

# Framsticks

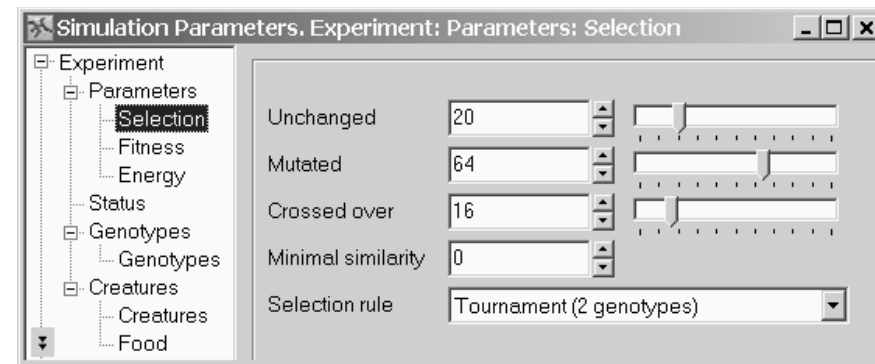
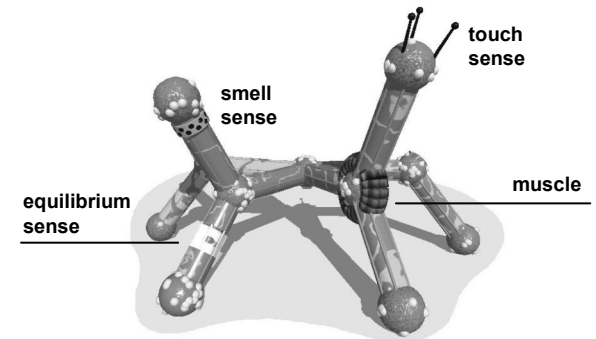
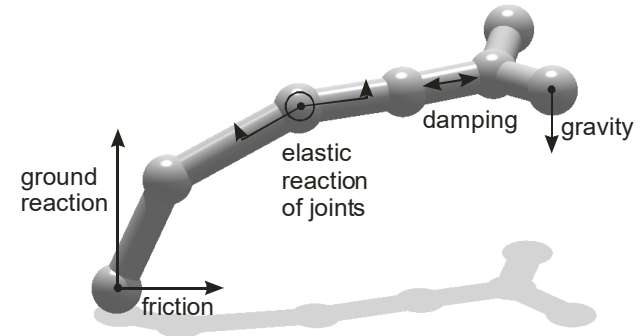
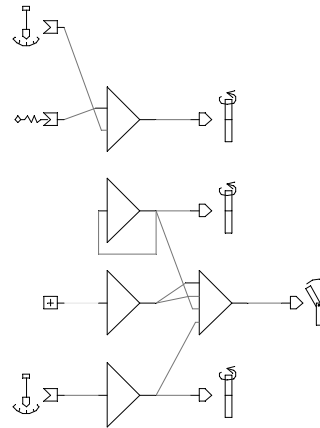
## general information

# General information

- developed since 1996
- authors and main developers: Maciej Komosinski (CS dept. at PUT) and Szymon Ulatowski
- other people involved in technical support, development, and experiments

# Main points of users' interest

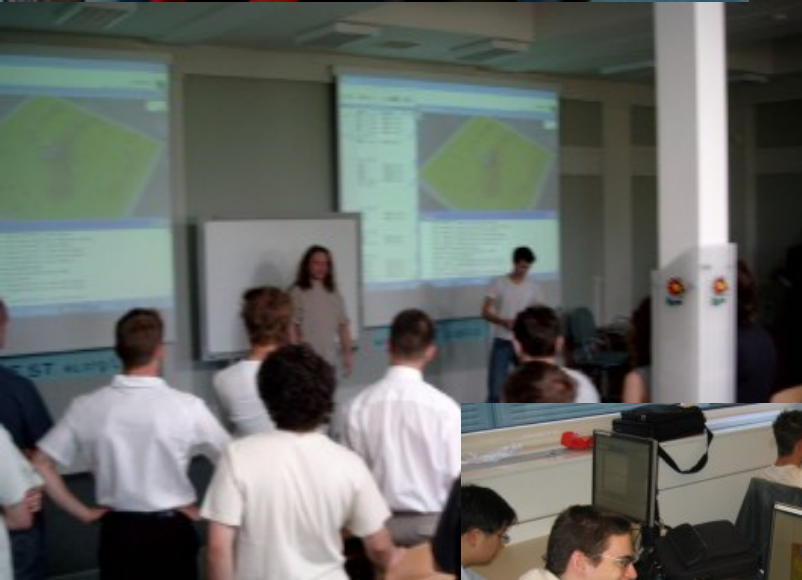
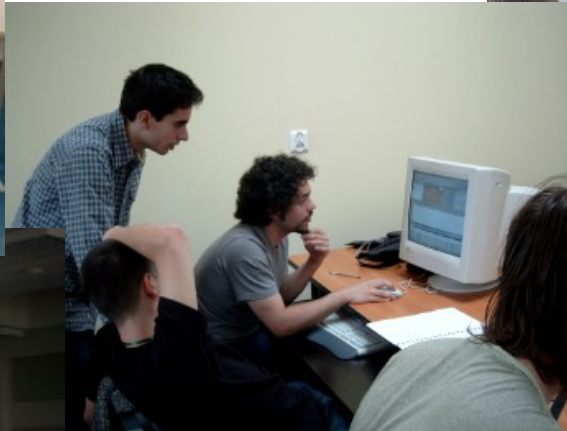
- simulation
- biology, evolution
- robotics
- neuroscience
- cognitive science
- computer science
- visualization
- education and understanding
- simplicity / complexity
- entertainment
- versatility



# Users

- regular users
- students
- teachers and scientists:
  - Virtual Life laboratory, Utrecht University, Netherlands
  - Bio-inspired Adaptive Machines Course at Autonomous Systems Lab, Lausanne, Switzerland
  - Cognitive Science Lab., Dept. of Philosophy, William Paterson University of New Jersey, USA
  - ...
- advanced users from all over the world

# Users



# Events

- articles in paper and electronic magazines
- interviews for newspapers, magazines, radio, and TV
- lectures, seminars, presentations, and demonstrations in conferences, workshops, academic institutions and popular shows
- third-party demonstrations (artistic exhibitions, thematic presentations – history of technology, evolution, medicine, etc.)

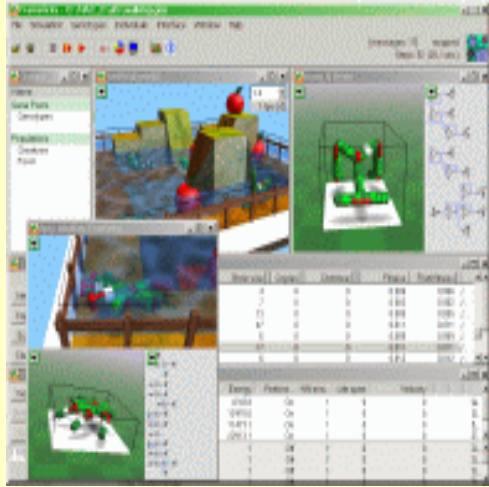
# Presentations invited by

- LEGO Lab, University of Aarhus, DK.
- TheoLab, Jena, DE. Friedrich Schiller University. Research Unit for Structure Dynamics and the Evolution of Systems
- University of Dortmund, DE. Chair of Systems Analysis, Department of Computer Science
- Max Planck Institute, Lipsk, DE.
- Santa Fe Institute, USA.
- European Summer School, PL
- Univ. of North Carolina at Charlotte, USA

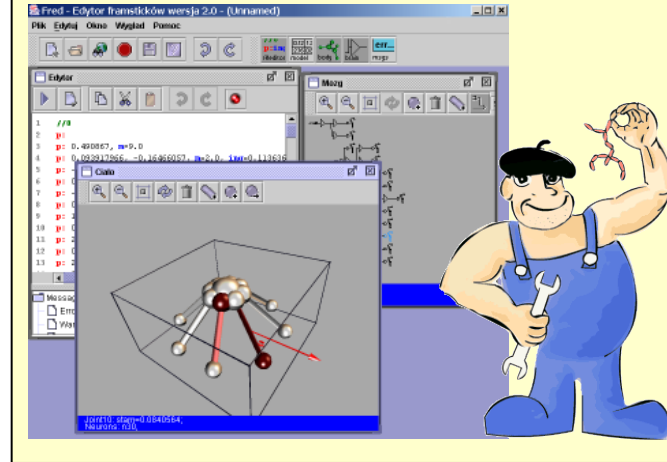


# Software

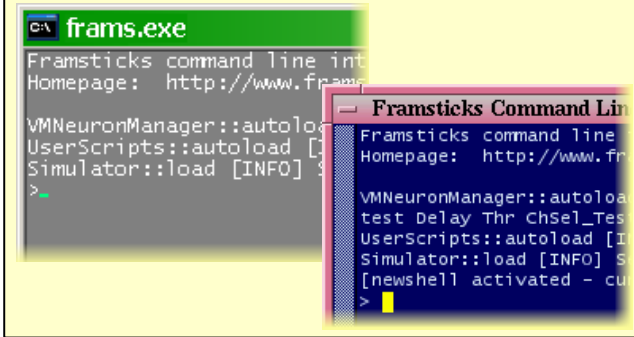
## Simulator GUI



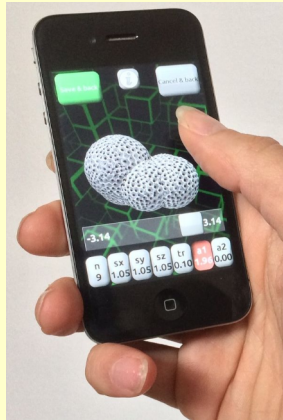
## Visual editor



## Simulator command-line and network server



## Artificial Life Lab (mobile app)



## Native library with C++ and python bindings

```
class FramsticksLib:
    def getSimplest(genetic_format) → str
    def evaluate(genotype_list: list[str]) → list[dict]
    def mutate(genotype_list: list[str]) → list[str]
    def crossOver(geno_parent1: str, geno_parent2: str) → str
    def dissimilarity(genotype_list: list[str]) → np.ndarray
    def isValid(genotype_list: list[str]) → list[bool]
```



## Command-line server

C:\> frams.exe

Framsticks command line interface  
Homepage: <http://www.framsticks.org>

VMNeuronManager::autoload  
UserScripts::autoload  
Simulator::load [INFO] S

[newshell activated - cu  
>

Framsticks Command Line

Framsticks command line  
Homepage: <http://www.framsticks.org>

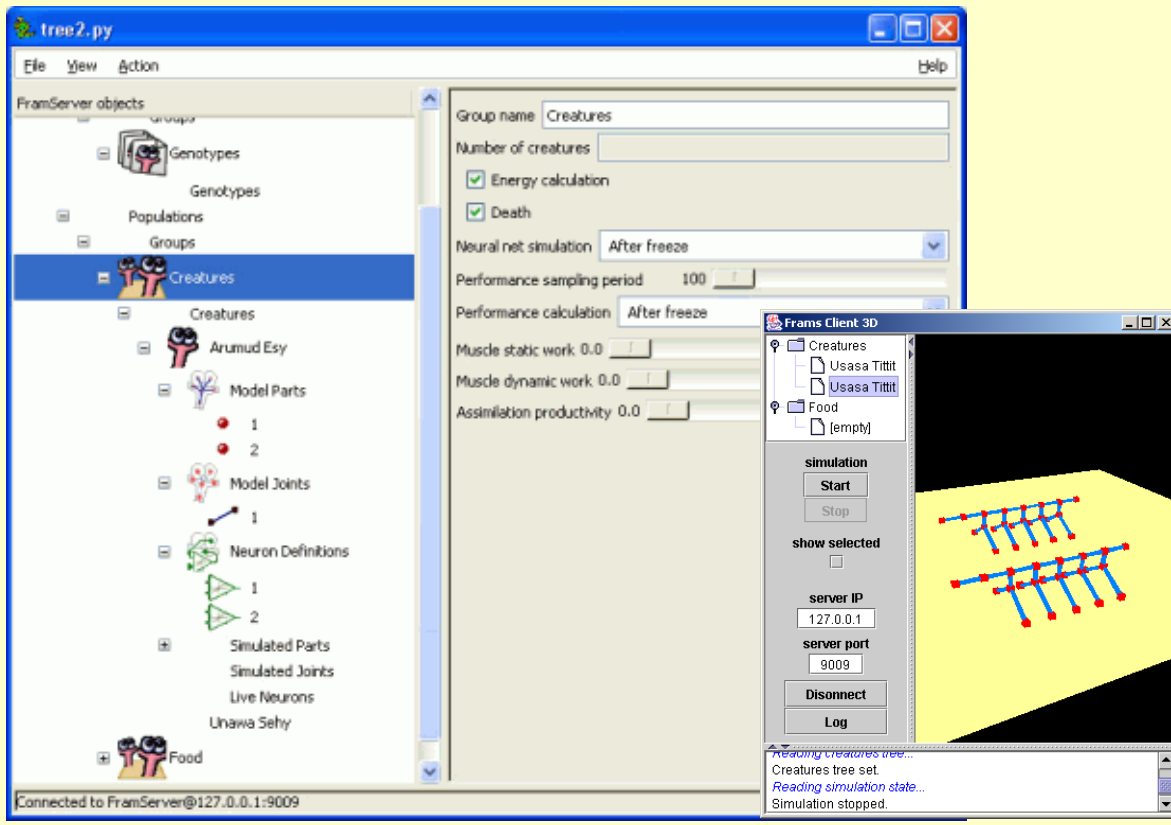
VMNeuronManager::autoload  
test Delay Thr ChSel\_Tes  
UserScripts::autoload [I  
Simulator::load [INFO] S  
[newshell activated - cu  
>

# Network Software

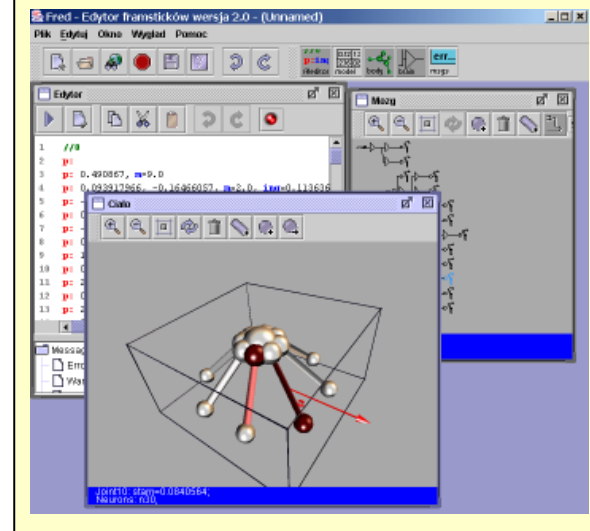
## Experimentation Center



## Portable clients

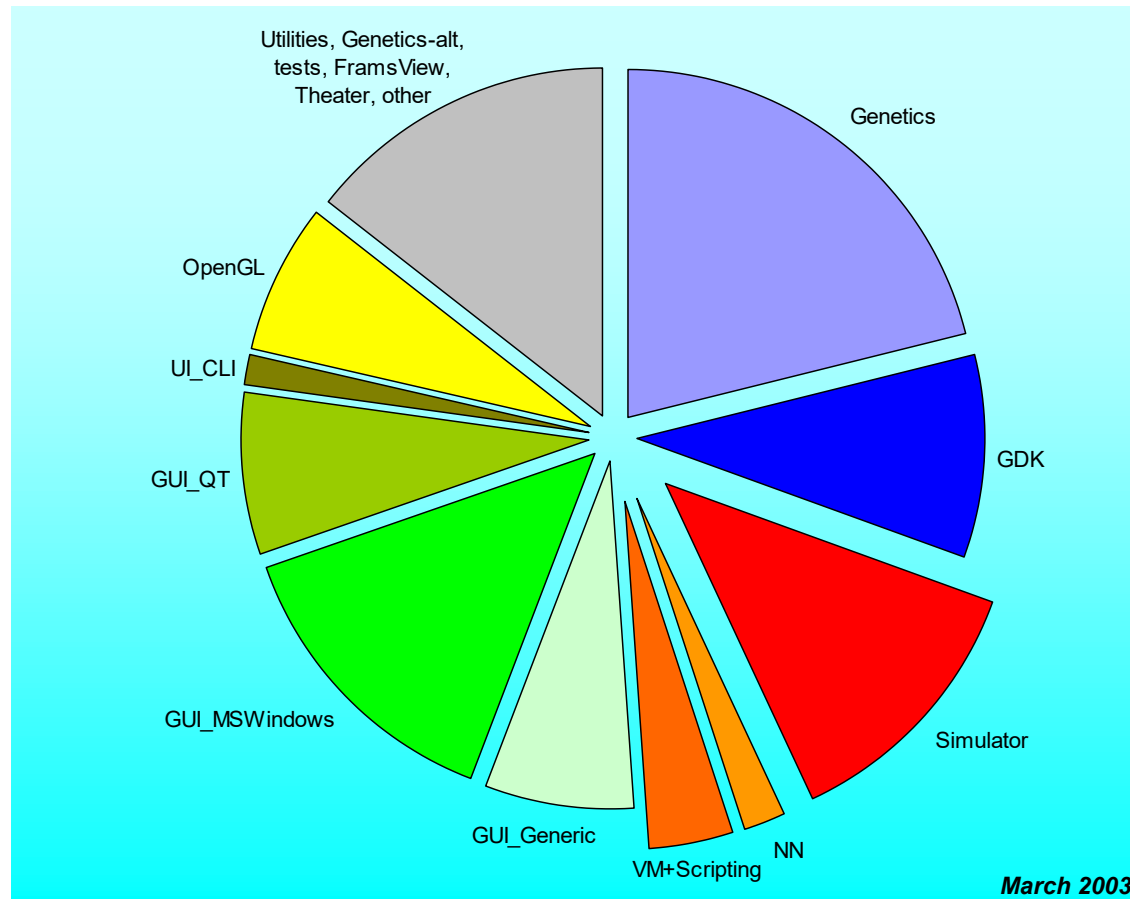


## Visual Editor – FRED



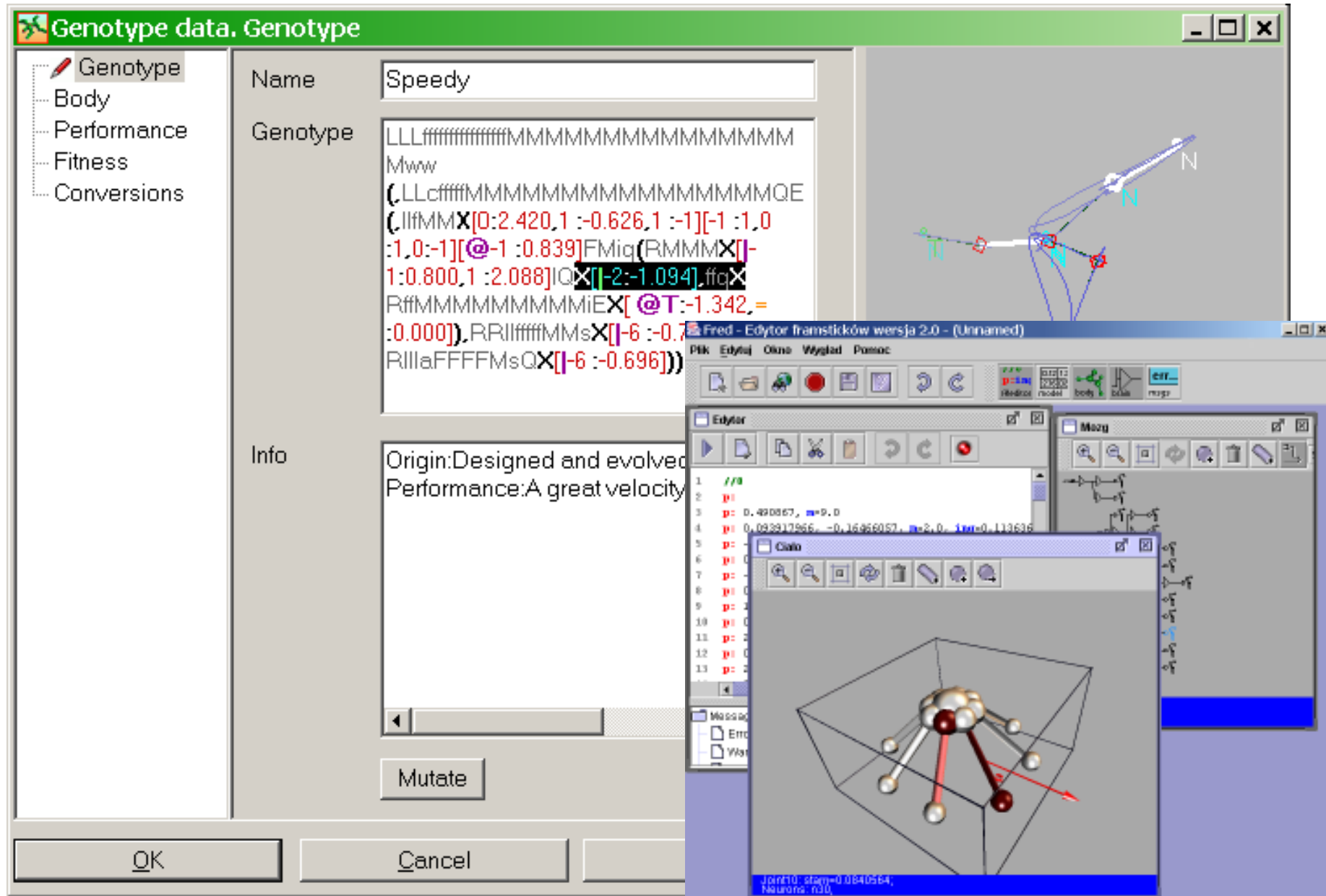
# Technical information (outdated)

- Sources: C++, lex, bison, m4, awk
- Sources available in SVN repository – SDK
- Total source:  
78 KLOC, 2 MB
- third-party libraries:  
PLIB, GLPNG

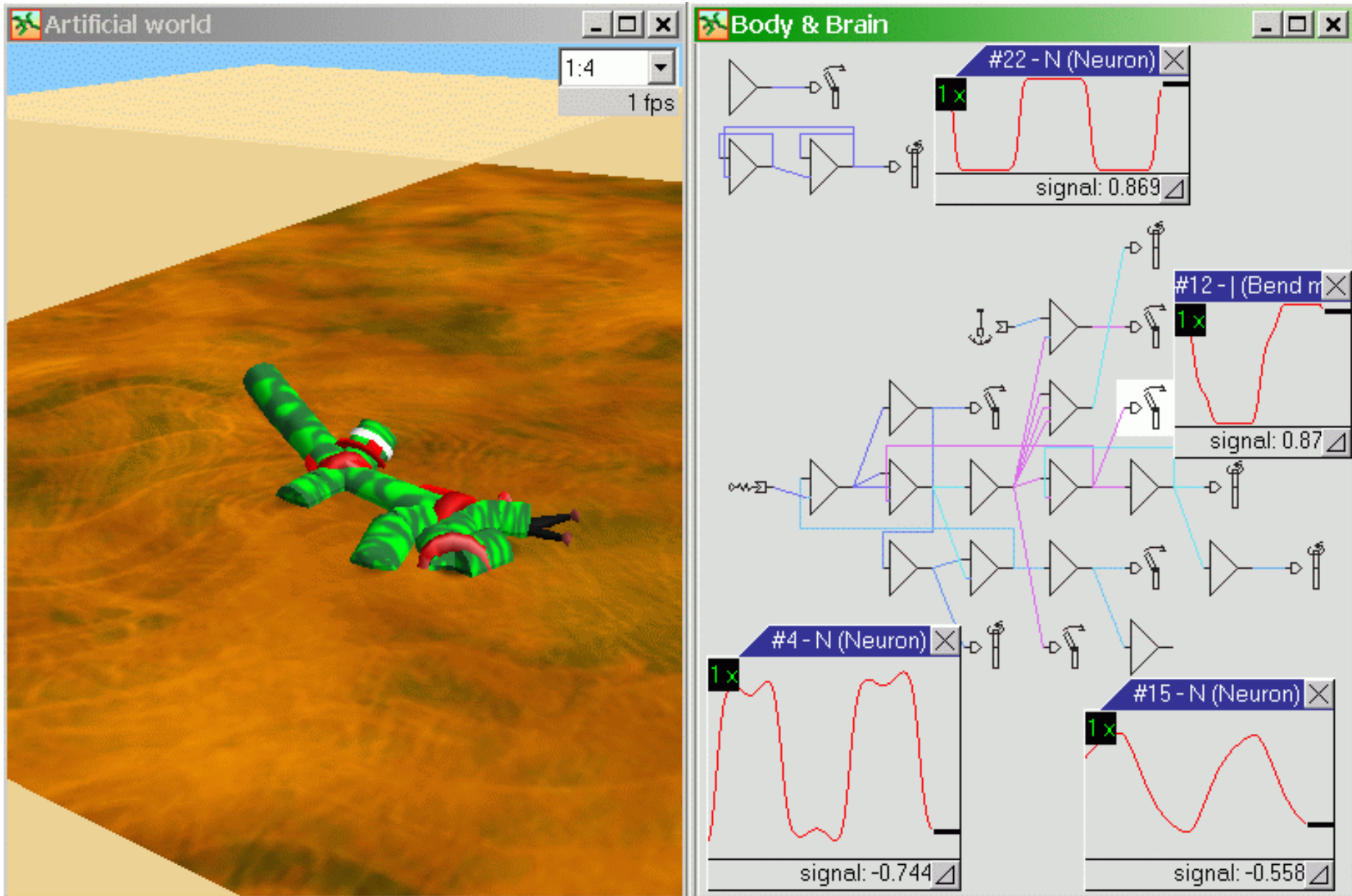


# Sample uses and experiments

- synthesizing (building) agents
  - studying agents' behaviors
  - optimizing agents
  - designing genetic representations
  - studying evolutionary dynamics, coevolution, migration, etc.
  - evolving neural and fuzzy controllers
  - understanding evolved brains
  - evolution of communication
  - designing custom user experiments
- 
- publications available from the web site



# Studying agents' behavior







# Potential behaviors

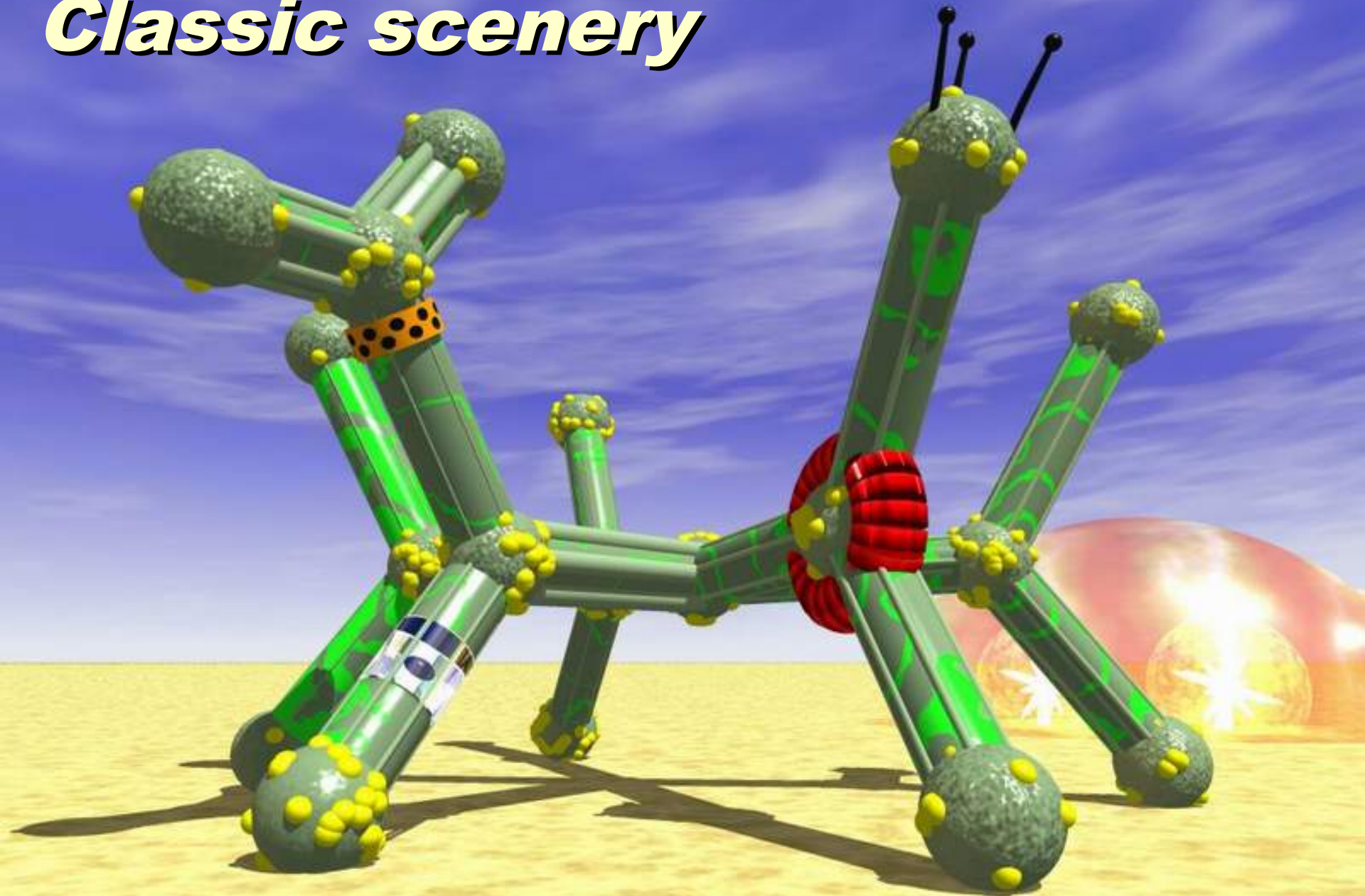
- walking/swimming/jumping/rolling/...
- memory
- predation, prey
- symbiosis, cooperation
- mutual identification and location
- preferences, group/social behaviors
- communication
- feelings, consciousness, ...?
- ...they discover, learn and utilize simulator imperfections!



# Framsticks: an open system

- custom fitness functions
- experiment definition scripts for custom definition of system framework; user-defined neuron types
- support for various genetic representations
  - conversion to the basic format
  - genetic operators
- network submission of experiment proposals and interesting genotypes; Experimentation Center
- discussion forums for users and developers
- custom definitions of visualization rules
- open source projects

# ***Classic scenery***

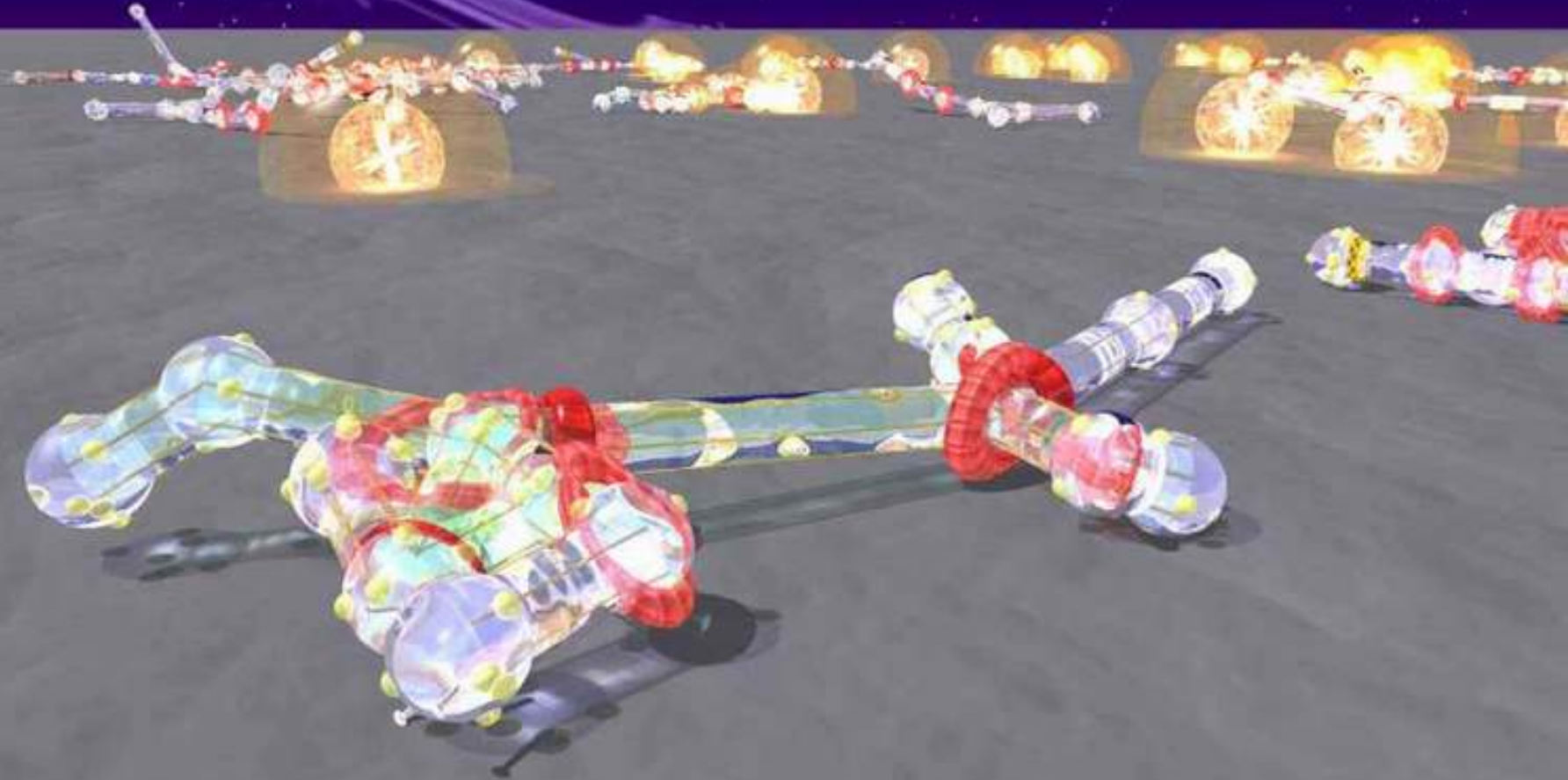


# ***Planet scenery***





# ***Planet scenery***



# ***Blocks scenery***





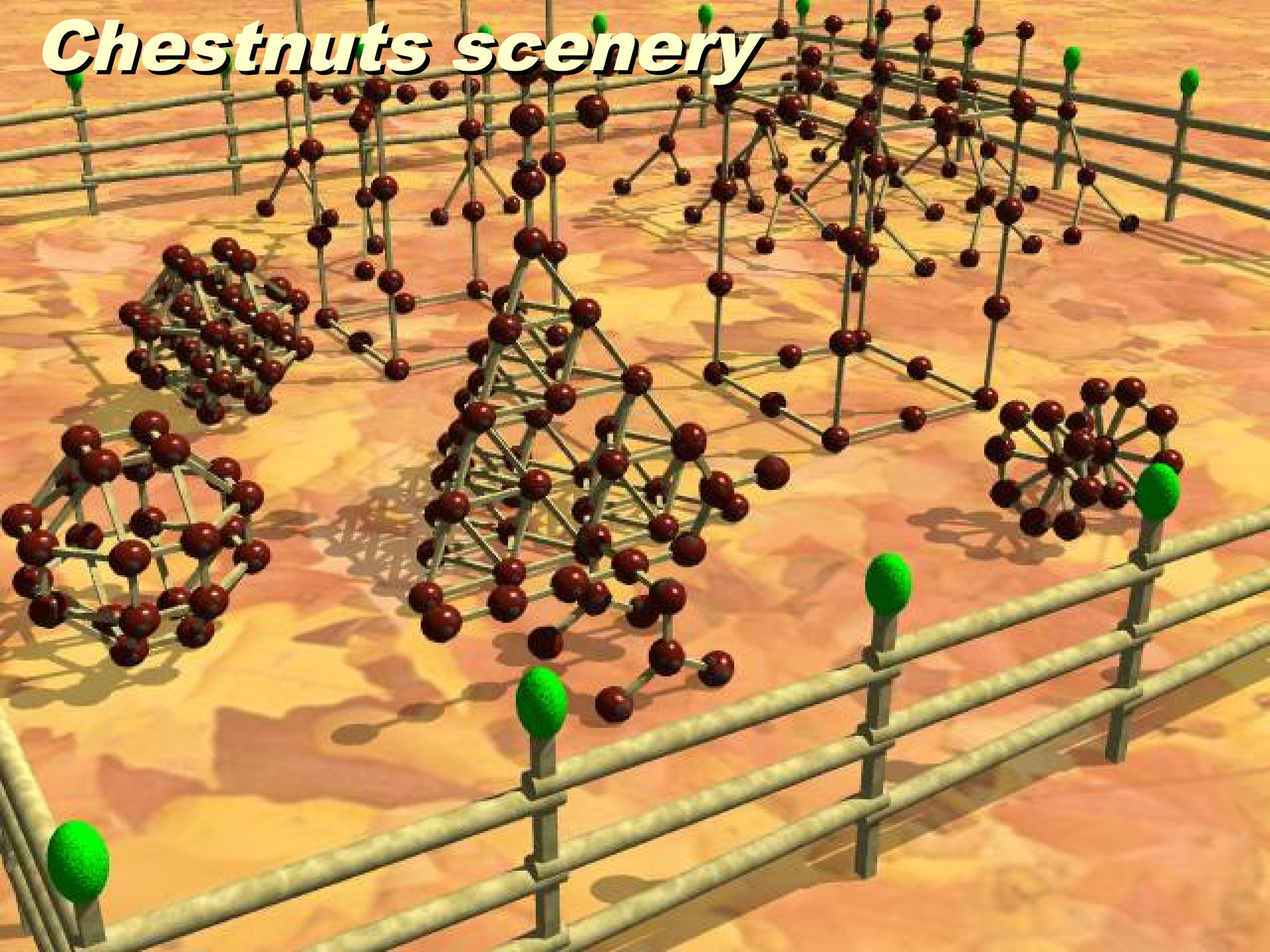


# ***Ghost scenery***





# ***Chestnuts scenery***



# ***Wookiee scenery***



# Further development and research

- network programs, distributed and parallel evolution
- other genetic representations
- more complex tasks / environments
- tools for analysis of emerged behaviors
- open-ended and spontaneous evolution
- more sensors/effectors (e.g. communication), more fitness criteria
- open system; cooperation

# Inspiration for EC, AL, and KD/ML

- decomposition of hard optimization problems
- hierarchical representation of solutions
- effective crossover operator and speciation
- measures of similarity of complex solutions; global convexity of the search space
- coevolution of solutions and constraints/fitness function
- properties of various solution encodings
- automatic analysis of evolution and agent behaviors
- active perception