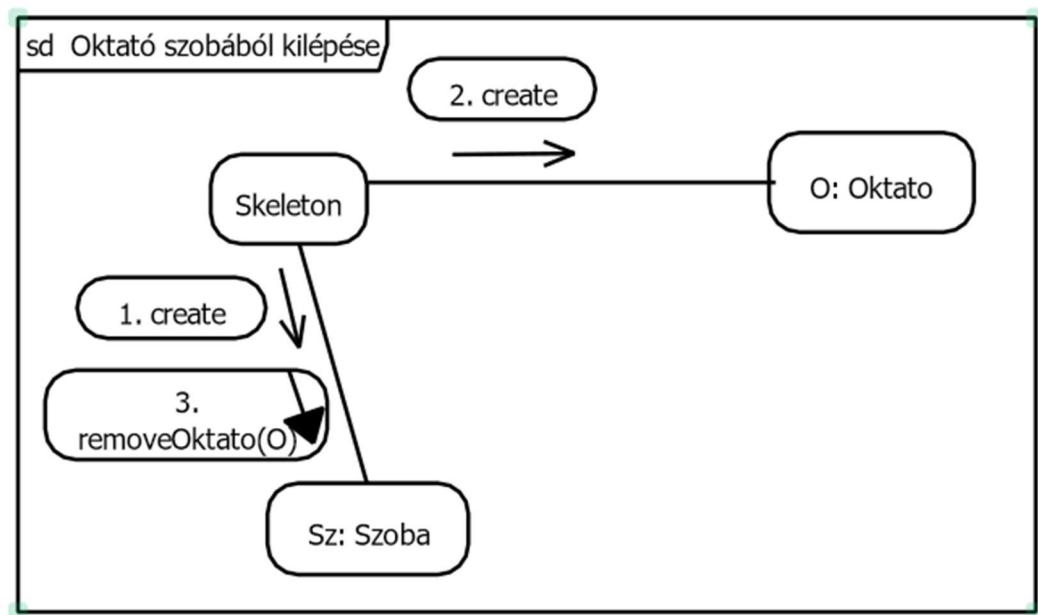
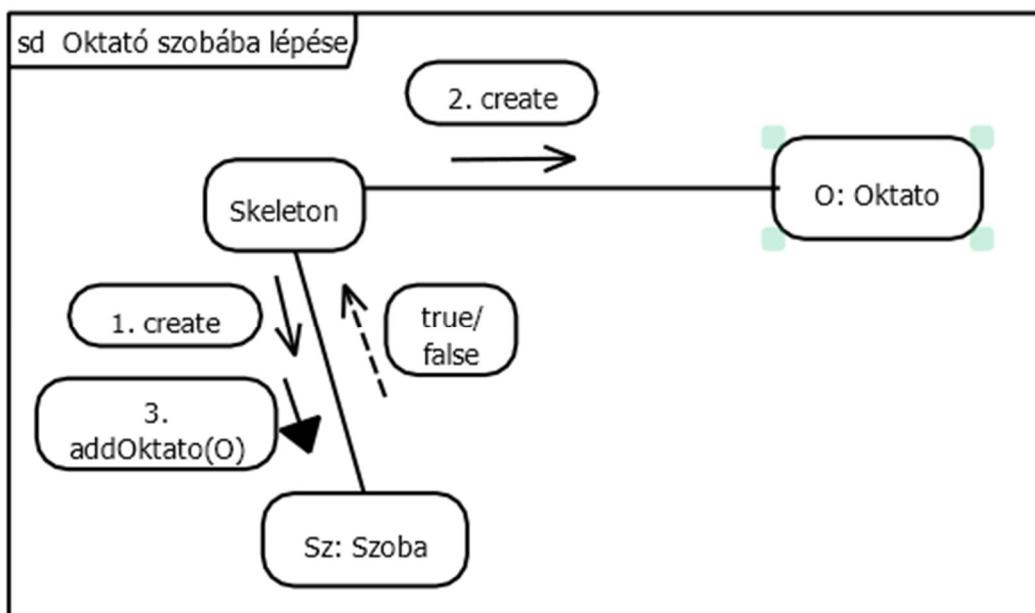


5.4.7 Játék vége logarléc felvétele után

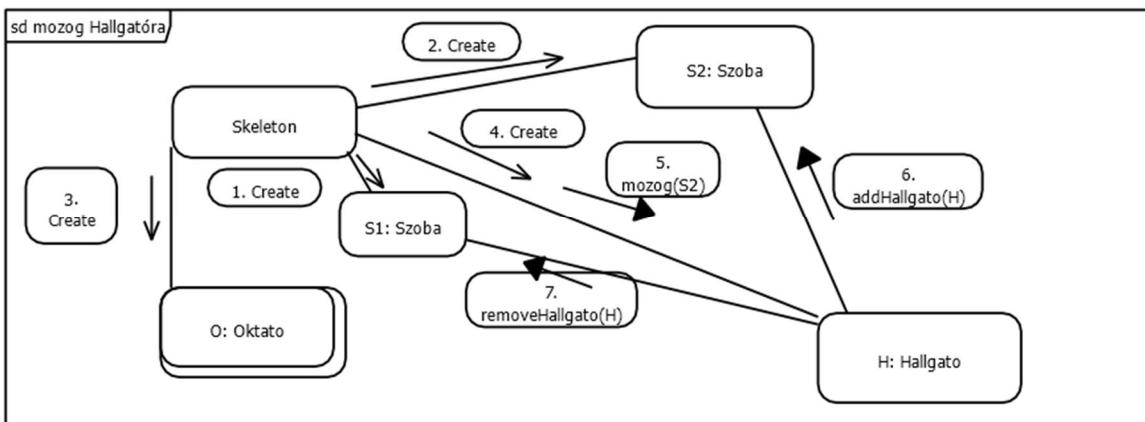


5.3.8 Oktató mozgása

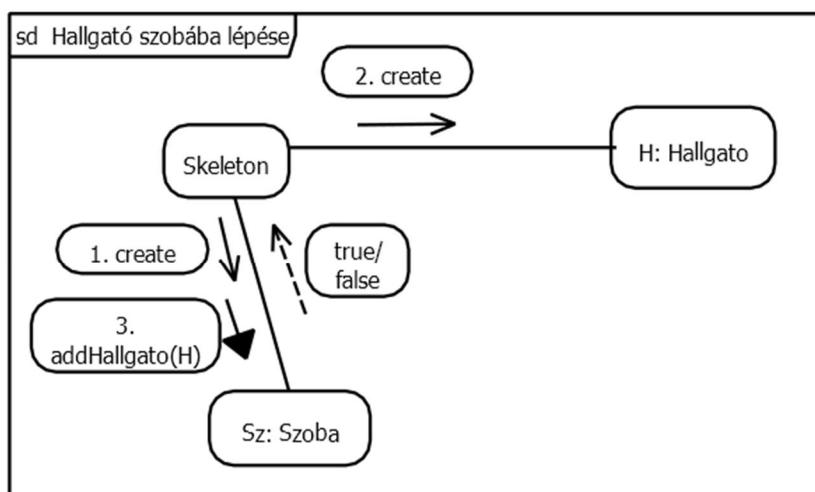


5.3.9 Oktató szobából kilépése**5.3.10 Oktató szobába lépése**

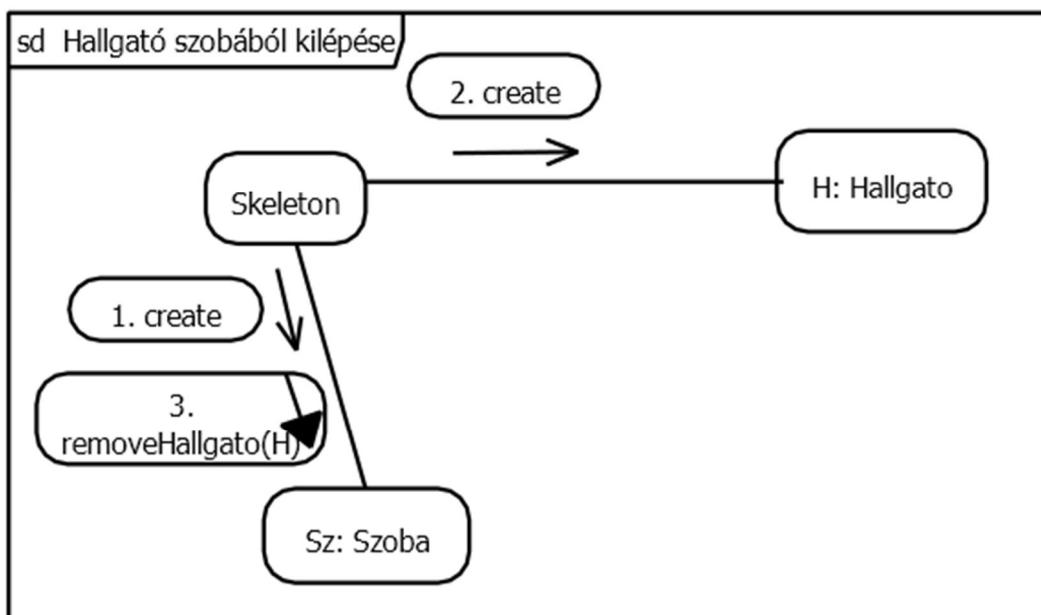
5.3.11 Hallgató mozgása



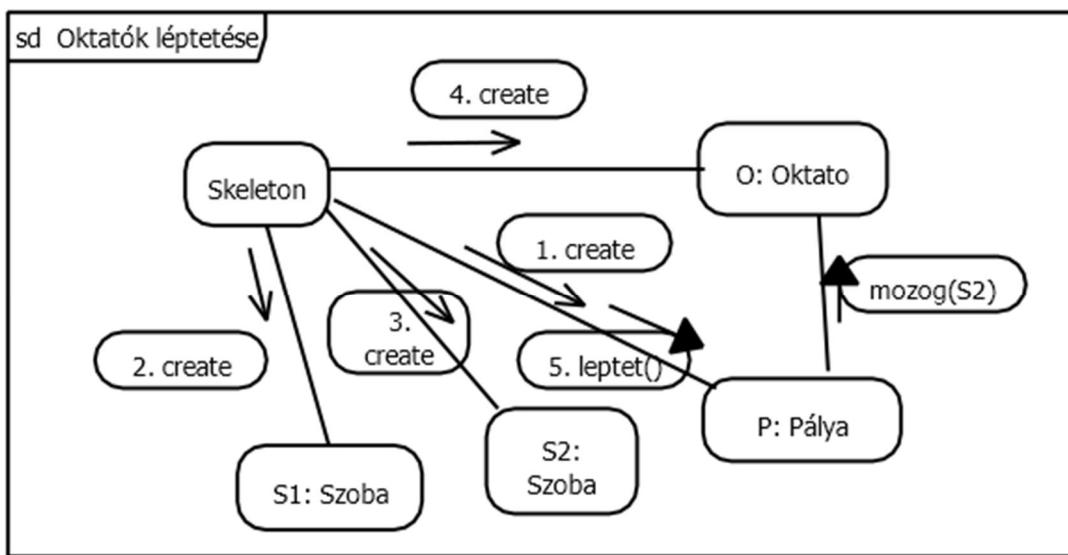
5.3.12 Hallgató szobába lépése



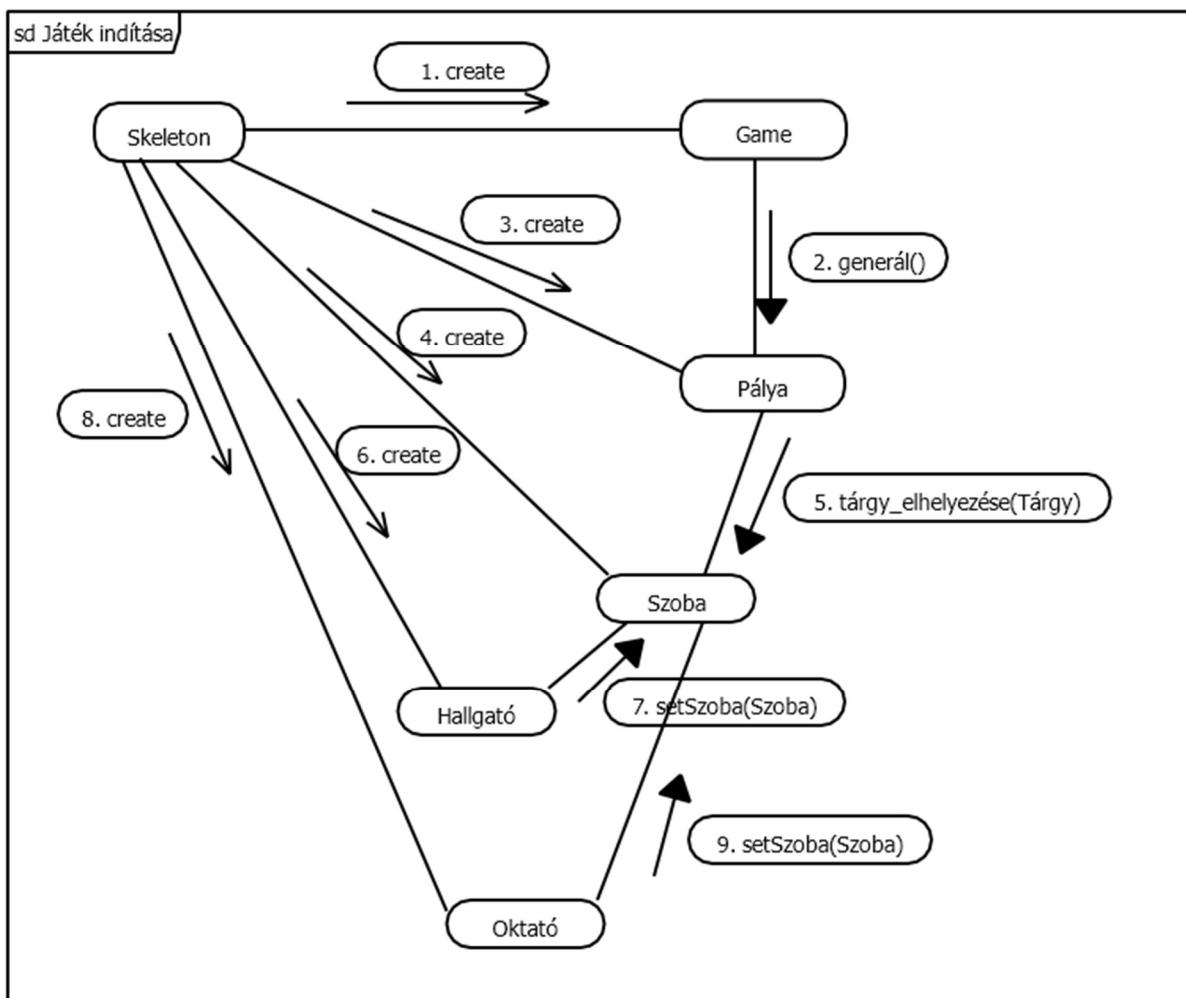
5.3.13 Hallgató szobából kilépése



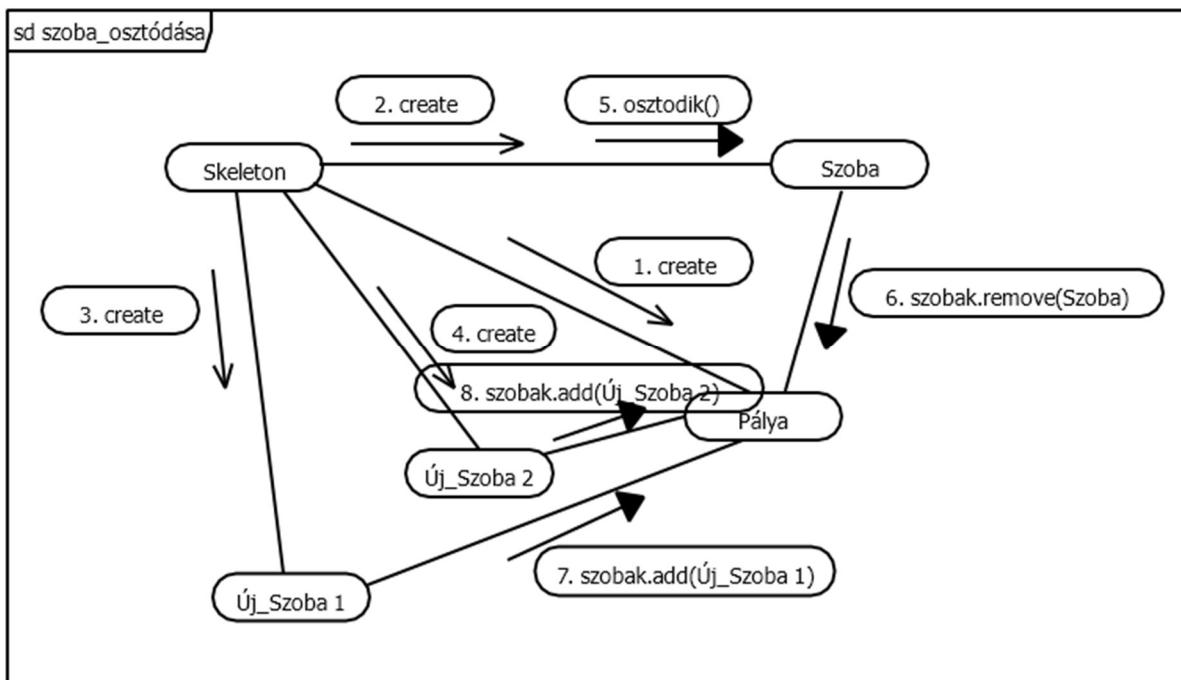
5.3.14 Oktatók léptetése



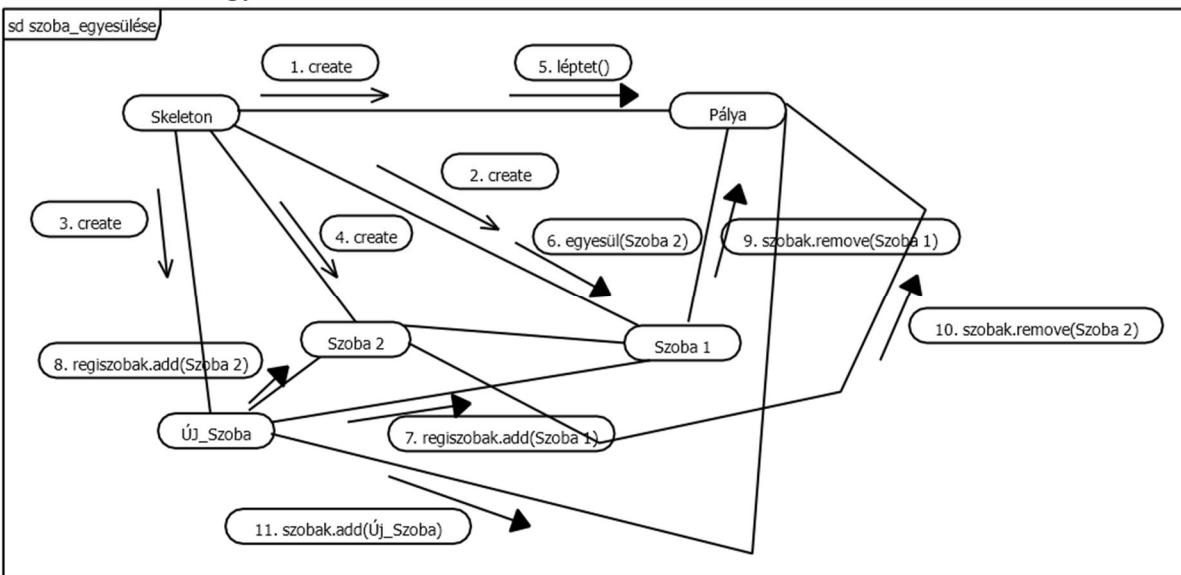
5.3.15 Játék indítása



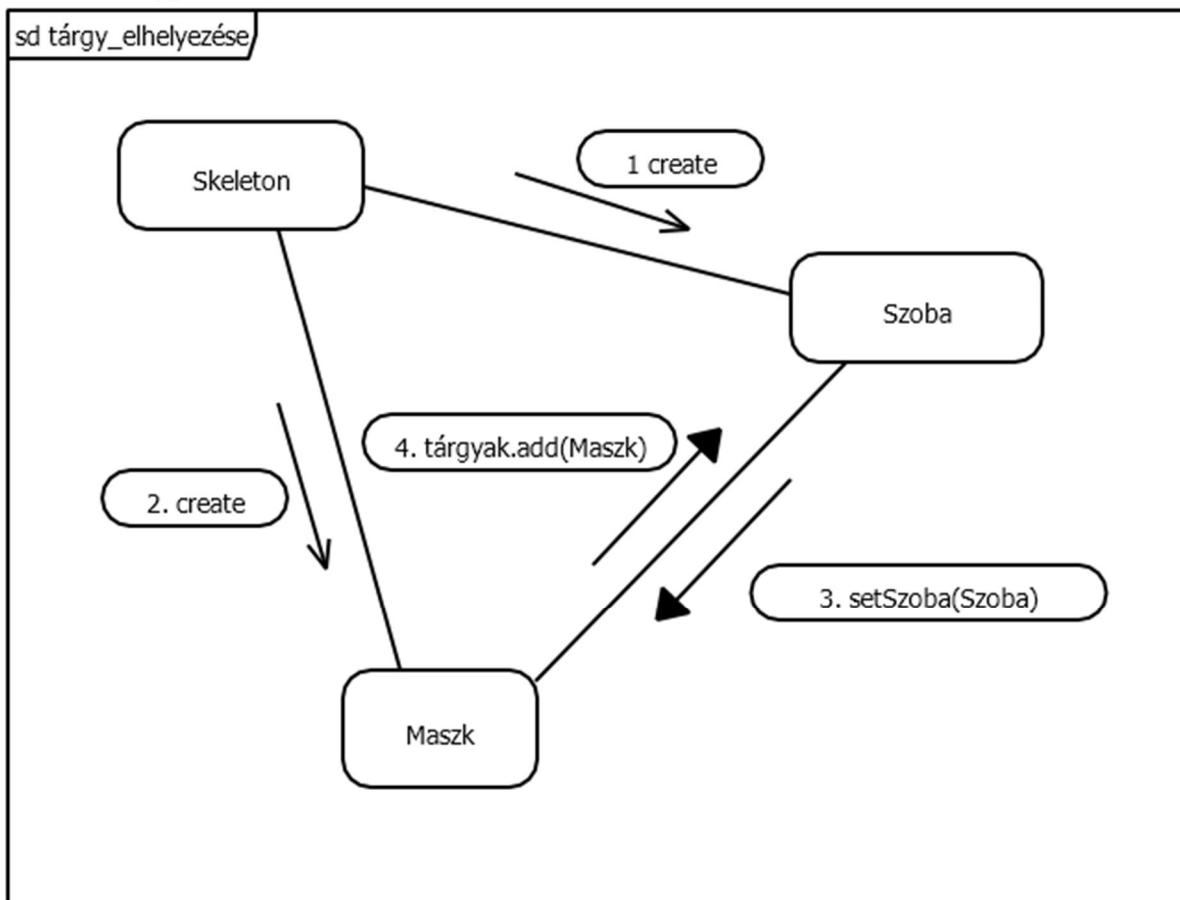
5.3.16 Szoba osztódása



5.3.17 Szobák egyesülése

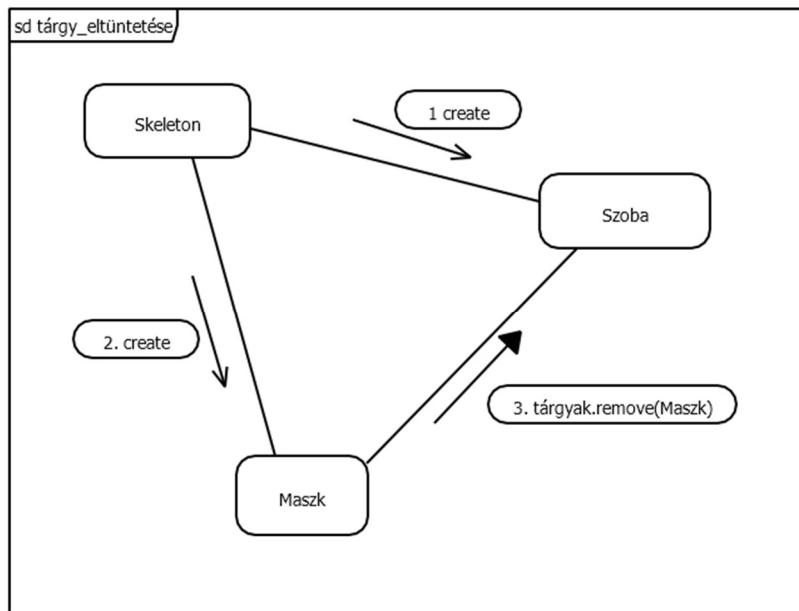


5.3.18 Tárgy elhelyezése



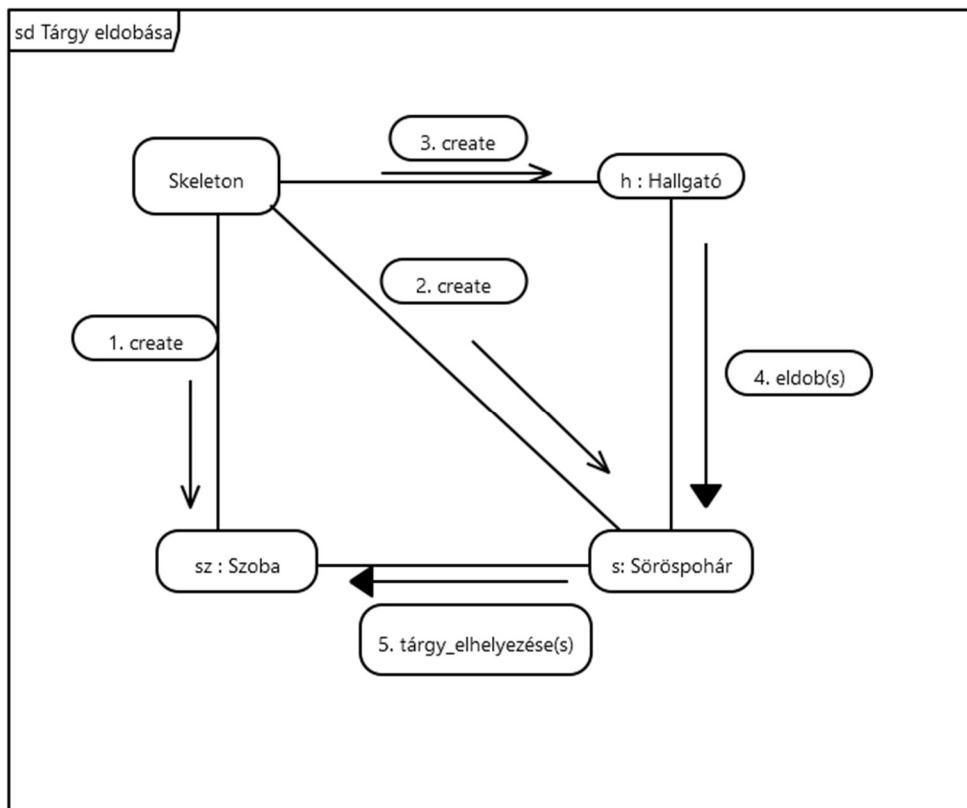
Az összes többi tárgy elhelyezése azonos a fenti példában szereplő Maszk tárgy elhelyezésével.

5.3.19 Tárgy eltüntetése



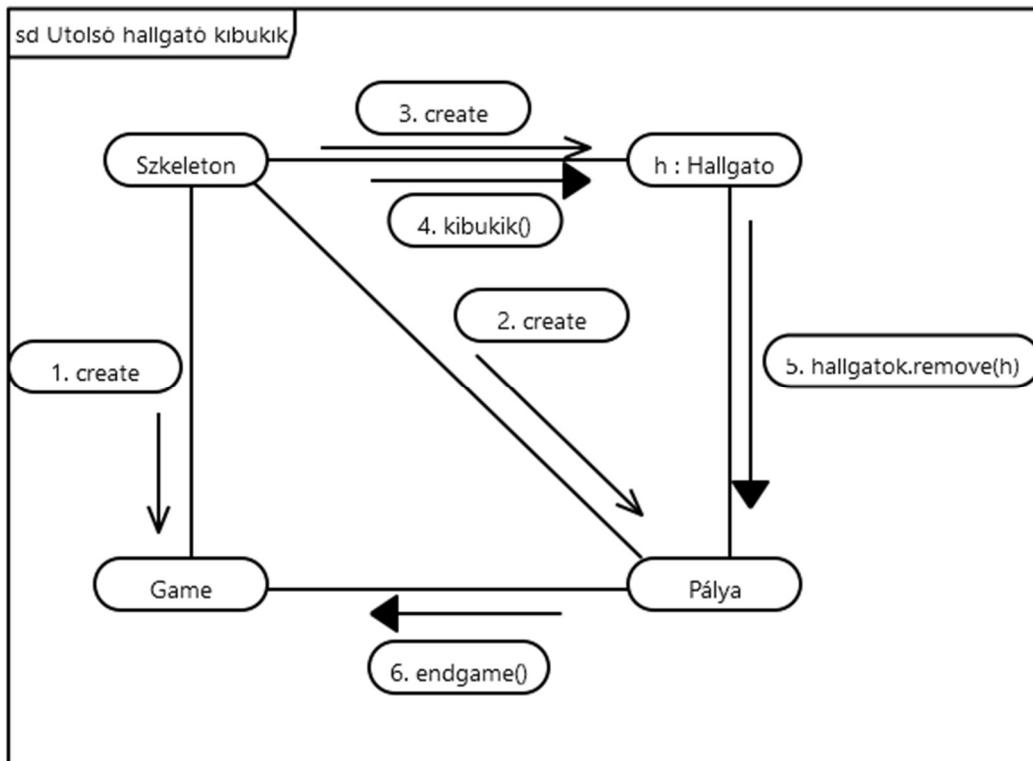
Az összes többi tárgy eltüntetése azonos a fenti példában szereplő Maszk tárgy eltüntetésével.

5.3.20 Tárgy eldobása

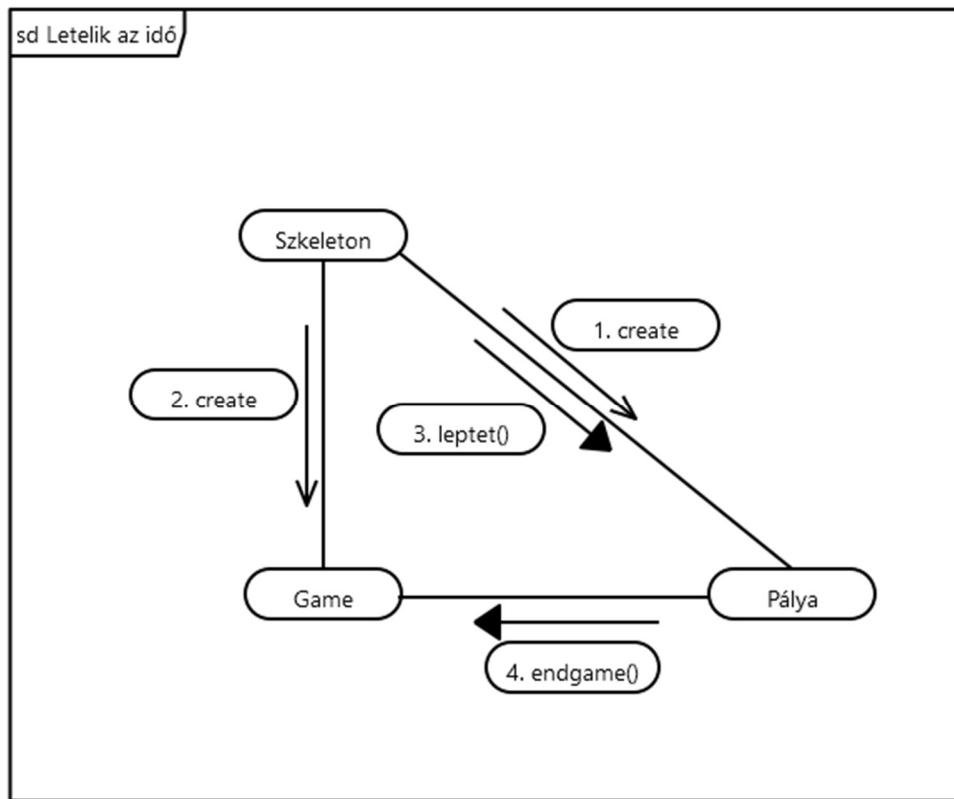


Összes tárgynál ugyanez a helyzet.

5.3.21 Utolsó hallgató kibukik

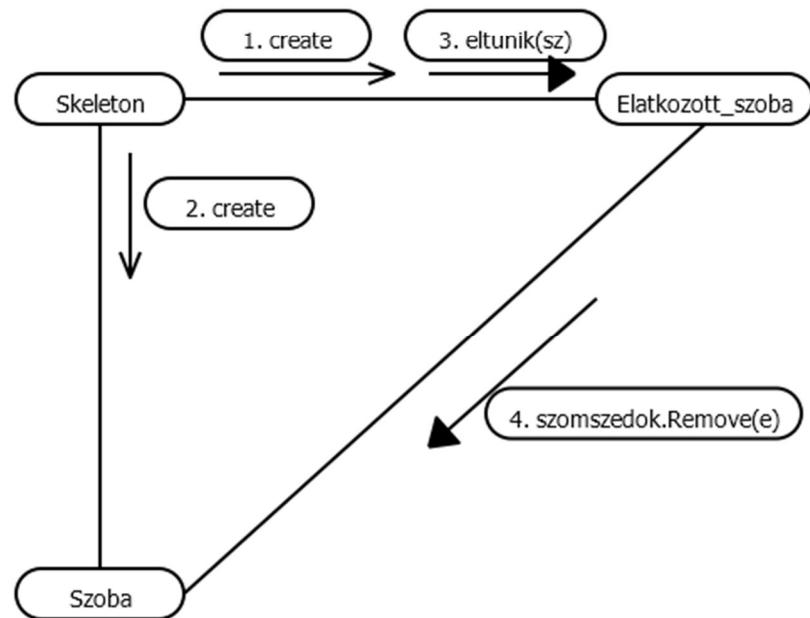


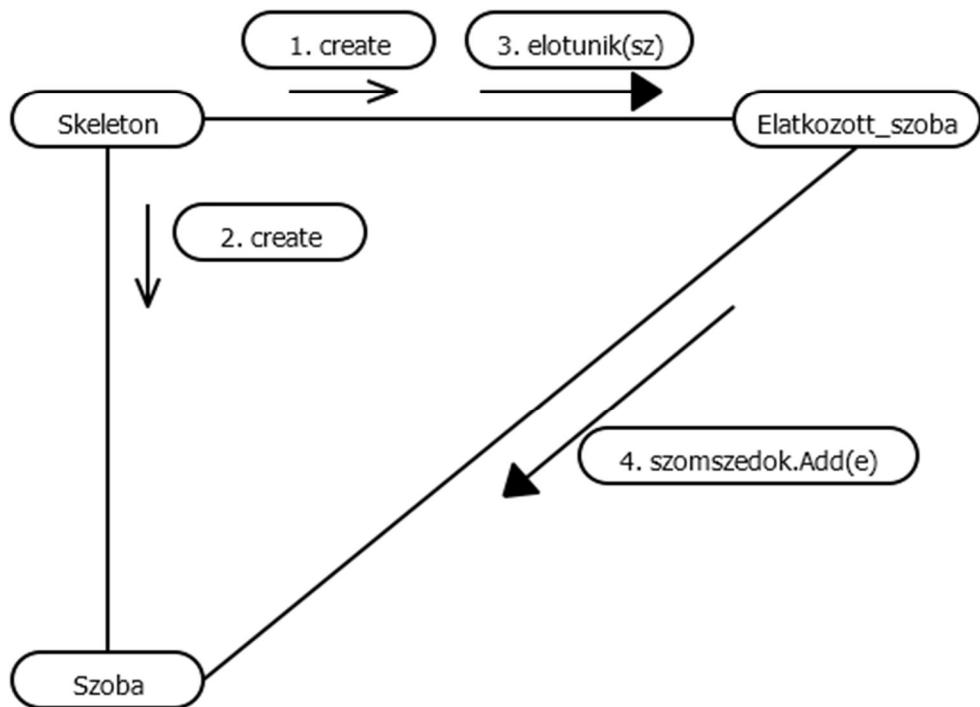
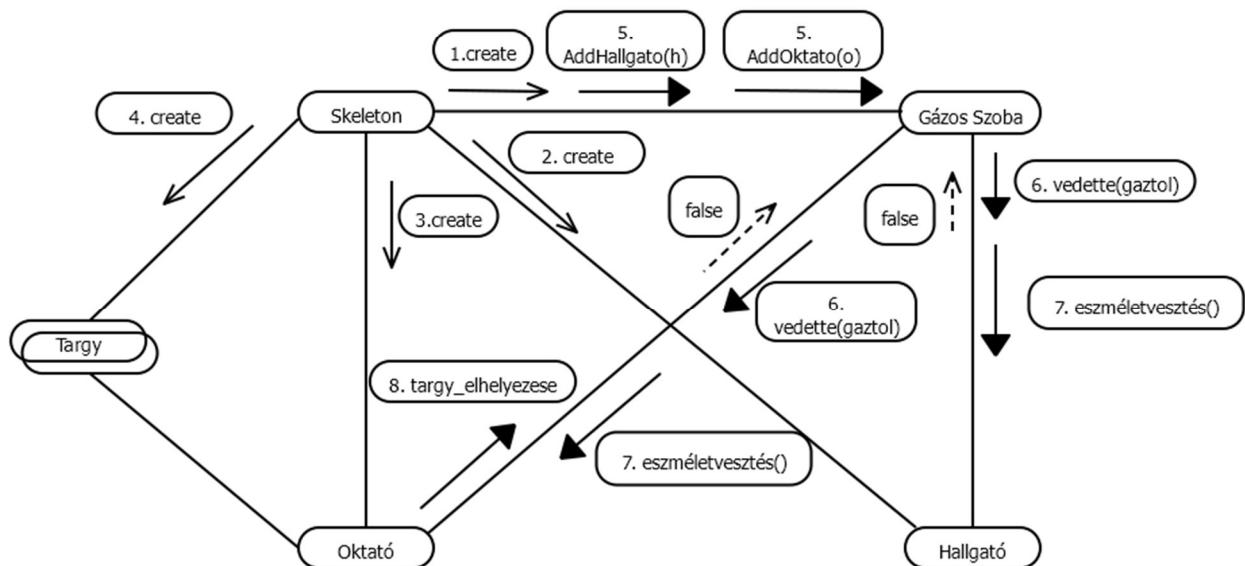
5.3.22 Letelik az idő



5.3.23 Ajtó eltűnése

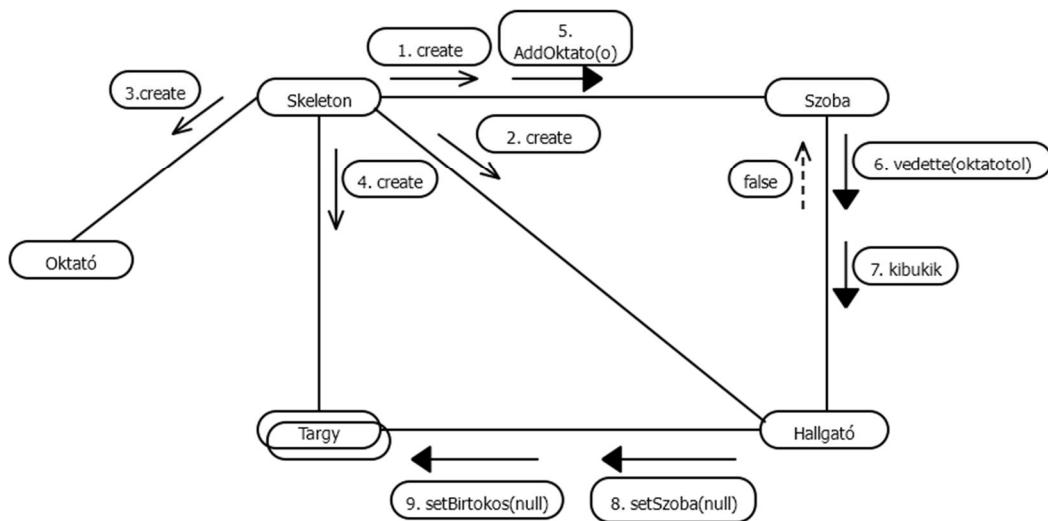
com Ajtó eltűnése



5.3.24 Ajtó előtűnése**com Ajtó előtűnése****5.3.25 Eszméletvesztés****com Eszméletvesztés**

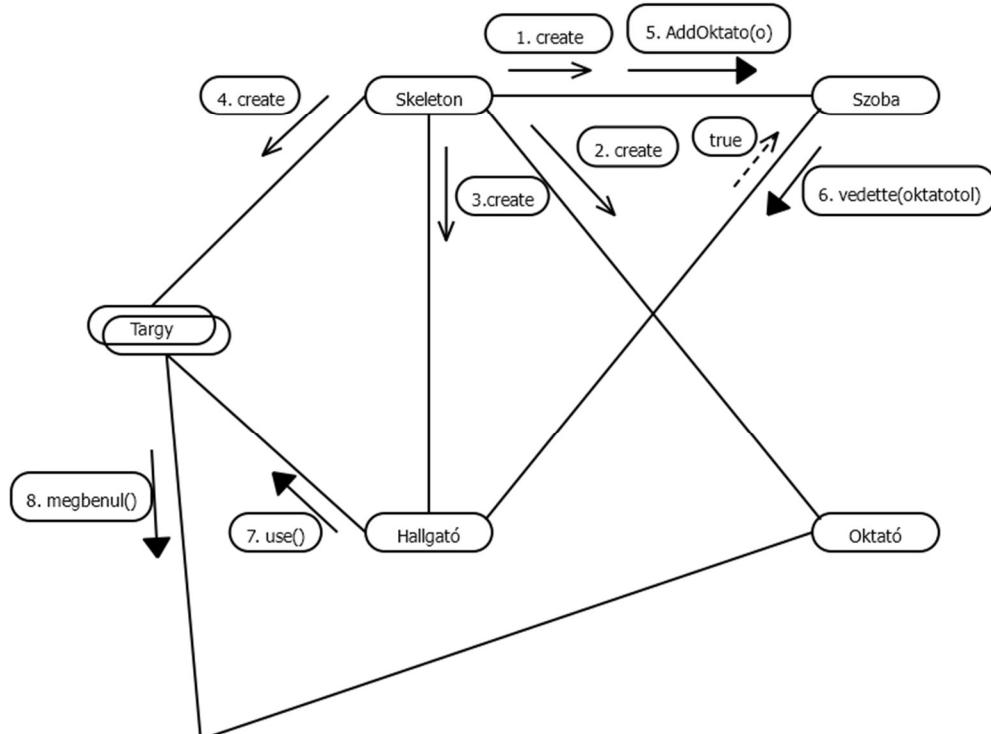
5.3.26 Hallgató kibuktatása

com Hallgató kibuktatása



5.3.27 Oktató megbénítása

com Oktatо bénítása



5.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2024.03.13. 20:00	2 óra	Csapat	Értekezlet. Döntés: Use-case-ek megbeszélése
2024.03.14. 21:30	1 óra	Csapat	Értekezlet. Döntés: Use-case-ek megbeszélése, feladatok kiosztása
2024.03.17. 21:00	1,5 óra	Csapat	Értekezlet. Döntés: Egyeztetés a kiosztott feladatokkal kapcsolatosan
2024.03.17. 22:00	3 óra	Németh	Use-case, szekvencia és kommunikációs diagramok
2024.03.17. 22:30	3 óra	Schulcz	Use-case, szekvencia és kommunikációs diagram.
2024.03.17. 22:30	3 óra	Czotter	Use-casek írása, szekvencia és kommunikációs diagramok.
2024.03.17. 22:30	3 óra	Tarsoly	Use-case, szekvencia és kommunikációs diagramok
		Hermann	
2024.03.18 11:00	1 óra	Csapat	Értekezlet. Döntés: Utolsó simítások megbeszélése

Szkeleton beadás

35 – pupakok

Konzulens:

Koczó Attila

Csapattagok:

Tarsoly Levente	E1IK75	tarsoly1@edu.bme.hu
Németh Gergely	TA9G0R	gergo.h.nemeth@gmail.com
Schulcz Gábor	EP3I9R	schulcz.gabor6@gmail.com
Czotter Benedek	TFB4FY	czottibeni@gmail.com
Hermann Máté Béla	T83K5I	mate3829@gmail.com

2024.03.25

6.1 Fordítási és futtatási útmutató

6.1.1 Fájllista

Fájl neve	Méret (KB)	Keletkezés ideje	Tartalom
Main.java	1	2024. 03. 24.	Főprogram, teszteket futtatja
Game.java	2	2024. 03. 24.	Game osztály implementációja
Karakter.java	3	2024. 03. 24.	Karakter osztály megvalósítása
Hallgato.java	3	2024. 03. 24.	Hallgato osztály megvalósítása
Oktato.java	2	2024. 03. 24.	Oktato osztály megvalósítása
Vedettseg.java	1	2024. 03. 24.	Enum a védettség eldöntésére
Palya.java	4	2024. 03. 24.	Palya osztály implementációja
Szoba.java	9	2024. 03. 24.	Szoba osztály implementációja
ElatkozottSzoba.java	2	2024. 03. 24.	ElatkozottSzoba osztály megvalósítása
Targy.java	2	2024. 03. 24.	Targy osztály megvalósítása
AktivTargy.java	1	2024. 03. 24.	AktivTargy osztály megvalósítása
PasszivTargy	1	2024. 03. 24.	PasszivTargy osztály implementációja

Camembert.java	1	2024. 03. 24.	Camembert osztály megvalósítása
Funkcio.java	1	2024. 03. 24.	Enum a tárgyak funkcióinak eldöntésére
Logarlec.java	2	2024. 03. 24.	Logarlec osztály megvalósítása
LogarlecPropertyChangeListener.java	1	2024. 03. 24.	LogarlecPropertyChangeListerner osztály megvalósítása
Maszk.java	1	2024. 03. 24.	Maszk osztály implementációja
Rongy.java	1	2024. 03. 24.	Rongy osztály megvalósítása
Sorospohar.java	1	2024. 03. 24.	Sorospohar osztály megvalósítása
Tranzisztor.java	2	2024. 03. 24.	Tranzisztor osztály implementációja
TVSZ.java	1	2024. 03. 24.	TVSZ osztály megvalósítása
Skeleton.java	25	2024. 03. 24.	Skeleton osztály megvalósítása. Itt történik a tesztek és a tesztprogram implementációja.

6.1.2 Fordítás

1. Importáljuk a forrásfájlokat tetszőleges Java IDE-be, és hozzunk létre belőlük egy projektet
2. Fordítsuk az IDE beépített szolgáltatásával.

6.1.3 Futtatás

A fordított kódot futasd az IDE megfelelő parancsával (Pl.: VSCode-ban Ctrl+F5).

6.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Schulcz Gábor	EP3I9R	20%
Tarsoly Levente	E1IK75	25%
Németh Gergely	TA9G0R	15%
Czotter Benedek	TFB4FY	25%
Hermann Máté Béla	T83K5I	15%

6.3 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2024.03.20. 20:00	2 óra	Csapat	Értekezlet Döntés: Előző feladat hibáinak megbeszélése.
2024.03.21. 21:00	1 óra	Csapat	Értekezlet: Döntés: Feladatok kiosztása.
2024.03.22. 14:00	3,5 óra	Czotter	Skeleton program szerkesztése 34-45, tesztesetek implementálása.
2024.03.24. 15:00	7 óra	Tarsoly	Skeleton program szerkesztése, 0-9 tesztesetek implementálása.
2024.03.24. 15:00	4 óra	Schulcz	22-33 tesztesetek implementálása
2024.03.24. 22:00	3.5 óra	Németh	46-57 tesztesetek implementálása
2024.03.25 8:00	2 óra	Hermann	10-21 tesztesetek implementálása, dokumentum formázása

Prototípus koncepciója

35 – pupakok

Konzulens:
Koczó Attila

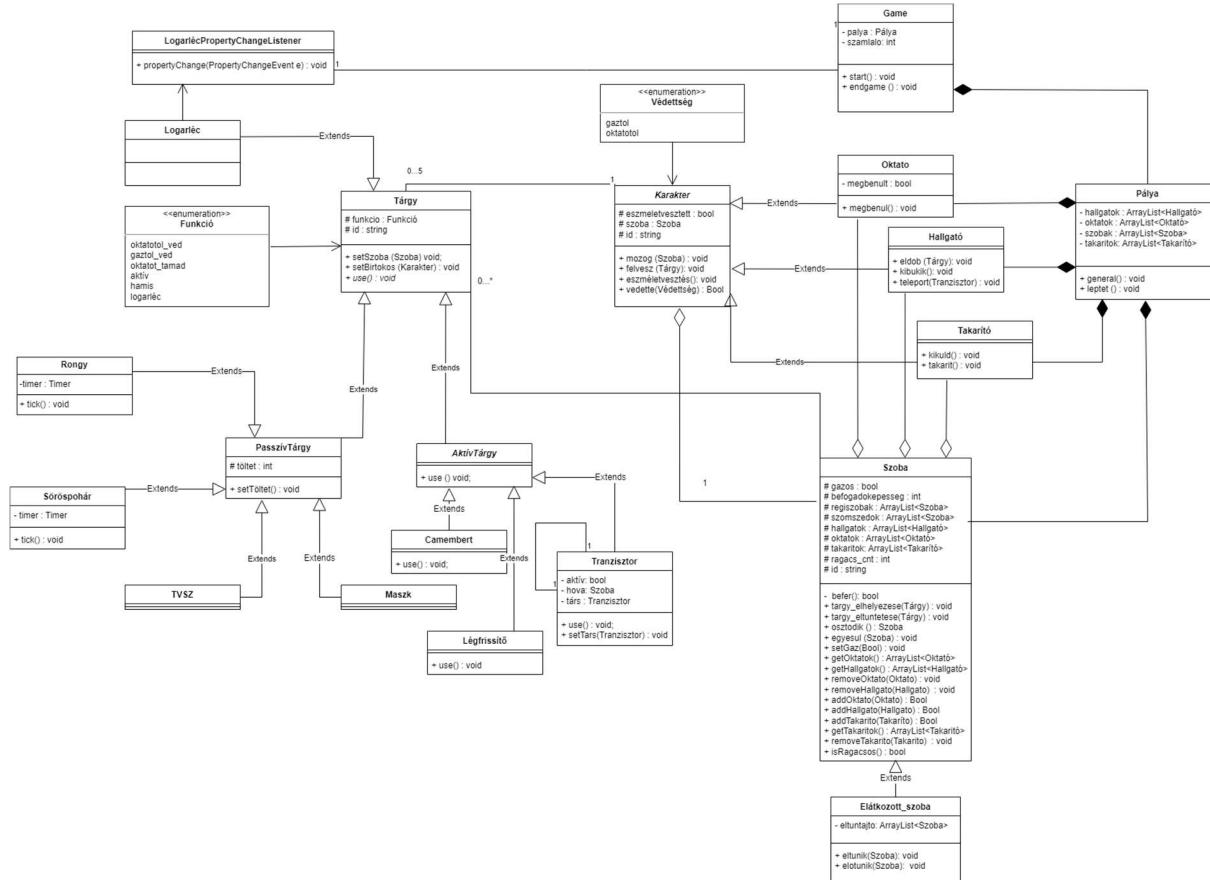
Csapattagok:

Tarsoly Levente	E1IK75	tarsolyl@edu.bme.hu
Németh Gergely	TA9G0R	gergo.h.nemeth@gmail.com
Schulcz Gábor	EP3I9R	schulcz.gabor6@gmail.com
Czotter Benedek	TFB4FY	czottibeni@gmail.com
Hermann Máté Béla	T83K5I	mate3829@gmail.com

2024.04.08.

7. Prototípus koncepciója

7.1.1 Módosult osztálydiagram



7.1.2 Új vagy megváltozó metódusok

Söröspohár osztály:

void use(): A söröspohár osztály use metódusa megváltozik és ez lesz felelős azért, hogy miután megvédi a hallgatót egy oktatótól, a hallgató egy nála lévő tárgyat eldobjon.

Karakter osztály:

void felvesz(Tárgy t): A karakter osztályban szereplő felvesz függvény, és ezen metódus felüldefiniált változatai a hallgató és oktató osztályokban oly módon változnak meg, hogy tárgyfelvétel előtt leellenőrzik, hogy az adott szoba ragacsos-e, és csak akkor lesz lehetőség felvenni tárgyat a szobából, ha nem ragacsos.

Tárgy osztály:

void use(): Absztrakt függvény, amely a tárgyak használatát írja le minden tárgynak. Egyes tárgyak (TVSZ, maszk, logarléc) esetén, ha az adott tárgy hamisnak van megjelölve, akkor a use() függvény funkcióját veszi, gyakorlatilag nem következik be változás a függvényhívást követően.

Szoba osztály:

- **Attribútumok**
- **int ragacs_cnt:** A szoba ragacsosságát számolja, alapesetben -1, ekkor ha belép valaki, nem nő az értéke. Ha kitakarítanak, 0 lesz és innen kezd felfele számlálni.
- **Metódusok**
- **bool addTakarito(Takarito):** hozzáad egy takarítót a listához.
- **void removeTakarito():** kitöröl egy takarítót a listából.
- **ArrayList<Takarito> getTakaritok():** visszaadja a szobában tartózkodó takarítók listáját
- **bool IsRagacsos():** visszaadja, hogy egy szoba ragacsos-e, ami azt jele

Takarító

- **Felelősség**

Ez az osztály felel a játék által irányított karakter, az oktató tulajdonságaiért.

- **Ősosztályok**

Karakter

- **Interfészek**

- **Asszociációk**

- **Attribútumok**

- **Metódusok**

- **void kikuld()** : a takarító a vele egy szobában tartózkodó más karaktereket kitessékel a szobából, emiatt átkerülnek egy másik szobába
- **void takarit()** : a takarítónő a jelenlegi szobában ha gázos, kiszellőztet, emiatt nem lesz többet gázos, illetve kitakarít, ennek köszönhetően ha egy adott számú karakter belép a szobába, ragacsossá válik, és nem lehet felvenni a tárgyakat

Légfrissítő

- **Felelősség**

Ez az osztály felel a légfrissítő tárgy tulajdonságaiért.

- **Ősosztályok**

Aktív tárgy, Tárgy

- **Interfészek**

- **Asszociációk**

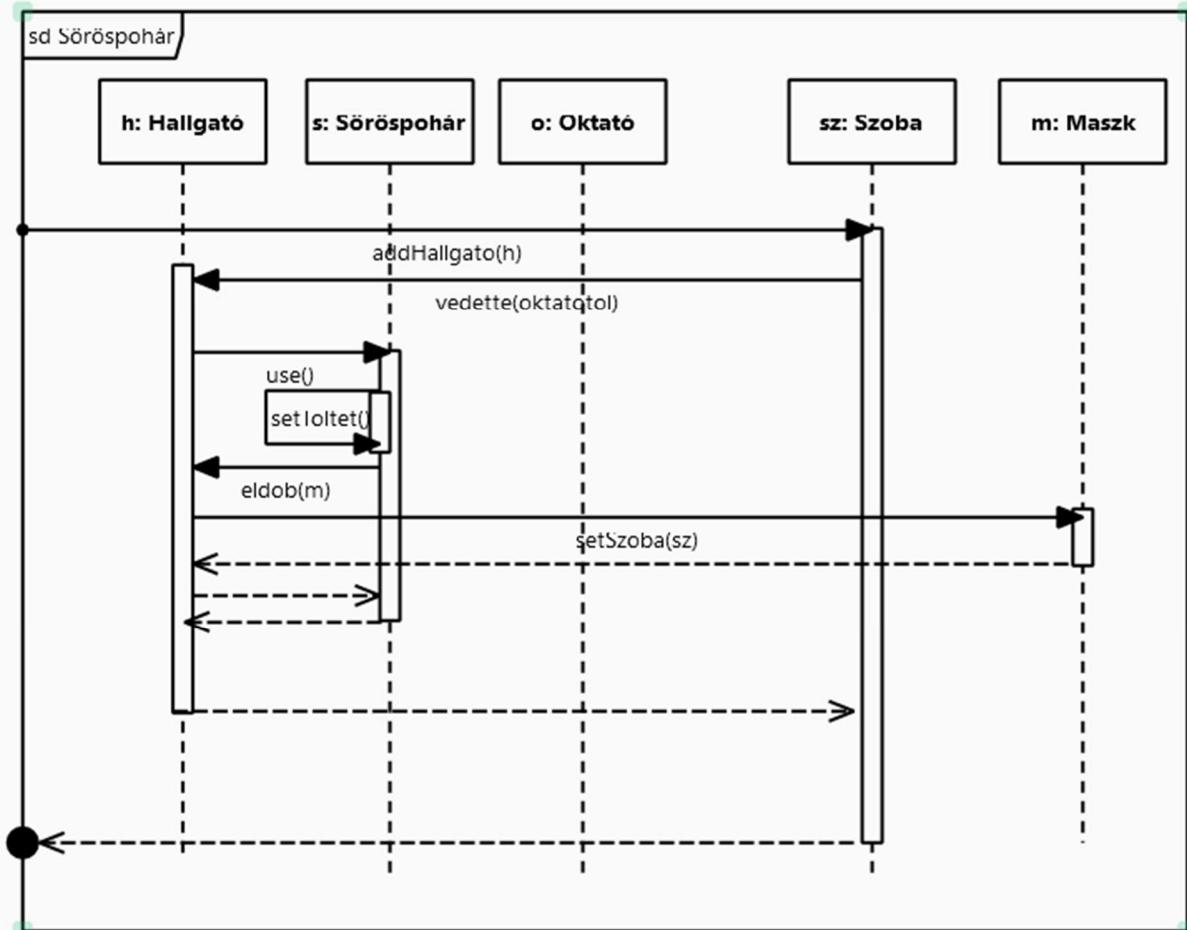
- **Attribútumok**

- **Metódusok**

- void use() : ez a függvény hajtja végre a tárgy képességét, vagyis, hogy a gázt eltünteti a szobából.

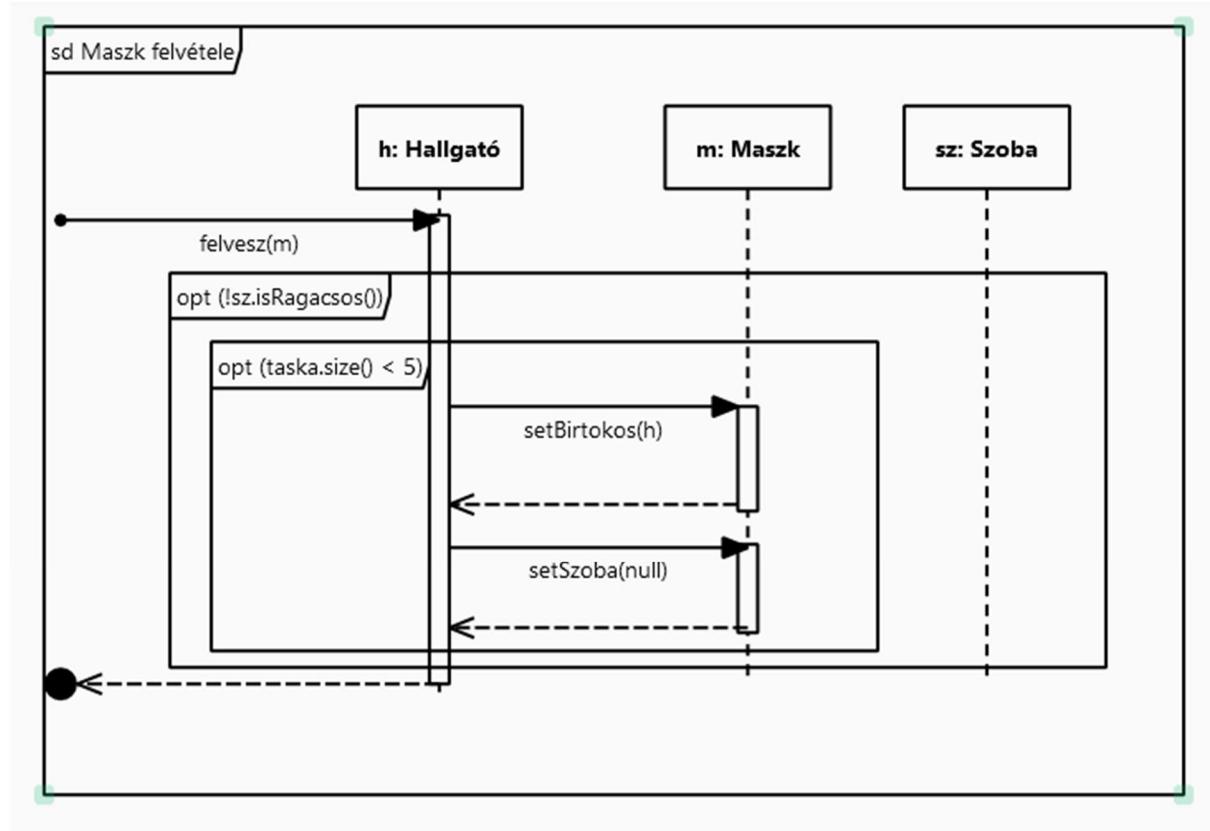
7.1.3 Szekvencia-diagramok

7.0.3.1 Söröspohár használata



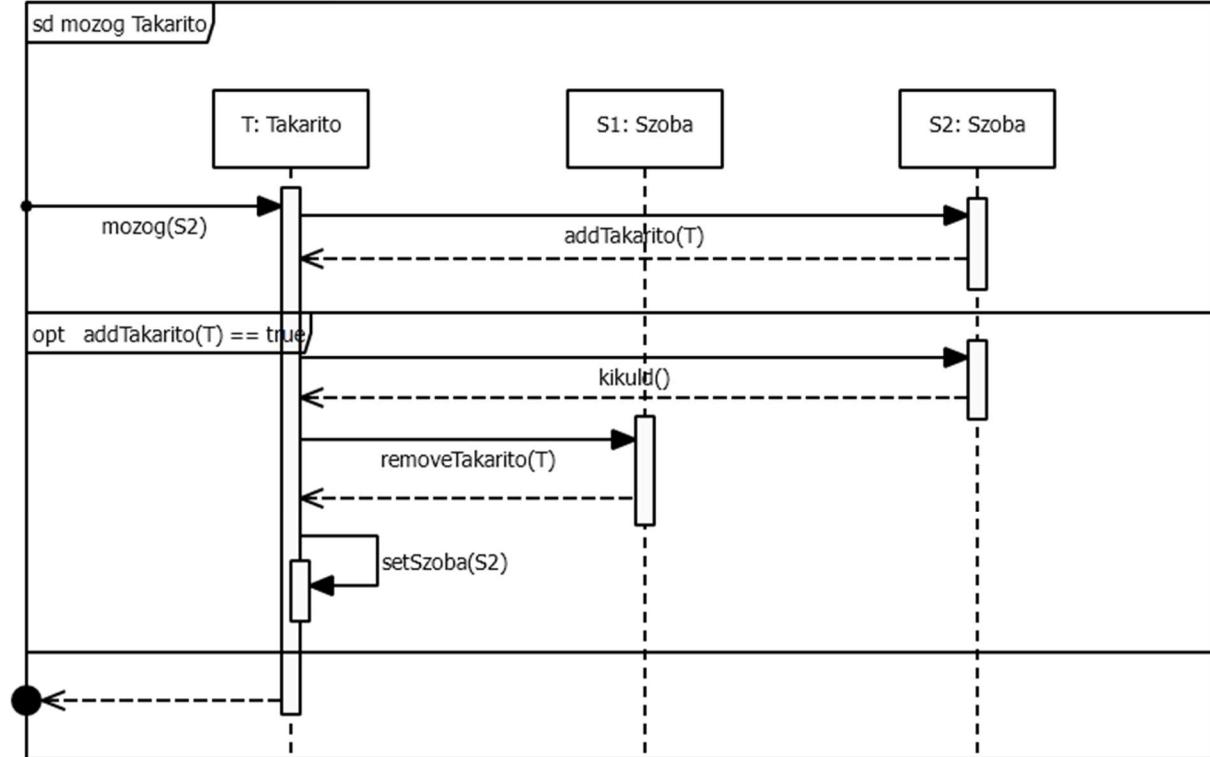
A Maszk ebben az esetben egy a hallgatónál lévő random tárgyat jelöl, bármilyen más tárgy is lehetne.

7.0.3.2 Tárgyfelvétel

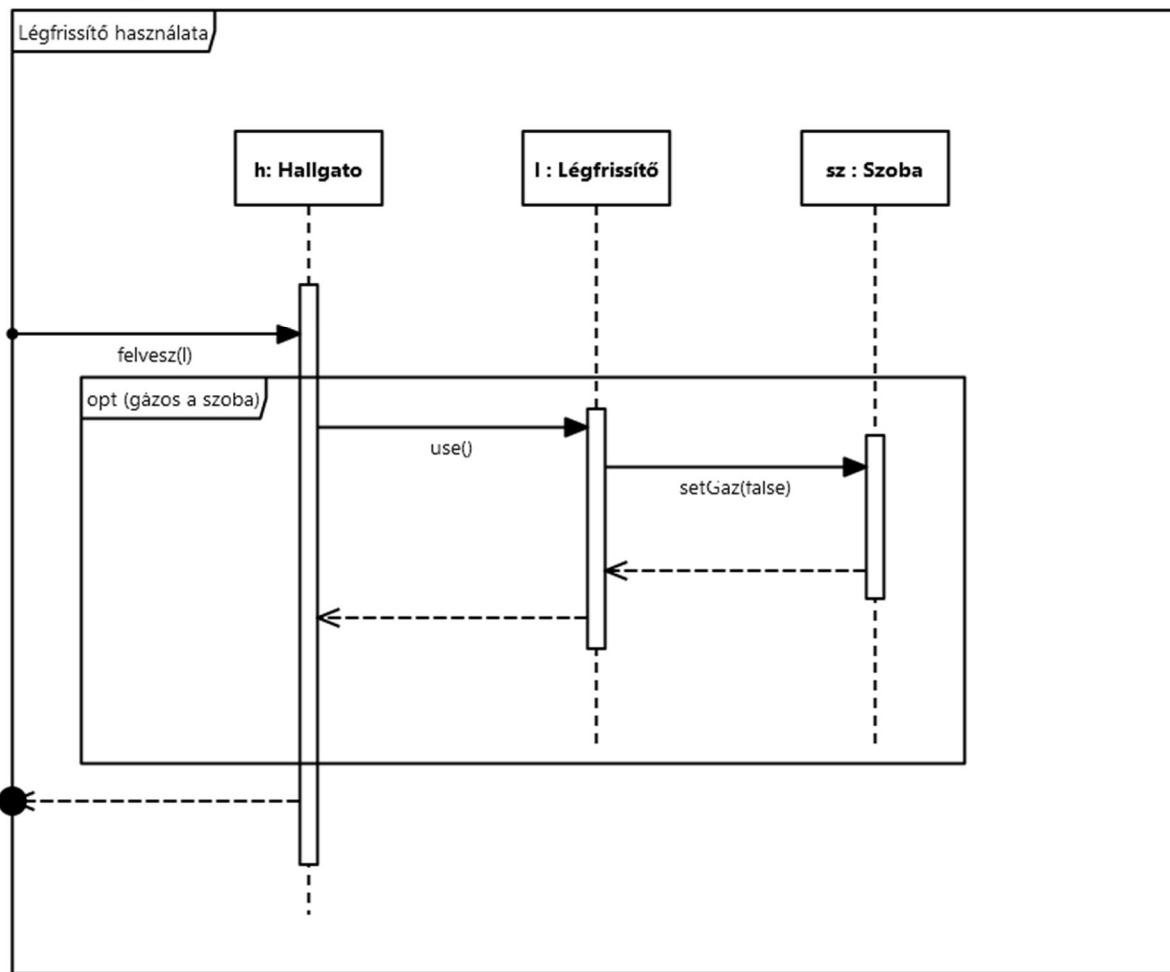


A tárgyfelvétel minden tárgy esetén hasonlóképpen zajlik.

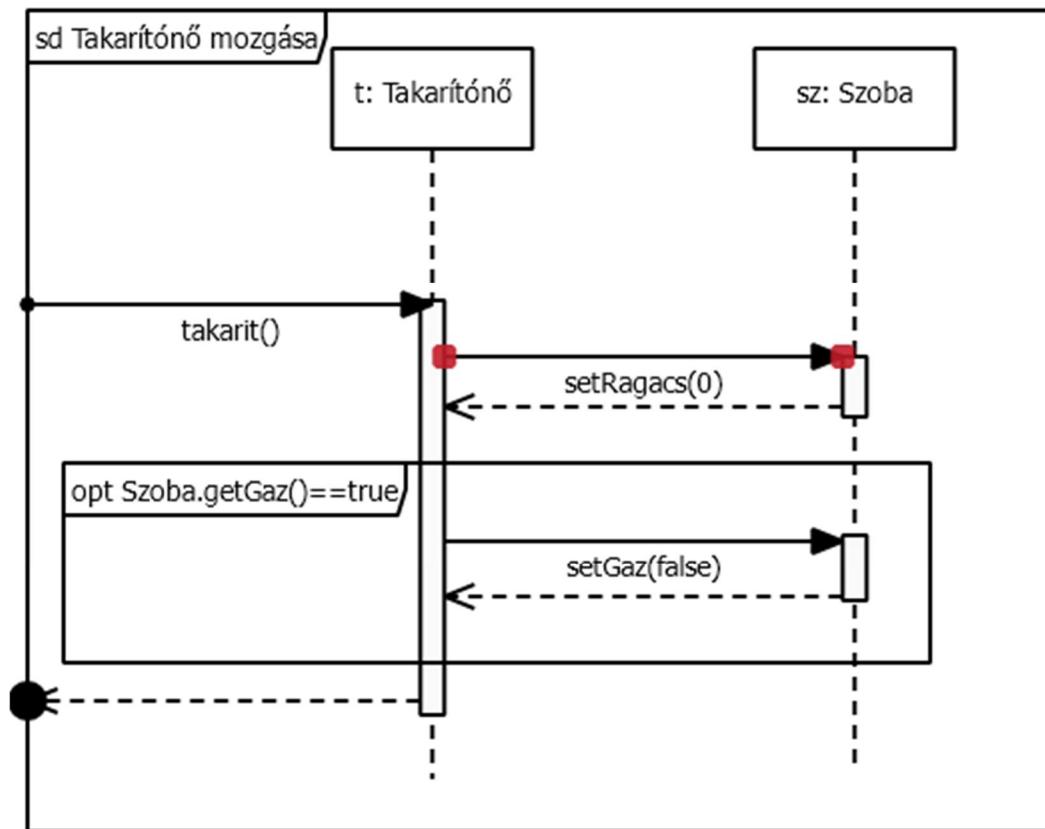
7.0.3.3 Takarító mozgása



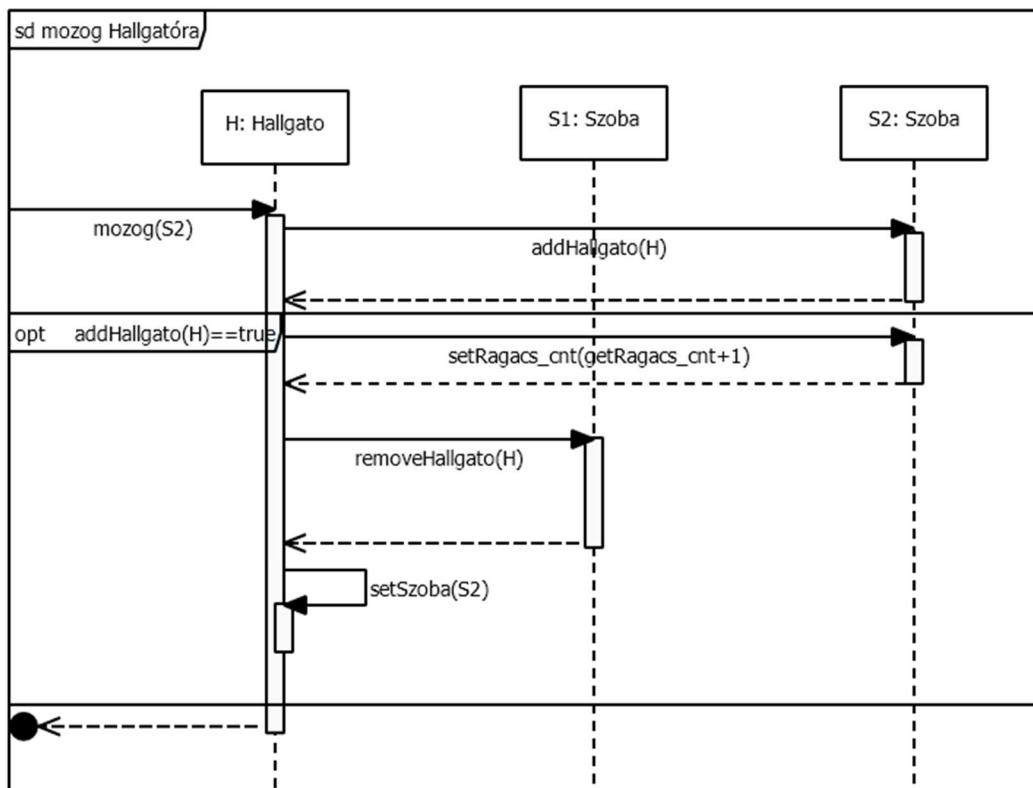
7.0.3.4 Légfrissítő használata



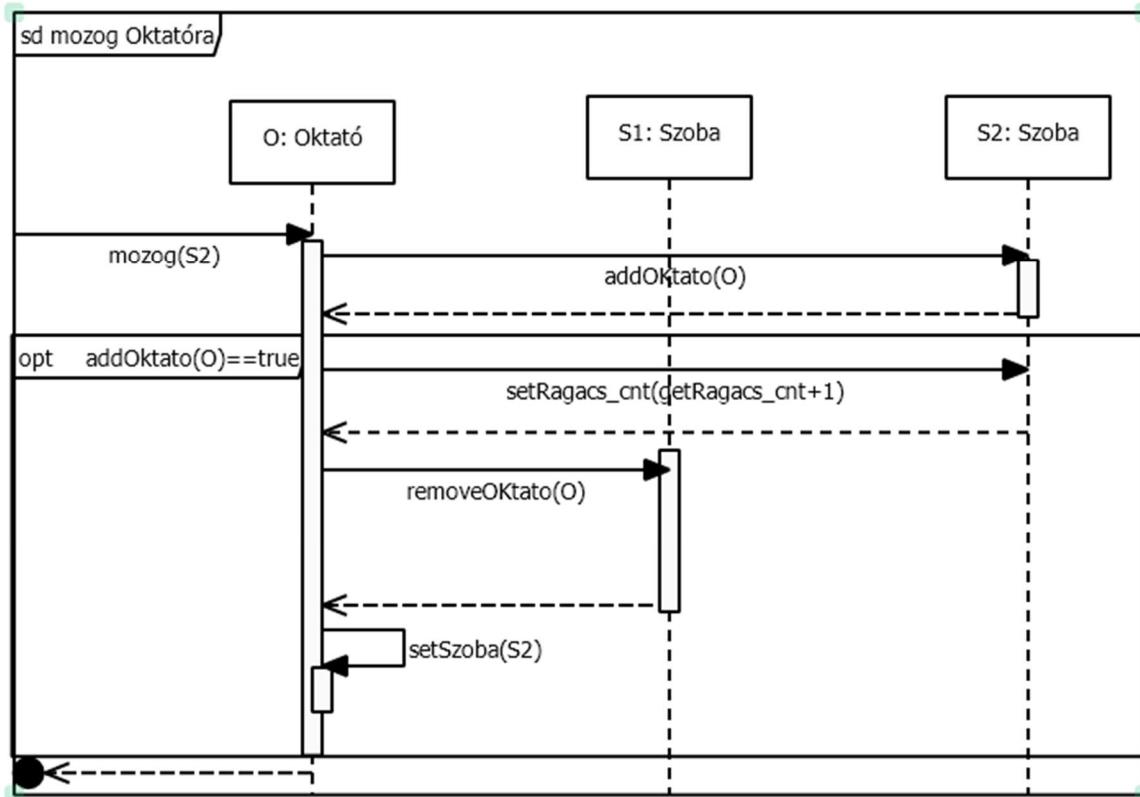
7.3.0.5 Takarítónő takarítása



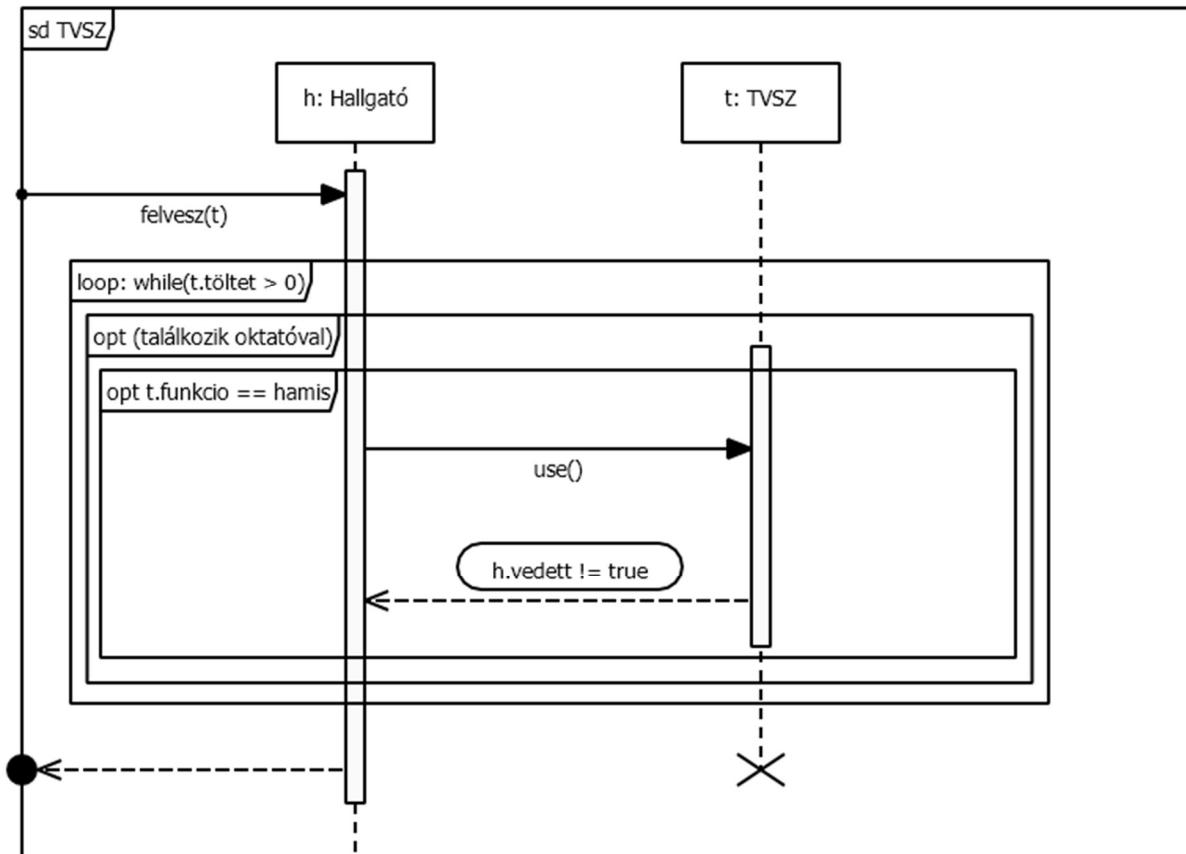
7.3.0.6 Hallgató mozgása



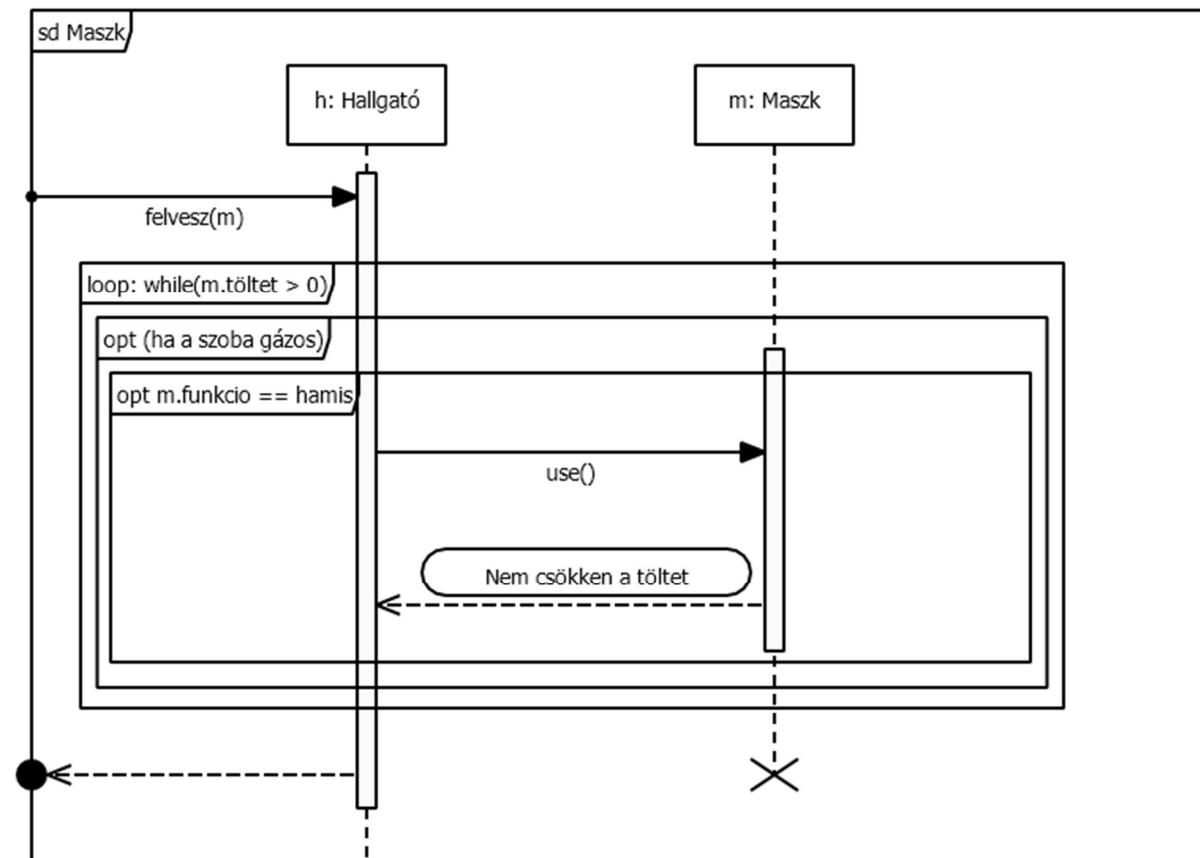
7.3.0.6 Oktató mozgása



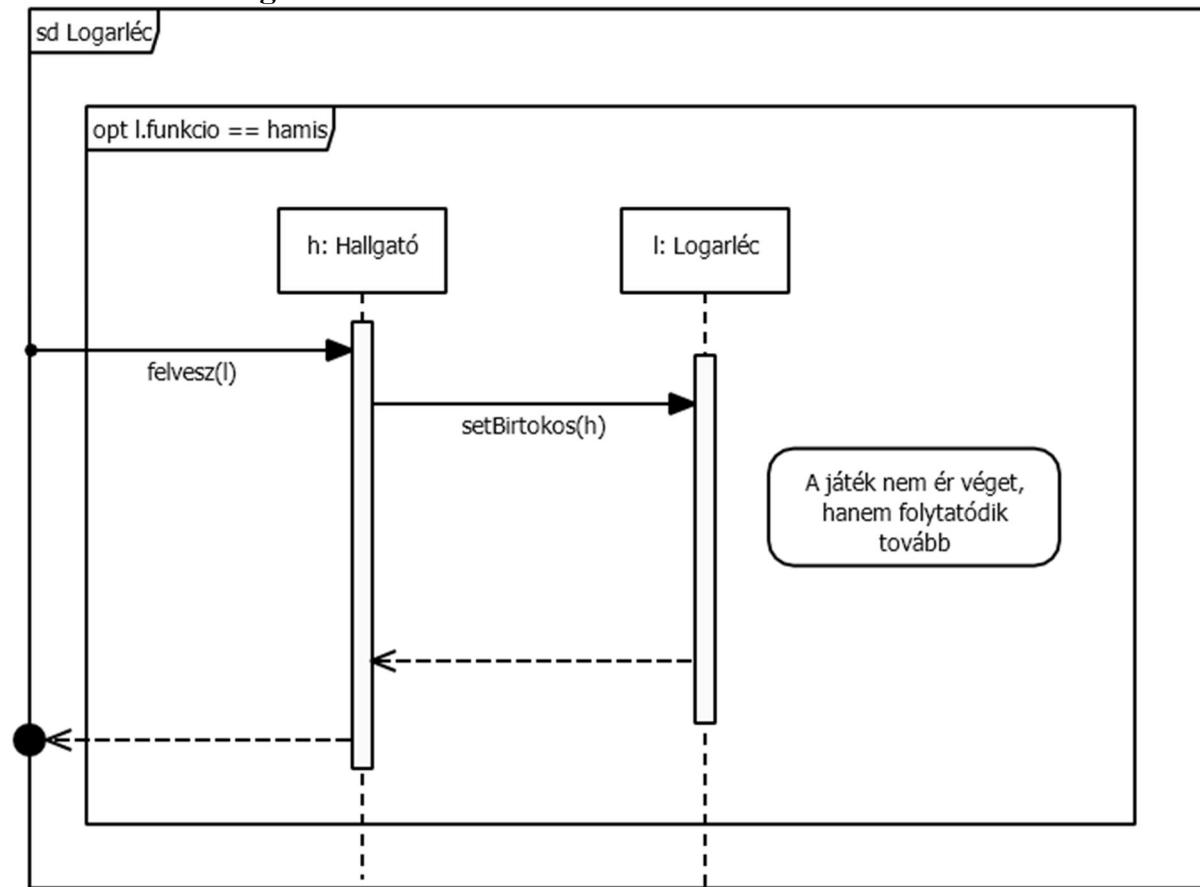
7.3.0.7 Hamis TVSZ használata



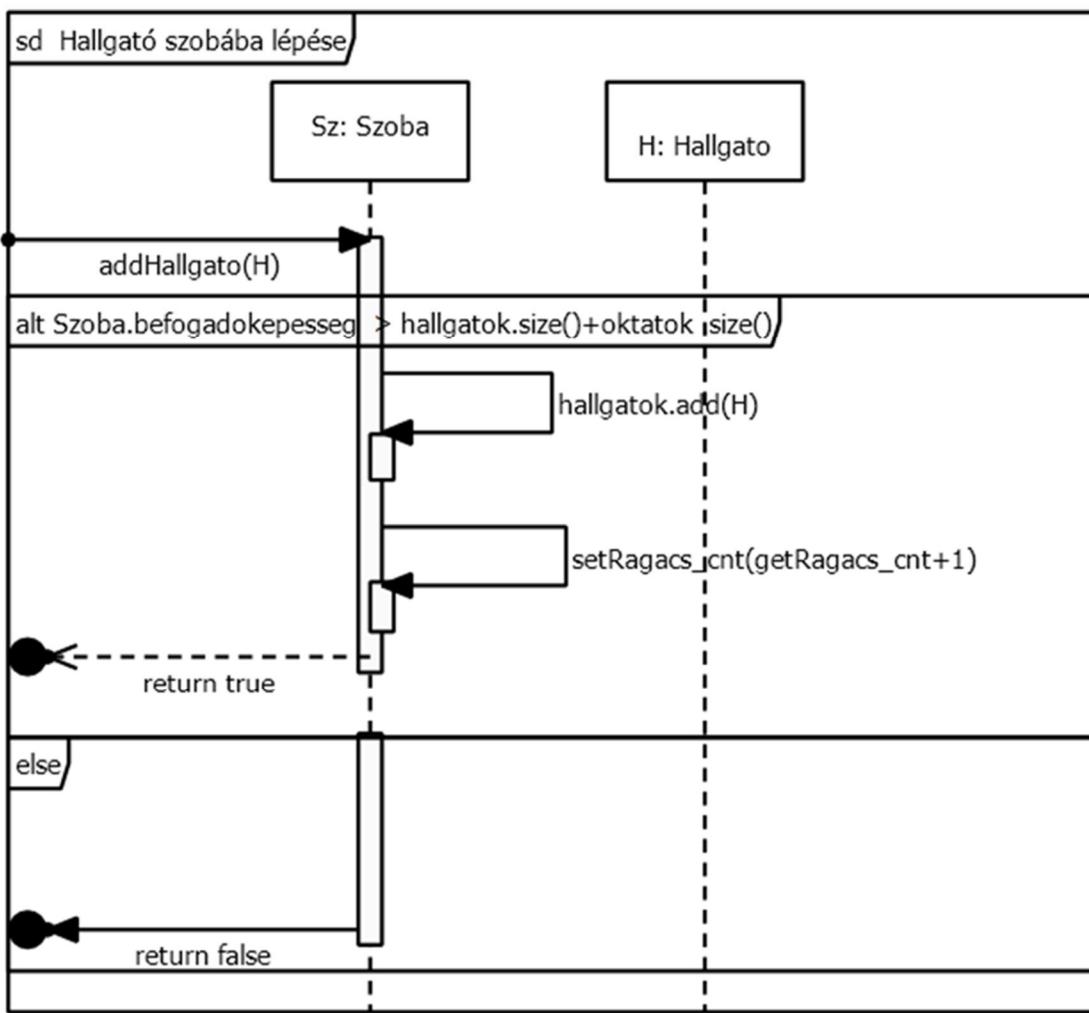
7.3.0.8 Hamis Maszk használata



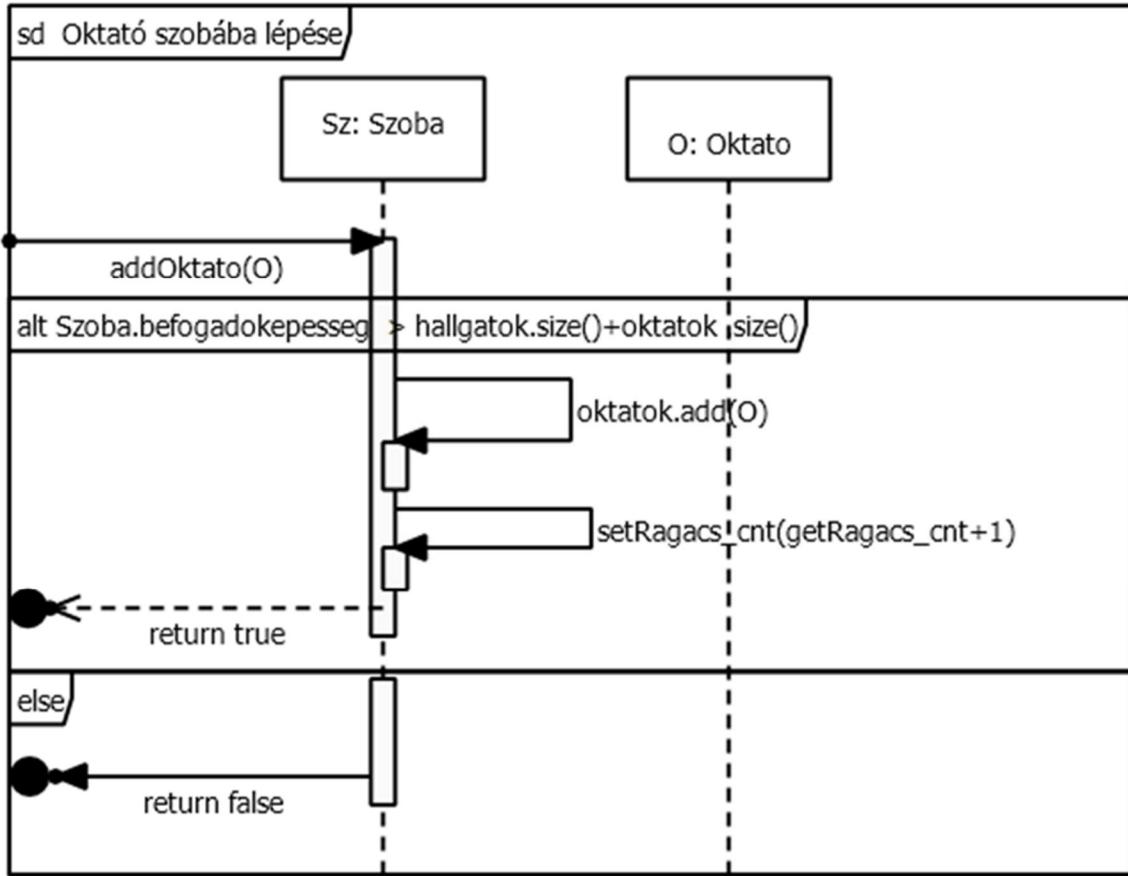
7.3.0.9 Hamis Logarléc felvétele



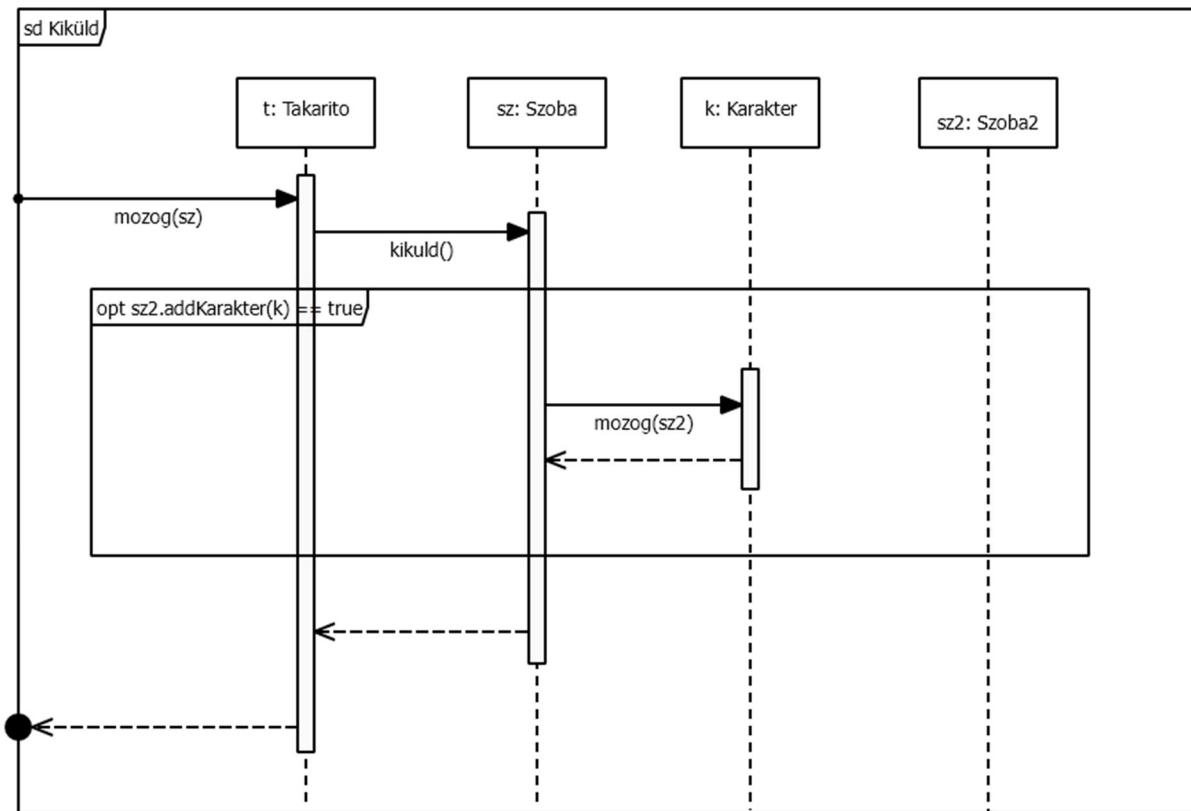
7.3.0.10 Add Hallgató



7.3.0.11 Add Oktató



7.3.0.12 Kiküld



A fenti példában a Karakter bármilyen játékbeli karakternek megfeleltethető, mivel mindegyikre úgyanúgy működik, illetve a kiinduló szobában tartózkodó összes karakterre értendő. Úgyanígy a Szoba2, a kiinduló szoba egy random szomszédját kívánja megjeleníteni, amelybe befér még az adott karakter. Ha nem fér be további szomszédot keresünk hasonló random módon

7.1 Prototípus interface-definíciója

7.1.1 Az interfész általános leírása

A prototípus, egy grafikus felülettel még nem rendelkező program, amely előre meghatározott parancsokkal vezérelhető. Működésének fő algoritmusai és a programot felépítő elemek már viszont többnyire ki vannak dolgozva, így minden lehetséges funkciója tesztelhető, végrehajtható. A prototípus vezérlésének alapját úgy alakítottuk ki, hogy képes legyen a program utasításokat szövegfájlok ból beolvasni és így megteremteni egy kiinduló helyzetet, amelyet később a console segítségével tovább módosíthatunk. Ezen kívül elmenthetjük a program jelenlegi állapotát a beolvasáshoz hasonlóan egy szöveges fájlba.

7.1.2 Bemeneti nyelv

load <fájl>

Leírás: A paraméterként megadott fájl tartalmát használva betölti és végrehajtja, az abban leírtakat.

Opciók: -

save <fájl>

Leírás: A paraméterként megadott fájlba kimenti a program jelenlegi állapotát.

Opciók: -

add_character <milyen karakter> <melyik szobába> <karakter id>

Leírás: Hozzáadunk a játékhoz egy új karaktert.

Opciók:

- <milyen karakter> hallgató hozzáadása esetén, ‘hallgato’, oktató hozzáadása esetén ‘oktato’, és takarító hozzáadása esetén ‘takarito’ az elvárt paraméter.
- <melyik szobába> A karakter melyik szobában kerül elhelyezésre
- <karakter id> Az elhelyezendő karaktert azonosító string, ami a karakter id attribútuma is lesz egyben.

move

Leírás: Karakter egyik szobából egy másikba mozog.

Opciók:

- <hallgato> <szoba> a paraméterben kapott hallgató a paraméterben kapott szobába mozog, ha tud.
- <oktato> <szoba> a paraméterben kapott oktató a paraméterben kapott szobába mozog, ha tud.
- <takarito> <szoba> a paraméterben kapott takarító a paraméterben kapott szobába mozog, ha tud.

spawn_item

Leírás: Hozzáadunk a játékhoz különböző tárgyakat.

Opciók:

- <targy> <szoba> <targy id> a paraméterben kapott tárgy típusból elhelyez egyet a paraméterül kapott szobában, a targy id értéke lesz a tárgy id attribútumának értéke

pick

Leírás: A játékos felvesz egy tárgyat.

Opciók:

- <hallgato> <targy> A paraméterben megadott hallgató felvesz egy paraméterben megadott tárgyat.

throw

Leírás: A játékos eldob egy tárgyat.

Opciók:

- <hallgato> <targy> A paraméterben megadott hallgató eldob egy paraméterben megadott tárgyat.

use

Leírás: A játékos használ egy tárgyat.

Opciók:

- <hallgato> <targy> A paraméterben megadott hallgató használja a paraméterben megadott tárgyat.

create room<milyen szoba>

Leírás: Hozzáadunk a játékhoz a paraméterben kapott szoba típusból egy új szobát.

Opciók:

- Lehet a szoba elátkozott: A paraméterben megadottnak megfelelően gázos, nem gázos, különböző befogadóképességű.
- Lehet a szoba nem elátkozott: A paraméterben megadottnak megfelelően gázos, nem gázos, különböző befogadóképességű.

toggle random

Leírás: Ki és be kapcsolja a véletlenszerűséget a programban.

Opciók:

- Ha be van kapcsolva, akkor ki kapcsolja, ha ki van kapcsolva, akkor bekapcsolja.

merge <szoba1> <szoba2>

Leírás: A két paraméterként kapott szoba egyesül

Opciók:**split <szoba>**

Leírás: A paraméterként kapott szoba osztódik, ha tud

Opciók:

- Ha a szoba, aminek osztóni kéne, nem egyesült egy másikkal, akkor nem történik semmi
- Ha már egyesült a szoba, akkor osztódik a két eredeti szobává

connect <szoba1><szoba2>

Leírás: A paraméterből kapott első szobából a második felé kerül egy ajtó

- **Opciók:** -

7.1.3 Kimeneti nyelv

load <fájl>

Leírás: A paraméterként megadott fájl tartalmát használva betölti és végrehajtja, az abban leírtakat.

Opciók:

- Fájl sikeresen betöltve
- Hiba történt a fájl betöltése közben

save <fájl>

Leírás: A paraméterként megadott fájlba kimenti a program jelenlegi állapotát.

Opciók:

- Sikeres mentés
- Sikertelen mentés

add_character <milyen karakter>

Leírás: Hozzáadunk a játékhoz egy új karaktert.

Opciók:

- Karakter sikeresen hozzáadva.
- Karakter hozzáadása sikertelen

move

Leírás: Karakter egyik szobából egy másikba mozog.

Opciók:

- Karakter mozgása sikeres.
- Karakter mozgása sikertelen.

spawn_item

Leírás: Hozzáadunk a játékhoz különböző tárgyakat.

Opciók:

- Tárgy hozzáadása sikeres.
- Tárgy hozzáadása sikertelen.

create_room<milyen szoba>

Leírás: Hozzáadunk a játékhoz különböző szobákat.

Opciók:

- Szoba hozzáadása sikeres.
- Szoba hozzáadása sikertelen.

toggle_random

Leírás: Ki és be kapcsolja a véletlenszerűséget a programban.

Opciók:

- Ki- vagy bekapcsolás sikeres.
- Ki- vagy bekapcsolás sikertelen.

merge <szoba1> <szoba2>

Leírás: A két paraméterként kapott szoba egyesül

Opciók:

- Egyesülés sikeres.
- Egyesülés sikertelen.

split <szoba>

Leírás: A paraméterként kapott szoba osztódik a két eredeti szobává

Opciók:

- Osztódás sikeres.

- Osztódás sikertelen.

7.2 Összes részletes use-case

Use-case neve	Load game
Rövid leírás	Egy adott fájlból betölt egy játék állást.
Aktorok	Controller
Forgatókönyv	1. Betöltődik a játék egy előre meghatározott helyzete.

Use-case neve	Save game
Rövid leírás	A játék jelenlegi helyzete elmentődik egy fájlba.
Aktorok	Controller
Forgatókönyv	1. A játék adott pillanatbeli helyzete mentésre kerül.

Use-case neve	Add character
Rövid leírás	Hozzáadunk a játékhöz egy új karaktert.
Aktorok	Controller
Forgatókönyv	1. A kiválasztott típusú karaktert hozzáadódik a játékhöz.

Use-case neve	Move hallgato
Rövid leírás	A hallgató egyik szobából egy másikba mozog.
Aktorok	Hallgató
Forgatókönyv	1. Ellenőrzés, hogy a hallgató befér-e a szobába. 2. Ha befér, bemehet, egyébként nem.

Use-case neve	Move hallgato söröspohár
Rövid leírás	A hallgató egyik szobából egy másikba mozog, ahol van oktató, de a hallgatonál van söröspohár, ami megvédi.
Aktorok	Hallgató
Forgatókönyv	1. Ellenőrzés, hogy a hallgató befér-e a szobába. 2. Ha befér, bemehet, egyébként nem. 3. Hallgatót megvédi a söröspohár. 4. Hallgató a söröspohár használata miatt eldob egy random itemet.

Use-case neve	Move hallgato tvsz
Rövid leírás	A hallgató egyik szobából egy másikba mozog, ahol van egy oktató, de a hallgatonál van tvsz, ami megvédi a hallgatót az oktatótól.
Aktorok	Hallgató
Forgatókönyv	1. Ellenőrzés, hogy a hallgató befér-e a szobába. 2. Ha befér, bemehet, egyébként nem. 3. Hallgatót megvédi a tvsz.

Use-case neve	Move hallgato maszk
----------------------	---------------------

Rövid leírás	A hallgató egyik szobából egy másikba mozog, ami gázos de a hallgatónál van maszk így nem veszti el az eszméletét.
Aktorok	Hallgató
Forgatókönyv	<ol style="list-style-type: none"> Ellenőrzés, hogy a hallgató befér-e a szobába. Ha befér, bemehet, egyébként nem. Hallgatót megvédi a maszk.

Use-case neve	Kibukas
Rövid leírás	A hallgató egyik szobából egy másikba mozog, ahol van egy oktató, és a hallgatónál nincs semmi olyan tárgy, ami megvédené az oktatótól, így kibukik.
Aktorok	Hallgató
Forgatókönyv	<ol style="list-style-type: none"> Ellenőrzés, hogy a hallgató befér-e a szobába. Ha befér, bemehet, egyébként nem. Hallgató kibukik.

Use-case neve	Hallgató eszméletet veszt
Rövid leírás	A hallgató egyik szobából egy másikba mozog, ami gázos, a hallgatót nem védi meg semmi a gáztól, így elveszti az eszméletét.
Aktorok	Hallgató
Forgatókönyv	<ol style="list-style-type: none"> Ellenőrzés, hogy a hallgató befér-e a szobába. Ha befér, bemehet, egyébként nem. Hallgató eszméletet veszt.

Use-case neve	Oktató eszméletet veszt
Rövid leírás	A oktató egyik szobából egy másikba mozog, ami gázos, az oktatót nem védi meg semmi a gáztól, így elveszti az eszméletét.
Aktorok	Oktató
Forgatókönyv	<ol style="list-style-type: none"> Ellenőrzés, hogy az oktató befér-e a szobába. Ha befér, bemehet, egyébként nem. Oktató eszméletet veszt.

Use-case neve	Move oktato
Rövid leírás	A oktató egyik szobából egy másikba mozog.
Aktorok	Oktató
Forgatókönyv	<ol style="list-style-type: none"> Ellenőrzés, hogy az oktató befér-e a szobába. Ha befér, bemehet, egyébként nem.

Use-case neve	Move takarito
Rövid leírás	A takarító egyik szobából egy másikba mozog.
Aktorok	Takarító
Forgatókönyv	<ol style="list-style-type: none"> Ellenőrzés, hogy a takarító befér-e a szobába. Ha befér, bemehet, egyébként nem.

Use-case neve	Spawn item
Rövid leírás	Hozzáadunk a játékhoz különböző tárgyat.

Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> Adott tárgy elhelyezése a szobában. A tárgy bekerül a szoba tárgyakról szóló nyilvántartásába.

Use-case neve	Pick
Rövid leírás	A hallgató felvesz egy tárgyat.
Aktorok	Hallgató
Forgatókönyv	<ol style="list-style-type: none"> A hallgató felveszi a tárgyat. Az a táskjába kerül. A tárgy törlődik a szobából.

Use-case neve	Throw
Rövid leírás	A hallgató eldob egy tárgyat.
Aktorok	Hallgató
Forgatókönyv	<ol style="list-style-type: none"> A hallgató eldobja a tárgyat. Az kikerül a táskjából. A tárgy megjelenik a szobában.

Use-case neve	Use
Rövid leírás	A hallgató használ egy tárgyat.
Aktorok	Hallgató
Forgatókönyv	<ol style="list-style-type: none"> A hallgató használja a tárgyat. A tárgy használódik.

Use-case neve	Create szoba
Rövid leírás	Hozzáadunk a játékhöz egy új szobát.
Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> A karakterek bemehetnek a szobákba. A szobák változhatnak.

Use-case neve	Toggle random
Rövid leírás	Ki és be kapcsolja a véletlenszerűséget a programban.
Aktorok	Controller
Forgatókönyv	Ha be van kapcsolva, akkor ki kapcsolja, ha ki van kapcsolva, akkor bekapcsolja.

Use-case neve	Merge
Rövid leírás	Egyesül két szoba
Aktorok	Controller
Forgatókönyv	<ol style="list-style-type: none"> Az egyesült szoba attribútumai megváltoznak az egyesült szoba szerint

Use-case neve	Split
Rövid leírás	Osztódik egy már egyesült szoba

Aktorok	Controller
Forgatókönyv	<p>1.</p> <ul style="list-style-type: none"> a. Ha a szoba, aminek osztódnia kéne nem egyesült egy másikkal, nem történik semmi b. Ha már egyesült a szoba, akkor osztódik a két eredeti szobává

7.3 Tesztelési terv

Teszt-eset neve	add_character_hallgato
Rövid leírás	A játékhoz hozzáadunk egy új hallgatót.
Teszt célja	Sikeresen hozzáadódik-e a játékhoz a hallgató.

Teszt-eset neve	add_character_oktato
Rövid leírás	A játékhoz hozzáadunk egy új oktatót.
Teszt célja	Sikeresen hozzáadódik-e a játékhoz a oktató.

Teszt-eset neve	add_character_takarito
Rövid leírás	A játékhoz hozzáadunk egy új takarítót.
Teszt célja	Sikeresen hozzáadódik-e a játékhoz a takarító.

Teszt-eset neve	spawn_item_sorospohar_szoba1
Rövid leírás	A söröspohár elhelyezése a szoba1-ben.
Teszt célja	Sikeresen elhelyeződik-e a söröspohár a szoba1-ben.

Teszt-eset neve	spawn_item_maszk_szoba1
Rövid leírás	A maszk elhelyezése a szoba1-ben.
Teszt célja	Sikeresen elhelyeződik-e a maszk a szoba1-ben.

Teszt-eset neve	spawn_item_logarlec_szoba1
Rövid leírás	A logarléc elhelyezése a szoba1-ben.
Teszt célja	Sikeresen elhelyeződik-e a logarléc a szoba1-ben.

Teszt-eset neve	spawn_item_tvsz_szoba1
Rövid leírás	A tvsz elhelyezése a szoba1-ben.
Teszt célja	Sikeresen elhelyeződik-e a tvsz a szoba1-ben.

Teszt-eset neve	spawn_item_legfrissito_szoba1
Rövid leírás	A légfrissítő elhelyezése a szoba1-ben.
Teszt célja	Sikeresen elhelyeződik-e a légfrissítő a szoba1-ben.

Teszt-eset neve	spawn_item_tranzisztor_szoba1
Rövid leírás	A tranzisztor elhelyezése a szoba1-ben.
Teszt célja	Sikeresen elhelyeződik-e a tranzisztor a szoba1-ben.

Teszt-eset neve	spawn_item_camembert_szoba1
------------------------	-----------------------------

Rövid leírás	A camembert elhelyezése a szoba1-ben.
Teszt célja	Sikeresen elhelyeződik-e a camembert a szoba1-ben.

Teszt-eset neve	spawn item rongy szoba1
Rövid leírás	A rongy elhelyezése a szoba1-ben.
Teszt célja	Sikeresen elhelyeződik-e a rongy a szoba1-ben.

Teszt-eset neve	move hallgato szoba1
Rövid leírás	A hallgató a szobájából a szoba1-be mozog, ha az adott szobába befér.
Teszt célja	Sikeresen átmegy-e a másik szobába, vagy marad a sajátjában, ha nem fér be.

Teszt-eset neve	move oktato szoba1
Rövid leírás	A oktató a szobájából a szoba1-be mozog, ha az adott szobába befér.
Teszt célja	Sikeresen átmegy-e a másik szobába, vagy marad a sajátjában, ha nem fér be.

Teszt-eset neve	move takarito szoba1
Rövid leírás	A takarító a szobájából a szoba1-be mozog, ha az adott szobába befér.
Teszt célja	Sikeresen átmegy-e a másik szobába, vagy marad a sajátjában, ha nem fér be.

Teszt-eset neve	pick hallgato1 tvsz
Rövid leírás	A hallgato1 felvesz egy tvszt.
Teszt célja	Sikeresen bekerül-e a táskába a tárgy

Teszt-eset neve	pick hallgato1 sorospohar
Rövid leírás	A hallgato1 felvesz egy söröspoharat.
Teszt célja	Sikeresen bekerül-e a táskába a tárgy

Teszt-eset neve	pick hallgato1 rongy
Rövid leírás	A hallgato1 felvesz egy rongyot.
Teszt célja	Sikeresen bekerül-e a táskába a tárgy

Teszt-eset neve	pick hallgato1 maszk
Rövid leírás	A hallgato1 felvesz egy maszkot.
Teszt célja	Sikeresen bekerül-e a táskába a tárgy

Teszt-eset neve	pick hallgato1 legfrissito
Rövid leírás	A hallgato1 felvesz egy légfrissítőt.
Teszt célja	Sikeresen bekerül-e a táskába a tárgy

Teszt-eset neve	pick hallgato1 tranzisztor
------------------------	----------------------------

Rövid leírás	A hallgato1 felvesz egy tranzisztor..
Teszt célja	Sikeresen bekerül-e a táskába a tárgy

Teszt-eset neve	pick hallgato1 camembert
Rövid leírás	A hallgato1 felvesz egy camembertet.
Teszt célja	Sikeresen bekerül-e a táskába a tárgy

Teszt-eset neve	throw hallgato1 tvsz
Rövid leírás	A hallgato1 eldob egy tvszt.
Teszt célja	Sikeresen kikerül-e a táskából a tárgy

Teszt-eset neve	throw hallgato1 sorospohar
Rövid leírás	A hallgato1 eldob egy söröspoharat.
Teszt célja	Sikeresen kikerül-e a táskából a tárgy

Teszt-eset neve	throw hallgato1 rongy
Rövid leírás	A hallgato1 eldob egy rongyot.
Teszt célja	Sikeresen kikerül-e a táskából a tárgy

Teszt-eset neve	throw hallgato1 maszk
Rövid leírás	A hallgato1 eldob egy maszkot.
Teszt célja	Sikeresen kikerül-e a táskából a tárgy

Teszt-eset neve	throw hallgato1 legfrissito
Rövid leírás	A hallgato1 eldob egy légfrissítőt.
Teszt célja	Sikeresen kikerül-e a táskából a tárgy

Teszt-eset neve	throw hallgato1 tranzisztor
Rövid leírás	A hallgato1 eldob egy tranzisztor..
Teszt célja	Sikeresen kikerül-e a táskából a tárgy

Teszt-eset neve	throw hallgato1 camembert
Rövid leírás	A hallgato1 eldob egy camembertet.
Teszt célja	Sikeresen kikerül-e a táskából a tárgy

Teszt-eset neve	use hallgato1 légfrissítő
Rövid leírás	A hallgato1 használ egy légfrissítőt
Teszt célja	Sikeresen kikerül-e a táskából a légfrissítő, és megjelenik a szobában, valamint elmúlik a gáz a szobából.

Teszt-eset neve	use hallgato1 tranzisztor
Rövid leírás	A hallgato1 teleportál egy tranzisztorral.
Teszt célja	Sikeresen átkerül a másik tranzisztor szobájába a hallgato1 és megváltozik a két tranzisztor szerepe.

Teszt-eset neve	use hallgato1 camembert
Rövid leírás	A hallgato1 használja a camembertet.

Teszt célja	Sikeresen kikerül-e a táskából a camembert és begázosítja a szobát.
--------------------	---

Teszt-eset neve	create szoba
Rövid leírás	Hozzáadunk a játékhöz egy új szobát.
Teszt célja	Sikeresen hozzáadódik-e a játékhöz a szoba.

Teszt-eset neve	create szoba gazos
Rövid leírás	Hozzáadunk a játékhöz egy új gázos szobát.
Teszt célja	Sikeresen hozzáadódik-e a játékhöz a gázos szoba.

Teszt-eset neve	create szoba elatkozott
Rövid leírás	Hozzáadunk a játékhöz egy új elátkozott szobát.
Teszt célja	Sikeresen hozzáadódik-e a játékhöz a elátkozott szoba.

Teszt-eset neve	create szoba elatkozott gazos
Rövid leírás	Hozzáadunk a játékhöz egy új gázos, elátkozott szobát.
Teszt célja	Sikeresen hozzáadódik-e a játékhöz a gázos, elátkozott szoba.

Teszt-eset neve	merge
Rövid leírás	Egyesít két szobát
Teszt célja	Sikeresen egyesül-e két normális szoba

Teszt-eset neve	merge gazos
Rövid leírás	Egyesít egy gázos és egy nem gázos szobát
Teszt célja	Gázos lesz-e az egyesült szoba

Teszt-eset neve	merge elatkozott
Rövid leírás	Egyesít egy elátkozott és egy nem elátkozott szobát
Teszt célja	Elátkozott lesz-e az egyesült szoba

Teszt-eset neve	merge elatkozott gazos
Rövid leírás	Egyesít egy elátkozott és egy gázos szobát
Teszt célja	Elátkozott és gázos lesz-e az egyesült szoba

Teszt-eset neve	split
Rövid leírás	Osztódik egy szoba két szobává
Teszt célja	Sikeresen osztódik-e a két szoba -> befogadóképesség, tárgyak, szomszédok jól változtak

Teszt-eset neve	split gazos
Rövid leírás	Osztódik egy szoba egy gázos és egy nem gázos szobává
Teszt célja	Sikeresen osztódik-e a két szoba -> gázosság jól változik

Teszt-eset neve	split elatkozott
Rövid leírás	Osztódik egy szoba egy elátkozott és egy nem elátkozott szobává
Teszt célja	Sikeresen osztódik-e a két szoba ->elátkozottság jól változik

Teszt-eset neve	split elatkozott gazos
Rövid leírás	Osztódik egy szoba egy elátkozott és egy gázos szobává
Teszt célja	Sikeresen osztódik-e a két szoba ->elátkozottság, gázosság jól változik

Teszt-eset neve	no split
Rövid leírás	Megpróbál osztódni egy nem egyesült szoba
Teszt célja	Sikertelen-e az osztódás

7.4 Tesztelést támogató segéd- és fordítóprogramok specifikálása

A prototípus tesztelése kétféleképpen tehető meg. Vagy egy fejlesztőkörnyezetben (például Visual Studio Code) futtatva, vagy parancssori alkalmazásként végezhető el. Az előre megírt tesztelési feltételek a program indulásakor betöltődnek, innen vagy egy fájl beolvasásával, vagy a segédletben leírt parancsokkal, előre megírt teszteket lehet futtatni.

7.5 Napló

Kezdet	Időtartam	Résznevők	Leírás
2024.04.03. 21:30	1 óra	Csapat	Értekezlet. Döntés: Heti teendők felosztása, megbeszélése.
2024.04.05 9:00	1 óra	Hermann	Use-case esetek megírása.
2024.04.05. 11:00	1 óra	Csapat	Értekezlet. Döntés: Tesztek felosztása.
2024.04.06. 9:30	2 óra	Czotter	add_character, move, spawn_item parancsokhoz tartozó dolgok megírása. 0.3.1 és 0.3.2 diagrammok elkészítése, hozzájáruló metódusok leírása.
2024.04.07.	1 óra	Hermann	Tesztesetek írása
2024.04.07		Csapat	Értekezlet.

			Döntés: Heti teendők felosztása, megbeszélése.
2024.04.07. 22:00	2 óra	Németh	load és save parancsok, .2.1 bekezdés, 0.3.3, 7.3.7, 7.3.8 és 7.3.9 diagrammok
2024.04.07 19:00	2 óra	Tarsoly	Tesztesetek, módostások
2024.04.07.19:00	1,5 óra	Schulcz	Tesztesetek, módostások
2024.04.08 12:00	1 óra	Hermann, Tarsoly	Végső ellenőrzés, formázás

Részletes tervezet

35 – pupakok

Konzulens:
Koczó Attila

Csapattagok:

Tarsoly Levente	E1IK75	tarsolyl@edu.bme.hu
Németh Gergely	TA9G0R	gergo.h.nemeth@gmail.com
Schulcz Gábor	EP3I9R	schulcz.gabor6@gmail.com
Czotter Benedek	TFB4FY	czottibeni@gmail.com
Hermann Máté Béla	T83K5I	mate3829@gmail.com

2024.04.15

8. Részletes tervezés

8.1 Osztályok és metódusok tervezése.

8.1.1 Szoba

- **Felelősség:**

A játékot felépítő szobák megvalósításáért felelős ez az osztály. A ki-be lépő karaktereket, tárgyak elhelyezését és eltüntetését, illetve a szobák osztódását és összeolvadását valósítja meg.

- **Ősosztályok:**

-

- **Interfészek:**

-

- **Attribútumok:**

- # hallgatok: ArrayList<Hallgato>
Lista a szobában aktuálisan tartózkodó hallgatók számára.
- # oktatok: ArrayList<Oktato>
Lista a szobában aktuálisan tartózkodó oktatók számára.
- # takaritok: ArrayList<Takarito>
Lista a szobában aktuálisan tartózkodó takarítók számára.
- # ragacs_cnt: int
Számláló a szoba ragacsosságának nyilvántartására.
- # id: string
Minden szobához tartozik egy egyedi azonosító, hogy könnyen meg lehessen különböztetni két szobát egymástól.
- # palya: Palya
A szobákat tároló pályához férhetünk hozzá, ezen az attribútumon keresztül.
- #gazos : bool
A szoba gázosságát tartja számon.
- #befogadokepesseg: int
A szobába beférő karakterek számát tárolja.
- #regiszobak: ArrayList<Szoba>
Ha egyesült, az eredeti szobák adatait tartja nyilván.
- #szomszedok: ArrayList<Szoba>
A szoba szomszédjait tárolja.
- #targyak: ArrayList<Targy>
A szobában levő tárgyakat tartalmazza.

- **Metódusok:**

- + getHallgatok(): ArrayList<Hallgato>
Visszadaja a hallgatók attribútum értékét.
- + setHallgatok(ArrayList<Hallgato>): void
A paraméterként kapott listát állítja be a hallgatók attribútum értékének.
- + getOktatok(): ArrayList<Oktato>

- Visszadaja az oktatók attribútum értékét.
- + setOktatok(ArrayList<Oktato>): void
A paraméterként kapott listát állítja be az oktatók attribútum értékének.
- + getTakaritok(): ArrayList<Takarito>
Visszaadja a takarítók attribútum értékét.
- + setTakaritok(ArrayList<Takarito>): void
A paraméterként kapott listát állítja be a takarítók attribútum értékének.
- + getRagacs_cnt(): int
Visszaadja a ragacs_cnt attribútum értékét.
- + setRagacs_cnt(int): void
A paraméterként kapott számot állítja be ragacs_cnt attribútum értékének.
- + getid(): string
Visszaadja az id attribútum értékét.
- + setid(string): void
A paraméterként kapott stringet állítja be az id attribútum értékének.
- + removeHallgato(Hallgato): void
Abban az esetben, ha egy hallgató kilép a szobából, akkor hívódik meg. Ebben a metódusban az egyetlen utasítás az, hogy a hallgatók attribútumból eltávolítsuk a szobát elhagyó hallgatót.
- + removeOktato(Oktato): void
Abban az esetben, ha egy oktató kilép a szobából, akkor hívódik meg. Ebben a metódusban az egyetlen utasítás az, hogy az oktatók attribútumból eltávolítsuk a szobát elhagyó oktatót.
- + removeTakarito(Takarito): void
Abban az esetben, ha egy takarító kilép a szobából, akkor hívódik meg. Ebben a metódusban az egyetlen utasítás az, hogy a takarítók attribútumból eltávolítsuk a szobát elhagyó takarítót.
- + addHallgato(Hallgato): bool
A metódus feladata lekezelni egy hallgató szobába lépésének kísérletét.
Rövid pszeudokód:
Ha a hallgató befér a szobába:
 hallgatók attribútumhoz hozzáadjuk az adott hallgatót
 visszatérési érték igaz
egyébként:
 visszatérési érték hamis
- + addOktato(Oktato): bool
A metódus feladata lekezelni egy oktató szobába lépésének kísérletét.
Rövid pszeudokód:
Ha az oktató befér a szobába:
 oktatók attribútumhoz hozzáadjuk az adott oktatót
 visszatérési érték igaz
egyébként:
 visszatérési érték hamis
- + addTakarito(Takarito): bool
A metódus feladata lekezelni egy takarító szobába lépésének kísérletét.
Rövid pszeudokód:
Ha a takarító befér a szobába:
 takarítók attribútumhoz hozzáadjuk az adott takarítót
 visszatérési érték igaz
egyébként:

visszatérési érték hamis

- + deleteHallgato(Hallgato): void

A metódus feladata az, hogy abban az esetben, ha kibukik a hallgató a szobában, ne csak a szoba hallgatói nyilvántartásából, hanem az pálya hallgatókról vezetett listájából is törlődjön.

Rövid pszeudokód:

removeHallgato(kibukott hallgató) függvény meghívása

Kibukott hallgató eltávolítása a pálya hallgatókról vezetett listájából.

Annak ellenőrzése, hogy maradt-e még hallgató a pályán:

ha nem maradt, pályának jelezni, hogy fejezze be a játéket.

- -befer(): bool

A függvény lényege, hogy megvizsgálja, befér-e a karakter a szobába

Rövid pszeudokód:

Ha befogadokepesseg - (hallgatók lista mérete + oktatok lista mérete + takaritok lista mérete) >= 1

visszatérési érték hamis

Egyébként

visszatérési érték igaz

- +isGazos(): bool

Visszaadja, hogy egy szoba gázos-e

- +setGaz(bool): void

A paraméterként kapott érték lesz a szoba gázossága

- +getBefogadokepesseg(): int

Visszaadja a szoba befogadóképességét

- +setBefogadokepesseg(int): void

A paraméterként kapott érték lesz a szoba befogadóképessége

- #getRegiszrobak(): ArrayList<Szoba>

Visszaadja a szoba egyesülése előtti szobák állapotát, ha nem egyesült, akkor egy üres listát

- #setRegiszrobak(ArrayList<Szoba>): void

A paraméterként kapott lista lesz a szoba egyesülés előtti szobák állapota

- +getSzomszedok(): ArrayList<Szoba>

Visszaadja a szoba szomszéd szobáit

- +setSzomszedok(ArrayList<Szoba>): void

A paraméterként kapott lista lesz a szoba szomszédjainak listája

- +addSzomszed(Szoba): void

A paraméterként kapott érték hozzáadódik a szoba szomszédjainak listájához

- +removeSzomszed(Szoba): void

A paraméterként kapott érték törlődik a szoba szomszédjainak listájából

- +getTargyak(): ArrayList<Targy>

Visszaadja a szobában található tárgyak listáját

- +setTargyak(ArrayList<Targy>): void

A paraméterként kapott lista lesz a szoba tárgyainak listája

- +targy_elhelyezese(Targy) : void

A paraméterként kapott tárgyat lehelyezi a szobába

Rövid pszeudokód:

tárgy szoba attribútuma ez a szoba lesz

szoba tárgyainak listájába adódik a tárgy

- +targy_eltuntetese(Targy) : void

A paraméterként kapott tárgy kikerül a szoba tárgyainak listájából

- +osztodik () : Szoba

A szoba szétesztódik a két eredeti szobájává

Rövid pszeudokód:

Ha a regiszerek lista nem üres

Létrejön szoba1 és szoba2, a regiszerek listában levőszobák tulajdonságaival

Az eredeti szobából kikerülnek a tárgyak és átkerülnek az eredeti szobába azoknak a regiszerek tárgyak listái szerint

Az eredeti szoba törlődik a szomszédjai listájából, és bekerülnek az új szobák

Az eredeti szoba törlődik a pályáról

Az új szobák a pályához adódnak

- +egyesul (Szoba) : void

A paraméterként kapott szobával egyesül a szobával

Rövid pszeudokód:

Létrejön az új egyesült szoba

új szoba regiszerek listájába adódik a paraméterből kapott szoba és ez a szoba

Ha valamelyik régi szoba gázos, vagy elátkozott volt, ez isazzá válik

Minden tárgyat elhelyez az új szobába a targy_elhelyezese függvényel

A régi szobák minden szomszédját az új szobának a szomszédjai közé veszi, és a régi szobákat törli a szomszédok szomszédjai közül, és az új szobát beleteszi a szomszédjaik listájába

A régi szobákat kiveszi a pályáról

Az új szobát pedig felveszi a pályára

- +isRagacsos() : bool

Visszaadja, hogy a szoba ragacsos-e

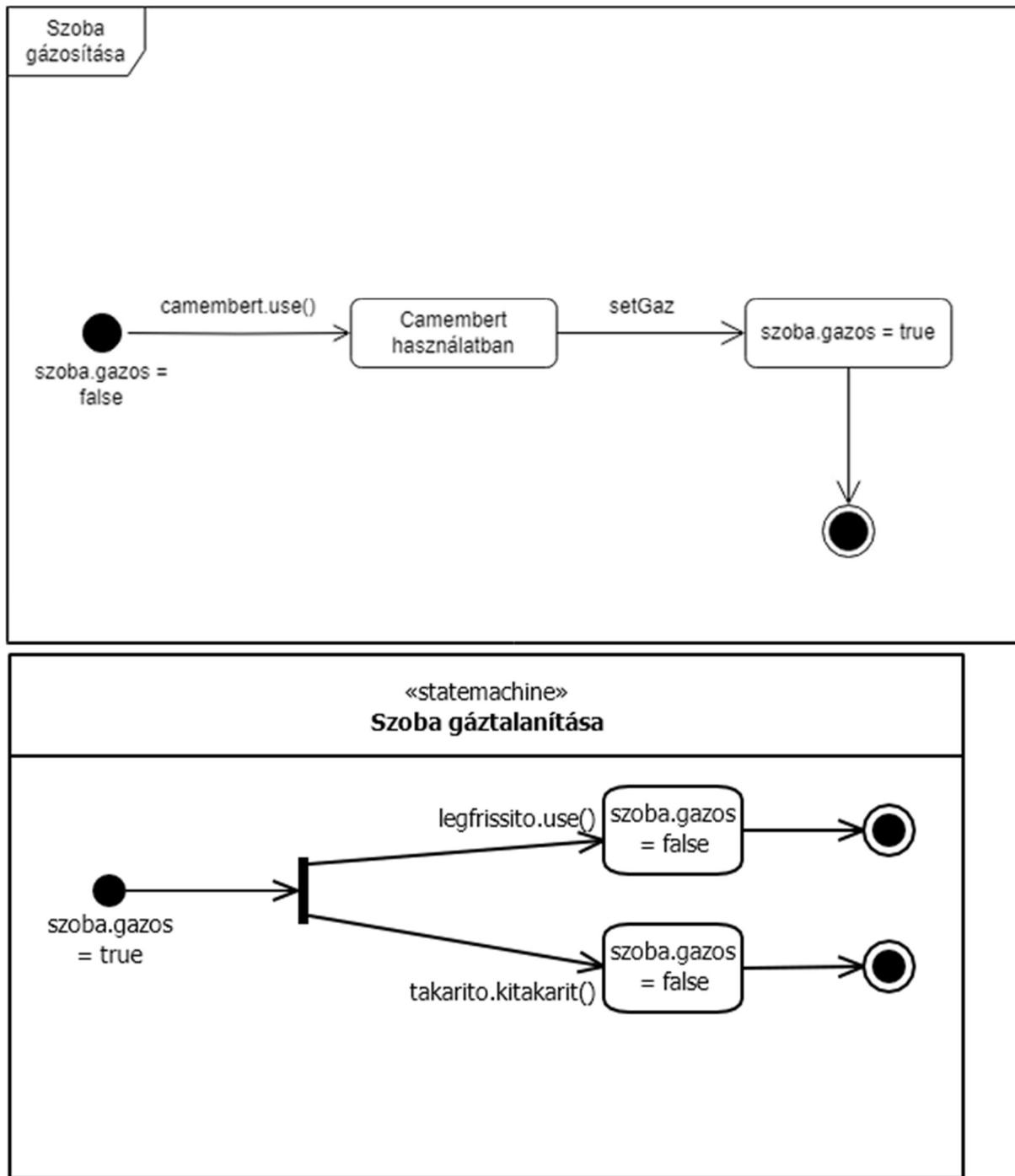
Rövid pszeudokód:

Ha a ragacsnt \geq mint a megadott ragacsossághoz kellő szám

visszatérési érték igaz

Egyébként

visszatérési érték hamis



8.1.2 Elátkozott_Szoba

- **Felelősség:**

Az elátkozott szobák megvalósításáért felelős ez az osztály. A szoba ajtajainak eltünéséért és előtünéséért felelős.

- **Ősosztályok:**

Szoba**• Interfészek:**

-

• Attribútumok:

- - eltuntajto: ArrayList<Szoba>
Azon szobákat tartalmazza ez a lista, amelyekhez vezető ajtók eltűntek.

• Metódusok:

- # getEltuntajto(): ArrayList<Szoba>
Visszaadja az eltuntajto attribútum értékét.
- # setEltuntajto(ArrayList<Szoba>): void
A paraméterként kapott listát állítja be az eltuntajto paraméteréül.
- # addEltuntajto(Szoba): void
Hozzáadja a paraméterként kapott szobát az eltuntajto listához.
- # removeEltuntajto(Szoba): void
Eltávolítja az eltuntajto listából a paraméterként kapott szobát.
- + elunik(Szoba): void
Az elátkozott szobának eltűnik a paraméterként kapott ajtaja
Rövid pszeudokód:
eltuntajto listába kerül a paraméterként kapott Szoba
A paraméterként kapott Szoba szomszédjai közül törlődik ez a szoba
Ennek a szomszédjai közül törlődik a paraméterként kapott Szoba
- + elotunik(Szoba): void
Az elátkozott szobának előtűnik egy paraméterként kapott eltűnt ajtaja
Rövid pszeudokód:
Ennek a szomszédjai közé adódik a paraméterként kapott Szoba
A paraméterként kapott Szoba szomszédjai közé adódik ez a szoba
eltuntajto listából törlődik a paraméterként kapott Szoba

8.1.3 Karakter**• Felelősség:**

A játékban szereplő karakterekért felelős ez az osztály. A karakterek mozgását, tárgy felvevését és eszméletvesztését és a számontartja a karakter tárgyait, eszméletvesztettségét, azonosítóját.

• Ősosztályok:

-

• Interfészek:

-

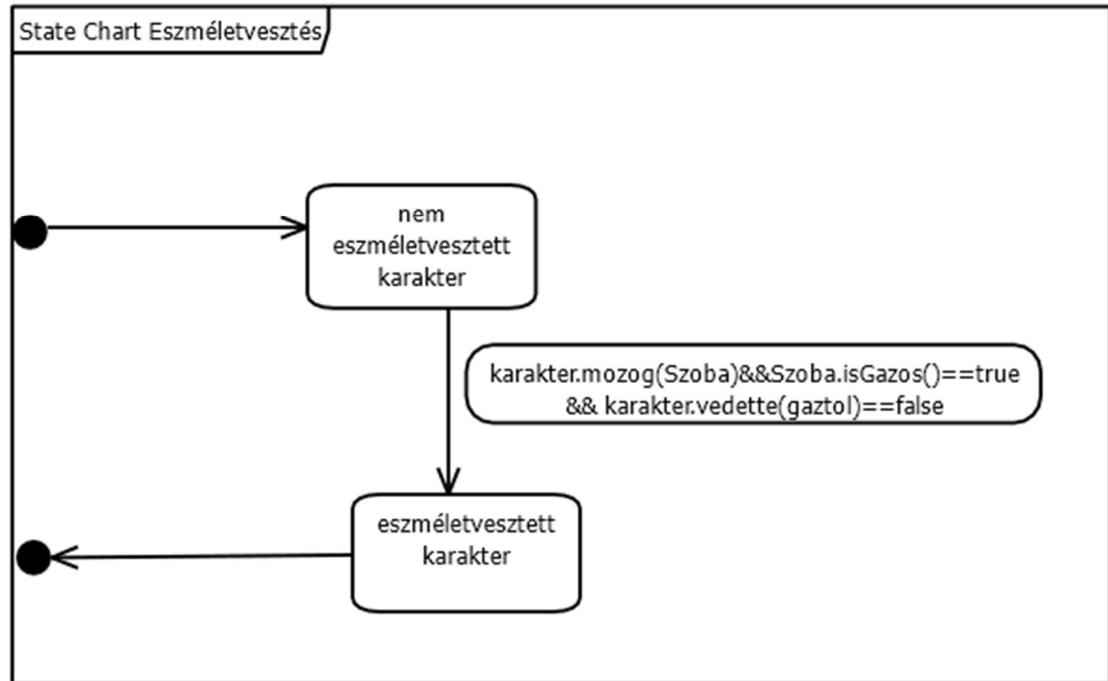
• Attribútumok:

- # eszmeletvesztett : bool

- Számítja, hogy a karakter eszméletvesztett-e.
- # szoba : Szoba
Számítja, hogy a karakter, melyik szobába van éppen.
- # id : string
A karakter azonosítója, hogy meglehessen különbözetni a többi karaktertől.
- # taska: ArrayList<Targy>
Lista a karakternél aktuálisan lévő tárgyak számára.

● **Metódusok:**

- + getEszmeletvesztett(): bool
Visszaadja, hogy a karakter eszméletvesztett-e.
- + setEszmeletvesztett(bool): bool
Beállítja a paraméternek megfelelően az eszméletvesztett értékét, hogy ha igaz akkor a karakter eszméletvesztett, ha nem nem.
- + getSzoba(): Szoba
Visszaadja, hogy a karakter melyik szobába van.
- + setSzoba(Szoba): void
Beállítja, hogy a karakter az attributumba megadott szobába van.
- + mozog (Szoba) : void
Az adott szobába helyezi a karaktert, ha befér és kiveszi az előző szobából, ahol volt, ez egy absztrakt függvény.
- + felvesz (Targy): void
A karakter az adott tárgyat felveszi a szobából és hozzáadja a táska listájához, ha lehetséges, ez egy absztrakt függvény.
- + eszméletvesztés(): void
Az eszméletetségét a karakternek igazra állítja és az összes tárgyat, amit a karakter birtokol, elveszi a karaktertől és a szobába teszi.
Rövid pszeudokód:
meghívja setEszmeletvesztett(bool) függvényt igaz paraméterrel, majd végigmegy a táska listán és a tárgyakra meghívja setBirtokos(Karakter) függvény és null értéket ad át neki, majd meghívja a getSzoba() szobán a targy_elhelyezese(Targy) függvényt a tárgyal, majd törli a táska listájából.
- + vedette(Vedettseg) : bool
Meglézi, hogy a karakter védett-e a paraméterben lévő dolgotól, azzal, hogy meglézi a karakter tárgyait és ha kell használja és visszaadja, hogy védett.
Rövid pszeudokód:
Ha az attribútum Vedettseg értéke, az oktatot van, akkor végig megy a karakter taska listáján és ha van olyan tárgya, aminek a funkció értéke oktatot támad vagy oktatot védi, akkor meghívja a tárgy use metódusát és visszatér igaz értékkel, ha nincs nála ilyen tárgy és azaz végigér a taska listán akkor visszatér hamis értékkel.
- + getTaska() : ArrayList<Targy>
Visszaadja a táska listát.



8.1.4 Oktato

- **Felelősség:**

A játékban szereplő oktatókért felelős ez az osztály. Az oktatók megbénulásának kezeléséét valósítja meg.

- **Ősosztályok:**

Karakter

- **Interfészek:**

-

- **Attribútumok:**

- -megbenult: bool
Számoltartja, hogy a karakter megbénult-e.

- **Metódusok:**

- + megbenul(): void
Az megbénult az oktatóknak igazra állítja.
- + mozog(Szoba) : void
Az adott szobába helyezi az oktatót, ha befér és kiveszi az előző szobából, ahol volt és a táskjában lévő tárgyakat is frissít.

Rövid pszeudokód:

meghívja az attributumba megadott szobán az addOktato(Oktato) függvényt, paraméterben megadva magát.

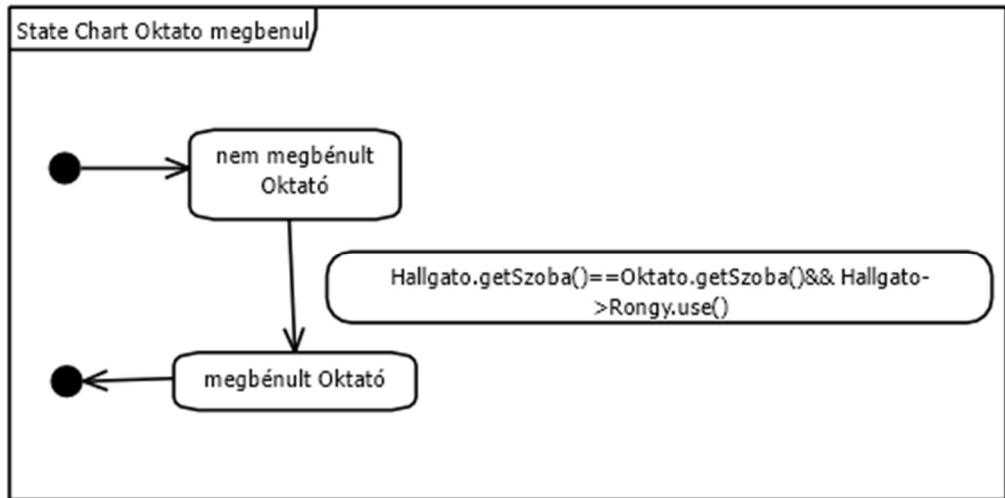
Ha az addOktato(Oktato) függvény visszatérési értéke igaz, akkor meghívja az eddigi getSzoba()-ra a removeOktato(Oktato) függvényt majd beállítja setSzoba(Szoba) a szobát és végigmegy a táska listáját és meghívja az összes tárgyára a setSzoba(Szoba) függvényt a attributumba az attribútumba kapott szobával.

Ha az addOktato(Oktato) függvény visszatérési értéke hamis, akkor visszatér.

- + felvesz (Tárgy): void

Az oktató az adott tárgyat felveszi a szobából és hozzáadja a táska listájához.
Rövid pszeudokód:

Az oktató hozzáadja a táskjához a tárgyat, setBirtkos(Karakter) függvényt meghívja a tárgyon ahol magát adja paraméterben, majd meghívja a szobán a targy_eltuntetese(Targy) függvényt a tárgyal.



8.1.5 Hallgato

- **Felelősség:**

A játékban szereplő hallgatókért felelős ez az osztály. A hallgatók kibukását, tárgy eldobását, teleportálását valósítja meg.

- **Ősosztályok:**

Karakter

- **Interfészek:**

-

- **Attribútumok:**

-

- **Metódusok:**

- + eldob (Tárgy): void

Attribútumba kapott tárgyat elhelyezi abba a szobába, ahol tartózkodik a hallgató és kiveszi a hallgatónak a tárgyai közül.

Rövid pszeudokód:

A setBirtkos(Karakter) függvényt meghívja a tárgyon ahol nullt ad paraméterben, majd meghívja a szobán a targy_elhelyezese(Targy) függvényt a tárgyal, majd törli a tárgyat a táska listájából.

- + kibukik(): void

Kiveszi a hallgatót a szobák hallgatók listájából, ezzel véget ér a hallgatónak a játék.

Rövid pszeudokód:

A getSzoba() szobán meghívja a deleteHallgato(Hallgato) függvényt magával és végigmegy a táska listán és meghívja rajtuk a

setSzoba(Szoba) és a setBirtokos(Karakter) függvényeket null paraméterrel.

- + teleport(Tranzisztor) : void

Az attribútumba kapott tranzisztort használja a hallgató.

Rövid pszeudokód:

Meghívja a kapott Tranzisztoron a use() függvényt.

- + mozog(Szoba) : void

Az adott szobába helyezi a hallgatot, ha befér és kiveszi az előző szobából, ahol volt és a táskjában lévő tárgyakat is frissíti.

Rövid pszeudokód:

meghívja az attributumba megadott szobán az addHallgato(Hallgato) függvényt, paraméterben megadva magát.

Ha az addHallgato(Hallgato) függvény visszatérési értéke igaz, akkor meghívja az eddigi getSzoba()-ra a removeHallgato(Hallgato) függvényt majd beállítja setSzoba(Szoba) a szobát és végigmegy a táska listáját és meghívja az összes tárgyára a setSzoba(Szoba) függvényt a attributumba az attribútumba kapott szobával.

Ha az addHallgato(Hallgato) függvény visszatérési értéke hamis, akkor visszatér.

- + felvesz (Tárgy): void

A hallgato az adott tárgyat felveszi a szobából és hozzáadja a táska listájához, ha lehetséges.

Rövid pszeudokód:

Ha a táska lista mérete 5, akkor visszatér, ha nem akkor a hallgató hozzáadja a táskjához a tárgyat, setBirtokos(Karakter) függvényt meghívja a tárgyon ahol magát adja paraméterben, majd meghívja a szobán a targy_eltuntetese(Targy) függvényt a tárgyal.

8.1.6 Takarito

- **Felelősség:**

A játékban szereplő takarítók felelős ez az osztály. A takarítók takarítását és a többi karakter szobából kiküldését valósítja meg.

- **Ősosztályok:**

Karakter

- **Interfészek:**

-

- **Attribútumok:**

-

- **Metódusok:**

- + kikuld() : void

A többi szobába tartozkodó karaktert áthelyezi egy másik szobába.

Rövid pszeudokód:

Addig megy a getSzoba().getSzomszedok() listán, amíg nem üres a getSzoba() szoba getHallgatok és a getOktatok listája és minden szobára Végigmegy a getSzoba() szoba a getHallgatok majd a getOktatok listáján és mindegyik karakterre meghívja a mozog függvényt az adott szomszéd szobára. Majd ha már üres lesz a Szoba getHallgatok és a getOktatok listája vagy végigér a getSzoba().getSzomszedok() listán, akkor visszatér.

- + takarit() : void

A szoba ragacs számlálóját nullára állítja.

- + mozog(Szoba) : void

Az adott szobába helyezi a takarítót, ha befér és kiveszi az előző szobából, ahol volt és a táskjában lévő tárgyakat is frissíti.

Rövid pszeudokód:

meghívja az attributumba megadott szobán az addTakarito(Takarito) függvényt, paraméterben megadva magát.

Ha az addTakarito(Takarito) függvény visszatérési értéke igaz, akkor meghívja az eddigi getSzoba()-ra a removeTakarito(Takarito) függvényt majd beállítja setSzoba(Szoba) a szobát és végigmegy a táska listáját és meghívja az összes tárgyára a setSzoba(Szoba) függvényt a attributumba az attribútumba kapott szobával.

Ha az addTakarito(Takarito) függvény visszatérési értéke hamis, akkor visszatér.

- + felvesz (Tárgy): void

A takarító az adott tárgyat felveszi a szobából és hozzáadja a táska listájához, ha lehetséges.

A takarító hozzáadja a táskjához a tárgyat, setBirtkos(Karakter) függvényt meghívja a tárgyon ahol magát adja paraméterben, majd meghívja a szobán a targy_eltuntetese(Targy) függvényt a tárgyal.

8.1.7 Tárgy

- **Felelősség:**

Az osztály a játékban megtalálható tárgyak megvalósításáért felel. Kezeli továbbá minden tárgy sajátos funkcióját a használatuk közben.

- **Ősosztályok:**

-

- **Interfészek:**

-

- **Attribútumok:**

- # funkcio: Funkcio

Az adott tárgy funkciójának megfelelő enumerációs attribútum. Egyedül azt dönti el pontosan, hogy milyen tárgyról is beszélünk.

- # id: string

A tárgyakat egyértelműen meghatározó azonosító.

- **Metódusok:**

- + setSzoba(Szoba): void

Beállítja az adott tárgyhoz tartozó szobát, amit paraméterként kap és ahol ezek után a tárgy megtalálható lesz.

- + setBirtkos(Karakter): void

Beállítja az adott tárgyhoz tartozó gazda karaktert, amit paraméterként kap és akinél a tárgy megtalálható lesz, amíg valamelyen módon meg nem válik tőle.

- + use(): void

Absztrakt függvény, amely a specifikus tárgyak esetében, azok használatáért felelős.

- + getId(): String

Getter függvény az Id attribútumra.

- + getFunkcio(): Funkcio

Getter függvény az Funkcio attribútumra.

- + setId(String): void

Setter függvény az Id attribútumra.

8.1.8 Funkció enumeráció

- **Felelősség:**
Egy adott tárgy funkcióját egyértelműen meghatározó felsorolás.
- **Ősosztályok:**
 -
- **Interfészek:**
 -
- **Attribútumok:**
 - oktatotol_ved
A tárgy képes megvédeni a játékosot egy oktató támadásai ellen.
 - gaztol_ved
A tárgy képes megvédeni a játékosot egy gázos szoba hatása ellen.
 - oktatot_tamad
A tárgy képes a játékos birtokában támadó hatással lenni oktatók ellen.
 - aktiv
A tárgyat aktívan képes használni egy játékos, tehát meg lehet választani a tárgy felhasználásának körülményeit.
 - hamis
A tárgy nem használható semmire, csupán magához venni tudja a játékos.
 - logarlec
A tárgy megszerzését követően a játékosok győzelmével véget ér a játék.

8.1.9 Passzív Tárgy

- **Felelősség:**
Az osztály a játékban megtalálható passzív tárgyak megvalósításáért felel. A passzív tárgyak felhasználását nem a játékos vezérli, hanem valami specifikus eseményhez vannak kötve, vagy egy adott időn belül használódnak fel.
- **Ősosztályok:**
Tárgy
- **Interfészek:**
 -
- **Attribútumok:**
 - # töltet: int
Egyes tárgyak felhasználásának száma egy bizonyos számhoz kötött. A töltet ennek a felhasználási számnak a felügyeletéért felel.
- **Metódusok:**

- + setToltet(): void
Beállítja, hogy az adott tárgy még mennyi alkalommal használható mielőtt nem tudja többet használni a játékos.
- + getToltet(): void
Getter függvény a Toltet attribútumra.

8.1.10 Aktív Tárgy

- **Felelősség:**

Az osztály a játékban megtalálható aktív tárgyak megvalósításáért felel. Az aktív tárgyak felhasználását a játékos közvetlenül vezérelheti, eldöntheti, hogy pontosan mikor és hogyan kívánja használni azokat.

- **Ősosztályok:**

Tárgy

- **Interfészek:**

-

- **Attribútumok:**

-

- **Metódusok:**

- + use(): void

Absztrakt függvény, amely a specifikus tárgyak esetében, azok használatáért felelős.

8.1.11 Camembert

- **Felelősség:**

Az osztály a játékban megtalálható Camembert tárgy megvalósításáért felel. A tárgy használatakor a szoba, ahol a hallgató tartózkodik, gázzal töltött állapotba kerül.

- **Ősosztályok:**

Aktív Tárgy

- **Interfészek:**

-

- **Attribútumok:**

-

- **Metódusok:**

- + use(): void

A függvény meghívása után a szoba, ahol a játékos tartózkodik, gázzal töltődik meg.

8.1.12 Légfrissítő

- **Felelősség:**

Az osztály a játékban megtalálható Légfrissítő tárgy megvalósításáért felel. A tárgy használatkor, gázos szobában lerakva semlegesíti a gázhatást.

- **Ősosztályok:**

Aktív Tárgy

- **Interfészek:**

-

- **Attribútumok:**

-

- **Metódusok:**

- + use(): void

A függvény meghívása után a gázos szobából, ahol a játékos tartózkodik, eltűnik a gáz.

8.1.13 Tranzisztor

- **Felelősség:**

Az osztály a játékban megtalálható Tranzisztor tárgy megvalósításáért felel. A tárgy lehetővé teszi a játékos számára, hogy a pályán két (előre meghatározott) helyszín között teleportálhasson. Mindezt két tranzisztor párosításával, majd elhelyezésekkel valósíthatja meg a felhasználó.

- **Ősosztályok:**

Aktív Tárgy

- **Interfészek:**

-

- **Attribútumok:**

- – aktív: bool

Azt határozza meg, hogy a tranzisztor aktiválva van-e, vagy sem.

- – hova: Szoba

Megadja, hogy a tranzisztor, teleportálást követően melyik szobában fogja elhelyezni a hallgatót.

- – társ: Tranzisztor

Megadja, hogy az adott tranzisztor melyik másikkal van összekapcsolva a pályán.

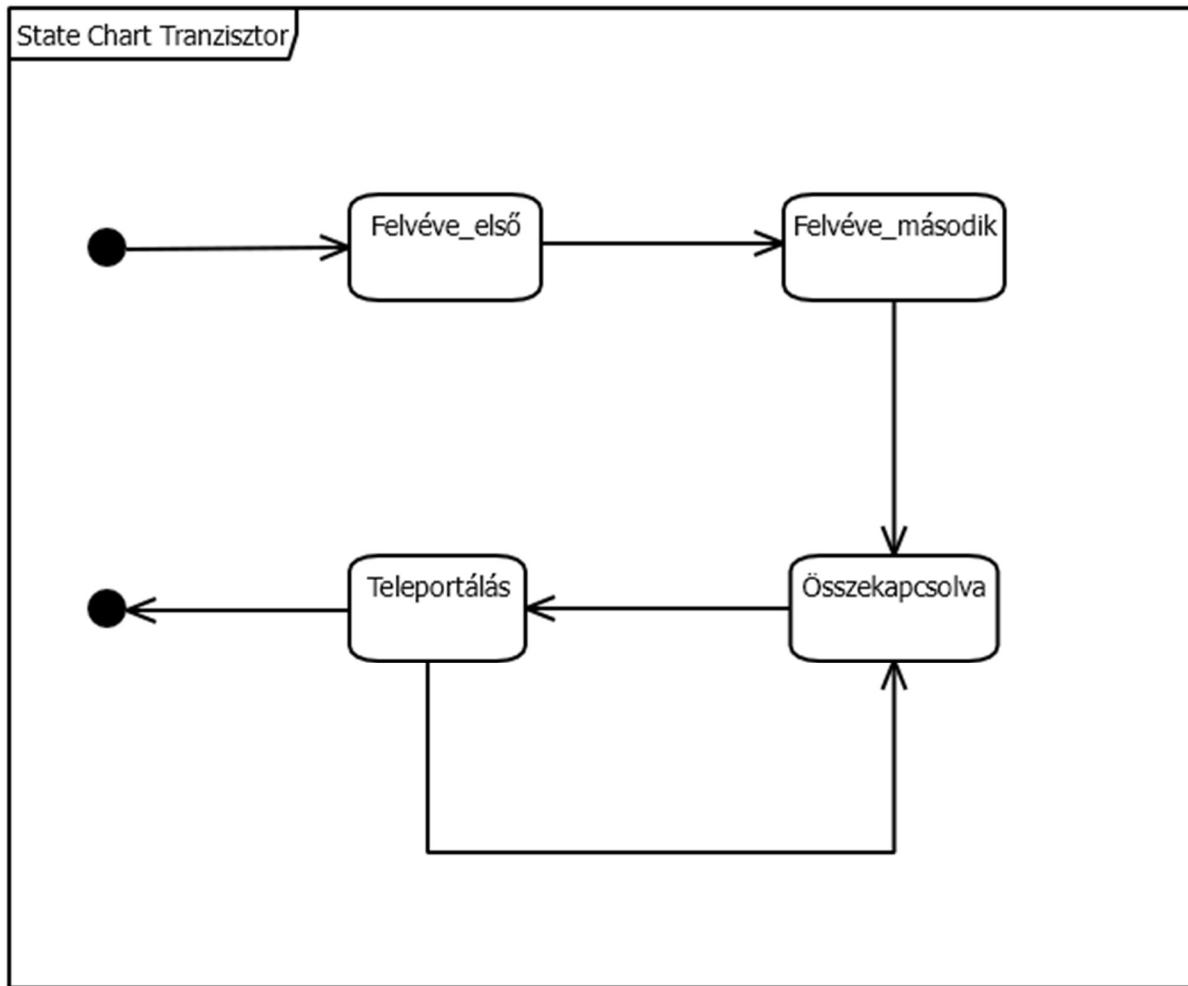
- **Metódusok:**

- + use(): void

A függvény meghívása után a játékos elteleportál abba a Szobába, ami az aktivált tranzisztor társát tartalmazzal.

- + setTars(Tranzisztor): void

A függvény beállítja, hogy az adott tranzisztor melyik másikkal van összeköttetésben.



8.1.14 Rongy

- **Felelősség:**

Az osztály a játékban megtalálható Rongy tárgy megvalósításáért felel. A rongyat egy adott ideig tudja használni a játékos, ami idő alatt meg tudja a segítségével bénítani a vele egy szobában lévő oktatókat. Mivel ez a tárgy egy passzív tárgy, ezért, ha egy szobába kerül egy hallgató, akinél van egy rongy, egy oktatával, akkor a tárgy hatása azonnal bekövetkezik.

- **Ősosztályok:**

Passzív Tárgy

- **Interfészek:**

-

- **Attribútumok:**

- - timer: Timer

Meghatározza, hogy mennyi ideig használható meg a Rongy.

- **Metódusok:**

- + tick(): void

Segít eldönteni, hogy pontosan meddig védi meg a játékos a Rongy az oktatók ellen. Amint a tárgyat magához veszi a hallgató, elindul egy Timer, ami idő alatt a tárgy használható. Ennek a Timer-nek a megfelelő nyomon követésében segít a függvény. Ha lejár az idő, akkor a tárgy eltűnik a hallgató inventory-ából.

8.1.15 Söröspohár

- **Felelősség:**

Az osztály a játékban megtalálható Söröspohár tárgy megvalósításáért felel. A söröspoharat egy adott ideig tudja használni a játékos, ami idő alatt meg tudja védeni a tárgy a szobában található oktatóktól. Mivel ez a tárgy egy passzív tárgy, ezért, ha egy szobába kerül egy hallgató, akinél van egy rongy, egy oktatával, akkor a tárgy hatása azonnal bekövetkezik

- **Ősosztályok:**

Passzív Tárgy

- **Interfészek:**

-

- **Attribútumok:**

- - timer: Timer

Meghatározza, hogy mennyi ideig használható meg a Söröspohár.

- **Metódusok:**

- + tick(): void

Segít eldönten, hogy pontosan meddig védi meg a játékost a Söröspohár az oktatók ellen. Amint a tárgyat magához veszi a hallgató, elindul egy Timer, ami idő alatt a tárgy használható. Ennek a Timer-nek a megfelelő nyomon követésében segít a függvény. Ha lejár az idő, akkor a tárgy eltűnik a hallgató inventory-ából.

8.1.16 TVSZ

- **Felelősség:**

Az osztály a játékban megtalálható TVSZ tárgy megvalósításáért felel. A TVSZ-t három alkalommal tudja használni a játékos, ami idő alatt meg tudja védeni a tárgy a szobában található oktatóktól. Mivel ez a tárgy egy passzív tárgy, ezért, ha egy szobába kerül egy hallgató, akinél van egy rongy, egy oktatóval, akkor a tárgy hatása azonnal bekövetkezik.

- **Ősosztályok:**

Passzív Tárgy

- **Interfészek:**

-

- **Attribútumok:**

-

- **Metódusok:**

-

8.1.17 Maszk

- **Felelősség:**

Az osztály a játékban megtalálható Maszk tárgy megvalósításáért felel. A Maszkot csak egy adott ideig tudja használni a játékos, ami idő alatt meg tudja védeni a tárgy egy esetleges gázos szoba hatásaitól. Mivel ez a tárgy egy passzív tárgy, ezért, ha a játékos belép egy gázos szobába, úgy azonnal felhasználásra kerül a Maszk, ha a hallgató birtokol egyet.

- **Ősosztályok:**

Passzív Tárgy

- **Interfészek:**

-

- **Attribútumok:**

-

- **Metódusok:**

-

8.1.18 Pálya

- **Felelősség:**

A játéktér kezeléséért és a léptetésért felelős. Eltárolja a játékban részt vevő karaktereket, valamint, legenerája a pályát, és lépteti a takarító, és oktató típusú karaktereket.

- **Ősosztályok:**

- **Interfész:**

-

- **Attribútumok:**

- hallgatok : ArrayList<Hallgato>
- oktatok : ArrayList<Oktato>
- szobak : ArrayList<Szoba>
- takaritok: ArrayList<Takarito>

- **Metódusok:**

- + general() : void

Ciklus amíg nem jön START szó

 hallgató hozzáadása

Ciklus hallgatók száma/3ig

 oktató hozzáadása

Ciklus amíg nem értünk a fájl végére

 sor eltárolása

 sor szétszedése space mentén

 első elem a szoba

 2. elem a gazos attribútuma

 3. elem a elatkozott szoba-e

 4. elem a befogadokepesseg attribútuma

 5-7. elem a tárgy

 többi elem a szoba szomszédja

Takarito hozzáadása

Karakterek lerakása a pályára

- + leptet() : void

Ciklus az oktató lista végéig

 Mozgás az oktatóval egy random szobába

Takarító mozgatása

Szobák osztása

Szobák összevonása

Elátkozott ajtók eltüntetése
Elátkozott ajtók előtüntetése

8.1.19 Game

- **Felelősség:**
Az idő műlásáért, illetve a játék kezdéséért és befejezéséért felel. Eltárol egy pályát és egy számlálót.
- **Ősosztályok:**
- **Interfészek:**
 -
- **Attribútumok:**
 - palya : Palya
 - szamlalo : int
 - logChange : LogarlecPropertyChangeListener
- **Metódusok:**
 - + start() : void
 - időzítő indítása
 - pálya generálása
 - + endgame() : void

HA az időzítő eléri a maximumot AKKOR játék vége vesztesen
 HA az utolsó hallgató is kibukott AKKOR játék vége vesztesen
 HA a LogarlecPropertyChangeListener szólt AKKOR játék vége győztesen.

8.1.20 LogarlecPropertyChangeListener

- **Felelősség:**
Azért felel, hogy figyelje mikor lesz felvéve a logarléc. Ha ez megtörténik akkor szól a pályának, hogy vége lehet a játéknak, a hallgató nyert.
- **Ősosztályok:**
- **Interfészek:**
 -
- **Attribútumok:**
 -
- **Metódusok:**
 - + propertyChange(PropertyChangeEvent e) : void

meghívja az endgame() függvényt

8.1.20 Logarléc

- **Felelősség:**
A logarléc tárgy osztálya. Ennek a tárgynak a felvételével lehet megnyerni a játékot.
- **Ősosztályok:**
 - Tárgy
- **Interfészek:**
 -
- **Attribútumok:**
 -
- **Metódusok:**
 -

8.2 A tesztek részletes tervei, leírásuk a teszt nyelvén

8.2.1 Teszteset1

- **Leírás**
Egy hallgató hozzáadása egy szobához.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Hallgató hozzáadása a szobához sikeresen megtörténik-e. Várható hibajegy: nem sikerül.
- **Bemenet**
create_room szoba1
add_character hallgato szoba1 h1
- **Kimenet**
Szoba hozzáadása sikeres.
Karakter sikeresen hozzáadva.

8.2.2 Teszteset2

- **Leírás**
Egy oktató hozzáadása egy szobához.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Oktató hozzáadása a szobához sikeresen megtörténik-e. Várható hibajegy: nem sikerül.
- **Bemenet**
create_room szoba1
add_character oktato szoba1 o1
- **Kimenet**
Szoba hozzáadása sikeres.
Karakter sikeresen hozzáadva.

8.2.3 Teszteset3

- **Leírás**

Egy takarító hozzáadása egy szobához.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Takarító hozzáadása a szobához sikeresen megtörténik-e. Várható hibajegy: nem sikerül.

- **Bemenet**

```
create_room szoba1
add_character takarito szoba1 t1
```

- **Kimenet**

Szoba hozzáadása sikeres.
Karakter sikeresen hozzáadva.

8.2.4 Teszteset4

- **Leírás**

Hallgató mozgása egyik szobából a másikba.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hallgató mozgása egy szobából a másikba sikeresen végbemegye-e. Várható hibajegy: nem fér be.

- **Bemenet**

```
create_room szoba1
create_room szoba2
connect szoba1 szoba2
add_character hallgato szoba1 h1
move h1 szoba2
```

- **Kimenet**

Szoba hozzáadása sikeres.
Szoba hozzáadása sikeres.
Szobák összekapcsolása sikeres.
Karakter sikeresen hozzáadva.
Karakter mozgása sikeres.

8.2.5 Teszteset5

- **Leírás**

Hallgató mozgása egy olyan szobába, ahol van oktató, de a hallgatónál van söröspohár, ami megvédi.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hallgatót sikeresen megvédi-e a söröspohár. Várható hibajegy: nem fér be a szobába, nem védi meg, és kibukik.

- **Bemenet**

```
create_room szoba1
create_room szoba2
connect szoba1 szoba2
add_character hallgato szoba1 h1
add_character oktato szoba2 o1
spawn_item sorospohar szoba1 s1
pick h1 s1
move h1 szoba2
```

- **Kimenet**

Szoba hozzáadása sikeres.
Szoba hozzáadása sikeres.
Szobák összekapcsolása sikeres.

Karakter sikeresen hozzáadva.
 Karakter sikeresen hozzáadva.
 Tárgy hozzáadása sikeres.
 Tárgy felvétele sikeres.
 Karakter mozgása sikeres.

8.2.6 Teszteset6

- **Leírás**
Oktató mozgása egyik szobából a másikba.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Oktató mozgása egy szobából a másikba sikeresen végbemegy-e. Várható hibajegy: nem fér be.
- **Bemenet**
create_room szoba1
create_room szoba2
connect szoba1 szoba2
add_character oktato szoba1 o1
move o1 szoba2
- **Kimenet**
Szoba hozzáadása sikeres.
Szoba hozzáadása sikeres.
Szobák összekapcsolása sikeres.
Karakter sikeresen hozzáadva.
Karakter mozgása sikeres.

8.2.7 Teszteset7

- **Leírás**
Takarító mozgása egyik szobából a másikba.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Takarító mozgása egy szobából a másikba sikeresen végbemegy-e. Várható hibajegy: nem fér be.
- **Bemenet**
create_room szoba1
create_room szoba2
connect szoba1 szoba2
add_character takarito szoba1 t1
move t1 szoba2
- **Kimenet**
Szoba hozzáadása sikeres.
Szoba hozzáadása sikeres.
Szobák összekapcsolása sikeres.
Karakter sikeresen hozzáadva.
Karakter mozgása sikeres.

8.2.8 Teszteset8

- **Leírás**
Hallgató mozgása egy olyan szobába, ahol van oktató, de a hallgatónál van tvsz, ami megvédi.
- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hallgatót sikeresen megvédi-e a tvsz. Várható hibajegy: nem fér be a szobába, nem védi meg, és kibukik.

- **Bemenet**

```
create_room szoba1
create_room szoba2
connect szoba1 szoba2
add_character hallgato szoba1 h1
add_character oktato szoba2 o1
spawn_item tvsz szoba1 tvsz1
pick h1 tvsz1
move h1 szoba2
```

- **Kimenet**

Szoba hozzáadása sikeres.
 Szoba hozzáadása sikeres.
 Szobák összekapcsolása sikeres.
 Karakter sikeresen hozzáadva.
 Karakter sikeresen hozzáadva.
 Tárgy hozzáadása sikeres.
 Tárgy felvétele sikeres.
 Karakter mozgása sikeres.

8.2.9 Teszteset9

- **Leírás**

Hallgató mozgása egy olyan szobába, ami gázos, de a hallgatonál van maszk, ami megvédi, így nem veszti el az eszméletét.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hallgatót sikeresen megvédi-e a maszk a gáztól. Várható hibajegy: nem fér be a szobába, nem védi meg, és eszméletetveszt.

- **Bemenet**

```
create_room szoba1
create_room gazosszoba1
connect szoba1 gazosszoba1
add_character hallgato szoba1 h1
spawn_item maszk szoba1 m1
pick h1 m1
move h1 gazosszoba1
```

- **Kimenet**

Szoba hozzáadása sikeres.
 Szoba hozzáadása sikeres.
 Szobák összekapcsolása sikeres.
 Karakter sikeresen hozzáadva.
 Tárgy hozzáadása sikeres.
 Tárgy felvétele sikeres.
 Karakter mozgása sikeres.

8.2.10 Teszteset10

- **Leírás**

Oktató mozgása egy olyan szobába, ami gázos, de az oktatónál van maszk, ami megvédi, így nem veszti el az eszméletét.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Oktató sikeresen megvédi-e a maszk a gáztól. Várható hibajegy: nem fér be a szobába, nem védi meg, és eszméletetveszt.

- **Bemenet**

```
create_room szoba1
create_room gazosszoba1
connect szoba1 gazosszoba1
add_character oktato szoba1 o1
spawn_item maszk szoba1 m1
pick o1 m1
move o1 gazosszoba1
```

- **Kimenet**

Szoba hozzáadása sikeres.
 Szoba hozzáadása sikeres.
 Szobák összekapcsolása sikeres.
 Karakter sikeresen hozzáadva.
 Tárgy hozzáadása sikeres.
 Tárgy felvétele sikeres.
 Karakter mozgása sikeres.

8.2.11 Teszteset11

- **Leírás**

Hallgató mozgása egy olyan szobába, ami gázos, és nincs nála semmilyen tárgy, ami megvédi a gáztól, így eszméletetveszt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hallgató eszméletetveszt-e. Várható hibajegy: nem fér be a szobába, nem veszt eszméletet.

- **Bemenet**

```
create_room szoba1
create_room gazosszoba1
connect szoba1 gazosszoba1
add_character hallgato szoba1 h1
move h1 gazosszoba1
```

- **Kimenet**

Szoba hozzáadása sikeres.
 Szoba hozzáadása sikeres.
 Szobák összekapcsolása sikeres.
 Karakter sikeresen hozzáadva.
 Karakter mozgása sikeres.

8.2.12 Teszteset12

- **Leírás**

Oktató mozgása egy olyan szobába, ami gázos, és nincs nála semmilyen tárgy, ami megvédi a gáztól, így eszméletetveszt.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Oktató eszméletetveszt-e. Várható hibajegy: nem fér be a szobába, nem veszt eszméletet.

- **Bemenet**

```
create_room szoba1
create_room gazosszoba1
connect szoba1 gazosszoba1
```

```
add_character oktato szoba1 o1
move o1 gazosszoba1
```

- **Kimenet**

Szoba hozzáadása sikeres.
 Szoba hozzáadása sikeres.
 Szobák összekapcsolása sikeres.
 Karakter sikeresen hozzáadva.
 Karakter mozgása sikeres.

8.2.13 Teszteset13

- **Leírás**

Hallgató mozgása egy olyan szobába, ahol van oktató, de a hallgatónál nincs olyan tárgy, ami megvédené őt, ezért kibukik.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Hallgatót kibukott-e. Várható hibajegy: nem fér be a szobába, nem bukik ki.

- **Bemenet**

```
create_room szoba1
create_room szoba2
connect szoba1 szoba2
add_character hallgato szoba1 h1
add_character oktato szoba2 o1
move h1 szoba2
```

- **Kimenet**

Szoba hozzáadása sikeres.
 Szoba hozzáadása sikeres.
 Szobák összekapcsolása sikeres.
 Karakter sikeresen hozzáadva.
 Karakter sikeresen hozzáadva.
 Karakter mozgása sikeres.

8.2.14 Teszteset14

- **Leírás**

Tárgyak elhelyezésének tesztje. Mivel minden tárgyra azonosan működik, és egy nem túl bonyolult funkció tesztje, ezért csak egy tárgyra tesztelünk.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Tárgy elhelyezése adott szobában sikeresen megtörténik-e. Várható hibajegy: nem sikerül elhelyezni.

- **Bemenet**

```
create_room szoba1
spawn_item maszk szoba1 m1
```

- **Kimenet**

Szoba hozzáadása sikeres.
 Tárgy hozzáadása sikeres.

8.2.15 Teszteset15

- **Leírás**

A paraméterként megadott fájl tartalmát használva betölti az abban leírtakat.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Valóban betöltődik-e a fájl tartalma, ahogy vártuk. Várható hibajegy: Nem sikerült a betöltés művelet.

- **Bemenet**
load file1
- **Kimenet**
Fájl sikeresen betöltve.

8.2.16 Teszteset16

- **Leírás**
A paraméterként megadott fájlba kimentjük a játék jelenlegi állapotát.
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Valóban megtörténik-e a fájlba mentés a helyes adatokkal, ahogy vártuk. Várható hibajegy: Nem sikerült a mentés művelet.
- **Bemenet**
save file1
- **Kimenet**
Fájl sikeresen mentve.

8.2.17 Teszteset17

- **Leírás**
Két szoba egyesülése
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Sikeresen egyesül-e a szoba, az egyesült szoba tulajdonságai megfelelőek lesznek-e
- **Bemenet**
create_room szoba1 2
create_room szoba2 5
create_room szoba3 1
connect szoba1 szoba3
connect szoba3 szoba1
spawn_item maszk szoba2 m1
merge szoba1 szoba2
- **Kimenet**
egyesules sikeres
befogadokepesseg sikeres
szomszed sikeres
targy sikeres

8.2.18 Teszteset18

- **Leírás**
Egy gázos és egy nem gázos szoba egyesítése
- **Ellenőrzött funkcionalitás, várható hibahelyek**
Sikeresen egyesül-e a szoba, az egyesült szoba tulajdonságai megfelelőek lesznek-e
- **Bemenet**
create_room szoba1 2
create_room szoba2 gazos 5
merge szoba1 szoba2
- **Kimenet**
egyesules sikeres
gazos egyesules sikeres

8.2.19 Teszteset19

- **Leírás**
Egy elátkozott és egy nem elátkozott szoba egyesítése
- **Ellenőrzött funkcionalitás, várható hibahelyek**

Sikeresen egyesül-e a szoba, az egyesült szoba tulajdonságai megfelelőek lesznek-e

- **Bemenet**

```
create_room szoba1 2
create_room szoba2 elatkozott 5
merge szoba1 szoba2
```

- **Kimenet**

```
egyesules sikeres
elatkozott egyesules sikeres
```

8.2.20 Teszteset20

- **Leírás**

Egy elátkozott nem gázos és egy nem elátkozott gázos szoba egyesítése

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Sikeresen egyesül-e a szoba, az egyesült szoba tulajdonságai megfelelőek lesznek-e

- **Bemenet**

```
create_room szoba1 gazos 2
create_room szoba2 elatkozott 5
merge szoba1 szoba2
```

- **Kimenet**

```
egyesules sikeres
elatkozott egyesules sikeres
gazos egyesules sikeres
```

8.2.21 Teszteset21

- **Leírás**

Két egyesült szoba megfelelően osztódik-e

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Sikeresen osztódik-e a szoba, visszaállnak-e az egyesülés előtti állapotukba

- **Bemenet**

```
create_room szoba1 2
create_room szoba2 5
create_room szoba3 1
connect szoba1 szoba3
connect szoba3 szoba1
spawn_item maszk szoba2 m1
merge szoba1 szoba2
split szoba1
```

- **Kimenet**

```
osztodas sikeres
befogadokepesseg sikeres
szomszed sikeres
targy sikeres
```

8.2.22 Teszteset22

- **Leírás**

Egy gázos és egy nem gázos egyesült szoba megfelelően osztódik-e

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Sikeresen osztódik-e a szoba, visszaállnak-e az egyesülés előtti állapotukba

- **Bemenet**

```
create_room szoba1 2
create_room szoba2 gazos 5
merge szoba1 szoba2
```

split szoba1

- **Kimenet**

osztodas sikeres

gazos osztodas sikeres

8.2.23 Teszteset23

- **Leírás**

Egy elátkozott és egy nem elátkozott egyesült szoba megfelelően osztódik-e

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Sikeresen osztódik-e a szoba, visszaállnak-e az egyesülés előtti állapotukba

- **Bemenet**

create_room szoba1 2

create_room szoba2 elatkozott 5

merge szoba1 szoba2

split szoba1

- **Kimenet**

osztodas sikeres

elatkozott osztodas sikeres

8.2.24 Teszteset24

- **Leírás**

Egy gázos nem elátkozott és egy elátkozott nem gázos egyesült szoba megfelelően osztódik-e

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Sikeresen osztódik-e a szoba, visszaállnak-e az egyesülés előtti állapotukba

- **Bemenet**

create_room szoba1 gázos 2

create_room szoba2 elatkozott 5

merge szoba1 szoba2

split szoba1

- **Kimenet**

osztodas sikeres

elatkozott osztodas sikeres

gázos osztodas sikeres

8.2.25 Teszteset25

- **Leírás**

Egy korábban egyesült, de már visszaosztódott szobára meghívódik az osztódás

- **Ellenőrzött funkcionalitás, várható hibahelyek**

Ebben az esetben nem kell történnie semminek, hiszen csak olyan szoba osztódhat, amely egyesült már

- **Bemenet**

create_room szoba1 2

create_room szoba2 5

merge szoba1 szoba2

split szoba1

split szoba1

- **Kimenet**

elso osztodas sikeres

masodik nem osztodas sikeres

8.2.26 Teszteset26

- **Leírás**

Egy tárgyat felvesz egy hallgató.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A tárgy megjelenik-e a karakternél, akkor sikeres ha nem akkor a teszt megbukott.

- **Bemenet**

```
create_room szoba1
spawn_item söröspohar1
add_character hallgato1
pick hallgato1 söröspohar1
```

- **Kimenet**

A tárgyfelvétel sikeres.

8.2.26 Teszteset27

- **Leírás**

Egy tárgyat eldob egy hallgató.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A tárgy eltűnik-e a karaktertől, akkor sikeres ha nem akkor a teszt megbukott.

- **Bemenet**

```
create_room szoba1
spawn_item söröspohar1
add_character hallgato1
pick hallgato1 söröspohar 1
throw hallgato1 sorospohar1
```

- **Kimenet**

A tárgy sikeresen eltűnt.

8.2.26 Teszteset28

- **Leírás**

Egy tárgyat használ egy hallgató.

- **Ellenőrzött funkcionalitás, várható hibahelyek**

A tárgy use függvénye helyesen hívódik-e. Ha nem hívódik meg akkor sikertelen a teszt.

- **Bemenet**

```
create_room szoba1
spawn_item camembert1
add_character hallgato1
pick hallgato1 camembert1
use hallgato1 camembert1
```

- **Kimenet**

A tárgy használata sikeres.

8.3 A tesztelést támogató programok tervei

Minden tesztesethez tartozni fog egy bemeneti fájl, ahol a parancsok lesznek az adott teszt vizsgálatához, illetve egy kimeneti fájl ahol az elvárt eredmény lesz. A load parancssal betöltött fájlból elindítjuk a tesztesetet és a kimenetet összehasonlítjuk a kimeneti fájllal. Ha egyezik akkor kiírja, hogy SUCCESS, ha nem akkor, hogy FAIL. Ezen kívül parancsosból is indíthatók tesztek. A tesztelésre külön programot nem használunk. Lesz egy testAll parancs ami az összes tesztet futtatja.

Napló

Kezdet	Időtartam	Résztvevők	Leírás
2024.04.10. 21:30	1 óra	Csapat	Értekezlet: Döntések: Javítások megbeszélése. Feladatok kiosztása.
2024.04.12. 9:00	3 óra	Czotter	8.2.1 - 8.2.14 tesztesetek megírása. Javítások 8.0.1 és 8.0.2 elkészítése. Szoba, ElátkozottSzoba osztályleírások egy részének elkészítése.
2024.04.14. 18:00	2 óra	Tarsoly	Szoba befejezése, és az osztódás és egyesülés tesztjeinek megírása.
2024.04.14. 21:00	1 óra	Csapat	Értekezlet: Döntések, kérdések megbeszélése, részletes tervezet átbeszélése.
2024.04.14. 23:30	2 óra	Németh	Tárgyakkal kapcsolatos tervezet.
2024.04.14. 23:00	2,5 óra	Schulcz	Javítás és Karakterekkel kapcsolatos osztályok és metódusok tervezet, állapotgépek.
2024.04.15. 11	2 óra	Hermann	Game, Pálya és Logarléc osztályok megírása, valamint a Throw, Pick, Use parancsokhoz a tesztek megírásam dokumentum szerkesztése.

Prototípus beadása

35 – pupakok

Konzulens:
Koczó Attila

Csapattagok:

Tarsoly Levente	E1IK75	tarsolyl@edu.bme.hu
Németh Gergely	TA9G0R	gergo.h.nemeth@gmail.com
Schulcz Gábor	EP3I9R	schulcz.gabor6@gmail.com
Czotter Benedek	TFB4FY	czottibeni@gmail.com
Hermann Máté Béla	T83K5I	mate3829@gmail.com

2024.04.30

10. Prototípus beadása

10.1 Fordítási és futtatási útmutató

10.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Game.java	7 kb	2024.04.17.	Game implementációja.
Hallgato.java	5 kb	2024.04.17.	Hallgato implementációja.
Karakter.java	4 kb	2024.04.17.	Karakter implementációja.
Oktato.java	3 kb	2024.04.17.	Oktato implementációja.
Takarito.java	2 kb	2024.04.17.	Takarito implementációja.
Vedettseg.java	1 kb	2024.04.17.	Vedettseg implementációja.
Palya.java	12 kb	2024.04.17.	Palya implementációja.
Proto.java	29 kb	2024.04.17.	Proto implementációja.
ElatkozottSzoba.java	2 kb	2024.04.17.	ElatkozottSzoba implementációja.
Szoba.java	13 kb	2024.04.17.	Szoba implementációja.
AktivTargy.java	1 kb	2024.04.17.	AktivTargy implementációja.
Camembert.java	1 kb	2024.04.17.	Camembert implementációja.
Funkcio.java	1 kb	2024.04.17.	Funkcio implementációja.
Legfrissito.java	1 kb	2024.04.17.	Legfrissito implementációja.
Logarlec.java	2 kb	2024.04.17.	Logarlec implementációja.
LogarlecPropertyChangeListener.java	1 kb	2024.04.17.	LogarlecPropertyChangeListerner implementációja.
Maszk.java	1 kb	2024.04.17.	Maszk implementációja.
PasszivTargy.java	1 kb	2024.04.17.	PasszivTargy implementációja.
Rongy.java	2 kb	2024.04.17.	Rongy implementációja.
Sorospohar.java	3 kb	2024.04.17.	Sorospohar implementációja.
Targy.java	3 kb	2024.04.17.	Targy implementációja.
Tranzisztor.java	3 kb	2024.04.17.	Tranzisztor implementációja.
TVSZ.java	1 kb	2024.04.17.	TVSZ implementációja.
Main.java	2 kb	2024.04.17.	Main implementációja.
input1.txt	1 kb	2024.04.29.	Teszteset bemenete.
input2.txt	1 kb	2024.04.29.	Teszteset bemenete.
input3.txt	1 kb	2024.04.29.	Teszteset bemenete.
input4.txt	1 kb	2024.04.29.	Teszteset bemenete.
input5.txt	1 kb	2024.04.29.	Teszteset bemenete.
input6.txt	1 kb	2024.04.29.	Teszteset bemenete.
input7.txt	1 kb	2024.04.29.	Teszteset bemenete.
input8.txt	1 kb	2024.04.29.	Teszteset bemenete.
input9.txt	1 kb	2024.04.29.	Teszteset bemenete.
input10.txt	1 kb	2024.04.29.	Teszteset bemenete.
input11.txt	1 kb	2024.04.29.	Teszteset bemenete.
input12.txt	1 kb	2024.04.29.	Teszteset bemenete.
input13.txt	1 kb	2024.04.29.	Teszteset bemenete.
input14.txt	1 kb	2024.04.29.	Teszteset bemenete.

input15.txt	1 kb	2024.04.29.	Teszteset bemenete.
input16.txt	1 kb	2024.04.29.	Teszteset bemenete.
input17.txt	1 kb	2024.04.29.	Teszteset bemenete.
input18.txt	1 kb	2024.04.29.	Teszteset bemenete.
input19.txt	1 kb	2024.04.29.	Teszteset bemenete.
input20.txt	1 kb	2024.04.29.	Teszteset bemenete.
input21.txt	1 kb	2024.04.29.	Teszteset bemenete.
input22.txt	1 kb	2024.04.29.	Teszteset bemenete.
input23.txt	1 kb	2024.04.29.	Teszteset bemenete.
input24.txt	1 kb	2024.04.29.	Teszteset bemenete.
input25.txt	1 kb	2024.04.29.	Teszteset bemenete.
input26.txt	1 kb	2024.04.29.	Teszteset bemenete.
input27.txt	1 kb	2024.04.29.	Teszteset bemenete.
input28.txt	1 kb	2024.04.29.	Teszteset bemenete.
input29.txt	1 kb	2024.04.29.	Teszteset bemenete.
input30.txt	1 kb	2024.04.29.	Teszteset bemenete.
input31.txt	1 kb	2024.04.29.	Teszteset bemenete.
input32.txt	1 kb	2024.04.29.	Teszteset bemenete.
input33.txt	1 kb	2024.04.29.	Teszteset bemenete.
output1.txt	1 kb	2024.04.29.	Teszteset kimenete.
output2.txt	1 kb	2024.04.29.	Teszteset kimenete.
output3.txt	1 kb	2024.04.29.	Teszteset kimenete.
output4.txt	1 kb	2024.04.29.	Teszteset kimenete.
output5.txt	1 kb	2024.04.29.	Teszteset kimenete.
output6.txt	1 kb	2024.04.29.	Teszteset kimenete.
output7.txt	1 kb	2024.04.29.	Teszteset kimenete.
output8.txt	1 kb	2024.04.29.	Teszteset kimenete.
output9.txt	1 kb	2024.04.29.	Teszteset kimenete.
output10.txt	1 kb	2024.04.29.	Teszteset kimenete.
output11.txt	1 kb	2024.04.29.	Teszteset kimenete.
output12.txt	1 kb	2024.04.29.	Teszteset kimenete.
output13.txt	1 kb	2024.04.29.	Teszteset kimenete.
output14.txt	1 kb	2024.04.29.	Teszteset kimenete.
output15.txt	1 kb	2024.04.29.	Teszteset kimenete.
output16.txt	1 kb	2024.04.29.	Teszteset kimenete.
output17.txt	1 kb	2024.04.29.	Teszteset kimenete.
output18.txt	1 kb	2024.04.29.	Teszteset kimenete.
output19.txt	1 kb	2024.04.29.	Teszteset kimenete.
output20.txt	1 kb	2024.04.29.	Teszteset kimenete.
output21.txt	1 kb	2024.04.29.	Teszteset kimenete.
output22.txt	1 kb	2024.04.29.	Teszteset kimenete.
output23.txt	1 kb	2024.04.29.	Teszteset kimenete.
output24.txt	1 kb	2024.04.29.	Teszteset kimenete.
output25.txt	1 kb	2024.04.29.	Teszteset kimenete.
output26.txt	1 kb	2024.04.29.	Teszteset kimenete.
output27.txt	1 kb	2024.04.29.	Teszteset kimenete.
output28.txt	1 kb	2024.04.29.	Teszteset kimenete.
output29.txt	1 kb	2024.04.29.	Teszteset kimenete.

output30.txt	1 kb	2024.04.29.	Teszteset kimenete.
output31.txt	1 kb	2024.04.29.	Teszteset kimenete.
output32.txt	1 kb	2024.04.29.	Teszteset kimenete.
output33.txt	1 kb	2024.04.29.	Teszteset kimenete.
ProtoTestMenu.txt	3 kb	2024.04.26	Proto teszt menüpontjainak bemenete.
Skeleton.txt	2 kb	2024.04.27	Skeleton bemenete.
Szoba_attributumok.txt	1 kb	2024.04.29	Szoba adatainak bemenete.
Szoba_szomszedok.txt	1 kb	2024.04.29	Szoba szomszédainak listája szöveges bemenetben.

10.1.2 Fordítás

Lépések:

- Nyissuk meg a forrásfájlokat tartalmazó könyvtárat cmdben
- Futtassuk a “javac Main.java” parancsot

10.1.3 Futtatás

- Futtassuk a “java Main” parancs segítségével a programot

10.2 Tesztek jegyzőkönyvei

10.2.1 Teszteset1

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:00

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:00
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.2 Teszteset2

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:02

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:02
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.3 Teszteset3

Tesztelő neve	Czotter Benedek
---------------	-----------------

Teszt időpontja	2024.04.29. 12:04
------------------------	-------------------

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:04
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.4 Teszteset4

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:06

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:06
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.5 Teszteset5

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:08

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:08
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.6 Teszteset6

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:10

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:10
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.7 Teszteset7

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:12

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:12

Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.8 Teszteset8

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:14

Tesztelő neve	Czotter Benedek
Teszt időpontja	2024.04.29. 12:14
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.9 Teszteset9

Tesztelő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:16

Tesztelő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:16
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.10 Teszteset10

Tesztelő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:18

Tesztelő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:18
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.11 Teszteset11

Tesztelő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:20

Tesztelő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:20
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.12 Teszteset12

Teszteleő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:22

Teszteleő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:22
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.13 Teszteset13

Teszteleő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:24

Teszteleő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:24
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.14 Teszteset14

Teszteleő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:26

Teszteleő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:26
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.15 Teszteset15

Teszteleő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:28

Teszteleő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:28
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.16 Teszteset16

Teszteleő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:30

Tesztelő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:30
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.17 Teszteset17

Tesztelő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:32

Tesztelő neve	Németh Gergő
Teszt időpontja	2024.04.29. 12:32
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.18 Teszteset18

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:34

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:34
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.19 Teszteset19

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:36

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:36
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.20 Teszteset20

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:38

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:38
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-

Változtatások	-
----------------------	---

10.2.21 Teszteset21

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:40

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:40
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.22 Teszteset22

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:42

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:42
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.23 Teszteset23

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:44

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:44
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.24 Teszteset24

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:46

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:46
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.25 Teszteset25

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:48

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:48
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.26 Teszteset26

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:50

Tesztelő neve	Schulcz Gábor
Teszt időpontja	2024.04.29. 12:50
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.27 Teszteset27

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:00

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:00
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.28 Teszteset28

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:01

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:01
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.29 Teszteset29

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:03

Tesztelő neve	Hermann Máté Béla
----------------------	-------------------

Teszt időpontja	2024.04.29. 13:03
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.30 Teszeset30

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:06

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:06
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.31 Teszeset31

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:10

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:10
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.32 Teszeset32

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:13

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:13
Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.2.33 Teszeset33

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:16

Tesztelő neve	Hermann Máté Béla
Teszt időpontja	2024.04.29. 13:16

Teszt eredménye	A teszt sikeres.
Lehetséges hibaok	-
Változtatások	-

10.3 Értékelés

Tag neve	Tag neptun	Munka százalékban
Czotter Benedek	TFB4FY	21
Németh Gergő	TA9G0R	17.5
Schulcz Gábor	EP3I9R	21
Tarsoly Levente	E1IK75	23
Hermann Máté Béla	T83K5I	17.5

10.4 Napló

Kezdet	Időtartam	Résznevők	Leírás
2024.04.17. 20:00	1,5 óra	Csapat	Értekezlet. Döntés: Program megírásának részfeladatainak kiosztása
2024.04.19. 15:00	1,5 óra	Tarsoly	Szoba nem karakterekhez fűződő részének implementálása, főmenü megírása
2024.04.19. 16:00	1 óra	Czotter	Szoba karakterekhez fűződő részének megírása.
2024.04.24. 21:00	1 óra	Csapat	Értekezlet. Döntés: Tesztek és a futó játék implementálásához való feladatok megbeszélése, kiosztása
2024.04.24. 22:00	1 óra	Czotter, Németh, Schulcz	Értekezlet: Döntés: Tesztek implementálásának megbeszélése, feladatok kiosztása.
2024.04.25. 22:00	1,5 óra	Tarsoly, Hermann	Értekezlet. Döntés: A futó játékhoz tartozó feladatok megbeszélése
2024.04.26. 14:00	3 óra	Hermann	Pálya implementálása
2024.04.27. 9:00	1,5 óra	Hermann	Randomizálás kapcsolásának megírása
2024.04.28. 19:00	2 óra	Tarsoly	Hallgató irányításának implementálása
2024.04.29. 9:00	15 óra	Czotter, Schulcz, Tarsoly	Proto implementálása.
2024.04.29. 16:00	8 óra	Németh	Proto implementálása.
2024.04.29. 9:00	8 óra	Hermann	Proto implementálása.

2024.04.30. 8:30	1 óra	Hermann	Dokumentum megtájékoztatása
------------------	-------	---------	--------------------------------

Grafikus felület specifikációja

35 – pupakok

Konzulens:
Koczó Attila

Csapattagok:

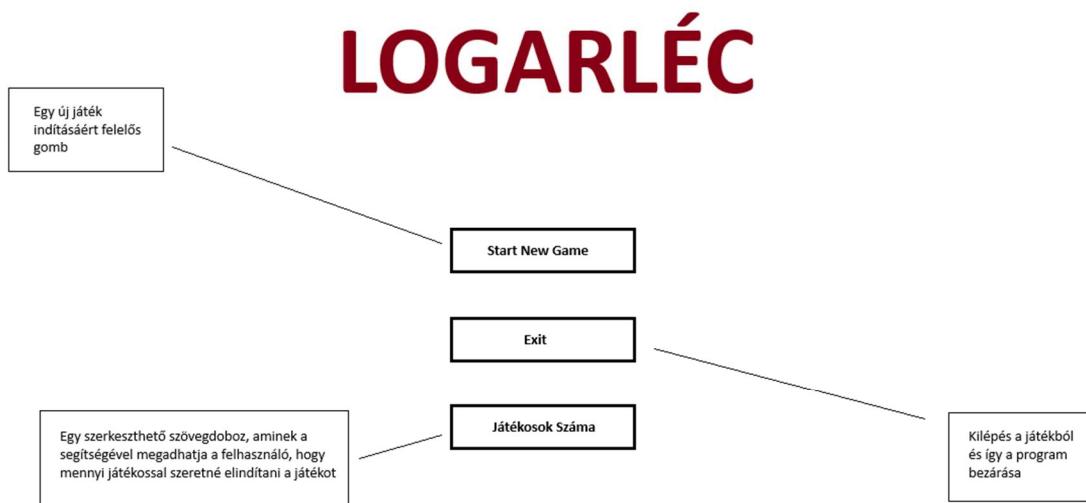
Tarsoly Levente	E1IK75	tarsoly1@edu.bme.hu
Németh Gergely	TA9G0R	gergo.h.nemeth@gmail.com
Schulcz Gábor	EP3I9R	schulcz.gabor6@gmail.com
Czotter Benedek	TFB4FY	czottiben1@gmail.com
Hermann Máté Béla	T83K5I	mate3829@gmail.com

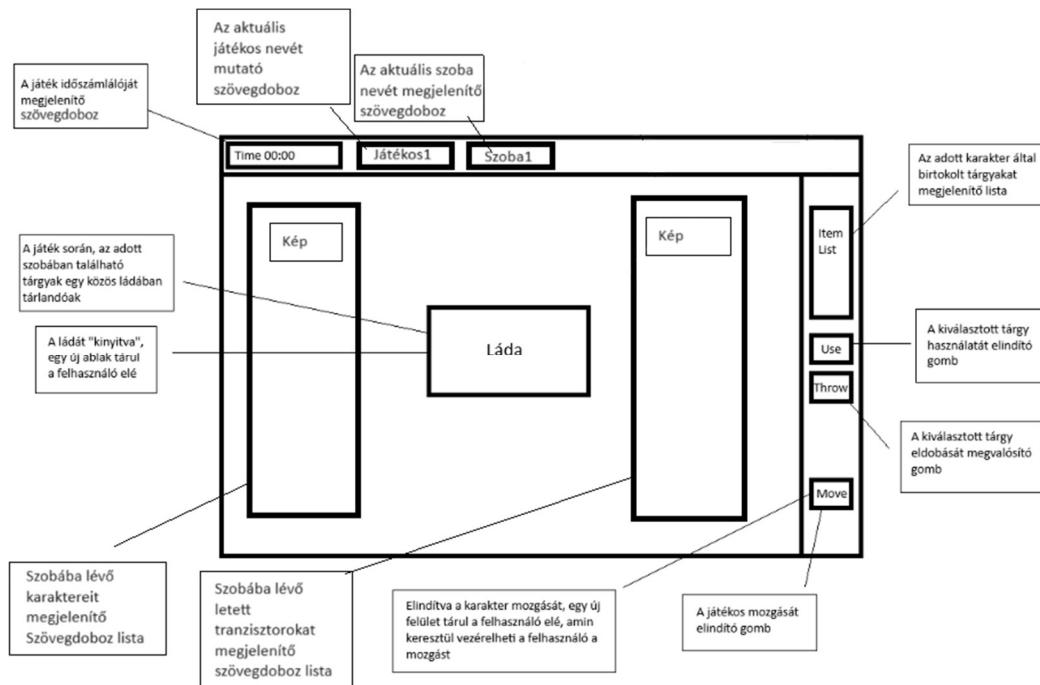
2024.05.06

11. Grafikus felület specifikációja

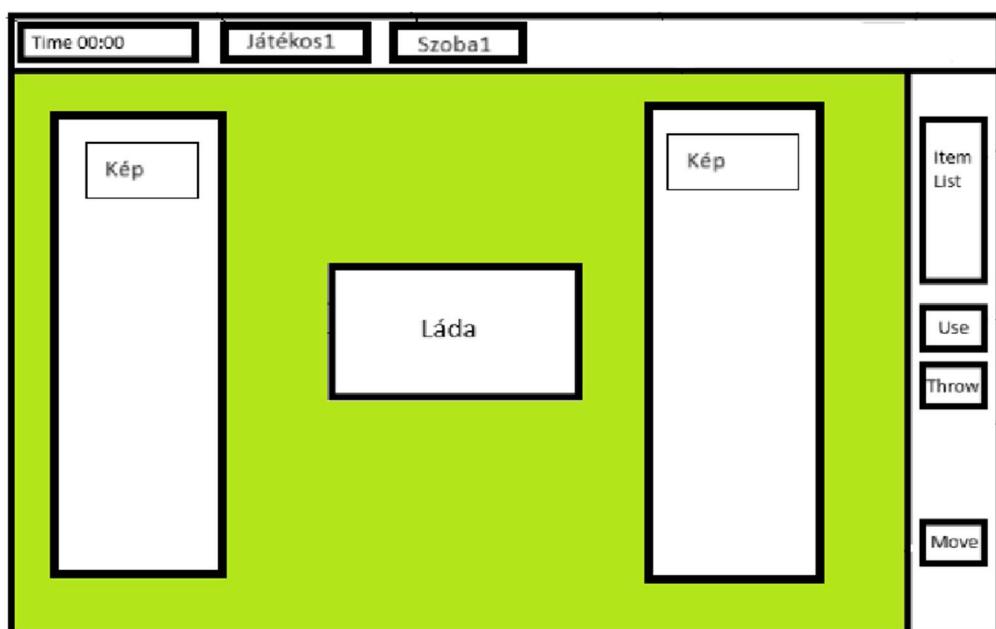
11.1 A grafikus interfész

HomeFrame:

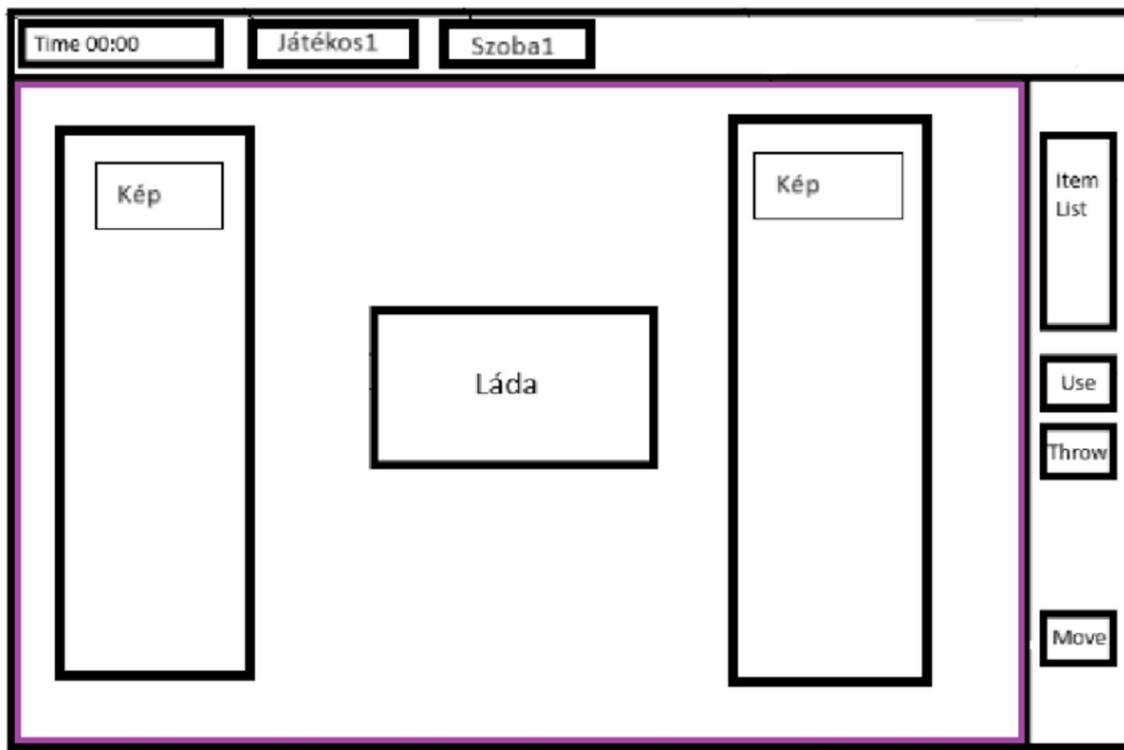


GameFrame:

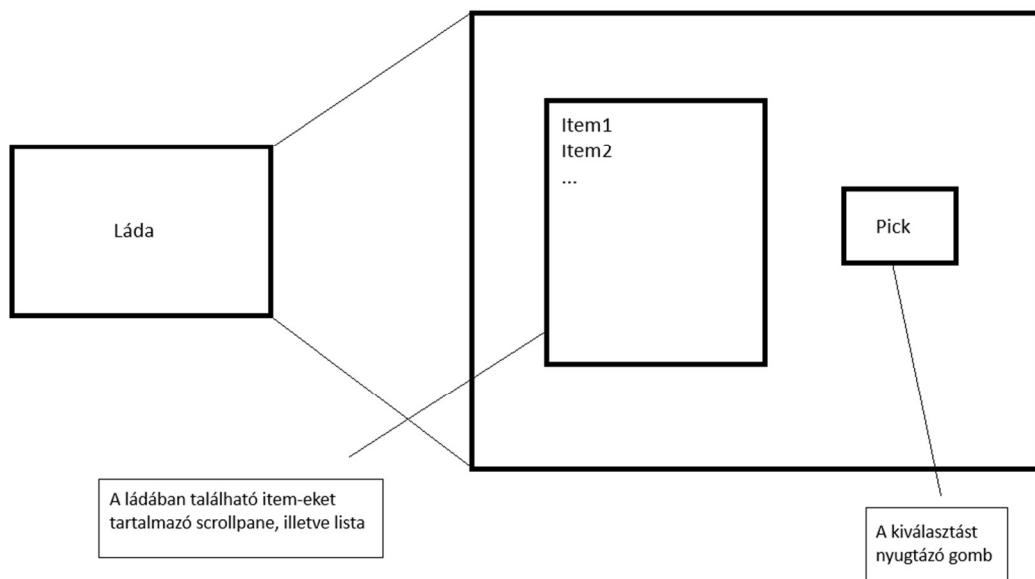
- Ha az adott szoba gázos, akkor a szobát “megtölti” a gáz, így a szoba hátttere zöld-re változik.

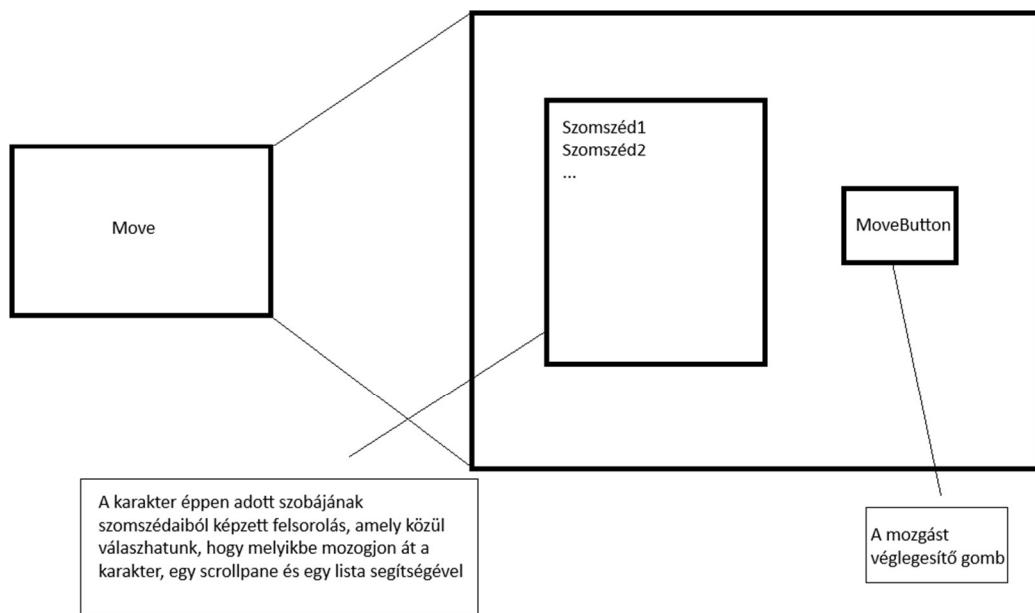


- Ha az adott szoba elátkozott, akkor a szoba kerete lila színre változik, jelezve a szoba elátkozott voltát.



ItemFrame:



RoomFrame:**Tárgyak megjelenítése grafikus felülettel:**

- Camembert



- Légfrissítő



- Tranzisztor



- Rongy



- Söröspohár

BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM

A Szeminárium X/30/2015-2016. ORsz. VIII. 31.)
száma határozata

A BME TANULMÁNYI ÉS VIZSGASZABÁLYZATRÓL

- TVSZ



- Maszk



- Logarléc



- A szobában az tárgyak tárolási helyeként szolgáló láda ikonja



- Hallgató



- Oktató



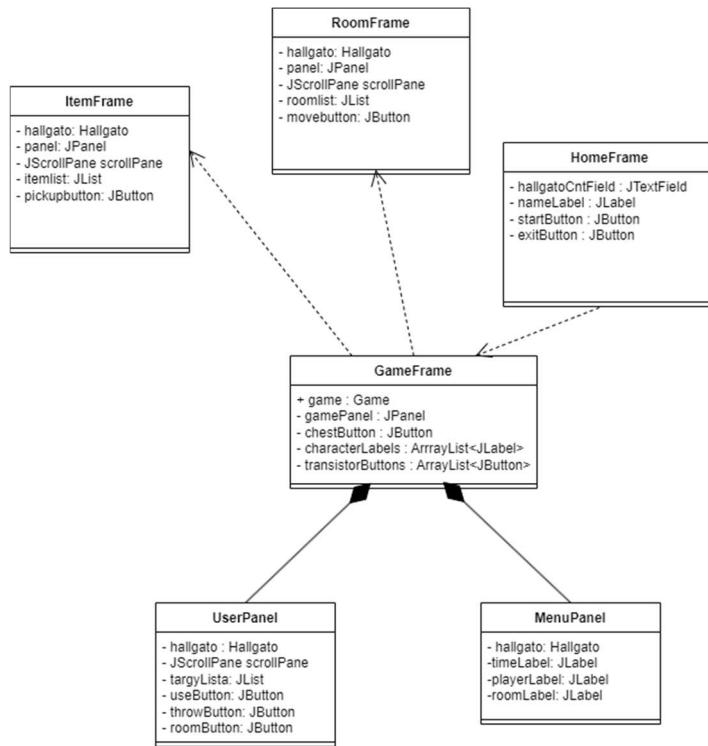
- Takarító

11.2 A grafikus rendszer architektúrája

11.2.1 A felület működési elve

A grafikus felületet MVC architektúrának megfelelően igyekeztünk megtervezni, ahol is elkülönül egymástól a program modellje, a megjelenítésért és a vezérlésért felelős logika. Az alapelvek közül a pull alapú választottuk, tehát a felület szól a modellnek, hogy változás történt, a felület ezután pedig újrarendezze magát annak függvényében, ahogy változott a modell állapota.

11.2.2 A felület osztály-struktúrája



11.3 A grafikus objektumok felsorolása

11.3.1 RoomFrame

Felelősség

Az osztály felelőssége az, hogy létrehozza azt a Framet, amelyen ki lehet választani a szomszédos szobák közül azt, ahova a hallgató a köre befejezését követően lépni szeretne.

Ősosztályok

JFrame

Interfészek

-

Attribútumok

- -panel: JPanel
 - Az objektumok elrendezését biztosító panel.
- - scrollpane: JScrollPane
 - A lista görgethetőségének biztosításáért felel
- -roomlist: JList
 - A szomszédos szobákat jeleníti meg.
- -movebutton: JButton

- Egyik szobából egy másikba lépést valósítja meg a hozzákapcsolt ActionListener segítségével..
- - hallgato: Hallgato
 - Az aktuális hallgatót adja meg.

Metódusok

-

11.3.2 ItemFrame

Felelősség

Az osztály felelőssége az, hogy létrehozza azt a Framet, amelyen ki lehet választani a szobában lévő tárgyak közül azokat, amelyeket fel szeretne venni a hallgató.

Ősosztályok

JFrame

Interfészek

-

Attribútumok

- -panel: JPanel
 - Az objektumok elrendezését biztosító panel.
- - scrollpane: JScrollPane
 - A lista görgethetőségének biztosításáért felel
- -itemlist: JList
 - A szobában felvehető tárgyak megjelenítését végzi.
- -pickupbutton: JButton
 - A szobában lévő tárgyak közül a kiválasztott tárgyat lehet vele felvenni a hozzákapcsolt ActionListener segítségével.
- - hallgato: Hallgato
 - Az aktuális hallgatót adja meg.

Metódusok

-

11.3.3 HomeFrame

Felelősség

Az osztály felelőssége az, hogy létrehozza azt a Framet, ami a program indításakor jön létre. Ezen tudjuk elindítani a játékot.

Ősosztályok

JFrame

Interfészek

-

Attribútumok

- -panel: JPanel
 - Az objektumok elrendezését biztosító panel.
- -exitButton: JButton
 - Kiléphetünk vele az alkalmazásból.
- -startButton: JButton
 - Ezzel indítjuk el a játékot.
- -hallgatoCount : JTextField
 - Ebbe beírva megadhatjuk hány játékos legyen.
- nameLabel : JLabel
 - Kiírja a játék nevét.

Metódusok**11.3.4 GameFrame****Felelősség**

Az osztály felel azért, hogy a fő játékkablak megjelenjen és mutassa a játékbeli változtatásokat.

Ősosztályok

JFrame

Interfészek

-

Attribútumok

- -gamePanel: JPanel
 - Az objektumok elrendezését biztosító panel.
- +game: Game
 - A játék maga innen lesz elérhető, hiszen itt folynak a játékbeli események.
- - chestButton : JButton
 - erre kattintva előjönnek a szobában lévő tárgyak
- - characterLabels : ArrayList<JLabel>
 - karaktereket ábrázoló ikon
- - transistorButtons : ArrayList<JButton>
 - erre kattintva lehet teleportálni másik szobába

Metódusok

11.3.5 UserPanel

- **Felelősség**

Az osztály reprezentálja a jelenlegi játékos tárgyait, és az azokhoz tartozó akciókat (használat, eldobás), illetve egy gombot, amellyel átmozoghat egy másik szobába.

- **Ősosztályok**

JPanel

- **Interfészek**

- **Attribútumok**

- - hallgato: Hallgato
 - A jelenlegi játékosost tárolja
- - JScrollPane scrollPane
 - A lista görgethetőségének biztosításáért felel
- - targyLista: JList
 - A karakterhez tartozó tárgyakat jeleníti meg
- - useButton: JButton
 - Az előbb leírt listából kiválasztott tárgy használatát valósítja meg
- - throwButton: JButton
 - Az előbb leírt listából kiválasztott tárgy eldobását valósítja meg
- - roomButton: JButton
 - A játékos másik szobába mozgásához való gombot valósítja meg

- **Metódusok**

11.3.6 Menupanel

- **Felelősség**

Felelőssége, hogy megjelenítse a játékban eltelt időt, az éppen aktív játékos nevét, valamint a jelenlegi szobát. Emellett tartalmaz egy gombot, amellyel a felhasználók be tudják zárni a játékot. Az osztály biztosítja, hogy a játékosok folyamatosan tájékoztatást kapjanak a játék állapotáról, és lehetőségük legyen az alkalmazás gyors bezárására

- **Ősosztályok**

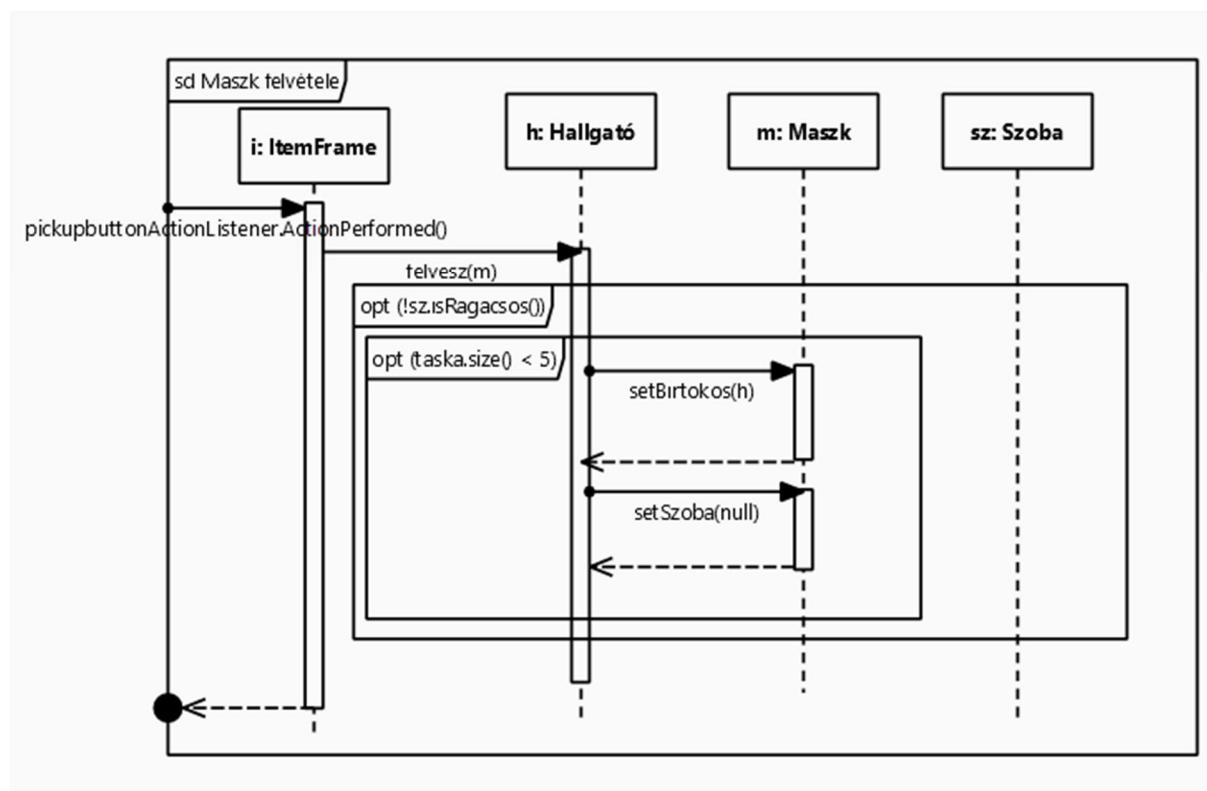
JPanel

- **Interfész**

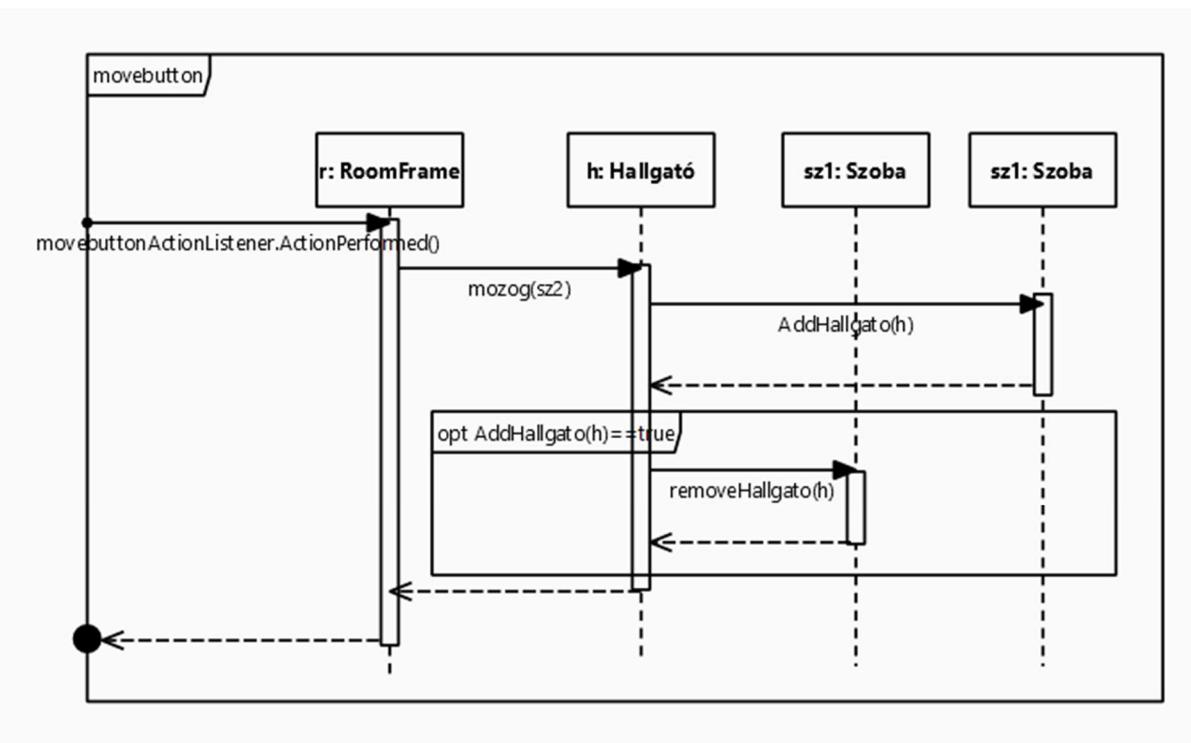
- **Attribútumok**
- **timeLabel: JLabel:** Jelzi a játékban eltelt időt. Láthatósága **private**, típusa **JLabel**.
- **playerLabel: JLabel:** Megjeleníti az éppen aktív játékos nevét. Láthatósága **private**, típusa **JLabel**.
- **roomLabel: JLabel:** Megjeleníti az aktuális szoba nevét. Láthatósága **private**, típusa **JLabel**.
- **Metódusok**

11.4 Kapcsolat az alkalmazói rendszerrel

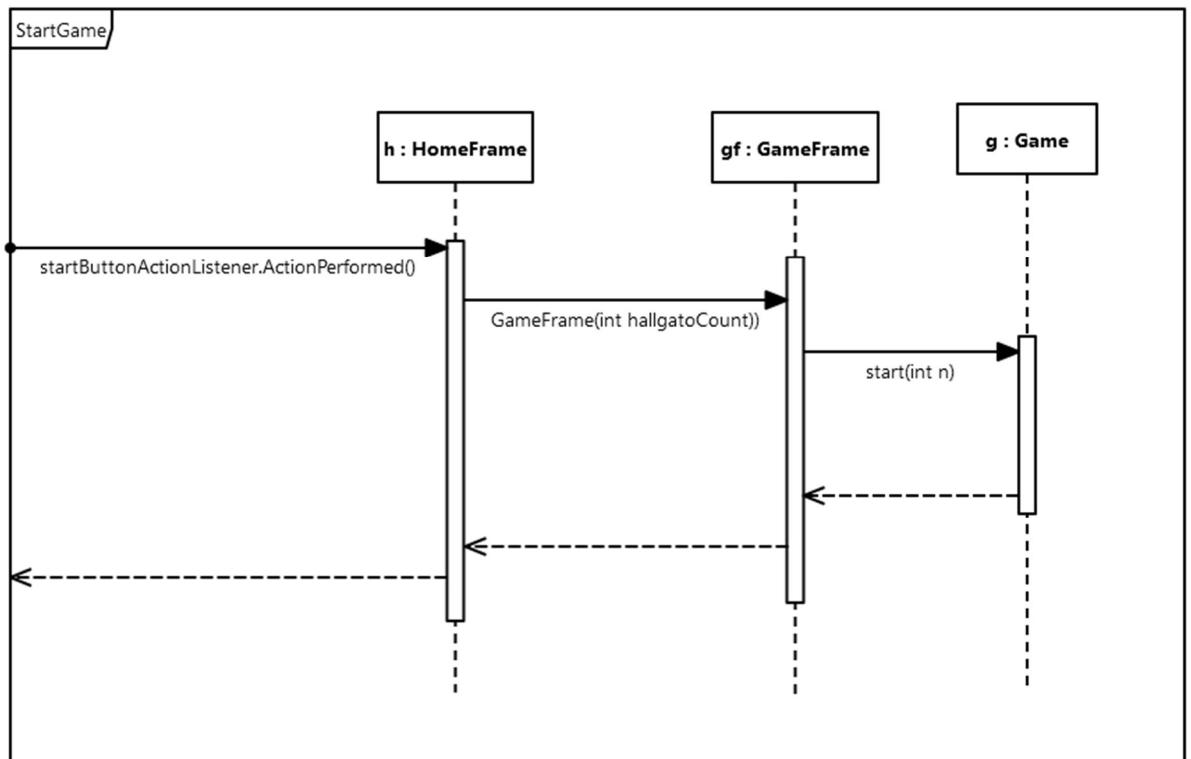
11.4.1 ItemFrame pickupbutton



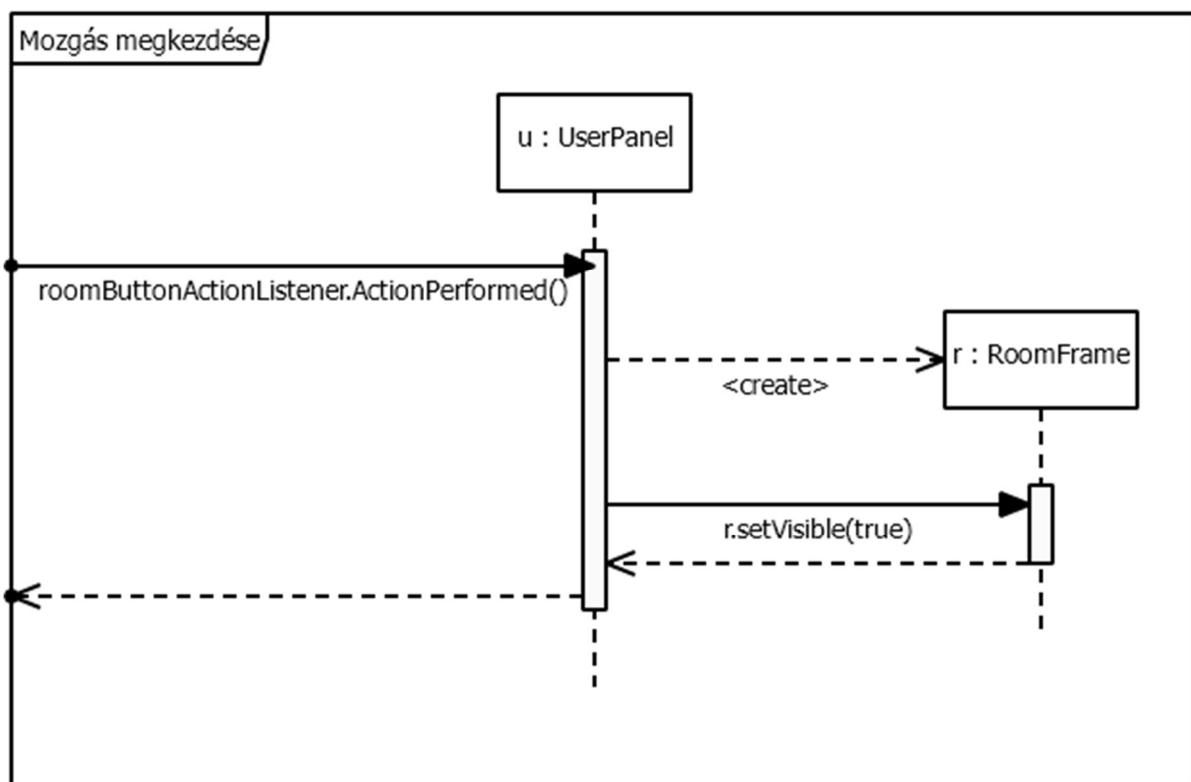
11.4.2 RoomFrame movebutton



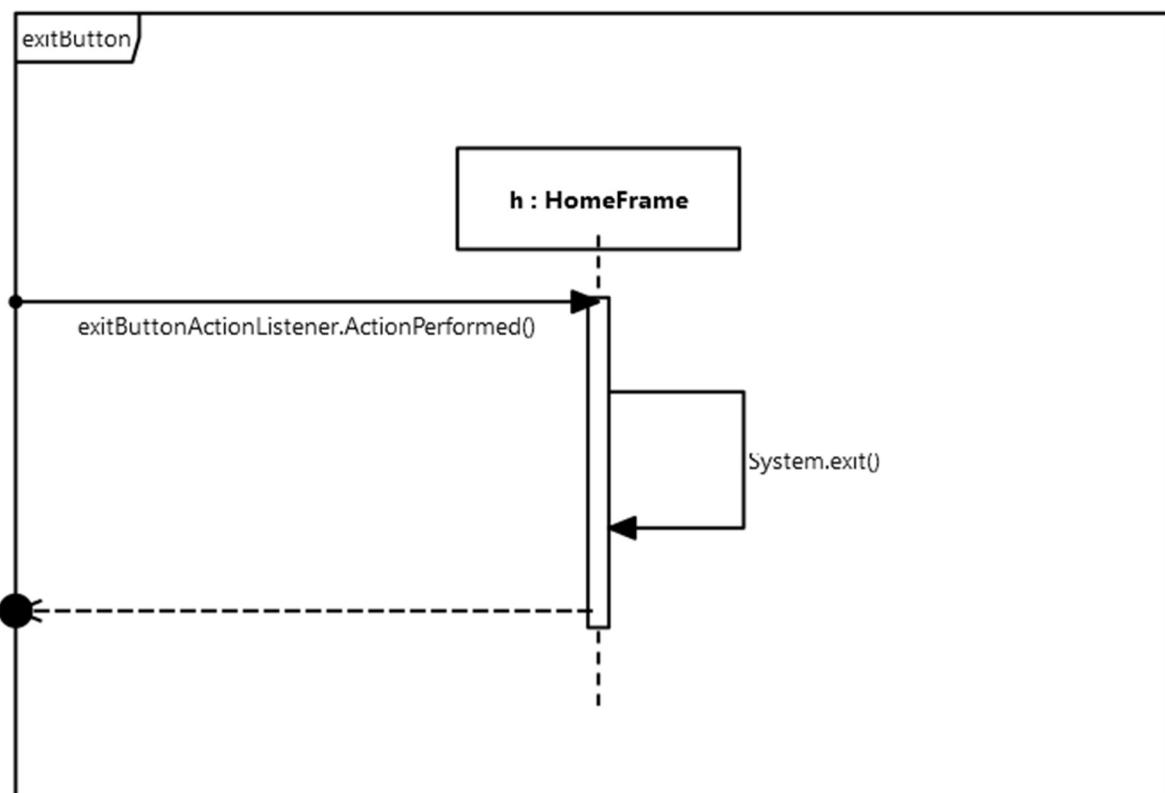
11.4.3 HomeFrame StartButton



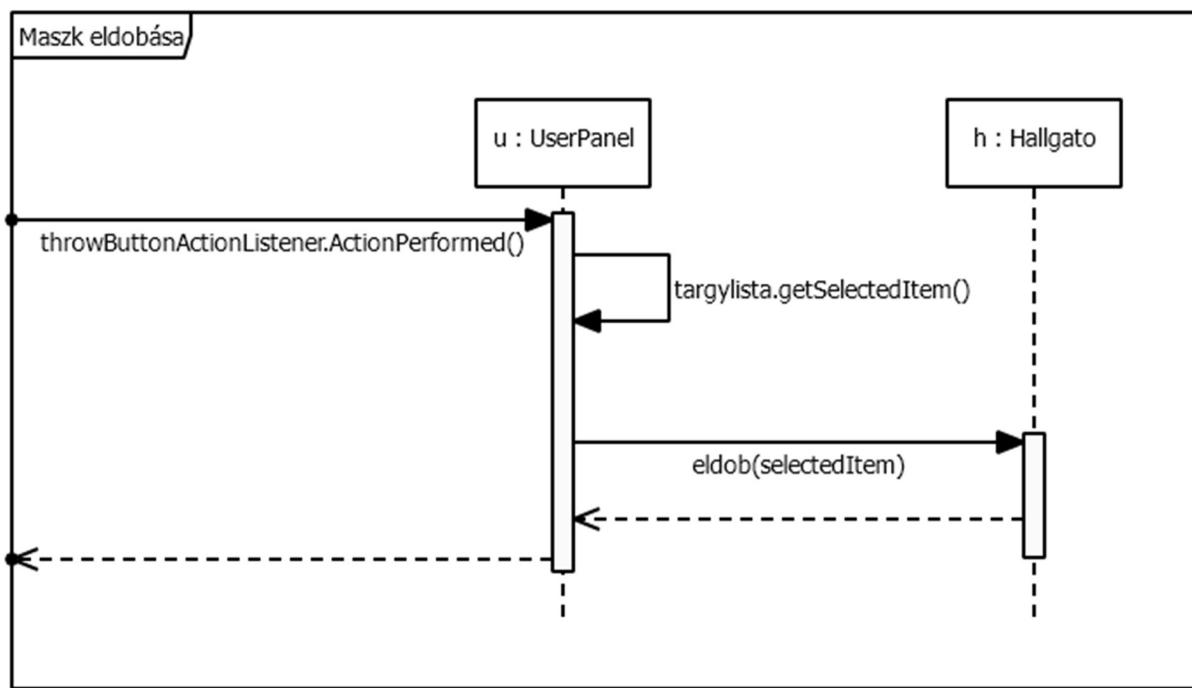
11.4.4 UserPanel roomButton



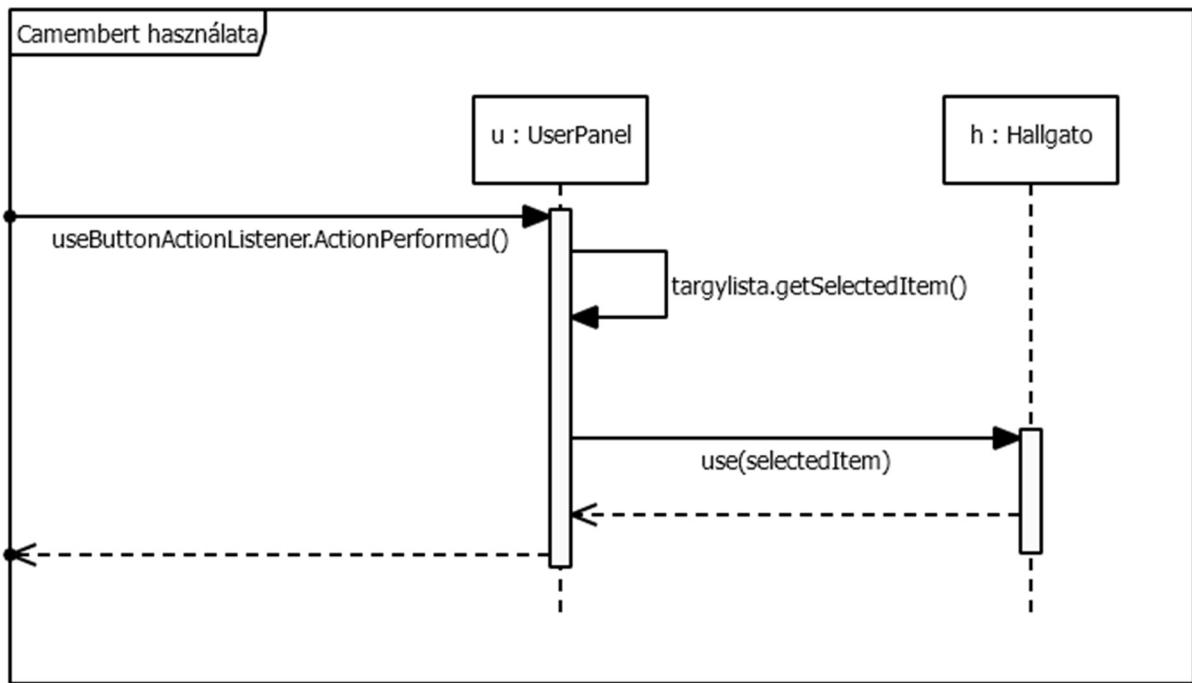
11.4.5 HomeFrame exitButton



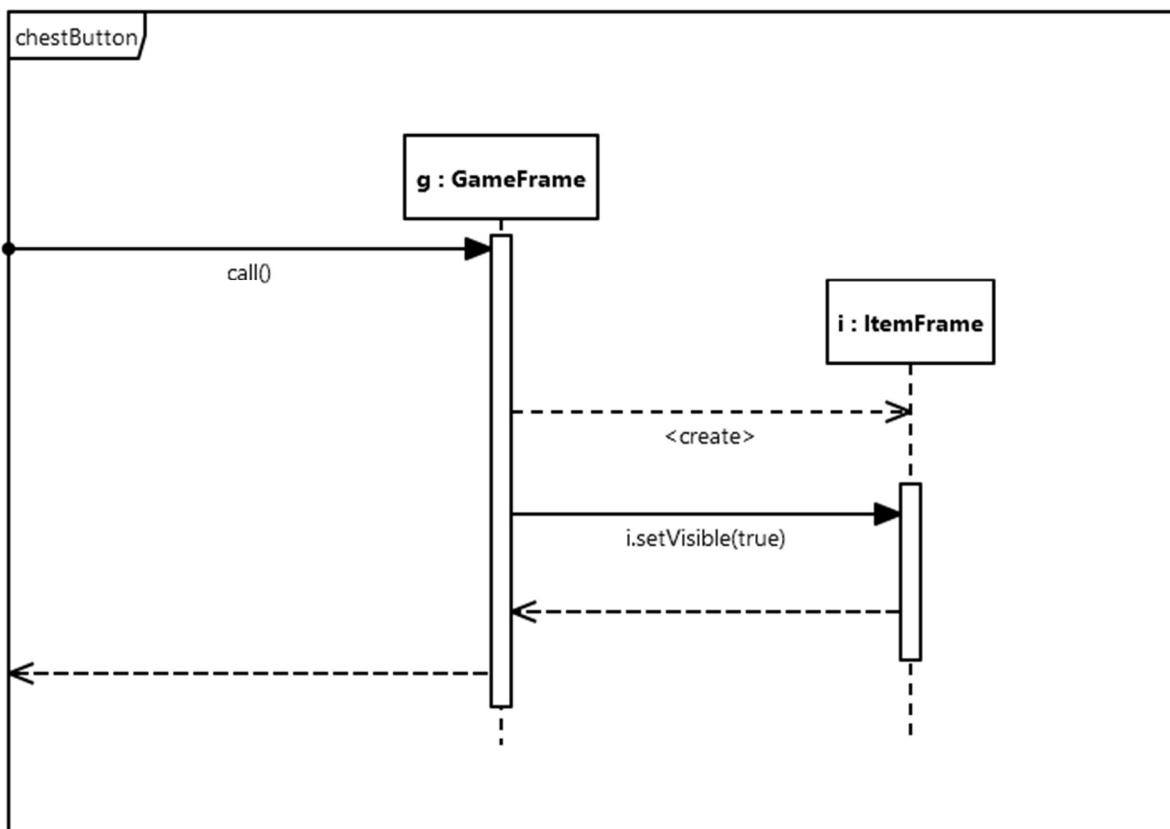
11.4.6 UserPanel throwButton



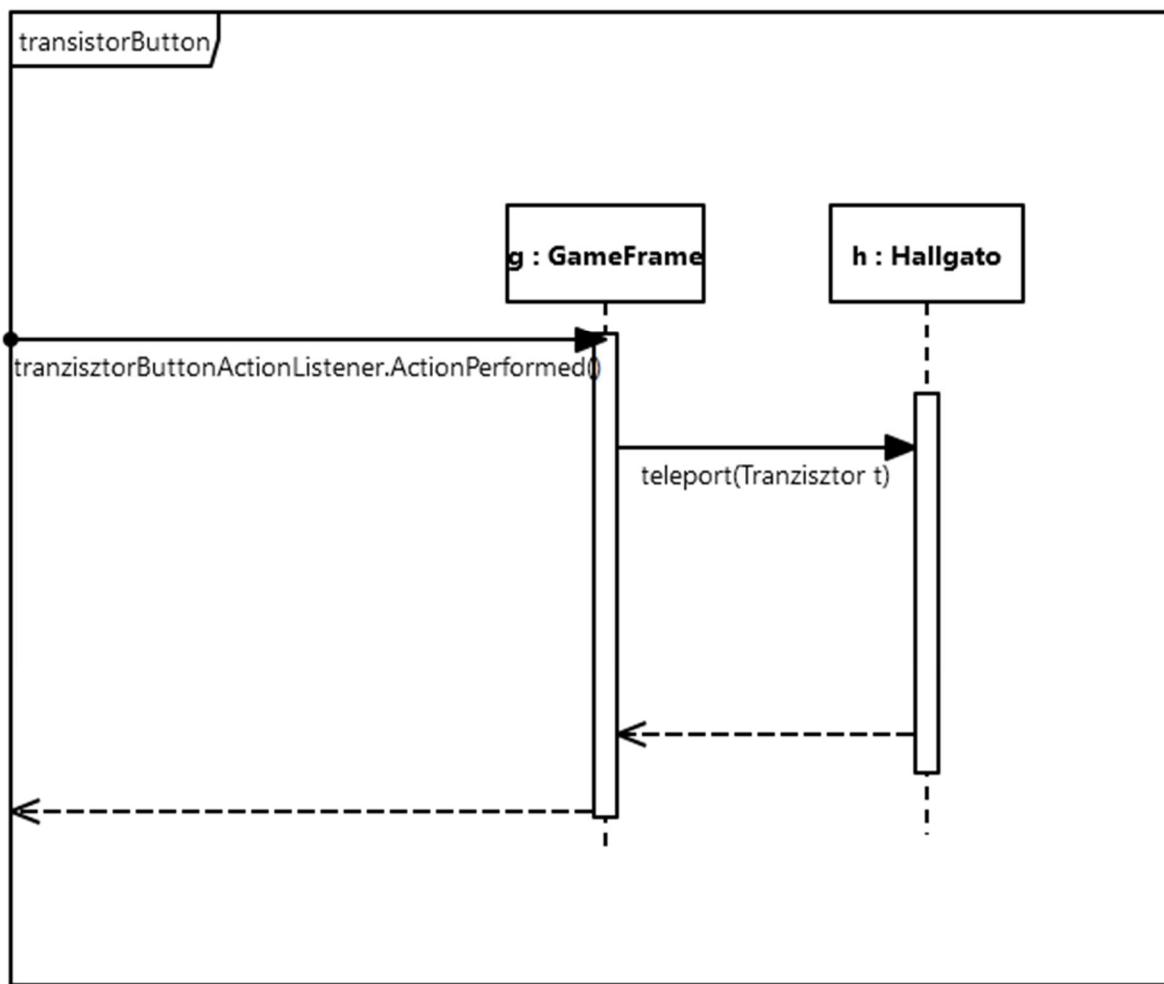
11.4.7 UserPanel useButton



11.4.8 GameFrame chestButton



11.4.9 GameFrame transistorButton



11.5 Napló

Kezdet	Időtartam	Résztvevők	Leírás
2024.05.02. 15:00	1,5 óra	Csapat	Értekezlet. Döntés: Grafikus felület kitalálása, részfeladatok kiosztása
2024.05.05. 18:00	1,5 óra	Czotter	11.4.1 és 11.4.2 szekvencia diagrammok, és ItemFrame és RoomFrame osztályok elkészítése.
2024.05.05. 22:00	1,5 óra	Németh	Grafikus felület elemei
2024.05.05. 19:00	1 óra	Schulcz	Menupanel
2024.05.05 18:00	1,5 óra	Tarsoly	UserPanel
2024.05.06. 10:00	1,5 óra	Hermann	GameFrame, HomeFrame, dokumentum megszerkesztése.

Grafikus változat beadása

35 – pupakok

Konzulens:

Koczó Attila

Csapattagok:

Tarsoly Levente	E1IK75	tarsolyl@edu.bme.hu
Németh Gergely	TA9G0R	gergo.h.nemeth@gmail.com
Schulcz Gábor	EP3I9R	schulcz.gabor6@gmail.com
Czotter Benedek	TFB4FY	czottibeni@gmail.com
Hermann Máté Béla	T83K5I	mate3829@gmail.com

2024.05.22

13. Grafikus változat beadása

13.1 Fordítási és futtatási útmutató

13.1.1 Fájllista

Fájl neve	Méret	Keletkezés ideje	Tartalom
Game.java	5 kb	2024.04.17.	Game implementációja.
Hallgato.java	4 kb	2024.04.17.	Hallgato implementációja.
Karakter.java	4 kb	2024.04.17.	Karakter implementációja.
Oktato.java	3 kb	2024.04.17.	Oktato implementációja.
Takarito.java	2 kb	2024.04.17.	Takarito implementációja.
Vedettseg.java	1 kb	2024.04.17.	Vedettseg implementációja.
Palya.java	12 kb	2024.04.17.	Palya implementációja.
ElatkozottSzoba.java	2 kb	2024.04.17.	ElatkozottSzoba implementációja.
Szoba.java	13 kb	2024.04.17.	Szoba implementációja.
AktivTargy.java	1 kb	2024.04.17.	AktivTargy implementációja.
Camembert.java	1 kb	2024.04.17.	Camembert implementációja.
Funkcio.java	1 kb	2024.04.17.	Funkcio implementációja.
Legfrissito.java	1 kb	2024.04.17.	Legfrissito implementációja.

Logarlec.java	2 kb	2024.04.17.	Logarlec implementációja.
LogarlecPropertyChangeListener.java	1 kb	2024.04.17.	LogarlecPropertyChangeListener implementációja.
Maszk.java	1 kb	2024.04.17.	Maszk implementációja.
PasszivTargy.java	1 kb	2024.04.17.	PasszivTargy implementációja.
Rongy.java	2 kb	2024.04.17.	Rongy implementációja.
Sorospohar.java	3 kb	2024.04.17.	Sorospohar implementációja.
Targy.java	3 kb	2024.04.17.	Targy implementációja.
Tranzisztor.java	3 kb	2024.04.17.	Tranzisztor implementációja.
TVSZ.java	1 kb	2024.04.17.	TVSZ implementációja.
Main.java	2 kb	2024.04.17.	Main implementációja.
GameFrame	4 kb	2024.05.17.	Ez a játéktér ablaka.
GamePanel	6 kb	2024.05.17.	Ezen történik a játék
HomeFrame	4 kb	2024.05.17.	Kezdőoldal, innen indul a játék.
ItemFrame	5 kb	2024.05.17.	Ez akkor jön elő amikor megnyitjuk a lánát és ezen vannak a szobában felvehető tárgyak.

MenuPanel	3 kb	2024.05.17.	Idő, hallgató és szoba mutatójának panelja.
RoomFrame	3 kb	2024.05.17.	Ez a frame felel a szoba választásáért.
UserPanel	8 kb	2024.05.17.	Ezen a panelen tud a hallgató dolgokat végrehajtani.
CamembertViewModel	2 kb	2024.05.17.	Camembertet köti össze a modellel.
GameViewModel	5 kb	2024.05.17.	Játékot köti össze a modellel.
HallgatoViewModel	6 kb	2024.05.17.	Hallgatót köti össze a modellel.
ItemViewModel	1 kb	2024.05.17.	Tárgyat köti össze a modellel.
LegfrissitoViewModel	2 kb	2024.05.17.	Légfrissítőt köti össze a modellel.
LogarlecViewModel	1 kb	2024.05.17.	Logarlécet köti össze a modellel.
MapViewModel	2 kb	2024.05.17.	Pályát köti össze a modellel.
MaszkViewModel	1 kb	2024.05.17.	Maszkot köti össze a modellel.
RongyViewModel	1 kb	2024.05.17.	Rongyot köti össze a modellel.

SorospoharViewModel	1 kb	2024.05.17.	Söröspoharat köti össze a modellel.
SzobaViewModel	8 kb	2024.05.17.	Szobát köti össze a modellel.
TranzisztorViewModel	2 kb	2024.05.17.	Tranzsisztort köti össze a modellel.
TVSZViewModel	1 kb	2024.05.17.	TVSZ-t köti össze a modellel.
Szoba_attributumok.txt	1 kb	2024.05.17	A pálya szobáinak attribútumait tárolja
Szoba_szomszedok.txt	1 kb	2024.05.17	A pálya szobáinak a szomszédait tárolja
camembert.png	297 kb	2024.05.17.	A Camembert sajtot reprezentáló kép
chest.png	81 kb	2024.05.17.	A szobában levő tárgyakhoz tartozó lánát reprezentáló kép
janitor.png	34 kb	2024.05.17.	A takarító karaktert reprezentáló kép
legfrissito.png	63 kb	2024.05.17.	A légfrissítő tárgyat reprezentáló kép
logarlec.png	136 kb	2024.05.17.	A logarléc tárgyat reprezentáló kép
maszk.png	129 kb	2024.05.17.	A maszk tárgyat reprezentáló kép

move.png	2 kb	2024.05.17.	A mozgáshoz tartozó gombot reprezentáló kép
rongy.png	42 kb	2024.05.17.	A rongy tárgyat reprezentáló kép
sorospohar.png	73 kb	2024.05.17.	A Szent söröspohár tárgyat reprezentáló kép
student.png	27 kb	2024.05.17.	A hallgató karaktert reprezentáló kép
teacher.png	33 kb	2024.05.17.	Az oktató karaktert reprezentáló kép
tranzisztor.png	65 kb	2024.05.17.	Az tranzisztor tárgyat reprezentáló kép
tvsz.png	387 kb	2024.05.17.	Az TVSZ tárgyat reprezentáló kép

13.1.2 Fordítás és telepítés

Lépések:

- Nyissuk meg a forrásfájlokat tartalmazó könyvtárat cmdben
- Futtassuk a “javac Main.java” parancsot

13.1.3 Futtatás

- Futtassuk a “java Main” parancs segítségével a programot

13.2 Értékelés

Tag neve	Tag neptun	Munka százalékban
Czotter Benedek	TFB4FY	21,5
Tarsoly Levente	E1IK75	22,5
Schulcz Gábor	EP3I9R	21
Németh Gergő	TA9G0R	17,5
Hermann Máté Béla	T83K5I	17,5

13.3 Napló

Kezdet	Időtartam	Résznevők	Leírás
2024.05.08. 20:30	2,5 óra	Csapat	Értekezlet: Döntések: Előző feladat hibáinak megbeszélése és javítása. Feladatok kiosztása.
2024.05.12. 20:00	2 óra	Csapat	Értekezlet: Döntések: Hogyan haladunk a kiosztott feladatokkal. Felbukkanó problémák kezelése.

2024.05.17. 10:00	3 óra	Tarsoly	HallgatoViewModel, UserPanel elkészítése, gombok bekötése
2024.05.17. 18:00	3 óra	Hermann	GameFrame, HomeFrame, GameViewModel elkészítése, gombok bekötése
2024.05.18. 10:00	2 óra	Németh	ItemViewModelek elkészítése
2024.05.20. 10:00	10 óra	Schulcz	Grafikus változat készítése.
2024.05.21. 10:00	7 óra	Németh	Grafikus vázlat készítése
2024.05.20. 10:00	10 óra	Tarsoly	Grafikus változat készítése.
2024.05.20. 10:00	10 óra	Czotter	Grafikus változat készítése.
2024.05.20. 10:00	8 óra	Hermann	Grafikus változat készítése.
2024.05.21. 17:00	8,5 óra	Czotter	Grafikus változat készítése.
2024.05.21. 18:00	2,5 óra	Schulcz	Grafikus változat készítése.
2024.05.21. 20:30	5,5 óra	Csapat	Értekezlet: Döntések: Grafikus változat befejezése.

14. Összefoglalás

14.1 A projektre fordított összes munkaidő

Tag neve	Munkaidő (óra)
Tarsoly Levente	98,5 óra
Czotter Benedek	97 óra
Schulcz Gábor	89 óra
Hermann Máté Béla	84,5 óra
Németh Gergő	82 óra
Összesen	451 óra

- A feltöltött programok forrássorainak száma**

Fázis	Kódsorok száma
Szkeleton	2139
Prototípus	2905
Grafikus változat	4326
Összesen	5805

14.2 • Projekt összegzés

14.2.1 Mit tanultak a projektből konkrétan és általában?

Czotter Benedek: csapatban dolgozni, egy az eddigieknel bonyolultabb és nagyobb fejlesztési folyamatot végigvinni

Schulcz Gábor: emberekkel megfelelően kommunikálni, kompromisszumokkal dolgozni, felelősséget vállalni

Németh Gergő: Egy nagyobb kaliberű feladatot úgy elkészíteni, hogy nem teljesen én osztom be, hogy mit és hogyan csinálok, hanem csapatban, összedolgozva megvalósítani azt. Ezek mellett a saját munkámért felelősséget vállalni úgy, hogy annak eredménye nem csak engem befolyásol, hanem akár a csapat összes tagját is.

Tarsoly Levente: Egy projekt elkészítését úgy, hogy fejlesztés előtt kell megtervezni azt, és ennek a nehézségeit.

Hermann Máté Béla: Szerintem a legnagyobb tanulságai a projektnek a csapatmunka és mindenkinél az időtervének egybehangolása, valamint az, hogy előre átgondoljuk a projektet voltak.

14.2.2 Mi volt a legnehezebb és a legkönnyebb?

Czotter Benedek: A legnehezebb a projekt elején megtervezni, és átlátni a feladatot. A legkönnyebb a szkeleton elkészítése volt.

Schulcz Gábor: Megérteni mit akar a feladat, példák nélkül.

Németh Gergő: A legnehezebb szerintem a projekt teljes átlátása volt, főleg amikor a beadandó feladatok és maga a projekt is kezdett bonyolodni. A legkönnyebb résznek az első feladatokat tartom.

Tarsoly Levente: A legnehezebb azoknak a feladatoknak megcsinálása volt, amihez nem volt előzetesen tanult tudásunk, főleg a tervezési folyamatnál. A legkönnyebbnek a kezelői felület elkészítése tűnt, de a megvalósítás közben kiderült, hogy ez sem a legtriviálisabb feladat.

Hermann Máté Béla: Számomra a legnehezebb kihívást az időbeosztás és a feladatokra szánt idő felmérése volt.

14.2.3 Összhangban állt-e az idő és a pontszám az elvégzendő feladatokkal?

Czotter Benedek: Szerintem igen.

Schulcz Gábor: Igen, szerintem.

Németh Gergő: Igen.

Tarsoly Levente: Nagyjából igen.

Hermann Máté Béla: Szerintem igen.

14.2.4 Ha nem, akkor hol okozott ez nehézséget?

Czotter Benedek: -

Schulcz Gábor: -

Németh Gergő: -

Tarsoly Levente: Talán az analízis modell kidolgozása érhetne több pontot.

14.2.5 Milyen változtatási javaslatuk van?

Czotter Benedek: Az elvárt objektum orientált megközelítésről egy picit több segédanyag, akár Szoftvertechnológia tárgyból.

Schulcz Gábor: Szoftvertechnológia és a szoftver projekt labor tárgyak egymáshoz igazítása szerintem segítséget nyújtana a hallgatóknak.

Németh Gergő: Szerintem nagy könnyebbseg lenne, ha többet tudnának a hallgatók az előzőekben megszerzett tudásukra alapozni esetleg más tárgyakból, amik a témahez kapcsolódnak és nem út közben kéne a tudásukat is frissíteni a feladatok elvégzése mellett.

Tarsoly Levente: Ahogy mások is mondták, a feladatokhoz tartozó elvárások igazítása az elhangzott anyagokhoz szoftvertechnológia tárgyakból, illetve szerintem a prog tárgyakból az OOP alapelvekre is lehetne nagyobb hangsúlyt fektetni valamelyen formában.

Hermann Máté Béla: A többi felvetés mellett én a laborok időpontját említeném meg, hogy hamarabb is neki lehessen állni a feladatoknak, ne szerdáról kelljen hértföre.

14.2.6 Milyen feladatot ajánlanának a projektre?

Czotter Benedek: Egy hasonló mint a mostani csak walking dead témában.

Schulcz Gábor: Scotland Yard társasjáték-hoz hasonló felépítésű logikai játék szerintem jó ötlet lehetne.

Németh Gergő: Esetleg egy, a klasszikus Cluedo társasjátékra hasonlító program elkészítése.

Tarsoly Levente: Civilization szerű körökre osztott stratégiai játék

Hermann Máté Béla: Egyetértek leventével, valamilyen stratégiai játék ahol nagyobb hangsúly van a játékmeneten.

14.2.7 Egyéb kritika és javaslat

Czotter Benedek: nincs

Schulcz Gábor: Nincs.

Németh Gergő: Nincs.

Tarsoly Levente: A program fordításához legyen lehetőség más fordítóeszközöket is használni, mint pl. Maven.

Hermann Máté Béla: Nincs.