DAIIA Assignment Report

Course: [Distributed Artificial Intelligence and AI Agents]
Assignment: [Homework 1]

[Lorenzo Deflorian, Riccardo Fragale, Jouzas Skarbalius]
KTH Royal Institute of Technology

November 13, 2025

Contents

1. Running Instructions

Import the ZIP file given on GAMA and go to the folder assignment1/models. You will find a file called Festival.gaml where our model for the homework is implemented. Then, from the interface of GAMA, click on the play button and you will see a simulation. Use the tools provided by the GAMA simulation interface to adjust the speed, read the outputs and verify on screen that everything is working correctly. Note that changing the parameters *numCenter,numShops and numGuests* will change the number of agents that will apply on the screen and be part of the simulation.

2. General Overview

2.1 Agents

We have defined 4 main species:

- InformationCenter
- Shop
- SecurityGuard
- Guest

We also have a subspecies of Guest that is called SmartGuest and has been implemented to obtain the objectives requested by challenge 1.

The agent of type **InformationCenter** is responsible to give informations to all the agents that are asking him where the shops are. He knows the locations of them and is also connected to a SecurityGuard that will be helpful for the purpose of the second challenge. regarding the aspect of this agent, it is depicted as a black square of length 5 and its locations is right in the center of the field where the simulation is happening. All the other agents know by default the position of the InformationCenter.

A **Shop** is an agent responsible of replenishing food or water to all the guests that are coming to them. They are divided into food shops and water shops and their position changes in each simulation. Water shops are depicted as grey triangles while food shops are red triangles.

The **Guest**, instead, takes part in the festival and basically wonders randomically unless it is either hungry or thirsty. In this case it goes to the InformationCenter and asks it where is it possible to find food or water to fulfill its need. Then he starts moving to the location given by the InformationCenter and "charges its batteries". Then, it continues moving aimlessly unless it feels hungry or thirsty again(or both). A guest is a pink dot.

As we said before, there is also the **SmartGuest** that has a memory and so it is able to remember the locations of the shops visited. In this way he could move less with respect to normal guests and it will satisfy its need faster. Said that, randomically it can go to

the InformationCenter because he wants to discover new shops. The SmartGuest appears on screen as a yellow dot.

Last but not the least, there is the **SecurityGuard** that is repsonsible to catch and kill all the guests that doesn't follow the rules of the festival and must be eliminated. The security guard is a blue dot and its starting position is northwest with respect to the InformationCenter.

2.2 Assumptions

Our model has 1 InformationCenter, 1 SecurityGuard, 4 Shops (two of which of type *food* and two of type *water*). Regarding the Guests, we have defined three Smart guests and seven guests without memory.

2.3 Goals

The goals of this homework are:

- Introduction to the Gama platform
- Working with agents
- Learning the GAMA syntax
- Creating different types of agents
- Starting basic simulations
- Little bit of movement and behaviour

3. Basic implementation

3.1 Explanation. The basic implementation required us to create a simulation of a festival where Guests get hungry or thirsty. If they do, they should go to an Information Center to ask for the nearest Store that gives them what they need. Afterward, the Guests should simply keep doing something until they get hungry/thirsty again (they wander randomically).

3.2 Code. Include relevant code snippets below:

```
# Example code snippet
def example():
    print("Hello, DAIIA!")
```

Optionally include screenshots with:

3.3 Demonstration. Provide two use cases for this section.

1. **Input:** Describe the input.
2. **Screenshot:** Include program execution/output.
3. **Interpretation:** Briefly explain the result.

4. Challenges

Challenge 1: Smart Guests

4.1 Explanation. For this challenge we implemented a new type of Guest, called SmartGuest, that has a memory of the shops he has visited. This way if he gets hungry or thirsty again, he can decide to either go to the InformationCenter to ask for a new shop or go directly to a known shop.

```
species SmartGuest parent: Guest{
    list <Shop> visitedPlaces;

    reflex manage_needs{
        if ((hunger >= hungerThreshold or thirsty >= thirstThreshold) and onTheWayToShop = false and targetShop = nil) {
            // No visited places go to info center
            if (length(visitedPlaces)) <= 0 {
                do go_infocenter;
                return;
            }

            bool visitNewPlace <- rnd(0, 1);
            if (!visitNewPlace) {
                do go_infocenter;
                return;
            }

            list shuffledPlaces <- shuffle(visitedPlaces);
            loop i from: 0 to: length(shuffledPlaces) - 1 {
                Shop candidateShop <- shuffledPlaces[i];
                if (hunger >= hungerThreshold) {
                    if (candidateShop.trait = "food") {
                        targetShop <- candidateShop;
                        onTheWayToShop <- true;
                        return;
                    }

                }

                if (thirsty >= thirstThreshold) {
                    if (candidateShop.trait = "water") {
                        targetShop <- candidateShop;
                        onTheWayToShop <- true;
                        return;
                    }

                }
            }
        }
    }
```

Figure 1: Code snippet for SmartGuest behavior

4.2 Code. The code snippet in Figure 1 shows how the SmartGuest decides whether to go to a known shop or ask the InformationCenter for a new one. With a probability of 50%, the SmartGuest chooses to visit a previously known shop from its memory of visited places. It randomly selects one shop that satisfies its current need (food or water) and heads there directly; otherwise it behaves like a normal Guest and goes to the InformationCenter.

Whenever a SmartGuest visits a shop, it adds that shop to its list of visited places so it can remember it for future needs, reducing travel time on repeated needs.

4.3 Demonstration. Provided use cases for this challenge:

1. A smart guest with no memory goes to the info center.
2. A smart guest with memory still goes to the infor center to discover a new shop.
3. A smart guest decides to go to a known food shop from its memory.
4. A smart guest decides to go to a known water shop from its memory.

4.3.1 Smart Guest with no memory

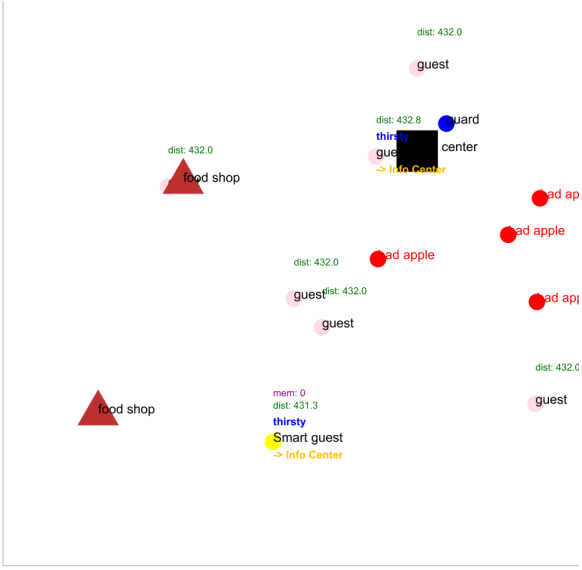

Figure 2: Smart Guest with no memory going to Information Center

In Figure **??**, we see a Smart Guest (yellow dot) that has no memory of shops. When it gets hungry, it goes to the Information Center (black square) to ask for the nearest food shop.

4.3.2 Smart Guest discovering new shop

In Figure **??**, we see a Smart Guest (yellow dot) that already has one shop in its memory. However, it decides to go to the Information Center (black square) to discover a new shop instead of going to the known one.

4.3.3 Smart Guest going to known food shop

In Figure **??**, we see a Smart Guest (yellow dot) that has a food shop in its memory. When it gets hungry, it decides to go directly to the known food shop (red triangle) instead of asking the Information Center.

Figure 3: Smart Guest discovering a new shop

4.3.4 Smart Guest going to known water shop

In Figure **??**, we see a Smart Guest (yellow dot) that has a water shop in its memory. When it gets thirsty, it decides to go directly to the known water shop (grey triangle) instead of asking the Information Center.

5. Final Remarks

Overall this assignment was great. We had the opportunity to learn the basics of the syntax of Gamma and we were challenged to implemented interesting things. We realized that changing a bit the inputs or minor details might lead to big changes in each simulation.

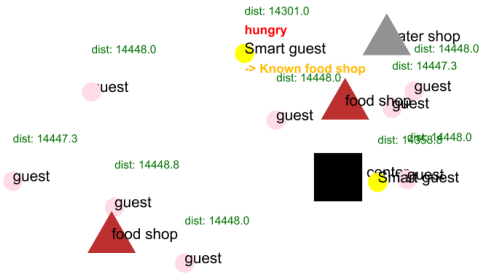

Figure 4: Smart Guest going to known food shop

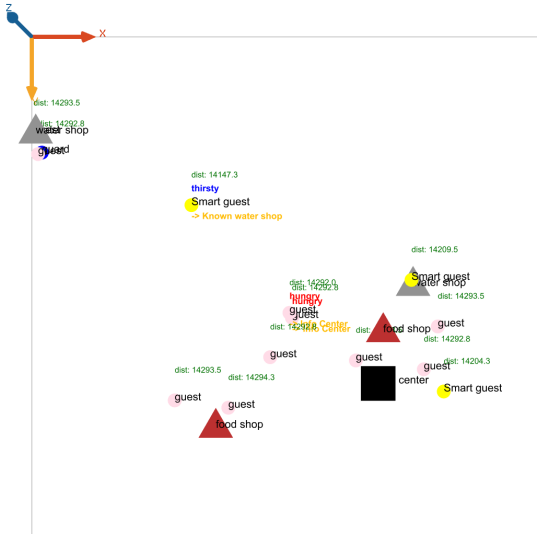

Figure 5: Smart Guest going to known water shop