MULTI-AGENTS SYSTEM PROJECT
COVID19 SIMULATION

I. PRESENTATION

The aim of this project is to define a multi-agents system to simulate the COVID19 pandemic.

The agents will move on a grid and infect each other depending on multiple parameters (neighbourhood, probability of infection). They will have several status : susceptible, infected with symptoms, infected without symptoms, recovered or deceased.

For this purpose, I have used the Repast Simphony agent-based modeling platform.

II. ARCHITECTURE

The first step constisted in creating the different classes needed in this project.
I defined an abstract class called *Agent* that is the base class for all the agents.
Then, I defined several implementations of this abstract class :

- one for the susceptible agents
- one for the deceased agents

I also created an abstract class for the infected agents that inherits from class *Agent* and add specific attributes/methods for infected agent.
This class has 2 children : one for the infected agents with symptoms and one for the infected agents without symptoms.

The figure 1 shows the class diagram used for the agents implementation. The arrows represent inheritance.

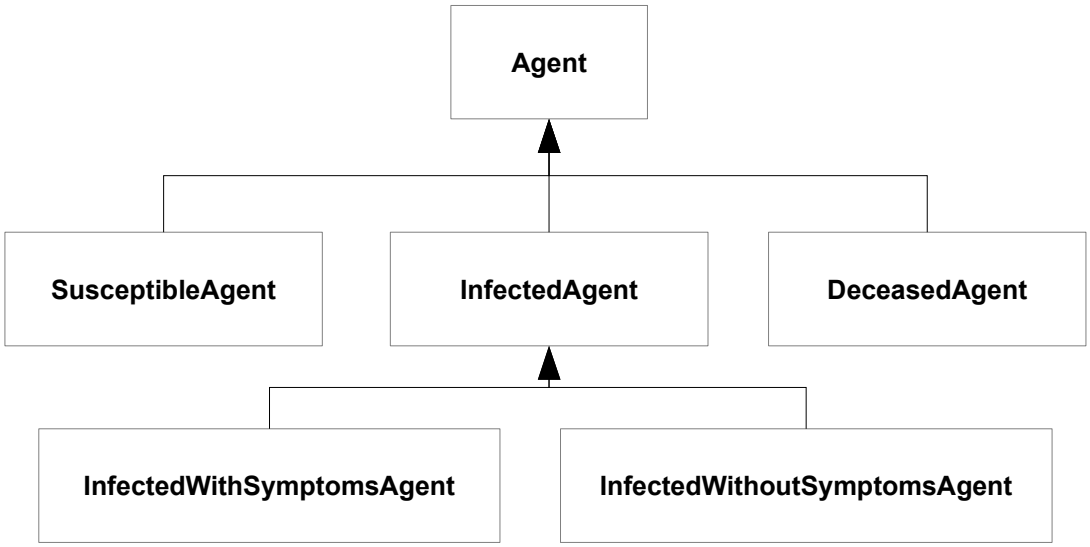


Figure 1 : class diagram of agents

I have also defined a class to handle the context, that implements *ContextBuilder* interface.

The agent class has mainly 2 methods :

- *computeNextStatus* that computes the next status according to the neighbourhood and the current status of the agent;
- *computeNextPositionAndApply* that computes the next position depending on the goal.

Those methods have to be “ordered”. Indeed, we want to compute the next status according to the current grid for each agent. Then, we want to move the agents.

For this purpose, I have used the *@scheduleMethod* annotation to specify a priority between those 2 methods.
The method *computeNextStatus* must be called before the method *computeNextPositionAndApply*, thus, the priority of *computeNextStatus* equals to 2 that is higher than the priority of *computeNextPositionAndApply* (priority = 1).

III. STATUS OF THE AGENTS

An agent can have different status (each status corresponds to a specific child of the abstract class *Agent*):

- He can be susceptible (*SusceptibleAgent* class) : this type of agent has to check at each step if there are infected agents in his neighbourhood. If yes, he will “become” an infected agent with a certain probability.
- He can be infected, either with or without symptoms (*InfectedWithSymptomsAgent* or *InfectedWithoutSymptomsAgent*). For infected agents with symptoms, the duration of the infection has been fixed to 14 steps. For infected agent without symptoms, this time has been fixed to 7 steps. At the end of this time, the agent becomes either recovered or deceased according to a certain probability.
- He can be deceased (*DeceasedAgent* class). With the aim of better understanding the events, he will be displayed during 3 steps and then be removed from the context and disappear from the grid.
- He can be recovered (*RecoveredAgent* class). Then, he can not any more be infected thanks to his antibodies.

When the next status of an agent is different from his current status, the following operations are done each time :
- remove the current agent from the context;
- create a new agent with the next status (using the corresponding class);
- set its location on the grid to its new position.

IV. PARAMETERS

I. Overview

I have defined several parameters to simulate different scenarios. The following table presents an overview of those parameters.

Name	Description	Possible values	Default value
Grid height	Allows to set the height of the grid	Integer	50
Grid width	Allows to set the width of the grid	Integer	50
Movement scenario	Selection of the movement scenario: - either random : agents move in a random way; - or attractive places : agents move towards attractive places	Random scenario Attractive places	Random scenario
Number of infected agents	Sets the number of infected agents at the initialization. The created agents are symptomatic (*InfectedWithSymptomsAgent*).	Integer	1
Number of susceptible agents	Sets the number of susceptible agents at the initialization.	Integer	400
Probability of being docile (wear the mask if mandatory)	This probability is used at the creation of the population of agents. If the mask strategy is activated, the created agent will wear the mask depending on this probability.	Float (between 0 and 1)	1

Name	Description	Possible values	Default value
Probability of being elderly	This probability is used at the creation of the population of agents. The age of the agent will be calculated according to this probability (we consider elderly from 65 years).	Float (between 0 and 1)	0,5
Probability of being healthy	This probability is used at the creation of the population of agents. The created agent will be healthy depending on this probability; otherwise, he will have medical conditions that increase his risk of being infected with symptoms or of dying.	Float (between 0 and 1)	0,7
Probability of infection	This probability is the average probability that an infected agent with symptoms infects a susceptible agent.	Float (between 0 and 1)	0,8
Probability of recovering	This probability is the average probability that an infected agent with symptoms recovers.	Float (between 0 and 1)	0,8
Strategy 1 for pandemic limitation	This parameter allows to specify which strategy must be applied to limit the pandemic. The distanciation, the face mask, the lockdown, the curfew and the isolation of infected people can be activated.	None Distanciation Face_mask Lockdown Curfew Isolation of infected people	None
Strategy 2 for pandemic limitation	This parameter allows to specify which strategy must be applied to limit the pandemic. The distanciation, the face mask, the lockdown, the curfew and the isolation of infected people can be activated.	None Distanciation Face_mask Lockdown Curfew Isolation of infected people	None
Strategy 3 for pandemic limitation	This parameter allows to specify which strategy must be applied to limit the pandemic. The distanciation, the face mask, the lockdown, the curfew and the isolation of infected people can be activated.	None Distanciation Face_mask Lockdown Curfew Isolation of infected people	None

II. Movement scenario

If the movement scenario choosen by the user is “Random_scenario”, then the agents moves randomly. At each step, they can move on a free cell of the neighbourhood. The neighbourhood is defined by the 8 cells around the cell where the agent is located.

This number is reduced if the agent is located on the border of the grid. We assume that the grid is a real location and thus is “finite”. The agents can not go outside of the grid or jump from top to bottom for example.

If the movement scenario choosen by the user is “Attractive_places”, then when the agents are created, we assigned them a goal : either school or shopping center.
In our “small world”, the school is located in the left-bottom corner and the shopping center is located in the right-top corner.
At the initialization of the simulation, the susceptible agents are created with one of those 2 goals with an equivalent probability.

When a susceptible agent “becomes” infected, the infected agent that is created keeps the same goal as the susceptible agent, except if the isolation of infected people is activated.
All the recovered agents moves randomly.

III. Number of infected agents

The user can specify the number of infected agents at the beginning of the simulation.
During the initialization of the simulation, the created agents are infected with symptoms and located randomly on the grid.

IV. Number of susceptible agents

The user can specify the number of susceptible agents at the beginning of the simulation.
During the initialization of the simulation, the susceptible agents are located randomly on the grid.

V. Probability of being docile

This probability is used only if the strategy of the mask is activated.
Indeed, when we create the agents at the initialization, we make them wear a mask according to this probability.
If probability equals 1, then all the agents of the simulaton wear the mask.
If this probability equals 0, then no agents wear the mask.

VI. Probability of being elderly

This probability allows to specify if the population is rather young or old.
At the initialization of the simulation, the agents are created and given an age. This age is computed according to this probability.
We consider as ederly, people who are older than 65 years. Elderly people are more likely to declare the Covid in a serious way.

VII. Probability of being healthy

This probability allows to specify if the agents have medical conditions that increase their risk to have the Covid with symptoms and to die.
At the initialization of the simulation, the agents are created and we specify if they are healthy according to this probability.

VIII. Probability of infection

We assume this probability to correspond to the average probability of infecting somebody in the neighbourhood when we are an infected agent with symptoms.
Thus, this probability must be refined depending on the type of neighbourhood. For example, for agent without symptoms, this probability must be decreased. See “Contamination” paragraph.

IX. Probability of recovering

This probability is used to decide if an infected agent should recover or die.
We assume that only people infected with symptoms can die.
Thus, for infected agents without symptoms, this probability is not used (all the agents recover).
Infected agents with symptoms will recover according to this probability.

X. Limitation strategies

The user can choose to apply 0, 1, 2 or 3 strategies of limitation. By default no strategy is applied (the 3 parameters are set to None).

1) Distanciation

If this strategy is chosen, then the agents give priority to cells that have the fewest neighbours when they have to choose their next position. They avoid the “contact”.

2) Face mask

When the agents are created, we specify if they wear a mask according to the probability of being docile. If they wear a mask, then the computation of the real probability of infection is decreased.

If an infected agent wears a mask and is in the neighbourhood of a susceptible agent, then the probability of being infected for the susceptible agent is multiplied by a factor less than 1, fixed at 0.3. The probability of infection decreases if the infected agent wears a mask.

If a susceptible agent wears a mask, then the probability of being infected by a neighbour is also multiplied by a factor less than 1, fixed at 0.3. This way, the probability is also decreased.

3) Lockdown

If this strategy is chosen, then we simulate a permanent lockdown. To simplify, the agents are created, located randomly on the grid and are “frozen” : they don't change their position during the simulation. This way, only the neighbours of the initial step can infect a susceptible agent. This can be seen as the contamination between family members that live together.

4) Curfew

If this strategy is activated, then we simulate a curfew. To simplify, agents are frozen during the curfew. We simulate a curfew from 10PM to 8AM by freezing the agents if the current tick modulo 24 is greater than 22 and less than 8.
This has been fixed in the source code to avoid additional parameters.

5) Isolation of infected people

As soon as an agent becomes infected, his goal is set to “hospital”. Thus, he will move towards the hospital during the time of his infection. The hospital has been located at the center of the grid.

XI. AGENTS MOVEMENTS

Depending on their goal, the agents will move either randomly or towards a specific point. An agent can move on the cells of its direct neighbourhood if they are free (we do not authorize 2 agents to be on the same cell for better visualization purpose).

For example, the agent A can move only on the yellow cells.

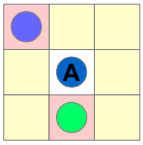


Figure 2 : neighbourhood of an agent

If the agent is on the border of the grid, then the number of possible cells will be reduced : indeed, the agents can not go outside the grid, and they don't “jump” from the right to the left of the grid for example.

As explained previously, the “random” or “towards” movement is decided according to the goal of the agent. Some strategies can also influence the choice of one cell among free cells of neighbourhood (distanciation for example).

XII. CONTAMINATION AND RECOVERING

I. Probability of being infected depending on the neighbourhood

I choose to impact an agent depending on his neighbourhood. 2 agents can not be on the same cell. This allows to have a better visualization when the pandemic spreads.
Thus, a susceptible agent can be infected, if there is one or more infected agents in his neighbourhood. On the figure 3, the neighbourhood is represented by the light blue cells. The susceptible agent is the blue circle; it has 2 infected agents in its neighbourhood : 1 has symptoms (the red circle), 1 is asymptomatic (the orange circle).

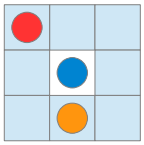


Figure 3 : neighbourhood of a susceptible agent

Of course, several parameters need to be taken into account to compute more precisely the probability of being infected.

The *probInf* parameter corresponds to the mean probability for one infected agent to infect a susceptible agent without mask.

But it must be weighted by the following parameters :

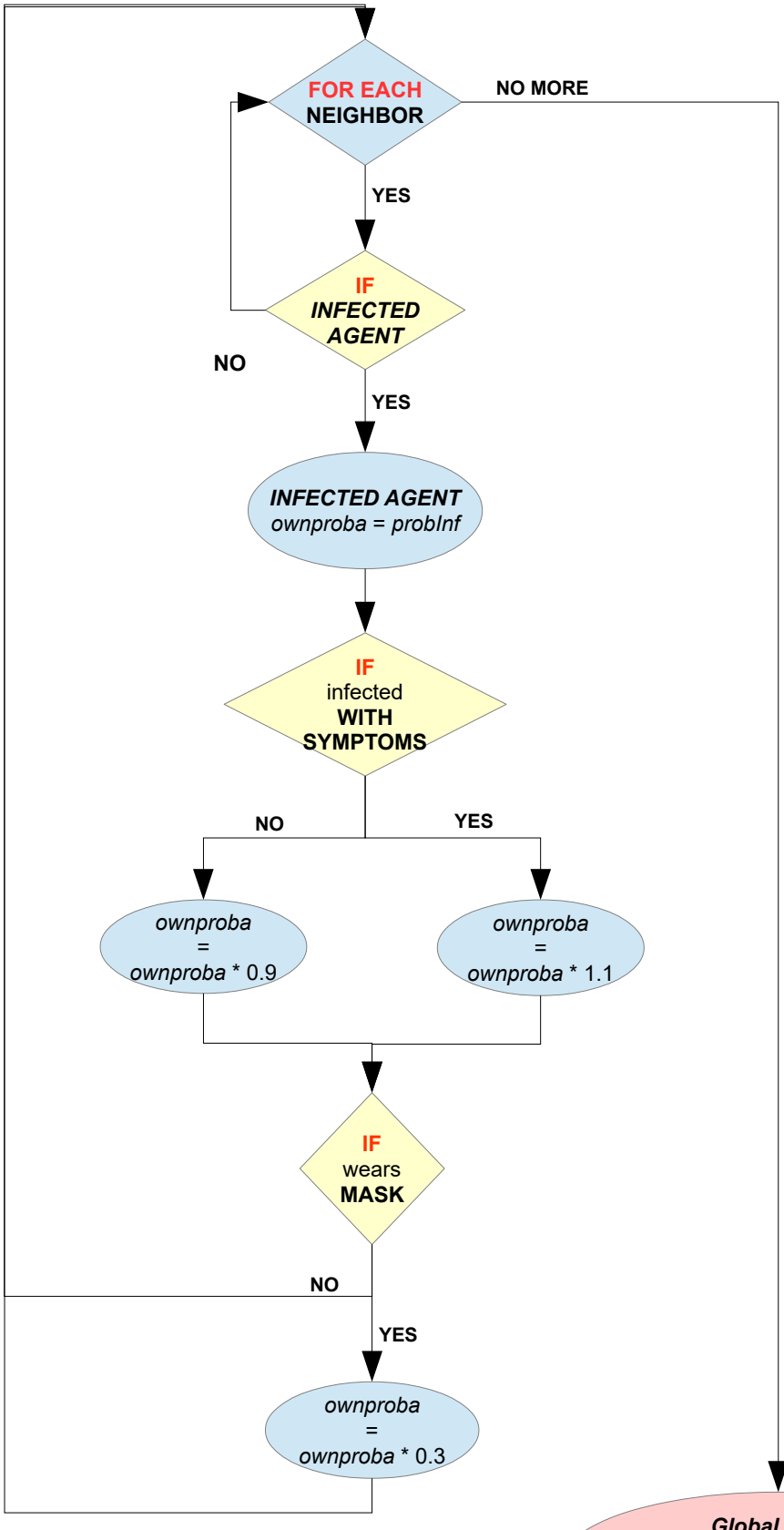
- the type of infected agents : an agent with symptoms is more likely to infect the susceptible agent ;
- does an infected agent in the neighbourhood wear the mask ? If yes, the probability must decrease ;
- how many infected agents are in the neighbourhood ?
- does the susceptible agent wear a mask ? If yes, it must decrease the probability of being infected.

Here is the choices I have done to compute the most realistic probability:

- contamination probability of one infected agent WITH symptoms = *probInf* * 1.1
- contamination probability of one infected agent WITHOUT symptoms = *probInf* * 0.9
- if an infected agent wears a mask, then the contamination probability is multiplied by 0.3.
- if there are several neighbors, I compute the average of all their probability.
- the probability must also increase depending on the number of neighbors. The more there are neighbors, the more the susceptible agent can be infected. Thus, I multiply the probability average by (1 + (N-1))*0.05.
  - For a neighbourhood with 1 agent, the probability is not changed (factor = 1).
  - For a neighbourhood with 2 agents, the probability is multiplied by 1.05.
  - For a neighbourhood with 2 agents, the probability is multiplied by 1.1.
  - For a neighbourhood with 3 agents, the probability is multiplied by 1.15.

I could have add the possibility to set all those factors by the user, but it would have complexified the userinterface with too much parameters ! Thus, I decided to fix them in the source code.
In addition, if the susceptible agent wears a mask, then the global contamination probability is multiplied by 0.3.

The figure 4 corresponds to the chart representing this flow.



ownproba	probability of the current neighbor
Global proba	probability of being infected for this neighborhood
N	number of neighbors

Figure 4 : flow chart of contamination probability by neighbourhood

When the probability linked to the neighbourhood has been computed, the susceptible agent can be infected with or without symptoms according to a specific probability.

This probability to be infected with symptoms must be increased for elderly and people with specific medical conditions.
Thus, I multiply the base probability of being infected with symptoms by:

- 1.2 for elderly people from 66 to 74 years old;
- 1.4 for elderly older than 75 years.

Then, if the agent is at increased risk, I multiply by a factor of 1.2.

Again, those factors are fixed in the source code because a lot of parameters are already handled by the user.

II. Probability of recovering

The probability of recovering entered in the parameters is an average for agents infected with symptoms.

If the agent is elderly or has increased risk due to medical conditions, the probability of recovering must be increased.
I have multiplied the probability of recovering probRec by

Thus, I multiply the base probability of recovering by:

- 0.8 for elderly people from 66 to 74 years old;
- 0.7 for elderly older than 75 years.

Then, if the agent is at increased risk, I multiply by a factor of 0.8.

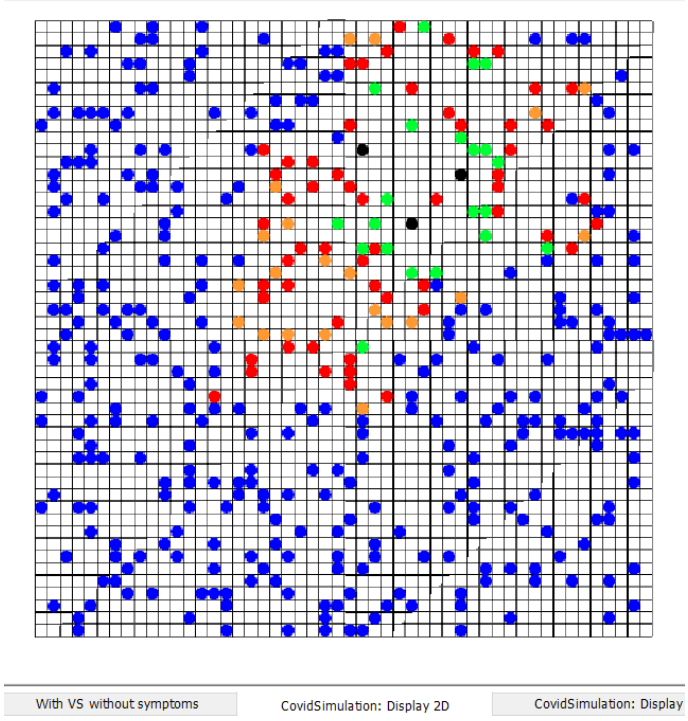
Once again, those factors are fixed in the source code because there are already a lot of parameters to handle by the user.

For agents infected without symptoms, the probability of recovering must be 1.

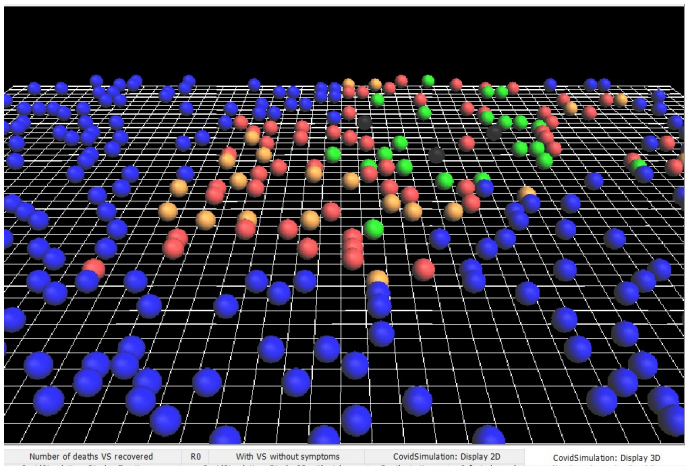
XIII. DISPLAYS

Several displays are proposed to the user :

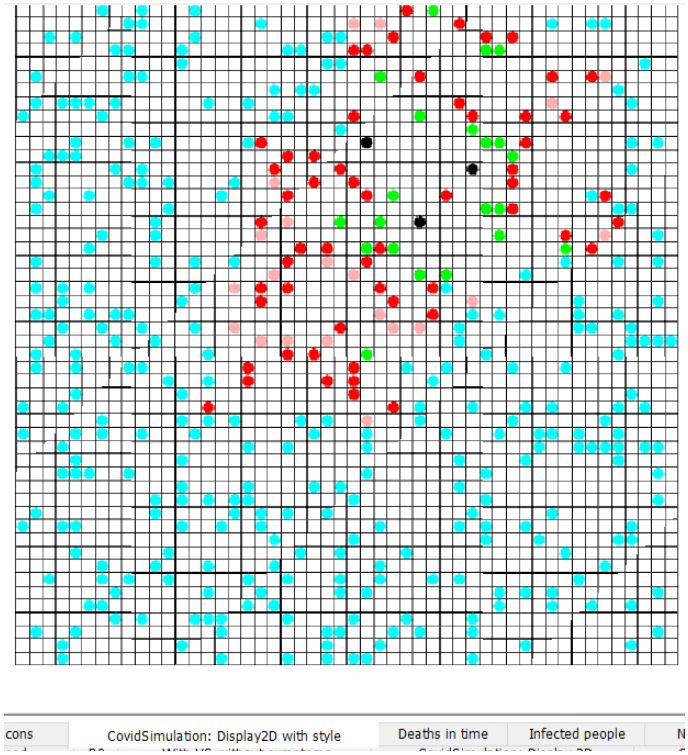
- Display 2D : a display in 2 dimensions that has been defined in the user graphical interface.



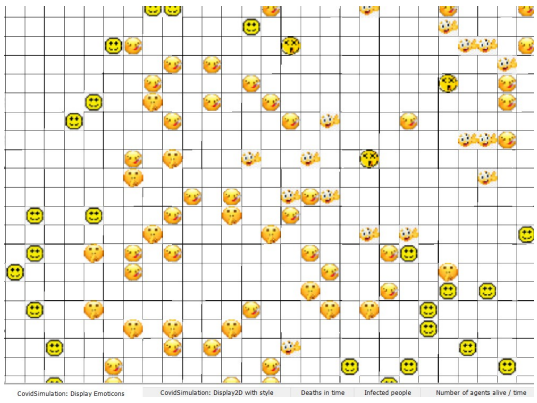
– Display 3D : a display in 3 dimensions that has been defined in the user graphical interface.



– Display 2D with style : it uses the class *DynamicAgentStyle* to handle the style of the agents.



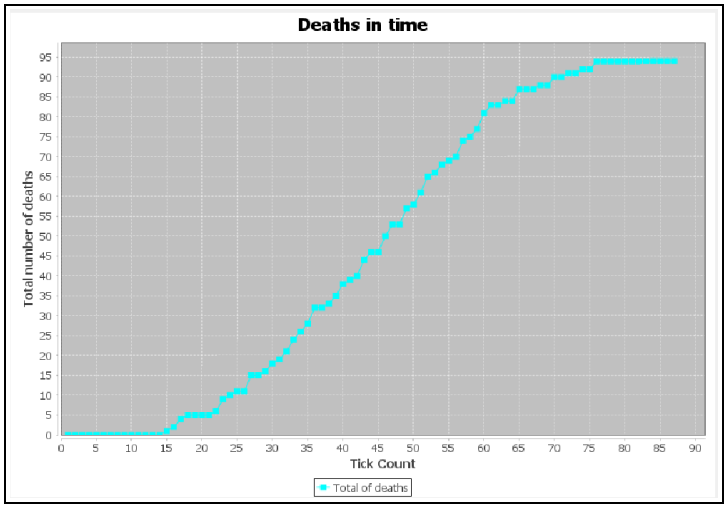
– Display emoticons : some icons are used instead of basic circles but it was much more for the fun. Indeed, this display is not really relevant as the icons are so small that it does not allow to see correctly the kind of emoticon used for a specific agent.



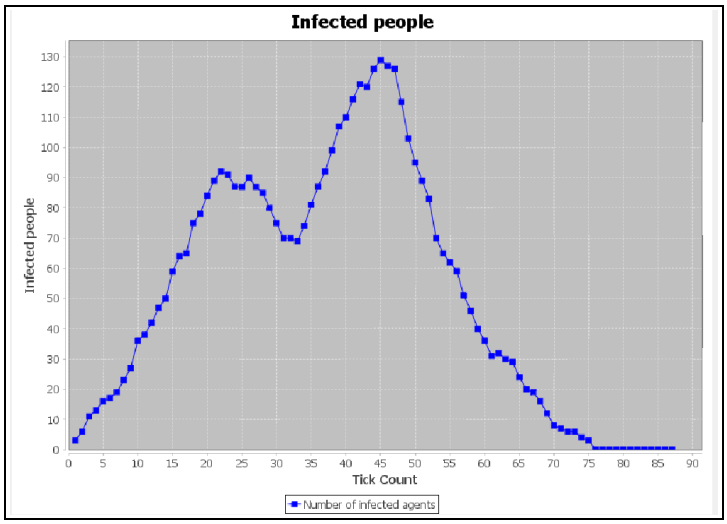
XIV. CHARTS

Several charts of “time-series” type have been created:

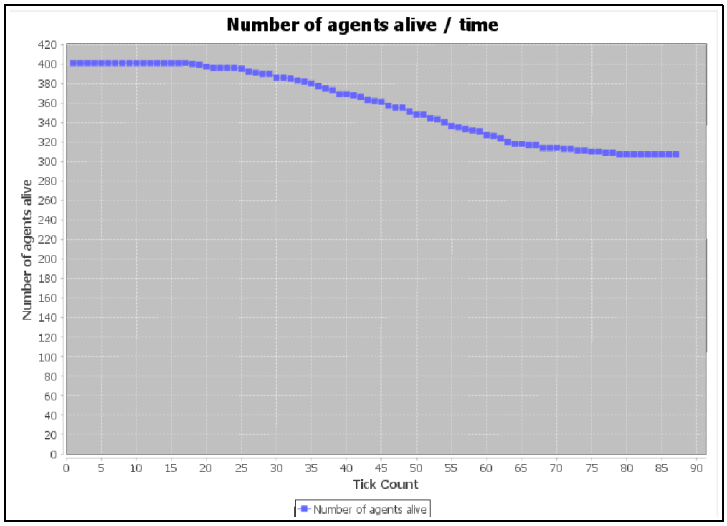
- total number of deaths during the simulation: it increases during the simulation and stabilizes when there is no more infected agents.



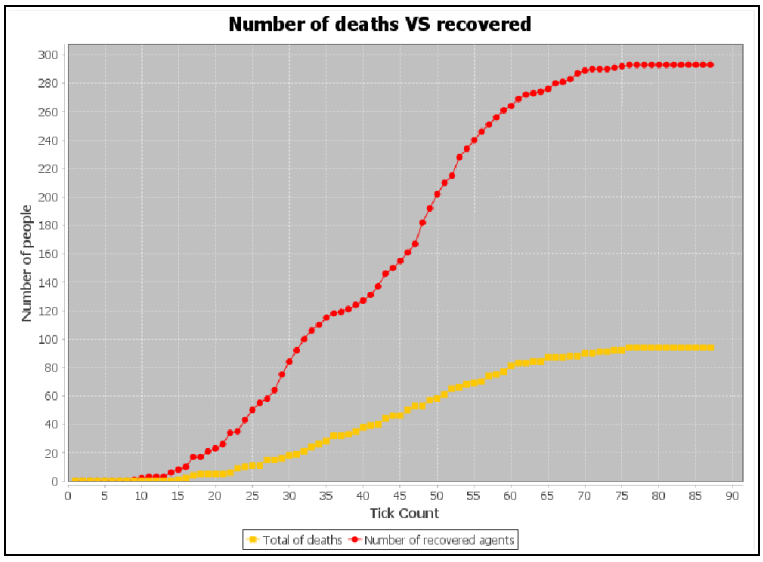
- number of infected people at a specific tick:



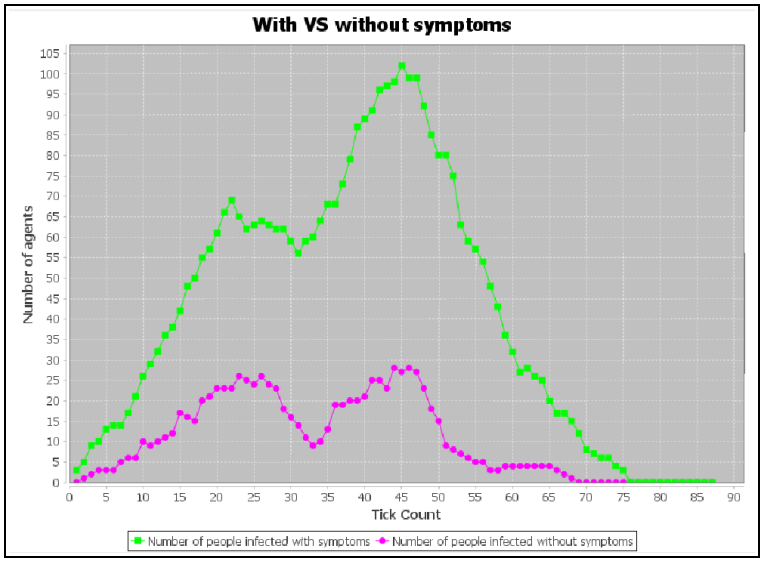
- number of alive people during the simulation:



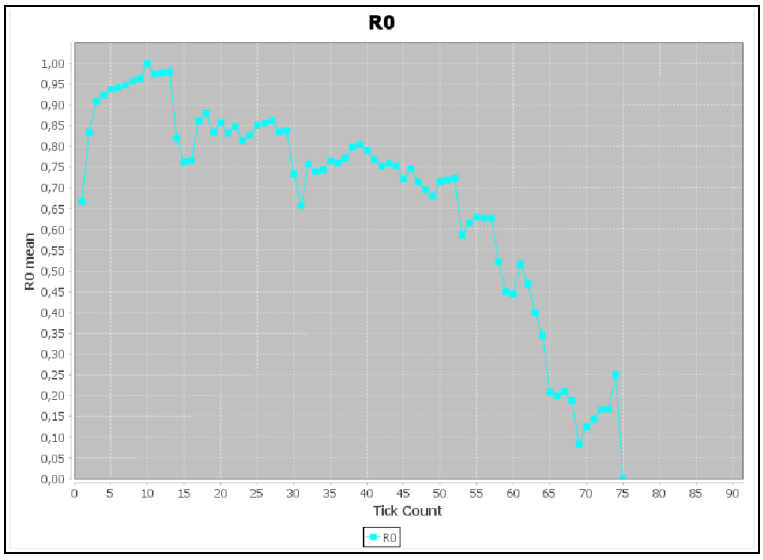
– comparison between the total number of deceased agents and the total number of recovered agents:



– comparison between the number of infected with or without symptoms at a tick:



– incidence rate R0 (average of the number of people infected by one infected agent) at a tick :



XV. SCENARIOS AND RESULTS

I. Basic scenario

The default parameters are used :

Name of the parameter	Value
Grid height	50
Grid width	50
Movement scenario	Random scenario
Number of infected agents	1
Number of susceptible agents	400
Probability of being docile (wear the mask if mandatory)	1
Probability of being elderly	0,5
Probability of being healthy	0,7
Probability of infection	0,8
Probability of recovering	0,8
Strategy 1 for pandemic limitation	None
Strategy 2 for pandemic limitation	None
Strategy 3 for pandemic limitation	None

Results: when the simulation is stabilized, we count 94 deaths and only 14 agents that have not been infected; thus, even with only one infected agent at the beginning, almost all the agents are infected at a moment.

II. Attractive places

Here, we used the default parameters except for the *movement scenario* set to Attractive places.

Results: when the simulation is stabilized, at tick 85, we count about 98 deaths and all the agents have been infected; thus, even with only one infected agent at the beginning, all the agents are infected at a moment.

III. Distanciation

Here, we used the default parameters except for the *strategy1* set to Distanciation.

Results: when the simulation is stabilized, at tick 38, we count about 6 deaths and only 18 agents that are recovered. All the other agents are still susceptible. Thus, the distanciation gives really good results.

IV. Face mask

Here, we used the default parameters except for the *strategy1* set to Face mask.

Results: when the simulation is stabilized, at tick 14, we count only 1 death and all the other agents are susceptible. This means that, when everybody wears the mask, the probobility of infection is really reduced. Thus, the face mask is a good strategy.

V. Lockdown

Here, we used the default parameters except for the *strategy1* set to Lockdown.

Results: the agents are “frozen”. Thus, the initial infected agent has infected only his neighbourhood. 1 additional agent has been infected. No deaths. Thus, the lockdown is a good strategy.

VI. Curfew

Here, we used the default parameters except for the *strategy1* set to Curfew.

Results: the agents are “frozen” during the curfew. When the simulation is stabilized, at tick 130, we count about 24 deaths and 81 agents that are recovered. All the other agents are still susceptible. Thus, the curfew is a strategy that is interesting to limit the pandemic without freezing all the population permanently (and avoid economic impact for example).

VII. Isolation of infected people

Here, we used the default parameters except for the *strategy1* set to Isolation of infected people. All the infected people move towards the hospital.

Results: all the infected people move towards the hospital. When the simulation is stabilized, at tick 54, we count about 33 deaths and 85 agents who have recovered. All the other agents are still susceptible. Thus, the isolation of infected people is quite a good strategy to limit the pandemic.

VIII. Other scenarios

This is an excerpt of scenarios, but we can, of course, play with all the parameters to imagine more other situations of simulation (older population, more anarchist population, …).

XVI. DIFFICULTIES ENCOUNTERED

I had some difficulties at the beginning to apply the way of thinking of Repast platform. Indeed, I was bothered by the fact that the logic, the visualization of the agents and the results of the simulation are embedded in the same classes.
From a logical point of view, a human can be healthy, sick or deceased. But it remains the same human. Thus I tried to create only one class Human with a status attribute. But I realized quickly that Repast was not really intended for this type of architecture. It was much more complex to count the number of infected agents, recovered agents...

Then, I created several classes of agent and when the status changed, I removed one agent from the context to create a new one with a different status.
This way, it was really easier to access to the different indicators of the simulation.

I still had to trick myself to get some “global” indicators, for example, the total number of deaths. I created a static attribute in the class *Agent* to be sure to always have access to it (I assumed that there is always one agent in the grid at any time). This way, I am able to use it in the data sets. Of course, this static attribute is reinitialized at each new simulation.
I would have preferred create this attribute in a kind of global simulation manager object. Maybe there was another solution than mine, but mine worked...

XVII. ADDITIONAL FEATURES

A javadoc documentation has been generated for all the classes.