



Universidade de Coimbra
Faculdade de Ciências e Tecnologia Universidade de Coimbra
Licenciatura em Engenharia Informática

Fundamentos de Inteligência Artificial

Veículos de Braitenberg

Projeto realizado por:

Alexandre van Velze – 2019216618 - vanvelze@student.uc.pt – pl5
João Carlos Silva – 2019216753 - joaosilvba@student.dei.uc.pt – pl5
Sofia Alves – 2019227240 - sbalves@student.dei.uc.pt – pl5

Professor orientador:

Nuno Lourenço

1. Introdução

No âmbito da cadeira de Fundamentos de Inteligência Artificial, este projeto visa estudar os agentes reativos, baseando-se nos veículos de Braitenberg, que possuem como objetivo representar dados comportamentos complexos, como o medo, agressividade, etc. Para analisar este tipo de comportamentos, foram desenvolvidos vários cenários com recurso ao software *Unity*.

2. Meta 1 - Sense It

Para esta meta, foi implementada uma funcionalidade que permite aos veículos detectarem outros com base na distância a que estão dos mesmos, tendo sido realizados testes para validar a mesma. Esta consta no ficheiro `CarDetectorScript.cs`.

Com esta finalidade em mente, criámos a função **GetClosestCar** que irá calcular, sucessivamente, a distância entre o carro atual e os restantes carros que se encontram na *Scene*, detentores da tag “CarToFollow”. De seguida, guardamos o índice correspondente ao carro mais próximo na variável **min**, que irá ser retornada no final da função.

Esta função irá ser chamada na função **Update**, que é invocada a cada frame, obtendo, assim, o carro mais próximo. Sabendo o carro mais próximo, calculamos o **output** (energia) que irá colocar o carro em movimento. Para tal, usamos a seguinte fórmula:

$$output = \frac{1}{r + 1}, \text{ sendo } r \text{ a distância até ao carro mais próximo}$$

Cenários usados para testar a funcionalidade implementada:

1. Carro sensível à luz (contendo a tag “CarToFollow”) com carro que deteta proximidade atrás do mesmo.
2. Carro sensível à luz (contendo a tag “CarToFollow”) com dois carros que detetam proximidade atrás do mesmo, sendo que um deles está mais próximo que o outro.
3. Dois carros que detetam proximidade, ambos contendo a tag “CarToFollow”.

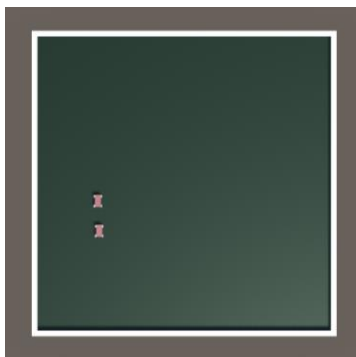


Fig. 1



Fig. 2

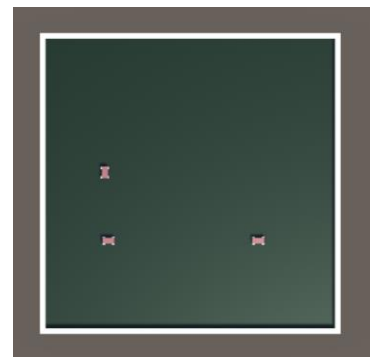


Fig. 3

3. Meta 2 - Tune it & Test it

Para esta meta, foi desenvolvida a função de ativação Gaussiana, bem como os respectivos limites e limiares (sendo que estes são aplicáveis em ambas as funções de ativação). Para funcionamento de certos cenários implementamos ainda a funcionalidade da inversa. Estas funcionalidades implementadas foram ainda testadas para vários cenários, os quais iremos falar mais à frente.

3.1 Funções de ativação

Cada veículo poderá usar dois tipos de funções de ativação: **linear** e **gaussiana**.

A primeira foi implementada na meta anterior e funciona de forma bastante intuitiva.

Temos uma relação diretamente proporcional entre a energia calculada pelos sensores e a aceleração com que os veículos se deslocam nos diferentes cenários, isto é, a quantidade de energia recebida será a mesma de aceleração.

A função é dada pela seguinte expressão:

$$f(x) = x, \text{ sendo } x \text{ a energia calculada pelos sensores}$$

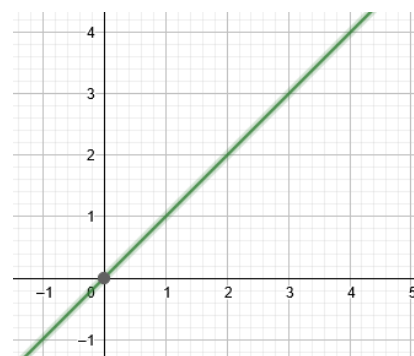


Fig. 4 – Função linear

A função gaussiana já funcionará de forma mais complicada, no sentido em que, a relação entre a energia e a aceleração dos veículos não será direta.

Para o cálculo da aceleração é preciso ter ainda em conta que média e desvio padrão a utilizar, pois estes serão determinantes no comportamento do veículo.

A função é dada pela seguinte expressão:

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

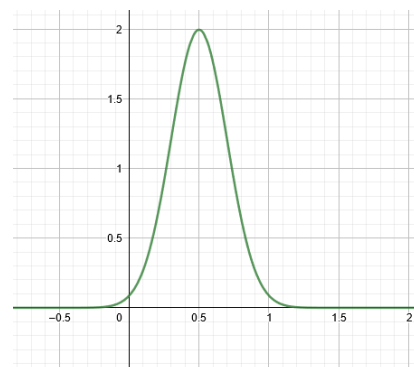


Fig. 5 – Função Gaussiana

onde x é a energia calculada pelos sensores, μ a média e σ o desvio padrão

Como referido, a mudança dos parâmetros de média e desvio padrão terá influência no comportamento dos veículos.

A variação da média implicará alterar o intervalo de valores de energia onde o veículo será mais ativo, no fundo teremos um deslocamento da função gaussiana segundo o eixo dos x 's. Esta transformação pode ser representada através da seguinte relação:

$$f(x) = g(x - (\mu_2 - \mu_1)),$$

μ_2 seria a nova média aplicada, μ_1 a média antiga e $g(x)$ a função antiga

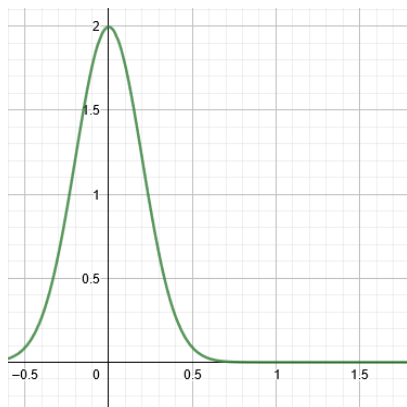


Fig. 6 – Função Gaussiana com média 0

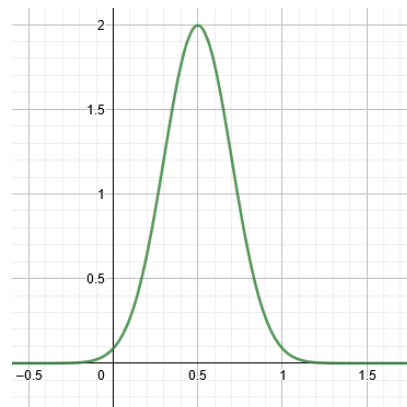


Fig. 7 – Função Gaussiana com média 0.5

Já com a variação do desvio padrão, as mudanças provocadas serão um pouco diferentes. Quanto maior o desvio padrão, menor a variação da aceleração, ou seja, para valores de energia diferentes, a aceleração será sempre muito parecida. Segue-se um exemplo onde o desvio padrão é igual a 10:

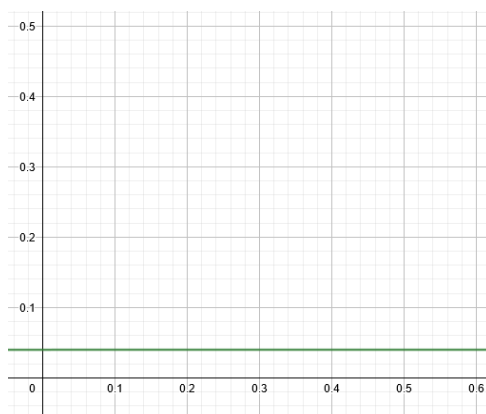
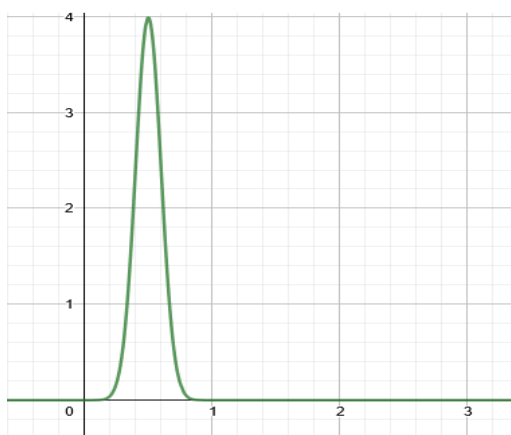


Fig. 8 – Função Gaussiana com desvio padrão 10

Utilizando um desvio padrão mais pequeno poderemos obter a certa altura um pico de aceleração (nomeadamente quando a energia for igual à média) que permitirá em certas situações ser utilizado por exemplo: para fazer curvas mais acentuadas, mudanças repentinas de velocidade. De seguida temos um exemplo onde o desvio padrão será 0.1:

Fig. 9 – Função Gaussiana com desvio padrão 0.1



Esta relação entre o pico de aceleração e o desvio padrão deve-se ao facto de que como o desvio padrão se apresenta sempre em denominadores na função gaussiana quanto maior o seu valor mais baixa será a variação de aceleração e quanto menor mais alta será essa variação.

3.2. Inversa

A inversa permite obter novos comportamentos ao inverter o output dos sensores, esta funcionalidade só é utilizada no CarDetectorScript.cs Com a função fornecida para o cálculo da energia obtemos um comportamento onde, à medida que a distância aumenta, a energia diminui, ou seja, uma relação de proporcionalidade inversa (Figura 10). Assim, utilizando esta funcionalidade, o comportamento será o contrário, isto é, à medida que a distância aumenta, a energia aumenta também (Figura 11).

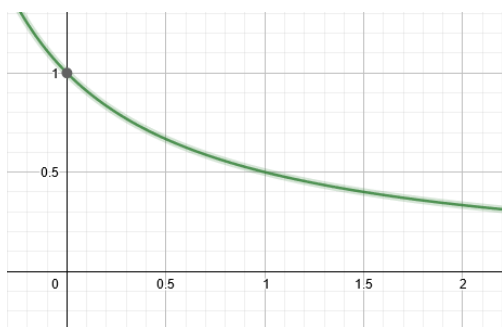


Fig. 10

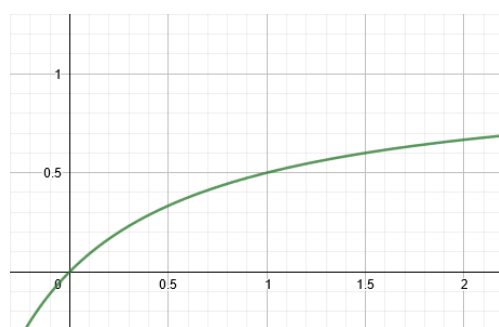


Fig. 11

3.3. Limites e Limiares

3.3.1. Limiares

Antes de calcular a função de ativação, é necessário verificar o valor atual de x. Caso este seja inferior ou superior aos valores fornecidos (**minX** e **maxX**, respetivamente), a aceleração resultante será 0. Caso contrário, procedemos ao cálculo da função de ativação.

3.3.2. Limites

Depois de aplicados os limiares (se aplicável), podemos também usar os limites, verificando o valor resultante da função de ativação. Se este for inferior ou superior aos valores indicados (**minY** e **maxY** respetivamente), igualamos a aceleração a um destes limites.

3.4. Cenários Testados

Para testar as funcionalidades implementadas, foi-nos proposto recriar três cenários: um círculo, um infinito e uma elipse, usando os veículos de Braitenberg. Além disso, foi também proposto a criação de outros cenários à nossa escolha.

Neste último, decidimos recriar um labirinto (medo), uma pista de corrida (agressivo) e uma perseguição de um carro da polícia (agressivo + lover).

Para cada um destes cenários, foi necessário ajustar os valores dos limites, limiares, média, desvio padrão, velocidade máxima e ainda características das luzes. Para tal, foi necessário a realização de vários testes, sendo que a seguir apresentam-se os valores alcançados com os mesmos e algumas observações sobre os mesmos.

3.4.1. Cenários Propostos

Círculo

Função de ativação: Gaussiana

Velocidade máxima: 15

Sensores de Luz (valores usados são iguais para ambas as rodas):

- Desvio Padrão: 0.12
- Média: 0.5

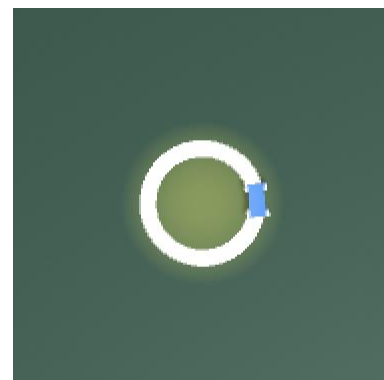


Fig. 12 – Círculo final

Para a realização desta trajetória, foram apenas utilizados os valores referidos no enunciado. Para efeitos de testagem, decidimos mudar os valores da média e do desvio padrão. Conseguimos observar que, ao aumentar muito a média, a certa altura o veículo deixa de andar. Tal deve-se ao facto de a função gaussiana inicialmente possuir valores muito perto de 0. Se aumentarmos muito a média, o nosso veículo irá começar com uma aceleração igual a 0, nunca conseguindo absorver energia suficiente para começar a mover-se. Assim, com um desvio padrão de 0.12, conseguimos valores para a média até 0.7, acima deste, o carro não se move (Figura 13). Optamos, no entanto, por continuar com o valor 0.5, visto que o círculo desenhado pelo carro é maior e, portanto, mais perceptível.

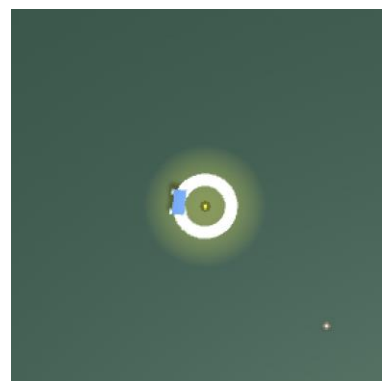


Fig. 13 – Círculo com média 0.7

Infinito

Função de ativação: Gaussiana

Velocidade máxima: 15

Sensores de Luz (valores usados são iguais em ambas as rodas):

- Limite inferior: 0
- Limite superior: 1.2
- Desvio Padrão: 0.12
- Média: 0.5

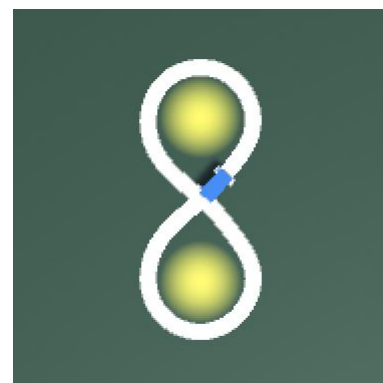


Fig. 14 - Infinito final

Inicialmente foi testado qual seria o resultado sem aplicar qualquer tipo de restrições à função de ativação. O mesmo ficou muito aquém do esperado. Ao aplicar um limite superior de 0.5, o resultado obtido é semelhante ao infinito, uma melhoria significativa. (Figura 15). Aumentando este limite para 1, a trajetória é praticamente igual ao esperado

(Figura 16). No entanto, foram testados valores superiores na procura de um melhor resultado, chegando à conclusão que 1.2 é o valor ideal, dado que acima destes valores o símbolo desfaz-se. O limite inferior não é relevante neste cenário, dado que a sua utilização não ajuda a alcançar o objetivo.



Fig. 15 – Infinito com limite superior 0.5



Fig. 16 – Infinito com limite superior 1

Elipse

Função de ativação: Gaussiana

Velocidade máxima: 15

Sensores de Luz:

- Roda direita:
 - Limite inferior: 0.3
 - Limite superior: 0.5
 - Desvio Padrão: 0.2
 - Média: 0.5
- Roda esquerda:
 - Limite inferior: 0.02
 - Limite superior: 0.385
 - Desvio Padrão: 0.2
 - Média: 0.5

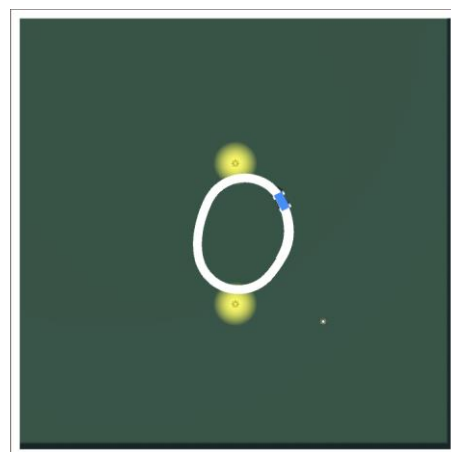


Fig. 17 – Elipse final

Na escolha dos valores para a elipse, a abordagem tomada para chegar aos valores acima descritos consistiu em imprimir os valores da aceleração (guardados na variável **output1**) e da energia (guardada na variável **output**), limitando assim a gama de valores à escolha.

Comparando com a roda esquerda, os limites definidos para os sensores da roda direita são superiores. O objetivo é fazer com que o carro vire para a esquerda na direção da luz posicionada mais acima, fazendo com que a roda da direita receba mais energia.

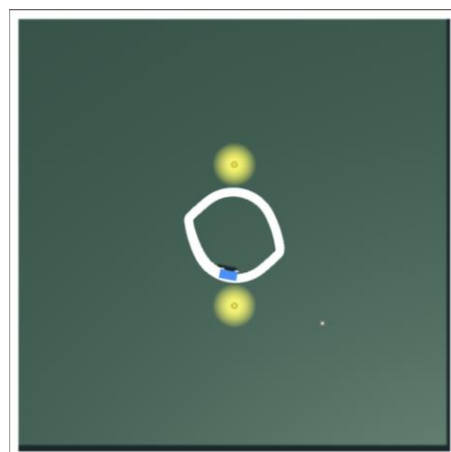


Fig. 18 – Elipse com limiares

Foram apenas aplicados limites uma vez que se constatou que, usando apenas estes, seria possível chegar ao efeito desejado. Ao aplicar os limiares 0.1 e 0.8, obtemos a trajetória apresentada na figura 18. Testamos com vários outros valores, mas sem sucesso, o que acabou por motivar o uso de apenas limites para a função de ativação.

O esperado seria uma trajetória de uma elipse mais centrada, estreita e constante no tempo. Por fim, apesar destes defeitos, o resultado obtido cumpre o conceito de uma elipse.

3.4.2. Cenários Adicionais

Pista de corrida

Função de ativação: Linear

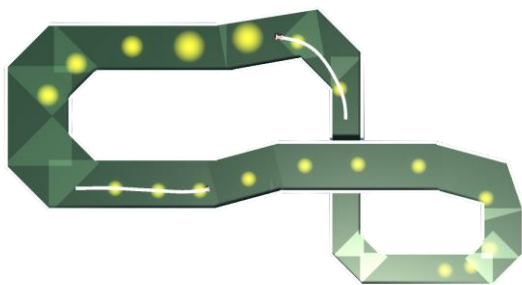
Velocidade máxima: 15

Sensores de Luz: Valores padrão

Com este cenário pretendemos testar a capacidade de o veículo ser agressivo perante outros elementos presentes no cenário, neste caso luzes. Nesta pista apresentamos 2 veículos que vão efetuar trajetos diferentes: um onde temos as luzes com *ranges* iguais e default (5) e outro onde temos *ranges* variados.

No primeiro percurso como explicado temos pequenas luzes de forma a que o veículo as consiga seguir até a um certo ponto. Foram feitos múltiplos ajustes no posicionamento das luzes já que uma pequena mudança na disposição das mesmas influenciaria o comportamento do carro. Isto acontece devido ao facto de que a energia calculada pelos sensores ser influenciada tanto pelo número de luzes como a distância das mesmas ao veículo. Portanto, uma luz mesmo a uma distância consideravelmente grande conseguiria afetar a forma como era descrito o trajeto, podendo por vezes resultar em curvas súbitas em locais não desejáveis ou mudanças de trajeto, completamente fora do esperado. Mas tirou-se partido disso para conseguir produzir a curva apertada. Ainda assim neste percurso não foram adicionadas mais luzes porque estas influenciariam como o carro se deslocaria na parte de cima da pista, fazendo com que este acabasse por embater nas paredes.

Já com o segundo percurso notamos que quanto maior o range das luzes mais fácil se tornava a gestão da direção para onde o carro iria, tendo uma maior diversidade de *ranges* para diferentes situações.



Com este cenário podemos então concluir que: com luzes mais pequenas fica mais fácil gerar curvas apertadas (*range* = 5), com luzes de tamanho médio é mais fácil gerar caminhos estáveis (*range* = 7) e finalmente com luzes de tamanho maior (*range* \geq 9) conseguimos ajustar facilmente a trajetória do nosso veículo. Estas conclusões ficam ainda mais evidentes depois de testado o trajeto 2.

Fig. 19 – Pista de corridas

Labirinto do Medo

Função de ativação: Linear

Velocidade máxima: 21

Sensores de Luz: Valores padrão

Para este cenário o objetivo consiste em percorrer o caminho que leva ao final do labirinto. O carro apresenta um comportamento de medo relativamente às luzes, desviando-se das mesmas até chegar ao lado oposto do labirinto. A velocidade escolhida é a velocidade máxima a que o carro consegue fazer o percurso com as luzes dispostas nas posições definidas na figura 20. Com valores acima de 21, o carro desvia-se do percurso, não completando o labirinto corretamente, como podemos observar na figura 21.



Fig. 20 – Labirinto final

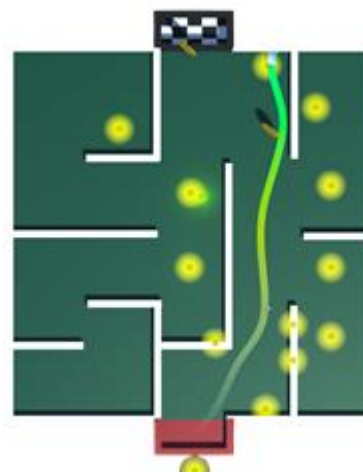


Fig. 21 – Labirinto com velocidade máx. 22

Parque

Guide:

Função de ativação: Linear

Velocidade máxima: 3

Sensores de Luz: Valores padrão

Police:

Função de ativação: Linear

Velocidade máxima: 20

Sensores de Proximidade (outros carros): sem restrições



Fig. 22 – Parque final

Este cenário pretende demonstrar dois tipos de comportamentos: agressividade e amor. Enquanto que o veículo *Guide* é agressivo com as luzes, seguindo o seu percurso, o *Police* demonstra um comportamento amoroso, seguindo atrás do *Guide*. Inicialmente, na construção deste cenário, a trajetória das luzes era mais oval, o que causou constrangimentos. A velocidade do *Guide* era muito baixa e inconsistente, o que por sua vez fez com que o *Police* tivesse dificuldades em adaptar-se à velocidade do

mesmo, causando choques entre ambos. Podemos assim concluir que, se a trajetória das luzes não for circular, torna-se mais difícil mostrar estes comportamentos, sendo necessário ajustes adicionais através da adição de limites e/ou limiares à função de ativação.

4. Conclusão

Ao longo do desenvolvimento deste trabalho, foram adquiridos conhecimentos acerca de como tirar partido dos diversos comportamentos representados pelos veículos de Braitenberg. Através do seu estudo, aprendemos a manipular sensores e funções matemáticas (funções de ativação gaussiana e linear), o que nos permitiu observar vários movimentos e escolher os valores mais convenientes para cada cenário. Observamos também que a disposição dos objetos (luzes, carros) influencia o comportamento dos veículos (como constatado no cenário “Pista de Corrida”).

Para além destas valências, ainda adquirimos conhecimentos sobre o Unity, ferramenta usada para criar os cenários. Assim, todos estes aspetos contribuíram para uma boa introdução à área da Inteligência Artificial.

5. Bibliografia

- Materiais disponibilizados pelo docente
- <https://www.geogebra.org/m/pddsmkw>
- <https://assetstore.unity.com/packages/3d/vehicles/land/low-poly-cars-101798>
- <https://assetstore.unity.com/packages/3d/props/food/simple-foods-207032>
- <https://assetstore.unity.com/packages/vfx/shaders/free-skybox-extended-shader-107400>
- <https://assetstore.unity.com/packages/3d/environments/landscapes/low-poly-forest-environment-package-173273>
- <https://assetstore.unity.com/packages/3d/props/polygon-starter-pack-low-poly-3d-art-by-synt-156819>
- <https://assetstore.unity.com/packages/3d/vegetation/nature-pack-low-poly-trees-bushes-210184>