



Hex-Ta(c)tique

Projet de programmation 2

AUTEURS

AL AYOUBI Ibrahim
BIGEY Raphaël
BONETTI Timothée
LEJEUNE Ivan

ENCADRANT

DA-SILVA Sébastien

12 mai 2024

Table des matières

1	Présentation du sujet	2
2	Technologies utilisées	3
3	Développements Logiciel : Conception, Modélisation, Implémentation	4
4	Algorithmes et Analyse	5
5	Analyse des résultats	6
6	Gestion du Projet	7
7	Bilan et Conclusions	8
8	Bibliographie	9
9	Annexes	10

1 Présentation du sujet

Dans le cadre de notre projet de programmation de L3 Informatique, nous avons choisi de travailler sur la résolution de jeux de stratégie combinatoire. Nous avons choisi de nous intéresser à deux jeux de stratégie combinatoire en particulier : le jeu du hex et le jeu de l'Awalé.

Ce projet est motivé par notre intérêt pour la théorie des jeux, et par notre désir de comprendre les mécanismes de résolution de jeux de stratégie combinatoire. L'étude de ces jeux nous permettra de mieux comprendre les algorithmes de recherche et d'optimisation, et de nous familiariser avec les techniques de programmation avancée. A l'avenir, ces résultats pourront être utilisés dans un cadre plus général pour la résolution de problèmes complexes, par exemple pour les échecs ou le go.

Il existe plusieurs approches possibles pour la résolution de jeux de stratégie combinatoire. Parmi les approches les plus courantes, on trouve les algorithmes de recherche en profondeur, les algorithmes de recherche de chemin, les algorithmes de recherche de meilleure réponse, les algorithmes de Monte-Carlo, les algorithmes de renforcement, etc. Pour ce projet, nous avons choisi d'implémenter deux algorithmes en particulier pour la résolution des jeux de hex et d'Awalé : l'algorithme Minimax avec élagage alpha-bêta pour le jeu de hex, et l'algorithme de Dijkstra pour le jeu de l'Awalé.

Les principaux avantages de ces algorithmes sont les suivants :

- L'algorithme Minimax avec élagage alpha-bêta est un algorithme de recherche qui permet de trouver la meilleure stratégie pour un joueur dans un jeu à deux joueurs. Cet algorithme est très efficace pour les jeux de stratégie combinatoire comme le hex, car il permet de réduire le nombre de nœuds explorés lors de la recherche de la meilleure stratégie.
- L'algorithme de Dijkstra est un algorithme de recherche de chemin qui permet de trouver le chemin le plus court entre deux nœuds dans un graphe. Cet algorithme est très efficace pour le jeu de l'Awalé, car il permet de trouver la meilleure stratégie pour un joueur en minimisant le nombre de graines capturées par l'adversaire.

Le cahier des charges détaillé est disponible en annexe. A RAJOUTER : Le cahier des charges détaillé

2 Technologies utilisées

Pour la réalisation de ce projet, nous avons utilisé les langages de programmation suivants :

- Python pour l'implémentation des algorithmes de résolution des jeux et la logique du jeu,
- HTML, CSS et JavaScript pour l'implémentation de l'interface graphique des jeux,
- UML pour la modélisation des classes et des cas d'utilisation,
- Git pour la gestion du code source et le suivi des versions,
- Visual Studio Code pour l'écriture du code et le débogage,
- GitHub pour l'hébergement du code source et la collaboration,
- LaTeX pour la rédaction du rapport,
- Draw.io pour la création des diagrammes UML

Nous avons choisi Python pour l'implémentation des algorithmes de résolution des jeux et la logique du jeu, car c'est un langage de programmation très populaire et très puissant, qui offre de nombreuses bibliothèques et modules pour le développement d'applications complexes. Python est également un langage de programmation très simple et très lisible, ce qui facilite la compréhension du code et la collaboration entre les membres de l'équipe.

Nous avons choisi HTML, CSS et JavaScript pour l'implémentation de l'interface graphique des jeux, car ce sont des langages de programmation efficaces et simples à comprendre, qui permettent de créer des interfaces graphiques interactives et ergonomiques. HTML est un langage de balisage qui permet de structurer les pages web, CSS est un langage de style qui permet de mettre en forme les pages web, et JavaScript est un langage de programmation qui permet de rendre les pages web interactives.

Nous avons choisi UML pour la modélisation des classes et des cas d'utilisation, car c'est un langage de modélisation très complet, qui permet de représenter de façon claire et précise la structure et le comportement des systèmes informatiques.

Nous avons choisi Git et GitHub pour la gestion du code source et le suivi des versions, car ce sont des outils très puissants avec lesquels nous sommes très familiers.

LaTeX a été choisi pour la rédaction du rapport, car c'est un langage de composition de documents très puissant et très flexible, qui permet de créer des documents de grande qualité typographique. Etant donné que nous avons déjà utilisé LaTeX pour d'autres projets, nous avons préféré continuer à l'utiliser pour ce projet.

Draw.io a été choisi pour la création des diagrammes UML, car c'est un outil très simple et qui répond parfaitement à nos besoins. Il permet de créer des diagrammes UML de façon très intuitive, et de les exporter dans différents formats.

3 Développements Logiciel : Conception, Modélisation, Implé- mentation

4 Algorithmes et Analyse

5 Analyse des résultats

6 Gestion du Projet

7 Bilan et Conclusions

8 Bibliographie

9 Annexes