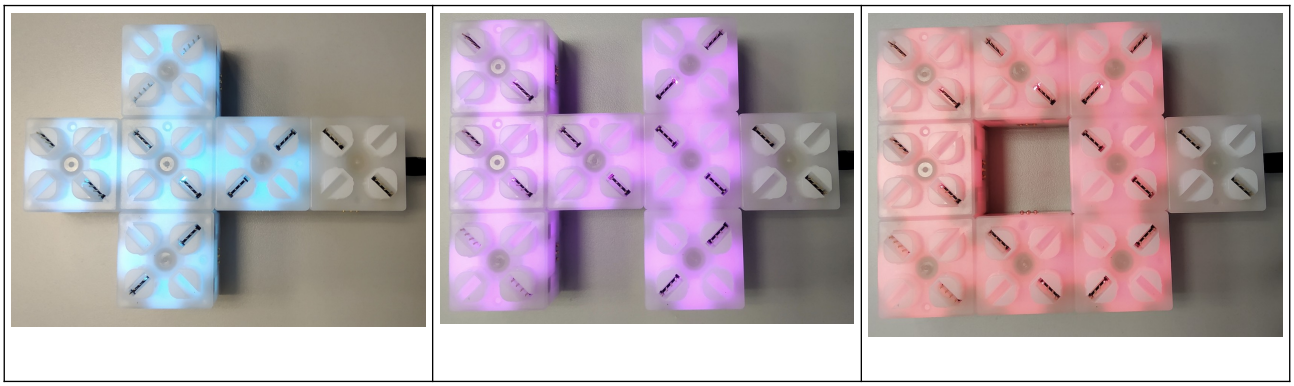
Modular robots programming and swarm robotics

Distributed shape recognition

This exercise will be about the blocks **collectively** and **dynamically** recognize the shape of the ensemble.

A limited number of shapes are to be recognized, and once a shape is recognized, a corresponding color should be taken by all to blocks in the ensemble.

The minimal set of shapes to be recognized is the following (you can add more if you want):



Constrains:

- Only one code:
  - All the blocks contains the same code
  - All the shapes are recognized by the same code
- The code should be made to run automatically (see -w parameter for the appLoader) so that you can add and remove blocks from your ensemble without having to send a "-j" command.
- Each time you add or remove a block from your ensemble, they shall try to find if they form a recognizable shape. If not, they shall turn to a default color (dark blue)

To make it simpler, you can put a restriction on the orientation of the blocks. They shall all face the same direction.

Steps

This problem of shape recognition is tricky and a step-by-step approach is recommended.

Step 1 - Detecting that a new neighbors has been added

Your code should detect changes in its neighbors by monitoring all 6 communications ports, and blink a few times when it detects a change.
Blink red when a block is removed
Blink green when a block is added

Step 2 – Detecting and advertising a change in the ensemble
Any block detecting a change in its neighbors should propagate this information to all the other blocks (you can use a simple flooding protocol).

Depending on the type of an event (addition or removal), all block receiving this message should momentarily blink in the corresponding color.

When doing a flooding, you have to be careful that you do not form propagations loops ! You can add a unique identifier (randomly selected) to your message so that your blocks can tell if they have already seen it (and should discard it) or not (and should forward it).

Step 3 – Trying to recognize the shape

Many different approach could work here …
We recommend to following simplifications:

- You have only one source of power (only one set of blocks is powered at a given time). You don't have to deal with the merging of independents sets of blocks.
- You can start the detection algorithm at the block that detected a new neighbor (it prevents all the block trying to build and image of the ensemble at the same time). Be careful however, when you add multiple blocks at once, you may have multiple starting points.