

Celluláris hullámszámítógépek projektmunka

Csornai Borbála
ZZMDFE

2020.05.31.

Hexagonal Game of Life

Bevezetés

A Game of Life John Horton Conway, brit matematikus 1970-ben kiadott találmánya. A játék 0 játékost igényel, hiszen az evolúciót csak a kezdeti állapot határozza meg, ebből megfigyelhetjük, hogy az egyes kiindulási pontok milyen irányba fejlődnek. További inputok nem szükségesek. A rendszer Turing-teljes, vagyis bármit, amit ki lehet algoritmikusan számolni, az kiszámolható vele.[1]

A szabály

Élő cellák esetén:

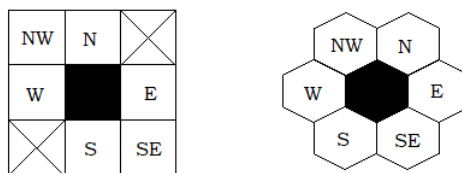
- Ha kettőnél kevesebb élő szomszédja van a túlzott elszigeteltség miatt meghal.
- Ha háromnál több élő szomszédja van túlnépesedés miatt meghal.

Halott cellák esetén:

- Mindenki, akinek pontosan három élő szomszédja van újraéled.

A feladat implementálása

Kiindulásként a hagyományos, négyzet-alapú játékot kezdtem tanulmányozni, majd összehasonlítottam a hatszög-alapú változattal. Megfigyeltem, hogy a két implementáció csupán a vizsgált cellák számában tér el. Így, az egyszerűbb megoldás érdekében párhuzamot vontam a négyzetes és a hexagonális egységek között. Ezt az 1. ábra szemlélteti.



1. ábra. A négyzetes és a hexagonális egységek összehasonlítása

Megfigyelhető, hogy a két struktúrát megfeleltethetjük egymásnak, úgy, hogy a négyzetes megoldás esetén a 8-ból 2 szomszédot nem veszünk figyelembe az élő szomszédok számolásakor.[2]

Esetemben a jobb felső $(i+1, j-1)$ és a bal alsó $(i-1, j+1)$ szomszéd vizsgálatakor élő cella esetén is 0-t adtam az összeghez.

Az implementáláshoz egy már létező, négyzetes játékot leíró Java kódot használtam alapul, a hexagonális szabályokra formázva.[3]

A programkód

```
function make2DArray(cols , rows) {
  let arr = new Array(cols);
  for (let i = 0; i < arr.length; i++) {
    arr[i] = new Array(rows);
  }
  return arr;
}

let grid;
let cols;
let rows;
let resolution = 10;

function setup() {
  createCanvas(600, 400);
  cols = width / resolution;
  rows = height / resolution;

  grid = make2DArray(cols , rows);
  for (let i = 0; i < cols; i++) {
    for (let j = 0; j < rows; j++) {
      grid[i][j] = floor(random(2));
    }
  }
}

function draw() {
  background(0);

  for (let i = 0; i < cols; i++) {
    for (let j = 0; j < rows; j++) {
      let x = i * resolution;
      let y = j * resolution;
      if (grid[i][j] == 1) {
        fill(255);
        stroke(0);
        rect(x, y, resolution - 1, resolution - 1);
      }
    }
  }

  let next = make2DArray(cols , rows);

  for (let i = 0; i < cols; i++) {
    for (let j = 0; j < rows; j++) {
      let state = grid[i][j];
      let sum = 0;
      let neighbors = countNeighbors(grid , i , j);
```

```
        if (state == 0 && neighbors == 3) {
            next[i][j] = 1;
        } else if (state == 1 && (neighbors < 2 || neighbors > 3)) {
            next[i][j] = 0;
        } else {
            next[i][j] = state;
        }
    }
}
grid = next;
}

function countNeighbors(grid, x, y) {
    let sum = 0;
    for (let i = -1; i < 2; i++) {
        for (let j = -1; j < 2; j++) {
            let col = (x + i + cols) % cols;
            let row = (y + j + rows) % rows;
            if ((i == -1 && j == 1) || (i == 1 && j == -1)) {
                sum += 0;
            } else {
                sum += grid[col][row];
            }
        }
    }
    sum -= grid[x][y];
    return sum;
}
```

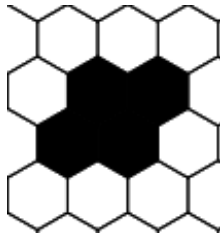
Érdekes struktúrák

A játék kellő idejű futtatása után jellemző struktúrák alakulnak ki. Esetünkben nagyrészt stabil szerkezeteket láthatunk, de megjelenhetnek oszcillátorok is. Ezekből a struktúrákból már nem tudjuk kimozdítani a játékot, stabil kimenet keletkezik.

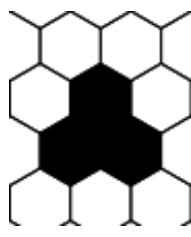
A megadott szabályok alapján a 6 szomszédos változatnál, a hagyományos Game of Life-fal ellentétben nem találtam mozgó struktúrákat (glidereket).

Stabil szerkezetek

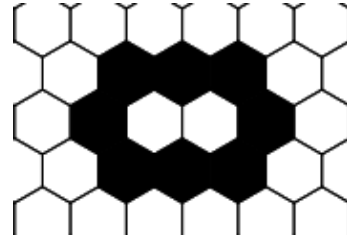
A 2., 3. és a 4. ábra stabil szerkezeteket ábrázol, melyek amint felveszik az alakzatot a játék végéig változatlanok maradnak.



2. ábra.

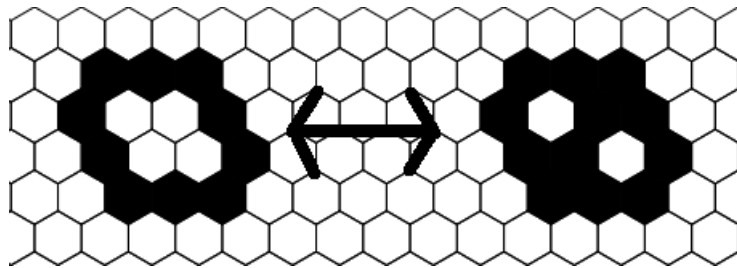


3. ábra.

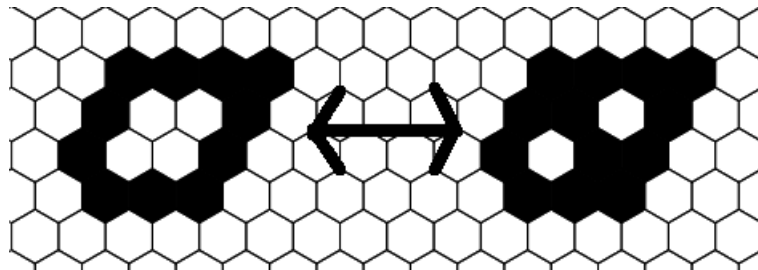


4. ábra.

Periodikus szerkezetek



5. ábra.



6. ábra.

Az 5. és a 6. ábra periodikus szerkezeteket, más néven oszcillátorokat mutat be. Az oszcillátor tulajdonsága, hogy k (periódus) lépésen belül visszatér az eredeti alakzatba. Az ábrázolt alakzatok periódusa 2.

A szabály változtatása

Halott cellák esetén

Ha csökkentem az élethez szükséges szomszédok számát, a cellák nagy része életre kel. Ha növelem, akkor nem figyelhető meg komolyabb változás.

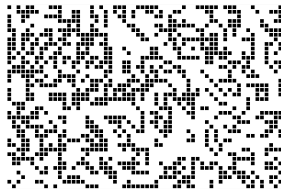
Élő cellák esetén

Ha csak a pontosan 3 szomszédval rendelkező cellák maradnak életben, a cellák nagy része meghal, egy oszcilláló szerkezet jön létre. Ha a 4 élő szomszédot is megengedem, akkor több oszcilláló is megjelenik.

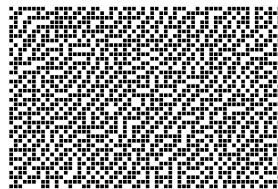
1 vagy 2 élő szomszéd esetén, az eredeti szabályrendszerhez képest kisebb stabil szerkezetek jönnek létre.

Ha kibővítem a szabályt úgy, hogy a cella 3 különböző esetben is életben maradhat a következő eredményeket kapom.

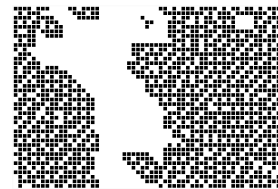
- 1, 2, 3 élő szomszéd $\rightarrow$ folyamatos változás (7. ábra)
- 2, 3, 4 élő szomszéd $\rightarrow$ szinte minden cella stabilan életben van, kevés oszcilláció figyelhető meg (8. ábra)
- 3, 4, 5 élő szomszéd $\rightarrow$ sokan életben maradnak, de üres területek, szigetek jönnek létre (9. ábra)



7. ábra.



8. ábra.



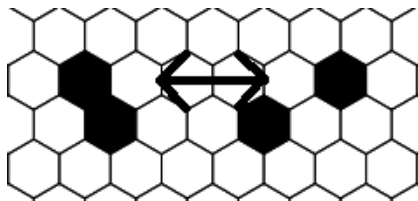
9. ábra.

Összehasonlítás az eredeti játékkal

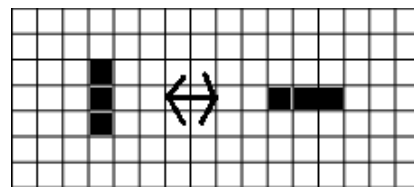
Ha a hexagonális játékot összehasonlítottam a hagyományos, négyzetrácsos Game of Life-fal, azt tapasztaltam, hogy a négyzetrácsos esetben lényegesen több oszcillátor és glider jelenik meg a végső kimeneten. A kapott eredmények és a különböző esetek alapján megkerestem azt a szabályrendszert, mely szerint a legközelebbi eredményt kapjuk az eredeti játékhoz.

Halott cellák esetén az életrekeléshez szükséges szomszédok számát csökkentem (3-ról 2-re), így több élő cellát kapunk. Ezzel ellentétben, élő cellák esetén az életben maradáshoz szükséges szomszédok számát növeltem (3-4-re), így több oszcilláció jön létre, a stabil szerkezetek száma pedig csökken.

Megfigyelve ezen oszcillációs struktúrákat, észrevehetjük, hogy vannak az eredeti játékhoz nagyon hasonló alakzatok. A 10. és a 11. ábra egymáshoz nagyon

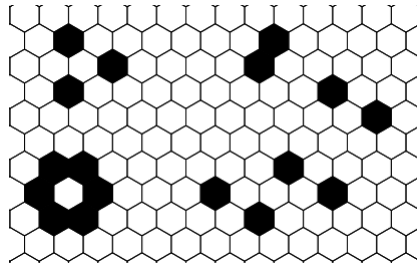


10. ábra.



11. ábra.

hasonló, 2 periódusú oszcillátorokat mutatnak be a hexagonális és a négyzetrácsos térben.



12. ábra. További oszcillációs struktúrák az új szabály szerint[4]

Összegzés

A Game of Life témaköre rengeteg érdekességet, kutatási lehetőséget tartalmaz. Céлом a beadandóval egy optimális szabály megtalálása volt a hexagonális játékhöz, mely a már jól ismert négyzet-alapú megvalósításhoz közeli eredményt ad. További vizsgálódás során még több érdekes struktúrát lehetne felfedezni. Mivel az eredeti játék Turing-teljes, így némi kutatás után a hexagonális változatról is beláthatnánk az univerzitást.

Irodalomjegyzék

- [1] Speciális Dinamikai Rendszerek: Sejtautomaták, Zsoldos János Available Online: https://web.cs.elte.hu/blobs/diplomamunkak/bsc_alkmat/2011/zsoldos_janos.pdf
- [2] 2D Hexagonal Cellular Automata: The Complexity of Forms, Vural Erdogan
- [3] Coding Challenge - The Game of Life Available Online: <https://thecodingtrain.com/CodingChallenges/085-the-game-of-life.html>
- [4] Hexlife - David Slaw Available Online: <https://davidsiaw.github.io/blog/2014/11/21/hexlife/>