

Universidade do Minho - Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Agentes Inteligentes

Trabalho prático

Modelação AUML para um

Sistema de Simulação de Fogos

Marcos Filipe Morais Andrade a59776

Sérgio Tiago Oliveira Jorge a77730

Vitor José Ribeiro Castro a77870

8 de Dezembro de 2019

Resumo

Este relatório descreve toda a fase de planeamento, arquitetura e implementação de uma solução multiagente, desenvolvida no âmbito da unidade curricular de Agentes Inteligentes, 2019, Universidade do Minho. O objetivo do trabalho proposto é construir uma solução baseada em agentes, que permita fazer um combate a fogos florestais de maneira eficiente, conforme os meios disponíveis.

Conteúdo

1	Introdução	3
1.1	Contextualização	3
1.2	Caso de Estudo	3
1.3	Estrutura do Relatório	3
2	Estado de arte	4
3	Modelação	5
3.1	Diagramas de Sequência	5
3.1.1	Alerta Novo Fogo	5
3.1.2	Apagar Incêndio	5
3.1.3	Reabastecer	6
3.2	Diagrama de Atividades	8
3.3	Diagramas de estados	9
3.4	Diagrama de Classes	10
4	Implementação	12
4.1	Novo Fogo	12
4.1.1	Agente Incendiário	12
4.1.2	Agente Central	12
4.2	Em viagem	12
4.2.1	Agente Quartel	12
4.2.2	Agente Bombeiro	12
4.2.3	Agente Interface	13
4.3	Apagar Incêndio	13
4.3.1	Agente Quartel	13
4.3.2	Agente Bombeiro	13
4.4	Reabastecer	14
5	Caso de Teste	15
6	Conclusão	16

1 Introdução

1.1 Contextualização

Este relatório surge no âmbito do trabalho prático da unidade curricular de Agentes Inteligentes, pertencente ao perfil de Sistemas Inteligentes do Mestrado Integrado em Engenharia Informática, da Universidade do Minho.

Nos últimos anos, com o aumento da gravidade e impacto dos fogos florestais em Portugal, tem sido feito um esforço maior em prevenir e combater este problema. Nessa perspetiva, foi proposta a conceção de um sistema multiagente simulador de combate a fogos, por forma a estudar a eficiência e capacidade de reação a estes acontecimentos.

1.2 Caso de Estudo

O projeto tem por base a concepção de um sistema, desenvolvido num ambiente JADE (Java Agent Development Framework), para monitorizar o surgimento de fogos, bem como a localização e recursos dos meios disponíveis para os combater, numa área de simulação.

Para a referida simulação haverá três tipos de agentes com funções distintas: agente incendiário, agente central e agente participativo. O agente incendiário será responsável por atear e comunicar o local onde se iniciou o fogo. O agente central receberá os dados do incendiário e tratará da comunicação e controlo das unidades de combate a incêndio disponíveis. Por sua vez, estas unidades de combate a incêndio são os agentes participativos, que têm recursos limitados de água e combustível para as suas deslocações. Para dar suporte aos recursos necessários, haverá locais de captação de água e combustível em locais aleatórios, definidos na criação inicial do mapa de simulação.

1.3 Estrutura do Relatório

Este projeto está dividido em duas fases, sendo a primeira associada ao planeamento da solução a construir e a segunda à implementação respetiva.

Inicialmente, aborda-se o Estado da Arte, na secção 2, onde se verifica o uso de simulações na prevenção de desastres naturais de sistemas multiagente.

Com vista a resolver as diferentes necessidades de modelação de um sistema por Agentes, utilizou-se a linguagem UML (*Unified Modeling Language*) modificada AgentUML. É especificada, na secção 3, a interação e protocolo existente entre os diversos agentes, por forma a fazer a simulação no formato mais eficiente possível. De seguida, são apresentadas as estratégias de implementação e os algoritmos utilizados nos diferentes estágios do problema.

Na secção seguinte é apresentado o caso de teste utilizado pelo grupo para testar as funcionalidades do programa e o relatório termina com uma conclusão onde o grupo aborda as dificuldades encontradas e aquilo que retirou de positivo com a realização deste trabalho.

2 Estado de arte

Os desastres naturais que atingem vários locais do mundo têm-se vindo a intensificar, nos últimos anos. Os governos têm a obrigação de proteger os cidadãos destes acontecimentos e agir da forma mais eficiente possível, causando o menor dano material e humano. São exemplos de desastres naturais que obrigaram a uma ação forte o *tsunami* na Indonésia (2004), a chuva de granizo com alta intensidade na China (2008), o tremor de terra no Japão (2011), as cheias na Índia (2013) [1] e, com afeção a Portugal, o incêndio de Pedrógão (2017), que vitimou 66 pessoas e feriu 254 [2].

Considerando a importância e impacto que os desastres naturais podem ter na população, há empresas dedicadas ao estudo e simulação destas catástrofes [3]. A capacidade de agir de forma dinâmica, com tolerância a falhas, e capacidade de recolher e tratar, em simultâneo, informação relativa ao plano de ação são características fortes destes sistemas. Um sistema multiagente permite, adicionalmente, tomar decisão de forma distribuída, tornando a ação ágil, mesmo que implique a ação humana. É possível ajustar o comportamento destes agentes mediante alterações de algoritmos, conforme a situação, o que faz deste sistema algo extremamente flexível e confiável [4].

Exemplos da aplicabilidade do funcionamento de sistemas multiagentes podem ser encontrados em várias indústrias. Um exemplo dessa aplicação é o controlo de recursos no serviço de ambulâncias do Reino Unido, em que se recorre a um sistema de multiagentes para simular as necessidades das áreas por este abrangido. A melhoria da qualidade da simulação trazida pelo sistema multiagente traz maior confiança na alocação feita, quer a nível de recursos (ambulâncias, médicos, paramédicos), quer a nível de urgência (gravidade do incidente) [5]. Adicionalmente, pode-se encontrar aplicabilidade em sistemas de gestão de tráfego urbano, em que diversos agentes monitorizam parâmetros ambientais (como nível de ruído, contagens, luz ambiente, entre outros) e reportam a sistemas centrais. Esta é uma aplicação que, em conjunto com os agentes nos semáforos, podem permitir acelerar zonas em que o tráfego está lento e ou mal aproveitado, contribuindo para um melhor fluxo urbano e melhorando os parâmetros gerais [6]. De entre outras aplicações, este tipo de sistemas é também usado em sistemas de transmissão de dados em tempo real, na área de saúde. Em determinados casos, a transmissão de dados de forma rápida e redundante é imperativa, pelo que muitas vezes os dados são recolhidos por sensores diversos e geridos por sistemas independentes, sob a forma de agentes [7].

No entanto, não há nenhum sistema autónomo baseado em agentes que tenha um uso na área do combate a incêndios, ao contrário do que acontece em indústrias como a aviação.

3 Modelação

De forma a representar e a especificar o sistema a implementar de uma forma padronizada, e para facilitar na compreensão, decidiu-se utilizar a modelação **AUML**, Linguagem de Modelagem Unificada para Agentes. Por isso, serão, de seguida, apresentados **diagramas de sequência** onde se fará a representação da troca de mensagens que existirá entre os diversos agentes. Será, também, mostrado o tipo de *performatives* e algum do conteúdo que terão. Depois, o **diagrama de atividades** facilitará a representação daquilo que os agentes fazem e, principalmente, daquilo que fazem internamente. Posteriormente, mostra-se o **diagrama de estados** que retrata os estados de comunicação e as transições que são feitas entre estes. A modelação termina com um **diagrama de classes** que ilustra a estrutura geral daquela que será a solução apresentada pelo grupo.

3.1 Diagramas de Sequência

3.1.1 Alerta Novo Fogo

O agente incendiário, em tempo aleatório, decide criar um novo incêndio, numa localização aleatória. De seguida, envia uma mensagem para o agente central com as coordenadas do incêndio que criou. É, por isso, uma comunicação que se torna essencial já que o quartel deve saber a localização dos incêndios para que possa alertar os agentes de combate a incêndio e fazer o respetivo controlo da situação. Então, quando essa localização chega ao quartel, este trata de registar o incêndio no DF, bem como o estado atual deste (a sua intensidade, se está a ser combatido e se está ou não ativo). O diagrama seguinte representa essa interação.

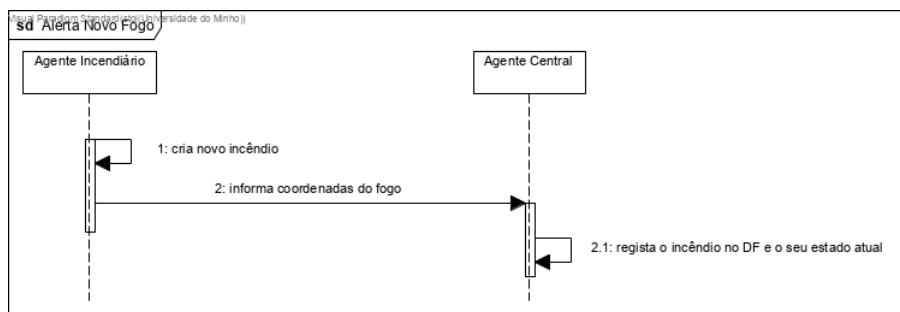


Figura 1: Diagrama que descreve a criação de um incêndio

3.1.2 Apagar Incêndio

Como foi visto anteriormente, o quartel é informado quando um incêndio é despoletado. Ou seja, a central verifica, efetivamente, que está a ocorrer um incêndio. De seguida, calcula, internamente e tendo em conta o DF de que dispõe, qual a unidade de combate a incêndio que deve selecionar para combater

o fogo, mediante a distância e os recursos que tem disponíveis. Após fazer este cálculo, envia mensagem para a unidade de combate selecionada, indicando as coordenadas para onde esta se deve dirigir. O agente de combate dirige-se, então, para o fogo e, chegado ao local, reporta enviando uma mensagem para o quartel. De seguida, o fogo é combatido pela unidade e, por cada unidade de água despendida, envia uma mensagem para a central. A central verifica, após esta mensagem, se o recurso despendido foi necessário para apagar o incêndio, sendo que, caso tenha sido, informa o agente de combate que pode parar a sua ação.

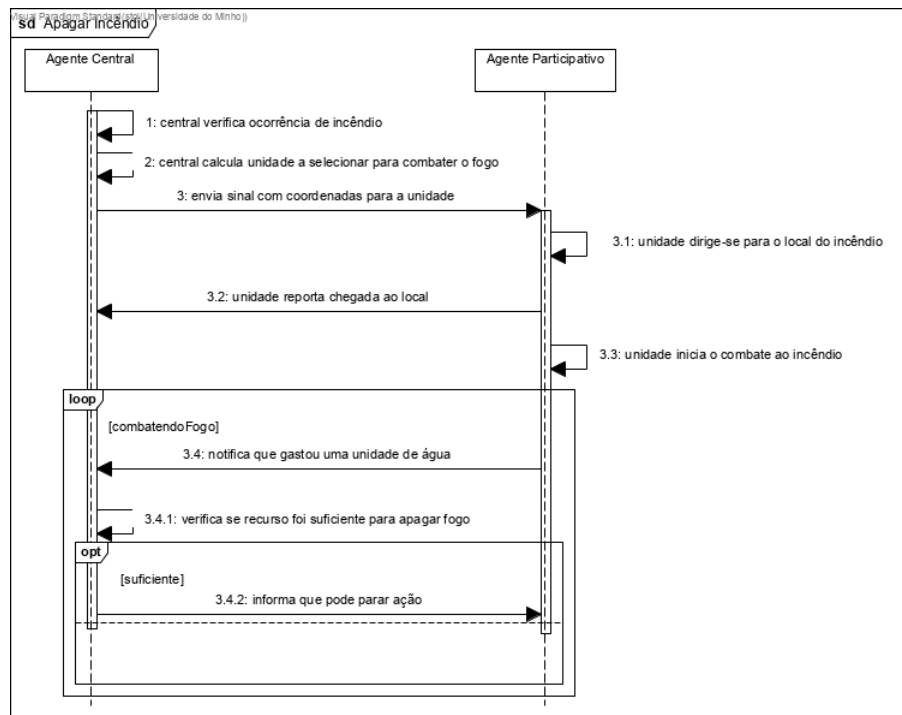


Figura 2: Diagrama que descreve o combate ao fogo

3.1.3 Reabastecer

Quando os agentes participativos fazem viagens, desde os incêndios, para os incêndios, ou para os postos de abastecimento, gastam combustível. Além disso, quando combatem os fogos, gastam, necessariamente, água. Por isso, devem ser capazes de reabastecer os dois recursos que gastam frequentemente. Para tal, as unidades verificam, periodicamente, os níveis atuais de recursos e a distância até aos mesmos. Se houver necessidade de abastecimento dirigem-se, logo que estejam fora de combate, aos devidos locais.

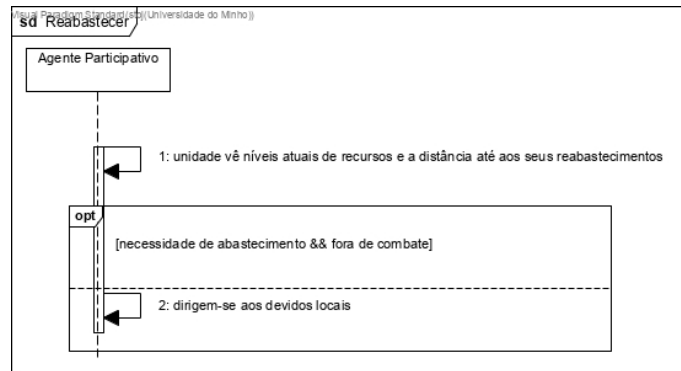


Figura 3: Diagrama que descreve o reabastecimento

3.2 Diagrama de Atividades

No diagrama abaixo podemos detetar a existência de três grupos distintos de agentes e como estes se comportam no sistema. São eles o Agente Incendiário, o Agente Central e o Agente Bombeiro.

Agente Incendiário O propósito dos agentes do tipo incendiário é tão somente iniciar um incêndio numa posição aleatória e, de imediato, reportar o incêndio ao agente central.

Agente Central O agente central, que equivale ao quartel general de operações, onde são tomadas as decisões de ação consoante os recursos disponíveis.

Agente Participativo Os agentes participativos são responsáveis por receber ordens do agente central com as coordenadas do incêndio, deslocar-se até o local e apagar o incêndio. Além disso, devem controlar os seus níveis de recursos de forma autónoma e segura.

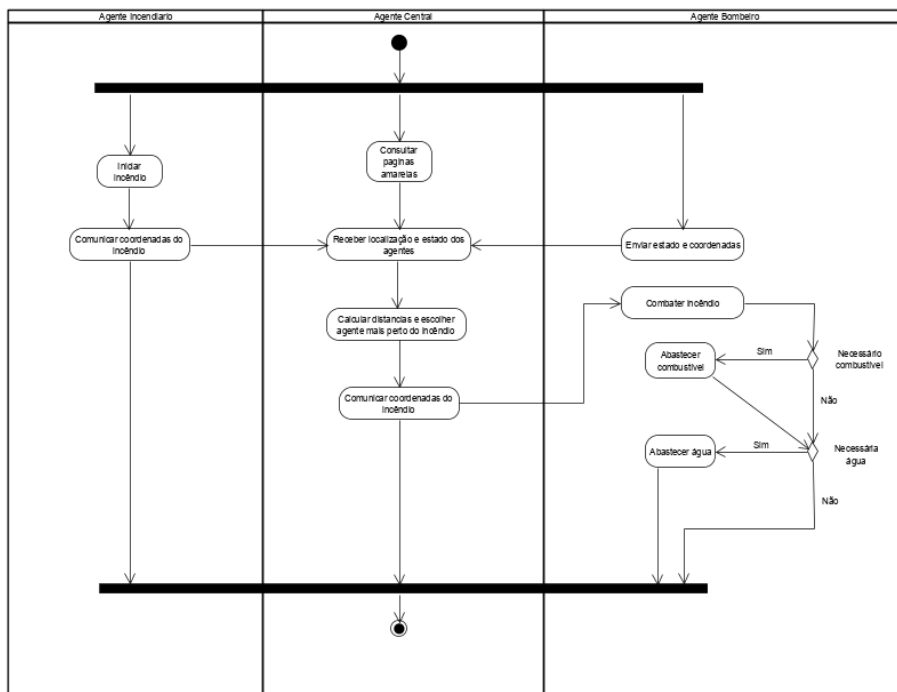


Figura 4: Diagrama de atividades dos agentes

3.3 Diagramas de estados

O diagrama de estados reproduz dois estados possíveis pertencentes aos agentes participativos. Portanto, o agente passa por dois tipos de estados:

- **Estado Standby**, ou de espera, é o estado que o agente toma quando não está a apagar incêndios. Neste estado o agente informa sempre ao agente central das suas coordenadas no mapa geográfico, esperando uma ordem de deslocação a incêndio em qualquer momento. A partir do momento em que recebe essa ordem, passa a um estado ativo de deslocação.
- **Estado Deslocação** consiste nas ações que um agente toma como objetivo apagar um incêndio. Tal estado consiste na deslocação até às coordenadas do incêndio e proceder a apagar o fogo. Por cada recurso de água gasto é enviada mensagem ao agente central, para que o mesmo possa controlar a dimensão do fogo. Posteriormente, é feito o controlo dos seus recursos, sendo que, caso o agente necessite, desloca-se para a base de abastecimento mais próxima. Findo o reabastecimento, volta ao estado de “Stand by” até ao próximo aviso de incêndio lhe seja comunicado.

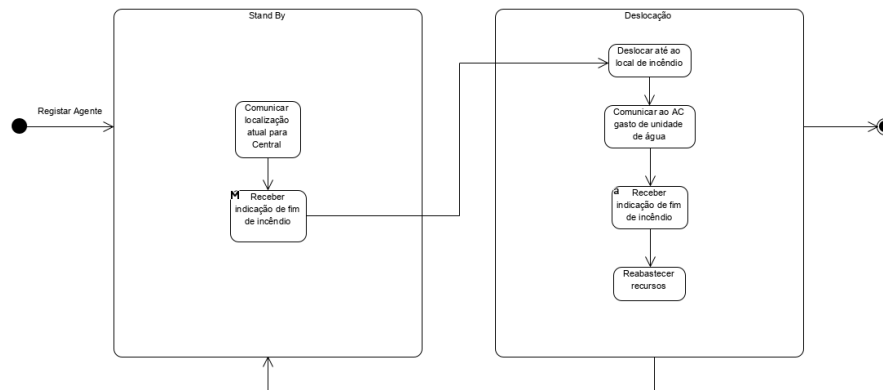


Figura 5: Diagrama de estado de um agente participativo

3.4 Diagrama de Classes

Há, desde logo, quatro classes principais:

- Classe **AgenteIncendiario** - responsável por atear os fogos;
- Classe **AgenteCentral** - representa o quartel dos bombeiros;
- Classe **AgenteParticipativo** - representa os agentes de combate a incêndios, ou seja, drones, aeronaves e camiões;
- Classe **AgenteInterface** - mostra ao utilizador como os agentes se movimentam entre incêndios.

As classes anteriormente referidas são especificações da classe *Agent*, que se trata de uma classe que está disponível através da biblioteca em estudo, *Jade*.

Verifica-se, então, que o agente participativo tem alguns métodos, como por exemplo o *Movimento* e *ReceberInfoCombate*, que são especificações da classe abstrata *CyclicBehaviour* e possibilitam que o agente tenha um movimento autónomo e que consiga receber coordenadas de incêndios, e *enviarInfoCentral*, que é uma especificação da classe abstrata *TicketBehaviour* e possibilita a que o agente possa enviar as coordenadas da sua localização de forma periódica. O agente participativo tem algumas variáveis de instância como a sua localização, a quantidade de água e combustível disponível nos seus reservatórios, a localização destino, booleanos de controlo e listas com a localização de bombas de combustível e locais de abastecimento de água.

O agente central tem o método *receiver* que constitui uma classe com um comportamento cíclico de forma a orientar todo o sistema.

O agente incendiário tem um método *incendiar*, com um comportamento cíclico e que lhe permite criar incêndios de forma totalmente aleatória.

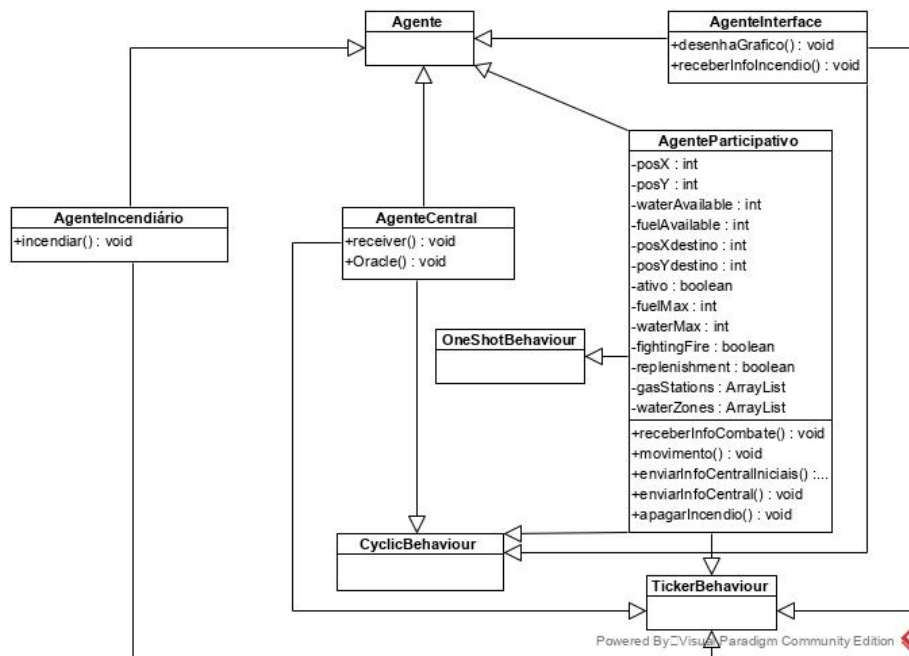


Figura 6: Diagrama de Classes

4 Implementação

4.1 Novo Fogo

4.1.1 Agente Incendiário

Com um comportamento *TickerBehaviour*, o agente incendiário calcula uma localização aleatória, cria um incêndio no local e informa a central das coordenadas.

4.1.2 Agente Central

Quando recebe uma performative *inform*, a mensagem é proveniente de um incendiário que acabou de criar um fogo. Por isso, recebe as coordenadas do incêndio e guarda-as na lista de fogos. Posteriormente, verifica se existe alguma bombeiro disponível para combater o fogo e, portanto, se existe encaminha-o para o incêndio.

Possui também um oráculo que permite monitorizar o estado dos incêndios. Caso se verifique que, por falta de meios, em algum momento, foi impossível lidar com um fogo, despoletar-se-á uma chamada de bombeiro para aquela localização, contando com o aumento da intensidade do fogo correspondente.

4.2 Em viagem

4.2.1 Agente Quartel

Quando recebe uma performative *propose*, um bombeiro está a informar as suas coordenadas e o seu estado, ao segundo, quando em modo ativo. Ou seja, só quando um bombeiro confirma que vai combater um incêndio.

4.2.2 Agente Bombeiro

O agente bombeiro, a partir de um comportamento *OneShotBehaviour* começa por enviar as suas coordenadas iniciais ao quartel. Além disso, informa que não se está a movimentar e que não está a combater nenhum fogo. Do mesmo modo, comunica acerca da quantidade de água e de combustível que possui. Definiu-se que essas quantidades estão na capacidade máxima, no início da aplicação.

Posteriormente e com um comportamento *TickerBehaviour*, a cada segundo, vai reportando sobre todas essas informações. Deste modo, a central, está sempre a par do estado de cada agente.

Quando o agente se está a mover, naturalmente, informa ao segundo da sua localização e dos seus recursos diretamente ao agente interface de forma a possibilitar a visualização do movimento do agente graficamente.

Ou seja, se o bombeiro recebe um pedido de combate a incêndio por parte da central, significa que não se encontra ocupado. Portanto, deste modo, com um *CyclicBehaviour*, quando este recebe um pedido, atualiza as suas coordenadas de destino, define a sua direção e confirma, ao agente central que, de facto, vai

combater o incêndio. Se, em vez disso, recebe uma confirmação por parte da central, então é expectável que o fogo tenha sido combatido com sucesso que o agente bombeiro tenha concretizado o trabalho ao qual se propôs. De seguida, pode então verificar se há uma necessidade de abastecimento antes de estar pronto para realizar o próximo trabalho.

4.2.3 Agente Interface

Para desenvolvimento de um simulador mais intuitivo e para efeito de uma melhor demonstração do trabalho feito, o grupo decidiu acrescentar uma nova funcionalidade ao agente interface com o auxílio das bibliotecas *LWJGL* (*Light-weight Java Game Library*) e *Slick2D*. Desta forma criou-se um “tabuleiro” a simular o terreno onde os fogos são acendidos pelo agente incendiário e apagados pelos agentes participativos. Toda a comunicação das posições, tipos de agentes participativos, quantidade de água e de combustível que os agentes têm disponível, postos de abastecimento e área de fogos são representados no mapa. Esta experiência permite-nos ver em tempo real as ações dos agentes e a interação entre os mesmos. O Agente Interface tem um *CyclicBehaviour* no qual recebe, continuamente, as coordenadas, as informações de água e as informações de combustível, relativamente a cada agente. Com esta informação podemos adicionar ou atualizar as estruturas de dados a cada ciclo para que possam ser representadas graficamente. O desenho e a interface gráfica da aplicação realiza-se com um *TickerBehaviour* que faz uma atualização ao gráfico de meio em meio segundo. Este método permite construir o “tabuleiro” com recurso às bibliotecas de *Java* (*LWJGL* e *Slick2D*) e *OpenGL*.

4.3 Apagar Incêndio

4.3.1 Agente Quartel

Quando recebe uma performative *confirm*, o bombeiro está a responder a uma proposta de combate a incêndio.

Quando recebe uma performative *inform_if*, um bombeiro está a reportar o gasto de unidade de água para um incêndio e, por isso, a central pode então reduzir a intensidade do fogo. Quando a quantidade de água gasta pelo bombeiro é a suficiente para a apagar o fogo em questão, a central, a partir de uma performative *confirm*, avisa o bombeiro que o fogo se encontra extinto.

4.3.2 Agente Bombeiro

O combate ao incêndio é através de um *TickerBehaviour* e, a cada segundo, o agente gasta uma unidade de água e informa a central relativamente à água que gastou e ao fogo que se encontra a combater.

4.4 Reabastecer

O reabastecimento é de extrema importância numa situação real. Garantir que os veículos não fiquem sem combustível garante, por sua vez, a sobrevivência de unidades materiais e humanas. Adicionalmente, é também necessário ajustar a recolha de água a situações de vida real.

Tendo em conta esta necessidade de representar a realidade, o abastecimento de combustível é feito mediante o mínimo atribuído a cada veículo. De facto, exige-se que cada veículo tenha sempre, no mínimo, o combustível suficiente para percorrer, ida e volta, 1 quadrante do mapa. Esta medida acontece tendo em conta a distribuição uniforme, conforme a realidade, de postos de combustível.

Por outro lado, a água segue diferentes padrões de abastecimento. Para o caso dos drones, como têm pouca quantidade de água, só faz sentido reabastecerem quando estiverem vazios. No caso dos aviões, apenas deve acontecer o reabastecimento quando estes estiverem sem água porque é um meio caro e ao qual se exige muita disponibilidade e eficiência de operação. Por fim, em relação aos camiões, estes tendem a não se querer ficar sem água, para evitar eventuais situações em que os operacionais que se deslocam no mesmo tenham que dar uma resposta rápida a um foco que tenta cercar um local de passagem.

Os agentes são confrontados constantemente com os níveis de água e combustível, calculando quando necessário qual o ponto de abastecimento mais próximo e mostrando-se indisponíveis para atender outros pedidos.

5 Caso de Teste

Para testar o programa elaborado, criou-se a topologia pedida pelo professor. Ou seja, criaram-se dez drones, duas aeronaves e cinco camiões. Além disso, o grupo criou quatro postos de gasolina (um em cada quadrante do mapa e em sítios aleatórios), de forma a simular a uma distribuição real. Criou também seis postos de abastecimento de água, em locais totalmente aleatórios do mapa

Com este sistema simples, verificou-se que os agentes tomaram as decisões esperadas neste contexto.

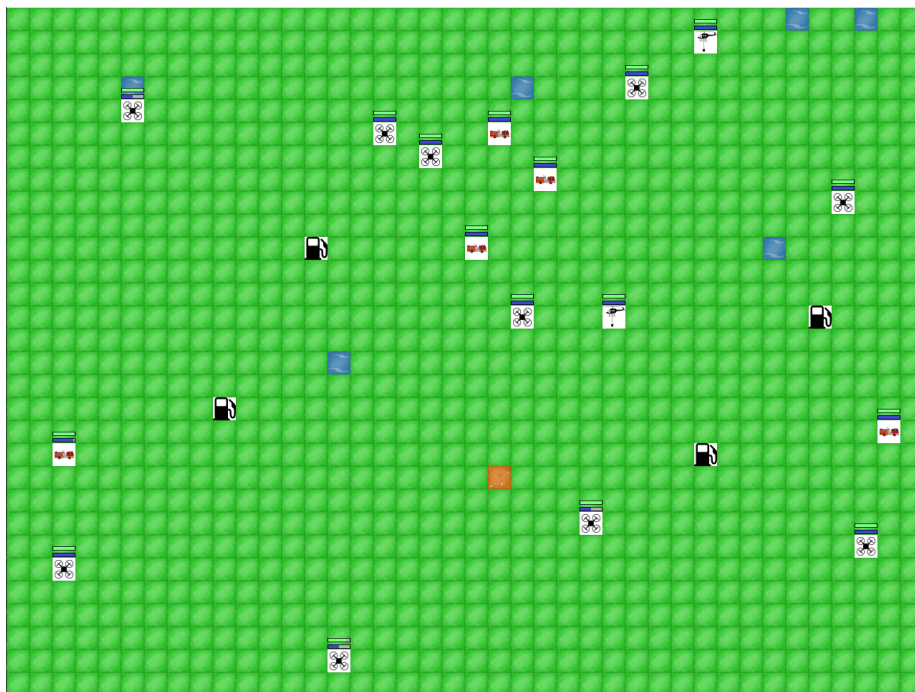


Figura 7: Interface Gráfica resultante do trabalho realizado

6 Conclusão

Com este trabalho, o grupo adquiriu conhecimentos relativos à computação baseada em agentes. Portanto, conseguiu aplicá-los através da concepção de uma arquitetura, de forma a resolver um problema real e, posteriormente, através da implementação de um programa que ilustra todo o trabalho realizado.

Inicialmente, o grupo realizou uma pesquisa acerca do problema a resolver e verificou-se a existência, atualmente, de imensos sistemas multiagente relacionados, ou seja, com o objectivo de auxiliar o controlo do combate aos incêndios. Portanto, a análise das diferentes arquiteturas de alguns desses sistemas possibilitou ter uma noção das técnicas e das vantagens e desvantagens de cada uma.

Na primeira fase deste projeto formulou-se o sistema a produzir, através do uso de técnicas de modelação UML e da variação para agentes com a linguagem AUML. Por isso, representou-se com algum detalhe as estratégias que o grupo idealizou para a resolução com sucesso do trabalho proposto.

O grupo achou por bem simplificar os processos de negociação entre os agentes, tornando-os proativos em algumas situações e cooperativos.

Na segunda fase do projeto colocou-se em prática o definido anteriormente. Assim, a implementação relevou ter uma dificuldade acessível já que foi possível definir a estratégia numa fase anterior mas, principalmente, porque os membros do grupo se sentiam à vontade com o desenvolvimento de software em JADE, graças às aulas práticas de Agentes Inteligentes.

A maior dificuldade encontrada deveu-se ao número de agentes existentes no sistema e a computação necessária para calcular qual a forma de combate de modo mais eficiente. Para resolver essa situação, tentou-se dividir ao máximo as computações pelos agentes. Adicionalmente, devido ao facto de haver uma necessidade de reabastecimento de água e combustível, tendo cada veículo as suas capacidades e velocidade, foi necessária uma adaptação singular para cada um deles.

Um ponto de falha do sistema é a Estação Central, uma vez que a mesma é única, conforme especificação pedida. Caso se quisesse um sistema mais tolerante a falhas, construir-se-iam duas estações (ou mais), tendo uma o controlo das operações. Caso essa falhasse, o banco de dados partilhado seria acessível a qualquer outra estação que, por processos de concorrência habituais, assumiriam o comando logo que possível.

Com vista à necessidade de otimização de recursos existentes, definiu-se que o combustível deveria ser reabastecido sempre que um veículo não conseguisse andar, pelo menos, um quadrante. Esta situação assume uma distribuição não uniforme dos incêndios, meio de combate, fontes de água e fontes de combustível, tal como numa situação real. Ainda mais, só é feita uma nova recolha de água por parte de um drone e por uma aeronave quando já não possuem qualquer unidade. A justificação de tal comportamento é devida, quanto ao drone, pela baixa capacidade de transporte de água. Por outro lado, a aeronave só o faz pelo custo associado a fazer recolhas que é associado a este tipo de veículo.

Por último, o grupo decidiu fazer experiências com JESS, que culminaram em

fazer uma biblioteca de todos os fogos existentes. Apesar de não ser introduzido como *key component* no projeto, permitiu desenvolver alguns conhecimentos e entender as particularidades destes processos de árvores em memória.

O resultado final é positivo, tendo o grupo construído a solução pedida, com alguns extras a nível de realismo (como o aumento de intensidade do fogo quando este não é tratado) e produzido uma interface gráfica de qualidade superior à habitual.

Referências

- [1] L. Zhou, X. Wu, Z. Xu, and H. Fujita, “Emergency decision making for natural disasters: An overview,” mar 2018.
- [2] “Incêndio florestal de Pedrógão Grande em 2017.” <http://www.mycbr-project.org/index.html>, 2017. Acedido em: 08/10/2019.
- [3] “Disaster Response Planning Using Agent-Based Simulation – AnyLogic Simulation Software.” <https://www.anylogic.com/disaster-response-applications-using-agent-based-modeling/>. Acedido em: 08/10/2019.
- [4] F. Fiedrich and P. Burghardt, “Agent-based systems for disaster management,” *Communications of the ACM*, vol. 50, no. 3, pp. 41–42, 2007.
- [5] G. I. Hawe, G. Coates, D. T. Wilson, and R. S. Crouch, “Agent-based simulation of emergency response to plan the allocation of resources for a hypothetical two-site major incident,” *Engineering Applications of Artificial Intelligence*, vol. 46, pp. 336–345, nov 2015.
- [6] “Multiagent Systems in Automotive Applications.” <https://www.intechopen.com/books/multi-agent-systems/multiagent-systems-in-automotive-applications>. Acedido em: 27/10/2019.
- [7] F. Derakhshan and S. Yousefi, “A review on the applications of multiagent systems in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 15, may 2019.