



Uniwersytet Bielsko-Bialski

Sprawozdanie

*Zajęcia: Grafika Komputerowa
(Ćwiczenia laboratoryjne)*

Prowadzący: prof dr. hab. Vasyl Martsenyuk

Laboratoria nr: 7

Temat ćwiczenia:

Zadanie_teksturyGL

Maksymilian Wójcik
Informatyka I stopnia
niestacjonarne
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. Wykorzystane komendy:

a)

```
var canvas; // the canvas on which we will draw.

var frameNumber = 0; // frame number for the animation

var sphere, cubeModel, cylinder, cone, torus, teapot, piramida; // model data for six objects.

/* The display function, which draws the content of the canvas.
 */
function draw() {
    var objectNumber = Number(document.getElementById("object").value); // which object to draw.

    /* First, draw the 2D scene, using a 256-by-256 viewport to get a power-of-two texture. */

    glDisable(GL_LIGHTING);
    glDisable(GL_DEPTH_TEST);
    glDisable(GL_TEXTURE_2D);
    glViewport(0, 0, 256, 256); // Note that canvas must be at least 256-by-256.
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    glOrtho(0, 7, -1, 5, -1, 1); // Limits required by 2D scene
    glMatrixMode(GL_MODELVIEW);

    draw2DScene();

    if (objectNumber == 7) {
        return; // Just show the 2D scene as the image in the canvas.
    }

    /* Copy the image into the texture. */

    glCopyTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, 0, 0, 256, 256, 0);

    /* Since we do not have mipmaps for the texture, we MUST set the texture
    minimization filter to GL_NEAREST or GL_LINEAR, since the default
    filter requires mipmaps. */
```

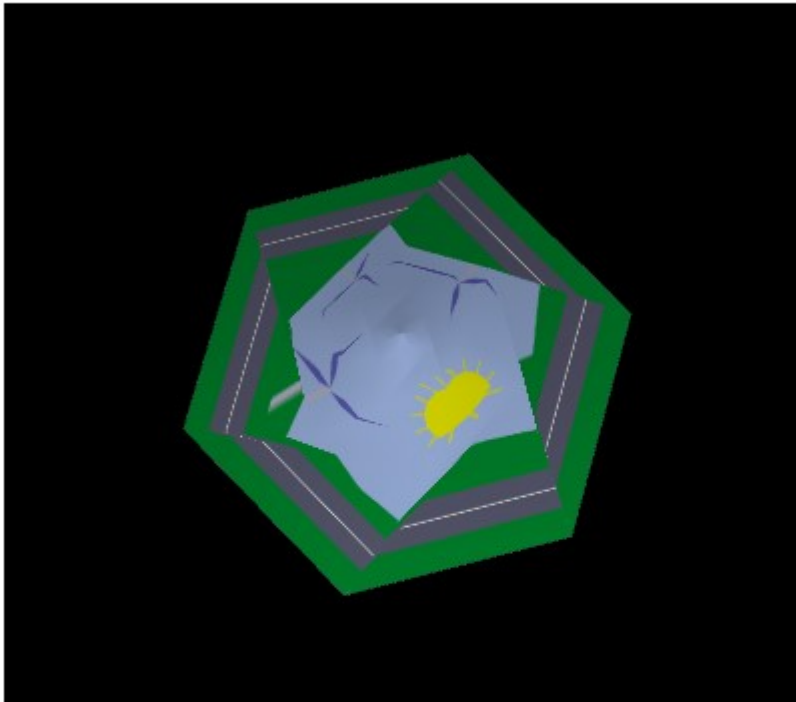
b)

```
function drawModel(model) {  
    glEnableClientState(GL_VERTEX_ARRAY);  
    glVertexPointer(3, GL_FLOAT, 0, model.vertexPositions);  
    glEnableClientState(GL_NORMAL_ARRAY);  
    glNormalPointer(GL_FLOAT, 0, model.vertexNormals);  
    glEnableClientState(GL_TEXTURE_COORD_ARRAY);  
    glTexCoordPointer(2, GL_FLOAT, 0, model.vertexTextureCoords);  
    glDrawElements(GL_TRIANGLES, model.indices.length, GL_UNSIGNED_BYTE, model.indices);  
    glDisableClientState(GL_VERTEX_ARRAY);  
    glDisableClientState(GL_NORMAL_ARRAY);  
    glDisableClientState(GL_TEXTURE_COORD_ARRAY);  
}  
  
function initGL() {  
    glEnable(GL_LIGHTING);  
    glEnable(GL_LIGHT0);  
    glEnable(GL_NORMALIZE);  
    glEnable(GL_DEPTH_TEST);  
    glMaterialfv(GL_FRONT_AND_BACK, GL_AMBIENT_AND_DIFFUSE, [ 1, 1, 1, 1 ]); // white, for texturing  
    glClearColor(0,0,0,1);  
}
```

3. Wynik działania:

a)

Drawing a Texture



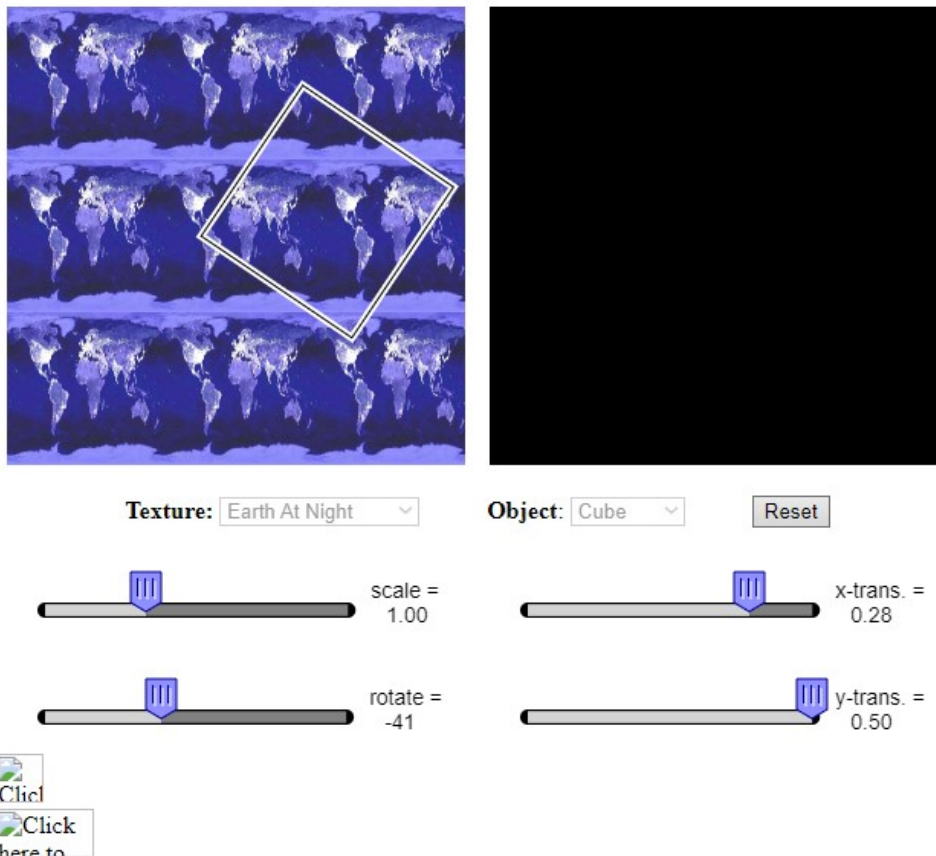
Object: Piramida ☐ Animate



b)

Can't access texture.

Note: Some browsers can't use a file from a local disk.



4.Wnioski

Piramida: Teksturowana piramida, gdzie każda powierzchnia jest pokryta teksturą.

Podstawa: Podstawa piramidy również pokryta teksturą, tworząca bardziej realistyczny wygląd.

Interakcja użytkownika: Możliwość obracania podstawy piramidy wokół osi Y za pomocą przeciągania myszką.

Teksturowanie: Ładowanie tekstur z plików i stosowanie ich do obiektów w scenie 3D.

WebGL: Użycie WebGL do renderowania grafiki 3D w przeglądarce, co pozwala na tworzenie interaktywnych aplikacji webowych z teksturami i oświetleniem.

Kod źródłowy: <https://github.com/mwojcik123/UBB-GK-MW>