

# Les réseaux antagonistes génératifs dans les jeux vidéos

Sarah Pirard

Décembre 2022

## 1 Introduction

Qu'est-ce qu'un réseau antagoniste génératif? Un réseau antagoniste génératif, également communément appelé "GAN" (generative adversarial network) est une classe d'algorithme qui permet de créer des images de synthèse à partir d'échantillons. Le principe est de mettre en opposition deux réseaux, en premier lieu le générateur, qui génère une image au discriminateur sur base d'un échantillon qui doit détecter si l'image est réelle ou si elle résulte de la création du générateur (AGGARWAL, MITTAL et BATTINENI). Ce processus a été mis en place en 2014 par (GOODFELLOW et al.). Il en existe de nombreuses variantes à ce jour (Asimopoulos et al., 2022). Les GAN sont actuellement utilisés dans de nombreux domaines notamment dans l'imagerie médicale et le traitement d'image et de vidéo. Le domaine qui nous intéresse dans cet article est le domaine des jeux vidéos (ASIMOPOULOS et al.).

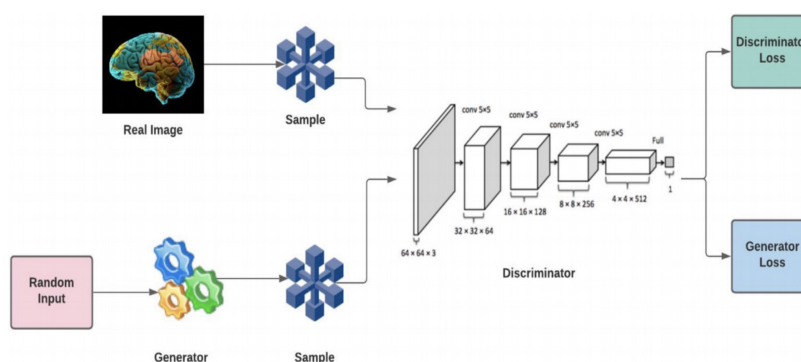


FIGURE 1 – fonctionnement d'un réseau antagoniste génératif

## 2 Quelles applications dans les jeux vidéos?

Les generative adversarial networks se sont montrés particulièrement efficaces en matière de PCG (Procedural Content Generation) dans le déploiement de nouveaux niveaux de jeux. Le PCG peut être utilisé pour générer des niveaux de jeu, des règles ou encore des textures pour les éléments graphiques. Il permet notamment de créer des mondes de jeux complexes en évitant de produire une surcharge de travail conséquente aux designers graphiques (Hendrikx et al, 2011). Cette approche a été récemment théorisée sous le nom de "Procedural Content Generation via Machine Learning" (PCGML) (Summerville et al.2018, Justesen et al. 2019; Guzdial et al 2018). Diverses systèmes d'architectures ont été utilisés en PCGML afin de générer du contenu de jeu. Les GAN ont gagné en popularité dans la production de jeux vidéos notamment grâce au fait qu'ils peuvent produire des environnements de jeux aléatoires sans nécessiter de supervision par un tiers. La méthode la plus accessible est basée sur une méthode de génération de nombres pseudo-aléatoire (PRNG). Les concepteurs utilisent ces modèles depuis les années quatre-vingts afin de générer des univers assez vastes, mais une démarche similaire est utilisée également pour concevoir les textures d'objets de jeu (KUMARAN, MOTT et LESTER).

Les GANs possèdent plusieurs caractéristiques clés qui permettent de les différencier dans la création de jeux vidéos des autres systèmes d'architecture. En premier lieu, ceux-ci permettent la modélisation de différents niveaux de jeux d'après une seule source. En appréhendant un modèle basé sur les GANs, il est possible de créer divers niveaux de jeux avec un gameplay similaire. Après avoir été joué, le niveau est ajouté au générateur comme exemple à confronter au discriminateur, ce qui permet au système de générer des niveaux de jeux continuellement. Une autre approche permet de se baser sur le gameplay d'un joueur pour générer progressivement des niveaux en utilisant des actions afin de produire un corpus (ACORNLEY).

### **3 Les différents types de réseaux antagonistes génératifs**

Les réseaux antagonistes génératifs étant particulièrement utilisés dans les jeux vidéos, un état des lieux est nécessaire. Il est également intéressant de savoir que les GANs sont en constante évolution, et de nouvelles structures sont créées de manière régulière selon les besoins. Un GAN particulièrement utilisé dans le milieu des jeux vidéos est le Deep Convolutional GAN, également nommé DCGAN. Il se montre particulièrement utile dans la création de nouveaux décors, bien que de nouvelles architectures se soient montrées plus efficaces par la suite, telles que les Spatial GAN, ou SGAN (JETCHEV, BERGMANN et VOLLGRAF)

D'autres systèmes d'architectures peuvent être également récurrents, tels que les PROGANs, qui sont basés sur l'expansion du réseau afin d'améliorer la stabilité et la qualité du réseau, le MSG-GAN, une variante dans laquelle le générateur et le discriminateur sont connectés. Il en existe de bien plus spécifiques également, tels que le Style-Based Generator GAN, qui permet de produire un nombre illimité de faux visages humains, ou le Time-Serie GAN, qui génère des séries réalistes dans différents domaines. D'autres systèmes seront abordés par après, étant des systèmes distinctifs, leur fonction sera mentionnée par après (ASIMOPOULOS et al.).

## 4 La synthèse des textures

Exprimé de manière théorique, une texture est la répétition d'une image d'une façon aléatoire. Le but dans la génération de textures lors de la création de jeux vidéos, et notamment la création du fond, est de capturer un échantillon de texture, dont le système permet la répétition de manière arbitraire. Deux méthodes différentes peuvent être utilisées. La première méthode consiste à replacer des échantillons visuels de textures afin de préserver ses propriétés visuelles. Cependant, cette méthode ne permet pas de produire une texture à proprement parler, mais simplement de réitérer un exemple encore et encore. La seconde méthode est de confronter des images statistiquement similaires afin de créer un fond avec une texture convaincante (JETCHEV, BERGMANN et VOLLGRAF).

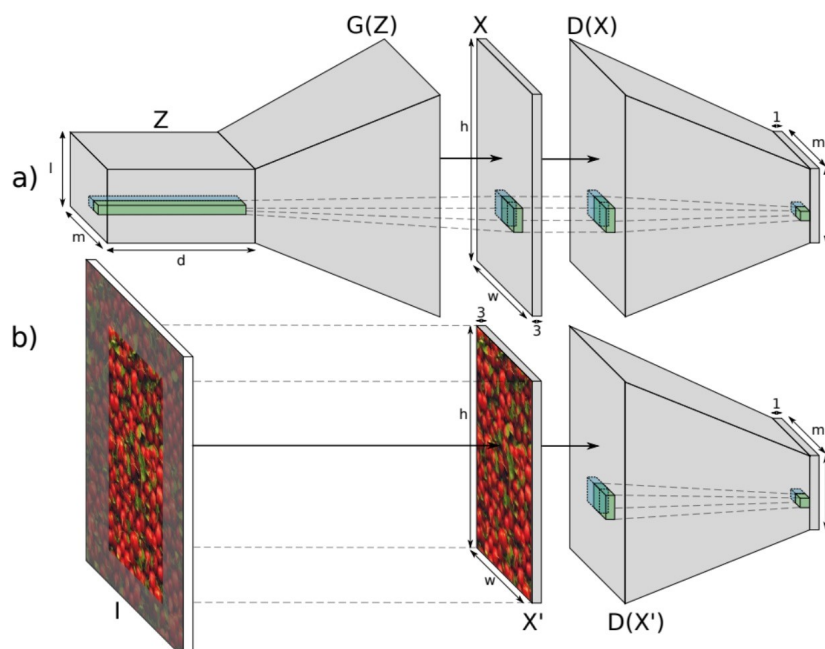


FIGURE 2 – modélisation de la création de texture par réseau antagoniste génératif

En 2015, les DCGANs (Deep Convolutional Generative Adversarial Network) ont été développés et ont permis de palier aux méthodes précédentes de manière plus efficace, quoique le manque de données tri-dimensionnelles occasionnaient parfois des manques de cohérences au niveau de l'altitude (RADFORD, METZ et CHINTALA). L'utilisation de Spatial GANs par après permet de générer des fonds avec de plus hautes altitudes beaucoup plus convaincantes (et).

## 5 La synthèse d'avatars

La création d'avatar prend une grande place dans les jeux vidéos, principalement dans les RPG (Role-Playing Game). La majeure partie des jeux vidéos de ce type permettent la personnalisation des avatars à l'aide de multiples paramètres customisables. Deux méthodes sont principalement utilisées aujourd'hui : le 3DMM (3D Morphable Model) et le F2P (Face-to-Parameter translation), le second étant plus facilement applicable au milieu des jeux vidéos que le premier, qui manque de profondeur au niveau du physique (SHI et al.).

Des chercheurs ont donc cherché à simplifier le processus en permettant la synthèse d'un avatar à l'aide d'une simple photo appliquée à un GAN. La difficulté réside dans le fait d'appliquer un visage neutre à l'avatar à l'aide de photo qui ne le sont pas forcément. Ce système a été appelé "PokerFace-GAN". Le rôle du discriminateur du GAN est de déterminer si l'expression du personnage est neutre ou pas. Le générateur applique alors différentes images au discriminateur jusqu'à l'obtention de la "poker face".

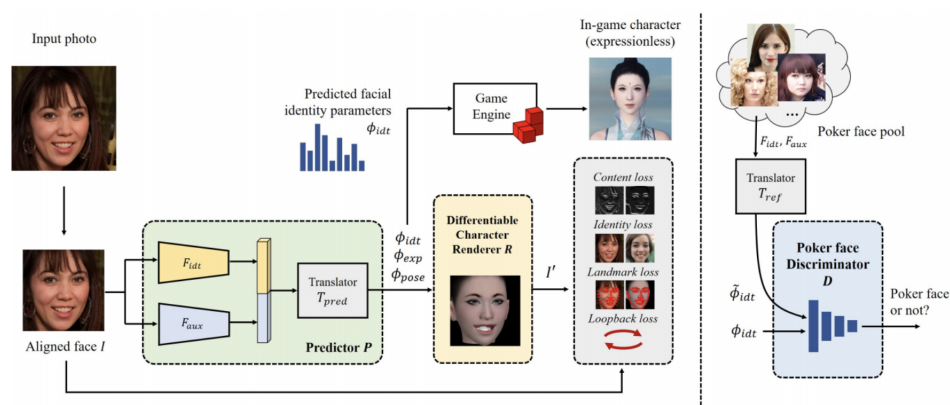


FIGURE 3 – synthèse d'un avatar émotionnellement neutre

## 6 La génération automatique de nouveaux niveaux

Il a été prouvé qu'afin qu'un jeu puisse être agréable, son niveau de difficulté ne doit être ni trop facile, ni trop difficile (RAJABI et al.). La diversité de style de jeux vidéos rend laborieux l'ajustement de la difficulté. Si en comparant la génération de niveaux de jeu par ordinateur et réalisé par un être humain, le processus chronophage que cela représente pour les développeurs les ont néanmoins poussé à générer des algorithmes permettant l'automatisation du procédé.

De plus, l'utilisation croissante de GANs, et plus particulièrement des DC-GANs dans la création d'image tend à automatiser le processus. En effet, le potentiel de la méthode permet un gain de temps non négligeable, ainsi qu'une marge d'erreur particulièrement réduite. Une procédure afin de réaliser différents niveaux consiste à réaliser des niveaux pré-designés, à les soumettre au générateur du GAN qui l'oppose alors au discriminateur. Le processus a été utilisé notamment pour le jeu Mario (RAJABI et al.).

Cependant, générer des niveaux de jeux de plateformes ne s'arrête pas uniquement aux décors. Les GANs ont également été utilisés afin de modifier la difficulté du jeu en fonction du joueur. Des paramètres tels que la vitesse, le type d'obstacles, leur nombre ainsi que le cycle selon lequel ils se présentent peuvent également être modifiés selon les constituants du niveau choisi (FUENTES PEREZ et al.).

Afin de réduire les désagréments liés à l'utilisation des GANs et ainsi automatiser la formation de niveaux de jeux, certains chercheurs se sont focalisés sur trois points. En premier lieu, la réduction du nombre d'informations nécessaires à apposer au discriminateur. En second lieu, l'augmentation de la qualité des niveaux produits; et enfin l'augmentation de la diversité des niveaux joués (TORRADO et al.). Les GANs utilisés auparavant pour la génération de niveaux étaient des DCGANs. Ceux-ci présentaient certains désavantages lié au fait qu'ils agissaient en couches convolutives : de ce fait, les couches d'un coin de l'écran n'avaient aucune corrélation avec celles d'un autre côté, étant situées trop loin l'une de l'autre, elles ne pouvaient générer un niveau jouable pour l'utilisateur.

Un nouveau type de GAN a donc été mis en place : le CESAGAN (Conditional-Embedding-Self-Attention Generative Adversarial Network). Cette évolution permettra des changements significatifs, tels que le nombre d'ennemis ainsi que le changement de l'avatar du joueur. En outre, cette architecture permet bien plus de contrôle sur le niveau d'application des PCG. Mais si ce mécanisme permet d'augmenter la jouabilité des niveaux, il nécessite néanmoins un nombre considérable de niveaux pour entraîner le discriminateur. C'est pourquoi des chercheurs ont mis en place un mécanisme d'amorce qui met en commun des

caractéristiques communes à un niveau. A chaque niveau les fonctionnalités du jeu sont contrôlées et une nouvelle série de niveaux est générée (TORRADO et al.).

## **7 Cas pratique : Doom**

Dans ce cas pratique, les chercheurs ont utilisés deux méthodes d'utilisation des GANs différentes. Dans la première, les image apposées au GAN ont été extraites d'un niveau créé de manière humaine, avec des caractéristiques de terrain, de hauteur et de position des objets. Le second GAN a été utilisé sans ces caractéristiques. Le but étant d'imiter la génération de niveaux de jeux similaire à la création humaine dans le domaine des jeux de tirs à la première personne, et le résultat fut concluant. (GIACOMELLO, LANZI et LOIACONO).

Les deux GANs utilisés étaient un conditional GAN pour le premier, et un unconditional GAN pour le second. Chaque niveau était représenté par ses caractéristiques les plus importantes : la zone, la hauteur du plafond, les objets, la segmentation de la pièce ainsi qu'un vecteur d'attributs qui permettait de décrire le niveau de manière efficace (le nombre de pièce, la largeur du périmètre, etc.). Pour évaluer les niveaux générés par les différents GANs, un algorithme SLAM a été utilisé. Les résultats ont démontrés qu'à force d'être entraîné, le conditional GAN s'améliore et génère des niveaux de qualité visuelle similaire à la création humaine.

## **8 Cas pratique : Mario**

Utilisant un corpus de niveau du jeu original Mario, ce cas pratique a permis de créer une variété de niveaux similaires. Cependant, la qualité des niveaux générés a été amélioré de manière significative grâce à l'application de la "Covariance Matrix Adaptation Evolution Strategy" (CMA-ES), qui a permis de générer au maximum les propriétés demandées. L'application se déroule en deux phases : la première le corpus de niveau est appliqué au générateur. La seconde phase les vecteurs latents qui produisent les niveaux sont recherchés. Comme dans d'autres cas, le but est de générer des niveaux de qualité similaires à la production humaine.

L'algorithme CMA-ES utilisé est un algorithme particulièrement utile dans le traitement de vecteurs évolutifs de nombres réels. L'application de cet algorithme a permis de tester les propriétés purement statistiques des niveaux générés, ainsi que sur les résultats des simulations de niveaux.

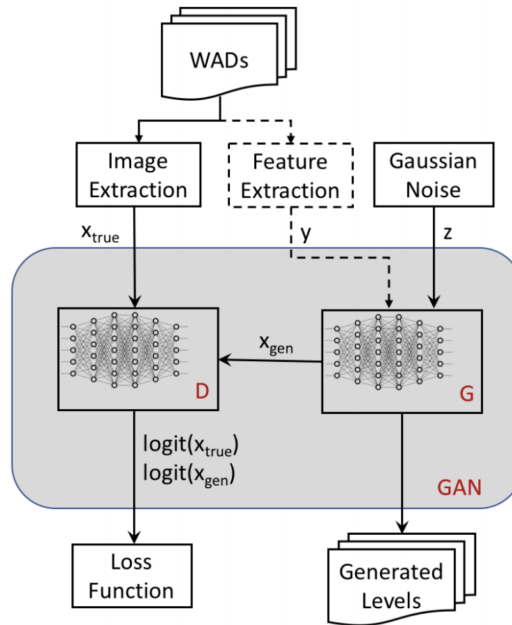


FIGURE 4 – principe de génération d’un niveau de jeu Doom par un réseau antagoniste génératif

## 9 Conclusion

En conclusion, si les applications des réseaux antagonistes génératifs sont multiples dans les jeux vidéos, celles-ci permettent la génération de différentes caractéristiques de ceux-ci de manière plus ou moins efficaces selon le cas étudié ou encore le domaine de spécialisation. Si les GANs se montrent particulièrement efficaces au niveau de la génération d’images et qu’il s’agit ici du domaine de spécialisation de ce système d’architecture, il reste néanmoins efficace dans d’autres domaines, qui sont par ailleurs en constante évolution. De plus, même si l’utilisation de réseaux antagonistes génératifs restent extrêmement utile dans la production de jeux vidéos car ils permettent d’optimiser le rendement, les résultats restent de qualité significativement inférieure à la production humaine en matière de production visuelle. Il serait donc probablement nécessaire soit d’apposer un corpus créé par l’humain, soit d’entraîner les GANs de manière plus significative, comme cela a pu être le cas dans l’étude sur Doom.



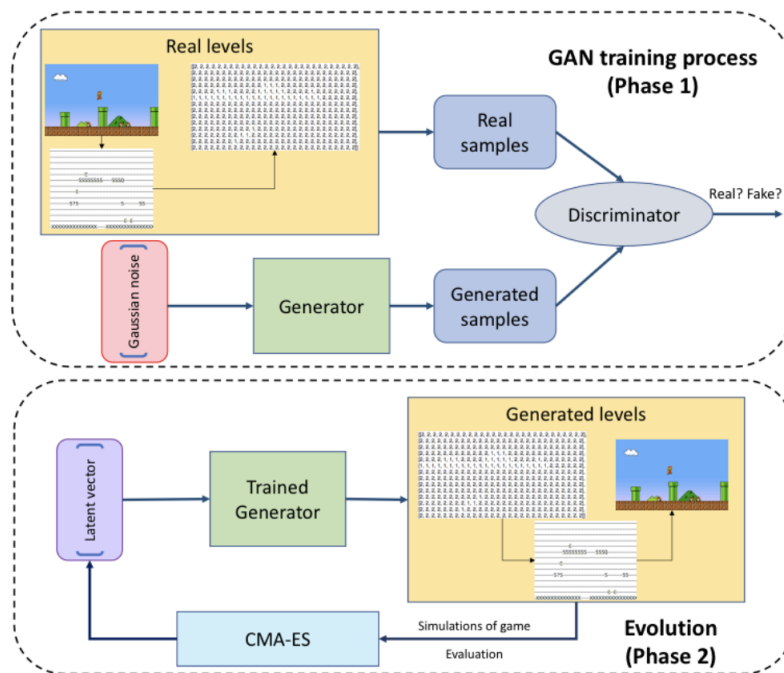


FIGURE 5 – principe de génération d'un niveau de jeu Mario par un réseau antagoniste génératif

## Références

### Références

- [1] Christopher ACORNLEY. « Using Generative Adversarial Networks to Create Graphical User Interfaces for Video Games ». In : *Proceedings of the 2021 International Conference on Multimodal Interaction*. ICMI '21: INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION. Montréal QC Canada : ACM, 18 oct. 2021, p. 802-806. ISBN : 978-1-4503-8481-0. DOI : 10.1145/3462244.3481273. URL : <https://dl.acm.org/doi/10.1145/3462244.3481273> (visité le 17/12/2022).
- [2] Alankrita AGGARWAL, Mamta MITTAL et Gopi BATTINENI. « Generative adversarial network: An overview of theory and applications ». In : *International Journal of Information Management Data Insights* 1.1 (1<sup>er</sup> avr. 2021), p. 100004. ISSN : 2667-0968. DOI : 10.1016/j.jjimei.2020.100004. URL : <https://www.sciencedirect.com/science/article/pii/S2667096820300045> (visité le 15/12/2022).
- [3] Dimitrios ASIMOPOULOS et al. « Generative Adversarial Networks: a systematic review and applications ». In : *SHS Web of Conferences* (2022).

- [4] Lizeth Joseline FUENTES PEREZ et al. « Dynamic game difficulty balancing in real time using Evolutionary Fuzzy Cognitive Maps with automatic calibration ». In : *Journal on Interactive Systems* 7.1 (18 nov. 2016), p. 1. ISSN : 2236-3297. DOI : 10.5753/jis.2016.668. URL : <https://sol.sbc.org.br/journals/index.php/jis/article/view/668> (visité le 02/01/2023).
- [5] Edoardo GIACOMELLO, Pier Luca LANZI et Daniele LOIACONO. *DOOM Level Generation using Generative Adversarial Networks*. 24 avr. 2018. arXiv : 1804.09154[cs,stat]. URL : <http://arxiv.org/abs/1804.09154> (visité le 17/12/2022).
- [6] Ian J. GOODFELLOW et al. *Generative Adversarial Networks*. 10 juin 2014. DOI : 10.48550/arXiv.1406.2661. arXiv : 1406.2661[cs,stat]. URL : <http://arxiv.org/abs/1406.2661> (visité le 01/01/2023).
- [7] Nikolay JETCHEV, Urs BERGMANN et Roland VOLLGRAF. *Texture Synthesis with Spatial Generative Adversarial Networks*. 8 sept. 2017. DOI : 10.48550/arXiv.1611.08207. arXiv : 1611.08207[cs,stat]. URL : <http://arxiv.org/abs/1611.08207> (visité le 17/12/2022).
- [8] Vikram KUMARAN, Bradford W MOTT et James C LESTER. « Generating Game Levels for Multiple Distinct Games with a Common Latent Space ». In : *Proceedings of the Sixteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-20)*. Sixteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. Palo Alto, California : AAAI Press, 19 oct. 2020, 360p. ISBN : 978-1-57735-849-7.
- [9] Alec RADFORD, Luke METZ et Soumith CHINTALA. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 7 jan. 2016. DOI : 10.48550/arXiv.1511.06434. arXiv : 1511.06434[cs]. URL : <http://arxiv.org/abs/1511.06434> (visité le 01/01/2023).
- [10] M RAJABI et al. « A dynamic balanced level generator for video games based on deep convolutional generative adversarial networks ». In : *Scientia Iranica* (16 nov. 2020), p. 1497-1514. DOI : 10.24200/sci.2020.54747.3897. URL : [http://scientiairanica.sharif.edu/article\\_22082.html](http://scientiairanica.sharif.edu/article_22082.html).
- [11] ryan rs spick ryan rs et james walker james. « Realistic and Textured Terrain Generation using GANs ». In : *European Conference on Visual Media Production*. CVMP '19: European Conference on Visual Media Production. London United Kingdom : ACM, 17 déc. 2019, p. 1-10. ISBN : 978-1-4503-7003-5. DOI : 10.1145/3359998.3369407. URL : <https://dl.acm.org/doi/10.1145/3359998.3369407> (visité le 17/12/2022).

- [12] Tianyang SHI et al. *Neutral Face Game Character Auto-Creation via PokerFace-GAN*. 17 août 2020. arXiv : 2008 . 07154[cs]. URL : <http://arxiv.org/abs/2008.07154> (visité le 02/01/2023).
- [13] Ruben Rodriguez TORRADO et al. *Bootstrapping Conditional GANs for Video Game Level Generation*. 3 oct. 2019. arXiv : 1910 . 01603[cs]. URL : <http://arxiv.org/abs/1910.01603> (visité le 17/12/2022).