Project: Develop simulator for models of simple programmable matter

Description: An active area of research at the intersection of theoretical computer science, nanotechnology, and robotics is "programmable matter": the design of smart materials that autonomously self-organize to create and re-configure structures made of many small independently operating devices. Simulation software is essential to explore and understand the capabilities and limitations of such devices. In a simplified theoretical model of such devices, we imagine each device existing on a two-dimensional surface, able to bind to each other, communicate with their immediate neighbors, and move relative to each other (by rotating around each other, for instance), all asynchronously.

The project is to create simulation software that enables the user to specify what program each device is running, to show a two-dimensional representation of the devices, and execute the device's programs and the toy "laws of physics" governing how they interact.

Such a model is abstract and intended to be adaptable to model several scenarios, from molecular-sized DNA walkers (http://science.sciencemag.org/content/357/6356/eaan6558) to macro-sized robots such as kilobots (http://science.sciencemag.org/content/345/6198/795). Therefore, the model and software should have user-controllable parameters, defining modeling choices such as

- **the two-dimensional geometry**: do the devices move continuously in 2D space, or discretely on a hexagonal or square grid?

- **the device shape**: if on a hexagonal grid, the precise shape of the device places constraints on what rotations are possible

- **connectivity constraints**: are the devices required to maintain connectivity to each other while moving?

- **invalid move semantics**: if an invalid move is attempted (e.g., moving into an occupied space), is the device able to detect that its attempt failed?

- **communication ability**: is each device able to read the entire memory of its neighbors, or is communication more limited, for example transmitting single symbols from a small alphabet?

Here's my preferences in order:

1. browser-based, with heavy computation done in the browser (using server only to do things like save files/store preferences, or perhaps not even using the server at all and just using localStorage). Programming language preferences in order
   1. Elm
   2. Dart
   3. Something else nicer/more strongly typed than plain Javascript (e.g., Typescript, ECMAScript 8)
2. desktop-based, programming language preferences in order
   1. Rust
   2. Go
   3. Python (be wary that it might be unbearably slow without some tricks like using PyPy, or NumPy or Cython for critical inner loops)
   4. Something functional like OCaml or Haskell