Assignment 2 - Negotiation and Communication (FIPA)

Group 28

Sara Moazez Gharebagh saramg@kth.se

Sebastian Lihammer lihammer@kth.se

24 November 2021

In this assignment, we were tasked with extending the festival simulation created in assignment 1 by adding auctions. These auctions were to be based on the Dutch auction method, and communication was to be handled through FIPA protocol. The major changes made to the code was to introduce a new species, the *Auctioneer*, and to significantly expand the *FestivalGuest* species.

1. *How to run*

Run Gama 1.8.1 and import the Festival folder as a new project. To run the basic simulation, use Assignment2.*gaml*. Press main to run the simulation. We recommend changing the speed to be slightly lower than maximum to better see what is happening. To run the challenge 1 version, use Challenge.*gaml* instead.

2. Species

2.1. Auctioneer

The *Auctioneer* species is spawned in a random place on the map and will send out a message inviting *FestivalGuest*s to an auction. If enough guests are interested and reach the auctioneer, something the auctioneer keeps track of through the guests adding themselves to its list of attendees, the auction will begin. The auctioneer will send out messages containing the current price to all the guests in the attendees-list. If a guest accepts the price, the auction is over and that guest wins, otherwise the price is lowered and the message is sent out again. If the price reaches the auctioneer's set minimum price, the auction is cancelled without a winner.

2.2. FestivalGuest

The *FestivalGuest* species retains the code and functionality it had in assignment 1, but it has been significantly expanded upon so that guests can partake in auctions. The changes primarily involve sending and receiving messages. Some guests (a randomized number) are interested in going to auctions, and these guests will receive messages from the auctioneer concerning auctions. When an auction is starting, the guests that are interested will move to the auctioneer and, if arriving on time, will add themselves to the auctioneer's list of attendees. At the start of the auction, each guest gets their own set budget, calculated by

subtracting a randomized amount from the starting price of the auction. If the current price for the item being sold reaches below this budget, the guest will buy the item.

2.3. Other species

The species *InformationCenter*, *FoodStore* and *DrinkStore* from assignment 1 are still included in the code for assignment 2, but their code has not been altered or expanded so their functionality remains the same.

3. Implementation

We started developing the auctioneer. We decided that it should be a brown squircle to distinguish it from the other agents. We first created the *initiate_auction* reflex, which informs the participants that an auction will begin with a FIPA request. After this, we implemented the reflex *send_messages*, which first sends the starting price, and later the updated prices, to the participants. After this we began working on expanding the festival guest's functionality: first we made sure that it could go to the auctioneer and then we implemented its ability to receive these messages and send messages of its own.

4. Results

The text below is an excerpt from the log of the simulation, which shows the messages printed when the auction is going on. We can see that the price starts at 940 and that it decreases with a random amount if no participant wants to buy the item for that price. At last, the item is sold to the highest bidder.

```
The participants are:
[FestivalGuest(20),FestivalGuest(24),FestivalGuest(26),Festiva
lGuest(31)]
The auction has begun, starting price is: 940
Minimum price is: 594
FestivalGuest(20) ready to pay: 810
FestivalGuest(24) ready to pay: 669
FestivalGuest(26) ready to pay: 739
FestivalGuest(31) ready to pay: 495
Updating price, current price is now: 893
Updating price, current price is now: 851
Updating price, current price is now: 831
```

```
Updating price, current price is now: 789
Item sold to FestivalGuest(20) auction is now over
```

5. Challenge 1

In the implementation for challenge 1 there are two auctioneers and two auctions that happen simultaneously. The auctioneers have set locations. Guests will be interested in a random item and when the auctioneers announce that an auction will soon begin for a random item, the guests that are interested in this item will participate in that auction.

The text below is an excerpt from the log of the simulation, which shows the messages printed for the auction process.

```
FestivalGuest(4) is interested in item: T-shirts
FestivalGuest(16) is interested in item: T-shirts
FestivalGuest(39) is interested in item: T-shirts
FestivalGuest(9) is interested in item: CDs
FestivalGuest(22) is interested in item: CDs
FestivalGuest(34) is interested in item: CDs
The participants for the auction of item: T-shirts are:
[FestivalGuest(39),FestivalGuest(16),FestivalGuest(4)]
The auction has begun, starting price of item: T-shirts is: 435
Minimum price for item: T-shirts is: 296
FestivalGuest(4) ready to pay: 329 for item: T-shirts
FestivalGuest(16) ready to pay: 48 for item: T-shirts
FestivalGuest(39) ready to pay: 206 for item: T-shirts
Updating price, current price is now: 422 of item: T-shirts
Updating price, current price is now: 386 of item: T-shirts
Updating price, current price is now: 375 of item: T-shirts
Updating price, current price is now: 351 of item: T-shirts
Updating price, current price is now: 323 of item: T-shirts
Item: T-shirts sold to FestivalGuest(4) auction is now over
The participants for the auction of item: CDs are:
[FestivalGuest(34),FestivalGuest(9),FestivalGuest(22)]
The auction has begun, starting price of item: CDs is: 789
Minimum price for item: CDs is: 469
FestivalGuest(9) ready to pay: 599 for item: CDs
FestivalGuest(22) ready to pay: 543 for item: CDs
FestivalGuest(34) ready to pay: 604 for item: CDs
Updating price, current price is now: 769 of item: CDs
Updating price, current price is now: 738 of item: CDs
Updating price, current price is now: 694 of item: CDs
Updating price, current price is now: 659 of item: CDs
Updating price, current price is now: 621 of item: CDs
Updating price, current price is now: 607 of item: CDs
```

```
Updating price, current price is now: 565 of item: CDs
Item: CDs sold to FestivalGuest(9) auction is now over
```

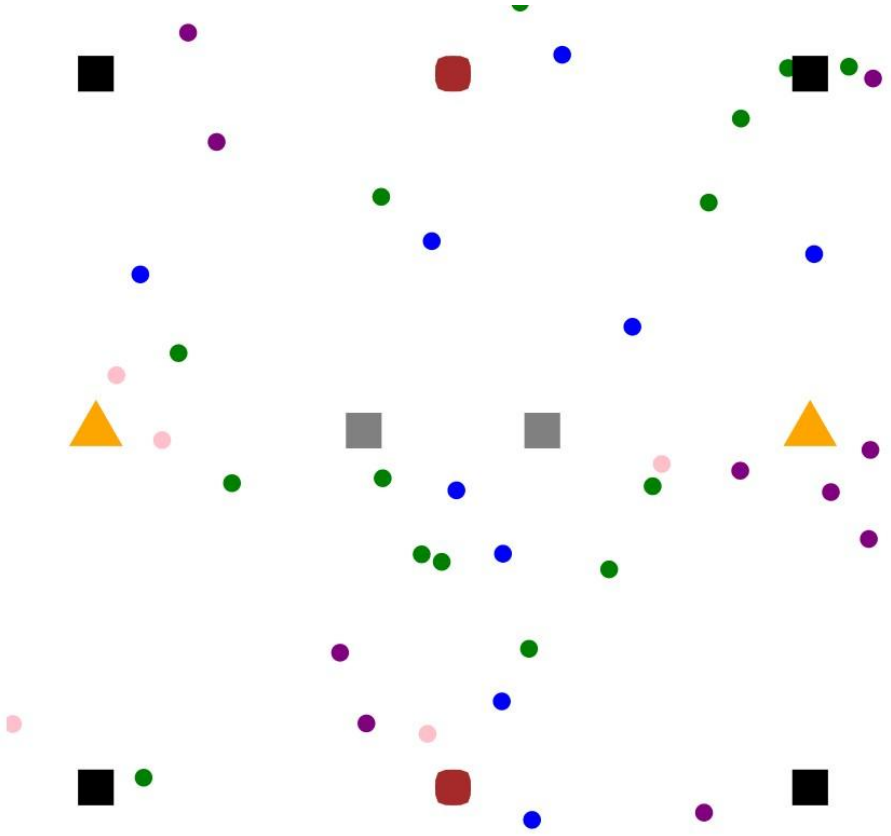

Figure 1: Screenshot of the simulation with Challenge 1 implemented

6. Discussion/Conclusion

Overall, the implementation of the basic assignment went well. But the challenge was a bit tricky and time consuming because it was hard to make two auctions work simultaneously. However, both members think that the assignment was very educational, especially when it comes to experimenting with the FIPA protocol.