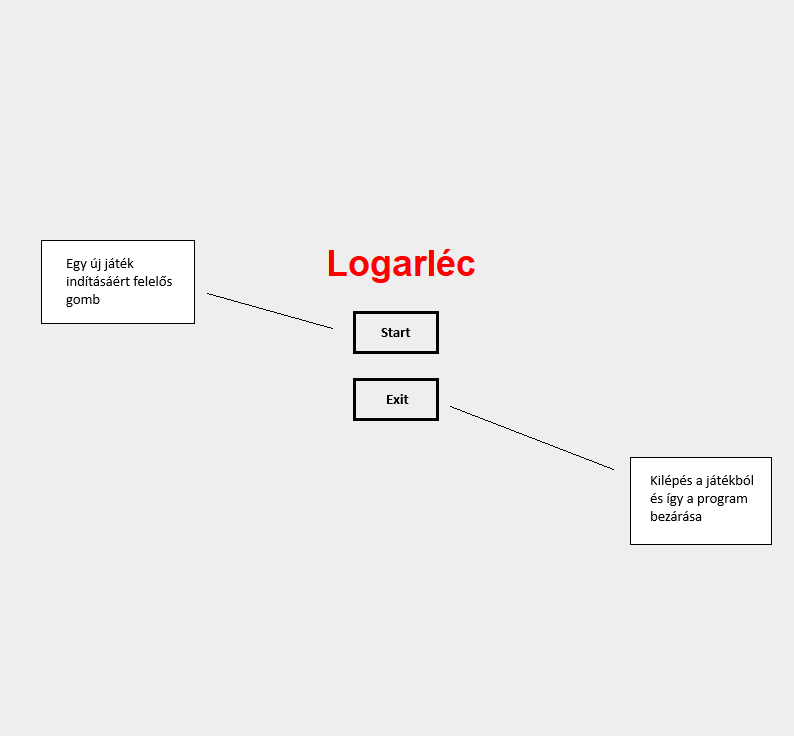
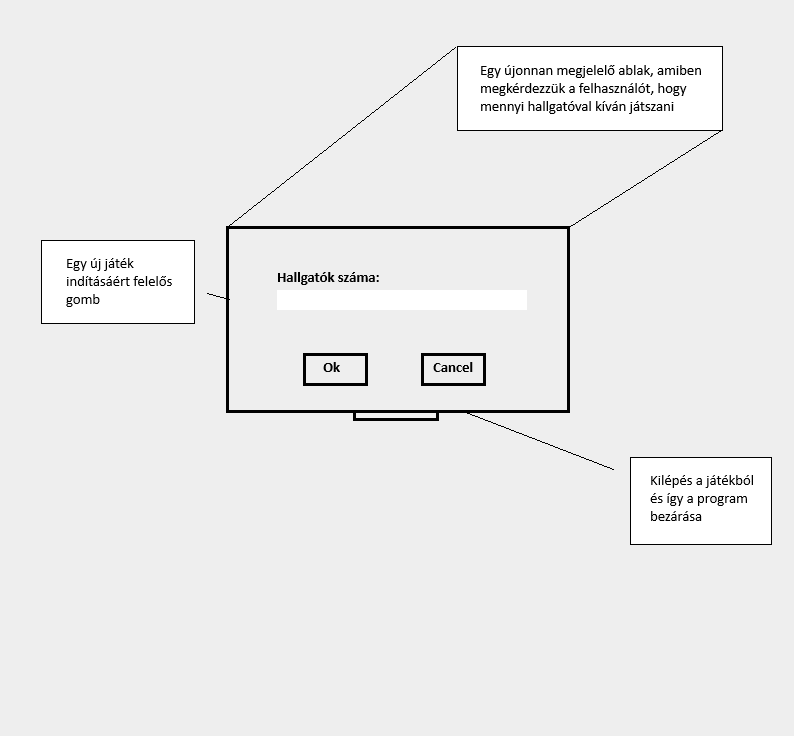
# **11.** **Grafikus felület specifikációja**

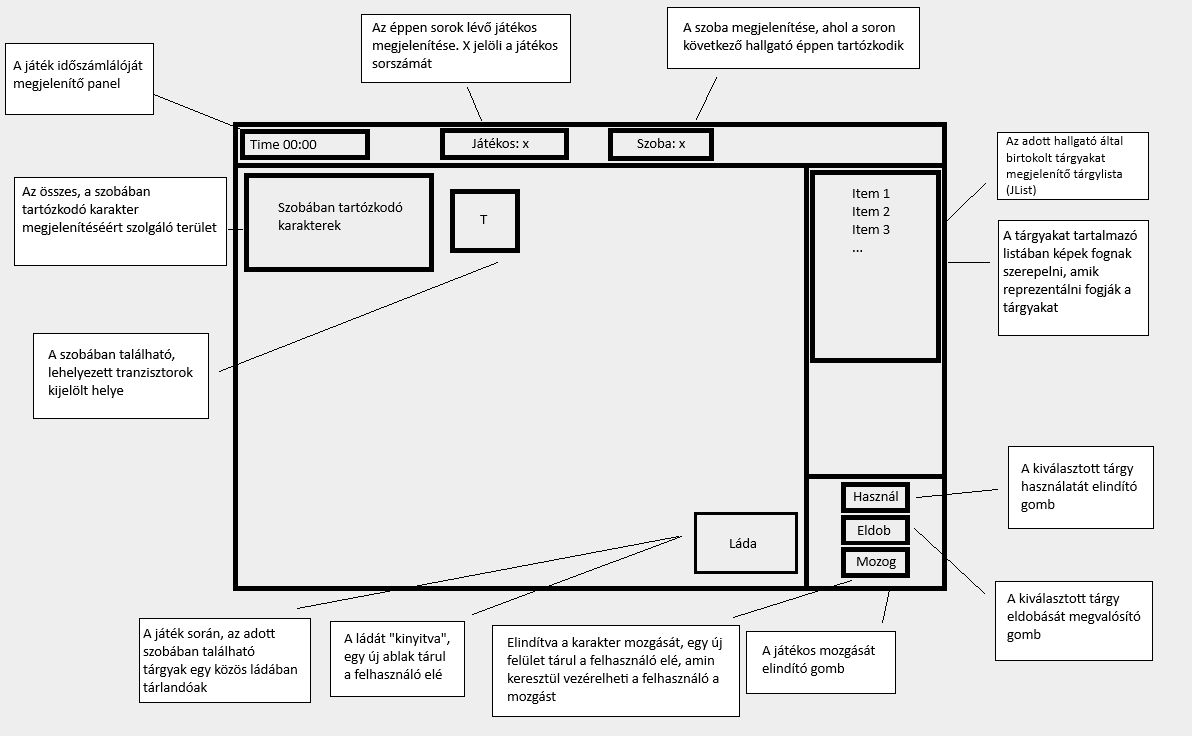
## **11.1** **A grafikus interfész**

**HomeFrame:**

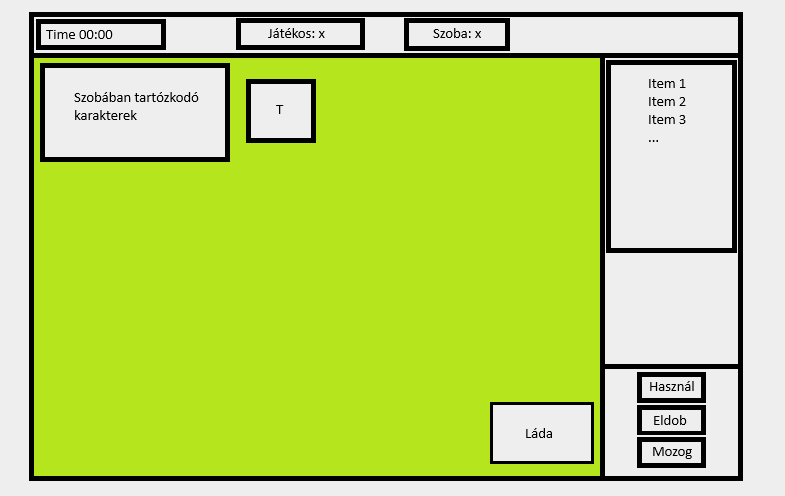




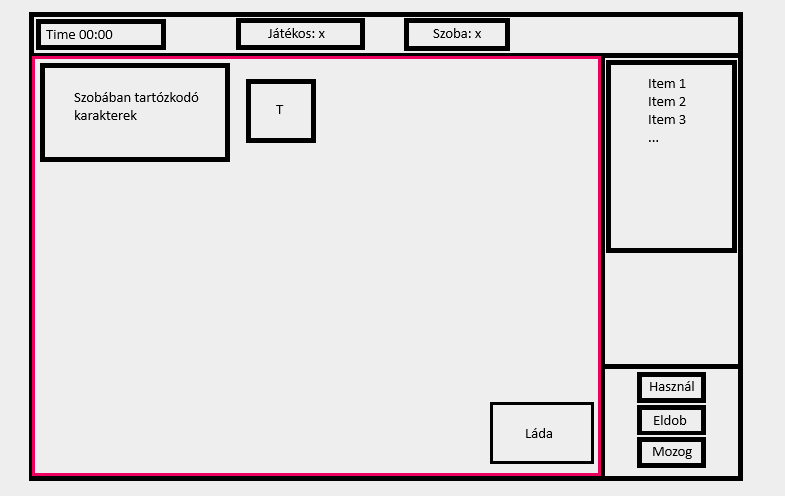
**GameFrame:**



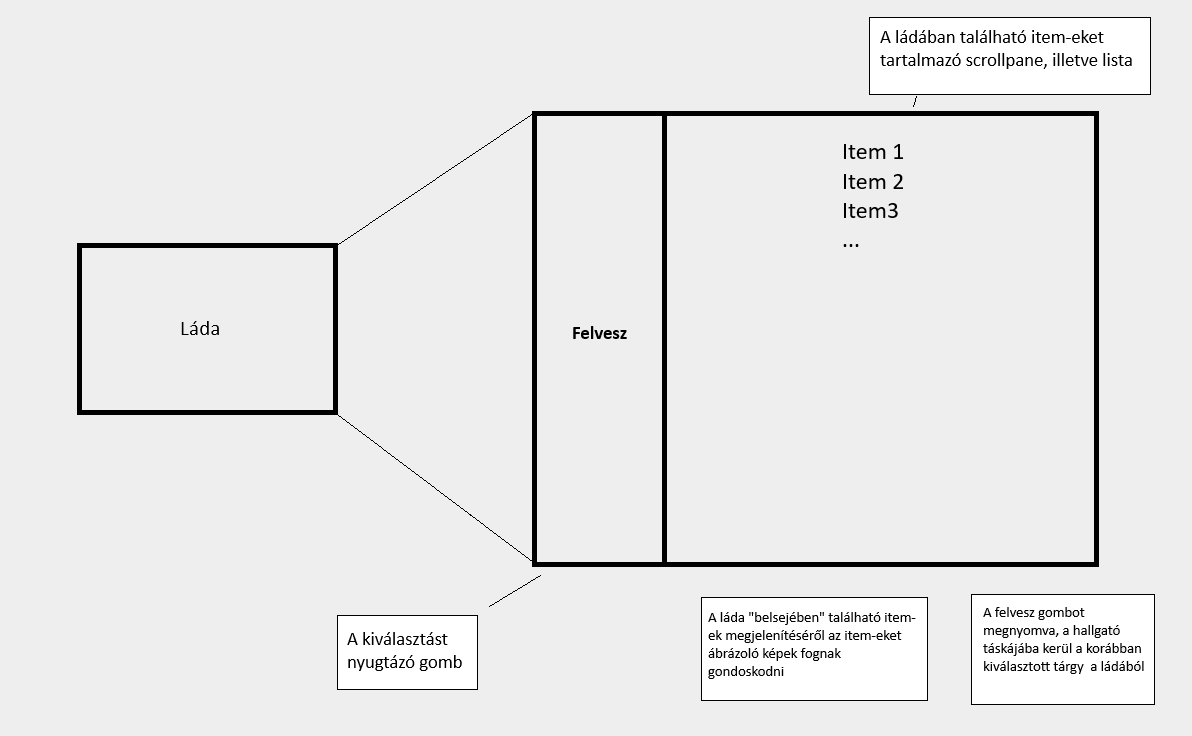
* Ha az adott szoba gázos, akkor a szobát “megtölti” a gáz, így a szoba háttere zöld-re változik.



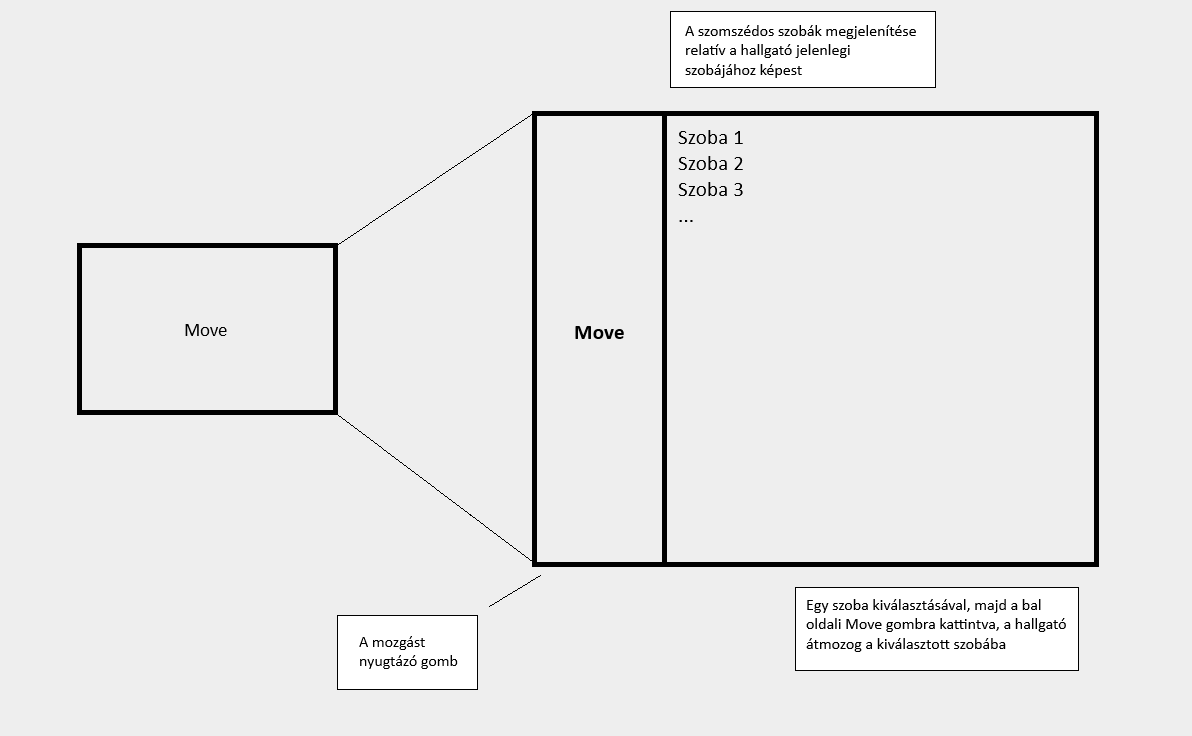
* Ha az adott szoba elátkozott, akkor a szoba kerete lilás színre változik, jelezve a szoba elátkozott voltát.



**ItemFrame:**



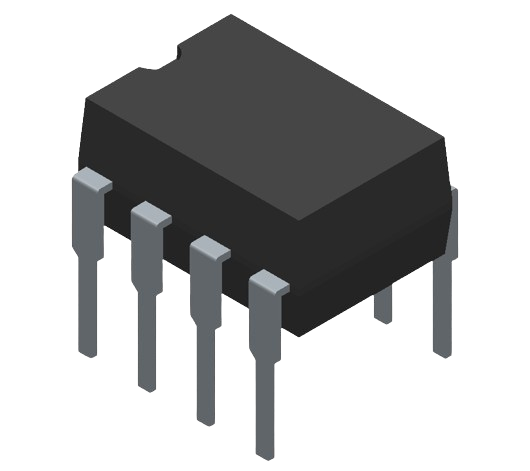
**RoomFrame:**



**Tárgyak megjelenítése grafikus felülettel:**

**** - Camembert

 - Légfrissítő

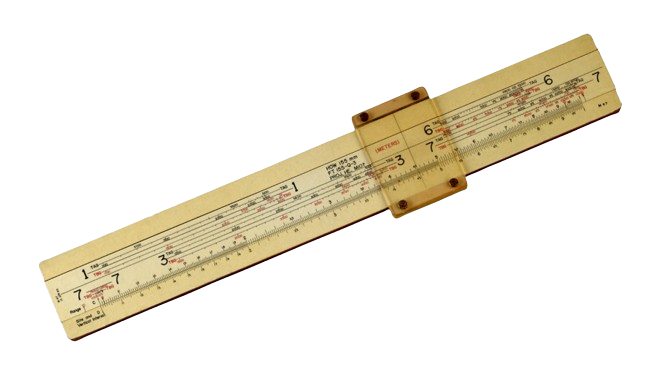
 - Tranzisztor

 - Rongy

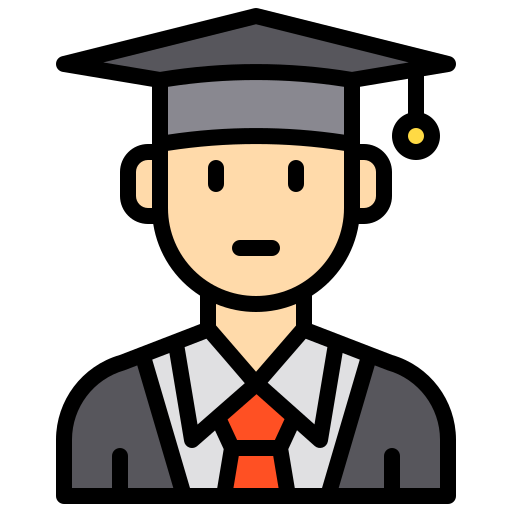
- Söröspohár

 - TVSZ

 - Maszk

 - Logarléc

 - A szobában az tárgyak tárolási helyeként szolgáló láda ikonja

 - Hallgató

 - Oktató

 - Takarító

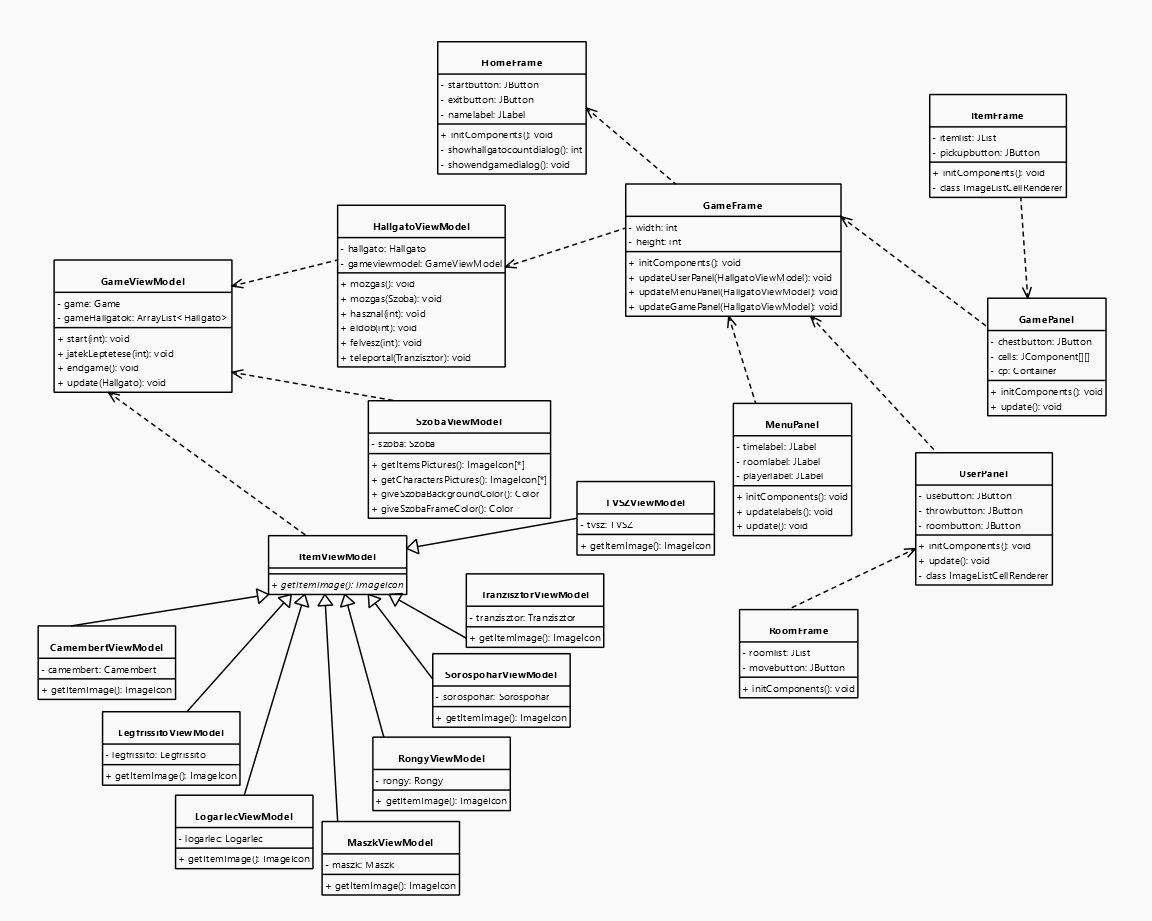
## **11.2** **A grafikus rendszer architektúrája**

### **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 alapút választottuk, tehát a felület szól a modellnek, hogy változás történt, a felület ezután pedig újrarajzolja magát annak függvényében, ahogy változott a modell állapota.

### **11.2.2** **A felület osztály-struktúrája**

[Osztálydiagram. Minden új osztály, és azon régiek, akik az újakhoz közvetlenül kapcsolódnak.]



## **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**

* -height: final int
  + A Frame magassága.
* -width: final int
  + A Frame szélessége.
* -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..

**Metódusok**

* -initComponents(HallgatoViewModel):void
  + Inicializálja és beállítja a RoomFrame komponenseit a megadott HallgatoViewModel alapján, létrehozza a movebutton actionlistener-ét.

**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**

* -height: final int
  + A Frame magassága.
* -width: final int
  + A Frame szélessége.
* -panel: JPanel
  + Az objektumok elrendezését biztosító panel.
* - scrollpane: JScrollPane
  + A lista görgethetőségének biztosításáért felel
* -itemlist: JList<ImageIcon>
  + A szobában felvehető tárgyak megjelenítését végzi.
* -pickupbutton: JButton
  + A szobában lévő tárgyak közül a kiválasztott tárgyat lehet vele felvenni a hozzákapcsolt ActionListener segítségével.

**Metódusok**

* -initComponents(SzobaViewModel, HallgatoViewModel):void
  + Inicializálja és beállítja az ItemFrame komponenseit a megadott SzobaViewModel és HallgatoViewModel alapján, létrehozza a pickupbutton actionlistener-ét.

**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**

* -height: final int
  + A Frame magassága.
* -width: final int
  + A Frame szélessége.
* -panel: JPanel
  + Az objektumok elrendezését biztosító panel.
* -exitButton: JButton
  + Kiléphetünk vele az alkalmazásból.
* -startButton: JButton
  + Ezzel indítjuk el a játékot.
* -hallagatocnt : int
  + A játékosok száma..
* nameLabel : JLabel
  + Kiírja a játék nevét.
* title: String
  + Az ablak címét tartalmazó HTML formázott sztring.

**Metódusok**

* -initComponents():void
  + Inicializálja és beállítja az HomeFrame komponenseit a, létrehozza a startButton és exitButton actionlistener-ét, létrehozza a GameViewModelt.
* -showHallgatoCountDialog():int
  + Megjeleníti a dialógusablakot, amelyen a felhasználó megadhatja a hallgatók számát és visszaadja a HomeFramenek az értékét.
* -showEndGameDialog(int):void
  + Megjeleníti a játék végét jelző üzenetet.

**11.3.4 GameFrame**

**Felelősség**

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

**Ősosztályok**

JFrame

**Interfészek**

**-**

**Attribútumok**

* -height: final int
  + A Frame magassága.
* -width: final int
  + A Frame szélessége.
* -gamePanel: GamePanel
  + Az karakterek, aktív tranzisztorok és a tárgyakat tartalmazó láda elrendezését biztosító panel.
* -menuPanel: MenuPanel
  + Az menüt biztosító panel, amely kiírja a játékba eltelt időt, aktív szobát és hallgatót.
* -userPanel: UserPanel
  + Az aktuális hallgató tárgyainak képeinek elrendezését biztosítja és az hallgató mozog gombját és a tárgyait használó, eldobó gombokat tartalmazza.

**Metódusok**

* -initComponents():void
  + Inicializálja és beállítja az GameFrame komponenseit a megadott, létrehozza és hozzáadja a különböző paneleket.
* +updateGamePanel(SzobaViewModel, HallgatoViewModel):void
  + Frissíti a GamePanel tartalmát a kapott szoba és hallgatói modell alapján.
* +updateUserPanel(HallgatoViewModel):void
  + Frissíti a UserPanel tartalmát a kapott hallgatói modell alapján..
* +updateMenuPanel(HallgatoViewModel):void
  + Frissíti a MenuPanel tartalmát a kapott hallgatói modell alapján.
* +showEszmeletvesztesDialog(HallgatoViewModel):void
  + Megjeleníti az eszméletvesztés dialógusablakot a kapott hallgatói modell alapján.
* +showKibukasDialog(HallgatoViewModel):void
  + Megjeleníti a kibukás dialógusablakot a kapott hallgatói modell alapján.

### **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**

* - hVM: HallgatoViewModel
  + A jelenlegi játékost modelljét tárolja
* - JScrollPane scrollPane
  + A lista görgethetőségének biztosításáért felel
* - targyLista: JList<ImageIcon>
  + 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**

* -initComponents():void
  + Inicializálja és beállítja a panel komponenseit és elrendezését, létrehozza a gombok actionlistenerjeit.
* +update(HallgatoViewModel):void
  + Frissíti a felhasználói felületet a megadott HallgatoViewModel alapján.
* -update(HallgatoViewModel):void
  + Frissíti a felhasználói felületet a megadott HallgatoViewModel alapján.

### **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

#### · **Interfészek**

-

#### · **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.
* updateTimer: Timer
  + Az időzítő, amely minden másodpercben frissíti az időt.
* Time: int
  + Az eltelt idő másodpercben.

#### · **Metódusok**

* -initComponents():void
  + Inicializálja a panel komponenseit és elrendezését.
* +update(HallgatoViewModel):void
  + Frissíti a játékos és a szoba labeleket a megadott HallgatoViewModel alapján.
* -updateLabels():void
  + Frissíti az idő labelt az eltelt idő alapján.
* -formatTime(int):string
  + Az időt másodpercben átalakítja "mm:ss" formátumra.

### **11.3.7** **Gamepanel**

#### · **Felelősség**

Felelőssége, hogy a játék tábláját reprezentálja. Ez a panel tartalmazza a játék különböző elemeit, mint a játékos, szoba, tárgyak és tranzisztorok, és biztosítja azok interakcióját.

· **Ősosztályok**

JPanel

#### · **Interfészek**

-

#### · **Attribútumok**

* -cp: Container
  + A panel tartalmát tartalmazó konténer.
* +GRID\_SIZE: int
  + A tábla mérete, amely 9x9 cellás rácsot jelöl.
* -cells: JComponent[][]
  + A cellák kétdimenziós tömbje, amelyek a játék elemeit tartalmazzák.
* -chestButton: JButton
  + A láda gomb, ami a tárgyak framejét hívja elő.
* -hVM: HallgatoViewModel
  + A hallgató nézetmodellje.
* -transistorButtons: ArrayList<JButton>
  + A tranzisztor gombok listája.
* -tranzisztorok: ArrayList<TranzisztorViewModel>
  + A tranzisztorok nézetmodelljeinek listája.

#### · **Metódusok**

* -initComponents():void
  + Inicializálja a panel komponenseit és elrendezését.
* +update(HallgatoViewModel):void
  + Frissíti a panel tartalmát a megadott szoba és hallgatói nézetmodell alapján.

### **11.3.8** **ImageListCellRenderer**

#### · **Felelősség**

Felelőssége, hogy egy JLabel, amely egyedi listacella megjelenítést biztosít ImageIcon elemekhez.

#### · **Ősosztályok**

JLabel

#### · **Interfészek**

ListCellRenderer<ImageIcon>

#### · **Attribútumok**

**-**

#### · **Metódusok**

* getListCellRendererComponent(JList<? extends ImageIcon>, ImageIcon, int, boolean, boolean): Component
  + Visszaad egy komponenst, amely az aktuális cella megjelenítésére szolgál a listában. Beállítja a cella ikonját és a háttér- és előtérszínt a kiválasztás állapotának megfelelően.
* showFakeItemDialog(): void
  + Figyelmeztető párbeszédablak megjelenítése, amely jelzi, hogy a kiválasztott tárgy hamis.

### **11.3.9** **GameViewModel**

#### · **Felelősség**

A GameViewModel osztály célja a játék logikájának és a nézet közötti kapcsolat biztosítása. Ez az osztály kezeli a játék állapotát, frissíti a felhasználói felületet, és irányítja a játékmenetet.

· **Ősosztályok**

-

#### · **Interfészek**

-

#### · **Attribútumok**

* - game: Game
  + A gamet tartalmazza..
* - gameHallgatok: ArrayList<Hallgato>
  + A játékban lévő hallgatók listája.
* - mapViewModel: MapViewModel
  + A térkép nézetmodellje.
* - gameFrame: GameFrame
  + A játék grafikus felhasználói felülete.

#### · **Metódusok**

* + start(int jatekosokszama): void
  + Elindítja a játékot a megadott játékosok számával.
* + endgame(): void
  + Befejezi a játékot.
* + jatekLeptetes(): void
  + Futtatja a játékmenet léptetését.
* + synchronized update(Hallgato h): void
  + Frissíti a játék keretét és paneljeit a megadott játékos állapotával és szobájával kapcsolatos információkkal.
* + kibukottehallgato(): ArrayList<Hallgato>
  + Visszaadja a kibukott hallgatók listáját.

### 

### 

### **11.3.10** **ItemViewModel**

#### · **Felelősség**

Az osztály felelős azért, hogy a tárgyak megjelenítésére használt képek a grafikus felületen valóban megjelenjenek. Ez egy absztrakt ősosztály, aminek a függvényét, később minden leszármazott megvalósítja.

#### · **Ősosztályok**

-

#### · **Interfészek**

-

#### · **Attribútumok**

-

#### · **Metódusok**

* +getItemImage():abstract ImageIcon
  + Absztrakt függvény, aminek a segítségével a tárgyakhoz tartozó képek megjeleníthetőek lesznek majd.

## 

### 

### **11.3.11** **CamembertViewModel**

#### · **Felelősség**

Az osztály felelős azért, hogy a Camembert tárgy megfelelően, képként megjeleníthető legyen, illetve használatát a nézetek különféle elemei is meg tudják hívni.

#### · **Ősosztályok**

ItemViewModel

#### · **Interfészek**

-

#### · **Attribútumok**

* -camembert: Camembert
  + A viewmodell által használandó tárgy, amivel később dolgozik is.

#### · **Metódusok**

* +getItemImage(): ImageIcon
  + Az absztrakt getItemImage megvalósítása a Camembert átlal.
  + Visszatér egy megfelelő méretű képpel, ami a játék során a tárgyat fogja szimbolizálni.
* +useCamembert(): void
  + A tárgy használatáért felelős függvény, aminek a segítségével a viewmodellen keresztül is használható lesz az adott tárgy.

### **11.3.12** **LegfrissitoViewModel**

#### · **Felelősség**

Az osztály felelős azért, hogy a Légfrissítő tárgy megfelelően, képként megjeleníthető legyen, illetve használatát a nézetek különféle elemei is meg tudják hívni.

#### · **Ősosztályok**

ItemViewModel

#### · **Interfészek**

-

#### · **Attribútumok**

* -legfrissito: Legfrissito
  + A viewmodell által használandó tárgy, amivel később dolgozik is.

#### · **Metódusok**

* -getItemImage(): ImageIcon
  + Az absztrakt getItemImage megvalósítása a Légfrssítő átlal.
  + Visszatér egy megfelelő méretű képpel, ami a játék során a tárgyat fogja szimbolizálni.
* -useLegfrissito(): void
  + A tárgy használatáért felelős függvény, aminek a segítségével a viewmodellen keresztül is használható lesz az adott tárgy.

### 

### **11.3.13** **HallgatoViewModel**

#### · **Felelősség**

A HallgatoViewModel osztály feladata, hogy a Hallgato objektumot reprezentálja a nézethez és kezelje annak állapotát és interakcióit.

· **Ősosztályok**

-

#### · **Interfészek**

-

#### · **Attribútumok**

* -hallgato: Hallgato
  + A Hallgato objektumot tárolja.
* -gameViewModel: GameViewModel
  + A GameViewModel objektumot tárolja.

#### · **Metódusok**

* +mozgas() : void
  + Megnyitja a RoomFrame ablakot a hallgató mozgatásához.
* +mozgas(Szoba szoba) : void
  + Mozgatja a hallgatót egy megadott szobába és frissíti a nézetet.
* +hasznal(int i) : void
  + Használja a hallgató táskájában lévő megadott tárgyat és frissíti a nézetet.
* +eldob(int i) : void
  + Eldobja a hallgató táskájában lévő megadott tárgyat és frissíti a nézetet.
* +felvesz(Targy targy) : void
  + Felveszi a megadott tárgyat és hozzáadja a hallgató táskájához.
* +teleportal(Tranzisztor t) : void
  + Teleportálja a hallgatót a megadott tranzisztor segítségével és frissíti a nézetet.
* +createItemFrame(SzobaViewModel szobaViewModel) : void
  + Létrehozza az ItemFrame ablakot.
* +createitemviewmodels() : ArrayList<ItemViewModel>
  + Visszaadja a hallgató táskájában lévő tárgyak ViewModel listáját.
* +giveTaskabanLevoTargyakKepe() : ImageIcon[]
  + Visszaadja a hallgató táskájában lévő tárgyak képeit.

### **11.3.14** **TranzisztorViewModel**

#### · **Felelősség**

Az osztály felelős azért, hogy a Tranzisztor tárgy megfelelően, képként megjeleníthető legyen, illetve használatát a nézetek különféle elemei is meg tudják hívni.

#### · **Ősosztályok**

ItemViewModel

#### · **Interfészek**

-

#### · **Attribútumok**

* -tranzisztor: Tranzisztor
  + A viewmodell által használandó tárgy, amivel később dolgozik is.

#### · **Metódusok**

* -getItemImage(): ImageIcon
  + Az absztrakt getItemImage megvalósítása a Tranzisztor által.
  + Visszatér egy megfelelő méretű képpel, ami a játék során a tárgyat fogja szimbolizálni.
* -useTranzisztor(): void
  + A tárgy használatáért felelős függvény, aminek a segítségével a viewmodellen keresztül is használható lesz az adott tárgy.

### 

### 

### **11.3.15** **RongyViewModel**

#### · **Felelősség**

Az osztály felelős azért, hogy a Rongy tárgy megfelelően, képként megjeleníthető legyen, a játék során.

#### · **Ősosztályok**

ItemViewModel

#### · **Interfészek**

-

#### · **Attribútumok**

* -rongy: Rongy
  + A viewmodell által használandó tárgy, amivel később dolgozik is.

#### · **Metódusok**

* -getItemImage(): ImageIcon
  + Az absztrakt getItemImage megvalósítása a Rongy átlal.
  + Visszatér egy megfelelő méretű képpel, ami a játék során a tárgyak fogja szimbolizálni.

### 

### 

### **11.3.16** **MapViewModel**

#### · **Felelősség**

A MapViewModel osztály feladata, hogy a Pálya objektumot reprezentálja a nézethez, és kezelje annak állapotát és interakcióit.

· **Ősosztályok**

-

#### · **Interfészek**

-

#### · **Attribútumok**

* -palya: Palya
  + A Pálya objektumot tárolja.
* -szobaviewmodels: ArrayList<SzobaViewModel>
  + A pálya szobáinak ViewModel listáját tárolja.
* -hallgatoviewmodels: ArrayList<HallgatoViewModel>
  + A pályán lévő hallgatók ViewModel listáját tárolja.

#### · **Metódusok**

* +leptet() : void
  + Meghívja a pálya léptet függvényét.

### **11.3.17** **SorospoharViewModel**

#### · **Felelősség**

Az osztály felelős azért, hogy a Söröspohár tárgy megfelelően, képként megjeleníthető legyen, a játék során.

#### · **Ősosztályok**

ItemViewModel

#### · **Interfészek**

-

#### · **Attribútumok**

* -sorospohar: Sorospohar
  + A viewmodell által használandó tárgy, amivel később dolgozik is.

#### · **Metódusok**

* -getItemImage(): ImageIcon
  + Az absztrakt getItemImage megvalósítása a Söröspohár átlal.
  + Visszatér egy megfelelő méretű képpel, ami a játék során a tárgyak fogja szimbolizálni.

### **11.3.18** **TVSZViewModel**

#### · **Felelősség**

Az osztály felelős azért, hogy a TVSZ tárgy megfelelően, képként megjeleníthető legyen, a játék során.

#### · **Ősosztályok**

ItemViewModel

#### · **Interfészek**

-

#### · **Attribútumok**

* -tvsz: TVSZ
  + A viewmodell által használandó tárgy, amivel később dolgozik is.

#### · **Metódusok**

* -getItemImage(): ImageIcon
  + Az absztrakt getItemImage megvalósítása a TVSZ átlal.
  + Visszatér egy megfelelő méretű képpel, ami a játék során a tárgyak fogja szimbolizálni.

### **11.3.19** **MaszkViewModel**

#### · **Felelősség**

Az osztály felelős azért, hogy a Maszk tárgy megfelelően, képként megjeleníthető legyen, a játék során.

#### · **Ősosztályok**

ItemViewModel

#### · **Interfészek**

-

#### · **Attribútumok**

* -maszk: Maszk
  + A viewmodell által használandó tárgy, amivel később dolgozik is.

#### · **Metódusok**

* -getItemImage(): ImageIcon
  + Az absztrakt getItemImage megvalósítása a Maszk átlal.
  + Visszatér egy megfelelő méretű képpel, ami a játék során a tárgyak fogja szimbolizálni.

### **11.3.15** **LogarlecViewModel**

#### · **Felelősség**

Az osztály felelős azért, hogy a Logarléc tárgy megfelelően, képként megjeleníthető legyen, a játék során.

#### · **Ősosztályok**

ItemViewModel

#### · **Interfészek**

-

#### · **Attribútumok**

* -logarlec: Logarlec
  + A viewmodell által használandó tárgy, amivel később dolgozik is.

#### · **Metódusok**

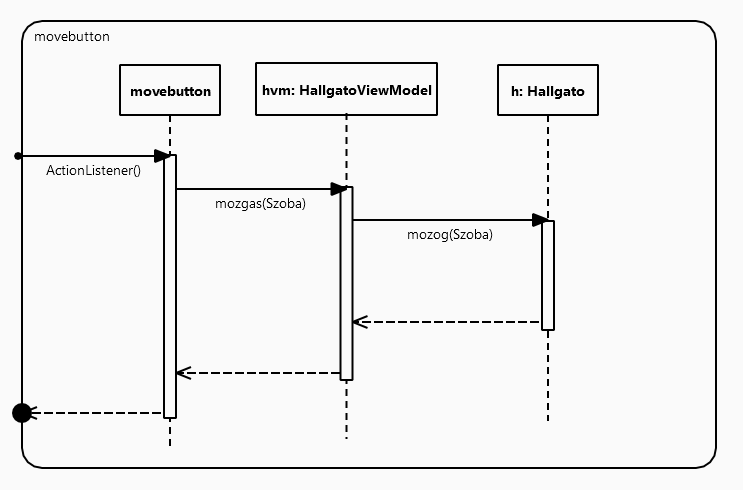
* -getItemImage(): ImageIcon
  + Az absztrakt getItemImage megvalósítása a Logarléc átlal.
  + Visszatér egy megfelelő méretű képpel, ami a játék során a tárgyak fogja szimbolizálni.

## 

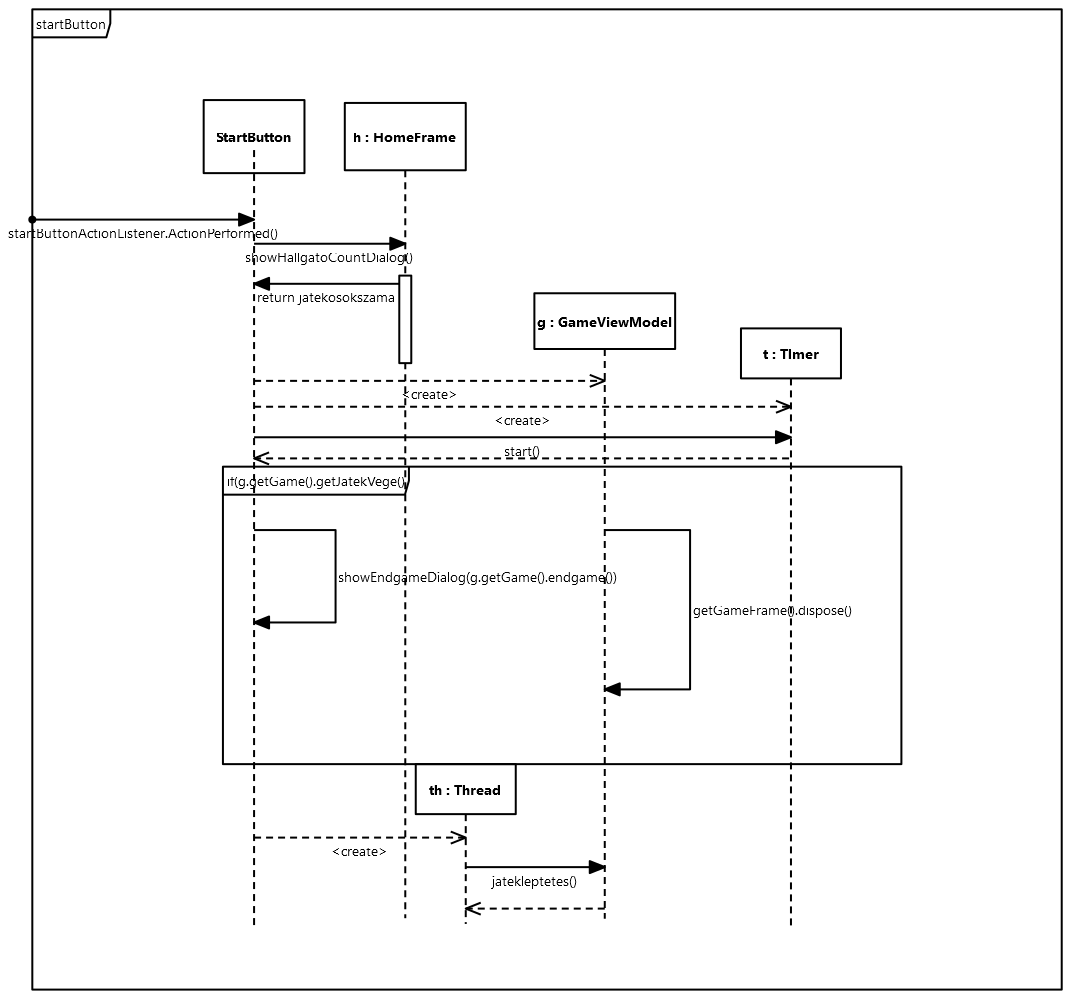
## **11.4** **Kapcsolat az alkalmazói rendszerrel**

**11.4.1 ItemFrame pickupbutton**

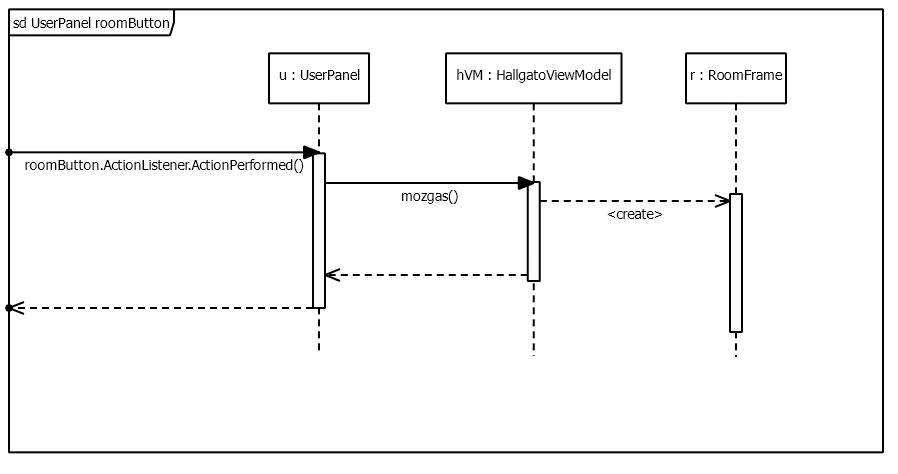
**11.4.2 RoomFrame movebutton**



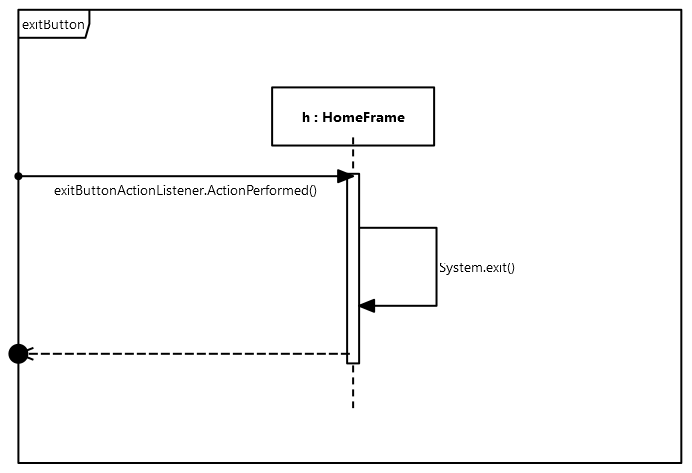
**11.4.3 HomeFrame StartButton**

****

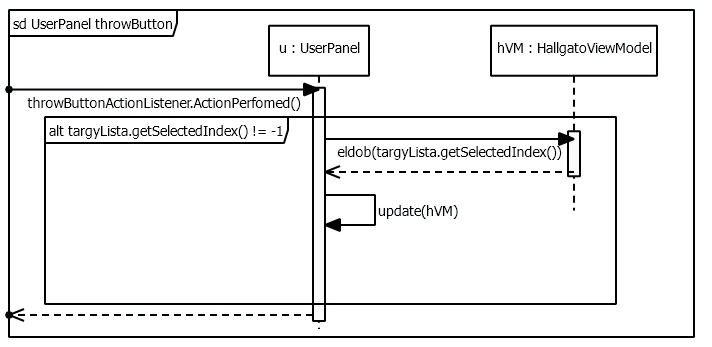
**11.4.4 UserPanel roomButton**

****

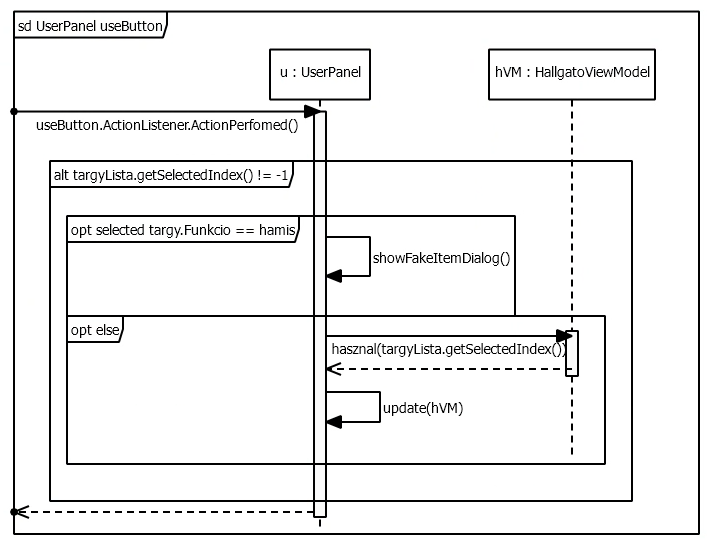
**11.4.5 HomeFrame exitButton**

****

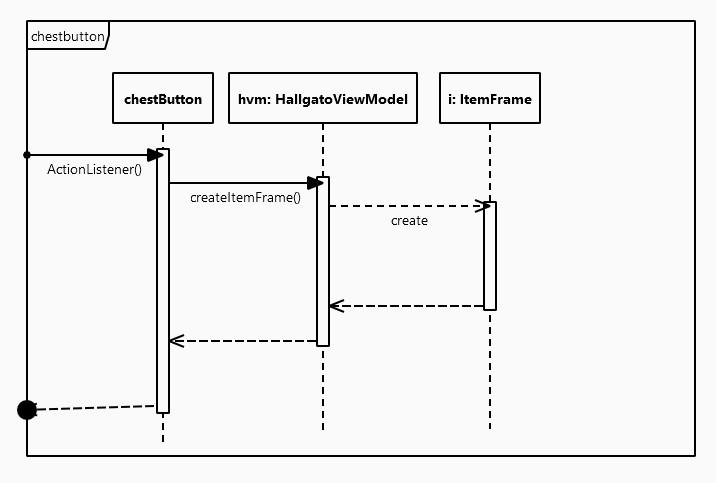
**11.4.6 UserPanel throwButton**

****

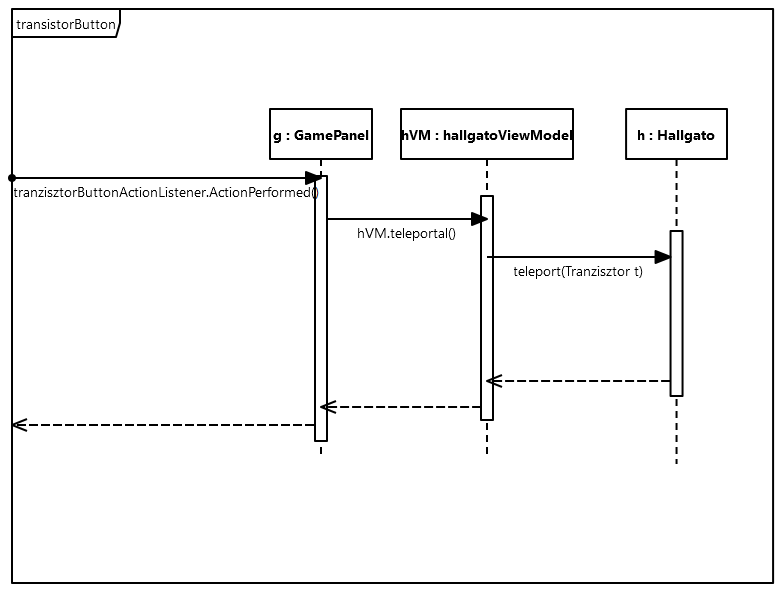
**11.4.7 UserPanel useButton**

****

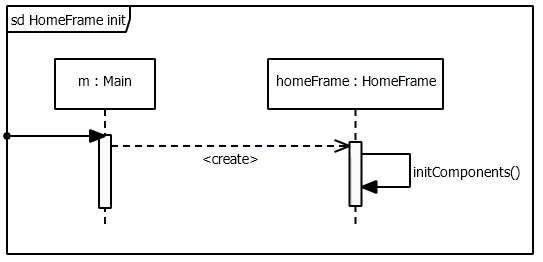
**11.4.8 GameFrame chestButton**

****

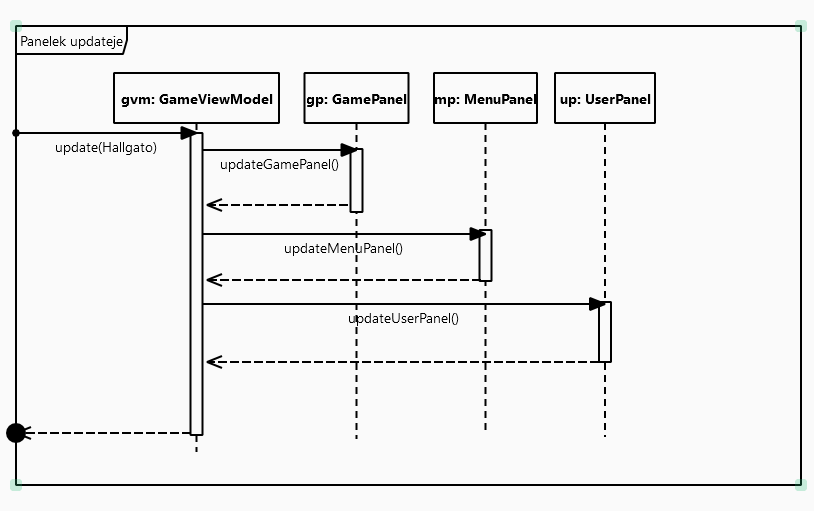
**11.4.9 GameFrame transistorButton**

****

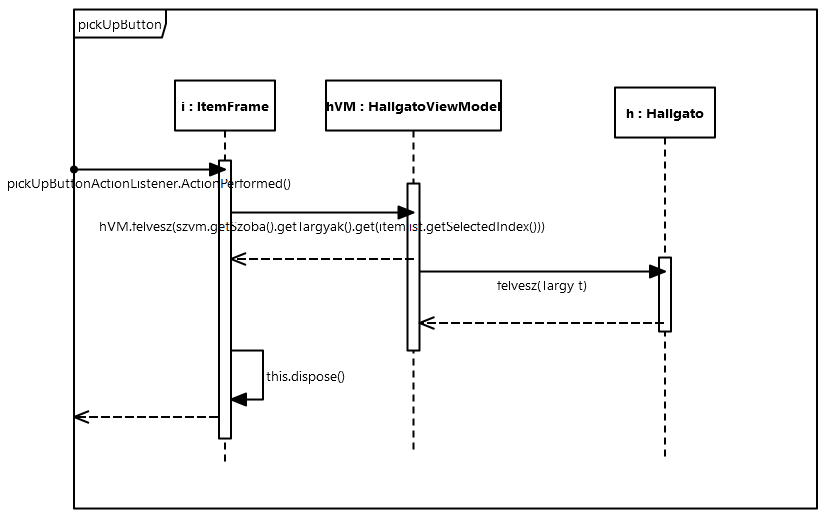
**11.4.10 HomeFrame inicializálás**

****

**11.4.11 Panelek updateje**

****

**11.4.12 pickupButton**

****

## **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 | 1,5 óra | Schulcz | Menupanel |
| 2024.05.05 18:00 | 1,5 óra | Tarsoly | UserPanel |