

Lab 03: GL Transformations (Chapter 7,9)

I. GL is only 3D. To fake 2D do transforms with a fixed z component.

II. ~~GL Translation~~ Basic Geometric Transformations

A. ~~glTranslate~~ Translation - $glTranslatef(tx, ty, tz)$
 $glTranslated(tx, ty, tz)$

B. Rotation - $glRotatef(theta, vx, vy, vz)$ - angle-axis formulation. Theta in degrees not radians

C. Scale - $glScalef(sx, sy, sz)$

D. GL stores composite transforms so the following:

```
glTranslatef(5, 6, 7);  
glRotatef(90, 0, 0, 1);  
glScalef(2, 3, 4);  
// Draw
```

note applied in opposite order as code

is stored as $M = TRS$

~~affect~~

* Ask downsides/effect

- need to be able to reverse operations

III. GL matrix operations

A. Matrix mode - There are 4 different matrices given GL stores

GL_MODELVIEW - world coordinates + camera

GL_PROJECTION - projection transformations

GL_COLOR - color modifications } not as useful

GL_TEXTURE - texture modifications

$glMatrixMode(*)$ - set current matrix

B. Reset transform - $glLoadIdentity()$

C. Set/get matrix - avoid using (unless you know better, eg. need shear)

~~$glLoadMatrixf()$~~ - pass in 16 element array pointer

$glLoadMatrixd(*)$

$glGetv(*)$, eg. $glGetFloatv(GL_MODELVIEW_MATRIX, matrix)$ to get 4x4 matrix as 16 element array

D. Manually multiply matrix - avoid using (unless you know better, eg. shear)

$glMultMatrixd(*)$ - pass in 16 element array pointer

E. Each matrix is also in a stack to allow easy reversal of computations

$glPushMatrix()$

$glPopMatrix()$

so for each model:

```
glPushMatrix()
```

```
// Model transforms specific
```

```
// Draw model
```

```
glPopMatrix()
```