



Sprawozdanie z Laboratorium nr 2

Przedmiotu Grafika komputerowa i komunikacja człowiek-komputer
Temat: Podstawy OpenGL

Kyrylo Semenchenko 273004
Informatyka Techniczna I stopnia
Wydział Informatyki i Telekomunikacji

23 października 2024

Spis treści

1	Wstęp	2
2	Wykonane zadania	2
2.1	Zadanie 1	2
2.2	Zadanie 2	2
2.3	Zadanie 3	3
2.4	Zadanie 4	3
2.5	Zadanie 5 (zadanie domowe)	5

1 Wstęp

W ramach drugiego laboratorium należało:

- Zapoznać się z podstawowymi elementami grafiki komputerowej,
- Zrozumieć proces powstawania obrazu w komputerze,
- Oswoić się z interfejsem OpenGL na przykładach 2D.

Do realizacji tych celów przygotowano środowisko pracy w wybranym języku programowania – Pythonie. Pobrano niezbędne biblioteki, a także przykładowy program. Następnie przystąpiono do wykonywania zadań.

2 Wykonane zadania

2.1 Zadanie 1

Należało:

- Przerobić kod,
- Narysować trójkąt, w którym każdy wierzchołek ma inny kolor,
- Pamiętać o ustawionych parametrach rzutni:
 - Zakres na osi X: od -100.0 (lewa strona) do 100.0 (prawa strona),
 - Zakres na osi Y: od -100.0 (dół okna) do 100.0 (góra okna),
 - Punkt o współrzędnych ($X = 0$, $Y = 0$) znajduje się w środku okna.

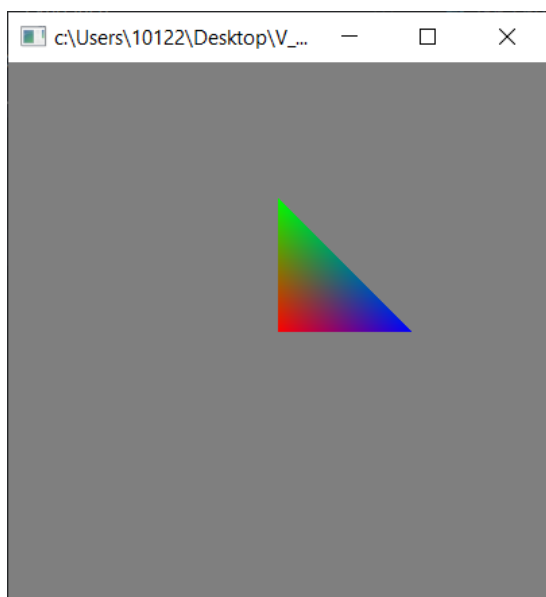
Wynik renderowania pierwszego kodu przedstawiono na Rysunku 1.

2.2 Zadanie 2

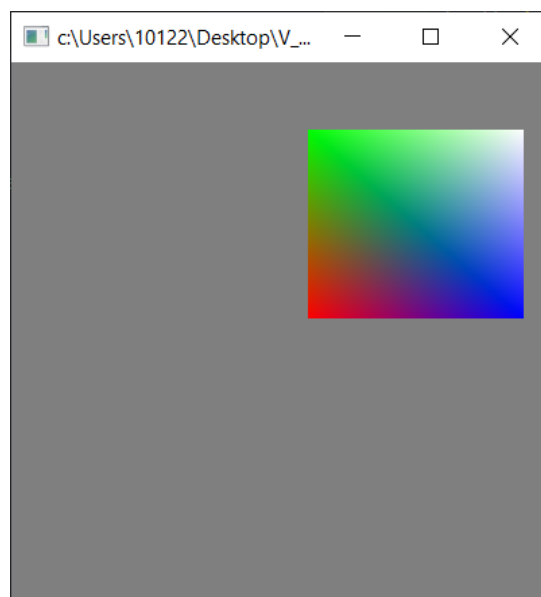
Należało:

- Dodać nową funkcję, która przyjmuje cztery argumenty:
 - Położenie na osi X – x ,
 - Położenie na osi Y – y ,
 - Rozmiar pierwszego boku – a ,
 - Rozmiar drugiego boku – b .
- Położenie (x , y) może wskazywać środek prostokąta lub jego wierzchołek. Punkt (x , y) określa się fachowo jako *punkt początkowy* (ang. *origin point*).
- Na tej podstawie należy wyznaczyć współrzędne reszty wierzchołków bryły.
- Do narysowania prostokąta należy wykorzystać dokładnie dwa trójkąty.
- Funkcję należy wywołać przykładowo w ramach `render()`.

Wynik renderowania kodu przedstawiono na Rysunku 2.



Rysunek 1: Renderowanie trójkąta z kolorowymi wierzchołkami.



Rysunek 2: Renderowanie prostokąta z kolorowymi wierzchołkami.

2.3 Zadanie 3

Należało wprowadzić losowość kolorów i deformacje w prostokącie. Wskazówki:

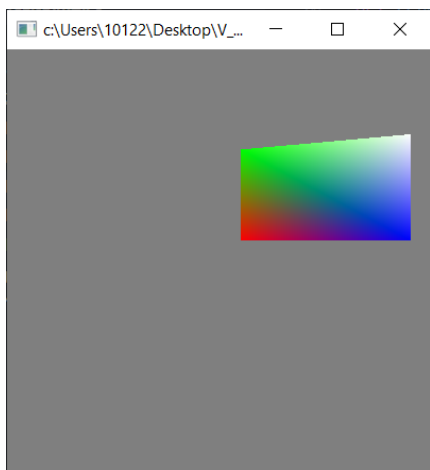
- Rozbudować funkcję z poprzedniego zadania, na przykład:
 - Dodać kolejny argument do funkcji – d – z domyślną wartością 0.0,
 - Nowy argument powinien sterować stopniem deformacji,
 - Można przeskalować rozmiary boków a i b .
- Uzyskać losową wartość w Pythonie:
 - Załadować bibliotekę `random`,
 - Przykładowe wywołanie: `random.random()`,
 - Przydatne może być także użycie `random.seed(...)`.

Wynik renderowania kodu pokazano na Rysunku 3.

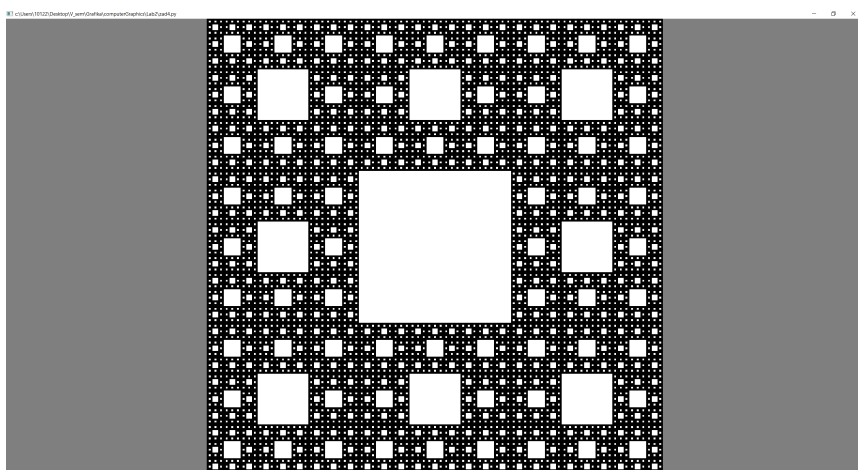
2.4 Zadanie 4

Należało:

- Istnieją dwa podejścia do narysowania tego fraktalu:
 - Rysować poszczególne małe prostokąty w wyznaczonych miejscach.
 - Narysować duży prostokąt i pomniejsze w miejscach "wycięć".
- Wykorzystać funkcje z poprzednich przykładów:
 - Najpierw narysować zarys fraktalu z ręcznie rozmieszczonych brył, aby wyznaczyć współrzędne interesujących nas prostokątów.



Rysunek 3: Renderowanie asymetrycznego prostokąta z kolorowymi wierzchołkami.



Rysunek 4: Fraktal rysowany przy użyciu rekurencji.

- Następnie ubrać całość w funkcję rekurencyjną i powtórzyć rysowanie w wyznaczonych współrzędnych. Z każdym stopniem rekurencji pomniejszać rozmiary boków.

- Stopień samopodobieństwa powinien być parametrem programu.

W tym zadaniu został zaimplementowany dywan Sierpińskiego (*Sierpinski Carpet*) przy użyciu OpenGL, gdzie złożoność fraktalu jest kontrolowana przez parametr głębokości rekurencji.

Opis algorytmu:

- Algorytm rekurencyjnie dzieli dany prostokąt na dziewięć równych mniejszych prostokątów.
- Prostokąt w centrum każdego podziału jest "wycięty" (pozostawiony biały), podczas gdy pozostałe są czarne.
- Z każdym kolejnym poziomem rekurencji rozmiary prostokątów są zmniejszane o jedną trzecią, a procedura jest powtarzana dla wszystkich mniejszych prostokątów oprócz środkowego.

Główne elementy kodu:

- Funkcja `draw_square` rysuje pojedynczy kwadrat o zadanej wielkości w odpowiednich współrzędnych.
- Funkcja `sierpinski_carpet` wywołuje funkcję `draw_square` rekurencyjnie, tworząc strukturę fraktala na podstawie aktualnych współrzędnych i głębokości.
- Program działa w oknie o rozmiarze 400x400 pikseli, gdzie parametr głębokości rekurencji jest pobierany od użytkownika.

Na Rysunku Rysunkku 4 przedstawiono renderowanie fraktala dywanu Sierpińskiego. Z każdym stopniem rekurencji zwiększa się liczba prostokątów, co prowadzi do bardziej złożonego wzoru. Algorytm generuje zróżnicowane struktury w zależności od głębokości rekurencji wybranej przez użytkownika.

2.5 Zadanie 5 (zadanie domowe)

Należało:

- Wybrać jeden z przykładów zaproponowanych jako zadania domowe. Dokument znajduje się na stronie prowadzącego.
- Interesujące były pomysły na generację fraktali. Nie trzeba było implementować różnych wariantów tego samego fraktala.

Został wybrany fraktal plazmowy (*plasma fractal*), a polecenie zostało wykonane na podstawie dostępnych materiałów.

Opis algorytmu:

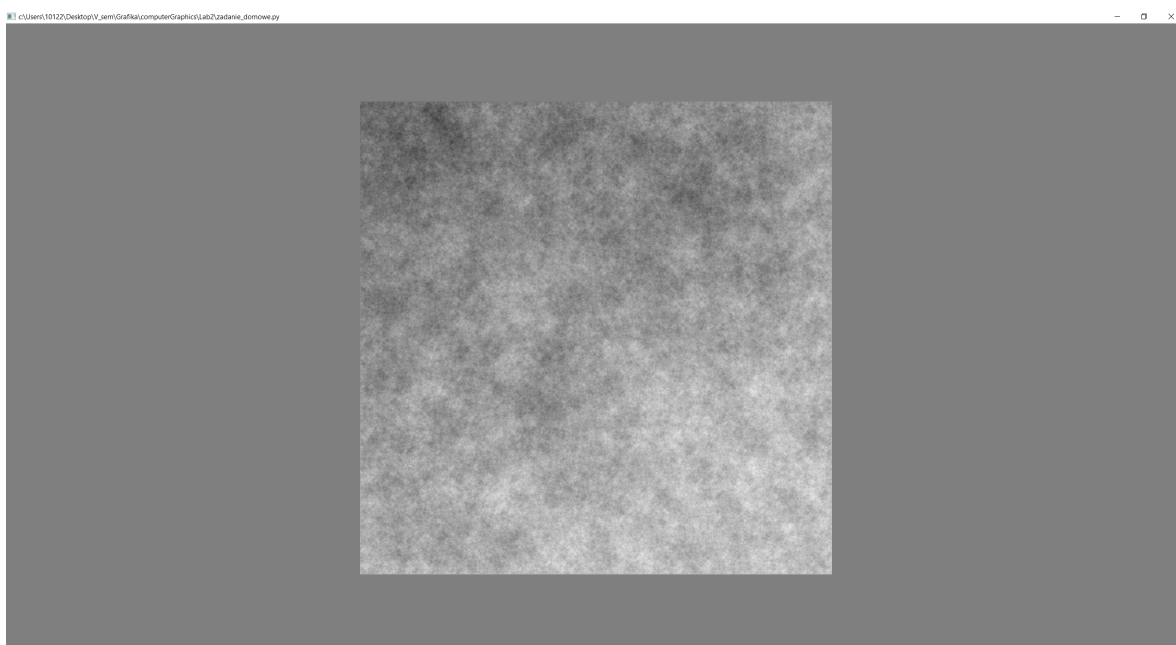
Fraktal plazmowy został wygenerowany na podstawie metody podziału kwadratu, która działa rekurencyjnie:

- Początkowy kwadrat jest podzielony na cztery mniejsze.
- Każdy wierzchołek kwadratu ma przypisany losowy kolor.
- Środkowy punkt kwadratu oblicza się jako średnią kolorów wierzchołków, z dodatkowym losowym przesunięciem, aby uzyskać efekt szumu.
- Algorytm jest powtarzany dla każdego z czterech nowych kwadratów, aż do osiągnięcia zadanego minimalnego rozmiaru.

Główne elementy kodu:

- Funkcja `plasma` realizuje rekurencyjne podziały i oblicza kolory na podstawie wartości wierzchołków.
- Funkcja `render` odpowiada za rysowanie fraktala na ekranie poprzez kolejne wywołania funkcji `plasma`.
- Program tworzy okno o rozmiarze 400x400 pikseli, w którym fraktal jest dynamicznie rysowany.

Na Rysunku 5 przedstawiono renderowanie fraktala plazmowego. Wygenerowany obraz ukazuje płynne przejścia kolorystyczne, które są efektem losowego szumu dodawanego do wartości kolorów wierzchołków kwadratów. Algorytm generuje zróżnicowane wzory w zależności od losowych wartości przypisywanych na każdym etapie rekurencji.



Rysunek 5: Fraktal plazmowy.