

ID2209 – Distributed Artificial Intelligence and Intelligent Agents

Assignment 1 – Festival
Group 15
Arthur Simonsson
Zainab Al-Saadi
17.11.21

Assignment

In this assignment, we were tasked with simulating a festival with guests, stores and information centers. The guests should wander around the festival area while not hungry or thirsty. If their status changes, they should seek out an information center and get directions for a store. After that they should walk towards the store and have their statuses changed again.

How to run

Run GAMA 1.7 and import festival.gaml as a new project. Press main to run the simulation.

Short description

Each festival guest is assigned boolean values of their status. If hungry or thirsty, they will seek out the location known by them of an information center. When arriving at a center, another boolean value called *assigned* will be flipped, thus knowing they should seek out a store for their needs. At the store, all their boolean values will be flipped back to their initial value and thus start wandering around the festival grounds.

Species

Guest agent

The behaviour of this agent is wandering until they become thirsty or hungry, that's when they seek an information center to guide them to a restaurant or a juice bar.

Store agent

A store can be a restaurant or a juice bar. Stores get randomly assigned as restaurants or bars during the program initiation.

Information Center agent

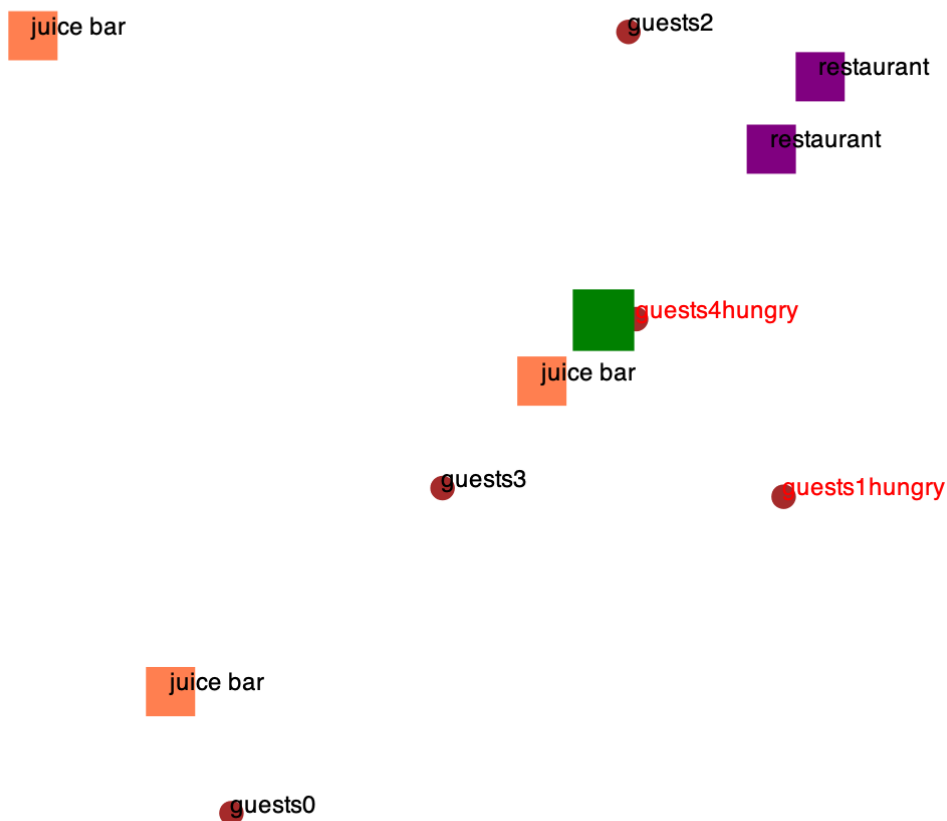
This agent keeps lists of the available restaurants and bars and is responsible for redirecting agents for the specific store depending on their status.

Implementation

We started with developing the guests, making sure their status changes when wandering. Then adding stores and the information center. The next logic to be implemented was that if a guest is hungry or thirsty they should move to the information center. When that was done, the guest arriving at the information center should be assigned a location of a store and thus move towards it. When at the store, it was coded that they should restart wandering around.

Results

The guests go to the information center when hungry or thirsty and find their assigned store accordingly. Below is a screenshot of the base solution, where the green square is the information center.



Challenge 1

To hold information of earlier visited stores, a list variable was added to guests. When assigned a target at the information center, that target was added to the list.

The added lists contain visited restaurants and bars respectively. When guests become hungry or thirsty, they can then choose between visiting a new place by asking the info-center or revisiting an old place, if the list is not empty. The probability of choosing the latest option is set to 60%. Afterwards, the total distance traveled **excluding** wandering was calculated with and without using the “brain”, and results showed the following:

Experiment 1

Description: Print total distance traveled by an agent. The value is printed after 1000 cycles. The probability of revisiting an old store is set to 60%.

Results:

No brain total distance of 5 agents:

travelled distance: 344, 312, 475, 326, 366

Use brain total distance of 5 agents:

travelled distance: 450, 405, 327, 349, 459

Notes: The random placing of stores affects this result.

Experiment 2

Description: Due to random placing of stores, the distances may vary significantly. Thus, to focus only on brain-usage effect on the total traveled distance, the location of stores was fixed around the x-axis in both the base and the ‘challenge1’ modules, to make the comparison more fair.

Results:

No brain total distance of 5 agents:

travelled distance: 346, 419, 435, 323, 374

Use brain total distance of 5 agents:

travelled distance: 351, 211, 242, 325, 248

Notes: we can see a slight difference between the first and the second approach.

Conclusion: Using a “small-brain-agent” reduces the total distance traveled.

Challenge 2

For this challenge, a bad guest and a security guard are created. When a bad guest arrives at a store, it gets reported by calling the security guard. The security guard then follows this guest and kills it when it reaches it. Afterwards, the guard goes back to its place, near the information center.

To differentiate these additions from the others, the bad guest has a different color and the guard is a black hexagon.

Discussion / Conclusion

Creating the behaviour proved no problem overall, it was not difficult to implement the desired behaviour. We learned how to assign targets for agents, and how to supply target information. Overall a fun assignment to get into GAMA.