심화전공실습

HW #05

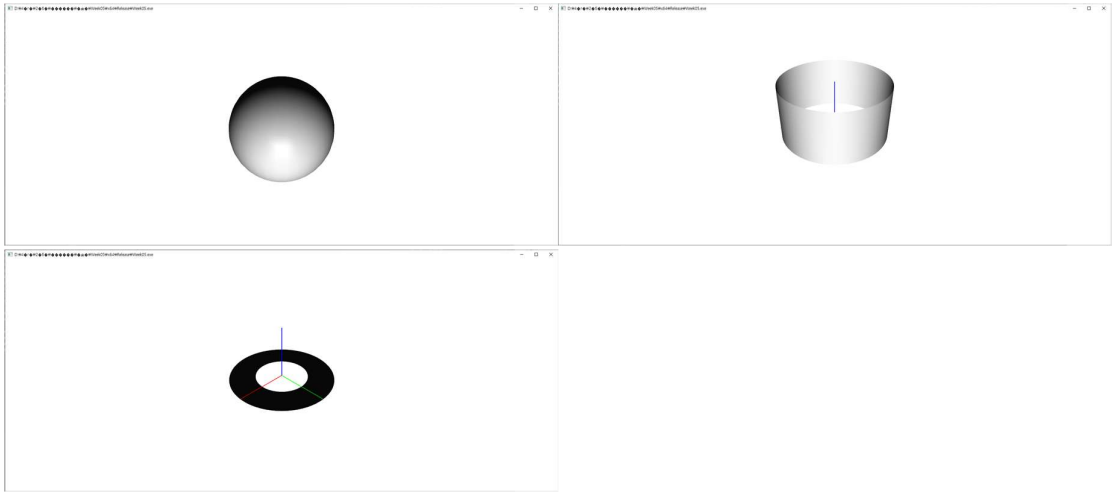
Self-scoring table

	P01	P02	P03	E01	E02	Total
Score	1	1	1	1	1	5

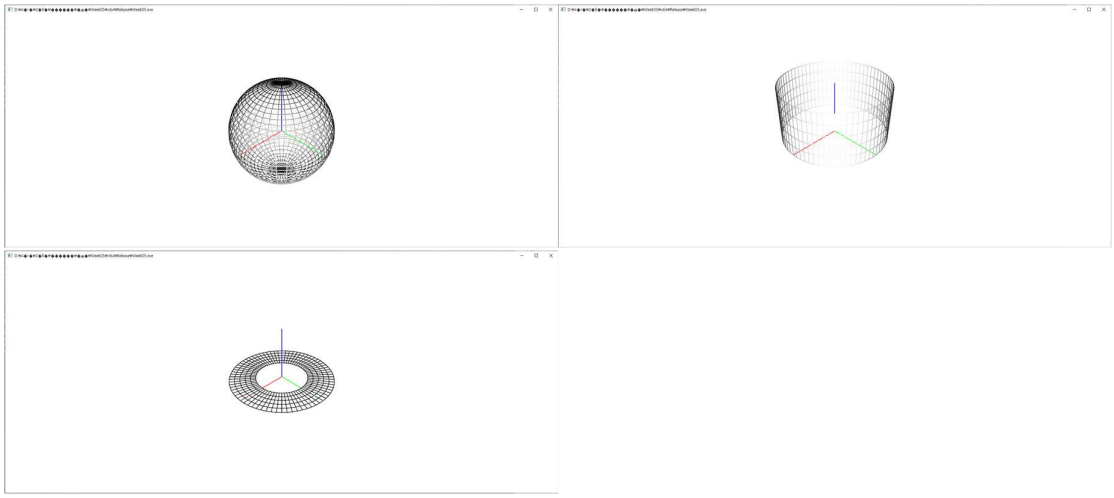
과목명	심화전공실습
학부	소프트웨어학부
학번	2019203010
이름	김민철
제출일자	2020년 10월 06일

I. Practice

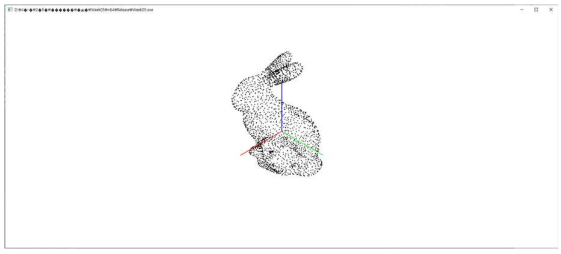
Practice 01. Draw OpenGL quadric objects: sphere, cylinder, disk



Practice 02. Polygon fill on/off

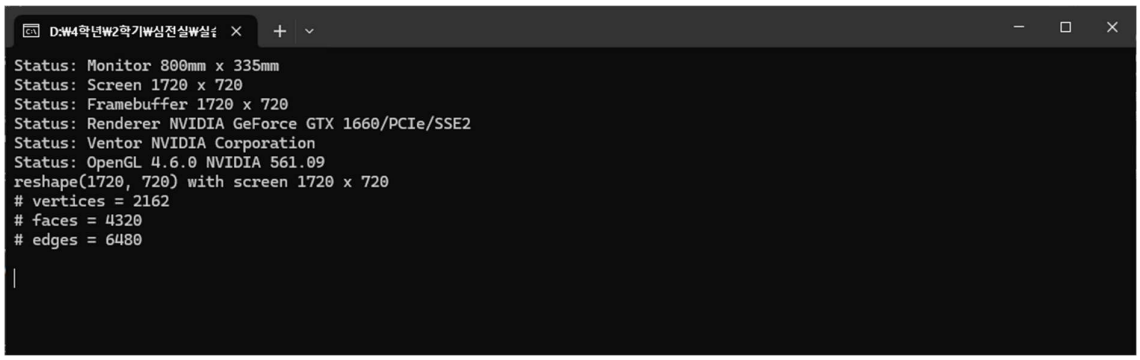


Practice 03. Read/draw a bunny model using points



II. Exercise

Exercise 01. Extract all the edges and print # of edges of a bunny model




```
// Vertices
vertex = new vec3[nVertices];
for (int i = 0; i < nVertices; i++)
    is >> vertex[i].x >> vertex[i].y >> vertex[i].z;

// Faces
face[0] = new int[nFaces];
face[1] = new int[nFaces];
face[2] = new int[nFaces];

int n;
for (int i = 0; i < nFaces; i++)
{
    is >> n >> face[0][i] >> face[1][i] >> face[2][i];
    if (n != 3) cout << "# vertices of the " << i << "-th faces = " << n << endl;
}

for (int i = 0; i < nFaces; i++) {
    edges.insert({ face[0][i], face[1][i] });
    edges.insert({ face[1][i], face[2][i] });
    edges.insert({ face[2][i], face[0][i] });
}
cout << "# edges = " << edges.size() << endl;
```

3 차원 메시 데이터를 읽어, 이 메시를 구성하는 edge 들을 추출하고 그 개수를 세는 코드를 작성한다.

1. 정점(Vertex) 데이터 읽기 :

vertex 배열에 각 정점의 3 차원 좌표(x, y, z)를 저장한다.

입력 파일에서 정점의 개수 nVertices 만큼 반복하며 각 정점의 좌표를 읽어들인다.

2. 면(Face) 데이터 읽기 :

face[0], face[1], face[2] 배열에 각 면을 구성하는 세 개의 정점의 인덱스를 저장한다.

입력 파일에서 면의 개수 nFaces 만큼 반복하며 각 면의 정점 인덱스를 읽어들인다.

3. 모서리(Edge) 추출 및 저장 :

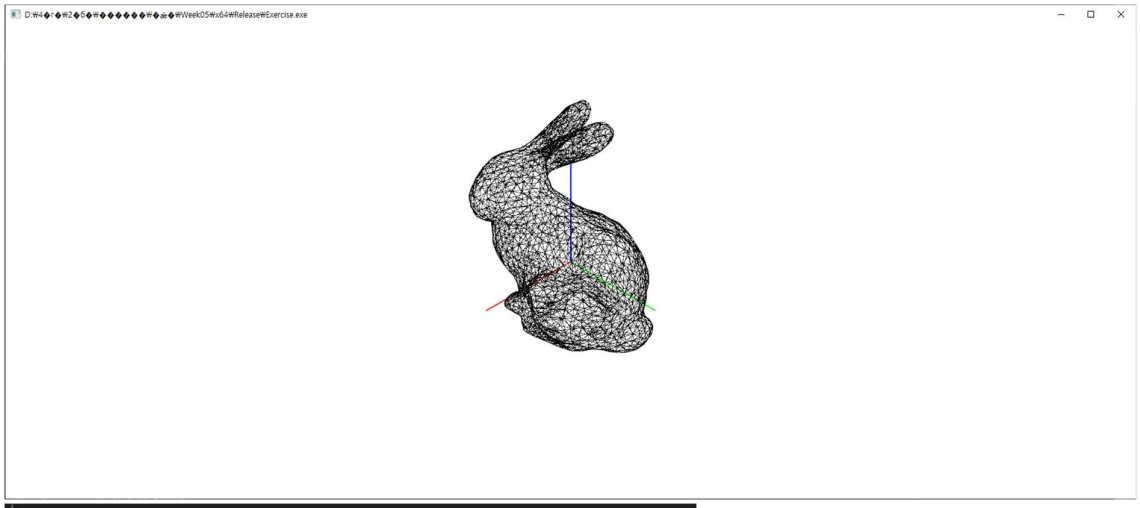
각 면에 대한 세 개의 에지를 추출합니다. 예를 들어, i 번째 면의 경우 {face[0][i], face[1][i]}, {face[1][i], face[2][i]}, {face[2][i], face[0][i]} 이렇게 각 면과 인접한 세 개의 에지를 얻을 수 있다.

추출한 에지를 edges 라는 set 에 삽입합니다. set 은 자동으로 정렬되며 중복을 허용하지 않기 때문에, 동일한 에지는 한 번만 저장되어 중복되는 edge 를 지울 수 있다.

4. 유일한 Edge 개수 계산 :

edges.size()를 통해 edges 에 저장된 유일한 에지의 개수를 구하고 출력합니다.

Exercise 02. Draw the extracted edges using lines



```
// Draw all the vertices of the bunny model
glLineWidth(1.0f);
glBegin(GL_LINES);
for (const auto& edge : edges) {
    for (int k : edge) {
        glVertex3f(vertex[k].x, vertex[k].y, vertex[k].z);
    }
}
glEnd();
```

앞서 계산된 메시의 Edge 정보를 이용하여 OpenGL 을 통해 3 차원 공간에 실제로 그려준다.

1. for (const auto& edge : edges) :

edges 집합에 저장된 모든 Edge 를 순회한다. 각 Edge 는 두 개의 정점 인덱스로 구성되어 있다.

2. for (int k : edge) :

현재 Edge 를 구성하는 두 개의 정점 인덱스를 순회한다.

3. glVertex3f(vertex[k].x, vertex[k].y, vertex[k].z); :

vertex[k]는 현재 정점의 3 차원 좌표를 나타낸다. 이 좌표를 OpenGL 에게 전달하여 선의 한 끝점을 설정한다. 즉, 각 에지에 대해 두 번 호출되어 선의 양 끝점을 지정한다.

III. 느낀점

OFF 파일 포맷을 읽어들여 정점과 면 정보를 추출하고, 에지를 계산하여 edges 집합에 저장하는 과정을 수행하면서 OpenGL을 이용한 3D 매시 데이터 처리에 대한 학습을 진행할 수 있었다.

추가로 Exercise를 진행하면서 어떤 자료구조를 이용해야하는지에 대한 고민을 통해 데이터를 효율적으로 관리하는 방법을 학습했고 set STL을 이용할 수 있었다.

앞으로 계속되는 과제와 학습을 통해 OpenGL에 대해 더 지식을 쌓아 갈 것이다.