Graphics in C

Dipendra Rai

Introduction

- Graphics is the way of providing visual way to see objects in action.
- A monitor is made up of different tiny spots made up of cross-section of columns and rows or ‘x’ and ‘y’ coordinates in it.
- The number of pixels (picture element) on the screen depends on the hardware (monitor and display card) and graphics driver for that hardware.
- Image on the monitor is displayed by illuminating or suppressing those dots (pixels). More the number of pixels on the monitor, better the quality of picture (resolution).

- Before understanding the graphics programming, first of all we must have knowledge of graphics driver.
- Graphics drivers are the collection of special files with extension **‘bgi’** which must be executed or loaded into memory of computer before using any other graphics functions.
- If such driver is not detected and initialized then no graphics can be seen or drawn.

- In graphics mode, we must follow some steps before starting to draw images which are explained below.
- **Step1:** Graphics driver must be detected before its initialization. It can be done by the calling graphics driver directly or indirectly.
- Direct call:

```
detectgraph(int *driver, int *mode);
```

- This function takes two parameters: one is for graphics driver and another is for graphics mode. If graphics driver is found in proper location, then the highest resolution available is provided with this auto detection. If graphics driver is not available, then a negative value is assigned for ‘driver’ parameter.
- Indirect call:

```
int driver = DETECT; (recommended)
```

Step2: After the graphics driver has been detected and assigned, it must be initialized which can be done with the ‘initgraph()’ library function.

- Syntax:

```
initgraph(int *driver, int *mode, char *path);
```

where,

*driver is the graphics driver which is an integer value and extracted from step1

*mode is the graphics mode which is an integer

*path is a string indicatingthe location of the graphics driver file. If nothing is specified (i.e. “ “ –a space), then it indicates that the graphics driver files(i.e. bgi files) are in the current directory.

Sample graphics code

```
#include<graphics.h>
#include<conio.h>
int main()
{
  int gd = DETECT, gm;
          initgraph(&gd,  &gm, "C:\\TC\\BGI");

  getch();
  closegraph();
  return 0;
}
```

Graphics function in C

- **getmaxx():** It gets the maximum numbers of columns (x-screen coordinate).
- **getmaxy():** It gets the maximum numbers of rows (y-screen coordinate).
- **line(x1,y1,x2,y2):** It draws a line startingfrom the coordinate (x1,y1) to (x2,y2).
- **circle(x,y,r):** It draws a circle with center having the coordinate (x,y) and radius r.
- **arc(x,y,sa,ea,hr,vr):** It draws an arc with the center at (x,y), starting angle sa, ending angle ea, horizontal radius hr and vertical radius vr.
- **ellipse(x,y,sa,ea,hr,vr):** It draws an ellipse with the center at (x,y), starting angle sa, ending angle ea, horizontal radius hr and vertical radius vr.

- **setlinestyle(linestyle,1,thickness):** It sets the pen style for drawing lines with line style between 0 to 4 and thickness between 1 to 3.
- Syntax:

```
setlinestyle(line_style, upattern, thickness);
```

Line_style can be any one of the given values:

Value	Line Style
0	solid line
1	dotted line
2	center line
3	line of hyphens
4	user defined line

- ‘uppattern’ is needed only when the line style is userdefined.
- ‘Thickness’ can be any one of the following:

1	normal width
2	medium width
3	thick width

- **settextstyle (font, direction, size):** It sets the style of the text.

  where font style –can be any value between 0 to 4.

  0 = default font

  1 = triplex font

  2 = small font

  3 = sans serif font

  4 = gothic font

  direction – can be any value between 0 to 1 where 0 is for horizontal and 1 is for vertical.

  Size can be any value between 0 to 10.
- **outtextxy(x, y, text):** It displays text in the x & y co-ordinate in the monitor.
- **outtext(text):** It displays given text in current coordinate in monitor.
- **setcolor(color_name):** It sets the color for the graphics to be displayed.
- **cleardevice():** It clears the graphics screen.
- **closegraph():** It closes the graphics driver after all the graphics functions are done.

Constructing Triangle

```
#include<graphics.h>
#include<conio.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();
        }
```

Drawing concentric circles

```
#include <graphics.h>
int main()
{
  int gd = DETECT, gm;
  int x = 320, y = 240, radius;

  initgraph(&gd, &gm, "C:\\TC\\BGI");

  for ( radius = 25; radius <= 125 ; radius = radius + 20)
    circle(x, y, radius);

  getch();
  closegraph();
  return 0;
}
```