Goal

End the game with more crystal than your opponent.

Crystal

The game takes place in a **lab**, in which two scientists in charge of **robot ants** are competing to find the most efficient way of gathering crystals.

However, the ants **cannot be controlled directly**. The ants will respond to the presence of **beacons**.

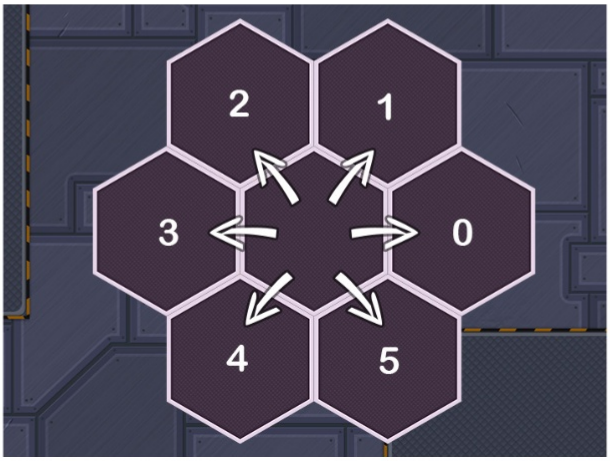
Rules

The game is played in turns. On each turn, both players perform any number of actions simultaneously.

The Map

On each run, the map is **generated randomly** and is made up of **hexagonal cells**.

Each cell has an **index** and up to six neighbors. Each direction is labelled `0` to `5`.



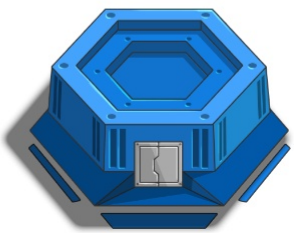
Hex directions

Each cell has a **type**, which indicates what the cell contains:

- `0` if it does not contain a resource
- `1` if it contains the **egg** resource
- `2` if it contains the **crystal** resource

The amount of resources contained in each cell is also given, and is subject to change during the game as the ants **harvest** cells.

A cell may also have a **base** on it. The players' ants will start the game on these bases.



Blue base *Egg*

Ants & Beacons

Both players start with several ants placed on their **bases**. The players cannot move the ants directly but can place **beacons** to affect their movement.

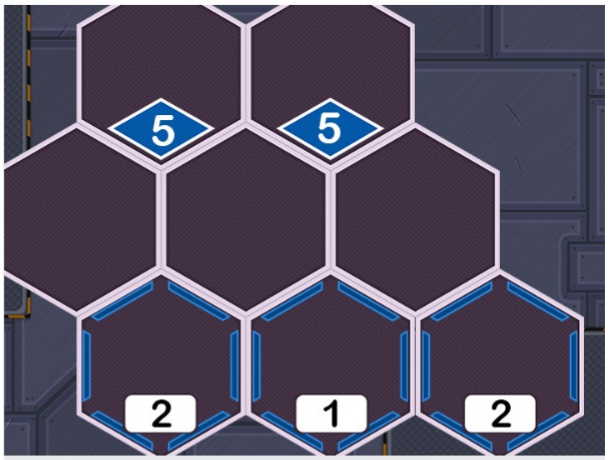
Players can place **any number** of beacons per turn but can only place **one each per cell**.

When placing a beacon, players must give that beacon a strength. These beacon strengths act as **weights**, determining the **proportion of ants** that will be dispatched to each one.

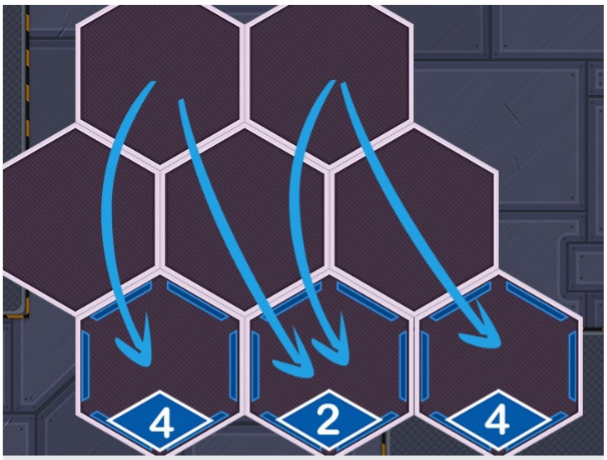
In other words, the **higher** the beacon strength, the greater the **percentage** of your ants that will be sent to that beacon.

Example

In the following example, there are three beacons of strength 2, 1, and 2.



White numbers in a colored diamond represent the ants. Here, 10 ants in total will be dispatched to the beacons.



The 10 ants will move to the three beacons, keeping the same proportions as the beacon strengths.

The ants will do their best to take the **shortest paths** to their designated beacons, moving at a speed of **one cell per turn**.

In between turns, the **existing beacons** are powered down and **removed from play.**

Use beacons to place your ants in such a way to create **harvesting chains** between your **bases** and a **resource**.

Harvesting Chains

In order to harvest **crystal** and score points, there must be an **uninterrupted chain** of **cells containing your ants** between the resource and your **bases**.

The amount of crystal harvested per turn is equal to the **weakest link** in the chain. In other words, it is the smallest amount of ants from the cells that make up the chain.

Here, the blue player will harvest 4 crystal per turn.

In games with multiple bases per player, the game will choose the best chain to either one of your bases.

The harvesting chains work the same way for the **egg resource**.

Harvesting an egg cell will spawn as many ants as resources havested. The ants will spawn on the player's base on the start of next turn

In games with multiple bases per player, the extra ants will spawn on **each base**, regardless of the base present in the harvest chain.

Harvesting is calculated separately for **each resource**, and for each one the game will automatically choose the **best chain** from its cell to your base.

Attack Chains

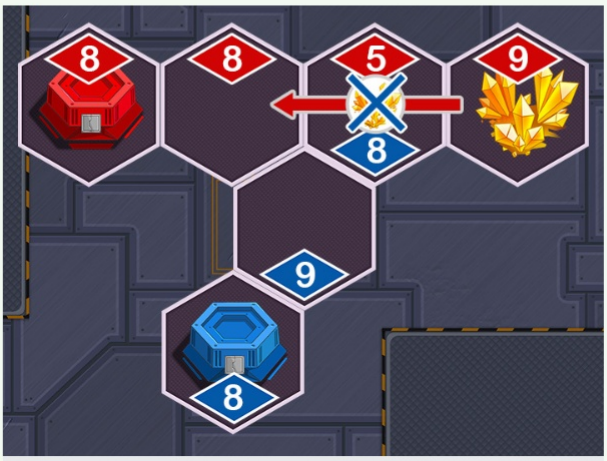
A player's harvest chains may be **broken** by their opponent's **attack chains**.

When computing harvest chains, some cells may have ants from both players. For each of these cells, the **attack chain** of both players is computed, and if one of the player has a lower value, this cell cannot be counted in the harvest chain.

The **attack chain value** for a given cell is the **weakest link** in a chain from that cell to one of the player's bases.

Example

The attack chains for the contested cell are: 5 for the red player and 3 for the blue player.
The harvest chain is unbroken.



The attack chains for the contested cell are: 5 for the red player and 8 for the blue player.
The harvest chain is broken

Actions

On each turn players can do any amount of valid actions, which include:

- `BEACON` `index` `strength`: place a beacon of strength `strength` on cell `index`.
- `LINE` `index1` `index2` `strength`: place beacons all along a path from `index1` to `index2`, all of strength `strength`. A shortest path is chosen automatically
- `WAIT`: do nothing
- `MESSAGE` `text`. Displays text on your side of the HUD.

Action order for one turn

1. `LINE` actions are computed
2. `BEACON` actions are computed
3. Ants move.
4. Eggs are harvested and new ants spawn.
5. Crystal is harvested and points are scored.

Note: when two players harvest from the same resource, they will both receive the full expected amount regardless of whether there is enough resource to support it.

Victory Conditions

- You have harvested at least half of the total **crystal** on the map before your opponent.
- You have more **crystal** than your opponent after `100` turns, or more **ants** if tied.

Defeat Conditions

Your program does not provide a command in the allotted time or it provides an unrecognized command

Debugging tips

- Hover over a tile to see extra information about it, including beacon `strength`.
- Use the `MESSAGE` command to display some text on your side of the HUD.
- Press the gear icon on the viewer to access extra display options
- Use the keyboard to control the action: space to play/pause, arrows to step 1 frame at a time.

Game Protocol

Initialization Input

First line: `numberOfCells` an integer for the amount of cells in the map.
Next `numberOfCells` **lines:** the cells, ordered by `index`. Each cell is represented by `8` space-separated integers:

- `type`: `1` for egg, `2` for crystal, `0` otherwise
- `initialResources` for the amount of crystal/egg here.
- `6` `neigh` variables, one for each **direction**, containing the index of a neighboring cell or `-1` if there is no neighbor

Next line: one integer `numberOfBases` containing the number of bases for each player.
Next line: `numberOfBases` integers for the cell indices where a **friendly base** is present
Next line: `numberOfBases` integers for the cell indices where an **opponent base** is present.

Input for One Game Turn

Next `numberOfCells` **lines:** one line per cell, ordered by `index`. `3` integers per cell:

- `resources`: the amount of crystal/eggs on the cell.
- `myAnts`: the amount of ants you have on the cell.
- `oppAnts`: the amount of ants your opponent has on the cell.

Output

All your actions on one line, separated by a `;`

- `BEACON` `index` `strength`. Places a beacon that lasts one turn
- `LINE` `index1` `index2` `strength`. Places beacons along a path between the two provided cells.
- `WAIT`. Does nothing
- `MESSAGE` `text`. Displays text on your side of the HUD.

Constraints

`1` ≤ `numberOfBases` ≤ `2`
`numberOfCells` < `100`
Response time per turn ≤ `100` ms
Response time for the first turn ≤ `1000` ms

StarterKit

Starter AIs are available in the Starter Kit. They can help you get started with your own bot. You can modify them to suit your own coding style or start completely from scratch.

Source code

The game's source will be available here.