Data Visualitation Project : Ray Tracing with VTK

Aymeric MILLAN & Arthur VIENS

Lecture given by Julien Wintz

11th of March 2022

Abstract

This documents contains the presentation of our work on implementing a RayTracing algorithm using the VTK library and Python's bindings.

First, we are going to do a brief sate of the art on raytracing, then we will explain our implementation logic and technical choices. To conclude, we will show some images rendered by our application.

Difficultés rencontrées :

Learning VTK's logic and syntax was a difficulty

1 Running the code

The source code is available on Mines' GitLab server. You can use the `environment.yml` file to setup your Python environment with conda.

To run the app, just launch the `main.py` file and let the UI guide you !

2 State of the art

Raytracing il ya ca ca ca

2.1 1er truc

Le 1er truc qu'on a trouvé sur l'algo

3 Implementation logic

3.1 Python's VTK bindings

Blabla why we used python, Blabla. Finir sur une phrase qui dit qu'on a utilisé les bindings avec qt..

3.2 User Interface : Qt Designer

We decided to use Qt (with either PySide or PyQt as a wrapper) for the User interface, this way, we don't have to make complex documentation about what keyboard shortcut to use to interact with ou VTK panel.

PICTURE OF SLIDERS

To build the UI we used Qt Designer, which generated a `.ui` file. This file is compiled at the start of our program into a `.py` file which is the corresponding Qt Windows, with all sizes and objects' labels automatically set. We do not have to touch this file once it is generated.This method allowed us to make a more complex UI to give the user more interaction with our VTK environment.

In the code, we had to make a binding between the objects' methods (e.g., `ValueChanged` for a Slider) and Python functions taht interacts directly with our VTK actors and/or sources.

Conclusions

VTK c'est une usine à gaz. C'est fort quand meme on peut render nimporte ou avec nimporte quel DIrectX, OpenGL, etc... Mais la portabilité a un cout : la complexité

Les bindings python sont efficaces

Peut être faire en C++ si cetait a refaire

Toujours plus d'améliorations possible, surtout au niveau e l'interface (bouger les objets avec la souris, image de prérendu avant de save un png, etc...)

Amélioration : Séparer en plusieurs classes python propre au liieu d'un gros fichier