Paper 102: Programming & Problem solving through C

Lecture-28:Graphics

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

Graphics functions in C

- Graphics can be used in almost any computer program
- Graphics can improve your program
- While other techniques are available, the use of graphic functions is most convenient to obtain graphics
- To run graphics, you will need a graphics adapter board and a graphics display adapter
- A character-only display will not work

Preparing for Graphics Functions

- There are few preliminary steps to perform before using graphics:
  - You must specify the linker that you plan to use graphics
  - The “graphics.h” header file must be included
  - Initialize the graphics system
- To inform linker about graphics, select Linker from Options menu and select Libraries submenu.
- Click on Graphics Library checkbox
- The Standard Runtime should already be checked

Graphics Functions, Cont’d

- “graphics.h” contains prototypes for all graphics functions.
- Many graphics functions use constants instead of numbers to specify values as colors, patterns and fonts
- graphics.h also contains the definitions of these constants
- The graphics system can be initialized by using function *initgraph()*.
- This function loads graphics driver (probably egavga.bgi) from disk and changes the display to appropriate graphics mode.
- Function *closegraph()* shuts down the graphics system and returns display to previous mode

Coordinate system

- The coordinate system of a graphics display has been predefined as follows
- In the Cartesian system, the origin (0,0) is the left bottom corner
- x goes on increasing to the right and y increases upward
- On graphics screen, the origin (0,0) is left top corner
- x still increases to the right, but y increases downwards

initgraph()

void far initgraph(int far *graphdriver,int far *graphmode, char far *pathtodriver);

- This function is of type void far
- Takes three arguments of type int far *.
- These are
  - address where graphics driver will be specified
  - Address where the mode will be specified
  - Address of a string holding the pathname of the file containing graphics driver
- The first argument (graphics driver) can be set to DETECT, graphics system will automatically select the graphics mode with highest resolution
- If DETECT is used, the mode variable need not be set

Lines and Circles

- line() is the function that draws lines

```
void far line(int x1, int y1, int x2, int y2);
```

- Requires four parameters,
  - x and y coordinates of the start of the line, and
  - x and y coordinates of the end
- circle() is the function that draws circle

```
void far circle(int xC, int yC, int radius);
```

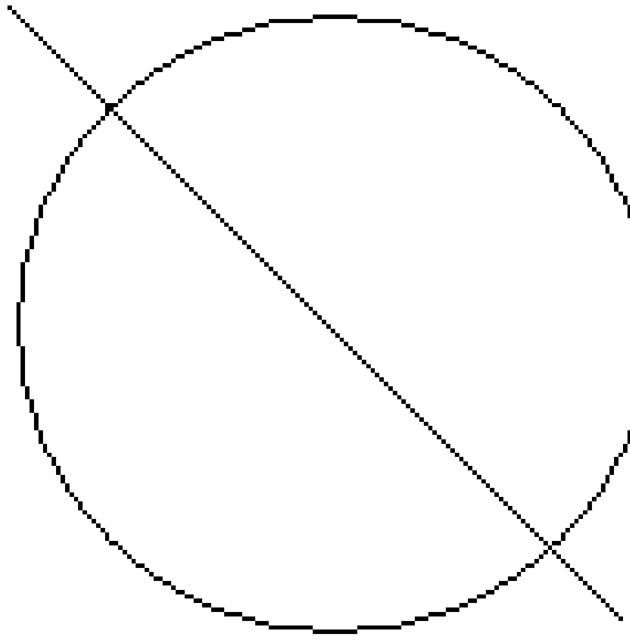
- Requires the x and y coordinate of center, and the circle’s radius

A simple example

```
#include<graphics.h>
#include<conio.h>
void main(void){
  int driver, mode;
  int x1=0, y1=0;
  int x2=199, y2=199;
  int xC=100, yC=100;
  int radius=90;

  driver=DETECT;
  initgraph(&driver, &mode, "h:\\software\\tc\\bgi");
  line(x1, y1, x2, y2);
  circle(xC, yC, radius);
  getch();
  closegraph();
}
```

A simple example: Output



Viewing the graphics

- You must view the graphics before the program ends
- Once display system has changed from graphics back to text, any graphics drawn is lost
- Thus, we have used getch()
- closegraph() restores the original text mode
  - In IDE, the use closegraph() is optional
  - In DOS, you may lose your cursor or end up with wrong size characters
  - De-allocates memory used by graphics system

Class assignment

- Write a program to draw concentric circles on the screen using graphics mode.
- Write a program to fill the screen with a grid, where each cell in the grid is of size 50 x 50 pixels.