

# **BIOSIM4 PORT FROM C++ TO FREE PASCAL**

---

**Is it worth the effort?**

Uwe Schächterle  
3.7 - 7.7.2023

## About the speaker

**STIHL**



Master degree in Computer Science at the university of Stuttgart

Master Thesis: Modellierung und Simulation bildgebender Systeme für partikelbeladene Strömungen

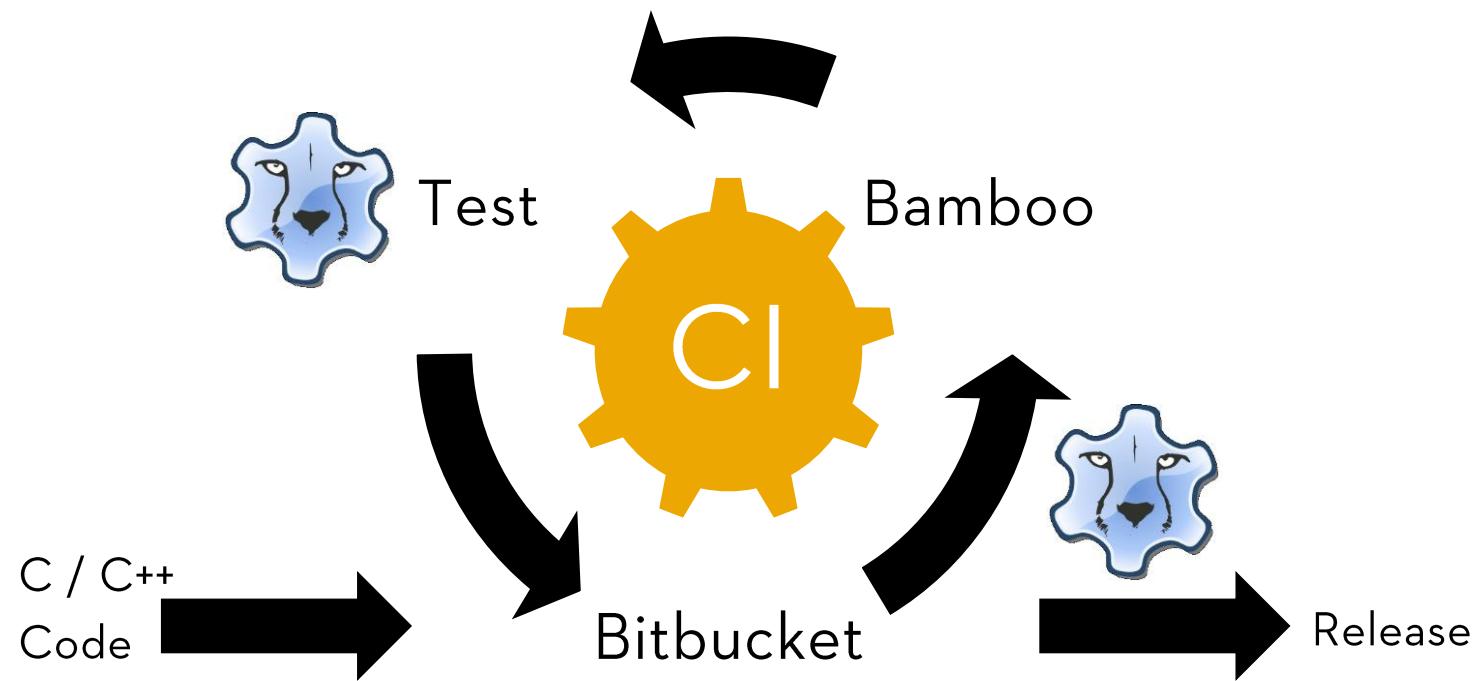
Embedded software engineer since 9 years (C and Free Pascal)

STIHL employee since 2018

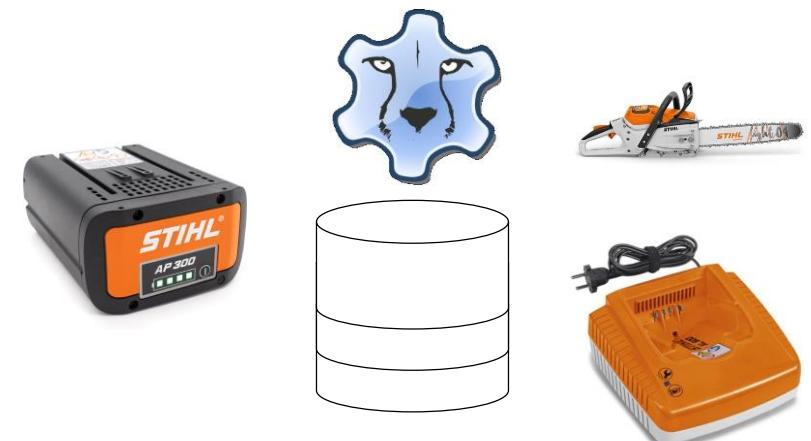
Responsible

- passive battery systems
- communication protocols between battery, charger and machines
- approval support for charger, active battery systems and as-system





Communication database



# This is a Manner talk

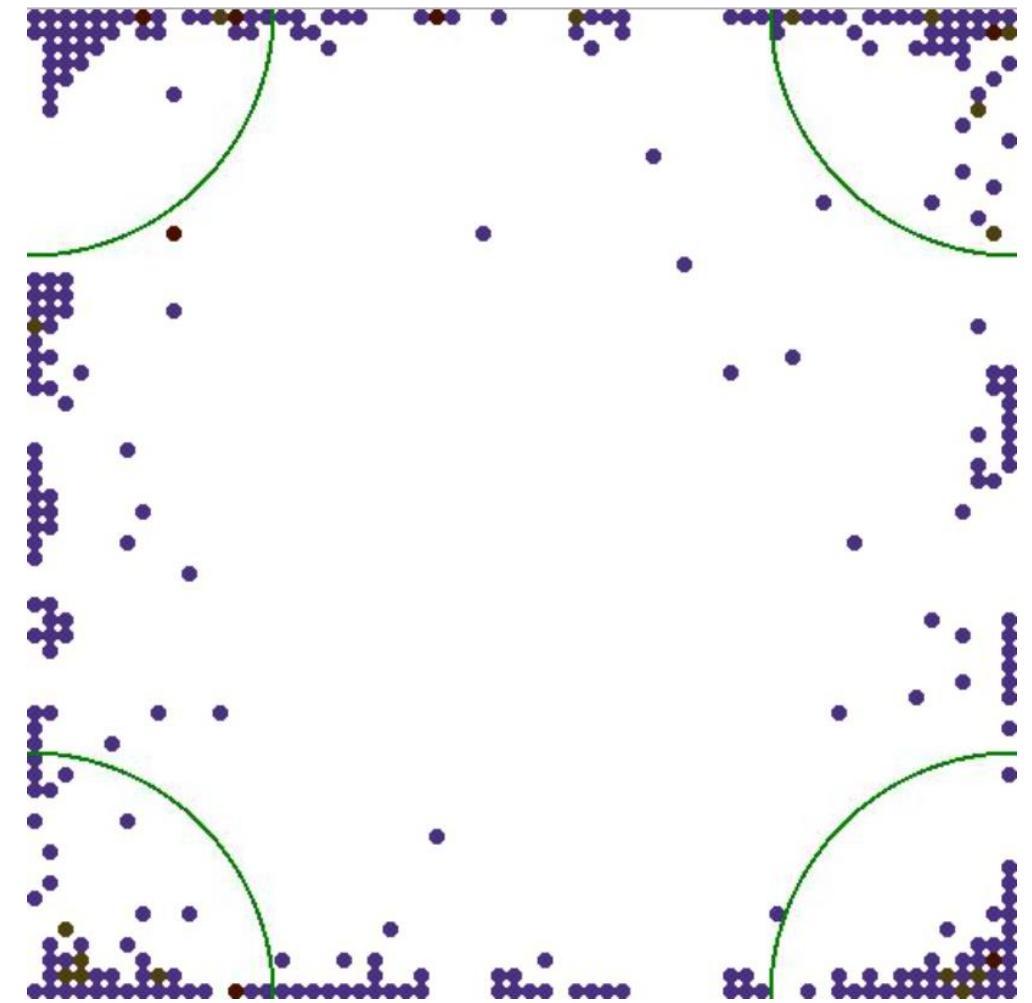
**STIHL**

When contributing a usefull content by asking or answering a question you get a Manner



Only one manner per person, need to be picked up at the speakers place after the talk

- What is Biosim4 ?
- Starting point
- C++ Libraries
- How the code was validated
- Comparison
- Improvements compared to the original
- Conclusion

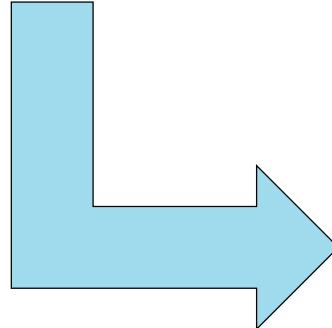
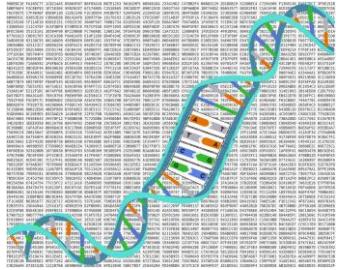


# What is Biosim4 ?

**STIHL**

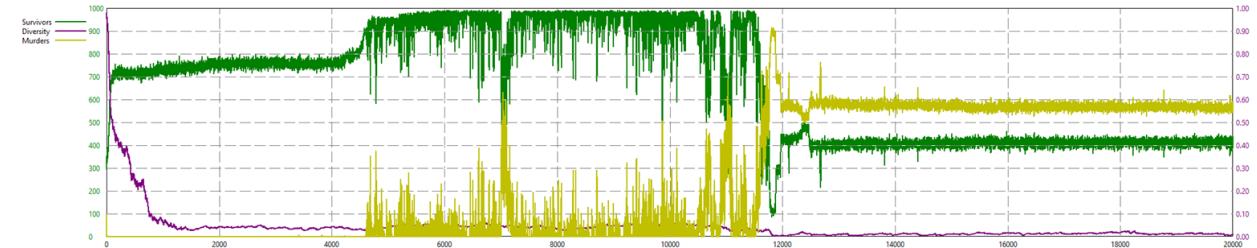
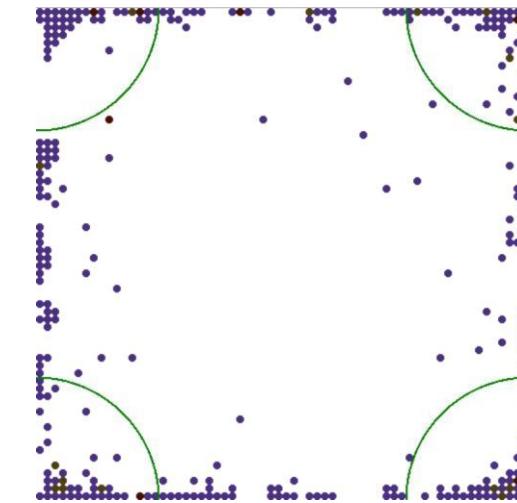
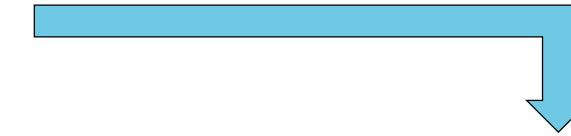
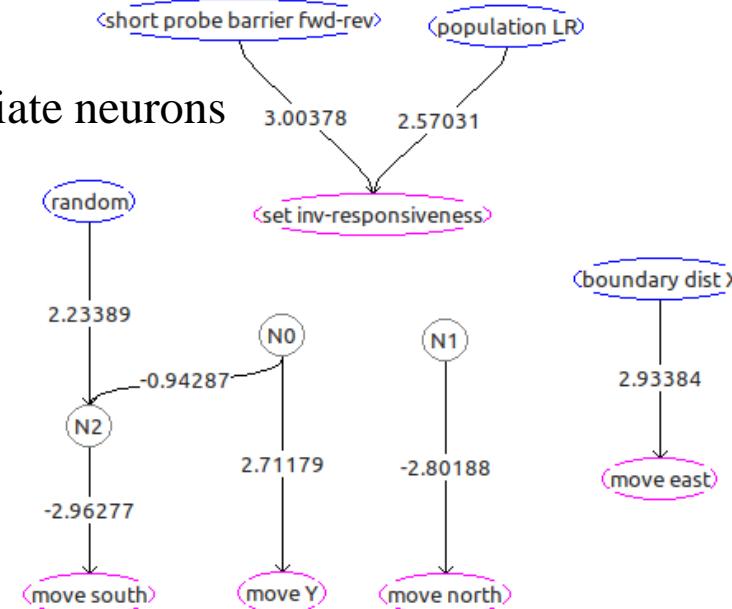
Based on the Book „The selfish gene“ by Richard Dawkins, David R. Miller programmed „Biosim4“

<https://github.com/davidrmiller/biosim4>



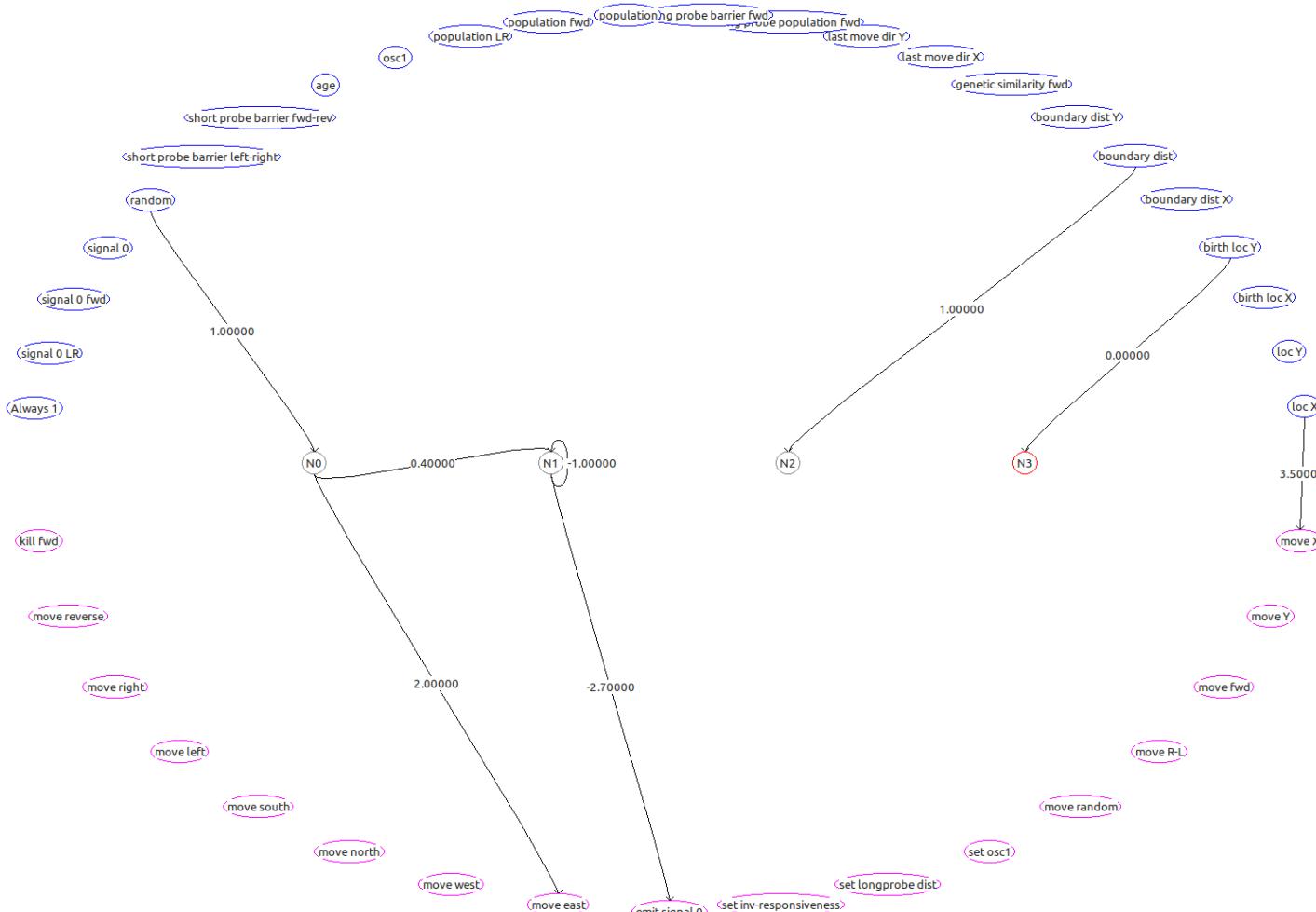
The individuals consists of

- actors
- sensors
- intermediate neurons



# DNA to Individuum conversion

**STIHL**



- 24 Sensors
- N-inner neurons
- 17 Actors

Each gene (32-Bit integer)  
forms one connection in the individuum

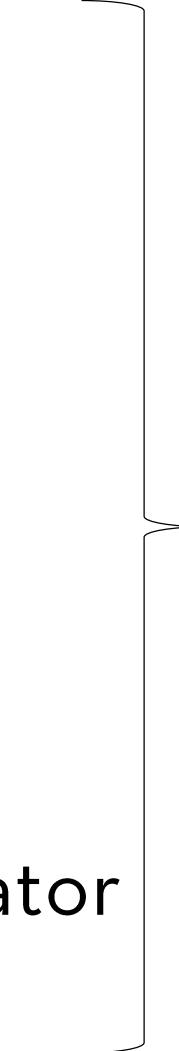
weight	sink	source
20000040B		

## Starting point

.35 C++ Files

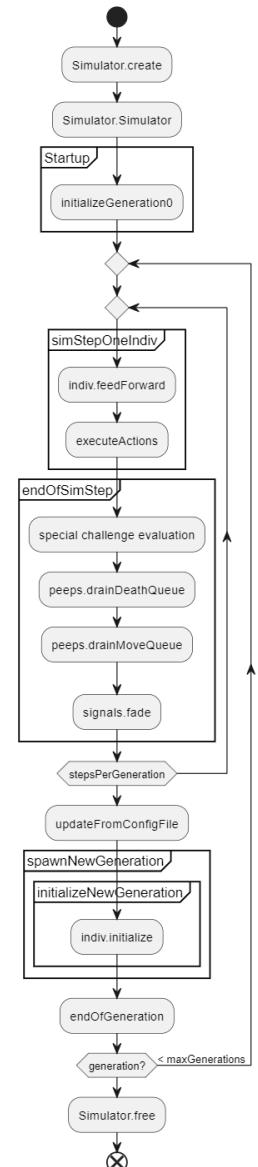
.3277 Lines of Code

- Anonymos functions
- Bitwise unions
- Multidimensional arrays
- Structs with methods
- Integer Enums
- Custom random number generator
- Pointer over pointers



Porting took ~6 weeks  
Debugging/validating  
another ~6 weeks

**STIHL**



- std::list
  - > Array of (integral part of FreePascal language)
- std::map
  - > TFreePascalGeneralMap (part of the <FPC-Source>/RTL/fgl.pp)
- OpenMP (parallel programming)
  - > MultiThreadProcs laz (part of the „Lazarus Component Library“ LCL)
- OpenCV indirectly used by Cimg.h (Video rendering Image creation)
  - > ugwave (ported from Michael Kohn , Robin Hahling from github)  
<https://github.com/rolinh/libugwave>
- gnuplot (command line plotting tool)
  - > usimplechart (own implementation)
- igraph (collection of network analysis tools)
  - > ugraph (own implementation)



1. FPC-Compiler / Heaptrace found lots of div by zero and memory leaks
2. Unitests (part of the C++ version) ✓

3. Running the simulation and reproducing the results of David R. Millers youtube video

<https://www.youtube.com/watch?v=N3tRFayqVtk>

- > Challenge 1 (move to the right) ✓
- > Challenge 13 (move left and right eights) ✓
- > Challenge 6 (weighted corners) ✗

4. Including a step by step tracer in both Free Pascal and C++ Code

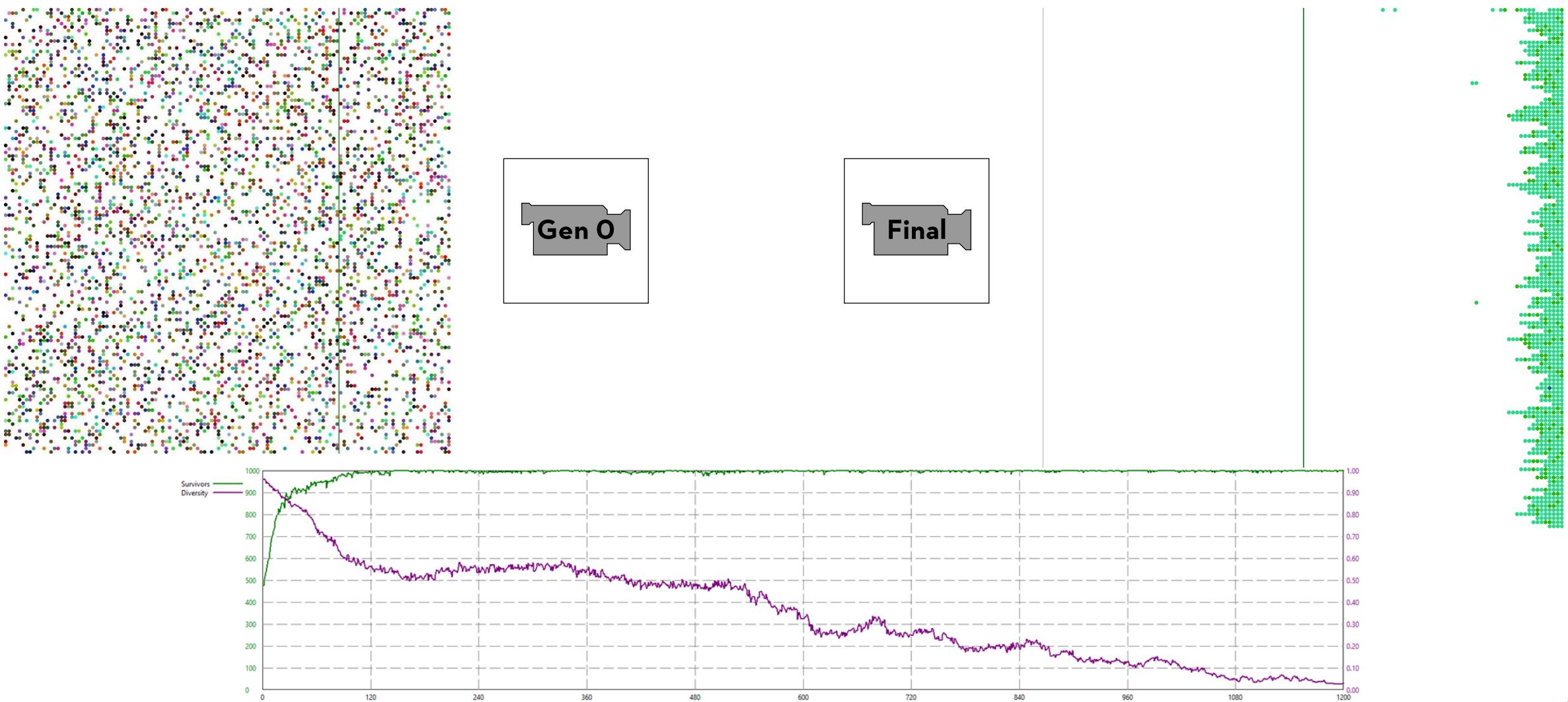
- > 200k tracepoints taken and compared during the first generation run of Challenge 6 ✓

<https://github.com/davidrmiller/biosim4/issues/86>

=> **Survival of the unfittest works, if the problem is easy**

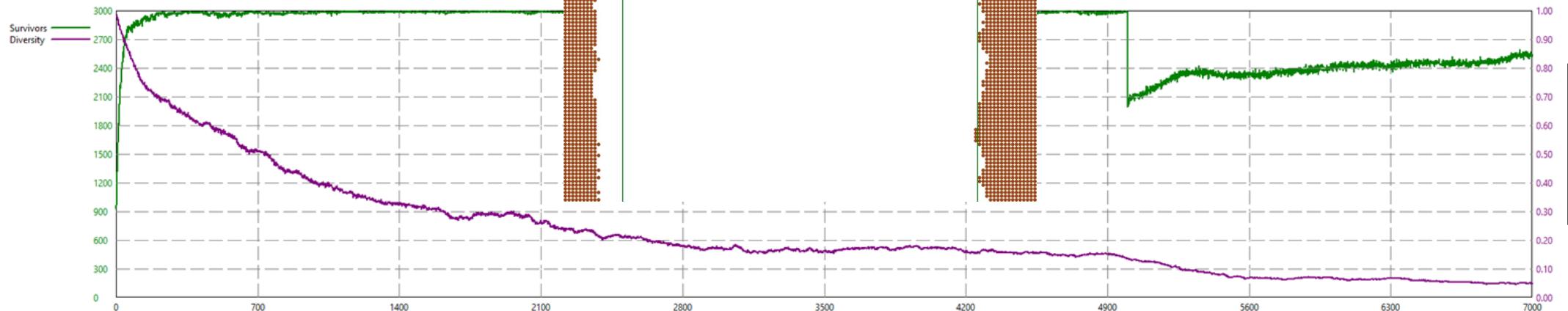
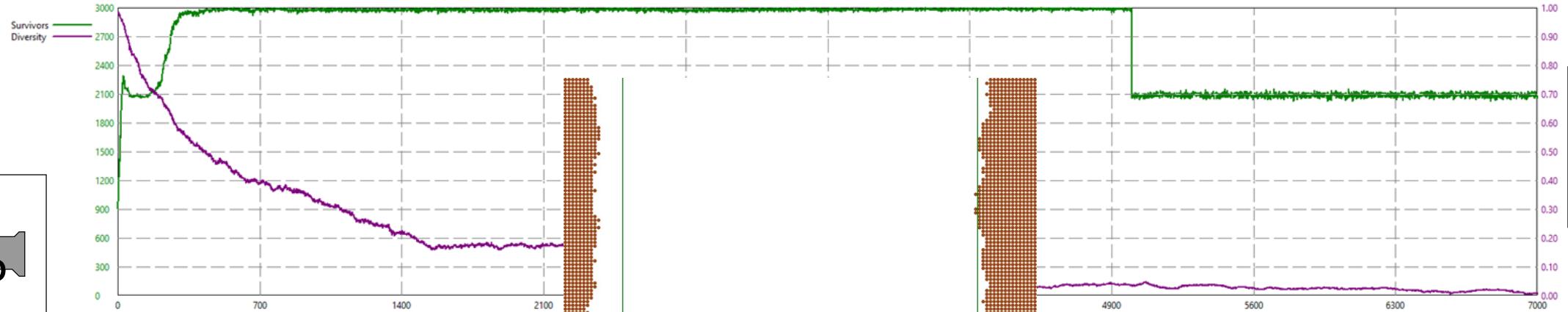
# Challenge 1 right half (go east ?)

**STIHL**



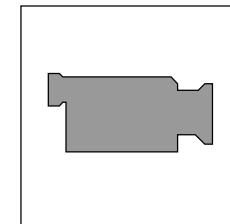
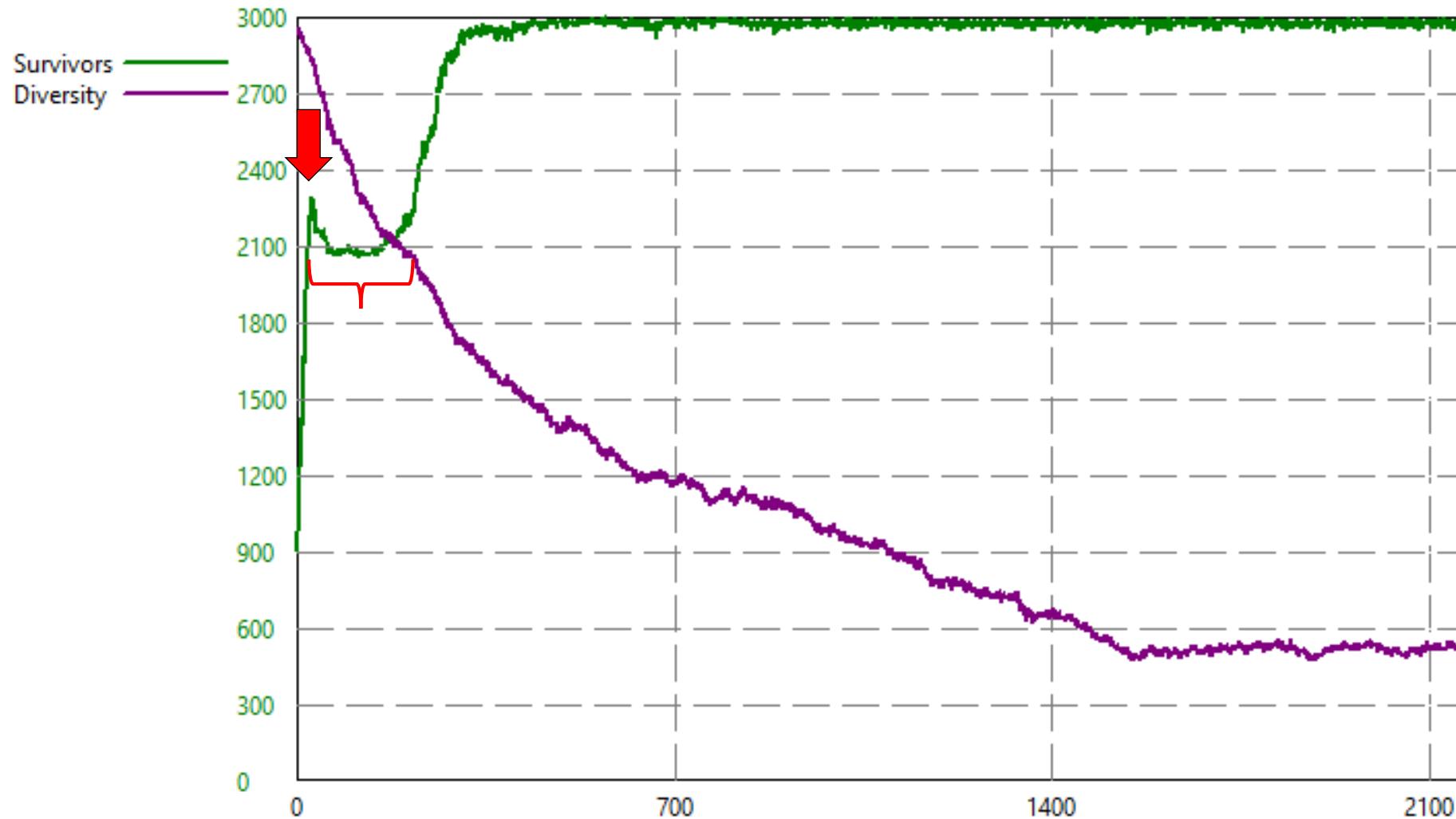
# Challenge 13 left / right eights (what are mutations for ?)

**STIHL**



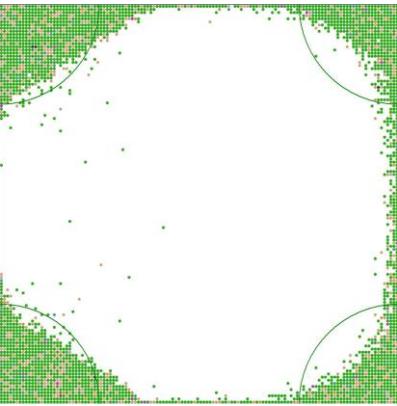
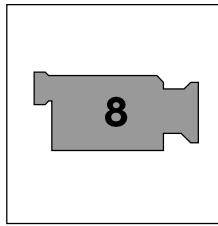
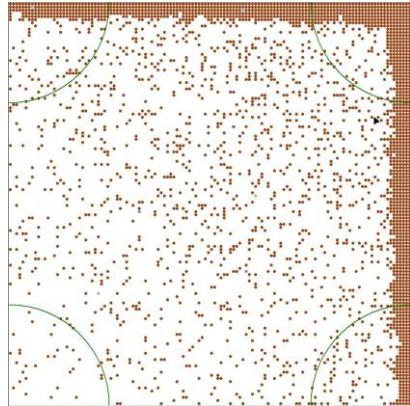
# Challenge 13 detail analysis (correct the overshooting)

**STIHL**

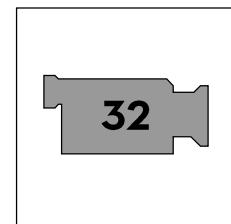
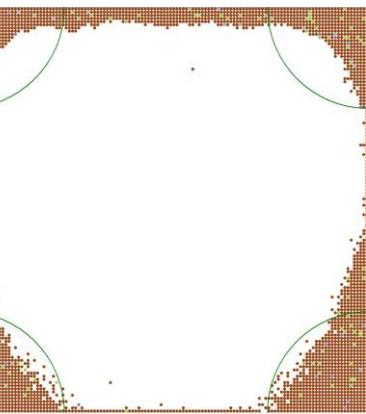
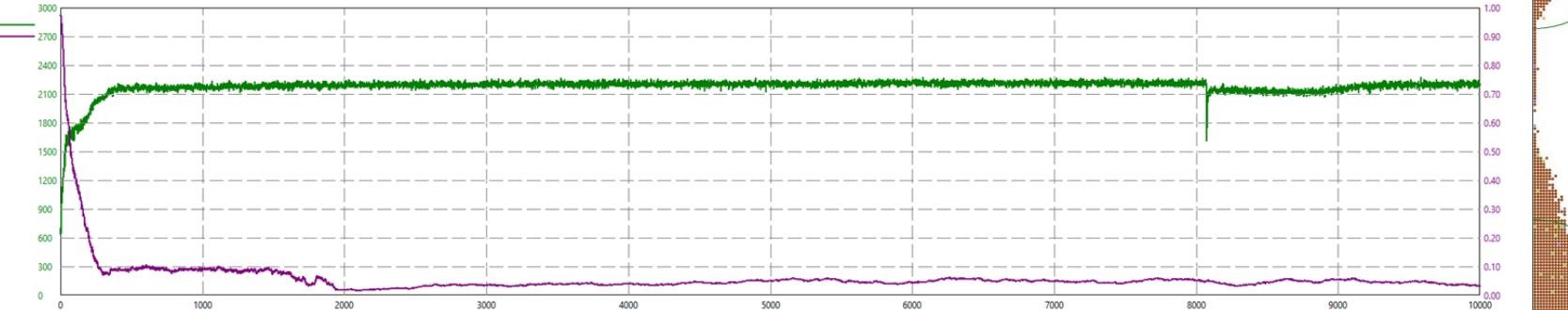
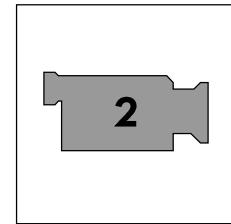


# Challenge 6 weighted corners (does brain size matter ?)

**STIHL**



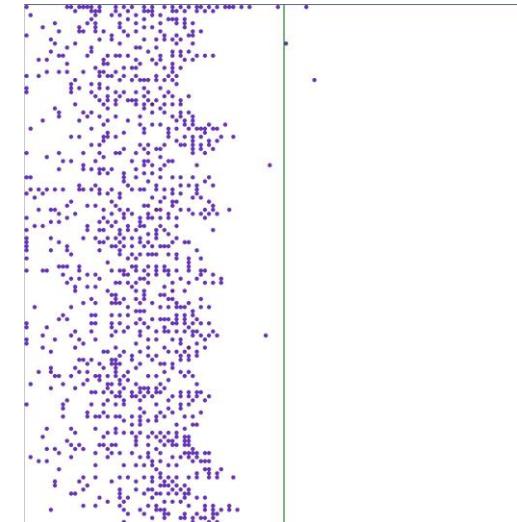
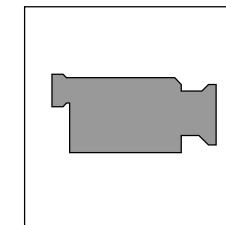
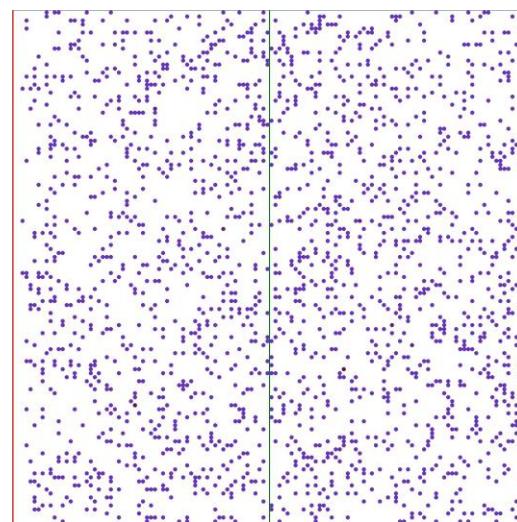
© ANDREAS STIHL AG & Co. KG



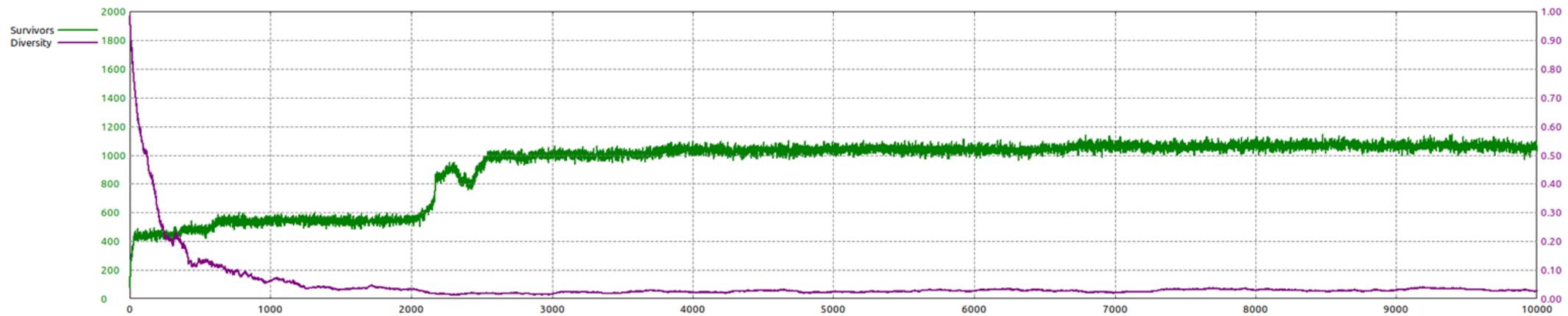
# Challenge 10 Radioactive walls (will they survive?)

**STIHL**

First half  
left side is  
radioactive

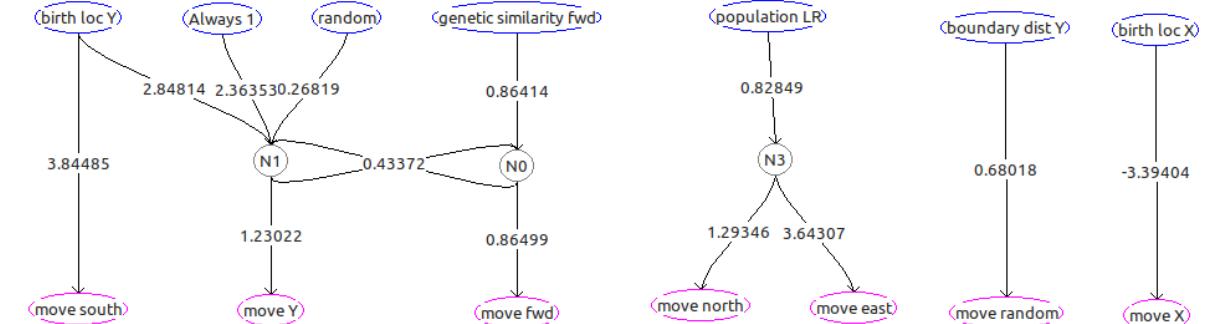
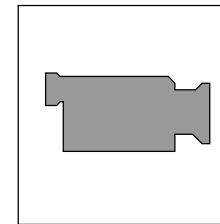
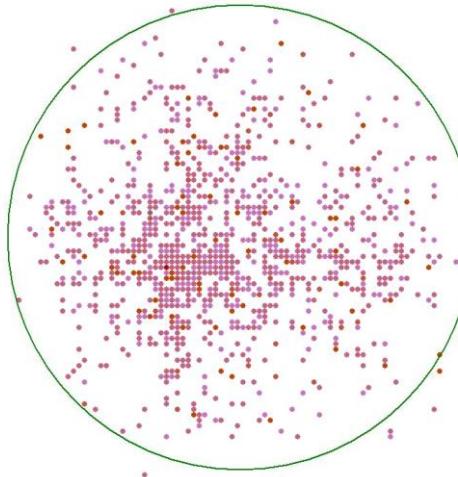


second half  
right side is  
radioactive

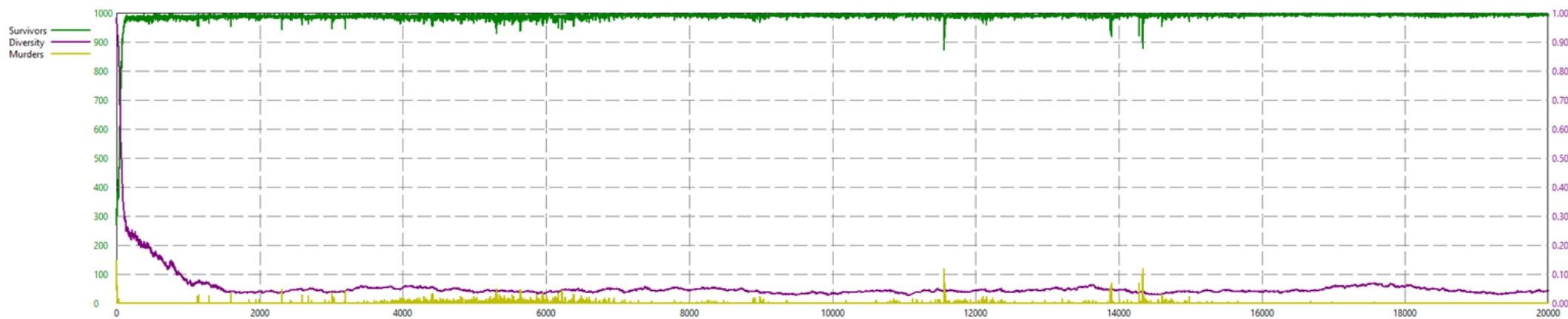


# Challenge 40 unweighted center (will the kill others?)

**STIHL**

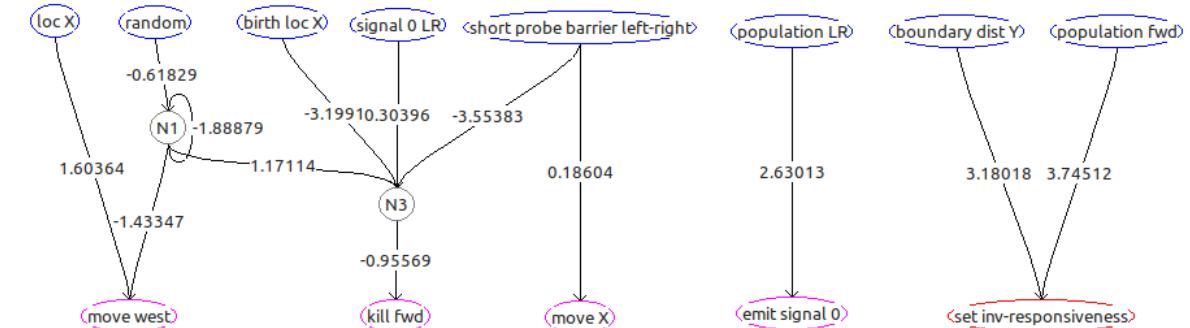
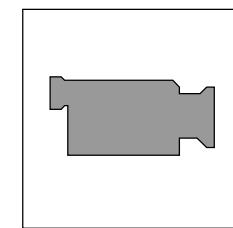
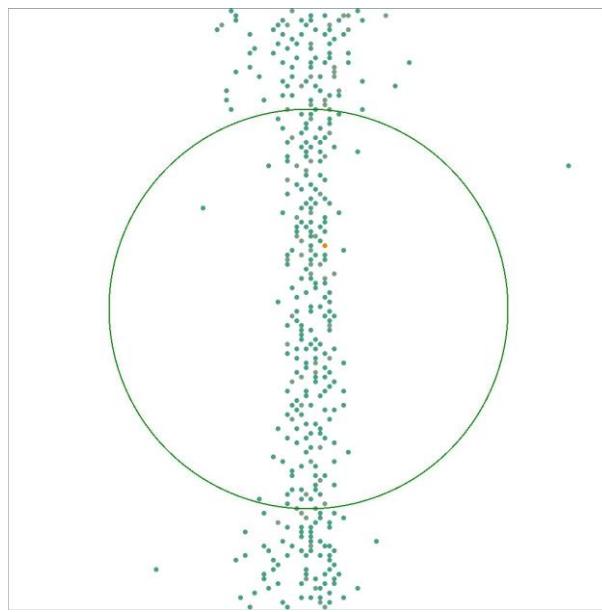
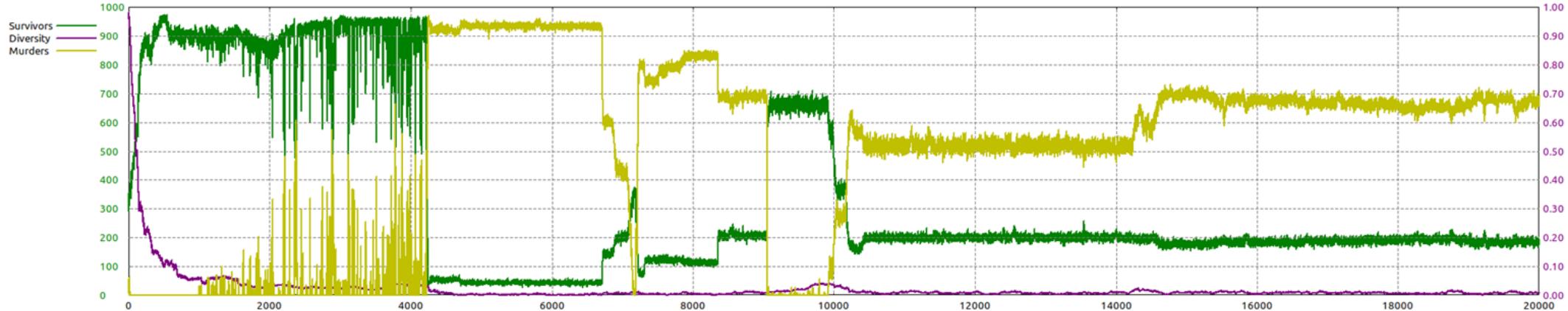


- Enough space for every individual
- Live in „harmony“



# Challenge 4 weighted center (will they kill others now?)

**STIHL**



- Bistable (killing, harmony)
- Killing makes the population behave like a population with small brains

Initial Execution time was 3:38 min (on Linux)

- acceleration of ~10s when new comfort features where moved from .ini to defines
- further improvement of ~2:03min by optimizing original Free Pascal **tanh** function
- Benchmark, done with deterministic0.ini, single threaded

	Windows	Linux
C++	2:26 min	0:56 min
FPC	1:04 min	1:25 min

Windows: Intel® Core™ i7-1185G7 CPU @ 3.00GHz × 8

Linux: Intel® Core™ i5-4210M CPU @ 2.60GHz × 4

- Single threaded vs. multi tredaded <https://www.lazarusforum.de/viewtopic.php?p=131483>
  - Windows scheduler seems to have a 16ms window  
tasks faster than 16 ms will be stretched to 16ms
  - Linux futex, no measurable waittimes

1. Ability to pause, shutdown or restart the simulation
2. Extra Gene Editor
3. Additional sensors introduced (birth location\*, always1)

Was it worth the effort ?

- Yes, one compact, library-independent code base, much shorter compilation time.  
Still a bit slower, but with lesser code complexity.
- No need for external tools (compiles directly on both Windows and Linux plattforms  
(MacOS not tested))

Potential future work:

- Design more challenges
- Understand how the individuals are actually working
- Implement more features like (balanced kill function, Energy cost system, ..)

Sourcecode and documentaion avialable under:

[https://github.com/PascalCorpsman/biosim4\\_FPC\\_translation](https://github.com/PascalCorpsman/biosim4_FPC_translation)