Raja User Manual

by Emmanuel Fleury, Grégoire Sutre

February 25, 2001

The Raja Project

Contents

Introduction

About this document

This document explain how to make a picture with raja. It doesn't describe the code or the implementation of raja, so if you want some of these informations just take a look at the Developper Manual.

We first give a first taste of raja with a very simple scene. Then we explain basically lights, basic forms (union, intersection, complement), mirors and transparency. Finaly we talk about cameras.

Why in Java???

Why not! Actually, Java2 is the only object oriented language which provide a such complete API. Moreover, this language is more close of object oriented programming than C++. That's why we decided to code RAJA in Java. Recoding RAJA in C++ would be quite difficult because we strongly use the Java2 API and some others libraries. But, if somebody want to do it, we would support him.

Chapter 1

Hello World!

This chapter is a (very) quick survey of Raja.

1.1 The Scene Description

We describe here a simple scene which only contains a sphere and one light.

A Scene is described by a `Solid` and a list of `Light`. As shown on the figure 1.1, A `Solid` can be a `BasicSolid` or an `Aggregate`. The list of `Light` is composed of `PointLightSource` and/or `DirectionalLightSource`.

The objects `Point3D`, `Vector3D` and `RGB` are basic objects. They are only described by `double` values and not of other objects.

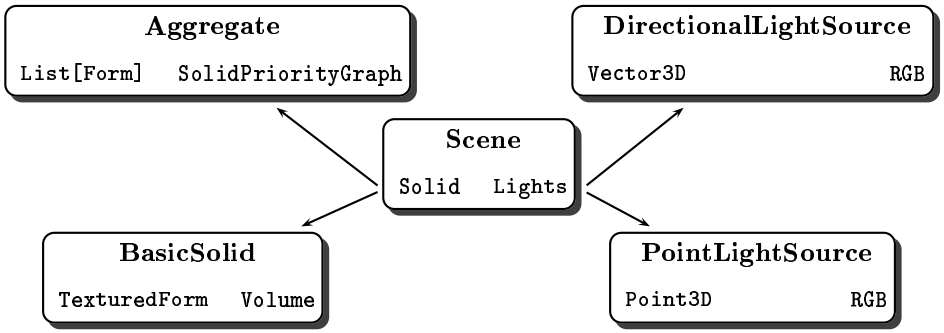

Figure 1.1: *Scene definition.*

The most important (and complex) object of a scene is `BasicSolid`. As shown on the figure 1.2, a `BasicSolid` is defined by a `TexturedForm` that can be a `BasicTexturedForm`, a `Complement`, an `Intersection`, an `Union` or an `Aggregate`. And by a `Texture`.

Let's play with some examples to understand better...

1.1.1 The Plane

A plane is defined by an `origin` and a `normal`. As we said before, this plane is a `BasicSolid` such that his `BasicTexturedForm` is a `Plane`. The `Texture` is defined by seven parameters:

- `kd`: Diffusion coefficient,
- `krl`: Local reflection coefficient, also called specular reflection,

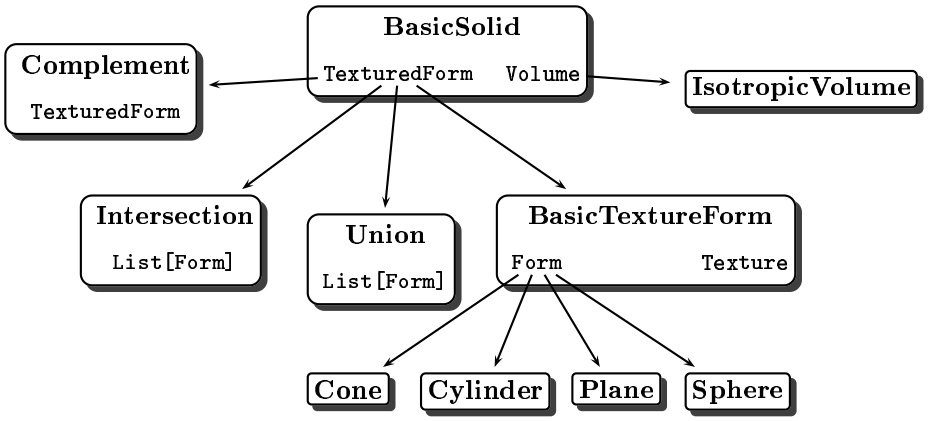

Figure 1.2: *`BasicSolid` definition.*

- `krg`: Global reflection coefficient, it represent the reflected light coming from other objects,
- `ktl`: Local refraction coefficient,
- `ktg`: Global refraction coefficient,
- `ns`: Specular exponent,
- `nt`: Refractive exponent.

For the moment, we can only define an `IsotropicVolume` with a given refraction coefficient (`0.0` in this example).

```
raja.shape.BasicSolid {
        texturedForm = raja.shape.BasicTexturedForm {
                form = raja.shape.Plane {
                        origin = raja.Point3D { 0.0 ; 0.0 ; -15.0 }
                        normal = raja.Vector3D { 0.0 ; 0.0 ; 1.0 }
                }
                 texture = raja.shape.PlainTexture {
                        localTexture = raja.shape.LocalTexture {
                                kd = raja.RGB { 0.6 ; 0.6 ; 0.6 }
                                krl = raja.RGB (@black) { 0.0 ; 0.0 ; 0.0 }
                                krg = @black
                                ktl = @black
                                ktg = @black
                                ns = 10
                                nt = 0
                        }
                }
        }
        volume = raja.shape.IsotropicVolume { 0.0 }
},
```

1.1.2 The Sphere

Similarly, we define the `Sphere` as a `BasicSolid` describe by a center and a radius.

```
raja.shape.BasicSolid {
        texturedForm = raja.shape.BasicTexturedForm {
```

```
                form = raja.shape.Sphere {
                        center = raja.Point3D { 70.0 ; 0.0 ; 6.0 }
                        radius = 20.0
                }
                texture = raja.shape.PlainTexture {
                        localTexture = raja.shape.LocalTexture {
                                kd = raja.RGB { 0.6 ; 0.0 ; 0.0 }
                                krl = @black
                                krg = @black
                                ktl = @black
                                ktg = @black
                                ns = 10
                                nt = 0
                        }
                }
        }
        volume = raja.shape.IsotropicVolume { 0.0 }
}
```

1.1.3 The light

We can have two type of light source:

- `PointLightSource`
- `DirectionalLightSource`

For the example we only describe `PointLightSource` (but you can extend that for the `DirectionalLightSource`).

```
raja.light.PointLightSource {
        origin = raja.Point3D { 30.0 ; 0.0 ; 45.0 }
        light =  raja.RGB { 1.0 ; 1.0 ; 1.0 }
}
```

1.1.4 The Complete Scene

We add some parameters to define:

- `backgroundLight`: The background light,
- `ambiantLight`: The ambient light,
- `ambiantVolume`: The refractive index of this scene.

```
raja.renderer.Scene {
    solid = raja.shape.Aggregate {
         solids = [
             raja.shape.BasicSolid {
                 texturedForm = raja.shape.BasicTexturedForm {
                     form = raja.shape.Plane {
                          origin = raja.Point3D { 0.0 ; 0.0 ; -15.0 }
                          normal = raja.Vector3D { 0.0 ; 0.0 ; 1.0 }
                      }
                      texture = raja.shape.PlainTexture {
                          localTexture = raja.shape.LocalTexture {
                              kd = raja.RGB { 0.6 ; 0.6 ; 0.6 }
```

```
                            krl = raja.RGB (@black) { 0.0 ; 0.0 ; 0.0 }
                            krg = @black
                            ktl = @black
                            ktg = @black
                            ns = 10
                            nt = 0
                        }
                    }
                }
                volume = raja.shape.IsotropicVolume { 0.0 }
            },
            raja.shape.BasicSolid {
                texturedForm = raja.shape.BasicTexturedForm {
                    form = raja.shape.Sphere {
                         center = raja.Point3D { 70.0 ; 0.0 ; 6.0 }
                         radius = 20.0
                    }
                    texture = raja.shape.PlainTexture {
                        localTexture = raja.shape.LocalTexture {
                            kd = raja.RGB { 0.6 ; 0.0 ; 0.0 }
                            krl = @black
                            krg = @black
                            ktl = @black
                            ktg = @black
                            ns = 10
                            nt = 0
                       }
                   }
               }
            volume = raja.shape.IsotropicVolume { 0.0 }
            }
         ]
         priorities = [[
         ]]
    }
    lights = [
        raja.light.PointLightSource {
            origin = raja.Point3D { 30.0 ; 0.0 ; 45.0 }
            light =  raja.RGB { 1.0 ; 1.0 ; 1.0 }
            }
        ]
    backgroundLight = raja.RGB { 0.0 ; 0.6 ; 1.0 }
    ambiantLight = raja.RGB { 0.1 ; 0.1 ; 0.1 }
    ambiantVolume = raja.shape.IsotropicVolume { 1.0 }
}
```

1.2 The Camera

```
raja.renderer.HorizontalCamera {
        origin = raja.Point3D { 0.0 ; 0.0 ; 0.0 }
        direction = raja.Vector3D { 1.0 ; 0.0 ; 0.0 }
        focal = 1.8
        screenWidth = 2.0
```

```
        screenHeight = 1.5
}
```

1.3 The command line

1.4 The GUI

Chapter 2

Basic Forms

2.1 Sphere

2.2 Plane

2.3 Cone

2.4 Cylinder

Chapter 3

Composiste Forms

3.1 Union

3.2 Intersection

3.3 Complement

Chapter 4

Mirors and Transparency

4.1 Mirors

4.2 Transparency

Chapter 5

Lights

5.1 RGB HowTo

5.2 Light Sources

Chapter 6

Camera

6.1 Horizontal Camera