

Projet d'Architecture Matérielle et Assembleur

ESGI – 2i

Soutenance : 15 Janvier 2025

Ce projet devra être réalisé en groupe de trois.

Mail de contact : nneveau@gmail.com

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

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 cercles et comment changer la couleur du dessin.

1 – Réalisation :

Etape 1 :

Vous allez créer des cercles dont le rayon et les coordonnées du centre seront tirés aléatoirement. Ils ne devront pas se chevaucher, ni être tangents..

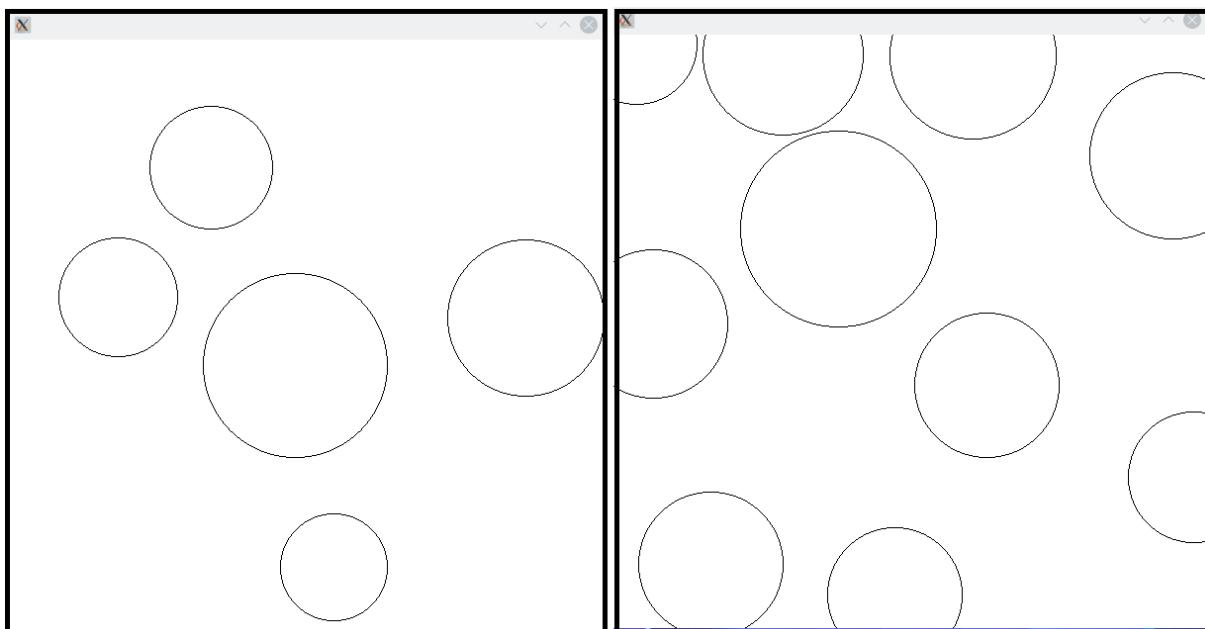


Figure 1 : à gauche : 5 cercles, à droite : 10 cercles

Etape 2 :

Les cercles initiaux tracés précédemment vous serviront de base pour la suite.
Vous allez maintenant tracer des cercles selon les règles suivantes :

- 1- Vous tirez aléatoirement les coordonnées en x et y du centre du cercle.
- 2- Si ces coordonnées placent ce centre dans un cercle existant, vous les abandonnez et retirez de nouvelles coordonnées.
- 3- Sinon on calcule le rayon du cercle pour qu'il soit tangent (=qu'il touche) au cercle existant le plus proche.
- 4- On répète l'opération pour le nombre de cercles voulus.

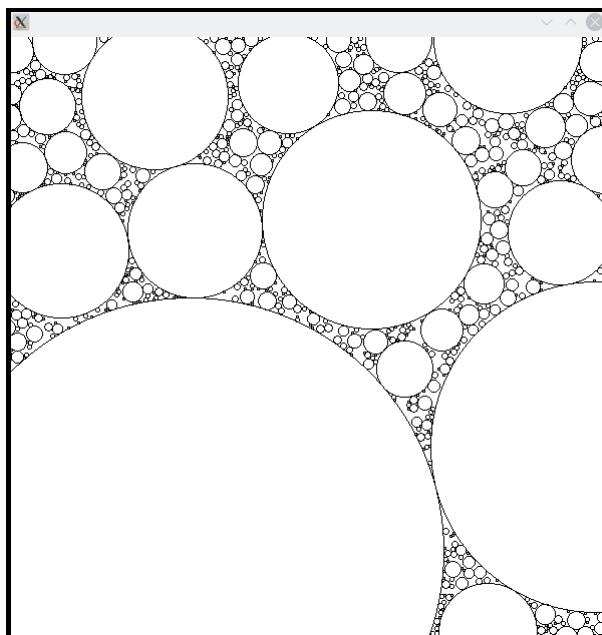


Figure 2 : Avec 2 cercles initiaux et 500 cercles tracés

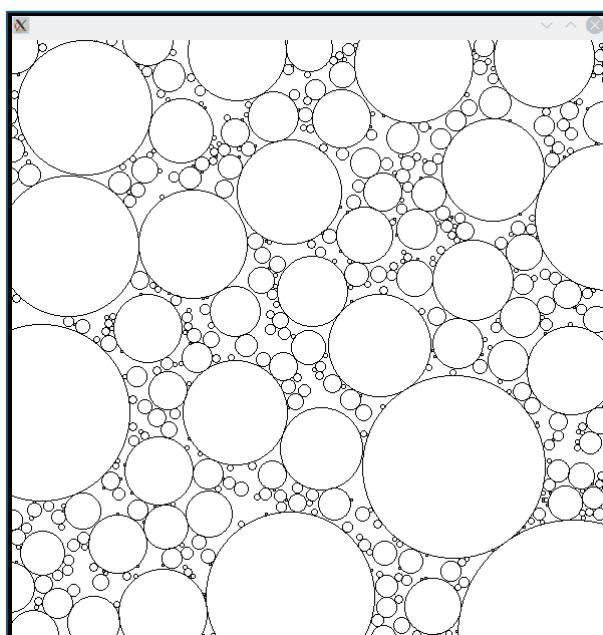


Figure 3 : Avec 10 cercles initiaux et 500 cercles tracés

Etape 3 :

Vous allez maintenant exécuter le même algorithme mais vous devrez contenir les cercles dans un grand cercle qui ne sera pas tracé :

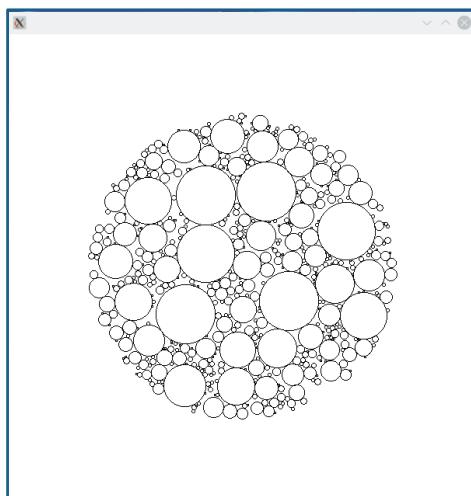


Figure 4 : Cercle externe $r=200$, 700 cercles tracés, 10 cercles initiaux

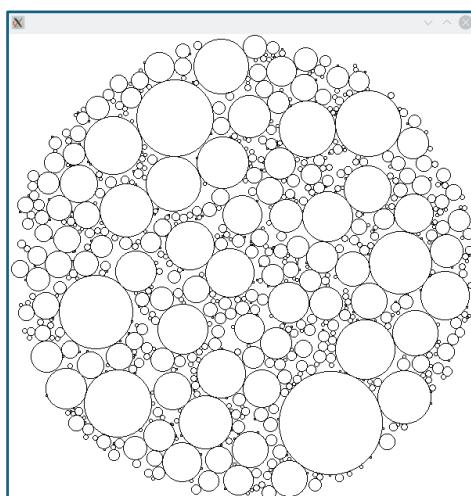


Figure 5 : Cercle externe $r=300$, 700 cercles tracés, 10 cercles initiaux

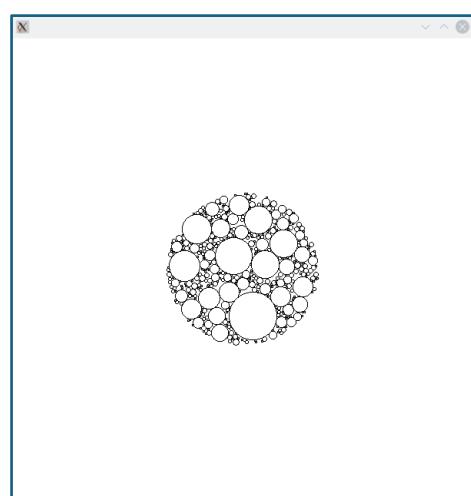


Figure 6 : Cercle externe $r=100$, 500 cercles tracés, 2 cercles initiaux

Etape 4 :

Nous allons à présent amener de la couleur dans nos cercles.

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).

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

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

Vous remplirez vos cercles de cercles concentriques en changeant les couleurs pour chacun des cercles concentriques selon votre tableau de couleur.

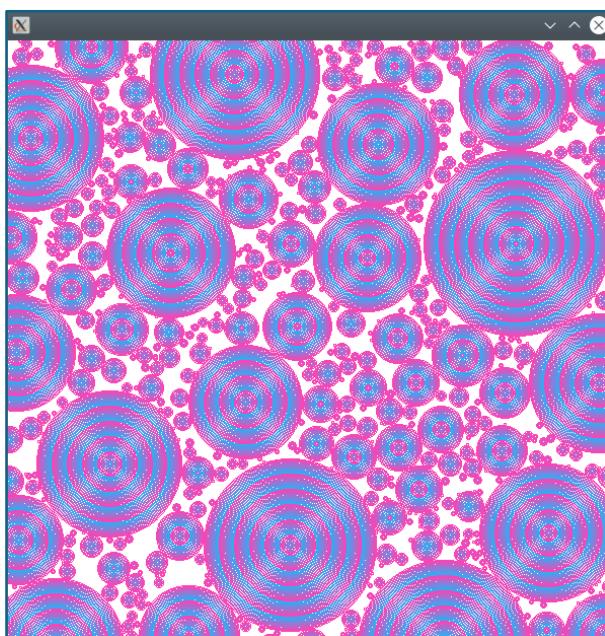


Figure 7 : Après étape 2, dégradé de 16 couleurs du rose au bleu, 700 cercles

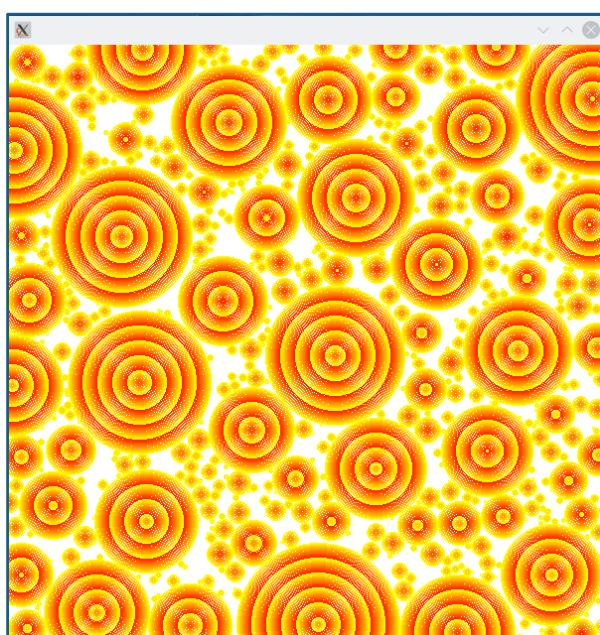


Figure 8 : Après étape 2, dégradé de 16 couleurs du jaune au rouge, 700 cercles

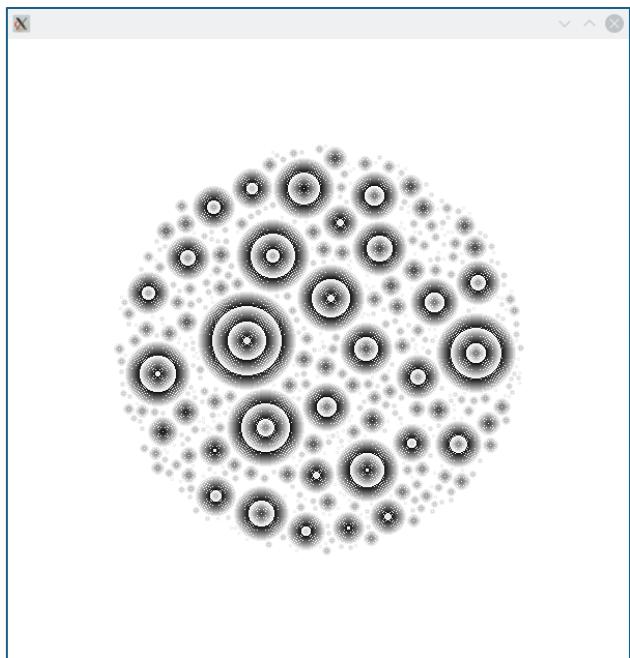


Figure 9 : Après étape 3, dégradé de 16 couleurs du blanc au noir, 700 cercles

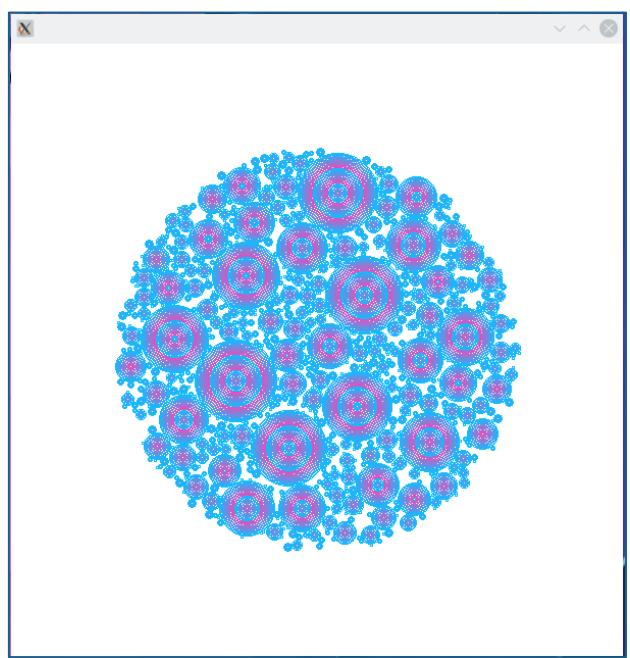


Figure 10 : Après étape 3, dégradé de 16 couleurs du bleu au rose, 700 cercles

2 – Attendus :

Vous devrez implémenter chacune des étapes individuellement.

Vous obtiendrez des points pour chacune des étapes.

L'étape 1 est nécessaire pour effectuer l'étape 2.

L'étape 2 est nécessaire pour effectuer l'étape 3.

Par contre, l'étape 4 peut être effectuée après l'étape 2 ou l'étape 3.

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)