Transformation Hierarchies

Often reuse the same geometry with different transformation matrices.

Matrix Stacks
- useful in scenes with a hierarchical structure
- allows return to a previously-used coordinate system

There exist in some graphics APIs. In WebGL or modern OpenGL, you will need to write your own, or use pass-by-value function calls.
### Transformation Hierarchies

#### Math view

\[
P_{\text{cam}} = M^{-1}_{\text{cam}} M_1 M_2 M_3 M_4 M_5 P_5
\]

\[
P_{\text{head}} = M^{-1}_2 M^{-1}_3 M^{-1}_4 M^{-1}_5 P_5
\]

\[
P_5 = \begin{pmatrix} M^{-1}_2 & M^{-1}_3 & M^{-1}_4 & M^{-1}_5 \end{pmatrix} P_5
\]

\[
(AB)^{-1} = B^{-1} A^{-1}
\]

\[
\rightarrow = M \text{ in the code below.}
\]

#### Code view

```cpp
M.Translate(a, b, 0);
M.Rotate(tf(\theta_1, 0,0,1));
DrawBody();
PushMatrix(M);
M.Translate(0,7,0);
M.DrawHead();
M=PopMatrix();
PushMatrix(M);
M.Translate(2.5,5.5,0);
M.Rotate(\theta_2, 0,0,1);
DrawRUArm();
M.Translate(0,-3.5,0);
M.Rotate(\theta_3, 0,0,1);
DrawRLArm();
M=PopMatrix();
...
```

looking at character from behind

Push/pop allows for an easy way to return to a previous coordinate system.

This could also be done implicitly by passing $M$, by value, to function calls that draw the individual parts.