Multiscale modeling		
Name Surname: **Jan Pikulski**	First report	Date: **14.11.2019**
Index number: **286220**		Grade:

1. Introduction

The main goal of this project was to create a program capable of generating various material microstructures, using the CA method. One of the most important points to follow during the implementation was to keep the program as user-friendly and versatile as possible. That required creating an easy to understand GUI, which not only allows to view the results of each generation, but also makes it possible to customize the whole process based on multiple user inputs.

In order to meet these expectations, Microsoft Visual Studio 2019 along with C# Windows Forms application was chosen for the technological stack in this project. Such a combination allows to create extremely flexible GUI applications in no time, compared even to Python, which comes equipped with multiple graphical libraries. The drag and drop designer of Windows Forms helps making small adjustments directly on the GUI instead of writing complicated lines of code, which then need to be recompiled and tested.

2. Basic interface

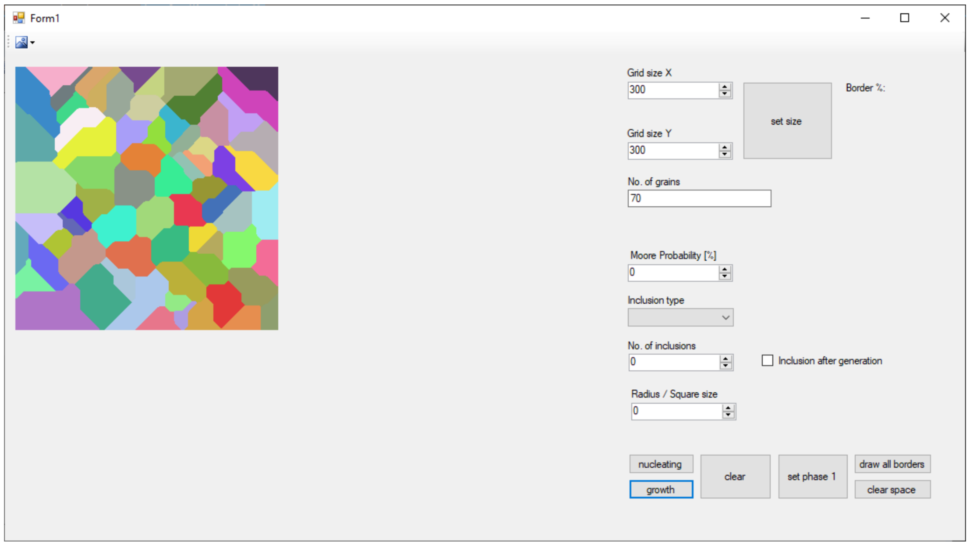
a) The main feature of the whole GUI visible on *figure 1* is the viewport, implemented using the PictureBox utility of WinForms. It allows the user to quickly assess whether the microstructure generation yielded satisfactory results, or it needs to be restarted with different parameters. The size of this viewport can be adjusted using the “Grid size X” and “Grid size Y” input fields, which change the width and height respectively, and clicking the “set size” button.

b) The next and most important control is the “No. of grains” input, which allows the user to set a specific number of grains that will take part in the microstructure generation process.

c) After all aforementioned inputs are provided, user needs to first click the “nucleating” button, which randomly places all grains across the viewport. Each grain has a unique color and ID (which is a concatenation of the R, G and B numbers of the assigned color).

d) The last button necessary to generate a microstructure is the “growth” button. After it is clicked, the CA method begins the first iteration, using Moore neighborhood. The algorithm ends when all cells have an ID assigned to them.

These are the basic functionalities, which allow to generate a simple microstructure. *Figure 1* shows an example output of such operations.



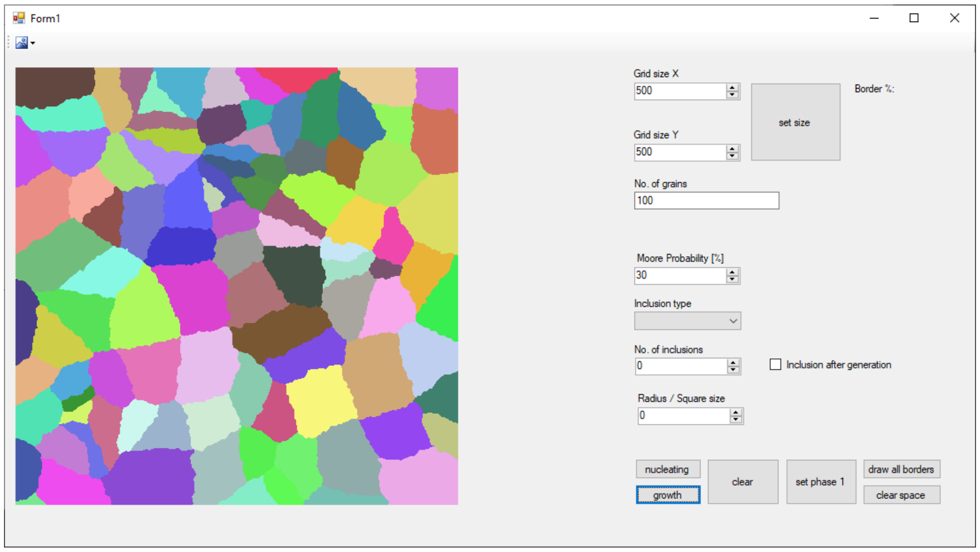
Grid size X: 300
Grid size Y: 300
Border %:
No. of grains: 70
Moore Probability [%]: 0
Inclusion type:
No. of inclusions: 0
☐ Inclusion after generation
Radius / Square size: 0

Figure 1 – Example output generated by the application for 70 grains

3. Optional functionalities

There are also additional functionalities capable of altering the generation process:

a) The “Moore Probability” input modifies the rules of cellular automata growth based on Moore neighborhood. *Figure 2* shows how setting the probability value to 30, modifies the microstructure.



Grid size X: 500
Grid size Y: 500
Border %:
No. of grains: 100
Moore Probability [%]: 30
Inclusion type:
No. of inclusions: 0
☐ Inclusion after generation
Radius / Square size: 0

Figure 2 – The influence of Moore probability on grain boundaries

It is clearly visible that with low probability, the grain boundaries become more jagged and closer to what we can observe in natural microstructures.

b) The application also gives users the ability to generate inclusions either before or after the microstructure is generated. In order to achieve the latter scenario, the “Inclusion after generation” checkbox must be checked. User selects one of two types of inclusions: circular or square, chooses their number and size and proceeds with the generation as normally. Inclusions generated at the beginning can be located anywhere on the picture, however those created after the microstructure is completed, will always be added on grain boundaries. *Figure 3* and *Figure 4* show how the difference between the two processes looks like.

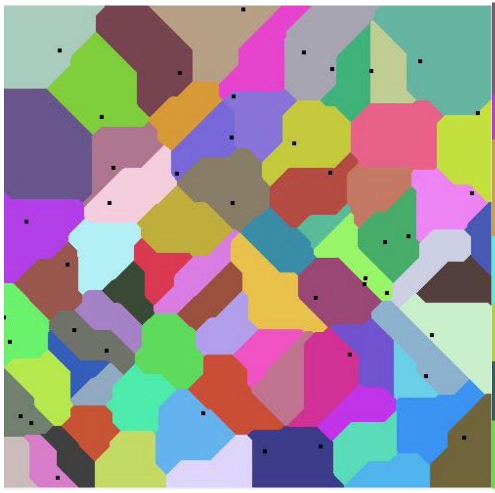


Figure 3 – Square inclusions before generation

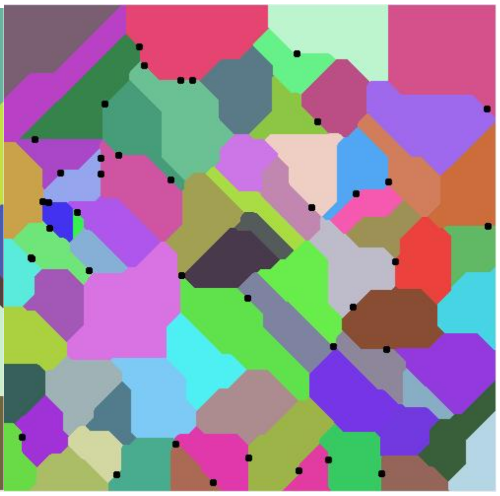


Figure 4 – Circular inclusions after generation

c) By left clicking on grains, user can select as many grains as he wants and set them as Phase 1 using the “set phase 1” button. This allows to create microstructures of dual-phase steels. *Figure 5* and *Figure 6* show the selection of grains and setting the first phase respectively.

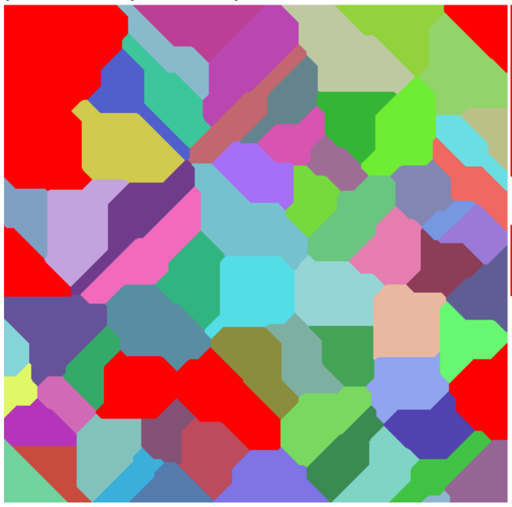


Figure 5 – Grain selection (bright red color)

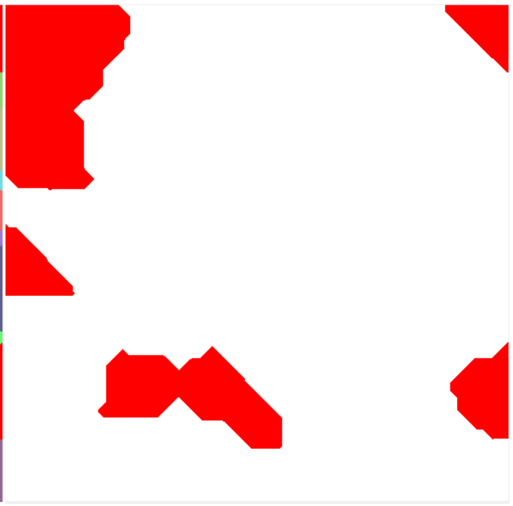


Figure 6 – Setting the first phase of microstructure

Figure 7 shows the completed dual-phase microstructure after second grain growth.

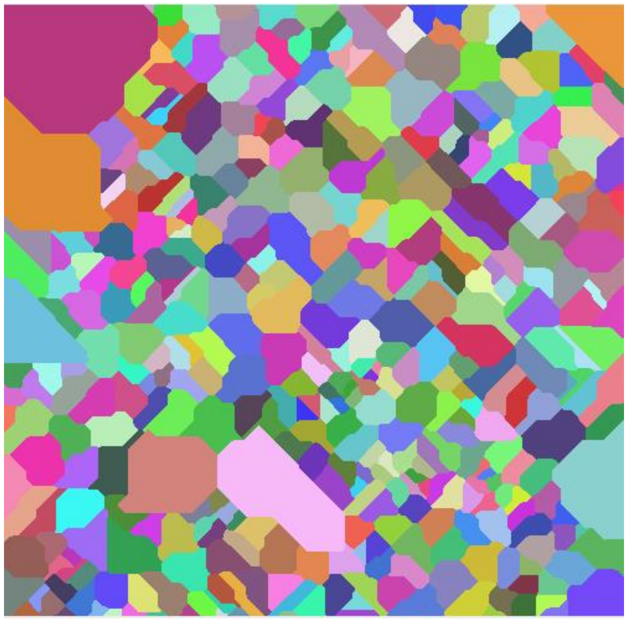
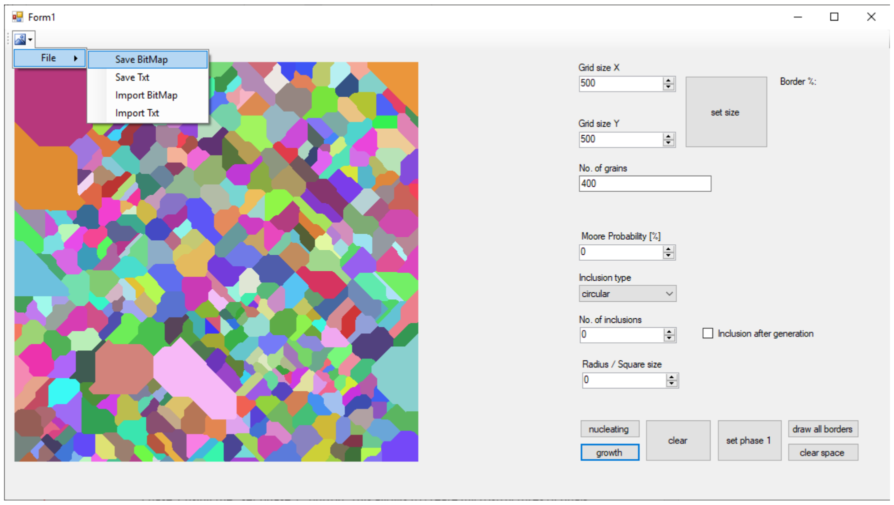


Figure 7 – Completed DP microstructure (bright red selection removed)

d) Additionally, user has the possibility to save and load generated microstructures as either .bmp or .txt files for future use. *Figure 8* shows how to access this functionality.



Grid size X: 500
Grid size Y: 500
No. of grains: 400
Moore Probability [%]: 0
Inclusion type: circular
No. of inclusions: 0
☐ Inclusion after generation
Radius / Square size: 0

Figure 8 – Menu item allowing to save generated microstructure

e) This program also allows the user to draw borders around all or specific grains for better visibility. It can be done either by clicking the "draw all borders" button or right-clicking on a grain. After that the "clear space" button can be clicked to remove everything except the boundaries. Effects of this operation can be seen on *Figure 9* and *Figure 10.* After this process, user can see how much space of the whole microstructure is occupied by borders. The percentage is displayed in the upper-right.

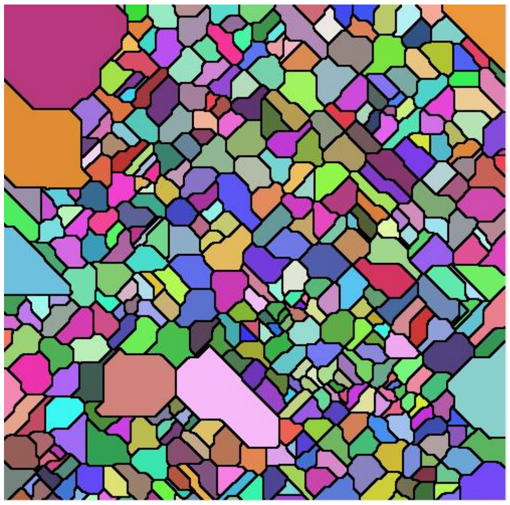


Figure 9 – Borders drawn around all grains

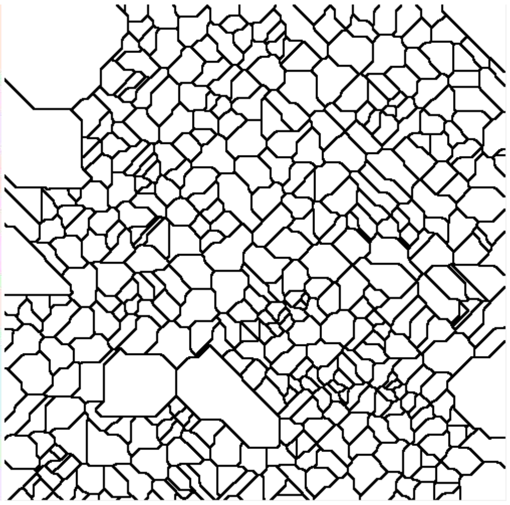


Figure 10 – All grains removed except the borders

4. Comparison and Conclusion

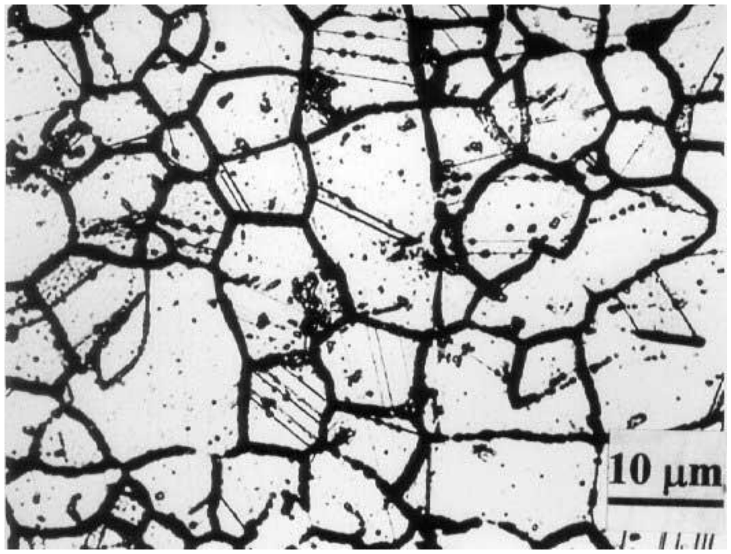


Figure 11 - Microstructure of a sensitized type 304 stainless steel

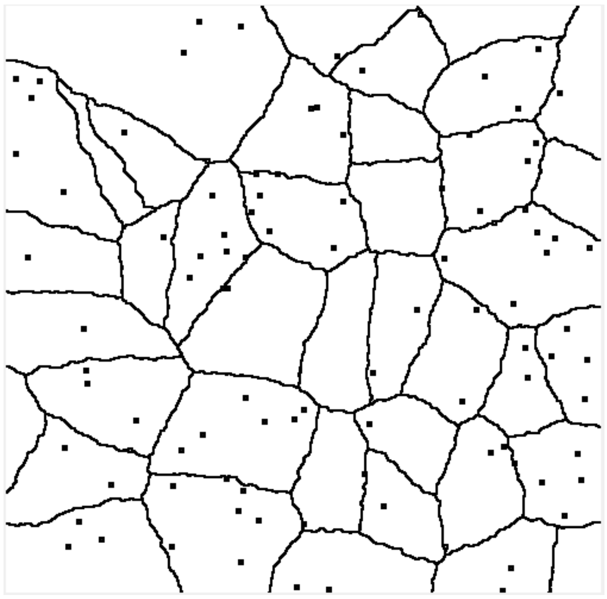


Figure 12 – Microstructure generated with inclusions

Figure 11 (https://commons.wikimedia.org/wiki/File:Microstructure_of_a_sensitized_type_304_stainless_steel.jpg) show a real life photo of type 304 stainless steel microstructure. On the other hand *Figure 12* is a microstructure generated with 50% Moore probability and square inclusions.

As we can see, the generated picture might not be a perfect replication of the real-life scenario. However, it is very close to the original and for analysis purposes it can be more than enough. The application is capable of generating different kinds of grain boundaries, from completely straight, to almost round. This allows to replicate most real-life scenarios to a certain degree of accuracy.

It is also worth noting, that there are still some limitations to this application. For example, there is no way of marking other material imperfections except for inclusions of different size. It creates a clearer and more simplified microstructure. It does not necessarily make it inaccurate or unusable for tests, however it is a point worth keeping in mind.

All technologies used during the implementation of this application proved to be extremely useful and almost irreplaceable. The combination of power and ease of use, coming from both WinForms and C# itself, made it possible to generate enormous amount of good quality code in a really short time. In addition, C# provides multiple functions designed for handling various bitmap operations, which can be very costly when written incorrectly. Thanks to this, the main responsibility of the programmer was to create efficient algorithms which implemented the CA method, as the tools provided by the platform were already extremely well optimized.

Overall, the application meets all the requirements established at the beginning of this project and is well equipped to generate microstructures, which are greatly similar to real-life scenarios.