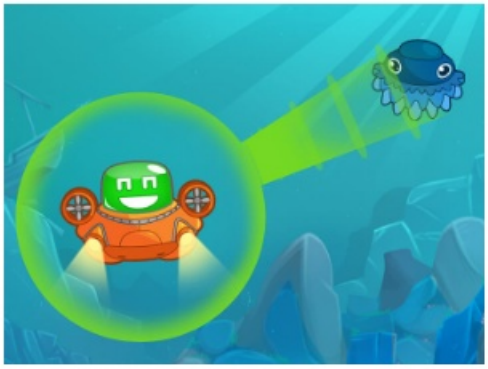
This challenge is based on a system of leagues.

For this challenge, multiple leagues for the same game will be available. Once you have proven your worth against the first Boss, you will access the higher league and unlock new opponents.

Goal

Win more points than your opponent by **scanning the most fish**.



To protect marine life, it is crucial to understand it. Explore **the ocean floor** using your drones to scan as many fish as possible to better understand them!

Rules

The game is played turn by turn. Each turn, each player gives an action for their drones to perform.

The Map

The map is a **square** of `10,000` units on each side. Length units will be denoted as "`u`" in the rest of the statement. The coordinate `(0, 0)` is located at the **top left** corner of the map.

Drones

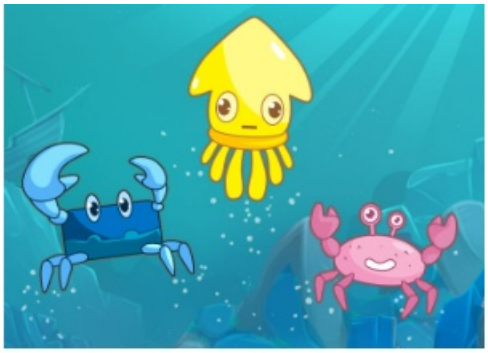
Each player has two drones to explore the ocean floor and scan the fish. Each turn, the player can decide to move their drone in a direction or not activate its motors.

Your drone continuously emits light around it. If a fish is within this **light radius**, it is automatically scanned. You can increase the **power** of your light (and thus your scan radius), but this will drain your **battery** .

In order to **save your scans** and score points, you will need to resurface with your drone.

Fish

On the map, different fish are present. Each fish has a specific `type` and `color`. In addition to the points earned if you scan a fish and bring the scan back to the surface, **bonuses** will be awarded if you scan all the fish of the **same type** or **same color**, or if you are the **first** to do so.



Each fish moves within a **habitat zone**, depending on its **type**. Only fish within the **light radius** of one of your drones will be visible to you.

Unit Details

Drones

Drones move towards the given point, with a maximum distance per turn of `600u`. If the **motors** are not activated in a turn, the drone will **sink** by `300u`.

At the end of the turn, fish within a radius of `800u` will be **automatically scanned**.

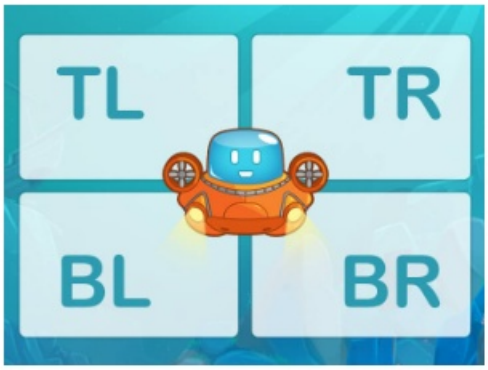
If you have increased the **power of your light**, this radius becomes `2000u`, but the **battery** drains by `5` points. If the powerful light is not activated, the battery recharges by `1`. The battery has a capacity of `30` and is **fully charged** at the beginning of the game.

If the drone is near the **surface** ($y \leq$ `500u`), the scans will be automatically saved, and points will be awarded.

Radar

To better navigate the dark depths, drones are equipped with **radars**. For each **creature** (fish or monster) in the game zone, the radar will indicate:

- **TL:** if the entity is somewhere **top left** of the drone.
- **TR:** if the entity is somewhere **top right** of the drone.
- **BR:** if the entity is somewhere **bottom right** of the drone.
- **BL:** if the entity is somewhere **bottom left** of the drone.



Note: If the entity shares the same x-coordinate as the drone, it will be considered as being on the left. If the entity shares the same y-coordinate as the drone, it will be considered as being on the top.

Fish

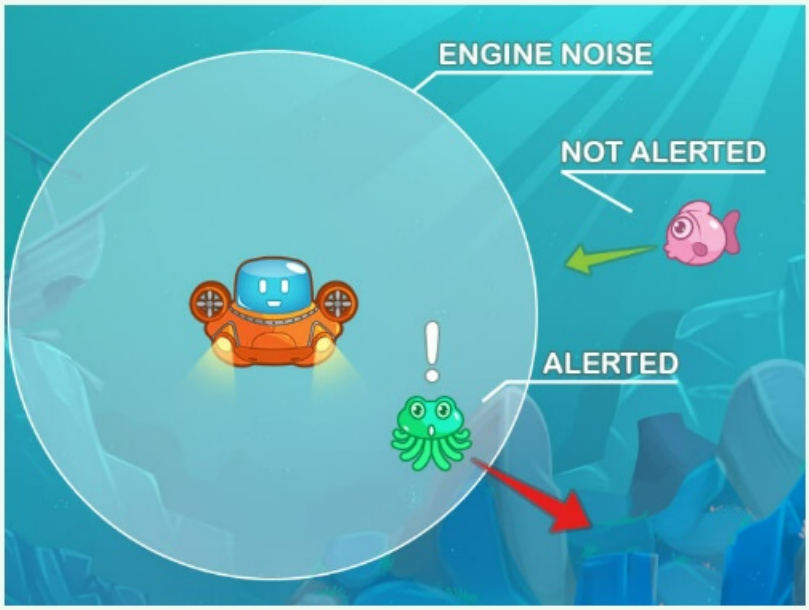
Fish move `200u` each turn, in a randomly chosen direction at the beginning of the game. Each fish moves within a habitat zone based on its type. If it reaches the edge of its habitat zone, it will **rebound** off the edge.

Fish	type	Y min	Y max
	0	2500	5000
	1	5000	7500

2 7500 10000

If a fish comes within 600u of another, it will begin to swim in the opposite direction to the nearest fish.

If a drone has its **motors activated** within a distance of less than 1400u, the fish will enter **"frightened" mode** in the next turn: in this mode, the fish will start swimming in the direction opposite to the nearest drone at a speed of 400u per turn. While frightened, the fish cannot **exit** its habitat on the y-coordinate (it will stay at that y-coordinate without bouncing), but if its x-coordinate becomes negative or greater than 9999, it will **permanently leave** the map and cannot be scanned anymore.



Score Details

Points are awarded for **each scan** depending on the type of scanned fish. Being the **first** to save a **scan** or a **combination** allows you to earn **double** the points.

Scan	Points	Points if first to save
Type 0	1	2
Type 1	2	4
Type 2	3	6
All fish of one color	3	6
All fish of one type	4	8

At the end of the game, **all unsaved scans** are **automatically saved**, and associated points are awarded.

Victory Conditions

- The game reaches 200 turns
- A player has earned enough points that their opponent cannot catch up
- Both players have saved the scans of **all remaining fish** on the map

Defeat Conditions

- Your program does not return a valid command within the given time for each of your

drones.

Debugging Tips

- Hover over an entity to see more information about it.
- Add text at the end of an instruction to display that text above your drone.
- Click on the gear icon to display additional visual options.
- Use the keyboard to control actions: space for play/pause, arrows for step-by-step forward movement.

Game Protocol

Initialization Input

First line: `creatureCount` an integer for the number of creatures in the game zone. Will always be `12`.
Next `creatureCount` **lines:** 3 integers describing each creature:

- `creatureId` for this creature's unique id.
- `color` `(0 to 3)` and `type` `(0 to 2)`.

Input for One Game Turn

Next line: `myScore` for you current score.
Next line: `foeScore` for you opponent's score.

Next line: `myScanCount` for your amount of **saved** scans.
Next `myScanCount` **lines:** `creatureId` for each scan scored.

Next line: `foeScanCount` for your opponent's amount of **saved** scans.
Next `foeScanCount` **lines:** `creatureId` for each scan scored by your opponent.

Next line: `myDroneCount` for the number of drones you control.
Next `myDroneCount` **lines:**

- `droneId`: this drone's unique id.
- `droneX` and `droneY`: this drone's position.
- `emergency`: unused in this league.
- `battery`: this drone's current battery level.

Next line: `foeDroneCount` for the number of drones your opponent controls.
Next `foeDroneCount` **lines:**

- `droneId`: this drone's unique id.
- `droneX` and `droneY`: this drone's position.
- `emergency`: unused in this league.
- `battery`: this drone's current battery level.

Next line: `droneScanCount` for the amount of scans currently within a drone.
Next `droneScanCount` **lines:** `droneId` and `creatureId` describing which drone contains a scan of which fish.

Next line: `visibleCreatureCount` the number of creatures within the light radius of your drones.
Next `visibleCreatureCount` **lines:**

- `creatureId`: this creature's unique id.
- `creatureX` and `creatureY`: this creature's position

- creatureX and creatureY: this creature's position.
- creatureVx and creatureVy: this creature's current speed.

Next line: radarBlipCount.
Next radarBlipCount lines: Two integers droneId, creatureId and a string radar indicating the relative position between each creature and each one of your drones. radar can be:

- TL: the creature is to the top-left of the drone.
- TR: the creature is to the top-right of the drone.
- BR: the creature is to the bottom-right of the drone.
- BL: the creature is to the bottom-left of the drone.

Output

Next myDroneCount lines: one valid instruction for each of your drones, in the same order the drones were provided to you:

- MOVE x y light: makes the drone move towards (x,y), engines on.
- WAIT light. Switches engines off. The drone will sink but can still use light to scan nearby creatures.

Set light to 1 to activate the powerful light, 0 otherwise.

Constraints

creatureCount = 12 in this league
myDroneCount = 2

Response time per turn ≤ 100 ms
Response time for the first turn ≤ 1000 ms

What awaits me in the following leagues?

- Deep sea monsters will roam

To Start

Why not start the battle with one of these **IA Starters**, provided by the team:

- C++ https://gist.github.com/CGjupoulton/7da72cf7298ca5faaded639dbde5be33
- JavaScript https://gist.github.com/CGjupoulton/8dda912e519671d440b8929e907e603a
- Java https://gist.github.com/CGjupoulton/0add0a53e404f7373025bb952bc67cc2
- Python https://gist.github.com/CGjupoulton/799531754313104f0485f112324251ec
- Ruby https://gist.github.com/CGjupoulton/8cf4c3ca01c8405e840e7f55325d1fe4
- TypeScript https://gist.github.com/CGjupoulton/bfad2283896c08a7f6c8203040fc486f

You can modify them to match your style or take them as an example to code everything from scratch.