ID2209 – Distributed Artificial Intelligence and Intelligent Agents

Assignment 3 – Agents & Utility

Course Coordinator
Mihhail Matskin

Course Assistants
Mehdi Satei
Óttar Guðmundsson
Shatha Jaradat

Group 18
Fredrik Bitén

Prateek Mathur

29/11-2018

Species

Guest

The guest is an agent. The guest has the following attributes

Color - shows the different behaviors of the guest

six variables for preferences, lights, visuals, sound, weather, band, security.

Target point - target location of where the agent is going

the guest agent implements the following behaviors:

Idle – Roaming around at the starting position, waiting for the concert to begin

Move – When the utility has been calculated, move towards the stage that benefits you the most

Dislikes crowds – The variable crowd_mass decides whether the guests enjoy crowded concerts or not. If The stage is to crowd the guest will change to a less crowded one, even if the utility will be lower.

Stage

the stage is an agent. The stage has the following attributes

six variables for utilities, lights, visuals, sound, weather, band, security.

attendance - number of people at the stage

crowd mass - number of people that are going to the stage

Costs: each stage has a fixed cost to run, and a variable cost dependent upon the attending guests. Security and air conditioning costs depend upon the guests; however, the bands have to be paid if stage is kept open.

the stage agent implements the following behaviors:

Open – Guests will arrive at the stage and enjoy the music

Closed – No guests have arrived, and the stage is closed (Creativity)

Stage director

Stage Director (Creativity)

The stage director is an agent. The stage director has the following attributes:

The stage agent implements the following behaviors:

Idle – Waiting for each guest to have picked a stage.

Moving - When all guests have picked a stage the director will move to the control room to shut down stages that have no attendance. The control room agent also keeps track of the total cost of running the festival.

Queen (Queens problem)

The queen is an agent. The queen has the following attributes:

Position – current position

Safe positions – Positions taken from the previous queen that shows safe places.

Implementation (Festival)

We started developing the guest with all its features and made sure that it could both implement the moving function so that it could travel across the map. We then made sure it could communicate with each other and the stage using the ask function.

The second agent implemented was the stage. We made sure that each stage got their own utility, so each stage could enjoy several guests if the preferences of each guest matched.

Challenge. The Utility was done by adding six different preferences for each guest. The guests calculate their preference with each stage to determine which stage matches their interests the most.

After calculation has been done, each agent starts traveling to their preferred stage. The crowd mass variable will increase for each agent that has decided to travel to one of the four stages.

The guest will have the knowledge of how many people are traveling and can decide on that factor if they want to still visit that stage or pick one with a smaller crowd.

If only two people are attending a concert, and one of them likes crowded places and the other one not, the first one will decide to change their location pick to a more crowded stage.

Implementation (Queens)

We started off by adding a NxN grid. We then added queen based on knowledge it got from asking the previous placed queen about safe positions. This was done until all queens were placed or until the was no safe positions left. If no possible position was left, we backtracked to the previous placed queen and asked her to check if she could find another position. We iterated through this until a solution was found. Then we displayed the results on screen

Results

From the logs we illustrate a bit from the challenge one part, and from the creative implementation of closing stages and asking guests to leave if too few guests are attending the concert.

```
**********************************************************************************************************************************************************
guest The highest Utility for a stage was given by stage(3) and the utility value was: 2.0727488892829364
The utility value for each stage was:
------------------------------
stage(0) = 1.5100050019897573
------------------------------
stage(1) = 1.6295757106451998
------------------------------
stage(2) = 1.818223894818253
------------------------------
stage(3) = 2.0727488892829364
**********************************************************************************************************************************************************
guest The highest Utility for a stage was given by stage(3) and the utility value was: 2.1678106063208946
The utility value for each stage was:
------------------------------
stage(0) = 2.0257285924809265
------------------------------
stage(1) = 1.9651731078858927
------------------------------
stage(2) = 1.9620315774661248
------------------------------
stage(3) = 2.1678106063208946
Initial total utility: 73.0
End total utility    : 73.80000000000001
```

```
stage(3) = 2.2227427942736275
stage(1) : Too few guests, please leave
stage(0) : the stage is closed
stage(1) : the stage is closed
```

Challenge 1

The first challenges were to add a behavior for each guest that impacted their choice of stage.

The initial pick of stage is made of calculating the utility for all stages. The guests then check how many people will attend or are attending the concert and do a new pick based on their liking of crowded stages.

Creative implementation

For the creativity part a stage director agent was added. The object of the agent is to have an overview of each stage and when all guests are happy with their stage choice, close the stages with 0 attendance to save money for the festival arrangers. The following illustration is an example when there are two stages not being attended. Since the band performing at each stage will have to be paid if people attend their concert, if no agent attends or too few guests are attending, the cost of the festival will be a constant high. By closing the stages, the net cost is dramatically reduced.

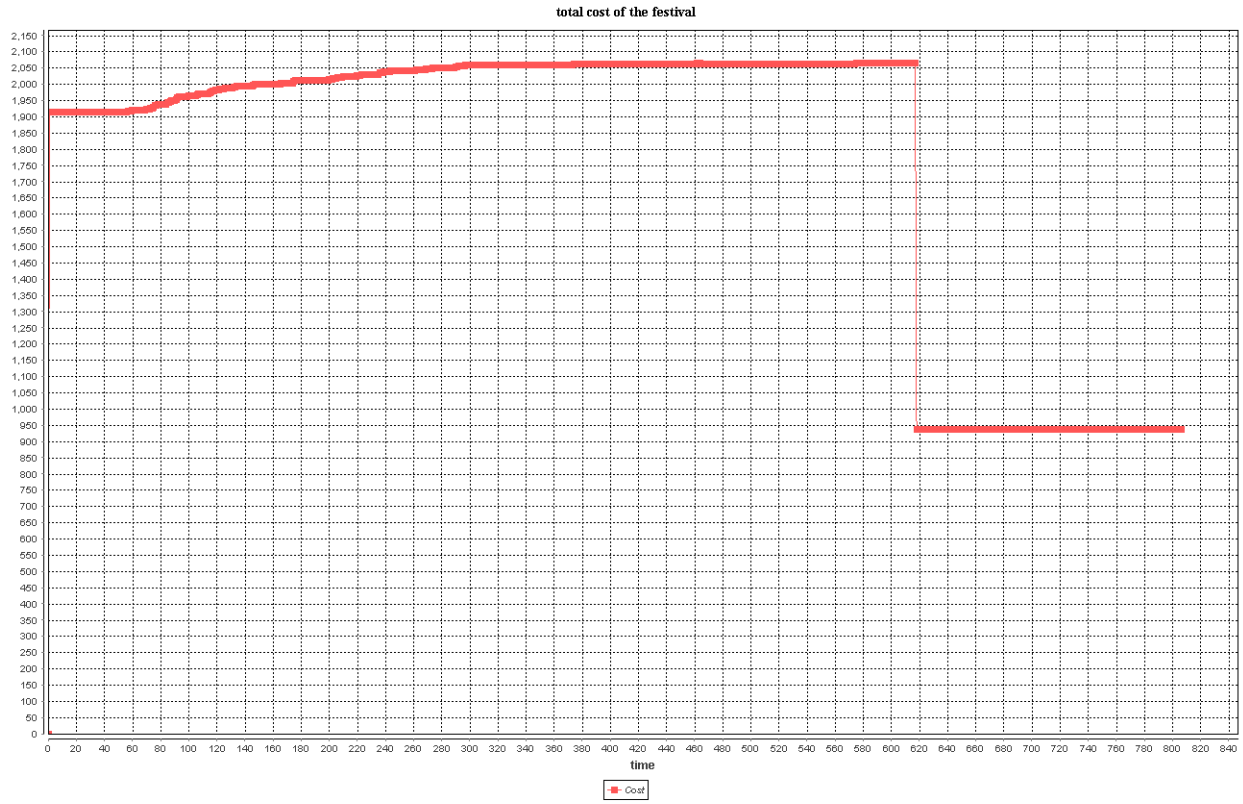

Figure: Total cost of running the festival. As people start entering and going to stages, the cost increases due to additional security and air conditioning required to keep guests happy. When guests are happy with where they are, the stage director closes the un-attended stages or asks the stages to send people away if there are too few people, and then closes the stage. Thus, removing the costs incurred by that stage.

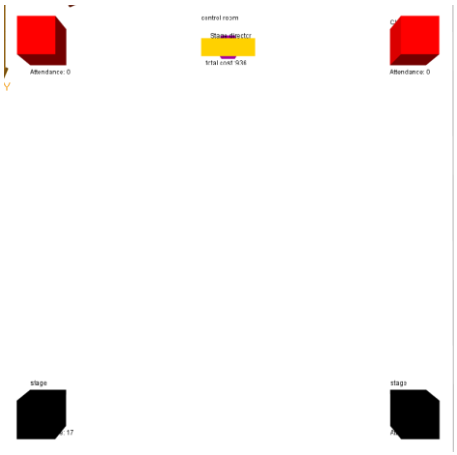

Figure: A snapshot of the simulation.

Qualitative/Quantitative questions	*Answer*
Time spent on finding and developing the creative part	*1 days*
In what area is your idea mostly related to…	*Real life situations*
On the scale of 1-5, how much did the extra feature add to the assignment?	*5-it added a factor which has a lot of meaning to it, some realism.*
On the scale of 1-5, how much did you learn from implementing your feature?	*3-Not as much as the previous two labs. We were both more familiar with the language and what to do.*

Discussion / Conclusion

We did not stumble across that many problems when doing the implementation. The hardest part was the queen's problem which took more time than the festival scenario. We were both familiar with the language, which made this lab a bit faster than the previous ones.

We did learn a lot from this. We argue that we should receive a bonus point for creativity in this assignment because the feature we added brought in a bit of realism in this type of scenario and added more the utility spectrum.