Desenvolvimento de um Agente autónomo para o jogo **Bomberman**

Trabalho de grupo

Inteligência Artificial / Introdução à Inteligência Artificial

Ano Lectivo de 2019/2020

Diogo Gomes
Luís Seabra Lopes

16 de Setembro de 2019

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 1 de Novembro de 2019. A versão final deve ser submetida até 22 de Novembro de 2019. 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 BomberMan, um jogo que se tornou popular através da consola de jogos Nintendo Entertainment System (NES) de finais do século passado.

No *Bomberman* tradicional, o jogador humano controla um agente que se que desloca num labirinto, com paredes que podem ser destruíveis ou indestrutíveis, em busca dos seus adversários. Todos os adversários devem ser eliminados antes de o *Bomberman* poder passar ao nível seguinte através de um portal (dito portal de saída) escondido por detrás de uma das paredes destruíveis.

Para poder alcançar estes objectivos, o *Bomberman* pode colocar bombas no seu caminho que explodem de forma omnidirecional num raio pré-definido (que pode variar com base em *powerups* adquiridos no decorrer do jogo) e passado um tempo pré-determinado (que pode ser controlado por um *powerup* especifico).

A pontuação do jogo é feita pelos níveis ultrapassados pelo agente. Todos os níveis têm um tempo limite para serem concluídos.

1 Objectivos

- Para ter nota positiva, o agente deverá ser capaz de ultrapassar o nível 1.

- 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 *Bomberman* inicia todos os niveis na mesma posição. Nenhum inimigo ou parede pode ser encontrada nas imediações directas do *Bomberman*.
- O *Bomberman* tem acesso em cada momento a todo o mapa, com a localização dos inimigos e das paredes destruíveis. Caso já tenha descoberto, recebe também a localização do portal de saída, localização de powerups não consumidos e uma lista comulativa de *powerups* já consumidos.
- O *Bomberman* pode-se deslocar utilizando os comandos **"w"** (*cima*), **"s"** (*baixo*), **"a"** (*esquerda*) e **"d"** (*direita*) e colocar bombas utilizando o comando **"B"**. Está ainda disponível um detonador remoto através do comando **"A"** após aquisição do *powerup* correspondente.
- Está disponivel 1 powerup por nível. Os powerups possiveis são:

Powerup	Descrição
Bombs	Aumenta o número de bombas que podem ser colocados em 1. Powerup cumulativo
Flames	Aumenta o raio das bombas em 1 quadrado. Powerup cumulativo
Speed	Aumenta a velocidade de movimentos do bomberman
Wallpass	Permite passar por cima de paredes destrutiveis
Detonator	Bombas deixam de explodir após um timeout, passando a mais antiga a explodir com o comando **"A"**
Bombpass	Permite passar por cima de bombas
Flamepass	Imunidade às explosões

- Cada inimigo tem velocidade, inteligência e capacidades especiais próprias. Pode distinguir os inimigos pelo seu nome (propriedade passada no documento que o agente recebe com a actualização de estado).
- O número e tipo de inimigos num dado labirinto é fixo e pré-determinado.
- As localizações do portal de saída e do *powerup* em cada nível é gerada aleatoriamente.
- Todos os inimigos partem de um posição aleatória a cada nível.
- Existe um limite de tempo para completar cada nível, que é fornecido no início do jogo.
- O *Bomberman* tem 3 vidas, e perde uma vida sempre que colide com um dos inimigos ou se uma explosão o atingir.
- A pontuação obtida em cada nível é calculada de acordo com a seguinte tabela:

Pontos	Descrição
100	destruição de um *Balloom*
200	destruição de um *Oneal*
400	destruição de um *Doll*
800	destruição de um *Minvo*

- A pontuação total do jogo é dada pelo somatório das pontuações obtidas nos sucessivos níveis.
- Sempre que o *Bomberman* morre, todo o nivel mantém o seu estado à excepção do *Bomberman* que volta à posição de partida.

IV Código e Apoio ao desenvolvimento

Um motor de jogo *Bomberman* escrito em Python encontra-se disponível em `https://github.com/dgomes/iia-ia-bomberman`.

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).

Um canal de apoio existe em `https://detiuaveiro.slack.com/messages/ai/` onde 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 inicio ao trabalho deve aceder ao link: `https://classroom.github.com/g/DIVkejPf` e desta forma criar o seu grupo e o seu *fork* do código. Deverá haver 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

VI Esclarecimento de dúvidas

Esclarecimentos sobre as principais dúvidas que venham a surgir durante a realização do trabalho serão colocados aqui.

1. **Questão**: Quantos níveis existem num jogo?

   Resposta: O conjunto de níveis, e respectivas características, a considerar em cada jogo no âmbito da avaliação de desempenho será anunciado em 2019/10/15.

2. **Questão**: Como será avaliado o desempenho dos agentes?

   Resposta: Os agentes serão avaliados pelo seu desempenho num conjunto de jogos com base no somatório das pontuações finais nos vários jogos. A pontuação final num jogo é aquela que o agente tiver no momento de *gameover* (quando completa o último nível ou quando perde a última vida). A distribuição das notas de avaliação de desempenho é ajustada com base na distribuição das pontuações totais, tendo em conta que, para ter nota 10 na componente de avaliação de desempenho, o agente deve conseguir completar o nível 1.

3. **Questão**: Como vão ser avaliados os trabalhos práticos?

   Resposta:

   Avaliação da 1ª entrega

   - Nesta entrega deve ser submetido apenas o código do agente.
   - Cada agente jogará 10 jogos.

   Avaliação da 2ª entrega

   - Serão adicionados novos inimigos e níveis. O código correspondente será fornecido logo após 1ª entrega.
   - Cada agente jogará 15 jogos.
   - Nesta entrega é necessário submeter uma apresentação (ver acima ponto I.3).
   - A avaliação da segunda entrega reflecte o desempenho do agente (90%), a qualidade da concepção (segundo a apresentação entregue, 7.5%) e a qualidade da implementação (2.5%).