

Desenvolvimento de um Agente autónomo para o jogo
Sokoban

Trabalho de grupo

Inteligência Artificial / Introdução à Inteligência Artificial

Ano Lectivo de 2020/2021

Diogo Gomes
Luís Seabra Lopes

7 de Outubro de 2020



I Observações importantes

1. Este trabalho deve ser realizado em grupos de 2 a 3 estudantes. **Em cada módulo Python submetido, deve incluir um comentário com o nome e número mecanográfico dos autores.**
2. Uma primeira versão do programa deverá ser submetida até 20 de Novembro de 2020. A versão final deve ser submetida até 11 de Dezembro de 2020. Nas duas submissões o trabalho pode ser submetido para lá do prazo, mas será penalizado em 5% por cada dia adicional.
3. Cada grupo deverá submeter o seu código através da plataforma *Classroom* do *GitHub*. **Na submissão final, inclua uma apresentação (tipo Powerpoint), em formato .pdf e com o nome presentation.pdf, com um máximo de cinco páginas, onde deve sumarizar a arquitectura do agente desenvolvido.**
4. **O código deverá ser desenvolvido em Python 3.5. O módulo principal deverá ter o nome student.py.**
5. **Se discutir este trabalho com colegas de outros grupos, inclua um comentário com o nome e número mecanográfico desses colegas. Se utilizar outras fontes, cite-as também.**
6. Todo o código submetido deverá ser original; embora confiando que a maioria dos grupos fará isso, serão usadas ferramentas de detecção de copianço. Alunos envolvidos em casos de copianço terão os seus trabalhos anulados.
7. O projecto será avaliado tendo em conta: **desempenho; qualidade da arquitectura e da implementação; e originalidade.**

II Tema do trabalho

Este trabalho envolve a aplicação de conceitos e técnicas de três capítulos principais da matéria leccionada, nomeadamente: programação em Python; arquitecturas de agentes; e técnicas de pesquisa para resolução de problemas.

No âmbito deste trabalho, deverá desenvolver um agente capaz de jogar de forma inteligente o jogo Sokoban, um jogo que se tornou popular através dos primeiros computadores pessoais como C64, Amiga, Spectrum, IBM/PC e Macintosh no início da década de 80 do século passado.

No *Sokoban* tradicional, o jogador humano controla um agente que se desloca num armazem repleto de caixas, sendo que o objectivo é arrumar as ditas caixas em locais pré-definidos no mapa, sinalizados por pequenos diamantes. Para movimentar as caixas o agente terá que as empurrar. Não existem adversários nem factores aleatórios. Cada mapa é um quebra-cabeças único de múltiplas soluções. O objectivo do jogo consiste pois em arrumar todas as caixas nas posições correctas minimizando o tempo e o numero de movimentos do agente.

A pontuação do jogo tem em conta os níveis ultrapassados pelo agente e a eficiência da soluções encontradas. Todos os níveis têm um tempo limite para serem concluídos.

1 Objectivos

- Para ter nota positiva, o agente deverá ser capaz de ultrapassar os primeiros 3 níveis.
- O agente deverá ser capaz de jogar um jogo num mapa de dimensões desconhecidas (utilizado apenas para avaliação).
- O desempenho será tanto melhor quantos mais níveis o agente completar.

III Funcionamento do jogo

- o *Sokoban* inicia cada nível numa posição pré-definida do mapa.
- O *Sokoban* tem acesso em cada momento a todo o mapa, com a localização das caixas.
- O *Sokoban* pode-se deslocar utilizando os comandos "w" (*cima*), "s" (*baixo*), "a" (*esquerda*) e "d" (*direita*).
- Existe um limite de tempo para completar cada nível, que é fornecido no início do jogo.
- O *Sokoban* corre sempre o risco de empurrar uma caixa para uma situação da qual não consegue recuperar, dando origem a uma situação de bloqueio (deadlock) que deve ser detectada pelo próprio agente.
- A pontuação obtida em cada nível é calculada em função do numero de caixas arrumadas, o tempo gasto e o número de movimentos necessários para alcançar o estado actual.
- A pontuação total do jogo é dada pelo somatório das pontuações obtidas nos sucessivos níveis.

IV Código e Apoio ao desenvolvimento

Um motor de jogo *Sokoban* escrito em Python encontra-se disponível em <https://github.com/dgomes/ia-ia-sokoban>.

Todas as entidades do jogo são representadas por classes.

Cada grupo desenvolve um agente criando um cliente que implementa o protocolo exemplificado no ficheiro `client.py`. Não é necessário modificações aos demais ficheiros, como por exemplo o `game.py` ou `mapa.py`, mas pode criar novos ficheiros, pastas, etc.

Se implementar uma nova funcionalidade ou implementar algum melhoramento ao motor de jogo e/ou visualizador pode criar um "Pull Request" (PR) na plataforma GitHub. Se a sua alteração for aceite, ser-lhe-á creditado um bónus na avaliação final até um máximo de 1 valor.

O agente desenvolvido deverá ser entregue num módulo com nome `student.py` e quando o agente se liga ao servidor deverá usar como *username* o número mecanográfico de um dos elementos do grupo (qualquer).

Existe um canal de apoio em <https://detiuaveiro.slack.com/messages/ai/> onde os alunos poderão colocar dúvidas e receber notificações de alterações.

Dada a novidade deste trabalho/código, é expectável que existam alguns bugs e ajustes durante o desenrolar do trabalho. Estejam atentos a modificações no servidor (`git pull`) e a notificações no *e-mail*, *Slack* e *e-learning*.

Para dar início ao trabalho, deve formar um grupo com colegas, aceder ao link <https://classroom.github.com/g/GPpMPF24> e fazer o *fork* do código para o grupo. Deverá ser feito apenas um *fork* por grupo. Um dos elementos do grupo cria o grupo associando os demais elementos, após este passo deverá ser criado o *fork* (não crie um novo *fork* sem todos elementos estarem registados).

V Conselhos

Comece por estudar o `client.py`. O código é muito básico e simples, pelo que comece por remodelar (*refactor*) o cliente de forma orientada a objectos.

1. Fazer `git log` para se manter informado de pequenas alterações que foram ou venham a ser feitas.
2. Acompanhar o canal `#ai` no Slack