**Chapter 1**

**INTRODUCTION**

**1.1 Introduction to Computer Graphics**

Computer Graphics is concerned with all aspects of producing pictures or images using a computer. We can create images that are indistinguishable from photographs of real objects. In other terms, Computer Graphics re the graphics created by the computers, and more generally, the representation and manipulation of image data by a computer.

The development of computer graphics has been driven both by the needs of the user community and by advances in hardware and software.

Typically, the term Computer Graphics refers to several different things.

* The representation and manipulation of image data by a computer.
* The various technologies used to create and manipulate images.
* The images so produced, and manipulating visual content.

**1.2 History of Computer Graphics**

The phrase Computer Graphics was coined in 1960 by William Fetter, a graphic designer for Boeing. The field of Computer Graphics developed with the emergence of computer graphics hardware. Early projects like the Whirlwind and SAGE projects introduced the CRT as a viable display and interaction interface and introduced the light pen as an input device.

Further advances in computing led to greater advancements in interactive computer graphics. In 1959, the TX-2 computer was developed at MIT's Lincoln Laboratory. A light pen could be used to draw sketches on the computer using Ivan Sutherland’s revolutionary Sketchpad software.

Also in 1961 another student at MIT, Steve Russell, created the first video game, Spacewar. E. E. Zajac, a scientist at Bell Telephone Laboratory (BTL), created a film called "Simulation of a two-giro gravity attitude control system" in 1963. In this computer generated film, Zajac showed how the attitude of a satellite could be altered as it orbits the Earth. Many of the most important early breakthroughs in computer graphics research occurred at the University of Utah in the 1970s.

The first major advance in 3D computer graphics was created at UU by these early pioneers, the hidden-surface algorithm. In order to draw a representation of a 3D object on the screen, the computer must determine which surfaces are "behind" the object from the viewer's perspective, and thus should be "hidden" when the computer creates (or renders) the image.

Graphics and application processing were increasingly migrated to the intelligence in the workstation, rather than continuing to rely on central mainframe and mini-computers. 3D graphics became more popular in the 1990s in gaming, multimedia and animation. Computer graphics used in films and video games gradually began to be realistic to the point of entering the uncanny valley. Examples include the later *Final Fantasy* games and animated films like *The Polar Express*.

**1.3 Applications of Computer Graphics**

The development of computer graphics has been driven both by the needs of the user community and by advances in hardware and software. The applications of computer graphics are many and varied. We can however divide them into four major areas.

* Display of information: More than 4000 years ago, the Babylonians developed floor plans of buildings on stones. Today, the same type of information is generated by architects using computers. Over the past 150 years, workers in the field of statistics have explored techniques for generating plots. Now, we have computer plotting packages. Supercomputers now allow researchers in many areas to solve previously intractable problems. Thus, Computer Graphics has innumerable applications.
* Design: Professions such as engineering and architecture are concerned with design. Today, the use of interactive graphical tools in CAD, in VLSI circuits, characters for animation have developed in a great way.
* Simulation and animation: One of the most important uses has been in pilots’ training. Graphical flight simulators have proved to increase safety and reduce expenses. Simulators can be used for designing robots, plan it’s path, etc. Video games and animated movies can now be made with low expenses.
* User interfaces: Our interaction with computers has become dominated by a visual paradigm. The users’ access to internet is through graphical network browsers. Thus Computer Graphics plays a major role in all fields.

**1.4 Introduction to OpenGL**

OpenGL is a software interface to graphics hardware. This interface consists of about 150 distinct commands that are used to specify the objects and operations needed to produce interactive three-dimensional applications. OpenGL is designed as a streamlined hardware-independent interface to be implemented on many different hardware platforms.

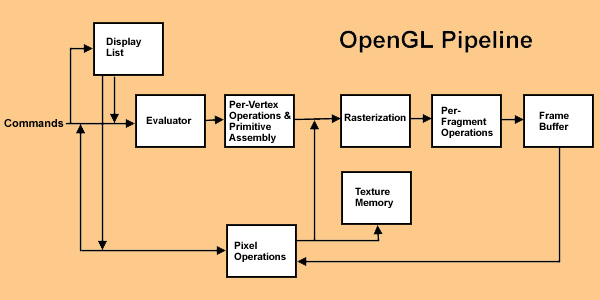
These are certain characteristics of OpenGL:

* OpenGL is a better documented API.
* OpenGL is much easier to learn and program.
* OpenGL has the best demonstrated 3D performance for any API.

The OpenGL specification describes an abstract API for drawing 2D and 3D graphics. Although it's possible for the API to be implemented entirely in software, it's designed to be implemented mostly or entirely in hardware.

In addition to being language-independent, OpenGL is also platform-independent. The specification says nothing on the subject of obtaining, and managing, an OpenGL context, leaving this as a detail of the underlying windowing system. For the same reason, OpenGL is purely concerned with rendering, providing no APIs related to input, audio, or windowing.

OpenGL is an evolving API. New versions of the OpenGL specification are regularly released by the Khronos Group, each of which extends the API to support various new features. In addition to the features required by the core API, GPU vendors may provide additional functionality in the form of *extensions*. Extensions may introduce new functions and new constants, and may relax or remove restrictions on existing OpenGL functions. Vendors can use extensions to expose custom APIs without needing support from other vendors or the Khronos Group as a whole, which greatly increases the flexibility of OpenGL. All extensions are collected in, and defined by, the OpenGL Registry.



1.4 OpenGL Pipeline

**1.5 Introduction to GLUT**

GLUT is the OpenGL utility toolkit, a window system independent toolkit for writing OpenGL programs. It implements a simple windowing API for OpenGL. GLUT makes it easier to learn about and explore OpenGL programming. GLUT provides a portable API so you can write a single OpenGL program that works across all PC and workstation OS platforms. GLUT is designed for constructing small to medium sized OpenGL programs.

While GLUT is well-suited to learning OpenGL and developing simple OpenGL applications, GLUT is not a full-featured toolkit so large applications requiring sophisticated user interfaces are better off using native window system toolkits. The GLUT library has both C, C++ (same as C), FORTRAN, and ADA programming bindings. The GLUT source code distribution is portable to nearly all OpenGL implementations and platforms.

GL

GLUT

GLX

Xlib, Xtk

Frame

buffer

OpenGL

Application

program

GLU

1.5 library organization of OpenGL

**1.6 Applications of OpenGL**

* **OpenGL** (**Open G**raphics **L**ibrary)is a cross-language, multi-platform API for rendering 2D and 3D computer graphics.
* The API is typically used to interact with a GPU, to achieve hardware-accelerated rendering.
* It is widely used in CAD, virtual reality, scientific visualization, information visualization, flight simulation, and video games.

**1.7 OpenGL primitives**

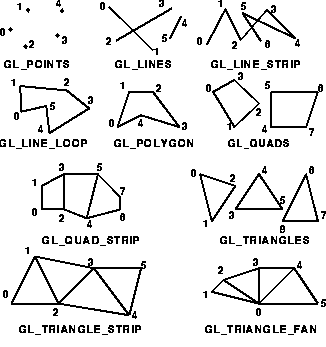
OpenGL supports two classes of primitives:

* Geometric Primitives
* Image(Raster) Primitives

Geometric primitives are specified in the problem domain and include points, line segments, polygons, curves and surfaces.

Raster primitives, such as arrays of pixels pass through a separate parallel pipeline on their way to the frame buffer.

There are ten basic OpenGL primitives:

****

1.7 OpenGL Primitives