Uniwersytet Rzeszowski

Piotr Szpila, Adrian Wilk

Informatyka, studia stacjonarne I stopnia, semestr 4

Projekt pt.:

Wizualizacja przeszkód dla robota EV3.

Rzeszów, 2017 r.

Spis treści

1. Opis ogólny:

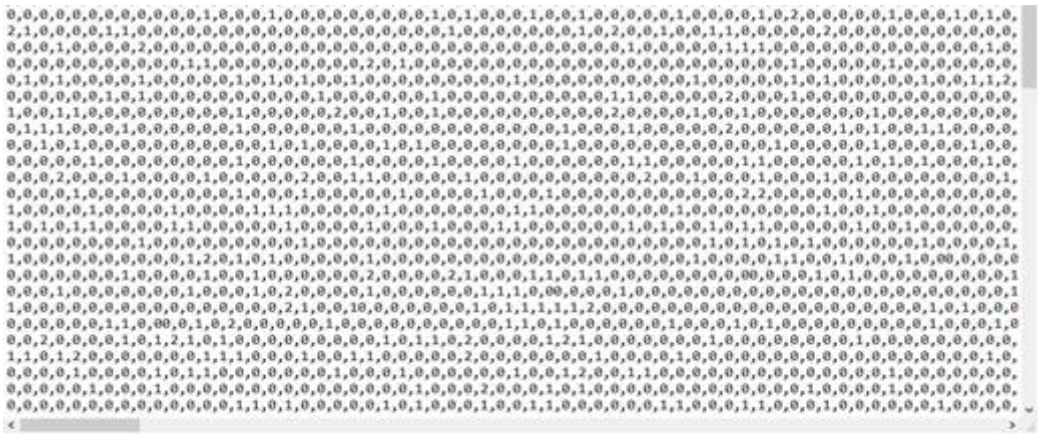
Nazwa programu wywodzi się od podstawowego problemu, „Obstacle Visualization for EV3 Robot”. Projekt ma za zadanie zwizualizować przeszkody dla robota EV3 za pomocą wczytanej macierzy z pliku *.matrix i wyświetlenie jej na panelu użytkownika.

2. Format danych:

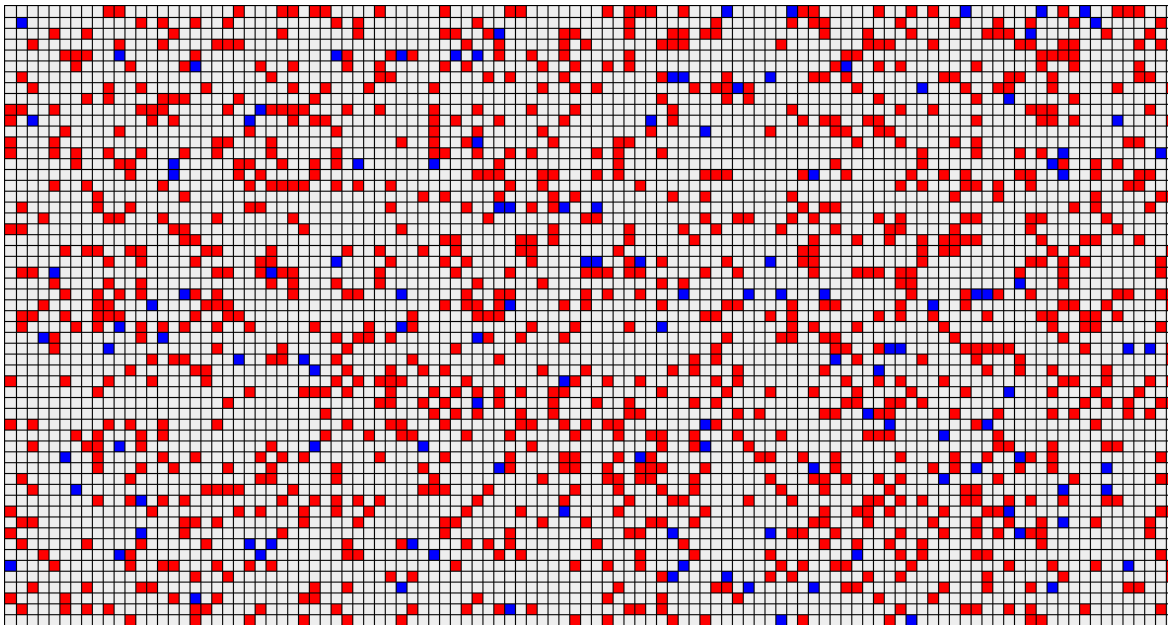
Formatem przechowującym dane macierzy jest format *.matrix. Plik może zawierać w sobie cyfry takie jak:

- **0-** brak przeszkody, kolor biały,
- **1-** przeszkoda, kolor czerwony,
- **2-** kolor zapasowy (w przyszłości będzie służyć do wyznaczania ścieżki robota), niebieski.

Wyżej wymienione cyfry oddzielamy przecinkami. Z kolei poniżej możemy zobaczyć jak powinien wyglądać plik:



Oraz wizualizacja w programie:



3. Odwzorowanie Macierzy w rzeczywistości:

Macierz przedstawiona graficznie w programie odpowiada rzeczywistości w skali:

1 komórka = 10 x 10 [cm]

Program poprzez wyświetlanie żółtych linijek wzdłuż panelu macierzy oraz przycisku siatki pomocniczej pozwala na dokładne prace nad zadaną macierzą.

4. Export danych:

Program pozwala nam na eksport danych do plików takich jak:

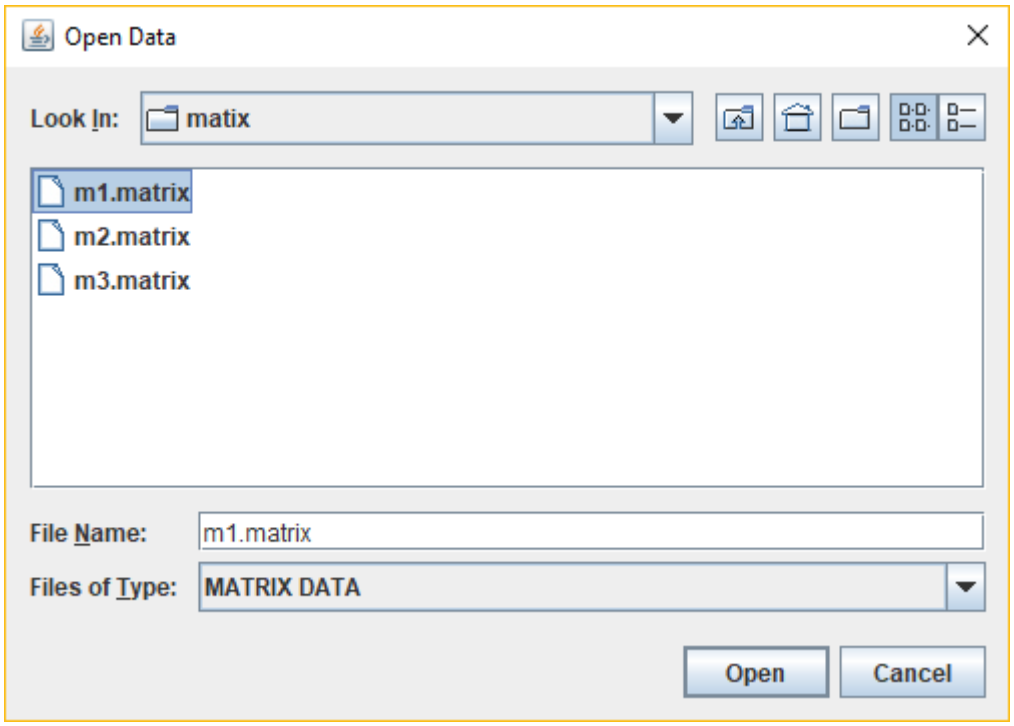
- JPG- Jest to najpopularniejszy format używany do zapisu zdjęć w celu publikacji ich w Internecie i wyświetlania na monitorze. Zdjęcie zapisane w JPEG zapisuje obraz do 16 milionów kolorów, jest kompatybilny z większością systemów operacyjnych (Mac, PC, Linux).

- PNG- Jest to format zdjęć stworzony z myślą o publikowaniu materiałów w Internecie. Obecnie występuje w dwóch wersjach różniących się ilością przechowywanych informacji. PNG-8 zapisuje obraz maksymalnie w 256 kolorach. PNG można jedynie zapisać obrazy w RGB, nie wspiera on kolorów CMYK przez co nie nadaje się do przygotowywania plików do druku. Format PNG zalecany jest przez W3C jako główny format grafiki dla sieci www.

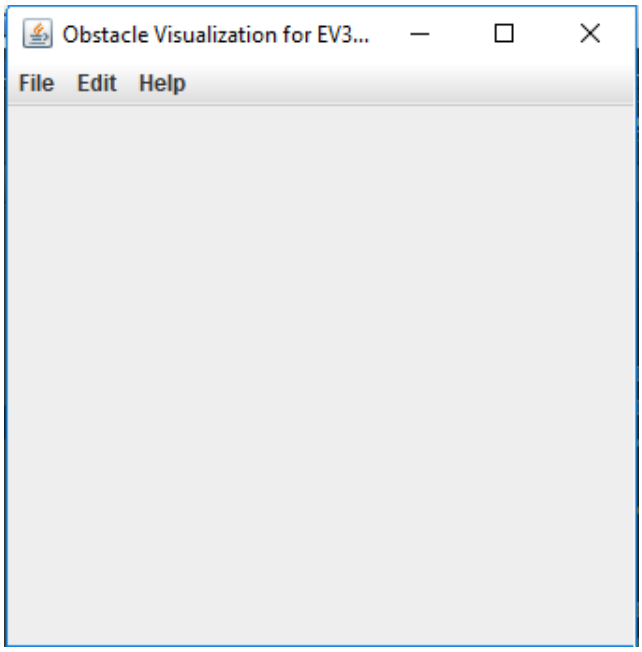
- BMP- Jest to format pliku z grafiką bitmapową. Pliki BMP są zdjęciami zoptymalizowanymi dla systemu operacyjnego Windows. Zdjęcia BMP są duże i nieskompresowane, lecz bardzo bogate w kolory, wysokiej jakości, kompatybilne ze wszystkim wersjami SO Windows i programami.

5. Zapoznanie ze środowiskiem programu:

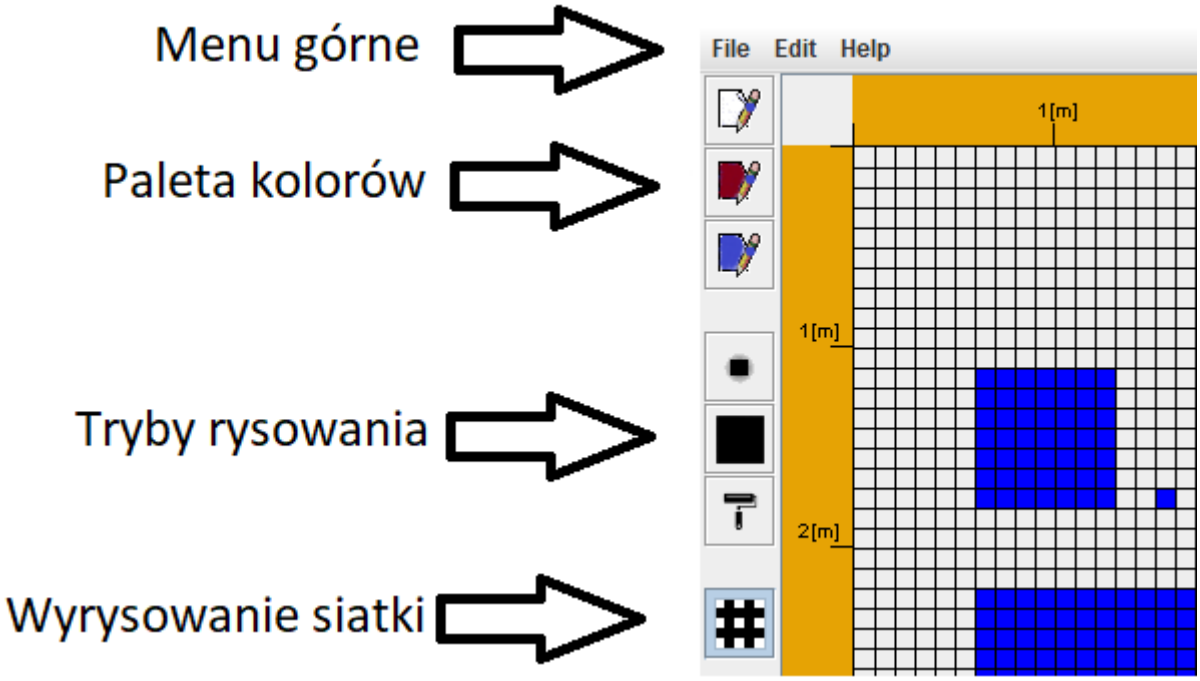
Aby korzystać z programu należy otworzyć plik „RobotLego.jar”. Następnie wyskoczy okienko do wybrania pliku z macierzą. Wczytujemy plik, na którym chcemy pracować:



Jeśli po wyświetleniu początkowego okna zamkniemy je, to program będzie mieć częściową funkcjonalność oraz będzie wyglądać następująco:



Kolejny obraz przedstawia panel użytkownika, na którym należy pracować, aby rozwiązać problem. Aplikacja zawiera dwa paski, menu boczne oraz menu górne.

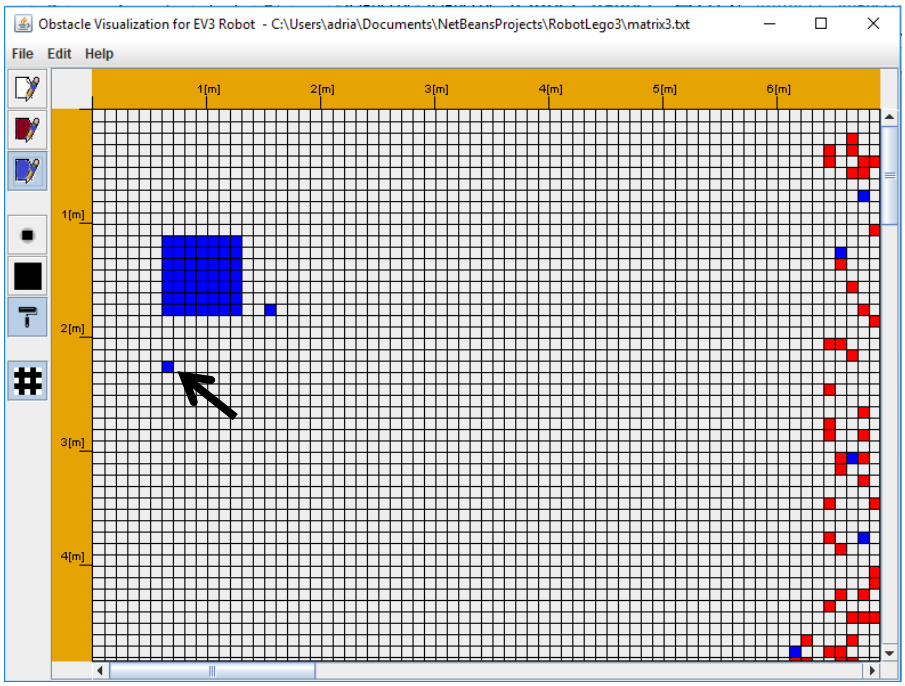


Menu boczne zawiera:

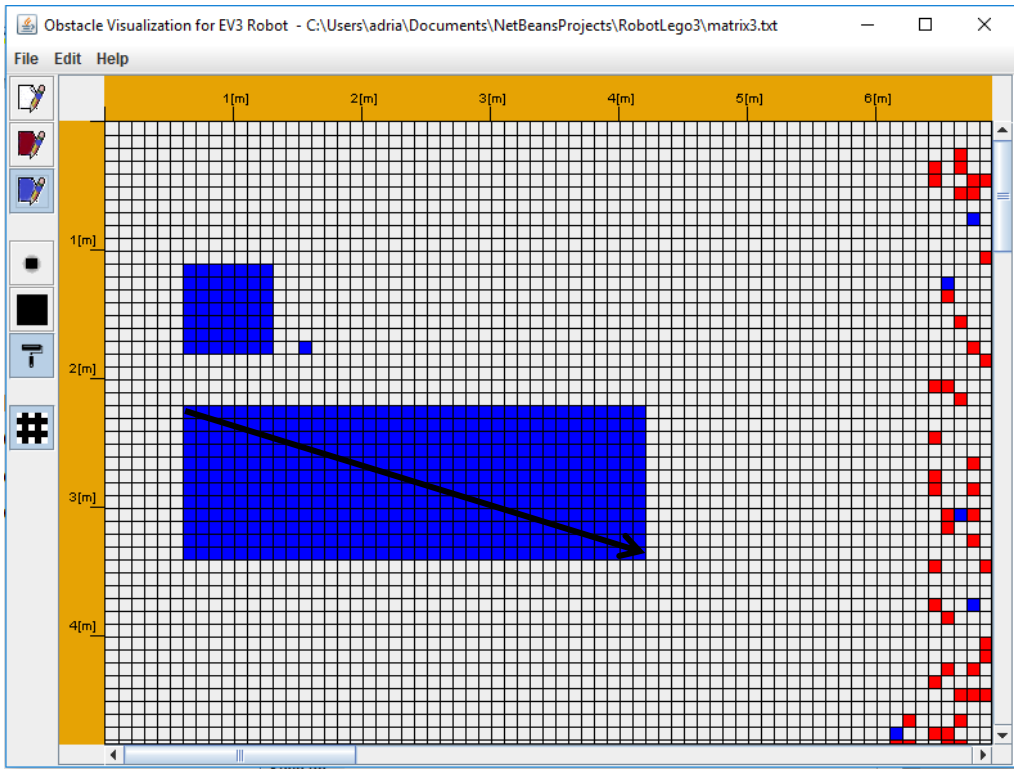
- ❖ paletę kolorów:
  - biały,
  - czerwony,
  - niebieski
- ❖ tryby rysowania:
  - mały kwadrat,
  - duży kwadrat,
  - tryb pędzla
- ❖ przycisk służący do wyrysowania siatki macierzy.

Aby narysować mapę wybieramy kolor, którym chcemy operować, a następnie tryb rysowania. Na niżej umieszczonym obrazie wybrany został tryb „pędzla” oraz została wyrysowana siatka macierzy. Aby narysować nim figurę należy wybrać punkt początkowy. Robimy to za pomocą wciśnięcia klawisza myszki.

Jeśli chcemy aby obraz został wyeksportowany z siatką należy zaaplikować przycisk siatki przed eksportem.

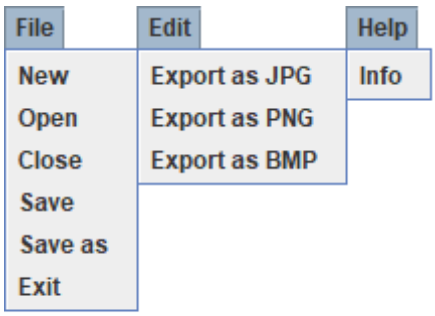


Po czym przeciągamy myszkę w wybranym kierunku.

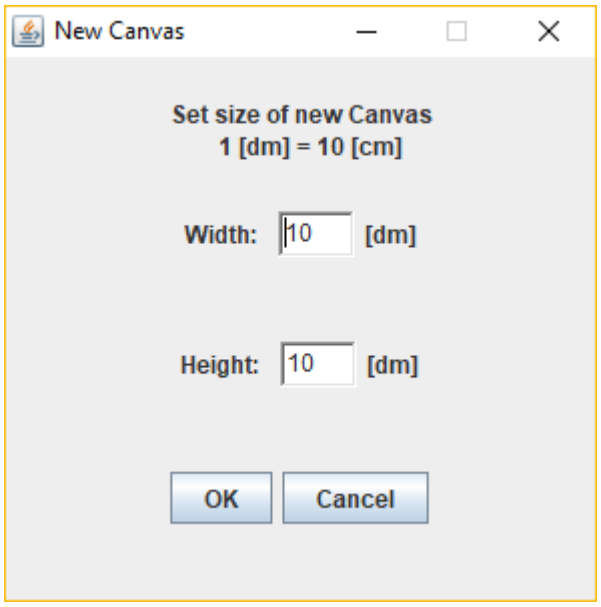


Pasek górny menu zawiera trzy opcje, takie jak:

- File:
  - New,
  - Open,
  - Close,
  - Save,
  - Save As,
  - Exit
- Edit zawiera:
  - *.jpg,
  - *.png,
  - *bmp
- Help zawiera informację dotyczącą programu.



Przycisk File -> New Pozwala na wygenerowanej nowej macierzy o zadanym rozmiarze. Dane wprowadzamy w nowym oknie, w jednostce decymetry. Gdzie 1[dm] = 10 [cm].

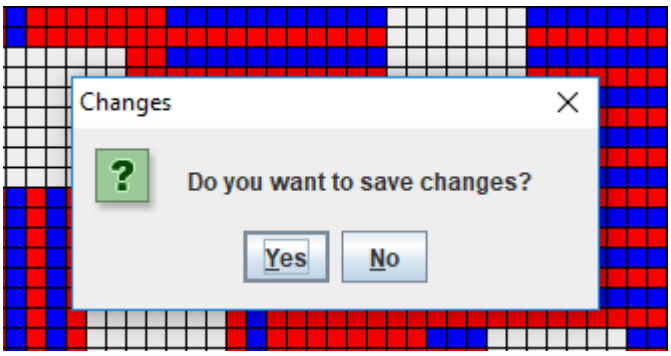


Po wygenerowaniu nowej macierzy, program pozwala jedynie na zapis przyciskiem File -> Save As.

Przycisk File -> Save zapisuje do pliku, z którego została otworzona dana macierz.

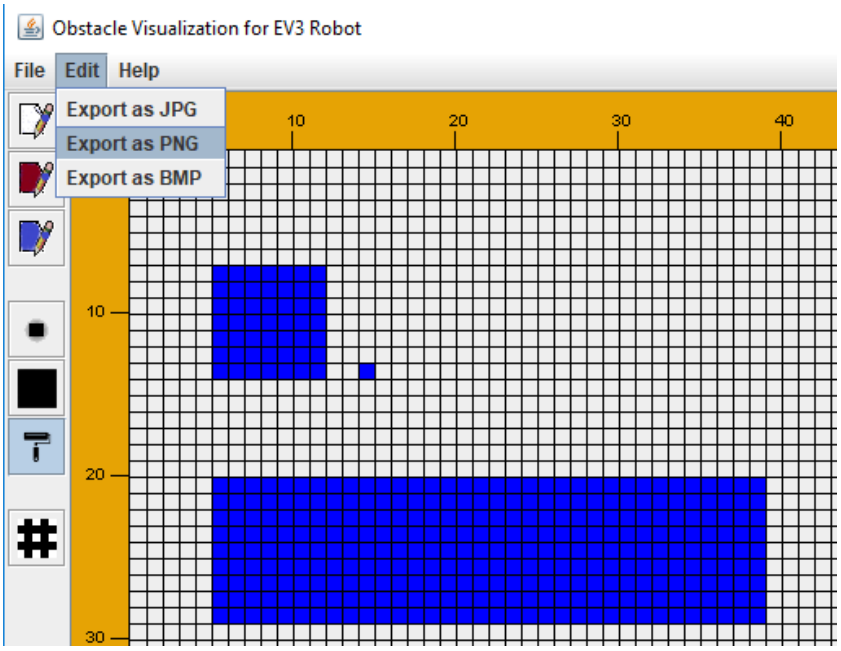
Przycisk File -> Save As pozwala na wybranie lokalizacji oraz nazwy pliku do jakiego chcemy zapisać edytowaną macierz.

Przycisk File -> Close zamyka on edytowaną macierz, a program ma częściową funkcjonalność. Jeśli w macierzy są niezapisane zmiany program zapyta czy je zapisać.

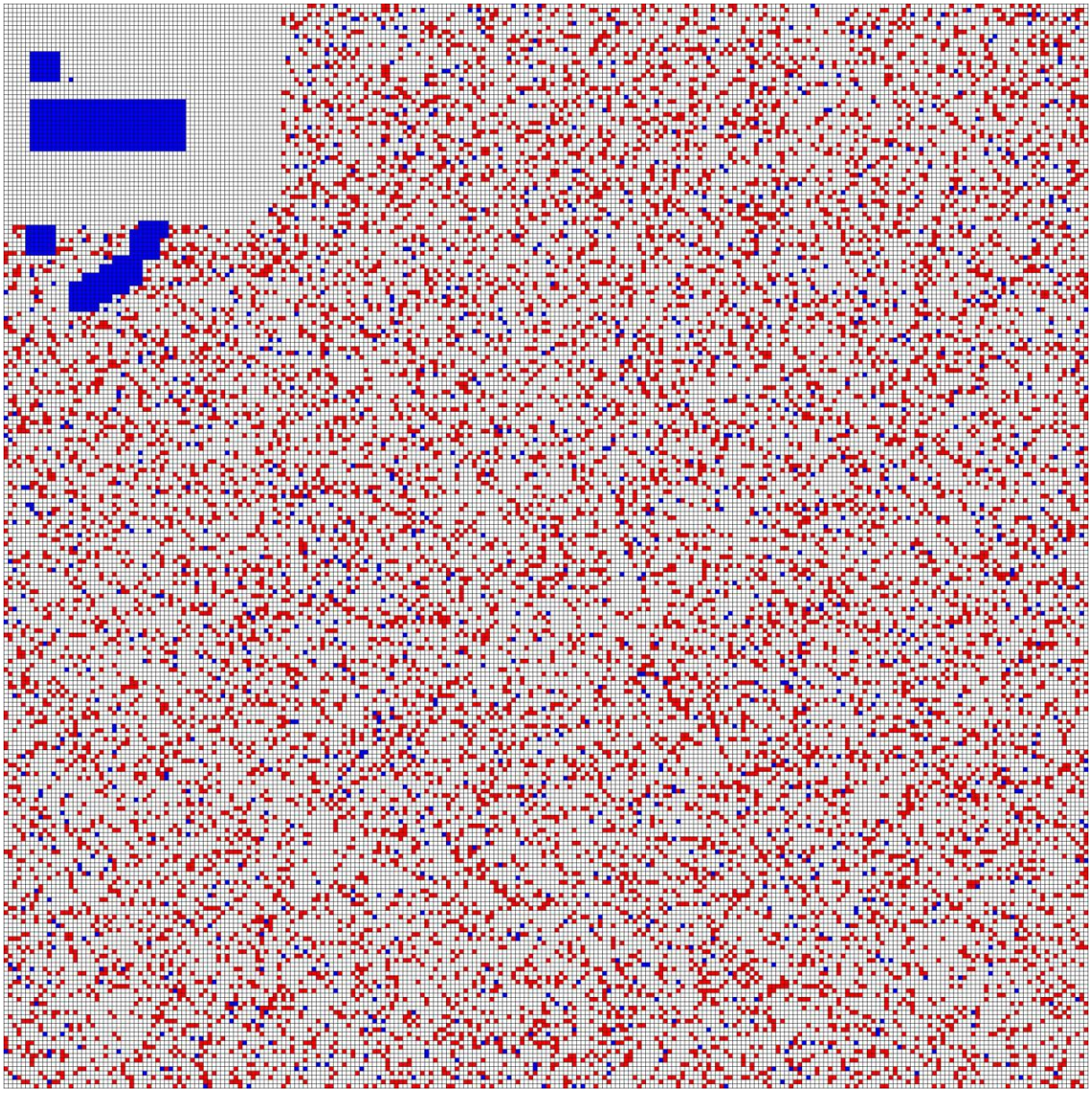


Z kolei kombinacja Przycisk File -> Exit zamyka cały program. Jeśli w macierzy są niezapisane zmiany program zapyta czy je zapisać.

Aby wyeksportować plik w formacie *.png należy kliknąć kolejno pasek górny menu Edit -> Export as PNG.



Wybieramy ścieżkę zapisu pliku po czym klikamy Save. Wyeksportowany obraz wygląda następująco:



6. Wykorzystane technologie:

Aplikacja została napisana w Javie 8. To najnowsza wersja języka Java, która zawiera nowe funkcje, rozszerzenia oraz poprawki mające na celu zwiększenie wydajności programowania i obsługi programów Java.

Przy tworzeniu aplikacji z graficznym interfejsem użytkownika (GUI) korzystaliśmy z bibliotek takich jak:

- ❖ Java AWT,
- ❖ Java Swing.

GUI Aplikacji zostało skonstruowane z kilku okien zagnieżdżonych w sobie. Na szczycie jest ramka (**Frame**), do której dodana została przewijana ramką (**ScrolledPane**). W niej znajdują się komponenty (dziedziczące po klasie **JComponent**) takie jak :

- **ContentPane**
- **VerticalRule**
- **HorizontalRule**

Na **ContentPane** rysowane są kwadraty (o wielkości zdefiniowanej przez pole **RESOLUTION**) wypełnione odpowiednim kolorem. Kwadraty o kolorze białym zostały pominięte by przyśpieszyć działanie programu, zastępuje jest z góry zdefiniowane białe tło **ContentPane**. W **ContentPane** rysowana jest też siatka, stworzona z linii (**Line2D**) .

Biblioteka SWING udostępnia ogromną liczbę składników takich jak przyciski, etykiety, pola tekstowe itd.

W projekcie wykorzystaliśmy również z biblioteki AWT przechwytywanie zdarzeń generowanych przez użytkownika (wciśnięcie przycisku, kliknięcie myszą, przeciąganie myszy itp.) za pomocą mechanizmu wychwytywania zdarzeń.

Klasa BufferedImage opisuje obraz z dostępem do buforu danych obrazu. Składa się z ColorModel oraz Raster- danych obrazu. Liczba i typów w SampleModel Rastera musi pasować do numeru i typów wymaganych przez ColorModel do reprezentowania tych kolorów i komponentów. Wszystkie obiekty BufferedImage mają górną lewą współrzędną rogu (0, 0). Każdy Raster używany do tworzenia BufferedImage musi więc mieć minX = 0 i minY = 0. Ta klasa polega na pobieraniu i ustawianiu metod Raster oraz metodach oznaczania koloru ColorModel.

Klasa Raster reprezentująca prostokątną tablicę pikseli. Składa się z modułu DataBuffer, który przechowuje wartości próbek i SampleModel, który opisuje, jak zlokalizować daną wartość próbki w module DataBuffer. Raster definiuje wartości pikseli zajmujących konkretny prostokątny obszar płaszczyzny, niekoniecznie zawierający (0, 0). Prostokąt, znany jako prostokąt ograniczający Raster i dostępny za pomocą metody getBounds, jest określany przez minX, minY, szerokość i wysokość. Wartości minX i minY definiują współrzędne górnego lewego rogu rastera. Odniesienia do pikseli poza prostokątem ograniczającym mogą powodować wyrzucenie wyjątku .