Project Report

Group 28

Sara Moazez Gharebagh saramg@kth.se

Sebastian Lihammer lihammer@kth.se

31 December 2021

1. Introduction

This report covers the implementation and results of the final project in the course *ID2209 - Distributed Artificial Intelligence and Intelligent Agents*. The main goal with the project is to show different behaviour and interactions between different types of agents. This is done in a festival simulation with different types of guests, having different types of attributes, needs and moods, interacting at different places. A global and interesting value to monitor was chosen, namely the number of happy and unhappy guests. This value changes when different types of guests interact and shows how the guests affect each other's mood. The project is implemented using the GAMA platform and FIPA is used for communication between guests.

2. Approach

The development process began by discussing the basic aspects that needed to be decided before writing any code. Firstly, we decided what kind of environment it should take place in. Then we decided what types of agents, attributes and rules for interaction that could be interesting to include. After that, based on the types of agents we chose, we decided on three different places where the agents could meet and interact. Since we had already implemented a festival simulation with guests in *Assignment 3*, it was an easy choice to have a festival simulation for this project also, since we could reuse the code.

When all the basic aspects were decided, we started to write the code. This was done via pair programming, where one programmed for 20 minutes and the other observed. Then the roles were switched. The code was frequently tested to make sure that new features added worked properly. If errors were encountered, we debugged the code by adding print statements and carefully reviewed the code. After every programming session, the code was reviewed, cleaned and comments were added to make sure that the code was easy to understand.

3. Experiments and Results

3.1. Guests and personalities

The most important species of agents used in the simulation is the *Guest* species, which represents festival guests moving around and interacting at the festival. Much of the basic

functionality of the *Guest* species, such as moving around and FIPA messaging, was created through code from the previous assignments.

Every guest is upon initialization assigned a personality type (“Party”, “Neutral” or “Chill”) and a diet type (“Vegan” or “Meat-eater”). The diet of a guest can be distinguished visually by the color of their head (“Vegan” guests have a light green head and “Meat-eater” guests have a black head) and body (“Party” guests have pink body, “Neutral” guests have a lemon body and “Chill” guests have a light blue body). Because personality types and diet types are assigned at random, there are technically six different types of guests in the simulation:

Diet	Vegan	Vegan	Vegan	Meat-eater	Meat-eater	Meat-eater
Personality	Party	Neutral	Chill	Party	Neutral	Chill

In addition to the core personality and diet types, each guest also has three different personality attributes: *kind*, *chatty* and *generous*, assigned a random value upon initialization. “Party” guests have a higher chance to get higher values on the *chatty* attribute. Also affecting the behavior of guests are the attributes *happiness*, which is initialized to 0.5 but changes value throughout the simulation depending on what happens to the guest, and *isAngry*, which determines if the guest is so unhappy that they have become angry (when their happiness reaches 0.0).

3.2. Locations

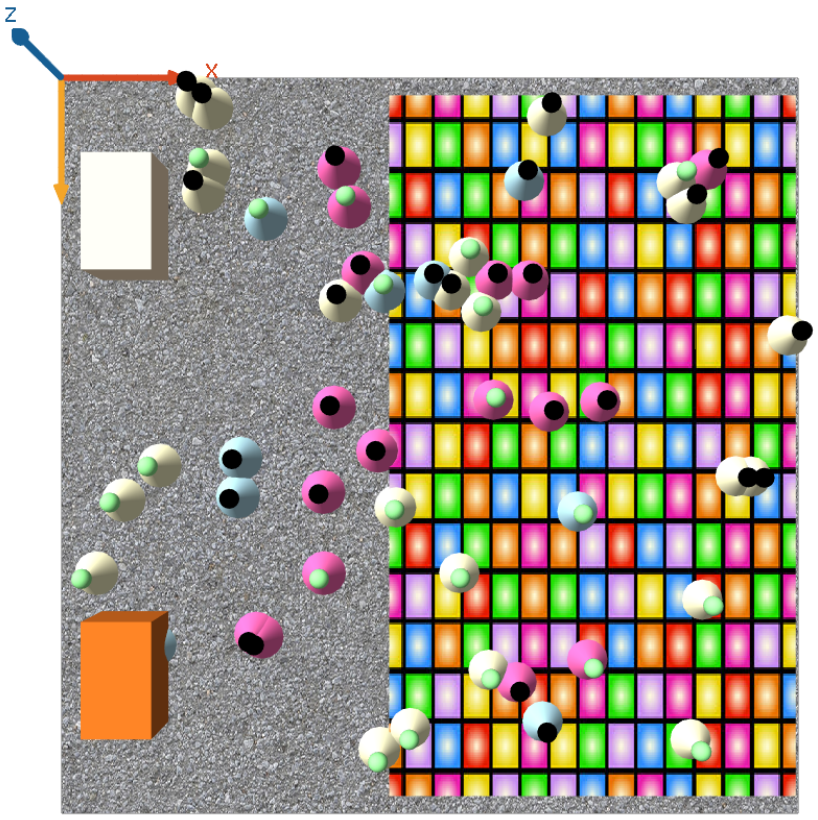


Figure 1: Screenshot of the simulation. The beige box in the top-left is the food stand and the orange box in the bottom-left is the bar.

The map used in the simulation includes three different locations; a large *dance floor*, a *food stand* and a *bar*. Guests visit these locations either when they need to (for instance, if a guest gets thirsty they will go to the bar) or as part of an interaction (for instance, a guest can ask another guest to dance).

3.3. Interactions

The simulation includes the following ten different types of interactions:

- **Scolding**: Several guests, if angry, can respond to different interactions by scolding the guest interacting with them. The happiness of the scolded guest will decrease by 0.3. The guest that is scolding will stop being angry and get a slight increase in happiness to 0.2.
- **Making noise**: “Party” and “Neutral” guests have an unique attribute called “makeNoise” which randomly increases. Once it reaches 1.0 or higher, and the guest is at the bar or food stand they will make noise. Nearby guests will react differently depending on their personality type; “Chill” guests will not like the noise and “Party”

guests will like the noise. “Neutral” guests will like the noise only if their happiness is equal to or greater than 0.5. The happiness of guests who do not like the noise is decreased by 0.3 and the happiness of guests who like the noise is increased by 0.1. If a guest is angry, they will scold the noise-maker.

- **Buying meat/vegan food**: When buying food at the food stand, “Vegan” guests will buy a vegan option while “Meat-eater” guests will buy something with meat. Nearby guests react differently depending on their diet type and their personality attributes. Vegans who are angry and unkind will scold someone for buying meat but vegans who are happy and chatty will simply inform the meat-eater about veganism and animal rights. Meat-eaters will not care about vegans buying vegan food, but the happiness of vegans who witness it will increase by 0.1.
- **Inform about veganism**: Chatty and happy vegans (i.e. both values above 0.5) who witness a meat-eater buying meat at the food stand will try to inform the meat-eater about veganism and animal rights. The informer’s happiness will increase by 0.1. If the person being informed is happy, they will consider going vegan (increasing their happiness by 0.1), otherwise they will simply be bothered (decreasing their happiness by 0.1).
- **Drink beer and chill**: “Chill” guests who are happy and at the bar will offer a nearby guest to drink some beer and chill with them. “Chill” guests, as well as happy “Neutral” guests, will accept whereas “Party” guests will not. Angry guests will scold the asking guest for asking. Guests who agree to having a beer and chilling will have their thirst reset to 0.0 and the happiness of both guests will increase by 0.2.
- **Offering to buy drink**: Generous and happy guests will when at the bar offer to buy a drink for a nearby guest. Happy guests will accept the offer, with the happiness of both the receiver and the buyer increasing with 0.1. Unhappy guests will simply decline the offer, decreasing the happiness of the guest making the offer by 0.2, and angry guests will scold the guest making the offer for asking.
- **Grab a snack**: “Neutral” and chatty guests who reach a location where there are people nearby ask a nearby person if they want to grab a snack with them. The guest receiving the offer will accept either if they are happy or if they are hungry (hunger below 0.3). The guests will then both move to the food stand and their happinesses will increase by 0.1. Angry guests will refuse the offer and scold the guest who asked them.

- **Chatting**: A “Neutral” or “Chill” guest who is chatty and happy will, after reaching a location where there are other people nearby, talk to the nearest person. The happiness of the person who gets talked to will increase by 0.1.
- **Ask to dance**: “Party” guests who are chatty and happy will instead of simply chatting to the nearest person ask them to dance. Other “Party” guests, as well as “Neutral” guests whose happiness is above 0.5 will accept to go dancing, upon which both the guests will move to the same spot on the dance floor and their happinesses will increase by 0.2. A “Chill” or unhappy guest will not accept to go dancing. A guest who is angry will scold the guest asking them to dance.
- **Ignore**: In certain situations, some requests or offers from guests will be ignored by other guests, which will decrease the happiness of the asking or offering guest by 0.2.

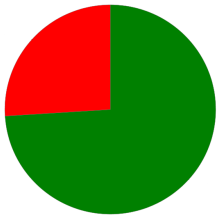
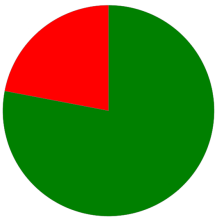
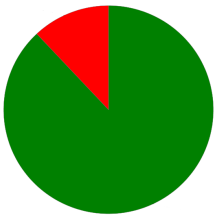
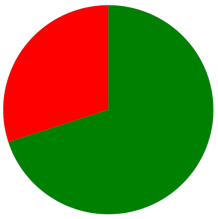
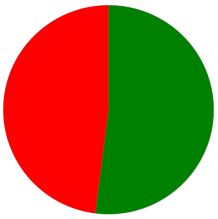
3.4. Results

3.4.1. Default values

The global and interesting value chosen to measure and observe during the experiment was the number of happy (happiness greater than or equal to 0.5) and unhappy (happiness below 0.5) people at any one given time. The happinesses of all the guests was measured through a species called *happinessMeasurer*. The *happinessMeasurer* loops over a list of all the guests every 100 cycles and counts the number of happy and unhappy guests. Its measurements are then displayed in a chart in a separate tab in GAMA.

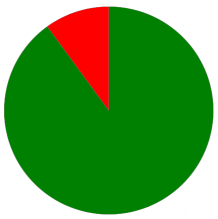
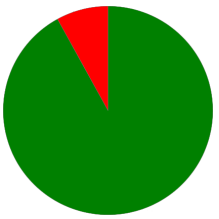
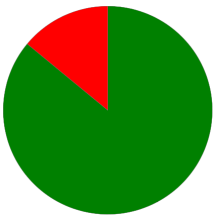
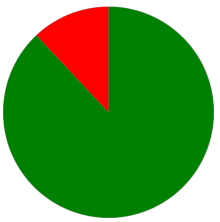
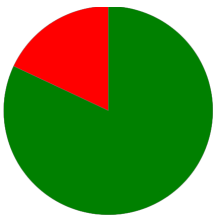
Given that the simulation runs continuously, and happinesses changes slightly randomly, there is no measurement for how many people were happy/unhappy at any point in the experiment and for how long and every run of the simulation gives different results. In general the number of unhappy people is around 10, sometimes going as low as around 5, and sometimes as high as around 20. Over longer periods of time, the number can fluctuate more dramatically.

The chart generated by the simulation provides a series of snapshots throughout, some of which are included in the table below:

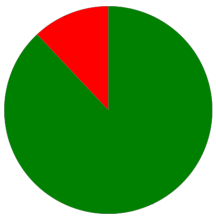
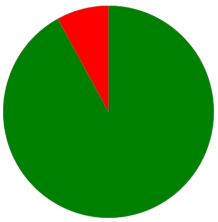
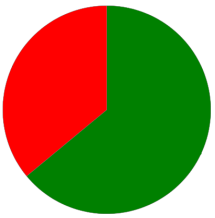
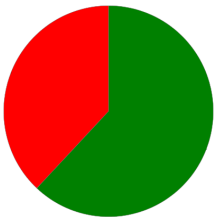
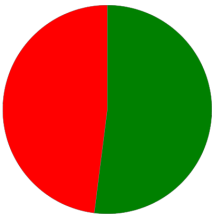
 200 cycles	 1000 cycles	 2000 cycles	 5000 cycles	 10000 cycles
37 guests (74 %)	39 guests (78 %)	44 guests (88 %)	35 guests (70 %)	26 guests (52 %)
13 guests (26 %)	11 guests (22 %)	6 guests (12 %)	15 guests (30 %)	24 guests (48 %)

3.4.2. Changing the values

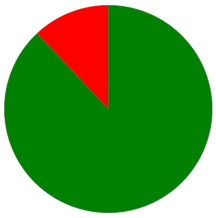
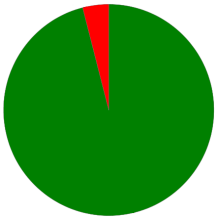
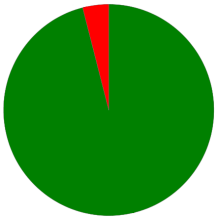
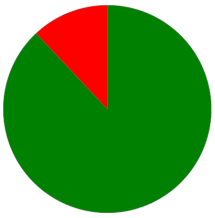
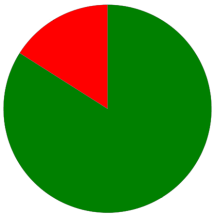
Because the majority of the interactions implemented in the simulation are relatively common, and many of the responses depend on the level of happiness of the guests, tweaking the effects of just a single type of interaction can have dramatic effects. For instance, removing the happiness decrease a guest gets after being scolded significantly increases the number of happy guests and makes the number fluctuate less dramatically:

 200 cycles	 1000 cycles	 2000 cycles	 5000 cycles	 10000 cycles
45 guests (90 %)	46 guests (92 %)	43 guests (86 %)	44 guests (88 %)	41 guests (82 %)
5 guests (10 %)	4 guests (8 %)	7 guests (14 %)	6 guests (12 %)	9 guests (18 %)

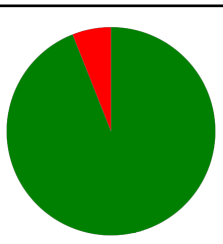
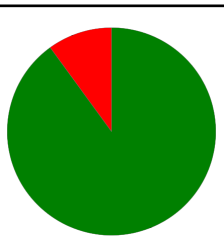
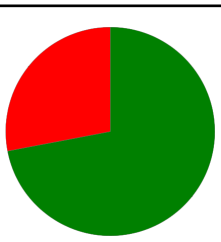
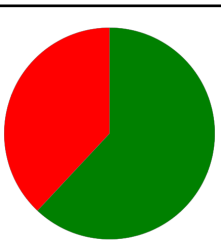
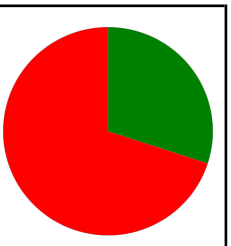
If the effects of scolding were instead increased by just 0.2, from -0.3 to -0.5, the number of unhappy guests instead increases dramatically sooner than previously and remains consistently high from about 2000 cycles onwards:

 200 cycles	 1000 cycles	 2000 cycles	 5000 cycles	 10000 cycles
44 guests (88 %)	46 guests (92 %)	32 guests (64 %)	31 guests (62 %)	26 guests (52 %)
6 guests (12 %)	4 guests (8 %)	18 guests (36 %)	19 guests (38 %)	24 guests (48 %)

If all changes of happiness that occur due to receiving a FIPA message are changed by 0.2 (for instance, changing + 0.1 to + 0.3 and - 0.2 to - 0.4), the result is that the number of happy guests is much greater:

 200 cycles	 1000 cycles	 2000 cycles	 5000 cycles	 10000 cycles
44 guests (88 %)	48 guests (96 %)	48 guests (96 %)	44 guests (88 %)	42 guests (84 %)
6 guests (12 %)	2 guests (4 %)	2 guests (4 %)	6 guests (12 %)	8 guests (16 %)

More precise changes can greatly increase the trend of guests becoming more unhappy over time. The table below shows snapshots of running the simulation with the effects of scolding on happiness increased from - 0.3 to - 0.7 and the effect of not liking a guest making noise also increased from - 0.3 to - 0.7:

 200 cycles	 1000 cycles	 2000 cycles	 5000 cycles	 10000 cycles

47 guests (94 %)	45 guests (90 %)	36 guests (72 %)	31 guests (62 %)	15 guests (30 %)
3 guests (6 %)	5 guests (10 %)	14 guests (28 %)	19 guests (38 %)	35 guests (70 %)

4. Discussion and Conclusions

With the default values, it can be seen in the results that the number of happy guests is much greater than the number of unhappy guests in the beginning, but there is a general trend of more guests being unhappy after the simulation has run for some time, after 7000 cycles or so, though the number of unhappy guests rarely reaches more than 50 %. This is probably because the attributes that affect the guests' interactions (hunger, thirst, make noise etc.) increase with a very small random number and because the happiness value increases/decreases with a very small number. Also, when guests get unhappy, they influence other guests to get unhappy. This is because they might, for example, scold, ignore or decline a drink offer which in turn will decrease the other guests happiness. These negative behaviors only begin to appear when guests already are unhappy, which happens later in the simulation. The more unhappy a guest gets, the more this affects other guests (for instance, only guests who are angry scold other guests). It can be seen, in section 3.4.2, that when we decrease the effect that scolding has on the happiness of the scolded guest, the number of happy guests increases significantly and when we increase the effect, the number of unhappy guests increases significantly.

In this project, we only experimented on the total number of happy and unhappy guests. What could have been interesting to also look at is how the different types of guests' happiness is affected and what types of guests are the ones that are unhappy. For example, are party people more often happy than chill people at a festival?

In real life situations, people can read their surroundings quite well and, for example, choose who they should ask to dance or talk to. If a person looks like they do not want to be disturbed, many people would probably stay away from them. Thus, after these experiments, we can not really make any conclusions about how different people with different personalities affect each other at a festival since this would have required more detailed measurements.