

Relatório Trabalho Prático 1

Braitenberg Vehicles



UNIVERSIDADE D
COIMBRA

Fundamentos da Inteligência Artificial
Licenciatura em Engenharia Informática
2022/2023

Filipe David Amado Mendes, nº 2020218797, filipe.mendes@student.uc.pt, PL1
João Miguel Fernandes Moura , nº 2020235800, joaomoura@student.dei.uc.pt, PL4

Índice

| | | |
|--------|----------------------------|----|
| 1. | Meta 1 - Sense it | 1 |
| 2. | Meta 2 - Tune it & Test it | 1 |
| 2.1. | Funções de ativação | 2 |
| 2.1.1. | Função Linear | 2 |
| 2.1.2. | Função Gaussiana | 2 |
| 2.2. | Limites e Limiares | 3 |
| 2.2.1. | Limiares | 3 |
| 2.2.2. | Limites | 3 |
| 2.3. | Cenários e Trajetórias | 4 |
| 2.3.1. | Círculo | 4 |
| 2.3.2. | Infinito | 5 |
| 2.3.3. | Elipse | 7 |
| 2.3.4. | Gran Turismo | 8 |
| 2.4. | Vídeos Demonstrativos | 10 |

1. Meta 1 - Sense it

Na primeira meta, era pedido que aumentássemos e explorássemos as capacidades dos veículos já disponibilizados. O código que nos foi dado, trazia já implementados os sensores de luz, permitindo aos veículos seguirem as luzes pontuais. Para concluir a meta, implementámos o sensor de carros e o sensor de blocos. A implementação destes baseou-se na lógica do sensor de luz, captando os objetos com as tags definidas e processando o output consoante a função escolhida.

Para o cálculo da energia foi utilizada a fórmula:

$$energia = \sum_{i=1}^n \frac{1}{\frac{distancia(p_i, sensor)^2}{r_i} + 1}$$

Esta meta permitiu perceber a lógica de nível mais básico por trás dos veículos de Braitenberg. Explorando um pouco os diferentes tipos de output e as diferentes implementações dos sensores, conseguimos entender um pouco sobre os 4 comportamentos possíveis dos veículos, medo, agressão, amor e exploração, explicados mais à frente.

2. Meta 2 - Tune it & Test it

Na meta final, foi-nos solicitado que implementássemos veículos capazes de recriar alguns trajetos detectando e seguindo as luzes, entre os quais um círculo, um infinito e uma elipse. Para tal, foi

necessário alterar um pouco a estrutura de código dos sensores de luz, implementando as duas funções de ativação dos sensores, linear e gaussiana e os seus respectivos limites e limiares.

2.1 Funções de ativação

2.1.1 Função Linear

A função de ativação linear, tal como demonstra o gráfico abaixo, devolve sempre um output diretamente proporcional ao valor da energia. Assim, a implementação em código não passa do cálculo da energia do sensor.

$$y = f(x) = x$$

Além disso, tal como será explicado posteriormente no relatório, foi também implementada a verificação dos limiares e limites, caso o utilizador o deseje. A fórmula para o cálculo da energia utilizada foi a seguinte:

$$energia = \sum_{i=1}^n \frac{1}{\frac{distancia(p_i, sensor)^2}{r_i} + 1}$$

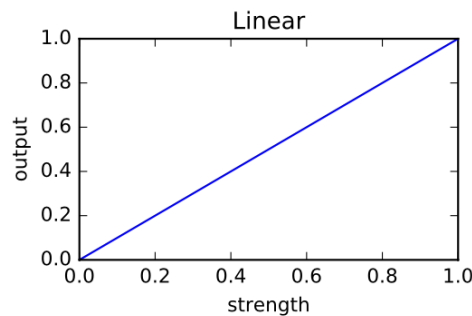


Figura 1 - Função Linear

2.1.2 Função Gaussiana

A função de ativação gaussiana, tal como a linear, recebe como parâmetro o valor da energia do sensor. Porém, além deste, recebe também outros dois: um valor de média (μ) e um desvio padrão (σ). A fórmula utilizada para o cálculo da função foi a seguinte:

$$G(energia) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{energia^2}{2\sigma^2}}$$

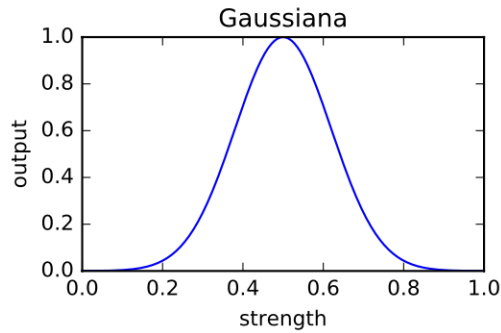


Figura 2 - Função Gaussiana

2.2 Limites e Limiares

2.2.1 Limiares

A implementação dos limiares e dos limites, ao contrário do que esperávamos, seguiu uma lógica simples e fácil de entender. Para os limiares, sendo estes os delimitadores da função de ativação nos valores de X , a lógica de pensamento passou muito por analisar os gráficos de funções lineares e gaussianas que nos foram disponibilizados. Concluimos que, para ambas as funções, caso os valores de X sejam superiores aos limiares, o valor resultante da função é o valor mínimo de Y . Podemos chegar a esta conclusão analisando o gráfico abaixo, onde é evidente que fora os limiares, mínimo de 0.25 e máximo de 0.75, as funções têm o mesmo valor: mínimo de $Y = 0.05$. Assim a implementação do código foi simples, verificando apenas se o output está dentro dos limiares e caso não esteja aplica o valor mínimo de Y .

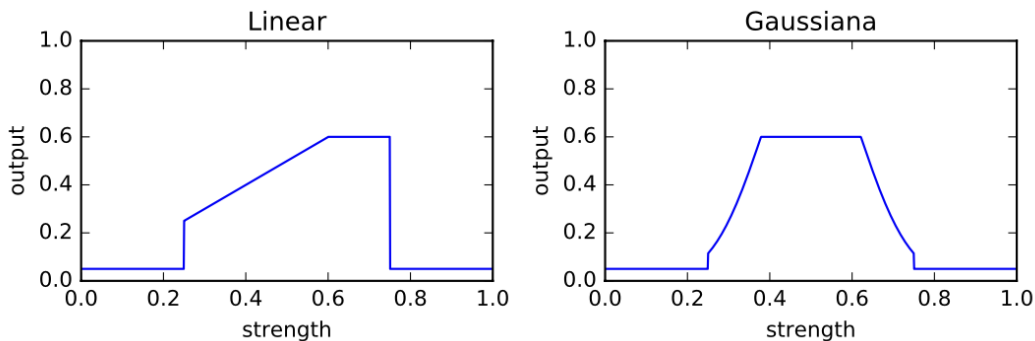


Figura 3 - Funções de ativação linear e gaussiana com limiares 0.25, 0.75 e limites 0.05, 0.6

2.2.2 Limites

A implementação dos limites seguiu a mesma lógica. Estes delimitam a função nos valores do Y , ou seja, o resultado da função de ativação. Assim, garantimos que o resultado esperado pela função nunca é superior ou inferior a um dado valor. Analisando o gráfico acima podemos ver bem definidos os limites, mínimo de 0.05 e máximo de 0.6. Isto levou-nos a um código com apenas condições de `if` e `else`, verificando se o output está dentro dos limites e, caso não esteja, se for inferior ao mínimo aplica-se o valor mínimo e se for superior ao máximo aplica-se o valor máximo.

2.3 Cenários e Trajetórias

Nesta fase do projeto foi importante definirmos a ordem pela qual iríamos trabalhar. Decidimos começar pelas trajetórias, primeiro o círculo, depois implementar o infinito, de seguida a elipse e apenas por último o cenário mais complexo de corrida (Gran Turismo). Esta ordem permitiu-nos entender melhor a forma como cada um dos sensores e respetivas funções de ativação funcionam, bem como o tipo de veículo que deveríamos usar para cada um dos casos.

Existem quatro tipos de comportamentos que podem ser aplicados aos veículos de Braitenberg: medo, agressão, amor e exploração. Medo e agressão resultam de um output aditivo, enquanto que amor e exploração resultam de um output subtrativo. Existe ainda outra distinção entre os comportamentos. Para implementar um veículo com um comportamento de medo ou de amor, associam-se os sensores às respetivas rodas, ou seja, sensor direito associado à roda direita e sensor esquerdo associado à roda esquerda. Por outro lado, para implementar um veículo com um comportamento de agressão ou de exploração, os sensores são invertidos, associando o sensor direito à roda esquerda e o sensor esquerdo à roda direita. Esta distinção de comportamentos é crucial na execução dos circuitos, sendo muitas vezes impossível executar uma trajetória com mais do que um comportamento.

A ordem que definimos ajudou-nos a perceber como cada um destes veículos se comporta e de que forma cada um deles é afetado pelo ambiente que o rodeia. Assim, conseguimos tirar as nossas conclusões e adequar a nossa abordagem a cada um dos cenários do trabalho.

2.3.1 Círculo

Para implementar esta cena, a nossa abordagem foi utilizar a função de ativação gaussiana para ambos os sensores. Para que o carro contorne a luz, uma das rodas tem que andar mais, a roda de fora. Para tal, a função gaussiana, com os sensores cruzados, permitiu realizar a trajetória pretendida sem movimentos inesperados.

Para a posição da luz pontal $x = 1$, $y = 1$, $z = 5$, com uma velocidade máxima de 5, foram definidos em ambos os sensores os mesmos limites e limiares:

$$\text{MinX} = 0.5, \text{MaxX} = 2, \text{MinY} = 1, \text{MaxY} = 3$$

com uma média e desvio padrão de:

$$\mu = 0.7, \sigma = 0.17$$

Depois de vários testes, ficou claro que mesmo pequenas alterações nos valores acima referidos podem ter consequências drásticas e alterar totalmente o resultado obtido. Concluímos que diminuindo o valor da média o carro percorria uma trajetória mais alargada, mais longe da luz, e com uma velocidade mais reduzida. Por outro lado, aumentando o valor da média, permitiu descrever o círculo mais próximo da luz e com uma velocidade final mais elevada, tal como demonstra a Figura 4.

No que diz respeito ao desvio padrão, não conseguimos relacionar a subida ou descida do seu valor com o comportamento do carro, apenas analisámos que, diminuindo demasiado o desvio padrão, o carro apenas segue em frente, deixando de girar em torno da luz.

Analisando os valores de MinX e MaxX concluímos que ao diminuir o valor de MinX, por exemplo até 0, o carro segue uma trajetória mais alargada e mais rápida. Por outro lado, aumentando o valor, a trajetória torna-se mais lenta e afunilada e, a certo ponto, o carro deixa de girar, apenas seguindo em frente. Quanto aos valores de MaxX, pelas experiências efetuadas, não conseguimos notar diferenças significativas, exceto algumas diferenças na velocidade final do carro, sendo estas, porém, um pouco aleatórias.

Por outro lado, os valores de MinY e MaxY são mais constantes. Diminuindo o valor de MinY, por exemplo para 0.5, o veículo flete um pouco no sentido contrário ao do círculo, porém ajusta a trajetória. Aumentando, o veículo segue em frente. Quanto ao valor de MaxY, o aumento do valor não mostra consequências no desempenho do carro, enquanto que a diminuição em demasia leva o carro a seguir em frente.

Com os resultados obtidos, ficou claro que a escolha dos valores a usar não é única, havendo mais que uma combinação de valores que levam a resultados semelhantes. Assim, a escolha dos valores que usamos, deve se à preferência do resultado obtido, em comparação com outras hipóteses que testamos.

Para estes valores, a trajetória descrita pelo veículo foi a seguinte:

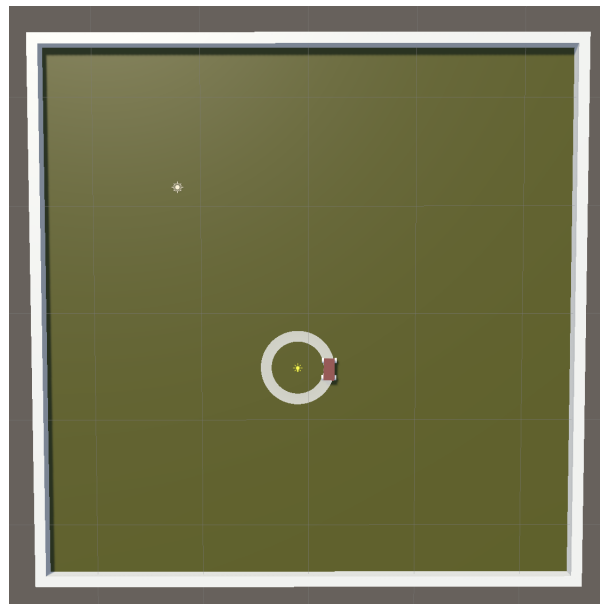


Figura 4 - Círculo concluído

2.3.2 Infinito

Para que o carro desenhe o infinito, seguimos a mesma lógica da cena anterior, usando sensores cruzados e novamente a função de ativação gaussiana, para ambos.

Para a posição das luzes pontuais $x = 0, y = 1, z = 5$ e $x = 0, y = 1, z = -5$, com uma velocidade máxima de 5, foram definidos em ambos os sensores os mesmos limites:

$$\text{MinY} = 0.5, \text{MaxY} = 1.8$$

com uma média e desvio padrão de:

$$\mu = 0.7, \sigma = 0.17$$

Ao contrário do círculo, optámos por não usar limiares e manter como delimitadores apenas os limites, isto é, MinY e MaxY. Esta escolha deve-se ao facto de nas primeiras experiências, usando apenas limites, a trajetória do carro estar praticamente correta. Assim, preferimos apenas testar e melhorar a trajetória do carro alterando os valores que já estavam a ser utilizados, evitando o risco de não conseguirmos completar esta cena da Meta 2.

Quanto aos limites, analisámos que as consequências da alteração dos valores não são tão acentuadas como no círculo. A alteração do valor de MinY dentro do intervalo de $[0, 1]$, não tem grandes repercussões na trajetórias, porém o aumento abrupto, por exemplo $\text{MinY} = 3$, faz com que o carro saia da trajetória e siga em frente. Por outro lado, concluímos que a diminuição do valor de MaxY, leva o carro a desenhar uma trajetória mais alargada, acabando, a certo ponto, por não cruzar o centro e não completar o infinito. Percebemos também, que o aumento do valor de MaxY, leva o carro a desenhar uma trajetória mais apertada e, caso o aumento seja demasiado, deixa de completar a trajetória corretamente.

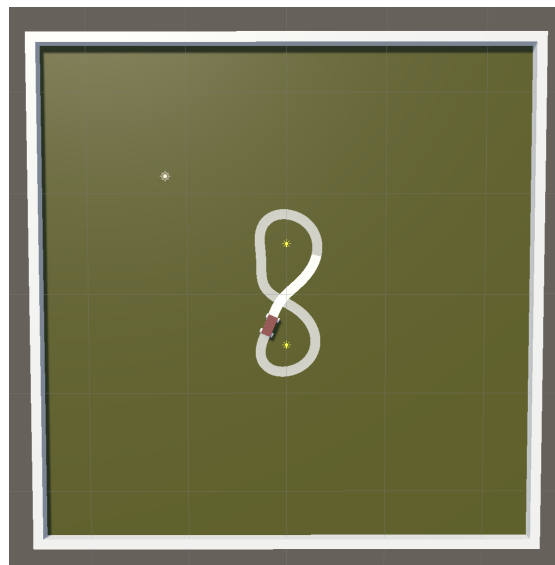


Figura 5 - Infinito com valor de $\text{MaxY} = 2.2$ no Sensor Gauss Esquerdo

Para os valores da média e do desvio padrão, começamos por testar os valores dados no enunciado, porém, sem resultado nenhum, decidimos testar os valores que tínhamos utilizado para a cena do círculo. Apesar de termos obtido resultados perto do esperado, antes de alterarmos os limites, decidimos testar valores diferentes para o desvio e a média. Concluímos que aumentando o desvio padrão a trajetória alarga, e diminuindo tem o efeito contrário, a trajetória afunila. Por outro lado, aumentando a média, leva

o carro a afunilar a trajetória e manter um círculo em torno da luz superior. Diminuindo a média, o carro alarga a trajetória e acaba por seguir em frente. Assim, decidimos manter os valores utilizados também para o círculo, visto que estes mostraram os melhores resultados.

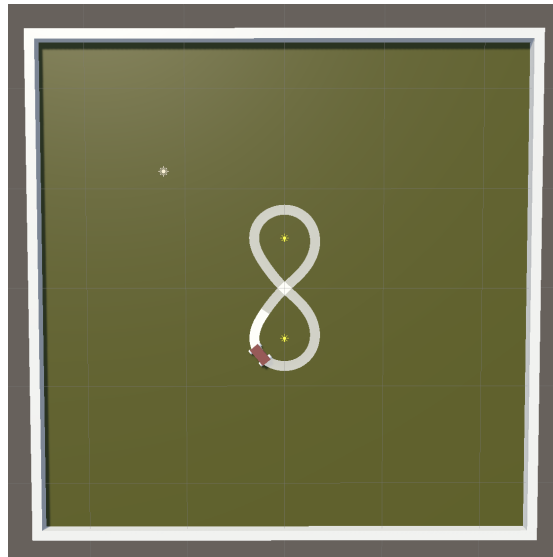


Figura 5 - Infinito Concluído

2.3.3 Elipse

A última das trajetórias a implementar foi a elipse. Nesta implementação seguimos uma lógica um pouco diferente. Tal como no círculo e no infinito usamos os sensores cruzados, visto que para curvar precisamos que uma das rodas ande mais que a outra. Porém, ao contrário das outras trajetórias, decidimos utilizar a função de ativação linear. Fizemos esta escolha, pois, depois de muitas experiências, concluímos que a escolha de valores para a função linear é mais simples e direta do que para a função gaussiana.

Assim, para a posição das luzes pontuais $x = 0, y = 1, z = 8$ e $x = 0, y = 1, z = -8$, com uma velocidade máxima de 30, definimos os sensores com os seguintes limites limiares:

Sensor Linear Direito: $\text{MinX} = 0, \text{MaxX} = 0.29, \text{MinY} = 0.2, \text{MaxY} = 0.28$

Sensor Linear Esquerdo: $\text{MinX} = 0, \text{MaxX} = 0.31, \text{MinY} = 0.29, \text{MaxY} = 0.3$

A escolha destes valores foi mais ponderada e minuciosa que a escolha dos valores para os circuitos anteriores. Como sempre, para que o carro vire, é necessário que uma das rodas tenha mais energia que a outra, neste caso a direita. Porém, como as funções de ativação dos sensores são lineares, esta diferença de energia entre as rodas teve que ser conseguida através dos limites. Assim definiu a roda direita com valores no intervalo $[0.29, 0.3]$ e a roda esquerda com valores no intervalo $[0.2, 0.28]$, sendo estes, depois de muitos testes, os valores que mostraram melhores resultados.

A alteração do valor MaxX, em ambos os sensores, mostrou também grandes diferenças no resultado. O aumento do valor permite o carro alongar a sua trajetória, chegando mais perto das luzes. Porém, leva também a uma trajetória pouco consistente, apresentando uma forma quase losangular, quando o valor de MaxX é superior a 0.36.

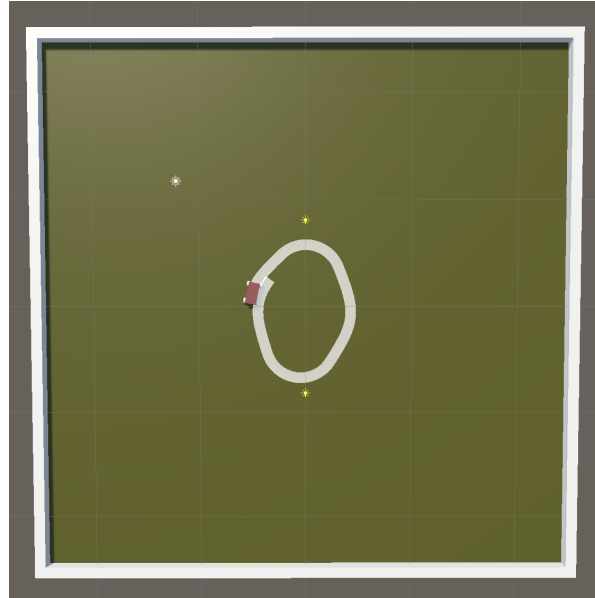


Figura 6 - MaxX = 0.4 no Sensor Linear Direito

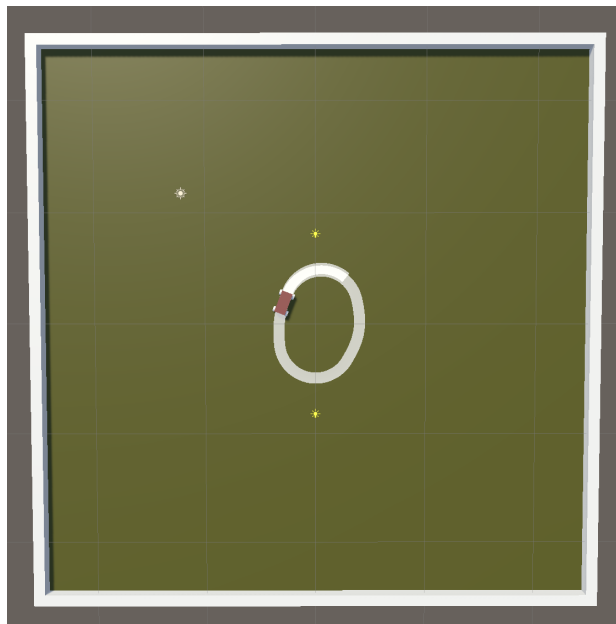


Figura 7 - Elipse completa

2.3.4 Gran Turismo

Neste circuito decidimos construir uma pista, em que dois carros tem como objetivo seguir as luzes até chegar ao meio, começando em lados opostos, “drivers”. Existe ainda um terceiro carro que tem apenas como objetivo seguir o segundo, “safety car”.

Os três carros foram implementados com um comportamento agressivo, aumentando a sua velocidade quando se dirigem ao seu alvo, as luzes, no caso dos dois “drivers”, ou o segundo carro, no caso do “safety car”.

Para a implementação deste circuito, decidimos usar funções de ativação lineares. Tendo este tipo de funções um output constante, a trajetória que os carros seguem torna-se mais simples de prever. Assim, os testes que realizamos foram focados maioritariamente na posição das luzes e o seu efeito no mapa.

Além dos sensores de luz, os “drivers” têm também implementados sensores de blocos. Neste tipo de sensores, ao contrário dos das luzes, os carros têm um comportamento de medo, tentando fugir de todos os objetos com a tag “Block”. Esta implementação permite que os carros completem a pista evitando tocar nas paredes. Por outro lado, o “safety car” tem implementado o sensor de carros, com os sensores lineares de output aditivo cruzados, dando ao carro o comportamento agressivo referido anteriormente.

Ao contrário da implementação dos “drivers”, a implementação do “safety car” não foi tão bem sucedida. O carro mostra um comportamento agressivo em relação ao “driver” à sua frente. Porém, como o “driver” se afasta rapidamente, seguindo as luzes do circuito, o “safety car” tem dificuldade em segui-lo, pois o output diminui drasticamente, acabando por seguir apenas em linha reta.

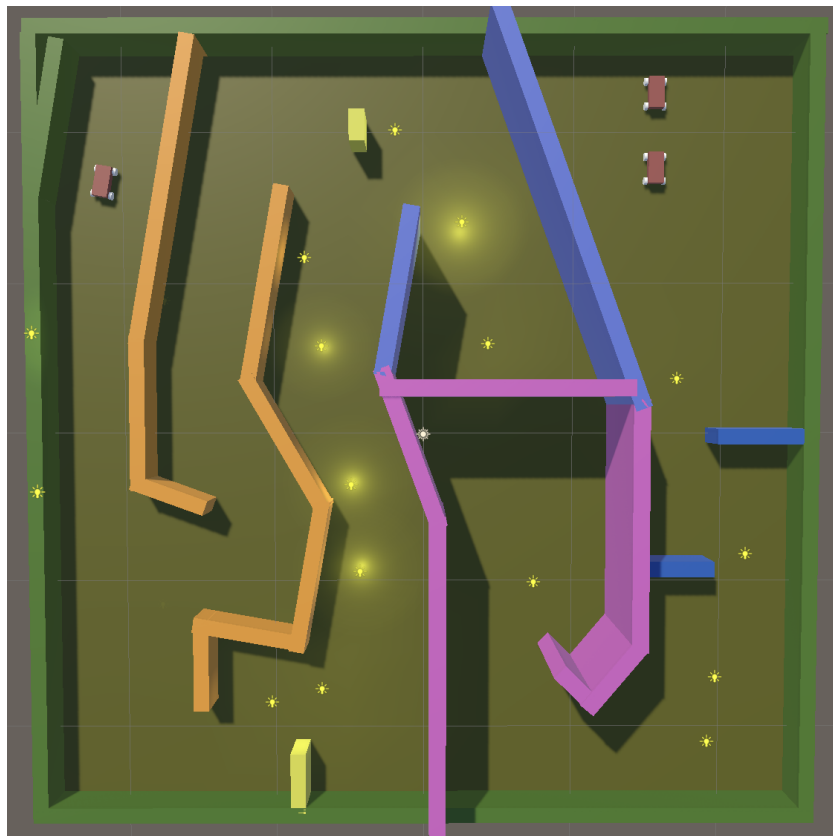


Figura 8 - Circuito Gran Turismo

2.4 Vídeos demonstrativos

Para ver o comportamento de todos os circuitos e carros descritos no relatório, podem ser conferidos os vídeos demonstrativos enviados na pasta “Vídeos”.

Os vídeos correspondentes a cada circuito são os seguintes:

2.3.1 Círculo - Círculo.mov

2.3.2 Infinito - Infinito.mov

2.3.3 Elipse - Elipse.mov

2.3.4 Gran Turismo - GranTurismo.mov