

Rapport 2 : Projet puissance 4

1. Evolution du projet

Dans le cadre de ce projet, nous développons un programme en Python pour tester les algorithmes Minimax, Alpha-Beta et MCTS sur le jeu de Puissance 4. Nous avons débuté par l'apprentissage des bases de développement en Python, l'analyse des algorithmes et la détermination de leur faisabilité pour le jeu de Puissance 4. Ensuite, nous avons codé deux des trois algorithmes et les avons testés individuellement. Nous avons également conçu une interface graphique pour tester les algorithmes. Nous avons utilisé la bibliothèque Pygame pour l'implémentation de l'interface graphique.

Nous avons réussi à travailler en équipe pour le développement de ce projet. La répartition des tâches a été équitable et nous avons pu respecter la plupart des délais.

2. Comparaison avec les objectifs fixés

Les objectifs initiaux de ce projet étaient les suivants :

a) Apprentissage de quelques bases de développement en Python , analyse des algorithmes minimax, alpha-beta et MCTS . Détermination de la faisabilité des algorithmes pour le jeu Puissance 4.

Nous avons réussi à atteindre cet objectif en apprenant, pour certains d'entre nous, les bases de développement en Python et en analysant les trois algorithmes. Nous avons pensé que ces trois algorithmes étaient réalisables pour le jeu Puissance 4 .

b) Conception d'une interface pour tester les algorithmes.

Nous avons atteint cet objectif en concevant une interface graphique pour le jeu de Puissance 4 à l'aide de la bibliothèque Pygame.

c) Codage des 3 algorithmes.

Nous avons réussi à coder deux des trois algorithmes demandés : Minimax et alphaBeta. Cela nous a permis de comprendre le fonctionnement de ces algorithmes et de les comparer.

Malheureusement, nous n'avons pas eu suffisamment de temps pour coder l'algorithme MCTS.

d) Test et comparaison des algorithmes .

Nous avons testé les algorithmes Minimax et alphaBeta individuellement et nous avons également joué contre elle en utilisant l'interface graphique que nous avons conçue.

Nous avons comparé les performances de Minimax et alphaBeta en jouant contre elle. Nous avons chronométré le temps de réflexion de chaque algorithme et avons constaté que l'algorithme alphaBeta était plus rapide que Minimax dans la plupart des cas.

Malgré le fait que nous n'avons pas réussi à programmer MCTS et que la comparaison des algorithmes a été faite uniquement en jouant contre les IAs et en chronométrant leur temps de réflexion, nous sommes en partie satisfaits des résultats que nous avons obtenus.

3. Premiers résultats

Nous avons testé Minimax et Alpha-Beta sur un plateau 6 par 7 et avons obtenu des résultats satisfaisants. Les deux algorithmes ont été capables de jouer au jeu de Puissance 4 de manière efficace. Cependant nous n'avons pas encore mis de grands moyens en place pour la comparaison des algorithmes. On s'est donc contenté de les tester en jouant contre eux et de les chronométrer.

Nous avons testé les algorithmes avec plusieurs valeurs de profondeur, notamment pour minimax mais nous n'avons pas été en mesure de choisir un paramètre optimal.

Malgré cela, nous avons tout de même pu comparer les performances des deux algorithmes. Nous avons constaté que l'algorithme Alpha-Beta était plus efficace que Minimax pour le jeu de Puissance 4. Il a été capable d'explorer plus d'arbres de jeu en moins de temps que l'algorithme Minimax.

4. Conclusion

En conclusion, nous sommes satisfaits des résultats obtenus jusqu'à présent pour ce projet. Bien que nous n'ayons pas pu finaliser l'implémentation de l'algorithme MCTS, nous avons réussi à atteindre une bonne partie des objectifs fixés et avons obtenu des bons résultats pour les algorithmes Minimax et Alpha-Beta.