

Assignment - 4

Name : Emmadi Sumith Kumar

Branch : CSE - B1

Subject : Computer Graphics

Roll No : UI20CS21

Write a program to implement flood fill and boundary fill algorithms using switch cases.

```
#include <bits/stdc++.h>
#include <graphics.h>
using namespace std;
void boundary(int x, int y, int fillcolour, int boundrycolour)
{
    if (getpixel(x, y) != boundrycolour && getpixel(x, y) != fillcolour)
    {
        putpixel(x, y, fillcolour);
        boundary(x + 1, y, fillcolour, boundrycolour);
        boundary(x, y + 1, fillcolour, boundrycolour);
        boundary(x - 1, y, fillcolour, boundrycolour);
        boundary(x, y - 1, fillcolour, boundrycolour);
    }
}
void flood(int x, int y, int newcolour, int oldcolour)
{
    if (getpixel(x, y) == oldcolour)
    {
        putpixel(x, y, newcolour);
        flood(x + 1, y, newcolour, oldcolour);
        flood(x - 1, y, newcolour, oldcolour);
        flood(x, y + 1, newcolour, oldcolour);
        flood(x, y - 1, newcolour, oldcolour);
    }
}
int main()
{
    int gdriver = DETECT, gmode, errorcode;
    initgraph(&gdriver, &gmode, NULL);
    int ch, p = 1;
```

```

int X = getmaxx();
int Y = getmaxy();
int cx = X / 2;
int cy = Y / 2;
line(cx, 0, cx, Y);
line(0, cy, X, cy);

do
{
    rectangle(cx + 50, cy - 150, cx + 150, cy - 50);

    cout << "Choice :\n 1: Boundry Fillcolour \n 2: Flood Fillcolour\nEnter
your choice : ";
    cin >> ch;
    int x, y;
    x=100;
    y=100;
    switch (ch)
    {
        case 1:
            // cout << "x and y : ";
            // cin >> x >> y;
            x = cx + x;
            y = cy - y;
            boundary(x, y, 5, 15);
            delay(4000);
            break;
        case 2:
            // cout << "x and y : ";
            // cin >> x >> y;
            x = cx + x;
            y = cy - y;
            flood(x, y, 15, 0);

            delay(4000);
            break;
        case 3:
            p = 0;
            break;
    }
}

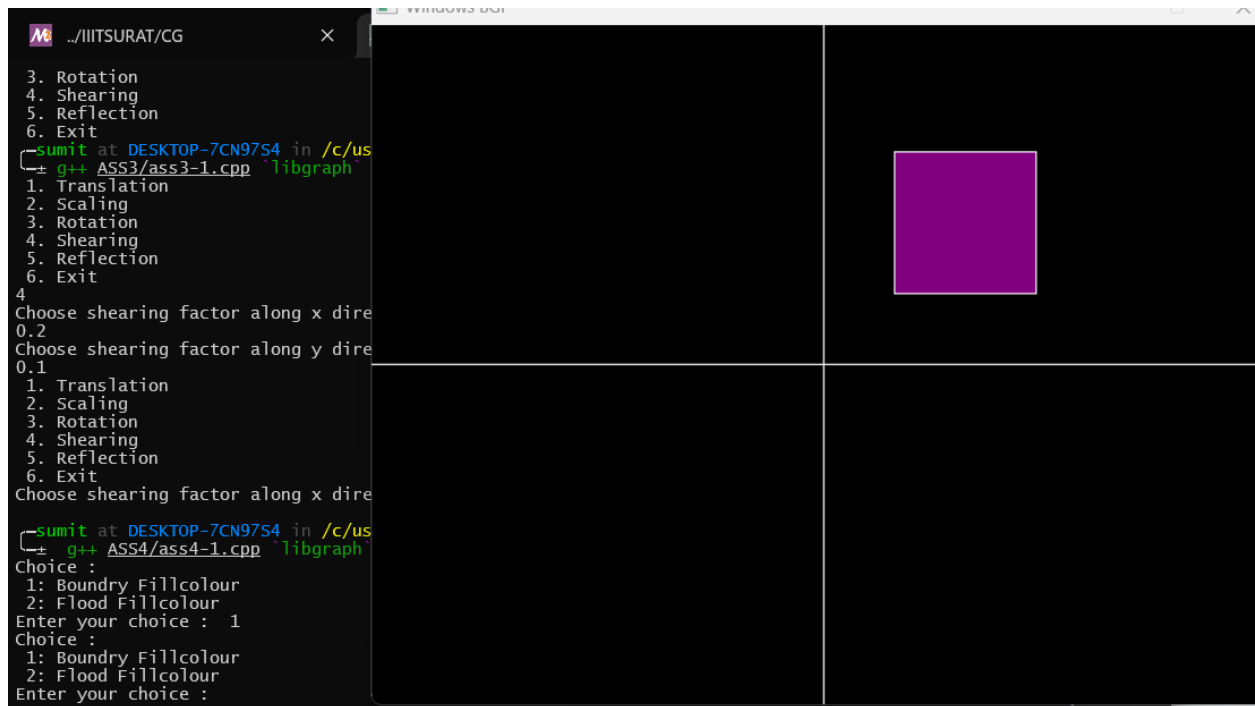
```

```

        default:
            cout << "Nothing";
            break;
    }
} while (p == 1);
getch();
// closegraph();
return 0;
}

```

1. boundary fill algorithm



2 .flood fill algorithm

```
Windows BGI
./IITSURAT/CG
5. Reflection
6. Exit
4
Choose shearing factor along
0.2
Choose shearing factor along
0.1
1. Translation
2. Scaling
3. Rotation
4. Shearing
5. Reflection
6. Exit
Choose shearing factor along
sumit at DESKTOP-7CN97S4
g++ ASS4/ass4-1.cpp
Choice :
1: Boundry Fillcolour
2: Flood Fillcolour
Enter your choice : 1
Choice :
1: Boundry Fillcolour
2: Flood Fillcolour
Enter your choice : 2
sumit at DESKTOP-7CN97S4
g++ ASS4/ass4-1.cpp
Choice :
1: Boundry Fillcolour
2: Flood Fillcolour
Enter your choice : 2
Choice :
1: Boundry Fillcolour
2: Flood Fillcolour
Enter your choice : 1
```

