Chapter IV

General Instructions

- This project will be corrected by humans only. You're allowed to organise and name your files as you see fit, but you must follow the following rules.
- The executable file must be named `RTv1`.
- Your `Makefile` must compile the project and must contain the usual rules. It must recompile and re-link the program only if necessary.
- If you are clever, you will use your library for your `RTv1`. Submit also your folder `libft` including its own `Makefile` at the root of your repository. Your `Makefile` will have to compile the library, and then compile your project.
- Your project must be written in accordance with the Norm. Only norminette is authoritative.
- You have to handle errors carefully. In no way can your program quit in an unexpected manner (Segmentation fault, bus error, double free, etc).
- Your program cannot have memory leaks.
- You'll have to submit a file called `author` containing your username followed by a '\n' at the root of your repository.

```
$>cat -e author
xlogin$
```

- You can use `MacOS` native `MinilibX` library already installed on the imacs, or you can use MinilibX from its sources that you'll need to integrate similarly to libft. Last option, you can use additional graphic libraries (`X11`, `SDL`, etc...). If the library you are using is not installed on the imacs, you will have to submit the sources of this library in your repository, and it will have to be automatically compiled, without doing anything more than compiling your project, exactly like `MinilibX` or like your `libft`. No matter which graphic library you can only use its basic drawings functions similar to `MinilibX`: Open a window, lit a pixel and manage events.

- Within the mandatory part, you are allowed to use only the following libc functions:
    - `open`
    - `read`
    - `write`
    - `close`
    - `malloc`
    - `free`
    - `perror`
    - `strerror`
    - `exit`
    - All functions of the math library (`-lm` man `man 3 math`)
    - All functions of the `MinilibX` or their equivalent in another graphic library.
- You are allowed to use other functions or other librairies to complete the bonus part as long as their use is justified during your defense. Be smart!
- You can ask your questions on the forum, on slack...

Chapter V

Mandatory part

Your goal is to be able, with the help of your program, to generate images according to Raytracing protocol.

Those computer generated images will each represent a scene, as seen from a specific angle and position, defined by simple geometric objects, and each with its own lighting system.

- Implement the Ray-Tracing method to create a computer generated image.
- You need at least 4 simple geometric objects as a base (not composed): plane, sphere, cylinder and cone.
- Your program must be able to apply translation and rotation transformation to objects before displaying them. For example a sphere declared at `(0, 0, 0)` must be able to successfully translate to `(42, 42, 42)`.
- Manage the redraw view without recalculating (basically with MinilibX, you can manage the expose properly): if a part of the window needs to be redrawn, it is best if you don't have to recalculate everything.
- Position and direction of any point of vision and of simple objects.
- Smallest light management : different brightness, shadows.

For the defence, it would be ideal for you to have a whole set of scenes with the focus on what is functional, facilitating that way the control of the elements to create. For example:

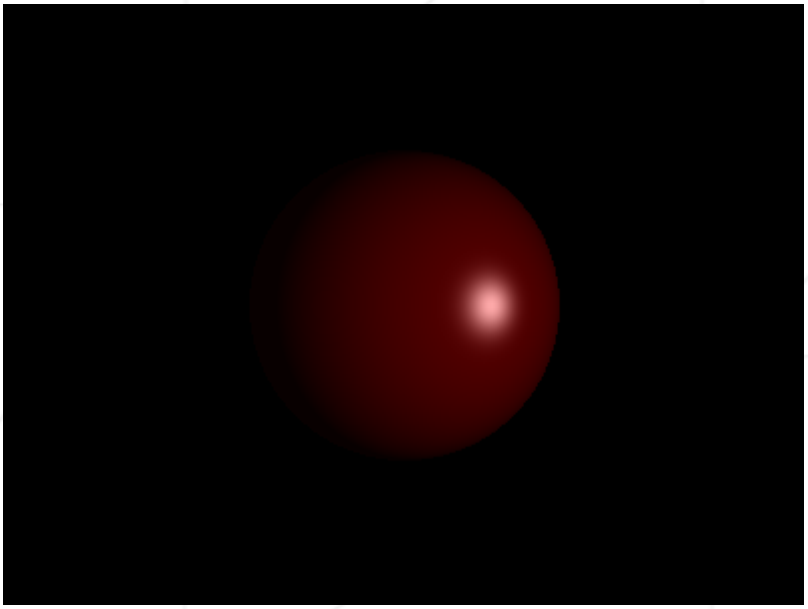

Figure V.1: A sphere, one spot, some shine (optional)

Figure V.2: A cylinder, one spot

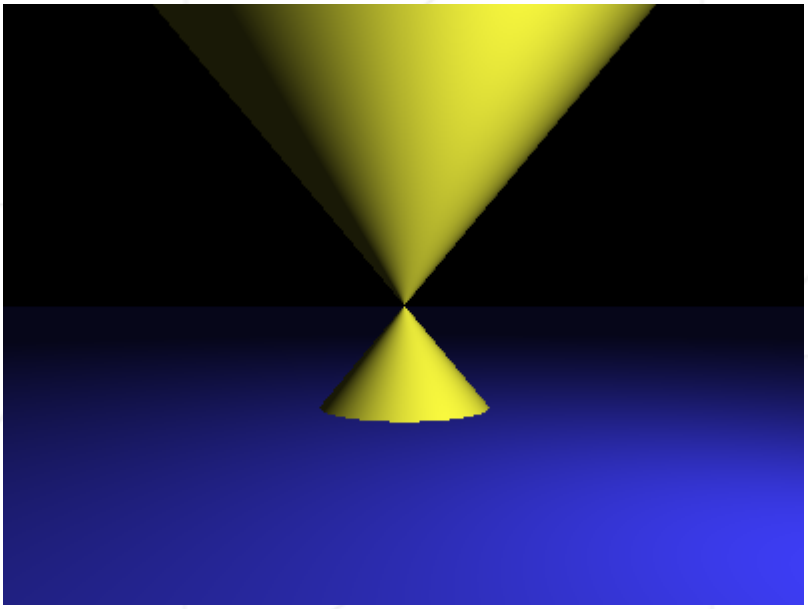

Figure V.3: A cone, a plane, one spot

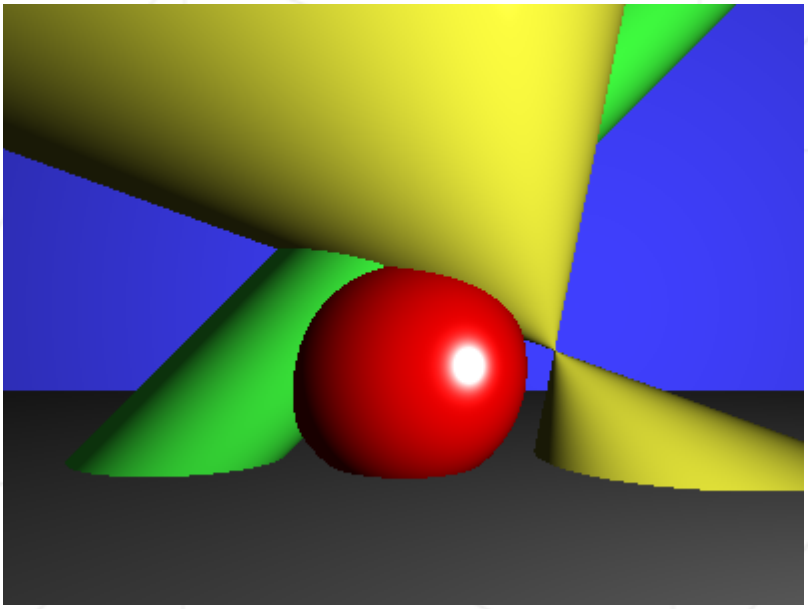

Figure V.4: A bit of everything, including 2 planes

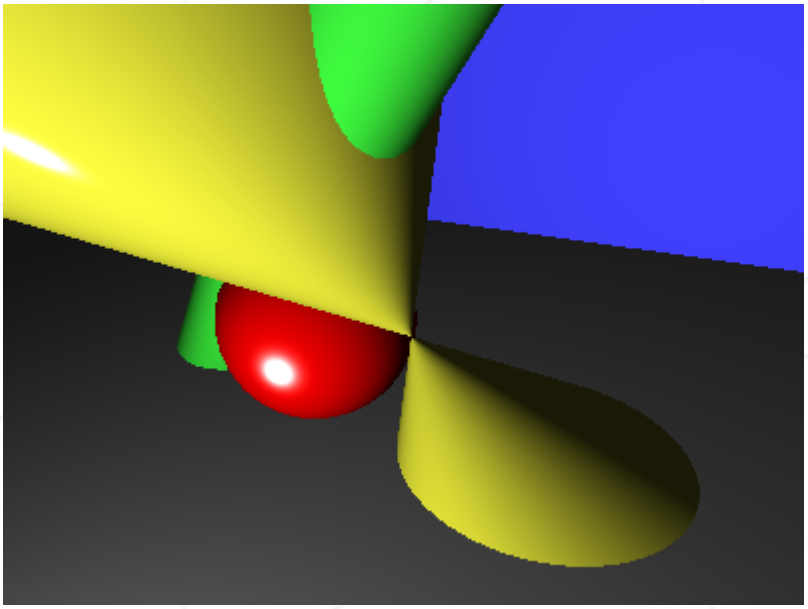

Figure V.5: Same scene different viewpoint

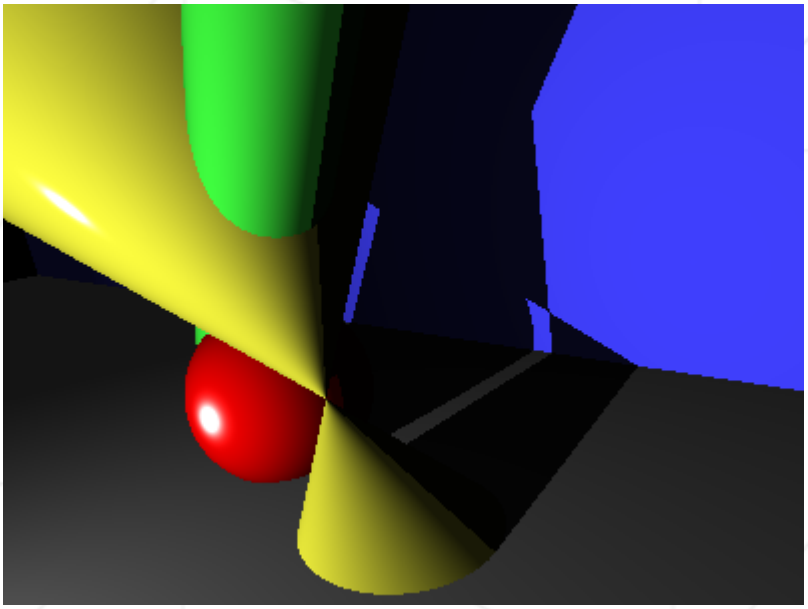

Figure V.6: This time with shadows

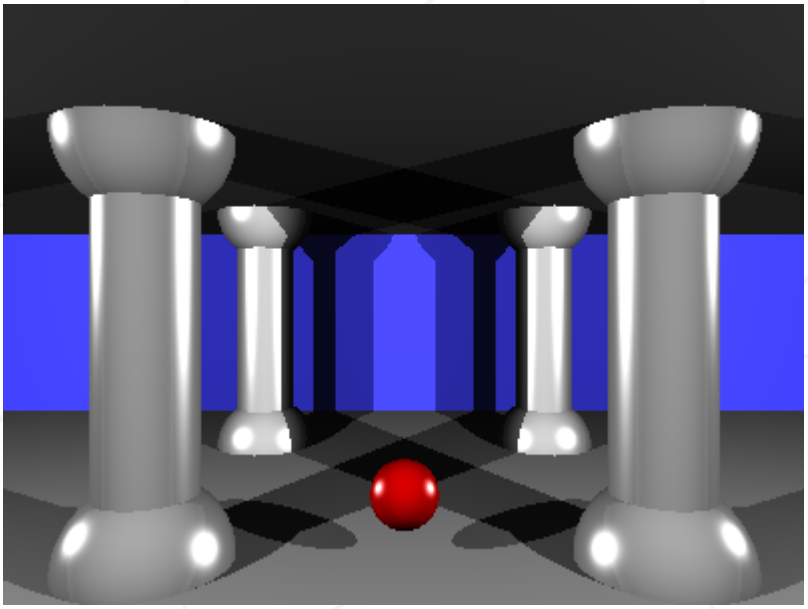

Figure V.7: And finally with multiple spots

Chapter VI

Bonus part

We will look at your bonuses if and only if your mandatory part is EXCELLENT. This means that your must complete the mandatory part, beginning to end, and your error management must be flawless, even in cases of twisted or bad usage. If that's not the case, your bonuses will be totally IGNORED.

Possible bonuses :

- Multi-spots
- Shine effect

No other bonuses will be accepted as they'll be part of the next project: RT.