**Triangle Mesh Rendere 구현**

1723871 장세진

|  |
| --- |
| 기본 구현사항(3점) : 파일 읽기 및 데이터 저장 / 렌더링 |
| Fstream을 통해 bunny.obj를 불러오기  텍스트, 오렌지, 스크린샷이(가) 표시된 사진  자동 생성된 설명 |
| Ifstream을 통한 bunny.obj를 총 5가지의 종류로 구분하여 parse 함수를 사용핮 않고 읽었다.   1. s[0] == '#' 일때는 comments로 바로 출력을 해주었다.      1. s[0] == 'v' && s[1] != 't' && s[1] != 'n'일때는 공백을 기준으로 파싱하여 2번 인덱스 부터 pts 벡터에 넣어 포인트 구성을 하였다.      1. s[0] == 'v' && s[1] == 'n'일때는 3번 인덱스를 시작으로 공백을 파싱하여 normals 벡터에 넣어 구성하였다.      1. s[0] == 'f'는 현 obj에서는 // 만 들어오기 때문에 //이전과 이후를 flag를 통해 구분하여 face 구조체에 넣어 faces 벡터에 push를 해주었다.   ㄴ   1. 그 외는 아무 프로세스가 아니란 것을 출력해주었다. |
| DrawModel에서 GL\_COLOR\_BUFFER\_FIT을 초기화해주고 faces 벡터에 저장된 면을 이용하여 normal과 pts에 저장한 좌표를 이용하여 다각형을 반복해서 그림  이후 AABB BVH를 구현하기 위해 DrawDFS함수 실행 |

다음 페이지 추가 구현

|  |
| --- |
| 추가 구현사항(3점) : AABB BVH |
| BVH를 구현하기 위한 기본 BV 정의 |
| BVH 정의 |
| BVH 생성자 구현 |
| BV 생성자 구현  fa벡터에 inputData값 복사  SetBV()함수 실행을 통한 AABB의 최대값, 최소값 매핑  fa벡터의 size가 2보다 큰 조건으로 생성자 실행    왼쪽, 오른쪽 자식을 구분하기 위해 child0, child1 변수 선언  SetBV를 통해 각 좌표의 최소, 최대값을 얻을 수 있으므로 min값과 max값의 중간값을 각 xMid, yMid, zMid에 저장    각각 최대값과 중간값과의 거리를 활용하여 가장 큰 축을 기준으로 분할 진행  중간값이 부모 AABB의 중간값보다 크면 오른쪽 작으면 왼쪽 자식에 추가 하기 위해 child0, child1에 구분하여 저장        구분하여 저장한 자식이 존재하면 다시 생성자 호출하여 자식 노드 생성 |
| 최하위 리프노드인지 확인하기 위한 bunnyBV 함수 |

|  |  |
| --- | --- |
| bunnyBV의 SetBV함수로 AABB를 위한 최소 좌표, 최대 좌표를 매핑 | |
| aabb를 그려주는 함수  색상은 파란색으로 하고 SetBV를 이용해 매핑된 좌표를 활용하여 육면체를 그림 | |
| 깊이 우선 탐색을 활용하여 AABB를 그리기 때문에 최대 리프노드의 깊이를 탐색하는 함수 선언 | |
| 깊이 우선 탐색의 방식으로 트리 노드의 AABB를 그림 | |
| MyInit 함수에 전역변수로 선언한 bvh를 생성해주고 최대 깊이값을 얻어낸다. | |
| MySpecial 함수에 키보드 위, 아래를 추가하여 depth에 값을 변경해주었고 최대값과 최소값을 확인하여 오버플로우나 언더플로우가 발생하지 않도록 방지 | |
| 실행 화면 | |
|  |  |
| Depth = 1 | Depth = 2 |
|  |  |
| Depth = 3 | Depth = 4 |
|  |  |
| Depth = 10 | Max Depth = 17 |