Partie 2 génération de map

Le multi threading

Pour modifier le programme initiale et mettre en place le multi threading j'ai commencé par importé une bibliothèque

```
from multiprocessing import Pool, cpu_count
```

celle-ci va pouvoir me permettre de lancer le programme avec plusieurs cœur de mon cpu simultanément et en théorie accélérer la réalisation de ma map

1 La génération de la carte

Le comparatif

Pour pouvoir voir l'efficacité de mon multi threading j'ai commencé par lancé sans multi thread avec en paramètre ceci :

```
MAP_SIZE = (1024, 1024) # taille de la carte
SCALE = 256 # échelle de la carte
EXPO_HEIGHT = 2 # exposant pour la normalisation de la hauteur
num_threads = cpu_count() - 1  # Nombre de threads pour le multithreading
```

J'obtiens donc un temps de génération de 159.9s

```
Génération de la carte...
Carte de hauteur générée avec la seed : 903
Slopemap generated
Temps de génération: 159.99055910110474 secondes pour une taille de 1024x1024
Carte générée avec succès.
```

Ensuite je lance la version multi threader et on peut voir que la map se génère beaucoup plus rapidement

```
Génération de la carte avec multithreading...
Carte de hauteur générée avec la seed : 147
Carte de pente générée.
Temps de génération avec multithreading: 34.91183114051819 secondes pour une taille de 1024x1024
Carte générée avec succès.
```

Pour voir l'efficacité du multithread je double encore la taille de la carte pour voir

```
MAP_SIZE = (2048, 2048) # taille de la carte
SCALE = 256 # échelle de la carte
EXPO_HEIGHT = 2 # exposant pour la normalisation de la hauteur
num_threads = cpu_count() - 1  # Nombre de threads pour le multithreading
```

Avec la version classique j'obtiens une map en 626s du coup approximativement 10min

```
Génération de la carte...
Carte de hauteur générée avec la seed : 889
Slopemap generated
Temps de génération: 626.994179725647 secondes pour une taille de 2048x2048
Carte générée avec succès.
```

et avec la version multi threader : 148s soit 2.5min soit une amélioration des performence de 77.41%

```
Génération de la carte avec multithreading...
Carte de hauteur générée avec la seed : 66
Carte de pente générée.
Temps de génération avec multithreading: 148.31918478012085 secondes pour une taille de 2048x2048
Carte générée avec succès.
```

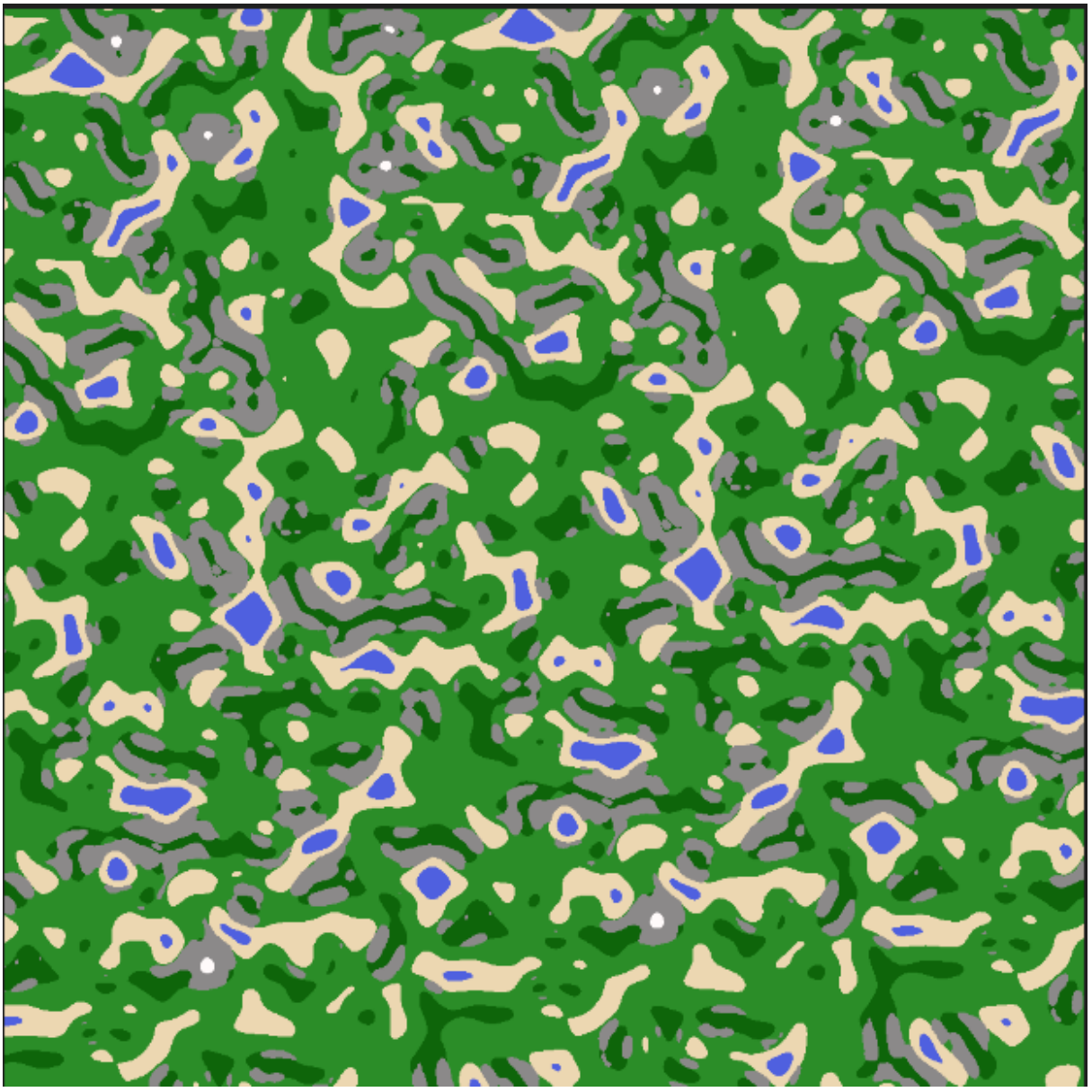
2 L'explication

Pour ce faire j'ai du adapter les les fonctions de génération

j'ai du rajouté des fonctions de calcule de block afin de séparé ma matrice en plusieurs chuncks, j'appelle ensuite dans mes fonctions les différents calcule dont j'ai besoin dans chacun des chuncks et ensuite je récupère le résultats de chaque chunks pour récupéré ma matrice entière.

2 le calcule du chemin le plus court

Je vais prendre en exemple une map généré de (1024 * 1024)

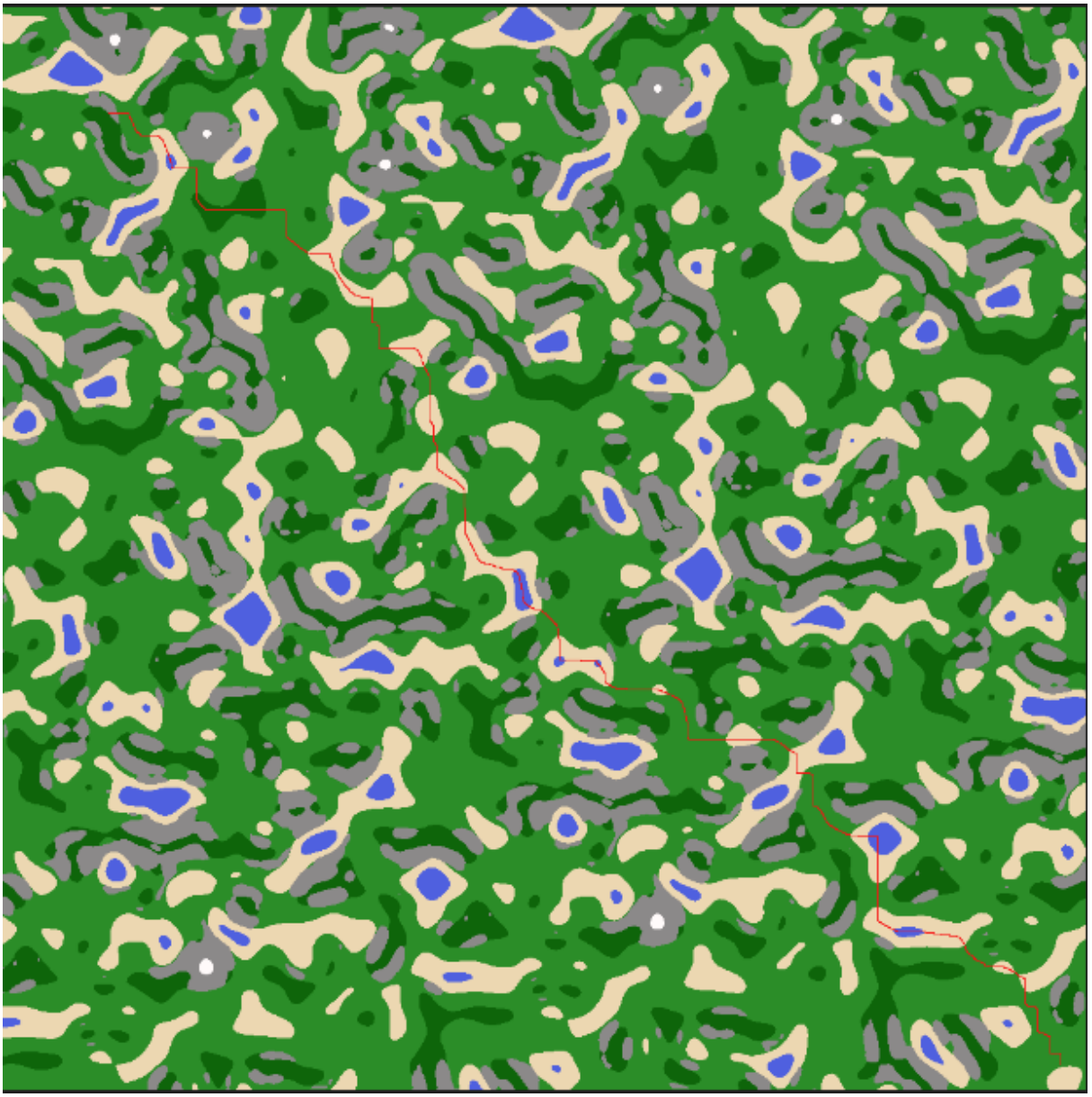


Maintenant le résultat du chemin le plus cours classique avec ces coordonnées la

```
Entrez la coordonnée x du point de départ: 100
Entrez la coordonnée y du point de départ: 100
Entrez la coordonnée x du point d'arrivée: 1000
Entrez la coordonnée y du point d'arrivée: 1000
```

j'obtiens ce resultat la

```
Nombre de points dans le chemin: 1801
Temps de calcul: 0.9048352241516113 secondes pour une taille de 1024x1024
```

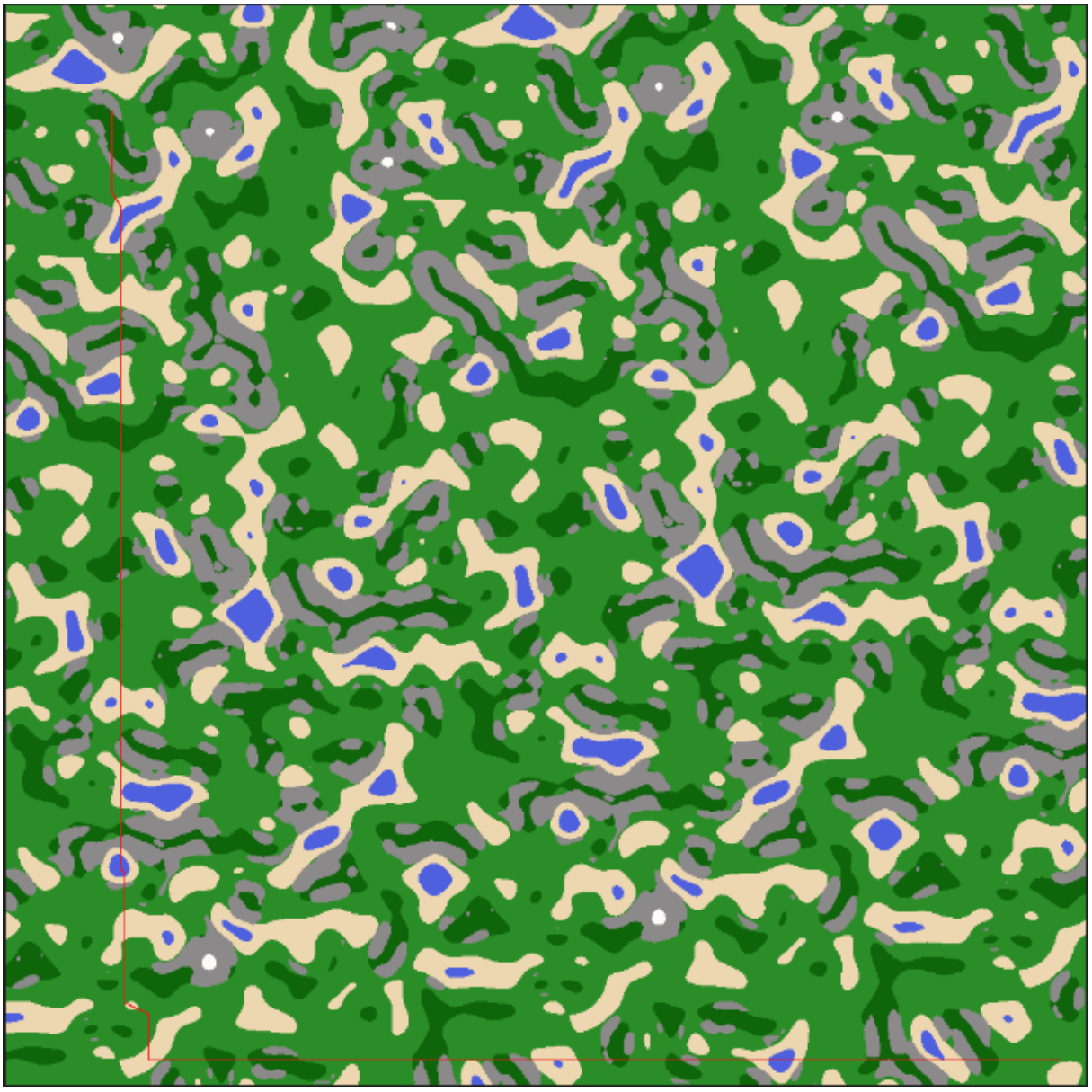


On obtiens donc ce resultat la

Maintenant la version avec le multiprocessing

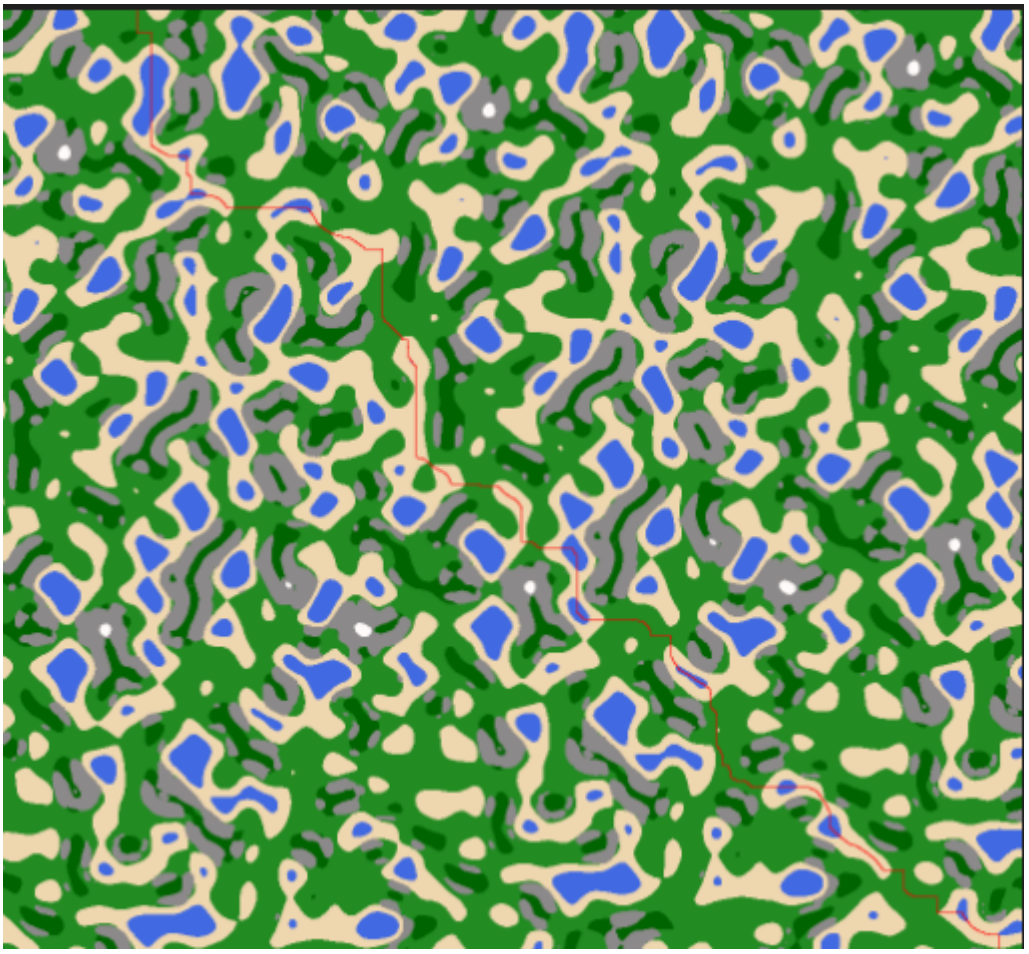
```
Nombre de points dans le chemin: 1801
Temps de calcul: 9.247450351715088 secondes pour une taille de 1024x1024
```

on obtient un calcule plus long car ici il y a très peu de point a calculer comme je cherche le chemin et uniquement 1 seul chemin le plus cours par conséquence le multi-processing ralenti le processus



En plus on peut voir que l’algo a tendance a avancé en ligne droite

Correction apres quelque modification sur la gestion de la synchronisation



j’arrive bien a obtenir le chemin le plus cours

Mise a jour du programme et utilisation

```
+-----------------------------------+
|               Menu                |
|-----------------------------------|
| 1. Générer la carte               |
| 2. Exporter la carte              |
| 3. Exporter la carte en CSV       |
| 4. Calculer et exporter le chemin |
| 5. Supprimer un fichier           |
| 6. Supprimer tous les fichiers    |
| 7. Générer la carte multithreading|
| 8. Générer tous les chemins       |
| 10. Quitter                       |
+-----------------------------------+
Choisissez une option:
```

Avec le multi-processing j’ai rajouté de nouvelles possibilités au menu avec

- 7 on peut utilisé le multi-processing pour la génération de la map qui apres plusieurs test est entre 60-70% plus rapide
- 4 est la version multi-processing du chemin le plus cours
- 8 est la version classique

il y a eu aussi une modification de l’exportation de la map avec le chemin le plus cours qui pouvait prendre du temps dans grande map exemple (2048,2048) en utilisant plusieurs thread de mon ordinateur