

INF304 — TP8

Curiosity Revolutions (3) Génération aléatoire de terrains

Ce TP est la suite des TP précédents ; placez-vous dans le répertoire de ces TP précédents, et copiez-y les fichiers nécessaires au TP8 :

```
cp -r /Public/304_INF_Public/TP8/* .
```

Au cours de ce TP, vous serez amenés à compléter des programmes fournis. Il est important de vous les approprier (relire les en-têtes de chacun afin de voir ce qu'ils contiennent par exemple).

De même, vous devrez compléter le fichier `Makefile` des TPs précédents, pour permettre la création des différents exécutables de ce TP.

Génération de terrains

► Exercice 1.

Compléter la fonction `generation_aleatoire` du paquetage `generation_terrains`. Compléter le programme `test_generation_terrains.c` afin de pouvoir générer un fichier de terrains contenant n terrains de dimensions $L \times H$ avec une densité d'obstacles $dObst$ (densité entre 0 et 1) pour les cases occupées, chaque terrain généré devant être valide, c'est-à-dire posséder un chemin de cases libres du centre au bord (utiliser la fonction `existe_chemin_vers_sortie` fournie). On positionnera le robot au centre du terrain.

Pour la génération aléatoire d'un terrain, vous utiliserez un des algorithmes vus en cours.

► Exercice 2.

Écrire dans le fichier, en plus de chaque terrain, la densité d'obstacle obtenue (à comparer avec la densité donnée attendue) et à la fin du fichier la densité d'obstacle moyenne sur les terrains ainsi que le pourcentage de terrain valides par rapport au nombre total généré.

Test de performance

► Exercice 3.

Écrire un programme `curiosity-perf` sur la base du programme précédent et en prenant exemple sur `curiosity` afin de tester la «performance» du programme

utilisé.

Le programme `curiosity-perf` devra prendre comme arguments :

```
curiosity-perf fichier_programme N L H d graine nb_step_max fichier_res
```

où :

- `fichier_programme` est le fichier contenant le programme-robot évalué
- `N` est le nombre de terrains utilisés pour l'évaluation
- `L`, impair, la largeur des terrains
- `H`, impair, la hauteur des terrains
- `d`, la densité d'obstacles
- `graine`, la graine du générateur aléatoire
- `nb_step_max`, le nombre de pas maximum pour chaque exécution du programme sur un terrain
- `fichier_res`, le nom du fichier dans lequel seront écrits les résultats.

Dans ce cas :

- Le fichier `fichier_programme` doit contenir un programme «infini».
- Le fichier `fichier_res` sera écrit avec les statistiques de résultat, avec le format suivant :
 - **[ligne 1]** : `n` (nombre de terrains testés)
 suivie de `n` lignes, une valeur par ligne
 - **[ligne i+1]** (pour le terrain numéro `i`) :
 - si `curiosity` est sorti, le nombre de pas effectués
 - sinon, `curiosity` est «bloqué» (-1), «tombé dans l'eau» (-2), «écrasé contre un rocher» (-3)
- enfin, le programme affiche (sur la sortie standard) des statistiques pertinentes de «performance» du programme : nombre et pourcentage de terrains d'où le robot est sorti/où il est resté bloqué/où il est rentré dans un obstacle ; nombre moyen de pas effectués pour les sorties.

Élaboration d'un programme plus complexe

Il est possible de créer des programmes infinis comme (voir annexe TD6 et 7) :

- Fait avancer indéfiniment le robot : «{ **A C !** } **C !**»
- Mesure la case devant lui et avance si la case est libre, tourne à gauche s'il y a un obstacle : «{ **1 M { G } { A } ? C !** } **C !**»

Soit le schéma général : «{ **cmds C !** } **C !**», exécutant indéfiniment la séquence de commandes «**cmds**».

► Exercice 4.

En partant de ce principe, élaborer PLUSIEURS programmes permettant au robot de tenter de sortir de n'importe quel terrain.

NB : il faut que vous soyez capable d'*expliquer* le comportement de ces programmes !

► **Exercice 5.**

Tester ces programmes sur différents terrains.

NB : sur de grands terrains ou un nombre de pas maximum important, il peut être nécessaire d'augmenter la taille de la pile.

Pour chacun de ces programmes, écrire :

- un terrain sur lequel le programme permet au robot de sortir ;
- un terrain sur lequel le robot reste bloqué (rentre dans un obstacle, ou boucle indéfiniment sans sortir du terrain).

Savez-vous *caractériser* les terrains sur lesquels le programme permet/ne permet pas au robot de sortir ?

► **Exercice 6.**

Tester au moins deux de vos programmes sur des terrains de diverses densités d'obstacles. Comparer les résultats statistiques de sortie, pour différentes valeurs de paramètres.

Compte Rendu

Déposer sur la page moodle du cours :

- Un document PDF avec la description de deux programmes plus performants que celui de base («Avancer indéfiniment»),
- Pour chacun des deux programme, les fichiers résultats du programme `curiosity_perf` pour les terrains générés :
 - $n=20, L = H = 25$, occupation totale de 40%
 - $n=20, L = H = 9$, occupation totale de 70%
 - Donner pour vos deux programmes les paramétrages de terrain où chacun des programmes :
 - sort «presque» toujours
 - ne sort «presque» jamais