

1 2



9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

Departamento de Engenharia Informática

Introdução à Inteligência Artificial
2024/2025 - 2º Semestre

Trabalho Prático Nº1:
Image Agents

Nota: A fraude denota uma grave falta de ética e constitui um comportamento inadmissível num estudante do ensino superior e futuro profissional licenciado. Qualquer tentativa de fraude levará à anulação da componente prática tanto do facilitador como do prevaricador, independentemente de ações disciplinares adicionais a que haja lugar nos termos da legislação em vigor. Caso haja recurso a material não original, as **fontes** devem estar explicitamente indicadas. Caso use ferramentas de IA na produção deste trabalho (e.g. ChatGPT), deverá identificar de forma clara todas as partes em que a ferramenta esteve envolvida. Note que, durante a defesa, deverá demonstrar ter conhecimento profundo dos conteúdos gerados pela ferramenta, sendo esse conhecimento objeto de avaliação.

1 Introdução

Agentes reativos são agentes que operam com base no estado atual do ambiente, funcionando através de regras simples de percepção ação, reagindo diretamente a estímulos e agindo sobre eles. São eficazes em ambientes dinâmicos, mas podem ter dificuldades em lidar com situações que necessitem de previsão ou adaptação a longo prazo. Para além disso, é interessante notar que estes princípios podem ser aplicados à manipulação de imagens, onde o ambiente considerado será representado por uma grelha bidimensional de pixels. Neste contexto, considera-se um agente reativo que existe num ambiente bidimensional representado por uma imagem, onde cada pixel corresponde a uma célula de uma grelha discreta. O agente desloca-se ao longo da grelha, podendo alterar os pixels de acordo com regras pré-definidas, possibilitando a aplicação de filtros e operações de imagem de forma iterativa. Este tipo de abordagem pode ser útil para tarefas como deteção de artefactos da imagem, edição e melhoria de imagem, deteção de padrões ou transformação estilística de imagens, explorando a interação direta entre o agente e o ambiente visual.

2 Objetivos Genéricos

O presente trabalho tem como objetivos genéricos:

1. Aprender a desenhar um agente reativo adequado a um problema e ambiente específico.
2. Aprender a especificar e formalizar os seguintes aspetos de um agente reativo:
 - (a) Perceções
 - (b) Ações
 - (c) Memória
 - (d) Sistema de Produções
3. Aprender a fazer uma descrição de alto-nível do comportamento desejado implementável através de um agente reativo.
4. Aprender a criar um agente reativo a partir de uma descrição de alto nível.

3 *Image Agents*

Este trabalho prático tem como objetivo principal a aquisição de competências relacionadas com a análise, desenvolvimento, implementação e teste de sistemas de produção para o controlo de agentes autónomos. Através deste projeto, os alunos terão a oportunidade de aplicar conceitos de inteligência artificial e explorar as suas capacidades no controlo de agentes que navegam num ambiente 2D.

3.1 Ambiente

O ambiente é constituído por uma grelha, construída a partir da cor dos pixels de uma imagem (ver Figura 1), e com tamanho variável, permitindo o ajuste do nível de detalhe. Neste ambiente, habitam agentes capazes de explorar através de percepções e ações. Esses agentes, através da navegação e modificação do ambiente, atuam como filtros. Como consequência, a interação de diferentes agentes com o ambiente resultará em diferentes tipos de transformações. Na Figura 1 podemos ver a atuação do agente *GlowTaker*, um agente que tem preferência por pixels da grelha com brilho maior, retirando brilho deles a cada passagem.



Figura 1: Exemplo da atuação do agente *GlowTaker*. Da esquerda para a direita: ambiente inicial e ambiente após a atuação do agente.

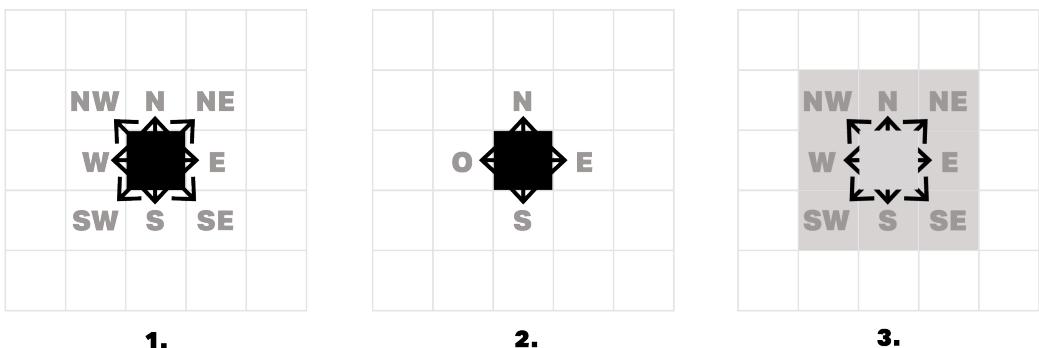


Figura 2: Propriedades do agente: 1. vizinhança das percepções 2. possibilidade de movimento 3. vizinhança das ações .

3.2 Os agentes

No contexto deste projeto, foram criados diferentes agentes que exploram o ambiente, podendo alterar ou não as casas por onde passam ou a sua vizinhança. De forma a navegar no ambiente de forma mais orientada os agentes podem recolher informações acerca da sua vizinhança, assim como da casa onde se encontram. Para isso, têm vários sensores orientados nas seguintes direções: Casa atual (A), Norte (N), Sul (S), Oeste (W), Este (E), Noroeste (NW), Sudoeste (SW), Nordeste (NE) e Sudeste (SE), como se pode observar na Figura 2.1.

3.2.1 Perceções

Cada agente possui 9 percepções, que lhe permitem extrair as cores da sua vizinhança (N, S, W, E, NW, SW, NE, SE), assim como a cor da própria casa (A). Além disso, consegue ainda extrair o brilho dessas mesmas posições, resultando em mais 9 percepções. Cada agente consegue também percecionar se essas mesmas casas foram marcadas por ele, resultando em mais 9 percepções. Finalmente, cada agente consegue também percecionar se a vizinhança está dentro ou fora dos limites da grelha, resultando em mais 8 percepções. De forma sumária, as percepções podem ser agrupadas da seguinte forma:

- Cor da vizinhança e da posição atual (9 Percepções)
- Brilho da vizinhança e da posição atual (9 Percepções)
- Vizinhança e posição atual marcada ou não pelo agente (9 Percepções)
- Vizinhança dentro ou fora dos limites da grelha (8 Percepções)

3.2.2 Ações

Cada agente pode movimentar-se para as células vizinhas que se encontram na mesma linha ou na mesma coluna (N, S, W, E), sendo as 4 ações designadas por (ver Figura 2.2.):

- Ir para o Norte (N)
- Ir para o Sul (S)
- Ir para o Oeste (W)
- Ir para o Este (E)

Cada agente pode ainda marcar em todas as direções disponíveis na vizinhança (N, S, W, E, NW, SW, NE, SE) com a cor do agente (ou com outra cor), incluindo a própria casa, criando um rastro, resultando em mais 18 possíveis ações (ver Figura 2.3). Tem ainda a capacidade de retirar brilho nas mesmas posições, quer da vizinhança, quer da própria casa, totalizando mais 9 possíveis ações possíveis. Podemos então acrescentar às ações de deslocação estas ações que podem ser agrupadas da seguinte forma:

- Marcar a vizinhança e a posição atual com a cor do agente (9 Ações)
- Marcar a vizinhança e a posição atual com uma cor específica (9 Ações)
- Retirar o brilho da vizinhança e da posição atual (9 Ações)

Com estas informações, os agentes podem avaliar as condições do ambiente, como quais áreas são mais brilhantes, quais contêm determinados padrões de cor, ou até identificar arestas entre diferentes regiões de intensidade, e posteriormente tomar uma ação de acordo com essas condições.

3.2.3 Memória

Os agentes a implementar possuem uma pequena memória que permite definir estados dos agentes a serem determinados na altura de implementar os agentes pedidos no enunciado. O uso de memória deverá ser justificado e fazer parte da modelação de cada agente pedido no enunciado.

3.2.4 Tipos de Agentes

Como mencionado anteriormente, neste ambiente habitam vários tipos de agentes que são caracterizados por diferentes comportamentos. De modo a facilitar o trabalho, a implementação do comportamento de um dos agentes, *Thresholder*, já é fornecida. O comportamento deste agente pode ser caracterizado da seguinte forma:

- *Thresholder* — Este agente percorre a grelha de pixels da esquerda para a direita de cima para baixo por ordem. A cada passo verifica se o brilho do pixel está abaixo de um valor de limiar (*threshold*) e caso esteja coloca o pixel a preto.

Neste contexto, existem quatro tipos agentes (*VertiBlade*, *HorizontalEx*, *GlowTaker* e *EdgeTitan*) que tiram partido das percepções disponíveis e atuam em correspondência, num comportamento determinístico e previsível. Estes agentes terão de ser modelados e implementados e podem ser caracterizados da seguinte forma:

- *VertiBlade* — Este agente identifica transições abruptas de intensidade no eixo horizontal, permitindo detetar diferenças de luminosidade ao longo da imagem. Será capaz de destacar os arestas verticais presentes na imagem.
- *HorizontalEx* — Semelhante ao agente anterior, mas identifica mudanças ao longo do eixo vertical, destacando arestas horizontais.
- *GlowTaker* — Este agente navega pelas casas vizinhas com maior brilho, i.e. casa com maior valor de intensidade. Após se mover para a casa de maior intensidade, retira-lhe parte brilho.

- *EdgeTitan* — Este agente tem preferência em encontrar e navegar nas arestas presentes no ambiente, seguindo e analisando a sua continuidade. Como resultado do seu comportamento são reconhecidas formas presentes no ambiente.

Cada agente terá um ambiente próprio para mostrar o seu comportamento base (ver Figura 3).

4 Tarefas

O objetivo deste trabalho consiste no **desenvolvimento do sistema de produções** que modela o comportamento dos agentes indicados. Deverá começar por descarregar o projeto do *UCStudent* e abri-lo no Processing.

O projeto apresenta já uma implementação funcional do ambiente, seguindo uma abordagem orientada a objetos. Na pasta Scripts/Engine existem vários ficheiros .pde para analisar.

```

1 // Extrai a Cor da Posicao
2 color checkColor();
3 color checkColorNorth();
4 color checkColorSouth();
5 color checkColorEast();
6 color checkColorWest();
7 color checkColorNorthEast();
8 color checkColorNorthWest();
9 color checkColorSouthEast();
10 color checkColorSouthWest();

11
12 // Extrai o Brilho da Posicao
13 float checkBrightness();
14 float checkBrightnessNorth();
15 float checkBrightnessSouth();
16 float checkBrightnessEast();
17 float checkBrightnessWest();
18 float checkBrightnessNorthEast();
19 float checkBrightnessNorthWest();
20 float checkBrightnessSouthEast();
21 float checkBrightnessSouthWest();

22
23 // Verifica se uma Posicao esta marcada pelo Agente
24 boolean isMarked();
25 boolean isMarkedNorth();
26 boolean isMarkedSouth();
```

```

27 boolean isMarkedEast();
28 boolean isMarkedWest();
29 boolean isMarkedNorthEast();
30 boolean isMarkedNorthWest();
31 boolean isMarkedSouthEast();
32 boolean isMarkedSouthWest();

33
34 // Verifica se a Posicao esta dentro da Grelha
35 boolean checkBoundsNorth();
36 boolean checkBoundsSouth();
37 boolean checkBoundsEast();
38 boolean checkBoundsWest();
39 boolean checkBoundsNorthEast();
40 boolean checkBoundsNorthWest();
41 boolean checkBoundsSouthEast();
42 boolean checkBoundsSouthWest();

```

Listagem 1: Métodos disponibilizados que definem as percepções dos agentes.

```

1 // Vai para uma Posicao
2 void goNorth();
3 void goSouth();
4 void goEast();
5 void goWest();

6
7 // Marca com a Cor do Agente
8 void mark();
9 void markNorth();
10 void markSouth();
11 void markEast();
12 void markWest();
13 void markNorthEast();
14 void markNorthWest();
15 void markSouthEast();
16 void markSouthWest();

17
18 // Marca com uma Cor Especifica
19 void mark(color c);
20 void markNorth(color c);
21 void markSouth(color c);
22 void markEast(color c);
23 void markWest(color c);
24 void markNorthEast(color c);
25 void markNorthWest(color c);

```

```

26 void markSouthEast(color c);
27 void markSouthWest(color c);
28
29 // retira brilho
30 void subtractBrightness();
31 void subtractBrightnessNorth();
32 void subtractBrightnessSouth();
33 void subtractBrightnessEast();
34 void subtractBrightnessWest();
35 void subtractBrightnessNorthEast();
36 void subtractBrightnessNorthWest();
37 void subtractBrightnessSouthEast();
38 void subtractBrightnessSouthWest();

```

Listagem 2: Métodos disponibilizados que definem ações dos agentes.

4.1 Implementação e Teste dos Agentes

Para implementar um agente basta criar e implementar com ajuda dos métodos providenciados a função *behaviour* numa nova *class* que tem de extender a *class Agent*. A pasta dos scripts já providencia ficheiros .pde para cada um dos quatro agentes pedidos pelo enunciado, pelo que se deve preocupar em implementar a função *behaviour*.

```

1 class TemplateAgent extends Agent {
2
3     TemplateAgent(PVector pos, int size, color c) {
4         super(pos, size, c);
5     }
6
7     void behaviour() {
8         // Implementar o Comportamento do Agente
9     }
10}

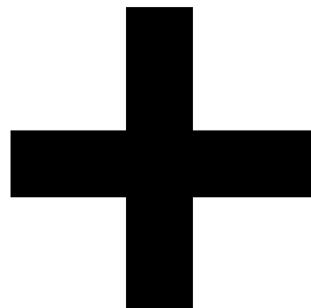
```

Listagem 3: Template Agent

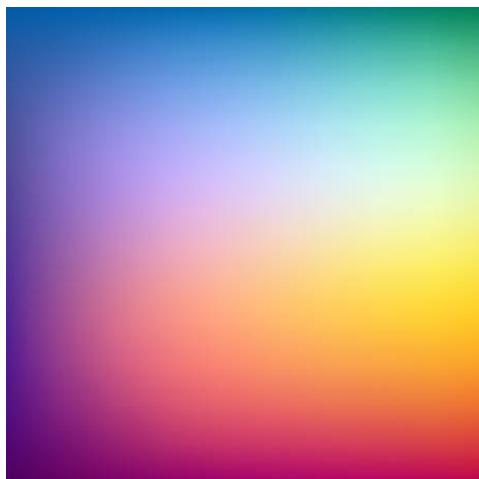
Depois da implementação, deverá testar os agentes em cada uma das imagens providenciadas na pasta *data/evalimages*. Deve ser feita uma análise do comportamento dos agentes em cada uma das imagens, assim como do resultado final. Cada agente **obrigatoriamente** terá de ser testado nos seus respetivos cenários (ver Figura 3).



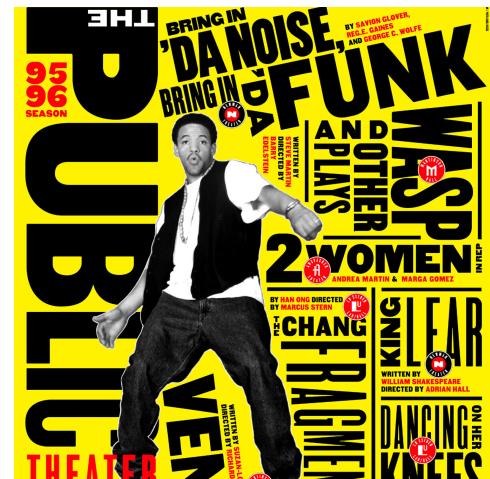
(a) *img1.png* para testar *HorizontalEx* e *VertiBlade*.



(b) *img2.png* para testar o *EdgeTitan*.



(c) *img5.png* para testar o *GlowTaker* e *Thresholder*.



(d) *img6.png* para testar o *GlowTaker* e *Thresholder*.

Figura 3: Os cenários de teste **obrigatórios** respetivos para cada agente.

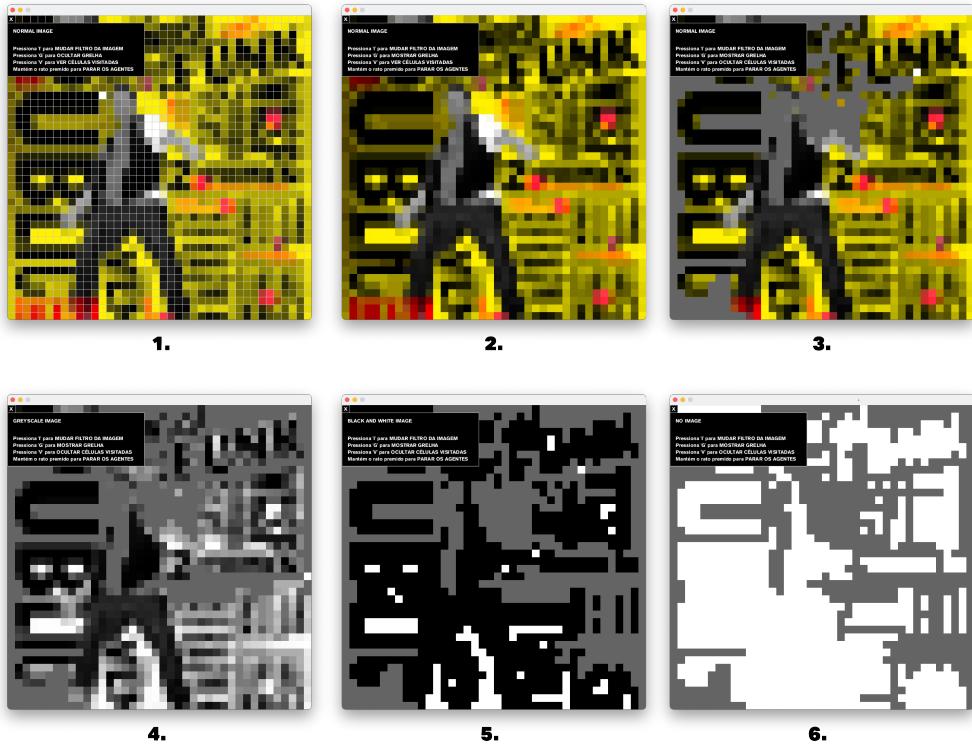


Figura 4: Interface do projeto com funcionalidades que disponibilizam informação extra sobre o agente e o ambiente.

5 Interface

Para facilitar a visualização do comportamento dos agentes, o projeto já inclui funcionalidades que fornecem informações adicionais sobre o agente e o ambiente (ver Figura 4). Existem vários tipos de transformações da grelha de base (imagem normal (Figura 4.1 — 4.3), imagem em tons de cinza (Figura 4.4), imagem binária a preto e branco (Figura 4.5) ou remoção da imagem (Figura 4.6). Estas transformações podem ser alteradas pressionando “I”. Além disso, existe também a possibilidade de visualizar (Figura 4.1) ou ocultar (Figura 4.2) a grelha que divide cada célula, pressionando “G”. O interface permite ainda a visualização das células visitadas pelos agentes (células a cinzento na Figura 4.6), pressionando “V”. **Nota Importante:** estas células são guardadas apenas para *debug* e, portanto, não devem ser utilizadas como percepções.

6 Extras

Após desenvolver e analisar o comportamento dos quatro agentes pretendidos (*VertiBlade*, *HorizontalEx*, *GlowTaker* e *EdgeTitan*) poderá criar outro(s) tipo(s) de agente(s). Assim como nos agentes obrigatórios, será necessário definir um conjunto de percepções e ações, bem como modelar o comportamento do seu agente reativo através de um sistema de produções. Serão valorizados agentes com comportamentos criativos e visualmente interessantes, desde que devidamente fundamentados.

7 Metas

O presente trabalho prático encontra-se dividido em 2 metas distintas:

Meta 1 – Modelação e desenvolvimento do Sistema de Produções

Nesta meta, deverá definir um conjunto de **percepções e ações**, bem como modelar o comportamento do seu agente reativo através de um **sistema de produções**.

Meta 2 – Implementação e Teste do Sistema de Produções em Processing

Deve **implementar as percepções e ações** definidas, bem como o sistema de produções no base de código fornecida em *processing*, **criando** assim os **diferentes agentes reativos** pedidos neste enunciado. Deverá **testar e avaliar o comportamento** esperado dos agentes **nos cenários** indicados na Figura 3.

8 Datas e Modo de Entrega

Os grupos têm uma dimensão máxima de 3 alunos. A defesa é obrigatória, bem como a presença de todos os elementos do grupo na mesma.

A entrega da meta 1 é opcional, chama-se no entanto a atenção dos alunos para a importância de concluir atempadamente esta meta. Para efeitos de nota apenas será considerada a entrega final e a defesa.

8.1 Meta 1 – Modelação e desenvolvimento do Sistema de Produções

Material a entregar:

- Um breve documento (max. 3 páginas), em formato pdf, com a seguinte informação:
 - Identificação dos elementos do grupo (Nomes, Números de Estudante, e-mails, Turma(s) Prática(s)).
 - Modelação dos comportamentos dos **quatro agentes** através de um sistema de produções.
 - Outra informação que considere pertinente relativamente a esta meta.

Modo de Entrega:

Entrega eletrónica através do Inforestudante.

Data Limite: 28 de Fevereiro de 2025

8.2 Meta 2 – Implementação e Teste do Sistema de Produções em Processing

Tal como indicado anteriormente, esta entrega será a única que tem um impacto direto na nota. O relatório deve conter informação relativa a **todo** o trabalho realizado. Ou seja, o trabalho realizado no âmbito das metas 1 e 2 deve ser **inteiramente descrito**, por forma a possibilitar a avaliação.

Material a entregar:

- Zip com o projeto, incluindo o código que implementaram e/ou alteraram, que deve estar devidamente comentado.
- Um relatório (max. 10 páginas), em formato pdf, com a seguinte informação:
 - Identificação dos elementos do grupo (Nomes, Números de Estudante, e-mails, Turma(s) Prática(s)).
 - Informação pertinente relativamente à globalidade do trabalho realizado. Incluindo os sistemas de produções, percepções e ações dos agentes.

Num trabalho desta natureza o relatório assume um papel importante. Deve ter o cuidado de descrever detalhadamente todas as funcionalidades implementadas, dando particular destaque aos problemas e soluções encontradas. Deve ser fácil ao leitor compreender o que foi feito e ter por isso capacidade de adaptar / modificar o código.

O relatório deve conter informação relevante tanto da perspetiva do utilizador como do programador. Não deve ultrapassar as 10 páginas, formato A4. Todas as opções tomadas deverão ser devidamente justificadas e explicadas.

Modo de Entrega:

Entrega eletrónica através do Inforestudante.

Data Limite: 14 de Março de 2025

9 Bibliografia

1. Inteligência Artificial: Fundamentos e Aplicações
Ernesto Costa, Anabela Simões
2. Artificial Intelligence: A Modern Approach
Stuart Russel, Peter Norvig