ID2209 – Distributed Artificial Intelligence and Intelligent Agents

Assignment 3 – Coordination & Utility

Group 27

Valeria Bladinieres Justo

Raúl Aznar Álvarez

29/11/2019

Task1: N Queens Problem

The problem of N queens consists on having a number of N queens on a chess board, and each queen must be placed on the board in a specific way where no queen can attack each other. In this task agents must be able to communicate and cooperate between each other to achieve the main goal. In order to solve the problem, the board has to be the same number of rows and columns as the number of queens. The rules are the same as in chess, the queens are able to move diagonally, vertically and horizontally. Additionally, the queens can communicate only the predecessor and the successor with the protocol FIPA. If the current queen has no available moves left, she notifies the predecessor to give her more moves, therefore the predecessor will change position to get new moves for the successor. Everything stops when the queens are correctly positioned.

Task2: Festival Highest Utility

In this part of the assignment, there are two main agents Guests and Stages. There are four stages in the festival, each of them have different attributes which gives them more value, like lights visual effects, sound, band popularity, dance chance, and duration. The guest choses the stage regarding their own preferences multiplied by the values of all stages, then decides the one which has the highest value and goes to that stage, this called the utility function. All the stages have different concerts at the same time and have different durations and start different ones after a random time. Additionally, distinct genders can be observed randomly at the Stages of music like Pop, Rock, Latin, Electro, and HipHop. The communication between the Stages and the Guests are throughout the protocol FIPA.

How to run

Run GAMA 1.7 and open the file Assignment3, there will be 4 scripts Assignment3_Task1.gaml, Assignment3_Task2.gaml, Assignment3_Challenge.gaml and Assignment3_creative.galm. Open the file which wants to be checked and press main (left top of script window) to run the simulation. Note that changing parameters of number of guests can be done without any repercussions.

Task1: Species

Queen

All of the queens start by positioning its selves in their respective column on the board, then the algorithm of moving each one of them to the corresponding row starts. The queens have four main reflexes, the first two reflexes are for the base case concerning the first queen and the other two reflexes are for the rest of the queens. The first action of the queen is to find all the possible places on the board where another queen cannot be reached, then it passes it to the successor and the successor checks if there is any of this positions that can be reached by her, if not it is notified to the predecessor that there are not possible moves and the predecessor moves and recalculates the possible positions for the successor. Otherwise, the queen will pass the possible positions to the successor and it goes on until all queens are positioned on the board, then the problem is solved.

Implementation

First, the chess board was created with a species grid using attributes like color to simulate the black and white background of the board. Then, the queens were created, each in their own row and static. In the end, the interaction and functionality between the queen was implemented, since it was the most challenging.

Results

In *figure* 1 it can be observed that 10 Queens are positioned in the chess board ready to star, then in *figure* 2 it can be seen the first round of positioning the queens until one stop because can't find any correct positions, in this case queen 7. In the end, in *figure* 3 all the queens are positioned in the correct way fulfilling all the rules.

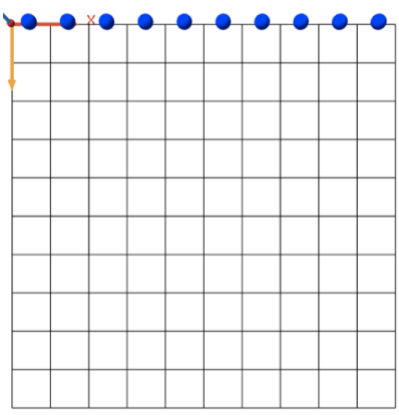

Figure 1

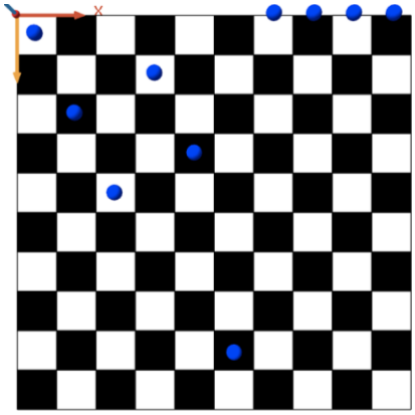

Figure 2

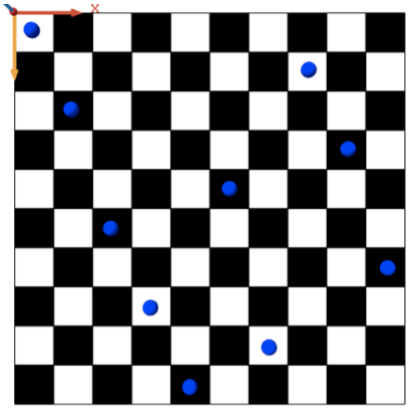

Figure 3

Task2: Species

Guest

These agents are wondering around at a random place in the festival, each one of them have its own preferences regarding the qualities of any concert (lights visual effects, sound, band popularity, dance chance, and duration), these preferences do not change over the time of the festival. Then, at any time the stages inform throughout FIPA protocol to the guest that there is a concert coming, and it send all the variables of the concert with it. Then the agent then processes an utility function for each stage, analyzing its preferences and the concert's, and in the end, the concert with the higher value is the one that the guest will go.

Stages

Each stage decides randomly the genre of the concert it is going to host, also the weight of the attributes. Then, it sends an inform through the FIPA protocol to the guest, sending the location of the stage and its attributes. In the end, concerts is finished since it has a delimited time, and it waits a random time to start another, the attributes are selected randomly again for the next hosting concert.

Implementation

First, we created all the needed agents in this case Guests and Stages. Then we started designing each agent with their basics for the graphic design. Then the development of both agents was simultaneous, in order to follow the protocol and their own behavior. The utility function was the core of this tasks, and it was the final part.

Results

In *figure* 1 it can be observed that the Guests are wandering around the festival, and they are informed by the stages about the concerts. Then, in *figure 2* it can be seen that the guest arrives at the stages which the utility function gave the highest value.

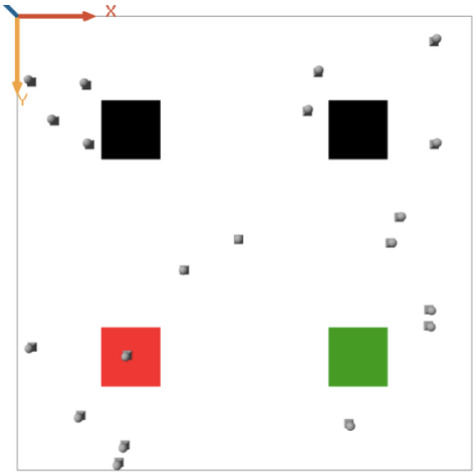

Figure 1

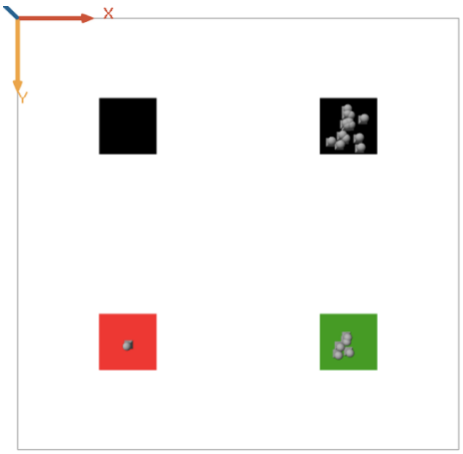

Figure 2

Figures 3 and 4 an example of the FIPA protocol, where highlighted in red it can be observed the CFP to all interested agents, the round of the price reductions since it did not get a propose, and in the end the auctioneer receives a propose and the winner agent is announced.

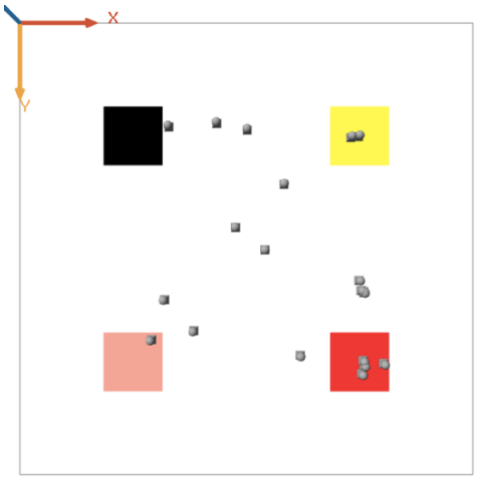

Figure 3

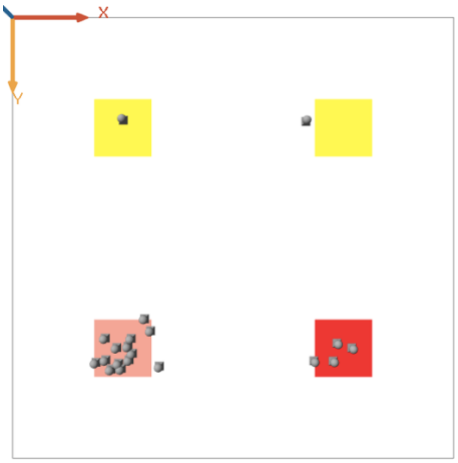

Figure 4

Challenge: Global utility

For the challenge, we are asked to take into consideration the crowd mass and the global utility. To do so, we assigned all the guests a preferred crowd size, where the guests feel more comfortable.

Firstly, the utility of a new stage is calculated, and then via FIPA a message is sent to the Leader (which is a new species that manages all the Guests). Once the Leader receives the inquiry, it calculates the utilities for the guest to go to the different Stages, taking in account the crowd that it will be with him/her. After that, we are able to see which ones are the utilities taking the crowd mass value into account. The formula used for that multiplies the utility prior to the crowd by 5 and then it divides it between the difference of the crowd size and the size in the stage (adding 0,1 in order to avoid infinites). With that the utilities for the Guest in the 4 different Stages are calculated.

Secondly, the calculation of the global utility is done. To do so, the same principle is used. We calculate using 4 different lists, what would be the result for all the guests if the Guest who has to decide goes to Stage0, Stage 1, Stage 2 or Stage 3. Using the formula from before the global utility for the 4 cases are calculated.

Finally, the Leader tells the Guest which is the best option to maximize the global utility and send the Guest to that Stage.

In the pictures from below we can see two examples. How the utility changes after using the crowd mass attribute. And how the decision affects the crowd. In Figure 5, the best utility prior to the crowd was Stage 0, using crowd was still Stage0, however, the one that maximized the global utility was Stage 1, and as a result it goes there. In Figure 6, before using crowd, the Stage 2 was the chosen one, but as Stage 3 had only 1 person and the desired crowd size for that Guest was 2, the formula for Stage 3 maximizes the results making that the choice for him/her. Luckily for him/her it was also the option which maximizes the global utility.

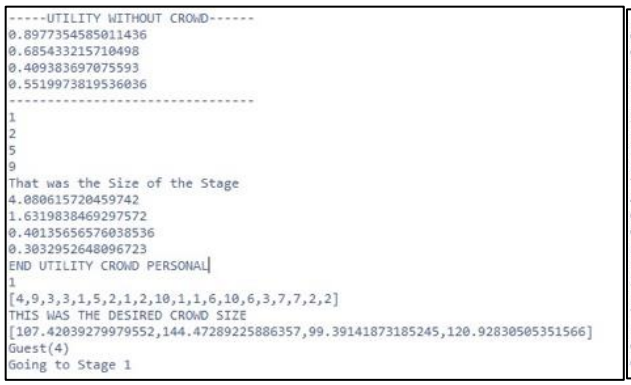

Figure 5

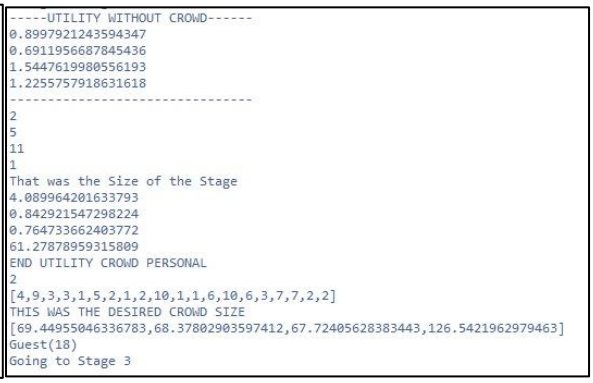

Figure 6

Creative implementation

In the creative tasks, we decided to implement lovers and add some images to represent them. First in the stone age, there are dinosaurs and they can go to four different parts (pond, volcano, plants and cave). During this process, a dinosaur can randomly be in-love with another dinosaur, the in-love agent goes to its lover location and lets it know he is in love by sending his preference attributives. Afterwards, the agent will see the information of the in-love dinosaur and decide if they are compatible. If so, they will send a positive response, otherwise, it will say he doesn't want to be with him. The compatibility is measured if each one of its elements is greater than 66% match within a variation of +-30% of the elements with the in-love dinosaur.

In *figure 7* it can be seen a dinosaur is in love and on its way to meet him, after meeting in *figure 8* it can be seen that the two dinosaurs are light in hearts since the agent accepted to be lovers since their compatibility were greater than 4 out of 6.

Figure 7

Figure 8

In *figure 9* it can be observed the steps of the message exchanging and the compatibility decision. In *Figure 9* the guest 12 accepts the guest 15 as a partner because the compatibility was 5/6. In *figure 10 the* compatibility is 3/6 therefore is less than 66% and it not accepted.

```
Guest15: I am in love with Guest12
Guest15: telling Guest12 I am in love
Guest12: being notified i have been liked from Guest(15)
------------
Likes of guest in love with me, Guest(15):
Lights: 0.7272548252412763
Visualeffects: 0.5281214081084241
Sound: 0.9474684790114938
BandPopularity: 0.30471381087620797
Dancechance: 0.4048342335225482
Duration: 0.38561452003716845
My ikes :
Lights: 0.7666067415678082
Visualeffects: 0.7763013059291998
Sound: 0.8353079852718232
BandPopularity: 0.7809824859498126
Dancechance: 0.5942068515818862
Duration: 0.25949242229929737
Their compabality is: 5
they are a couple: Guest12 and Guest(15)
------------
    Guest(12) has said YES to me Guest15
```

Figure 9

```
Guest6: I am in love with Guest13
Guest6: telling Guest13 I am in love
Guest13: being notified i have been liked from Guest(6)
------------
Likes of guest in love with me, Guest(6):
Lights: 0.6314846795831809
Visualeffects: 0.9021696256834698
Sound: 0.7637896293303301
BandPopularity: 0.548144519206091
Dancechance: 0.5036291250601128
Duration: 0.133487270037577
My ikes :
Lights: 0.051468510408300694
Visualeffects: 0.7428868481273441
Sound: 0.7586181444568307
BandPopularity: 0.22878706676612648
Dancechance: 0.5462243808795598
Duration: 0.435519893364975
Their compabality is: 3
------------
    Guest(13) has said NO to me Guest6
```

Figure 10

After 1500 times, the scenario changes completely since it was hit by a meteorite, and the new comes. The era of the robots. All the couples are reset it, because everything was whipped out and started again. In *figure 11 it* can be observed the new era, where the places the robots visit are futuristic!

Figure 11

Answer Qualitative/Quantitative questions

In what area is your idea mostly related to	Image processing, time managing, FIPA protocol.
Time spent on finding and developing the creative part	8
On the scale of 1-5, how much did the extra feature add to the assignment?	4
On the scale of 1-5, how much did you learn from implementing your feature?	4

Discussion / Conclusion

Creating this project was more challenging than the past two, we encountered numerous problems regarding coordination and utility. The challenge was much more complicated than the previous ones, and since there were two different essential parts to develop in assignment the attention was a little disperse. The creative idea was fun and interesting as always, we decided to take a different perspective again and play between the visuals and the challenge. In general, the assignment was exciting and curious, we learned how to apply real computer science problems in a different perspective like the N queens in combination with the FIPA protocol, also the coordination between agents and the use of utility functions.