Multiscale Modelling

Simple grain growth CA and Monte Carlo

Daniel Skimina

List of content:

1. Introduction

Main idea of Cellular Automata method is to divide part of the investigated material into 1,2,3 dimentional grid. Each cell of that grid is called Cellular Automaton. All of the cellular automatons are known as Cellular Automata Space. Cellular Automata works in discrete time steps. After each time step the state of each cells in lattice is updated synchronically. CA algorithm base on states of cells from previous step and theirs neighbours.

2. My application

2.1 Base application

2.1.1 GUI Description

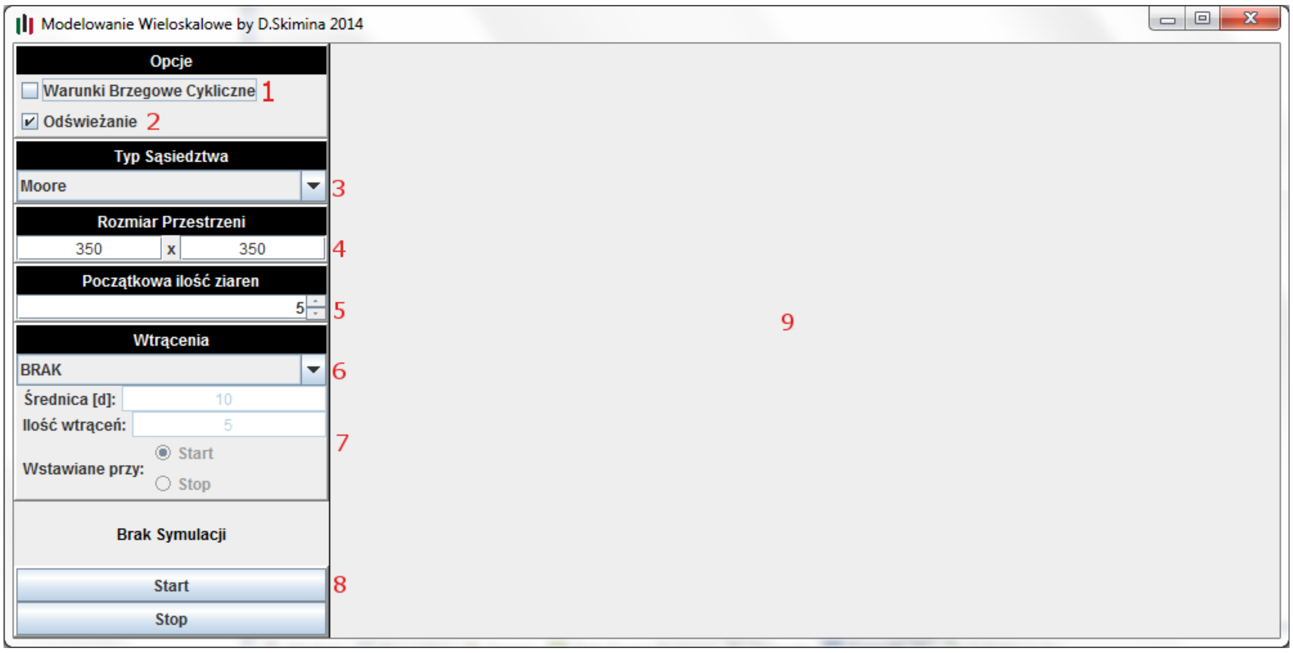
My application to simulating grain growth using CA method is written in JAVA. Additional fetures used in building application:

- SWING (to build GUI)
- Maven (to manage dependencies and builds control)
- GIT (to control versioning)

The source code of the project can be found on GitHub:

https://github.com/danielsky/ModelowanieWieloskalowe

The main idea of my program was to wrote the core of application and graphical engine. The next modifications use the core application but bring too much changes to include them into core. So every next modification is written ass separate program which uses the same base mechanisms.

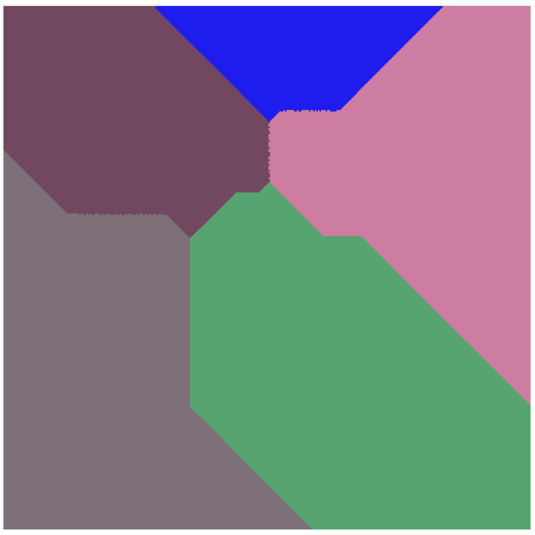


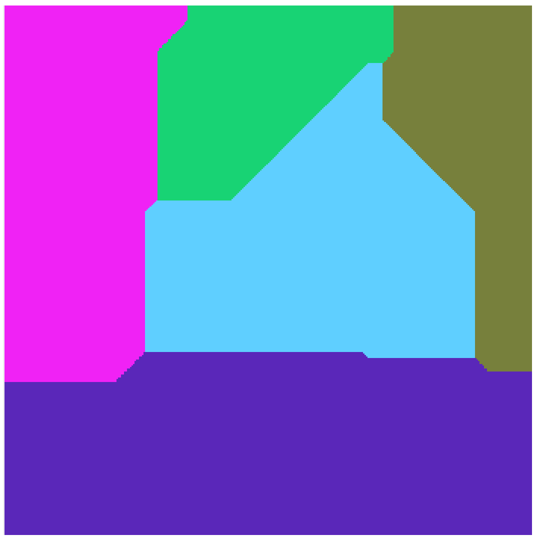
1. Boundary conditions
   1. Checked – Periodic boundary conditions
   2. Unchecked – Absorbing boundary conditions
2. Refreshing – After each time step visualisation in workspace is refreshed

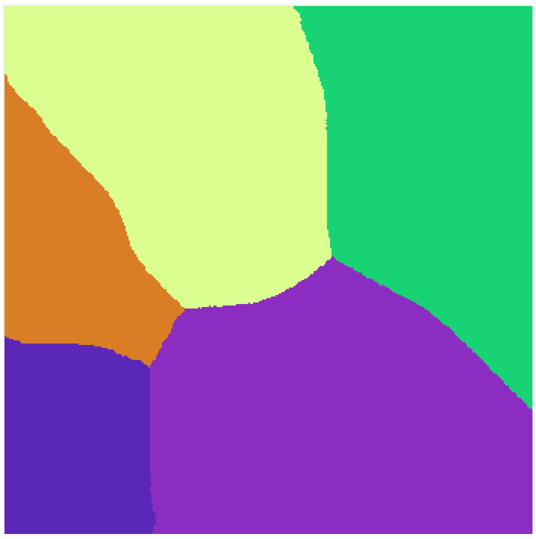
3. Neighbourhood type. One of the following four:
   1. Moore
   2. von Neumann
   3. Randomly pentagonal
   4. Randomly hexagonal
4. Size of cellular automata space
5. Initial number of grains
6. Inclusions type
   1. BRAK – no Inclusions
   2. KWADRAT – inclusions as squares
   3. KOLO – inclusions as circles
7. Options for incusions (active only when one of KWADRAT or KOLO is selected):
   1. Diameter of inclusion
   2. Number of inclusions
   3. Time of inserting inclusions (Beginnig/End)
8. Button for Start simulation
9. Workspace for visualisation

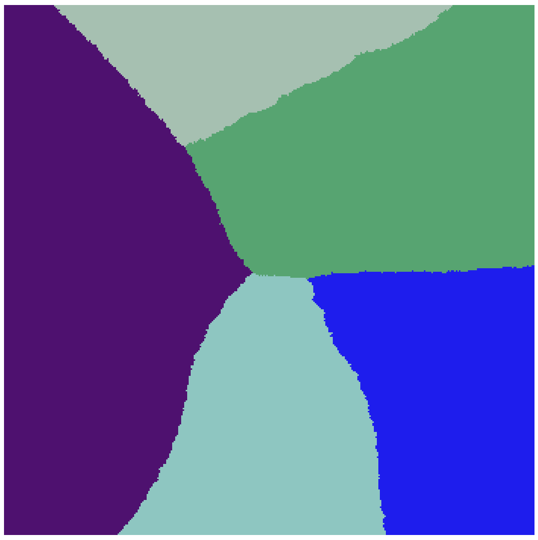
2.1.2 Sample generated structures:

2.1.2.1 Common parameters: Absorbing boundary conditions, 5 grains at start

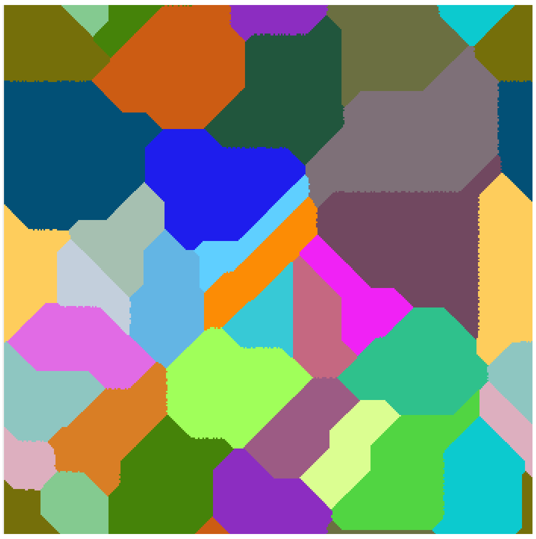

1: Moore

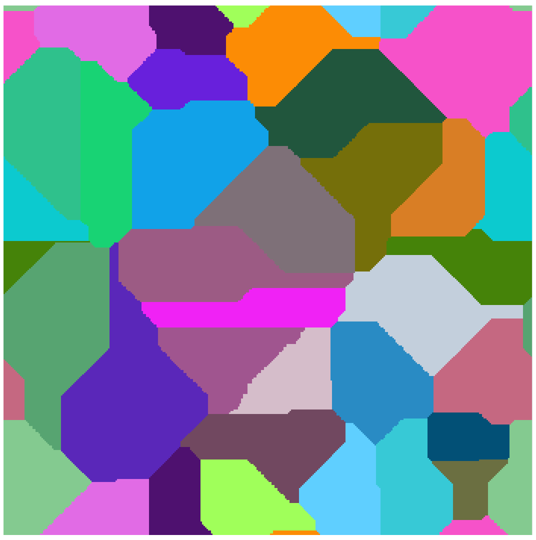

2: von Neumann

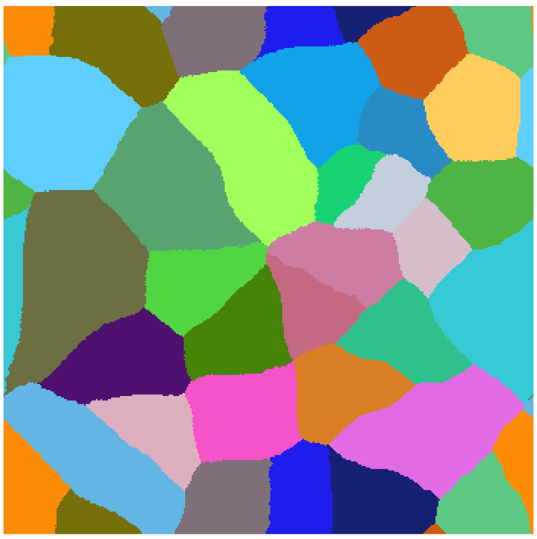

3: Hexagonal

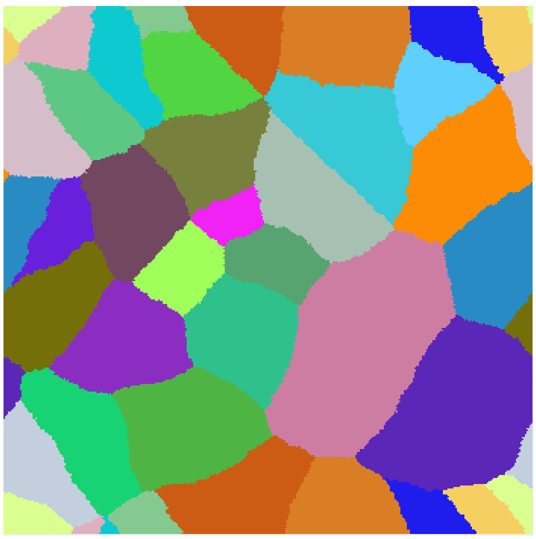

4: Pentagonal

2.1.2.2 Common parameters: Periodic boundary conditions, 30 grains at start

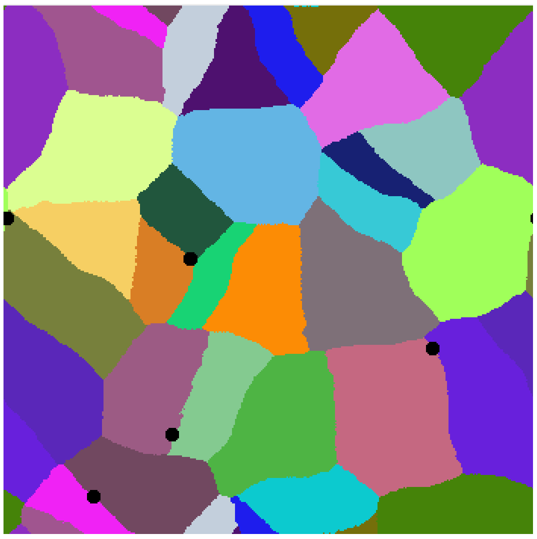

Moore

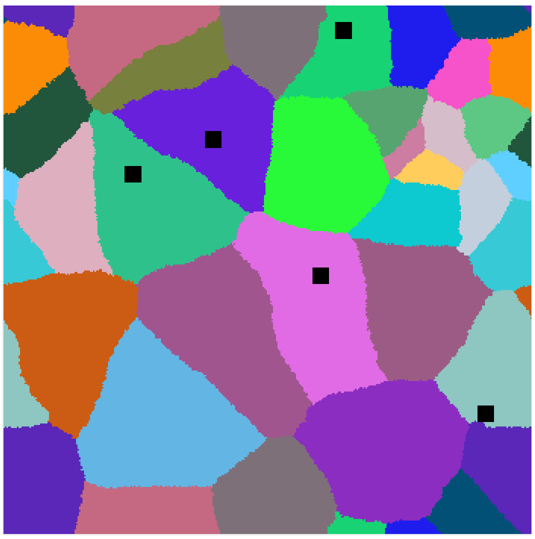

von Neumann


Hexagonal


Pentagonal

2.1.2.3 Structures with inclusion. Periodic boundary conditions, initial number of grains – 30


Hexagonal, Circle inclusions, Insert at stop

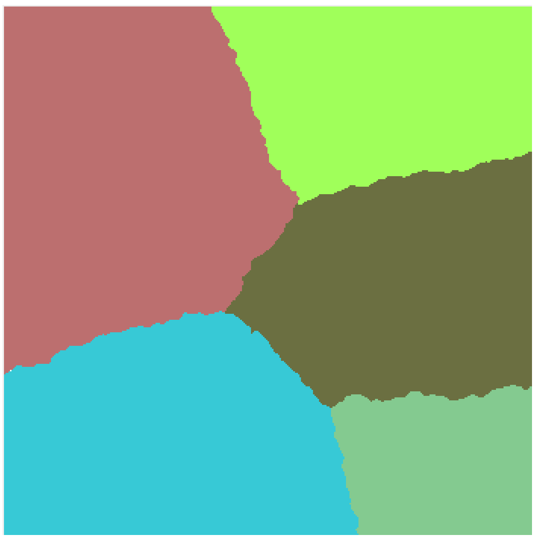

Pentagonal, Square inclusions, Insert at start

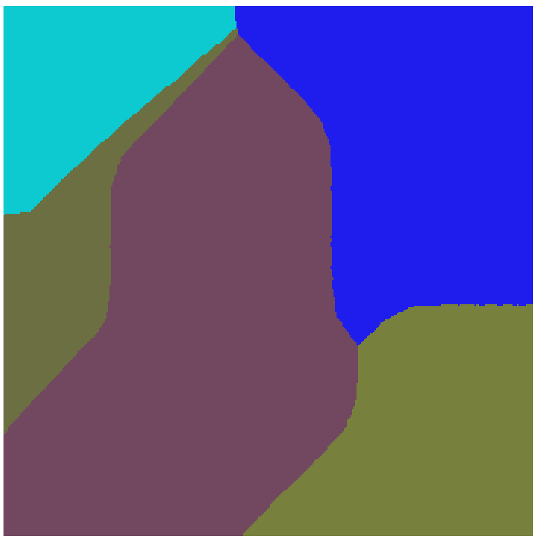
2.2 „4 rule" modification

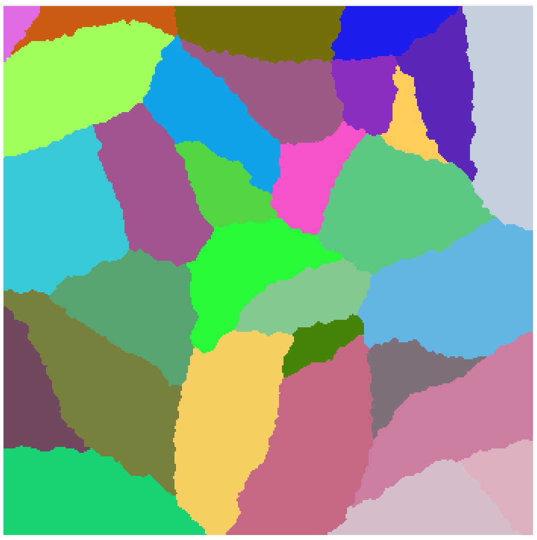
The four rule modification is the special transition rule. The cell to change it's state must meet one of the following conditions:

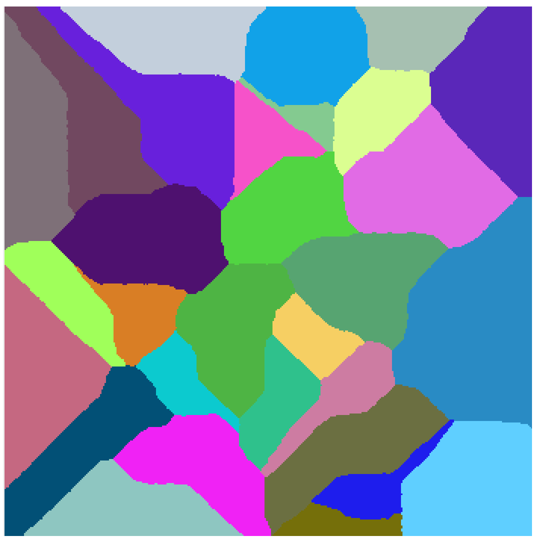
- Moore neighbourhood, minimum number of neighbors to change = 5
- von Neumann neighbourhood, minimum number of neighbors to change = 3
- Further Moore neighbourhood, minimum number of neighbors to change = 3
- Moore neighbourhood, X% probability to change

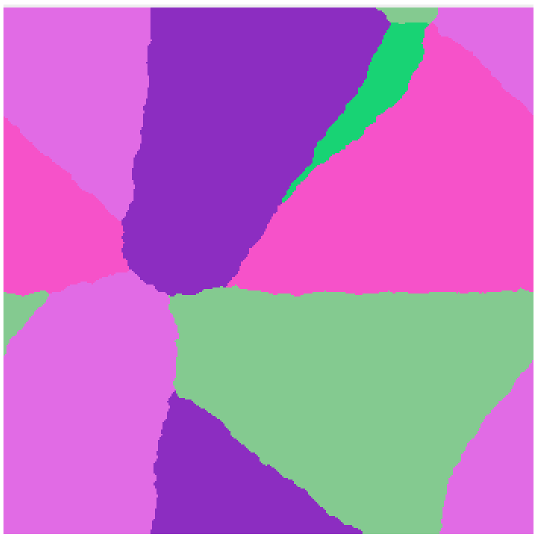
2.2.1 Sample generated structures:

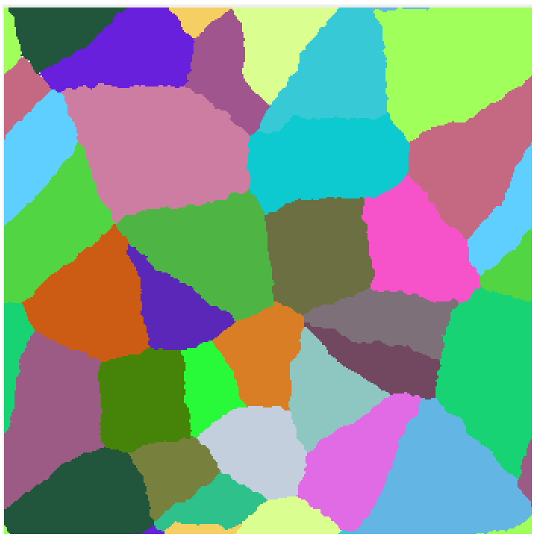

5 initial grains, 30% probability, absorbing boundary conditions


5 initial grains, 80% probability, absorbing boundary conditions


30 initial grains, 30% probability, absorbing boundary conditions

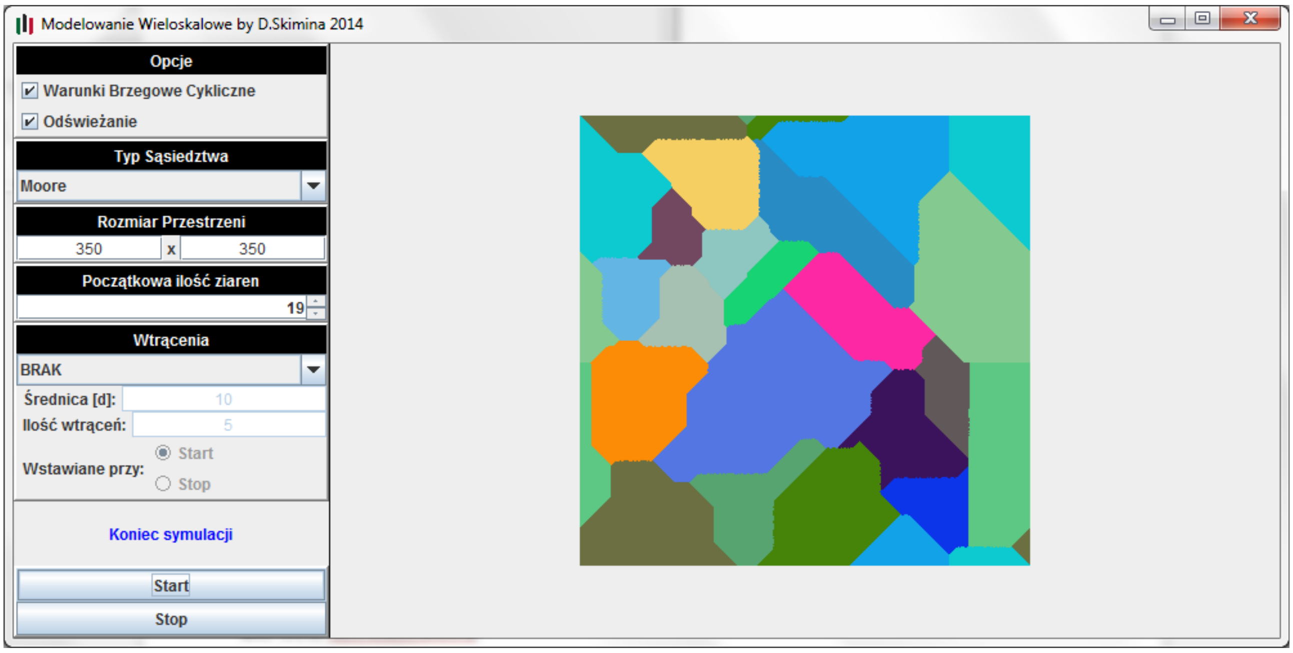

30 initial grains, 80% probability, absorbing boundary conditions


5 initial grains, 50% probability, periodic boundary conditions


30 initial grains, 50% probability, periodic boundary conditions

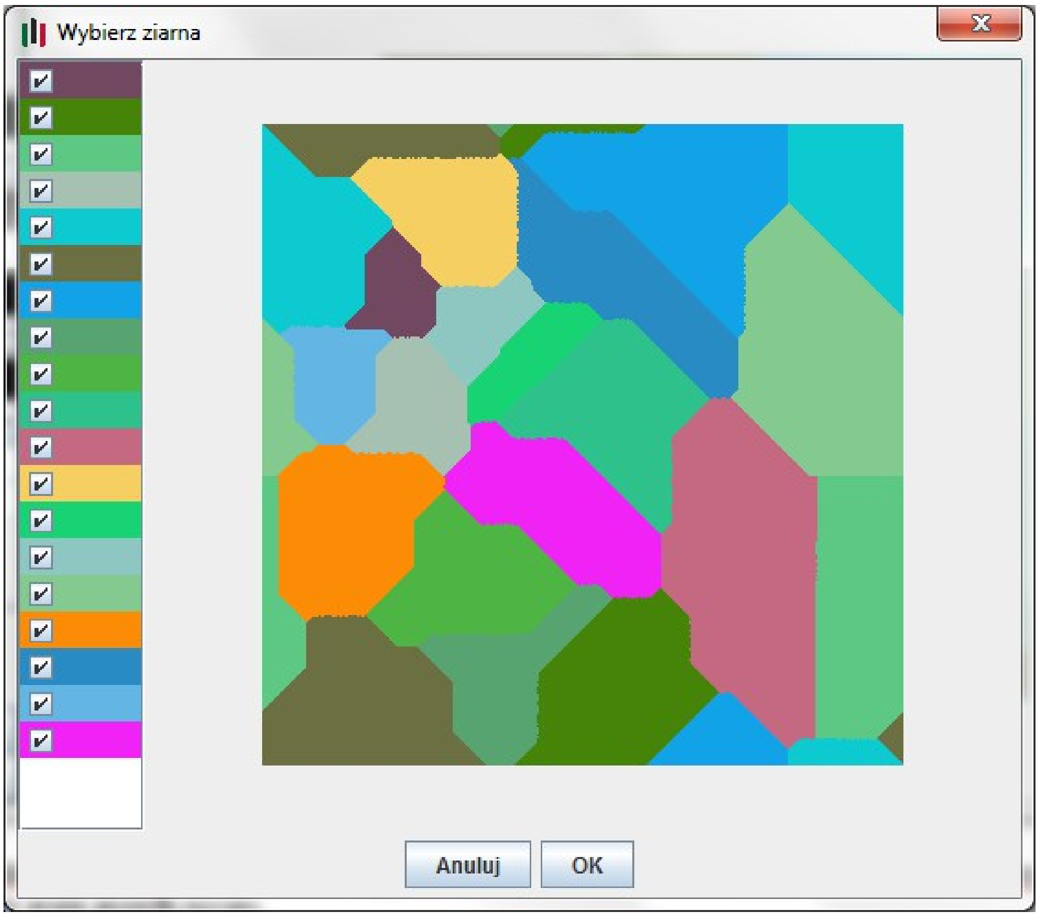
2.3 2nd modification

Next modification was to select some of grains after growing, and next delete them. In newly created empty space next grain growth occurs.



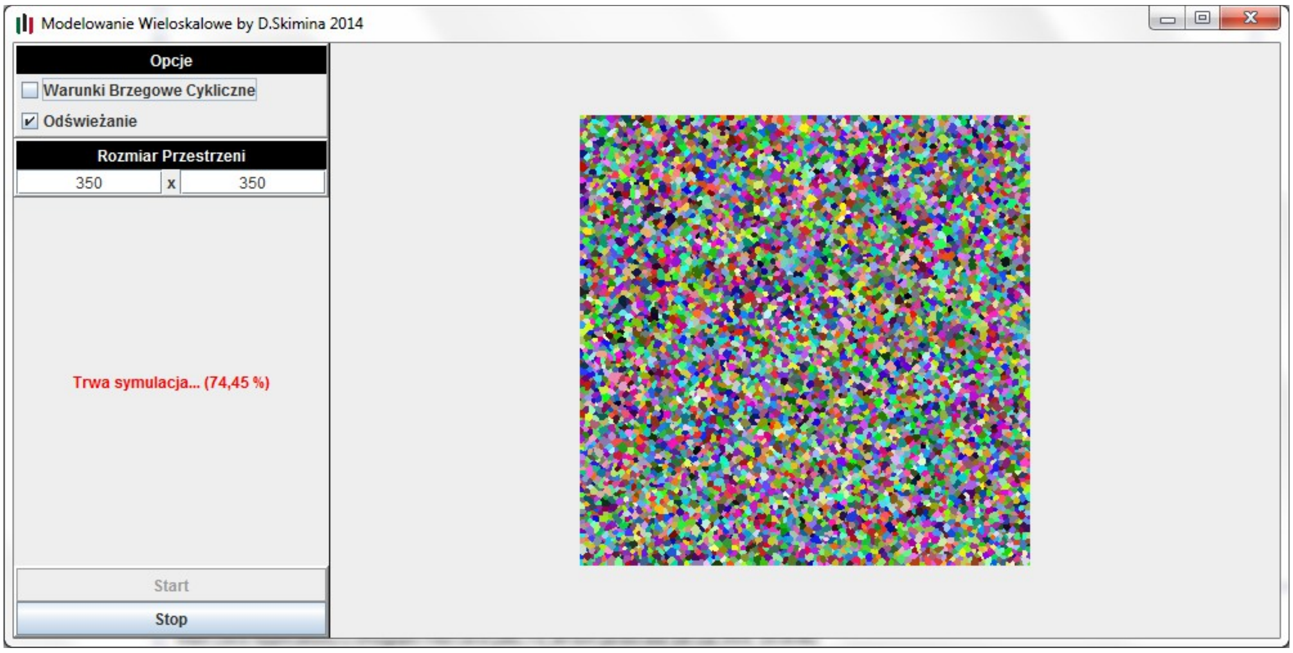
[✓] Warunki Brzegowe Cykliczne
[✓] Odświeżanie
Typ Sąsiedztwa: Moore
Rozmiar Przestrzeni: 350 x 350
Początkowa ilość ziaren: 19
Wtrącenia: BRAK
Średnica [d]: 10
Ilość wtrąceń: 5
Wstawiane przy: [✓ Start] Stop
Koniec symulacji

Dialog for choosing grains to delete:



2.4 Monte Carlo

This group of alghoritms is mainly designed for problems that are difficult to solve with deterministic alghoritms. Main idea of Monte Carlo approach is to randomly select one of the cell from space, temporary change their energy, and if temporary energy is less than current, then temporary state is saved. MC finish when all cells from space has chanse to change their energy.


Modelowanie Wieloskalowe by D.Skimina 2014

Opcje

[ ] Warunki Brzegowe Cykliczne

[✓] Odświeżanie

Rozmiar Przestrzeni

350 x 350

Trwa symulacja... (74,45 %)