

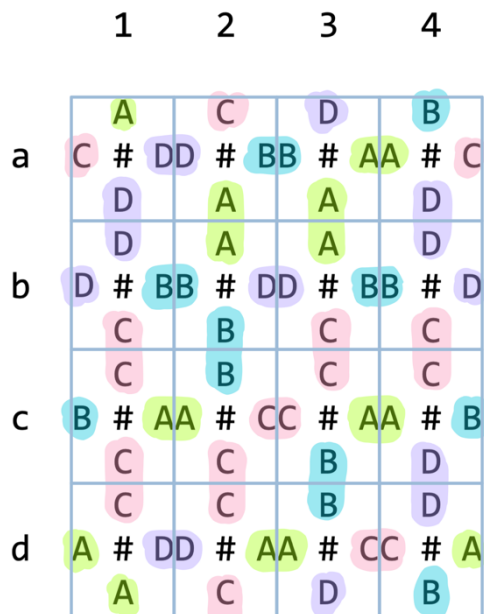
Projet de langage C

Introduction

Le but de ce projet facultatif est de réaliser un jeu de type puzzle qui s'inspire du célèbre Eternity II. Dans ce puzzle, on a des pièces carrées avec un motif sur chaque côté. On doit placer ces pièces sur un plateau carré de manière à ce que les côtés contigus de deux pièces voisines aient le même motif.

Voici une représentation sous forme textuelle, avec une police de caractères de largeur fixe, d'un puzzle terminé de 16 pièces sur une grille de 4 par 4. Les motifs sont des lettres majuscules. Un caractère # indique le centre de chaque pièce.

	1	2	3	4
	A	C	D	B
a	C # DD # BB # AA # C			
	D	A	A	D
	D	A	A	D
b	D # BB # DD # BB # D			
	C	B	C	C
	C	B	C	C
c	B # AA # CC # AA # B			
	C	C	B	D
	C	C	B	D
d	A # DD # AA # CC # A			
	A	C	D	B



La figure de droite identifie plus précisément les différentes pièces. Les motifs associés aux cotés des pièces sont représentés par des lettres majuscules. Le coté gauche de la première pièce d'une ligne est considéré comme contigu avec le coté droit de sa dernière pièce. De même, le coté supérieur de la pièce située en haut d'une colonne est considéré comme contigu avec le coté inférieur de la pièce située en bas de cette colonne.

Dans cette représentation, le puzzle est terminé. Mais dans le jeu, les pièces ne sont pas disposées correctement au départ, et l'utilisateur doit faire des modifications pour reconstituer une solution. Le tableau de jeu peut avoir différentes tailles : 4x4 (16 pièces), 5x5 (25 pièces), 6x6 (36 pièces), voire 7x7 (49 pièces).

Variantes

Vous pouvez choisir de travailler sur deux variantes, une simple et une complexe, décrites ci-dessous. Vous pouvez valider d'abord la version simple et ensuite la version complexe. Dans les deux cas, L'ajout d'une IA pourra vous rapporter des points supplémentaires et pourra être validée séparément si le projet est livré suffisamment tôt. Les points pouvant être rapportés par les différentes variantes sont donnés dans un tableau à la fin de ce document.

Puzzle simple

Dans cette version du jeu, les pièces sont bien orientées mais mal placées. Leurs positions initiales sont aléatoires. On doit pouvoir trouver une solution simplement en échangeant certaines pièces, sans les tourner. Pour échanger deux pièces, l'utilisateur saisit une chaîne de 4 caractères représentant leurs coordonnées. Par exemple, en saisissant b2c4, il demande d'échanger les pièces actuellement situées en b2 et en c4. Le programme affiche alors la nouvelle configuration et son nombre de conflits, c'est-à-dire de cotés contigus dont les lettres sont différentes. S'il n'y a aucun conflit, le programme affiche « Solution trouvée » et donne le nombre d'échanges ayant été réalisés pour obtenir cette solution.

Puzzle complexe

Dans cette version, les pièces peuvent être non seulement mal placées mais en plus mal orientées. L'utilisateur peut soit demander l'échange de deux pièces comme dans la version simple, soit la rotation d'une pièce d'un quart de tour en saisissant uniquement ses coordonnées. Par exemple pour demander la rotation de la pièce c3, il tape juste c3. Pour le reste, le fonctionnement est identique à celui de la version simple.

Mission spéciale

Cette mission consiste à doter votre programme d'une petite IA lui permettant de trouver lui-même une solution. Vous pouvez utiliser le principe de la recherche locale qui consiste à essayer de réaliser des modifications (échanges de pièces et rotations dans le cas de la version complexe) permettant de réduire progressivement le nombre de conflits. Mais il faut prévoir une part de hasard et autoriser parfois des modifications qui augmentent le nombre de conflits pour éviter un blocage dans une configuration ne pouvant pas être améliorée.

Livrables

Il vous est demandé de fournir un code fonctionnel pouvant être compilé et exécuté. Pour des raisons pratiques, le code complet doit être dans un unique fichier source. Il doit être bien présenté et commenté. En plus, vous devez fournir un petit dossier qui présente les structures de données et algorithmes utilisés de manière claire, concise et pédagogique, en au plus 5 pages. Dans ce mini rapport, vous ne devez pas détailler le code de vos fonctions, mais vous pouvez donner les déclarations de vos structures de données. Des dessins explicatifs, par exemple réalisés à la main et photographiés, seront très appréciés. Si vous terminez le projet, ou une partie du projet, avant la dernière séance de TP, vous pourrez faire une démonstration à votre enseignant de TP. N'hésitez pas à lui montrer vos résultats intermédiaires afin qu'il vous signale d'éventuels problèmes et vous conseille pour la suite.

La date limite de livraison est le mercredi 15 avril 2020.

Barème

Ce tableau donne le nombre maximum de points pouvant être obtenus avec les différentes versions du projet. L'IA peut être relativement simple et des indications vous seront données sur le canal Projet de l'équipe info4A Microsoft Team. Une IA plus sophistiquée pourra donner lieu à un bonus.

		version simple		version complexe	
		sans IA	avec IA	sans IA	avec IA
taille fixe (4x4)	solution initiale fixe	4	×	5	×
	solution initiale aléatoire	5	6	6	7
taille variable	solution initiale aléatoire	6	7	7	8

↓
taille variable : 4x4 ou 5x5 ou 6x6 ou 7x7 .

Solution initiale fixe signifie que votre configuration de départ est obtenue par mélange aléatoire des pièces d'une même et unique solution que vous avez réalisée à la main (comme celle que je donne en exemple). Solution initiale aléatoire signifie que vous devez produire aléatoirement une solution puis en mélanger les pièces pour créer la configuration initiale.