

# Pisanie aplikacji graficznych w nowoczesnym OpenGL

## Zadania kwalifikacyjne

Jakub Szewczyk [kubasz51@gmail.com](mailto:kubasz51@gmail.com)  
Grzegorz Uriasz [gorbak25@gmail.com](mailto:gorbak25@gmail.com)

4 czerwca 2016

Ostateczną wersję rozwiązań zadań proszę przesłać mailowo do nas w terminie podanym na stronie WWW. Zalecamy wcześniejsze przesłanie nam rozwiązań zadań/zadania na nasze adresy mailowe - będziemy wtedy w stanie coś podpowiedzieć w przypadku problemów technicznych. Nie wymagamy wykonania dużej ilości podpunktów w zadaniach - większa ich ilość wymagana będzie dopiero przy dużej ilości uczestników. Tak więc wyślij nam to co udało się Tobie wykonać. Jeżeli coś jest niejasne, to napisz do nas - z chęcią nakierujemy na literaturę bądź wytłumaczymy.

### 1 Czemu chcesz uczestniczyć w tych warsztatach? - Obowiązkowo

Odpowiedz na to pytanie ściśle i zwięźle. Nie jest to zadanie typu "Gdybyś był owocem to jakim uzasadnij twoje stanowisko na minimalnie dwudziestu stronach A4" - oczekujemy tylko krótkiego opisu.

### 2 Drobną rozgrzewką - wczytywanie danych binarnych

Napisz program, który z podanego na wejściu pliku zip (standardowego, bez szyfrowania, nie jest podzielony na części) wypisze listę zawartych tam plików wraz z ich rozmiarami w bajtach. Nie wolno korzystać z gotowych bibliotek do wczytywania plików zip. Pomocne mogą być strony:

Wikipedia - Zip (file format)

PKZIP File format specification

### 3 Macierze

#### 3.1 Podstawowe operacje na macierzach

$$\begin{pmatrix} 4 & 8 & 3 \\ 5 & 3 & 0 \\ 9 & 4 & 2 \end{pmatrix} \times \begin{pmatrix} 4 \\ 3 \\ 6 \end{pmatrix} + \begin{pmatrix} 5 \\ -1 \\ -4 \end{pmatrix} - \left[ \begin{pmatrix} 3 \\ 7 \\ 1 \end{pmatrix} \times (7 \ 3 \ 5) \right] \times \begin{pmatrix} -1 \\ 0 \\ 2 \end{pmatrix} = ?$$

#### 3.2 Składanie przekształceń w 2D

Znajdź macierz  $A$  która po przemnożeniu przez pewien wektor  $B$  zwróci nam wektor  $C$  który powstaje z wektora  $B$  poprzez ciąg przekształceń geometrycznych:

1. Translacja wektora  $B$  o wektor  $(-5, 1)$
2. Dwukrotne przeskalowanie otrzymanego wektora
3. Obrót otrzymanego wektora względem początku układu współrzędnych o kąt  $\frac{\pi}{4}$  przeciwnie do ruchu wskazówek zegara

Matematycznie macierz  $A$  spełnia warunek  $A \times B = C$ . Odpowiedz na pytanie czy macierz  $A$  zależy od tego czy  $B$  jest punktem czy wektorem.

### 3.3 Odwracanie macierzy

Odwróć macierze:

$$\begin{pmatrix} 7 & 5 & 8 \\ 2 & 7 & 0 \\ 12 & 4 & 4 \end{pmatrix}^{-1} \quad \text{i} \quad \begin{pmatrix} 2\sqrt{3} & 3\sqrt{3} \\ \sqrt{3} & 5\sqrt{3} \end{pmatrix}^{-1}$$

## 4 Zadanie Implementacyjne 1 – fraktal

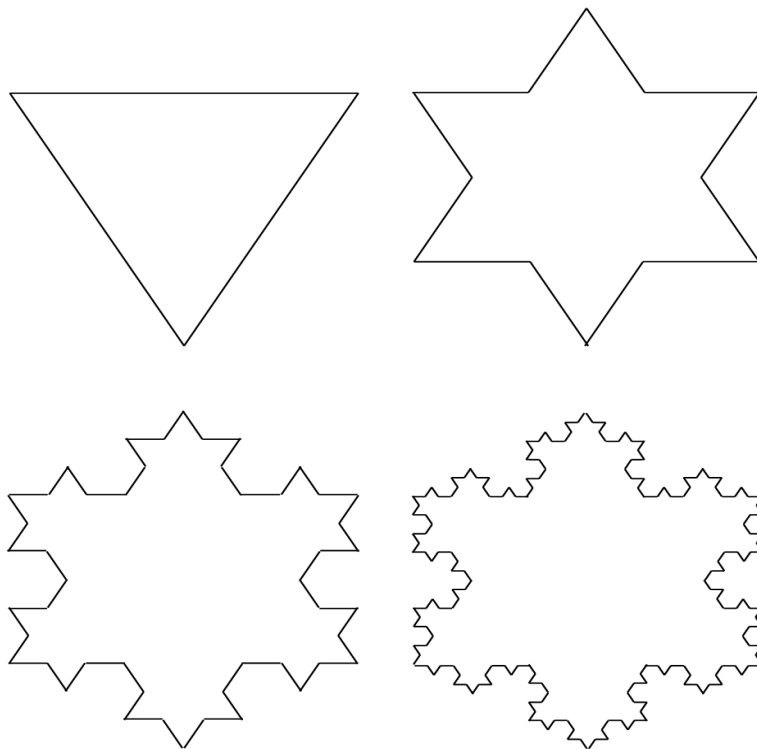
Zalecamy wykonanie tego zadania w C/C++, ale nie mamy nic przeciwko zrobieniu go w innym języku programowania.

Rozwiązaniem tego zadania jest kod źródłowy programu oraz kilka przykładów działania tego programu - najlepiej wygenerowane przez niego fraktale dla pewnych ustalonych przez Ciebie parametrów. Jeżeli zrobiłeś tylko jeden podpunkt to o nic się nie martw - po prostu wyślij nam to rozwiązanie.

### 4.1 Śnieżynka Kocha

Korzystając z dowolnej biblioteki umożliwiającej eksportowanie obrazu do pliku takiej jak np. LodePNG czy SFML wygeneruj Śnieżynkę Kocha.

Śnieżynka Kocha jest fraktalem o nieskończonej długości, ale pomimo tego jako że da się ją zmieścić na skończonej powierzchni można wygenerować pewne jej przybliżenie. Program na wejściu powinien zapytać o podanie ilości iteracji algorytmu oraz nazwę pliku do którego ma zostać zapisany wynikowy fraktal. W każdej iteracji algorytmu należy każdy odcinek podzielić na trzy równe części a następnie środkowy z nich zamienić na trójkąt równoboczny bez podstawy, skierowany na zewnątrz. Zerową iteracją jest trójkąt równoboczny. Oto, co kolejne iteracje algorytmu powinny generować:



## 4.2 Eksporter bez kompresji do BMP

W poprzednim podpunkcie do wyeksportowania grafiki użyłeś zewnętrznej biblioteki - zmodyfikuj teraz twój program tak aby obraz był zapisywany do niezkompresowanej bitmapy, nie można przy tym korzystać z zewnętrznych bibliotek do eksportowania obrazu. Pomocny może być link: [Wikipedia - BMP\(file format\)](#)

## 4.3 Dla Ambitnych - Zbiór Mandelbrota

Korzystając z dowolnej biblioteki umożliwiającej eksportowanie obrazu do pliku takiej jak np. LodePNG czy SFML dołóż do twojego programu możliwość wygenerowania zbioru Mandelbrota. Zbiór Mandelbrota jest zbiorem punktów na płaszczyźnie zespolonej. Aby móc narysować ten zbiór, przyjmij, że w układzie współrzędnych obrazka oś góra-dół reprezentuje oś liczb urojonych a oś lewo-prawo reprezentuje oś liczb rzeczywistych. Załóż że punkt  $z = 0 + 0 * i$  znajduje się dokładnie w samym środku obrazka. W takim układzie załóż że dla obrazka o wysokości  $h$  i szerokości  $w$  punkt o współrzędnych  $(x, y)$  w układzie obrazka reprezentuje liczbę  $z = (\lfloor w/2 \rfloor - x) + (\lfloor h/2 \rfloor - y) * i$ . Zbiór Mandelbrota tworzą punkty  $c \in \mathbb{C}$  dla którego ciąg zdefiniowany w ten sposób:

$$z_1 = c, z_{n+1} = z_n^2 + c \quad (1)$$

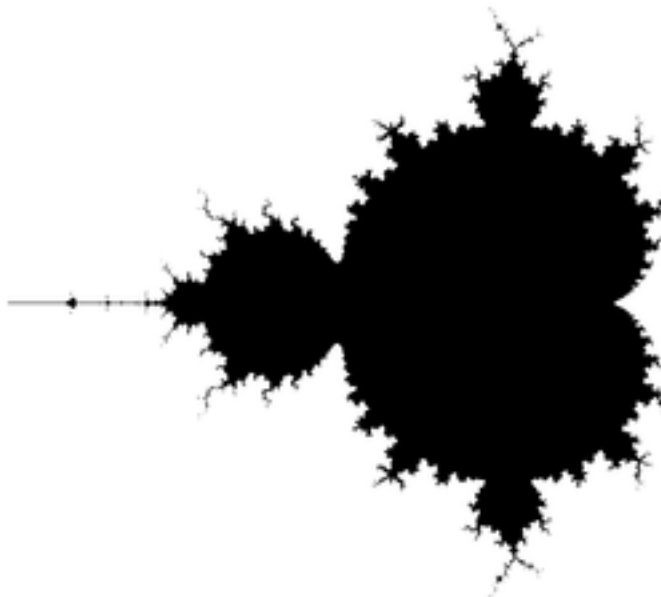
Spełnia równanie:

$$\forall n \in \mathbb{N} \Re(z_n)^2 + \Im(z_n)^2 \leq 4 \quad (2)$$

Podnoszenie liczby zespolonej  $z = a + b * i$  gdzie  $a, b \in \mathbb{R}$  do kwadratu definiujemy:

$$z^2 = (a^2 - b^2) + (2 * a * b) * i \quad (3)$$

Osoby które nigdy wcześniej nie spoktały się z liczbami zespolonymi proszone są o patrzenie na liczbę  $z = a + b * i$  jako punktu na płaszczyźnie o współrzędnych  $(a, b)$ . Warunek (2) należy sprawdzać przez zadaną z góry liczbę elementów ciągu (1). Przyjmij że punkty należące do zbioru Mandelbrota są czarne a te leżące poza nim są białe. Jeżeli wszystko zrobiłeś dobrze to powinieneś otrzymać obrazek podobny do tego:



Powodzenia i do zobaczenia w Wierchomli!