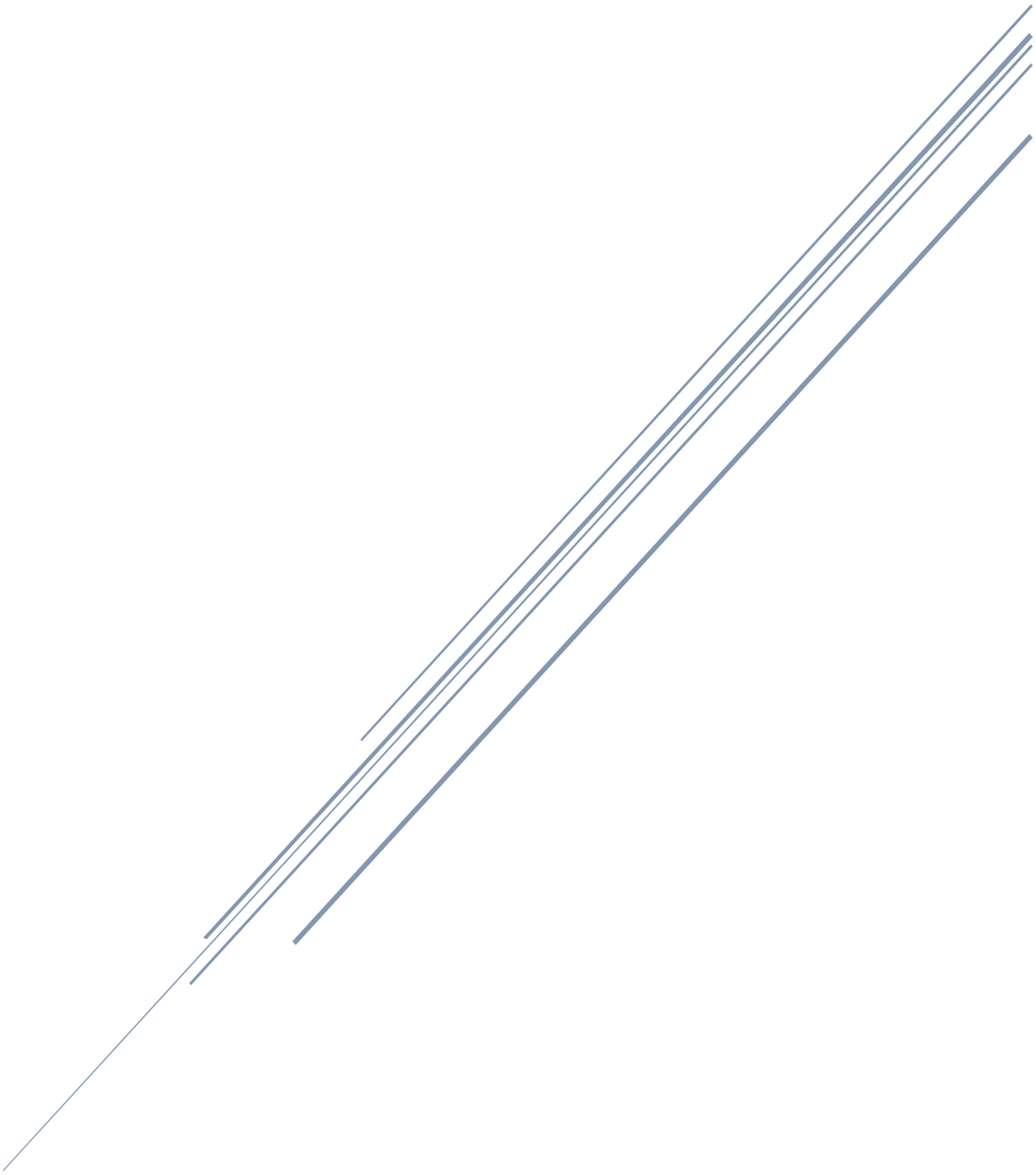
RAPPORT DE PROJET

PUISSANCE 4

Par CHARFI Mokhtar DAOUDI Ismail DJEDIDI Rime et ZIRA Mecit



INSTITUT GALILEE
UE PROJET

Présentation

Dans le cadre de l'UE Projet, nous avons choisi comme sujet le jeu du Puissance 4 à implémenter en langage Java. Nous avons effectué ce projet à 4 : CHARFI Mokhtar, DAOUDI Ismail, DJEDIDI Rime et ZIRA Mecit. Nous nous sommes très vite mis d'accord sur la répartition des tâches Mokhtar et Rime pour la partie « logique » (Règle du jeu et IA) et Ismail et Mecit pour l'interface graphique (statique et dynamique).

Pour mettre en place l'interface graphique, l'idée de départ était d'utiliser la bibliothèque graphique Swing et le projet a été entamé sur cette idée. Puis nous nous sommes rendu compte que la bibliothèque JavaFx serait plus efficace donc l'idée de changer de bibliothèque a fait débat. Néanmoins, nous sommes restés sur Java Swing par manque de temps.

Problèmes rencontrés et solutions apportées

L'un des premiers problèmes que nous avons rencontrés était de mêler l'affichage du tableau graphique et le traitement des règles du jeu. Pour cela, nous avons eu l'idée de dissocier le plateau de jeu « logique » et le plateau d'affichage. Notamment, en implémentant une nouvelle classe Case pour pouvoir manipuler le plateau facilement.

Puis, s'en est suivi plusieurs problèmes avec le bot. L'idée d'utiliser un algorithme minimax, a été évoqué. Cependant, nous ne savions pas comment le mettre en place. Donc, dans un premier temps, nous avons établis un bot qui traite les coups avec seulement une profondeur de 2. Par la suite, nous avons eu une réunion avec M. Breuvart qui nous a beaucoup orienté : l'idée était d'étudier chaque coup possible et les coups possibles suivants récursivement jusqu'à une profondeur fixée au préalable en ne retenant à chaque fois que les situations les plus favorables. Le nombre de situations étudiés étant exponentiel le temp de calcul deviendrait vite trop long, il faudrait alors stocker les scores des situations déjà calculés pour ne pas avoir à les recalculer par la suite. Ainsi l'algorithme minimax était plus clair mais quelques problèmes persistaient : comment déterminer une méthode de calcul du score du plateau entier alors que notre fonction de vérification ne s'applique que sur un jeton ? Tout simplement, nous avons choisi de donner un score arbitraire au plateau selon les alignements possibles (si une case vide a le potentiel d'aligner 4 jetons c'est +-50, pour un alignement de 3 c'est +-10 etc…). Cette méthode reste à améliorer, car il y a certainement une manière plus juste et plus complète de déterminer le score du plateau. Une fois la fonction de score et l'algorithme minimax implémentés, nous pouvions aller jusqu'à une profondeur de recherche de 5. Nous voulions augmenter encore cette profondeur et après des recherches sur internet nous avons appris l'existence d'une amélioration à apporter à l'algorithme minimax pour encore améliorer son temp de calcul : l'élagage alpha beta. Cette méthode consiste à éliminer des situations dont on sait qu'elles ne seront pas utiles. Nous avons donc adapté notre algorithme pour pouvoir y intégrer cette méthode de traitement,

ce qui nous a permis d'avoir un programme plus performant, pouvant aller jusqu'à une profondeur de recherche de 10 en gardant un temps de calcul convenable.

Outre les problèmes liés au jeu, nous avons dû faire face à des problèmes dans l'interface graphique. Pour respecter le classique Puissance 4 nous souhaitions faire des boutons ronds, toutefois, faire arrondir les boutons n'est pas une mince affaire sur Java Swing. C'est pourquoi, nous avons gardé la forme des boutons et les avons remplacés par des images qui donnent l'air de jouer avec des jetons.

Un autre problème lié à l'interface graphique est le fait que pour arrêter le jeu suite à une victoire, nous avons rendus les boutons non cliquables. Seulement, cette manœuvre induit un changement de couleur des boutons. Nous avons décidé d'intégrer ce problème à l'interface graphique ! Maintenant, lorsqu'il y a une victoire seuls les boutons qui attribue la victoire restent visibles.

Rapport individuel

Mokhtar

La tâche qui m'était attribuée était la partie « logique » du projet. Nous avons rencontré un certain nombre de difficultés que nous avons réussi à tous surmonter grâce à une bonne communication et un bon travail de recherche. Personnellement je me suis beaucoup documenté sur internet pour découvrir les différentes stratégies pour créer une IA. J'ai aussi du renoué avec java que je n'avais pas utilisé depuis le premier semestre et ait appris à utiliser certaines fonctionnalités de ce langage. J'ai trouvé la réalisation de ce projet enrichissante mais j'aurais aimé avoir plus de temps pour le perfectionner. Nous avons écrit le code au fur et à mesure, on a donc dû à plusieurs reprises revenir sur notre code pour l'adapter aux difficultés qu'on rencontrait. Si c'était à refaire je reprendrais complètement la structure du projet pour avoir code plus propre et plus optimisé.

Ismail

J'ai bien aimé le travail en équipe c'était la première fois que je le fais cette année. Moi j'ai fait l'interface graphique avec swing, c'était ma première expérience avec une interface graphique dans tous les langages informatiques que j'ai utilisés. J'ai eu une difficulté de pouvoir faire des buttons ronds sur java swing du coup j'ai choisi de créer des images et les mettre à chaque button.

Si j'aurai l'opportunité de faire un projet en java avec une interface graphique je vais choisir JavaFX ou une autre bibliothèque graphique plus moderne au lieu de Swing parce que avec swing on peut juste faire des interfaces modestes.

Rime

Je suis plutôt satisfaite de notre travail, et si c'était à refaire je m'y serais prise plus tôt pour optimiser davantage notre jeu mais aussi, comme l'a proposé M.Breuvart, ajouter un programme client-serveur et permettre à des joueurs de jouer à distance.

Mecit