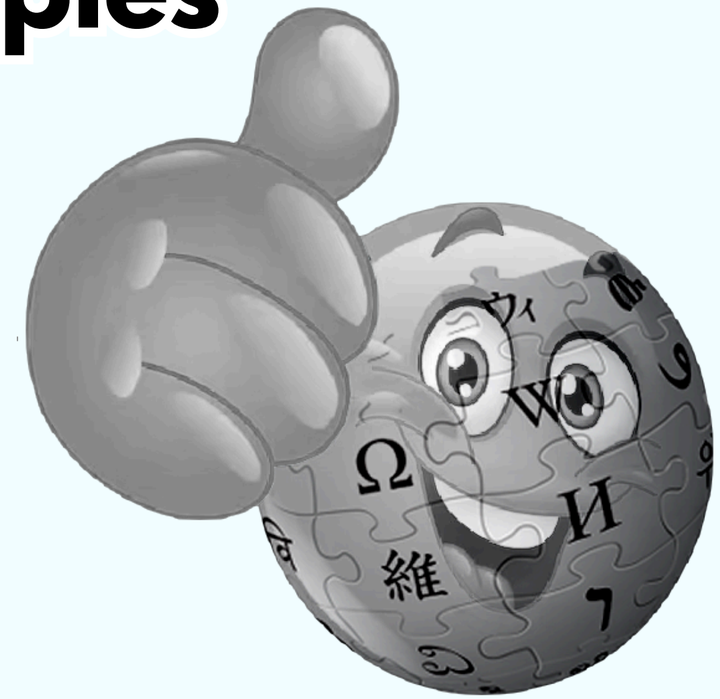
FORMATION

GENERATION PROCEDURALE

Par Crizomb

Sommaire :

Introduction et exemples



Définition :

Création de contenu numérique

- **niveau de jeu**
- **modèles 3D**
- **dessins / texture 2D**
- **animation**
- **musique**
- **histoire / dialogues**

En grande quantité, de manière automatisée répondant à un ensemble de règles définies par des algorithmes.

Introduction

Définition :

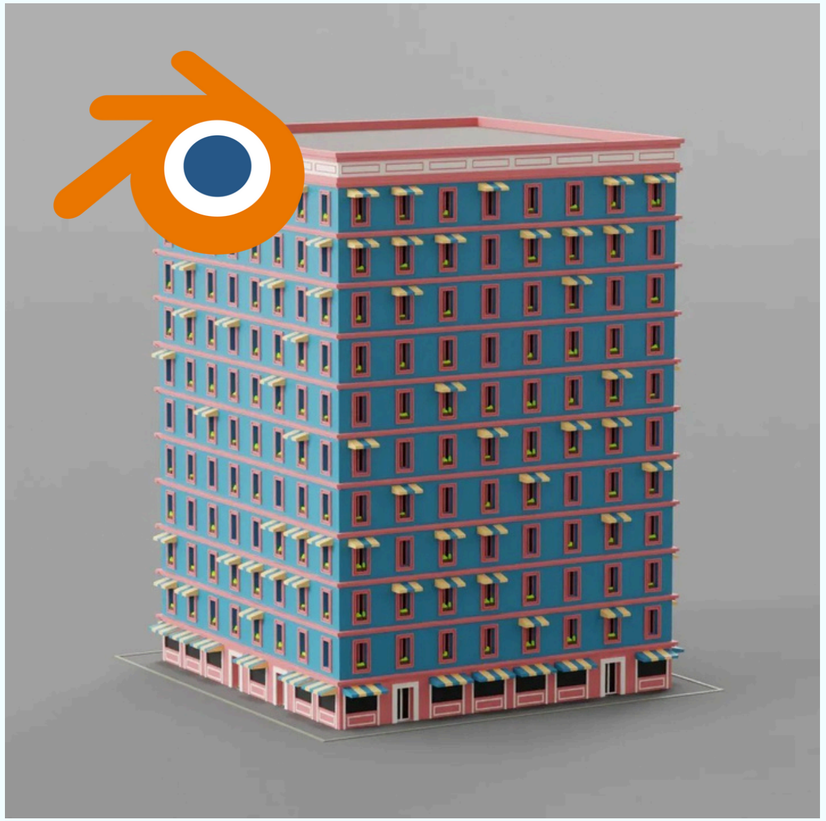
Création de contenu numérique

- **niveau de jeu**
- **modèles 3D**
- **dessins / texture 2D**
- **animation**
- **musique**
- **histoire / dialogues**

En grande quantité, de manière automatisée répondant à un ensemble de règles définies par des algorithmes.

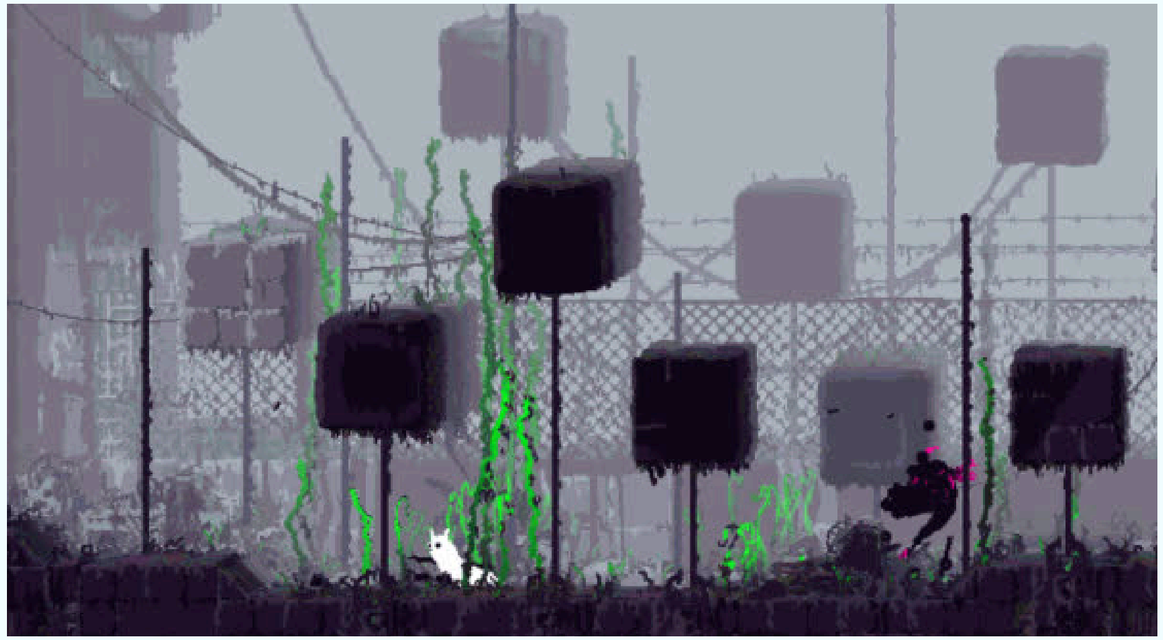
Exemples génération procédurale (1/3)

Niveau de jeu : Monde minecraft

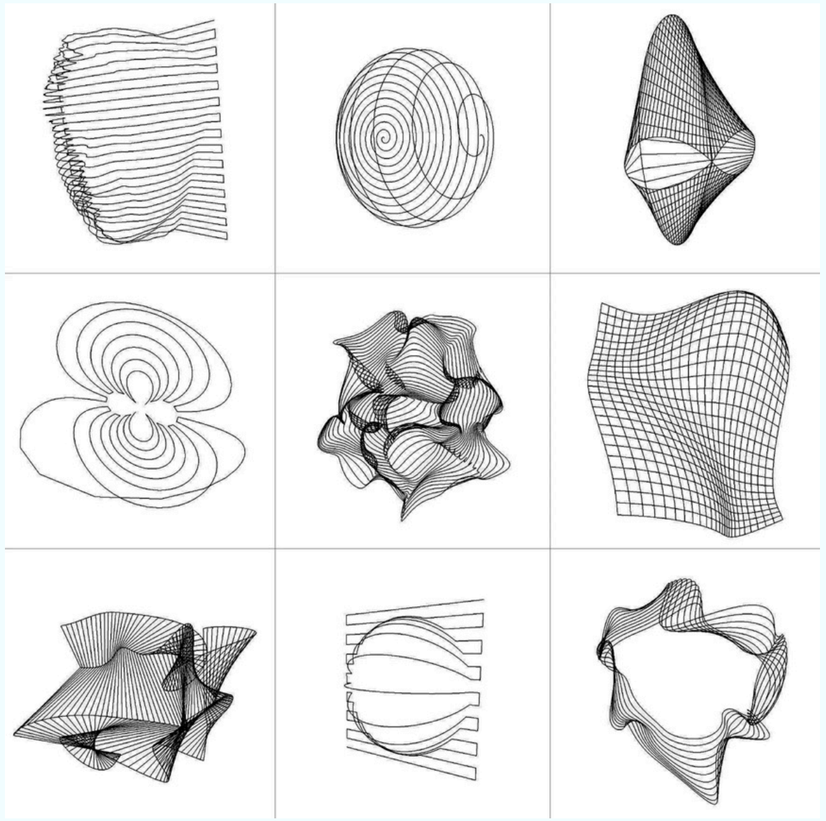


Modèle 3D :
Cet immeuble
(nombre de fenètres, fréquence cache ombre...)

Exemples génération procédurale (2/3)

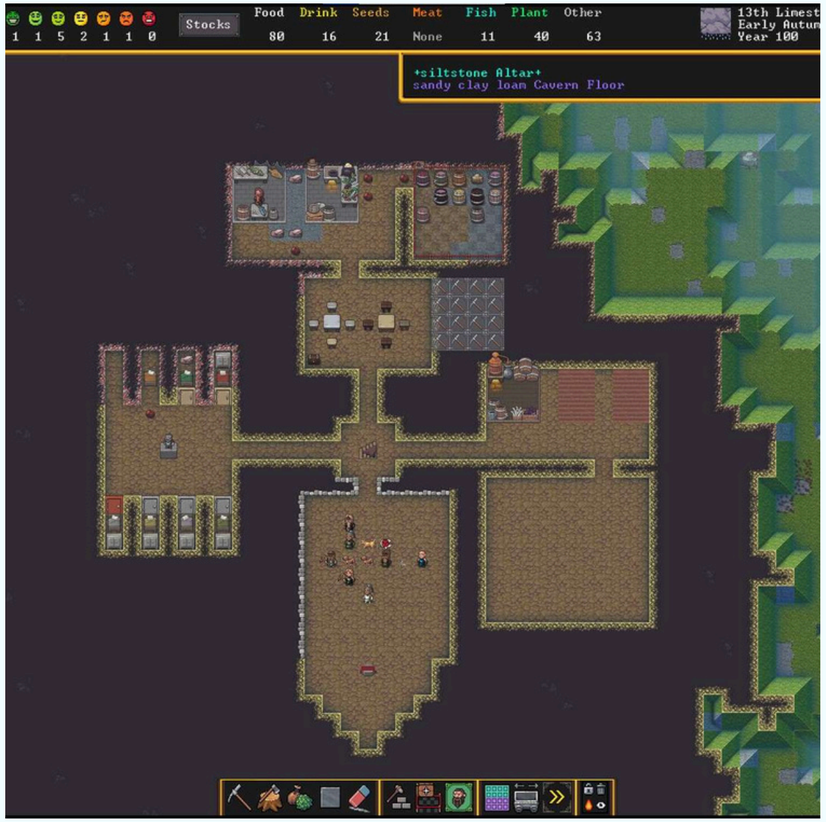


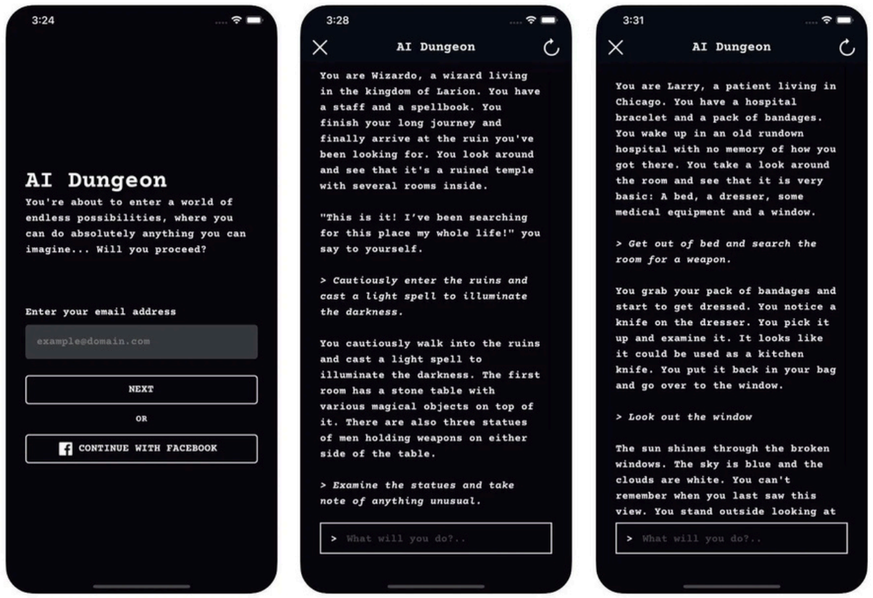
Animation :
Rain World



Dessins / Textures :
Lignes topologiques sur modèle 3D

Exemples génération procédurale (3/3)



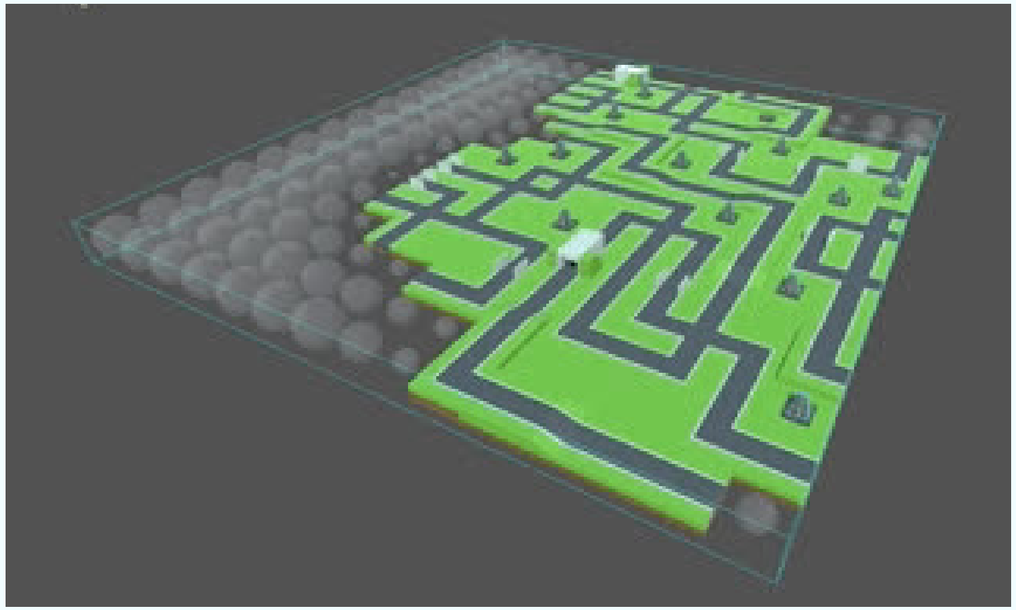


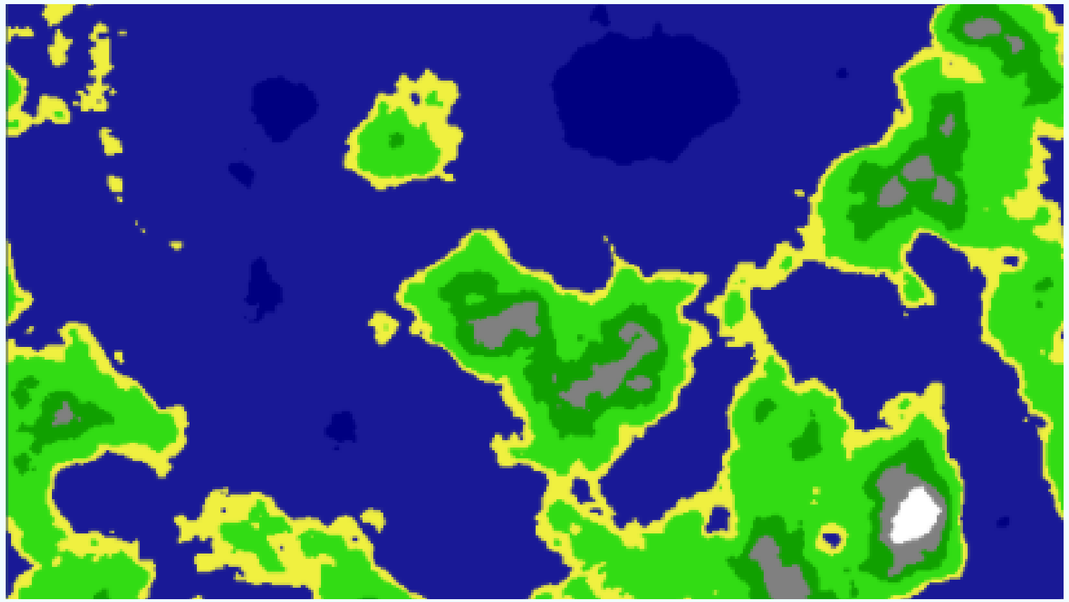
Histoires procédurales

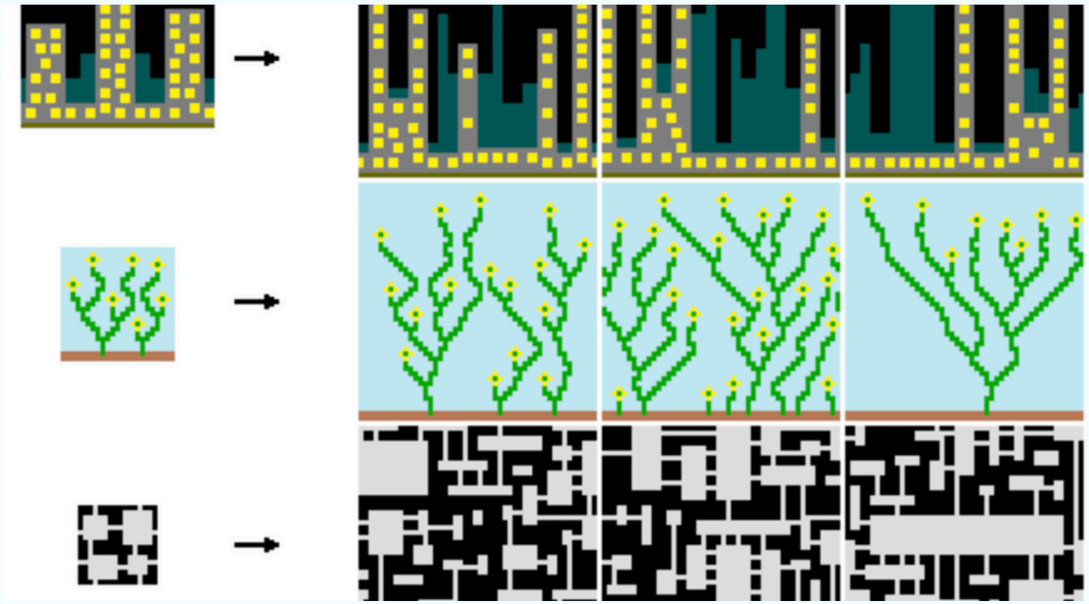
Concepts de Base (1/2)

<u>Contrôle et Paramétrage :</u>
Ajustements paramétriques pour varier les résultats

Concepts de Base (2/2)





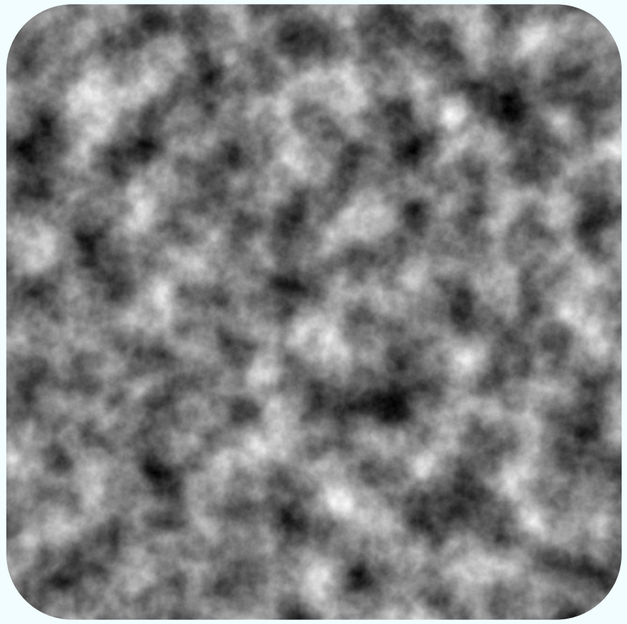


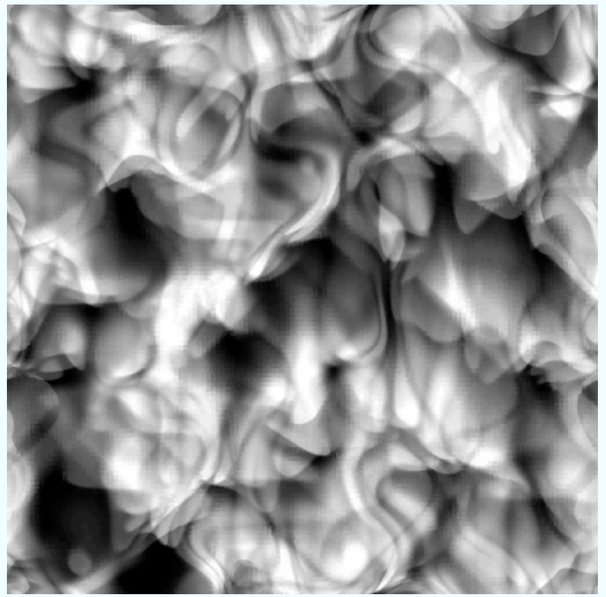
Règles et Contraintes :

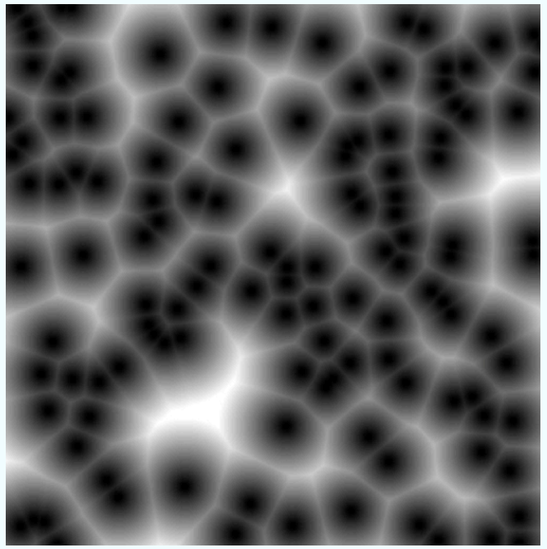
- Océan profond proche de Océan
- Plage proche de plaine
- Pas de montagne puis océan profond

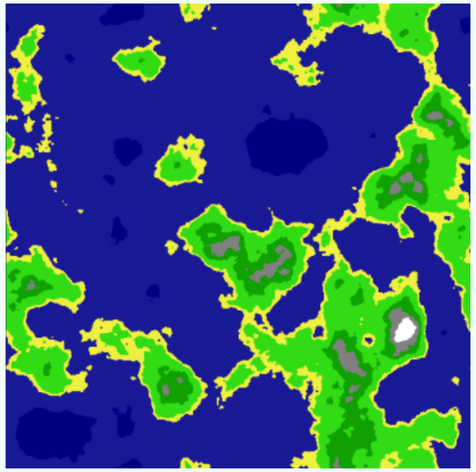
- Routes doivent se relier entre elles
- Tiles de sol peuvent pas donner tiles de fleur

Présentation d’algorithmes : Noised based





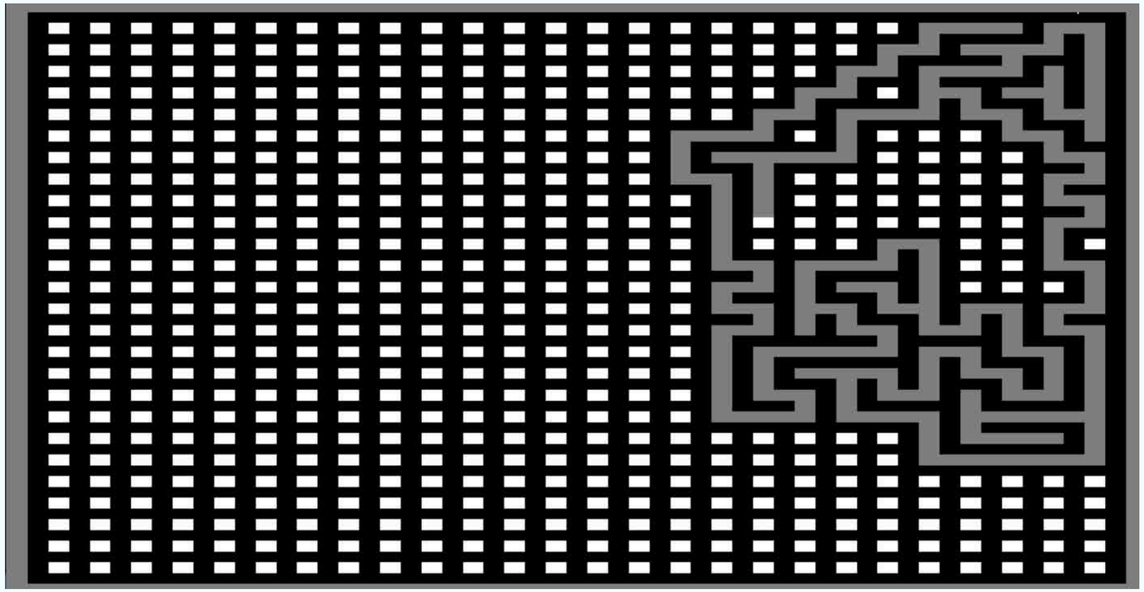


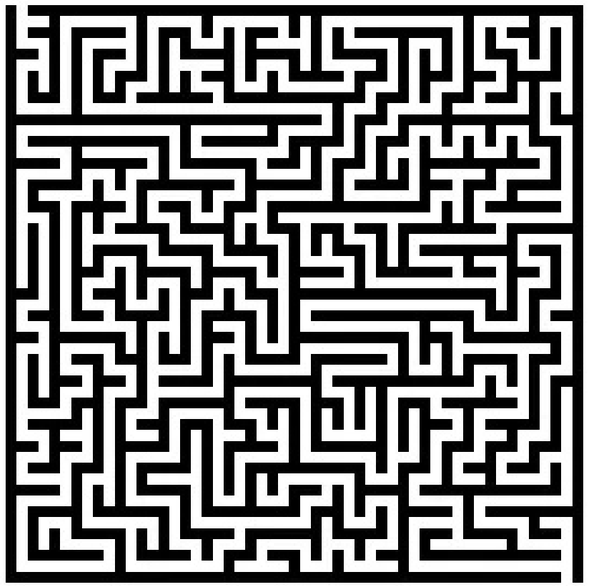


Partir d’une texture (généré ou non procéduralement) et créer des chose à partir de cette texture

- Facile à faire
- Assez général, en fonction des textures
- Manque de contrôle
- Parfait pour des trucs d’ensembles (Topographique naturels etc…)

Présentation d’algorithmes : Maze generation



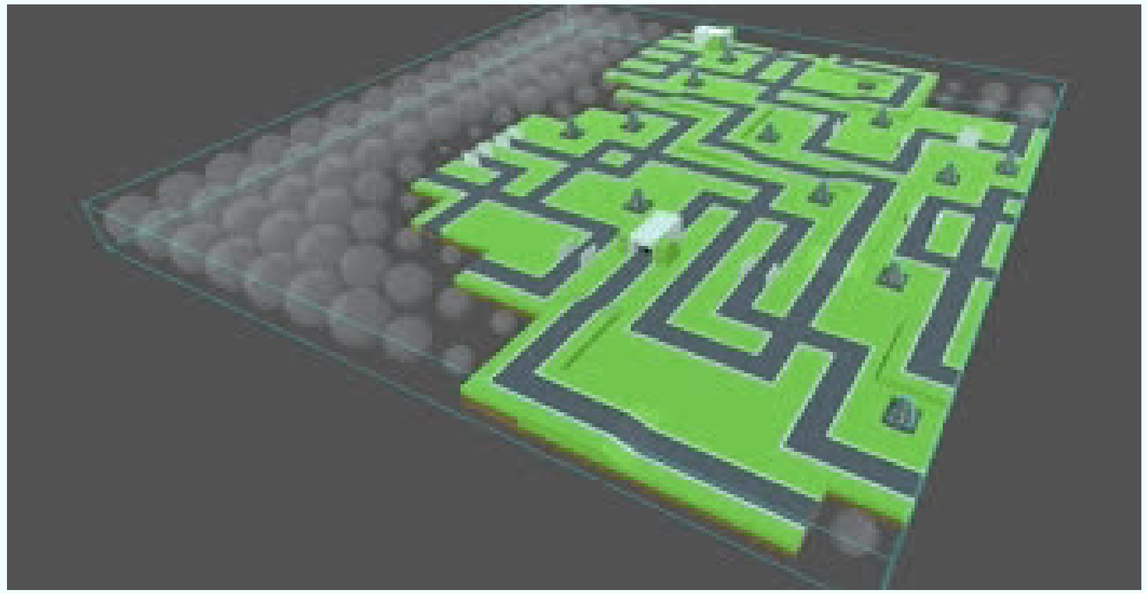


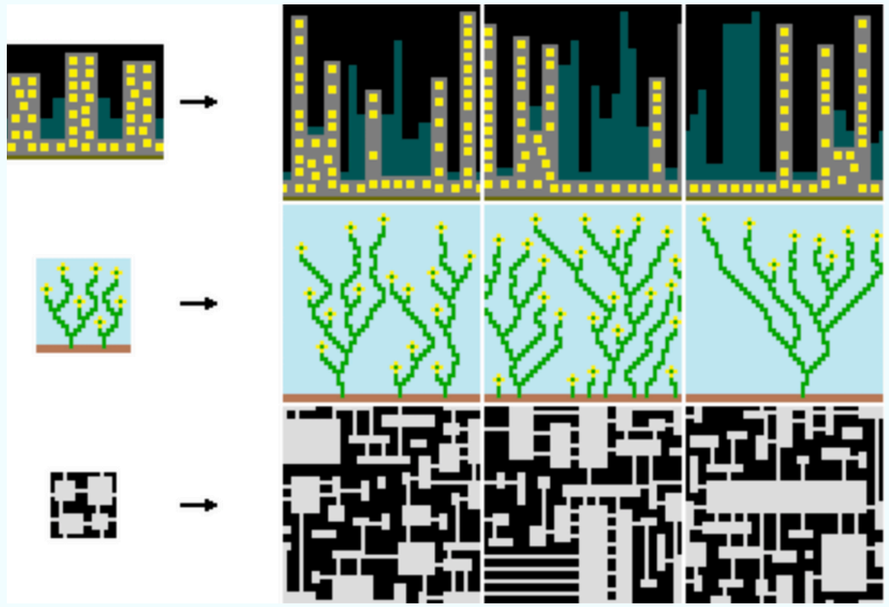
Partir d’une carte avec des cellules séparées par des murs. Partir d’un point de départ

A chaque étape:
1) choisir aléatoirement une cellule non exploré voisine de la tête
2) Creuser un trou entre la tête et la cellule choisie
3) Si pas de cellules non-explorés voisines de la tête, revenir en arrière jusqu’a trouver une cellule avec des cellules voisines non explorées, et repartir de là
(Besoin d’une pile de ‘undo’ pour le 3 (backtracking))

- Difficulté moyenne à faire
- Assez spécifique
- Gros manque de contrôle
- Parfait pour faire un labyrinthe ou une map de donjon d’ensemble (liens entre les salles)

Présentation d’algorithmes : Wave function collapse





Principe :

- Partir de règles d’adjacences (Route verticale peut donner sur route verticale ou virage)
- Partir d’une carte vide
- Choisir la case avec le moins d’entropie (le moins d’état possible)
- Collapse : Choisir aléatoirement une des tiles possible sur la case avec le moins d’entropie
- Répeter

- Difficile à faire
- Contrôle théoriquement infinie (Règles d’adjacences peuvent être 2x2 ou 3x3 ou 10x10)
- Très long à écrire les règles d’adjacences
- Difficile en pratique à converger et à bien contrôler
- Parfaite pour des maps avec des règles d’adjacences complexes (ville, routes, terrain naturel détaillé avec rivières…)

Conclusion et Perspectives

Hésitez pas à combiner toutes ces techniques, et faire vos propres algos à la mano, il faut feel la génération procédurale, il faut être un avec le monde qu'on veut créer, il faut ouvrir ses chakras

L'IA donne pas mal d'opportunités pour créer des infinités de dialogues, de comportements pour des npc etc...
ça peut être interessant aussi

https://github.com/Crizomb/ProceduralGenerationGodot

Maze generation et perlin noise based map ici + wave function coollapse buggé