

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.

Theta, la réponse de Harris, est définie pixel par pixel : plus *Theta* est grande, plus le pixel est un *coin*.

I. DÉTECTEURS

A. *Q4*

La fonction d'intérêt de Harris se base sur le fait d'assigner une valeur à chaque pixel. Cette valeur mesure à quel point ce pixel est une *intersection*, c'est-à-dire un *coin*.

Dire qu'un pixel est un coin signifie qu'on cherche à quel point il ressemble à l'intersection de deux bords. Autrement dit, à quel point ce pixel correspond à un endroit où l'image change fortement dans deux directions perpendiculaires. C'est utile, parce qu'avec une petite fenêtre de pixels (la fenêtre W), on peut estimer à quel point ce qu'on observe change au niveau de la structure.

Intuitivement, si on déplace légèrement la fenêtre W autour d'un pixel, on observe :

- **Zone plane (sans texture)** : la fenêtre change très peu → ce n'est pas un coin.
- **Bord** : la fenêtre change peu en se déplaçant *le long* du bord, mais change beaucoup en le *traversant*.
- **Coin** : la fenêtre change beaucoup dans presque n'importe quelle direction.

Ce que fait Harris, c'est utiliser les gradients I_x et I_y pour savoir à quel point l'intensité change selon x et selon y . Si autour du pixel il y a un changement fort dans une seule direction, on est plutôt sur un bord ; mais si les changements forts sont dans deux directions, on parle d'un coin. Ainsi,