Relatório do trabalho prático "Braitenberg Vehicles"

Fundamentos de Inteligência Artificial

04 de março de 2023

—

Henrique Costa 2020214120 PL2
uc2020214120@student.uc.pt

Gonçalo Senra 2020213750 PL2
uc2020213750@student.uc.pt

João Coelho 2020235901 PL2
uc2020235901@student.uc.pt

—

Fernando Machado

Nuno Lourenço

Índice

Meta 1- Sense it

Nesta meta, foram implementados os sensores de veículos e os sensores de blocos.

Nos sensores de veículos, o veículo precedente tem apenas de seguir o carro com a Tag "CarToFollow" mais próximo de si, sendo que, para isso, tivemos de iterar todos os carros com esta Tag para determinar qual o mais próximo. Para que o carro seguisse outro dentro das condições mencionadas anteriormente, aplicamos as ligações dos sensores às rodas conforme um veículo do tipo "lover" (sensor esquerdo ligado a roda direita e vice-versa), obtendo-se uma aproximação ao objeto que neste caso, é bastante reativa uma vez que, da forma que está implementada a função que calcula a energia de output, quanto mais próximo estiver, maior será a sua energia.

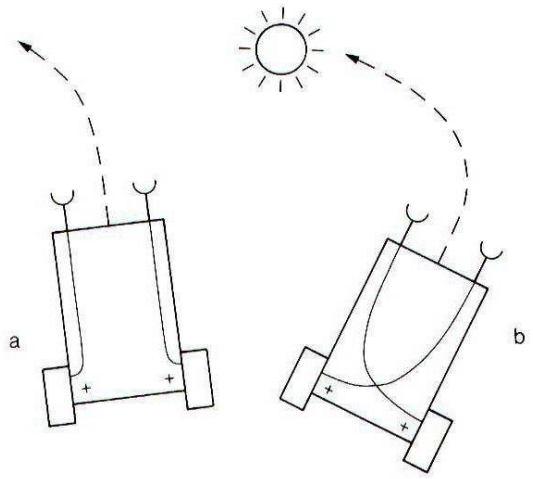

Figura 1 - Braitenberg Vehicles (a – explorador, b - lover).

Nos sensores de obstáculos, a abordagem foi quase totalmente distinta porque apesar de o veículo identificar o obstáculo pela sua Tag "Block", em vez de ir em direção ao mesmo, acaba por obter um comportamento de tipo de explorador (sensor direito ligado a roda direita e vice-versa), visto que o objetivo é evitar estes blocos obtendo na mesma, mais energia conforme a sua proximidade ao mesmo.

Por fim, com a realização de testes conseguimos depurar que as interações pretendidas foram implementadas com relativa fiabilidade sendo que, para evitar colisões entre o carro perseguidor e seguido, teremos de implementar as funções requeridas na próxima meta.

Meta 2 – Tune it & Test it

2.1-Funções

Numa fase inicial desta meta foram implementadas funções de ativação (linear e gaussiana), funções capazes de impor limiares de ativação mínimos e máximos e, por fim, funções capazes de impor limites superiores e inferiores.

Para desenvolver a função de ativação gaussiana, usámos a lógica do cálculo de energia total explicada no enunciado do trabalho prático, no entanto, foi necessário substituir a fórmula da função linear pela fórmula da função gaussiana, **figura 2**.

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

Figura 2 – Função de ativação gaussiana.

De seguida, para desenvolver uma função capaz de impor limiares mínimos e máximos, foi necessário verificar o output resultante da aplicação da fórmula anterior. Caso este valor seja maior que o máximo definido ou menor que o mínimo será devolvido um output com o valor do mínimo do Y. Podemos verificar esta lógica ao observar os gráficos da **figura 3** onde foram aplicados limiares de 0.25 e 0.75. Caso o output seja maior que 0.75 ou menor que 0.25, o valor do output é automaticamente igual a 0.

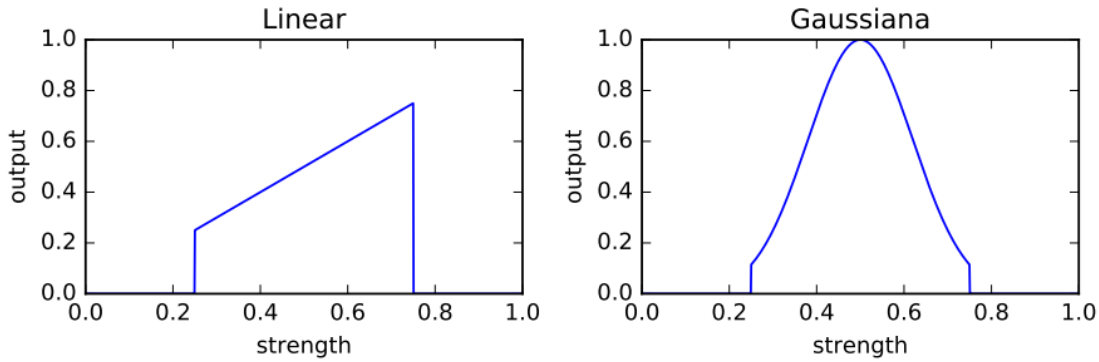

Figura 3 – Gráficos com limiares de 0,75 e 0,25 aplicados.

Por fim, a lógica para desenvolver uma função para aplicar limites máximos e mínimos foi bastante parecida à anterior. Caso o valor do output, calculado pela mesma fórmula, seja maior que o Y máximo, é devolvido um valor de output igual ao Y máximo. Por outro lado, quando o valor do output é menor que o Y mínimo, é devolvido um output igual ao Y mínimo. Podemos mais uma vez observar a **figura 4,** enquanto o valor do output for superior a 0.6, o mesmo é transformado em 0.6 e o mesmo acontece para o limite mínimo de 0.05.

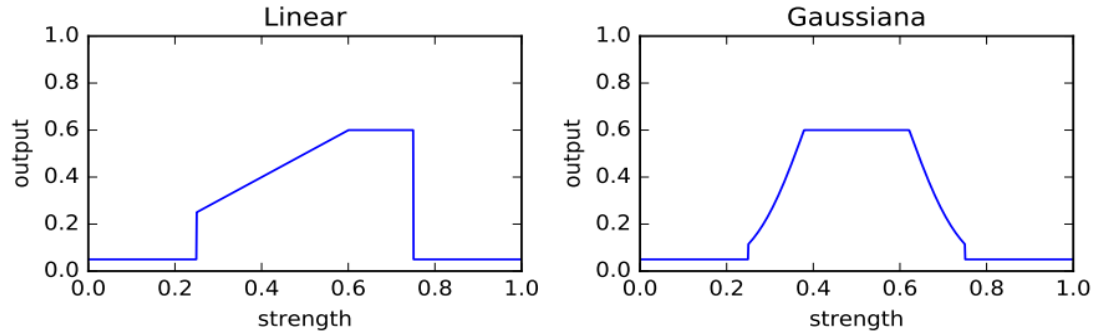

Figura 4 – Gráficos com limites de 0,05 e 0,6 aplicados.

2.2 - Comportamentos

Após conseguirmos desenvolver todas as funções necessárias, começámos a testar diferentes valores para tentar replicar os comportamentos requeridos no enunciado (círculo, infinito e elipse). **Mesmo conseguindo implementar um comportamento circular com a função de ativação linear**, o mesmo não foi possível para os restantes comportamentos devido às diversas limitações que a função linear impõe. Com isto decidimos que o melhor seria tentar implementar todos os comportamentos com uso da função gaussiana.

2.2.1 – Círculo

Após isso, decidimos que o círculo seria o comportamento mais fácil de reproduzir e por isso, foi o primeiro comportamento que tentámos demonstrar. Antes mesmo de começar a testar valores, podemos concluir que o veículo tem de ser do tipo "**lover**", pois para conseguirmos fazer um movimento circular a roda mais afastada da fonte de luz terá de ser a que tem maior energia e vice-versa.

Tendo isto em conta estes foram os diversos valores que testámos até conseguirmos executar um movimento circular:

Tabela 1 – Valores dos 11 testes executados para o movimento circular.

	Sensor	1	2	3	4	5	6	7	8	9	10	11
Min X	E	0.1	0.2	0.2	0.25	0.2	0.35	0.25	0.25	0.25	0.25	0.25
	D											
Max X	E	0.75	0.75	0.75	0.75	0.75	0.9	0.75	0.75	0.75	0.75	0.75
	D											
Min Y	E	0.05	0.05	0.05	0.3	0.2	0.3	0.3	0.3	0.3	0.3	0
	D											
Max Y	E	0.6	0.4	0.4	0.6	0.6	0.8	0.6	0.6	0.6	0.6	0.6
	D											
Std Dev	E	0.12	0.12	0.12	0.1	0.1	0.3	0.25	0.2	0.2	0.2	0.2
	D											
Mean	E	0.5	0.5	0.4	0.4	0.6	0.6	0.85	0.8	0.8	0.8	0.8
	D											
V. Max		20	20	20	20	20	20	20	20	50	45	20

Legenda: E – sensor esquerdo; **D** – sensor direito; **V. Max** – velocidade máxima; **Célula com cor vermelha** – não tem o comportamento pretendido; **Célula com cor amarela –** tem o comportamento pretendido, mas apresenta falhas; **Célula com cor verde –** tem o comportamento pretendido;

Tendo em conta que para o círculo os valores são iguais para ambos os sensores, obtivemos os seguintes resultados:

1. A velocidade da roda de fora do veículo varia constantemente entre x (velocidade variável) e 0, com isto o movimento circular é bastante defeituoso;
2. O veículo anda em linha reta e quando deteta a fonte de luz desvia-se da mesma;

3. O veículo faz um movimento circular, mas o raio é pequeno;
4. O veículo consegue fazer um círculo em sentido anti-horário, com um raio pequeno;
5. O veículo anda em linha reta e quando deteta a fonte de luz desvia-se da mesma;
6. O veículo consegue fazer um círculo grande em sentido horário;
7. O veículo consegue fazer um círculo algumas vezes, mas depois sai da trajetória;
8. O veículo consegue fazer um círculo em sentido anti-horário;
9. O veículo abandonava a trajetória;
10. O veículo consegue fazer um círculo em sentido anti-horário;
11. O veículo fica parado;

Ao analisar os resultados obtidos podemos verificar que, caso o veículo já execute um movimento circular, se alterarmos o **desvio padrão** (Std Dev) e a **média** (mean) para o dobro, o raio do círculo que o veículo faz também irá subir para o dobro, podemos verificar isso ao comparar os resultados dos testes **4** e **8.**

De seguida, concluímos também que, caso o veículo já execute um círculo, a velocidade máxima permitida para a variável **V. Max** é 45.

Descobrimos também que caso a variável **Min Y** seja 0, o veículo não vai executar nenhum movimento. Este comportamento pode ser explicado pelo facto de a fonte de luz não possuir alcance suficiente para fornecer energia ao veículo.

2.2.2 – Infinito

Após conseguirmos demonstrar o comportamento circular, começámos a testar valores para demonstrar o infinito. Tal como o veículo anterior, para conseguirmos demonstrar este tipo de movimento o veículo tem de ser do tipo "**lover**" pelos mesmos motivos já explicados anteriormente.

Tendo isto em conta estes foram os diversos valores que testámos até conseguirmos executar o infinito:

Tabela 2 – Valores dos 11 testes executados para o movimento do infinito.

	Sensor	1	2	3	4	5	6	7	8	9	10	11
Min X	E	0.25	0.25	0.25	0.3	0.25	0.45	0.55	0.25	0.25	0.25	0.25
	D											
Max X	E	0.65	0.65	0.65	0.75	0.75	0.75	0.75	0.65	0.75	0.75	0.75
	D											
Min Y	E	0.05	0.05	0.05	0.25	0.15	0.35	0.35	0.25	0.2	0.1	0.1
	D											
Max Y	E	0.5	0.5	0.5	0.9	0.6	0.9	1.5	0.65	0.6	0.6	0.6
	D											
Std Dev	E	0.15	0.2	0.2	0.2	0.35	0.2	0.2	0.05	0.35	0.295	0.295
	D											
Mean	E	0.6	0.6	0.4	0.6	0.8	0.6	0.7	0.45	0.75	0.715	0.715
	D											
V. Max		20	20	20	20	20	20	20	20	20	40	20

Tendo em conta que para o infinito os valores são iguais para ambos os sensores, obtivemos os seguintes resultados:

1. O veículo inverte o sentido e a velocidade da roda de fora do veículo varia constantemente entre x (velocidade variável) e 0, com isto o movimento é bastante defeituoso;
2. O veículo contorna as fontes de luz por fora e não consegue fazer o infinito;
3. O veículo anda em linha reta no sentido norte-este;
4. O veículo faz a primeira volta do infinito e segue em direção à luz;
5. O veículo faz o infinito com defeitos e achatado;
6. O veículo faz a primeira volta e quase consegue cruzar;
7. O veículo faz círculos na luz superior;
8. O veículo faz um círculo na luz superior, mas sai da trajetória;
9. O veículo faz um infinito quase perfeito;
10. O veículo faz um infinito, mas possuiu pequenos desvios na trajetória;
11. O veículo faz um infinito perfeito;

Ao analisar os resultados obtidos podemos verificar que, se o veículo estiver a demonstrar um comportamento do tipo "**amendoim**", ou seja, sem cruzar no meio das luzes, é necessário aumentarmos a média ("**Mean"**) e o desvio padrão ("**Std Dev"**) simultaneamente, tal como podemos observar entre os resultados dos testes **2** e **11**.

De seguida, concluímos também que, caso o veículo já execute um infinito, a velocidade máxima permitida para a variável **V. Max** tem de ser menor do que 40.

2.2.3 – Elipse

Por fim, só faltava demonstrar o comportamento da elipse, portanto começámos a testar valores para tal. Mais uma vez, para atingir o comportamento desejado, o veículo precisa de ser do tipo "**lover**" para adquirir mais energia na roda do lado de "fora".

Logo os valores obtidos com os nossos testes foram os seguintes:

Tabela 3 – Valores dos 6 testes executados para o movimento da elipse.

	Sensor	1	2	3	4	5	6	7
Min X	D	0.25	0.25	0.25	0.25	0.2	0.25	0.25
	E	0.25	0.25	0.25	0.25	0.2	0.25	0.25
Max X	D	0.75	0.75	0.75	0.75	0.75	0.75	0.75
	E	0.75	0.75	0.75	0.75	0.75	0.75	0.75
Min Y	D	0.4	0.35	0.35	0.4	0.2	0.4	0.4

	E	0.4	0.35	0.35	0.4	0.27	0.4	0.5
Max Y	**D**	0.4	0.35	0.35	0.4	0.6	0.4	1.4
	E	1.4	1.5	1.5	1.4	0.6	1.4	0.5
Std Dev	**D**	0.045	0.71	0.9	0.045	1.05	0.045	0.7
	E	0.531	0.6	0.42	0.531	0.6	0.531	0.05
Mean	**D**	0.375	0.35	0.35	0.375	0	0.375	0.7
	E	0.62	0.65	0.67	0.62	0.75	0.62	0.7
V. Max		15	15	15	14.45	20	16	16

Tendo em conta que para a elipse os valores podem e devem diferir para ambos os sensores, obtivemos os seguintes resultados:

1. O veículo faz uma linha reta com sentido norte;
2. O veículo faz um círculo e meio e começa a fazer um efeito tipo "mola" para a esquerda;
3. O veículo faz quase uma elipse;
4. O veículo faz apenas duas elipses;
5. O veículo faz uma elipse achatada;
6. O veículo faz quatro elipses;
7. O veículo faz um círculo e meio e começa a fazer um efeito tipo "mola" para a direita;
8.

Após vários testes não conseguimos encontrar nenhuma gama de valores que nos permitisse observar um movimento de elipse perfeito. Como foi referido acima, o nosso melhor resultado foi observar o veículo a fazer 4 elipses antes de perder a trajetória, movimento que pode ser explicado pela incapacidade de o veículo descrever a primeira curva de uma maneira perfeita, pois caso o mesmo não aconteça a elipse começará a ficar cada vez mais larga a cada volta efetuada.

No entanto, se verificarmos os resultados dos testes **1**, **4** e **6** podemos verificar que a velocidade pode melhorar a execução da primeira curva, por exemplo, ao aumentarmos ligeiramente a velocidade do teste **4** obtermos o comportamento observado no teste **6** que já se assemelha ao resultado pretendido.

2.3 - Mundos

Finalmente, após implementar com alguma fiabilidade todos os comportamentos, decidimos começar a contruir um mundo capaz de testar a maioria das propriedades dos nossos veículos. Para isso a nossa principal ideia foi desenvolver um labirinto (**figura 5**).

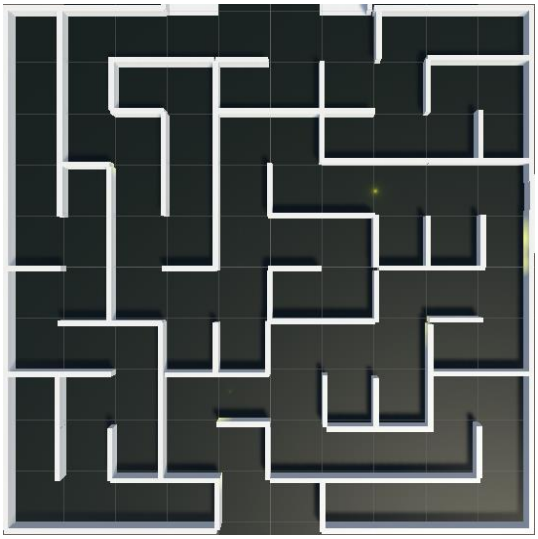

Figura 5 – Labirinto.

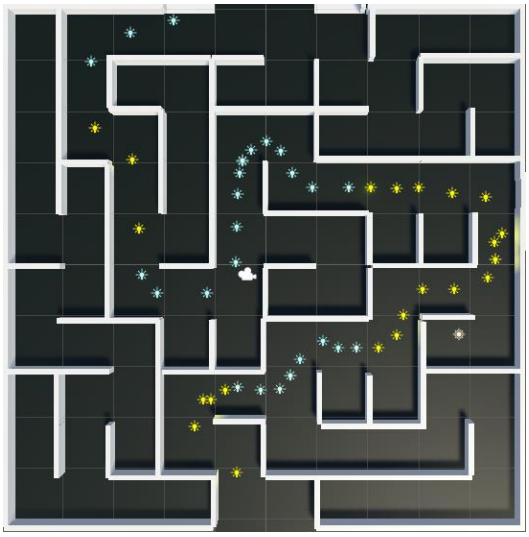

Figura 6 – Labirinto com rasto de luzes.

Com o principal cenário construído, recorrendo a um gerador de labirintos online, decidimos mostrar que os nossos veículos conseguiam detetar luzes e blocos para se movimentarem. De modo a completar o labirinto, o nosso veículo irá seguir um rasto de luzes (**figura 6**), isto é possível pois com recurso à função de ativação linear conseguimos fazer com que o veículo tenho um ganho de energia constante sempre que detete luzes num ângulo de 180º à sua frente. Para o veículo conseguir terminar todo o percurso foi necessário ter especial atenção à posição das luzes em curvas apertadas para termos a certeza que o veículo tem energia suficiente para as realizar. Outra nota importante é que o veículo precisa de ser do tipo "**lover**", ou seja, sensor esquerdo ligado à roda direita e vice-versa, pois nós queremos que o mesmo vá de encontro às luzes e não o oposto.

De seguida, para o veículo conseguir fazer uma trajetória suave sem bater, decidimos também aplicar uma função de ativação linear para detetar paredes (blocos), no entanto, contrariamente à deteção de luzes tivemos de aplicar um comportamento do tipo "**explorer**", ou seja, ligar o sensor esquerdo à roda esquerda e sensor direito à roda direita. Deste modo ao detetar paredes o nosso veículo irá ganhar velocidade e afastar-se-á da parede. De notar que tal só é possível se aplicarmos a tag "**block**" às paredes.

Para finalizar, decidimos mostrar ainda outro comportamento no mesmo cenário. No final do labirinto irão se encontrar 6 outros veículos que irão estar à espera do veículo principal, esses veículos serão os "guardas" do tesouro que se encontra na última sala, logo para serem guardas eficazes irão detetar o veículo principal com recurso à tag "**CarToFollow**" e a sensores de carros com funções de ativação linear (**figura 7**).

Por fim o principal objetivo do veículo principal será chegar à fonte de luz que se encontra dentro do baú, sendo que para isso terá de se desviar dos seis "guardas", decidimos ser uma implementação interessante de sensores de veículos pois o resultado da simulação pode diferir de teste para teste.

Figura 7 – Sala final com os guardas.

Nota: Algumas figuras do relatório encontram-se desatualizadas devido a aplicação de texturas.

Nota: Foi implementado um script para movimentar a camara no sentido do eixo Z (Teclas: W, A).

Conclusão

Com o decorrer do projeto, foi possível cimentar os conteúdos aprendidos nas aulas uma vez que para a sua realização eminenciou-se a necessidade de uma testagem exaustiva. No entanto, a testagem foi um processo desgastante muito porque foi difícil obter os resultados desejados. Por outro lado, a criação de um mundo para testar as funcionalidades dos sensores/veículos foi uma atividade interessante porque requeria o nosso lado mais criativo e ao mesmo tempo, foi uma forma de visualizar todos os comportamentos em simultâneo.

Bibliografia

- https://pt.wikipedia.org/wiki/Fun%C3%A7%C3%A3o_de_Gauss - Função de ativação gaussiana
- https://www.mazegenerator.net/ - Gerador de labirintos
- https://assetstore.unity.com/packages/3d/props/interior/treasure-set-free-chest-72345 - Modelo do baú
- Stone Wall Texture Pack 01 #Texture#Pack#Stone#Wall | Stone wall texture, Texture painting, Hand painted textures (pinterest.com) – Textura de paredes
- Vídeo para Feed&Play em anexo