Unit 1

Introduction to Graphics in C

Introduction

- In C_graphics, the graphics.h functions are used to draw different shapes like circles, rectangles, etc, display text(any message) in a different format (different fonts and colors).
- By using the functions in the header graphics.h, programs, animations, and different games can also be made.

Example

```
#include<stdio.h>
#include<graphics.h>
void main()
{
      // gm is Graphics mode which is a computer display
//mode that generates image using pixels.
      // DETECT is a macro defined in "graphics.h" header file
   int gd = DETECT, gm;

   // initgraph initializes the graphics system by loading a
   // graphics driver from disk
   initgraph(&gd, &gm, "c:\\turboc3\\bgi");
// circle function
   circle(250, 200, 50);

   getch();
   // closegraph function closes the graphics mode and
//deallocates all memory allocated by graphics system .
   closegraph();
   getch();
}
```

C graphics functions

1. arc
2. bar
3. bar3d
4. circle
5. cleardevice
6. closegraph
7. drawpoly
8. ellipse
9. fillellipse
10. fillpoly
11. floodfill
12. getarccords
13. getbkcolor
14. getcolor
15. getdrivername
16. getimage
17. getmaxcolor
18. getmaxx
19. getmaxy
20. getpixel
21. getx
22. gety
23. graphdefaults
24. grapherrormsg
25. imagesize
26. line
27. lineto
28. linerel
29. moveto
30. moverel
31. outtext
32. outtextxy
33. pieslice
34. putimage
35. putpixel
36. rectangle
37. sector
38. setbkcolor
39. setcolor
40. setfillstyle
41. setlinestyle
42. settextstyle
43. setviewport
44. textheight
45. textwidth

Line function in c

- 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.The code given below draws a line.
- Declaration: void line(int x1, int y1, int x2, int y2);

Example

```
#include <graphics.h>
#include <conio.h>
void main()
{
  int gd = DETECT, gm;
  initgraph(&gd, &gm, " c:\\turboc3\\bgi ");
  line(100, 100, 200, 200);
  getch();
  closegraph();
  return 0;
}
```

Bar function in c

- Declaration: void bar(int left, int top, int right, int bottom);
- Bar function is used to draw a 2-dimensional, rectangular filled in bar .
- Coordinates of left top and right bottom corner are required to draw the bar.
- 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.
- Current fill pattern and fill color is used to fill the bar. To change fill pattern and fill color use setfillstyle.

```
#include <graphics.h>
#include <conio.h>
main()
{
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
  bar(100, 100, 200, 200);
  getch();
  closegraph();
  return 0;
}
```

Circle function in c

- Declaration: void circle(int x, int y, int radius);
- Circle function is used to draw a circle with center (x,y) and third parameter specifies the radius of the circle. The code given below draws a circle.

```
#include<graphics.h>
#include<conio.h>
main()
{
   int gd = DETECT, gm;
   initgraph(&gd, &gm, "C:\\TC\\BGI");
   circle(100, 100, 50);

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

drawpoly function in c

- Drawpoly function is used to draw polygons i.e. triangle, rectangle, pentagon, hexagon etc.
- Declaration: void drawpoly(int num, int *polypoints);
- num indicates (n+1) number of points where n is the number of vertices in a polygon, polypoints points to a sequence of (n*2) integers .
- Each pair of integers gives x and y coordinates of a point on the polygon. We specify (n+1) points as first point coordinates should be equal to (n+1)th to draw a complete figure.
- To understand more clearly we will draw a triangle using drawpoly, consider for example,the array :-
  int points[] = { 320, 150, 420, 300, 250, 300, 320, 150};
- points array contains coordinates of triangle which are (320, 150), (420, 300) and (250, 300). Note that last point(320, 150) in array is same as first.

```
#include <graphics.h>
#include <conio.h>
main()
{
   int gd=DETECT,gm,points[]={320,150,420,300,250,300,320,150
};
   initgraph(&gd, &gm, "C:\\TC\\BGI");
   drawpoly(4, points);
   getch();
   closegraph();
   return 0;
}
```

Example

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
    int driver=DETECT,mode;
    int a[10]={150,150,200,150,120,200,50,200,150,150};
    initgraph(&driver,&mode,"c:\\turboc3\\bgi");
    drawpoly(5,a);
    getch();
    closegraph();
}
```

rectangle function in c

- Declaration: void rectangle(int left, int top, int right, int bottom);
- rectangle function 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. The code given below draws a rectangle.

```
#include<graphics.h>
#include<conio.h>
main()
{
  int gd = DETECT, gm;
  initgraph(&gd, &gm, "C:\\TC\\BGI");
  rectangle(100,100,200,200);
  getch();
  closegraph();
  return 0;
}
```

setcolor function in c

- Declaration: void setcolor(int color);
- In Turbo Graphics each color is assigned a number. Total 16 colors are available. Strictly speaking number of available colors depends on current graphics mode and driver.
- For Example :- BLACK is assigned 0, RED is assigned 4 etc. setcolor function is used to change the current drawing color.e.g. setcolor(RED) or setcolor(4) changes the current drawing color to RED. Remember that default drawing color is WHITE.

```
#include<graphics.h>
#include<conio.h>
main()
{
  int gd = DETECT, gm;
  initgraph(&gd,&gm,"C:\\TC\\BGI");
  circle(100,100,50);         /* drawn in white color */
  setcolor(RED);
  circle(200,200,50);         /* drawn in red color   */

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

closegraph function in c

- closegraph function closes the graphics mode, deallocates all memory allocated by graphics system and restores the screen to the mode it was in before you called initgraph.
- Declaration: void closegraph();