IA51: AI for Games and Serious Games - Final Exam P2017

Duration: 2h.
No document nor calculator nor smart phone nor touchpad allowed.
English and French are accepted.
Malus of 1 point for dirty sheets.
Each part must be written on separated sheets.

Part 1: Agent Simulator (15 points)

Question 1.1: Theory Questions

1. (2 pts) Cite the four properties of an agent according to the definition of Michael Wooldridge. Explain briefly each of them.
2. (1 pts) Agents may interact together by using the stigmergy principle. What is the stigmergy principle? Explain briefly the key elements of it.
3. (1 pts) What is the difference between an influence provided by an agent and an action in the environment?
4. (2 pts) Considering a collection of agents that inhabit in the simulation, several methods for *executing* the agents exist in order to simulate the theoritical parallel execution of the agents. Excluding any thread-based or network-based approach, propose a method or an algorithm for executing the agents.

Exercise 1.2:

You want to simulate a group of children who are playing the tag game ("jeu du chat"). A group of players (two or more) decide who is going to be "it". The player selected to be "it" then chases the others, attempting to get close enough to "tag" one of them (touching them with a hand) while the others try to escape. A tag makes the tagged player "it".

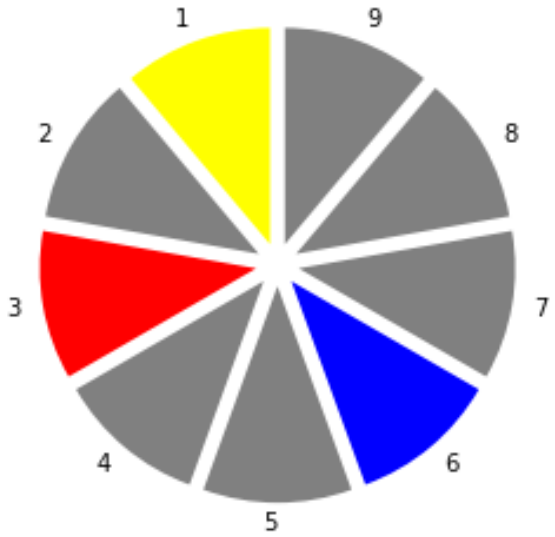
1. (1 pts) The list of environment objects and agent bodies must be stored into the environment entity of the simulator. What is the efficient data structure for storing these objects and bodies? What are the constraints and the limits of the selected data structure?
2. (3 pts) Write the algorithm for the player selected to be "it" and chasing the other players. This algorithm must enable the player to *chase* the closest player, and to *avoid collision* with objects and other players.
3. (2 pts) Write the algorithm for the player who want to *evade* the chasing player and *avoid collision* with objects and other players.

Exercise 1.3: (3 pts)

You want to write a perception algorithm for the agents. This novel perception algorithm supports occlusion of objects: occluded objects[1] must not be seen by the agents, even if they are in the perception field-of-view. Assuming that the list of the objects in the field-of-view is provided by the function:
`List<Object> getFovObjects()`
propose an algorithm that removes the occluded objects from the list. If you have made several design hypothesis, you must quickly explain them.

Part 2: Machine Learning: Turn around (5 points)

Consider the circle of states above, which depicts the states of an MDP. The agent can choose between two actions: going one step clock-wise or one step counter-clock-wise.



Description of the reward $R(s, a)$:

- The agent receives a reward of 5 in state 1 (yellow state).
- The agent receives a reward of -3 in state 3 (red state).
- The agent receives a reward of 10 in state 6 (blue state).
- The agent receives a reward of -1 in every other state.

To simplify the computations, we assume $\gamma = 0.5$.

1. Recall the algorithm Q-Iteration.
2. Apply 3 iterations of the Q-iteration method: show the matrix of the Q-values for each iteration.
3. What is the greedy policy assuming the Q-values obtained after 3 iterations?

[1] "objets cachés"