



Forum: Moteur de jeu GameBlender et alternatives

Topic: hARMful engine

Subject: Re: hARMful engine

Posté par: Bibi09

Contribution le : 3/6/2020 9:27:51

J'ai pu charger les images dans le moteur mais j'ai encore rien appliqué du tout au rendu. Il me manquait une donnée représentée sous forme de "texture".

C'est un tableau en 2D, et non une image à proprement parler même si pour des raisons de praticité elle est enregistrée comme tel.

En fait, chaque pixel code deux valeurs (canaux rouge et vert) et ces valeurs sont utilisées pour calculer des choses dans les matériaux.

C'est un peu comme une normal map où chaque canal (RGB) code un vecteur (coordonnées XYZ) pour "modifier la surface" d'un objet 3D. En soit, une normalmap n'est pas une image, c'est des coordonnées de vecteurs.

Cette image ne varie pas, une fois qu'elle est calculée on peut l'utiliser. Elle ne dépend pas du tout d'une image HDR, j'aurais très bien pu faire ça dans un nouveau programme mais c'était plus pratique ainsi.

Le programme ne la calcule pas systématiquement, il faut spécifier une option pour le faire. Par défaut, le programme ne la calcule donc pas !

Alors, ça semble tout bête ce truc rouge mais en fait ça demande pas mal de calculs vectoriels, c'est donc assez long. Ici, c'est un calcul sur CPU mais on peut très bien le faire en GPU. J'avais envie de changer.

Le truc rigolo c'est que j'ai fait le rendu en multithreadé à la façon d'un [rendu dans Blender](#). J'ai découpé l'image en tiles de 32x32 pixels et chaque thread fait les rendus case par case. Dans le Gestionnaire de tâches, le CPU est actif à 100% car j'utilise autant de threads que disponibles sur le CPU. C'est un peu sauvage mais j'ai pas envie de mettre 56000 options dans le soft...

J'ai pas souvent l'occasion de faire des programmes qui fonctionnent en multithreading mais je trouve ça toujours très amusant d'imaginer le bousin mouliner à fond quand on l'exécute. Et j'ai pu apprendre des choses en C++ à ce niveau-là.