

SUJET SAE 2.02 : Exploration algorithmique d'un problème

L'Apprenti Ordonnateur



Introduction

Au royaume de Krystalia, les sorciers, magiciens et autres enchantereurs pullulent dans tous les coins. En effet, les énergies magiques abondent, grâce aux cristaux de pouvoir, répartis dans le pays dans les différents temples de magie. Chaque temple est dédié à une couleur, et possède en son coeur un cristal qui doit être de la même couleur : on dit que les cristaux sont alignés. De temps en temps, des sorciers dépassent un peu les bornes de ce qui est permis par la magie des mortels, ce qui déclenche des tempêtes de magie. Suite à une telle tempête, les cristaux changent de couleur **et subissent une permutation**. Ainsi, le cristal du temple vert devient rouge, celui du temple rouge est jaune, etc. Il convient alors de réaligner les cristaux, et la méthode la plus simple est de les déplacer pour les amener au temple de leur couleur. C'est le travail du Grand Ordonnateur de Krystalia de procéder à ce réalignement après chaque tempête. Malheureusement, le grand Ordonnateur est en vacances quand se déclenche une énorme tempête qui a

permuté tous les cristaux. Et c'est à vous, l'apprenti ordonnateur, qu'incombe la difficile tâche du réalignement des cristaux...

La carte

Un scénario est caractérisé par la liste des temples avant le réalignement, donnés par leurs coordonnées, leur couleur puis la couleur du cristal qu'ils contiennent. S'il y a N temples, les couleurs seront encodées par les entiers de 1 à N. Un scénario est encodé dans un fichier texte, et contient N lignes contenant 4 entiers, comme "12 -4 3 7", ce qui signifie :

- la position du temple est aux coordonnées (12,-4)
- la couleur du temple est 3
- le temple contient le cristal de couleur 7

Les règles

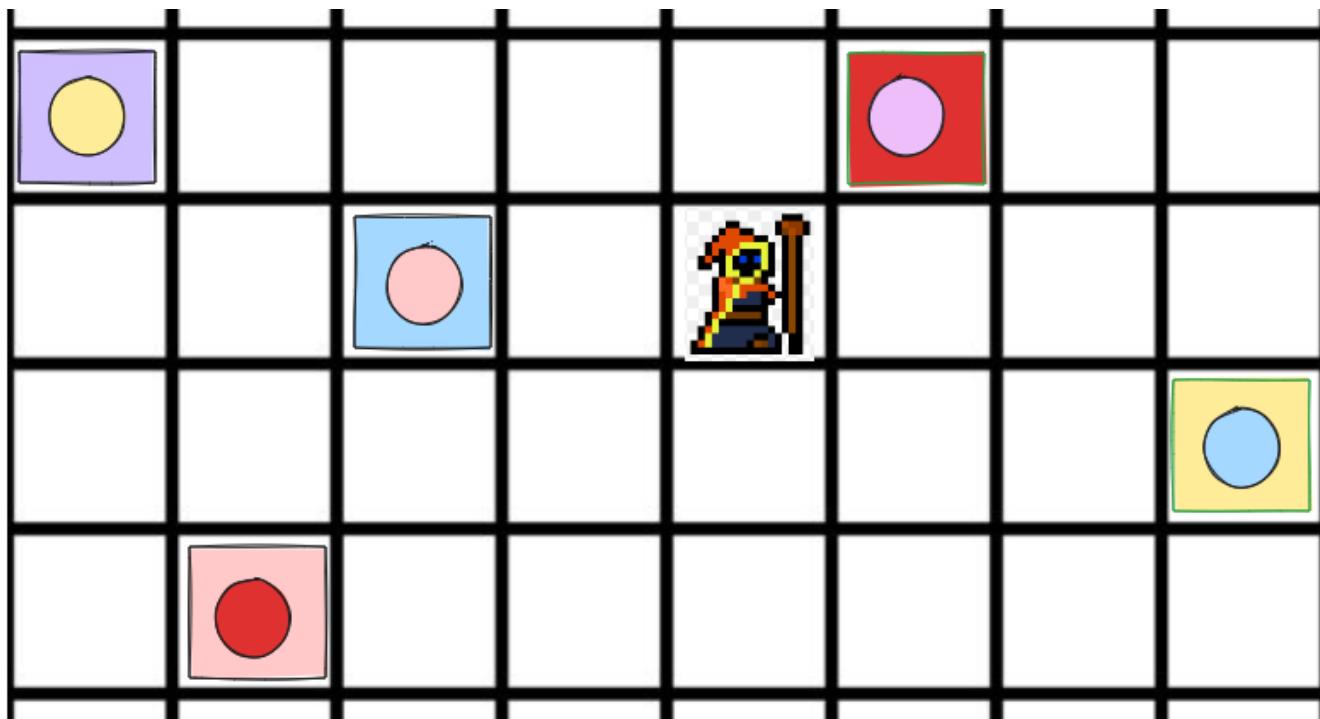
- L'apprenti commence sur la case (0,0) et ne peut se déplacer que d'une case par tour, dans les quatre directions de base.
- L'apprenti ne peut porter qu'un seul cristal à la fois, à cause des interférences. Au départ, l'apprenti ne porte aucun cristal.
- Si l'apprenti ne possède pas de cristal, en entrant sur la case d'un temple il peut emporter le cristal du temple. S'il porte un cristal, il peut échanger les cristaux, en déposant au temple celui qu'il portait et en emportant celui du temple.
- L'apprenti ne peut pas déposer un cristal sur une case vide ou dans un temple contenant déjà un cristal, sauf échange dans un temple comme expliqué ci-dessus.
- L'apprenti peut également traverser un temple comme si c'était une case vide, sans déposer ni prendre de cristal.

L'objectif de l'apprenti

L'objectif de l'apprenti est de réaligner les cristaux le plus rapidement possible, c'est à dire en minimisant le nombre total de déplacements (**la manipulation des cristaux ne prend pas de temps**). Typiquement, en partant de la case (0,0), il se rendra à un premier temple pour prendre un cristal, ira à un second temple déposer le premier cristal et prendre l'autre, et ainsi de suite. Vous pouvez vous rendre compte que les seules positions à réellement étudier sont celles où l'apprenti se trouve sur un temple, et les données pertinentes à ce moment pour caractériser la situation sont les positions des cristaux et de l'apprenti. Dès que l'apprenti apporte au bon temple le dernier cristal et que tous les cristaux sont alignés, l'objectif est considéré réalisé et on compte à ce moment le nombre total de déplacements effectués.

L'objectif du programmeur

Votre objectif est de réaliser un algorithme qui résolve le problème pour un scénario donné et une interface graphique permettant de visualiser les résultats. L'interface se compose de boutons, menus, zone de texte et d'une partie graphique (canvas). Voici un exemple de représentation graphique, que vous n'êtes pas obligés de suivre.



(échange des niveaux 1 et 2)

Niveau 0

L'interface permet de charger un scénario, de visualiser graphiquement la position des temples, des cristaux, de leur couleurs, et la position de l'apprenti. Quand on clique sur une case, l'apprenti se déplace de pas en pas jusqu'à cette case. Si la case est un temple qui ne contient pas de cristal, l'apprenti y dépose son cristal ou échange avec le sien. Un compteur affiche le nombre total de déplacements. On peut ainsi résoudre le problème "à la main".

Essayez que pour un scenario donné on puisse facilement réinitialiser et relancer un autre algorithme de résolution afin de comparer les résultats. Essayez également de ne pas demander de saisir des informations au clavier et que tout se fasse à la souris.

Niveau 1

Vous implémentez **un** algorithme de tri de votre choix (sélection, bulle, insertion etc). L'apprenti va considérer les temples dans l'ordre des couleurs comme s'il s'agissait d'un tableau et procéder par échanges. Un menu ou un bouton permet de lancer l'algorithme et de visualiser toute l'exécution.

Niveau 2

Vous implémentez un algorithme "heuristique" personnel qui tente de résoudre le problème le plus efficacement possible, et on peut visualiser son exécution. **Un algorithme heuristique est un algorithme qui ne résout pas forcément le problème de manière optimale mais qui est simple à concevoir et fournit une bonne approximation de l'optimal dans la plupart des cas.**

Niveau 3

Vous implémentez un algorithme permettant de calculer la solution optimale, l'interface affiche le nombre de pas nécessaires et permet également de visualiser l'exécution de cette solution.

Vous pouvez, bien entendu, ajouter ce que vous désirez dans ce programme, tant au niveau graphique qu'au niveau algorithmique, ce qui donnera lieu à des points bonus.

Si vous avez réalisé différents algorithmes, il serait pratique d'avoir une zone qui affiche les différentes valeurs obtenues par les différents algorithmes.