

# CS 418: Interactive Computer Graphics

---

## Hierarchical Modeling

Eric Shaffer

Based on slides from John Hart

# Hierarchical Modeling

- Lots of things have nested coordinate systems
  - ...otherwise known as local frames of reference
    - Any body with joints
    - The solar system
- Hierarchical modeling is a method for drawing such things
  - We keep a stack of affine transformations
  - Transforms are pushed and popped
    - interspersed with draw commands

# Build a Robot

...sort of...more like a rectangle with one arm...

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

We will use old-style OpenGL code to illustrate the ideas  
...it's less verbose

We will see it translated to WebGL at the end of class

**glutSolidCube(X)** draws a cube with side length of 2

**glPushMatrix()**

pushes a copy of current modeling matrix onto the stack

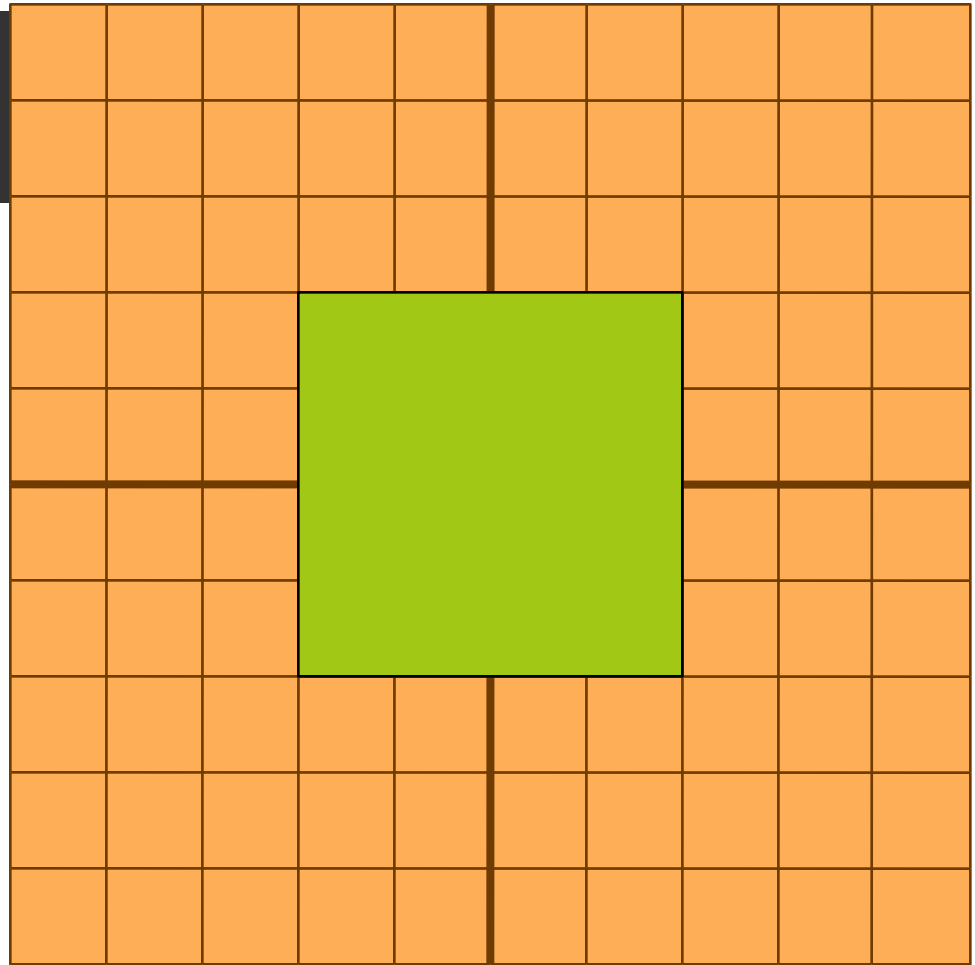
**glPopMatrix()** pops a matrix off the top of the stack

sets current matrix equal to returned matrix

**glScale/glTranslate/glRotate** all multiply the current matrix  
by the given transform matrix (on the right-hand-side)

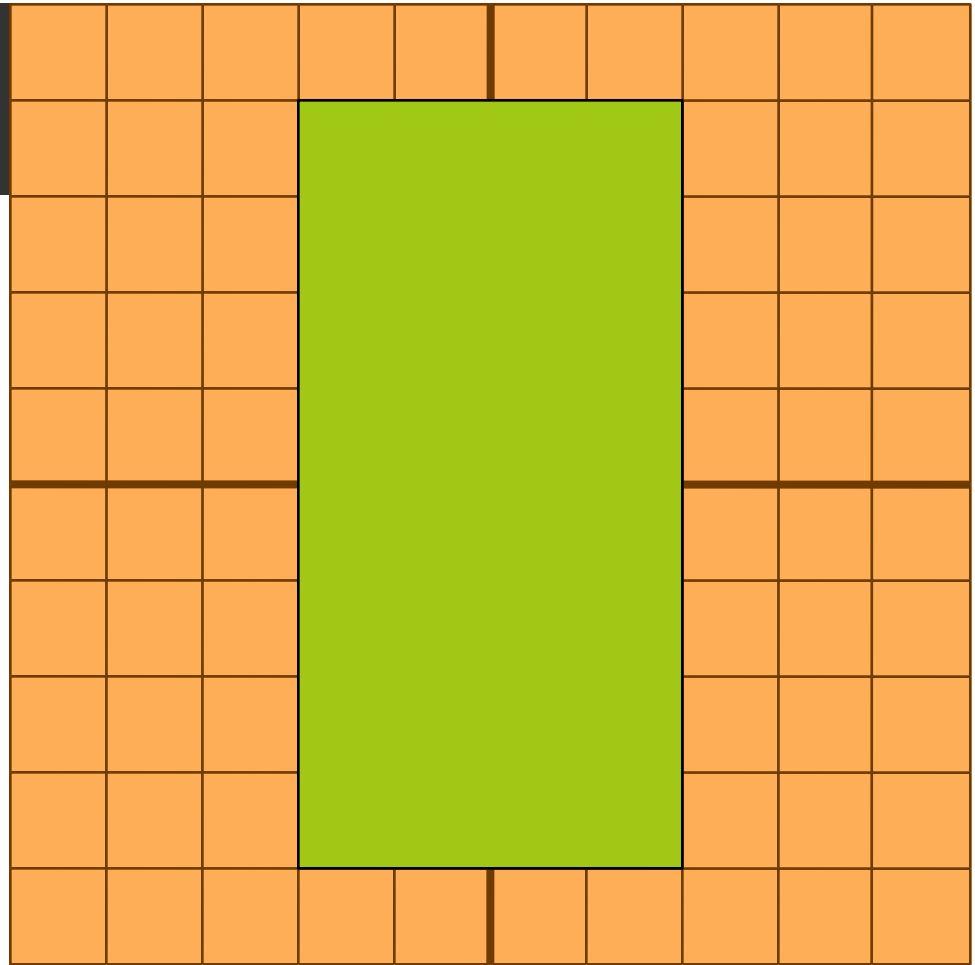
# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```



# Build a Robot

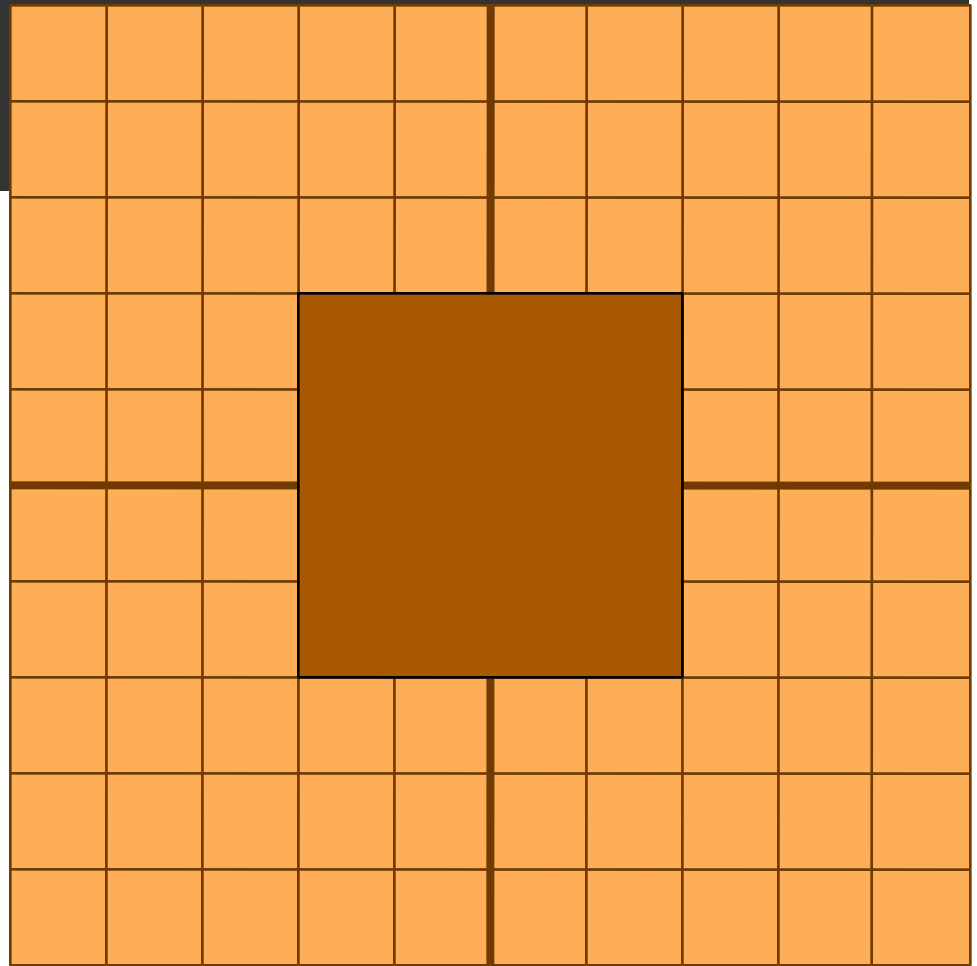
```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```



$$\begin{array}{|c|} \hline \text{Green Rectangle} \\ \hline \end{array} = \begin{bmatrix} 1 & & & \\ & 2 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \times \begin{array}{|c|} \hline \text{Small Green Square} \\ \hline \end{array}$$

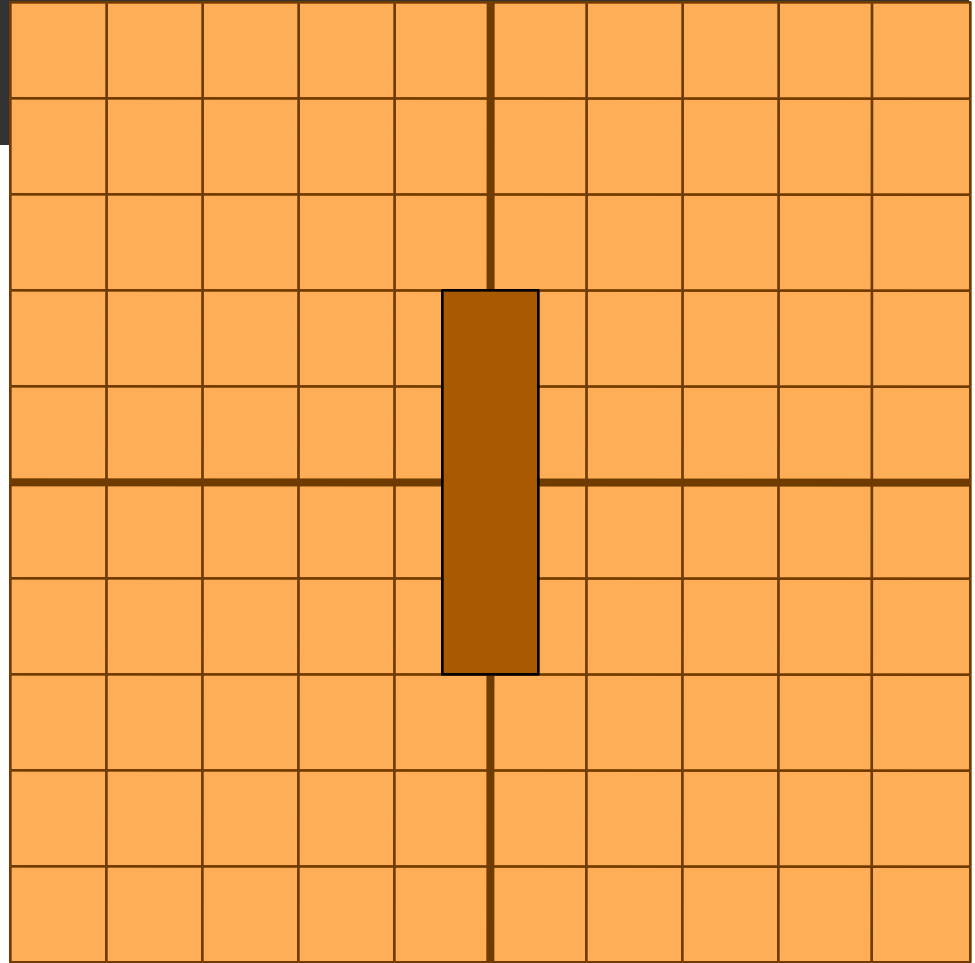
# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```



# Build a Robot

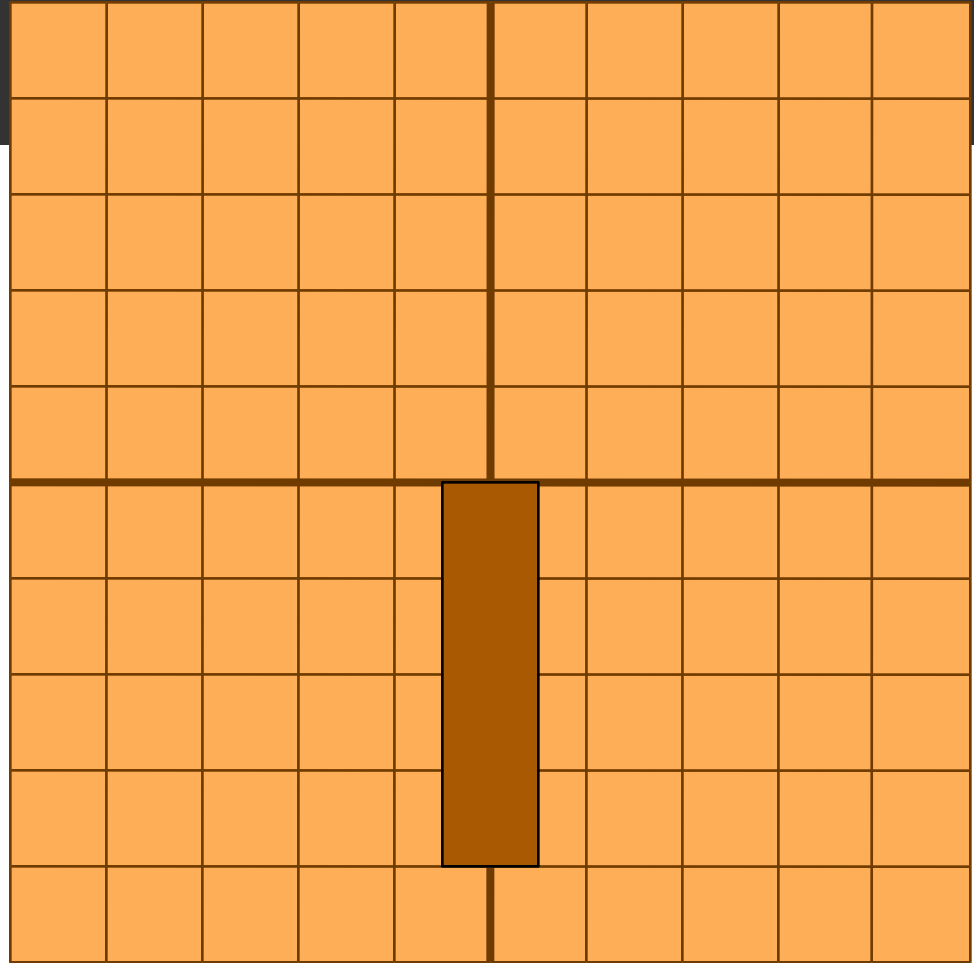
```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```



$$\begin{bmatrix} \frac{1}{4} & & & \\ & 1 & & \\ & & \frac{1}{4} & \\ & & & 1 \end{bmatrix}$$

# Build a Robot

```
glPushMatrix();
glPushMatrix();
glScalef(1.0,2.0,1.0);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(1.25,2,0.0);
glRotatef(shoulder,0,0,1);
glPushMatrix();
glTranslatef(0,-1,0.0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(0,-2,0);
glRotatef(elbow,0,0,1);
glTranslatef(0,-1,0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
```

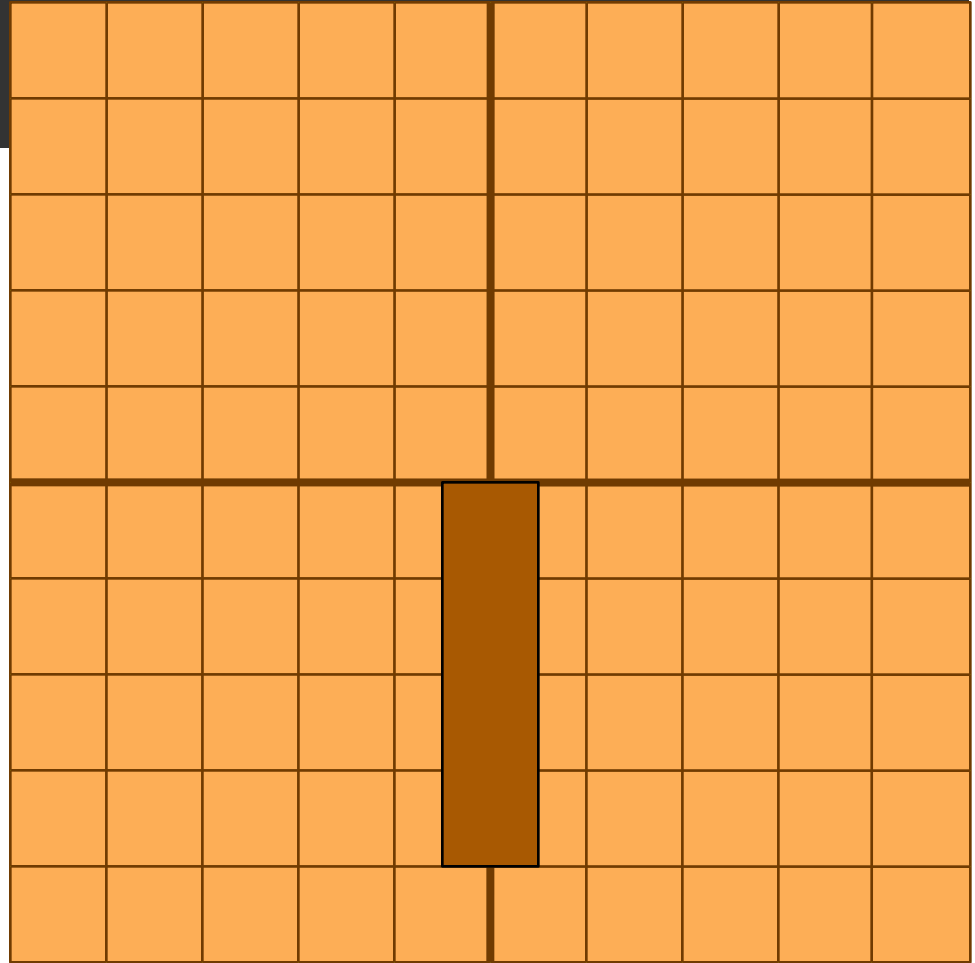


$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{4} & & & \\ & 1 & & \\ & & \frac{1}{4} & \\ & & & 1 \end{bmatrix}$$



# Build a Robot

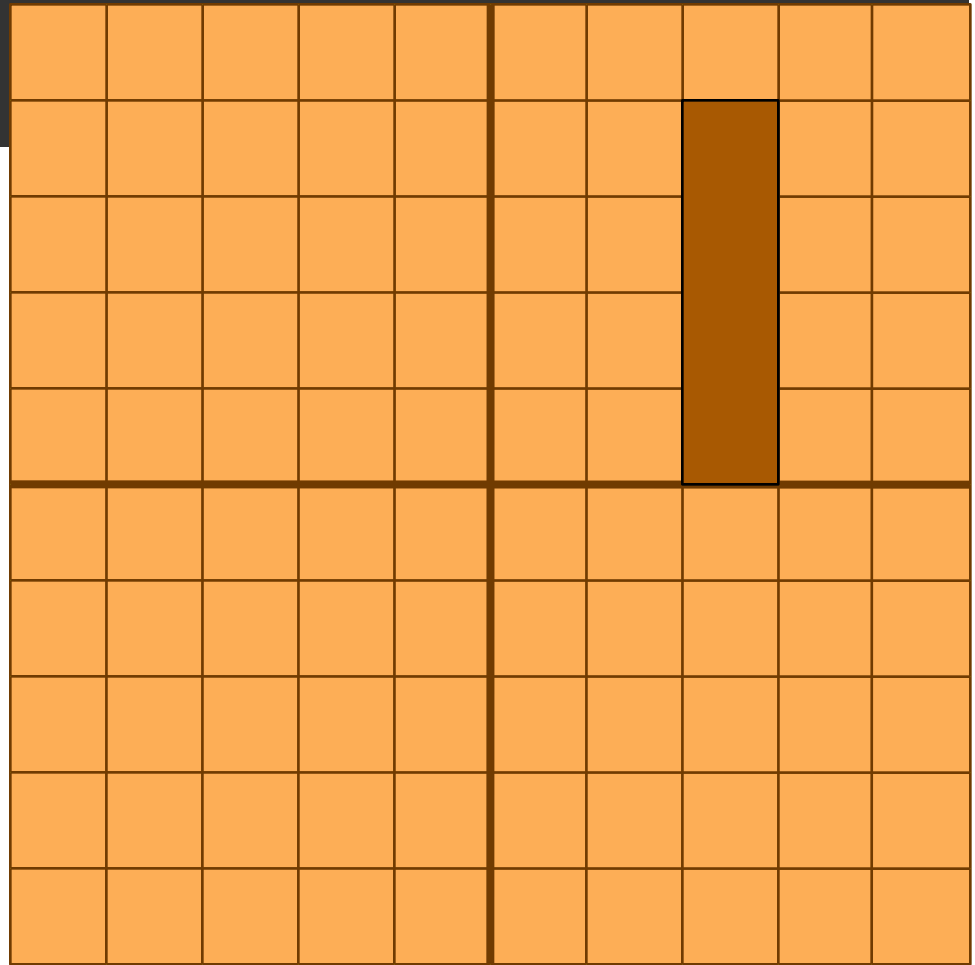
```
glPushMatrix();
glPushMatrix();
glScalef(1.0,2.0,1.0);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(1.25,2,0.0);
glRotatef(shoulder,0,0,1);
glPushMatrix();
glTranslatef(0,-1,0.0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(0,-2,0);
glRotatef(elbow,0,0,1);
glTranslatef(0,-1,0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
```



$$\begin{bmatrix} c & -s & & \\ s & c & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & -1 & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{4} & & & \\ & 1 & & \\ & & \frac{1}{4} & \\ & & & 1 \end{bmatrix}$$

# Build a Robot

```
glPushMatrix();
glPushMatrix();
glScalef(1.0,2.0,1.0);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(1.25,2,0.0);
glRotatef(shoulder,0,0,1);
glPushMatrix();
glTranslatef(0,-1,0.0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(0,-2,0);
glRotatef(elbow,0,0,1);
glTranslatef(0,-1,0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
```

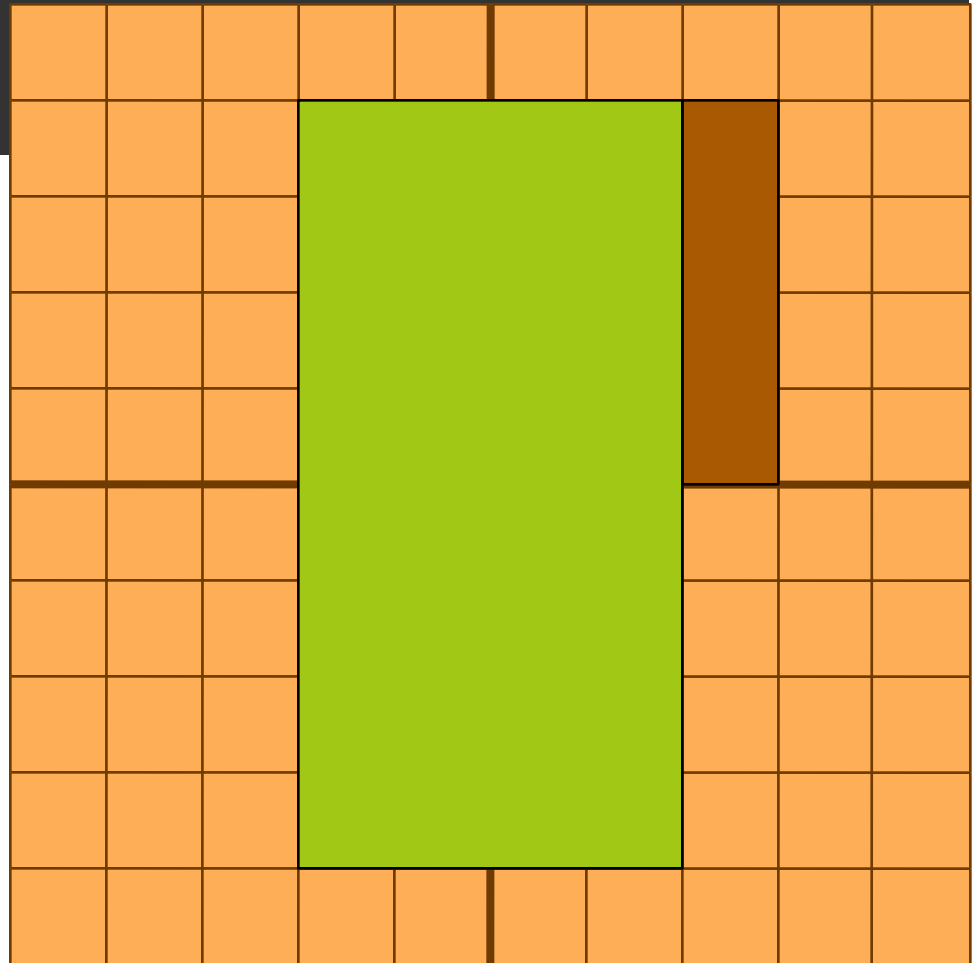


$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{4} \\ & 2 \\ & & 1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \\ & & 1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{4} & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

# Build a Robot

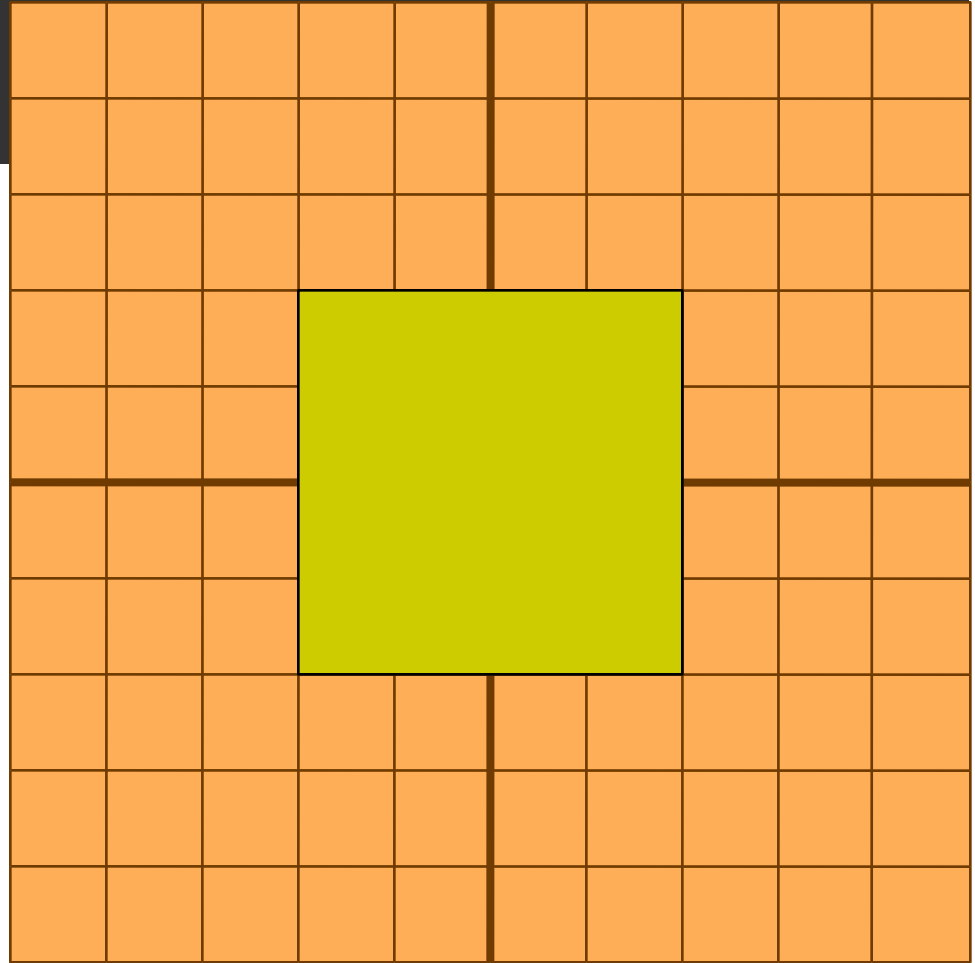
```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

*push/pop  
matrix  
keeps  
body  
scale  
from  
affecting  
shoulder*



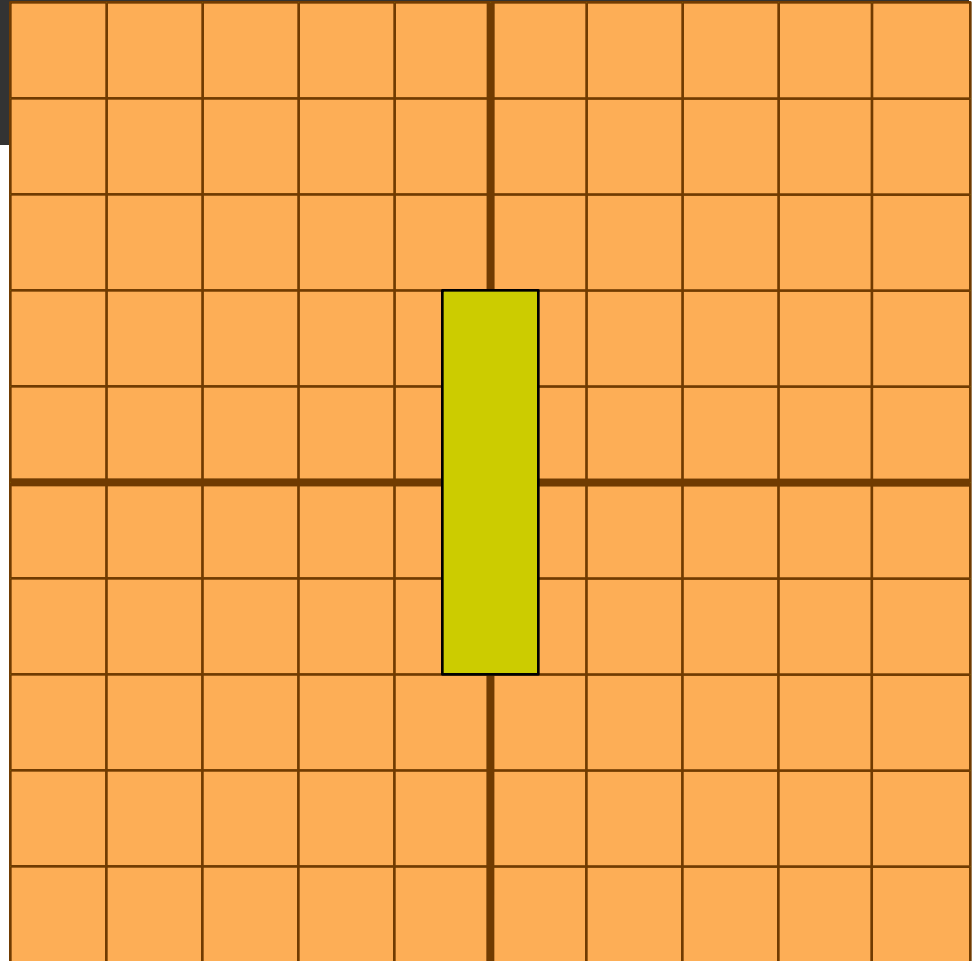
# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```



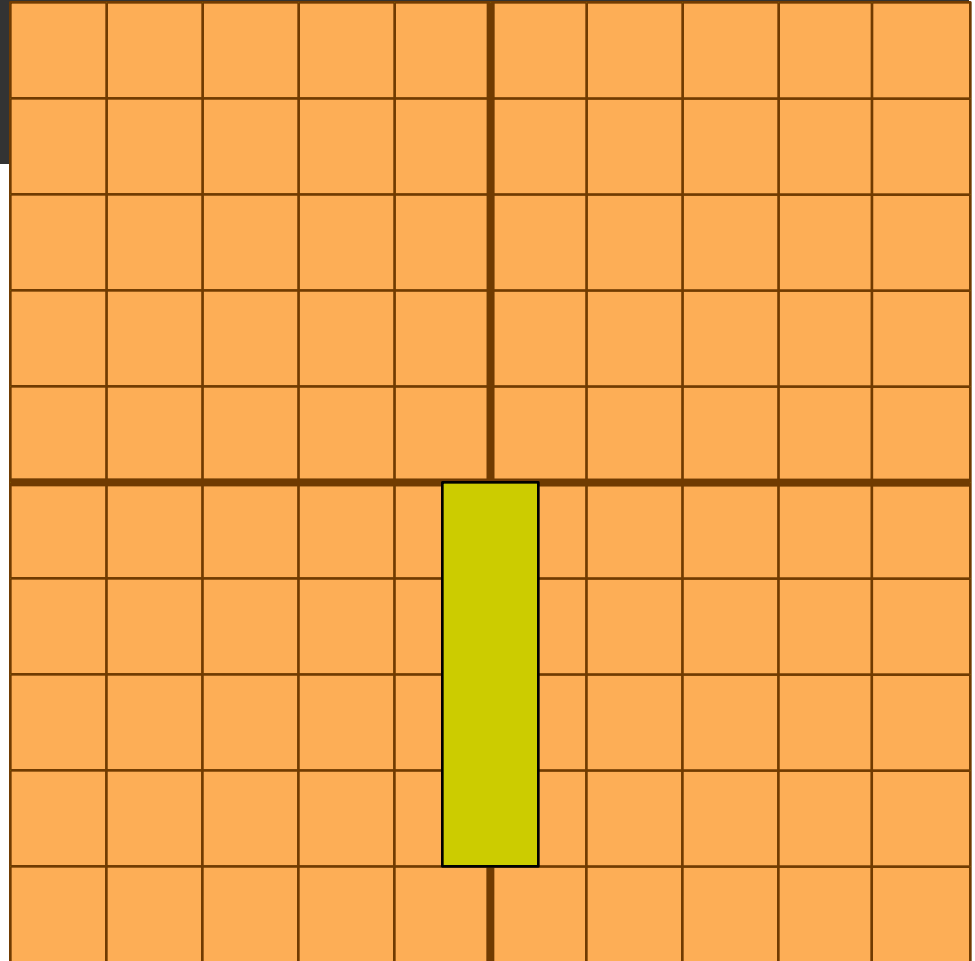
# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```



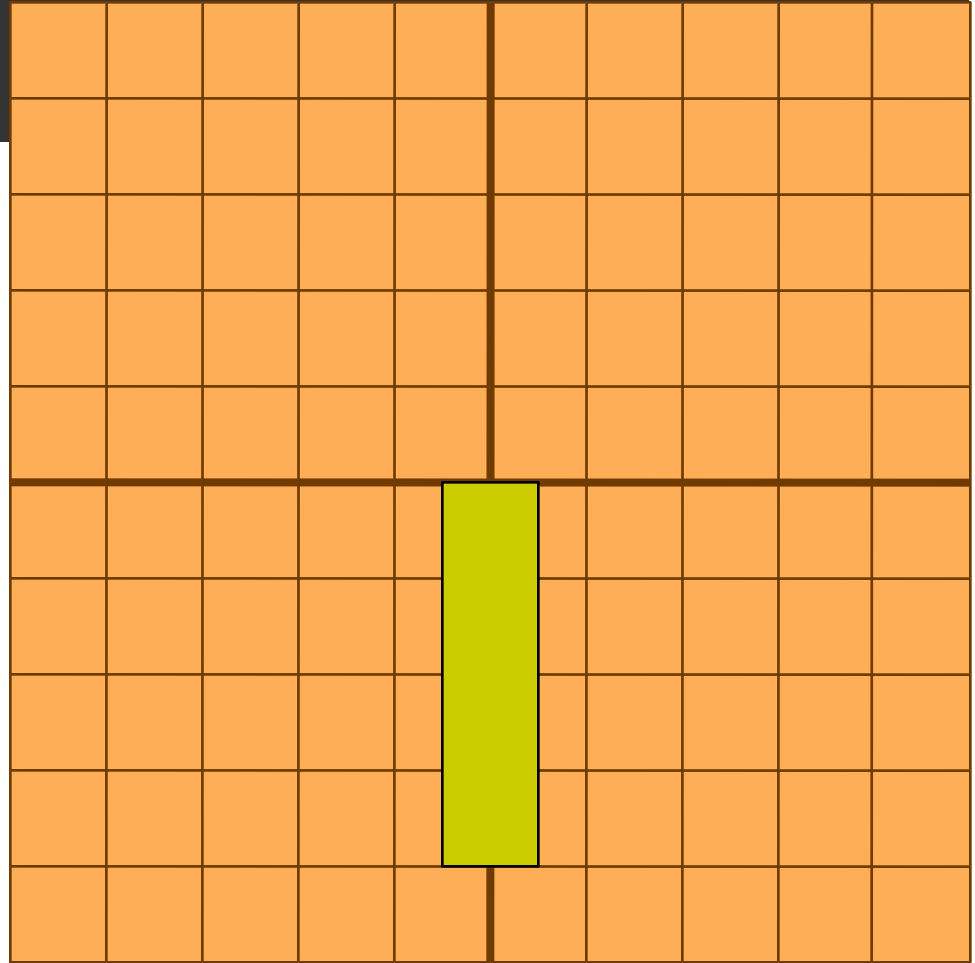
# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```



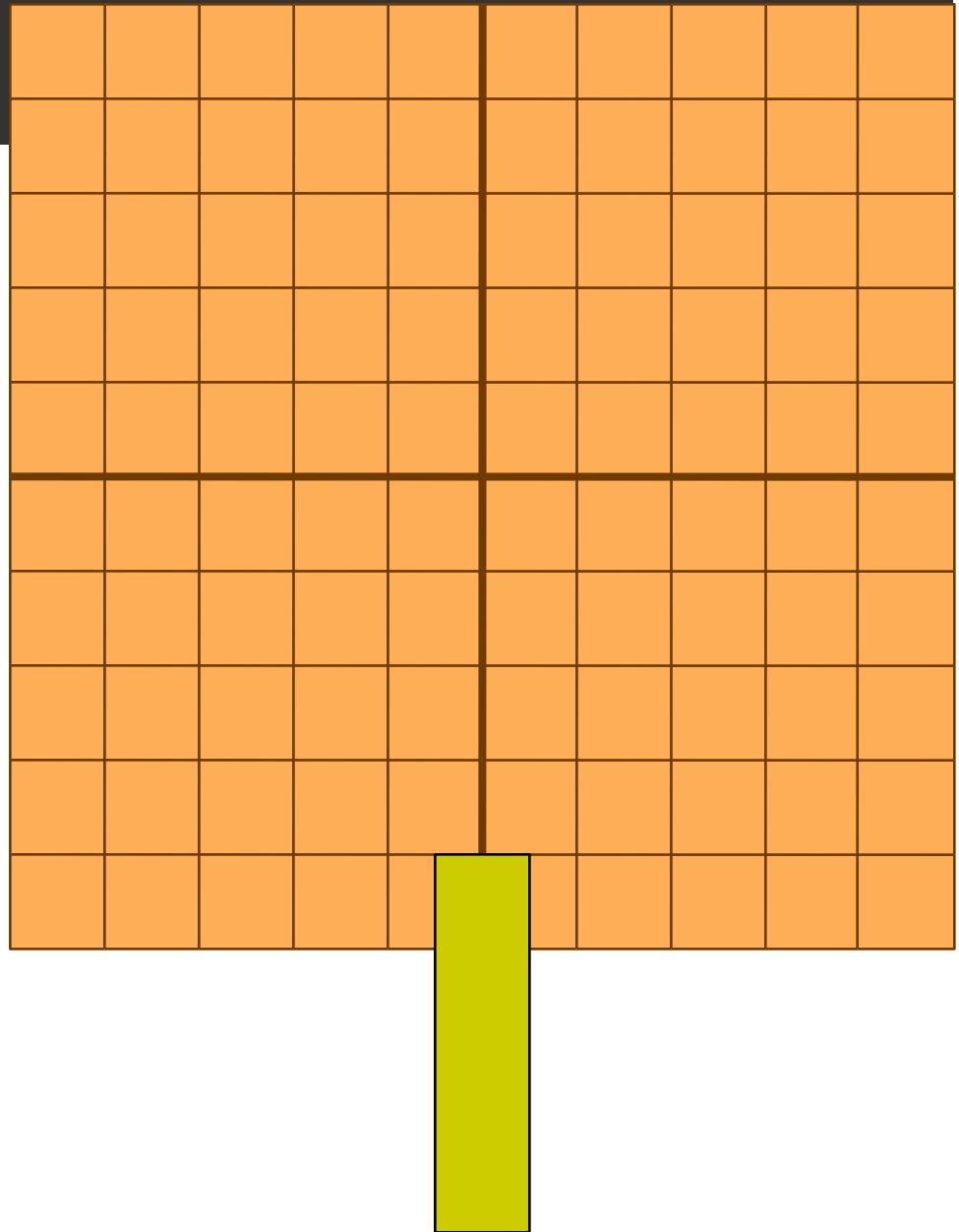
# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```



# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

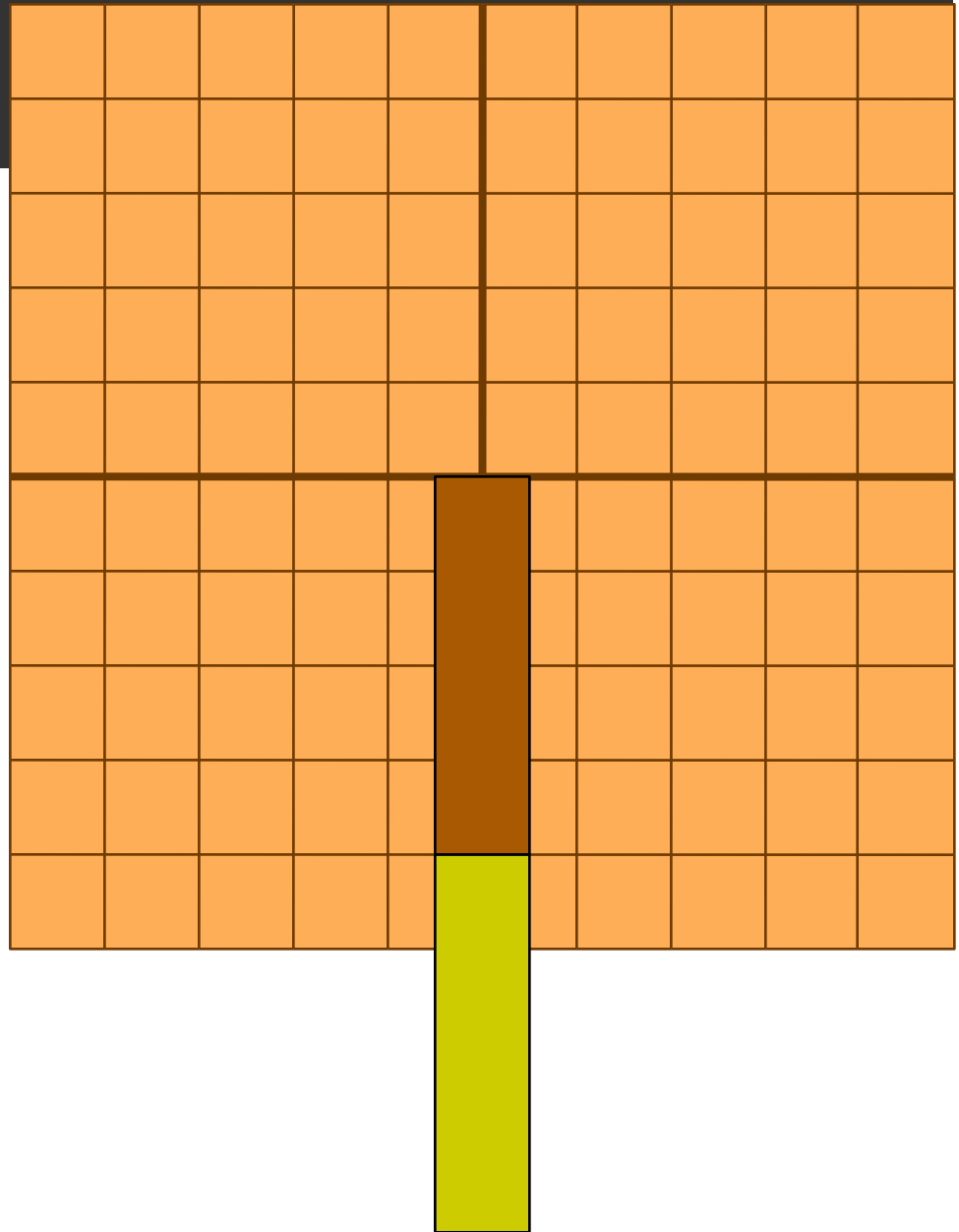




# Build a Robot

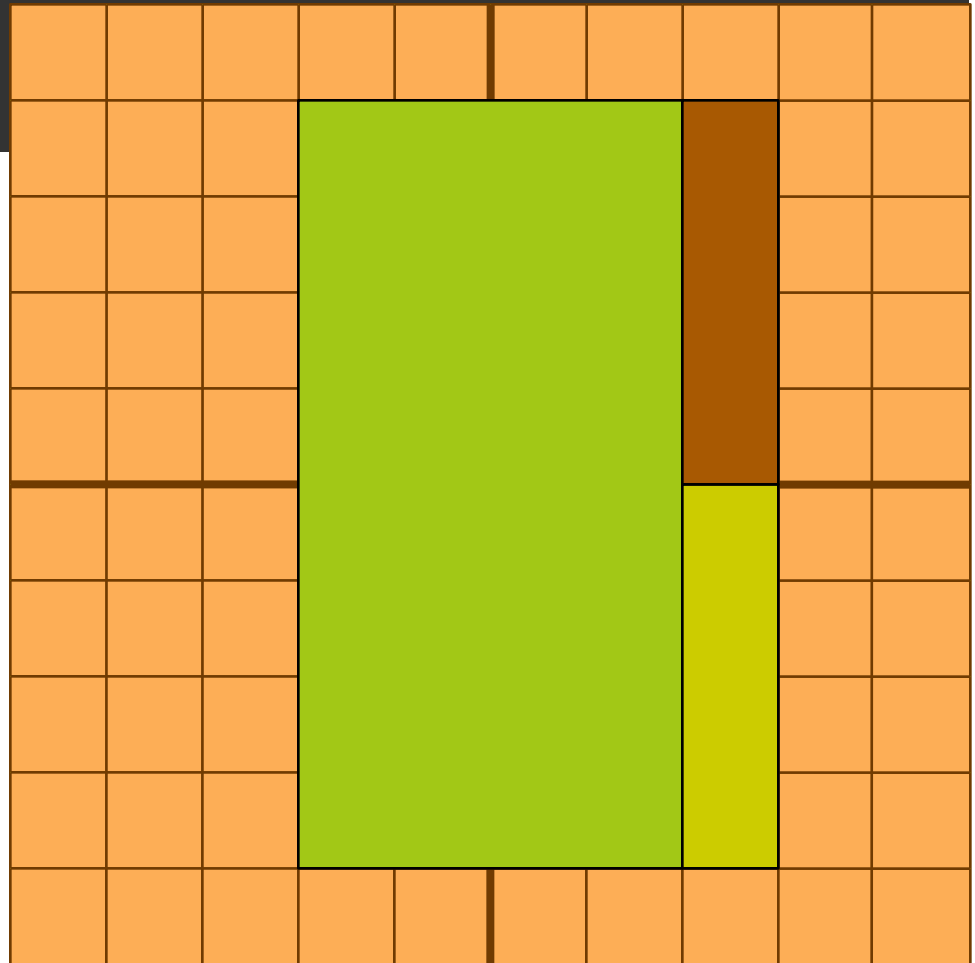
```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

*push/pop  
allows the  
forearm  
to ignore  
these*



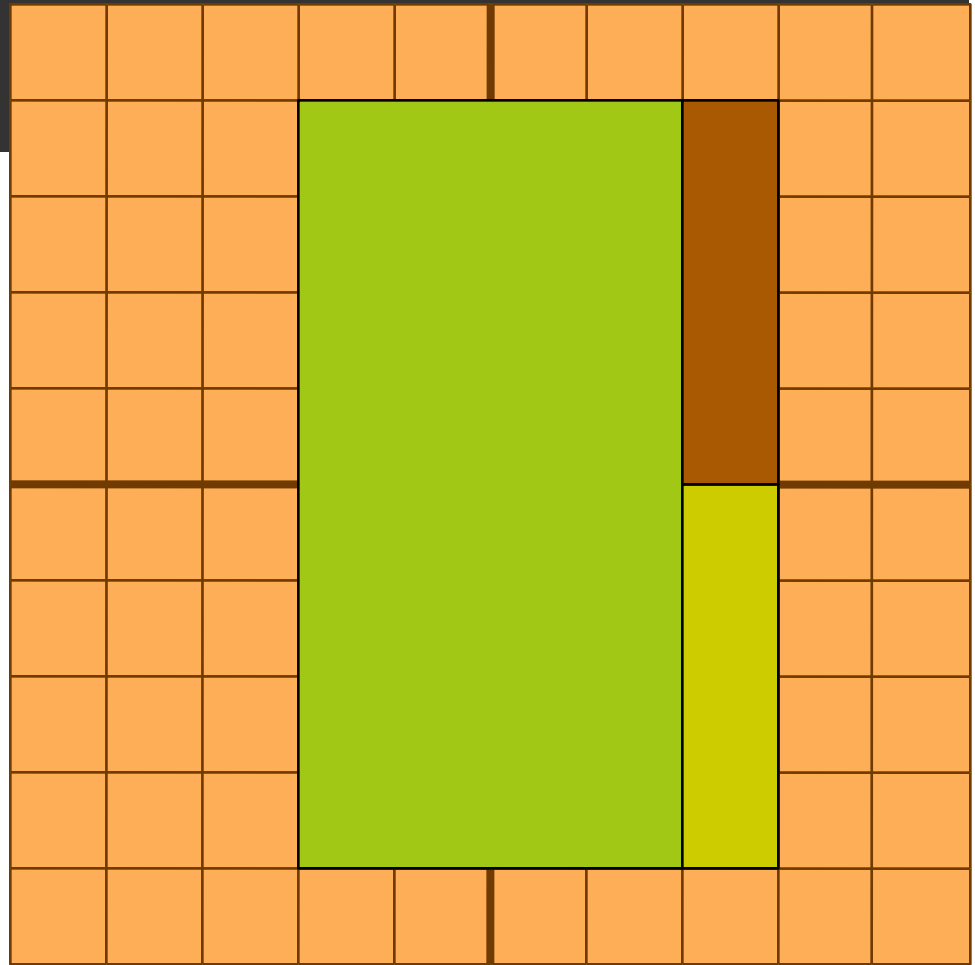
# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```



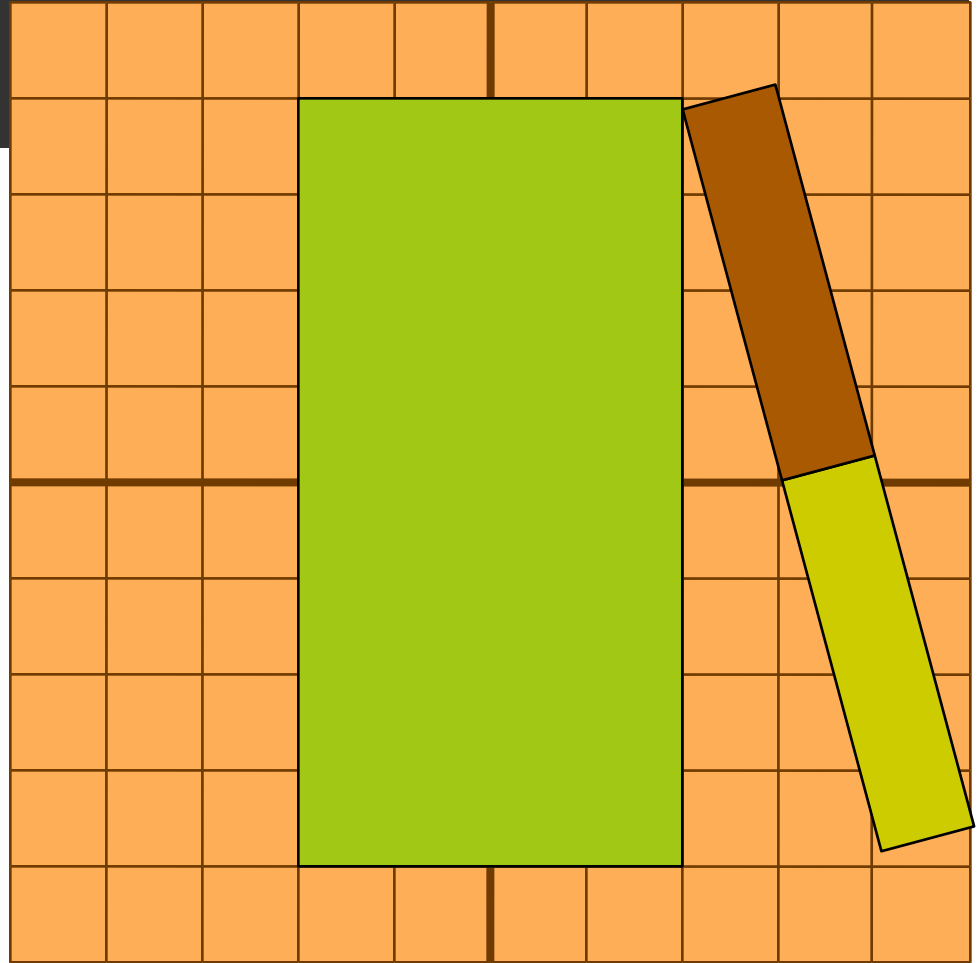
# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```



# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```



# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

Modelview:

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

Stack:

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

# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

Modelview:

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

Stack:

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

# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

Modelview:

$$\begin{bmatrix} 1 & & & \\ & 2 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Stack:

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

# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

Modelview:

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

Stack:

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



# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

Modelview:

$$\begin{bmatrix} 1 & & 1\frac{1}{4} \\ & 1 & 2 \\ & & 1 \\ & & & 1 \end{bmatrix}$$

Stack:

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

# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

Modelview:

$$\begin{bmatrix} 1 & & 1\frac{1}{4} \\ & 1 & 2 \\ & & 1 \\ & & 1 \end{bmatrix} \begin{bmatrix} c & -s & & \\ s & c & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Stack:

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

# Build a Robot

```

glPushMatrix();
glPushMatrix();
glScalef(1.0,2.0,1.0);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(1.25,2,0.0);
glRotatef(shoulder,0,0,1);
glPushMatrix();
glTranslatef(0,-1,0.0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(0,-2,0);
glRotatef(elbow,0,0,1);
glTranslatef(0,-1,0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();

```

Modelview:

$$\begin{bmatrix} 1 & & 1\frac{1}{4} \\ & 1 & 2 \\ & & 1 \\ & & 1 \end{bmatrix} \begin{bmatrix} c & -s & & \\ s & c & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Stack:

$$\begin{bmatrix} c & -s & 1\frac{1}{4} \\ s & c & 2 \\ & & 1 \\ & & 1 \end{bmatrix}, \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

# Build a Robot

```
glPushMatrix();
glPushMatrix();
glScalef(1.0,2.0,1.0);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(1.25,2,0.0);
glRotatef(shoulder,0,0,1);
glPushMatrix();
glTranslatef(0,-1,0.0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(0,-2,0);
glRotatef(elbow,0,0,1);
glTranslatef(0,-1,0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
```

Modelview:

$$\begin{bmatrix} c & -s & 1\frac{1}{4} \\ s & c & 2 \\ & & 1 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & -1 \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Stack:

$$\begin{bmatrix} c & -s & 1\frac{1}{4} \\ s & c & 2 \\ & & 1 \\ & & 1 \end{bmatrix}, \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

# Build a Robot

```
glPushMatrix();
glPushMatrix();
glScalef(1.0,2.0,1.0);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(1.25,2,0.0);
glRotatef(shoulder,0,0,1);
glPushMatrix();
glTranslatef(0,-1,0.0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(0,-2,0);
glRotatef(elbow,0,0,1);
glTranslatef(0,-1,0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
```

Modelview:

$$\begin{bmatrix} c & -s & 1\frac{1}{4} \\ s & c & 2-c \\ & 1 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ 1 \\ \frac{1}{4} \\ 1 \end{bmatrix}$$

Stack:

$$\begin{bmatrix} c & -s & 1\frac{1}{4} \\ s & c & 2 \\ & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

Modelview:

$$\begin{bmatrix} c & -s & 1\frac{1}{4} \\ s & c & 2 \\ & & 1 \\ & & 1 \end{bmatrix}$$

Stack:

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

# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

Modelview:

$$\begin{bmatrix} c & -s & 1\frac{1}{4} \\ s & c & 2 \\ & & 1 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & -2 \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Stack:

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

# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

Modelview:

$$\begin{bmatrix} c & -s & 1\frac{1}{4} \\ s & c & 2-2c \\ & & 1 \\ & & & 1 \end{bmatrix} \begin{bmatrix} c' & s' & & \\ s' & c' & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Stack:

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



# Build a Robot

```
glPushMatrix();
glPushMatrix();
glScalef(1.0,2.0,1.0);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(1.25,2,0.0);
glRotatef(shoulder,0,0,1);
glPushMatrix();
glTranslatef(0,-1,0.0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(0,-2,0);
glRotatef(elbow,0,0,1);
glTranslatef(0,-1,0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
```

Modelview:

$$\begin{bmatrix} cc'-ss' & cs'-c's & 1\frac{1}{4} \\ c's+cs' & cc'+ss' & 2-2c \\ & & 1 \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & \\ & 1 & & -1 \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Stack:

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

# Build a Robot

```
glPushMatrix();
glPushMatrix();
glScalef(1.0,2.0,1.0);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(1.25,2,0.0);
glRotatef(shoulder,0,0,1);
glPushMatrix();
glTranslatef(0,-1,0.0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
glTranslatef(0,-2,0);
glRotatef(elbow,0,0,1);
glTranslatef(0,-1,0);
glScalef(0.25,1.0,0.25);
glutSolidCube(2.0);
glPopMatrix();
```

Modelview:

$$\begin{bmatrix} cc' - ss' & cs' - c's & 1\frac{1}{4} + c's - cs' \\ c's + cs' & cc' + ss' & 2 - 2c - cc' - ss' \\ & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{4} \\ 1 \\ \frac{1}{4} \\ 1 \end{bmatrix}$$

Stack:

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

# Build a Robot

```
glPushMatrix();  
glPushMatrix();  
glScalef(1.0,2.0,1.0);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(1.25,2,0.0);  
glRotatef(shoulder,0,0,1);  
glPushMatrix();  
glTranslatef(0,-1,0.0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();  
glTranslatef(0,-2,0);  
glRotatef(elbow,0,0,1);  
glTranslatef(0,-1,0);  
glScalef(0.25,1.0,0.25);  
glutSolidCube(2.0);  
glPopMatrix();
```

Modelview:

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

Stack:

# Hierarchical Modeling in WebGL

```
var mvMatrix = mat4.create();
var mvMatrixStack = [];

function mvPushMatrix() {
    var copy = mat4.clone(mvMatrix);
    mvMatrixStack.push(copy);
}

function mvPopMatrix() {
    if (mvMatrixStack.length == 0) {
        throw "Invalid popMatrix!";
    }
    mvMatrix = mvMatrixStack.pop();
}
```

- Implement your own matrix stack
- Used for modeling...not projection

# Hierarchical Modeling in WebGL

```
var transformVec = vec3.create();

gl.viewport(0, 0, gl.viewportWidth, gl.viewportHeight);

gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);

mat4.ortho(pMatrix, -5, 5, -5, 5, 2, -2);

mat4.identity(mvMatrix);

mvPushMatrix();

mvPushMatrix();

vec3.set(transformVec, 1.0, 2.0, 1.0);

mat4.scale(mvMatrix, mvMatrix, transformVec);

//draw body

gl.bindBuffer(gl.ARRAY_BUFFER, vertexPositionBuffer);

gl.vertexAttribPointer(shaderProgram.vertexPositionAttribute,
                        vertexPositionBuffer.itemSize, gl.FLOAT, false, 0, 0);

gl.bindBuffer(gl.ARRAY_BUFFER, vertexRedColorBuffer);

gl.vertexAttribPointer(shaderProgram.vertexColorAttribute,
                        vertexRedColorBuffer.itemSize, gl.FLOAT, false, 0, 0);

setMatrixUniforms();

gl.drawArrays(gl.TRIANGLES, 0, vertexPositionBuffer.numberOfItems);
```

# Projection Matrix in WebGL

```
<script id="shader-vs" type="x-shader/x-vertex">
  attribute vec3 aVertexPosition;
  attribute vec4 aVertexColor;
  uniform mat4 uMVMMatrix;
  uniform mat4 uPMatrix;

  varying vec4 vColor;

  void main(void) {
    gl_Position = uPMatrix*uMVMMatrix*vec4(aVertexPosition, 1.0);
    vColor = aVertexColor;
  }
</script>
```

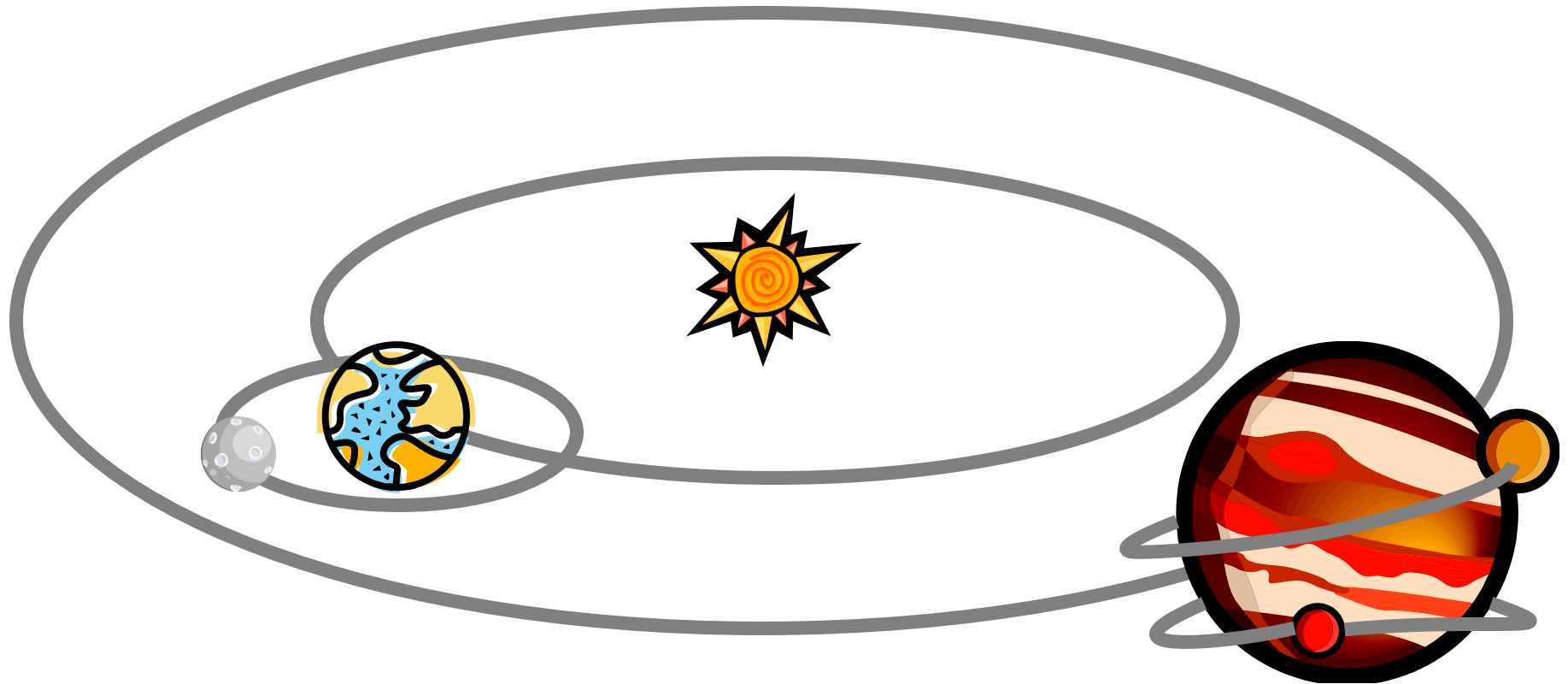
- Allows your world coordinate system to be different than clip coordinates
- Used for modeling...not projection
- Need to set up and send down another matrix uniform

# Orthogonal Projection in WebGL

```
function draw(){...  
  gl.viewport(0, 0, gl.viewportWidth, gl.viewportHeight);  
  gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);  
  mat4.ortho(pMatrix, -5, 5, -5, 5, 2, -2);  
  ...}
```

- Allows you to work in a box like clip space
- You decide the box dimensions
- Parallel projection onto the view plane

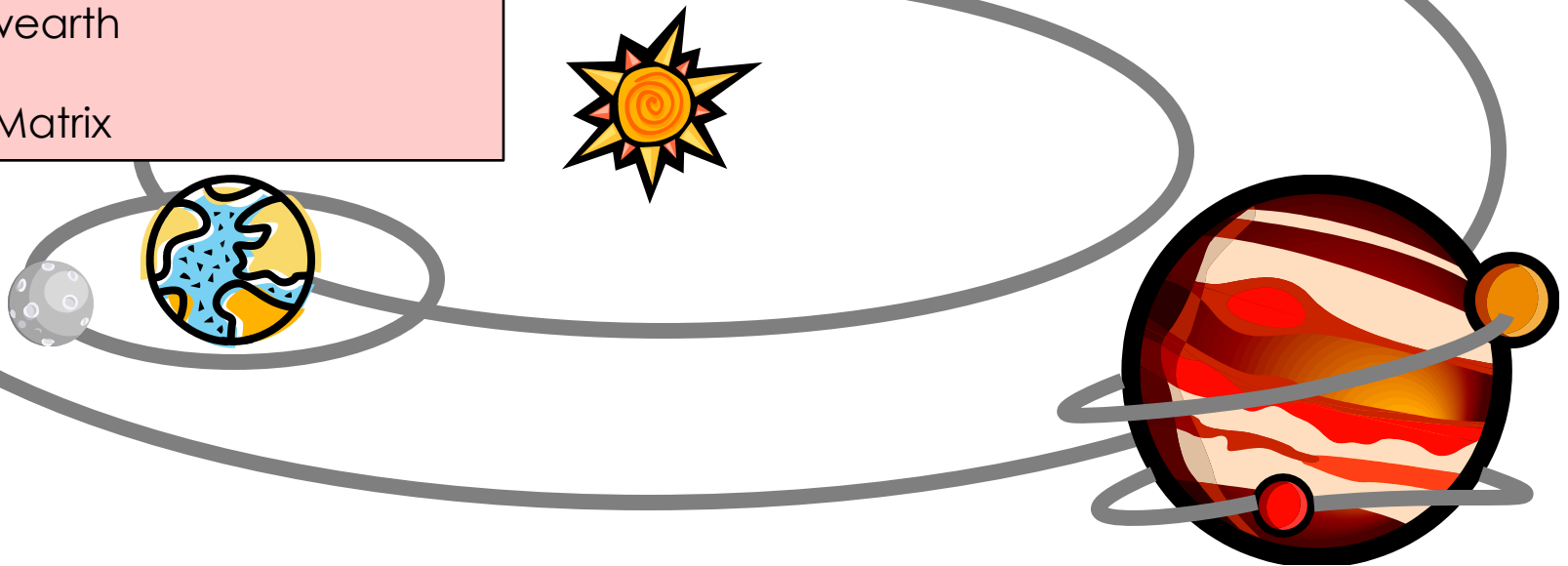
# Solar System





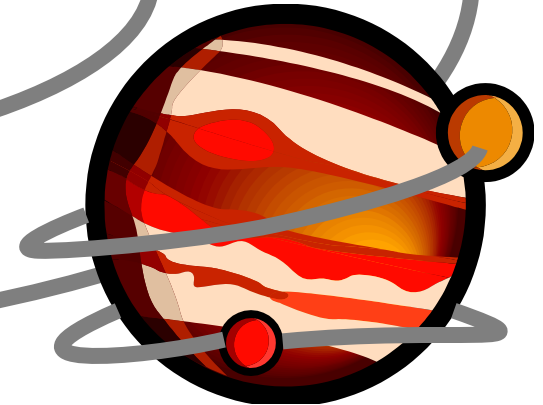
# Solar System

```
PushMatrix  
Rotate 360*days/365,(0,1,0)  
Translate (AU,0,0)  
Rotate 23,(1,0,0)  
Rotate 360*hours/24,(0,1,0)  
drawearth  
...  
PopMatrix
```



# Solar System

```
PushMatrix  
Rotate 360*days/365,(0,1,0)  
Translate (AU,0,0)  
Rotate 23,(1,0,0)  
Rotate 360*hours/24,(0,1,0)  
drawearth  
...  
PopMatrix
```



```
PushMatrix  
Rotate 360*days/27,(0,1,0)  
Translate 238856,0,0  
Rotate -360*days/29,(0,1,0)  
drawmoon  
PopMatrix
```