BRESSET NICOLAS
BISOTTO ROMAIN

PROJET JAVA AVANCE

2022

Pour le generator on a décidé de créer 4 booléens qui stockent s'il y a ou pas des connections dans les 4 directions. Le nord et l'ouest sont calculer à partir des pièces précédentes ou du bord du plateau et l'est et le sud sont tiré au sort selon un argument passé en paramètre. Ensuite on fait tourner chaque pièce au hasard pour les mélanger.

Pour le solveur au début, on testait toutes les combinaisons possibles selon un algorithme depht first, mais pour un tableau de n*m la complexité est de 4^(n*m), alors, pour le simplifier, on a trouvé un moyen de fixer les pièces en fonction de leur placement à côté des bordures afin de trouver la seule position possible de la pièce, les fixer et essayer d'en déduire les pièces adjacentes, en effet chaque pièce fixé au préalable divise par 4 la complexité de l'algorithme et une fois le maximum de pièce fixé on peut utiliser l'algorithme depth first pour trouver la position des dernières. Mais pour encore plus optimiser, l'algorithme depth first on test la branche visitée pour savoir si elle est possible, et si elle ne l'est pas on retourne directement en arrière. Pour implémenter ça on a fait un algorithme récursif.

On a choisi de faire une interface graphique pour faciliter les tests du programme, et cette dernière est assez développée pour pouvoir se passer des options en ligne de commande.

Si vous avez d'autres questions, n'hésitez a nous contacter a nicolas.bresset@dauphine.eu ou romain.bisotto@dauphine.eu.