

# SPRAWOZDANIE

Lab nr 2

Celem ćwiczeń była modyfikacja kodu aby zmienić figurę 2D z trójkąta na inną.

KOD:

```
// Nagłówki
#include "stdafx.h"
#include <GL/glew.h>
#include <SFML/Window.hpp>

// Kody shaderów
const GLchar* vertexSource = R"glsl(
#version 150 core
in vec2 position;
in vec3 color;
out vec3 Color;
void main(){
Color = color;
gl_Position = vec4(position, 0.0, 1.0);
}
)glsl";

const GLchar* fragmentSource = R"glsl(
#version 150 core
in vec3 Color;
out vec4 outColor;
void main()
{
outColor = vec4(Color, 1.0);
}
)glsl";

int main()
{
    sf::ContextSettings settings;
    settings.depthBits = 24;
    settings.stencilBits = 8;
```

```
// Okno renderingu  
sf::Window window(sf::VideoMode(800, 600, 32), "OpenGL", sf::Style::Titlebar |  
sf::Style::Close, settings);
```

```
// Inicjalizacja GLEW  
glewExperimental = GL_TRUE;  
glewInit();
```

```
// Utworzenie VAO (Vertex Array Object)  
GLuint vao;  
glGenVertexArrays(1, &vao);  
glBindVertexArray(vao);
```

```
// Utworzenie VBO (Vertex Buffer Object)  
// i skopiowanie do niego danych wierzchołkowych  
GLuint vbo;  
glGenBuffers(1, &vbo);
```

```
GLfloat vertices[] = {  
0.5f, 0.5f, 1.0f, 0.0f, 0.0f,  
0.5f, -0.5f, 0.0f, 1.0f, 0.0f,  
-0.5f, -0.5f, 0.0f, 0.0f, 1.0f,  
-0.5f, 0.5f, 1.0f, 0.0f, 0.0f  
};  
glBindBuffer(GL_ARRAY_BUFFER, vbo);  
glBufferData(GL_ARRAY_BUFFER, sizeof(vertices), vertices, GL_STATIC_DRAW);
```

```
// Utworzenie i skompilowanie shadera wierzchołków  
GLuint vertexShader =  
    glCreateShader(GL_VERTEX_SHADER);  
glShaderSource(vertexShader, 1, &vertexSource, NULL);  
glCompileShader(vertexShader);
```

```
// Utworzenie i skompilowanie shadera fragmentów  
GLuint fragmentShader =  
    glCreateShader(GL_FRAGMENT_SHADER);  
glShaderSource(fragmentShader, 1, &fragmentSource, NULL);  
glCompileShader(fragmentShader);
```

```
// Zlinkowanie obu shaderów w jeden wspólny program
GLuint shaderProgram = glCreateProgram();
glAttachShader(shaderProgram, vertexShader);
glAttachShader(shaderProgram, fragmentShader);
glBindFragDataLocation(shaderProgram, 0, "outColor");
glLinkProgram(shaderProgram);
glUseProgram(shaderProgram);

// Specyfikacja formatu danych wierzchołkowych
GLint posAttrib = glGetAttribLocation(shaderProgram, "position");
glEnableVertexAttribArray(posAttrib);
glVertexAttribPointer(posAttrib, 2, GL_FLOAT, GL_FALSE, 5 * sizeof(GLfloat), 0);
GLint colAttrib = glGetAttribLocation(shaderProgram, "color");
glEnableVertexAttribArray(colAttrib);
glVertexAttribPointer(colAttrib, 3, GL_FLOAT, GL_FALSE, 5 * sizeof(GLfloat), (void*)(2
* sizeof(GLfloat)));

// Rozpoczęcie pętli zdarzeń
bool running = true;
while (running) {
    sf::Event windowEvent;
    while (window.pollEvent(windowEvent)) {
        switch (windowEvent.type) {
            case sf::Event::Closed:
                running = false;
                break;
        }
    }
    // Nadanie scenie koloru czarnego
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);
    glClear(GL_COLOR_BUFFER_BIT);

    // Narysowanie trójkąta
    //ta na podstawie 3 wierzchołków
    glDrawArrays(GL_POLYGON, 0, 4);
    // Wymiana buforów tylni/przedni
    window.display();
}
// Kasowanie programu i czyszczenie buforów
glDeleteProgram(shaderProgram);
glDeleteShader(fragmentShader);
```

```
glDeleteShader(vertexShader);  
glDeleteBuffers(1, &vbo);  
glDeleteVertexArrays(1, &vao);  
// Zamknięcie okna renderingu  
window.close();  
return 0;  
}
```

1. Dodałam współrzędne jednego, czwartego wierzchołka do tablicy vertices.

```
GLfloat vertices[] = {  
    0.5f, 0.5f, 1.0f, 0.0f, 0.0f,  
    0.5f, -0.5f, 0.0f, 1.0f, 0.0f,  
    -0.5f, -0.5f, 0.0f, 0.0f, 1.0f,  
    -0.5f, 0.5f, 1.0f, 0.0f, 0.0f  
};
```

2. Zmieniłam rysowanie trójkąta na rysowanie wielokąta, przekazałam w funkcji 4 wierzchołki.

z `glDrawArrays(GL_TRIANGLE, 0, 3);`  
na `glDrawArrays(GL_POLYGON, 0, 4);`

Powstała figura:

