**Introdução à Inteligência Artificial**

Ano letivo 2019/2020

**Relatório**

**Trabalho Prático Nº1**

**Reactive D31: The AI Awakens**

**Meta 1**



**Trabalho Realizado por:**

Alexandre Maria Martins Magalhães Teixeira Serra – 2017248031 – [amserra@student.dei.uc.pt](mailto:amserra@student.dei.uc.pt) – PL6

João Gabriel de Matos Fernandes – 2017247486 – [joaof@student.dei.uc.pt](mailto:joaof@student.dei.uc.pt) – PL2

João Pedro de Sá Dinis – 2017248548 – [joaodinis@student.dei.uc.pt](mailto:joaodinis@student.dei.uc.pt) – PL6

Departamento de Engenharia Informática

Faculdade de Ciências e Tecnologia da Universidade de Coimbra

O nosso foco para esta meta foi implementar um sensor no robô que deteta obstáculos. Para tal, usamos o sensor de recursos como referência.

**Informações Relevantes:**

Sempre que o robô deteta um obstáculo roda num certo ângulo (que pode diferir entre cada mapa) e afasta-se desse. Depois do robô apanhar todos os recursos, colocamos a sua energia (strength) a 0. Logo, passado uns segundos o robô para por completo.

    void FixedUpdate() {

        ObjectInfo obstacle;

        obstacle = GetClosestObstacle();

        //Caso ainda nao tenha apanhado os recursos todos, continua a afastar se de obstaculos

        if(agent.resourcesGathered < agent.maxObjects) {

            //Caso encontre um obstaculo roda afasta se num angulo dado pela variavel angleOffSet

            if (obstacle != null) {

                angleToClosestObstacle = obstacle.angle + angleOffset;

                // Formula no enunciado

                strength = 1.0f / (obstacle.distance + 1.0f);

            }

        } else { //Caso apanhou os recursos todos, o robo fica parado

            strength = 0;

            angleToClosestObstacle = 0;

        }

    }

Na função GetLinearOutput (presente no ficheiro BlockDetectoScipt.cs), multiplicamos a variável strength, por um fator de força, que permite controlar a velocidade com que o robô se afasta dos obstáculos.

    public float GetLinearOuput() {

        return strength \* strengthFactor; // Variar a forca

    }

**Objetivos Alcançados:**

* O robô consegue completar os mapas a1 e a2 com relativa facilidade
* Criámos dois novos mapas para testar mais exaustivamente o comportamento do robô para diferentes configurações e mapas
* O agente consegue evitar a maior parte dos obstáculos

**Dificuldades:**

* Por vezes o robô tende a escolher evitar obstáculos ao invés de apanhar os recursos
* Quando o robô não consegue detetar nenhum recurso dentro do seu campo de visão (por exemplo, uma parede encontra-se entre o robô e o recurso) o robô fica parado ou a bater entre as paredes
* Caso o último recurso esteja perto de um obstáculo, o robô bate no obstáculo depois de apanhar o recurso