Unit 11
Introduction to Graphics

Concepts of Graphics

- Graphics in C involves creating images and visual designs using basic shapes, colors, and other graphical elements.
- C provides a number of graphics libraries that allow developers to create graphical applications and games.

graphics.h Header File

- graphics.h is a C header file that provides a library of functions for creating basic graphics programs in C.
- It was a part of the old Borland Turbo C and Microsoft Visual C++ compilers but is not a standard C library and is not supported by modern compilers.
- The graphics.h library includes functions for drawing basic shapes such as lines, rectangles, circles, and polygons, as well as functions for setting colors, fonts, and text modes.
- It also provides functions for handling mouse and keyboard input.

Graphics Initialization and Modes

- Graphics options are used in c-programming to draw different graphical shapes.
- First of all we have to call the initgraph() function that will initialize the graphics mode on the computer.
- Call to function initgraph() is done as:

```
initgraph(&gdriver, &gmode, "path_to_driver");
```

- initgraph() initializes the graphics system by loading a graphics driver from disk (or validating a registered driver) then putting the system into graphics mode.
- initgraph() also resets all graphics settings (color, palette, current position, viewport, etc.) to their defaults, then resets graph result to 0.

gdriver:

- It is declared as integer variable that specifies the graphics driver to be used.

gmode:

- It is also declared as integer variable which specifies the initial graphics mode (unless gdriver = DETECT).
- If gdriver = DETECT, initgraph sets gmode to the highest resolution available for the detected driver.

path_to_driver:

- Specifies the directory path where initgraph() looks for graphics drivers (e.g. egavga.bgi in turbo c++).

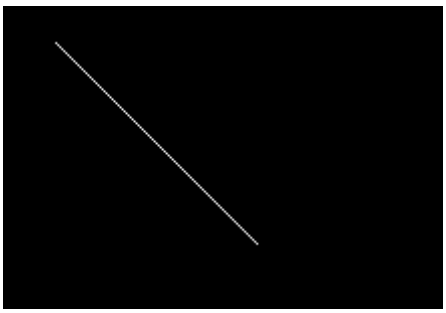
1. If the driver is not there, initgraph() looks in the current directory.
2. If path_to_driver is null, the driver files must be in the current directory.

- After a call to initgraph, gdriver is set to the current graphics driver, and gmode is set to the current graphics mode.
- gdriver = DETECT auto-detects the attached video adapter at run time and pick the corresponding driver.
- If we tell initgraph to auto-detect, it calls detectgraph to select a graphics driver and mode.
- Normally, initgraph() loads a graphics driver by allocating memory for the driver, then loading the appropriate .BGI file from disk.

Example 1: Draw a line

```
/* C graphics program to draw a line */
#include<graphics.h>
int main()
{
        int gd = DETECT, gm;
        /* initialization of graphic mode */
        initgraph(&gd, &gm, (char*)"");
        line(100,100,200, 200);
        getch();
        closegraph();
        return 0;
}
```

Output:



Basic Graphics Functions in C

Here are brief definitions and syntax of some basic graphics functions in C:

1. line(): Draws a line between two specified points.

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

2. arc(): Draws a circular arc of specified radius and angle.

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

3. circle(): Draws a circle with specified center and radius.

Syntax: circle(int x, int y, int radius);

4. ellipse(): Draws an ellipse with specified center, horizontal radius, and vertical radius.

   Syntax: ellipse(int x, int y, int start_angle, int end_angle, int x_radius, int y_radius);

5. floodfill(): Fills a bounded region with a specified color.

   Syntax: floodfill(int x, int y, int border_color);

6. getmaxx(): Returns the maximum x-coordinate of the screen.

   Syntax: int max_x = getmaxx();

7. getmaxy(): Returns the maximum y-coordinate of the screen.

   Syntax: int max_y = getmaxy();

In these functions, the arguments represent the following:

- x1, y1, x2, y2: The starting and ending coordinates of the line.
- x, y: The center point of the arc or circle.
- start_angle, end_angle: The starting and ending angles of the arc.
- radius: The radius of the circle or arc.
- x_radius, y_radius: The horizontal and vertical radii of the ellipse.
- border_color: The color of the border of the region to be filled.

The graphics functions in C are part of the graphics.h library, which is an outdated library and is not supported by modern compilers.

Exercise

1. Why do we need graphics functions? Write a program to draw a circle. (5) [TU 2074]
2. Write a program to draw a circle using graphics function. (5) [TU 2077]
3. Write a program to draw a line using graphics function. (5) [TU 2078]