

Multiagent Simulation

MABS Lab Work #1

STÉPHANE GALLAND

UNIVERSITÉ DE TECHNOLOGIE DE BELFORT-MONTBÉLIARD

ref: UTBM-INFO-IA54-MABS-LW1
ver: 1.2
date: 2019/12/03

www.utbm.fr

Université de Technologie de Belfort-Montbéliard
Laboratoire Connaissance et Intelligence Artificielle Distribuées
13, rue Ernest Thierry-Mieg
90010 Belfort cedex, France

1/ GOAL OF THIS LAB WORK SESSION

The goal of this lab work session is to write a simulator of the famous Pacman game.

You shall learn:

- How to write the agent perception computation.
- How to write the Ghost behavior.

2/ FIRST STEP: PREPARE THE DEVELOPMENT ENVIRONMENT

In this section, you will find the tasks to do for preparing your development environment.

2.1/ INSTALLATION OF THE ECLIPSE TOOLS

Recommended version of SARL : 0.10 (stable)

1. Download the **Eclipse product** that contains the compilation tools for the SARL programming language : <http://www.sarl.io>
2. Uncompress the Eclipse product for SARL.
3. Launch the downloaded Eclipse product.
4. Open the wizard for creating a SARL project, with the menu:
> File > New > Project > SARL > Project
5. Enter the name of the project.
6. Because the code template of this lab work requires several functions that are provided by Janus, you must install the Janus sarl run-time environment in the libraries of your project:
7. Click on the tab with the name `Libraries`.
8. In the list of the libraries, remove the `SARL Libraries`.
9. In the list of the libraries, add the `Janus Libraries`.
10. Click on "Finish", the SARL project should be created.
11. You must ensure that the configuration of your SARL project is correct (may be still a bug in the SARL environment):
 - a) Open the dialog of the properties of the SARL project by clicking on:
Right click on project > Properties > SARL > Compiler > Output Folder
 - b) Check if the field "Directory" is set to a source folder that is existing in your SARL project. If not change the property with):
`src/main/generated-sources/sarl`

Your Eclipse environment is now ready for the lab work session. You should now install the code skeleton provided by the teachers.

2.2/ INSTALLATION OF THE CODE SKELETON

The teachers provide a code skeleton that should be completed by you for finishing the tasks related to this lab work session. The steps to follow for installing the code skeleton are:

1. Get the Zip Archive file that contains the code skeleton: `LW2_skeleton.zip`.
2. Do not uncompress the Zip file.
3. Open the wizard for importing the code skeleton into the SARL project:
Right click on project > Import > General > Archive File
4. Select on the local file system the downloaded file of the code skeleton; and click on "Finish".
5. The source folders of your project shall contains SARL and Java code.
Some errors are appearing since they are related to the missed part of the code that must be provided by you.
6. Clean the workspace for ensuring that each existing file is compiled:
Menu > Project > Clean

Figure 1 on the page 4 gives an example of the structure of the SARL project that you should obtain.

2.3/ SARL DOCUMENTATION

The documentation for the SARL syntax, and the provided elements is available at:
<http://www.sarl.io/docs/official>

2.4/ BUG REPORT AND QUESTIONS

In case you have discovered an issue in the SARL tools, you could submit a description of it to the SARL development team via the Github interface:
<https://github.com/sarl/sarl>

In case you cannot discuss with your teacher, you could use the SARL forums to the SARL development team via the Google Group interface:
<https://groups.google.com/forum/#!forum/sarl>

3/ BRIEF DESCRIPTION OF THE CODE SKELETON

The skeleton contains an implementation of the environment model (a maze) `fr.utbm.info.ia54.environment`, and the agent that is managing the environment. The subpackages are or will be:

- `fr.utbm.info.ia54.environment.maze` is the package that contains the definition of the maze, and the objects inside.
- `fr.utbm.info.ia54.environment.agent` is the package in which you could find the definition of the agent and the interaction space that are needed for supported the maze environment in the simulation.

It is recommended to read this code and the associated Javadoc.

The package `fr.utbm.info.ia54.players` contains the playing agents, i.e. the ghosts.

The packages `fr.utbm.info.ia54.math` and `fr.utbm.info.ia54.ui` contain mathematical tools and the graphical user interface. The file `fr/utbm/info/ia54/PacManSimulator.java` contains the main program.

4/ WORK TO BE DONE DURING THE LAB WORK SESSION

The following sections describe the work to be done during this lab work session.

4.1/ CAPACITY TO MANAGE A MAZE

The agent `Environment` is in charge of managing the maze and the interactions with the playing agents (the ghosts), and the pacman. The agent `Environment` has the knowledge of the maze. It means that this agent has the capacity to manage the maze. This capacity is `MazeManager`, defined in the file `fr/utbm/info/ia54/environment/agent/Capacities.sarl`.

You must create a skill, i.e. an implementation of a capacity, for the capacity `MazeManager`. In the code skeleton, this skill is named `DefaultMazeManagerSkill`, and all the functions marked with `TODO` must be coded. The skill contains a reference to the maze.

You should:

- a) implement the functions in `DefaultMazeManagerSkill` that implements the functions defined in `MazeManager`.

4.2/ GHOST BEHAVIOR

The agent `Ghost` must be defined in the file `fr/utbm/info/ia54/players/Players.sarl`

This agent should:

- Pursue the pacman if it is in the field-of-view of the ghost, and the pacman has no super power.
- Evade the pacman if it is in the field-of-view of the ghost, and the pacman has the super power.
- Move randomly, but avoiding to turn back in the middle of a corridor.

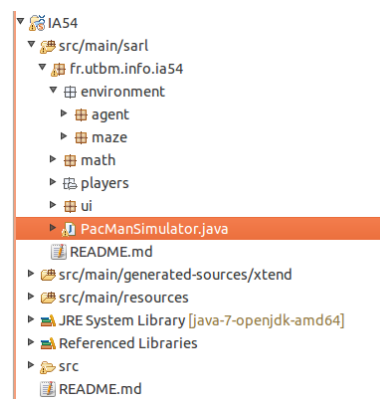


Figure 1: Example of the structure of a SARL project

