Goal

Control more **patches** than your opponent at the end of the match.

Rules

Robots are deployed in a field of abandoned electronics, their purpose is to refurbish patches of this field into functional tech.

The robots are also capable of self-disassembly and self-replication, but they need **raw materials** from structures called **Recyclers** which the robots can build.

The structures will **recycle** everything around them into raw matter, essentially removing the patches of electronics and revealing the **Grass** below.

Players control a **team** of these robots in the midst of a **playful competition** to see which team can control the most patches of a given scrap field. They do so by **marking** patches with their team's color, all with the following constraints:

- If robots of both teams end up on the same patch, they must disassemble themselves **one for one**. The robots are therefore removed from the game, only leaving at most **one** team on that patch.
- The robots **may not cross the grass**, robots that are still on a patch when it is completely recycled must therefore disassemble themselves too.

Once the games are over, the robots will dutifully re-assemble and go back to work as normal.

A blue-team robot.   *A red-team robot.*

Map

The game is played on a grid of variable size. Each tile of the grid represents a patch of scrap electronics. The aim of the game is to control more tiles than your opponent, by having robots **mark** them.

Each tile has the following properties:

- `scrapAmount` : this patch's amount of usable scrap. It is equal to the amount of turns it will take to be completely recycled. If zero, this patch is **Grass**.
- `owner` : which player's team controls this patch. Will equal **-1** if the patch is neutral or **Grass**.

Robots

Any number of robots can occupy a tile, but if units of opposing teams end the turn on the same tile, they are removed **1 for 1**. Afterwards, if the tile still has robots, they will mark that tile.

After moving all robots to the middle tile, only one blue robot remains and the tile is marked.

Robots may not occupy a **Grass** tile or share a tile with a **Recycler**.

Recyclers

Recyclers are structures that take up a tile. Each turn, the tile below and all adjacent tiles are used for recycling, reducing their `scrapAmount` and providing **1** unit of matter to the recycler's owner.

If the tile under a recycler runs out of scrap, the recycler is dismantled.
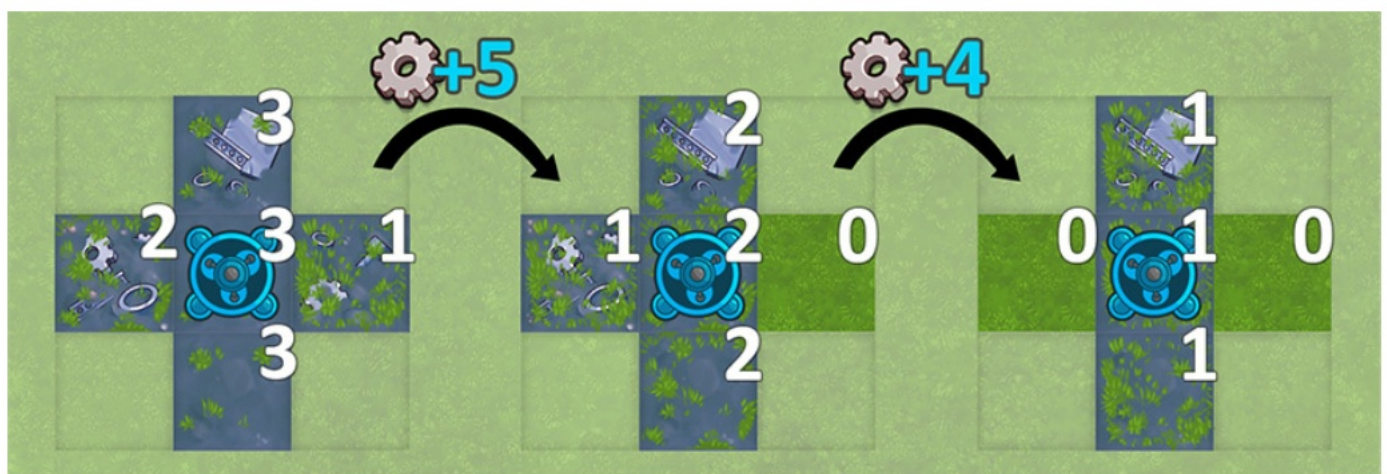
Any tile within reach of your recyclers will grant 1 matter per turn and their `scrapAmount` will decrease.

A given tile can only be subject to recycling **once** per turn. Meaning its `scrapAmount` will go down by **1** even if a player has **multiple** adjacent Recyclers, providing that player with only **1** unit of matter. If a tile has adjacent Recyclers from **both** players, the same is true but both players will receive **1** unit of matter.

Matter

10 units of matter can be spent to create a new robot, or to build another **Recycler**.
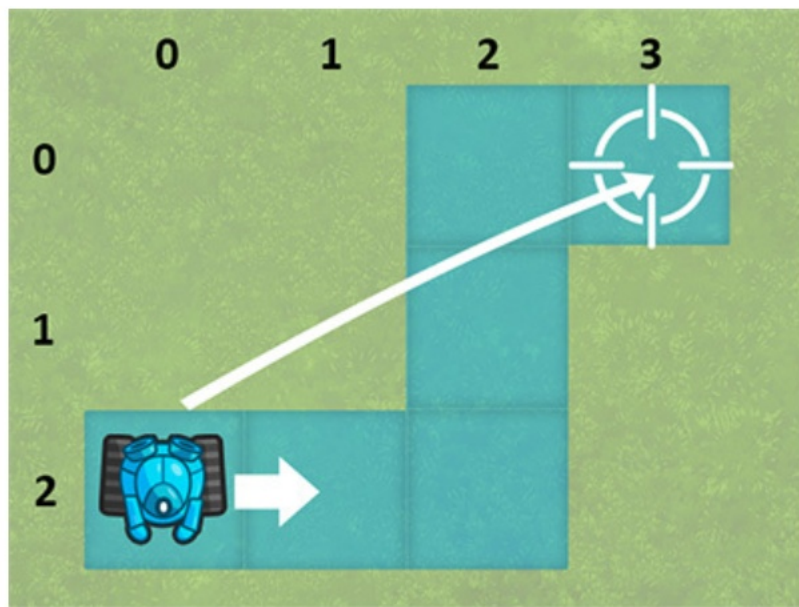
At the end of each turn, both players receive an extra **10** matter.

Actions

On each turn players can do any amount of valid actions, which include:
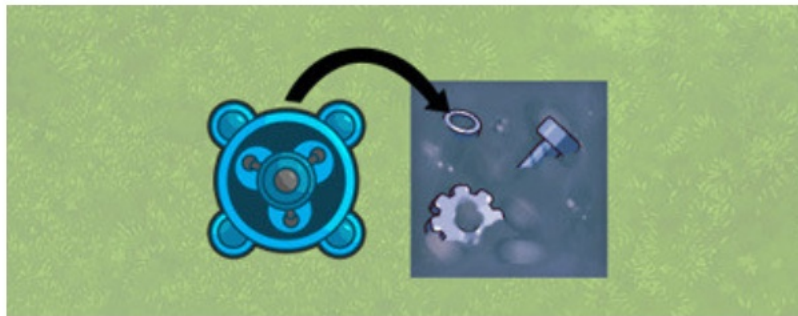
- **MOVE**: move a number of units from a tile to an adjacent tile. You may specify a non adjacent tile to move to, in which case the units will automatically select the best MOVE to approach the target.
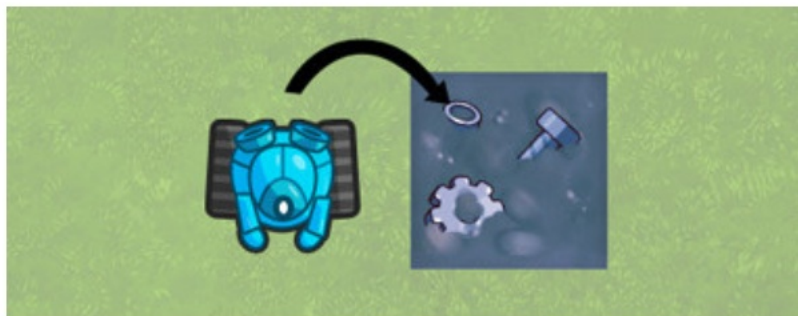
A MOVE *to* **(3,0)** *will result in this robot stepping into* **(1,2)**.

- BUILD : erect a Recycler on the given empty tile the player controls.

- SPAWN : construct a number of robots on the given tile the player controls.

Action order for one turn

1. BUILD actions are computed.
2. MOVE and SPAWN actions are computed simultaneously. A robot cannot do both on the same turn.
3. Units of opposing teams on the same tile are removed one for one.
4. Remaining robots will mark the tiles they are on, changing their owner .
5. Recyclers affect the tiles they are on and the 4 adjacent tiles that are not Grass.
6. Tiles with size 0 are now **Grass**. Recyclers and robots on that tile are removed.
7. The players receive 10 base matter as well as the matter from recycling.

Victory Conditions

The winner is the player who controls the most
tiles after either:

- A player no longer controls a single tile.
- **20** turns have passed without any tile changing `scrapAmount` or `owner`.
- **200** turns have been played.

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 it's **history**.
- 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.

Technical Details

- A tile's `owner` will not change if there are no robots on it at end of turn.
- If the target of a `MOVE` is unreachable, the robots will target the reachable tiles closest to the given destination, preferring the one closest to the center of the map.
- When selecting a path to `MOVE` to a distant tile, the robots will take the shortest route, preferring to stay near the center of the map when possible.
- `MOVE` and `SPAWN` happen simultaneously and cannot conflict with each other. However, they may be cancelled by a `BUILD` action, even if it comes later in the player's output, or is part of the opponent's actions.

Game Protocol

Initialization Input

One line: two integers `width` and `height` for the size of the map. The top-left tile is **(x,y) = (0,0)**.

Input for One Game Turn

First line: two integers `myMatter` and `oppMatter` for the amount of matter owned by each player.
Next `height` **lines:** one line per cell, starting at **(0,0)** and incrementing from left to right, top to bottom. Each cell is represented by **7** integers:

The first **4** variables describe properties for this tile:

- **scrapAmount**: the number of times this tile can be recycled before becoming **Grass**.
- **owner**:
  - **1** if you control this cell.
  - **0** if your opponent controls this cell.
  - **-1** otherwise.
- **units**: the number of units on this cell. These units belong to the **owner** of the cell.
- **recycler**: **1** if there is a recycler on this cell. This recycler belongs to the **owner** of the cell. **0** if there is no recycler on this cell.

The next **3** variables are helper values:

- **canBuild**: **1** if you are allowed to **BUILD** a recycler on this tile this turn. **0** otherwise.
- **canSpawn**: **1** if you are allowed to **SPAWN** units on this tile this turn. **0** otherwise.
- **inRangeOfRecycler**: **1** if this tile's **scrapAmount** will be decreased at the end of the turn by a nearby recycler. **0** otherwise.

Output

All your actions one one line, separated by a **;**

- **MOVE** **amount** **fromX** **fromY** **toX** **toY**. Automatic pathfinding.
- **BUILD** **x** **y**. Builds a recycler.
- **SPAWN** **amount** **x** **y**. Adds unit to an owned tile.
- **WAIT**. Does nothing.
- **MESSAGE** **text**. Displays text on your side of the HUD.

Constraints

12 ≤ **width** ≤ **24**
6 ≤ **height** ≤ **12**
Response time per turn ≤ **50** ms
Response time for the first turn ≤ **1000** ms

Keep Off The Grass!

The life of a **Recyclo-Bot** is a simple one.

Mark scrap for refurbishment, build recyclers, move on to the next field of scrap and repeat, all while respecting the Prime Directive: **"Keep Off The Grass"**. But sometimes, even the most cheerful little **Recyclo-Bot** can get a bit bored by these repetitive tasks.

This is why, once in a while, the self-proclaimed **Recyclo-Boyz** like to organize the **Great Scrap Marking Competition**, a friendly joust between two teams where the one having marked the most scrap with their color at the end of a timer is declared the winner.

However, during a match the robots may only use raw materials recycled from the scrap field they are standing on! All tricks are allowed, even recycling to such an extent that the honoured **Grass** is uncovered, blocking off a patch of scrap from the opponent... or completely pulling the rug out from under oneself, if not careful enough.

Starter Kit

Starter AIs are available in the .... 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.