Chapter 6
Graphics Standards

Credit hours: 4 hrs

Contents

6.1 Need for Machine Independent Graphical Languages

6.2 Graphical Standards: PHIGS, GKS

6.3 Graphics Software Standards and Language Binding

6.4 Overview of Graphics File formats

6.5 Visualization of Data Sets

6.1 Need for Machine Independent Graphical Languages

- Machine languages are the first generation of programming languages.
- A machine language is a machine-dependent, low-level language that uses binary code to interact with a specific computer system.
- A machine-dependent language works only on a specific computer system and its components.
- A Machine Independent language is one that can run on any machine. An example of this would be Java.
- Machine Independent language can take the compiled code for any given machine and run it on the machine we are attempting to run it on.
- A TRULY machine-independent language would produce exactly the same output no matter which computer it was run on.

- In computer graphics, a machine-independent language refers to a programming language or specification that is not tied to the specific hardware architecture of a computer.
- Examples include OpenGL and Vulkan. These machine-independent languages abstract hardware specifics, promoting flexibility and widespread use.
- The need for a machine-independent language in computer graphics arises from several important considerations:
  1. Portability
  2. Maintenance
  3. Compatibility
  4. Interoperability
  5. Development Efficiency
  6. Distribution
  7. Standardization

- A machine-independent language in computer graphics is essential for:
  1. **Portability:**
     Enables graphics programs to run on diverse hardware without modification.
  2. **Maintenance:**
     Simplifies updates and bug fixes, as code remains consistent across platforms.
  3. **Compatibility:**
     Ensures graphics programs work seamlessly with evolving hardware architectures.
  4. **Interoperability:**
     Facilitates interaction with other software components or systems.
  5. **Development Efficiency:**
     Allows focus on logical aspects, improving development efficiency.
  6. **Distribution:**
     Enables platform-independent distribution of graphics applications.
  7. **Standardization:**
     Contributes to establishing common practices and standards in graphics development.

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 Softwares.
  a) General Programming Packages:
  b) Special Purpose Application Packages:

a) 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

b) 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.
- **3-D Modeling 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

6.2 Graphical Languages

- Graphical languages in the context of computer graphics refer to programming languages or systems specifically designed for creating and manipulating graphical elements, such as images, animations, and user interfaces.
- These languages provide a set of tools and syntax that make it easier for developers to express graphical concepts and operations.
- Examples include:
  - OpenGL for general graphics,
  - GLSL for shaders,
  - HTML/XML for web interfaces, and
  - GUI description languages like Qt Designer.
- These languages simplify the development of graphical applications, from games to user interfaces, by providing specific tools and syntax tailored to visual content.

6.3 Graphics Software Standard

- 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.
- **a) General Kernel System (GKS)**
- **b) PHIGS (Programmer's Hierarchical Interactive Graphics System)**

a) 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.
- There are five types of primitives in GKS.
  - **Poly line –** draws sequence of connected lines
  - **Poly maker –** 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.

b. PHIGS (Programmer’s Hierarchical Interactive Graphics System)

- PHIGS, short for the Programmer's Hierarchical Interactive Graphics System, is 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

6.4 Overview of Graphics File formats

Different graphics file formats are available:

1. JPEG (Joint Photographic Experts Group)
2. TIFF (Tagged Image File Format)
3. GIF (Graphics Interchange Format)
4. BMP (Bitmap))
5. PNG (Portable Network Graphics)

1. JPEG

- Joint Photographic Experts Group, JPEG, 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 (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

- GIF (Graphics Interchange Format) is in normal use limited to an 8-bit palette, or 256 colors (while 24-bit color depth is technically possible).
- 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

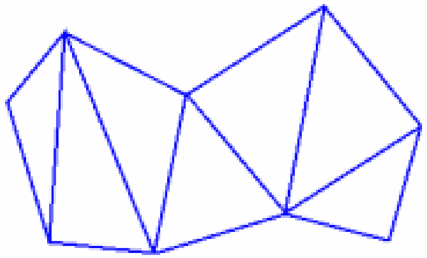
- The BMP file format (Windows bitmap) handles graphic files within the Microsoft Windows OS.
- Typically, BMP files are uncompressed, and therefore large and lossless; their advantage is their simple structure and wide acceptance in Windows programs.

5. PNG

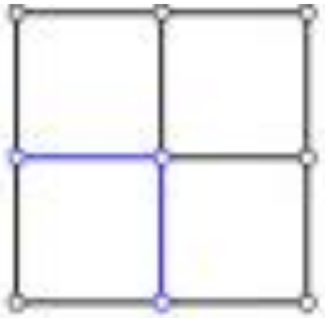
- The PNG (Portable Network Graphics) file format was created as a free, open-source alternative to GIF.
- The PNG file format supports eight-bit paletted images (with optional transparency for all palette colors) and 24-bit truecolor (16 million colors) or 48-bit truecolor with and without alpha channel - while GIF supports only 256 colors and a single transparent color.

6.5 Data structure in computer graphics

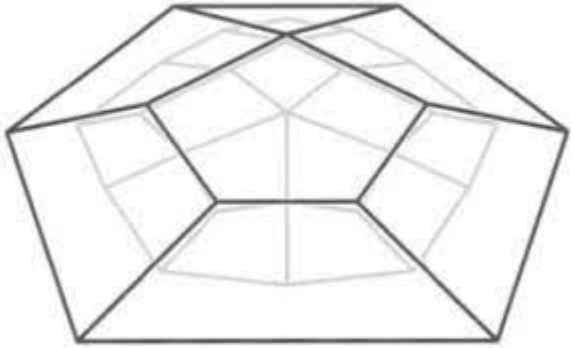
A) Triangle mesh: A **triangle mesh** is a type of polygon mesh in computer graphics. It comprises a set of triangles (typically in three dimensions) that are connected by their common edges or corners. The data structure representing the mesh provides support for two basic operations, inserting triangles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



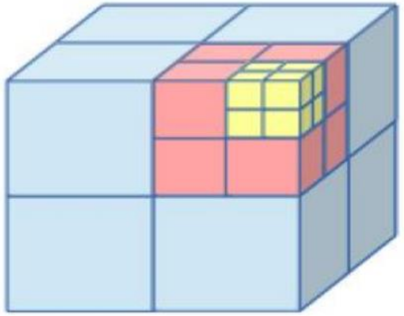
B) Quad-edge: A **quad-edge** data structure is a computer representation of the topology of a two-dimensional or three-dimensional map, that is, a graph drawn on a (closed) surface. It represents simultaneously both the map, its dual and mirror image

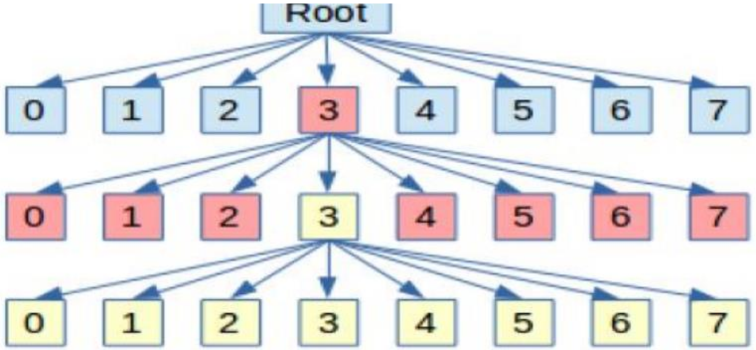


C) Polygon Mesh: A **polygon mesh** is a collection of vertices, edges and faces that defines the shape of a polyhedral object in 3D computer graphicsand solid modeling. The faces usually consist of triangles (triangle mesh), quadrilaterals, or other simple convex polygons, since this simplifies rendering,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



D) 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





6.6 Visualization of Data sets

- **Data visualization** is actually a set of data points and information that are represented graphically to make it easy and quick for user to understand.
- Data visualization is good if it has a clear meaning, purpose, and is very easy to interpret, without requiring context.
- Tools of data visualization provide an accessible way to see and understand trends, outliers, and patterns in data by using visual effects or elements such as a chart, graphs, and maps
- There are various ways to visually represent datasets, depending on the nature of the data and the insights you want to convey.
- Here are some common ways:

- **Line Chart:**
  - Used to display data points over a continuous interval or time.
  - Effective for showing trends and patterns.
- **Bar Chart:**
  - Ideal for comparing values of different categories.
  - Can be used as a horizontal or vertical bar graph.
- **Histogram:**
  - Shows the distribution of a single variable.
  - Divides data into bins and represents frequencies.
- **Scatter Plot:**
  - Displays individual data points on a two-dimensional graph.
  - Useful for visualizing relationships between two variables.

- **Pie Chart:**
  - Represents parts of a whole.
  - Suitable for displaying the proportion of different categories.
- **Area Chart:**
  - Similar to a line chart but with the area beneath the line filled.
  - Useful for showing cumulative data.
- **Bubble Chart:**
  - Enhances a scatter plot by adding a third dimension (size of the bubble).
  - Useful for displaying three-dimensional data.
- **Box-and-Whisker Plot (Box Plot):**
  - Visualizes the distribution of a dataset.
  - Shows median, quartiles, and potential outliers.

- **Treemap:**
  - Represents hierarchical data using nested rectangles.
  - Each rectangle's size represents a quantitative value.
- **Network Graph:**
  - Visualizes relationships between entities (nodes) and connections (edges).
  - Useful for displaying complex relational data.
- **3D Plot:**
  - Represents three-dimensional data.
  - Can be used in scatter plots, surface plots, or other chart types.

Exam Questions:

1. Why is it required to choose machine independent graphical languages while working on a project? Explain various phases of project development. [2011 spring]
2. Explain any two graphical languages you are familiar with highlighting their significance. [2011 spring]
3. What does a project mean? Consider a project of your kind, what sort of plans would you make (better be specific on your product) to develop the product that ensure all your necessities? [2012 fall]
4. Explain the things to be considered while developing a project. [2012 spring]
5. What is the significance of making plans for a project? What things should be considered during the project development? [2013 fall /spring]
6. Why we need machine independent graphical language? Explain briefly about any two of the graphical file formats. [2011 fall, 2013 fall]
7. Explain different file formats. [2014 fall]
8. Explain the APIs used in OpenGL for rendering graphical objects. [2015 spring]
9. Why is OpenGL considered to be cross language and cross platform collection of application programming interfaces for rendering objects? Explain any four OpenGL APIs that you are familiar with. [2016 fall]
10. Differentiate between RGB and CMYK color models. Explain any two graphical file formats. [2016 spring]
11. What is meant by a project? Describe the significance of making plans for project development with appropriate illustrations. [2011 fall]
12. Explain Graphics File format. Explain with example, the need for machine independent graphical language[2017 fall]
13. Why machine independent programming language is used? Discuss about OPENGL. [2017 spring]

14. Explain the need for machine independent graphical languages, and also explain about GKS. [2018 fall]
15. Explain OPEN GL. [2018 spring]
16. Define OpenGL. Explain the different file format used in Graphics to save image. [2019 fall]
17. Explain about PHIGS and GKS languages. [2019 spring]
18. Explain about PHIGS and GKS .Also list out the available graphical file format. [2020 fall]
19. Explain how machine independent graphical language are more preferable to develop graphical project. [2021 fall]
20. Why do we need the machine independent graphical languages? List out some of the graphical file formats and explain them in short. [2021 spring]
21. What is OpenGL? Why GLUT is implemented in OpenGL. Explain Callback function. [2022 fall]
22. Write short notes on:
    - a) Graphical file formats [2012 fall]
    - b) Explain different file formats [2012 spring]
    - c) Project development [2014 fall]
    - d) OpenGL [2015 spring, 2021 fall]
    - e) PHIGS [2011 fall]
    - f) Need of machine independent graphical languages [2019 fall]