LAB REPORT

Course No: CSE 4202.

Course title: Sessional based on 4201.

Submitted to:
Mahit kumar paul
Lecturer,
Department of Computer Science and Engineering,
Rajshahi University of Engineering & Technology.

Submitted by:
Arun kundu
Roll: 143045
Section: A
4th Year, even Semester.
Department of Computer Science and Engineering.
Rajshahi University of Engineering & Technology.

TASK OF MAIN FUNCTION’S PARAMETER :

The parameters to main represent the command line parameters provided to the program when it was started.

The argc parameter represents the number of command line arguments.

The char **argv[] is an array of strings (character pointers) representing the individual arguments provided on the command line.

TASK OF FUNCTIONS USED IN MAIN():

glutinit :

glutInit will initialize the GLUT library and negotiate a session with the window system. During this process, glutInit may cause the termination of the GLUT program with an error message to the user if GLUT cannot be properly initialized.

glutInit also processes command line options, but the specific options parse are window system dependent.

glutInitDisplayMode:

glutInitDisplayMode sets the initial display mode.

The initial display mode is used when creating top-level windows, subwindows, and overlays to determine the OpenGL display mode for the to-be-created window or overlay.

There are many types of display mode such as GLUT_RGBA, GLUT_INDEX, GLUT_SINGLE, GLUT_DOUBLE, GLUT_ALPHA etc.

glutInitDisplayMode(GLUT_SINGLE);

glutInitWindowSize :

glutInitWindowSize set the initial window size.

Width in pixels. Height in pixels.

void glutInitWindowSize(int width, int height).

glutInitWindowPosition :

glutInitWindowPosition set the initial window position.

x Window X location in pixels. y Window Y location in pixels.

void glutInitWindowPosition(int x, int y);

glutCreateWindow: Creates a top-level window. The name will be provided to the window system as the window's name.

int glutCreateWindow(char *name);

glutDisplayFunc :

Sets the display callback for the current window. When GLUT determines that the normal plane for the window needs to be redisplayed, the display callback for the window is called.

void glutDisplayFunc(void (*func)(void));

glutMainLoop :

Enters the GLUT event processing loop. This routine should be called at most once in a GLUT program. Once called, this routine will never return. It will call as necessary any callbacks that have been registered.

void glutMainLoop(void);

TASK OF FUNCTIONS USED IN DISPLAY():

glBegin: :

Specifies the primitive or primitives that will be created from vertices presented between glBegin and the subsequent glEnd. Ten symbolic constants are accepted: GL_POINTS, GL_LINES,GL_LINE_STRIP, GL_LINE_LOOP, GL_TRIANGLES, GL_TRIANGLE_STRIP, GL_TRIANGLE_FAN, GL_QUADS,GL_QUAD_STRIP, and GL_POLYGON.

void glBegin(GLenum mode);

glVertex :

glVertex commands are used within glBegin/glEnd pairs to specify point, line, and polygon vertices. The current color, normal, texture coordinates, and fog coordinate are associated with the vertex when glVertex is called.

Void glVertex2f (GLfloat x, Glfloat y);

glFlush :

Force execution of GL commands in finite time.

Different GL implementations buffer commands in several different locations, including network buffers and the graphics accelerator itself. glFlush empties all of these buffers, causing all issued commands to be executed as quickly as they are accepted by the actual rendering engine. Though this execution may not be completed in any particular time period, it does complete in finite time.

Void glFlush(Void);

OPENGL :

Opengl short for "Open Graphics Library" is an application programming interface (API) designed for rendering 2D and 3D graphics. It provides a common set of commands that can be used to manage graphics in different applications and on multiple platforms.

By using OpenGL, a developer can use the same code to render graphics on a Mac, PC, or mobile device. Nearly all modern operating systems and hardware devices support OpenGL, making it an easy choice for graphics development. Additionally, many video cards and integrated GPUs are optimized for OpenGL, allowing them to process OpenGL commands more efficiently than other graphics libraries.