컴퓨터그래픽

ClassAssignment 2

2016025996

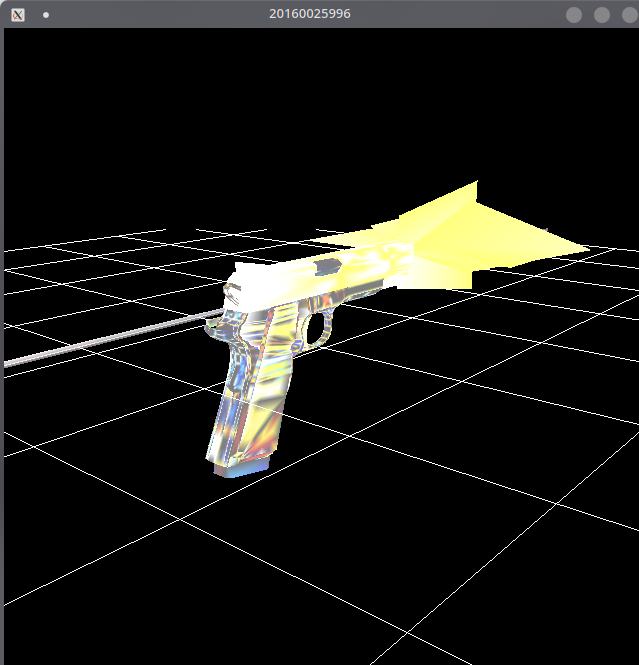
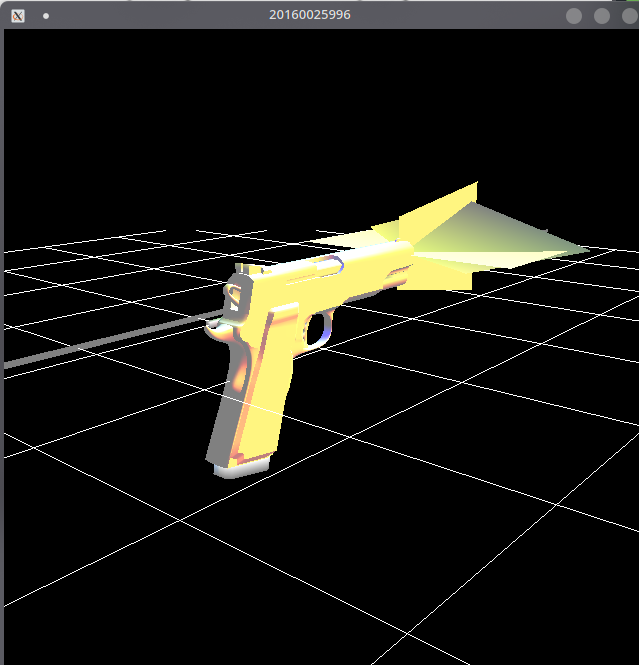
조성진

ClassAssignment에서 구현한 사항 으로는 ClassAssignment1에서 사용한 프로그램 코드에서 grid plane만 그리고, 추가적으로 그렸던 사람로봇은 삭제했다. 그리고 드래그&드롭을 사용해서 obj파일을 읽어오는 기능을 만들었으며, obj파일을 내부적으로 파싱해서 vertex와 normal과 face를 읽어와서, face와 vertex와 normal의 조합으로 그림을 그렸다. 그림을 그릴 때 glDrawElements()를 사용했다.

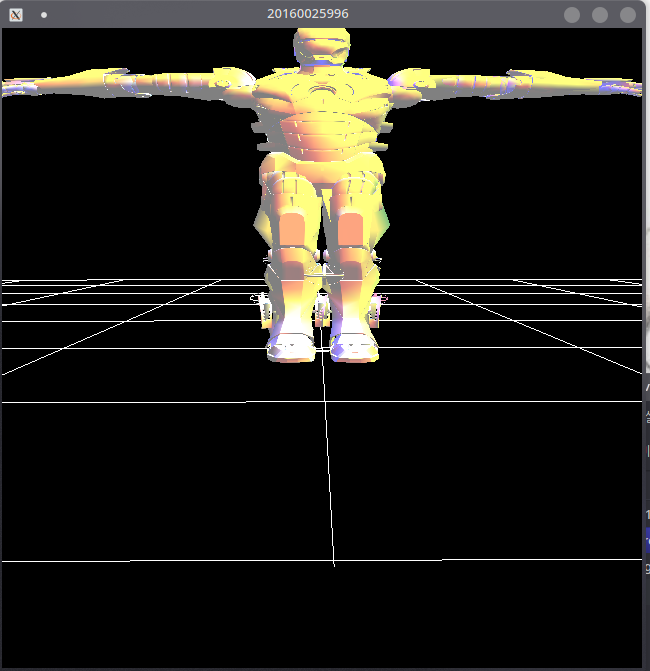
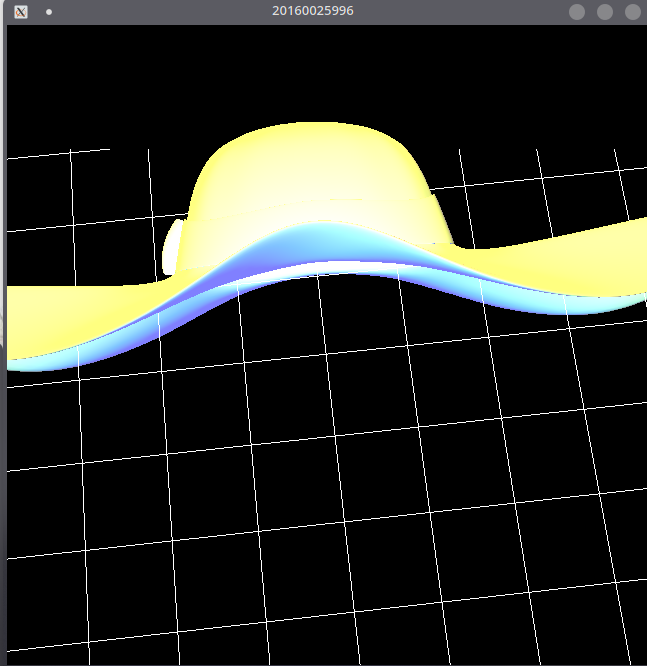
그리고 Z키를 눌러서 wireframe과 solid frame간의 모드스위치를 사용했다.(토글방식으로 1과 XOR시키는 방식으로 구현했다.) obj파일을 파싱해서 그림도 그리지만, 콘솔창에 파일이름과 face갯수 같은 정보들을 출력하도록 만들었다. ClassAssignment2에서 내가 사용한 광원은 3가지이다. 각각의 빛은 (3,4,5) , (4,5,3), (0, 10, 0)으로 light source를 위치시키고, (0, 10, 0)으로부터의 빛만 point light로 만들고 나머지는 directional light로 만들었다.

S키를 누르는면 shading의 방식이 토글되는데(위의방식과 마찬가지로) 처음에 S키를 누르지 않은상태에서는 obj파일 내에 적혀있는 normal을 기준으로 그림을 그리게된다. 하지만 S키를 누른다면 vertex 주변의 normal들을 모두 더해서 normalize시키는 smooth shading을 그릴수있다. 물론 normal을 모두 다 계산하기때문에 연산량이 매우 증가한다. 3개 이상의 vertex를 가지는 polygon들도 모두 traingulation해서 그리게되는데, 예를들어 vertex의 counterclockwise order로 obj파일에 4 3 2 6 7 8 9 로 face가 나와있으면 (4 3 2) (4 2 6) (4 6 7) (4 7 8) (4 8 9)방식으로 triangulation을 했다. 물론 다른방식으로도 할 수 있겠지만 최소한 연산은 최소화하면서 counterclockwise를 유지하는 방식으로는 제일 좋은 방법이라고 생각했다.

실제로 구현후 사이트에서 찾은 몇몇 파일들을 실행시킨 예시이다.

물론 각각 그림마다 크기의 차이가 존재해서, 몇몇의 사진은 직접 glScalef를 사용후 캡쳐한 사진이다.

각각 normal을 file로부터 읽어왔는지 아니면 normal을 직접계산했는지의 차이에 의해 달라지는 총의 모습을 그렸고

왼쪽은 밀짚모자를 그려본것이며 오른쪽은 아이언맨을 그려봤다.