ID2209 – Distributed Artificial Intelligence and Intelligent Agents

Assignment 1

Group 15

Zhinan Gao

Jinyao Zhou

Nov. 17 2023

In this assignment, we were tasked with creating different types of agents; Stores, Guests and an Information Center. Each type of agent has different behaviors and together they simulate a festival.

How to run

Run GAMA 1.9.2 and import the submission. To run the basic simulation, run asgmt1_base.gaml. Press main to run the simulation. To run the challenge version, run asgmt1_challenge1and2.gaml. By changing the duration of cycle we can slow down the simulation process and get a clearer view of how the simulation works. By changing the nPeople parameter we can set different guest numbers. We can also set different hunger/thirst values to different levels.

Species

Guest

In this simulation, Guest agents engage in dynamic movement, initially wandering until they experience hunger or thirst. Each guest's hunger and thirst values increment randomly until reaching a predefined threshold. Subsequently, guests head to the Information Center to obtain the location of a store. Upon receiving this information, they navigate to the designated store.

Store

Store agents are categorized as either food stores or drink stores, possessing minimal code that sets their location and colors.

Information Center

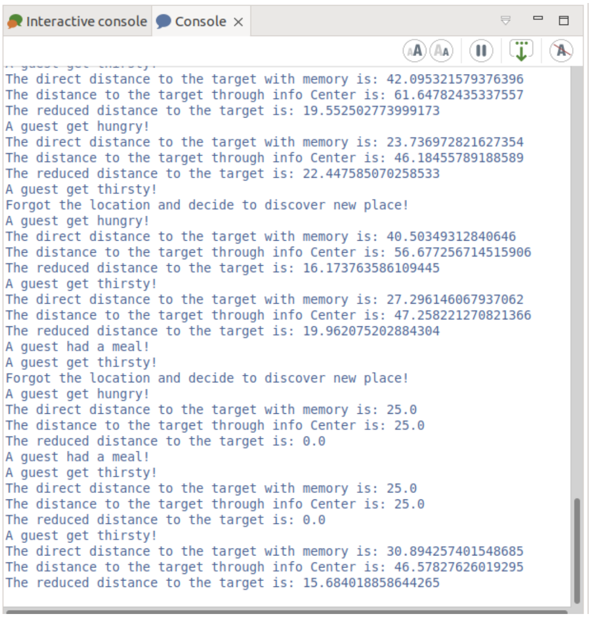
Information Center agents play a crucial role in informing guests about the location of the nearest food store when hungry or the nearest drink store when thirsty.

Implementation

The creation of store agents, encompassing 2 food stores and 2 drink stores, along with the development of 1 information center agents, marked the initial stages of our simulation.

Results

As you can see in the following pic, We successfully completes Challeng 1 and 2.



```
The direct distance to the target with memory is: 42.095321579376396
The distance to the target through info Center is: 61.64782435337557
The reduced distance to the target is: 19.552502773999173
A guest get hungry!
The direct distance to the target with memory is: 23.736972821627354
The distance to the target through info Center is: 46.18455789188589
The reduced distance to the target is: 22.447585070258533
A guest get thirsty!
Forgot the location and decide to discover new place!
A guest get hungry!
The direct distance to the target with memory is: 40.50349312840646
The distance to the target through info Center is: 56.677256714515906
The reduced distance to the target is: 16.173763586109445
A guest get thirsty!
The direct distance to the target with memory is: 27.296146067937062
The distance to the target through info Center is: 47.258221270821366
The reduced distance to the target is: 19.962075202884304
A guest had a meal!
A guest get thirsty!
Forgot the location and decide to discover new place!
A guest get hungry!
The direct distance to the target with memory is: 25.0
The distance to the target through info Center is: 25.0
The reduced distance to the target is: 0.0
A guest had a meal!
A guest get thirsty!
The direct distance to the target with memory is: 25.0
The distance to the target through info Center is: 25.0
The reduced distance to the target is: 0.0
A guest get thirsty!
The direct distance to the target with memory is: 30.894257401548685
The distance to the target through info Center is: 46.57827626019295
The reduced distance to the target is: 15.684018858644265
```

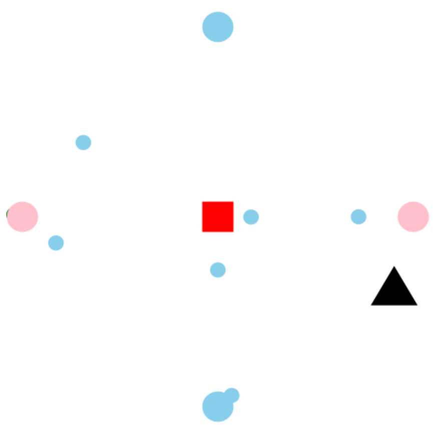


Figure 1: A screenshot of the final solution.

Challenge 1: Memory Function

In the model, the memory function allows Guest agents to remember the locations of stores so they can head directly there in the future when needed, instead of going through the information center each time. The implementation of the memory function involves the following steps:

Initialize Memory: Each Guest agent starts with foodStoreMem and drinkStoreMem attributes, which are initially set to nil, indicating no remembered store locations.

Update Memory: When a Guest agent enters an information center and acquires the location of a store, it updates its memory attributes foodStoreMem or drinkStoreMem.

Utilize Memory: When a Guest agent becomes hungry or thirsty, if there is a stored location in memory, it will set that location as the target (target) directly, bypassing the information center.

Forgetting Rate: After each time a Guest agent fulfills its needs at a store, there is a certain probability (forgetRate) that it forgets the store's location, implemented by the flip(forgetRate) function. If the result is true, the corresponding memory is set to nil.

Challenge 2: Security Guard Function

The security guard function allows SecurityGuard agents to intervene when they detect a Guest with bad behavior. The implementation of this function involves the following steps:

Identify Bad Behavior: Within the Guest agent, a boolean attribute isBad is used to flag whether there is bad behavior. This attribute has a certain chance to be set to true upon agent creation.

Security Intervention: SecurityGuard agents periodically check their target attribute. If there is no target, the agent will patrol (wander). If there is a target and the target's isBad attribute is true, the SecurityGuard will move toward the target's location.

Handling Bad Behavior: Once a SecurityGuard reaches the location of a Guest agent with bad behavior, it executes the code within an ask statement block, which will make the Guest perform the die action, thus simulating the process of security intervention.

Discussion / Conclusion

The model demonstrated a dynamic festival environment where the interactions between agents influenced overall behavioral patterns. Memory function and forgetfulness rate were key factors affecting the efficiency of resource location. The presence of security personnel was crucial in maintaining order. This model can serve as a basis for further research on optimizing festival layout to enhance resource location efficiency and overall safety.