

Syllabus projet

Année : 2025-2026

Enseignant(s)

Email(s)

VIDAL Nicolas

nvidal@myges.fr

2026-5A-IABD-DRL

1 Matières, formations et groupes

Matière liée au projet : **M2 - t2 - deep reinforcement learning**

Formations : -

Nombre d'étudiant par groupe : **3** Règles de constitution des groupes: **Imposé**

Charge de travail estimée par étudiant : **30,00 h**

2 Sujet(s) du projet

Type de sujet : **Imposé**

3 Détails du projet

Objectif du projet (à la fin du projet les étudiants sauront réaliser un...)

Evaluer les différentes techniques d'apprentissage par renforcement profond vues en cours sur 1 nouvel environnement parmi la liste ci-dessous (en plus des précédents).

Comprendre les différents atouts de chaque algorithme et savoir quand les appliquer.

Descriptif détaillé

ATTENTION : TOUTE IMPOSSIBILITE POUR LE GROUPE D'ETUDIANT D'EXPLIQUER LE CODE UTILISE POUR LE PROJET CONDUIRA A L'INVALIDATION DU PROJET.

Environnements de départ :

- pour tests : Line World
- pour tests : Grid World
- pour tests : TicTacToe versus Random
- + 1 au choix parmi :
 - Quarto (vs Random ou Heuristique)
> <https://boardgamearena.com/gamepanel?game=quarto>
 - Bobail (vs Random ou Heuristique)
> <https://boardgamearena.com/gamepanel?game=bobail>
 - Pond (versus Random ou Heuristique)
> <https://boardgamearena.com/gamepanel?game=pond>
 - ColorPop (Solo ou Vs Randoms ou Heuristiques)
> <https://boardgamearena.com/gamepanel?game=colorpop>

Types d'agents à étudier :

- Random
- TabularQLearning (quand possible)
- DeepQLearning
- DoubleDeepQLearning
- DoubleDeepQLearningWithExperienceReplay
- DoubleDeepQLearningWithPrioritizedExperienceReplay
- REINFORCE
- REINFORCE with mean baseline
- REINFORCE with Baseline Learned by a Critic
- PPO A2C style
- RandomRollout
- Monte Carlo Tree Search (UCT)
- Expert Apprentice
- Alpha Zero
- MuZero
- MuZero stochastique

Métriques à obtenir (attention métriques pour la policy obtenue, pas pour la policy en mode entraînement) :

- Score moyen (pour chaque agent) au bout de 1000 parties d'entraînement
- Score moyen (pour chaque agent) au bout de 10 000 parties d'entraînement
- Score moyen (pour chaque agent) au bout de 100 000 parties d'entraînement
- Score moyen (pour chaque agent) au bout de 1 000 000 parties d'entraînement (si possible)
- Score moyen (pour chaque agent) au bout de XXX parties d'entraînement (si possible)
- Temps moyen mis pour exécuter un coup

Si la partie est de durée variable :

- Longueur moyenne (nombre de step) d'une partie au bout de 1000 parties d'entraînement
- Longueur moyenne (nombre de step) d'une partie au bout de 10 000 parties d'entraînement
- Longueur moyenne (nombre de step) d'une partie au bout de 100 000 parties d'entraînement
- Longueur moyenne (nombre de step) d'une partie au bout de 1 000 000 parties d'entraînement (si possible)
- Longueur moyenne d'une partie au bout de XXX parties (si possible)

Il sera également nécessaire de présenter une interface graphique permettant de regarder jouer chaque agent et également de mettre à disposition un agent 'humain'.

Pour chaque environnement et chaque algorithme, les étudiants devront étudier les performances de l'algorithme et retranscrire leur résultats.

Les étudiants devront fournir l'intégralité du code leur ayant permis d'obtenir leurs résultats ainsi que les modèles (keras/tensorflow/pytorch/jax/keras_core/burn) entraînés et sauvegardés prêts à être exécutés pour confirmer les résultats présentés.

Les étudiants devront présenter ces résultats dans un rapport où devront figurer l'évolution des métriques d'apprentissage au cours de l'entraînement) ainsi qu'une présentation. Dans ces derniers, les étudiants devront faire valoir leur méthodologie de choix d'hyperparamètres, et proposer leur interprétation des résultats obtenus

Ouvrages de référence (livres, articles, revues, sites web...)

Reinforcement Learning: An Introduction de Richard S. Sutton and Andrew G. Barto

Outils informatiques à installer

tensorflow / keras / pytorch / jax / keras_core / burn

4 Livrables et étapes de suivi

| | | | |
|---|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| 1 | Etape intermédiaire | 1er Gameplay choisi Implémenté (ainsi que les environnements de tests) - Simulation de jeu avec joueur random (calculer le nombre de parties / seconde) - Jeu avec Joueur humain + GUI - Proposition de description de l'état du jeu (vecteur d'encoding) - Proposition de description d'une action du jeu (vecteur d'encoding) Rendu : (tout sur MyGES, pas simplement un lien git) > code source > démonstration rapide > documents de spécifications (encoding vectors) | dimanche 29/03/2026 23h59 |
| 3 | Rendu final | Soutenance Rendu : (tout sur MyGES, pas simplement un lien git) > code source > démonstration rapide > Readme de reproduction des résultats / lancement démo > Rapport conséquent sur les expérimentations menées / résultats obtenus et observations critiques de ces résultats > documents de résultats (metrics sur les gameplays avec tous les algos) > Slides de présentation utilisés pour la soutenance | dimanche 26/04/2026 23h59 |

5 Soutenance

Durée de présentation par groupe : **20 min** Audience : **A huis clos**

Type de présentation : **Présentation / PowerPoint**

Précisions :