

Deuxième partie

Projet ISN :12/13

La première partie avait pour but de vous familiariser avec l'environnement et d'étudier deux algorithmes de génération de labyrinthes : aléatoires et fractals.¹

Dans cette deuxième partie nous allons continuer notre cheminement dans le monde des labyrinthes...

1. Qualité d'un labyrinthe

Les premiers labyrinthes construits ont des allures bien différentes. Écrire en python des programmes prenant en entrée un labyrinthe et calculant :

- sa densité : c'est à dire le ration cases noires / cases blanches,
- sa plus grande ligne droite,
- son nombre de cul-de-sacs,
- son nombre de carrefours.

Les deux derniers points se calculeront plus facilement sur des labyrinthes dont le couloirs ne font pas plus d'une case de largeur, vous aurez des exemples dans le répertoire **qualite**.

2. D'autres algorithmes

Comme vous aurez pu le constater en parcourant les ressources à votre disposition sur le sujet, il existe de nombreux algorithmes de génération de labyrinthes.

L'algorithme de Prim en fait partie.

Votre travail consiste à :

- comprendre cet algorithme, c'est à dire à être capable à l'aide d'une feuille et d'un stylo (et d'une gomme !) de générer des labyrinthes de petites tailles,
- le comparer avec un autre algorithme de votre choix (autre que ceux vus précédemment),
- le coder en python.

1. J'espère pouvoir voir bientôt de jolies figures en particulier sur les labyrinthes fractals (autres que celles données dans l'énoncé)