AGH University of Science and Technology

Faculty of Metals Engineering and Industrial Computer Science

Department of Applied Computer Science and Modelling

Monte Carlo
grain growth algorithm

Przemysław Wal

269029

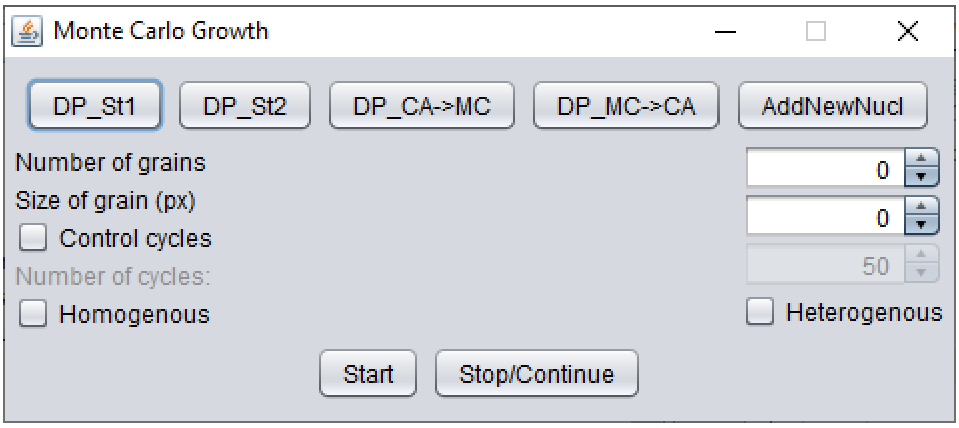
Multiscale modelling

Kraków 2019

Table of Contents

1. Monte Carlo grain growth algorithm

GUI has some specific functions responsible for different features (**Picture 0**)
Neigborhood: Moore (not visible, by default)
Number of Grains: possibility to choose how many grains we need to simulate
Size of Grain: possibility to choose how many pixels each grain has
Control cycles: possibility to choose how many cycles the simulation will take
DP_St1, DP_St2 – buttons responsible for substructures MC-MC
DP_CA->MC - buttons responsible for dual phase CA-MC
DP_MC->CA - buttons responsible for dual phase MC-CA
AddNewNucl - buttons responsible for showing energy visualization/microstructure view
Homogenous/Heterogenous - buttons responsible for energy distribution



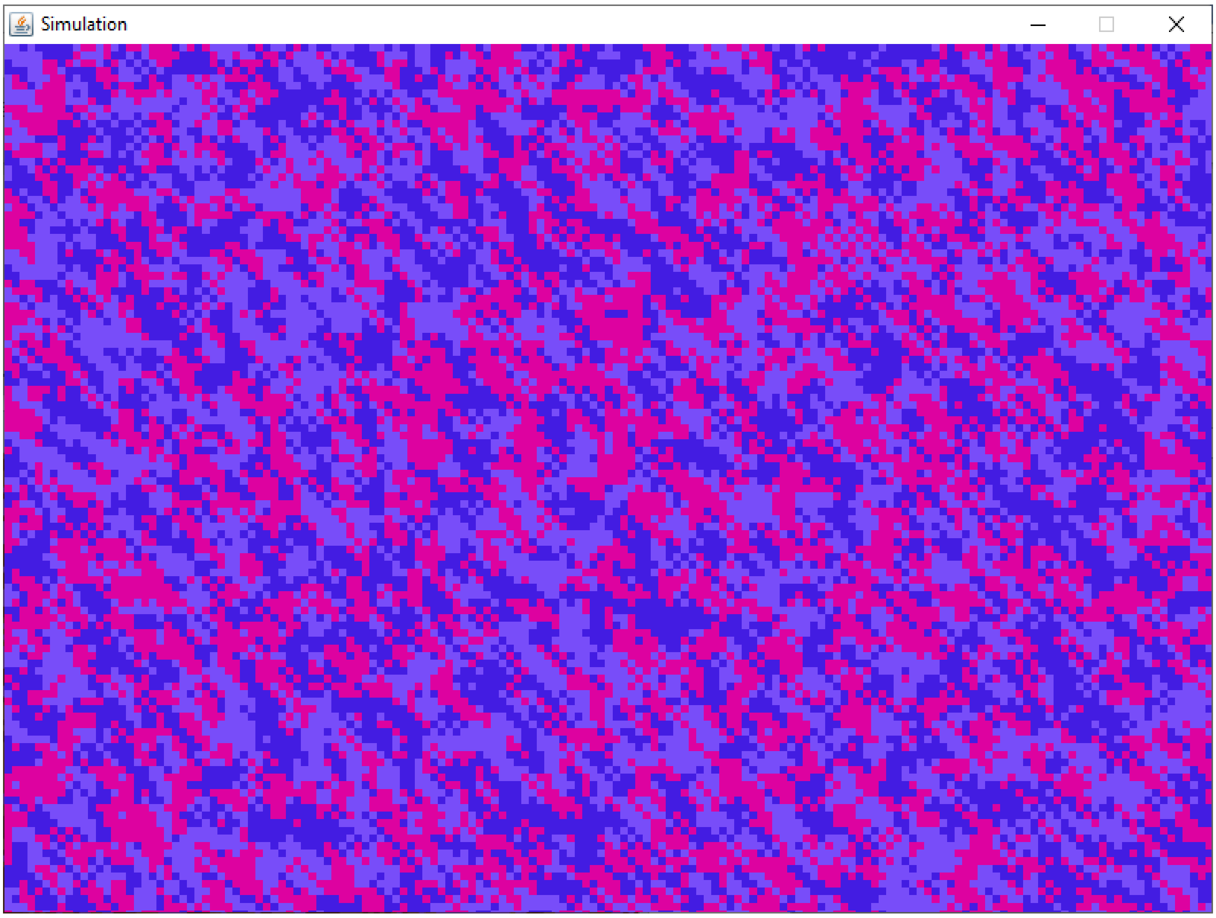
Picture 0 – Control Panel

Whole project can be find on GitHub Project Repository: https://github.com/procesor777

1.1 Example of Monte Carlo grain growth

Number of grains: 3
Size of grains (px): 5
Number of cycles: 50

Example below (**Picture 1.).**

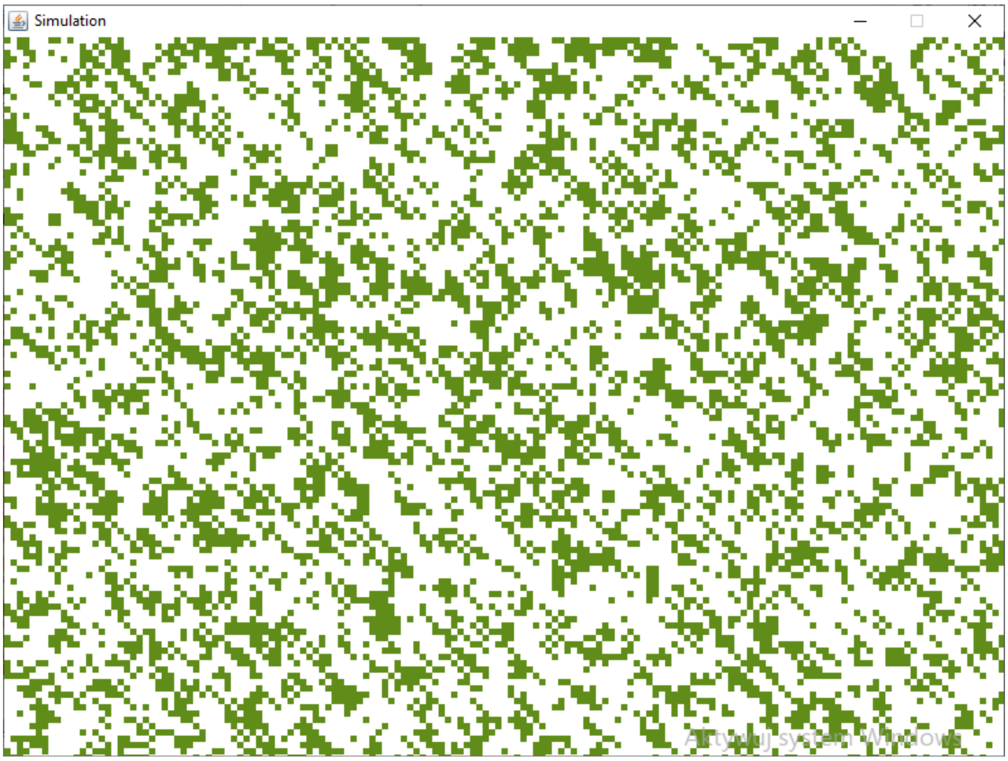


Picture 1 Example of Monte Carlo grain growth

2. Modification of MC grain growth algorithm - substructures CA, MC

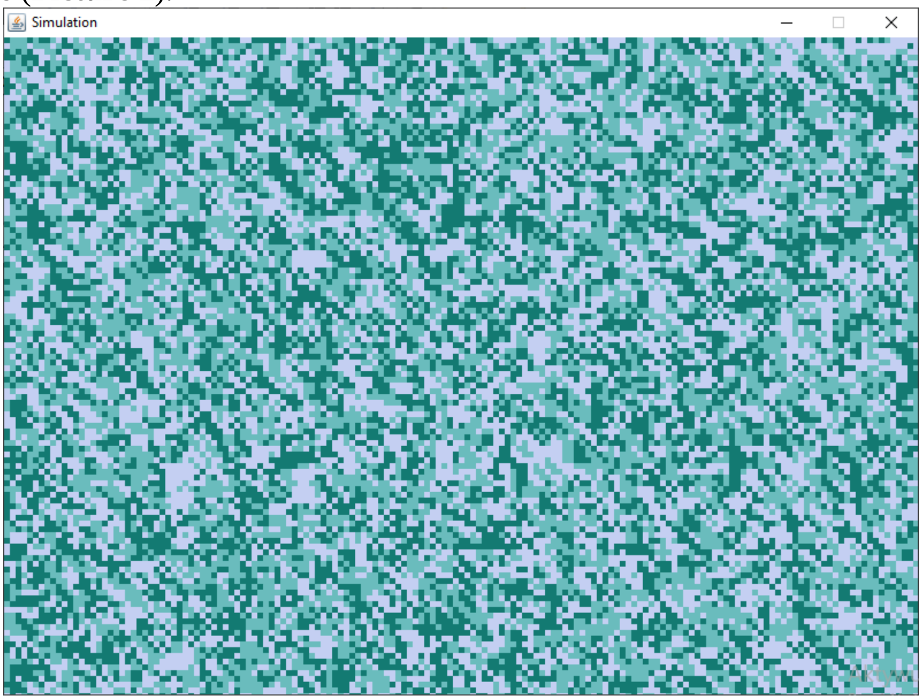
2.1 MC -> MC

On the picture below (***Picture 2***) we can see the randomly choosen cells which will be used as the substructure in next simulations.



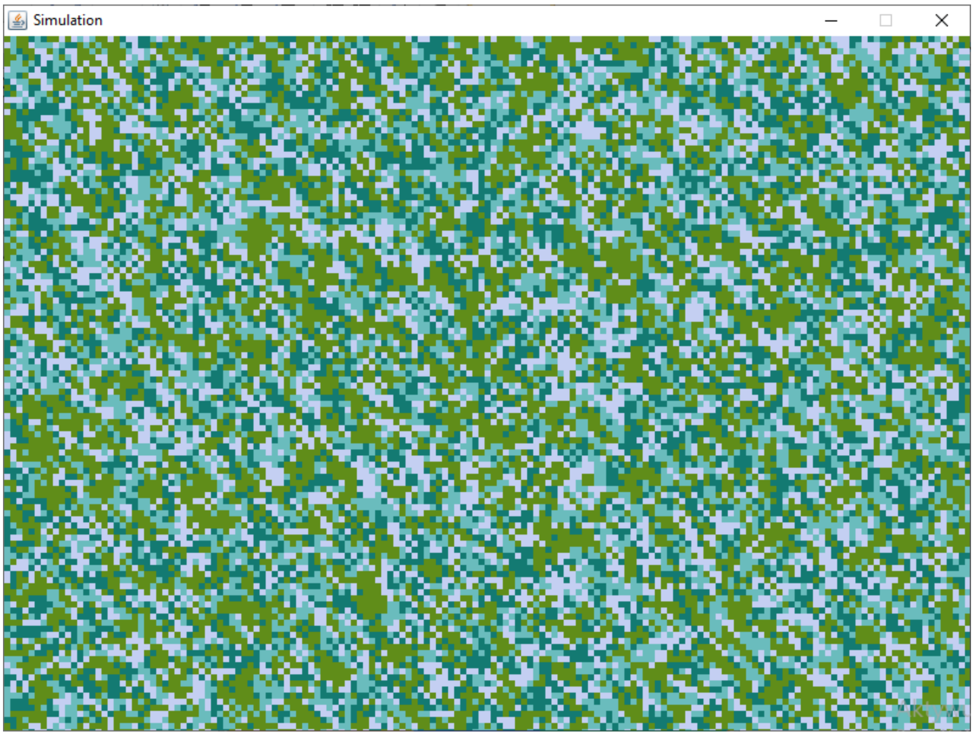
Picture 2 Example of randomly choosen one type of grains from Monte Carlo simulation

Next simulation of Monte Carlo grain growth (**Picture 3**), on which we will add the substructure (**Picture 2**).



Picture 3 Example of another Monte Carlo simulation

And after the adding the substructure, we can see the result on the picture below (**Picture 4**).

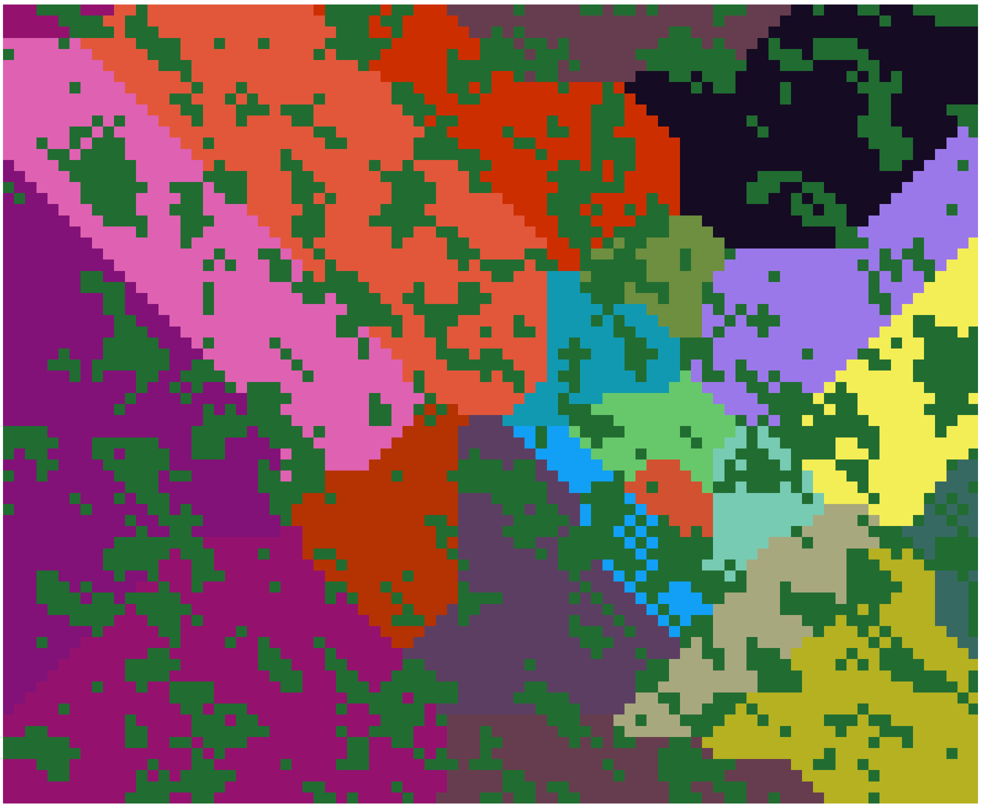


Picture 4 Example of added substructure to MC simulation

2.2 MC->CA

We will use the substructure from the **Picture 2**. And on the **Picture 5** we can see the randomly generated Cellular Automata Grain Growth, on which we will add substructure (**Picture 6**).

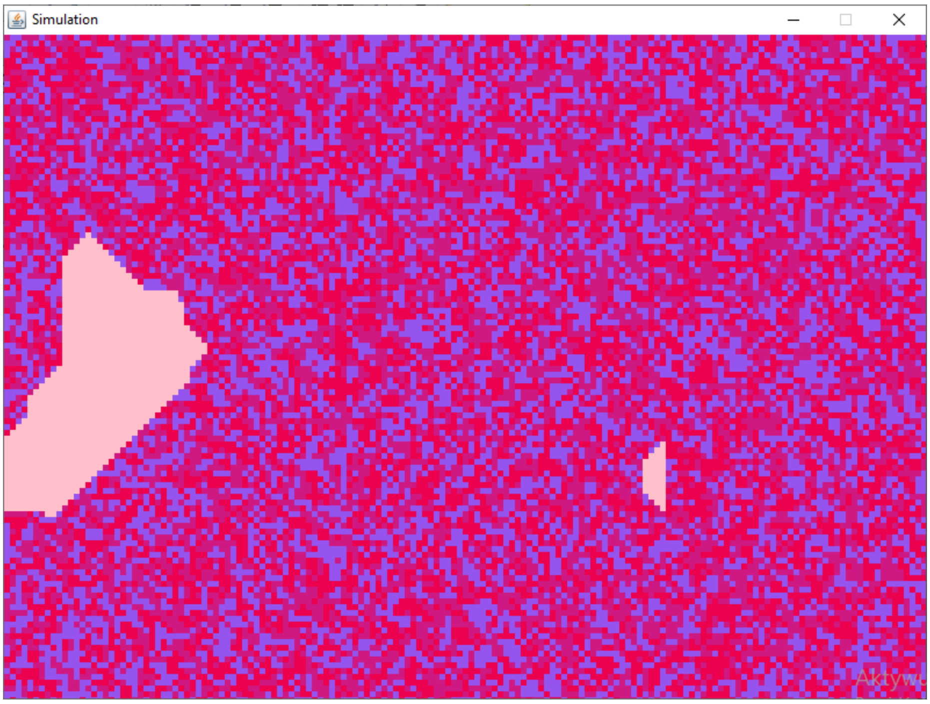
Picture 5 Example of Cellular Automata grain growth


Picture 6 Example of added substructure to Cellular Automata Grain Growth

2.3 CA->MC

On the picture below (**Picture 7**) we can see randomly choosen grain, which will be later added as substructure to the Monte Carlo simulation (**Picture 8**).

Picture 7 Example of substructure from Cellular Automata



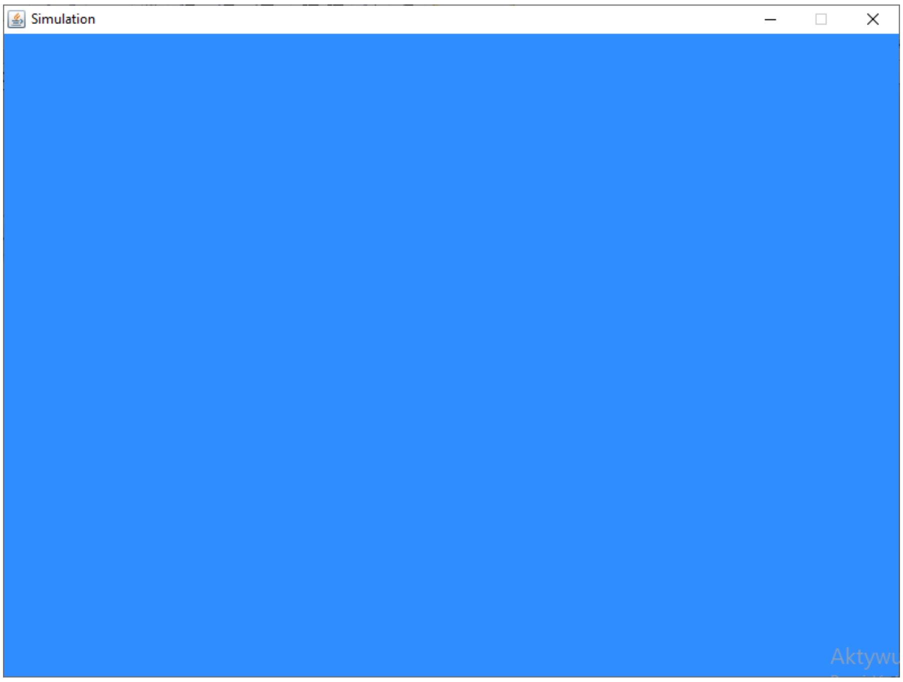
Picture 8 Example of added substructure from Cellular Automata to Monte Carlo.

3. SRX MC: energy distribution, nucleation, growth

3.1 MC static recrystallization algorithm - energy distribution

3.1.1 Homogenous

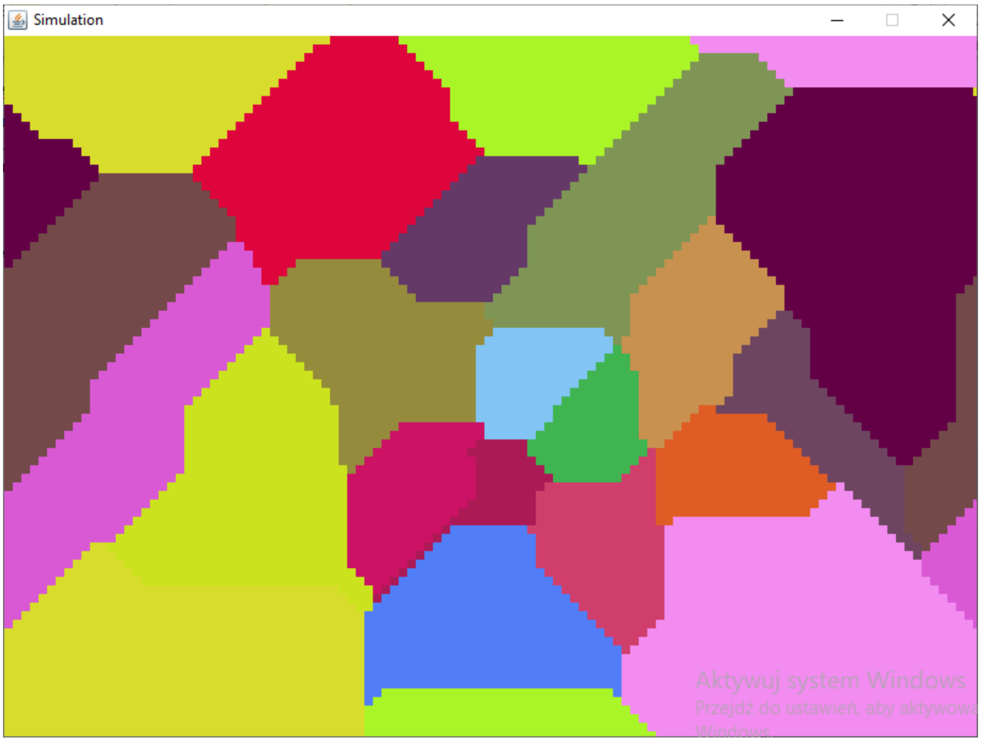
On the picture below (**Picture 9**), we can see the example of homogonous energy distribution. Each cell have also set the energy H to H=5.



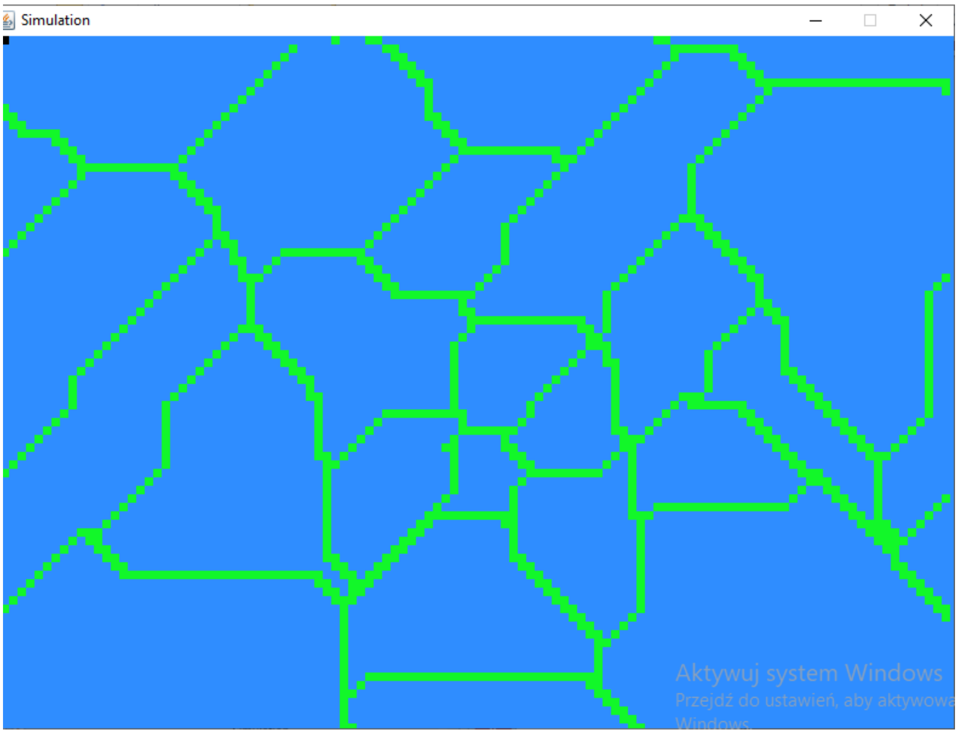
Picture 9 Example of homogenous energy distribution.

3.1.2 Heterogenous

On the **Picture 11**, we can see the example of heterogenous energy distribution.
Below (**Picture 10**) we can see the Cellular Automata simulation.
Each cell on the boundary have also set the energy H to H=5 and each cell in the middle between the boundaries have the energy set to H=2;



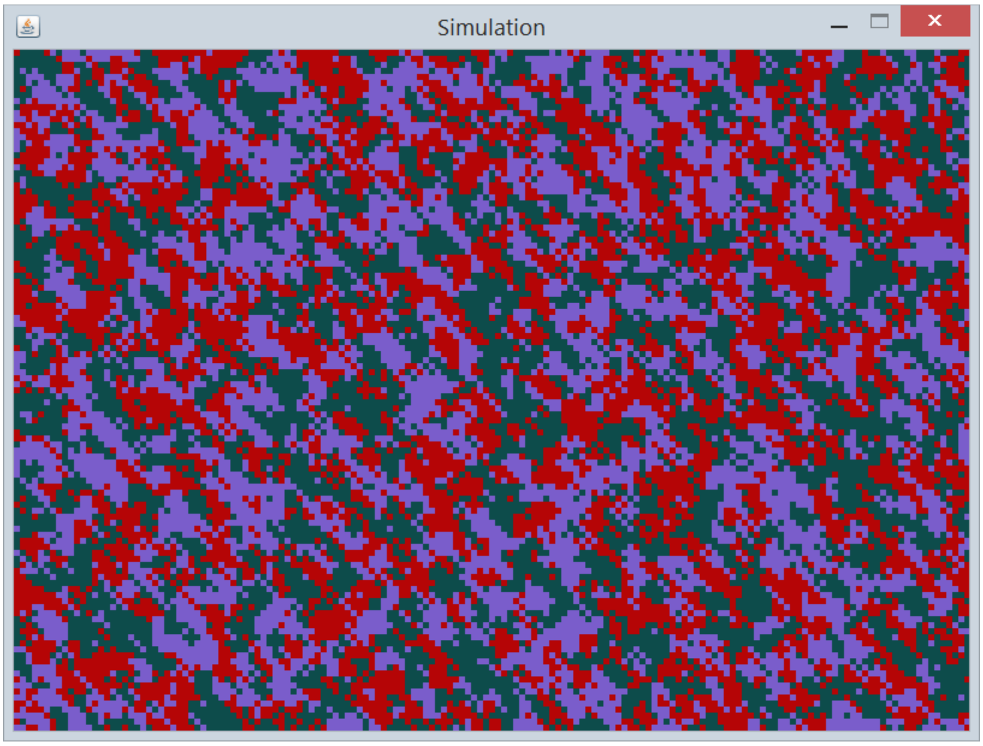
Picture 10 Example of Cellular Automata simulation used for heterogenous distribution of energy



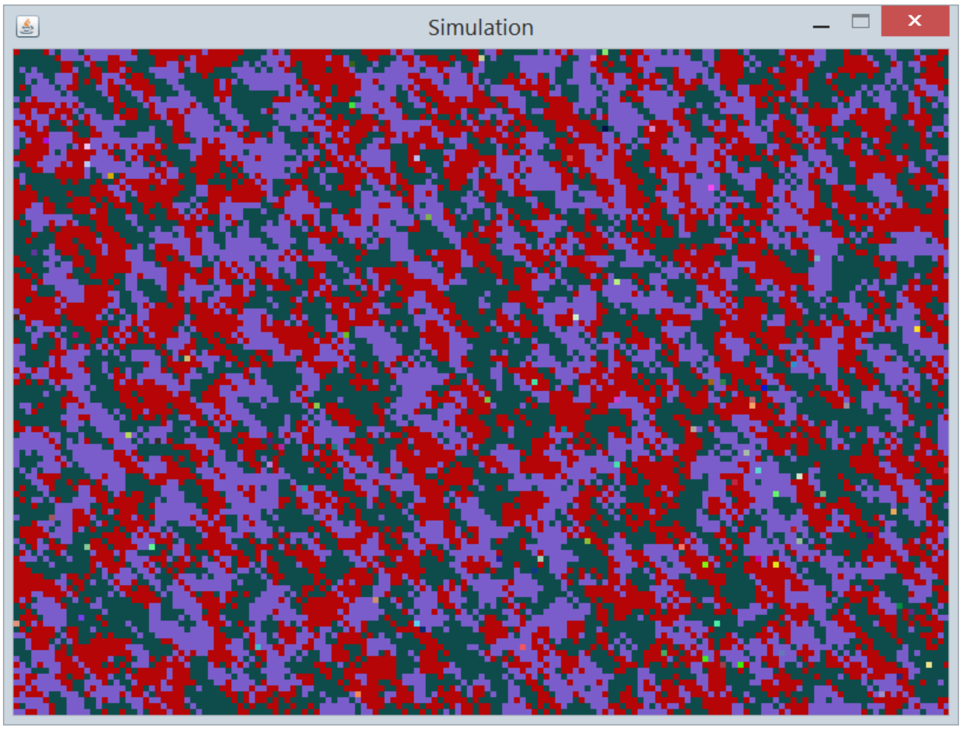
Picture 11 Example of heterogenous energy distribution

3.2 MC static recrystallization algorithm – nucleation

On the **Picture 13** we can see the example of adding 100 new nucleons to finished simulation (**Picture 12**).

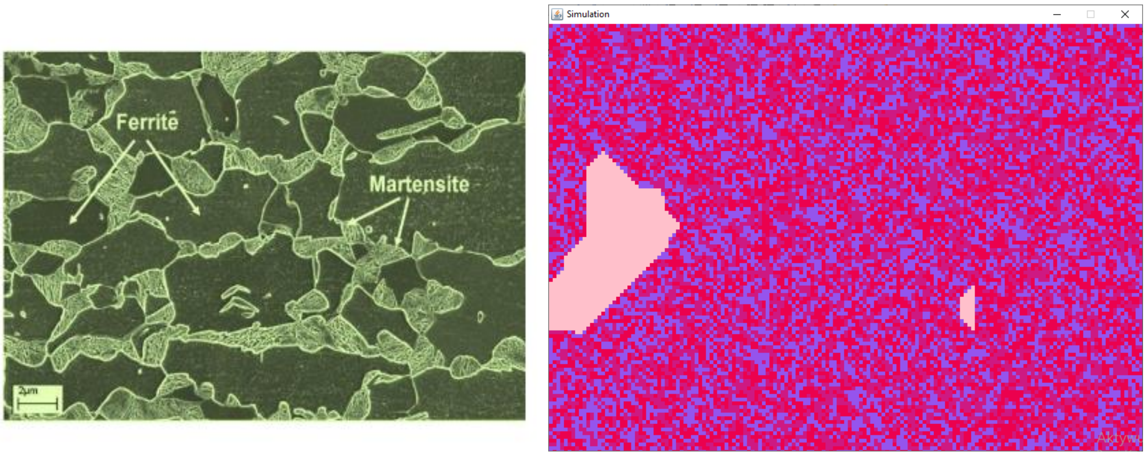


Picture 12 Example of another Monte Carlo simulation

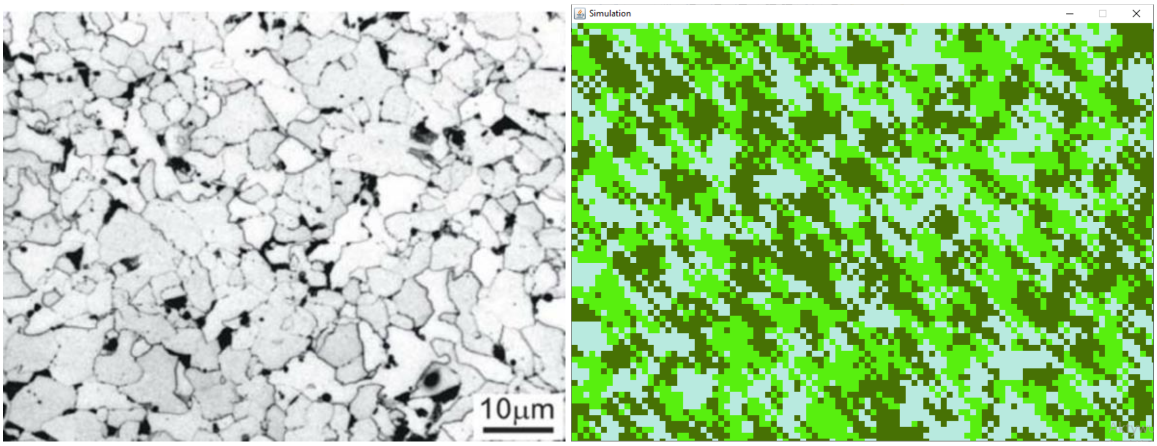


Picture 13 Example of adding new nucleons to already finished simulation

4. Comparison to real microstructure



Picture 14 – Comparison od real DP steel and generated one
(http://ispatguru.com/dual-phase-steels)



Picture 15 – Comparison of real mictrostructure (Steel V-N) and generated one
(https://www.researchgate.net/figure/Optical-micrograph-showing-the-ferrite-and-pearlite-microstructure-in-the-final-strip-for_fig1_29814367)

5. Conclusions

- Making this project allowed us to learn how the Monte Carlo method works.
- The main disadvantage of Monte Carlo mehod is that it takes quite long to make a calculation
- Generated microstructures differ a lot from real ones – that's caused by an imperfect implementation of model
- It allowed us also to train more the programming language which we choose for this project.
- Despite adding many features, you can still develop and improve this project.