



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

MibSim: Uma proposta colaborativa de robôs autônomos

Yuri Vitor Pourre

Orientadora
Kate Cerqueira Revoredo

RIO DE JANEIRO, RJ – BRASIL

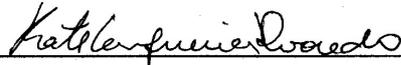
DEZEMBRO DE 2014

MibSim: Uma proposta colaborativa de robôs autônomos

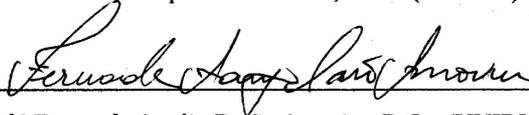
Yuri Vitor Pourre

Projeto de Graduação apresentado à Escola de
Informática Aplicada da Universidade Federal do
Estado do Rio de Janeiro (UNIRIO) para obtenção do
título de Bacharel em Sistemas de Informação.

Aprovado por:



Prof.ª Kate Cerqueira Revoredo, D.Sc. (UNIRIO)



Prof.ª Fernanda Araújo Balão Amorim, D.Sc. (UNIRIO)



Prof. Alexandre Luis Correa, D.Sc. (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

DEZEMBRO DE 2014

Agradecimentos

Gostaria de agradecer a todos que me ajudaram a chegar até aqui.

Aos funcionários da UNIRIO do setor administrativo que desde o primeiro contato me auxiliaram com um sorriso no rosto.

Aos professores da UNIRIO que me fizeram enxergar a graduação de uma maneira mais positiva.

À orientadora Kate pelo seu empenho e incentivo para que eu seguisse a carreira acadêmica.

Aos amigos que ouviram incansáveis, promessas pós-formatura.

Aos meus pais pelo incentivo diário.

À minha noiva Yuri principalmente pela paciência.

RESUMO

O trabalho apresentado a seguir apresenta uma sugestão de como melhorar as soluções utilizadas atualmente para a exploração (ou atuação) de robôs autônomos em ambientes desconhecidos utilizando técnicas de Inteligência Artificial como o planejamento e a colaboração entre agentes. O trabalho descreve uma proposta de processo de exploração implementado na forma de um simulador para cenários onde existe um local no ambiente conhecido pelos agentes atuantes e também considera locais de parada caso seja necessário.

A proposta pode ser usada em diversas situações reais que hoje são realizadas sem qualquer tipo de coordenação automatizada como por exemplo no gerenciamento de frotas de caminhões ou na limpeza de uma casa por meio de robôs aspiradores de pó.

Palavras-chave: Simulador, exploração, planejamento, colaboração, inteligência artificial.

ABSTRACT

The work presented below presents a suggestion to improve the solutions currently used for the exploration (or actuation) of autonomous robots in unknown environments using Artificial Intelligence techniques as planning and collaboration between agents. The work describes a proposal for exploration process implemented as a simulator for scenarios where there is a place known by active agents, also considers stopping places if necessary.

The proposal can be used in several real world applications that today are performed without any kind of automated coordination such as the management of fleets of trucks or cleaning a house through vacuum cleaner robots.

Keywords: Simulator, exploration, planning, collaboration, artificial intelligence.

Sumário

1 Introdução.....	10
1.1 – Motivação.....	10
1.2 – Objetivos.....	12
1.3 – Organização do texto.....	12
2 Fundamentação Teórica.....	13
2.1 – Hierarquia de Necessidades de Maslow.....	13
2.2 – Inteligência Artificial.....	16
2.3 – Representação do Conhecimento.....	16
2.4 – Planejamento.....	18
2.5 – Percepção.....	20
2.6 – Agentes.....	20
2.7 – Sistemas Multiagentes.....	21
2.8 – Colaboração.....	22
3 Descrição da Proposta.....	23
3.1 – Ambiente.....	24
3.2 – Bases estratégicas.....	24
3.3 – Setores.....	25
3.4 – Perímetro de Exploração.....	26
3.5 – Agentes.....	27
3.6 – Espécies.....	30
3.7 – Nutrientes.....	33
3.8 – Fontes de Nutrientes.....	33
4 Funcionamento do Sistema.....	41
4.1 – Inicialização.....	41
4.2 – Rodada de Execução.....	42
4.3 – Comportamento dos Agentes.....	43

4.4 – Barra de Nutrientes.....	45
4.5 – Tarefas de Planejamento.....	46
4.6 – Designação.....	46
4.7 – Fim da simulação.....	47
4.8 – Possíveis Cenários.....	47
5 Conclusão.....	50
Trabalhos Futuros.....	51

Índice de Figuras

Figura 1: Foto de longa exposição do movimento de um Roomba.....	11
Figura 2: Representação piramidal da Hierarquia de Necessidades de Maslow.....	14
Figura 3: Representação de uma matriz usando tiles.....	17
Figura 4: MibSim como matriz de números.....	18
Figura 5: Imagem do MibSim em execução.....	24
Figura 6: Imagem de uma base Ultralisk.....	25
Figura 7: Ilustração do primeiro perímetro de exploração.....	26
Figura 8: Ilustração dos primeiros perímetros de exploração.....	26
Figura 9: Representação do fim do ciclo de vida.....	29
Figura 10: Arte Conceitual de um Ultralisk.....	30
Figura 11: Arte Conceitual de um Hydralisk.....	31
Figura 12: Arte Conceitual de um Lurker.....	32
Figura 13: Diagrama de Casos de Uso.....	39
Figura 14: Diagrama UML do MibSim.....	40
Figura 15: Configuração dos valores iniciais.....	41
Figura 16: Posicionamento dos agentes.....	42
Figura 17: Raio de alcance do sensor.....	44
Figura 18: Modelo de Transição de Estados.....	45
Figura 19: Barra de Nutrientes.....	46
Figura 20: Hydralisk voltando para a base.....	47

Índice de Tabelas

Tabela 1: Relação entre Ações e Consumo Metabólico.....	29
Tabela 2: Atributos de um Ultralisk.....	31
Tabela 3: Atributos de um Hydralisk.....	32
Tabela 4: Atributos de um Lurker.....	33

Lista de Abreviaturas

NASA – Administração Nacional do Espaço e da Aeronáutica

I.A. – Inteligência Artificial

2D – Duas dimensões

UML – Unified Modeling Language

1 Introdução

O sistema descrito ao longo deste trabalho, chamado de *Mibsim*, consiste em um simulador de um ambiente virtual com regras bem definidas, onde seres fictícios caminham e procuram por nutrientes.

Essa proposta apresenta um sistema colaborativo de exploração com uma temática de ficção científica alienígena baseada principalmente em literaturas que surgiram ao redor do jogo *Starcraft* da empresa *Blizzard Entertainment*, lançado em Março de 1998[26]. Apesar da abordagem fictícia, os conceitos empregados no sistema podem ser utilizados em situações reais de exploração onde seja possível coordenar diversos agentes por um ambiente mesmo que desconhecido, onde todos conheçam a posição de uma base fixa.

Essa situação não é tão incomum assim no mundo real e teria grande utilidade na exploração de Marte, por exemplo, realizada pela NASA utilizando os robôs gêmeos, *Spirit* e *Opportunity* (lançados em 10 de Junho e 7 de Julho de 2003 respectivamente). Essa missão pode ser descrita como uma missão em que dois robôs com as mesmas características são lançados a um planeta desconhecido para pousar em uma posição pré-determinada (e conhecida pelos robôs)[30].

1.1 – Motivação

Robôs autônomos vêm surgido a uma velocidade incrível, muitos estão realizando tarefas domésticas outros estão sendo usados em linhas de produção na indústria, mas a maioria deles não aproveita a presença de outros indivíduos semelhantes para dividir ou otimizar as tarefas.

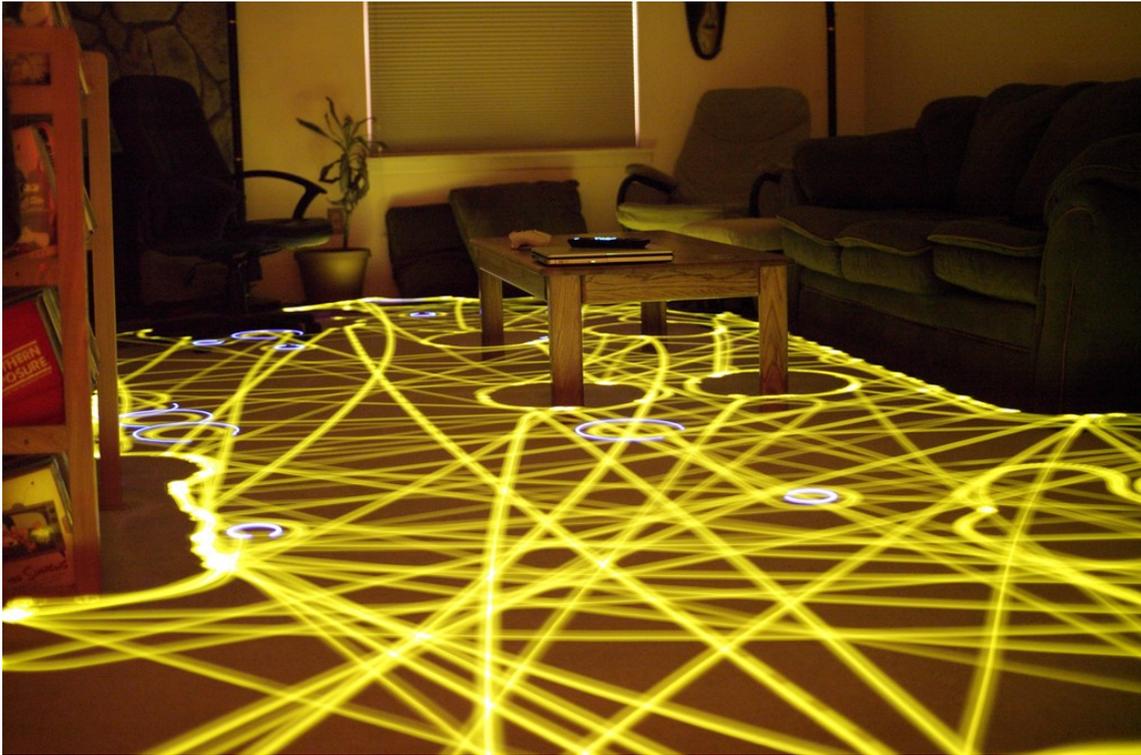


Figura 1: Foto de longa exposição do movimento de um Roomba

A figura 1 mostra a trajetória feita por um aspirador de pó autônomo durante a sua tarefa de aspirar esta casa. O robô responsável por executar essa trajetória foi um Roomba, um robô aspirador de pó mais famoso da atualidade, ele atua sozinho e não depende da base ou de outros robôs para tomar decisões ou planejar a rota de limpeza[9]. Como consequência deste trabalho independente mesmo que haja outros robôs iguais a ele no mesmo recinto, fazendo o mesmo trabalho, eles não conseguem interagir entre si, portanto não podem determinar a melhor maneira de aspirar a casa ou evitar que passem por lugares em que outros já passaram, economizando energia e agilizando o serviço. Cada um trata os demais como meros obstáculos, desperdiçando a oportunidade de atuar em conjunto.

Com o uso das propostas do MibSim, seria possível mapear a casa e ainda evitar que os Roombas passassem muitas vezes pelos mesmos lugares, economizando bateria e tempo.

O MibSim pode ser usado em situações reais onde é preciso mapear um ambiente desde que os agentes que vão atuar sobre o ambiente conheçam um ponto fixo neste ambiente. Uma aplicação prática do sistema - não tão distante quanto Marte - poderia ser a coordenação da rotina de um grupo de robôs autônomos que aspiram o pó

do chão de uma residência e precisam passar por todos os lugares acessíveis sem que sua bateria acabe no meio de uma dessas limpezas. O requisito para aplicar o sistema pode ser atendido já que os modelos mais modernos de aspiradores autônomos conhecem a posição de uma base fixa onde retornam para recarregar sua(s) bateria(s) quando estão com pouca energia [6].

O sistema proposto foi desenvolvido levando em consideração também o fato de que a base desses robôs normalmente fica ligada na tomada, então a maior parte do processamento deve ser feita na base, principalmente para economizar a bateria dos robôs autônomos.

1.2 – Objetivos

Pensando sobre como melhorar as soluções utilizadas atualmente para a exploração (ou atuação) de robôs autônomos em ambientes desconhecidos, decidi especificar um sistema que utiliza técnicas de Inteligência Artificial como a colaboração entre agentes para otimizar essas tarefas.

1.3 – Organização do texto

O presente trabalho está estruturado em capítulos e, além desta introdução, será desenvolvido da seguinte forma:

- Capítulo II: Trata de toda a fundamentação teórica para embasar o leitor sobre os assuntos que serão abordados ao longo deste trabalho.
- Capítulo III: Define a proposta do sistema *MibSim*, e explica os termos e técnicas utilizadas.
- Capítulo IV: Mostra o sistema em funcionamento e situações não convencionais que podem ocorrer dependendo da configuração dos elementos do ambiente.
- Capítulo V: Conclusões – Reúne as considerações finais, assinala as contribuições da pesquisa e sugere possibilidades de melhorias e aprofundamento posterior.

2 Fundamentação Teórica

Este capítulo aborda assuntos importantes para o entendimento do sistema proposto, como: A hierarquia de necessidades de Maslow, subcampos da Inteligência Artificial como Planejamento, Percepção e Representação do Conhecimento.

2.1 – Hierarquia de Necessidades de Maslow

A Hierarquia de Necessidades de Maslow [1] é uma teoria da psicologia proposta por Abraham Maslow. Maslow diz que as necessidades humanas podem ser organizadas seguindo uma hierarquia a qual ele chamou de hierarquia da motivação humana (chamada posteriormente de hierarquia de necessidades de Maslow). A hierarquia de necessidades de Maslow é comumente representada em forma de pirâmide com os maiores e mais fundamentais níveis de necessidades na base, e a necessidade de auto-realização no topo[2]. Uma necessidade é substituída pela seguinte (de um nível mais alto na hierarquia), a medida em que começa a ser satisfeita. Assim, por ordem crescente de prioridade, as necessidades são classificadas em: fisiologia, segurança, afiliação, estima e auto-realização.

Como o ser-humano é um ser complexo, diferentes motivações dos vários níveis da hierarquia de necessidades de Maslow podem manifestar-se ao mesmo tempo. O próprio Maslow referia-se aos níveis utilizando termos como: relativo, geral, e primário; ao invés de falar com o foco em uma única necessidade por vez[2].



Figura 2: Representação piramidal da Hierarquia de Necessidades de Maslow

(Adaptado de: http://www.photaki.es/foto-maslow-piramide_612936.htm)

2.1.1 – Níveis da Hierarquia de Necessidades

Nível de Fisiologia: São requisitos físicos para a sobrevivência humana. Se esses requisitos não forem satisfeitos, o corpo humano não vai conseguir funcionar adequadamente. As necessidades fisiológicas são as necessidades mais urgentes, importantes e precisam ser satisfeitas primeiro. Ar, água e comida são requisitos metabólicos necessários para sobrevivência em todos os animais, incluindo os humanos. Roupas e abrigo também são considerados necessidades neste primeiro nível da hierarquia[1].

Nível de Segurança: Com o seu nível fisiológico relativamente satisfeito, a necessidade por segurança ganha precedência e dominância no comportamento do indivíduo. Na ausência de segurança física – provinda de guerra, desastre natural, violência na família, etc. - as pessoas podem experimentar estresse pós-traumático. Este nível é mais evidente em crianças pois elas não escondem a necessidade de sentirem-se

seguras[1]. Necessidades de segurança incluem: segurança pessoal, segurança financeira, saúde e bem-estar, e segurança contra acidentes/doenças.

Nível de Afiliação: Depois dos níveis fisiológico e de segurança satisfeitos, vai emergir o terceiro nível, das necessidades humanas interpessoais e que envolve sentimentos de pertencimento. De acordo com Maslow, humanos precisam ter a sensação de pertencimento e aceitação de seus grupos sociais, não importando se são grupos grandes ou pequenos. Humanos precisam amar e ser amados – sexualmente ou não-sexualmente – pelos outros[1]. Deficiências neste nível da hierarquia de Maslow – como hospitalismo, negligência, ostracismo, etc – podem impactar na habilidade de formar e manter emocionalmente relacionamentos emocionais em geral, como: amizade, intimidade e família.

Nível de Estima: Todos os humanos tem necessidade de sentirem-se respeitados; isto inclui necessidade de auto-estima e auto-respeito. Estima representa o típico desejo humano de ser aceito e valorizado pelos outros. O nível de estima pode ser dividido em dois conjuntos: o primeiro, o desejo por força, alcançar objetivos, confiança em encarar o mundo e busca por independência e liberdade; o segundo grupo é formado pelo anseio por reputação, reconhecimento, atenção, importância[1]. As pessoas em uma profissão ou hobby buscam por reconhecimento. Essas atividades dão à pessoa a sensação de contribuição ou valor. Baixa auto-estima ou complexo de inferioridade podem ser resultados do desbalanceamento deste nível hierárquico.

Nível de Realização pessoal: “O que um homem tem potencial para ser”. Esta citação forma a base da necessidade por auto-realização. Este nível de necessidade refere-se a o que uma pessoa consegue realizar em seu máximo potencial. Maslow descreve este nível como o desejo de realizar tudo o que for possível, para se tornar o máximo que puder. Esse desejo específico pode variar para cada indivíduo. Por exemplo, um indivíduo pode ter um enorme desejo de ser um grande inventor, outros, podem expressar esse desejo na forma de artes[2]. Como mencionado anteriormente, Maslow acreditava que para entender esse nível de necessidade a pessoa não deveria apenas atingir os níveis anteriores mas satisfazê-los plenamente.

2.2 – Inteligência Artificial

Inteligência Artificial é uma área de conhecimento que envolve três disciplinas acadêmicas: psicologia, filosofia e ciência da computação[3]. Sustentada por esses três alicerces, pode ser descrita como a atividade de desenvolver algoritmos de computação baseados no comportamento de seres vivos[10] na maioria das vezes com o objetivo de resolver problemas práticos do mundo-real.

O problema geral de simular (ou criar) inteligência foi quebrado em inúmeros sub-áreas de pesquisas, as principais são: raciocínio, representação do conhecimento, planejamento, aprendizado, percepção, processamento de linguagem natural, e a habilidade de manipular objetos[4] os tópicos a seguir abordam as áreas de Inteligência Artificial utilizadas no sistema proposto.

2.3 – Representação do Conhecimento

Representação do Conhecimento é um subcampo da Inteligência Artificial, que como o nome sugere, é voltada para a representação. Existem diversas maneiras de representar o conhecimento, mas para o entendimento da proposta, é necessário apenas entender como é possível representar um ambiente bidimensional e pontos espaciais utilizando matrizes.

2.3.1 – Representação de um ambiente bidimensional

Uma maneira de representar um ambiente 2D (duas dimensões) é utilizar uma matriz bidimensional com marcações para identificar os objetos interessantes.

Um estilo de jogos de computador que ficou muito famoso no passado é o chamado *rogue-like*, uma característica desse estilo é representar todo o cenário utilizando apenas matrizes. A exibição para o jogador era feita no próprio terminal, mostrando caracteres relativos aos pontos importantes como os personagens, inimigos, itens, paredes.

Em um exemplo hipotético, poderíamos ter a seguinte matriz M:

$$M = \begin{pmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 8 & 3 & 0 & 0 & 8 & 0 & 8 \\ 8 & 0 & 1 & 0 & 8 & 0 & 8 \\ 8 & 8 & 8 & 8 & 8 & 8 & 8 \end{pmatrix}$$

Nesse caso, cada número seria a representação de algum conceito, o número 8 poderia representar uma área bloqueada como uma pedra ou a parte de uma parede, o número 1 poderia ser o personagem e o número 3 um inimigo qualquer. Dessa forma, poderíamos representar um jogo inteiro usando apenas alguns números.

2.3.2 – Representação gráfica de um mapa (tiles)

Com as melhorias provindas pelo avanço da tecnologia, e consequente melhoria da capacidade de processamento dos computadores, a exibição das matrizes não precisam mais ser feitas utilizando letras e números em um terminal de fundo preto, ao invés disso, os caracteres são renderizados na forma de imagens, mesmo que internamente o programa ainda continue utilizando as matrizes de números. Uma técnica para desenhar elementos que repetem-se constantemente chama-se renderização por tiles, o que se faz é dividir a tela do computador em uma espécie de matriz com colunas e linhas de blocos de tamanhos iguais chamados *tiles* (ou azulejos na tradução literal), ao ler os números da matriz, o programa renderiza as imagens dos tiles seguindo a mesma ordem[31].

Por exemplo: na matriz M , onde encontramos o valor 0, poderíamos exibir o tile associado ao número 0 que é o bloco verde (onde o personagem pode andar por cima), onde encontramos o número 8, poderíamos renderizar o tile de água, e assim por diante. A imagem formada seria análoga à imagem vista na figura 3.

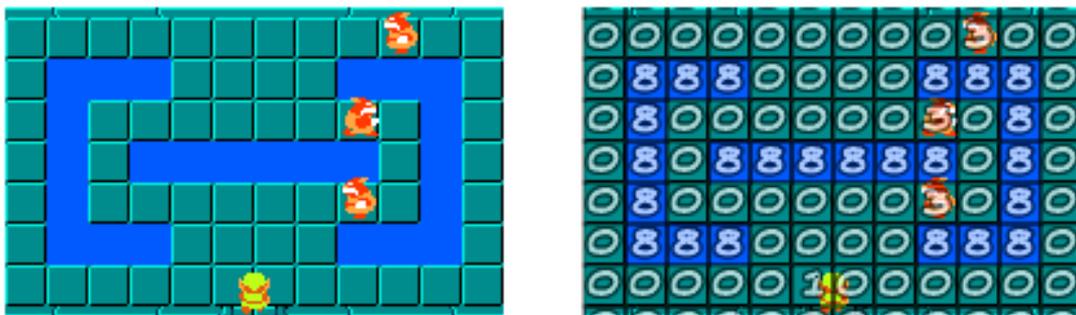


Figura 3: Representação de uma matriz usando tiles.

(Adaptado de:

<http://pobierzgryprogramy.pl/PobierzGryProgramy.pl/upload/screenshot2357-3.gif>)

No MibSim, temos uma representação muito similar à apresentada na figura 4.

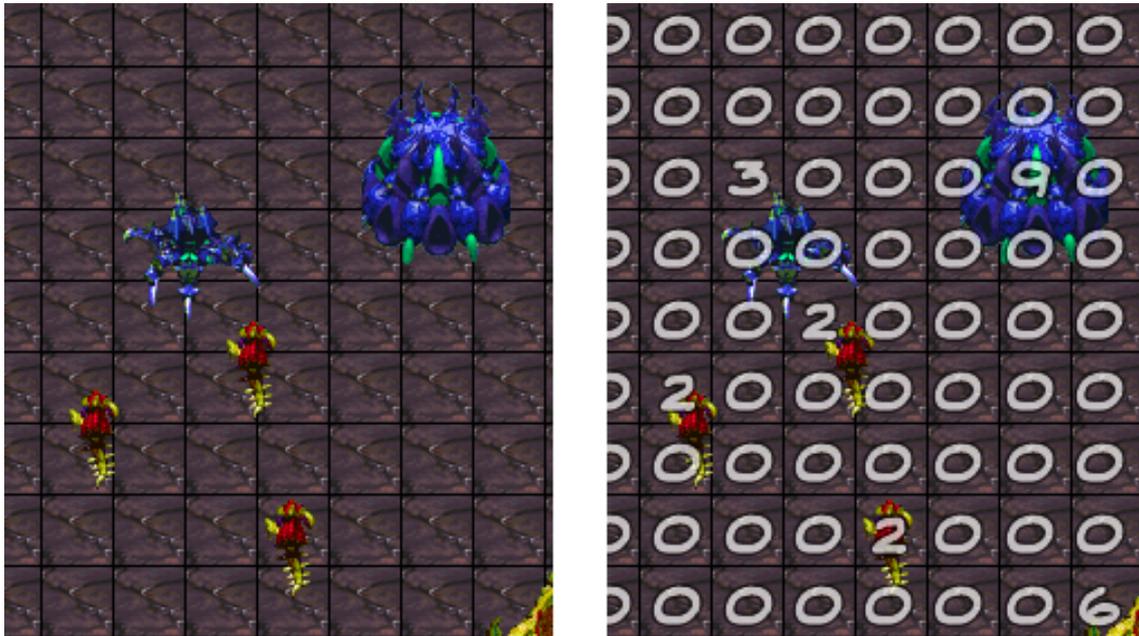


Figura 4: MibSim como matriz de números

2.4 – Planejamento

Planejamento autônomo ou simplesmente planejamento é um ramo da inteligência artificial que consiste em partir de uma situação inicial e realizar estratégias ou sequências de ações para atingir um objetivo, é tipicamente usada por agentes inteligentes, robôs autônomos e veículos não tripulados. Diferente do controle clássico e problemas de classificação, as soluções são complexas e devem ser descobertas e otimizadas em um espaço multidimensional. Planejamento também está relacionado com teoria de decisão. As soluções resultam de um processo iterativo de tentativa e erro comumente visto em I.A..

Dada uma descrição de possíveis estados iniciais do mundo, uma descrição de objetivos desejados, e uma descrição do conjunto de possibilidades de ações, o problema de planejamento é encontrar o plano que é garantido (de qualquer estado inicial) de gerar uma sequência de ações que leva a um dos objetivos finais[7].

O problema mais simples possível de planejamento conhecido como problema clássico de planejamento, é determinado por:

- um único estado inicial conhecido;

- ações sem duração;
- ações determinísticas, e que só podem ser executadas uma de cada vez;
- um único agente;

Com apenas um único estado inicial, o planejamento sempre parte da mesma configuração inicial, e caso isso não ocorresse, seria necessário voltar à configuração inicial desejada no planejamento ou ter um planejamento para cada configuração inicial diferente.

Lidar com ações sem duração significa não preocupar-se em ter que esperar o fim de uma ação para começar a próxima.

Ações determinísticas, são ações as quais seus efeitos podem ser observados e/ou avaliados, desta maneira é possível prever o resultado da ação. Se as ações só puderem ser executadas uma de cada vez, não é necessário preocupar-se com paralelismo e principalmente se uma ação pode interferir na outra caso ocorram simultaneamente.

Um único agente não precisa colaborar ou tomar decisões baseado nas percepções de outrem se estiver sozinho durante o planejamento.

No exemplo dado, se o nosso robô estivesse sentado, por exemplo, ele precisaria levantar (chegar ao estado inicial desejado), para que a sequência de ações funcionasse corretamente.

Um exemplo desse cenário mais simplista de planejamento poderia ser um agente “calculadora” que precise calcular o valor de uma expressão;

O estado inicial consiste em começar com o valor 0.

Se pudermos desprezar o tempo que as operações aritméticas levam para ser feitas, temos ações sem duração.

Como podemos prever os resultados, temos as operações aritméticas como ações determinísticas. Por ordem de precedência da expressão temos que as ações só podem ser executadas uma após a outra (e não de forma concorrente). Como o agente atuaria sozinho e não precisaria interagir com outros, este exemplo satisfaria o cenário de problema de planejamento mais simples possível.

2.5 – Percepção

Percepção da máquina é a capacidade de usar a entrada de dados por meio de sensores (como câmeras, microfones, sensores táteis, sonar e outros mais exóticos) para deduzir os aspectos do mundo[4].

Em um ambiente virtual, os sensores são “enganados” para receber dados que parecem reais mas que são gerados por esse ambiente, no sistema proposto os agentes percebem o mundo ao ser redor por um único sensor de formato circular e com um alcance pré-definido.

2.6 – Agentes

A definição mais simples de agentes e a que melhor se encaixa neste trabalho é: indivíduo ou conjunto de indivíduos que percebem o ambiente por meio de sensores e interagir com o ambiente por meio de atuadores[4]. Do ponto de vista computacional, o agente é um processo autônomo que comunica-se com uma aplicação[11].

Outras características que descrevem um agentes e separam das demais entidades de software são:

Autonomia: Capacidade de tomar decisões e agir sem interferência direta de agentes humanos ou outros agente computacionais, para que isso seja possível, o agente precisa ter alguma inteligência de forma a perceber o ambiente e garantir sua sobrevivência.

Habilidade social: Característica dos agentes de trocar informação com outros agente, sejam eles humanos ou computacionais a afim de melhorar sua capacidade de resolução de problemas ou até ajudar outros agentes em suas tarefas. Esta habilidade requer que os agentes saibam como e quando interagir e trocar essas informações, este “saber” pode ser previamente programado ou adquirido ao longo da simulação;

Reação: A partir da percepção dos sensores o agente deve ser capaz de reagir aos estímulos do ambiente;

Continuidade Temporal: Durante a simulação, os agentes são processados de forma contínua, e podem estar tanto ativos, em *foreground*, como adormecidos, em *background*;

Iniciativa: Os agentes não reagem somente ao seu ambiente, também apresentam um comportamento orientado à satisfação de seus objetivos;

Orientação a Objetivos: Um agente deve ser capaz de lidar com tarefas complexas em alto nível. O modo que essa tarefa será fragmentada em subtarefas menores e a ordem dessas subtarefas cabe ao próprio agente decidir;

Mobilidade: É a habilidade de um agente de movimentar-se pelo ambiente, ocupando diferentes espaços e recursos ao longo do tempo;

Benevolência: É a ideia de que o agente não possui objetivos conflituosos e que cada agente irá sempre tentar fazer o que lhe for pedido;

Visão Local: Nenhum agente tem uma visão global do sistema, ou o sistema é complexo demais para um agente fazer uso prático desse conhecimento.

Outras características que também merecem destaque são a racionalidade (hipótese que o agente sempre vai agir para alcançar seus objetivos e não o oposto), adaptabilidade (capacidade do agente se adaptar a possíveis mudanças no ambiente) e colaboração (capacidade de trabalhar colaborativamente com outros agentes, avaliando possíveis enganos especialmente nos comandos provenientes dos agentes humanos).

Auto Organização: Os agentes não precisam de fontes externas para se organizarem.

2.7 – Sistemas Multiagentes

Sistemas multiagentes consistem em agentes e seu ambiente. Tipicamente pesquisas sobre sistemas multiagentes referem-se a agentes de software. Entretanto, os agentes em sistemas multiagentes podem ser robôs, humanos ou grupos de humanos. Um sistema multiagentes pode ter uma combinação entre humanos e agentes onde todos podem interagir com o ambiente.

Os ambientes dos sistemas podem ser: acessíveis ou inacessíveis, estáticos ou dinâmicos, determinístico ou não-determinístico, discretos ou contínuos, e episódico ou não-episódico.

Ambiente Acessível: Ambiente cujo agente consegue perceber seu estado completo por meio de sensores.

Ambiente Estático: O ambiente não muda enquanto um agente está decidindo qual ação tomar.

Ambiente Determinístico: O estado seguinte do ambiente pode ser completamente determinado a partir de seu estado corrente e as ações tomadas pelos agentes.

Ambiente Discreto: Um ambiente discreto é aquele que só pode assumir um conjunto predefinido de estados. Desta forma, o xadrez, damas e outros jogos de tabuleiro, constituem ambientes discretos pois existe um número finito(embora muito grande) de estados possíveis do tabuleiro;

Ambiente Episódico: Quando a experiência do agente é dividida em episódios. Cada episódio consiste em perceber e então reagir. Os episódios não afetam os próximos.

2.8 – Colaboração

Colaborar significa cooperar para a realização de qualquer coisa [23]. Colaboração pode ser entendida como o trabalho em conjunto para atingir um objetivo comum.

A colaboração pode ser usada para concluir qualquer tipo de tarefa onde mais de um indivíduo possa realizar simultaneamente. Como montar o mapa com as ruas de uma cidade, por exemplo, se todas as pessoas de uma cidade resolvessem elaborar um mapa da cidade onde moram usando apenas os conhecimentos de cada um.

Poderia haver uma grande folha em branco no centro da cidade onde cada um poderia desenhar as ruas no mapa, nesse caso, o conhecimento ficaria centralizado, para terem acesso ao mapa, cada pessoa teria que ir ao local onde está o mapa para ter a versão atualizada. Ou cada um poderia ter a sua própria folha em branco e conforme as pessoas fossem se encontrando nas ruas, poderiam pedir que uma adicionasse informações no mapa da outra, nesse caso o conhecimento seria descentralizado e cada pessoa teria uma versão diferente do mapa e com redundância de informações.

Para evitar que a mesma parte do mapa fosse feita por mais de uma pessoa, precisaria haver uma coordenação entre os participantes para definir quem seria o responsável por reunir as informações de qual parte do mapa.

Usando a colaboração, a longa tarefa de fazer o mapa da cidade poderia ser dividida dentre as várias pessoas, agilizando o processo de mapear as ruas de uma cidade.

3 Descrição da Proposta

A proposta apresentada a seguir descreve uma maneira de utilizar colaboração entre agentes para a resolução de tarefas com o foco em colaboração, o modelo utilizado é genérico o suficiente para que com poucas mudanças possa ser aplicado em diferentes situações, que vão desde a limpeza de uma residência feita por robôs aspiradores de pó até explorações espaciais em planetas ainda desconhecidos pela raça humana. O único pré-requisito para que o modelo funcione é que haja um ponto no mapa conhecido previamente pelos agentes.

Utilizando os conceitos de planejamento, colaboração e inteligência artificial, foi desenvolvido o MibSim, um simulador de ambientes virtuais, com regras bem definidas, populado por diversos agentes exploradores de diferentes espécies que interagem com o ambiente, com o intuito de alimentar-se e explorar todo o ambiente.

Os agentes colaboram para fazer o reconhecimento do mapa reportando as fontes que forem encontradas durante o trajeto para a base estratégica, a qual centraliza o conhecimento acerca das fontes conhecidas pelos agentes.

Os algoritmos de decisão foram feitos de maneira que não produzam nenhuma saída aleatória (ou não-determinística), desta maneira, partindo de um estado inicial do sistema, acontecerá sempre a mesma cadeia de eventos. Desse modo, é possível analisar determinadas situações revendo os estados pelos quais a simulação passou anteriormente.

Cada agente pertence a uma espécie específica e cada espécie possui uma base estratégica, que é o lugar onde todos os agentes precisam ir para receberem suas designações e saberem qual ponto do mapa precisa ser explorado pela sua espécie. Depois de reconhecer o setor, os agentes precisam voltar para a base reportando o que foi encontrado na área designada.

O ambiente é dividido em setores, e cada espécie delimita o ambiente utilizando a posição de sua base como o setor central, espécies diferentes delimitam o cenário de

maneira diferente. A descoberta das fontes pelos agentes se dá por proximidade. Um agente percebe uma fonte quando passa por ela por uma distância menor do que o alcance do seu sensor.

A execução da simulação é feita por rodadas/turnos. A cada turno da simulação todos os agentes tomam uma ação baseada em suas próprias avaliações, cada possível ação gera um decréscimo a sua reserva de nutrientes e se essa reserva se esgotar completamente, o ciclo de vida do agente chega ao fim (vide seção 3.5.4).

3.1 – Ambiente

O ambiente do simulador *MibSim* é um ambiente virtual discreto, representado por um enorme mapa formado por *tiles* (azulejos pela tradução literal), e é nesse mapa onde se encontram os agentes, as bases, as fontes e acontece toda a simulação.



Figura 5: Imagem do MibSim em execução

3.2 – Bases estratégicas

As bases estratégicas (ou apenas bases) são estruturas fixas dispostas ao longo do mapa que servem como repositório de conhecimento. Cada espécie possui uma única

base cuja localização é conhecida por todos os indivíduos daquela espécie desde a inicialização do sistema. A base é um lugar de encontro para os agentes onde eles podem reportar os resultados da sua exploração e ter ciência sobre a posição de fontes descobertas por outros agentes. A base também é responsável por coordenar os agentes designando-os para explorar setores ainda desconhecidos. Como todos os agentes conhecem o setor onde a base se encontra, as primeiras designações da base estratégicas são feitas aos setores adjacentes à base (também chamado de primeiro perímetro).



Figura 6: Imagem de uma base Ultralisk

3.3 – Setores

O ambiente é dividido pelas espécies de *Mibsim*s em Setores. Cada espécie divide o ambiente em setores que ocupam blocos iguais de área 3x3 *tiles*. O que difere na divisão de cada espécie é a numeração dos setores que é feita de forma relativa à base. O primeiro setor é conhecido por todos os agentes de uma mesma espécie é o bloco da base estratégica (ou setor base), ele representa o centro relativo do mapa e serve como referência para as explorações. Para fins de orientação, os setores recebem uma numeração composta por dois números e um hífen separando-os o primeiro número indica o perímetro, e o segundo indica a ordem do setor no perímetro. A numeração dos setores é feita sempre em sentido horário, começando do setor mais acima e mais a esquerda. Mesmo que um setor esteja em um lugar inalcançável do mapa ele é considerado na numeração, pois sem mapear todo o mapa, os agentes não têm como diferenciar a borda do mapa de um obstáculo que ocupe todo o setor.



Figura 7: Ilustração do primeiro perímetro de exploração

3.4 – Perímetro de Exploração

Os perímetros de exploração são os setores adjacentes aos setores do último perímetro de exploração conhecido por completo pela base. No início da simulação como nenhum setor foi explorado ainda, a base considera-se como o último perímetro conhecido por todos, sendo assim, o primeiro perímetro de exploração são os 8 setores ao redor da base.

Por causa de sua forma quadrada, o perímetro de exploração sempre é formado por um número de setores múltiplo de 8. Ex: 8, 16, 24...

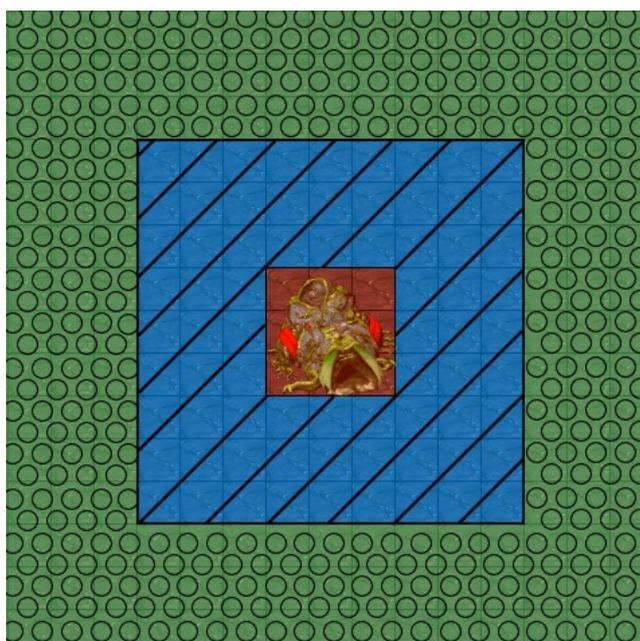


Figura 8: Ilustração dos primeiros perímetros de exploração

Legenda:

-  Base
-  1º perímetro
-  2º perímetro

3.5 – Agentes

Os agentes são os indivíduos que “habitam” o simulador *MibSim*, representados graficamente como alienígenas de aparência estranha[14]. Cada agente pertence a uma espécie e todos os agentes de uma mesma espécie possuem aparência semelhante. Os agentes percebem o ambiente por meio de sensores de atuação (de formato circular) que captam as fontes ao seu redor.

No mundo real, percebemos o mundo ao nosso redor por meios dos sentidos: tato, audição, paladar, olfato e visão. No mundo virtual, não precisamos lidar com todos esses “sensores”. Para lidar com menos valores sem que se comprometa o resultado da simulação, o *Mibsim* abstrai os cinco sentidos dos agentes e apenas considera como sensor, um campo de visão de 360 graus limitado por um raio em pixels, medido a partir do centro do agente. Desta maneira, o indivíduo percebe e interage apenas com as fontes que estão dentro desse raio.

Baseado na hierarquia de necessidades de Maslow, a prioridade de cada agente é sobreviver, para tanto os agentes precisam procurar e consumir nutrientes (encontrados nas fontes). Quando sua fome está saciada, os agentes voltam-se para a exploração do mapa, dirigindo-se para a base para receber suas designações e retornando à base para reportar o que foi encontrado ao explorar o setor designado, comunicando sobre o posicionamento de fontes fixas descobertas nessa exploração, aumentando as chances de sobrevivência da espécie.

3.5.1 – Atributos Básicos

As características básicas dos agentes (alcance dos sensores, capacidade da reserva de alimentos, velocidade de movimentação e o limiar de fome) são determinadas pela espécie a qual ele pertence, essas características são iguais em todos os indivíduos de uma mesma população/espécie. Nenhum dos atributos básicos é alterado ao longo da simulação.

Velocidade: Cada agente tem um atributo que determina a velocidade de seus movimentos.

Raio dos Sensores: Valor que determina o alcance dos sensores em pixels.

Metabolismo: Velocidade com que os nutrientes são consumidos pelos *Mibsim*s para realizar as ações.

Capacidade da Reserva de Nutrientes: a capacidade total de armazenamento de nutrientes. Essa capacidade está diretamente ligada à autonomia da espécie, a quantidade de ações que um agente consegue tomar sem precisar parar para reabastecer.

Valor Atual da Reserva de Nutrientes, a quantidade corrente de nutrientes na reserva.

Limiar de Fome: Valor que se atingido pelo valor atual da reserva de nutrientes, faz com que o agente desista temporariamente do seu plano de exploração para ir até a fonte de nutrientes mais próxima.

3.5.2 – Movimentação

Os agentes sempre movimentam-se em linha reta indo da posição onde estão até o alvo, a primeira tarefa antes de começar a andar é calcular o ângulo até o alvo, depois, o agente gira para ficar alinhado perfeitamente o alvo e só a partir daí ele anda para frente.

Supondo que um agente de velocidade 5 pixels(px) esteja na posição (100,100) e o alvo seja um setor na posição (200,200), nesse caso, o ângulo do cálculo é 45° pois a distância em x e em y são iguais. Para cada iteração que o agente não tenha atingido o alvo, sua posição é modificada seguindo a fórmula:

$$x = x + \text{velocidade} * \cos(\text{angulo})$$

$$y = y + \text{velocidade} * \sin(\text{angulo})$$

Esse incremento faz com que o agente movimente-se em linha reta até o alvo. Utilizando os valores do exemplo:

$$\sin(45^\circ) = \frac{\sqrt{2}}{2} \quad , \quad \cos(45^\circ) = \frac{\sqrt{2}}{2}$$

Com o agente na posição (100, 100) , temos que na primeira movimentação:

$$x = 100 + 5 * \frac{\sqrt{2}}{2} = 100 + 3,535 = 103,535$$

$$y = 100 + 5 * \frac{\sqrt{2}}{2} = 100 + 3,535 = 103,535$$

Sendo assim, a nova posição do agente será (103,535 ; 103,535). Essa lógica é repetida sucessivas vezes até que o agente chegue até o alvo desejado ou sua reserva de nutrientes chegue ao fim (vide seção 4.3).

3.5.3 – Consumo metabólico

O consumo da reserva é feito com decréscimos na capacidade atual da reserva de nutrientes multiplicado pelo valor do metabolismo em *u.m.* (Unidade de Metabolismo) de acordo com o valor multiplicador da ação realizada. Cada ação possui um valor multiplicador associado.

Tabela 1: Relação entre Ações e Consumo Metabólico

Ação	Multiplicador
Respirar	1
Andar	5
Reportar	10

Em uma espécie de metabolismo 3 *u.m.* por exemplo, perde 15 unidades de metabolismo (5 x 3 *u.m.*) a cada turno que o agente esteja andando.

3.5.4 – Fim do ciclo de vida

O ciclo de vida de um agente chega ao seu fim caso a sua reserva de nutrientes atinja o nível mínimo. Quando isso acontece, este agente é removido do simulador e na posição onde ele se encontrava é adicionada uma fonte **fixa, não-renovável**, com a mesma quantidade de nutrientes que o agente possuía em sua reserva, a reserva criada é do mesmo tipo do nutriente necessário ao agente.



Figura 9: Representação do fim do ciclo de vida

Na figura 9 temos à esquerda um hydralisk faminto e sua barra de energia acima dele marcando a quantidade de nutrientes, a cor vermelha significa que a quantidade de nutrientes está em um nível abaixo do limiar de fome (vide seção 4.4). Na parte à direita da seta, podemos ver o mesmo agente “morto” com sua reserva de alimentos completamente vazia.

3.6 – Espécies

As espécies descritas abaixo tratam de seres fictícios baseados na literatura que surgiu ao entorno do jogo digital *Starcraft* da empresa *Blizzard Entertainment*, lançado em Março de 1998[26].

3.6.1 – Ultralisks

Os ultralisks são a maior espécie presente no *MibSim*, apesar de seu tamanho conseguem mover-se relativamente rápido. Seu corpo é formado por uma dura carapaça, locomove-se utilizando seus dois membros inferiores, seus dois membros superiores possuem, nas extremidades, duas estruturas letais, altamente cortantes[26][27].



Figura 10: Arte Conceitual de um Ultralisk

(Adaptado de: <http://media.blizzard.com/sc2/game/units/zerg/rotate/ultralisk.png>)

Tabela 2: Atributos de um Ultralisk

Cor Predominante:	Verde
Nutriente Necessário:	Água
Capacidade da Reserva:	10.000 u.m.
Limiar de Fome:	3.300 u.m.
Velocidade:	5px
Metabolismo:	3um

3.6.2 – Hydralisks

São os menores seres do *MibSim*, são herbívoros, suas mandíbulas possuem serras, acredita-se ser uma evolução das lagartas. Locomovem-se com o movimento de sua calda. Seus dois membros superiores possuem estruturas afiadas nas extremidades[26][28].



Figura 11: Arte Conceitual de um Hydralisk

(Adaptado de: <http://media.blizzard.com/sc2/game/units/zerg/rotate/hydralisk.png>)

Tabela 3: Atributos de um Hydralisk

Cor Predominante:	Amarelo
Nutriente Necessário:	Água
Capacidade da Reserva:	5.000 u.m.
Limiar de Fome:	3.000 u.m.
Velocidade:	8px
Metabolismo:	2 u.m.

3.6.3 – Lurkers

Movimentam-se sobre 6 pernas, seus espinhos são formados por Adamantium[26][29].



Figura 12: Arte Conceitual de um Lurker

(Adaptado de:

http://img1.wikia.nocookie.net/__cb20100307215811/starcraft/images/0/0e/Lurker_SC1_Art1.jpg)

Tabela 4: Atributos de um Lurker

Cor Predominante:	Azul
Nutriente Necessário:	Adamantita
Capacidade da Reserva:	8.000un
Limiar de Fome:	3.600un
Velocidade:	6px
Metabolismo:	5u.m.

3.7 – Nutrientes

Os nutrientes necessários para a vida dos agentes são:

Água: Líquido composto de hidrogênio e oxigênio, sem cor, cheiro ou sabor, transparente em seu estado de pureza; quimicamente, é formado de dois átomos de hidrogênio e um de oxigênio [23].

Adamantita: Composto metálico de Adamantium puro em sua forma bruta. A estrutura do adamantium é extremamente estável, fazendo com que seja difícil moldá-lo mesmo em temperaturas altas o suficiente para mantê-lo na forma líquida. Em sua forma sólida, é da cor cinza escura metálica [24].

3.8 – Fontes de Nutrientes

As fontes de nutrientes são estruturas espalhadas ao longo de todo o ambiente que são utilizadas pelos agentes com o intuito de encher a própria reserva de nutrientes.

Elas podem ser classificadas quanto a sua mobilidade como: fixas ou móveis. Fontes fixas têm sua posição determinada antes do início da execução e essa posição nunca muda ao longo de toda a simulação, as fontes móveis são os próprios agentes.

As fontes também podem ser classificadas quanto a sua capacidade de produzir nutrientes como: renováveis (aquelas que produzem nutrientes inesgotavelmente) ou não-renováveis (fontes finitas, geradas ao fim do ciclo de vida dos agentes, vide seção 3.5.4). As fontes não-renováveis têm o diferencial de serem removidas do simulador quando seus nutrientes se esgotam.

3.9 – Atores

Usuário: O usuário do sistema, responsável por definir as posições iniciais e iniciar a simulação.

Agentes: Seres virtuais que vivem no MibSim e interagem com o ambiente virtual, as bases estratégicas e com as fontes de nutrientes.

Base Estratégica: Elemento fixo no mapa que decide para onde vai cada agente que solicita um novo setor para ir.

3.10 – Requisitos Funcionais

#Requisito Funcional	Descrição do Requisito Funcional
RF01	Usuário deve poder definir a posição dos elementos no ambiente virtual
RF02	Usuário deve ser capaz de configurar os atributos iniciais dos agentes
RF03	Usuário deve poder iniciar a simulação
RF04	Usuário deve poder finalizar a simulação
RF05	Agente deve poder solicitar uma designação
RF06	Agente deve poder explorar o setor designado
RF07	Agente deve poder alimentar-se de uma fonte de nutrientes
RF08	Base deve ser capaz de designar o agente a um setor

3.11 – Casos de Uso

Aqui serão descritos os casos de usos utilizados para representar as ações do sistema.

3.11.1 – Índice dos Casos de Uso

#Caso de Uso	Nome UC	Ator Principal
UC01	Definir Posição dos Elementos	Usuário
UC02	Configurar Atributos Iniciais	Usuário
UC03	Iniciar Simulação	Usuário
UC04	Finalizar Simulação	Usuário
UC05	Solicitar Designação	Agente
UC06	Explorar Setor Designado	Agente

UC07	Alimentar-se	Agente
UC08	Designar Agente	Base Estratégica

3.11.2 – Descrição dos Casos de Uso

3.11.2.1 – Caso de Uso: UC-01 Definir posição dos Elementos

Descrição: Este caso de uso especifica a ação de posicionar os elementos (fontes, bases estratégicas e agentes) antes do início da simulação.

Ator Principal: Usuário.

Pré-condição: Simulação ainda não tenha sido iniciada.

Pós-condição: Simulador possui configuração feita pelo usuário.

Requisito Funcional: RF01. Usuário deve ser capaz de definir a posição dos elementos no ambiente virtual.

Fluxo Básico:

1. Ator informa ao sistema qual tipo de elemento deseja adicionar ao ambiente do simulador.
2. Ator informa ao sistema a posição em que o elemento será fixado.
3. Sistema exibe o elemento posicionado no mapa.

O caso de uso se encerra.

Fluxo Alternativo: No passo 2, caso o ator informe uma posição que já esteja ocupada por outro elemento, o fluxo não avança para passo 3 até que a posição informada esteja desocupada.

3.11.2.2 – Caso de Uso: UC-02 Configurar Atributos Iniciais

Descrição: Este caso de uso especifica a ação de configurar os atributos iniciais (como: nutriente necessário, capacidade da reserva, limiar de fome, metabolismo, velocidade) de cada espécie de agentes.

Ator Principal: Usuário.

Pré-condição: Simulação ainda não tenha sido iniciada.

Pós-condição: Atributos das espécies de agentes são configurados.

Requisito Funcional: RF02. Usuário deve ser capaz de configurar os atributos iniciais dos agentes

Fluxo Básico:

1. Ator informa ao sistema qual atributo deseja alterar.
2. Ator informa ao sistema o novo valor do atributo.
3. Sistema informa atributo alterado e o novo valor.

O caso de uso se encerra.

3.11.2.3 – Caso de Uso: UC-03 Iniciar Simulação

Descrição: Este caso de uso especifica a ação de iniciar a simulação.

Ator Principal: Usuário.

Pré-condição: Simulação ainda não tenha sido iniciada.

Pós-condição: Simulação é iniciada.

Requisito Funcional: RF03. Usuário deve poder iniciar a simulação.

Fluxo Básico:

1. Ator informa ao sistema que deseja iniciar a simulação.
2. Simulação é iniciada.

O caso de uso se encerra.

3.11.2.4 – Caso de Uso: UC-04 Finalizar Simulação

Descrição: Este caso de uso especifica a ação de finalizar a simulação.

Ator Principal: Usuário.

Pré-condição: Simulação tenha sido iniciada.

Pós-condição: Simulação é finalizada.

Requisito Funcional: RF03. Usuário deve poder finalizar a simulação.

Fluxo Básico:

1. Ator informa ao sistema que deseja finalizar a simulação.
2. Simulação é finalizada.

O caso de uso se encerra.

3.11.2.5 – Caso de Uso: UC-05 Solicitar Designação

Descrição: Este caso de uso especifica a interação entre o agente e a base quando ele precisa solicitar a posição de um setor a ser explorado.

Ator Principal: Agente.

Pré-condição: Agente tenha ido ao último setor designado ou ainda não tenha sido designado.

Pós-condição: Agente ganha uma designação (toma ciência do próximo setor a ser explorado por ele).

Requisito Funcional: RF05. Agente deve poder solicitar uma designação

Fluxo Básico:

1. Agente aproximasse da base estratégica de sua espécie.
2. Agente solicita uma nova designação.
3. A base informa ao agente todas as fontes conhecidas até o momento.
4. A base estratégica informa o novo setor a ser visitado.

O caso de uso se encerra.

Fluxo Alternativo: No passo 2, caso o agente já tenha visitado o último setor designado, ele informa à base o estado do setor e se encontrou alguma fonte no caminho.

3.11.2.6 – Caso de Uso: UC-06 Explorar Setor Designado

Descrição: Este caso de uso especifica a ação do agente ao explorar um setor.

Ator Principal: Agente.

Pré-condição: Agente precisa ter uma designação.

Pós-condição: Agente cumpre a sua tarefa de designação

Requisito Funcional: RF06. Agente deve poder explorar o setor designado

Fluxo Básico:

1. Agente anda até o setor designado.
2. Agente chega ao setor.

O caso de uso se encerra.

3.11.2.7 – Caso de Uso: UC-07 Alimentar-se

Descrição: Esse caso de uso descreve o comportamento do agente quando o mesmo sente fome.

Ator Principal: Agente.

Pré-condição: Agente está com fome e conhece pelo menos uma fonte de nutrientes.

Pós-condição: Agente está com sua reserva de nutrientes cheia.

Requisito Funcional: RF07. Agente deve poder alimentar-se de uma fonte de nutrientes.

Fluxo Básico:

1. Agente mede a distância até cada fonte conhecida por ele.
2. Agente anda até a fonte mais próxima.
3. Agente chega até a fonte.
4. Agente alimenta-se até que sua reserva de nutrientes esteja completamente cheia.

O caso de uso se encerra.

3.11.2.8 – Caso de Uso: UC-08 Designar Agente

Descrição: Esse caso de uso descreve como a base define o setor a ser explorado pelo agente.

Ator Principal: Base Estratégica.

Pré-condição: Agente solicita nova designação.

Pós-condição: Agente recebe designação.

Requisito Funcional: RF08. Base deve ser capaz de designar o agente a um setor.

Fluxo Básico:

1. A base verifica qual é o perímetro de exploração corrente.
2. A base verifica qual foi o último setor do perímetro a ser designado.
3. A base calcula o próximo setor a ser designado.
4. A base informa a designação ao agente

O caso de uso se encerra.

3.11.3 – Diagrama de Casos de Uso

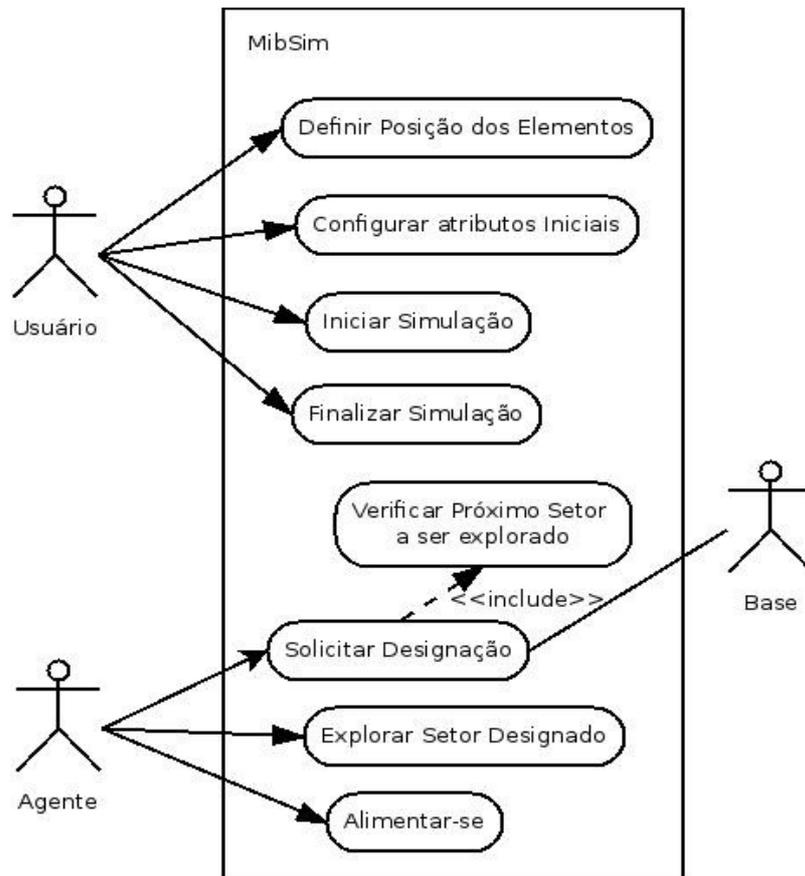


Figura 13: Diagrama de Casos de Uso

3.12 – Modelo conceitual

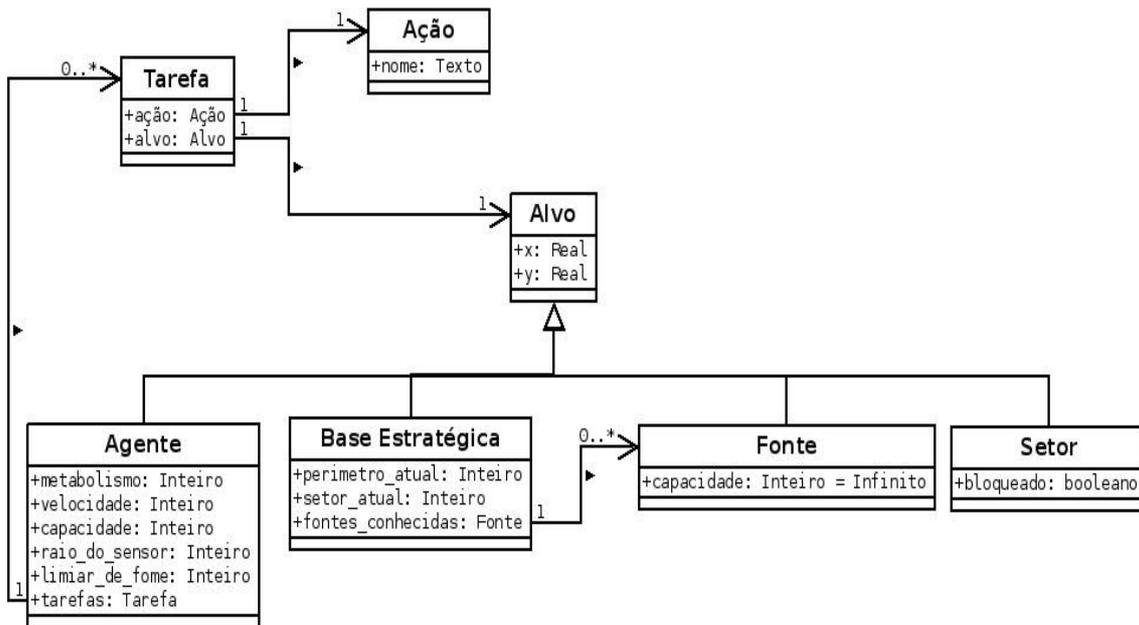


Figura 14: Diagrama UML do MibSim

A figura 14 mostra o modelo conceitual do MibSim feito em UML. As principais classes do sistema são: Agente, Base Estratégica, Fonte e Setor. Todas elas são também alvos, dessa maneira, com o algoritmo de movimentação orientado por alvo é possível fazer o agente caminhar até qualquer objeto de um desses tipos.

Cada agente possui uma lista de tarefas, que por sua vez é composta por uma ação e um alvo. O alvo da tarefa pode ser: um agente, uma base estratégica, a fonte ou um setor. As ações possíveis são: Explorar, Reportar, Alimentar-se.

A ação de explorar está ligada a um setor então quando a tarefa é explorar o alvo é o setor a ser explorado.

A ação de reportar é refere-se a reportar um setor à base estratégica que nesse caso é o alvo.

Alimentar-se é a ação relativa às fontes de alimento, quando a ação é alimentar-se o alvo é uma fonte de alimentos.

4 Funcionamento do Sistema

Nesta seção as partes que compõe o sistema implementado são descritas. Há detalhes desde as telas de configuração inicial até os algoritmos de comportamento e movimentação dos agentes. Por fim, é explicado como sistema lida com determinadas situações curiosas, como dois agentes sendo designados para o mesmo setor.

4.1 – Inicialização

Antes da simulação começar, o usuário do sistema precisa definir a posição das fontes (que permanece a mesma ao longo de toda a simulação), a posição dos agentes e as características das populações. Todos os agentes começam com as reservas de nutrientes cheias. Uma vez definidas as configurações iniciais, é possível começar as rodadas de execução da simulação.

	Ultralisk	Hydralisk	Lurker
Attributes			
Need:	◀ water ▶	◀ water ▶	◀ adamant. ▶
Speed:	- 5 +	- 8 +	- 6 +
Capacity:	- 10000 +	- 5000 +	- 8000 +
Metabolism:	- 3 +	- 2 +	- 5 +
Hunger Threshold:	- 3300 +	- 3000 +	- 3600 +

Figura 15: Configuração dos valores iniciais

A figura 15 mostra a tela de configuração inicial do sistema. Antes da simulação começar, o usuário pode configurar os valores dos atributos de cada espécie.



Figura 16: Posicionamento dos agentes

A figura 16 mostra a etapa de posicionamento das fontes e dos agentes, note que a espécie selecionada é mostrada abaixo da seta do mouse com transparência e o agente ainda não foi colocado na posição indicada. Somente quando o usuário clicar, o agente é inserido no ambiente.

Essa tela de posicionamento só aparece antes do início da simulação, logo não é possível inserir os agentes em tempo real mas da maneira que o MibSim foi proposto, cada novo agente é automaticamente incorporado às tarefas da base estratégica permitindo que seja possível adicionar agentes a qualquer momento.

4.2 – Rodada de Execução

A cada rodada de execução, os agentes tomam decisões individuais sobre qual ação tomar naquele momento (baseado em suas próprias avaliações) e para cada ação tomada, o indivíduo tem sua reserva de nutrientes afetada. A rodada de execução termina assim que todos os agentes tomam uma ação.

4.3 – Comportamento dos Agentes

O algoritmo que rege o comportamento de cada agente é baseado na hierarquia de necessidades de Maslow (vide seção 2.2.1).

As ações tomadas pelos agentes estão diretamente relacionadas aos níveis de sua reserva de nutrientes. Predominantemente o comportamento dos agentes tem caráter exploratório, tentando sempre explorar e reportar os setores designados pela base. Quando a *satisfação* do agente está abaixo do ***Limiar de Fome*** seu comportamento volta-se exclusivamente para a procura por comida, ignorando completamente a designação corrente.

Quando a reserva de nutrientes retorna a um nível acima do limiar de fome, o agente altera automaticamente sua prioridade, neste momento, este agente retoma o último objetivo, seja ele exploratório ou de reportagem.

Ao reportar à base, os indivíduos também têm seu nível da reserva de nutrientes afetado, caso sua reserva de alimentos atinja novamente um nível abaixo do limiar de fome, sua prioridade passa automaticamente a ser a busca por alimento.

A busca por alimentos se dá da seguinte maneira: o indivíduo mede a distância do ponto onde ele está até cada fonte de alimento conhecida, em seguida escolhe a mais próxima como alvo, caso duas ou mais fontes estejam à mesma distância, o agente escolhe a primeira fonte em ordem de perímetro e setor da esquerda para a direita, de cima para baixo. O indivíduo tenta chegar a seu alvo até que sua energia se esgote, caso consiga atingí-lo antes que sua reserva se esgote, ele alimenta-se da fonte até que seu reservatório atinja o nível máximo.

Algumas populações de agentes podem alimentar-se de mais de um tipo de fonte, quando isso ocorre, o reservatório é compartilhado entre essas fontes e não há diferenciação entre a origem do alimento de forma que qualquer fonte pode encher o reservatório da mesma maneira. Nestes casos, quando o agente precisa escolher de qual fonte vai se alimentar, considera as fontes de todos esses tipos. No *Mibsim* existem três tipos diferentes de fontes: fontes de água, fontes de açúcar e fontes de adamantita.

A percepção das fontes, por parte dos agentes se dá por proximidade, um agente percebe uma fonte quando passa por uma distância menor do que o alcance do seu sensor.



Figura 17: Raio de alcance do sensor

A figura 17 mostra a aparência do sistema em funcionamento, o terreno do ambiente dividido em tiles e um dos agentes com sua barra de nutrientes completamente cheia.

O círculo escuro ao redor do agente representa o alcance do sensor, o agente percebe fontes que estejam dentro deste raio. (vide seção 3.5.1).

Descrevendo o comportamento em termos de planejamento temos:

Estados: Alimentando-se, Andando até a base, andando até o setor designado e andando até a fonte mais próxima.

Estado Inicial: Andando até a base.

Ações: Andar até o setor, explorar setor, andar até a base, reportar à base, andar até uma fonte conhecida, alimentar-se.

Modelo Transicional:

A: Alimentando-se

B: Andando até a base

S: Andando até o setor designado

N: Andando até a fonte mais próxima

Os símbolos são as legendas das setas, como fome, recebe designação, chegou ao setor, reserva completa e reserva incompleta.

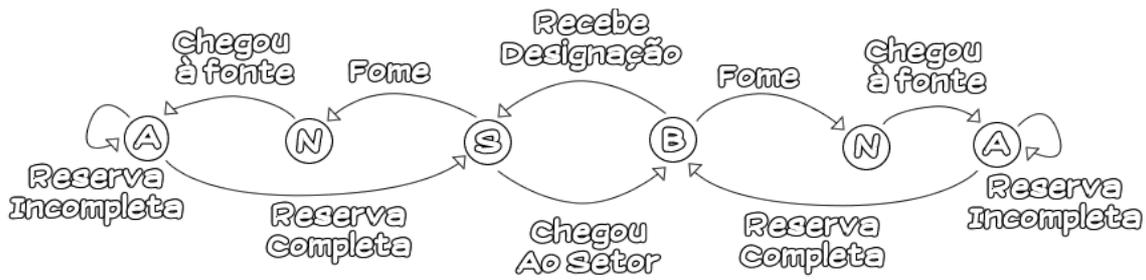


Figura 18: Modelo de Transição de Estados

A figura 18 temos o modelo transaccional dos estados dos agentes. Os agentes sempre começam com sua reserva de nutrientes cheia e assim que a simulação começa, eles partem em direção a base (estado **B** - Andando até a base). Se um agente sentir fome antes de chegar até a base, ele suspende imediatamente a ação de ir até a base e passa para o estado **N** (andando até a fonte mais próxima) e só retoma a ação de ir à base quando enche sua reserva de nutrientes. Assim que ele chega até a base, ele reporta a posição das fontes que tenham sido encontradas no caminho (as quais ele não tenha reportado anteriormente) e recebe uma designação.

Ao receber uma designação, seu estado passa para **S** (Andando até o setor designado), e novamente, se no meio do caminho sentir fome, o estado muda imediatamente para o estado **N**(andando até a fonte mais próxima), até que ele chegue a uma fonte e encha toda a sua reserva de nutrientes retornando para o estado anterior **S**.

Chegando ao setor, seu estado muda para **B** novamente e o agente começa a andar em direção à base.

4.4 – Barra de Nutrientes

A barra de nutrientes é a representação gráfica da reserva de nutrientes dos agentes no *MibSim*, cada espécie possui uma capacidade máxima fixa de armazenamento de nutrientes e um valor pré-definido como limiar de fome. Esses valores interferem diretamente na predominância do comportamento dos indivíduos daquela espécie (vide seção 3.6.3).



Figura 19: Barra de Nutrientes

Na figura 19 temos dois exemplos de barra de nutrientes, a mais acima representa a barra cheia, colorida de verde e preenchida totalmente. Já na barra de baixo, temos o nível de nutrientes abaixo do limiar de fome, nesse caso, a barra é pintada de vermelho para indicar que o agente está faminto.

4.5 – Tarefas de Planejamento

As tarefas de planejamento são formadas por uma ação e por um alvo. As ações que um agente pode realizar são: Explorar, Reportar, Alimentar-se e Medir a distância até um alvo. Os alvos podem ser as fontes ou a base da espécie.

Alvo ou ponto de ação é o lugar exato onde a ação deve ocorrer. Uma tarefa de planejamento sempre está associada a um alvo.

4.6 – Designação

Assim que a simulação começa, todos os agentes estão com suas respectivas reservas de nutrientes cheias então querem explorar, como ainda não possuem nenhuma designação todos os agentes dirigem-se à base da espécie. Ao primeiro contato com o primeiro agente, a fonte verifica se todos os setores do primeiro perímetro já foram explorados, a base é considerada o perímetro zero, e como a base é de conhecimento dos agentes e dela própria, um novo perímetro é definido com o perímetro corrente. Supondo que apenas um único agente chegou à base nesse primeiro instante, a base dá a ele a designação para explorar o **Setor 1-1**, e aguarda uma nova requisição. Caso um outro agente chegue à base logo após essa designação, a base o designa para o **Setor 1-2** e assim sucessivamente.

Supondo que 8 agentes receberam suas designações e ainda não voltaram para a base, se um nono agente chega à base, será designado ao **Setor 1-1** e ciclo se repete.

4.7 – Fim da simulação

O fim da simulação pode ser determinado pelo usuário, caso contrário, ocorre quando todos os agentes chegam ao fim do seu ciclo de vida, ou quando alguma espécie reconhece todos os setores do sistema.

4.8 – Possíveis Cenários

Nos subtópicos abaixo são descritos os possíveis cenários que podem ser entendidos como os estados em que os agentes podem encontrar-se.

4.8.1 – Planejamento de ida à Base

Esse é o estado inicial de cada agente, nesse cenário, seu alvo é a base estratégica e sua capacidade da reserva de nutrientes está acima do limiar de fome, e permanece assim até que o agente chegue ao alvo.



Figura 20: Hydralisk voltando para a base

Plano: Planejamento de ida à Base

Estado Inicial: Capacidade da Reserva de Nutrients acima do Limiar de Fome

Pré-Requisito(s): Ser a primeira ida à Base ou ter explorado o último setor designado

Objetivo: Chegar até a base

Descrição do Plano:

<p><i>Início</i></p> <ul style="list-style-type: none">- Marcar a base como alvo- Andar até o alvo- Se não for a primeira ida à base Reportar último setor- Receber informação sobre setores reportados por outros agentes <p><i>Fim</i></p>

4.8.2 – Planejamento de Exploração

A situação que ocorre imediatamente após o agente reportar o que foi encontrado no último setor explorado, nesse momento, o agente solicita à base estratégica uma próxima designação e inicia seu plano para chegar até o setor designado.

Plano: Planejamento de Exploração

Estado Inicial: Capacidade da Reserva de Nutrients acima do Limiar de Fome

Pré-Requisito(s): Ter uma designação

Objetivo: Chegar até o setor designado

Descrição do Plano:

<p><i>Início</i></p> <ul style="list-style-type: none">- Marcar o setor como alvo- Andar até o alvo- Verificar todas as partes do setor- Retornar para a base (planejamento anterior) <p><i>Fim</i></p>

4.8.3 – Planejamento para saciar a fome

No momento em que a capacidade corrente da reserva de nutrientes atinge ou fica abaixo do limiar de fome, o agente abandona (temporariamente) o plano que estava seguindo para ir até uma fonte e sanar sua fome. Depois de encher sua reserva, ele retoma o plano abandonado.

Plano: Planejamento para saciar a fome

Estado Inicial: Capacidade da Reserva de Nutrients abaixo do Limiar de Fome

Pré-Requisito: Agente está com fome

Objetivo: Alcançar uma fonte de nutrientes

Descrição do Plano:

<p><i>Início</i></p> <ul style="list-style-type: none">- Identificar a fonte conhecida mais próxima Medir a distância das fontes conhecidas Escolher a de menor distância (no caso em que as distâncias são iguais, o agente escolhe o primera fonte conhecida pela base). <ul style="list-style-type: none">- Marcar a fonte como alvo- Andar até o alvo- Consumir a fonte (alimentar-se) até que sua reserva fique completamente cheia <p><i>Fim</i></p>
--

4.8.4 – Dois agentes são designados para o mesmo ponto no mapa

Durante a execução do plano de ida à base, outros agentes podem solicitar designações à base, seguindo a ordem em que a base concede as designações (vide seção 4.7), se $8 \times P$ (sendo P o número do perímetro) agentes solicitarem por designações, o próximo agente será designado para um setor já designado a outro agente. Esse fato não afeta em nada o planejamento, no caso em que dois agentes são designados para o mesmo setor, eles vão realizar o mesmo trajeto (a menos que precisem buscar fontes de nutrientes). Ao reportar à base, ela irá ignorar o agente que reportar depois, essa redundância de designações pode ser útil em ambientes de muita mortalidade, se um dos agente morrer no caminho, o outro já estará cumprindo com o planejamento, economizando tempo.

4.8.5 – Situação em que o agente encontra uma borda do mapa

Quando um agente é designado para um setor mas no caminho, ele encontra um obstáculo, o agente não tenta desviar do obstáculo, simplesmente volta para a base e reporta o setor como inacessível. Essa situação é inevitável quando os perímetros de exploração ficam muito grandes. Este recurso também é utilizado para não causar efeitos colaterais indesejáveis no sistema por conta do algoritmo de movimentação não contornar obstáculos.

5 Conclusão

O trabalho apresentado mostra que com um pouco de conhecimento de Inteligência Artificial é possível construir um simulador que pode ser aplicado à situações do mundo real.

Apesar de simples, o algoritmo de comportamento dos agentes no *MibSim* aborda situações úteis em aplicações reais como impedir que os agentes não ultrapassem as bordas do mapa e evitar que a reserva de nutrientes/energia dos agentes esgote-se.

No caso do Roomba ou outro aspirador de pó autônomo, a limpeza tornaria-se mais eficiente, pois além de mapear o ambiente previamente, o esforço para aspirar o ambiente poderia ser dividido pelo número de robôs que houvessem no recinto. Outro fator importante é que novos robôs poderiam ser integrados ao grupo a qualquer momento sem causar nenhum efeito colateral indesejado, bastando que soubessem a posição da base.

Uma limitação do *MibSim* é não permitir a distribuição do conhecimento sobre a posição das fontes de alimento, havendo sempre a necessidade de interagir com a base estratégica para fornecer e receber informações sobre o posicionamento das fontes. Esse modelo é conveniente para o exemplo de robôs aspiradores de pó que já possuem uma base fixa conectada à rede elétrica. Voltando ao caso da exploração de Marte, onde a energia vem de forma constante e do Sol, não haveria necessidade de retornar a base para receber novas designações, desde que o conhecimento provindo dos setores já visitados fosse distribuído entre os agentes.

Trabalhos Futuros

Desviar de Obstáculos: Adicionar a funcionalidade de movimentar os agentes sem que eles ignorassem os obstáculos, isso permitiria o uso do sistema em uma gama muito mais ampla de situações.

Levar a ideia para o mundo real: Apesar dos exemplos anteriores fazerem referência a robôs autônomos, o MibSim não precisa necessariamente de robôs autônomos para funcionar, com algumas adaptações, seria possível utilizar o simulador para otimizar a logística de empresas de transportes terrestres, por exemplo. O sistema designaria os veículos para as filiais mais próximas, e diferente dos GPS, os trechos poderiam receber pesos dependendo do estado da estrada ou risco de saques. Os postos de combustível poderiam ser mapeados (manualmente ou com uso de sistemas externos como: GPSs ou Guias Automotivos).

Seguindo a lógica do sistema, quando o tanque do veículo estivesse baixo, o veículo mediria a distância das fontes (postos de combustível) mais próximos considerando o peso (dificuldade do trajeto) e iria para o melhor posto.

Esse comportamento evitaria abastecimentos tardios e com isso riscos de pane seca e possíveis saques além de otimizar a logística de entregas. Como apontado anteriormente, o sistema poderia ser utilizado em empresas de transporte ou frete que precisem garantir que os veículos não parem por falta de combustível.

Referências Bibliográficas

- [1] Maslow, A. Harold, A theory of human motivation, 1943.
- [2] Maslow, A. Harold, Motivation and personality. New York, NY: Harper., 1954.
- [3] Sharple, Mike et. al., Computers and Thought: A practical Introduction to Artificial Intelligence, 1989.
- [4] Russel, S. J.; Norvig, P., Artificial intelligence: a modern approach (3ª Edição). Prentice Hall, New Jersey, 2009.
- [5] CORREA, M. A. Filho. Arquitetura de diálogos entre agentes cognitivos distribuídos. Rio de Janeiro: COPPE da UFRJ, Tese de Doutorado, 1994.
- [6] Forlizzi, Jodi, and Carl DiSalvo. "Service robots in the domestic environment: a study of the roomba vacuum in the home". ACM, 2006.
- [7] Ghallab, Malik; Nau, Dana S.; Traverso, Paolo, Automated Planning: Theory and Practice, Morgan Kaufmann, 2004.
- [8] Poole, David; Mackworth, Alan; Goebel, Randy. Computational Intelligence: A Logical Approach. New York: Oxford University Press, 1998.
- [9] Layton, Julia. "How Robotic Vacuums Work." Disponível em: <<http://electronics.howstuffworks.com/gadgets/home/robotic-vacuum.htm>>. Acesso em: 10 de Novembro de 2014.
- [10] Durkin, John. "Expert Systems: Design and Development." *New York: Prentic Hall* 502, 1994.
- [11] FIPA. "Foundation for Intelligent Physical Agents". Abstract Architecture. Disponível em: <<http://www.fipa.org/specs/fipa00001/index.html>> Acesso em: 10 de Novembro de 2014.
- [12] Wason, P. C.; Shapiro, D. "Reasoning". In Foss, B. M. *New horizons in psychology*. Harmondsworth: Penguin, 1966.
- [13] Dreyfus, Hubert; Dreyfus, Stuart. *Mind over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*. Oxford, UK: Blackwell, 1986.
- [14] Gladwell, Malcolm. Blink. New York: Little, Brown and Co. ISBN 0-316-17232-4, 2005.

- [15] Tecuci, Gheorghe. "Artificial Intelligence". Wiley Interdisciplinary Reviews: Computational Statistics (Wiley) 4 (2): 168–180. doi:10.1002/wics.200, 2012.
- [16] Solomonoff, Ray. "An Inductive Inference Machine". Dartmouth Summer Research Conference on Artificial Intelligence, 1956.
- [17] Kearns, Michael J.; Vazirani V. Umesh. "An Introduction to Computational Learning Theory". The MIT Press, 2004.
- [18] Weng, J.; McClelland; Pentland, A.; Sporns, O.; Stockman, I.; Sur, M.; Thelen, E. "Autonomous mental development by robots and animals". *Science* 291: 599–600, 2001.
- [19] Nilsson, Nils. Artificial Intelligence: A New Synthesis. Morgan Kaufmann Publishers, 1998.
- [20] Luger, George; Stubblefield, William. "Artificial Intelligence: Structures and Strategies for Complex Problem Solving (5ª ed.)". The Benjamin/Cummings Publishing Company, Inc. ISBN 0-8053-4780-1, 2004.
- [21] Cycorp, Home of smarter solutions. Disponível em: <<http://www.cyc.com/>>. Acesso em: 18 de Novembro de 2014
- [22] ConceptNet5. Disponível em: <<http://conceptnet5.media.mit.edu/>>. Acesso em 18 de Novembro de 2014.
- [23] Dicionário Michaelis versão online. Disponível em: <michaelis.uol.com.br/moderno/portugues/>. Acesso em: 15 de Novembro de 2014.
- [24] Avengers Volume 1, #201-202. Marvel Comics. Novembro – Dezembro 1980.
- [25] Men In Black, Wiki. Disponível em: <<http://meninblack.wikia.com/wiki/Category:Species>>. Acesso em: 26 de Outubro de 2013.
- [26] Underwood, Peter, Bill Roper, Chris Metzen and Jeffrey Vaughn. *StarCraft (Manual)*. Irvine, Calif.: Blizzard Entertainment, 1998.
- [27] Ultralisk. Starcraft Wiki. Disponível em: <<http://starcraft.wikia.com/wiki/Ultralisk>>. Acesso em: 19 de Novembro de 2014.
- [28] Hidralisk. Starcraft Wiki. Disponível em: <<http://starcraft.wikia.com/wiki/Hydralisk>>. Acesso em: 19 de Novembro de 2014.
- [29] Lurker. Starcraft Wiki. Disponível em: <<http://starcraft.wikia.com/wiki/Lurker>>. Acesso em: 19 de Novembro de 2014.

[30] Mars Exploration Rover Mission: Overview. Disponível em:

<<http://mars.nasa.gov/mer/overview/>>. Acesso em: 21 de Novembro de 2014.

[31] White, Matthew. "Real-time optimally adapting meshes: terrain visualization in games.". *International Journal of Computer Games Technology* 2008 (2008): 12.