6.837 Intro to Computer Graphics, Fall 2003 Assignment 2: Transformations and additional primitives

Scene description file grammar

```
                file  ::=  camera lights background group

              camera  ::=  orthographicCamera | perspectiveCamera

orthographicCamera  ::=  OrthographicCamera {
                             center Vec3f
                             direction Vec3f
                             up Vec3f
                             size float
                         }

 perspectiveCamera  ::=  PerspectiveCamera {
                             center Vec3f
                             direction Vec3f
                             up Vec3f
                             angle float
                         }

             lights  ::=  Lights {
                             numLights int
                             light ^numLights
                         }

              light  ::=  directionalLight

  directionalLight  ::=  DirectionalLight {
                             direction Vec3f
                             color Vec3f
                         }
```

```
    background ::= Background {
                       color Vec3f
                       ambientLight Vec3f
                   }

         group ::= Group {
                       numObjects int
                       (object3D | material object3D) ^ numObjects
                   }

      object3D ::= group | transform | sphere | plane |
                   triangle | triangleMesh

     transform ::= Transform {
                       transformation*
                       object3D
                   }

transformation ::= Translate { Vec3f }   |
                   Scale { Vec3f }   |
                   XRotate { float }   |
                   YRotate { float }   |
                   ZRotate { float }   |
                   Rotate { Vec3f float }   |
                   Matrix { float ^ 16 }   |

        sphere ::= Sphere {
                       center Vec3f
                       radius float
                   }

         plane ::= Plane {
                       normal Vec3f
                       offset float
                   }

      triangle ::= Triangle {
                       vertex0 Vec3f
                       vertex1 Vec3f
                       vertex2 Vec3f
                   }
```

```
triangleMesh  ::=  TriangleMesh {
                       obj_file string
                   }

    material  ::=  Material {
                       diffuse_color Vec3f
                   }

       Vec3f  ::=  float float float
```

How to read a grammar:

Start with the top level rule, in this case **`file`**. To expand a rule, substitute the appropriate definitions for each of the elements in bold. Some special notation:

|	means "OR"
^ *n*	means "exactly *n* elements"
*	means "*zero or more* elements"