

Préface

Unity est un outil de création 3D pour Mac OS et Windows. Les moteurs de jeux sont les mécanismes qui fonctionnent en coulisse de tous les jeux vidéo. Ce "moteur" prend les décisions, qu'il s'agisse des graphismes ou des mathématiques qui régissent chaque image à l'écran. À partir du rendu – l'affichage des graphismes à l'écran, l'intégration d'une méthode de contrôle et d'un ensemble de règles à suivre pour le jeu –, les développeurs s'appuient sur le moteur pour "construire" le jeu. Les moteurs de jeu 3D actuels reposent sur une quantité impressionnante de code méticuleusement écrit. C'est pourquoi ils sont souvent revendus, modifiés et réutilisés, une fois que le jeu pour lequel ils ont été créés est terminé. Le moteur Unreal Engine d'Epic Games en est un bon exemple. Développé à l'origine, à la fin des années 90 pour Unreal – un jeu de tir à la première personne (FPS) –, il a ensuite été réutilisé sous licence par d'autres développeurs et a connu un énorme succès dans des centaines de simulations et de jeux commerciaux plus récents.

En raison du niveau de complexité et du coût que représente la création de tels moteurs de jeu commerciaux, il est très difficile de se lancer dans l'industrie du jeu vidéo sans étudier en profondeur les langages de programmation comme le C++. En effet, les jeux sur console et ordinateur sont construits autour du C++ car il s'agit du langage le plus efficace en termes de rapidité de calcul actuellement, et que la structure et les commandes des moteurs de jeux commerciaux exigent des milliers et des milliers de ces lignes de code pour fonctionner. Le code de Unity utilise la compilation juste-à-temps (JIT, *Just In Time*) et la bibliothèque C++ Mono (libre de droits et d'accès au code source). Grâce à la compilation JIT, le code écrit pour Unity est compilé par Mono juste avant son exécution. C'est essentiel pour les jeux qui doivent exécuter du code à des moments précis. En plus de la bibliothèque Mono, les fonctionnalités Unity tirent également parti d'autres bibliothèques logicielles, comme le moteur physique PhysX de Nvidia, OpenGL et DirectX pour le rendu 3D et OpenAL pour l'audio. Toutes ces bibliothèques sont intégrées à l'application, si bien que vous n'aurez pas besoin d'apprendre à utiliser chacune d'entre elles ; il vous suffit d'apprécier ce qu'elles vous permettent de réaliser en toute transparence dans Unity.

Les développeurs de moteurs de jeux construisent également des outils pour commander les fonctions de code qu'ils ont créées. Par exemple, un ensemble d'instructions définit la forme (ou topographie) d'un terrain, son aspect visuel, et même la façon dont sont gérées ses déformations dans le jeu. Mais cet ensemble d'instructions serait inefficace en tant qu'élément du moteur de jeu s'il n'était pas relié à un outil visuel permettant de contrôler ces propriétés. Souvent, les développeurs de moteur de jeu créent donc une interface

graphique (GUI, *Graphical User Interface*) afin que les membres de l'équipe de développement du jeu puissent plus facilement manipuler les éléments du moteur. Ces outils facilitent non seulement le processus de création du jeu mais rendent également le moteur accessible aux acheteurs potentiels et aux équipes de postproduction. Cela est également vrai pour Unity, et une très importante communauté d'utilisateurs partage ses outils sous forme de plugins. Pour en savoir plus, visitez le wiki de la communauté Unify à l'adresse : <http://www.unifycommunity.com/wiki>.

Beaucoup de personnes tentées par le développement se heurtent à la difficulté d'apprendre les langages de programmation comme C++ et les moteurs qui l'utilisent. Si on n'a pas suivi des études de programmation ou d'animation par ordinateur, il est en effet difficile de se lancer dans l'apprentissage des concepts, des méthodes et des principes de conception nécessaires à la production de jeux vidéo. Unity Technologies est une des entreprises qui a décidé de rectifier cette situation. Après avoir créé son propre moteur de jeu en 2001, cette société de développement de jeux danoise a cherché à rendre ses outils abordables en offrant une solution simple axée sur l'utilisateur que tout le monde puisse utiliser. L'équipe a choisi de conserver le code source qui alimente le moteur dans les coulisses et de fournir une interface graphique complète afin que l'utilisateur puisse contrôler le code source de ce puissant moteur sans jamais avoir à créer lui-même de nouveaux éléments dans celui-ci. Ce facteur a rendu Unity très populaire auprès des nouveaux développeurs et explique probablement pourquoi vous lisez cet ouvrage. En définissant des concepts logiques et en regroupant par catégorie les méthodes communes utilisées pour produire des jeux, Unity met la puissance de son moteur au service de l'utilisateur, lui permettant d'obtenir un maximum de résultats avec un minimum d'efforts, afin qu'il se concentre sur le facteur le plus crucial de tous : le gameplay (la jouabilité).

En attirant de nombreux développeurs de jeux, Unity a comblé un vide sur le marché du développement des jeux vidéo, ce que peu d'éditeurs peuvent prétendre. Unity est en effet l'un des moteurs de jeux les plus dynamiques de son secteur car il permet de produire des jeux 3D professionnels aussi bien sur Mac, sur PC ou sur le Web. Comme le moteur existe également dans des versions destinées à la Nintendo Wii et à l'iPhone d'Apple, il offre un accès au marché des ordinateurs mais aussi à celui des consoles et des mobiles une fois que vous maîtrisez les bases.

L'évolution rapide de l'industrie du divertissement et du marketing nécessite que les jeux soient produits rapidement. Aussi, de nombreuses entreprises se tournent vers des solutions intégrées comme Unity pour que leurs créateurs sortent de meilleurs produits le plus facilement possible. Depuis la sortie en 2009 de la version 2.5 et ses premiers pas sous Windows, Unity est de plus en plus utilisé. Mais qu'est-ce que Unity ? Comment fonctionne-t-il ? Que peut-il réaliser ? Et surtout, comment ce programme permet-il de développer des jeux 3D en seulement quelques semaines ?

Le contenu de cet ouvrage

Ce livre est conçu pour couvrir un ensemble d'exemples faciles à suivre, qui culminent dans la production d'un jeu de tir à la première personne (FPS) en 3D dans un environnement interactif (une île). En abordant les concepts communs aux jeux et à la production 3D, vous allez voir comment utiliser Unity pour créer un personnage jouable qui interagit avec le monde du jeu et comment réaliser des énigmes que le joueur devra résoudre pour terminer la partie.

Voici un aperçu, chapitre par chapitre, des thèmes abordés :

- **Chapitre 1. Bienvenue dans la troisième dimension.** Ce chapitre couvre les concepts clés que vous aurez besoin de comprendre pour compléter l'exercice de ce livre. Il présente rapidement les concepts 3D et les processus utilisés par Unity pour créer des jeux.
- **Chapitre 2. Environnements.** Votre monde virtuel est, à l'origine, totalement vide ! Vous allez donc commencer par étudier les différentes manières d'intégrer des terrains, d'importer des modèles 3D créés en dehors de Unity. Vous découvrirez également comment utiliser d'autres fonctionnalités du moteur Unity comme le son et les lumières afin de définir l'environnement du jeu et de le rendre fonctionnel.
- **Chapitre 3. Personnages jouables.** Chaque jeu a besoin d'un héros, n'est-ce pas ? Au cours de ce chapitre, vous étudierez chacun des éléments qui constituent le personnage incarné par le joueur, depuis les contrôles d'entrée en passant par les caméras et la gestion des collisions. Une fois que vous aurez appris de quoi est formé le personnage du joueur, vous l'intégrerez dans l'environnement du jeu et vous effectuerez une petite promenade sur votre île.
- **Chapitre 4. Interactions.** Les jeux vidéo reposent avant tout sur les interactions dans le monde virtuel. Que serait un personnage jouable s'il ne pouvait effectuer aucune action ? Nous présenterons donc ici la détection de collision et le raycasting. Vous verrez comment combiner ces techniques avec du code et des animations pour transformer votre création statique en éléments qui réagissent aux actions du joueur.
- **Chapitre 5. Éléments préfabriqués, collections et HUD.** Il est essentiel de donner au joueur un sentiment d'accomplissement dans le jeu. Pour cela, vous aurez besoin de lui rappeler les actions qu'il a effectuées et de lui donner un but à atteindre. Pour cela, vous créerez un système d'affichage tête haute ou HUD (*Heads Up Display*). Autrement dit, vous apprendrez comment afficher du texte et des illustrations qui changent dynamiquement au cours du jeu.

Vous réaliserez ensuite un mini-jeu de collecte d'objets, dans lequel le personnage du joueur devra ramasser des piles pour pouvoir entrer dans un bâtiment de l'île.

- **Chapitre 6. Instanciation et corps rigides.** Presque tous les scénarios de jeu que vous pouvez imaginer imposent de créer ou de "reproduire" des objets dans l'environnement. Il est essentiel pour tous les développeurs débutants de connaître et de savoir utiliser cette notion, appelée *instanciation* en programmation, qui consiste à créer des objets lors de l'exécution du jeu.

Vous verrez comment améliorer l'interactivité de votre jeu en créant un jeu de tir simple ; en lançant des objets sur des cibles, le joueur pourra débloquent une partie de l'environnement. Vous découvrirez ainsi non seulement l'instanciation, mais aussi comment utiliser les corps rigides.

- **Chapitre 7. Systèmes de particules.** De nos jours, un jeu en 3D doit impérativement proposer quelques effets visuels intéressants pour attirer le joueur. Dans ce dessein, vous créerez un feu de bois pour que le personnage du jeu se réchauffe. Pour cela, vous utiliserez deux systèmes de particules, l'un pour les flammes et l'autre pour la fumée.

Vous verrez comment les systèmes de particules permettent d'imiter le comportement d'un feu et comment rendre chaque particule réaliste en utilisant des images. Enfin, vous éteindrez ce feu afin de donner un objectif au joueur : l'allumer pour ne pas périr de froid !

- **Chapitre 8. Conception de menus.** Pour qu'un jeu soit agréable, ses menus doivent avoir un aspect professionnel et être faciles à utiliser. Qui voudrait d'un jeu dont le bouton Démarrer est impossible à trouver ? Ce chapitre abordera les différentes façons de créer des menus et d'autres interfaces utilisateur avec lesquelles le joueur pourra interagir.

Vous élaborerez des menus en utilisant à la fois les textures GUI et la classe GUI afin de créer des interfaces redimensionnables qui puissent être utilisées aussi bien dans les jeux destinés au Web que dans les applications autonomes.

- **Chapitre 9. Dernières retouches.** Lorsque vous réalisez un jeu, en particulier dans Unity, il arrive un moment où vous créez un élément interactif dont vous êtes si fier que vous souhaitez lui ajouter une touche supplémentaire pour le mettre en valeur.

Au fil de ce chapitre, nous détaillerons l'utilisation du son, les effets de lumière, le rendu des traînées lumineuses et d'autres effets dynamiques simples à implémenter, afin de transformer votre simple jeu fonctionnel en produit fini.

- **Chapitre 10. Compilation et partage.** Vous verrez comment exporter votre jeu pour le Web et en tant que projet autonome. Nous étudierons les différents réglages dont vous devez tenir compte lorsque vous préparez le produit fini, comme la qualité graphique, les commandes de contrôles, et plus encore.

- **Chapitre 11. Procédures de tests et lectures complémentaires.** Vous verrez comment continuer votre apprentissage au-delà de cet ouvrage et comment recueillir des informations auprès des testeurs pour améliorer votre jeu. Nous vous aiderons à préparer votre projet pour qu'il soit testé par un large public afin de recueillir des avis et des commentaires et produire des jeux encore meilleurs !

Ce dont vous avez besoin

Pour réaliser les exercices de cet ouvrage, il vous faut :

- Une copie installée du logiciel Unity (une version d'essai est disponible sur le site Unity3D.com).
- Une connexion Internet afin de télécharger des modèles 3D et des ressources depuis la page dédiée à cet ouvrage sur le site de Pearson (<http://www.pearson.fr>).
- Un ou plusieurs logiciels de modélisation 3D, bien que cela ne soit pas obligatoire. Tous les matériaux utilisés sont fournis. Si vous débutez dans le domaine de la modélisation, vous pouvez télécharger une application gratuite compatible avec Unity comme Blender sur le site Blender.org.

À qui s'adresse cet ouvrage

Formateur sur Unity depuis plusieurs années, j'ai découvert que le principal grief de ses utilisateurs ne portait pas sur le logiciel lui-même, mais plutôt sur le manque de didacticiels destinés aux novices ne possédant pas une formation en programmation.

Dans le contexte actuel, cette situation est rare, bien sûr, mais un outil comme Unity, qui facilite autant la production, accroît l'urgence d'un tel guide didacticiel.

Si vous êtes un concepteur ou un animateur qui souhaite faire ses premiers pas dans le développement de jeux vidéo, ou si vous avez tout simplement passé de nombreuses heures assis devant des jeux vidéo et que vous fourmilliez d'idées, Unity et ce livre pourraient constituer le point de départ idéal. Nous supposons que vous n'avez aucune connaissance en production de jeux vidéo et nous vous expliquons tout depuis le commencement. Nous vous demandons simplement d'être passionné et de vouloir réaliser de grands jeux.

Conventions typographiques

Cet ouvrage utilise plusieurs styles de texte afin d'établir une distinction entre les différents types d'informations. En voici quelques exemples ainsi qu'une explication de leur signification.

Le code qui apparaît dans le corps du texte est présenté de la manière suivante : "Nous pouvons inclure d'autres contextes, par l'utilisation de la directive `include`".

Voici un exemple de bloc de code :

```
if(collisionInfo.gameObject.name == "matchbox"){
    Destroy(collisionInfo.gameObject);
    haveMatches=true;
    audio.PlayOneShot(batteryCollect);
```

Les nouveaux termes et les mots importants sont en *italiques*. Les mots affichés à l'écran, dans les menus ou les boîtes de dialogue, par exemple, seront en petites capitales comme ceci : "Cliquez sur le bouton SUIVANT pour passer à l'écran suivant."

Dans la notation des raccourcis clavier, la première variante est toujours pour Mac OS, la seconde pour Windows. Appuyez sur Entrée/F2 signifie donc Entrée sur Mac OS et F2 sur Windows.

Codes sources en ligne

Les fichiers des exemples de code ainsi que les instructions permettant de les utiliser sont disponibles depuis le site Pearson (<http://www.pearson.fr>) en suivant le lien "Codes sources" à la page consacrée à cet ouvrage.