

# 11. Grafikus felület specifikációja

## 15 – gizs-qtyusch

Konzulens:  
Goldschmidt Balázs

### Csapattagok

Dancsó Marcell	AZTVS7	dancsomarci@gmail.com
Tóth András	FKMV9M	4ndras21@gmail.com
Bajnok Vencel	X748Q2	bajnokvencel@edu.bme.hu
Róna Balázs Ferenc	HBTGON	rona.balazs@gmail.com
Pelczhoffer Tamás	Z5KWH6	pelczhoffer.tamas@icloud.com

2022.05.02.

## 11. Grafikus felület specifikációja

### 11.1A grafikus interfész



**Szabad terület mező:**



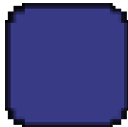
**Óvóhely mező:**



**Raktár mező:**



*Laboratórium mező:*



Felszerelés helye:



Fejsze felszerelés:



Kesztyű felszerelés:



Köpeny felszerelés:



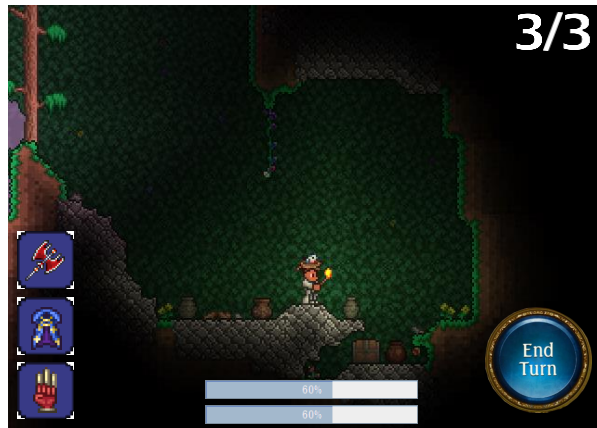
Zsák felszerelés:



Kör befejezése gomb:

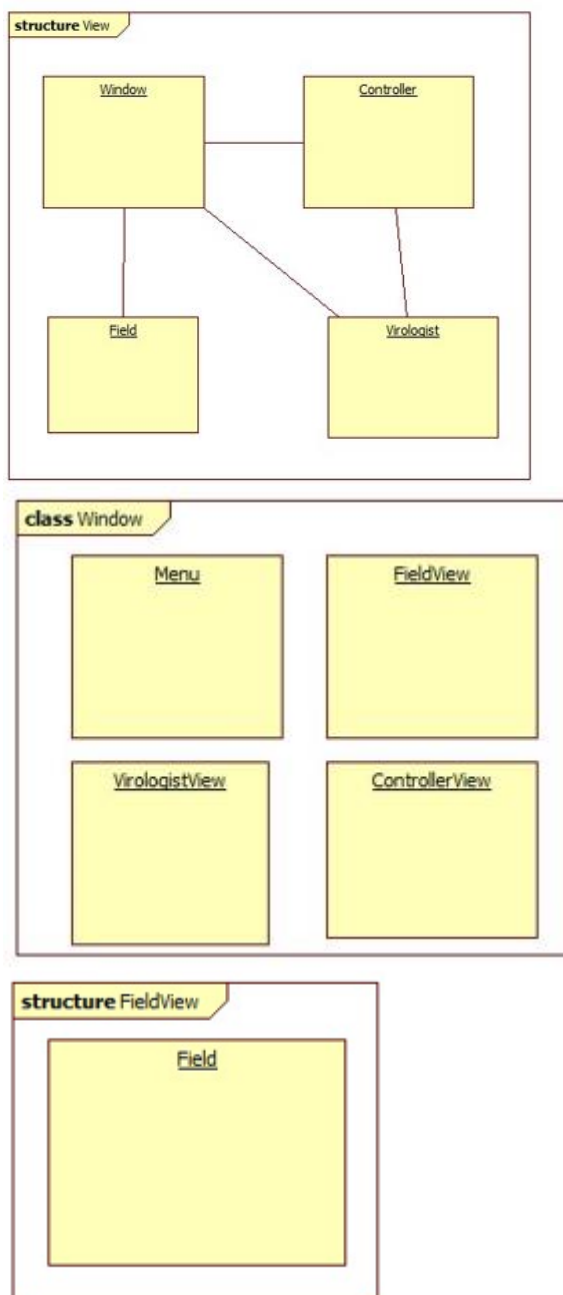


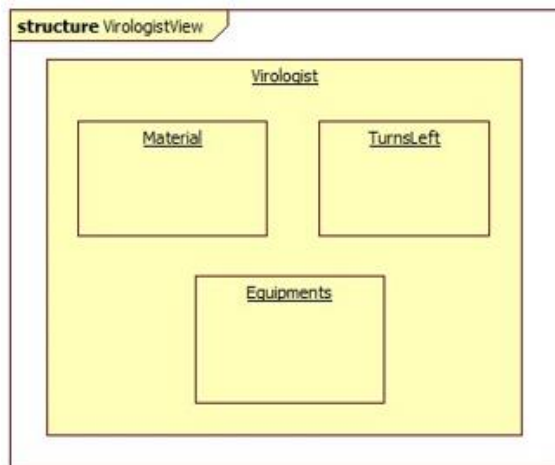
Szöveges értesítés cselekvésekről:



*Egy konkrét példa:*

## 11.2A grafikus rendszer architektúrája





### 11.2.1 A felület működési elve

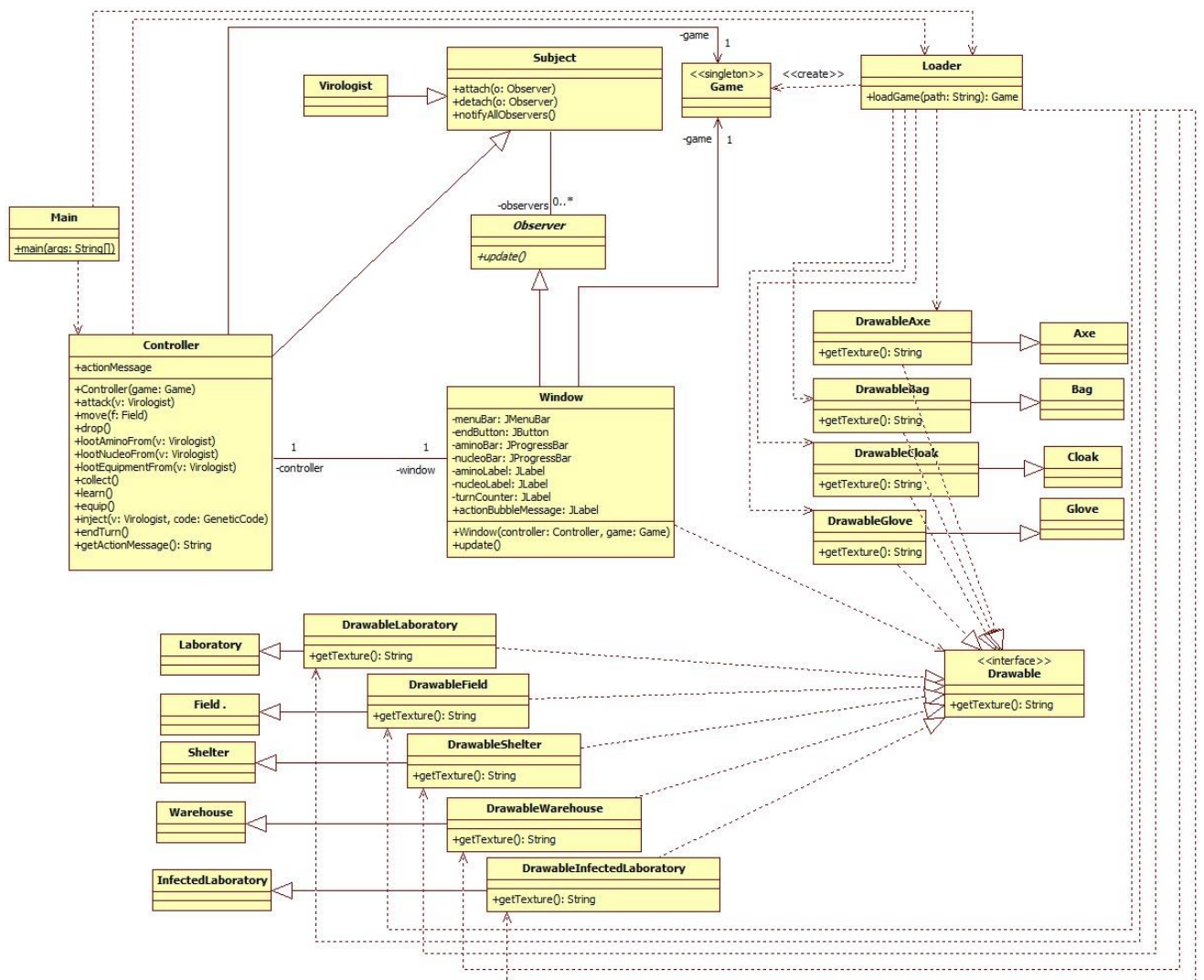
A felület kialakítása az MVC tervezési minta alapján készült.

A **modell**-hez tartoznak az analízis modellben kialakított osztályok, kiegészülve a Subject-el. A virológus öröklí ezt a Subject osztályt. Ez annyi változást jelent, hogy amikor a belső állapota megváltozik (eldob valamit, vagy csökken az anyagkészlete stb...) értesíti a megfigyelőket, akik rá vannak csatlakozva. A megfigyelő a View lesz, aki értesítés során újrarajzolódik.

A **view** egy Observer absztrakt osztályból, és az őt realizáló Window osztályból áll. A Window dolga a kirajzolás/ a java swing elemek megfelelő kezelése. A kirajzoláshoz a modellben lévő adatokra szüksége lehet, így azt lekérdezheti.

A **controller** a Controller osztály testesíti meg. Ugyan a bemenet a View-n keresztül érkezik, mert swing-ben a felhasználói bemenetet az ablak kezeli, de a konkrét funkcionalitás az adott bemenetre a Controller felséterelete, így az összes bemenet delegálódik neki. Ezen bemenetek hatására a modellt megfelelően irányítja. Képes továbbá a View-nak szólni, ha frissülnie kell. A pálya inicializálásáért is a controller felel, azon belül is a Loader osztály, ami sima Field, és Equipment típusok helyett a Drawable változatukat ad át a game-nek. Ez azért kell, hogy a textúrákat a modell változtatása nélkül tudjuk objektumorientáltan tárolni.

## 11.2.2 A felület osztály-struktúrája



## 11.3A grafikus objektumok felsorolása

### 11.3.1 Main

- **Felelősség**

*Az alkalmazás belépési pontja.*

- **Metódusok**

- **+void main(args: String[]):** Az alkalmazás belépési pontja, ami inicializál mindent. Parancssori argumentumként kapja a pályabetöltő file path-jét, aminek tartalma a prototípusban megjelent pályaleíró nyelv szintaxisával egyezik meg.  
load the game using Loader, and args  
initialize Controller with game

### 11.3.2 Drawable

- **Felelősség**

*Textúrainformációt tartalmazó osztályok interface-e.*

- **Metódusok**

- **+String getTexture():** Visszaadja a textúrainformációt String formátumban.

### 11.3.3 DrawableLaboratory

- **Felelősség**

*Olyan labor, aminek van textúrainformációja.*

- **Ősosztályok**

*Laboratory*

- **Interfészek**

*Drawable*

- **Metódusok**

- **+String getTexture():** Visszaadja a laborhoz tartozó textúraelérhetőségét.

### 11.3.4 DrawableField

- **Felelősség**

*Olyan field, aminek van textúrainformációja.*

- **Ősosztályok**

*Field*

- **Interfészek**

*Drawable*

- **Metódusok**

- **+String getTexture():** Visszadja a field-hez tartozó textúraelérhetőségét.

### 11.3.5 DrawableShelter

- **Felelősség**

*Olyan shelter, aminek van textúrainformációja.*

- **Ósosztályok**

*Shelter*

- **Interfészek**

*Drawable*

- **Metódusok**

- **+String getTexture():** Visszadja a shelterhez tartozó textúraelérhetőségét.

### 11.3.6 DrawableWarehouse

- **Felelősség**

*Olyan warehouse, aminek van textúrainformációja.*

- **Ósosztályok**

*Warehouse*

- **Interfészek**

*Drawable*

- **Metódusok**

- **+String getTexture():** Visszadja a warehousehoz tartozó textúraelérhetőségét.

### 11.3.7 DrawableInfectedLaboratory

- **Felelősség**

*Olyan fertőzőttlabor, aminek van textúrainformációja.*

- **Ósosztályok**

*InfectedLaboratory*

- **Interfészek**

*Drawable*

- **Metódusok**

- **+String getTexture():** Visszadja a fertőzőttlaborhoz tartozó textúraelérhetőségét.



### 11.3.8 DrawableAxe

- **Felelősség**

*Olyan balta, aminek van textúrainformációja.*

- **Ősosztályok**

*Axe*

- **Interfészek**

*Drawable*

- **Metódusok**

- **+String getTexture():** Visszaadja a baltához tartozó textúraelérhetőséget.

### 11.3.9 DrawableBag

- **Felelősség**

*Olyan bag, aminek van textúrainformációja.*

- **Ősosztályok**

*Bag*

- **Interfészek**

*Drawable*

- **Metódusok**

- **+String getTexture():** Visszaadja a bag-hez tartozó textúraelérhetőséget.

### 11.3.10 DrawableCloak

- **Felelősség**

*Olyan köpeny, aminek van textúrainformációja.*

- **Ősosztályok**

*Cloak*

- **Interfészek**

*Drawable*

- **Metódusok**

- **+String getTexture():** Visszaadja a köpenyhez tartozó textúraelérhetőséget.

### 11.3.11 DrawableGlove

- **Felelősség**

*Olyan kesztyű, aminek van textúrainformációja.*

- **Össztályok**

*Glove*

- **Interfészek**

*Drawable*

- **Metódusok**

- **+String getTexture():** *Visszaadja a kesztyűhöz tartozó textúraelérhetőséget.*

### 11.3.12 Loader

- **Felelősség**

Felelőssége inicializálni a modellt, egy kapott elérési útvonalon lévő „.txt” kiterjesztésű fileból, amiben a játék leírása található, a Prototípusban definiált pályaleíró nyelv formátumában. (Mivel a Game singleton mintát követ, elég inicializálnia, és onnantól mindenki ugyanazt a „modellt” éri majd el.) Lényegében ő felel a pályaleíró nyelvért, és ha más formátumot akarunk használni elég ezt az osztályt lecserélni.

- **Metódusok**

- **+void loadGame(path: String): Game**

```
//command is the starting none whitespace line
//or the first none whitespace line after the "end" line
//line: sequence of characters with an new line character at the end
For every command in file:
    If command is "Field":
        initialize new DrawableField of the drawable version of given Type
        with given Param
        drop drawable version of given equipment on newly created Field if
        present
    ElseIf command is "Neighbours":
        set neighbouring relations between given Field pairs separated by
        a whitespace character
    ElseIf command is "Virologist":
        initialize a Virologist
        set Name of the Virologist with given Name
        set ActionCount with given ActionCount
        set Equipments with given Equipment
        set Nucleotide level with given Nucleo
        set AminoAcid level with given Amino
        set agents with given agents
        set codes with given GeneticCode
        place the Virologist on given StartingPos
    End
End
return the newly initialized game
//For details on input format possibilities and a deeper knowledge on
syntax look at the documentation
```

### 11.3.13 Controller

- **Felelősség**

Felelőssége a View-ből érkező felhasználói input mögöttes logikájának kezelése. Utóbbi hatására a modell megfelelő irányítása, valamint képes a View-nak információval szolgálni, hogy visszajelzést tudjon adni a felhasználónak, miként kezelődik a modell.

- **Őszosztályok**

Subject

- **Attribútumok**

- **-actionMessage: String** Ebben a változóban tárolódik a View-nak szánt üzenete.
- **-game: Game** Az irányított játék referenciája.

- **Metódusok**

- **+void Controller(game: Game):** Inicializálja a game tagváltozót.
- **+void attack(v: Virologist):** Az aktuálisan soron lévő játékosnak delegálja a hívást.  
 get currentPlayer from game  
 delegate call to currentPlayer  
 set actionMessage to "Attacking" + currentPlayer's name  
 notify window to redraw
- **+void move(f: Field):** Az aktuálisan soron lévő játékosnak delegálja a hívást.  
 get currentPlayer from game  
 delegate call to currentPlayer  
 set actionMessage to "moving from" + currentPlayer's old location to  
 currentPlayer's newLocation  
 notify window to redraw
- **+void drop():** Az aktuálisan soron lévő játékosnak delegálja a hívást.  
 get currentPlayer from game  
 delegate call to currentPlayer  
 set actionMessage to "Dropping one of equipments"  
 notify window to redraw
- **+void lootAminoFrom(v: Virologist):** Az aktuálisan soron lévő játékosnak delegálja a hívást.  
 get currentPlayer from game  
 delegate call to currentPlayer  
 set actionMessage to "looting AminoAcid from" + v's name  
 notify window to redraw
- **+void lootNucleoFrom(v: Virologist):** Az aktuálisan soron lévő játékosnak delegálja a hívást.  
 get currentPlayer from game  
 delegate call to currentPlayer  
 set actionMessage to "looting Nucleotide from" + v's name  
 notify window to redraw
- **+void lootEquipmentFrom(v: Virologist):** Az aktuálisan soron lévő játékosnak delegálja a hívást.  
 get currentPlayer from game  
 delegate call to currentPlayer  
 set actionMessage to "looting equipment from" + v's name  
 notify window to redraw
- **+void collect():** Az aktuálisan soron lévő játékosnak delegálja a hívást.  
 get currentPlayer from game  
 delegate call to currentPlayer

- ```

    set actionMessage to "collecting material"
    notify window to redraw

```
- **+void learn():** Az aktuálisan soron lévő játékosnak delegálja a hívást.  

```

    get currentPlayer from game
    delegate call to currentPlayer
    set actionMessage to "learning magic"
    notify window to redraw

```
  - **+void equip():** Az aktuálisan soron lévő játékosnak delegálja a hívást.  

```

    get currentPlayer from game
    delegate call to currentPlayer
    set actionMessage to "equipping"
    notify window to redraw

```
  - **+void inject(v: Virologist, code: GeneticCode):** Az aktuálisan soron lévő játékosnak delegálja a hívást.  

```

    get currentPlayer from game
    delegate call to currentPlayer
    set actionMessage to "injecting" + v's name with code's name
    notify window to redraw

```
  - **+void endTurn():** Az aktuálisan soron lévő játékosnak delegálja a hívást. Valamint a Window-t, mint observert átcsatolja az aktuális játékosra.  

```

    get currentPlayer from game
    detach window from currentPlayer
    delegate endTurn call to currentPlayer
    get newCurrentPlayer from game
    attach window to newCurrentPlayer
    set actionMessage to „It's finally my turn!...”
    notify window to draw

```
  - **+String getActionMessage():** Visszadja a View-nak szánt acitionMessage üzenetét.

### 11.3.14 Subject

- **Felelősség**

Egy megfigyelt osztályt szimbolizál, amit Observer objektumok megfigyelhetnek, valamint a subject tud nekik jelezni. Felfogható úgy is, hogy azon osztályok, akik ezt öröklík, képesek egy jelezni más objektumoknak, ha egy meghatározott esemény bekövetkezik.

- **Attribútumok**

- **-observers: Observer[0..\*]:** A megfigyelő objektumok tárolódnak itt.

- **Metódusok**

- **+ void attach(o: Observer):** Felcsatol egy paraméterül kapott observert.  

```

    if observer is not attached:
        attach observer
    End

```
- **+ void detach(o: Observer):** Leccsatolja a paraméterül kapott observert.  

```

    if observer is attached:
        detach observer
    End

```
- **+ void notifyAllObservers():**Értesíti az összes felcsatolt observert.  

```

    For all observer in observers:
        update observer
    End

```

### 11.3.15 Observer

- **Felelősség**

Fel lehet csatlakoztatni egy Subject típusú objektumra, ha a Subject értesítést küld az observereinek, akkor erről értesülnek az osztály példányai.

- **Metódusok**

- **+void update():** Ez a függvény fut le ha értesítik az observert.

### 11.3.16 Virologist

- **Felelősség**

Az osztály azzal egészül ki, hogy örökli a Subject-et. Akkor értesíti az observereit, ha változik a belső állapota.

- **Össztályok**

Subject

- **Metódusok**

- **void Move(Field field):** Egy paraméterben kiválasztott szomszédos mezőre viszi a játékost. Ezután értesíti a megfigyelőit.
- **void Drop():** Egy utasítás, ami során egy random tárgyat eldob az aktuális mezőre. Ezután értesíti a megfigyelőit.
- **void LootAminoAcidFrom(Virologist v):** A paraméterül kapott virológustól megpróbál aminosavat zsákmányolni. Ezután értesíti a megfigyelőit.
- **void LootNucleotideFrom(Virologist v):** A paraméterül kapott virológustól megpróbál nukleotidot zsákmányolni. Ezután értesíti a megfigyelőit.
- **void LootEquipmentFrom(Virologist v):** A paraméterül kapott virológustól megpróbál egy tárgyat zsákmányolni. Ezután értesíti a megfigyelőit.
- **void Collect():** Egy utasítás, ami során megpróbál anyagot gyűjteni az adott mezőről. Ezután értesíti a megfigyelőit.
- **void Learn():** Egy utasítás, ami során megpróbál egy genetikai kódot megtanulni az adott mezőről. Ezután értesíti a megfigyelőit.
- **void Equip():** Egy utasítás, ami során megpróbál felszerelést gyűjteni az adott mezőről. Ezután értesíti a megfigyelőit.
- **void AddAgent(Agent a):** Egy Agent-et ad magához, amin persze meghívja az Apply metódust is. Ezután értesíti a megfigyelőit.
- **void AddEquipmen(Equipment e):** Egy Equipment-et ad magához, amin persze meghívja az Apply metódust is. Ezután értesíti a megfigyelőit.
- **void AddGeneticCode(GeneticCode code):** Egy genetikai kódot ad magához. Ezután értesíti a megfigyelőit.
- **void Inject(Virologist v, GeneticCode code):** Az ágensek felkenését indítja el a kiválasztott virológuson, a kiválasztott ágenssel. Ezután értesíti a megfigyelőit.
- **boolean TargetedWith(Agent a):** Az adott virológust ezzel a metódussal lehet megcélozni ágensfelkenéssel, visszatér, hogy sikeres volt-e a felkenődés. Ezután értesíti a megfigyelőit.
- **void StealAminoAcid(Virologist self):** A paraméterül kapott virológus akar aminosavat zsákmányolni az adott másik virológustól. Ezután értesíti a megfigyelőit.

- **void StealNukleotid(Virologist self):** A paraméterül kapott virológus akar nukleotidot zsákmányolni az adott másik virológustól. Ezután értesíti a megfigyelőit.
- **void StealEquipment(Virologist self):** A paraméterül kapott virológus akar felszerelést zsákmányolni az adott másik virológustól. Ezután értesíti a megfigyelőit.
- **void RemoveGeneticCodes():** Törli az összes megtanult genetikai kódot. Ezután értesíti a megfigyelőit.
- **void RemoveAgents():** Törli az összes tárolt Agent-et. Ezután értesíti a megfigyelőit.
- **void DecreaseActions():** A fennmaradt teljesíthető interakciók számát csökkenti. Ezután értesíti a megfigyelőit.
- **void EndTurn():** Befejezi vele a kört a játékos, ha nem szeretne több interakciót végrehajtani. Ezután értesíti a megfigyelőit.
- **void AddAminoAcid(int delta):** A paraméter mértékével növeli a nukleotid mennyiségét. Ezután értesíti a megfigyelőit.
- **void AddNucleotide(int delta):** A paraméter mértékével növeli a nukleotidok mennyiségét. Ezután értesíti a megfigyelőit.
- **void RemoveNucleotide(int delta):** Csökkenti az aminosav mennyiségét a paraméter mértékével. Ezután értesíti a megfigyelőit.
- **void RemoveAminoAcid(int delta):** Csökkenti az aminosav mennyiségét a paraméter mértékével. Ezután értesíti a megfigyelőit.
- **void IncreaseLimit(int delta):** Megnöveli a hordozó kapacitást a paraméter mértékével. Ezután értesíti a megfigyelőit.
- **void DecreaseLimit(int delta):** Lecsökkenti a hordozó kapacitást a paraméter mértékével. Ezután értesíti a megfigyelőit.

### 11.3.17 Window

- **Felelősség**

A view-t reprezentáló osztály, ami az ablakozás kezeléséért felelős. A felhasználói inputot továbbítja a Controller-nek. Ha értesíti valamelyik Subject, akkor újrarajzolódik, amihez információt a Modell-től kér le.

- **Ősosztályok**

Observer

- **Attribútumok**

- **-game: Game:** A hozzá tartozó modell-t éri el rajta keresztül.
- **-controller: Controller:** A hozzá tartozó controllert éri el rajta keresztül.
- **-menuBar: JMenuBar** A felhasználói input felvételéért felelős menü.
- **-endButton: JButton** A kör vége jelzésére alkalmas gomb.
- **-aminoBar: JProgressBar** Aminosav érték kijelzésére szolgáló ui elem.
- **-nucleoBar: JProgressBar** Nukleotid érték kijelzésére szolgáló ui elem.
- **-aminoLabel: JLabel** Aminosav érték kijelzésére szolgáló ui elem felirata.
- **-nucleoLabel: JLabel** Nukleotide érték kijelzésére szolgáló ui elem felirata.
- **-turnCounter: JLabel** Hátralévő körök száma kijelzésére alkalmas UI elem.
- **-actionBubbleMessage: JLabel** Legutóbbi cselekvés kijelzésére alkalmas UI elem.

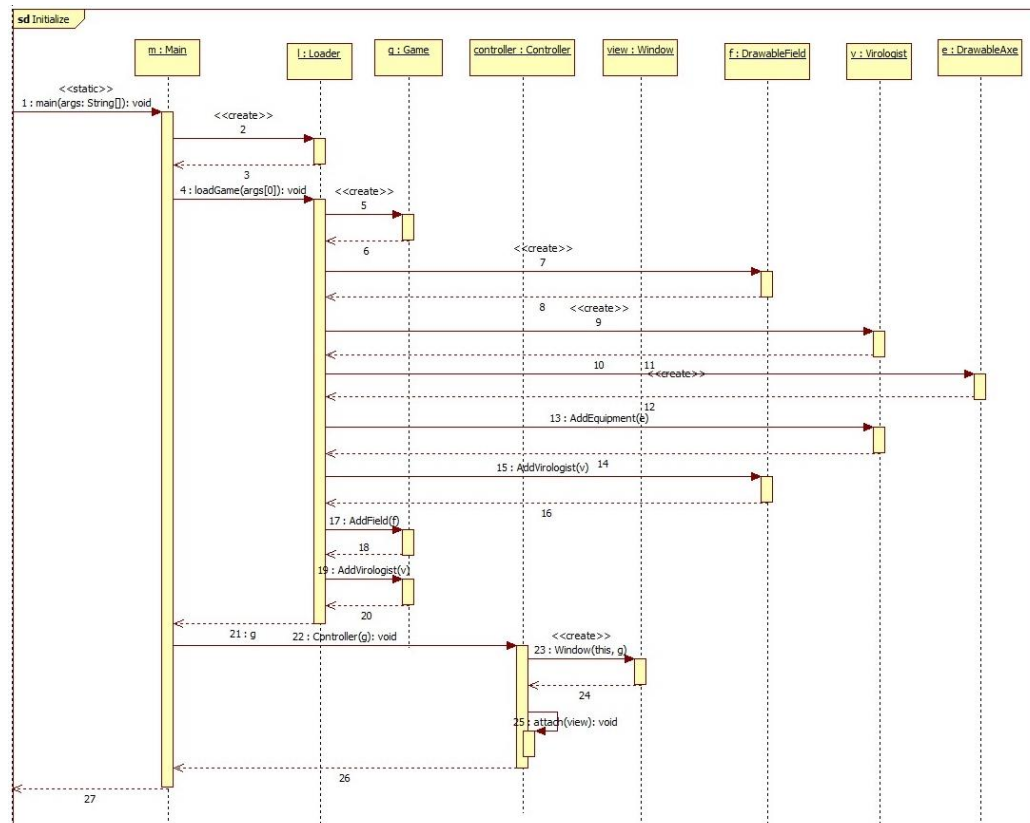
- **Metódusok**

- **+void Window(controller: Controller, game: Game):** Az ablak inicializálása, controller tagváltozó beállítása. Game tagváltozó beállítása. Ablak kirajzolása. (lásd update())
- **+void update():** Újrarajzolja az ablakot, a tagváltozóban tárolt UI elemek segítségével.

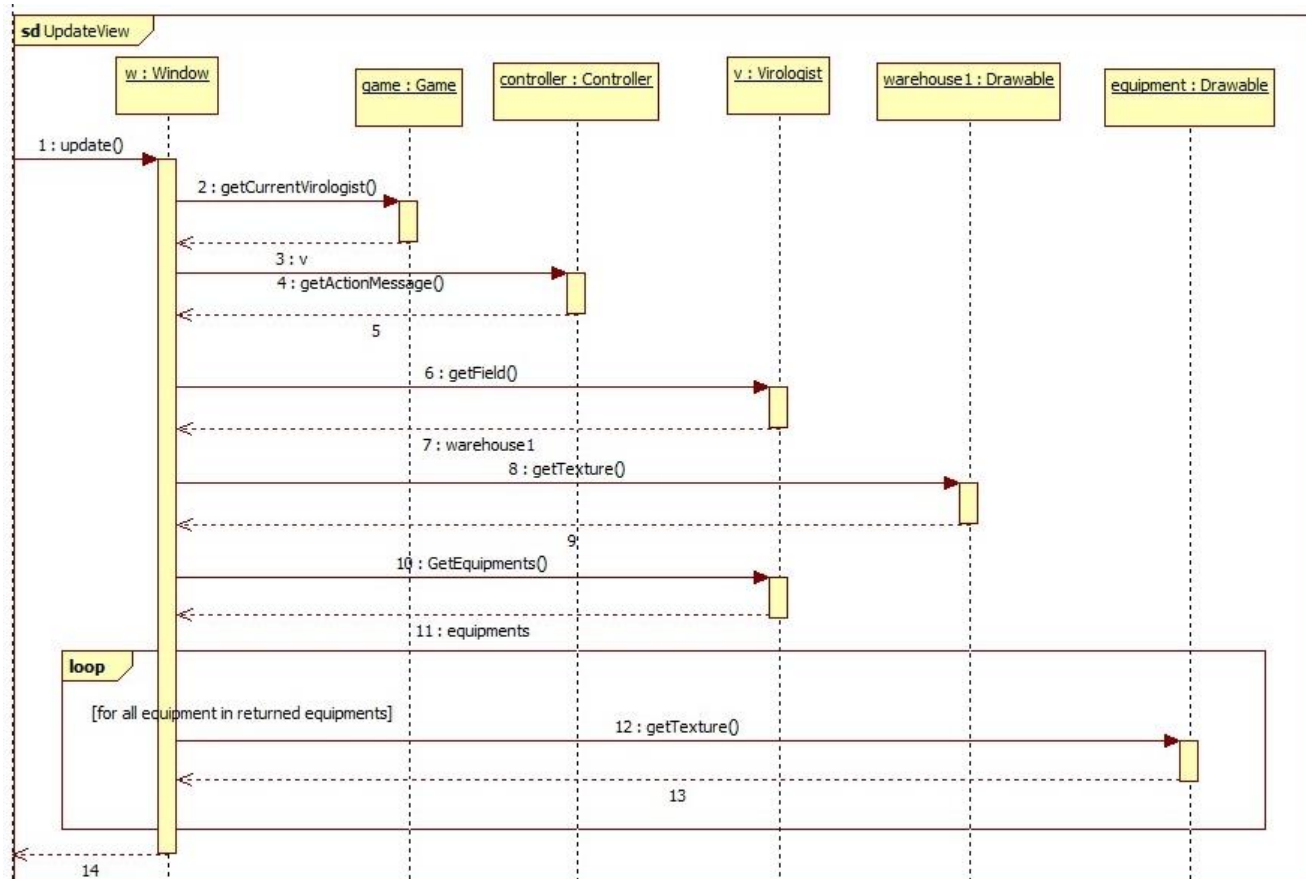
```
get current V Virologist from game
update aminoBar with data from V
update aminoLabel with data from V
update nucleoBar with data from V
update nucleoLabel with data from V
update turnCounter with data from V
get actionMessage from Controller
update ActionBubbleMessage with actionMessage
repaint self
```

## 11.4Kapcsolat az alkalmazói rendszerrel

### 11.4.1 Initialize

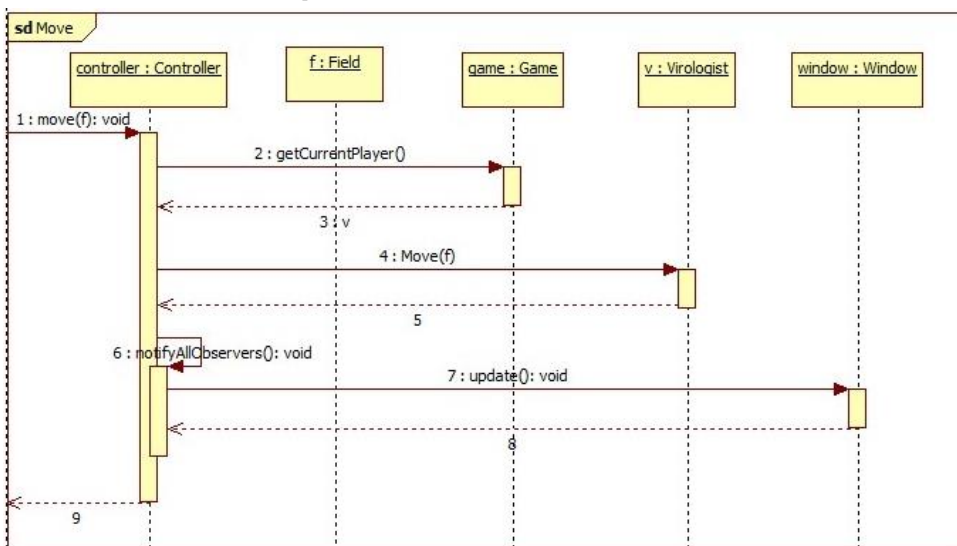


### 11.4.2 UpdateView



m.j.:A „controller”, azon belül pedig a loader felelőssége, hogy a modellben lévő Field-ek, és equipmentek, nem a modellbeli objektumok legyenek, hanem a Drawable változataik. Mellesleg a Window kezelni tudja majd azt is, ha ezt a felelősséget nem teljesíti a Loader, csak ilyenkor nem lesz annyira magasszintű ergonomiailag a játékelmény. (Másszóval default textúrák lesznek. pl.: bármilyen Field-nek a standard Field textúra lesz a textúrája, valamint bármely equipmentnek egy egyszínű szürke háttér.)

### 11.4.3 UserInput





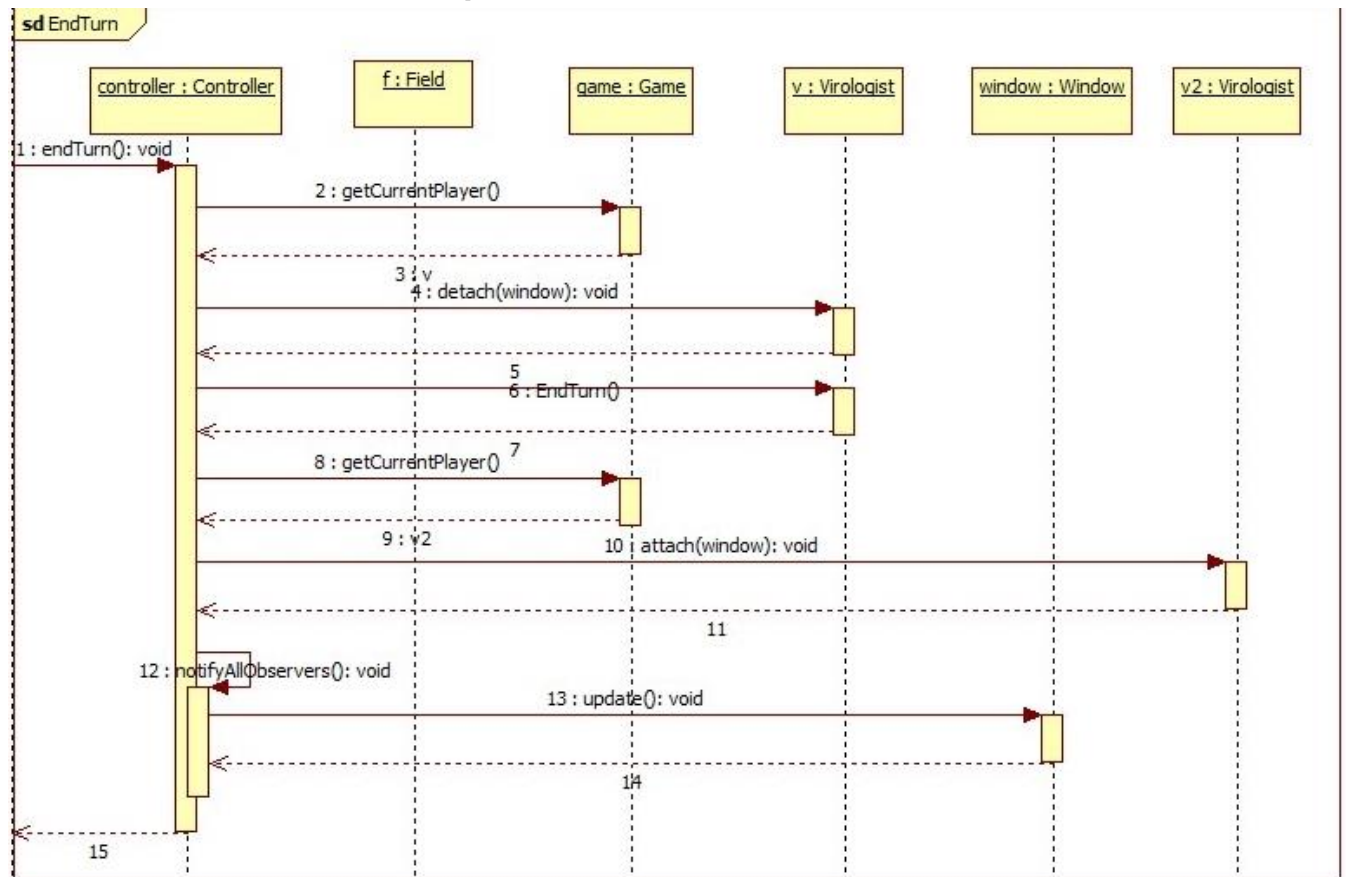
A szekvenciadiagram pontosan így néz ki az attack, drop, equip, learn, collect, loot, inject alternatívákra is, azzal a különbséggel, hogy a hívásdelegáció (4. metódushívás a fenti szekvencián) nem Move(f), hanem a megfelelően felparaméterezett:

Attack(v: Virologist), Drop(), Equip(), Learn(), Collect(), lootAminoAcidFrom(v: Virologist), LootNucleotideFrom(v: Virologist), Inject(v: Virologist, code: GeneticCode)

Ezeket tintatakarékossági okokból nem tüntettük fel.

m.j.: Az EndTurn hívás teljesen más lásd lentebb.

#### 11.4.4 EndTurn user input



**11.5Napló**

| <b>Kezdet</b>    | <b>Időtartam</b> | <b>Résztevők</b>                                | <b>Leírás</b>                                                                                                                         |
|------------------|------------------|-------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| 2022.04.30 14:30 | 2 óra            | Dancsó<br>Tóth<br>Bajnok<br>Róna<br>Pelczhoffer | Értekezlet,<br>felhasználói felület<br>közös megtervezése<br>vizuálisan, grafikus<br>specifikáció<br>elkészítése. Feladat<br>elosztás |
| 2022.05.01 12:00 | 2,5 óra          | Dancsó                                          | Struktúra diagram.                                                                                                                    |
| 2022.05.01 14:30 | 2,5 óra          | Tóth                                            | Szekvencia<br>diagrammok.                                                                                                             |
| 2022.05.01 18:00 | 2 óra            | Bajnok<br>Róna<br>Pelczhoffer                   | Dokumentálás                                                                                                                          |