

Sprawozdanie: Ćwiczenia nr. 2

Andrzej Borecki, indeks: 226205

28 stycznia 2019

1 Omówienie tematu

Celem zajęć było zaprezentowanie podstawowych możliwości biblioteki graficznej OpenGL razem z toolkit'em GLUT. Wykonane zostało poprzez realizację kilku programów realizujących kolejne funkcje biblioteki w rysowaniu 2D, począwszy od samego stworzenia okna i narysowania w nim podstawowego kształtu, do prób stworzenia nieregularnego w kształcie oraz kolorach, rekurencyjnie generowanego Dywanu Sierpińskiego.

2 Omówienie kodu

Poniższy kod realizuje rysunek ośmiu kwadratów o zdeformowanych kształtach i losowych kolorach. Aby dalej rozwijać program należałoby wywołać rekurencyjne tworzenie rzeczonych ośmiu kwadratów w miejsce każdego, już istniejącego kwadratu.

```
#include "pch.h"
#include <windows.h>
#include <gl/gl.h>
#include <gl/glut.h>
#include <cstdlib> // wykorzystywane przy używaniu funkcji rand()
#include <ctime> // korzysta z niej time(NULL)

GLfloat deform = 1; // stopień deformacji kolejnych kwadratów

void RenderScene(void)
{
    glClear(GL_COLOR_BUFFER_BIT); // czyszczenie okna aktualnym kolorem
    for (int i = -25; i <= 75; i = i + 50)
    {
        for (int j = -25; j <= 75; j = j + 50)
        {
            if (!(i == 25 && j == 25))
            {
                // wykonanie 8 'kwadratów' z jednego dużego
                // pomijamy środkowy 'kwadrat'
                glBegin(GL_POLYGON); // zadeklarowanie typu rysowanej figury
                glColor3f(rand() % 10 * 0.1, rand() % 10 * 0.1, rand() % 10 * 0.1);

                // losowanie koloru wierzchołka w formacie RGB
            }
        }
    }
}
```

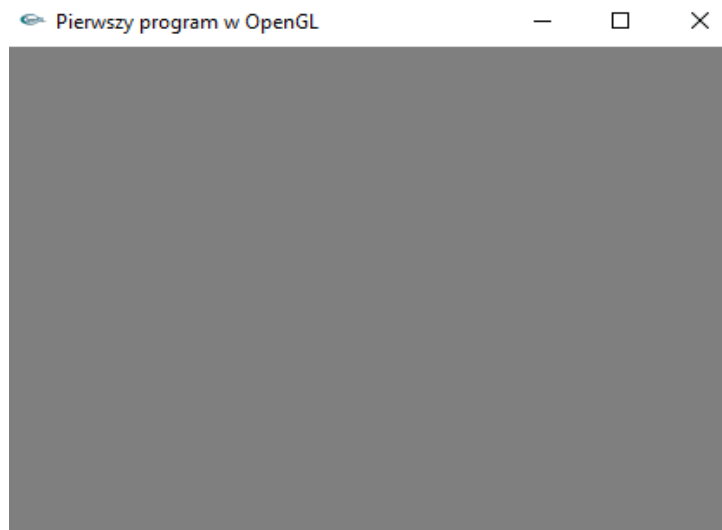
```

// analogicznie dla pozostałych
glVertex2f(i - 50 + (rand() % 15)*deform - 15*deform,
j - 50 + (rand() % 15)*deform) - 15*deform;

//tworzenie deformacji przez losowa zmianę pozycji wierzchołka
//analogicznie dla pozostałych
glColor3f(rand() % 10 * 0.1, rand() % 10 * 0.1, rand() % 10 * 0.1);
glVertex2f(i + (rand() % 15)*deform - 15*deform, j - 50 +
(rand() % 15)*deform) - 15*deform;
glColor3f(rand() % 10 * 0.1, rand() % 10 * 0.1, rand() % 10 * 0.1);
glVertex2f(i + (rand() % 15)*deform - 15*deform, j +
(rand() % 15)*deform) - 15*deform;
glColor3f(rand() % 10 * 0.1, rand() % 10 * 0.1, rand() % 10 * 0.1);
glVertex2f(i - 50 + (rand() % 15)*deform - 15*deform, j +
(rand() % 15)*deform) - 15*deform;
glEnd(); // koniec deklaracji danej figury
    }
}
glFlush(); // przekazanie poleceń do wykonania
}

```

3 Rezultat prac



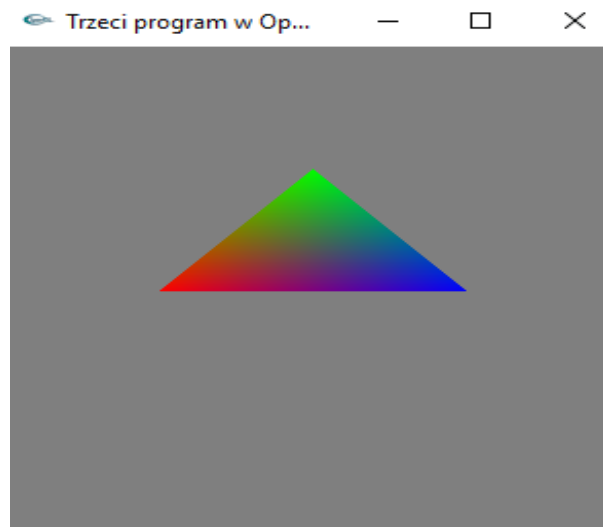
Rysunek 1: Stworzenie pierwszego programu



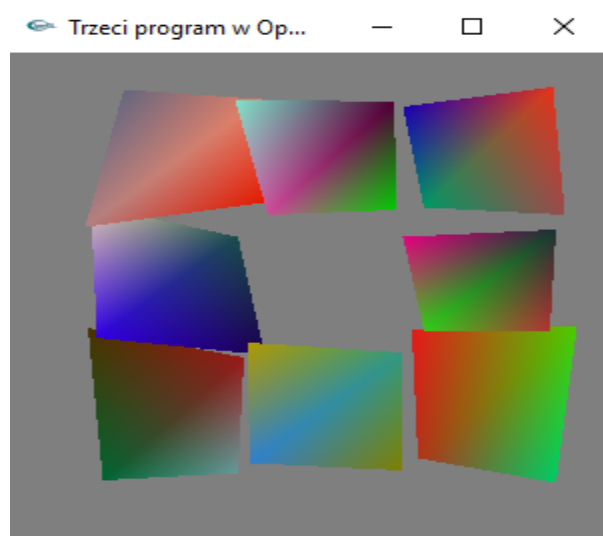
Rysunek 2: Wygenerowanie pierwszej figury



Rysunek 3: kolorowanie figury na konkretne kolory



Rysunek 4: kolorowanie RGB



Rysunek 5: kwadrat stworzony nierekurencyjnie z ośmiu mniejszych kwadratów, nieregularnych kształtem, z losowymi kolorami RGB