

# B2- C Graphical Programming

B-MUL-151

## Raytracer 2

3D Layout Engine

v1.2



# Raytracer 2

## 3D Layout Engine

---

**binary name:** raytracer2  
**repository name:** raytracer2  
**repository rights:** ramassage-tek  
    **language:** C  
    **group size:** 3-4  
**compilation:** via Makefile, including re, clean and fclean rules

---



- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- 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).



This project contains a bonus section, which includes an exhaustive list of all bonuses you may implement in addition to the other mandatory sections.

Raytracer2 consists of completing a program that generates a 3D image.

Raytracer2's mandatory section is made up of content from Raytracer1, as follows:

- 3D objects' drawing: Sphere, Plane, Cylinder and Cone,
- the possibility of transferring and turning the objects in every direction,
- the possibility of transferring and turning the camera in every direction,
- light,
- shadows, with at least one light source.

The difference between Raytracer 1 and Raytracer 2 is that in Raytracer 2, having only these effects will give you a grade of -5. Therefore, with a perfect presentation, you could reach 0.

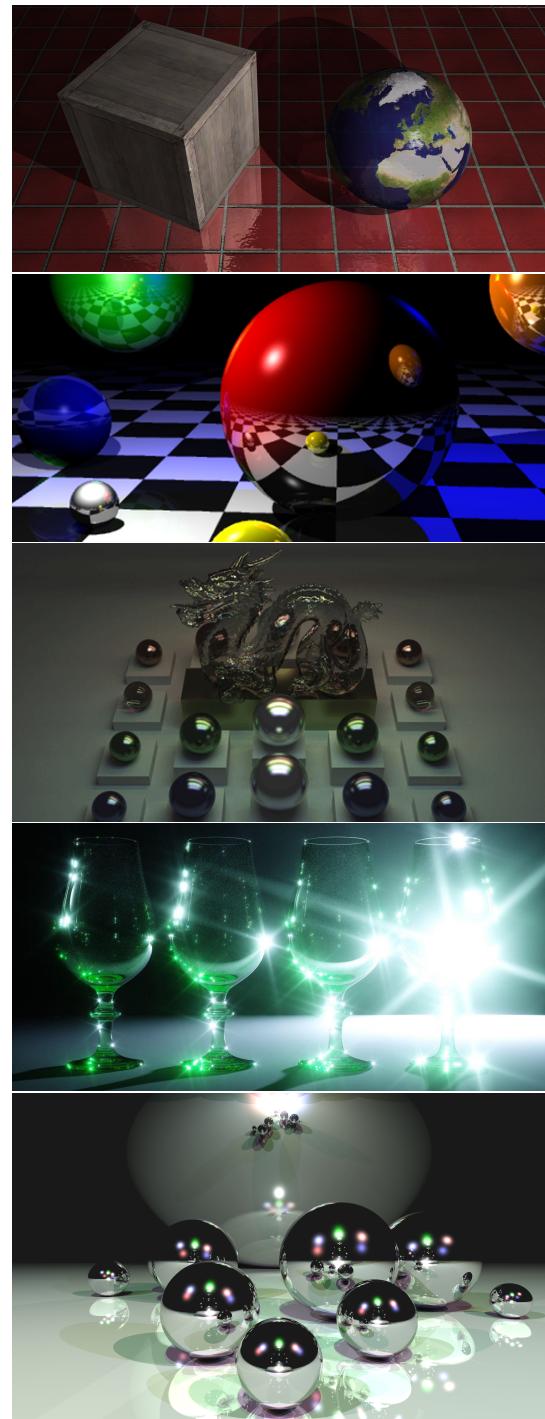
Please note that oversights in the mandatory section result in heavy penalties.

In order to earn a respectable grade, you have to add some features to your program. Further down, you will find a non-exhaustive list of functionalities: don't stop there.

Implementing is part of the project, as well as planning them and organize the code adequately.

Below is a list of **non-exhaustive** features:

- Atmospheric lighting
- Brightness
- Limited objects
- Composed objects
- Facets system
- Disrupting the normal vector (bump-mapping, ridges)
- Color disruption (Checkerboard, Perlin noise,...)
- Direct light
- Parallel light
- Reflection
- Transparency
- Diffraction
- Réfraction
- Focal range
- Anti-aliasing
- Brightness/reflection/transparncy disruptions
- Procedural textures
- Textures from a file
- Negative objects
- Textured cutting out with the help of transparency
- Exotic objects:
  - Layers
  - Torus
  - Möbius strip
  - Klein bottle
  - Quadrics
- Background image
- Stereoscopy
- Multi-threading
- Multi-process
- Network dispatching
- Runtime transfer
- Rendering export
- 3DSMax, Pauv'Ray, Maya,... file importing
- Equations evaluation
- WYSIWYG scene editor
- Video mode





## Authorized Functions

- C Math library (-lm)
- Pthread library (-lpthread)
- open
- close
- read
- write
- malloc
- free

### CSFML functions:

- sfRenderWindow\_isOpen
- sfRenderWindow\_pollEvent
- sfRenderWindow\_waitEvent
- sfRenderWindow\_clear
- sfRenderWindow\_drawSprite
- sfRenderWindow\_display
- sfRenderWindow\_create
- sfRenderWindow\_destroy
- sfRenderWindow\_close
- sfTexture\_create
- sfTexture\_updateFromPixels
- sfTexture\_destroy
- sfSprite\_create
- sfSprite\_setTexture
- sfSprite\_destroy
- all of System module's functions
- all of Window module's functions
- all of Audio module's functions