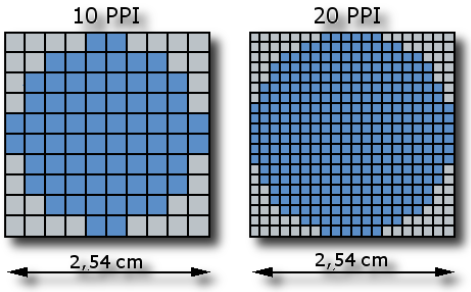
CHAPTER – 11

Graphics

Graphics are visual images or designs on some surface, such as a wall, canvas, screen, paper, or stone to inform, illustrate, or entertain. In contemporary usage, it includes a pictorial representation of data, as in c manufacture, in typesetting and the graphic arts, and in educational and recreational software. Images that are generated by a computer are called computer graphics.

Examples are photographs, drawings, line art, graphs, diagrams, typography, numbers, symbols, geometric designs, maps, engineering drawings, or other images.



In digital imaging, a pixel, pel, or picture element is a physical point in a raster image, or the smallest addressable element in an all points addressable display device; so it is the smallest controllable element of a picture represented on the screen.

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 **intigraph ( )**.This intigraph ( ) takes in it 2 main arguments as input namely **gd** and **gm.**

- ✗ 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 C graphics.

Graphics function in C and Pixel concept

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.

1. Function:

putpixel(x, y, color)

- Purpose:

T**he 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).

2. 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.

3. Function:

circle(x, y, radius);

- Purpose:

The header file graphics.h contains **circle()** function which draws a circle with center at (x, y) and given radius.

4. Function:

rectangle(int left, int top, int right, int bottom);

- Purpose:

rectangle() is used to draw a rectangle. Coordinates of left top and right bottom corner are required to draw the rectangle. left specifies the X-coordinate of top left corner, top specifies the Y-coordinate of top left corner, right specifies the X-coordinate of right bottom corner, bottom specifies the Y-coordinate of right bottom corner.

5. Function:

line(int x1, int y1, int x2, int y2);

- Purpose:

line function is used to draw a line from a point(x1,y1) to point(x2,y2) i.e. (x1,y1) and (x2,y2) are end points of the line.

6. Function:

arc(int x, int y, int start_angle, int end_angle, int radius);

- Purpose:

arc() function which draws an arc with center at (x, y) and given radius. start_angle is the starting point of angle and end_angle is the ending point of the angle. The value of the angle can vary from 0 to 360 degree.

7. Function:

ellipse(int x, int y, int start_angle, int end_angle, int x_radius, int y_radius)

- Purpose:

In this function x, y is the location of the ellipse. x_radius and y_radius decide the radius of form x and y. start_angle is the starting point of angle and end_angle is the ending point of angle. The value of angle can vary from 0 to 360 degree.

8. Function:

flood(int x, int y, int new_col, int old_col)

- Purpose:

floodfill() function is used to fill an enclosed area.

9. Function:

getmaxy();

- Purpose:

getmaxy() function which returns the maximum Y coordinate for current graphics mode and driver.

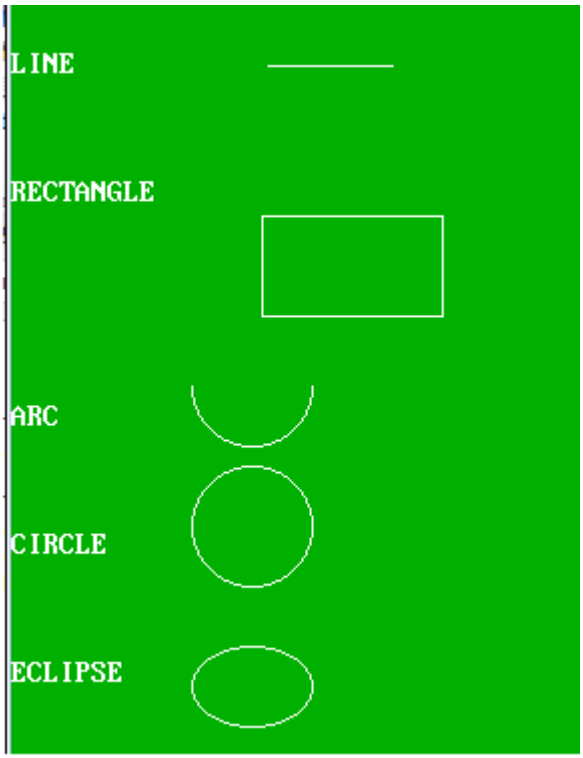
10. Function:

getmaxx();

- Purpose:

getmaxx() function which returns the maximum X coordinate for current graphics mode and driver.

```c
#include<graphics.h>
#include<conio.h>
void main()
{
   intgd=DETECT,gm;
   initgraph (&gd,&gm,"c:\\tc\\bgi");
   setbkcolor(GREEN);
   printf("\t\t\t\n\nLINE");
   line(50,40,190,40);
   printf("\t\t\n\n\n\nRECTANGLE");
   rectangle(125,115,215,165);
   printf("\t\t\t\n\n\n\n\n\n\nARC");
   arc(120,200,180,0,30);
   printf("\t\n\n\n\nCIRCLE");
   circle(120,270,30);
   printf("\t\n\n\n\nECLIPSE");
   ellipse(120,350,0,360,30,20);
   getch();
}
```



Other Sample Programs

https://www.javatpoint.com/computer-graphics-programs