

Représentation visuelle de l'*expressive range* d'un générateur procédural

Direction de thèse : Axel Buendia
Encadrement de thèse : Stéphanie Mader
Laboratoire CEDRIC (CNAM)

Mai 2023

1 Contexte

La génération procédurale est utilisée dans de nombreux jeux vidéo car elle permet de créer une grande variété de contenus. Elle peut par exemple être utilisée pour la création d'univers en 3D vastes et explorables par la-e joueur-se, ainsi que pour produire le contenu et les situations narratives et ludiques dans ce monde virtuel.

Les mécaniques et les règles encadrants cette génération procédurale sont pensées par les *game designers* et l'algorithme est écrit par les développeur-ses. La génération procédurale repose généralement sur des fonctions aléatoires ou plus récemment sur des algorithmes de *machine-learning* pour créer des contenus et des variations [Liu et al., 2020].

Nous nous intéressons particulièrement à l'utilisation de génération procédurale pour créer des expériences de jeu impossible sans celle-ci comme dans les jeux vidéo *Dwarf Fortress*, *Minecraft*, ou *Spelunky*. En effet, la génération procédurale ouvre des possibilités pour les *game designers* en leur permettant d'explorer la création d'expériences de jeu reposant sur de larges possibilités. Elle peut par exemple permettre aux joueur-ses de faire des choix qui impacteront la suite de leur aventure, de prendre des décisions dans un monde complexe simulé qui évoluera en fonction de ces décisions. Cet aspect est similaire aux enjeux de la narration interactive qui vise à donner la possibilité à l'ordinateur de générer la suite de l'histoire ou tout du moins donner l'impression que l'histoire s'adapte aux actions de la personne qui joue. Par exemple, *Facade* est un jeu vidéo qui génère des réponses en fonction des phrases écrites au clavier par la personne qui joue [Thompson, 2020]. Le jeu enchaîne plusieurs situations pensées par les développeur-ses mais invisibles pour la-e joueur-se [Mateas and Stern, 2005]. Ces situations s'apparentent ainsi à des niveaux dans le sens où elles forment un ensemble qui constitue le jeu vidéo.

Dans ce cadre, du point de vue des *game designers*, la génération procédurale au sens large est un outil de création. Selon les outils, leur niveau de contrôle sur la génération est plus ou moins élevé. Gillian Smith propose d'en distinguer quatre types du point de vue d'un-e joueur-se interagissant avec des jeux vidéo contenant ce type de moteur génératif [Smith, 2014]. Mais la même échelle peut être utilisée pour parler de la conception :

- Aucune participation humaine
- La participation humaine se limite à définir des paramètres
- La participation humaine consiste à évaluer le contenu généré pour orienter l'algorithme
- Co-participation entre l'humain et la machine pour générer le contenu

L'objectif de la participation humaine est de paramétrer, orienter ou collaborer avec l'algorithme de génération procédurale pour que les contenus et variations de contenus de celui-ci correspondent à des intentions d'expérience qu'il n'est pas possible à ce jour de transmettre immédiatement à la machine. Un générateur procédural possède un *expressive range*, soit le style et la variété générés [Smith and Whitehead, 2010], et l'objectif de cette collaboration est de forger cet *expressive range*.

En effet, le point fort de la génération procédurale est aussi sa problématique fondamentale : comment saisir, évaluer, et itérer l'*expressive range* généré par un algorithme ? A fortiori, pour un *game designer*, comment vérifier que l'algorithme de génération procédurale et son paramétrage soutiennent les intentions d'expérience de jeu ?

La génération procédurale ne doit pas rendre le jeu fade : les contenus ne doivent pas être trop répétitifs, la difficulté équilibrée, les variations perceptibles par les joueur-ses, etc. et dans le même temps, la génération procédurale ne doit pas créer d'événements, d'objets ou de situations s'écartant beaucoup des intentions d'expérience de jeu. Cet équilibre entre variation et uniformisation est un enjeu de conception majeur pour les *game designers*.

2 Objectifs et enjeux

Notre objectif est d'étudier, si et comment, un outil informatique de représentations visuelles de cet *expressive range* aide les *game designers* à se saisir plus rapidement et efficacement de la génération procédurale.

Si les outils de génération sont nombreux, il existe encore peu d'outils visant à évaluer automatiquement l'espace généré procéduralement à destination des *game designers*. Danesh, un *plugin* s'ajoutant au moteur de jeu vidéo *Unity*, propose une représentation visuelle d'un algorithme de génération procédurale, mais reste limité à *Unity* [Cook et al., 2022]. Smith et Whitehead ont aussi développé un outil de visualisation pour Launchpad, un éditeur de niveau 2D qu'ils avaient développé [Smith and Whitehead, 2010].

Ainsi, les outils existants sont souvent dédiés et liés à un type de jeu vidéo ou à un moteur / éditeur de niveaux.

Ainsi, cette thèse vise à développer un outil informatique permettant d'afficher différents types de représentations visuelles de l'*expressive range* d'algorithmes de génération procédurale. L'objectif est de permettre aux *game designers* de cerner plus rapidement l'*expressive range* du générateur et les effets des modifications de paramétrage effectué, notamment en montrant des cartographies de ses possibilités et de ses limites, mais aussi pour montrer les contours avec les générations les plus extrêmes ou les parcours de joueur·ses les plus extrêmes. Par exemple, que se passe-t-il si un·e joueur·se prend tous les mauvais choix ?

Une mise en forme visuelle a un intérêt non négligeable lorsqu'il s'agit de donner du sens à des informations [Baker et al., 2009]. Les *game designers* et développeur·ses pourront alors l'utiliser autant en conception qu'en analyse, offrant de nouvelles façons de concevoir et réfléchir un jeu vidéo [Neil, 2015].

Un des nouveaux enjeux de la génération procédurale est lié à la génération basée sur des techniques d'apprentissage comme le *deep learning* [Liu et al., 2020]. Notre outil pourrait alors s'adapter pour les évaluer aussi.

De plus, il permettrait à des studios de jeux vidéo avec un budget et un temps limité d'économiser des ressources en *playtests* ou développement de leur propre outil. Aussi, dans le cadre de la génération procédurale, un algorithme peut aussi refléter les biais de ses *designers* [Smith, 2016]. Un tel outil pourrait alors prendre en compte ce fait pour les identifier à la source, permettant ainsi de rendre un jeu vidéo plus inclusif, c'est-à-dire s'adressant à toutes et à tous.

Enfin, pour permettre à l'outil de s'intégrer dans différents *pipelines* de développement indépendamment des outils déjà présents, nous visons le développement d'un outil autonome prenant en entrée des traces de données issues du vérificateur du générateur procédural, d'un·e joueur·se synthétique ou de *playtests* pour afficher des informations sur l'*expressive range* du générateur.

3 Échéancier

La procédure est divisée en quatre étapes de travail :

- Revue de littérature sur l'existant dans le domaine de la génération procédurale de contenu ainsi que les différents types d'outil de visualisation liées aux jeux vidéo. Il sera accompagné d'entretiens et de questionnaires à destination de *game designers*, développeur·ses et chercheur·ses qui permettront de mieux rendre compte des besoins dans ce domaine pour élaborer le design de l'outil.
- Développement de l'outil informatique avec plusieurs représentations visuelles d'un algorithme de génération procédurale.

- Évaluation de l’outil informatique auprès de *game designers*.
- Rédaction de la thèse.

Références

- [Baker et al., 2009] Baker, J., Jones, D., and Burkman, J. (2009). Using visual representations of data to enhance sensemaking in data exploration tasks. *Journal of the Association for Information Systems*, 10(7) :533–559.
- [Cook et al., 2022] Cook, M., Gow, J., Smith, G., and Colton, S. (2022). Dadesh : Interactive tools for understanding procedural content generators. *IEEE Transactions on Games*, 14(3) :329–338.
- [Liu et al., 2020] Liu, J., Snodgrass, S., Khalifa, A., Risi, S., Yannakakis, G. N., and Togelius, J. (2020). Deep learning for procedural content generation.
- [Mateas and Stern, 2005] Mateas, M. and Stern, A. (2005). Procedural authorship : A case-study of the interactive drama façade.
- [Neil, 2015] Neil, K. (2015). *Game Design Tools : Can They Improve Game Design Practice ?* These de doctorat, Paris, CNAM.
- [Smith, 2014] Smith, G. (2014). Understanding procedural content generation : a design-centric analysis of the role of PCG in games. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 917–926, Toronto Ontario Canada. ACM.
- [Smith, 2016] Smith, G. (2016). A Proceduralist View on Diversity in Games. *Journal of Games Criticism*, 3(a).
- [Smith and Whitehead, 2010] Smith, G. and Whitehead, J. (2010). Analyzing the expressive range of a level generator. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games*, pages 1–7, Monterey California. ACM.
- [Thompson, 2020] Thompson, T. (2020). The story of facade : The ai-powered interactive drama | ai and games 49.