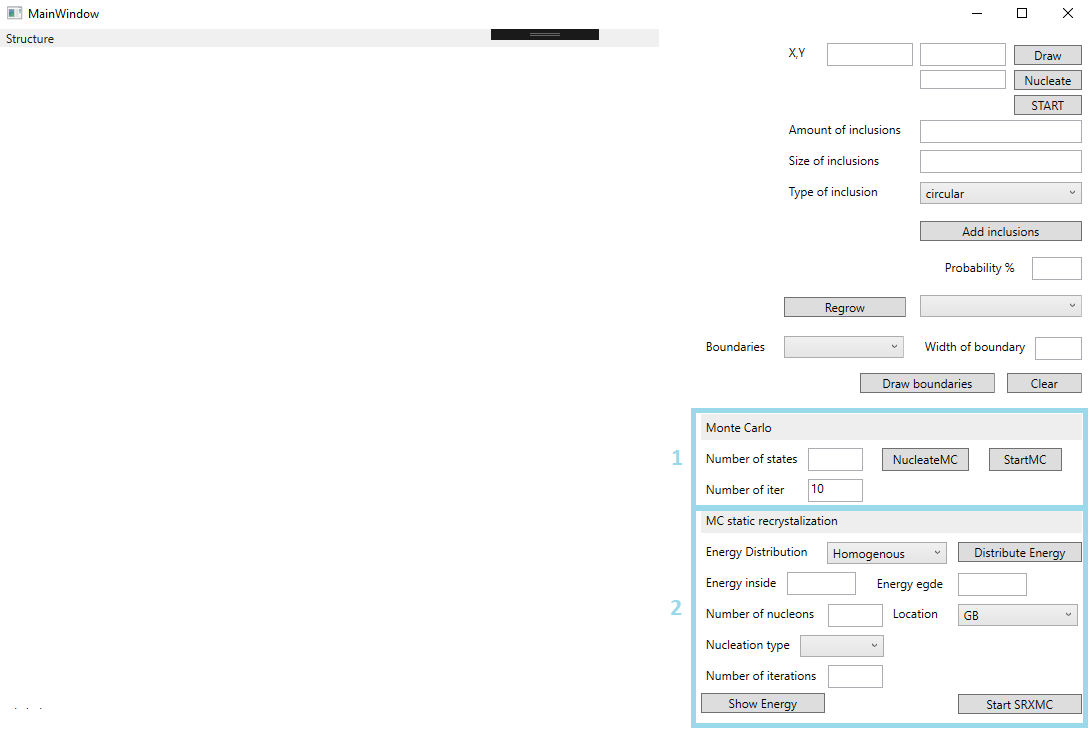
**Karol Rodak**

**MULTISCALE MODELLING – SECOND REPORT**

**1. DESCRIPTION OF GRAPHICAL USER INTERFACE**

Main function of the program is to simulate grain growth in the microstructure and different processes connected with that. During the second part of project main task was to implement Monte Carlo algorithm and Monte Carlo recrystallization. The figure (*Figure 1*) represents final graphical user interface of created program.



*Figure 1 GUI*

Elements of GUI are grouped and colored in sections to easier describe their purpose.

First part of interface merged in the figure is connected with Monte Carlo algorithm.

User could proceed with this algorithm to simulate grain growth processes. Before run algorithm is required to define dimensions of table represents microstructure. User define dimension by writing proper values to X,Y textboxes and confirming by “*Draw*” button. (*Figure 2*)



*Figure 2 Dimension texboxes*

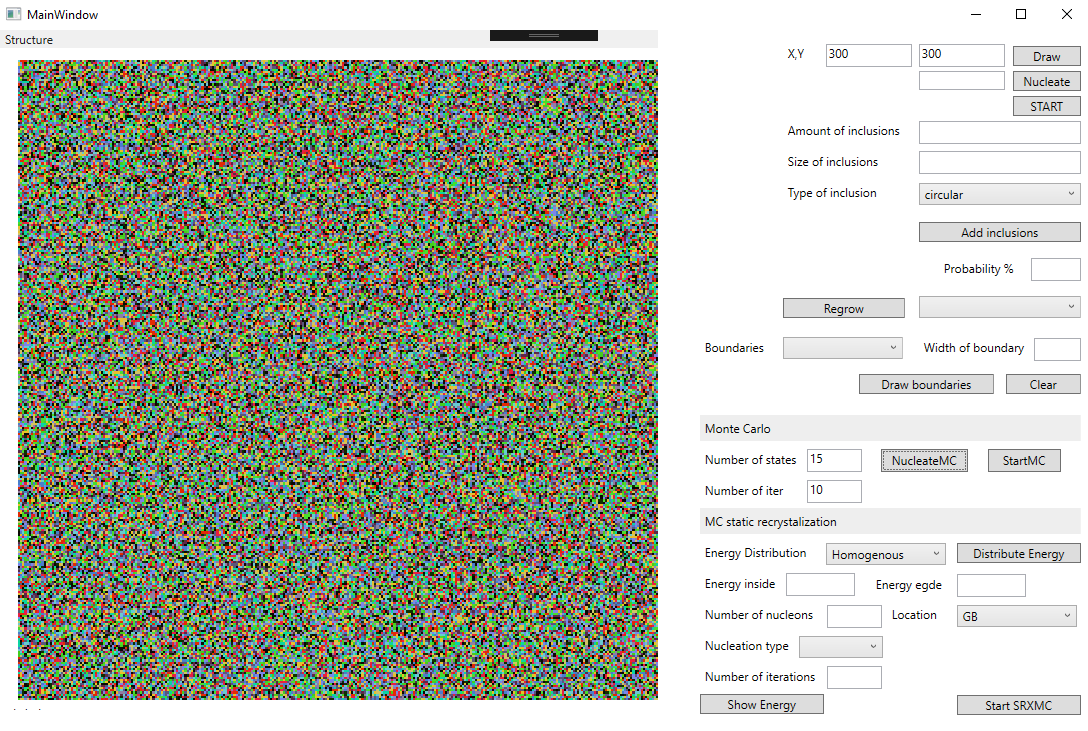
The next step is fill box with Number of States. This parameter describes how many possible states can be generated during Monte Carlo algorithm. Then is required to select how many iterations Monte Carlo algorithm should works. Despite 1 MC step is described by processing every grain in the microstructure in the program not every grain is processing. It is caused by optimization of algorithm. After clicking “*StartMC*” button MC algorithm runs.

In the second highlighted part of interface user could distribute energy in the grains. Selection “*Homogenous*” option makes that all grains get the same value of energy which is mean value of energy inside and energy on egdes. By clicking “*Heterogenous*” option energy is distributed with higher value on egdes and smaller value inside. Button “Show energy” allows showing in the image energy distribution of all grains.

**2. PROGRAM FUNCTIONS**

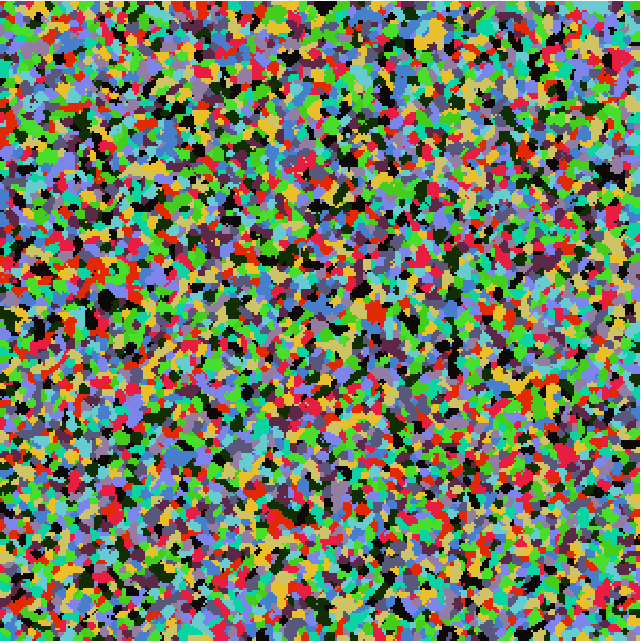
**MONTE CARLO ALGORITHM**

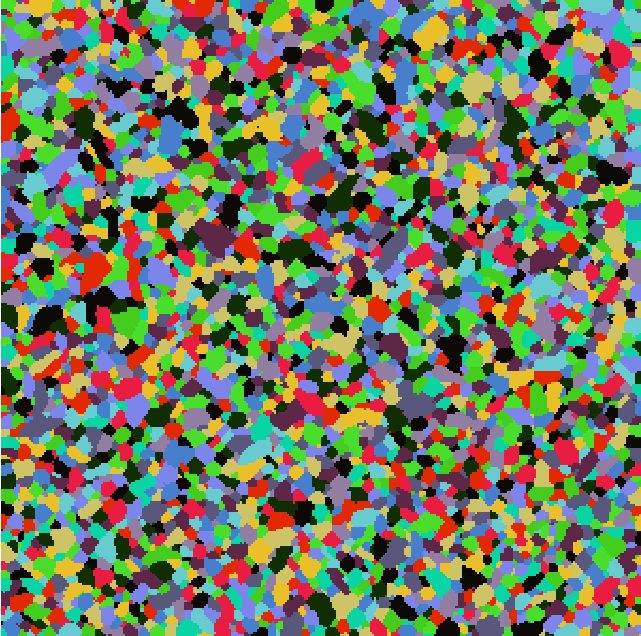
First function of the program is simulating Monte Carlo algorithm. By clicking “*Nucleate*” button all table is filled by grains with one of the states declared earlier by user in program. Figure shows example space generated by nucleating Monte Carlo grains. ( *Figure 3*).

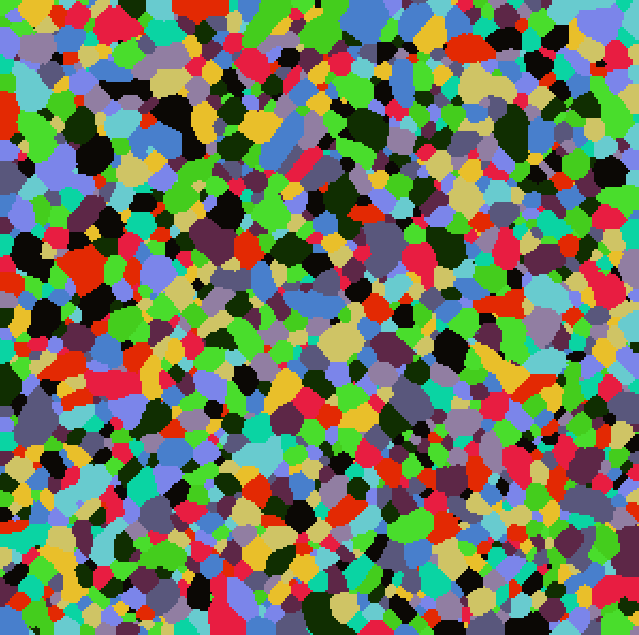


*Figure 3 Monte Carlo generated grains*

This space is generated from 15 possible states. In comparison to CA algorithm in this case at the beginning all grains all filled and during processing grains are combine into bigger structures. Next figure (*Figure 4*) presents result of algorithm after 5, 10 and 20 iterations.



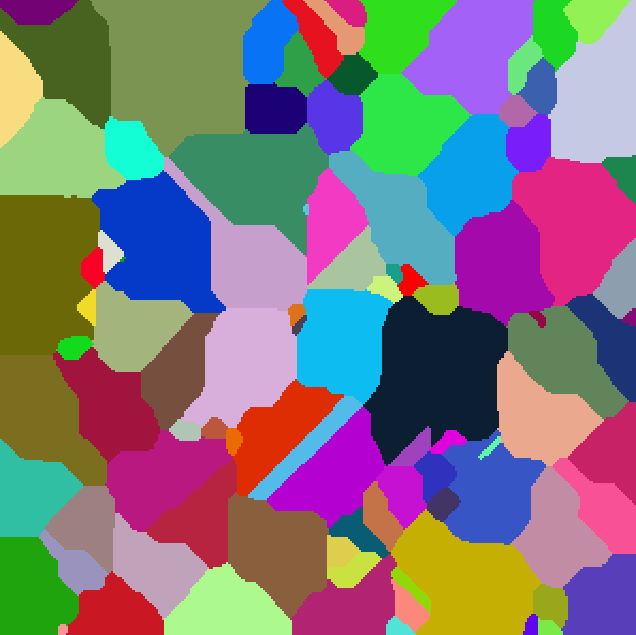


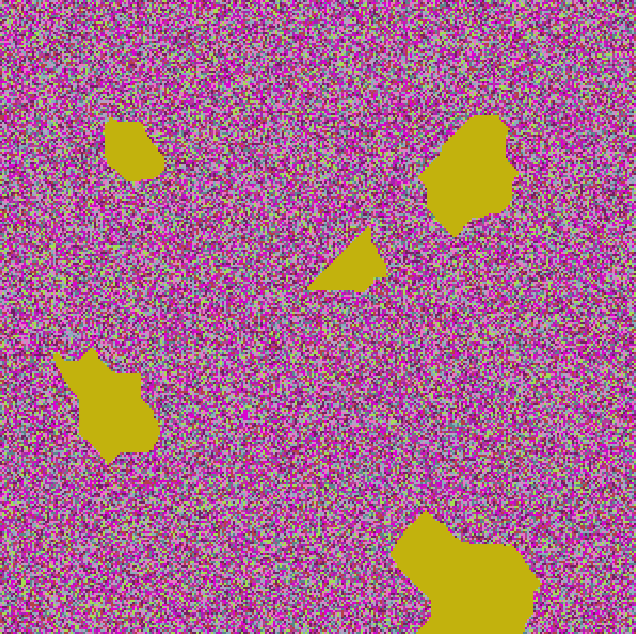


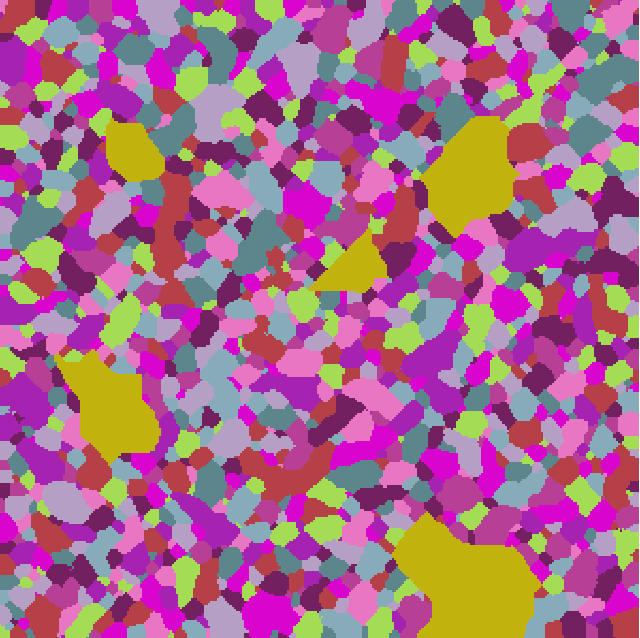
*Figure .4 5,10,20 iteration of Monte Carlo*

**CA->DP->MC**

Next function of program is generating grain growth by CA algorithm or MC algorithm and creating Dual Phase. After that user can fill empty fields by nucleating grains randomly or by nucleating grains everywhere like in MC algorithm. Then is possible to run both of algorithms to generate final structures. In this part 2 implemented algorithms of grain growing can be used together to reach more satisfying results. Figure 5 presents CA grain growth, making Dual Phase and then Monte Carlo grain growth.

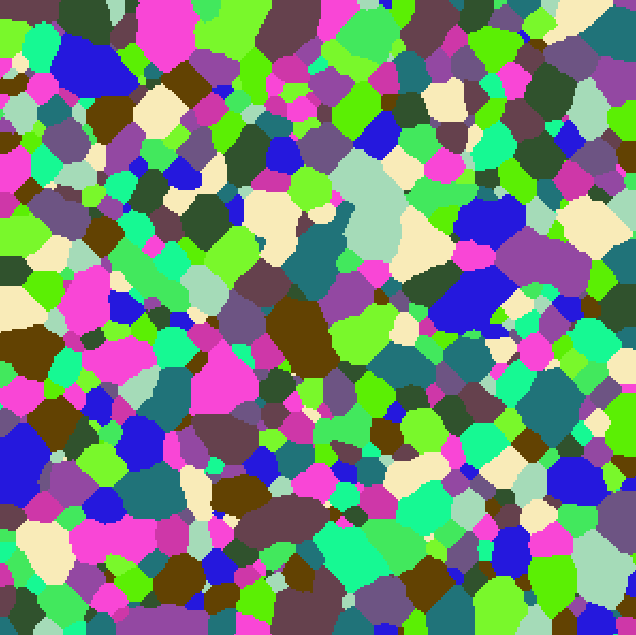


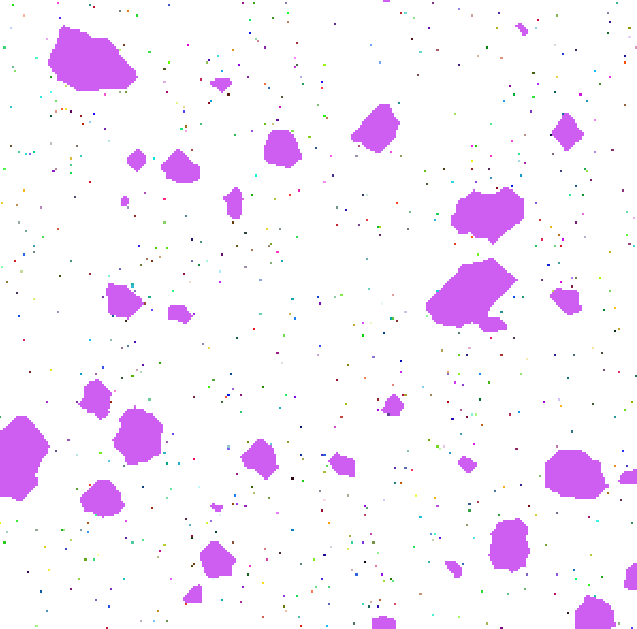


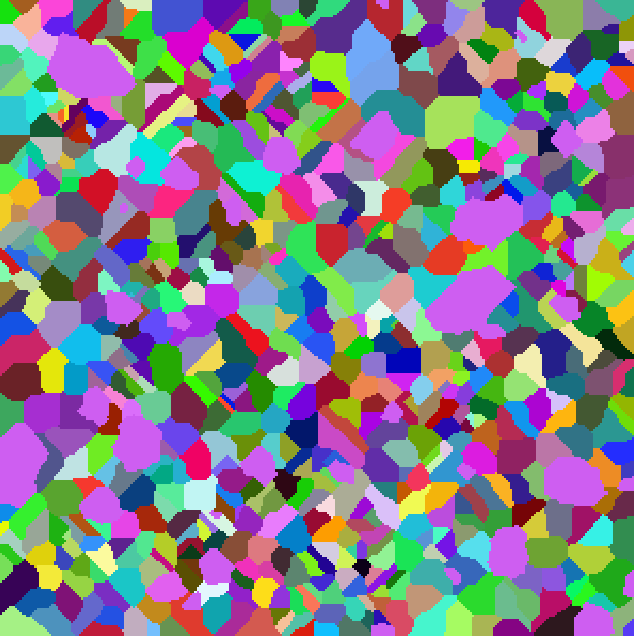


*Figure .5 CA->DP->MC*

Figure 6 presents the other situation. First MC grain growth, making Dual Phase and then CA algorithm.







*Figure .6 MC->DP->CA*

**ENERGY DISTRIBUTION**

In this part of program it is possible to distribute energy to grains after growth. There are 2 options of distributing. First option is homogenous which means that every grain has the same value of energy. This case is presented at figure 7, there is only one colour of energy for all grains.



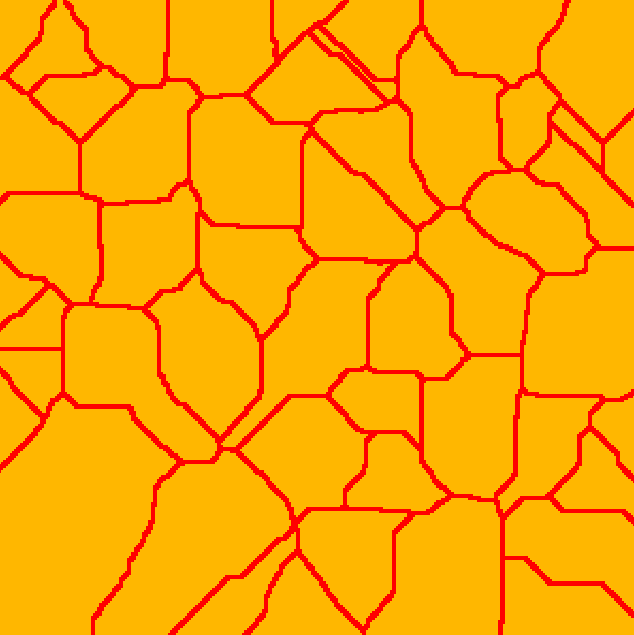


*Figure .7 Homogenous distribution of energy*

The second option is to heterogeneous distribution energy. That means that there is more energy on the edges of grains, what is presented by figure 8.

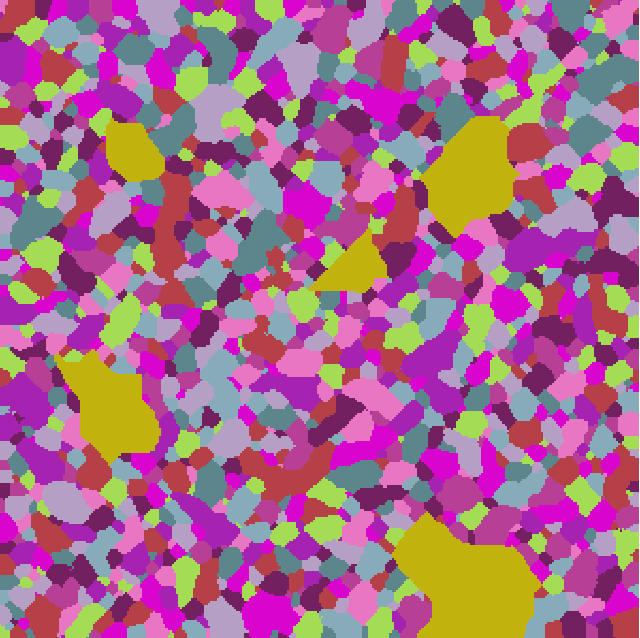
User can adjust parameters of energy distribution using second highlighted part on figure presented User Interface.





*Figure .8 Hetoerogenous distribution of energy*

**3. COMPARISON OF GENERATED MICROSTRUCTURES WITH REAL MICROSTRUCTURES AND CONCLUSIONS**



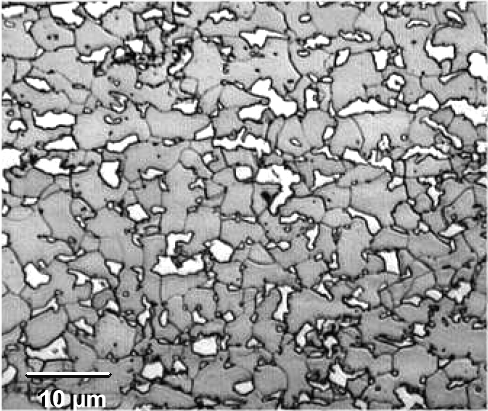


Figure 9. Comparison of microstructures

Real microstructure from and that generated by program are similar in general. Grains are different, their size, color and shape but by adjust parameters of algorithms using program it is possible to reach satisfactory results. Monte Carlo algorithm can improve Cellular Automata algorithm to generate more complex microstructures which characteristic is common to real microstructures.

Not all of the functionalities of the project are implemented yet. Part of the project responsible for recrystallization using modified Monte Carlo algorithm on not ready yet, but there is a good base to improve program, to implement additional functionalities. Already implemented algorithm could also helpful with some simple simulations of grain growth, to reach some basic result of microstructure.

Implementing this project there was chance get knowledge about Monte Carlo algorithm, to know which problem can be solved by this method. There was also good opportunity to improve your programming skills like: working with graphical interface or optimizing algorithms which is very important is such kind of a project.

Picture of real microstructure taken from:

https://www.tf.uni-kiel.de/matwis/amat/iss/kap\_9/advanced/t9\_2\_1b.html