



Uniwersytet Bielsko-Bialski

Sprawozdanie

*Zajęcia: Grafika Komputerowa
(Ćwiczenia laboratoryjne)*

Prowadzący: prof dr. hab. Vasyl Martsenyuk

Laboratoria nr: 5

Temat ćwiczenia:

Zadanie_OpenGL1

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

1. Polecenie:

Stworzyć dwa obiekty przy użyciu OpenGL (w języku JavaScript). Po uruchomieniu zakończonego programu naciśnięcie jednego z klawiszy numerycznych 1 lub 2 spowoduje wybranie wyświetlanego obiektu. Program ustawia wartość zmiennej globalnej, `objectNumber`, aby powiedzieć, który obiekt ma zostać narysowany. Użytkownik może obracać obiekt za pomocą klawiszy strzałek, `PageUp`, `PageDown` i `Home`. Podprogram `display()` jest wywoływany, aby narysować obiekt.

Obiekt 1. Korkociąg wokół osi $\{x \mid y \mid z\}$ zawierający N obrotów. Punkty są stopniowo powiększane. Ustalić aktualny kolor rysujący na $\{\text{zielony} \mid \text{niebieski} \mid \text{brązowy} \mid \dots\}$.

Obiekt 2. Pyramida, wykorzystując dwa wachlarze trójkątów oraz modelowanie hierarchiczne (najpierw tworzymy podprogramę rysowania jednego trójkąta; dalej wykorzystując przekształcenia geometryczne tworzymy piramidę). Podstawą piramidy jest wielokąt o N wierzchołkach.

2. Wykorzystane komendy:

a)

```

function drawPyramid(numSides) {
  glBegin(GL_TRIANGLES);

  for (let i = 0; i < numSides; ++i) {
    const angle1 = (i * 2 * Math.PI) / numSides;
    const angle2 = ((i + 1) * 2 * Math.PI) / numSides;

    glColor3f(randomColor(), randomColor(), randomColor());
    glVertex3f(0, 0, 1);
    glVertex3f(Math.cos(angle1), Math.sin(angle1), 0);
    glVertex3f(Math.cos(angle2), Math.sin(angle2), 0);
  }

  glColor3f(1, 0, 0);
  for (let i = 0; i < numSides; ++i) {
    glVertex3f(0, 0, 0);
    glVertex3f(
      Math.cos((i * 2 * Math.PI) / numSides),
      Math.sin((i * 2 * Math.PI) / numSides),
      0
    );
    glVertex3f(
      Math.cos(((i + 1) * 2 * Math.PI) / numSides),
      Math.sin(((i + 1) * 2 * Math.PI) / numSides),
      0
    );
  }
  glEnd();
}

```

```

function drawSpiral(numSegments) {
  // zmiana koloru spirali na brązowy
  glColor3f(139 / 255, 69 / 255, 19 / 255);
  glBegin(GL_LINE_STRIP);
  const spiralAngle = Math.PI / (numSegments * 0.5);
  for (let i = 0; i < numSegments * numSegments; i++) {
    const x = Math.cos(spiralAngle * i) * (0.03 * i);
    const y = i / (2 * numSegments);
    const z = Math.sin(spiralAngle * i) * (0.03 * i);
    glVertex3d(x, y, z);
  }
  glEnd();
}

```

3. Wynik działania:

a)

rotating objects in opengl 1.1

(rotate using arrow keys, page up, page down, and home keys.)



rotating objects in opengl 1.1

(rotate using arrow keys, page up, page down, and home keys.)



4.Wnioski

Korkociąg: Użycie linii pozwala na łatwe rysowanie i animowanie korkociągu z punktami zmieniającymi się stopniowo.

Piramida: Modelowanie hierarchiczne umożliwia złożenie piramidy z wachlarzy trójkątów, co zapewnia modularność i elastyczność.

WebGL: Biblioteka WebGL w JavaScript umożliwia tworzenie interaktywnych scen 3D, które działają bezpośrednio w przeglądarce.

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