



Alain KARAPETIAN, Timeo RAPHOZ, Enzo VIGNE

2025-2026

MARIO: Cahier des charges

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

Master 1

Janvier 2026

Table des Matières

1 RAPPEL DU PROJET	3
1.1 <i>Objectif général</i>	3
1.2 <i>Analyse de l'existant</i>	3
2 DÉFINITION DES TÂCHES ET VALIDATIONS	3
2.1 <i>Revue de littérature et montée en compétences</i>	3
2.2 <i>Développement de l'API</i>	4
2.3 <i>Optimisation automatique des hyper-paramètres</i>	4
2.4 <i>Mise en place d'un guide pour les utilisateurs</i>	5
2.5 <i>Mise en place d'une interface CLI / GUI</i>	6

1 RAPPEL DU PROJET

1.1 *Objectif général*

L'objectif du projet est de rendre l'utilisation de techniques de *Reinforcement Learning (RL)* et *Multi-Agent Reinforcement Learning (MARL)* les plus agnostiques de connaissances expertes que possible. Cette plateforme doit s'appuyer sur des technologies (librairies) existantes afin de permettre au mieux d'automatiser des tâches techniques habituellement réalisées à la main.

À noter que ce projet a pour vocation d'avoir une application locale, ce qui veut dire que son API n'est utilisable que localement pour l'application.

1.2 *Analyse de l'existant*

Critère / Plateformes	Gama	Netlogo	Cromas	Matlab	MARIO
Configuration d'agent (RL)	[]	[]	[]	[x]	[x]
Agnostique aux langages	[]	[x]	[x]	[]	[x]
Documentation	bonne	bonne	moyenne	Moyenne	bonne
Interface de visualisation	[x]	[x]	[x]	[x]	[x]

2 DÉFINITION DES TÂCHES ET VALIDATIONS

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.

2.1 *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 objec-

tif 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.

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

2.2 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	Technologies	Critère de Validation
Sélection et l'initialisation d'environnements.	L'API doit permettre la sélection et l'initialisation de différents environnements.	PettingZoo, MAgent2	Tests unitaires
Configuration d'algorithmes MARL/RL	L'API doit permettre de configurer les algorithmes RL/MARL.	PPO, MAPPO	Tests unitaires
Gestion intégrée des phases	L'API doit permettre la pré-configuration de l'environnement ainsi que l'entraînement et les tests.	PettingZoo, MAgent2	Tests unitaires

2.3 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	Technologies	Critère de Validation
Optimisation de l'efficacité d'apprentissage	Utilisation d'algorithmes pour l'optimisation automatique des hyper-paramètres et arrêt des essais non prometteurs (élagage).	Librairie Optuna	Tests unitaires
Parallélisation des essais	Possibilité d'exécuter les essais de multiples agents en parallèle sans modification du code source.	Librairie Optuna	Tests unitaires
Visualisation	Outils graphiques pour observer l'évolution et les résultats des optimisations effectuées.	Librairie Optuna	Tests unitaires

2.4 **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
Tutoriel écrit pour l'utilisateur	Apporter un guide complet pour un utilisateur ayant des connaissances techniques limitées.	Tests utilisateurs / Feedback
Complément vidéo	(Optionnel) Vidéo de démonstration pour illustrer la prise en main de l'outil.	Tests utilisateurs / Feedback

2.5 ***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	Technologies	Critère de Validation
Commandes documentées (CLI)	Commandes de base nécessaires au lancement du programme, accompagnées de leur documentation.	-	Tests unitaires
Gestion de l'environnement	Commandes spécifiques pour créer et sélectionner l'environnement pour le système multi-agent.	-	Tests unitaires
Sélection des algorithmes	Commandes permettant de choisir un ou plusieurs algorithmes à intégrer aux agents.	-	Tests unitaires
Paramétrage et optimisation	Commandes pour sélectionner, paramétrier et optimiser les hyperparamètres.	-	Tests unitaires
Interface graphique Web	Interface visuelle permettant d'exécuter les commandes de manière intuitive.	Vue.js / React / Angular	Tests unitaires / fonctionnels