

Student: Kamil Siuzdak

Field of study: Applied Computer Science

Subject: Multiscale modeling 2020/2021

Report

The entire project was carried out in 5 stages, as planned. The main task was to generate a matrix on the variables X (determining the width) and the variable Y (determining the height), and then forming a grain growth on it, which begins in a single cell. Thanks to various functions, you can personalize your expectations.

Realized individual requirements:

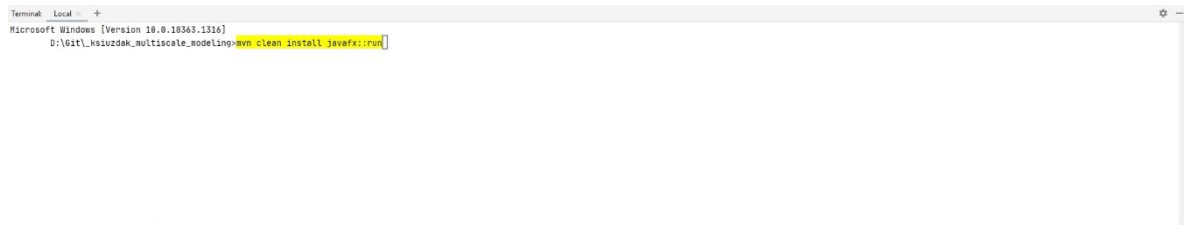
1. Enter the X size of the matrix (example 300, the larger the matrix the more detailed)
2. Enter the Y size of the matrix (example 300)
3. Enter the Number of Grains
4. Transition between borders as a form (Yes / No)
5. Creation of Shape Control, defined by the Number of Probability (the smaller the grains will be more like scattered stains)
6. Enter the Number of Inclusions
7. Enter the Inclusion Radius
8. Placement of Inclusions at borders with the choice of form (Yes / No)
9. Choosing the Shape of Inclusions (Circle / Square)
10. Import from a text file
11. Export to a text file
12. Export to bitmap
13. Selecting a grain and creating a structure - Substructure and Dual Phase
 - a) Dual Phase - after growing, we choose grains that change the color to the same. Then we have several grains of one color and the rest of the grains, just like each one in a different color.
 - b) Substructure - we carry out the usual growth of grains, then in the grains that have grown anew we generate the embryos of new grains, these embryos grow but only in the area of the grain in which they were located, i.e. the border of their growth is the border of each grain.
15. Select the grains to get boundary marking
16. Removal of grains contents, leaving only boundaries of before created grains

Application Language: Java

Used Library: JavaFX version 15 (an open source, next generation client application platform for desktop, mobile and embedded systems built on Java)

Used Software: IntelliJ IDEA

The main assumption was to make the application as clean and easy to use as possible so that the user would know how to navigate it at the first contact of starting the program.



Screenshot 1 Starting the app with a command: mvn clean install javafx::run

This is how the application interface looks like (the program is responsive, it adjusts to the size)

The screenshot shows the 'Grain Growth' application window. The title bar reads 'Grain Growth'. The main content area has a title 'Grain Growth' at the top. Below it, there are nine numbered annotations pointing to specific input fields:

1. X size: (text input)
2. Y size: (text input)
3. Number of cells: (text input)
4. Boundary transition: (dropdown menu, currently set to 'NO')
5. Number of probability: (text input)
6. Number of inclusions: (text input)
7. Inclusion radius: (text input)
8. Inclusion boundary: (dropdown menu, currently set to 'NO')
9. Inclusion shape: (dropdown menu, currently set to 'SQUARE')

At the bottom, there are three buttons: 'Generate', 'Shape Control', and 'Import'. To the left of the 'Generate' button is a blue link 'Generate matrix'. To the right of the 'Import' button is a blue link 'Import from file'.

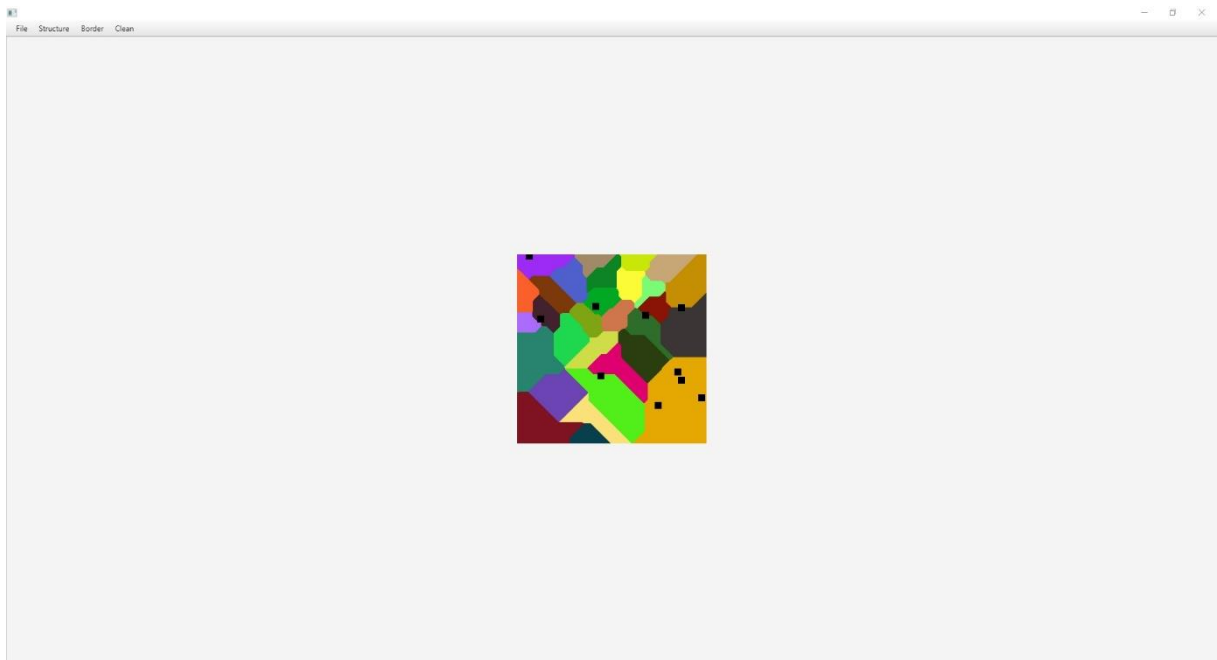
Screenshot 2 The first view after starting the program. Numbers assigned to “Realized individual requirements”

The screenshot shows the 'Grain Growth' application window with the same layout as the first screenshot, but with the input fields filled with values:

- X size: 250
- Y size: 250
- Number of cells: 20
- Boundary transition: YES
- Number of probability: (empty)
- Number of inclusions: 30
- Inclusion radius: 10
- Inclusion boundary: YES
- Inclusion shape: CIRCLE

The buttons 'Generate', 'Shape Control', and 'Import' are still present at the bottom.

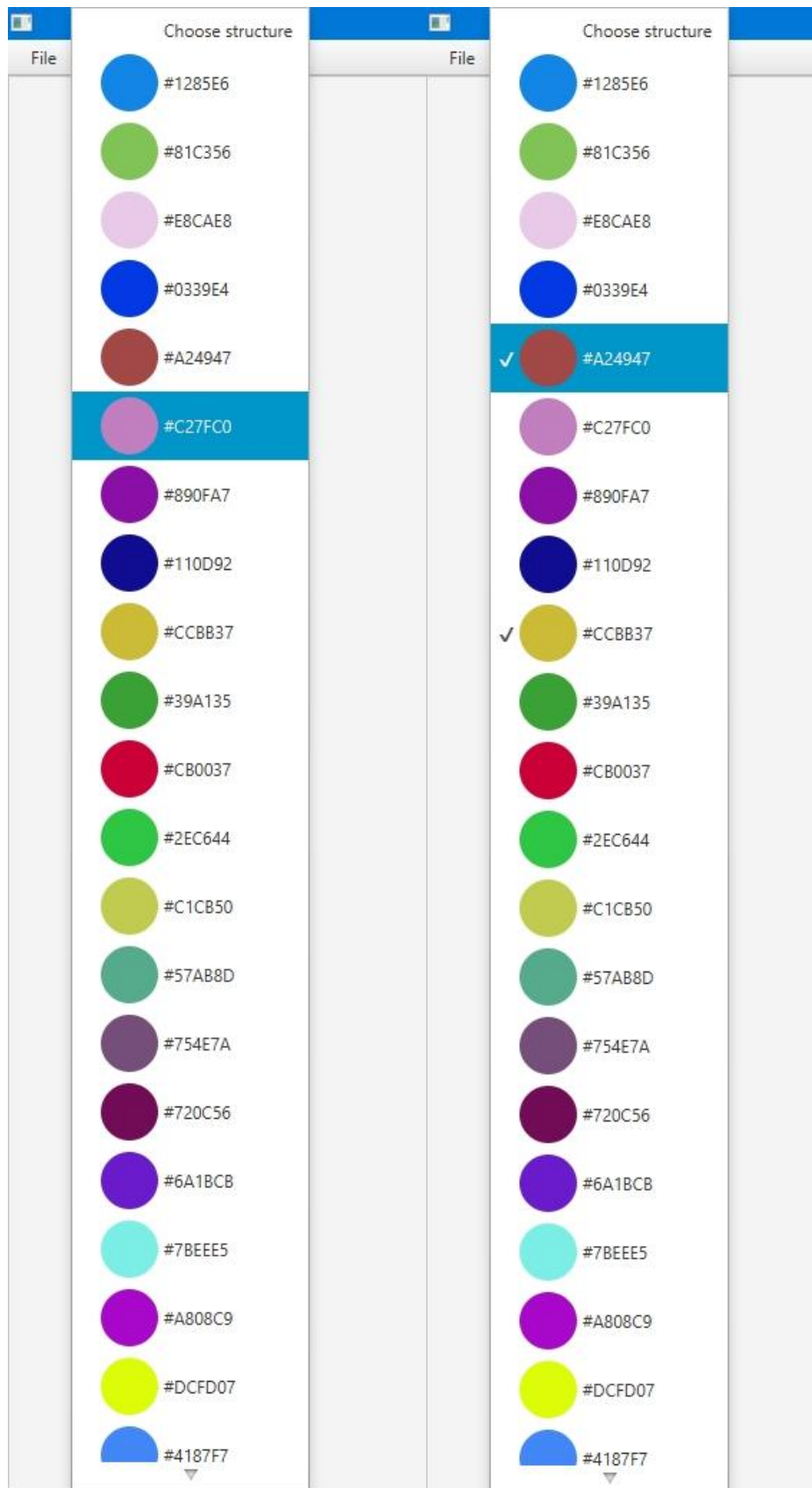
Screenshot 3 View with completed forms



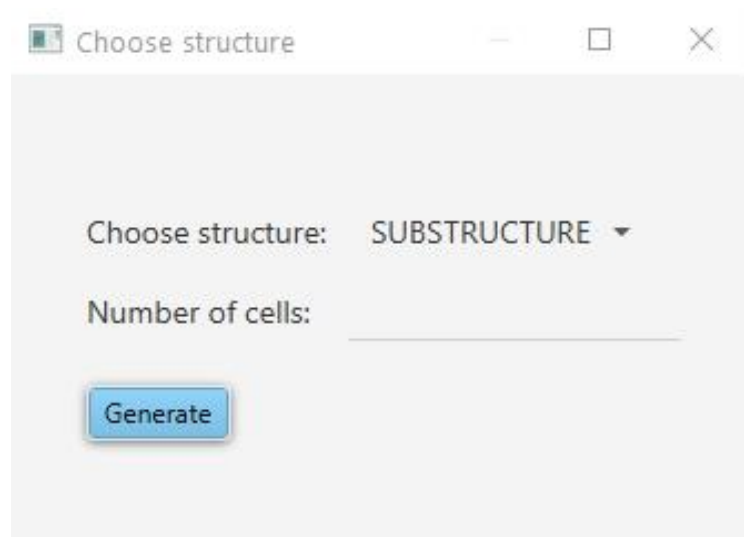
Screenshot 4 Generated matrix with specific parametrs



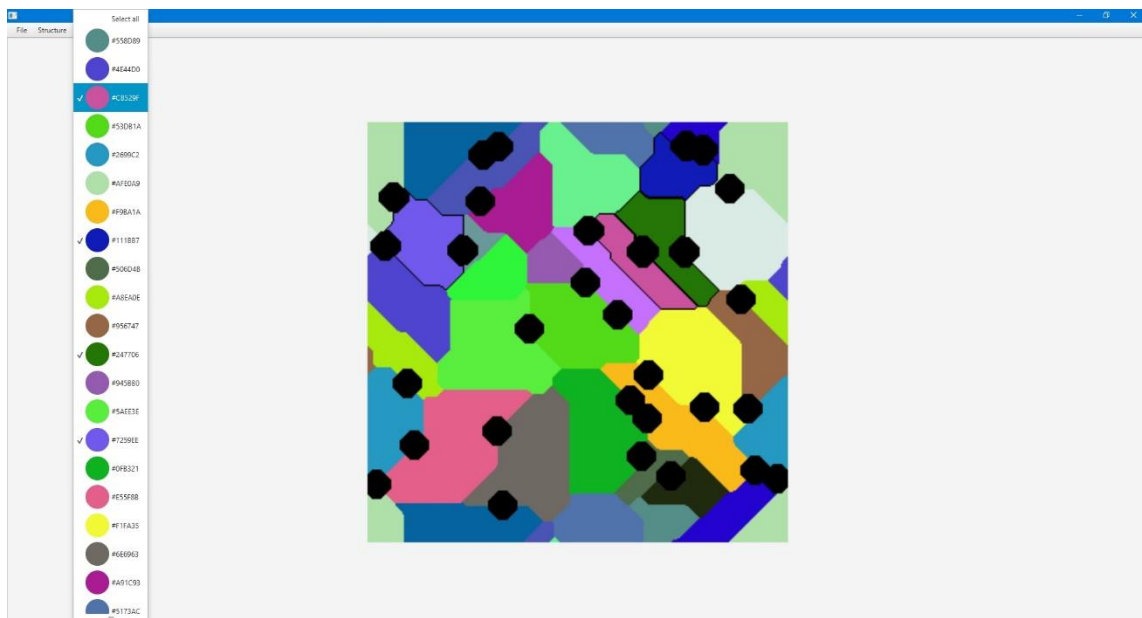
Screenshot 5 Possibility to enlarge the view with the mouse scroll



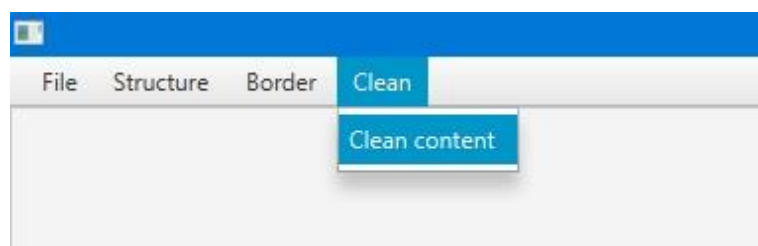
Screenshot 6 Select colors for the dual phase or substructure



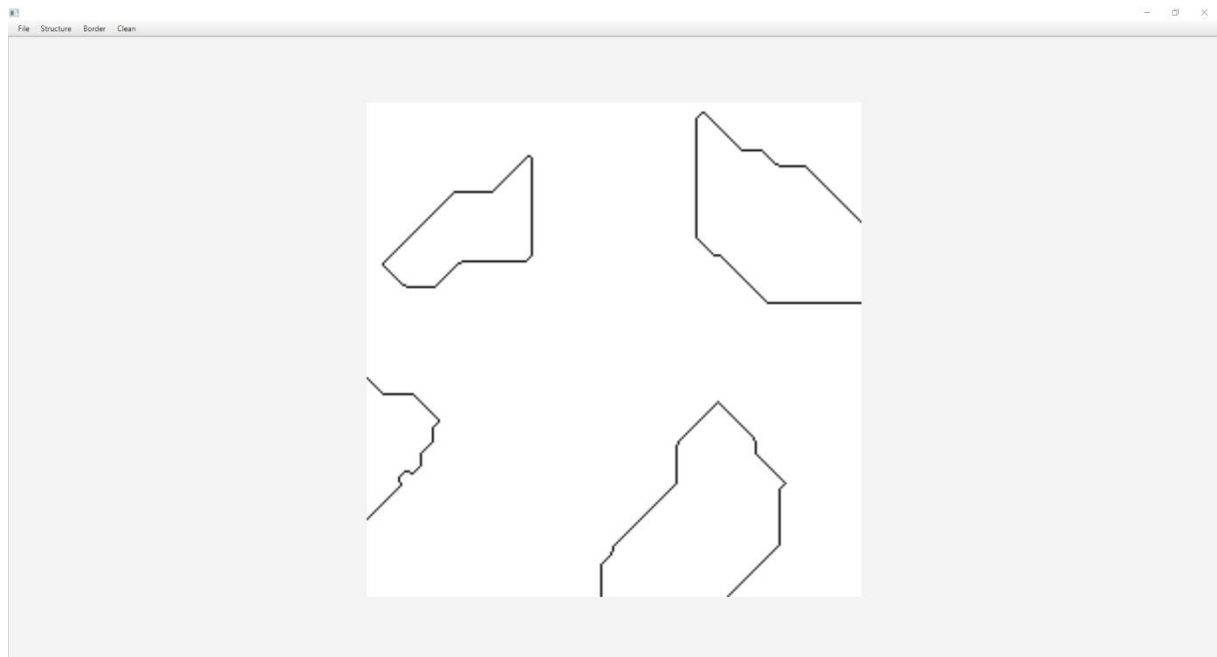
Screenshot 7 Choose structure you would like to generate, then enter the new number of grain



Screenshot 8 Mark the grains to form the boundaries



Screenshot 9 Clean content to achieve only boundaries



Screenshot 10 The obtained grain boundaries after cleaning

Summary

The application will be a useful tool for people who would like to observe the progressive processes of grain growth with different properties

GitHub: <https://github.com/kmlsiuzdak/Multiscale-Modeling>