

## CHAPTER 2

### OpenGL

#### 2.1 Introduction to OpenGL

OpenGL (Open Graphics Library) is a hardware-independent, operating system independent, vendor neutral graphics API specifications. It is a standard specification defining a cross language, cross platform API for writing applications that produce 2D and 3D computer graphics. Many vendors provide implementation of this specification for a variety of hardware platforms. The interface consists of over 250 different function calls which can be used to draw complex three-dimensional scene from simple primitives. OpenGL has been designed using a client/server paradigm, allowing the client application and the graphics server controlling the display hardware to exist on the same or separate machines. OpenGL was developed by Silicon Graphics Inc. (SGI) in 1992 and is widely used in CAD, virtual reality, scientific visualization, and flight simulation. It is used in video games, where it competes with Direct3D on Microsoft platforms. OpenGL is managed by a non-profit technology consortium, the Khronos group.

GLUT is designed to fill the need for a window system independent programming interface for OpenGL programs. Removing window system operations from OpenGL is a sound decision because it allows the OpenGL graphics system to be retargeted to various systems including powerful but expensive graphics workstations as well as mass-production graphics systems like video games, set-top boxes for interactive television, and PCs. GLUT simplifies the implementation of programs using OpenGL rendering. The GLUT routines also take relatively few parameters.

Some features of OpenGL include the following:

- Geometric and raster primitives
- RGBA or color-index mode
- Display list or immediate mode
- Viewing and modelling transformations
- Lighting and shading
- Hidden surface removal (Depth Buffer)
- Texture Mapping

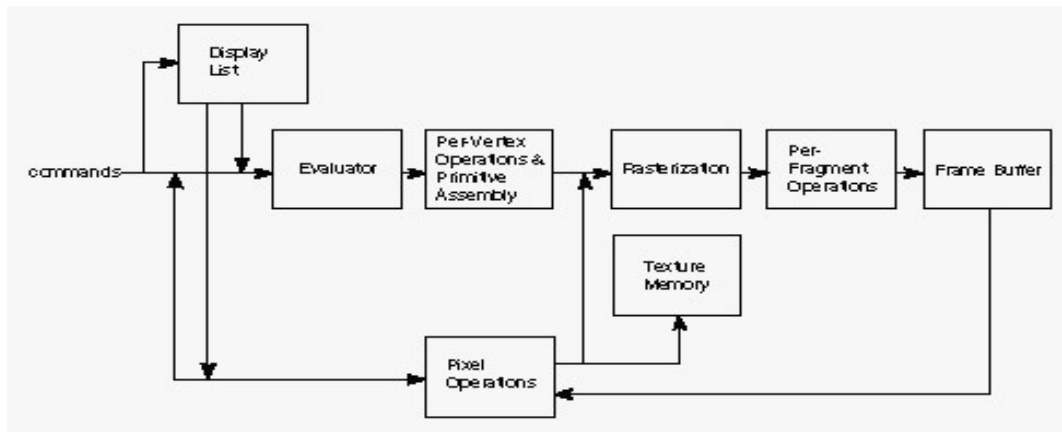


Fig 2.1: OpenGL Block Diagram

The figure shown above gives an abstract, high-level block diagram of how OpenGL processes data. In the diagram, commands enter from the left and proceed through what can be thought of as a processing pipeline. Some commands specify geometric objects to be drawn, and others control how the objects are handled during the various processing stages.

We can choose to accumulate some of commands in a display list for processing at a later time. The evaluator stage of processing provides an efficient means for approximating curve and surface geometry by evaluating polynomial commands of input values. Rasterization produces a series of frame buffer addresses and associated values using a two-dimensional description of a point, line segment, or polygon. Each fragment so produced is fed into the last stage, per-fragment operations, which perform the final operations on the data before it's stored as pixels in the frame buffer.

## 2.2 Limitation

- OpenGL is case-sensitive.
- Line Color, Filled Faces and Fill Color not supported.
- Bump mapping is not supported.
- Navigation render is not supported.
- 3D measurement is not supported.
- Streaming of individual 3D objectives is not supported.

## 2.3 Advantages of OpenGL

- **Stable**

OpenGL implementation has been available for more than seven years on a wide variety of platforms. Additions to the specification are well controlled and proposed updates are announced in time for developers to adopt changes. A backward compatibility requirement ensures that existing applications do not become obsolete.

- **Reliable and portable**

All OpenGL applications produce consistent visual display result on OpenGL API-complaint hardware, regardless of operating system or windowing system.

- **Easy to use**

OpenGL is well structured with an intuitive design and logical commands. Efficient OpenGL routines typically result in application with fewer lines of code than those that make up programs generated using other graphics libraries or packages. In addition,OpenGL drives encapsulate information about the underlying hardware, freeing the application developer from having to design for specific hardware features.

- **Well-documented**

Numerous books have been published about OpenGL, and a great deal of sample code is readily available, making information about OpenGL inexpensive and easy to obtain.

- **Industry standard**

An independent consortium, the OpenGL architecture review board, guides the OpenGL specification. With board industry support,OpenGL is the truly open,vendor-neutral, multiplatform graphics standard.