

# Rapport de documentation de stage

Raoelisolonarivony - MISA M2

22 Janvier 2017

## A Survey of Procedural Methods for Terrain Modelling (2009) of Ruben M. Smelik et al.

La création manuelle des contenus des mondes virtuels 3D est laborieuse et répétitive. Le principe de la modélisation procédurale est de créer les contenus (les textures, les modèles géométriques, ...) d'une manière plus automatisée.

Cette note recense les méthodes procédurales appliquées sur la **modélisation de terrain**, évaluant le réalisme de leur résultat, la performance et le contrôle que les utilisateurs peuvent avoir sur la procédure.

Il concerne la distinction de plusieurs couches dans le modèle du terrain, chacune contenant des caractéristiques naturelles (terre, eau, couches de végétation) et faites par l'homme (route et couches urbaines).

### Height-maps

Les "Height-maps" ou champs de hauteur sont des grilles à deux dimensions contenant des valeurs d'élévation, souvent utilisés comme base d'un modèle de terrain.

Les anciens algorithmes sont basés sur la **subdivision**. Miller (1986) décrit la subdivision itérative du height-map. A chaque itération un nouveau point d'élévation est assigné à la moyenne des valeurs des coins d'un triangle ou d'un diamant additionnée d'une correction aléatoire. L'intervalle de la correction diminue (réduit de moitié par exemple) à chaque itération pour paramétrer la rugosité du height-map.

Les récentes générations de height-maps reposent sur les **générateurs de bruits fractals** (Fournier et al., 1982; Voss, 1985), comme le bruit de Perlin (Perlin 1985, Perlin 2001), qui génère des bruits en prenant un échantillon et en interpolant les vecteurs gradient du height-map. Redimensionner et ajouter plusieurs niveaux de bruit pour chaque point du height-map conduit à des structures naturelles, pseudo montagneux.

Ces height-maps peuvent être transformés sur la base de filtres communs d'image (ex. lissage) ou sur les simulations de phénomènes physiques, comme l'**érosion**. L'érosion thermique réduit par exemple les changements aigus pour l'élévation, en distribuant itérativement le matériau depuis les points les plus hauts vers les plus bas, jusqu'à ce qu'un angle de talus, i.e. un angle maximale de stabilité pour le matériau (comme la pierre ou le sable) soit atteint.

Bien que ces algorithmes d'érosions ajoutent du réalisme aux terrains montagneux, ils sont réputés lents, nécessitant un nombre élevé d'itérations. Une récente recherche s'est penchée sur des algorithmes d'érosion interactifs, souvent en portant les algorithmes sur le GPU. Des exemples prometteuses comprennent (Anh et al. 2007) et (Stava et al., 2008).

Une limitation des height-maps est qu'ils ne supportent pas les rochers surplombants et les grottes (cavités). Gamito et Musgrave (2001) proposent un système de gauchissement de terrain qui résulte à un surplomb régulier artificiel. Une méthode (Peytavie et al., 2009) procure une structure plus élaborée avec différentes couches de matériaux qui supportent les rochers, les arches, les surplombs et les grottes. Leurs modèles de terrain sont visuellement plausibles et naturels.

## Rivières, Océans et Lacs

Les algorithmes de génération de rivières peuvent s'exécuter soit pendant soit après la génération des height-maps.

Kelley et al (1988) prennent un réseau de rivières comme la base d'un height-map. Ils commencent par une unique rivière droite et la subdivisent récursivement, produisant un réseau. Un squelette pour le height-map se forme, complété ensuite en utilisant une fonction d'interpolation. Le type de climat et les constituants du sol impactent la forme du réseau du courant.

Prusinkiewicz et Hammel (1993) combinent la génération d'une rivière courbée avec un schéma de subdivision de height-map. Du triangle de départ d'une rivière, un bord est marqué comme l'entrée de la rivière et l'autre bord comme sa sortie. Dans l'étape de subdivision, le triangle est divisé en triangles plus petits, et la course de la rivière à partir de l'entrée jusqu'à la sortie peut prendre alors plusieurs formes. Un point négatif de cet approche le niveau d'élévation constant et bas de la rivière, qui traverse les terrains montagneux de manière non naturelle.

Une approche décrite par Belhadj et Audibert (2005) crée un height-map avec les crêtes de montagne et les réseaux de rivières. Commençant avec une carte vide, ils placent une paire de particules de crête sur une élévation particulièrement haute et les déplacent dans toutes les directions. Une courbe de Gauss est dessinée sur le height-map par les positions des particules à chaque itération. Ils placent ensuite les particules de la rivière sur le sommet de la crête de la montagne et les laissent s'écouler vers le bas suivant des règles simples de physique. Pour les crêtes de montagnes et les vallées irriguées par un réseau de rivières denses, la méthode est rapide et efficace.

A l'exception des rivières, les océans et les lacs et leurs connections, les réseaux de courants, deltas et chutes d'eau, ont retenu peu d'attention. La formation des lacs n'est pas considéré du tout. Les océans sont communément générés en prenant un niveau fixe d'eau (ex. 0m) ou en commençant par un algorithme d'inondation à partir des points à faible élévation. Teoh (2008) a également déclaré que la recherche sur ce domaine est incomplète : plusieurs rivières et caractéristiques côtières n'ont pas été traitées. Il propose des algorithmes simples et rapides pour les rivières sinueux, les deltas et la formation de la plage.

## Les modèles de plantes et la distribution de la végétation

Les modèles d'arbres et des plantes procéduraux sont générés puis placés automatiquement dans le modèle de terrain.

Les procédures de génération permettent d'obtenir des modèles de chaque espèce de plantes :

- les systèmes à grammaire formelle comme le L-system de Lindenmayer (1990). Il s'agit de faire croître la plante depuis la racine en passant par les branches jusqu'aux feuilles en appliquant des règles de production.
- la modélisation des composantes de la plante par un graphe, Lintermann et Deussen (1999). La génération se fait en parcourant les composantes connexes du graphe. (exemple du logiciel XFrog)

Le placement des plantes automatique facilite la génération de végétation plus dense comme les forêts :

- la simulation de l'écosystème pour une surface à partir d'une carte de hauteur et de composition du sol, d'une carte d'eau et des propriétés des plantes comme leur taux de croissance (Deussen et al. 1998, Hammes 2001)

La modélisation procédurale de la végétation produit des résultats fiables et sont déjà bien appliquées dans les jeux vidéos modernes, par exemple en utilisant le package commercial SpeedTree.

## Les réseaux routiers

Les méthodes de génération de réseaux routiers pour les villes traitées dans le document sont basées sur les pattern, les L-system, les simulations d'agents et les champs tensoriels.

Le réseau routier peut être générer simplement en modélisant une grille de carré (Greuter et al. (2003)). Le bruit de déplacement peut être ajouté aux points de la grille pour créer un réseau moins répétitif, mais le réalisme de cette technique est limité.

L'élaboration des routes par le biais des modèles (ou "templates"), comme proposée par Sun et al. (2002) consiste à déterminer des patterns fréquents dans les réseaux routiers réels et de les reproduire. Le template correspondant au pattern peut être basé sur la population (implémenté comme un diagramme de Voronoi des centres de population). Des règles simples sont appliquées pour vérifier leur validité : quand des zones d'impasse sont rencontrées (ex. les océans), elles sont annulées ou déviées. Les routes principales sont souvent courbées pour éviter les gradients d'élévation large.

Parish et Müller (2001) utilisent un L-system étendu pour faire croître leur réseau routier. Le L-system est conduit par des objectifs précis (goal-driven) comme la densité moyenne de la population (les rues tentent de connecter les centres de population et les patterns spécifiques de routes). Leur L-system est étendu avec des règles qui ont une tendance à connecter les routes nouvellement proposées avec les intersections existantes. Les rues sont aussi insérées dans les zones restantes comme des grilles simples.

Kelly et McCabe (2007) introduisent l'éditeur de ville interactive CityGen, dans lequel un utilisateur définit les routes principales en plaçant des noeuds dans le terrain 3D. Les régions incluses par ces routes peuvent être complétées avec les grilles style Manhattan ou les routes de croissance industrielle avec des routes sans issues.

Glass et al. (2006) décrivent des répliques de structures de route rencontrées dans les colonies informelles d'Afrique du Sud en utilisant la combinaison d'un diagramme de Voronoi pour les routes

majeures avec des L-système ou des subdivisions régulières avec ou sans bruit de déplacement pour les routes mineures. Ils ont raisonnablement réussi à recréer les patterns observés.

Lechner et al. (2003) présentent une approche basée sur des agents : ils divisent la ville en zones incluant non seulement les zones résidentielles, commerciales ou industrielles, mais également des zones spéciales comme les bâtiments gouvernementaux, les squares, et les institutions. Ils placent deux agents, appelé *extender* et *connector*, sur une position de semence dans la carte du terrain. L'*extender* recherche les zones non connectées dans la ville. Quand il trouve une zone qui est localisée non loin d'un réseau routier existant, il cherche le chemin le plus convenable pour connecter cette zone au réseau routier. Cette méthode donne des résultats plausibles, mais son désavantage est son temps d'exécution long.

Chen et al. (2008) proposent une modélisation interactive de réseaux routier en utilisant les champs tensoriels. Ils définissent la création de patterns communs de routes en utilisant des champs tensoriels. Un réseau routier est généré à partir d'un champ tensoriel, en traçant des lignes à partir des points de semence dans les directions des vecteurs propres jusqu'à ce qu'une condition d'arrêt soit remplie. Ensuite, suivant la courbe tracée, de nouveaux points de semence sont placés pour tracer les lignes dans la direction perpendiculaire cette fois-ci. Des utilisateurs peuvent placer des bases de champs tensoriels, comme un pattern radial, lisser le champ, ou utiliser un pinceau pour contraindre localement le champ vers une direction spécifique. Du bruit peut être appliqué pour rendre le réseau routier moins régulier et plausible.

Dans les méthodes discutées, l'influence de la carte du terrain sous-jacente et du profil d'élévation est à degré variable prise en compte. La plupart des méthodes prennent seulement des mesures basiques pour éviter des routes montantes avec des pentes trop raides et des routes qui traversent des corps d'eau. Kelly and McCabe (2007) planifient le chemin précis de leurs routes principales entre les noeuds mis en place par l'utilisateur pour avoir autant de possibilités dans l'élévation que possible. Seulement, pour les terrains durs cette mesure ne sera pas adéquate et le terrain doit être modifié pour l'accommodation de la route. Bruneton and Neyret (2008) proposent une méthode simple et efficace pour mixer les profils des routes dans les height-map en utilisant les shaders.

## Les environnements urbains

Kelly and McCabe (2006) donnent un point de vue élaboré de plusieurs approches pour la génération d'environnement urbains. Watson et al. (2008) donne un point de vue pratique de l'état de l'art.

L'approche la plus commune pour générer des villes procéduralement est de commencer par un réseau de routes et d'identifier les régions polygonales incluses dans les rues. La subdivision de ces régions produisent des lots, pour lesquels différentes méthodes de subdivisions existent, voir par ex. Parish and Müller (2001) ou Kelly and McCabe (2007). Pour remplir ces lots avec des immeubles, soit la forme du lot est utilisée directement comme empreinte d'immeuble, ou l'empreinte d'un immeuble est adapté pour le lot. En faisant simplement l'extrusion de l'empreinte jusqu'à une hauteur aléatoire, on peut générer une ville avec des gratte-ciels ou des immeubles de bureau.

Pour obtenir des formes d'immeubles intéressantes, plusieurs approches ont été mises en oeuvre.

Greuter et al. (2003) génère des immeubles de bureau en combinant plusieurs formes primitives dans un plan d'étage et en faisant l'extrusion de cette forme pour différentes hauteurs. Parish and Müller (2001) commencent par un plan d'étage rectangulaire et appliquent un L-system pour affiner l'immeuble. Ces approches sont surtout utiles pour des modèles simples d'immeubles de bureau.

Wonka et al. (2003) introduisent le concept de *split grammar*, une grammaire formelle indépendante du contexte (context-free formal grammar) dont le design est fait pour produire des modèles d'immeubles.

Müller et al. (2006) appliquent une grammaire appelée *shape grammar*. La propriété principale d'une shape grammar est qu'elle utilise des règles dépendant du contexte (context-sensitive), tandis qu'une split grammar utilise des règles indépendantes du contexte (context-free), qui permettent la possibilité de modéliser les toits et les formes arrondies (rotated shapes). En plus des immeubles d'affaire bien connus, la grammaire peut aussi modéliser des immeubles résidentiels.

Bien que les shape grammars dans Müller et al. (2006) peuvent générer des modèles d'immeubles visuellement convaincants, Finkenzeller and Bender (2008) notent que l'information sémantique à propos du rôle de chaque forme dans l'immeuble final est manquante. Ils proposent de capturer cette information sémantique dans un graphe typé. Finkenzeller (2008) décrit avec plus de détails la génération des façades et des toits de leur système (voir fig. 1c).

Young et al. (2004) décrivent une méthode pour créer des maisons au style vernaculaire du sud-est de la Chine en utilisant un shape grammar étendue. La grammaire est hiérarchique et commence au niveau de la ville (tandis que dans d'autres méthodes un shape grammar est appliqué à l'empreinte d'un immeuble individuel). La grammaire produit ensuite les rues, les blocs d'habitation, les routes, et même les maisons de productions avec des composantes comme les portails, les fenêtres, les murs et les toits.

Müller et al. (2007) utilisent une approche très différente pour la construction des façades des immeubles. Leur méthode prend un image unique de la façade de l'immeuble réel comme entrée et est capable de reconstruire un modèle détaillé de la façade en 3D, en utilisant une combinaison d'image et de génération de shape grammar.

Bien que les méthodes ci-dessus donnent rapidement et visuellement des résultats plaisants, les villes qu'elles génèrent manquent de structure réaliste. Une nouvelle recherche incorpore les théories et les modèles d'aménagement de territoire urbain existantes dans le processus de génération. Groenewegen et al. (2009). Ils prennent en compte un grand éventail de facteurs significatifs, incluant le plan historique de la ville et de l'attraction que certains types de terrain (côtes, océans, rivières) peuvent avoir par ex. les districts industriels ou résidentiels de grande classe. Weber et al. (2009) utilisent des modèles comparables (comparaison un peu simplifiée) pour une simulation d'une ville dans le temps. Leur système est rapide (environ 1s pour un an de simulation) et interactif, ce qui veut dire que l'utilisateur peut guider la simulation en changeant les routes ou en mettant une peinture pour le sol utilisé sur le terrain.

## **Génération automatique d'environnements virtuels urbains (2016) de Tiavina NIVOLALA**

Le master de Tiavina s'intéresse à la génération de villes de façon procédurale reposant toutefois sur des modèles de bases stockées dans une base de données. Les modèles rendent flexibles l'introduction de variations dans la structure de la ville et des bâtiments et procurent une sémantique à la génération.

Les standards pour la représentation (description, gestion et stockage) des modèles et procédures

sont indiqués tels que IFC, CityGML, QUASY ou CityTree. Une synthèse de ces standards explicitent les forces (paramétrisation possible, sémantique riche) et faiblesses (volumétrie des fichiers, complexité, modules inutilisés, validation XML coûteuse en mémoire) de chacun et les relations entre eux.

Elle présente les visualisations de villes. La visualisation directe avec OpenGL en temps réel qui utilise la triangulation adaptative des modèles de terrains virtuels et la gestion des textures suivant les niveaux de détails pour gérer des modèles 3D précis à partir d'une base au format 3D. La visualisation basée sur les modèles 3D s'appuient sur les standards KML, X3D, etc. La génération peut être effectuée à partir des données topographiques pour le modèle d'élévation, les routes sont superposées dessus, les bâtiments sont issus des plans au sol par extrusion verticale, les textures sont raffinées. La visualisation basée sur CityGML est présentée par plusieurs exemples comme les framework de visualisation en ligne sous les format CityGML et KML et permettent de générer des scènes avec des niveaux de détails différents selon le point de vue de l'utilisateur.

Elle discute ensuite de la génération procédurale des villes et décrit deux frameworks principaux : undiscovered worlds et CityGen. Pour Undiscovered worlds, aucun import ou export de données est nécessaire, le framework est pseudo-infinies (extensible tout le long du point de vue de l'utilisateur) et formé de trois composantes : View frustum filling, la mise en cache des géométries et leur génération. Le framework CityGen est proposé pour automatiser la création de la géométrie de la ville et un système interactif pour modifier les villes. Ce processus se déroule en trois étapes : la génération du réseau routier primaire, la génération de réseau routier secondaire et la génération des bâtiments. Une synthèse est fournie et compare les techniques de visualisations des approches par modèles aux approches procédurales.

Elle présente sa méthodologie pour le projet et des étapes de sa réalisation, entre autre la conception et l'alimentation de la base de données puis la génération de l'environnement suivant cette base. La base est constituée des éléments composant l'environnement de manière hiérarchisée. Elle a utilisé PostGreSQL et son extension PostGIS. Elle discute du modèle solide pour représenter les géométries. Elle indique le schéma relationnel de la base et son alimentation à partir de modèles créés sous Blender et insérer à travers un script Python.

## **New generation crowd simulation algorithms (2014) of Julien Pettre and Ming Lin.**

La foule est constituée d'un ensemble d'individus localisés dans un même endroit partageant les mêmes objectifs. Chaque individu a des interactions avec ses voisins (dépendant de facteurs individuels et du contexte physique, psychologique, social et environnemental).

La foule en mouvement dans les places publiques, les immeubles ou les événements forment surtout des interactions physiques : les individus s'évitent, se suivent, se regroupent ou se dispersent. Combinées, ces interactions forment une structure émergente à grande échelle et consistent les principales caractéristiques des mouvements.

Deux classes majeures d'approches se distinguent : l'approche macroscopique (considérer l'aspect global du mouvement de la foule comme la modélisation de celle-ci comme un fluide visqueux) et l'approche microscopique ou locale qui produit des trajectoires d'agent individuel lisses, basée sur les modèles d'interactions entre les agents.

L'objectif de l'approche microscopique est de calculer les comportements macroscopiques en simulant les interactions entre le peuple à une échelle locale (par exemple, la simulation des piétons).

Les éléments de la simulation sont :

1. La sélection des voisins : trouver l'ensemble de voisins en interaction avec l'agent

2. Le modèle d'interaction locale : comment chaque interaction influence le mouvement des agents
3. La combinaison des interactions : l'intégration de toutes les interactions pour un agent donné

Les modèles d'interaction locale **basées sur la vitesse** sont capables de simuler comment les humains dirigent leur mouvement en anticipant ceux des autres, une percée au niveau du réalisme de la simulation.

Leur particularité est la prédiction des conditions du trafic global depuis des modèles numériques d'interactions physiques d'une foule en mouvement : éviter les obstacles statiques ou en mouvement dans le voisinage, ie. éviter les collisions et l'interpénétration entre les corps simulés.

Eviter la collision par anticipation signifie effectuer les manoeuvres d'évitement bien avant que les agents soient proches les uns des autres.

Dans les modèles basés sur la vitesse, les interactions sont définies en fonction de la position de chaque agent (son état) et de sa vitesse (dérivée de la position par rapport au temps).

Une décomposition, pour chaque agent, du domaine de vitesses atteignables (tous les mouvements global qu'un agent peut effectuer) en deux domaines : le domaine des vitesses admissible et celui des vitesses inadmissibles.

L'espace admissible est l'ensemble des vitesses avec lesquelles un agent peut se déplacer sans risque de futures collisions. Le domaine inadmissible contient les vitesses provoquant des collisions.

La notion de temps est liée à celle du risque de collision par la TTC (Time-to-collision) qui décrit le risque suivant la dimension de temps.