



Processo seletivo RobôCIn 2020 - Simulação 2D

Descrição do problema:

A Simulação 2D tem seu funcionamento todo baseado em agentes autônomos, onde 11 jogadores (players) interagem para compor um time de futebol simulado e cada player faz o gerenciamento de suas próprias ações conforme a dinâmica do ambiente. Assim surgem diversos desafios computacionais, visto que os players precisam lidar com troca de mensagens entre si, visualizar o ambiente a volta, gerenciar suas energias e tomar ações com base nisso para compor um time de futebol simulado. Dentre esses desafios, alguns tópicos se destacam, cada tópico tem problemas dos quais precisamos achar soluções para tornar o time competitivo. Para o projeto, escolha um (ou mais) dos tópicos abaixo e desenvolva.

1) Formações dos players

- Os players procuram seguir formações definidas, ocorrendo o chaveamento delas de acordo com a posição da bola e a situação do jogo. Por exemplo, quando o time está com a bola no campo do adversário e em uma situação de ataque os jogadores vão seguir a formação ofensiva. Os adversários são estudados antes das partidas, podendo existir problemas na sua formação que possam ser explorados. Procure fazer uma análise dos pontos negativos e positivos da(s) formação(ões) e busque uma solução para os problemas encontrados.

2) Comportamentos individuais

- Comportamentos individuais são chaves quando é preciso que o player realize uma ação para uma determinada situação, tal como fazer a marcação de um outro player adversário ou se posicionar na projeção da bola para fazer o bloqueio do toque adversário. A partir da nossa base de código que será disponibilizada, implemente um tipo de comportamento para o jogador de uma posição específica (ex: atacante).

3) Análise de jogo

- Os projetos de análise de jogo estão divididos em etapas. Não é necessário concluir todas etapas ter o projeto aceito/concluído. Gostaríamos de ver suas ideias ao longo do desenvolvimento do projeto.
 - 3.1 (Etapa 1): Após cada partida jogada, os dados de comunicação entre o servidor e os agentes (jogadores) é guardada em um registro (log). Dentro desse registro existem dados que, quando tratados, podem se tornar informações preciosas de inteligência para entendermos melhor como está sendo o desempenho dos nossos



algoritmos. Usando os logs de partidas que disponibilizamos, desenvolva uma ferramenta de análise desses dados, extraindo informações importantes da partida e relevantes para notar o desempenho do time.

3.2 (Etapa 2): Na equipe escrevemos muito código a partir do zero que várias pessoas irão ler e trabalhar sobre. Escrever um código que funcione é importante, entretanto, escrever um código que seja de fácil entendimento para outros membros da equipe, reutilizável e que não seja tão complicado de sustentar é tão importante quanto. É por conta disso que existem diversas metodologias e arquiteturas para se escrever software. Uma delas é o design pattern chamado *Singleton*. Utilizando sua ferramenta construída em 3.1, adapte seu código conforme a arquitetura de um *Singleton*.

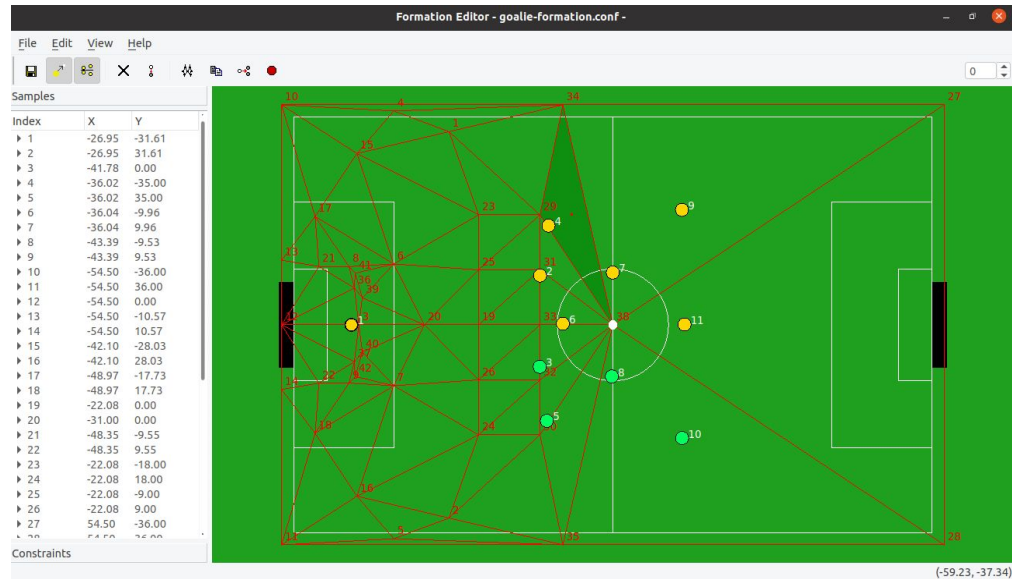
4) Tomada de decisão

- O processo de tomada de decisão do agente é um processo complexo, passa por uma geração de possíveis ações, projeção do ambiente ao ser feita essa ação e avaliação do estado, a ação que gera o estado com melhor avaliação é tomada. Modificar a função de avaliação do estado é o que altera o comportamento do agente. Sua função é modificar o comportamento do agente em função do tempo da simulação.

Restrições de projeto:

Projeto 1:

- A análise das formações podem ser feitas de forma visual, por meio da execução dos jogos ou com a análise dos logs ;
- Devem ser listados os pontos negativos e positivos da(s) formação(ões) com a sua justificativa, seria interessante mostrar visualmente os problemas com exemplificações práticas;
- O chaveamento das formações pode ser encontrado no arquivo `strategy.cpp` do código base.
- A ferramenta utilizada para construir ou modificar as formações é a Fedit, disponível no repositório da seletiva e sendo exemplificada a seguir:



Projeto 2:

- Usar código base do Agent2d e implementar nele os comportamentos (C++), acesse esse repositório para saber onde pode ser implementada a função;
- Seguir as regras da Simulação 2D ao implementar o comportamento;

Projeto 3:

- Pode ser implementado em qualquer linguagem;
- Logs dos jogos em .csv para serem analisados podem ser baixados [aqui](#) ou no [repositório](#) oficial da seletiva ;
- Desenvolva análises que façam sentido e explique como ela pode ser usada, exemplo: uma das análises que posso tirar desses logs é a posse de bola de um determinado time, com isso posso criar uma heurística para definir a posse de bola do time durante o jogo.

Projeto 4:

- Usar código base do Agent2d e implementar nele os comportamentos (C++), acesse esse repositório para saber onde pode ser implementada a função;
- Seguir as regras da Simulação 2D ao implementar o comportamento;

Pode ser escolhido mais de um projeto para ser apresentado, apenas tenha noção do tempo e no que você deseja focar em aprender com tudo isso. De resto, divirta-se!

Na sua apresentação, tente responder às seguintes perguntas:



- Na sua opinião, e após sua pesquisa, quais contribuições da abordagem desenvolvida?
- Existem pontos em que suas abordagem podem ser melhoradas ou otimizadas?
- O que você aprendeu pesquisando e desenvolvendo o projeto?
- Quais dificuldades encontradas durante o desenvolvimento?

Links úteis:

- Site oficial da Simulação 2D com as regras - [aqui](#).
- Site com alguns TDPs (*Team Description Paper*) - [aqui](#).
- Link Github do código base - [aqui](#)