

INF1771 - INTELIGÊNCIA ARTIFICIAL

TRABALHO FINAL – DESAFIO FINAL IA

Website: <http://augustobaffa.pro.br/site/inf1771-inteligencia-artificial-desafio-dos-drones/>

Descrição:

Sua missão é enviar um drone para terras distantes e controla-lo remotamente. Seu objetivo é encontrar a maior quantidade de tesouros possível, evitando os obstáculos e os perigos que possa encontrar pela frente! Você não é o único: outros drones também serão enviados para o mesmo destino e estarão atrás dos mesmos tesouros. Para poder combater-los, cada drone possui uma arma que pode ser usada quando necessário. Seu drone possui os seguintes sensores que ajudam a reconhecer o ambiente: sensor de brisa para evitar poços, abismos ou penhascos; sensor de flashes para evitar possíveis vórtices que podem teletransporta-lo; sensor de objetos para detectar itens no chão (redLight para power ups, blueLight para tesouros ou weakLight quando incerto); áudio para ouvir os “passos” dos adversários e sensor da mira que identifica um adversário na sua frente.

Forneceremos apenas o equipamento necessário para missão (drone, servidor para conexão e um devkit em python, java, C/C++ ou C# para ajuda-lo). Sua tarefa será programar a inteligência artificial que controlará nosso drone. Até o dia da missão, vamos manter um servidor para simular as condições do ambiente para treina-lo até o grande dia. Confiamos no seu conhecimento em IA para conseguirmos os tesouros! Boa sorte!



Figura 1: Jogo em acontecimento – Servidor (Versão 1.0)

O trabalho 4 consiste em implementar uma interface principal em C, C++, C#, Java, python ou **qualquer linguagem** utilizando **qualquer método/técnica de Inteligência Artificial** apresentado na disciplina. Não é necessário desenvolver uma interface gráfica para o agente.

Características:

- O labirinto pode ser representado por uma matriz 59 x 34.
- O agente inicia em uma posição aleatória do labirinto.
- O Agente terá 100 pontos de energia inicial.
- A munição do agente dá 10 de dano nos inimigos.
- Os inimigos têm 100 pontos de energia inicial. (São outros agentes)
- Powerups que carregam energia.
- Munição ilimitada.
- O agente pode executar as seguintes ações:
 - **Mover_para_frente;**
 - **Mover_para_trás** (andar de ré);
 - **Virar_a_esquerda** (rotação de 90°);
 - **Virar_a_direita** (rotação de 90°);
 - **Pegar_objeto** – Para pegar o outro (se ele existir) na posição em que o agente se encontra;
 - **Atirar** – Para atirar a munição em linha reta na direção que o agente está olhando – A munição é ilimitada e tem alcance até colidir com uma posição bloqueada;
 - **Observar** – Recebe as informações do mundo ao seu redor;
- Cada ação executada possui o custo de -1 (andar, virar para a esquerda, direita, etc). Os demais eventos possuem os seguintes custos/recompensas:
 - **Cair em um poço** (obstáculo) = -1000;
 - **Ser morto pelos inimigos** = -10;
 - **Matar um inimigo** = +1000;
 - **Atirar** = -10;
 - **Pegar** = -5 + ganho do item; (a tentativa mesmo que não tenha nada)

OBS: Os valores podem ser ajustados para a partida ou diferentes no servidor.
- O agente possui os seguintes sensores:

- Em coordenadas adjacentes aos inimigos, exceto diagonal, o agente ouve um som de passos;
- Em coordenadas adjacentes a um poço/obstáculo, exceto diagonal, o agente sente uma brisa;
- Em coordenadas adjacentes ao inimigo que teletransporta, exceto diagonal, o agente percebe um flash;
- Em coordenadas onde existem itens o agente percebe o brilho de uma luz;
 - **Redlight** = powerup
 - **Bluelight** = tesouros
 - **Greenlight** = veneno (não será utilizado nos mapas do trabalho)
 - **Weaklight** = indefinido
- Ao caminhar contra uma parede o agente sente um impacto. As laterais do labirinto são paredes; O mapa também pode conter outras posições bloqueadas;
- Quando um inimigo recebe um dano, o agente que disparou o tiro é notificado;
- Quando o tiro de um inimigo acerta o agente, ele é notificado;
- O labirinto possui os seguintes elementos:
 - **Poços**;
 - **Moedas de ouro** (+1000);
 - **Anéis de ouro** (+500);
 - **Poções** (power up +10, +20 ou +50)
 - **Obstáculos que bloqueiam o caminho**;
 - **Teletransportes**

OBS: Os valores podem ser ajustados para a partida ou diferentes no servidor.

- O agente não tem acesso às informações do mapa.
- Cada partida possui 10 minutos. O jogo se o agente morrer por dano ou ao cair em um poço.
- Ao entrar em uma coordenada onde existe um teletransporter, o agente “acorda” em um lugar aleatório do labirinto, podendo ser um local seguro ou a coordenada de um inimigo.

Dinâmica do servidor:

O servidor utiliza um protocolo texto simples para as comunicações e é implementado utilizando um Socket TCP/IP ouvindo a porta "8888". Os parâmetros são separados utilizando o símbolo ponto e vírgula (";"). A maioria dos comandos não requer parâmetros.

O servidor possui três estados:

- **Ready:** 30 segundos para os agentes se prepararem ou conectarem;
- **Game:** jogo em andamento;
- **Gameover:** 30 segundos para os agentes poderem processar os resultados dos jogos;

Durante os estados Ready e Gameover somente os comandos básicos do servidor estarão disponíveis (scoreboard, estados do jogo, troca de nome, troca de cor e chat). Os comandos que controlam o agente são desabilitados.

Observações enviadas pelo servidor:

- **blocked:** último movimento não foi feito. Destino está bloqueado;
- **steps:** há um inimigo próximo há até 2 passos de distância de manhatan;
- **breeze:** há uma brisa (buraco) adjacente (1 passo em distância de manhatan);
- **flash:** há um clarão (teletransporte) adjacente (1 passo em distância de manhatan);
- **blueLight:** há uma luz azul fraca (tesouro) na posição do jogador;
- **redLight:** há uma luz vermelha fraca (powerup) na posição do jogador;
- **greenLight:** há uma luz verde fraca (veneno) na posição do jogador;
- **weaklight:** há uma luz fraca não identificável na posição do jogador;
- **enemy:** detectado inimigo em até 10 passos na direção o qual o jogador está olhando. Normalmente apresentado como "eneny#xx", onde xx é o número de passos.
- **damage:** o jogador levou um dano
- **hit:** o jogador acertou um tiro

Comandos do Servidor:

Os comandos do servidor estarão disponíveis através de métodos implementados na classe disponibilizada pelo devkit. Segue uma lista de comandos e seus métodos equivalentes:

w	sendForward(); - anda para frente
s	sendBackward(); - anda de ré
a	sendTurnLeft(); - virar a esquerda 90º
d	sendTurnRight(); - virar a direita 90º
t	sendGetItem(); - pegar item
e	sendShoot(); - atirar
o	sendRequestObservation(); - receber observações (separado por ; e ,)
g	sendRequestGameStatus(); - receber o status do jogo (estado, tempo atual)
q	sendRequestUserStatus(); - receber status do usuário (posição, estado do agente, pontos e energia)
p	sendRequestPosition(); - receber posição do agente
u	sendRequestScoreboard(); - lista de usuários logados e pontos
quit	sendGoodbye(); - desconectar do jogo
name	params[1]: name sendName – trocar de nome
say	params[1]: msg sendSay – enviar mensagem
color	params[3]: r(0-255), g(0-255), b(0-255) sendColor, sendColor(color) – trocar de cor

Requisitos do Trabalho:

- O programa deve ser implementado em qualquer linguagem e utilizar uma interface Socket TCP/IP com o servidor (porta 8888. Serão disponibilizados DevKits).
 - Servidor de treino: **atari.icad.puc-rio.br**
- O programa deve implementar qualquer método demonstrado na disciplina (Busca, Lógico, Incerteza, Aprendizado de Máquina, Fuzzy, Waypoints e/ou Máquina de Estados).
- O programa deve exibir um log (em tela) das ações realizadas.

Data de Entrega:

10/dez (quarta-feira) na sala de aula

Forma de Entrega:

O programa deverá ser executado durante a competição e o código deverá ser disponibilizado em repositório git.

Dúvidas?

Podem enviar as dúvidas para email ou Discord