

RoboCup 3D

Abril 27, 2006

FCPortugal team



R😊bocup 3D

☒ Liga RoboCup

☒ Simulador

☒ FCPagent 3D

Robocup

- O Robocup é uma competição a nível mundial que se desenrola todos os anos
- Visa o estudo e desenvolvimento na IA e Robótica, em particular dos Sistemas Multi-agente.

Robocup

- A primeira edição decorreu em 1997 em Nagoya no Japão
- RoboCup 2004 – Lisboa
- RoboCup 2005 – Osaka
- RoboCup 2006 - Bremen



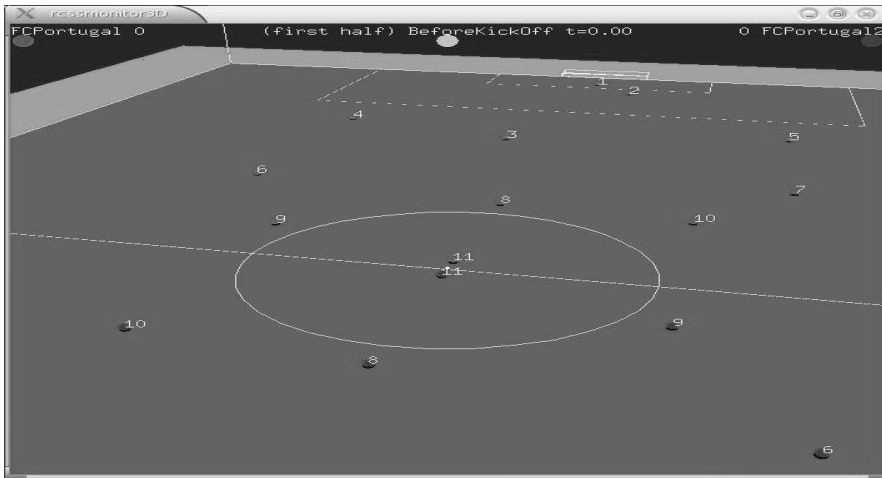
Robocup

- ❏ As competições dividem-se em 4 grupos:
- ❏ RobocupSoccer
- ❏ RobocupRescue
- ❏ RobocupJunior
- ❏ Robocup@home ~ new

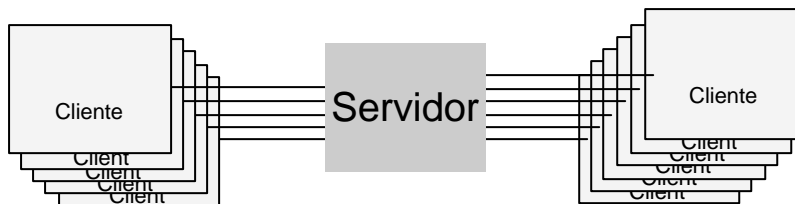
Robocup

- ❏ RobocupSoccer – O futebol foi a grande motivação.
- ❏ RobocupRescue – A busca e salvamento em cenários de catástrofe
- ❏ RobocupJunior – Robocup nas escolas
- ❏ Robocup@Home - O alvo é promover o desenvolvimento das aplicações robóticas que possam ajudar o ser humano na vida diária.

Simulador Robocup 3D

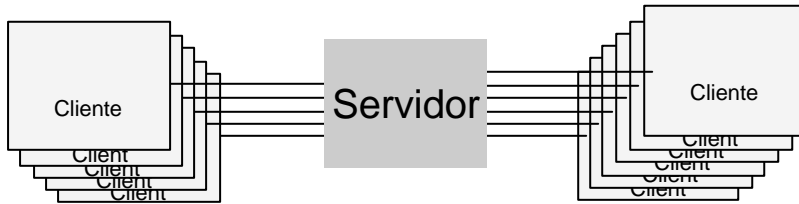


Clientes & Servidor

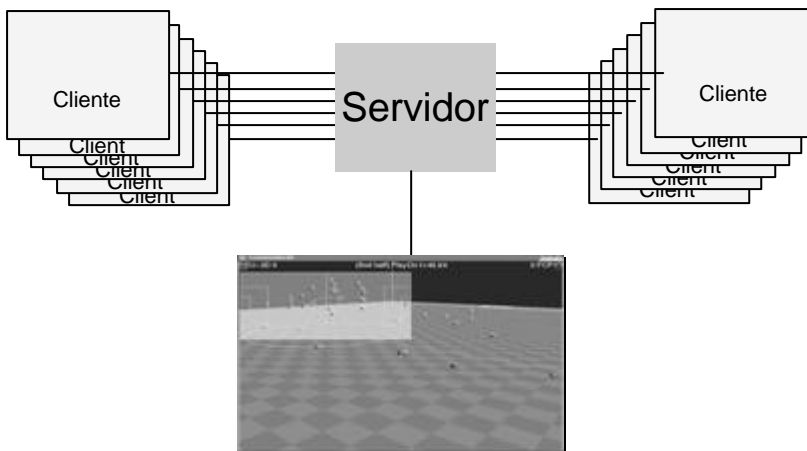


- Um servidor
- 11 agentes (clientes) por equipa
- Comunicação Cliente/Servidor via UDP/IP

Clientes & Servidor



☒ Todas as comunicações são efectuadas através do servidor!



☒ O visualizador pode ser usado durante um jogo ou para ler posteriormente os logs gerados.

☒ OpenGL

❏ O DebugMonitor

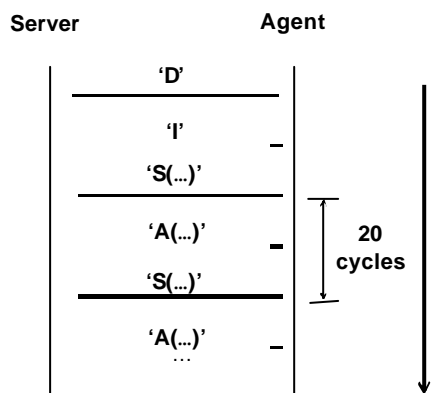


Simulador

O que é preciso?

- ❏ SPADES - pacote de software que permite a criação de um sistema de simulação Multi-agente.
- ❏ RCSSSERVER3D – Servidor.
- ❏ EXPAT – Manipulação de ficheiros XML.
- ❏ BOOST – Livrarias.
- ❏ ODE - Conjunto de livrarias para a simulação da dinâmica dos corpos rígidos.

Comunicação Servidor / Cliente



Perceptors

- Mecanismo que permite ao agente conhecer o estado do mundo
- Tipos de perceptors:
 - ☞ Vision
 - ☞ GameState
 - ☞ AgentState
 - ☞ Hear

Vision Perceptor

- Lista os objectos dentro do campo de visão do agente (marcadores, balizas, bola, jogadores)
- Visão omnidireccional
- regular visionperceptor (360°) ou RestrictedVisionPerceptor (90°) na versão 0.5
- Objectos podem esconder outros objectos (ainda não implementado)
- Erros de calibração e de distância

Vision Perceptor

•Formato da Mensagem:

•Sint int (Vision

(Flag (id <id>) (pol <distância> <ângulo horizontal> < ângulo vertical>)) ...

(Goal (id <id>) (pol <distância> <ângulo horizontal> < ângulo vertical>)) ...

(Ball (pol <distância> <ângulo horizontal> < ângulo vertical>)) ...

(Player (team string) (id <id>) (pol <distância> <ângulo horizontal> < ângulo vertical>)) ...

)

Vision Perceptor

●Exemplo:

S760 770 (Vision (Flag (id 2_l) (pol 70.228 44.7407 0.139703)) (Flag (id 2_r) (pol 70.7899 -53.7605 0.194215)) (Goal (id 1_l) (pol 53.5517 79.0481 0.295366)) (Goal (id 2_l) (pol 54.8914 71.3933 0.26277)) (Goal (id 1_r) (pol 54.2 -87.4828 0.181149)) (Goal (id 2_r) (pol 55.4752 -80.1392 0.0430096)) (Ball (pol 10.0897 -5.56432 0.0256475)) (Player (team FCP3D) (id 1) (pol 48.5214 74.1505 0.414638)) (Player (team FCP3D) (id 2) (pol 23.756 50.2793 0.449999)) (Player (team ZAR) (id 3) (pol 24.6399 -60.1059 0.272185)) (Player (team ZAR) (id 4) (pol 29.7736 -47.3717 0.335875)) (Player (team ZAR) (id 11) (pol 9.95938 -3.07117 0.703956))...)

Gamestate Perceptor

- ❖ Descreve estado do jogo e tempo de simulação
- ❖ O primeiro GP enviado descreve algumas das variáveis do jogo
- ❖ Estados possíveis são:
"BeforeKickOff", "KickOff_Left", "KickOff_Right", "PlayOn", "KickIn_Left", "KickIn_Right", "corner_kick_left", "corner_kick_right", "goal_kick_left", "goal_kick_right", "offside_left", "offside_right", "GameOver", "Goal_Left", "Goal_Right", "free_kick_left", "free_kick_right", "unknown".

Gamestate Perceptor

- **Formato do primeiro GP:**
- (GameState (unum int) (team string) (FieldLength float) (FieldWidth float) (FieldHeight float) (GoalWidth float) (GoalDepth float) (GoalHeight float) (BorderSize float) (AgentMass float) (AgentRadius float) (AgentMaxSpeed float) (BallRadius float) (BallMass float) (time float) (playmode string))
- **Formato dos outros GP:**
- (GameState (time float) (playmode string))

Gamestate Perceptor

- Exemplos:
(GameState (unum 10) (team right) (FieldLength 107) (FieldWidth 72) (FieldHeight 40) (GoalWidth 7.32) (GoalDepth 2) (GoalHeight 0.5) (BorderSize 10) (AgentMass 75) (AgentRadius 0.22) (AgentMaxSpeed 10) (BallRadius 0.111) (BallMass 0.437928) (time 0) (playmode BeforeKickOff))
- (GameState (time 2.06) (playmode PlayOn))

Agentstate Perceptor

- Informa o jogador acerca do seu estado:
- Ângulos da câmera
- Temperatura (ainda não utilizado)
- Nível da bateria

Agentstate Perceptor

- **Formato da mensagem:**
- (AgentState
(pan_tilt <pan in degrees> <tilt in degrees>)
(battery <battery level in percent>)
(temp <temperature in degree>)
)
- **Exemplo:**
- (AgentState (pan_tilt -100 0) (battery 99.8056) (temp 23))

Hear Perceptor

- Enviado quando um jogador envia um SayEffector (diz alguma coisa):
- Tamanho máximo da mensagem 512 caracteres
- Alcance máximo de 50m
- Cada jogador só pode ouvir uma mensagem de cada equipa a cada dois ciclos
- Se mais de uma mensagem for enviada o servidor escolhe uma aleatoriamente
- Canais diferentes para cada equipa

Hear Perceptor

- **Formato:**
- (hear <time> <direction in degree> <message>)
- **Exemplos:**
- (hear 0.8 -179.99 Mensagem1)
- (hear 0.4 self Mensagem2)

Effectors

- Mecanismo que permite ao agente interagir com o mundo

- Tipos de effectors:

- Create
- Beam
- Drive
- Kick
- Init
- Catch
- Say
- Pantilt

Create Effector

- Primeira effector a ser enviado
- Selecciona e cria o tipo de robot seleccionado
- Para já só há um tipo fixo

- Formato: (create)

Init Effector

- ❑ Selecciona a equipa e o número do agente
- ❑ Antes de ser executado os effectors seguintes não funcionam correctamente
- ❑ **Formato:**
 - ❑ A(init (unum <number>) (teamname <string>))
- ❑ **Exemplo:**
 - ❑ A(init (unum 2) (teamname FCP3D))

Beam Effector

- ❑ Move instantaneamente o agente para a posição desejada sem erro
- ❑ Só pode ser efectuado quando o estado do jogo é “beforekickoff”
- ❑
 - ❑ **Formato:** A(beam <x> <y> <z>)
 - ❑ **Exemplo:** A(beam -1 10.0246 0.0)

Drive Effector

- Permite ao agente posicionar-se durante o jogo através da aplicação de uma força (máximo 100)
- Força aplicada em todos os ciclos seguintes até ser executado outro Driveeffector
- Consome bateria e necessita que nesta não esteja vazia
- Pode ser utilizado para empurrar a bola ou outro jogador
- Só pode ser utilizado se o jogador estiver no chão

Drive Effector

- Formato:
A (drive <x> <y> <z>)
- Exemplo:
A(drive 99.8532 5.41651 0)
- A(drive 0 0 0) - Para parar

Kick Effector

- Permite ao agente chutar a bola com a potência e ângulo desejado
- Força aplicada durante 10 ciclos
- Potência entre 0 e 100% da potência máxima
- Ângulo entre 0° e 50°
- Bola só pode ser chutada se estiver a menos de 0,04 metros
- Só pode ser utilizado se o jogador estiver no chão

Kick Effector

- Formato:
A(kick <angle> <power>)
- Exemplo:
A(kick 20.0 80.0)

Catch Effector

- ❑ Só o jogador 1 pode executá-lo (guarda-redes)
- ❑ A bola tem que estar na grande área, perto do agente (a menos de “catch margin” – actualmente 2 metros) e o estado de jogo ser “playon”
- ❑ Depois do guarda-redes apanhar a bola todos os jogadores a menos de 5 metros são afastados e esta é colocada no chão à frente do guarda-redes

Formato: A(catch)

Say Effector

- ❑ Envia as mensagens para os outros jogadores da equipa
- ❑ As mensagens tem um alcance de 50m
- ❑ Comprimento máximo de 512 caracteres

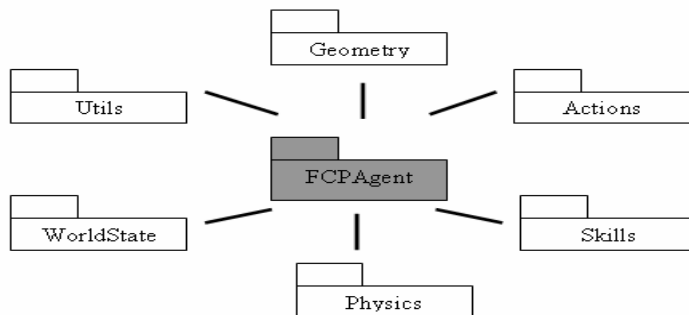
❑ **Formato:** (say Mensagem)

PanTilt Effector

- ❑ Muda a direcção da câmara do agente (rotação do pescoço)
- ❑ Tem erros associado
- ❑ Formato: (pantilt <panDelta> <tiltDelta>)

FCPAgent

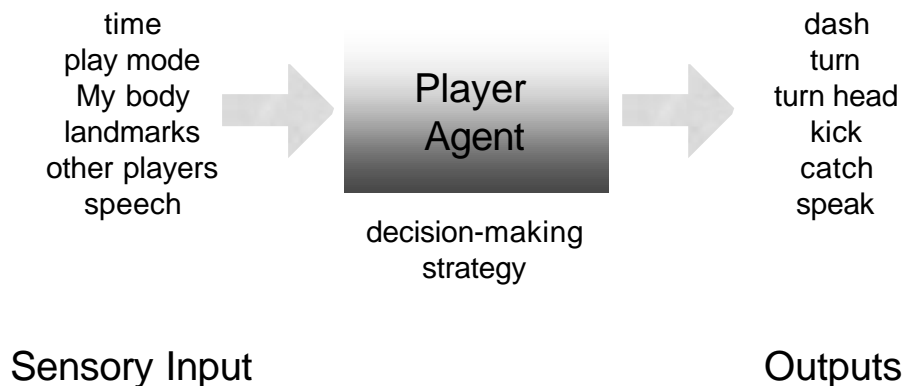
- ❑ O FCPAgent é composto de 6 módulos



FCPagent

- WorldState — Apresenta todas as informações à cerca do estado do mundo.
- Physics — Simula as interações entre os objectos
- Geometry — Implementa as classes usadas na simulação do mundo 3D
- Skills - Apresenta as acções de baixo nível que um agente pode executar: Chutar, mover o corpo,...
- Actions — Conjunto de Skills que em conjunto produzem acções de alto nível
- Utilis — Criar logfiles, comunicação com o servidor, etc

FCPagent



FCPagent

WORLD STATE

- ☒ O mais complexo
- ☒ Contem todas as informações que um agente precisa antes de agir.

FCPagent

PHYSICS

- ☒ Reproduz as acções físicas entre os corpos e o mundo envolvente.
- ☒ Podemos estimar a velocidade e aceleração aplicados sobre os corpos.

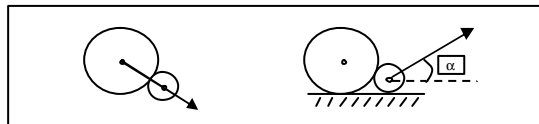
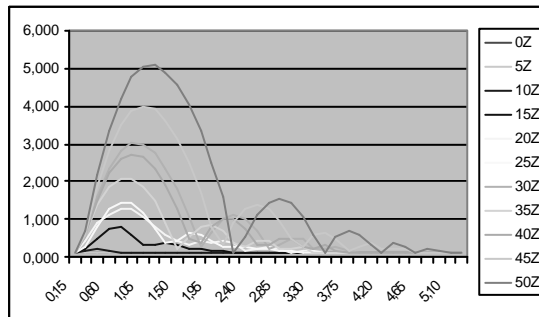
FCPagent

SKILLS

- KICK (Ang, Pot)

Permite ao agente chutar a bola se este estiver à *Kickable distance*.

Pot [0, 100]
Ang [0°, 50°]



References

- Todas as informações sobre o Robocup podem ser encontradas no RoboCup Soccer Server manual.
— <http://sourceforge.net/projects/sserver>