Patryk Jakubiec

Multiscale Modeling – report #2

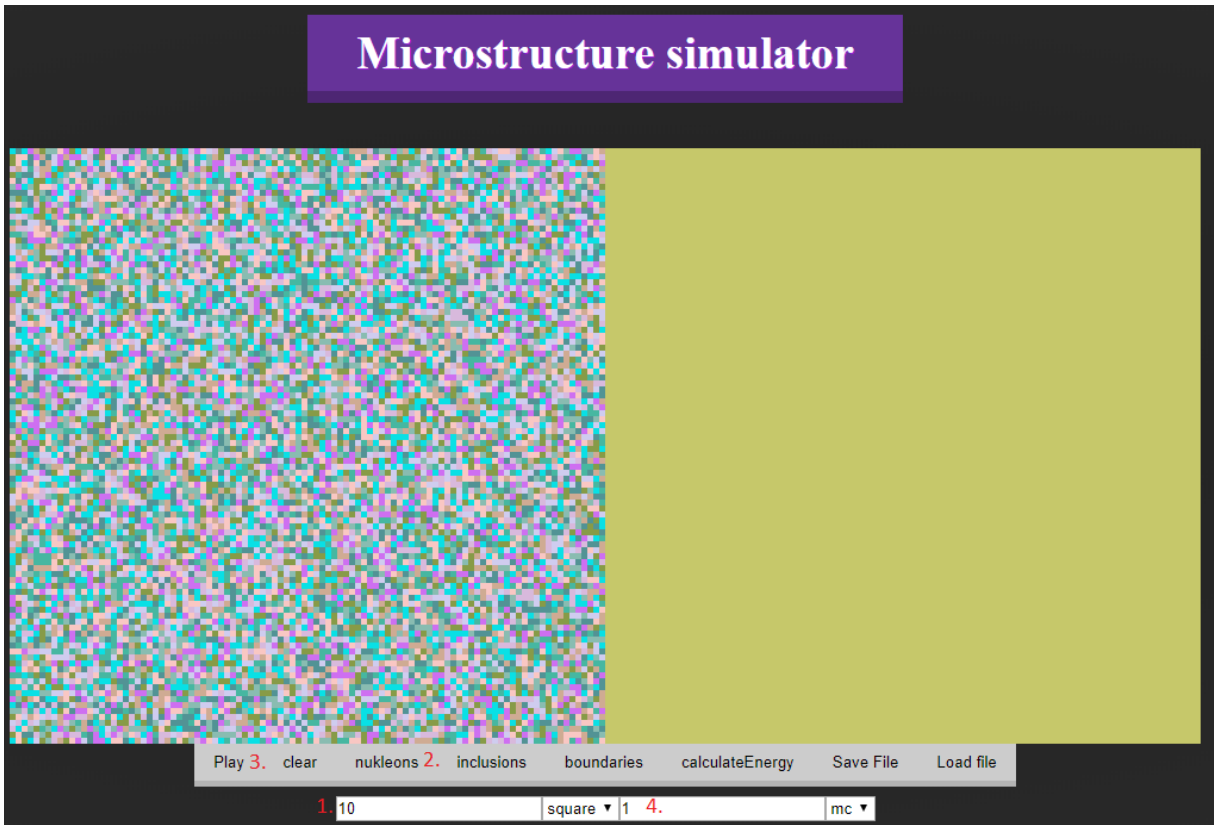
Introduction

Monte Carlo grain growth algorithm was implemented using the cellular automata structures from previous part of the application which are characterized by:

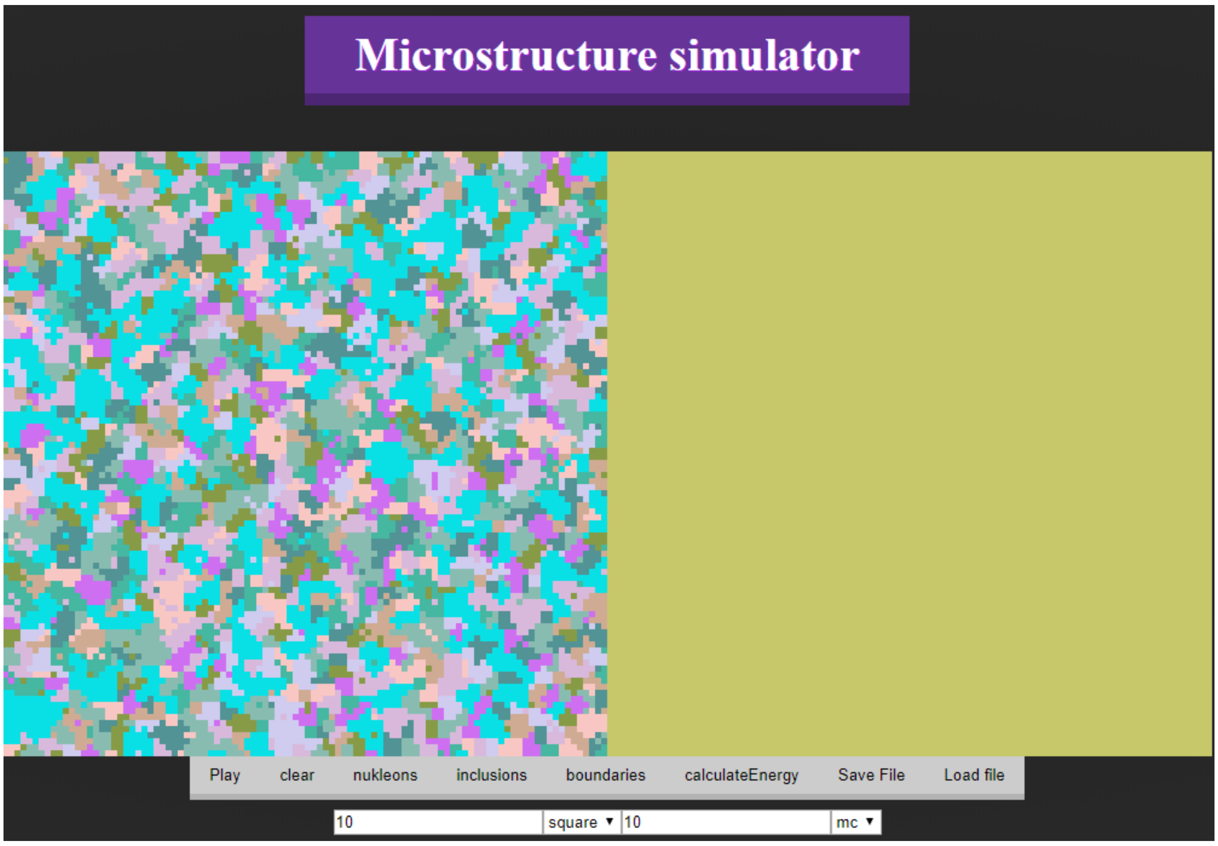
1. Grid-space where cells live. In this project grid is represented by a two dimensional array although cellular automaton support infinite number of dimensions.
2. Each cell has it's own state. The simplest example has the two possibilities of state-on and off. This project uses RGB colors to define state. White one represents off, black one sets cell to blocked state and any other colors represent on state

Monte Carlo growth

User is able to set the number of nucleon types into a text area[1] which will be randomly generated onto grid. It's done after pressing the *nucleons* button[2] which is responsible for selecting a two random numbers from the range of grid size and placing a new nucleon into the cell located under those coordinates. The process ends when there are no empty cells left. The *play* button[3] runs the Monte Carlo algorithm. Program creates a list of possible coordinates to choose one from it randomly and calculate the energy for that cell. When the process of calculating energy for that cell is finished, the coordinates of that cell are deleted from the list. The loop continues until the list isn't empty which ensure that all of the nucleons will be chosen. The whole process repeats the number times provided by the user in a text area[4]. After calculating all of the energies the old grid is replaced with an updated one.


Microstructure simulator
Play 3. clear nukleons 2. inclusions boundaries calculateEnergy Save File Load file
1. 10 square 1 4. mc

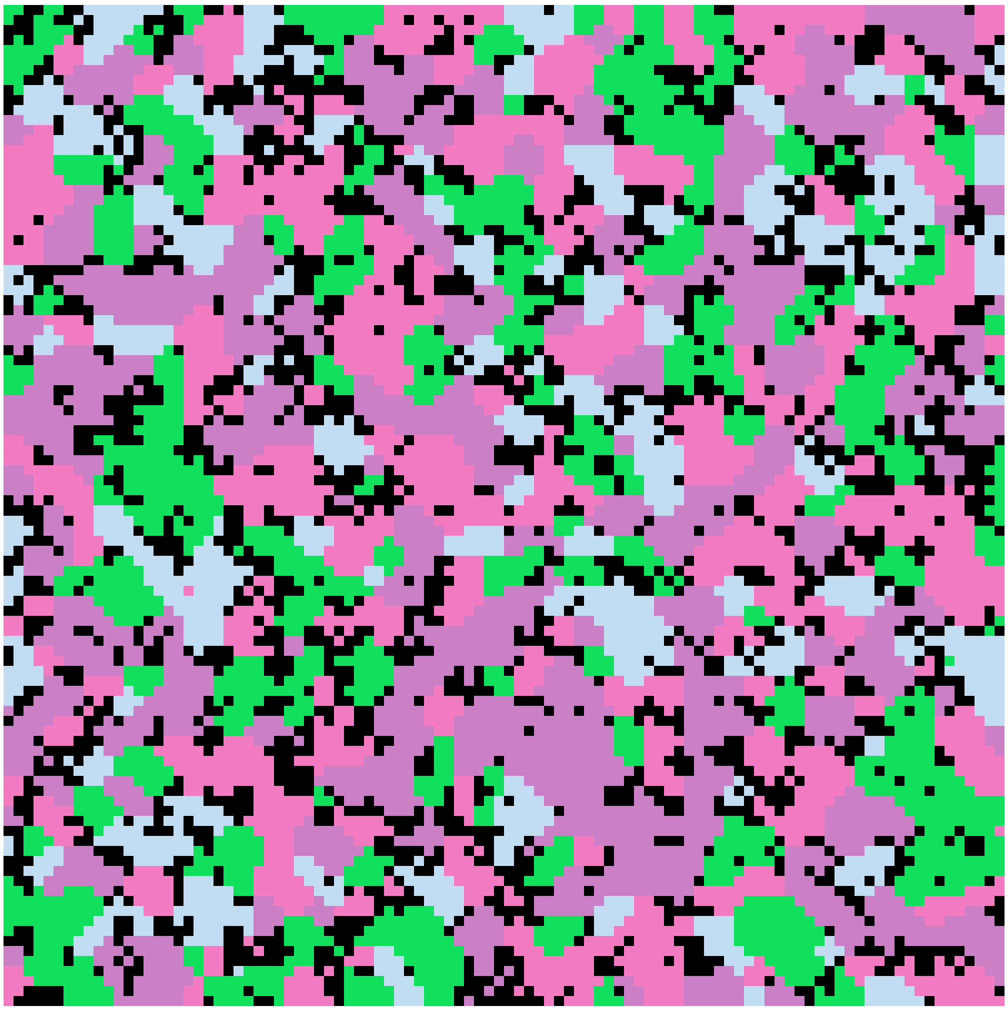
Picture 1: Graphic user interface


Microstructure simulator
Play clear nukleons inclusions boundaries calculateEnergy Save File Load file
10 square 10 mc

Picture 2: Grid after 10 iterations of Monte Carlo growth

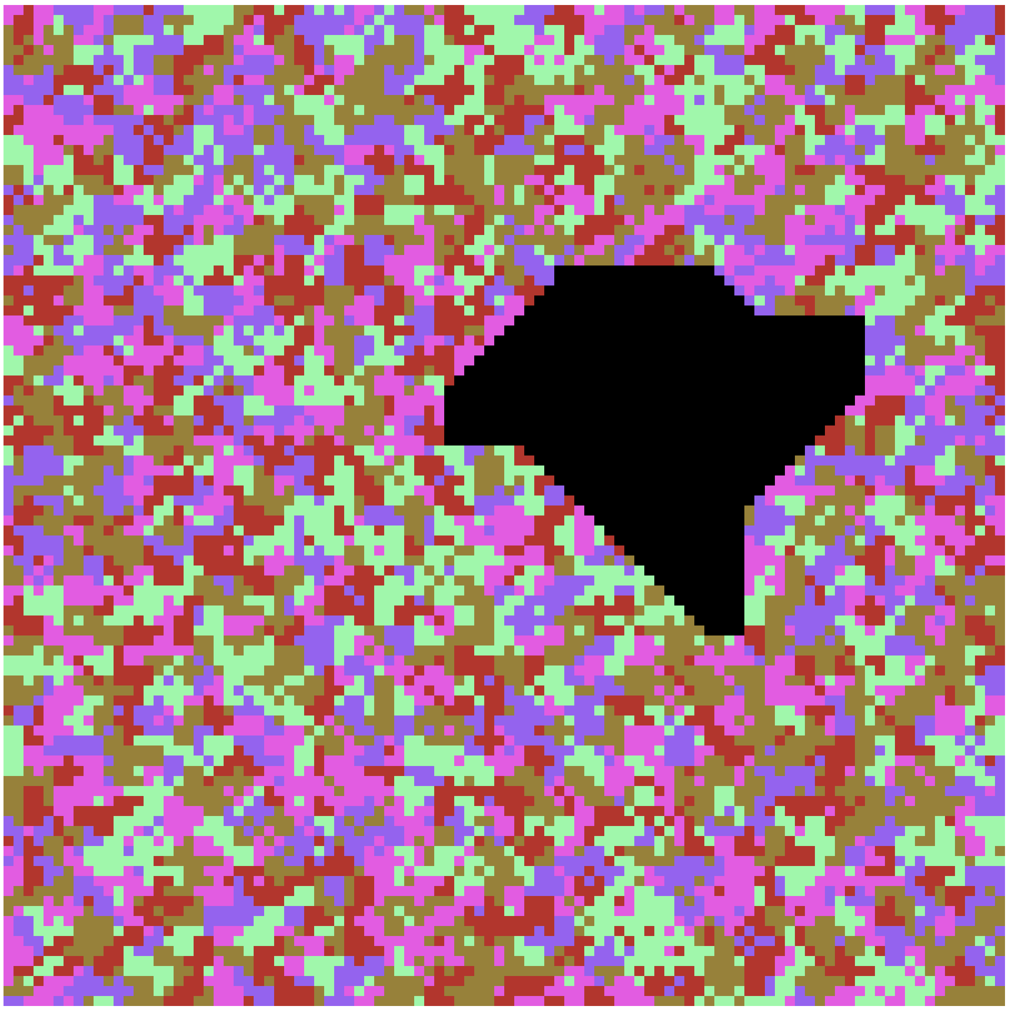
Dual Phase

User is able to block preferred nucleons by simply clicking with left mouse button on them. The selected grains will turn into a second phase and won’t participate in further growth of the grid.



Picture 3: The Monte Carlo growth with blocked phase

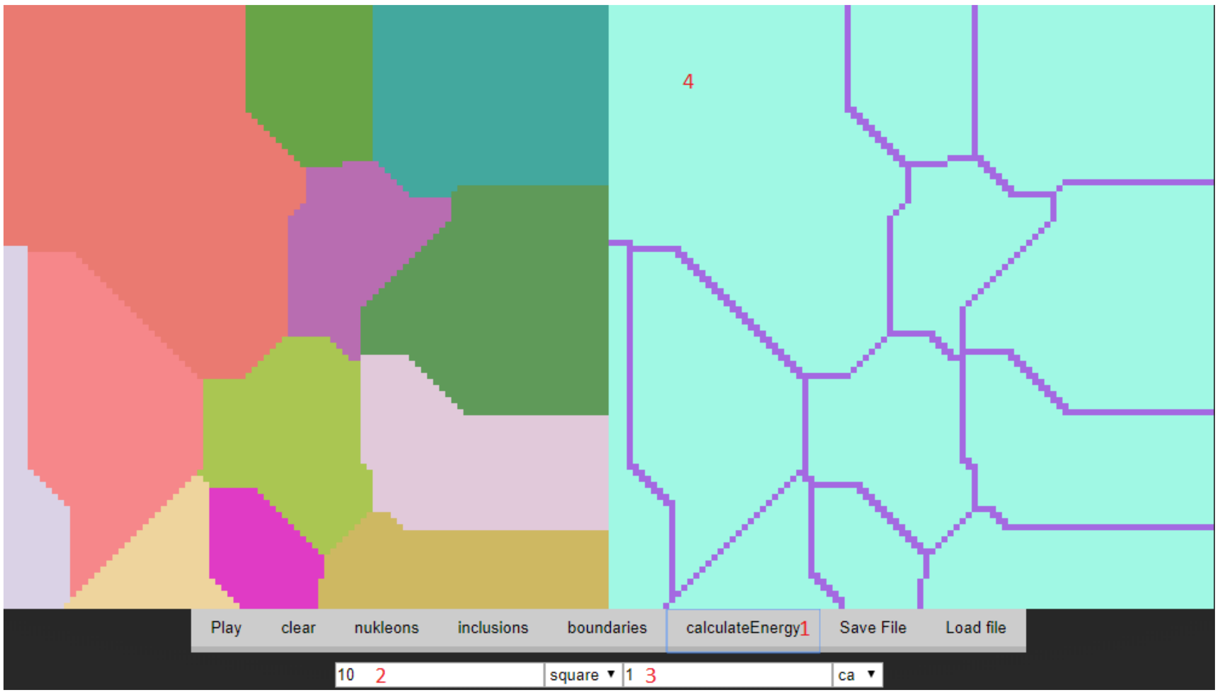
User is also able to generate a cellular automata simple growth, block phase, clear the rest of the cells and generate Monte Carlo grid with the previously blocked phase.



Picture 4: Monte Carlo grid with blocked simple phase

Energy distribution

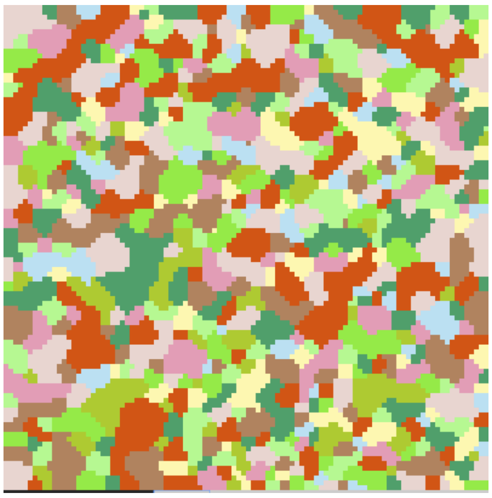
The application lets user to add an energy into the cells. It's done by clicking the *calculate Energy*[1] button which takes values from text area[2] and [3] which correspond to amount of energy added into boundaries[2] and into the phases[3]. The distribution of the energy is visualized on the right grid[4].

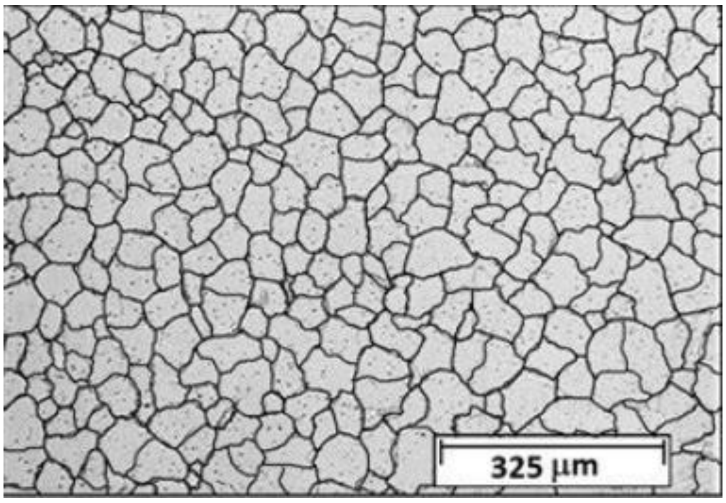


Picture 5: Heterogeneous energy distribution

Analysis

The pictures below present the microstructure generated by the application using Monte Carlo growth algorithm and a real photography of the microstructure. The application presents an image similar to the one received by taking a photography of the microstructure. It also does look more realistic in comparison to the simple cellular automata grain growth





Conclusions

This report presents the web application designed to handle process of the microstructure simulation. The application is written using ReactJS and the Monte Carlo grain growth logic is based on *cellular automata*. The application has all the must have features like setting the number of phases or creating a dual phase microstructure. The performance of the application can and will be improved in future by using a React Redux.