Basic OpenGL Transformations

- glTranslate* (tx, ty, tz);
  - glTranslatef (25.0, -10.0, 0.0);
- glRotate* (theta, vx, vy, vz)
  - glRotatef (90.0, 1.0, 0.0, 0.0);
  - Vector v = (vx, vy, vz) defines the orientation for a rotation axis that passes through the coordinate origin
- glScale* (sx, sy, sz);
  - glScalef (2.0, -3.0, 1.0);

1

OpenGL Matrix Operation

- glMatrixMode (GL_MODELVIEW);
  - Modelview matrix is used to store and combine geometric transformations
- glLoadIdentity ();

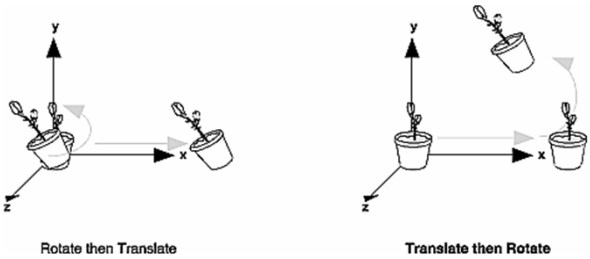
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- OpenGL maintains a matrix stack for each matrix mode
- You may want to save the composite matrix to create multiple views
- glPushMatrix (); copies the current matrix at the top of stack
- glPopMatrix (); destroys the matrix at the top stack, and the second matrix becomes the current matrix

2

Order of Transformations

- Each successive transformation command multiplies a new matrix **M** by the current matrix **C** to yield **CM**.
- Finally, vertices *v* are multiplied by the current matrix.
- This process means that the last transformation command called in your program is actually the first one applied to the vertices: **CMv**.
- Thus, one way of looking at it is to say that you have to specify the matrices in the reverse order



Rotate then Translate Translate then Rotate

OpenGL Example

```
glMatrixMode (GL_MODELVIEW);
glLoadIdentity ();
```

current matrix

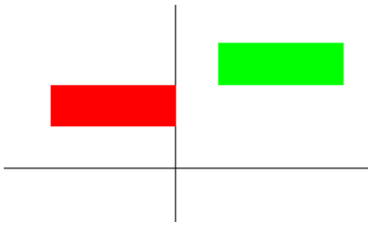
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
glColor3f (0.0, 1.0, 0.0);
glRecti (50, 100, 200, 150);
```

```
glColor3f (1.0, 0.0, 0.0);
glTranslatef (-200.0, -50.0, 0.0);
glRecti (50, 100, 200, 150);
```

current matrix

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -200 \\ 0 & 1 & -50 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -200 \\ 0 & 1 & -50 \\ 0 & 0 & 1 \end{bmatrix}$$



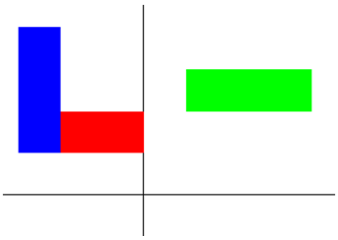
OpenGL Example (cont'd)

glColor3f (0.0, 0.0, 1.0);
glLoadIdentity ();

current matrix $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

glRotatef (90.0, 0.0, 0.0, 1.0);
glRecti (50, 100, 200, 150);

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} cos90 & -sin90 & 0 \\ sin90 & cos90 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



5

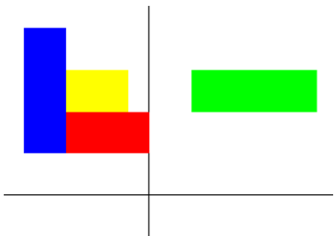
OpenGL Example (cont'd)

glColor3f (1.0, 1.0, 0.0);
glLoadIdentity ();

current matrix $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

glScalef (-0.5, 1.0, 1.0);
glRecti (50, 100, 200, 150);

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} -0.5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



6

OpenGL Example

glMatrixMode (GL_MODELVIEW);
glLoadIdentity ();

current matrix → top ←

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

glColor3f (0.0, 1.0, 0.0);
glRecti (50, 100, 200, 150);

current matrix → top ←

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

glPushMatrix ();

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

glColor3f (1.0, 0.0, 0.0);
glTranslatef (-200.0, -50.0, 0.0);
glRecti (50, 100, 200, 150);
glColor3f (0.0, 0.0, 1.0);

current matrix → top ←

$$\begin{bmatrix}1 & 0 & -200\\ 0 & 1 & -50\\ 0 & 0 & 1\end{bmatrix}$$

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

glPopMatrix ();

current matrix → top ←

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

7

glPushMatrix ();

current matrix → top ←

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

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

glRotatef (90.0, 0.0, 0.0, 1.0);
glRecti (50, 100, 200, 150);
glColor3f (1.0, 1.0, 0.0);

current matrix → top ←

$$\begin{bmatrix}cos90 & -sin90 & 0\\ sin90 & cos90 & 0\\ 0 & 0 & 1\end{bmatrix}$$

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

glPopMatrix ();

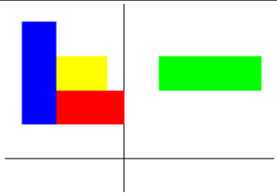
top ←

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

glScalef (-0.5, 1.0, 1.0);
glRecti (50, 100, 200, 150);

current matrix → top ←

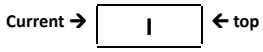
$$\begin{bmatrix}-0.5 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$



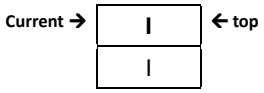
8

OpenGL Example – 3 Cubes

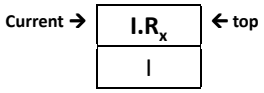
DrawAxes ();



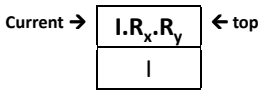
glPushMatrix ();



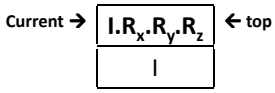
glRotatef ((float) nXAngle, 1.0, 0.0, 0.0);



glRotatef ((float) nYAngle, 0.0, 1.0, 0.0);



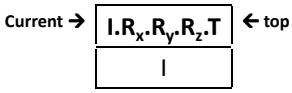
glRotatef ((float) nZAngle, 0.0, 0.0, 1.0);



DrawCube (4);

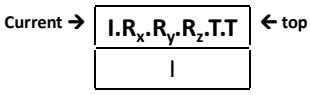
OpenGL Example – 3 Cubes (cont'd)

glTranslatef (100, 0, 0);



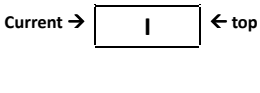
DrawCube (3);

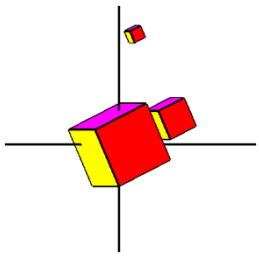
glTranslatef (0, 150, 0);



DrawCube (2);

glPopMatrix ();





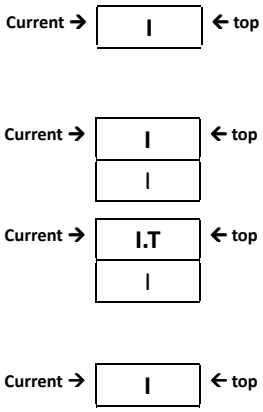
OpenGL Example – 36 Snowmen

```
for(int i = -3; i < 3; i++)
   for(int j=-3; j < 3; j++) {

         glPushMatrix();

         glTranslatef (i * 10.0, 0, j * 10.0);
         drawSnowMan ();

         glPopMatrix();
   }
```



OpenGL Example – 36 Snowmen (cont'd)

```
void drawSnowMan()
{
glColor3f(1.0f, 1.0f, 1.0f);

// Draw Body
glTranslatef(0.0f ,0.75f, 0.0f);
glutSolidSphere(0.75f,20,20);

// Draw Head
glTranslatef(0.0f, 1.0f, 0.0f);
glutSolidSphere(0.25f,20,20);

// Draw Eyes
glPushMatrix();
glColor3f(0.0f,0.0f,0.0f);
glTranslatef(0.05f, 0.10f, 0.18f);
glutSolidSphere(0.05f,10,10);
glTranslatef(-0.1f, 0.0f, 0.0f);
glutSolidSphere(0.05f,10,10);
glPopMatrix();

// Draw Nose
glColor3f(1.0f, 0.5f , 0.5f);
glutSolidCone(0.08f,0.5f,10,2);
}
```