

UNIVERSIDADE DE COIMBRA

FIA PL1 – Relatório TP1



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA

2019216764 – João Monteiro – uc2019216764@student.uc.pt

2019216747 – João Melo – uc2019216747@student.uc.pt

2019216809 – Miguel Faria – uc2019216809@student.uc.pt

Índice

Introdução.....	3
Meta 1.....	4
Detecção de Luz.....	4
Detecção de Proximidade.....	4
Meta 2.....	5
Cenário 1 – Círculo.....	6
Cenário 2 – Infinito.....	6
Cenário 3 – Elipse.....	7
Cenário 4 – Exploração	8
Conclusão	9
Bibliografia.....	10

Introdução

Este projeto tem como objetivo desenvolver competências e conhecimentos relativos a veículos de Braitenberg e estudo dos seus vários comportamentos, com recurso à ferramenta *Unity*. Este projeto encontra-se dividido em duas metas: *Sense it* e *Tune it & Test it*.

Na primeira meta, encontram-se presentes as funcionalidades desenvolvidas relativas à deteção de luzes e de veículos.

Na segunda meta, para além da função linear já existente, foi criada uma função gaussiana com intuito de reproduzir os cenários propostos. Foi também realizado um cenário onde se procurou implementar e explorar os comportamentos destes veículos.

Meta 1

Esta meta tem por objetivo desenvolver as capacidades de detecção de luz e proximidade, devendo os sensores utilizados responder, respetivamente, a fontes de luz e veículos. Nesta meta foram apenas manipulados os ficheiros da *scene* "DetectCars" da diretoria "Meta1" dos materiais fornecidos.

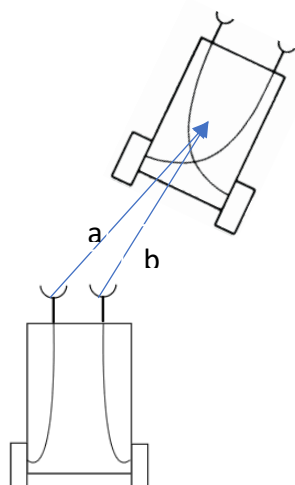
Deteção de Luz

A funcionalidade de detecção de luz já vinha fornecida com a *template*, tendo sido efetuada apenas uma mudança no comportamento do veículo, passando este a ser "agressivo", relativamente às luzes existentes no seu ambiente. Para isto procedemos à associação do sensor esquerdo com a roda direita e do sensor direito com a roda esquerda.

Deteção de Proximidade

Para desenvolver a capacidade de detecção de proximidade, foi adaptado o pseudocódigo lecionado na aula. Após a detecção do carro mais próximo, fica apenas em falta a função de *output*. A fórmula $\frac{1.0f}{\min + 1.0f}$, indicada na aula, foi inicialmente usada como referência. Observou-se que usando esta fórmula se obtém uma velocidade máxima de $1 * \text{Max Speed}$, para uma distância ao carro de 0. Ao aumentar a distância, verifica-se que a velocidade do carro seguidor diminui, fazendo com que o intervalo entre os veículos aumente cada vez mais.

Com o objetivo do carro com a componente de proximidade ter um comportamento "agressivo" mais notável perante o veículo mais próximo, recorreu-se a uma função linear crescente. Para isto apenas utilizámos a variável de distância obtida. Ao contrário do veículo com detecção de luz referido anteriormente, este não necessita de possuir os sensores trocados em relação às rodas, uma vez que é usada a distância entre cada um dos sensores e o veículo. Assim, o sensor que possuir a maior distância irá ter um valor de output superior, fazendo com que a roda associada a este gire com maior intensidade e virando o veículo para o lado pretendido.



Sendo $a > b$, onde a e b , são as distâncias entre o veículo e os sensores, a roda associada ao sensor da esquerda irá rodar mais rápido, uma vez que a fórmula de output provém apenas da distância medida. O veículo irá mover-se de modo a ir ao encontro do carro à sua frente

Para além dos *scripts* alterados, foram modificados alguns parâmetros no *Unity*.

O *Max Speed* foi ajustado para 1, para permitir que não ocorram grandes colisões.

A variável *Angle* foi alterada para 210, para evitar um comportamento anormal por parte do carro seguidor, caso o carro a ser seguido se encontre na sua traseira.

Figura1. Exemplo da função de output implementada

Meta 2

O objetivo principal desta meta é alterar o comportamento do carro, modificando valores da função gaussiana, limiares (*threshold*) de ativação e limites, de maneira a conseguir realizar os cenários Círculo, Infinito e Elipse. Para a sua realização, foram efetuadas modificações nos ficheiros das *scenes* "Circle", "Infinite" e "Eclipse".

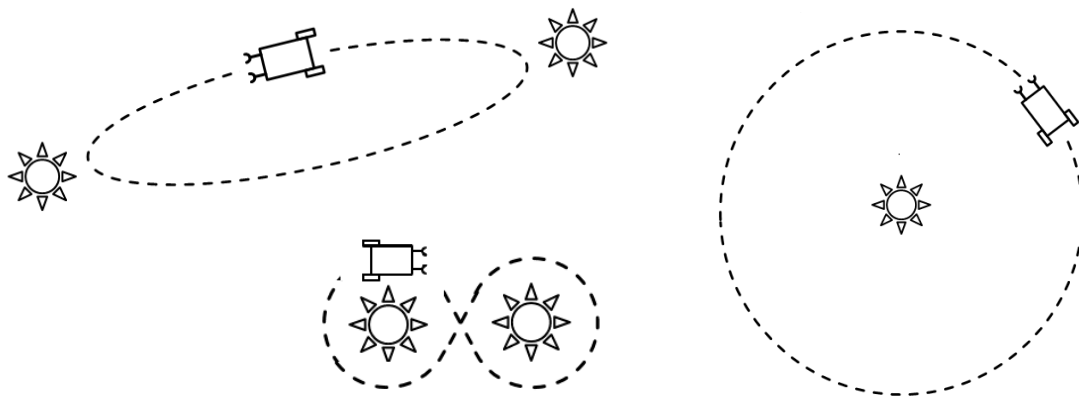


Figura2. Cenários da Meta 2

De maneira a resolver de forma adequada os diferentes cenários, foi utilizada a seguinte função gaussiana:

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

onde x é o *output* dos sensores, σ representa o desvio padrão (*stdDev*) que irá influenciar a altura do máximo da função e μ (*mean*) representa a média que irá influenciar a translação da função.

A Figura3 representa o movimento do veículo segundo esta expressão, onde antes do máximo o veículo será atraído para a luz e após irá afastar-se, nunca colidindo com esta, como ocorria com o uso de uma expressão linear. Isto ocorre em veículos cuja associação entre sensor e roda resulta num comportamento convencionado como atrativo.

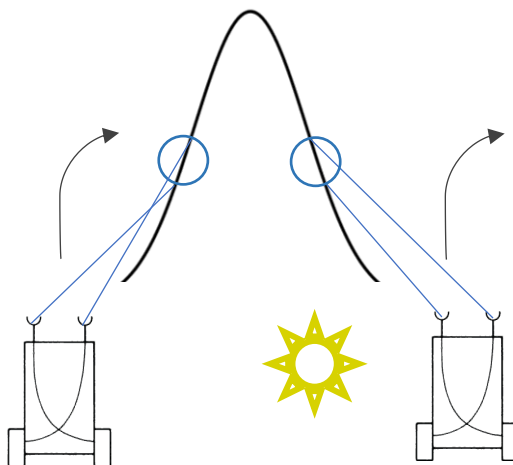


Figura3. Cenários da Meta 2

Para além da função implementada, foram utilizados limiares, para restringir o *output* mínimo e máximo recebido dos sensores, e limites, para controlar o valor máximo e mínimo enviado para as rodas pela função.

Cenário 1 – Círculo

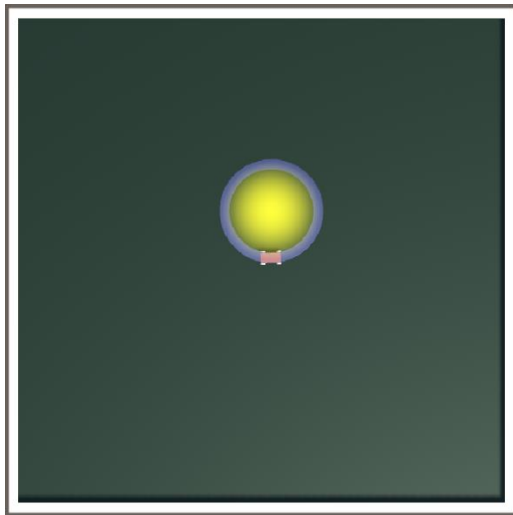


Figura4. Cenário 1 - Círculo

Para a realização deste cenário apenas foram utilizados diretamente os valores referidos no enunciado em ambos os sensores, sem qualquer alteração, tomando os seguintes valores: a função gaussiana tem $\mu = 0.5$ e $\sigma = 0.12$, os limiares são 0.25 e 0.75 e os limites são 0.05 e 0.6.

Uma vez que existem limites, no início do cenário os valores devolvidos para as rodas irão ser iguais em ambas, movendo-se em linha reta até o valor do output dos sensores diminuir.

De seguida, a roda exterior permanece com velocidade estática e a interior não, orbitando a luz de forma circular.

A Figura4 demonstra o cenário obtido segundo esta implementação.

Cenário 2 – Infinito

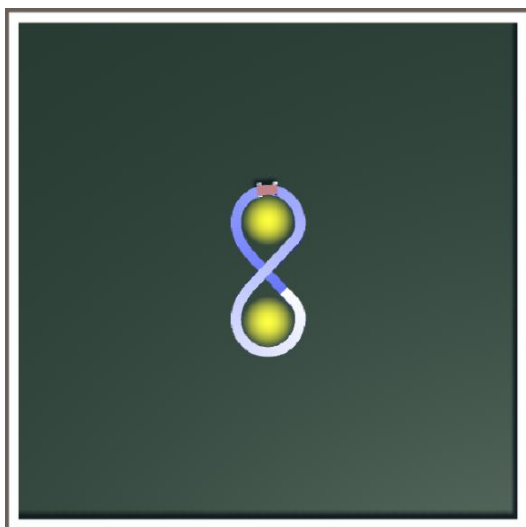


Figura5. Cenário 1 - Infinito

Neste cenário foram utilizados os mesmos valores do cenário anterior, apenas realizando um pequeno ajuste no desvio padrão, passando a ser $\sigma = 0.09$.

Mantendo a lógica referida acima, verifica-se que neste cenário o comportamento será o mesmo ao contornar a luz, porém a existência de uma segunda luz permite que o valor da velocidade da roda exterior vá variando quando esta passa a interior, e vice-versa.

Cenário 3 – Elipse

No caso da elipse, estão associadas a cada sensor funções gaussianas com valores diferentes, dado que nos permite fazer com que a roda interior tenha uma velocidade menor que a exterior, atingindo assim, mais facilmente, a curvatura da elipse.

Para obter este cenário, e tendo em conta o referido acima, a função gaussiana associada ao sensor da esquerda (que irá retornar um valor para a roda da direita) irá possuir uma média inferior de modo a este alcançar o máximo de valor mais cedo.

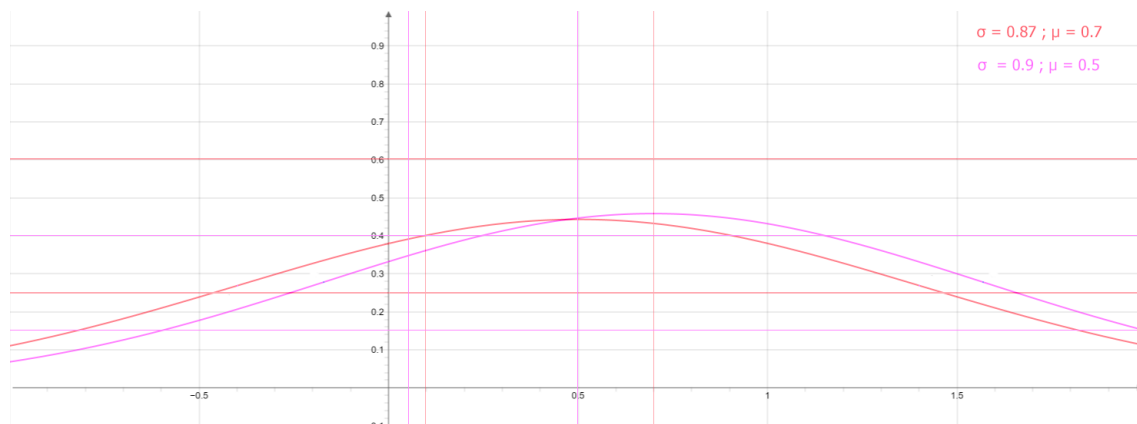


Figura6. Gráficos das funções gaussianas utilizadas e respetivos limites e limites

Analisando gráficos de funções gaussianas, podemos verificar também que uma função com desvio padrão mais alto atingira um máximo mais baixo, contudo a curvatura ascendente ocorre mais cedo.

Deste modo, para o sensor direito, que possui uma média superior e um desvio padrão inferior, ao aplicar um limite máximo de retorno para a roda, consegue-se fazer com que este permaneça com uma velocidade mais reduzida enquanto a da roda direita aumenta, permitindo descrever a curva.

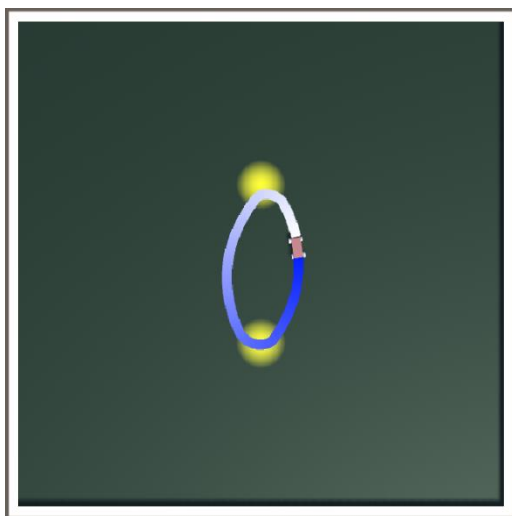


Figura7. Cenário 3 - Elipse

De acordo com esta solução, o resultado obtido é o presente na Figura7.

Esta implementação apenas realiza consistentemente uma elipse um número finito de vezes, acabando eventualmente por se afastar progressivamente abandonando a área possível para a realização desta forma.

Cenário 4 – Exploração

O seguinte cenário foi criado para consolidar e continuar a exploração dos comportamentos destes veículos. As *assets* para criar o cenário usado, foram obtidas através da *Unity Assets Store*^[1].

Este cenário representa uma pista, no qual o objetivo é o veículo dar o número máximo de voltas válidas possível. Sendo uma volta válida classificada como um percurso que é iniciado e terminado na meta, percorrendo a pista inteira sem sair dos limites da mesma.

O veículo possui ambos os sensores de luz e detecção, os dois com funções lineares. Em relação à luz possui um comportamento de agressão e em relação a entidades com a *tag* “CarToFollow” possui um comportamento de medo. Esta *tag* foi atribuída a todas as paredes da pista para o veículo evitar colidir com estas. O nome da *tag* não foi alterado devido a cenários anteriores para evitar código duplicado.

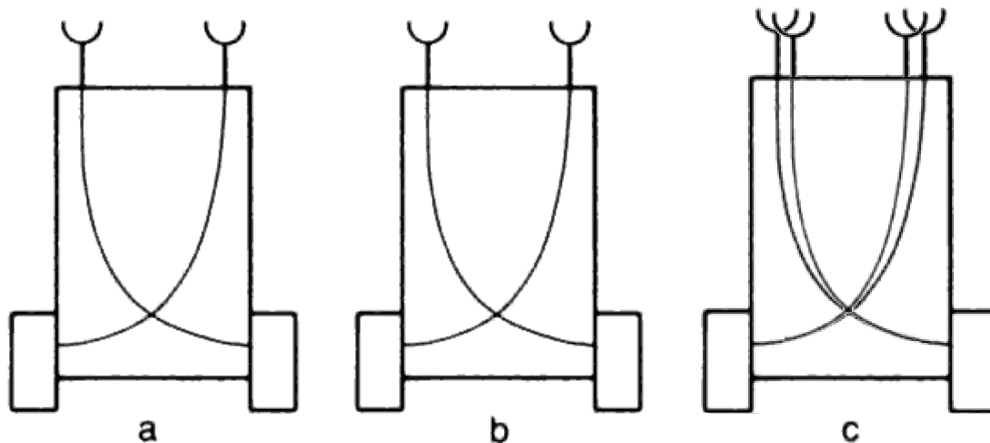


Figura8. Veículo Cena4

A associação “a”, faz correspondência de sensores de luz com as rodas, esta associação irá revelar um comportamento de agressão perante uma luz.

A associação “b”, faz correspondência de sensores de detecção com as rodas, esta associação irá revelar um comportamento de medo perante outros objetos. Apesar de esta configuração ser a convencionada como de atração, como mencionado na meta 1, o valor de *output* provém apenas da distância, por essa razão o sensor que detetar a distância maior em relação ao objeto irá provocar a roda oposta rodar com mais intensidade evitando a colisão.

O veículo “c” será o resultado da junção de a e b.

Para possibilitar a adição de valores entre sensores, foi necessário adicionar uma condição no ficheiro *CarBehaviour2a.cs*, que ocorre no caso de ambas as variáveis *Detect Cars* e *Detect Lights* estarem assinaladas. O ficheiro *CarDetectionLinearScript.cs* também necessita ser modificado, possuindo agora limites e limiares. Para obter um comportamento adequado, foi também necessário adicionar uma variável que permite possuir *Maxspeed* diferentes para cada *output* obtido, sendo esta designada por *divider* estando presente também no ficheiro *CarDetectionLinearScript.cs*.

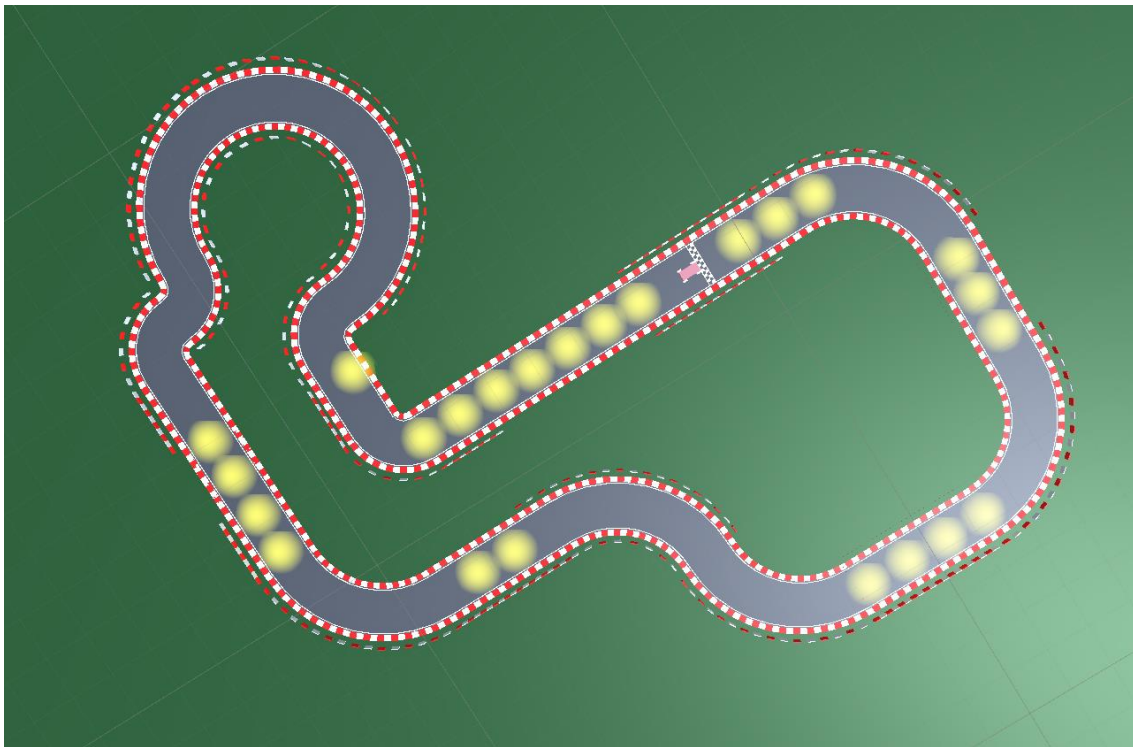


Figura9. Cenário 4 -Extra

Conclusão

Em suma, este trabalho permitiu desenvolver e aprofundar conhecimentos sobre veículos de Braitenberg e os seus complexos comportamentos. É possível concluir que as funções de ativação são o principal agente no resultado destas ações. Permitiu adquirir alguns fundamentos sobre a ferramenta Unity.

Bibliografia

[1] <https://assetstore.unity.com/packages/3d/environments/roadways/modular-lowpoly-track-roads-free-205188>

[2] <https://elsenaju.eu/Functions/Gaussian-Plotter.htm>