



UNIVERSITÉ PARIS 8 - VINCENNES À SAINT-DENIS

Algorithmique avancée

Solver de stitches

Ludovic RENAULT

Date de rendu : le 10/12/2023

Tuteur – Université : Benjamin DUPONT

Table des matières

1	Présentation	1
1.1	Tout savoir sur stitches	1
1.2	Bon fonctionnement et installation	2
1.2.1	Utilisation du Programme	2
2	Implémentation	5
2.1	Idée générale	5
2.2	Algorithme de résolution	6
2.3	Fonctions utiles	7
3	Amélioration et optimisation	9
3.1	Amélioration de l'algorithme	9
3.2	Proposition d'alternatives	10
3.3	Bonification de l'expérience utilisateur	10
4	Conclusion	11

Chapitre 1

Présentation

1.1 Tout savoir sur stitches

Stitches est un casse-tête captivant, il se distingue par sa simplicité apparente et la profondeur de réflexion qu'il exige des joueurs. Les règles du jeu sont conçues pour offrir une expérience engageante tout en stimulant la réflexion stratégique des participants.

Règles Fondamentales de Stitches :

- Relier Chaque Bloc à TOUS ses Voisins :
L'objectif principal dans Stitches est de relier chaque bloc à tous ses voisins. Cette connexion est établie par un "point de couture" entre deux blocs adjacents.
- Points de Couture entre Deux Blocs Adjacents :
Chaque "point de couture" représente une connexion entre deux cellules adjacentes, soit horizontalement, soit verticalement, de deux blocs distincts. Les joueurs doivent manœuvrer stratégiquement pour établir ces liens cruciaux.
- Interdiction de Points Convergeants :
Deux "points de couture" ne peuvent pas partir du même trou. Cette règle ajoute une couche de complexité au jeu, obligeant les joueurs à planifier soigneusement leurs mouvements pour éviter les intersections problématiques.
- Chiffres Indicateurs :

Les chiffres disposés autour de la grille servent de guides essentiels. Ils indiquent le nombre de trous présents dans la ligne ou la colonne respective. Les joueurs doivent respecter ces indicateurs pour atteindre la solution optimale.

1.2 Bon fonctionnement et installation

Assurez-vous d'avoir un compilateur C installé sur votre système. Vous pouvez utiliser un compilateur tel que GCC (GNU Compiler Collection) pour compiler le code source.

```
gcc -o stitches_game stitches.c -lm
```

Exécutez le Programme compilé :

Après la compilation réussie, exécutez le programme compilé pour lancer le jeu.

```
./stitches_game
```

Vous devriez maintenant voir le menu principal du jeu, prêt à explorer les grilles et défier votre esprit avec les puzzles de Stitches.

1.2.1 Utilisation du Programme

Choix de la Grille

Le programme vous présentera un menu avec plusieurs grilles parmi lesquelles choisir. Entrez le numéro correspondant à la grille que vous souhaitez résoudre.

Une fois la grille sélectionnée, le programme vous donnera la possibilité de résoudre la grille vous-même ou de demander la solution automatisée.

Si vous choisissez de jouer vous-même, suivez les instructions pour placer ou retirer des points de couture à l'aide des coordonnées fournies. Si vous optez pour la solution automatisée, le programme tentera de résoudre

la grille pour vous.

Veillez à bien changer la `MAX_SIZE` dans le code lorsque que vous voulez tester des grilles plus grandes ou plus petites.

En suivant ces instructions, vous devriez être en mesure d'installer et de faire fonctionner le jeu Stitches sans problème.

Chapitre 2

Implémentation

2.1 Idée générale

L'implémentation de Stitches a nécessité une réflexion approfondie sur la manière de rendre le jeu jouable, visuellement agréable et l'expérience globale plaisante pour les joueurs. La conception du jeu repose sur plusieurs aspects clés.

1. Structure de Données

La grille du jeu est représentée par une structure de données comprenant des cellules, des blocs, et des liens entre différentes zones. Chaque cellule contient des informations sur le point de couture, la zone à laquelle elle appartient, et un poids pour des fonctionnalités potentielles.

2. Affichage Visuel

Pour faciliter la compréhension visuelle, chaque zone dans la grille est représentée par un chiffre distinct. Lors de l'affichage, des couleurs sont attribuées à chaque zone, facilitant ainsi la distinction entre les différentes parties de la grille. Par exemple, une couleur spécifique peut être associée à chaque chiffre, offrant un repère visuel clair pour les joueurs.

3. Importation de Grilles

L'approche choisie pour intégrer des grilles dans le jeu est d'importer des configurations à partir de fichiers. Chaque fichier représente une grille avec des chiffres définissant les zones. Cette méthode permet d'ajouter de nouvelles grilles facilement, offrant ainsi une variété de

puzzles aux joueurs.

4. Expérience de Jeu

Les joueurs ont la possibilité de résoudre les grilles manuellement en plaçant des points de couture aux positions souhaitées ou de demander au programme de résoudre automatiquement la grille. Cette flexibilité permet aux joueurs de choisir entre défier leur logique ou observer la résolution automatisée.

5. Interaction Utilisateur

Pour améliorer l'interaction utilisateur, une fonctionnalité a été ajoutée pour permettre aux joueurs de placer et de retirer des points de couture dans la grille. Cela crée une expérience plus interactive et engageante, offrant aux joueurs un contrôle direct sur le déroulement du jeu.

2.2 Algorithme de résolution

L'algorithme de résolution de Stitches repose sur des choix de conception spécifiques pour garantir une implémentation efficace et logique. Nous allons maintenant explorer certains aspects clés du code pour comprendre pourquoi ces choix ont été faits.

Utilisation du Backtracking :

Le choix d'utiliser le backtracking découle de la nature même du problème. Stitches nécessite une exploration approfondie des différentes combinaisons de points de couture pour trouver la solution optimale. Le backtracking offre une approche permettant une exploration complète tout en évitant les branches qui conduisent à des solutions impossibles. Cela garantit de trouver une solution si elle existe.

Validation des mouvements :

La fonction `isValid` vérifie la validité d'un mouvement avant de l'effectuer. Cela garantit que les règles du jeu ne sont pas enfreintes, telles que l'ajout de points de couture sur des cellules déjà occupées ou la liaison des cellules qui ne seraient pas accolées.

Automatisation de la résolution

La fonction `solve` peut être utilisée pour automatiser la résolution des grilles.

Elle explore les différentes configurations jusqu'à trouver la solution. L'utilisation de la récursivité permet une approche systématique de l'exploration des possibilités.

2.3 Fonctions utiles

Explorons quelques-unes des fonctions utiles qui n'ont pas encore été mentionnées dans le contexte du jeu Stiches :

1. Fonction d'Initialisation
La fonction 'initializeGrid' est responsable de l'initialisation de la grille du jeu en fonction des données fournies, y compris la lecture de la grille depuis un fichier.
2. Fonction d'Affichage de la Grille Importée :
La fonction 'displayImportedGrid' affiche la grille importée avec des couleurs pour représenter visuellement les différentes zones.
3. Fonction de Vérification de la Solution :
La fonction 'checkSolution' vérifie si la grille actuelle constitue une solution valide du jeu.
4. Fonction de Lecture de la Taille depuis un Fichier :
La fonction 'readSizeFromFile' lit la taille de la grille à partir d'un fichier.
5. Fonction de Menu Principal :
La fonction 'displayMenu' affiche le menu principal du jeu, permettant au joueur de choisir la grille à résoudre.
6. Fonction de Lecture de Grille depuis un Fichier :
La fonction 'readGridFromFile' lit une grille depuis un fichier et la stocke dans une structure de données.

Ces fonctions contribuent à la gestion globale du jeu, de l'initialisation de la grille à la lecture des données depuis un fichier, en passant par la vérification de la solution et l'affichage du menu.

Chapitre 3

Amélioration et optimisation

L'implémentation d'un jeu tel que 'Stitches' soulève toujours la question cruciale de l'efficacité et de l'expérience utilisateur. Dans ce chapitre, nous examinerons divers axes d'amélioration et d'optimisation pour renforcer la qualité du code et la jouabilité du jeu. Nous aborderons des aspects tels que l'optimisation de l'algorithme existant, l'exploration d'alternatives plus performantes, et la proposition d'améliorations visant à enrichir l'expérience utilisateur. Ces suggestions visent à perfectionner le jeu et faire qu'il soit non seulement fonctionnel, mais aussi le plus agréable et réactif possible pour les joueurs.

De plus la taille des grilles n'est pas variable, un autre axe d'amélioration serait de rendre la taille maximale d'une grille variable pour pouvoir autant jouer sur des grilles de 5x5 ou 7x7 voire 10x10 sans changer le code et recompiler.

3.1 Amélioration de l'algorithme

L'intégration de l'élagage alpha-bêta constitue une amélioration significative de l'algorithme de backtracking dans le contexte du jeu Stitches. En ajoutant cette technique d'optimisation, le processus de résolution du puzzle peut être considérablement accéléré. L'élagage alpha-bêta permet d'éliminer certaines branches de l'arbre de recherche qui ne contribueraient pas à la découverte d'une solution optimale. Dans le cadre spécifique de Stitches, où la complexité du problème peut rapidement augmenter, cette optimisation devient particulièrement pertinente pour garantir des temps de résolution plus courts et une expérience de jeu plus réactive.

3.2 Proposition d’alternatives

Algorithme de Recherche Locale : Un autre type d’algorithme qui pourrait être exploré est l’algorithme de recherche locale. Celui-ci commence avec une solution initiale et explore les solutions voisines dans l’espoir de converger vers une solution optimale. Cela pourrait être particulièrement efficace si le backtracking prend trop de temps.

Algorithme Génétique : Les algorithmes génétiques sont souvent utilisés pour résoudre des problèmes d’optimisation combinatoire. Ils peuvent être adaptés à ce problème en considérant différentes combinaisons de points de suture comme des individus et en évoluant ces solutions au fil des générations.

Parallélisation : Si la performance est un problème, la parallélisation de certaines parties de l’algorithme pourrait être envisagée. Cela pourrait accélérer le processus sur des systèmes avec plusieurs cœurs.

Algorithme Heuristiques : Les algorithmes heuristiques sont conçus pour trouver des solutions rapidement, même s’ils ne garantissent pas toujours la solution optimale. Un exemple d’algorithme heuristique qui pourrait être adapté au problème de Stitches est l’algorithme A*. A* utilise une fonction heuristique pour évaluer la probabilité qu’une certaine branche de recherche conduise à une solution. Il explore les branches les plus prometteuses en priorité, ce qui peut accélérer la recherche de solutions dans un espace de recherche complexe comme celui de Stitches.

Cependant, il est important de noter que l’utilisation d’algorithmes heuristiques introduit souvent un compromis entre la rapidité de résolution et la garantie d’optimalité. Dans le contexte de Stitches, où la recherche de la solution optimale est le but principal, cela pourrait nécessiter des ajustements appropriés de la fonction heuristique pour assurer une bonne balance entre la qualité de la solution et le temps de calcul.

3.3 Bonification de l’expérience utilisateur

Pour améliorer l’expérience utilisateur, l’ajout d’une interface graphique interactive pourrait rendre le jeu plus convivial et permettre aux joueurs de placer ou retirer des points de suture de manière intuitive.

Chapitre 4

Conclusion

En guise de conclusion, nous avons donc ici un solver du jeu 'stitches' qui résoud les grilles de 5x5 et 7x7. Il utilise un algorithme dit 'naïf' qui lui permet de parcourir tout l'arbre de possibilité et donc résoudre un puzzle dans un temps imparti. Il peut être encore optimisé pour résoudre des grilles plus rapidement et /ou plus grandes. Ou alors d'autres approches peuvent aussi être une autre source d'optimisation.