Mission Sidequest 15.2: Ship Arming: EMP Missile

Start date: 30 September 2017
Due: 23 October 2017, 23:59

Readings:

- Textbook Section 2.5

With the sabotaging of the flagship and the likelihood of numerous upcoming battles in the near future, the ship's Chief Combat Officer, having consulted with Grandmaster Hartin, has requested for emergency augmentation of the ship's armaments. Scottie has been swiftly re-integrating several old, but powerful weapons that have previously been decommissioned, back into the ship's main weapons system. The hardware is mostly in good working condition, but Scottie is facing several problems with some of the modules in the software. Grandmaster Hartin has asked that once the numerical modules are repaired and working, spare manpower may be directed to rework the weapons modules.

Briefing Report

This is the briefing report for the **EMP Missile**.

The EMP Missile was envisioned as a long-range, high-precision alternative to the EMP Super Blast, allowing the delivery of pinpoint EMP payloads to specific clusters of hostile ships, instead of releasing a large blast that paralysed ships only in the immediate vicinity. Furthermore, an ingenious breakthrough allowed the EMP payloads to distinguish between hostile and friendly ships, paralysing only the former. However, this drove up costs, and near the completion of the missile's research and development, it was put on indefinite hold due to diversion of research funds and priorities, and eventually decommissioned. Now, our ship's engineers are trying to salvage what they can from the project.

Both the hardware and software of the EMP Missile are actually in good working condition. However, practical concerns now require us to add in an additional optimisation module for the missile. Originally, it had been envisioned that the EMP Missile would be fired with relative impunity. But in its current state, the high cost and limited number of missiles mean that each target should be chosen wisely.

Technical Specifications

The EMP Missile has an essentially unlimited range, allowing it to target any hostiles that our ship's radar system can detect. On contact with its target, it releases an EMP blast with radius r, and any other hostile ships caught within the blast will also be taken out. What is interesting is that the EMP shockwave will not only destabilize hostile enemy ships caught in its blast, but cause those ships to explode with yet another EMP shockwave of the same

radius r. This chain reaction goes on until there are no more hostile enemy ships in the blast radius.

The ship has K EMP missiles, and all of them will share the same blast radius r. The ship's engineer can only tune the ship's blast radius to a precision of 2 decimal places. The initial blast radius for all missiles can be set to any arbitrary value (up to 2 d.p.), but intuitively the larger the blast radius, the more power the module will consume. The ship captain therefore wants to *minimize* the the blast radius used and yet be able to destroy all hostile ships with the K missiles available.

The optimisation module should accept as input a list of hostile ships, each described as a 4-tuples (ship-ID, x-coord, y-coord, z-coord), as well as the number of missiles available for deployment, and return a 2-tuple (radius, list of hostile ships to target), such that:

1. The blast radius (to 2 d.p.) is minimized
2. The number of missiles deployed is minimized

in that order of priority. That is, the optimisation module should attempt to take out all hostile ships in sight with the available number of missiles in as small blast radius as possible. Use the given constants in the template and the signature:

```
function acquire_missile_targets(hostile_list, number_of_missiles) {
    ...
}
```

Example

The command

```
acquire_missile_targets(
    list(
        list("TIE0001", 100, 200, -80),
        list("TIE0002", 100, 200, 0),
        list("TIE0003", 100, 200, 50),
        list("TIE0004", 100, 200, 120),
        list("TIE0005", 100, 200, 200),
        list("TIE0006", 100, 200, 250),
        list("TIE0007", 100, 200, 320),
        list("TIE0008", 100, 200, 500),
        list("TIE0009", 100, 200, 800),
        list("TIE0010", 100, 200, 820)
    ),
    3
);
```

could return as a possible solution

```
(80, ("TIE0009", "TIE0008", "TIE0001"))
```

Note that this is not the only possible solution. Any equally good solution is acceptable.

Task Constraints

1. The number of enemy targets (elements in `hostile_list`) will be no more than 50
2. The coordinates of each enemy target (`x_coord`, `y_coord` and `z_coord`) will be an integer between -1,000,000 and 1,000,000
3. The number of missiles available to you will be at least 1 and no more than the number of enemy targets present

This mission consists of **two** tasks.

Task 1:

Describe concisely one or several possible solutions that you have considered. Explain the benefits or disadvantages of each solution taking into consideration the order of growth in time and space, and complexity of implementing the solution. Identify the solution you will be implementing for Task 2.

Task 2:

Write the optimisation module. Use the given constants and signature in the template provided. Comment concisely what your program is doing in relation to the description provided in Task 1.

End of briefing report. Please enquire on the forums if you need any further clarification.

Submission

To submit your work to the Academy, remember to click "Save", then click "Finalize Submission". Note that submission is final and that any mistakes in submission requires extra effort from a tutor or the lecturer himself to fix.