Grupa: IO gr.1	Ćwiczenie: Lab 1	Imię Nazwisko: Malwina Cieśla
Wizualizacja Danych		

Cel ćwiczenia:

Zapoznanie z programowaniem grafiki przy użyciu shader'ów. Przygotowanie projektu umożliwiającego rysowanie sceny z użyciem potoku renedowania zgodnego z OpenGL 3.3+.

Przebieg ćwiczenia:

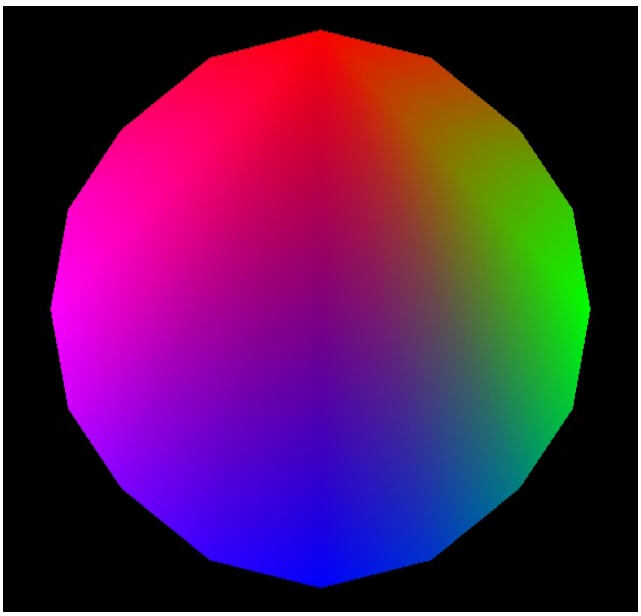
Wykonanie poszczególnych kroków opisanych w punkcie 4 w instrukcji pozwoliło na utworzenie okna oraz zapewnienie interakcji z użytkownikiem. Do tego należało wykorzystać bibliotekę SFML - Simple and Fast Multimedia Library oraz biblioteki GLEW (The OpenGL Extension Wrangler Library).
W moim zadaniu stworzyłam koło składające się z 16 wierzchołków opisanych w tablicy vertices. Każdy wierzchołek opisany jest 5 parametrami: dwa pierwsze odpowiadają za rozmieszczenie wierzchołka, a trzy pozostałe za kolor (RGB). Dodatkowo w każdym wierzchołku mojej figury ustawiłam inny kolor RGB, aby stworzyć figurę wielokolorową. Poniżej przedstawiam tablicę vertices oraz otrzymane koło:

```
// Narysowanie kolorowego koła
glDrawArrays(GL_POLYGON, 0, 16);
```

Ilustracja 1: Fragment kodu

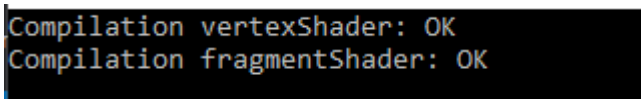
```
GLfloat vertices[] = {
0.0f, 0.7f, 1.0f, 0.0f, 0.0f,
0.25f, 0.63f, 0.8f, 0.2f, 0.0f,
0.45f, 0.45f, 0.5f, 0.5f, 0.0f,
0.57f, 0.25f, 0.2f, 0.8f, 0.0f,
0.61f, 0.0f, 0.0f, 1.0f, 0.0f,
0.57f, -0.25f, 0.0f, 0.8f, 0.2f,
0.45f, -0.45f, 0.0f, 0.5f, 0.5f,
0.25f, -0.63f, 0.0f, 0.2f, 0.8f,
0.0f, -0.7f, 0.0f, 0.0f, 1.0f,
-0.25f, -0.63f, 0.2f, 0.0f, 1.0f,
-0.45f, -0.45f, 0.5f, 0.0f, 1.0f,
-0.57f, -0.25f, 0.8f, 0.0f, 1.0f,
-0.61f, 0.0f, 1.0f, 0.0f, 1.0f,
-0.57f, .25f, 1.0f, 0.0f, 0.8f,
-0.45f, 0.45f, 1.0f, 0.0f, 0.5f,
-0.25f, 0.63f, 1.0f, 0.0f, 0.2f
};
```

Ilustracja 3: Tablica vertices



Ilustracja 2: Otrzymana figura

Następnie dodałam kod sprawdzający poprawność kompilacji shader'ów oraz wyświetlający odpowiedni komunikat w konsoli łącznie z kodem ewentualnego błędu.
Po włączeniu konsoli w terminalu pojawia się komunikat:



Ilustracja 4: Zawartość terminala

Fragment kodu sprawdzający poprawność kompilacji wraz z dostępnymi możliwościami (error) przedstawiam poniżej:

```
// Utworzenie i skompilowanie shadera wierzchołków
cout << "Compilation vertexShader: ";
GLuint vertexShader =
    glCreateShader(GL_VERTEX_SHADER);
    glShaderSource(vertexShader, 1, &vertexSource, NULL);
    glCompileShader(vertexShader);
    GLint status; glGetShaderiv(vertexShader, GL_COMPILE_STATUS, &status);
    if(status==GL_TRUE)
        cout << "OK" << endl;
    else {
        char buffer[512];
        glGetShaderInfoLog(vertexShader, 512, NULL, buffer);
    }

// Utworzenie i skompilowanie shadera fragmentów
cout << "Compilation fragmentShader: ";
GLuint fragmentShader =
    glCreateShader(GL_FRAGMENT_SHADER);
    glShaderSource(fragmentShader, 1, &fragmentSource, NULL);
    glCompileShader(fragmentShader);
    GLint status2;
    glGetShaderiv(fragmentShader, GL_COMPILE_STATUS, &status2);
    if (status2 == GL_TRUE)
        cout << "OK" << endl;
    else {
        char buffer[512];
        glGetShaderInfoLog(fragmentShader, 512, NULL, buffer);
    }
```

Ilustracja 5: Fragment kodu