Computer Graphics: Unit 8: Graphical Languages:

8.1: Need for Machine Independent Graphical languages:

→ A general purpose graphics package provides users with a variety of functions for creating and manipulating pictures.

→ Provides extensive set of graphics functions that can be used in high level programming language such as C, or FORTRAN.

→ Basic function include generating picture components, setting color and intensity values, selecting views and applying transformations.

→ User can generate displays without worrying about how graphics operations work.

→ It provides portability feature.

→ Other tasks includes geometric transformation, viewing information, subdivision of components parts, etc.

→ General programming packages + Special purpose Application Packages.

Overview of Graphics file format:

→ Graphics image are stored digitally using a small number of standardized graphic file formats. Lets discuss one by one.

1) JPEG: Joint Photographic Expert Group is a lossy compression method. Its file name extension is JPEG or JPG. Every digital camera can save images in the JPEG/JIF format, which supports eight-bit grayscale image & 24-bit color images (8 bits for each R, G, B).

2) TIFF: Tagged Image file format.

→ Flexible format, normally save 8 bits or 16 bit color.

→ Extension = TIFF or TIF

→ Not widely supported by web browsers.

3. GIF: Graphics Interchange format.
→ Limited to 8-bit palette. 256 colors.
→ Suitable for storing graphics with few colors such as diagrams, shape, logos etc.
→ More effective if large areas have single color.
→ Less effective for photographic images.

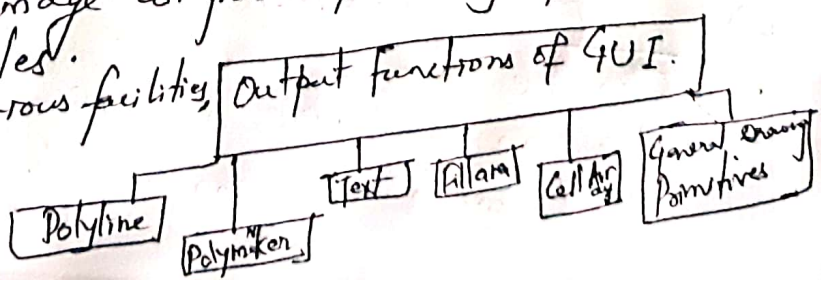
4). BMP: Handles graphic files within the Microsoft Windows OS.
→ typically uncompressed, therefore large and lossless.
→ wide acceptance in Windows programs.

5). PNG: (Portable Network Graphics)
→ created as a free, opensource alternative to GIF.
→ Support 8-bit palletted images and 24-bit true color (16 million colors) or 48-bit truecolor with and without alpha-channel.

---

* Explain GKS and PHIGS → Graphical Languages.

1) GKS: Graphical Kernel System (1977)
→ First ISO standard for low-level computer graphics.
→ Used for 2-dimensional graphics.
→ Production & manipulation of 2-D pictures.
→ In GKS, pictures are constructed from a number of basic building blocks. These building blocks are called as primitive.
→ 6 types of GKS primitives are
(A) Polyline: ability to draw one or more st. lines.
(B) Polymarker: draw a symbol
(C) Text: Add text to given coordinates / string of characters.
(D) Fill Area: display specified area, hollow, solid etc.
(E) Cell array: display image composed of variety of colors or gray scales.
(F) General Drawing Primitives: various facility, drawing softwares.



② PHIGS: Programmer's Hierarchical Interactive Graphics System.

→ It is basically a library of about 400 functions that allow the user to display and interact with 2D and 3D graphics.
→ It is flexible, device independent standard that should prove useful in development of CAD applications.
→ Hides hardware dependent details from the user.
→ It provides set of familiar graphics object called primitives.

PHIGS Primitives:

(A) Polyline: draw a sequence of connected line segments.
(B) Polymarker: marks a sequence of point with a symbol.
(C) Fill Area: defines boundary of an area to be displayed.
(D) Fill Area Set: defines the boundaries of a set of areas to be displayed as one.
(E) Text: sequence of characters
(F) Annotation Text: draws a sequence of characters to annotate a drawing.
(G) Cell Array: which displays an image.

---

4 Graphical Software Standard:

① General purpose Packages: provide extensive set of graphic function. Can be used in high level languages.
Eg C or FORTRAN.
Eg: GL (Graphics Library system on Silicon Graphics equipment)
- Contains all functions of Lines, polygons, circles etc.

② Special Purpose Application Packages: Designed for non-programmers, generate displays without knowing much about graphics operations. Eg: CAD, artists painting program etc.
→ Many graphics software available such as
① Paint Program
② Photo Manipulation Program
③ CAD
④ 3-D Modelling Programs
⑤ Animation.

DATA STRUCTURE IN COMPUTER GRAPHICS:

① Triangle Mesh: is a type of polygon mesh in computer graphics.
→ Comprises a set of triangles (3 Dimensions) that are connected by their edges or corners.
→ Inserting and removing triangles facilities.

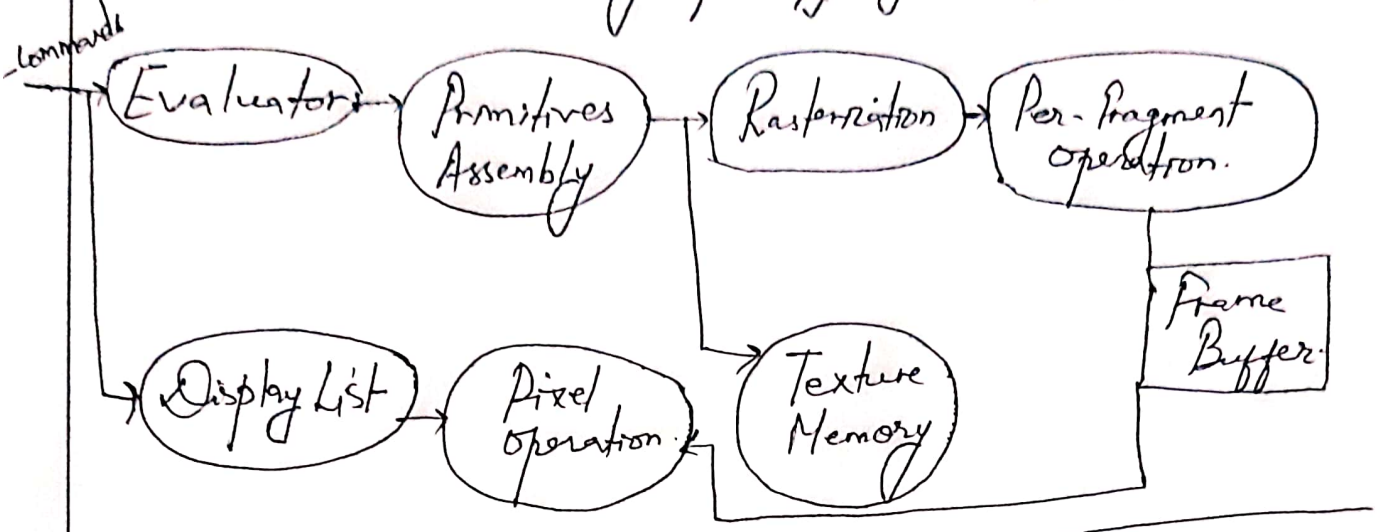
② Quad Edge: is a computer representation of the topology of 2D or 3D map.
→ represents both map - i.e, dual and mirror image.

③ Polygon Mesh: is a collection of vertices, edges and faces that defines the shape of polyhedral object in 3D computer graphics and solid modelling.
→ faces generally consist of triangles, quadrilaterals or other simple convex polygons.

④ Octree: is a tree data structure in which internal node has exactly eight children.
→ mostly used to partitioned a three dimensional space by recursively subdividing it.

OpenGL: Low level graphics library specification.
→ Programmer gets familiar with geometric primitives like points, lines, polygons, images and bitmaps.
→ provides a set of commands that allow the specification of geometric objects in 2 or 3 dimension.
→ OpenGL Utility Toolkit (GLUT) has been also created to aid in the development of more complicated 3D objects as sphere, etc.
→ There are 250 different function calls which can be used to draw complex 2 & 3 D scenes.

→ Open Graphics Library.
→ Cross Language - Can be implemented in any languages
→ Cross platform - use any operating systems, iOS, android etc.



→ Basic Primitives
GL - Points - helps to plot point
GL - Lines - " " " Line
GL - Line Strip :
GL - Line-loop :
GL - triangle ~~strip~~ →
GL - triangle - strip →
GL - triangle - Fan →
GL - Quads →
GL - Quad Strip →
GL - Polygon →

Examples:
① Foremost
② Simple GLUT
③ Frame Buffer Object.
④ 2D Heat Simulation
⑤ Vertex Fragment Shadder.
etc

→ OpenGL provides powerful but primitive set of rendering command. There are several libraries to simplify our programming tasks.
① OpenGL Utility Library - uses low level openGL.
② OpenGL Utility Toolkit (GLUT) - independent toolkit, hide complexities.