EXP 05:- BOUNDARY FILL & FLOOD FILL

Aim: To implement Area Filling Algorithm: Boundary Fill, Flood Fill.

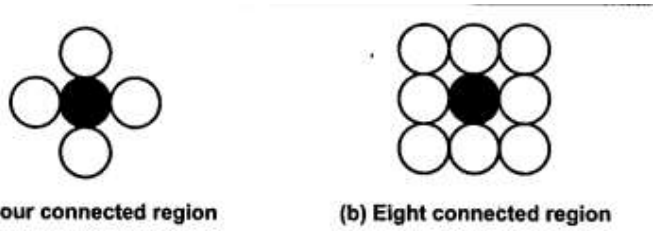
Objective:

Polygon is an ordered list of vertices as shown in the following figure. For filling polygons with particular colors, we need to determine the pixels falling on the border of the polygon and those which fall inside the polygon. Objective is to demonstrate the procedure for filling polygons using different techniques.

Theory:

1) Boundary Fill algorithm –

Start at a point inside a region and paint the interior outward toward the boundary. If the boundary is specified in a single color, the fill algorithm processed outward pixel by pixel until the boundary color is encountered. A boundary-fill procedure accepts as input the coordinate of the interior point (x, y), a fill color, and a boundary color.



(a) Four connected region (b) Eight connected region

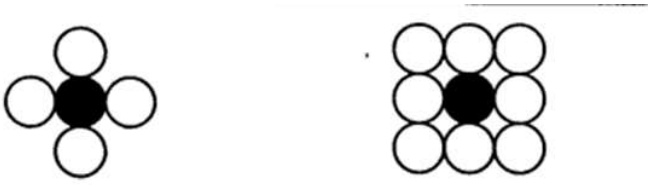
Procedure:

```
boundary_fill (x, y, f_color, b_color)
{
if (getpixel (x, y) != b_colour && getpixel (x, y) != f_colour)
        {
        putpixel (x, y, f_colour)
        boundary_fill (x + 1, y, f_colour, b_colour);
        boundary_fill (x, y + 1, f_colour, b_colour);
        boundary_fill (x - 1, y, f_colour, b_colour);
        boundary_fill (x, y - 1, f_colour, b_colour);
        }
}
```

2) Flood Fill algorithm –

Sometimes we want to fill an area that is not defined within a single color boundary. We paint such areas by replacing a specified interior color instead of searching for a boundary color value. This approach is called a flood-fill algorithm.

1. We start from a specified interior pixel (x, y) and reassign all pixel values that are currently set to a given interior color with the desired fill color.
2. If the area has more than one interior color, we can first reassign pixel values so that all interior pixels have the same color.
3. Using either 4-connected or 8-connected approach, we then step through pixel positions until all interior pixels have been repainted.



(a) Four connected region (b) Eight connected region

Procedure -

```
flood_fill (x, y, old_color, new_color)
{
if (getpixel (x, y) = old_colour)
        {
        putpixel (x, y, new_colour);
        flood_fill (x + 1, y, old_colour, new_colour);
        flood_fill (x - 1, y, old_colour, new_colour);
        flood_fill (x, y + 1, old_colour, new_colour);
        flood_fill (x, y - 1, old_colour, new_colour);
        flood_fill (x + 1, y + 1, old_colour, new_colour);
        flood_fill (x - 1, y - 1, old_colour, new_colour);
        flood_fill (x + 1, y - 1, old_colour, new_colour);
        flood_fill (x - 1, y + 1, old_colour, new_colour);
        }
}
```

Program:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void boundaryfill(int x, int y,int fill_c, int boundary_c)
```

```
{ int current;
 current = getpixel(x,y);
 if(current != boundary_c && current != fill_c)
 {
 putpixel(x,y,fill_c);
 boundaryfill(x,y+1,fill_c,boundary_c);
 boundaryfill(x,y-1,fill_c,boundary_c);
  boundaryfill(x+1,y,fill_c,boundary_c);
   boundaryfill(x-1,y,fill_c,boundary_c);
   }
 }

void ff(int x,int y,int old_c, int new_c)
{
int current;
current = getpixel(x,y);
if(current == old_c)
{
putpixel(x,y,new_c);
ff(x,y+1,old_c,new_c);
ff(x,y-1,old_c,new_c);
ff(x+1,y,old_c,new_c);
ff(x-1,y,old_c,new_c);
}
}

void main() {
    int x,y,x1,y1,r,gd=DETECT,gm,d;
    x=0;
    initgraph(&gd,&gm,"..\\bgi");
    printf("Enter radius");
    scanf("%d",&r);
    printf("Enter centre coordinates (x,y & x1,y1)");
    scanf("%d %d %d %d",&x,&y,&x1,&y1);
    circle(x,y,r);
    circle(x1,y1,r);
    boundaryfill(x,y,5,15);
    ff(x1,y1,0,9);
    getch();
```

```
    closegraph();
}
```

Output:-

Conclusion:-

1] BOUNDARY FILL:-Boundary fill algorithm is a technique used in computer graphics to fill the interior of a closed shape with a specified color.\

2] FLOOD FILL:-Flood fill algorithm is a graphics technique to fill contagious area with a specified color.it starts from a seed point and spreads outwards, cha nging the color of adjacent pixels until a boundary color is encountered