

Distribution géométrique de régions de biomes

De la géométrie vers la topologie

Introduction

Au travers ce rapport je vais présenter la méthode utilisée pour placer des instances de biomes dans la géométrie afin de satisfaire des contraintes d'adjacence données par l'utilisateur. Dans cette méthode, nous réalisons un découpage de l'espace qui est ensuite labellisé pour assigner une instance de biome par région générée. Par ce découpage de l'espace, on peut récupérer le graphe topologique associé.

Cette méthode est utilisée au 09/08/2022.

Table des matières

Introduction.....	1
Table des matières.....	1
Entrées utilisateur.....	1
Sortie	2
Méthode	2
Améliorations possibles	6

Entrées utilisateur

L'utilisateur fournit au programme 2 fichiers d'entrée :

- Un modèle de biome qui décrit le type d'instances qu'il souhaite générer ainsi que la quantité d'instances. Chacun de ces types de biomes peuvent eux-mêmes décrire les types d'instances et la quantité désirée. D'autres propriétés peuvent être intégrées (surface, forme, profondeur, ...), mais dans cette méthode naïve, ce n'est pas pris en compte. Par exemple, le biome « Mayotte » peut se définir par 1x « Île », 5x « Plage », 4x « Lagon » (qui contiennent 2x « Patate de corail » et 1x « Rocher »), 8x « Profondeurs », etc.
- Une liste de règles d'adjacence, dans laquelle on précise, pour chaque type de biome, les biomes qui peuvent être accolés. Dans l'exemple précédent, on peut dire que « Ile » peut être adjacent à « Plage », que « Plage » peut être adjacent à « Ile » et « Lagon », que « Profondeurs » ne peut qu'être adjacent à « Lagon », etc.

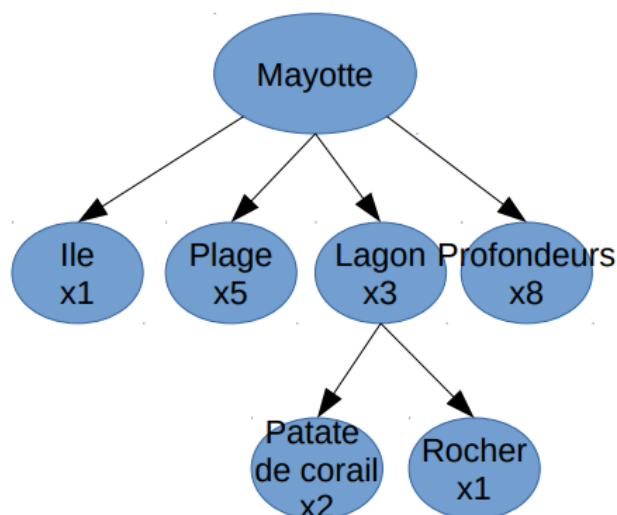


Figure 1. Modèle de biome pour la génération de "Mayotte"

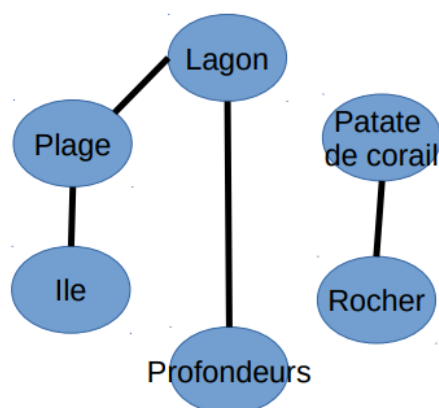
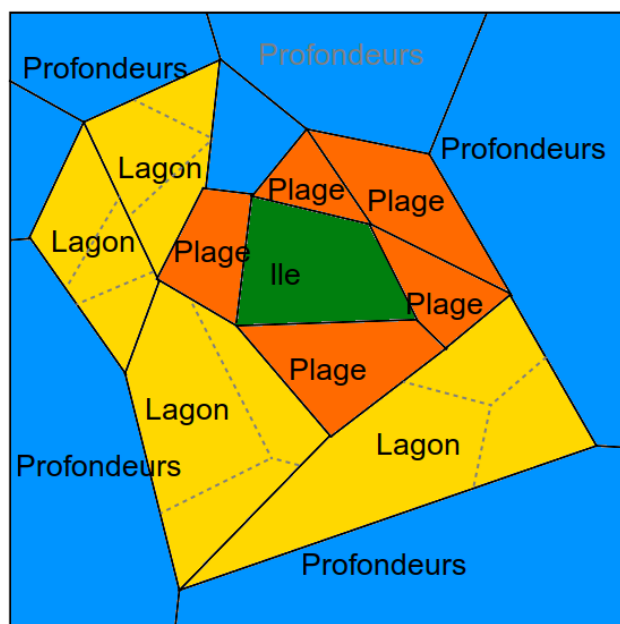


Figure 2. Contraintes d'adjacences

Sortie

En sortie de l'algorithme nous retrouvons une liste de régions géométriques labellisées par les instances qui leurs sont associées. Ces régions remplissent toute la surface demandée et minimise le nombre d'infractions aux règles d'adjacences fournies par l'utilisateur.



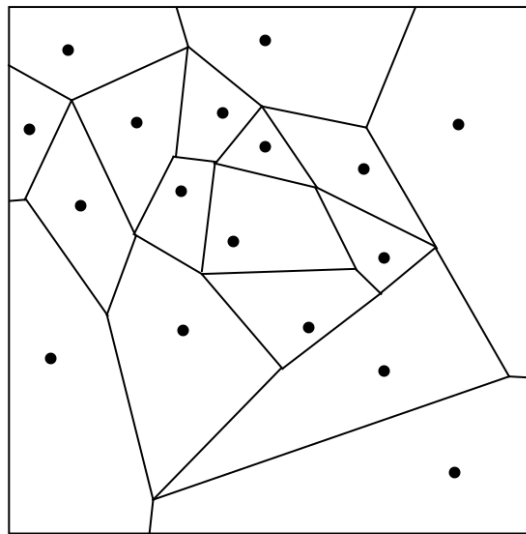
Méthode

La distribution de l'espace se réalise de manière hiérarchique en partant du haut du modèle vers les sous-biomes. La distribution de l'espace des sous-biomes prend tout l'espace du biome parent. Les espaces de biomes d'un même niveau hiérarchique ne se superposent pas.

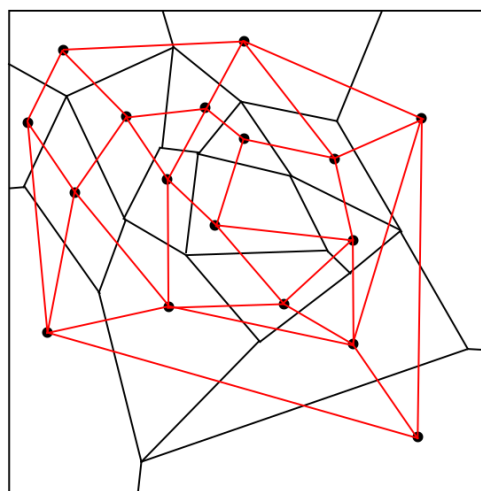
On va considérer que les biomes au premier niveau de la hiérarchie doivent remplir un espace carré de côté X, afin qu'on puisse facilement le rasteriser en une grille uniforme.

Tout d'abord, on calcule le nombre de régions à générer au premier niveau de la hiérarchie. C'est simplement le cumul des quantités d'instances par type. Dans l'exemple donné, on a $1 \times \text{« Île »} + 5 \times \text{« Plage »} + 4 \times \text{« Lagon »} + 8 \times \text{« Profondeurs »} = 18$ régions.

On distribue uniformément 18 germes (points) dans la région du biome parent, ici le carré de côté X. Un diagramme de Voronoï permet de découper l'espace en N cellules telles que tous les points d'une cellule sont plus proches que le germe associé qu'aucune autre germe en distance euclidienne. Les cellules résultantes sont « clippées » (découpées) afin d'être contenues dans la région du biome parent.



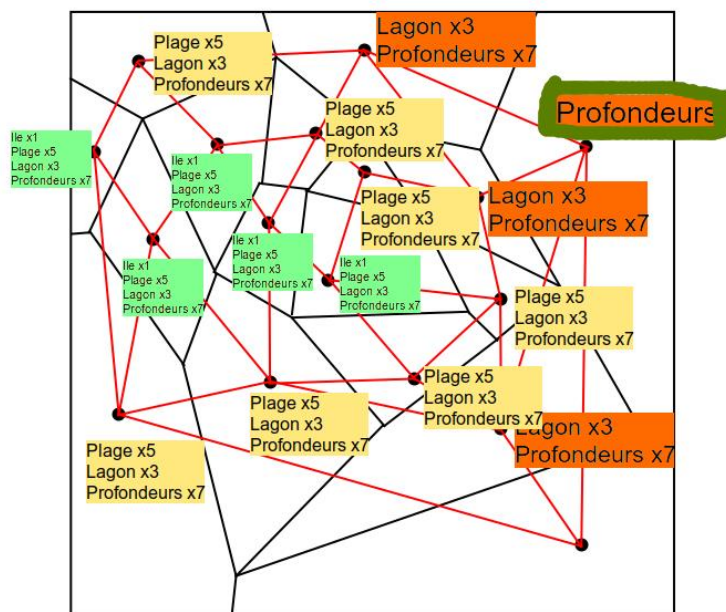
On récupère aisément la Triangulation de Delaunay qui est un graphe dont les nœuds sont les germes générés et dont les arêtes représentent une adjacence géométrique.



On va ensuite labelliser les nœuds dans le but d'assigner une instance de biome par cellule tout en satisfaisant les contraintes d'adjacences imposées par l'utilisateur. Pour cela on applique une variante de l'algorithme de Wave Function Collapse, la « Wave Function Collapse sans remise ».

Dans cet algorithme, on considère que chaque nœud peut être labellisé par l'une des 18 instances possibles (superposition des 18 états possibles). On choisit un nœud aléatoire et on lui fixe (« collapse ») un état aléatoire parmi les 18 possibles. Les nœuds adjacents mettent à jour leurs états superposés, sachant que le biome défini sur un nœud ne peut pas être adjacent à certains autres biomes. Si cela change les états possibles d'un nœud, alors l'information de changement est partagée aux nœuds voisins, qui vérifient à leur tour s'il faut mettre à jour leur liste de biomes possibles, et ainsi de suite.

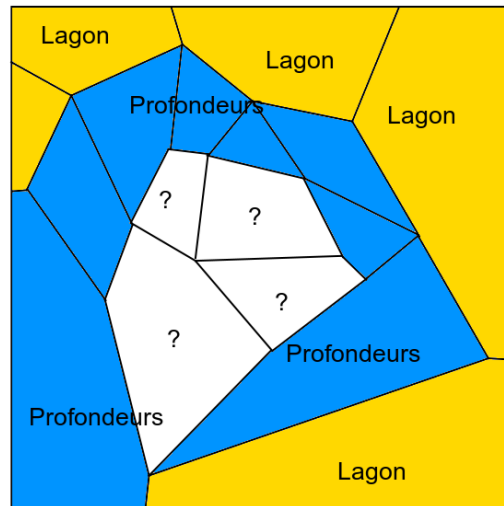
Par exemple, si le nœud choisi décide de devenir un biome « Profondeurs », alors d'après la liste de contraintes d'adjacences, les voisins doivent impérativement être des « Lagon » ou « Profondeurs », et donc peuvent retirer les biomes « Île » et « Plage » de leurs états possibles. Parce que l'on sait qu'ils ne peuvent avoir que ces états-ci, leurs voisins ne peuvent pas être des « Île » et donc peuvent retirer ce biome de leurs états possibles.



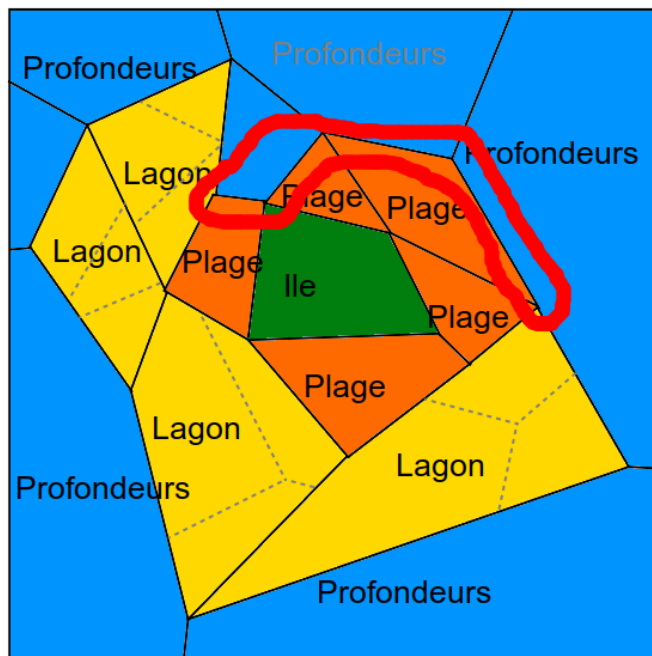
La variante proposée est dite « sans remise » car si une instance est placée à un endroit, on refuse qu'elle soit dupliquée. Tous les nœuds retirent de leurs états possible cet état.

Pour maximiser les chances à l'algorithme de collapser entièrement sans échec, on répète l'opération en choisissant le nœud dont le nombre d'états possible est le plus faible (entropie la plus basse).

L'algorithme termine quand tous les nœuds se voient assigner une instance de biome (collapse sans échec) ou lorsqu'un nœud ne peut pas recevoir un biome sans enfreindre une règle d'adjacence. Dans ce cas le plus rapide est de recommencer l'algorithme avec un graphe remis à zéro.



Parce que nous contraignons l'algorithme sans remise, il est possible qu'aucune configuration ne puisse convenir. De ce fait nous adaptons l'algorithme pour optimiser la répartition : nous répétons l'algorithme un nombre conséquent de fois (je propose N^2 avec N le nombre de nœuds), en enregistrant la configuration finale ayant correctement labellisé le plus de nœuds. Ensuite nous nous concentrons sur les nœuds non-labellisés. On choisit un nœud invalide au hasard, puis on vérifie les états possibles si on ignore l'un de ses voisins, puis deux de ses voisins, etc. jusqu'à avoir au moins un état possible. Dans ce cas on lui affecte un biome et on répète l'opération pour un autre nœud non-labellisé.



Enfin, chaque nœud est labellisé par une unique instance de biome. Sachant que chaque nœud de la Triangulation de Delaunay est associé à un unique germe du Diagramme de Voronoï, et donc à une unique cellule, on peut affecter une région géométrique à chaque instance de biome à générer.

Le processus complet est répété pour chaque biome ayant au moins un biome enfant dans la hiérarchie en distribuant les germes dans la région géométrique tout juste calculée pour ce biome.

Améliorations possibles

Ici seule l'adjacence est prise en compte dans le calcul de l'entropie, mais il serait possible d'intégrer des propriétés de l'environnement comme une heatmap de températures en entrée utilisateur ou encore une grossière heightmap. Ainsi les nœuds, ayant une information géométrique, peuvent connaître la température ou la profondeur « cible ». De ce fait, le biome « Île » aura plus de chance d'être affecté au nœud ayant la plus faible profondeur, ou encore les biomes « Profondeurs » seraient majoritairement affectés aux nœuds avec la plus faible température cible.

Et même sans l'intervention de l'utilisateur on pourrait affiner des cartes de températures, d'éclairage, de courants, etc. Par exemple, sachant que les îles sont au-dessus du niveau de la mer et que les profondeurs plongent à quelques centaines de mètres, on peut grossièrement estimer une heightmap à partir de la distribution des biomes précédemment, ce qui peut nous aider à placer les biomes du prochain niveau de la hiérarchie.

