Computer Graphics Lab Manual – Lab 1

Course: Computer Graphics

Programming Language: C

Lab Title: Introduction to Computer Graphics & Basic Drawing Using C

1. Objectives

- To understand the basic structure of a graphics program in C.
- To set up and initialize the graphics mode.
- To draw basic shapes (pixel, line, rectangle, circle) using C graphics functions.
- To become familiar with the concept of coordinate system in computer graphics.
- To understand colors in graphics and their usage.

2. Theory

2.1 Graphics Mode Initialization

Computer graphics in C (using BGI/WinBGIm libraries) requires:

- Initializing graphics driver
- Initializing graphics mode
- Specifying the directory of BGI files (or using current folder)
- The coordinate system begins at:
- **(0,0)** = top-left corner of the screen
- Coordinates increase:
- **+x → right**
- **+y → downward**

2.2 Basic Drawing Functions in C

1. `putpixel(x, y, color)`

Draws a single pixel at (x,y) with specified color (0-15). Only `putpixel()` accepts color directly.

2. `line(x1, y1, x2, y2)`

Draws a straight line between (x1, y1) and (x2, y2). Color must be set with `setcolor()`.

3. `rectangle(left, top, right, bottom)`

Draws an outline of a rectangle. Color must be set with `setcolor()`.

4. `circle(x, y, r)`

Draws a circle with center (x, y) and radius r. Color set by `setcolor()`.

5. `setcolor(color)`

Sets the drawing color for lines, rectangles, circles, etc. Color names must be in CAPITAL letters or use integer codes 0-15.

6. `setbkcolor(color)`

Sets the background color.

7. `closegraph()`

Closes the graphics mode.

3. Colors in BGI Graphics

Color Name	Code
BLACK	0
BLUE	1
GREEN	2

CYAN	3
RED	4
MAGENTA	5
BROWN	6
LIGHTGRAY	7
DARKGRAY	8
LIGHTBLUE	9
LIGHTGREEN	10
LIGHTCYAN	11
LIGHTRED	12
LIGHTMAGENTA	13
YELLOW	14
WHITE	15

True colors (hex RGB) are not supported in classic BGI.

4. Basic Structure of a C-graphics program:

```
#include <stdio.h>      // Standard Input/Output
#include <graphics.h>   // MUST be included for BGI functions
#include <conio.h>      // For getch() to pause the screen

void main() {
 // 1. Declare Graphics Variables
 int gd = DETECT, gm;
// gd: Graphics Driver (DETECT asks system to auto-detect best driver)
// gm: Graphics Mode

// 2. Initialize the Graphics Mode
// The third argument is the path to the BGI files (*.BGI).
// Use "" if BGI files are in the current working directory,
// or the full path (e.g., "C:\\TurboC3\\BGI").
initgraph(&gd, &gm, "C:\\TurboC3\\BGI");

// 3. Write Drawing Code Here
// Example: Draw a line from (100, 100) to (200, 200)
setcolor(WHITE); // Set color for the line
line(100, 100, 200, 200);

 // 4. Pause and Close
 getch();       // Waits for a key press
 closegraph();   // Closes the graphics mode

}
```

5. Program Examples

Program 1: Initialize Graphics Mode

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

```
int main()
{
int gd = DETECT, gm;
initgraph(&gd, &gm, ""); // BGI files in same folder

outtext("Graphics Mode Initialized Successfully");

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

Program 2: Draw Basic Shapes and Colors

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

int main()
{
int gd = DETECT, gm;
initgraph(&gd, &gm, "");

// Draw pixel in RED
putpixel(100, 100, 4);

// Draw a BLUE line
setcolor(BLUE);
line(50, 50, 200, 50);

// Draw a GREEN rectangle
setcolor(GREEN);
rectangle(50, 100, 200, 200);

// Draw a YELLOW circle
setcolor(YELLOW);
circle(300, 150, 50);
getch();
closegraph();
return 0;
}
```

Program 3: Display text name using `outtext()`

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

int main()
{
   int gd = DETECT, gm;
   initgraph(&gd, &gm, "");

   setcolor(YELLOW);
   outtextxy(100, 100, "This is computer graphics lab");

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

6. Lab Tasks / Exercises

1. Write a program to initialize graphics mode.
2. Draw the following shapes with different colors:
    a. A line
    b. A rectangle
    c. A circle
    d. Two or more pixels at different locations
3. Display your name using `outtext()`.