



**Universidade do Minho**  
Escola de Engenharia

José António da Cunha e Silva

**Desempenho de tarefas num cenário virtual  
dotado de características emocionais**

Dezembro, 2008



**Universidade do Minho**

Escola de Engenharia

José António da Cunha e Silva

**Desempenho de tarefas num cenário virtual  
dotado de características emocionais**

Tese de Mestrado em Informática

Trabalho efectuado sob a orientação do  
**Professor César Analide**

Dezembro, 2008

É AUTORIZADA A REPRODUÇÃO INTEGRAL DESTA TESE APENAS PARA EFEITOS DE INVESTIGAÇÃO, MEDIANTE DECLARAÇÃO ESCRITA DO INTERESSADO, QUE A TAL SE COMPROMETE;

Universidade do Minho, \_\_\_/\_\_\_/\_\_\_\_\_

Assinatura: \_\_\_\_\_

# Agradecimentos

Aproveito este capítulo para agradecer a todos os intervenientes, a oportunidade concedida para realizar este Mestrado. Agradeço em particular ao professor Cesar Analide pelo esforço em acompanhar o meu trabalho e pelos conselhos dados para o aperfeiçoar, e à empresa *Alert Life Sciences Computing, SA* pelo tempo disponibilizado para poder realizar este projecto.



## Resumo

O trabalho relatado neste documento insere-se na área científica da Inteligência Artificial, mais especificamente no contexto do desenvolvimento de sistemas multi-agente e de agentes sintéticos. No âmbito dos sistemas multi-agente e dos agentes inteligentes, pretende-se contribuir com desenvolvimentos que permitam construir agentes sintéticos "humanizados", no sentido em que se procurará focar estes desenvolvimentos na disponibilização de uma componente emocional no comportamento destas entidades computacionais. O caso de estudo deste projecto, passa por implementar um jogo de guerra *multiplayer*, denominado de *Capture The Flag*, em que cada um dos jogadores é um agente integrado numa equipa.

Esta dissertação descreve as diferentes fases do projecto, desde a investigação até à conceptualização e implementação. Podem ser especificadas as seguintes fases: definição dos objectivos do projecto, estudo dos fundamentos teóricos relacionados com sistemas multi-agente, pesquisa de sistemas e modelos semelhantes ao que é pretendido implementar, definição da arquitectura e implementação do sistema.

Os objectivos definidos para o projecto consistem essencialmente na caracterização do sistema a implementar, isto é, um sistema modular, constituído por um ambiente independente dos agentes, com um protocolo de comunicação transparente e genérico. Por sua vez, os agentes devem ter as suas tarefas bem definidas e um modelo de gestão de emoções associado, capaz de gerar, manter e dissipar todas as emoções consideradas.

Ao longo do projecto são abordados vários conceitos relacionados com agentes sintéticos, sistemas multi-agente cooperativos e modelação emocional. Estes conceitos focam-se essencialmente em características que conferem a um agente sintético maior ou menor qualidade ou credibilidade: autonomia, reactividade, pró-actividade, sociabilidade, aprendizagem e emotividade.

Para além de ter como objectivo a investigação de sistemas baseados em agentes emocionais, este projecto pretende construir uma base que sirva de estudo a outros aspectos como o trabalho em equipa, definição de estratégias e nível de inteligência revelado. Estes pontos também serão avaliados ao longo desta dissertação. O objectivo principal será então avaliar o modelo emocional implementado, através do nível de imprevisibilidade que este transmite ao sistema.

**Palavras-Chave:** Sistema Multi-agente Cooperativo, Agente Sintético, Modelo Emocional, *Soar*.

## Abstract

The project here presented is related to the scientific area of Artificial Intelligence, more specifically on modelling multi-agent systems and synthetic agents. The main purpose is to contribute with developments that allow the construction of "humanized" synthetic agents. These developments will focus on displaying an emotional component in the behavior of these computational entities. The project case study is the implementation of a multiplayer war game, named *Capture The Flag*, in which all players are agents integrated in a team.

This dissertation describes all project stages, since investigation to conceptualization and implementation. The following stages can be specified: defining all project objectives, studying the theoretical notions related to multi-agent systems, researching similar projects and models, defining the architecture and implementing the system.

The objectives defined to this project basically characterize the system to implement, that is, a modular system, with an environment completely separated from the agents, and a transparent and generic communication protocol. The agents have their tasks well defined and an associated emotion management model, capable of generate, maintain and dissipate all considered emotions.

Along the project, several concepts related to synthetic agents, multi-agent systems and emotional modelling are approached. These concepts focus essentially on features that provide more or less quality or credibility to a synthetic agent: autonomy, reactivity, pro-activity, sociability, learning and emotion.

Besides having the goal of investigating systems based in emotional agents, this project is designed to provide a study base to other aspects, including team work, defining strategies and intelligence level demonstrated. These aspects will also be assessed during this dissertation. The main goal will be to assess the implemented emotional model, through the level of unpredictability transmitted to the system.

**Keywords:** Cooperative Multi-agent System, Synthetic Agent, Emotional Model, *Soar*.

# Conteúdo

Lista de Figuras	iii
Lista de Tabelas	iv
<b>1 Introdução</b>	<b>1</b>
1.1 Motivações . . . . .	2
1.2 Objectivos . . . . .	2
1.3 Organização do documento . . . . .	3
<b>2 Contextualização</b>	<b>4</b>
2.1 Agentes sintéticos . . . . .	4
2.2 Sistemas multi-agente cooperativos . . . . .	6
2.3 Modelo emocional . . . . .	7
<b>3 Estado da Arte</b>	<b>9</b>
3.1 Gamebots . . . . .	9
3.2 Modelo <i>Cathexis</i> . . . . .	12
3.3 Microsoft XNA Game Studio . . . . .	15
3.4 Soar . . . . .	16
<b>4 Trabalho Realizado</b>	<b>21</b>
4.1 Jogo <i>Capture the Flag</i> . . . . .	21
4.2 Arquitectura do sistema . . . . .	23
4.3 Implementação dos agentes . . . . .	27
4.4 Modelo emocional . . . . .	29
<b>5 Resultados</b>	<b>33</b>
<b>6 Conclusões e Trabalho Futuro</b>	<b>39</b>
Bibliografia	41





# Lista de Figuras

2.1	Sistema multi-agente. . . . .	6
3.1	Arquitectura do sistema <i>Gamebots</i> . . . . .	11
3.2	Estrutura genérica da arquitectura do Modelo <i>Cathexis</i> . . . . .	13
3.3	Estrutura e funcionamento de um sistema emocional. . . . .	13
3.4	Estrutura de um domínio de problemas no <i>Soar</i> . . . . .	17
3.5	Estrutura da memória no <i>Soar</i> . . . . .	18
4.1	Cenário implementado para o sistema. . . . .	22
4.2	Arquitectura do sistema. . . . .	23
4.3	Protocolo de comunicação do sistema. . . . .	26
4.4	Diagrama de classes do sistema. . . . .	27
4.5	Estrutura das características dos agentes no <i>Soar</i> . . . . .	29
4.6	Exemplo de um jogo a decorrer. . . . .	30
5.1	Características da base construída. . . . .	34
5.2	Alterações possíveis ao sistema implementado. . . . .	35
5.3	Domínio de problemas do agente espião. . . . .	37
5.4	Exemplo de um jogo a decorrer com o modelo emocional activo. . . . .	38

# Lista de Tabelas

4.1	Cálculo da intensidade dos sistemas emocionais. . . . .	32
4.2	Associação entre emoções e comportamentos. . . . .	32



# Capítulo 1

## Introdução

A área de Inteligência Artificial e o estudo de agentes sintéticos tem sido alvo de um interesse crescente. Mais do que estudar o comportamento de agentes implementados, com base em regras pré-estabelecidas, esta dissertação pretende avaliar as diversas possibilidades para conferir a um sistema de agentes cooperativos, um nível aceitável de imprevisibilidade. Para objectivar este nível de imprevisibilidade, podem ser estabelecidos à partida critérios de avaliação ao sistema a implementar:

- Possibilidades de decisão face a uma determinada situação. O observador não deve ser capaz de prever facilmente a acção a tomar por parte do agente, face a um determinado obstáculo ou situação;
- Interpretação do estado global do sistema por parte do observador. Num cenário em que todos os agentes possuem tarefas e objectivos bem definidos, não deve ser possível, através da experiência de observação, prever a tendência de comportamentos ou acontecimentos do sistema.
- Linearidade da relação objectivos-acções. Se o observador tiver a noção à partida, da tarefa a desempenhar pelo agente, pode tentar prever através do seu senso comum, o conjunto de acções necessárias para o cumprimento da mesma. A introdução de outros factores nesta relação pode dificultar esta previsão, principalmente se estes factores impliquem a alteração constante do plano de acções do agente.

Face a estes critérios, é necessário encontrar um modelo de implementação, que permita diversificar as acções a tomar pelos agentes e dificultar a previsão do *output* global do sistema. Para isto, podem ser encontradas soluções, através do estudo do comportamento humano. Pode-se facilmente constatar que a imprevisibilidade do comportamento humano deve-se em grande parte ao estado emocional do indivíduo. Uma determinada emoção tanto pode originar comportamentos inesperados, como influenciar os comportamentos que se esperavam, face a determinada situação. Para além disto, tanto as decisões tomadas como as consequências das mesmas, podem originar novas emoções que influenciem os comportamentos futuros.

## 1.1 Motivações

A principal motivação deste projecto, passa então por implementar um sistema que contenha agentes que consigam superar as expectativas do observador, face aos critérios definidos anteriormente. A abordagem a seguir tem em consideração as manifestações emocionais do agente, ou seja, pretende-se construir um modelo emocional que possa ser integrado na implementação dos agentes sintéticos.

Mais do que implementar um modelo emocional para os agentes, este projecto pretende analisar e considerar os critérios de qualidade de um agente sintético. Conceitos como a autonomia, capacidade de aprendizagem e reactividade têm de ser integrados nos agentes implementados, uma vez que deixa de ser pertinente o estudo de outros conceitos, caso os agentes não tenham uma complexidade capaz de reagir a todos os acontecimentos definidos no cenário onde estão incluídos.

Existem também motivações relacionadas com a arquitectura do sistema. A implementação de um sistema multi-agente modular, permite que sirva como base de investigação e implementação em contexto de comunidade. Isto significa que, conhecendo as regras de comunicação entre o ambiente e os agentes, e as acções possíveis do agente, deverá ser possível a implementação sem qualquer dependência de tecnologias ou estruturas.

## 1.2 Objectivos

Para a implementação do sistema multi-agente desta dissertação, foram delineados os seguintes objectivos:

- Construção de um cenário que sirva como base de acção para os diferentes agentes implementados. Este cenário deve ser concebido, de forma a disponibilizar os obstáculos e objectivos que vão influenciar as decisões dos agentes;
- Definição das regras do cenário concebido. Estas regras englobam, por exemplo, a definição de tipos e equipas de agentes, a forma como as acções dos mesmos alteram o estado do cenário e o motor físico (velocidade de deslocação dos agentes, interacção entre objectos, etc.). Estas regras são importantes na medida em que regem as expectativas do observador e a comunicação entre o ambiente e a componente de suporte à decisão dos agentes;
- Definição da arquitectura do sistema. Este ponto é bastante importante, considerando os objectivos pretendidos para o mesmo. A ligação dos agentes ao cenário deve ser modular e a comunicação entre o cenário e os agentes deve ser feita de uma forma transparente e genérica, ou seja, a informação sobre o ambiente deve ser passada a todos os agentes, sendo que são estes que se encarregam de a filtrar, consoante as suas necessidades. Da mesma forma, os agentes devem ter a sua componente de suporte à decisão bem definida, de forma a ser facilmente alterada ou mesmo substituída;

- Implementação dos agentes. Este ponto também é importante, na medida em que a capacidade de interpretação do estado do sistema e de resposta aos desafios impostos, vai influenciar a qualidade do projecto, enquanto base de investigação ao comportamento de agentes sintéticos;
- Construção do modelo emocional. A forma como as emoções são geradas, mantidas e dissipadas deve ser a mais realista possível. Para isto, deve ser concebido um modelo que, para além de englobar as emoções que normalmente influenciam o comportamento humano, deve fornecer ao observador uma percepção de que as emoções são algo subtil ao agente e não algo imposto ao sistema e que apenas traga "ruído" ao real objectivo do projecto.

### 1.3 Organização do documento

Este documento regista todos os acontecimentos relevantes decorridos durante a realização do projecto. Durante os 6 capítulos aqui apresentados, é feita uma contextualização, tendo em conta os principais conceitos teóricos envolvidos no projecto, passando depois para uma visão mais específica, enunciando tecnologias e modelos usados. A parte seguinte apresenta uma abordagem mais prática, através da descrição do projecto implementado e principais desafios encontrados e resultados produzidos. O documento possui então os seguintes capítulos:

- Contextualização - apresentação dos conceitos base do projecto;
- Estado da Arte - descrição das tecnologias abordadas durante o projecto e razões que levaram à sua utilização;
- Trabalho Realizado - descrição do sistema implementado. Para além das principais decisões tomadas, é apresentada a arquitectura da aplicação e a aplicação dos conceitos teóricos mencionados anteriormente;
- Resultados - demonstração das potencialidades do sistema implementado e comparação com os objectivos inicialmente propostos;
- Conclusões e trabalho futuro - notas finais após a conclusão do projecto. São também enumeradas as possíveis melhorias e caminhos a percorrer com o objectivo de melhorar o projecto.

## Capítulo 2

# Contextualização

Neste capítulo, é descrito em pormenor o enquadramento desta dissertação e os fundamentos teóricos que servem de base ao foco de investigação e implementação do projecto. Mais concretamente, é feita uma definição detalhada de um agente sintético, os paradigmas associados a este modelo computacional e a evolução do mesmo. Passamos depois para as motivações e objectivos da integração de modelos emocionais nestes agentes, introduzindo também conceitos como a personalidade e a diversidade comportamental.

### 2.1 Agentes sintéticos

Um agente sintético pode ser descrito como uma personagem independente e interactiva, que age de uma forma credível [3]. Sendo esta uma definição de um agente sintético ideal, facilmente se conclui que estas três características são as principais formas de avaliar a sua qualidade. Para perceber o grau de influência que cada uma destas características apresenta, é necessário compreender o funcionamento de um modelo baseado em agentes.

O modelo computacional baseado em agentes foi um ponto revolucionário para a área de Inteligência Artificial, pois possibilitou o estudo e a modelação de sistemas computacionais que simulassem entidades dotadas de inteligência, mais precisamente, a capacidade de tomar decisões autónomas. Uma entidade pode ser considerada um agente, caso reúna as seguintes características [7]:

- Capacidade de resolução de um determinado domínio de problemas, com limites de acção bem definidos;
- Integração num determinado ambiente. O ponto em comum passa pela ligação ao ambiente, através da recepção de *inputs* e interferência no mesmo, através do envio de *outputs*;
- Definição clara dos objectivos. Um agente é sempre concebido para realizar uma determinada tarefa;
- Autonomia. O agente tem sempre controlo sobre o seu estado e comportamento;
- Comportamento flexível e dinâmico. O agente deve ser capaz de responder prontamente a alterações no ambiente e antecipar objectivos futuros.



Aprofundando as propriedades que caracterizam um agente, podem ser definidas duas categorias: noção forte e noção fraca de agente. A noção fraca diz respeito ao conjunto mínimo de características[9]:

- Autonomia: conseguem operar sem intervenção de outros agentes;
- Reactividade: detectam os eventos que ocorrem no ambiente e respondem adequadamente a qualquer mudança no mesmo;
- Pró-Actividade: tomam a iniciativa de realizar as suas próprias acções, influenciadas pelos seus objectivos;
- Sociabilidade: relacionam-se com outros agentes durante a resolução dos problemas. Esta relação pode ser de comunicação, competição ou cooperação.

A noção forte caracteriza os agentes que apresentam um conjunto de mais-valias:

- Mobilidade: capacidade de se movimentarem através da rede formada pelos restantes agentes, por forma a executar as suas tarefas;
- Intencionalidade: definição de objectivos e planeamento de estratégias para os atingir;
- Aprendizagem: capacidade de adquirir conhecimento. Este conhecimento pode ter origem em detecção de padrões de comportamento ou de preferências;
- Competência: eficiência e sucesso na execução das suas tarefas;
- Veracidade: os agentes não fornecem informação falsa de forma intencional;
- Racionalidade: capacidade em recusar tarefas impossíveis de executar, contraditórias com os seus princípios ou quando não são compensados em termos do risco, custo ou esforço;
- Benevolência: aceitam ou adoptam objectivos de terceiros desde que estes não entrem em conflito com os seus princípios;
- Emotividade: integram características próprias do ser humano, tais como as emoções.

No âmbito deste projecto, existem algumas características da noção forte de agente, que serão abordadas e mesmo implementadas. A característica que assume maior importância é a emotividade, uma vez que é o que se pretende com a construção do modelo emocional dos agentes, para atingir o objectivo principal do projecto, isto é, a imprevisibilidade do sistema. Outras características que podem ser referidas são a aprendizagem e a competência.

## 2.2 Sistemas multi-agente cooperativos

Como se pode constatar, a utilidade do modelo computacional baseado em agentes vai muito além da implementação de agentes inteligentes, sendo cada vez mais usado em Sistemas Distribuídos. A figura 2.1 representa um esquema genérico de um modelo baseado em agentes. Analisando-o, podemos observar as características comuns a um sistema deste género: a integração num determinado ambiente e a interacção entre agentes, para a resolução de dependências associadas às suas tarefas. A estas interacções podem estar associados protocolos, não só para a criação de grupos sociais, mas também para estruturar o comportamento colectivo do sistema.

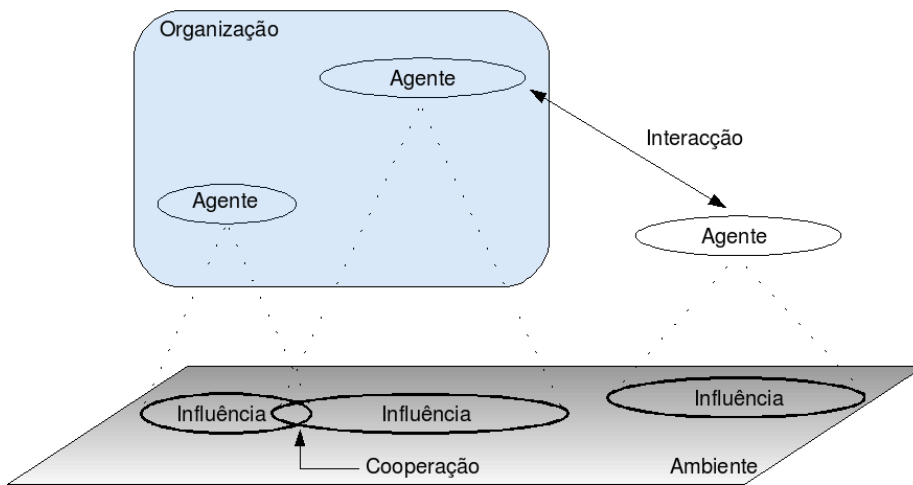


Figura 2.1: Sistema multi-agente.

A forma como é implementada a cooperação entre os agentes é fundamental para o sistema, uma vez que interfere com duas características essenciais acerca do nível de conhecimento adquirido: a exactidão dos dados recebidos e a quantidade de informação recolhida. Uma vez que torna-se incomportável o processamento de toda a informação do sistema, é necessário adoptar a melhor estratégia, com base no objectivo global. Podem ser categorizados dois tipos de estratégia de aquisição e monitorização de informação acerca do estado dos elementos do sistema [4]:

- Esquema centralizado: utilização de algoritmos complexos para centralizar num único agente, toda a informação dos restantes, sendo que esse agente tem exclusivamente a função de monitorizar o sistema. É um esquema que garante um elevado grau de conhecimento, mas limita a exactidão do mesmo. Para além disto, exige também elevado poder de processamento do agente responsável por este processo;
- Esquema distribuído: o processamento de conhecimento sobre o estado do sistema é feito pelos vários agentes. O facto de cada agente apenas processar informação considerada relevante, faz com que o seu conhecimento global seja incompleto. No entanto, como o seu esforço é concentrado na

informação a que consegue aceder, é possível detectar situações relevantes com maior exactidão.

As duas estratégias apresentam as suas vantagens e desvantagens. No enquadramento do projecto, o objectivo passará por assumir um compromisso entre as duas, com vista a balancear os aspectos positivos e negativos.

Ao fazer uma retrospectiva dos pontos essenciais de um modelo baseado em agentes, permanece a ideia de um modelo bastante previsível, ou seja, os agentes são implementados para realizar uma determinada tarefa, que deverá ser completada, independentemente das adversidades encontradas. Como já foi explicado, o objectivo do projecto passa precisamente pelo afastamento desta previsibilidade. Na próxima secção, são apresentados os conceitos e estratégias a usar para atingir este objectivo.

## 2.3 Modelo emocional

Nesta secção abordamos outro conceito importante, associado a um sistema baseado em agentes inteligentes: o realismo do seu comportamento. Isto é avaliado, não só pela credibilidade transmitida ao observador, mas também pelas expectativas do mesmo, uma vez que este irá fazer uma avaliação subjectiva com base nos seus modelos. Estes modelos consistem num conjunto de noções que descrevem o comportamento humano, mais concretamente a sua diversidade, e que se resumem a dois pontos essenciais: a personalidade e as emoções. Estes pontos estão directamente ligados, na medida em que a personalidade influencia a forma como são geridas as emoções e a emoção desencadeada, em conjunto com a personalidade, define o comportamento do agente, face a determinada situação.

Na construção de um modelo emocional associado a um agente inteligente, podem ser encontrados quatro pontos em comum [3]:

- Mecanismo de interpretação do *input*: o agente possui sensores que captam os estímulos internos (fome, cansaço, etc.) e externos (acontecimentos decorridos no ambiente ou interações com outros agentes). Estes estímulos são usados para actualizar o seu estado emocional, em conjunto com o estado actual e a personalidade;
- Modelo computacional de gestão das emoções: define a forma como o estado emocional do agente é actualizado, ou seja, quais os critérios que activam uma determinada emoção e qual a sua duração e intensidade ao longo do tempo. Este é o ponto central de qualquer modelo emocional, onde a personalidade desempenha um papel fundamental, uma vez que permite que a mesma emoção e as mesmas situações levem a comportamentos diferentes. Por estas razões, os modelos emocionais derivam maioritariamente de teorias de psicologia;
- Relação entre o estado emocional e as acções do agente: uma vez que o observador não é capaz de aceder ao estado emocional do agente, este tem de ser reflectido nas acções tomadas. Podem ser adoptadas as estratégias de *scripting* ou planeamento. Na estratégia de *scripting*, é definida uma listagem de possíveis acções e respectivo estado emocional necessário para as

desencadear. A estratégia de planeamento é mais complexa, uma vez que inclui no modelo emocional, noções como o sucesso no cumprimento de objectivos, existência de recompensas, e associação entre estados emocionais e prioridade dos objectivos a atingir. Como noutros pontos já mencionados, este também vai ser implementado no projecto, com o objectivo de balancear as duas estratégias;

- Expressão das emoções para o exterior: este aspecto do modelo é bastante importante, na medida em que permite a transmissão ao observador, do realismo desejado ao sistema de agentes emocionais. Mais do que seleccionar as acções que um agente deverá tomar com um determinado estado emocional, é necessário que essas mesmas acções sejam executadas no exterior de forma a reflectir esse estado emocional. Uma vez que este ponto está mais relacionado com técnicas gráficas e de animação, não será tão abordado como os restantes.

Ao analisar todos os componentes que constituem um modelo emocional, podemos caracteriza-lo pelo nível de pormenor atingido, ao simular as emoções e a personalidade de um ser humano. Uma abordagem mais superficial limita-se apenas a relacionar emoções com comportamentos e estímulos, enquanto que uma abordagem mais específica tenta simular os processos que constituem a actividade cerebral. Existem também outros componentes que podem ser integrados num modelo emocional, tais como a interacção entre agentes, em que são formulados modelos de outros agentes, que podem posteriormente ser usados para aprendizagem e escolha de novos comportamentos.

O modelo implementado terá estas estas noções como base e irá obrigatoriamente focar-se em alguns componentes em detrimento de outros.

## Capítulo 3

# Estado da Arte

Neste capítulo, estão listadas as principais fontes de ideias e informação que influenciaram a progressão do projecto. É feita uma divisão entre os modelos que inspiraram a arquitectura e concepção, e as ferramentas que permitiram a implementação de todos os conceitos delineados.

### 3.1 Gamebots

O jogo *Gamebots* foi um ponto de partida para a realização deste projecto, uma vez que partilha de alguns dos objectivos e conceitos definidos. Trata-se de um jogo *multiplayer* e multi-agente, ou seja, permite tanto a participação de jogadores humanos, como de agentes implementados. Mais do que um jogo, este projecto pretende ser uma base de investigação para sistemas multi-agente, distinguindo-se de outros sistemas, através das seguintes características [5]:

- Suporte para múltiplas tarefas, através da definição de vários tipos de jogos, pondo à prova as capacidade dos agentes implementados se adaptarem a novos desafios;
- Sistema extensível através do seu carácter modular, permitindo a inserção de novos jogos e desafios, suportando assim a investigação a longo prazo;
- Suporte para múltiplos cenários. Isto significa que é possível alternar entre vários ambientes onde o agente pode ser integrado;
- A possibilidade de interacção entre agentes humanos e agentes sintéticos, permite o estudo comparativo sobre a forma como os problemas são resolvidos pelos diferentes agentes. Para além disto é também possível estudar novas formas de competição e colaboração.

O projecto *Gamebots* revela-se mais ambicioso e abrangente do que o projecto descrito nesta dissertação. O projecto a implementar pretende essencialmente estudar a interferência emocional no comportamento dos agentes, ou seja, tem objectivos mais específicos e apenas se irá focar num único tipo de jogo. No entanto, podem ser encontrados objectivos e linhas de orientação em comum, que levaram a que o *Gamebots* tenha sido analisado para estruturar as bases do projecto implementado:

- A construção de um jogo com diferentes equipas, para a investigação de agentes sintéticos. Esta é uma estratégia cada vez mais usada e que se revela bastante útil para o desenvolvimento de agentes inteligentes. Para além do *Gamebots*, também pode ser referido o torneio *Robocup* [6], um dos mais populares na comunidade de investigação em agentes sintéticos inteligentes, devido às suas potencialidades no estudo de novos modelos de trabalho em equipa, representação de hierarquias de planeamento e tarefas, e interacção entre agentes;
- Implementação modular do projecto, com o objectivo de o tornar extensível. Um sistema modular permite a fácil substituição dos seus componentes (um novo cenário ou um novo tipo de agente), bem como uma maior facilidade em disponibilizar o sistema para a comunidade;
- A definição de agentes com tarefas específicas e um papel definido na equipa onde está integrado. Apesar do projecto descrito nesta dissertação apenas se focar num tipo de jogo, existem vários tipos de agentes, cada um com funções bem definidas. Esta variedade terá de ser implementada por forma a ser facilmente estendida (novas tarefas, novas funções ou novos tipos de agentes);
- Possibilidade de investigação de várias características de um sistema multi-agente. O facto deste projecto se focar num ponto específico, não deve limitar a possibilidade de análise dos restantes. Mesmo com a construção de um modelo emocional e com a formulação de conclusões que realcem as consequências desse modelo, o projecto deverá permitir extrair análises diversas, tais como a cooperação entre os elementos de uma equipa, o nível de competitividade atingido e a eficiência conseguida na resolução dos desafios e tarefas apresentadas.

A figura 3.1 representa a arquitectura definida para o jogo *Gamebots* [5]. Foi definido um modelo cliente-servidor que permite o acesso remoto por parte dos diferentes clientes (tanto jogadores como agentes). De seguida são descritos em pormenor os diferentes componentes da arquitectura, uma vez que, de uma forma genérica, estes assemelham-se aos usados neste projecto.

### Motor gráfico

O jogo *Gamebots* usa o motor gráfico do jogo *Unreal Tournament*, um *First Person Shooter* em 3D, com um motor físico integrado e com uma variedade de ferramentas, sendo que a percepção do mundo virtual por parte do jogador está limitada à sua localização. São disponibilizados três tipos de jogos neste motor, todos estes implementados no *Gamebots*:

- *DeathMatch*: jogo com tempo limite em que o jogador tenta obter o máximo de pontos através, eliminando os jogadores adversários;
- *Capture The Flag*: duas ou mais equipas tentam capturar a bandeira em território inimigo e trazê-la de volta para a sua base;
- *Domination*: os jogadores tentam obter o domínio do terreno, através do controlo de *domination points*.

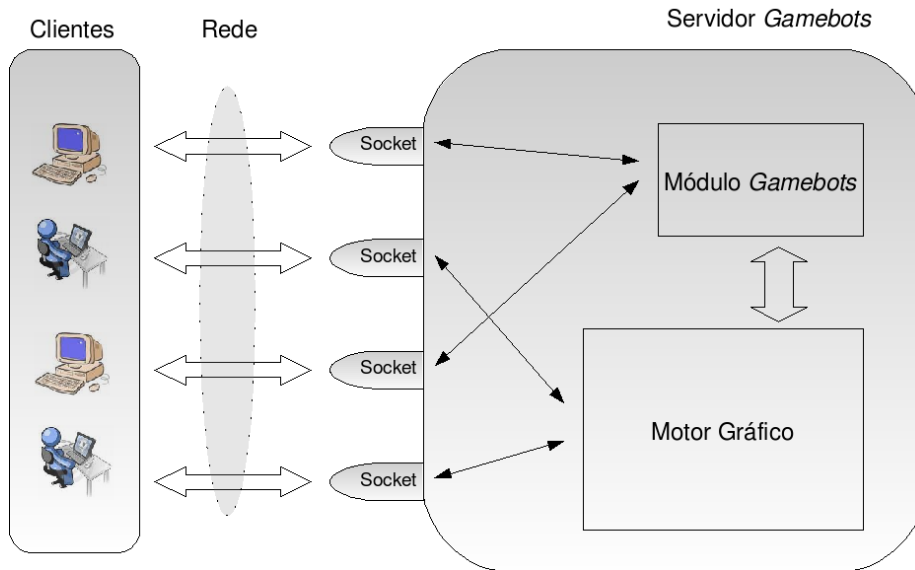


Figura 3.1: Arquitectura do sistema *Gamebots*.

Uma das características mais apelativas do uso deste motor gráfico, passa pela disponibilização da linguagem *UnrealScript*: uma ferramenta que permite aos programadores, controlar a lógica do jogo, a interacção entre objectos, adicionar e remover objectos do cenário e ajustar parâmetros do motor físico. Finalmente, outra grande vantagem do uso do motor do *Unreal Tournament* é o seu suporte multi-plataforma que possibilita a fácil integração com sistemas já existentes.

### Módulo *Gamebots*

O módulo *Gamebots* é integrado no motor gráfico para realizar três importantes tarefas: providenciar informação sensorial ao agente, decodificar as acções dos agentes no cenário e providenciar a comunicação entre jogadores ou agentes. Todas estas tarefas em conjunto, formam o protocolo de interacção entre os agentes.

O sistema comunica com os agentes através de três tipos de mensagem: estabelecimento da conexão, mensagens síncronas e mensagens assíncronas. O estabelecimento de conexão permite ligar o agente ao servidor através de um *handshake*, onde é anunciado o tipo de jogo e as condições de vitória. As mensagens síncronas permitem enviar ao agente num período regular de tempo, toda a informação sobre o ambiente a que este pode ter acesso, bem como o seu próprio estado. As mensagens assíncronas permitem informar o agente de acontecimentos relevantes no cenário que não podem ficar pendentes até ao envio da mensagem síncrona (encontro com um inimigo ou colisão com uma parede). Todas as mensagens têm uma estrutura bem definida, para facilitar e uniformizar a implementação dos agentes.

As acções do agente são enviadas, através da mesma estrutura usada pelo servidor. As mensagens disponíveis são STOP, JUMP, RUNTO e CHANGEWE-

APON e é possível indicar ao servidor a intenção de se deslocar para um determinado objecto ou jogador para simplificar o processo de decisão e diminuir o número de mensagens enviadas. Como foi dito anteriormente, a interacção entre jogadores e agentes também é objecto de estudo e por esta razão, é possível enviar mensagens pré-definidas entre estes, com o objectivo de aplicar ou alterar a tática de uma equipa. As mensagens “Defend the base”, “Hold your position” e “Cover me” são alguns destes exemplos.

### Cliente

Devido ao seu carácter modular e à sua arquitectura cliente-servidor com um protocolo de comunicação bem estruturado, o *Gamebots* permite que a implementação do agente seja independente da plataforma ou ferramenta usada. Alguns exemplos de agentes, incluem implementações em Java, Soar, TCL, C++ e PHP. São também disponibilizadas *frameworks* para implementação, com o objectivo de, não só abstrair-se de pormenores mais complexos, como o protocolo de comunicação, mas também providenciar ferramentas de *debug*. Um exemplo disto é o *Javabot* que inclui uma API para implementação, uma API para visualização do cliente e uma aplicação para executar e gerir os clientes.

## 3.2 Modelo *Cathexis*

O Modelo *Cathexis* foi o modelo estudado para formular o modelo emocional usado neste projecto. É um modelo que procurou inspiração em várias áreas como a Psicologia, Etologia e Neurobiologia, para tentar reproduzir um modelo que se aproximasse o máximo possível da forma como são geridas as emoções nos seres humanos. Neste modelo, todas as emoções e temperamentos são estruturados numa rede composta por sistemas emocionais. Cada um destes sistemas é responsável pela gestão de uma família de emoções e consequentes processos.

### Estrutura

O diagrama da figura 3.2 representa de uma forma genérica, a arquitectura do Modelo *Cathexis* [10]. O sistema de geração de emoções é composto pelos sistemas emocionais referidos anteriormente, que através dos estímulos externos (acontecimentos no ambiente) e internos (nível de motivação, processos biológicos, etc.) detectados, são responsáveis por detectar as condições ideais para desencadear ou inibir a emoção associada ao respectivo sistema. Finalmente, cada sistema emocional informa o sistema comportamental acerca do seu estado, que por sua vez selecciona a acção apropriada.

A figura 3.3 mostra em detalhe a estrutura de um sistema emocional, mais concretamente os seguintes componentes:

- Quatro tipos de sensores: neuronais, motores, motivacionais e cognitivos. O *input* destes sensores aumenta ou diminui a intensidade das emoções a que se aplicam;
- Parâmetro  $\alpha$ : controla a activação da emoção. Assim que a intensidade atingir este valor, o sistema liberta o sinal de *output* para os outros sistemas emocionais e para o sistema comportamental;



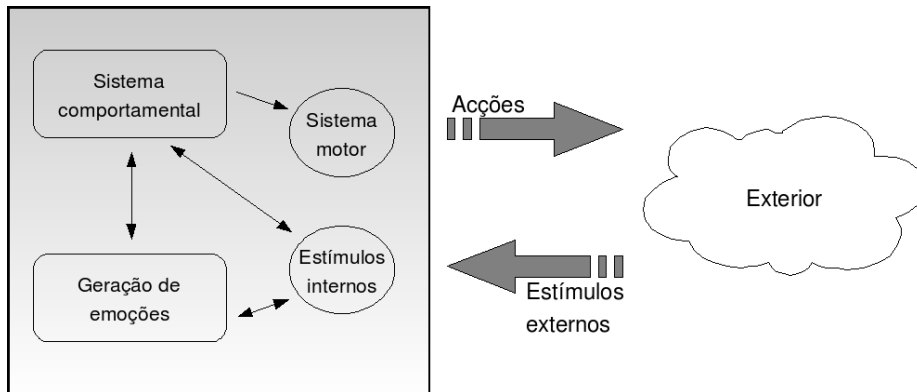


Figura 3.2: Estrutura genérica da arquitectura do Modelo *Cathexis*.

- Parâmetro  $\omega$ : especifica o valor de saturação do sistema emocional, ou seja, o valor máximo para o parâmetro  $\alpha$ ;
- Função de decaimento  $f()$ : responsável pela atenuação do valor da intensidade no sistema emocional, isto é, controla a duração da emoção ao longo do tempo;
- Interação entre sistemas emocionais: assim que a intensidade de um sistema emocional atinge o seu parâmetro  $\alpha$ , este pode produzir um efeito excitatório ou inibitório noutros sistemas emocionais (por exemplo, o medo inibe a alegria e a tristeza pode desencadear a raiva).

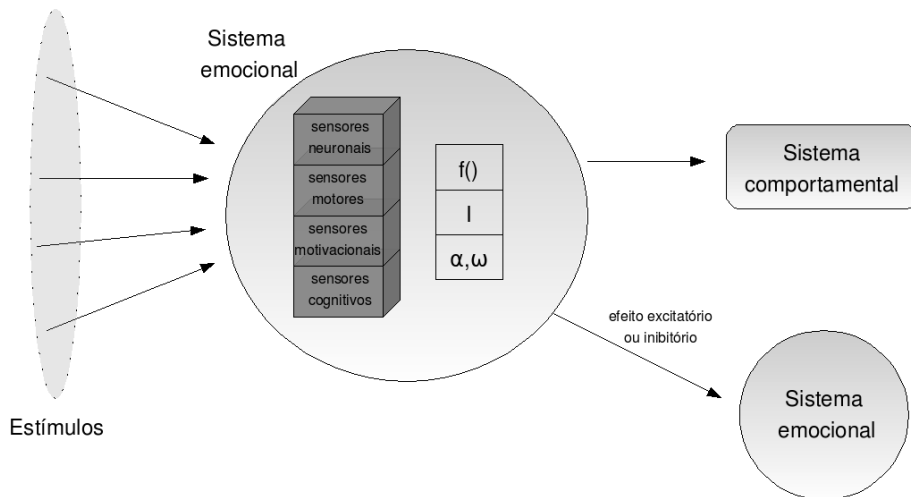


Figura 3.3: Estrutura e funcionamento de um sistema emocional.

### Conjunto de emoções

Outro ponto importante num modelo emocional, passa por definir quais as emoções básicas que vão ser consideradas, uma vez que não existe um consenso acerca da definição deste conceito. No Modelo *Cathexis*, é assumido que as emoções básicas existentes definem-se como um conjunto de emoções completamente distintas entre si e que evoluíram para nos preparar para as várias tarefas do quotidiano como o amor, o perigo e a perda. É também assumido que cada emoção básica é de facto uma família de emoções que partilham determinadas características entre si. Pode-se então concluir que cada sistema emocional representa uma emoção básica e no Modelo *Cathexis* existem seis sistemas emocionais: Raiva, Medo, Alegria, Tristeza, Repulsa e Surpresa.

Existem ainda outros conceitos implementados no Modelo *Cathexis* para além das emoções básicas, uma vez que ao contrário de outros modelos, este não considera que todas as emoções são básicas. Ao analisar este modelo, pode-se concluir que existe a possibilidade de várias emoções serem desencadeadas com intensidades diferentes. Estas situações provocam o aparecimento de novas emoções, que podem ser definidas como uma mistura de emoções básicas ou emoções secundárias e são diferentes, consoante a intensidade presente em cada sistema emocional.

O Modelo *Cathexis* considera também uma variante das emoções: os sentimentos. Apesar de muitos modelos não fazerem distinção entre sentimentos e emoções, este define que os sentimentos são desencadeados da mesma forma que as emoções mas a níveis de excitação mais baixos, significando então que os sentimentos estão integrados no sistema emocional correspondente. Como exemplo disto, podemos considerar o sentimento de euforia que é desencadeado com um nível de excitação baixo, mas há medida que a intensidade aumenta progressivamente poderá dar lugar à emoção de alegria, quando o parâmetro  $\alpha$  for atingido. Isto leva a concluir que existe uma maior concentração de sentimentos do que emoções, o que é válido se analisarmos as emoções humanas.

### Intensidade emocional

Através dos conceitos apresentados anteriormente, pode ser concluído que a intensidade de cada emoção é obtida através do contributo de vários factores. A fórmula usada pelo Modelo *Cathexis*, para obter o valor exacto da intensidade  $I$ , no instante  $t$ , para a emoção  $e$  é a seguinte:

$$I_{et} = X(f(I_{et-1}) + \sum_k L_{ke} + \sum_l G_{le} \cdot I_{lt} - \sum_m H_{me} \cdot I_{mt}), \quad (3.1)$$

onde:

- $I_{et-1}$  é a intensidade da emoção  $e$  para o instante  $t-1$ ;
- $f()$  é a função de decaimento da emoção  $e$ ;
- $L_{ke}$  é o valor do parâmetro  $k$  de eleição da emoção  $e$ ;
- $G_{le}$  é efeito excitatório da emoção  $l$  na emoção  $e$ ;
- $H_{me}$  é efeito inibitório da emoção  $m$  na emoção  $e$ ;
- $X()$  é a função que restringe a intensidade entre 0 e o valor do parâmetro  $\omega$ .

Os parâmetros de eleição de emoções definem as condições que interferem na intensidade dos sistemas emocionais e estão directamente relacionados com os

sensores integrados nos mesmos. Sendo assim, cada sensor analisa um conjunto destes parâmetros, ou seja, o sensor neuronal analisa os factores relacionados com a actividade cerebral, o sensor motor analisa processos motores como expressões faciais ou posturas corporais, o sensor motivacional analisa o nível de motivação associado a cada emoção (influenciado por factores como a fome, sede, nível de dor ou mesmo outras emoções desencadeadas) e o sensor cognitivo inclui factores como a interpretação de acontecimentos, crenças, desejos e memória.

Da mesma forma que o valor da intensidade aumenta com os factores anteriormente mencionados, é necessário definir uma forma para as emoções se dissiparem ao longo do tempo, uma vez que um estado emocional não se mantém activo indefinidamente. No Modelo *Cathexis* é definida uma função de decaimento por cada sistema emocional, permitindo assim que a intensidade de cada emoção diminua de forma distinta. Caso não surja nenhum *input* que aumente a intensidade da emoção, a função de decaimento, continuará a diminuir a intensidade da emoção em cada ciclo, acabando esta por se dissipar.

### 3.3 Microsoft XNA Game Studio

O *Microsoft XNA Game Studio* surge como uma nova ferramenta para os programadores de jogos. Apesar de não estar directamente relacionada com a investigação de agentes sintéticos inteligentes, é importante mencioná-la no contexto deste projecto, devido às vantagens que foram tidas em conta, para a sua escolha na implementação. Trata-se de uma nova *framework* baseada na *framework* .NET que permite a criação de jogos nas plataformas *Windows* e *Xbox 360*. As suas principais características são [1]:

- *XNA Framework*: permite a criação de jogos 2D e 3D, usando a linguagem C# e um conjunto de bibliotecas que facilitam a implementação dos objectos e lógica do jogo. A API fornecida nesta *framework* permite a abstracção de pormenores mais específicos como a plataforma a que o jogo se destina;
- *XNA Game Studio Professional*: destinado a programadores profissionais, que necessitam de ferramentas adicionais para a criação de jogos comerciais;
- *XNA Framework Content Pipeline*: conjunto de ferramentas disponibilizadas com o objectivo de reduzir a curva de aprendizagem no desenvolvimento de jogos 3D. O principal problema passa por converter o diferente conteúdo digital, para um formato que possa ser usado no jogo. Para atenuar este problema, esta *framework* usa o *Visual Studio* para conceber, organizar e importar o conteúdo digital do jogo.

O uso do *Microsoft XNA Game Studio* para diminuir a complexidade de implementação de um jogo, é o ponto-chave na inclusão desta ferramenta no projecto. Para além de ser possível diminuir a curva de aprendizagem no uso da ferramenta, é também possível diminuir o tempo despendido exclusivamente nos pormenores do motor gráfico do projecto. Esta vantagem permite a concentração em pormenores mais importantes como a integração do modelo cliente-servidor no jogo e a definição do respectivo protocolo de comunicação. Para

além disto, também é possível implementar o jogo, tendo em vista outros objectivos como a estruturação do jogo para permitir a modularidade do mesmo e a fácil integração dos agentes no cenário.

### 3.4 Soar

Mais do que uma ferramenta ou uma linguagem de programação, o *Soar* pode ser definido como uma arquitectura cognitiva genérica, usada na implementação de sistemas que demonstrem um comportamento inteligente [2]. Os principais objectivos desta arquitectura são:

- Abranger toda a gama de tarefas que podem ser suportadas por um agente sintético inteligente;
- Representar e usar formas apropriadas de conhecimento (procedimentos, declarações, episódios, etc.);
- Implementar todos os métodos de resolução de problemas;
- Possibilitar a interacção com o mundo exterior, através da implementação de mecanismos de *input* e *output*;
- Aprendizagem de todos os aspectos referentes às tarefas existentes e sua execução.

Podem também ser enunciados os princípios básicos do *Soar*, que foram considerados na concepção da sua arquitectura:

- Todas as decisões tomadas, têm em conta a combinação de conhecimento relevante em tempo de execução. Isto inclui informação de *input*, problemas resolvidos anteriormente e informação retornada pela memória a longo-prazo;
- Múltiplos mecanismos de aprendizagem e múltiplas representações da memória a longo-prazo;
- Funcionalidade e performance: devem ser implementadas todas as primitivas necessárias para simular as capacidades cognitivas humanas (tomada reactiva de decisões, percepção, compreensão e raciocínio deliberado, planeamento, diferentes formas de aprendizagem, etc.). Todos estes mecanismos devem ser implementados, através de algoritmos que sejam eficientes.

#### Domínios de problemas

A arquitectura *Soar* usa a noção de domínio de problemas para estruturar os comportamentos ou decisões a tomar [8]. A figura 3.4 representa a estrutura de um domínio de problemas e demonstra as duas vantagens: engloba todas as decisões possíveis face a uma determinada situação (visão estática) e o percurso do comportamento do agente ao longo do tempo (visão dinâmica). A representação em triângulo pretende ilustrar o desdobramento de possibilidades ao longo do tempo, sendo que a progressão é feita através da alteração do estado ou descrição da situação actual do agente. O processo de decisão é iniciado através

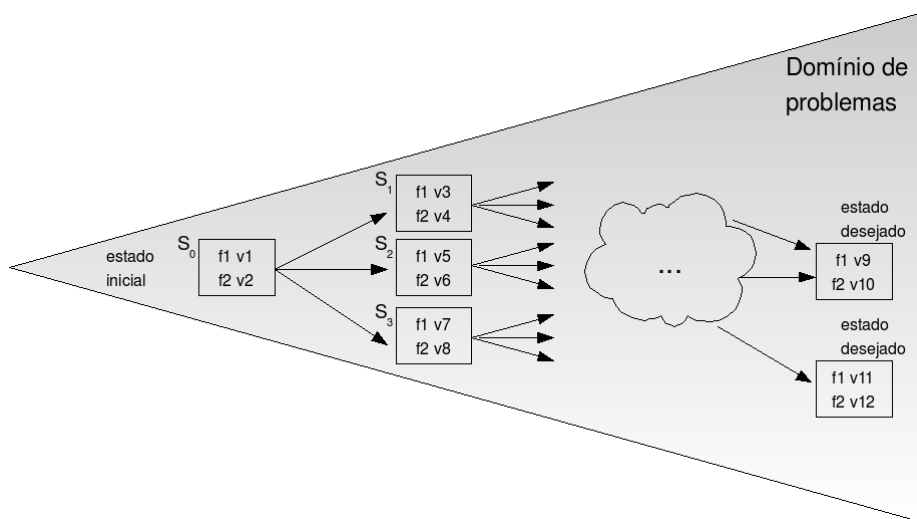


Figura 3.4: Estrutura de um domínio de problemas no *Soar*.

de um estado inicial ( $S_0$ ), onde estão definidos os valores iniciais ( $v_1$  e  $v_2$ ) para as características alteradas em cada estado ( $f_1$  e  $f_2$ ).

Ao conceber o domínio de problemas de um agente, é necessário indicar o estado inicial do mesmo e o estado ou estados desejados (representam os objectivos finais do agente). Todos os comportamentos do agente consistem no percurso ao longo destes estados, percurso este que é feito através dos operadores representados na figura 3.4. Nesta estrutura, um estado é definido como um conjunto de características e valores que reflectem a situação actual externa e interna do agente. A função do operador passa por alterar estas mesmas características e valores, produzindo assim um novo estado actual.

Para que a progressão no domínio de problemas seja útil para o agente, a selecção dos operadores não pode ser feita aleatoriamente. Esta selecção é um ponto fundamental na arquitectura *Soar*, uma vez que, se o agente tiver conhecimento de que a aplicação de um operador irá conduzi-lo a um dos seus objectivos, então esse operador será seleccionado. Estes são os conceitos essenciais na implementação da arquitectura *Soar*: objectivos, estados e operadores.

### Arquitectura

Uma das principais influências na concepção da arquitectura do *Soar*, passa pela forma como o conhecimento é representado. É através dos diferentes tipos de conhecimento que é definida a estrutura de memória, usada no processo de decisão. Esta estrutura está representada na figura 3.5 e é constituída pelos seguintes elementos:

- Memória a longo-prazo: armazena o conhecimento que existe, independentemente da situação actual. A esta memória pode estar associada a procedimentos, semântica ou episódios. Os procedimentos indicam quando e como uma determinada acção ou tarefa é executada. A semântica consiste numa estrutura que representa o conhecimento que o agente possui acerca

do ambiente que o rodeia. Por sua vez, a memória baseada em episódios consiste nas recordações do agente, referentes a experiências passadas;

- Memória de execução: conhecimento específico da situação actual. Para além desta informação específica, este tipo de memória pode também conter conhecimento geral relevante, factos e memórias de experiências passadas, que são úteis para a resolução do problema actual. Esta memória é representada no *Soar*, através da noção de estado mencionada anteriormente, ou seja, um conjunto de características e valores que definem a situação actual;
- Processo de decisão: usa a memória de execução como ferramenta para tomar decisões referentes à situação actual;
- Percepção e Acção: mecanismos de *input* e *output* usados pelo agente para obter informação acerca da situação actual a resolver e influenciar o ambiente com as suas decisões.

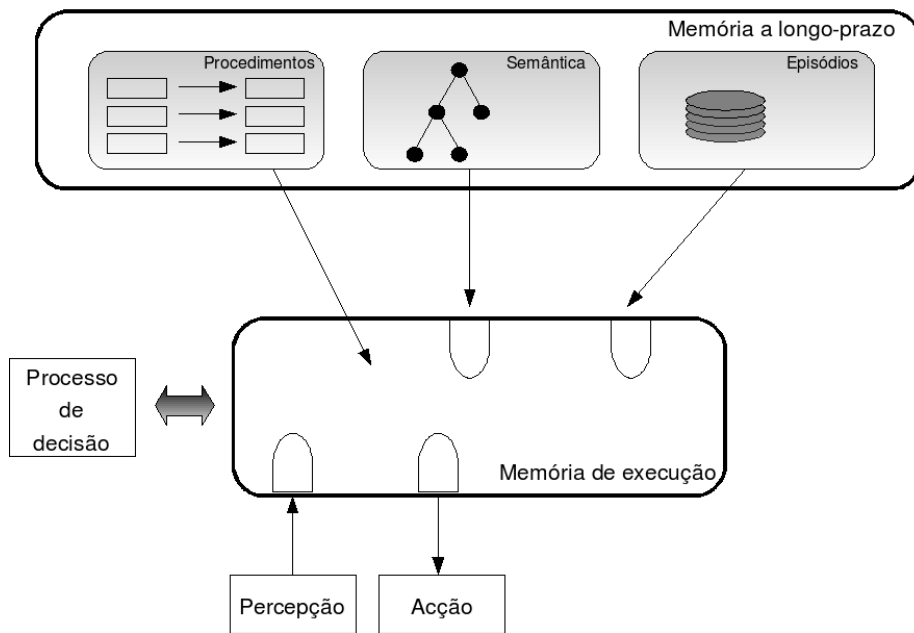


Figura 3.5: Estrutura da memória no *Soar*.

Para além dos diferentes componentes na estruturação do conhecimento da arquitectura *Soar*, a figura 3.5 representa também o mecanismo de extracção de conhecimento da memória a longo-prazo para a memória de execução. Este mecanismo pode ser feito de forma automática ou deliberada, antes ou durante o processo de decisão. Independentemente da forma como a extracção é feita, esta tem de ser feita sempre, uma vez que a memória a longo-prazo nunca pode ser acedida directamente.

Os diferentes tipos de memória enunciados anteriormente participam no processo de decisão, de uma forma complementar. A memória procedimental as-

sume um papel mais importante no controlo do comportamento, sendo mapeado directamente no conhecimento dos operadores, responsáveis pela alteração do estado do agente. Os outros tipos de memória (semântica e episódios) são usados essencialmente quando a memória procedimental estiver incompleta ou inadequada à situação actual.

A arquitectura *Soar* possui ainda outro mecanismo bastante importante para o processo de decisão: a criação de sub-estados para resolução de situações de impasse. Como o processo de decisão é feito com base em operadores que são activados, quando certos requisitos são satisfeitos, há a possibilidade de vários operadores serem activados simultaneamente. O sub-estado criado pelo *Soar* contém toda a informação relevante para resolver o impasse. Na maioria dos casos, é extraída informação de episódios anteriores ou da semântica do agente, para seleccionar o operador mais favorável, no sentido de atingir os objectivos definidos. Também é possível indicar que a escolha entre os diferentes operadores activados, pode ser feita de forma aleatória, caso tenham o mesmo grau de importância na resolução do problema.

## Aprendizagem

Qualquer arquitectura que seja definida para providenciar inteligência a um agente sintético, necessita de implementar mecanismos de aprendizagem eficazes e abrangentes. Os principais pontos a ter em conta, na construção de um mecanismo de aprendizagem, passam por definir o que deve ser aprendido, qual a fonte de conhecimento e em que momento deve ser feita a aprendizagem. Uma vez que a aprendizagem afecta de forma permanente o conhecimento do agente, esta deve actuar directamente na memória a longo-prazo, em todas as suas estruturas, mais precisamente no conteúdo destas: regras ou procedimentos, factos e episódios. Na arquitectura *Soar* estão implementados os seguintes métodos de aprendizagem [8]:

- **Extracção:** este foi durante muito tempo, o único mecanismo de aprendizagem usado na arquitectura do *Soar* e surge como consequência da criação dos sub-estados, quando é atingida uma situação de impasse, ou seja, quando é detectada a falta de conhecimento necessária para tomar uma decisão. Nestes casos, o sub-estado criado obriga a que sejam avaliados os operadores seleccionados, providenciando as condições para a pesquisa na memória a longo-prazo, de informação relevante para a resolução do impasse. Após todo este processo ter terminado, o conhecimento para resolver o impasse ficou disponível. O processo de aprendizagem aqui descrito passa então pela definição de regras, associadas a este novo conhecimento, com o objectivo de serem usadas no futuro;
- **Aprendizagem reforçada:** método de aprendizagem baseado em recompensas. Estas recompensas tanto podem ser positivas como negativas e consistem em formas de providenciar *feedback* ao agente, acerca das consequências das suas decisões. Por exemplo, se a última decisão contribuiu ou prejudicou o alcance dos objectivos estabelecidos, a consequente recompensa pode informar o agente. Estas recompensas podem afectar a memória a longo-prazo, servindo assim como mecanismo de aprendizagem;

- Memória baseada em episódios: aprendizagem que tira partido do conhecimento obtido através da experiência. O conhecimento baseado na experiência, permite recorrer a eventos no passado, tanto para procurar soluções a problemas actuais, como para tentar prever as consequências de determinada decisão. No *Soar*, um episódio consiste num sub-conjunto de elementos na memória de execução. Durante o processo de decisão, o episódio com mais semelhanças em relação aos elementos actuais da memória de execução, é recriado e usado para activar determinadas regras. A principal diferença entre a memória baseada em episódios e as restantes, consiste no facto de que o episódio é tratado como um todo, isto é, todos os elementos do episódio são aprendidos e retornados quando este é usado;
- Memória semântica: estruturas estáticas na memória de execução, que contêm o conhecimento do agente acerca do ambiente. A forma como este conhecimento é extraído da memória a longo-prazo para a memória de execução, é semelhante à forma como os episódios são retornados. Apesar da memória semântica ser mais compacta do que a de um episódio, o objectivo é o mesmo, ou seja, estas estruturas vão ser usadas para deduzir novas regras e novo conhecimento.



## Capítulo 4

# Trabalho Realizado

Neste capítulo estão descritas todas as decisões e desafios ocorridos durante a implementação do projecto. Os pontos a salientar dizem respeito à arquitectura definida para o sistema, e a comparação entre os fundamentos teóricos que servem de base à implementação e o que foi de facto implementado na construção dos agentes.

### 4.1 Jogo *Capture the Flag*

O jogo escolhido para implementar neste projecto foi o jogo *Capture the Flag*, também implementado no sistema *Gamebots* mencionado anteriormente. Neste jogo, são definidas duas bases, em que cada uma está associada a uma equipa. É posicionada uma bandeira em cada base e o objectivo passa então por capturar a bandeira, situada no território inimigo e trazê-la para a própria base. Este foi o jogo escolhido, uma vez que levanta outros desafios e tarefas:

- A base pode ser implementada de forma a dificultar a sua entrada e permitir a definição de estratégias de defesa e ataque;
- Cada equipa terá dois objectivos gerais: capturar a bandeira inimiga e impedir que a sua base seja atacada. Isto permite estratégias de equipa diversificadas;
- Podem ser definidas funções para cada elemento da equipa. Por exemplo, pode ser designado um elemento especial da equipa, responsável por capturar a bandeira.

A figura 4.1 mostra o cenário implementado. Estão representados os principais obstáculos e pontos de referência para os agentes. A base de cada equipa está rodeada por uma muralha, contendo assim dois pontos de entrada. A posição destes dois pontos de entrada não é fornecida a nenhum dos agentes do jogo, aumentando assim o grau de dificuldade. Os agentes deverão ser capazes de procurar estas entradas e evitar os obstáculos ao longo do cenário. Um dos principais requisitos tidos em conta na sua construção, passou por torná-lo completamente dinâmico. Isto significa que todos os elementos do cenário podem ser recolocados em posições diferentes, sem alterar a lógica de implementação do jogo. Podem ser construídos novos obstáculos, alterar a posição e formato das

bases e a posição das bandeiras. Desta forma, os agentes implementados serão informados sobre a posição das bandeiras, no momento em que são carregados no cenário, mas nunca terão informação sobre a fisionomia do cenário.

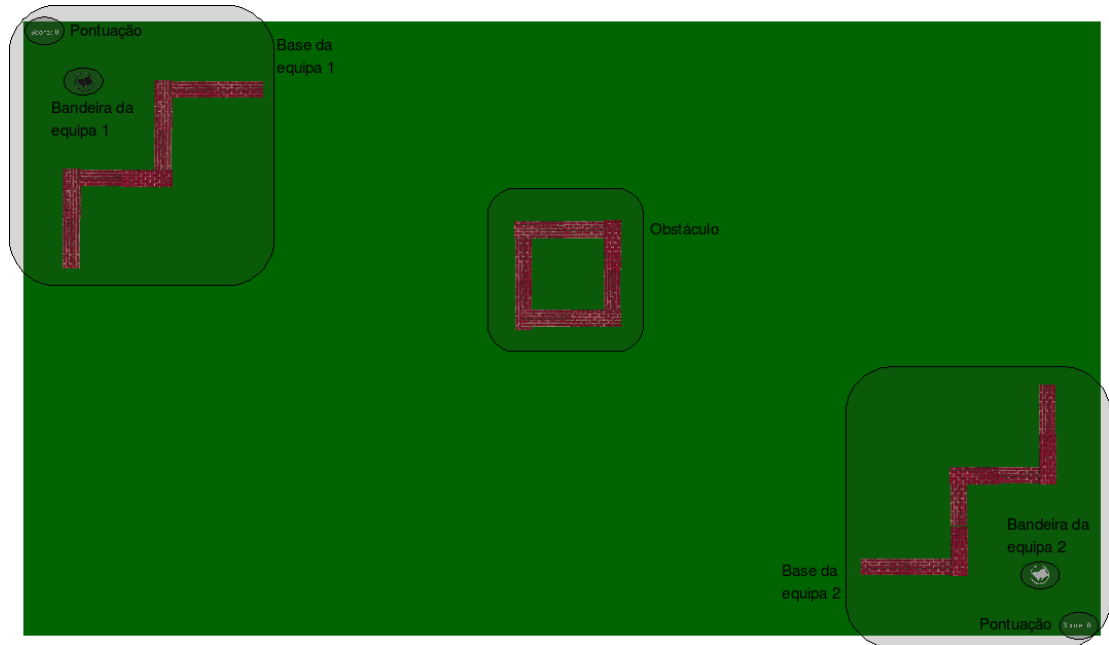


Figura 4.1: Cenário implementado para o sistema.

Quanto à definição das equipas, foram implementados três tipos de elementos, cada um com tarefas e limitações bem definidas

- Espião: elemento responsável por se infiltrar na base inimiga e capturar a bandeira, trazendo-a de volta para a sua base;
- Tropa: único elemento capaz de matar elementos da equipa adversária;
- Médico: elemento com capacidade de curar elementos da própria equipa.

Os diferentes tipos de elementos afectam directamente a forma como os agentes vão ser implementados. Por exemplo, o facto de tanto médico como o espião não terem a capacidade de disparar contra elementos da equipa adversária, faz com que estes tenham uma reacção defensiva, quando se deparam com tropas adversários. Para além disso, como cada agente tem diferentes tarefas e objectivos, o seu processo de decisão tem obrigatoriamente de divergir para estados distintos.

Finalmente, existem outras regras adicionais ao sistema. Todos os jogadores são carregados numa posição específica do cenário, que neste caso consiste em carregá-los sequencialmente, junto à base da equipa a que pertencem. Para o espião capturar a bandeira, necessita de entrar na base inimiga e atingir a posição exacta onde a bandeira se encontra, trazendo-a posteriormente para a

posição onde se encontra a sua própria bandeira. Sempre que o espião conseguir realizar este percurso é adicionado um ponto à sua equipa. Quando um jogador é abatido, este tem de aguardar um tempo de espera pré-definido, sendo depois carregado na sua posição inicial, junto à base. No caso do espião, se este possuir a bandeira quando é abatido, então esta volta a ser posicionada na respectiva base. No caso do médico, para este conseguir curar um elemento da sua equipa, tem de se aproximar deste e aguardar até a saúde do mesmo atingir os 100%. Sempre que um tropa dispara contra um jogador (pode até atingir um elemento da mesma equipa), este perde 10% da sua saúde.

## 4.2 Arquitectura do sistema

Como foi descrito nos objectivos, a arquitectura do sistema deverá ser modular, com uma clara divisão entre o processamento gráfico e o processamento dos agentes. A figura 4.2 ilustra em detalhe a arquitectura implementada e as tecnologias usadas em cada fase. Podem ser identificados os seguintes componentes:

- Motor gráfico;
- Protocolo de comunicação;
- Processamento de mensagens;
- Módulo de suporte à decisão;
- Módulo de gestão de emoções;

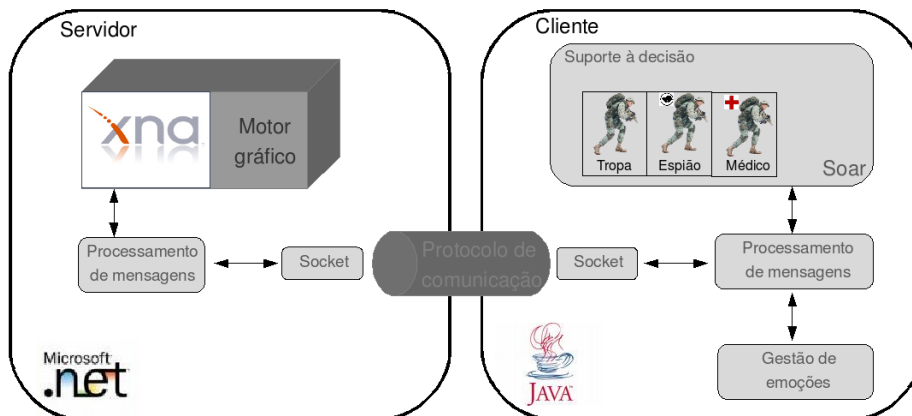


Figura 4.2: Arquitectura do sistema.

### Motor gráfico

O motor gráfico foi implementado em C#, usando as potencialidades da *framework Microsoft XNA*. Foi possível implementar um modelo que permitiu

separar a lógica das diferentes fases do processamento gráfico (inicialização, carregamento e actualização) e dos diferentes objectos (cenário, obstáculos e agentes). Isto foi possível graças à estrutura do código, fornecida pela *framework*:

- Função *Initialize()*: permite ao jogo inicializar tudo o que é necessário antes de ser executado. Isto pode incluir serviços ou conteúdo não gráfico;
- Função *LoadContent()*: é chamada uma vez por jogo e é usada para carregar todo o seu conteúdo;
- Função *Update(GameTime gameTime)*: recebe como parâmetro uma imagem do tempo de jogo, e permite que o jogo execute lógica como a actualização de conteúdo (verificar colisões, receber *input* e actualizar o estado dos jogadores). Esta função é executada periodicamente, sendo que o período de *loop* é definido na variável *TargetElapsedTime* que, por defeito, tem o valor de 1/60 segundos;
- Função *Draw(GameTime gameTime)*: esta função é invocada logo após a função *Update* e permite desenhar todo o conteúdo gráfico do jogo.

Foram ainda adicionadas as seguintes secções para melhorar a legibilidade e manutenção do código:

- Inicialização de todos os obstáculos do jogo;
- Inicialização dos agentes no jogo;
- Actualização do movimento dos agentes;
- Verificação de colisão entre os diferentes objectos;
- Actualização e controlo das balas disparadas pelos agentes;
- Reposição de agentes mortos em jogo;
- Processamento paralelo de ligação entre o servidor e os diferentes clientes;
- Processamento e envio das mensagens para os agentes, com a informação sobre o ambiente

### Protocolo de comunicação e processamento de mensagens

Um dos objectivos delineados na construção do protocolo de comunicação entre o ambiente e os agentes foi a sua modularidade e transparência. Isto significa que a estrutura das mensagens e o seu processamento tem de ser feito de tal forma a que seja o agente a seleccionar a informação que pretende utilizar e não o contrário. Um exemplo disto é a informação sobre a posição das bandeiras no cenário que é enviada a todos os agentes mas que é descartada pelo agente do tipo médico. Isto permite que do lado do servidor não sejam tirados pressupostos sobre a forma como o agente executa o seu processo de decisão, e que mais tarde este mesmo processo seja alterado para utilizar informação adicional.

Na definição e implementação do protocolo de comunicação, houveram pontos que não foram considerados para simplificar este processo. Não existem

métodos de autenticação, encriptação e verificação da integridade das mensagens e como tal, este pode ser um ponto de melhoria do projecto no trabalho futuro, principalmente se este sistema evoluísse para ser usado a uma escala global.

O processo de troca de mensagens é representado na figura 4.3. Podem ser identificadas as seguintes fases:

1. O servidor envia aos agentes, a informação sobre o ambiente. Esta informação inclui informação genérica (coordenadas das bandeiras ou de posições estratégicas) e informação recolhida pelo sensor do agente, isto é, o conteúdo que rodeia o seu campo de visão;
2. São enviados ao agente todos os parâmetros usados para inicializar e actualizar o seu estado emocional. Neste caso, como a informação a enviar é homogénea, a mensagem é estruturada com um separador entre todos os valores;
3. Recepção e processamento da mensagem com informação sobre o ambiente. Como a informação enviada nesta mensagem é heterogénea, cada um dos campos possui um identificador único, usado para fazer o processamento no lado do cliente, através do uso de expressões regulares. A cada secção de informação da mensagem, está associada uma expressão regular que recolhe esta informação e passa-a ao sistema de suporte à decisão do agente;
4. Recepção e processamento da mensagem com os parâmetros a usar pelos sistemas emocionais do agente;
5. Envio para o servidor, da decisão tomada pelo agente. Esta decisão pode passar por andar numa determinada direcção, disparar contra um adversário, ou até decidir não desempenhar qualquer acção;
6. Envio para o servidor, do estado emocional do agente. Apesar deste estado emocional ser usado maioritariamente do lado do cliente, para definir o seu comportamento, o estado emocional também é processado do lado do servidor, não só para este ficar visível para o observador, mas também para processar algumas manifestações visuais associadas a algumas emoções (nível de motivação associado às emoções de alegria e tristeza, que afectam a velocidade de deslocação do agente).

### Módulo de suporte à decisão

O módulo de suporte à decisão foi implementado, usando as potencialidades da arquitectura *Soar* e foi integrado na implementação do cliente em Java, através da ferramenta SML (*Soar Markup Language*). Esta ferramenta implementa um *kernel* do *Soar* em várias linguagens (C++, Java, TCL) e permite comunicar entre a aplicação e este *kernel*, através de mensagens em XML. As regras e operadores definidos no *Soar* são implementados em ficheiros à parte (um por agente) e carregados no *kernel*, dependendo do tipo de agente. As bibliotecas existentes no SML permitem executar as tarefas necessárias para o processo

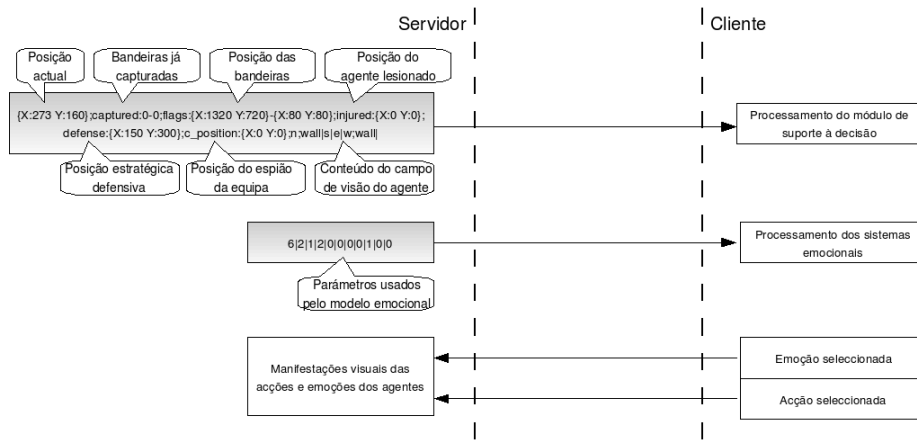


Figura 4.3: Protocolo de comunicação do sistema.

de decisão: criação do agente, definição do estado inicial (atribuição dos valores iniciais às características do agente), selecção de operadores e extracção do resultado do processo de decisão.

### Módulo de gestão de emoções

O módulo de gestão de emoções foi integrado na implementação do cliente em Java. Foi criada uma classe que recebe os parâmetros enviados pelo servidor e contem todos os sistemas emocionais que usam estes mesmos parâmetros para calcular a intensidade de todas as emoções. Para cada um destes sistemas emocionais, o módulo de gestão de emoções executa as seguintes etapas:

- Inicialização da intensidade, valor de activação e valor limite, de todas as emoções;
- Cálculo da intensidade das emoções, com base nos parâmetros enviados pelo servidor;
- Dissipação periódica das emoções. Esta etapa só é executada, caso a intensidade de uma das emoções atinja o valor de activação. Neste caso, a intensidade de todas as emoções diminui em períodos síncronos pré-definidos (20 ciclos). Esta etapa tem uma duração máxima de 100 ciclos, depois da qual volta à etapa anterior;
- Selecção da emoção do agente. São consideradas todas as emoções com intensidade superior ao respectivo valor de activação e é seleccionada uma de forma aleatória.

Uma análise ainda mais detalha à arquitectura do sistema, pode ser feita através do diagrama de classes representado na figura 4.4.

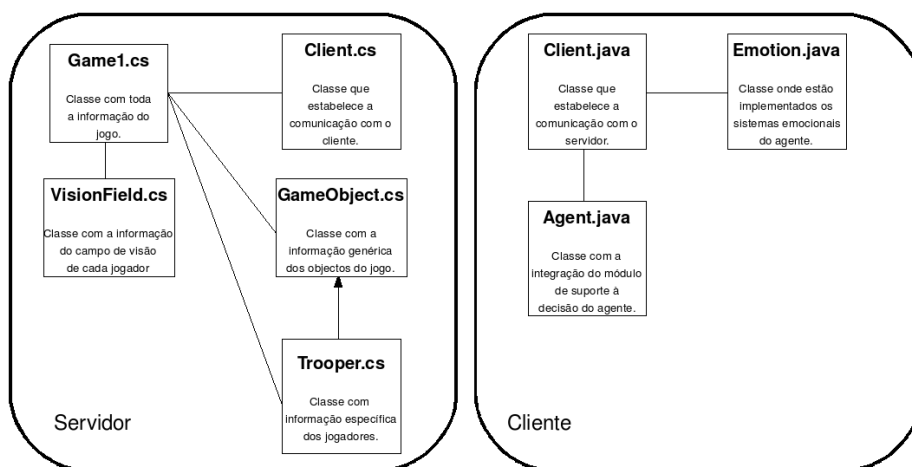


Figura 4.4: Diagrama de classes do sistema.

### 4.3 Implementação dos agentes

A implementação dos agentes foi feita no componente de suporte à decisão identificado na arquitectura do sistema, sendo que o principal objectivo passou por tirar partido das potencialidades da arquitectura *Soar*, para ultrapassar as dificuldades impostas pelas regras do jogo.

O primeiro passo na implementação dos agentes, passa por identificar claramente os requisitos de cada um, considerando sempre os seus objectivos e limitações:

- **Espião:** deve saber em que posição estão as duas bandeiras e definir o seu percurso em direcção a uma delas, dependendo se já capturou a bandeira inimiga ou não. Ao longo deste percurso, terá que ser capaz de se desviar de possíveis obstáculos e evitar inimigos;
- **Tropa:** sendo o único agente com capacidade de eliminar elementos da equipa adversária, pode assumir uma posição defensiva (defender a própria base) ou uma posição ofensiva (auxiliar o espião na captura da bandeira adversária). Dependendo da posição que assume, este agente tem de saber quais as posições estratégicas para defender a base, ou qual a posição actual do espião da própria equipa. Em qualquer um dos casos, o objectivo essencial passa por eliminar os jogadores inimigos que consiga detectar no seu campo de visão;
- **Médico:** tem como principal tarefa, dirigir-se a elementos da própria equipa que se encontrem feridos. Para simplificar a tarefa deste agente, é lhe fornecida a posição do jogador mais ferido. Como não tem a capacidade de disparar sobre elementos da equipa adversária, deverá também evitar os tropas inimigos.

Após identificar os requisitos, é necessário definir uma estrutura que caracterize a noção de estado no *Soar*. Esta estrutura é constituída por um conjunto de características e valores, que identificam o agente e possibilitam que este execute

o processo de decisão, com base nos requisitos acima mencionados. Esta estrutura é ilustrada na figura 4.5, onde se podem identificar as seguintes secções:

- *desired-direction*: usado pelo espião para saber se pretende dirigir-se para a base inimiga ou regressar à própria base. Este parâmetro é calculado através da equipa do agente, e da captura ou não da bandeira, por parte de algum membro da própria equipa;
- *agent*: contem informação recolhida do ambiente. Inclui as coordenadas do agente, das duas bandeiras, das posições defensiva e ofensiva e do elemento mais lesionado da equipa. Para além destas coordenadas, inclui também um parâmetro que indica se a equipa já capturou a bandeira ou não, um parâmetro com a equipa onde o agente está integrado, e outro parâmetro com o estado emocional actual;
- *sensor-field*: informação do campo visual do agente. Em cada uma das posições adjacentes do agente podem ser detectados inimigos, obstáculos ou companheiros de equipa.
- *move*: contem informação sobre a acção de deslocamento do agente, caso seja esta a seleccionada. Inclui a direcção de deslocamento, a estratégia do agente (se assume uma abordagem ofensiva ou defensiva) e o estado da acção (se já está efectuada ou não);
- *shoot*: contem informação sobre a acção de disparo sobre um inimigo, caso seja esta a seleccionada. Inclui a direcção de disparo e o estado da acção (se já está efectuada ou não);

A capacidade de implementar mecanismos de aprendizagem no *Soar*, não foi muito explorada neste projecto. Ainda assim, foi elaborada uma regra que possibilitou a implementação de algumas estruturas descritas anteriormente: quando o agente decide avançar para uma posição, não deverá regressar à posição de partida no estado seguinte. Esta regra torna o agente mais inteligente na definição do percurso até ao seu objectivo e permitiu implementar os seguintes mecanismos de aprendizagem:

- Memória baseada em episódios: é armazenado um sub-conjunto de elementos da memória de execução para ser usado nas próximas decisões. Neste caso, o episódio apenas contem a direcção na qual o agente se movimentou;
- Memória semântica: são definidas estruturas para suportar o conhecimento que o agente possui acerca do ambiente. Neste caso, é definida a estrutura que permite ao agente, identificar a direcção oposta à direcção para a qual decidiu mover-se, uma vez que é esta direcção que tem de evitar.

Chegando a este ponto da implementação, já foi atingido o objectivo da construção de um sistema modular multi-agente, em que é possível carregar agentes remotamente, através de um protocolo de comunicação genérico e transparente. Os agentes têm as suas tarefas bem definidas e demonstram alguma inteligência, na tentativa de atingir os seus objectivos. A figura 4.6 demonstra um exemplo de um jogo a decorrer entre duas equipas constituídas por cinco elementos cada (dois tropas, um médico e um espião).



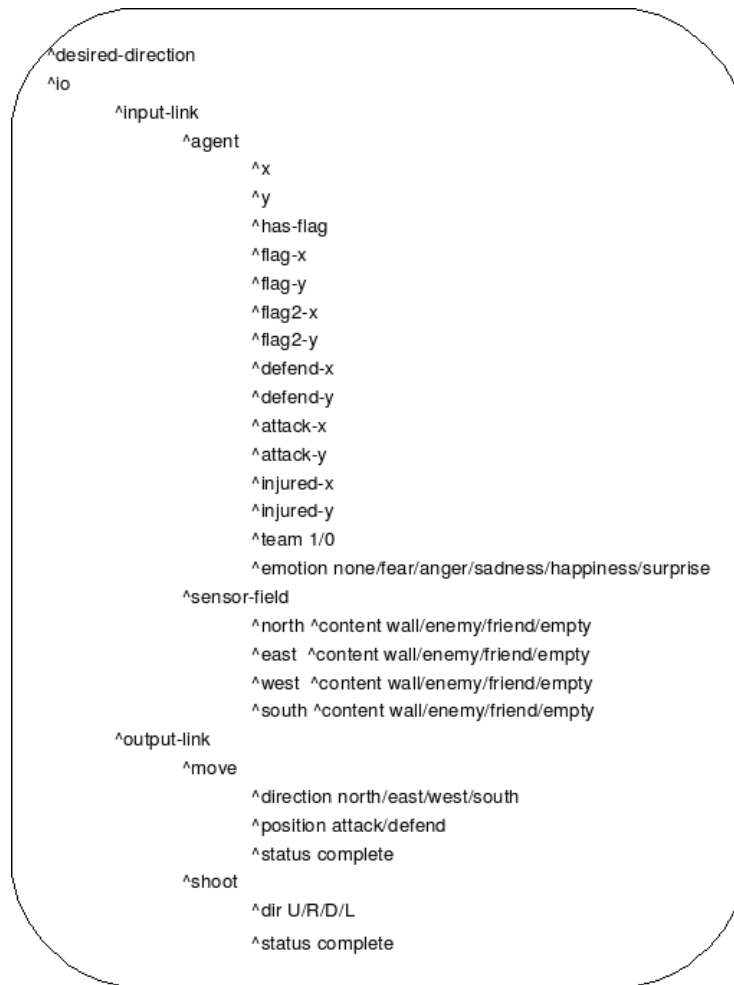


Figura 4.5: Estrutura das características dos agentes no *Soar*.

## 4.4 Modelo emocional

Após a implementação do cenário e dos agentes, resta implementar o passo final que pretende introduzir no sistema, os conceitos que formam o principal objectivo deste projecto: a imprevisibilidade dos agentes na realização das suas tarefas. Isto é feito ao nível do módulo de gestão de emoções do sistema, inspirado no Modelo *Cathexis* descrito anteriormente, através de uma versão simplificada deste. A principal simplificação feita, consistiu na alteração da fórmula do cálculo da intensidade:

$$I_{et} = X(f(I_{et-1}) + \sum_k L_{ke}) \quad (4.1)$$

A alteração da função acontece devido aos pressupostos assumidos na construção do modelo emocional:

- Apenas uma emoção pode ser desencadeada num dado momento. Isto faz



Figura 4.6: Exemplo de um jogo a decorrer.

com que não seja possível haver o conceito de mistura de emoções;

- As emoções primárias foram implementadas com base em sistemas emocionais que são activados quando a sua intensidade atinge um determinado parâmetro. O conceito de sentimento (elementos contidos nos sistemas emocionais e desencadeados a intensidades mais baixas) não foi considerado na implementação do modelo emocional;
- As emoções são independentes entre si. Isto significa que, quando uma emoção é desencadeada, não produz qualquer efeito na intensidade das restantes. É por este motivo que a equação do modelo emocional foi simplificada;
- Com a implementação de cada uma das emoções possuindo o seu próprio sistema emocional, significa que existe a possibilidade de várias emoções serem desencadeadas simultaneamente. A personalidade de cada agente determina qual ou quais as emoções que vão prevalecer, perante uma situação destas. Para além disso, a personalidade define também os valores de activação e saturação (parâmetros  $\alpha$  e  $\omega$ ) para cada emoção. Na implementação deste projecto, o conceito de personalidade não foi muito desenvolvido, na medida em que os parâmetros  $\alpha$  e  $\omega$  são os mesmos para todos os agentes e a escolha entre as várias emoções desencadeadas, é feita de forma aleatória.

Analisando o módulo de gestão de emoções implementado, é possível verificar outra alteração à equação do Modelo *Cathexis*. Esta alteração passa por dividir

a equação em duas fases:

$$I_{et} = X \left( \sum_k L_{ke} \right), \text{ se } I_{et-1} < \alpha \quad (4.2)$$

$$I_{et} = f(I_{et-1}), \text{ se } I_{et-1} > \alpha \quad (4.3)$$

Isto significa que, em cada instante, a intensidade da emoção é actualizada com o somatório de todos os parâmetros de entrada que possam influenciar essa mesma emoção, enquanto o seu valor de activação não for atingido, e está sempre limitada através do parâmetro  $\omega$ . Quando a intensidade ultrapassar o parâmetro  $\alpha$ , os parâmetros de entrada deixam de ser considerados e a função de decaimento actua sobre o valor da intensidade no instante anterior. Para simplificar este processo foi implementada a função de decaimento  $f(I_{et}) = I_{et-1} - k$  em que  $k$  é uma constante definida com o mesmo valor para todas as emoções.

No cálculo da intensidade, foi definido um conjunto de parâmetros de entrada, cada um com um grau de importância distinto para cada emoção:

- Diferença entre o número total de mortos da equipa adversária e o número total de mortos da própria equipa (d);
- Diferença entre a pontuação da própria equipa e a pontuação da equipa adversária (s);
- Número de inimigos mortos pelo agente (m);
- Assistir à morte de um elemento da própria equipa (0 e 1 como valores possíveis). Este parâmetro é activado sempre que um companheiro de equipa morre no campo de visão agente, e volta a ser desactivado quando este morre (a);
- Frustração no cumprimento de objectivos (0 e 1 como valores possíveis). Esta frustração é calculada com base na acumulação de obstáculos no campo de visão do agente. Se este número atingir um determinado limite, significa que a progressão do agente não está a ser feita de forma eficiente. O *reset* a este parâmetro é feito de forma cíclica (f);
- Número de inimigos presentes no campo de visão do agente (i);
- Cansaço do agente (0 e 1 como valores possíveis). Este parâmetro é activado após um determinado período de tempo, simulando o desgaste sentido pelo agente, sendo feito o *reset* sempre que este morre (c);
- Recuperação total da saúde (0 e 1 como valores possíveis). Sempre que um agente consegue obter auxílio de um médico para recuperar totalmente a sua saúde, este parâmetro é activado. O *reset* é feito sempre que o agente morre (r);
- Aparecimento de um tropa inimigo no campo de visão do agente (0 e 1 como valores possíveis). Sempre que este parâmetro é activado, é logo desactivado no instante seguinte (t).

Tabela 4.1: Cálculo da intensidade dos sistemas emocionais.

Emoção	Intensidade	Parâmetros
Raiva	$I = -d + a - s$	$\alpha = 7; \omega = 11$
Medo	$I = -d + a + 3i$	$\alpha = 6; \omega = 10$
Alegria	$I = d + s + 7r + m$	$\alpha = 8; \omega = 12$
Tristeza	$I = -d - s + c + f$	$\alpha = 8; \omega = 12$
Surpresa	$I = 6t$	$\alpha = 5; \omega = 10$

Tabela 4.2: Associação entre emoções e comportamentos.

Emoção	Comportamento
Raiva	Perde a noção da tática adoptada. Deixa de assumir posições defensivas ou ofensivas, procurando inimigos ao longo do cenário.
Medo	Paralisa, ficando sem capacidade de reacção.
Alegria	Fica motivado. Avança no terreno a uma velocidade superior.
Tristeza	Fica desmotivado. Avança no terreno a uma velocidade inferior.
Surpresa	Sem comportamento associado.

A tabela 4.1 mostra em detalhe, a relação entre a intensidade de cada emoção e os parâmetros acima mencionados. Como foi dito anteriormente, é possível constatar que é atribuído um grau diferente a cada parâmetro, dependendo da relevância e do domínio de valores do mesmo. Os parâmetros  $\alpha$  e  $\omega$  também estão representados na tabela 4.1 e foram definidos com base no número e tipo de parâmetros de entrada.

A tarefa final na construção do modelo emocional, passa por associar comportamentos às emoções. Este aspecto é bastante importante, uma vez que são os comportamentos que transmitem implicitamente ao observador, o estado emocional do agente. A tabela 4.2 mostra a associação feita entre as emoções implementadas e os respectivos comportamentos. Neste passo, a personalidade também desempenha um papel decisivo. No âmbito deste projecto, é considerado que todos os agentes desencadeiam reacções semelhantes, perante os mesmos estados emocionais. Esta tarefa exigiu que o módulo de suporte à decisão do sistema fosse ligeiramente alterado, para ser integrado o conceito de emoção. Sendo assim, o agente passa a ter conhecimento da emoção desencadeada durante o processo de decisão, possibilitando assim que os comportamentos descritos na tabela sejam integrados nesse mesmo processo.

## Capítulo 5

# Resultados

Neste capítulo são detalhados todos os resultados do projecto, focando essencialmente os pontos de investigação que foram delineados no início do mesmo. É feita uma listagem dos principais objectivos traçados e explicado o que foi feito para os atingir. No que diz respeito aos objectivos que não foram atingidos na totalidade, é feita uma análise ao caminho a seguir no trabalho futuro.

### Cenário

O cenário construído para o jogo serve de base para definir as regras e complexidade do ambiente. Os objectivos propostos para este componente foram: construir um cenário realista e providenciar os desafios e tarefas aos agentes. O cenário construído já foi explicado em detalhe no capítulo anterior, sendo que neste capítulo pretende-se verificar se este cumpre os requisitos definidos.

O facto de permitir a construção dinâmica dos obstáculos, faz com que este ambiente torne o processo de decisão do agente mais complexo. O agente é carregado no cenário sem poder retirar qualquer pressuposto acerca da posição dos obstáculos e da forma como os pode evitar. Existem ainda outras regras que conseguem transmitir realismo ao jogo. Todas as colisões entre objectos são detectadas, impedindo a sua sobreposição (interacção entre agentes ou entre agentes e obstáculos). Algumas emoções conseguem ser processadas pelo cenário, mais concretamente a alegria e a tristeza, com o objectivo de produzir manifestações visuais das mesmas. Neste caso, as emoções detectadas afectam a velocidade de deslocamento do agente, permitindo assim que um agente motivado se consiga deslocar mais rapidamente do que um agente cansado ou desmotivado.

A figura 5.1 mostra a principal preocupação na concepção do cenário, isto é, a construção da base de cada equipa. Aqui é possível observar os principais desafios impostos aos agentes, principalmente para a atacar e capturar a bandeira:

1. Existência de apenas duas entradas na base em posições opostas. Os agentes não têm conhecimento acerca da posição destas duas entradas e para dificultar ainda mais, a dimensão destas foi definida para que só seja possível a entrada de um agente de cada vez;

## 5. Resultados

---

2. A bandeira colocada no interior da base está posicionada de forma a que o agente tenha de se desviar da sua trajectória para encontrar uma das entradas da base. Isto acontece, uma vez que se o agente definir uma trajectória directa para a bandeira, irá obrigatoriamente encontrar obstáculos e até mesmo inimigos. Este revelou-se o principal desafio de implementação dos agentes;
3. Definição de duas posições defensivas estratégicas. Estas posições são enviadas aos agentes, para serem usadas na estratégia da equipa, dificultando ainda mais a entrada na base;
4. A base é construída com um formato irregular e com uma extensão de obstáculos relativamente grande.

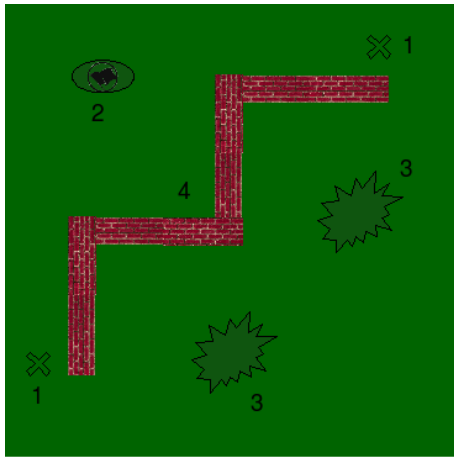


Figura 5.1: Características da base construída.

De uma forma geral, os objectivos para o cenário foram atingidos. No entanto existem pontos a melhorar, como por exemplo, a existência de vários tipos de obstáculos (portas, terreno irregular, etc.), permitir a interacção entre obstáculos e agentes, por forma a que seja mesmo possível usar obstáculos no auxílio ao cumprimento de objectivos e diversificar o tipo de jogos existentes no cenário, com o objectivo de diversificar as tarefas possíveis de cada agente.

### Arquitectura modular

Como foi explicado anteriormente, a arquitectura do sistema foi concebida para obedecer a determinados requisitos: sistema modular e com um protocolo de comunicação genérico entre o ambiente e os agentes. Deve haver uma separação clara entre o domínio de acção do ambiente e o processamento dos agentes. No capítulo anterior foi explicado em detalhe, a arquitectura implementada para atingir estes objectivos. Para demonstrar os resultados obtidos, é apresentado o exemplo da figura 5.2, em que são propostas várias alterações ao sistema, sem necessidade de alterar a estrutura base da arquitectura e a lógica implementada:

## 5. Resultados

1. Adicionar mais campos às mensagens trocadas entre o cliente (agente) e o servidor (ambiente). Através do *parsing* das mensagens, usando expressões regulares é possível alterar a estrutura das mensagens, sem afectar os campos já utilizados. Isto é bastante útil, por exemplo, para enviar para os agentes, nova informação adicionada no ambiente, ou caso sejam adicionados mais agentes;
2. Activar/desactivar a gestão de emoções. Em caso de ser necessário analisar outros aspectos dos agentes que impliquem retirar as emoções do sistema, esta comutação é feita com um esforço mínimo e sem implicações para os restantes módulos;
3. Adicionar um novo tipo de agente ao sistema. As alterações necessárias são minimizadas, caso o sistema seja modular. No servidor, é necessário alterar a mensagem enviada ao agente, caso este necessite de mais informação para além da que já é enviada. É também necessário alterar a componente de processamento gráfico do agente, não havendo necessidade de alterar a lógica do cenário ou outros componentes. No cliente é também necessário implementar o agente, através do *Soar*. Para além disto, é também necessário alterar a classe que faz o interface entre os agentes e o cliente, para passar a reconhecer o novo agente.

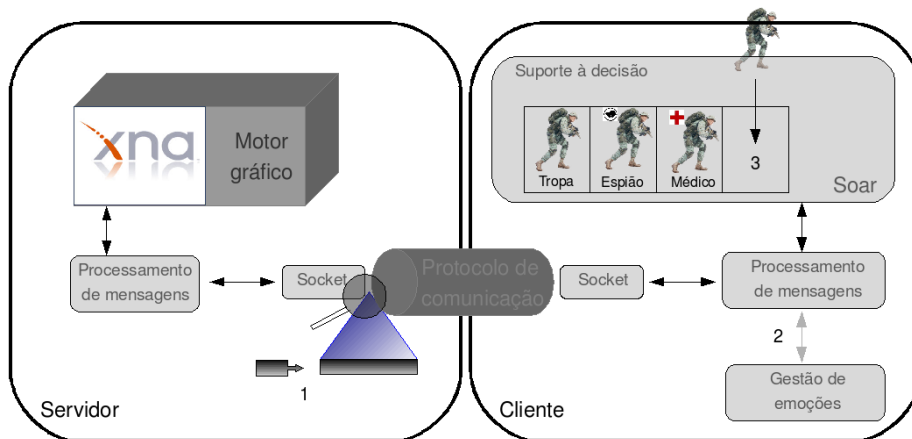


Figura 5.2: Alterações possíveis ao sistema implementado.

Resumindo a análise à modularidade da arquitectura concebida, pode-se concluir que esta está mais desenvolvida do lado do cliente. O trabalho futuro passa essencialmente pela alteração de alguns componentes do lado do servidor, em que o principal objectivo passará pelo suporte para vários tipos de jogo, sendo possível a sua adição e remoção, com o mínimo esforço possível de implementação.

### Agentes inteligentes

É possível analisar o nível de inteligência dos agentes implementados, através da sua capacidade de resolução de problemas. A figura 5.3 ilustra uma secção

## 5. Resultados

---

do domínio de problemas de um dos agentes, neste caso o espião, sendo esta uma forma de avaliar a inteligência do agente, pois permite saber até que ponto o agente está preparado para lidar com todos os desafios possíveis.

Como foi dito anteriormente, o maior desafio implementado no cenário, consiste na entrada na base inimiga. Após a implementação dos agentes, verifica-se que este desafio é ultrapassado, ou seja, o agente é capaz de detectar que existe um obstáculo ao longo da trajectória que definiu, e conseqüentemente necessita de alterar a sua abordagem. Isto foi possível através da implementação dos mecanismos de memória referidos anteriormente, ou seja, como o agente sabe que não deve regressar à posição de origem, uma vez que significa que não fez progressos no seu processo de decisão, então vai procurar outras direcções possíveis, ou seja, faz o percurso à volta da base, até encontrar uma entrada. Apesar deste mecanismo funcionar, necessita de aperfeiçoamentos, uma vez que continua a ser uma abordagem muito baseada em tentativa e erro.

Analisando os restantes desafios listados na figura 5.3, verifica-se que o objectivo proposto de preparar o agente para decidir sobre todos os problemas impostos pelo cenário foi cumprido. Existem no entanto alguns aspectos a melhorar. A qualidade dos agentes poderá melhorar significativamente se o conceito de memória for mais explorado. Neste sentido, o objectivo poderá passar por implementar no agente, a capacidade deste recordar o caminho por onde já passou, evitando assim os percursos cíclicos e diminuindo a abordagem de tentativa e erro. Também podem ser feitas melhorias em relação ao processo de decisão do agente. O domínio de problemas poderá ser alargado para, por exemplo, permitir que sejam tomadas decisões que saiam do âmbito da estratégia de equipa, se o agente concluir que é o melhor para atingir os seus objectivos. Outro ponto de evolução no projecto poderá passar por implementar processos de decisão mais complexos, com o objectivo de melhorar as estratégias de equipa. Um exemplo ideal é a possibilidade de enviar uma equipa para aniquilar a defesa da base inimiga, antes de enviar o espião, ou mesmo encurralar o espião na sua base, facilitando a sua eliminação.

### Gestão adequada das emoções

Para finalizar a análise dos resultados do projecto, é necessário verificar se os objectivos definidos para o modelo emocional foram atingidos: as emoções devem ser geridas de uma forma realista, abrangendo todas as emoções que afectem o comportamento humano, sendo que estas devem ser integradas no sistema de uma forma subtil.

A figura 5.4 mostra um exemplo do jogo a decorrer, com o modelo emocional já integrado. Apesar das emoções serem algo que deve ser deduzido a partir dos comportamentos do agente, foi adicionado um campo de texto a cada agente, com a emoção actual do mesmo, para garantir que o comportamento actual está relacionado com o seu estado emocional. Após a observação de vários jogos, foi possível concluir que o objectivo de integrar as emoções de forma subtil no sistema foi atingido, uma vez que os agentes continuaram a cumprir as suas tarefas, com a diferença de haverem ligeiros desvios nos seus comportamentos e motivações. O modelo que serviu de base para este projecto, permitiu atingir os objectivos restantes, uma vez que este se baseou em estudos em áreas como a Psicologia e a Neurobiologia. Isto significa que é garantida a credibilidade das bases para elaboração das diferentes emoções e a forma como estas são



## 5. Resultados

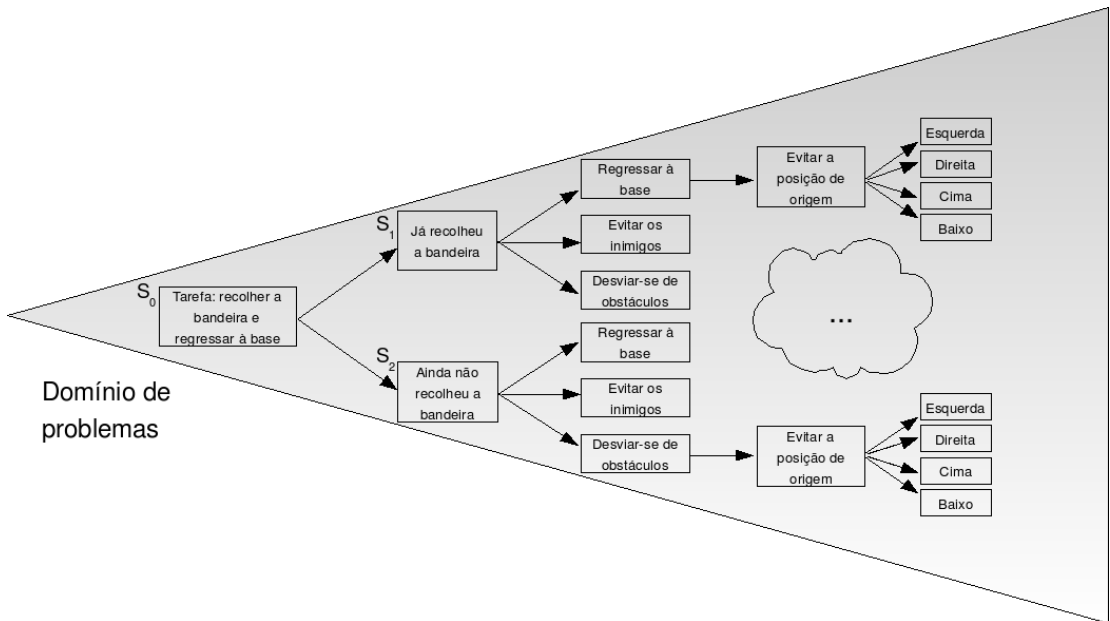


Figura 5.3: Domínio de problemas do agente espião.

desencadeadas e dissipadas.

Apesar dos objectivos propostos terem sido uma preocupação ao longo da implementação do modelo emocional, existem vários aspectos a melhorar para aumentar a imprevisibilidade do comportamento dos agentes. Estas melhorias passam essencialmente pela implementação do conceito de interacção entre diferentes sistemas emocionais, o que permitia que várias emoções influenciassem o comportamento do agente no mesmo instante (mistura de emoções), e as emoções desencadeadas produzissem um efeito excitatório ou inibitório nas restantes. Outra melhoria possível passa pelo aperfeiçoamento do sistema comportamental, ou seja, definir um conjunto mais variado de comportamentos associados a cada emoção. Isto implica o desenvolvimento do conceito de personalidade (não muito explorado neste projecto) para permitir que seja seleccionado o comportamento mais adequado a cada estado emocional e que seja feita uma selecção das emoções desencadeadas, quando existem várias emoções com intensidade elevada. Outro conceito abordado na descrição do modelo *Catheris* foi o de sentimento, integrado nos vários sistemas emocionais e desencadeado a intensidades mais baixas. Ao implementar este conceito no sistema, será possível oferecer uma variação ainda maior de comportamentos ao agente.

## 5. Resultados

---

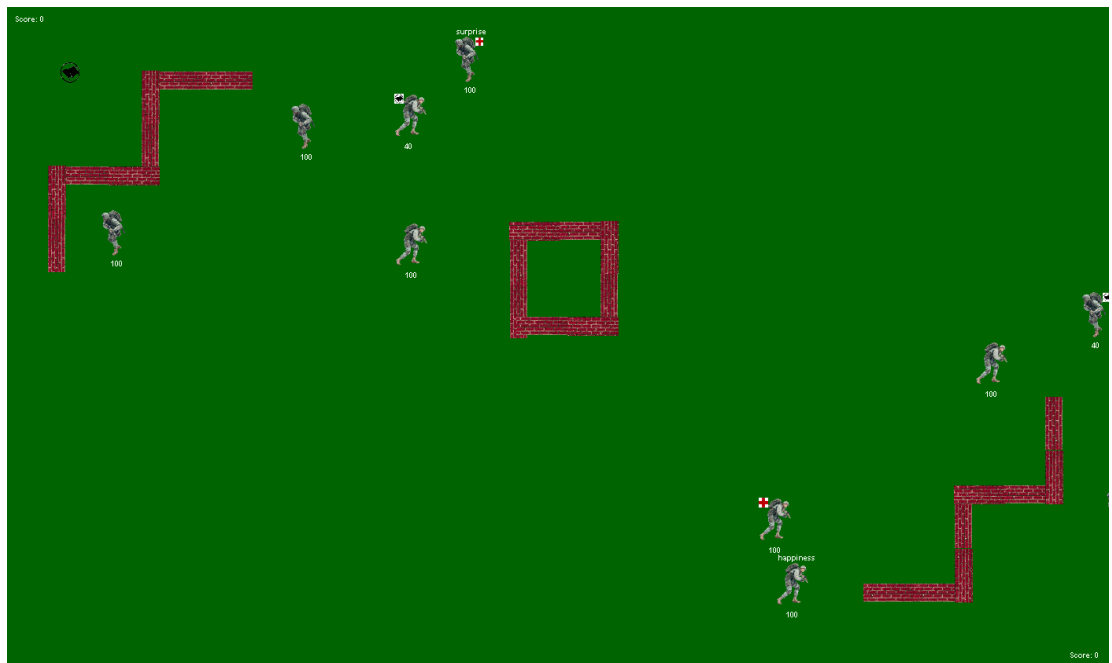


Figura 5.4: Exemplo de um jogo a decorrer com o modelo emocional activo.

## Capítulo 6

# Conclusões e Trabalho Futuro

Ao fazer um resumo do que foi descrito no trabalho realizado e na análise de resultados, pode ser concluído que os objectivos propostos foram atingidos, havendo no entanto alguma margem de desenvolvimento futuro para aperfeiçoar a principal característica pretendida para o sistema, isto é, a imprevisibilidade. Este desenvolvimento passa essencialmente pelo compromisso entre dois factores: a capacidade de decisão do agente e a diversificação de emoções e comportamentos associados.

Este projecto permitiu explorar vários conceitos teóricos associados aos agentes sintéticos como a autonomia, capacidade de resolução de problemas, aprendizagem e memória. Uma das principais conclusões a tirar destes conceitos, é a necessidade de os aplicar todos na implementação de qualquer agente, para este ser capaz de desempenhar tarefas e tomar decisões de forma eficiente. Isto foi demonstrado através da implementação de um mecanismo simples de memória que permitiu aumentar significativamente a inteligência do agente, quando este passou a saber que ao mover-se ao longo do cenário, não deverá regressar à posição anterior, uma vez que significa que não fez progressos para atingir os seus objectivos.

Outro aspecto explorado neste projecto foi a cooperação de diferentes tipos de agentes, com tarefas distintas entre si. Foram definidos três tipos de agentes, cada um com a capacidade de responder a todos os desafios do ambiente e de desempenhar as tarefas para as quais foi concebido. O sistema foi implementado de forma a que no futuro seja possível integrar mais tipos de agentes, com um esforço mínimo de implementação.

Quanto ao modelo emocional, este foi integrado no sistema para aumentar a imprevisibilidade do mesmo. O objectivo foi cumprido, tendo sempre a preocupação de não alterar drasticamente o comportamento dos agentes, ao ponto de comprometer a capacidade de cumprimento das suas tarefas. Apesar de ter sido mantida uma linearidade entre emoções e comportamentos, o número de factores que influenciam as emoções era grande o suficiente, para dificultar a previsão do estado emocional do agente e consequentemente o comportamento adoptado.

Finalmente, pode-se também concluir que o sistema foi implementado, com

## 6. Conclusões e Trabalho Futuro

---

base numa arquitectura que permita novos desenvolvimentos em componentes independentes, sem afectar os restantes, como por exemplo, adicionar novos agentes, nova informação no ambiente, novos obstáculos ou novas emoções.

No que diz respeito aos possíveis desenvolvimentos a incluir no trabalho futuro, podem ser considerados os seguintes pontos:

- Diversificar mais o cenário do jogo, através da existência de diferentes tipos de obstáculos, com os quais o agente pode interagir;
- Implementar mais tipos de jogos, permitindo assim que os agentes tenham um número maior de tarefas a executar;
- Desenvolver as estruturas de memória do agente. O conceito de memória está directamente ligado com a capacidade de aprendizagem. As melhorias podem estar relacionadas, por exemplo, com o armazenamento de episódios durante o jogo, para reconhecer padrões na estratégia da equipa adversária ou formas de aumentar o sucesso da própria equipa, através do recurso a casos passados de estratégias bem sucedidas. A memória semântica, ou seja, o conhecimento que o agente possui acerca do funcionamento do ambiente, também poderá evoluir. Podem ser definidas estruturas que permitem ao agente, por exemplo, reconhecer que um conjunto de obstáculos podem constituir um edifício, ou como usar uma porta;
- Aumentar a complexidade das decisões que o agente pode tomar. O principal objectivo será a definição de estratégias de equipa que revelem uma maior cooperação e inteligência;
- Completar o modelo emocional através da implementação ou aperfeiçoamento de vários conceitos: existência de mistura de emoções e sentimentos, melhoria do sistema de associação entre emoções e comportamentos, e abordagem mais aprofundada ao conceito de personalidade.

# Bibliografía

- [1] Microsoft Developer Network.  
<http://msdn.microsoft.com>.
- [2] Soar Home Page.  
<http://sitemaker.umich.edu/soar/home>.
- [3] John Arnold.  
Emotional Modeling with Synthetic Agents.  
2003.
- [4] Milind Tambe Gal A. Kaminka.  
Robust Agent Teams via Socially-Attentive Monitoring.  
2000.
- [5] Carnegie Mellon University Information Sciences Institute, University of Southern California; Computer Science Department.  
Gamebots: A 3D Virtual World Test-Bed For Multi-Agent Research.  
2001.
- [6] Information Sciences Institute and University of Southern California Computer Science Department.  
Building agent teams using an explicit teamwork model and learning.  
1998.
- [7] Nicholas R. Jennings.  
On agent-based software engineering.  
1999.
- [8] Paul Rosenbloom Jill Fain Lehman, John Laird.  
A Gentle Introduction To Soar, An Architecture For Human Cognition.  
2006.
- [9] Nicholas R. Jennings Michael J. Wooldridge.  
Intelligent Agents: Theory and Practice.  
1995.
- [10] Juan D. Velásquez.  
Modeling Emotions and Other Motivations in Synthetic Agents.  
1997.