COMP371 – Computer Graphics – TA2 – Winter 2012 Issued: January 30, 2012
Instructor: S. Mokhov Informally Due: February 13, 2012, 13:15

Theory Assignment 2

This is the second warm up theory assignment to primarily prepare yourselves for Quiz 2 from some theoretical concepts we have learned as well as a bit more of OpenGL. Most of it can be answered from the slides with some numerical modifications. The OpenGL part does not really need to be fully written and compiled, but it's good to do so as a practice. This is an informal assignment without marks assigned, and servers only as a preparatory measure for the quiz highlighting points to focus on.

Part I
Theory

1. Given an eye position at $e = (3, 3, 1)$, a look-at point $a = (3, 4, 1)$ and an up-vector $\vec{v}_{up} = (0, 0, 1)$, compute the corresponding world-camera transformation matrix $M_{WC \to VC}$. Show your intermediate components $T, \vec{n}, \vec{u}, \vec{v}, R$.

   Determine the view space coordinates of the following world space points.

   (a) $(3, 1, 2)$
   (b) $(4, 1, -1)$
   (c) $(0, 0, 0)$

2. Consider the combination $P = \alpha_1 \cdot P_1 + \alpha_2 \cdot P_2 + \alpha_3 \cdot P_3$. Show the three cases to make a test whether P is outside, inside, or on the edge of the triangle formed from P_1, P_2, P_3.

3. Use the Midpoint Line Conversion algorithm to perform scan conversion of the cubic curve $y = ax^3$ in the interval $-\frac{1}{\sqrt{3a}} \leq x \leq \frac{1}{\sqrt{3a}}$.

4. Describe the meaning of each component (each letter) of the following color models and how one can convert from one to another: RGB, CMY, HSV, YIQ.

5. Define the Cohen-Sutherland line-clipping algorithm, and use it to clip a line segment ad similar to the lecture slides except that a is located in 0110 and c and b are located exactly at the opposite corners of the clipping window, while d is still in 1001, but adjusted such that all a, b, c, d still lie on the same straight line.

6. Define the Sutherland-Hodgeman polygon clipping algorithm.

Part II
OpenGL

1. Enumerate all the multiple matrix stacks in OpenGL and describe each one's purpose.

2. For your small program from TA1, fill in the triangle with a darker shade of gray to a lighter shade of gray and make it translucent with the opacity value of 0.5.