**SPRAWOZDANIE**

**Projekt: Fraktale**

**Informatyka I**

**Bartosz Kościanek, nr. indeksu 292749**

**28.12.17**

**Spis treści**

1. [Wstęp](#_Wstęp)
2. [Podstawy działania programu](#_Podstawy_działania_programu)
   1. [Lindenmayer](#_Lindenmayer)
   2. [Turtle](#_Turtle)
3. [Poszczególnie fraktale – efekty](#_Poszczególne_fraktale_–)
   1. [Płatek Kocha](#_Płatek_Kocha)
   2. [Wyspa Kocha](#_Wyspa_Kocha)
   3. [Trójkąt Sierpińskiego](#_Trójkąt_Sierpińskiego)
   4. [Fraktal Vicseca](#_Fraktal_Vicseca)
   5. [Fraktale na liczbach zespolonych](#_Fraktale_na_liczbach)
      1. [Zbiór Mandelbrota](#_Zbiór_Mandelbrota)
      2. [Zbiór Julii](#_Zbiór_Julii)
      3. [Płonący Statek](#_Płonący_Statek)
      4. [Armada z Płonącego Statku](#_Armada_z_Płonącego)
      5. [Buddhabrot](#_Buddhabrot)
   6. [Paproć Barnsleya](#_Paproć_Barnsleya)
   7. [Krzywa smocza (Smok Heighwaya / Smok Levy’ego)](#_Krzywa_smocza)
4. [Zakończenie](#_Zakończenie)

# **Wstęp**

**Cele projektu**

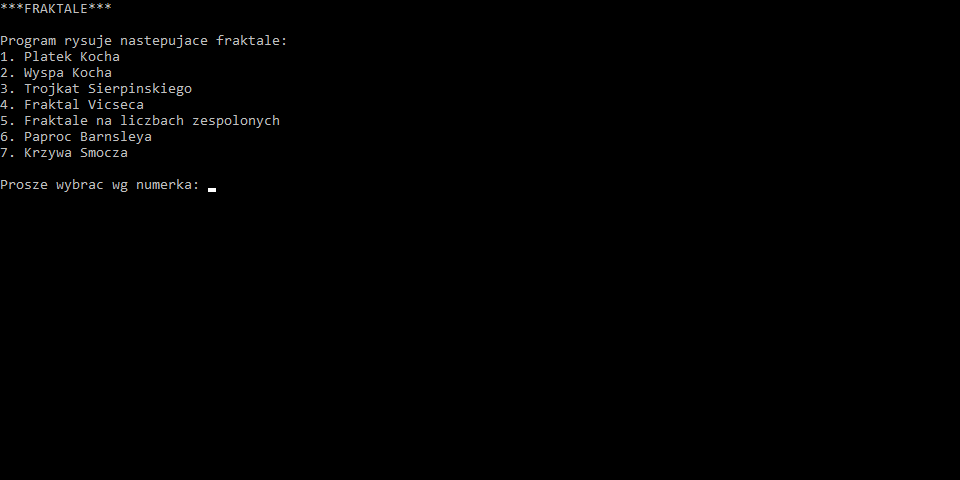
Podstawowym celem projektu było osiągnięcie wizualnie zadowalających prezentacji fraktali głównie przy po pomocy grafiki żółwia oraz systemu Lindenmayera. Postanowiłem jednak poszerzyć zakres o fraktale na liczbach zespolonych, dodając „Płonący Statek” oraz „Buddhabrot” do fraktali z instrukcji X.

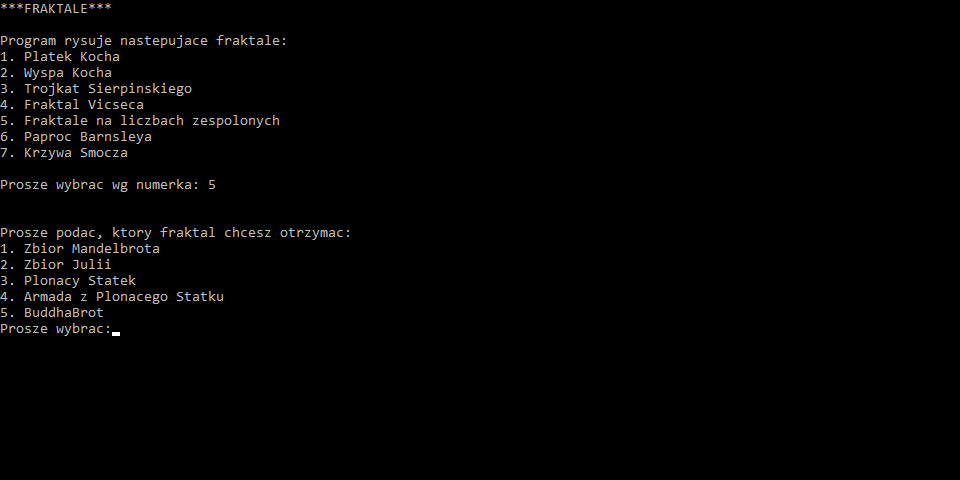
Dodatkowo projekt miał w założeniu posiadać prosty interfejs i przedstawiać użytkownikowi wszystkie opcje, a w przypadku wyboru opcji samodzielnego rysowania dodatkowo prowadzić użytkownika przez kolejne etapy w zrozumiały sposób.

Żeby program był wystarczająco czytelny i niepoplątany, postanowiłem skorzystać z metody „dziel i rządź” i rozłożyć program na kilka mniejszych, każdy rysujący oddzielny typ fraktala.

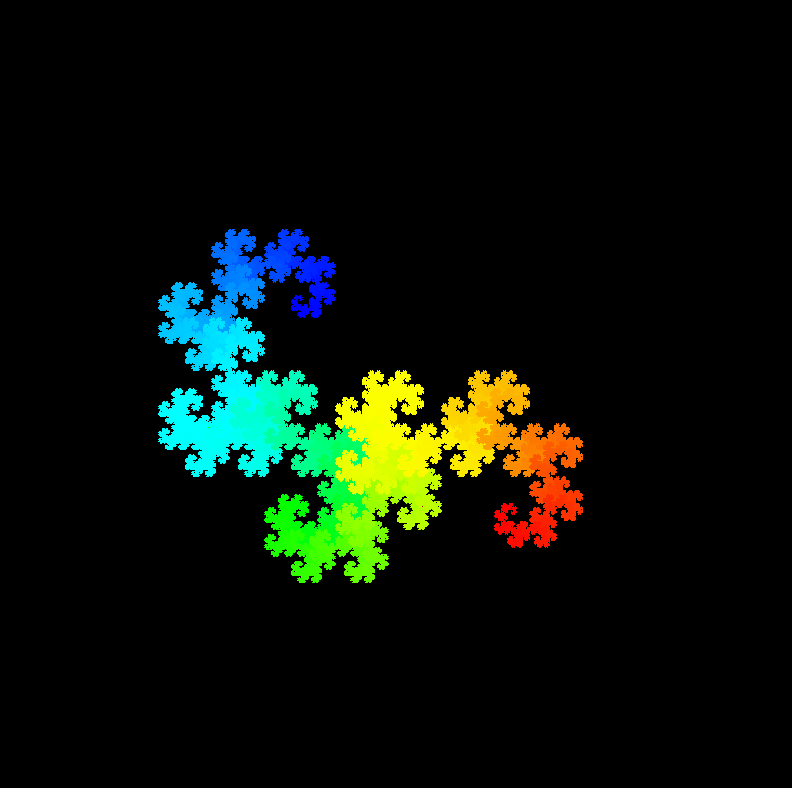
**Działanie i wygląd**

Na początku zostaje przedstawione użytkownikowi menu zawierające wszystkie możliwe do narysowania typy fraktali. W tym momencie należy podać według numeru.



Następnie w zależności od wyboru typu może zostać przedstawiony kolejny wybór, np.:

Dalej odbywa się już właściwe rysowanie fraktala według wybranych przez użytkownika opcji. Niektóre części programu wymagają wprowadzenia samodzielnie dużej ilości danych liczbowych, więc zostały dodane opcje umożliwiające rysowanie oparte jedynie na wprowadzonych współrzędnych i rozmiarze.

Po wprowadzeniu odpowiednich danych na ekranie otworzy się okno graficzne, a w nim, po upływie czasu zależnego od fraktala i liczby iteracji, narysowany zostanie wybrany fraktal:

W przypadku kilku z fraktali wprowadzony został kolor, aby pokazać kolejność i sposób rysowania, oraz aby dodać walorów estetycznych.

# **Podstawy działania programu**

## Lindenmayer

Za jeden z dodatkowych celów programu obrałem sobie wykorzystanie systemu zmian łańcucha znaków Lindenmayera (w skrócie „L-system”).

Metoda polega na wprowadzeniu zmiennych oraz stałych, następnie odpowiadających zmiennym zasadom (np. 1 ⭢ 212) tak, aby program wiedział w jaki sposób ma zamieniać zmienne.

Następnie podaje się aksjomat, czyli początkowy zestaw znaków do transformacji, a po dłuższej chwili (w zależności od liczby iteracji) w pliku „lindenmayer.txt” użytkownik uzyskuje gotowy, przekształcony ciąg znaków. Dodatkowo funkcja zwraca liczbę znaków znajdujących się w pliku, aby można było jej użyć do określenia liczby iteracji pętli w innych programach.

Plik ten można następnie wykorzystać jako zestaw instrukcji dla rysującego żółwia. Wszystko zależy od przyjętych instrukcji i opisujących je zmiennych oraz stałych.

Przykładowe działanie:

## Turtle

Turtle to prosty program, którego jedyną funkcją jest rysowanie linii wzdłuż trasy, po której stworzony żółw się porusza. Żółw przyjmuje postać wektora, w którym przechowywane są: współrzędne x i y, kąt tworzony przez żółwia z osią OX oraz długość linii, którą żółw rysuje. Po każdym ruchu żółw zmienia swoje współrzędne do miejsca, w którym skończył rysować, co pozwala na tworzenie rysunków, mając do dyspozycji zestaw instrukcji, np. wyników działania Lindenmayer.cpp.

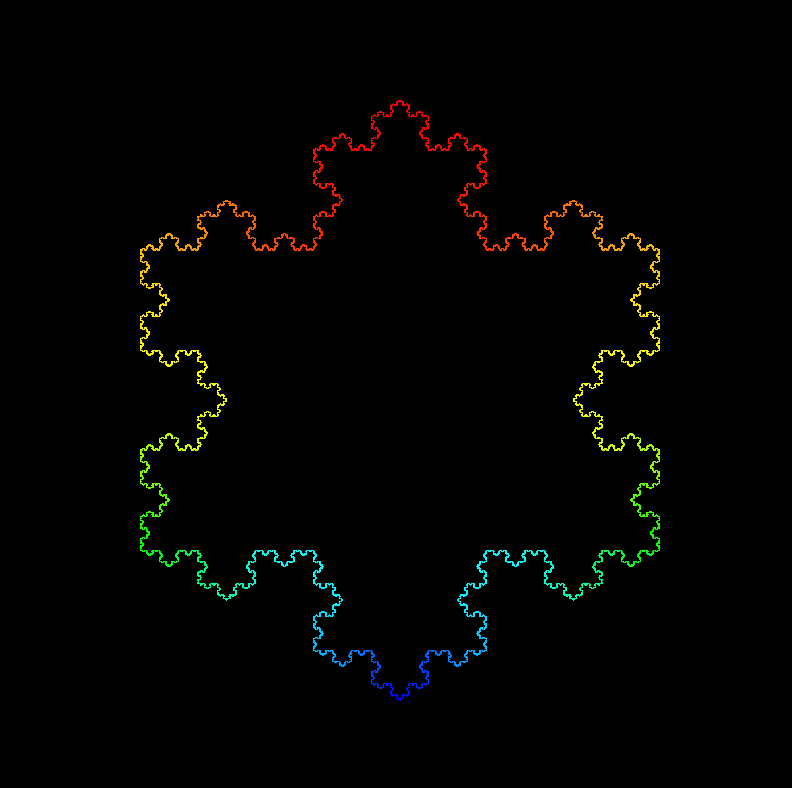
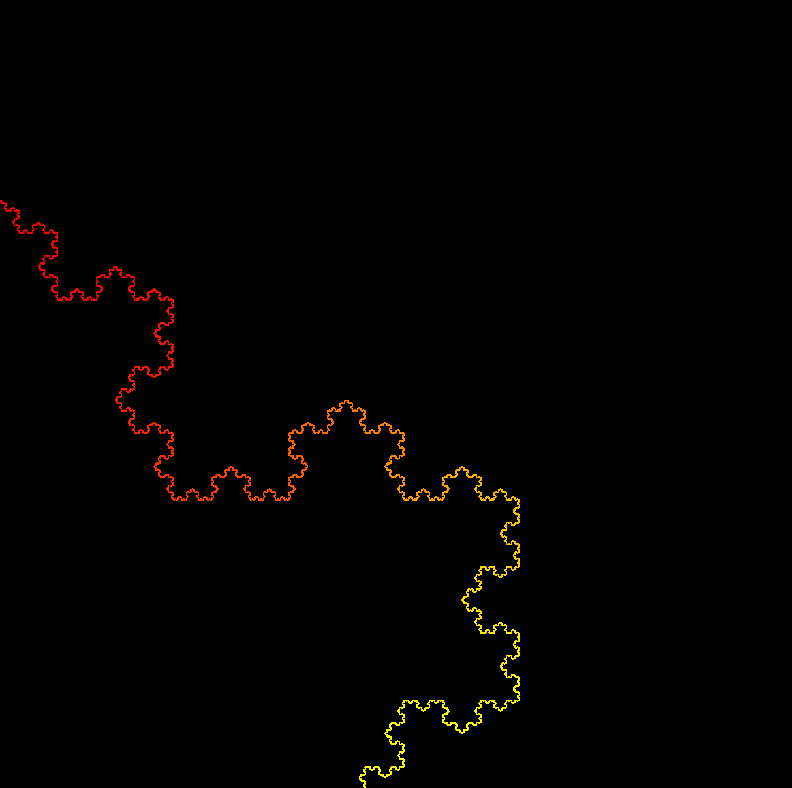
Dostępne są jedynie 4 funkcje, wystarczające do działania programu:

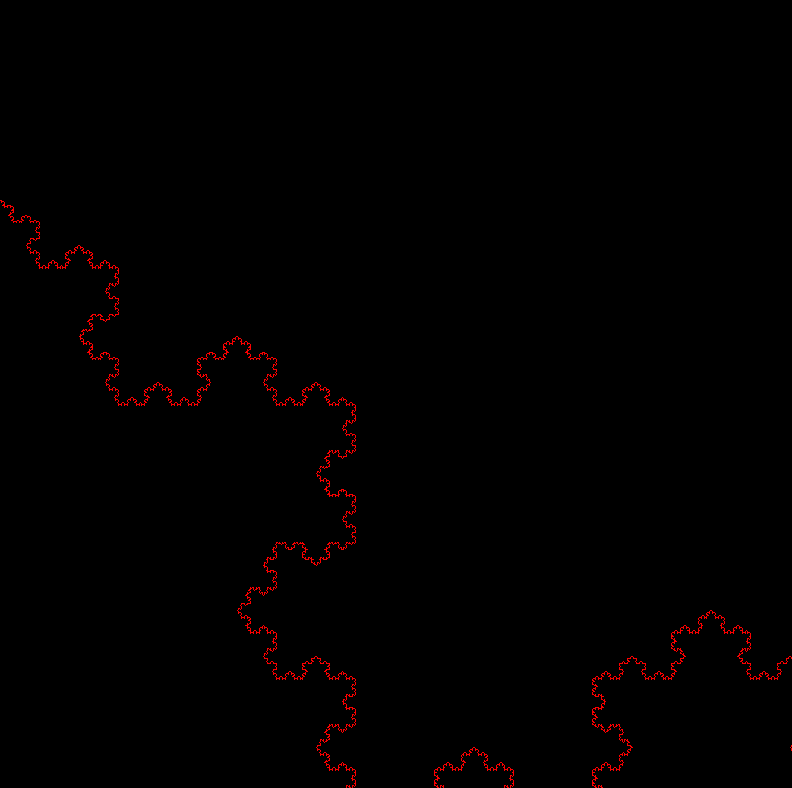
1. void draw(double \*turt) – rysowanie i przemieszczanie do przodu
2. void turnL(double \*turt) – obrót w lewo o zadaną liczbę stopni
3. void turnR(double \*turt) – obrót w prawo o zadaną liczbę stopni
4. void turnB(double \*turt) – powrót do miejsca sprzed narysowania linii

Użytkownik rozpoczynając pracę z Turtle.cpp powinien mieć gotowego żółwia – tablicę.

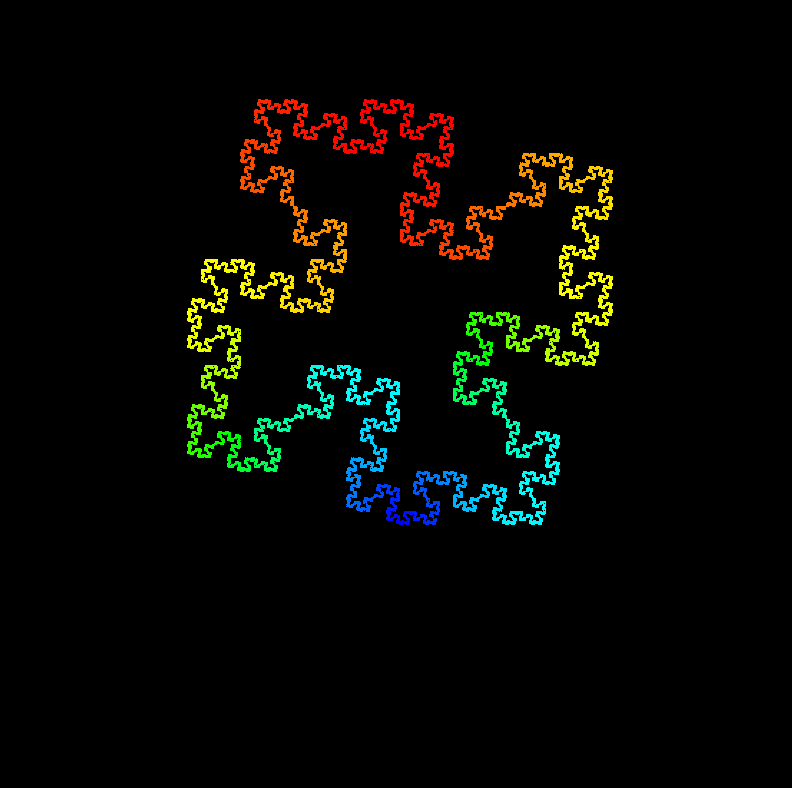
# **Poszczególne fraktale – efekty**

## Płatek Kocha

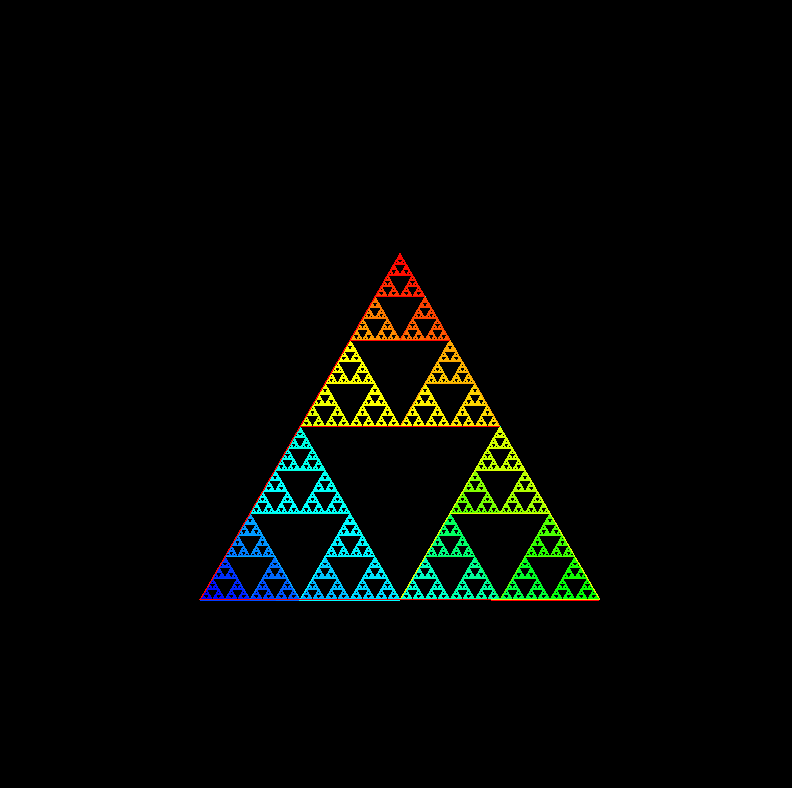




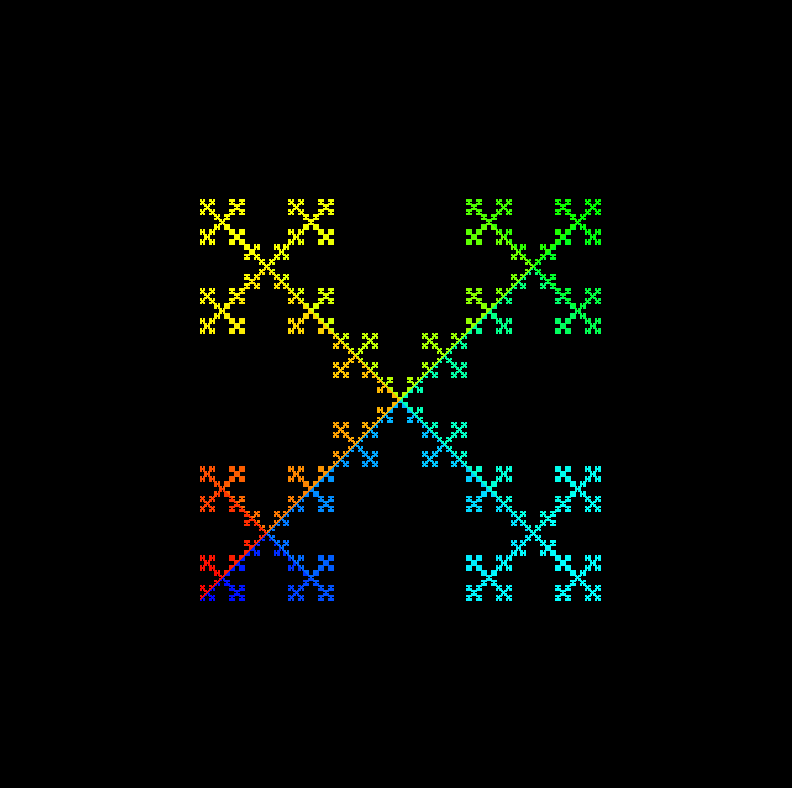
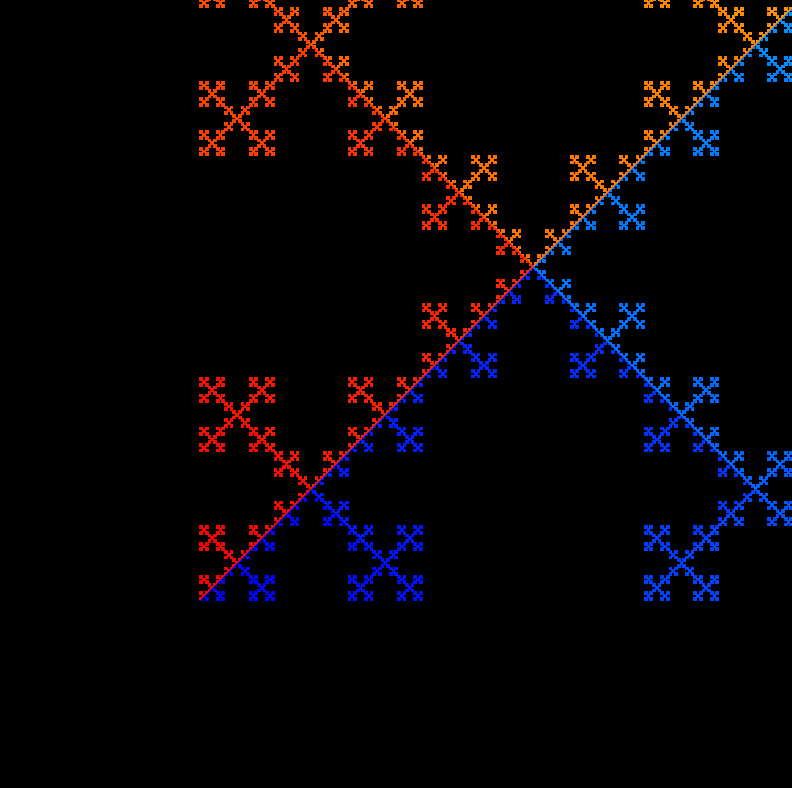
## C:\Users\bkosc\OneDrive\Documents\ShareX\Screenshots\2018-01\PROJEKT_2018-01-10_15-47-49.pngWyspa Kocha



## Trójkąt Sierpińskiego

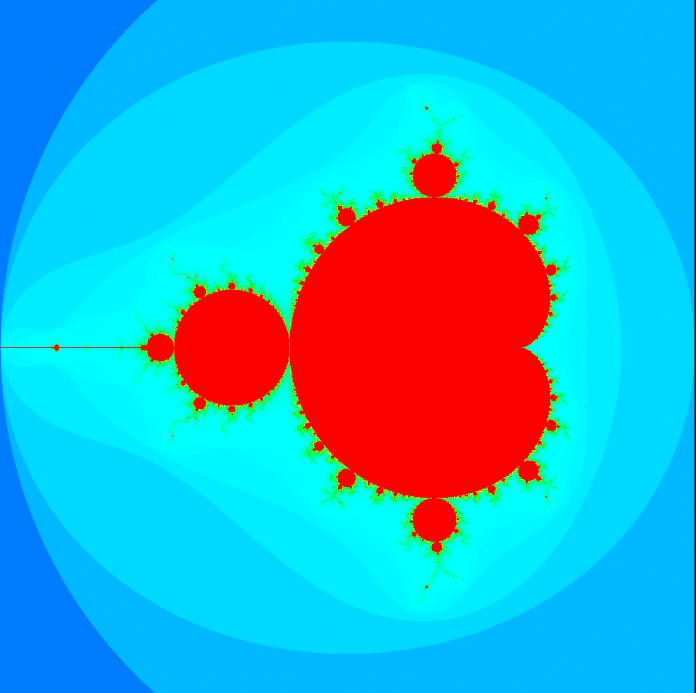
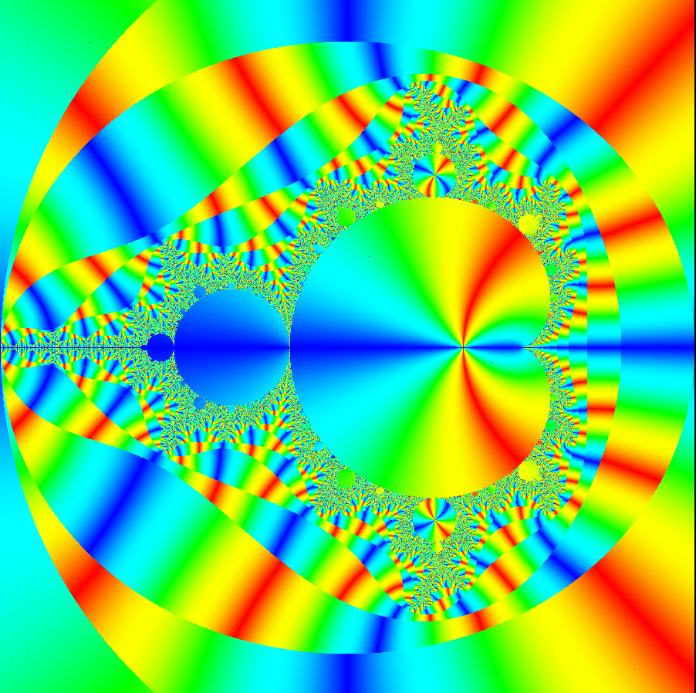


## Fraktal Vicseca

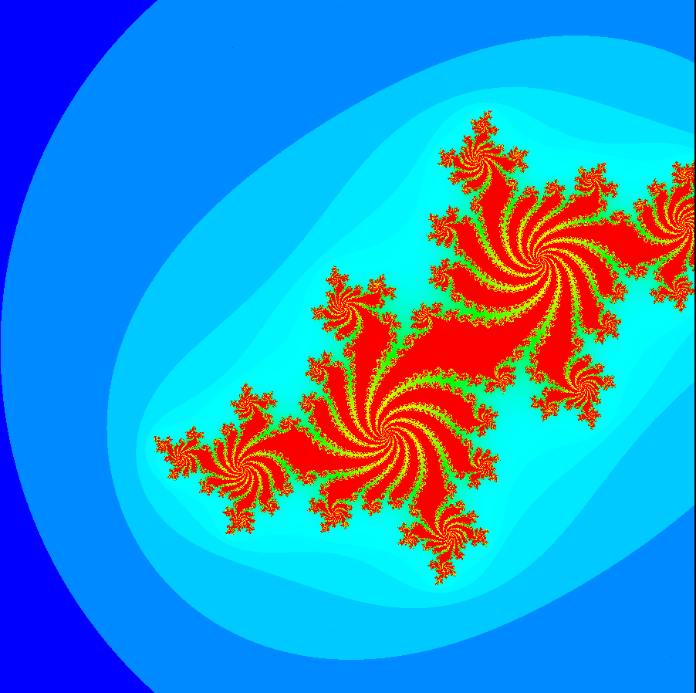
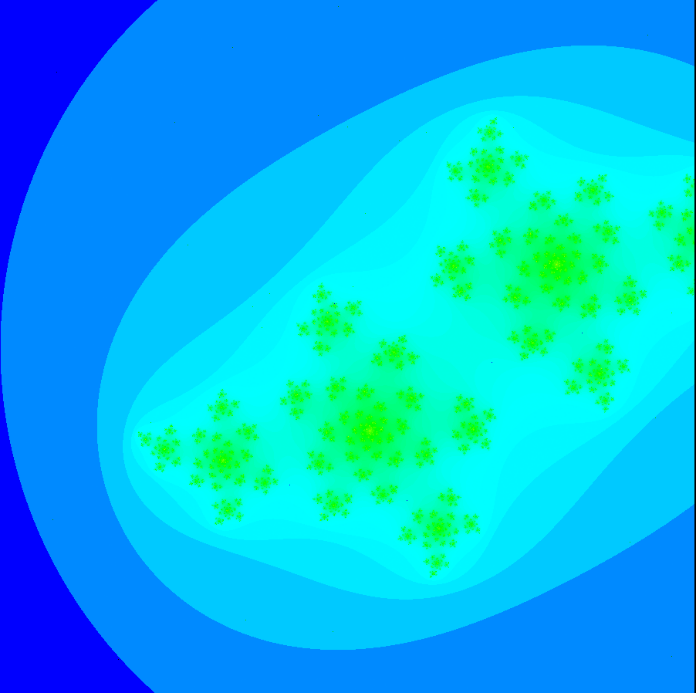
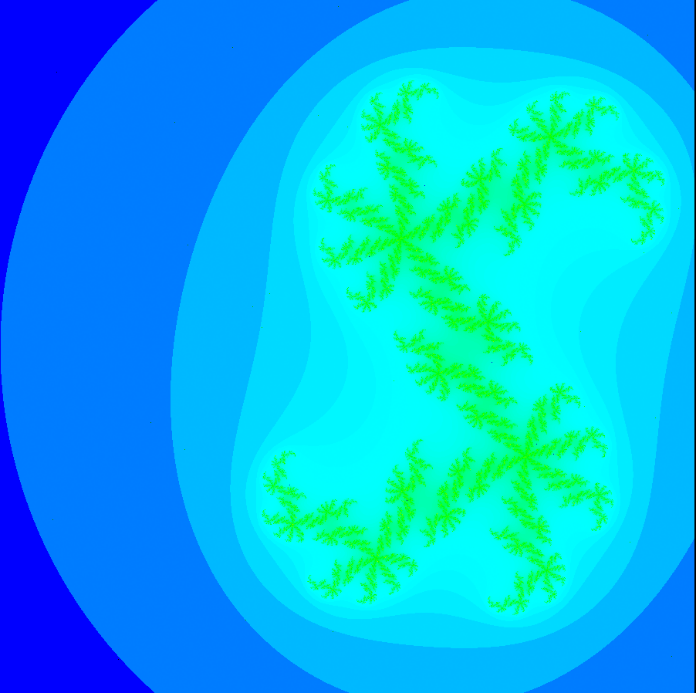


# Fraktale na liczbach zespolonych

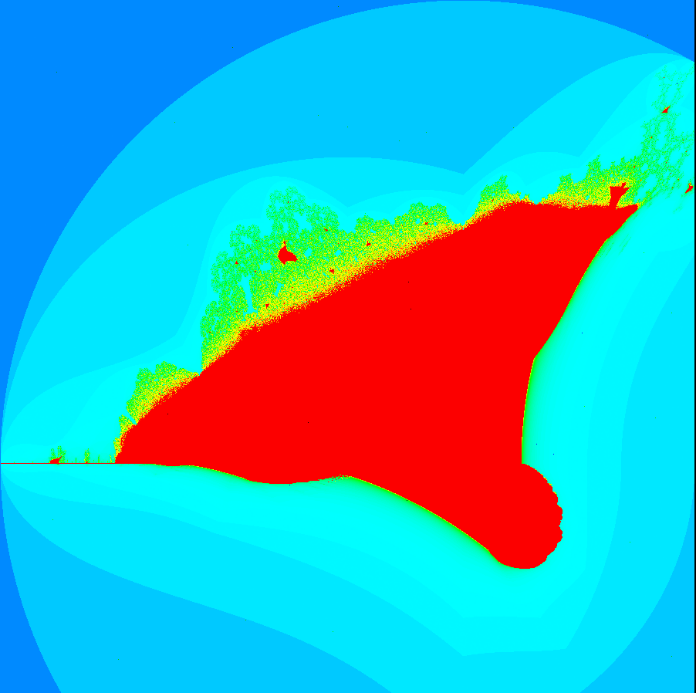
## Zbiór Mandelbrota



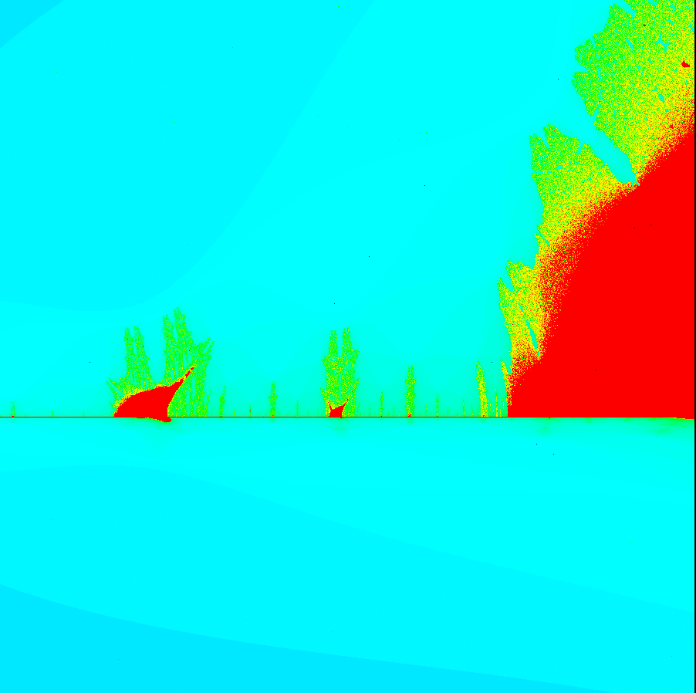
## Zbiór Julii



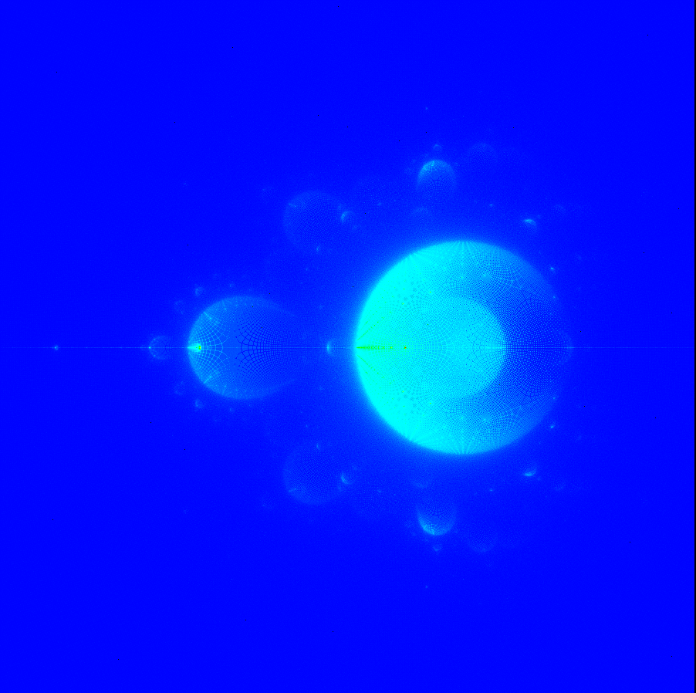
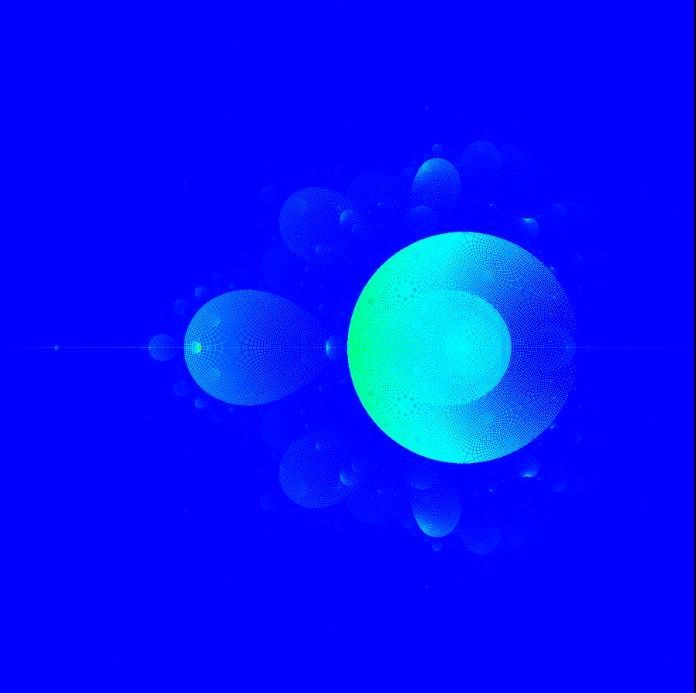
## Płonący Statek



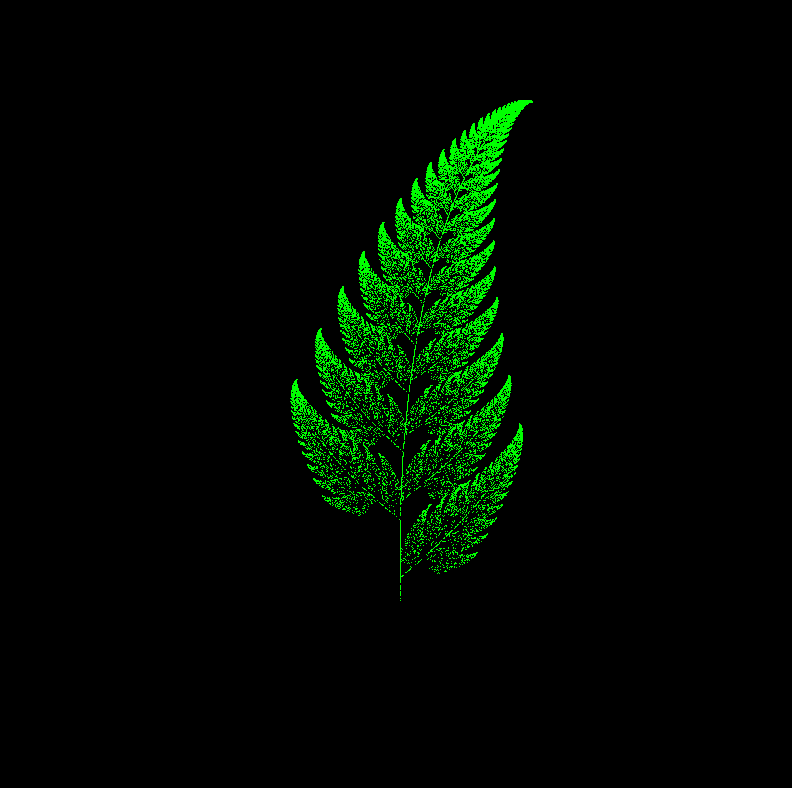
## Armada z Płonącego Statku



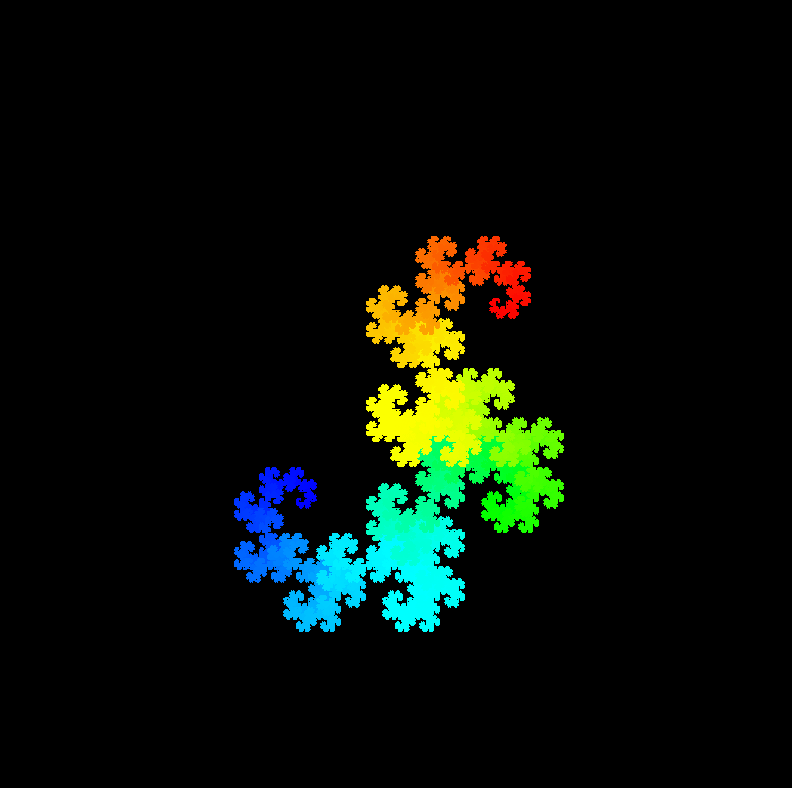
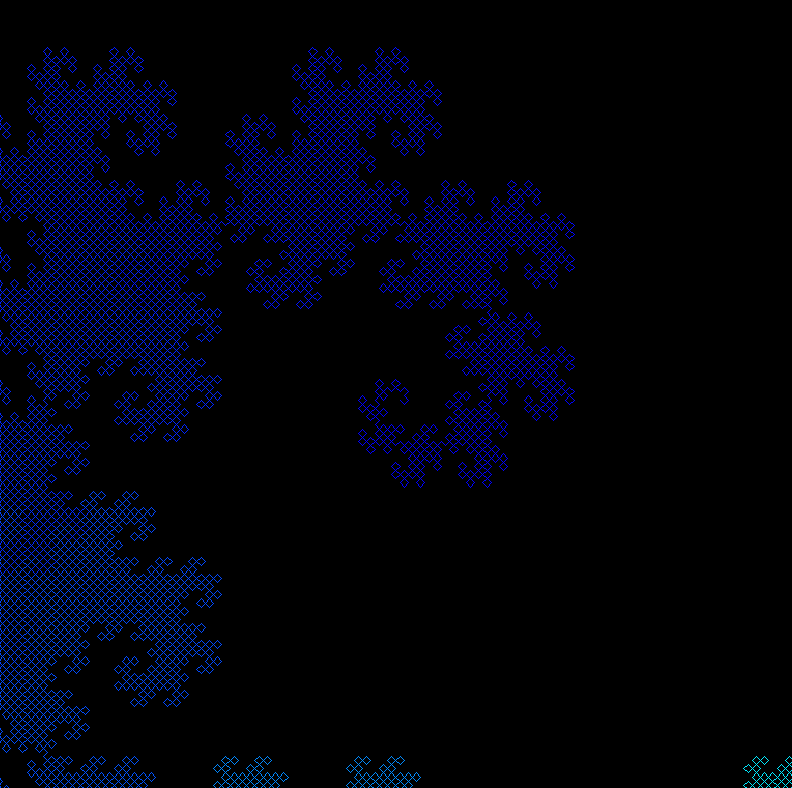
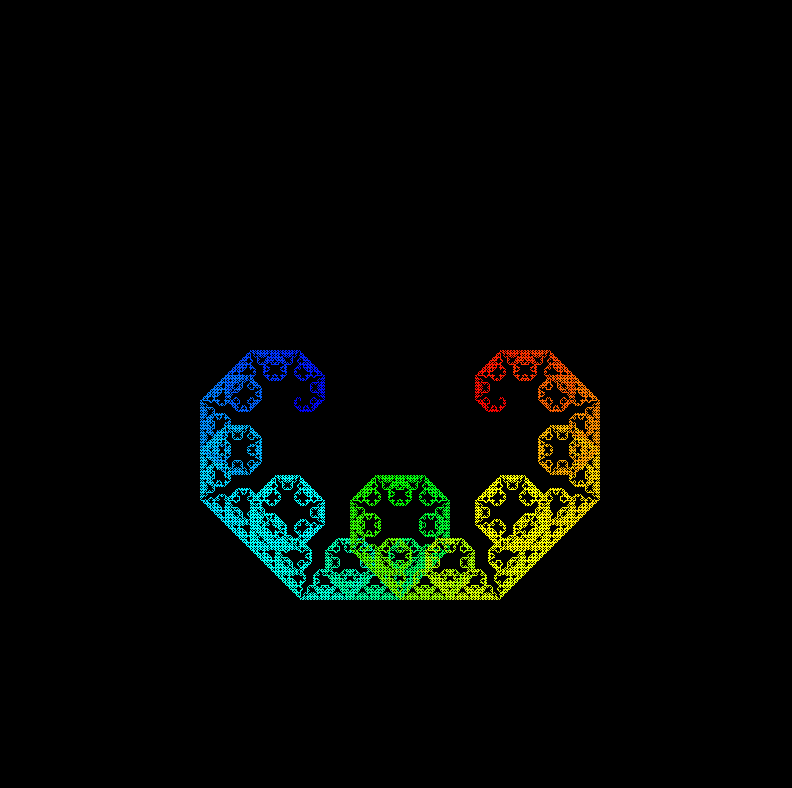
## Buddhabrot

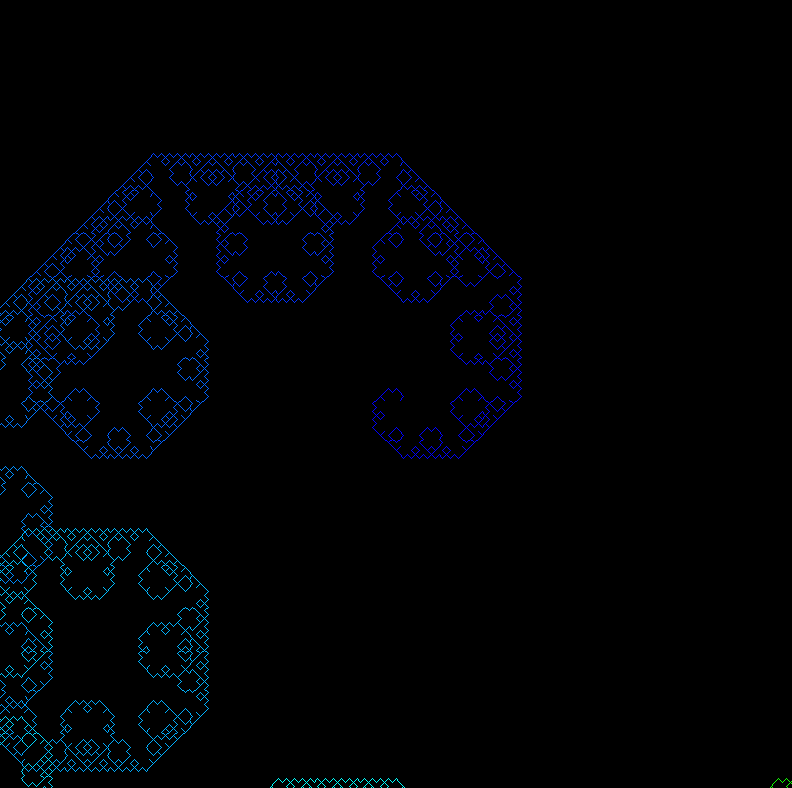


## Paproć Barnsleya



## Krzywa smocza





# **Zakończenie**

Uważam, że na obecne umiejętności i biorąc pod uwagę założenia wyczerpałem temat fraktali i projekt jest ostatecznie zakończony, ale nie bez miejsca na kilka ulepszeń.

Przede wszystkim w zakresie liczb zespolonych istnieje o wiele więcej funkcji mogących wyprodukować fraktale. Szczególnie nieprzewidywalnie i ładnie mogły by się prezentować efekty funkcji, które nie są wielomianami. Można by również wykorzystać metodę Newtona, na której dobre opisanie brakuje miejsca.

Kolejnym godnym rozważenia aspektem było by przybliżanie widoku i przesuwanie fraktala, choć przy ilości czasu potrzebnego na wygenerowanie jednego obrazu taka opcja mogła by wymagać obniżenia dokładności.

Podsumowując, jestem zadowolony z wykonanej pracy i poświęconego czasu. Tworzenie fraktali i posługiwanie się C w celu stworzenia tego projektu kompletnie mnie zaabsorbowało i myślę że w przyszłości postaram się poszerzyć możliwości programu.