Assignment No. 9

Title :- chessboard (4x4) ~~3~~2-D transformation & filling.

problem statement :- Write C++ program to draw 4x4 chessboard rotated 45° with horizontal axis. Use Bresenham algorithm to draw all the lines. Use seed fill algorithm to fill black squares of the rotated chessboard.

Objective :- To learn & implement the transformation, filling in computer graphics.

s/w used :- Qt creator, CPP.

Theory :-

• Bresenham's line drawing :-

It uses the addition & substraction due to which it is faster than DDA. It is used in computer aided design, animation.

Algorithm :-

```
$\epsilon \leftarrow 0$    $y \leftarrow y_1$
for $x \leftarrow x_1$ to $x_2$ do
    plot point at $(x, y)$
    IF $(\epsilon + m < 0.5)$
        $\epsilon = \epsilon + m$
    else
        $y = y + 1$
        E ← E + m - 1
    ENDIF
ENDFOR
```

- Rotation at 45° (2-D transformation)

In rotation, we rotate the object at particular angle θ from origin. from the figure we can see that p(x,y) is located at angle θ from the horizontal X coordinate with distance E from origin.

$$\begin{vmatrix} x' \\ y' \\ 1 \end{vmatrix} = \begin{vmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} X \\ y \\ 1 \end{vmatrix}$$

Rotated Matrix   Rotation matrix   co-ordinate matrix

Pseudocode for multiplication:

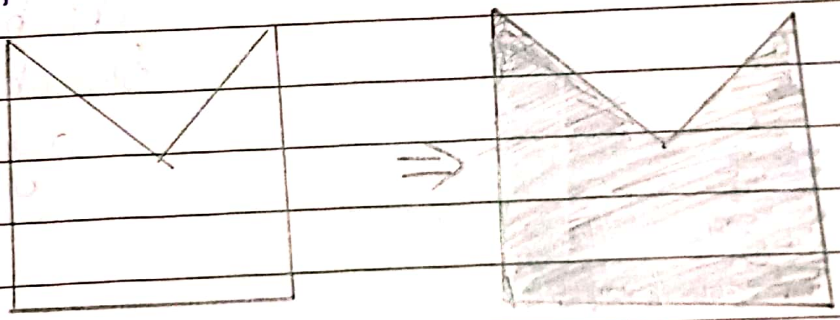
```
procedure MatrixMulti (CMatrix, TMatrix)
    input:  CMatrix , TMatrix
    output:  TransformedMatrix
  for (i=0 ; i<n; i++)
        for (j=0 ; j<n; j++)
              TransformedMatrix [i][j] = 0
                    for (k=0; k<n; k++)
```

```
Transformed Matrix [i][j] = Transformed Matrix [i][j] +
                                (Matrix [i][k] *
                                     F Matrix [k][j];

      end for
     end for
    end for
  end MultiMatrix
```

- Polygon filling :-

process of colouring area of polygon.



Filling the polygon means highlighting all the pixels which lies inside the polygon with any colour other than background.

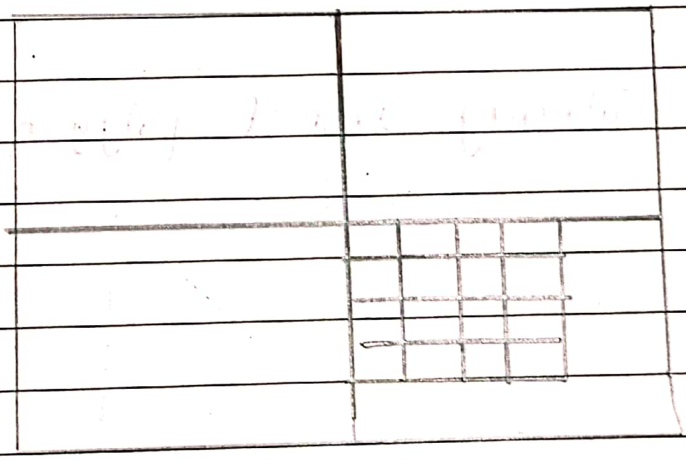
Pseudocode :-

```
Algorithm flood-fill (x, y, fillcolor, bkcolor)
    start
        if (get pixel (x,y) == bkcolor) then
```

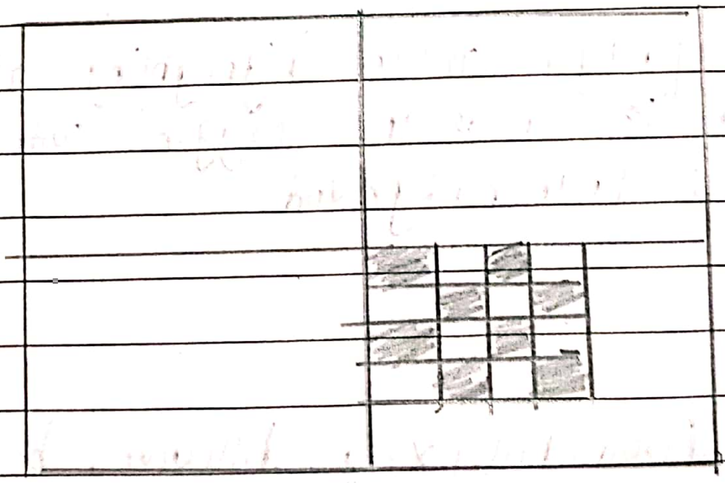
```
Begin
    Flood-fill (x+1, y, fillcolor, bkcolor)
    Flood-fill (x, y+1, fillcolor, bkcolor)
    Flood-fill (x-1, y, fillcolor, bkcolor)
    Flood-fill (x, y-1, fillcolor, bkcolor)
END

END
```

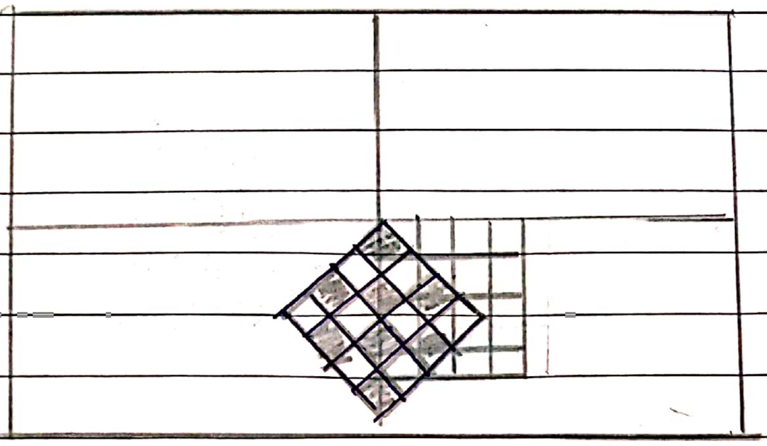
• Output:



Creating the chessboard with Bresenham's line drawing algorithm.



filling the chessboard with Flood-fill polygon filling algorithm.



Conclusion :-

we have learn & implement the Bresenham's line drawing algorithm, polygon flood-fill algorithm, 2-D Rotation. on chessboard.