

# Mafia in Town

Agentes e Inteligência Artificial Distribuída

Novembro 2020

## Grupo 16

Francisco Gonçalves

201704790

João Lírio

201705254

José Pedro Baptista

201705255

## Apresentação do projeto

O nosso projeto consiste em simular o jogo de cartas [Aldeia](#) (ou Werewolves), com recurso a agentes e comunicação peer-to-peer, num ambiente distribuído. Para alcançar o descrito, utilizamos principalmente a biblioteca JADE (Java Agent DEvelopment Framework).

Existem muitas versões deste jogo, todas relativamente semelhantes. Em seguida apresentamos a nossa, e as respetivas regras.

# Descrição do jogo

A nossa edição do jogo, **Mafia in Town**, consiste em 3 fações que lutam pelos seus interesses: **Town**, **Mafia** e **Neutral** (cada uma com os seus papéis internos). O desenvolvimento do jogo dá-se através da transição entre noite e dia. Usamos os termos votado e morto para representar quando um jogador é eliminado.

Durante o **dia**, ocorre uma discussão entre todos os agentes, seguida duma votação que dita que jogador será eliminado (pode ser *skipped* - ninguém é escolhido) e durante a **noite** certas personagens intervêm resultando em revives, eliminações ou só suspeição.

Fação	Subfação	Descrição
Town	Villager	Apenas tem a função de votar (ou não) em alguém para ser enforcado.
	Healer	Todas as noites pode salvar alguém (incluindo a si mesmo) de um possível ataque da máfia.
	Detective	Todas as noites pode investigar alguém. Não obtém confirmação relativamente à subfação dessa pessoa, apenas se é ou não suspeita, conforme tenha ou não atividade noturna.
Mafia	Leader	Todas as noites escolhe uma (ou mais) pessoa(s) para atacar. Se a Mafia tiver algum Killing, um desses agentes trata de executar a ordem. Caso o Leader esteja só, ele próprio executa a ordem.
	Killing	Todas as noites mata a pessoa que o Mafia Leader ordenar. Caso o Leader já não esteja em jogo, este pode decidir quem atacar.
Neutral	Jester	Ganha o jogo se for votado de dia pela aldeia.

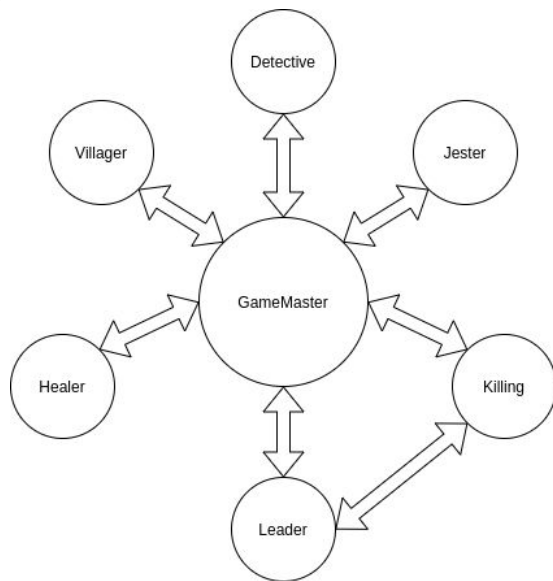
## Condições de fim de jogo

O jogo termina sempre com uma fação vencedora.

Caso não reste nenhum jogador da **Town**, ganha a **Mafia**, no caso inverso ganha a **Town** e na situação em que o Jester é votado de dia a fação **Neutral** sai vitoriosa.

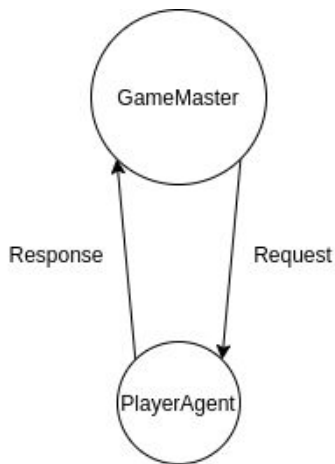
# Esquema global de Agentes

Na nossa implementação deste jogo, existem dois tipos de agentes completamente distintos: o **GameMaster** e o **PlayerAgent**. O primeiro tem a função gerir o jogo, isto é, de ditar as fases do dia, fases de voto, fases de discussão, e manter todos os outros agentes informados do estado do jogo. O **PlayerAgent**, é, em contrapartida, uma simulação do jogador, isto é, tem a função de tomar decisões de jogo, com base na informação que o **GameMaster** partilhou com ele e o seu objetivo no jogo. Este varia conforme a sua **sub-classe**, e há uma para cada tipo de sub-faço, p.e. **Villager** ou **Leader**.

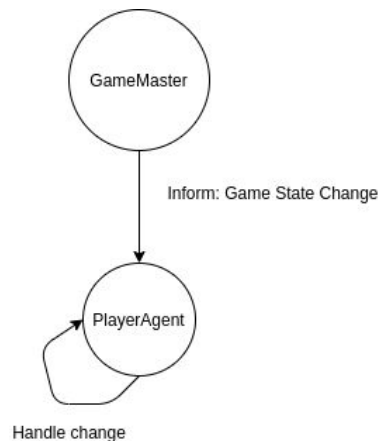


# Interação e Protocolos (comunicação)

No decorrer deste projeto foram desenvolvidos vários Behaviours e Protocolos para fins de interação entre agentes. Decidimos distinguir aqueles que são utilizados mais vezes e / ou são mais complexos.

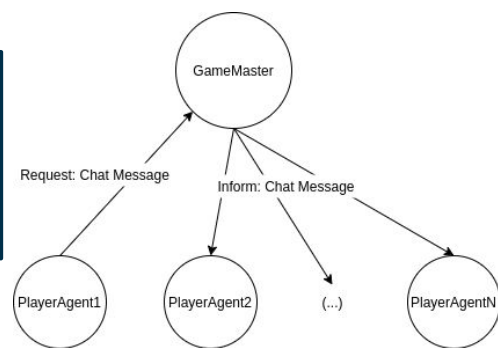


Este tipo de interação é a mais utilizada ao longo do nosso jogo. É utilizada por todos os **PlayerAgents**, entre outros casos, na fase inicial, onde o GameMaster informa todos os jogadores do nome dos restantes jogadores, e na escolha do jogador que o Healer pretende visitar durante a noite

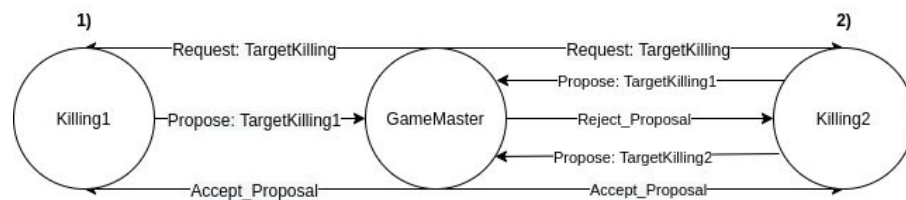


Este tipo de interação é também bastante utilizada e igualmente imprescindível para o correto funcionamento do nosso jogo. Todas as atualizações do estado do jogo que não envolvam responder ao **GameMaster** são realizadas deste modo.

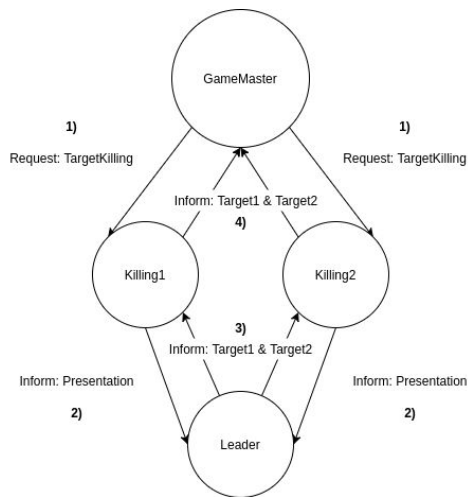
Esta interação é utilizada para todas as mensagens que ocorrem durante a fase de discussão (mensagens de chat). Neste caso, um agente que queira partilhar alguma informação com os restantes, envia a mensagem como request ao **GameMaster**, e este transmite ao todos os outros como inform.



Esta interação representa um processo de negociação entre o **GameMaster** e um agente **Killing**. Sempre que não existe nenhum **Leader** em jogo, este protocolo é posto em uso. Tem por finalidade impedir que 2 ou mais agentes do tipo **Killing** escolham o mesmo alvo durante a noite. Para isso, eles devem de tomar uma decisão, e comunica-la ao **GameMaster** como proposta. Este, verifica se o alvo já lhe tinha sido comunicado por mais algum **Killing**. Em caso afirmativo, recusa a proposta, e o agente **Killing** deve fazer uma nova (com outro alvo).



Este protocolo é utilizado sempre que existem **Killings** e **Leaders** em jogo. O agente **Killing**, ao receber o request do **GameMaster**, apresenta-se ao seu **Leader** e espera que o mesmo lhe dê um alvo para atacar. Ao receber a resposta do **Leader**, este responde ao **GameMaster** com o alvo que o **Leader** escolheu.



# Estratégias dos agentes

Os agentes têm diferentes estratégias durante o jogo dependendo do seu objetivo final. No entanto, todos têm como base nas suas decisões os níveis de suspeição em relação a todos os outros agentes vivos. O nível de suspeição varia ao longo do jogo, sendo que essa variação não é igual para todos os agentes, já que têm um atributo **trait** que pode ser **peaceful**, **mild**, **aggressive** e **over the line**. Este parâmetro representa o comportamento/caráter do jogador que influencia a variação de suspeição.

Os agentes da **Mafia** optam na maior parte das vezes por matar aqueles que à partida são os agentes que a Town menos suspeita. E durante o dia votam quase sempre *Skip*, a não ser que haja um agente altamente suspeito que vá ser votado por todos.

O **Detective** procura sempre investigar agentes que nunca tenha investigado e que ao mesmo tempo sejam dos mais suspeitos. Se o agente investigado tiver atividade durante a noite a sua suspeita aumenta, caso contrário diminui.

O **Healer** procura salvar durante a noite aqueles que são menos suspeitos de forma a que durante o dia seja mais fácil fazer a decisão sobre em quem votar. Caso este salve alguém que foi atacado, a suspeita de ambos diminui para todos os agentes.

Durante o dia, toda a **Town** procura votar no agente que para eles é o mais suspeito de todos.

# Outros Mecanismos

Na fase inicial do jogo, todos os agentes fazem uso da funcionalidade de **Páginas Amarelas** do Jade para descobrir o **GameMaster** e guardam o DFAgentDescription obtido, para uso futuro. Após isto, os **PlayerAgents** registam-se também nas páginas amarelas do Jade, da seguinte forma:

```
ServiceDescription sd = new ServiceDescription();  
sd.setName("AgentType");  
sd.setType("Player");  
dfad.addServices(sd);
```

Este registo (e este serviço, em particular) é muito útil, uma vez que vai permitir ao **GameMaster**, apenas com uma pesquisa, encontrar todos os **PlayerAgents** (DFAgentDescription) e guardá-los numa estrutura de dados interna, para que possa comunicar facilmente com eles durante o decorrer do jogo.

# Experiências realizadas

As seguintes experiências foram realizadas com a seguinte população inicial: 16 villagers, 3 healers, 1 detective, 1 jester, 2 killers, 1 leader.

Na tabela é possível ver o número de sobreviventes de cada facção no fim da partida.

	Villagers	Healers	Detective	Jester	Killer	Leader
Experiência 1	0	0	0	0	2	0
Experiência 2	12	3	1	0	2	1
Experiência 3	0	0	0	0	2	1
Experiência 4	8	3	1	0	0	0
Experiência 5	0	0	0	0	1	1



# Experiências realizadas

Observando o resultado das experiências podemos destacar a experiência 2 em que o vencedor foi o **Jester** por ter sido expulso pela aldeia durante o dia. Isso foi motivado pela investigação do **Detective** que detetou atividade noturna pela parte do **Jester**.

Na experiência 3 a **Town** foi vencedora e isso deveu-se principalmente ao facto de a **Mafia** não ter morto qualquer **Healer** ou **Detective** durante a noite, visto que estes têm papéis muito importantes na defesa da aldeia e na descoberta da **Mafia**.

Na experiência 3, a **Mafia** conseguiu por acaso eliminar o **Detective** logo na primeira noite o que facilitou bastante o seu trabalho acabando por vencer sem qualquer baixa.

# Conclusões e observações

Pensamos que o resultado deste projeto foi positivo, sendo que os principais objetivos do mesmo foram cumpridos. O jogo foi implementado na sua totalidade e tal como descrito na especificação.

À medida que fomos avançando no projeto, percebemos que seria um grande desafio implementar a lógica das decisões dos agentes de forma precisa, uma vez que existem várias variáveis que podem condicionar as decisões dos mesmos no que toca a atacar, investigar, salvar, votar durante o dia e posteriormente processar o resultado dessas decisões.

Apesar disso, conseguimos encontrar um equilíbrio razoável fazendo com que os vencedores das partidas e os sobreviventes variem de forma significativa.

Para trabalho **futuro**, seria interessante adicionar mais fatores que condicionassem as ações dos agentes de forma a que tomassem decisões mais racionais e naturais em determinadas situações principalmente no que toca ao voto de expulsão que acontece durante o dia. Seria também interessante que parâmetros como a **trait** se refletissem ainda mais no desenlace do jogo, simulando o comportamento natural, dum jogador que tem conhecimento do estado jogo muito limitado.