

# Symulacja procesów biologicznych - Laboratorium 4

## Automaty komórkowe

Zofia Ratajczak, Wiktoria Jagodzinska

### Implementacja

W implementacji skorzystano z języka Python oraz biblioteki Matplotlib do wizualizacji wyników.

### Warunki początkowe

- Siatka: 50 x 50 (zgodna z wymaganiami)
- Komórki przyjmują wartości 1 (żywa) lub 0 (martwa)

### Sąsiedztwo i warunki brzegowe

Zastosowano sąsiedztwo Moore'a (8 sąsiadów) z periodycznymi warunkami brzegowymi (tzn. krawędzie „zawijają się”)

### Reguły przejścia

1. Żywa komórka z  $<2$  żywymi sąsiadami umiera
2. Żywa komórka z 2–3 żywymi sąsiadami przeżywa
3. Żywa komórka z  $>3$  żywymi sąsiadami umiera
4. Martwa komórka z dokładnie 3 żywymi sąsiadami ożywa

### Aktualizacja stanu

Stosowana jest aktualizacja synchroniczna – cała siatka aktualizowana jest jednocześnie na podstawie poprzedniego stanu.

### Opis kodu

Program umożliwia wybór trybu symulacji:

- „1” – 3 wzorce ciągłego życia
- „2” – 2 oscylatory
- „3” – działo (glider gun)

- „4” – Matuzalec

Funkcja `animate()` tworzy animację ewolucji siatki na przestrzeni epok. Po każdej epokach wyświetlana jest liczba żywych komórek.

Animacja nie jest zapisywana, a wyświetlana przy każdym uruchomieniu funkcji.

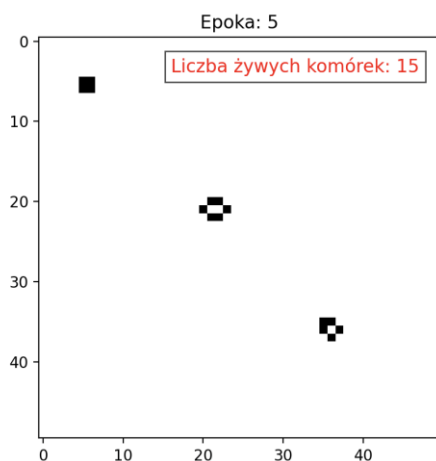
Podana jest ilość komórek na początku i na końcu symulacji, możliwe jest też jej śledzenie w czasie rzeczywistym.

Program jest ustawiony do ciągłej możliwości przeprowadzenia symulacji w wybranym trybie, zakończenie dopiero po wybraniu trybu “end”.

## Wyniki symulacji

### 1. Wzorce ciągłego życia

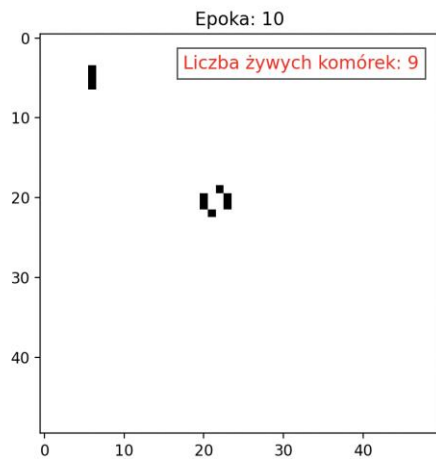
- Wzorce: blok, ul, łódź (lub inne wybrane)
- Liczba iteracji: min. 5
- Komórki pozostały bez zmian, co potwierdza ich „niezmiennosc”



*Wynik symulacji podczas symulacji ostatniej epoki (również dla poniższych)*

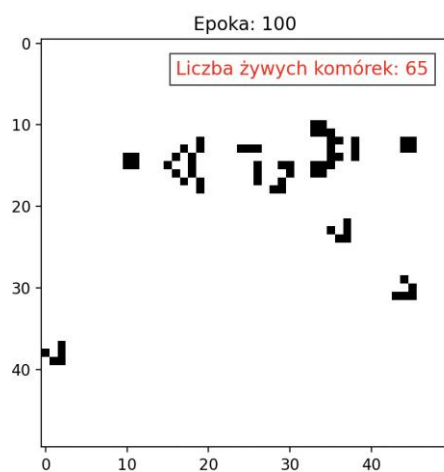
### 2. Oscylatory

- Wzorce: np. blinkier + żaba
- Iteracje: min. 10–20 (aby zobaczyć 5 pełnych cykli)
- Potwierdzona cykliczność wzorców



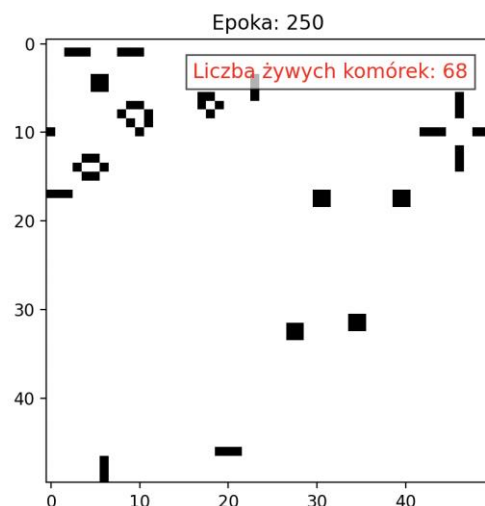
### 3. Działo (Glider gun)

- Liczba iteracji: min. 100
- Zaobserwowano co najmniej 3 glidery wystrzelone w przestrzeń



### 4. Matuzalec

- Złożony wzorec, który rozwija się długo zanim osiągnie stabilność
- Liczba iteracji: min. 100
- Obserwacja: na początku chaotyczny rozwój → stopniowe wygaszanie → stabilizacja (około 230 epokii)



## Wnioski

- Implementacja algorytmu działa poprawnie i odwzorowuje znane zachowania „Gry w życie”.
- Wyniki są zgodne z materiałami z wykładu.
- Zastosowanie bibliotek Pythona (NumPy, Matplotlib) ułatwiło analizę i wizualizację.
- Czas obliczeń był krótki – symulacje przebiegały płynnie dla siatki 50x50.
- Dla bardziej złożonych symulacji (np. Matuzalech) czas przetwarzania rośnie, ale nadal mieści się w akceptowalnym zakresie.

```
Wybierz symulację ('1' dla symulacji wzorców ciągłego życia, '2' dla symulacji oscylatorów, '3' dla symulacji wybranego działu, '4' dla symulacji Matuzalecha lub 'end' aby zakończyć): 1
Symulacja wzorców ciągłego życia:
Początkowa liczba żywych komórek: 15
2025-06-29 22:00:49.748 Python[75957:2752320] +[IMKClient subclass]: chose IMKClient_Modern
2025-06-29 22:00:49.748 Python[75957:2752320] +[IMKInputSession subclass]: chose IMKInputSession_Modern
Końcowa liczba żywych komórek: 15
Wybierz symulację ('1' dla symulacji wzorców ciągłego życia, '2' dla symulacji oscylatorów, '3' dla symulacji wybranego działu, '4' dla symulacji Matuzalecha lub 'end' aby zakończyć): 2
Symulacja oscylatorów:
Początkowa liczba żywych komórek: 9
Końcowa liczba żywych komórek: 9
Wybierz symulację ('1' dla symulacji wzorców ciągłego życia, '2' dla symulacji oscylatorów, '3' dla symulacji wybranego działu, '4' dla symulacji Matuzalecha lub 'end' aby zakończyć): 3
Symulacja działu:
Początkowa liczba żywych komórek: 36
Końcowa liczba żywych komórek: 65
Wybierz symulację ('1' dla symulacji wzorców ciągłego życia, '2' dla symulacji oscylatorów, '3' dla symulacji wybranego działu, '4' dla symulacji Matuzalecha lub 'end' aby zakończyć): 4
Symulacja Matuzalecha:
Początkowa liczba żywych komórek: 5
Końcowa liczba żywych komórek: 68
Wybierz symulację ('1' dla symulacji wzorców ciągłego życia, '2' dla symulacji oscylatorów, '3' dla symulacji wybranego działu, '4' dla symulacji Matuzalecha lub 'end' aby zakończyć): end
Koniec programu.
```

Komunikat wyświetlany użytkownikowi podczas symulacji w każdym trybie.

(linie 5 i 6 są wewnętrznymi komunikatami komputera nie mającymi wpływu na kod)