**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**

# 

## **4.3** **Osztályok leírása**

### **4.3.1 *AktívTárgy***

#### · **Felelősség**

Egy absztrakt osztály, amely az aktív tárgyakat 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 eszmeletvesztett : 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, meghívhassa 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 megbenul() : ha van megfelelő tárgy a hallgatónál és egy szobába kerülnek akkor az oktató megbénul és nem tud kibuktatni hallgatót.

### **4.3.11PasszívTárgy**

#### · **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> szobak: 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> regiszobak : 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) : eltűntet 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<Oktatok> getOktatok() : visszaadja a szobában tartózkodó oktatók listáját
* ArrayList<Hallgató> getHallgatok() : visszaadja a szobában tartózkodó hallgatók listáját
* void removeOktato() : kitöröl egy oktatót a listából
* void removeHallgató() : 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 osztállyal azért, hogy tudja, melyik karakter birtokolja ezt a tárgyat.
* Szoba osztállyal 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 settere
* void setBirtokos(Karakter): tárgy birtokosának settere
* 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**

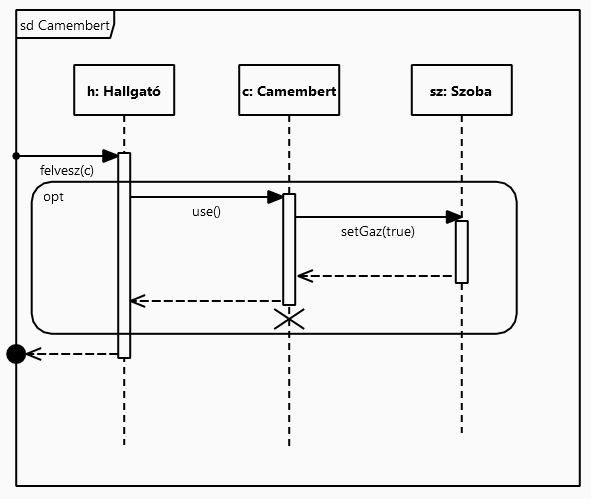
* int töltet: Ez számlálja azt, hogy az adott TVSZ hányszor használható még.

#### · **Metódusok**

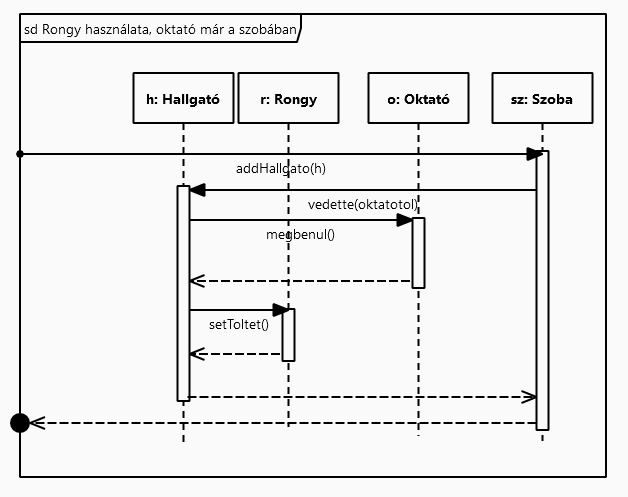
* void use(): Amikor a játékos egy szobába lép, amelyben oktató is tartózkodik, akkor használja ezt a tárgyat, ezzel a töltete eggyel csökken, viszont folytathatja a játékot.

**4.4** **Szekvencia diagramok**

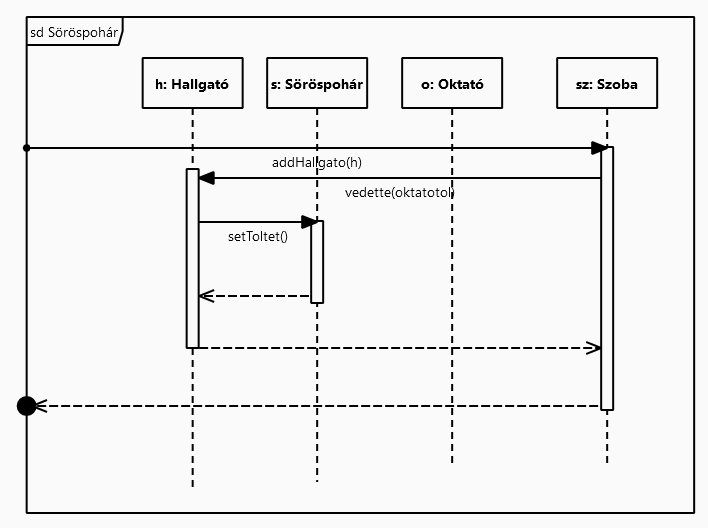
**4.4.1 Káposztás camembert használata**



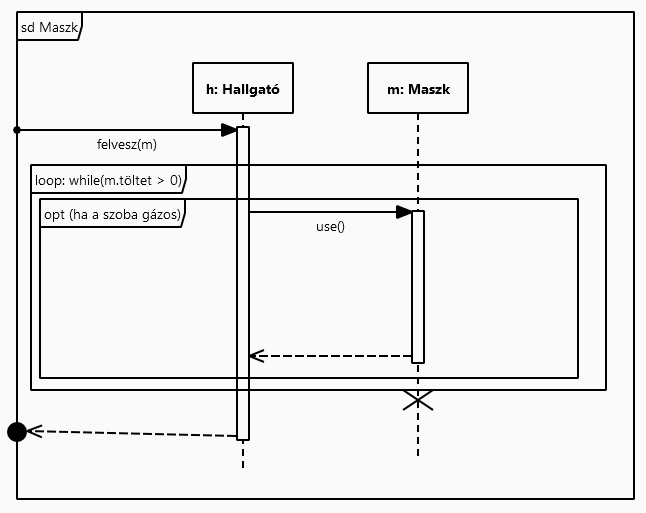
# **4.4.2 Rongy használata**



**4.4.3 Söröspohár használata**



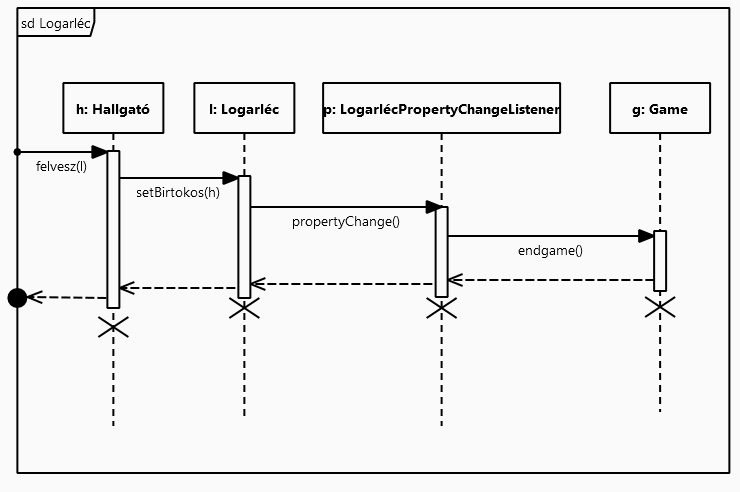
**4.4.4 FFP2-es maszk használata**



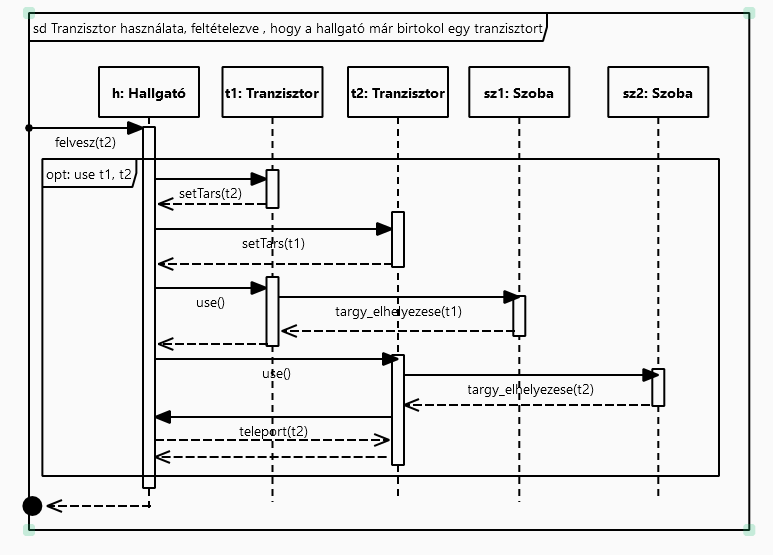
**4.4.5 TVSZ használata**

# 

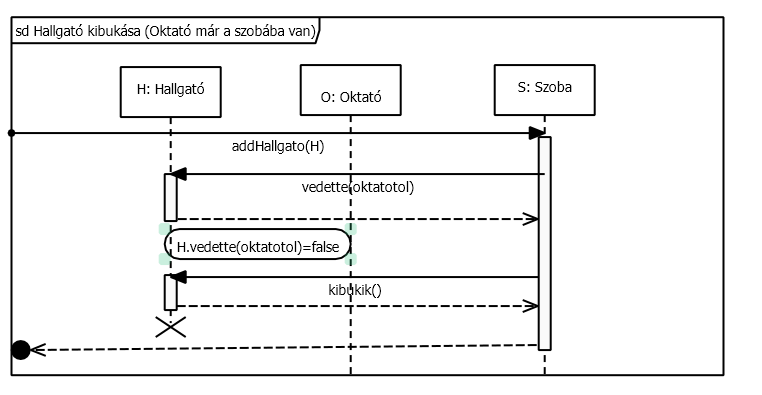
**4.4.6 Logarléc**



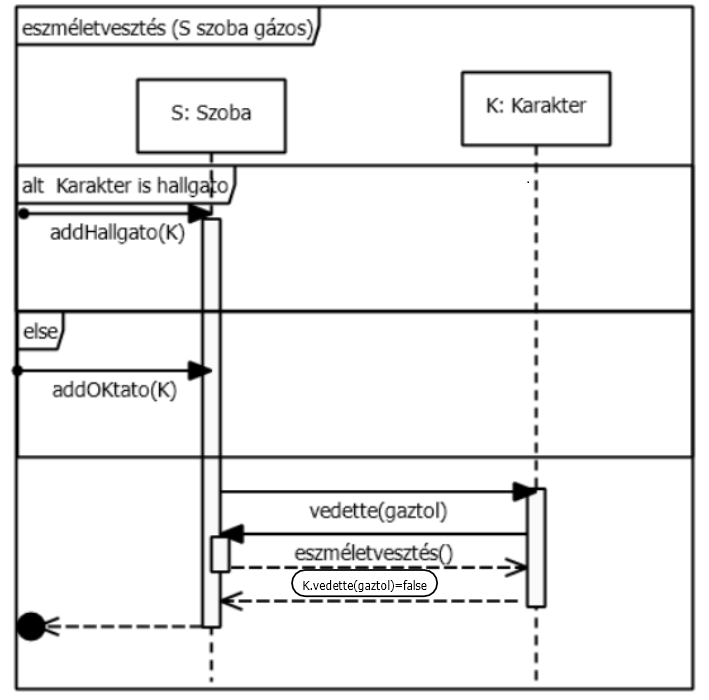
**4.4.7 Tranzisztor használata**



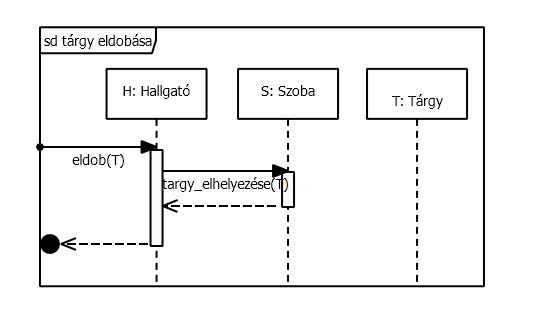
**4.4.8 Hallgató kibukása**



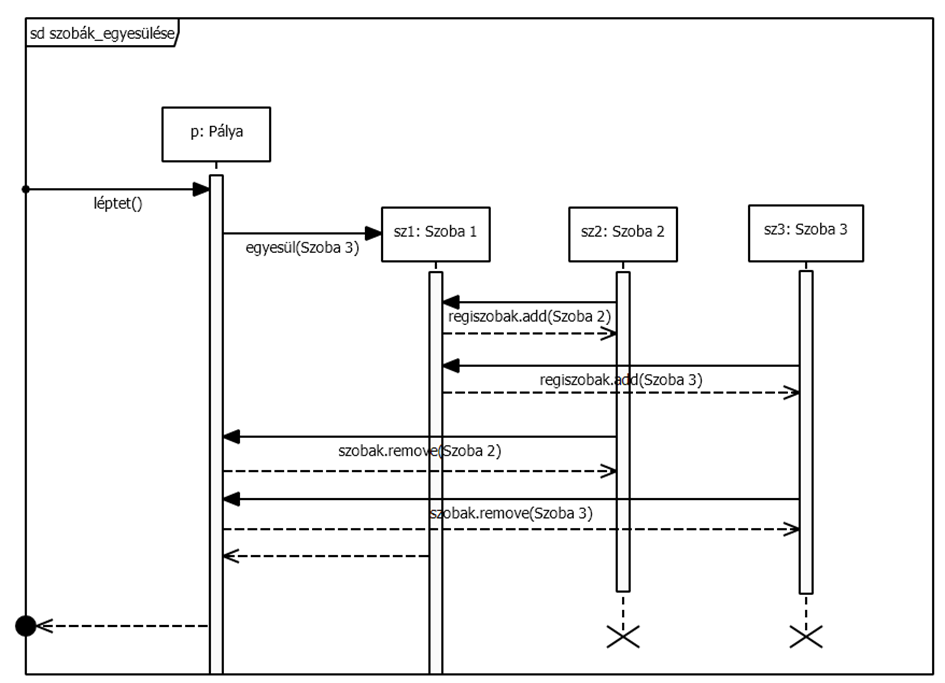
**4.4.9 Karakterek eszméletvesztése**

****

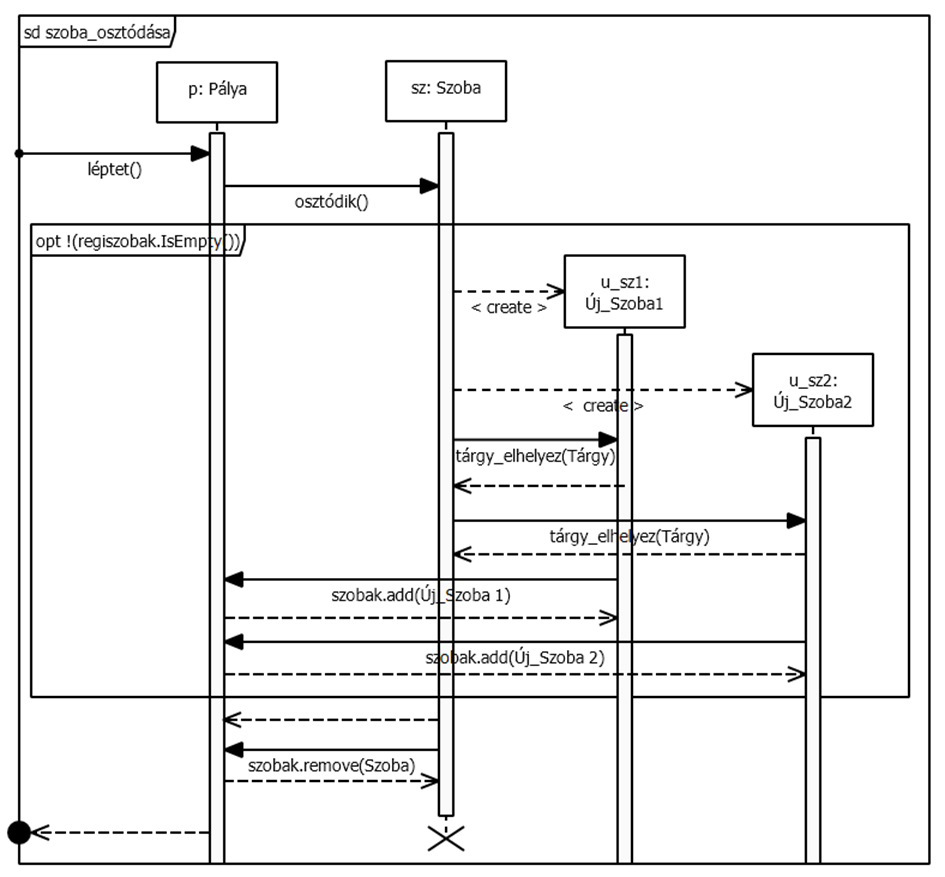
**4.4.10 Eldobás**

****

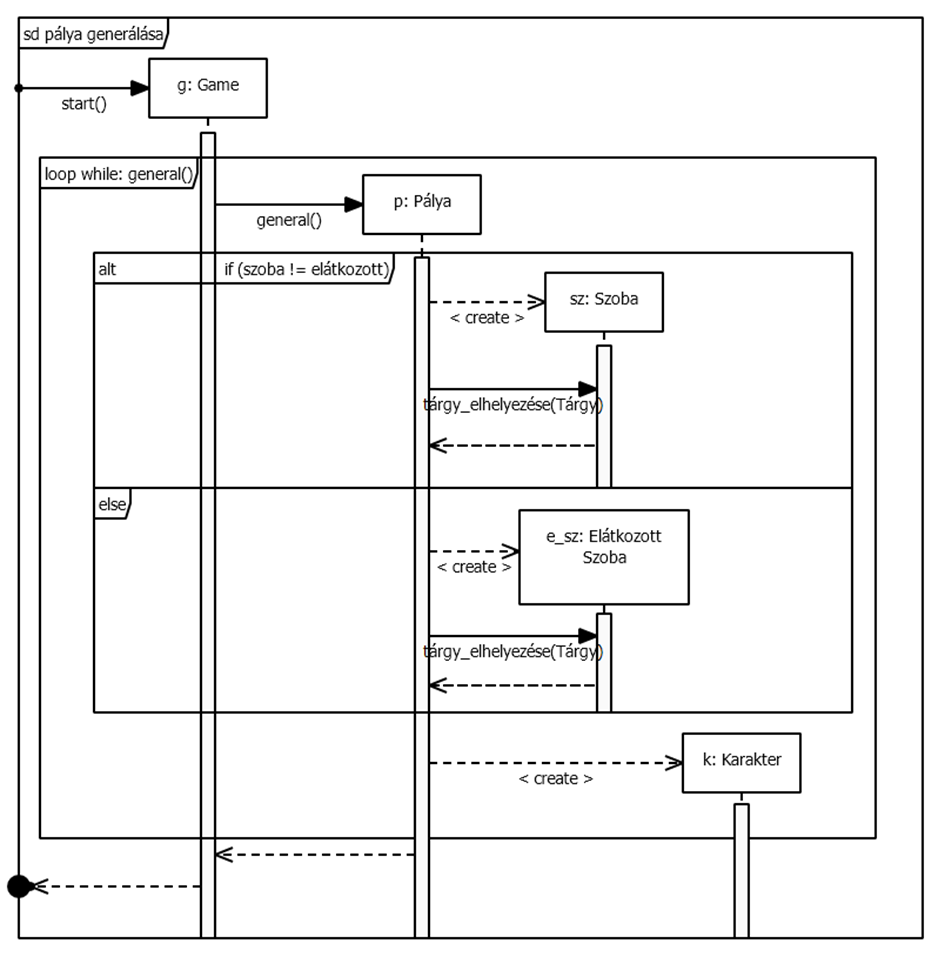
**4.4.11 Szobák egyesülése**

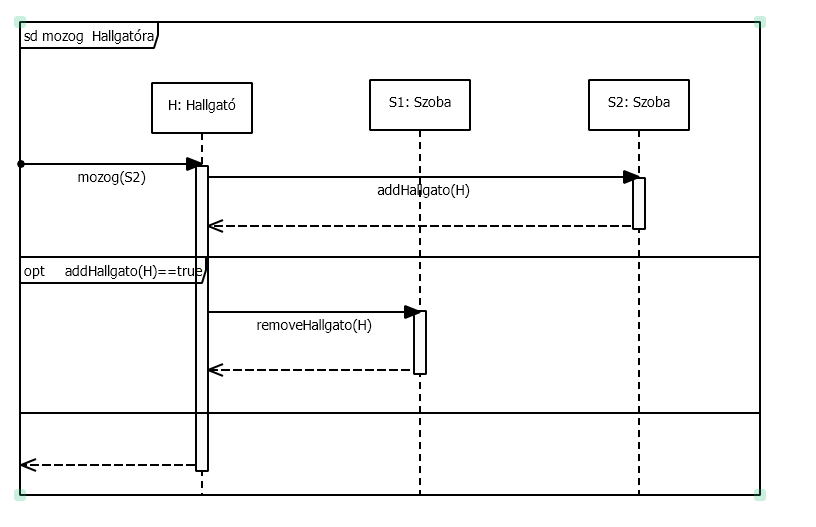


**4.4.12 Szobák osztódása**

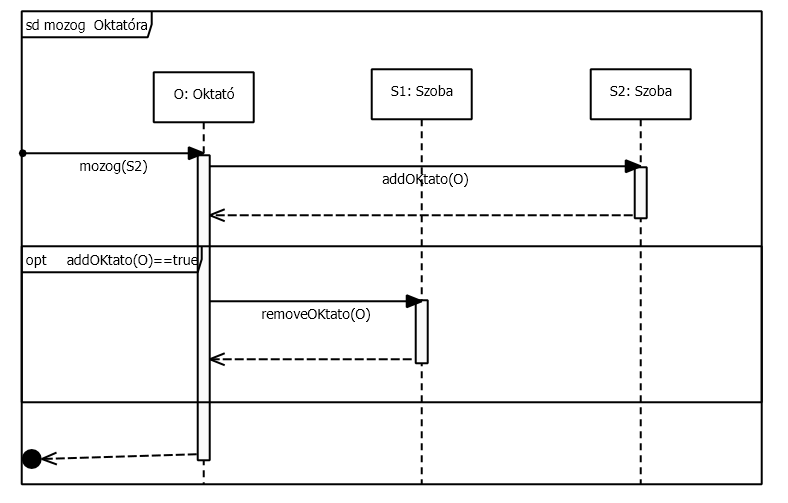


**4.4.13 Pálya generálása**



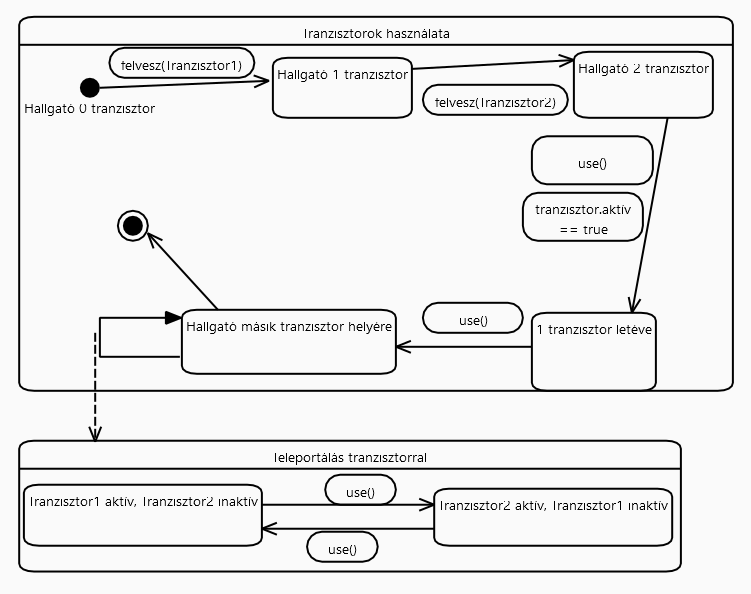
**4.4.14 Mozgás hallgatóra**

**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 |

# 