

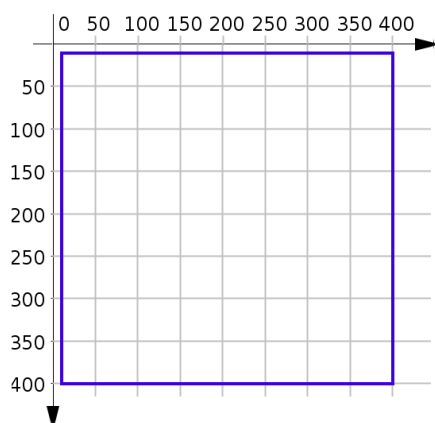
**Prérequis:** Il est conseillé d'avoir fait le TP Tkinter.

Un diagramme de Voronoï est un découpage du plan en parties, appelées cellules, à partir d'un ensemble de points appelés germes. Chaque cellule est associée à un germe. Elle est formée de l'ensemble des points du plan plus proches de ce germe que des autres.

Nous allons découper le plan en cellules. L'idée centrale est donc de regarder pour tous les points du plan sa distance à chaque germe: on pourra alors comparer ces distances et ainsi conclure. Nous démarrons avec trois germes placés en A(100;50), B(150;200) et C(200;150).

On pourra utiliser tkinter pour effectuer les tracés.

1. Pourquoi peut-on se contenter de ne comparer que les carrés des distances entre points et germes ?
2. Ecrire une fonction `carreA(x, y)` qui renvoie le carré de la distance en un point de coordonnées  $(x, y)$  et A.
3. Compléter votre programme en écrivant une fonction `colore(x, y)` qui colorie en bleu le point le point de coordonnées  $(x, y)$  s'il est plus près de A, en vert s'il est plus près de B.
4. Ecrire une fonction `parcours` qui parcourt tous les pixels de l'écran pour tracer le diagramme de Voronoï sur la zone carrée de diagonale (10;10) à (400;400).



Zone de dessin: on colorie tous les pixels

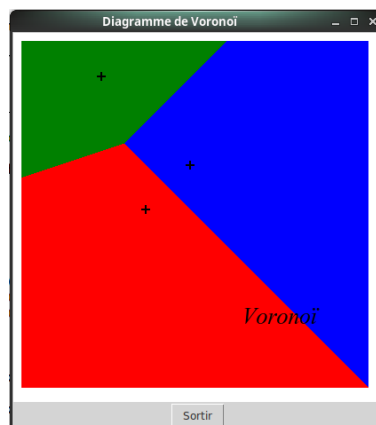


Diagramme de Voronoï pour A, B, C.

5. Ecrire une fonction `marque_point(x, y)` qui marque d'une croix les germes.
6. Ecrire une fonction `carre_dist(x, y, x_g, y_g)` qui calcule le carré de la distance entre  $M(x, y)$  et un autre point de coordonnées  $(x_g, y_g)$ . Modifier votre programme en conséquence.
7. Compléter le programme avec les germes de coordonnées: C(200;150), D(300;300), E(350;100), F(300;200).

## 0.1 Partie B

On veut maintenant placer les germes au hasard.

1. Ecrire une fonction `point_alea()` qui renvoie un dictionnaire de 5 points aux coordonnées aléatoires dans la zone de (0; 0) à (x\_max; y\_max).

```
>>>point_alea(5, 400, 400)
{'A': (295, 337), 'B': (90, 356), 'C': (225, 334), 'D': (114, 361), 'E': (163, 109)}
```

2. Adapter les fonctions pour utiliser ce dictionnaire.

## 0.2 Partie C: classe de terminale

Reprendre ce TP en utilisant la classe `Point()` du cours POO.