

# ITESCIA – 2ème année de Manager en Ingénierie Informatique

Spécialité Gaming & Smart Tech

## La génération procédurale dans les jeux-vidéo



Présenté et soutenu par Pascal RAMOS

En binôme avec Kévin ZID

Session 2015-2018

# Sommaire

Sommaire.....	I
Abstract.....	2
Introduction.....	3
Développement de la partie recherche.....	4
1. Chronologie du procédural dans le jeu-vidéo.....	4
2. Techniques employées .....	6
a. Principes.....	6
b. Techniques procédurales avancées .....	9
3. Avantages et inconvénients .....	II
a. Avantages.....	II
b. Désavantages .....	13
Développement de l'étude personnelle .....	17
Conclusion.....	18
Sources .....	19
Bibliographie – Webographie .....	20
Index .....	21
Sommaire des annexes.....	22
Annexes .....	23

## **Abstract**

*To do*

# Introduction

*“Videogames arose out of a natural wish to make computers do fun things“*

Nolan Bushnell

C'est par cette citation de Nolan Bushnell, le fondateur d'Atari, que j'aimerais commencer ce mémoire de veille technologique traitant de la génération procédurale dans l'univers des jeux-vidéo. Car en fin de compte, qu'y aurait-il de plus amusant que de jouer dans un monde, qui comme le nôtre, nous ferait découvrir sans arrêt de nouvelles facettes. Et c'est en partie grâce à la génération procédurale qu'aujourd'hui nous sommes capables de créer des jeux-vidéo impressionnants de par leurs profondeurs et leurs réalismes.



La génération procédurale, c'est justement rendre les choses plus réalistes, plus vraies, qui nous impressionnent dans la réalité. Il est vrai qu'un monde généré sera toujours moins beau qu'un monde créé de la main d'artistes. Mais essayer de représenter par exemple l'Espace de cette manière nous prendrait des centaines d'années.

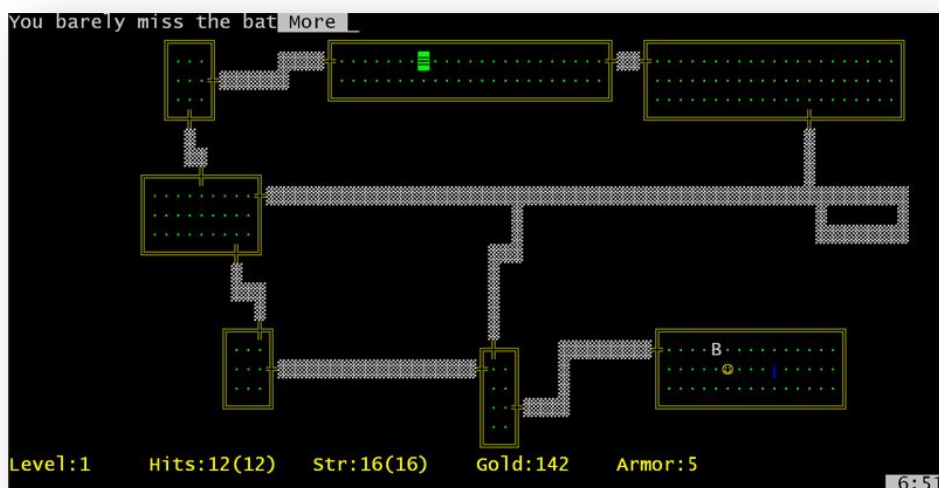
# Développement de la partie recherche

## 1. Chronologie du procédural dans le jeu-vidéo



Tout le monde pourrait penser que la génération procédurale est une technologie récente, et oui, en prenant en compte la taille des mondes générés de manière procédurale de nos jours (cf. No Man's Sky : 18 446 744 073 709 551 616 de corps célestes générés), difficile de croire qu'un jeu utilisant une telle technologie puisse tenir dans une cartouche de 20 kilo octets...

Et pourtant, la technologie procédurale remonte à bien longtemps, et a été employée pour la première fois par un jeu qui est d'ailleurs très connu pour sa difficulté et l'inspiration qu'il a donné à de nombreux autres jeux du genre, j'ai nommé : **Rogue** (1980).



Dans Rogue, le joueur doit descendre dans un vaste ensemble de souterrains pour y trouver l'Amulette de Yendor et la ramener à la surface. Pour chaque session de jeu, il ne dispose que d'une seule vie et d'une seule sauvegarde, empêchant les retours en arrière. Ce système de mort permanente est associé à un mécanisme de création de niveaux aléatoire, conduisant à des parties toujours différentes.

Peu après la découverte de cette technologie, un incontournable du jeu vidéo est né. Longtemps considéré comme l'un des jeux les plus innovants de l'histoire du jeu vidéo, on parle bien évidemment d'Elite (Mr.Micro, 1984). Jeu de combat et de commerce spatial où le joueur incarne un commerçant équipé d'un cargo armé. Ayant le choix de rester dans la légalité ou au contraire de se laisser embarquer dans la contrebande et la piraterie. Allant de planètes en planètes et de stations en stations, chacune possédant ses propres lois. Un jeu dans lequel on ne peut gagner, seulement progresser pour s'enrichir et donc d'acquérir de meilleurs vaisseaux.



Utilisant à bon escient la technologie procédurale, puisque nous avons tout de même pas moins de 8 galaxies disponibles pour un nombre de 2000 étoiles dans justement ... 20 kilo octets ! On peut parler de véritable prouesse technologique pour l'époque.



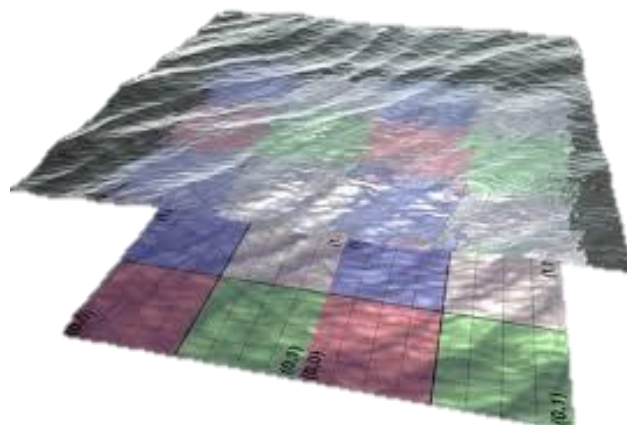
Le concept de niveaux toujours différents a été beaucoup apprécié par la communauté des joueurs et a donc été repris par les concepteurs puis amélioré bien des fois (notamment avec l'arrivée de la 3D isométrique). Nous sommes alors passés par des jeux comme **Diablo** (Blizzard North, 1997), **Terraria** (Re-Logic, 2011) ou même **Candy Crush** (King, 2012), le fameux et très connu jeu mobile.



## 2. Techniques employées

### a. Principes de base

La génération procédurale consiste finalement à créer du contenu (ressources ou fonctionnalités) à l'aide d'un algorithme et de paramètres qui lui sont passés.



Choisir la génération procédurale pour son jeu c'est dans un premier temps s'infliger de nombreuses heures de travail et de réflexion, mais cela facilite grandement les tâches suivantes, car elle permet d'ajouter ensuite du contenu bien plus facilement que de créer par exemple tous ses niveaux à la main.

Le grand avantage de cette solution est que le joueur aura chaque fois une phase de jeu unique. Tel Christophe Colomb découvrant l'Amérique, le joueur se sentira explorateur de ces niveaux extrêmement variés.

Une façon simple et efficace de captiver le joueur et de ne pas le lasser avec des événements scriptés ou répétitifs.



Cependant attention à ne pas produire l'effet inverse ! En effet, une surutilisation de cette technologie avec trop peu de textures différentes par exemple entrainerait des niveaux extrêmement grands mais toujours trop ressemblants. Donc répétitifs. (Voir annexe : “ Effet No man’s sky “)

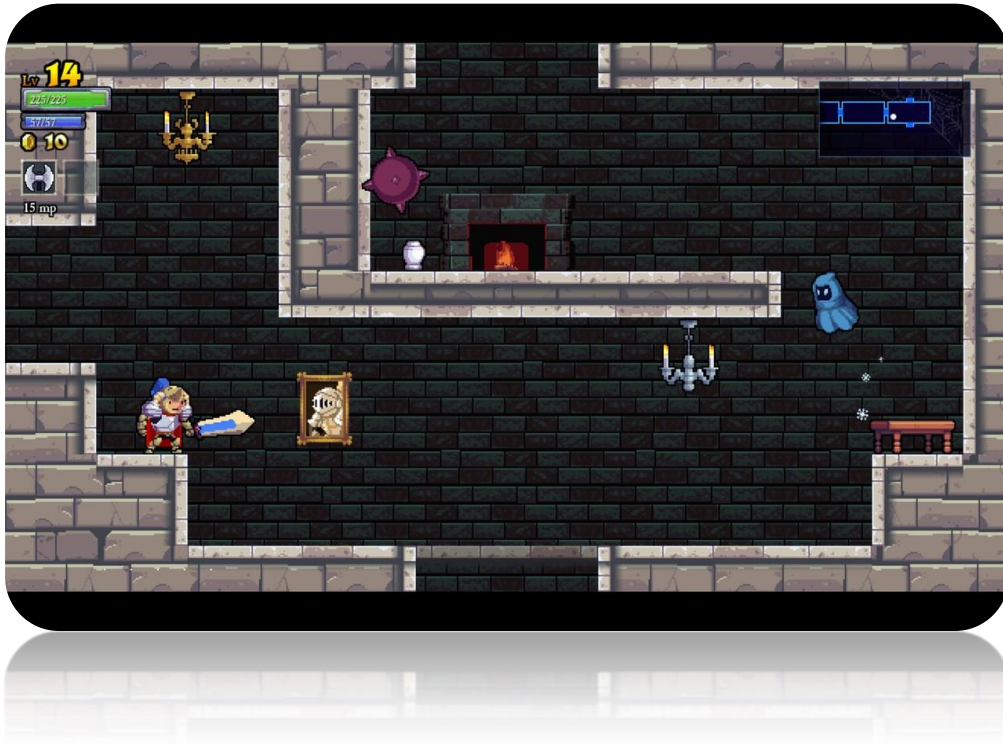
Côté maintenance, comme le jeu est généré par la machine, il est bien plus simple d'effectuer des tests unitaires et donc de tester automatiquement le bon fonctionnement de celui-ci.

Souvent les jeux procéduraux (notamment les jeux procéduraux 2D) utilisent un système de grille. Chaque grille possède une entrée et une sortie au minimum, le but étant pour le joueur d'atteindre cette ou ces sortie(s), tout en commençant par l'entrée.





Dans le but de challenger le joueur, tout un tas de mécanismes viendront entrer en interaction avec celui-ci afin de rendre plus complexe l'arrivée du joueur à cette ou ces sortie(s). On peut penser par exemple à un labyrinthe, à des monstres, à des pièges ou encore à des événements aléatoires. Et bien entendu il doit toujours exister un chemin afin d'aller de l'entrée à la sortie, sans quoi le niveau est impossible à terminer.



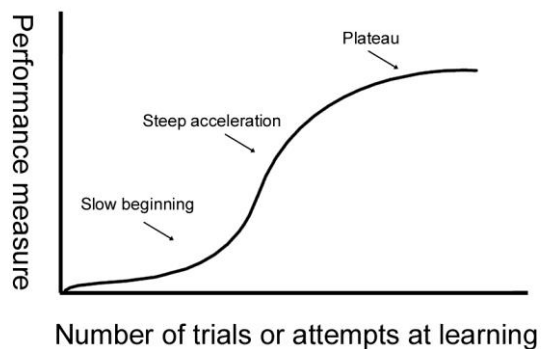
De nos jours, on utilise beaucoup la technologie de game analytics, vous savez, la technologie récupérant les données des anciennes parties des joueurs afin de rendre le jeu plus proches de leurs attentes. La génération procédurale permet également d'arriver à ce même but comme par exemple en adaptant le jeu en fonction des résultats précédents du joueur.

Prenons un cas où un joueur fait quatre parties et meurs à chaque fois.

Le joueur en mourant de nombreuses fois entretient une certaine frustration.

Le jeu est trop dur ; c'est trop répétitif de devoir refaire le même niveau ; Je meurs chaque fois au même endroit car ce n'est pas intuitif ;

Pour éviter de perdre notre joueur il est nécessaire de faire en sorte qu'au bout d'un certain nombre d'échecs, l'algorithme le prenne en compte et adapte le contenu au type de joueur.



C'est la même chose si le joueur trouve le jeu trop simple, et commence donc à s'ennuyer.

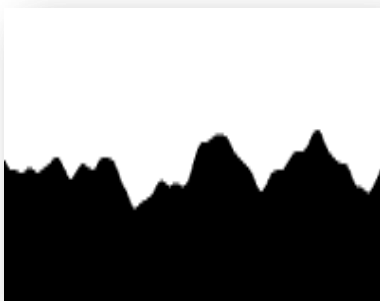
La génération procédurale peut donc être une forme de solution pour la « learning curve », la courbe d'apprentissage du joueur.

## b. Techniques procédurales avancées

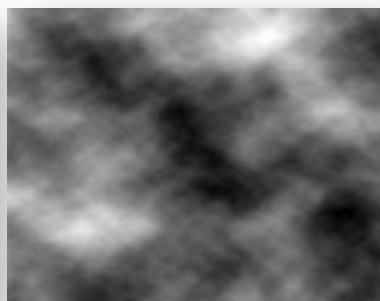
Les techniques procédurales dans le jeu-vidéo sont assez complexes c'est pourquoi nous ne verrons pas en détail chacune des technologies employées, cependant, de nos recherches ressortent plusieurs techniques intéressantes dont nous allons parler.

Tout d'abord, parlons d'une technique très utilisée dans la génération procédurale, elle se nomme le bruit de Perlin. On appelle un bruit toute image « parasitée » aléatoirement.

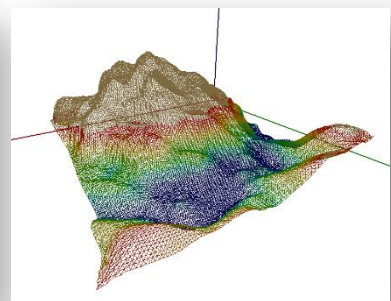
Voici un exemple de bruit de Perlin sur une, deux et trois dimensions.



Bruit de Perlin 1D



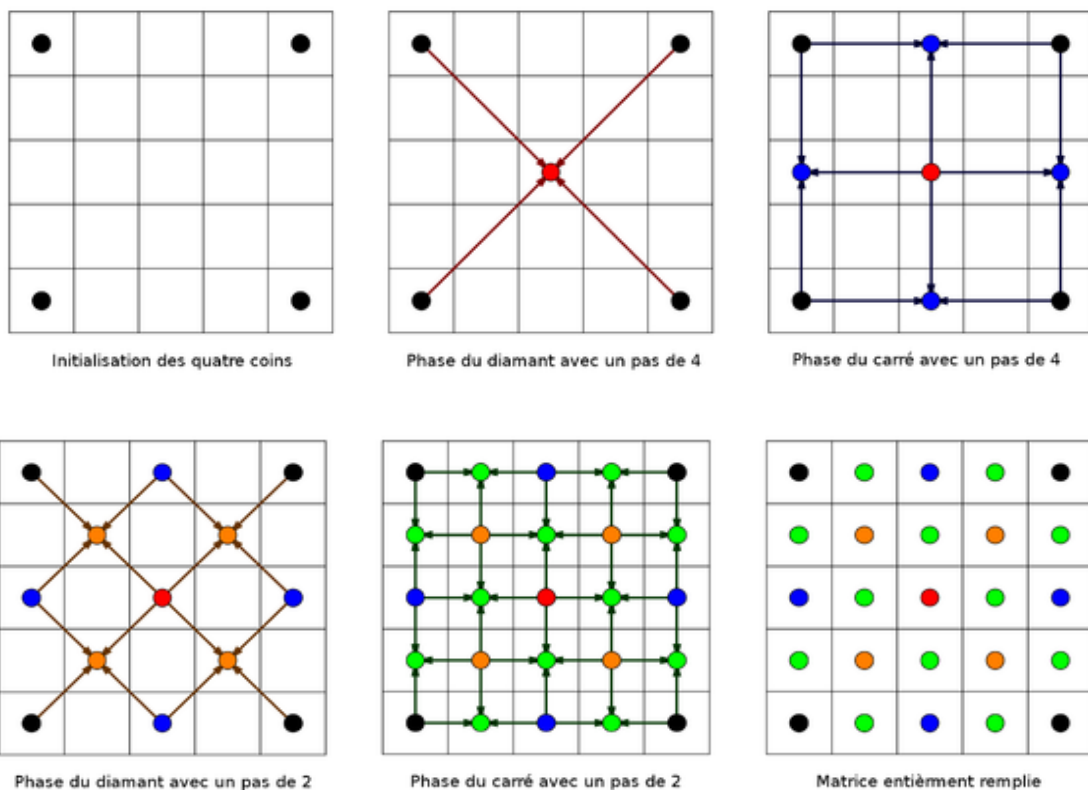
Bruit de Perlin 2D



Bruit de Perlin 3D

Comme vous pouvez le voir, le bruit de Perlin est principalement utilisé pour la génération de cartes de hauteur (heightmap). Mais un autre algorithme est très connu pour la génération de ces mêmes maps : L'algorithme de Fournier, midpoint displacement ou son amélioration : diamond-square algorithm (Algorithme diamond-carré).

Cet algorithme permet, en cinq étapes, de placer tous les points d'une matrice et donc de générer des cartes complètes. Voici les cinq étapes de la procédure.



Et voici un exemple de rendu avec l'application de l'algorithme :



Midpoint displacement



Diamond-square

### 3. Avantages et inconvénients

#### a. Avantages

##### La rentabilité

L'un des premiers avantages à retenir concernant la génération procédurale, est la rentabilité. En effet, il est possible de créer des environnements gigantesques avec un algorithme prédéfini, cela convient parfaitement aux jeux indépendants, ayant un budget limité. C'est plus rentable quand on pense à **Daggerfall** (Bethesda Softworks, 1996) qui englobe une surface équivalente à 161 600 km<sup>2</sup>, contrairement à un jeu comme **Skyrim** (Bethesda Game Studios, 2011) entre 35 et 40 km<sup>2</sup> fait à la main, le coût ne serait pas le même si **Daggerfall** devait être fait à la main. Et à l'époque encore moins sachant que nous étions limité par l'espace de stockage. Un jeu qui peut être terminé rapidement, comme un jeu linéaire de notre époque en 3D, est moins intéressant pour les développeurs en matière de création de niveau, tandis qu'un jeu exploitant la génération procédurale pousse le joueur à passer plus de temps sur le jeu pour découvrir toutes les possibilités, niveaux générés aléatoirement par exemple. C'est donc plus rentable pour le développeur.



## La diversité

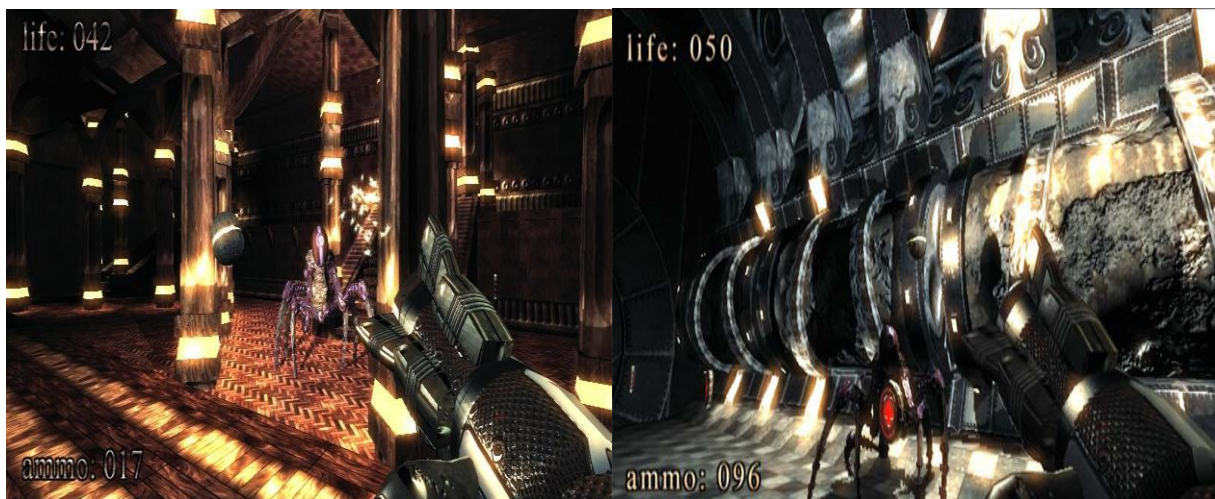
Avec la création de niveaux aléatoire, chaque niveau est différent, que ce soit en matière de décors, d'objets, d'ennemis etc. Cela pousse le joueur à rejouer et à refaire les niveaux, surtout lorsque l'on pense à **Diablo** avec les quêtes à accomplir qui ne sont jamais les mêmes. Le joueur peut passer plus de temps pour essayer de terminer le jeu puisqu'il ne connaît pas toutes les quêtes, et s'il meurt, il sera toujours captivé par l'inconnu. Aussi, concernant les ennemis, le joueur sera pris d'envie de les Farmer pour obtenir des objets rares, objets lâchés par les ennemis suivant toujours une règle de génération aléatoire. Ils ne donneront jamais les mêmes objets, ce qui évite de connaître d'avance quel type d'ennemi à abattre pour obtenir tel type d'objet.





## L'économie d'espace de stockage

Il fût un temps dans le jeu vidéo où l'espace de stockage était très limité. La capacité des disques durs n'était pas la même, nous parlions de méga-octets à l'époque de **Diablo** tandis qu'aujourd'hui nous comptons en giga-octets pour des jeux récents comme **GTA**, **Far Cry** etc. La taille des jeux augmente exponentiellement en fonction de l'évolution grandissante des supports. Un exemple pour montrer que la génération procédurale peut permettre de réduire drastiquement la taille d'un jeu, notamment avec les textures, avec la démo **.kkrieger** ([.theprodukt, 2004](#)) qui a pu tenir sur 96 kilo-octets pour 7 minutes de jeu. Un exploit quand on visualise cette démo et la qualité des décors.



### b. Désavantages

#### La répétitivité

Concernant les jeux de l'époque, qui utilisaient beaucoup la génération procédurale pour éviter de peser lourd sur les espaces de stockage très limité. Les algorithmes utilisés généraient des environnements parfois trop répétitifs, on pense notamment à **Daggerfall**. En réalité, le contenu utilisé par les algorithmes était placé plusieurs fois dans différents endroits du monde. Ce qui donnait une forte impression de déjà-vu pour le joueur. Certes le

monde était vaste et gigantesque, mais la taille nuis rapidement à l'expérience du joueur, surtout si les décors deviennent répétitifs. On imagine mal parcourir un monde aussi grand avec quasiment les mêmes décors à chaque kilomètre parcouru. Cela concerne également les quêtes, qui ne se démarquent pas dans l'objectif même, comme par exemple « tuer la créature X », « tuer la créature Z », « Trouver un objet A et le rapporter au personnage D » ou « Trouver un objet B et le rapporter au personnage E ». Les quêtes se ressemblent parfois trop souvent, cela devient répétitif en plus du décor.



### Le manque de richesse

En s'appuyant sur l'exemple donné pour la répétitivité, on peut aussi dire que cela induit un manque de richesse. En effet, un algorithme utilisé pour générer des éléments du jeu de manière aléatoire, ne se préoccupe pas de l'expérience de jeu mais seulement d'apporter du contenu. En règle générale, c'est le Level Designer qui va apporter cette expérience de joueur notamment en intégrant une ambiance particulière, avec des décors placés judicieusement pour soit déranger le joueur dans l'avancement, soit le reconforter et le faire sourire. Le Level Designer peut ainsi se mettre à la place du joueur et lui transmettre des émotions contrairement à un algorithme de génération procédurale.



Commencer avec de petites zones est préférable quand on utilise de la génération procédurale, on peut tester et corriger les bugs plus rapidement, et ensuite transférer l'ensemble sur une surface plus étendue. C'est ce que Chris Roberts (*Star Citizen*, CIG) a privilégié pour son jeu en indiquant qu'il était préférable de travailler sur une centaine de systèmes planétaires pour l'instant ( tout de même ), que de laisser les joueurs se perdre à travers moult étoiles dans moult galaxies différentes. Enfin, cela oblige le développeur à prendre son temps, au départ, car ensuite on peut générer l'environnement très rapidement lorsque l'on a repéré les bugs sur la première partie. C'est ce qui distingue un jeu utilisant la génération procédurale d'un jeu fait à la main, où le rythme de développement reste régulier mais plus long.



# Développement de l'étude personnelle

*To do*



# Conclusion

Le concept de génération procédurale dans le domaine du jeu-vidéo nous porte à croire qu'un avenir prometteur se tient devant nous.

Une nouvelle ère, source inépuisable et infinie, un moyen de rendre le joueur captivé par un monde qui ne cesse d'évoluer et de se diversifier, où il est quasiment impossible de tout découvrir et terminer un jeu à 100 % ou faire un speedrun digne de ce nom capable de nous tenir devant l'écran pendant 1 heure grand maximum.

C'est par ce concept que les développeurs peuvent se faciliter la tâche et donner vie à une expérience nouvelle, où le potentiel est énorme et peut rassembler toute une communauté de joueurs, allant bien plus loin qu'un simple MMORPG, où l'envie de rejouer, et de rejouer encore, est un luxe dont on ne peut se priver.

De ce fait, cette expérience convient parfaitement aux joueurs les plus chevronnés, toujours en quête de plus de défis et de challenges, terminer un jeu qui offre une multitude de possibilités, poussant l'aléatoire à devenir lui-même joueur et attirer le possesseur de la manette dans les abysses et l'amener à se dépasser.

Encore faut-il rendre cette expérience à la hauteur, et non pas créer un monde quasi-infini où l'infini devrait être plutôt limité, pour éviter au joueur de s'ennuyer devant des décors, certes pas exactement similaire, mais dont l'ambiance et le gameplay porte à se répéter un certain nombre de fois pour obtenir le même résultat.

# Sources

<http://www.gamelove.com/news-actu/la-generation-procedurale-judicieuse-pour-la-creativite-144846>

<http://www.numerama.com/sciences/133517-la-generation-procedurale-ou-comment-le-jeu-video-devient-infini.html>

<https://tel.archives-ouvertes.fr/tel-01196438/document>

<http://www.benicourt.com/blender/tag/generation-procedurale/>

<http://khayyam.developpez.com/articles/algo/perlin/>

<http://jeux.developpez.com/tutoriels/evenements/game-connection-2013-europe/conferences/Generation-procedurale/>

<http://linuxfr.org/news/je-cree-mon-jeu-video-e10-generation-procedurale-de-carte-partie-1#diamant-carr%C3%A9-diamond-square>

<http://toutcequibouge.net/2014/09/la-generation-procedurale-dans-le-jeu-video/>

<http://indius.org/dossiers/generation-procedurale-jeu-video/>

<http://khayyam.developpez.com/articles/algo/perlin/>

<http://starcitizenkane.com/generation-procedurale/>

# Bibliographie – Webographie

*To do*

## Index

*To do*

## Sommaire des annexes

*To do*



## **Annexes**

*To do*