**What is Computer Graphics:** Computer Graphics art of drawing pictures animation and making visual effects by the use of computers with the help of some algorithms , mathematical models and some hardware devices .

It is the use of computers to create and manipulate pictures on a display device. It comprises of software techniques to create, store, modify, represents pictures.

Today computer graphics is entirely different from the earlier one. It is not possible. It is an interactive user can control the structure of an object by the use of various input devices.

**Applications of Computer Graphics:**

1. **Use in Biology:** Molecular biologist can display a picture of molecules and gain insight into their structure with the help of computer graphics.
2. **Computer-Generated Maps:** We can make a computer generated maps by the use of computer graphics which can be used for town planners and transportation engineers and they can use the usefull data for planning purposes.
3. **Graphical Presentation:** We can display the information in graphical form for example bar charts , line graphs and pie charts. The information which is displayed in graphical can be be understood very easily . it can be used in the fields like Financial Reports

- Statistical Reports
- Mathematical Reports
- Scientific Reports

- Economic Data for research reports
- Managerial Reports

4. **Entertainment:** Computer Graphics are now commonly used in making motion pictures, music videos and television shows.
5. **Educational Software:** Computer Graphics is used in the development of educational software for making computer-aided instruction.
6. **Printing Technology:** Computer Graphics is used for printing technology and textile design.

**Raster Scan Displays:** Are the commonly used monitors which uses CRT stand for cathode ray tube. In the Raster scan definition of the picture are stored in the memory which is known as refresh buffer or frame buffer . Electron beam will move the whole screen one row at a time and will turn beam of light on and off based on the picture definition. By doing the picture pattern is created.

Refresh rate is the frequency at which screen updates its picture. The refresh rate of a raster scan is usually 60-80 frames per second.

**Raster-Scan Display Processor:** Its main aim to digitalized a picture by converting the intensity values in into pixels values.

It can also display color areas , creating different line style etc

**ADVANTAGES:**

- Real life images with different shades can be displayed.
- Color range available is bigger than random scan display.

**DISADVANTAGES:**

- Resolution is lower than random scan display.
- More memory is required.
- Data about the intensities of all pixel has to be stored.

**Random Scan:** Also know as vector display. in this electron beam are directed only that part of the area where image is actually drawn.

It draws picture one line at a time.

Pen plot is and example of random scan .

The area where the definition of picture is stored is know as refresh display file , it sotres picture definition as a set of line drawing commands.

The commands will be:

- Draw a line from A to B
- Draw a line from B to C
- Draw a line from C to D
- Draw a line from D to A

**ADVANTAGES:**

- Higher resolution as compared to raster scan display.

- Produces smooth line drawing.
- Less Memory required.

**DISADVANTAGES:**

- Realistic images with different shades cannot be drawn.
- Colour limitations.

Look up tables : In computer graphics, lookup tables (LUTs) are essentially pre-calculated tables used to accelerate computations. Here's a simple breakdown:

- In graphics, one common type of lookup table is used for colors. Imagine you have an image with millions of pixels, each represented by a combination of red, green, and blue (RGB) values. Instead of recalculating the exact color of each pixel every time you display the image, you can use a lookup table.
- The lookup table, in this case, would contain pre-defined RGB values. So, instead of storing the RGB values directly in the image data, you store an index number for each pixel that corresponds to a specific color in the lookup table.
- When the image is displayed, the system looks up each index in the table and retrieves the corresponding color. This process significantly speeds up the rendering of the image, especially for complex graphics.

2. **Transformation Lookup Tables**:

- Lookup tables can also be used for various transformations, such as converting between color spaces, applying filters, or performing mathematical operations.

- For instance, suppose you're applying a complex mathematical function (like a logarithm or a sine wave) to every pixel in an image. Calculating these functions individually for each pixel can be time-consuming. Instead, you can pre-calculate the function values for a range of input values and store them in a lookup table.

- Then, during image processing, rather than computing the function for each pixel individually, you simply look up the pre-calculated value from the table based on the input value. This greatly speeds up the transformation process.

OpenGL (Open Graphics Library) is a cross-platform API (Application Programming Interface) for rendering 2D and 3D vector graphics. It provides a set of functions for performing various graphics operations, such as drawing geometric shapes, applying textures, and manipulating matrices for transformations. Here's a breakdown of its key concepts:

1. **Cross-Platform**: OpenGL is designed to work across different operating systems (Windows, macOS, Linux, etc.) and hardware

configurations. This makes it a popular choice for developing graphics applications that need to run on multiple platforms.

2. **Rendering Pipeline**: OpenGL follows a graphics pipeline model, where geometry and other data are processed through various stages to produce the final image. These stages typically include vertex processing, primitive assembly, rasterization, fragment processing, and framebuffer operations.

3. **Rendering Primitives**: OpenGL supports different types of primitives for rendering, such as points, lines, and polygons. These primitives can be used to construct complex shapes and scenes.

4. **Transformation and Viewing**: OpenGL provides functions for defining transformations (translation, rotation, scaling) and setting up a viewing frustum for perspective or orthographic projection. This allows developers to position and orient objects in 3D space.

5. **Shading and Lighting**: OpenGL supports shading languages (like GLSL - OpenGL Shading Language) for defining custom vertex and fragment shaders. Shaders enable developers to implement advanced lighting effects, texture mapping, and other visual enhancements.

6. **Texture Mapping**: Textures can be applied to surfaces in OpenGL to add detail and realism to objects. The API provides functions for loading, configuring, and mapping textures onto geometry.

7. **Buffers and Memory Management**: OpenGL utilizes buffers to store data such as vertices, textures, and framebuffers. These buffers can be manipulated efficiently using OpenGL functions to improve rendering performance.

8. **State Management**: OpenGL maintains a set of global state variables that control various rendering behaviors. These include settings for blending, depth testing, stencil operations, and more.

9.Integration with Windowing Systems**: OpenGL can be integrated with windowing systems (like GLFW, SDL, or native windowing APIs) to create graphical user interfaces and handle input events.

Double buffering is a technique used in computer graphics to improve the smoothness and efficiency of rendering animations and other real-time graphics. It involves using two separate buffers to store image data: one buffer for rendering and another for display. Here's how it works:

1. **Rendering Buffer**: This buffer is used by the graphics processing unit (GPU) to draw the next frame of the animation or scene. The rendering buffer is typically not directly visible to the user and can be modified by the GPU without affecting what is currently being displayed on the screen.

2. **Display Buffer**: This buffer holds the image that is currently being displayed on the screen. It is visible to the user and represents the latest complete frame of the animation or scene.

The key concept behind double buffering is that while one buffer is being displayed on the screen, the other buffer is being rendered by the GPU. Once the rendering of the next frame is complete, the roles of the buffers are swapped:

- The contents of the rendering buffer are copied or swapped into the display buffer.
- The rendering buffer is then cleared or prepared for the next frame to be rendered.

This swap typically occurs during the vertical blanking interval, a brief period of time when the monitor is not actively refreshing the screen. By performing this swap during the vertical blanking interval, the transition between frames appears seamless to the viewer, eliminating artifacts such as tearing.

Benefits of double buffering include:

1. **Smooth Animation**: By separating rendering and display, double buffering helps prevent flickering and tearing that can occur when updating the screen mid-frame.

2. **Optimized Performance**: While one buffer is being displayed, the GPU can work on rendering the next frame concurrently, improving overall performance and reducing latency.

3. **Synchronization**: Double buffering provides a synchronized mechanism for updating the display, ensuring that changes to the image are applied at the appropriate time to avoid visual artifacts.

Double buffering is a fundamental technique used in real-time graphics applications such as video games, multimedia applications, and graphical user interfaces (GUIs) to provide a smoother and more responsive user experience.