

# Modelling/Geometric Transformations

Rajat Khanduja  
09010137

Bharat Khatri  
09010164

# Contents

<b>1</b>	<b>Modelling Transformations</b>	<b>2</b>
1.1	OpenGL examples . . . . .	2
<b>2</b>	<b>Translation</b>	<b>4</b>
2.1	Introduction . . . . .	4
2.2	Translation in OpenGL . . . . .	4
<b>3</b>	<b>Scaling</b>	<b>5</b>
<b>4</b>	<b>2D Rotation</b>	<b>6</b>
<b>5</b>	<b>Reflection</b>	<b>7</b>
<b>6</b>	<b>Shear</b>	<b>8</b>

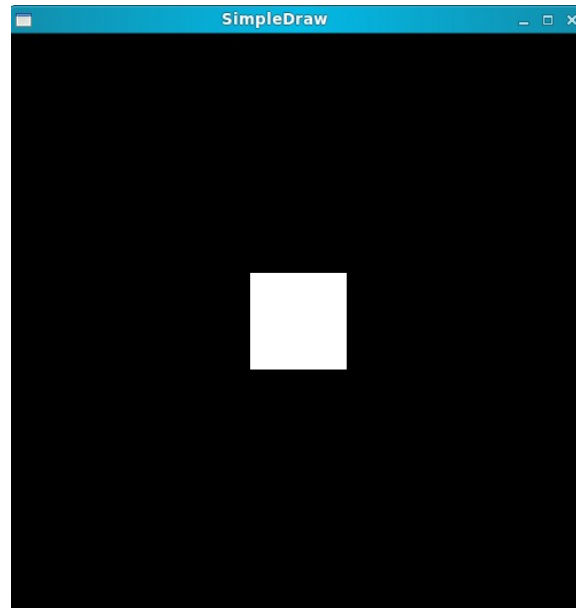


Figure 1: A square drawn using OpenGL.

## 1 Modelling Transformations

Changes in orientation, size and shape are accomplished with Geometric transformations that alter the co-ordinate description of objects. The basic geometric transformations are :-

- Translation
- Scaling
- Rotation
- Reflection
- Shear

### 1.1 OpenGL examples

In the text that follows, we use some OpenGL examples to illustrate how the transformations are performed using OpenGL functions. All transformations are being applied to a square of edge length 1 centred at  $(0,0)$ , unless stated otherwise.

Following is the part of the code to construct the square in discussion.

```
glPushMatrix ();  
glBegin (GL_POLYGON);  
glVertex2f (-0.5, -0.5);  
glVertex2f (-0.5, 0.5 );  
glVertex2f ( 0.5, 0.5 );  
glVertex2f ( 0.5, -0.5);  
glEnd ();  
glPopMatrix ();
```

In the discussions that follow, only the transformation code is provided which has to be used before plotting the vertices (*glVertex2f*) but after pushing the matrix using *glPushMatrix*

## 2 Translation

### 2.1 Introduction

A *translation* is applied to an object by repositioning it along a straight line path from one coordinate location to another. Mathematically, translation is achieved by adding the translation distance,  $t_x$  and  $t_y$ , to the original coordinate position  $(x, y)$  to move to the new position  $(x', y')$ .

$$x' = x + t_x$$

$$y' = y + t_y$$

Translation can be achieved by the following transformation :-

For 2D translation :-

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For 3D translation :-

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

### 2.2 Translation in OpenGL

Translation can be achieved in OpenGL using the function *glTranslate*.

The same could be achieved using the transformation matrix described in (1).

```
// Column major matrix.
GLdouble t_x = 1;
GLdouble t_y = 1;
GLdouble t_z = 0;
GLdouble translation_matrix[] = { 1, 0, 0, 0,
                                   0, 1, 0, 0,
                                   0, 0, 1, 0,
                                   t_x, t_y, t_z, 1};

glMultMatrix (translation_matrix);
```

The effect of the same can be seen in Figure 2.2.

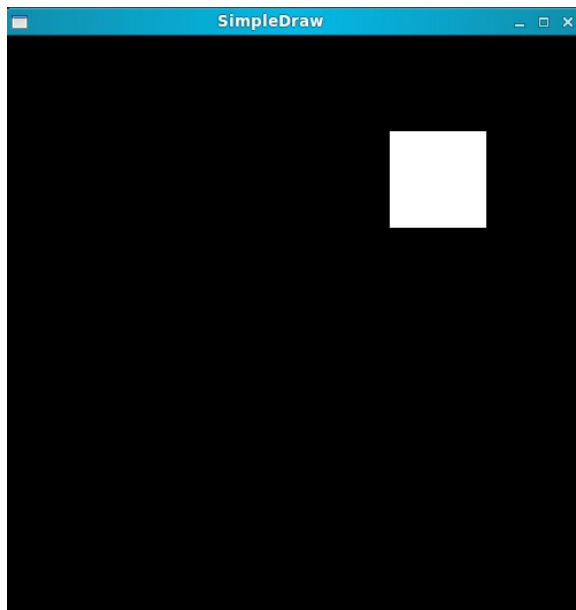


Figure 2: Translated square

### 3 Scaling

*Scaling* a coordinate means multiplying each of its components by a scalar.

$$\begin{bmatrix} c & 0 & 0 & 0 \\ 0 & c & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## 4 2D Rotation

A *2D Rotation* is applied to an object by repositioning it along a circular path in any of the three x-y, y-z or z-x planes.

## 5 Reflection

A *reflection* is a transformation that produces the mirror image of an object relative to an axis of reflection by rotating the object  $180^\circ$  about the reflection axis.



## 6 Shear

A *transformation* that distorts the shape of an object such that the transformed shape appears as if the object were composed of internal layers that had been caused to slide over each other.