

# A PROGRAMOZÁS ALAPJAI 3.

VÉGLEGESÍTETT HÁZI FELADAT SPECIFIKÁCIÓ

WIREWORLD

KÉSZÍTETTE: MAJOR BÁLINT, TMJP5V  
[major@outlook.hu](mailto:major@outlook.hu)

KÉSZÍTÉS FÉLÉVE: 2022/23/1

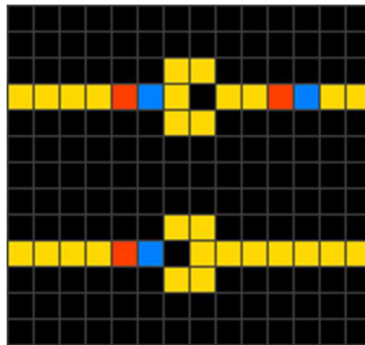
# TARTALOMJEGYZÉK

Informális házi feladat leírás.....	3
Formális házi feladat leírás .....	4
Az elkészítendő szoftver célja .....	4
Elvárások .....	4
Elvárt bemenet .....	4
Elvárt kimenet.....	4
Implementálandó funkciók.....	4
Use_case .....	4
Felhasználói interakció .....	5
Fejlesztéshez használt technológiák.....	5
Egyéb követelmények.....	6
Célközönség .....	6
Források: .....	6

## Informális házi feladat leírás

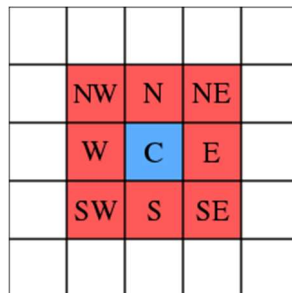
A Wireworld a sejt automaták csoportjába tartozó „játék”, hasonló a game of lifehoz, de ebben a felhasználónak lehetősége van egyszerűbb logikai hálózatok megépítésére. A programban egy cellának négy állapota lehet (ezeket általában 0-tól 3-ig számozzák és színek különböztetik meg őket), ezek a 0.üres (fekete), 1.elektron fej (kék), 2.elektron fark (piros), 3.vezeték/vezető(sárga). A cellák a játék indítását követően a szomszédos cellák állapotától függően változnak. A szabályok a következők:

- 0->0 Az üres üres marad
- 1->2 Az elektron halad a vezetékben és a fejből fark lesz
- 2->3 Az elektron tovább halad és a fark is eltűnik, helyette újra vezeték lesz
- 3->1 Ha pontosan 1, vagy 2 szomszédos elektronfej van a cella mellett, emiatt tud az elektron haldni a vezetéken



A képen két a játékban megvalósított dióda látható, a felső vezető az alsó pedig fordított irányban.

A Wireworld az úgynevezett Moore szomszédságon alapul, vagyis egy cella szomszédja minden olyan cella, ami pontosan egy cella távolságra található (bármely irányba) a kétdimenziós rácson, ez azt jelenti, hogy minden cellának nyolc szomszédja van.



A Moore modell alapján szomszédos cellák

A felhasználó a program indításakor egy menürendszert fog először látni, ebből választhat, hogy fájlból szeretne betölteni egy már elkezdett projektet, vagy újat szeretne kezdeni, esetleg kilépni. Ha újat kezd akkor egy 2 dimenziós rácsot fog látni, amelynek minden cellája üres. Ekkor elkezdheti az általa megvalósítani kívánt hálózat építését, ezen a szimulációt elindíthatja, megállíthatja. A megépített hálózatot fájlba mentheti.

# Formális házi feladat leírás

## Az elkészítendő szoftver célja

A programom célja egy az ügyfelek adatainak és az ő fogyasztásukkal kapcsolatos adatok felvételéhez, számlák kiírásához, befizetéséhez, szerződés kötéséhez, egyenleg lekérdezéséhez és fogyasztás bejelentéséhez alkalmas egyszerűsített objektum modell megvalósítása, valamint ehhez egy main függvény, amely alkalmas a tesztelésére.

## Elvárások

### Elvárt bemenet

A program futtatásakor egy egyszerű menü rendszerrel fogja tudni a felhasználó vezérelni, a menü pontokat és a program bezárását, ezt kattintással tudja majd a felhasználó kiválasztani.

A pályán a mezők állapotát is kattintással tudja majd módosítani, így kijelölni a pályát.

A szimulációt indítani és megállítani is egy-egy az ablakon megjelenő kattintható mezővel lehet majd irányítani, ugyanúgy ahogy a pálya mentését is.

### Elvárt kimenet

A program a pálya mentése esetén egy fájlba fogja kiírni a pálya adatokat, futás közben pedig a cellák állapotának változását fogja kijelezni a színek változtatásával, ezeken kívül más kimenetet megjeleníteni, vagy visszaadni.

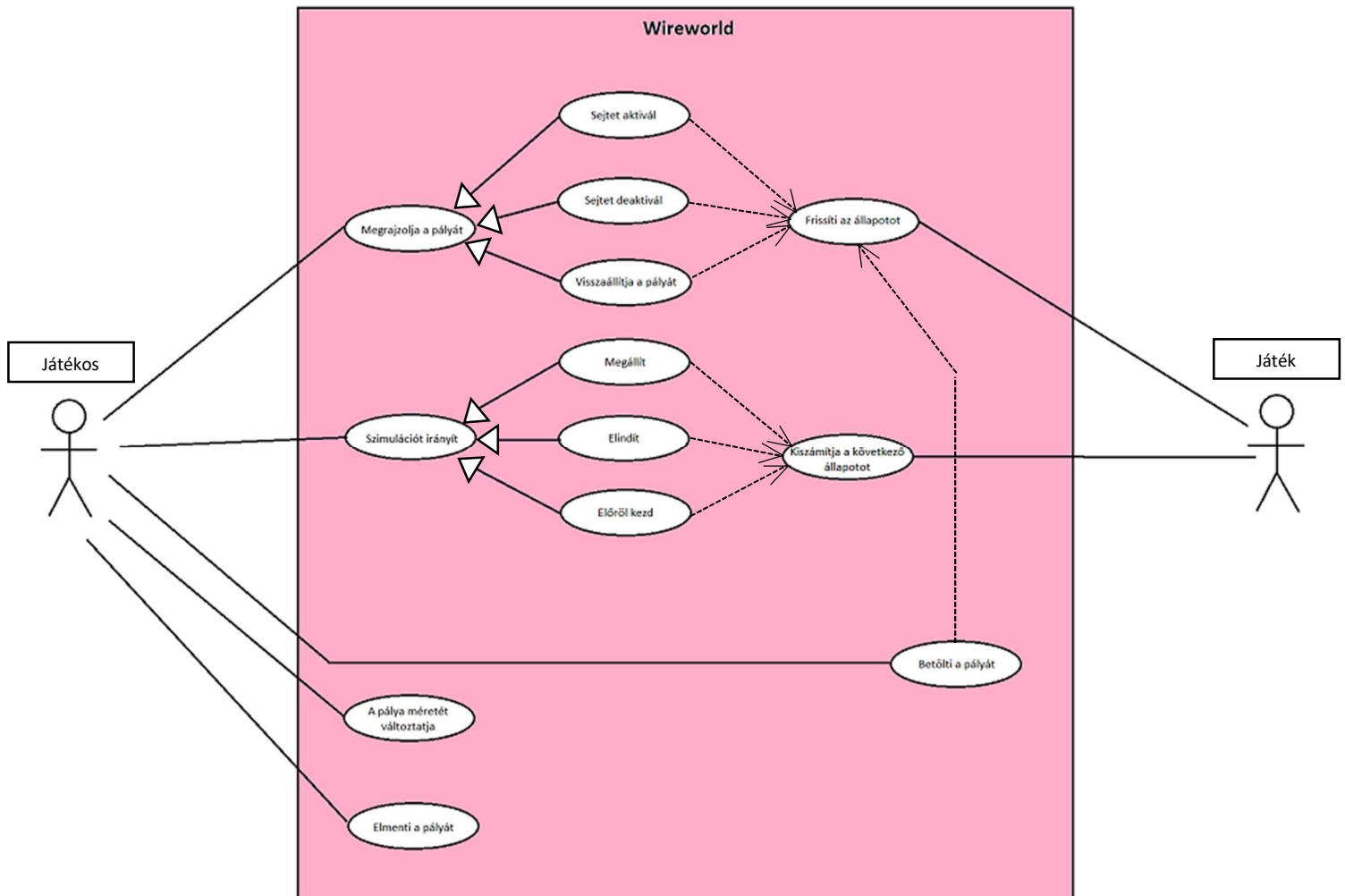
## Implementálandó funkciók

- Új szimuláció indítása
- Meglévő szimuláció betöltése
- Elkezdett szimuláció mentése
- A 2 dimenziós tervező rács méretének változtatása
- Szimuláció elindítása
- Szimuláció megállítása
- Szimuláció előről kezdése
- Egy segítő fájl megnyitása (ez pár logikai kapu tervét tartalmazza)
- Program bezárása

## Use\_case

- Játékos:
  - Új szimulációt indít – Létrehoz egy üres pályát az alapbeállításokkal
  - Méret változtatás – A pálya méretén lehet növelni és csökkenteni, hogy igazodjon a megvalósítani kívánt szimulációhoz
  - Szimuláció betöltése – Betölt fájlból egy előre elkészített pályát
  - Szimuláció indítása – Elindítja az elektront a pályán, vagy folytatja a szimulációt
  - Szimuláció megállítása – Megállítja az elektronokat a pályán éppen ott, ahol tartanak
  - Szimuláció mentése – Egy fájlba menti a pályát a program, ahonnan később be is tudja tölteni
  - Szimuláció előről kezdése – Az elektronokat letörli a pályáról, és megállítja a szimulációt

- „Segítő” fájl megnyitása – Megnyit egy dokumentumot, amiben egy pár logikai kapu terve szerepel
- Program bezárása – Leállítja a program futását
- Számítógép:
  - Újra tölti a pályát – Megvizsgálja a pálya jelenlegi állapotát és ez alapján kiszámítja, hogy milyen lesz a pálya a következő állapota



## Felhasználói interakció

A programot eclipsből (vagy más erre alkalmas IDE-ből) lehet majd futtatni. Ha már fut a program a felhasználó egy három elemből álló menüből kiválaszthatja, hogy mit szeretne. Ezek lehetnek az új szimuláció, a betöltés és a kilépés. Majd ez előbbi kettő választása esetén a tervezőrács jelenik meg, ahol a cellára való kattintással változtathatja a cella állapotát. Ezek mellett a képernyőn megjelenik a mentés, indítás, megállítás és az előről kezdés, valamint az a gomb, amivel pár logikai kapu tervét lehet megnézni egy dokumentumban.

## Fejlesztéshez használt technológiák

A programomat Java nyelven írom meg, integrált fejlesztőkörnyezetként eclipse IDE 2022-öt fogok használni.

## Egyéb követelmények

### Célközönség

A program leginkább oktatási és tanulási célokra lehet alkalmas, például a villamos mérnöki képzésen a digitális technika 1 tárgyhoz lehet hasznos, de vállalkozó szellemű felhasználók akár bonyolultabb logikai hálózatokat is építhetnek és kipróbálhatnak benne. Játéknak annyira nem nevezhető, de remélhetőleg szórakoztatóbb (és egyszerűbb) lesz a használatával a logikai egyenletek eredményének megtalálása

### Források:

- <https://en.wikipedia.org/wiki/Wireworld>
- [https://en.wikipedia.org/wiki/Moore\\_neighborhood](https://en.wikipedia.org/wiki/Moore_neighborhood)
- <https://nagyfotletek.wordpress.com/2019/10/15/seitautomatak/>