ID2209 – Distributed Artificial Intelligence and Intelligent Agents

Assignment 2 – Festival Auction

Group 36

Matay Mayrany

20.11.2019

Festival Auction

In this assignment, we were tasked with creating a simulation of an auction in GAMA, and implement a series of events associated with how a Dutch auction functions.

How to run

Run GAMA 1.7 and import Auction.gaml as a new project. Press main to run the simulation. Note that changing the number of participants will increase the number of attendees, and the number of initiators will increase the number of auctioneers.

Species

Participant Agent

This agent was responsible for participating in the auction when the auctioneers agent starts it. It will take the price offered by the auctioneer and accept it if it is less than or equal to what the participant is willing to pay, otherwise it will reject it.

Auctioneer Agent

This agent is responsible for starting the auction and informing the participants. It then messages from the participants and the first one to accept the offered price will be announced as the winner. If everyone rejects the auctioneer agent will drop the price by a random interval. Finally, if the price drops below the minimum price set by the auctioneer, they will end the auction.

Implementation

Started by implementing the auctioneer agent. It informs agents that the auction started then sends a cfp message to all potential buyers with the initial offer. The participant was implemented by basically constantly reading to see if the cfp messages are empty. If they are not it reads the offered price, compares it to what it is willing to pay, then accepts or rejects. The function for auctioneer reading the responses was then implemented and if someone accepted they win, otherwise the price is dropped and a new cfp message with the new price is sent to all potential buyers. This goes on until a winner is found or the price drops below the minimum.

Results

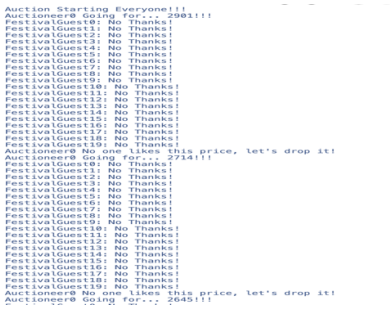

Figure 1: Screenshot of the log showing the auction starting

```
Auctioneer0 Going for... 1938!!!
FestivalGuest0: No Thanks!
FestivalGuest1: No Thanks!
FestivalGuest2: No Thanks!
FestivalGuest3: No Thanks!
FestivalGuest4: No Thanks!
FestivalGuest5: No Thanks!
FestivalGuest6: No Thanks!
FestivalGuest7: I accept!!!
FestivalGuest8: No Thanks!
FestivalGuest9: No Thanks!
FestivalGuest10: No Thanks!
FestivalGuest11: No Thanks!
FestivalGuest12: No Thanks!
FestivalGuest13: No Thanks!
FestivalGuest14: No Thanks!
FestivalGuest15: No Thanks!
FestivalGuest16: No Thanks!
FestivalGuest17: No Thanks!
FestivalGuest18: No Thanks!
FestivalGuest19: No Thanks!
 Found a winner! FestivalGuest(7) won for 1938
```

Figure 2: A screen shot of the log showing the end of the auction

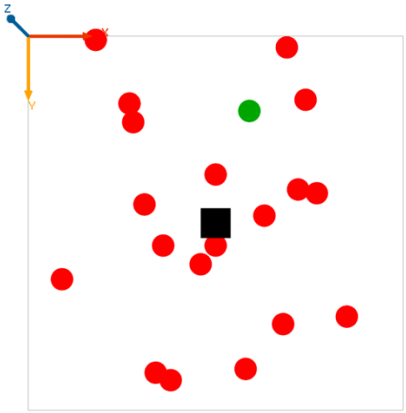

Figure 2: A screenshot of the simulation at the end, showing the auction winner in green.

Challenge 1

N/A

Challenge 2

N/A

Creative implementation

N/A

Discussion / Conclusion

Similar to the previous assignment the level of logic that we are required to implement, for the basic part at least, is quite simple. The problems that I faced were mostly in synchronizing the communication between the festival participants and the auctioneer so no one speaks out of turn or reads messages at an incorrect time, leading to unpredictable behavior.