BIOSIM4
PORT FROM C++ TO FREE PASCAL

Is it worth the effort?

Uwe Schächterle
3.7 – 7.7.2023

About the speaker

Master degree in Computer Science at the university of Stuttgart

Master Thesis: Modellierung und Simulation bildgebender Systeme für partikelbeladene Strömungen

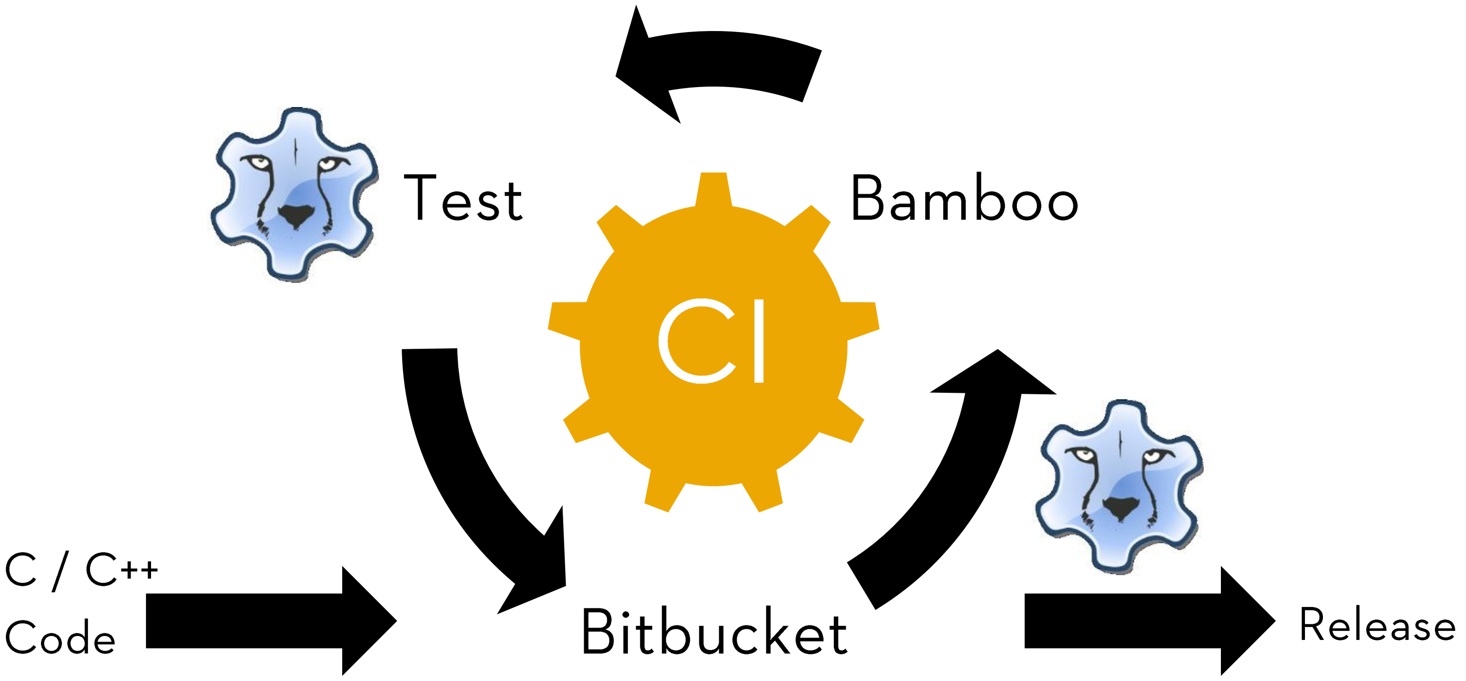
Embedded software engineer since 9 years (C and Free Pascal)

STIHL employee since 2018

Responsible

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

STIHL and Lazarus



Test

Bamboo

CI

C / C++
Code

Bitbucket

Release

Communication database

STIHL
AP 300

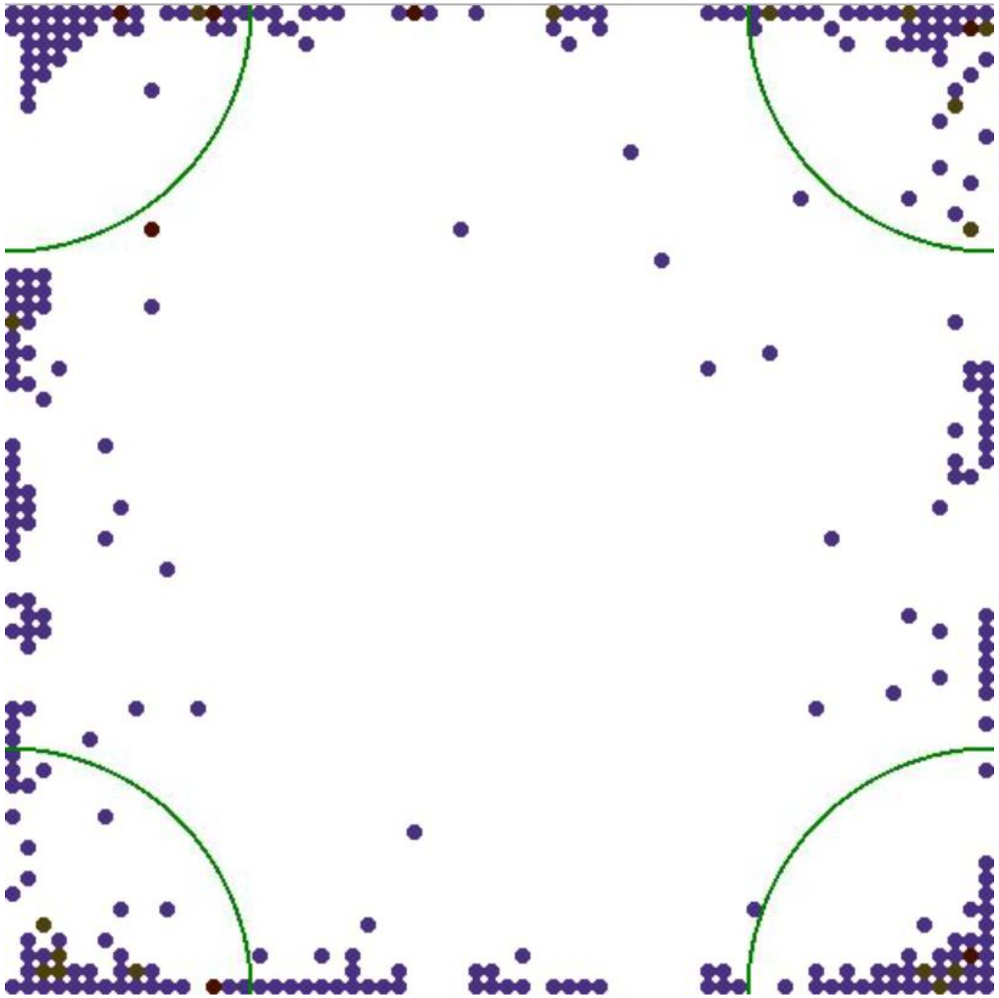
Introduction

- What is Biosim4 ?

- Starting point
- C++ Libraries

- How the code was validated

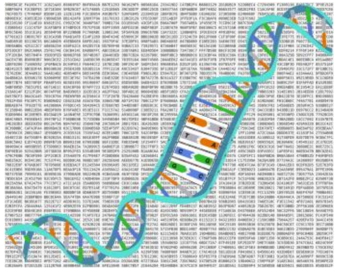
- Comparison
- Improvements compared to the original
- Conclusion



What is Biosim4 ?

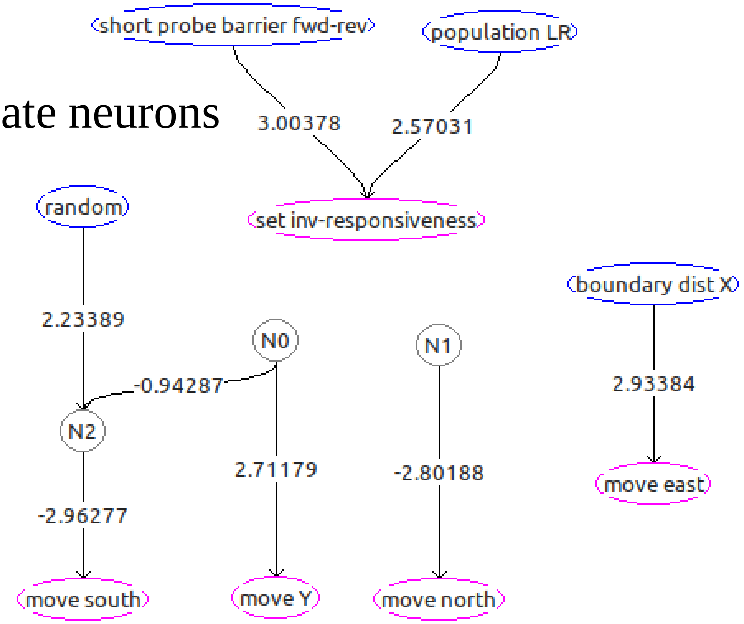
Based on the Book „The selfish gene“ by Richard Dawkins, David R. Miller programmed „Biosim4“

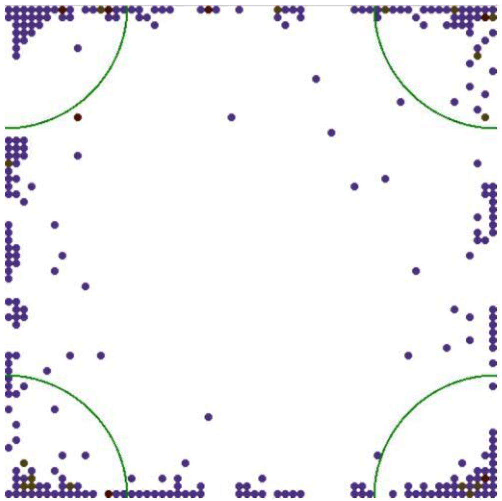
https://github.com/davidrmiller/biosim4

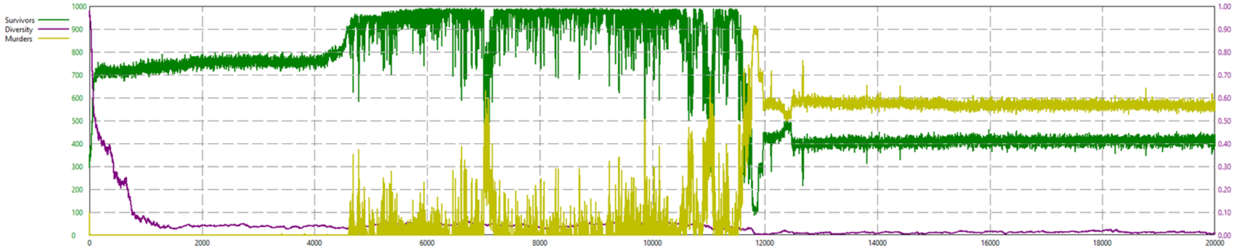


The individuals consists of

- actors
- sensors
- intermediate neurons





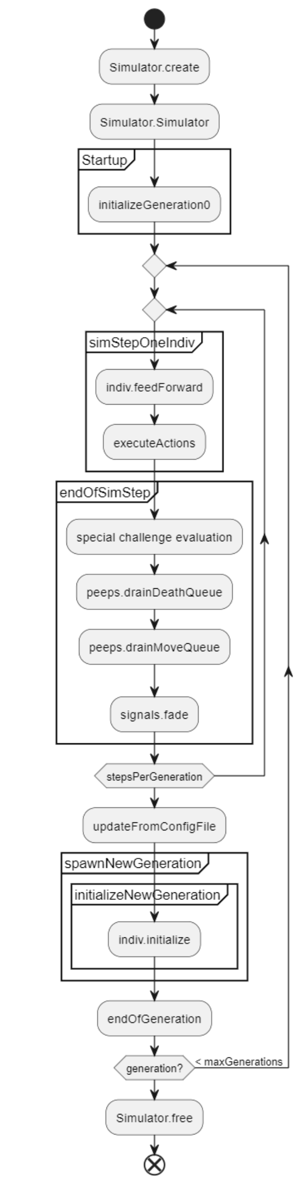


Starting point

- 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



C++ Libraries

- 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)
  -> ugwavi (ported from Michael Kohn , Robin Hahling from github)
  https://github.com/rolinh/libgwavi

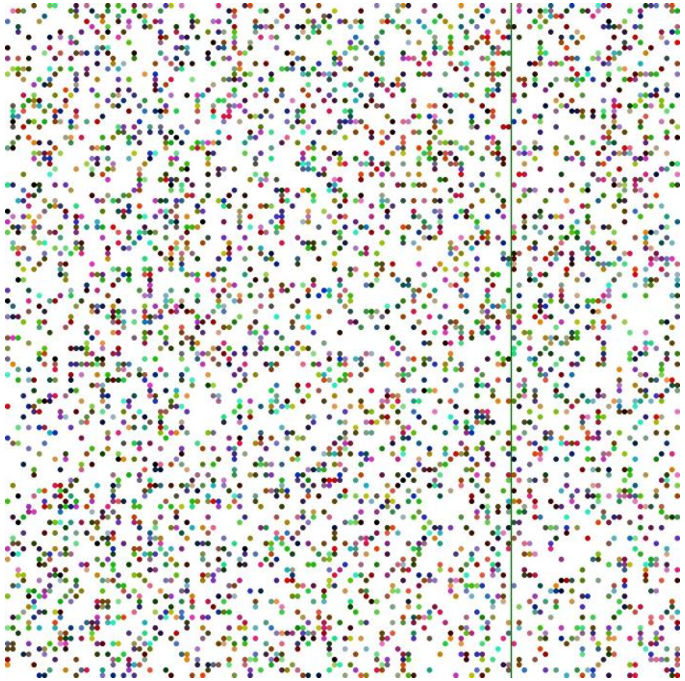
- gnuplot (command line plotting tool)
  -> usimplechart (own implementation)
- igraph (collection of network analysis tools)
  -> ugraph (own implementation)

How the code was validated

1. FPC-Compiler / Heaptrace found lots of div by zero and memory leaks
2. Unittests (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) ✓
   - -> Challange 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

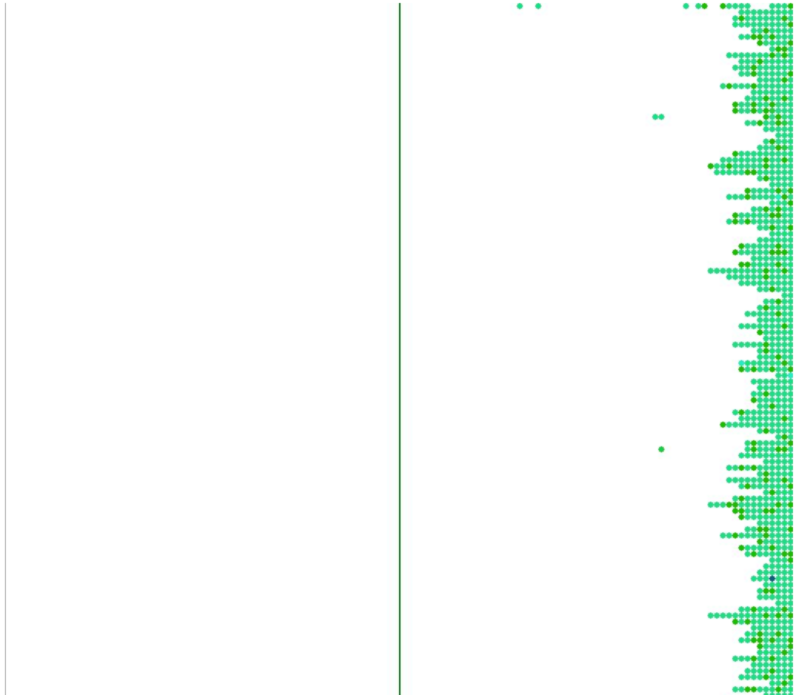
=> Survivel of the unfittest works, if the problem is easy

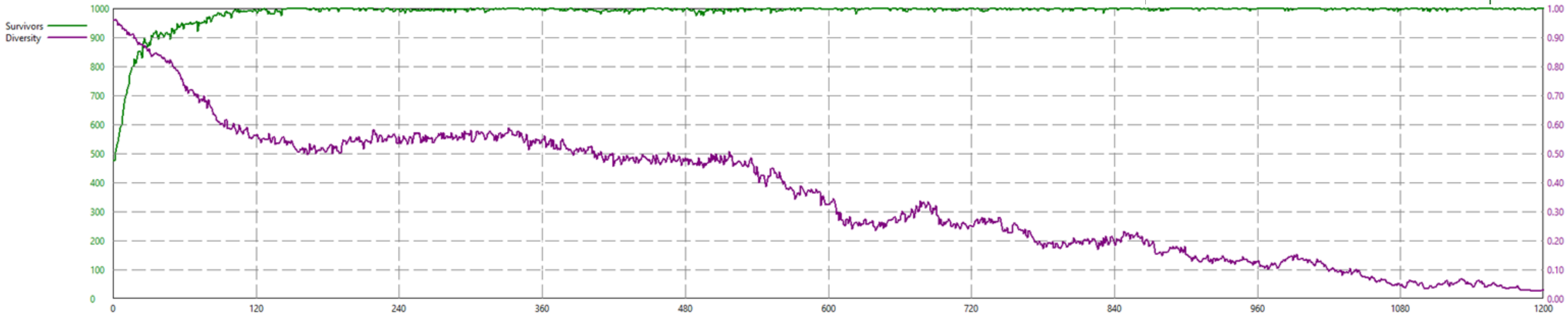
Challenge 1 right half (go east ?)



Gen 0

Final





Comparison

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 optimizating 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 treaded 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

Some improvements added

1. Ability to pause, shutdown or restart the simulation

2. Extra Gene Editor

3. Additionl sensors introduced (birth location*, always1)

Conclusion

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