컴퓨터그래픽스 HW_3

소프트웨어학부

2015726056

조동희

self-scoring table

	1	2	3	Total
Score	1/1	1/1	1/1	3/3

Texture를 입히기 위해 과제로 제공된 marble, wood, check raw 파일을 읽어올 필요가 있었다.

```
GLchar image[1][512][512][3];
```

```
void loadMarbleTexture() {
    FILE *file[1];
    fopen_s(&file[0], "marble.raw", "rb");
    for (int h = 0; h < 512; h++) {
        fread(image[0][h], 3, 512, file[0]);
    }
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB8, 512, 512, 0, GL_RGB, GL_UNSIGNED_BYTE, image[0]);
    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);

    fclose(file[0]);
}
```

```
void loadWoodTexture() {
    FILE *file[1];
    fopen_s(&file[0], "wood.raw", "rb");
    for (int h = 0; h < 512; h++) {
        fread(image[0][h], 3, 512, file[0]);
    }
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB8, 512, 512, 0, GL_RGB, GL_UNSIGNED_BYTE, image[0]);
    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);

    fclose(file[0]);
}
```

```
void loadCheckTexture() {
    FILE *file[1];
    fopen_s(&file[0], "check.raw", "rb");
    for (int h = 0; h < 512; h++) {
        fread(image[0][h], 3, 512, file[0]);
    }
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB8, 512, 512, 0, GL_RGB, GL_UNSIGNED_BYTE, image[0]);
    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);

    fclose(file[0]);
}
```

fopen_s를 함수를 이용해 raw 파일을 읽고 과제에서 요구하는 512, 512 형식의 rgb를 GLchar 배열에 불러왔다.
그리고 Texture Mapping은 S-T 좌표계를 사용하는데, 시작점은 0으로 하고 끝점은 1로 한다는 것이다.

```
GLuint texID[3];

void init() {
    glEnable(GL_TEXTURE_2D);
    glGenTextures(3, texID);

    glBindTexture(GL_TEXTURE_2D, texID[0]);
    loadMarbleTexture();

    glBindTexture(GL_TEXTURE_2D, texID[1]);
    loadWoodTexture();

    glBindTexture(GL_TEXTURE_2D, texID[2]);
    loadCheckTexture();
```

불러온 각각의 texture를 texID에 저장하고 바인딩해준다.

```
void keyboard(GLFWwindow* window, int key, int scancode, int action, int mods) {
    if (action == GLFW_PRESS || action == GLFW_REPEAT) {
        switch (key) {
        case GLFW_KEY_Q:
        case GLFW_KEY_ESCAPE: glfwSetWindowShouldClose(window, GL_TRUE); break;
        case GLFW_KEY_1: texNum = 0; break;
        case GLFW_KEY_2: texNum = 1; break;
        case GLFW_KEY_3: texNum = 2; break;
        }
    }
}
```

1,2,3을 누르는 것에 따라 화면에 나오도록 한다.

그래프에 torus를 그리면서 texture mapping을 같이 한다.

```
void drawTexturedCube() {
    glBindTexture(GL_TEXTURE_2D, texID[texNum]);
    for (int i = 0; i < 36; i++) {
        for (int j = 0; j < 18; j++) {

            vec3 p1(p[i][(j + 1) % 18].x - p[i][j].x, p[i][(j + 1) % 18].y - p[i][j].y, p[i][(j + 1) % 18].z - p[i][j].z);
            vec3 p2(p[(i + 1) % 36][j].x - p[i][j].x, p[(i + 1) % 36][j].y - p[i][j].y, p[(i + 1) % 36][j].z - p[i][j].z);
            vec3 normalVector = cross(p2, p1);
            vec3 middleVector = p[i][j] + p[i][(j + 1) % 18] + p[(i + 1) % 36][j] + p[(i + 1) % 36][(j + 1) % 18];

            glColor3f(1, 1, 1);
            glBegin(GL_QUADS);
            glTexCoord2f((float)i / 36, (float)j / 18);
            glNormal3f(normal[i % 36][j % 18].x, normal[i % 36][j % 18].y, normal[i % 36][j % 18].z);
            glVertex3f(p[i][j].x, p[i][j].y, p[i][j].z);

            glTexCoord2f((float)i / 36, (float)(j + 1) / 18);
            glNormal3f(normal[i % 36][(j + 1) % 18].x, normal[i % 36][(j + 1) % 18].y, normal[i % 36][(j + 1) % 18].z);
            glVertex3f(p[i][(j + 1) % 18].x, p[i][(j + 1) % 18].y, p[i][(j + 1) % 18].z);

            glTexCoord2f((float)(i + 1) / 36, (float)(j + 1) / 18);
            glNormal3f(normal[(i + 1) % 36][(j + 1) % 18].x, normal[(i + 1) % 36][(j + 1) % 18].y, normal[(i + 1) % 36][(j + 1)
            glVertex3f(p[(i + 1) % 36][(j + 1) % 18].x, p[(i + 1) % 36][(j + 1) % 18].y, p[(i + 1) % 36][(j + 1) % 18].z);

            glTexCoord2f((float)(i + 1) / 36, (float)j / 18);
            glNormal3f(normal[(i + 1) % 36][j % 18].x, normal[(i + 1) % 36][j % 18].y, normal[(i + 1) % 36][j % 18].z);
            glVertex3f(p[(i + 1) % 36][j].x, p[(i + 1) % 36][j].y, p[(i + 1) % 36][j].z);
            glEnd();
        }
    }
}

float* nomalVec = new float[3];
for (int i = 0; i < 36; i++) {
    for (int j = 0; j < 18; j++) {
        float line1[3] = { p[i % 36][j % 18].x - p[i % 36][(j + 1) % 18].x, p[i % 36][j % 18].y - p[i % 36][(j + 1) % 18].y, p[i % 36][j % 18].z - p[i % 36][(j + 1) % 18].z };
        float line2[3] = { p[(i + 1) % 36][(j + 1) % 18].x - p[i % 36][(j + 1) % 18].x, p[(i + 1) % 36][(j + 1) % 18].y - p[i % 36][(j + 1) % 18].y, p[(i + 1) % 36][(j + 1) % 18].z - p[i % 36][(j + 1) % 18].z };

        nomalVec[0] = line1[1] * line2[2] - line1[2] * line2[1];//외적
        nomalVec[1] = line1[2] * line2[0] - line1[0] * line2[2];
        nomalVec[2] = line1[0] * line2[1] - line1[1] * line2[0];

        float size = sqrt((nomalVec[0] * nomalVec[0]) + (nomalVec[1] * nomalVec[1]) + (nomalVec[2] * nomalVec[2]));

        nomalVec[0] /= size;//nomarlize
        nomalVec[1] /= size;
        nomalVec[2] /= size;

        normal[i % 36][j % 18].x += nomalVec[0];
        normal[i % 36][j % 18].y += nomalVec[1];
        normal[i % 36][j % 18].z += nomalVec[2];

        normal[i % 36][(j + 1) % 18].x += nomalVec[0];
        normal[i % 36][(j + 1) % 18].y += nomalVec[1];
        normal[i % 36][(j + 1) % 18].z += nomalVec[2];

        normal[(i + 1) % 36][j % 18].x += nomalVec[0];
        normal[(i + 1) % 36][j % 18].y += nomalVec[1];
        normal[(i + 1) % 36][j % 18].z += nomalVec[2];

        normal[(i + 1) % 36][(j + 1) % 18].x += nomalVec[0];
        normal[(i + 1) % 36][(j + 1) % 18].y += nomalVec[1];
        normal[(i + 1) % 36][(j + 1) % 18].z += nomalVec[2];
    }
}
```

glTexCoord2f를 이용하여 glVertex3f로 지정한 점에 대해 Texture 맵핑 좌표를 설정해 줄 때, glVertex3f로 지정한 위치에 대해 glTexcoord2f 좌표가 서로 일치시켜 mapping하게 되는 것이다. 이에 외적을 통해 vertex에서의 nomal 벡터를 구하고 정규화한다.