Lab Exercise Date 05-04-2018

1. Write a C-program to implement the following shading algorithms for a cube object. Given the light source and the observer at the same point.
   a. Flat/Uniform Shading
   b. Gouraud shading

Instructions

For making cube you can use **bar3d(100,100,200, 200, 100, 1)**

Flat Shading

● Flat shading (also known as faceted shading [6]) is, perhaps, the simplest shading model. With flat shading, each entire polygon is drawn with one color.

● It tends to produce a blocky, unrealistic look.

● To calculate the color of the face, we need to know the normal of the polygon. Once per polygon the normal is used to calculate the influence of lights in the scene and then the entire facet is painted with the resulting color.

● It is very inexpensive to compute. It was used in early 3D video games like Star Fox and Virtual Racing because it was the only practical way to shade an object with the available computational resources at the time.

● It is frequently used for prototyping and testing because it can be rendered so efficiently.

Gouraud Shading Algorithm

```
1.  Determine the average unit normal vector at each vertex of the polygonal surface.
2.  Let $C_R$ = color of the rightmost edge pixel on the ith scan line, $x_r$ = the x-coordinate of
    the rightmost edge pixel, $C_L$ = color of the leftmost edge pixel on the ith scan line and
    $x_l$ = the x-coordinate of the leftmost edge pixel. Compute and store the scan line
    constant $C_i = (C_R – C_L)/(x_r – x_l)$ for each scan line that lies within the projected area of
    polygon.
3.  Let $C_T$ = color of the topmost vertex pixel on the jth left edge, $y_t$ = the y-coordinate of
    the topmost vertex pixel of the edge, $C_B$ = color of the leftmost vertex pixel of the jth
    left edge and $y_l$ = the y-coordinate of the leftmost vertex pixel of the edge. Compute
    the edge constant $C_j = (C_B – C_T)/(y_t – y_l)$ for all the left edge of the polygon.
4.  Initialize a temporary variable color = 0.0.
5.  for each scan line i within the polygonal area starting from the top do
6.    for each pixel on the scan line within the projected surface area, starting from left do
7.      if it is a vertex pixel then
8.        color = color computed using the lighting model.
9.      else if it is a pixel on a left edge then
10.       color = color of the pixel on the previous scan line and same edge + $C_j$.
11.     else
```

```
12.         color = color of the previous pixel on the same scan line + $C_i$.
13.      **end if**
14.    **end.**
15.  **end.**
```