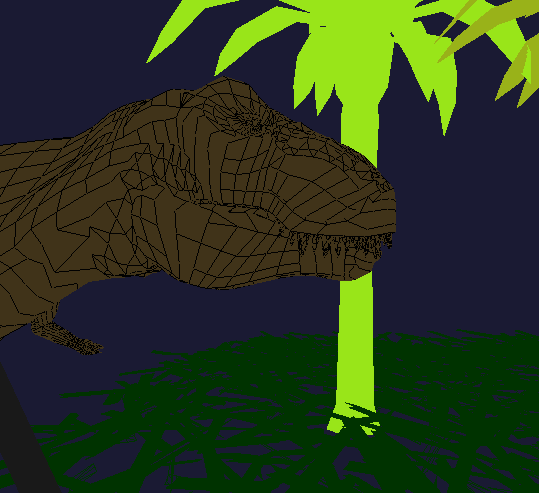
Silhouette



**Description:**

Today you will be working on code to find and display the silhouette edge of a mesh object. The silhouette edge is the edge between faces that are in view and faces that are not in view.

The way to calculate this is to take the dot product of the look vector and the face normal for each face. If this product is positive, then the face is facing towards the camera, but if the product is negative, then the face is facing away from the camera.

To be able to do this calculation, you will need to build a data structure to keep track of all of the edges in the mesh, such that each edge knows which two faces it exists between.

I originally built this using an Edge List which was made up of an array of linked lists. The array index was equal to the lower-numbered vertex in the edge. This guaranteed that only one copy of each edge would be added into the edge list. Since the edge list was built via iteration through the face list, the first time a face contained an edge the edge would be added to the list, and when the second face for that edge was reached, that second face would be added to the edge.

The data structure I built is included but you can build your own if you'd prefer. The Edge List only needs to be built once per .ply file, but it needs to be iterated through every time the silhouette edge is calculated.

Since the viewer allows rotation around the y axis, the look vector has components in X and Z which can be calculated from the angle of rotation (i.e., when rotation=0, lookX=0.0 and lookY=1.0).

**Your Task:**

* Fill in the empty functions in ply.cpp.
  + findEdges fills in the edgeList
  + renderSilhouette uses the edgeList to find and draw all of the edges which are part of the silhouette.

**Files Given:**

ply.cpp – Fill in findEdges and renderSilhouette

ply.h – edgeList is declared here.

geometry.h – the edge struct is declared here.

data/ - some .ply files

Algebra.h

entity.cpp

entity.h

main.cpp

**Compiling: (On the Mac)**

g++ -Wall -Wextra main.cpp ply.cpp entity.cpp -I/Library/Frameworks/GLUI.framework/Headers/ -framework OpenGL -framework GLUT -framework GLUI -o movie