Anonyme : https://openclassrooms.com/forum/sujet/tout-sur-la-creation-d-un-jeu-3d-63423



* [Signaler](https://openclassrooms.com/forum/sujet/tout-sur-la-creation-d-un-jeu-3d-63423#alertActionModal)
* [Citer](https://openclassrooms.com/forum/sujet/tout-sur-la-creation-d-un-jeu-3d-63423)

[Le 21 mai 2012 à 20:49:59](https://openclassrooms.com/forum/sujet/tout-sur-la-creation-d-un-jeu-3d-63423#message-7388447)

Bonjour à tous !  
  
Depuis maintenant quelques années que je fréquente ce forum, j'ai croisé très régulièrement de nombreuses idées reçues sur les moteurs de jeu, ces environnements facilitant la création d'un jeu vidéo. On peut citer parmi eux :

* Unreal Engine (UDK), un des plus anciens et connus, toujours le leader ;
* Source Engine, le moteur des jeux Valve, enseigné depuis longtemps sur le SdZ ;
* Unity 3D, un moteur générique plus récent ;
* CryEngine, le moteur de Crytek ;
* Et bien d'autres encore.

J'ai décidé de me lancer dans cette sorte de grande FAQ car ces technologies sont finalement assez méconnues, et souvent délaissées par les amateurs au motif qu'ils seraient "trop simples". Ce sujet cherche à tordre le coup à quelques idées reçues et à donner quelques conseils sur la création de jeu vidéo en général.

Je prendrais régulièrement l'exemple d'UDK car c'est le moteur que je connais le mieux : ne prenez pas ça comme de la publicité. Les concurrents cités au dessus sont parfaitement utilisables.

**Idée reçue N°1 : les jeux vidéo, c'est impossible**

La création de jeux 3D, c'est un grand classique, notamment sur le Site du Zéro. Qui n'a jamais rêvé de créer son jeu ?   
Souvent, la première des remarques qui arrive sur un tel sujet, c'est

**Citation : Message typique**

Tu n'arriveras jamais à faire un jeu 3D. Il faut une équipe de 200 personnes et dix ans de travail.

Mais pourtant, c'est faux ! Aujourd'hui, il y a des outils de développement adaptés qui permettent en quelques mois, à une équipe réduite de développer un FPS basique, comme un clone de Counter-Strike par exemple. Bien entendu, ça demande un travail acharné et des compétences solides en programmation, en design, en modélisation, en son et c'est la vraie difficulté : **aucune solution technique ne fera le boulot à votre place.**  
  
En revanche, pourquoi implémenter le N° moteur de rendu 3D de la planète ? C'est un problème ancien qui a déjà été résolu des centaines de fois. Le principe d'un moteur de jeu est de centraliser les problèmes les plus courants pour faciliter le travail :

* Rendu 3D
* Code réseau
* Gestion du son
* Importation 3D
* Création de niveaux

Pour autant, un moteur de jeu typique vous proposera une démo, et un code très basique qui ne gèrera probablement rien de plus qu'un personnage et une arme. C'est encore à vous de créer votre jeu vidéo : simplement, au lieu d'assembler des bibliothèques plus ou moins complètes et compatibles, vous utilisez un environnement intégré.

**Idée reçue N°2 : un jeu vidéo, c'est facile, il suffit de motivation**

**Citation : Message typique**

On est motivés, c'est sûr qu'on va réussir. ;)

Malheureusement, c'est aussi faux que l'extrême précédent. Il faut des compétences tellement variées qu'il est essentiel de se regrouper pour créer un jeu 3D. Listons donc les postes critiques d'un tel projet, sur le plan technique :

* Programmeur(s)
* Modélisateur(s) 3D, level designer
* Artiste son / Compositeur
* Webmaster (eh oui...)

A ces compétences techniques, se rajoute la tâche difficile de gérer le projet. Un tel poste n'est pas concevable sans une expérience basique de tous les domaines cités : ça ne s'improvise pas, ce n'est pas le poste de celui qui ne sait rien faire, ce n'est pas le poste du glandeur, ce n'est pas non plus le chef qui décide de tout. C'est un poste difficile. ;)  
  
Attention à votre recrutement : ne recrutez pas vos amis proches parce qu'ils sont motivés, là tout de suite sur le coup. Recrutez sur la base de l'expérience, de la compétence, des travaux réalisés par le passé : ne recrutez jamais quelqu'un qui n'a jamais fait quelque chose de sérieux avant, de lui-même !

**Idée reçue N°3 : il faut des outils payants pour créer un jeu vidéo**

**Citation : Message typique**

J'ai besoin de 1000€ pour acheter Photoshop.

Dans l'absolu, ce n'est pas complètement faux. La suite Adobe au grand complet peut se révéler un allié de poids, les logiciels 3D de grandes entreprises également. En revanche, surtout si vous débutez, sachez découvrir les solutions gratuites voire libres.   
  
Par exemple, il est parfaitement possible de développer un niveau de jeu vidéo moderne avec Blender et GIMP dans un moteur de jeu tel qu'UDK, qui est lui-même gratuit pour un usage non commercial.  
  
Attention tout de même : ne soyez pas sectaires non plus. Vous vous apercevrez bien vite que ce qui compte, c'est la vitesse de développement et la qualité du travail plus que l'outil lui-même : choisissez avant tout ce qui vous convient !

**Idée reçue N°4 : avec un moteur de jeu, on ne programme pas**

**Citation : Message typique**

UDK/Unity/.. c'est pour les nuls, on ne fait pas de vraie programmation.

C'est probablement le propos le plus courant sur les moteurs de jeu : c'est aussi le plus faux.  
Prenons l'exemple d'Unreal Engine ou UDK : son projet d'exemple est constitué d'environ 400 000 lignes de code. Si vous lisez ces lignes, il y a fort à parier que vous avez déjà programmé, mais quelle quantité de code avez-vous écrit dans votre vie ? Si vous êtes programmeur de métier, probablement plusieurs milliers ou dizaines de milliers... Imaginez dix fois cette quantité : oui, il faut réellement programmer avec un moteur de jeu, et pas qu'un peu.  
  
La même critique est souvent faite sur les langages employés dans ces moteurs, parfois considéré comme des scripts sans complexité technique. Il n'en est rien, voici une liste des langages employés par chaque moteur :

* UDK : UnrealScript, un clone de Java optimisé et réduit
* Source Engine, Cry Engine : C++
* Unity3D : C#, ou Javascript au choix

Ne voyez pas un moteur de jeu comme un logiciel pour créer votre jeu, voyez-le comme un environnement de développement, un laboratoire en quelque sorte, remplis d'outils pratiques.

**Idée reçue N°5 : les moteurs de jeu sont limités**

**Citation : Message typique**

Si tu veux faire un jeu original, il faut tout programmer soi-même avec OpenGL.

Là encore, c'est une erreur courante. Vous avez, plutôt que des limitations, des choix techniques déjà faits à bas niveau qui ne vous laissent pas la liberté de les modifier. En revanche, ces choix n'interfèrent pas avec le gameplay, le développement du jeu vidéo en lui-même, pour plusieurs raisons.

* D'abord, un moteur de jeu est un environnement de développement et pas une machine à faire des jeux : il ne prend pas les décisions à votre place.
* Le code d'un moteur ne s'attaque jamais au gameplay, ce n'est pas sa mission : il s'attaque à des problèmes de bas niveau et implémente des mécanismes basiques comme par exemple la notion de joueur, d'équipe, de score, mais sans jamais en fixer les règles : c'est à vous de les choisir et le moteur ne le fait pas pour vous.
* Plus généralement, le moteur s'arrête à relativement bas niveau. Si vous souhaitez faire de la classe "Joueur" du moteur un véhicule ou un bâtiment, rien ne vous en empêche ! C'est le principe.
* Enfin, si vous vous sentez limité par un aspect du moteur, souvenez-vous que bien souvent il permet de faire appel à du code extérieur, par exemple si vous souhaitez gérer Kinect...

En d'autres termes, ce qui est essentiel c'est de comprendre que vous aurez encore énormément de travail, même avec un moteur de jeu et rien qu'en programmation. Ce n'est pas une solution de **facilité** : c'est une solution **réaliste**.  
  
Certains tendent à catégoriser les moteurs de jeu comme un outil automatique pour faire un clone de Call of Duty. Ne vous y laissez surtout pas prendre, c'est une vision **particulièrement erronée** de la réalité. En particulier, un moteur comme UDK ou Unity ne vous force pas à faire un type de jeu particulier, la seule limite est d'être en 3D : que ça soit un jeu de course, un FPS, une simulation, c'est possible et pas moins adapté dans un cas que dans l'autre.

**Idée reçue N°6 : le moteur de jeu fournit du contenu gratuit, on n'a pas besoin d'en créer soi-même**

Oui et non. Effectivement, un moteur tel qu'UDK propose une base de travail très conséquente et suffisante pour créer un jeu basique. Mais vous ne devez **surtout pas** vous en tenir là, ce n'est pas l'objectif ! Il faut voir ces éléments comme des démonstrations, des aides au développement pour créer votre propre contenu.

[](https://user.oc-static.com/files/355001_356000/355206.jpg)  
  
[](https://user.oc-static.com/files/367001_368000/367728.jpg)  
Le style visuel, c'est ça qui décide de la personnalité de votre jeu

Un jeu vidéo, ce n'est pas juste un programme, c'est une œuvre artistique et ludique et vous devez être bien conscient qu'en utilisant des éléments créés par d'autres, vous enlevez toute âme à votre création. C'est bien dommage !  
Prenez le temps de développer votre contenu, recrutez d'autres créateurs pour vous appuyer.

**Idée reçue N°7 : on peut faire aussi bien qu'avec un moteur de jeu, en faisant tout soi-même**

**Citation : Message typique**

On peut faire aussi bien avec OGRE donc on va le faire.

C'est un discours ambitieux, et malheureusement irréaliste. En pratique, il faut **réellement** des années de développement pour créer un jeu vidéo si on ne s'appuie pas sur un environnement existant, et à ma connaissance les projets 3D amateur réalisés de la sorte ont un taux d'échec particulièrement dramatique.  
  
Le plus grand piège, c'est de vous imaginer que ça sera facile avec un moteur de jeu, et que pour avoir du challenge, il faut faire tout soi-même. C'est une erreur terrible, et souvent fatale à des projets qui autrement auraient pu fonctionner. Même avec un moteur, vous avez encore des mois de travail devant vous, à plusieurs.   
  
Souvent, les moteurs de jeu intègrent des solutions propriétaires, fermées et que vous ne trouverez pas ailleurs. Le choix de travailler avec vous appartient, mais leur statut de leader est bien souvent mérité : compression vidéo, génération de verdure, interfaces ou optimisation, ce genre de domaine d'expertise demande énormément de travail. Sachez mettre à profit un outil existant plutôt que de tout refaire vous-mêmes.

**Mais alors, OpenGL ou OGRE, ça sert à rien ?**

Non, évidemment, ces outils ne sont pas inutiles. Ils permettent eux aussi, dans l'absolu, de réaliser un jeu vidéo et ils ont un très grand intérêt pédagogique. Il existe d'ailleurs quelques projets fonctionnels basés sur ces technologies, sans moteur de jeu. Mais il faut bien voir deux éléments distincts :

* l'aspect pédagogique de son travail, ce qu'on va apprendre ;
* l'aspect objectif, ce qu'on va réaliser au final.

Ces deux aspects s'opposent dans une certaine mesure : on ne peut pas gagner sur l'un sans perdre sur l'autre, pour un temps donné. Quand on réalise un site Web, on peut faire les deux sans que ça prenne des années, mais dans le cas d'un jeu vidéo c'est une toute autre histoire au vu du travail considérable.   
  
Si le résultat importe peu et qu'on veut apprendre, employer OGRE ou carrément OpenGL est une excellente idée, mais nuit considérablement aux chances de succès sur le plan du jeu lui-même tel qu'un joueur le percevra, ainsi qu'à sa qualité technique : rendu, performances, fiabilité... A vous de choisir votre objectif : n'espérez cependant pas concilier les deux.

**En conclusion...**

Il n'y a pas de solution garantie pour réussir un jeu vidéo 3D. **Mais il y a une solution très fiable pour échouer**, testée et désapprouvée avant vous par des centaines de passionnés et cette solution pour un échec, c'est de tout faire soi-même en comptant faire aussi bien. Vous avez encore un doute ? Parcourez les archives du Site du Zéro pour y retrouver des projets de jeux 3D...  
  
N'hésitez pas à commenter dans ce sujet, à donner votre avis, à réagir. C'est aussi le but.

Merci d'exposer vos avis avec argumentation posée et construite, donnez des exemples.