**OpenGL:**  OpenGL is the premier environment for developing portable, interactive 2D and 3D graphics applications. It provides a common set of commands that can be used to manage graphics in different applications and on multiple platforms. Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), bringing thousands of applications to a wide variety of computer platforms.

Silicon Graphics Inc., (SGI) began developing OpenGL in 1991 and released it on June 30, 1992; applications use it extensively in the fields of computer-aided design (CAD), virtual reality, scientific visualization, information visualization, flight simulation, and video games. Since 2006 OpenGL has been managed by the non-profit technology consortium Khronos Group.

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.

**glut.h:** It denotes GLUT (OpenGL Utility Toolkit) header file. GLUT is a window system independent toolkit for writing OpenGL programs. It implements a simple windowing application programming interface (API) for OpenGL. GLUT makes it considerably easier to learn about and explore OpenGL programming. GLUT provides a portable API so programmer can write a single OpenGL program that works across all PC and workstation OS platforms. The GLUT library has both C, C++ (same as C), FORTRAN, and Ada programming bindings.

The toolkit supports:

* Multiple windows for OpenGL rendering
* Callback driven event processing
* Sophisticated input device
* An 'idle' routine and timers
* A simple, cascading pop-up menu facility
* Utility routines to generate various solid and wire frame objects
* Support for bitmap and stroke fonts
* Miscellaneous window management functions

**int main(int argc, char\*\* argv):** argv is a pointer to an array of null terminated strings, and argc says how large this array is. They are automatically passed to programmer when he/she starts the program and enter main(). argv[0] is a pointer to a string which holds the name of the executable file, including full path. argv[1] is the first argument, programmer passes to the program when starting it, and so on.

**glutInit(&argc,argv):** glutInit is used to initialize the GLUT library. Here, GLUT library is initialized with unmodified argc and argv variable from main. glutInit also negotiates 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.

**glutInitDisplayMode(GLUT\_SINGLE):** glutInitDisplayMode sets the initial display mode. **GLUT\_SINGLE** is argument to glutInitDisplayMode. GLUT\_SINGLE denotes bit mask to select a single buffered window. This is the default if neither GLUT\_DOUBLE or GLUT\_SINGLE are specified.

**glutInitWindowSize() :** glutInitWindowSize sets the window size. void glutInitWindowSize(int width, int height); width : width in pixels, height: height in pixels.

**glutInitWindowPosition**():glutInitWindowPosition sets the initial window position.

void glutInitWindowPosition(int x, int y); x: window x location in pixels, y: window y location in pixels.

**glutCreateWindow():** glutCreateWindow creates a top-level window.

int glutCreateWindow(char \*name); name: ASCII character string for use as window name.

**glutDisplayFunc():** glutDisplayFunc sets the display callback for the current window.

void glutDisplayFunc(void (\*func)(void)); func: the new display callback function.

**glBegin():** glBegin delimit the vertices of a primitive or a group of like primitives.

void glBegin( GLenum mode); mode 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.

**glVertex2f():** it specifies a vertex. void glVertex2f( GLfloat x, GLfloat y ); x specifies the x-coordinate of a vertex. Y specifies the y-coordinate of a vertex.

**glEnd():**The glBegin and glend functions delimit the vertices of a primitive or a group of like primitives. This function has no parameters. This function does not return a value.

**glFlush():** The glFlush function forces execution of OpenGL functions in finite time. This function has no parameters. This function does not return a value.

glutMainLoop(): glutMainLoop enters the GLUT event processing loop.

void glutMainLoop(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.

**References:**

[01]. <https://www.opengl.org/about/>

[02]. <https://techterms.com/definition/opengl>

[03]. <https://en.wikipedia.org/wiki/OpenGL>

[04]. <https://www.opengl.org/resources/libraries/glut/>

[05].https://www.opengl.org/discussion\_boards/showthread.php/134762-What-are-argcp-and- argv-in-Glutinit-function

[06]. <https://www.opengl.org/resources/libraries/glut/spec3/node10.html>

[07]. https://www.opengl.org/resources/libraries/glut/spec3/node12.html

[08]. https://www.opengl.org/resources/libraries/glut/spec3/node11.html

[09]. https://www.opengl.org/resources/libraries/glut/spec3/node16.html

[10]. https://www.opengl.org/resources/libraries/glut/spec3/node46.html

[11]. https://www.khronos.org/registry/OpenGL-Refpages/gl2.1/xhtml/glBegin.xml

[12]. https://docs.microsoft.com/en-us/windows/desktop/opengl/glvertex2f

[13]. https://docs.microsoft.com/en-us/windows/desktop/opengl/glend

[14]. https://docs.microsoft.com/en-us/windows/desktop/opengl/glflush

[15]. <https://www.opengl.org/resources/libraries/glut/spec3/node14.html>

Heaven`s Light is Our Guide

Rajshahi University of Engineering & Technology



Assignment on: **Basic Study on OpenGL.**

Course Title : Computer Graphics and Animations.

Course No. : CSE4201

Submitted To:

Mahit Kumar Paul

Lecturer

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology.

Submitted By:

Utsha Das

Roll: 143034

Department of Computer Science & Engineering

Rajshahi University of Engineering & Technology.