**11. Grafikus felület specifikációja**

1. **Grafikus felület specifikációja**
   1. ***A grafikus interfész***

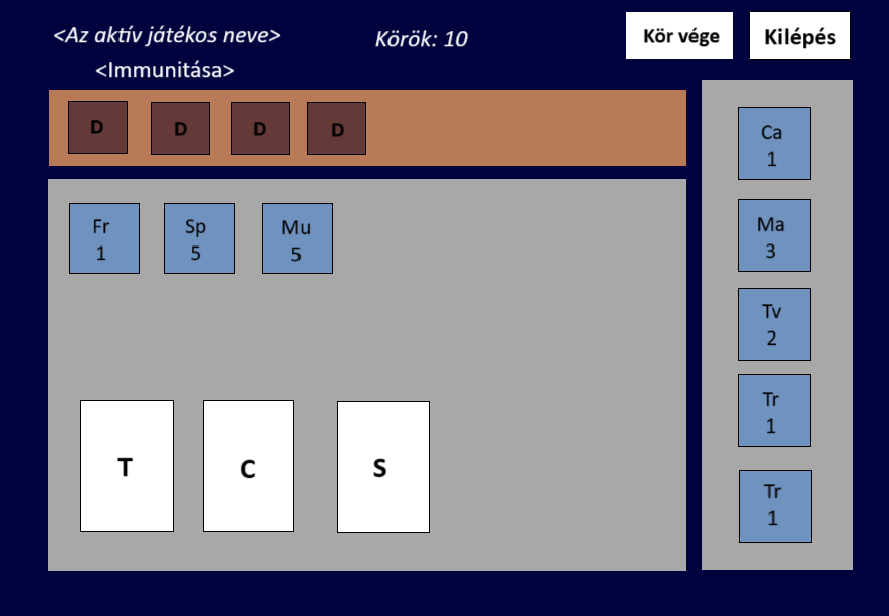
**

**A főmenü:**

A menü 3 különböző ablak, amelyek egymás hatására jelennek meg.

* Az első megjelenő ablakban a felhasználó, indíthat egy új játékot vagy kiléphet az alkalmazásból.
* Új játék indítása esetén a felhasználónak először meg kell adnia a játékosok számát. Majd a “következő” gomb megnyomásával elnevezheti őket.
* Minden játékosnak meg kell adni egy nevet, ezt a szöveges mező kitöltésével majd a “következő” gomb megnyomásával teheti meg a felhasználó.

A menük tetején a játék neve olvasható.

**

**A játék felülete:**

A játék folyamán mindig az aktív játékos adatai állapota és helyzete lesz látható, továbbá a hátralévő körök száma.

Az ablakon megjelenő elemek:

* Az aktív játékos neve, immunitása.
* Hátralévő körök száma.
* Kör vége gomb: ha a játékos szoba váltás nélkül akarja befejezni a körét akkor azt ezzel a gombbal teheti meg.
* Kilépés gomb: ha a játékos idő előtt abba szeretné hagyni a játékot, akkor ezzel a gombbal kiléphet vissza a főmenübe.
* A világos barna hátterű sáv: Az aktív játékos aktuális szobájának ajtói jelennek itt meg. Sötétbarna “D” feliratú négyzetként.
* Jobb oldali szürke hátterű oszlop: Itt jelenek meg a játékos táskájában lévő tárgyak, kék hátterű négyzetként. A tárgyak neve és usage értéke van feltüntetve minden tárgyon.
* Nagyobb szürke hátterű téglalap: A játékos aktuális szobáját megjelenítő elem. A felső részében láthatóak a szobában lévő tárgyak, a fentebb említett megjelenítést alkalmazva. Alul láthatóak a szobában tartózkodó karakterek. Fehér, mozgásképtelen karakterek esetén kék, háttérrel, “T” betű esetén Oktatóról, “C” esetén Takarítóról, “S” esetén pedig Hallgatóról van szó. A szoba háttérszíne zöld, ha az gázos, alapesetben pedig szürke.
  1. ***A grafikus rendszer architektúrája***
     1. **A felület működési elve**

A megjelenítés pull alapelvű, mivel a View minden esetben a modellt kérdezi le annak változásairól. Minden interakció után lefut egy update függvény amiben a View újra lekéri a modell új állapotát a Controlleren keresztül.

* + 1. **A felület osztály-struktúrája**

**

* 1. ***A grafikus objektumok felsorolása***
     1. **(Módosítás) Controller**
        + **Asszociáció**
* **mainMenuView : MainMenuView:** a menü megjelenítéséért felelős osztály
* **gameView : GameView:** a játék megjelenítéséért felelős osztály
  + 1. **MainMenuView**
       - **Felelősség**

A játék "kezdőképernyője". Innen lehet elindítani a játékot, illetve kilépni belőle. Indítást követően sorban bekéri a szükséges paramétereket (játékosok száma, nevük), majd inicializálja a GameView-t.

* + - * **Ősosztályok**

JFrame

* + - * **Attribútumok**
* **- newGameButton : JButton:** a kezdőképernyőn ezzel léphetünk át a játék megkezdése előtti inicializáló ablakra
* **- exitButton : JButton:** a játékból való kilépésre szolgál
* **- nextButton : JButton:**
  + - a játékosszám megadásakor továbblép a játékosok neveinek megadására
  + - a játékosok neveinek megadásakor továbblép a következőre
  + - utolsó játékos esetén elindítja a játékot
* **- nameTextField : JTextField:** a játékosok ide írhatják be a nevüket
* **- playerCount : JFormattedTextField:** a játékosok ide írhatják be, hogy hányan vannak
* **- playersLeft : int:** nyomon követi, hogy még hány játékos nem írta be a nevét
  + - * **Metódusok**
* **+ MainMenuView():** konstruktor, felépíti a kezdeti nézetét, kettő gombbal
* **+ displayPlayerNumberSelection():** átalakítja az ablakot, hogy megadható legyen a játékosszám
* **+ displayPlayerNameSelection():** átalakítja az ablakot, hogy az első játékos megadhassa a nevét
* **+ updatePlayerNameSelection():** frissíti az ablakot, hogy a következő játékos megadhassa a nevét
  + 1. **GameView**
       - **Felelősség**

A játék nézetéért felelős osztály. Többek között megjeleníti a jelenlegi játékos táskáját, és a szobát, amiben tartózkodik.

* + - * **Ősosztályok**

JFrame

* + - * **Attribútumok**
* **- activePlayer: Student** - az éppen soron lévő játékos
* **- activePlayerLabel: JLabel** - a jelenlegi játékos nevének megjelenítése
* **- turnsLeftLabel: JLabel** - a hátralévő körök számának megjelenítése
* **- itemsPanel: JPanel -** a panel, amelyen a játékos táskájában lévő tárgyak jelennek meg
* **- doorsPanel: JPanel -** a panel, amelyen a szobában lévő ajtók jelennek meg
* **- roomView: RoomView -** annak a szobának a megjelenítése, ahol a játékos tartózkodik
* **- exitButton: JButton -** gomb, amivel ki lehet lépni a játékból
* **- endTurnButton: JButton -** gomb, amivel a soron lévő játékos be tudja fejezni a körét
  + - * **Metódusok**
* **+ GameView(Student activeStudent, int turnsLeft)**: konstruktor, amely beállítja a paraméterként megadott hallgatót az activePlayer-re, ezen felül létrehozza a pályát: kiírja az activePlayer nevét, megjeleníti az aktív játékos táskájában található tárgyakat, a szobát, amelyben tartózkodik, és az abban lévő ajtókat. Ezen felü kiírja a hátralévő körök számát és megjeleníti az exitButton-t és az endTurnButton-t is.
* **+ updateView(Student activeStudent, int turnsLeft):** a megadott játékos szerint frissíti a pályát, és a hátralévő körök számát
  + 1. **RoomView**
       - **Felelősség**

Az aktuális szobát megjelenítő osztály. A szobában jelennek meg a benne tartózkodó játékososk továbbá az ott felvehető tárgyak.

* + - * **Ősosztályok**

JPanel

* + - * **Attribútumok**
* **- itemsPanel**: **JPanel -** a szobában lévő tárgyakat megjelenítő panel.
* - **charactersPanel: JPanel -** a szobában lévő játékosokat megjelenítő panel.
  + - * **Metódusok**
* **+ RoomView(Room room) -** az osztály konstruktora, a kapott szoba adatai alapján feltölti a panelt, karakterekkel és tárgyakkal.
* - **createLook(): void -** a panel megjelenítéséhez tartozó paramétereket állítja be (háttérszín, méret…).
  + 1. **CharacterView**
       - **Felelősség**

A karakterek grafikus kinézetéért felel, egyik fő funkciója, hogy a karaktert megjelenését kezeli fagyott állapotának megfelelően.

* + - * **Ősosztályok**

JLabel

* + - * **Metódusok**
* **+CharacterView(Character c)**: konstruktor, benne állítódik be a megjelenés, illetve a paraméterként kapott karakter frozen értéke alapján állítja be a színét.
  + 1. **ButtonActionListener**
       - **Felelősség**

A felületen megjelenő különböző bal egérgomb-nyomásokat kezeli le, meghívja a megfelelő metódusokat.

* + - * **Ősosztályok**

ActionListener;

Inner class a Controller osztályban

* + - * **Attribútumok**
* **-command: String**: a konstruktorban kapott parancs, amit majd az actionPerformed() függvény használ fel
  + - * **Metódusok**
* **+ButtonActionListener(String cmd)**: konstruktor, paraméterként kap stringként egy commandot, ami a gomb funkciója
* **+void actionPerformed(ActionEvent e)**: meghívja a Controller osztály (ebben az osztályban inner class) commandCalls(command) függvényét, ami elvégzi a megfelelő metódusok meghívását.
  + 1. **RightClick\_MouseListener**
       - **Felelősség**

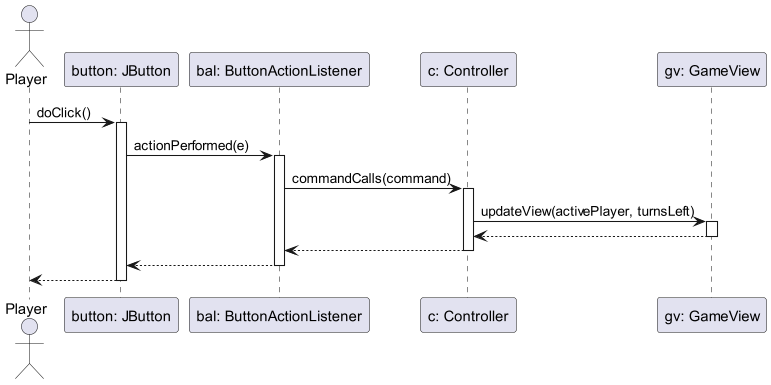
A felületen megjelenő különböző jobb egérgomb-nyomásokat kezeli le, meghívja a megfelelő metódusokat.

* + - * **Ősosztályok**

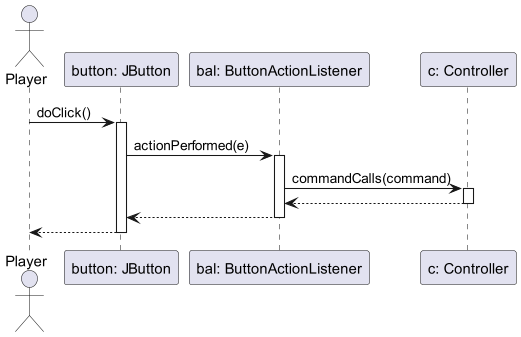
MouseAdapter;

Inner class a Controller osztályban

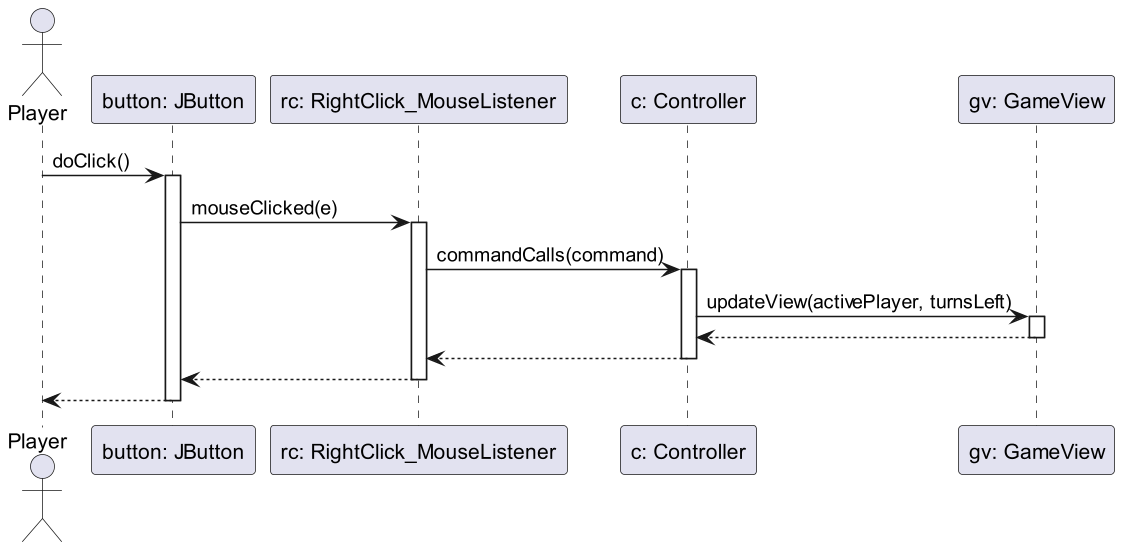
* + - * **Attribútumok**
* **-command: String**: a konstruktorban kapott parancs, amit majd az actionPerformed() függvény használ fel
  + - * **Metódusok**
* **+RightClick\_MouseListener(String cmd)**: konstruktor, paraméterként kap stringként egy commandot, ami a gomb funkciója
* **+void mouseClicked(MouseEvent e)**: meghívja a Controller osztály (ebben az osztályban inner class) commandCalls(command) függvényét, ami elvégzi a megfelelő metódusok meghívását. Mindez annak függvényében történik, hogy a jobb egérgomb volt-e megnyomva, ezt ellenőrzi először a függvény.
  1. ***Kapcsolat az alkalmazói rendszerrel***
     1. **Játékirányító gombok használata (Ajtó gomb, Tárgy gomb, Kör vége gomb)**



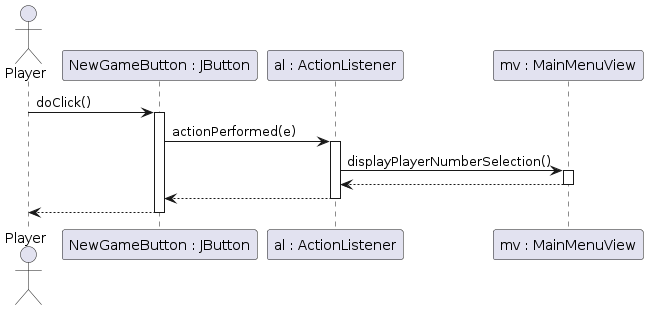
* + 1. **Kilépés gomb használata**

****

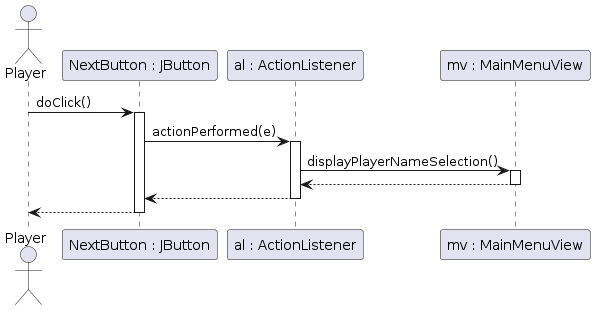
* + 1. **Jobb egérgombbal gombra kattintás**



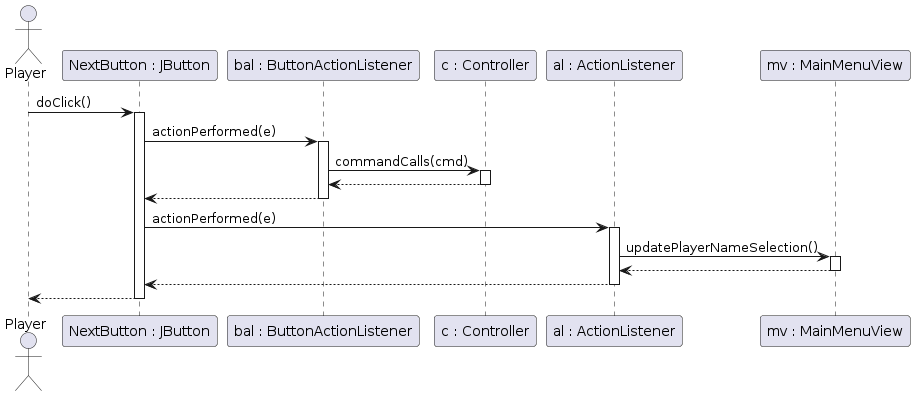
* + 1. **NewGameButton használata**



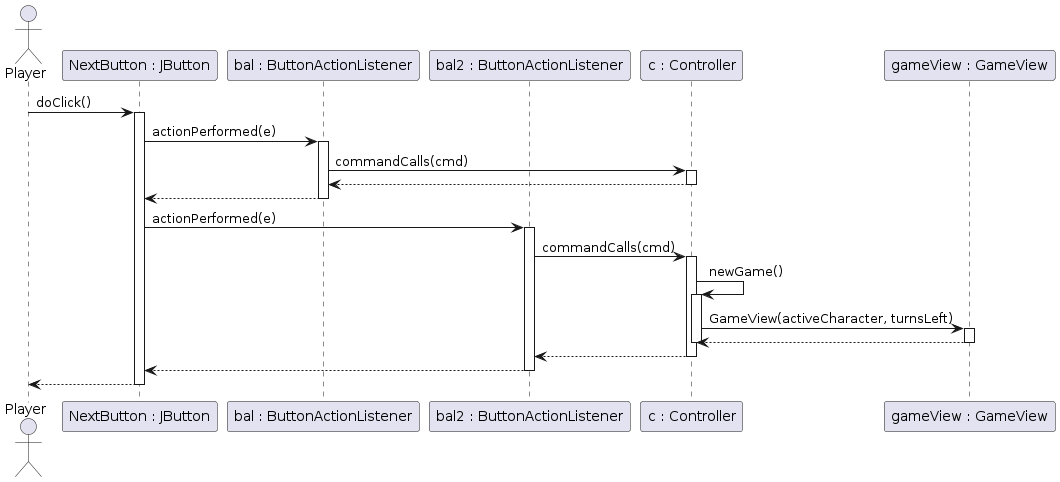
* + 1. **NextButton első használata (karakterek számának megadása)**

****

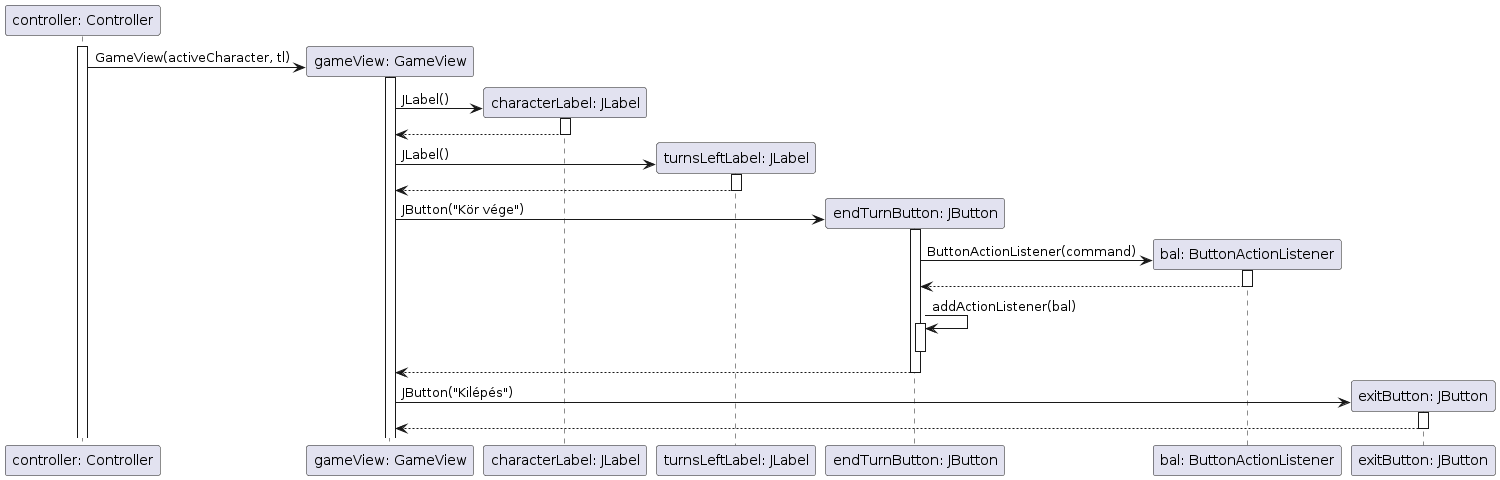
* + 1. **NextButton használata első név megadásától az utolsó előttiig**

****

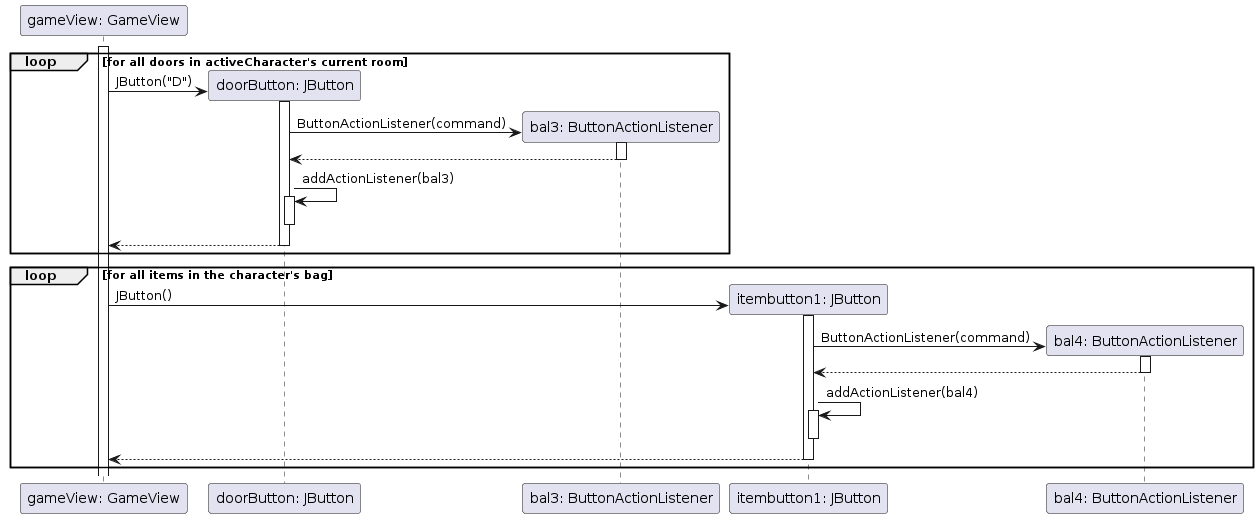
* + 1. **NextButton utolsó használata (utolsó név megadása)**



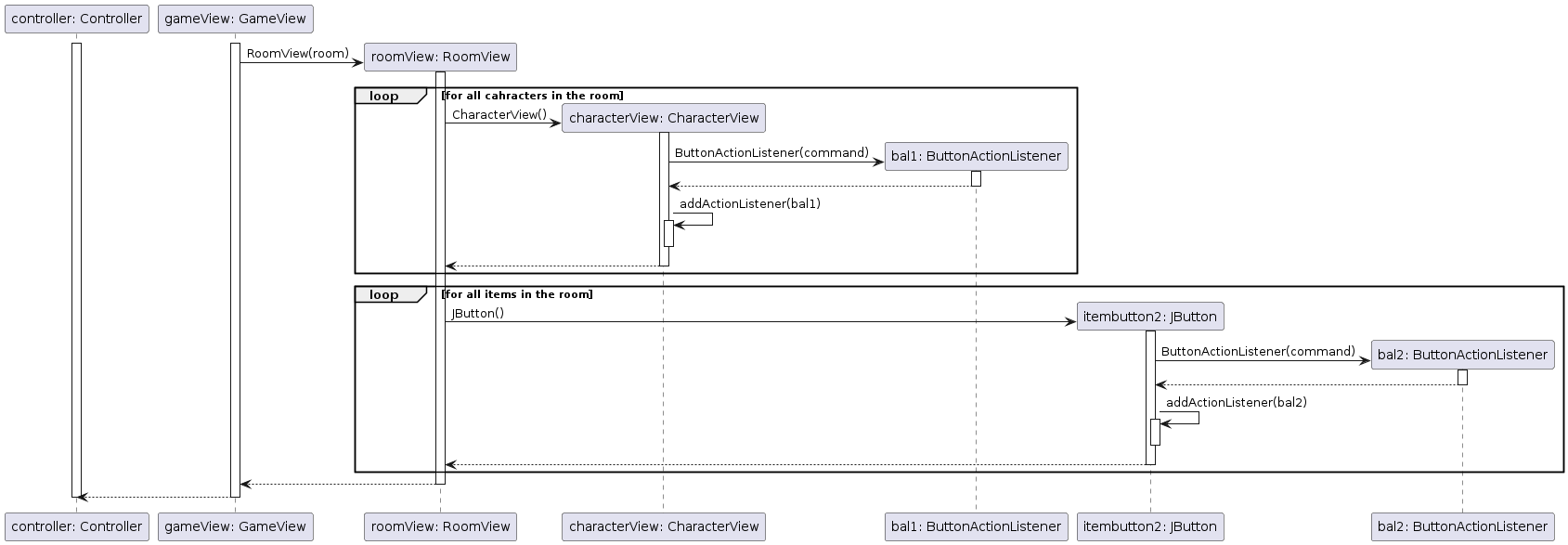
* + 1. **A grafikus felület felépítése (új játék vagy valamilyen update hatására)**

****

* + 1. **Felépítés folytatása**

****

* + 1. **Felépítés folytatása**

****