

Génération procédurale de ville

Présentation du 27 novembre 2015 de l'UE Moteur de Jeu

De Rousiers et El Aouaji

Génération procédurale et application à la création de ville

La génération procédurale existe depuis longtemps dans le milieu de l'informatique (on remarquera notamment les fractales en informatique graphique) et dans le secteur du jeu vidéo depuis le milieu des années soixante-dix.

Sa première apparition est due à un problème de stockage, les machines de l'époque n'ayant pas assez de capacité de stockage pour gérer beaucoup de niveaux, certains ont commencé à être générés pendant l'exécution du jeu (niveaux de rogue-like en ASCII pour les premiers jeux utilisant la génération procédurale).

A mesure que les capacités de stockage augmentaient, le problème s'est estompé et la génération procédurale s'est retrouvée utilisée pour augmenter le contenu. De « Elite » et ses millions de galaxies générées (même si seulement 6 furent inclut dans le jeu final, les producteurs ayant peur d'écœurer le joueur), à « Minecraft » et son monde infini en passant par « Left 4 Dead » et son IA adaptative, les exemples ne manquent pas.

Son application à la génération de ville nécessite plusieurs étapes :

- Générer les axes de communications (route, métro, voie ferrée...)
- Créer les îlots d'habitation
- Construire les bâtiments

Ces trois étapes nécessitent chacune de prendre en compte le terrain sur lequel on se trouve et de nombreuses données peuvent être fournies suivant le système utilisé (carte de densité, carte topologique...).

Construction des infrastructures de communications

Pour construire les axes de communications, on va se baser sur deux principaux modèles :

- Le pattern européen, constitué de routes sinueuses, de rond-points, de virages et d'arrondissements (Paris)
- Le pattern américain, avec des routes droites, parallèles ou perpendiculaires les unes aux autres (New-York)

Ces axes de communications vont permettre de délimiter les îlots d'habitation, les zones de construction des bâtiments.

Il existe trois principes de constructions particulièrement intéressants :

- Le Grid Layout : qui consiste à poser une grille sur le terrain, chaque ligne correspondant à une route et chaque cellule à un îlot. Très basique et rapide, il peut être utile dans le pattern américain si on le complexifie un peu en fusionnant ou découpant des cellules. Dans le cas contraire, ce modèle manque de réalisme.
- Le L-System : plus réaliste que le précédent, il nécessite cependant des données en entrée qui vont constituer l'alphabet (rond-point, virage en épingle...) ainsi qu'une série de règles qui vont constituer la grammaire du système. C'est un bon compromis, alliant vitesse et réalisme, il faut néanmoins noter que le résultat est très sensible aux données fournies.
- L'« Agent based simulation » : bien plus long à mettre en place, c'est néanmoins la solution la plus complète. Elle se base sur un ensemble d'agents représentant les acteurs du développement urbain (urbaniste, architecte, ouvrier...). Cette simulation va permettre de réaliser une ville avec un développement réaliste car la ville va évoluer au fil du temps, c'est d'ailleurs le seul vrai point faible de la méthode, son temps de calcul. Il est aussi assez difficile d'estimer le résultat final de par la liberté laissée à la méthode.

Génération de bâtiments

Le parcellement des zones

A partir des routes générées précédemment, on va pouvoir construire nos bâtiments qui vont occuper les zones formées par les routes.

Tout d'abord, il va falloir créer des parcelles au sein de ces zones. Pour ce faire, la zone va être subdivisée récursivement et de manière aléatoire. On peut subdiviser les zones de différentes manières et ainsi obtenir des parcelles de formes variées (triangulaires, rectangulaires, hexagonales, etc...). La subdivision s'arrête une fois que la parcelle atteint (ou dépasse) une taille minimale définie à l'avance.

Une fois la subdivision terminée, les parcelles trop petites pour accueillir un édifice ou celles qui n'ont pas d'accès direct à une route (elles sont entourées par d'autres parcelles) sont supprimées, elles peuvent alors servir de cour intérieure ou autre.

Modélisation des bâtiments

Maintenant que les parcelles ont été créées, on va construire un bâtiment sur chacune d'elles. On suppose qu'une parcelle accepte exactement un édifice. On se concentre également sur la méthode des L-Systèmes, qui est la plus répandue pour la génération procédurale de ville.

Un L-Système est une grammaire permettant de définir un ensemble de mots admissibles, à partir d'un alphabet donné (les variables), un axiome de départ et des règles de production.

- L'alphabet contient des formes géométriques élémentaires : parallélépipède, cylindre, sphère...
- Un axiome de départ est choisi. Dans notre exemple, c'est une boîte englobante de l'édifice à construire, généré à partir de la forme de la parcelle qui va l'accueillir.
- Les règles de productions sont nombreuses : transformations (mise à l'échelle, translations, rotations), extrusions, duplications.

On considère des types de bâtiments pour simplifier le système. Typiquement, des gratte-ciels, des habitations, des commerces, par exemple. Chaque type a son propre ensemble de règles de productions. En effet, un gratte-ciel est trop différent d'une maison pour utiliser les mêmes règles que cette dernière.

Maintenant que le système a été posé, on itère sur l'axiome de départ autant de fois que souhaité. La boîte englobante est alors représentée par une variable et est membre gauche d'une règle de production. Le résultat de cette règle est redéveloppé tant que le nombre d'itérations n'est pas atteint. De part la nature de cette méthode, il est facile d'implémenter du LOD, en baissant le nombre d'itération du système pour les éléments les plus éloignés à l'écran.

Le produit de l'application des règles est un mot admissible par cette grammaire. Il faut donc analyser ce mot (*parsing*) pour permettre l'affichage par les fonctions de rendu graphique.

Texturisation des bâtiments

La modélisation étant effectuée, il ne reste plus qu'à donner de la texture à nos bâtiments. Pour y arriver, nous allons organiser les façades des édifices sous la forme d'une superposition de structures de grilles (*Grid Layouts*). Chaque grille aura pour but de représenter un type d'élément de la façade : fenêtres, portes, plaques et autres.

Un *Grid Layout* est créée à partir d'un groupe d'intervalles qui sont des suites ordonnées d'entiers qui définissent l'espacement par rapport à une unité de base. Les intervalles modélisant la verticale et l'horizontale sont alors associés par des fonctions logiques simples (ET, OU, par exemple).

Une fois les grilles créées, d'autres fonctions vont déterminer les cellules dites « actives », c'est-à-dire, les cellules qui seront affichées sur la façade finale. Enfin, les dernières fonctions auront pour but d'appliquer la texture aux cellules actives, suivant l'élément de façade auxquelles elles correspondent.

Intérêts dans le monde du Jeu Vidéo

Les principaux intérêts de la génération procédurale de ville dans le milieu du Jeu Vidéo sont assez communs.

Il y a tout d'abord le réalisme, avec une méthode de construction efficace et bien calibrée pour le résultat escompté, des règles précises et bien définies, le résultat peut être bluffant et égaler le travail d'une équipe d'artistes.

Et si le résultat n'est pas jugé suffisamment bon, le second intérêt le comble. En effet, la rapidité de la génération procédurale permet de créer une ville sans grande identité mais bien remplie afin que les artistes n'aient plus qu'à se concentrer sur quelques points d'intérêts pour la rendre vivante et riche.

Ces deux aspects proviennent d'une évolution de la mentalité dans le monde du Jeu Vidéo. Une fois que les espaces de stockages sont devenus assez importants pour gérer beaucoup de contenu, la tendance fut d'augmenter la taille des équipes artistiques pour créer de plus en plus de données. Il est malheureusement très vite apparu qu'augmenter le nombre d'artistes travaillant sur un même projet n'augmentait bientôt plus, ni la quantité, ni la qualité du contenu. Le chemin a donc été parcouru en sens inverse, réduisant le nombre d'artiste et les associant à la génération procédurale.

Ceci permet de créer des données de qualité par les artistes et d'en faire en quantité via les méthodes procédurales.

Cette inversion de tendance conduit maintenant les plus gros projets, tel « Star Citizen » (plusieurs millions de budget), à constituer une équipe entièrement dédiée à la génération procédurale. Cette équipe aura pour but de générer pendant plusieurs années un contenu quasi infini pour le jeu afin de remplir les différents mondes explorables et les immenses cités qu'ils abriteront.

Les limites dans le domaine du jeu vidéo

Le *level design*

Un des problème de ce système est le manque de richesse des édifices modélisés. Même si le système générant la ville est plus ou moins complexe, il manquera toujours les détails qu'aurait pu insérer un artiste dans la création des bâtiments de sa ville comme des éléments sur la façade ou des architectures impossibles pour ces systèmes. Il peut créer des constructions extrêmement différenciées plus facilement qu'un générateur.

De plus, cette méthode est limitée au nombre de types de bâtiments définis et augmenter ce nombre la complexifierait. La création d'édifices aux rôles divers devient compliqué, ce qui n'est pas le problème d'un artiste qui peut à priori modéliser tout type de bâtiment.

Enfin, cela dépend grandement des données en entrée du système, que ce soit pour les textures, ou les règles de productions. Donc, il y a une limite de création à laquelle un modélisateur humain ne devrait pas faire face avec de l'imagination.

La complexité

La génération procédurale de ville est une opération très coûteuse en temps de calcul. La création des routes ne prend qu'un dizaine de secondes, cependant la génération des bâtiments dure plusieurs minutes pour une ville assez large (comme Manhattan). De ce fait, il n'est pas possible d'utiliser cette technique dans un jeu en temps-réel.

Conclusion

La génération procédurale appliquée aux villes est une technique très intéressante. Bien que peu utilisée dans les jeux vidéos (mais bien plus dans une optique de recherche), elle permet néanmoins de créer des villes immenses et variées et va être de plus en plus employée.

De nombreuses techniques existent et permettent de remplir la plupart des demandes des développeurs. Si les plus efficaces sont trop gourmandes pour le temps réel, elles restent très utile durant la production du jeu afin de créer une base destinée aux artistes. Ceux-ci pourront alors l'utiliser afin de créer une ville avec une réelle identité.

Références :

Un article en anglais très intéressant sur la génération procédurale appliquée aux villes et expliquant en détails les différentes méthodes de création des axes de communication notamment :

<http://gamesitb.com/SurveyProcedural.pdf>

Rapport de thèse sur la génération procédurale de monde, contient des parties dédiées à la génération de terrain ainsi qu'à la génération de ville :

<https://tel.archives-ouvertes.fr/tel-00841373/document>

Un projet datant des années 2000 en OpenGL dont le but est de fournir un « screensaver » générant à la volée une ville de nuit :

<https://code.google.com/p/pixelcity/>

Un article sur la création de réseaux logiques et fermés, pouvant aider à créer les axes de communications :

http://nliataud.fr/wiki/articles/generateur_reseau