SIMULATION OF A SPACE MODULE IN THE ARTEMIS MISSION CONTEXT

Advanced Data Structures for the Artemis Mission

Tutorial goal:

This tutorial expands your simulation capabilities by incorporating **advanced data structures** to handle complex resource management and trajectory planning challenges. You will focus on the practical application of binary trees for resource allocation and graphs for optimizing travel paths between Earth, the Moon, and other points of interest in Artemis missions.

So let's start !

Part 1: Implementing Binary Trees for Resource Management

Context:

Continuing from TP 1, where basic classes and inheritance were introduced, this part will involve creating a binary search tree (BST) to manage varying resource needs efficiently for different types of space modules.

Step 1: Creating the ResourceTree Class

Learning Objective:

- Understand how to implement and utilize binary trees to manage sorted data efficiently.
- Learn to perform insert, search, and traversal operations that will help in resource allocation.

Exercise:

1. Define the ResourceTree class that will manage resources like oxygen, water, and food.
2. Implement methods to add resources, search for specific resources, and display all resources in a sorted manner.
3. Simulate adding resources to the tree, searching for specific items, and displaying the tree's contents.

Step 2: Resource Allocation Simulation

Learning Objective:

- Apply the binary search tree structure you've built to simulate and manage the allocation of resources based on specific mission requirements.

Exercise:

1. Simulate different scenarios where resources need to be dynamically allocated to modules based on mission requirements such as duration and destination.

2. Create instances for both habitat and cargo modules, using the resource tree to manage their specific needs.

Part 2: Graphs for Trajectory Optimization

Context:

Building on the graph skills from TP 2, explore the path optimization with Dijkstra's algorithm to include considerations like fuel costs and mission time.

Step 1: Implementing the TrajectoryGraph Class

Learning Objective:

- Apply graph theory to solve problems related to paths and network flows.
- Utilize Dijkstra's algorithm to find the shortest path and understand the importance of optimal path planning in logistics.

Exercise:

1. Define the TrajectoryGraph class and add edges representing possible travel paths with associated costs like fuel and time.
2. Implement Dijkstra's algorithm within the class to determine the most efficient travel routes.
3. Simulate finding the shortest path from Earth to the Moon, considering factors like fuel availability and mission urgency.

Conclusion:

TP 2 will enable you to apply complex data structures in a practical context, enhancing your problem-solving skills and understanding of algorithmic strategies. This project prepares you for more sophisticated challenges, such as those found in real-world aerospace engineering tasks related to future Artemis missions and beyond.