

**Rohan Nyati**  
**500075940**  
**R177219148**  
**B-5 AI&ML SEM-5**

## **Experiment-1**

### **→ What is OpenGL?**

Open Graphics Library (OpenGL) is a cross-language cross-platform API for rendering 2D and 3D Vector Graphics. OpenGL API is designed mostly in hardware.

This API is defined as a set of functions which may be called by the client program. Although functions are similar to those of the C programming language but it is language independent. The earlier version consisted of a library called OpenGL utility library. Later to make things easier libraries like OpenGL utility Toolkit, GLEE, GLEW and Gliding were added. The implementation is through Mesa 3D, an open source implementation of OpenGL. It can also be implemented through codeblocks. It can do pure software rendering and it may also use hardware acceleration on BSD, Linux, and other platforms by taking advantage of Direct Rendering Infrastructure.

### **→ What is GLU/GLUT?**

The OpenGL Utility Toolkit (GLUT) is a library of utilities for OpenGL programs, which primarily perform system-level I/O with the host operating system. Functions performed include window definition, window control, and monitoring of keyboard and mouse input. Routines for drawing a number of geometric primitives (both in solid and wireframe mode) are also provided, including cubes, spheres and GLUT also has some limited support for creating pop-up menus.

The two aims of GLUT are:

- 1) Allow creation of rather portable codes between operating systems
- 2) Make learning OpenGL easier.

The OpenGL Utility Library (GLU) is a computer graphics library for OpenGL.

It consists of a number of functions that use the base OpenGL library to provide higher-level drawing routines from the more primitive routines that OpenGL provides. It is usually distributed with the base OpenGL package. GLU is not implemented in the embedded version of the OpenGL package, OpenGL ES.

It allows mapping between screen and world coordinates, drawing of quadric surfaces and polygons, gives an extended range of transformation routines for setting up viewing volumes and simple positioning of the camera. It also provides additional primitives for use in OpenGL applications, including spheres, cylinders and disks.

### → **What is OpenGL architecture?**

The architecture of OpenGL is based on a client-server model. An application program written to use the OpenGL API is the "client" and runs on the CPU. The implementation of the OpenGL graphics engine is the "server" and runs on the GPU. Geometry and many other types of attributes are stored in buffers called Vertex Buffer Objects. These buffers are allocated on the GPU and filled by your CPU program.

### **CODE:**

```
#include<GL/freeglut.h>
```

```
void init(){  
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);  
    glutInitWindowSize(640,480);  
    glutInitWindowPosition(1000,200);  
    glutCreateWindow("Simple Window");  
}
```

```
void display()  
{  
    glClearColor(1.0,1.0,1.0,0.0);  
    glClear(GL_COLOR_BUFFER_BIT);  
    glFlush();  
}
```

```
int main(int argc,char **argv)  
{  
    glutInit(&argc,argv);  
    init();  
    glutDisplayFunc(display);  
    glutMainLoop();  
  
    return 0;  
}
```

