심화전공실습

HW #14

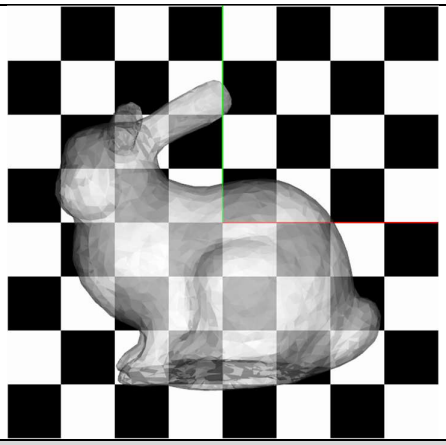
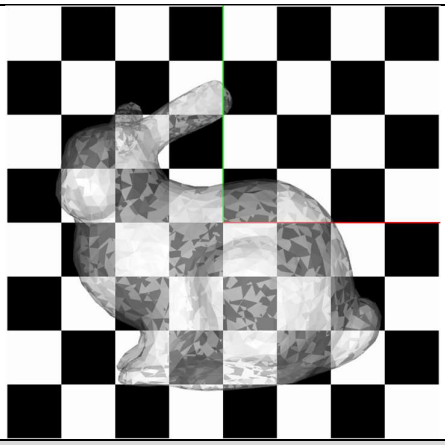
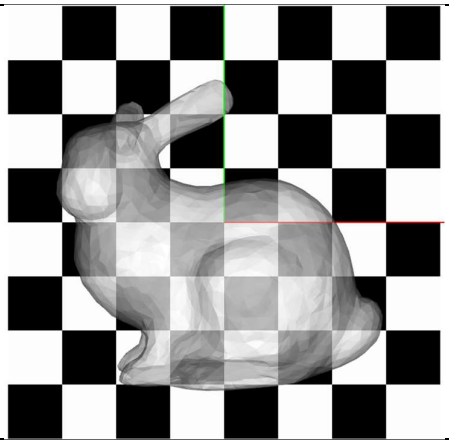
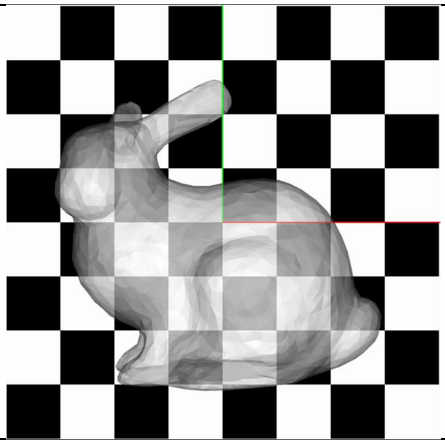
Self-scoring table

	P01	P02	P03	E01	Total
Score	1	1	1	1	5

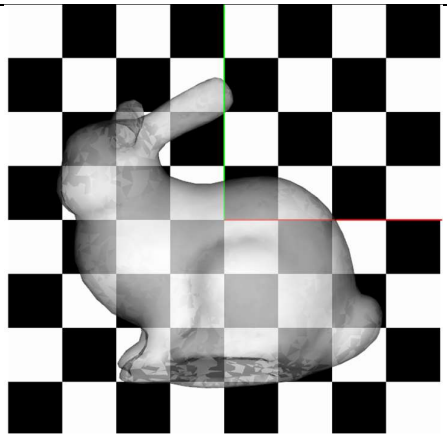
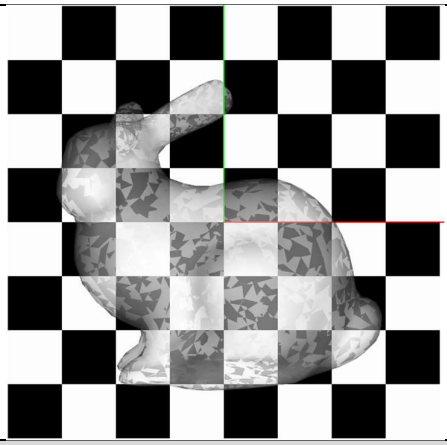
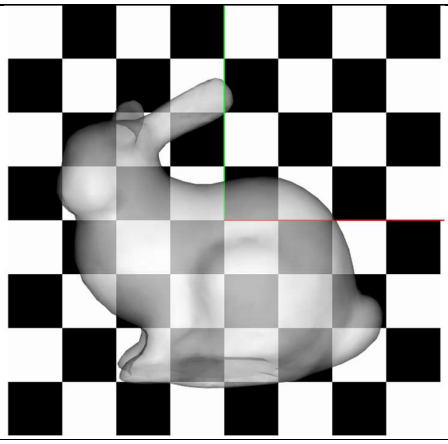
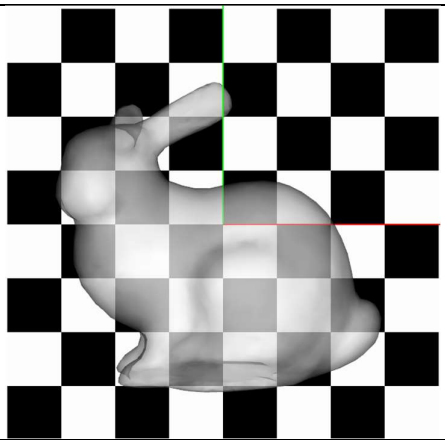
과목명	심화전공실습
학부	소프트웨어학부
학번	2019203010
이름	김민철
제출일자	2024년 12월 08일

I. Practice

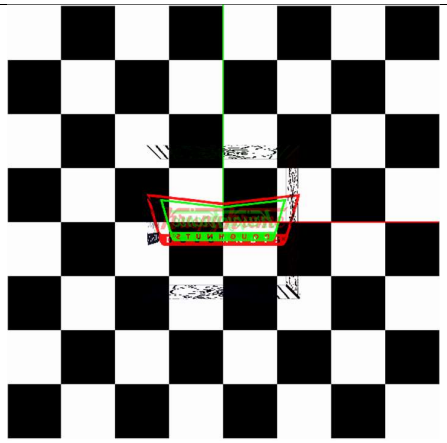
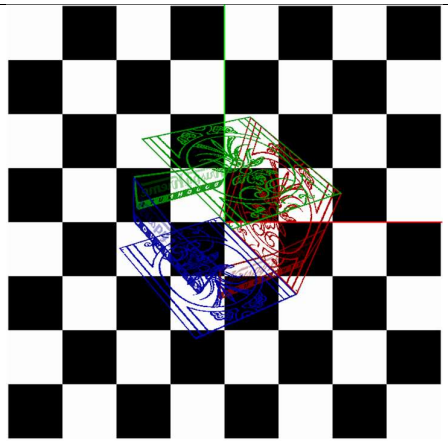
Practice 01. Turn on/off depth sorting to a translucent flat bunny

Depth sorting off	
Depth mask = false	Depth mask = true
	
Depth sorting on	
Depth mask = false	Depth mask = true
	

Practice 02. Turn on/off depth sorting to a translucent smooth bunny

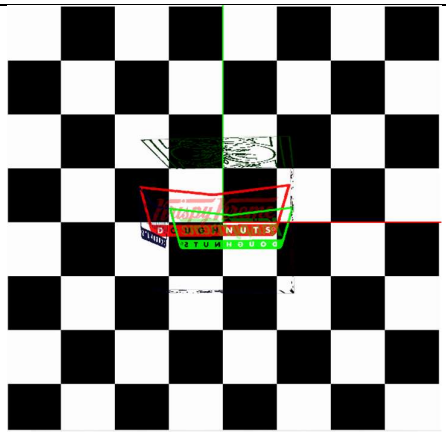
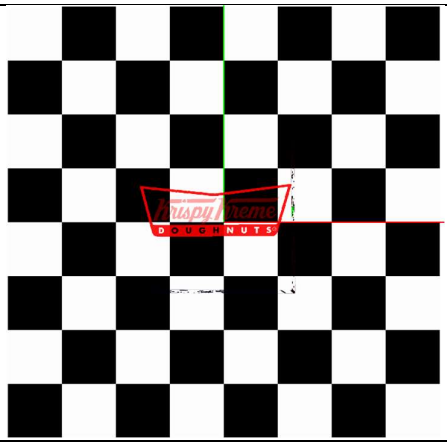
Depth sorting off	
Depth mask = false	Depth mask = true
	
Depth sorting on	
Depth mask = false	Depth mask = true
	

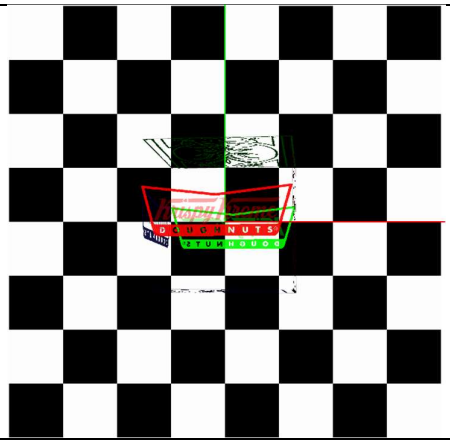
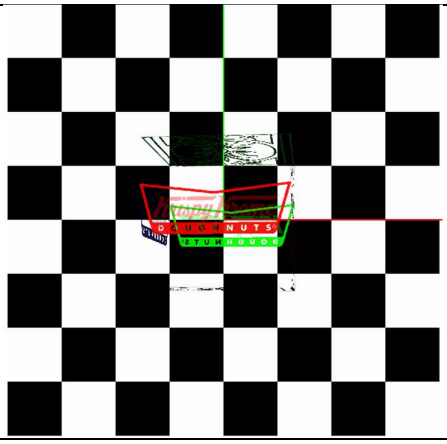
Practice 03. Alpha textured cube without depth sorting

Frontal	Rotation(side)
	

II. Exercise

Exercise 01. Alpha textured cube with/without depth sorting

Depth sorting off	
Depth mask = false	Depth mask = true
	

Depth sorting on	
Depth mask = false	Depth mask = true
	

```
struct DepthSortData
{
    int i;
    vec4 center;
    float depth;
};
DepthSortData* fdsd = NULL;
int compare(const void* a, const void* b)
{
    float depth1 = ((DepthSortData*)a)->depth;
    float depth2 = ((DepthSortData*)b)->depth;

    if (depth1 > depth2)return -1;
    if (depth1 < depth2)return 1;

    return 0;
}
```

1. 자료구조 선언 및 null 초기화

Depth sort 를 위해 자료구조를 선언하고 null 값으로 초기화한다. Compare 함수는 깊이를 비교해 값을 출력하는 함수이다.

```
void sortCubeFace()
{
    // Get the current model view matrix
    GLfloat modelView[16];
    glGetFloatv(GL_MODELVIEW_MATRIX, modelView);

    // OpenGL employs the row vector convention
    // and glm employs the column-major representation.
    mat4 M = make_mat4(modelView);

    // Depth of every face
    for (int i = 0; i < 6; i++)
    {
        // Face center in the eye coordinate system
        vec4 center_eye = M * fdsd[i].center;

        // The camera faces the negative z-axis in OpenGL
        fdsd[i].depth = -center_eye.z;
    }
    // Depth sorting
    qsort(fdsd, 6, sizeof(DepthSortData), compare);
}
```

2. sortCubeFace 함수

큐브의 각 면에 대해 depth 를 가져오고 그 depth 를 기준으로 카메라로부터 멀리 떨어진 것부터 순서대로 정렬해 저장하는 함수이다.

```
for (int i = 0; i < 6; i++)
{
    center = vec3(0, 0, 0);
    for (int j = 0; j < 4; j++)
        center += vertex[i][j] / 4.0f;
    fdsd[i].i = i;
    fdsd[i].center = vec4(center, 1.0f);
}

// Texture coordinate
texcoord[0] = vec2(0, 1);
texcoord[1] = vec2(1, 1);
texcoord[2] = vec2(1, 0);
texcoord[3] = vec2(0, 0);


// Sort cube faces
sortCubeFace();
// Cube
for (int i = 0; i < 6; i++)
{
    int iFace = fdsd[i].i;
    glBindTexture(GL_TEXTURE_2D, texID[iFace + 1]);

    glBegin(GL_QUADS);
    glNormal3fv(value_ptr(normal[iFace]));
    for (int j = 0; j < 4; j++)
    {
        glTexCoord2fv(value_ptr(texcoord[j]));
        glVertex3fv(value_ptr(vertex[iFace][j]));
    }
    glEnd();
}
```

3. drawSortedTexturedCube() 함수의 동작 설명

주어진 큐브의 각 면을 깊이(depth) 순서대로 정렬하여 화면에 그려줌으로써, 가까운 면이 먼 면을 가리지 않도록 하여 정확한 3D 효과를 구현한다.

1. fdsd 초기화 :

- fdsd 배열의 모든 요소의 i 값을 0 으로 초기화합니다. 이는 각 면에 대한 임시적인 인덱스 역할을 합니다.GL_REPEAT: 텍스처가 범위를 벗어나면 반복적으로 그려짐.
- fdsd 배열의 각 요소의 center 값을 해당 면의 중점을 나타내는 4 차원 벡터로 설정합니다. 이 벡터는 깊이를 계산하는 데 사용됩니다.

2. 깊이 기반 정렬 :

- sortCubeFace() 함수를 호출하여 fdsd 배열을 깊이 값을 기준으로 오름차순으로 정렬합니다. 즉, 카메라에서 가장 먼 면부터 가까운 면 순으로 정렬됩니다.

3. 면 그리기 :

- 정렬된 fdsd 배열을 순회하며, 각 요소의 인덱스를 iFace 에 저장합니다.
- iFace 를 이용하여 큐브의 각 면을 순서대로 그립니다. 먼저 그려지는 면이 나중에 그려지는 면에 가려지므로, 깊이 순서대로 그려짐으로써 정확한 3D 효과를 얻을 수 있습니다.

III. 느낀점

OpenGL을 이용하여 알파 블렌딩을 구현하는 프로젝트를 진행하며, 3D 그래픽의 기본 원리와 구현 과정에 대한 이해를 깊이 할 수 있었습니다. 특히, Depth Sort를 통해 큐브의 면들을 정확한 순서로 그려내는 과정은 추후 더 공부하게될 3D 그래픽 렌더링 파이프라인의 중요성을 실감하게 해주게하는 실습이었습니다.