

Projekt - Uogólniony Game of Life na siatkę trójkątnej

1. Wstęp - automaty komórkowe

Automaty komórkowe są narzędziem generującym nowe stany komórek na podstawie stanów danej komórki i jej sąsiadów w poprzednim kroku. Takie podejście pozwala na utworzenie układów, które same ewoluują, o ile tylko użytkownik dostarczy im zbiór zasad i stan początkowy.

Przykładem 2-wymiarowego automatu jest gra Game of Life, gdzie zmieniające się komórki przypominają poruszające się obiekty. Reguły zachowań komórki są tutaj określone przez liczbę żywych/aktywnych sąsiednich komórek i mówią czy w następnym kroku komórka pozostanie żywa (Stay), czy urodzi się, jeżeli jest nieaktywna (Born). Istnieją różne odmiany tej gry. Tematyką projektu jest implementacja Game of Life na siatkę trójkątną.

Użyte narzędzia:

- Javascript
 - Program korzysta z zewnętrznych bibliotek Chart.js i p5.js, które do załadowania wymagają połączenia z Internetem
- Html
- Css

2. Game of life na siatkę trójkątnej

W przypadku mojego programu, komórka posiada 6 sąsiadów. Przestrzeń automatu jest definiowana w 2 wymiarach, jednak podczas obliczeń można ją dla uproszczenia zrzutować do 1D. Wtedy dla komórki o indeksie k_i , sąsiedzi mają następujące indeksy:

- $k_i - w - 1$
- $k_i - w$,
- $k_i - 1$
- $k_i + 1$
- $k_i + w$
- $k_i + w + 1$

,gdzie w - szerokość przestrzeni 2D

Na rysunku 2.1 widoczna jest komórka oraz jej sąsiedzi na obrazku wyjściowym. Ich ustawienie jest różne dla parzystych i nieparzystych komórek.



Rysunek 2.1 - Zaznaczenie komórki i jej sąsiadów dla komórek typu parzystego i nieparzystego

Biorąc pod uwagę, że komórka może posiadać od 0 do 7 sąsiadów, oraz są dwa typy zasad - Stay i Born, zbiór zasad może być jednym z $2^7 * 2^7 = 16384$ różnych, które wpływają na ewolucję automatu, wraz ze stanem początkowym przestrzeni komórek.

3. Instrukcja obsługi programu

Program posiada proste GUI zawierające ustawienia, wyświetlacz wyniku i wyświetlacz wykresu statystycznego.

- **Ustawienia użytkownika:**

Użytkownik może podać wymiary przestrzeni, czas między poszczególnymi epokami, a także zbiór zasad dla automatu. Ustawienia są zatwierdzane w momencie naciśnięcia przycisku GENERATE. W tym momencie są również odblokowane przyciski akcji: START/STOP i STEP. Pierwszy z nich uruchamia ewolucję, aż do zatrzymania, a drugi wykonuje jeden krok ewolucyjny. Screen z poprawnymi ustawieniami widnieje na Rysunku 3.1.

Długość przestrzeni:

Szerokość przestrzeni:

Czas między epokami(ms):

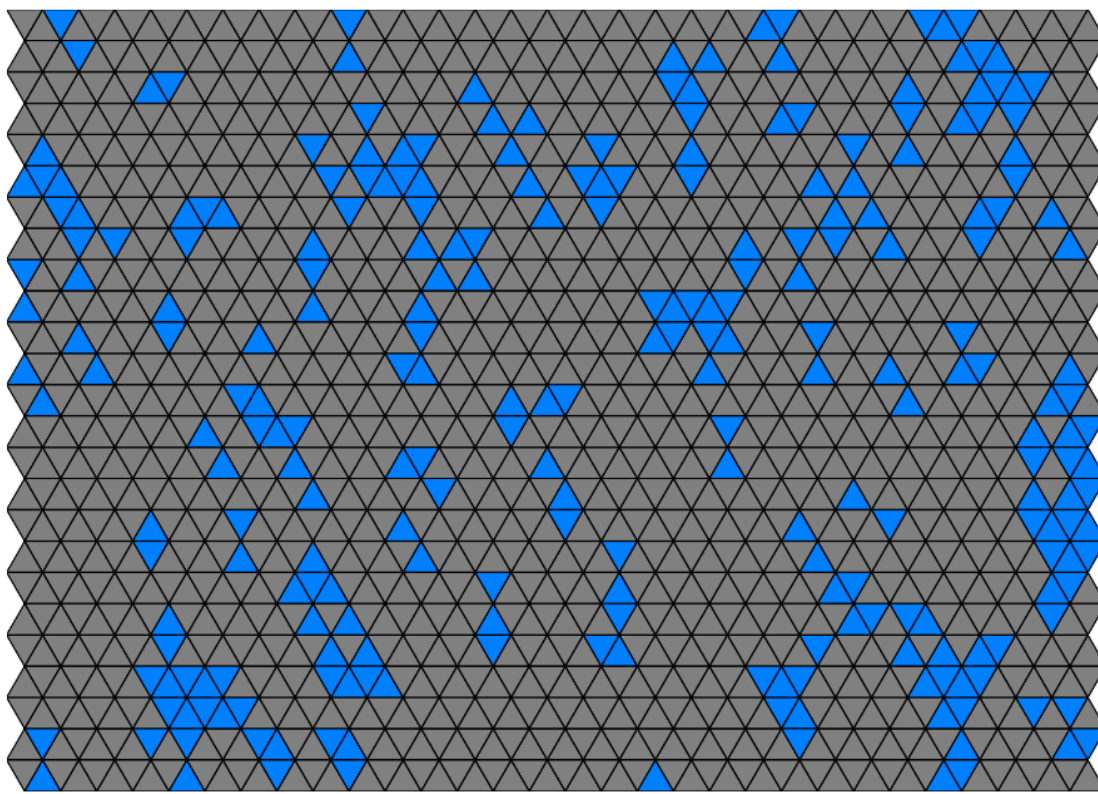
Żywa komórka pozostaje żywa(Ilość sąsiadów):
☐ 0 ☒ 1 ☒ 2 ☐ 3 ☒ 4 ☒ 5 ☐ 6

Powstaje nowa żywa komórka(Ilość sąsiadów):
☐ 0 ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☒ 5 ☒ 6

Rysunek 3.1 - Ustawienia użytkownika.

- **Wyświetlacz wyniku:**

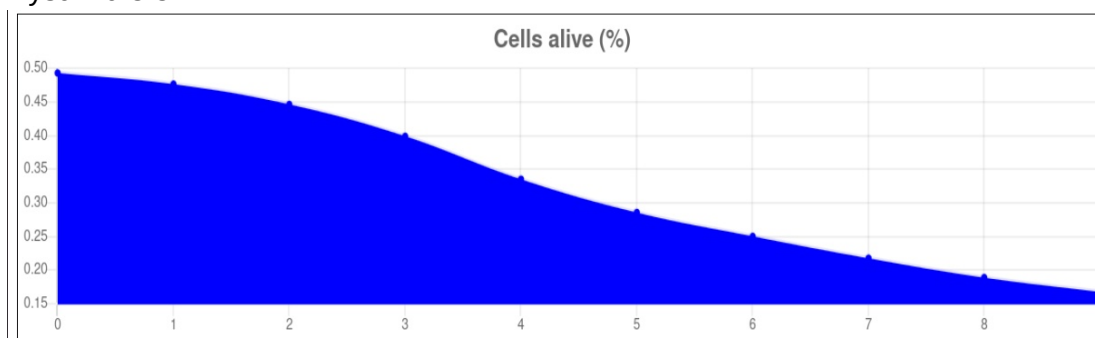
2-wymiarowa siatka trójkątnych komórek, które wizualizują ewolucję automatu. Wyświetlacz wyświetli się dopiero do naciśnięcia GENERATE i ruszy gdy użyjemy START lub STEP. Widoczny na Rysunku 3.2.



Rysunek 3.2 - Wynik widoczny na wyświetlaczu

- **Wyświetlacz wykresu statystycznego:**

Jest tu widoczny wykres przedstawiający % żywych komórek w kolejnych epokach. Wykres zmienia się po każdym kroku ewolucyjnym. Przykład na Rysunku 3.3.

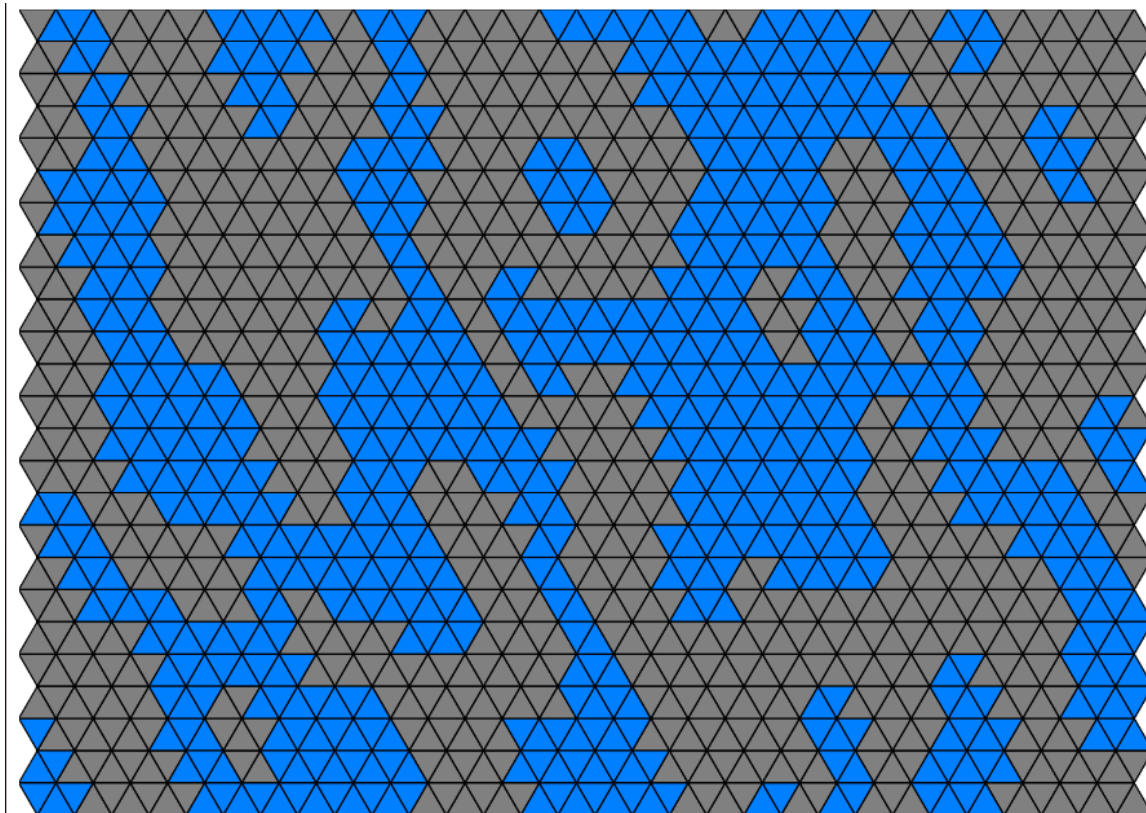


Rysunek 3.3 - Statystyka pokazująca procent żywych komórek w kolejnych epokach.

4. Ciekawe, charakterystyczne zachowania automatu

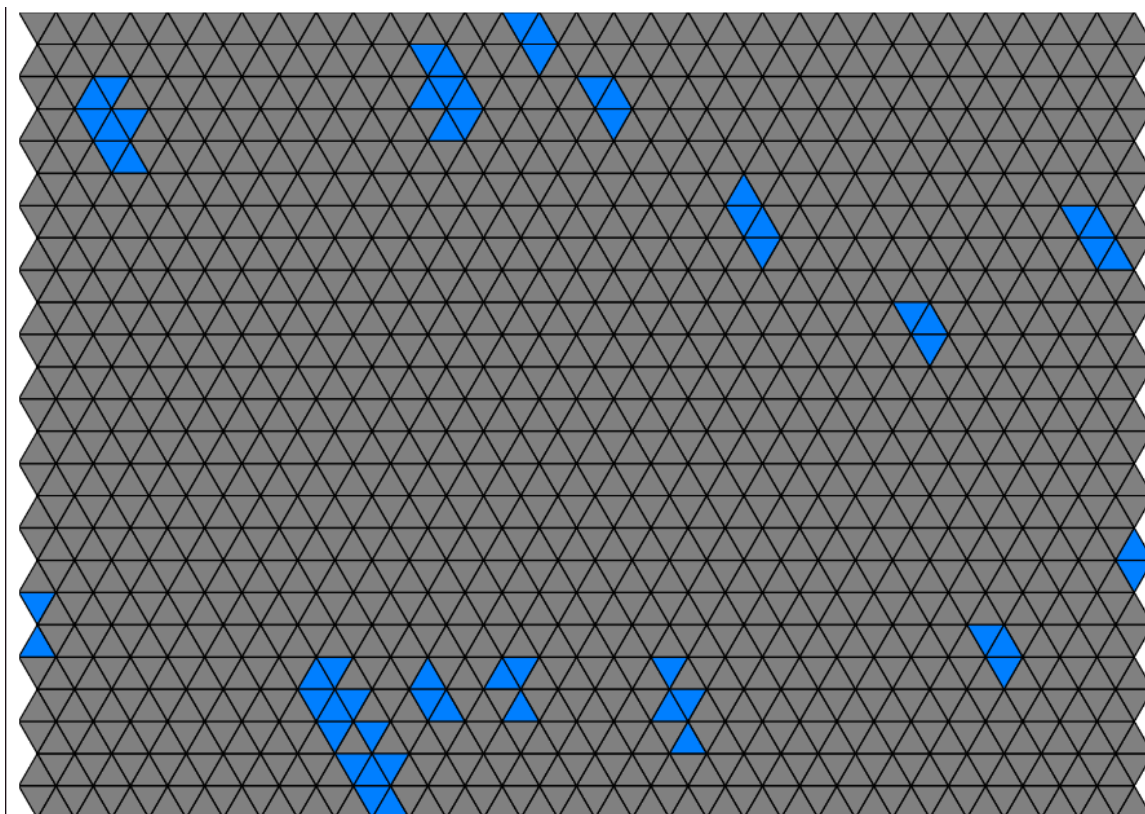
Podczas testowania programu zauważyłem, że pewne kombinacje zasad prowadzą do podobnych wyników. Poniżej przedstawiam kilka takich zasad. Wyniki zostały otrzymane dla losowego stanu początkowego, przy około 50% żywych osobników: ("..." oznacza dowolne poprzedzające lub następujące wartości)

- **Stay(..., 3,4,5,6), Born(4,5,6)** - ten zbiór doprowadza do gromadzenia się komórek w duże, ciągle obszary. Przykład na Rysunku 4.1.



Rysunek 4.1 - Stan końcowy dla automatu o zasadach Stay(3,4,5,6), Born(4,5,6)

- **Stay(2,3), Born(0,1,2)** - podobnie jak poprzednio, komórki gromadzą się w obszary, jednak tym razem są w stanie ciągłej oscylacji.
- **Stay(2,3), Born(3...)** - te zasady prowadzą do długich ewolucji, w wyniku których, w przestrzeni pozostają jedynie małe, charakterystyczne struktury. Mogą pojawić się migacze. Jeden z wyników zamieściłem na Rysunku 4.2.



Rysunek 4.2 - Stan końcowy dla automatu o zasadach Stay(2,3), Born(3,4,5,6)

- **Stay(1), Born(3...)** - taka konfiguracja też działa podobnie jak powyżej, jednak teraz migacze pojawiają się dużo częściej.