**Felhasználói dokumentáció**

A szimuláció indításához, el kell helyezned egy txt fájlt a Saves mappán belül. Ennek formátumát a specifikáció adja meg. Utána le kell buildelni a projektet az editorban és futtatni F9-cel. A program kiírja a kezdeti állapotot, amit a fájból beolvasott, utána kiírja a következő iterációját a sejteknek.

**Programozói dokumentáció**

A program adatszerkezete egy 2 dimenziós tömb (továbbiakban map), pointerek tömbjére mutató pointerként megvalósítva, ami a heapben lakik.

A fájlbeolvasás függvénye (részlet):

Paraméterként meg kell adni a fájl elérési útját és visszatér egy dinamikusan foglalt char\*\* típusú adatszerkezettel, amit a hívó felelőssége felszabadítani.

int row, col;

//file beolvasása

FILE\* save = fopen(filename, "r");

//játéktér lefoglalása

char\*\* map = (char\*\*) malloc(row \* sizeof(char\*));

^^^ Itt inicializáljuk a mapot és lefoglalunk neki annyiszor 8 byteot (64 bites oprrendszer esetén egy pointer mérete), ahány sorra lesz szükségünk.

//sorok lefoglalása és feltöltése

int itemCount = 0;

for(int i = 0; i<row; i++){

map[i] = (char\*) malloc(col \* sizeof(char));

^^^ Itt minden sorban annyi byteot foglalunk, ahány oszlopra lesz szükség.

for(int j = 0; j<col; j++){

char c = fgetc(save); <<< Ezzel pedig karakterenként olvassuk a fájlt.

map[i][j] = c;

}

//új sor karakter átugrása

fgetc(save);

//if(feof(save)) break;

}

A sejtek állapotának kirajzolása printMap(char\*\* map):

Paramétere char\*\* map és visszatérési értéke nincs. Mellékhatása, hogy kirajzolja a sejteket állapotuk szerint a konzolra iteratívan.

A sejtek következő állapotának beállítása nextState(char\*\* map):

Paramétere char\*\* map, visszatérése pedig szintúgy nincs, mivel címszerinti paraméterátadással dolgozik.

A countNeighbours(char\*\* map, int row, int col) bevesz egy mapot és annak dimenzióit. ezekből megállapítja a szabályok alapján, hogy hány élő szomszédja van egy sejtnek.

int countNeighbours(char\*\* map, int row, int col){

int count = 0;

int rowLength = 9;

int colLength = 9;

for(int i = row-1; i<=row+1; i++){

for(int j = col-1; j<=col+1; j++){

if(i < 0 || i >= rowLength) continue;

if(j < 0 || j >= colLength) continue;

if(i == row && j == col) continue;

if(map[i][j] == '#') count++;

}

}

return count;

}

A main függvényben egy rövid teszt található a program alapfunkcióinak szemléltetésére.