Grafika komputerowa i komunikacja człowiek komputer

Laboratorium 5

Teksturowanie powierzchni obiektów

Wykonał:

Ivan Hancharyk 264511

Termin zajęć:

WT/TP/8:00

1. Cel ćwiczenia

Celem ćwiczenia jest zapoznanie się z podstawowymi technikami teksturowania powierzchni obiektów płaskich i trójwymiarowych z wykorzystaniem mechanizmów biblioteki OpenGL z rozszerzeniem GLUT.

2. Wstęp teoretyczny

Teksturowanie jest jedną z podstawowych technik stosowanych w grafice komputerowej do nadania realizmu generowanych obiektów. Polega ono na nakładaniu obrazów, zwanych teksturami, na powierzchnię obiektów 2D lub 3D. Dzięki teksturowaniu można symulować różnorodne materiały i szczegóły, takie jak drewno, metal, tkaniny, czy powierzchnie organiczne, bez konieczności zwiększania złożoności geometrycznej obiektu. Tekstury definiowane są jako dwuwymiarowe mapy bitowe, które mogą być przechowywane w różnych formatach plików, np. TGA. Format TGA (Truevision Graphics Adapter) jest popularny w aplikacjach graficznych dzięki swojej prostocie i wsparciu dla wielu trybów kolorystycznych, w tym 24-bitowego koloru RGB i 32-bitowego RGBA z kanałem alfa.

W OpenGL proces teksturowania obejmuje kilka kluczowych etapów. Po pierwsze, tekstura musi zostać załadowana do pamięci za pomocą funkcji takich jak **LoadTGAImage**. Funkcja ta odczytuje dane obrazu i udostępnia szerokość, wysokość oraz format tekstury, umożliwiając jej dalsze przetwarzanie. Następnie, za pomocą funkcji **glTexImage2D**, tekstura jest definiowana jako obiekt OpenGL i wiązana z powierzchnią obiektu docelowego. Współrzędne tekstury, przypisywane każdemu wierzchołkowi obiektu, określane są za pomocą funkcji **glTexCoord2f**. Dzięki temu OpenGL może interpolować teksturę między wierzchołkami podczas renderowania.

Proces teksturowania wymaga także zdefiniowania trybu filtrowania i parametrów nakładania tekstur. Filtry, takie jak **GL_LINEAR**, pozwalają na wygładzanie tekstur w przypadku zmiany ich rozmiarów, co zapobiega artefaktom wizualnym. Ponadto, funkcja **glTexEnvi** określa sposób, w jaki tekstura wpływa na kolor obiektu, oferując tryby takie jak **GL_MODULATE**, który łączy kolor tekstury z oświetleniem obiektu, oraz **GL_REPLACE**, który zastępuje kolor obiektu kolorami tekstury.

Teksturowanie ma zastosowanie w wielu dziedzinach, od gier komputerowych, przez wizualizacje architektoniczne, aż po efekty specjalne w filmach.

Współczesne techniki, takie jak mapowanie normalnych czy tekstury proceduralne, stanowią rozwinięcie tej podstawowej metody, umożliwiając tworzenie bardziej złożonych i realistycznych efektów wizualnych.

3. Realizacja zadania

- Wczytanie pliku TGA za pomocą funkcji **LoadTGAImage()**

```
pBytes = LoadTGAImage( FileName: "D1_t.tga", &ImWidth, &ImHeight, &ImComponents, &ImFormat);
```

- Ładowanie tekstury z pliku

```
void ApplyTexture(const char* filename) {
    GLbyte* pBytes;
    GLint ImWidth, ImHeight, ImComponents;
    GLenum ImFormat;

    pBytes = LoadTGAImage(filename, &ImWidth, &ImHeight, &ImComponents, &ImFormat);
    if (!pBytes) {
        cout << "Nie można załadować pliku tekstury: " << filename << endl;
        return;
    }

    glTexImage2D( target: GL_TEXTURE_2D, level: 0, internalformat: ImComponents, ImWidth, ImHeight, border: 0, ImFormat, type: GL_UNSIGNED_BYTE, pBytes);
    free(pBytes);
    glEnable( cap: GL_TEXTURE_2D);
    glTexEnvi( target: GL_TEXTURE_ENV, pname: GL_TEXTURE_ENV_MODE, param: GL_MODULATE);
    glTexParameteri( target: GL_TEXTURE_2D, pname: GL_TEXTURE_MIN_FILTER, param: GL_LINEAR);
    glTexParameteri( target: GL_TEXTURE_2D, pname: GL_TEXTURE_MAG_FILTER, param: GL_LINEAR);
}
```

Funkcja **ApplyTexture** odpowiada za ładowanie tekstury z pliku .tga i ustawienie jej w OpenGL. Po wybraniu pliku, funkcja korzysta z **LoadTGAImage**, aby załadować dane tekstury i uzyskać jej szerokość, wysokość, format i komponenty. Następnie za pomocą glTexImage2D przesyła teksturę do pamięci GPU, gdzie zostaje przypisana do bieżącego obiektu. Funkcja ustawia parametry teksturowania, takie jak filtracja (np. liniowa) i sposób nakładania tekstury (np. GL_MODULATE), oraz aktywuje teksturowanie za pomocą glEnable(GL_TEXTURE_2D).

- Teksturowanie jajka

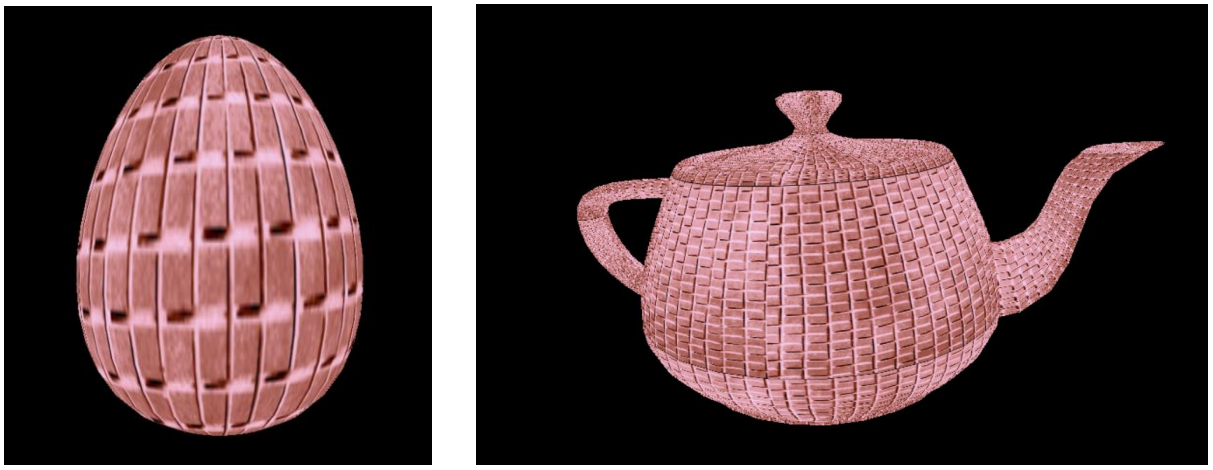
```
void Egg() {
    Points();

    for (int i = 0; i < N - 1; i++) {
        for (int j = 0; j < N - 1; j++) {
            glBegin( mode GL_TRIANGLES);
            glNormal3fv(vectors[i][j]);
            glTexCoord2f( s: (float) i / (N-1),  t: (float)j / (N - 1));
            glVertex3f(points[i][j].x, points[i][j].y, points[i][j].z);
            glNormal3fv(vectors[i + 1][j]);
            glTexCoord2f( s: (float)(i+1) / (N - 1),  t: (float)j / (N - 1));
            glVertex3f(points[i + 1][j].x, points[i + 1][j].y, points[i + 1][j].z);
            glNormal3fv(vectors[i + 1][j + 1]);
            glTexCoord2f( s: (float)(i+1) / (N - 1),  t: (float)(j+1) / (N - 1));
            glVertex3f(points[i + 1][j + 1].x, points[i + 1][j + 1].y, points[i + 1][j + 1].z);
            glEnd();

            glBegin( mode GL_TRIANGLES);
            glNormal3fv(vectors[i][j]);
            glTexCoord2f( s: (float)i / (N - 1),  t: (float)j / (N - 1));
            glVertex3f(points[i][j].x, points[i][j].y, points[i][j].z);
            glNormal3fv(vectors[i][j + 1]);
            glTexCoord2f( s: (float)i / (N - 1),  t: (float)(j+1) / (N - 1));
            glVertex3f(points[i][j + 1].x, points[i][j + 1].y, points[i][j + 1].z);
            glNormal3fv(vectors[i + 1][j + 1]);
            glTexCoord2f( s: (float)(i + 1) / (N - 1),  t: (float)(j + 1) / (N - 1));
            glVertex3f(points[i + 1][j + 1].x, points[i + 1][j + 1].y, points[i + 1][j + 1].z);
            glEnd();
        }
    }
}
```

Funkcja **Egg** zajmuje się generowaniem i renderowaniem powierzchni jajka z nałożoną teksturą. W pierwszej kolejności wywołuje funkcję Points, która oblicza współrzędne wierzchołków siatki punktów oraz wektory normalne dla każdego wierzchołka. Następnie funkcja rysuje trójkąty, które tworzą powierzchnię jajka. W każdym wierzchołku trójkąta przypisywane są współrzędne tekstury za pomocą glTexCoord2f, współrzędne w przestrzeni trójwymiarowej za pomocą glVertex3f, a także wektory normalne, które odpowiadają za poprawne oświetlenie. Na koniec cała siatka jest renderowana, tworząc gładką, trójwymiarową powierzchnię jajka z teksturą.

Efekt działania programu:



Teksturowane jajko i czajnik

4. Wnioski

Podczas ćwiczeń udało się zapoznać z techniką teksturowania w OpenGL oraz zastosować ją w praktyce na przykładzie obiektów 3D. Kluczowym aspektem było zrozumienie procesu ładowania tekstur, definiowania ich parametrów oraz przypisywania współrzędnych tekstur do wierzchołków.