ID2209 – Distributed Artificial Intelligence and Intelligent Agents

Assignment 2 – GAMA and Agents

Group 14
Ilias Merentitis
Chrysoula Dikonimaki

23.11.2021

In this assignment, our task was to implement different types of auction using the FIPA protocol.

How to run

Run Gama.exe and import the .gaml files.Press the green button to execute the experiment called myExperiment (each .gaml file has a separate experiment).

Species

Guest

The Guests are agents that wander around the premises of the festival and take part in auctions where items of their interest are being sold (this does not affect their participation in any auction. That is to say, they don't have to go close to the auctioneer in order to participate). Depending on the type of the auction or the amount they are willing to pay they offer or accept bids. The aim of the hosts is that the agent willing to pay the most gets the item.

Auctioneer

The Auctioneer is an agent constituting part of the Festival. He is responsible for selling a number of available items, trying to find the guest willing to pay the most. Throughout the experiment 3 kinds of auctions are implemented, therefore the exact methodology varies, and will be explained later on.

Implementation

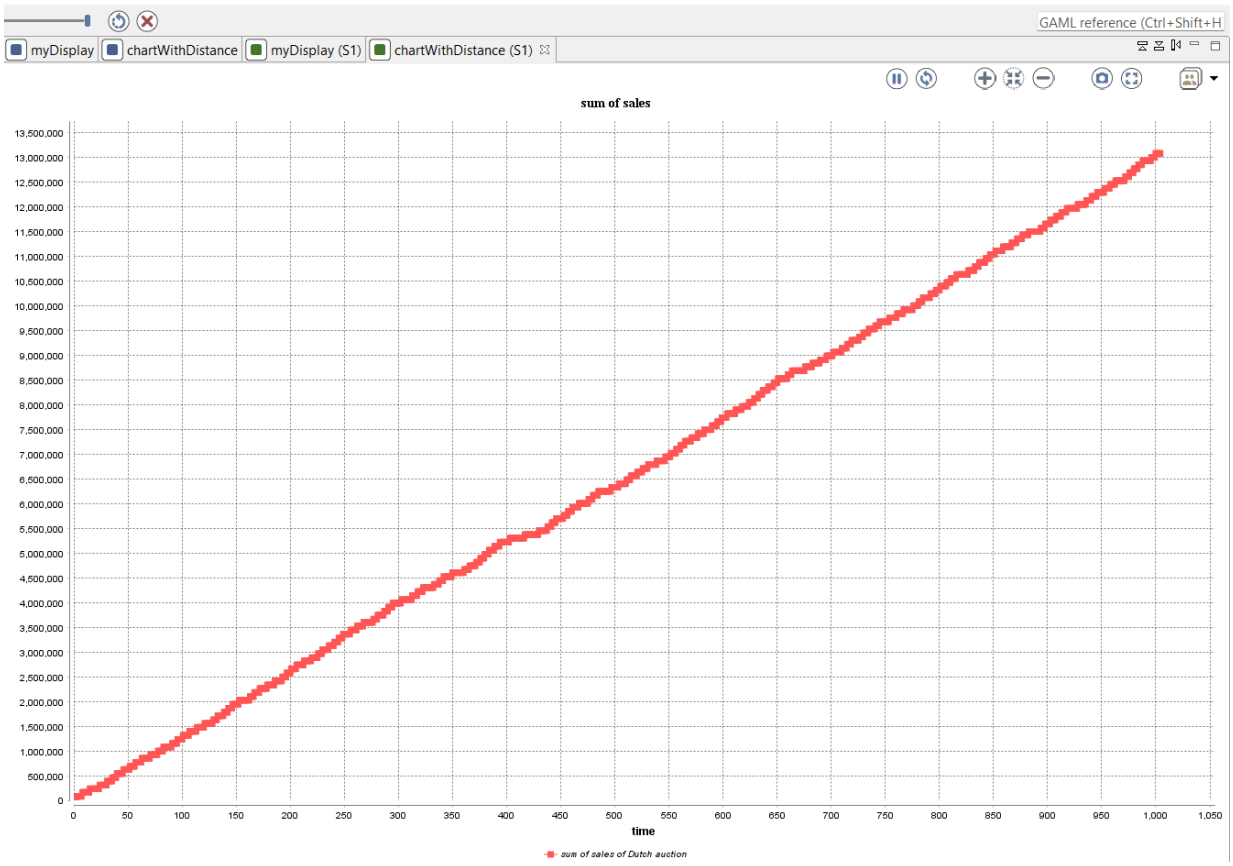
Three separate files (Festival1.gaml, Festival2.gaml, Festival3.gaml) have been created. Festival1 contains the implementation of the basic part and the implementation of the challenge 1, ie: the Dutch auction. Festival2 is the English auction implementation and Festival3 is the Sealed bid auction implementation.

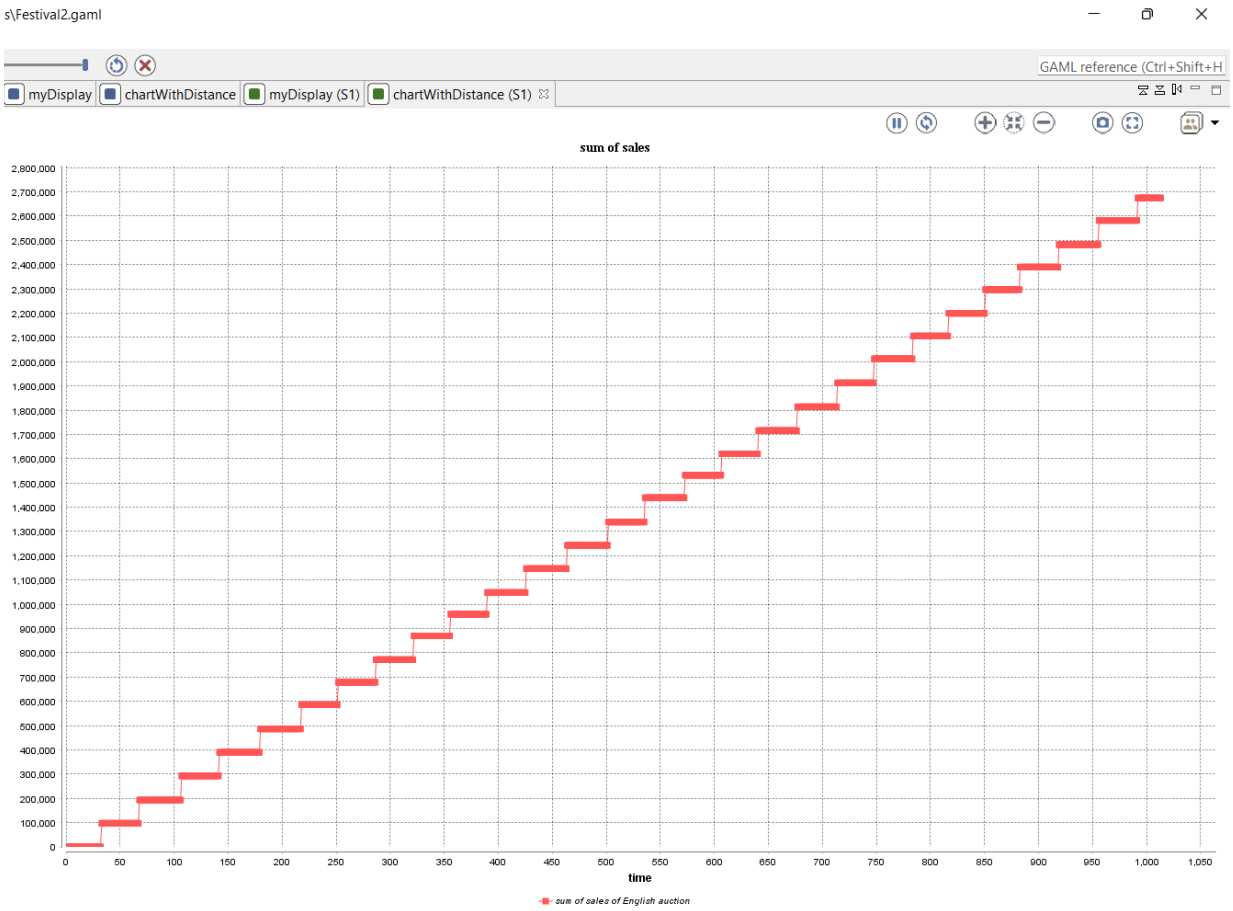
Results

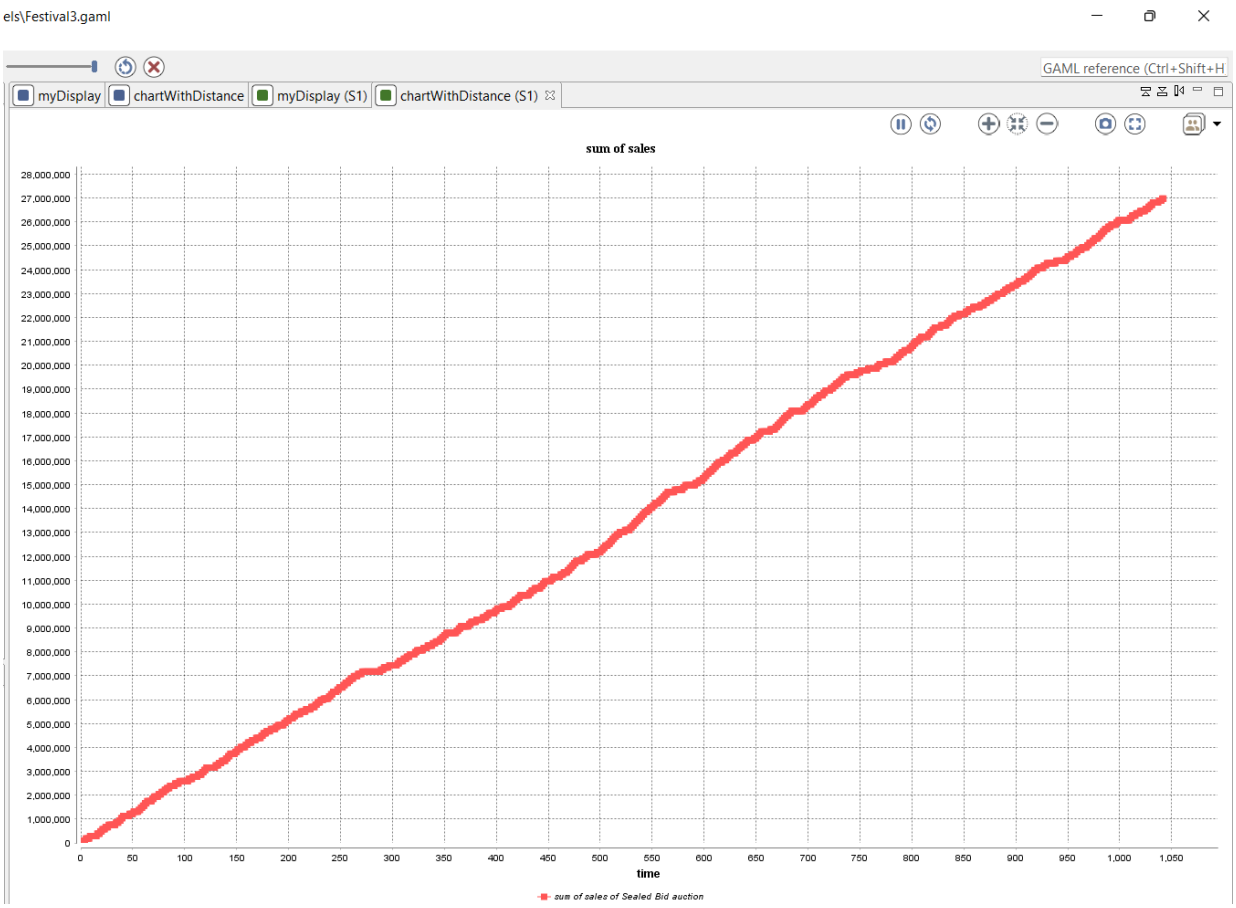
In order to demonstrate how each solution affects the earning of a festival, we have created chart builds of the total amounts being collected for each experiment. To eliminate any differences due to randomised variables we have also provided the same seeds.

Below you can see screenshots showing the total amounts collected for each type of auction after about 1000 time steps.

ls\Festival1.gaml




s\Festival2.gaml
GAML reference (Ctrl+Shift+H
myDisplay
chartWithDistance
myDisplay (S1)
chartWithDistance (S1)
sum of sales
2,800,000
2,700,000
2,600,000
2,500,000
2,400,000
2,300,000
2,200,000
2,100,000
2,000,000
1,900,000
1,800,000
1,700,000
1,600,000
1,500,000
1,400,000
1,300,000
1,200,000
1,100,000
1,000,000
900,000
800,000
700,000
600,000
500,000
400,000
300,000
200,000
100,000
0
0
50
100
150
200
250
300
350
400
450
500
550
600
650
700
750
800
850
900
950
1,000
1,050
time
sum of sales of English auction



An important aspect to note is that the methodology of each auction type is not the only factor at play when comparing the earrings of the auction. Since an auction can start by an idle auctioneer with a 40% probability at every time step (a choice that was totally arbitrary), that means that auctions that last less (ex: sealed bid) will start and finish more times within a given time frame.

Base concept & Challenge 1: Multiple Auctions in the Festival

The base concept of the experiment was to implement a Dutch auction. While the experiment runs, an idle auctioneer has a 40% chance to be willing to start a new auction. He will then select an item (out of 4 in our case) and set the starting price at 100.000Δρχ. It is expected that no guest will accept this in the beginning and not submit an offer of acceptance. Thus the auctioneer will gradually lower the price until the new price eventually falls under the limit of a guest's maximum price. When that happens the guest wins the item.

For challenge 1 we enabled the Festival to host more than one auctioneer at the same time. That means that more than 1 auction can happen simultaneously. In fact this is set as a parameter in the init part of the code. Setting the numAuctioneers <- 1 will essentially have challenge 1 working as the basic concept of the experiment.

When an auction begins, guests that have selected as an item of interest the kind of item the auctioneer is willing to sell get informed by the new auction (inform) and then sent the initial

offer with the artificially high price (cfp). If an agent does not agree (ie: is not willing to pay the current amount) he sends back an offer with 0Δρχ as price. However, if he is willing to pay then he sends back an offer with the same amount as the current price. This is interpreted as him willing to pay the current price. At this point he is selected as a winner and the auction ends, with a 40% probability to start again at any point.

Challenge 2: Different Auctions (English, Sealed bid)

Challenge 2 was an expansion of the basic idea but including 2 additional types of auctions. We selected the English auction for Festival2 and the Sealed Bid auction for festival3.

Both auctions start with the Auctioneer sending an inform message to all guests interested in a particular type of object.

In the Sealed Bid auction, which is the simplest one, guests make a randomised offer and send it as an agree message. The Auctioneer loops through the responses and selects the highest bidder to pronounce a winner.

The English auction is a bit more complicated. After the correct guests have been informed of the new auction, the auctioneer sends a cfp message to them with an initial price, which is artificially set to something very low, so that we can ensure that all participating guests will take part in the first round (although we didn't have to do that necessarily). The exit condition of a winning guest in the loop of messages is when the proposed price is the one he proposed in the previous round. This is how we know he has won. He proposes it again and the auctioneer will then get it, as well as 0 value responses by all other participants, which have been interpreted as withdrawn. Otherwise, a guest may be willing to propose a raise which corresponds to a fraction of his maximum price (or his actual maximum price if there is little room for proposals left). One by one all but 1 auctioneer will stop submitting raises (considered withdrawn) and thus the last one standing wins the auction.