FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

Departamento de Engenharia Informática

Relatório Final

Braitenberg Vehicles

Fundamentos de Inteligência Artificial (FIA)

João Catré Nº 2019218953 – joaocatre@student.dei.uc.pt – PL8

Marco Pais Nº 2019218981 – marcopais@student.dei.uc.pt – PL8

Tiago Oliveira Nº 2019219068 – tiagooliveira@student.dei.uc.pt – PL8

2021/2022

Professor: Nuno António Marques Lourenço
Sérgio Miguel Martins Rebelo

Índice

Introdução

Fizemos este trabalho no âmbito da cadeira de Fundamentos de Inteligência Artificial, visando criar e analisar o comportamento de agentes reativos autónomos. Usando os veículos de *Braitenberg*, e com recurso ao Unity, criámos agentes com vários comportamentos distintos, por carros e/ou luzes, que veremos ao longo do trabalho. Foram, também, criadas duas funções de ativação: linear e gaussiana.

Meta 1

Após a aula de introdução ao projeto, onde foi testado a *Unity Scene* "Main", foi carregada a segunda *Unity Scene* existente no *package* fornecido ("DetectCars"). Nessa *Scene* existiam dois (2) carros, sendo que o primeiro carro (*"Vehicle2aGuide"*) tinha, inicialmente, um comportamento de "*Fear"*, em relação à luz. Para poder transformar o comportamento desse carro num comportamento *"Agressive"* foi necessário trocar os *scripts* dos sensores de luz existentes nas rodas do carro (o *script* da esquerda passou para a direita e vice-versa). Esta alteração permitiu ao carro circular pelo tabuleiro, seguindo as fontes de luz.

O passo seguinte foi colocar o carro de trás (*"Vehicle2a"*) com um comportamento de *"Fear"*, ou seja, era necessário que o carro seguisse o carro mais próximo e aumentando ou diminuindo a velocidade consoante a distância para o carro mais próximo. Sempre que o carro *"Vehicle2a"* se afastasse do carro *"Vehicle2aGuide"* a sua velocidade aumentaria e caso se aproximasse demasiado a velocidade iria diminuir, de modo a não bater nem ultrapassar o carro da frente. O *"Vehicle2a"* também poderia ter sido implementado com um comportamento de *"Lover"*, no entanto seria necessário utilizar o inverso do output (1 - output).

Para que se pudesse implementar o comportamento descrito anteriormente foi necessário alterar 2 *scripts*, *"CarDetectorScript.cs"* e *"CarDetectorGaussScript.cs"*, sendo o segundo associado aos sensores do *"Vehicle2a"*. Em primeiro lugar, foi adicionada a função *"GetVisibleCars"* que retorna um *array* com os vários objetos com a *tag "CarToFollow"* existentes no campo de visão do *"Vehicle2a"*. Após o retorno do *array* é calculada a distância para cada um desses objetos e definido o mais próximo do *"Vehicle2a"*.

Para calcular o output a ser fornecido às rodas do veículo foi usada a função de Gauss. Antes da utilização da função tiveram de ser criadas *if-else clauses* que aumentam ou diminuem o desvio padrão fazendo com que a velocidade do carro fosse ajustada ao movimento do carro guia. Para determinar os melhores valores para essas condições tivemos de recorrer a vários testes de tentativa-erro.

Meta 2

Na segunda meta os objetivos são um pouco diferentes. Quisemos melhores resultados e, para tal, recorremos a duas funções - Gaussiana e linear. Para além destas funções, recorremos, também, ao uso de limiares e limites, em ambas as funções, e, ainda, 2 variáveis: *Mean* (média) e *Std Dev* (desvio padrão) na função Gaussiana.

Os limiares atuam sobre o domínio da função e servem para que o intervalo de energia esteja limitado entre os valores do mesmo. Já os limites agem sobre o contradomínio da função e servem para ignorar ruídos, isto é, ignorar valores muito baixos de energia e valores de energia muito altos, para que não sejam produzidos resultados inesperados.

O desvio padrão (*Std Dev*) serve para regular a suavidade da curva Gaussiana, ou seja, para controlar o quão depressa é feita a aproximação do veículo ao objeto desejado (por exemplo, outro veículo ou fonte luminosa). A média (*Mean*) serve para deslocar o centro da curva gaussiana no eixo Ox, isto é, quanto maior for o seu valor melhor a resposta a curtas distâncias e quanto menor o valor melhor a resposta a longas distâncias.

De seguida, procedeu-se à implementação dos cenários propostos: círculo, elipse e infinito. Como qualquer projeto de IA é necessário realizar diversas experiências, testando vários valores, de modo a obter os mais adequados. Assim, juntamente com alguma informação pertinente para cada cenário, serão mostrados alguns dos imensos testes realizados até obter os melhores valores em cada cenário.

Círculo

No primeiro cenário, o círculo, tivemos como objetivo colocar o veículo a fazer uma trajetória circular. Assim, ligámos o sensor da esquerda à roda direita (e o da direita à roda da esquerda), para que o veículo tivesse um comportamento agressivo. O sensor direito (mais afastado da luz) recebe menos energia e, consequentemente, menor é a velocidade da roda esquerda, enquanto o sensor esquerdo recebe mais energia, tornando assim a velocidade da roda direita superior. Assim, garantimos que o veículo, após um ajuste inicial no seu movimento, realiza uma trajetória circular em torno da luz.

Para obter os resultados pretendidos e perceber que a configuração das figuras 1 e 2 eram as que obtemos melhores valores tivemos de testar vários valores, começando por colocar os dois sensores com o mesmo desvio padrão. Assim, para garantir que o carro ao se aproximar da luz iria virar foi necessário limitar o valor obtido pela função de Gauss, para o sensor da direita, pois este está ligado à roda esquerda do veículo, recebendo assim mais luz. Com esse limite foi possível garantir que o sensor da esquerda tinha um valor de retorno da função de Gauss superior, fazendo com que a roda direita tivesse mais velocidade.

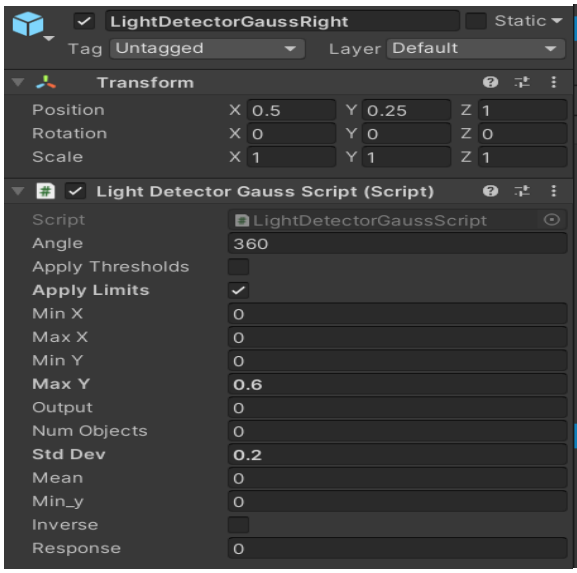

Figura 1: Parâmetros sensor direito

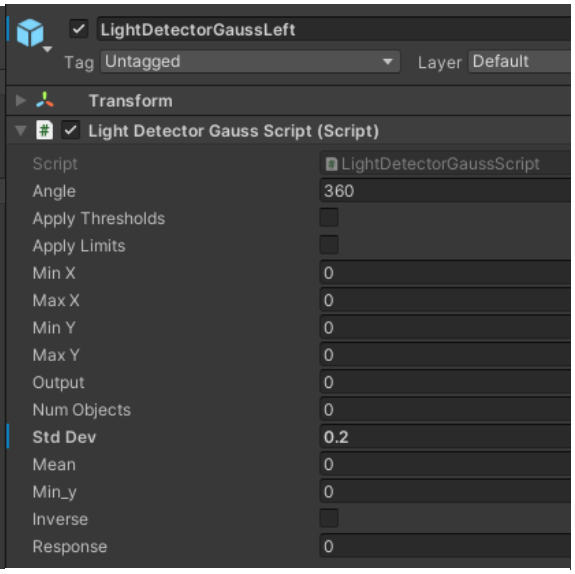

Figura 2: Parâmetros sensor esquerdo

Elipse

No segundo cenário, a elipse, tivemos como objetivo colocar o veículo a fazer uma elipse entre duas luzes. Para este cenário foi necessário realizar imensos testes e experiências, visto que foi o cenário mais complicado de implementar.

Optámos, mais uma vez, por um modelo agressivo. Trocámos os sensores (sensor direito com roda esquerda e roda direita com sensor esquerdo) para obter velocidades superiores na roda direita e inferiores na roda da esquerda. Como a roda esquerda receberá valores superiores mais rapidamente em relação à roda direita, logicamente, trocar os sensores foi a opção mais correta.

Para concretizar a trajetória deste cenário tivemos de obter transições mais bruscas, comparando com a trajetória circular, pois à medida que o veículo se aproxima mais da luz é necessário conseguir fazer a curvatura mais acentuada, sendo isso possível através de uma maior velocidade na roda direita (um desvio padrão inferior no sensor esquerdo).

Após muitas experiências com os limites, limiares, média e desvio padrão, obtivemos os valores mostrados nas figuras 3 e 4 e uma trajetória como demonstrado na figura 5.

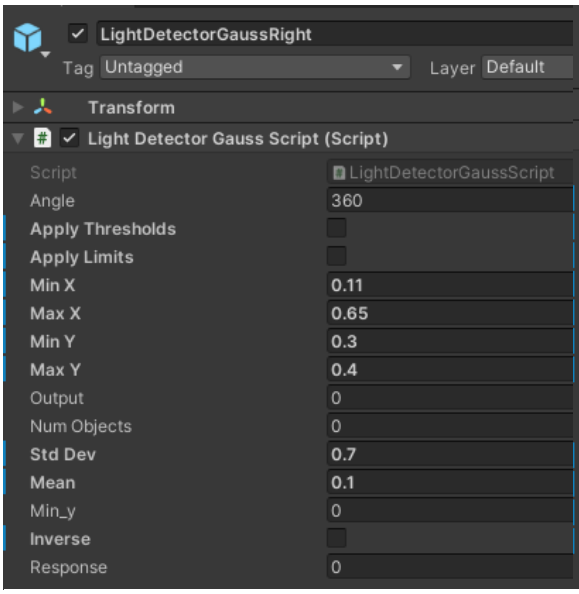

Figura 3: Parâmetros sensor direito

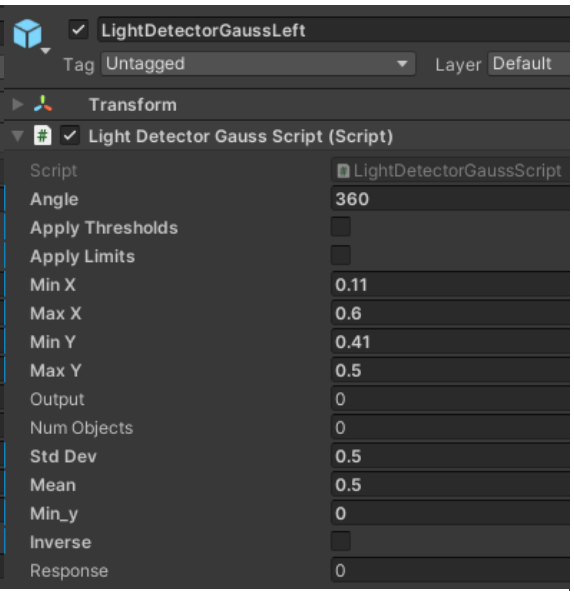

Figura 4: Parâmetros sensor esquerdo

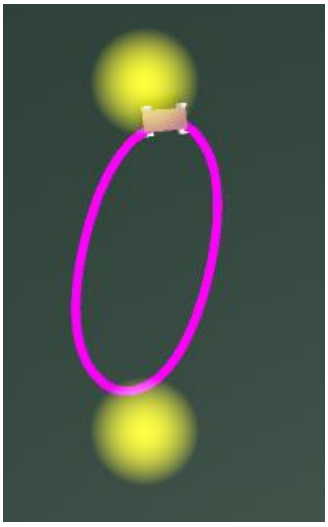

Figura 5: Trajetória elipsoidal

Podemos observar que a elipse está ligeiramente curva. Aplicando pequenas alterações, obtivemos resultados muito diferentes e menos satisfatórios. Muitos dos testes realizados consistiram em aplicar thresholds (limiares) e limites. No entanto ao aplicar os thresholds verificámos que o carro efetuava uma curva demasiado abrupta, quando muito próximo da luz. No caso da aplicação dos limites apenas conseguimos colocar o carro a desviar mais para a direita ou mais para a esquerda, consoante os valores que aplicávamos em cada sensor, fazendo com que a elipse fosse mais fechada ou mais aberta. No entanto, quase todas as vezes em que utilizámos limites a elipse nunca ficava completa.

Tal como na trajetória circular, foram testados diversos valores para o desvio padrão, mas neste cenário tivemos ainda de aplicar valores de média para cada um dos sensores.

Infinito

No último cenário, o infinito, queríamos o veículo a fazer, praticamente, duas trajetórias consecutivas circulares, uma em volta da luz de cima e outra em torno da luz de baixo. Para conseguir este efeito, optámos por seguir um comportamento de atração, no qual ambos os sensores estão ligados à roda do lado correspondente, isto é, o sensor esquerdo está ligado à roda esquerda e o direito à roda direita.

Ao contornar a luz de cima, o veículo irá percorrer uma trajetória circular, mas há um momento em que o veículo se encontra à mesma distância das duas luzes (a energia recebida, de ambas as fontes luminosas, é a mesma, neste momento). Assim o veículo irá ficar, momentaneamente, com a mesma velocidade em ambas as rodas. À medida que se aproxima mais da luz de baixo, recebe mais energia desta fonte luminosa e, consequentemente, repete o mesmo processo explicado inicialmente para a fonte luminosa superior. O ciclo repete-se, percorrendo uma trajetória em forma de infinito, como desejado.

Para este cenário foi necessário colocar o desvio padrão igual nos dois sensores, fazendo com que no instante em que os sensores recebem a mesma quantidade de luz o carro descreva um movimento retilíneo. Foi ainda necessário usar limites (para ambos os sensores) para o valor retornado no fim de aplicar a função de Gauss não ser inferior a 0,5.

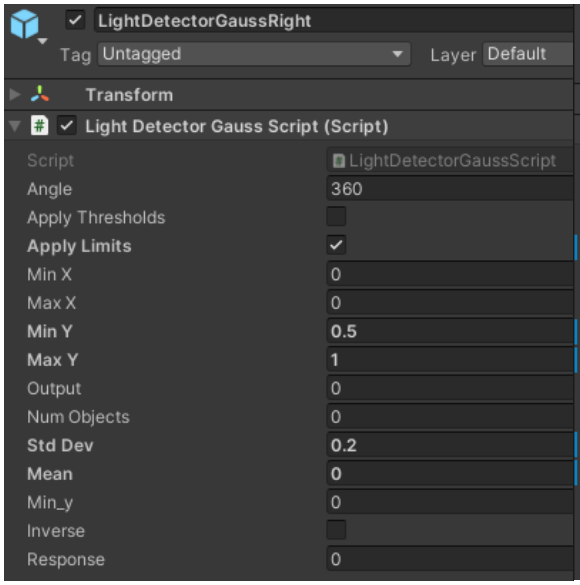

Figura 6: Parâmetros sensor direito

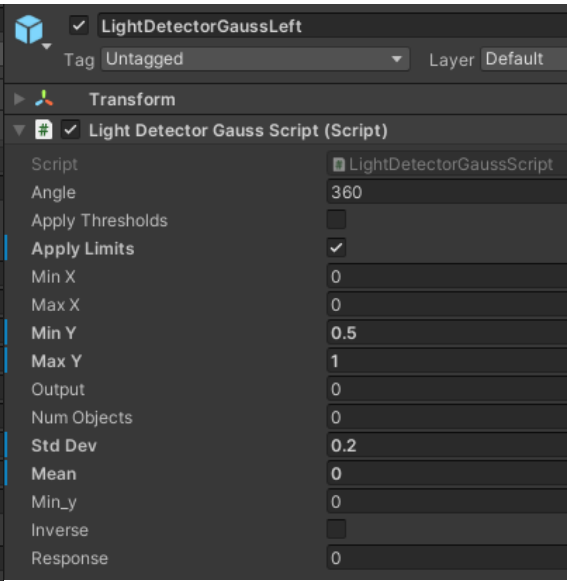

Figura 7: Parâmetros sensor esquerdo

Conclusão

Em suma, ao estudar, explorar e fazer experiências com os veículos de *Braitenberg*, conseguimos adquirir uma melhor compreensão em relação aos mesmos e, consequentemente, compreender melhor o comportamento de agentes reativos autónomos.

Como pudemos perceber ao realizar este trabalho, os agentes reativos autónomos necessitam de ser bastante aperfeiçoados de modo a obter bons resultados. Para tal, é necessário realizar imensas experiências para melhor entender o comportamento destes agentes.

Referências

justinhabit. (2013). *instructables circuits*. Obtido de instructables: https://www.instructables.com/Braitenberg-vehicles-1-4-vehicles-that-3mot3/#:~:text=Step%203%3A%20Choose%20Your%20Destiny!%20(setting%20the%20Modes)

Wikipedia. (14 de setembro de 2021). Obtido de From Wikipedia, the free encyclopedia: https://en.wikipedia.org/wiki/Braitenberg_vehicle