

UNIVERSITY OF AUCKLAND

COMPSCI 765

INTERACTIVE COGNITIVE SYSTEMS

Multi-Agent Mafia

Author:

Bradley MOORFIELD

Sebastian CRAM

Derek GALEA

UPI:

bmoo063

scra069

dgal061

June 3, 2015

Contents

1 Introduction

1.1 Overview

Mafia [1] is a social party game where an informed minority, the mafia, conflicts with an uninformed majority, the innocents. The game is typically played with seven to ten players, with two to three mafia for best results. Many variations on the rules exist, but the basic game focuses on the village as a whole debating the identities of the mafia and voting on one player to lynch during the day, with the mafia choosing one player to kill during the night. The game ends when all the mafia have been eliminated, or the mafia outnumber the innocents. Outside of these rules, the players may do anything to try convince the other players of their innocence, or accuse another player of being mafia.

Other roles can be assigned, including a medic and a sheriff. The medic has an opportunity during the night to select one individual to be saved, should they be targeted by the mafia. The sheriff has an opportunity during the night to select one individual and be informed by the game master whether that person is mafia or not.

The game master controls the flow of the game and decides when to advance the game onto the next phase. They request the individuals with a particular role to take action at the appropriate time and announce the results of the nights activities. The game master simply applies the rules of the game, so their actions will be coded in as rules based on the game state.

The other players will not have full knowledge of the game state, so they will form beliefs about the other agents playing the game. They will have beliefs about their own and others roles in the game. They can observe the other agents actions in the game and form beliefs based on those observations. Some basic actions include making an accusation, asking others about their beliefs and making claims about their role in the game. To give them some way to assess the truthfulness of other agents claims we may include unintentional actions like a wavering voice, hesitation or perspiration. So there would be same chance if an agent lies about their role, they do so in a wavering voice. For example, one agent accuses another of being mafia. The accused claims they are the medic and perspires. The accuser then forms the belief that they are mafia, as do the other agents. Then when the

vote is called they are likely to lynch them.

One potential way to make the inferences about agents roles more complex is to give the agents some rudimentary personality. Then if the accuser believes the accused is a shy person there is a competing explanation for their perspiration. Other relevant personality traits would be aggressiveness and honesty. These traits would influence how often they make accusations or lie.

1.2 Motivation

Mafia allows for a variety of challenges from mathematical, computational, and cognitive standpoints. Braverman et al. [2] shows extensive mathematical modelling can be applied to even a simple environment. Computationally it follows from the likes of the Facade and Versu systems in having a simulated set of agents interacting with a user in order to tell a story. The simulated agents also need to interact with each other, as each agent is independent.

Haan et al. [3] have previously implemented a simple mafia playing agent to demonstrate their multi-agent framework, however little reasoning of other agents true beliefs was performed. Our goal is for agents (both real and simulated) to be able to make inferences as to the true status of other agents based on individual announcements and other observables, and perhaps uniquely it requires agents to lie and identify the lies of others. This is a cognitively demanding task, so it will be interesting to see if we can find ways to make the agents intelligently deceptive.

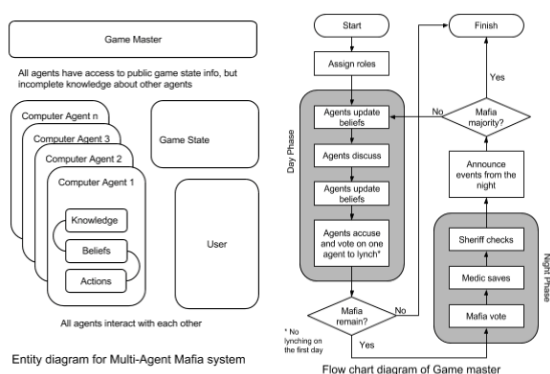


Figure 1: Example image.

2 Analysis

The game starts with the game master assigning roles randomly to all of the players. The game master initiates a discussion phase, then begins the first night phase. The game master first asks the mafia to decide together who to assassinate. Then the medic is called on to select someone to save. Then the sheriff is called on to investigate one player. The game master announces the results of the night and assesses whether the game end conditions are met. Agents update their knowledge and beliefs based on the results. The game master then moves the game into the discussion phase. Agents can make announcements about their beliefs or ask others about their beliefs. To ensure discussions don't go on forever we can limit each agent to two speech actions (this is a problem in real life games and the game master has to exercise judgement about when to terminate discussions). The game master then moves the game into the accusation phase. Each agent can accuse up to two other people. The game master then allows the accused to defend themselves. They may make claims about themselves or a belief about another player. For example, I am the sheriff, and I know Bob is mafia or just deny the accusation or remain silent. Then each player casts their vote publically.

From information about who voted to lynch, inferences can be made, so knowledge about how people voted should be added to each agents knowledge base. For example, if one player made an accusation and claimed they were the sheriff and knew the accused was mafia, a vote is taken and the accused is lynched and they turned out to only be a citizen, the accuser was lying. This may lead players to conclude the accuser is mafia.

The game master again assesses whether the game end conditions are met. If not, another round of discussion takes place. Then another night falls and the night cycle repeats until one team has won.

3 Design

We intend to implement our multi-agent system using Jason [4], a multi-agent systems development platform. The AgentSpeak language is used to program the behaviour of individual agents, based on the belief-desire-intention software model. Jason is developed in Java and is easily customised, which will allow us to quickly

implement a game master and graphical interface to interact with, while we work on building the cognitive agents.

3.1 Overall Architecture

3.2 User Modelling

3.3 Planning Logic

4 Practical

4.1 Illustration

4.2 Limitations

4.3 Demonstration

5 About the Authors

5.1 Bradley Moorfield

5.2 Sebastian Cram

5.3 Derek Galea

6 Conclusion

References