ETGG2801 — Graphics 1
Lab 3 — Diffuse Texture Mapping
Fall 2015

Overview

In this lab, you will be building a model viewer program for viewing textured models. Your program should allow the user to move (along the x, y and z axes) and rotate (about the x, y and z axes) the model.

Tasks

1. Modify the **OBJReader** class to allow it to read Wavefront OBJ files that include UV coordinates.
   - The UV coordinates are stored as two real numbers, preceded by "vt".
   - The faces are stored differently for models containing UV coordinates. Now, each face not only contains an index into the list of vertices, but an index into the list of UV coordinates, as well.
   - Example: f 29/27 27/25 20/18
   - From this example, 29, 27 and 20 are indices into the vertex list, while 27, 25 and 18 are indices into the list of UV coordinates.
2. Implement a **Camera** class for representing your scene's camera (this will be stored in the **Scene** class created in the previous lab). Instead of translating/rotating the model itself, the user will be controlling the camera viewing the model.
   - The camera's transformation matrix will be part of the model-view matrix from the previous labs. The model matrix moves the model from object coordinates to world coordinates, while the view matrix moves the model from world coordinates to camera coordinates (also known as eye coordinates).
   - The projection matrix will depend on the type of camera (orthographic or perspective), and should be configurable by the user (pressing a key will switch between orthographic and perspective projection). This matrix moves the model from camera coordinates to clip coordinates (due to the fact that this space is where an object, or portion of, can be determined to be visible or not, and thus possibly clipped).

Transformations

[WORLD_COORDINATES] = [MODEL_MATRIX] x [OBJECT_COORDINATES]

[CAMERA_COORDINATES] = [VIEW_MATRIX] x [WORLD_COORDINATES]

[CLIP_COORDINATES] = [PROJECTION_MATRIX] x [CAMERA_COORDINATES]

[NORMALIZED_DEVICE_COORDINATES] = [CLIP_COORDINATES] / Wc

Wc = the fourth coordinate in the Vector4 class (Vector4 is in **homogenous coordinates** due to this fourth value).