# 7. Sejtautomaták

Csillag Barnabás Gellért (COTNU3)

2020. május 2.

# 1. Az elméleti háttér

Jelen jegyzőkönyvben két modellel fogunk foglalkozni, amelyek értelmezhetőek sejtautomatákként. Az első ilyen modell a **Conway-féle életjáték**. Ezen modell esetében adott egy valamekkora négyzetrácsos tábla. A tábla minden eleme egy sejtet jelképez, amely lehet élő, vagy halott. A rendszer egy véletlenszerű állapotból indul, és egy sejt állapotát az adott körben az határozza meg, hogy az előző körben a vele szomszédos sejtek milyen állapotban voltak:

- ha $n$ darab élő szomszédja van, akkor marad az eddigi állapotában,
- ha $n+1$ élő szomszédja van, akkor mindenképpen élő lesz a következő körben,
- minden egyéb esetben elpusztul.

A szomszédságba az adott sejtet jelképező négyzettel a sarkán érintkező egyéb négyzetek is beletartoznak.

A második modell a **Bak-Tang-Wiesenfeld-féle homokdomb modell**. Ez egy két dimenziós modell, ami szintén egy négyzetrácsos táblaként képzelhető. A négyzetekben egy-egy diszkrét magasságú homok oszlop van. Ezen oszlopok magassága lehet 0,1,2,3,4,5,6, vagy 7. A rendszer léptetése úgy történik, hogy ha az adott oszlop magassága nagyobb háromnál, akkor levonódik belőle négy, ellenben a négy szomszédja magasságához hozzáadódik egy-egy - minden körben minden oszlop ledőlhet egyszer, ha ezen feltétel teljesül.

Ahhoz, hogy a modell jól definiált legyen, szükséges határfeltételek - például meg kell adni, hogy milyen szabályok legyenek érvényesek a tábla szélén. A tábla szélén vehetünk egyrészt nyitott határfeltételt, amely annyit tesz, hogy ott a homokszemek kiesnek a rendszerből. Ugyanígy használhatunk periodikus határfeltételt is, így ami homokszem a tábla egyik végén leesik, az a tábla másik végén ugyanott hozzáadódik a rendszer egy pontjához. Az is lehet egy további határfeltétel, hogy egy vagy több helyen folyamatosan további homokot szórunk a dombra.

Kezdeti feltételnek minden esetben vehetjük azt, hogy a tábla minden elemének értéke hét, így minden oszlop a maximumon van a nulladik időpillanatban.

## 2. Megvalósítás

A **Conway-féle életjáték** esetében egy valamilyen c++-os mátrix vagy tömb-típust terveztem használni, de nem találtam a célnak megfelelőt. Ebből kifolyólag std:vector-okkal dolgoztam - megfelelő jelölésrendszerrel lehet őket is két indexes tömbökként kezelni. Az adatvektorok elemei egyek vagy nullák lehetnek: az egy jelöli az élő sejtet, a nulla a halottat. A rendszer indításakor véletlenszerűen vannak kisorsolva a sejtek állapotai, vagyis az első táblán minden sejt ötven százalék eséllyel lesz élő, és ötven százalék eséllyel holt.

Az adatokat tároló objektum jellege azt is jelentette a kódom esetében, hogy a határfeltételek (a tábla széli szabályok) nem könnyen megadhatók voltak, hanem csak körülményesen lehetett őket implementálni. Három feltételt írtam meg (az élő határt, a periodikus határt, és a konstans véletlen határt), mert a feladat kiírásában szereplő "nyílt peremfeltételt"-t, mint kifejezést nem tudtam értelmezni. Az élő határ mindössze annyit tesz, hogy ugyanazon szabályok érvényesek a tábla szélén, mint bárhol máshol annak ellenére, hogy az ott található sejteknek nyilvánvaló okokból kevesebb szomszédja van. A periodikus határfeltétel azt jelenti, hogy például a egy bal szélső elem bal oldali szomszédjának ugyanazon sor jobb szélső eleme van tekintve, és így tovább. A konstans véletlen határ abban merül ki, hogy a tábla határán levő sejtek megmaradnak az eredeti, véletlenszerű állapotukban.

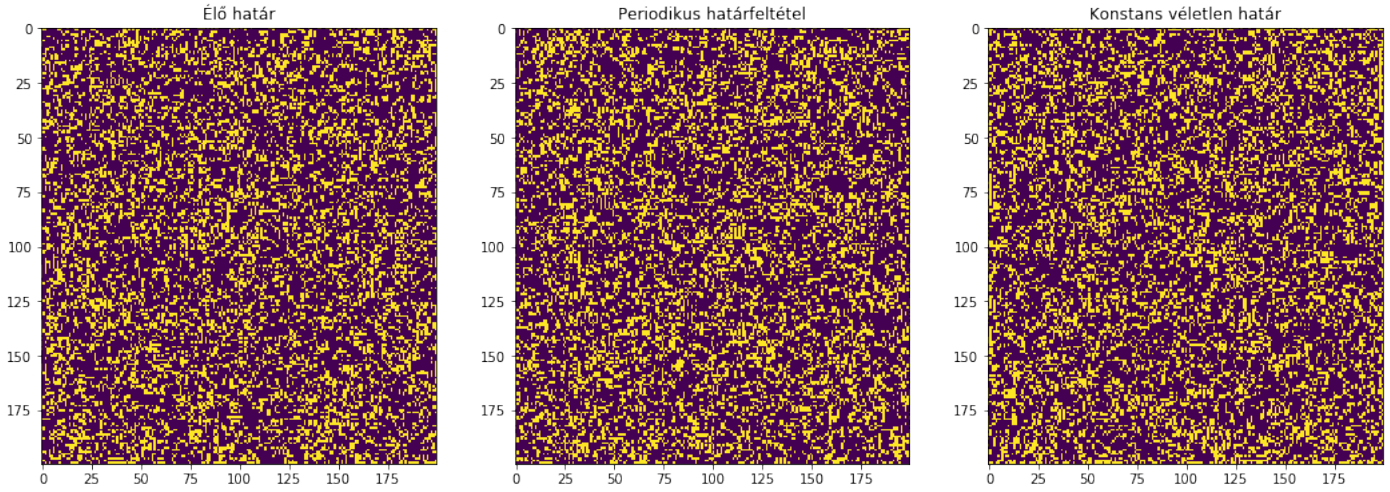
A **Bak-Tang-Wiesenfeld-féle homokdomb modell** esetében adta magát, hogy ismét az std::vector osztályt alkalmazzam jobb híján. Leprogramoztam egy tetszőleges méretű táblát, amelynek meg lehet adni periodikus és nyílt határfeltételt, továbbá nyílt határfeltételt plusz homokszórással. A periodikus határfeltételnek azonban nem látom értelmét úgy, ha a tábla minden elemét hétről indítjuk, mert ennek azt kell eredményeznie, hogy továbbra is hét lesz minden elem. Ezt könnyen be lehet látni: minden leborulás során négy szomszédos oszlop kap egy-egy egység homokot. Az elején minden oszlop hetes magasságú, így három lesz az értéke a leborulás után. Azonban mivel minden oszlop leborul, és a határfeltétel miatt minden oszlopnak van négy szomszédja, ezért minden oszlophoz a kör végéig hozzáadódik négy egységnyi homok, így újra minden oszlop magassága hét.

Az ábrákat és az animációkat Python nyelven készítettem a szimuláció futása során kimentett adatvektorokból.

# 3. Eredmények

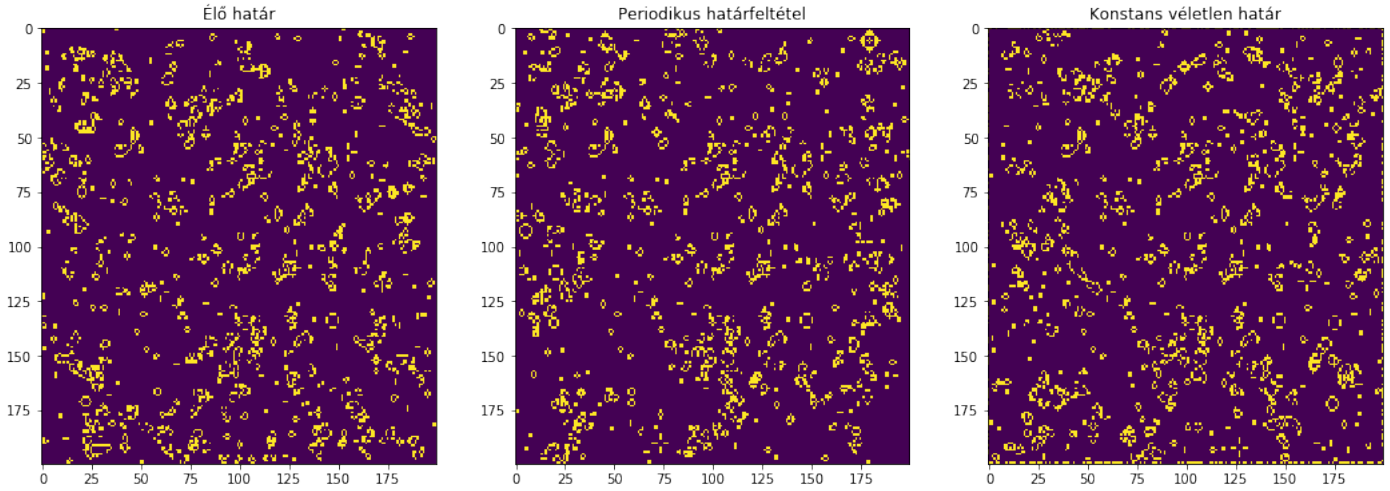
## 3.1. Conway-féle életjáték

Az animációkat a jegyzőkönyv mellé csatolom.




1. ábra. Conway sejtautomata különböző határfeltételekkel futtatott szimulációinak végeredménye $n = 1$ esetén, 100 léptetést követően.

Az $n = 1$ esetben a rendszer egy rendezetlennek tűnő oszcillációt mutat minden határfeltétel mellett, ahogy azt a "perhatarn1.gif", "konsthatarn1.gif", "elohatarn1.gif" animációkon is láthatjuk.




2. ábra. Conway sejtautomata különböző határfeltételekkel futtatott szimulációinak végeredménye $n = 2$ esetén, 100 léptetést követően.

Szűkebb értelemben véve az $n = 2$-es esetet szokták a Conway-féle életjátéknak nevezni, mert ezen konfigurációban jönnek létre a legnagyobb számban érdekes, stabil sejt-csoportosulások. Az ilyen stabil csoportosulási típusokra számos példa található az algoritmus Wikipédia oldalán [1]. A 2. ábrán, és a csatolt animációkon is számos ilyen felfedezhető.

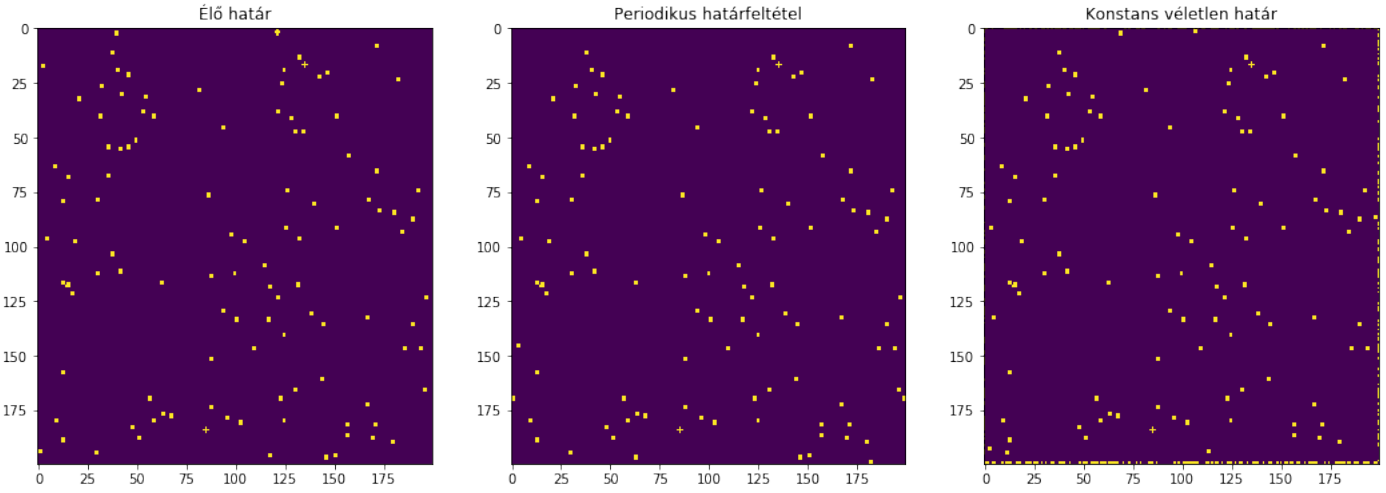
A különböző határfeltétellel futtatott szimulációk között csak eltérések láthatóak - nem sok táppont van, amely alapján össze lehetne őket hasonlítani. A jobb oldali ábrán egyértelműen látszik, hogy a határ lemaradt a képről, mert a kép bal és felső szélén található sejtek egyáltalán nem tűnnek véletlenül sorsoltnak. Ez valamilyen ábrázolási hiba, ami szerencsére az animációkon nem jelentkezett, viszont a többi ilyen ábrán is látszani fog.

Az ide tartozó animációk a következők:

- "perhatarn2.gif" - periodikus határfeltétellel a 100. lépésig futtatott szimuláció,
- "konsthatarn2.gif" - véletlen konstans határfeltétellel a 100. lépésig futtatott szimuláció,
- "elohatarn2.gif" - élő határral a 100. lépésig futtatott szimuláció,
- "perhatarn2h.gif" - periodikus határfeltétellel a 400. lépésig futtatott szimuláció,
- "konsthatarn2h.gif" - véletlen konstans határfeltétellel a 400. lépésig futtatott szimuláció,
- "elohatarn2h.gif" - élő határral a 400. lépésig futtatott szimuláció.

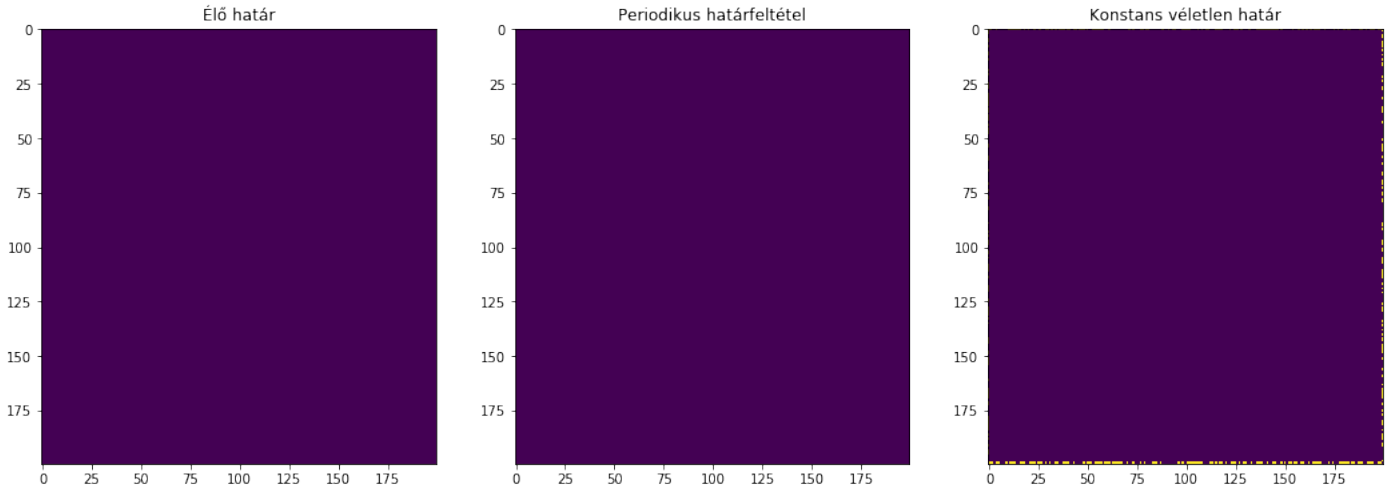
Itt kivételesen hosszabb szimulációkat is futtattam, mert a többi $n$ beállításhoz képest sokkal hosszabb folyamatok vannak jelen.

Az animációknál a továbbiakban nem írom ki, hogy mi hova tartozik, mert ugyanúgy neveztem el őket, mint ezen legutóbbiakat, csak az "n" mögötti szám változott a szimulációtól függően.




3. ábra. Conway sejtautomata különböző határfeltételekkel futtatott szimulációinak végeredménye $n = 3$ esetén, 100 léptetést követően.

Míg az előző ábrán és animációkban alakzatok széles tárháza volt felfedezhető, itt csupán négy elemből álló négyzeteket láthatunk. Ha jobban megvizsgáljuk az ábrákat, azt találhatjuk, hogy felettébb hasonlítanak, csak a két jobb oldalin vannak olyan csoportok, amelyek a bal oldalin nincsenek. Ezen hasonlóság oka feltételezésem szerint, hogy a generátor, amely a véletlen egyeseket és nullákat generálja a rendszer kezdőállapotához, nem működik megfelelően, és mindig ugyanazon számsor elejét használja fel. A különbségek okai pedig egyértelműen a határ különbözőségeiben keresendők. Azért az élő határos kezdőfeltételekkel ellátott szimuláció végeredményénél van a legkevesebb csoport, mert ott kevesebb az interakció a sejtek között a középsőnél, és nincsenek állandó élő sejtek a tábla szélén, mint a jobb oldalinál.

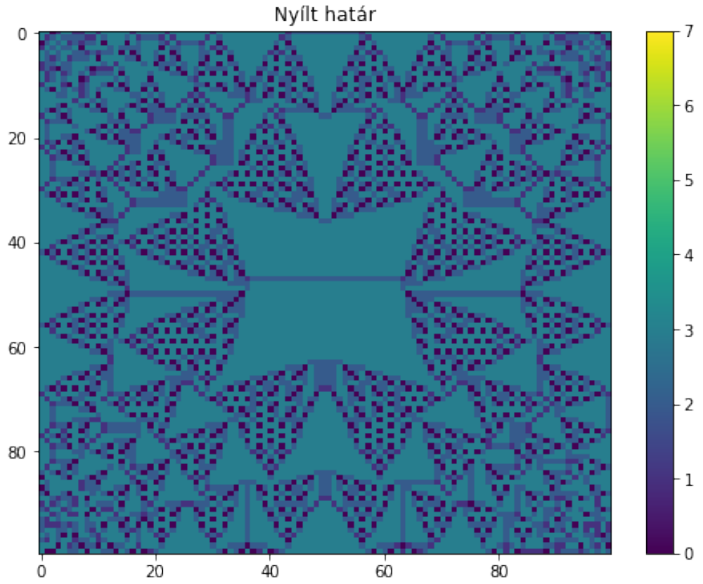



4. ábra. Conway sejtautomata különböző határfeltételekkel futtatott szimulációinak végeredménye $n = 4$ esetén, 100 léptetést követően.

Az $n = 4, 5, 6, 7, 8$ esetekben ugyanazon végeredményt kapjuk: nincsenek stabil csoportosulások, és végeredményben az összes sejt elhalálozik. Annyi különbség fedezhető fel köztük az animáció alapján, hogy $n$ minél nagyobb, annál gyorsabban elpusztul az összes sejt. Ez érthető, hiszen nyilvánvalóan ezek közül $n = 4$ szomszéddal rendelkező sejtből van a legtöbb, és $n = 8$ szomszéddal rendelkező sejtből van a legkevesebb az ilyen táblákon.

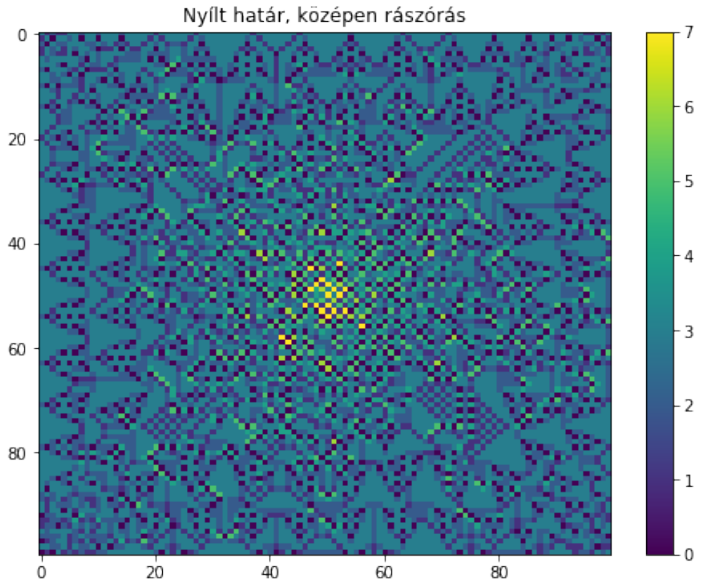
### 3.2. Bak-Tang-Wiesenfeld-féle homokdomb modell

A szimulációk eredményei alapján készült animációkat a jegyzőkönyv mellé csatoltam.




5. ábra. Bak-Tang-Wiesenfeld-féle homokdomb nyílt határfeltétellel futtatott szimulációinak végeredménye 10000 léptetést követően.

Mivel a végeredmény nagy hasonlóságot mutat a szakirodalomban fellelhető eredményekkel, vagyis a program feltételezhetően jól működik.




6. ábra. Bak-Tang-Wiesenfeld-féle homokdomb nyílt határfeltétellel, és további, az $[50, 50]$ indexű pontba körönként három egységnyi homok szórásával futtatott szimulációinak végeredménye 10000 léptetést követően.

Láthatóan magasabb lett a középső régió az odaszórt homoktól, ami elvárható volt.

## 4. Összefoglalás

A jegyzőkönyv során bemutatásra került kétféle, általam írt algoritmus, amelyek közül az egyik a Conway-féle életjátékot, a másik pedig egy homokdomb-modellt valósított meg. Utóbbi esetében megvizsgálásra kerültek a különböző szabályokkal és határfeltételekkel futtatott szimulációk eredményei. Arra a következtetésre jutottunk, hogy $n = 1$ esetén a rendszer kaotikus viselkedést mutat, $n = 2$ esetén visszakapjuk a szűkebb értelemben vett Conway-féle életjátékot, amelynél rengetegféle stabil alakzat figyelhető meg, $n = 3$ esetében csak kis négyszögek bizonyultak stabilnak, $n > 3$ esetében pedig nem voltak stabil formációk, így minden sejt gyorsan kipusztult. A különböző határfeltételek hatását is megvizsgáltuk.

A homokdomb modell esetében az elméleti bevezetőt követően bemutatásra került a program működése, és az általa szolgáltatott eredmények. Sajnos a skálázási törvényt az órai dián belinkelt jegyzetből nem értettem meg, így a kitevőjét sem tudtam lemérni.

## Hivatkozások

[1] `https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life`