



# Uniwersytet Bielsko-Bialski

## Sprawozdanie

*Zajęcia: Grafika Komputerowa  
(Ćwiczenia laboratoryjne)*

*Prowadzący: prof dr. hab. Vasyl Martsenyuk*

**Laboratoria nr: 6**

**Temat ćwiczenia:**

**Zadanie\_Swiatlo**

Maksymilian Wójcik  
Informatyka I stopnia  
niestacjonarne  
4 semestr  
gr. 1A

## **1.Polecenie:**

Celem jest stworzenie piramidy z użyciem różnych materiałów określonych wariantem zadania i umieszczenie jej na „podstawie”. Użytkownik może obracać podstawę wokół osi Y, przeciągając mysz w poziomie. Scena wykorzystuje globalne światło otoczenia (ambient) oraz źródło światła o kształcie kuli z możliwością animacji obrotu wokół piramidy.

Aby wykonać laboratorium w JavaScript polecane jest zapoznanie z plikami .html: `four-lights-demo.html` oraz `materials-demo.html`

## **2. Wykorzystane komendy:**

a)

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>rotating objects example</title>
    <link rel="stylesheet" href="../demo.css" />
    <script src="../script/demo-core.js"></script>
    <script src="../script/glsim.js"></script>
    <script>
      let cam;
      let animateCheckbox;
      let baseCheckbox;
      let ambientLightCheckbox;
      let viewpointLightCheckbox;
      let whiteLightCheckbox;
      let isAnimating = false;
      let frameCount = 0;

      function sphere(radius, slices, stacks) {
        for (let j = 0; j < stacks; j++) {
          let lat1 = (Math.PI / stacks) * j - Math.PI / 2;
          let lat2 = (Math.PI / stacks) * (j + 1) - Math.PI / 2;
          let sinLat1 = Math.sin(lat1);
          let cosLat1 = Math.cos(lat1);
          let sinLat2 = Math.sin(lat2);
          let cosLat2 = Math.cos(lat2);
          glBegin(GL_TRIANGLE_STRIP);
          for (let i = 0; i <= slices; i++) {
            let lon = ((2 * Math.PI) / slices) * i;
            let sinLon = Math.sin(lon);
            let cosLon = Math.cos(lon);
            let x1 = cosLon * cosLat1;
            let y1 = sinLon * cosLat1;
            let z1 = sinLat1;
            let x2 = cosLon * cosLat2;
            let y2 = sinLon * cosLat2;
            let z2 = sinLat2;
            glNormal3d(x2, y2, z2);
            glVertex3d(radius * x2, radius * y2, radius * z2);
            glNormal3d(x1, y1, z1);
            glVertex3d(radius * x1, radius * y1, radius * z1);
          }
          glEnd();
        }
      }

      function setLights() {

```

b)

```

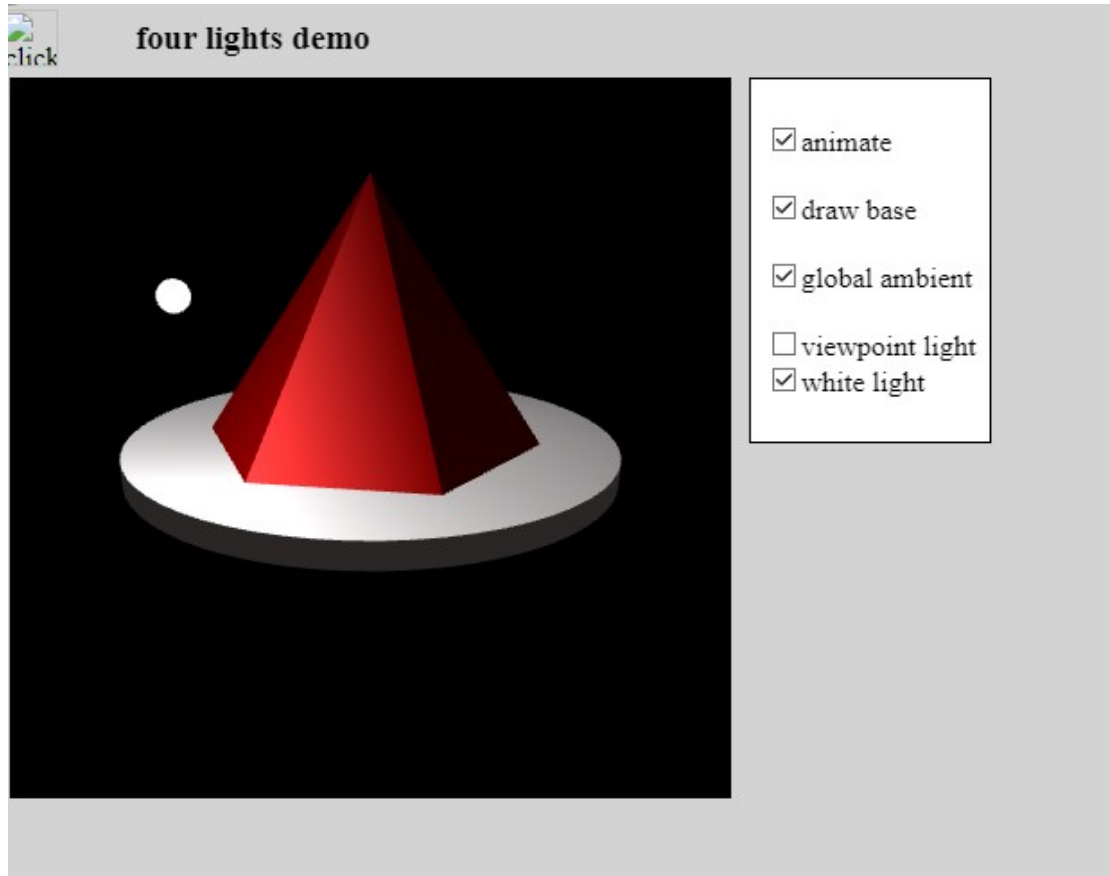
}

function init() {
    try {
        glsimUse("maincanvas");
    }
    catch (e) {
        document.getElementById("canvas-holder").innerHTML="<p><b>Sorry, an error occurred:<br>" +
            e + "</b></p>";
        return;
    }
    document.getElementById("modelSelect").value = "Teapot";
    document.getElementById("modelSelect").onchange = newModel;
    document.getElementById("materialSelect").value = "0";
    document.getElementById("materialSelect").onchange = newMaterial;
    document.getElementById("reset").onclick = function() { camera.lookAt(2,2,5); draw() };
    slidersDiffuse = new SliderCanvas(document.getElementById("diffuse-sliders"));
    slidersDiffuse.addSlider({label:"Red", max:1, step:0.001, value:0.6, decimals:3});
    slidersDiffuse.addSlider({label:"Green", max:1, step:0.001, value:0.6, decimals:3});
    slidersDiffuse.addSlider({label:"Blue", max:1, step:0.001, value:0.6, decimals:3});
    slidersDiffuse.draw();
    slidersSpecular = new SliderCanvas(document.getElementById("specular-sliders"));
    slidersSpecular.addSlider({label:"Red", max:1, step:0.001, value:1, decimals:3});
    slidersSpecular.addSlider({label:"Green", max:1, step:0.001, value: 1, decimals:3});
    slidersSpecular.addSlider({label:"Blue", max:1, step:0.001, value: 1, decimals:3});
    slidersSpecular.draw();
    slidersShininess = new SliderCanvas(document.getElementById("shininess-slider"));
    slidersShininess.addSlider({min:0, max:128, value:16});
    slidersShininess.draw();
    slidersDiffuse.onChange = sliderChange;
    slidersSpecular.onChange = sliderChange;
    slidersShininess.onChange = sliderChange;
    initGL();
    camera = new Camera();
    camera.setScale(0.8);
    camera.lookAt(2,2,5);
    draw();
    camera.installTrackball(draw);
}
</script>

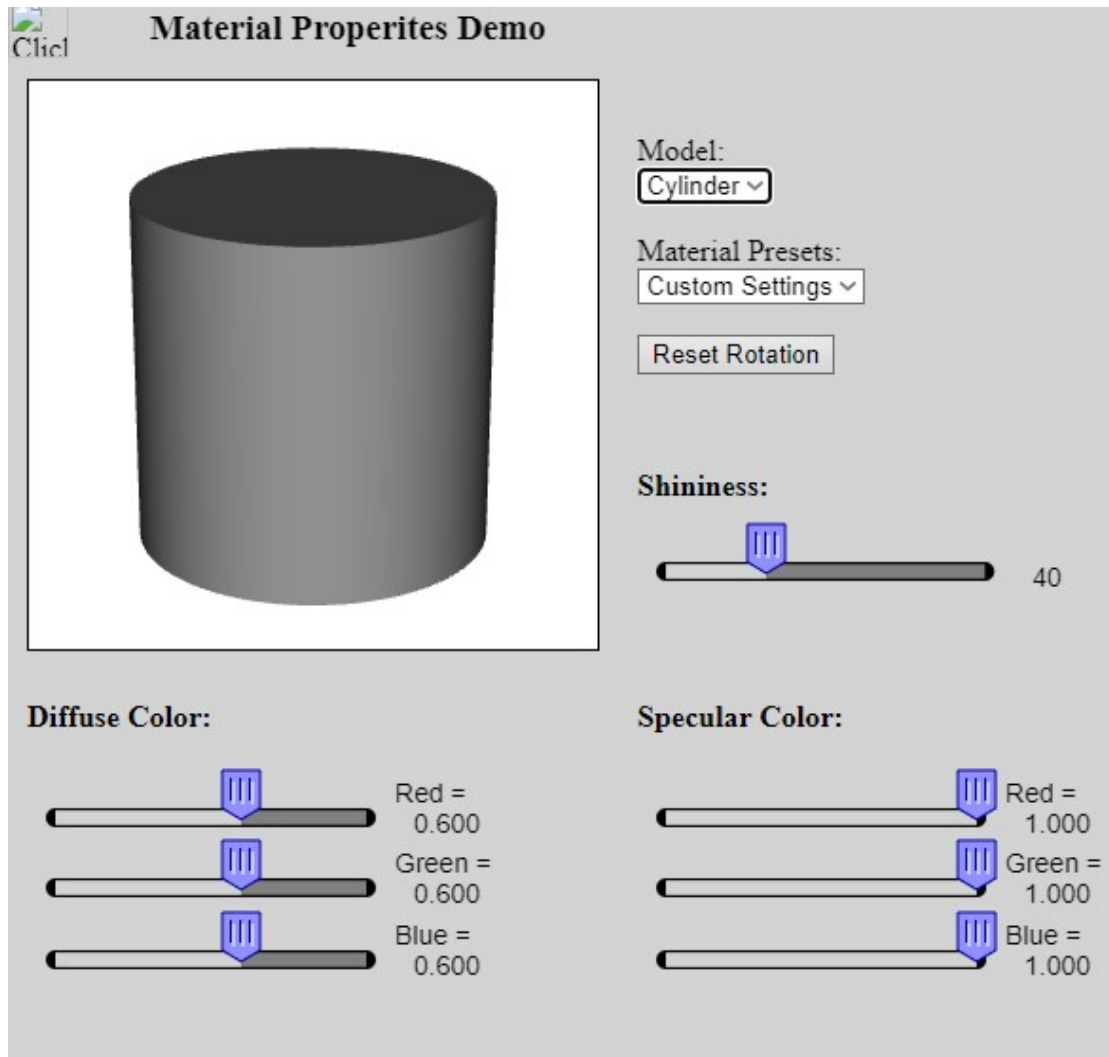
```

### 3. Wynik działania:

a)



b)



## 4.Wnioski

**Piramida:** Użycie trójkątów do stworzenia piramidy i przypisanie różnych kolorów każdemu bokowi.

**Podstawa:** Prostokątna podstawa z jednolitym kolorem.

**Oświetlenie:** Zastosowanie globalnego światła otoczenia (ambient) oraz kierunkowego źródła światła dla realistycznego wyglądu.

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

**WebGL:** Użycie WebGL do renderowania grafiki 3D w przeglądarce, co pozwala na tworzenie interaktywnych aplikacji webowych.

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