Assignment 4: Hardware Assignment

Question: [1]: In the diagram above, u can see a 3D pyramid in its own model space. V0, V1, V2, V3 are the three vertices. The height of the pyramid is "d". Two of the faces are perpendicular to each other and V0 coincides with the origin.
You can also see the 3D world space, where three points are marked EC.xyz, MaxV.xyz and MaxH.zyz. These points are the center, the max vertical limit and the max horizontal limit of an ellipse (marked in red). There is also a light source, whose direction from origin is the unit vector -L.ijk
The assignment is as follows.
Write the pseudocode for an animation which rotates the pyramid along the circumference of the ellipse and shades the vertices using vertex shaders in the following manner. If any face of the pyramid is front facing towards the light, the vertices belonging to that face, are shaded Blue, else they are shaded gray. If a vertex is common between a front facing and back facing pyramid face, its painted blue.
Assume vertices are arranged in clockwise manner.

Solution: The pseudo code along with the explanation is given below

Consideration (1): For the 'front-facing' part mentioned in the question, a face is considered front facing if it receives some intensity from the light source which is at least equal to a threshold value (consider it analogous to subtending some solid angle by the face at the source)

Consideration (2): Consider this pseudo code for a single cycle which could be iterated infinitely giving the same results.

Algorithm: Vertex shading in a pyramid revolving along the circumference of an ellipse

(1) Defining the major and minor axes of ellipse (a is the half of major axis and b is the half of minor axis)

$$\begin{aligned} \overrightarrow{a} &= \overrightarrow{Max_H} - \overrightarrow{E_c} \\ \overrightarrow{b} &= \overrightarrow{Max_V} - \overrightarrow{E_c} \end{aligned}$$

Here,

$$\begin{aligned} \overrightarrow{a} &= a \cdot \hat{i_M} \\ \overrightarrow{b} &= b \cdot \hat{j_M} \end{aligned}$$

(i_M,j_M,k_M) form the model coordinate system (with E_c as origin, i_M,j_M for the elliptical plane.

(2) The ellipse can be parametrically defined as (w.r.t parameter 't'):

$$x_M = a \cdot cos(t), y_M = b \cdot sin(t) \tag{1}$$

Also,

$$\overrightarrow{x_M} = x_M \cdot \hat{i_M}, \overrightarrow{y_M} = y_M \cdot \hat{j_M} \tag{2}$$

We can consider 't' as time because for a single cycle, t is the angle increasing from 0 to 2π by the vector $\overrightarrow{a}$ revolving in a circle centered around E_c. This is how the parametric equation was derived actually.

(3) Let H_{BM} be the calculated transformation matrix from Model frame to the Body frame. That is,

$$\overrightarrow{r_M} = \overrightarrow{r} - \overrightarrow{E_c}$$
$$\overrightarrow{r_B} = H_{BM} \cdot \overrightarrow{r_M}$$

H_{BM} comprises of a transformation matrix $T(-E_c)$ followed a rotation matrix, which can be decomposed into X,Y,Z Euler angles.

(4) Now, for the input vertex V_0, which is rotating along the circumference of the ellipse, we can substitute $\overrightarrow{r} = \overrightarrow{V_0}$, from which we can calculate $\overrightarrow{V_{0,M}}$ and subsequently get $\overrightarrow{V_{0,B}}$ as $\overrightarrow{V_{0,B}} = H_{BM} \cdot \overrightarrow{V_{0,M}}$. All the other vertex coordinates of the pyramid can be calculated by applying the same transformation as they are at a fixed distance from V_0, behaving like a rigid body. Or, in the body frame, they are on the orthogonal axes at a distance 'd', which can be considered directly.

(5) Now that we've set our pyramid in rotation with the coordinates varying with respect to the parameter 't' corresponding to time (we can set a frequency separately), let us go to the vertex shading part.
Taking our earlier mentioned considerations in mind, there are two possible cases:

- **Case (1):** One of the vertex is shaded gray, rest all are shaded blue
- **Case (2):** All the vertices are shaded blue

This can be justified as follows:

We can see that the pyramid has two different kinds of sides; three of one type and one of other type. Each vertex is a part of three faces, corresponding to three normals.

Now, most of the times, there will be one vertex which will be gray because in most of the orientations of the pyramid along its path, the light source will luminify either 3, or 2 or 1 face ((out of the two kinds), this happens when the face normal alings with $-\overrightarrow{L}$).

Sometimes, when the normal aligns with the sum of the 3 faces of same kind, or is nearly aligned, all four vertices will be shaded . i.e., $(\hat{n}_1 + \hat{n}_2 + \hat{n}_3) \cdot (\overrightarrow{-L}) \approx 0$

(6) We'll use a simple technique to determine whether a vertex be shaded blue or black. As we have three normal vectors corresponding to a vertex (face normals), we'll take the dot product of these three with the light source unit vector from origin ($\overrightarrow{-L}$). We'll do this for all four vertices. The one with the least value will be shaded gray.

$$d = (\hat{n}_i + \hat{n}_j + \hat{n}_k) \cdot (\overrightarrow{-L}) \tag{3}$$

Where, $i, j, k \epsilon S$ - $[l]$; $S = 0, 1, 2, 3$ and l is the vertex for which we are calculating the sum of dot products. (Face j is opposite to vertex j)

When all the four values are equal, i.e. $(\hat{n}_1 + \hat{n}_2 + \hat{n}_3) \cdot (\overrightarrow{-L}) = 0, \hat{n}_i; i = 1, 2, 3$ will subtend $45°$ with $\overrightarrow{-L}$

We can define some threshold for dot product here, e.g. $|d| < 0.1$, which can be based on empirical calculations according to our needs.

(7) For the last part, in the vertex buffer code we use, we just need to make a small change. Here, we need to separately consider the inputs for the vertex shader pipeline, i.e. The normal vectors at each vertex from all the adjacent faces.

(8) The remaining operations in the vertex shader architecture remain the same. These include normalization and clip space transformation. We can use the float4 representation for homogeneous coordinates and float3 representation for regular unit vectors, normals.
Conclusion: This approach can be used to complete the given task in the problem.

****

Soham Kulkarni, EE19BTECH11053, IIT Hyderabad