

Fundamentos de Inteligencia Artificial

Relatório da Primeira Meta TP1

Estudante João Moreira joaomoreira@student.dei.uc.pt 2020230563
TP1
Estudante Tomás Pinto tomaspinto@student.dei.uc.pt 2020224069
TP1
Professor Dr. Nuno Lourenço

1 Introdução

Os veículos de Braitenberg são conhecidos pela sua simplicidade: um robo com um sensor¹ ligado ao motor de uma roda. Com esta mecânica simples, facilmente obtemos robos com complexidades incríveis.

Braitenberg, no seu livro propôs quatorze veículos, dos quais quatro mais famosos:

1. *Fear* - Muda de direção e aumenta a sua velocidade, quando deteta a luz. Transmite a sensação de medo da luz.
2. *Aggressor* - Vai em direção à luz e aumenta consoante a sua proximidade. Transmite a sensação de agressividade sobre a luz.
3. *Explorer* - Quando deteta a luz, diminui a velocidade e muda a sua direção.
4. *Lover* - Quando deteta a luz, diminui a velocidade e mantém a sua direção.

2 work

Foi-nos disponibilizado "prefabs" que continham um objeto (fonte de luz) e os seus sensores já implementados. A nossa tarefa consistiu em explorar e melhorar as capacidades dos veículos com objetivo de permitir a deteção de outros veiculos e obstaculos (blocos), para além das luzes.

2.1 Luz

Começamos por modificar o veiculo inicial, do tipo *Fear*, que se afastava da luz para um comportamento do tipo *Agressor*. Para isto trocamos os sensores lineares de deteção de luz do veículo, ou seja, ligamos o sensor esquerdo à roda direita e vice versa.

2.2 Veículos

Seguidamente passamos a fase de interação entre veiculos, em que o objetivo era fazer com que um veículo seguisse outro, sendo assim do tipo *Lover*. Para isto tivemos de desenvolver o script *CarDetectorScript.cs*, semelhante ao script de deteção de luz, que permite a deteção de veículos que tenham a tag "CarToFollow", interagindo com o que se encontrar mais proximo. Para o calculo da energia usamos:

$$Energy = \frac{1}{distance(closestCar, car) + 1}$$

No *CarDetectorLinearScript* invertemos a energia com:

$$Energy_{final} = 1 - Energy_{initial}$$

para que o carro tenha mais energia quando mais distante do veículo que segue.

¹Normalmente sensores de luz são os mais citados. No entanto, trabalhos com narizes eletronicos têm sido realizados.

2.3 Blocos

Para expandir as capacidades do nosso veículo, implementamos um *explorer*. Desta forma, o carro irá explorar todo o espaço, quando detectar um bloco a sua velocidade baixará e desvia-se do objeto.

A sua implementação está nas classes *BlockDetectorScript* e *BlockDetectorLinearScript*. Para o cálculo da energia usamos:

$$Energy = \frac{1}{distance(closestBlock, car)}$$

Vale ter em atenção que a energia é calculada com o bloco mais próximo ao carro.

No *BlockDetectorLinearScript* invertemos novamente a energia com:

$$Energy_{final} = 1 - Energy_{initial}$$

para que o carro tenha mais energia quanto mais distante do bloco.