

Labyrinthe

Bruno Guillon

bruno.guillon@liafa.univ-paris-diderot.fr

PI4 – Projet Informatique

Résumé

Le but de ce projet est de réaliser un petit jeu basé sur les labyrinthes. L'utilisateur pourra déplacer un personnage dans un labyrinthe afin d'en trouver la sortie.

1 Labyrinthe

Un labyrinthe est un graphe *connexe* sans *cycle* (autrement dit un arbre) dont deux sommets sont marqués et donnés (l'*entrée* et la *sortie*). Autrement dit, entre chaque paire de sommets (ou *positions*) il existe un unique *chemin simple* (sans demi-tour).

De base le projet devra générer aléatoirement un labyrinthe sur une grille rectangulaire, de taille arbitraire. Chaque position est une case de la grille qui peut être connectée ou non à ses 4 voisines.

1.1 Génération aléatoire

La génération aléatoire se fera à l'aide de la structure UNION-FIND, dont vous trouverez une riche documentation sur wikipedia (par exemple).

L'idée principale est de partir d'une grille dans laquelle aucune case n'est connectée à une autre (c'est à dire un graphe sans arête). Ainsi, chaque case est une composante connexe. Puis on connecte aléatoirement des cases de composantes connexes différentes (de façon à ne pas créer de cycle), jusqu'à n'avoir plus qu'une composante connexe.

Votre programme devra être suffisamment général pour s'adapter à d'autres sortes de graphe, tels que les grilles hexagonales, les tores, ou les graphes non plainaires (cf. Partie 3).

1.2 Vérification

Il peut être utile d'avoir une fonction de vérification, qui vérifie qu'un "graphe" est bien un labyrinthe. Vous pouvez aussi créer une fonction d'édition, qui permet de créer un labyrinthe à la main (vérifiez que c'est bien un labyrinthe

lors de la validation). Enfin, n'attendez pas d'avoir une interface graphique pour afficher et tester votre générateur !

2 Interface

Vous commencerez par une interface graphique simple, ou tout le labyrinthe est affiché (avec mise à l'échelle en fonction de sa taille). Vous y ajouterez ensuite un personnage (un symbole simple) que vous pourrez déplacer dans les quatre directions.

Ajoutez-y la possibilité d'avoir une visibilité réduite (seule une petite partie du labyrinthe autour du personnage est visible).

3 Pour aller plus loin

Vous devez implanter **au moins une** des extensions suivantes, les bonus seront appréciés à leur juste valeur :

1. Le personnage a une position fixe sur l'écran : c'est le labyrinthe qui se décale sous ses pieds. Il est commandé par trois commandes : avancer, tourner (de 90°) à gauche et tourner (de 90°) à droite.
2. Même question, mais c'est le labyrinthe qui tourne lorsque qu'on veut tourner. Ainsi le personnage est fixe et toujours orienté vers le haut.
3. Le labyrinthe est plus compliqué et peut comporter des tunnels, ou des téléporteurs, ou des passages à un seul sens, ou est un tore. . .
4. Une version multijoueur du jeu : une course, à partir de l'entrée, le premier qui arrive à la sortie a gagné.
5. (*) Même chose en réseau.
6. Une intelligence artificielle qui cherche la sortie (mais qui n'a pas accès à l'ensemble du labyrinthe, bien sûr).
7. (*) Même chose avec différents niveaux.
8. Ajouter des objets dans le labyrinthe, tel qu'une échelle, qui permet d'escalader (une fois seulement) un mur, une dynamite pour casser un mur, des clefs qui permettent d'ouvrir des portes, ou des cailloux pour jouer au petit poucet. . .
9. Du graphisme avancé avec différentes textures pour les murs (forêt dense, mur de brique, crocodiles. . .).
10. Ajouter des monstres (avec des déplacements simple, par exemple aléatoire par défaut mais attiré par le personnage lorsqu'il est visible).
11. Une visibilité réelle : on ne voit que jusqu'au prochain mur, dans chaque direction. . .
12. Une autre bonne idée **à discuter avec votre chargé de td. . .**