

Desenvolvimento de um agente autónomo para o jogoCobra

Tarefa de grupo

Inteligência Artificial

Ano Letivo de 2024/2025

Diogo Gomes
Luís Seabra Lopes

9 de outubro de 2024



Notas importantes

- 1.º Este trabalho deverá ser realizado em grupos de 2/3 alunos. Em cada módulo Python submetido, deverá incluir um comentário com o nome dos autores e número mecanográfico.
- 2.º Uma primeira versão do programa deverá ser submetida até 27 de novembro de 2024. A versão final deverá ser submetida até 18 de dezembro de 2024. Em ambas as submissões o trabalho poderá ser submetido fora do prazo, mas será penalizado em 5% por cada dia adicional.
- 3.º Cada grupo deverá submeter o seu código através do *Sala de aula GitHub* plataforma. Na submissão final inclua uma apresentação (tipo Powerpoint) em formato .pdf e nomeado apresentação.pdf, com um máximo de cinco páginas, onde deverá resumir a arquitetura do agente desenvolvido.
- 4.º O código deve ser desenvolvido em pelo menos Python 3.11. O módulo principal deve ser nomeado estudante.py.
5. Se discutir este trabalho com colegas de outros grupos, inclua um comentário com o nome e o número mecanográfico desses colegas. Se utilizar outras fontes, cite-as também.
- 6.º Todo o código enviado deverá ser original; embora confiando que a maioria dos grupos cumprirá este requisito, serão utilizadas ferramentas para detetar o plágio. Os alunos envolvidos em casos de plágio terão a tarefa cancelada.
- 7.º O projeto será avaliado tendo em conta: desempenho; qualidade da arquitetura e implementação; e originalidade.

II Visão Geral

Este trabalho envolve a aplicação de conceitos e técnicas de três capítulos principais do curso de IA, a saber: programação Python; arquiteturas de agentes; e técnicas de pesquisa para a resolução automatizada de problemas.

No âmbito deste trabalho, deverá desenvolver um agente capaz de jogar de forma inteligente o jogo Snake, popularizado pela Nokia no seu telefone 6110 de 1998.

Baseado no jogo Blockade original de 1976. Visto de uma perspectiva de cima para baixo, cada jogador controla uma “cobra” com uma posição inicial fixa. A “cabeça” da cobra avança continuamente, incapaz de parar, crescendo cada vez mais. Deve ser dirigido de forma a evitar embater em quaisquer obstáculos, incluindo o corpo do adversário e também o seu próprio corpo. Geralmente é jogado no modo multijogador. Ganha o jogador que sobreviver durante mais tempo. As versões para um jogador possuem uma ou mais cobras controladas pelo computador.

Nesta versão do Snake introduzimos algumas modificações. A principal modificação é que cada cobra tem uma visão limitada do mundo, ou seja, não vê para além de uma determinada distância da cabeça da cobra. Além dos alimentos “normais” que aumentam o corpo da cobra em uma unidade, existem superalimentos e venenos. Os superalimentos podem, por exemplo, fazer crescer a cobra em mais do que uma unidade e/ou aumentar a visão da cobra. Os venenos, por outro lado, podem diminuir a espessura da cobra (removendo unidades da cauda da cobra) ou diminuir a sua visão. Super Food e Poison são indistinguíveis, e o jogador só sabe o que é quando o come.

Superalimentos e venenos:

- PONTOS: pode ser um superalimento (aumenta a pontuação do jogador) ou um veneno (diminui a pontuação do jogador) com uma quantidade aleatória de pontos.

- **COMPRIMENTO:** pode aumentar ou diminuir o tamanho da cobra (quantidade aleatória [-5, 5])
- **ALCANCE:** pode aumentar ou diminuir o alcance de visão da cobra (quantidade aleatória [-5, 5])
- **TRAVERSE:** ao iniciar a cobra pode atravessar paredes e o mundo é uma esfera, mas quando este veneno é comido a cobra vive num mundo plano e pode embater nas paredes. Comer TRAVERSE uma segunda vez altera o comportamento.

Os alunos devem elaborar estratégias para jogar no modo autónomo (marcando pontos por crescer o máximo possível dentro de um limite de tempo) e no modo multijogador (mesmas condições, mas com cobras extra a competir pelos mesmos recursos).

1 Objetivos

- Para obter uma nota positiva, o agente deve ser capaz de aumentar o tamanho do corpo da cobra em 100 vezes o tamanho inicial.
- Marque o máximo possível. (ver abaixo detalhes sobre a marcação)

III Regras do Jogo

- *Cobra* começa com um mapa que pode estar vazio ou conter paredes.
- O mapa é uma esfera (envolve-se em si própria) e a cobra pode colidir consigo própria, contra outras cobras (na versão multijogador) e contra paredes que possam existir no mapa. As paredes não podem ser movidas ou criadas durante o jogo. No entanto, existe um veneno que pode transformar as fronteiras do mapa em paredes (e o mundo torna-se plano).
- O *Cobra* jogador (agente estudante) tem acesso limitado ao mapa. O servidor do jogo apenas passa informação do mapa nas proximidades da cabeça da cobra. O alcance de visão pode alterar-se (aumentar ou diminuir) com base nos alimentos ingeridos.
- O agente controla um cursor, através do qual pode mover a Cobra na vertical ou na horizontal (na diagonal não é permitido).
- Em *Cobra*, pode utilizar os comandos "c" (*subir*), "e" (*descer*), "um" (*deslocar para a esquerda*) e "d" (*deslocar para a direita*).
- Um jogo de cobras é composto por um único mapa. Cada jogo pode ter um mapa diferente.
- Cada mapa do jogo tem um valor de tempo limite. Se a cobra não se matar, o jogo terminará no valor de tempo limite definido. Nos jogos multijogador, o jogo continua até que todos os jogadores morram ou expirem.

Código IV e Apoio ao Desenvolvimento

UM *Cobra* motor de jogo escrito em Python está disponível em <https://github.com/dgomes/ia-snake>.

Todas as entidades do jogo são representadas por classes.

Cada grupo desenvolve um agente criando um cliente que implementa o protocolo exemplificado no `cliente.py` ficheiro. Nenhuma modificação noutros ficheiros é necessária (não pode alterar `jogo.py`), mas pode criar novos ficheiros, pastas, etc.

Se implementar uma nova funcionalidade ou implementar alguma melhoria no motor de jogo e/ou visualizador, pode criar um “Pull Request” (PR) na plataforma GitHub. Se a sua alteração for aceite, ser-lhe-á creditada uma bonificação na avaliação final até ao máximo de 1 ponto (em 20).

O agente desenvolvido deve ser entregue num módulo denominado `estudante.py`. O agente deve ligar-se ao servidor de jogo local, utilizando como *nome de utilizador* o número mecanográfico de um dos elementos do grupo (qualquer).

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

Dada a novidade do motor de jogo, prevê-se que ocorram alguns bugs e ajustes no decorrer do trabalho. Esteja atento às modificações do servidor (configure o repositório do professor como um repositório upstream e pesquise/mesclasse frequentemente) e notificações em *e-mail*, *Folga* e *e-learning*.

Para iniciar o trabalho, deve formar um grupo com colegas e aceder ao link <https://sala.deaula.github.com/a/rnI3I4bM> que irá *garfoar* o código do grupo. Apenas um *garfo* deve ser feito por grupo. Um dos elementos do grupo cria o repositório do grupo e associa os restantes elementos. Após esta etapa, o *garfo* será criado automaticamente (não crie um novo *garfo* se os elementos estiverem registados).

V Recomendações

- 1.º Comece por `estudarcliente.py`. O código é muito básico e simples, por isso comece por *refatoração* o cliente para algo mais orientado para um Agente de IA autónomo.
2. Periodicamente busque o repositório original para atualizar o seu código.
- 3.º Corra o registro do git para acompanhar pequenas alterações que foram feitas.
- 4.º Siga o canal #ai no Slack

VI Esclarecimento de dúvidas

Serão aqui colocados esclarecimentos sobre as principais dúvidas que possam surgir durante a execução da obra.

1. Pergunta: Como será avaliado o desempenho dos agentes?

Resposta: Os agentes serão avaliados quanto ao seu desempenho num conjunto de jogos com base na soma das pontuações finais desses jogos. A pontuação final de um jogo é aquela que o agente possui no momento da partida. *jogo Over*.

2. Pergunta: Como será avaliado o trabalho prático?

Resposta: As pontuações totais do jogo para cada agente enviado são mapeadas em marcas tendo em conta a distribuição das pontuações totais. Uma pontuação total de 100.000 está mapeada para 20/10. A mediana das pontuações totais está mapeada para 16/20. A pontuação total máxima está mapeada para 20/20. Outras pontuações totais são mapeadas linearmente.

Avaliação da 1ª entrega

- Neste delivery apenas deverá ser enviado o código do agente.
- Cada agente jogará 10 jogos e será obtida a média desses 10 jogos.

Avaliação da 2ª entrega

- Cada agente jogará 10 jogos.
- Nesta entrega é necessária a apresentação de uma apresentação (ver ponto I.3 supra).
- A avaliação da segunda entrega reflete o desempenho do agente (90%), a qualidade do design (de acordo com a apresentação entregue, 7,5%) e a qualidade da implementação (2,5%).