



Universidade Federal Rural de Pernambuco

Disciplina: Sistemas Multiagentes

Professor: Pablo Azevedo Sampaio

Semestre: 2019.2

Descrição do Projeto (versão de 14/09/2019)

Vocês devem criar um programa para controlar **um time** de 8 jogadores para o Robocup 2D, usando uma biblioteca fornecida para comunicar com o servidor e usando Behavior Trees.

1. Grupos de Alunos

Fazer individual ou em **duplas**.

2. Biblioteca

Vocês usarão uma biblioteca Java para ajudar a desenvolver os times. A biblioteca simplifica o uso do simulador, oferecendo:

- percepções globais e perfeitas das posições dos jogadores e da bola;
- ações de mais alto nível, aceitando vetores (para direção ou posição), etc;
- facilidades para conectar e rodar um time no servidor.

Importante: algumas ações podem ter resultados imperfeitos (exemplo: a ação de rotação “turn” pode girar demais ou de menos). Isso é característica do simulador.

3. Requisitos do Time

O programa de vocês deve controlar **8 jogadores**, sendo 7 na linha e 1 goleiro.

Requisitos **técnicos**:

1. Cada jogador deve rodar de forma autônoma e descentralizada como uma *thread* separada.
2. Pode haver comunicação entre os jogadores, mas somente por meio de mensagens de *broadcast* (para todos) para o time em formato de texto.
3. A “inteligência” dos agentes deve ser programada com Behavior Trees.
4. Seu time deve ter um nome único, configurado no código.

Requisitos quanto ao **comportamento** dos **jogadores de linha**:

1. Os jogadores devem ter posicionamentos distintos, que devem ser definidos, a cada momento, em função deste critério, pelo menos:

- i. a posse de bola (se está com o seu time ou não); ou seja, seu time deve ter diferentes formações quando estiver com ou sem a bola.
2. Algum jogador deve ser capaz de, intencionalmente, **correr com a bola** (“carregar a bola”). Isso pode ser feito por meio de pequenos chutes (ações “kick”).
3. Deve haver frequentes **trocas de passes** intencionais entre jogadores do mesmo time.
4. Deve haver alguma forma de **jogada de ataque**, envolvendo mais de um jogador sempre que o time tiver a posse da bola.
5. Deve haver **jogadas de defesa** (marcação ou desarmes) sempre que o time estiver sem a posse da bola.
6. Deve haver equilíbrio e racionalidade nas ações. Exemplos de situações proibidas:
 - i. Todos os jogadores fazerem a mesma ação.
 - ii. Todos ficarem parados.
 - iii. Ações sem sentido para o momento.

Para o **goleiro**, há dois requisitos:

1. O goleiro deve se posicionar de forma racional, sendo este posicionamento definido com base, pelo menos, na posição da bola no campo.
2. O goleiro deve ser capaz de defender a bola quando esta passar suficientemente perto dele e com velocidade favorável à defesa.

Quanto aos **estados do jogo**, o seu time deve ser capaz de se comportar adequadamente em todos os estados da partida (ver `MatchPerception.getState()`). Alguns exemplos de estados:

- Prepare to Kick-off (posicionamento antes da liberação para o chute inicial)
- Kick-off (liberado para o chute inicial)
- Play on (em jogo)
- Free kick (falta, tiro livre), etc.

4. Primeira Entrega (16/outubro/2019)

O seu time deve cumprir os requisitos acima, mas está liberado de cumprir estes:

- Jogadores de linha, requisito 3 (troca de passes).
- Goleiro, requisito 2 (defender).
- Basta tratar dois estados do jogo: “prepare to kick-off”, “kick-off”, “play on” e “goal kick”.

Seu time deve ser capaz de ganhar de um time de referência que será fornecido.

Será feita uma competição entre os times, valendo parte da nota.

Entregar:

- Documento mostrando visualmente e explicando a Behavior Tree adotada no time.
- Projeto do Eclipse com o código. O nome do projeto deve ser o nome dos alunos.

5. Segunda Entrega (data a definir)

O seu time deve cumprir todos os requisitos.

A segunda versão do seu time deve ganhar do time da primeira entrega.

Espera-se mais sofisticação no posicionamento dos jogadores (por exemplo: posicionar-se livre de adversários) e nas jogadas e passes (por exemplo: lançamentos).

- Fortemente recomendado: usar técnicas da **aula 5**.

Entregar:

- Documento mostrando visualmente e explicando a Behavior Tree adotada no time.
- Projeto do Eclipse com o código. O nome do projeto deve ser o nome dos alunos.

6. Nota

Os requisitos serão usados para definir a maior parte da nota (mas tente fazer além do que é pedido).

Parte da avaliação pode ser definida por meio de partidas entre os times criados pelos diferentes grupos.

A avaliação envolverá entrevista e análise do código fonte.

7. Projetos Alternativos [Opcional]

Se não quiser desenvolver para a Robocup2D, vocês podem desenvolver algum outro time de agentes cooperativos para alguma outro simulador ou competição multiagente disponível na internet, mediante aprovação do professor. Entre em contato explicando sua idéia.

Um site com diferentes simuladores (usado em competições anuais, abertas para todo o mundo) é <https://multiagentcontest.org/>.

Vocês também podem conversar comigo sobre idéias de projetos que podem virar artigo ou TCC, se continuados depois da disciplina.