XML Syntax and Scene Definition Language

Table of Contents

XML Syntax

The scene definition language aims to define objects such as "scene", "camera", "sphere", etc. The scene definition language syntax is in XML format which is an extremely popular format for representing documents. In XML, elements are defined between '<' '>' clauses where the first word is the element's name. Each element has a list of attributes which is basically a list of key-value pairs. XML can define a hierarchy of elements. In our case there is only a single root element with multiple direct children. The root element is the scene element which contains all the other elements. A valid scene definition file must have at least one scene element and one camera element. For a given object some attributes are required and some are optional and receive some default value (determined in the application).

Basic XML example

```
<scene
      background-col="0.3 0.5 0.74"
      ambient-light="1 1 1"
>

      <camera
      eye="0 0 0"
      direction="0 0 -1"
      screen-width="2"
      screen-dist="1"
      up-direction="0 1 0"
      />


      <sphere
      center="0 0 -2"
      radius="0.5"
      />

      <MyNewObject
      MyProperty="7"
      />

</scene>
```

XML Parser further explanation

1. Properties are saved from the XML to a type **Map<String, String>**, we call them **attributes** in the code. The **key** of the map's element is a string which is the property name and the **value** is also a string.

   For example:

   XML file:

   `center = "0.25 -0.25 -1"`

   the property is "center" and the value is `"0.25 -0.25 -1"`

2. The **SceneDescriptor** class 3 members:
   - `protected Map<String, String> sceneAttributes` – the attributes written in the beginning of the XML just below the <scene. They are supposed to hold algorithmic parameters, or scene basic properties that you want. In the example: `background-col, ambient-light`
   - `protected Map<String, String> cameraAttributes` – the attriburtes written inside the <camera part in the XML. In the example `eye, direction, screen-width, screen-dist, up- direction`
   - `protected List<Element> objects` – all other independent parts in the XML which will be read into a list of objects like lights, surfaces and whatever you decide. In the example `sphere,MyNewObject.` These should be stored on classes that inherit from **IInitiable** as well!

3. For your convenience you can turn string to numbers by

```
import java.util.Scanner;
...
String v = "0 1 2";
Scanner s = new Scanner(v);
          x = s.nextDouble();
          y = s.nextDouble();
          z = s.nextDouble();
          s.close();
```

Scene Definition Language

Scene

This element defines global parameters about the scene and its rendering.

- background-col - (rgb) color of the background. default = (0,0,0)
- background-tex - (string) texture file for the background. default = null. Just a filename (see note above)
- max-recursion-level – (number) limits the number of recursive rays (recursion depth) when calculating reflections. Default = 10.
- ambient-light - (rgb) intensity of the ambient light in the scene. I_{AL} from the lecture notes. default = (0,0,0)
- super-samp-width [bonus] - (number) controls how fine is the super sampling grid. If this value is N then for every pixel an N*N grid should be sampled, producing N*N sample points, for every pixel, which are averaged. default = 1. This number is truncated to an integer.
- use-acceleration [bonus] - (number) should be 0 or 1. Enable (1) or disable (0). When this parameter is 1 in a complex scene there should be a significant increase in performance. default = 0. If you implement this, you should also supply a scene XML to demonstrate that it works.

Camera

- eye - (3d coord) the position of the camera pinhole. Rays originate from this point. p_0 point from the slides.
- direction - (vector) the explicit direction the camera is pointed at. towards vector from the slides.
- look-at - (3d coord) this point along with the eye point can implicitly set the camera direction.
- up-direction - (vector) the "up" direction of the camera. up vector from the slides.
- screen-dist - (number) the distance of the screen from the eye, along the direction vector. $dist$ from the slides.
- screen-width - (number) the width of the screen, (in scene coordinates of course) this in effect controls the opening angle of the camera (frustum). $width$ from the slides. default = 2.0

Lights

There can be multiple sources of light in a scene, each with its own intensity color, each emitting light and causing shadows. All light objects can take the following parameter:

- color - (rgb) the intensity of the light. I_0 from the slides. default = (1,1,1) - white.

dir-light: A light emitted from infinity with parallel rays

- direction - (3d coord) the direction in which the light shines at.

omni-light: A light emitted from a single point in all directions

- pos - (3d coord) the point where the light emanates from.
- attenuation - (3 numbers- k_c, k_l, k_q) the attenuation of the light as described in the slides. Default = (1,0,0) - no attenuation, only constant factor.

spot-light: A light emitted from a single point in a specific direction

- pos - (3d coord) the point where the light emanates from.
- direction - (3d coord) the main direction in which the light shines at.
- attenuation - (3 numbers- k_c, k_l, k_q) the attenuation of the light as described in the slides. Default = (1,0,0) - no attenuation, only constant factor.

Surfaces

Every surface has a material. In this exercise we model materials as just a flat color (you are welcome to do textures for extra bonus points). The following parameters may occur on every type of surface:

- mtl-diffuse - (rgb) the diffuse part of a flat material (K_D) default = (0.7, 0.7, 0.7)
- mtl-specular - (rgb) the specular part of the material (K_S) default = (1, 1, 1)
- mtl-ambient - (rgb) the ambient part of the material (K_A) default = (0.1, 0.1, 0.1)
- mtl-emission - (rgb) the emission part of the material (I_E) default = (0, 0, 0)
- mtl-shininess - (number) the power of the $(V \cdot R)$ in the formula (n). default = 100
- reflectance - (number) the reflectance coefficient of the material. K_S from slides. default = 0 (no reflectance).

Example for a Geometric Primitive

disc: A disc is defined by its center, radius and normal to its plane.

- center - (3d coord)
- radius - (number)

- normal – (3d coord)