Assignment 1 - Distributed AI & Intelligent Agents

Perttu Jääskeläinen Group 1 Gabriele Morello
perttuj@kth.se morello@kth.se

November 15, 2022

Contents

1 Introduction 1

2 Implementation 2
2.1 Basic implementation 2
2.2 Challenge 1 - Implementing the brain 2
2.3 Challenge 2 - Adding a security guard 3

3 Discussion 3

1 Introduction

In this assignment, we implemented a festival simulation using the GAMA platform to simulate guests at the event. Using randomized values, the event participants will get more and more hungry or thirsty, and will eventually have to find some store location to get drinks and/or food. The store locations are received from an information centre known to all guests, which will tell them which direction to head in in order to fulfill their needs.

In addition to this implementation, we also improved the functionality by implementing:

- A brain for the festival goers

This means that guest will remember the location of the previous store that they received from the information center. However, in certain scenarios, the guests will want to explore other stores in the area - and will re-visit the information centre to receive another location.

- A security guard kicking out (killing) rude guests

Certain guests will misbehave at the event, and in such cases, a security guard will be dispatched by the information centre to deal with them.

2 Implementation

2.1 Basic implementation

This subsection covers the species included in the basic implementation of the festival. The global section is responsible for instantiating the correct amount of agents, specifying distance threshholds (when an agent is close enough to a store, for example), and the initial location of the information center.

The **FestivalGuest** species describes the individual festival guests - who will eventually get hungry and/or thirsty, and navigate to and communicate with the information center to receive directions to fulfill their needs. When not hungry/thirsty, the festival guests will wander around the area and party with each other.

The **InformationCenter** is a species which responds to guests who have questions about where to find stores. The information center is aware of all store locations.

The **FoodStore**, **DrinkStore** and **DrinkFoodStore** are species that either sell food, drinks, or both. Festival goers will be redirected to one of these stores by the information center to fulfill their needs. To avoid unnecessary trips, guests will buy both food and drink if it's available.

The guest will perform reflexes in the following scenarios: **beIdle** when they are neither hungry or thirsty, **imHungryOrThirsty** when they're heading toward the information center (when hungry/thirsty), **getDirections** when they arrive at the information center, **moveToTarget** when they're heading to a store, and **enterStore** when they're near a store they're heading to.

All stores are triangles, the **FestivalGuest**s are **blue** squares, and the **InformationCenter** is a **green** circle. The stores and guests will change color depending on their state - **red** if they're hungry/selling food, **yellow** if they're thirsty/selling drinks, and **orange** if they're both.

Figure 1 shows an example run of the festival (including challenge implementations).

2.2 Challenge 1 - Implementing the brain

To implement the brain, we extended the functionality of **FestivalGuest**:s by adding a list of previous locations for each store type - Food/Drink/DrinkFood. When the guest was either hungry, thirsty, or both - they could either use a previously known location (if such a location existed) - or decide to further explore the area by re-visiting the **InformationCenter** for a new location. Each time the guest visited the **InformationCenter**, they would "extend their brain" with information about a new location.

The modified reflex were the following: **imHungryOrThirsty**, to have a probability to "look in the brain" for a location rather than head to info center. **getDirections**, to store locations in our brain when receiving them from the info center. We also added the **memoDrink/Food/DrinkFood** to the **FestivalGuest** species to keep track of the locations.

To compare the performance, we used static locations for the stores on Y = 25; with the information center located at 50, 75. The average steps taken by guests without a brain was 110, while with a brain, it improved to a value between 50-60. We compared the results by running the experiment **FestivalBasic** and **FestivalChallenge1** and comparing the results.

The implementation of Challenge 1 can be found in the **FestivalChallenge1** model.

2.3 Challenge 2 - Adding a security guard

For the second challenge, we created a new species - **SecurityGuard**, who walks around the area until they get notified by the information center that someone is misbehaving. In such scenarios, the **SecurityGuard** will start moving in the misbehaving guests direction - killing them when they reach them!

In addition to the new **SecurityGuard** species, we also slightly modified the **InformationCenter** to look out for bad guests in their vicinity - and if any are noticed, they phone the security guard to let them know.

The new reflexes were the following: **anyBadGuests** to look for bad guests near the information center, **notifyGuard** to call a security guard when we find bad guests - **targetAcquired** when the security guard is chasing a guest, and **targetReached** when reaching the guest. We also added the **isBad** property to the **FestivalGuest**, which specifies if they're behaving badly or not. We made the **beIdle** reflex in guests have a probability of 1% for guests to turn bad - which eventually causes all guests to be killed. Misbehaving guests will turn black instead of blue, which makes them easy to see in the simulation.

The implementation of Challenge 2 can be found in the **FestivalChallenge2** model.

3 Discussion

The basic implementation itself was very straight forward - we knew what logic we wanted to implement, and based on the tutorial session we (kind of) knew the syntax to do so. However, learning to do things in the "Gama way" proved to be a bit challenging - where we had some struggles with getting the syntax working correctly and understanding behavior.

For the challenges, we had to further explore the GAMA API to learn how to find agents of certain species, which proved to be a challenge as well. When we learned the API though, it was fairly straight forward to implement - which was the intention of this assignment.

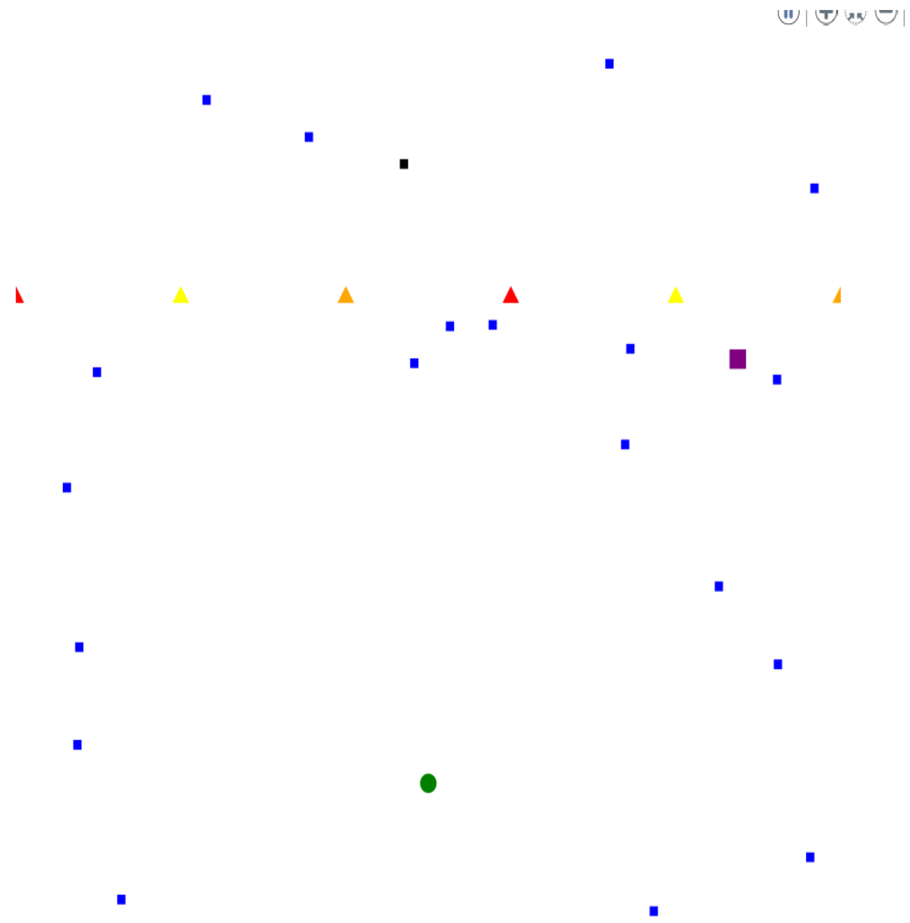

Figure 1: *Example run of the simulation and colors of festival guests (blue), security guard (purple), misbehaving guests (black), stores (triangles) and the information center (circle).*