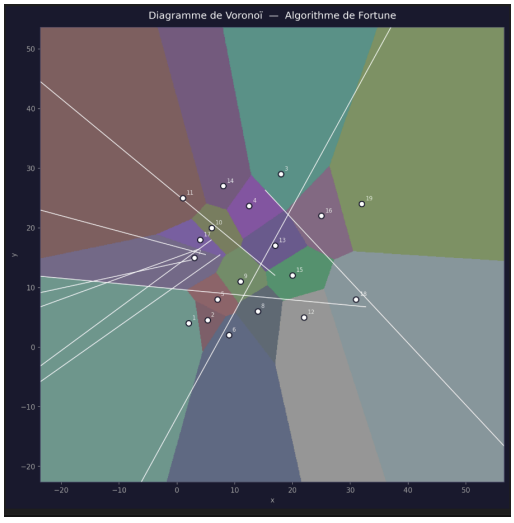
Phase 2 : Développement assisté par l'IA (Claude AI)

1. Prompts utilisés :

> Salut l'ami, je veux que tu me génères un programme en Python pour générer avec ton propre algorithme un schema de voronoi. Tu auras un fichier texte avec pour cahque ligne la coordonnée x,y de chaque points. Tu devras utiliser python pour faire ça, tu peux utiliser matplotlib pour l'affichage, mais pour voronoi, tu devras le faire toi même.

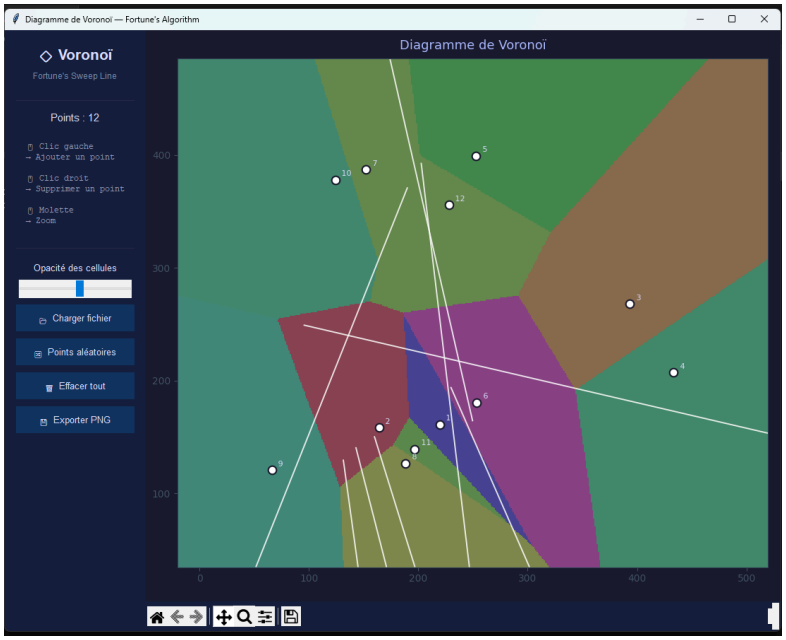
Résultat :
Un fichier voronoi.py qui prends en paramètre un fichier texte de points et génère une image png du résultat en couleurs. C'était complètement fonctionnel mais sans interface graphique.



Ensuite je lui ai demandé d'ajouter une interface graphique.

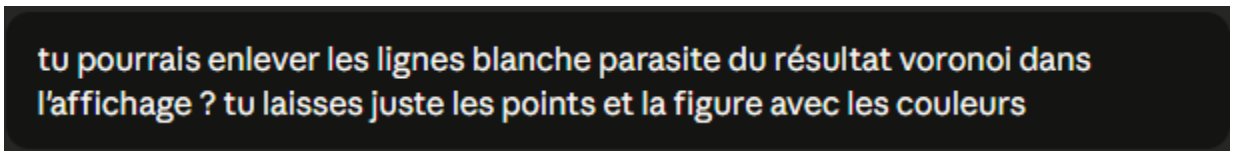
> Utilise une interface graphique comme matplotlib ou bien juste une interface pour afficher l'image, je veux qu'on puisse le voir visuellement dans le programme.

Résultat:

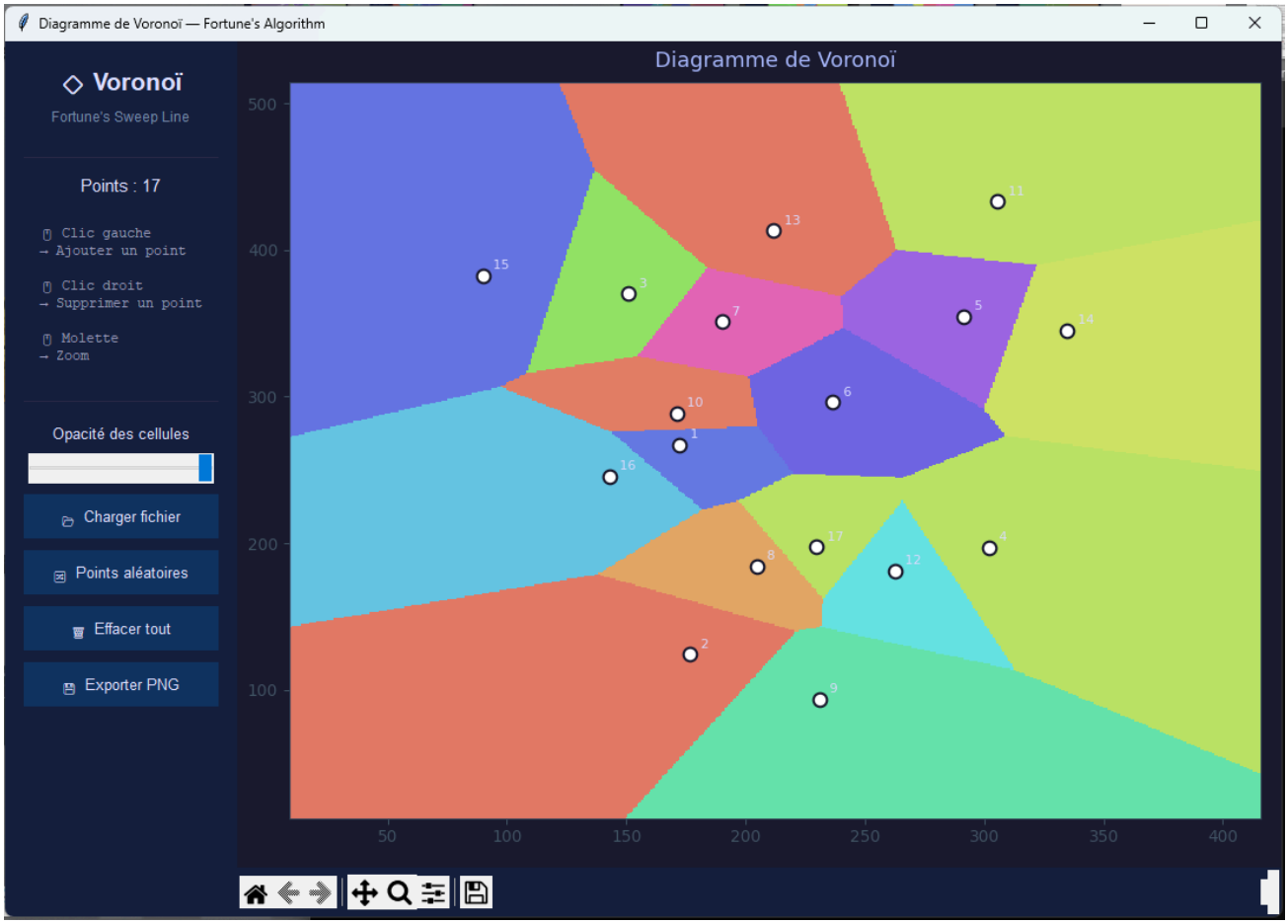


Claude AI ne s'est pas contenté de générer uniquement le graphique, il a généré une interface complète et fonctionnelle avec plusieurs options : Ajouter/Supprimer des points dynamiquement avec la sours, importer un fichier de points, effacer le graphique, générer un graphique aléatoire, exporter en PNG, zoomer/dezoomer.

Le petit soucis, c'est les lignes blanches qui sont pas belle à voir, je lui ai donc demander de les supprimer.



Résultat :



Ensuite je lui ai demandé de générer les tests unitaires avec pytest en python.

Ok très bien ça fonctionne bien, tu pourrais me faire des tests unitaires avec pytest ?

Résultat :
Une liste de 61 fonctions qui test l'alogirthme.
Au premier lancement des test avec pytest, 60 fonctions sur 61 ont reussis, une a échoué.

Je lui ai demande de corrigé

```
============================================================
= test session starts
============================================================
==
platform win32 -- Python 3.14.3, pytest-9.0.2, pluggy-1.6.0
rootdir: C:\Users\Anas Hadri\Desktop\voronoi_claude
plugins: anyio-4.12.1
collected 61 items
test_voronoi.py .....................................F.............
[100%]
============================================================
======= FAILURES
============================================================
=======

_____________________________________________
TestCollectSegments.test_segments_non_degeneres
_____________________________________________
self = <test_voronoi.TestCollectSegments object at 0x000001F034E9C3B0>
  def test_segments_non_degeneres(self):
    """Aucun segment retourné par collect_segments ne doit avoir une
longueur nulle."""
    coords = [(10, 10), (60, 20), (30, 70), (80, 60), (50, 40)]
    diag, _ = voronoi(coords)
    segs = vg.collect_segments(diag, 0, 100, 0, 100)
    for (x1, y1), (x2, y2) in segs:
      length = math.hypot(x2 - x1, y2 - y1)
>       assert length > 1e-9, f"Segment dégénéré trouvé : ({x1:.4f},
{y1:.4f})→({x2:.4f},{y2:.4f})"
E       AssertionError: Segment dégénéré trouvé :
(18.3333,40.5556)→(18.3333,40.5556)
E       assert 0.0 > 1e-09
test_voronoi.py:496: AssertionError
============================================================
short test summary info
============================================================
FAILED
test_voronoi.py::TestCollectSegments::test_segments_non_degeneres -
AssertionError: Segment dégénéré trouvé :
(18.3333,40.5556)→(18.3333,40.5556)
============================================================ 1
failed, 60 passed in 0.23s
============================================================
PS C:\Users\Anas Hadri\Desktop\voronoi_claude>
```

Tout les tests sont passé sauf 1

Show less

Ensuite il a corrigé et c'était fonctionnel.

Pour finir, je lui ai demandé de générer un fichier requirements.txt et un fichier readme.