AIC2024: Space Games

1 Lore

Two teams are currently competing in the 26th edition of the *Space Games*, where they must colonize a planet. Believe it or not, this is one of the highest viewed competitions on Earth! One of the main attractions is that *everything* is allowed in order to sabotage the opponent. Moreover, the atmosphere of these planets is usually not breathable and participants (astronauts) must wear a space suit at almost all times.

To make things more interesting, the sponsors of the Space Games provide only the flimsiest and cheapest space suits they can find, which break when performing almost any type of action (fortunately, the astronauts are rescued whenever their space suit break, although people suspect that this happens because it would be hard for the sponsors to find replacements).

Can you guide your team to victory, even in these helpless circumstances?

2 The Game

The objective of both teams is to collect as much oxygen as they can in 1000 rounds. However, if a team manages to sabotage all of the opponent's headquarters (HQ), they win automatically, regardless of the oxygen that each team has at the moment. If both teams survive until the very end and end up with exactly the same amount of oxygen (including the oxygen on each astronaut!), the winner team is chosen according to the following criteria, in order:

1. The team with more HQs.

2. The team with more settlements.

3. The team with more care packages.

4. The team with more sum of Health Points on their HQs.

5. Randomly.

3 Economy

Oxygen is the only resource in this game. It not only serves as one of the main winning criteria, but it also allows astronauts to move through the planet safely. The HQs produce oxygen at a rate of 10 units per round. Astronauts can gather additional oxygen by collecting certain types of care packages.

It is important to note that oxygen is **not** a global resource: whenever an astronaut gathers oxygen, that oxygen goes directly to its *parent* structure (see Section 8) and this oxygen is not shared with other structures from the same team.

4 Map

The map consists of a grid of dimensions between 30×30 and 60×60. Astronauts and structures occupy exactly one tile and all the tiles of the map are accessible by astronauts except if there is water, another astronaut, a HQ or a settlement. Each tile is of one of three possible types: Land, Water and Hot Zone (you may see that some maps will also have mountains for aesthetic purposes, but they will be read as "Water" inside the code, as they have the same behavior).

Each Hot Zone is guaranteed to be a connected set of at least 9 tiles. At each round of the game, there is a small chance that the sponsors drop one or several Care Packages on non-water tiles, with increased chance on Hot Zones. There can only be one package in each tile. If a sponsor drops a new package on top of an existing package, the old package is replaced by the new one.

Units (astronauts and owned structures) have finite vision range. Moreover, the vision between units is not shared. This means that all objects on the map that are outside of a given unit's vision range cannot be queried during that unit's turn.

Each team starts with between one and three HQs and it is guaranteed that all maps are symmetric. This symmetry may be horizontal, vertical or rotational. Care packages are always dropped in symmetrical tiles as well.

This year **maps do not have an offset**. That means that a map with dimensions (X, Y) will have as corners locations [0,0], [X-1, 0], [0, Y-1] and [X-1, Y-1]. Units can also query the dimensions of the map at any time.

5   Unit

Each unit has a unique identifying number (ID) chosen randomly between 1 and 10.000. There are two types of units: **astronauts** and **structures**. Whenever a new astronaut is created, there is a period of 5 turns in which the unit is still in construction and can't move, attack or perform any other action. Whenever an astronaut is built, the structure that builds it must specify the amount of oxygen to include in its suit. At the end of every turn in which the astronaut is not in construction, it loses 1 oxygen. If an astronaut reaches 0 oxygen, it leaves the map. Whenever an astronaut is recruited, the recruiter can decide to equip it with a care package. Equipping an astronaut with a care package gives special abilities to the equipped astronaut and sometimes allows them to perform additional actions, as listed in Section 9.

The parameters of the astronauts are indicated in the following table. All distances are shown in squared units. For instance, an astronaut with 12 vision range at $(0,0)$ can query which object is at $(2,1)$ since $2^2 + 1^2 \leq 12$, but it cannot query which object is at $(2,3)$ since $2^2 + 3^2 > 12$.

Movement Range	2
Vision Range	25
Movement Cooldown	1
Action Range	2

Additionally, there are two types of owned structures, whose parameters are listed below. Structures don't require or consume oxygen, but they can't move. Moreover, structures have health points that are decreased by 1 whenever they are sabotaged. If these health points reach 0, the structure is destroyed.

	HQ	Settlement
Vision Range	64	49
Health Points	5	2
Action Range	2	2

There are two additional Neutral structures: the Dome and the Hyperjump. These structures cannot be controlled by the teams and have 1 Health Point (are destroyed by one instance of sabotage).

6   Movement Cooldown

Astronauts can only move whenever their movement cooldown is strictly less than 1. Every time that the astronaut moves, its movement cooldown increases by 1. This increase value is the one given in the table in Section 5. If the astronaut moves in a diagonal direction, the cooldown added is multiplied by 1.4142, which is approximately $\sqrt{2}$. At the beginning of every turn, the movement cooldown decreases by 1.

7   Actions

This year there is the unique mechanic that **when astronauts perform a non-movement action** (jumping is considered a movement action)**, their space suit breaks and they leave the map**. The list of actions that astronauts can perform is the following:

- **Move**: Moves the astronaut to a given direction (as stated in the previous section, it has a cooldown).

- **Jump**: Jumps to a direction a specified amount of spaces (from 1 to 3). It can only be performed by astronauts when on top of a Hyperjump. (it has no cooldown)

- **Sabotage**: Sabotages an adjacent astronaut or structure. If an astronaut is sabotaged, its space suit breaks and is teleported back to base. If a structure is sabotaged, it loses 1 Health Point.

- **Terraform**: Terraforms a tile. Whenever an astronaut (from any team) ends its turn in a terraformed tile, it loses half the oxygen it would normally lose. Terraforming does not remove the distinction between Land and Hot Zones, it is simply an additional property.

- **Build Hyperjump**: Builds a Hyperjump on a specified location. Hyperjumps are neutral structures that allow astronauts to jump up to three tiles following any of the four main directions (North, South, East and West) without breaking their suit and avoiding obstacles and impassable tiles in between. Building a Hyperjump can only be performed in tiles where there is no other structure and if the astronaut is equipped with a Hyperjump package.

- **Build Settlement**: Builds a Settlement on a specified location. Settlements are structures that can store oxygen and care packages, and can also recruit astronauts in the same way as the HQ does. However they don't have natural oxygen production. Building a settlement can only be performed in tiles where there is no other structure or astronaut, and if the astronaut is equipped with a Settlement package.

- **Build Dome**: Builds a Dome on a specified location. Domes are neutral structures that decrease the movement cooldown of all astronauts at distance 25 from it by half. Building a Dome can only be performed on tiles where there is no other structure and if the astronaut is equipped with a Dome package.

- **Broadcast**: The astronaut broadcasts an integer value. These values can be freely read by other allied units anywhere on the map.

- **Retrieve:** Retrieves an adjacent Care Package. This package is sent directly to its parent structure (see Section 8).

- **Transfer oxygen:** Transfers all oxygen to the specified unit.

HQs and settlements can broadcast in the same way as astronauts (without dying), and can also perform an **enlist** action. This action enlists a new astronaut with a given amount of oxygen. HQs and settlements also have the option of equipping the enlisted astronaut with a care package (if available).

8   Parent Structures

The parent of a unit is defined as follows. The HQ's parent is itself. The parent structure of an astronaut is the structure in which that astronaut was enlisted. The parent structure of a Settlement is the parent of the astronaut that created it. If a settlement s is destroyed, all units that had s as a parent update their parent to the parent of s. If an HQ h is destroyed, all units that had h as a parent update their parent to another alive HQ. Note that this definition guarantees that all units have a parent at all times.

9   Care Packages

Care Packages are resources that can be found around the map and allow astronauts that equip them to perform additional actions. These actions are listed below:

- **Settlement Package:** Allows the astronaut to build a Settlement.

- **Dome Package:** Allows the astronaut to build a Dome.

- **Hyperjump Package:** Allows the astronaut to build a Hyperjump.

- **Radio:** Whenever the astronaut broadcasts, its space suit loses half of the remaining oxygen (rounded up) instead of breaking.

- **Reinforced Suit:** Whenever the astronaut sabotages or is sabotaged, the suit loses half of the remaining oxygen (rounded up) instead of breaking.

- **Survival Kit:** The oxygen that the astronaut loses at the end of each turn is reduced by half (this stacks with terraformed tiles).

There are two additional Care Packages - **Oxygen Tank** and **Plants** - which cannot be equipped. Instead, whenever an astronaut retrieves an oxygen tank, its parent structure gains 200 oxygen. When an astronaut retrieves plants, its parent structure gains 1 additional oxygen production per turn.

At each round of the game, each tile has some probability p that the sponsors decide to drop one of these care packages. This probability p depends on the type and is 10 times larger in Hot Zones. The probabilities are given below (in %).

Settlement Package	0.0005
Dome Package	0.0017
Hyperjump Package	0.002
Radio	0.008
Reinforced Suit	0.01
Medicine	0.01
Oxygen Tank	0.015
Plants	0.015

10   Communication and Vision

Each unit can only sense the objects (units, tiles, etc.) that are inside its vision radius. Vision is not shared. This means that objects detected by a given unit might not be detected by others.

Units run independently and they don't share memory. However, units can communicate using broadcasts. Whenever a unit broadcasts an integer, each other allied unit stores it in a buffer. Units can query the oldest message and clean the buffer (among other functionalities) using the methods provided in the *UnitController* class.

11   Energy

Energy is an approximate indicator of the number of basic instructions that each unit or structure can perform during each round. More precisely, each bytecode instruction performed by a given unit or structure consumes one unit of energy (except internal operations of the methods provided in the documentation, these consume a constant amount of bytecode which is specified there). For users not familiarized with Java, it is not necessary to know how bytecode works, however it is good to have in mind that it is somewhat proportional to the number of code instructions. The amount of energy consumed up to a certain instruction can be accessed at any time using the methods in *UnitController*. Whenever a unit or structure surpasses the amount of energy allowed (currently set to 15000 for units and 100000 for structures), the unit pauses and continues running the remaining instructions during its next turn.

12   User Instructions

Players must fill the *run* method of the *UnitPlayer* class, which is run independently by all units and structures of the player's team. This function has a *UnitController* as input, which is used to give orders to the given unit/structure and to get information of the visible tiles, among others (check the documentation for more details).

Run does the following: Whenever a new unit or structure is created, their *run* method is executed until either the unit finishes its turn by calling the *yield* method, or when it surpasses the amount of allowed energy. If a unit or structure returns from its run method, it dies. Because of this it is suggested to keep its instructions inside a *while(true)* statement, and to finish each iteration with a call to *yield* (check *nullplayer* and *demoplayer*).

13   Implementation Info

You may skip this section if you're not enough familiarized with Java.

Each unit and structure is run in an independent thread. Each of these threads gets reactivated every time the unit/structure is scheduled (once every round, following the same relative order every round). For safety reasons, it is forbidden to use any method or class outside *java/lang*, *java/math* and *java/util* (and some of the sub-classes of these). It is also forbidden to use static variables (which include switch statements, since they internally do so).

Even though codes that break these rules are automatically detected the instrumenter, we are going to check all the finalists' codes manually to be sure that everyone is playing a fair game. If a team breaks any of these rules without informing any of the devs, it will be disqualified.