

SPRAWOZDANIE

Zajęcia: Grafika komputerowa

Prowadzący: mgr inż. Mikołaj Grygiel

Laboratorium 7

26.02.2025

Temat: "Tekstury w OpenGL"

Wariant 6

Bartłomiej Mędrzak

s61324

Informatyka I stopień,

stacjonarne,

4 semestr,

Gr.1A

1. Polecenie:

Celem jest teksturowanie piramidy z użyciem dwóch sposobów ładowania tekstur: użycie tekstury z buforu kolorów (rysowanie w Panel); ładowanie tekstury z pliku (trzy pliki przykładowe do pobrania).

2. Wprowadzane dane:

Do istniejącego kodu dodałem piramidę o podstawie dziesięciokątnej oraz teksturę ziemi.

3. Wykorzystane komendy:

Bufor kolorów :

```
function drawCustomPyramid() {
    var pyramidApex = [0, 1.05, 0];
    var basePoints = [];
    var baseSideCount = 10;
    var baseRadius = 1.0;

    for (var i = 0; i < baseSideCount; i++) {
        var angle = (2 * Math.PI * i) / baseSideCount;
        basePoints.push([baseRadius * Math.cos(angle), -1.0, baseRadius * Math.sin(angle)]);
    }

    glBegin(GL_TRIANGLES);
    for (var i = 0; i < baseSideCount; i++) {
        var pt1 = basePoints[i];
        var pt2 = basePoints[(i + 1) % baseSideCount];

        var nX = (pt1[0] + pt2[0]) * 0.5;
        var nY = 2.0;
        var nZ = (pt1[2] + pt2[2]) * 0.5;
        glNormal3d(nX, nY, nZ);

        glTexCoord2f(0.5, 1.0); glVertex3dv(pyramidApex);
        glTexCoord2f(0.0, 0.0); glVertex3dv(pt1);
        glTexCoord2f(1.0, 0.0); glVertex3dv(pt2);
    }
    glEnd();

    glBegin(GL_POLYGON);
    glNormal3d(0, -1.0, 0);
    for (var i = 0; i < baseSideCount; i++) {
        var vert = basePoints[i];
        var texCoordU = (vert[0] / baseRadius + 1.0) / 2.0;
        var texCoordV = (vert[2] / baseRadius + 1.0) / 2.0;
        glTexCoord2f(texCoordU, texCoordV);
        glVertex3dv(vert);
    }
    glEnd();
}

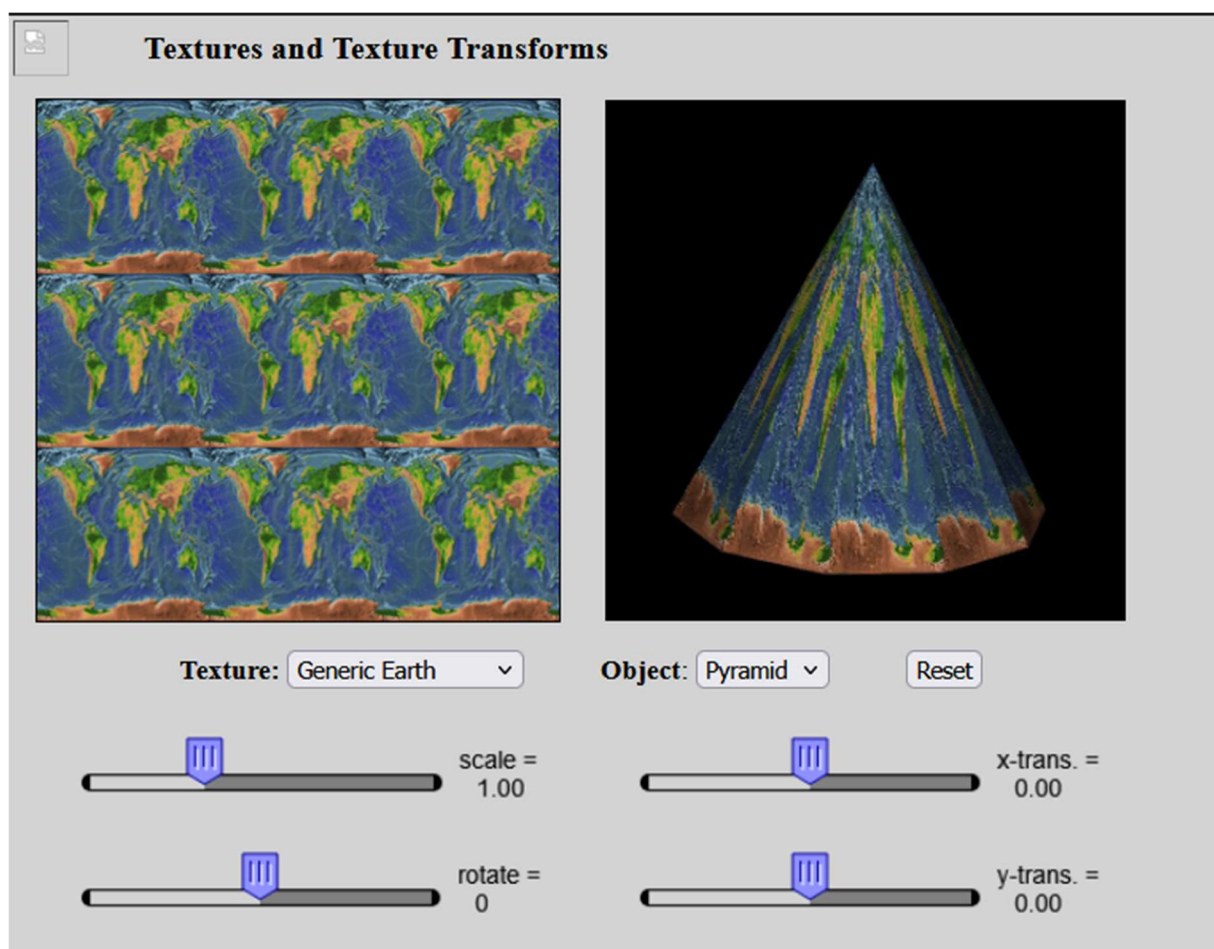
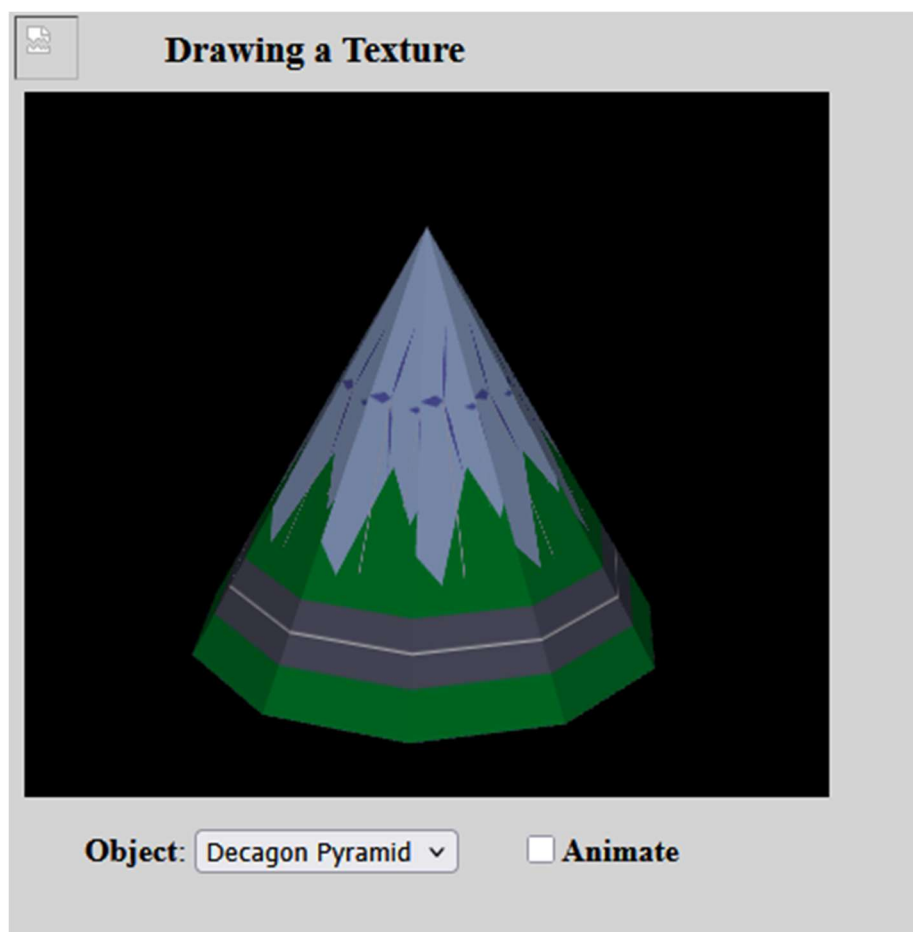
function initGL() {
    glEnable(GL_LIGHT0);
    glEnable(GL_NORMALIZE);
    glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE, [ 1, 1, 1, 1 ]); // white material for texturing.
}
```

Tekstura:

```
function drawPyramid() {  
    var pyramidApexCoords = [0, 1.3, 0];  
    var basePolygonPoints = [];  
    var numberOfBaseSides = 10;  
    var radiusOfBase = 1.2;  
  
    for (var i = 0; i < numberOfBaseSides; i++) {  
        var angleForVertex = (2 * Math.PI * i) / numberOfBaseSides;  
        basePolygonPoints.push([  
            radiusOfBase * Math.cos(angleForVertex),  
            -1.01,  
            radiusOfBase * Math.sin(angleForVertex)  
        ]);  
    }  
  
    glBegin(GL_TRIANGLES);  
    for (var i = 0; i < numberOfBaseSides; i++) {  
        var p1 = basePolygonPoints[i];  
        var p2 = basePolygonPoints[(i + 1) % numberOfBaseSides];  
  
        var normX = (p1[0] + p2[0]) * 0.52;  
        var normY = 1.05;  
        var normZ = (p1[2] + p2[2]) * 0.52;  
        glNormal3d(normX, normY, normZ);  
  
        glTexCoord2f(0.5, 1.0); glVertex3dv(pyramidApexCoords);  
        glTexCoord2f(0.0, 0.0); glVertex3dv(p1);  
        glTexCoord2f(1.0, 0.0); glVertex3dv(p2);  
    }  
    glEnd();  
  
    glBegin(GL_POLYGON);  
    glNormal3d(0, -1, 0);  
    for (var i = 0; i < numberOfBaseSides; i++) {  
        var baseVtx = basePolygonPoints[i];  
        var texU = (baseVtx[0] / radiusOfBase + 1.0) / 2.0;  
        var texV = (baseVtx[2] / radiusOfBase + 1.0) / 2.0;  
        glTexCoord2f(texU, texV);  
        glVertex3dv(baseVtx);  
    }  
    glEnd();  
}
```

https://github.com/castehard33/Grafika_Komputerowa/tree/main/7%20Tekstury%20w%20OpenGL

4. Wynik działania:



5. Wnioski:

Laboratorium udowadnia, że OpenGL oferuje elastyczne podejścia do teksturowania, umożliwiając zarówno dynamiczne generowanie tekstur z bufora kolorów (co pozwala na animowane lub proceduralnie tworzone obrazy), jak i efektywne wykorzystanie gotowych map tekstur wczytywanych bezpośrednio z plików graficznych.