

# 11. Grafikus felület specifikációja

62 – Matukain

Konzulens:

Salvi Péter

## Csapattagok

Monostori Dóra

PNNV8Z

dorimonostori@gmail.com

Marianna

Jónás Gergely Péter

UEYXGL

jonas.gergely24@gmail.com

Tóth Tódor

OWEKB9

todortoth@gmail.com

Blasek Balázs

CW4D3S

blasek.balazs@edu.bme.hu

Csordás Bence

PU8FXP

bencecsordas@edu.bme.hu

2025.05.02.

## 11. Grafikus felület specifikációja

### 11.1 A grafikus interfész

A grafikus felület 1280x720 pixel felbontású lesz. A játék megjelenítéséhez kizárólag a Java alapértelmezett Swing és AWT osztályai (pl. JPanel, JFrame, Graphics) kerülnek felhasználásra, külső grafikus könyvtárakat nem alkalmazunk. A kezelőfelület fix méretű, az összes UI-elem (pl. háttér) programozottan, Java beépített rajzoló műveleteivel kerül megjelenítésre, kivéve a játékállapot által dinamikusán változó elemeket, mint a fonalak, spórák, rovarok és tektonok állapota.

### 11.1.1 *Játék kezdete*

A grafikus felület indulása után megjelenik egy nyitóképernyő, melyen egy gomb található a „Játék kezdete”, ha a felhasználó rákattint, akkor megjelenik egy új képernyő, melyen a játékosok felosztása látható pl. Játékos 1 - Gombász, Játékos 3 - Rovarász. Ezek után ha a felhasználó bárhova kattint a képernyőn, akkor elkezdődik a játék.

Nyitóképernyő:



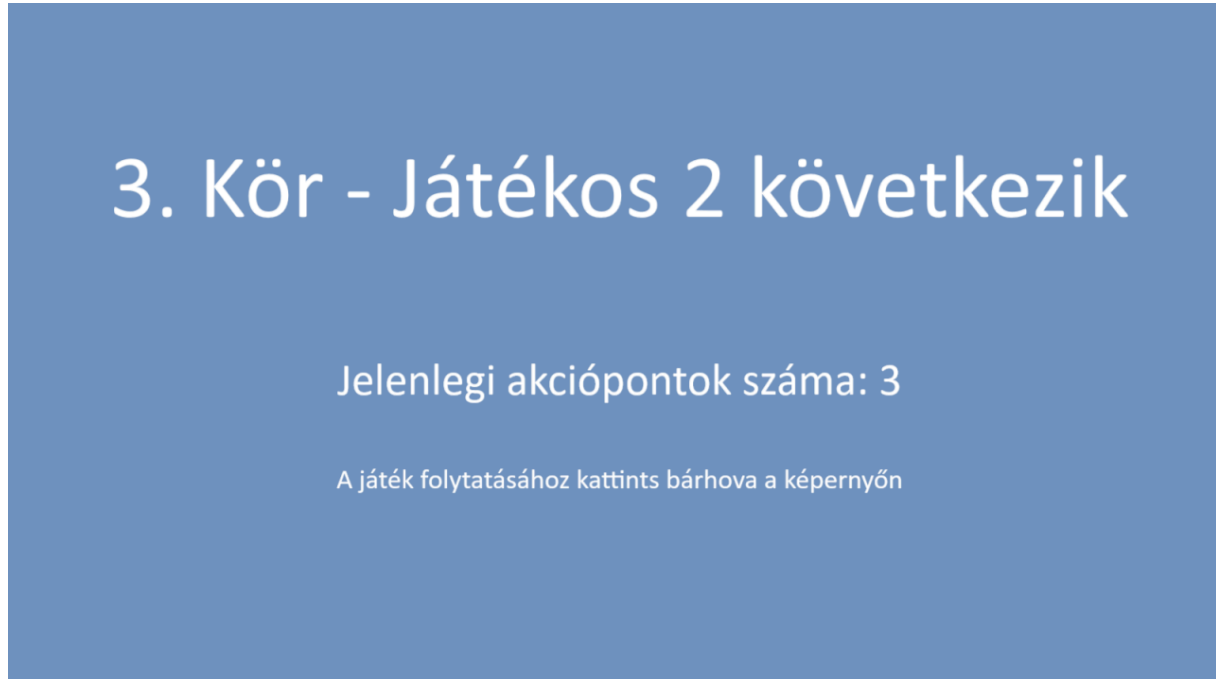
Szerepek felosztása:



### 11.1.2 Képernyő a játék közben

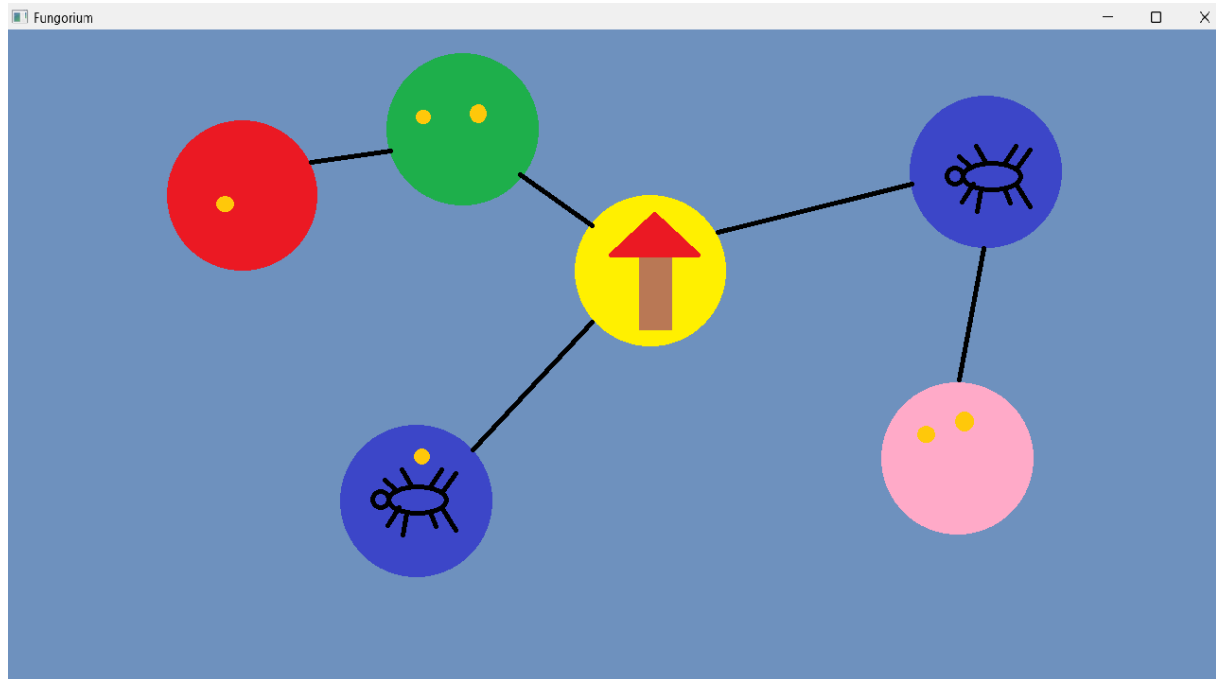
Mindig mikor a következő játékos következik, akkor megjelenik a képernyőn, hogy hányadik kör van, hogy ki következik és, hogy jelenleg hány akciópontja van a játékosnak. Itt a felhasználó, ha bárhova kattint a képernyőn, akkor a játék folytatódik az adott játékoskal.

Ez az alábbi képhez hasonlóan fog kinézni:



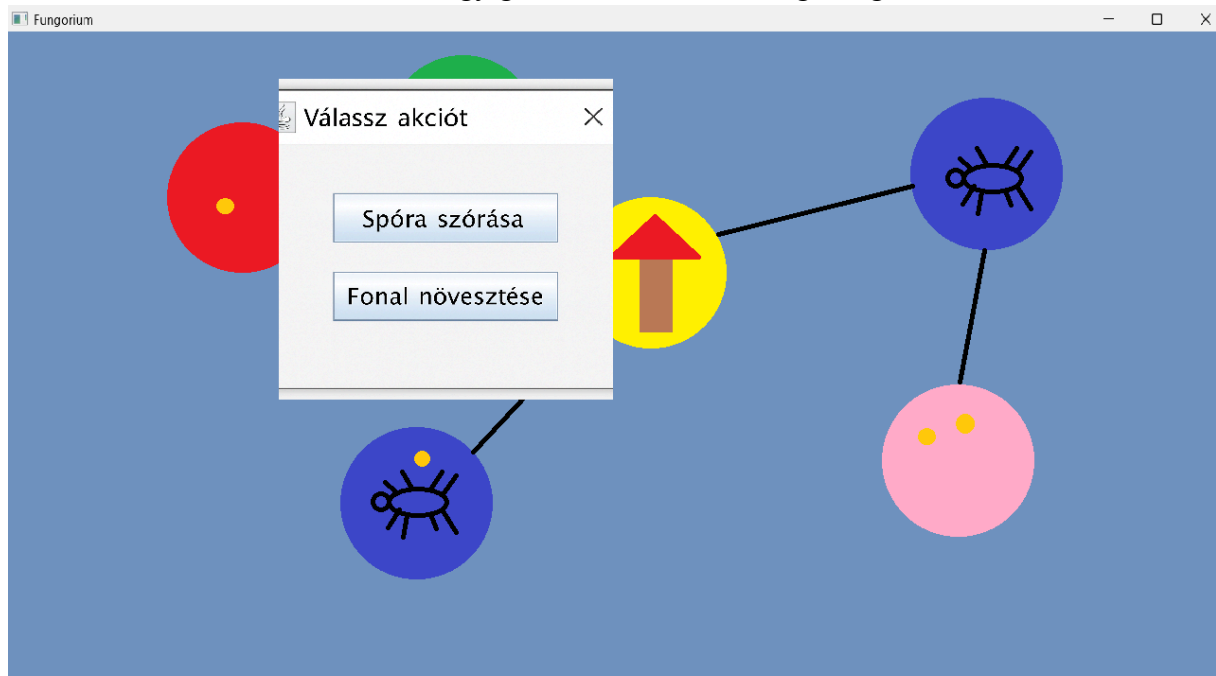
A pályán a különböző fajta tektonok különböző színekkel jelennek meg, a rózsaszín tekton egy MaxEgyFonal tektont reprezentál, a kék a TobbFonalTekton, a sárga a FonalMegtarto, a zöld a GombatestNelkuli, a piros pedig a Felszivodos. Minden spórát egy-egy narancssárga kör reprezentál. A játékosok mindig az adott objektumra kattintva tudják kiválasztani, hogy mit szeretnének éppen csinálni.

A térkép a játék egyik időpontjában:



Ha a játékosok egy olyan objektumra kattintanak, amivel van lehetőségük valamilyen akcióra (pl. gombász gombára tud kattintani, de rovarra nem), akkor megjelenik, hogy milyen akciók közül választhatnak.

Például, ha a felhasználó rákattint egy gombára ez az ablak fog felugrani:



## 11.2 A grafikus rendszer architektúrája

### 11.2.1 A felület működési elve

A grafikai megjelenítés során az MVC elvet követve a következő képpen oszlanak meg a feladatok:

- **Model**

A játék belső logikáját már implementáltuk az előző fázisban, így azt a már meglévő osztályok biztosítják. Különösképpen a gameLogic osztály, mely a játék kulcsfontosságú logikai elemeit (tektonok térképe, játékosok, hatások kifejtése) tárolja és implementálja. Ez az osztály tartalmazza az aktuális állapotot, amit a View meg tud jeleníteni.

- **View**

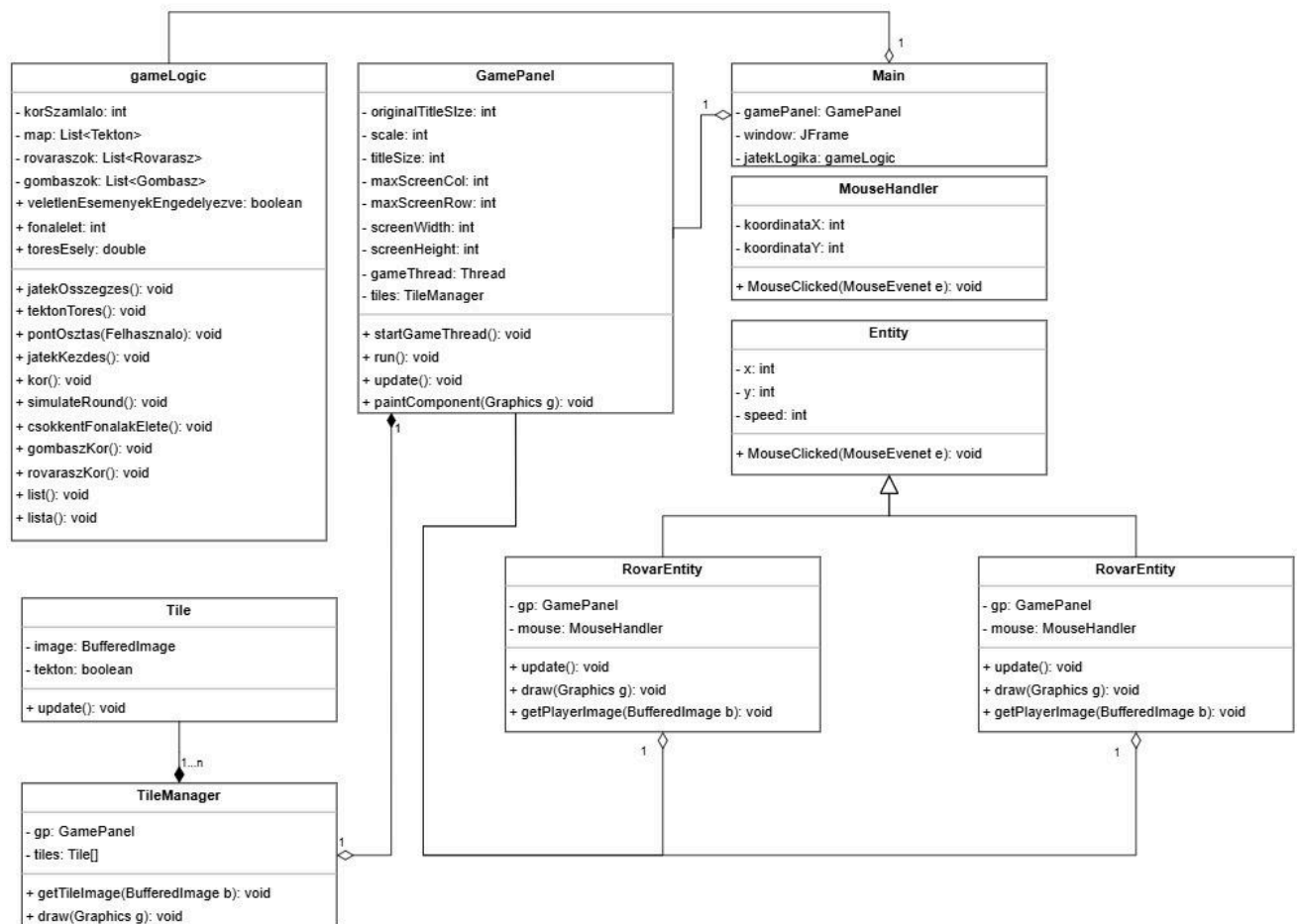
A grafikus megjelenítést végző osztályok mint pl. a GamePanel és a Main, felelősek azért, hogy eldöntsék a modell alapján mit kell kirajzolni. A GamePanel tudja, hány pixel található az ablakon, ehhez szükséges a Tile és TileManager osztályok. A Main ezeknek a megjelenítéséért felelős.

- **Controller**

A vezérlők a játék játékos oldali irányításának szempontjából kulcsfontosságúak, pl. A MouseHandler segítségével lehet eldönteni, hogy az adott felhasználó éppen melyik gombját/rovarát szeretné kiválasztani, hogy az adott körben lépjen vele. Ezek az osztályok ugyan hatással vannak (lehetnek) a modell állapotára, a modellből megjelenített képet hatásukkal közvetlenül nem változtatják meg.

A grafikus osztályok közül a GamePanel feladata, a modellhez való kapcsolódás, ezt a Main-ben található játékLogika objektumon keresztül teszi meg. Az osztály a modell pull alapú lekérdezését is megvalósítja, melyre egy Timer objektumot használ, 30 ms-os időközökkel a zavartalan élmény érdekében. Az új állapot alapján elvégzi az egyes Tile-ok újrarajzolását, mely alapján változhat a kép.

### 11.2.2 A felület osztály-struktúrája



## 11.3 A grafikus objektumok felsorolása

### 11.3.1 Main

- **Felelősség**

Ez az osztály teszi futtathatóvá a programunkat. A grafikus megjelenítésre a JFrame illetve a GamePanel osztályokat fogjuk használni.

- **Attribútumok**

- **GamePanel gamePanel:** Ezzel tesszük lehetővé, hogy majd ez az osztály belenyúljon a játék ablakába
- **JFrame window:** Ezzel hozzuk létre a játék ablakát, amit majd a GamePanelben fogunk frissíteni, újra rajzolni
- **gameLogic jatekLogika:** Ezzel tesszük lehetővé, hogy a játék hozzáférjen a már elkészített modellhez

### 11.3.2 GamePanel

- **Felelősség**

A pályát kezelő/szefoglaló, a képernyő frissítéséért és újra rajzolásért felelős osztály.

- **Össztályok**

JPanel->GamePanel

- **Interfészek**

Implements Runnable

- **Attribútumok**

- **int originalTileSize:** Eredeti csempék mérete
- **int scale:** skálázódás, hogy a méretarányok jók legyenek
- **int tileSize:** scale\*originalTileSize, azaz a teljes méret
- **int maxScreenCol:** Maximális oszlopszám
- **int maxScreenRow:** Maximális sorszám
- **int screenWidth:** Képernyő szélesség
- **int screenHeight:** Képernyő magasság
- **Thread gameThread:** A játékbeli idő kezelésére fog kelleni
- **TileManager tiles:** A játékbeli ablak megjelenítésére szolgál

- **Metódusok**

- **public void startGameThread():** Elindítja az idő kezelésére létrehozott threadet, és így elindítja a run()-t.
- **override public void run():** A játékbeli időt kezeli. Akkor lesz vége, ha a játéknak is vége lesz
- **public void update():** Frissíti a programot a történések után
- **public void paintComponent(Graphics g):** Ez a metódus rajzolja ki a gombokat, rovarokat, tektonokat stb.

### 11.3.3 MouseHandler

- **Felelősség**

Az egérekattintás kezeléséért felelős osztály

- **Össztályok**

MouseListener->MouseHandler

- **Attribútumok**

- **int koordinataX:** Az egér kattintás X koordinátája.
- **int koordinataY:** Az egér kattintás Y koordinátája.

- **Metódusok**

- **override public MouseClicked(MouseEvent e):** A kattintott helyhez hozzárendeli a koordinátákat a képernyőn.

### 11.3.4 Entity

- **Felelősség**

Az entitásokért felelős osztály, mint pl. a rovarok.



- **Attribútumok**
  - **int x**: Az entitás pozíciójának x koordinátája.
  - **int y**: Az entitás pozíciójának y koordinátája.
  - **int speed**: Az entitás mozgási sebessége.
- **Metódusok**
  - **override public MouseClicked(MouseEvent e)**: A kattintott helyhez hozzárendeli az entitásokat

### 11.3.5 RovarEntity

- **Felelősség**  
Az entitásokért felelős osztály, mint pl. a rovarok.

- **Össztályok**  
Entity->RovarEntity

- **Attribútumok**
  - **GamePanel gp**: A pálya amelyen a rovar van.
  - **MouseHandler mouse**: Az egérkezelő, amely visszatér a kattintással.
- **Metódusok**
  - **public void update()**: Frissíti a Rovar helyzetét a pályán.
  - **public void draw(Graphics g)**: Megrajzolja a Rovar karakterét.
  - **public void getPlayerImage(BufferedImage b)**: Betölti a Rovar képét.

### 11.3.6 GombaEntity

- **Felelősség**  
A Gomba entitásokat kezelő osztály

- **Össztályok**  
Entity->GombaEntity

- **Attribútumok**
  - **GamePanel gp**: A pálya amelyen a gombavan.
  - **MouseHandler mouse**: Az egérkezelő, amely visszatér a kattintással.
- **Metódusok**
  - **public void update()**: Frissíti a Gomba helyzetét a pályán.
  - **public void draw(Graphics g)**: Megrajzolja a Gomba karakterét.
  - **public void getPlayerImage(BufferedImage b)**: Betölti a Gomba képét.

### 11.3.7 Tile

- **Felelősség**  
Az pixelekért felelős osztály.

- **Attribútumok**
  - **BufferedImage image:** A pixel textúrája.
  - **boolean tekton:** Megmondja, hogy az adott pixel tekton-e vagy sem. Ez igenis fontos tényező, hiszen ez segít a szomszédok meghatározásában, fonalak húzásában, kattintásokban, rovar elhelyezésében, stb
- **Metódusok**
  - **public void update():** Frissíti a csempéket a pályán

### 11.3.8 TileManager

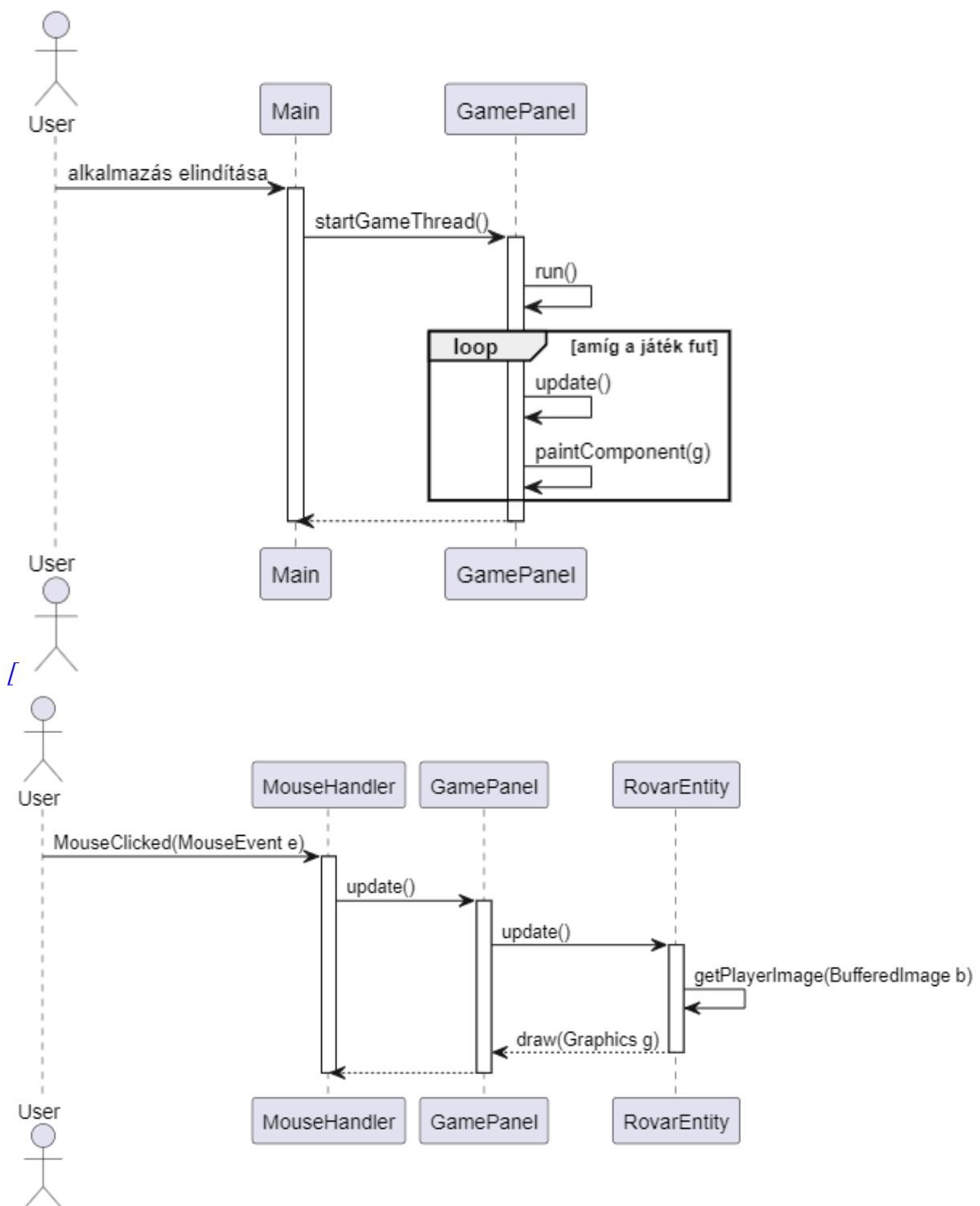
- **Felelősség**

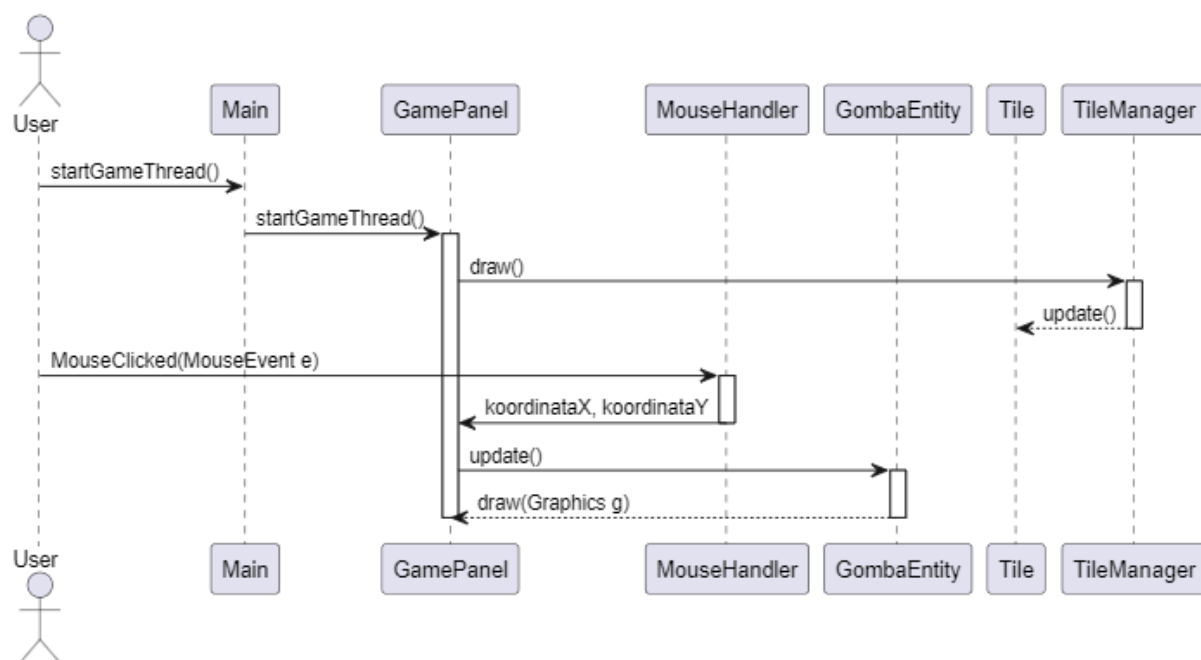
A Tile-okat összefogó és kezelő osztály.
- **Attribútumok**
  - **GamePanel gp:** Ide rajzolja majd ki a csempéket a draw() segítségével
  - **Tile[] tiles:** A különböző csempéket tároló tömb.
- **Metódusok**
  - **public void getTileImage(BufferedImage b):** visszaadja, hogy milyen típusú "csempe" van az adott helyen
  - **public void draw(Graphics g):** Kirajzolja a kezdő térképet a kijelzőre

### 11.3.9 Maps

- **Felelősség**

A bemeneti mapot tárolja számok segítségével txt formában

**11.4 Kapcsolat az alkalmazói rendszerrel**



## 11.5 Napló

| Kezdet            | Időtartam | Résztevők                                           | Leírás                                                                                                                                                              |
|-------------------|-----------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2025.05.02 22:00  | 1 óra     | Monostori,<br>Blasek,<br>Csordás,<br>Jónás,<br>Tóth | Értekezlet<br>Döntések:<br>Tóth és Monostori elkészíti a 11.1 és 11.2-es részeket<br>Blasek elkészíti a 11.4-es részt<br>Jónás és Csordás elkészíti a 11.3-as részt |
| 2025.05.03 10:00  | 4 óra     | Csordás                                             | 11.3-as rész elkezdése                                                                                                                                              |
| 2025.05.03. 19:00 | 1 óra     | Tóth<br>Monostori                                   | 11.1 és 11.2 feladatok átbeszélése és felosztása                                                                                                                    |
| 2025.05.03 20:00  | 3 óra     | Monostori                                           | 11.1, 11.1.1, 11.1.2 pontok kidolgozása, screenshot szerű képek elkészítése                                                                                         |
| 2025.05.03 20:00  | 2 óra     | Tóth                                                | 11.2.1-es rész elkészítése                                                                                                                                          |
| 2025.05.04 11:00  | 5 óra     | Jónás                                               | 11.3 és 11.6 elkészítése.                                                                                                                                           |
| 2025.05.04 22:00  | 1 óra     | Tóth                                                | 11.2.2-es osztálydiagram elkészítése                                                                                                                                |
| 2025.05.04 23:00  | 1 óra     | Blasek                                              | Dokumentum szerkesztés                                                                                                                                              |
| 2025.05.05 0:00   | 3 óra     | Blasek                                              | 11.4-es szekvenciadiagramok készítése                                                                                                                               |

## 11.6 Felhasznált forrás

A grafikus felület elkészítésének megvalósításához kicsit utánajártunk. Nem akartunk külső könyvtárat használni hozzá, hiszen nem áll elegendő idő rendelkezésünkre azt elsajátítani, így natív megoldásra volt szükségünk. Az alábbi YouTube lejátszási listát találtuk, mely egyelőre szerintünk lefed mindent, ezért forrásként használtuk jelen dokumentum 11.3-as alpont elkészítéséhez.

[https://youtube.com/playlist?list=PL\\_QPQmz5C6WUF-pOODsbsKbaBZqXj4qSq&si=UjuUVIFynTuII0FP](https://youtube.com/playlist?list=PL_QPQmz5C6WUF-pOODsbsKbaBZqXj4qSq&si=UjuUVIFynTuII0FP)