Assignment No- 11

Title :- 3-D transformation

problem statement :- Write C++ program for draw 3-D cube & perform following transformation on OpenGL.
- Scaling
- Translation
- Rotation about axis.

objective :- To learn & implement 3-D transformation on openGL.

Outcome :- Student will able to understand different types of transformation on OpenGL.

S/w / h/w :- Qt creatore, OpenGL.

Theory :-

OpenGL :-

Open graphics library is specification defining a cross language cross platform API foR the Creating application that in 3-D computeR graphics. The interface consist of oveR 250 different function call. we use it foR drawing complex 3-D scenes foR simple programming.

3-D transformation :-

It means chaging some graphics into something else by applying rules of 3-D objects.

like 2-D transformation it also have three - four types

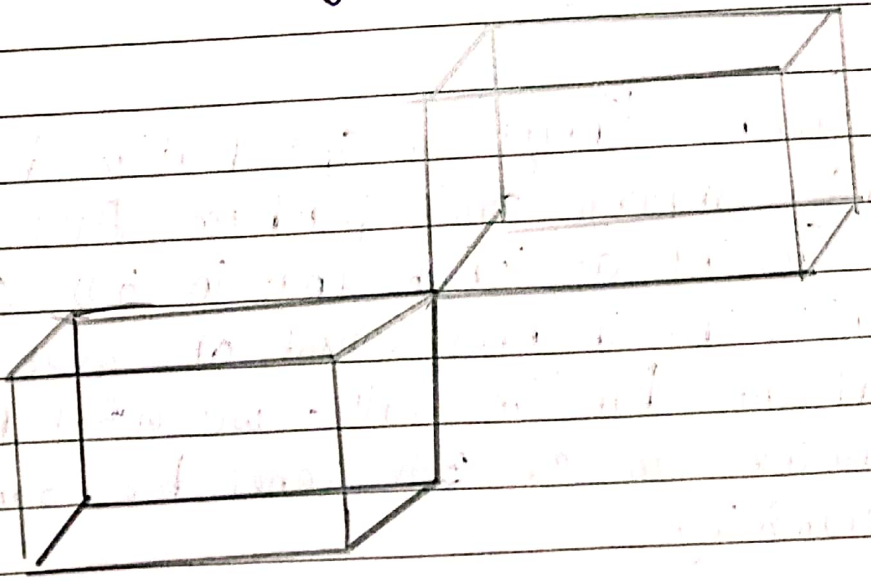
i) translation
ii) scaling
iii) Rotation.
iv) Reflection

transformation take place on 3-D plane called 3-D tranformation.

- Translation :-

The translation can also lie inteepret as add in constant vectoe at any point.

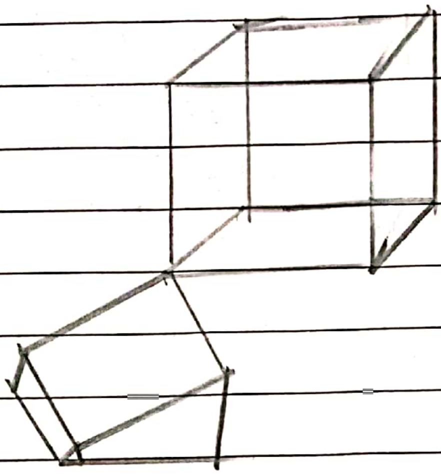
means moving object to different position.



Scanned with CamScanner

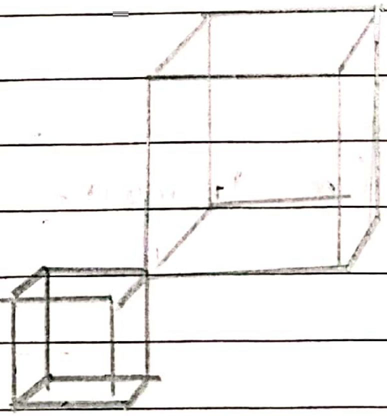
- Rotation :-

It is process of rotate the 3-D object at particular angle. It can be clockwise or anti-clockwise.



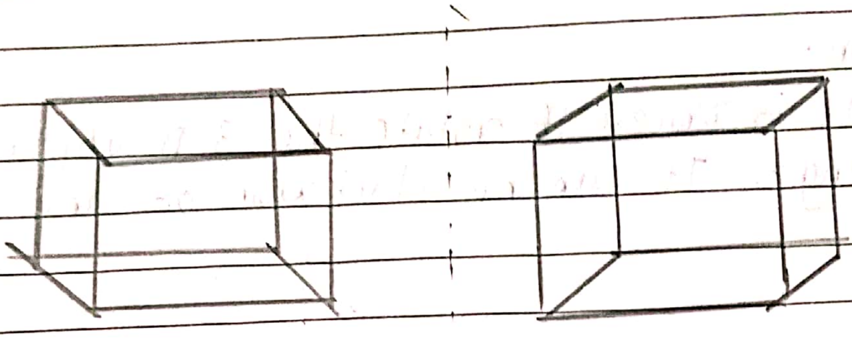
- Scaling :-

It is refering to altering the size of object for appearing farther or nearer from view.



- Reflection :-

It is used to emulate the reflective object like mirrors & surfaces.



Algorithm :-

```
1. Translate (tx, ty, tz)
      {
      matrix4x4  m
      matrix set identity (m);
        m[0][3] = tx;
         m[1][3] = ty;
          m[2][3] = tz;
       matrixmultiply (m, thematrise)

     }
```

```
2. Scale ( sx, sy, sz)
       {

       matrix4x4  m;
      matrix setidentity (m);
        m[0][0] = sx;
        m[0][3] = (1-sx);
        m[1][1] = sy;
m[1][3]= (1-sy)
 m[2][2]=  sz
  m[2][3]=  1-sz
 matrix multiply (m, thematrix)
}

3. Rotate (float angle)
   {
   matiix4x4  m
   matrixSetidentity (m);
    angle= angle  *  22/ 1266;
      m[1][1]  = cos angle
      m[1][2]  = -sin angle
      m[2][1]  =  sin angle
       m[2][2]  = cos angle

   }
```

Conclusion::

we have learn & implement 3-D tranformation On OpenGL.