

IAS

Projet Génération Procédurale de Données

Générer des niveaux de jeux de Plateforme

Mai 2020

1 Contexte

L'une des frontières actuelles de l'Intelligence Artificielle est l'imitation de la créativité humaine. La *Créativité Computationnelle* est le domaine de recherche qui s'intéresse à la production d'objets divers grâce à des méthodes informatiques, dont l'IA.

Journalisme automatique, musique, design, jeu vidéo sont parmi les domaines les plus murs industriellement. L'IA collabore ici avec l'humain: l'entrée du système est souvent fabriquée par un humain, et la sortie sera également le plus souvent modifiée (ou choisie) par un humain également.

Le jeu vidéo est un domaine d'application qui se prête bien à l'utilisation de ces techniques, pour créer des nouveaux objets de jeu, de nouvelles classes de personnages, ou même générer des niveaux entiers du jeu. Les techniques peuvent reposer sur l'utilisation de templates (par exemple pour des générateurs de "missions" dans certains jeux en ligne) ou des méthodes constructives avec des briques de base fabriquées par les humains, mais on peut aussi utiliser des algorithmes de recherche heuristique, ou d'apprentissage machine. Le jeu infinite Mario par exemple¹, génère automatiquement des niveaux.

2 Sujet

Ce sujet vous permet d'explorer la génération procédurale de contenu, dans le cadre de la génération de niveaux de jeux de plate-forme en 2D, dans un cadre extrêmement simplifié afin de vous concentrer sur les méthodes. Vous n'aurez pas à réaliser les personnages autonomes, juste les décors et placements d'objets statiques. Vous pourrez, si vous le désirez, utiliser les ressources graphiques disponibles sous licence gratuite comme par exemple celles-ci (que ce soit pour la génération d'images ou de niveau unity. Les ressources sont disponibles pour unity sur l'*asset store* également) :

<https://pixel-frog.itch.io/pixel-adventure-1>



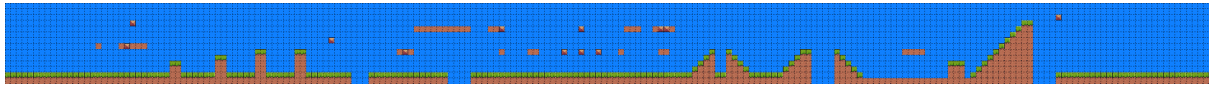
Pour ce travail, un niveau de jeu de plate-forme peut se voir comme une image en 2D.

Sur cette image, on va composer des *tiles* de même taille, pour former une mosaïque.

Vous avez la possibilité d'utiliser Unity3D (avec les outils 2D), ce qui vous permettra d'utiliser les ressources et modèles disponibles, ainsi que certaines bibliothèques. Mais vous pouvez aussi travailler avec des images, ou d'autres outils, surtout si vous n'êtes pas à l'aise avec Unity3D ou si cela ne vous motive pas plus que cela, ou de choisir selon les algorithmes disponibles.

¹<https://supermariobros.io/infinite-mario-bros>

Ci dessous: un niveau simple inspiré du premier niveau du jeu Mario, édité avec l'outil Tiled à partir des ressources disponibles sur <https://pixel-frog.itch.io> pour vous donner une idée de ce qui est attendu en sortie. Un exemple de code vous sera fourni pour vous mettre le pied à l'étrier.



2.1 Premier niveau, sans intelligence poussée pour prendre le problème en main

Lors d'une première étape, vous utiliserez des règles, des *contraintes* que vous définirez, pour placer une à une les éléments du jeu. On parle ici d'approche *constructive*. Suivant la qualité de réalisation de cette étape, vous pourrez obtenir jusqu'à 10 points. Vous pourrez, par exemple, commencer par générer des colonnes à partir des tiles, puis travailler avec un ensemble pré-défini de colonnes de tiles, ou autre arrangement.

Par exemple, une contrainte pourrait consister à dire qu'il ne faut pas des "trous" trop larges, ou de différence de hauteur trop grande pour deux colonnes successives.

Une fois le problème bien compris grâce à cette première réalisation, vous pourrez implémenter au choix une des approches suivantes pour obtenir une meilleure note.

2.2 Wave Function Collapse

Cette technique, proche du machine learning, a la particularité de ne nécessiter qu'un seul exemple de données et peut permettre d'obtenir des résultats spectaculaires pour la génération de niveaux. Il existe une implémentation de cet algorithme pour Unity, que vous pouvez utiliser pour ce projet. Vous pouvez également utiliser cet algorithme qui existe pour des langages variés sur des images.

L'idée de l'algorithme est de partir d'un exemple modèle, et de le parcourir par morceau, pour "apprendre" échantillon par échantillon les règles de sa construction.

Voir un tutorial ici pour l'utilisation de plusieurs implémentations: <http://www.procjam.com/tutorials/wfc/>.

Vous pouvez trouver des exemples spectaculaires d'utilisation de cet algorithme, par exemple ici: <https://marian42.de/article/wfc/>

2.3 Recherche (et recherche Méta Heuristique)

Dans cette approche, on va considérer l'espace/l'ensemble de tous les niveaux possibles, et définir une fonction *fitness* (comme une fonction de coût), afin de rechercher dans cette espace quels sont les "meilleurs" niveaux à sélectionner pour la génération. Des algorithmes classiques: algorithme glouton, recuit-simulé, algorithme de recherche génétique.

Plusieurs méthodes sont décrites dans ce papier. Matthew Guzdial, enseignant-chercheur en génération procédurale, introduit ces méthodes dans cette vidéo.

3 Rendu

En plus des livrables du projet, vous soumettrez quelques niveaux générés par vos programmes et qui vous plaisent, vous nous expliquerez pourquoi.

4 Références utiles

- travailler avec des tilemaps pour unity
- Génération procédurale et Créativité Computationnelle, sur la chaîne de Matthew Guzdial https://www.youtube.com/channel/UCWNuToQtu_EGUWo4Ct-2aFQ
- Utilisation d'algorithmes génétiques
- Wave Function Collapse pour Unity: <https://selfsame.itch.io/unitywfc>
- Wave Function Collapse pour des images: <http://www.procjam.com/tutorials/wfc/>