EXPERIMENT NUMBER 1(A)

FAMILIARIZATION OF COMPUTER GRAPHICS HARDWARE AND SOFTWARE

AIM

To familiarize with various computer graphics hardware and software

INTRODUCTION

Computer Graphics is the creation of pictures with the help of a computer. The end product of the computer graphics is a picture it may be a business graph, drawing, and engineering.

The role of computer graphics insensible. In today life, computer graphics has now become a common element in user interfaces, T.V. commercial motion pictures

In computer graphics, two or three-dimensional pictures can be created that are used for research. Many hardware devices algorithm has been developing for improving the speed of picture generation with the passes of time. It includes the creation storage of models and image of objects. These models for various fields like engineering, mathematical and so on.

APPLICATIONS

Computer Graphics has numerous applications, some of which are

- GUIs (Graphical User Interfaces)
- Weather Maps
- Satellite Imaging
- Medical Imaging
- Photo Enhancement

Cathode Ray Tube (CRT)

CRT, short for Cathode Ray Tube, is a technology used in traditional computer monitors and televisions. The image on these displays are created by firing electrons from the back of the tube of phosphorous located towards the front of the screen.

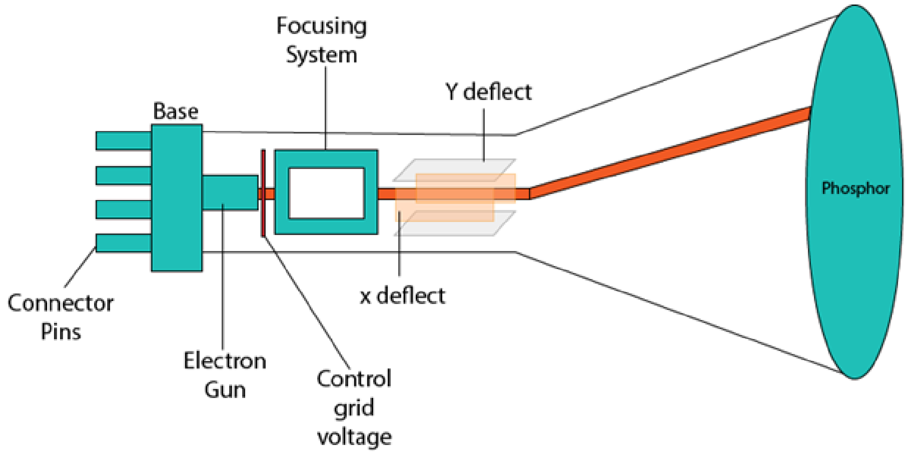


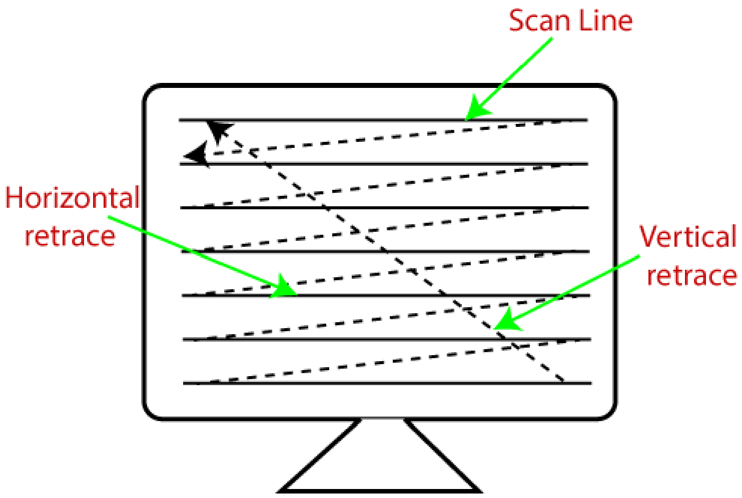
Image of Cathode Ray Tube

Once the electron heats the phosphorus, they light up, and they are projected on a screen. The colour you view on the screen is produced by a **blend of red, blue and green** (RGB) light.

Raster Scan Display

It is based on intensity control of pixels in the form of a rectangular box called Raster on the screen. Information of on and off pixels is stored in refresh buffer or **Frame buffer**.

Televisions in our house are based on Raster Scan Method. The raster scan system can store information of each pixel position, so it is suitable for realistic display of objects. Raster Scan provides a refresh rate of 60 to 80 frames per second.



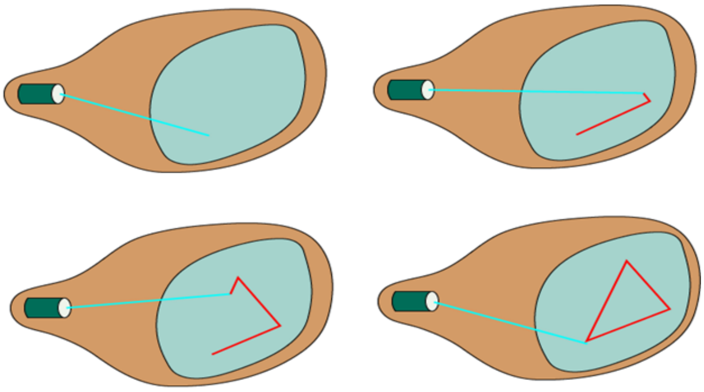
Frame Buffer (also known as Raster or bit map), the positions are called picture elements or pixels. There are Beam refreshing is of two types

1. <u>Horizontal retracing:</u> The return to the left of the screen, after refreshing each scan line.
2. <u>Vertical retracing:</u> When the beam starts from the top left corner and reaches the bottom right scale, it will again return to the top left side.

Advantages	Disadvantages
Realistic Images	Low Resolution
Shadow Scenes are possible	Expensive
Million different colours can be generated	

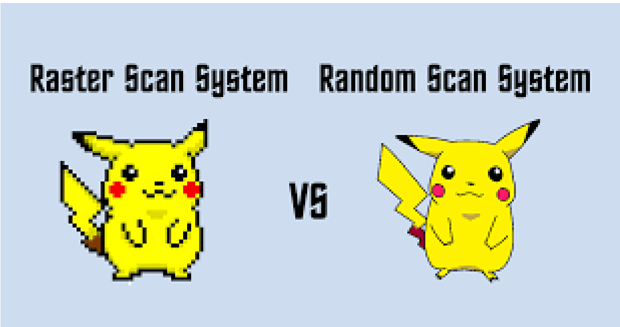
<u>Random Scan Display</u>

It uses an electron beam which operates like a pencil to create a line image on CRT Screen. The picture is constructed out of sequence of straight-line segment. Each line segment is drawn on the screen by directing the beam to move from one point on the screen to the next, where its x & y coordinates define each point. After drawing the picture. The system cycles back to the first line and design all the lines of the image 30 to 60 time each second.



Advantages	Disadvantages
High Resolution	Cannot display realistic shades scenes
Produces smooth line drawings	Expensive

Difference Between Raster Scan Display and Random Scan Display



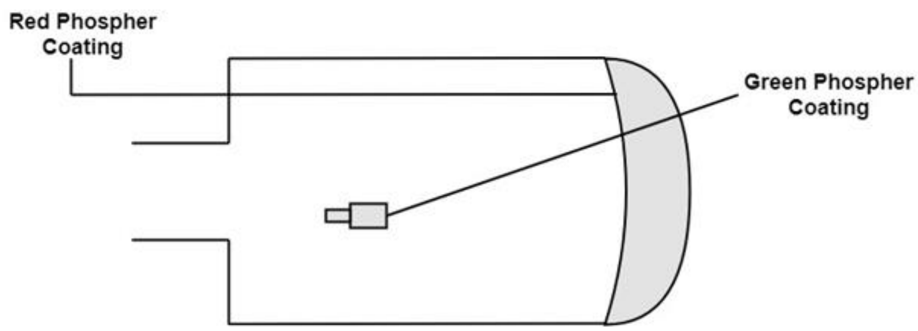
Random Scan Display

The CRT Monitor display by using a combination of phosphors. The phosphors are different colours.

There are two popular approaches for producing colour displays with a CRT are:

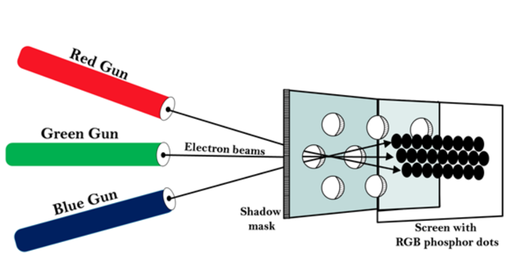
1. Beam Penetration Method: The Beam-Penetration method has been used with random-scan monitors. In this method, the CRT screen is coated with two layers of phosphor, red and green and the displayed colour depends on how far the electron beam penetrates the phosphor

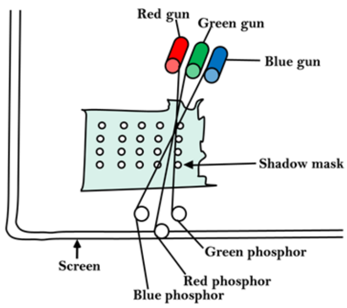
layers. **This method produces four colours only, red, green, orange and yellow.** A beam of slow electrons excites the outer red layer only; hence screen shows red colour only. A beam of high-speed electrons excites the inner green layer. Thus, screen shows a green colour



Advantages	Disadvantages
Inexpensive	Only 4 colours are possible
	Quality of pictures are not as good compared to another method

2. <u>Shadow-Mask Method:</u> Commonly used in raster-scan system because they produce a much wider range of colours than the beam-penetration method. Used in the majority of colour TV sets and monitors.





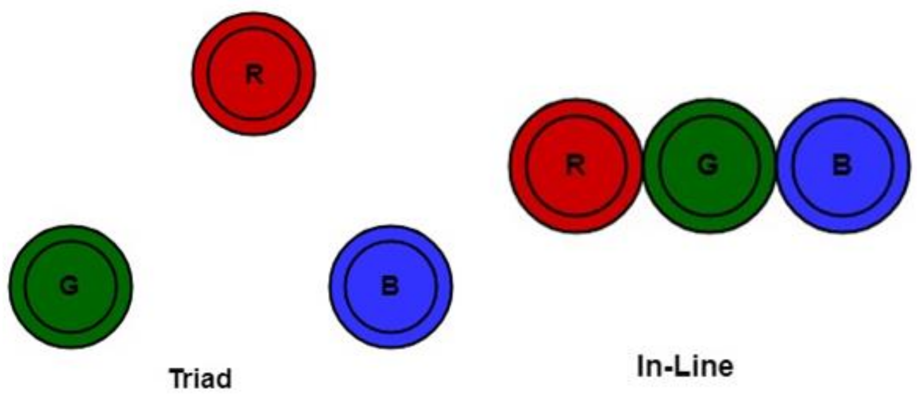
The Shadow mask CRT

<u>Construction:</u> It has 3 phosphor dots at each pixel position. Which emits red, green and blue lights.

This type of CRT has 3 electron guns, one for each colour dot and a shadow mask grid just behind the phosphor coated screen. Shadow mask grid is pierced with small round holes in a triangular pattern.

Arrangements:

- Triad arrangement of red, green, and blue guns. The deflection system of the CRT operates on all 3 electron beams simultaneously; the 3 electron beams are deflected and focused as a group onto the shadow mask, which contains a sequence of holes aligned with the phosphor-dot patterns. When the three beams pass through a hole in the shadow mask, they activate a dotted triangle, which occurs as a small colour spot on the screen. The phosphor dots in the triangles are organized so that each electron beam can activate only its corresponding colour dot when it passes through the shadow mask.
- Inline arrangement: Another configuration for the 3 electron guns is an Inline arrangement in which the 3 electron guns and the corresponding red-green-blue colour dots on the screen, are aligned along one scan line rather of in a triangular pattern. This inline arrangement of electron guns in easier to keep in alignment and is commonly used in high resolution colour CRT's.



Advantages	Disadvantages
Realistic Image	Expensive compared to monochrome CRT
Million different colours to be generated	Relatively poor resolution
Shadow scene are possible	Convergence Problem

Flat Panel Display

Refers to a class of video devices that have reduced volume, weight and power requirement compare to CRT.

There are 2 types of flat panel display

1. **Emissive Display:** Devices that convert electrical energy into light. Examples are Plasma Panel, thin film electroluminescent display, and LED (Light Emitting Diodes).
2. **Non-Emissive Display:** Use optical effects to convert sunlight or light from some other source into graphics patterns. Examples are LCD (Liquid Crystal Device).

Input Devices

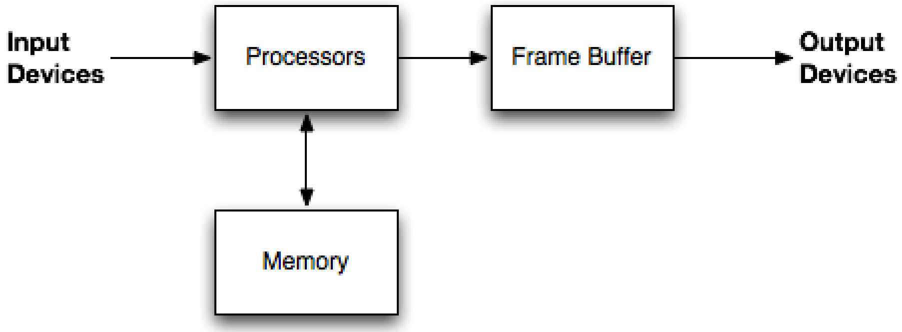
Hardware that is used to transfer input to the computer. The data can be in the form of text, graphics, sound, and text.

Example: Keyboard, Mouse, Trackball, Joystick, Image Scanner, etc.

Output Devices

Display data from the memory of the computer. Output can be text, numeric data, line, polygon, and other objects.

Example: CRT Display, LED Display, LCD Display, Printer, etc.



CONCLUSION

Familiarized with various computer graphics hardware and software and their working.

EXPERIMENT NUMBER 1(B)

FAMILIARIZATION OF DISPLAY STANDARDS

AIM

To familiarize with various computer graphics display standards.

INTRODUCTION

Every computer display is composed of a rectangular array of pixels (picture elements). The more pixels, the more detail may be shown in a given amount of space. This is termed the **resolution**. Each pixel may display a given number of distinct colour's; this is the **colour depth**. In order to work together, operating systems, graphics boards, and monitors support a number of standard video modes. As hardware has improved, and users have become more demanding, video modes have tended towards higher resolutions and greater colour depth. This requires a larger amount of dedicated memory, either on the computer motherboard, or on the graphics card.

In some modes, the software addresses each pixel individually; these are graphics modes. In order to speed processing, DOS systems often address pixels in character boxes of a fixed size; these are termed text modes. Each box is allocated a symbol from the ASCII character set. For example, mode 18 supports a resolution of 640x480, for a total of 307,200 addresses. Using a character block of 16x8, 30 lines of 80 characters each may be displayed. As there are now only 2,400 distinct addresses, performance is greatly enhanced. This explains the phenomenon familiar to DOS users: software takes a lot longer to display a graphic image than a screen full of text.

There have been several DOS graphics standards. Each is backwards compatible with those which have gone before. Resolutions beyond SVGA are commonly available, though standardization is poor. Various "standards" (Example EVGA, 8514/A, XGA, and XGA-2) support 1024x768 resolution at different colour depths. The trend in the last two years has been towards a proprietary mix of display modes driven by specialized graphics accelerators. These have been designed to optimize performance for particular applications (AutoCAD) or environments.

Some of the more common graphics cards and monitor standards for PC computers are:

● SVGA (Super VGA): One of the more popular labels placed on video cards and monitors. A SVGA card or monitor is capable of displaying more pixels (dots on the screen) and/or colours than basic VGA. For example, an SVGA graphics card may be able to display 16-bit colour with a resolution of 800x600 pixels.

● 3D Acceleration Cards: These cards include specialized hardware that speeds up the process of displaying three-dimensional images on the screen. They are usually designed to work with an SVGA monitor.

● VGA (Video Graphics Adapter): Currently the base standard for PC video cards and monitors. True VGA supports 16 colours at 640x480 pixels or 256 colours at 320x200 pixels.

● XGA: A standard used on some IBM PS/2 models. XGA supports 256 colours at 1024x728 pixels, or 16-bit colours at 640x480 pixels.

● EGA (Enhanced Graphics Adapter): Following CGA, an adapter that could display 16 colours with a screen resolution of 640x350 pixels.

● CGA (Colour Graphics Adapter): The first colour monitor and graphics cards for PC computers. Capable of producing 16 colours at 160x200 pixels.

● MDA (Monochrome Display Adapter): A monitor or graphics card that can display only one colour. No longer in common use but may be found on some older systems. Usually supports only text.

● **Hercules Graphics Card:** A card that enabled a PC to display graphics on an MDA monitor. CONCLUSION Familiarized with various computer graphics display standards.

CONCLUSION

Familiarized with various computer graphics display standards.

EXPERIMENT NUMBER 1(C)

FAMILIARIZATION OF OPENGL

AIM

To familiarize with OPENGL and its functionalities.

INTRODUCTION

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).

GLUT is designed to fill the need for a window system independent programming interface for OpenGL programs. The interface is designed to be simple yet still meet the needs of useful OpenGL programs. Removing window system operations from OpenGL is a sound decision because it allows the OpenGL graphics system to be retargeted to various systems including powerful but expensive graphics workstations as well as mass-production graphics systems like video games, set-top boxes for interactive television, and PCs.

GLUT simplifies the implementation of programs using OpenGL rendering. The GLUT **application programming interface (API)** requires very few routines to display a graphics scene rendered using OpenGL. The GLUT routines also take relatively few parameters.

Open Graphics Library (OpenGL) is a cross-language (language independent), cross-platform (platform independent) API for rendering 2D and 3D Vector Graphics (use of polygons to represent image). OpenGL API is designed mostly in hardware.

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 J Kilgard, to hide the complexities of differing window APIs.

Include Files

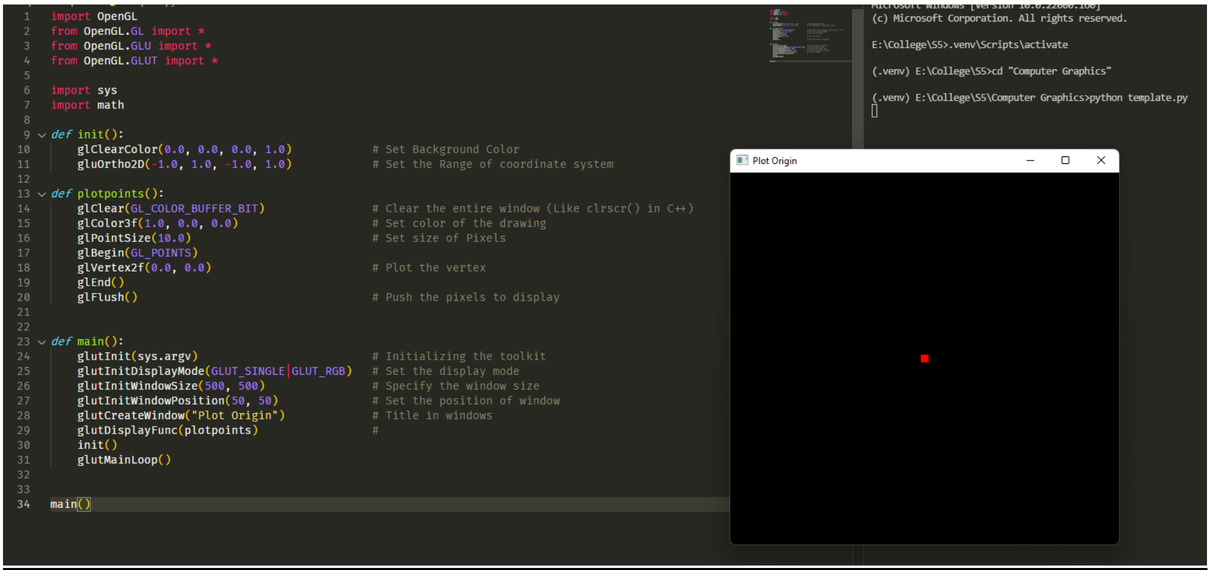
For all OpenGL applications, you want to include the gl.h header file in every file. Almost all OpenGL applications use GLU, the aforementioned OpenGL Utility Library, which also requires inclusion of the glu.h header file. So almost every OpenGL source file begins with:

```
import OpenGL
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *
```

Program to plot a point

```
import OpenGL
from OpenGL.GL import *
from OpenGL.GLU import *
from OpenGL.GLUT import *

import sys
import math

def init():
    glClearColor(0.0, 0.0, 0.0, 1.0)            # Set Background Color
    gluOrtho2D(-1.0, 1.0, -1.0, 1.0)            # Set the Range of coordinate system

def plotpoints():
    glClear(GL_COLOR_BUFFER_BIT)                # Clear the entire window (Like clrscr() in
 C++)
    glColor3f(1.0, 0.0, 0.0)                    # Set color of the drawing
    glPointSize(10.0)                           # Set size of Pixels
    glBegin(GL_POINTS)
    glVertex2f(0.0, 0.0)                        # Plot the vertex
    glEnd()
    glFlush()                                   # Push the pixels to display


def main():
    glutInit(sys.argv)                          # Initializing the toolkit
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB)   # Set the display mode
    glutInitWindowSize(500, 500)                # Specify the window size
    glutInitWindowPosition(50, 50)              # Set the position of window
    glutCreateWindow("Plot Origin")             # Title in windows
    glutDisplayFunc(plotpoints)                 #
    init()
    glutMainLoop()


main()
```

Output



Result

Familiarized with various OpenGL functionalities and a program to plot a point have been executed successfully and the point have been plotted on a window.

Abhinav Rajesh

20219003

CS-A S5