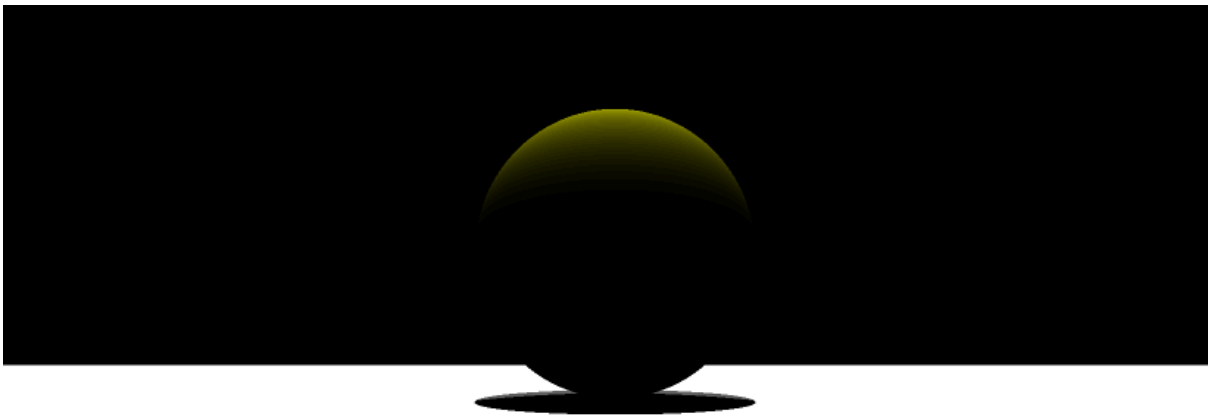
Raytracer

Technical documentation



Made by :

- Axel Fradet **axel.fradet@epitech.eu**
- Mathieu Robert **mathieu1.robert@epitech.eu**
- Kylian Tranchet **kylian.tranchet@epitech.eu**
- Théophile Jérôme-Rocher **theophile.jerome-rocher@epitech.eu**
    - **Promotion Epitech Nantes 2027. 2nd year.**

Useful links (technical):

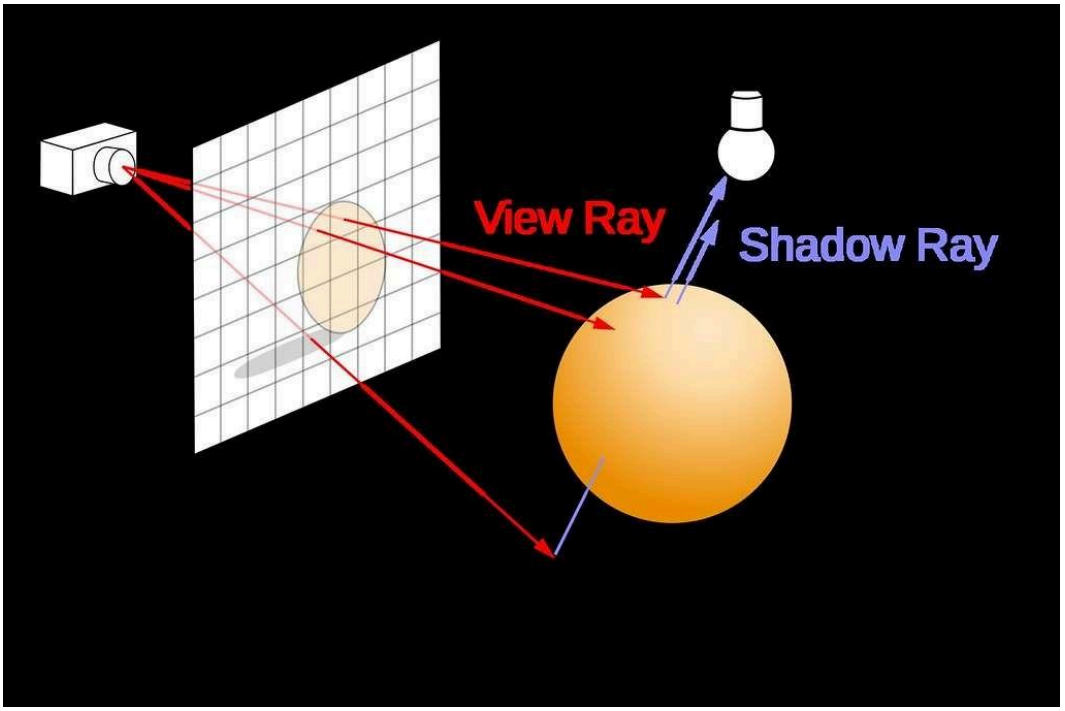
- Github : https://github.com/Njord201/Raytracer
- Github Project : https://github.com/users/Njord201/projects/7
- Commits norm : https://www.conventionalcommits.org/en/v1.0.0/
- Octree Algorithm Documentation : https://www.geeksforgeeks.org/octree-insertion-and-searching/

Execution:

The root makefile allows you to make, make re, make fclean, make clean, make doc.

Technical brief:

For this Raytracing, we simply have a ray (origin + vector) emitted by a camera, passing through a "rectangle", and depending on whether this ray intersects with a primitive, we put the computed color on the rectangle. Simple vector calculations, intersection calculations...



Technical brief on optimizations:

To optimize image generation, we had to implement an algorithm. In this case, the "Octree" algorithm (cf. useful links). We also have anti-aliasing by supersampling to get smoother renders.

Technologies:

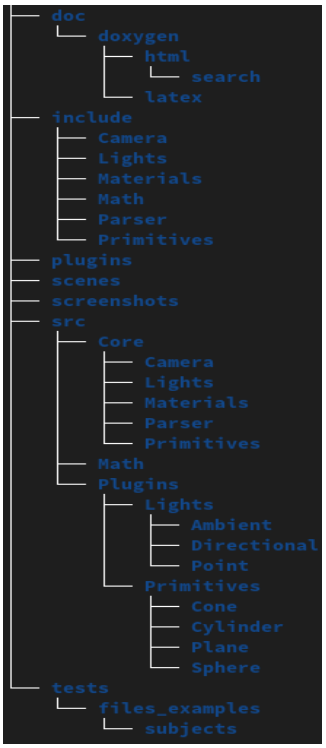
- C++ (Language), Make (Tool), SDL2 (Graphical Interface)
- For the documentation: Doxygen, LaTeX, GraphViz
    - Run “**sudo yum install doxygen doxygen-latex graphviz"**

Developer constraints:

- Use the “conventional commits” norm.
- Use Camel case norm.
- Open issues, create a branch from the issue, then push on main only through pull requests. Minimum 2 reviews approved.
- Rigorous coding (standard / cleanliness).
- Follow the current’s projet architecture.
- Use and update the Github’s project.

Small brief on the structure:

- Primitives, lights, etc. must be "plugins", i.e .so or shared objects, which are then re-injected when loaded. We now have a Makefile compiling the Core and plugins in two steps, each with its own Makefile (makefile in Plugins/ or Core/)...



include/ all the .hpp files.

src/ all the .cpp files.

scenes/ contains .cfg files for the scenes used with Raytracer.

screenshots/ some screenshots taken by the developers.

plugins/ the compiles shared objects .so plugins.

tests/ contains tests.

doc/ the documentation.

Add a new primitive (developers):

- Create a new primitive in include/Primitives, following the logic of existing primitives derived from the IPrimitive interface.
- Same for the C++ code in src/Plugins/Primitives.
- Don’t forget to update the Makefiles with the right references.

Access advanced technical documentation:

If you would like more information on the classes and structures used in our program. You can access **refman.pdf** in **/doc**. You can run "**make doc**" to re-generate the documentation, this pdf and also **/doc/doxygen/html** which will later contain a site you can launch to access the Raytracer doxygen documentation, featuring graphics.