

Projet de FOSYMA

Projet - Wumpus Multi-agent

B. Thanh Luong, Gualtiero Mottola



TABLE DES MATIÈRES

1	Introduction	1
2	Présentation des Agents	1
2.1	Comportement des Agents	2
3	Processus de Communication	2
3.1	Les Processus	2
3.2	Les Outils	2
3.3	turcs pas implementé	2
4	Conclusion	2

1. INTRODUCTION

Ce projet Consiste a développer une version multi-agent d'un jeu Fortement inspiré de "Hunt the Wumpus" cette variante du jeu est définie de la façon suivante : un ensemble d'agents en coopération sont placé dans un environnement inconnu on pour mission d'explorer cet environnement et de récupérer un maximum de trésors qui sont disséminé dans cet environnement. Un agent Wumpus se trouve également dans l'environnement, il se déplace aléatoirement et a pour but de gêner l'exploration et la récupération des trésors.

2. PRÉSENTATION DES AGENTS

Les Trois type d'agent utilisables pour récolter un maximum de Trésors sur la carte sont les suivants : les agents explorateurs qui n'ont pas la possibilité de récupérer des ressources, leur seul but est d'explorer la carte, des Agents Collecteurs qui on un sac a dos correspondant a un type de trésor (TREASURE ou DIAMONDS) et qui on une méthode permettant de récupérer ce type de trésor et le placé dans leur sac si celui ci n'est pas plein, On note que lorsque cette action est exécuté une partie du trésor est perdue. Enfin le dernier type d'agent, l'agent tanker qui ne peux pas ramasser de trésor mais a un sac a dos de capacité illimité, tous les agents collecteurs on la possibilité de donner leur trésors a l'agent tanker. Ce sons les quantité présentes dans L'agent tanker qui seront comptabilisé a la fin de l'exécution.

2.1. COMPORTEMENT DES AGENTS

les comportements de nos trois types d'agents sont tous implémentés sous la forme de `FSMBehaviours` qui sont des automates finis, Chaque état du FSM est un behaviour qui est exécuté selon l'ordre défini par l'utilisateur. nous allons décrire dans cette section le comportement principal de chaque agent, puis dans la section suivante la suite de behaviours qui leur permet de communiquer et qui est identique pour tous les types d'agents.

```
1  FSMBehaviour fsmBehaviour = new FSMBehaviour();
2  fsmBehaviour.registerFirstState(new ExploreBehavior(this),"Exp");
3  fsmBehaviour.registerState(new CheckMailBehavior(this),"Ckm");
4  fsmBehaviour.registerState(new RequestConnectionBehaviour(this),"Com");
5  fsmBehaviour.registerState(new SendMapBehaviour(this),"Smp");
6  fsmBehaviour.registerState(new ReceiveMapBehaviour(this),"Rmp");
7
8  fsmBehaviour.registerTransition("Exp","Ckm",1); //explore to check mail
9
10 fsmBehaviour.registerTransition("Ckm","Com",1); //check mail to start com
11 fsmBehaviour.registerTransition("Ckm","Smp",2); //check mail to send map
12
13 fsmBehaviour.registerTransition("Com","Rmp",1); //com to receive
14
15 fsmBehaviour.registerTransition("Smp","Rmp",1); // send to receive
16 fsmBehaviour.registerTransition("Smp","Exp",2); // send to exp
17
18 fsmBehaviour.registerTransition("Rmp","Exp",1); // receive to explore
19 fsmBehaviour.registerTransition("Rmp","Smp",2); // receive to send
20
21 addBehaviour(fsmBehaviour);
```

La classe offre des méthodes pour enregistrer les états et les transitions qui définissent l'ordre des behaviours

AGENT EXPLORATEUR

AGENT COLLECTEUR

AGENT TANKER

3. PROCESSUS DE COMMUNICATION

3.1. LES PROCESSUS

send map et tout

3.2. LES OUTILS

ici dikstra par ex

3.3. TURCS PAS IMPLEMENTÉ

4. CONCLUSION