



Forum: Moteur de jeu GameBlender et alternatives

Topic: Où sont les limites de blender ?

Subject: Re: Où sont les limites de blender ?

Posté par: barbapapa

Contribution le : 24/10/2014 7:59:04

Bonjour.

Je suis assez d'accord avec la plupart des choses qui ont été dites.

- Sur le pathfinding, on est très limité parce qu'il manque quelques steering behaviours (<http://www.red3d.com/cwr/steer/>). De plus, je ne sais pas s'il est possible le système des navmeshes pour ajouter des obstacles en cours de partie. Enfin, le système d'obstacle avoidance (RVO (reciprocal velocity obstacles)) produit des effets bizarres (je n'ai pas trop regardé le code source (blender KX_ObstacleSimulation.cpp, mais il me semble qu'il n'utilise pas la dernière version de la librairie RVO (Je suis pas sûr). Perso je travaille sur un autre système de pathfinding (Bon il sera sûrement pas aussi bien que celui de Blender, et il est en Python, mais il aura l'avantage de permettre l'ajout d'obstacles en temps réel... Enfin, je suis pas sûr que ça ne soit pas déjà possible avec une astuce dans le système de Blender). Donc si tu veux essayer de traduire mon système de pathfinding (qui utilise un algorithme très récent (jump point search) dans le code source de Blender, ça peut être rigolo.

- Sur les filtres GLSL, c'est vrai que ça serait bien d'avoir un set de filtres (et de shaders) un peu plus important à disposition dans Blender. Sinon, je ne sais pas trop où en est le projet Harmony (de Moguri je crois), mais il me semble avoir vu passer un message sur BA il y a pas longtemps, et je crois que ça avance. On attend toujours aussi avec impatience l'intégration shader water/underwater de Martins Upitis, mais si tu veux essayer de l'intégrer dans Blender, ça peut être un travail intéressant, bien que compliqué. Perso, j'ai laissé tomber le GLSL pour le moment par manque de doc compréhensible pour un noob. Il y a aussi le shader sky de Simon Wallner (inclus dans le .blend de démo du shader water/underwater de Martins Upitis), qui peut être bien intéressant à intégrer dans le code source pour de beaux cycles jour/nuit, et sûrement plus facile à implémenter que le shader water/underwater (dans le menu world par exemple :).

- Je suis d'accord avec Hooker95 sur un système de particules qui devrait être intégré à l'interface de Blender. Panzergame a fait un gros travail et une belle démo. J'espère que son travail sera intégré à Blender.

Sinon, le Xmitter de Solarlune (

<http://blenderartists.org/forum/showthread.php?204644-X-Emitter-V1-10-2-11-1-11-BGE-Particle-System-Bugfix>) est vraiment sympa et facile d'usage, mais en python, donc peut-être moins performant qu'une solution intégrée au code source (enfin je sais pas). L'idéal serait de faire marcher la carte graphique pour gérer les particules mais bon, ça doit être sacrément compliqué.

- Sur le support du réseau, Agoose77 travaille en ce moment sur un addon multiplayer. Je ne l'ai jamais testé, mais je lui fait confiance pour pondre un truc intéressant. Sinon, il reste toujours la possibilité de jouer avec le module socket en python, et ça marche plutôt bien d'après les quelques exemples que j'ai vus. Mais bon, c'est vrai que

l'implémentation d'une solution vraiment intégrée au code source de Blender peut être un travail intéressant.

- Sur le threading, je m'en suis déjà servi à plusieurs reprises et je n'ai jamais eu de problème. Pas le module multiprocessing, hein, le module threading. Couplé au module Queue, on peut faire du multitask sans problème (enfin je dis ça parce que je n'ai jamais eu de problème mais je ne dis pas qu'il ne peut pas y en avoir). Par contre, là, je ne vois pas ce qu'on peut faire dans le code source de Blender.? (A quoi penses-tu Bénicourt?)

- Sur la portabilité vers les consoles et les téléphones, c'est vrai que c'est une lacune. Il faudrait trouver quelques développeurs qui connaissent bien à la fois Blender et le fonctionnement de ces machines... Je me demande si pour Unity, ils n'ont pas payé des développeurs professionnels PS3, XBOX, Android, Windows Phone etc... avec les sous qu'ils ont gagné en vendant la version pro d'Unity) pour implémenter l'export des fichiers créés dans Unity vers ces appareils...

Conclusion: Je me rappelle que Matpi disait que quand on commence à toucher au code source de Blender, c'est assez simple d'essayer de faire un filtre 2D. Ca pourrait être un bon départ, j'imagine. Sinon, l'implémentation du shader sky de Simon Wallner pour les cycles jour/nuit (qui peut peut-être être intégré comme un filtre 2D (?) ou dans le menu world (ça serait génial ça) peut être un travail super intéressant et stimulant j'imagine. Mais bon, je n'y connais rien au code source de Blender.

Bonne journée.

EDIT: J'avais pas vu que tu étais en train d'écrire un pavé aussi ebrain. Si j'avais su, j'aurais essayé de faire plus court