

Licence 3 INFO - UE Projet 2

Cette UE est une UE de programmation collective où une équipe de quatre étudiants et étudiantes doivent choisir un projet à réaliser parmi la liste ci-après. Le langage de programmation n'est pas imposé mais doit être validé par les encadrants de TP. Les projets sont construits autour d'une question scientifique où il s'agit de développer des algorithmes pour les expérimenter sur un problème précis. L'accent est donc mis sur la modélisation, l'algorithmique et l'expérimentation. Des pointeurs vers de la documentation sont fournis pour chaque projet.

L'évaluation de l'unité se fait en deux temps : une *évaluation à mi-parcours* et une *évaluation de fin de parcours*. L'évaluation de mi-parcours est organisée lors de la cinquième séance de TP. Chaque équipe doit rendre la veille un prototype et faire une présentation avec support de 10 minutes devant l'enseignant et les autres équipes. La présentation doit expliquer les objectifs et la problématique du projet, le découpage du projet en sous-tâches, l'état d'avancement et le planning prévisionnel. L'évaluation de fin de parcours consiste en la remise en fin de semestre du code, d'un rapport et d'une présentation du travail réalisé lors d'une soutenance de 15 minutes en présence de deux encadrants de TP. Cette soutenance doit présenter l'entièreté du projet réalisé.

La liste des sujet est donnée ci-dessous et sont classés selon les différents blocs de compétence du master informatique *Algorithmiques et Systèmes Intelligents*, c'est-à-dire avec un coloration Intelligence Artificielle, Science des Données ou Algorithmique & Structures informatique.

Sujet IA : Jeu de Tron multi-joueur et coalitions

Le *jeu de Tron* est une variante multi-joueurs du célèbre jeu du serpent. Sur une grille de taille fixe, chaque joueur contrôle un point qui se déplacer dans les quatres directions. À chaque déplacement, le joueur laisse derrière lui un mur infranchissable. Le but du jeu est d'être le dernier à survivre, sachant qu'une collision avec un mur, l'adversaire ou un bord du plateau entraîne une défaite. La technique classiquement employée pour ce jeu est un algorithme minmaxi / negamax avec élagage alpha-beta où l'heuristique tient compte de la proportion de terrain contrôlé par chaque joueur. L'objectif de ce projet est d'évaluer la performance d'une équipe de joueurs contre une autre en fonction de leur différence de taille, de leur différence en profondeur du raisonnement et de la taille de la grille. Pour ce faire, il s'agira d'implanter une variante multi-joueur du minmax comme *MAXN* où chaque joueur maximise individuellement son heuristique ou *Paranoid* où chaque joueur considère que tous les autres sont contre lui, puis de l'étendre pour représenter des équipes (il est possible de s'inspirer d'un algorithme appelée SOS pour *Socially Oriented Search*).

- Implémentation d'un jeu de Tron (sans joueur humain)
- Implémenter une visualisation du jeu
- Implémentation d'un algorithme MAXN ou Paranoid
- Étendre cet algorithme pour spécifier des équipes en s'inspirant de SOS

- Question scientifique : quelle influence la profondeur de recherche a-t-elle sur le jeu en fonction de la taille des équipes et de la taille de la grille ?
- Ressources : [Negamax avec alphabeta](#), [MAXN et Paranoid](#), [MAXN avec coupes alphabeta](#), [Algorithme SOS](#)

Sujet IA : Jeu de Hex et bandits manchots

Le *jeu de Hex* est un jeu combinatoire abstrait pour deux joueurs qui se joue sur un tablier d'hexagones blancs de taille 14×14 . Les joueurs, Bleu et Rouge, colorient tour à tour un hexagone blanc et l'objectif est de former une ligne de sa couleur qui relie Est et Ouest pour le joueur Bleu, et Nord et Sud pour le joueur Rouge. La combinatoire du jeu étant grande, il est pertinent d'utiliser un algorithme de *recherche arborescente de Monte Carlo* (MCTS) classiquement utilisé au Go pour résoudre le jeu. Cet algorithme, inspiré de stratégies de joueurs sur bandits manchots, va explorer petit à petit l'arbre du jeu en simulant des parties entières, dans la limite d'un budget disponible. Plus le budget est élevé, plus un joueur pourra approximer les meilleurs coups à jouer. Nous pouvons alors nous demander quelle influence le rapport de budget entre les deux joueurs a sur leur probabilité de victoire, sachant que le premier joueur est fortement avantage. Il s'agira donc de modéliser le jeu, implanter l'algorithme et l'expérimenter. Ce projet pourrait être réalisé sur un autre type de jeu comme Reverse Hex, Chameleon ou Quarto..

- Implémenter un jeu de Hex (sans joueur humain)
- Implémenter une visualisation du jeu
- Implémenter un algorithme de recherche arborescente de Monte Carlo (MCTS)
- Implémenter l'optimisation RAVE sur MCTS ou d'implémenter un autre type de jeu (Reverse Hex, Chameleon, Quarto, etc.)
- Question scientifique : quelle influence le rapport de budget entre les deux joueurs a-t-il sur leur probabilité de victoire en fonction de la taille de la grille ?
- Ressources : [Hex Wiki](#), [MTCS](#), [MCTS Rave](#)

Sujet SD : Analyse d'identité d'auteurs et détection de plagiat

La quantité croissante de documents mis en ligne augmente grandement les risques de plagiat. L'objectif de ce projet est de mettre en oeuvre des techniques de traitement de la langue naturelle pour identifier l'origine et/ou la paternité de documents textuels (ou telle ou telle partie de documents) afin de combattre les tentatives de plagiat. Il s'agira d'analyser le contenu textuel de jeux de données comme celui proposé sur le page eCampus du cours, d'implémenter des *mesures de similarité* entre ces contenus et de les regrouper en utilisant des *techniques d'apprentissage automatique*. Des informations de plus haut niveau, comme des entités nommées ou un typage de l'information, pourront être intégrées afin d'obtenir des meilleurs résultats. Il s'agira alors de fournir une étude quantitative des résultats de ces algorithmes, et d'expliquer pourquoi telle ou telle approche est performante (ou non).