FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

Braitenberg Vehicles UC

Trabalho Prático 1 – Meta 1 da disciplina de Fundamentos da Inteligência Artificial 2021/2022

Miguel Pedroso	2019218176 PL8
João Dionísio	2019217030 PL8
Simão Craveiro	2019210557 PL2

Introdução

Nesta meta 1 do trabalho da disciplina de introdução à inteligência artificial era-nos pedido que nos ambientássemos com o ficheiro base disponibilizado pelo professor e que desenvolvêssemos os sensores que permitem a detecção de outros carros e que os aplicássemos num deles para seguir os restantes.

Desenvolvimento

```
void Update()
{
    minDistance = 0;
    GameObject[] cars = null;       //Arraylist de carros
    if (useAngle)
    {
        cars = GetVisibleCars();    //Só devolve os carros no ângulo de visão
    }
    else
    {
        cars = GetAllCars();        //Devolve todos os carros
    }
    GameObject closestCar = null;

    //Encontra o carro mais próximo e calcula a distância a que está
    foreach (GameObject car in cars)
    {
        //print (1 / (transform.position - light.transform.position).sqrMagnitude);
        float carDistance = (transform.position - car.transform.position).sqrMagnitude;
        print("carDistance:" + carDistance);
        if (carDistance < minDistance || minDistance == 0)
        {
            minDistance = carDistance;
            closestCar = car;
        }
        //Debug.DrawLine (transform.position, light.transform.position, Color.red);
    }
    // A velocidade é proporcional à distância do carro mais próximo
    output = (float)0.010*minDistance;
    if(output > 20)
    {
        output = 20;
    }
    //output = 1 / (minDistance + 1);

}
```

Figura 1 - Código desenvolvido para os sensores que detetam o carro mais próximo.

```
public virtual float GetOutput() { throw new NotImplementedException(); }

// Returns all "CarToFollow" tagged objects. The sensor angle is not taken into account.
GameObject[] GetAllCars()
{
    return GameObject.FindGameObjectsWithTag("CarToFollow");
}

// Returns all "CarToFollow" tagged objects that are within the view angle of the Sensor.
// Only considers the angle over the y axis. Does not consider objects blocking the view.
GameObject[] GetVisibleCars()
{
    ArrayList visibleCars = new ArrayList();
    float halfAngle = angle / 2.0f;
    //Só considera os objetos com a Tag "CarToFollow"
    GameObject[] cars = GameObject.FindGameObjectsWithTag("CarToFollow");

    foreach (GameObject car in cars)
    {
        Vector3 toVector = (car.transform.position - transform.position);
        Vector3 forward = transform.forward;
        toVector.y = 0;
        forward.y = 0;
        float angleToTarget = Vector3.Angle(forward, toVector);

        if (angleToTarget <= halfAngle)
        {
            visibleCars.Add(car);
        }
    }

    return (GameObject[])visibleCars.ToArray(typeof(GameObject));
}
```

Figura 2 - Código que devolve todos os carros na cena ou todos os carros visíveis na cena.

Contactos

João Dionísio - uc2019217030@student.uc.pt
Miguel Pedroso - uc2019218176@student.uc.pt
Simão Craveiro - uc2019210557@studen.uc.pt