

Fundamentos da Inteligência Artificial

Trabalho Prático N°1- Braitenberg Vehicles

Licenciatura em Engenharia Informática- Faculdade de Ciências
da Universidade de Coimbra

Trabalho realizado por:

- Carlos Soares – 2020230124 cmmsoares@sapo.pt PL3
- Daniel Simões – 2019236042 daniel.simoess20@hotmail.com PL6
- Duarte Carvalho – 2020226658 77duarte77@gmail.com PL2

Índice

Meta 1

- Introdução
- Procedimentos
- Decisões
- Cenários

Meta 2

- Funções lineares e gaussianas
- Limites (limits) e limiares (thresholds)
- Cenários criados
- Cena 1: Circulo
- Cena 2: Elipse
- Cena 3: Infinito

Conclusão

Trabalho Prático No1: Braitenberg Vehicles

Meta 1 – Sense It

Breve introdução a veículos Braitenberg e relação com trabalho proposto:

Os Veículos Braitenberg são uma série de veículos robóticos simples projetados pelo cientista italiano Valentino Braitenberg. Estes veículos são projetados para demonstrar o conceito de como conexões simples entre sensores e atuadores podem levar a comportamentos complexos e adaptativos, por ambas as partes.

Os veículos Braitenberg originais consistiam em uma fonte de luz, um foto-sensor e duas rodas. O comportamento do veículo era controlado por uma conexão simples entre o foto-sensor e as rodas, de modo que a velocidade das rodas era proporcional à intensidade da luz detectada pelo foto-sensor. Isso levou a comportamentos interessantes, como fototaxia, onde o veículo se moveria em direção a uma fonte de luz, e anti-fototaxia, onde o veículo se moveria para longe de uma fonte de luz.

No contexto atual, o deste primeiro trabalho prático, o objetivo passa por expandir as capacidades dos veículos Braitenberg adicionando novos tipos de sensores e objetos.

Os novos sensores incluem sensores para detectar outros veículos e obstáculos, além do foto-sensor original. Os novos objetos incluem fontes de luz, outros veículos e obstáculos (blocos). Ao adicionar esses novos componentes, o comportamento dos veículos pode ser tornar mais complexo e adaptativo, permitindo a criação de comportamentos novos e interessantes.

Procedimento a tomar para a realização do trabalho:

Para completar esta primeira meta do trabalho, começamos por analisar o código existente (fornecido pelos docentes) e trabalhar o mesmo de forma a conseguir o objetivo proposto desta primeira milestone. Foi trabalhado de forma que o carro detecta-se blocos entre outras coisas.

Dois dos scripts trabalhados foram: -CarBehaviour

-BlockDetector

Foram criados objetos para colocar nos cenários feitos para o projeto, que, por sua vez, foi uma das etapas da realização de trabalho, a criação de cenários.

Era necessário alterar código de forma que se considerasse o carro mais próximo, ou seja, o carro com a tag "CarToFollow", criando o script "CarDetector", porém, acabou por não ser usado.

A energia seria calculada da seguinte forma:

Em resumo, o procedimento passou por modificar o código existente para adicionar novos sensores e objetos, testar as novas funcionalidades, criar novos ambientes e ajustar o sistema para garantir o desempenho ideal.

$$output = \frac{1}{distancia_minima + 1}$$

Decisões realizadas pelo grupo a ter em questão:

Para se notar que um veículo estava de facto a ser seguido, foi aplicada a velocidade de “70” ao “CarToFollow” e “20” ao carro que iria seguir o mencionado.

Para o carro seguir a luz, o detetor de luz à esquerda ficou ligado ao script eu detetava a luz à direita, o mesmo realizado para o detetor de luz à direita do carro. Esta decisão faz com que o carro se aproxime da luz e fica realizado um dos objetivos do trabalho.

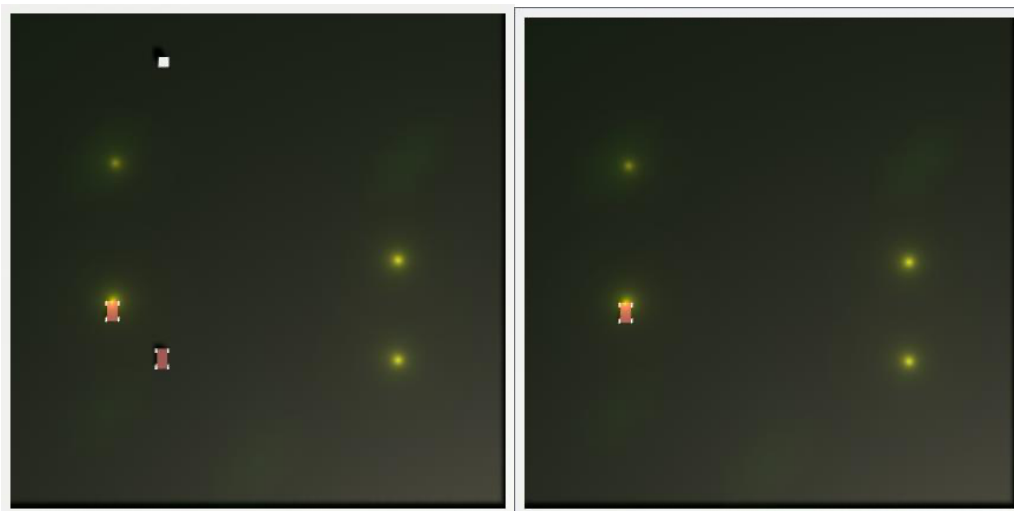
Pelo contrário, queríamos que o carro detetasse blocos e se afastasse do mesmo e, neste caso, o detetor de blocos e o script usado possuem o mesmo “lado”, para que o carro se afaste do bloco.

Por fim, o ultimo objetivo era que um carro seguisse um outro carro. Foi usado o script do “CarBehaviour” e, como o objetivo é que o carro se aproxime dum outro, o sensor que iria detetar o carro é o oposto do “script” usado, ou seja, tal como no detetor de luz, o carro iria seguir algo, neste caso, o “CarToFollow”.

Cenários criados:

Figura 1 – Cenário criado que mostra o carro com a tag “CarToFollow” e o carro que seguiria este mesmo.

Quando ativada a checkbox de detetar luz, blocos ou carros, o veículo faz o suposto.



Meta 2 – Tune it & Test it

Nesta meta implementámos três cenas diferentes com os veículos a percorrer o mapa com diferentes padrões usando então os sensores de luz lineares e gaussianos. Criou-se então o seu mapa com as paredes e o chão para delinear o território.

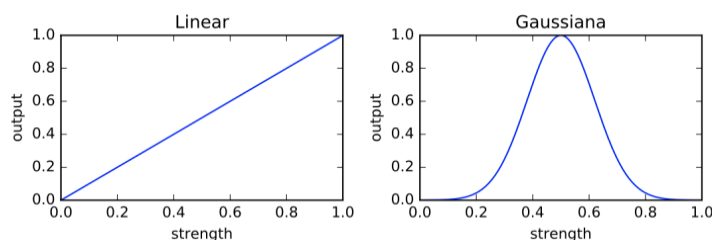
O corpo docente disponibilizou os *scripts* necessários para se poder implementar as diferentes versões do *lightDetectorScript.cs*, neste caso *lightDetectorLinear.cs* e *lightDetectorGauss.cs*.

Trabalhou-se então nestes dois scripts de forma a implementar o círculo, a elipse e o infinito.

Funções lineares e gaussianas:

Numa função linear podemos observar que à medida que o valor de x, ou seja a strength, aumenta, o valor do output aumenta proporcionalmente. Isto leva-nos a concluir que o movimento do carro, quando aplicada esta função, terá uma trajetória linear.

Numa função gaussiana podemos observar que existe uma subida repentina do output à medida que o valor da strength aumenta; quando chega ao pico (seu valor máximo) acontece uma descida repentina. Isto leva-nos a concluir que o movimento do carro consegue modelar-se a trajetos mais complicados.



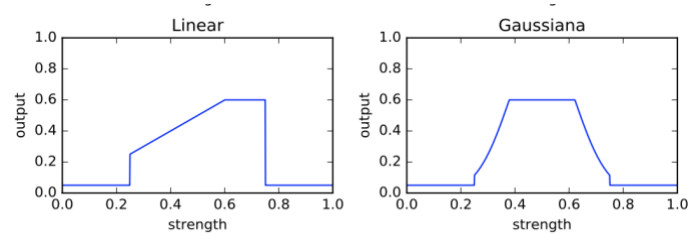
Limites (limits) e limiares (thresholds):

Os limites (eixo y) são os valores que condicionam a output, que é calculado através da função:

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

Os limiares ou (eixo x) são os valores que condicionam a strenght, que é aplicada ao carro. Como podemos observar nos gráficos, se aplicarmos os limites e limiares, conseguimos obter valores mais restritos de modo a controlar de uma melhor forma o percurso do carro.

Os limites e limiares aplicados em conjunto com as funções já explicadas acima - lineares e gaussianas - ajudam- nos a garantir que conseguimos obter o objetivo desta segunda meta: conseguirmos que o carro percorra de forma correta o trajeto proposto nos três cenários propostos.



Cenários criados:

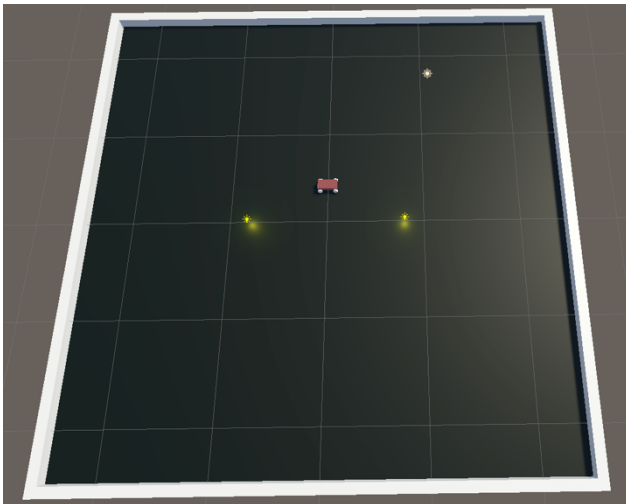


Figura 2- Elipse



Figura 1- Infinity

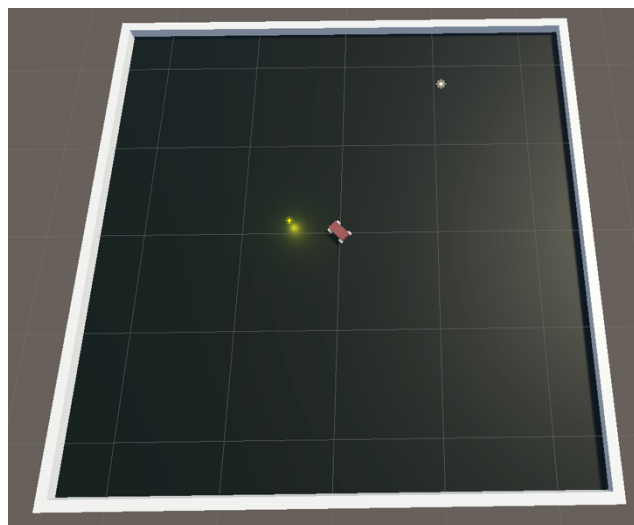


Figura 3-Circle

Cena 1: Circulo

Nesta cena para fazer o carro percorrer um circulo à volta de uma luz, atribuímos o **lightDetectorGaussRight** e **lightDetectorLinearRight** na componente **Right LD** e, de igual modo, o **lightDetectorGaussLeft** e **lightDetectorLinearLeft** em **Left LD**.
O carro percorreu então a luz, mantendo sempre esta do seu lado esquerdo continuamente.

Limites e limiares:

Neste teste apenas se alteraram os valores da média e do desvio, 0.5 e 0.1 respetivamente. Com a manipulação dos dois primeiros parâmetros foi-nos possível controlar a velocidade atingida pelo carro e, consequentemente, chegar ao trajeto esperado.

Cena 2: Elipse

Nesta cena para fazer o carro percorrer a elipse à volta de uma luz, atribuímos o **lightDetectorGaussRight** na componente **Right LD** e, de igual modo, o **lightDetectorGaussLeft** em **Left LD**.

O carro percorreu então o caminho em direção à luz e volta para trás em direção à outra luz do mapa. Faz esse percurso algumas vezes até sair da rota e começar a fazer círculos junto à luz do topo.

Limites e limiares:

Neste teste alterara-se os valores dos limites e limiares tal como o desvio e a média.

Cena 3: Infinito

Nesta cena para fazer o carro percorrer um circulo à volta de uma luz, atribuímos o **lightDetectorGaussLeft** na componente **Right LD** e, de igual modo, o **lightDetectorGaussRight** em **Left LD**.

O carro, consequentemente, desloca-se em direção à primeira luz, contorna-la e, de seguida, contorna a luz oposta, fazendo o que estávamos à espera.

Limites e limiares:

Neste teste alterou-se os limites e lineares, tal como o desvio e a média, sendo estas 0.1 e 0.5, respetivamente.

Conclusão

Com este trabalho conseguimos solidificar os conhecimentos obtidos nas, aulas teóricas e práticas, sobre agentes reativos, mais concretamente os veículos de Braitenberg, com um trabalho conjunto de teste nos diferentes cenários.

Com a disponibilização do código-fonte por parte dos docentes, conseguimos-nos focar mais na parte de teste e perceber como os demais valores e variáveis podem alterar o resultado final. A adaptação ao Unity foi suave, graças também ao tutorial efetuado pelo professor.

Tivemos algumas dificuldades com o segundo cenário (a elipse), conseguindo apenas parcialmente o resultado pretendido.

Em suma, este trabalho prático ajudou a consolidar os conhecimentos já lecionados anteriormente, e a influência das funções lineares e gaussianas no comportamento dos diferentes agentes.