Więcej, niż Gra w Życie (SmoothLife)

Modelowanie komputerowe

Maciej Matyka

Plan

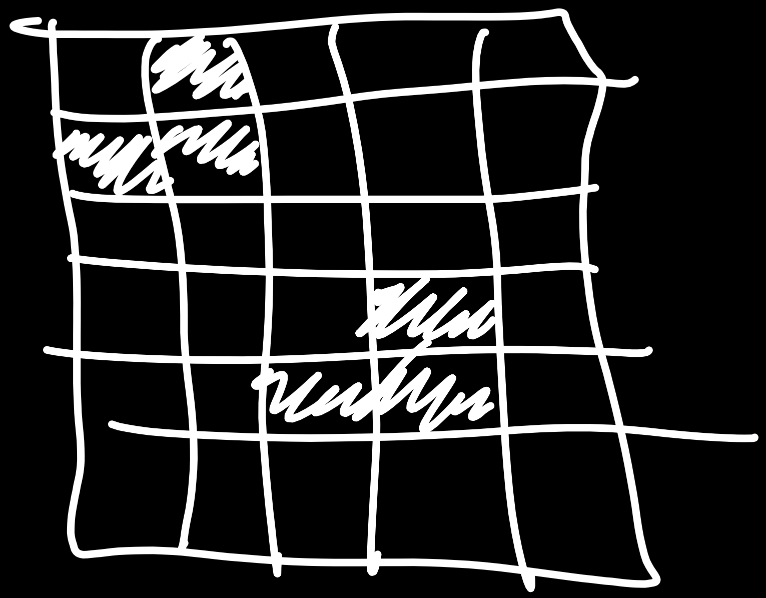
- Automaty komórkowe ogólnie
- Sąsiedztwo
- Plamy
- Gra w życie
- Piasek
- Gaz HPP
- Więcej, niż gra w życie (Larger than life)
- SmoothLife
- Elementarne Automaty Komórkowe (Wolfram)

Automat komórkowy

Proste zasady lokalne -> globalne efekty w układzie

Czym jest **automat komórkowy?**

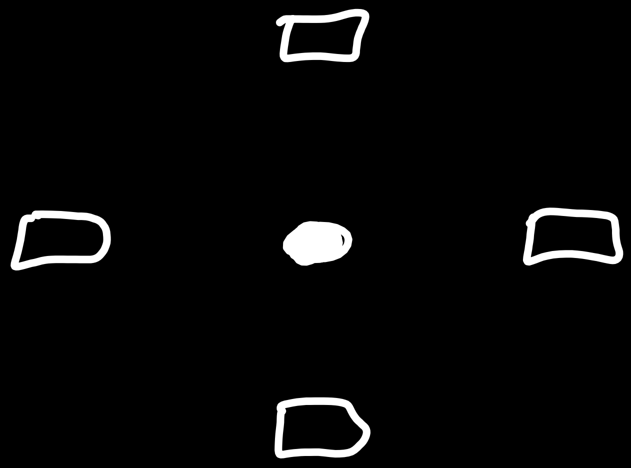
- Model **dyskretny** w czasie
- Model **dyskretny** w przestrzeni (1D, 2D itd.)
- Stan układu = wartości w komórkach
- **Synchroniczna** zmiana stanu komórek
- Zmiana stanu komórki zgodnie z **regułą**
- **Reguła** - funkcja stanu komórki i otoczenia



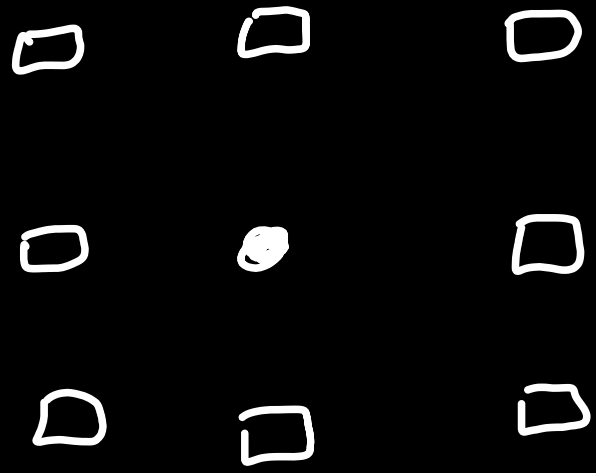
Historia

- Stanisław Ulam
  https://en.wikipedia.org/wiki/Stanislaw_Ulam
- John von Neumann - 1948 -
  https://en.wikipedia.org/wiki/John_von_Neumann
- John Conway – 1970 – Gra w życie
  https://en.wikipedia.org/wiki/John_Horton_Conway
- Stephen Wolfram – New Kind of Science – 2002
  https://en.wikipedia.org/wiki/A_New_Kind_of_Science

Sąsiedztwo



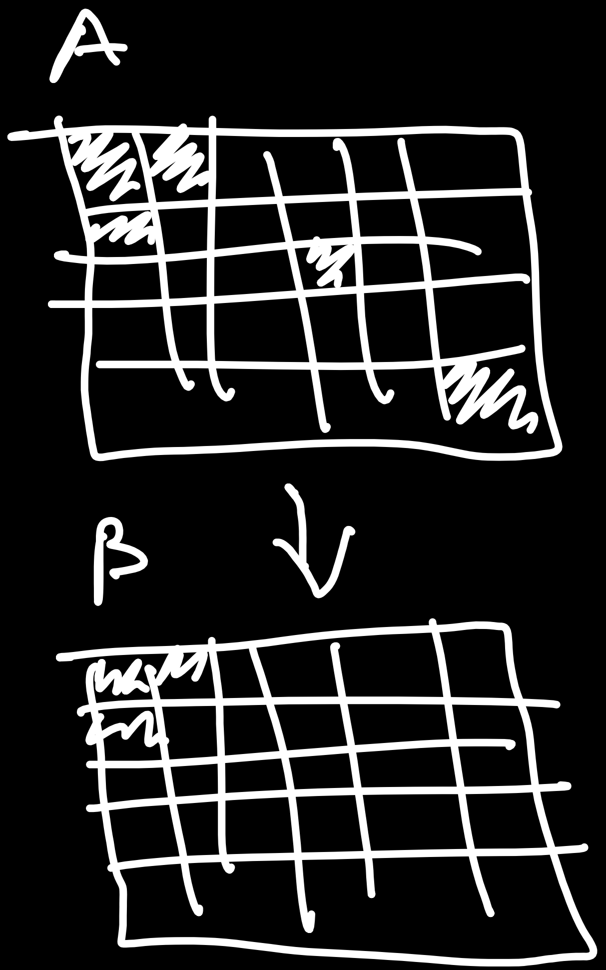
von Neumanna



Moore'a

Algorytm Ping-Pong

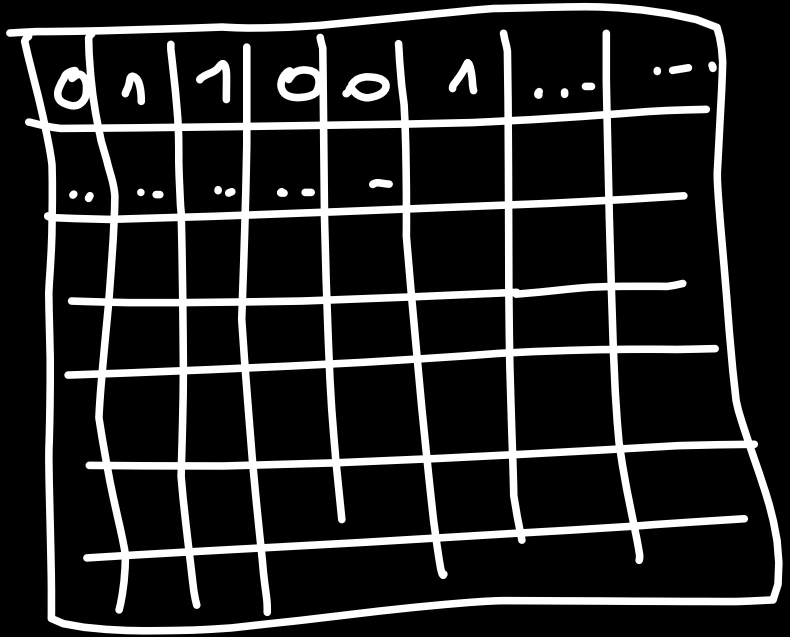
- Inicjalizuj układ
- Nałóż warunek początkowy (stan t=0)
- Dla każdej komórki ze stanu A (synchronicznie):
  - Oblicz regułę z sąsiedztwa
  - Zmień stan komórki
  - Zapisz jej wartość do stanu B
- Skopiuj stan B do stanu A (lub zamień)
- Wizualizacja, obliczenia
- Powtórz pętlę



Paski na zebrze?

Plamy - najprostszy model

- Losowy stan początkowy p=0.5

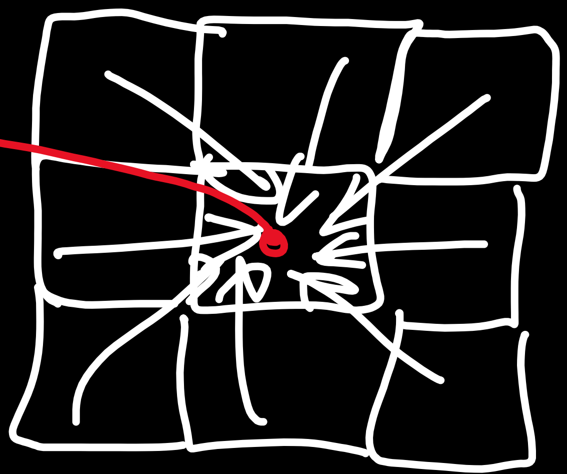


l, B. and Droz, M., Cellular Automata Modeling of Physical Systems,(1998). *Cambridge UP*, pp.122-137.

Algorytm

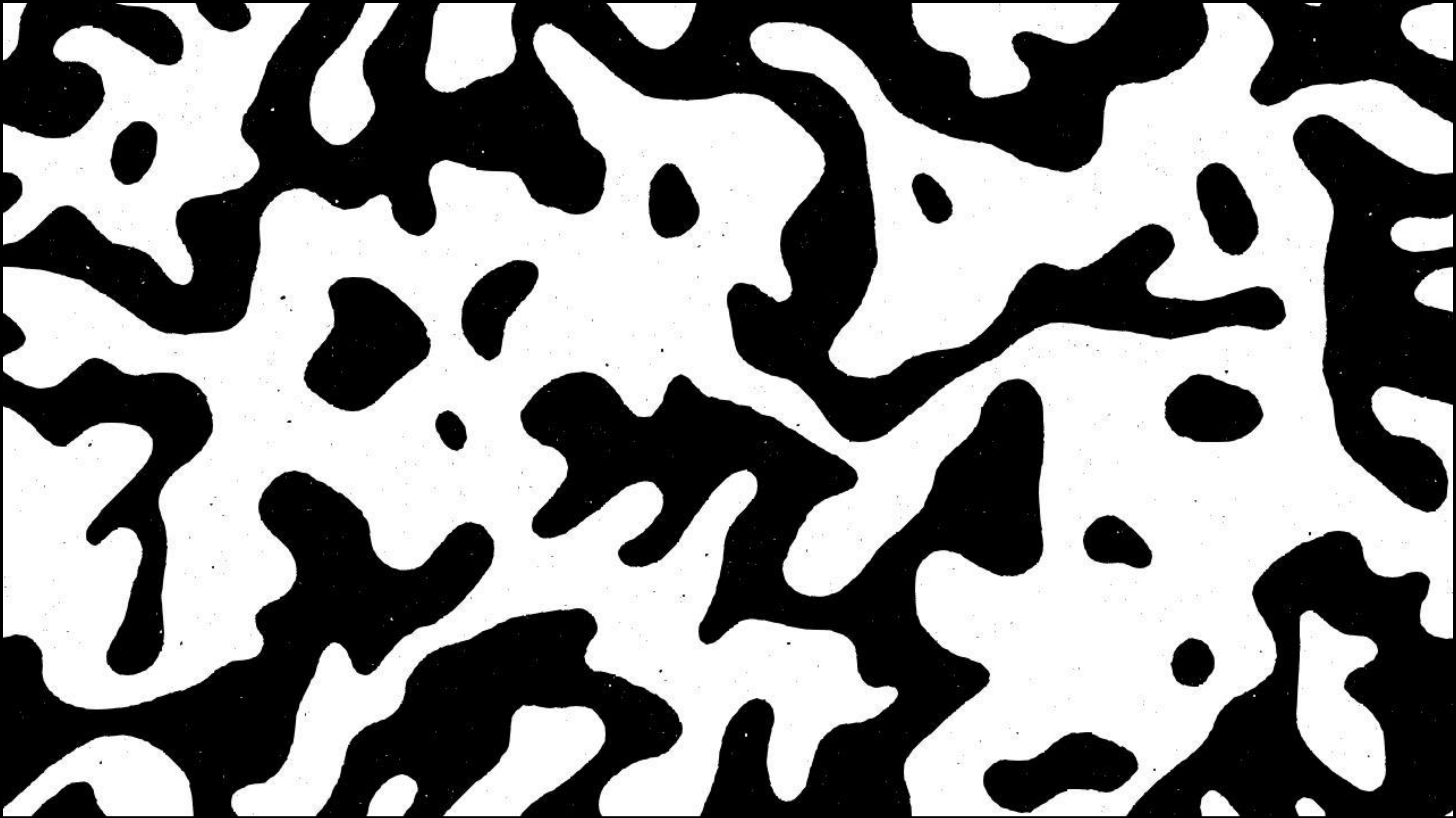
- Policz sumę sąsiadów komórki
- Ustal jej nową wartość:

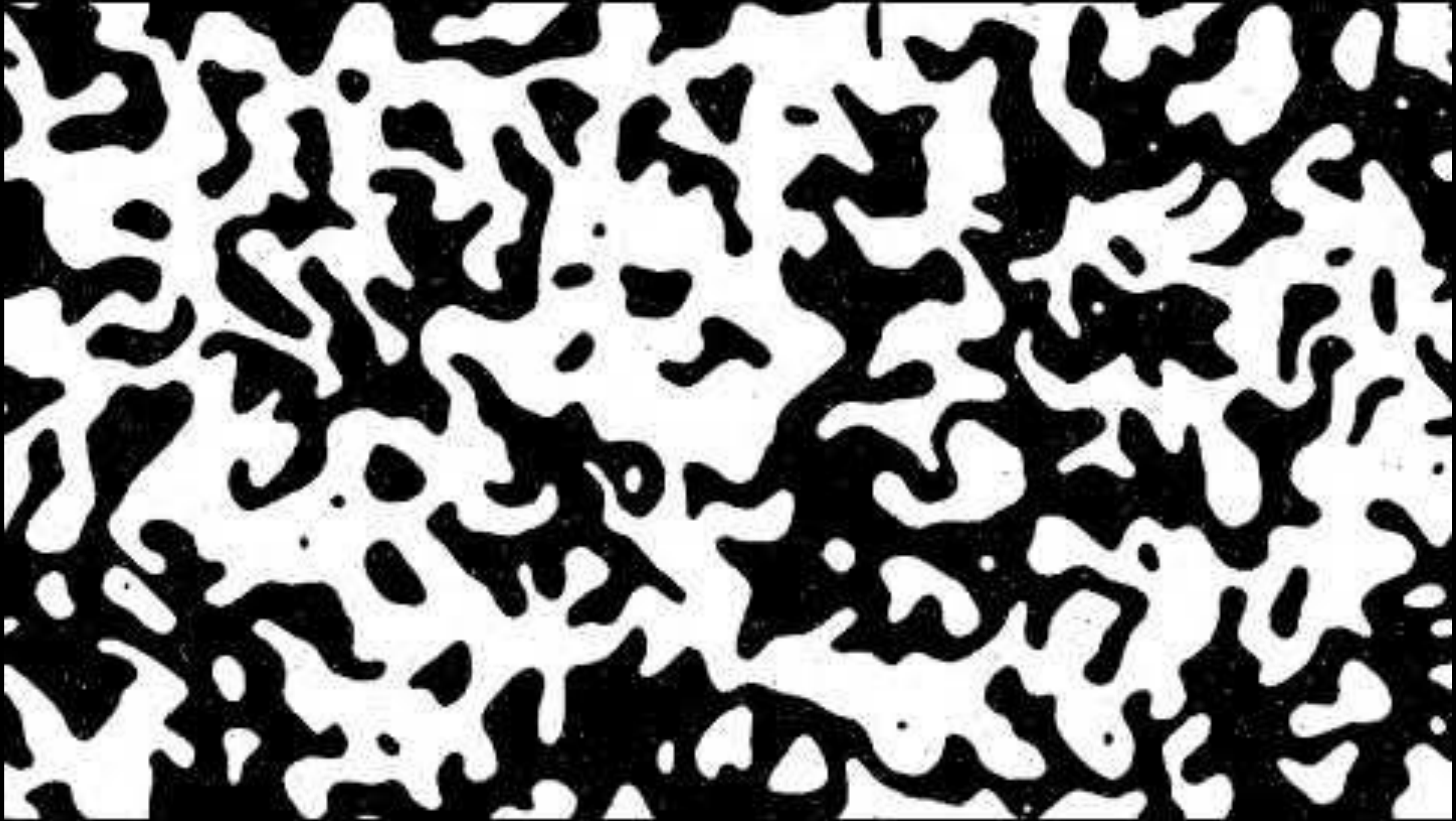
s =	0	1	2	3	4	5	6	7	8	9
	0	0	0	0	1	0	1	1	1	1



-

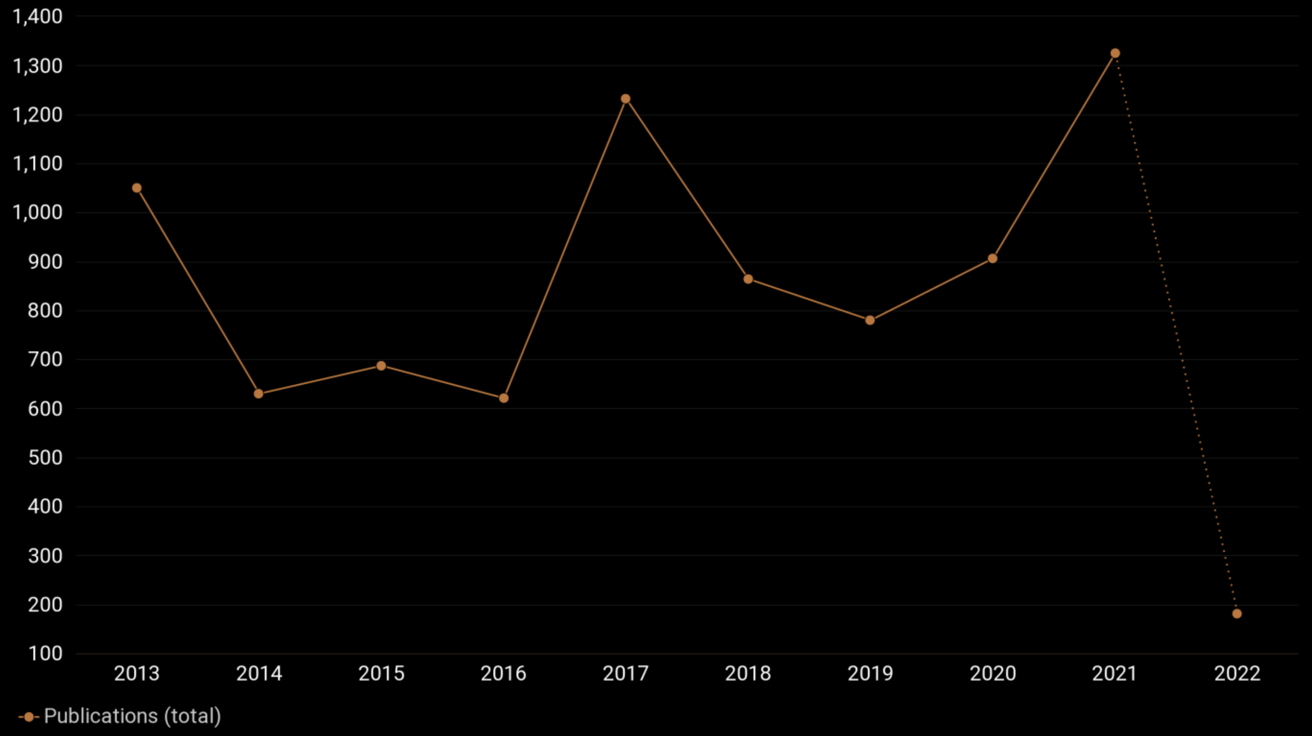
l, B. and Droz, M., Cellular Automata Modeling of Physical Systems,(1998). *Cambridge UP*, pp.122-137.





Gra w życie

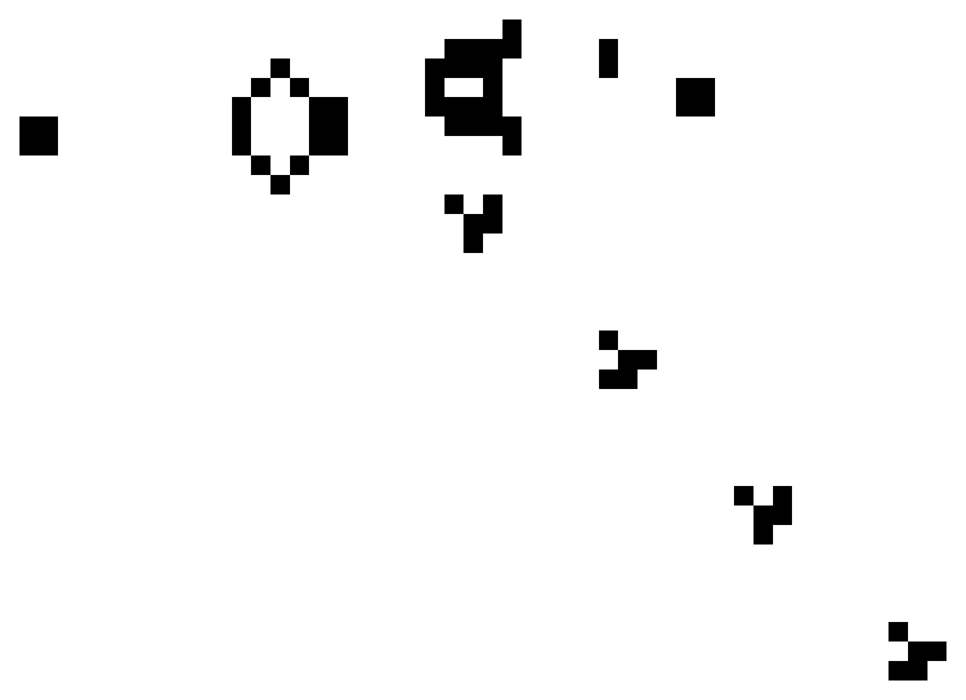
Game of Life



Source: https://app.dimensions.ai
Exported: April 27, 2022
Criteria: "game of life" in full data.

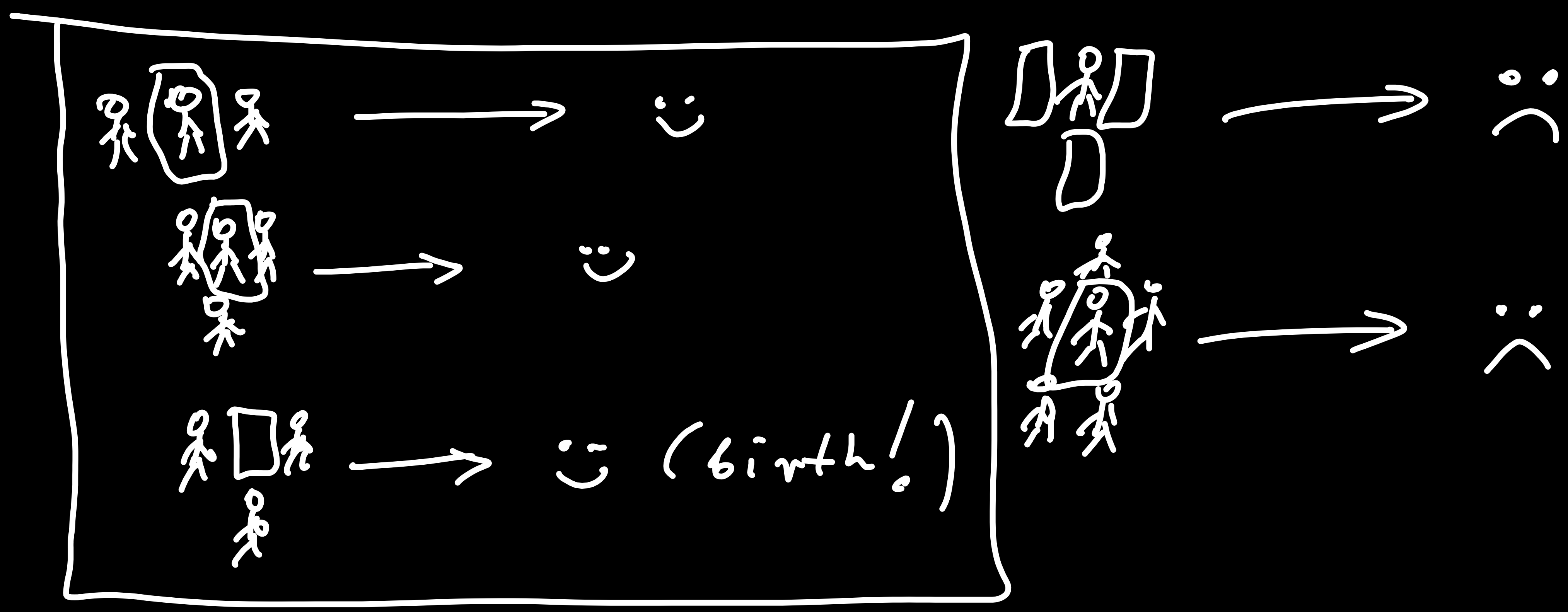
- John Conway (1937-2020)
- Automat komórkowy
- Żywe/martwe komórki typu boolowskiego (0 lub 1)

Gra w życie



John Conway (1937-2020)

Gra w życie

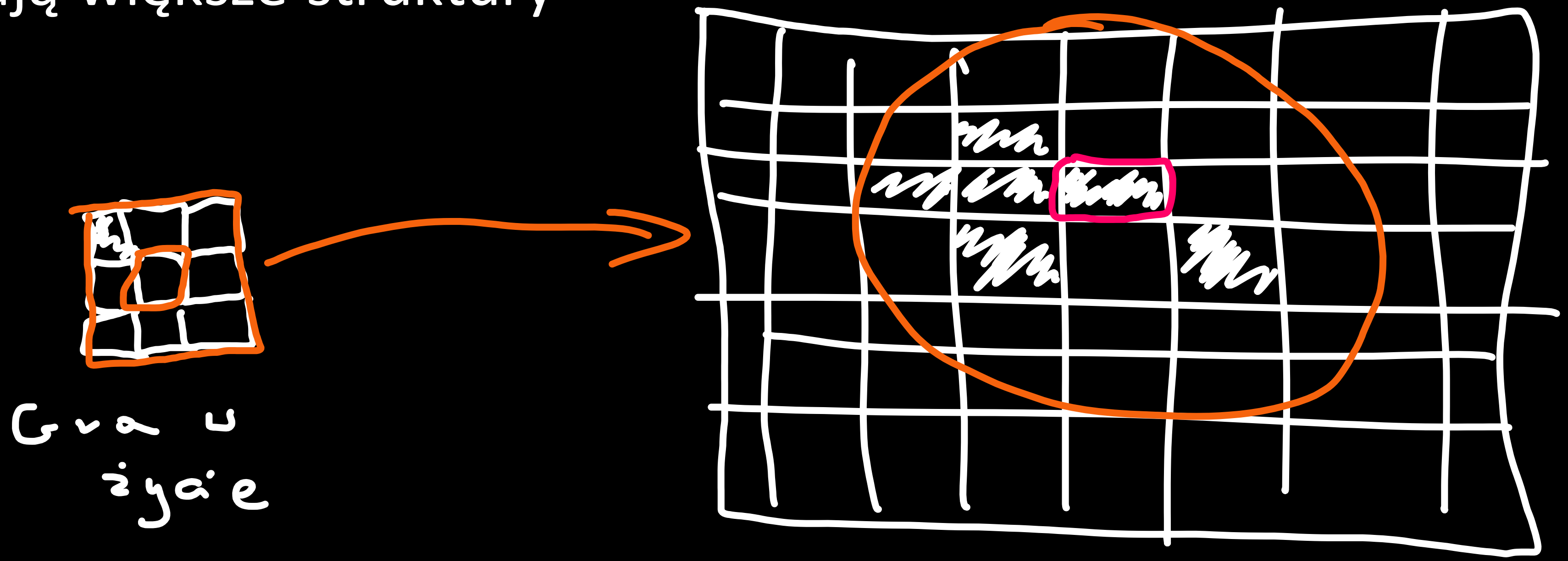


https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life

Więcej, niż gra w życie

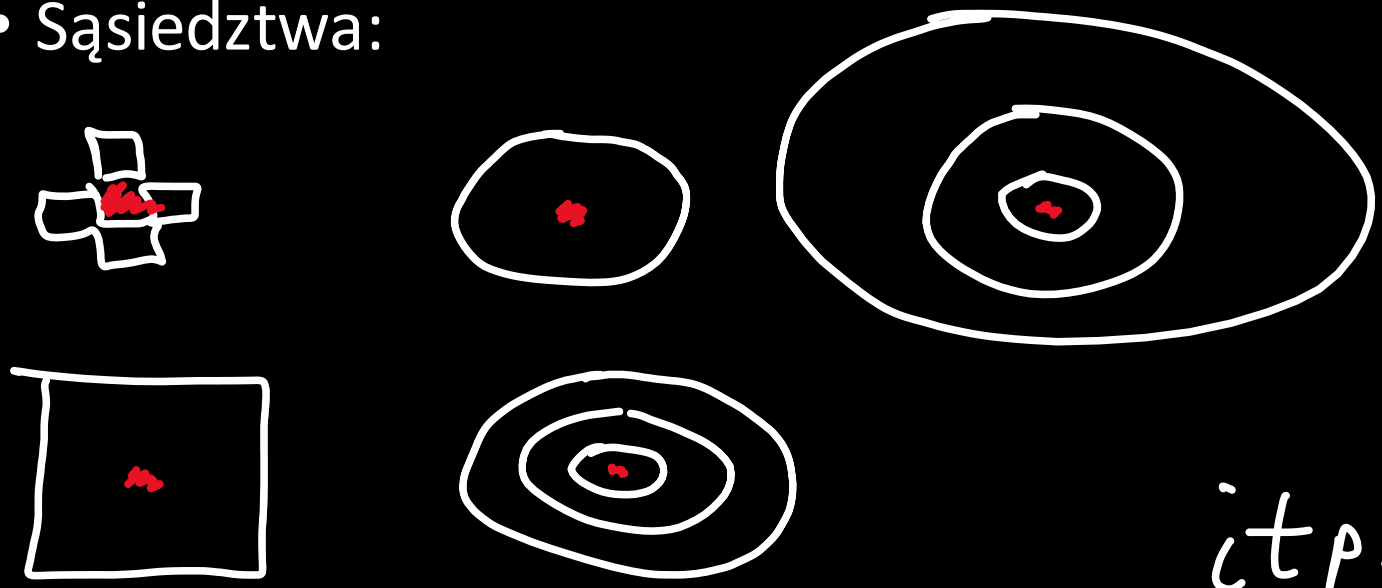
"Larger than life"

- **Rozszerzone sąsiedztwo**
- Bardziej skomplikowane reguły przeżywania i narodzin
- Powstają większe struktury



"Larger than life"

- Sąsiedztwa:



itp.

- Cała sztuka polega na dobraniu odpowiedniej funkcji przeżycia.

Larger than life

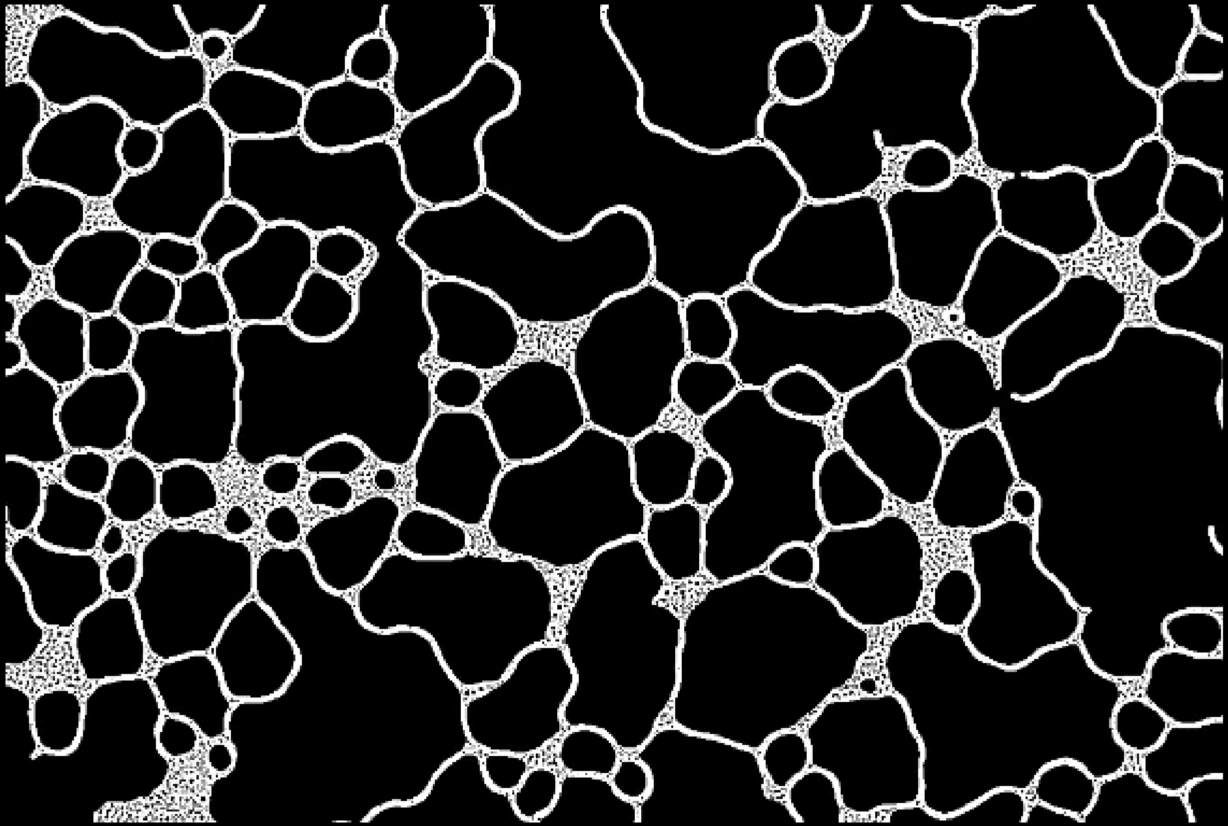
- R=5 ("bugs" - kwadratowe sąsiedztwo)
- S >= 34 && s <= 58 -> komórka przeżywa (1->1)
- S>=34 && s<=45 -> komórka się rodzi (0 lub 1 -> 1)

- Ciekawe źródła:

https://www.youtube.com/c/slackermanz
https://tatasz.github.io/compound_ca/
https://youtu.be/RZyy5bwXrn8

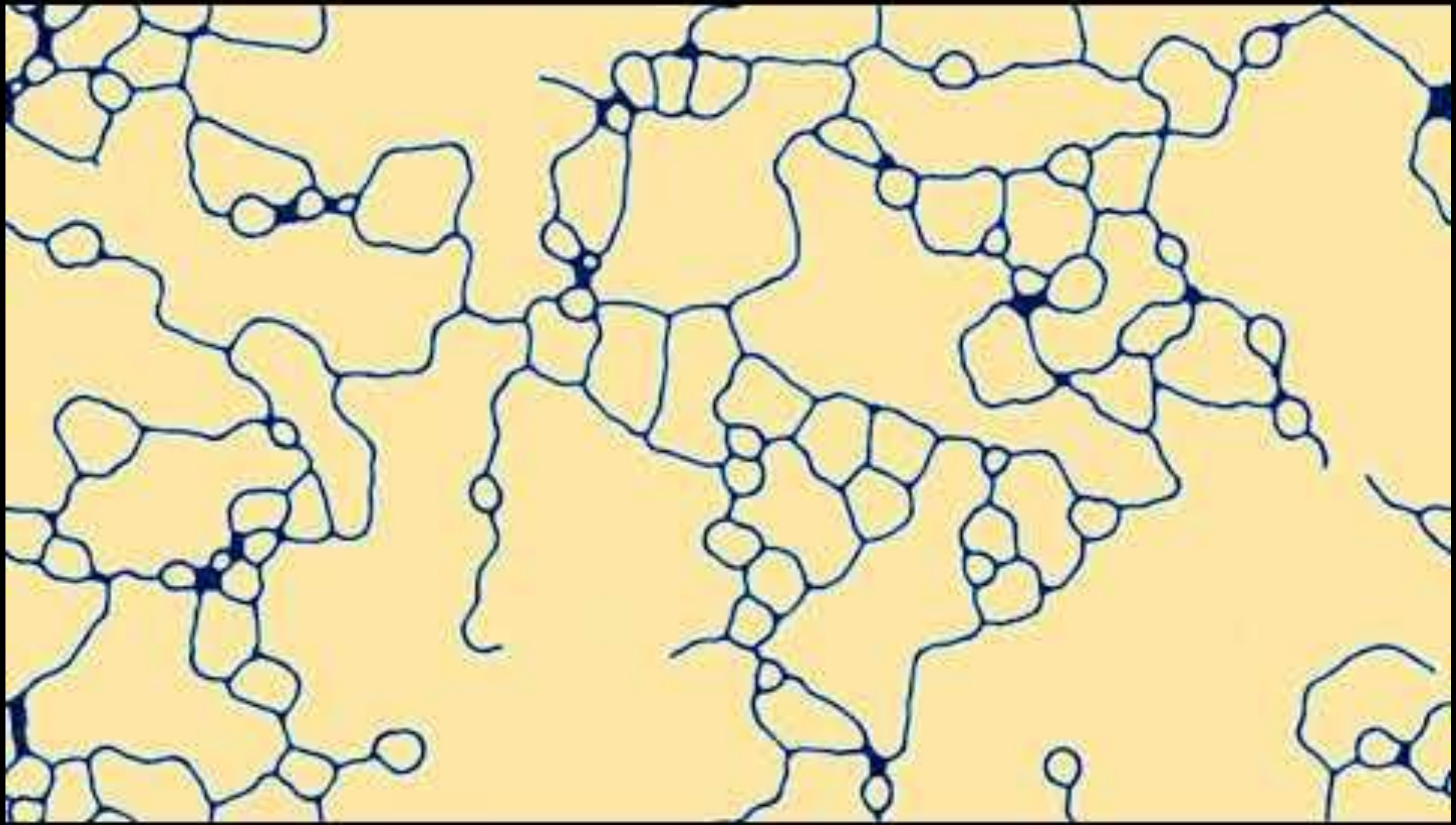
Larger than life

Wrażliwość na sąsiedztwo i szczegóły



Sąsiedztwo kołowe, r=5, warunki:
przeżywa dla 34 <= s <= 58, rodzi się 34<s<45

Sąsiedztwo kwadratowe, r=5, warunki:
przeżywa dla 34 <= s <= 58, rodzi się 34<=s<=45



Struktury klastrowe w "Larger than life"

Glidery w "Larger than life" -
https://youtu.be/wRGPZl7NcKI

Multiple Neighborhood Cellular Automata

- MNCA - uogólnienie automatów komórkowych
- Duża dowolność w doborze sąsiedztw
- Możliwe stany binarne lub ciągłe (0/1 lub 0-1)

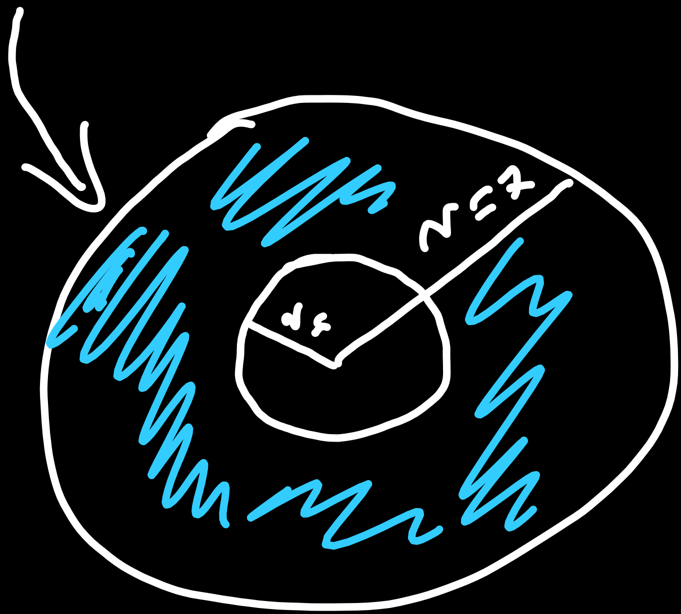
Zachowania odkryte w MNCA

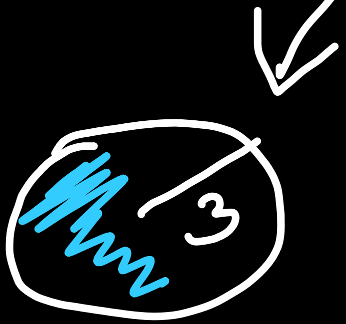
- Tworzenie pojedynczych "osobników"
- Siły oddziaływania (przyciąganie / odpychanie)
- Tworzenie silnych wiązań między obiektami
- Ruch kolektywny (flocking)
- Replikacja form
- Oddziaływania elastyczne
- Tworzenie kryształów
- W niektórych wariantach spełnione zasady zachowania (np. masy)
-

(za https://slackermanz.com/understanding-multiple-neighborhood-cellular-automata/)

Przykład 1 MCNA, "cells"

- Dwa sąsiedztwa
  - Sąsiedztwo 1: koło o promieniu 3
  - Sąsiedztwo 2: koło o promieniu 7 bez sąsiedztwa 1





"Cells"

```
if (avg >= 0.120   && avg <= 0.150) { res = 0; }
if (avg >= 0.210 && avg <= 0.220) { res = 1; }
if (avg >= 0.350 && avg <= 0.500) { res = 0; }
if (avg >= 0.750 && avg <= 0.850) { res = 0; }
```

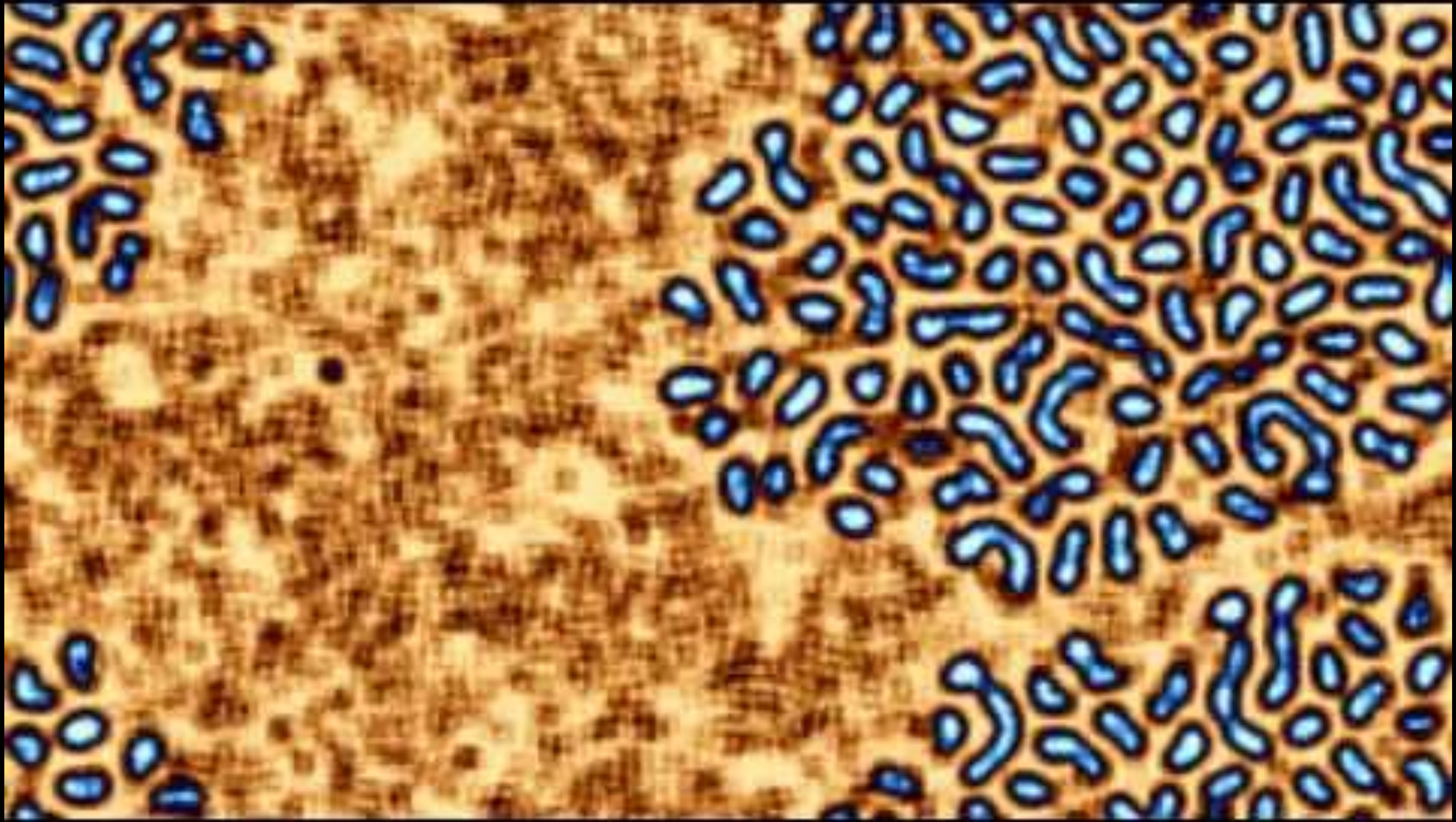
Sąsiedztwo 1

```
if (avg2 >= 0.100 && avg2 <= 0.280) { res = 0; }
if (avg2 >= 0.430 && avg2 <= 0.550) { res = 1; }
```

sąsiedztwo 2

Algorytm

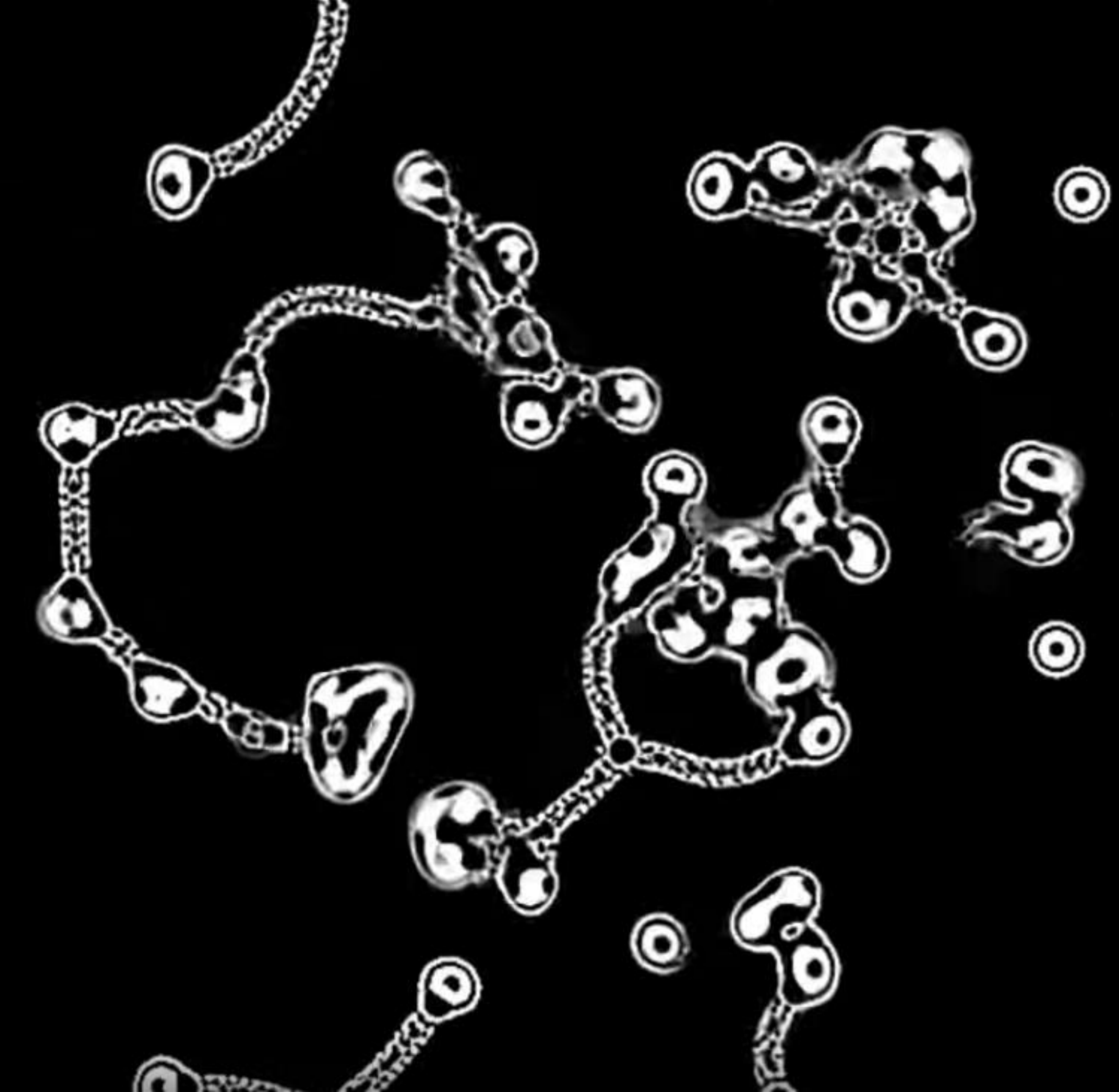
1. Inicjalizuj tablice
2. Sumuj wartości z sąsiedztwa
3. Oblicz średnie avg/avg2
4. Zastosuj reguły (umieranie, przeżywanie)
5. Rysuj i powtarzaj od 2.



"Cells MNCA"

Przegląd:

https://slackermanz.com/understanding-multiple-neighborhood-cellular-automata/



Smooth Life*

Generalization of Conway's "Game of Life" to a continuous domain - SmoothLife

Stephan Rafler
Nürnberg, Germany
frlndmr@web.de

December 8, 2011

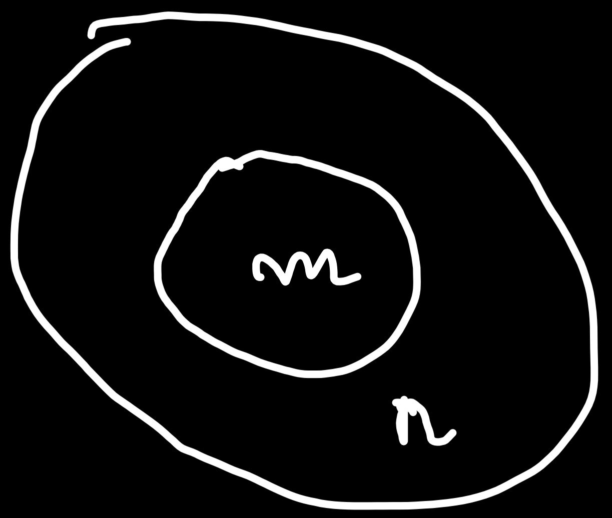
Abstract

We present what we argue is the generic generalization of Conway's "Game of Life" to a continuous domain. We describe the theoretical model and the explicit implementation on a computer.

https://arxiv.org/abs/1111.1567
- przegląd pracy

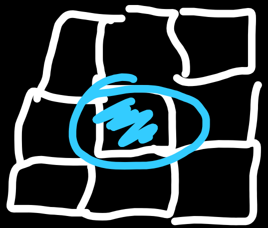
SmoothLife

- Model podobny do "Larger than Life" MNCA
- Wartości 0-1 w komórkach (ciągłe)
- Całkowanie po dwóch przyległych otoczeniach **m** i **n**

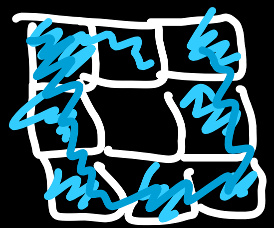


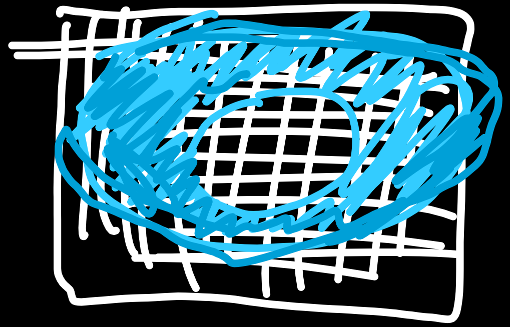
Game of Life vs SmoothLife

- Wartość komórki - żywa/martwa



- Sąsiedztwo





SmoothLife

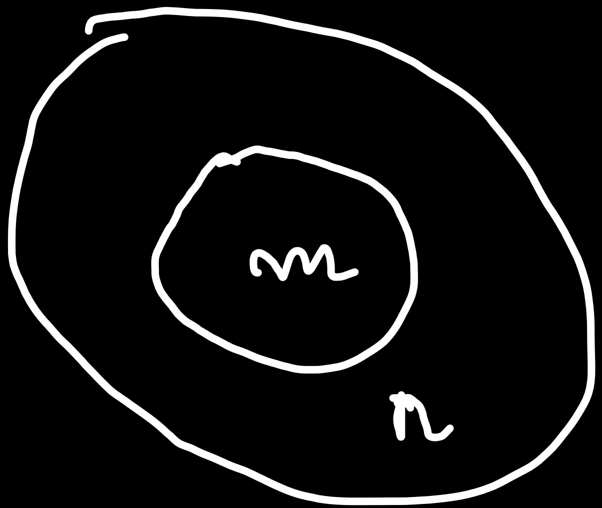
we can determine the filling of the cell or ”inner filling” m by the integral

$$m = \frac{1}{M} \int\limits_{|\vec{u}|<r_i} d\vec{u}\, f(\vec{x}+\vec{u},t) \tag{1}$$

and the neighborhood or ”outer filling” n by the integral

$$n = \frac{1}{N} \int\limits_{r_i<|\vec{u}|<r_a} d\vec{u}\, f(\vec{x}+\vec{u},t) \tag{2}$$

where N and M are normalization factors such that the filling is between



SmoothLife

- Funkcja przejścia ("transition function") postaci σ(n,m)

smoothness. For example we could define

$$\sigma_1(x, a) = \frac{1}{1 + \exp(-(x-a)4/\alpha)} \tag{3}$$

$$\sigma_2(x, a, b) = \sigma_1(x, a)\ (1 - \sigma_1(x, b)) \tag{4}$$

$$\sigma_m(x, y, m) = x(1 - \sigma_1(m, 0.5)) + y\,\sigma_1(m, 0.5) \tag{5}$$

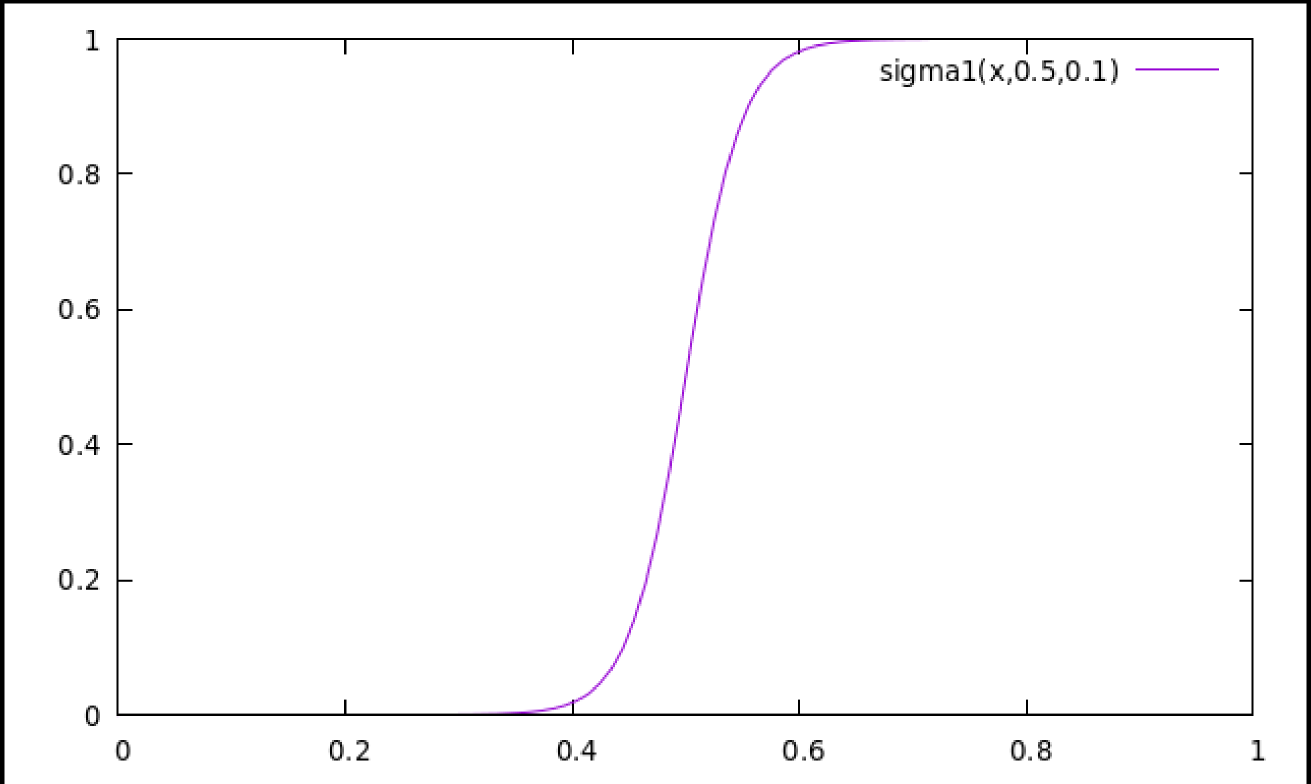
then we can define the transition function as

$$s(n, m) = \sigma_2(n, \sigma_m(b_1, d_1, m), \sigma_m(b_2, d_2, m)) \tag{6}$$

where birth and death intervals are given by $[b_1, b_2]$ and $[d_1, d_2]$ respectively.

(uwaga: w praktyce lepiej zamienić sigma2 z sigmam – o tym dalej)

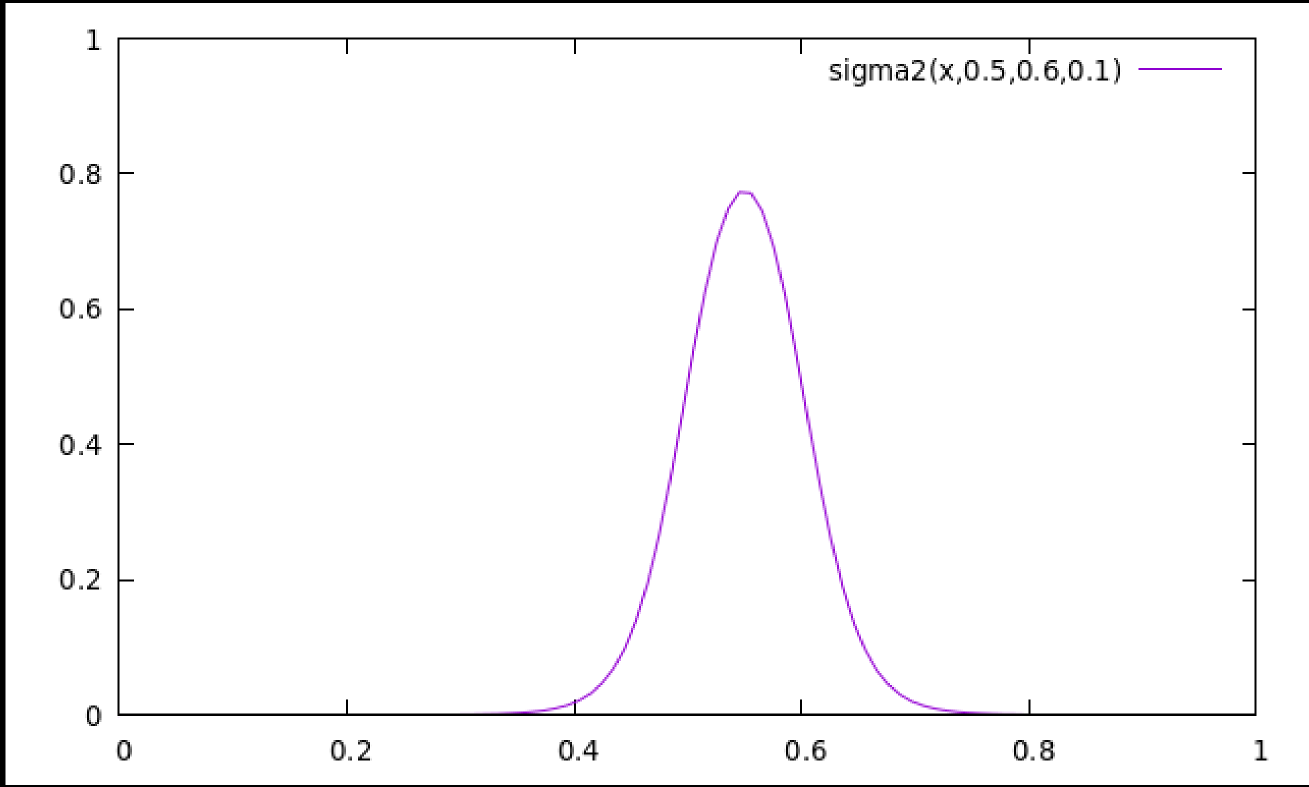
σ1(x, a, α)



$$\sigma_1(x, a) = \frac{1}{1 + \exp(-(x - a)4/\alpha)}$$

sigma1(x, a, alpha) = 1.0 / (1.0 + exp(-(x - a) * 4 / alpha))
plot [0:1] σ1(x,0.5,0.1)

σ2(x, a, α)

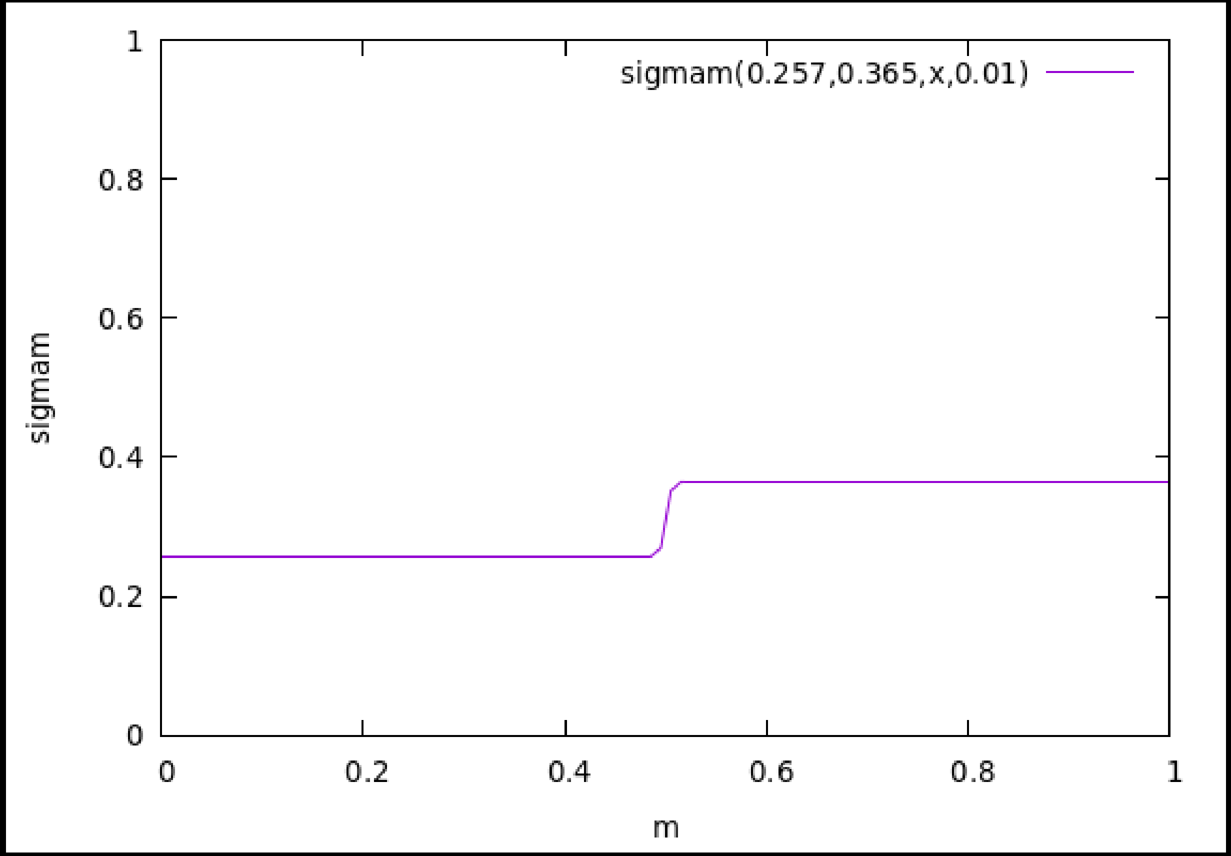


$$\sigma_2(x, a, b) = \sigma_1(x, a)\ (1 - \sigma_1(x, b))$$

sigma2(x,a,b,alpha)=sigma1(x,a,alpha)*(1-sigma1(x,b,alpha))
plot [0:1] σ1(x,0.5, 0.6, 0.1)

σ**m**(b, d, **m**, α)

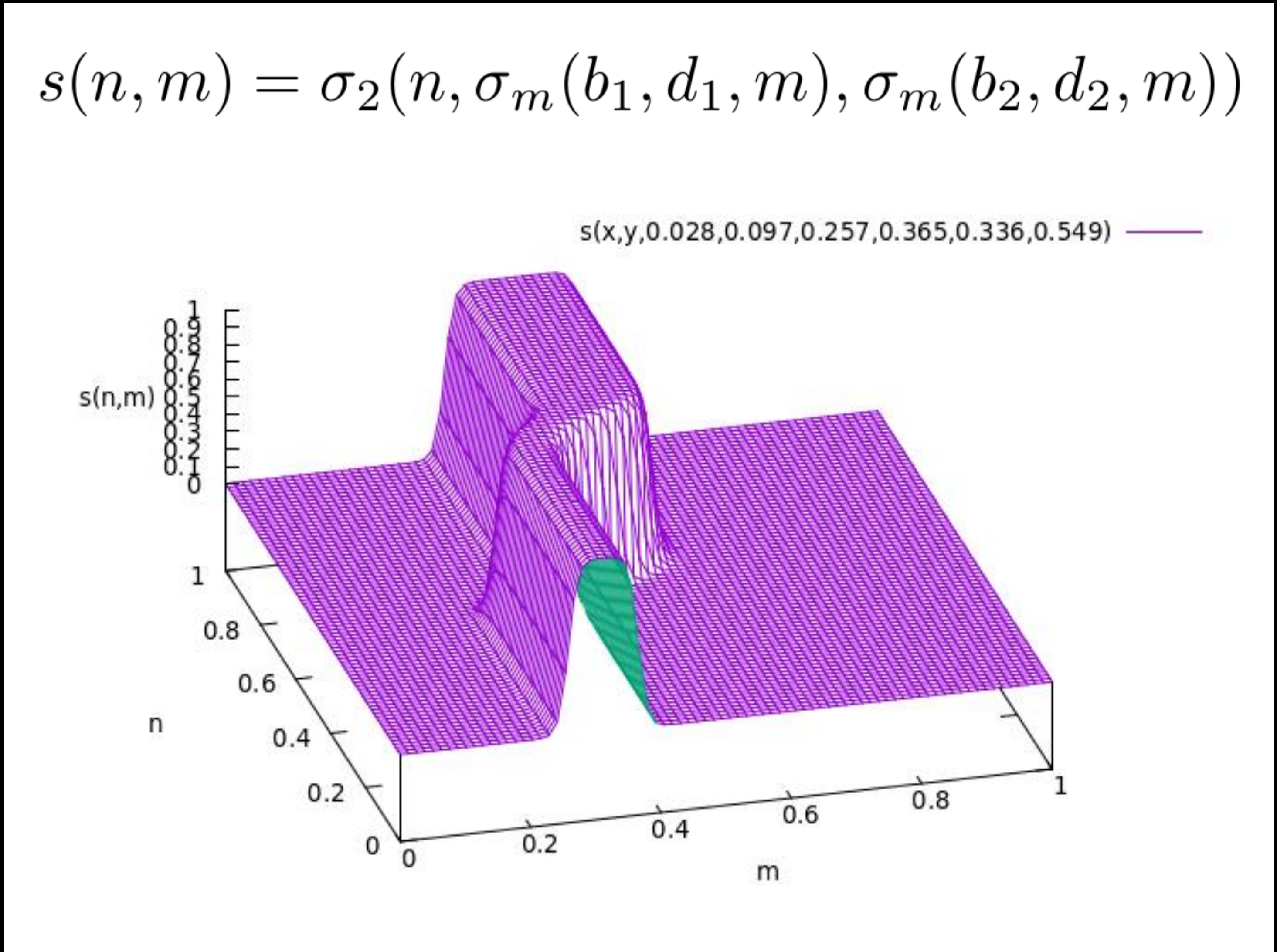
$$\sigma_m(x, y, m) = x(1 - \sigma_1(m, 0.5)) + y\,\sigma_1(m, 0.5)$$



sigmam(b,d,m,alpha) = b*(1-sigma1(m,0.5,alpha)) + d*sigma1(m,0.5,alpha)

plot [0:1] [0:1] sigmam(0.257,0.365,x,0.01)

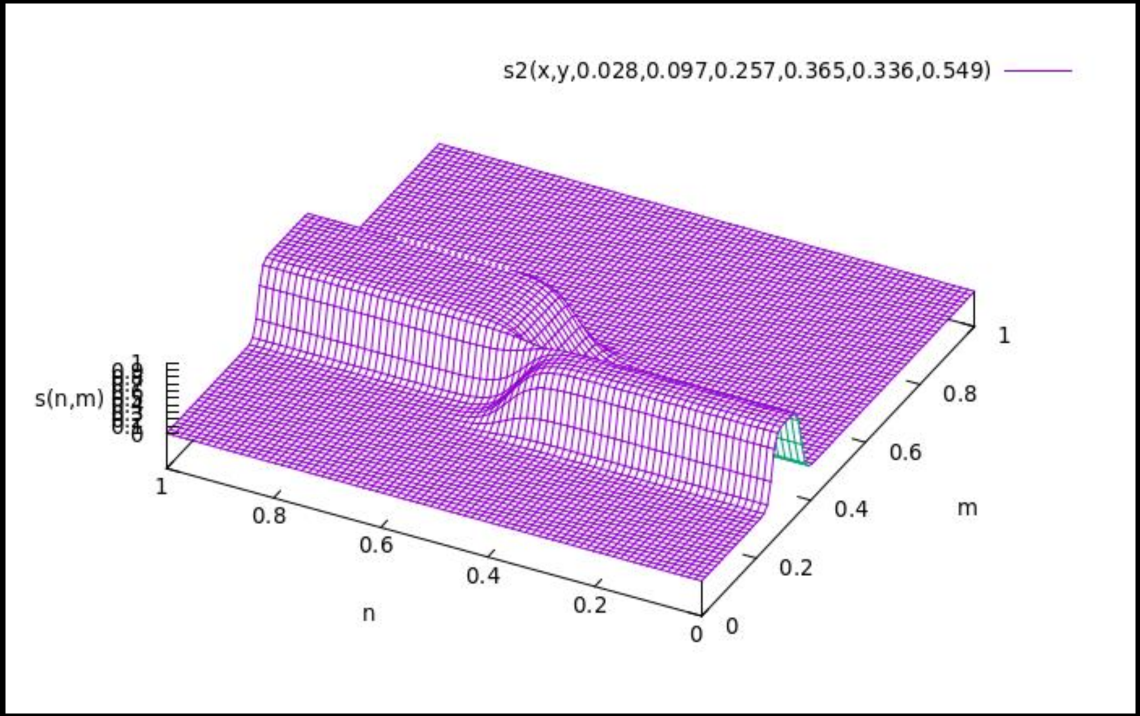
s(n,m)

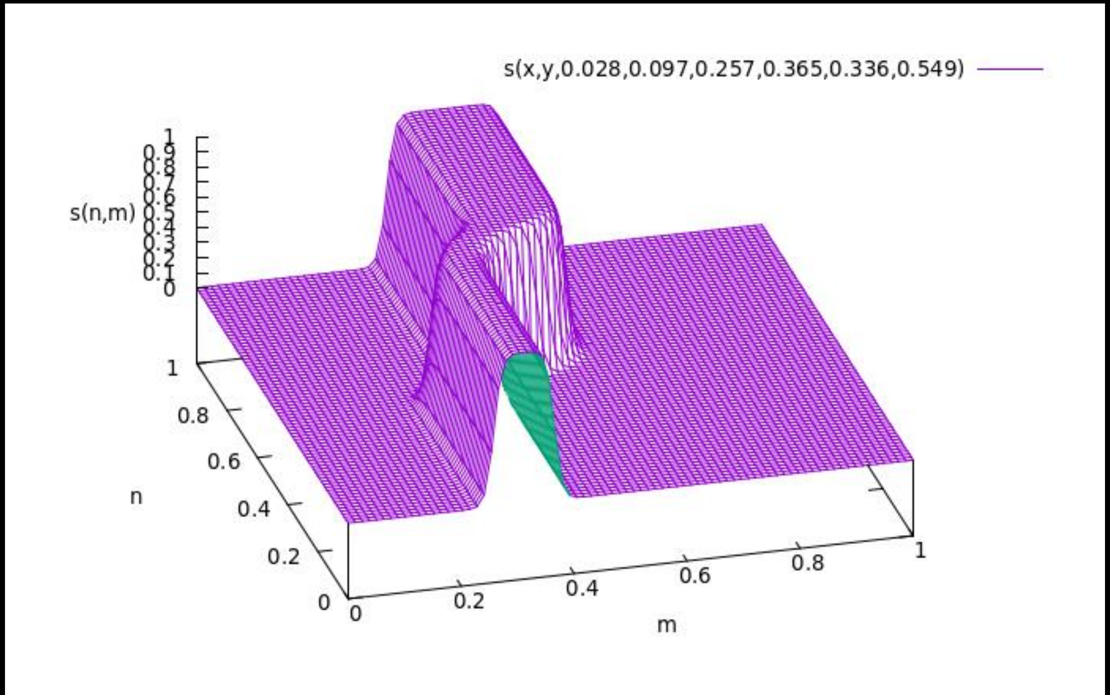


s(n,m,alphan,alpham,b1,b2,d1,d2)=sigma2(n,sigmam(b1,d1,m,alpham), sigmam(b2,d2,m,alpham),alphan)

splot [0:1] [0:1] s(x,y,0.028,0.097,0.257,0.365,0.336,0.549) w pm3d

s2(n,m) - inna postać, inne zachowanie modelu





σ2=σm(σ2(n, b1, b2, αn), σ2(n, d1, d2, αn), m, αm)

s2(n,m,alphan,alpham,b1,b2,d1,d2) = sigmam(sigma2(n,b1,b2,alphan),sigma2(n,d1,d2,alphan), m, alpham)

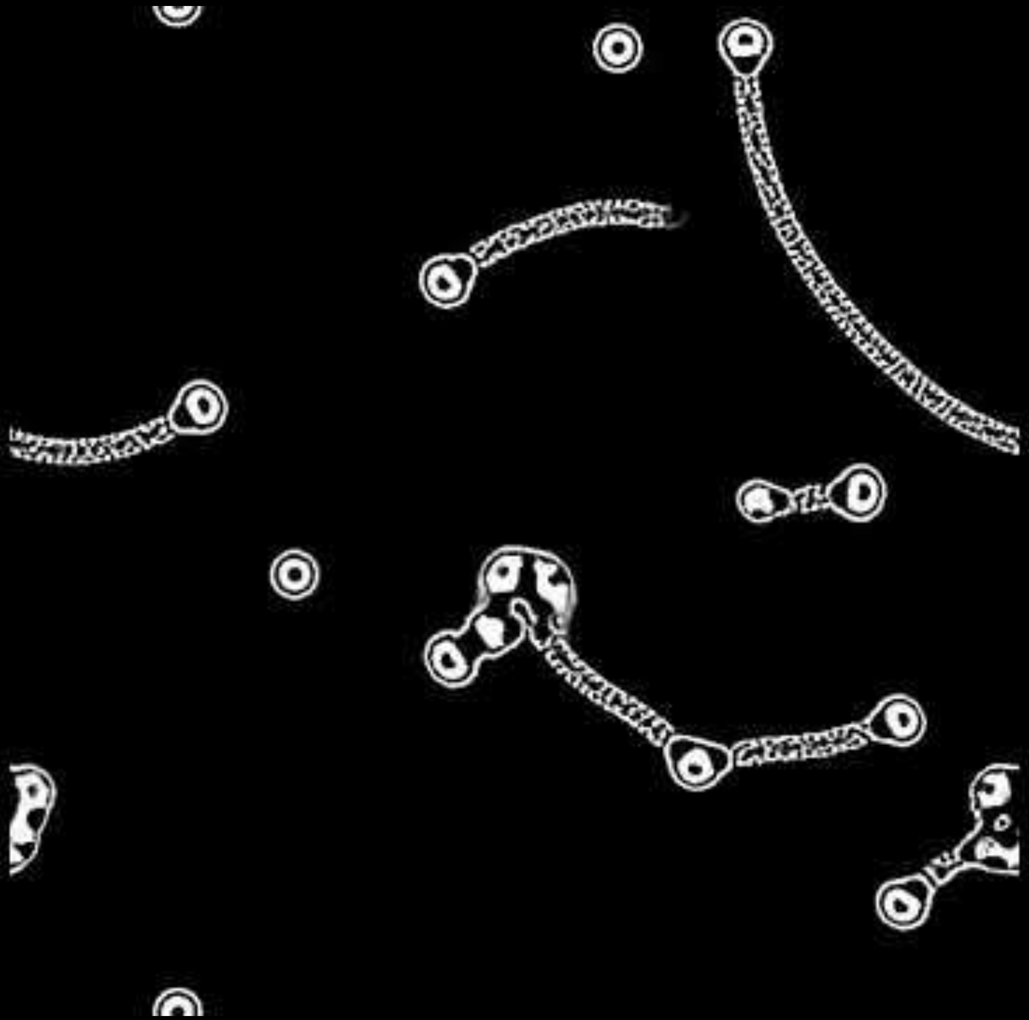
Nowy stan w SmoothLife

$$new = old + dt * (2\, s(n,m) - 1)$$

Pełny algorytm SmoothLife

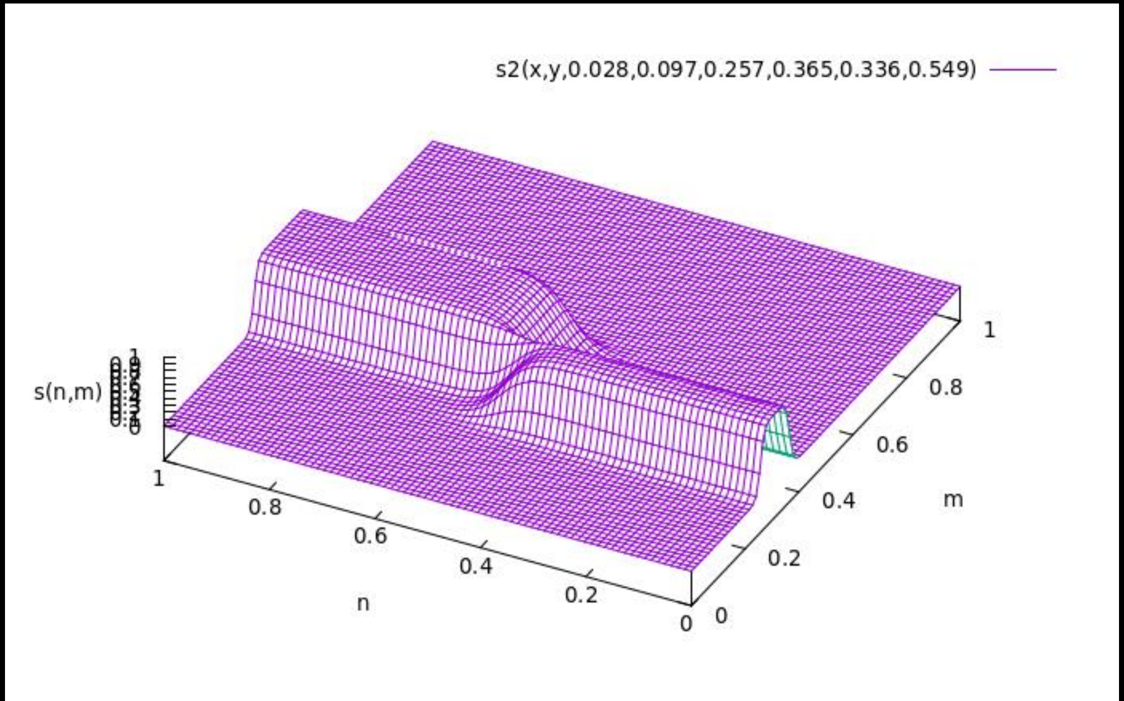
- Inicjalizuj mapę "życia" f(x,y) - wartości 0-1
- Oblicz stopień zajętości sąsiedztwa 1 (m)
- Oblicz stopień zajętości sąsiedztwa 2 (n)
- Oblicz funkcje sigma1/2/m
- Wyznacz nową wartość:
  - **new** = old + dt * (2 s(n,m) - 1)
- Normalizuj "new" do wartości 0-1

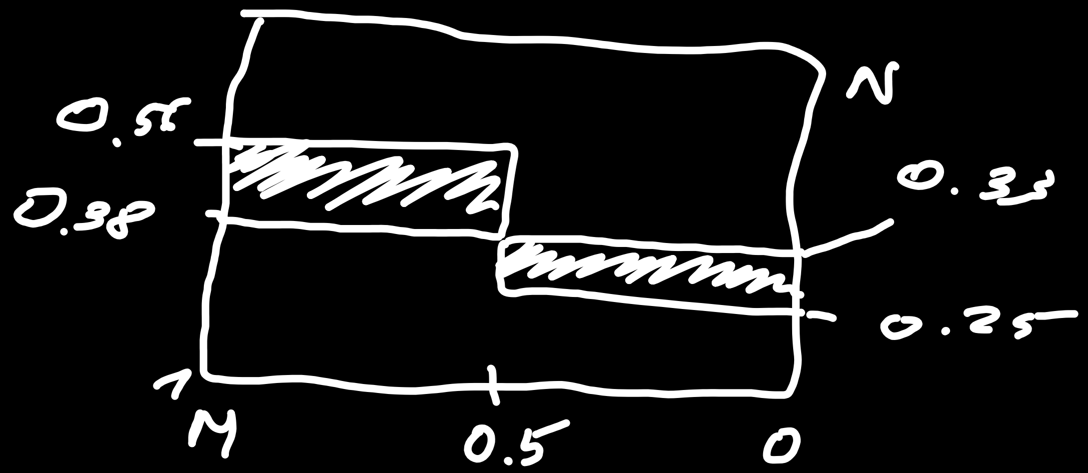
SmoothLife



https://youtu.be/PUsWxzfuUmA

Basic SmoothLife*





https://www.shadertoy.com/view/sdKSzV

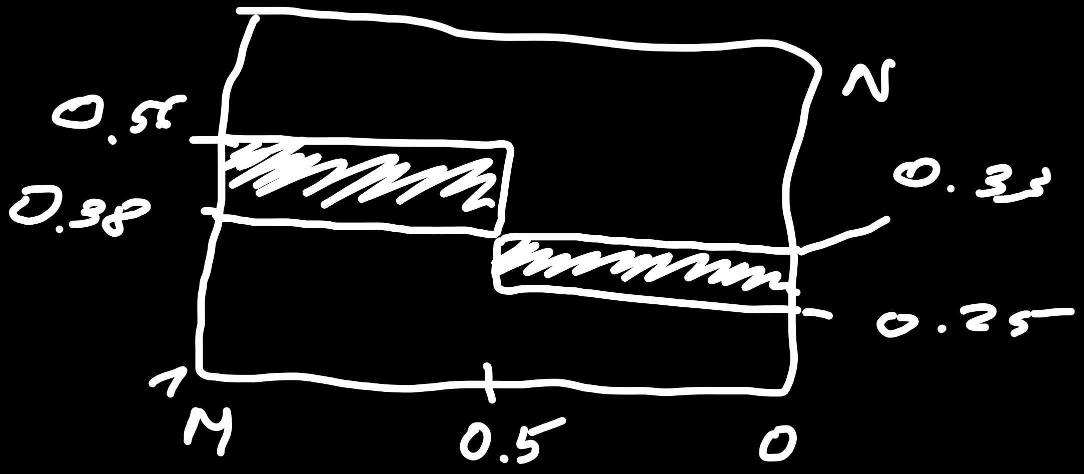
SmoothLifeS* - bariery aktywacji

M<0.5 (martwa)

```
if(N > 0.25  &&  N<0.33)
        new = old + dt;
else
        new = old - dt;
```

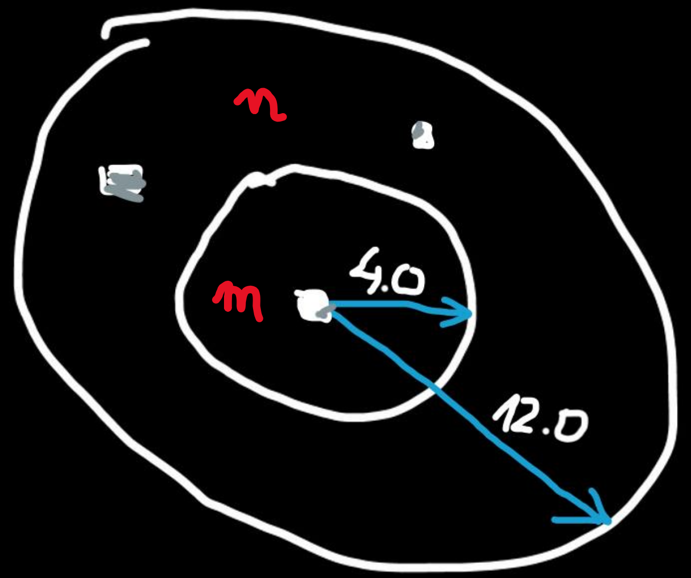
M>0.5 (żywa)

```
if(N >0.38   &&   N<0.56)
        new = old + dt;
 else
        new = old - dt;
```



https://www.shadertoy.com/view/sdKSzV

Simple SmoothLife



- Komórki częściowo żywe/martwe
- Dwa sąsiedztwa
- Niecałkowite bariery dla przeżycia:

Dodanie życia: $(m<0.5, n>0.25, n<0.33)$ lub $(m>0.5, n>0.35, n<0.51)$

Odjęcie życia: $(m<0.5, n>0.25, n<0.33)$ lub $(m<0.5, n>0.25, n<0.33)$

SmoothLifeS - algorytm

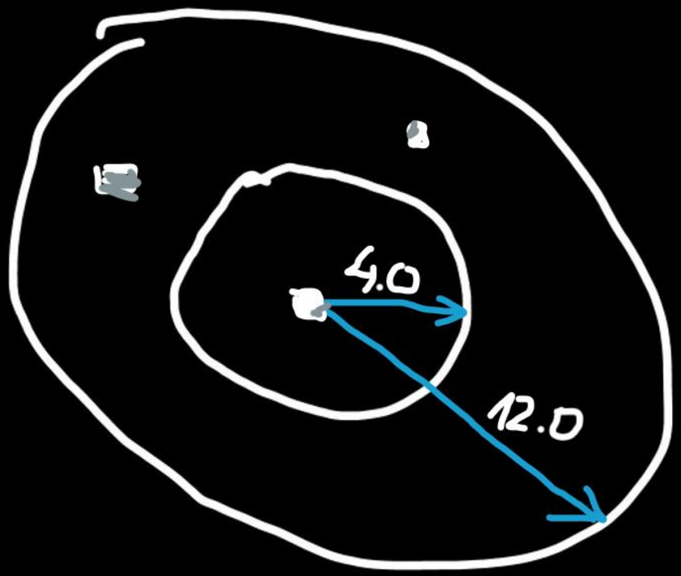
1) Przygotowanie warunku początkowego (wartości od 0 do 1)
2) Obliczenia gęstości żywych komórek w otoczeniu punktu m
3) j.w. w otoczeniu n (torus 2D)
4) Dodanie / odjęcie wartości zgodnie z barierami aktywacji

(kolorowanie wg M*N)

https://youtu.be/6Wh5E6VerGA

Słowo na koniec

- Dlaczego R1/R2=3? Np. R1=4, R2=12?



Smooth Life bounds

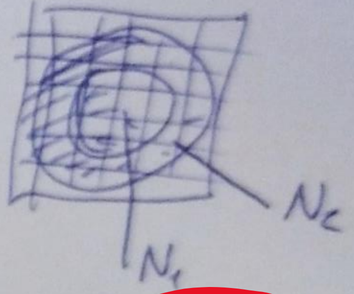
$R_1 = 4$

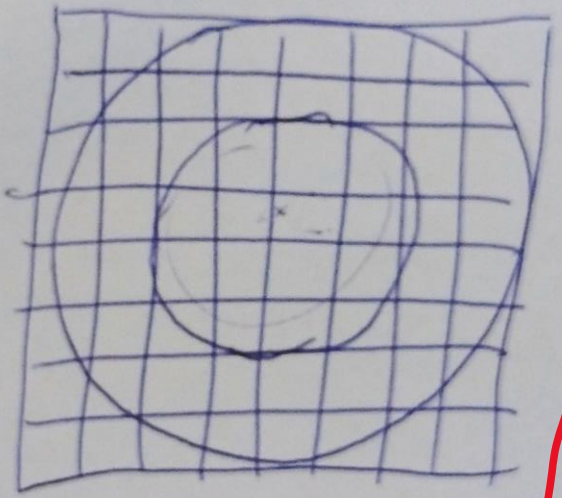
$R_2 = 12$

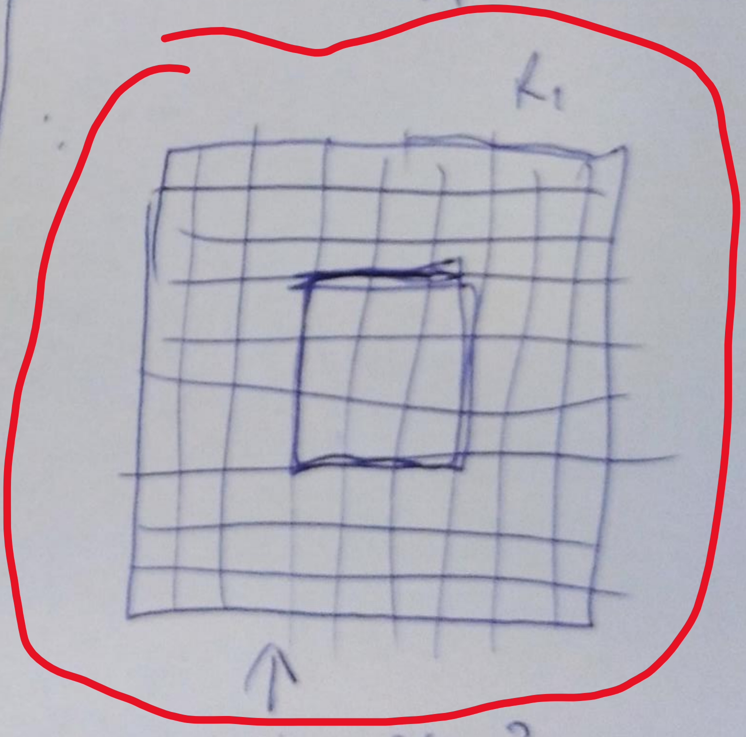
$V_1 = \pi R_1^2 = \pi \cdot$

$V_2 = \pi \cdot ($

$\frac{N_1}{V_1}$, $\rho_2 = \frac{N_2}{V_2}$ $\longrightarrow$ discretized smoothlife

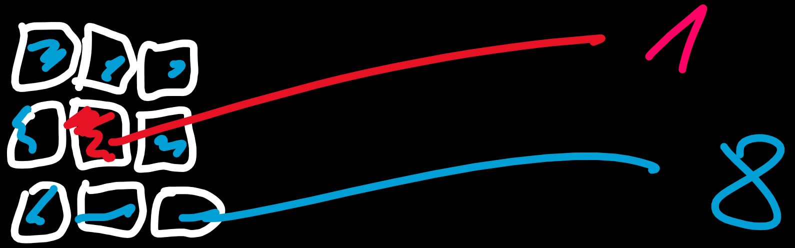






Wynik dla kwadratowego? (wstawić)

N1/N2 Dla gry w życie



$$\frac{N_1}{N_2} = \frac{1}{8}$$

bounds

R_1 / R_2

$$V_1 = \pi R_1^2 = \pi \cdot 16 = 50{,}26$$

$$V_2 = \pi \cdot (\cancel{}\, 144 - 16) = 128\pi = 402{,}12$$

$V_2 = \pi \cdot (144 - 16) = 128$

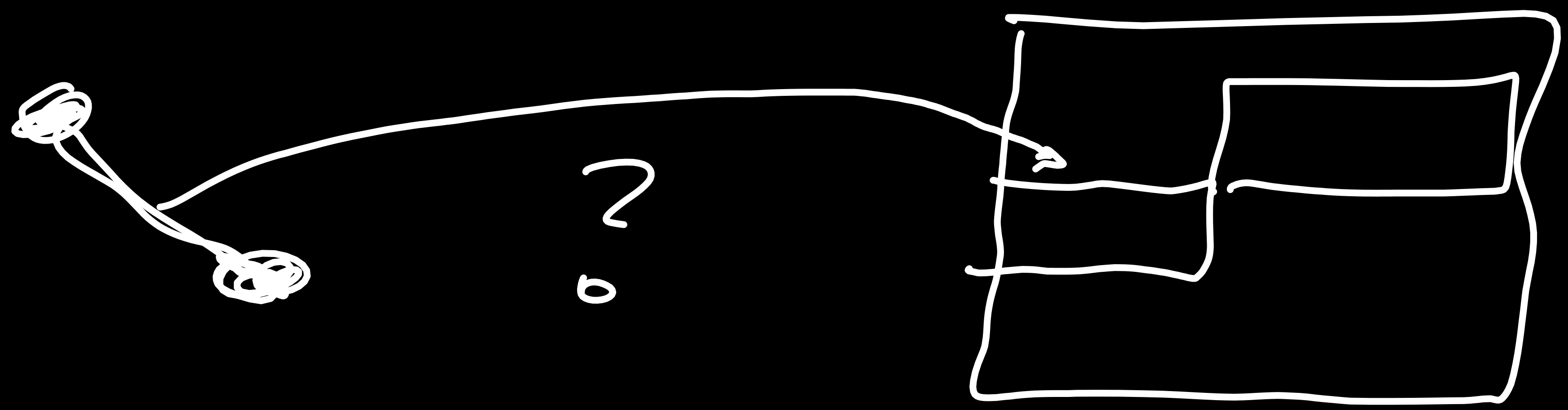
$$\boxed{V_2 / V_1 = 8}$$

number of smo

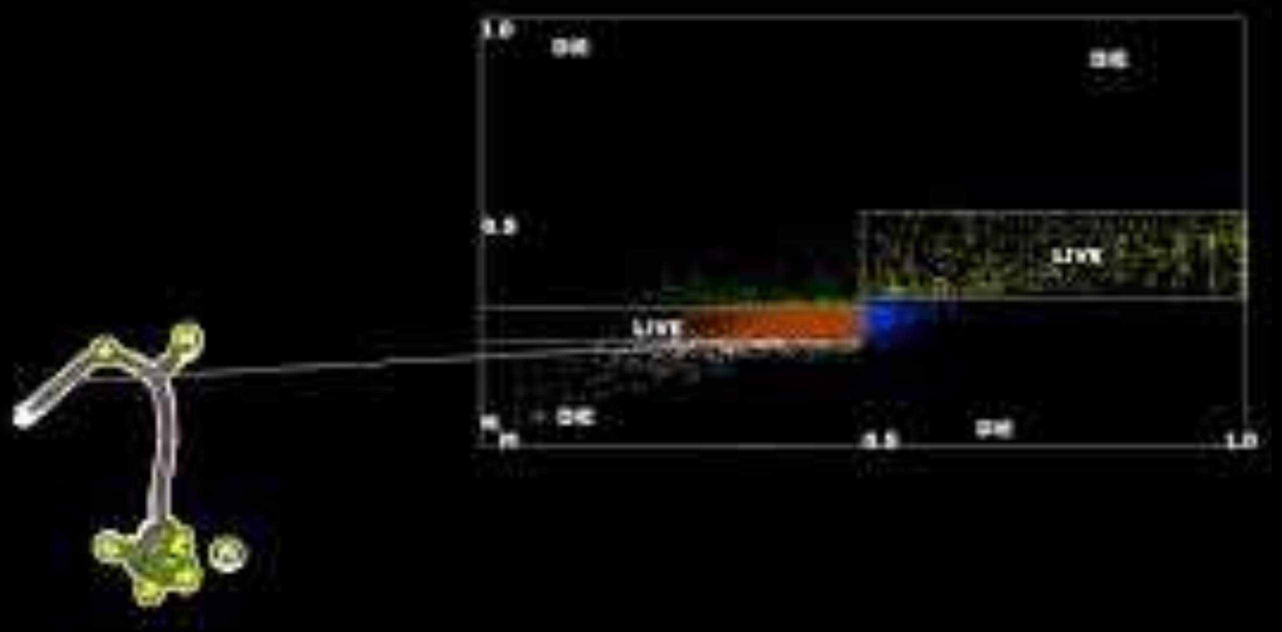
DLA $R1/R2 = 1/3$

Analiza modelu SmoothLife

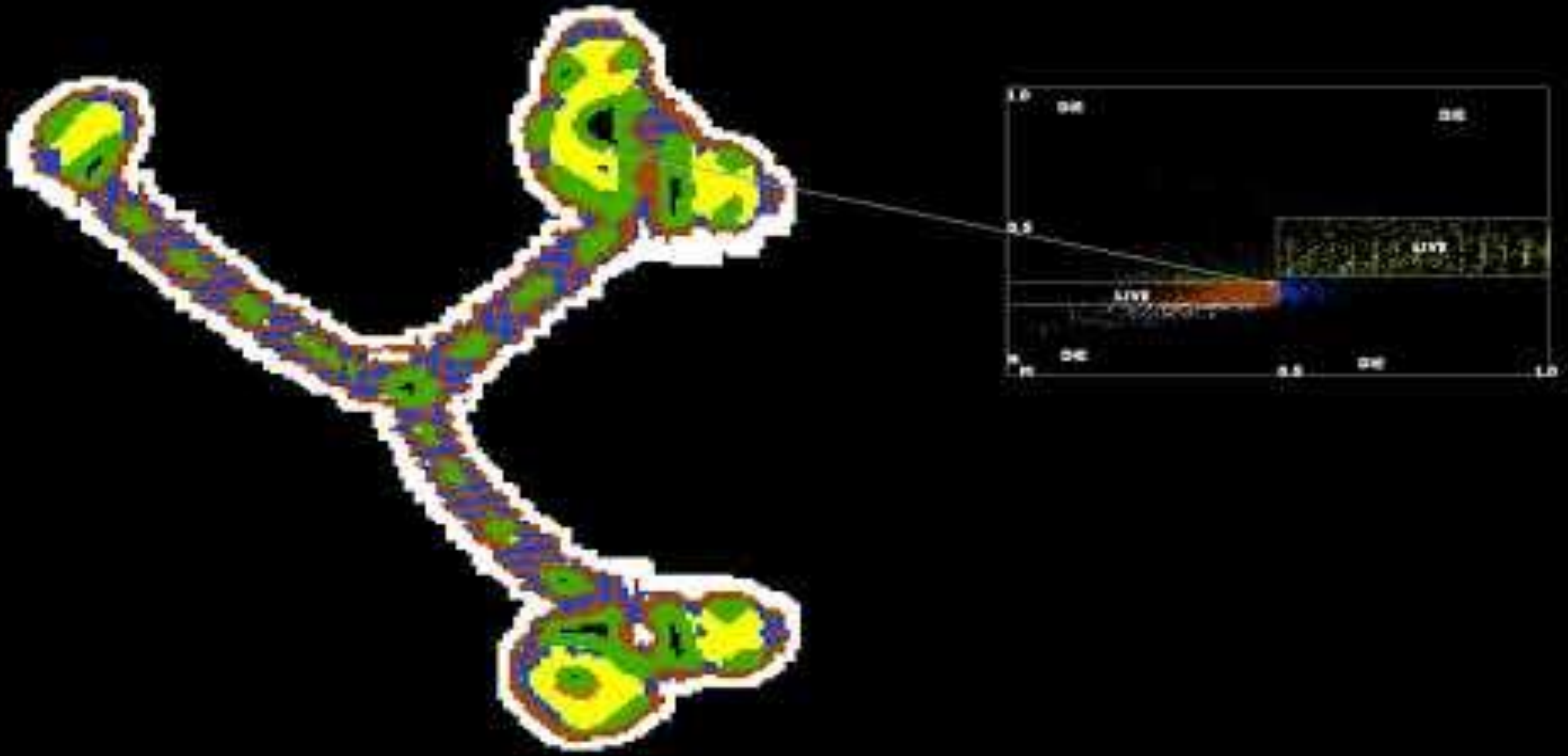
- Rozkład komórek w przestrzeni
- Klastrowanie faz
- Obiekty zanurzone w środowisku (otoczenie, które je chroni)
- Jaka jest dynamika tych zmian?



"Diagram
fazowy"
N vs M



"Diagram fazowy" N vs M



Koniec