

6.837 Intro to Computer Graphics, Fall 2003

Assignment 3: Ray Tracing and Phong Materials

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 | pointLight
```

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

```
pointLight ::= PointLight {  
    position Vec3f  
    color Vec3f  
    ( attenuation float float float )+  
}
```

```

materials ::= Materials {
    numMaterials int
    material ^ numMaterials
}

material ::= phongMaterial

phongMaterial ::= PhongMaterial {
    diffuseColor Vec3f
    ( specularColor Vec3f )+
    ( exponent float )+
    ( transparentColor Vec3f )+
    ( reflectiveColor Vec3f )+
    ( indexOfRefraction float )+
}

background ::= Background {
    color Vec3f
    ambientLight Vec3f
}

group ::= Group {
    numObjects int
    (object3D | materialIndex 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
            }

materialIndex ::= MaterialIndex int

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"

+ means "*zero or one* elements"