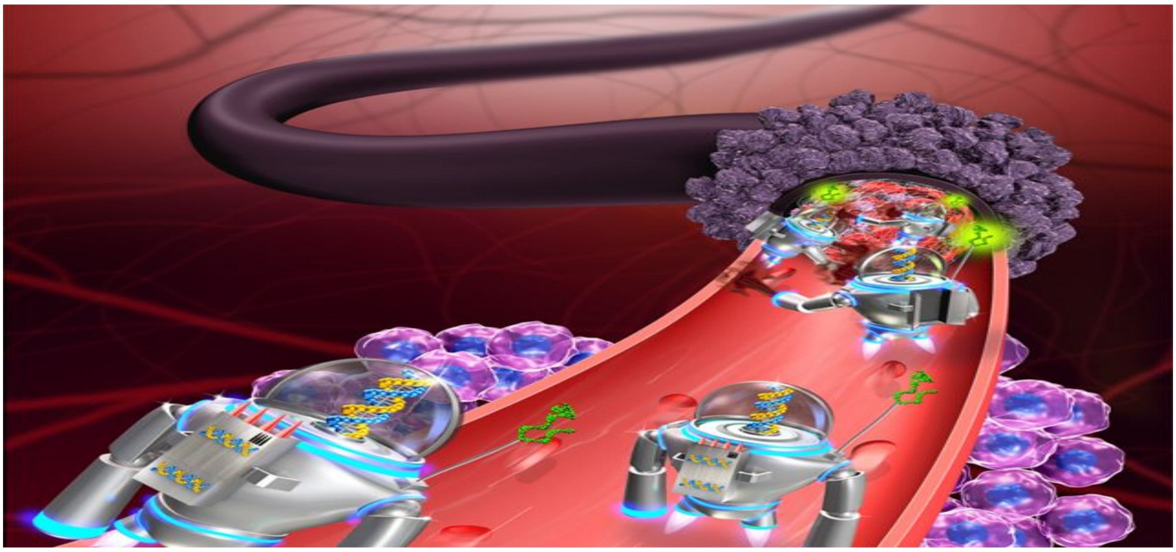
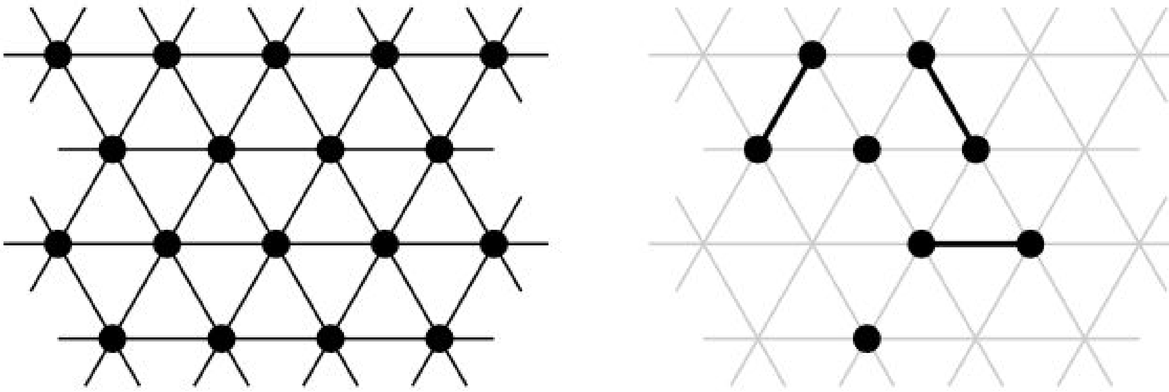


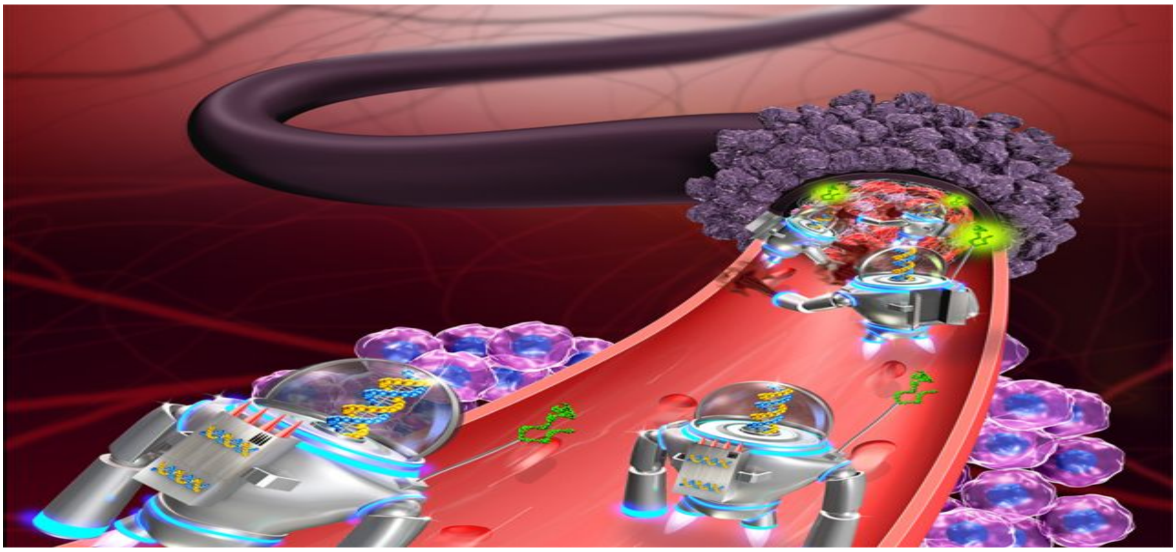
Fish Shape Movement With Programmable Matter:

My goal in this project is to make movement , using programmable matter , as a fish shape disconnected from the outside world and reaching the goal setted without external intervention.The main reason for choosing the fish shape movement is to (in theory) make the movement in fluid smoother and easier due to the fish aerodynamic shape , and i want to mimic it using programmable matter.

Amoebot Model (pictures and general explanation):

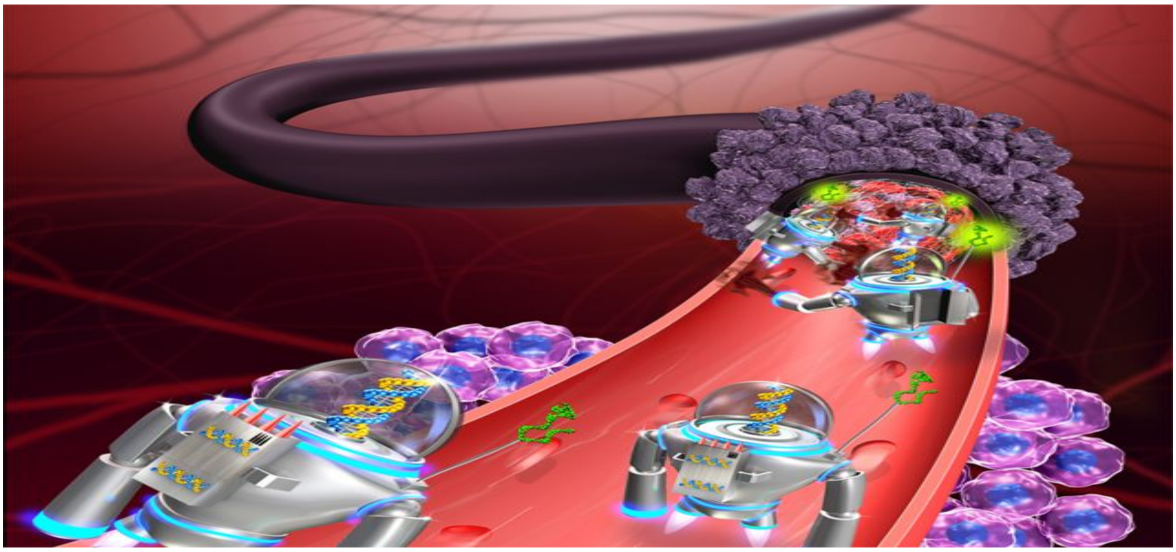


The left half of the figure depicts a section of the infinite equilateral triangular graph. Nodes are shown as black circles. The right half shows five particles on the graph. When depicting particles we draw the graph as a gray mesh without nodes. A particle occupying a single node is depicted as a black circle, and a particle occupying two nodes is depicted as two black circles connected by an edge



Shape organization:
My programmable matters have limited memory due to its size and will be referred as particles.We will be using n amount of particles in this article (the more particles the bigger the fish shape which is not necessary in our demonstration i will be using 8 particles to simplify).
the nanoparticles are injected in **any** arbitrary shape composed of a constant number of equilateral triangles of unit size (it needs to be connected) and using the “Universal Shape Formation for Programmable Matter” (in References) the particles can build that shape in $O(\sqrt{n})$ movements , our algorithm relies only on local information meaning every nanoparticle hold his own state (specified below) and every connected particle (we will call them neighbors) to the gate (specified below) have access to that shared data .

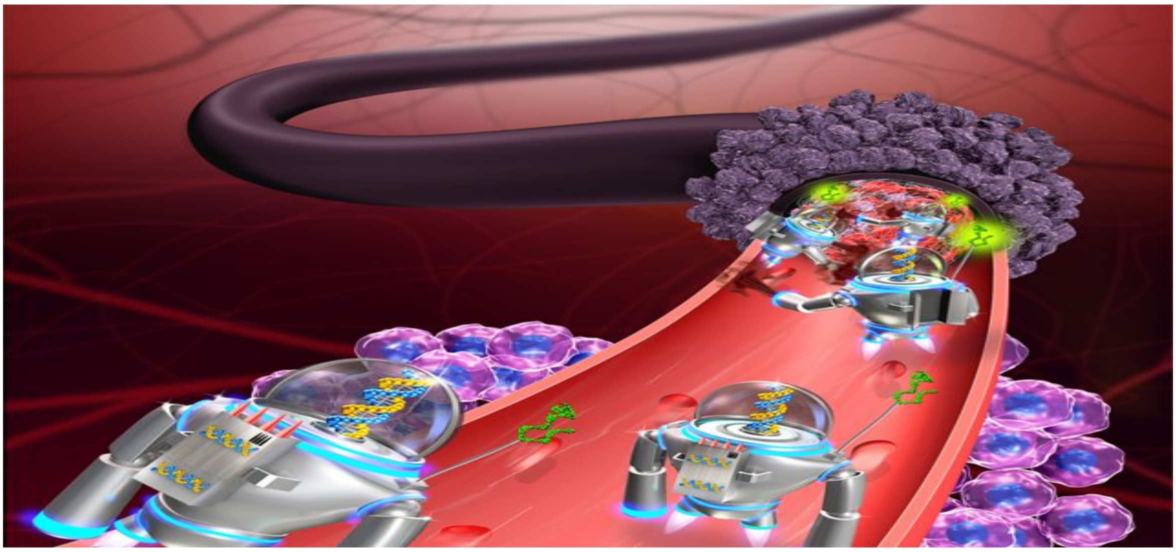
Gate and Link between nanoparticles:
The nanoparticles we are using have six connection gates but in our demonstration 4 are enough , to be connected to the gate the nanoparticle needs to be an adjacent node in the “Amoebot model” as mentioned before , also each nanoparticles have two states:
1) Expanded state - we take two nodes on the graph mentioned above by expanding we first change our shared data to be “1” and then every neighbor nanoparticle can access to our shared data.
2)Contracted state - we take one node on the graph mentioned above by contracting we first set our shared data to be “0” and then we “pull” every neighbor by forcing them to expand (we need to make sure there



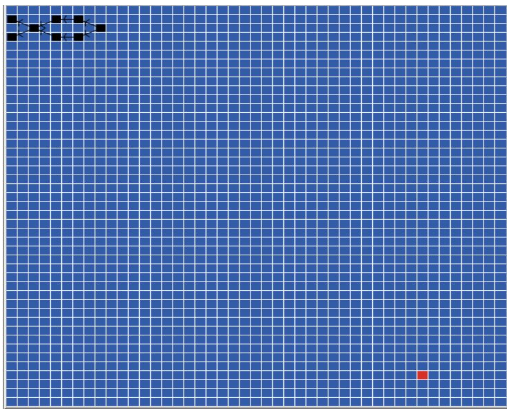
is no neighbor expanded before we do it and this is what the shared data is for).

Leader of the particles:
Now come the more complex part , in addition to the above stats we will use a different nanoparticle for the “head” , the particle who decide where to move and pull the others behind him , there are many options (in the last link) to do that but in my project i decided to use the magnetic swimmer referred in “Undulatory Locomotion of Magnetic Multilink Nanoswimmers” article and the main reason is because of the fast movement.

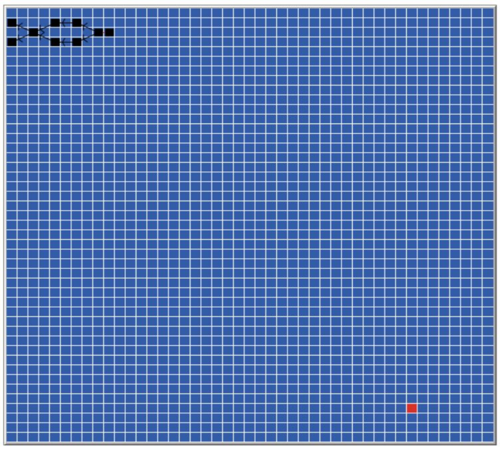
Programming the particles:
We will program the head by giving him the coordinate for his predicted movements to do once he is released in blood and adding a movement condition , if non attached nanorobot hold “1” move else wait and try again (the wait is not mandatory) , we will attach our nanorobots in a “behind” and “forward” method by setting 0-2 gates being the “ahead” and 3-6 the “behind” , also because our “simpler” nanorobots knows the data in their neighbor and they know their own stat we will program them in a way that if an attached neighbor behind you hold “1” you hold “1” as well even if contracted since it has his own data and his neighbor he can change it back to “0” if his neighbor behind him hold “0” that way we insure we will not expand an already expanded nanorobot.



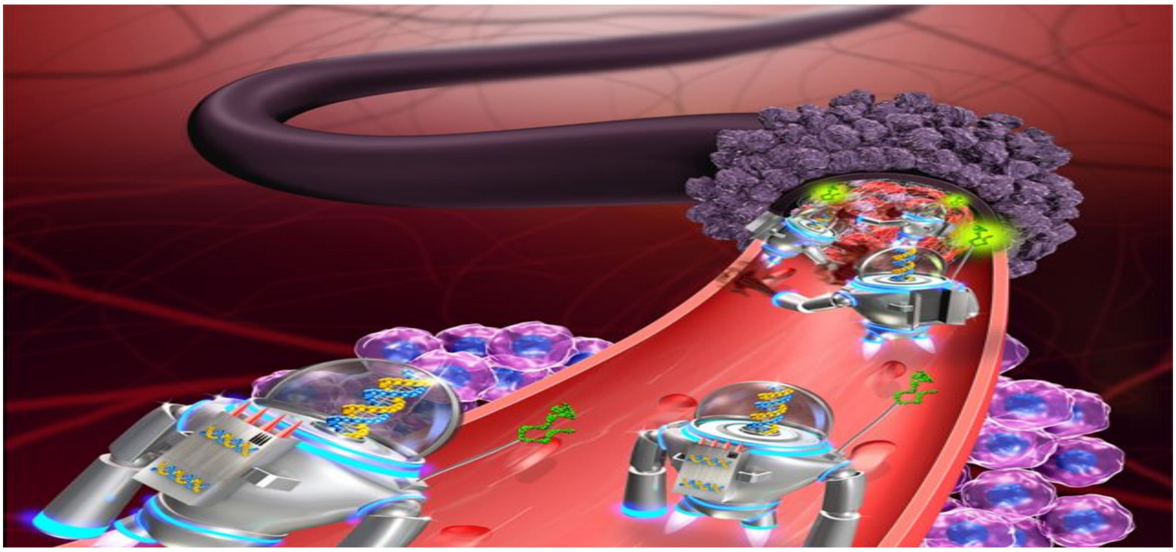
We will now show my simulation:

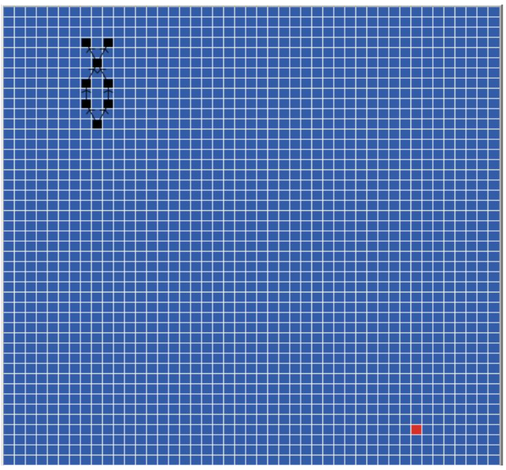


The fish shape of nanorobots is in the top left corner with black links to connect between each particle and the “goal” is in bottom right.

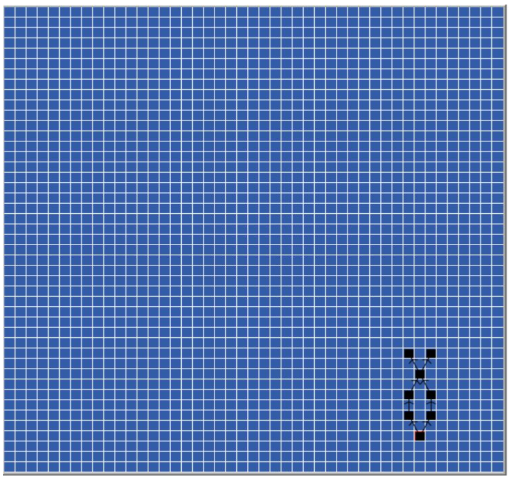


By expanding the head only we can see the link in color “red” is to show that the particle attached is expanded

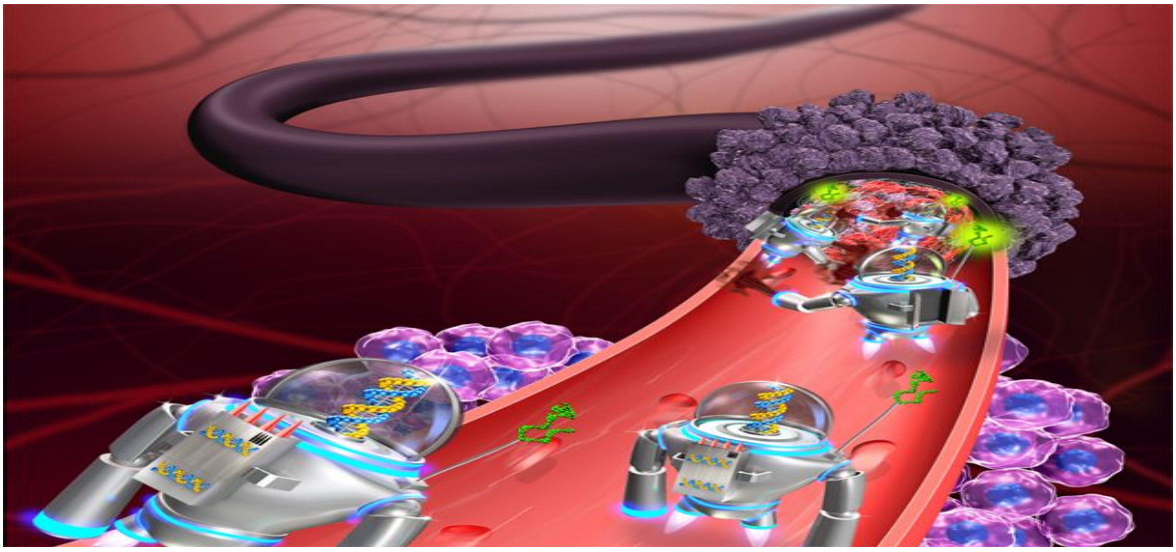




Our fish after making a turn right.



Our fish at “goal”.



Pseudo code:(for general shapes)

move forward:(the ldr is the one moving and pulling the rest after him):

1.check if none hold “1”

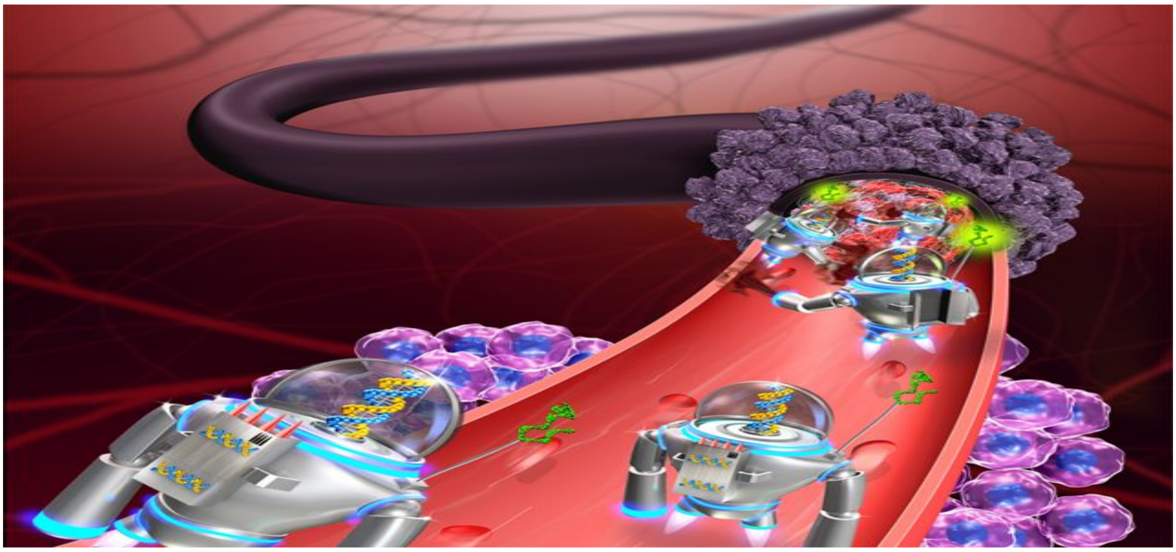
    1.a true -> I) expand ldr to designated point by expanding
        II) shrink ldr and pull connected particles
        III) if arrived to goal stop else repeat 1
    1.b else -> wait and repeat 1 until arrived to “goal”

rotate left/right:(optional theory):

1.check if none hold “1”

    1.a true -> I) expand ldr to the end of the rotation point by expanding n times
        II) shrink ldr and pull connected particles by using “center mass” to relocate “simple particles” to return the shape as it was before the rotation
        III) if arrived to goal stop else repeat 1
    1.b else -> wait and repeat 1 until arrived to “goal”

By moving forward when finishing the algorithm we remain with the same shape , reason being is we are shifting every particle in the same direction with the same size.(similar to shifting)
By rotating in theory we would finish as well with the same shape this is yet to be proven because it depends on the implementation of the center mass and the possibility to make bigger link and getting back to designated location in my algorithm(fish shape) it works because i'm not using a center mass but a specific algorithm similar to the general idea.



Reference:

Amoebot model:
https://cs.uni-paderborn.de/fileadmin/informatik/fg/ti/Publications/Publications2014/spaa0067-derakhshandeh_1_.pdf

Universal Shape Formation for Programmable Matter:
https://cs.uni-paderborn.de/fileadmin/informatik/fg/ti/Publications/Publikationen2016/SPAA2016-2.pdf

Undulatory Locomotion of Magnetic Multilink Nanoswimmers:

https://pubs.acs.org/doi/abs/10.1021/acs.nanolett.5b01981

Nanorobot Movement: Challenges and Biologically inspired solutions :
http://s2is.org/Issues/v1/n1/papers/paper6.pdf