ID2209 – Distributed Artificial Intelligence and Intelligent Agents

Lab Assignment 1 Report

Group 28

Enrique Perez Soler — Daniel Conde Ortiz

December 15, 2019

1 Introduction

As new as we were with the GAMA simulation platform, the introductory *HelloWorld* project showed in the introduction to the labs provided enough information to understand how it worked and how to obtain the results from the simulations. It also allowed us to further understand the potential of the tool by providing a working example, with it we were able to follow certain good practices and fulfill this very first assignment.

In this assignment we simulate a festival in which people are dancing and sometimes they become hungry or thirsty and go to an information point to ask where bars and food trucks are. Then, they go there and satisfy their needs.

2 How to run

Run GAMA 1.8 and import Lab1.gaml as a new project. Press *Lab1* to run the simulation. At the beginning of the file you can change how many agents of each species are there. The default values are: 10 guests, 2 bars, 2 food trucks and 1 information point.

3 Species

3.1 FestivalGuest

It dances (wanders) until, randomly, becomes hungry or thirsty. It knows the location of the information point so it goes there and asks for a food truck or a bar, depending on its needs. When it gets the location of the store, it goes there and replenishes its needs, going back to a random spot on the festival where it continues dancing.

There are three main variables, two of type boolean that indicate if the guest is *hungry* or *thirsty* and one last one of type "point", named *targetPoint*, that indicates a location where it should go next.

There are several reflexes such as *enterInfo* and *enterStore*, that are activated once the agent gets close to the shops or information point and asks them. Moreover, there's also the *beIdle* reflex where it wanders around the festival until a need appears and the reflex *moveToTarget* comes into action. This one is in charge of moving the agent.

3.2 InfoPoint

It's represented as a yellow box or cube located in the center of the map and it's in charge of indicating the location of the bars and food trucks to the guests. For simplicity in our code, we implemented this function in the *FestivalGuest* species.

3.3 Bar

As mentioned before, for simplicity the *Bar* species has no major role to play as its functionalities have been added in the *FestivalGuest* species as a reflex. Theoretically, it was in charge of replenishing the *thirst* of guests. In our version of the code it does nothing apart from being assigned a random location each time a simulation is ran. They are represented by blue boxes.

3.4 FoodTruck

For simplicity the *FoodTruck* species has no major role to play as its functionalities have been added in the *FestivalGuest* species as a reflex. Theoretically, it was in charge of replenishing the *hunger* of guests. In our version of the code it does the same as the *Bar* species. They are represented by green boxes.

4 Implementation

We started with a simple scenario with only one guest and making it *wander*, to get used to the language. The introduction *HelloWorld* example helped us a lot at the time of figuring out how the species related to the reflexes. Once we had it figured out, we proceeded to add all the necessary species for the assignment. After that, we started to make the festival guests go to the information point whenever they felt hungry and we did so randomly using a *reflex*.

The next step was figuring out how to communicate between agents, specifically the communication between the *Information Point* and the *Bars/FoodTrucks* to get/select the location of one of them depending on which need had to be replenished. We did some research in the GAMA platform web page and found a way of choosing by using *ask* and *one_of*. Finally, we made them go to a random location and start dancing again, with a final *reflex*.

5 Results and Conclusion

When implementing the lab we felt comfortable with the new programming language. It was somewhat intuitive and simple although at first it was hard to apply previous notions of programming and problem solving algorithms in the GAMA platform. When setting up the simulation (figure 1) we won't always have the same distribution of the stores and guests as these are assigned random locations each time a simulation is run. When running the simulation we can appreciate the guests *wandering* and turning *cyan* when they are *thirsty* or *green* when they are feeling *hungry*. In either case they would proceed to the *Info Point* to ask for directions to a *Bar* or *Food Truck* respectively (figure 2)

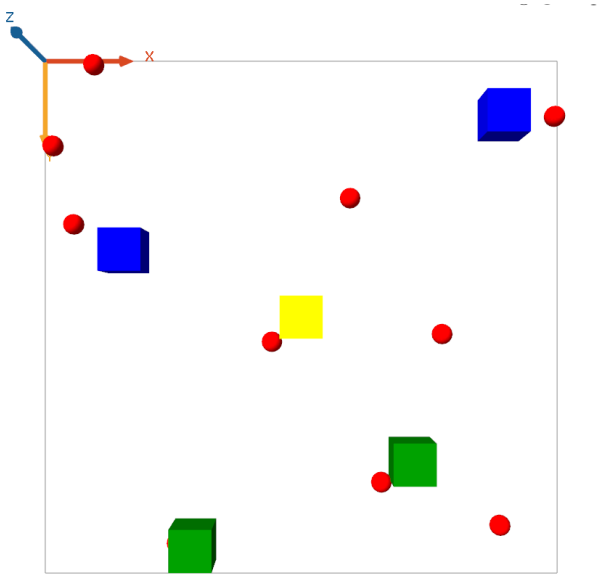

Figure 1: Set up before running a simulation

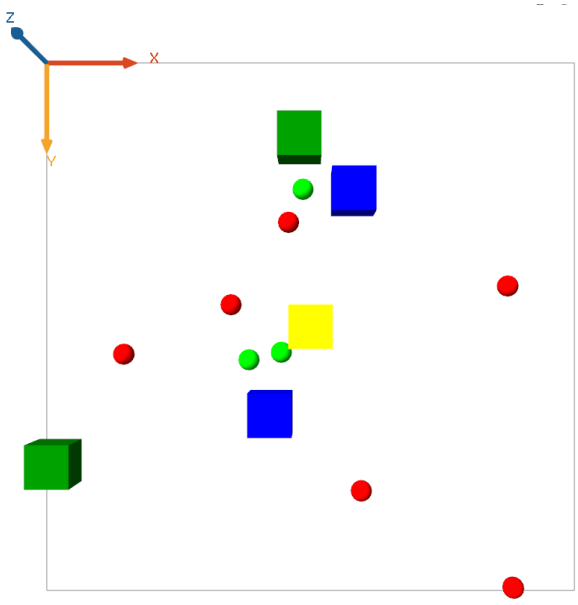

Figure 2: State of *FestivalGuests* in a running simulation (It's from a different simulation)