Unit-11

Introduction to graphics

If all pictures are built by concept of pixel then wondering how each picture differ that is how some picture appear more brighter while some other have a shady effect. All this is by the concept or technically terminology called as resolution.

So let’s have an insight on this important terminology.

Resolution is the number of rows that appear from top to bottom of a screen and in turn the number of pixels or pixel elements that appear from left to right on each scan line. Based on this resolution only the effect of picture appears on screen. In other words greater the resolution greater will be the clarity of picture. This is because greater the number of dots greater will be sharpness of picture. That is resolution value is directly proportional to clarity of picture. There are generally two modes available namely text and graphics. In a graphics mode we have generally the following adapters namely CGA called as Color Graphics Adapter, EGA and VGA. Each adapter differs in the way of generating colors and also in the number of colors produced by each adapter. Pixel being a picture element when we consider the graphics mode each pixel has a color associated with it. But the way these colors are used depends on adapters because each adapter differs in the way they handle colors and also in the number of colors supported. Having known about adapters now let us start knowing on how to start switching to graphics mode from text mode in other words how to start using pixel and resolution concepts. This is done by a function called initgraph ( ).This initgraph ( ) takes in it 2 main arguments as input namely gd and gm. In this gd has the number of mode which has the best resolution. This is very vital for graphics since the best resolution only gives a sharper picture as we have seen before. This value is obtained by using the function called as getgraphmode ( ) in C graphics. The other argument gm gives insight about the monitor used, the corresponding resolution of that, the colors that are available since this varies based on adapters supported. This value is obtained by using the function named as getmodename ( ) in Cgraphics.

Graphics function

There are numerous graphics functions available in c. But let us see some to have an understanding of how and where a pixel is placed in each when each of the graphics function gets invoked.

Function:

putpixel(x, y, color)

Purpose:

The functionality of this function is it put a pixel or in other words a dot at position x, y given in inputted argument. Here one must understand that the whole screen is imagined as a graph. In other words the pixel at the top left hand corner of the screen represents the value (0, 0).Here the color is the integer value associated with colors and when specified the picture element or the dot is placed with the appropriate color associated with that integer value.

Function:

line(x1, y1, x2, y2)

Purpose:

The functionality of this function is to draw a line from (x1,y1) to (x2,y2).Here also the coordinates are passed taking the pixel (0,0) at the top left hand corner of the screen as the origin. And also one must note that the line formed is by using number of pixels placed near each other.

Function:

getpixel(x, y)

Purpose:

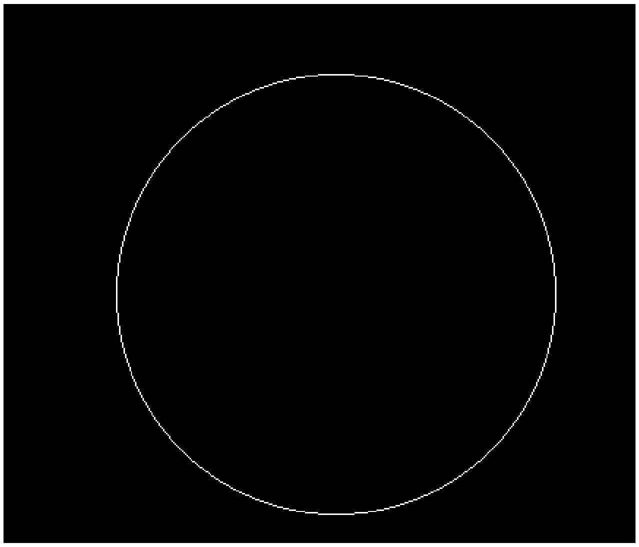
This function when invoked gets the color of the pixel specified. The color got will be the integer value associated with that color and hence the function gets an integer value as return value. So the smallest element on the graphics display screen is a pixel or a dot and the pixels are used in this way to place images in graphics screen in C language.

Example

- Write a program to draw a circle

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,"c:\\tc\\bgi");
circle(300,200,150);
setcolor(WHITE);
getch();
closegraph();
}
```

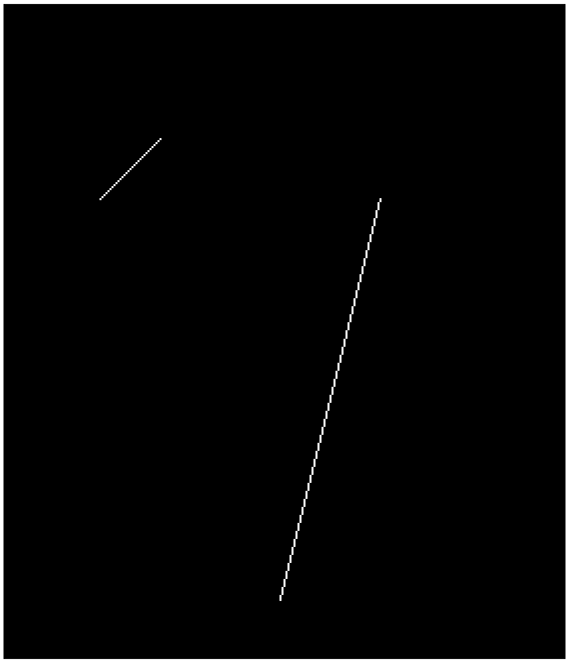
Output



- Write a program to draw line in C programming.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,"c:\\tc\\bgi");
line(90,70,60,100);
line(200,100,150,300);
setcolor(WHITE);
getch();
closegraph();
}
```

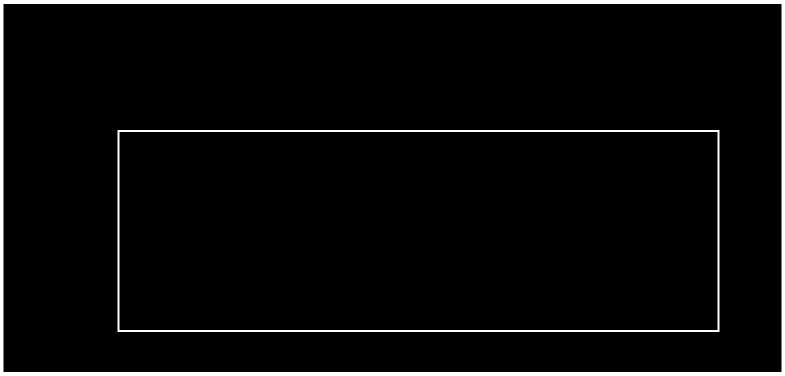
Output



- Write a program to draw a rectangle

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,"c:\\tc\\bgi");
rectangle(200,100,500,200);
setcolor(WHITE);
getch();
closegraph();
}
```

Output



- Write a program to draw a triangle.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,"c:\\tc\\bgi");
line(200,100,10,20);
line(10,20,50,60);
line(50,60,200,100);
setcolor(WHITE);
getch();
closegraph();
}
```

Output