Projet du sodoku

Partie 0 du Projet - Comprendre le Sodoku

Le vendredi 20 octobre 2023

1. Commencez par discuter de ce que nous avons appris du jeu de Sodoku.

2. Chaque membre du groupe explique sa vision personnelle du jeu, ses règles et son objectif.

3. Élaborez sur les points clés et les règles du Sodoku pour vous assurer que tous les membres du groupe ont une compréhension commune du jeu.

4. planifier la partie 1

Partie 1 du Projet - Développement des premiers Algorithmes :

1. Algorithme de Génération de Grille Résolue :

- Créez un algorithme qui génère de manière aléatoire une grille de Sodoku résolue.

- Expliquez comment cet algorithme fonctionne s’assurer qu'il respecte les règles du Sodoku.

2. Algorithme de Génération de Grille Non Résolue :

- Développez un algorithme qui utilise la grille générée précédemment pour créer une nouvelle grille de Sodoku non résolue.

- Cette nouvelle grille doit avoir une unique solution, la grille précédemment générée.

3. Algorithme d'Affichage dans le Terminal :

- Créez un algorithme qui affiche la grille de Sodoku dans le terminal.

- Assurez-vous que l'affichage est clair et facile à lire pour les utilisateurs.

Compte rendu

Note de la réunion avec Madame Bouhlla qui a eu lieu le 27/10/2023,

1. Avancement du Projet

Madame Bouhlla a noté que l'équipe a déjà réalisé des progrès significatifs dans le projet. Cependant, elle a souligné l'importance de répartir les tâches de manière précise pour éviter de les accomplir simultanément.

1. Partie 2 du Projet

Madame Bouhlla a discuté de la prochaine phase du projet, en mettant l'accent sur les aspects suivants :

1. Niveaux de Difficulté: Il a été suggéré de diviser le jeu en trois niveaux de difficulté :

- Facile : 15 % de la grille vide.

- Moyen : 35 % de la grille vide.

- Difficile : 60 % de la grille vide.

1. Méthode de Gestion de la Difficulté : Il a été proposé de créer une méthode pour gérer les niveaux de difficulté, afin de garantir une expérience de jeu équilibrée.
2. Algorithme d'Aide au Joueur : Il a été discuté de la mise en place d'un algorithme d'assistance pour aider les joueurs à progresser dans le jeu.
3. Interaction avec l'Utilisateur : Madame Bouhlla a souligné l'importance de bien gérer l'interaction avec l'utilisateur pour offrir une expérience utilisateur optimale.

description littérale de chaque fonction

1. void PrintSodoku(GameInfo gameInfo): Cette fonction affiche une grille de Sudoku ainsi que les informations du joueur (nom, âge, niveau et partie en cours).

2. bool IsSodoku(int SodokuMatrice[SIZE][SIZE], int ligne, int colonne, int element): Cette fonction vérifie si l'élément donné peut être inséré dans la grille de Sudoku à la position spécifiée sans violer les règles du Sudoku.

3. bool SodokuSolution(int SodokuMatrice[SIZE][SIZE]): Cette fonction tente de résoudre la grille de Sudoku en utilisant une approche récursive.

4. void GenerateDagonaleBlocks(int SodokuMatrice[SIZE][SIZE]): Cette fonction génère des blocs diagonaux pour initialiser une grille de Sudoku.

5. void grillenonresolu(int SodokuMatrice[SIZE][SIZE], LEVEL difficulty): Cette fonction génère une grille de Sudoku non résolue en fonction du niveau de difficulté spécifié.

6. bool solution(GameInfo jeu): Cette fonction vérifie si une partie de jeu est résolue en examinant la grille de Sudoku.

1. void tovide(int SodokuMatrice[SIZE][SIZE]): Cette fonction initialise une matrice de Sudoku avec des valeurs vides.

2. void printMatrix(int matrice[SIZE][SIZE]): Cette fonction affiche une matrice.

3. int is\_regular\_file(const char \*path): Une fonction qui vérifie si un chemin spécifié pointe vers un fichier régulier.

4. void printpart(): Cette fonction répertorie les fichiers dans un répertoire "sudoku\_doc" s'ils sont des fichiers réguliers.

5. char\* create(char \*folderName): Cette fonction crée un dossier portant le nom spécifié s'il n'existe pas déjà.

6. void saveGameInfo(const char \*folderPath, const GameInfo \*gameInfo): Cette fonction enregistre les informations de jeu dans un fichier texte dans le dossier spécifié.

7. int loadGameInfo(const char \*folderPath, const char \*gameName, GameInfo \*gameInfo): Cette fonction charge les informations de jeu à partir d'un fichier texte dans le dossier spécifié.

8. void game(GameInfo jeu): Cette fonction gère le déroulement d'une partie de Sudoku. Les joueurs peuvent entrer des valeurs et jouer au Sudoku. La partie est sauvegardée périodiquement.

9. void run(int sodokumatrice[SIZE][SIZE], char \*folderName): Cette fonction gère le lancement du jeu, permettant aux joueurs de créer une nouvelle partie ou de continuer une partie existante.

10. int main(): La fonction principale du programme qui initialise le générateur de nombres aléatoires, crée un répertoire pour les sauvegardes de jeu et lance le jeu.

Chacune de ces fonctions a un rôle spécifique dans le fonctionnement du jeu de Sudoku ou dans la gestion des fichiers et des répertoires associés.

Certaines améliorations et fonctionnalités supplémentaires

1. Différents Niveaux de Difficulté :

- Au lieu de simplement choisir un niveau de difficulté (facile, moyen, difficile), nous pouvons permettre aux joueurs de personnaliser la difficulté en ajustant des paramètres tels que le nombre initial de cases remplies.

2. Temps de Jeu:

- Suivre le temps que prend un joueur pour résoudre le puzzle de Sudoku et afficher le temps au fur et à mesure. Cela ajoute un élément de compétition et de défi.

3. Score et Classements :

- Implémenter un système de notation basé sur le temps ou d'autres critères.

4. Conseils et Indices:

- Ajouter un système de conseil ou d'indices pour aider les joueurs s'ils sont bloqués, avec une pénalité sur leur score.

5. Sauvegarde Automatique :

- Mettre en place une sauvegarde automatique régulière de la partie en cours, de sorte que les joueurs puissent reprendre là où ils se sont arrêtés en cas de fermeture du jeu.

6. Langues et Traductions:

- Introduire des options de langues et de traductions pour rendre le jeu accessible à un public plus large.