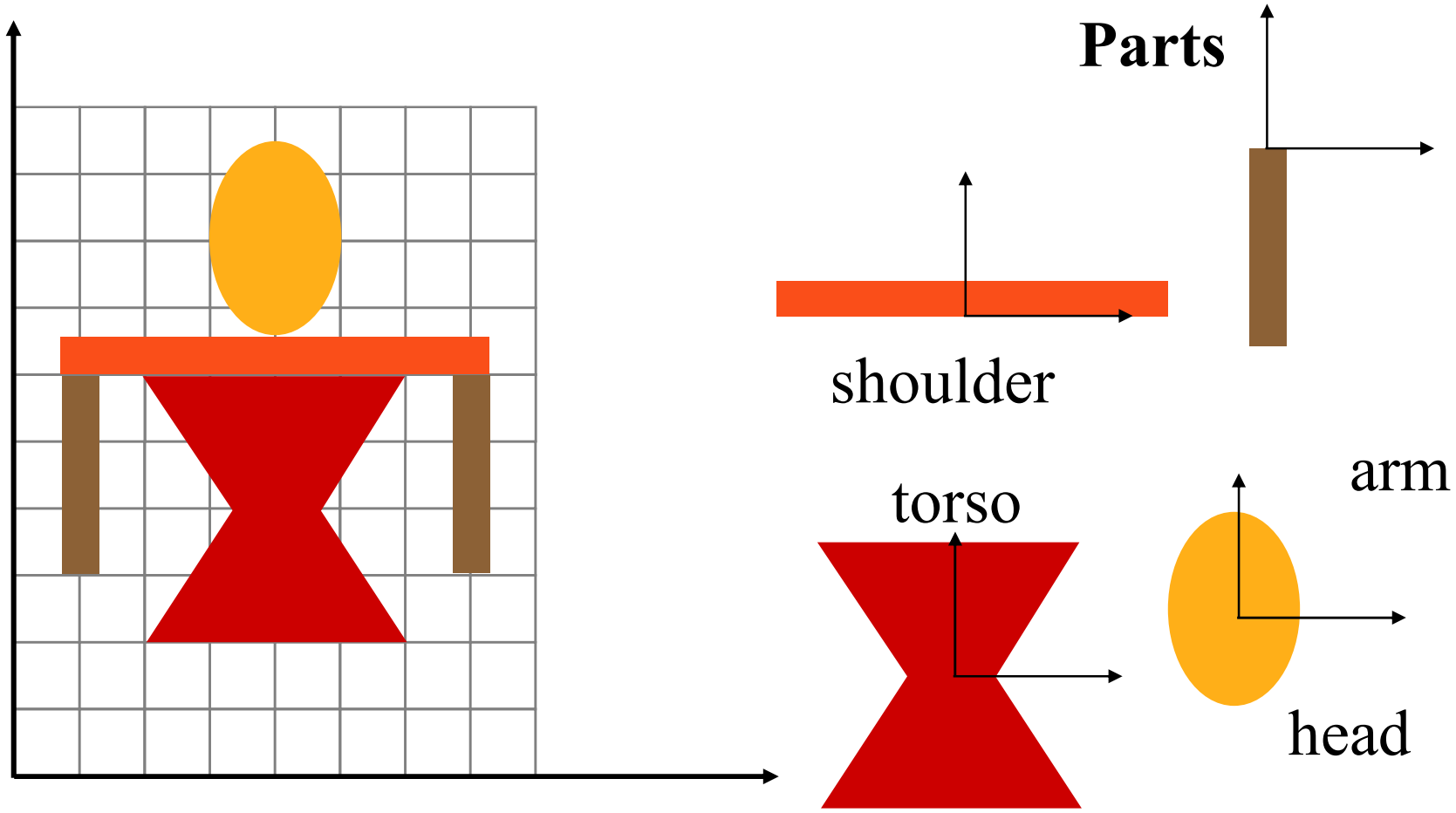
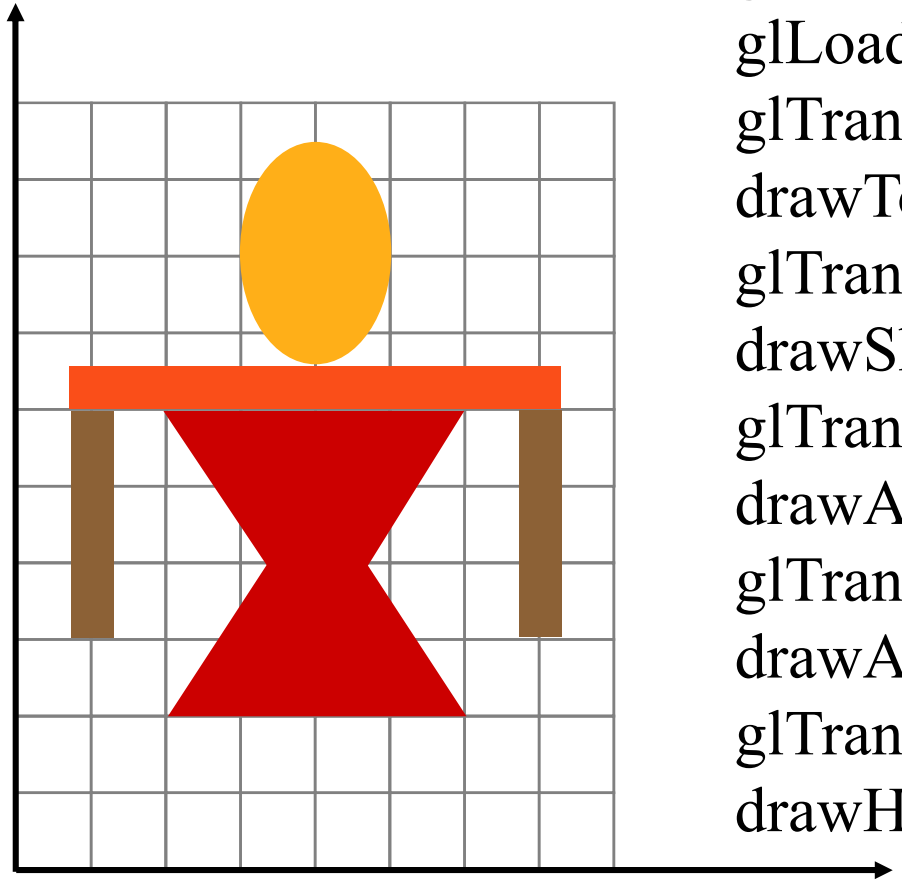
Goal: Draw figure using component parts.



Brute Force - Don’t Do This!

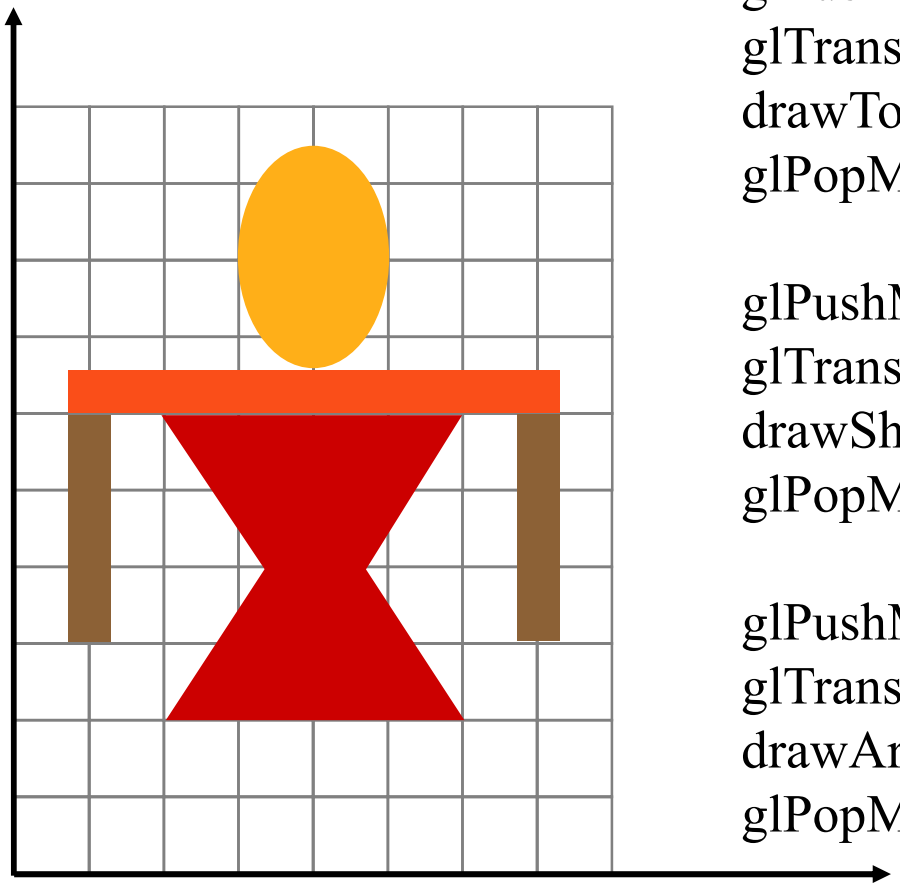
Dependent sequence



```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glTranslate(4,4); *
drawTorso();
glTranslate(0,2);
drawShoulder();
glTranslate(3,0);
drawArm();
glTranslate(-6,0);
drawArm()
glTranslate(3,2);
drawHead();
```

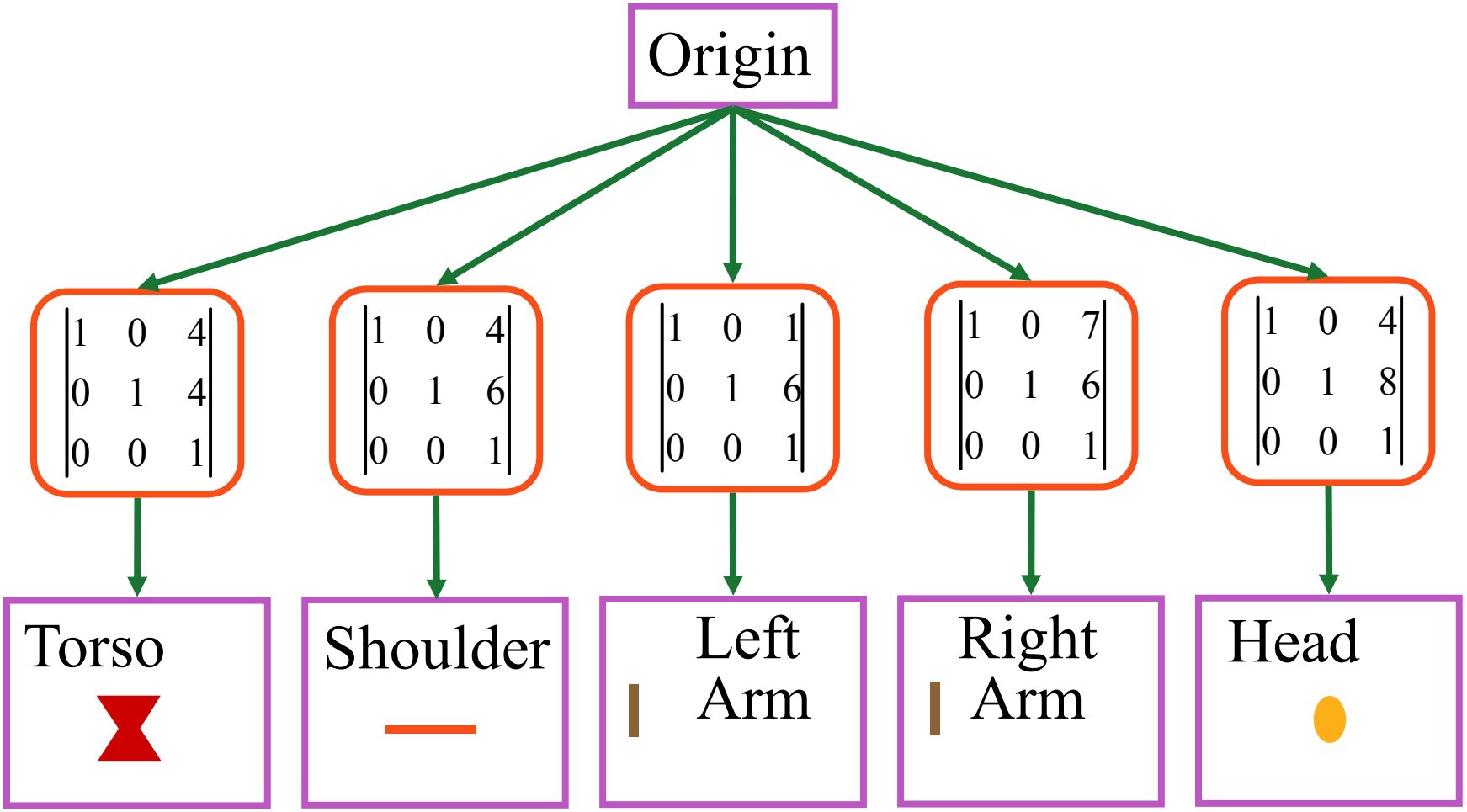
* glTranslate with 2 integer arguments is shorthand.

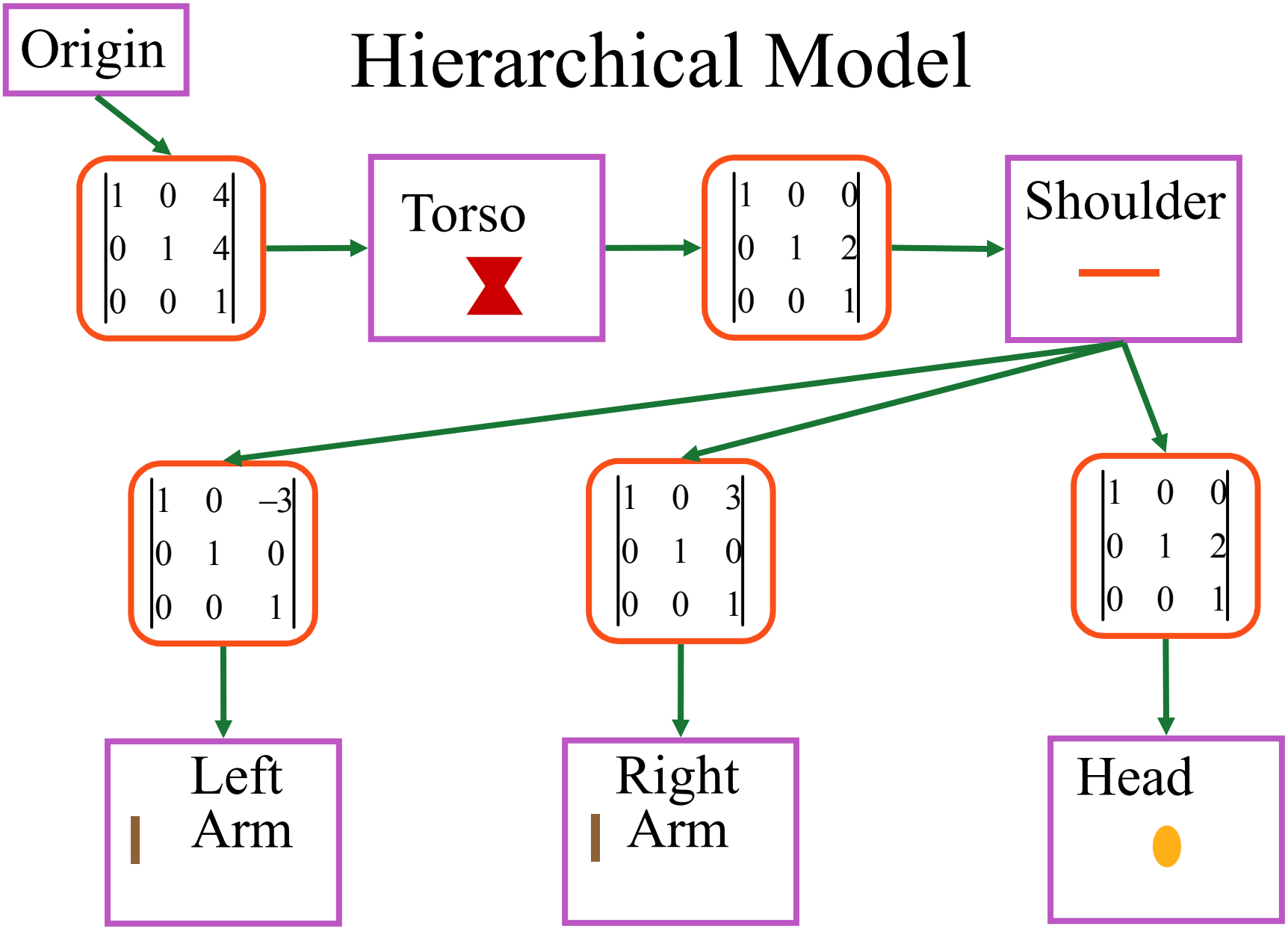
A Bit Better - Parts Independent



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

glPushMatrix();
glTranslate(4,4);
drawTorso();
glPopMatrix();

glPushMatrix();
glTranslate(4,6);
drawShoulder();
glPopMatrix();

glPushMatrix();
glTranslate(1,6);
drawArm();
glPopMatrix();

glPushMatrix();
glTranslate(7,6);
drawArm();
glPopMatrix();

glPushMatrix();
glTranslate(4,8);
drawHead();
glPopMatrix();
```

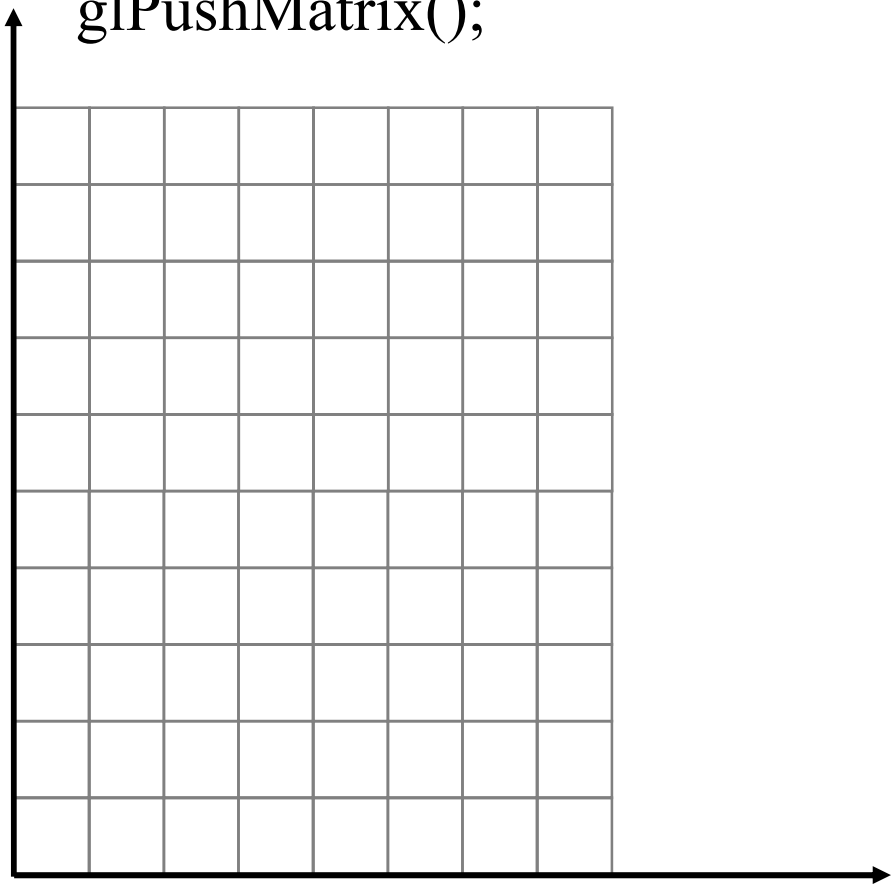
Think in terms of a Graph




Origin
Hierarchical Model
1 0 4
0 1 4
0 0 1
Torso
1 0 0
0 1 2
0 0 1
Shoulder
1 0 –3
0 1 0
0 0 1
1 0 3
0 1 0
0 0 1
1 0 0
0 1 2
0 0 1
Left Arm
Right Arm
Head

Initialization

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



Stack

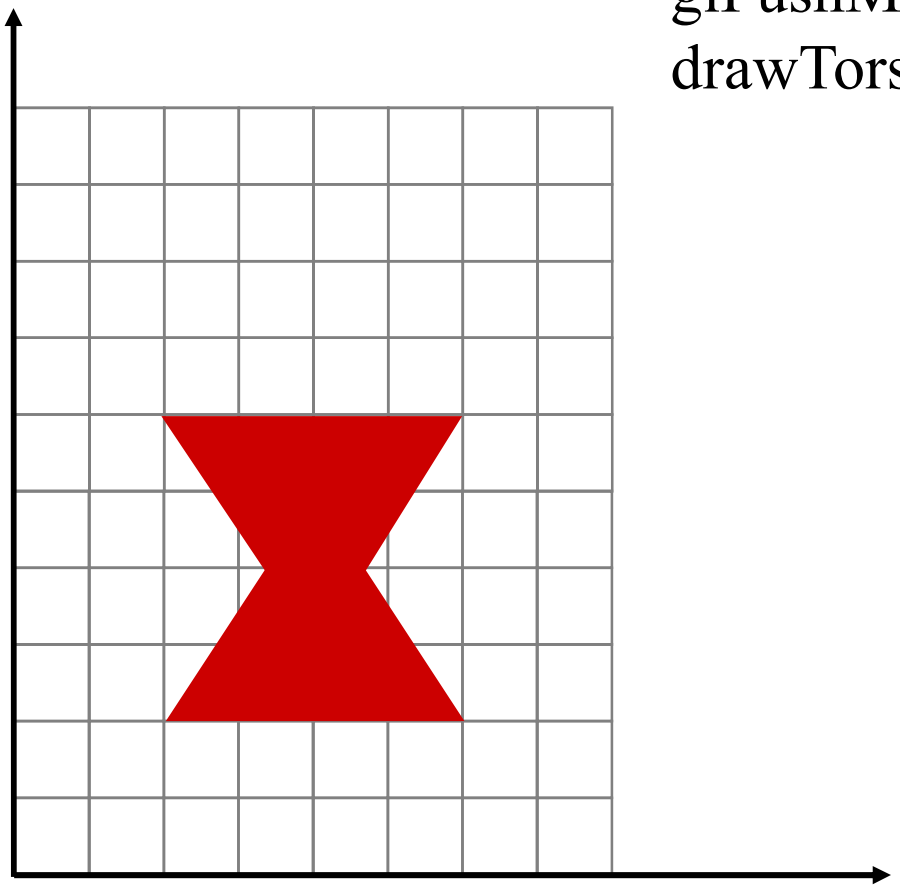
$$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

MODELVIEW Transformation

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

Torso

```
gltranslate(4,4);
glPushMatrix();
drawTorso();
```



Stack

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

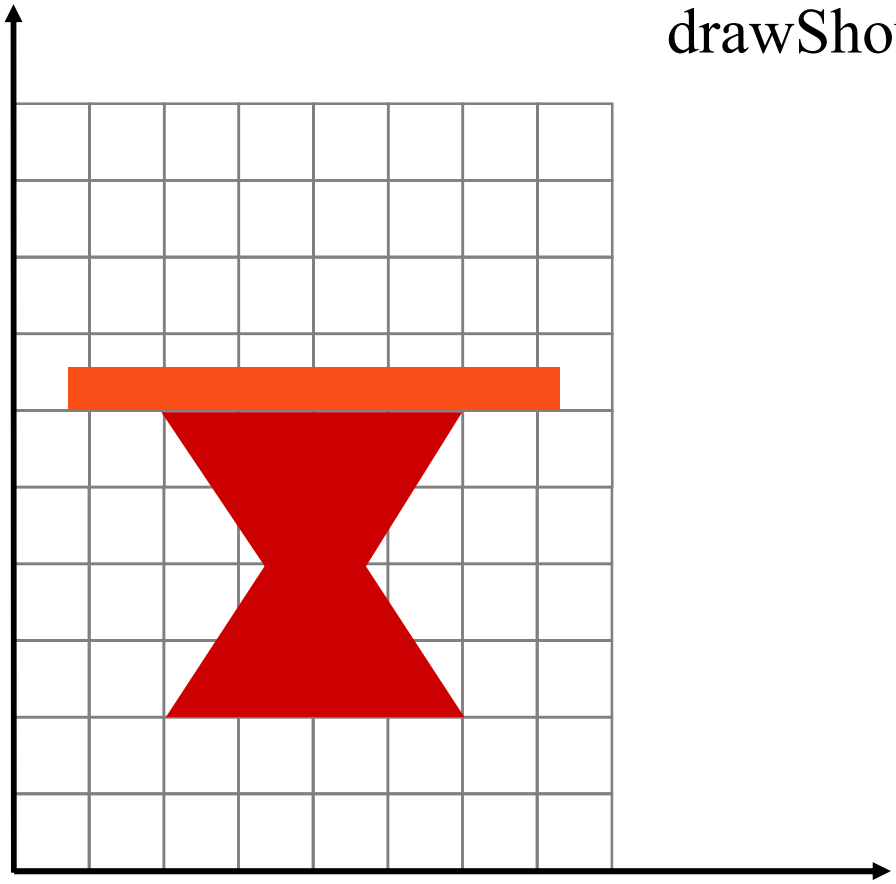
$$\begin{vmatrix} 1 & 0 & 4 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{vmatrix}$$

MODELVIEW Transformation

$$\begin{vmatrix} 1 & 0 & 4 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{vmatrix}$$

Shoulders

```
gltranslate(0,2);
glPushMatrix();
drawShoulders();
```



Stack

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

$$\begin{vmatrix} 1 & 0 & 4 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{vmatrix}$$

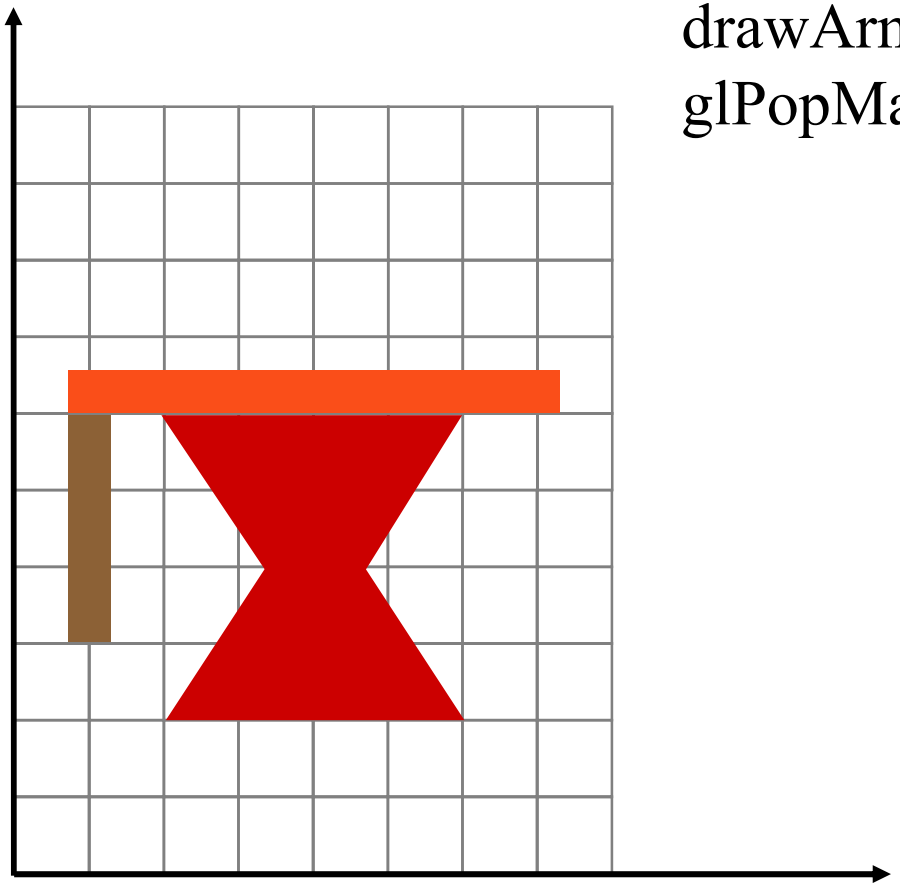
$$\begin{vmatrix} 1 & 0 & 4 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{vmatrix}$$

MODELVIEW Transformation

$$\begin{vmatrix} 1 & 0 & 4 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{vmatrix}$$

Left Arm

```
glPushMatrix();
gltranslate(-3,0);
drawArm();
glPopMatrix();
```



Stack

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

$$\begin{vmatrix} 1 & 0 & 4 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{vmatrix}$$

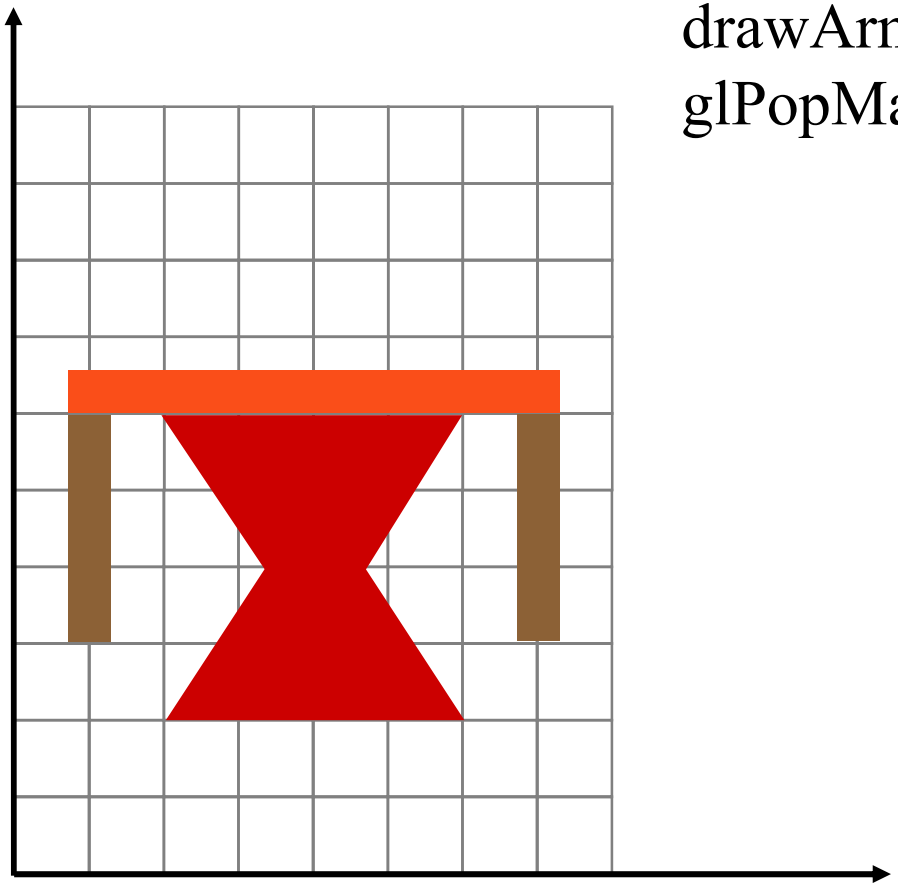
$$\begin{vmatrix} 1 & 0 & 4 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{vmatrix}$$

MODELVIEW Transformation

$$\begin{vmatrix} 1 & 0 & 1 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{vmatrix}$$

Right Arm

```
glPushMatrix();
gltranslate(3,0);
drawArm();
glPopMatrix();
```



Stack

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

$$\begin{vmatrix} 1 & 0 & 4 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{vmatrix}$$

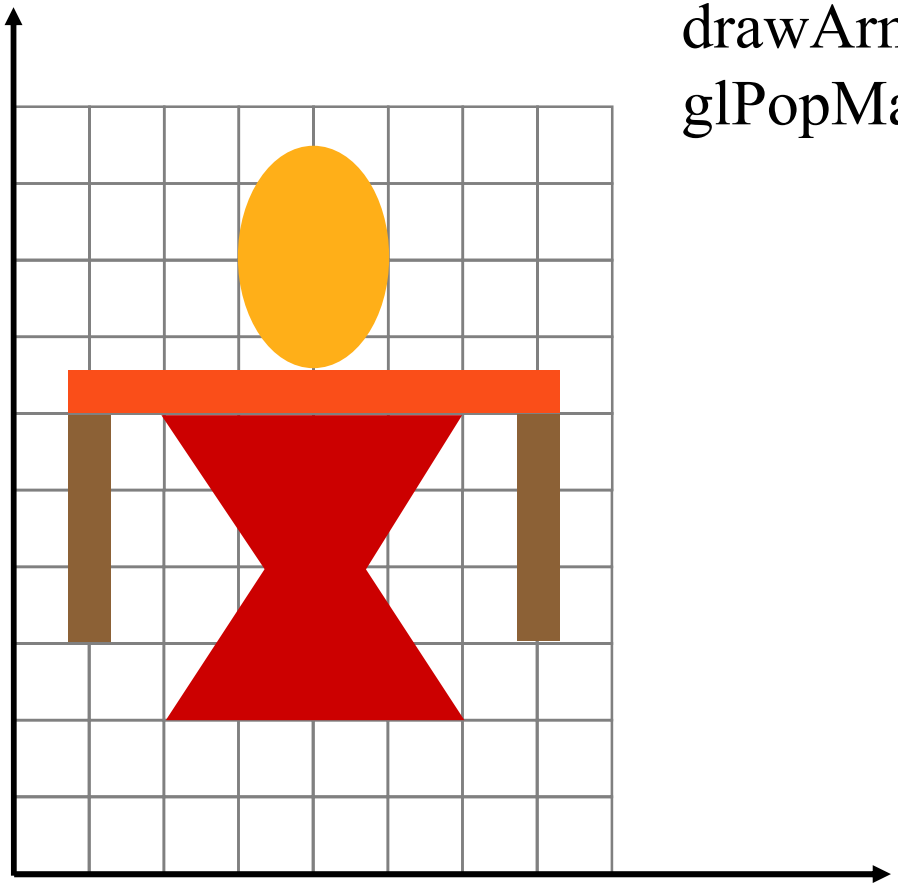
$$\begin{vmatrix} 1 & 0 & 4 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{vmatrix}$$

MODELVIEW Transformation

$$\begin{vmatrix} 1 & 0 & 6 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{vmatrix}$$

Head

```
glPushMatrix();
gltranslate(0,2);
drawArm();
glPopMatrix();
```



Stack

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

$$\begin{vmatrix} 1 & 0 & 4 \\ 0 & 1 & 4 \\ 0 & 0 & 1 \end{vmatrix}$$

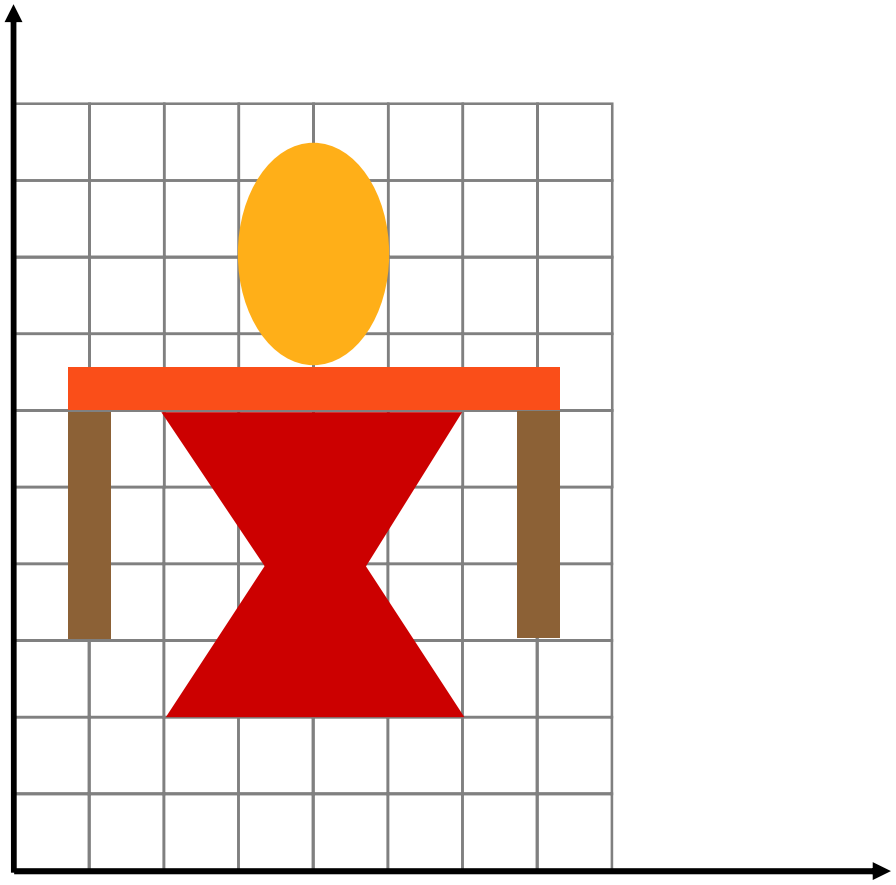
$$\begin{vmatrix} 1 & 0 & 4 \\ 0 & 1 & 6 \\ 0 & 0 & 1 \end{vmatrix}$$

MODELVIEW Transformation

$$\begin{vmatrix} 1 & 0 & 4 \\ 0 & 1 & 8 \\ 0 & 0 & 1 \end{vmatrix}$$

Cleanup

```
glPopMatrix();
glPopMatrix();
```



Stack

MODELVIEW Transformation

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

Summary of Code

```
glLoadIdentity();
glPushMatrix();
   gltranslate(4,4);
   drawTorso();
   glPushMatrix();
      gltranslate(0,2);
      drawShoulders();
      glPushMatrix();
          gltranslate(-3,0);
          drawArm();
      glPopMatrix();
      glPushMatrix();
          gltranslate(3,0);
          drawArm();
      glPopMatrix();
      glPushMatrix();
          gltranslate(0,2);
          drawHead();
      glPopMatrix();
   glPopMatrix();
glPopMatrix();
```

Indenting of code is a convenient way to indicate state of the transformation stack.

The four levels correspond with the four levels in the hierarchical model.