

Data Visualisation Project : Ray Tracing with VTK

Aymeric MILLAN & Arthur VIENS

Lecture given by Julien Wintz

11th of March 2022

Abstract

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

Ray tracing was first introduced in 1980 by Turner Whitted in his paper on [An Improved Illumination Model for Shaded Display](#)

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 our 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 that interact directly with our VTK actors and/or sources.

Conclusions

VTK c'est une usine à gaz. C'est fort quand même on peut rendre n'importe quoi avec n'importe quel DirectX, OpenGL, etc... Mais la portabilité a un coût : la complexité

Les bindings python sont efficaces

Peut être faire en C++ si c'était à refaire

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

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