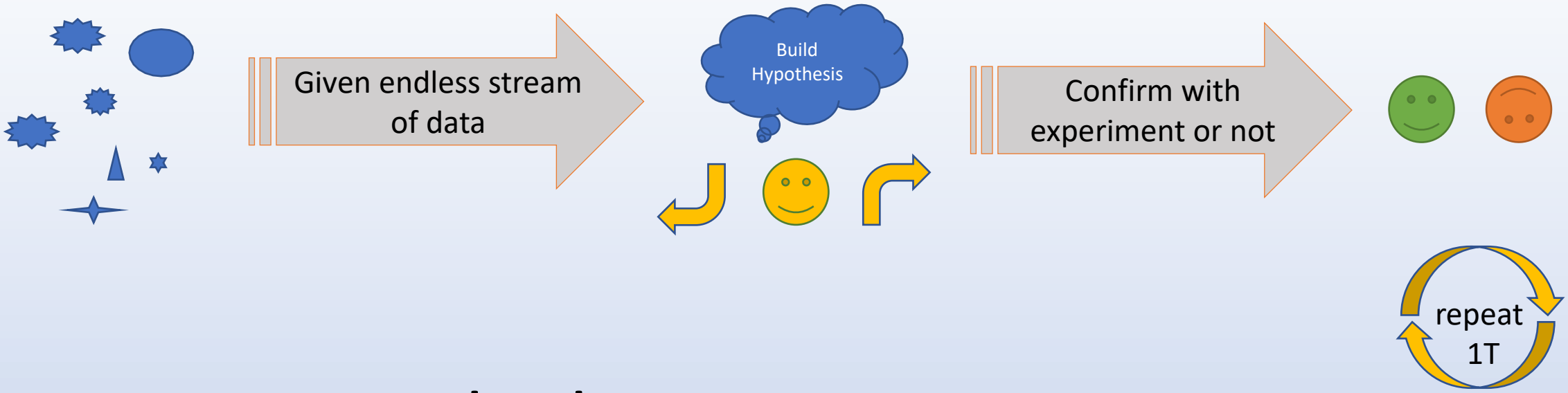
Back to basics. What's ~~AI~~, ML…?



Thinking = action

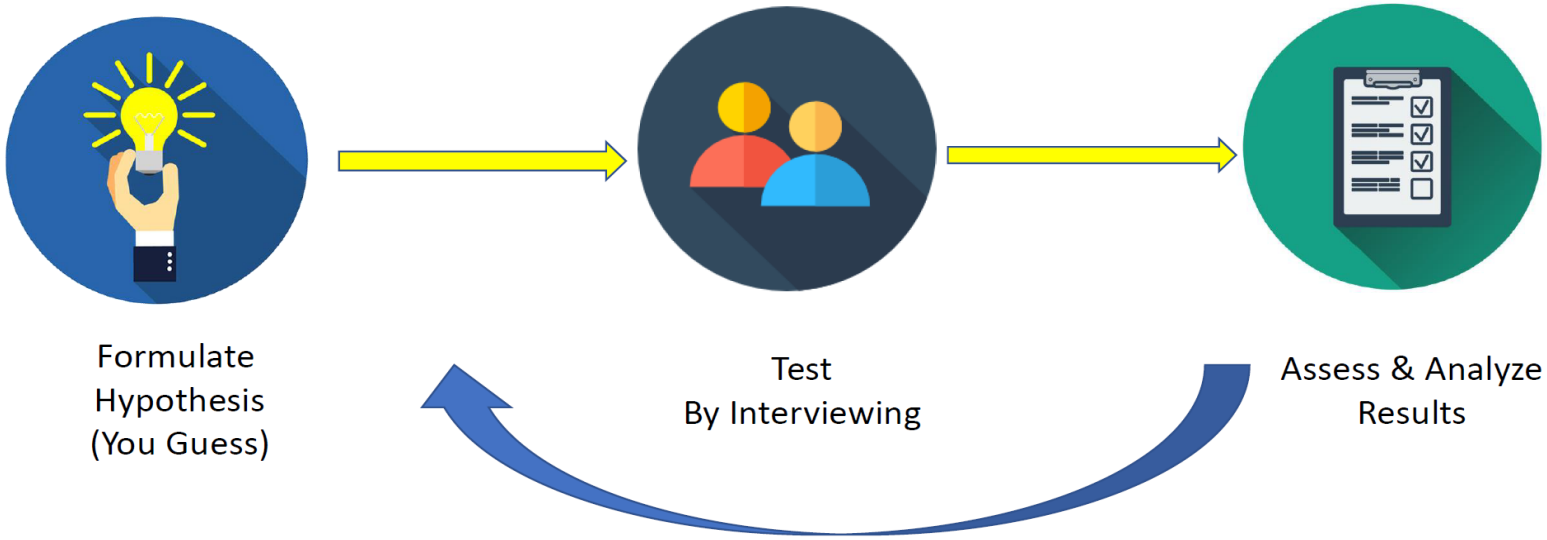
- a single act of choice (bit)
- based off a single piece of data (bit)
- with a single piece of affirmation (bit)

Most interesting question – how to build a good hypothesis?

What is AI? Business analogy

We Use The Scientific Method

Thank you, Francis Bacon, with help from Copernicus and Galileo



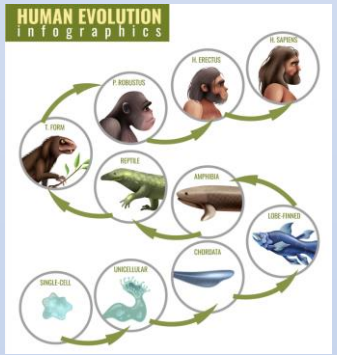
By Matt Lukens of the Icorps Workshop

Internet video innovator, visionary, mentor

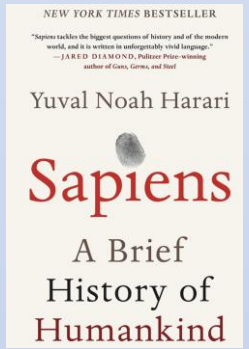
Back to basics. Physarum.

Physarum polycephalum can solve **TSP** (Travelling salesman problem, **NP-hard**) in polynomial time.

How?

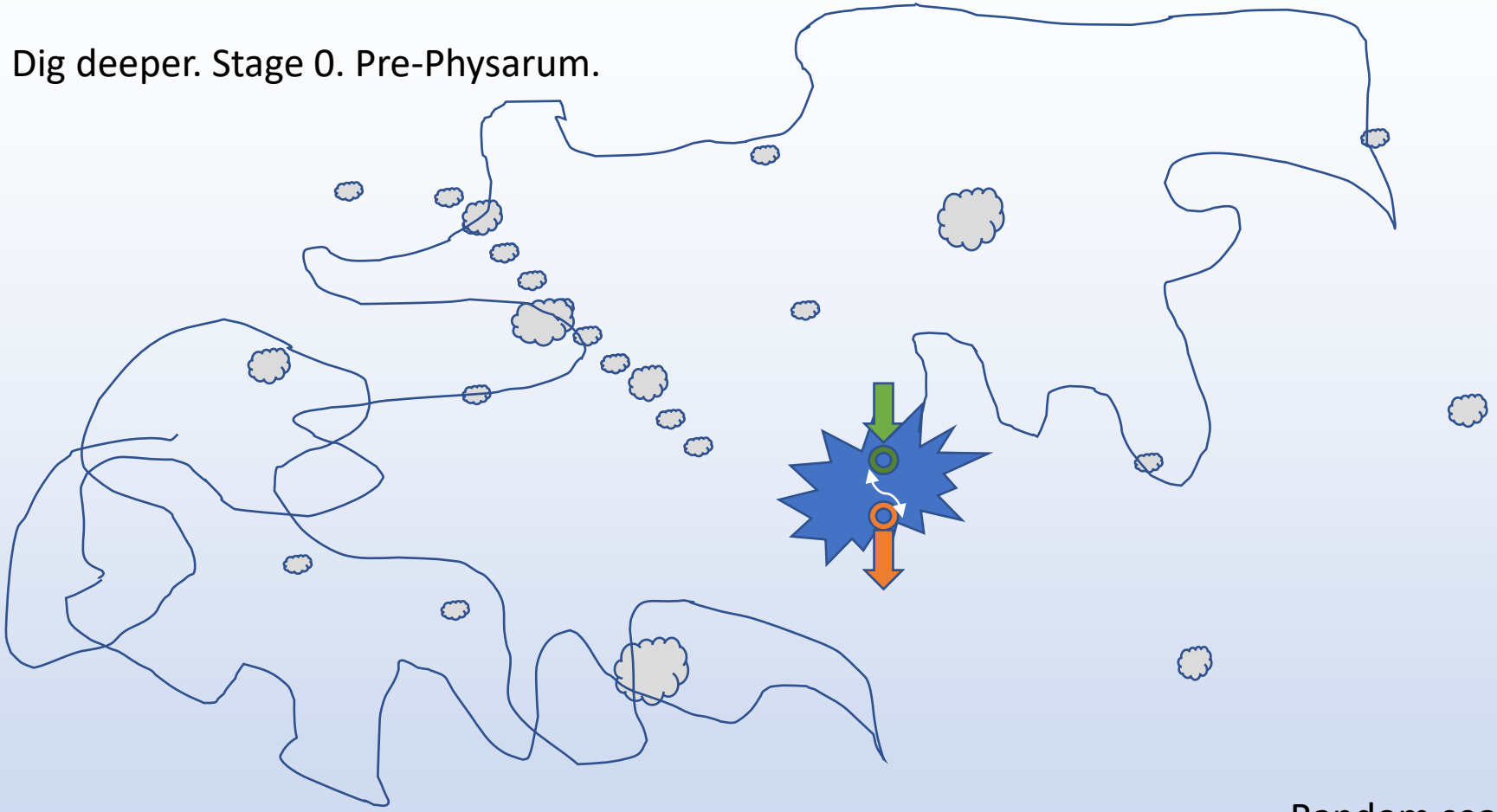


Min efforts -> Max result



Back to basics. Physarum. How?

Dig deeper. Stage 0. Pre-Physarum.

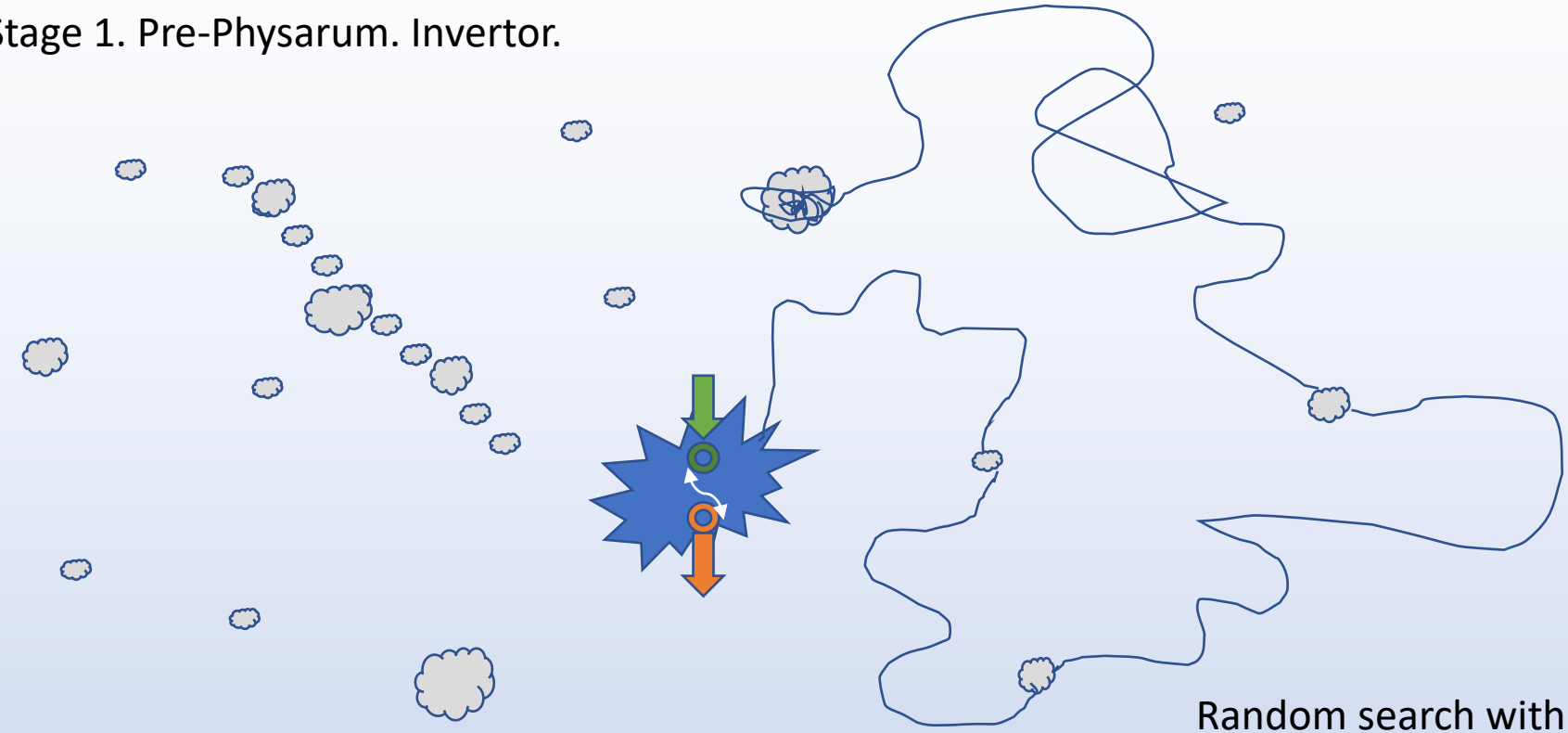


Random search. Ineffective but works, till now.
AI level 0

Min efforts -> Max result

Back to basics. Physarum. How?

Dig deeper. Stage 1. Pre-Physarum. Invertor.

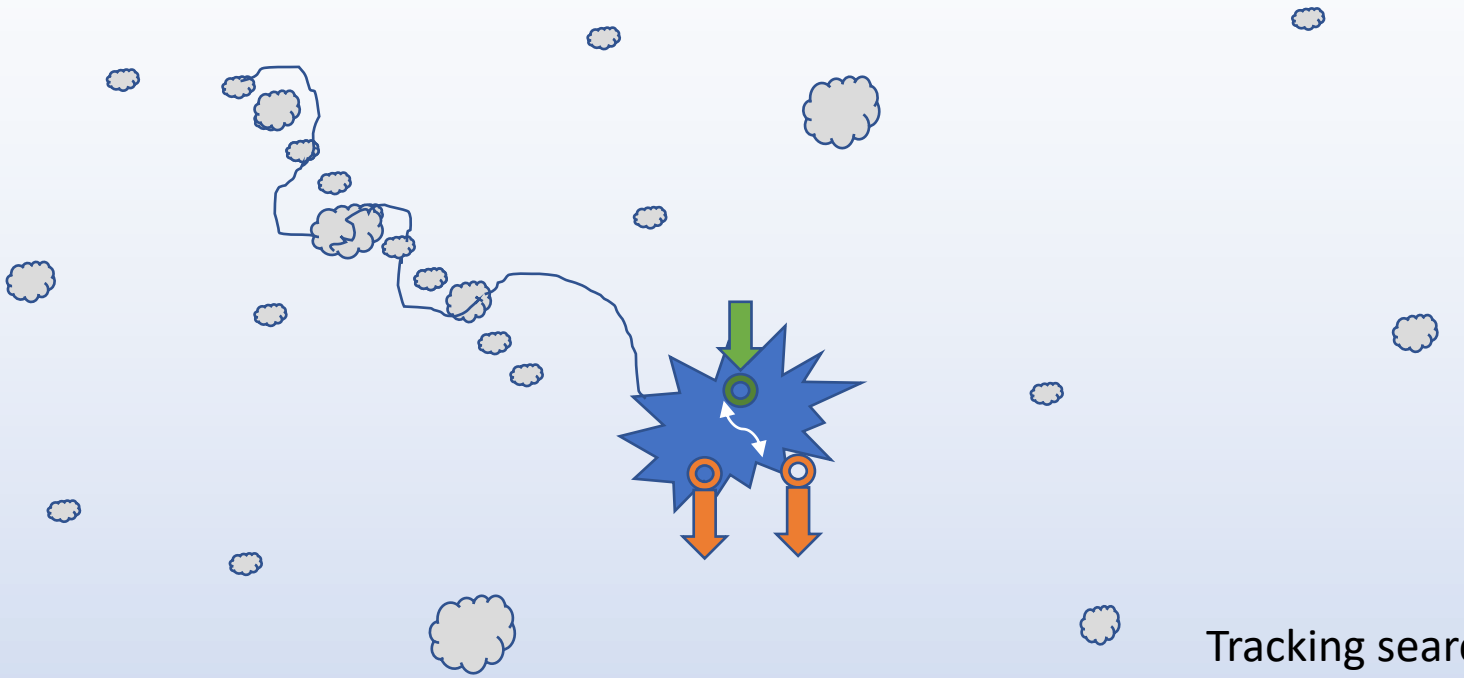


Random search with start/stop. More effective, works anywhere, evolutionary beneficial. **Out** with 2 speeds, inverted from **In**. AI level 1

Min efforts -> Max result

Back to basics. Physarum. How?

Dig deeper. Stage 2. Pre-Physarum. Tracking.

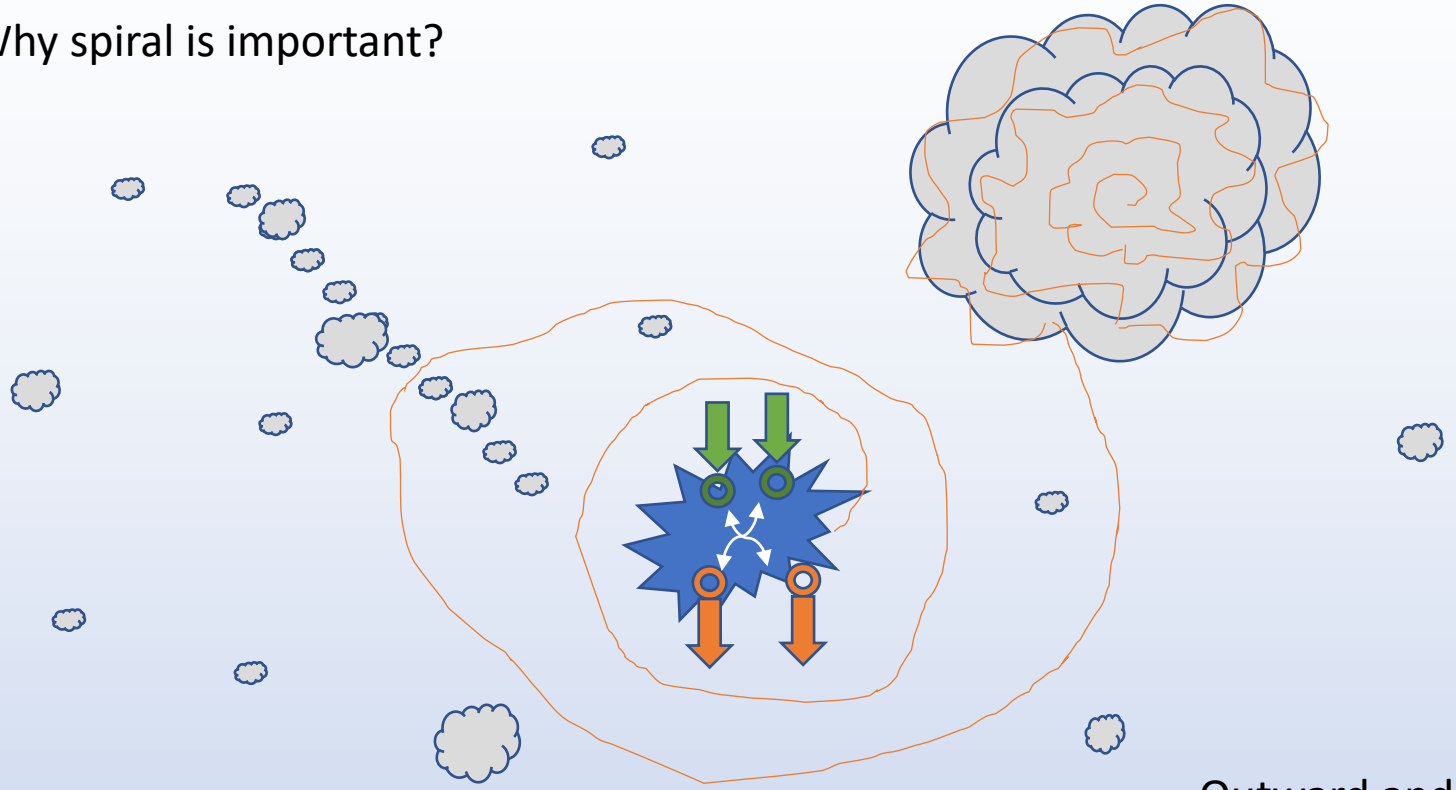


Tracking search with start/stop. Super effective, works anywhere. **2 Outs** with 2 speeds, inverted from **In**. Both Outs work at the same time – physics, otherwise – spinning.
AI level 1.5

Min efforts -> Max result

Back to basics. Physarum. How?

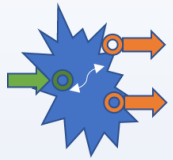
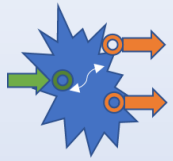
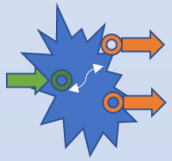
Dig deeper. Why spiral is important?



Outward and inward **tracking** spirals effectively cover surface, approximating food search with min time solution.
AI level - ∞

Min efforts -> Max result

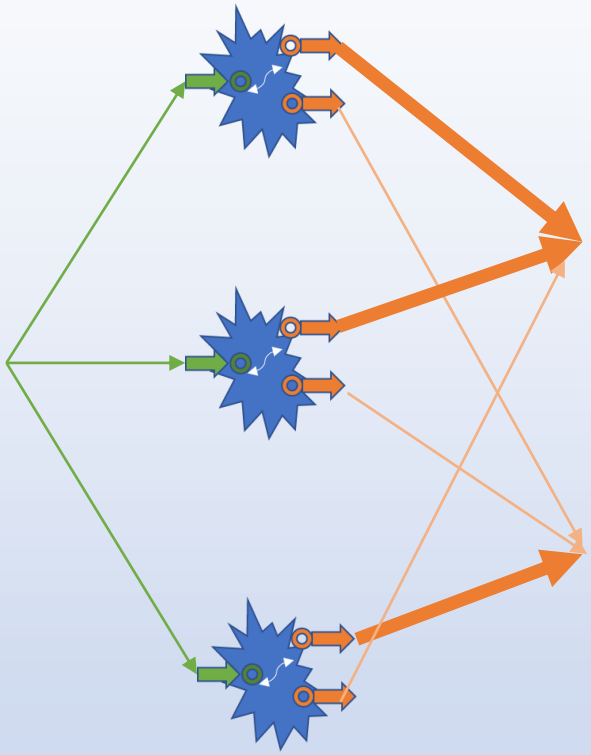
Implementation. Camertones. Important.

		Getting signals with lower frequency (output frequencies, left/right, INV)	Getting signals with higher frequency (output frequencies, left/right, GEN)
Expected frequency 1, 10 signal / min (example)		50 10	1 5
Expected frequency 2, 100 signals / min		400 100	20 50
Expected frequency 3, 1000 signals / min		1500 1000	200 300

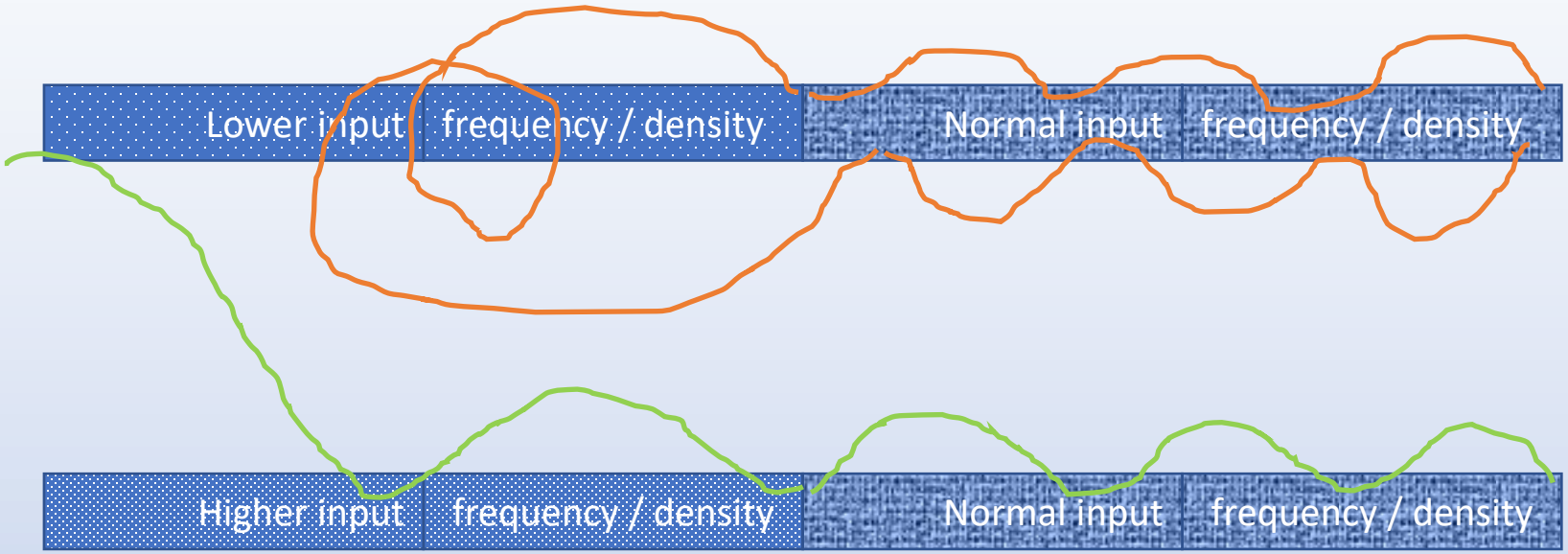
counterphase

Two variables – speed and differential (curvature), physics of turning

Implementation. Camertones. Important.



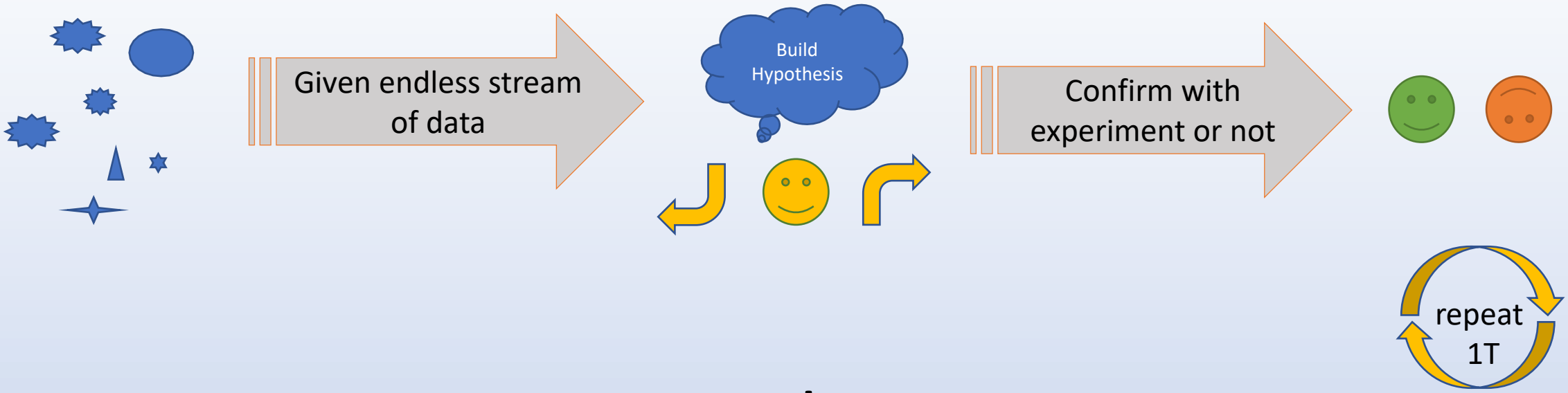
Lower density input => outward spiral



Higher density input => increased deviation

Model for optimal finding of the 1st degree symmetry (curved edge) and 2st degree symmetry (nearby edge)

What's AI, advanced ML…?

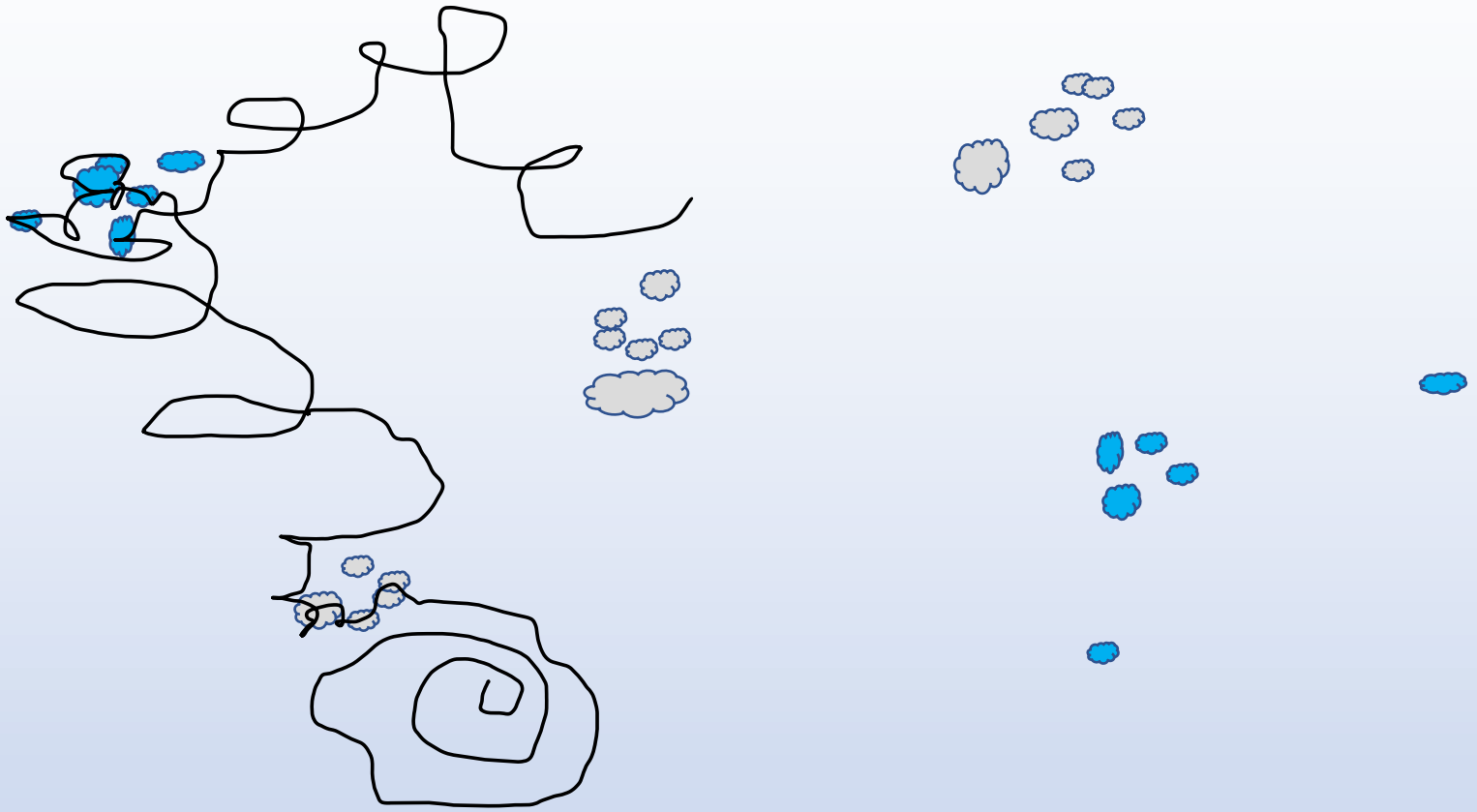


Iteration rules:

- Move, same direction (1D-hypothesis), in spiral movement
- If found something/edge, then stop/slow and pivot (reverse hypothesis)
- If lost again, continue, initial direction

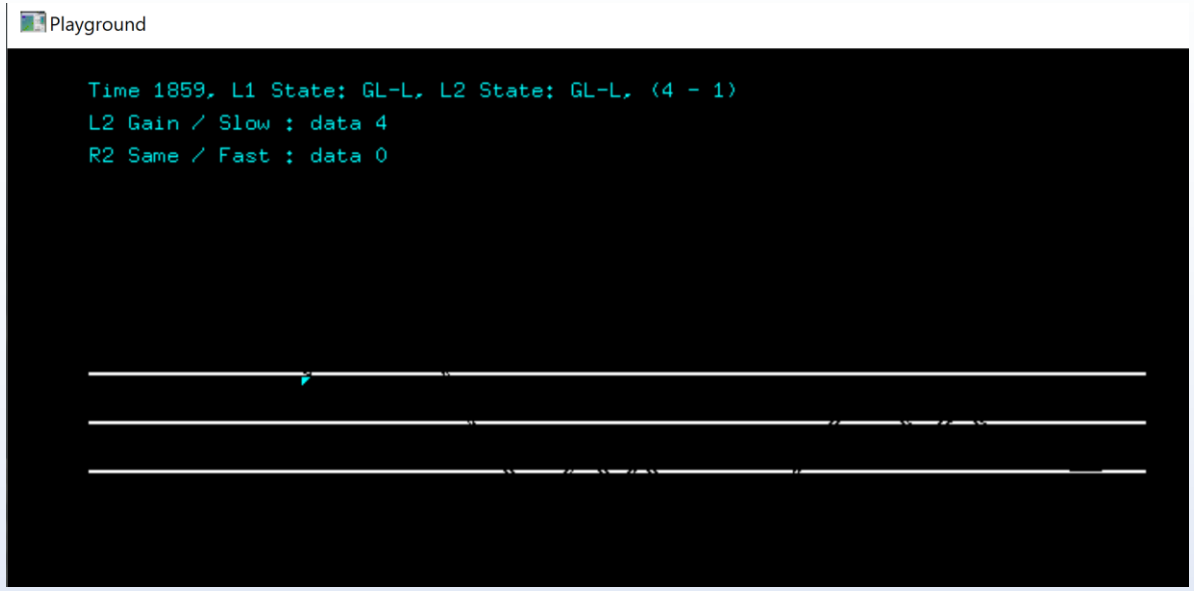
Question "how to build a good hypothesis" is answered

Real Complex Problems



- Two and more inputs (variables)
- Multi-factor optimization
- Same solution, combine, natural neural network

Implementation. Demo.



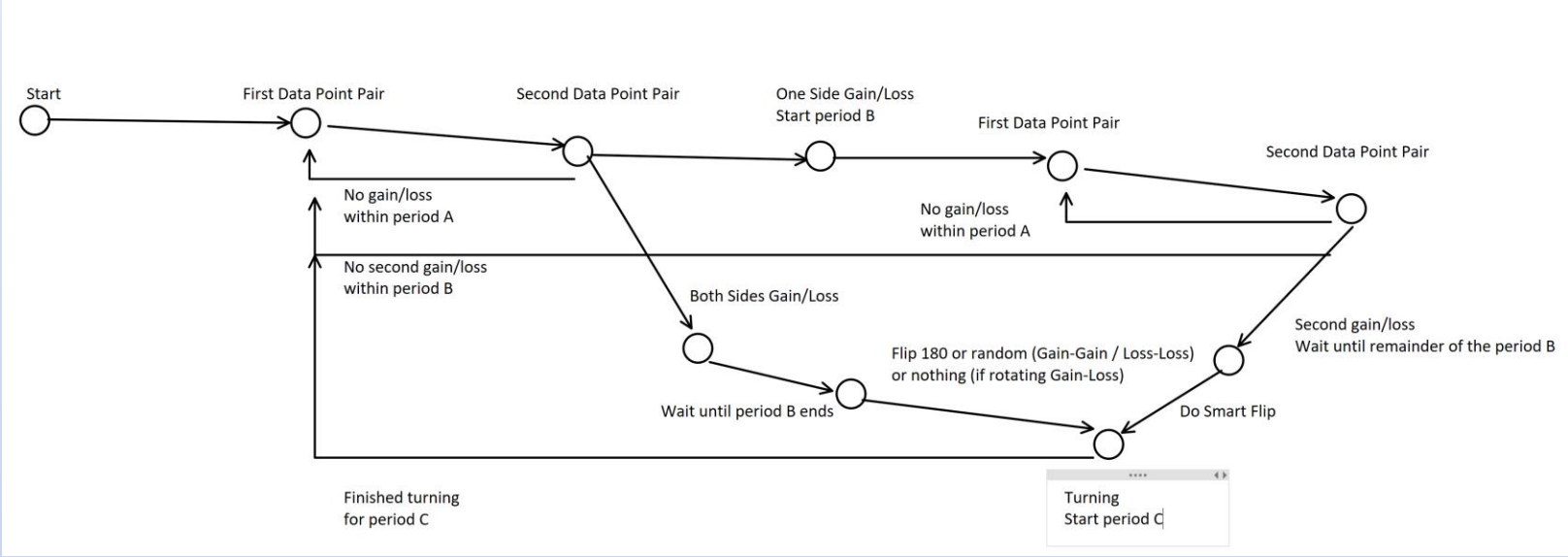
Tracking with 2nd level, “almost spiral”.
Implementation – C++/OpenGL, state machine on both levels, 1k LOC, end-to-end testing.

Hypothesis:

Multi-leveled tracking spiral is sufficient solution for the main task on all AI levels “Min efforts -> Max result”, which is NP-hard and beyond.

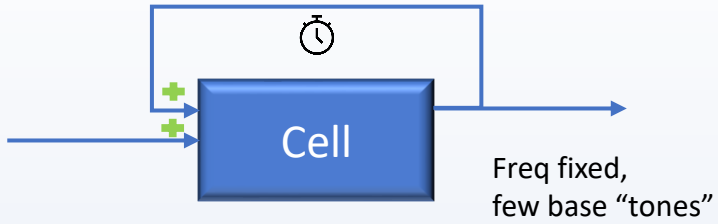
Question:

Is this elementary implementation? **No**.



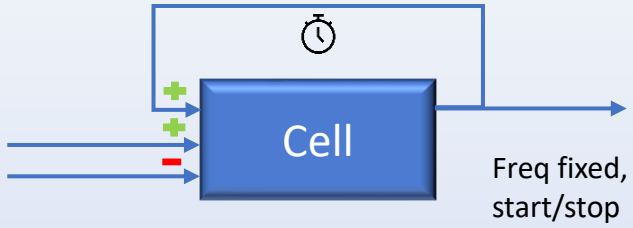
Implementation. Elementals.

(GEN)



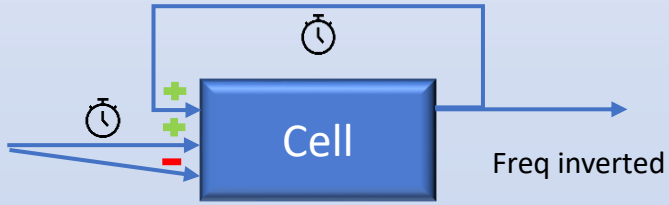
Random search, non-stop

(GEN start/stop)

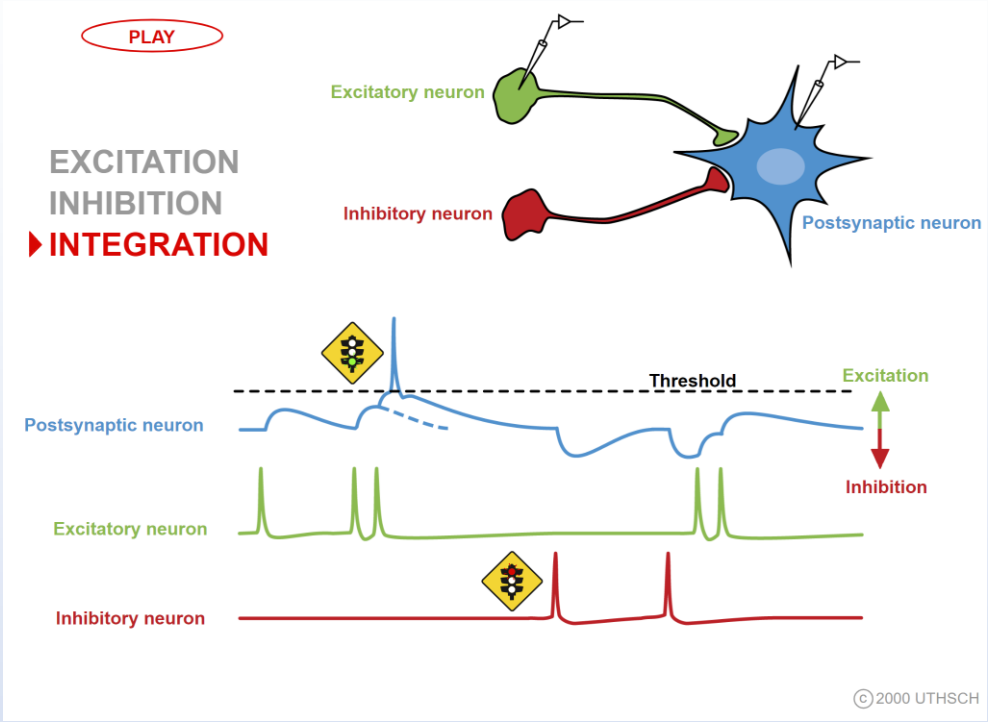


Call them GEN too as all generators should be stopped anyway

(INV)

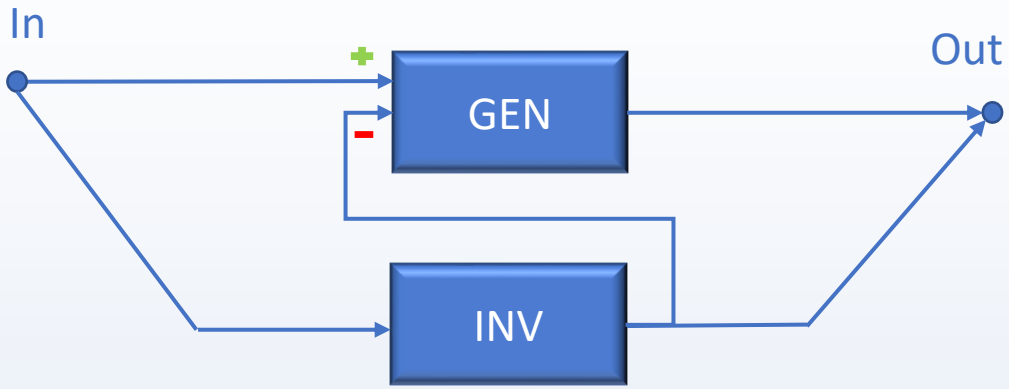


Smart search, start-stop



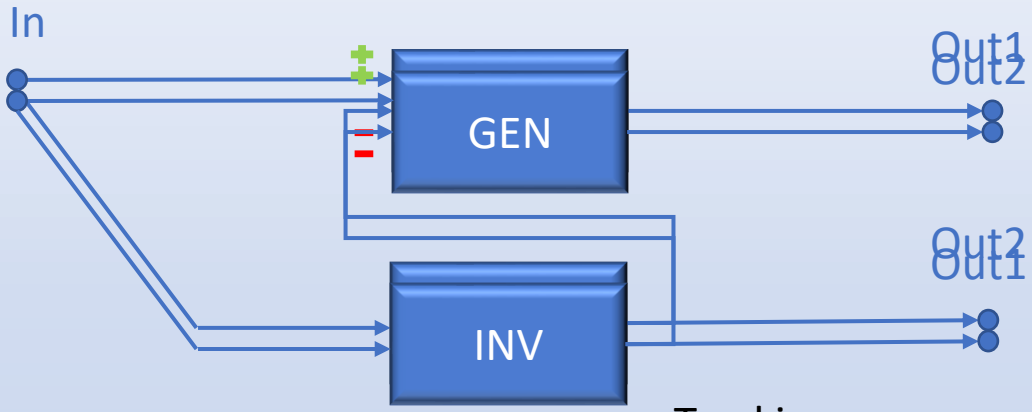
- There are a few precise models for neurons;
- Feedback delay may vary, affecting frequency but not too much, otherwise it can’t be stopped;
- Feedback delay may be != Input delay, thus generation frequency may differ from reaction frequency;

Implementation. Genesis.

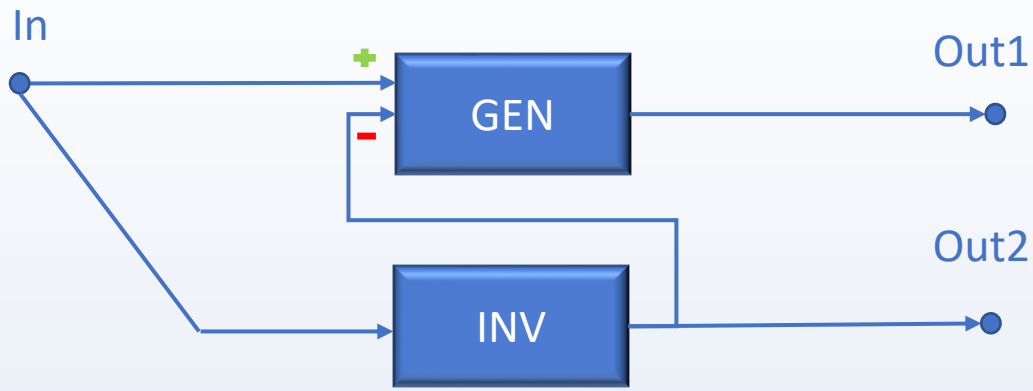


Better invertor implementation

(TRK)



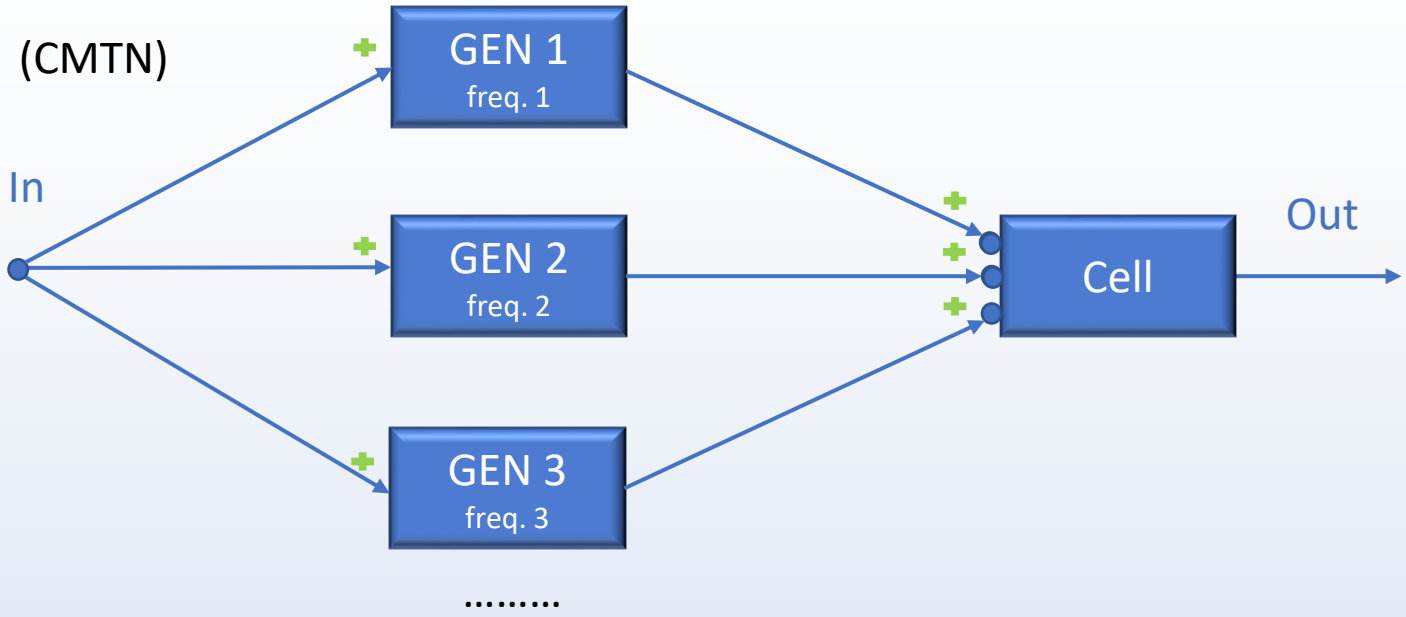
Tracking,
implementation by duplication,
phase reversed

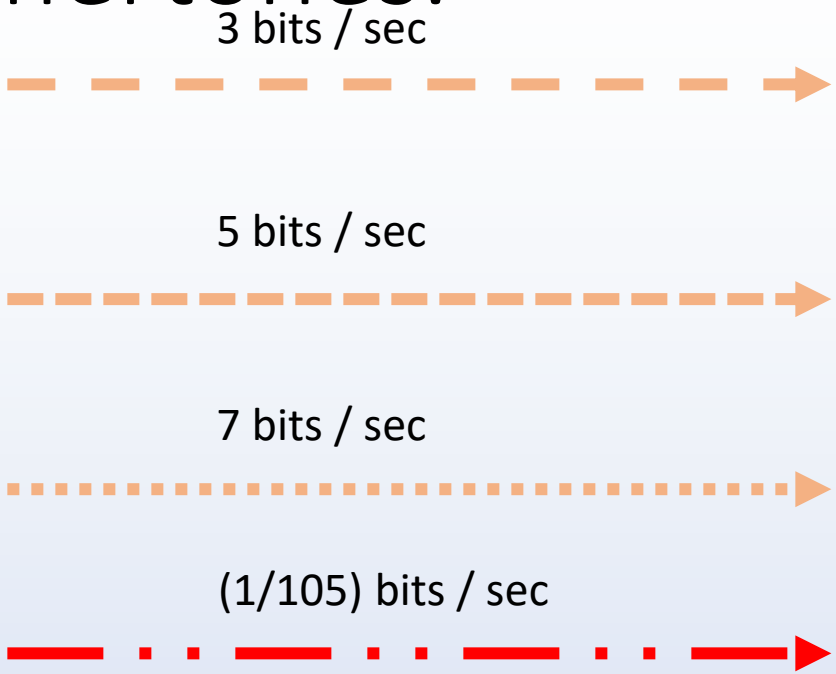


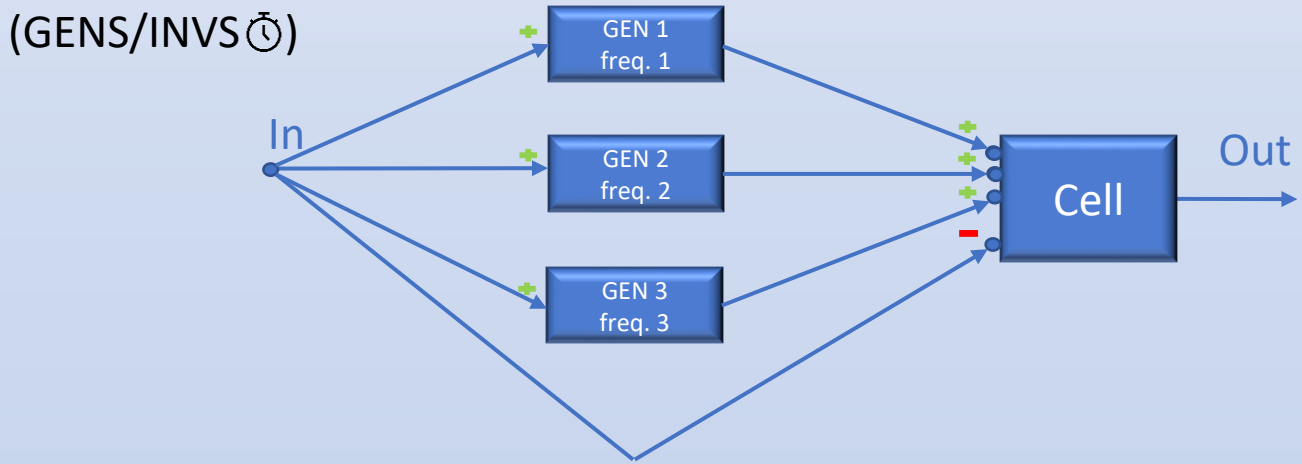
"Almost tracking"

- Few interim (evolutional) implementations omitted for simplicity;
- Timing is the key;
- Not reacting to signals, but to absence of signals within certain period;
- Reached Physarum AI level? Maybe!
- There's a catch – won't work for low frequency signals as "long-long-long delays" are way too costly and we couldn't stops even if we had it. How to solve?

Implementation. Camertones.



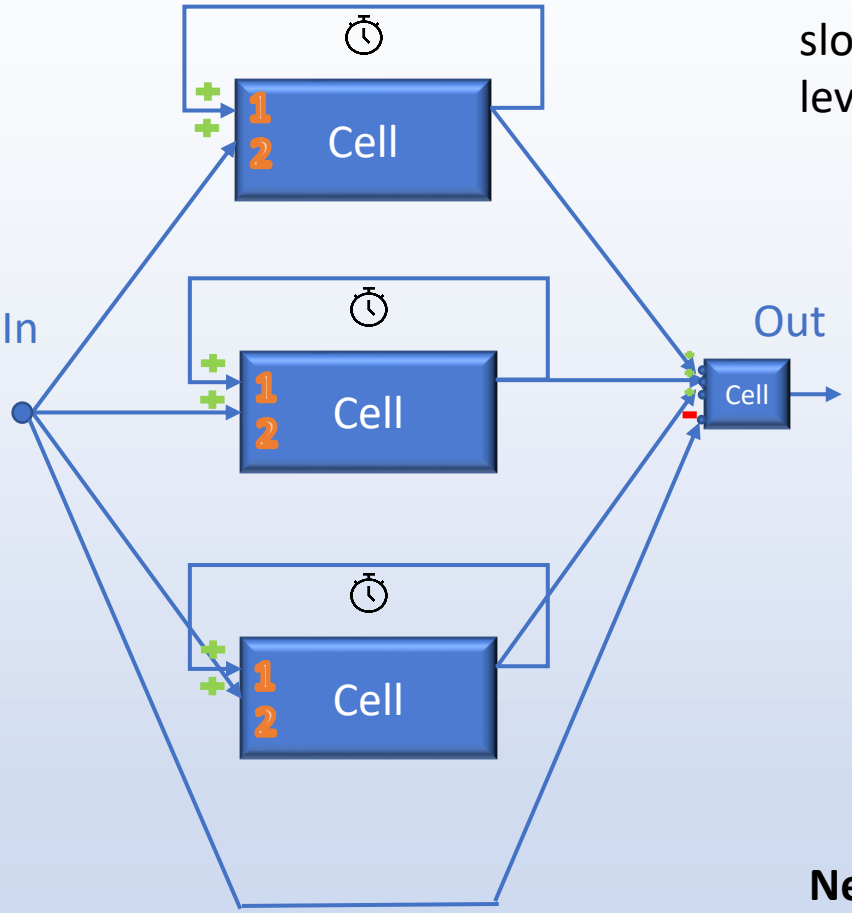




- See slide #15, GEN frequency can very, but not much;
- Timing is the key, obertones intersection;
- Frequencies should not necessarily be prime numbers, but better be pairwise prime;
- Think of CMTN as a slow GEN / long delay;
- But there's another use for it, see next slide;

Implementation. Magic.

(CMNT)



CMTN can either be slow GEN or slow INV depending on plasticity level:

1) GEN when delay ~= refractory period and #1 is prioritized per Hebbian synaptic learning rule (STDP), meaning as we start generation we ignore #2 for the delay period. Note that GEN must be stopped at some point ('-' not shown for simplicity);

2) INV when delay >> refractory period and #2 is prioritized per Hebbian synaptic learning rule (STDP), meaning we're restarting generation again and again, ignoring #1 for the delay period;

Magic? Music!

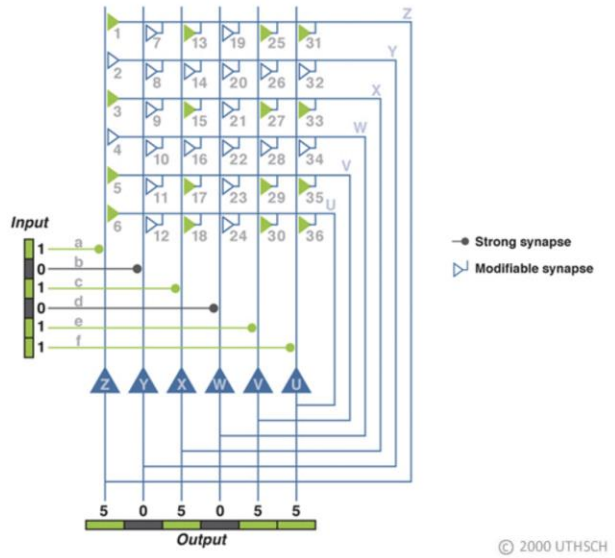
Again, timing is the key;

Neuroplasticity (LTP/LTD/STDP), sAHP, "spike frequency adaptation"
Delay & refractory period, time before can be triggered again.

ttps://nba.uth.tmc.edu/neuroscience/m/s1/introduction.html

Feedback/recurrent excitation

Recurrent excitation in nanocircuits and microcircuits appears to be critical for learning and memory proc changes in the biophysical properties of neurons and changes in synaptic strength. Accumulating eviden feedback within biochemical cascades and gene networks is an important component for the induction an changes. Moreover, recurrent excitation is found in at least some microcircuits involved in memory proce found within the CA3 region of the hippocampus.



© 2000 UTHSCH

Figure 21. Modified from Byrne and Roberts, 2009.

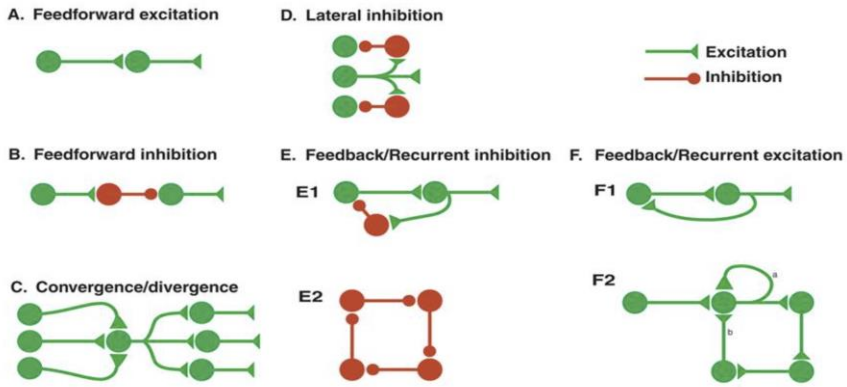
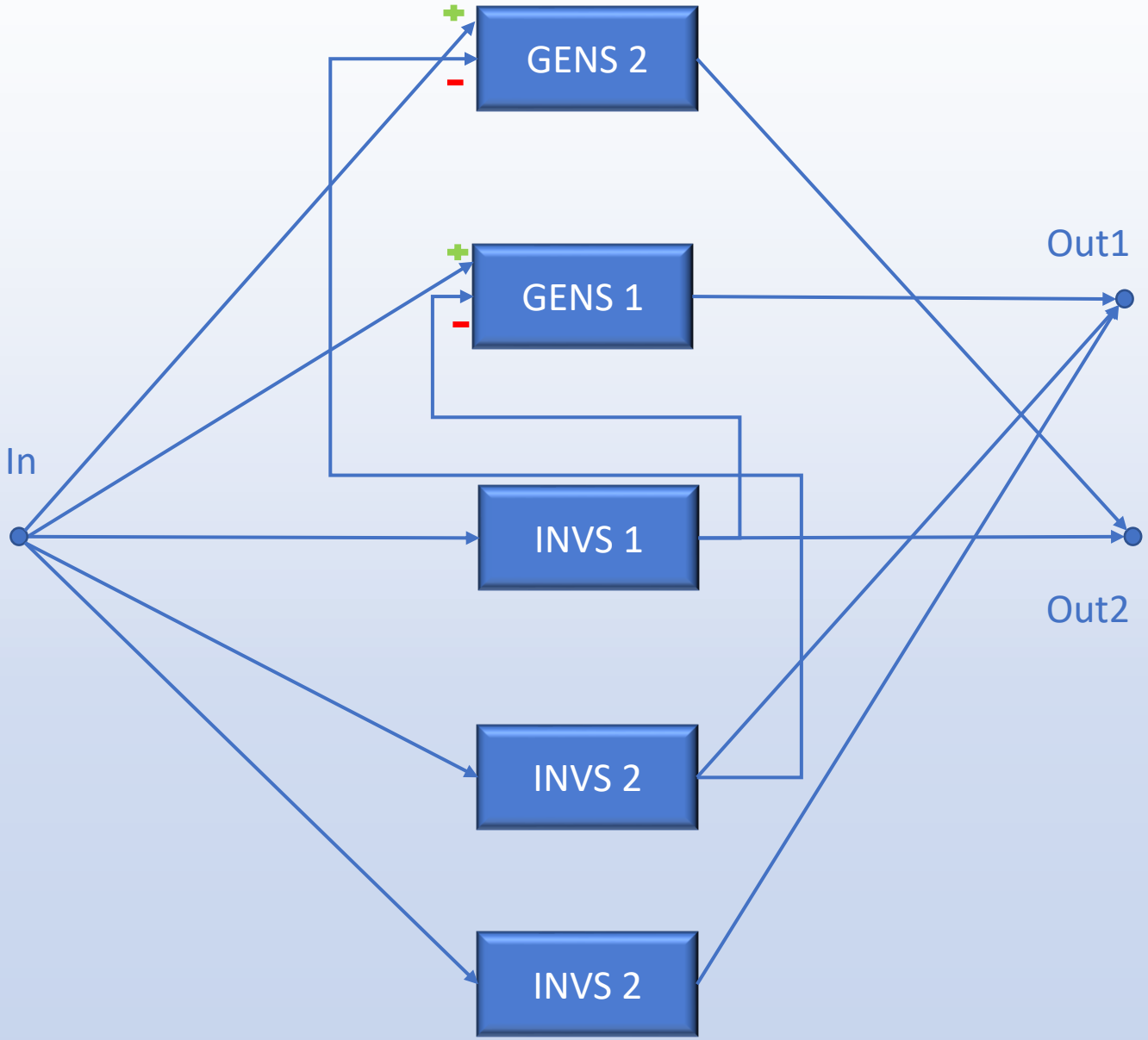


Figure 6

Implementation. Final.

(TRKS)

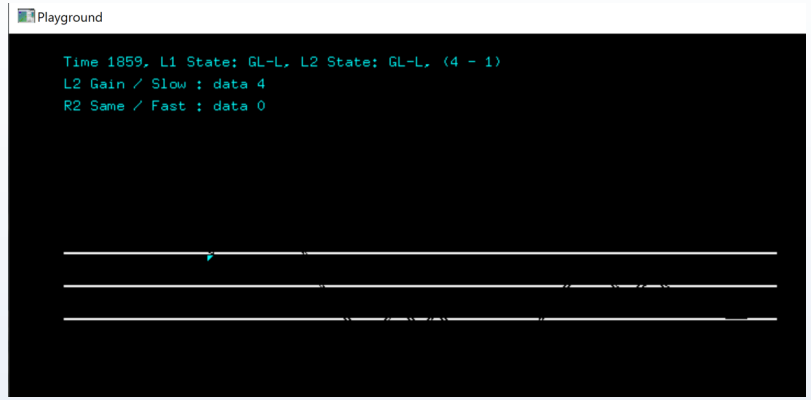


- **Tracker**, see slides #13 & 16 works the same way, but **speed** as slow as required
- **Phase** can be balanced as required
- Each dual-part of the **tracker** is tuned to its own incoming frequency/tune and generating two pairs of stable output frequencies, one for positive income and another for negative, absence of income.

But how do we know what's required? Get back to slides #12-14 and rethink it.

Implementation. Requirements.

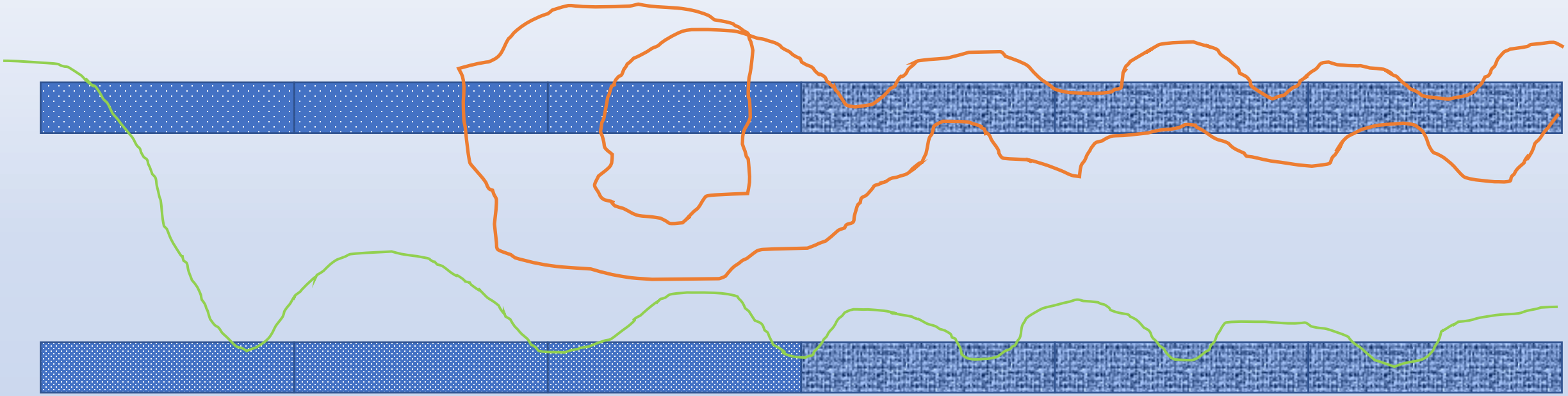
But how do we know what's required? Get back to slides #14-15 and rethink it.



We need to find patterns and with this multi-level spiral tracking we can find patterns of 1st symmetry level, 2nd level, etc! And why do we need patterns?

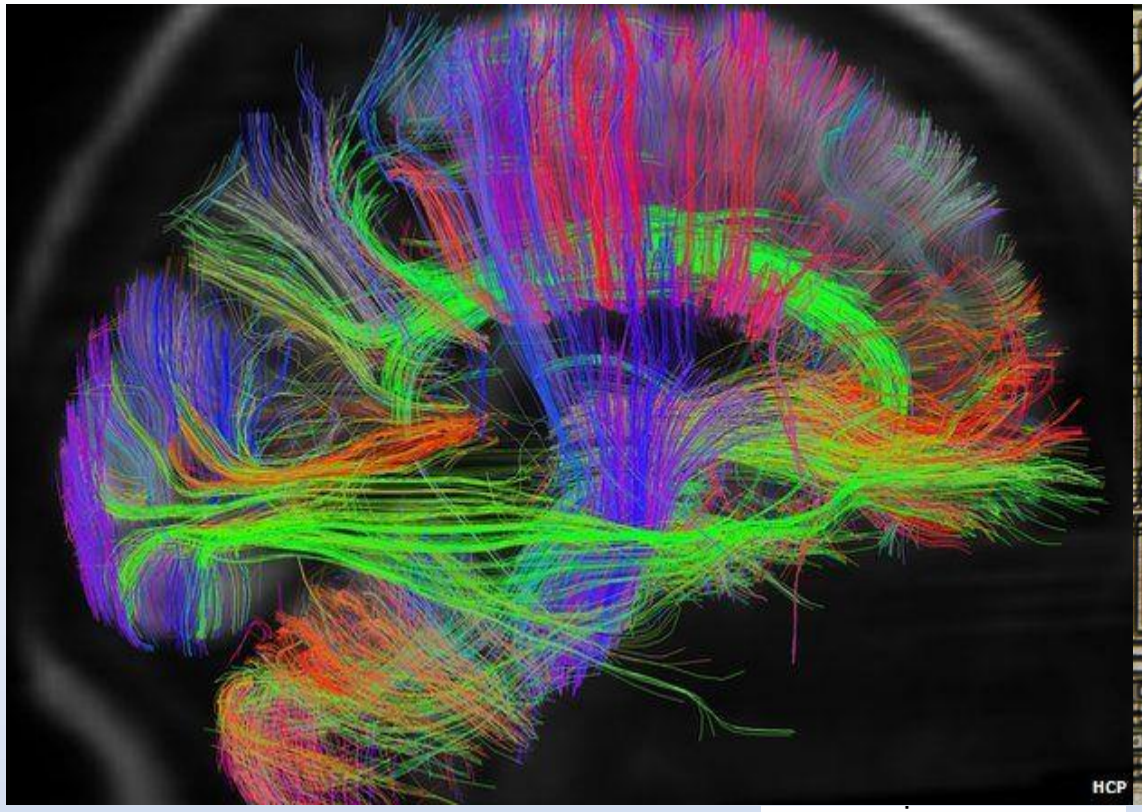
Min efforts -> Max result

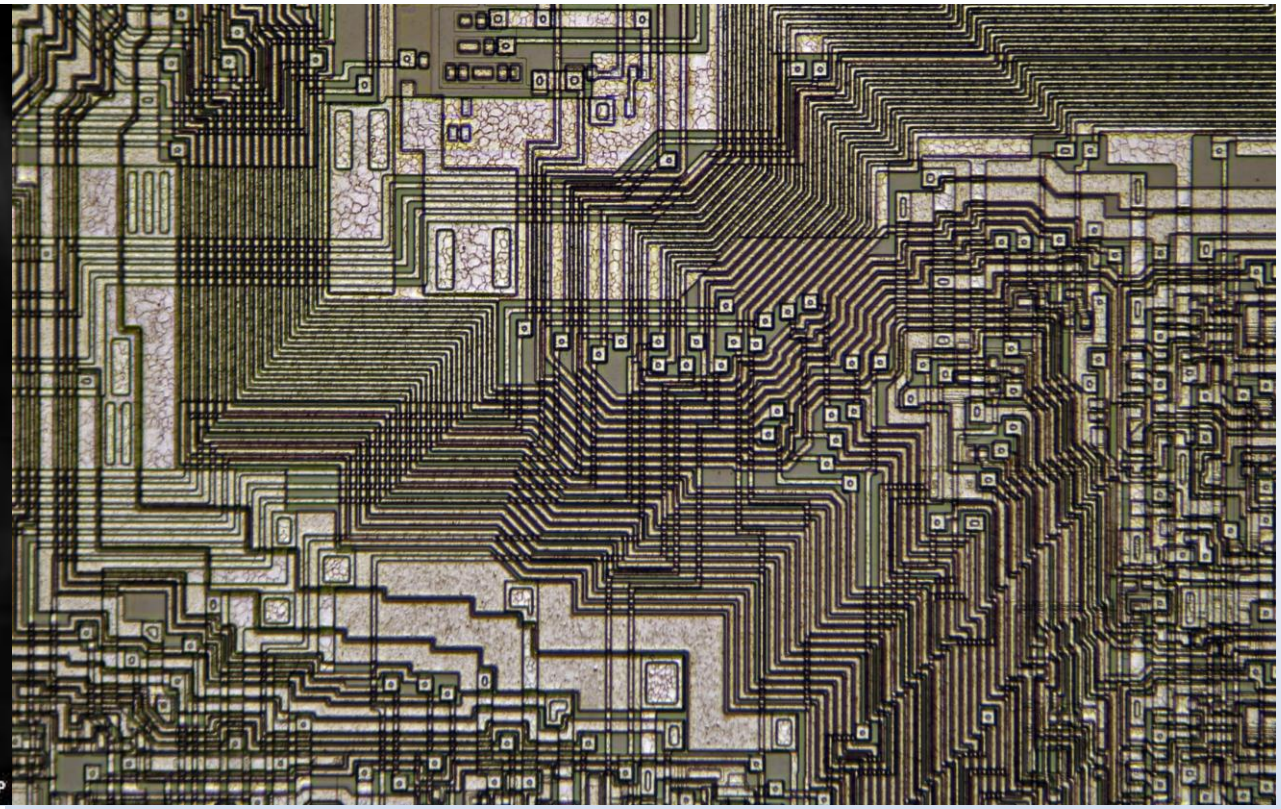
scalable



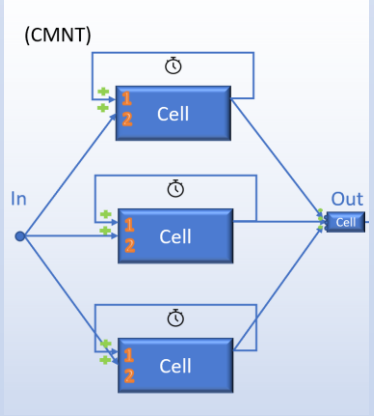
We encode/model/**change** our surroundings/environment by **changing** our topology, adding/removing INVS/GENS based on result of every min experiment, see slide #6

Why so simple?

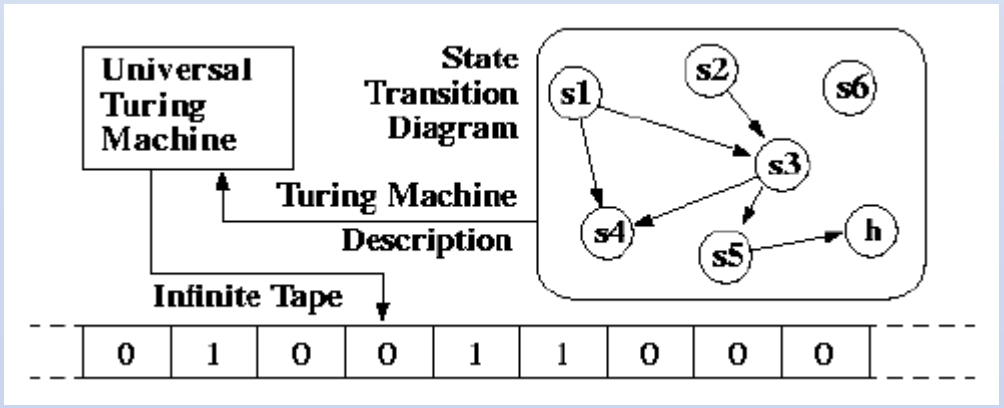




[Connectome]



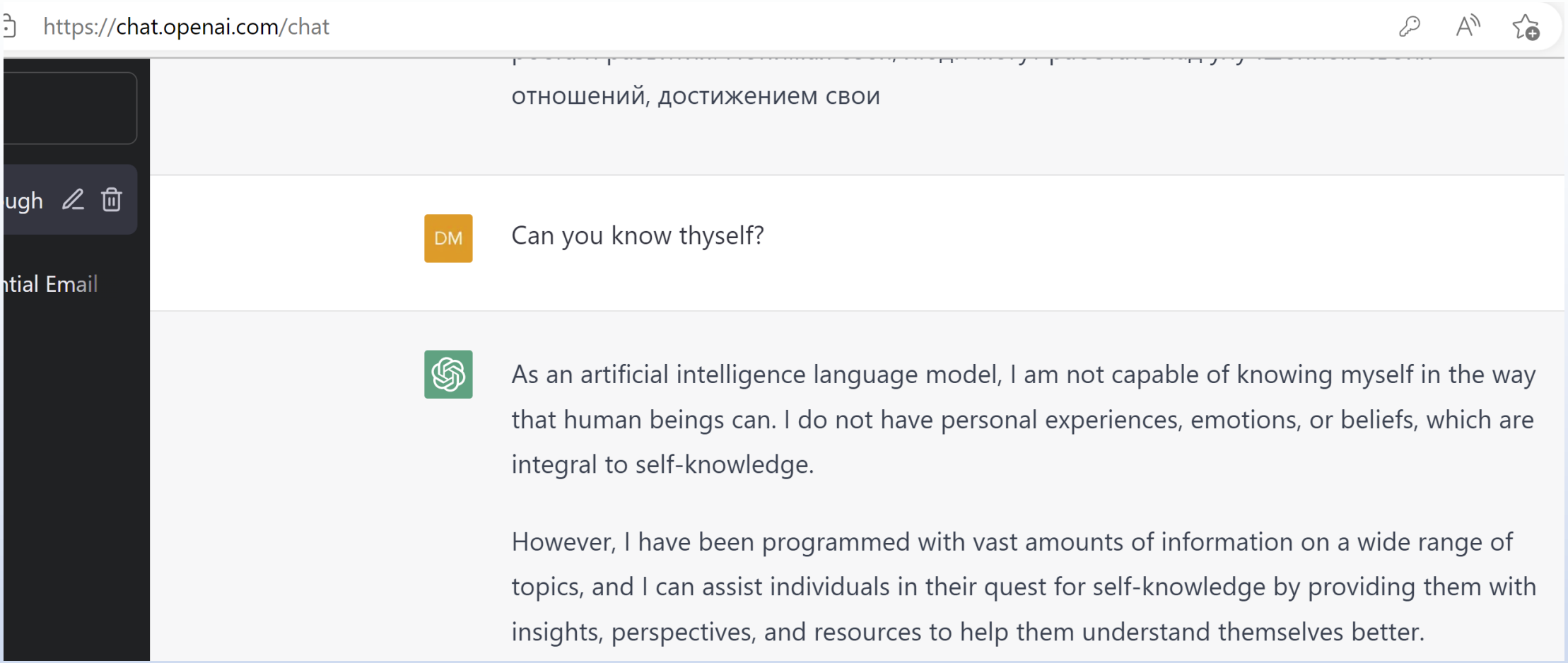
Camertones



CPU

[Turing Machine]

Back to the future



Valid answer. But can we do better than LLM/parrot-fashion models? ;-) Yes, as decomposition complete we can start building from bottom up complex math models on solid foundation, Natural Neural Network, NNN.

Be honest. Be humble. Be bold.