ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Programming Techniques II
Exercise 13

HS 15
Prof. M. Troyer

Task 13.1   Preparation

Install Boost Python from your operating system's package repository (apt, rpm, port, brew) or download
the source from http://www.boost.org/ and follow the *build* instructions.

Keep in mind that the Python version boost::python is compiled for has to be exactly the version of the
Python interpreter you are using to run the importing code.

Task 13.2   PyPennaLV

Upgrade your Python Penna simulation to pennaLV by introducing the `bear` class and the interactions
between `bear` and `sheep`. Try to get results that are qualitatively similar to your C++ version.

Task 13.3   Boost Python

Use boost::python as shown in the lecture (the example code can be found in the lecture repository under
`tools/boost.python`) to compile the random number generator at
`lecture/exercise/extern/include/util/random.hpp`. You can then move the object file into your
Python code directory and import it as a module in your Python implementation. Try to achieve the
exact same numerics with your C++ and Python codes.

Task 13.4   Sphinx

Document a few functions and classes in Python using Sphinx and generate the HTML output. Consult
the wiki entry about documentation on the repository to set up Sphinx.

Task 13.5   Git Bisect

Find out on which date you got rid of the polymorphic animal base class. `git bisect` together with a
helper script in any language is a very efficient tool for this task.

Enter into bisect mode with `git bisect start`, set the `bad` and `good` commits, launch a `run` and don't
forget to return to your working state with `git bisect reset`.

Task 13.6   Fastest Python Challenge (optional)

Make your PennaLV implementation from the PyPennaLV task as fast as possible. You are only allowed
to use "standard" Python libraries (numpy and scipy qualify, whereas a boost::python import of a C++
implementation doesn't).

The performance will be measured in seconds with the `time` command line utility.

The submission deadline for this challenge is Wednesday 16.12.15 at 05:00 in the morning. Notify us
via the mailing list (pt2_hs15_ta@lists.phys.ethz.ch) with a link to your uploaded solution on your PT2
repository on GitLab.