Rapport IA et Jeux

Achraf BOUCHTITA

achraf.bouchtita@etu.univ-amu.fr

Lina DIANI

lina.diani@etu.univ-amu.fr

Projet encadré par

Stephane AYACHE

stephane.ayache@univ-amu.fr

Table des matières

Remerciements

Nous souhaitons remercier notre professeur encadrant, Monsieur Stéphane Ayache, pour son soutien durant ce projet. Ses explications claires ont grandement contribué à notre compréhension globale ainsi qu'aux détails spécifiques du projet. Sa disponibilité pour résoudre nos difficultés lors de l'implémentation des fonctions a été très précieuse, nous permettant de progresser efficacement face aux défis rencontrés.

Chapitre 1

Introduction

I Contexte et Objectif

Le projet Puissance4 a été réalisé dans le cadre du module IA&JEUX . Ce jeu, un classique des jeux de société, se joue à deux joueurs sur une grille de 6 lignes et 7 colonnes, avec pour objectif d'aligner quatre jetons de sa couleur en ligne, colonne ou diagonale. Le but principal de ce projet était de développer une version numérique du jeu en utilisant le langage C, avec deux interfaces : une console et une interface graphique basée sur SDL2. Un autre objectif clé était l'implémentation d'une intelligence artificielle (IA) permettant à un joueur humain de jouer contre une IA, offrant ainsi une expérience de jeu enrichie et diversifiée.

II Description du Jeu

Règles du Jeu

Le jeu se joue à deux joueurs. Chaque joueur à tour de rôle insère un jeton dans une colonne non pleine. Le premier joueur à aligner quatre de ses jetons horizontalement, verticalement ou diagonalement remporte la partie. Si la grille est complètement remplie sans qu'aucun joueur n'aligne quatre jetons, la partie est déclarée nulle.

Fonctionnalités implémentées

— Jeu en mode console : Permet aux joueurs de jouer via la console.
— Jeu en mode graphique : Utilise SDL2 pour offrir une interface graphique agréable.
— Mode deux joueurs : Deux joueurs humains peuvent jouer l'un contre l'autre.
— Mode joueur contre IA : Un joueur humain peut jouer contre une IA avec deux niveaux de difficulté (DFS et Minimax).

Chapitre 2

Conception et Architecture Générale

Le projet Puissance 4 est structuré de manière modulaire, répartissant les responsabilités entre plusieurs fichiers source et d'en-tête, ce qui permet une gestion claire et organisée du code. Chaque fichier a un rôle bien défini dans la construction du jeu, facilitant ainsi la maintenance et l'évolution du projet.

- **board.c et board.h** : Ces fichiers gèrent la grille de jeu. La gestion inclut l'initialisation de la grille, l'affichage de son état actuel, et la mise à jour de la grille après chaque coup joué par un joueur ou l'IA. Les fonctions essentielles comme initGame(), initBoard(), getChildBoard(), et printBoard() se trouvent ici.

- **list.c et list.h** : Ces fichiers sont responsables de la gestion des structures de données nécessaires, principalement des listes chaînées. Ces listes sont utilisées pour suivre les états successifs du jeu, facilitant la mise en œuvre de l'algorithme Minimax pour l'IA. Les fonctions telles que nodeAlloc(), freeItem(), initList(), popFirst(), et popBest() permettent la gestion efficace des nœuds de la liste.

- **puissance4.c et puissance4.h** : Ces fichiers contiennent la logique principale du jeu ainsi que l'implémentation de l'intelligence artificielle. Cela inclut l'insertion de jetons, la vérification des conditions de victoire, et l'évaluation de l'état de la grille. Les algorithmes d'IA, y compris Minimax avec élagage alpha-bêta et une IA de sélection aléatoire, sont également implémentés ici. Les fonctions clés incluent insertToken(), checkWin(), evaluateBoardState(), minimax(), et getBestMove().

- **mainconsole.c** : Ce fichier sert de point d'entrée pour le jeu en mode console. Il contient la fonction main() qui initialise le jeu, gère les entrées des joueurs, et appelle les fonctions appropriées pour mettre à jour et afficher la grille. Il offre également une interface textuelle pour choisir le mode de jeu (contre un autre joueur ou contre l'IA).

- **gui.c et gui.h** : Ces fichiers gèrent l'interface graphique du jeu en utilisant SDL2. Ils incluent les fonctions nécessaires pour initialiser SDL, rendre la grille de jeu, afficher les jetons, et gérer les événements utilisateur (comme les clics de souris). Les fonctions telles que initSDL(), drawGrid(), renderText(), et showWinnerMessage() sont implémentées ici pour créer une expérience utilisateur graphique fluide et interactive.

- **maingui.c** : Ce fichier sert de point d'entrée pour le jeu en mode graphique. Il contient la fonction main() pour le mode graphique, qui initialise SDL, affiche le menu principal, et gère les boucles de jeu. Il appelle les fonctions de gui.c pour rendre l'interface graphique et gérer les interactions utilisateur.

Chapitre 3

Algorithme de l'IA

Dans ce projet Puissance4, deux algorithmes d'intelligence artificielle ont été implémentés pour offrir des niveaux de défi variés aux joueurs : l'algorithme de Recherche en Profondeur (DFS) et l'algorithme Minimax avec élagage alpha-bêta. Chacun de ces algorithmes a ses propres caractéristiques, avantages et inconvénients.

I Algorithme de Recherche en Profondeur (DFS)

L'algorithme de recherche en profondeur (DFS) explore de manière exhaustive toutes les possibilités de jeu jusqu'à une certaine profondeur pour prendre des décisions. À chaque étape, l'algorithme choisit une colonne et insère un jeton, puis évalue l'état du jeu en continuant à explorer les mouvements suivants jusqu'à atteindre une profondeur maximale ou un état terminal (victoire, défaite ou match nul).

II Algorithme Minimax avec Élagage Alpha-Bêta

L'algorithme Minimax est une méthode de recherche utilisée pour trouver le meilleur coup possible en évaluant les positions de jeu jusqu'à une certaine profondeur. Il utilise un score pour chaque position possible et choisit le mouvement qui maximise les chances de victoire tout en minimisant les chances de défaite. L'algorithme commence par évaluer l'état actuel du jeu. Ensuite, il explore tous les coups possibles jusqu'à une profondeur donnée, calculant un

score pour chaque mouvement. L'élagage alpha-bêta est utilisé pour améliorer l'efficacité de l'algorithme en élaguant les branches de l'arbre de décision qui ne peuvent pas influencer le résultat final. Enfin, l'algorithme sélectionne le coup qui a le meilleur score évalué.

III Comparaison des Algorithmes

Caractéristique	**DFS (Recherche en Profondeur)**	**Minimax avec Élagage Alpha-Bêta**
Exploration	Explore toutes les possibilités jusqu'à une certaine profondeur	Explore toutes les possibilités jusqu'à une certaine profondeur, avec élagage des branches inutiles
Avantages	Simple à implémenter, exploration exhaustive	Très stratégique, élagage pour une meilleure efficacité
Inconvénients	Lent et gourmand en ressources pour des profondeurs élevées	Complexe à implémenter, nécessite plus de ressources de calcul
Performance	Moins performant pour des arbres de décision larges	Plus performant grâce à l'élagage alpha-bêta
Complexité	Complexité temporelle élevée pour des profondeurs élevées	Complexité réduite grâce à l'élagage alpha-bêta
Utilisation	Adapté pour des profondeurs de recherche limitées	Adapté pour des jeux stratégiques avec arbres de décision larges

TABLE 3.1 – Comparaison des algorithmes DFS et Minimax avec élagage alpha-bêta

Chapitre 4

L'interface Graphique

L'interface graphique du projet Puissance 4 vise à améliorer l'expérience utilisateur en offrant une représentation visuelle et interactive du jeu. Contrairement à une interface console, une interface graphique permet aux utilisateurs de voir le plateau de jeu, les jetons, et d'interagir avec le jeu de manière intuitive en utilisant la souris. Pour ce projet, les bibliothèques `SDL` et `SDL_ttf` ont été utilisées. `SDL` gère les opérations de rendu graphique et les événements, tandis que `SDL_ttf` permet le rendu du texte.

I Initialisation de l'Interface

- **Initialisation de SDL** : `SDL_Init(SDL_INIT_VIDEO)` initialise les sous-systèmes vidéo de SDL. En cas d'échec, un message d'erreur est affiché et le programme est arrêté.
- **Création de la fenêtre** : `SDL_CreateWindow` crée une fenêtre de dimensions définies (700x600) où le jeu sera affiché. Si la création échoue, le programme affiche une erreur et se termine.
- **Création du rendu** : `SDL_CreateRenderer` crée un contexte de rendu accéléré pour la fenêtre, permettant de dessiner sur celle-ci. En cas d'échec, le programme affiche une erreur et se termine.
- **Initialisation de SDL_ttf** : `TTF_Init` initialise la bibliothèque `SDL_ttf`. Si l'initialisation échoue, un message d'erreur est affiché et le programme est arrêté.
- **Chargement de la police** : `TTF_OpenFont` charge une police de caractères (`arial.ttf`) utilisée pour le texte. En cas d'échec, un message d'erreur est affiché et le programme se termine.

II Rendu Graphique

— **Nettoyage de l'écran** : `SDL_SetRenderDrawColor` et `SDL_RenderClear` définissent la couleur de fond (blanc) et nettoient l'écran.
— **Dessin de la grille** : Des lignes noires sont dessinées pour représenter la grille de 6 lignes et 7 colonnes.
— **Dessin des jetons** : Les positions des jetons 'X' et 'O' sur le plateau sont parcourues, et chaque jeton est dessiné à sa position respective. Les jetons rouges représentent les 'X' et les jetons jaunes représentent les 'O'. La fonction `SDL_RenderFillCircle` est utilisée pour dessiner des cercles pleins représentant les jetons.

III Gestion des Événements

La capture et le traitement des événements utilisateur sont essentiels pour rendre l'interface interactive. Voici comment cela est géré :

— **Événements de fermeture** : Les événements de type `SDL_QUIT` permettent de fermer l'application proprement lorsque l'utilisateur clique sur le bouton de fermeture de la fenêtre.
— **Événements de clic de souris** : Les événements de type `SDL_MOUSEBUTTONDOWN` détectent les clics de souris et permettent de déterminer la colonne où le joueur souhaite placer son jeton.

IV Boucles de Jeu

Les boucles de jeu `gameLoopPvP` et `gameLoopPvAI` gèrent le déroulement du jeu en mode joueur contre joueur et joueur contre IA respectivement :

— **`gameLoopPvP`** : Cette fonction gère le déroulement d'une partie entre deux joueurs humains. Les clics de souris sont capturés pour insérer des jetons dans les colonnes correspondantes, et les conditions de victoire ou de match nul sont vérifiées après chaque mouvement.
— **`gameLoopPvAI`** : Cette fonction gère le déroulement d'une partie entre un joueur humain et une IA. Les mouvements de l'IA sont déterminés par l'algorithme choisi (DFS ou Minimax), et les conditions de victoire ou de match nul sont vérifiées après chaque mouvement.

V Affichage des Messages de Victoire

— **Affichage du message de victoire** : La fonction `renderText` affiche le message de victoire ou de match nul au centre de l'écran.

— **Affichage du bouton de recommencement** : Un bouton "Recommencer" est dessiné à l'écran pour permettre à l'utilisateur de relancer une partie en cliquant dessus.

Chapitre 5

Gestion du projet

I Gestion de Projet

La réalisation du projet Puissance 4 a nécessité une organisation rigoureuse et une gestion efficace des différentes phases de développement. Nous avons adopté une méthodologie de travail collaborative, avec des réunions régulières et une répartition claire des tâches. Voici un aperçu des principales étapes et de la gestion de projet :

I.1 Planification et Réunions

Nous avons établi un plan de travail détaillé avec des réunions hebdomadaires sur Discord pour suivre l'avancement du projet et résoudre les problèmes. En cas de blocages techniques, des rencontres en présentiel ont été organisées. La collaboration continue sur Discord a assuré une communication fluide, tandis que les sessions de debugging en présentiel ont permis de résoudre rapidement les problèmes. Le code était partagé et géré sur GitHub, facilitant ainsi la collaboration et le suivi des modifications.

I.2 Répartition des Tâches

— **Achraf BOUCHTITA** : Responsable de la réadaptation des structures de données, Achraf a travaillé sur l'optimisation et la gestion des données du jeu. Il a également contribué de manière significative à l'implémentation de l'intelligence artificielle (IA), en particulier dans le développement et l'optimisation des algorithmes.

— **Lina DIANI** : Chargée de l'interface graphique, Lina a développé l'interface utilisateur en utilisant SDL2, rendant le jeu interactif. Elle a aussi participé à l'implémentation de l'IA, en intégrant les algorithmes développés par Achraf et en s'assurant de leur bon fonctionnement dans l'interface graphique.

Malgré la répartition des tâches définies, nous avons régulièrement pratiqué le pair programming et collaboré sur les parties respectives du projet de chacun. Cette approche nous a permis d'apprendre les uns des autres, de corriger nos erreurs et de renforcer nos compétences collectives tout en maintenant la qualité du travail individuel.

Chapitre 6

Conclusion

Le projet Puissance 4, développé dans le cadre du module IA&JEUX, a permis d'approfondir divers aspects du développement logiciel, incluant la conception algorithmique, la gestion des structures de données et la création d'interfaces utilisateur graphiques. En réalisant ce jeu classique, nous avons combiné des compétences en programmation avancées en C et en développement d'interfaces graphiques.

Le développement s'est articulé autour de plusieurs modules clés : la gestion du plateau de jeu, les structures de données, la logique du jeu et l'IA, ainsi que les interfaces utilisateur en mode console et graphique. Chaque module a été conçu pour garantir une expérience de jeu fluide et intuitive. Les algorithmes implémentés pour l'insertion des jetons, la vérification des conditions de victoire et la gestion du plateau assurent une fonctionnalité robuste et efficace.

L'intelligence artificielle a été un volet majeur du projet, avec l'implémentation de deux algorithmes différents : Minimax avec élagage alpha-bêta et une recherche en profondeur (DFS). Ces algorithmes ont été choisis pour leurs approches distinctes dans la prise de décision, offrant ainsi différents niveaux de défi pour les joueurs. Une comparaison entre ces algorithmes a permis de mettre en lumière leurs avantages et inconvénients respectifs, enrichissant notre compréhension de l'IA dans les jeux de stratégie.

L'interface graphique, développée avec SDL2 et SDL_ttf, offre une expérience utilisateur immersive et engageante. Les éléments graphiques, tels que le plateau de jeu et les jetons, ont été conçus pour assurer une interaction fluide et intuitive.

En somme, le projet Puissance 4 a été une occasion précieuse de mettre

en pratique des compétences en programmation, en conception d'algorithmes et en développement d'interfaces utilisateur. Nous avons réussi à créer un jeu complet et fonctionnel qui allie esthétique et performance.

Chapitre 7

GIT

Vous pouvez trouver notre projet sur **GIT**. Pour un accès facile, voici le QR code correspondant :

FIGURE 7.1 – Cliquez sur l'image pour accéder au dépôt Git