

Production de labyrinthes 3D

ALoui Mathias, BADAGBON Jacques, BERTIN-JOHANNET Roland, POVEDA Maxime, RAHNI Dyhia

Aix-Marseille Université, France, `prenom.nom@etu.univ-amu.fr`

1 Des labyrinthes cubiques

Nous allons tout d'abord créer des labyrinthes cubiques seulement. La forme régulière et simple du cube va nous permettre de développer plus facilement les labyrinthes pour ensuite adapter nos techniques pour d'autres formes telles que des pyramides ou des sphères.

1.1 Décomposés en étages

Nous nous sommes inspirés des puzzles cubiques produit par INSIDE3[1].

- Chaque labyrinthe est composé de plusieurs étages empilés l'un sur l'autre.
- Chaque étage est composé de deux demi-étages.
- Chaque demi-étage est un labyrinthe carré 2D.

Pour résoudre le labyrinthe, il faut résoudre tous les étages. Chaque étage à une entrée sur le dessus qui permet d'accéder au premier demi-étage, un passage entre le premier demi-étage et le second et une sortie sur le dessous du deuxième demi-étage. Le principe :

1. Mettre la bille dans l'entrée du premier étage.
2. Résoudre le premier demi-étage pour atteindre le point de passage vers le second demi-étage.
3. Résoudre le second demi-étage pour atteindre la sortie de l'étage actuel.
4. Reprendre depuis l'étape 2 jusqu'à ce que tous les étages soient résolus.

1.2 Modulables

Le choix de décomposer chaque étage en deux demi-étages n'est pas anodin. Cela va nous permettre de placer sur tous les étages l'entrée et la sortie au même endroit (au centre). Ainsi, il sera possible au client d'empiler les étages dans l'ordre de son choix et même de les retourner. Le client pourra donc modifier le labyrinthe à sa convenance et combiner les étages de plusieurs labyrinthes ou acheter de nouveaux étages afin de créer ses propres labirythes.

1.3 Avec plusieurs niveaux de difficulté

Il nous est possible de créer des étages avec des difficultés différentes. Nous avons déjà pensé à 3 niveaux de difficultés possibles :

- Facile :
Les deux demi-étages sont identiques. Une fois le premier résolu, il suffit de faire le chemin inverse dans le second demi-étage pour atteindre la sortie.
- Moyen :
Les deux demi-étages sont différents. Il faut résoudre le premier demi-étage pour atteindre le second qui doit être également résolu pour atteindre la sortie.



© ALOUI, BADAGBON, BERTIN-JOHANNET, POVEDA, RAHNI;
sous licence Creative Commons CC-BY

- Difficile :

Les demi-étages sont partitionner en sous-labyrinthe et plusieurs points de passage permettent de passer du premier demi-étage au second et inversement. Il faut résoudre une partie du premier demi-étage pour atteindre le premier point de passage vers le second demi-étage, puis résoudre une partie du second demi-étage pour atteindre le second point de passage vers le premier demi-étage, etc... jusqu'à ce que l'on puisse atteindre la sortie.

2 Génération des labyrinthes

Nous avons choisi d'implémenter la génération de labyrinthe dans le langage Python.

2.1 Représentation sous forme de graphes

Nous allons représenter les demi-étages sous forme de graphes. Chaque composante d'un graphe correspondra à une cellule du labyrinthe. La composante contiendra des informations permettant la génération du demi-étage et les arrêtes correspondront aux murs entre les cellules.

2.2 Algorithme de fusion aléatoire de chemins

Nous allons utiliser une version modifiée de l'algorithme de fusion aléatoire de chemins[2]. L'algorithme permet de créer un labyrinthe parfait, c'est-à-dire un labyrinthe où toutes les cellules sont reliées entre-elles par un unique chemin. Nous allons donc le modifier dans le but de créer des labyrinthes simples pour les niveaux de difficultés facile et moyen ou des labyrinthes partitionnés pour le niveau de difficulté difficile. Nous allons donc pouvoir jouer sur le nombre de partitions dans un demi-étage ainsi que les tailles de celles-ci.

3 Une interface 3D

Pour l'affichage du labyrinthe, nous allons utiliser le logiciel Blender et cela pour deux raisons :

- une prise en main relativement simple et rapide.
- la compatibilité avec le langage Python.

Nous devrions être en mesure d'afficher les étages ainsi qu'un labyrinthe complet (cube).

4 Prochaines étapes

1. Commencer l'implémentation des graphes et de l'algorithme afin de créer des labyrinthes 2D (demi-étage).
2. Utiliser Blender pour afficher ces labyrinthes 2D.
3. Implémenter la création d'un étage complet.
4. Implémenter la création d'un labyrinthe complet.
5. Utiliser Blender pour afficher ces labyrinthes 3D.
6. Implémenter les niveaux de difficulté.

Références

- 1 Doug Solutions. Inside ³ - le labyrinthe 3d. URL : <https://www.insidezecube.com/>.
- 2 Wikipédia. Modélisation mathématique de labyrinthe — wikipédia, l'encyclopédie libre, 2020. [En ligne; Page disponible le 2-janvier-2020]. URL : http://fr.wikipedia.org/w/index.php?title=Mod%C3%A9lisation_math%C3%A9matique_de_labyrinthe&oldid=165955332.