Multi-agent systems
Project 1, 2 & 3

Linda Belkessa, Mahdi Zargayouna

To send by e-mail by 13 December 2024

General

Projects are carried out in groups of two students. A report will accompany the project, detailing :
— design choices (types of agents, interaction modes, representation of the environment, etc.),
— the structuring of the program (control system, environment, agents, etc.)
— Difficulties encountered (design, programming, etc.),
— a user manual (step by step).
— Deliverables : an archive, named after the students' names, containing :
1. The source codes of the project (in an archive, named after the students' names)
2. A report explaining the design choices, the difficulties encountered and a user manual for the application. Particular attention will be paid to the quality of the report.

— The report and source codes should be sent by email to *linda.belkessa@univ-eiffel.fr* and please consider using a Cloud for the sending to avoid saturating the mailbox.
— 15 minutes demos are planned for 20 December 2023.
A great deal of attention will have to be paid to the report.

Project 1 : Public Transport Mobility Simulation

Design and implement a multi-agent system that represents the movements of passengers and public transport vehicles in a city. The city is described at a minimum by roads and bus stops. Passengers have an origin and a destination in the city that they can reach by foot or by public transport. The public transport follow predefined timetables. The objective is to simulate the movements of passengers and vehicles. In a second step, we would like to introduce disturbances in the system. The disturbances can be a disconnection of a road or the breakdown of a vehicle. The objective is to observe and quantify the impact of these disturbances on the travel time of passengers.

To do so, you will need :

1. Define the system parameters
2. Represent the city (e.g. grid)
3. Represent the public transport vehicles
4. Define passengers and vehicles movements (agents behaviors)

Any enrichment of the project will be rewarded with bonuses.

Project 2 : STRIPS planner

Design and build a planner (forward or backward chaining) working with the STRIPS language. The program takes as input a text file describing :

1. the possible actions (an action is made of PRE, DEL and ADD rules),
2. the initial state of the world, and
3. a goal to achieve.

It provides as output the sequence of actions (if any) leading from the initial world to the world satisfying the goal.

You can either define a generic planner with a standard search strategy (Depth-first or Breadth-first), or a specific planner to a certain problem, with a heuristic of your own.

Project 3 : Shared Micromobility Demand and Recharging Simulation

Design a multi-agent simulation to manage a basic fleet of shared micromobility vehicles (such as bikes or scooters) in a city. The goal is to simulate how vehicles are used and how they need to be recharged or maintained, without requiring complex optimization or real-time redistribution.

Vehicle States

Each vehicle can be in one of these states :
— **Available** : ready for a user to pick up.
— **In Use** : being used by a user.
— **Needs Recharging** : out of power and waiting for a recharge.

Basic Rules

User Agents

— Each user decides whether to pick up a vehicle based on availability within a certain distance (number of cells for example).
— After completing their trip, they return the vehicle, which consumes a fixed amount of battery.

Vehicle State Transitions

— **Available** : Starts here ; ready to be picked up by a user.
— **In Use** : When a user picks up a vehicle, it becomes in use.
— **Needs Recharging** : After a trip, if the vehicle battery is low, it moves to (Needs Recharging) and cannot be picked up again until recharged.

Recharge

Recharging occurs automatically after a set time (or can be skipped with instant recharging for simplicity).

Project Steps

Define Basic Parameters

— **Total vehicles and user demand probability** : Decide on the total number of vehicles and how often users need them.
— **Battery depletion rate** : Each trip depletes a set amount of battery.

Environment Setup

Use a small grid to represent a city.

Simulation Scenarios

— **Random Demand** : Users appear randomly across the grid to pick up vehicles.
— **High-Demand Zone (optional)** : Set a zone with a higher demand to observe the system response to peak use.

Track Key Metrics

— **Vehicle availability** : Percentage of vehicles that are available at any given time.
— **User wait times** : Time users spend waiting for an available vehicle.
— **Vehicle utilization** : Average percentage of time each vehicle spends in use.

Bonus (Optional)

— Simulate a rebalancing process by manually adjusting vehicle positions at the end of each simulation period.
— Experiment with different battery depletion rates or demand probabilities to observe how these factors affect the system performance.