Patryk Jakubiec

Multiscale Modeling - report #1

Introduction

Simple grain growth was implemented by using cellular automata which is a discrete model studied not only in applied computer science but also in mathematics and physics. Cellular automata are characterized by:

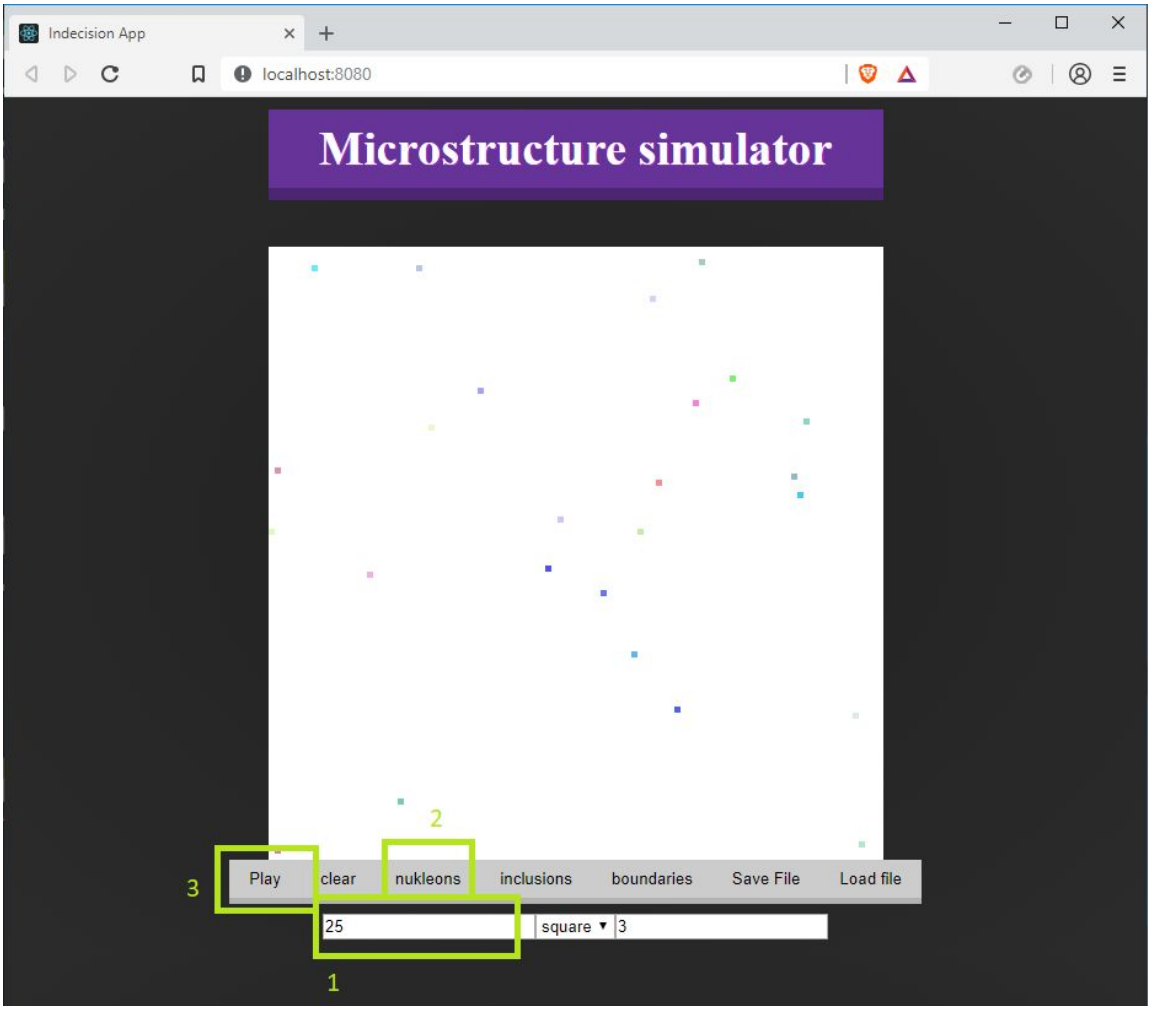
1. Grid - space where cells live. In this project grid is represented by a two dimensional array although cellular automaton support infinite number of dimensions.
2. Each cell has it's own state. The simplest example has the two possibilities of state - on and off. This project uses RGB colors to define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

Whole project was implemented using ReactJS technology which is a great solutions when it comes to re-rendering only some components and not the whole webpage.

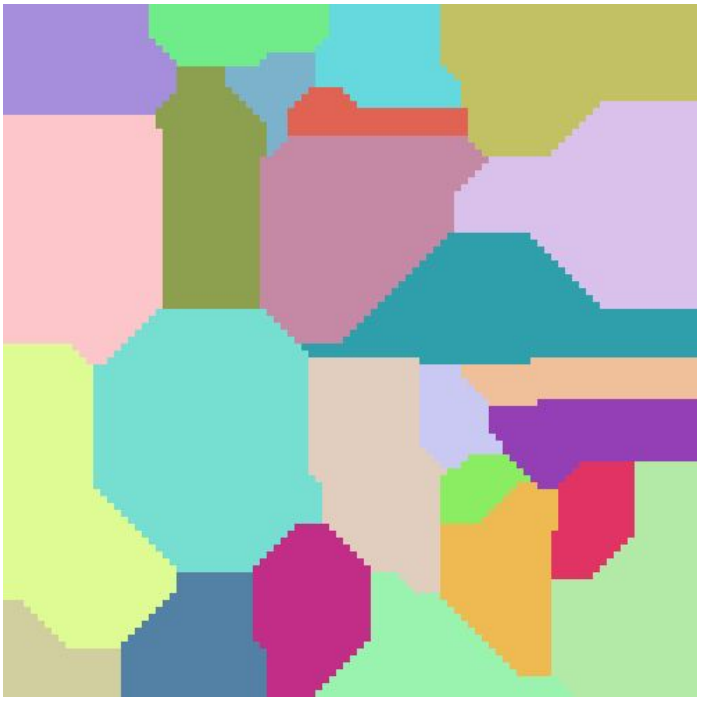
Simple grain growth

User is able to set the number of nukleons which will be randomly generated onto grid. It's done by typing a preffered ammount of nucleons into a text area [1](which is later also responsible for number of inclusions inserted into the microstructure) and then pressing the *nucleons* button[2]. The *play*[3] button runs the calculations and shows the final microstructure.

The whole window is resizable and works well on both small and large displays.



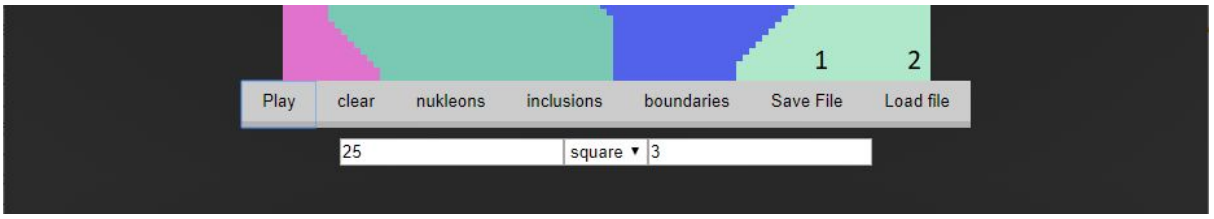
Picture 1: Graphic user interface



Picture 2: Final look of the microstructure

Import & Export

This project allows user to both import and export microstructure representation as a *.png* or *.json* files. The json files are handled via *Save File*[1] and *Load File*[2] buttons provided in GUI while png files are saved using the right click on canvas context menu, which is unable to show on a screenshot because context menu is clear when the browser lose focus.

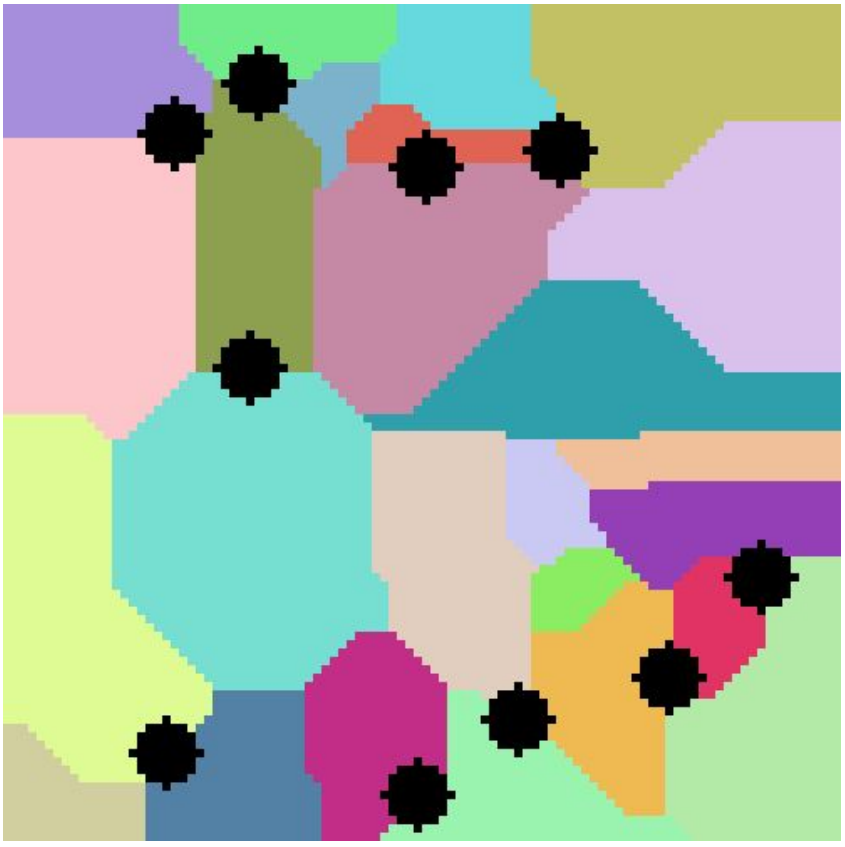


Picture 3: Import & Export buttons

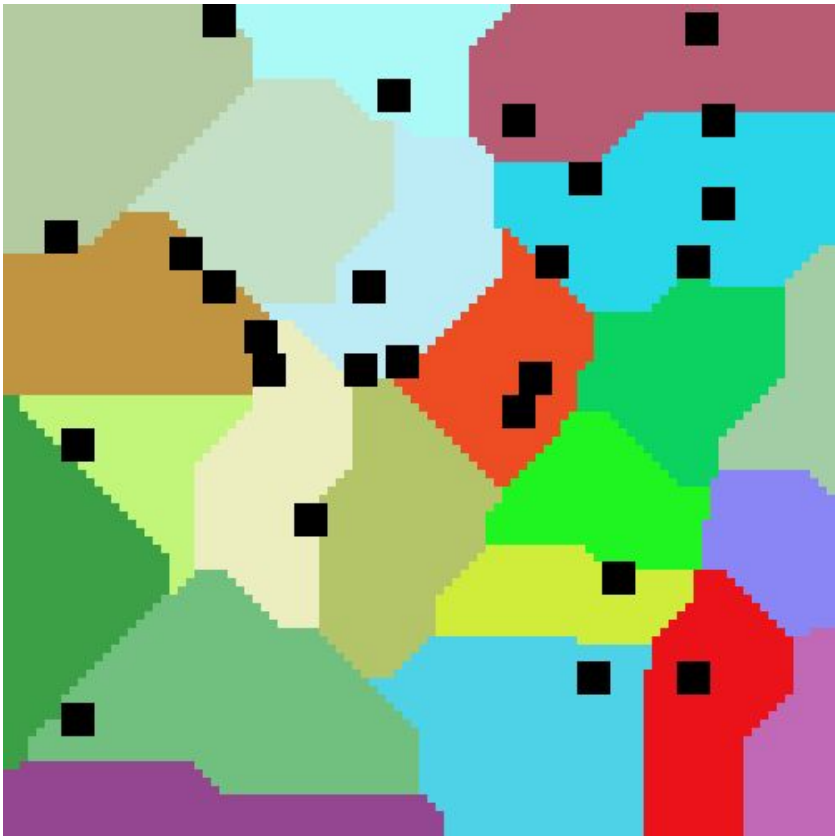
Inclusions

Application provides user an ability to add inclusions into the microstructure. It is possible to do that before and after the simulation is finished. User can chose the number[1] of inclusions inserted, their shape[2] and also a radius[3]

Picture 4: Inclusions parameters



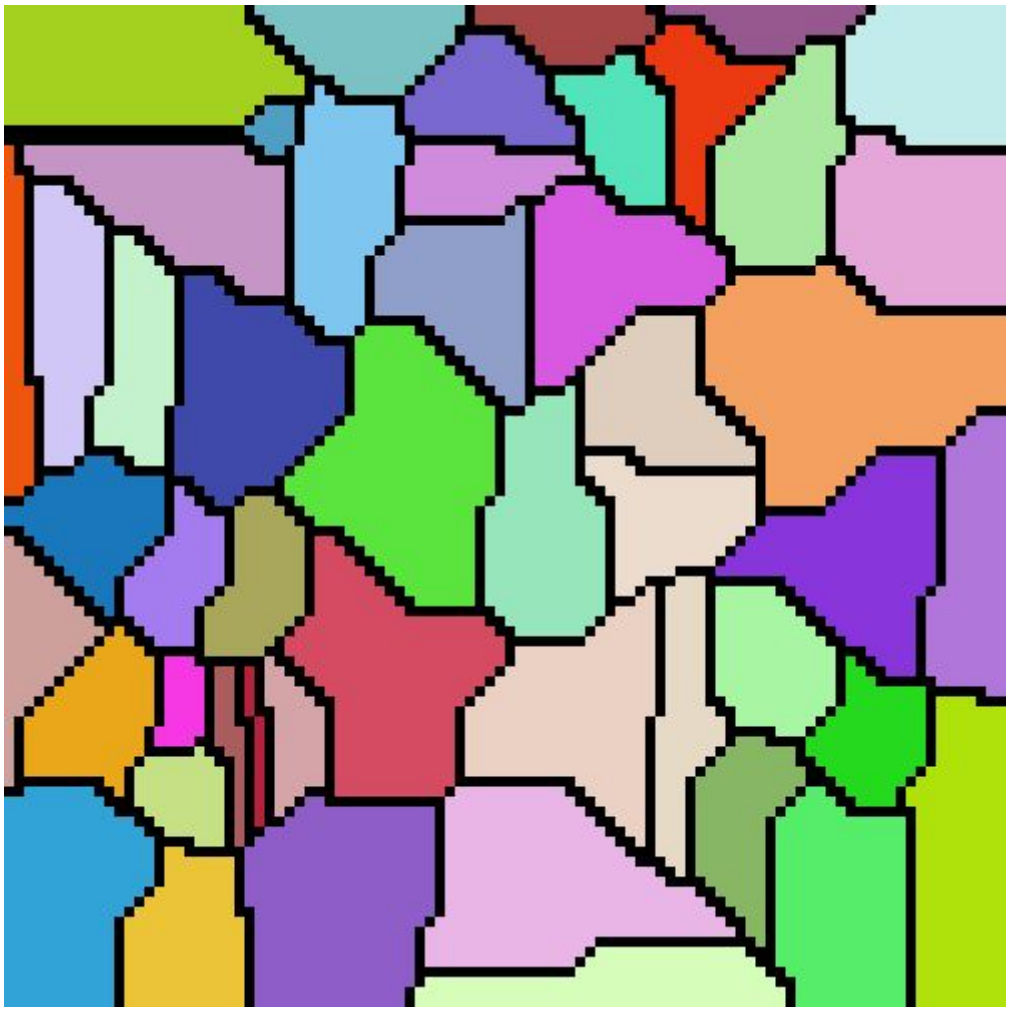
Picture 5: Round inclusions added after simulation



Picture 6: Square inclusions added before simulation

Boundaries

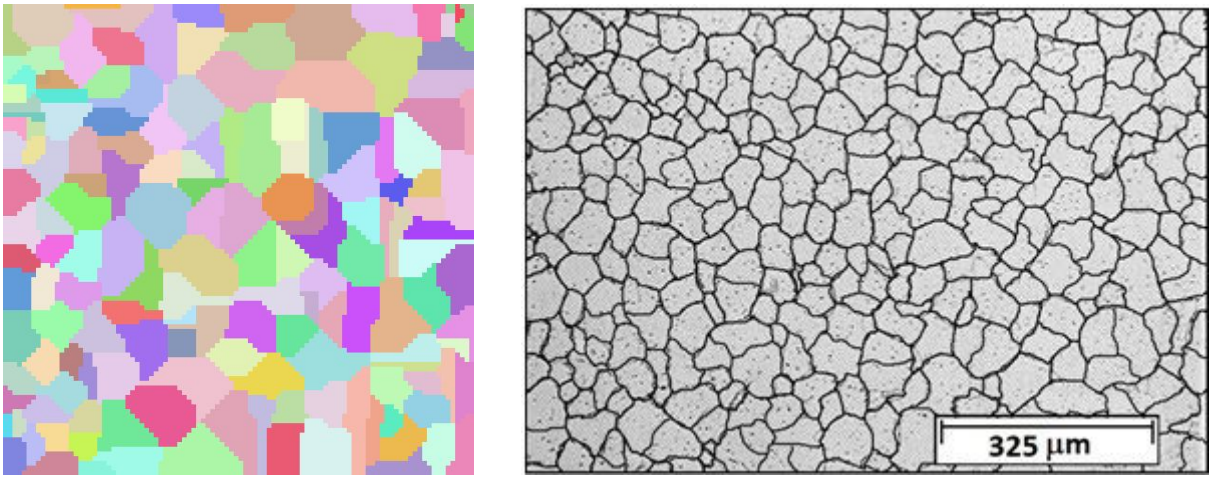
User is able to mark the boundaries of the microstructure grains by simply clicking the *boundaries* button.



Picture 7: Marked boundaries

Analysis

The pictures below presents the microstructure generated by the application and a real photography of the microstructure. The application presents an image similar to the one received by taking a photography of the microstructure altought it has less details



Picture 8: Comparsion of microstructures

Source: https://srizam.blogspot.com/2010/10/microstructure-pengukuran-nombor-saiz_02.html

Conclusions

This report presents the web application designed to handle process of the microstructure simulation. The application is written using ReactJS and the grain growth logic is based on *cellular automata* algorithms. Thanks to CA simulation results are much more understandable as it is well visually represented. It is easier to understand grain growth, when it is possible to see simulation of it. The renders of microstructures generated by the application are simillar to the real life ones, especially after marking the boundaries of the grains.