B4 - Object-Oriented Programming

B-OOP-400

Raytracer

How to create your configuration files

Raytracer

binary name: raytracer
build tool: CMakeLists.txt

- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

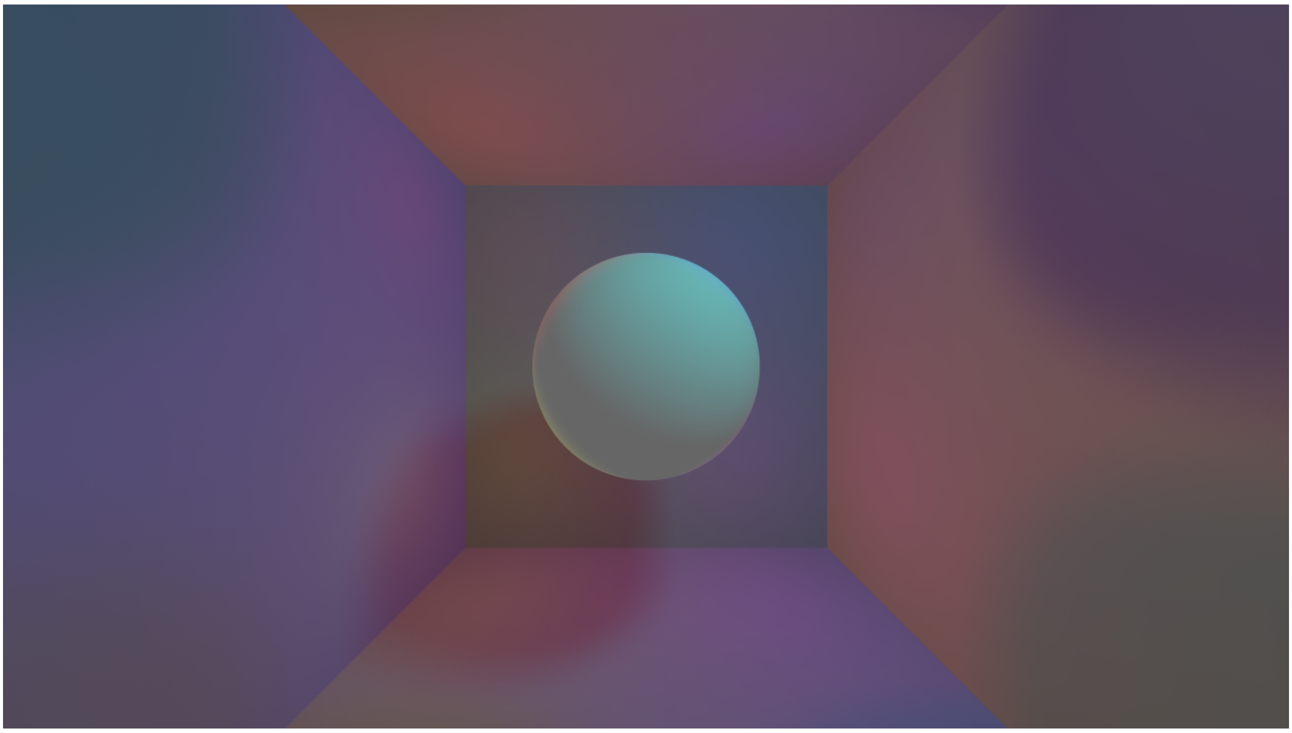

Figure 1: Raytracer_image

How to create your configuration files

What is a configuration file ?

A configuration file is a file that contains the parameters of a program. It is used to store the parameters of a program in a file so that they can be easily modified. If you want to create your own scene with your own objects, you will have to create a configuration file. The file is parsed by the program and the objects are created according to the parameters in the file with the libconfig++ library.

Why create a configuration file ?

For our Raytracer project we had to implement lights (ambient light, directional light), primitives (sphere, plane, cone…) and materials (flat color…) in dynamic libraries (.so) to be able to create scenes and create objects in this scenes with configuration files. So the program will read the configuration files in order to create the scenes with the objects in a dynamic way. If you want to add a new object, you just have to create a new dynamic library and add it to the configuration file.

Create your configuration files for the Raytracer project

First step

First you will have to create a file with the extension (.cfg). This file will contain the parameters of your scene and the objects you want to create in this scene.

Second step

Set the camera parameters in the file. You will have to set the position of the camera, the direction of the camera, the field of view and the resolution of the image.

```
camera:
{
    origin = [4, 8, 4];
    lookAt = [0, 0, 0];
    up = [0, 1, 0];
    vfov = 90;
    aspectWidth = 16;
    aspectHeight = 9;
};
```

Third step

Set the parameters of the renderer in the file. You will have to set the type of renderer (ppmrenderer, sfml-renderer), the name of the file, the width and the height of the image.

```
renderer:
{
    type = "ppmrenderer";
    filename = "test.ppm";
    width = 854;
    height = 480;
    maxDepth = 4;
    samplesPerPixel = 1;
};
```

samplePerPixel is for anti-aliasing.

Fourth step

Setup the materials in the file. You will have to set the type of material (flatcolor, phong, lambertian), the color of the material and the parameters of the material.

```
materials = (
    { name = "redmat"; type = "flat_color"; color = [1, 0, 0]; },
    { name = "greenmat"; type = "flat_color"; color = [0, 1, 0]; },
    { name = "bluemat"; type = "flat_color"; color = [0, 0, 1]; }
);
```

For the color you will have to set the color in RGB format. For example, if you want to set the color red you will have to set the color to [1, 0, 0].

Fifth step

Setup the lights in the file. You will have to set the type of light (ambient, directional), the color of the light and the parameters of the light.

```
lights = (
    { type = "ambient_light"; color = [1, 1, 1]; intensity = 0.4; },
    { type = "directional_light"; direction = [0, 1, -1]; color = [1, 1, 1];
        intensity = 0.6; }
);
```

Sixth step

Finally, you will have to setup the objects in the file. You will have to set the type of object (sphere, plane, cone...), the material of the object, the position of the object and the parameters of the object.

```
objects = (
    { type = "cone"; position = [0, 0, -2]; radius = 2; material = "greenmat"; height
        = 3; rotation = [45, 6, 4];},
    { type = "cone"; position = [0, 0, 2]; radius = 1; material = "redmat";  height =
        5; translation = [0, 1, 1];},
);
```

You can also add a rotation or a translation to the object.
Is here you can add some variables to create your own object if you want to add more shared library.
How to create your own shared library

PROJECT CARRIED OUT BY:

- Lucas HAUSZLER
- Axel IDOUX
- Thibault GUYONY
- Bastien GERARD