

11. Grafikus felület specifikációja

18 – Mumbai IT Solution

Konzulens:
Goldschmidt Balázs

Csapattagok

Cardinael Jan Maerten	M1GVE2	cardinael.jan@gmail.com
Görömbey Lilla Zsófia	EBQG95	gorombeylilla@gmail.com
Riba Miklós Pál	J519A1	ribamiki@outlook.hu
Király Bálint	EQF1M0	kiraly.balint@edu.bme.hu
Szagos Máté Antal	DKQQXE	szakos.mate@gmail.com

2024.05.05.

11. Grafikus felület specifikációja

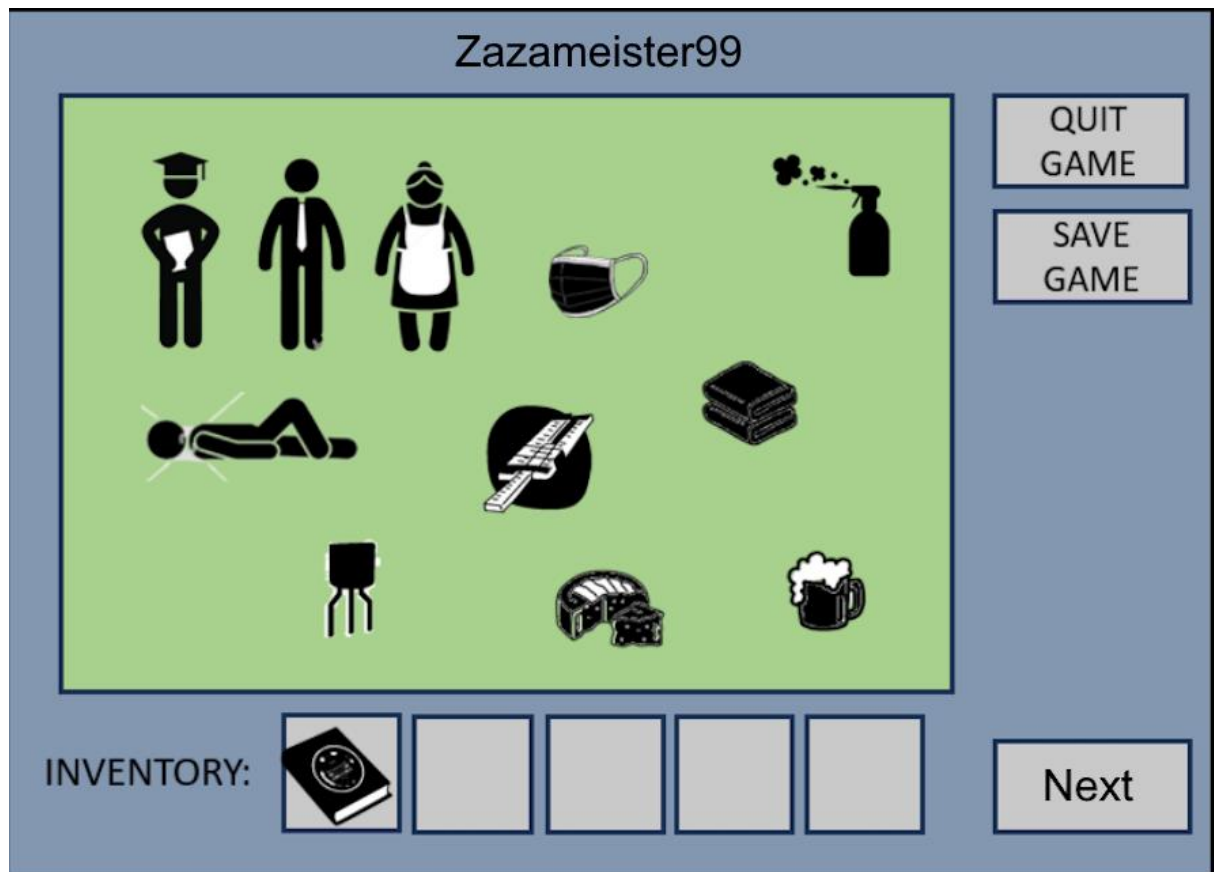
11.1A grafikus interfész



Menü



Szoba, benne személyek és tárgyak. Jobb oldalt a szobából nyíló ajtók.
A nem ájult hallgató Zuzumester88-hoz tartozik, aki éppen a soron következő játékos és van nála egy TVSZ.

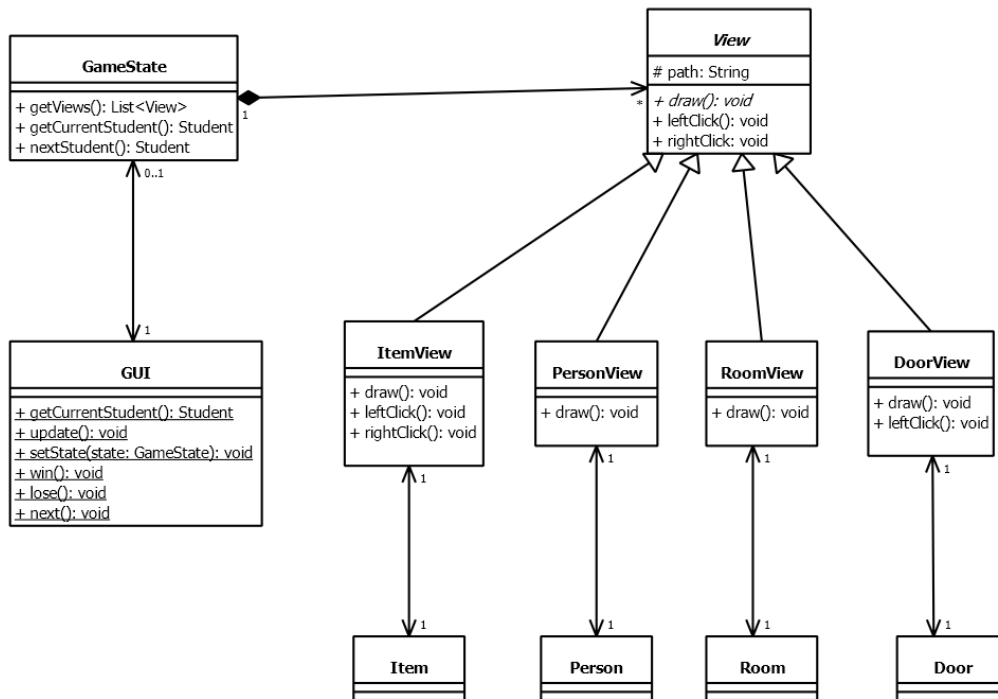


Elgázosított szoba, benne személyek és tárgyak. A szoba ajtajai éppen elbújtak, mivel a szoba el van átkozva.

A nem ájult hallgató Zazameister99-hez tartozik, aki éppen a soron következő játékos és van nála egy TVSZ.

11.2 A grafikus rendszer architektúrája

cls GUI ClassDiagram



11.2.1 A felület működési elve

A felület megalakításához azt az elvet követjük, hogy a modell értesíti a felületet. Ezt a GUI statikus osztály `update()` metódusán keresztül éri el, amit a modell minden eleme bármikor meghívhat (célszerűen akkor, amikor valamely attribútuma változott). A GUI osztály minden `update()`-kor értesíti a felület többi elemét, akik újrarajzolják a megjelenítendő objektumaikat.

11.2.2 A felület osztály-struktúrája

Lásd: Fenti diagram

11.3A grafikus objektumok felsorolása

11.3.1 GUI

- **Felelősség**

Statikus osztály, a View osztályok összefogását, változásokról való értesítését végzi.

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **-GameState state:** az aktuális játékállapothoz tartozó GameState példány.

- **Metódusok**

- **+Student getCurrentStudent():** visszaadja a state.currentStudent-et.
- **+void update():** minden Entity ezt hívja meg, amikor tulajdonságai frissülnek, ennek hatására minden View újra rajzolja az adott Entity-jét.
- **+void win():** a Student ezt hívja meg ha hozzákerül egy Logarléc.
- **+void lose():** a state hívja ha már nincs több élő Student aki sorra kerülhetne.
- **+void setState(GameState):** beállítja a state-t.
- **+void next():** a Next gombra kattintva hívódik meg, ez jelzi, hogy a következő játékos következik.

11.3.2 GameState

- **Változtatások:**

- A GameState külön számon tartja a Student típusú entitásokat, mivel őket kell majd szólítani mint játékosokat. A játékosok egymás után kerülnek sorra. Az éppen soron lévő játékost adja vissza a GameState újonnan bevezetett metódusa:

+ **Student getCurrentStudent():** visszaadja a soron lévő játékos Student objektumát.

- A View objektumokat az entitásokkal együtt hozzuk létre, így minden entitáshoz tartozik egy View példány.

+ **List<View> getViews():** visszaadja a View objektumok listáját.

- Új metódus annak jelzésére, hogy a következő Student jön:

+ **Student nextStudent()**: ha meghívódik akkor a currentStudent a soron következő Student lesz, ezzel is tér vissza.

11.3.3 View

- **Felelősség**

Megjelenítő (View) osztályok absztrakt ősosztálya..

- **Ősosztályok**

Nincs

- **Interfészek**

Nincs

- **Attribútumok**

- **#String path**: gyermek által tárolt Entity megjelenítéséhez szükséges képek mappájának elérési útja.

- **Metódusok**

- **+void draw()**: leszármazottak által tárolt Entity megjelenítése.
- **+void leftClick()**: akkor hívódik meg, ha a bal egérgommbal rányomtunk az ikonra
- **+void rightClick()**: akkor hívódik meg, ha a jobb egérgommbal rányomtunk az ikonra

11.3.4 ItemView

- **Felelősség**

Item-et megjelenítő osztály.

- **Ősosztályok**

View

- **Interfészek**

Nincs

- **Attribútumok**

- **-Item item**: megjelenítendő Item referencia.

- **Metódusok**

- **+void draw()**: tárolt Item megjelenítése aktív/inaktív állapota alapján.
- **+void leftClick()**: ha az Item-nek van tulajdonosa, aktiválja az Item-et, ha pedig egy .szobában van elejtve, akkor a soron lévő Student-tel felveteti az Item-et.
- **+void rightClick()**: ha az Item-nek van tulajdonosa, eldobatja vele az Item-et.

11.3.5 PersonView

- **Felelősség**

Person-t megjelenítő osztály.

- **Ősosztályok**

View

- **Interfészek**

Nincs

- **Attribútumok**

- **-Person person:** megjelenítendő Person referencia.

- **Metódusok**

- **+void draw():** tárolt Person megjelenítése ájult/ájulatlan állapota alapján.
- **+void leftClick():** üres
- **+void rightClick():** üres

11.3.6 RoomView

- **Felelősség**

Room-ot megjelenítő osztály.

- **Ősosztályok**

View

- **Interfészek**

Nincs

- **Attribútumok**

- **-Room room:** megjelenítendő Room referencia.

- **Metódusok**

- **+void draw():** tárolt Room megjelenítése elgázosított/elgázosítatlan állapota alapján.
- **+void leftClick():** üres
- **+void rightClick():** üres

11.3.7 DoorView

- **Felelősség**

Door-t megjelenítő osztály.

- **Ősosztályok**

View

- **Interfészek**

Nincs

- **Attribútumok**

- **-Door door:** megjelenítendő Door referencia.

- **Metódusok**

- **+void draw():** tárolt Door megjelenítése/eltüntetése attól függően, hogy a Door-t tartalmazó Room elátkozott-e vagy sem.
- **+void leftClick():** az éppen soronlévő játékost átlépteti az ajtón
- **+void rightClick():** üres

11.3.8 Entity

- **Változtatások:**

Az alábbi függvények minden nem absztrakt implementációjukban visszatérésük előtt meghívják a GUI osztály update() függvényét:

+void tick()

11.3.9 Person

- **Változtatások:**

Az alábbi függvények minden nem absztrakt implementációjukban visszatérésük előtt meghívják a GUI osztály update() függvényét:

+void drop(Item item)

+void move(Door door)

+void pickup(Item item)

11.3.10 Student

- **Változtatások:**

+void slideRuleNotification(): meghívja a GUI-nak a win() metódusát

11.3.11 Item

- **Változtatások:**

Az alábbi függvény minden nem absztrakt implementációjában visszatérése előtt meghívja a GUI osztály update() függvényét:

+void activate()

11.3.12 Room

- **Változtatások:**

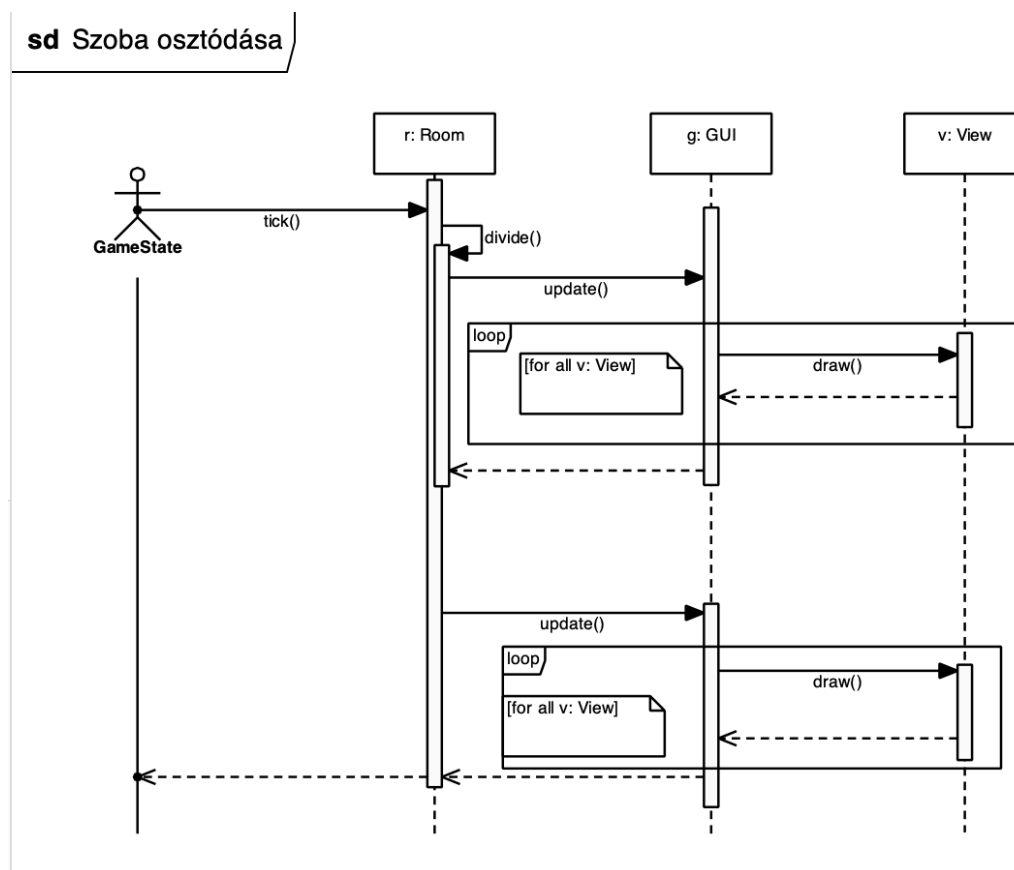
Az alábbi függvények visszatérésük előtt meghívják a GUI osztály update() függvényét:

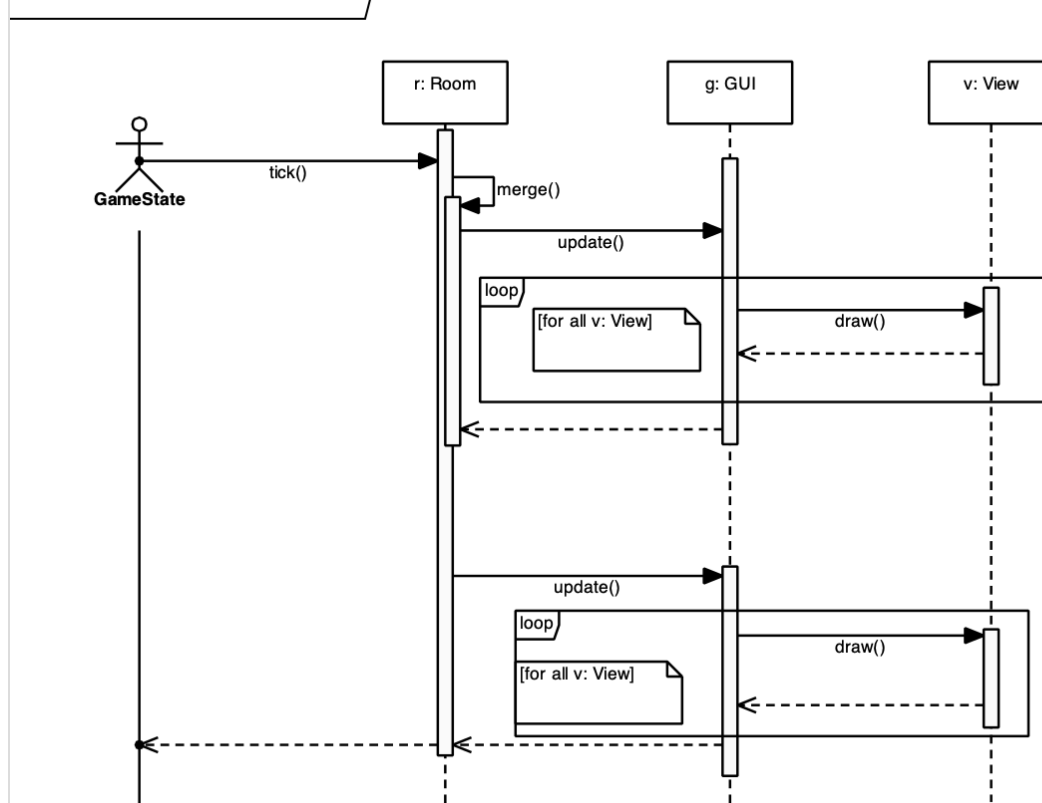
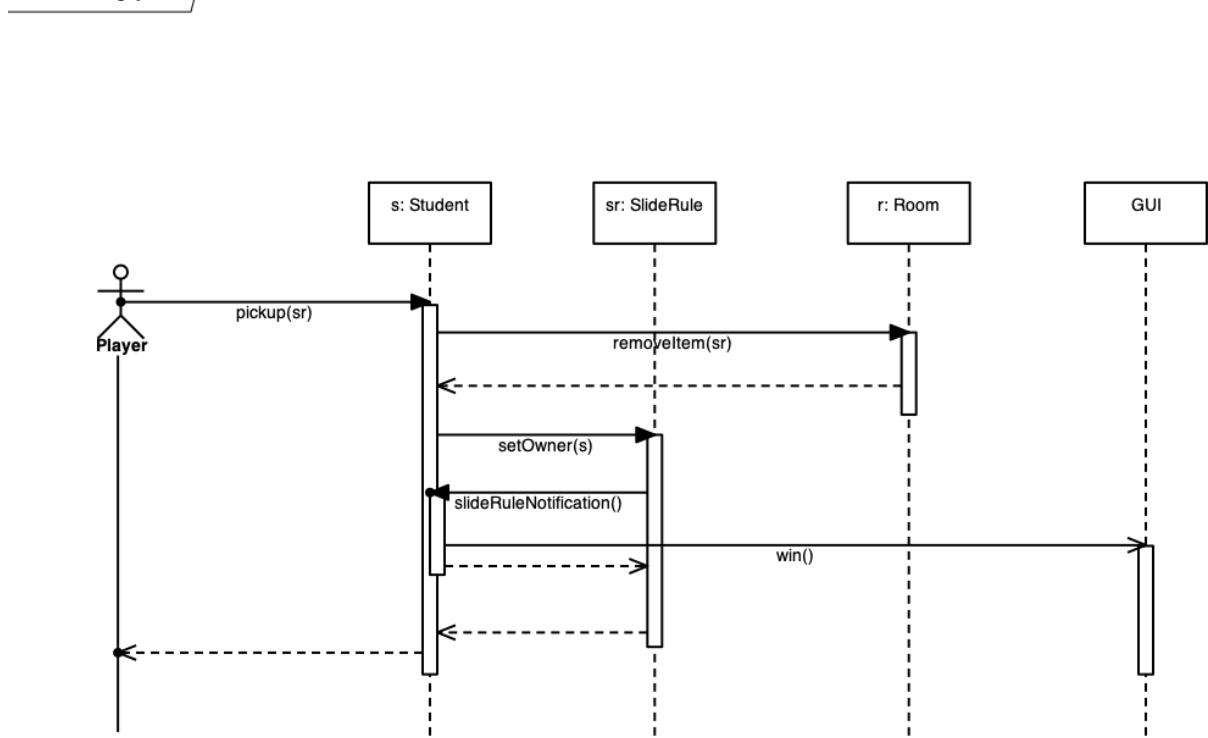
+void divide()

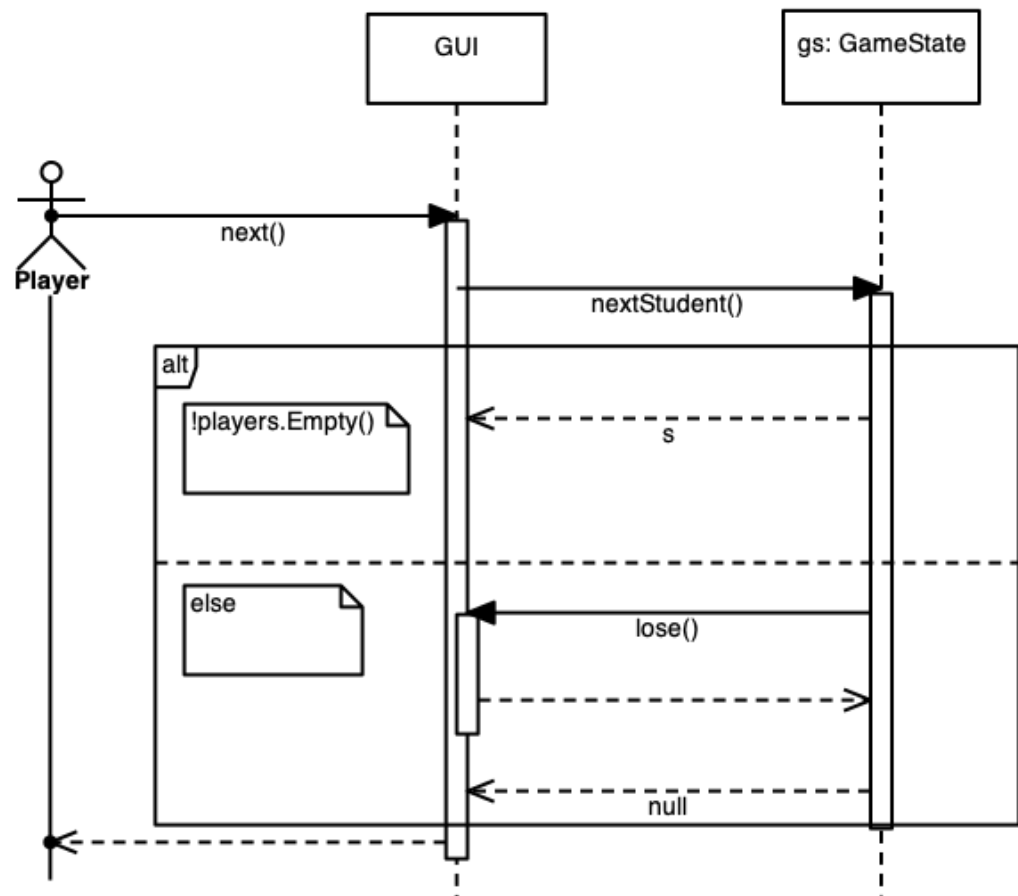
+void merge()

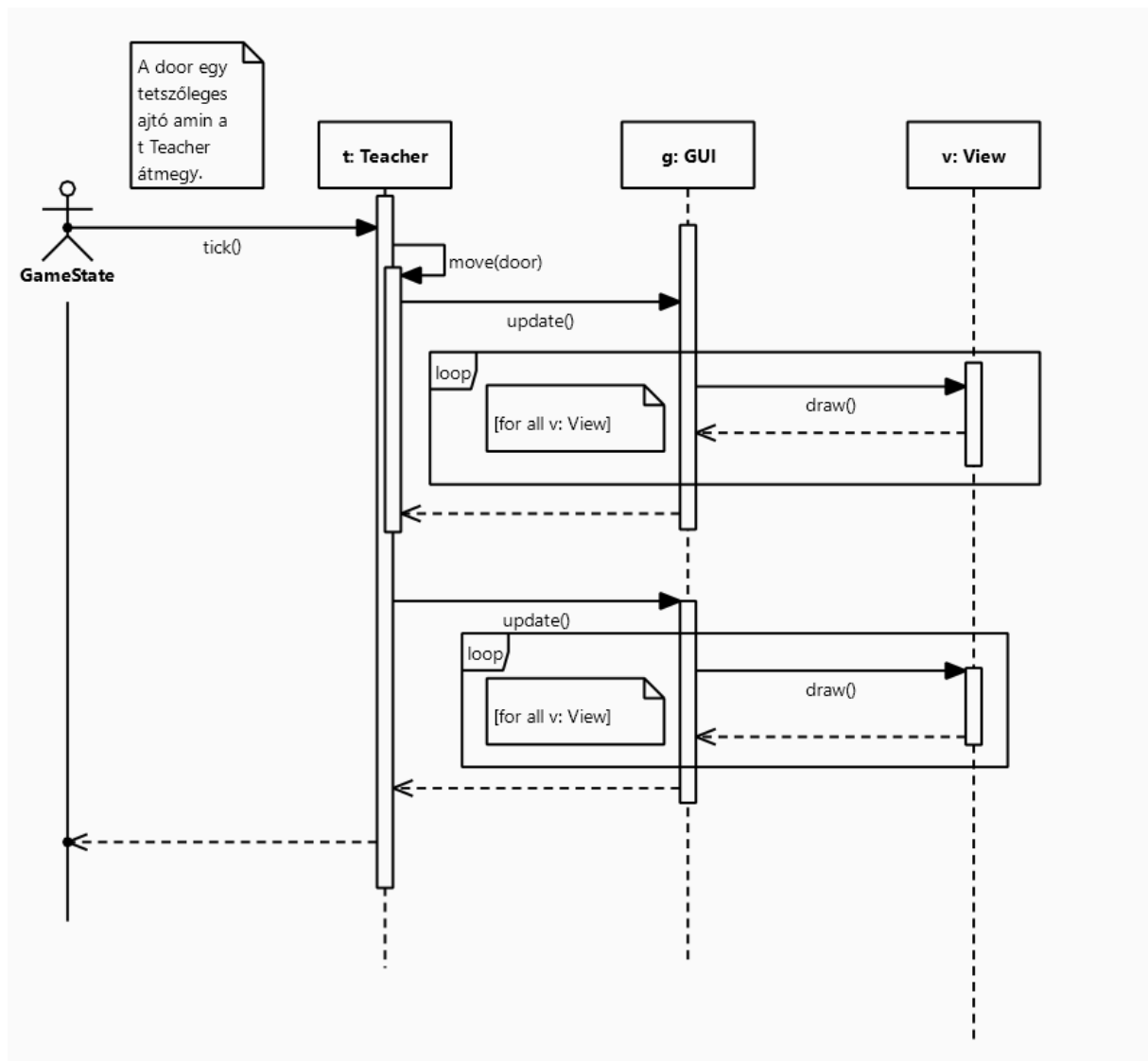
11.4Kapcsolat az alkalmazói rendszerrel

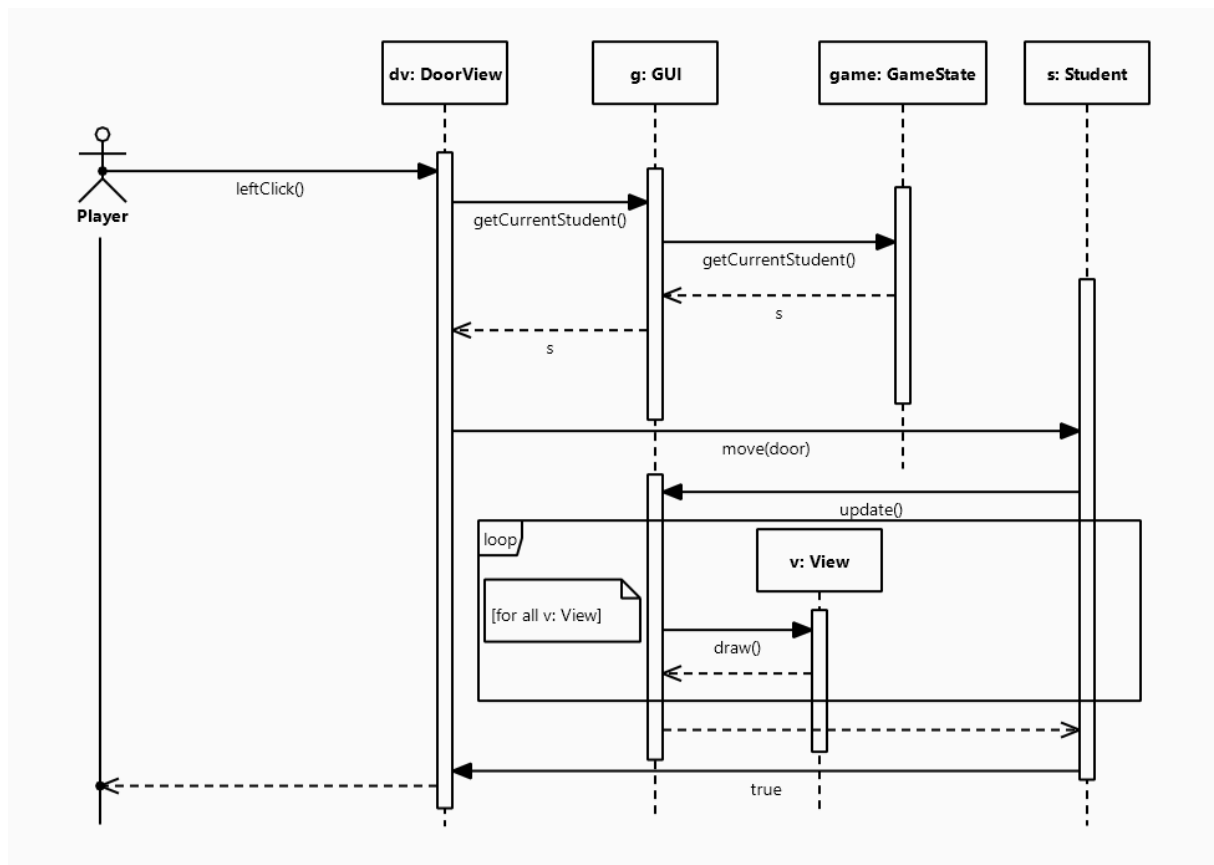
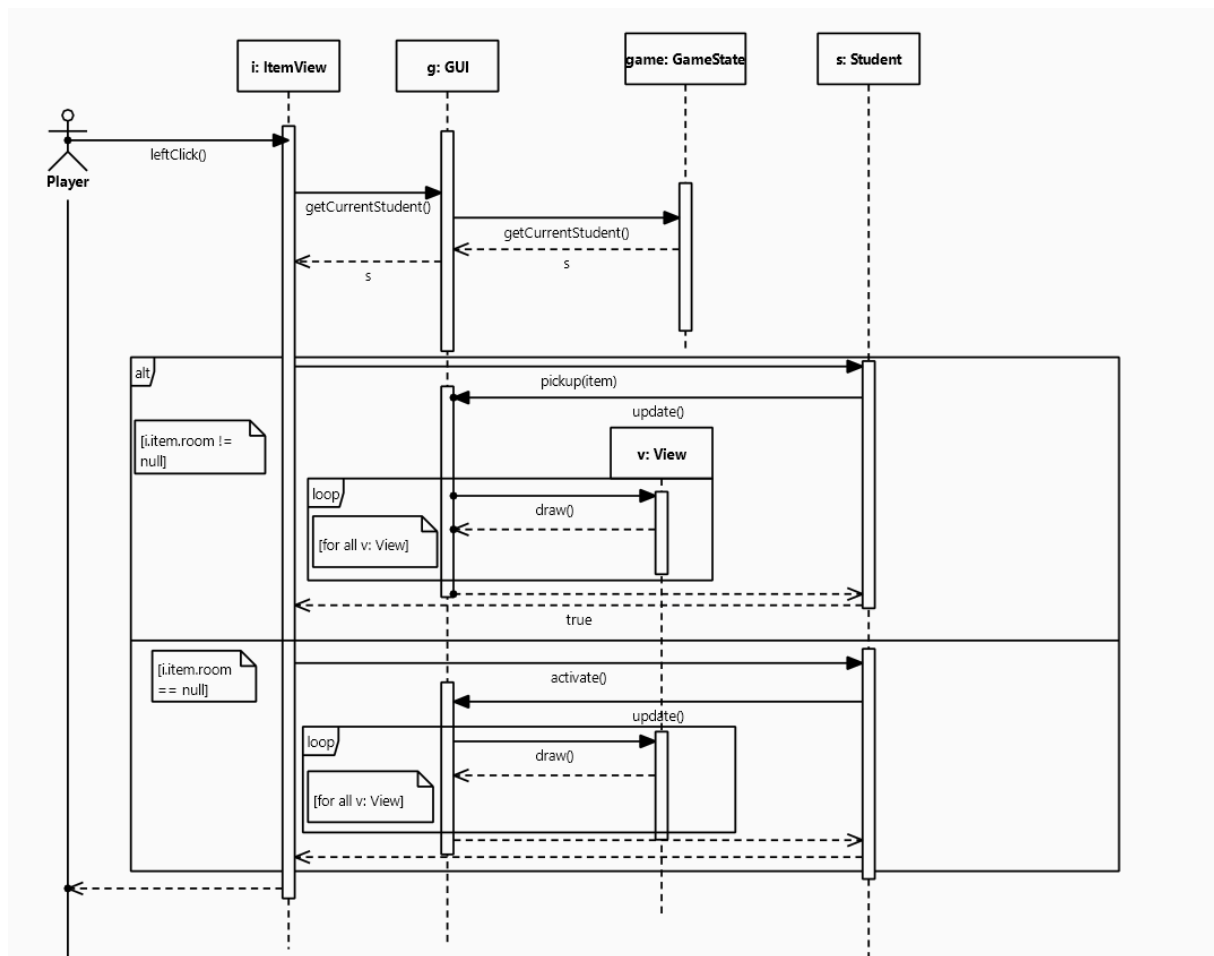
Az alábbi diagramokon a modell működését nem írtuk le, csak azokat a részeket tüntettük fel, amik relevánsak a View szempontjából.

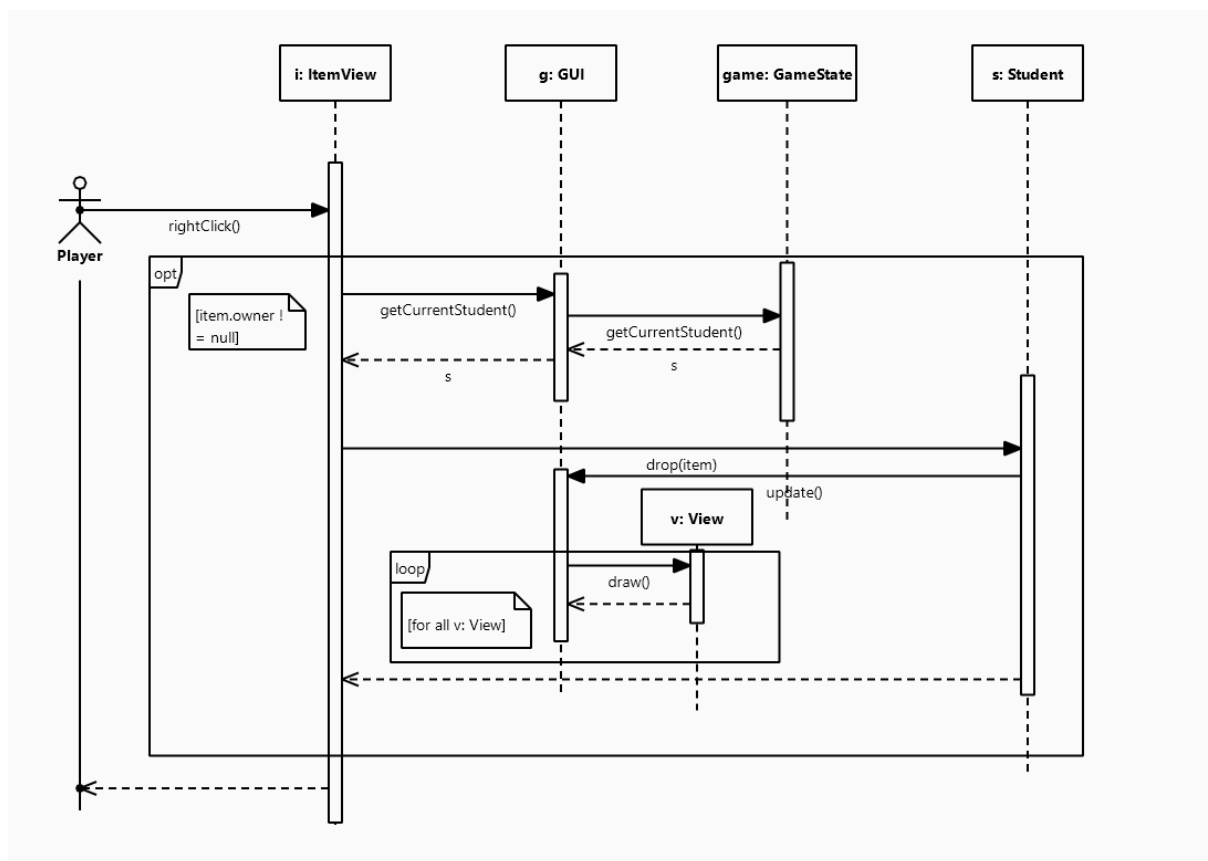


sd Szobák összeolvadása**sd Játék megnyerése**

sd Játék elvesztése







Napló

Kezdet	Időtartam	Résztevők	Leírás
2024. 05. 05. 13:00	4 óra	Riba	11.2.1 és 11.3 fejezet
2024. 05. 05. 13:00	2 óra	Görömbey	11.1 fejezet
2024. 05. 06. 10:00	3 óra	Király	Szekvenciadiagramok, osztálydiagram
2024. 05. 06 11:00	2 óra	Szakos	Az osztálydiagram elkészítése és közreműködés a tervezésben.
2024.05.06 10:00	3 óra	Cardinael	Szekvenciák