

Rapport MIF24

Aurélien CHEMIER, Arnaud DUHAMEL, Maëlyss FARGES

2013/2014

Table des matières

1	Introduction	3
2	Agent 1	3
2.1	Description de l'agent	3
2.2	Attributs	3
2.3	Constructeur\Destructeur	3
2.4	save()	4
2.5	chooseResult()	4
2.6	chooseExperience(const Resultat& r)	4
2.7	addMotivation(const Interaction& i)	4
3	Agent 2	4
4	Agent 3	4
5	Conclusion	4

1 Introduction

L'objectif de ce projet est d'implémenter un agent qui apprend à effectuer les interactions positives sans connaître à priori son système motivationnel (mot_1 ou mot_2) ni son environnement (env_1 ou env_2).

- Deux expériences sont possibles $E = \{e_1, e_2\}$
- Deux résultats sont possibles $R = \{r_1, r_2\}$
- Il y a donc quatre interactions possibles : $E \times R = \{i_{11}, i_{12}, i_{21}, i_{22}\}$

Pour coder l'agent nous avons utilisé le langage C++ et Qt.

2 Agent 1

2.1 Description de l'agent

Les environnements sont :

- $env_1 : e_1 \Rightarrow r_1, e_2 \Rightarrow r_2$ (i_{12} et i_{21} ne se produisent jamais).
- $env_2 : e_1 \Rightarrow r_2, e_2 \Rightarrow r_1$ (i_{11} et i_{22} ne se produisent jamais).

Les Systèmes motivationnels sont :

- $mot_1 : v(i_{11}) = v(i_{12}) = 1, v(i_{21}) = v(i_{22}) = -1$
- $mot_2 : v(i_{11}) = v(i_{12}) = -1, v(i_{21}) = v(i_{22}) = 1$

Concrètement, deux interactions donnent un résultat positif et deux autres donne un résultat positif.

2.2 Attributs

Listing 1 – Attributs de la classe Agent

```
SystemeMotivationnel m_motivation;
int m_motivationScore;
QList<Interaction> m_trace;
const Environnement& m_environnement;
QMap<int, Experience> m_exp;
```

2.3 Constructeur\Destructeur

Listing 2 – Constructeur et destructeur de l'agent

```
/**
 * @brief Agent::Agent
 * @param e
 */
Agent::Agent(const Environnement& e): m_environnement(e), m_motivationScore(0)
{
    QFile file("trace.txt");
    if(!file.open(QIODevice::WriteOnly))
    {
        qDebug() << "Ouverture_du_fichier_" << "trace.txt\" << "_impossible";
        return;
    }
}
```

```
        file.close();  
    }  
  
    /**  
     * @brief Agent::~~Agent  
     */  
    Agent::~~Agent ()  
    {  
    }
```

2.4 save()

2.5 chooseResult()

2.6 chooseExperience(const Resultat& r)

2.7 addMotivation(const Interaction& i)

3 Agent 2

4 Agent 3

5 Conclusion