ID2209 – Distributed Artificial Intelligence and Intelligent Agents

Project Report - Group 19

Group 19

Abdullah Abdullah
Aksel Uhr
31/12-2021

1. Introduction

The purpose of the project was to implement different behavior and interactions among various types of agents at a festival. The agents used are “*party animal*”, “*chill guy*”, “*talkative*”, “*food lover*” and “*bad guy*”. All interactions take place in two different modules, namely inside the pub or at a concert. Based on the agents' different personality traits and thresholds, the different agents can decide whether to stay at a particular place and talk to someone or have lunch together . For instance, it may be based on generosity level and tolerance of noise.

2. Approach

Initially, a structure of the project was created. More specifically, different graphical components and a festival area were added along with colors and shapes, in order to create a project skeleton.

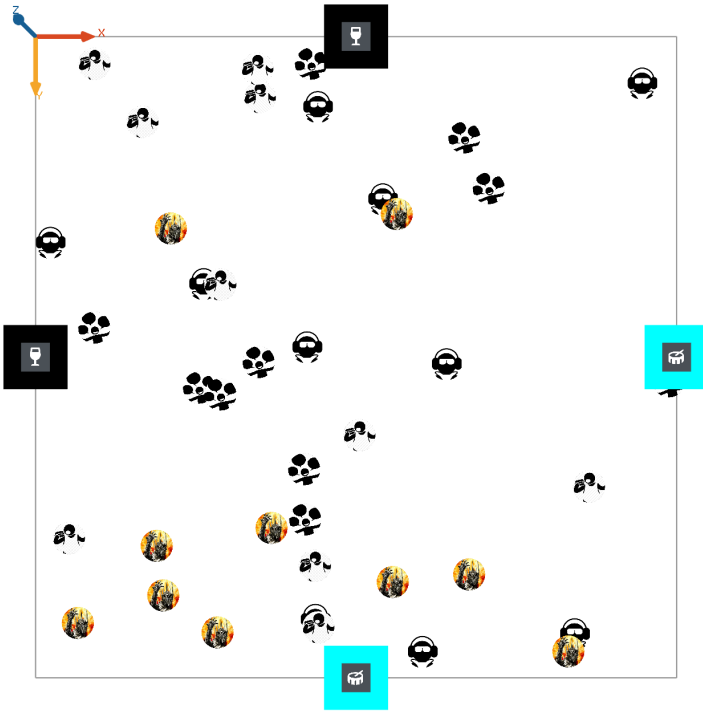


Fig 1. Festival area: different agents (different icons), concert halls (drum icon inside cyan squares) and pubs (wine glass icon inside black squares).

Furthermore, parent agents were created in order to reduce the complexity. The parent agents hold the shared behavior and properties of each agent. For instance, the festival guest parent holds the shared functionality of all festival guests and the module parent agent has the shared communication reflexes for pubs and concerts. Upon initiation of each *specific* agent, the different attribute values (e.g. generosity) are set.

The different ‘personalities’ within the set of festival guest agents were chosen to be somewhat dissimilar to each other in order to make the decisions to interact slightly more

rational and hence provide more dynamic interactions (i.e. not allowing everyone to become friends with each other). For example, a “talkative guy” might not have a high probability of becoming friends with a “chill guy” because he may be looking for someone that is somewhat talkative or a party guy.

There are two types of communication structures: FIPA and communicating within the distance of a module (concert or pub). The former is used when the pub or a concert multicasts/publishes the events of opening or closing, whereas the latter is used for interactions among agents on a local level.

For measuring the number of drink invitations inside the festival we have managed the invitation procedure inside the parent agent which means that drink invitations can be made by all types of agents irrespective of their traits and personalities. “numberOfDrinkInvitations” variable is used to store the invitations and displayed in a graph. For measuring the total number of interactions between different types of guests at a pub or concert made within a simulation, the global variable “numberOfInteractions” is monitored and displayed in a graph. The numberOfInteractions are decreased when a riot is started by a bad guy agent in the festival.

Furthermore, the variable “globalHappiness" is for capturing the global happiness value inside the festival which updates after 10 cycles during the simulation. The global happiness value is calculated on the basis of a generous trait for each type of guest which increases each time when a guest gets a drink invitation. The overall global happiness inside the festival decreases when a riot is started by the bad guy inside one of the modules in the festival. The global happiness value ranges between 0 to 1.

3. Agents

In order to have common methods for all the guest types and the event modules we used the approach of parent class. We have defined two parent classes (agents), FestivalGuests which is parent class for all the guest type agents in our simulation and FestivalModules which is parent class for the event modules which are pub and a concert in our simulation.

3.1 Guests Parent Class (FestivalGuests)

The FestivalGuests class acts as parent class for all different types of festival guests. This class has different attributes, e.g. targetLocation, moveToModule and moveToWandering, alreadyInModule. It also holds the different attributes or personal traits corresponding to each child guest’s personality: generous, noisy, talky, foody and bully all initialized with low values. One attribute is named moduleTimer which is basically just a cycle counter in the simulation and is used to allow each guest to enjoy a certain module for 100 cycles and then move back to wandering in the festival also depending on the probability if a guest wants to move to wandering or to another module.

FestivalGuests class contains the common functions for all the guest agents which is invitation of drink and the acceptance of drink by the guest where the generous trait of the guest is increased by 1 percent when accepting a drink from fellow guests inside a module.

Another common function is to establish communication with the different modules (pubs and concerts) in order to retrieve directions to their locations. In detail, a guest asks randomly for any module direction by sending a request for directions to the FestivalModule. If a guest asks for a pub, it will reply with the location to any of the two pubs, if the pub is opened. If a guest asks for a concert, it will reply with the location if the concert genre matches the guest's prefered music genre.

When the guest has received the location and walked to the pub or concert, it establishes a new communication link with the specific module (a pub or a concert). Now, the communication will be at a local level within the module in order to communicate with other various guests within this certain module and hence allow searching for new friends. After an arbitrary amount of time, the Festival Guests timer will eventually exceed the timer threshold of 100, and if that is the case, it will go back to being wandering in the festival.

3.2 Different Guest Agents (FestivalGuests children)

The child agents implement the parent agent's functionality and behave accordingly. However, their personality will vary upon initiation. Each guest type will have its own personality which means different values for the attributes like generous, noisy, talky, foody and bully. Different types of guest agents used in our simulation are as follows.

3.2.1 PartyAnimal

A PartyAnimal will have a higher probability of being noisy compared to other guest types while the other attributes will be random between 0 and 1.

A Party Animal also has a favorite genre list and upon entering the concert hall it will ask for the genre of the concert and compare it with its own list and if the genre is not according to his taste he will leave the concert.

A party animal when enters the pub it can ask for the guest list inside the pub and will analyze the overall mood of the pub by calculating the average noise of the pub guests, if its noisy enough for the agent they will stay there and try to find a random friend in the pub to have a drink with and enjoy the pub else they will leave if they found the place to be too quiet.

3.2.2 ChillGuy

A ChillGuy will have a higher probability of being generous compared to other guest types while the other attributes will be random between 0 and 1.

Chill Guys have a max level for noise acceptance in a pub or conert. So when they move towards a pub or concert the first thing they will do is to ask for a guest list from the module and analyze the noise level of the module by calculating the average noise level of all the guests inside that particular module. If the place is too noisy, chill guys will leave the place and go back to wandering and if not they will stay there and interact with the guests on a local level which we show in the console.

3.3.3 Talkative

A Talkative guy will have a higher probability of being a talkative person compared to other guest types while the other attributes will be random between 0 and 1.

A talkative guy has a minimum talkative value which forces the agent to stay at the module. The agent upon arriving a module asks for the guest list and calculates the average value for “talky” attribute and it meets the minimum level then the agent looks for a friend with high talky value and after that the agent starts the local communication with the friend and also adds it to his own friend list.

3.3.4 FoodLover

A FoodLover will have a higher probability of being foody compared to other guest types while the other attributes will be random between 0 and 1.

A Food Lover is interested in good food and friends having similar desires. In this agent's behaviour we are also checking if the pub is open or not because for food we thought that it should be mostly available at the pub and not at a concert. Food Lover checks if the pub is open and how is the food quality. Each Food Lover has a hunger level. If the pub is open, food quality is good and the food lover is hungry enough, the agent then looks for a partner or friend so that both can eat together. If the agent finds a friend with a high value of being foody, the agent will communicate locally and invite him for food else the agent will eat alone.

3.3.5 BadGuy

A BadGuy will have a higher probability of being a bully compared to other guest types while the other attributes will be random between 0 and 1.

A Bad Guy has a bully value threshold and when that value is met the Bad Guy starts riots in different modules inside the festival which results in the reduction of the global happiness level of the festival and also the number of interactions between guests are reduced. Bad Guy asks for the pub opening and if the pub is open the agent just enjoys three and if the threshold is met the agent starts the riot by some means inside the module and same goes at the concert hall.

3.3 Modules Parent Class (FestivalModules)

The FestivalModule class acts as parent class for all different types of festival modules which are a pub and a concert hall in our simulation. This class has a list of festival guests in order to keep track of all certain child agents during the runtime. It also has a timer that controls different event occurrences throughout the iterations. The agent waits for requests by festival guests and informs them about locations or guest lists. The former is used to direct child agents to do different modules (either pubs or concerts), and the latter is used for interaction among child agents at a bar or a restaurant. In order to establish communication between agents, it publishes an event which child agents may subscribe to.

The FestivalModules class has some common attributes like typeOfModule, timer and guest list. Some common functions are handling the requests from the guest agents and choosing a random guest for the drink invitation from a fellow guest.

3.4 Different Module Agents (FestivalModules children)

The dissimilarities between the two different child festival modules follows: A *Pub* has a closing and opening period that is communicated to control whether child agents can enter the pub or not. A *ConcertHall* also has a timer, but instead of opening and closing (it is always opened), it will change genre after the time has exceeded the timer threshold in order to affect new child guests. Different types of module agents used in our simulation are as follows.

3.4.1 Pub

A pub agent has some attributes like moduleOpen, pubOpenTimer, pubCloseTimer which are used to simulate the opening and closing of a pub. Pub agent also has a light system which simulates different lighting to show that a pub is actually open now. Pub agent has a method where it informs about the opening or closing of pub along with food quality offered.

3.4.2 Concert Hall

A concert hall agent has a timer which is used to change the genre played in the concert. Concert Hall agent has a method where it informs about the genre played in the concert hall.

4. Experiments and results

4.1 Params and Icons

For the simulation we used 50 guest agents in the festival. We had 10 Party Animal agents, 10 chill guys, 10 food lovers, 10 talkative persons and 10 bad guys in the festival. Two pubs and two concert halls at fixed locations and having a square of (10,10).

We used different shapes or icons for our agents according to their personalities and roles so that they can be distinguished clearly. The icons are as follows

PartyAnimal

ChillGuy

FoodLover

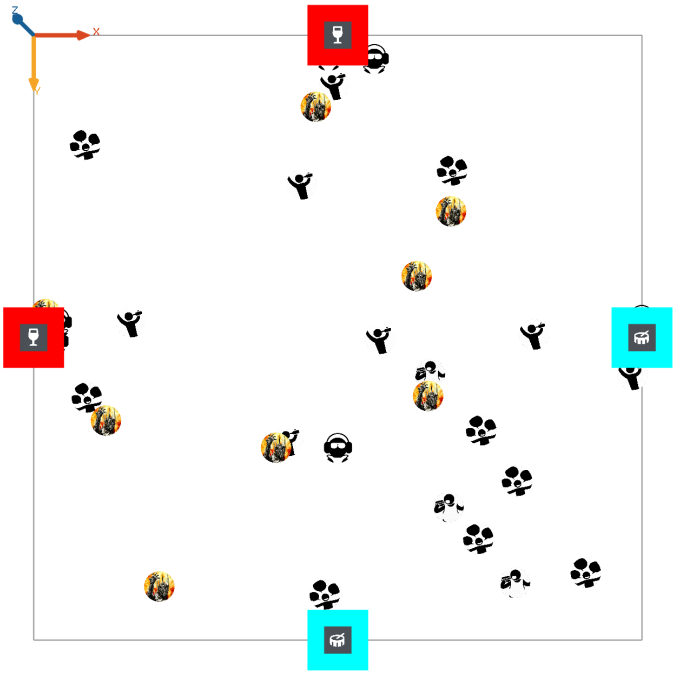
Talkative

BadGuy (Sauron from Lord of the Rings)

Concert Hall Icon

Pub Icon

4.2 Simulation



Interactive console | Console

```
found a foody friend to eat with named FoodLover2
PartyAnimal4 received the music genre Metal
PartyAnimal4 does not like music in ConcertHall0 and left
ChillGuy8 finds it too noisy in Pub0 and left
ChillGuy7 says ConcertHall0 is nice and he can enjoy with all these people
Pub0 opened pub
Pub1 opened pub
ChillGuy6 finds it too noisy in ConcertHall0 and left
found a talky friend named Talkative4
ChillGuy3 finds it too noisy in Pub1 and left
ChillGuy0 increased generous value of ChillGuy7 to 0.42479166462984297
ChillGuy0 says ConcertHall0 is nice and he can enjoy with all these people
ChillGuy5 says Pub1 is nice and he can enjoy with all these people
ChillGuy4 increased generous value of ChillGuy5 to 0.8002230022760808
found a foody friend to eat with named FoodLover2
ChillGuy4 finds it too noisy in Pub1 and left
PartyAnimal1 received the music genre Metal
PartyAnimal1 does not like music in ConcertHall0 and left
ConcertHall0 started concert with genre Metal
ConcertHall1 started concert with genre Pop
FoodLover0 increased generous value of FoodLover1 to 0.9430307338800126
BadGuy4 started riot in ConcertHall1 which results in loss of global happiness
found a foody friend to eat with named FoodLover2
found a foody friend to eat with named FoodLover2
FoodLover8 says food is not good at Pub0 so no reason to stay
found a talky friend named BadGuy1
ChillGuy9 finds it too noisy in ConcertHall0 and left
ChillGuy1 increased generous value of BadGuy1 to 0.9003660714773658
ChillGuy1 finds it too noisy in Pub1 and left
found a friend named PartyAnimal6
FoodLover5 says food is not good at Pub0 so no reason to stay
PartyAnimal5 received the music genre Metal
PartyAnimal5 finds it too quiet in ConcertHall0 and left
FoodLover4 increased generous value of Talkative4 to 0.2995472523895089
found a foody friend to eat with named Talkative4
ChillGuy2 finds it too noisy in Pub0 and left
ChillGuy6 finds it too noisy in ConcertHall0 and left
```

Fig 2. Festival area and Console: different agents (different icons), concert halls (drum icon inside cyan squares) and pubs (wine glass icon inside black squares), Console showing different outputs.

4.3 Results and Graphs

We ran our simulations continuously and got different console and map outputs and guests were acting according to their personalities and we also displayed three different graphs other than the main map to show some features of our festival. The graphs are globalHappinessChart, globalDrinkInvitations and globalFriendlyInteractions.

4.3.1 Global Happiness Chart (Globally managed)

The Global Happiness chart is calculated on the basis of increasing generosity value of each guest over the span of time spent inside the festival during simulation. In our festival each time when a guest becomes generous enough to buy a drink for a fellow guest, the generous attribute for that particular guest increases and hence this results in overall happiness inside the festival that people are being kind to each other and they love to share and have a get together. The max range for global happiness is one as we are calculating on the basis of taking the average generous value of all guests divided by total number of guests.

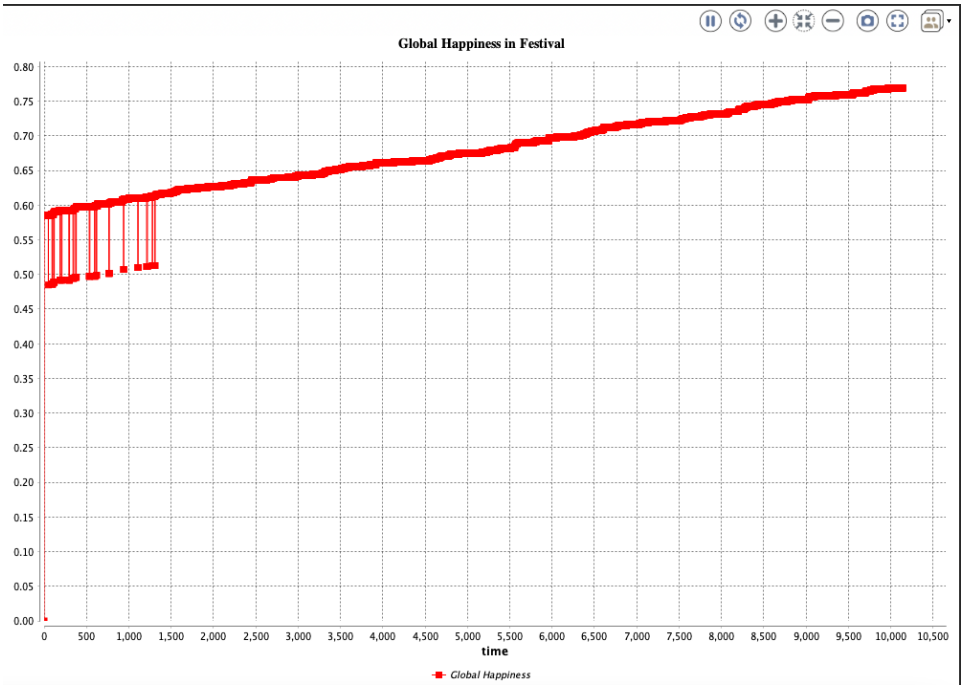


Fig 3. Global Happiness Chart.

In the given graph, spikes which have resulted in decrease of global happiness is because of riots started by the bad guy in the festival but the spirit of other guests does not affect the happiness of the festival and it continues to grow.

4.3.2 Global Drink Invitations

Each time when a guest reaches the generous level to be able to buy a drink for a fellow guest and the possibility of invitation is met, the guest agent requests the pub or concert to invite a fellow guest for a drink and when the fellow guest accepts the request we then increment the global drink invitations variable and on the basis of this the global happiness value is affected.

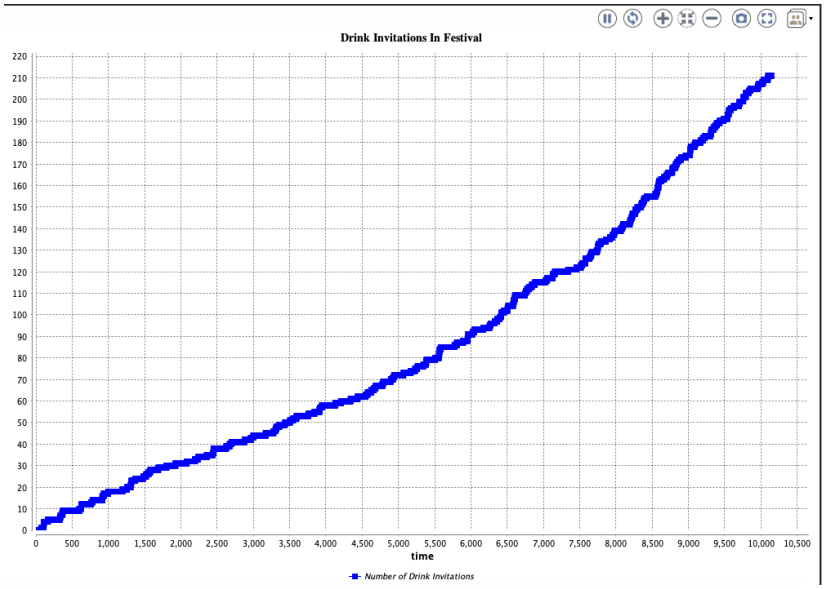


Fig 4. Global Drink Invitations.

4.3.3 Global Friendly Interactions

Each time when a guest agent interacts with another guest through local communication inside one of the modules we count it as global friendly interactions. For some guest agents we store a friend list.

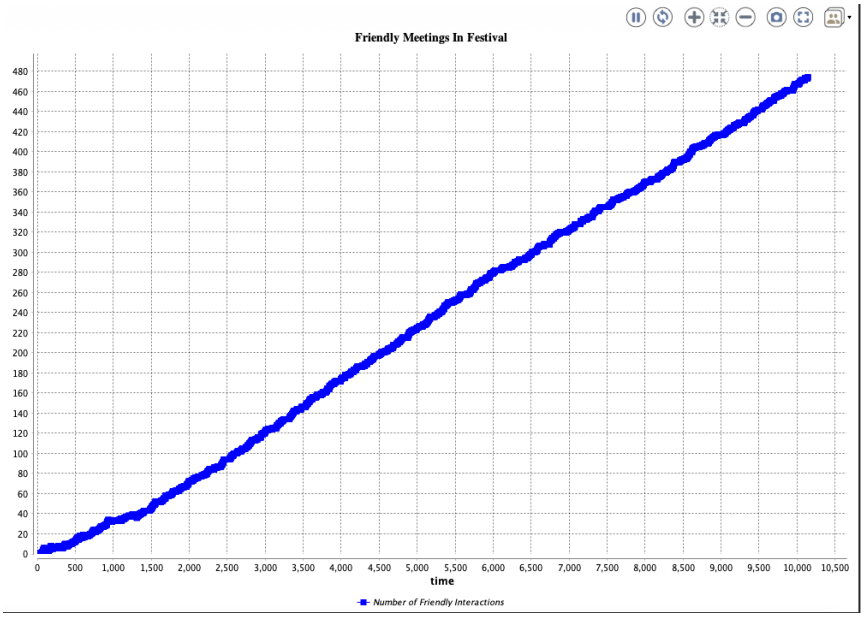


Fig 5. Global Friendly Interactions.

5. Discussion and Conclusion

Overall the project implementation went well because we have been doing the festival related models in the assignments and we decided to go with a festival model for the project also so we had the initial idea how things are working and how we can manage the communication using FIPA between the event modules and guests. The Parent classes made a difference as we managed to have a clean code by putting the common methods and behaviours between different agents in their parent classes. Some things that could even further be improved in the project was to manage the bad guy in the festival by introducing a guard in the festival but we did something similar in one of the assignments so we decided not to introduce that again here. Another improvement could be that instead of doing local communication we could have used FIPA for communication between guests. But overall the project went well as we managed to have some icons to represent different agents more clearly in our simulations.