

Implémentation du jeu Hex via la Programmation Orientée Objet et stratégies de décision en C++

1st Jair Anderson Vasquez Torres
Ingénieur Degree Programme STIC
ENSTA Paris
Paris, France
jair-anderson.vasquez@ensta.fr

2nd Santiago Florido Gomez
Ingénieur Degree Programme STIC
ENSTA Paris
Paris, France
santiago.florido@ensta-paris.fr

Abstract—Ce projet implémente en C++ un système complet pour le jeu Hex, conçu principalement comme un exercice de Programmation Orientée Objet. La solution organise le domaine du jeu au moyen de classes qui encapsulent le plateau, l'état, les coordonnées et les règles, permettant la génération de coups légaux et la vérification de victoire au moyen d'un parcours BFS sur les connexions des pions. Sur cette base s'intègrent des stratégies de décision (p. ex., Negamax avec hachage et table de transposition) et une évaluation de positions découplée du moteur, qui peut être heuristique ou s'appuyer sur un modèle neuronal de valeur intégré à l'exécutable (exportable en TorchScript), sans lier la conception à une architecture spécifique. De plus, une interface graphique (GUI) en SFML a été développée pour faciliter l'interaction, la visualisation du plateau et les tests du comportement des agents.

Index Terms—Hex game, C++, Object-Oriented Programming, Negamax, Transposition Table, TorchScript, SFML.

I. INTRODUCTION

Dans son espèce, Hex est un jeu de stratégie abstraite pensé pour deux joueurs, c'est aussi un jeu d'information parfaite ce qui implique qu'à tout moment les joueurs connaissent toute l'information du plateau et il n'y a pas d'information cachée pour aucun des deux joueurs et sans hasard qui se joue sur un plateau avec des cellules hexagonales généralement en forme de losange mais dans la version modifiée réalisée avec des cellules de décalage intercalé. Chaque joueur a une paire de côtés opposés du plateau assignés comme objectif, chaque joueur doit à son tour correspondant placer une pièce de sa couleur dans une cellule vide à l'intérieur du plateau.

L'objectif principal du jeu consiste à ce que chaque joueur parvienne à la connexion d'une chaîne connectée de pièces de la même couleur avant que cette connexion soit réalisée par son adversaire le premier à réaliser la connexion gagne la partie, étant donnée la constitution du jeu celui-ci ne peut pas se terminer en match nul il doit toujours exister un joueur gagnant, dans le présent projet on propose un plateau de base de 7 x 7, qui peut être modifié par les joueurs pour avoir un plateau de plus grande dimension.

Dans la configuration proposée, on propose la possibilité de s'affronter comme utilisateur de l'application à un autre utilisateur qui se trouve sur le même dispositif, comme cela fonctionne de manière traditionnelle, le jeu, ou bien de s'affronter à un adversaire virtuel qui peut être un modèle

heuristique qui a été défini en utilisant un moteur de règles de base qui expose des opérations et des coups légaux et de vérification de victoire sur lequel se monte un agent Negamax qui utilise une évaluation heuristique en ressemblant au fonctionnement d'un capteur de qualité des mouvements du joueur au tour dans le nœud dans lequel se trouve donnée l'évaluation ou un modèle impulsé par l'implémentation d'un modèle qui utilise la même logique de Negamax, mais qui au lieu d'utiliser une expression heuristique pour déterminer une valeur pour chaque mouvement possible utilise une MLP, qui a été obtenue au moyen de reinforcement learning avec des données de parties en direct entre le modèle et des versions antérieures ou l'heuristique.

On présentera dans ce projet des éléments de sa configuration de base en termes de structure et de fonctionnement, ainsi que la manière dont ont été appliqués les concepts de la programmation orientée objet pour l'obtention du produit final présenté.

REFERENCES

- [1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, April 1955.