BY: Mukunda Paudel
paudelmuku@gmail.com

Chapter - 8 Graphical Language

Vector and raster displays are two substantially different hardware technologies for creating images on the screen. Raster display now dominant hardware technology because they support several modern applications like

- ✓ *fill area with uniform colors repeated patterns* in two or more colors vector can only simulate filled areas with closely spaced sequences of parallel vectors
- ✓ raster display *stores images in pixel forms* and can read or write moved copies or any manipulations

❖ Primary goal of standardized graphics software is ***portability***

Why machine independent graphics language?

- ❖ When packages are designed with standard graphics functions software can be moved easily from one hardware to another without standards, program designed for one hardware often can't be transferred to another system without extensive rewriting of the programs.
- ❖ In international and national standards organizations in many countries have cooperated in an effort to develop a generally accepted standards for computer graphics
- ❖ Graphical Kernel System(GKS) was adopted as the first graphics software standard by the international standard organization and others like ANSI
- ❖ Original GKS was 2D and later on 3D extension was developed
- ❖ Second standard "Programmers Hierarchical interactive graphics standard (PHIGS) is an extension of GKS with increased capabilities for object modeling, color specifications surface rendering and picture manipulations.
- ❖ An extension of PHIGS called PHIGS+ was developed to provide 3D surface shading capabilities standard graphics functions are defined as a set of specifications that is independent of any programming language

Language Binding

Gives the syntax for accessing the various standard graphics functions from this language e.g. PRIGS, GKS function for specifying a sequence of n-1 connected 2D straight line segment polyline (n,x,y)

- ❖ FORTRAN polyline can be drawn with GPLCALLGPR(n, x, y) where x, y are end points

- C gpolyline(n, pts)where pts is list of end points

Graphics Software

- Graphics software refers to a program or collection of programs that enable a person to manipulate images or models visually on a computer.
- There are two general classifications for graphical Software.

General Programming Packages:

- General programming packages provides an extensive set of graphics functions that can be used in a high-level programming language, such as C or FORTRAN. E.g. GL (Graphics Library) system on Silicon Graphics equipment, which includes basic functions for generating picture components (straight lines, polygons, circles, and other figures), setting colors and intensity values, selecting views, and applying transformations

Special Purpose Application Packages:

- Special purpose applications packages are, in contrast designed for non-programmers, so that users can generate displays without worrying about how graphics operations work.
- The interface to the graphics routines in such packages allows users to communicate with the programs in their own terms. For example: the artist's painting programs and various business, medical, and Computer-Aided Design (CAD) systems.

There are many graphics software available in market they can be categories as:

- ✓ **Paint program:** Paint program works with bit map images.

- ✓ **Photo manipulation program:** It works with bit map images and is widely used to edit digitized photographs.

- ✓ **Computer Aided Design program:** CAD software is used in technical design fields to create models of objects that will be built or manufactured. CAD software allows user to design objects in 3 dimensions and can produce 3-D frames and solid models.

- ✓ **3-D Modelling Programs:** It is used to create visual effects. 3-D modeling program works by creating objects like surface, solid, polygon etc.

- ✓ **Animation:** Computer are used to create animation for use in various fields, including games, and movies composing to allow game makers and film makers to add characters and objects to scenes that did not originally contain them.

Software standards

The primary goal of standardized graphics software is portability. When packages are designed with standard graphics functions, software can be moved easily from one hardware system to another and used in different implementations and applications. Without standards, programs designed for one hardware system often cannot be transferred to another system without extensive rewriting of the programs.

1. General Kernel System (GKS)

- ❖ The Graphical Kernel System (GKS) was the first ISO standard for low-level computer graphics
- ❖ Introduced in 1977. The main purpose of GKS is the production and manipulation of 2D pictures in a way that does not depend on the system of graphical device used.
- ❖ In GKS, pictures are constructed from a number of basic building blocks. These basic building blocks are called as primitives.

There are five types of primitives in GKS.

- Polyline – draws sequence of connected lines
- Polymaker – marks a sequence of points with same symbol
- Fill Area – displays a specified area
- Text – draws a string of characters
- Cell Array – displays an image composed of a variety of colors or gray scales.

2. PHIGS (Programmer's Hierarchical Interactive Graphics System)

- ❖ Programmer's Hierarchical Interactive Graphics System
- ❖ Basically a library of about 400 functions that allow the user to display and interact with 2-D and 3-D graphics.
- ❖ It is an international standard, being created by the International Organization for Standardization (ISO).
- ❖ PHIGS hides hardware-dependent details from the user; so, for example, it allows an application draw on a plotter the same way it draws on a computer screen.
- ❖ PHIGS provides a set of familiar graphics objects called *primitives*, each with attributes that control its location, orientation, color, and appearance.

PHIGS Primitives

- **Polyline:** which draws a sequence of connected line segments;
- **polymarker:** which marks a sequence of points with a symbol;
- **fill area:** which defines the boundary of an area to be displayed;
- **fill area set:** which defines the boundaries of a set of areas to be displayed as one;
- **Text:** which draws a sequence of characters;
- **Annotation text:** which draws a sequence of characters to annotate a drawing;
- **Cell array:** which displays an image;

Graphical File Formats

1. JPEG

- ❖ Joint Photographic Experts Group) is a lossy compression method
- ❖ JPEG-compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format.
- ❖ The JPEG/JFIF filename extension is JPG or JPEG.
- ❖ Nearly every digital camera can save images in the JPEG/JFIF format, which supports eight-bit grayscale images and 24-bit color images (eight bits each for red, green, and blue).
- ❖ JPEG applies lossy compression to images, which can result in a significant reduction of the file size.

2. TIFF

- The TIFF format was originally developed by the Aldus Corporation, and was intended primarily for use in scanning and desktop publishing.
- TIFF (Tagged Image File Format) format is a flexible format that normally saves eight bits or sixteen bits per color (red, green, blue) for 24-bit and 48-bit totals, respectively
- Usually using either the TIFF or TIF filename extension.
- TIFF image format is not widely supported by web browsers.

3. GIF

- The GIF format was developed in 1987 by CompuServe Incorporated, primarily for use on the Internet.
- The current version, GIF89a, was released in 1990. It supports color depths from 1-bit (monochrome) to 8-bit (256 colors)
- GIF always stores images in compressed form, using lossless LZW compression. Other supported features include interlacing and transparency
- GIF is most suitable for storing graphics with few colors, such as simple diagrams, shapes, logos, and cartoon style images, as it uses LZW lossless compression, which is more effective when large areas have a single color, and less effective for photographic images.

4. BMP

- The BMP format (sometimes referred to as a Device Independent Bitmap) was developed by Microsoft as the native raster format of the Windows operating system.
- The BMP file format (Windows bitmap) handles graphic files within the Microsoft Windows OS.
- Typically, BMP files are uncompressed, and therefore large and lossless; their advantage is their simple structure and wide acceptance in Windows programs.

5. PNG

- The Portable Network Graphics (PNG) format was developed by the PNG Development Group in 1996, to provide an open alternative to GIF and the associated licensing issues with LZW compression
- The PNG (Portable Network Graphics) file format was created as a free, open-source alternative to GIF.

- The PNG file format supports eight-bit palleted images (with optional transparency for all palette colors) and 24-bit true color (16 million colors) or 48-bit true color with and without alpha channel - while GIF supports only 256 colors and a single transparent color.

6. PSD

- The PhotoShop Document (PSD) format was developed by Adobe as the default format for their digital image and design package, PhotoShop.
- This is a proprietary format that supports RLE compression and colour depths up to 32-bit. It is used primarily for saving images that are being edited within PhotoShop, but is not widely used as an interchange format

Data structure in Computer Graphics

1. Triangle mesh:

- **Triangle mesh** is a type of polygon mesh in computer graphics.
- It comprises a set of triangles (typically in three dimensions) that are connected by their common edges or corners.
- The data structure representing the mesh provides support for two basic operations, inserting triangles and removing triangles.

2. Quad-edge:

- **Quad-edge** data structure is a computer representation of the topology of a two-dimensional or three-dimensional map, that is, a graph drawn on a (closed) surface.
- It represents simultaneously both the map, its dual and mirror image

3. Polygon Mesh:

- **Polygon mesh** is a collection of vertices, edges and faces that defines the shape of a polyhedral object in 3D computer graphics and solid modeling.
- The faces usually consist of triangles (triangle mesh), quadrilaterals, or other simple convex polygons, since this simplifies rendering, but may also be composed of more general concave polygons, or polygons with holes.

4. Octree:

- An **octree** is a tree data structure in which each internal node has exactly eight children. Octrees are most often used to partition a three-dimensional space by recursively subdividing it

Introduction to OpenGL

- (Open Graphics Library) is a standard specification defining a cross language cross-platform API for writing applications that produce 2D and 3D computer graphics. The interface consists of over 250 different function calls which can be used to draw complex three-dimensional scenes from simple primitives OpenGL is widely used in CAD, virtual reality, scientific visualization, information visualization, and flight simulation
- OpenGL's basic operation is to accept primitives such as points, lines and polygons, and convert them into pixels At its most basic level OpenGL is a specification, meaning it is simply a document that describes a set of functions and the precise behaviors that they must perform. From this specification, hardware vendors create implementations — libraries of functions created to match the functions stated in the OpenGL specification, making use of hardware acceleration where possible. Hardware vendors have to meet specific tests to be able to qualify their implementation as an OpenGL implementation.
- OpenGL is a low-level, procedural API, requiring the programmer to dictate the exact steps required to render a scene.

OpenGL serves two main purposes:

- To hide the complexities of interfacing with different 3D accelerators, by presenting the programmer with a single, uniform API.
- To hide the differing capabilities of hardware platforms, by requiring that all implementations support the full OpenGL feature set (using software emulation if necessary).

OpenGL is a low-level graphics library specification. It makes available to the programmer a small set of geometric primitives - points, lines, polygons, images, and bitmaps.

- OpenGL provides a set of commands that allow the specification of geometric objects in two or three dimensions, using the provided primitives,

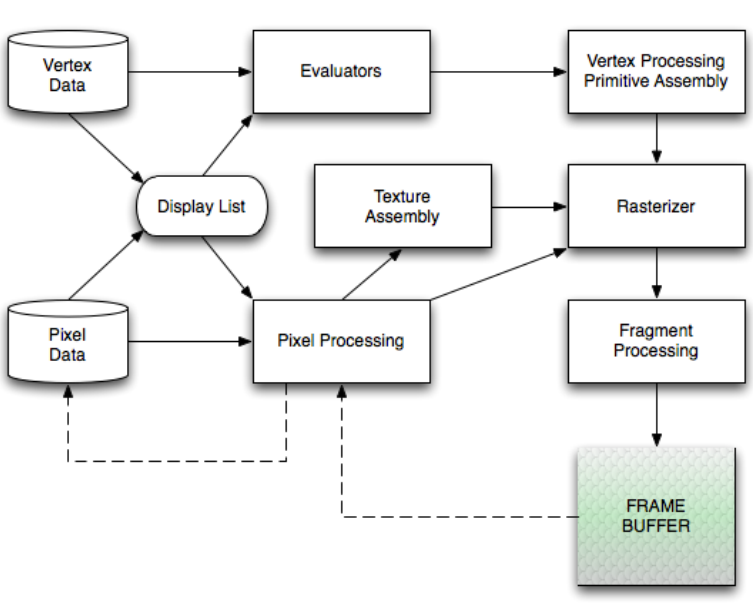
together with commands that control how these objects are rendered (drawn).

- ❖ Since OpenGL drawing commands are limited to those that generate simple geometric primitives (points, lines, and polygons), the OpenGL Utility Toolkit (GLUT) has been created to aid in the development of more complicated three-dimensional objects such as a sphere, a torus, and even a teapot.

- ❖ GLUT may not be satisfactory for full-featured OpenGL applications, but it is a useful starting point for learning OpenGL.

Rendering Pipeline

- Rendering Pipeline is the sequence of steps that OpenGL takes when rendering objects.
- Vertex attribute and other data go through a sequence of steps to generate the final image on the screen.
- There are usually 9-steps in this pipeline most of which are optional and many are programmable.



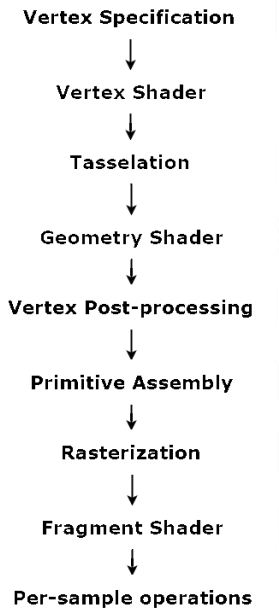


Figure a): OpenGL Rendering Pipeline **b)** Sequential Execution of OpenGL Pipeline

Vertex Specification: List an ordered list of vertices that define the boundaries of the primitive. Along with this, one can define other vertex attributes like color, texture coordinates etc.

Vertex Shader: Vertex Shader is a program that manipulate the vertex data. The ultimate goal of vertex shader is to calculate final vertex position of each vertex. Vertex shaders are executed once for every vertex (e.g. in case of a triangle it will execute 3-times) that the GPU processes.

Tessellation: Optional stage. In this stage primitives are tessellated i.e. divided into smoother mesh of triangles.

Geometry Shader: Optional stage. The work of Geometry Shader is to take an input primitive and generate zero or more output primitive.

Vertex Post Processing: This is a fixed function stage i.e. user has a very limited to no control over these stages. The most important part of this stage is **clipping**. Clipping discards the area of primitives that lie outside the viewing volume.

Primitive Assembly: This stage collects the vertex data into an ordered sequence of simple primitives (lines, points or triangles).

Rasterization: An important step in pipeline. The output of rasterization is a fragments.

Fragment Shader: The fragment shader runs for each fragment in the geometry. The job of the fragment shader is to determine the final color for each fragment.

Per-sample Operations: There are few tests that are performed based on user has activated them or not. Some of these tests for example are Pixel ownership test, Scissor Test, Stencil Test, Depth Test.

Finally, Geometric data (vertices, line, and polygons) follow a path through the row of boxes that includes evaluators and per-vertex operations, while pixel data (pixels, images and bitmaps) are treated differently for part of the process. Both types of data undergo the same final step before the final pixel data is written to the frame buffer

Graphics Libraries

- Graphics lib reduces amount of time to repeat a job over n over. A program written on an IBM machine cannot run on an apple machine w/o major revisions of the program. The growing usage of computer graphics on different platforms raises the need for standardization. With the standards, no major revisions of computer programs supported across different platform is necessary.
- The Graphical Kernel System (GKS) was the earliest attempt as such standardization. A later attempt came in the form of The Programmers Hierarchical Interactive Graphic Standard. (PHIGS). This have since been adopted by the American Standards Institute (ANSI) and the International Standard Organizations (ISO).
- Various other attempts such as ISO include: GL, Open GL etc. All this standards define their own machine independent ways of assembly and storage of their data. A labeled set of output primitives in PHIGS is called a structure while in GKS is called a segment and in GL it's called an object.

OpenGL utility libraries

- OpenGL provides a powerful but primitive set of rendering command, and all higher-level drawing must be done in terms of these commands. There are several libraries that allow you to simplify your programming tasks, including the following:

  - ✓ **OpenGL Utility Library (GLU)** contains several routines that use lower-level OpenGL commands to perform such tasks as setting up matrices for specific viewing orientations and projections and rendering surfaces.
  - ✓ **OpenGL Utility Toolkit (GLUT)** is a window-system-independent toolkit, written by Mark Kilgard, to hide the complexities of differing window APIs.
  - ✓ **Simple DirectMedia Layer (SDL)**
  - ✓ **OpenGL User Interface Library (GLUI)**
  - ✓ **OpenGL Extension Wrangler Library (GLEW)**
  - ✓ **OpenGL Easy Extension library (GLEE)**
  - ✓ **Fast, Light Toolkit" (FLTK)**

End of Chapter