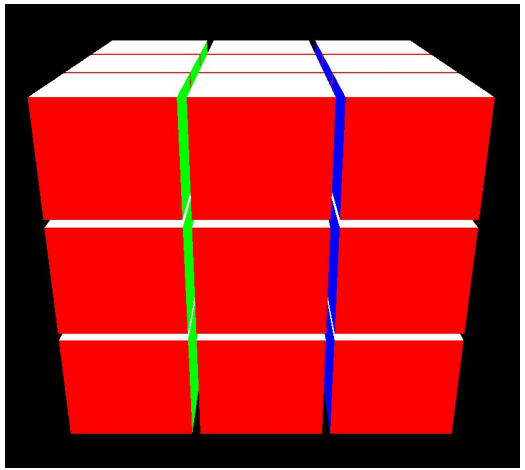
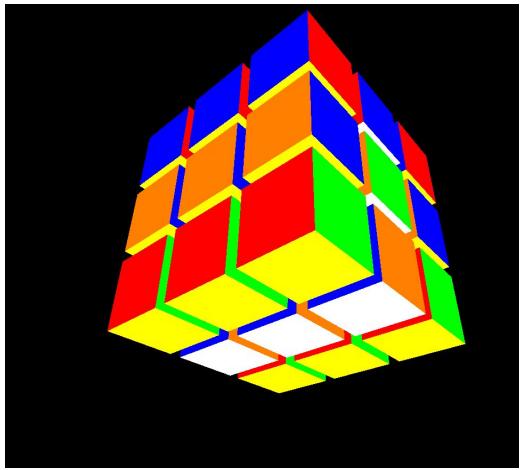
ICG Project
RUBIK'S CUBE

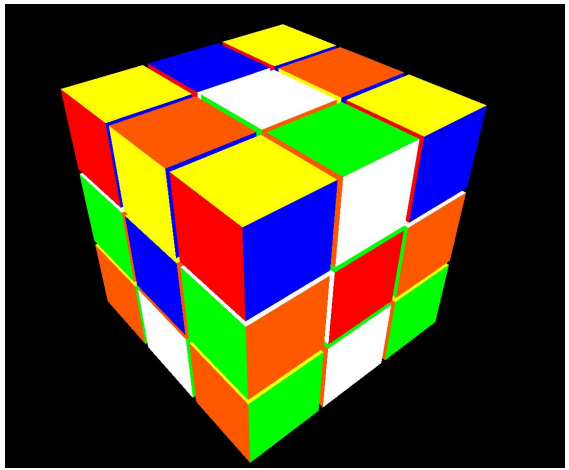
SHERIN K KARYIL
CED18I050

Idea

To simulate the working of a Rubik's Cube using opengl in C++







Controls

➔ **Cube Visualization (rotating the whole cube)**

L : rotate right

J : rotate left

I : rotate up

K : rotate down

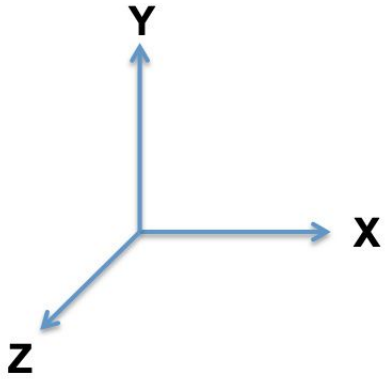
+ : increase space between cubes

- : decrease space between cubes

Controls

➔ **Selecting Layer for Rotation**

Along X-axis	Along Y-axis	Along Z-axis
Q: select 1st layer	A: select 1st layer	Z: select 1st layer
W: select 2nd layer	S: select 2nd layer	X: select 2nd layer
E: select 3rd layer	D: select 3rd layer	C: select 3rd layer



Controls

➔ **Layer Rotation**

O : Rotate Forward

U : Rotate Backward

Compilation and Run

```
g++  cube.cpp  -lglut -lGL -lGLU  -o CUBE

./CUBE
```

CUBE WORKING

1. Whole Cube Rotation

```
glRotatef(rot_x, 1.0, 0.0, 0.0);
glRotatef(rot_y, 0.0, 1.0, 0.0);
```

The above two commands are used to rotate the whole Cube around the x axis and the y axis respectively.
On pressing the keys I, J, K, L , the values of rot_x and rot_y are changed leading to the change in orientation of the whole Cube

2. Increment/Decrement space b/w individual cubes

```
glTranslatef((x - 1) * (cube_size+gap), (y - 1) * (cube_size +gap), (z - 1) * (cube_size+gap));
```

By changing the value of gap, the space in between individual cubes inside the Rubik's Cube can be modified. This is done using the + and - keys.

3. Layer Rotations

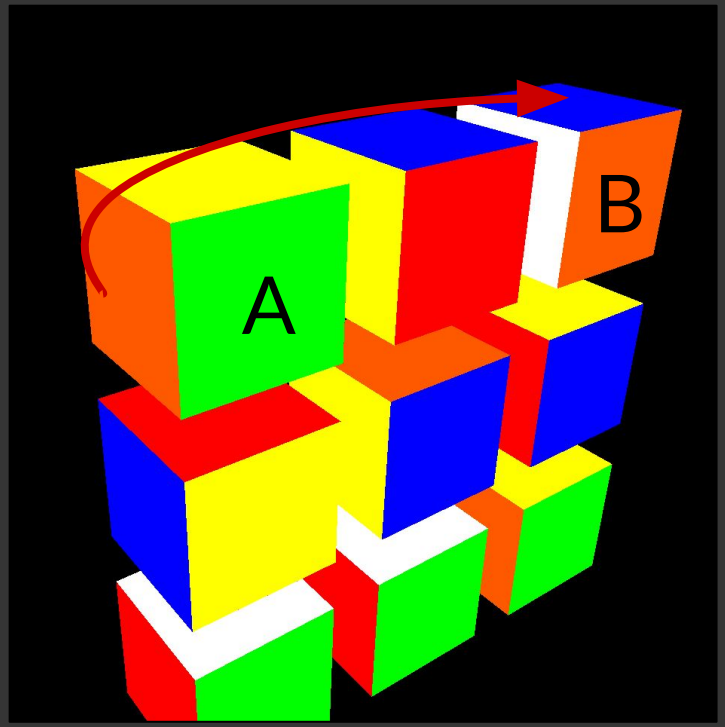
First we select the required layer using the corresponding key.

Then we apply rotations only to the individual cubes within that layer. To apply rotations specifically to individual cubes without affecting the others in the Rubik's Cube, we make use of glPushMatrix() and glPopMatrix()

```
glPushMatrix();
        //Apply translation
        //Apply Rotation
        //Draw individual cubes
glPopMatrix();
```

The above algorithm is used to rotate individual cubes of only the SELECTED LAYER

MAPPING OF CUBES BEFORE LAYER ROTATION



Consider any random layer as given in the image. Suppose we have to rotate the layer clockwise by 90 degree.
If we simply rotate the cube B by 90 degree clockwise, we would get WHITE colour at the top. But this is wrong as the correct colour should have been Orange .
So to avoid this, we need to first copy cube A to cube B's location, and then rotate it by 90 degrees clockwise.By doing so we get the correct colour Orange on top after the rotation.

Similar mapping is done for all cubes in the layer to be rotated. The mapping is different for clockwise and anti-clockwise rotations

FUNCTIONS USED

- glEnable(GL_DEPTH_TEST) : Enabled to do depth comparisons and update the depth buffer. Without it, it'll be like drawing on a 2D screen
- gluPerspective( ) : to specify the field of view in the x and y axes
- glLoadIdentity() : to reset the transformations applied on the current matrix
- gluLookAt() : to specify the observer and the reference positions
- glRotatef() and glTranslatef() : to rotate or translate the current matrix
- glPushMatrix() and glPopMatrix() : to ensure that the transformations done to any one cube inside the Rubik's cube doesn't affect the orientation of the other cubes.
- glBegin(GL_QUADS) : to draw the faces of a cube
- glutPostRedisplay() : In the next iteration through glutMainLoop, the window's display callback will be called. This is done so that the updated transformations gets reflected on the screen

Thank You

CED18I050
SHERIN K KARYIL