Harsh Kaushik
20103085
C2.1

Page No.
Date

Experiment 5

Aim: Implement flood fill and boundary fill to fill a polygon

Theory:
Flood fill is an algorithm mainly used to determine a bounded area connected to a given node in a multimensional array. Boundary fill is the algorithm used frequently in computer graphics to fill a desired color inside a closed polygon

Algorithm:

Flood fill:

```
flood fill (x, y, old color, new color){
  put pixel (x, y, new color)
  flood fill (x+1, y, old-color, new color)
  flood fill (x-1, y, old-color, new-color)
  flood fill (x, y+1, old, new-color)
  flood fill (x, y-1, oldcolor, new-color);
  flood fill (x+1, y+1, old color, new color);
  flood fill (x-1, y-1, old color, new color);
  floodfill (x-1, y-1, old-c, new-c);
  flood fill (x-1, y+1, old-c, new c);
}
```

Boundary fill algorithm:

```
boundary fill (x, y, f-colour, b-color){

    if (getpixel (x,y) != b-color && getpixel (x,y) != f-color
      putpixel (x,y ,f-color)
      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);
}.
```

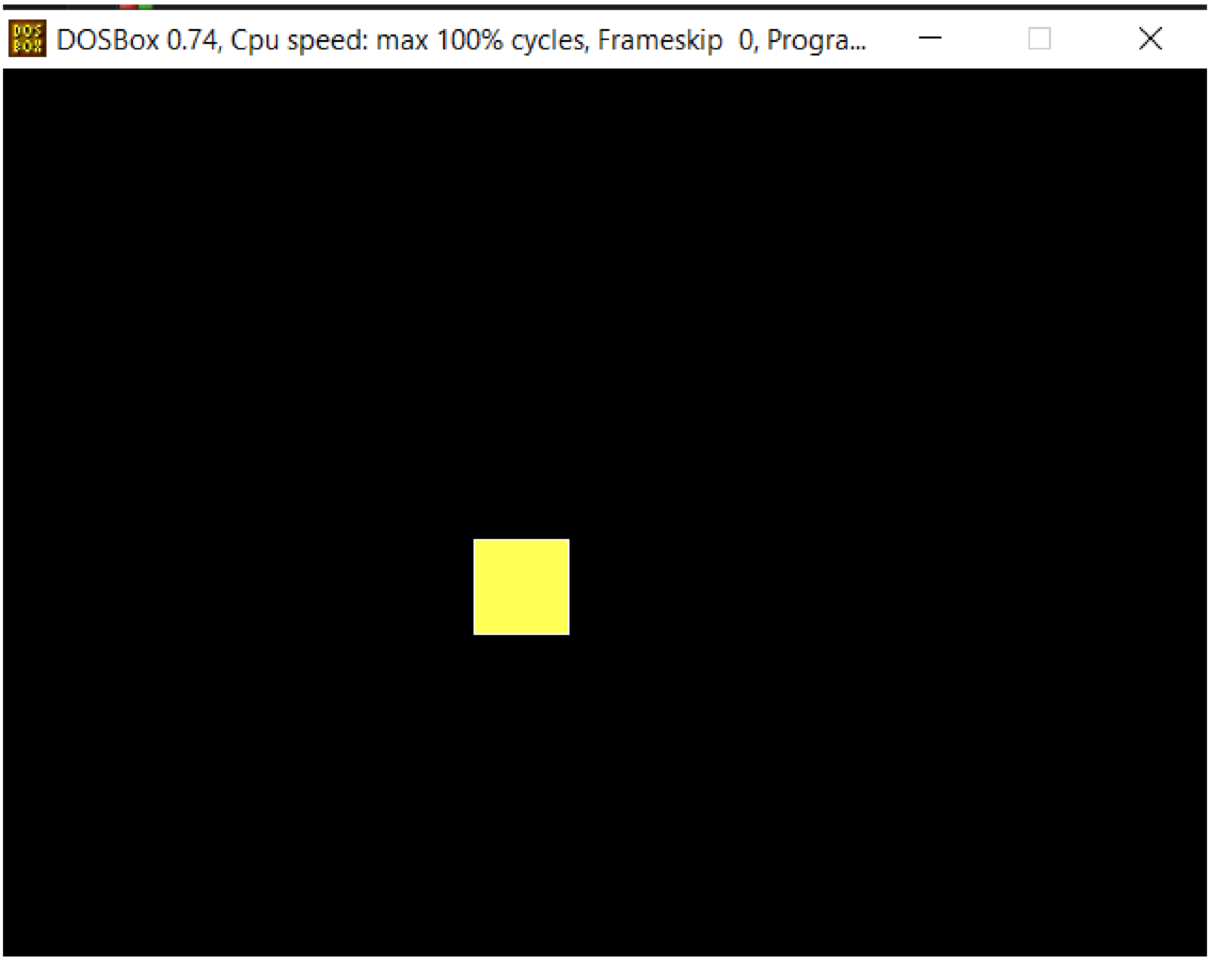
PROGRAM:

Code: Flood Fill

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

void fooldfill(int x, int y, int fill, int old){
        if(getpixel(x,y)==old){
                putpixel(x,y,fill);
                fooldfill(x+1,y,fill,old);
                fooldfill(x,y+1,fill,old);
                fooldfill(x-1,y,fill,old);
                fooldfill(x,y-1,fill,old);
        }
}
int main(){
        int gm, gd=DETECT;
        int x,y;
        initgraph(&gd,&gm,"c:\\turboc3\\bgi");
        rectangle(250,250,300,300);
        fooldfill(260,260,YELLOW,0);
        delay(5000);
        getch();
        closegraph();
        return 0;
}
```

OUTPUT:



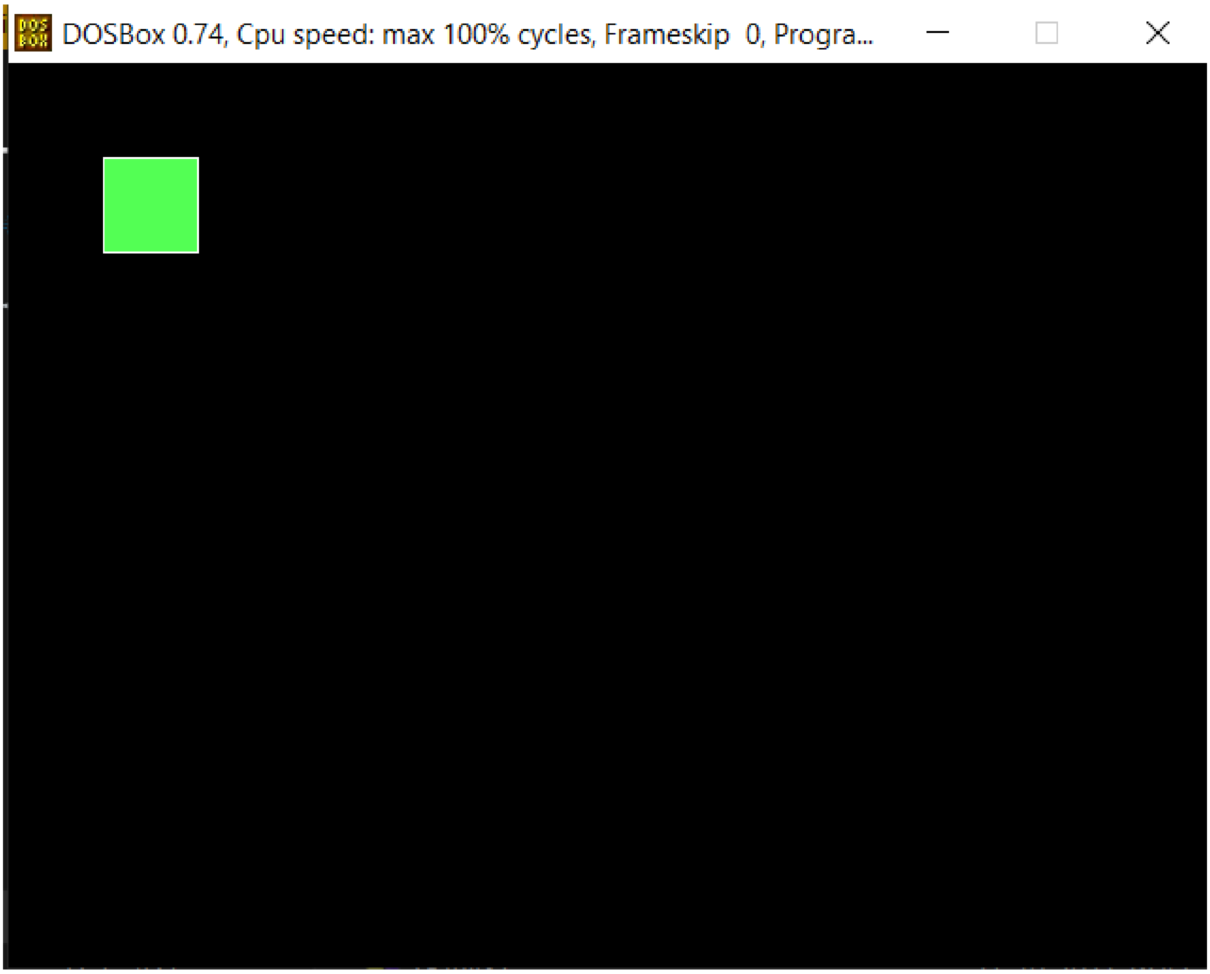
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

Code 2: Boundary Fill

```
#include<stdio.h>
#include<graphic.h>
void boundaryfill(int x, int y, int fill, int boundary){
        if(getpixel(x,y)!=boundary && getpixel(x,y)!=fill){
        putpixel(x,y,fill);
        boundaryfill(x+1,y,fill,boundary);
        boundaryfill(x-1,y,fill,boundary);
        boundaryfill(x,y+1,fill,boundary);
        boundaryfill(x,y-1,fill,boundary);
        boundaryfill(x+1,y+1,fill,boundary);
        boundaryfill(x+1,y-1,fill,boundary);
        boundaryfill(x-1,y+1,fill,boundary);
        boundaryfill(x-1,y-1,fill,boundary);
        }
}

int main(){
        int gm, gd=DETECT;
        int x,y;
        initgraph(&gd,&gm,"c:\\turboc3\\bgi");
        rectangle(100,100,50,50);
        boundaryfill(55,55,10,15);
        delay(5000);
        getch();
        closegraph();
        return 0;
}
```

OUTPUT:



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...