컴퓨터 그래픽스

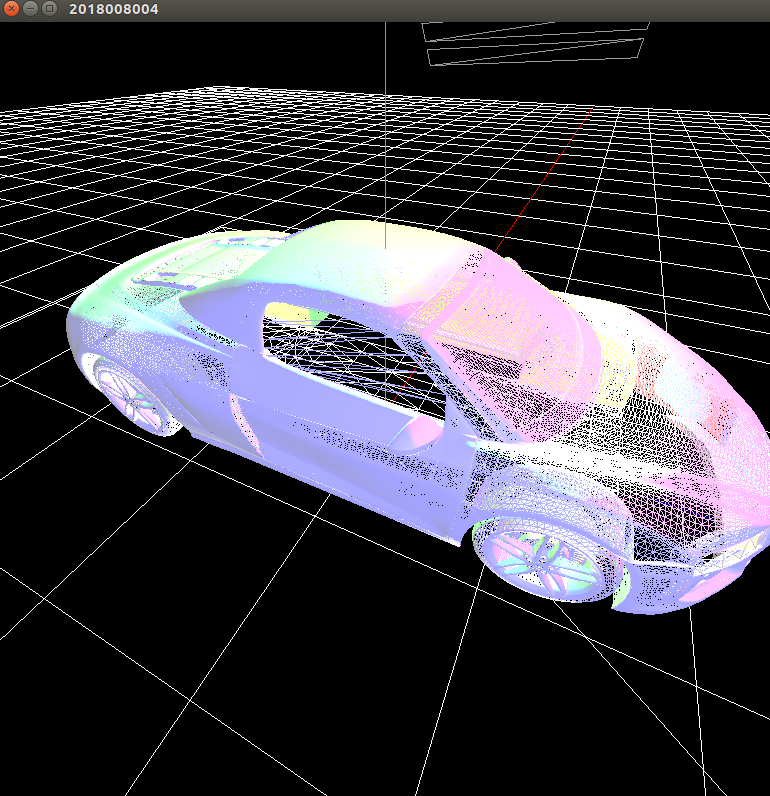
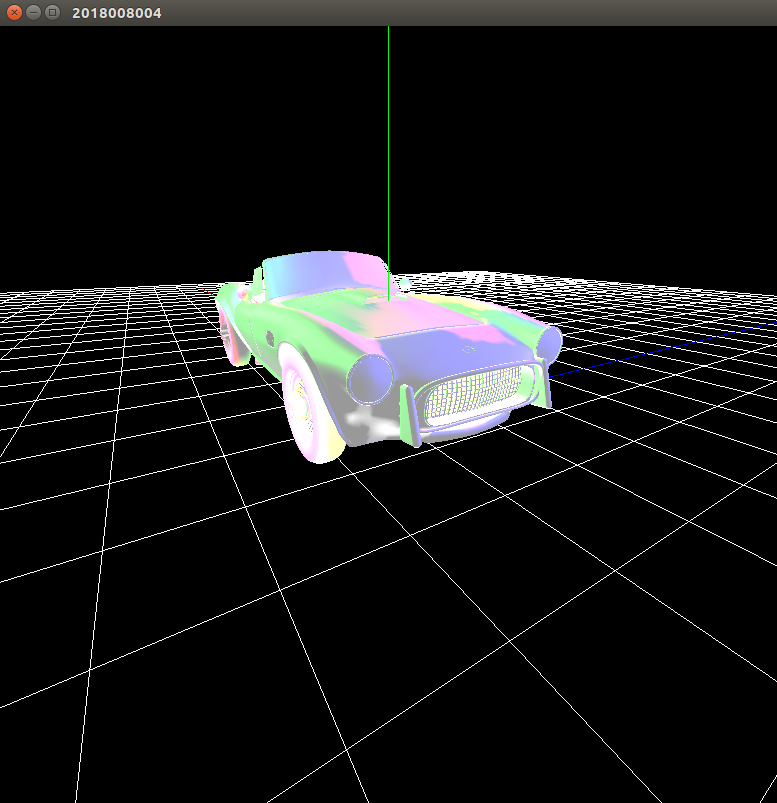
이윤상 교수님

**Report for Class assignment2**

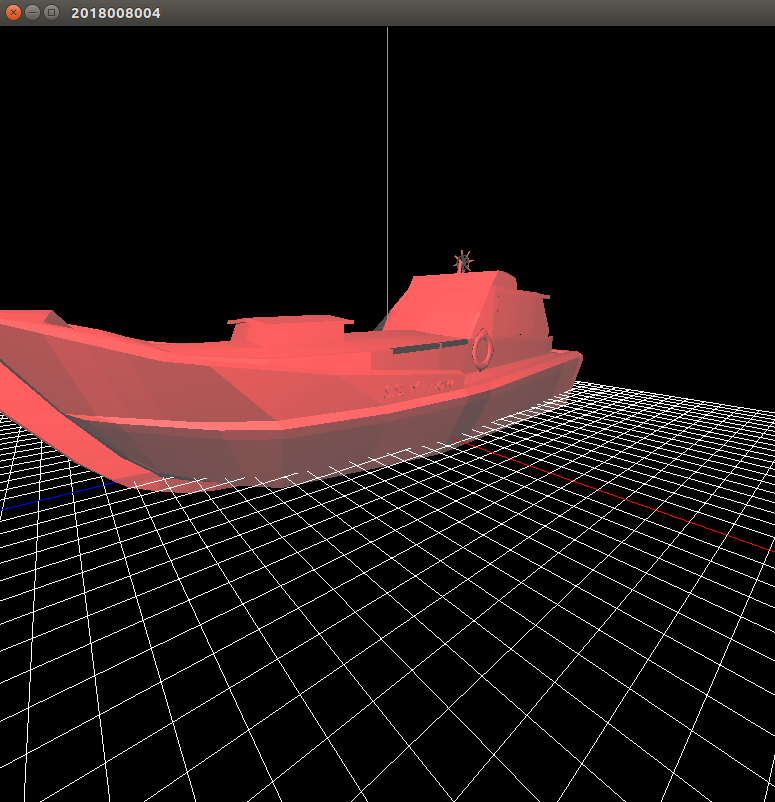
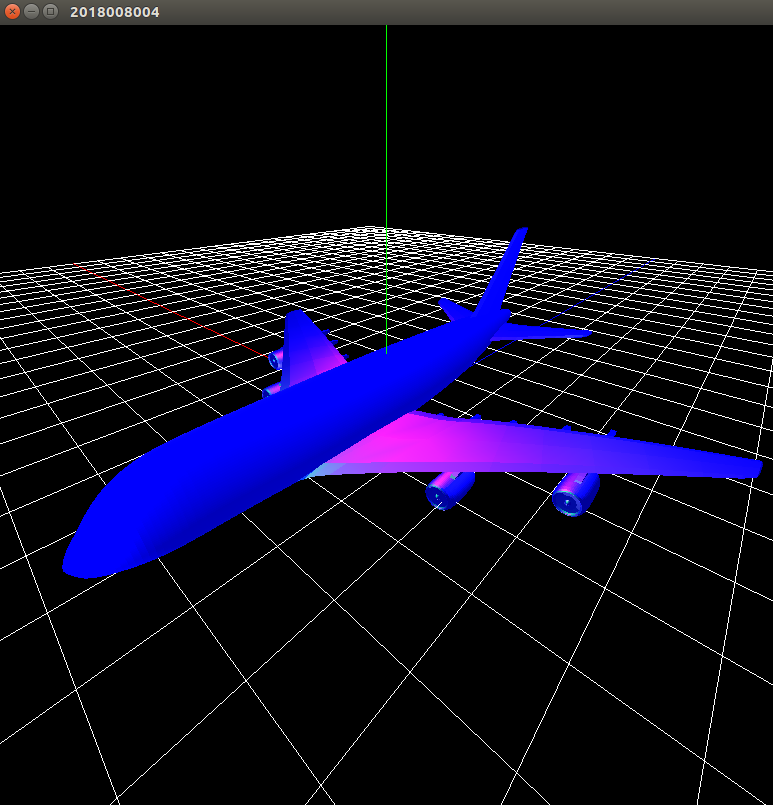
1. **Which requirements you implemented.**
2. Camera manipulation

과제1번의 코드를 skeleton code로 사용하였습니다.

1. Load an obj file and render it
2. Drag-drop을 지원하기 위해 glfwSetDropCallback을 통해 file의 path를 얻어와 여는 방식을 사용했습니다. 다른 obj file을 drag-drop시 이전에 obj file을 지우고 새로운 obj file을 render합니다.
3. Obj file을 읽어오면 vertex, vertex의 normal vector, face index를 각각 gVertex, gNorm, gFace에 담습니다. Face의 index가 3이상이기 때문에 gFace 3배열을 담도록 하였습니다. Mesh를 그리기 위해 glDrawArrays를 사용했습니다.
4. Z 키를 누르면 wireframe, solid mode가 toggle 되게끔 keycall back에서 gDrawWire의 값을 바꿉니다. glPolygonMode의 2번째 argument를 GL\_LINE과 GL\_FILL을 사용했습니다.
5. Obj file을 drag-drop하여 읽어온 경우에 관련된 정보를 console에 출력했습니다.
6. Lighting
7. Light source는 4개로, lightPos에 위치 4개를, lightColor에 4개를 넣어 사용했습니다. 위치는 (2,2,2),(2,2,-2),(-2,2,2),(0,-5,0)로 모두 point light를 사용했고, 빛의 색은 R, G, B, (.5,.5,.5)를 사용했습니다.
8. Extra credits.
9. S키를 눌러 smooth shading과 obj file에서 읽은 normal data로 shading하는 것을 toggle할 수 있습니다. render에서 gSmoothShading에 따라 다르게 mesh를 그립니다. Vertex의 normal vector는 face norm vector를 모두 구하여 해당 vertex를 사용하는 face들의 normal vector를 평균을 내어 구하였습니다. gShading에 glDrawArrays를 위한 정보를 채워 넣었습니다.
10. Vertex를 4개 이상 쓰는 face들을 vertex norm을 유지하며 triangulation해주었습니다. 간단하게 를 새롭게 gFace[0]에 append하는 식으로 구현하였고, n vertex face에 대해 n - 2개의 새로운 face를 만듭니다.
11. Screenshot of obj file render



**[0] [1]**



**[2] [3]**

Reference

[0] <https://free3d.com/3d-model/ac-cobra-269-83668.html>

[1] <https://free3d.com/3d-model/audi-r8-spyder-22646.html>

[2] <https://free3d.com/3d-model/airplane-a380-725748.html>

[3] <https://free3d.com/3d-model/ss-minnow-boat-45365.html>

*End of report*