

**UNIVERSIDADE FEDERAL DE ALFENAS
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

Thales da Silva Coutinho

**ESTUDO E IMPLEMENTAÇÃO DE INTELIGÊNCIA
ARTIFICIAL PARA FUTEBOL DE ROBÔS**

Alfenas, 07 de dezembro de 2011.

UNIVERSIDADE FEDERAL DE ALFENAS
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**ESTUDO E IMPLEMENTAÇÃO DE INTELIGÊNCIA
ARTIFICIAL PARA FUTEBOL DE ROBÔS**

Thales da Silva Coutinho

Monografia apresentada ao Curso de Bacharelado em
Ciência da Computação da Universidade Federal de
Alfenas como requisito parcial para obtenção do Título de
Bacharel em Ciência da Computação.

[Orientador: Prof. Luiz Eduardo da Silva]

Alfenas, 07 de dezembro de 2011.

Thales da Silva Coutinho

**ESTUDO E IMPLEMENTAÇÃO DE INTELIGÊNCIA
ARTIFICIAL PARA FUTEBOL DE ROBÔS**

A Banca examinadora abaixo-assinada aprova a monografia apresentada como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Alfenas.

Prof. Flávio Barbieri Gonzaga

Universidade Federal de Alfenas

Prof. Humberto César Brandão

Universidade Federal de Alfenas

Prof. Luiz Eduardo da Silva (Orientador)

Universidade Federal de Alfenas

Alfenas, 07 de dezembro de 2011.

Dedico este trabalho a minha avó, por sempre ter incentivado meus estudos e acreditado em minha capacidade.

AGRADECIMENTO

Aos meus pais, Antônio **Flávio Cardoso Coutinho** e **Benedita Raimunda da Silva Coutinho**, por sua dedicação e preocupação com os meus estudos. A minha avó, **Neide Cardoso Faria** pelo apoio irrestrito em minha formação acadêmica.

Aos meus professores do curso de Bacharelado em Ciência da Computação (BCC) da Unifal-MG, cujos conselhos e lições me deram a direção nessa jornada tao importante em minha vida.

Ao professor **Luiz Eduardo da Silva** pela sua orientação neste trabalho e confiança em minha capacidade de concluí-lo.

Ao professor **Paulo Bressan** pelo conselho que apenas um verdadeiro mestre poderia oferecer.

Aos professores **Flávio Barbieri Gonzaga** e **Eliseu Miguel** pela orientação no projeto de Iniciação Científica que tive oportunidade de participar que em muito contribuiu para minha formação.

Ao professor **Eliseu César Miguel** pelos conselhos no momento certo que me fizeram recuperar minha ambição e confiança em meu trabalho.

Aos amigos Tarick, Lyniker, Guilherme e Luiz Henrique por terem estado ao meu lado em momentos difíceis que enfrentei longe de minha família.

A todos os grandes amigos que tive o prazer de conhecer ao longo destes anos e que me presentearam com companhia agradável e momentos inesquecíveis.

Thales da Silva Coutinho

“Podemos praticar atos nobres sem ter que dominar a terra e o mar.”

Albert Einstein

RESUMO

O Futebol de Robôs se apresenta a comunidade científica com o intuito de ser um problema padrão para a pesquisa nas áreas de Inteligência Artificial e Robótica. O problema consiste em realizar uma partida de futebol utilizando robôs como jogadores, este cenário envolve diversas áreas de conhecimento relacionadas.

A idéia por trás da adoção de um problema padrão é que isto possibilita que diversas teorias e técnicas sejam utilizadas, testadas e, por estarem sendo aplicadas a um mesmo cenário, facilmente comparadas. A pouco tempo atrás o Xadrez computacional era visto como um problema padrão pela comunidade científica, porém muitos acreditam que este problema já esteja resolvido. O Futebol de Robôs traz um cenário dinâmico de tempo real, muito mais complexo em relação ao Xadrez computacional, além de propor a utilização de inteligências em conjunto com o agente que deve utilizá-las: um robô.

Uma das iniciativas existentes para incentivar a pesquisa e o desenvolvimento deste problema é o *RoboCup Soccer Simulator 2d* que realiza uma simulação de um jogo de futebol entre robôs, excluindo a robótica e permitindo que apenas a área da Inteligência Artificial seja estudada e desenvolvida. Neste trabalho propomos a criação de uma inteligência artificial para este simulador utilizando uma árvore de decisões.

Palavras-Chave: Inteligência Artificial, Futebol de Robôs, RoboCup, Agentes Autônomos, Árvore de Decisões.

ABSTRACT

Robot Soccer presents itself to the scientific community as a standard problem to research in Artificial Intelligence and Robotics. The problem consists in performing a soccer match using robots as players, this scenario involves many different related areas of knowledge.

The main idea behind the adoption of a standard problem is the possibility to use, test and, with the same scenario, easily compare many different theories. Not too long ago, Computer Chess was seen by the scientific community as a standard problem in Artificial Intelligence, but now many scientists believe that this problem is solved. Robot Soccer brings a dynamic real time environment, much more complex compared to Computer Chess, and it proposes the use of intelligence alongside the agents that should use it: robots.

One of the existing initiatives to incentive research and development in Robot Soccer is the RoboCup Soccer Simulator which simulates a robot soccer match, this way the robotics it's excluded, allowing the Artificial Intelligence to be fully developed and studied. On the present work we propose the development of an Artificial Intelligence for this simulator using a Decisions Tree. |

Keywords: | Artificial Intelligence, Robot Soccer, RoboCup, Autonomous Agents, Decision Tree. |

LISTA DE FIGURAS

FIGURA 1- UMA ÁRVORE DE DECISÃO PARA ESCOLHER ENTRE ESPERAR OU NÃO POR UMA MESA.	36
FIGURA 2 - TRÊS AGENTES DO MESMO TIME (<i>KEEPERS</i>) TENTANDO MANTER A BOLA(<i>BALL</i>) FORA DO ALCANCE DE DOIS AGENTES ADVERSÁRIOS(<i>TAKERS</i>) DENTRO DOS LIMITES (<i>BOUNDARY</i>) DE UM DETERMINADO SETOR (STONE <i>ET AL.</i> , 2005)	38
FIGURA 3 - ARQUITETURA DO SIMULADOR.....	40
FIGURA 4 - ARQUITETURA DO CLIENTE.....	42
FIGURA 5 - ÁRVORE DE DECISÕES PARA AGENTE EM POSSE DA BOLA	46
FIGURA 6 - ÁRVORE DE DECISÕES PARA AGENTE SEM A POSSE DA BOLA	47
FIGURA 7 - <i>FLAGS</i> EXISTENTES NO AMBIENTE DE SIMULAÇÃO (ROBOCUP SOCCER SIMULATOR 2D - USERS MANUAL)	49
FIGURA 8 - DISPOSIÇÃO DE <i>FLAGS</i> EM UM SETOR DO CAMPO UTILIZADO POR UM JOGADOR	50
FIGURA 9 - JOGADOR SEM NENHUMA <i>FLAG</i> EM SEU CAMPO DE VISÃO	51
FIGURA 10 - JOGADOR COM APENAS UMA <i>FLAG</i> EM SEU CAMPO DE VISÃO	51
FIGURA 11 - JOGADOR COM DUAS <i>FLAGS</i> INTERIORES EM SEU CAMPO DE VISÃO.....	52
FIGURA 12 - JOGADOR COM UMA <i>FLAG</i> INTERIOR E UMA <i>FLAG</i> EXTERIOR EM SEU CAMPO DE VISÃO	53
FIGURA 13 - SETORES DE CAMPO NO AMBIENTE DE SIMULAÇÃO	54
FIGURA 14 - SETOR DE DEFESA E CAMPO DE VISÃO DO JOGADOR 2 NO LADO ESQUERDO DO CAMP.....	56
FIGURA 15 - SIMULAÇÃO EM ANDAMENTO	58
FIGURA 16 - SIMULAÇÃO FINALIZADA	59

LISTA DE TABELAS

TABELA 1 – COMPARAÇÃO ENTRE XADREZ COMPUTACIONAL E FUTEBOL DE ROBÔS (KITANO <i>ET AL.</i> , 1997)	30
TABELA 2 - SETORES DE DEFESA PARA JOGADORES DO LADO ESQUERDO DO CAMPO.....	54
TABELA 3 - SETORES DE ATAQUE PARA JOGADORES DO LADO ESQUERDO DO CAMPO	55
TABELA 4 - CLIENTE PARA SERVIDOR - CONECTAR, RECONECTAR, E DESCONECTAR.....	72
TABELA 5 - DE SERVIDOR PARA CLIENTE - INFORMAÇÕES SENSORIAIS	73
TABELA 6 - DE CLIENTE PARA SERVIDOR - AÇÕES DE JOGADOR	74

LISTA DE ABREVIACÕES

FIRA	<i>Federation of International Robot-Soccer Association</i>
RoboCup	<i>Robot World Cup</i>
FIFA	<i>Fédération Internationale de Football Association</i>

SUMÁRIO

1 INTRODUÇÃO	25
1.1 JUSTIFICATIVA E MOTIVAÇÃO	26
1.2 PROBLEMATIZAÇÃO.....	27
1.3 OBJETIVOS	28
1.3.1 Gerais	28
1.3.2 Específicos	28
2 REVISÃO BIBLIOGRÁFICA	29
2.1 AGENTES AUTÔNOMOS.....	29
2.2 FUTEBOL DE ROBÔS	30
2.3 ROBOCUP	31
2.4 ROBOCUP SOCCER SIMULATOR.....	32
2.5 INTELIGÊNCIA ARTIFICIAL	33
2.5.1 O Que é IA?	33
2.5.2 Fundamentos de IA	33
2.6 AGENTES INTELIGENTES.....	33
2.7 APRENDIZADO.....	34
2.7.1 Árvore de Decisões.....	34
2.8 TOMADA DE DECISÕES	36
2.8.1 Aprendizado por Reforço.....	37
2.9 TRABALHOS RELACIONADOS.....	38
3 INTELIGÊNCIA ARTIFICIAL PARA FUTEBOL DE ROBÔS EM AMBIENTE SIMULADO	39
3.1 FUNCIONAMENTO DO SIMULADOR.....	39
3.2 FUNCIONAMENTO DO CLIENTE.....	41
3.3 TOMADA DE DECISÕES	43
3.3.1 Regras de Decisão.....	43
3.3.2 Árvore de Decisões.....	46
3.3.3 Posicionamento em Campo	48
4 VALIDAÇÃO DO MÓDULO CLIENTE PARA O AMBIENTE DE SIMULAÇÃO	57
4.1 EXECUÇÃO DE UMA PARTIDA DE FUTEBOL.....	57
4.2 VERIFICAÇÃO DA OBEDIÊNCIA DAS AÇÕES	59
5 CONCLUSÕES E TRABALHOS FUTUROS	61
5.1 CONCLUSÕES	61
5.2 TRABALHOS FUTUROS	62
6 REFERÊNCIAS BIBLIOGRÁFICAS	65
7 APÊNDICES E ANEXOS	67
7.1 APÊNDICE A	67
1.1 APÊNDICE B.....	71

1

Introdução

Este capítulo apresenta as razões pelas quais este trabalho foi desenvolvido, bem como o que este buscou resolver e a proposta utilizada nesta solução. Na Seção 1.1 são apresentadas as justificativas para o desenvolvimento deste trabalho. A Seção 1.2 apresenta o problema central que este trabalho procura resolver. Na Seção 1.3 são apresentados os objetivos que se pretende atingir com este trabalho.

Atualmente pesquisas e desenvolvimentos ligados as áreas de Inteligência Artificial e Robótica têm sido amplamente discutidos não só no âmbito acadêmico. O Instituto Tecnológico de Aeronáutica (ITA) tem pesquisado o uso de agentes autônomos para Futebol de Robôs em ambiente simulado (Xavier *et al.*, 2006) bem como desenvolvido agentes robóticos para ambientes reais de Futebol de Robôs (Griip & Matsuura, 2007).

O Futebol de Robôs é uma das muitas iniciativas utilizadas para incentivar trabalhos ligados a Inteligência Artificial e Robótica. Através da adoção deste problema padrão pode-se avaliar e acompanhar, de maneira clara e objetiva, o progresso das pesquisas na área (Xavier *et al.*, 2006).

O problema abordado pelo Futebol de Robôs consiste em integrar a percepção, as ações e o comportamento de múltiplos agentes robóticos durante uma partida de futebol, fazendo-os buscar um mesmo objetivo comum: a vitória da partida.

Hoje em dia, existem diversas organizações de torneios de Futebol de Robôs, iniciativas voltadas a promover e incentivar as pesquisas na área. Como por exemplo a FIRA (*Federation of International Robot-soccer Association*)¹, organização que promove torneios de diversas categorias, variando os cenários e os agentes

xxvxxv

¹ www.fira.net

robóticos utilizados, em determinadas categorias os robôs são humanoides, em outras são cubos, etc.

Outra entidade voltada a promover o Futebol de Robôs é a RoboCup², a organização promove torneios utilizando agentes robóticos humanoides, dentre seus objetivos a RoboCup propõe que até o ano de 2050 um time de robôs será capaz de vencer a então seleção campeã da Copa do Mundo da FIFA (*Fédération Internationale de Football Association*) em um jogo de futebol definido pelas regras da associação (Kitano *et al.*, 1997). Em outra categoria a RoboCup promove torneios em um ambiente virtual, onde os robôs são simulados e apenas seu comportamento, ações e percepções definem os acontecimento da partida, ou seja, este cenário exclui a construção eletrônica e mecânica do agente robótico, permitindo um foco maior em seus componentes de software: Percepção e Inteligência Artificial.

Desta maneira este trabalho tem como proposta um estudo e desenvolvimento de uma Inteligência Artificial para o ambiente de simulação *RoboCup Soccer Simulator 2d*, utilizando estratégias conhecidas do futebol. |

1.1 Justificativa e Motivação

| O problema da tomada de decisões em agentes robóticos independentes é que não existe comunicação explícita entre os agentes, ou seja, os agentes não trocam informações relevantes como posição por exemplo. Logo, um agente deve extrair tais informações de maneira autônoma e tomar decisões baseado não somente em suas próprias variáveis, mas também levando em consideração informações sobre os outros agentes trabalhando em seu ambiente.

A importância da abordagem deste problema é que em muitos casos a comunicação direta entre agentes robóticos trabalhando em conjunto não é possível ou não é viável.

xxvixxvi

² www.robocup.org

No contexto do Futebol de Robôs os agentes robóticos tomam a forma de jogadores de futebol, estes não possuem uma comunicação direta entre si, mas cada jogador precisa de informações sobre os outros jogadores para tomar uma decisão. Para cada jogador é importante saber a posição, direção, velocidade e outras informações sobre os outros jogadores para que uma decisão seja tomada.

Uma vez que o jogador possa adquirir tais informações, é essencial que a decisão tomada por ele seja coerente com o estado dos outros jogadores. Esta decisão pode ser tomada por um mecanismo inteligente que avalie as muitas opções possíveis e escolha a melhor possível considerando o estado de todos os jogadores, da bola e qualquer outra informação relevante à decisão. |

1.2 Problematização

[O Futebol de Robôs subdivide-se em três principais problemas: Montagem e manutenção do agente robótico, percepção do ambiente de jogo e seus componentes e tomada de decisões.

A tomada de decisões cabe ao componente de software e é o foco deste trabalho. Cada agente robótico deve tomar decisões, de maneira independente, quanto ao melhor movimento a ser executado em determinado momento, entre as ações possíveis do agente estão: mover-se, driblar, chutar, passar e bloquear. Em vista deste problema questiona-se: como decidir entre as muitas decisões possíveis qual a melhor ação a ser tomada por um agente robótico autônomo no contexto de simulação do *Robocup Soccer Simulator 2d*? |

1.3 Objetivos

1.3.1 Gerais

O objetivo geral deste trabalho foi estudar e Implementar uma Inteligência Artificial para o ambiente de simulação *RoboCup Soccer Simulator 2d*. Além disso, definir e implementar um módulo de jogo utilizando tal inteligência para o ambiente de simulação em questão. |

1.3.2 Específicos

Levando em conta o objetivo geral definido anteriormente, os objetivos específicos deste trabalho foram:

- Estudar o funcionamento e as características do ambiente de simulação
- Estudar técnicas e algoritmos de Inteligência Artificial.
- Estudar técnicas de futebol que possam ser aplicadas no contexto do simulador.
- Definir o módulo de jogo para o ambiente de simulação.
- Implementar o módulo de jogo para o ambiente de simulação.
- Implementar a Inteligência Artificial sobre o módulo de jogo criado.
- Elaborar a monografia. |

2

Revisão Bibliográfica

Este capítulo apresenta o estado da arte dos conceitos necessários para um melhor entendimento desta monografia. Inicialmente são definidos alguns conceitos para melhor compreensão do problema abordado. Posteriormente apresenta-se os conceitos de Inteligência Artificial aplicados neste trabalho.

2.1 Agentes Autônomos

O conceito de agentes autônomos é relativamente recente, tendo surgindo através dos estudos em Inteligência Artificial. Wooldridge *et al.* (1995) destaca a dificuldade em se definir o termo “agente” para a computação baseada em agentes, da mesma maneira que é difícil definir o termo “inteligência” na Inteligência Artificial. Isso ocorre principalmente pelo fato do termo ser utilizado em muitas áreas relacionadas, sendo difícil encontrar uma definição universal.

Apesar de não existir uma definição concreta para o termo “agente”, existe um consenso entre autores de que um agente deve ser autônomo, ou seja, deve ser capaz de executar sua função sem nenhuma interferência humana externa (Russel & Norvig, 2003).

Brooks (1990) defende a ideia de que agentes autônomos podem exibir comportamento inteligente, sem necessariamente utilizar representações simbólicas explícitas de conhecimento, sugerindo o uso de padrões de comportamento inteligente.

Atualmente a robótica se encontra em constante evolução e seus robôs estão sendo utilizados para realizar diversas tarefas. E em muitos casos, o uso de múltiplos agentes robóticos é necessário para realizar uma tarefa de forma cooperativa. O Futebol de Robôs apresenta um destes casos, onde onze agentes devem cooperar para atingir um único objetivo, além de possuírem uma situação adversa: oponentes tentando impedi-los.

2.2 Futebol de Robôs

Futebol de Robôs é uma iniciativa internacional voltada para a pesquisa e educação, visando promover desenvolvimentos ligados as áreas de Inteligência Artificial e Robótica Inteligente (Kitano *et al.*, 1997). A proposta é que através da adoção de um problema padrão seja possível avaliar diversas teorias, algoritmos e arquiteturas.

Em um passado recente o Xadrez computacional foi um problema padrão amplamente abordado, e as pesquisas levaram a uma grande evolução na área de Inteligência Artificial. Porém alguns pesquisadores já consideram que o objetivo final do problema Xadrez computacional já tenha sido alcançado. Em 1996 ocorreu uma série de seis jogos entre um supercomputador da IBM chamado *Deep Blue Team* e o mestre de xadrez Gary Kasparov, da Rússia. O russo venceu três partidas, perdeu uma e empatou duas. Um ano depois, em uma revanche chamada de “o mais espetacular evento de xadrez na história”, Garry Kasparov foi derrotado pelo *Deep Blue Team*, perdendo três partidas, vencendo duas e empatando uma. Por isso a adoção de um novo problema padrão surge como opção para incentivar novas pesquisas em áreas tecnológicas das indústrias do futuro (Kitano *et al.*, 1997). Kitano considera que o Futebol de Robôs possa suprir essa necessidade e mostra como o domínio do problema abrange de maneira mais completa os novos desafios.

Tabela 1 - Comparação entre Xadrez Computacional e Futebol de Robôs (Kitano *et al.*, 1997)

	Xadrez	Futebol de Robôs
Ambiente	<i>Estático</i>	<i>Dinâmico</i>
Mudança de Estado	<i>Em turnos</i>	<i>Tempo real</i>
Acesso a Informações	<i>Completo</i>	<i>Incompleto</i>
Leitura de Sensor	<i>Simbólico</i>	<i>Não Simbólico</i>
Controle	<i>Central</i>	<i>Distribuído</i>

A liga de Futebol de Robôs RoboCup é projetada para tratar problemas e complexidades do mundo real e cobrir as áreas de Inteligência Artificial e Robótica.

Assim como o grande objetivo de uma Inteligência Artificial para Xadrez era ser capaz de derrotar grandes mestres do jogo, a instituição possui um objetivo similar para o Futebol de Robôs: em meados de 2050, um time de jogadores de futebol robóticos humanoides seja capaz de vencer um jogo de futebol, de acordo com as regras da FIFA, contra a então seleção campeã do mundo.

2.3 Robocup

O professor Alan Mackworth, da Universidade de Colúmbia, foi o primeiro a mencionar a ideia de robôs jogando futebol em um artigo titulado “*On Seeing Robots*” em 1992. No mesmo ano, de maneira independente, um grupo de pesquisadores japoneses organizou o “Workshop de Grandes Desafios em Inteligência Artificial”, onde discussões sobre possíveis grandes desafios na área de Inteligência Artificial levaram a ideia de utilizar o futebol para promover a ciência e a tecnologia. Após alguns estudos sobre os impactos sociais, econômicos e tecnológicos que os trabalhos nesta área trariam, um grupo de pesquisadores concluiu que o projeto era viável e desejável. Então este grupo resolveu promover um torneio de Futebol de Robôs, o projeto foi nomeado *Robot World Cup*, abreviado como “RoboCup”.

Desde então a instituição promove torneios em diversas categorias visando incentivar o desenvolvimento da área da Robótica e da Inteligência Artificial. Existem categorias que utilizam robôs pequenos e com poucos movimentos, a principal categoria utiliza robôs humanoides que tentam imitar os movimentos humanos, e há também uma liga simulada, onde as partidas acontecem em ambiente de simulação, este cenário exclui o agente robótico como parte do problema, o que possibilita que toda a atenção seja dirigida ao desenvolvimento da Inteligência Artificial que controla os agentes durante a simulação.

2.4 Robocup Soccer Simulator

Com o intuito de reduzir o tempo de desenvolvimento de sistemas de controle robótico, existem diversos sistemas disponíveis para simular o ambiente em que o robô será utilizado. Para o Futebol de Robôs alguns simuladores permitem uma precisa configuração de robôs através de seu ambiente de simulação, o ÜberSim é um desses simuladores. Segundo Browning (Browning & Tryzelaar, 2003), a variabilidade no projeto de robôs significa que deve ser fácil adicionar e reconfigurar robôs em um ambiente simulado.

A Robocup possui um simulador para Futebol de Robôs que exclui a construção de agentes robóticos da equação, este simulador é utilizado nas competições da liga simulada da instituição. Esta liga simulada ocorre através de um ambiente de simulação distribuído pela própria organização: *o RoboCup Soccer Simulator 2d*. O simulador possui um funcionamento simples, atuando como um servidor contendo todas as informações do ambiente de jogo, aos desenvolvedores cabe a tarefa de criar um *software* cliente que se comunique com este servidor, enviando mensagens contendo as ações dos jogadores e recebendo informações sobre o ambiente para utilizar em suas decisões.

Cada jogador de cada time deve abrir uma conexão com o servidor, desta maneira pode existir um único cliente para criar uma conexão para cada um dos jogadores de seu time, ou diversos clientes, cada um representando um jogador. Em ambos os casos cada jogador possui uma conexão independente com o servidor, e troca suas próprias mensagens com o mesmo, isto torna cada jogador um agente autônomo, que deve tomar decisões de maneira independente, porém cooperando com outros agentes autônomos para alcançar o mesmo objetivo: a vitória.

2.5 Inteligência Artificial

2.5.1 O Que é IA?

Ao longo do tempo a Inteligência Artificial surgiu de maneira quase natural dentro da ciência, e à medida que a área evolui sua definição muda de autor para autor. Segundo Bellman (Bellman, 1978) Inteligência Artificial é a “Automatização de atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado...”. Posteriormente a Inteligência Artificial foi definida como um esforço para fazer com que computadores pensem no sentido literal (Haugeland, 1985). Definições mais atuais tentam mostrar o lado mais aplicável da Inteligência Artificial, fugindo da ficção. Segundo Poole *et al.*(1998) “A Inteligência Computacional é o estudo do projeto de agentes inteligentes”.

Todas essas definições tentam mostrar o objetivo mais simples da Inteligência Artificial: solucionar problemas que os computadores atuais não podem solucionar de forma trivial, problemas que são naturalmente “humanos”.

2.5.2 Fundamentos de IA

Buscar uma forma de fazer com que máquinas solucionem problemas através de um raciocínio inteligente é uma tarefa extremamente complexa, e ao longo dos anos diversas áreas contribuíram para formar as ideias que hoje são utilizadas na IA. Segundo Russel e Norvig (2003) algumas disciplinas se destacam, como a matemática com algoritmos e o Teorema da Incompletude, a Neurociência com estudos do funcionamento do cérebro, e outras.

2.6 Agentes Inteligentes

Um agente é tudo que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre esse ambiente por intermédio de atuadores

(Russel & Norvig, 2003). O ser humano pode ser considerado um agente inteligente, olhos, ouvidos, narinas e outros órgãos atuam como sensores, percebendo o mundo ao seu redor, já suas mãos, pernas, boca e outros membros fazem o papel de atuadores.

Em agentes robóticos os sensores e atuadores tomam outras formas, como câmeras e braços mecânicos que permitem que o agente interaja com o mundo ao seu redor.

No contexto do *Robocup Soccer Simulator 2d* os agentes tomam a forma de jogadores de futebol, estes agentes percebem o ambiente ao seu redor através de sensores de visão e audição que são vistos na forma de mensagens enviadas pelo servidor ao cliente do jogador. O corpo do jogador é o atuador, correndo, chutando e bloqueando adversários.

2.7 Aprendizado

Para poder atuar sobre o mundo em que existe um agente inteligente deve poder analisar as informações de seus sensores para tomar uma decisão sobre o que fazer em seguida, esta tomada de decisões pode ser feita de diversas maneiras.

2.7.1 Árvore de Decisões

Uma árvore de decisão possui como entrada uma situação na forma de um conjunto de atributos e, como saída, uma decisão. Os atributos de entrada podem ser variáveis discretas ou contínuas.

O processo alcança uma decisão através de uma série de testes. Cada nó da árvore de decisões representa um teste, cada valor possível deste teste gera uma ramificação na árvore e as folhas representam uma possível decisão.

Um clássico exemplo da árvore de decisões é o problema de decidir por esperar ou não por uma mesa em um restaurante, tomamos a decisão “VaiEsperar”

como predicado de objetivo (Russel & Norvig, 2003). Neste problema possuímos a seguinte lista de atributos:

- Alternativa: há um restaurante alternativo por perto
- Bar: o restaurante possui um bar confortável para espera
- Sex/Sab: o dia da semana é sexta ou sábado
- Faminto: estamos com fome
- Clientes: número de pessoas no restaurante (Nenhum, Alguns ou Cheio)
- Preço: A faixa de preço do restaurante (\$, \$\$ ou \$\$\$)
- Chovendo: se está chovendo
- Reserva: possuímos uma reserva de mesa no restaurante
- Espera estimada: tempo de espera estimado (<10 minutos, de 10 a 30 minutos, de 30 a 60 minutos, >60 minutos)

Observe que os atributos “Clientes”, “Preço” e “Espera” foram normalizados para que tomassem a forma de variável discreta, tornando o exemplo exemplo mais simples.

Russel e Norvig (2003) definem a árvore para o problema da seguinte maneira:

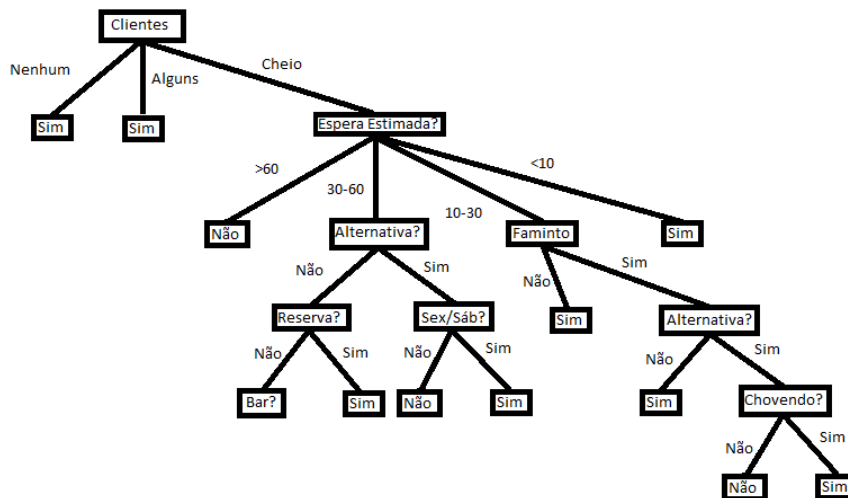


Figura 1- Uma árvore de decisão para escolher entre esperar ou não por uma mesa.

Onde as folhas “sim” definem que iremos esperar por uma mesa e as folhas “não” definem que não iremos esperar por uma mesa.

As configurações do mundo em questão definem o valor das variáveis utilizadas na decisão pela árvore, que por sua vez define a ação a ser tomada pelo agente inteligente. Em um caso em que o restaurante está cheio, com uma espera estimada de 10 a 30 minutos por uma mesa, o agente está faminto, e não há uma outra alternativa para se obter uma refeição, a decisão tomada seria esperar por uma mesa.

2.8 Tomada de Decisões

O princípio da aprendizagem em Inteligência Artificial é que as percepções do agente inteligente não sejam utilizadas apenas para tomar decisões e agir, mas

também para melhorar a habilidade do agente em futuras ações (Russel & Norvig, 2003).

2.8.1 Aprendizado por Reforço

Aprendizado por Reforço (Sutton & Barto, 1998) é um método de aprendizado de máquina projetado para permitir que um agente autônomo maximize sua eficiência em longo prazo através de experimentação repetida e interação com seu ambiente. Já tem sido demonstrado empiricamente que este método pode garantir a convergência a uma política de controle ótima (Russel & Norvig, 2003).

Em ambientes muito dinâmicos, como no Futebol de Robôs, os agentes possuem apenas uma visão parcial do mundo, além de possuírem ruídos, ou seja, eles não podem perceber o mundo exatamente como ele é, e não podem alterá-lo exatamente como planejado. A comunicação entre os agentes pode ser limitada, e os mesmos devem tomar ações em tempo real, o tradicional paradigma de Inteligência Artificial utilizando entrada para ativar ações, como no Xadrez Computacional, pode não ser muito eficiente neste cenário.

O aprendizado por reforço é um bom método para realizar decisões sequenciais. Tudo isso torna este método bem preparado para enfrentar os desafios do RoboCup, porém alguns problemas surgem com o uso desta técnica. Apesar de seus recursos, a aprendizagem por reforço pode ser muito difícil de calibrar em grandes domínios em razão do crescimento exponencial de estados em relação ao número de variáveis de estado, fenômeno conhecido como a “maldição da dimensionalidade” (Stone *et al.*, 2005).

É perfeitamente imaginável que cada agente deva aprender de forma autônoma, compartilhando um mínimo de informações durante a fase de aprendizado, já que durante sua atividade suas ações serão realizadas de forma autônoma, e cada agente cooperativo terá um papel único no mundo em que estão, atacante, goleiro, zagueiro, etc. cada agente representa um jogador em uma determinada posição, e seu aprendizado deve condizer com sua função.

2.9 Trabalhos relacionados

Ao longo dos últimos anos muitos trabalhos tem sido desenvolvido para o simulador *Robocup Soccer Simulator 2d*, alguns descrevem apenas funções específicas de um agente no ambiente de simulação, outros descrevem arquiteturas e algoritmos para times inteiros.

Peter Stone (Stone *et al.*, 2005) descreve como agentes inteligentes devem agir para manter a bola em sua posse ou de seus companheiros, mantendo-a fora do alcance de adversários.

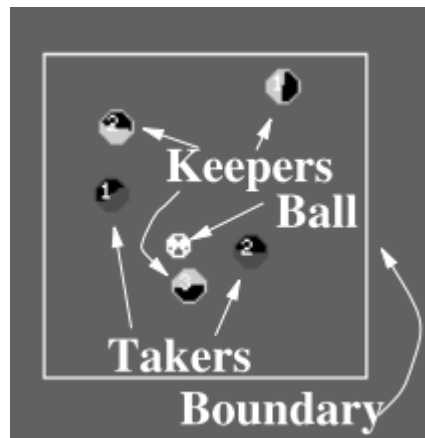


Figura 2 - Três agentes do mesmo time (*keepers*) tentando manter a bola(*ball*) fora do alcance de dois agentes adversários(*takers*) dentro dos limites (*boundary*) de um determinado setor (Stone *et al.*, 2005)

Stone ainda define um método de Aprendizado por Reforço para a tarefa de manter a posse de bola.

Raphael O. Xavier (Xavier *et al.*, 2006) desenvolveu um time completo baseado no *UvA Trilearn*, um time que utiliza uma arquitetura de *threads* de três camadas com um avançado método de sincronização, um modelo probabilístico, e uma hierarquia de habilidades para que um agente tome uma decisão (Kok *et al.*, 2003). O *UvA Trilearn* venceu diversas competições oficiais e paralelas desde 2001, já o trabalho de Raphael O. Xavier venceu uma competição nacional não oficial.

3

Inteligência Artificial para Futebol de Robôs em Ambiente Simulado

Este capítulo apresenta os conceitos e métodos empregados por este trabalho para o desenvolvimento de uma Inteligência Artificial para Futebol de Robôs em ambiente simulado. A Seção 3.1 apresenta a arquitetura do ambiente de simulação utilizado. Na Seção 3.2 a arquitetura do módulo Cliente desenvolvido para o Simulador é apresentada. Por fim a Seção 3.3 apresenta as regras de tomada de decisão utilizadas.

O ambiente escolhido para o desenvolvimento deste trabalho foi o *Robocup Soccer Simulator 2d* descrito na Seção 2.4. Antes de mostrar como o simulador foi utilizado e o comportamento dos agentes em seu ambiente implementado, é necessário entender melhor o funcionamento do simulador e do cliente: módulo que representa os agentes em ação sobre o ambiente de simulação.

3.1 Funcionamento do Simulador

O simulador *Robocup Soccer Simulator 2d* é dividido em módulos cooperativos, sendo o módulo principal da simulação o servidor. O servidor atua como o núcleo da simulação, guardando todas as informações necessárias sobre o mundo e os agentes atuando sobre o mesmo.

Através de troca de mensagens o servidor envia e recebe informações de outros dois módulos: o monitor e o cliente. O monitor é utilizado para visualizar o andamento da simulação, bem como para passagem de mensagens alheias aos

agentes para o simulador como: iniciar a partida, advertir um jogador com cartão, paralisar a partida quando ocorrerem situações não tratadas pelo servidor, etc.

Um ou mais clientes representam os agentes em atuação sobre o cenário simulado, o cliente, representando um determinado jogador, envia ao servidor uma mensagem referente a uma ação que deseja executar, o servidor por sua vez trata a mensagem recebida e realiza as alterações correspondentes. A cada ciclo o servidor envia ao monitor mensagens sobre o estado atual da simulação e aos clientes mensagens referentes à visão que estes possuem do mundo em questão, bem como outras informações relevantes sobre a simulação.

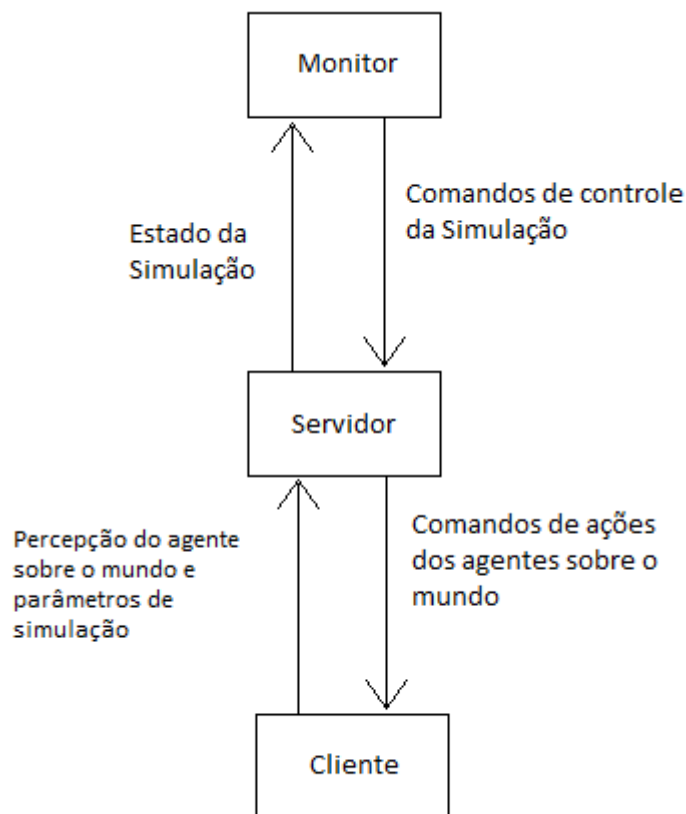


Figura 3 - Arquitetura do Simulador

3.2 Funcionamento do Cliente

Sendo o cliente responsável pela representação dos agentes em ação sobre a simulação ele deve ser capaz de perceber o mundo ao seu redor e tomar ações baseado nas informações disponíveis. No presente trabalho o Cliente desenvolvido foi dividido em diversos módulos, sendo os principais: *Team*, *Player* e *Body*.

- *Team*: é responsável por inicializar todos os jogadores de um time da simulação, após essa inicialização este módulo não realiza mais nenhuma ação.
- *Player*: representa um agente em ação, é ele o responsável por tomar as decisões do jogador dentro do ambiente de simulação.
- *Body*: é o corpo do jogador, neste módulo a comunicação com o Servidor é feita e as informações sensoriais e os parâmetros da simulação são guardados. *Body* e *Player* são executados em *Threads* diferentes.

Após a inicialização dos jogadores, o módulo *Body* passa a receber informações do servidor a cada ciclo de simulação.

As ações do jogador são decididas pelo módulo *Player*, baseadas nas informações contidas em *Body*, e passadas ao *Body* para que este envie a mensagem correspondente ao Servidor a cada 100ms.

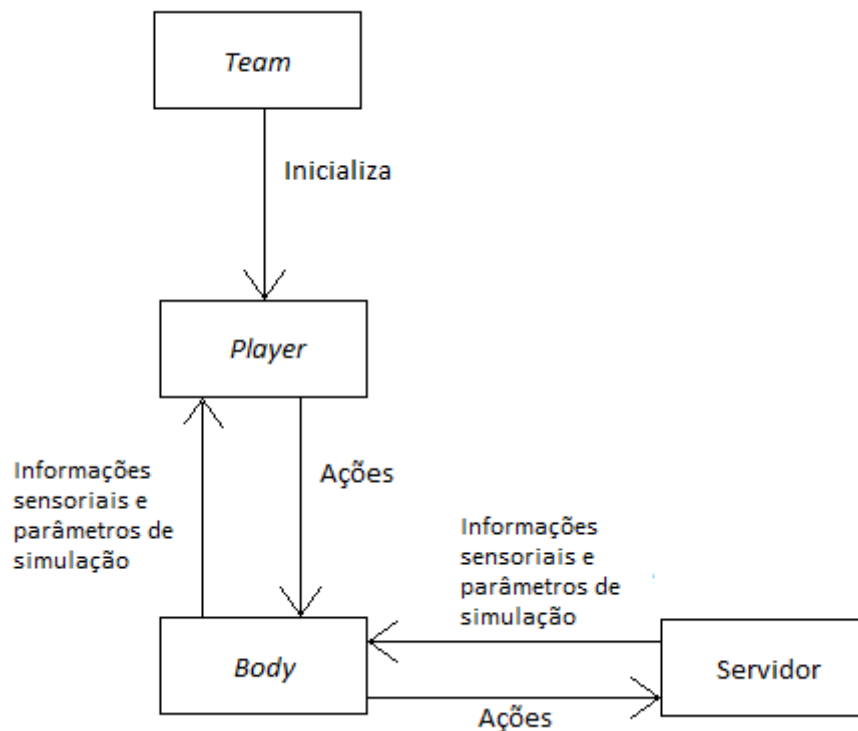


Figura 4 - Arquitetura do Cliente

As mensagens recebidas pela classe *Body* possuem um formato específico do simulador, e devem ser tratadas pelo mesmo para que as informações contidas na mensagem sejam extraídas e armazenadas. As mensagens enviadas pelo servidor são:

- (*server_param* (parâmetro valor) (parâmetro valor) ... (parâmetro valor)): esta mensagem contém os parâmetros utilizados na simulação, como *kickable_margin* que define a distância mínima entre o jogador e a bola para que um chute possa ser executado.
- (*see* tempo (nomeDoObjeto distancia direção mudançaDeDistancia mudançaDeDireção direçãoCorpo direçãoCabeça)): nesta mensagem o agente recebe informações sensoriais sobre seu campo de visão e os objetos.
- (*hear* tempo origem "Mensagem"): esta mensagem possui um propósito geral de comunicação na simulação, o campo "origem" define quem enviou a mensagem e é utilizada principalmente para mensagens de alteração de

modo de jogo, nesse caso a mensagem é enviada pelo *referee*, através do Monitor.

- (*sense_body* ((atributo valor) (atributo valor) ... (atributo valor)): aqui o agente recebe informações sobre seus próprios atributos como: força, velocidade, etc.

Todas essas mensagens são tratadas e armazenadas pelo módulo *Body* e então suas informações podem ser utilizadas pelo jogador para que o mesmo tome suas decisões.

3.3 Tomada de Decisões

3.3.1 Regras de Decisão

Existe um conjunto restrito de ações possíveis para o agente no contexto do simulador, neste trabalho foi definido um conjunto de condições que levam o agente a executar determinada ação. As ações do agente dependem da posse da bola em determinado instante, se a bola estiver em posse do jogador, suas ações são:

- Chutar ao gol
 - Caminho para o gol está livre, não há jogadores bloqueando a trajetória da bola
 - O jogador está a uma distância mínima do gol
 - Não há companheiro melhor posicionado
- Avançar com a bola
 - Existe um caminho livre de marcação para avançar em direção a linha de fundo
- Driblar
 - Há um ou mais adversários no caminho do jogador
 - Não há companheiro melhor posicionado e livre de marcação

- Tocar
 - Há um ou mais adversários no caminho do jogador
 - Há um companheiro melhor posicionado, livre de marcação

Caso a bola esteja em posse de um companheiro o jogador possui apenas uma ação:

- Posicionar-se para receber a bola e/ou atacar

Esta ação obviamente depende da função do jogador em campo e do posicionamento dos adversários.

Se a bola estiver em posse do adversário as ações possíveis são:

- Marcar outro adversário
 - Adversário com a bola não está em seu setor do campo
 - Existe outro adversário em seu setor se posicionando para atacar
- Desarmar
 - Adversário com a bola está em seu setor do campo

Por fim existe o caso em que a bola não está em posse de nenhum jogador em ambos os times, ou seja, está solta em algum lugar do campo, neste caso o jogador pode:

- Ir em direção a bola
 - Não há outro companheiro mais próximo a bola
- Posicionar-se para atacar
 - O jogador mais próximo a bola é um companheiro, este caso é tratado como se a bola estivesse em posse de um companheiro
- Posicionar-se para defender
 - O jogador mais próximo a bola é um adversário, este caso é tratado como se a bola estivesse em posse do adversário

Há ainda os casos de bola parada, nestes casos a determinação do jogador a cobrar é feita pela função do jogador em campo segundo a tabela seguinte.

Tabela 2 -

Ação	Jogador
Saída de bola do círculo central	<i>Atacante camisa número 9</i>
Cobrança de falta	<i>Meio campo camisa número 10</i>
Cobrança de escanteio pela esquerda	<i>Lateral esquerdo</i>
Cobrança de escanteio pela direita	<i>Lateral Direito</i>
Cobrança de pênalti	<i>Atacante camisa número 9</i>
Cobrança de tiro de meta	<i>Goleiro</i>
Cobrança de lateral pela esquerda	<i>Lateral esquerdo</i>
Cobrança de lateral pela direita	<i>Lateral direito</i>

A rigor o jogador mais próximo a bola é quem executa a cobrança, porém o jogador mais próximo a bola em geral será o jogador cujo setor de ataque ou defesa é o setor onde a cobrança deve ser feita, desta maneira um jogador não precisa comprometer a formação tática da equipe para cobrar uma bola parada.

Caso a cobrança seja feita por um companheiro, o jogador trata o caso em que a bola está em posse de um companheiro e se posiciona para atacar, caso contrário a bola está em posse do adversário e o jogador se posiciona para defender.

A partir das regras acima podemos definir uma Árvore de Decisões para um agente em Futebol de Robôs.

3.3.2 Árvore de Decisões

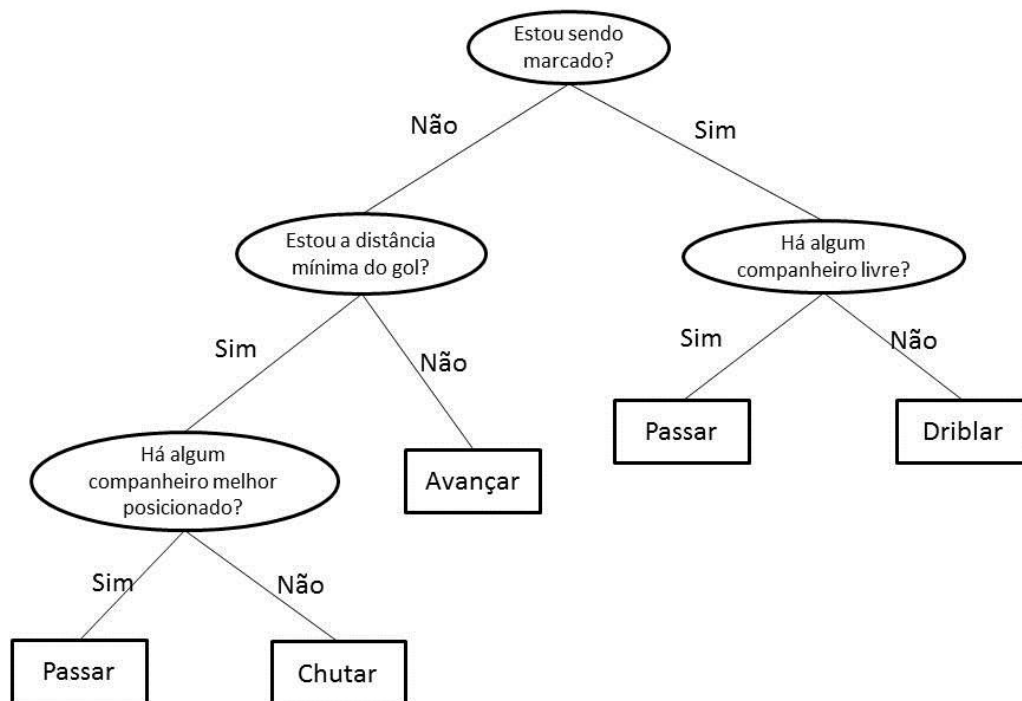


Figura 5 - Árvore de Decisões para agente em posse da bola

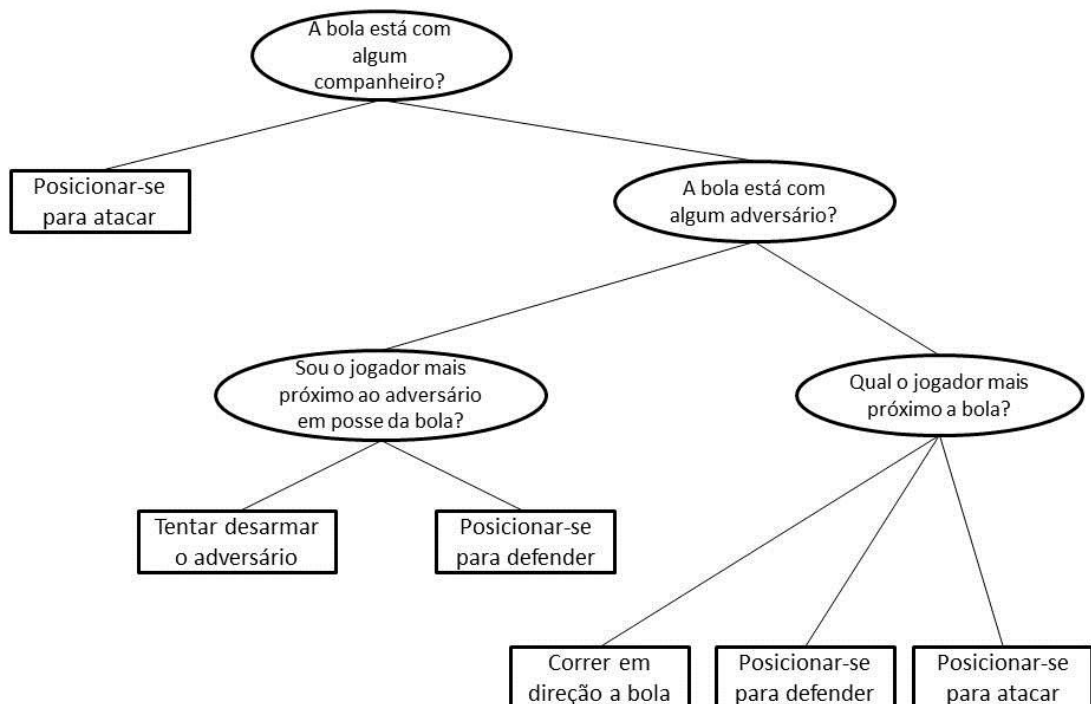


Figura 6 - Árvore de Decisões para agente sem a posse da bola

As decisões não triviais nos nós da Arvore de Decisões são decididas da seguinte maneira:

- Posse da bola: a bola esta em posse de um jogador se o mesmo se encontra a uma distância menor ou igual à distância mínima necessária para que o jogador possa executar um chute na bola.
- Jogador marcado: um jogador está sendo marcado caso este possua um adversário muito próximo. Se há um jogador adversário no campo de visão do jogador, então este está a frente do jogador, onde o jogador possui visão. O fato deste adversário estar a uma distância muito pequena caracteriza este adversário como um marcador.
- Companheiro melhor posicionado: um companheiro encontra-se melhor posicionado caso esteja a uma distância menor em relação ao gol adversário e não exista adversários o marcando.

Algumas ações executadas pelo agente não são triviais, a execução de tais ações é definida da seguinte maneira:

- Tentar desarmar o adversário: para esta ação o jogador deve se aproximar da bola em posse de um adversário o suficiente para chutá-la, e então chutar a bola para uma posição mais favorável, onde exista um companheiro próximo.
- Driblar: neste caso o jogador define a direção do adversário marcando e chuta a bola na diagonal deste adversário, podendo então correr novamente em direção a bola, ficando em uma posição mais favorável e, se possível, livre da marcação do adversário.
- Posicionar-se para atacar: esta ação é executada quando a bola esta em posse de um companheiro ou um companheiro é o jogador mais provável para alcançar a bola, o jogador neste caso deve se posicionar em um setor do campo determinado pela função definida para este jogador.
- Posicionar-se para defender: analogamente a ação de posicionar-se para atacar, esta ação é executada quando um adversário esta em posse da bola ou é o jogador mais provável para alcançá-la. O jogador deve buscar uma posição no campo seguindo uma definição de posição defensiva de acordo com sua função no campo.

3.3.3 Posicionamento em Campo

Para se posicionar em campo um agente deve mover-se através dele. Comportamentos de movimentação básicos consistem de duas ações executadas sequencialmente: rotacionar-se e correr (Kim *et al.*, 1997).

No ambiente de simulação utilizado neste trabalho os agentes não possuem coordenadas cartesianas triviais, e sim possuem apenas a sua percepção do mundo ao seu redor: a distância e direção dos objetos dentro de seu campo de visão. Para que o agente possa localizar-se o ambiente de simulação posiciona *flags* em

diversos pontos do campo, o jogador pode ver estas flags e a distância e direção das mesmas, e assim determinar sua localização.

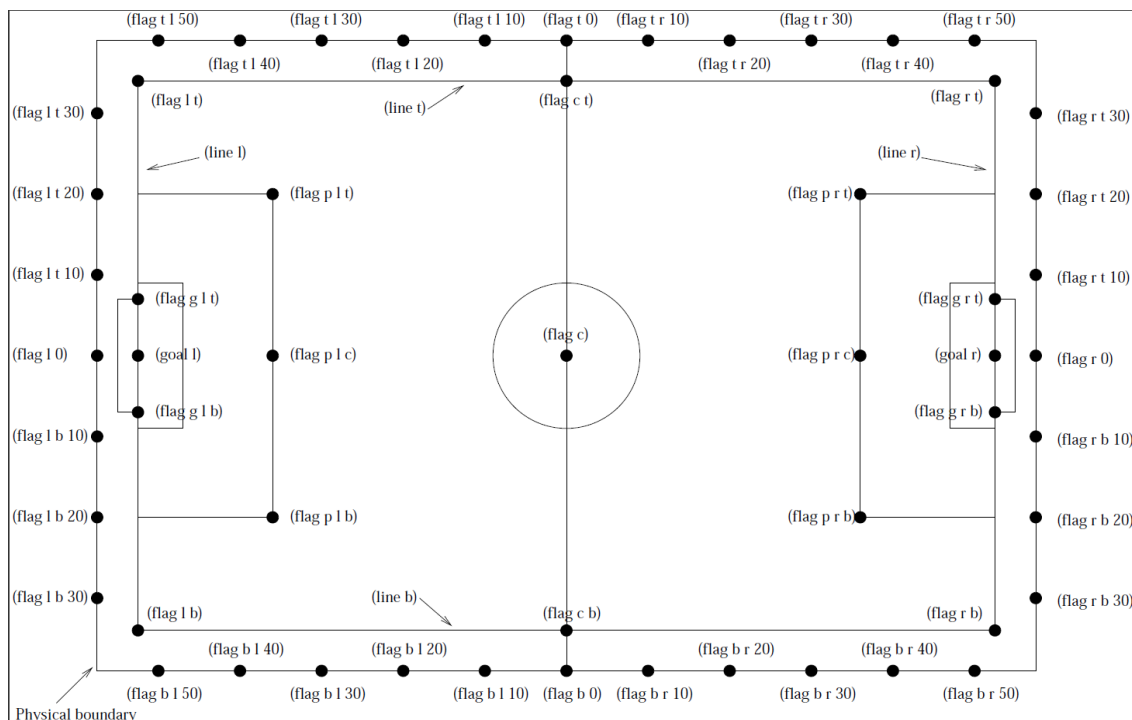


Figura 7 - Flags existentes no ambiente de simulação (Robocup Soccer Simulator 2d - Users Manual)

Neste trabalho foi definido um determinado setor do campo, baseado nas *flags* existentes no ambiente simulado, para cada jogador atacar e defender, cada um destes setores é definido por oito *flags*, a presença destas *flags* no campo de visão do agente define se o mesmo se encontra no setor em que deveria estar.

Para que isto seja possível oito *flags* são utilizadas: quatro internas e quatro externas.

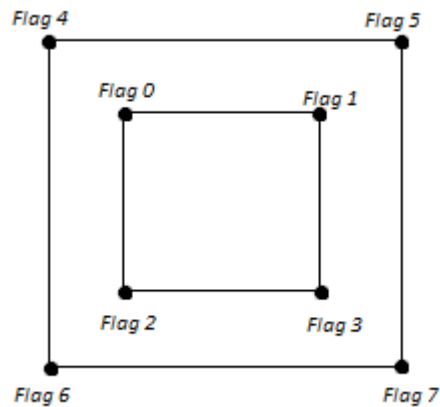


Figura 8 - Disposição de *flags* em um setor do campo utilizado por um jogador

Na figura acima vemos a disposição das *flags* que definem um setor de campo, chamaremos as *flags* 0, 1, 2 e 3 de interiores, e as *flags* 4, 5, 6 e 7 de exteriores. Para que um agente esteja corretamente posicionado dentro de um setor de campo ele deve estar entre as *flags* interiores, as *flags* exteriores são utilizadas para verificação quando o agente se encontra dentro do setor definido pelas *flags* interiores.

Para decidir se está ou não dentro do setor em que deveria estar o agente avalia quais e quantas *flags* estão em seu campo de visão. Se não houver nenhuma *flag* visível o agente vira seu corpo procurando por uma das *flags* que definem seu setor.

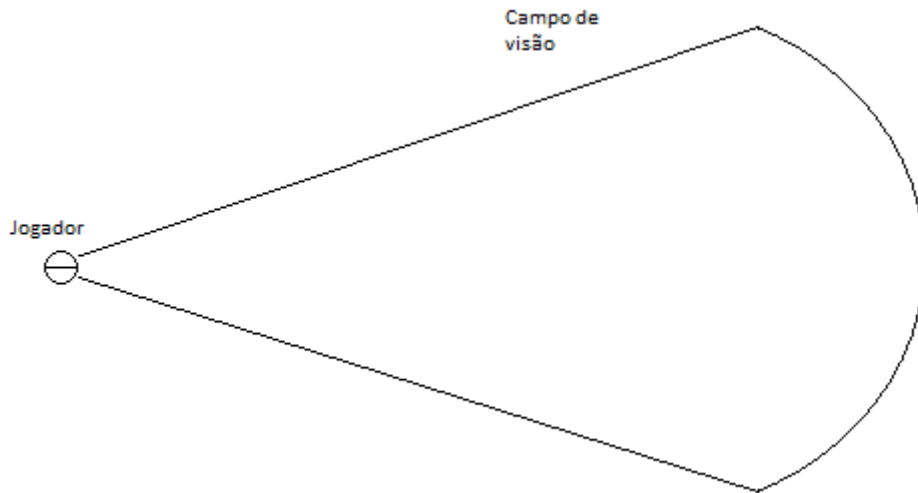


Figura 9 - Jogador sem nenhuma *flag* em seu campo de visão

Se apenas uma *flag* estiver visível o agente avalia sua distância, se essa distância for pequena o agente apenas vira o corpo procurando pelas outras *flags*, caso contrário a distancia é muito grande e é provável que o agente esteja fora de seu setor, então ele deve correr em direção a esta única *flag* visível.

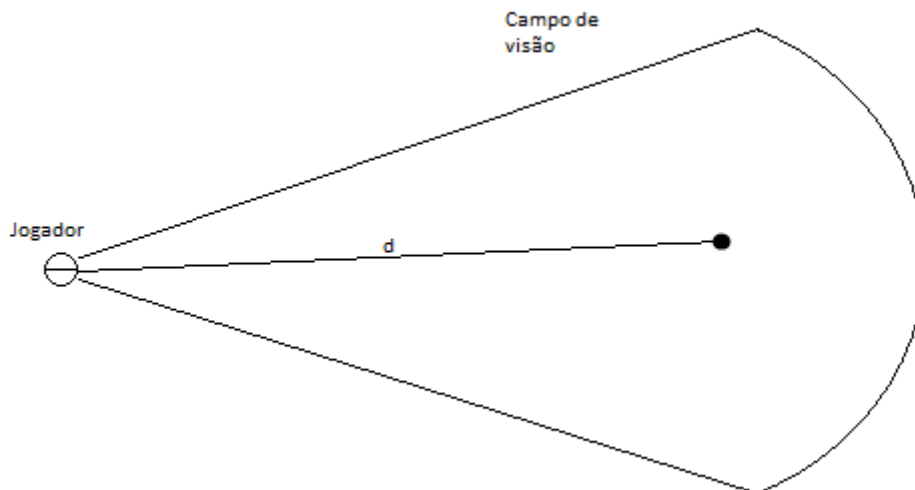


Figura 10 - Jogador com apenas uma *flag* em seu campo de visão

Quando duas *flags* estiverem visíveis o jogador deve definir se as *flags* são interiores ou exteriores, caso ambas sejam interiores o jogador esta fora de seu

setor, pois a posição do jogador entre as *flags* interiores não permite que este consiga visualizar duas *flags* interiores simultaneamente, então o agente deve correr em direção a *flag* visível mais distante.

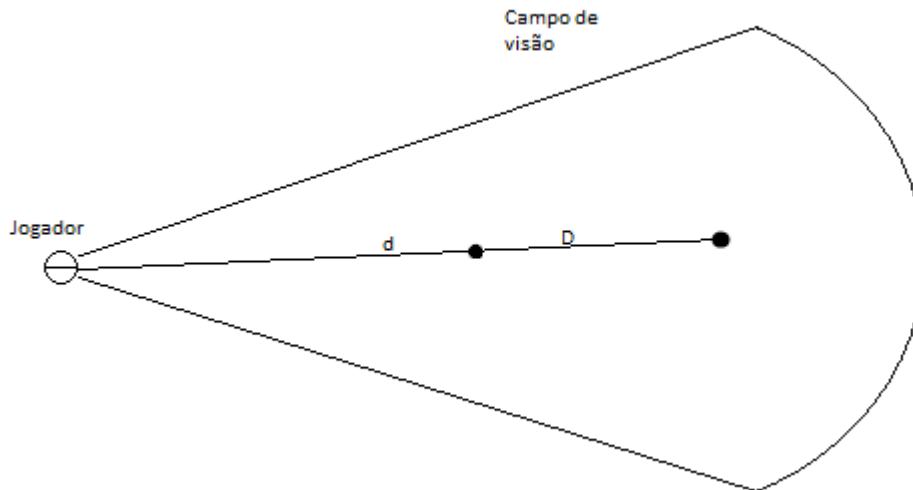


Figura 11 - Jogador com duas *flags* interiores em seu campo de visão

Caso uma das *flags* seja exterior o agente verifica qual das *flags* está mais próxima a ele, quando a *flag* mais próxima for interior e a distância para esta *flag* for pequena, o agente se encontra corretamente posicionado, mas se a *flag* mais próxima for a exterior então o jogador deve correr em direção a *flag* interior em seu campo de visão.

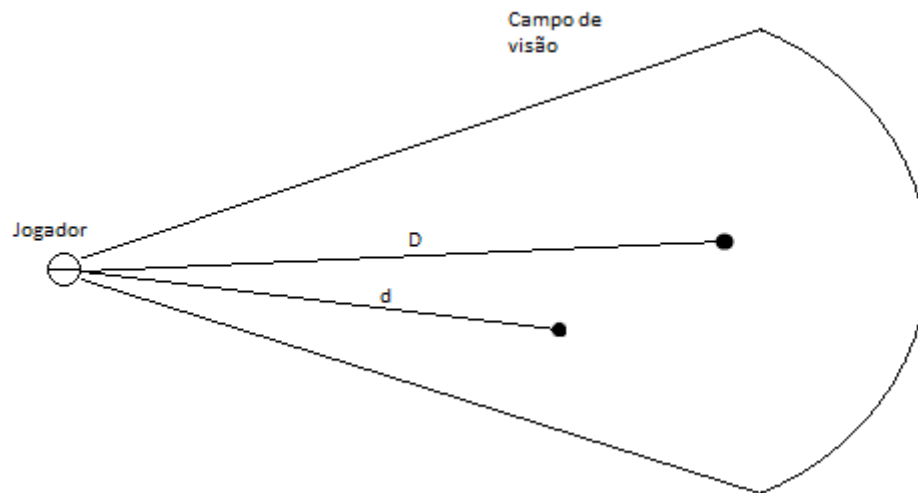


Figura 12 - Jogador com uma *flag* interior e uma *flag* exterior em seu campo de visão

Cada jogador possui uma função em campo, logo cada agente possui um setor definido para situações de ataque e outro setor definido para situações de defesa.

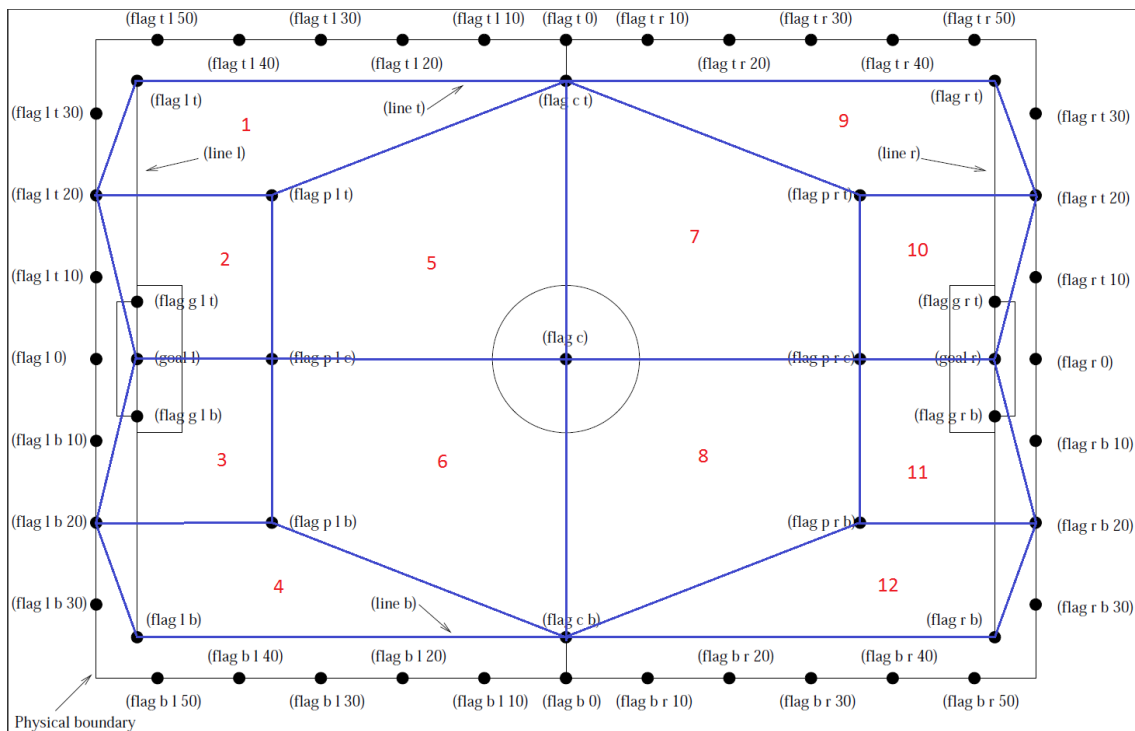


Figura 13 - Setores de campo no ambiente de simulação

Neste trabalho foram definidos os setores de ataque e de defesa para os jogadores numerados de 2 a 11. Para cada lado do campo os setores são definidos pelas tabelas abaixo.

Tabela 2 - Setores de Defesa para jogadores do lado esquerdo do campo

Jogador	Setor	Flags Interiores	Flags Exteriores
2	3	<i>gl; fplc; flb20; fplb</i>	<i>flt10; fct; flb30; fbl20</i>
3	2	<i>flt20; fplt; gl; fplc</i>	<i>flt30; ftl20; flb10</i>
4	4	<i>flb20; fplb; flb; fcb</i>	<i>flb10; fc; flb30; fbr10</i>
5	1	<i>flt; fct; flt20; fplt</i>	<i>flt30; ftr10; flt10; fc</i>
6	6	<i>fplc; fc; fplb; fcb</i>	<i>fglt; fprr; flb; fbr10</i>
7	8	<i>fc; fprc; fcb; fprb</i>	<i>fplt; fgrr; fbl10; frb</i>
8	5	<i>fplt; fct; fplc; fc</i>	<i>flt; ftr10; fglt; fprb</i>
9	7	<i>fprc; gr; fprb; frb10</i>	<i>fct; ftr10; fbr20; frb20</i>
10	6	<i>fplc; fc; fplb; fcb</i>	<i>fglt; fprr; flb; fbr10</i>
11	5	<i>fplt; fct; fplc; fc</i>	<i>flt; ftr10; fglt; fprb</i>

Tabela 3 - Setores de Ataque para jogadores do lado esquerdo do campo

Jogador	Setor	Flags Interiores	Flags Exteriores
2	6	<i>fplc; fc; fplb; fcb</i>	<i>fglt; fppt; flb; fbr10</i>
3	5	<i>fplt; fct; fplc; fc</i>	<i>flt; ftr10; fglb; fprb</i>
4	12	<i>fprb; frb20; fcb; frb</i>	<i>fc; frb10; fbl10; frb30</i>
5	9	<i>fct; frt; fppt; frt20</i>	<i>flt10; frt30; fc; frt10</i>
6	8	<i>fc; fprc; fcb; fprb</i>	<i>fplt; fgrr; fbl10; frb</i>
7	11	<i>fprc; gr; fprb; frb10</i>	<i>fct; frt10; fbr20; frb20</i>
8	7	<i>fct;fppt; fc; fprc</i>	<i>fil10; frt; fplb; fgrb</i>
9	10	<i>fppt; frt20; fprc; gr</i>	<i>ftr20; frt30; fcb; frb10</i>
10	8	<i>fc; fprc; fcb; fprb</i>	<i>fplt; fgrr; fbl10; frb</i>
11	7	<i>fct;fppt; fc; fprc</i>	<i>fil10; frt; fplb; fgrb</i>

Tabela 4 - Setores de Defesa para jogadores do lado direito do campo

Jogador	Setor	Flags Interiores	Flags Exteriores
2	7	<i>fct;fppt; fc; fprc</i>	<i>fil10; frt; fplb; fgrb</i>
3	10	<i>fppt; frt20; fprc; gr</i>	<i>ftr20; frt30; fcb; frb10</i>
4	12	<i>fprb; frb20; fcb; frb</i>	<i>fc; frb10; fbl10; frb30</i>
5	9	<i>fct; frt; fppt; frt20</i>	<i>flt10; frt30; fc; frt10</i>
6	8	<i>fc; fprc; fcb; fprb</i>	<i>fplt; fgrr; fbl10; frb</i>
7	6	<i>fplc; fc; fplb; fcb</i>	<i>fglt; fppt; flb; fbr10</i>
8	7	<i>fct;fppt; fc; fprc</i>	<i>fil10; frt; fplb; fgrb</i>
9	5	<i>fplt; fct; fplc; fc</i>	<i>flt; ftr10; fglb; fprb</i>
10	8	<i>fc; fprc; fcb; fprb</i>	<i>fplt; fgrr; fbl10; frb</i>
11	7	<i>fct;fppt; fc; fprc</i>	<i>fil10; frt; fplb; fgrb</i>

Tabela 5 - Setores de Ataque para jogadores do lado direito do campo

Jogador	Setor	Flags Interiores	Flags Exteriores
2	8	<i>fc; fprc; fcb; fprb</i>	<i>fplt; fgrr; fbl10; frb</i>
3	7	<i>fct;fppt; fc; fprc</i>	<i>fil10; frt; fplb; fgrb</i>
4	4	<i>flb20; fplb; flb; fcb</i>	<i>flb10; fc; flb30; fbr10</i>
5	1	<i>flt; fct; flt20; fplt</i>	<i>flt30; ftr10; flt10; fc</i>
6	6	<i>fplc; fc; fplb; fcb</i>	<i>fglt; fppt; flb; fbr10</i>
7	2	<i>flt20; fplt; gl; fplc</i>	<i>flt30; flt20; flb10</i>
8	5	<i>fplt; fct; fplc; fc</i>	<i>flt; ftr10; fglb; fprb</i>
9	2	<i>flt20; fplt; gl; fplc</i>	<i>flt30; flt20; flb10</i>
10	6	<i>fplc; fc; fplb; fcb</i>	<i>fglt; fppt; flb; fbr10</i>
11	5	<i>fplt; fct; fplc; fc</i>	<i>flt; ftr10; fglb; fprb</i>

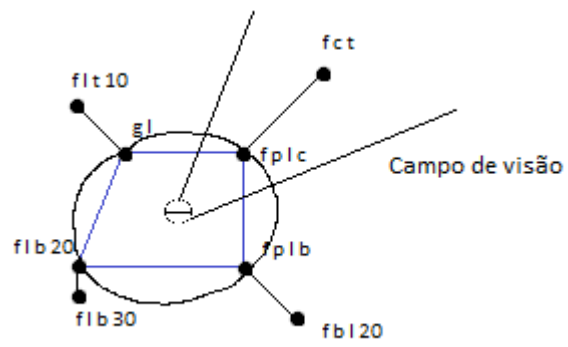


Figura 14 - Setor de defesa e campo de visão do jogador 2 no lado esquerdo do camp

4

Validação do módulo Cliente para o ambiente de simulação

Este capítulo apresenta os resultados relacionados ao módulo Cliente produzido neste trabalho para o ambiente de simulação Robocup Soccer Simulator.

Este trabalho trouxe como resultado o desenvolvimento de um Cliente para o ambiente de simulação *Robocup Soccer Simulator 2d*. Para validar estes artefatos foram definidos os seguintes passos:

- Execução de uma partida de futebol através do ambiente de simulação utilizando o módulo desenvolvido para ambos os times em jogo.
- Verificação da obediência das ações dos agentes em relação à árvore de decisões definida.

4.1 Execução de uma partida de futebol

Para a execução da partida de futebol foram utilizadas três máquinas, uma executando o Servidor e o Monitor e uma máquina para cada time presente na partida, ambas utilizando a mesma árvore de decisões definida anteriormente.

Primeiramente o Servidor foi executado na máquina servidora, seguido pela execução do Monitor.

Então as duas máquinas restantes executaram os módulos Cliente, uma máquina com os onze jogadores de um time e outra com os onze jogadores de outro time.

Após todos os clientes estarem conectados ao Servidor o Monitor foi utilizado para iniciar a partida, executada em dois tempos de cinco minutos, sendo necessária a reinicialização da partida entre cada tempo através do Monitor.

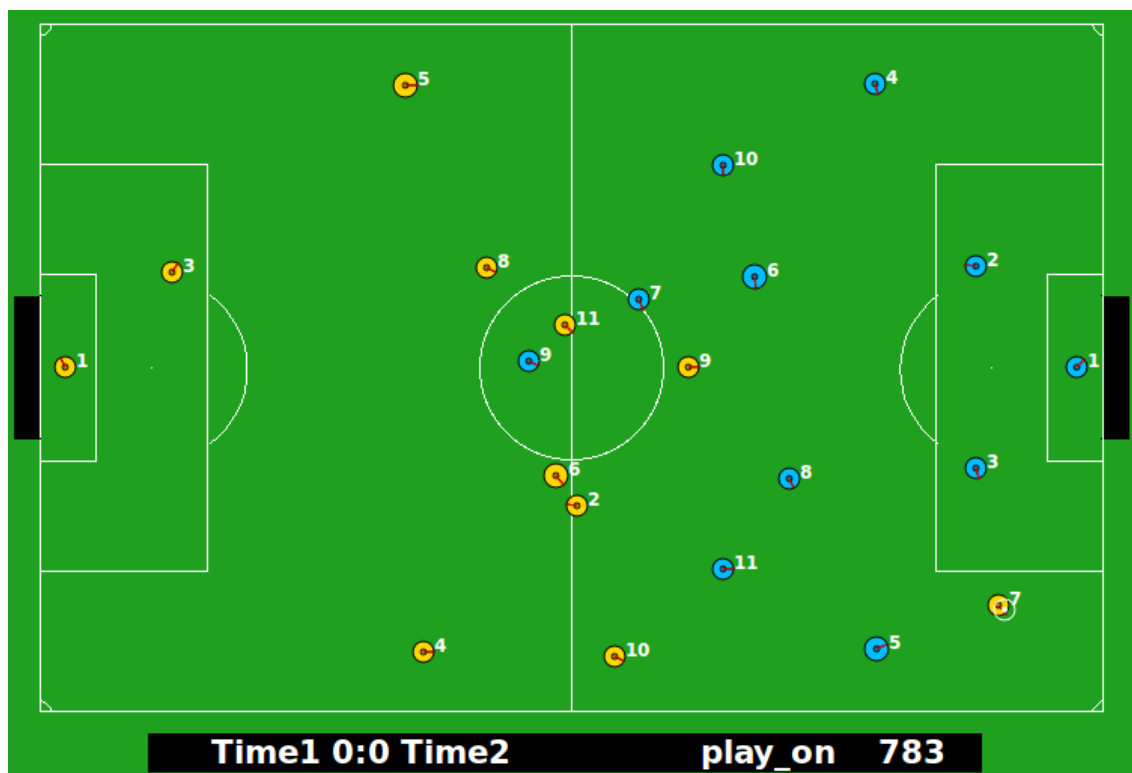


Figura 15 - Simulação em andamento

A partida ocorreu sem problemas e terminou com o resultado de 3 a 1 para o time do lado esquerdo do campo.

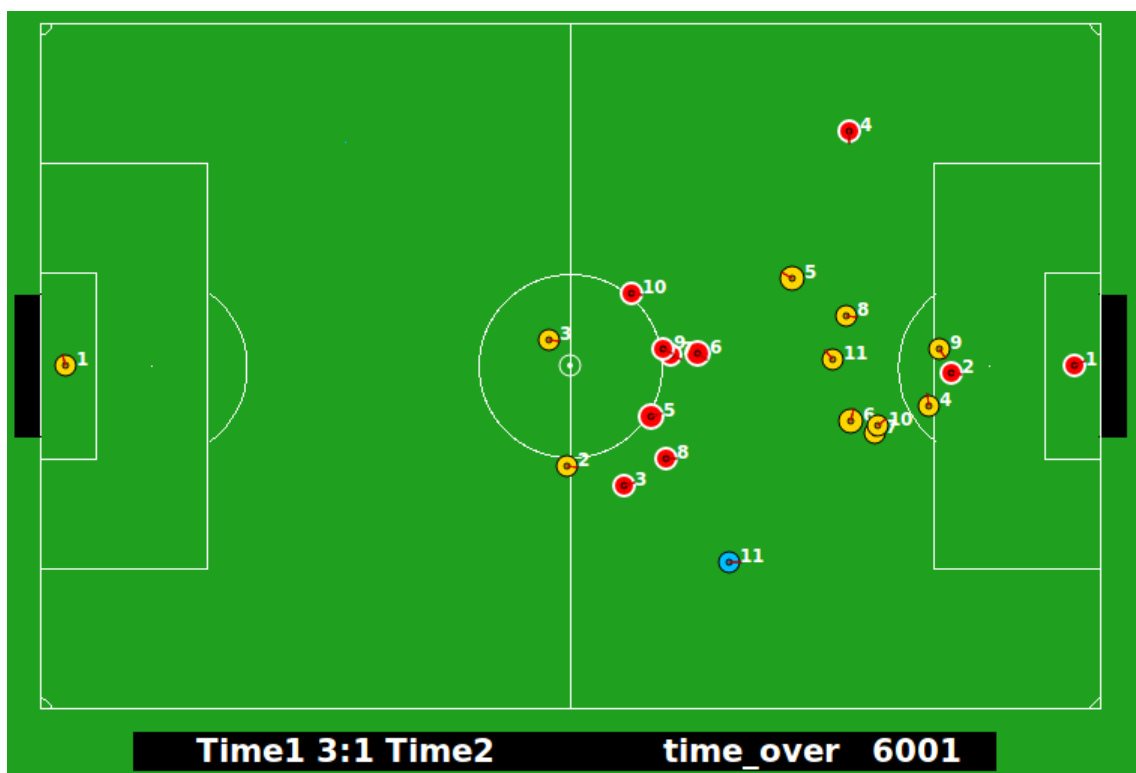


Figura 16 - Simulação finalizada

4.2 Verificação da obediência das ações

Durante a execução da partida foi possível verificar a obediência dos agentes em relação as ações definidas pela Árvore de Decisões implementada. Em alguns casos o agente tomou decisões diferentes das esperadas, isso acontece em consequência de algumas falhas em relação a troca de mensagens implementada entre o Cliente e o Servidor.

Em determinados momentos o agente tomava decisões com informações que já deveriam ter sido descartadas em função da alteração do mundo em questão.

5

Conclusões e trabalhos futuros

Neste capítulo são apresentadas as conclusões obtidas através do desenvolvimento deste trabalho bem como sugestões para trabalhos futuros que possam continuar o esforço do presente trabalho.

5.1 Conclusões

O Futebol de Robôs realmente propõe ao meio acadêmico um desafio que envolve diversas áreas, cujos estudos podem trazer grandes avanços no futuro.

O uso de um ambiente de simulação na abordagem do problema do Futebol de Robôs permite que o trabalho seja totalmente focado no desenvolvimento do *software* que compõe o sistema do problema em questão.

Do presente trabalho podemos concluir que o uso de uma Árvore de Decisões para um ambiente de tempo real como o Futebol de Robôs, requer o uso de outras técnicas em conjunto para alcançar um resultado satisfatório.

Além disso, este trabalho não explorou todos os recursos oferecidos pelo ambiente de simulação, situações como o posicionamento do agente em campo poderiam ser otimizadas se o agente pudesse utilizar todas as *flags* visíveis em campo para seu posicionamento.

Outros recursos como: apontar para uma direção, falar e mover apenas o pescoço do jogador, poderiam melhorar drasticamente a tomada de decisões do agente.

5.2 Trabalhos Futuros

5.2.1 Reestruturação da obtenção e utilização de informações sensoriais por parte do cliente

Trabalhos futuros podem melhorar o aproveitamento dos recursos fornecidos pelo simulador, e otimizar a maneira como é feita a troca de informações entre o Cliente e o Servidor e o tratamento dessas informações pelo Cliente a fim de impedir que o agente tome decisões baseado em informações que não condizem com o real estado do ambiente.

No presente trabalho apenas as informações sensoriais enviadas pelo Servidor em cada ciclo de simulação são utilizadas. Em certos momentos pode ser interessante que o Cliente requisite o envio de informações sensoriais por parte do Servidor.

5.1.2 Aprendizagem

Outra opção para a melhora da tomada de decisões do agente é o uso de um método de aprendizagem para que o agente se lembre das decisões tomadas e possa repeti-las em situações futuras.

Um método de aprendizado possível para este caso é o Aprendizado por Reforço, através deste método seria possível calibrar variáveis utilizadas nos nós da Árvore de Decisão, como no caso de chutar a bola ao gol caso o jogador esteja a uma distância mínima do gol, a Aprendizagem por Reforço poderia ser utilizada para calibrar o valor dessa distância mínima bem como outras variáveis.

Em conjunto com este método pode ser interessante aumentar o número de nós da árvore de decisão a fim de cobrir situações não consideradas pela Árvore de Decisões deste trabalho.

5.1.3 Uso de recursos do ambiente de Simulação ignorados

Por fim, existem outros dois recursos oferecidos pelo ambiente de simulação que poderiam aumentar a eficiência geral do time durante uma partida. Durante a simulação os onze jogadores iniciais possuem os mesmos atributos, mas existem outros sete tipos de jogadores, com atributos diferentes, disponíveis para serem

utilizados durante o jogo através de substituições. O simulador permite que uma outra aplicação Cliente execute a função de Técnico, realizando substituições e passando mensagens aos jogadores, esta passagem de mensagens trás diversas novas possibilidades para alteração de estratégia durante a execução da simulação, através de mudanças táticas na formação da equipe e mudanças técnicas com substituições dos jogadores em campo.

O outro recurso que pode ser utilizado em um trabalho futuro é o goleiro, no presente trabalho nenhum tratamento especial foi dado ao goleiro, mas o simulador permite que um dos jogadores conectados faça este papel durante a simulação, o goleiro possui certas vantagens no ambiente da simulação, podendo pegar a bola com as mãos, assim mantendo a bola em sua posse enquanto se movimenta dentro da grande área.

5.1.4 Tomada de Decisões com relação ao futuro

No presente trabalho um agente toma uma decisão de acordo com sua atual percepção do mundo em que atua. Isso faz com que o agente tome decisões imprecisas em determinados momentos. Para melhorar a tomada de decisões seria interessante que o agente fosse capaz de prever o estado do mundo em um instante futuro, isso pode ser feito utilizando informações sobre a direção e a velocidade dos objetos visíveis bem como avaliando a distância e direção destes objetos em instantes passados.

6

Referências Bibliográficas

Bellman, R. E. *"An introduction to artificial intelligence: Can computers think?"*. Ed. Body & Fraser Pub. Co., 1978

Brooks, R. A., *"Elephants don't play chess"*, Robotics and Autonomous Systems, Elsevier Science Publishers, 1990

Browning, B., Tryzelaar, E., *"ÜberSim: A Multi-Robot Simulator for Robot Soccer"*. 2003

Gripp, J. A. B., Matsuura, J. P. *"Desenvolvimento de Robô Móvel Radiocontrolado"*. Anais do 13º encontro de Iniciação Científica e Pós Graduação do ITA, 2007

Haugeland, J. *"Artificial Intelligence: the very idea"*. Ed. MIT Press, 1985

Kim, J. O., Kim, J. H., Shim, H. S., Jung, M. J., Choi, I. H., *"A Cooperative Multi-Agent System and Its Real Time Application to Robot Soccer"*, International Conference on Robotics and Automation, Albuquerque, New Mexico - 1997

Kitano, H., Asada, M., Osawa, E., Noda, I., Kuniyoshi, Y., Matsubara, H. *"Robocup: A Challenge problem for AI"*. AI Magazine, Vol. 18, No 1, 1997.

Kok, R. J., Vlassis, V., Groen, G. *"UvA Trilearn 2003: Team Description"*. Faculty of Science, University of Amsterdam, 2003.

Mackworth, A. "On seeing Robots". *Mind Readings: introductory selections in cognitive science*, Ed. Paul Thagard, 1998, pp. 271 - 275

Poole, D., Mackworth, A., Goebel, R. "Computational intelligence: a logical approach". Ed. Oxford University Press, 1998.

Russel, S., Norvig, P., "Artificial Intelligence: A modern Approach", Ed. Prentice Hall, 2003.

Stone, P., Sutton, R. S., Kuhlmann, G., "Reinforced Learning for RoboCup Soccer Keepaway", 5 International Society for Adaptive Behavior, 2005, Vol. 13(3): 165-188

Sutton, R. S., Barto, A. G., *Journal of Cognitive Neuroscience*, pp. 126-134

Wooldridge, M., Jennings, N. R. "Intelligent Agents: theory and practice", The Knowledge Engineering Review, Vol. 10:2, 1995

Xavier, R. O., Barbosa, R., Matsuura, J. P. "O time de futebol simulado ITANDROIDS 2D". III Encontro de Robótica Inteligente, Campo Grande - MS

"Robocup Soccer Simulator Users Manual".

http://sourceforge.net/apps/mediawiki/sserver/index.php?title=Main_Page

7 Apêndices e Anexos

7.1 Apêndice A

Instalando o Robocup Soccer Simulator 2d no Ubuntu 11.04

Nota: Este tutorial é destinado a instalação do Robocup Soccer Simulator 2d no sistema operacional Ubuntu. Note que para este tutorial foram utilizadas versões específicas do Soccer Simulator e do Ubuntu, apesar dos procedimentos descritos abaixo poderem funcionar para versões diferentes de ambos os sistemas, seu perfeito funcionamento não pode ser garantido através destas instruções.

Obtendo e instalando o Ubuntu 11.04

Baixe o arquivo .iso do Ubuntu 11.04 em www.ubuntu.com. Este é um arquivo de imagem de disco, será necessário queimá-lo em um CD para realizar a instalação.

Após queimar o CD contendo o Ubuntu 11.04, insira-o em em seu Drive de CD e reinicie a máquina com o CD como primeira opção de boot. Isso fará com que o menu de instalação do Ubuntu seja iniciado, siga as instruções para concluir a instalação.

Obtendo os pacotes necessários

Abra um terminal (Applications → Accessories → Terminal) e digite:

```
#sudo apt-get install flex  
#sudo apt-get install libxt-dev  
#sudo apt-get install libaudio-dev  
#sudo apt-get install libxrender-dev
```

```
#sudo apt-get install libfreetype6-dev
#sudo apt-get install libfontconfig1-dev
#sudo apt-get install libgtkgl2.0-dev
#sudo apt-get install libboost-all-dev
#sudo apt-get install bison
#sudo apt-get install libqt4-core
#sudo apt-get install qt4-dev-tools
```

Nota1: A primeira vez que você executar um comando “sudo” no terminal a senha de será requisitada.

Nota2: Entre cada comando “apt-get install” pode ser necessário confirmar a instalação pacote (y). Em alguns casos o download pode demorar alguns minutos.

Preparando os diretórios

No diretório home crie uma nova pasta chamada rc.

Dentro da pasta rc crie uma nova pasta chamada 2d.

Obtendo o Soccer Simulator 2d

Baixe os arquivos abaixo e salve-os no diretório “rc/2d/” criado no passo 3.

- Base 12.1.3:
<http://sourceforge.net/projects/sserver/files/rcssbase/12.1.3/rcssbase-12.1.3.tar.gz/download>
- Server 15.0.1:
<http://sourceforge.net/projects/sserver/files/rcssserver/15.0.1/rcssserver-15.0.1.tar.gz/download>

- Monitor 15.0.0:

<http://sourceforge.net/projects/sserver/files/rcssmonitor/15.0.0/rcssmonitor-15.0.0.tar.gz/download>

Extraia os arquivos na mesma pasta, para isso clique com o botão direito do mouse no arquivo e selecione a opção “Extract Here”, faça isso para os 3 pacotes baixados.

Nota: Note que novas versões de qualquer um dos 3 pacotes podem ter sido lançadas quando você estiver lendo este tutorial. Os *links* acima levam diretamente aos downloads das versões utilizadas aqui. Os passos abaixo podem funcionar para estas novas versões, mas talvez seja necessário obter outros pacotes não descritos no passo 2.

Instalando o Soccer Simulator 2d

- Instalando o Base

O Base é um conjunto de arquivos necessários para a instalação do Server e do Monitor.

Para instalá-lo abra um terminal e navegue para a pasta onde o Base foi extraído.

```
#cd /rc/2d/rcssbase-12.1.3
```

Agora é necessário configurar e compilar os arquivos necessários. Basta utilizar os comandos abaixo:

```
#!/configure
```

```
#sudo make
```

```
#sudo make install
```

- Instalando o Server

O Server é responsável por controlar a partida. Para instalar basta repetir os passos da Instalação do Base, mas desta vez na pasta do Server. Se ainda estiver no terminal utilizado no passo anterior digite:

```
#cd ../rcssserver-15.0.1
```

```
#./configure
```

```
#sudo make
```

```
#sudo make install
```

- Instalando o Monitor

O Monitor é utilizado para assistir a partida enquanto acontece. Note que este pacote é necessário para o funcionamento do Simulador, é através dele que as ações do arbitro humano são executadas, além disso é muito útil para fins de depuração e teste. Siga os mesmos passos para instalar:

```
#cd ../rcssmonitor-15.0.0
```

```
#./configure
```

```
#sudo make
```

```
#sudo make install
```

Testando o Server e o Monitor

Para executar o Servidor e o Monitor basta executar o comando

```
#rcsoccersim
```

Você verá algumas mensagens do Servidor aparecerem no terminal e uma nova janela será aberta exibindo o Monitor.



Agora basta criar o Cliente. Para instruções sobre como criar um Cliente e sobre a utilização do Server e do Monitor, baixe o manual do *RoboCup Soccer Simulator 2d* em

<http://sourceforge.net/projects/sserver/files/rcssmanual/9-20030211/manual-20030211.pdf/download>

1.1 Apêndice B

Protocolos de comunicação entre Cliente e Servidor do *Robocup Soccer Simulator 2d*

Para que um cliente se comunique com o servidor do *Robocup Soccer Simulator 2d* o primeiro passo é realizar a conexão com o servidor, isto é feito enviando uma mensagem ao servidor contendo uma *String* “(init nomeDoTime [(version versão)])”. Esta mensagem deve ser enviada a porta 6000 (padrão) da máquina onde o Servidor está rodando.

O servidor deverá responder com uma mensagem do tipo “(init lado numero playMode)”.

Por exemplo: (init Time1 (version 15))

A resposta do servidor, caso este seja o primeiro cliente a se conectar, seria:

(init l 1 beffore_kick_off)

Tabela 4 - Cliente para Servidor - Conectar, reconectar, e desconectar.

De cliente para servidor	Resposta
(init <i>TeamName</i> [(version <i>VerNum</i>)] [(goalie)])	(init <i>Side Unum PlayMode</i>)
<i>TeamName</i> ::= (- _ a-z A-Z 0-9)+	
<i>VerNum</i> ::= a versão do protocolo (e.x. 15)	<i>Side</i> ::= l r
	<i>Unum</i> ::= 1 ~ 11
	<i>PlayMode</i> ::= um dos play modes
	(error connection_failed)
	(error illegal_command_form)
	(error illegal_teamname)
	(error illegal_teamname_or_too_long_teamname)
	(error no_more_team)
	(error no_more_player_or_goalie_or_illegal_client_version)
(reconnect <i>TeamName Unum</i>)	(reconnect <i>Side PlayMode</i>)
<i>TeamName</i> ::= (- _ a-z A-Z 0-9)+	
	<i>Side</i> ::= l r
	<i>PlayMode</i> ::= um dos modes
	(error connection_failed)
	(error socket_open_failed)
	(error illegal_command_form)
	(error illegal_client_version)
	(error no_such_team_or_player)
	(error cannot_reconnect_while_playon)
(bye)	

É importante notar que a porta 6000 é utilizada apenas na conexão inicial com o Servidor, ao receber a resposta do Servidor cabe ao cliente utilizar o *socket* para descobrir a porta de origem da mensagem do Servidor, e utilizar esta porta para mensagens futuras.

Após a conexão, o servidor passa a enviar mensagens sensoriais para o Cliente. Cabe ao cliente interpretar tais mensagens.

Tabela 5 - De Servidor para Cliente - Informações sensoriais

Do Servidor para o Cliente	
(sense_body <i>Time</i>)	<i>Time</i> ::= ciclo de simulação do servidor
(view_mode { high low } { narrow normal high })	<i>Stamina</i> ::= número real positivo
(stamina <i>Stamina Effort StaminaCapacity</i>)	<i>Effort</i> ::= número real positivo
(speed <i>AmountOfSpeed DirectionOfSpeed</i>)	<i>StaminaCapacity</i> ::= número real positivo
(head_angle <i>HeadAngle</i>)	<i>AmountOfSpeed</i> ::= número real positivo
(kick <i>KickCount</i>)	<i>DirectionOfSpeed</i> ::= -180 ~ 180 graus
(dash <i>DashCount</i>)	<i>HeadAngle</i> ::= -180 ~ 180 graus
(turn <i>TurnCount</i>)	<i>*Count</i> ::= 0 ou inteiro positivo
(say <i>SayCount</i>)	<i>ArmMovableCycles</i> ::= 0 ou inteiro positivo
(turn_neck <i>TurnNeckCount</i>)	<i>ArmExpiresCycles</i> ::= 0 ou inteiro positivo
(catch <i>CatchCount</i>)	<i>ArmTargetDistance</i> ::= positive real number
(move <i>MoveCount</i>)	<i>ArmTargetDirection</i> ::= -180 ~ 180 degrees
(change_view <i>ChangeViewCount</i>)	<i>UniformNumber</i> ::= 1 ~ 11
(arm (movable <i>ArmMovableCycles</i>) (expires <i>ArmExpiresCycles</i>))	<i>TackleExpiresCycles</i> ::= 0 ou inteiro positivo
(target <i>ArmTargetDistance ArmTargetDirection</i>)	<i>FoulChargedCycles</i> ::= 0 ou inteiro positivo
(count <i>PointtoCount</i>)	
(focus (target { none l <i>UniformNumber</i> r <i>UniformNumber</i> })	
(count <i>AttentiontoCount</i>)	
(tackle (expires <i>TackleExpiresCycles</i>) (count <i>TackleCount</i>))	
(collision { none [(ball)] [(player)] [(post)]})	
(foul (charged <i>FoulChargedCycles</i>) (card { none yellow red }))	
(see <i>Time ObjInfo</i> *)	<i>Distance</i> ::= número real positivo
<i>Time</i> ::= ciclo de simulação do soccerserver	<i>Direction</i> ::= -180 ~ 180 graus
<i>ObjInfo</i> ::= (<i>ObjName Distance Direction DistChange DirChange BodyFacingDir HeadFacingDir [PointingDir] [t k]</i>)	<i>DistChange</i> ::= número real
(<i>ObjName Distance Direction DistChange DirChange [PointingDir] [t k]</i>)	<i>DirChange</i> ::= número real
(<i>ObjName Distance Direction [PointingDir] [t k]</i>)	<i>BodyFacingDir</i> ::= -180 ~ 180 graus
(<i>ObjName Direction</i>)	<i>HeadFacingDir</i> ::= -180 ~ 180 graus
	<i>ointingDir</i> ::= -180 ~ 180 graus
<i>ObjName</i> ::= (p [" <i>Teamname</i> "]	<i>Teamname</i> ::= Strign
[<i>UniformNumber</i> [goalie]])	<i>UniformNumber</i> ::= 1 ~ 11
(b)	

| (g [l|r])
 | (f c)
 | (f [l|c|r] [t|b])
 | (f p [l|r] [t|c|b])
 | (f g [l|r] [t|b])
 | (f [l|r|t|b] 0)
 | (f [t|b] [l|r] [10|20|30|40|50])
 | (f [l|r] [t|b] [10|20|30])
 | (l [l|r|t|b])
 | (P)
 | (B)
 | (F)
 | (G)

(hear Time referee Message)	Time ::= ciclo de
(hear Time Sender "Message")	simulação do servidor
(hear Time Direction our UniformNumber "Message")	Direction ::= -180 ~ 180
(hear Time Direction opp "Message")	graus
(hear Time our UniformNumber)	UniformNumber ::= 1 ~
(hear Time opp)	11
(hear Time Online_Coach Coach_Language_Message)	Message ::= string
Sender ::= online_coach_left online_coach_right coach self	Online_Coach ::= online_
	coach_left online_coac
	h_right

Após ser capaz de tartar as mensagens recebidas do Servidor, o próximo passo é enviar comandos de ações ao mesmo. Cada comando possui uma sintaxe diferente.

Tabela 6 - De Cliente para Servidor - Ações de jogador

De Cliente para Servidor	Apenas uma vez por ciclo
(attentionto Side Number) (attentionto off) Side ::= l r our opp TeamName TeamName ::= (- _ a-z A-Z 0-9)+ Number ::= 1 ~ 11	Não
(catch Direction) Direction ::= <i>minmoment</i> ~ <i>maxmoment</i> graus	Sim
(change_view Width Quality) Width ::= narrow normal wide Quality ::= high low	Não

(dash <i>Power Direction</i>)	Sim
(dash <i>Power</i>)	
<i>Power</i> ::= min_dash_power ~ max_dash_power	
<i>Direction</i> ::= min_dash_angle ~ max_dash_angle	
Se <i>Direction</i> for omitido, a direção 0 é usada automaticamente.	
Nota: <i>dash</i> para trás consome o dobro de stamina.	
(kick <i>Power Direction</i>)	Sim
<i>Power</i> ::= minpower ~ maxpower	
<i>Direction</i> ::= minmoment ~ maxmoment graus	
(move <i>X Y</i>)	Sim
<i>X</i> ::= qualquer número real	
<i>Y</i> ::= qualquer número real	
Nota: O árbitro automaticajusta a posição do jogador se ele se mover para for a do campo.	
(pointto <i>Distance Direction</i>)	Não
(pointto off)	
<i>Distance</i> ::= qualquer número real	
<i>Direction</i> ::= qualquer número real	
pointto é relative à direção de <i>global neck angle</i>	
(say " <i>Message</i> ")	Não
(say <i>Message</i>)	
<i>Message</i> ::= String	
(sense_body)	Não
O servidor retorna uma mensagem de <i>sense_body</i>	
(score)	Não
O servidor retorna	
(score <i>tempo nossaPontuação pontuaçãoDeles</i>)	
(tackle <i>Direction</i> [<i>Foul</i>]) :	Sim
<i>Direction</i> ::= minmoment ~ maxmoment graus	
<i>Power</i> ::= -	
max_back_tackle_power ~ max_tackle_power	
<i>Foul</i> ::= true false	
Se <i>Foul</i> não for omitida o jogador tenta cometer uma falta intencional.	
(turn <i>Moment</i>)	Sim
<i>Moment</i> ::= minmoment ~ maxmoment graus	

(**turn_neck** *Angle*)

Sim

Angle ::= **minneckmoment** ~ **maxneckmoment** graus

turn_neck é relative a direção do corpo, pode ser

invocado no mesmo ciclo que dash e kick
