

UNIVERSITÉ PIERRE ET MARIE CURIE

PLAN DE DÉVELOPPEMENT

TER DE MAÎTRISE D'INFORMATIQUE EN INTELLIGENCE  
ARTIFICIELLE ET DÉCISION

---

# Semi-supervised Learning Agents

---

*Auteurs :*

Lan ZHOU

Matthieu ZIMMER

*Superviseurs :*

Paolo VIAPPIANI

Paul WENG

4 février 2013

Version 1.0

## Table des matières

<b>1</b>	<b>Aperçu du TER</b>	<b>2</b>
1.1	Objectifs . . . . .	2
1.2	Références . . . . .	2
1.3	Définitions et acronymes . . . . .	2
<b>2</b>	<b>Organisation du TER</b>	<b>4</b>
2.1	Participants . . . . .	4
2.2	Activités globales . . . . .	4
2.3	Tâches jalonnées . . . . .	4
2.4	Rôles et responsabilités . . . . .	5
2.5	Calendrier . . . . .	5

# 1 Apercu du TER

## 1.1 Objectifs

Nous devons développer une bibliothèque indépendante du domaine dans lequel l'agent évolue. Elle fournira plusieurs algorithmes de base de l'apprentissage par renforcement (Q-Learning, SARSA, ...), plusieurs critères de performance, ainsi qu'une ouverture sur l'apprentissage semi-supervisé : c'est à dire avec les retours de l'expert pris en compte.

Dans un premier temps, l'expert pourra simplement compenser la fonction de récompense en précisant si l'agent a bien ou mal agit. Dans un second temps, si le temps le permet, l'expert pourra également agir sur le choix de l'action à entreprendre lors de l'exploration de l'agent, ou encore dire à l'agent s'il est temps d'exploiter ou d'explorer.

Parallèlement au développement de la bibliothèque, afin d'avoir une application pratique de la théorie, on utilisera ces algorithmes dans le simulateur TORCS sur l'apprentissage automatique de la conduite de voiture sur circuit. Nous modifierons également l'interface TORCS pour intégrer les retours positifs ou négatif de l'expert, voire des retours plus complexes cité précédemment. Lan Zhou (2013)

## 1.2 Références

Lan Zhou, M. Z. (2013). *Cahier des charges - TER Semi-supervised Learning Agents*.

PDMIA, G. (2008). *Processus Décisionnels de Markov en intelligence artificielle*. Laboratoire Informatique Paris 6, Equipe MAIA - INRIA (LORIA).

Richard S. Sutton, A. G. B. (1998). *Reinforcement Learning : An Introduction*. MIT Press, Cambridge, MA.

TORCS, T. (2001). The open racing car simulator. <http://torcs.sourceforge.net/>.

## 1.3 Définitions et acronymes

**Env. Réduit** Il s'agit de l'environnement de TORCS avec des perceptions limitées pour l'agent conducteur :

- son angle (par rapport à la droite au milieu de la route)
- la distance entre son centre de gravité et le milieu de la route

ainsi que des actions limitées pour la voiture :

- son accélérateur
- son freinage
- sa direction

On s'attend donc à ce que l'agent dans cette environnement roule toujours avec la première vitesse.

**Env. Complexe** Il s'agit toujours de l'environnement de TORCS mais agrémenté de nouvelles perceptions

- coincé ou non (dans un mur)
- vitesse (pour savoir comment accélérer/freiner)
- RPM (pour savoir quelle vitesse mettre)
- friction du sol (pour déterminer s'il est sur la route ou en dehors, quel type de terrain, ...)
- distance du prochain virage (pour déterminer l'accélération et la vitesse maximale)
- angle du prochain virage (pour le prendre plus sec)

et de nouvelles actions

- boîte à vitesse

**CdC** Cahier des charges

**PdP** Plan de développement

**APR** Apprentissage par Renforcement

En plus de ces définitions, les documents permettant de bien comprendre le plan sont les suivants : PDMIA (2008) et Richard S. Sutton (1998).

## 2 Organisation du TER

### 2.1 Participants

Encadrants :

- Paolo VIAPPANI (Paolo.Viappiani@lip6.fr)
- Paul WENG (Paul.Weng@lip6.fr)

Etudiants :

- Lan ZHOU (lan\_612zhou@yahoo.cn)
- Matthieu ZIMMER (contact@matthieu-zimmer.net)

À l'heure actuelle, il n'y a pas de liens avec d'autres TER, enseignants, chercheurs, laboratoires, entreprises.

### 2.2 Activités globales

**Scripts** Maintenir et développer un ensemble de script permettant de compiler, installer, nettoyer, lancer des tests et faciliter l'ensemble du développement du projet.

**Documentation** Gérer et

**Intégration Linux** Maintenir la compatibilité des sources sous le système Linux.

**Intégration Mac** Maintenir la compatibilité des sources sous le système Mac.

### 2.3 Tâches jalonnées

**Conception bibliothèque** Réfléchir aux différentes classes et méthodes nécessaire pour concevoir une bibliothèque réutilisable. Concevoir les diagrammes de classe.

**Implémentation Q-learning** Implémenter l'algorithme Q-learning dans la bibliothèque de façon générique et réutilisable.

**Implémentation SARSA** Implémenter l'algorithme SARSA dans la bibliothèque de façon générique et réutilisable.

**Interfacer Q-learning - Env. Réduit**

**Interfacer SARSA - Env. Réduit**

**Implémentation fonction d'approximation**

Interfacer Q-learning - Env. Complexe

Interfacer SARSA - Env. Complexe

Implémenter mesure de Perf.

Implémenter fonction de Récompenses

Modifier interface TORCS

## 2.4 Rôles et responsabilités

test

## 2.5 Calendrier

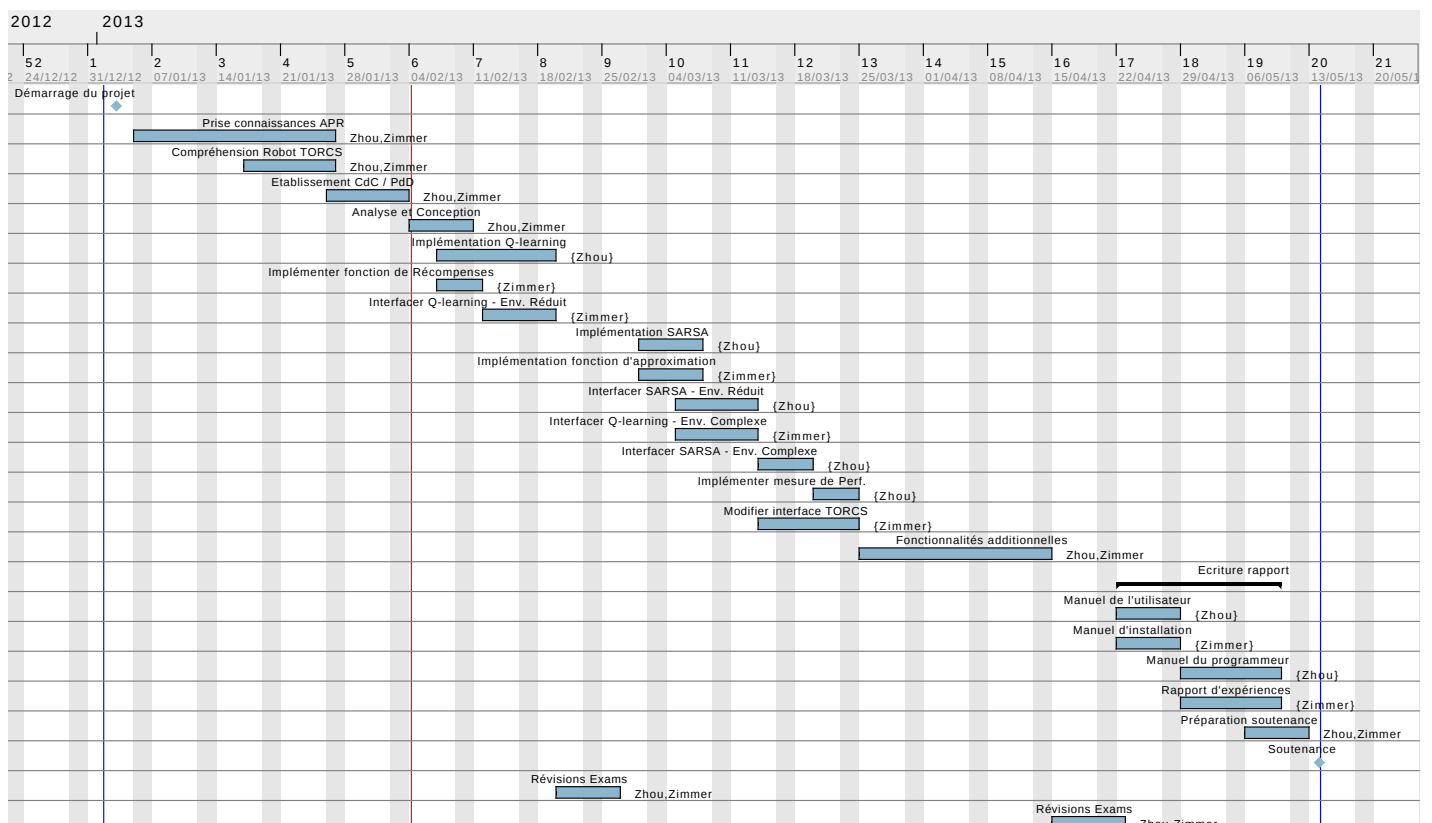


FIGURE 1 – Diagramme de Gantt (zoomer pour meilleur qualité)

## 2 ORGANISATION DU TER

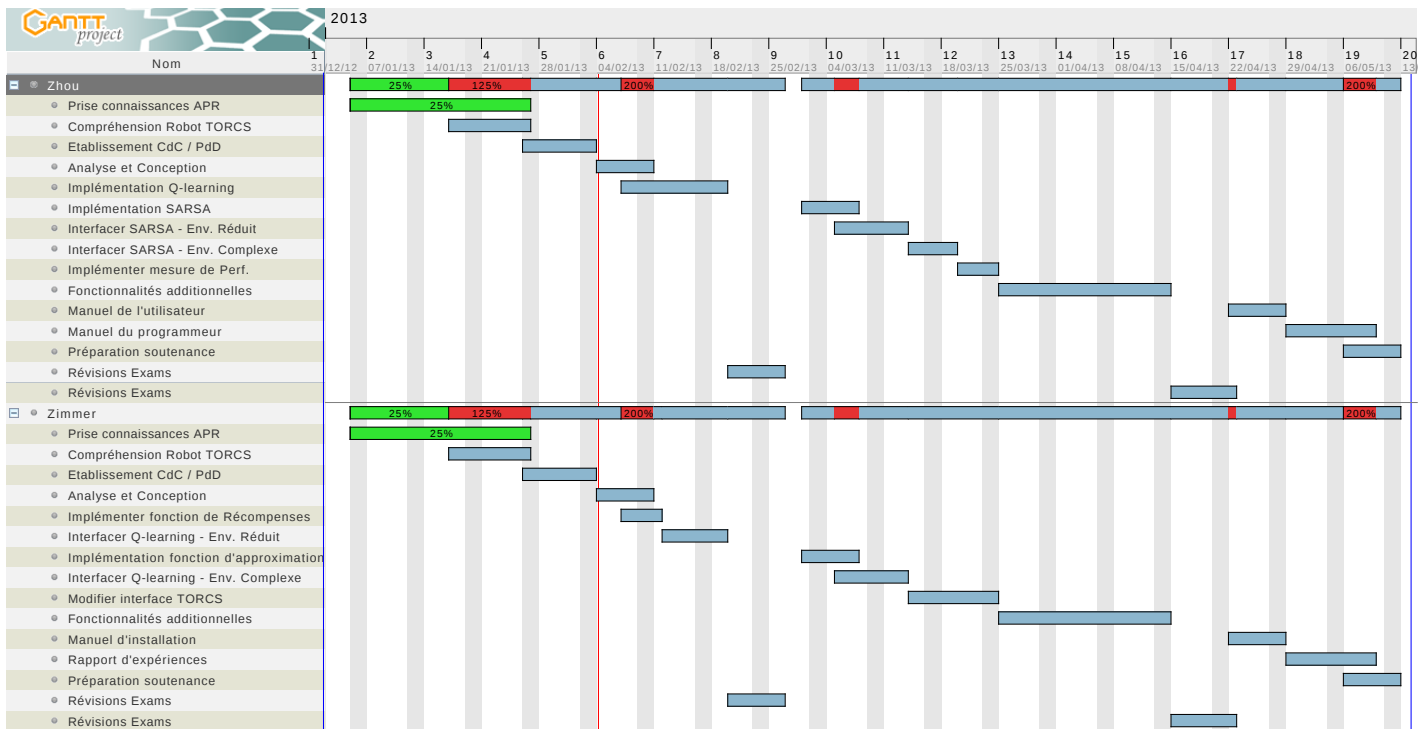


FIGURE 2 – Diagramme de Gantt par Ressource (zoomer pour meilleur qualité)

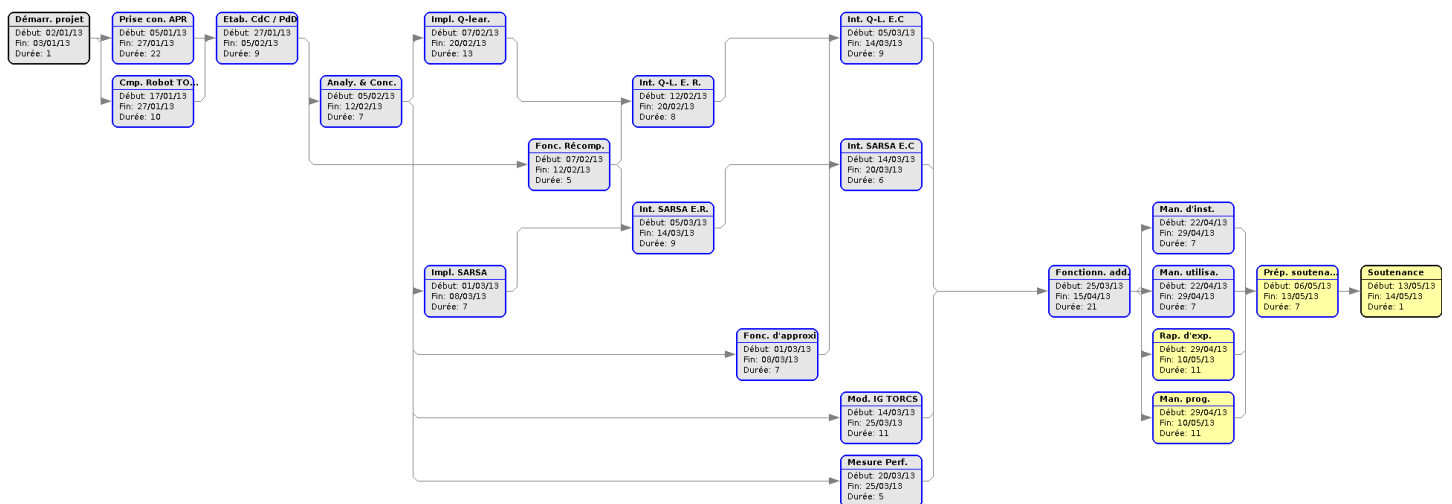


FIGURE 3 – Diagramme de PERT (zoomer pour meilleur qualité)