

Projet d'Architecture Matérielle et Assembleur

ESGI – 2A

Soutenances : Février 2025

Ce projet devra être réalisé en groupe de trois.

Mail de contact : **nneveu@gmail.com**

Vous allez réaliser en assembleur NASM X86-64 sur Linux l'implémentation de l'algorithme du diagramme de Voronoï.

Merci à Frédéric Baudoin pour sa suggestion d'algorithme et pour son aide.

Fichier fourni :

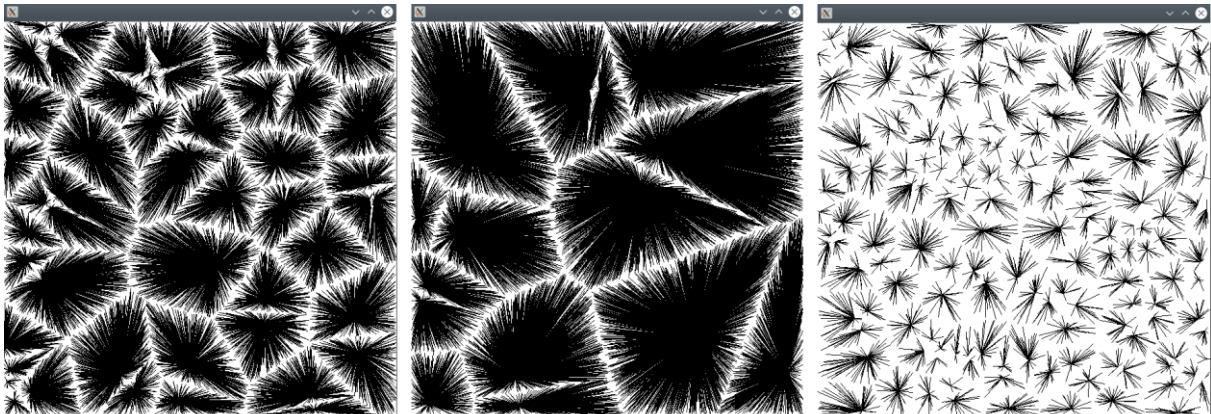
code_pour_dessiner.asm : contient le code permettant d'ouvrir une fenêtre et d'y dessiner. Il montre notamment comment tracer des lignes et comment changer la couleur du dessin.

1 – Réalisation :

Etape 1 :

on tire aléatoirement un certain nombre de foyers dont on stocke les coordonnées dans un ou deux tableaux.

On tire ensuite aléatoirement les coordonnées d'un certain nombre de points et pour chacun de ces points, on le relie au foyer dont il est le plus proche.



*Figure 1 : à gauche : 50 foyers, 10000 points, au milieu: 15 foyers, 10000 points,
à droite : 150 foyers, 2000 points*

Etape 2 :

Vous allez créer un tableau contenant des couleurs (format des couleurs : 0xRRVVBB où RR représente le niveau de rouge, VV le niveau de vert et BB le niveau de bleu, entre 00 et ff en hexadécimal) qui formeront un dégradé.

Pour créer des dégradés et obtenir les références des codes couleurs :

<https://photokit.com/colors/color-gradient/?lang=fr>

A partir de l'étape 1, vous allez maintenant colorer chaque foyer ainsi que les traits le reliant aux points qui y sont associés avec une couleur contenue dans le dégradé. Un foyer aura une couleur unique.

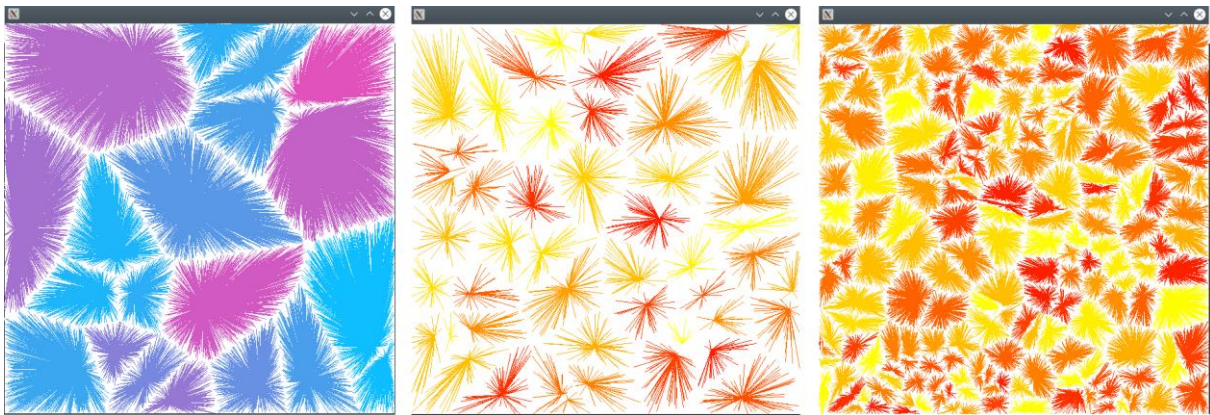


Figure 2 : à gauche : dégradé Bleu vers Rose, 20 foyers, 10000 points, au milieu : dégradé Jaune vers Rouge, 50 foyers, 1500 points, à droite : dégradé Jaune vers Rouge, 200 foyers, 15000 points

Etape 3 :

Sur le même modèle que l'étape 2, vous allez maintenant colorer chaque trait individuel d'un foyer avec une couleur issue du dégradé utilisé. Les traits reliés à un foyer n'auront donc plus forcément la même couleur, on verra apparaître le dégradé dans chaque foyer.



Figure 3 : à gauche : dégradé Bleu vers Rose, 30 foyers, 10000 points, au milieu : dégradé Jaune vers Rouge, 50 foyers, 5000 points, à droite : dégradé Vert vers Rouge, 100 foyers, 5000 points

Etape 4 :

Vous allez maintenant implémenter l'algorithme de Voronoï.

Chaque foyer va définir une zone colorée constituée des points pour lesquels le foyer est le plus proche. Vous devrez tirer un certain nombre de couleurs aléatoirement pour remplir les différentes zones.

Vous parcourez chaque point de la fenêtre, vous déterminez quel est le foyer le plus proche et colorez le point avec la couleur affectée à ce foyer.

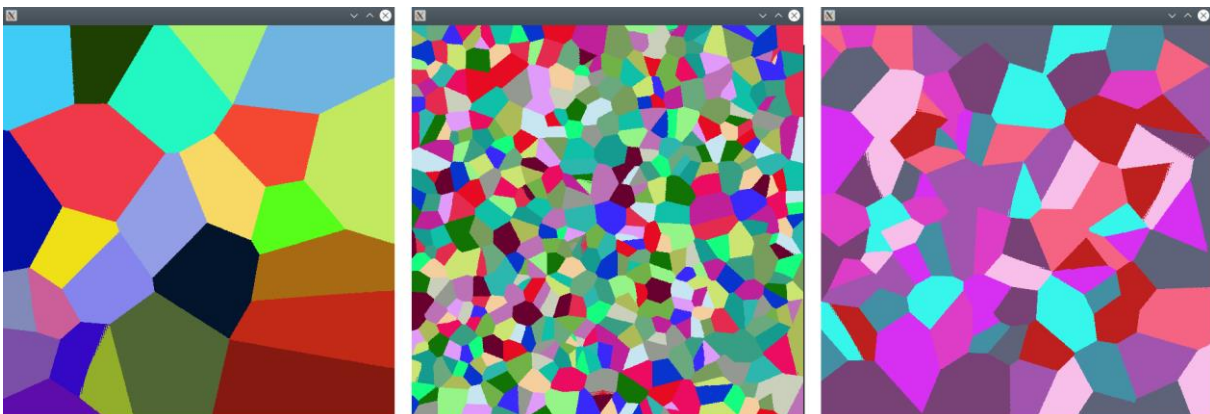


Figure 4 : à gauche : 25 foyers, 25 couleurs, au milieu : 500 foyers, 25 couleurs, à droite : 100 foyers, 10 couleurs.

2 – Attendus :

Vous devrez implémenter chacune des étapes individuellement.
Vous obtiendrez des points pour chacune des étapes.

Vous devrez pouvoir modifier rapidement le nombre de foyers (toutes les étapes), le nombre de points (étapes 1 à 3) ainsi que le nombre de couleurs (étape 4).

Votre programme devra comporter au minimum deux fonctions :

- Une de tirage aléatoire, qui prend en argument une valeur x et qui retournera une valeur aléatoire contenue dans $[0, x]$.
- Une qui calcule la distance entre deux points : elle prendra en arguments les coordonnées x et y des deux points et retournera la distance selon la formule :
$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$
 (on arrondira le résultat).

NB : Pour le tirage de nombres aléatoires, vous utiliserez l'instruction RDRAND de l'assembleur NASM 64 bits.

Cette instruction prend en argument un registre d'au moins 2 octets et retourne, dans ce registre, une valeur comprise entre 0 et la valeur maximale correspondant à la taille du registre passé en argument (0 à 65535 pour un registre de 16 bits par exemple).

La valeur générée par RDRAND n'est valide que si le flag CF du registre d'état est égal à 1 après l'appel à l'instruction. Il est donc nécessaire de la relancer si CF=0.

RAPPEL :

Il vaut mieux un programme qui est bancal mais personnel qu'un programme qui fonctionne parfaitement et que vous avez récupéré chez quelqu'un d'autre.

C'est plus satisfaisant moralement et cela énerve beaucoup moins le prof (et l'excuse du code qu'on a écrit il y a longtemps et dont on ne se souvient plus ne fonctionne pas, et n'a d'ailleurs jamais fonctionné, merci)