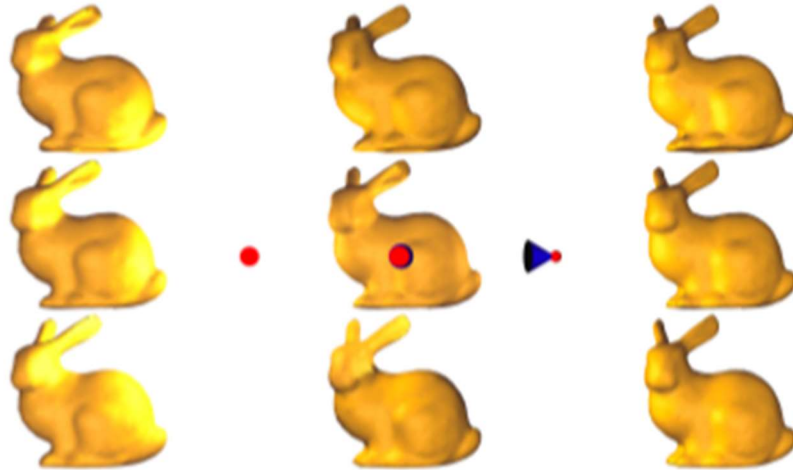


Computer Graphics



Shading

P07_SHADING_CPP

이민재 | Computer Graphics [심화전공실습 1] | 2020/10/18

	P01	P02	E01	E02	Total
SCORE	1	1	1	1	4

P01 (Smooth/flat shading of spheres with various tessellations)

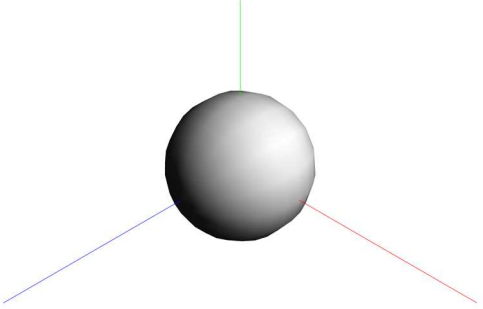
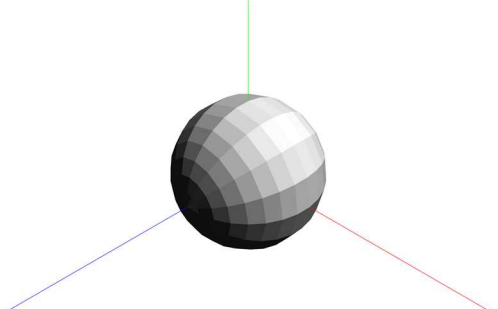
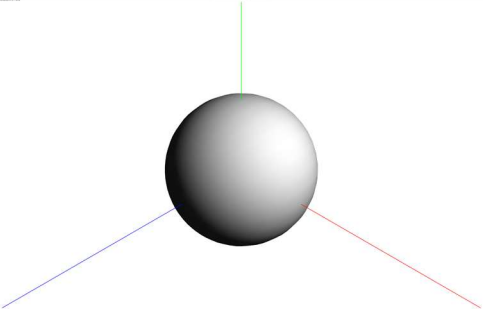
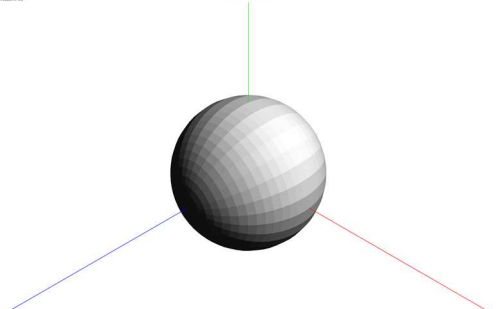
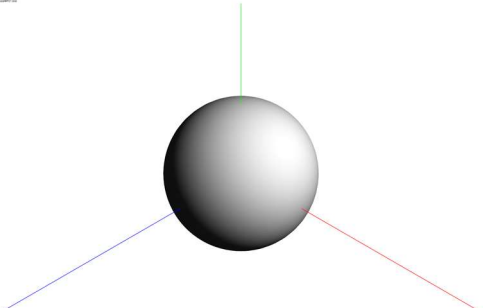
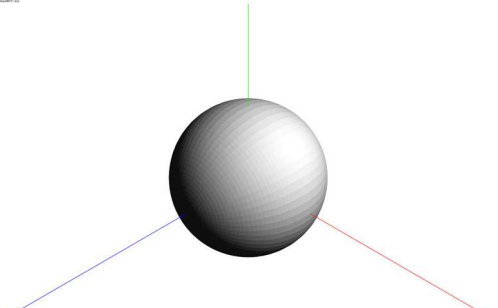
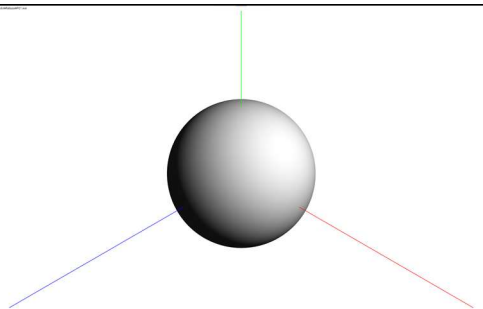
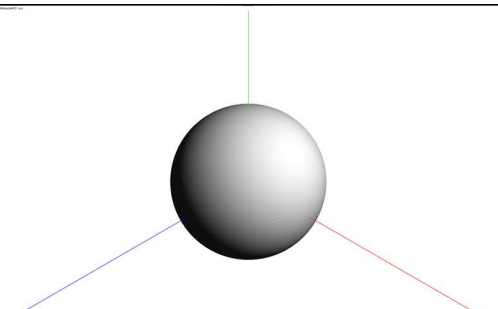
<SNAPSHOT>

* *SLICES*

SPECIFIES THE NUMBER OF SUBDIVISIONS AROUND THE Z AXIS (SIMILAR TO LINES OF LONGITUDE).

**STACKS*

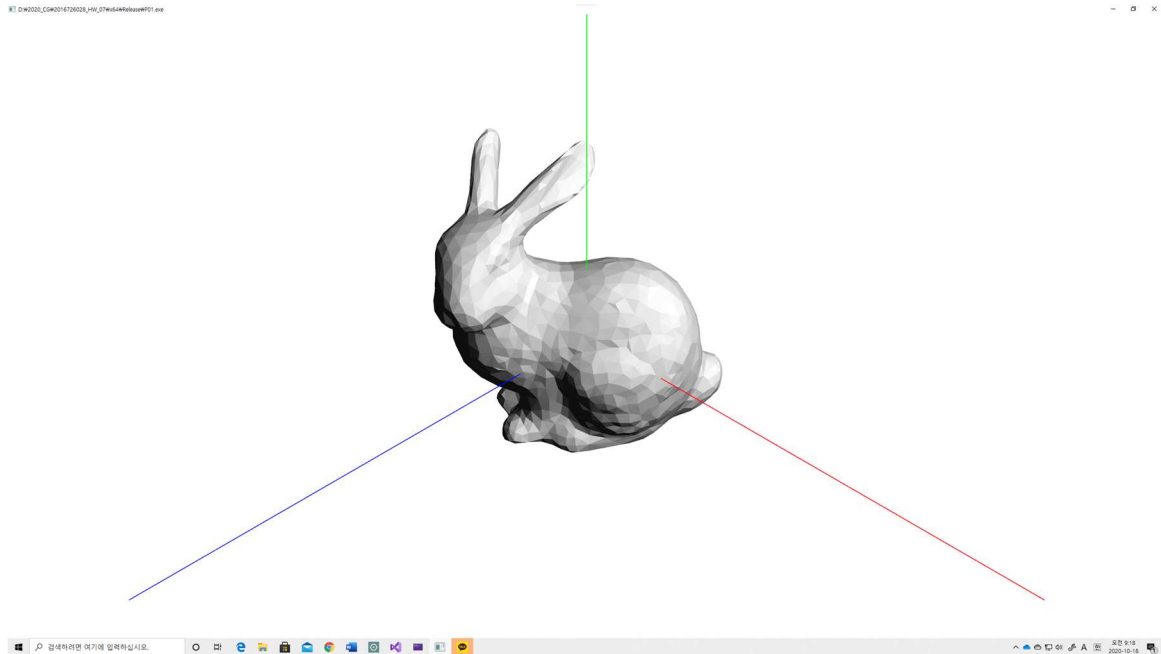
SPECIFIES THE NUMBER OF SUBDIVISIONS ALONG THE Z AXIS (SIMILAR TO LINES OF LATITUDE).

TESELLATION	SMOOTH	FLAT
16 SLICES/STACKS		
32 SLICES/STACKS		
64 SLICES/STACKS		
128 SLICES/STACKS		

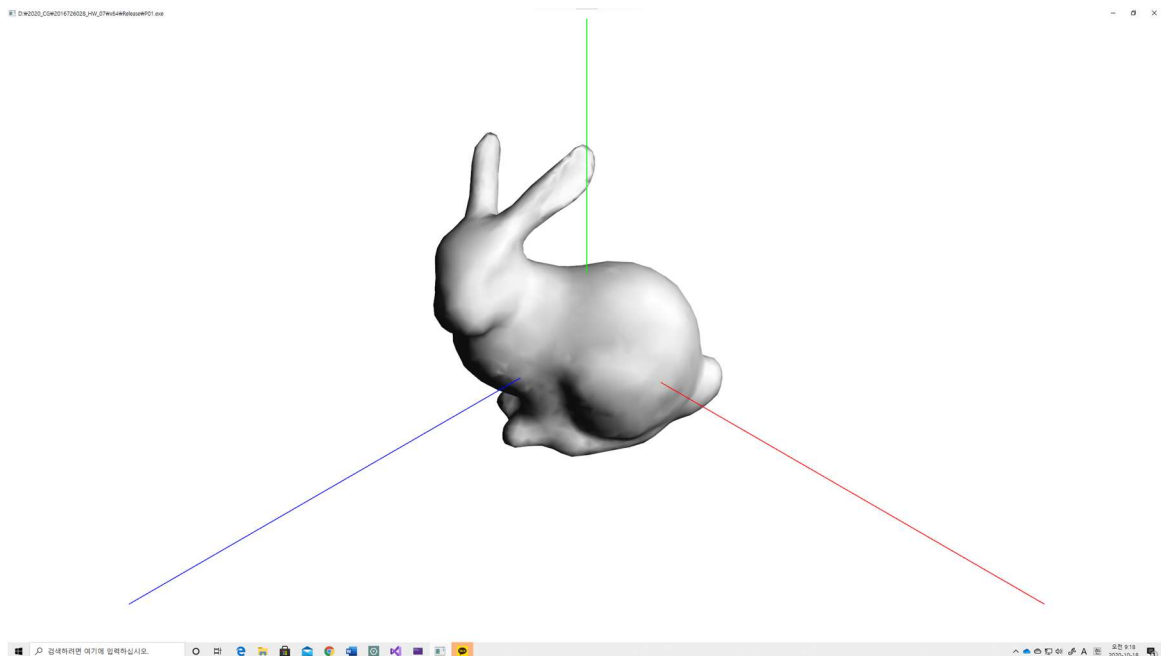
P02 (Normal vector estimation for flat/smooth shading of a bunny model)

<SNAPSHOT>

Using Face Normal vectors (flat approximation of bunny model)

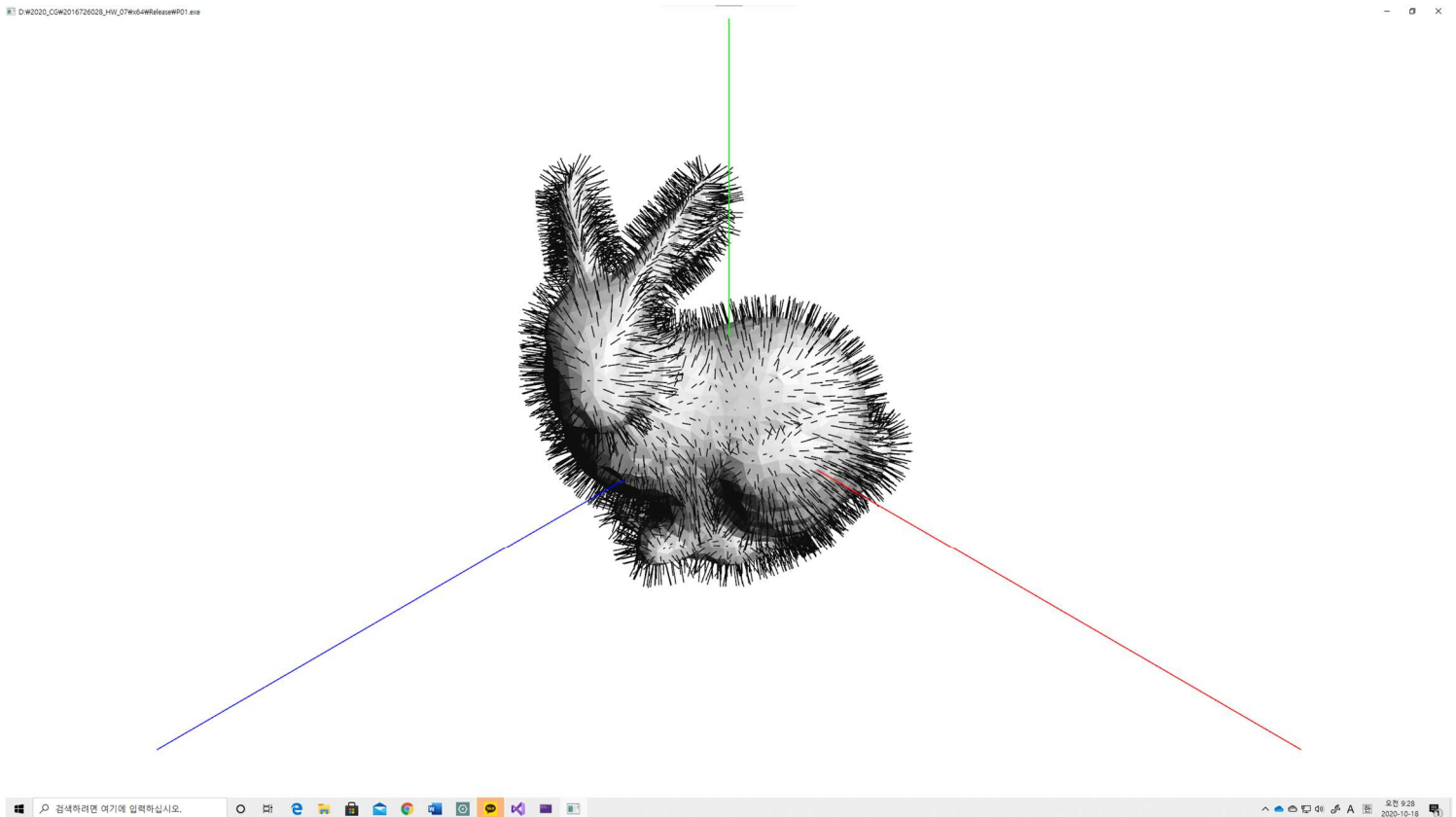


Using Vertex Normal vectors (smooth approximation of bunny model)



E01 (Draw normal vectors of triangles of the bunny model)

<SNAPSHOT>



<EXPLANATION>

각 삼각형의 중점에서 바깥쪽으로 나가는 FACE NORMAL VECTOR 을 표현하기 위해 먼저 한 삼각형을 이루는 각 버텍스들을 이용하여 중점 POINTS 를 추출하고 컨테이너에 저장하였다.

```
// middle_Points
mPoints = new glm::vec3[nFaces];
for (int i = 0; i < nFaces; i++) {
    glm::vec3 p = (vertex[face[0][i]] + vertex[face[1][i]] + vertex[face[2][i]]) / 3.0f;
    mPoints[i] = p;
}
```

그 이후 , 선분 출력을 위해 중점과 FORMAL VECTOR 값을 더하여, 중점과 FORMAL VECTOR 위의 점을 선분으로 이어 출력하였다.

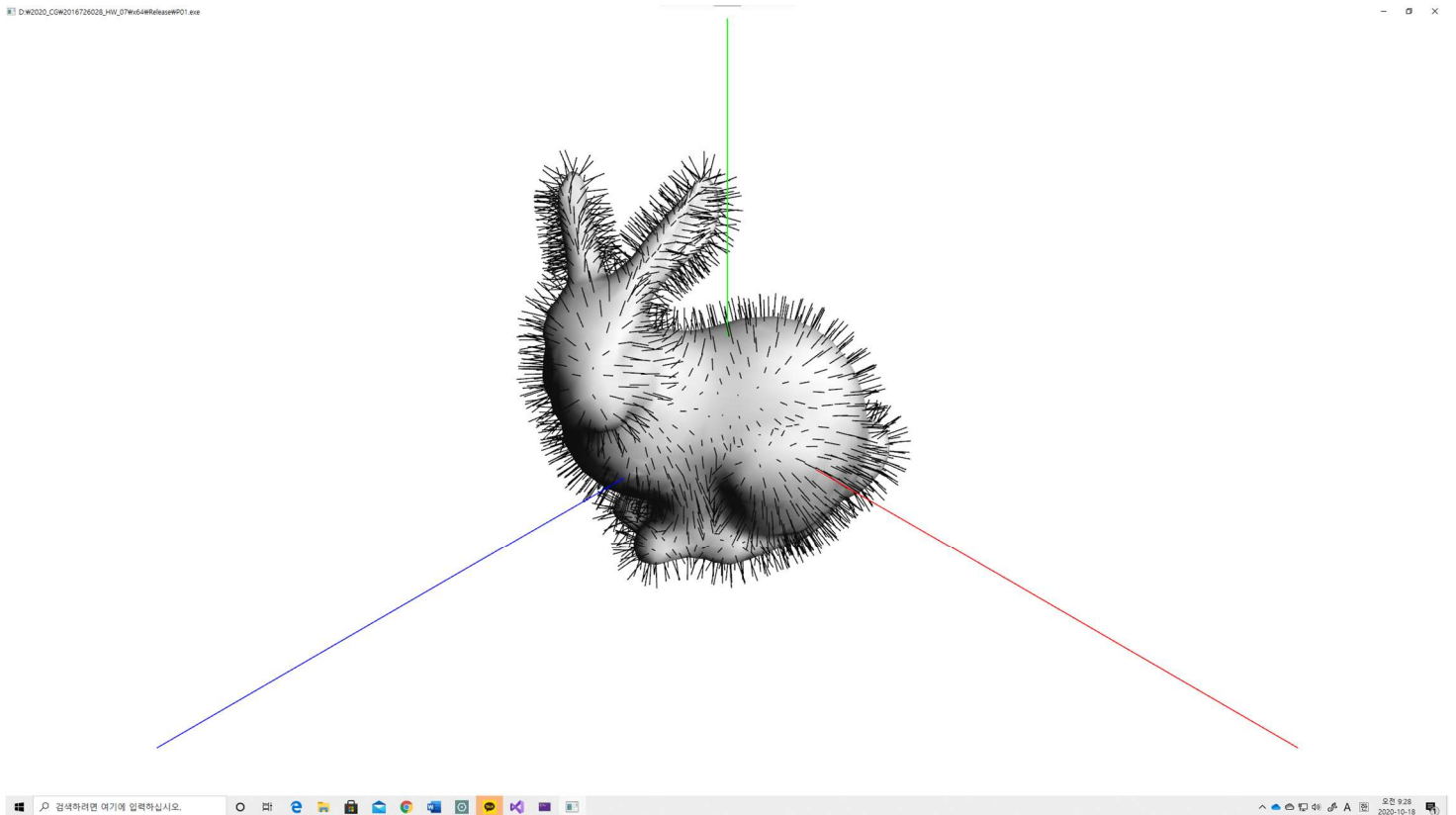
```
glBegin(GL_LINES);
for (int i = 0; i < nFaces; i++)
{
    glVertex3fv(value_ptr(mPoints[i])); //삼각형 중점 (시작점)
    glVertex3fv(value_ptr((mPoints[i]+fnormal[i]/8.0f))); // fnormal 벡터 위의 점
}

glEnd();
```

보다 나은 출력 결과를 위해 FORMAL 방향 SCALAR 값을 8 로 나누어 조절하였다.

E02 (Draw normal vectors of vertices of the bunny model)

<SNAPSHOT>



<EXPLANATION>

```
glBegin(GL_LINES);
for (int i = 0; i < nFaces; i++)
{
    for (int j = 0; j < 3; j++)
    {
        glVertex3fv(value_ptr(vertex[face[j][i]])); //삼각형 구성하는 vertex
        glVertex3fv(value_ptr((vertex[face[j][i]] + vnormal[face[j][i]] / 8.0f))); // vertex to vnormal 벡터 위의 점
    }
}
glEnd();
```

VERTEX NORMAL VECTOR의 경우, FACE 구조체에 각 버텍스가 저장되어 이미 되어 있고 VNORMAL의 경우에도 각 버텍스마다 계산되어 저장되어 있으므로(VNORMAL[FACE[J][I]]) E01과 동일하게 GL_LINES를 이용하여 버텍스와, 버텍스 NORMAL VECTOR 위의 한점을 이어 출력하였다. 역시, 보기좋은 출력을 위하여 VNORMAL SCALAR 값을 8로 나누어 주었다.