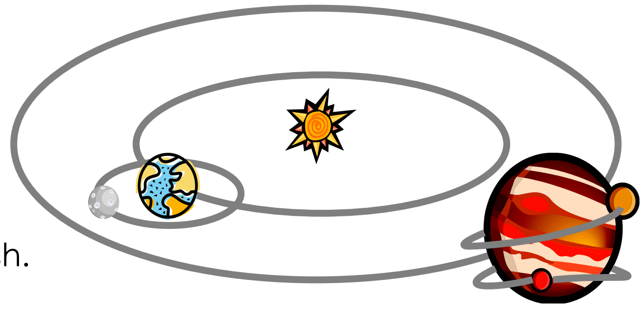
CS 418: Interactive Computer Graphics
In-class Worksheet 2

Viewing and Hierarchical Modeling

1. Modeling a (part of) a Solar System

Fix the following code so that it correctly models the Earth and Moon positions. You can insert Push/Pop commands and rearrange any lines of code you wish.



```
PushMatrix
Scale(7917,7917,7917)          // Set Earth diameter
Translate (AU,0,0)              // AU = distance from Earth to Sun
Rotate 360*days/365,(0,1,0)   // Rotation around sun
Scale (2159,2519,2519)          // Set moon diameter
Rotate 360*days/27,(0,1,0)     //  Moon rotation around Earth
Translate 238856,0,0            // Earth to moon distance
DrawMoon
DrawEarth
PopMatrix
```

2. **Memory Layout of Matrices in WebGL**

   Suppose we have the following transformation matrix:

$$\begin{bmatrix} a & b & c & t_x \\ d & e & f & t_y \\ g & h & i & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

   Write down the column-major layout of the matrix in memory:

   Write down the row-major layout of the matrix in memory:

   Our vertex shaders have typically transformed vertex positions using a line of code like this:

```
gl_Position = uMVMatrix*vec4(aVertexPosition, 1.0);
```

   Suppose in your JavaScript code your matrices are laid out in row-major instead of column-major order. How could you change the vertex shader code to accommodate that? Just alter the one line. Note: Assume that your browser only supports GLSL ES 1.0 (which if it's Chrome or Edge it probably does) and so does not have `tranpose` function built-in.