



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA



Fundamentos de Inteligência Artificial
2022/2023 - 2º semestre

Trabalho Prático Nº1: Braitenberg Vehicles
Meta 1

Trabalho realizado por:
Bruno Sequeira nº2020235721
Rui Santos nº 2020225542
Tomás Dias nº 2020215701

Introdução:

O trabalho tem como objetivos aquisição de competências de desenvolvimento de aplicações no Unity como ambiente de simulação, pretendendo-se personalizar os veículos de brautenberg, tanto a nível funcional como visual. Tal implica expandir os “prefabs” distribuídos com o projeto de forma a acrescentar novos sensores de detecção de veículos, novas funções de ativação e novas cenas.

Sense it:

O objetivo desta meta é a partir de um cenário que contém 10 luzes pontuais e dois carros, que o carro da frente (“Vehicle2aGuide”) seguisse um comportamento de “agression” a partir dos sensores das rodas, e que o segundo veículo o perseguisse com um comportamento de lover. E noutro cenário, que contém uma luz pontual, 1 carro e 1 bloco, o carro desvia-se do bloco e vai em direção da luz.

Para isso tivemos os seguintes passos:

1. Comportamento do Vehicle2aGuide (cenário -> detectCar).

Após executarmos o cenário que nos foi disponibilizado, o comportamento do carro “Vehicle2aGuide” era o oposto que o que desejamos, pelo facto de ter um comportamento de fear em relação às luzes pontuais, enquanto que o esperado era um comportamento de “agression”, ou seja, ir em direção às luzes.

Com o intuito de resolver este problema, trocamos os scripts nos sensores do veículo, sendo associado ao sensor esquerdo o script “LightDetectorLinearRight” e no direito o script “LightDetectorLinearLeft”.

2. Colocação de um veículo a seguir outro (cenário -> detectCar).

Neste passo era necessário desenvolver o script CarDetectorScript.cs, sendo que tem certas diferenças com o script da detenção de luzes. Tivemos de alterar a fórmula utilizada para calcular a energia total recebida pelo sensor, visto que os veículos não têm atributo range, não fazia sentido utilizá-lo na fórmula da detenção dos veículos. logo a fórmula foi:

$$\text{distância} = (\text{transform.position} - \text{car.transform.position}).\text{magnitude}.$$

Tal como no passo nº1 tivemos de colocar os sensores(cruzar) para que tivesse um comportamento de “agression”, ou seja, ir em direção ao carro perseguido.(Vehicle2aGuide).

3. Colocação de obstáculos. (cenário -> main)

Neste cenário temos 1 carro, 1 bloco e uma luz pontual. Sendo que neste passo era necessário que o carro tivesse um comportamento de medo em relação, ou seja, “fugir” do bloco e um comportamento de “agressão” em relação ao ponto de luz.

O objetivo era que o veículo desviasse do bloco e fosse em direção ao ponto de luz, sendo este objetivo concluído com sucesso. Adicionamos 2 sensores de blocos, tivemos de adicionar no script “CarBehaviour.cs” o RightBD e LeftBD, para que possamos colocar nos sensores os scripts correspondentes sendo que estes ficaram em paralelo, visto que é um comportamento de medo.

Para isso tivemos de criar mais 2 scripts, “BlockDetectorScript.cs” sendo que a energia total recebida pelo sensor foi calculada com:

`distancia = (bloco.transform.position - transform.position).sqrMagnitude;`

E a “BlockDetectorScript.cs” que retornava o output: $1/\text{distancia}+1$.

Após calcular o valor das energias dos sensores, tivemos de alterar o código do script “CarBehaviour2a.cs”, visto que o carro detetava as energias da luz e do bloco, tivemos de comparar se o valor da energia dos sensores da luz eram maiores que os sensores do bloco, se fossem então o carro teria de ir em direção à luz, se fosse o contrário significava que ainda não tinha “ultrapassado” o bloco.