

Projet d'Architecture Matérielle et Assembleur

ESGI – 2A
Soutenance Février 2024

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 de la Marche de Jarvis.

Il permet de déterminer l'enveloppe convexe d'un nuage de points. Il est aussi appelé Wrapping Gift Algorithm car il s'agit d'envelopper les points comme on le ferait avec un cadeau.

Voilà son algorithme :

Soit E l'ensemble des points que l'on veut entourer (E est composé d'au moins 3 points).
Soient P et Q deux points.

On détermine le point L le plus à gauche de E, c'est-à-dire celui dont l'abscisse est la plus petite (C'est le premier point de notre enveloppe).

$P \leftarrow L$.

Répéter

On ajoute P à l'ensemble H des points de l'enveloppe.

On définit Q comme le point suivant P dans l'ensemble E.

Pour chaque point I de l'ensemble E

On regarde si le triangle PIQ tourne dans le sens des aiguilles d'une montre

Si c'est le cas, I est un meilleur candidat que Q pour l'enveloppe donc $Q \leftarrow I$

$P \leftarrow Q$

Tant que $P \neq L$

Dessiner l'enveloppe

Pour calculer l'orientation du triangle PIQ, on va utiliser le produit vectoriel entre les vecteurs \overrightarrow{PI} et \overrightarrow{PQ} . Si ce produit vectoriel est supérieur ou égal à 0, alors le triangle PIQ est orienté dans le sens des aiguilles d'une montre et le point I est un meilleur candidat que Q pour l'enveloppe.

On définit le produit vectoriel de deux vecteurs \overrightarrow{AB} et \overrightarrow{BC} par :

$$(x_{BC} \times y_{AB}) - (x_{AB} \times y_{BC})$$

Rappel :

$$x_{AB} = x_B - x_A$$

$$y_{AB} = y_B - y_A$$

Le nombre de points de l'ensemble E devra pouvoir être modifié (utiliser un %define pour définir une constante égale au nombre de points)

Ce projet se déroulera en trois étapes.

1-

Vous allez écrire un programme qui tire aléatoirement les coordonnées des points de l'ensemble E. Pour cela, 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.

Vous dessinerez ce nuage de points grâce à la fonction fournie.

Vous devrez écrire une fonction qui prend en argument la valeur maximale de la plage de valeurs aléatoires et qui retourne la valeur aléatoire générée.

Exemples :

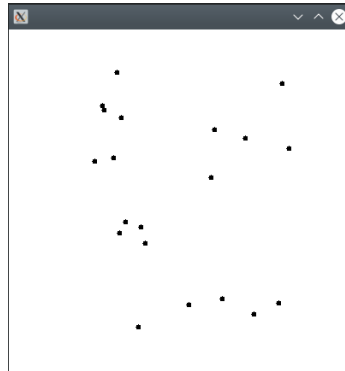


Figure 1 : Un nuage de 20 points

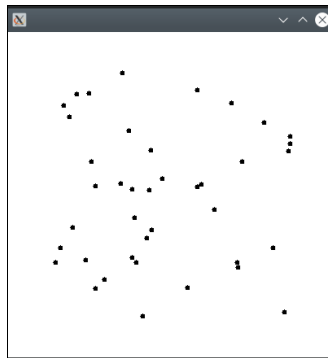


Figure 2 : Un nuage de 40 points

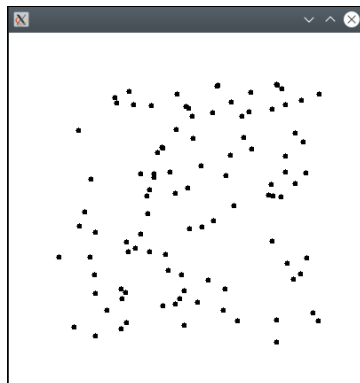


Figure 3 : Un nuage de 100 points

2-

Vous allez écrire le programme permettant de dessiner l'enveloppe correspondant au nuage de points généré précédemment. Vous suivrez l'algorithme de Jarvis donné plus haut.

Vous dessinerez cette enveloppe autour du nuage de points de l'étape précédente.

Vous devrez écrire une fonction qui permet de tester l'orientation du triplet de points P, I et Q dans l'algorithme.

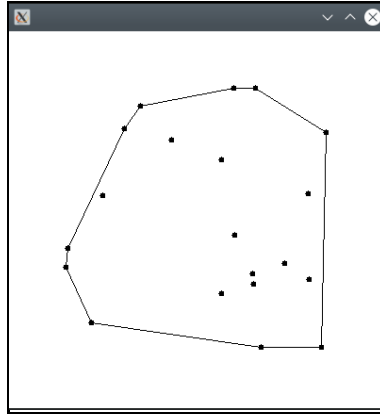


Figure 4 : L'enveloppe d'un nuage de 20 points

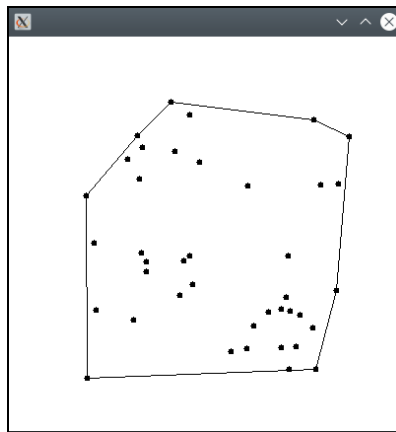


Figure 5 : L'enveloppe d'un nuage de 40 points

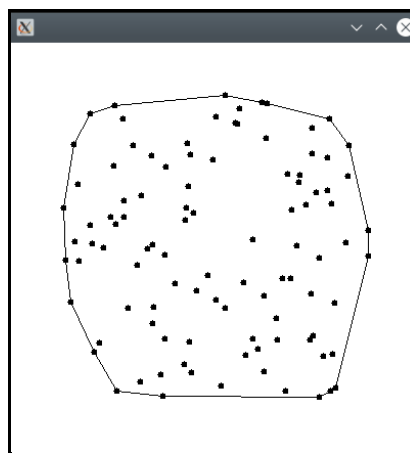


Figure 6 : L'enveloppe d'un nuage de 100 points

3-

Dans cette dernière étape, vous allez tirer aléatoirement les coordonnées d'un point et vous allez tester si ce point est contenu dans l'enveloppe ou pas. Vous afficherez le résultat. Pour tester l'appartenance d'un point, il faut regarder, grâce à la fonction d'orientation écrite précédemment, si le point est situé à gauche ou à droite de chaque arête de l'enveloppe. S'il est situé à droite pour **toutes** les arêtes, alors le point appartient à l'enveloppe, sinon, il est à l'extérieur.

Soient A et B les extrémités d'une arête et C le point tiré aléatoirement, alors si le produit vectoriel de \overrightarrow{AB} et \overrightarrow{BC} est négatif, le point C est situé à droite de l'arête AB.

Vous dessinerez en rouge le point à vérifier et vous écrirez dans la console s'il appartient ou non à l'enveloppe.

Exemples :

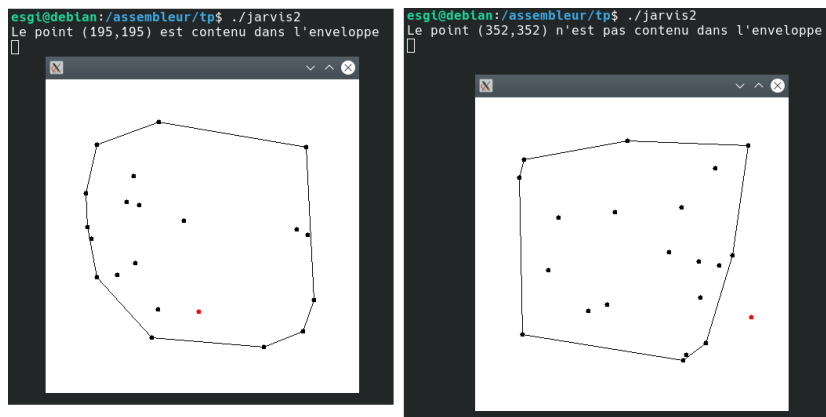


Figure 7 : Avec un nuage de 20 points

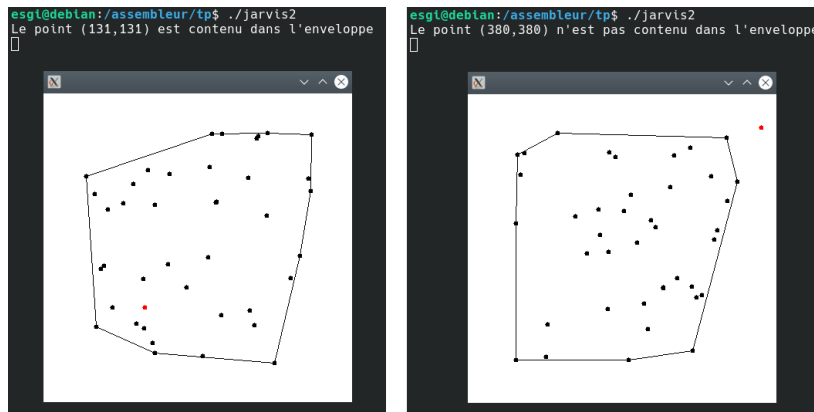


Figure 8 : Avec un nuage de 40 points

Lien utile :

<https://iq.opengenus.org/gift-wrap-jarvis-march-algorithm-convex-hull/>

Fichier fourni :

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