

Projeto realizado por:

- *Mário Lemos N.º 2019216792*
- *Pedro Martins N.º 2019216826*
- *Regina Amado N.º 2019210521*

## Meta 1 (Sense It):

Cenário de teste:

**Cenário A:** Neste cenário temos um veículo de Braitenberg e uma fonte de iluminação diretamente à sua frente. O carro é dotado de dois sensores de deteção de luz (um esquerdo e um direito).

**Cenário B:** O cenário de teste utilizado consiste de dois veículos de Braitenberg sendo o da frente referido neste documento como **carro X** e o de trás como **carro Y**. Na cena existem diversas fontes de iluminação que formam um círculo ao longo do terreno. **X** é dotado de dois sensores de iluminação (1 à esquerda e 1 à direita) e **Y** é dotado de dois sensores de proximidade (cujo código para deteção de objetos não se encontra desenvolvido inicialmente).

Alterações ao(s) script(s):

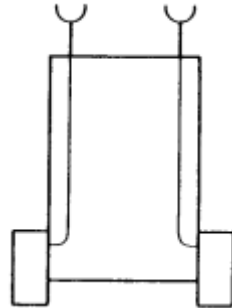
Nesta meta procedeu-se essencialmente ao desenvolvimento do código para detetar o objeto mais próximo com base em sensores de proximidade. Com este intuito procedemos a criar os seguintes aspetos do ficheiro CarDetectorScript:

- **GetVisibleCars:** Esta função devolve uma lista contendo todos os os carros que se encontrem dentro do ângulo de visão dos sensores de proximidade do carro e que possuam uma TAG pré-definida (“CarToFollow” no exemplo do projeto).
- **Update:** Dentro desta função procedemos ao cálculo do valor do output a ser depois interpretado pelo carro. Para isso, foi determinado qual o carro da lista fornecida pela função acima se encontra mais próximo dos sensores do **Carro Y**, o que leva o nosso carro a reagir de acordo com X.

Após implementadas estas duas funções, o script para detetar o objeto mais próximo está pronto.

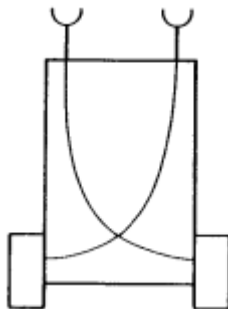
## Testes efetuados (Experimentação):

De forma a conseguir um bom conhecimento sobre o funcionamento dos sensores e como eles interagem com o meio que o rodeia. Começámos por realizar testes no veículo presente no **Cenário A**. Sem fazer qualquer alteração, o carro aproximá-va-se da fonte de iluminação e, quando se encontrava perto, fazia uma curva drástica para a esquerda.



**Figura 1** – Veículo de Braitenberg com dois sensores ligados às suas rodas

A **Figura 1** representa a configuração inicial do veículo, e permite explicar o porquê do acima descrito acontecer. Um dos sensores estará mais estimulado que o outro, o que origina uma maior output e, consequentemente, maior velocidade da roda conectada ao sensor. Tudo isto leva a uma curva forte no sentido contrário à posição do sensor no carro. Assim, o novo objetivo é tornar o nosso veículo mais agressivo, com o intuito de testar os seus limites seguiu-se a seguinte abordagem:



**Figura 2** – Veículo de Braitenberg com dois sensores ligados a rodas opostas.

A **Figura 2** constitui a nova configuração testada para a nosso veículo, cada sensor está conectado a uma roda oposta a si. Assim, quando um sensor é mais estimulado, a velocidade irá aumentar na roda oposta a este, o que origina uma curva na direção do sensor e, consequentemente, da fonte de iluminação.

De seguida, e após o conhecimento adquirido nos testes passados, começámos a trabalhar no **Cenário B**. Numa primeira iteração, foi notado que no cenário de testes, o **carro Y** deslocase muito ligeiramente em reação à aceleração por parte do **carro X**, rapidamente perdendo o mesmo e parando. Como primeira medida de teste, procedemos à troca dos sensores de

proximidade da esquerda e da direita entre si, o que originou uma reação mais agressiva por parte do **carro Y** contudo, também nesta sequência de testes ficou para trás bastante rápido.

Para aumentar ainda mais a reação de **Y**, aumentámos o seu parâmetro de velocidade, testando com múltiplos valores, ordenados pela seguinte sequência. Todos os tempos obtidos representam, **pelo menos**, 3 testes, de forma a garantir a fiabilidade dos mesmos:

1. **Velocidade 5 (Velocidade Original)**
2. **Velocidade 10:** O **carro Y** continua a rapidamente perder o contacto com **X** e, consequentemente, a parar.
3. **Velocidade 25:** O **carro Y** é capaz de acompanhar **X** durante cerca de menos de 1 segundos, perdendo-o mais uma vez de vista após este período de tempo elapsar.
4. **Velocidade 50:** O **carro Y** acompanha durante um bom bocado **X**, contudo após este realizar a primeira curva, o **carro Y** perde o contacto próximo e prossegue numa trajetória errática.
5. **Velocidade 75:** O **carro Y** é agora bastante mais rápido que **X**, o que o leva a embater no último após cerca de 2 segundos de testes.
6. **Velocidade 60:** Esta foi a melhor velocidade encontrada e que garante que o **carro Y** consiga perseguir **X** durante um período de tempo aceitável (3-4 segundos).

Como última medida de teste, procedemos a alterar o valor devolvido pela função **getOutput** no script **CarDetectorLinearScript.cs** de forma a devolver como intensidade o inverso do output ( $1 - output$ ); assim é possível ao **carro Y**, quando se encontra afastado, aumentar a sua velocidade para se aproximar de **X** e quando se encontra bastante próximo diminuir a velocidade, evitando assim qualquer colisão. Isto permite-nos ter um maior controlo sob as ações de **Y**. A velocidade foi reduzida para valores menores (5 unidades). Numa primeira instância, era notório que com a última configuração dos sensores (sensor esquerdo ligado às rodas direitas e sensor direito ligado às rodas esquerdas) o **carro Y** aproximava-se do da frente e realizava uma curva abrupta para se afastar dele. Ora, se o output obtido é 1 – o output calculado, isso quer dizer que o sensor mais estimulado vai devolver uma intensidade menor às rodas conectadas, o que origina o desvio abrupto. Assim, de forma a resolver este problema voltámos a ligar os sensores às rodas do mesmo lado.

O comportamento final descrito demonstra a capacidade de um carro seguir o outro com base nos seus sensores de proximidade. Na versão presente à data de entrega deste relatório, o carro de trás é capaz de seguir o da frente, realizar uma curva com ele, seguindo-o de forma controlada durante cerca de 10 segundos. Após perder contacto aumenta a sua velocidade de forma a reencontrá-lo e, quando se aproxima novamente, diminui a sua velocidade de forma a tentar-se aproximar.

## Meta 2 (Tune It & Test It):

Cenário de teste:

**Cenário A:** Encontra-se presente um veículo de Braintenberg dotado de dois sensores de detecção de luz posicionado no centro do cenário. Existe uma fonte de iluminação diretamente em frente ao carro e o objetivo deste cenário é, através do uso de um *script* de detecção de luz gaussiano, colocar o veículo a fazer um círculo à volta da luz em questão.

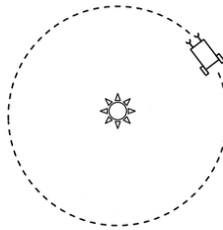


Figura 3 – Veículo de Braintenberg a realizar a trajetória descrita no **Cenário A**

**Cenário B:** Encontra-se presente um veículo de Braintenberg dotado de dois sensores de detecção de luz posicionado no centro do cenário. Existem duas fontes de iluminação próximas do carro e o objetivo deste cenário é, através do uso de um *script* de detecção de luz gaussiano, colocar o veículo a fazer o infinito à volta das duas luzes.

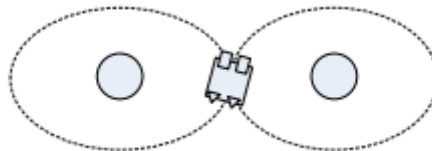


Figura 4 – Veículo de Braintenberg a realizar a trajetória descrita no **Cenário B**

**Cenário C:** Encontra-se presente um veículo de Braintenberg dotado de dois sensores de detecção de luz posicionado no centro do cenário. Existem duas fontes de iluminação próximas (um pouco mais afastadas entre si comparativamente ao **Cenário B**) e o objetivo deste cenário é, através do uso de um *script* de detecção de luz gaussiano, colocar o veículo a fazer uma elipse entre as duas luzes.

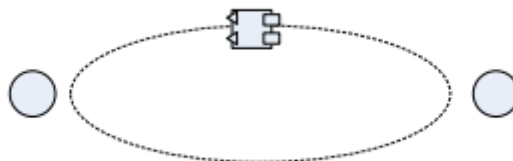


Figura 5 – Veículo de Braintenberg a realizar a trajetória descrita no **Cenário C**

Alterações ao(s) script(s):

No âmbito desta meta procedeu-se a desenvolver o código nos scripts de deteção de luz e proximidade gaussianos, com o propósito de gerar uma melhor reação dos veículos aos cenários propostos. Assim foram desenvolvidas três funções, nos *scripts* **LightDetectorGaussScript.cs** e **CarDetectorGaussScript.cs**:

- **GetOutput**: Retorna o *output* a fornecer a cada uma das rodas do veículo. É capaz de devolver o valor calculado ou o inverso ( $1 - \text{output}$ ) dele, consoante o uso de uma *flag inverse*.
- **GaussianOutput**: Responsável por devolver o valor gaussiano final a passar ao carro em questão. Aplica os limiares (**min\_x** e **max\_x**) e os limites (**min\_y** e **max\_y**) à função gaussiana, de forma a poder gerar um melhor controlo sobre o valor retornado.
- **GaussianFunction**: Responsável por calcular e aplicar a fórmula da função gaussiana e passar à função acima descrita o valor da operação. Desta forma irá ser aplicada a seguinte fórmula:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Onde:

- $\sigma$  representa o desvio padrão;
- $\mu$  representa a média;
- $x$  representa a intensidade do estímulo dos sensores.

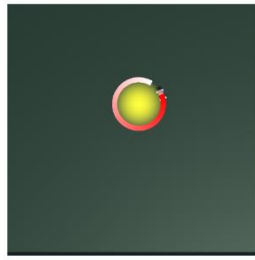
Testes efetuados (Experimentação):

Após a configuração da função gaussiana, começámos a fase de testes e o primeiro a testar foi o **Cenário A**. Neste cenário colocámos os sensores trocados, a *flag* de inverter a falso, e usámos os seguintes parâmetros nas variáveis:

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.12	0.5	0.25	0.75	0.05	0.6

1º valores testados no **Cenário A**

Os valores iniciais permitiram ao nosso veículo obter o comportamento desejado logo do início. Para ser mais visível o efeito neste cenário e nos que se seguem, adicionámos ao carro um *trail* de forma a ser mais perceptível a trajetória descrita ao longo do teste.



**Figura 6** – Veículo de Braitenberg a descrever um círculo em torno da luz

De seguida, e após o sucesso do **Cenário A**, decidimos começar a fazer testes no **Cenário B**, de forma a colocar o veículo a descrever o infinito à volta das fontes de iluminação. Para isso, inicializámos a *flag* de inverter a falso, com os sensores trocados e a função de ativação gaussiana, suportada pelos seguintes valores:

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.12	0.5	0.25	0.75	0.05	0.6

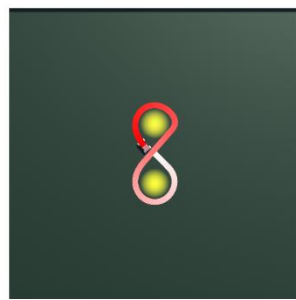
1º valores testados no **Cenário B**

Com os valores iguais aos introduzidos e bem sucedidos no cenário anterior, o carro não conseguiu descrever a trajetória pretendida, desviando-se para fora do objetivo. Isto levou-nos a remover os limites e limiares, o que levou o veículo a deslocar-se em direção à luz e a efetuar um círculo ao redor da primeira fonte de iluminação à qual se dirigiu, falhando assim também no objetivo de atingir o infinito. Na terceira ronda de testes, usaram-se os seguintes parâmetros:

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.12	0.5	0	1	0	1

3º valores testados no **Cenário B**

Com estes valores o nosso veículo conseguiu efetuar bem a curva na primeira fonte de iluminação e prosseguir de seguida à segunda luz, completando o ciclo do infinito. Esta trajetória foi observada de forma indeterminada durante mais de 2 minutos, o que demonstra que o carro é capaz de manter constantemente o trajeto desejado neste cenário.



**Figura 7** – Veículo de Braitenberg a descrever o infinito entre duas luzes

De forma a encerrar os casos de testes iniciais, procedemos a avançar para o **Cenário C**, com o objetivo de colocar o veículo a descrever uma elipse entre as duas fontes de iluminação. À semelhança dos outros cenários, a *flag* de inverter encontra-se a falso, com os sensores trocados. Começámos por usar os seguintes valores:

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.12	0.5	0	1	0	1

1º valores testados no **Cenário C**

Foi imediatamente notório que os mesmos não se adaptavam a este cenário, o que levou o carro a abandonar o trajeto desejado. A partir deste ponto, começámos a refletir sob que mudança seria mais crucial para fornecer aos sensores do veículo de forma a ele descrever uma curva perto da fonte de iluminação que o permitisse dirigir em direção à outra, mantendo assim o curso da elipse. Ora, o desvio padrão é um valor que permite alterar o comprimento e a largura da curva da função gaussiana; assim faria sentido alterar primeiro este valor. Começámos por fornecer os seguintes dados:

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.12	0.5	0	1	0.05	0.6

Alteração feita no sensor **esquerdo** no **Cenário C**

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.3	0.5	0	1	0.05	0.6

Alterações feitas no sensor **direito** no **Cenário C**

O aumento do desvio padrão no sensor direito de 0.12 para 0.3 pretende achatar a curva da função de ativação gaussiana, o que irá permitir ao nosso veículo rodar mais a roda direita e assim descrever o arco necessário para escapar da fonte de iluminação e começar a elipse. Após várias tentativas decidimos ficar com o valor de 0.27, visto que foi aquele que demonstrou a melhor reação. Neste caso, o veículo descrevia uma rotação circular em torno da luz.

De seguida começámos a pensar em alterar os valores da média, de forma a poder deslocar a função gaussiana no gráfico e assim poder fornecer ao veículo a parte da função desejada para a intensidade do sinal fornecida. Com isto em mente, chegámos aos seguintes valores:

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.12	0.25	0	1	0.05	0.6

Alterações feitas no sensor **esquerdo** no **Cenário C**

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.4	0.5	0	1	0.05	0.6

Alterações feitas no sensor **direito** no **Cenário C**

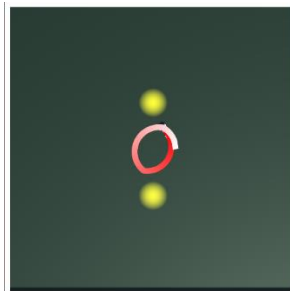
Como é visível acima, houve um ajuste da média no sensor esquerdo de 0.5 para 0.25, o que permite puxar a função ligeiramente atrás e assim controlar melhor o carro. Também é de notar a alteração do desvio padrão de 0.27 para 0.4, de forma a melhor acompanhar as alterações feitas. O veículo assim adquiriu um melhor comportamento, provando-se capaz de efetuar a rotação para a esquerda, semelhante à pretendida para concretizar a elipse, contudo efetua-a depois da fonte de iluminação. Numa terceira etapa de testes começámos a mexer com os limites e limiares dos sensores. Em seguida segue-se mais algumas alterações efetuadas:

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.12	0.25	0	1	0.05	0.8

Alterações feitas no sensor **esquerdo** no **Cenário C**

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.3	0.25	0	0.3	0.05	0.6

Alterações feitas no sensor **direito** no **Cenário C**



**Figura 8** – Veículo de Braitenberg a descrever uma elipse entre duas luzes

O limite máximo no sensor esquerdo foi alterado de 0.6 para 0.8, o que permitiu aumentar o output transmitido à roda esquerda e, assim, aumentar a velocidade da mesma, e o limiar máximo do sensor direito passou de 1 para 0.3 o que permite reduzir, em grande parte, a sua ativação aquando da aproximação da luz, aumentando assim a rotação do veículo. Com estes dados, o veículo finalmente descreve uma elipse entre as duas fontes de iluminação, conseguindo concretizar cerca de 4 iterações antes de perder o controlo, como demonstrado na **Figura 8**. No âmbito de melhorar o trajeto já obtido, fizeram-se pequenas alterações aos sensores, obtendo-se assim a seguinte configuração:

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.1	0.25	0	1	0	0.8

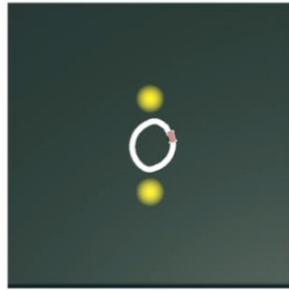
Alterações feitas no sensor **esquerdo** no **Cenário C**

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.27	-0.1	0	1	0.05	0.6

Alterações feitas no sensor **direito** no **Cenário C**



Foram considerados diversos valores para a média (0.5 -> 0.1 -> -0.1) e para o limiar máximo (0.25 -> 0.35 -> 0.3 -> 0.27), tendo sido efetuados múltiplos testes em cada sequência de forma a garantir a fiabilidade dos resultados obtidos aquando da experimentação. Com os valores presentes nas duas tabelas acima, o veículo descreve uma trajetória mais consistente e alinhada com as luzes, conseguindo-se perceber perfeitamente os contornos da elipse formada (**Figura 9**).



**Figura 9** – Veículo de Braitenberg a descrever uma elipse entre duas luzes

#### Cenários criados:

Após conseguirmos realizar as trajetórias determinadas nos cenários de testes previamente descritos, procedemos à criação de um novo cenário com o propósito de testar veículos com diversos sensores de proximidade e deteção de luz, com funções de ativação linear e gaussiana. Para isso, foram criados vários circuitos de testes, que se encontram guardados em **Assets/Scenes/Meta2/New scenery/**:

- **Circuito A:** Loop circular à volta de uma fonte de iluminação de um veículo com os sensores trocados usando uma função de ativação gaussiana.
- **Circuito B:** Pequeno circuito com 4 luzes e um veículo de Braitenberg entre elas, de forma a adotar um comportamento semelhante ao teste feito no **Cenário B** anterior.
- **Circuito C:** Pequeno circuito de luzes oval, no qual um veículo *lover*, dotado de sensores com função de ativação gaussiana, percorre-o com bastante consistência e outro, de deteção de carros (também *lover*) com função gaussiana, o segue de forma constante.
- **Circuito D:** Cenário de testes no qual são colocados dois veículos de Braitenberg, tendo um deles a si associada uma TAG que permite ao outro persegui-lo através de uma função de ativação gaussiana. Para referência foram usados os valores de limiares 0.25 e 0.75, limites 0.05 e 0.6, desvio padrão 0.12, média 0.5 e sensores e função de ativação ambos invertidos.
- **Circuito E:** Circuito de dimensões consideráveis onde um veículo, dotado de sensores que funcionam através de uma função de ativação gaussiana e linear, percorre o espaço perseguindo diversas luzes e completando uma volta ao traçado.
- **Circuito F:** Pequena demonstração de dois veículos *lover* a efetuar um circuito semelhante ao **Circuito C**, um atraído pelas múltiplas luzes (Função linear) e o outro atraído pelo primeiro veículo (Usando função de ativação gaussiana). É notório que o veículo gaussiano acompanha sem problemas o da frente, ao invés do uso de uma função linear, como demonstrado no **Cenário B** da **Meta 1**, quer este tenha os sensores configurados com função gaussiana ou linear, provando assim a sua versatilidade.

Relativamente ao **Circuito E**, as **Figuras 10 e 11** demonstram o trajeto percorrido por cada veículo dependendo da sua função de ativação:



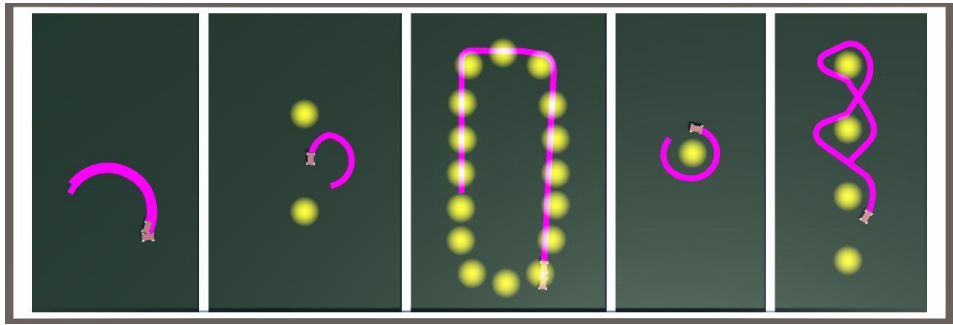
**Figuras 10 e 11** – Circuito do **Circuito E** com função de ativação linear e gaussiana, respetivamente

Como é notório, a **Figura 11** demonstra que o veículo com a função gaussiana consegue cumprir o trajeto num tempo médio de 1 minuto e 9 segundos (foram realizadas 5 cronometragens de tempo), usando as definições listadas na tabela abaixo (definições estas obtidas através de experimentação com o limiar máximo), enquanto que com uma função de ativação linear o carro consegue cumprir apenas uma porção circuito. É de notar que foram necessárias bastantes alterações da velocidade de veículo, nomeadamente 20->30->45->50->47, sendo 47 a velocidade onde se observou o melhor comportamento.

$\sigma$	$\mu$	Min_x	Max_x	Min_y	Max_y
0.12	0.5	0	0.75	0.05	0.6

Configurações dos sensores no **Circuito E**

Como conclusão podemos notar que usando uma função de ativação gaussiana é muito mais fácil percorrer o trajeto, tanto que funciona relativamente bem com diversas combinações de luzes. Já na linear, foram necessários alguns ajustes da posição das luzes de forma a permitir ao veículo percorrer alguma distância. Pouca adaptabilidade é a grande desvantagem quando comparada à função gaussiana, como provado pelos resultados acima descritos. Em ambas as configurações, o carro era agressivo.



**Figura 12** – Veículos de Braitenberg a descrever as trajetórias dos **Circuitos A a D**

Os testes da **Figura 12** foram feitos numa fase inicial pós conclusão dos cenários de testes e, como tal, serviu de plataforma para os **Circuitos E e F** e os testes que nele ocorreram. Como nota final, na deteção de luz o linear tem uma prestação inferior ao gaussiano, necessitando de um traçado ideal para funcionar como desejado. A melhor combinação de veículos encontrada corresponde ao uso de scripts de deteção de carro e luzes gaussianos.