Michał Prokop

**Multiscale Modelling – 2nd report**

1. **Technology:**
2. **Programming language (C++):**

C++ is a [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language" \o "General-purpose programming language). It has [imperative](https://en.wikipedia.org/wiki/Imperative_programming" \o "Imperative programming), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming" \o "Object-oriented programming) and generic programming features, while also providing facilities for [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language" \o "Low-level programming language) [memory](https://en.wikipedia.org/wiki/Memory_(computing)" \o "Memory (computing)) manipulation.

It was designed with a bias toward [system programming](https://en.wikipedia.org/wiki/System_programming) and [embedded](https://en.wikipedia.org/wiki/Embedded_software" \o "Embedded software), resource-constrained and large systems, with [performance](https://en.wikipedia.org/wiki/Performance_(software)), efficiency and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including [desktop applications](https://en.wikipedia.org/wiki/Application_software), servers (e.g. [e-commerce](https://en.wikipedia.org/wiki/E-commerce), [Web search](https://en.wikipedia.org/wiki/Web_search_engine) or [SQL](https://en.wikipedia.org/wiki/SQL) servers), and performance-critical applications (e.g. [telephone switches](https://en.wikipedia.org/wiki/Telephone_switches" \o "Telephone switches) or [space probes](https://en.wikipedia.org/wiki/Space_probes" \o "Space probes)). C++ is a [compiled language](https://en.wikipedia.org/wiki/Compiled_language" \o "Compiled language), with implementations of it available on many platforms. Many vendors provide [C++ compilers](https://en.wikipedia.org/wiki/List_of_compilers#C.2B.2B_compilers), including the [Free Software Foundation](https://en.wikipedia.org/wiki/Free_Software_Foundation" \o "), [Microsoft](https://en.wikipedia.org/wiki/Microsoft), [Intel](https://en.wikipedia.org/wiki/Intel), and [IBM](https://en.wikipedia.org/wiki/IBM).

C++ is standardized by the [International Organization for Standardization](https://en.wikipedia.org/wiki/International_Organization_for_Standardization) (ISO), with the latest standard version ratified and published by ISO in December 2017 as [*ISO/IEC 14882*](https://en.wikipedia.org/wiki/C%2B%2B#Standardization)*:2017* (informally known as [C++17](https://en.wikipedia.org/wiki/C%2B%2B17)). The C++ programming language was initially standardized in 1998 as *ISO/IEC 14882:1998*, which was then amended by the [C++03](https://en.wikipedia.org/wiki/C%2B%2B03), [C++11](https://en.wikipedia.org/wiki/C%2B%2B11) and [C++14](https://en.wikipedia.org/wiki/C%2B%2B14) standards. The current [C++17](https://en.wikipedia.org/wiki/C%2B%2B17) standard supersedes these with [new features](https://en.wikipedia.org/wiki/C%2B%2B17" \o "C++17) and an enlarged [standard library](https://en.wikipedia.org/wiki/C%2B%2B#Standard_library). Before the initial standardization in 1998, C++ was developed by [Bjarne Stroustrup](https://en.wikipedia.org/wiki/Bjarne_Stroustrup" \o "Bjarne Stroustrup) at [Bell Labs](https://en.wikipedia.org/wiki/Bell_Labs) since 1979, as an extension of the [C language](https://en.wikipedia.org/wiki/C_(programming_language)) as he wanted an efficient and flexible language similar to C, which also provided high-level features for program organization. [C++20](https://en.wikipedia.org/wiki/C%2B%2B20) is the next planned standard thereafter.

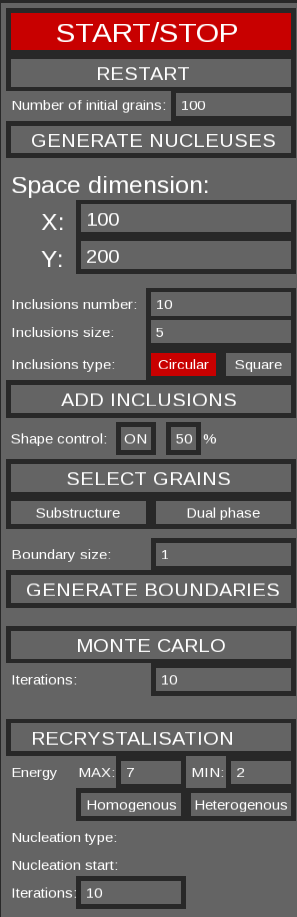
Many other programming languages have been influenced by C++, including [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [D](https://en.wikipedia.org/wiki/D_(programming_language)), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), and newer versions of C.

1. **Graphic Library:**

Simple and Fast Multimedia Library (SFML) is a cross-platform software development [library](https://en.wikipedia.org/wiki/Library_(computing)" \o "Library (computing)) designed to provide a simple [application programming interface](https://en.wikipedia.org/wiki/Application_programming_interface" \o "Application programming interface)(API) to various multimedia components in computers. It is written in [C++](https://en.wikipedia.org/wiki/C%2B%2B) with [bindings](https://en.wikipedia.org/wiki/Language_binding" \o "Language binding) available for [C](https://en.wikipedia.org/wiki/C_(programming_language)), [Crystal](https://en.wikipedia.org/wiki/Crystal_(programming_language)), [D](https://en.wikipedia.org/wiki/D_(programming_language)), [Euphoria](https://en.wikipedia.org/wiki/Euphoria_(programming_language)), [Go](https://en.wikipedia.org/wiki/Go_(programming_language)), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [Julia](https://en.wikipedia.org/wiki/Julia_(programming_language)), [.NET](https://en.wikipedia.org/wiki/.NET_Framework), [Nim](https://en.wikipedia.org/wiki/Nim_(programming_language)), [OCaml](https://en.wikipedia.org/wiki/OCaml" \o "OCaml), [Python](https://en.wikipedia.org/wiki/Python_(programming_language)" \o "Python (programming language)), [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)), and [Rust](https://en.wikipedia.org/wiki/Rust_(programming_language)" \o "Rust (programming language)). Experimental mobile ports were made available for [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) and [iOS](https://en.wikipedia.org/wiki/IOS" \o "IOS) with the release of SFML 2.2.

SFML handles creating and input to [windows](https://en.wikipedia.org/wiki/Window_(computing)" \o "Window (computing)), and creating and managing [OpenGL](https://en.wikipedia.org/wiki/OpenGL" \o "OpenGL) contexts. It also provides a graphics module for simple [hardware acceleration](https://en.wikipedia.org/wiki/Hardware_acceleration) of [2D computer graphics](https://en.wikipedia.org/wiki/2D_computer_graphics) which includes text rendering using [FreeType](https://en.wikipedia.org/wiki/FreeType" \o "FreeType), an audio module that uses [OpenAL](https://en.wikipedia.org/wiki/OpenAL" \o "OpenAL) and a networking module for basic [Transmission Control Protocol](https://en.wikipedia.org/wiki/Transmission_Control_Protocol" \o "Transmission Control Protocol) (TCP) and [User Datagram Protocol](https://en.wikipedia.org/wiki/User_Datagram_Protocol" \o "User Datagram Protocol) (UDP) communication.

SFML is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software" \o ") provided under the terms of the [zlib/png license](https://en.wikipedia.org/wiki/Zlib_License" \o "Zlib License). It is available on [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://en.wikipedia.org/wiki/Linux), [macOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS) and [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD" \o "FreeBSD). The first version v1.0 was released on 9 August 2007, the latest version v2.5.1 was released on 15 Oct 2018.

1. **GUI:**

- START/STOP button – starts or stops simulation

- RESTART button – clears space, restarts simulation

- Number of initial grains which will be growing in simulation

- button which generates nucleuses in space

- text fields for setting space dimension

- text fields for setting inclusions number, size

- toggle button for setting inclusions type

- button which generates inclusions in space

- button for adding „Shape Control” and probabilisty of Rule 4

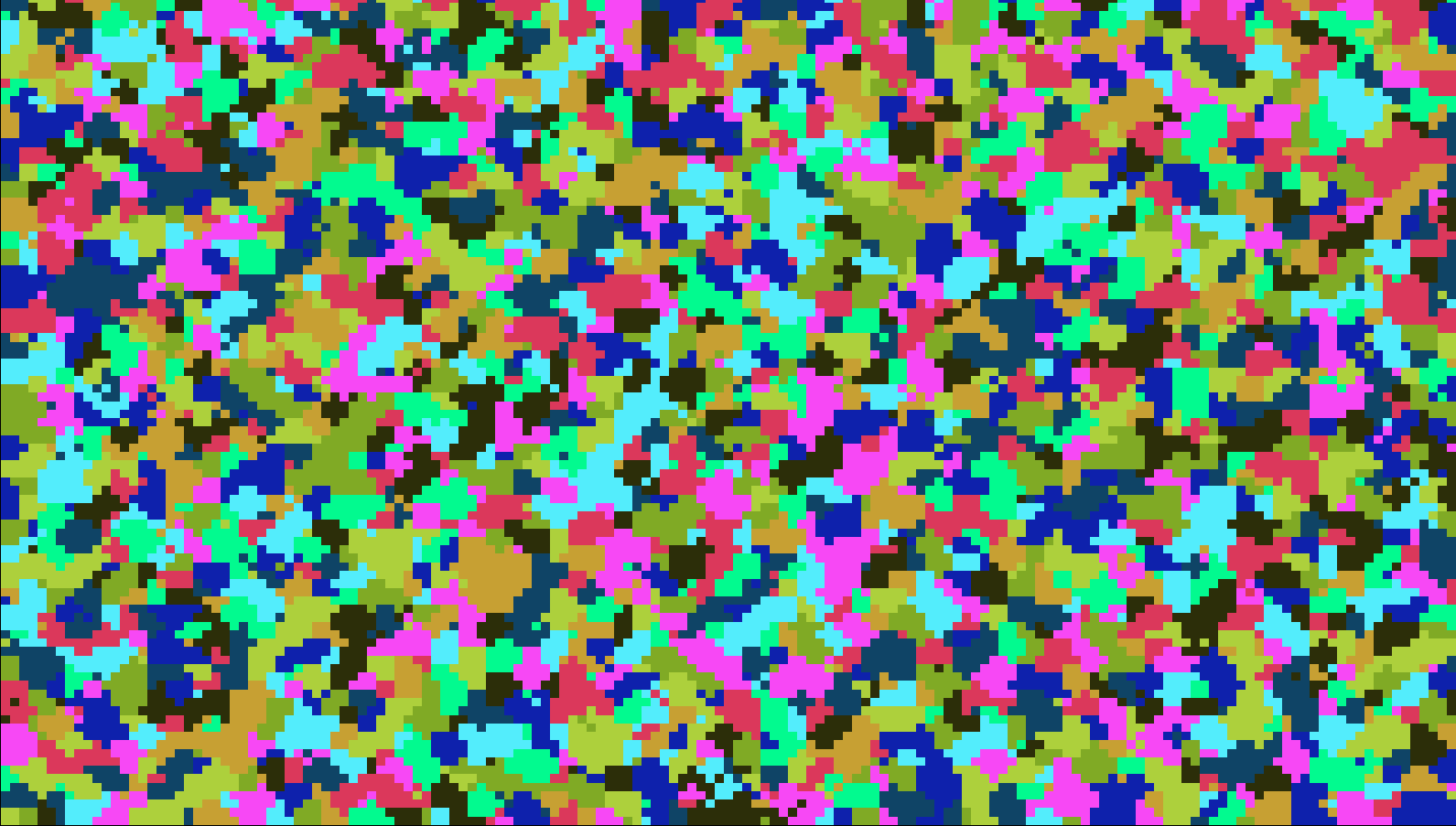
- grains selection and substructure or dual phase generation

- button for generating boundaries

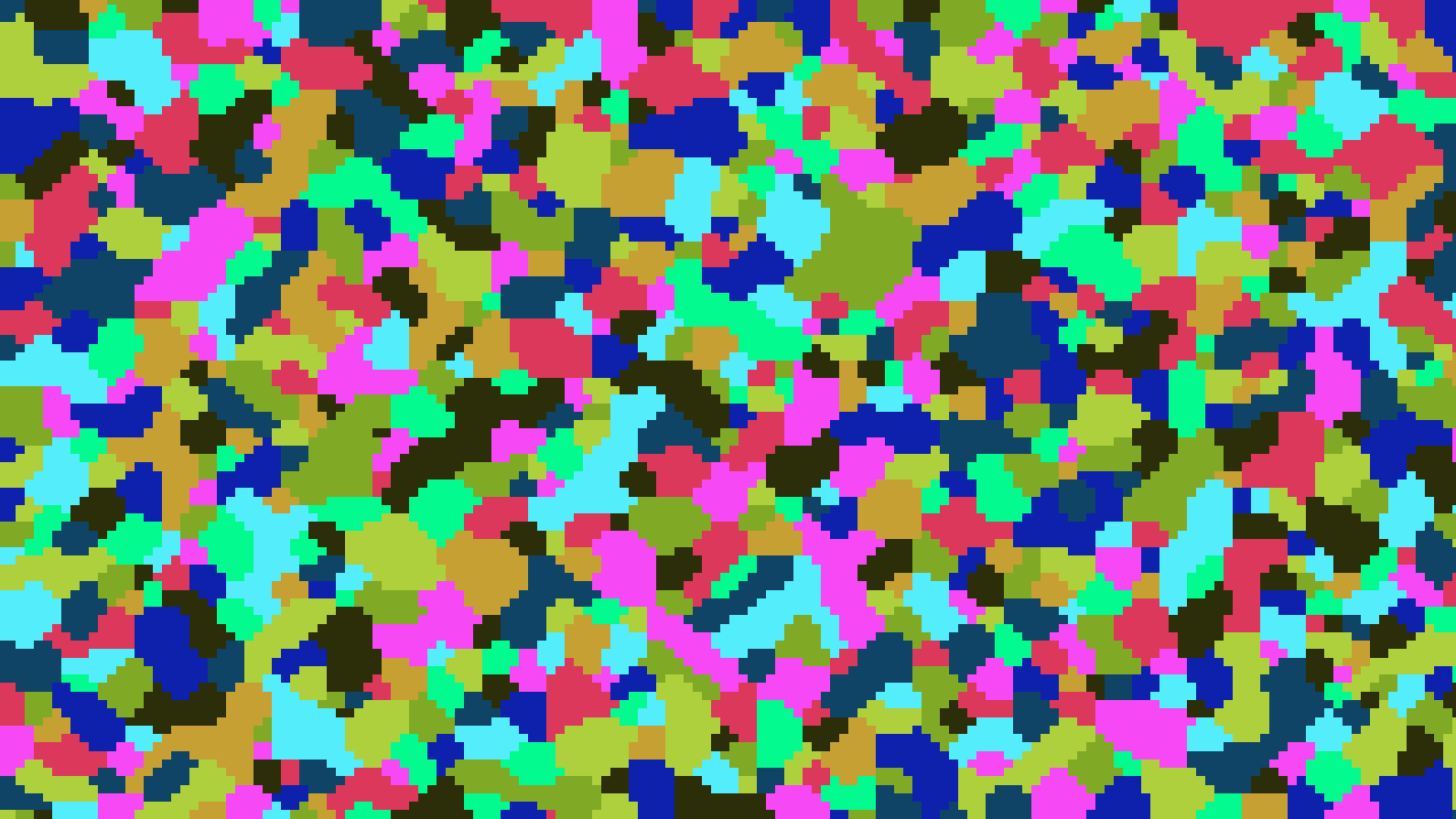
**Fig 1.** GUI

1. **Results:**
2. **Monte Carlo**

We can generate initial map using Monte Carlo algorythm



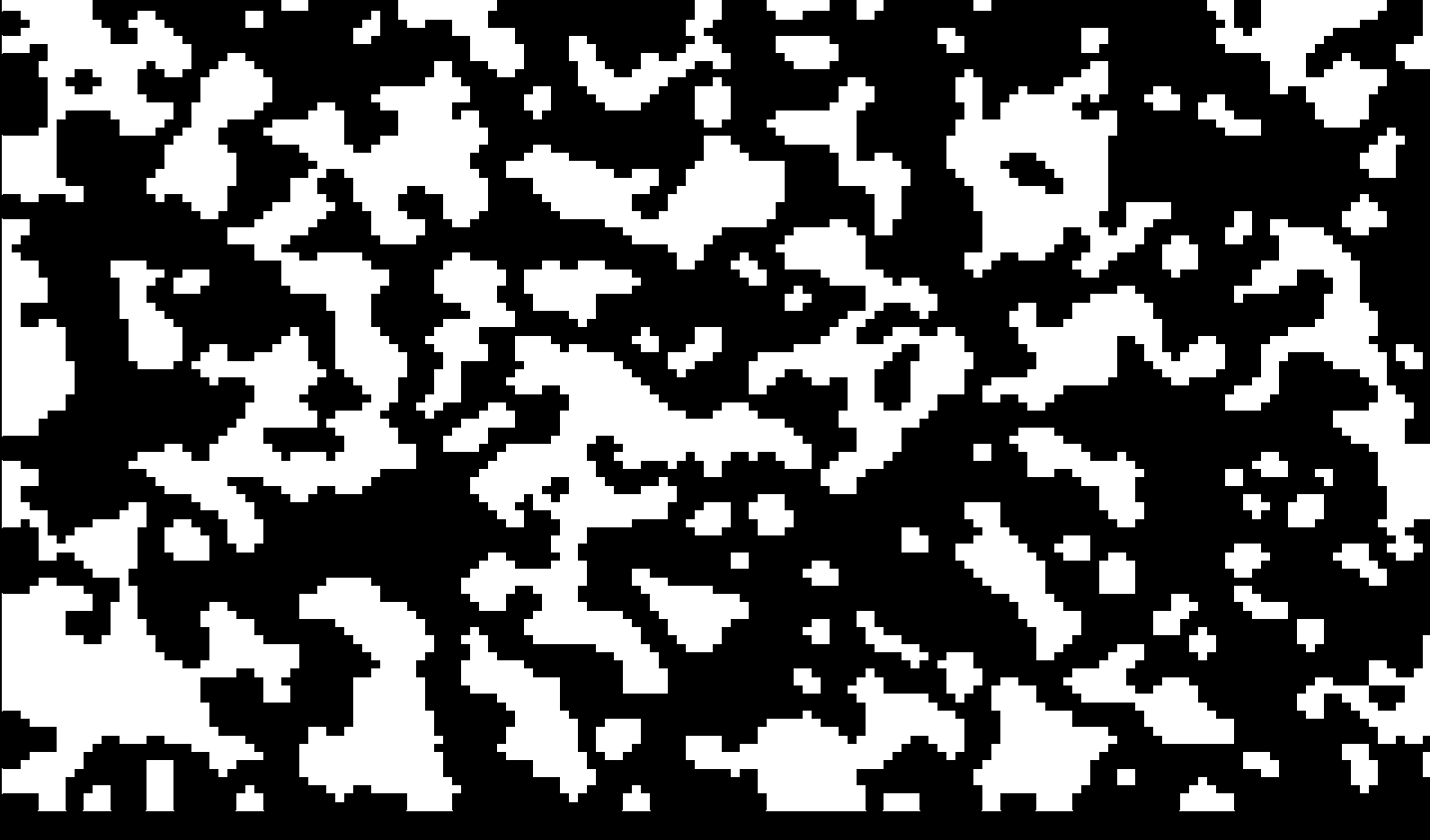
**Fig. 2.a** Initial space (100X200) – after 1 iteration



**Fig. 2.b** Space (100X200) – after 15 iterations

1. **MC 🡪 Dual phase**

We can generate dual phase steel from Mone Carlo simulation



**Fig. 3.a.** Space (100X200) after adding „Dual Phase” option

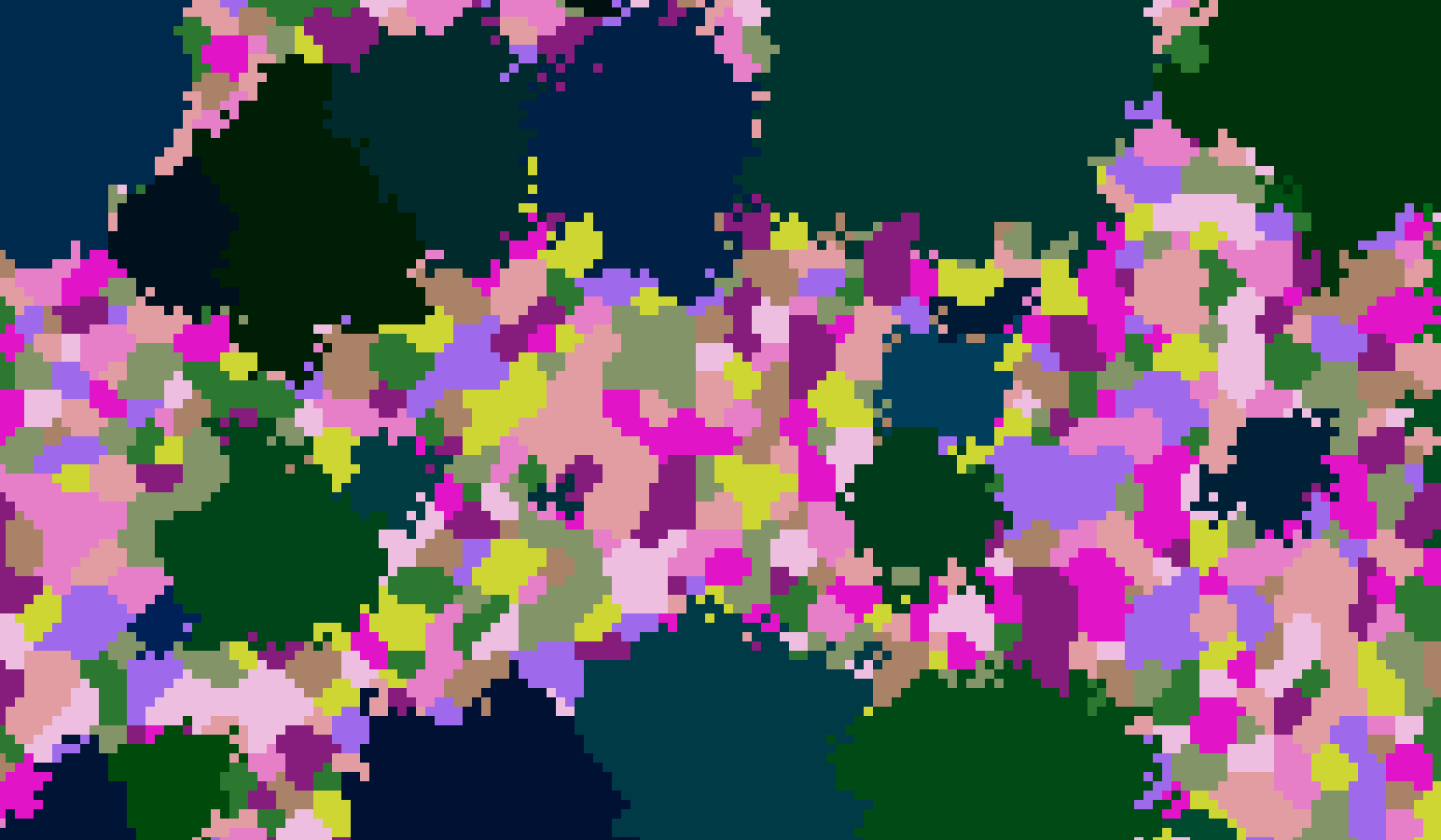
1. **SRX**



**Fig. 4.a** Space (100X200) – after 15 iteration with Monte Carlo

****

**Fig. 4.b** Space (100X200) – heterogenous energy distribution



**Fig. 4.c** Space (100X200) – after 15 iterations of SRX

1. **Conclusions:**

This program is a simple way of simulating real grain growth. It can be efficient and computed parallely. Acheived structures resemble microstructures of real steal and can be useful for researchers. After introducing additional rules of neighbourhood and growth we could improve the level of exactness so it would be more reliable. Nevertheless, it is a good start.

1. **Literature**

[1] Chatterjee S., Murugananth M., Bhadeshia H.k.D.H., δ-TRIP Steel. Materials Science and Technology, Issue 23/2007, s. 817-823.

[2] Lis A.k., Gajda B., Modelling of the DP and TRIP microstructure in the CMnAlSi automotive steel, Jurnal of Achievements in Materials and Manufacturing Engineering, Vol. 15 Issue 1-2/2006, s. 127-134.

[3] Yi H.L., Lee k.Y., Bhadeshi H.k.D.H., Stabilisation of ferrite in hot rolled δ-TRIP steel, Materials Science and Technology, Vol. 27, No. 2/2011/525-529.

[4] Senkara J., Współczesne stale karoseryjne dla przemysłu motoryzacyjnego i wytyczne technologiczne ich zgrzewania, Przegląd Spawalnictwa nr 11/2009, s. 3-7.

[5] Hofmann H., Mattissen D., Schaumann T. W., Advanced cold rolled steels for automotive applications, Steel Research International Issue 1/2009, s. 22-28.

[6] Prospekt AHSS, Material data sheet. Cold-rolled quenched & tempered steel, boronalloyed, for cold forming and subsequent hot forming, (suitable for press hardening), Salzgitter Flachstahl 11-12/2005.