

# Rapport TP2

## 8INF802 - Simulation de systèmes

## Partie 1:

La classe Generator implémente le générateur de nombre pseudo aléatoire de type récurrence linéaire à deux terme,  $X_i = aX_{i-1} + c \bmod m$  avec  $X_0$  pour seed.

On a  $a = 1664525$ ,  $c = 1013904223$  et  $m = 2^{32}$

Lorsqu'un générateur est créé il prends en argument une seed  $X_0$ .

La fonction *nextRand* permet de calculer  $X_i$  stocké dans *value*, et renvoie cette valeur divisée par  $m$  afin qu'elle soit comprise entre 0 et 1.

La fonction *nextInt* permet de générer un entier entre la borne inférieur (inclus) et la borne supérieur (inclus) donné en argument de la fonction.

## Partie 2:

Pour tester le générateur, j'ai simulé 1000 fois le lancé de deux dés afin de faire la somme du résultat des deux dés.

On a comme hypothèse Nulle  $H_0$  : Les données générées peuvent avoir été échantillonnées selon la distribution uniforme

La seed utilisé pour obtenir le tableau de résultat suivant est 164635156.

somme des dés	2 ou 3	4	5	6	7	8	9	10	11 ou 12
proportions du générateur	74	87	125	127	183	130	105	85	84
proportions cible	83	83	111	139	167	139	111	83	83

On calcul ensuite  $C = \sum_{i=1}^k \frac{(Y_i - E_i)^2}{E_i}$  avec  $Y_i$  les proportions obtenues grâce au générateur et  $E_i$  les proportions recherchées.

On obtient  $C = 6.4706$  sachant que  $\chi^2_{0.05,8} = 15.51$

$C < \chi^2_{0.05,8}$  On peut donc dire que  $H_0$  vraie. Notre générateur est donc valide et donne de bon résultat.

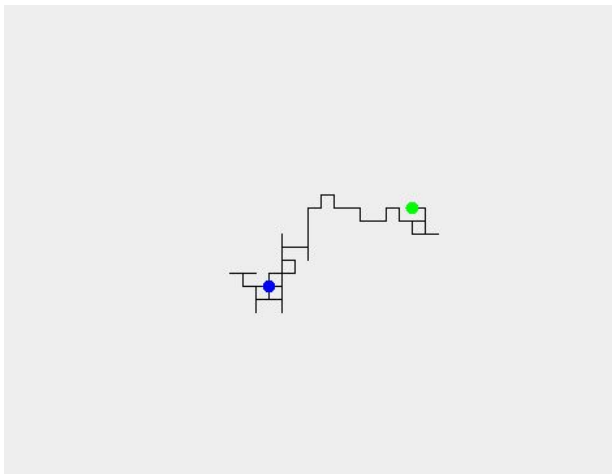
## Partie 3:

La classe *RandomWalk* implémente les trois types de marche aléatoire demandés avec les fonctions *ClassicWalk()* pour la marche Classique, *noRetrunWalk()* pour la marche sans retour et *uniquePassageWalk()* pour la marche avec passage unique.

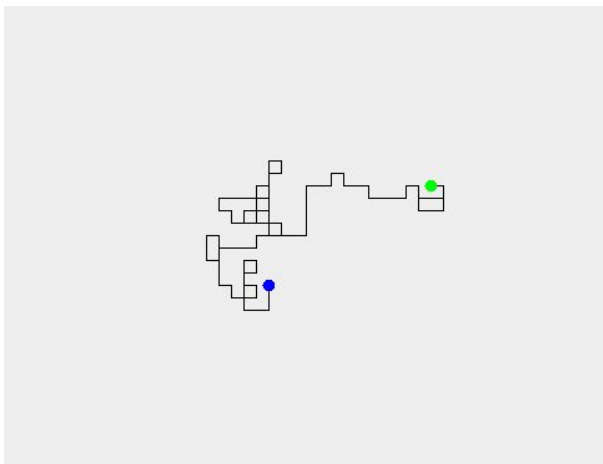
Lors de la création d'un objet on appelle le constructeur *RandomWalk(int n, String type, int seed)* avec *n* le nombre de pas à réaliser, *type* le type de marche ( "C" pour classique, "S" pour sans retour et "U" pour passage unique) et *seed* la seed pour créer le générateur.

La fonction *startWalk()* permet d'exécuter la bonne fonction selon le type de la marche aléatoire de *RandomWalk* et renvoie la distance entre le point de départ et le point d'arrivée.

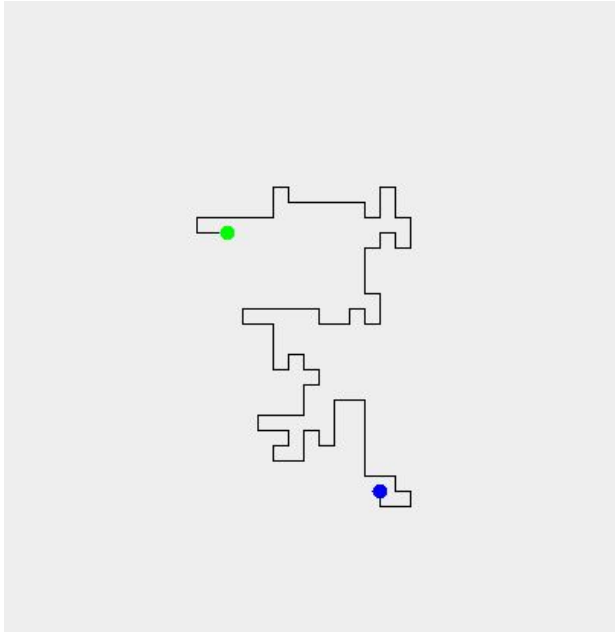
Les résultats suivant ont été obtenu avec une *seed* = 34525645 et un nombre de *step* = 100 (le point vert correspond au point de départ et le bleu au point d'arrivé)



Cas marche classique type = "C"



Cas marche sans retour type = "S"

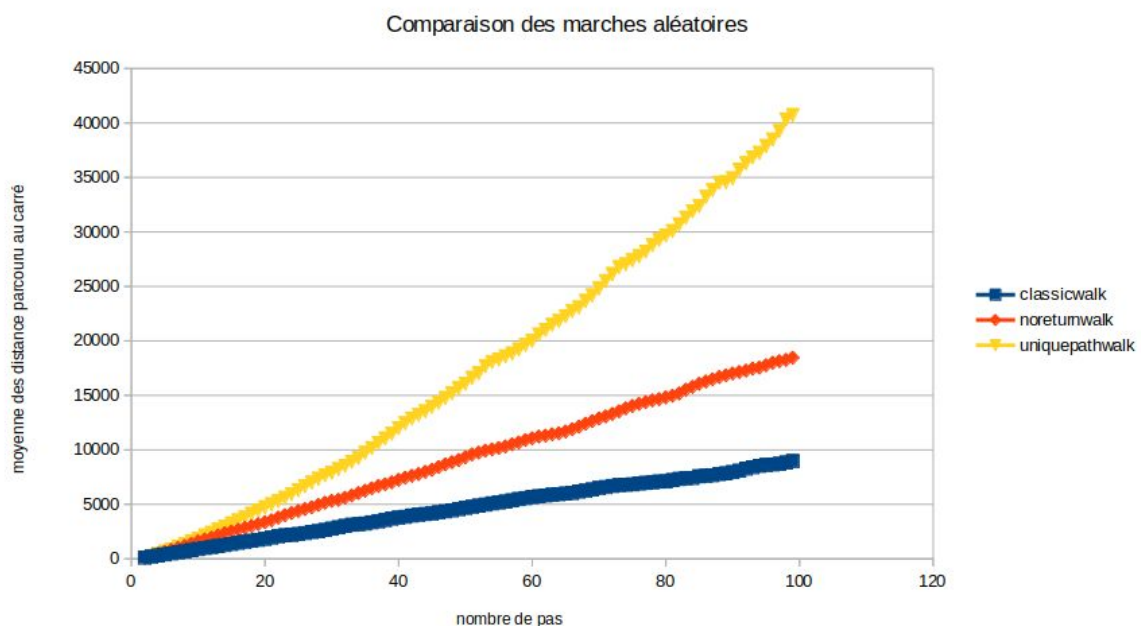


Cas marche unique passage type = "U"

Note : Si le nombre de pas est trop grand (exemple 500), il est possible que le chemin obtenu soit faux ou que le programme n'en trouve pas selon la seed utilisé.

La classe *CompareWalk* permet de remplir les fichiers "outClassic.txt", "outNoReturn.txt" et "outUnique.txt" avec la moyenne des distances au carré entre le point de départ et le point d'arrivée selon le nombre de pas. Les arguments pour exécuter le main sont *seed* pour la première seed utilisée pour le test puis *numberOfTry* le nombre d'essais pour chaque nombre de pas. Les résultats sont ensuite utilisés pour créer des graphiques sur Excel.

Pour obtenir les résultats suivants *seed* = 16161131 et *numberOfTry* = 1000 et le nombre de pas va de 2 à 100



## Manuel d'utilisation:

Pour tester le programme allez dans le dossier executable, vous pourrez y trouver 3 fichiers .jar :

La commande : `java -jar TP2_TestGenerator.jar`

permet d'exécuter le main de la classe TestGenerator qui affichera les résultats décrits dans la partie 2.

La commande : `java -jar TP2_RandomWalk.jar [int n] [String type] [int seed]`

permet d'exécuter le main de la classe RandomWalk et d'afficher le chemin obtenu

exemple : `java -jar TP2_RandomWalk.jar 100 U 16112645`

La commande : `java -jar TP2_CompareWalk.jar [int seed] [int numberOfTry]`

permet d'exécuter le main de la classe CompareWalk et de générer les fichiers "outClassic.txt", "outNoReturn.txt" et "outUnique.txt" stockant la distance au carré entre le départ et l'arrivée selon le nombre de pas.

exemple `java -jar TP2_CompareWalk.jar 1654226 1000`