

JULIO CESAR MAFRA

**PROTÓTIPO DE FORMAÇÃO EM TIMES
DE FUTEBOL DE ROBÔS UTILIZANDO
ROBÓTICA BASEADA EM
COMPORTAMENTO**

Prof. Jomi Fred Hübner - Orientador



ROTEIRO DA APRESENTAÇÃO

- ➔ Introdução
- ➔ Objetivos
- ➔ Fundamentação Teórica
- ➔ Desenvolvimento
- ➔ Conclusão

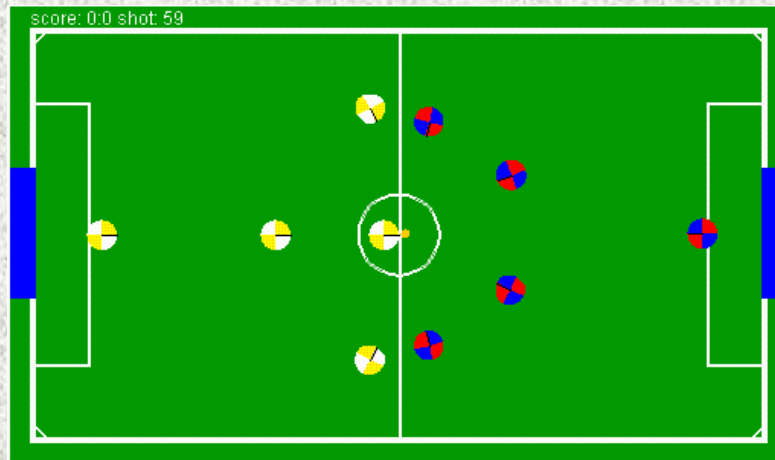
INTRODUÇÃO

➡ RoboCup

➡ Estratégia de Jogo

OBJETIVO

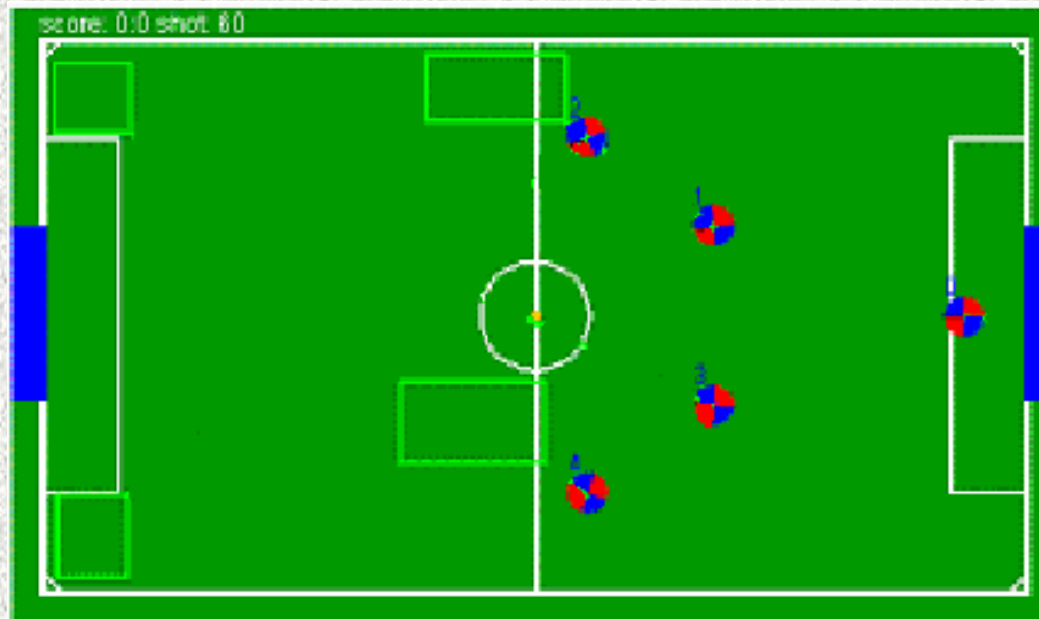
- ➔ Desenvolver protótipo de formação em times de futebol de robôs, que jogam futebol em um ambiente simulado de duas dimensões (2D), utilizando a robótica baseada em comportamento



OBJETIVO

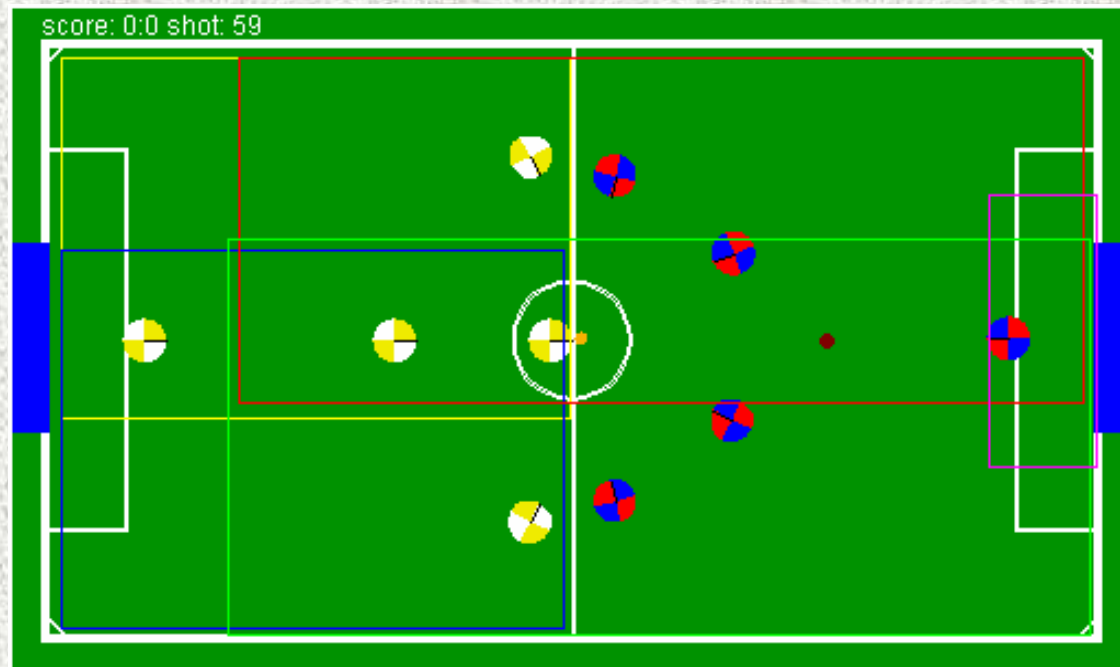
Objetivos Específicos

- ➔ Determinar a área de atuação de um agente robô jogador de futebol através do desenvolvimento de classe java



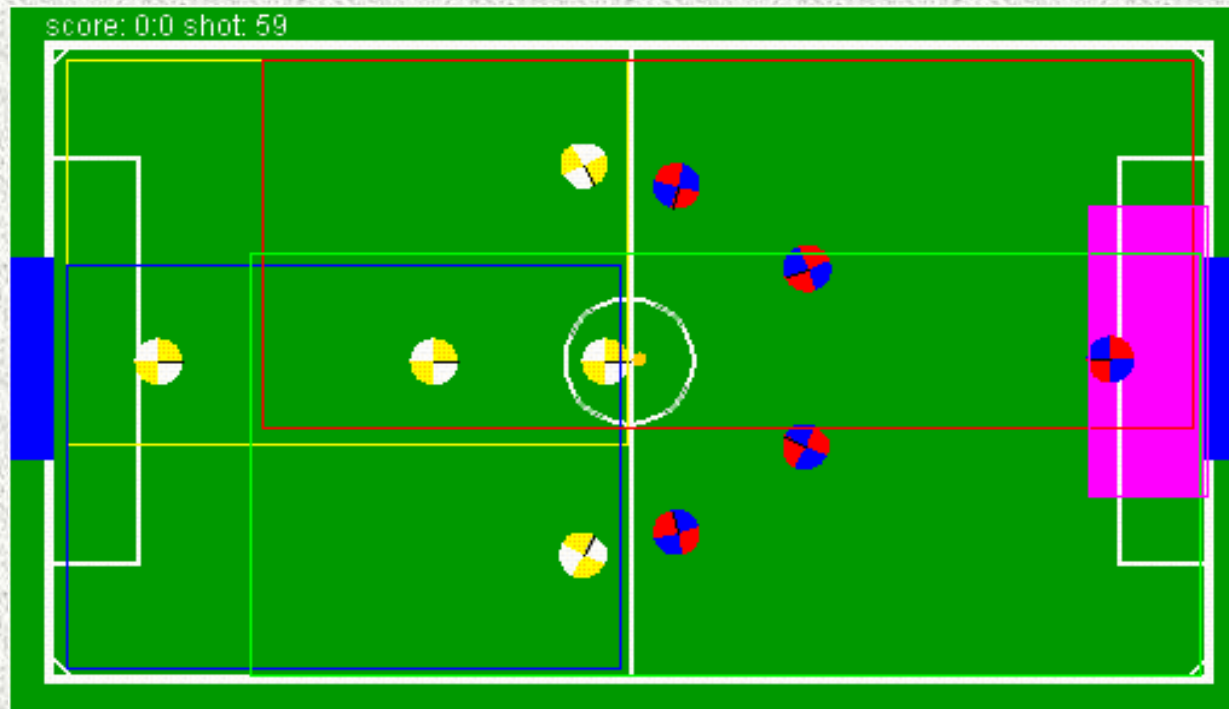
OBJETIVO

- ➔ Determinar a área de atuação com base na função de um agente robô jogador de futebol



OBJETIVO

- ➔ Desenvolver agentes robôs jogadores de futebol com comportamentos distintos (goleiro, atacante, lateral)



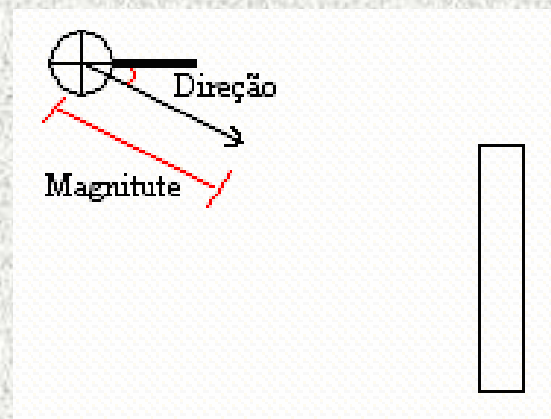
FUNDAMENTAÇÃO TEÓRICA

- ➔ Campo Potencial
- ➔ Arquitetura de Esquemas
- ➔ TeamBots Simulator(TBSim)

FUNDAMENTAÇÃO TEÓRICA

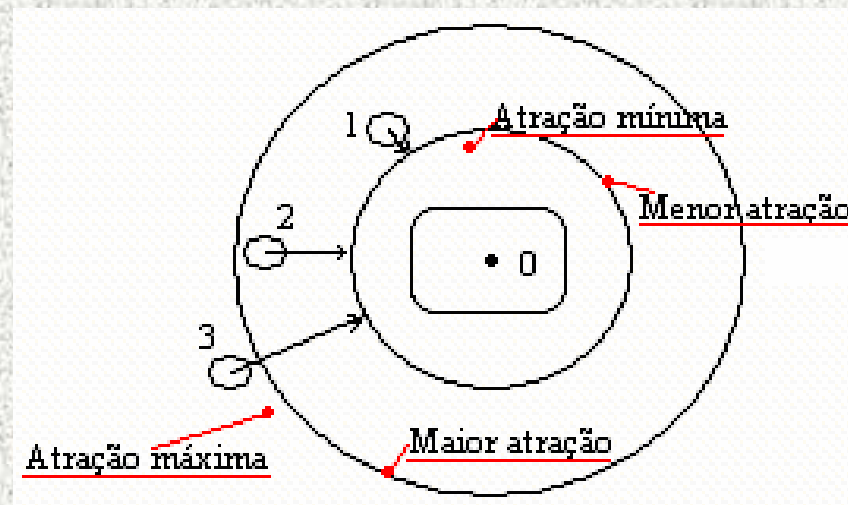
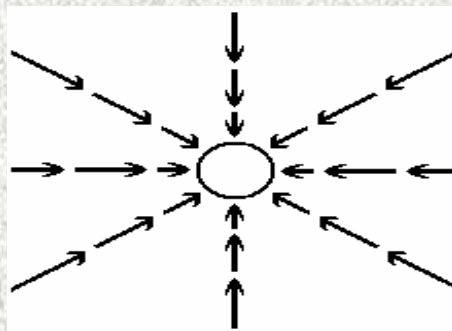
Campo Potencial

- ➔ Trajetórias sem obstáculos
- ➔ Força potencial



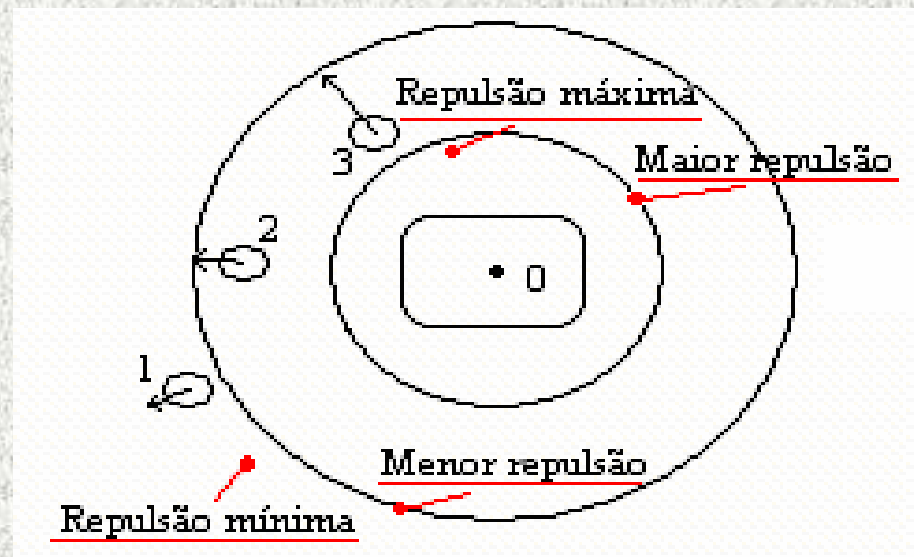
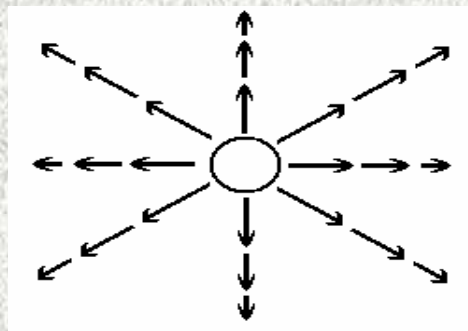
FUNDAMENTAÇÃO TEÓRICA

Campo de Atração



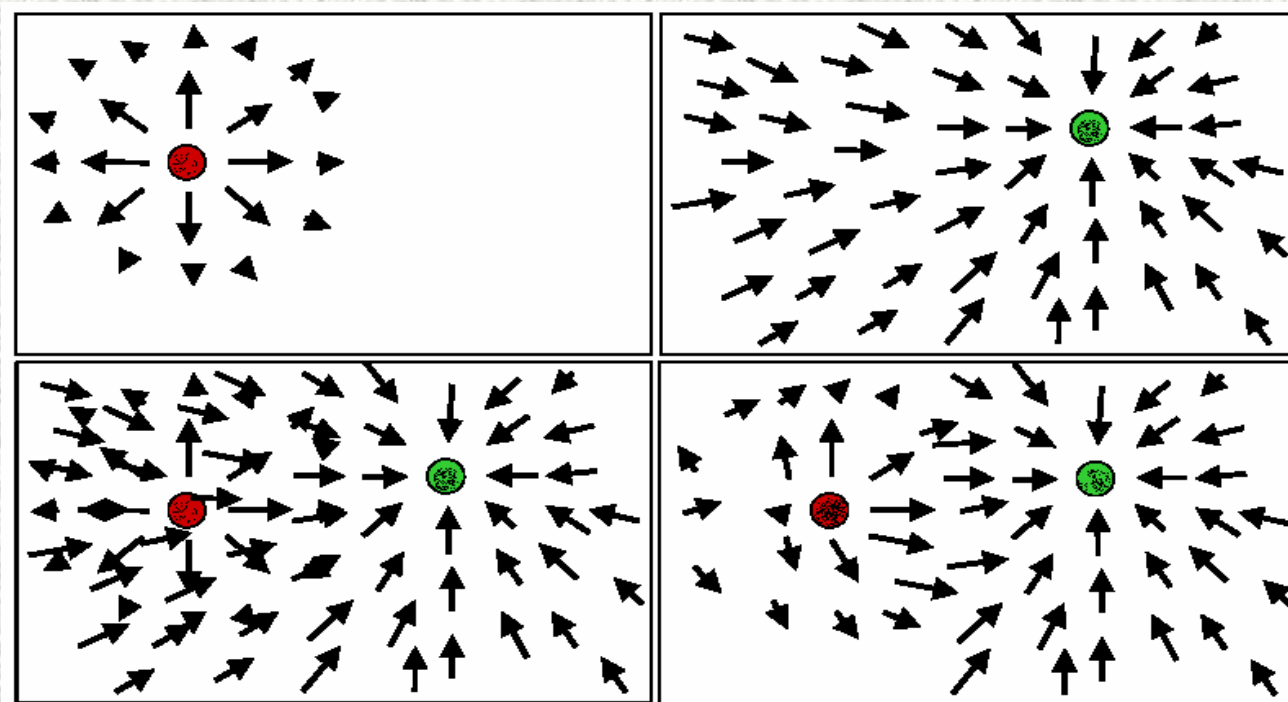
FUNDAMENTAÇÃO TEÓRICA

Campo de Repulsão



FUNDAMENTAÇÃO TEÓRICA

Utilização de Campos Potenciais



FUNDAMENTAÇÃO TEÓRICA

Arquitetura de Esquemas

Esquema Motor

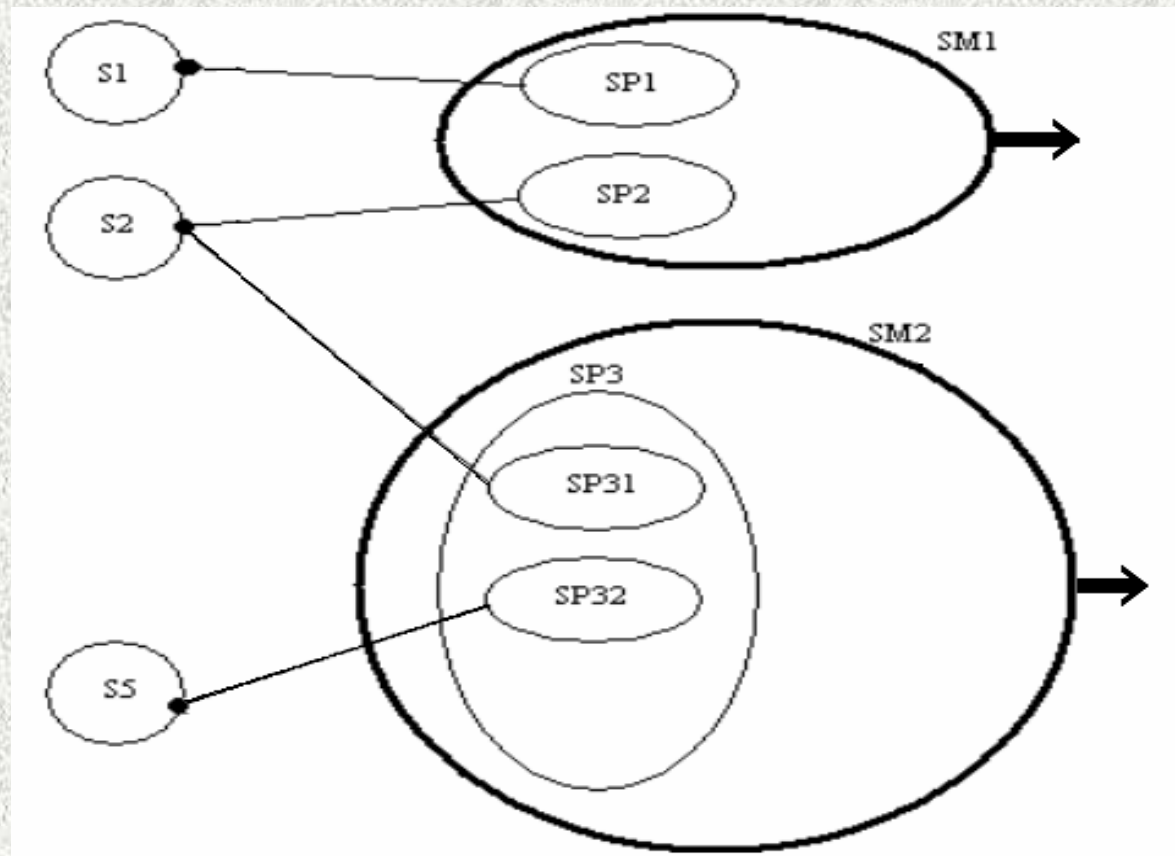
➡ Atividade física

Esquema Perceptivo

➡ Ambiente

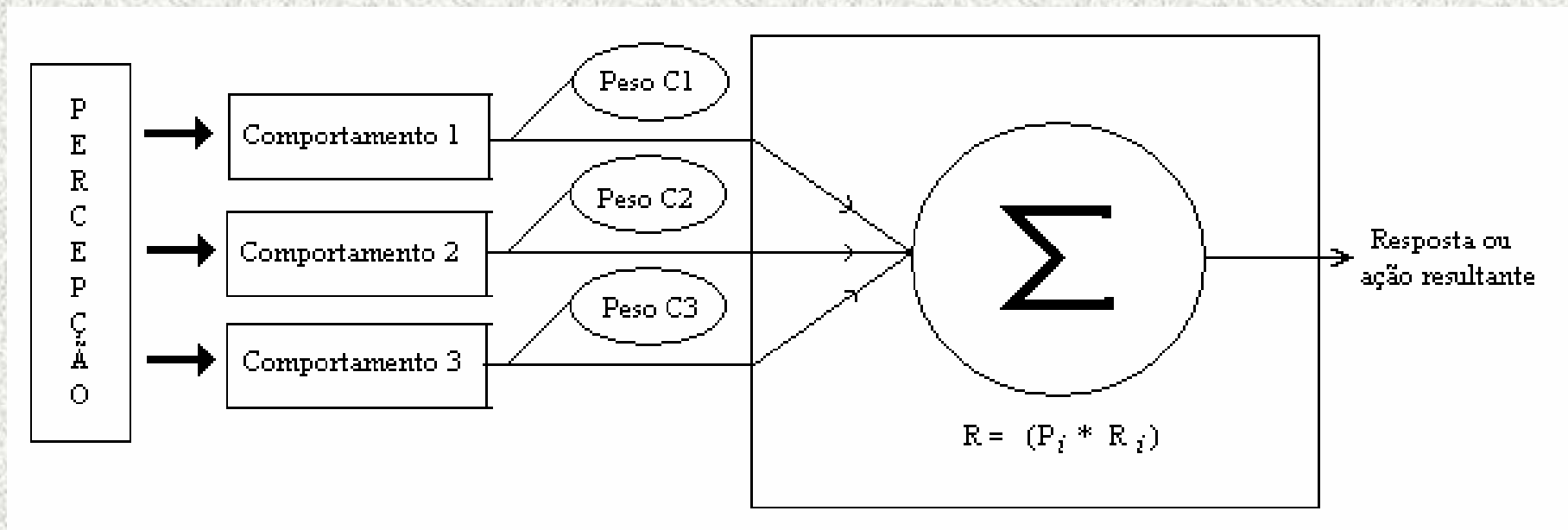
FUNDAMENTAÇÃO TEÓRICA

Arquitetura de Esquemas



FUNDAMENTAÇÃO TEÓRICA

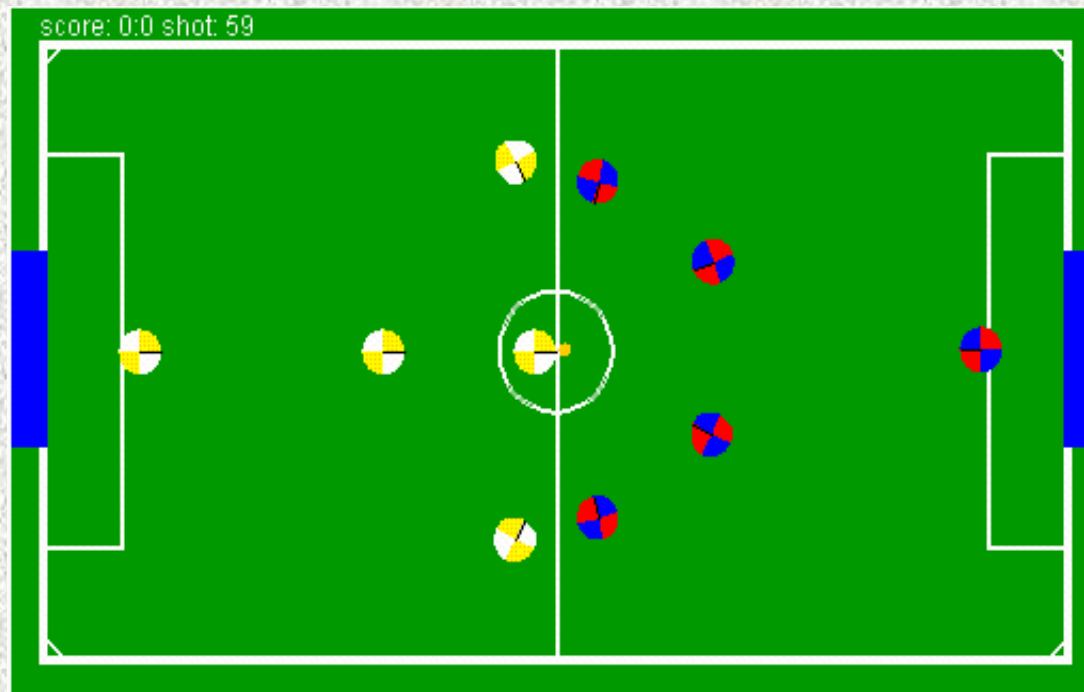
Construções de Comportamentos



FUNDAMENTAÇÃO TEÓRICA

TBSim

- ➔ clay
- ➔ abstractrobot



DESENVOLVIMENTO

- ➔ Requisitos do Problema a Ser Tratado
- ➔ Classe `v_RectangularAttraction_v`
- ➔ Comportamento Simples
- ➔ Comportamento Composto
- ➔ Time Formado

DESENVOLVIMENTO

Requisitos do Problema a Ser Tratado

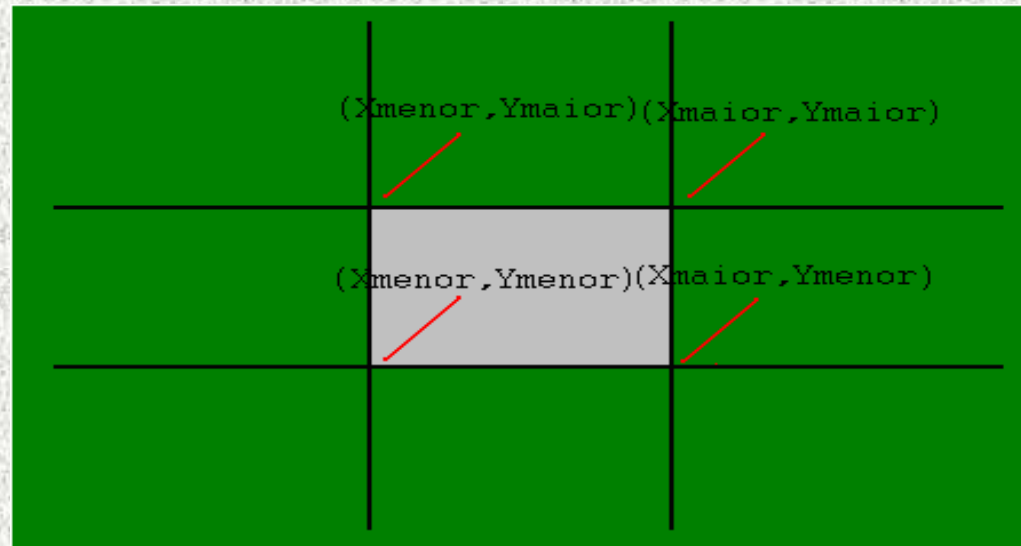
Objetivo

- ➡ Definir área de atuação para o agente robô jogador de futebol

DESENVOLVIMENTO

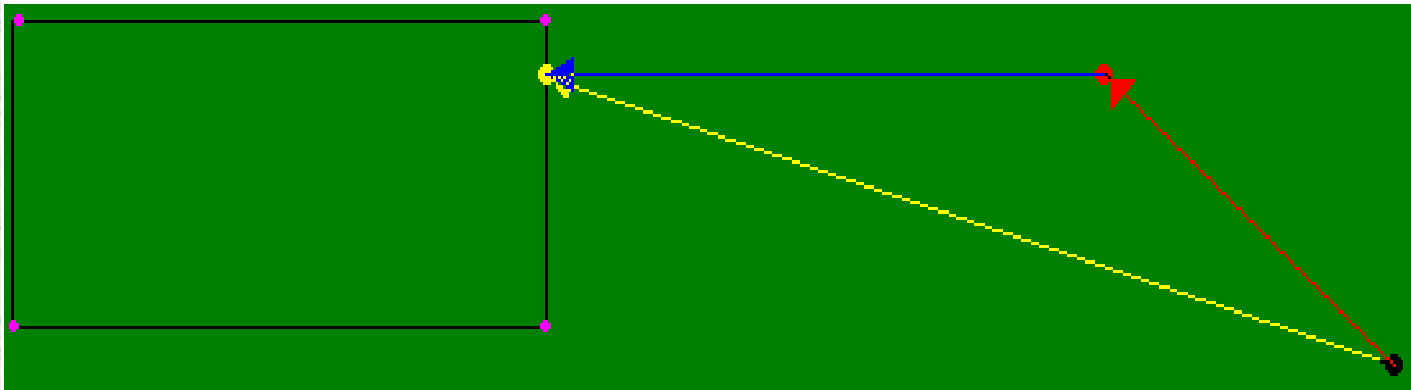
Requisitos:

➔ Criticar parâmetros



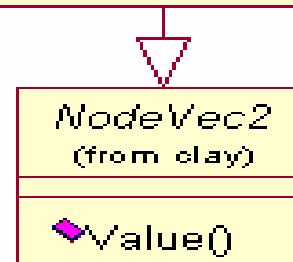
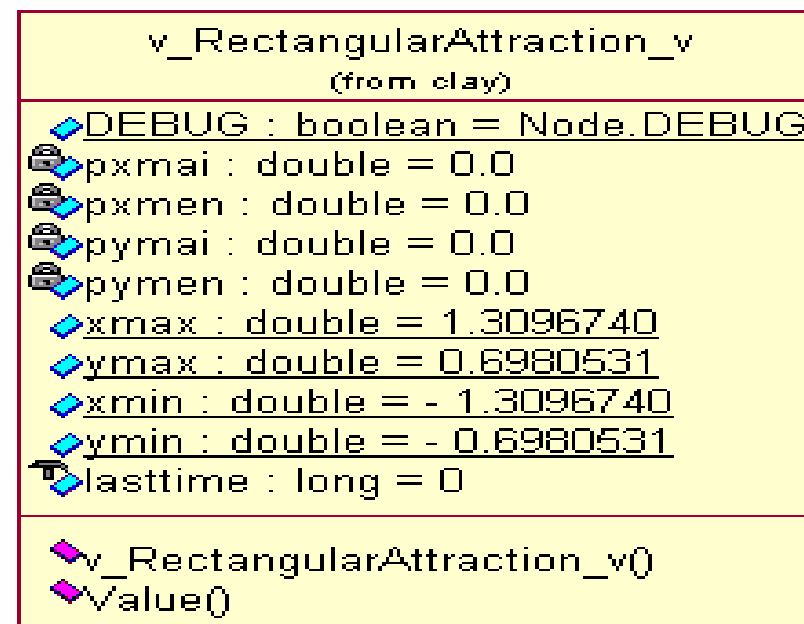
DESENVOLVIMENTO

- ➔ Definir extremidades da área de atuação
- ➔ Identificar ponto de atração
- ➔ Calcular vetor resultante
- ➔ Retornar vetor da atração



DESENVOLVIMENTO

Classe v_RectangularAttraction_v



DESENVOLVIMENTO

Construtor

```
24     public v_RectangularAttraction_v(double ptoxini, double ptoyini,  
25                                     double ptoxfim, double ptoyfim,  
26                                     Vec2 im1)
```

➔ Parâmetros

➔ Críticas

```
20     public static final double xmax = 1.3096740;  
21     public static final double ymax = 0.6980531;  
22     public static final double xmin = -1.3096740;  
23     public static final double ymin = -0.6980531;
```

➔ Normalizações

DESENVOLVIMENTO

Ponto de Atração

1	2	3
4	5	6
7	8	9

```
Se (Xmenor <= Xrobo >= Xmaior) E (Ymenor <= Yrobo >= Ymaior)
Entao Comeco
  Xponto = zero
  Yponto = zero
FIM
```

```
Se (Xmenor > Xrobo < Xmaior)
Entao Xponto = Xmenor
```

```
Se (Xmenor < Xrobo < Xmaior)
Entao Xponto = Xrobo
```

```
Se (Xmenor < Xrobo > Xmaior)
Entao Xponto = Xmaior
```

```
Se (Ymenor > Yrobo < Ymaior)
Entao Yponto = Ymenor
```

```
Se (Ymenor < Yrobo < Ymaior)
Entao Yponto = Yrobo
```

```
Se (Ymenor < Yrobo > Ymaior)
Entao Yponto = Ymaior
```

DESENVOLVIMENTO

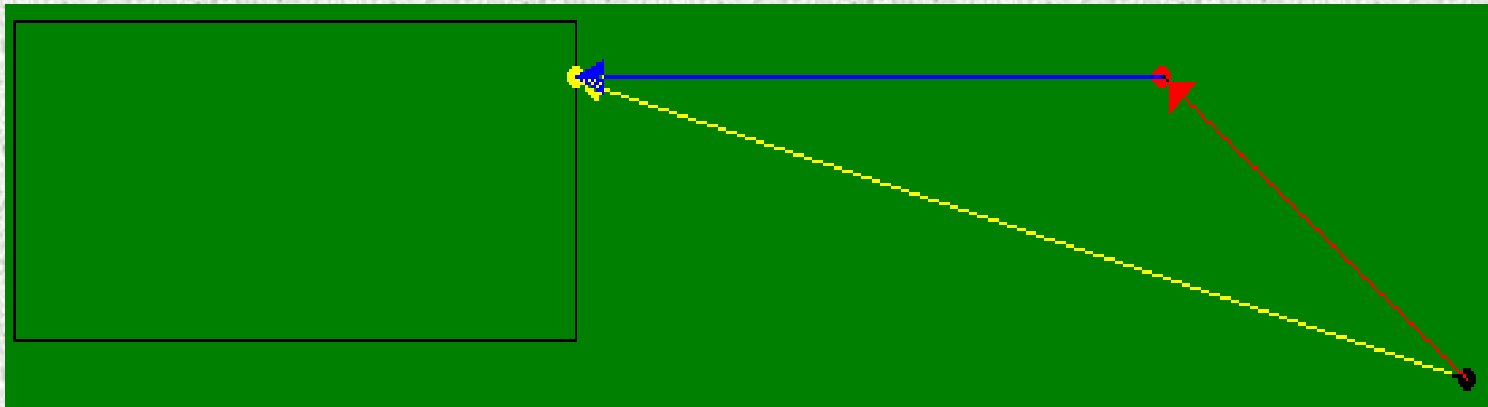
Implementação do cálculo do ponto de atração

```
108     if(!( ((pxmen <= goal.x) && (pxmai >= goal.x)) &&
109           ((pymen <= goal.y) && (pymai >= goal.y)) ))
110     {
111         /* regioao Esquerdo */
112         if ((pxmen < goal.x) && (pxmai < goal.x))
113             last_val.setx(-pxmai);
114         /* regioao Direito */
115         else if ((pxmen > goal.x) && (pxmai > goal.x))
116             last_val.setx(-pxmen);
117         /* regioao Centro */
118         else if ((pxmen <= goal.x) && (pxmai >= goal.x))
119             last_val.setx(-goal.x);
120
121         /* regioao Alto */
122         if ((pymen < goal.y) && (pymai < goal.y))
123             last_val.sety(-pymai);
124         /* regioao Baixo */
125         if ((pymen > goal.y) && (pymai > goal.y))
126             last_val.sety(-pymen);
127         /* regioao Centro */
128         else if ((pymen <= goal.y) && (pymai >= goal.y))
129             last_val.sety(-goal.y);
```


DESENVOLVIMENTO

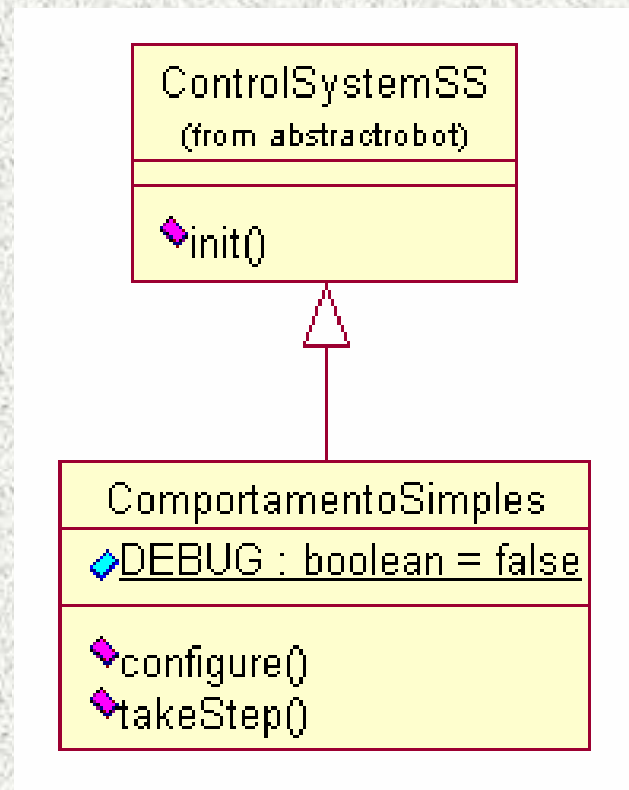
Vetor Resultante

```
130     goal.sub(last_val);  
131     last_val = goal;  
132 }
```



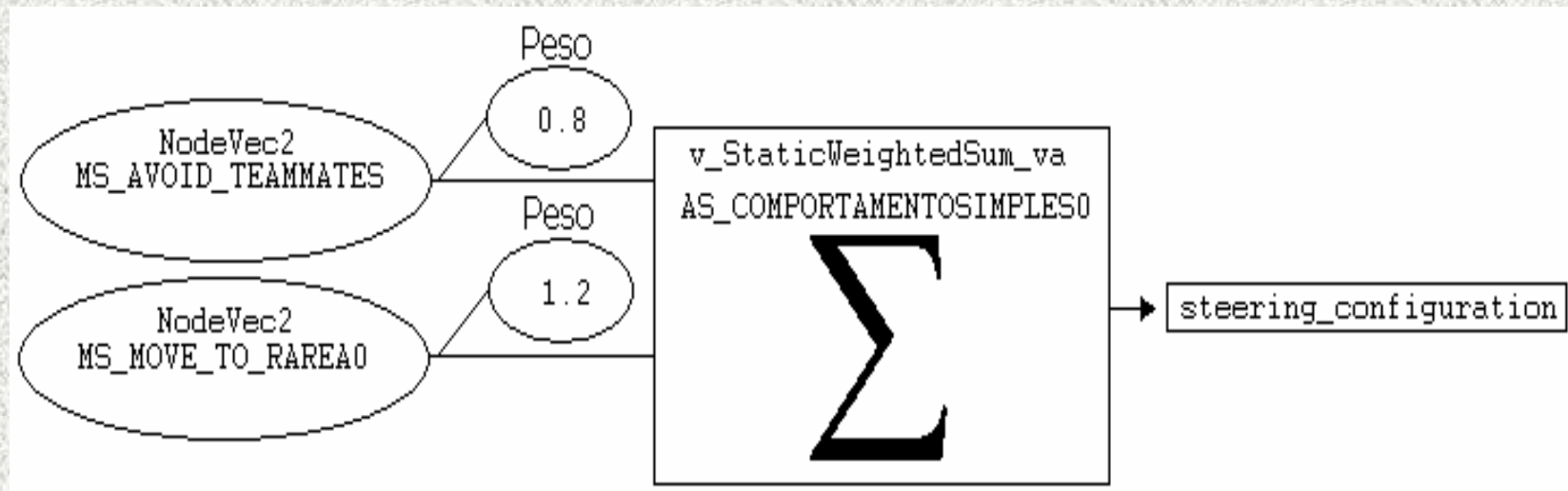
DESENVOLVIMENTO

ComportamentoSimples



DESENVOLVIMENTO

Ilustração do ComportamentoSimple

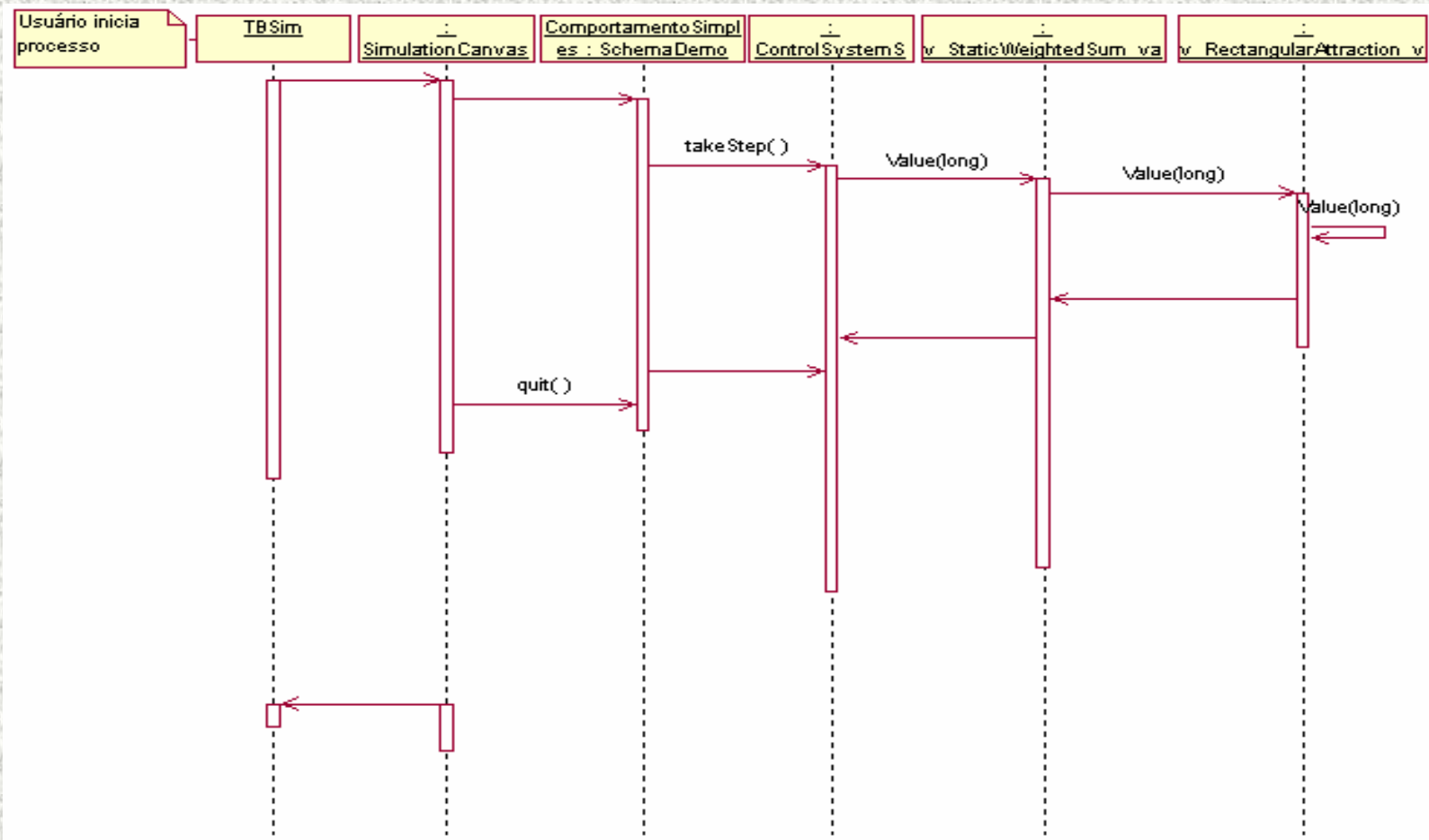


DESENVOLVIMENTO

```
67     if (n_roboto == 0) // área E
68     {
69         // Esquema motor para área retangular
70         NodeVec2
71         MS_MOVE_TO_RAREA0
72         = new v_RectangularAttraction_v(+1.3, +0.001, +1.27, -0.001, PS_GLOBAL_POS);
73
74         //=====
75         // CONSTRUÇÃO
76         //=====
77         // Esquema motor para evitar parceiros
78         v_StaticWeightedSum_va AS_COMPORAMENTOSIMPLES0 =
79         new v_StaticWeightedSum_va();
80         AS_COMPORAMENTOSIMPLES0.weights[0] = 0.8;
81         AS_COMPORAMENTOSIMPLES0.embedded[0] = MS_AVOID_TEAMMATES;
82
83         AS_COMPORAMENTOSIMPLES0.weights[1] = 1.2;
84         AS_COMPORAMENTOSIMPLES0.embedded[1] = MS_MOVE_TO_RAREA0;
85         steering_configuration = AS_COMPORAMENTOSIMPLES0;
86     }
87     else if (n_roboto == 2) // área A
```

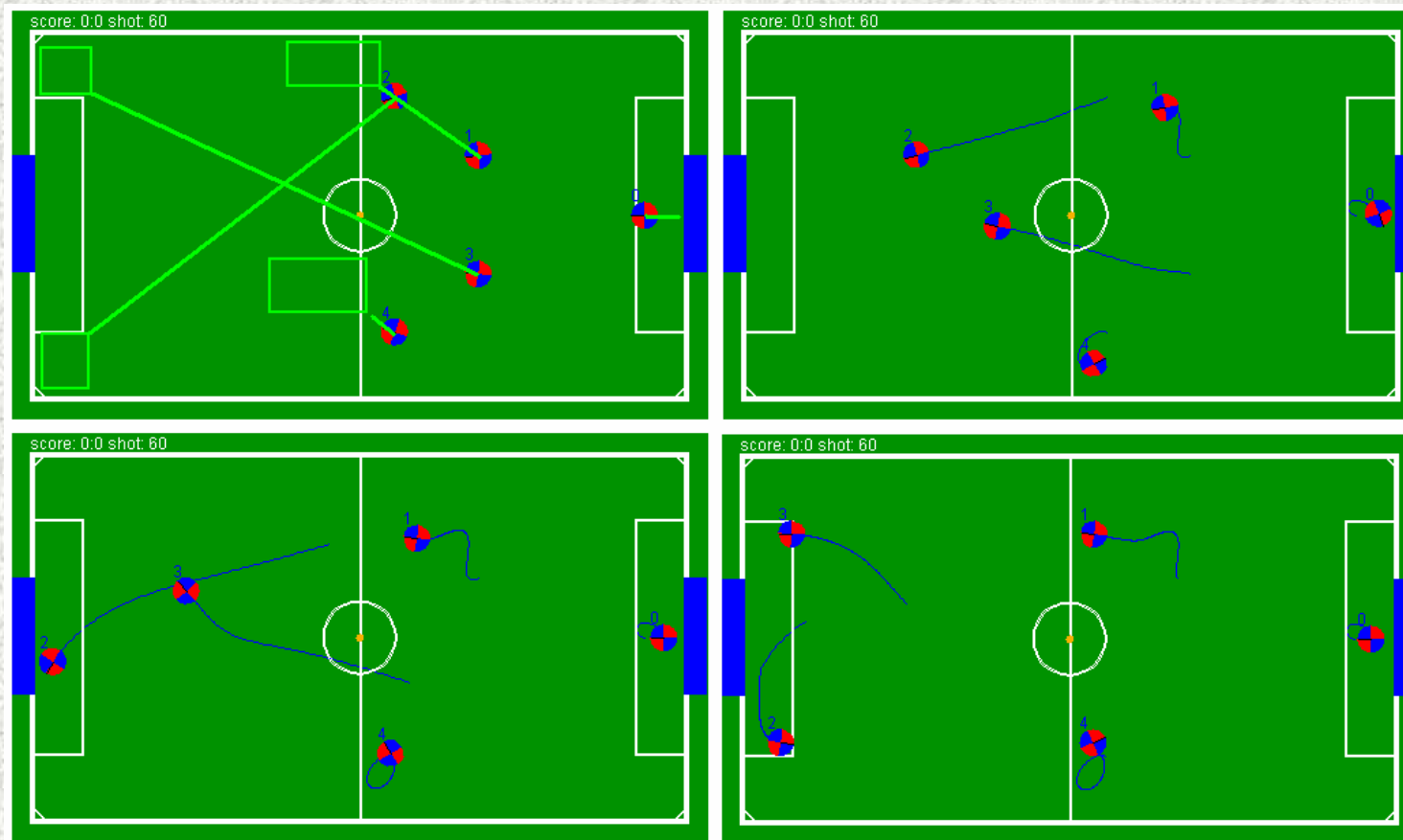
DESENVOLVIMENTO

Funcionamento da Classe v_RectangularAttraction_v



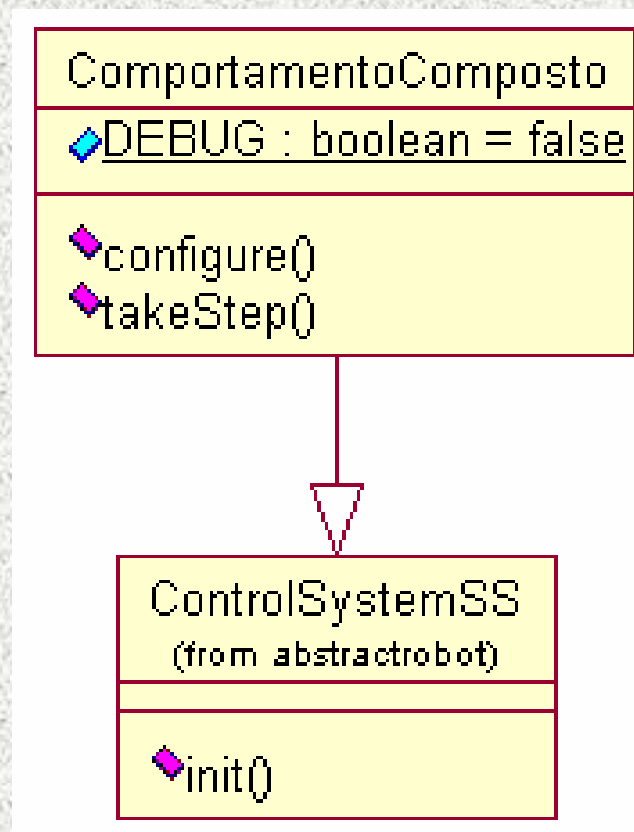
DESENVOLVIMENTO

Operacionalidade

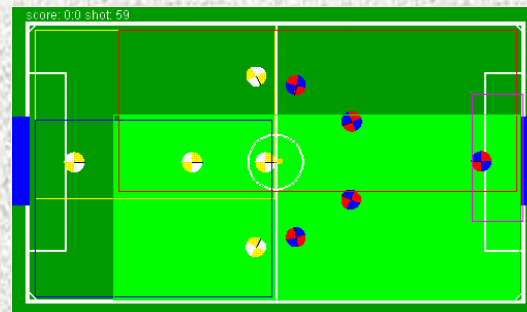
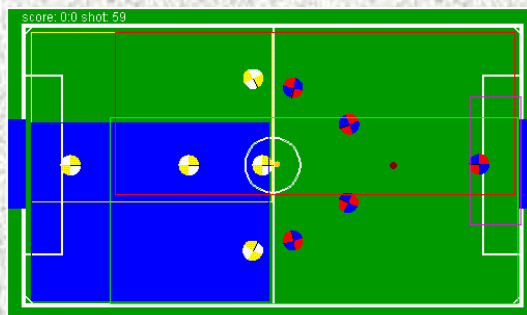
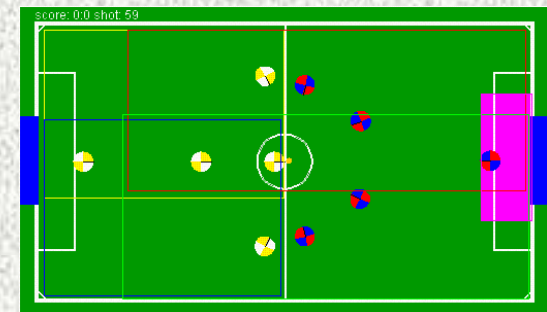
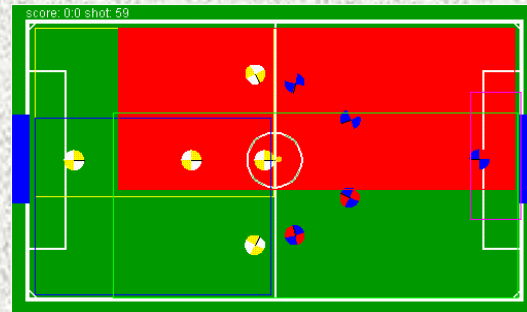
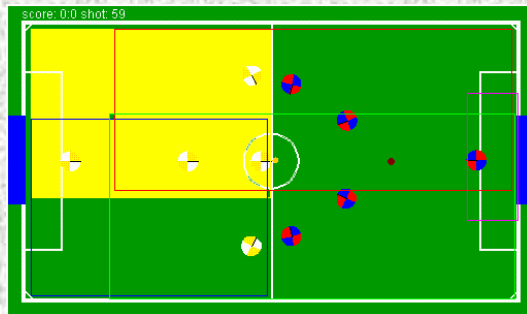


DESENVOLVIMENTO

ComportamentoComposto

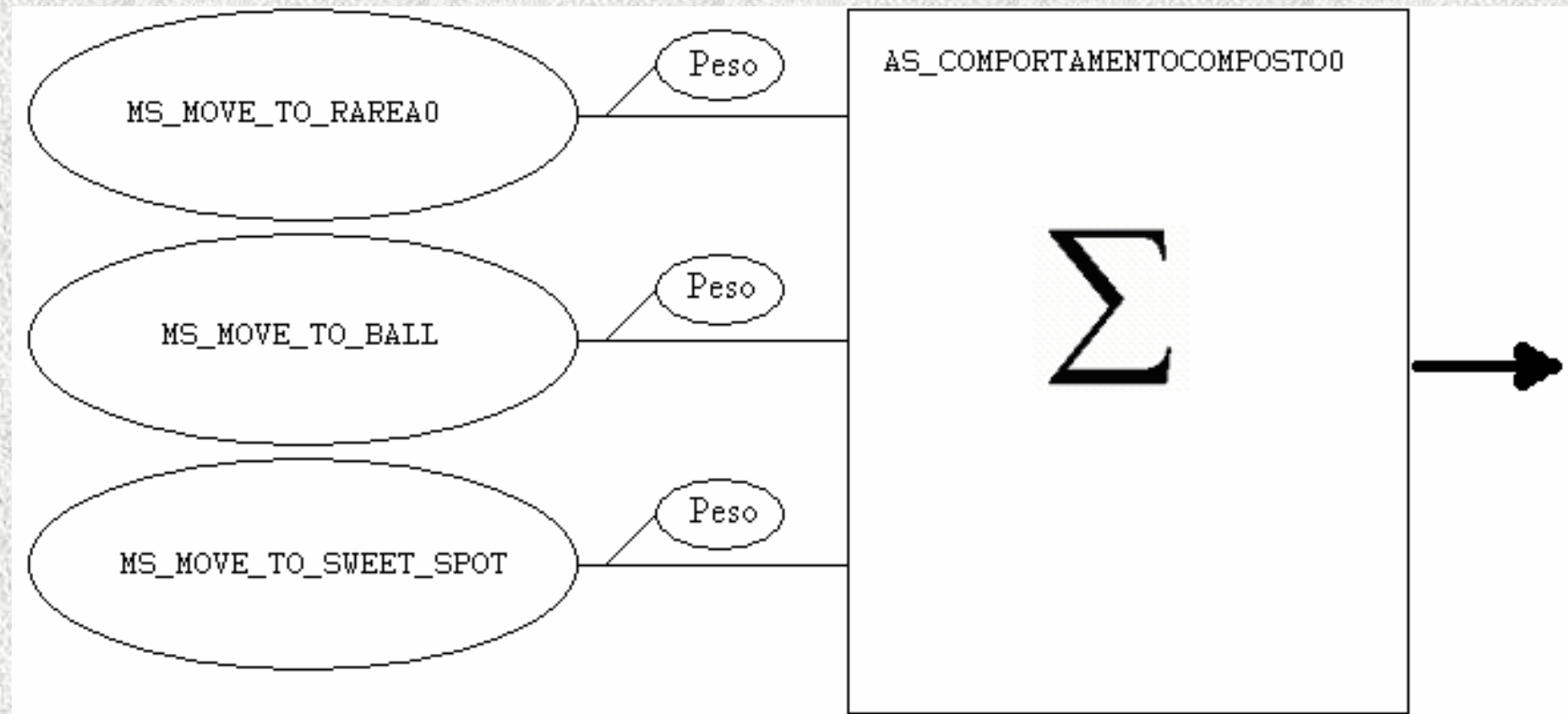


DESENVOLVIMENTO



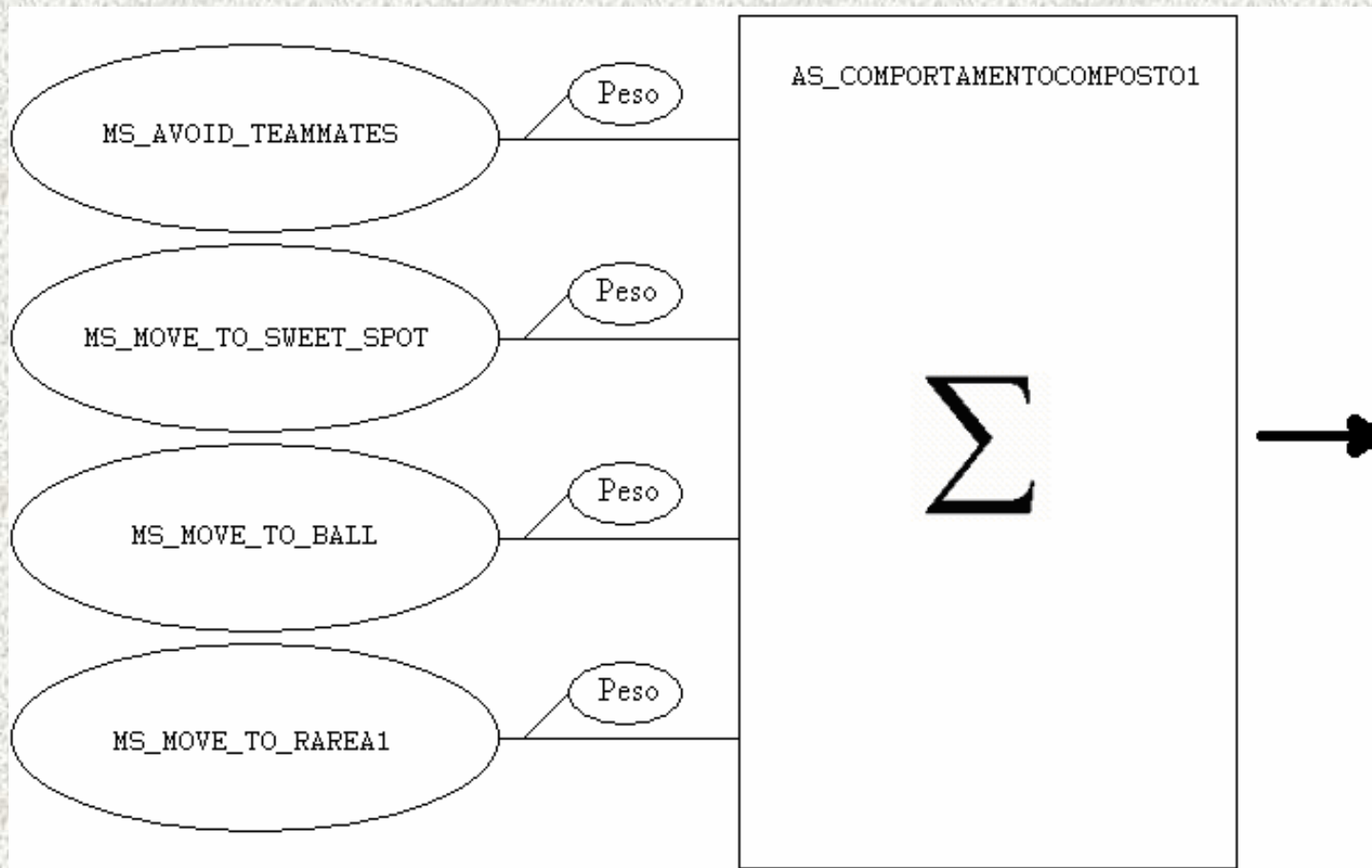
DESENVOLVIMENTO

Goleiro



DESENVOLVIMENTO

Ataque e defesa



PROBLEMAS E DIFICULDADES

- ➔ Linguagem Java
- ➔ Documentação TeamBots
- ➔ Depuração

CONCLUSÕES

- ➔ Desenvolvimento da classe `v_RectangularAttraction_v`
- ➔ Desenvolvimento de classes adicionais de Comportamentos

EXTENSÃO

- ➔ Desenvolvimento de classe que implemente repulsão retangular
- ➔ Desenvolvimento de novos comportamentos a fim de facilitar o objetivo, a vitória do time

Ciências da Computação

