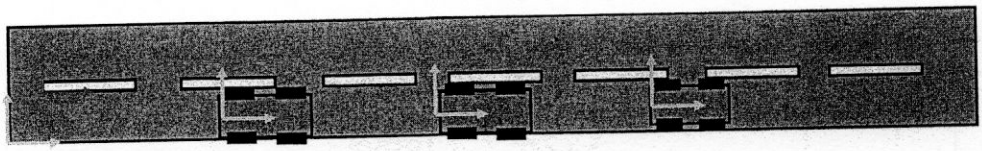
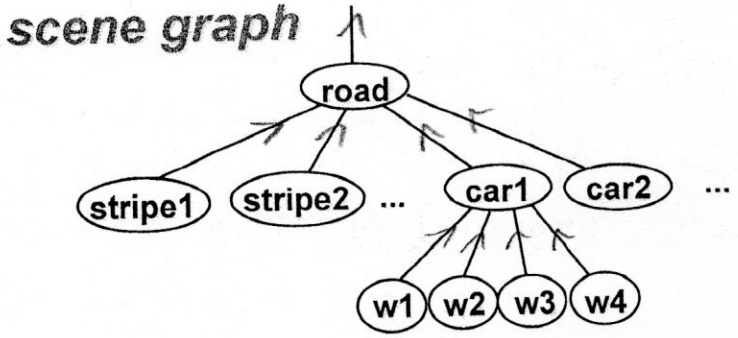
Ref.

Transformation Hierarchies



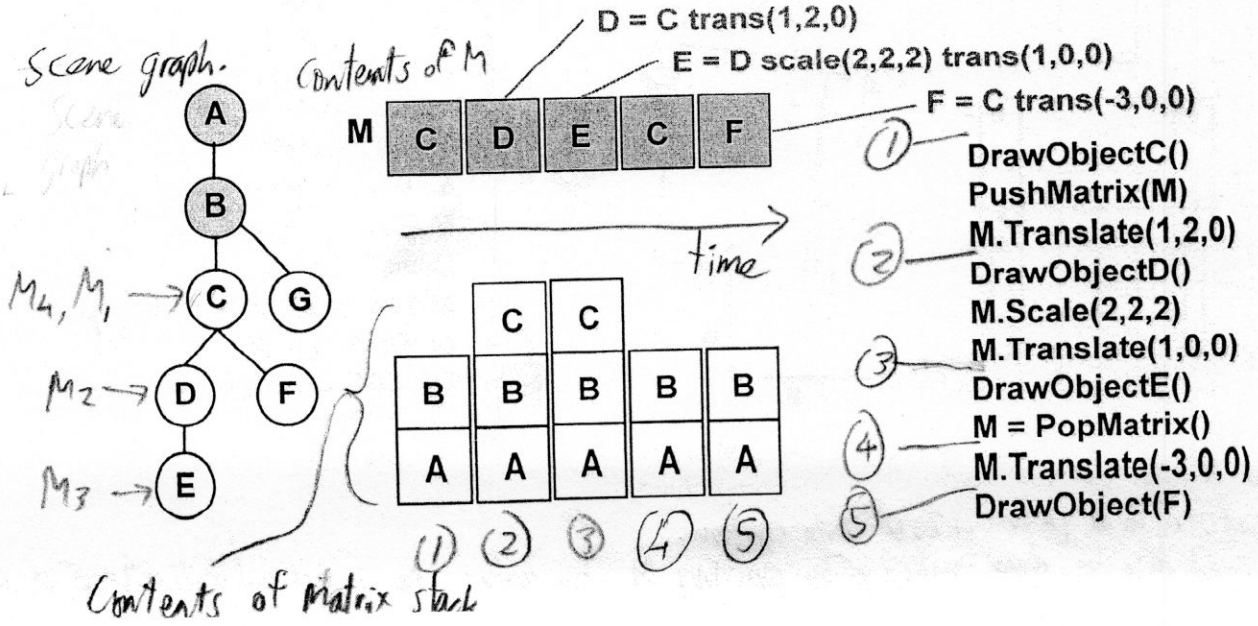


Often reuse the same geometry with different transformation matrices.

Matrix Stacks

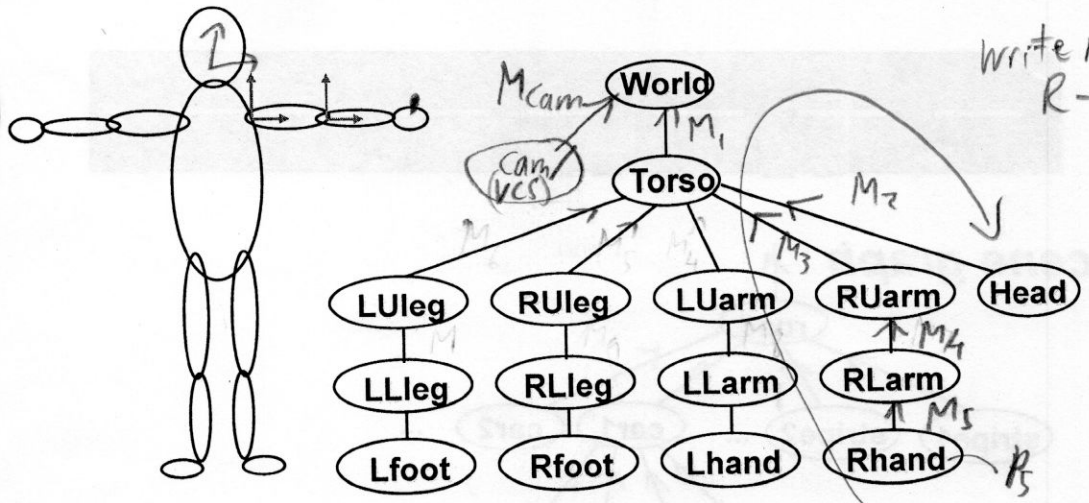
These exist in some graphics APIs.
In WebGL or modern OpenGL, you will need to write your own, or use pass-by-value function calls.

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



Transformation Hierarchies

Scene graph view



Write matrices R-to-L

Math view

$P_{cam} = M_{Cam}^{-1} M_1 M_3 M_4 M_5 P_5$

$P_{vcs} = M_{view} \cdot M_{model} \cdot P_5$ ↳ = M in the code below.

$P_{Head} = M_2^{-1} M_3 M_4 M_5 P_5$

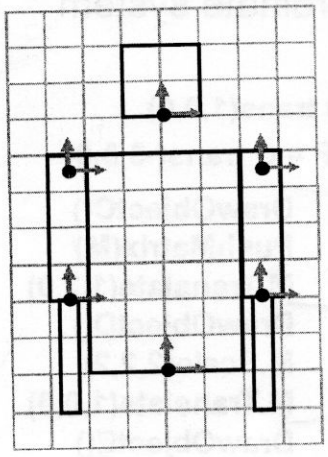
$P_5 = M_5^{-1} M_4^{-1} M_3^{-1} M_2^{-1} P_{Head}$

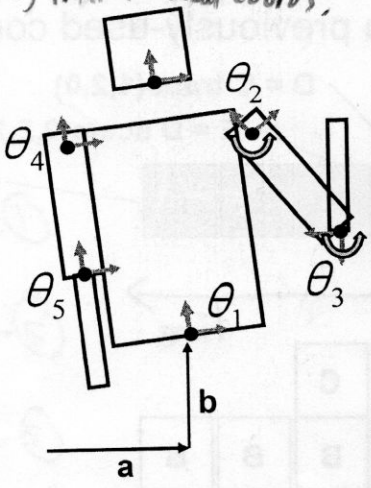
$(AB)^{-1} = B^{-1} A^{-1}$

Transformation Hierarchies

Code view

Code: Write matrices L-to-R, i.e., think in local coords.





looking at character from behind

```
M.Translate(a,b,0);          } M1
M.Rotatef(θ1,0,0,1);
DrawBody();
PushMatrix(M);
  M.Translate(0,7,0);        } M2
  M.DrawHead();
M=PopMatrix();
PushMatrix(M);
  M.Translate(2.5,5.5,0);    } M3
  M.Rotate(θ2,0,0,1);
  DrawRUarm();
  M.Translate(0,-3.5,0);     } M4
  M.Rotate(θ3,0,0,1);
  DrawRLarm();
M=PopMatrix();
... (draw left arm)
```

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.