COMPUTER GRAPHICS AND MULTIMEDIA

LAB ASSIGNMENT 2

By: Sparsh Arya

Registration number: 17BEC0656

Slot: L3+L4

Bresenham’s circle drawing algorithm

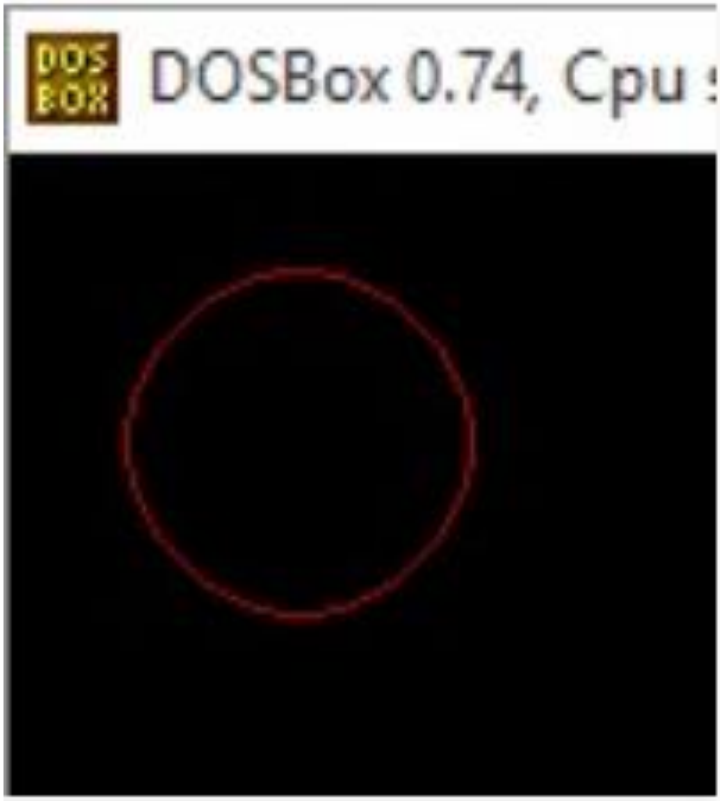
Algorithm

1. Set initial values of (xc, yc) and (x, y)
2. Set decision parameter d to d = 3 – (2 * r).
3. Repeat steps 4 to 8 until x < = y
4. Call drawCircle(int xc, int yc, int x, int y) function.
5. Increment value of x.
6. If d < 0, set d = d + (4*x) + 6
7. Else, set d = d + 4 * (x – y) + 10 and decrement y by 1.
8. Call drawCircle(int xc, int yc, int x, int y) function

Code

```
// C-program for circle drawing
// using Bresenham’s Algorithm
// in computer-graphics
#include <stdio.h>
#include <dos.h>
#include <graphics.h>

// Function to put pixels
// at subsequence points
void drawCircle(int xc, int yc, int x, int y)
{
    putpixel(xc+x, yc+y, RED);
    putpixel(xc-x, yc+y, RED);
    putpixel(xc+x, yc-y, RED);
    putpixel(xc-x, yc-y, RED);
    putpixel(xc+y, yc+x, RED);
    putpixel(xc-y, yc+x, RED);
    putpixel(xc+y, yc-x, RED);
    putpixel(xc-y, yc-x, RED);
}

// Function for circle-generation
// using Bresenham's algorithm
void circleBres(int xc, int yc, int r)
{
    int x = 0, y = r;
    int d = 3 - 2 * r;
    while (y >= x)
    {
        // for each pixel we will
        // draw all eight pixels
        drawCircle(xc, yc, x, y);
        x++;

        // check for decision parameter
        // and correspondingly
        // update d, x, y
        if (d > 0)
        {
            y--;
            d = d + 4 * (x - y) + 10;
        }
        else
            d = d + 4 * x + 6;
        drawCircle(xc, yc, x, y);
        delay(50);
    }
}


// driver function
int main()
{
    int xc = 50, yc = 50, r2 = 30;
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");  // initialize graph
    circleBres(xc, yc, r);    // function call
    return 0;
}
```

OUTPUT



The end.