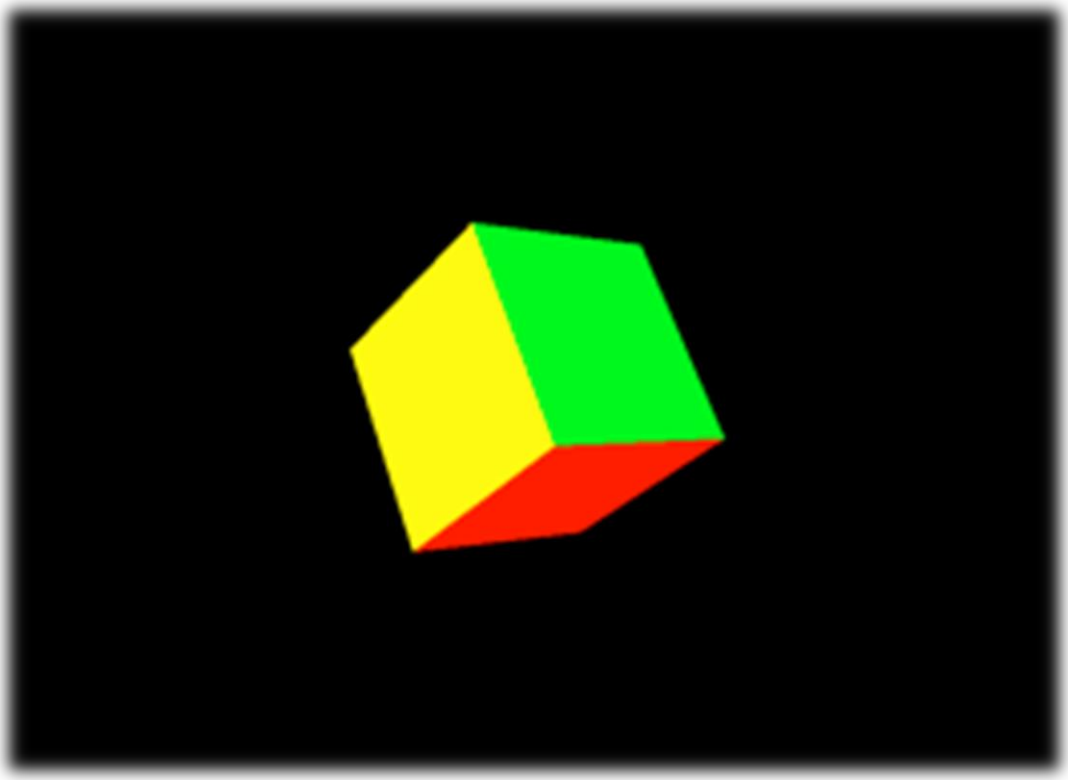
OpenGL

Assignment Report



Priyank Joshi

17812234

30/10/2016

Introduction

The ability to produce an effective scene in OpenGL using C programming, having complex and simple objects in the scene and animations controlled by the user using the key (keyboard).

Design Features

1. Drawing Objects

The program used GLUT functions to create objects such as the toy simulator in the middle, snow man (right side) and penguin (left side) shown in the scene. These GLUT functions that were used in the program were glutSolidTetrahedron(), glutSolidCube(), glutSolidTorus(), glutSolidSphere() and glutSolidCone() with a combination of glTranslatef(), glScalef() and glRotatef() produced composite objects to create toy simulator, snowman, penguin and pyramids (at the background). These composite objects consist of the simple objects named above, with combination of translation, scaling and rotation. Along with material properties of the objects using the functions glMaterialfv() giving specular and shininess gives surface finish of each objects in the scene to look attractive and realistic. Using glEnable(GL_COLOR_MATERIAL) enables the object colors in the scene. Additionally, glEnable(GL_DEPTH_TEST) is to enable getting realistic objects drawn in the scene. Without that it could result in ugly objects and not realistic at all.

On the other hand, transparency has also been included in this program. This is done by glEnable(GL_BLEND) and glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA) functions and hence using glColor4f() for the red, the last parameter which is Alpha, gives the effect of transparencies, which ranges from 0.0 to 1.0.

2. Background and Light

The background is set to black color using glClearColor(), this is so because the objects drawn in the scene will stand out, this is due to the light sources used, therefore, drawn objects can be seen. There are 4 light sources that are used in this program. These are; Ambient, Diffuse, Specular, and Position. Additionally, light sources do have a relationship with Red, Green and Blue colors (RGB) that brings in the effects of the light source colors. So for this program, ambient light was used where by the it gives a universal light source, does

not matter where the objects are, and it gets scattered in all directions equally. Diffuse lighting makes the light (rays) come from one from one direction. Specular lighting comes from a particular direction, and is shown greatly in shiny surfaces of objects because it shows the bouncing off from the surface in a specific direction. Furthermore, position light just to specify the position of the light source. glEnable(GL_LIGHT0) enables the light sources (white). Moreover, glEnable(LIGHTING) enables the lighting calculations.

3. Key Functions and Animation

The key functions have also been implemented and printed on the terminal for the user. However, not all functions are working as per the requirement, it has some defects in it. Additionally, this key function is in the keyboard(unsigned char key, int x, int y) and is called in the main() function to enable key events. The defect as mentioned before is that for example when the user presses X or x, it does not do the whole animation. In other words, the user has to keep pressing the key in order for the full animation to occur. So therefore when user keeps holding ‘x’ or ‘X’ it rotates the scene along the x-axis. When user keeps holding ‘y’ or ‘Y’ it rotates the scene about the y-axis. Additionally, when the user keeps holding ‘Z’ it will zoom in and ‘z’ will zoom out. When the user keeps pressing ‘F’ or ‘f’ the animation (rotates about the y-axis) faster, while holding ‘S’ or ‘s’ makes the rotation about the y-axis (animation) slower. Having the key ‘t’ or ‘T’ will stop the rotation. Having the pressed the ‘Esc’ key will instantly exit the program.

4. Reshape Function

The reshape function is called in the main() function once when the program first runs and everytime your window is reshaped. Additionally, the glViewPort function enables mapping to the new size in the window without the scene getting damaged. glMatrixMode() specifies which matrix is the current matrix and glLoadIdentity() just loads the current matrix with the identity matrix. gluPerspective() is to set the correct perspective.

References

C++ OpenGL #1 - Some basic GLUT. (2016). *YouTube*. Retrieved 30 October 2016, from https://www.youtube.com/watch?v=SAmD_Aq1Un4

Li, Ling. Computer Graphics Lecture notes (2016). Curtin University.

OpenGL Programming Guide : Table of Contents. (2016). *Glprogramming.com*. Retrieved 30 October 2016, from http://www.glprogramming.com/red/index.html

OpenGL Video Tutorial - Home. (2016). *Videotutorialsrock.com*. Retrieved 30 October 2016, from http://www.videotutorialsrock.com/

Tutorials. (2016). *Lighthouse3d.com*. Retrieved 30 October 2016, from http://www.lighthouse3d.com/tutorials/

(2016). Retrieved 30 October 2016, from http://tobias.isenberg.cc/graphics/LabSessions/OpenGLProject