Paper 102: Programming & Problem solving through C

Lecture-29:Graphics II

Aiusha V.Hujon, Dept. of Computer Science, SAC

Program to find screen size

```
#include<graphics.h>
#include<conio.h>

void main(void){
   int driver=DETECT;
   int mode;
   int maxx, maxy;
   int left, top, right, bot;
   initgraph(&driver, &mode, "c:\\tc\\bgi");
   maxx=getmaxx();
   maxy=getmaxy();
   left=top=0;
   right=maxx;
   bot=maxy;
   rectangle(left, top, right, bot);
   getch();
   closegraph();
}
```

Line Width and Style

- setlinestyle() function indicates a parameter to specify different kinds of dotted lines, but this applies only to straight line
- takes three arguments that specify, style, pattern, and thickness

```
void far setlinestyle(int style, int
        unsigned pattern,int thickness)
```

Line Width and Style, Cont'd

- *style* has the following options

Value	Constant
0	SOLID_LINE
1	DOTTED_LINE
2	CENTER_LINE
3	DASHED_LINE
4	USERBIT_LINE

- If USERBIT_LINE is used, the *pattern* argument defines a 16 bit value

  thickness has two possible values:

Value	Constant
1	NORMAL_WIDTH
3	THICK_WIDTH

Example

```
#include<graphics.h>
#include<conio.h>
#define IGNORED 0
void main(void){
  int driver=DETECT, mode;
  int x1=0, y1=0;
  int x2=199, y2=199;
  int xC=100, yC=100;
  int radius=99;
  initgraph(&driver, &mode, "c:\\tc\\bgi");
  setlinestyle(DASHED_LINE, IGNORED, THICK_WIDTH);
  line(x1, y1, x2, y2);
  circle(xC, yC, radius);
  getch();
  closegraph();
}
```

Using Color

- setcolor() function allows us to specify colors for lines, circles,

```
void far setcolor(int color)
```

- Colors used will depend on graphics mode. For standard vga, they are

Value	Constant
0	BLACK
1	BLUE
2	GREEN
3	CYAN
4	RED
5	MAGENTA
6	BROWN
7	LIGHTGRAY

Using Color, Cont'd

Value	Constant
8	DARKGRAY
9	LIGHTBLUE
10	LIGHTGREEN
11	LIGHTCYAN
12	LIGHTRED
13	LIGHTMAGENTA
14	YELLOW
15	WHITE

Example - Using Color

```
#include<graphics.h>
#include<conio.h>
#define IGNORED 0
void main(void){
   int driver=DETECT, mode;
   int x1=0, y1=0;
   int x2=199, y2=199;
   int xC=100, yC=100;
   int radius=99;
   initgraph(&driver, &mode, "c:\\tc\\bgi");
   setlinestyle(DASHED_LINE, IGNORED, THICK_WIDTH);
   setcolor(GREEN);
   line(x1, y1, x2, y2);
   setcolor(RED);
   circle(xC, yC, radius);
   getch();
   closegraph();
}
```

Ellipses

- ellipse() function allows us to draw ellipses

```
void far ellipse(int xE, int xY, int stAngle,
                 int endAngle,int xRad, int yRad)
```

- where

xE, yE	center of ellipse
stAngle, endAngle	starting and ending angles
xRad, yRad	radii of ellipse along x and y directions respectively

Example- Draw an Ellipse

```
#include<graphics.h>
#include<conio.h>
void main(void){
   int driver=DETECT, mode;
   int xE=150, yE=100;
   int xRad=150, yRad;
   int stAngle=0, endAngle=360;
   int radius=99;
   initgraph(&driver, &mode, "c:\\tc\\bgi");
   for(yRad=0; yRad<100; yRad+=10)
       ellipse(xE, yE, stAngle, endAngle, xRad, yRad);
   getch();
   closegraph();
}
```

Polygon

- Polygons are figures containing arbitrary number of straight line segments
- drawpoly() function allows two parameters,
  - Number of points to connect
  - Address of a list containing the points to connect. This is an array containing x and y values
- Syntax

```
void far drawpoly(int number,int far *addrList)
```

Example- Draw an Polygon

```
#include <graphics.h>
#include <conio.h>

void main(void)
{
   int pane[]={150,50, 180,20, 180,120, 150,150, 150,50};
   int driver=DETECT, mode;
   initgraph(&driver, &mode,"c:\\tc\\bgi");

   drawpoly(5, pane);
   getch();
   closegraph();
}
```

Class assignment

- Wap to draw a triangle inside a circle on the screen
- Wap to draw the two ellipse intersecting one another.