

CONAV : Ray Tracing

Avis sur le tutoriel

Globalement, ce tutoriel est très bien expliqué : le code est clair, le but de chaque fonction est explicité, les mécanismes physiques qui entrent en jeu sont assez simples à comprendre si l'on a lu le document le cours sur le lancer de rayon au préalable, et l'auteur justifie même les choix qu'il fait. De plus, comme le code est déjà donné dans sa quasi-intégralité, on peut dire que ce tutoriel est très simple.

Peut-être même trop simple, car au bout de quelques heures de code, le manque d'interactivité a tendance à transformer ce tutoriel en un exercice de recopiage du code, sans que ;'on ne cherche trop à comprendre ce qu'il se passe réellement. Je pense que ce document gagnerait beaucoup à faire participer plus son lecteur, en lui demandant de trouver le code correspondant à certaines fonctions simples (par exemple les fonctions de réflexions/réfractions), qui n'augmentent pas tellement la difficulté et qui forcent à lire les explications données entre 2 parties de code.

Changement apportés :

Durant ce tutoriel, j'ai opéré différents changements par rapport au code qui nous est proposé, certaines fois par choix, d'autres par nécessité.

- Dès le chapitre 1, à la place de la fonction « cout » d'iostream, qui nous renvoie chaque pixel dans le terminal, j'ai préféré utilisé la bibliothèque fstream, pour stocker ces données dans un fichier ppm, que je lis ensuite avec IrfanView.
- Dans le chapitre 5, la notation « MAXFLOAT » ne fonctionnait pas pour moi, j'ai utilisé « FLT_MAX » à la place
- Dans le chapitre 6, la fonction « drand48 » est désormais obsolète, je l'ai donc recréée :

```
double drand48() { return (double(rand()) / double(RAND_MAX)); }
```
- Dans le chapitre 8, j'ai dû rajouter la ligne suivante à la fonction « sphere::hit » :

```
rec.mat_ptr = mat_ptr;
```

Ce changement n'étant pas explicité dans le tutoriel, j'ai perdu pas mal de temps pensant avoir fait une faute de frappe dans le reste du code, avant de comprendre la raison de l'exception indiquée par Visual Studio.

Lien vers le GitHub :

<https://github.com/hydrexao/ConavRayTracing>

