FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA

Departamento de Engenharia Informática

Introdução à Inteligência Artificial
2019/2020 - 2º Semestre

Practical Work Nº1:
Reactive D31: The AI Awakens

Note: Fraud denotes a serious lack of ethics and constitutes unacceptable behaviour in a student of higher education and future licensed professional. Any attempt at fraud will lead to the annulment of the practical component of both the facilitator and the offender, regardless of any additional disciplinary action that may take place under the terms of the legislation in force. If non-original material is used, the sources must be explicitly indicated.

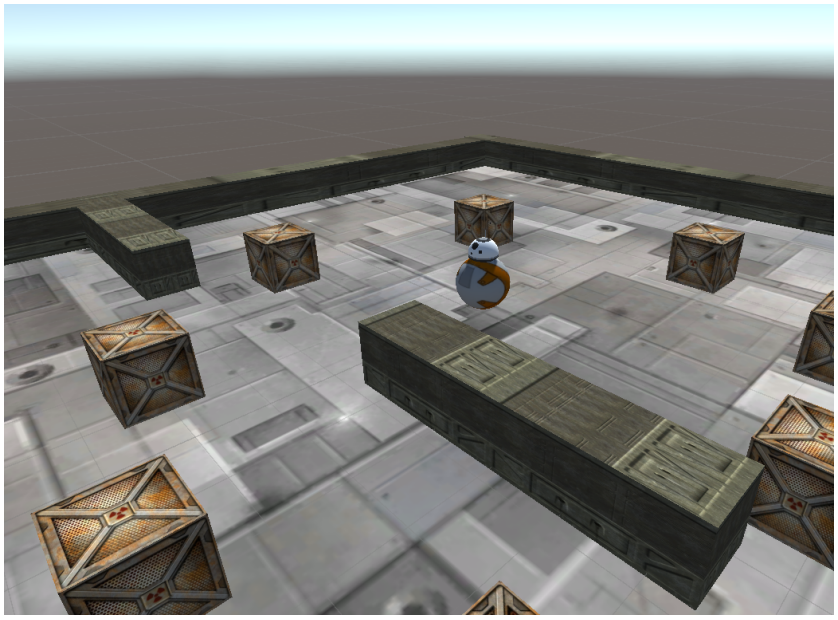

Figura 1: Example of an ambient with D31.

1 Introduction

Reactive agents are agents that perceive the environment and perform actions through these same perceptions through a system of simple rules. Although they are based on simple processes, it does not imply that with this type of agents it is not possible to observe complex behaviours. An emblematic example of this type of agents is described by the neuro-anatomist Valentino Braitenberg in his book "Vehicles - Experiments in Synthetic Psychology".

This practical work intends to implement self-reactive reactive agents. In this case, the robot D31 is in an environment populated by obstacles and resources and needs help to navigate it.

The implementation will be done through Unity. Unity is a game engine developed by Unity Technologies, especially notable for its cross-platform portability. Among its specifications is to distinguish the physical and graphical engines, the extensive library and respective documentation, as well as support for C # and JavaScript languages. Although its purpose is to support game development, these features also allow Unity to be used as a simulation tool, with applicability in Intelligence and Artificial Life, capable of functioning as an environment for realistic simulations. In this case, it will be used for implementation and real-time testing of various reactive agents.

2 Generic Objectives

The present practical work has as generic objectives:

1. The acquisition of application development skills in Unity as a simulation environment;

2. The acquisition of skills related to the analysis, development, implementation and testing of autonomous reactive agents.

These generic objectives will be achieved through group work and experimentation, thus promoting these capabilities.

3 Assignment

As the name of the practical work suggests, it is intended to be personalized, both at the functional and aesthetic level of the agent provided. This implies expanding the prefabs distributed with the project, in order to add new sensors, obstacles, etc ...

This practical work is divided into 2 distinct goals:

1. Goal 1 - Sense it

2. Final Goal - Tune it Test it

3.1 Goal 1 – Sense it

The prefab "d31-r"provided only provides a type of sensor for resources, which reacts to their existence in the agent's field of view.

The first task to perform is to expand the code provided to allow different types of object sensors. The new agent must support sensors:

1. Resources[1]

2. Obstacles

These sensors must respond, respectively, to:

1. Resources

2. Blocks

[1]Este sensor já está implementado no prefab "d31-r" fornecido.

The output of the sensors is calculated as a function of the object that is closest to its field of view, conditioned by the direction between agent D31 and the object.

You should, therefore, test all the functionalities by constructing the variant of agent D31 that take advantage of these sensors. Test the agents in the environments provided mapa1a and mapa1b and others that you will create that illustrate the different functionalities.

3.2 Final Goal - Tune it & Test it

3.2.1 Tune it

In the original version of the agent "d31-r"the function of activating the sensors is, by default, linear.

The calculation of energy based on the resource sensor is done as follows:

$$energia = \frac{1}{distance(obj, sensor) + 1} \tag{1}$$

where obj is the closest point to the sensed object. Using the linear activation function (in the code provided, the function GetLinearOutput()), the sensor output is calculated using eq 1 without any change or filter.

However, the function of linear activation imposes serious restrictions and makes it impossible to program various interesting behaviours. You must make the necessary changes so that it is possible to specify, for each sensor:

1. The desired linear, Gaussian or logarithmic activation function;
2. Minimum and maximum active threshold;
3. Upper and lower limit;

Figure 2 shows examples of these functions.

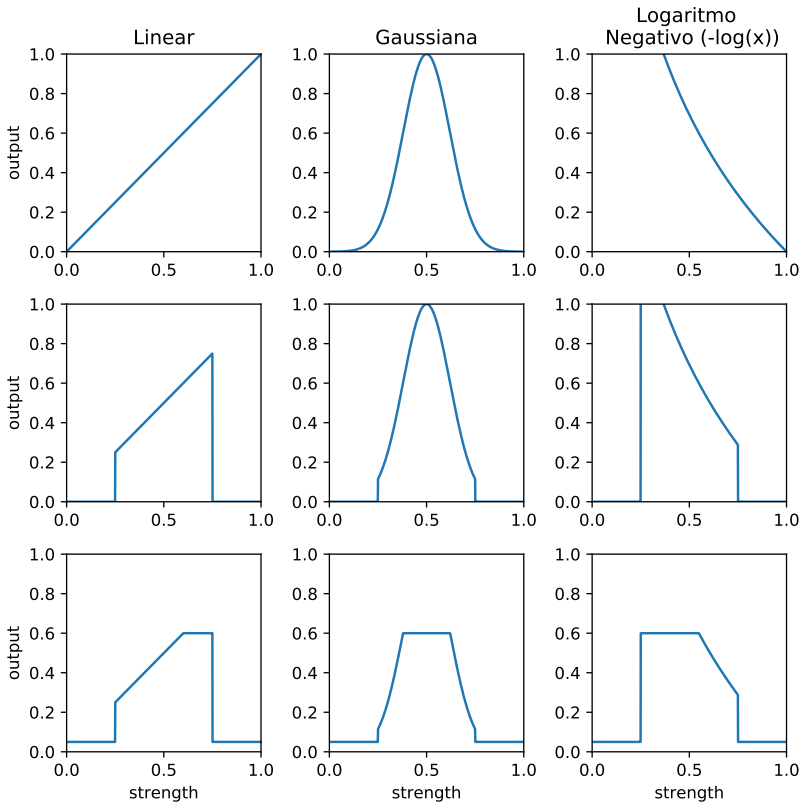

Figura 2: Examples of activation functions: linear, gaussian, negative logaritm with different parameters. Gaussian with a $\mu = 0.5$ and $\sigma = 0.12$. Second line with thresholds of 0.25 and 0.75 (x values). Third line we use the same thresholds and add limits of 0.05 and 0.6 (y values).

In addition to the energy calculation, there are configuration parameters for each sensor that must consider: (i) sensor range (range of sensor); (ii) increase in the angle of the sensor rays (angle of sensor), eg since the sensors act around the agent (360 degrees), setting 10 in this parameter (default value in the sensor provided) implies that the 360 degrees around of the sensor are divided into rays every 10 degrees (36 rays equally spaced around the sensor).

3.2.2 Test it

Taking advantage of the different types of parameters, activation functions, thresholds and limits, create agents that can solve the maps provided:

- The objective will be to collect resources without the agent falling off

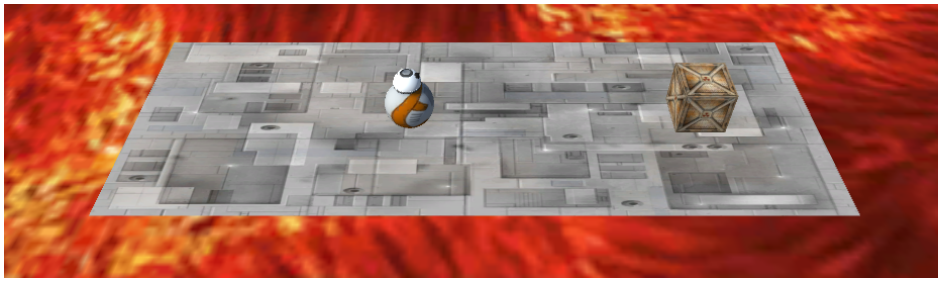

Figura 3: Collect without falling!

Figura 4: Obstacles and resources map.

the platform (Figure 3).

- map2b -The objective will be to collect all resources by dodging the blocks and without falling off the platform (Figure 4). On this map, you can adjust the agent's initial position within the zone with a darker texture.

hrough the use of activation functions, thresholds and thresholds, create an agent that explores the environment without colliding with the existing blocks. There is no restriction on the number of sensors or activation functions, however, you should try to find a good compromise between performance and resource savings.

In addition to solving the proposed environments, try to build environ-

ments that illustrate the properties of the different agents and that test their capabilities. Analyze the agents' behaviour, indicating their strengths and weaknesses. You can also use the information provided, time and resources collected, to compare your agents.

4 Dates and Project Delivery

4.1 Meta 1 – Sense it

Material to be delivered:

- The code developed so far, duly commented;
- A brief document (max. 3 pages), in pdf format, with the following information:
  - Identification of the elements of the group (Names, Student Numbers, e-mails, Practice Group (s))
  - Information pertinent to this goal, achieved objectives and difficulties.

Delivery:

Electronic delivery through Inforestudante. Deadline: **1st March 2020**

4.2 Final Goal – Tune it & Test it

Material to be delivered:

- The code developed, duly commented, for each of the goals;
- A report (max. 10 pages), in pdf format, with the following information:
  - Identification of the elements of the group (Names, Student Numbers, e-mails, Practice Group (s))
  - Relevant information regarding the overall work carried out.

In a work of this nature, the story takes on an important role. Care must be taken to describe in detail all the implemented functionalities, with particular emphasis on the problems and solutions found. It should be easy for the reader to understand what was done and therefore have the ability

to adapt/modify the code. For each agent developed, it should describe the expected behaviour and the way in which that behaviour was achieved.

Experimentation is an essential part of the development of AI applications. Thus, you should describe in detail the experiments carried out, analyze the results, draw conclusions and make changes (if justified) depending on the experimental results, in order to optimize the performance of your agents.

The report must contain relevant information from both the user's and the programmer's perspective. It should not exceed 10 pages, A4 format. All options taken must be duly justified and explained.

Delivery:

Electronic delivery through Inforestudante. Deadline **15th March 2020**

5 Bibliography

- **Inteligência Artificial: Fundamentos e Aplicações**
  Ernesto Costa, Anabela Simões
- **Vehicles: Experiments in Synthetic Psychology**
  Valentino Braitenberg