**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:**

**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.



**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);

}

}

**Program:**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<dos.h>

void boundaryfill(int x, int y, int f\_color, int b\_color)

{

if(getpixel(x,y) != b\_color && getpixel(x,y) != f\_color)

{

putpixel(x,y,f\_color);

delay(10);

boundaryfill(x+1,y,f\_color,b\_color);

boundaryfill(x,y+1,f\_color,b\_color);

boundaryfill(x-1,y,f\_color,b\_color);

boundaryfill(x,y-1,f\_color,b\_color);

}

}

int main()

{

int gm,gd,radius;

int x, y;

clrscr();

detectgraph(&gd,&gm);

initgraph(&gd,&gm,"..\\BGI");

printf("Enter x and y positions for circle\n");

scanf("%d %d",&x, &y);

printf("Enter radius of circle\n");

scanf("%d",&radius);

circle(x, y, radius);

boundaryfill(x, y, 3, 15);

getch();

closegraph();

return 0;

}

**Output:** 

**Conclusion:** Comment on

1. Importance of Flood fill
2. Limitation of methods
3. Usefulness of method