



FACULDADE DE CIÊNCIAS E TECNOLOGIA  
DA UNIVERSIDADE DE COIMBRA  
DEPARTAMENTO DE ENGENHARIA  
INFORMÁTICA

2015/2016  
2º Semestre

# INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

## TRABALHO PRÁTICO 1

### *BRAITENBERG VEHICLES*

Ana Inês Mesquita Fidalgo – 2013134819 – aimf@student.dei.uc.pt – PL6  
Andreia Filipa Palma Gonçalves – 2007018949 - andreiag@student.dei.uc.pt – PL4  
Pedro Filipe Matos Godinho Gabriel Coelho – 2009116949 - pfcoelho@student.dei.uc.pt  
– PL2

6 de março de 2016

# ÍNDICE

|   |   |
|---|---|
| Introdução .....                            | 3 |
| Veículos .....                              | 4 |
| • Inteligente .....                         | 4 |
| • Apanha-bolas.....                         | 4 |
| • Afasta-cilindros.....                     | 5 |
| • Vai e Repele Próximos.....                | 5 |
| • Vai e Repele Afastados.....               | 5 |
| • Tolo.....                                 | 5 |
| • Oito.....                                 | 6 |
| • Elipse .....                              | 6 |
| Implementação das funções de activação..... | 7 |
| Limites e limiares.....                     | 8 |
| Conclusão.....                              | 9 |

# INTRODUÇÃO

No âmbito da unidade curricular de Introdução à Inteligência Artificial foi-nos proposta a realização do primeiro trabalho prático sobre agentes reactivos. Através da implementação de veículos de Braitenberg, deveremos adquirir e consolidar os conceitos de agentes reactivos, ambientes, percepções e acções.

O trabalho foi realizado no motor de jogos Unity com recurso à linguagem de programação C#. A partir do ficheiro fornecidos pelos docentes, no qual o veículo apenas tinha um tipo de sensor – fotosensor, foram implementados novos, com diversos comportamentos. No presente relatório são descritas as funcionalidades implementadas, os veículos e seus comportamentos.

## VEÍCULOS

O primeiro veículo, fornecido pelos docentes, apenas tinha um tipo de sensor – fotosensor – e um tipo de objecto – fonte de luz. A partir deste, foram criados vários cenários, nos quais outros veículos têm sensores que detectam luz e proximidade, respondendo, respectivamente, a fontes de luz e obstáculos existentes no ambiente. Assim, foram implementados os seguintes tipos de veículo:

- Inteligente;
- Apanha-bolas;
- Afasta-cilindros;
- Vai e Repele Próximos;
- Vai e Repele Afastados;
- Tolo;
- Oito;
- Eclipse.

### Inteligente

Este veículo desvia-se de todos os objectos existentes no ambiente: luzes e blocos.

No script *CarBehaviourBlocoScript* são obtidos os valores dos quatro sensores: dois do tipo *WallDetector* e dois *LightDetector*, respectivamente, detectam blocos e fontes de luz. De seguida, o valor da velocidade das rodas de cada lado será a multiplicação do *MaxSpeed* com a soma dos respectivos sensores esquerdos e direitos.

Os sensores *WallDetector* funcionam através da obtenção das posições de todos os cubos, sendo que depois é calculada a distância entre o veículo e cada um deles. Quanto aos sensores *LightDetector*, aplicou-se a mesma lógica, isto é, calculam-se as distâncias entre o veículo e a fonte de luz.

### Apanha-bolas

No ambiente encontram-se várias esferas, sendo que uma delas pode ser controlada pelo utilizador através do teclado. Os sensores do veículo detectam as esferas e ele vai contra elas empurrando-as segundo uma lógica física.

Em *CarBehaviourJogoScript* temos dois sensores *WallDetector2Script*. Se o valor do output do sensor esquerdo for superior ao direito, irá aplicar à roda esquerda a multiplicação do sensor direito com *MaxSpeed* e à direita, a multiplicação do sensor esquerdo com *MaxSpeed* e  $3/2$ . Caso contrário, atribui à roda esquerda a multiplicação do sensor direito com  $3/2$  e *MaxSpeed*.

Os sensores conseguem procurar as esferas, da mesma forma que localizam os cubos, ou seja, com base nas distâncias. O facto de termos uma esfera que se move consoante as indicações do utilizador, em nada interfere com o comportamento do veículo, ele detecta-a da mesma forma que as outras e vai contra ela.

## Afasta-cilindros

Para além de esferas, o ambiente também tem cilindros. O veículo deve afastar-se destes e ir até às esferas. Novamente, existe uma esfera diferente controlada pelo utilizador através de teclado.

Este script tem dois sensores *WallDetector* que pesquisam os objectos do ambiente que tenham a *tag* “dominó”. Os outros sensores *WallDetector2Script* vão procurar todos os objectos com a *tag* “esferas”.

## Vai e Repele Próximos

No ambiente apenas existe um tipo de objectos - fontes de luz. Os sensores do veículo detectam a que se encontra mais próxima. Ao aproximar-se dela, desvia-se novamente.

A implementação está no script *CarBehaviourFar*, no qual existem dois sensores *LightDetectorFarScript*. Este script é semelhante ao *LightDetector*, no entanto, depois de calculadas as distâncias, ele dirige-se para a fonte de luz mais próxima, afastando-se segundo a mesma lógica do veículo já fornecido pelos docentes.

## Vai e Repele Afastados

Este veículo encontra-se no mesmo ambiente que o anterior. Também detecta as luzes e depois repele-as, mas ao contrário do primeiro, procura a que se encontra mais afastada.

Apesar de todos os esforços, não conseguimos concretizar o comportamento deste veículo.

## Tolo

Novamente com apenas fontes de luz no ambiente, ao detectar uma, este veículo dirige-se para ela, mas ao contrário dos anteriores, não a repele e fica a girar eternamente no local.

O script *CarBehaviourVai* é semelhante ao *CarBehaviourJogoScript*, no entanto, os sensores são do tipo *LightDetectorScript*. Quanto aos cálculos, caso o output do sensor direito seja superior ao esquerdo, o valor da velocidade da roda esquerda será a multiplicação de duas vezes o *MaxSpeed* pelo sensor direito e, na roda direita, será aplicado o resultado da multiplicação do sensor esquerdo por *MaxSpeed*. Caso seja inferior, fará o contrário, o dobro da multiplicação do sensor esquerdo pela variável será o valor atribuído à roda direita.

## Oito

Com apenas duas fontes de luz existentes no ambiente, o veículo contorna-as, num circuito em forma de oito.

Em *CarBehaviourToLight*, os sensores são *LightDetectorScript*. O valor da velocidade da roda direita é a multiplicação de *MaxSpeed* com o sensor esquerdo subtraindo-se o output do sensor direito. Na roda esquerda, multiplica-se o sensor direito e retira-se o valor do output do sensor esquerdo.

Depois de algumas tentativas falhadas a tentar elaborar este veículo, pesquisámos sobre o trabalho de Braitenberg e descobrimos que cada sensor pode estar ligado às duas rodas, logo foi a partir desta possibilidade que implementámos este comportamento.

## Elipse

No mesmo ambiente com apenas duas fontes de luz, este veículo deverá ficar no meio das duas a rodar.

No script *roda*, os sensores são do tipo *LightDetectorScript*. Às rodas são aplicados os valores da multiplicação de *MaxSpeed* pelo sensor do mesmo lado.

## IMPLEMENTAÇÃO DAS FUNÇÕES DE ACTIVAÇÃO

No enunciado, foi-nos pedido que aplicássemos funções de activação ao cálculo de output dos sensores de todos os veículos, das quais já nos foi fornecida uma função linear. Foi-nos ainda pedida, a implementação de uma função de activação gaussiana, sobre a qual pesquisámos a sua fórmula:

$$f(x) = a \cdot e^{-\frac{(x-b)^2}{2c^2}}$$

Esta permite que o output dependa das seguintes variáveis: y máximo pretendido, média da energia das fontes de luz que incidem nos sensores e o desvio padrão. Os valores foram obtidos através de tentativa-erro, de modo a obter o comportamento esperado do veículo.

## LIMITES E LIMIARES

Para além das funções de activação, os docentes pediram-nos a definição de limites na distancia máxima e mínima a ser vista e na energia máxima e mínima recebida pelos sensores de luz.

Isto foi definido através de variáveis chamadas limite máximo, limite mínimo, limiar máximo e limiar mínimo. Servem para que na aplicação da função gaussiana, dada a forma em “sino”, os valores de output nunca cheguem a zero de modo a não anularem a velocidade do veículo.



## CONCLUSÃO

Com a realização deste trabalho prático, conseguimos consolidar os conhecimentos gerais sobre agentes reactivos e, especificamente, sobre o trabalho desenvolvido por Valentino Braitenberg.

Quanto ao Unity, foram sentidas algumas dificuldades no início, pois nenhum dos elementos do grupo tinha trabalhado anteriormente com esta ferramenta. Com o desenrolar do trabalho, a utilização deste motor de jogos foi-se tornando mais fácil e intuitiva, sendo que podemos considerar que a aquisição de aptidões foi bem sucedida.

Relativamente à implementação dos diversos veículos, obviamente, uns foram implementados rapidamente, com código-fonte bastante simples, sendo que noutros sentimos algumas dificuldades, não conseguindo compreender de que forma deveríamos modificar os sensores, para que o veículo tivesse o comportamento compreendido. Apesar de não termos conseguido implementar algumas funcionalidades que pensámos inicialmente, acabámos por desenvolver outras que surgiram ao longo do trabalho.

Em suma, consideramos que conseguimos atingir a maioria dos objectivos propostos pelos docentes.