

# Specifikáció:

## változtatott

### Sejtautomaták: Wireworld és Életjáték

A **sejtautomaták** olyan dolgok, amelyek egy rácson helyezkednek el. Minden cellának van egy állapota (pl. élő vagy halott). A szimuláció időlépésekben (generációkban) halad. Minden egyes lépésben a cellák állapota egyszerre frissül, egy meghatározott szabályrendszer alapján. Ez a szabályrendszer az adott cella és annak közvetlen szomszédjainak aktuális állapotát írja le.

Conway Életjátéka (Game of Life):

Egy kétdimenziós rácson működik, ahol minden cella két állapotban lehet: élő vagy halott. A szomszédságot általában a 8 szomszédos cella (Moore-szomszédság) jelenti. A szabályai a következők:

1. **Születés:** Egy halott cella élővé válik, ha pontosan 3 élő szomszédja van.
2. **Túlélés:** Egy élő cella élő marad, ha 2 vagy 3 élő szomszédja van.
3. **Halál:** Egy élő cella halottá válik, ha túl kevés (0 vagy 1) vagy túl sok (4 vagy több) élő szomszédja van (izoláció vagy túlnépesedés).

Wireworld:

A Wireworld egy speciális sejtautomata, amelyet digitális áramkörök szimulálására terveztek. Négy állapota van:

1. Üres
2. Vezető
3. Elektronfej
4. Elektronfarok

A szabályai (Moore-szomszédság alapján):

1. Az **Üres** cella mindig Üres marad.
2. Az **Elektronfej** a következő lépésben **Elektronfarok** lesz.
3. Az **Elektronfarok** a következő lépésben **Vezető** lesz.
4. A **Vezető** akkor válik **Elektronfej**-jé, ha pontosan 1 vagy 2 szomszédos cella Elektronfej állapotú. Minden más esetben Vezető marad.

Továbbiak:

A cél, hogy a játék testre szabható legyen. A felhasználónak lehetősége van a standard szabályok (pl. \$S23/B3\$) megváltoztatására, beleértve a túlélési (\$S\$) és születési (\$B\$) szabályokat

#### User case-ek:

- . Új szimuláció létrehozása: A Felhasználót alapvetően a Game Of Life-ba dobja
- . A rendszer egy ablakot jelenít meg, ahol a felhasználó kiválaszthatja a szimuláció típusát: "Életjáték" vagy "Wireworld"
- . **Új szimuláció létrehozása** A **Fájl > Új szimuláció** menüpontban adhatod meg a pálya méretét (szélesség, magasság) és típusát:
- . Életjáték szabályainak testreszabása Amennyiben a Felhasználó az Életjáték típust választotta, egy "Szabályok testreszabása" gomb/opció is elérhető
- . Ennek kiválasztásakor a rendszer egy beállító felületet jelenít meg
- . A Felhasználó itt szövegesen megadhatja a **Túlélés (\$S\$) szabályait**
- . A Felhasználó megadhatja a **Születés (\$B\$) szabályait**

#### Pálya kinézete

- . A Felhasználó egy megnyitott (új vagy betöltött) szimulációs térben van, a szimuláció áll
- . A Felhasználó az egérkurzorral alul el tudja indítani a játékot
- . Meg tudja állítani a játékot
- . *Életjáték esetén:* A kattintás váltogatja a cella állapotát (Élő/Halott).
- . *Wireworld esetén:* A Felhasználó egy eszköztárból kiválasztja a kívánt állapotot (Üres, Vezető, Elektronfej, Elektronfarok), és ezzel nyomja rá a rácsra.

#### Szimuláció futtatása

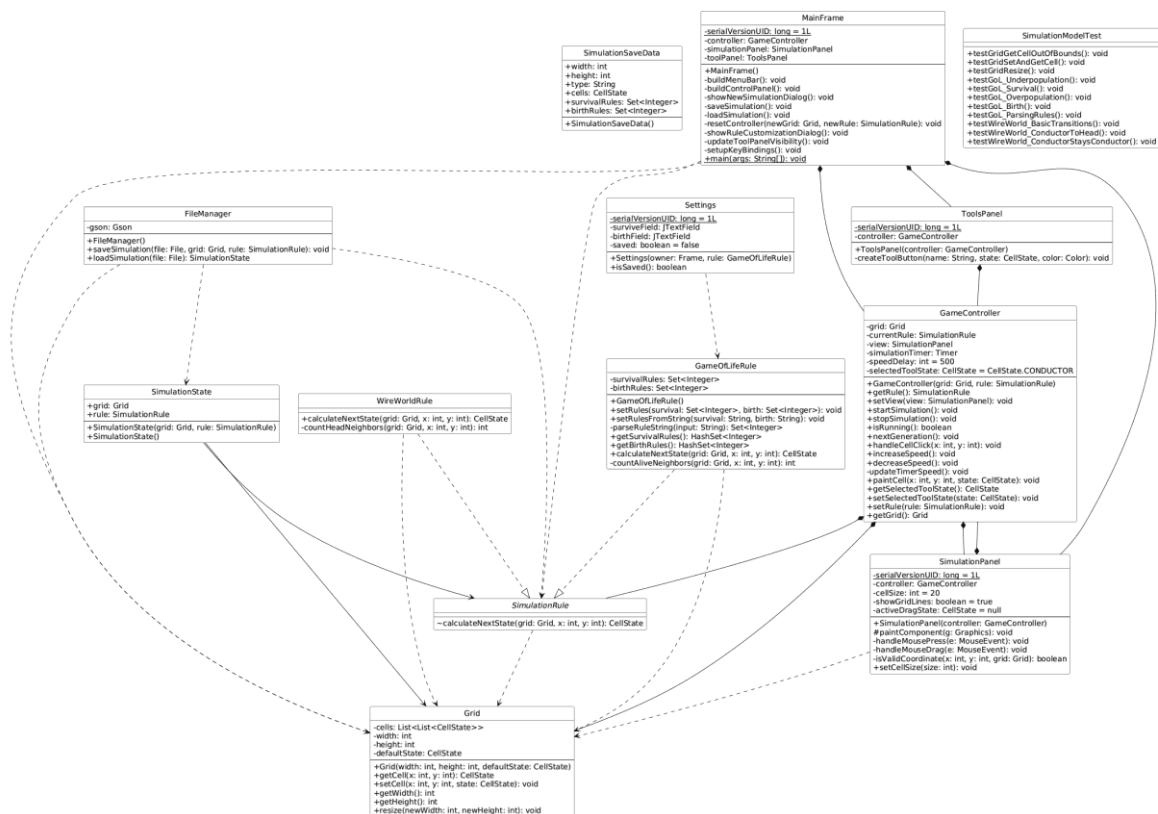
- . A Felhasználó megnyomja az "Indítás" (Start/Play) gombot.
- . A rendszer elkezdi léptetni a generációkat, és frissíti a rács nézetét.
- . A Felhasználó a fel és le nyíllal tudja állítani a sebességet
- . Szimuláció mentése A Felhasználó a menü "Mentés" opcióját választja
- . A rendszer elmenti a teljes szimulációs állapotot egy projekt fájlba, amely tartalmazza: ■ A szimuláció típusát (Wireworld/Életjáték).

- A rács méreteit és a cellák aktuális állapotát.
- Életjáték esetén a testreszabott szabályrendszert (S/B szabályok, szomszédság).
- Szimuláció betöltése A Felhasználó a fő menü "Betöltés" opcióját választja.
- A Felhasználó kiválaszt egy korábban mentett projektfájlt. A rendelkezésre álló fájlokat jeleníti meg a rendszer és azokból lehet választani
- A rendszer betölti a fájl tartalmát, beállítja a megfelelő szimulációs típust, a szabályokat (ha Életjáték), és megjeleníti a rácson a mentett állapotot.

### 3. Megoldási ötlet

A fájlok (pályák) JSON formátumból lesznek lementve, és betöltve

UML DIAGRAM:



A dokumentációhoz, JAVADOc-ot használtam minden fontosabb függvény és osztályhoz.