**1. OpenGL 메시 렌더링**

**(1) 소스 코드**

|  |
| --- |
| for(long faceIDs = 0; faceIDs < nFaces; faceIDs++) {  // Face normal  calFaceNormal(Cube.faces[faceIDs].vertIDs[0] - smfTag,  Cube.faces[faceIDs].vertIDs[1] - smfTag,  Cube.faces[faceIDs].vertIDs[2] - smfTag,  normalTmp);  Cube.normF[faceIDs].norm[0] = normalTmp[0];  Cube.normF[faceIDs].norm[1] = normalTmp[1];  Cube.normF[faceIDs].norm[2] = normalTmp[2];  // Vertex normal  for(int i = 0; i < 3; i++) { // Vertex normal에 영향을 주는 Face normal을 누적한다  Cube.normT[Cube.faces[faceIDs].vertIDs[i]].norm[0] += Cube.normF[faceIDs].norm[0];  Cube.normT[Cube.faces[faceIDs].vertIDs[i]].norm[1] += Cube.normF[faceIDs].norm[1];  Cube.normT[Cube.faces[faceIDs].vertIDs[i]].norm[2] += Cube.normF[faceIDs].norm[2];  }  }  // Vertex normal을 정규화한다(즉, 벡터 크기가 1이 됨)  float len;  for(long vertIDs = 0; vertIDs < nVerts; vertIDs++)  {  len = (float)sqrt(Cube.normT[vertIDs].norm[0]\*Cube.normT[vertIDs].norm[0]  + Cube.normT[vertIDs].norm[1]\*Cube.normT[vertIDs].norm[1]  + Cube.normT[vertIDs].norm[2]\*Cube.normT[vertIDs].norm[2]);  if(len == 0.) len = 1.;  Cube.normT[vertIDs].norm[0] /= len;  Cube.normT[vertIDs].norm[1] /= len;  Cube.normT[vertIDs].norm[2] /= len;  } |

: 파일에서 값을 읽는 초기화 함수에서 미리 Face Normal과 Vertex Normal을 계산해둔다. 이 때, Vertex Normal은 각 점에 영향을 끼치는 Face Normal들의 합을 누적하고 정규화 하는 과정을 거친다.

|  |
| --- |
| if(wireFlag) { // if(wireFlag == GL\_TRUE)  glDisable(GL\_LIGHTING); // 뼈대만 보는 Wirefram  glColor3f(0.3f, 1.0f, 0.3f); // 선색은 초록  for(long faceIDs = 0; faceIDs < nFaces; faceIDs++) {  // 단순 선에 관한 normal이므로 그냥 무조건 Face normal을 이용한다  glNormal3f(Cube.normF[faceIDs].norm[0], Cube.normF[faceIDs].norm[1], Cube.normF[faceIDs].norm[2]);  glBegin(GL\_LINE\_LOOP);  for(int i = 0; i < 3; i++) {  glVertex3f(Cube.verts[Cube.faces[faceIDs].vertIDs[i] - smfTag].x,  Cube.verts[Cube.faces[faceIDs].vertIDs[i] - smfTag].y,  Cube.verts[Cube.faces[faceIDs].vertIDs[i] - smfTag].z);  }  glEnd();  }  }  else { // else if(wireFlag == GL\_FALSE)  glEnable(GL\_LIGHTING); // Wirefram에서 조명을 껐으므로 면을 보기 위해서 조명을 켜자  glBegin(GL\_TRIANGLES);  for(long faceIDs = 0; faceIDs < nFaces; faceIDs++) {  if(!shadeFlag) // if(shadeFlag == GL\_FALSE) // Face normal을 이용하므로 한번만 설정해주면 됨  glNormal3f(Cube.normF[faceIDs].norm[0], Cube.normF[faceIDs].norm[1], Cube.normF[faceIDs].norm[2]);  for(int i = 0; i < 3; i++) {  if(shadeFlag) // if(shadeFlag == GL\_TRUE) // Vertex normal을 이용하므로 점마다 모두 설정해주면 됨  glNormal3f(Cube.normT[Cube.faces[faceIDs].vertIDs[i]].norm[0],  Cube.normT[Cube.faces[faceIDs].vertIDs[i]].norm[1],  Cube.normT[Cube.faces[faceIDs].vertIDs[i]].norm[2]);  glVertex3f(Cube.verts[Cube.faces[faceIDs].vertIDs[i] - smfTag].x,  Cube.verts[Cube.faces[faceIDs].vertIDs[i] - smfTag].y,  Cube.verts[Cube.faces[faceIDs].vertIDs[i] - smfTag].z);  }  }  glEnd();  } |

: Flag에 따라 Face normal/Vertex normal 전환, Wireframe이 되도록 GL\_LINE\_LOOP, GL\_TRIANGELS를 이용해 그린다.

**(2) 실행 화면**

 



