IA51: Artificial intelligence for serious game
Intermediate Exam P2018

Duration: 2h00.
Paper documents are allowed. No electronic device allowed.
English recommended, French accepted.

Part 1: Questions (5 points)

Question 1.1:

In the definition of the concept of “Agent”, four major characteristics are defined. **Provide at least two of these characteristics, and briefly explain them.**

Question 1.2:

Consider an agent-based simulator with a situated environment. **Draw a figure that describes the general software architecture of the simulator.** On this figure, you must draw:

- the agents;
- the avatars;
- the environment;
- the graphical user interface.

Question 1.3:

Explain briefly the ‘influence-reaction” model. In your explanation, your must use the words “influence“, “reaction” and “action”.

Part 2: Environment Model

The goal of this part is to define an environment that could be used for road traffic simulation. It may contain urban objects and vehicles.

Question 2.1: Environment Model (5 points)

Write the UML class diagram for the environment and its internal data structure. You should select the best data structure for supporting the continuous space of the environment. You must detail the following elements in the class diagram:

- the different types of environment objects;
- the data structure that contains the environment objects;
- the relationship with the agents.

Question 2.2: Perception Function Prototype (1 points)

Let the function `computePerception`, which computes the collection of perceived objects for each agent. **What is the prototype of this function?**

Question 2.3: Perception Algorithm (6 points)

Write the perception algorithm that corresponds to the content of the function `computePerception`, and following the UML class diagram that is proposed in the first question of this part. You could write it with pseudo-code, or the SARL language.

Part 3: Force-based Algorithm (3 points)

Let the agent A. The goal of this part of the exam is to provide the algorithm for the "force-based" algorithm related to A. This behavior permits to A to seek to a position in the environment, as fast as possible, and avoiding collisions. The inputs of the algorithm are:

- p_A is the current position of A (position x_A, y_A);
- $\vec{v}_A$ is the current velocity vector of A (not a unit vector, vector with its length in m/s);
- S_A is the maximal speed capability of the agent A (in m/s);
- A_A is the maximal linear acceleration capability of the agent A (in m/s^2);
- T is the point to reach (position x_T, y_T).
- O is set of the positions of the objects around A (position x_{o_i}, y_{o_i} where o_i is an object around).

The output of the algorithm is the linear motion vector $\vec{m} = (m_x, m_y)$ that has its length in m/s^2.
m/s means "speed in meters per second", and m/s^2 means "acceleration in meters per square second."

Provide the "force-based" algorithm. You could write it with pseudo-code, or the SARL language.