

Futebol de Robôs Bípedes: Simulação 3D

Prof. Marco Simões



Agenda

■ Parte I: Introdução

◆ Futebol de Robôs

- ★ Histórico

- ★ Motivação

- ★ Robocup e suas Ligas

◆ Bahia Robotics Team

■ Parte II: A Liga de Simulação 3D

◆ Objetivos e Descrição

◆ Arquitetura do Simulador

◆ Instalação do Simulador

◆ Modelo do robô NAO na Simulação 3D

◆ Estrutura de mensagens do Simulador 3D

Agenda

- Parte III: Planejando Movimentos em robôs Humanóides Simulados
 - ◆ Abordagens para criar um time de futebol de robôs 3D
 - ◆ Conceito de Movimentos dos robôs na liga 3D
 - ◆ A Ferramenta IFAS3D
- Parte IV – Programando IA a partir de um time-base
 - ◆ O que já vem pronto em um time base
 - ◆ Estrutura e conceitos do Magma Offenburg 2011

Parte I: Introdução

Breve Histórico

- O primeiro problema padrão proposto foi o jogo de Xadrez (1950)
- Em 1997, o Deep Blue da IBM derrotou o campeão mundial de xadrez Kasparov
 - Foram 06 partidas disputadas entre março e novembro de 1997
 - 03 partidas empataram, Kasparov venceu 01 (a 1ª) e o Deep Blue venceu 02 (a 2ª e a 6ª) vencendo a série por 2x1

Breve Histórico

■ 1992

- Prof. Alan MackWorth mencionou pela 1ª vez a idéia de robôs jogadores de futebol
- Tóquio – Workshop on Grand Challenges in Artificial Intelligence

■ 1993

- Minoru Asada, Yasuo Kuniyoshi, Hiroaki Kitano e outros propuseram a Robot J-League
- Projeto renomeado para Robot World Cup (RoboCup) por fortes pressões internacionais
- Itsuki Noda (Japão) vinha usando o jogo de futebol para suas pesquisas em sistemas multiagentes e iniciou o desenvolvimento de um simulador
- O Laboratório do Prof. Minoru Asada (Osaka) e a Prof. Manuela Veloso e seu aluno Peter Stone (Carnegie Mellon University) vinham trabalhando em robôs jogadores de futebol
- Setembro: 1º anúncio público da iniciativa e início das discussões das regras e infraestrutura

Breve Histórico

■ 1995

■ 1ª demonstração pública do Soccer Server – simulador oficial de futebol de robôs

■ Anunciada a 1ª edição dos jogos e conferências Robot World Cup em 1997

■ 1996

■ Pré-RoboCup – Osaka: 8 times competindo em simulação de futebol e demonstração de robôs de médio porte

■ 1997

■ I RoboCup – Nagoya: mais de 40 times competiram em ligas simuladas e físicas para mais de 5.000 espectadores

Objetivo da RoboCup



By the middle of the 21st century, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup.

- Projeto Marco
- Desafio Padrão

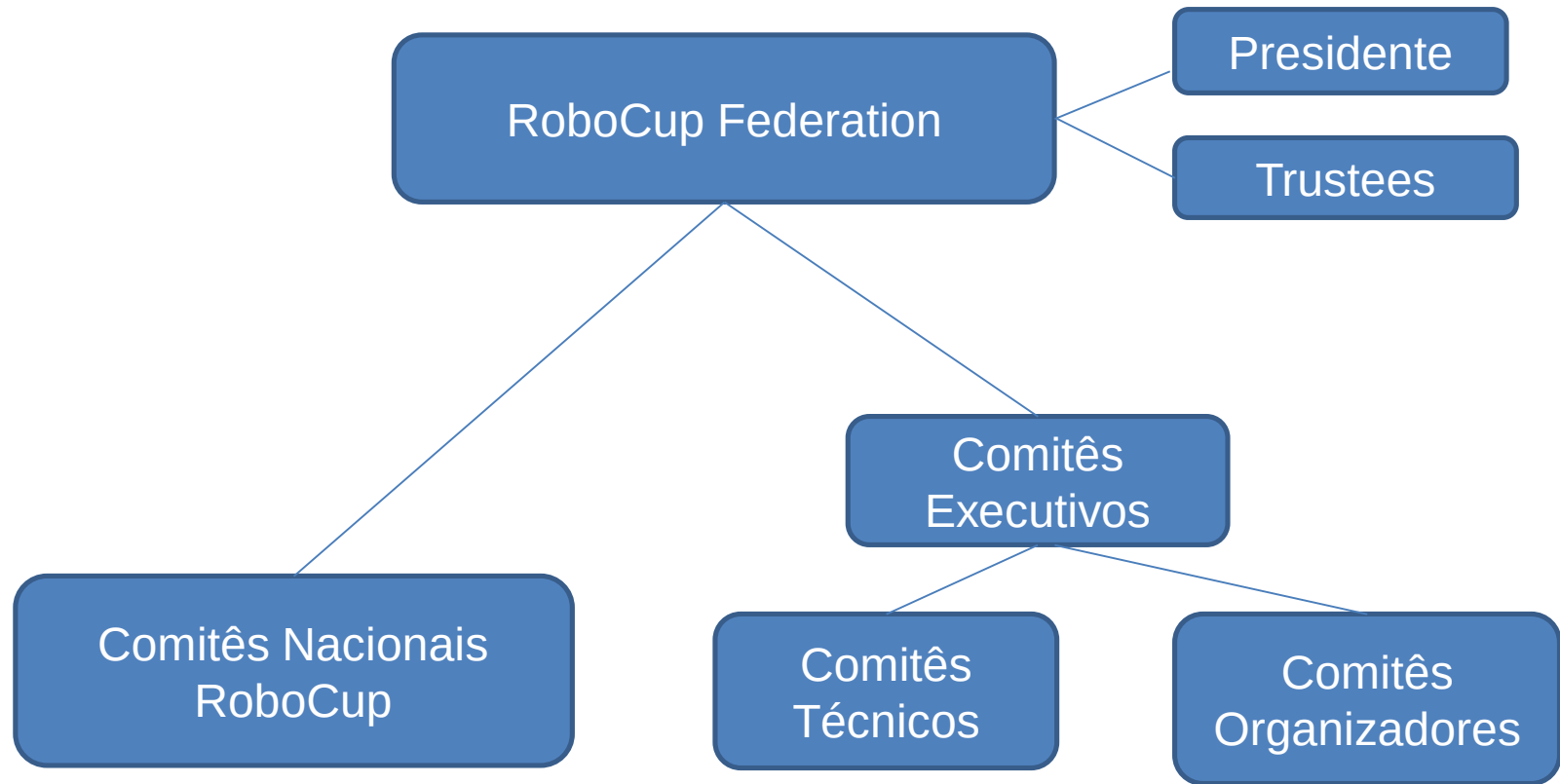
Xadrez x RoboCup

	Xadrez	RoboCup
Ambiente	Estático	Dinâmico
Mudança de Estado	Por jogada	Tempo Real
Acessibilidade	Completa	Parcial
Sensores	Simbólicos	Não Simbólicos
Controle	Central	Distribuído

Motivação para o Uso do Futebol de Robôs

- Reúne grande partes dos desafios presentes em problemas eminentemente distribuídos do mundo real
 - Veículos autônomos;
 - Busca de informação em bases de dados distribuídas não estruturadas;
 - Planejamento de geração de energia elétrica;
 - Recomposição de linhas de transmissão;
 - Automação doméstica;
 - Resgate de vítimas em desastres;
 - Controle de tráfego aéreo e urbano, etc
- Jogo em tempo real, com ambiente dinâmico
- Demanda coordenação entre agentes independentes para a solução de um problema complexo

Organização RoboCup



Organização da RoboCup

- Evento Mundial Itinerante Anual: RoboCup
 - ▣ Copa do Mundo de Robótica
 - ▣ Simpósio RoboCup
 - ▣ Equipes precisam qualificar-se para ir ao evento (dezembro a fevereiro de cada ano)
- Comitês Nacionais/Regionais RoboCup
 - ▣ Brazilian National Committee
 - ▣ Organizam eventos “Open” – RoboCup Brazil Open, Latin American RoboCup Open

RoboCup no Brasil

- Competição Brasileira de Robótica – CBR
 - ▣ RoboCup Brazil Open – Oficial desde 2007
 - ▣ Competição IEEE Brasileira de Robôs
- Latin American Robotics Competitions – LARC
 - ▣ Latin American RoboCup Open
 - ▣ IEEE Latin American Competition

Histórico LARC e CBR

Ano	CBR	LARC
2002	-	Santiago, Chile
2003	Bauru, SP	Bauru, Brasil
2004	Salvador, BA	Mexico
2005	São Luis, MA	São Luis, Brasil
2006	Campo Grande, MS	Santiago, Chile
2007	Florianópolis, SC	Mexico
2008	Salvador, BA	Salvador, Brasil
2009	Brasília, DF	Valparaiso, Chile
2010	S. Bern. do Campo,SP	São Bernardo do Campo, Brasil
2011	São João Del Rey, MG	Bogotá, Colômbia
2012	Fortaleza, CE	Fortaleza, Brasil
2013	Fortaleza, CE	Arequipa, Peru

Histórico RoboCup

Ano	Local	Ano	Local
1997	Nagoya, Japão	2006	Bremen, Alemanha
1998	Paris, França	2007	Atlanta, EUA
1999	Estocolmo, Suécia	2008	Suzhou, China
2000	Melbourne, Austrália	2009	Graz, Áustria
2001	Seattle, EUA	2010	Cingapura
2002	Fukoka/Busan, Japão	2011	Istambul, Turquia
2003	Padova, Itália	2012	México
2004	Lisboa, Portugal	2013	Eindhoven, Holanda
2005	Osaka, Japão	2014	João Pessoa, Brasil

Próximas Competições

■ CBR/LARC

◆ 2014 – São Carlos, SP, Brasil – 18 a 22 de outubro –
<http://www.cbrobotica.org>

■ RoboCup

◆ **2014 – João Pessoa, Brasil –** <http://www.robocup2014.org>

◆ 2015 - Tailândia

Ligas RoboCup

☐ Futebol

- ☐ Robôs de pequeno porte
- ☐ Robôs de médio porte
- ☐ Robôs Humanóides
- ☐ Plataforma Padrão
- ☐ Simulação

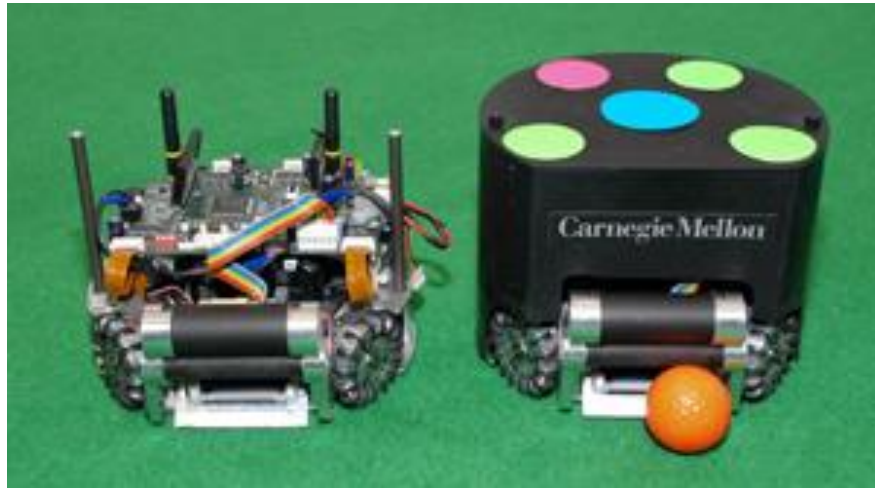
☐ RoboCup Rescue

- ☐ Robôs Físicos
- ☐ Robôs Simulados

☐ Robocup@Home

☐ Robocup Junior

Robôs de Pequeno Porte



Robôs de Médio Porte



Robôs Humanóides



Plataforma Padrão



Robocup Rescue



☐ Clique para editar o formato do texto da estrutura de tópicos



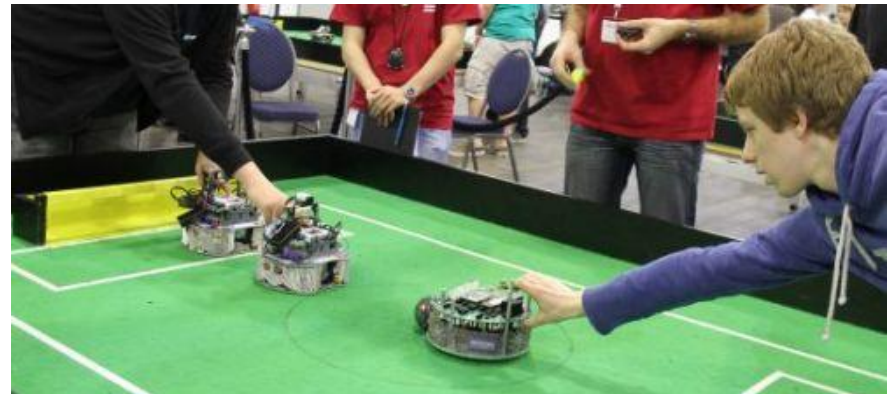
■ Quarto nível

■ Quinto nível

Robocup@Home



Robocup Junior



Futebol de Robôs Simulados

- 2D e 3D

- Utilizam simuladores disponibilizados gratuitamente (GNU)

 - ◆ RCSSERVER (2D)

 - ◆ SimsPark (3D)

- Os simuladores rodam em Linux (3D também em Windows)

- Comunica-se com os jogadores usando interface sockets

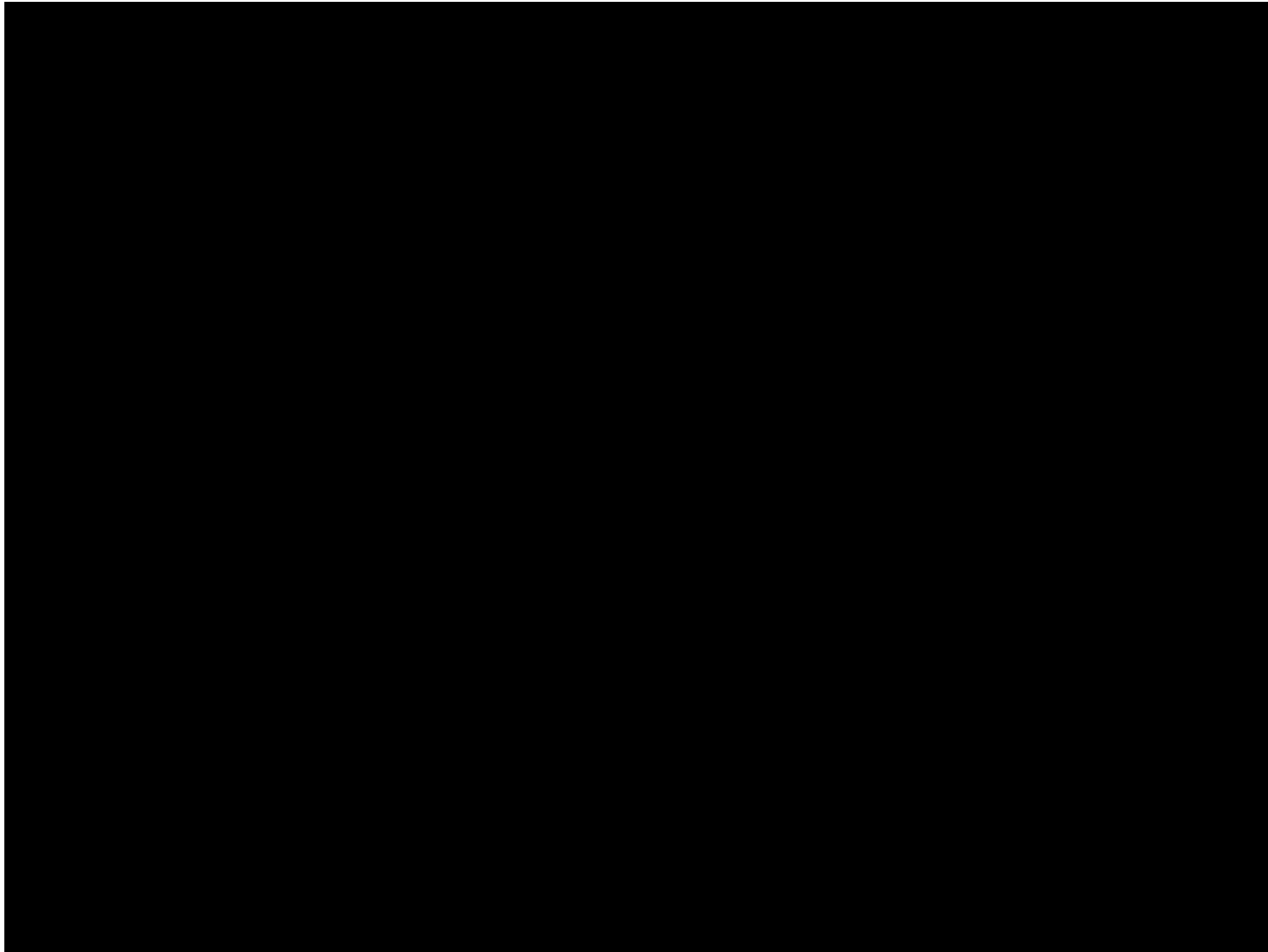
- Os jogadores (agentes inteligentes) podem ser implementados em qualquer linguagem

 - ◆ Mais usadas: C/C++ e Java

3D



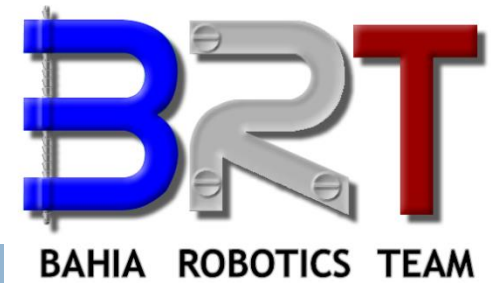
3D – Final CBR 2013



2D: Robocup Brazil 2009 – decisão do 3º lugar: Bahia2D x MecaTeam



Bahia Robotics Team



☐ Consórcio Baiano de Grupos de Pesquisa interessados em investigar o uso de Inteligência Artificial para a programação de times de robôs autônomos, conforme a proposta da Robocup Federation

☐ Criado e organizado pelo ACSO/UNEB

☐ Histórico de parcerias com estudantes e pesquisadores de diversas instituições baianas (UFBA, SENAI/CIMATEC, FIB, FTC, etc)

☐ Representado nos comitês executivos da RoboCup Brasil

☐ Representado na comissão organizadora do mundial RoboCup 2014

☐ Organizador da etapa Bahia da OBR

Conquistas do BRT

☐ RoboCup

- ☐ Qualificado para os mundiais 2007, 2008, 2009, 2010 e 2011, 2013 e 2014

- ☐ **3o Lugar Mundial – MR – 2009**

- ☐ 4o Lugar Mundial – MR - 2010

☐ CBR

- ☐ Campeão Brasileiro – 3D – 2013

- ☐ Campeão Brasileiro - @Home - 2013

- ☐ Campeão Brasileiro – MR- 2007

- ☐ 3o Lugar – 2D – 2007

- ☐ 3o Lugar – 2D - 2009

Conquistas do BRT

- LARC

- Campeão Latino-americano e Brasileiro – MR – 2008 e 2010
- Vice-campeão Latino-americano e Brasileiro– 2D – 2008
- Vice-campeão Latino-americano e Brasileiro – 3D – 2010
- Vice-campeão Latino-americano e Brasileiro – IEEE Open -2012
- 3o Lugar Latino-americano e Brasileiro – 3D – 2012

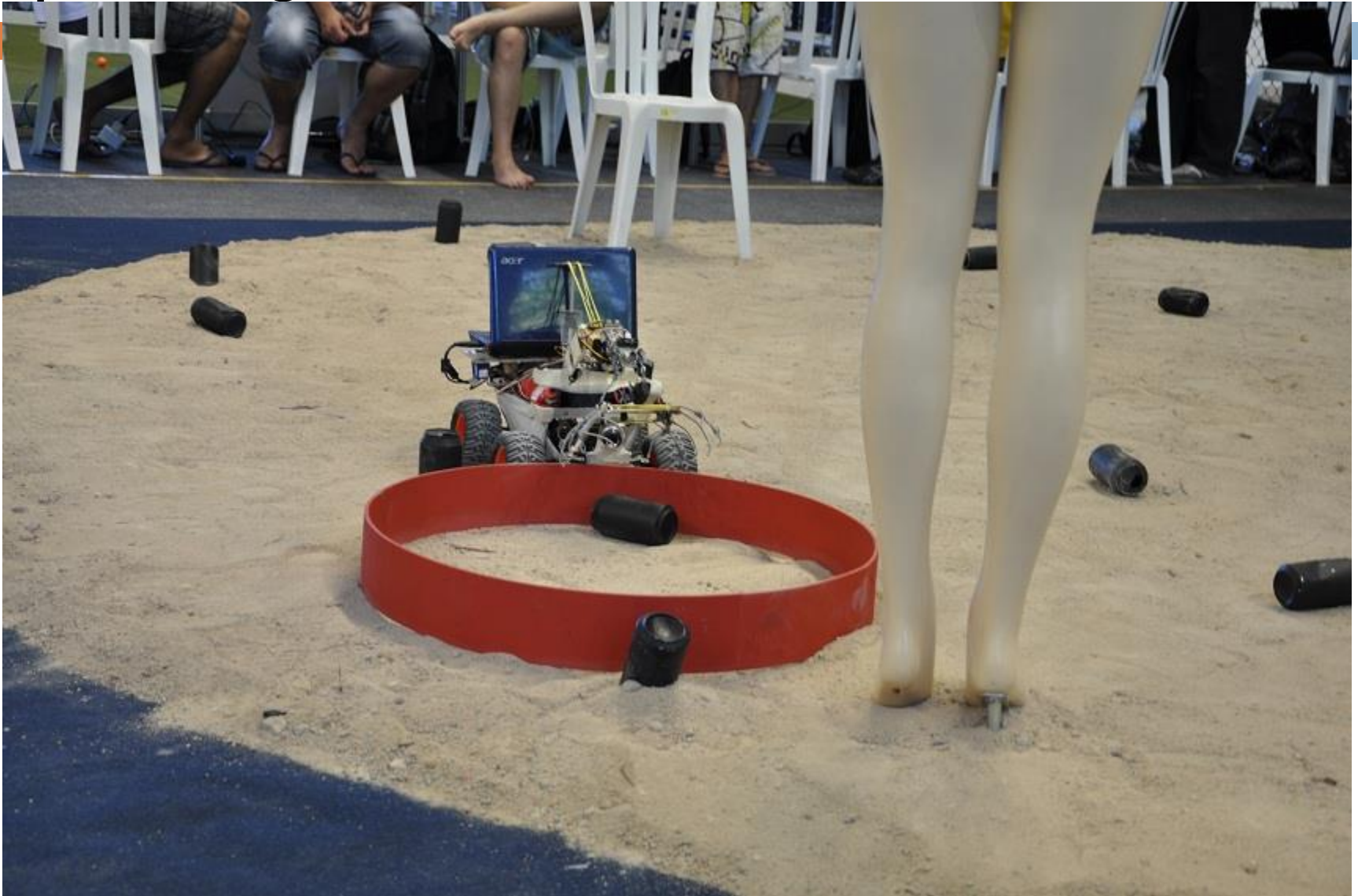
- Outras Competições

- German Open 2008 – Vice-campeão - MR
- Japan Open 2009 – Vice-campeão - MR
- Japan Open 2010 – Campeão – MR
- Iran Open 2010 – Vice-campeão – MR
- Iran Open 2011 – Vice-campeão – MR
- Portuguese Open 2014 – 3º Lugar – 3D

Realidade Mista (MR)



IEEE Open: Competição Intermediária para Ligas Físicas



Parte II: A Liga de Simulação 3D

Conceitos e Desafios da Liga 3D

- A liga 3D tem como principal desafio simular robôs jogando em 3 dimensões
 - Aproxima-se mais do desafio global da RoboCup
- Desde 2008, utiliza robôs reais na sua simulação
 - NAO da Aldebaran Robotics, usado na liga Standard Platform
- Utiliza o simulador SimsPark
 - Simulador geral de robótica que permite simular diversos modelos de robôs reais ou fictícios

Conceitos e Desafios da Liga 3D

■ Atualmente, as questões de visão computacional são abstraídas

- ◆ Visão chega para os robôs já pré-processadas como na liga 2D

- ◆ Existem simuladores como o MS Robotics Studio onde os robôs recebem quadros de video para serem processados

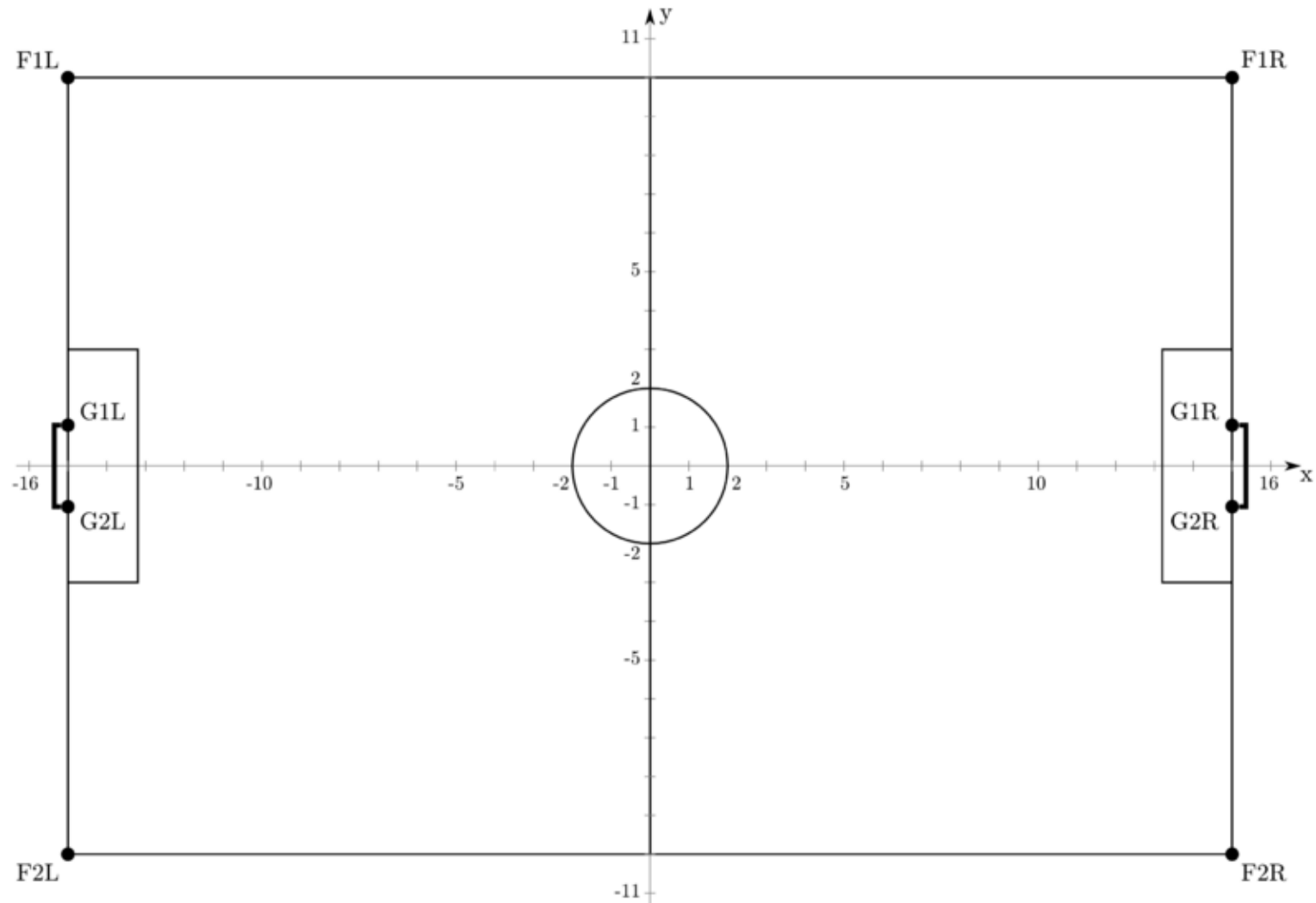
■ Os desafios presentes são uma extensão da 2D, pois os robôs são humanóides e portanto possuem dezenas de articulações a serem manipuladas

■ Atualmente jogam 11 robôs por time

■ Não existem técnicos nem jogadores reservas

■ Conexão TCP

Dimensões e Flags Percebidas por um Agente Jogador



Árbitros

□Automático

- ▣ Detecta gols, escanteios, laterais e tiro de meta.
- ▣ Impedimento apenas experimental
- ▣ Alguns tipos de falta: beam

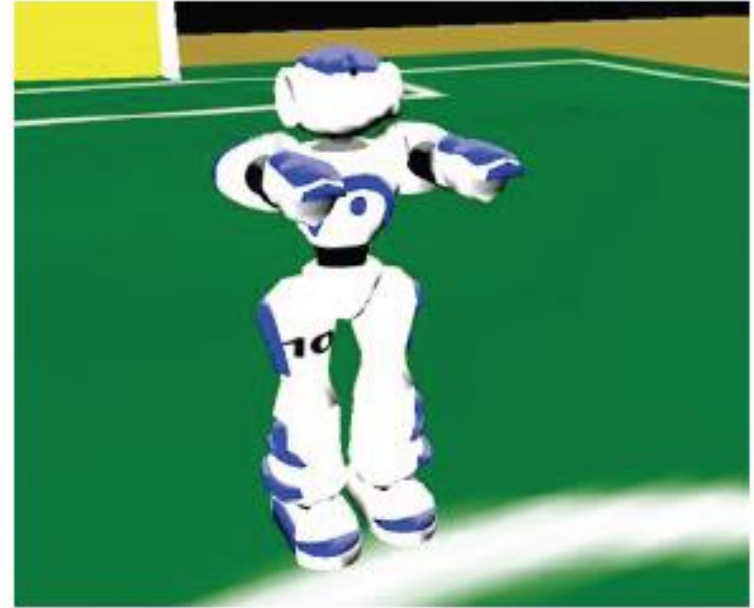
□Humano

- ▣ Situações de jogo parado são assinaladas e resolvidas pelo árbitro humano (movendo a bola para um local aleatório ou cobranças de falta)

NAO



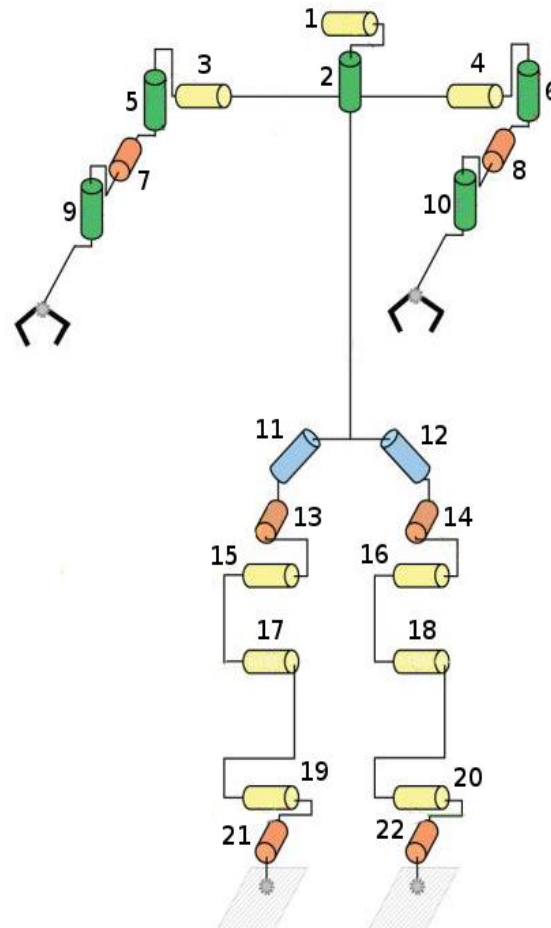
(a) real robot



(b) virtual robot

- Tem 57 cm de altura e pesa 4,5 Kg
- É um robô bípede com 22 graus de liberdade

Esqueleto do NAO



Instalação do SIMSPARK e RCSSSERVER3D

☐ Visite:

http://simspark.sourceforge.net/wiki/index.php/Main_Page

Executando o Simulador

- `rcssserver3d` executa apenas o servidor
- `rcssmonitor3d --server <ip_servidor>`
executa apenas o monitor
- `rcsoccersim3d` executa servidor e monitor
- `./robviz.sh` executa o monitor RoboViz

RCSSServer3D executando



RoboViz

□ Visite: <https://sites.google.com/site/umroboviz/>

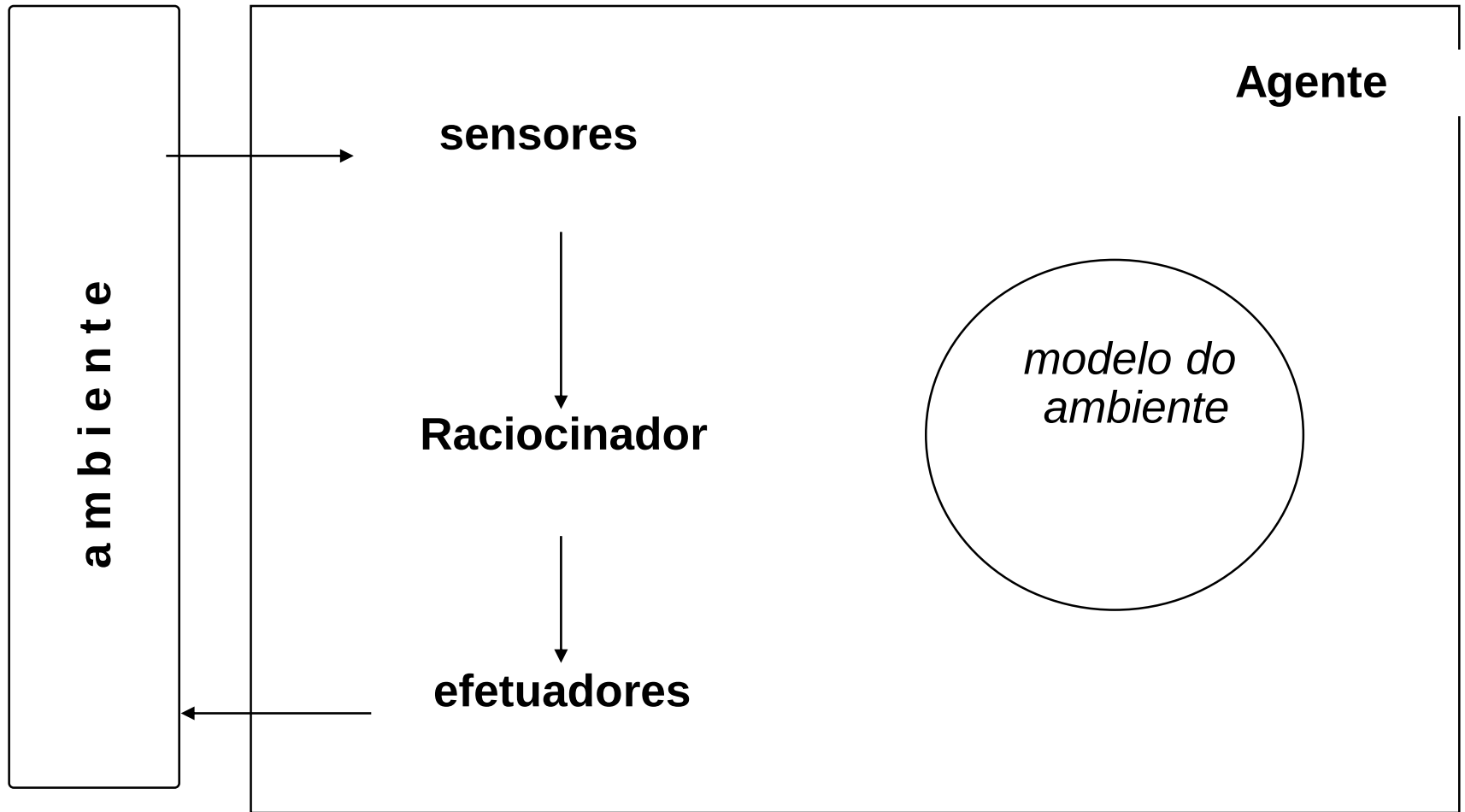


Playmodes do Servidor

ver ./plugin/soccer/soccertypes.h

- BeforeKickOff - antes de iniciar o jogo
- KickOff_**SIDE** – saída de bola para time **SIDE**
- PlayOn – Bola em jogo
- KickIn_**SIDE** – lateral para o time **SIDE**
- corner_kick_**SIDE** – escanteio para o time **SIDE**
- goal_kick_**SIDE** – tiro de meta para o time **SIDE**
- offside_**SIDE** – impedimento para o time **SIDE**
- GameOver – fim de jogo
- Goal_**SIDE** – gol para o time **SIDE**
- Free_kick_**SIDE** – falta para o time **SIDE**
- Unknown – desconhecido
- SIDE**= **Right** ou **Left**

Agente Inteligente



Formato Geral da Mensagem entre Agente e Simulador: Sensores e Efetadores

- Mensagens são strings ASCII prefixadas (cada mensagem) por um inteiro sem sinal de 32 bits (ordem big endian)
- Sensores
 - ▣ Gerais: GyroRate, HingeJoint, Accelerometer, ForceResistance
 - ▣ Futebol: Vision, GameState, Hear
- Efetadores ou Atuadores
 - ▣ Gerais: Create, HingeJoint
 - ▣ Futebol: Init, Beam, Say

Sensores Gerais

■ GyroRate – fornece informações sobre a mudança de orientação de uma parte do corpo

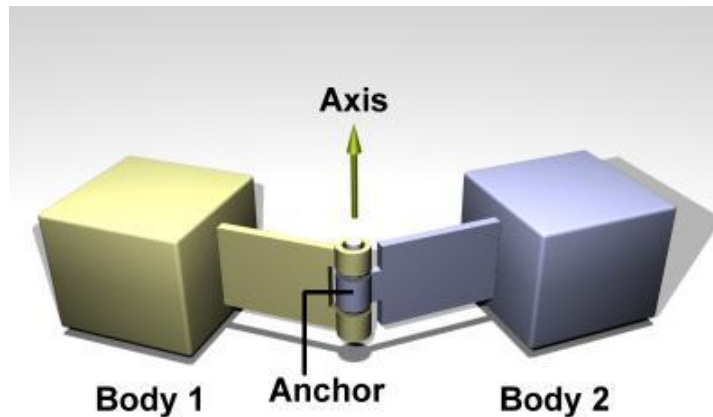
–Formato: (GYR (n <name>) (rt <x> <y> <z>))

–Ex: (GYR (n torso) (rt 0.01 0.07 0.46))

■ HingeJoint – fornece informações sobre a situação atual de uma articulação específica

–Formato: HJ (n <name>) (ax <ax>))

–Ex: (HJ (n laj3) (ax -1.02))



Articulações do NAO

Parte do Corpo	Sensor	Efetuator	Parte do Corpo	Sensor	Efetuator
Pescoço	hj1	he1	Cabeça	hj2	he2
Ombro esq.	laj1	lae1	Ombro dir.	raj1	rae1
Braço esq.	laj2	lae2	Braço dir.	raj2	rae2
Cotovelo esq.	laj3	lae3	Cotovelo dir.	raj3	rae3
Antebraço esq.	laj4	lae4	Antebraço dir.	raj4	lae4
Quadril esq. 1	llj1	lle1	Quadril dir. 1	rlj1	rle1
Quadril esq. 2	llj2	lle2	Quadril dir. 2	rlj2	rle2
Coxa esq.	llj3	lle3	Coxa dir.	rlj3	rle3
Canela esq.	llj4	lle4	Canela dir.	rlj4	rle4
Tornozelo esq.	llj5	lle5	Tornozelo dir.	rlj5	rle5
Pé esq.	llj6	lle6	Pé dir.	rlj6	rle6

Sensores Gerais

■ ForceResistance – força que atua em uma parte do corpo

◆ Formato: (FRP (n <name>) (c <px> <py> <pz>) (f <fx> <fy> <fz>))

–Ex: (FRP (n lf) (c -0.14 0.08 -0.05) (f 1.12 -0.26 13.07))

■ Accelerometer – aceleração relativa à condição de queda livre

◆ Formato: (ACC (n <name>) (a <x> <y> <z>))

◆ Ex: (ACC (n torso) (a 0.00 0.00 9.81))

Sensores Específicos do Futebol

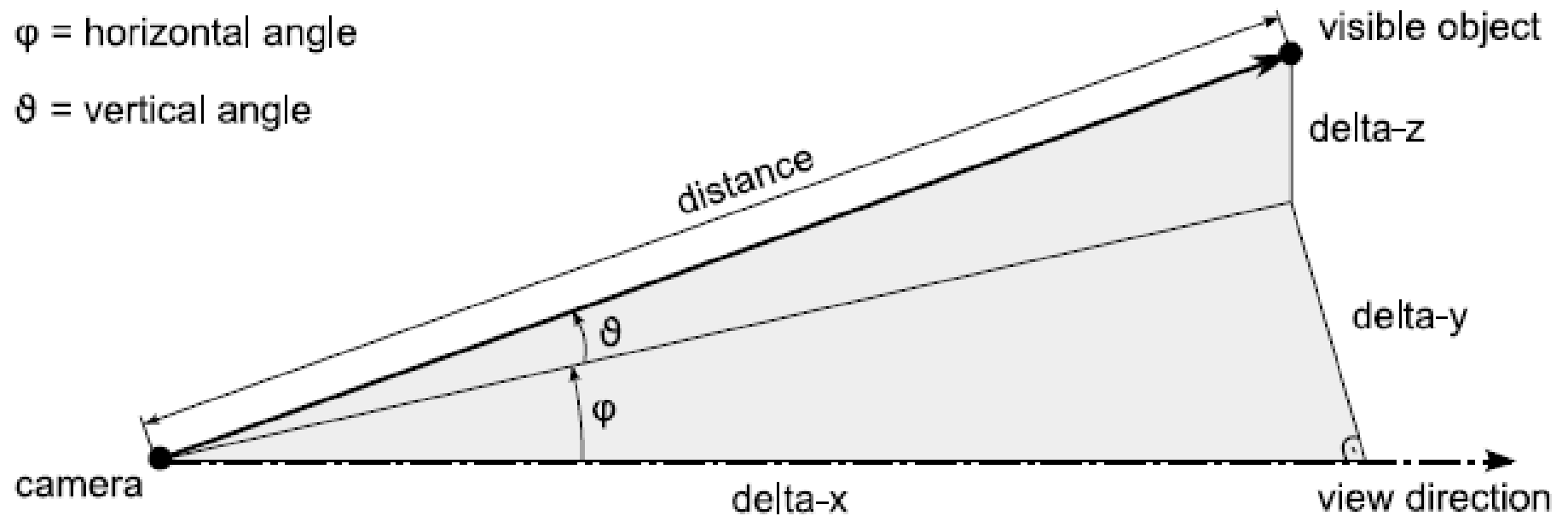
□ Visão

- Câmera embarcada com software de visão computacional inteligente acoplado
- Restricted Vision – 120 graus
- Direção da visão (pan e tilt) – pode ser mudada movendo as juntas da cabeça e pescoço
- Este sensor retorna uma lista de objetos vistos com suas distâncias, ângulo no plano horizontal, ângulo latitudinal
- Uma mensagem a cada 3 ciclos (0.06 s)

Sensor de Visão

φ = horizontal angle

θ = vertical angle



Sensor de Visão

□ Objetos Visíveis

- ▣ FLAGS (F1L, F1R, F2L, F2R)

- ▣ Postes (G1L, G1R, G2L, G2R)

- ▣ Bola (B)

- ▣ Linha (L (ponto 1) (ponto2))

- ▣ Jogadores (P seguido de (team <teamname>) (id <playerID>))

□ Formato: (See (<name> (pol <distance> <angle1> <angle2>)) +

(P (team <name>) (id <playerID>) (<body> (pol <distance> <angle1>
<angle2>)) +)

(L (pol <distance> <angle1> <angle2>) (pol <distance> <angle1>
<angle2>)) +)

Sensor de Visão - exemplo

```
(See (F1L (pol 19.11 111.69 -9.57)) (F2L (pol 16.41  
-115.88 -11.15)) (F2R (pol 45.49 -18.74 -4.00))  
(G1L (pol 9.88 139.29 -21.07)) (G2L (pol 8.40 -  
156.91 -25.00)) (B (pol 18.34 4.66 -9.90))  
(P (team Bahia3D) (id 1) (head (pol 16.98 -0.21  
3.19)) (rlowerarm (pol 16.83 -0.06 2.80)) (llowerarm  
(pol 16.86 -0.36 3.10)) (rfoot (pol 17.00 0.29 1.68))  
(lfoot (pol 16.95 -0.51 1.32))))  
(L (pol 12.11 -40.77 -2.40) (pol 12.95 -37.76 -2.41))  
(L (pol 12.97 -37.56 -2.24) (pol 13.32 -32.98 -2.20)))
```

Sensores Específicos do Futebol

■ GameState

- ◆ Informa tempo de jogo e playmode.
- ◆ Na 1ª mensagem informa placar
- ◆ Formato: (GS (t <time>) (pm <playmode>))
- ◆ Ex: (GS (t 0.00) (pm BeforeKickOff))

Sensores Específicos do Futebol

■ Hear

- ◆ Uma mensagem a cada 02 ciclos de percepção (0.04 s)
- ◆ Canais Distintos por time
- ◆ Mensagens limitadas a 20 bytes
- ◆ Distância máxima: 50 m
- ◆ Formato: (**hear** <time> (**self**|<direction>) <message>)
- ◆ Ex1: (**hear** 12.3 **self** passa)
- ◆ Ex2: (**hear** 10.5 20.5 Quem esta livre?)

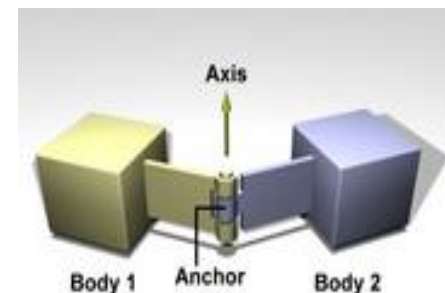
Efetadores Gerais

■ Create

- ◆ Materializa o robô no ambiente da simulação
- ◆ Formato: (`scene <filename>`)
- ◆ Ex: (`scene rsg/agent/nao/nao.rsg`)

■ HingeJoint

- ◆ Modifica o ângulo do eixo único da junta
- ◆ Formato: (`<name> <ax>`)
- ◆ Ex: (`lae3 5.3`)



Efetuadores Específicos do Futebol

■ Init

- ◆ Enviado uma vez após o Create
- ◆ Informa de que time é o jogador e o seu número
- ◆ Formato: (init (unum <playernumber>)(teamname <yourteamname>))
- ◆ Ex: (init (unum 1)(teamname Bahia3D))

■ Beam

- ◆ Move o robô para uma posição específica antes do jogo começar (BeforeKickOff)
- ◆ Formato: (beam <x> <y> <rot>)
- ◆ Ex: (beam 10.0 -10.0 0.0)

Efetuadores Específicos do Futebol

■ Say

- ◆ Envia mensagens para outros agentes
- ◆ Formato: (say <message>)
- ◆ Ex: (say passa)

Ativando a geração de logs

- Edite `simspark.rb`

- Substitua `$recordLogfile = false` por `$recordLogfile = true`

- Execute o simulador

- Após a partida o log estará no arquivo `sparkmonitor.log`

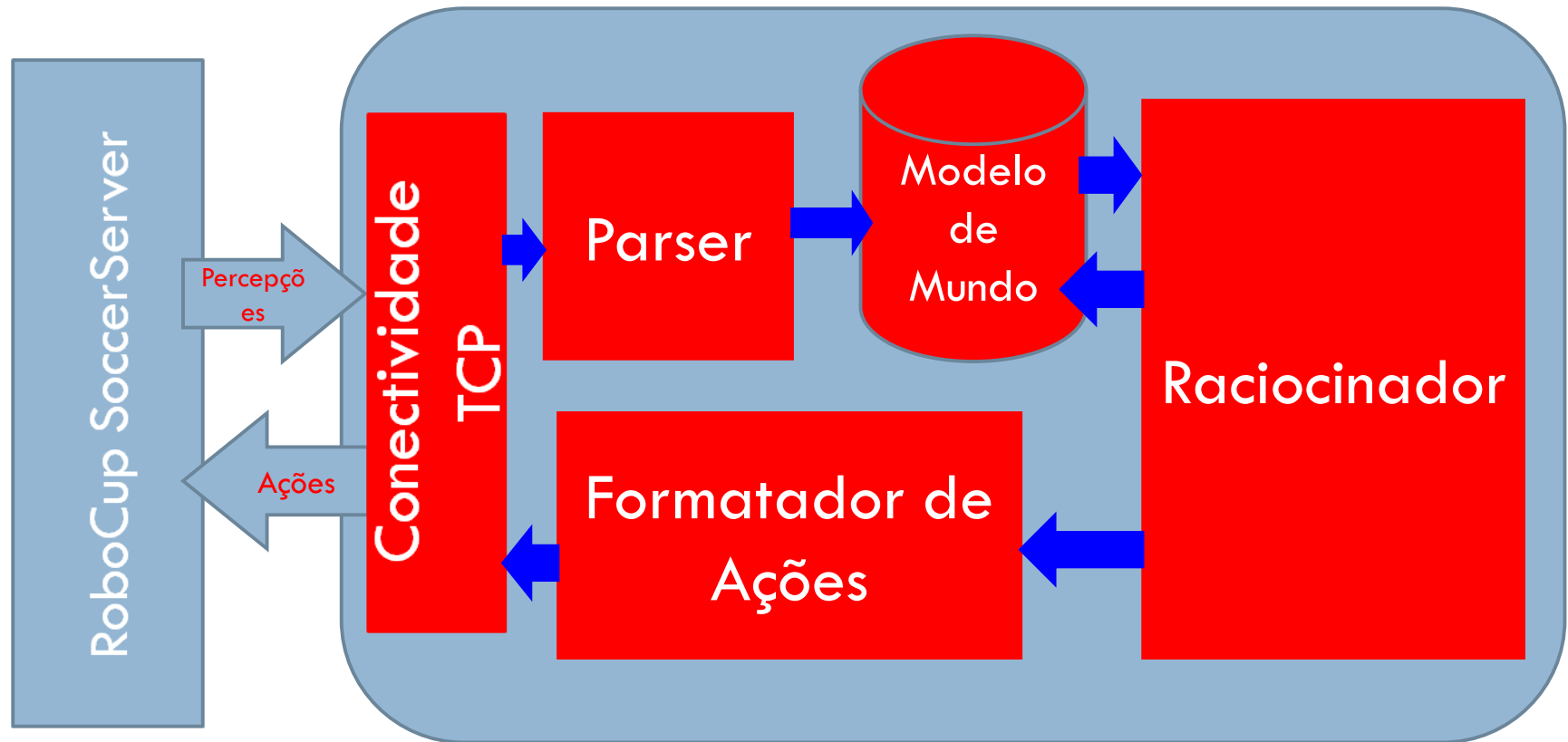
- Reproduzindo Logs

 - ◆ `./roboviz.sh --logfile /usr/bahia3d/logfiles/sparkmonitor.log`

 - ◆ `-Rcssmonitor3d --logfile /usr/bahia3d/logfiles/sparkmonitor.log`

PARTE III: Planejando
Movimentos em robôs
Humanoides Simulados

Estrutura mínima de um agente Jogador



Abordagens para Construir um Time 3D

- ▣ Construção Total

- ▣ Modelar movimentos primitivos pré-definidos a partir do movimento das juntas: levantar, chutar, etc.

- ▣ Uso de Time Base

- ▣ Nexus3D 2010

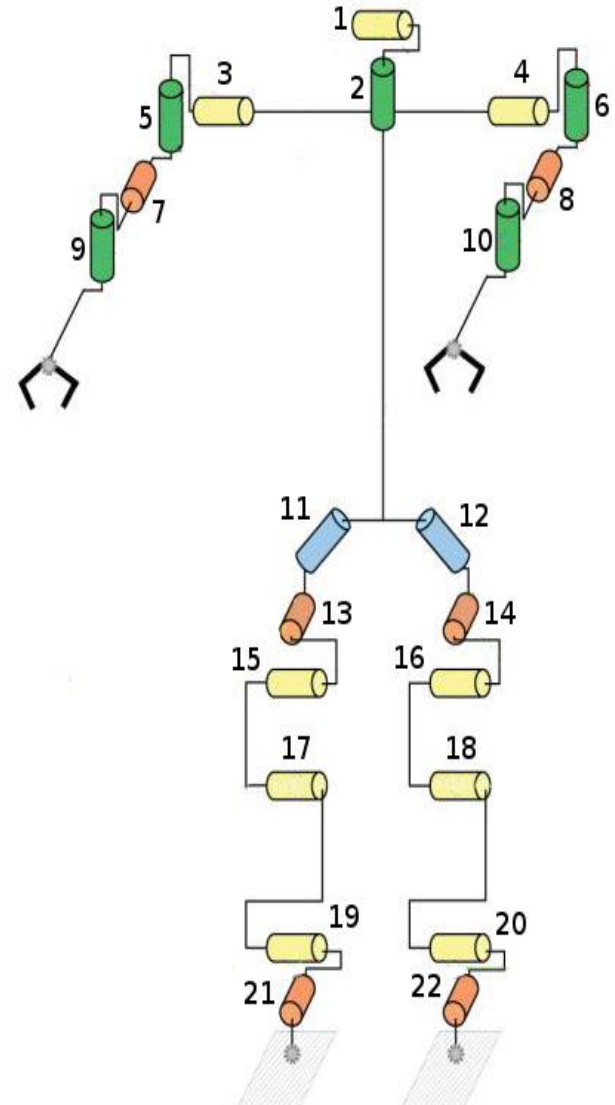
- ▣ Delta3D 2010

- ▣ MagmaOffenburg 2011- <http://robocup.fh-offenburg.de>

- ▣ Conectividade TCP, movimentos básicos, parser, modelo de mundo básico e formatador de ações prontos

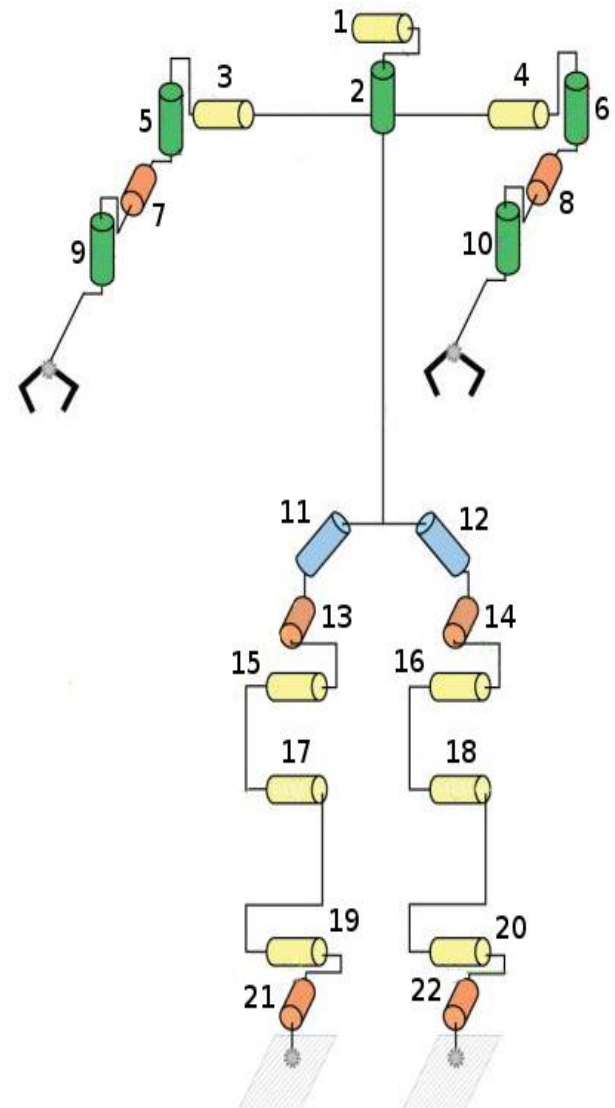
Articulações do NAO

- 1 –hj2 cabeça vert.
- 2 – hj1 cabeça hor.
- 3 / 4 – (r/l)aj1 braço cima-baixo
- 5 / 6 – (r/l)aj2 braço esq-dir
- 7 / 8 – (r/l)aj3 gira o braço
- 9 / 10 – (r/l)aj4 dobra-estica cotovelo
- 11 / 12 – (r/l)lj1 gira a perna
- 13 / 14 – (r/l)lj2 abre-fecha pernas
- 15 / 16 – (r/l)lj3 perna frente-trás
- 17 / 18 – (r/l)lj4 dobra-estica joelho
- 19 / 20 – (r/l)lj5 pé cima-baixo
- 21 / 22 – (r/l)lj6 pé esq-dir



Limites em graus

- 1 – hj2 -45 a 45
- 2 – hj1 -120 a 120
- 3 / 4 – (r/l)aj1 -120 a 120
- 5 / 6 – (r/l)aj2 -95 a 1(r) / -1 a 95(l)
- 7 / 8 – (r/l)aj3 -120 a 120
- 9 / 10 – (r/l)aj4 -1 a 90(r) / -90 a 1(l)
- 11 / 12 – (r/l)lj1 -90 a 1
- 13 / 14 – (r/l)lj2 -45 a 25(r) / -25 a 45(l)
- 15 / 16 – (r/l)lj3 -25 a 100
- 17 / 18 – (r/l)lj4 -130 a 1
- 19 / 20 – (r/l)lj5 -45 a 75
- 21 / 22 – (r/l)lj6 -25 a 45(r) / -45 a 25(l)



IFAS3D

□ InterFace Analítica do Simulador 3D

FileServerScript

Server Status

Cycle Server: 14.06

Game Status: BeforeKickOff

Game Time: 0

Internal Perceptor

Accelerometer:

0-0.289.81

GyroRate:

000

FRP

Right Foot

C: 0.01-0-0.02

F: 000

Left Foot

C: 0.010-0.01

F: 000

Beam

X0Y0Angle0

BEAM

Current Position Robot

X: 5.22712Y: 0Angle: 0

Joint

Step Counter: 0

	Current Position	Speed	Duration	T.Initial	Ang.Init
hj1:	0	0	0	0	0
laj1:	-0	0	0	0	0
laj2:	-0	0	0	0	0
laj3:	0	0	0	0	0
laj4:	-0	0	0	0	0
llj1:	-0	0	0	0	0
llj2:	0	0	0	0	0
llj3:	1.67	0	0	0	0
llj4:	0	0	0	0	0
llj5:	0	0	0	0	0
llj6:	0	0	0	0	0

	Current Position	Speed	Duration	T.Initial	Ang.Init
hj2:	0	0	0	0	0
raj1:	-0	0	0	0	0
raj2:	0	0	0	0	0
raj3:	0	0	0	0	0
raj4:	0	0	0	0	0
rlj1:	-0	0	0	0	0
rlj2:	-0	0	0	0	0
rlj3:	1.67	0	0	0	0
rlj4:	0	0	0	0	0
rlj5:	0	0	0	0	0
rlj6:	-0	0	0	0	0

START

STOP

RESET

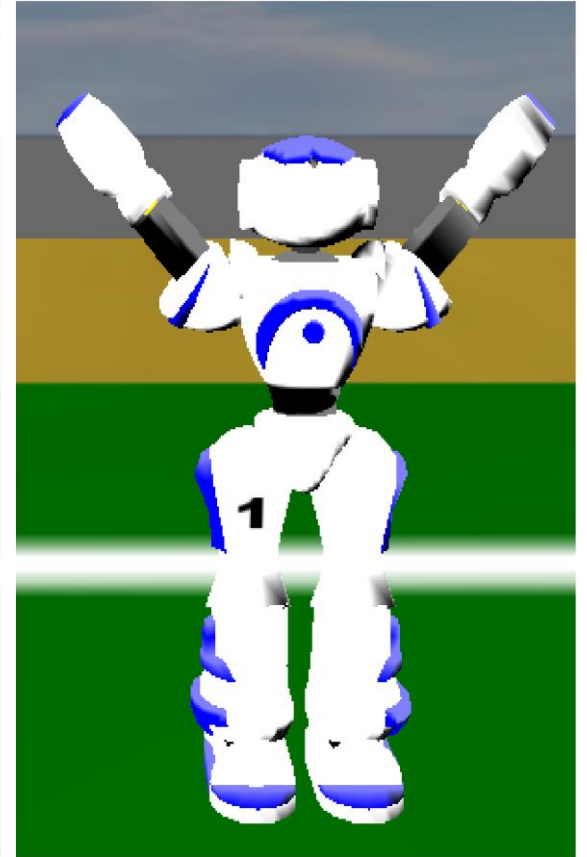
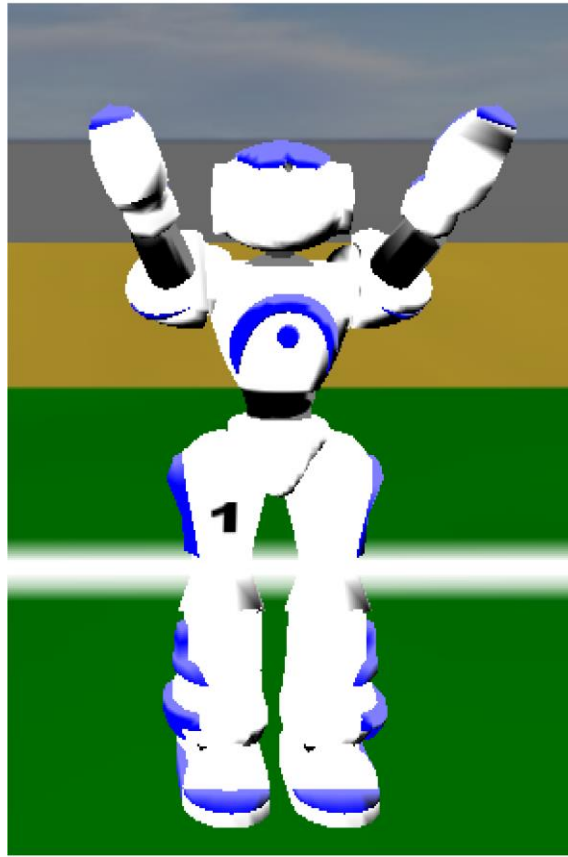
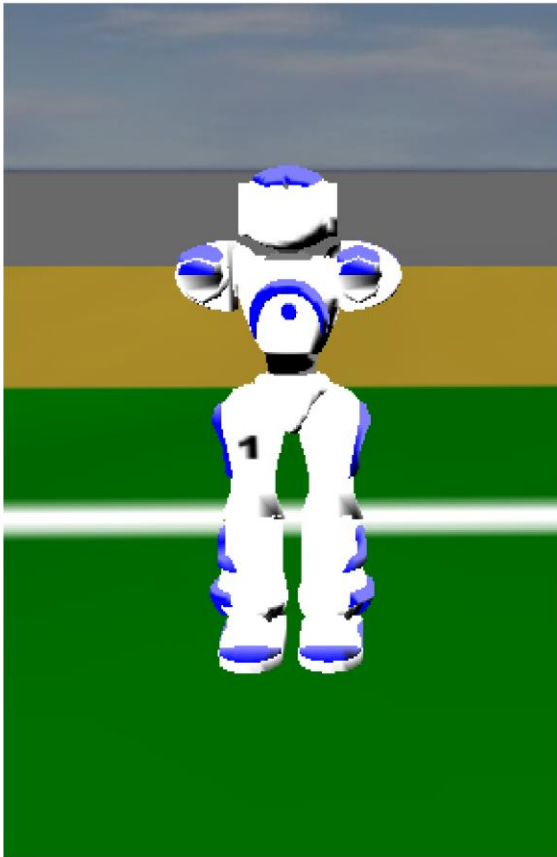
Movimentos

□Ex.A: Levantar o braço esquerdo



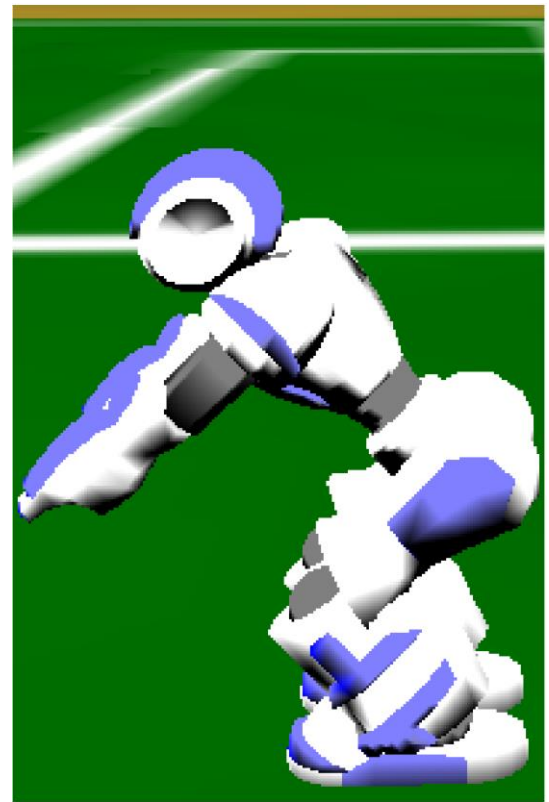
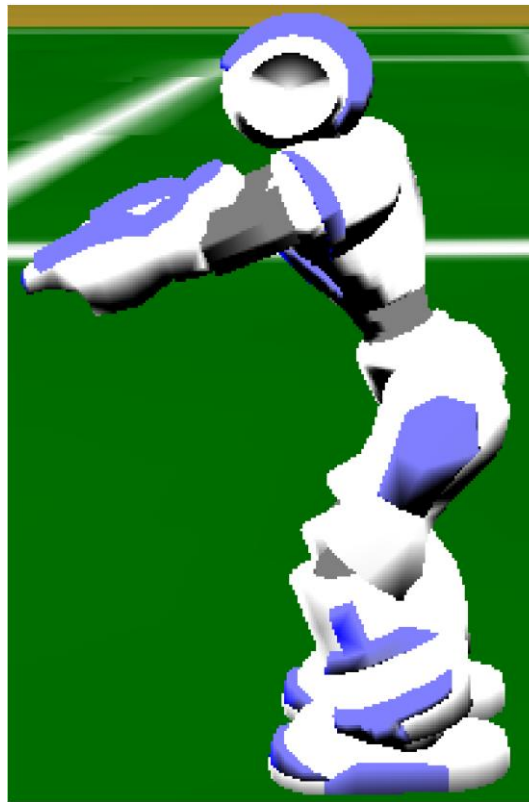
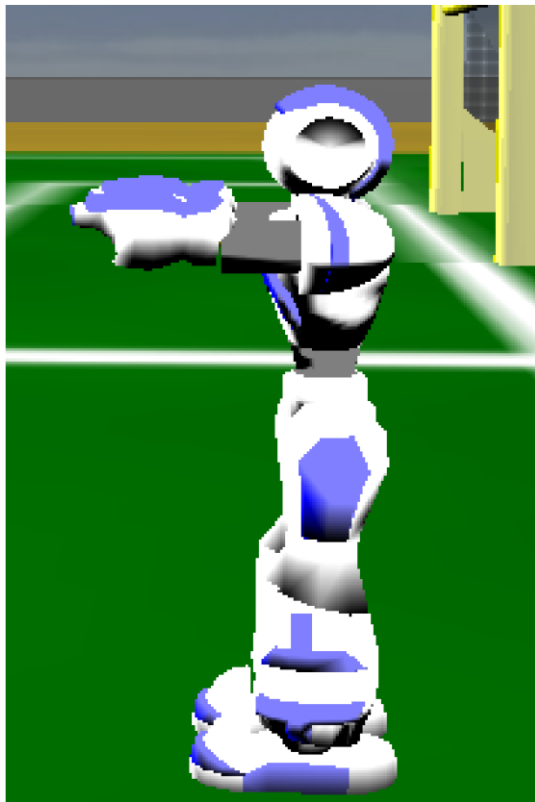
Movimentos

□Ex.B: Levantar os braços abertos



Movimentos

□Ex.C: Agachar



Movimentos

□ Tabela ângulos x juntas x velocidades

<i>Relação Juntas, Ângulos e Velocidades</i>			
Movimento	Juntas	Posição Final(graus)	Velocidade(rad/s)
A	4	90	1.57
B	3	90	1.57
	4	90	1.57
	5	-45	-0.785
	6	45	0.785
C	15	90	1.57
	16	90	1.57
	17	-90	-1.57
	18	-90	-1.57
	19	45	0.785
	20	45	0.785

*** Considerando o tempo de duração de 50 ciclos (1 segundo)**

Scripts

□ Script para o movimento B:

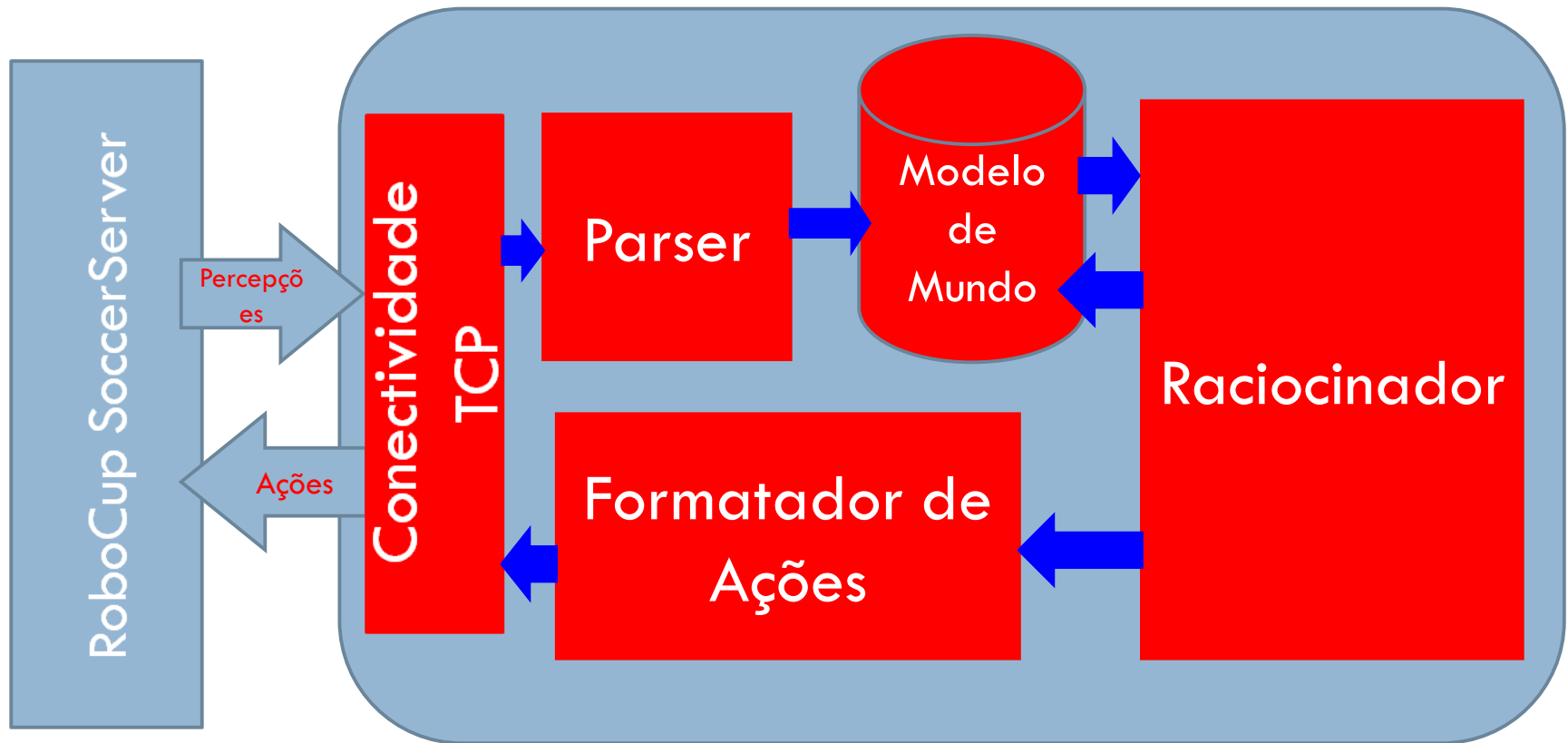
```
<script>
  <step>
    <duration>50</duration>
    <joint>
      <laj1>1.57</laj1>
      <raj1>1.57</raj1>
    </joint>
  </step>
  <step>
    <duration>50</duration>
    <joint>
      <laj2>0.785</laj2>
      <raj2>-0.785</raj2>
    </joint>
  </step>
</script>
```



PARTE IV: Programando a IA a partir de um time base



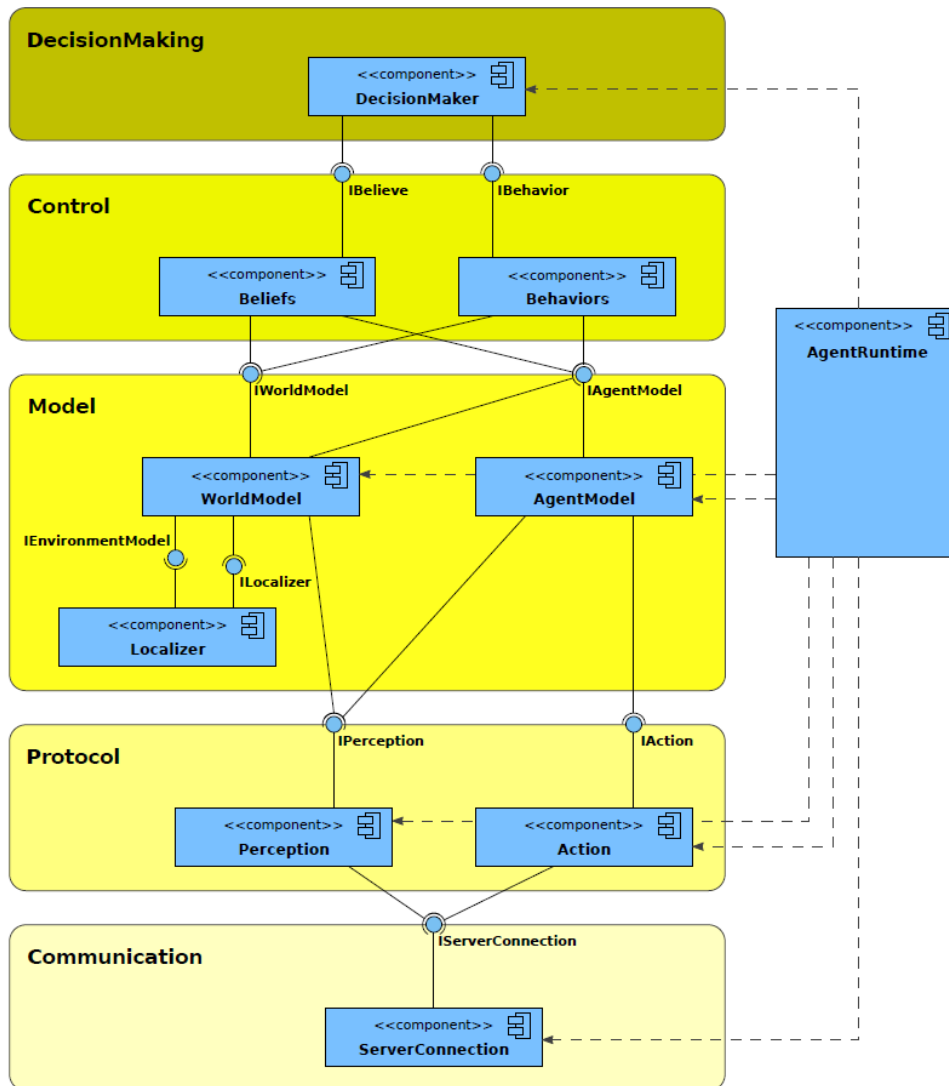
Times Base



Magma Offenburg 2011

- University of Applied science Offenburg, Prof. Klaus Dorer
- Código em Java
- Respeita as boas práticas de Engenharia de Software
- Framework magma-AF => serve de base para construir um agente jogador de futebol
- Equipe disputa sempre as primeiras posições nas competições mundiais e nacionais

Framework magma-AF



AgentRunTime

1. Parse mensagem recebida (Perception)
2. Atualizar estado interno (AgentModel)
3. Atualizar estado do ambiente (WorldModel)
4. Decidir o próximo comportamento (DecisionMaker)
5. Mapear as ações executadas do AgentModel em ações concretas (AgentModel)
6. Codificar e enviar mensagem para o servidor (Action)

Estrutura do Projeto



- Eclipse
- Src – código-fonte
- Config – arquivos de configuração
- srcTest – código-fonte dos testes
- Bin – scripts gerais de início (start)
- Lib – bibliotecas de tempo de execução e teste
- Log – arquivos de log são gravados aqui

Estrutura de Pacotes Java

magma.agent – componentes independentes de robô

- general – componente de execução
- meta – meta modelo do ambiente e do robô
- connection – estabelece conexão com o servidor
- perception – parser das mensagens que chegam
- action – codifica mensagens enviadas ao servidor
- worldmodel – informação que o agente tem sobre o ambiente
- Belief – conjunto (fuzzy) de crenças indicando situações
- Behavior – conjunto de comportamentos formados por diversas ações
- Decision – decide e executa um comportamento

Estrutura de Pacotes Java



magma.robot – componentes e definições específicas de cada robô => corresponde ao magma.agent quando necessário

magma.util – utilitários utilizados em todo o código como cálculos geométricos, padrões de projeto, etc.

Classe Main

magma.robot.RoboCupClient

- Nome do time
- Player ID
- Host
- Port
- Log Level (fine, finer, finest, warn, **severe**, info)
- Server Version (64)
- Decision Maker ID (0)
 - 0: decisão baseada no Player ID (1->goleiro, outro-> jogador de linha)
 - 1: goleiro
 - 2: jogador de linha
 - 3: tomador de decisão simples (para testes)
 - 4: cobrança de pênaltis
 - 5: faz nada

Executando um Agente



Eclipse

- Exemplo: RoboCupClient (9) 2011.launch

Console

- Scripts em bin

Junit Tests

- AllMagmaTests.launch

Desafios para a Liga 3D



- Drop-in Challenge
- Run Challenge

Referências Oficiais

- ▣ Site Oficial da Liga de Simulação
 - ▣ http://wiki.robocup.org/wiki/Soccer_Simulation_League
- ▣ Lista de e-mail RoboCup 3D mundial
 - ▣ <https://lists.sourceforge.net/lists/listinfo/sserver-three-d>
- ▣ Documentação do RCSSServer3D
 - ▣ <http://simspark.sourceforge.net/wiki>
- ▣ Lista de E-mails RoboCup Brasil 3D
 - ▣ cbr-3d@googlegroups.com -
<https://groups.google.com/forum/?fromgroups#!forum/cbr-3d>

Sites e Contatos

- msimoes@uneb.br
- <http://www.acso.uneb.br/brt>



Prof. Marco Simões