Davide Tessarolo – ID: 211457

AOSE Final Project Report

In this brief report I will explain my take on the second exercise of the project, showing the various components and detailing the problems encountered.

# Solution

The solution is composed by a number of different agents and artifacts, so here is a list with all different implementation details about them:

* **GameManager**: the GameManager’s implementation is very similar to the provided one, with some minor additions to the *createBox* desire. Here, instead of creating only two beliefs for the box’s start and destination position, an additional two are created to detail the box’s start and destination areas. This addition was made in order to have a cleaner way for Drones and RailBots to check on how to manage the box.
* **PickupArea**: the PickupAreas are rather simple and either call drones to pick up newly spawned boxes or deliver boxes that arrived at destination.
* **Drones**: the Drone agent completes a number of different tasks: it delivers boxes to the RailBots (which it has to wait for in case they are busy) or, on the contrary, it delivers boxes from RailBots to the different PickupAreas. Also, it needs to recharge after every travel.
* **RailBot**: this agent has a pre-existing desire, namely *sortBox*. This default desire is the one who makes the RailBot sort the various boxes taken from the ExchangeArea. Drones can delete di desire and swap it with the passBox one. The passBox desire can branch into two different ones: if the start and destination areas of the box are the same, then the RailBot immediately calls a drone, if they are different it picks up the box and brings it to the ExchangeArea, notifying the SortingBot.
* **SortingBot**: the SortingBot’s implementation is pretty straightforward: like for the RailBot, it has a pre-existing sortBox desire, with a belief need\_to\_sort(Box) prerequisite. RailBots add a new need\_to\_sort(Box) belief each time they deposit a new box onto the ExchangeArea.

# Problems encountered

The two main problems I’ve encountered during the execution of this second part of the project where: how to make agents, especially drones, deliver the boxes without having the start and destination area, and how to make RailBots and the SortingBot deliver the boxes (since I couldn’t use locks, especially on the SortingBot).

The first problem, as I’ve already said, was solved by having the GameManager add the box’s start and destination area, as beliefs, to the artifact. The second problem was circumvented by not using locks, but rather a system where either the agent waits until a belief gets added to its KB (SortingBot) or it does something else until the belief gets added to the KB (RailBots, they keep bringing boxes to the ExchangeArea until, eventually, a box comes from the SortingBot).

# Possible improvements

Possible improvements to the system, in order to make it faster at boxes delivery, could be made by modifying the way drones get called by PickupAreas and RailBots, possibly prioritizing the nearest drone available. This would cut fly and response time from drones.