LEBANESE AMERICAN UNIVERSITY

Department of Electrical and Computer Engineering

Intelligent Engineering Algorithms

COE 544/744

Fall 2021

Instructor: Joe Tekli

Course Projects Description

I- Objective

Allowing students to implement, test, and manipulate software agents with intelligent features, e.g., intelligently perceiving their surrounding environment, resisting noise, capable of approximate (fuzzy) data processing, and/or simulating human behavior.

II- General Procedure:

- Each student group will work on two projects designated by the instructor, which are described in the following section.
- For each project, students are supposed to design the corresponding PEAS conceptual model of the agent in question, describing: i) the environment, ii) its sensors, iii) its actuators, and iv) the corresponding performance functions (cf. Chapter 2). Therefore, students must explain and support their choices in terms of the intelligent methods or algorithms adopted to implement the codes of the agents (when applicable).
- A short technical report, 3-5 pages long, formatted following the standard *IEEE Computer Society Transactions* template (provided on Blackboard), should be submitted to the instructor along with the software source code of each project. The technical report should follow a general scientific organization, including the following main sections:
  - Introduction: subject and objective(s) of the project
  - Background: context, prerequisites, and existing solutions
  - Proposal: underlying concepts, building blocks, design models, & implementation
  - Experimental evaluation: test protocol, test metrics, test data, & experimental results
  - Conclusion: synthesis, personal experience, and project perspectives

  Note that section titles, order, and overall presentation organization can (and should) be adapted and fine-tuned based on the nature of each project.
- A project presentation/demonstration will also be conducted per group. Presentation duration: 15 minutes (10 minutes for groups of two).
- Projects' submission and presentation deadlines:
  - Project # 1: ***Monday 3rd of November 2020***
  - Project # 2: *End of semester* (to be specified by the instructor).

III- Project Topics

The projects proposed this semester mainly focus on search algorithms. Project topics are briefly described below, and mainly cover the design, implementation, and testing of: 1) a state-space search vacuum cleaner agent, which is later extended into a: 2) multi-agent adversarial and stochastic vacuum cleaning environment.

1. State-space Search Vacuum Cleaner Agent

The first project aims at designing and building an intelligent agent modeling an automatic vacuum cleaner (cf. examples in Ch. 2-3). We would like to model and implement the software agent governing the behavior of a vacuum cleaner operating in an environment consisting of a tiled surface made of $n \times m$ tiles (where n and m can be chosen by the user). The tiled surface can also have boundaries (walls) that cannot be traversed by the agent. A tile can have two states: clean or dirty. Dust distribution can be completely random, it can follow a certain probability law, or it can be manually defined by the user. The objective of the vacuum cleaner is to clean, automatically and autonomously, the floor of the chamber in a way to maximize performance.

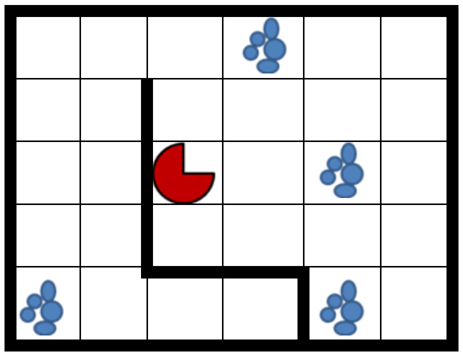

Figure 1. Sample environment for a vacuum cleaner agent, consisting of 5×6 tiles, the agent being represented as a red *packman*, while dirt is represented as small blue dots. Walls and obstacles can be added to make the environment more realistic, like the wall separating the above surface into two rooms.

<u>Hints</u>: Students need to address the problem as a state-space search problem definition for navigating to a particular position in a maze. They can create a new search problem definition for finding a path that passes through all of the dirt tiles in the maze (multiple-goal search problem). The students can consider different heuristics, or greedy techniques, to improve the quality of the agent.

2. Multi-agent Adversarial Vacuum Cleaner Agent

The second project aims at extending the first one where the environment will include another "dirt producer" agent (represented in black in Figure 2) whose job is to dump dirt on the floor. In other words, the two agents will be competing: the dirt producer agent will try to create a mess while the vacuum cleaner agent will try to clean it up. Both of them will be activing simultaneously on the environment. The agents cannot be positioned on the same tile at the same time. Students can implement a "timed" version of the environment where both agents have a certain amount of time (set by the user) to try to conquer each other by either producing or sucking more dirt. An external performance evaluation function should be designed to decide about the winner. The scenario can also be extend to consider more than one vacuum cleaner and dirt producer agents working in the same environment at the same time, where the number of agents and their nature is chosen by the user. In such a scenario, students can evaluated the performance of each agent to identify the best vacuum cleaner and best dirt producer among them all, considering different algorithms to design each agent.

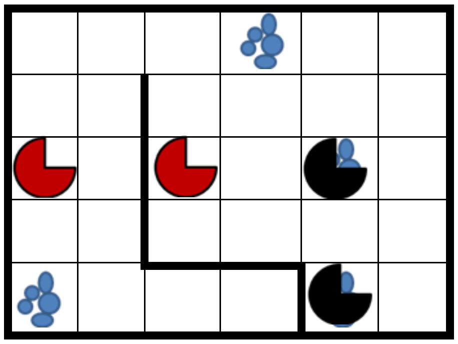

Figure 2. Sample environment for a vacuum cleaner agent, consisting of 5×6 tiles, the agent being represented as a red *packman*, and including two dirt producing agents represented as black *packmen*.

Hints: The students need to address the second project as an adversarial and a stochastic search problem. They can investigate *adversarial search* algorithms, such as the *Minimax* algorithm and the *Alpha-Beta pruning* algorithm. They can also investigate stochastic search algorithms such as *Expectimax*, as well as designing evaluation functions for limited-depth search. They can also investigate machine learning models to enhance the agents' performance through supervised learning.