Artificial Intelligence and Intelligent Agents

CW1 – Knowledge Representation & Automated Planning

Student: César Gutiérrez Carrero (H00319640)

**1 Introduction**

This report covers the principal aspects of the solution. It describes the decisions taken regarding the design of the problem, and provides an overview of the layout of the spacecraft and the space, making it easier for the reader to understand the code.

**2 The problem´s solution**

The solution uses types to represent the objects it has to work with. There are four kinds of devices: probes, landers, MAVs, and capsules. The personnel are split into various roles: captain, engineer, science officer, navigator, and rescuer. The ship consists of several sections: bridge, launch bay, science lab, engineering, and bedrooms. And, lastly, the space is divided into regions which can contain entities such as, planets and nebulas.

**2.1 Predicates**

To begin with, there is a very basic predicate stating that two sections are connected so that a person can pass from one to another. The same idea is applied to define that two regions are adjacent so the ship can travel between them. These regions can also contain entities (a planet or a nebula). And, since there can exist asteroid belts in the space, there is a predicate for when a region has one. For planets, there exist two predicates, one declares that it has high radiation, and the other states whether the planet has actually a landing place.

In order for someone to move around the ship, first, it is necessary to state in which section that person is. Besides, each role has some actions available: a navigator can receive the order to travel to a certain region; a science officer can transfer plasma collected from a specific nebula; an engineer can be repairing the spacecraft inside a MAV and could also be calling for help if the MAV is disabled; a rescuer can be rescuing a disabled MAV.

Then, there are predicates for any of the states a device can have on the spacecraft. The first is common for any device and defines when it is on board. In addition, a probe can be destroyed, and lander can crash and a MAV can be disabled. Also, a probe could be deployed studying a planet or a nebula in a certain region, a lander can land on a particular planet and deploy one or two antennae to communicate the results of the exploration.

Moreover, other predicates express the status of the spacecraft and what it is doing. It can be either on a region of space or landed on a planet. It might be travelling that is, the captain has order to go somewhere and the ship is on a travel until it reaches the destination. It could also be damaged. It can be scanning the surface of a planet or exploring it. It also needs to keep track of the planets already scanned. It can have the touchdown location and/or the results of a planetary scan of a planet. It can also have studies of plasma from a particular nebula. And, it is necessary to know where that plasma is inside the spacecraft.

**2.2 Actions**

* Move. It allows one person to move from one section to another.
* Order\_to\_travel\_to\_region. The captain orders a navigator to travel to a particular region. This action makes the ship start *travelling* which keeps other actions from firing until it reaches the destination.
* Travelling\_to\_region. It allows the spacecraft to travel from one region to another. When the region the ship has travelled to and the destination are the same, it stops *travelling*.
* Sent\_to\_repair\_ship. Given it is *damaged*, it sends an engineer inside a MAV to repair the spacecraft. There are three engineers involved in this operation, the pilot, the one that monitors the operation from engineering, and the one that controls everything from the launch bay. If the region *has\_nebula*, the MAV is *disabled* and the pilot is *calling\_for\_help*.
* Call\_back\_mav. Given the MAV is not *disabled*, the action brings it back to the ship fixing it. It also returns the pilot back to the launch bay.
* Sent\_to\_collect\_plasma and sent\_to\_scan\_planet. These actions are very similar, given the ship is neither *damaged* nor *travelling* and a probe is *on\_board*, it deploys a probe either to study a nebula or a planet. In the latter case, the action also considers that the spacecraft is not *scanning\_surface\_of\_planet* and that it has not been *scanned\_planet* yet, preventing two probes from scanning a planet simultaneously and ensuring that a planet is not scanned twice.
* Call\_back\_probe\_with\_plasma and call\_back\_probe\_from\_planet. They bring a probe *on\_board*, leaving the collected *plasma\_from\_nebula\_at\_section* launch bay or the *info\_of\_touchdown\_location* in case the planet *has\_place\_to\_land*. Besides, the second action marks the planet as scanned too.

*Studies\_of\_plasma\_from\_nebula* can only be achieved when there is *plasma\_from\_nebula\_at\_section* science lab. In order to do this, an officer will have to move first to the launch bay to collect the plasma (action: officer\_collects\_plasma\_from) and then go to the science lab to study it (action: officer\_studies\_plasma\_from), with the predicate *transferring\_plasma\_from* active in between.

*Results\_of\_planetary\_scan* from a particular planet can only be achieved if there is a lander *exploring\_planet* with one or two deployed antennae (action: receive\_scanning\_results). For this to happen, first, a lander needs to land on the planet provided it has *info\_of\_touchdown\_location* (action: attempt\_to\_land\_on\_planet).

**3 Additional feature**

* Actions *launch\_spacecraft* and *land\_spacecraft*.
* Actions *rescue\_disabled\_MAV* and *enable\_MAV\_and\_return*.

**4 Layout of the spacecraft and space**

**5 How well does the domain scale?**

Choose a planning problem and increase some feature of the domain (e.g., the number of locations to explore, the number of people on the spacecraft, the number of probes, etc.). Does this have an effect on the planning times or the plan length? Does the planner ever fail to generate a plan? Include a brief description of these results in the report (maximum 2 additional pages). Use tables and graphs to illustrate the data you collected. Speculate on the robustness of your domain and how you might improve it to scale to larger problem instances.