

## EXPLICAÇÃO DO PROJETO FINAL DA DISCIPLINA DE INTRODUÇÃO ÀS TÉCNICAS DE PROGRAMAÇÃO - UNIDADE 3 (TURMA 02)

Foi desenvolvido um *bot* para o jogo de simulação capaz de vencer a partida de forma simples, sem receber multas e sutilmente eficiente. O *bot* conta com um algoritmo que encontra lugares de pesca e de venda mais perto do qual o *bot* está posicionado. Além disso, o código, no qual o *bot* foi criado, está modularizado através de funções e de fácil entendimento.

Explicando o código de forma resumida, a função principal, *main*, contém um laço que chama a função *readData* sucessivas vezes, na qual é responsável por ler os dados do mapa e do *bot* a cada rodada. Logo após isso, a função *findFishingSpots* é chamada e realiza a operação de encontrar locais de pesca mais próximos ao *bot* e que não tenha apenas um peixe, pois isso faria com que o *bot* recebesse multa. O ponto principal dessa função é que ela chama outra função, *shortestDistance*, que calcula a distância entre dois pontos. Seguindo, é feita a movimentação do *bot* até o local de pesca mais próximo chamando a função *movement*. No local pesca, o *bot* pesca até que sobre apenas um peixe. Por fim, restando um peixe, a função *findFishingPorts* é chamada para que se busque o porto mais próximo do *bot* (de forma parecida com a função *findFishingSpots*) e encontrando, chamasse novamente a função *movement* para o *bot* dirigisse ao porto para então vendesse os peixes.

Portanto, foi pensando em uma lógica que seja simples e sutilmente eficiente. A ideia de buscar locais de pesca que não tenha apenas um peixe atrelado com a ação de pescar até que reste um peixe facilita a implementação, pois evita que o *bot* passe duas vezes no mesmo local de pesca e impede o *bot* de levar multa por impacto ambiental de não haver mais peixe naquele lugar. Além disso, a ação de pescar até que se reste um peixe e em seguida dirigisse ao porto não gera multa por quilo excedente, tendo em vista que em um local de pesca não ultrapassa 9 quilos. Dessa forma, o código foi escrito para que seja de fácil entendimento, fácil implementação e um tanto quanto reutilizável.

**Victor Gabriel Oliveira do Couto.**