ID2209 – Distributed Artificial Intelligence and Intelligent Agents

# Assignment 1 – GAMA & Agents

*Group 17*
Nagasudeep Vemula
Ayushi Shah

12/11/2019

The purpose of this assignment is to get an introduction to the GAMA language and get a hands on experience of the same by implementing a simple simulation which involves configuration of the agents. In this assignment we simulate a Festival experience which consists of guests, an information centre and some stores for food and drinks. We get introduced to different types of agents and in turn learning about their movements and behaviours.

## How to run

Run Gama 1.8 and import Festival.gaml,Festival_Challenge.gaml and Festival_Creative.gaml as new projects. Press main to run the simulation for the same and use the slider to vary the speed of the simulation.

## Agents

### Guests

The most essential part of this simulation ,these agents are representatives of guests at a festival who are enjoying the music or generally just having a good time. Most of the time they may be idle/wandering .However each guest has the attributes food, thirst which is, they may get hungry or thirsty and then search for food and drink stores for the same. To handle this the agents information centre and shops have been provided.

### Information Centre

This agent holds the location of the various stores for food and drink and it is used to convey the same to the guests who don't know where to go. Guest agents by default have the location of the information centre building and from here they are directed to stores.

### Shops

Each guest has the traits Hunger and Thirst ,which get drained as the festival wears on. To replenish their hunger and thirst they must go to the shops which sell food and drink. These agents are of two types, that is either a shop for buying drinks or a shop for buying food. Guests communicate with the information centre for location of these agents.

## Implementation

Initial development began with the guest agents where we programmed their behaviour for moving around and defined the different reflexes they could have for all the cases. Moving skill is defined by GAMA as the minimal set of behaviours required for agents that are able to move on different topologies. Reflexes are defined for the guest agent for the various scenarios that can arise. These are briefly explained as follow

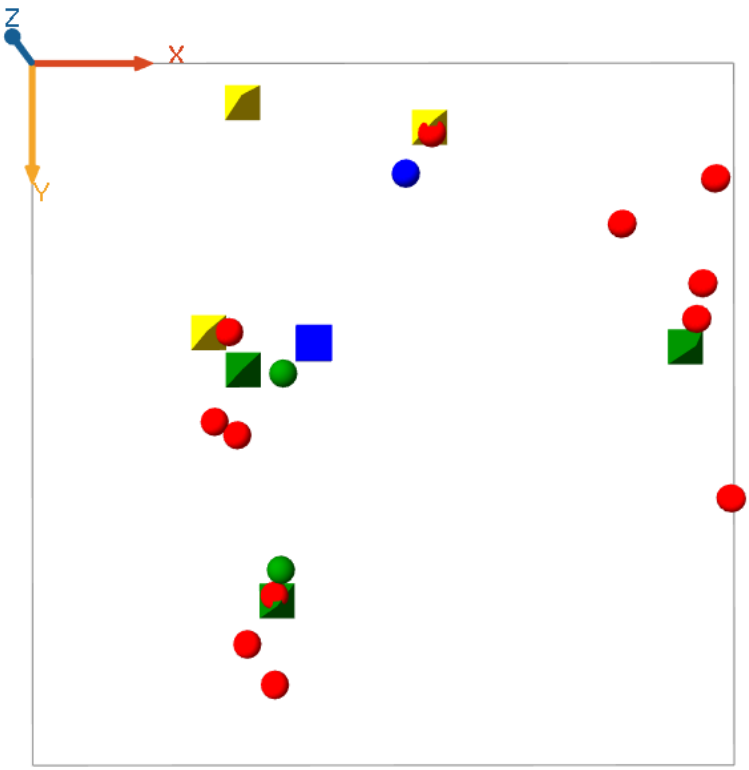
- reflex beFestive : In this case the agent has no target location to move towards and they are generally just moving around the festival and wandering(do wander).
- reflex thirstyHungry-Used to simulate hunger and thirst for the guests.
- reflex moveToTarget -Guest gets a target allocated and has to move towards the same.
- reflex reachInfoCenter- Guest has to move towards the information centre.
- reflex isThisAStore- For identifying if it is a food or drink store.

## Results

The festival simulation runs as expected ,some of the snapshots from the log and simulation are displayed below:-
In the below log it is seen that first guest5 heads to the infoCenter on getting hungry and then gets food from the foodstore, Similarly for guest 13 feeling thirsty.

```
guest5 is hungry, heading to infoCenterSp0
guest10 is thirsty, heading to infoCenterSp0
guest2 is thirsty, heading to infoCenterSp0
guest4 is thirsty, heading to infoCenterSp0
guest9 is hungry, heading to infoCenterSp0
guest3 is hungry, heading to infoCenterSp0
guest14 is hungry, heading to infoCenterSp0
guest7 is thirsty, heading to infoCenterSp0
guest8 is thirsty, heading to infoCenterSp0
guest13 is thirsty, heading to infoCenterSp0
guest12 is hungry, heading to infoCenterSp0
guest11 is hungry, heading to infoCenterSp0
guest8 getting drink at drinkStore2
guest3 getting drink at drinkStore1
guest5 getting food at foodStore0
guest13 getting drink at drinkStore2
```

Furthermore, below is the screenshot of the simulation map as seen in GAMA for the given question



## Challenge 1

In order to implement a memory in the agents such that they remember the places they have visited we implement a list of buildings called guestbrain. Guestbrain is an array which stores the information if the guest has visited a foodstore or drinkstore and location of that store is saved in their memory. For the case where guests are asking guests in the vicinity the statement ask guest_distance() is used. This is just for searching for the guests in the vicinity being the range mentioned.

To check the difference in the time taken for checking with info centre and time taken with checking with agent we have used an attribute gain whose value is always negative.

## Challenge 2:The Bad Apples

This is the scenario where some guests that misbehave must be removed from the festival. This is primarily enabled by creation of a new agent namely the security guard. In order to do this,initially the identification of these bad agents must take place and this is done by the information centre,which does this with the help of the reflex checkForBadGuest. Following this identification of the misbehaving agents, the reflexes catchBadGuest and badGuestCaught are used by the security agent along with the die functionality in order to remove these notorious agents.

```
reflex checkForBadGuest
{
    ask guest at_distance infoCenterSize
    {
        if(self.isBad)
        {
            guest badGuest <- self;
            ask Security
            {
                if(!(self.targets contains badGuest))
                {
                    self.targets <+ badGuest;
                    write 'InfoCenter found a bad guest (' + badGuest.name + '), sending Security after it';
                }
            }
        }
    }
}
```

The above screenshot shows the reflex for the information centre.

```
reflex catchBadGuest when: length(targets) > 0
{
    //this is needed in case the guest leaves before security catches them
    if(dead(targets[0]))
    {
        targets >- first(targets);
    }
    else
    {
        do goto target:(targets[0].location) speed: securitySpeed;
    }
}

reflex badGuestCaught when: length(targets) > 0 and !dead(targets[0]) and location distance_to(targets[0].location) < 0.2
{
    ask targets[0]
    {
        write name + ': exterminated by Robocop!';
        do die;
    }
    targets >- first(targets);
}
```

Above screenshot deals with the reflexes needed for the security to manage the duties.

## Conclusions

The assignment gave us a good introductory exposure to the gama platform and helped us understand how working with agents really takes place in a simulation when it comes to describing their attributes and reflexes which are crucial foundational concepts moving forward. It also gave us some insights on how simulations can be conducted.