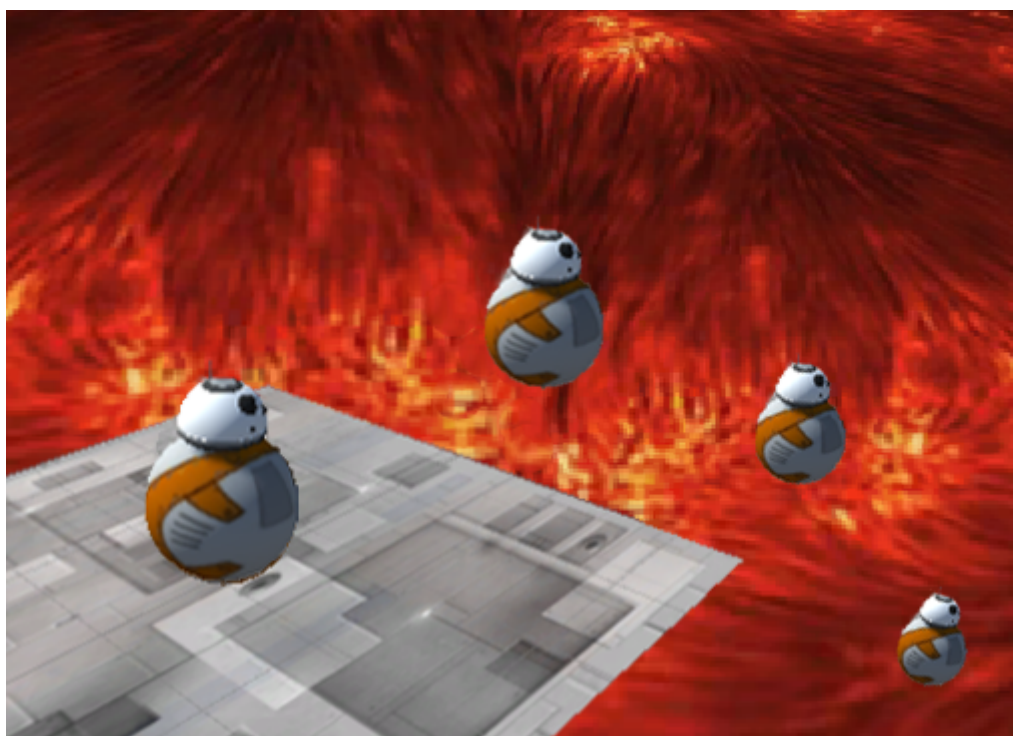


Trabalho Prático Nº1: Reactive D31: The AI Awakens

Fundamentos de Inteligência Artificial 2020/2021



Coimbra, 21 de março 2021

Maria Paula de Alencar Viegas
Sofia Margarida Ribeiro da Silva
Sofia Meireles Fonseca Costa

2017125592
2018293871
2018296218

viegas@student.dei.uc.pt	PL6
sofiasilva@student.dei.uc.pt	PL5
sofiacosta@student.dei.uc.pt	PL5

ÍNDICE

OBJETIVO	3
FUNCIONALIDADES IMPLEMENTADAS	3
DESCRIÇÃO DO COMPORTAMENTO ESPERADO PELO AGENTE	5
TESTES	6
ANÁLISE RESULTADOS	8
CONCLUSÃO	9
ANEXOS	9

OBJETIVO

O trabalho tinha como objetivo a implementação de funções de ativação que permitissem ao robot completar os mapas fornecidos sem bater em paredes, nem cair na lava ao apanhar os pickups. Para isso, foram utilizados dois sensores, um para detetar os recursos e outro as paredes.

FUNCIONALIDADES IMPLEMENTADAS

Para alcançar nossos objetivos, utilizamos o motor de jogos Unity com o package *tp1-alunos-2021.unitypackage* fornecido pelo professor. O script criado na primeira meta, *BlockDetectorScript.cs*, e o script *ResourceDetectorScript.cs* contém nossas funções de ativação e a verificação dos limiares:

- linear

```
public float GetLinearOutput(float limiarMinBlock, float limiarMaxBlock)
{
    //se a strength está fora do limiar, o output é 0
    if (strength >= limiarMaxBlock || strength <= limiarMinBlock)
    {
        return 0f;
    }
    return strength;
}
```

- gaussiana

```
public virtual float GetGaussianOutput(float limiarMinBlock, float limiarMaxBlock)
{
    float b = (float)0.5; //centro
    float c = (float)1;    //largura do sino
    float y;
    //se a strength está fora do limiar, o output é 0
    if (strength >= limiarMaxBlock || strength <= limiarMinBlock)
    {
        return 0f;
    }
    y = (float)Math.Exp((-0.5f) * Math.Pow((strength - b) / c, 2));
    return y;
}
```

- logarítmica negativa

```
public virtual float GetLogaritimicOutput(float limiarMinBlock, float limiarMaxBlock)
{
    float y;
    //se a strength está fora do limiar, o output é 0
    if (strength >= limiarMaxBlock || strength <= limiarMinBlock)
    {
        return 0f;
    }
    y = -1 * (float)Math.Log(strength);
    return y;
}
```

$$f(x) = a * \log(strength)$$

As variáveis *limiarMinBlock* e *limiarMaxBlock* são das funções de ativação do Block Detector e as variáveis *limiarMinResource* e *limiarMaxResource* são das funções de ativação Resource Detector. Elas são declaradas como public em *LinearRobotUnitBehaviour.cs*, ou seja, atribuímos valores na interface do Unity, e passadas como parâmetro às funções de ativação.

Em *LinearRobotUnitBehaviour.cs* guardamos a informação dos sensores em *resourceAngle/blockAngle* e, dependendo da função de ativação escolhida, em *resourceValue/blockValue*. Em seguida, verificamos se output de força *resourceValue/blockValue* está dentro dos limites *limitInfResource/limitInBlock* e *limitSupResource/limitSupBlock*. Caso contrário, igualamos o output ao limite. Finalmente, consideramos o peso *weightResource/weightBlock* no output e aplicamos a força ao robot.

```
void Update()
{
    //Resources
    resourceAngle = resourcesDetector.GetAngleToClosestResource();
    switch (funcResource)
    {
        case "gauss":
            // get sensor data
            resourceValue = resourcesDetector.GetGaussianOutput(limiarMinResource, limiarMaxResource);
            break;

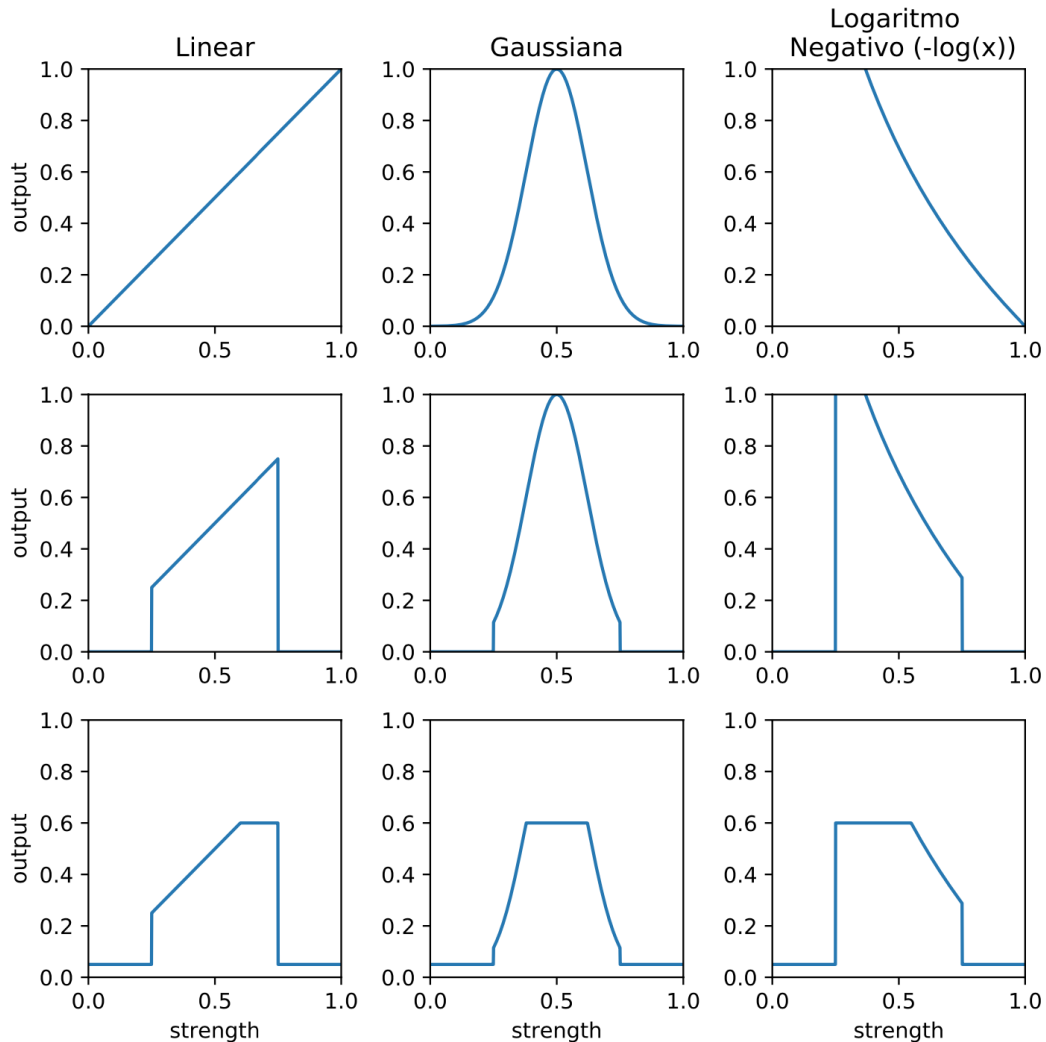
        case "log":
            // get sensor data
            resourceValue = resourcesDetector.GetLogaritmicOutput(limiarMinResource, limiarMaxResource);
            break;

        case "linear":
            // get sensor data
            resourceValue = resourcesDetector.GetLinearOutput(limiarMinResource, limiarMaxResource);
            break;
    }

    //quando a força está fora do limite, igualamos ao próprio limite
    if (resourceValue < limitInfResource)
    {
        resourceValue = limitInfResource;
    }
    else if (resourceValue > limitSupResource)
    {
        resourceValue = limitSupResource;
    }

    resourceValue = resourceValue * weightResource;
    // Apply to the ball
    applyForce(resourceAngle, resourceValue); // go towards
}
```

DESCRIÇÃO DO COMPORTAMENTO ESPERADO PELO AGENTE



Vemos pelos gráficos acima representados que diferentes funções de ativação cumprem diferentes expectativas. Enquanto uma função linear aumenta o output conforme aumenta o estímulo sensorial, uma função logarítmica produz um output maior no momento de detecção e gradualmente diminui conforme aumenta a força. Já a função de ativação gaussiana tem um arranque inicial até um ponto e depois gradativamente reduz o output.

Esperamos então que a função logarítmica seja adequada para a percepção de *resources*, que a função gaussiana seja apropriada para, por exemplo, o mapa 2a para que ele seja capaz de parar antes de cair na lava e que a função linear seja conveniente para detecção de *block*, já que queremos output de energia maior conforme ele aproxima-se da parede.

TESTES

Realizámos cerca de 140 testes com o intuito de descobrir a melhor combinação para passar cada mapa. Em seguida, seguem os melhores testes para cada função/mapa.

Mapa	Função Resource	Função Block	Limiar Min Resource	Limiar Max Resource	Limiar Min Block	Limiar Max Block	Limite Inf Resource	Limite Sup Resource	Limite Inf Block	Limite Sup Block	Time elapsed
1a	gauss	linear	0.1	0.5	0.2	0.8	0.2	0.6	0.15	0.7	0:14
1a	gauss	linear	0	0.5	0.2	0.8	0.05	0.4	0.005	0.7	0:14
1a	gauss	gauss	0	0.5	0.2	0.8	0.05	0.4	0.05	0.7	0:11
1a	gauss	gauss	0	0.6	0.2	0.6	0.05	0.4	0.005	0.7	0:11
1a	gauss	log	0.1	0.8	0.05	0.5	0	0.3	0.05	0.3	0:21
1a	gauss	log	0.1	0.8	0.05	0.5	0	0.3	0.05	0.3	0:18
1a	linear	linear	0.05	0.45	0.2	0.4	0	0.2	0.1	0.4	0:06
1a	linear	linear	0.2	0.4	0	0.2	0	0.2	0	0.7	0:06
1a	linear	gauss	0	0.6	0.05	0.1	0	0.1	0.1	1	0:10
1a	linear	gauss	0	0.7	0	0.1	0	0.1	0.05	1	0:10
1a	linear	log	0	1	0	0.1	0	0.2	0.2	1	0:09
1a	linear	log	0	1	0	0.05	0	0.1	0.05	1	0:10
1a	log	linear	0	0.3	0	0.5	0.1	0.3	0	0	0:33
1a	log	linear	0	0.3	0	0.65	0.1	0.5	0.2	0.4	0:14
1a	log	log	0	0.7	0	1	0.15	0.4	0.2	0.4	0:2
1a	log	log	0	0.7	0	1	0.15	0.4	0.2	0.4	0:21
1a	log	gauss	0.2	0.6	0.2	0.7	0.15	0.45	0.25	0.4	0:33
1a	log	gauss	0.2	1	0	1	0.3	0.5	0	0.3	0:23
1b	gauss	linear	0	0.9	0.3	0.6	0	0.1	0.05	0.2	0:25
1b	gauss	linear	0	0.7	0.3	0.6	0	0.1	0.1	0.2	0:25
1b	gauss	gauss	0	0.7	0.3	0.6	0	0.1	0.1	0.2	0:24
1b	gauss	gauss	0	0.8	0.3	0.6	0	0.1	0.05	0.2	0:25
1b	gauss	log	0	0.5	0.3	0.7	0	0.1	0.05	0.2	0:25
1b	gauss	log	0	0.5	0.3	0.6	0	0.2	0.05	0.3	0:25
1b	linear	linear	0	0.9	0.05	0.35	0	0.1	0	0.8	0:18
1b	linear	linear	0	1	0	0.5	0	0.1	0	0.5	0:20

1b	linear	gauss	0	1	0	1	0	0.3	0.1	0.5	0:23
1b	linear	gauss	0.1	1	0	1	0.05	0.1	0.1	0.15	0:11
1b	linear	log	0.1	0.6	0	1	0.05	0.1	0.1	0.25	0:14
1b	linear	log	0	1	0	0.1	0	0.1	0.1	0.25	0:14
1b	log	log	0	1	0	1	0	0.5	0.1	0.5	0:11
1b	log	log	0	0.8	0	0.7	0	0.5	0.1	0.5	0:12
1b	log	gauss	0	1	0	1	0	0.5	0.1	0.4	0:12
1b	log	gauss	0	1	0	1	0	0.2	0.1	0.4	0:16
1b	log	linear	0	1	0	1	0	0.2	0.1	0.4	0:16
1b	log	linear	0	1	0	1	0	0.5	0.1	0.5	0:14
2a	gauss	---	0	0.2	0	0	0	0.07	0	0	0:08
2a	gauss	---	0	0.3	0	0	0	0.04	0	0	0:09
2a	linear	---	0	0.17	0	0	0	0.13	0	0	0:06
2a	linear	---	0	0.17	0	0	0	0.1	0	0	0:07
2a	log	---	0	0.3	0	0	0	0.1	0	0	0:05
2a	log	---	0	0.3	0	0	0	0.035	0	0	0:09
2b	gauss	linear	0	0.8	0.2	0.6	0.3	0.7	0.05	0.6	0:27
2b	gauss	linear	0	0.6	0.2	0.6	0.3	0.7	0.01	0.5	0:28
2b	gauss	gauss	0	0.7	0.05	0.2	0.2	0.7	0.05	0.5	0:26
2b	gauss	gauss	0	0.4	0.05	0.6	0.2	0.7	0.1	0.6	0:32
2b	gauss	log	0	0.7	0.2	0.5	0	0.4	0.05	0.2	0:37
2b	gauss	log	0	0.6	0.05	0.6	0	0.7	0.1	0.6	0:42
2b	linear	linear	0	0.3	0.3	0.65	0.1	1	0.15	0.2	1:16
2b	linear	linear	0.1	0.3	0.3	0.65	0.1	1	0.15	0.25	1:16
2b	linear	gauss	0.2	0.95	0.4	0.8	0	0.8	0	0.8	0:48
2b	linear	log	0.2	0.8	0.4	0.8	0.2	0.8	0.15	0.85	0:48
2b	log	log	0	1	0.3	0.65	0	0.25	0.15	0.25	1.3
2b	log	log	0	1	0.3	0.65	0	0.25	0.1	0.25	0:44
2b	log	linear	0	1	0.3	0.65	0	0.25	0.15	0.35	0.44
2b	log	gauss	0	1	0.3	0.65	0	0.25	0.15	0.3	0:41
2b	log	gauss	0	1	0.3	0.65	0	0.3	0.15	0.35	0:38

ANÁLISE RESULTADOS

- Mapa 1a:

Para este mapa, o melhor tempo que obtivemos foi usando a função de ativação linear tanto para os blocos como para os resources. Vimos que todas as combinações de funções são similares até antes do último pickup. No final, ele tende a bater na parede ou ter dificuldades de conciliar a função de ativação do resource e do bloco, o que causa que o robot dê voltas em torno deste pickup. Ter uma função mais brusca como a linear ajudou a ter mais precisão nesta parte final.

- Mapa 1b

Neste mapa, a combinação que melhor resultou foi a utilização da função de ativação linear para os resources e da função de ativação gaussiana para os blocos. Isto talvez porque o mapa apresenta muitas paredes e combinando gaussiana e linear temos uma força intensificada para os resources e uma cautela com as paredes.

- Mapa 2a

No mapa 2a, o melhor resultado foi obtido usando a função de ativação logarítmica. Isto está de acordo com as nossas expectativas iniciais, visto que o pico de energia é quando o resource é detetado e desacelerando logo de seguida. Assim, evitamos que o D31 caia na lava.

- Mapa 2b

Em relação a este mapa, a combinação que apresentou melhores resultados foi a gaussiana tanto para os blocos como para os resources. Isso pode ser explicado uma vez que é um mapa complexo e a função gaussiana produz movimentos mais cautelosos com um boost inicial seguido de uma gradual perda de força.

CONCLUSÃO

Tendo em conta as diferenças entre computadores, os resultados obtidos nos mapas podem variar com os casos de teste aqui propostos.

Vimos dificuldades ao fazer os testes dado que as diferentes máquinas podiam produzir diferentes resultados, além de repararmos discrepância nos resultados dependendo se estávamos a realizar testes em conjunto via Discord ou separadas ou mesmo se o Unity já estava a correr por muito tempo seguido.

As expectativas nem sempre foram atendidas pois achamos difícil ter o total controle sobre as forças, principalmente quando o *pickup* se encontra próximo da *wall* e as funções se sobrepõem.

Através dos resultados obtidos nos diversos testes realizados, concluímos que tanto as funções de ativação como os limites e os limiares têm grande influência na velocidade de execução. As soluções encontradas são diferentes entre os mapas, uma vez que estes apresentam características muito distintas.

ANEXOS

Em anexo, segue a Unity package do nosso projeto de nome *projeto1_FIA_meta2_final.unitypackage*