

Assignment - 5 - B2

Problem - Write C++ program to draw a convex polygon and fill it with desired colour using seed fill algorithm. Use mouse interfacing to draw polygon.

Objective : To draw a polygon and fill it with desired colour with the help of mouse interfacing.

Outcome :

- ① Able to draw a polygon with mouse interfacing and desired colour.
- ② Able to fill polygon using seed fill algorithm with different colours.

* S/W and H/W requirements - Qt creator, Fedora OS.

Algorithms :

DDA algorithm

- ① Take inputs (x_1, y_1) & (x_2, y_2)
- ② $dx = (x_2 - x_1)$, $dy = (y_2 - y_1)$
- ③ if $(abs(dx) \geq abs(dy))$
 $len = abs(dx)$
 else
 $len = abs(dy)$
- ④ $x_{inc} = dx / len$.
- ⑤ $y_{inc} = dy / len$.

Seed-filling

```

(6) x = x1, y = y1
    i = 0
    while (i < len)
    {
        setpixel (x, y, color)
        x = x + xinc
        y = y + yinc
        i++
    }

```

Seq Mouse Interfacing algorithm

```

if (start)
{
    int p = ev->pos().x();
    int q = ev->pos().y();
    a[ver] = p;
    b[ver] = q;
    if (ev->button() == Qt::RightButton) {
        dda (a[ver], b[ver], a[0], b[0]);
        start = false;
    }
    else
    {
        if (ver > 0) {
            dda (a[ver], b[ver], a[ver-1], b[ver-1]);
        }
        ver++;
    }
}

```


{93} Seedfill algorithm

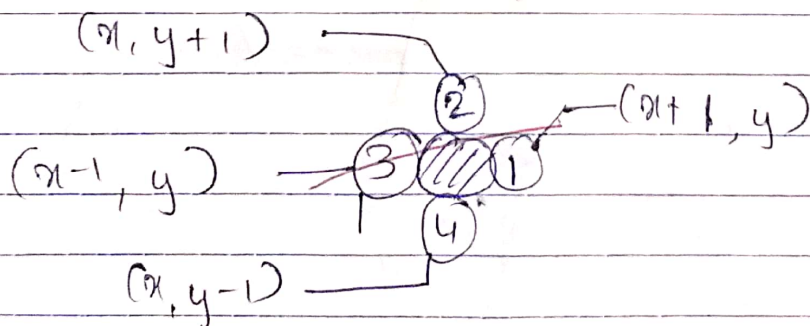
```

Q Rgb current = img.pixel(p, q);
if (bg == current)
{
    img.setPixel(p, q, color, rgb(0));
    seedfill(p+1, q, bg);
    seedfill(p-1, q, bg);
    seedfill(p, q+1, bg);
    seedfill(p, q-1, bg);
}

```

- * Advantages of mouse interfacing seed fill
- ① The colour of boundary and the colour to be filled can be same or different.
 - ② The colour of boundary can be different in seed fill algorithm.

Seed filling explanation

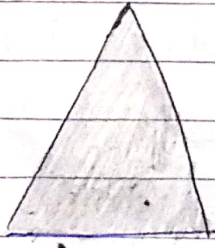
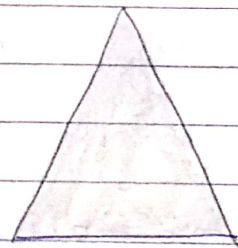
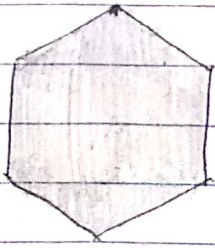
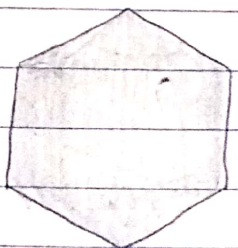


This algorithm is particularly useful when the region or polygon has no uniform colour boundaries.

This algorithm is also called as flood fill algorithm or forest fire algorithm because it spreads from a single point i.e. seed point in all the directions.

The limitations of boundary fill algorithm are overcome in seed fill algorithm. Like boundary fill algorithm this algorithm also begins with seed point which must be surely inside the polygon. Now, instead of checking the boundary colour this algorithm checks whether the pixel is having the polygon's original colour i.e. previous or old colour.

Test Cases

Test Case	Expected O/P	Actual O/P	Result
1) colour = red input given by mouse All fill colour = white			Pass
2) colour = green input given by mouse. fill colour = red.			Pass

Conclusion: Seed fill algorithm is implemented successfully with mouse interfacing and polygon drawing.