

컴퓨터그래픽스

HW03 Texture Mapping



담당 교수님 : 최민규 교수님

제 출 날 짜 : 2018. 12. 14.

학 과 : 컴퓨터소프트웨어학과

학 번 : 2014707040

이 름 : 유 진 혁

구현 방법 및 결과 화면

■ Draw a textured, shaded torus

- 구현 방법

```
// Render a torus
void RenderTorus()
{
    // Clear the window with current clearing color
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    gluLookAt(0.6, 0.8, 1, 0, 0, 0, 0.1, 1, 0);

    DrawAxis();
}
```

RenderTorus 함수는 화면에 그려지는 함수로 등록되어 있다. 과제 자료의 예시처럼, 카메라의 위치 벡터를 이동시키고 세 축을 그렸다.

```
// Use white as material
glColor3f(1, 1, 1);
glEnable(GL_COLOR_MATERIAL);
glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);

// Turn on the light
glEnable(GL_LIGHTING);
glEnable(GL_LIGHT0);

// Start drawing with the specified texture
LoadTexture();
glEnable(GL_TEXTURE_2D);
```

그리고 흰색을 재질로 하여 다른 색의 개입 없이 온전히 texture의 이미지가 torus를 덮도록 하였다. GL_LIGHT0의 default 조명을 켜고 LoadTexture 함수로 texture 이미지를 가져온 뒤, texture 모드를 활성화하였다.

```
void LoadTexture()
{
    // Open the texture image file
    ifstream fin;
    fin.open(fileName, ios::binary);
    for (int i = 0; i < 512; i++)
        for (int j = 0; j < 512; j++)
        {
            image[i][j][0] = fin.get();
            image[i][j][1] = fin.get();
            image[i][j][2] = fin.get();
        }
    fin.close();

    // Specify a texture image and parameters
    glTexImage2D(GL_TEXTURE_2D, 0, 3, 512, 512, 0, GL_RGB, GL_UNSIGNED_BYTE, image);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
}
```

LoadTexture 함수는 위와 같이 구현되어 있다. 문자열 변수 fileName에 저장된 파일명의 RAW 이미지를 열고 image 배열에 R, G, B 값을 나눠 저장했다. fileName에는 "marble.raw"를 초기값으로 넣었다. 그리고 배열 image에 저장된 이미지를 texture로 등록하고 매개변수들도 위와 같이 지정하였다.

```
InitializePoint(1.6);
DrawTorus();

// Finish drawing with the specified texture
glDisable(GL_TEXTURE_2D);

// Turn off the light
glDisable(GL_LIGHTING);
glDisable(GL_LIGHT0);

glutSwapBuffers();
}
```

다시 RenderTorus 함수로 돌아와서 나머지 부분을 보면, torus의 정점들과 법선벡터를 계산해 저장하는 함수 InitializePoint를 호출하고 DrawTorus로 torus를 그렸다. 그린 뒤에는 활성화하였던 texture와 lighting을 비활성화하였다.

```
void DrawTorus()
{
    glBegin(GL_QUADS);
    {
        for (int i = 0; i < 36; i++)
        {
            for (int j = 0; j < 18; j++)
            {
                // The lower-left vertex of the quad
                glNormal3fv(n[i][j]);
                glTexCoord2f(i / 36.0, j / 18.0);
                glVertex3fv(p[i][j]);

                // The lower-right vertex of the quad
                glNormal3fv(n[(i + 1) % 36][j]);
                glTexCoord2f((i + 1) / 36.0, j / 18.0);
                glVertex3fv(p[(i + 1) % 36][j]);

                // The upper-right vertex of the quad
                glNormal3fv(n[(i + 1) % 36][(j + 1) % 18]);
                glTexCoord2f((i + 1) / 36.0, (j + 1) / 18.0);
                glVertex3fv(p[(i + 1) % 36][(j + 1) % 18]);

                // The upper-left vertex of the quad
                glNormal3fv(n[i][(j + 1) % 18]);
                glTexCoord2f(i / 36.0, (j + 1) / 18.0);
                glVertex3fv(p[i][(j + 1) % 18]);
            }
        }
    }
    glEnd();
}
```

torus를 그리는 DrawTorus 함수를 보면, 각 정점들을 그리기 전에 정점의 법선벡터를 지정해주어 Gouraud Shading이 적용되게 하였다. 배열 n에는 각 정점의 법선벡터가 저장되어 있다. 그리고 texture를 늘려 torus 전체를 감싸게 해야 하므로, texture의 s-t 좌표를 각 vertex마다 비율에 맞게 지정해주었다.

```
void keyboard(unsigned char key, int x, int y)
{
    switch (key)
    {
        case '1':
            fileName = "marble.raw";
            glutPostRedisplay();
            break;
        case '2':
            fileName = "wood.raw";
            glutPostRedisplay();
            break;
        case '3':
            fileName = "check.raw";
            glutPostRedisplay();
            break;
        default:
            break;
    }
}
```

keyboard 함수에서 '1', '2', '3' 키를 누를 때마다 변수 fileName의 값을 바꾸어 적용되는 texture가 변경되게 하였다.

- 결과 화면

