Distributed Artificial Intelligence and Intelligent Agents ID2209

Assignment 3

Group 22
Ayushman Khazanchi
Vasigaran Senthilkumar
30.11.2021

Introduction

In Assignment 3, we were tasked with implementing agents communicating with each other in a chess grid of size N*N to solve N queen's problem. In the second task we were tasked with using communication among agents to decide which concerts to attend to based on the result of their utility functions. In the N queen's problem task the Queen has to communicate only to its predecessor or the successor, and all the queens have to finally move to an ideal position where no two queens are in same row, column, or diagonal. We used FIPA protocol to communicate between agents.

How to run

Note: This section assumes user knows how to run Gama IDE and how to run experiments within the IDE.

First task can be run from Queen.gaml
Second task can be run from StageUtility.gaml

Species

Grid

A grid which consists of cells and the queen agents occupy the coordinates on the grid. The size of the grid N*N where N is the number of queens in the N Queens problem.

Queen Agent

There are N number of Queen agents in a grid of size N*N. The queen agents are initialised with their predecessor, index, and successor. The first queen is placed randomly in the grid. For the other agents it is first checked whether any other queens are in their row or column or diagonal after which they are placed. In case they have no space then they communicate with their predecessor and successor.

Stage Agent

The stage agent represents the stages where the concerts are taking place. Each stage agent has attributes that can attract a guest: lightshow, speakers, band, size of stage, and smoke. These are set to new values at the start of each concert.

Guest Agent

The guest agent represents the guests that are attending the music festival. They all have the same attributes as the stage agent but these attributes represent the guest's desires. Depending on which stage maximizes the utility function of the guest agent's the guest goes to watch that music concert. The guest's attributes do not change through the model.

Implementation

N-Queen

The first main task is done in Queen.gaml. The N number of Queens are created. The first queen occupies a random position in the grid. The predecessor of the queen(queen[0]) is the queen itself and the successor is the second queen (queen[1]). The first queen informs its successor queen to find a position on the grid. The successor queen chooses position on the grid so that no other earlier queen occupies the same row or same column or falls along the diagonal of the position. Then it asks its successor agent to find its position. At certain point the queen might not be able to find a position that matches the above conditions. Hence it asks its predecessor to relocate through a conversation. The predecessor checks if it can occupy another position. If it's unable to find a safe location, it in turn asks its predecessor to relocate. This continues until a predecessor finally finds a position. Then the predecessor communicates with their successor that it has relocated and asks the successor to relocate. The process continues until all the queens occupy ideal positions on the grid.

Following are the number of cycles taken to solve the N-queen problem for various different values of N.

N value	No of cycles for simulating N queen problem
4	2
8	23
12	1700
16	7300

Utility Functions

The utility functions task is run from StageUtility.gaml. Four stage locations are initialized on a grid that represent the four places where a concert can take place. Each stage agent starts a concert and defines some attributes associated to the concert. These attributes are lightshow, speakers, band, size, smoke. These are communicated to guests via the FIPA request protocol. Each concert runs for a set time (concertDuration) and concerts for each stage can start at different times.

The guest agents are initialized with a set of attributes that are the same as that of the stage however the values of these attributes for the guest agent do not change once they're initialized. The guest then calculates the maximum utility for each stage and they head to watch the concert at the stage where they receive the maximum utility. Once a concert is over, the guest proceeds back to the centre area of the map to then decide on which next concert to attend once they start.

Discussion / Conclusion

We enjoyed working on the simulation. Solving the N-Queen problems without using the algorithm itself and communicating via queens was a unique approach to the problem. We understood agent communication and utility functions in more detail. We strengthened our understanding of FIPA communication through GAML as well.