Piotr Adryan

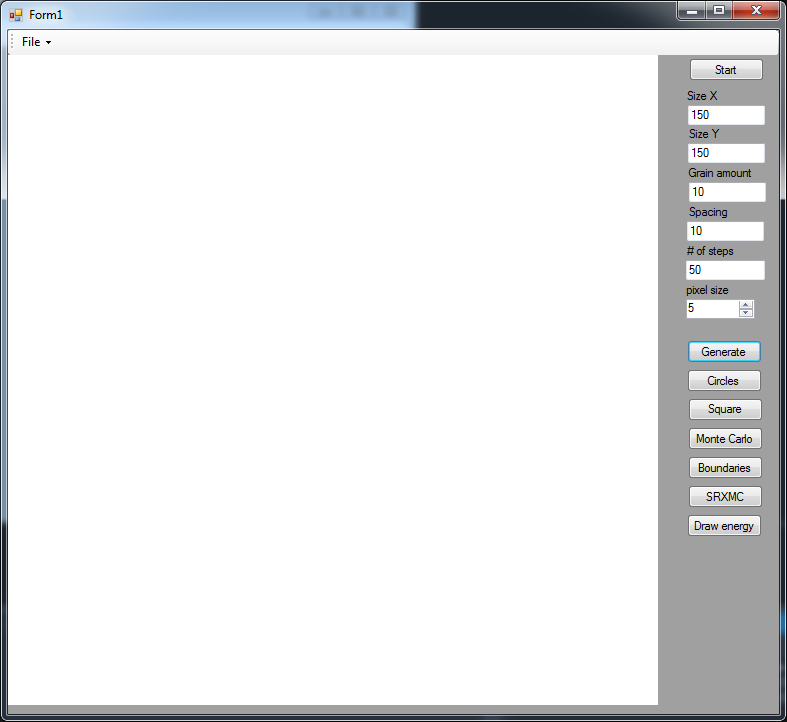
Informatyka Stosowana, IO

**Report 1**

The goal of the project: To create a simulation of Grain Growth using Monte Carlo algorithm

Part 1: The interface:

The options included in the interface(pic. 1):

* The size of the simulation space
* the number of generated grains
* Importing and exporting the current state of the simulation to .txt and .bmp files
* Selecting and deleting individual grains
* Selecting the number of steps of MC algorithm
* Simulating the recrystallization process on an existing microstructure
* Drawing the energy levels including recrystallized grains

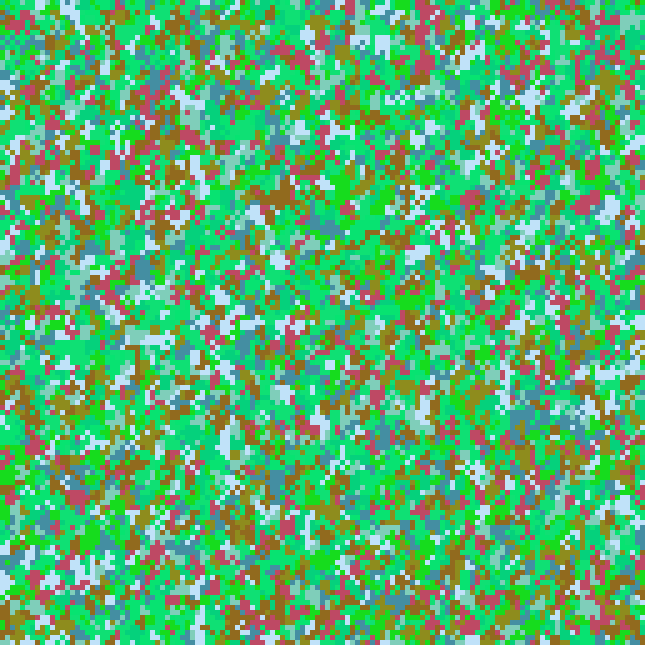
Picture 1

Task 1: Monte Carlo grain growth algorithm

In this task we implement the Monte Carlo algorithm, which works by choosing a point form a list of points that are on the boundary between grains, calculating it's energy, changing the Id of the point and checking the energy value again - the algorithm's goal is to decrease the energy in the test space.

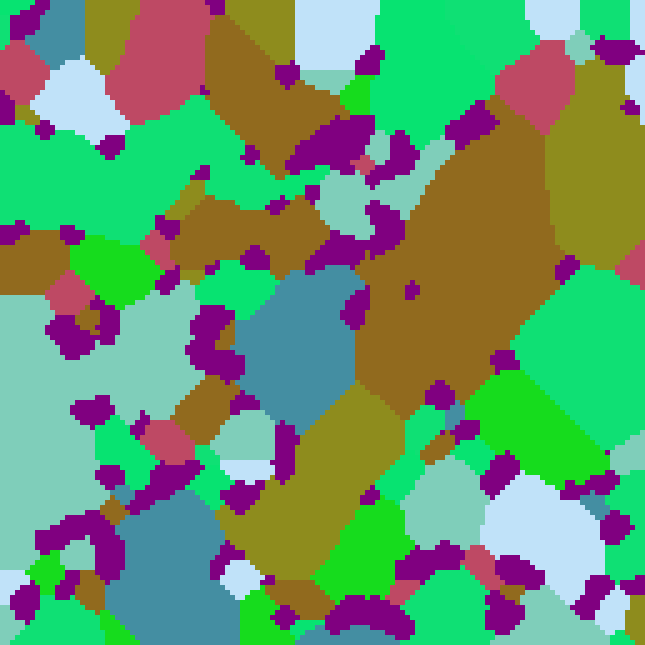
As such, if the new energy value isn't lower than the first one, the change of id is reverted and he next point on the list is checked. When all the points had a chance to change their energy one step of the MC algorithm is completed.

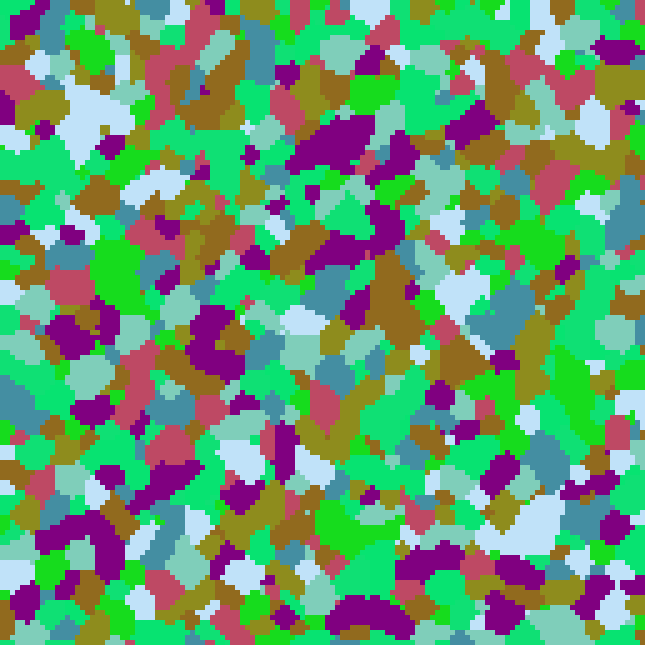
The algorithm can be used on a microstructure of Grown grains from previous project, but for this project the generation method consists of filling the entire space by inserting a random ID grain into each and every cell. A result of such generation with parameters of 10 grains and 150x150 space is shown on pic.2



Picture 2

Task 2: Dual phase microstructure:

This task in parallel to the task from first project has the goal of implementing the dual phase into the microstructure. In an existing microstructure you select a grain by clicking on it with a mouse and all grains with the selected ID will turn into Dual Phase grain, which is exempted from any changes that occur within the material. The Dual Phase grains are characterized by a purple color that is unique to those selected grains. The Shown pictures will show an example: in pic. 3 we can see the selected grain immediately after generation, and pic. 4 shows the same microstructure after 250 steps of MC algorithm have passed.

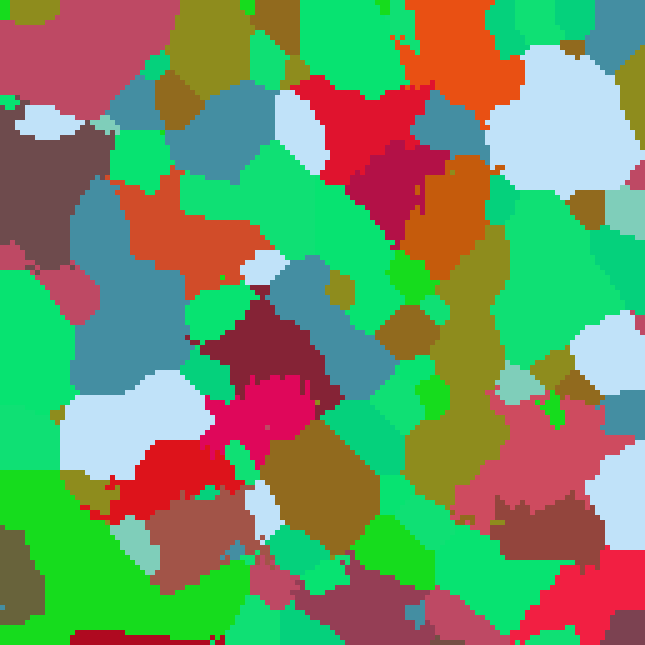


Picture 4

Picture 3

Task 3: MC static recrystallization algorithm

In this task we implement the algorithm for static recrystallization of microstructures. It happens as choosing a few cells which will contain new grain ID that is to be the recrystallized grain. Then we check every cell in the workspace if the checked cell has a recrystallized neighbor. if that happens we calculate it's energy, but we add to that an additional amount of energy. We then change the id to the recrystallized grain one and check energy again. If the energy before is higher then we accept the change and go to the next cell. In the meantime the normal rules of MC algorithm take place. The initial growth of recrystallized grains is showed in pic. 5, where the red grains are the recrystallized ones.



Picture 5

Task 4: Display of Energy distribution

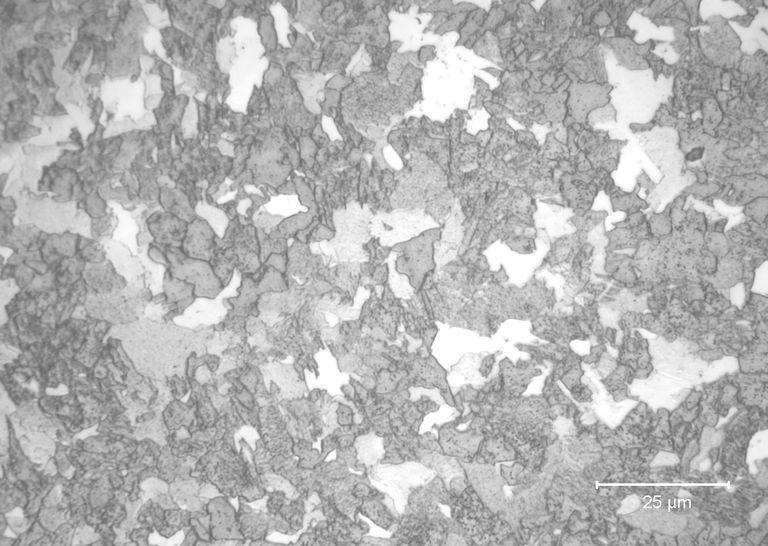
In this task a functionality is implemented that instead of showing each grain id in a different color the simulation is shown with different color for each energy level, which shows as 3 distinct possibilities: the low energy color that happenbs inside of a grain, the high energy of rgrain boundaries that by their nature have higher energy levels which are added to during energy distribution, and the recrystallized grains, which show as yet another color due to spending all their energy on the recrystallization process. In this simulation depicted in pic. 6, the inside of a grain is shown as blue, the grain boundaries are shown as green, and the recrystallized grains are shown as red.

Conclusions:

The metallurgic industry constantly faces problem of being forced to do complicated and costly test on metallic samples, which can be simplified thanks to the usage of the Monte Carlo algorithm. The algorithm can be useful due to it's good balance between the speed of the simulation, and the faithfulness to how the real life materials behave. In those terms it is much better at generating realistic looking samples than previous Cellular automata algorithm.

The algorithm can be additionally optimized using several methods, such as the one used in this code where only the grain boundaries are checked, since no energy level changes should occur inside of a grain., and instead of normal timer the whole program could be paralelly executed using several threads, which would significantly increase the speed of simulation for larger sizes of the workspace.

Shown below are two pictures of microstructures, with pic.7 showing a simulated microstructure using MC algorithm, and pic.8 being a photo of a real metal microstructure.



Picture 8

Picture 7