



FCTUC FACULDADE DE CIÊNCIAS  
E TECNOLOGIA  
UNIVERSIDADE DE COIMBRA

# Projecto de Introdução à Inteligência Artificial

*2017 - 2018*

## - META II -

---

Leonardo Machado Alves Vieira - 2015236155  
Pedro Filipe Gomes Ramos de Carvalho - 2015232484  
Tiago Miguel Vitorino Simões Gomes - 2015238615

# Explicação dos Scripts

Em anexo encontram-se os scripts desenvolvidos e utilizados para o projecto e respectivas experiências.

## - ***CarBehaviour.cs***

Contém toda a lógica de comportamento do carro, nomeadamente a forma como o carro está configurado para responder ao que é detectado pelos sensores.

É possível associar um qualquer número de sensores ao carro através de um array do tamanho desejado, sendo também possível configurar todos esses sensores individualmente (ligação à roda, limiares, limites, função de ativação, excitação ou inibição). Todas estas definições são acessíveis pelo editor do Unity, pelo que se consegue alterar todo o comportamento do carro sem ter que fazer alguma alteração ao script.

Em todos os instantes, o script lê as configurações de todos os sensores, calculando o output de cada e fazendo uma média dos outputs de cada roda, permitindo assim que nunca se ultrapasse o limite de velocidade imposto pela interface do Unity (variável *maxSpeed*).

## - ***DetectorScript.cs***

Super classe de sensores que define os comportamentos transversais aos dois tipos de sensor, nomeadamente a pesquisa de objectos (feita sempre em relação ao centro destes) e as funções de ativação (linear e gaussiana). É também aqui que é definida a forma como as configurações mencionadas acima afectam os outputs. O ângulo de pesquisa pode ser alterado através do editor do Unity.

## - ***LightDetectorScript.cs*** e ***ProximityDetectorScript.cs***

Classes descendentes de *DetectorScript* que definem os comportamentos específicos de cada tipo de sensor, como quais os objectos a ter em conta e qual a fórmula de força (energia do sensor). No caso da luz é seguida a fórmula do enunciado, enquanto no caso dos blocos é usada a fórmula  $S = 1 \div (distance + 1)$ .

## - ***ObjectMovementScript.cs*** e ***BlockDeletionScript.cs***

Dois scripts auxiliares a testes. O primeiro é um script simples, a que se pode associar um GameObject para que este seja movimentado com o teclado (setas ou WASD). É necessário para o teste #7. O segundo pode ser adicionado a blocos com um *Collider* “*Is Trigger*”, fazendo com que estes sejam destruídos quando tocados (teste #8).

**Nota:** Alguns detalhes da estrutura do código estão mais explícitos em comentários nos devidos ficheiros.

## Algumas notas acerca da inibição e excitação:

- Os sensores excitadores seguem os gráficos *Strength - Output* apresentados no enunciado.
- Para ter veículos com os comportamentos *lover* e *explorer* são necessários comportamentos de inibição. Assim, criámos dois tipos de inibição, a que chamamos *Pure Inhibitory* e *Altered Inhibitory*.

- O comportamento *Pure Inhibitory* é obtido através de uma simples inversão vertical dos gráficos de excitação do enunciado, isto é:  $I_{output} = 1 - E_{output}$ .
- *Altered Inhibitory* é obtido invertendo-se o output antes da aplicação de limites e limiares. Isto permite, por exemplo, ter limiares de ativação que desativam o movimento do veículo, o que pode ser útil em algumas circunstâncias. Este comportamento é representado nos gráficos da *Figura 1*.

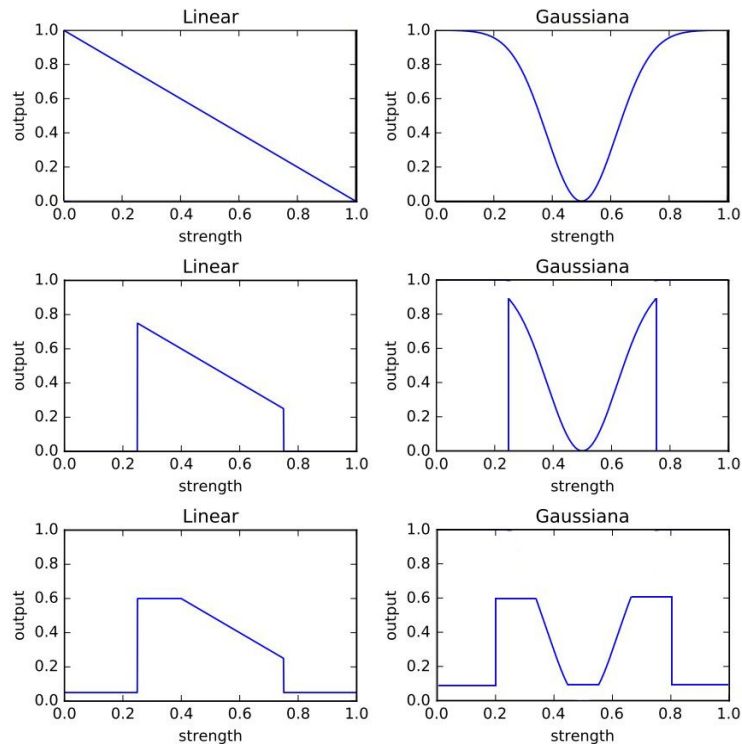


Figura 1 - Gráficos Strength-Output para o comportamento *Altered Inhibitory*

## Testes

A nossa abordagem para os testes consistiu primeiramente na idealização e debate sobre os cenários que iríamos realizar e testar. Depois disso, debatemos sobre os resultados esperados para cada cenário. Por fim, construímos e testamos os mesmos, comparando os resultados obtidos com os esperados, procurando sempre explicação para os comportamentos dos carros.

### Teste #1 - 4 carros lineares

- **Cenário desenvolvido**

Foram feitos testes com o intuito de testar os 4 tipos básicos de carros (medroso, agressivo, apaixonado e explorador). Foram colocados dois sensores de luz nos veículos (ambos à frente, mas um de cada lado). O output dos veículos segue uma função linear, sem limiares ou limites, tendo os seus sensores um ângulo de visão de 180 graus para os dois primeiros carros e 270 graus para os outros dois. Foram criados dois cenários separados no *Unity*, um para um carro medroso e um agressivo, e outro para os restantes. Os dois cenários são visualmente semelhantes, sendo apenas compostos por uma luz central e com os carros virados para a mesma.

- **Resultados esperados**

Para cada carro é esperado o seguinte comportamento:

**Medroso** - É expectável que este carro se desvie da luz com uma velocidade progressivamente maior até que se desvie o suficiente para a luz sair do seu ângulo de visão, parando depois disto. Isto acontece porque o sensor do lado esquerdo e direito estão ligados à roda esquerda e direita respectivamente, seguindo uma função linear excitatória. Assim sendo quando o carro se aproxima da fonte de luz, a roda mais próxima irá ganhar mais velocidade do que a que se encontra mais longe, fazendo com que o carro se desvie.

**Agressivo** - De maneira semelhante ao carro anterior, no agressivo é suposto que a velocidade aumente progressivamente em direção à fonte de luz, parando também após esta sair do seu ângulo de visão. Isto acontece porque, contrariamente ao carro medroso, os sensores da esquerda e direita estão ligadas às rodas opostas. Como o sensor mais perto da luz tem mais energia do que o sensor mais longe, a roda mais afastada irá ter mais velocidade provocando uma trajetória de colisão com a fonte.

**Apaixonado** - Este carro, ao contrário dos dois descritos previamente, segue um comportamento de inibição. Na prática, estes sensores comportam-se como travões, diminuindo a velocidade das rodas associadas à medida que estes se aproximam da fonte. Isto proporciona um comportamento em que o carro se desloca em direção da luz, andando cada vez mais devagar à medida que se aproxima. Tal acontece pois os sensores estão ligados às rodas de maneira idêntica ao medroso, fazendo com que a roda mais próxima perca velocidade à medida que nos aproximamos da fonte. De notar que o carro apenas para permanentemente ao pé da luz caso seja utilizado o modo de inibição alterada e com limiar mínimo de activação. Caso se use o modo de inibição pura, depois do carro passar pela luz os sensores terão o valor mínimo de energia, que corresponde ao maior *output* de velocidade possível.

**Explorador** - Por último, o explorador utiliza sensores iguais aos do apaixonado, mas com uma ligação às rodas semelhante ao agressivo. Por esta razão, prevê-se que este carro se desvie da luz e graças ao seu ângulo de visão dê uma breve curva circundando a mesma.

- **Resultados obtidos**

Resultados obtidos para o explorador e apaixonado encontram-se representados na figura 2 e os resultados para o medroso e agressivo encontram-se na figura 3.



Figura 2

Figura 3

## Teste #2 - 4 carros gaussianos

- **Cenário desenvolvido**

Foram feitos testes com o intuito de testar novamente os 4 tipos básicos de carros, sendo que desta vez se colocou sensores de proximidade onde havia sensores de luz, para além de o output seguir agora uma função gaussiana com ângulo de visão de 270 graus em todos os sensores. Foram criados dois cenários separados no *Unity*, um para um carro medroso e um agressivo, e outro para os restantes. Os dois cenários são visualmente semelhantes, sendo apenas compostos por um bloco central e dois carros virados para este.

- **Resultados esperados**

Para cada carro é esperado o seguinte comportamento:

**Medroso** - Tem um comportamento semelhante ao medroso linear referido acima. Utiliza nos dois sensores uma média de 0.25 e um desvio padrão de 0.08. Este carro desvia-se do objecto, no entanto, por não chegar ao lugar onde a força é igual à média, a velocidade nunca é máxima. Também será de notar que o carro nunca pára completamente porque a função gaussiana nunca tem o valor de 0

**Agressivo** - Tem um comportamento semelhante ao agressivo linear. No entanto, por ser gaussiano, com uma média de 0.2 e um desvio padrão de 0.06, ao chegar perto do bloco, ocorre uma curva brusca. Isto acontece porque o sensor mais próximo ultrapassa a média de força da função gaussiana fazendo com que a roda mais afastada (direita) perca velocidade enquanto a roda associada ao sensor mais afastado ainda se encontra no pico de output da função.

**Apaixonado** - Assim como o apaixonado linear, este carro dirige-se ao bloco central diminuindo a velocidade à medida que se aproxima. No entanto, sendo gaussiano com uma média de 0.3 e um desvio padrão de 0.12, este carro irá parar a alguma distância do bloco. Isto acontece pois a função gaussiana inibida atinge o 0 quando o valor da força recebida pelos sensores iguala o valor da média da função.

**Explorador** - Este último carro tem um comportamento muito semelhante à sua “versão linear” desviando-se da luz e curvando um pouco em torno da mesma. Destaca-se a diferença de velocidades nas duas versões devido à média de 0.25 e ao desvio padrão de 0.08 da função gaussiana, que faz com que o pico de output se observe quando a força dos sensores é de 0.25.

- **Resultados obtidos**

Resultados obtidos para o explorador e apaixonado encontram-se representados na figura 4 e os resultados para o medroso e agressivo encontram-se na figura 5.

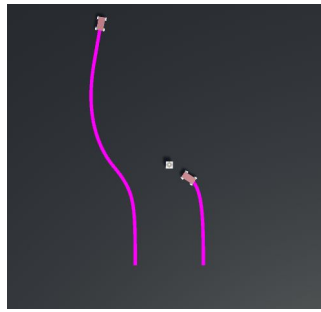


Figura 4

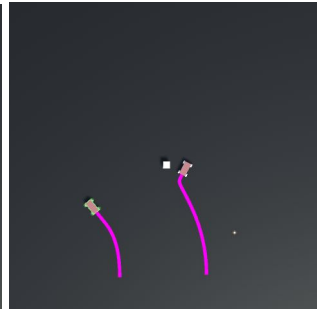


Figura 5

### Teste #3 - Inibição pura e inibição alterada

- **Cenário desenvolvido**

Foi desenvolvido no *Unity* um cenário com o intuito de comparar as duas funções de inibição criadas para o projecto. Como tal, idealizou-se um ambiente simples semelhante aos acima descritos. Uma luz no centro e dois carros virados para ela, cada um com um dos comportamentos inibidores. Como as diferenças das duas funções só são notadas na presença de limites e limiares, configurámos cada carro para ser uma adaptação de um carro explorador gaussiano, com limites de  $0.1$  e  $0.8$  e limiares de  $0.2$  e  $0.7$ .

- **Resultados esperados**

É esperado que os carros mostrem comportamentos completamente diferentes. Ambos os sensores começam abaixo do limiar de ativação, pelo que o carro inibidor puro irá apresentar um output inverso ao limite mínimo ( $1 - 0.1 = 0.9$ ), enquanto que o alterado apresentará, efectivamente, o output mínimo de  $0.1$ . Ao aproximarem-se da luz, ambos estarão fora dos limiares. O inibidor puro irá reduzir ligeiramente a velocidade e fazer a habitual viragem brusca (descrita noutros testes), sendo que depois irá voltar a estar abaixo do limiar mínimo, retomando o comportamento anterior. Já o inibidor alterado, estará limitado a  $0.8$ , pelo que irá manter esse output constante após a viragem brusca, estabilizando em volta da luz.

- **Resultados obtidos**

A figura 6 mostra os resultados obtidos (puro a verde, alterado a roxo)

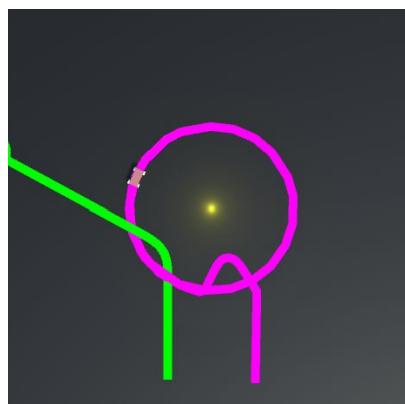


Figura 6

## Teste #4 - Caminho de blocos

- **Cenário desenvolvido**

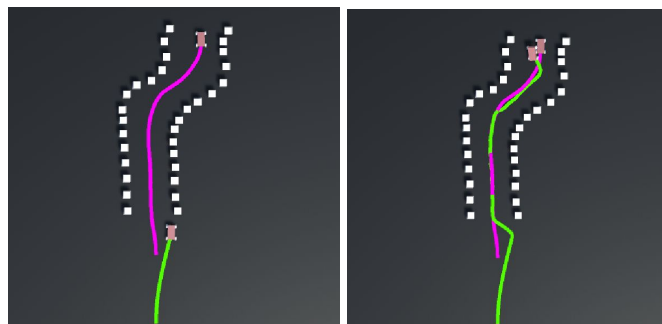
Foi desenvolvido um cenário com o objectivo de testar o comportamento de 2 veículos com sensores de proximidade quando deparados com um caminho rodeado de blocos. Ambos têm os sensores na mesma posição que nos testes anteriores com ângulo de visão de 270 graus, sendo que o primeiro veículo tem um comportamento medroso linear, enquanto que o outro tem um comportamento agressivo gaussiano, com uma média de 0.2 e desvio 0.05. O caminho de obstáculos está representado na *figura 7*.

- **Resultado esperado**

Como o primeiro carro tem um comportamento medroso, esperava-se que percorresse o interior do percurso evitando o contacto com as paredes, pois estaria constantemente a afastar-se dos blocos. Já no caso do segundo carro, o comportamento agressivo terá as mesmas características que as explicadas no segundo teste, pelo que fará curvas bruscas assim que chega perto de um bloco. Após cada uma destas curvas, passará a estar mais próximo de outro bloco, ao qual se vai aproximar e voltar a fazer uma curva, mantendo-se, assim, sempre no centro do percurso, ainda que com um comportamento mais errático.

- **Resultado obtido**

O resultado obtido encontra-se representado nas *figuras 7 e 8*, tendo o carro linear uma risca roxa e o gaussiano uma risca verde.



*Figura 7*

*Figura 8*

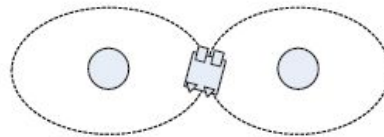
## Teste #5 - Forma de um oito

- **Cenário desenvolvido**

Foi desenvolvido um cenário com o objectivo de fazer com que um veículo faça um percurso semelhante a um oito, tendo duas fontes de luz no interior dos círculos do oito. Assim, usou-se um veículo agressivo gaussiano com sensores de luz posicionados como nos outros testes e com ângulo de visão de 360. Após muita experimentação chegou-se à conclusão que os valores de média 0.35 e desvio-padrão 0.13, com limite inferior de 0.1 e superior de 0.4, seriam os ideais para esta experiência, para os dois sensores.

- **Resultado esperado**

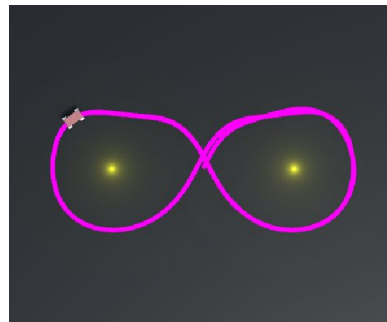
O resultado esperado era que tivesse um comportamento similar ao representado no enunciado (*figura 9*). Sendo que um oito é uma figura simétrica em que ambas as rodas tomam a posição de roda interior em diferentes alturas, era importante garantir total igualdade de parâmetros entre os sensores, para que se comportem de forma igual em situações simétricas. O comportamento gaussiano levaria o veículo a manter uma certa distância da luz ainda que tivesse uma clara atração à mesma. Ao fazer o contorno completo de uma das luzes, aproximar-se-ia da outra luz, que passaria a estar perto o suficiente para causar atração necessária para o veículo passar a contorná-la, variando assim sucessivamente entre as luzes.



*Figura 9*

- **Resultado obtido**

O resultado obtido encontra-se representado na *figura 10*.



*Figura 10*

## Teste #6 - Elipse

- **Cenário desenvolvido**

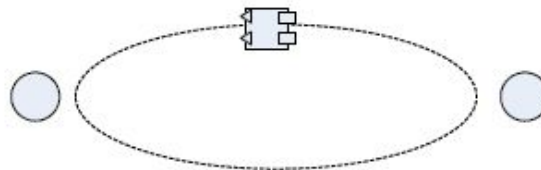
Outro dos objectivos do trabalho era criar um carro que exibisse um comportamento elíptico na área interior de duas fontes de luz. Para este fim, optamos por utilizar um carro Agressivo (pois queremos que o carro se aproxime das luzes) com sensores gaussianos (para podermos mais facilmente ajustar os detalhes do comportamento do agente). De forma a reproduzir o comportamento desejado é necessário que o agente alterne entre as luzes que determinam a maior parte do seu comportamento, isto levou-nos a limitar o ângulo de visão dos sensores para 270°. Por fim, há dois detalhes deste comportamento que levantam algumas dificuldades, a elipse é uma forma simétrica, isto sugere que devemos utilizar um comportamento igual em ambas as rodas; no entanto a curva apertada que ocorre nos extremos da forma sugere uma configuração assimétrica. A solução que encontramos para este problema foi a escolha de uma configuração igual em ambos os sensores ( $\mu =$



0.4;  $\sigma = 0.1175$ ), exceto um limite superior de velocidade na roda de dentro (0.5, metade da velocidade máxima da roda externa). Um detalhe final que implementamos foi o reposicionamento dos sensores para o centro do veículo.

- **Resultado esperado**

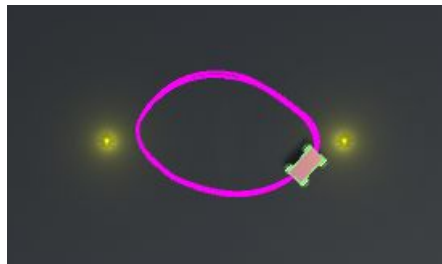
O resultado esperado era que tivesse um comportamento similar ao representado no enunciado (*figura 11*). A configuração gaussiana dos sensores faria com que o veículo procurasse manter-se a uma determinada distância da luz. Ao limitarmos a velocidade da roda interior conseguiríamos forçar uma curva mais apertada que levaria o veículo a focar-se na luz mais distante. Espera-se também que o posicionamento central dos sensores facilite este processo de “alteração de foco”. Este comportamento iria repetir-se nas duas luzes de forma simétrica, o que deveria levar o veículo a deslocar-se na forma de uma elipse.



*Figura 11*

- **Resultado obtido**

O resultado obtido encontra-se representado na *figura 12*.



*Figura 12*

## Teste #7 - Luz que move (com obstáculos)

- **Cenário desenvolvido**

Este cenário foi criado com o objetivo de demonstrar a adaptabilidade que os veículos têm em relação à alteração das condições do mundo em tempo real. Criou-se um veículo com 4 sensores (2 de luz e os outros 2 de proximidade colocados à frente, um de cada lado por tipo) com comportamento linearmente agressivo à luz e gaussianamente medroso aos blocos ( $\mu = 0.5$ ;  $\sigma = 0.12$ ), com comportamento excitador e sem qualquer tipo de limites ou limiares. O cenário contém vários blocos e uma luz que pode ser movimentada com o teclado.

- **Resultado esperado**

O resultado esperado era que o veículo seguisse constantemente a luz, mesmo que esta se movimente, ao mesmo tempo que evitava colisões com blocos.

- **Resultado obtido**

O resultado obtido encontra-se representado na *figura 13*, estando o percurso da luz representado por um amarelo translúcido.



*Figura 13*

## Teste #8 - Devorador de blocos

- **Cenário desenvolvido**

Neste cenário o objectivo era criar um veículo que tenta “devorar” todos os blocos existentes. Para isso, tem dois sensores de proximidade configurados de forma linearmente agressiva, com limites mínimos desiguais (0.1 no sensor esquerdo, 0.05 no direito).

- **Resultado esperado**

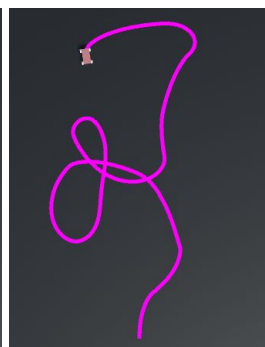
A desigualdade nos limites mínimos faz com que o carro rode no sentido contrário aos ponteiros do relógio enquanto não encontra nenhum bloco suficientemente próximo. Assim, terá tendência a acabar por encontrar todos os blocos.

- **Resultado obtido**

O resultado obtido encontra-se representado nas *figuras 14 e 15*.



*Figura 14 - Antes*



*Figura 15 - Depois*