Program 3

Aim: Write a program to draw/create a teddy bear (circular face tummy eyes and mouth) using Midpoint Circle drawing Algorithm.

Theory:

As in the raster line algorithm, we sample at unit intervals and determine the closest pixel position to the specified circle path at each step. For a given radius r and screen center position (xc, yc), we can first set up our algorithm to calculate pixel positions around a circle path centered at the coordinate origin (0, 0). Then each calculated position (x, y) is moved to its proper screen position by adding x to xc and yto yc.

Along the circle section from x = 0 to x = y in the first quadrant, the slope of the curve varies from 0 to -1. Therefore, we can take unit steps in the positive x direction over this octant and use a decision parameter to determine which of the two possible y positions is closer to the circle path.

Algorithm:

1. Calculate the initial value of the decision parameter as

$$p0=5/4-r$$

2. At each xk position, starting at k = 0, perform the following test: If pk< 0, the next point along the circle centered on (0, 0) is (xk+ 1, yk) and

$$Pk+1 = Pk + 2xk+1+ 1$$

Otherwise, the next point along the circle is (xk + 1, yk - I) and where

$$2xk+1 = 2xk+ 2 \text{ and } 2yk+1 == 2yk - 2.$$

3. Determine symmetry points in the other seven octants.

4. Move each calculated pixel position (x, y) onto the circular path centered on (xc, yc) and plot the coordinate values:

$$y=y+yc \text{ and } x =x+xc$$

5. Repeat steps 3 through 5 until x > y.

Code:

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

void setpixel(int xc, int yc, int x, int y){
    putpixel(xc + x, yc + y, 15);
    putpixel(xc + x, yc - y, 15);
    putpixel(xc - x, yc + y, 15);
    putpixel(xc - x, yc - y, 15);
    putpixel(xc + y, yc + x, 15);
    putpixel(xc + y, yc - x, 15);
    putpixel(xc - y, yc + x, 15);
    putpixel(xc - y, yc - x, 15);
}

void midptcircle(int xc, int yc, int r){
    int p = 1 - r;
    int x = 0, y = r;
    setpixel(xc, yc, x, y);
    while(x<y){
        x++;
        if(p<0){
            p += 2 * x + 1;
        }else{
            y--;
            p += 2 * (x - y) + 1;
        }
        setpixel(xc, yc, x, y);
    }
}

int main(){

    int g_mode,g_driver=DETECT;
    initgraph(&g_driver,&g_mode,"C:\\TURBOC3\\BGI");
```

```
    //body
    midptcircle(300, 290, 90);

    //face
    midptcircle(300, 150, 50);

    //legs
    midptcircle(250, 380, 28);
    midptcircle(350, 380, 28);

    //hands
    midptcircle(210, 230, 30);
    midptcircle(390, 230, 30);

    //ears
    midptcircle(250, 100, 20);
    midptcircle(250, 100, 15);
    midptcircle(350, 100, 20);
    midptcircle(350, 100, 15);

    //eyes
    midptcircle(280, 140, 6);
    midptcircle(280, 140, 3);
    midptcircle(320, 140, 6);
    midptcircle(320, 140, 3);

    //mouth
    midptcircle(300, 172, 20);
    midptcircle(300, 156, 4);
    midptcircle(300, 177, 10);

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

Output: