



Szoftver projekt laboratórium

GRAFIKUS FELÜLET SPECIFIKÁCIÓJA

Csapat

61 - Zeusz

Konzulens

Dr. Simon Balázs

Csapattagok

Balla Gergely	[NEPTUN]	[email]
Joób Zalán Miklós	[NEPTUN]	[email]
Nagy Alexandra	[NEPTUN]	[email]
Tóth Boldizsár	[NEPTUN]	[email]
Zelei Mátyás	[NEPTUN]	[email]

2024. május 6.

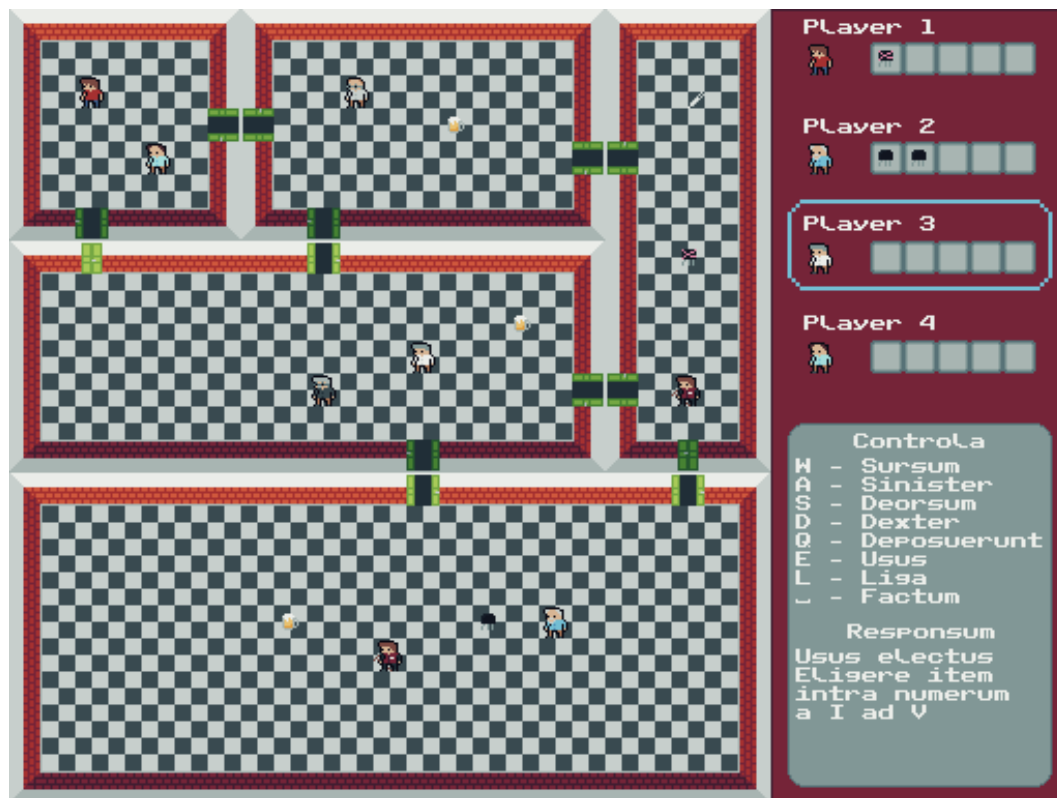
11. fejezet

Grafikus felület specifikációja

11.1. A grafikus interfész



11.1. ábra. Menü terve



11.2. ábra. Játék felület terve

11.1.1. Vezérlés

A program indításakor a főmenü nyílik meg. A főmenüből új játék indítani, illetve kilépni lehet. Új játék indításakor a játéklablak nyílik meg.

Itt az aktuálisan körön lévő játékost lehet irányítani. A térképen mozgatni a 'WASD' gombokkal lehet. Egy tárgyat felvenni úgy lehet, hogy a játékost ráléptetjük a csempére, amin van. Ajtóra lépés esetén a játékos megkísérel átlépni az ajtón. Falra nem lehet lépni. Más személy által elfoglalt csempére nem lehet lépni.

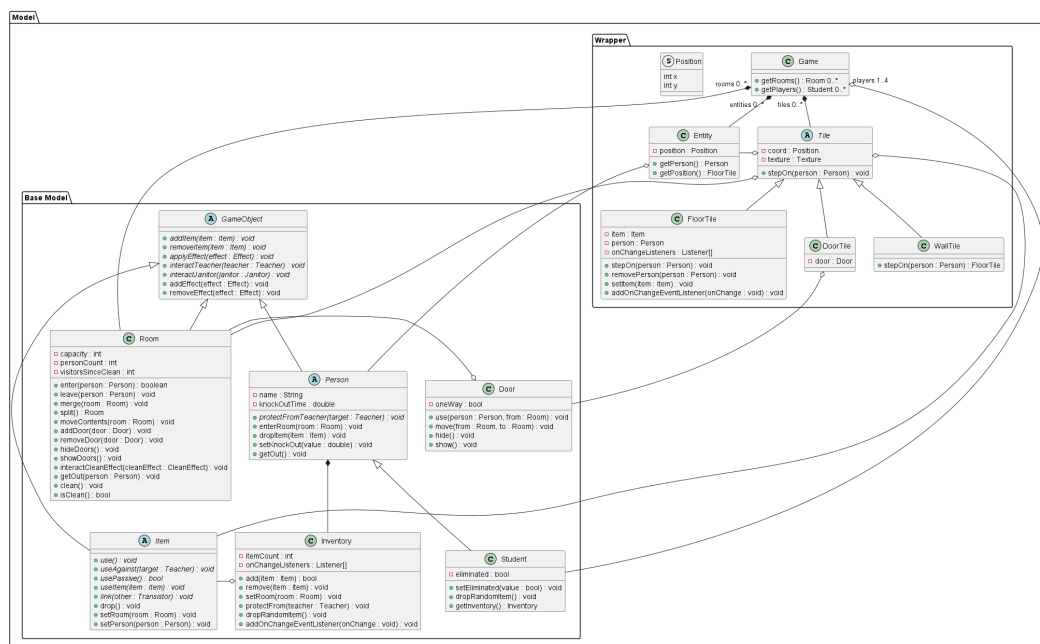
A játékosoknál lévő tárgyak az oldalsó panelen láthatók. 'E' gomb megnyomásakor kiválaszthatunk egy tárgyat, hogy használjuk. 'Q' gombbal hasonlóan eldobunk egy tárgyat. 'L' gombbal összelinkeljük a nálunk lévő tranzisztorokat. Space megnyomásával befejezzük a körünket, a következő játékos jön.

11.2. A grafikus rendszer architektúrája

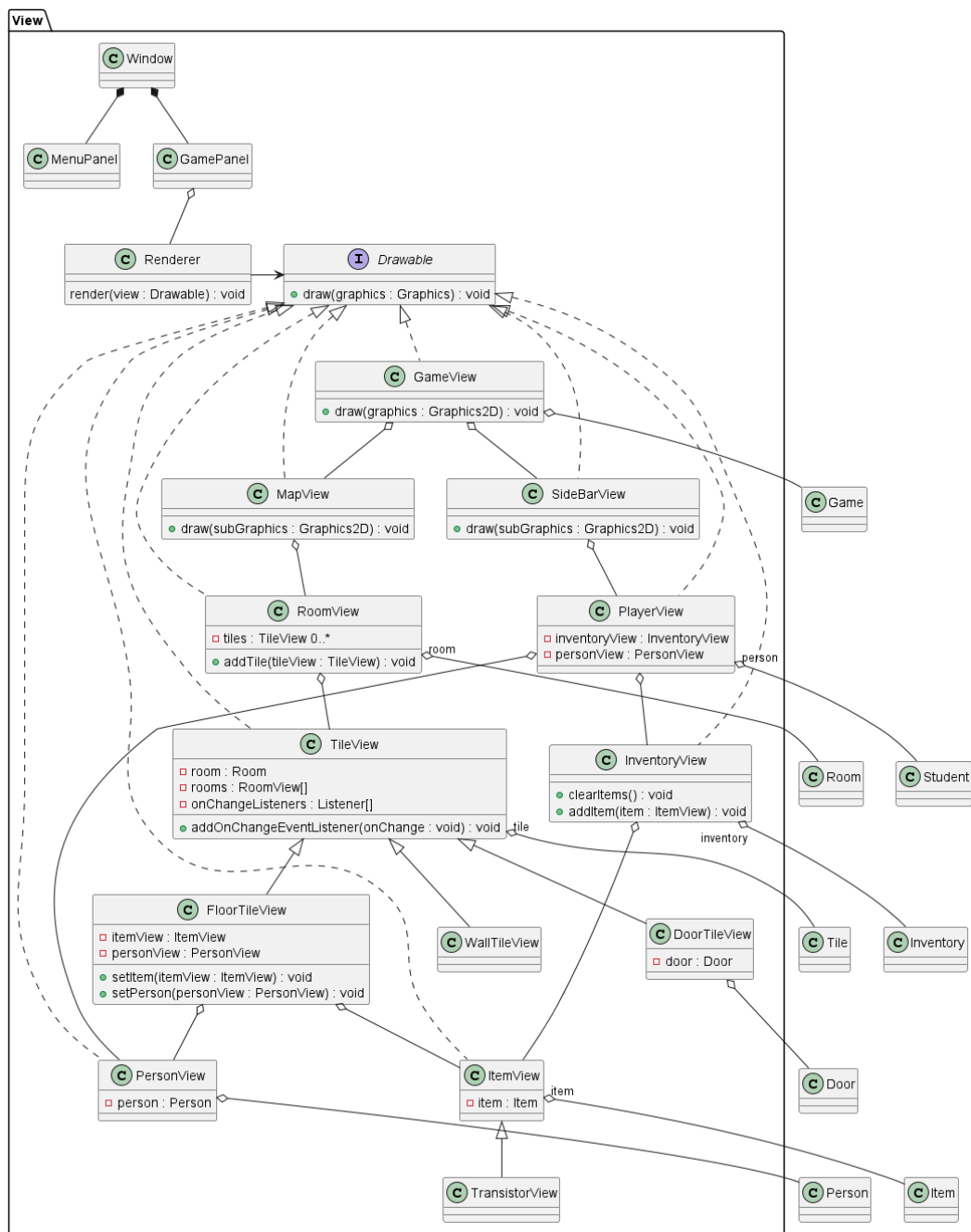
11.2.1. A felület működési elve

A játékunkhoz az MVC mintát választottuk. 3 részre bontjuk a programkódot. A Modellre, mely kezeli az alkalmazás logikai részét. A Nézetre, mely megjeleníti a különböző objektumokat. A Vezérlőre, mely a felhasználói bementeket kezeli. A modellt kevert módon implementáljuk. Az MVC modellt a számos előnye miatt választottuk. Ilyen az egyidejű fejlesztés lehetősége, a rétegek függetlensége és a könnyen változtathatóság. A hátrányait a mintának is érdemes megjegyezni, de a gyengébb olvashatóság és a több boilerplate kód ellenére is megérte ezt a mintát választanunk.

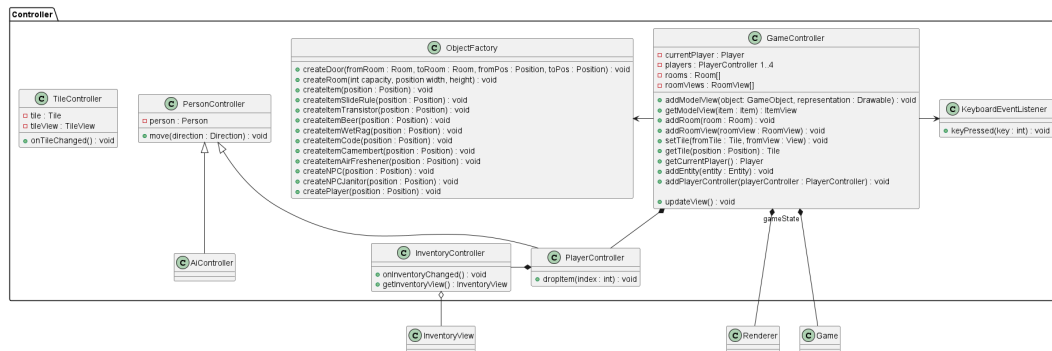
11.2.2. A felület osztály-struktúrája



11.3. ábra. Modell



11.4. ábra. View



11.5. ábra. Controller

11.3. A grafikus objektumok felsorolása

11.3.0.1. AIController

■ Felelősség

Az oktatók és takarítók irányítása

■ Interfészek

-

■ Ősosztály

PersonController

■ Attribútumok

◇ *entity* : *Entity* - Az oktató/takarítót reprezentáló entitás.

■ Metódusok

◇ *turn()* : *void* - Az oktató/takarító van soron, random irányba megpróbál lépni.

11.3.0.2. DoorTile

■ Felelősség

Az ajtó reprezentálása.

■ Interfészek

-

■ Ősosztály

Tile

■ Attribútumok

◇ *door* : *Door* - Az pálya abszrakt reprezentációjában hozzá tartozó ajtó.

■ Metódusok

◇ *stepOn(person : Person) : boolean* - Valaki megpróbál átlépni az ajtón.

11.3.0.3. DoorTileView

■ Felelősség

Egy ajtó kirajzolása.

■ Interfészek

Drawable

■ Ősosztály

-

■ Attribútumok

◇ *doorTile : DoorTile* - A nézethez tartozó DoorTile objektum.

■ Metódusok

-

11.3.0.4. Drawable

■ Felelősség

Interfész biztosítása a kirajzolható objektumoknak.

■ Metódusok

◇ *draw(graphics : Graphics) : void* - Az objektum kirajzolása az adott grafikus kontextusban.

11.3.0.5. Entity

■ Felelősség

A játékban szereplő embereknek ad egy wrapper-t, ami tárolja a pozícióját.

■ Interfészek

-

■ Ősosztály

-

■ Attribútumok

◇ *position : Position* - Az entitás x,y koordinátái világ térben

■ Metódusok

- ◇ *getPerson()* : *person* - Az entitás által reprezentált abszrakt modell beli ember lekérdezése.
- ◇ *getPosition()* : *Position* - Az entitás helyzetének lekérdezése.

11.3.0.6. FloorTile

■ Felelősség

Szobák padlóit reprezentáló csempe.

■ Interfészek

-

■ Össosztály

Tile

■ Attribútumok

- ◇ *item* : *Item* - A csempén található elejtett tárgy, ha van.
- ◇ *person* : *Person* - A csempén álló személy, ha van.
- ◇ *onChangeListeners* : *Listener 0..** - Az inventory megváltozását figyelő eseménykezelők.

■ Metódusok

- ◇ *stepOn(person)* : *boolean* - Egy ember erre a mezőre lép. Igazat ad vissza, ha sikeres (nem állt ott más).
- ◇ *removePerson(person)* : *void* - Az ember elhagyta a mezőt.
- ◇ *setItem(item)* : *void* - A tárgy hozzáadása a cellához.
- ◇ *addOnChangeEventEventListener(onChange)* : *void* - A Tile változásának figyelése

11.3.0.7. FloorTileView

■ Felelősség

Egy padlóelem kirajzolása.

■ Interfészek

Drawable

■ Össosztály

-

■ Attribútumok

- ◇ *floorTile* : *FloorTile* - A mező, amit ez a nézet reprezentál.
- ◇ *itemView* : *ItemView* - A cellán lévő tárgy nézete (ha van).
- ◇ *personView* : *PersonView* - A cellán lévő személy nézete (ha van).

■ Metódusok

- ◇ *setItem(itemView : ItemView): void* - A mezőn lévő tárgy nézetének beállítása.
- ◇ *setPerson(personView : PersonView): void* - A mezőn álló ember nézetének beállítása.

11.3.0.8. Game

■ Felelősség

A játékbeli pálya és játékosok tárolása.

■ Interfészek

-

■ Össosztály

-

■ Attribútumok

- ◇ *tiles : Tile 0..** - A játékteret alkotó mezők.
- ◇ *rooms : Room 2..** - A játékteret alkotó szobák.

■ Metódusok

- ◇ -

11.3.0.9. GameController

■ Felelősség

A játék központi vezérlője, ismeri a játékállapotot, a megjelenítőket, illetve az alacsonyabb szintű vezérlőket, és ezek kapcsolatát.

■ Interfészek

-

■ Össosztály

-

■ Attribútumok

- ◇ *players : PlayerController 1..4* - Játékosok vezérlői
- ◇ *npcs : AiController 1..** - Oktatók és takarítók vezérlői
- ◇ *modelViews : Map<Object, Drawable>* - A modellbeli objektumok és őket reprezentáló nézetek kapcsolatát tárolja.
- ◇ *renderer : Renderer* - Kirajzolja a játékot.
- ◇ *gamePanel : GamePanel* - A kirenderelt játékot megjelenítő panel.

■ Metódusok

- ◇ *addModelView(object : Object, view : Drawable) : void* - Egy modell objektum és hozzátartozó nézet összerendelése.
- ◇ *getModelView(object : Object) : Drawable* - Az objektum nézetének lekérdezése.
- ◇ *addRoom(room : Room) : void* - Egy szoba hozzáadása a kontrollerhez.
- ◇ *addRoomView(roomView) : void* - Egy szoba nézetének hozzáadása a kontrollerhez.
- ◇ *setTile(tile : Tile, position : Position) : void* - A pályán a megadott pozícióban elhelyez egy cellát.
- ◇ *getTile(position : Position) : Tile* - A megadott pozícióban lévő csempe lekérdezése.
- ◇ *getCurrentPlayer() : void* - A soron következő játékos lekérdezése.
- ◇ *addEntity(Entity) : void* - Entitás hozzáadása a kontrollerhez.
- ◇ *addPlayerController(controller : PlayerController) : void* - Játékos-kontroller hozzáadása a kontrollerhez.
- ◇ *addAiController(controller : AiController) : void* - Oktató / takarító kontrollerének regisztrálása.

11.3.0.10. GamePanel

■ Felelősség

A játékot megjelenítő panel.

■ Interfészek

-

■ Össztály

JPanel

■ Attribútumok

- ◇ *renderer : Renderer* - A nézeteket kirajzoló objektum.

■ Metódusok

- ◇ *repaint() : void* - A renderer által rajzolt képet jeleníti meg teljes méretében.

11.3.0.11. GameView

■ Felelősség

A pálya és a különböző felhasználói felületek kirajzolása.

■ Interfészek

Drawable

■ Össztály

-

■ Attribútumok

- ◇ *game* : *Game* - A játékállapot.
- ◇ *mapView* : *MapView* - A pálya nézete
- ◇ *sideBarView* : *SideBarView* - Az oldalsó panel nézete

■ Metódusok

- ◇ *draw(graphics)* : *void* - Kirajzolás.

11.3.0.12. InventoryController**■ Felelősség**

Egy játékos tárgyainak nyilvántartása, és a változtatások kezelése.

■ Interfészek

-

■ Ősosztály

-

■ Attribútumok

- ◇ *inventoryView* : *InventoryView* - Az inventory nézete

■ Metódusok

- ◇ *onInventoryChanged(inventory)* : *void* - Az Inventory változásának kezelése.

11.3.0.13. InventoryView**■ Felelősség**

Egy játékos tárgyainak megjelenítése.

■ Interfészek

Drawable

■ Ősosztály

-

■ Attribútumok

-

■ Metódusok

- ◇ *clearItems()* : *void* - Tárgynézetek törlése.
- ◇ *addItem(itemView)* : *void* - Egy új tárgy megjelenítése az inventoryban.

11.3.0.14. ItemView

- Felelősség

Egy tárgy kirajzolása.

- Interfészek

Drawable

- Össztály

-

- Attribútumok

- ◇ *item* : *Item* - A tárgy, amihez a nézet tartozik.

- ◇ *sprite* : *Sprite* - A tárgyat megjelenítő kép.

- Metódusok

- ◇ -

11.3.0.15. KeyboardEventListener

- Felelősség

A billentyűzet felügyelete.

- Interfészek

-

- Össztály

-

- Attribútumok

-

- Metódusok

- ◇ *keyPressed(key)* : *void* - A megfelelő esemény végrehajtása a billentyű lenyomásának hatására.

11.3.0.16. MapView

- Felelősség

A pálya kirajzolása.

- Interfészek

Drawable

■ Össztály

-

■ Attribútumok

◇ *roomViews* : *RoomView* 0..* - A pályát alkotó szobák nézetei.

■ Metódusok

◇ *draw(graphics : Graphics) : void* - Kirajzolás

11.3.0.17. PersonController

■ Felelősség

Össztály biztosítása a PlayerControllernek és az AIControllernek.

■ Interfészek

-

■ Össztály

-

■ Attribútumok

-

■ Metódusok

-

11.3.0.18. PersonView

■ Felelősség

Egy személy kirajzolása.

■ Interfészek

Drawable

■ Össztály

-

■ Attribútumok

◇ *person* : *Person* - A nézethez tartozó személy objektum.

◇ *sprite* : *Sprite* - Az embernek megfelelő kép, amivel kirajzolódik.

- Metódusok

-

11.3.0.19. PlayerController

- Felelősség

A játékos irányítása.

- Interfészek

-

- Ősosztály

PersonController

- Attribútumok

- ◇ *inventoryController* : *InventoryController* - A játékos eszköztárának vezérlője

- Metódusok

- ◇ *move(direction)* : *void* - A játékos elmozdítása a megfelelő irányba.

11.3.0.20. PlayerView

- Felelősség

Egy játékos kirajzolása az oldalsó panelre.

- Interfészek

Drawable

- Ősosztály

-

- Attribútumok

- ◇ *inventoryView* : *InventoryView* - A játékos inventoryjának nézete.

- ◇ *personView* : *PersonView* - A játékoshoz tartozó kinézet.

- Metódusok

- ◇ -

11.3.0.21. RoomView

- Felelősség

A pálya kirajzolása.

■ Interfészek

Drawable

■ Ősosztály

-

■ Attribútumok

◇ *tiles* : *TileView* 0..* - A szobát alkotó mezők nézetei.

◇ *room* : *Room* - A nézethez tartozó modellbeli objektum.

■ Metódusok

◇ *addTile(tileView)* : *void* - Egy új cella hozzáadása a szobához.

11.3.0.22. SideBarView

■ Felelősség

Az oldalsó panel kirajzolása.

■ Interfészek

Drawable

■ Ősosztály

-

■ Attribútumok

-

■ Metódusok

◇ *draw(graphics Graphics)* : *void* - Kirajolás

11.3.0.23. Tile

■ Felelősség

Absztrakt osztály biztosítása a csempék számára. Az absztrakt osztály megvalósítja a csempe helyzete számontartásához szükséges funkciót.

■ Interfészek

-

■ Ősosztály

-

■ Attribútumok

- ◇ *room* : *Room* - A szoba, amihez a mező tartozik.
- ◇ *onChangeListeners* : *Listener 0..** - A mezőt figyelő eseménykezelők.

■ Metódusok

- ◇ *stepOn(person)* : *boolean* - A személy megpróbál rálépni a csempére. Ha már van rajta személy, akkor ez nem sikerül. A visszatérés a csempe, amin ezután lesz.
- ◇ *addOnChangeListener(onChange)* : *void* - A Tile változásának figyelése

11.3.0.24. TileController

■ Felelősség

A játék csempéinek a kezelése.

■ Interfészek

-

■ Ősosztály

-

■ Attribútumok

- ◇ *tile* : *Tile* - A hozzátartozó cella.
- ◇ *tileView* : *TileView* - A hozzátartozó cella nézete.

■ Metódusok

- ◇ *onTileChanged()* : *void* - A csempe változásának a kezelése.

11.3.0.25. TileView

■ Felelősség

Egy mező kirajzolása.

■ Interfészek

Drawable

■ Ősosztály

-

■ Attribútumok

- ◇ *tile* : *Tile* - A mező, amit ez a nézet reprezentál.

■ Metódusok

-

11.3.0.26. TransistorView

■ Felelősség

Egy tranzisztor kirajzolása. Azért van ennél az egy tárgynál külön szükség rá, mivel míg a többi tárgynál elég a sprite-ot létrehozásnál megadni, addig a tranzisztor állapotától függően az változhat.

■ Interfészek

Drawable

■ Ősosztály

ItemView

■ Attribútumok

◇ *transistor* : *Transistor* - A tranzisztor, amit megjelenít.

■ Metódusok

◇ *draw(graphics : Graphics) : void* - Kirajzolja a tranzisztort, megjelenítve azt is, hogy aktív-e.

11.3.0.27. WallTile

■ Felelősség

Vizuálisan is megjeleníthető pályahatárok biztosítása.

■ Interfészek

-

■ Ősosztály

Tile

■ Attribútumok

-

■ Metódusok

◇ *stepOn(person) : boolean* - Hamisat ad vissza, mivel falra nem lehet lépni.

11.3.0.28. WallTileView

■ Felelősség

Egy faldarab kirajzolása.

■ Interfészek

Drawable

■ Ősosztály

-

■ Attribútumok

-

■ Metódusok

-

11.3.0.29. ObjectFactory

■ Felelősség

Factory mintát követő objektum gyártás

■ Interfészek

-

■ Ősosztály

-

■ Attribútumok

-

■ Metódusok

- ◇ *createRoom(int capacity, position width, height)* : *Room* - Létrehoz egy szobát és a benne lévő csempéket: Falcsempék a szélére, padlócssempék a többi helyre. Modellbeli és view-beli objektumokat is gyárt
- ◇ *createDoor(Room fromRoom, Room toRoom, position fromPos, position toPos)* : *Door* - ajtó gyártása: modellbeli ajtó + modellbeli és view-beli csempék.
- ◇ *createPlayer(position position)* : *Student* - játékos gyártása: Student, hozzátartozó entity és view
- ◇ *createNPC(position position)* : *Teacher* - NPC gyártása: Teacher, hozzátartozó entity és view
- ◇ *createNPCJanitor(position position)* : *Janitor* - NPC gyártása: Janitor, hozzátartozó entity és view
- ◇ *createItemSlideRule(position position)* : *SlideRule* - Item + view gyártása
- ◇ *createItemBeer(position position)* : *Beer* - Item + view gyártása
- ◇ *createItemWetRag(position position)* : *WetRag* - Item + view gyártása
- ◇ *createItemCode(position position)* : *CodeOfStudies* - Item + view gyártása
- ◇ *createItemMask(position position)* : *Mask* - Item + view gyártása
- ◇ *createItemCamembert(position position)* : *Camembert* - Item + view gyártása

- ◇ *createItemAirFreshener(position position) : AirFreshener - Item + view gyártása*
- ◇ *createItemTransistor(position position) : Transistor - Item + view gyártása*
- + még egy item gyártása

11.3.1. Megváltozott osztályok

Az alapelvünk az átalakításnál az volt, hogy a lehető legkevesebb dolgot kelljen megváltoztatni a már meglévő osztálystruktúrában, ezért ahol lehetett inkább új osztályokba csomagoltuk a meglévő modell objektumait.

11.3.1.1. Inventory

■ Felelősség

Az entitások által birtokolt tárgyak számontartása a modellben.

■ Interfészek

Updatable

■ Ősosztály

-

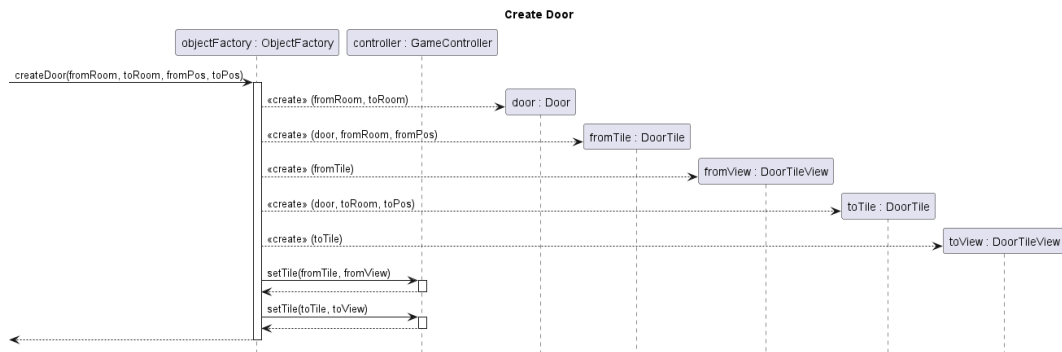
■ Attribútumok

- ◇ *onChangeListeners : Listener 0..** - Az inventory megváltozását figyelő eseménykezelők.

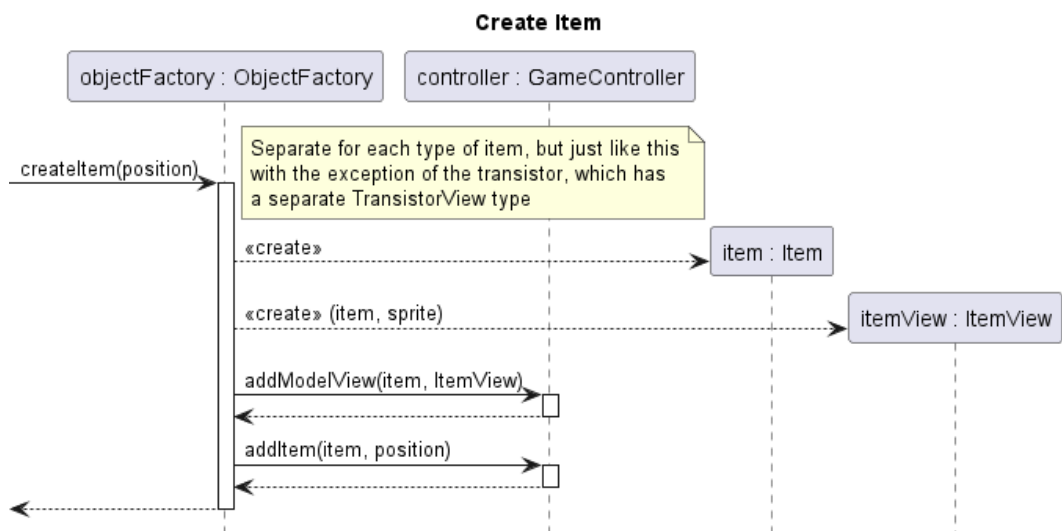
■ Metódusok

- ◇ *addOnChangeEventListener(listener : Listener) : void* - Listener feliratkoztatása az inventory változás event-ére.
- ◇ *onChanged() : void* - Értesíti a feliratkozott eseménykezelőket a változásról.
- ◇ *add(item : Item) : boolean* - A korábbi működésen túl meghívja az *onChanged()* metódust.
- ◇ *remove(item : Item) : void* - A korábbi működésen túl meghívja az *onChanged()* metódust.

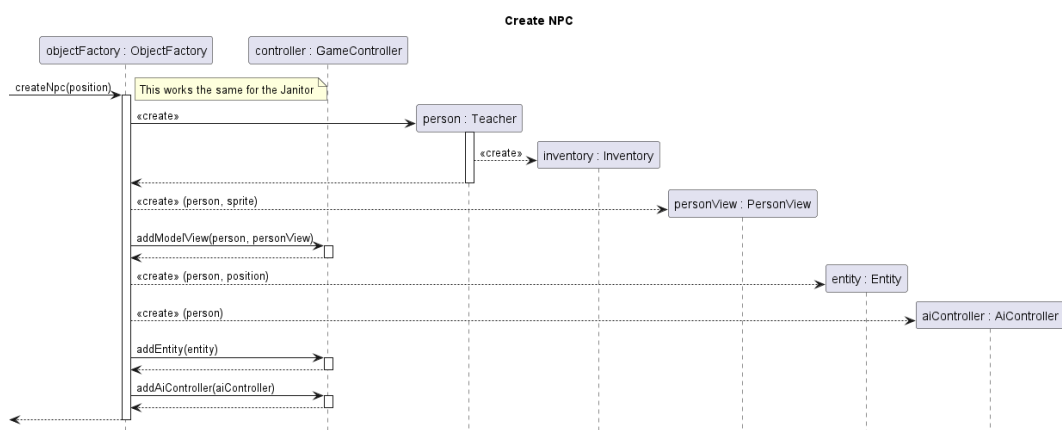
11.4. Kapcsolat az alkalmazói rendszerrel



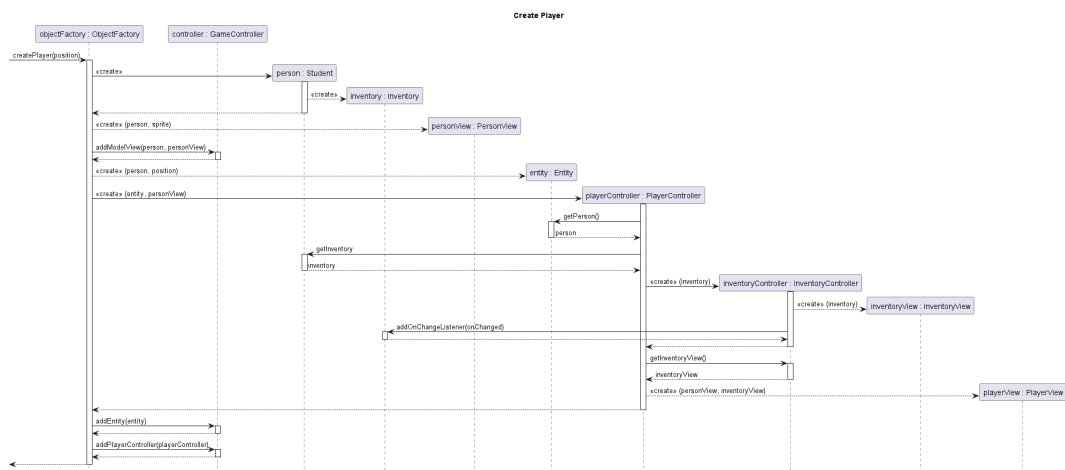
11.6. ábra. Ajtó létrehozása



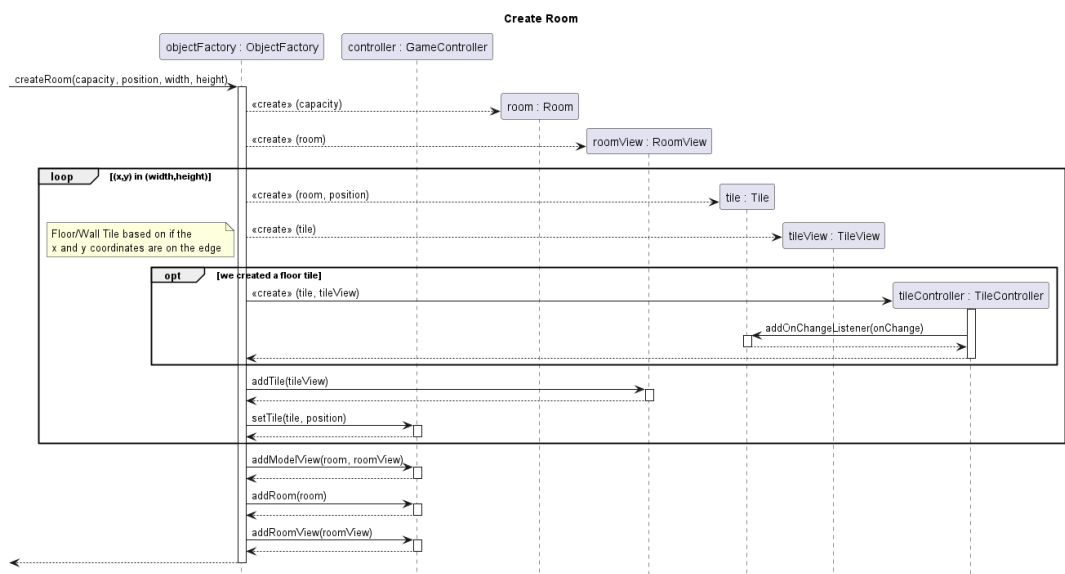
11.7. ábra. Tárgy létrehozása



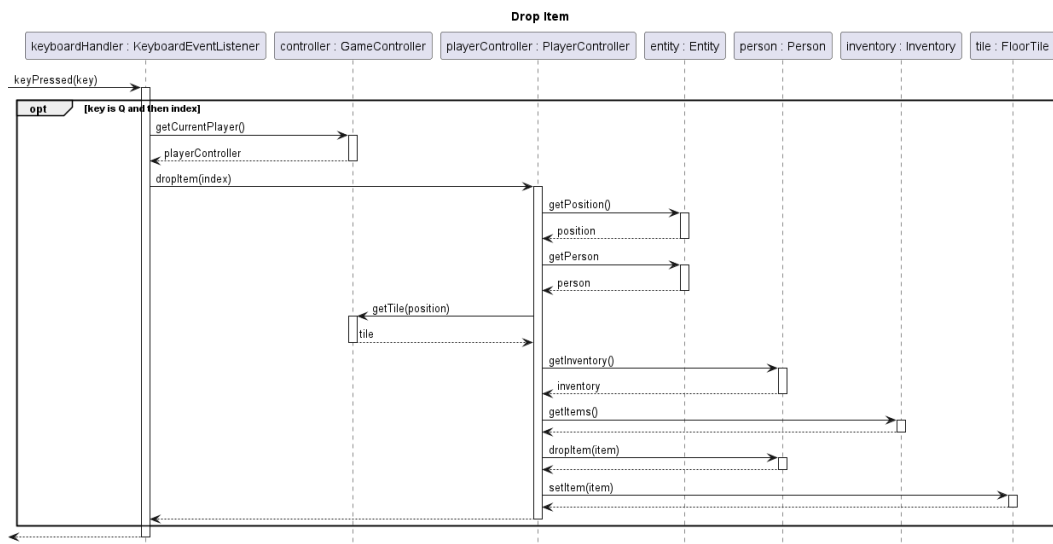
11.8. ábra. NPC létrehozása



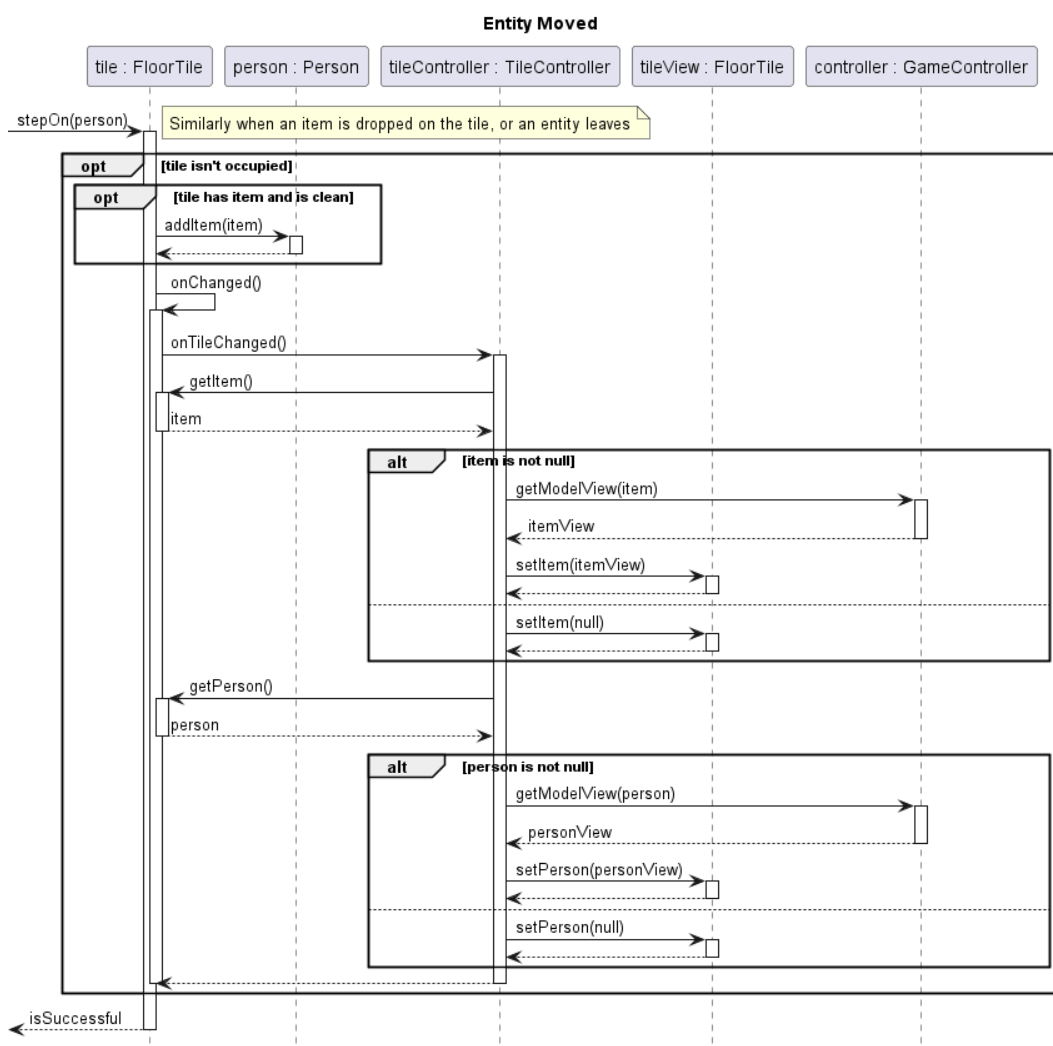
11.9. ábra. Játékos létrehozása



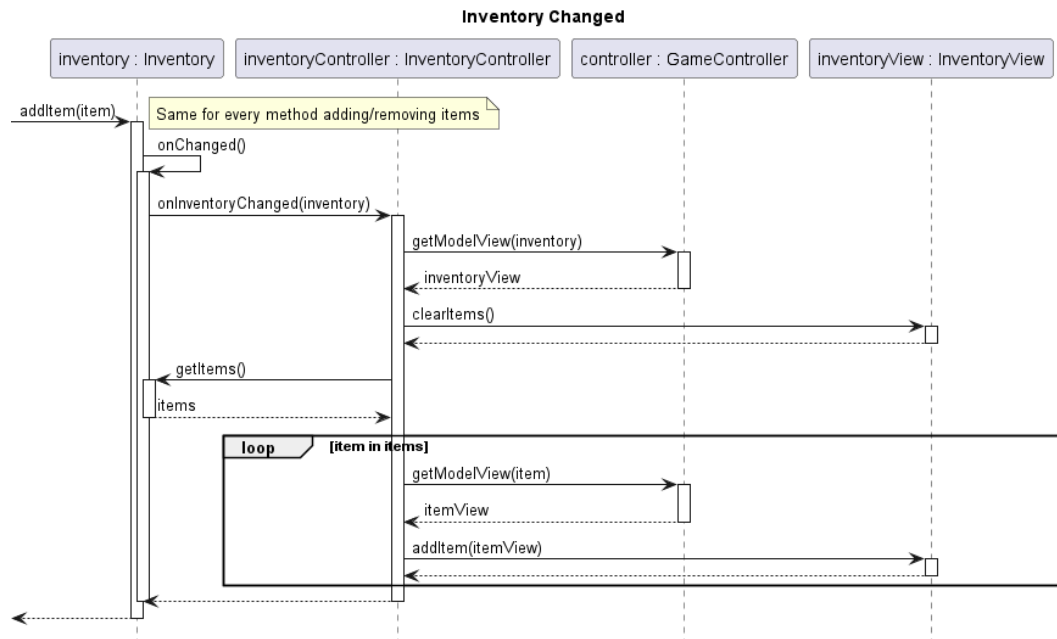
11.10. ábra. Szoba létrehozása



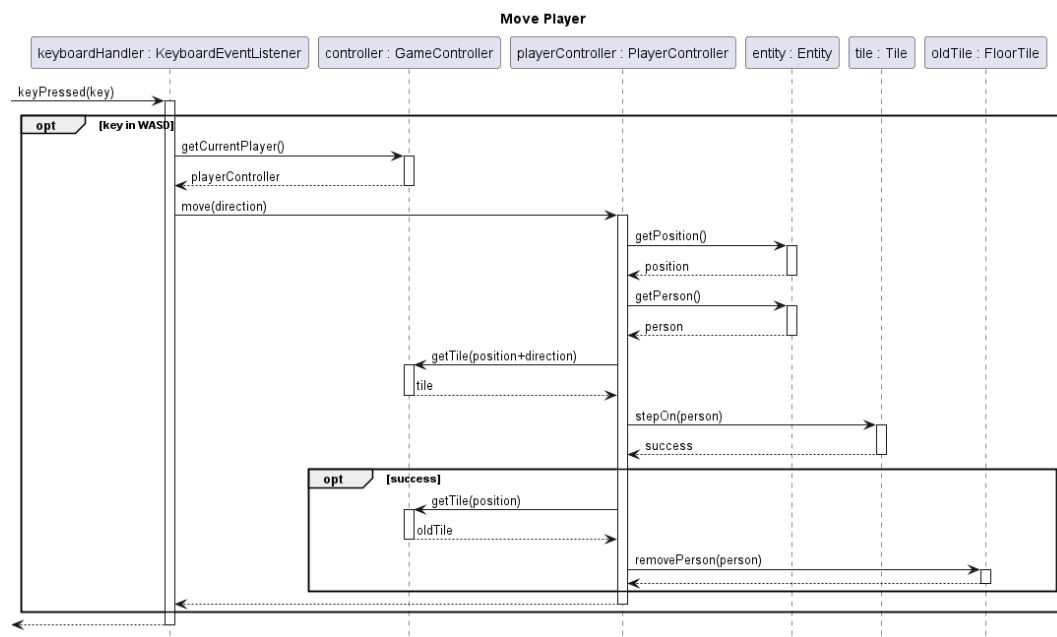
11.11. ábra. Tárgy eldobása



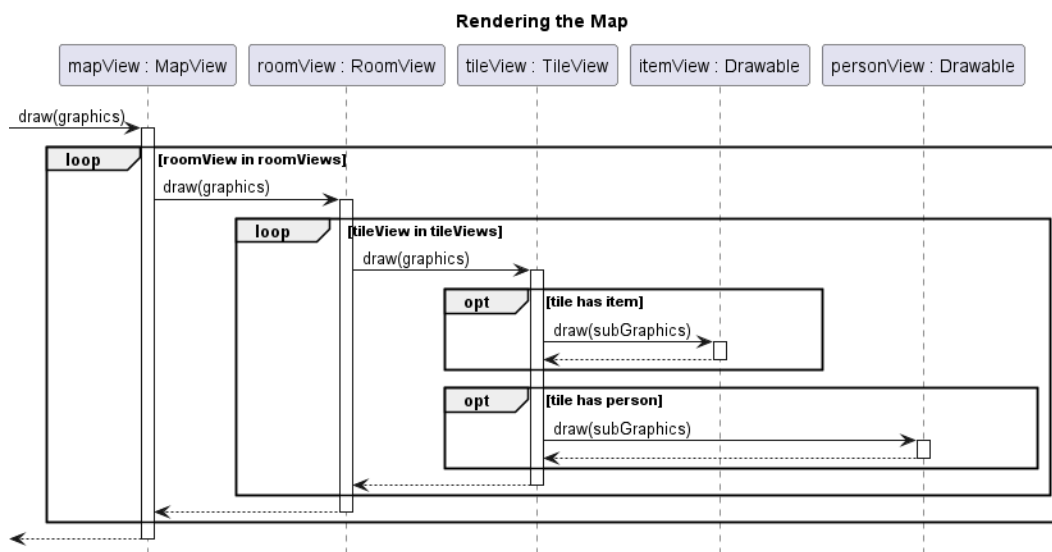
11.12. ábra. Mozgás



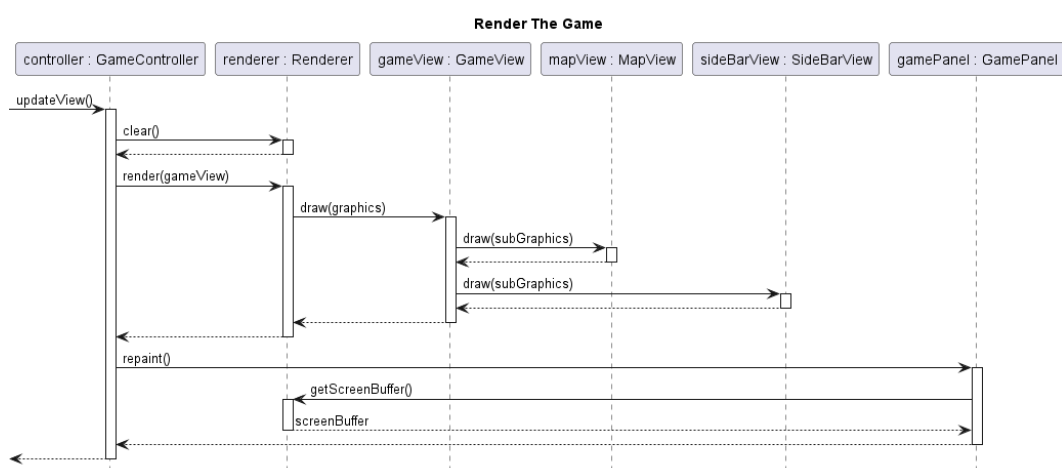
11.13. ábra. Inventory megváltozása



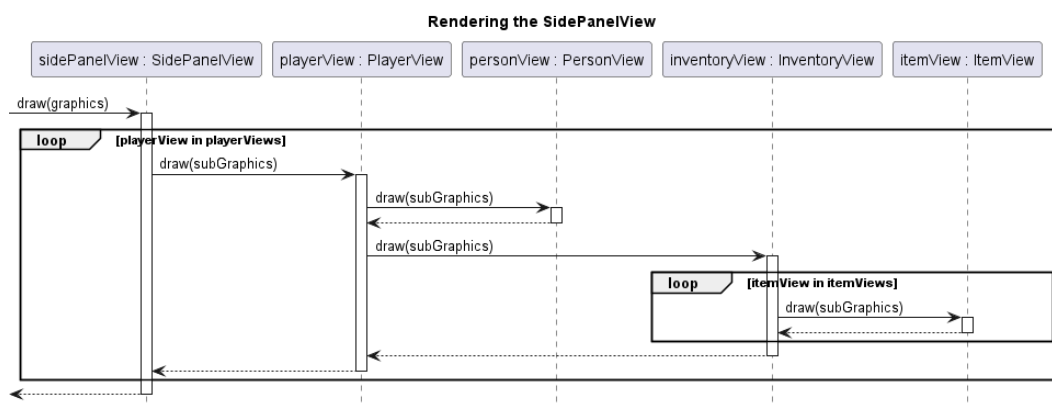
11.14. ábra. Játékos mozgatása



11.15. ábra. Pálya kirajzolása

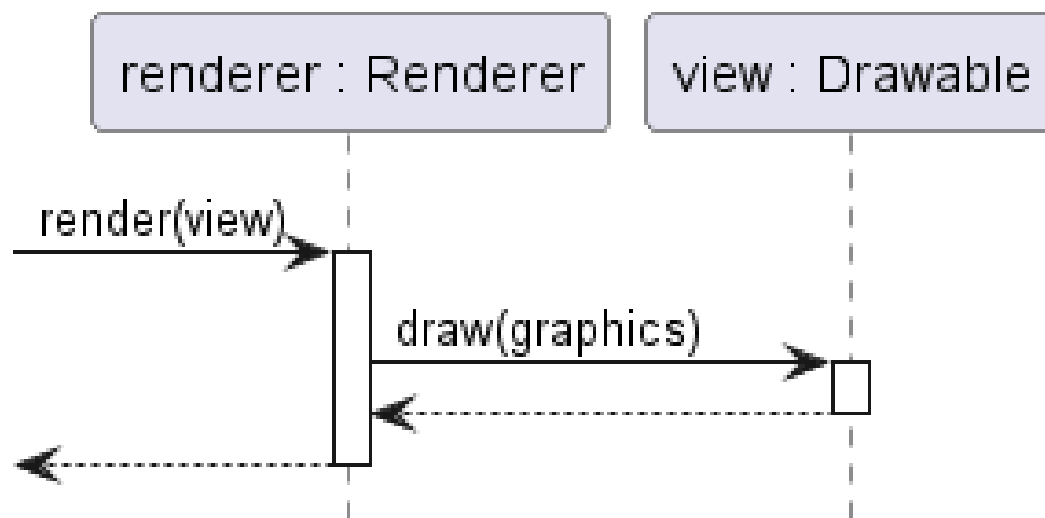


11.16. ábra. Játék kirajzolása



11.17. ábra. Oldalsó panel kirajzolása

Renderer Rendering



11.18. ábra. Renderer működése

11.5. Napló

Kezdet	Időtartam	Résztevők	Leírás
máj. 2. 13:00	3 óra	Tóth	Grafikus elemek tervezése.
máj. 4. 20:00	2 óra	Tóth	MVC megértése, architektúra tervezése.
máj. 5. 10:00	1 óra	Balla	Osztálydiagram ötlet
máj. 5. 14:00	2 óra	Zelei	Főmenü tervezése.
máj. 5. 14:30	1,5 óra	Nagy	Tervezés.
máj. 5. 16:00	5 óra	Joób	Tervezés, osztálydiagram kiegészítés.
máj. 5. 22:00	2 óra	Balla	Osztálydiagram és szekvencia ötlet írása.
máj. 5. 18:00	5 óra	Tóth	MVC architektúrának megfelelő struktúra kialakítása.
máj. 6. 00:00	5 óra	Tóth	Fontosabb szekvencia diagrammok elkészítése, osztály hierarchia finomítása.
máj. 6. 07:00	2 óra	Zelei	Osztályleírások készítése.
máj. 6. 08:30	4,5 óra	Nagy	Osztályleírások javítása, konzisztencia ellenőrzése.
máj. 6. 10:00	1,5 óra	Tóth	Osztályleírások javítása.