



Alain KARAPETIAN, Timeo RAPHOZ, Enzo VIGNE

2025-2026

Tuteur : Julien SOULE

MARIO: Plan de développement

Master Mathématiques Informatiques Appliquées aux
Sciences Humaines et Sociales (MIASHS)

Master 1

Janvier 2026

Table des Matières

1	Plan de développement	3
1.1	<i>Mise en oeuvre</i>	3
1.2	<i>Revue de littérature et montée en compétences</i>	3
1.3	<i>Développement de l'API</i>	3
1.4	<i>Optimisation automatique des hyper-paramètres</i>	4
1.5	<i>Mise en place d'un guide pour les utilisateurs</i>	5
1.6	<i>Mise en place d'une interface CLI / GUI</i>	6

1 Plan de développement

1.1 *Mise en oeuvre*

Dans le cadre du projet, plusieurs tâches doivent être réalisées. Celles-ci correspondent aux fonctionnalités à implémenter afin d'assurer le bon fonctionnement de la plateforme. Ces tâches peuvent être organisées et planifiées de manière itérative et chronologique. Les différentes parties présentées ci-dessous détaillent l'ensemble des tâches ainsi que leurs critères de validation. Il est toutefois important de préciser que les technologies proposées sont susceptibles d'évoluer si d'autres solutions sont jugées plus appropriées. Concernant la répartition du travail, il est pour l'instant prévu que le groupe avance simultanément sur les mêmes tâches, la durée estimée quant à elle est une approximation prévisionnelle et non définitive.

1.2 *Revue de littérature et montée en compétences*

Cette première phase du projet est consacrée à la préparation et à la structuration des bases nécessaires à son bon déroulement. Cette phase préliminaire a pour objectif d'assurer une compréhension commune et approfondie du sujet, des technologies, outils et frameworks associés à travers une revue de littérature approfondie et l'étude des concepts fondamentaux liés au Reinforcement Learning (RL) et au Multi-Agent Reinforcement Learning (MARL). Elle permet notamment d'identifier les approches existantes, les bonnes pratiques, les limites des solutions actuelles et les choix technologiques pertinents pour la conception et le développement de la plateforme.

Durée estimée : 41 jours

Cette étape est indispensable et constitue un prérequis au lancement des phases de conception et d'implémentation du projet.

1.3 *Développement de l'API*

Une API (Interface de Programmation d'Application) est un ensemble de fonctions et de protocoles permettant à différents logiciels ou composants de communiquer entre eux. Elle sert à exposer les fonctionnalités nécessaires au projet de manière standardisée, pour que d'autres programmes puissent les utiliser sans connaître les détails internes de l'application.

Intitulé de la tâche	Description	Critère de Validation	Durée es-timée
Sélection et l'initialisation d'environnements.	L'API doit permettre la sélection et l'initialisation de différents environnements.	Tests unitaires : Initialiser un environnement de jeu OvercookedAI	14 jours
Configuration d'algorithmes MARL/RL	L'API doit permettre de configurer les algorithmes RL/MARL (ex : PPO, MAPPO, etc...).	Tests unitaires : sélectionner un algorithme PPO en passant par les différents paramétrages	20 jours
Gestion intégrée des phases	L'API doit permettre la pré-configuration de l'environnement ainsi que l'entraînement et les tests.	Tests unitaires : pré-configuration de l'environnement OvercookedAI, pré-configuration de l'entraînement avec l'algorithme PPO et des tests	20 jours

Debut : 20/01/2026

Fin : 03/04/2026

Durée estimée : 54 jours

1.4 Optimisation automatique des hyper-paramètres

Les hyper-paramètres désignent les paramètres d'un modèle d'apprentissage permettant d'optimiser la qualité et la rapidité de l'apprentissage. Sélectionner et régler les bonnes valeurs d'un hyper-paramètre sert à adapter l'apprentissage du modèle aux contraintes et exigences du problème à résoudre.

Intitulé de la tâche	Description	Critère de Validation	Durée es-timée
Optimisation de l'efficacité d'apprentissage	Utilisation d'algorithmes pour l'optimisation automatique des hyperparamètres et arrêt des essais non prometteurs (élagage).	Tests unitaires : exécuter l'optimisation des hyperparamètres sur l'algorithme de PPO précédemment configuré.	14 jours
Parallélisation des essais	Possibilité d'exécuter les essais de multiples agents en parallèle sans modification du code source.	Tests unitaires : Configurer plusieurs essais d'entraînement et exécuter plusieurs optimisations simultanées des algorithmes lancés.	9 jours
Visualisation	Outils graphiques pour observer l'évolution et les résultats des optimisations effectuées.	Tests unitaires : Afficher un graphe représentant deux hyperparamètres sélectionnés montrant l'évolution des résultats.	6 jours

Debut : 06/04/2026

Fin : 14/05/2026

Durée estimée : 29 jours

1.5 Mise en place d'un guide pour les utilisateurs

Un tutoriel/guide peut grandement aider un nouvel utilisateur qui souhaiterait utiliser une application qu'il ne connaît pas.

Intitulé de la tâche	Description	Critère de Validation	Durée estimée
Tutoriel écrit pour l'utilisateur (Wiki)	Apporter un guide complet pour un utilisateur ayant des connaissances techniques limitées.	Tests utilisateurs / Feedback : faire remplir une grille de satisfaction (note de 1 à 10 + commentaire optionnel) couvrant chaque étapes du wiki.	3 jours
Complément vidéo	(Optionnel) Vidéo de démonstration pour illustrer la prise en main de l'outil.	Tests utilisateurs / Feedback	3 jours

Debut : 15/05/2026

Fin : 22/05/2026

Durée estimée : 6 jours

1.6 Mise en place d'une interface CLI / GUI

Optionnellement une interface CLI voir GUI facilitant l'utilisation de l'application. Le but d'une interface CLI est d'exécuter l'application via des lignes de commandes (texte), un GUI permet de faire ce qu'un CLI fait, mais grâce à une interface visuelle, rendant l'application plus accessible.

Intitulé de la tâche	Description	Critère de Validation	Durée estimée
Commandes documentées (CLI)	Commandes de base nécessaires au lancement du programme, accompagnées de leur documentation.	Tests unitaires: tester que les commandes réalisent les actions souhaités	2 jours

Intitulé de la tâche	Description	Critère de Validation	Durée es-timée
Gestion de l'environnement	Commandes spécifiques pour créer et sélectionner l'environnement (OvercookedAI, algorithme PPO) pour le système multi-agent.	Tests unitaires: tester que les commandes créent et sélectionnent l'environnement	3 jours
Sélection des algorithmes	Commandes permettant de choisir un ou plusieurs algorithmes à intégrer aux agents.	Tests unitaires: tester que les bons algorithmes sont bien sélectionnés avec les commandes	2 jours
Paramétrage et optimisation	Commandes pour sélectionner, paramétrer et optimiser les hyperparamètres.	Tests unitaires : Validation des sorties de commandes (paramétrage et optimisation réussit).	2 jours
Commande de test et rendu du résultat	Commandes pour lancer les suites de tests et visualiser les rapports d'exécution.	Tests unitaires : tester la visualisation des rapports, qu'ils soient bien générés	3 jours
Interface graphique Web	Interface visuelle permettant d'exécuter les commandes de manière intuitive.	Tests unitaires / fonctionnels : tests permettant de s'assurer que les interactions avec l'interface exécutent les bonnes actions	6 jours

Debut : 25/05/2026**Fin : 17/06/2026****Durée estimée : 18 jours**