



Forum: Moteur de jeu GameBlender et alternatives

Topic: Réflexions sur la démarche de création d'un jeu vidéo

Subject: Réflexions sur la démarche de création d'un jeu vidéo

Posté par: [_tibo_](#)

Contribution le : 2/9/2007 21:31:33

Bonjour à tous,

suite à [cette discussion](#), je crée un sujet où l'on va essayer de mettre en commun ce qu'on sait sur la méthode de création d'un jeu vidéo.

Idéal serait qu'on arrive ensemble à dégager un walkthrough général, à détailler un peu les étapes, et à regrouper des liens vers des tutos pour chaque étape.

J'éditerais le premier post au fur et à mesure, et au final on devrait obtenir un truc assez cohérent pour être condensé dans un tuto

J'attends vos idées/expériences/conseils et tout ça !

1 - Notions clés

Les contraintes du temps réel

Qui dit jeu vidéo, dit dans la majorité des cas 3D en temps réel.

La 3D temps réel s'oppose à la 3D précalculée, où le temps de rendu d'une image peut durer plusieurs heures, voir plusieurs jours. En 3D temps réel, 30 à 50 images devront être affichées chaque seconde pour une bonne fluidité ! Il va sans dire que, pour permettre cette vélocité, bon nombre de sacrifices sont nécessaires.

Modélisation

Moins un modèle a de faces, plus il est léger à afficher.

Lorsque l'on modélise en cherchant à limiter au maximum le nombre de faces, on parle de Low Poly Modeling (LPM).

Pour information :

Un personnage de DOOM 3 contient entre 2000 et 6000 polygones.

Un personnage de WoW contient moins de 2000 polygones, un bâtiment de 1000 à 5000 polygones.

Solid Snake, le héros de Metal Gear Solid 2, contient 4000 polygones.

Un personnage ou monstre de Half-Life 2 contient 7500 polygones environ, un monstre 3000 polygones.

// ces polycounts sont tirés de jeux récents tournant sur des moteurs dédiés optimisés pour l'affichage d'un grand nombre de polygones. Tout dépend du moteur utilisé, mais pour vos créations il faudra généralement viser beaucoup moins !

Pour connaître le nombre de faces d'un objet, sélectionnez cet objet, passez en mode Edit et regardez en haut à droite de votre écran : le champ Fa:x-y indique le nombre de faces de l'objet (y) et le nombre de faces sélectionnées (x).

Attention, Blender compte le nombre de faces, qu'elles soient quadratiques ou triangulaires. Un moteur temps réel affiche que des triangles, donc pour obtenir une donnée significative il faut généralement multiplier par 2 le nombre de faces donné par Blender. N'oubliez pas que si vous utilisez un modifier Mirror pour modéliser, il faudra également multiplier par 2...

Matériaux/Textures

La plupart des moteurs 3D affichent correctement que les textures UVMappées. De plus l'UVMapping permet de contrôler parfaitement l'application des textures sur le modèle, et ainsi d'ajouter facilement des détails, sans augmenter le polycount !

Les moteurs sont généralement plus à l'aise avec les textures en TGA ou PNG, et dont les dimensions sont des puissances de 2 : 128x128, 256x256, 512x512, 1024x1024, etc...

Pensez à profiter du canal Alpha des images PNG et TGA pour créer des formes complexes sur un simple plan : vous réaliserez ainsi des flammes, des poils/cheveux, des bords irréguliers, des herbes/feuilles, sans augmenter le polycount.

Pensez à utiliser les fonctions de **baking**, plus d'informations dans la section *Effets Spéciaux*.

Si de simples textures vous paraissent insuffisantes pour votre jeu, et que vous souhaitez profiter des fonctionnalités des cartes graphiques récentes en ce qui concerne les matériaux, vous pouvez vous pencher sur les *shaders GLSL*. A vous le *normal mapping*, le *toon shading*, et autres effets sympatiques !

Animation

Pour pouvoir être *appelées* par GameBlender lors d'évènements spécifiques, les animations sont enregistrées sous forme d'**Actions**.

Par exemple, appuyer sur une touche du clavier pourra déclencher la lecture d'une action, disons l'action Action:Jump qui correspond à un saut du personnage. Maintenir appuyé une autre touche déclenchera la lecture en boucle d'une action contenant un cycle de marche, etc. Il est donc important de définir toutes les actions possibles par un personnage, afin de constituer une sorte de bibliothèque des mouvements qu'il peut réaliser.

Les **contraintes sur les armatures** (IK Solver, Copy Rotation, etc) ne sont pas prises en compte par GameBlender. Vous pouvez par contre les utiliser pour faciliter la création de vos actions.

Disons que vous utilisez une contrainte de type IKSolver sur la main d'un personnage, pour faciliter l'animation de son bras. Si vous ne placez des clefs d'animation **que** sur le bone de la main, les autres bones du bras suivront le mouvement : c'est le principe d'un IKSolver. Mais si vous lancez l'action dans GameBlender, seule la main bougera ! Pensez donc à placer des clefs d'animation sur **tous** les bones...

Effets spéciaux

Toujours dans une logique de gains de performances, la plupart des "effets spéciaux" réalisables avec Blender seront complètement inutilisables en temps réel. Oubliez tout de suite les simulations de fluide, les softboudys, les systèmes de particules, la radiosit ...

Voici quelques solutions moins gourmandes en ressources :

Utilisez des **plans textur s en Alpha**. Ils permettront de r aliser   moindre frais un grand nombre d'effets int ressants : des cheveux, des bords de tissus d chir s, de l'herbe...

Rien n'emp che d'animer ces textures, pour r aliser des flammes, de la fum e, un drapeau, de l'eau...

Vous pouvez **baker les r sultats du rendu**, c'est   dire transformer en b te texture le r sultat de longues minutes de calcul de radiosit ... Vous pourrez ainsi obtenir facilement des  clairages r alistes sans aucune perte de performance. Evidemment, cet  clairage n'est pas dynamique, mais bien utilis  les r sultats peuvent  tre tr s convaincants.

[:Tuto vid o : Modeling a low poly character](#)

[A very good video tutorial has been released by Montage Studio, showing the entire process for modeling a Lowpoly character with Blender, for using in game projects. The video has a total running time of approximately 50 min, and it's in high quality 1024 x 768.](#)

[Mod lisation d'un visage low-poly](#)

survol de techniques par David Revoy, un artiste de talent, ses oeuvres sont splendides !

[Tutoriel : mod lisation et texturage pour les jeux vid o](#)

Un petit ver de terre LPM, par Batmur

[Making-Off : Styl ](#)

Un exemple simple de personnage low-poly

[Oto the cleaner](#)

Plusieurs tutos concernant le low-poly, l'animation de personnages, le level-design

[Tutoriel : Real Time Character Modeling Tutorial](#)

Un tuto sur la mod lisation d'un personnage low-poly (ou plutot middle-poly)

[Tuto  clair : La radiosit e](#)

[i]Tase explique rapidement le principe de la radiosit 

[Tutoriel : D pliage UV d'une t te](#)

Pas pour Blender, mais le principe reste le m me

[Tutoriel : Bases de l'UVMapping](#)

Pour bien comprendre le principe de l'UVMapping

[Tutoriels : textures avec Gimp](#)

Diff rents tutso pour apprendre   cr er des textures sous Gimp.   appliquer sur vos mod les UVMap s, bien sur !

[Tutoriel : Easy use of GLSL shaders in game engine](#)

"Easy use", c'est vite dit, il n'empêche que si vous voulez profiter des fantastiques effets permis par les GLSL, il va falloir passer par là !

Gestion de projet

PERT, GANTT, travail à distance, serveurs SubVersion...

Gestion de l'équipe

Constituer une équipe, question de la hiérarchie, motivation/dynamisme...

[Nouveau projet : comment le présenter à la communauté ? et pourquoi :](#)

Constituer une équipe dans le cadre d'un projet libre, ça s'apprend !

[ProgRegroup :](#)

Un site qui se propose d'aider les "électrons libres" (des gens compétents dans un domaine qui cherchent une équipe) à trouver le projet qui leur convient, et inversement.

2 - Les phases du projet

I. Phase créative

- L'idée / le concept
- Etude de faisabilité
- Estimations des coûts et temps
- Recherche de documentation
- Début Cahier des charges
- Ecriture d'un scénario + fil conducteur
- Mise en place (papier hein) d'un game play
- CDC définitif
- Création de la bible graphique (story compris)

II. Pre-production

- Analyse technique
- Constitution de l'équipe de production
- Répartition des tâches
- L'organigramme de production
- Recherche de financement

III. Production

- Programmation du moteur
- Mod des premiers meshes et import pour tests dans le moteur
- Mod définitive perso et decors et tout quoi
- Programmation de finition
- Réalisation des cinématiques

IV. Debug et tests

- Beta tests
- Sortie du jeu

3 - Liens en vrac

[Le monde des Ronrons : Journal](#) :

Ce document est une sorte de grand témoignage/making-of/postmortem qui intéressera je pense tout ceux qui se posent des questions sur la création d'un jeu vidéo professionnel. -Christophe Kohler