PiE 2022 python module – final Project

please read carefully!

1. List of final projects

Sample project 1. Create the prototype of a discrete element code. Assume that every particle is a point mass, that does not interact with another particle unless the intercenter distance between two particles is less than sum of their radii, and repel with the normal force linearly proportional to overlap between particles. Use velocity Verlet algorithm (https://en.wikipedia.org/wiki/Verlet_integration) to compute the motion of particles. Employ the brute force algorithm – *N(N-1)/2* checks – to detect new contacts between particles at very step of time integration. Model the pool game – motion of the resting particles after hitting by one with the large velocity. To ensure the stability of time integration, employ a stable timestep $dt = 0.01\ (M/k)^{1/2}$, k is the coefficient of proportionality between overlap and contact force ($\overrightarrow{f_{ij}} = k\left(\overrightarrow{r_{ij}}/r_{ij}\right)(r_{ij} - R_i - R_j)$), M is the reduced mass of colliding particles ($M = m_1m_2/(m_1+m_2)$, for equal particles, $M = m/2$, m is the mass of the particle). ME: visualize your results with Paraview using the interface function from Lecture 4.

Sample project 2. Create a prototype of celestial mechanics code. The principles are the same as for the discrete element code, but the interaction between every couple of particles *i* and *j* is based on the gravitational potential:

$$\overrightarrow{f_{ij}} = \frac{\overrightarrow{r_{ij}} G m_i m_j}{{r_{ij}}^3} = \overrightarrow{r_{ij}}(k(r_{ij})),$$

where m_i, m_j are masses of the interacting particles, r_{ij} is the vector connecting centers of *j* and *i*. Note that every particle interacts with every other particle at every step of time integration. Also note that the stiffness k is changing, and is singular at $r_{ij} = 0$. Use cutoff radius to avoid the singular part of the potential and define the stable timestep. Check the validity of the code on the known two-body solutions (ellipses, parabolas, hyperbolas – see https://en.wikipedia.org/wiki/Two-body_problem). You can also approach one of the known solutions of three body problems (https://www.youtube.com/watch?v=8_RRZcqBEAc). Model a system of 100 small planets revolving around the Sun. Make sure you see the effect from both the planet-sun interaction and perturbations caused by presence of other planets. ME: visualize your results with Paraview using the interface function from Lecture 4.

Sample project 3. Create a program that stores the periodic table of chemical elements and draws it using the capabilities of Matplotlib. You can use **pandas** library to simplify the code, but you are encouraged to try designing your own, user-defined memory structures based on python built-in types. Use one of the csv files available online (e.g. https://gist.github.com/GoodmanSciences/c2dd862cd38f21b0ad36b8f96b4bf1ee) containing the information of chemical elements. Try to create a periodic table with clean representation of every element, that would be suitable for chemistry classroom. ME: add some user interface interactivity using **pyqt** library (e.g., the detailed info on the element by mouse click).

Custom project. As the final project, you can develop something that suits better to your interests and area of specialization. In this case, please remember that:

- If you opt for a custom project, please submit your proposal (title of the project and its short description) first. Once your proposal is approved, you can start working on the project.

- The project should be written using the tools that have been studied in the course. The goal of the project is to display that you are comfortable with concepts discussed in the Lectures. You may want to use some extra functional that was not covered in the course (e.g. **csv** lib for easier input-output, or **plotly** for graphics) but you submit something like a trade robot that uses 10 different dependencies (pandas/scikit learn/tensorflow/pytorch…), I will likely return your submission back to you. Also, please do not submit platform-specific, commercial or partially closed source.
- Your project may be the part necessary for your PhD, but please be advised that the teacher does not have an opportunity to get into 100 different PhD projects – please keep the project high level enough to be easily understood by the person outside of your area of expertise.
- Remember that your project is expected to correspond to approximately 15-20 hours of independent work (not half a year, not 5 minutes).

2. What/how to submit?

Your final grade is based 100% on your final submission. If should consist of two parts:

- Report
- Source code

The report should be 5-10 pages long. It should have a title page with

- Your name,
- name of the class,
- title of the project,
- name of the instructor,
- the date.

The report should contain

- the minimal necessary introduction,
- problem formulation,
- all the underlying math,
- description of the code and the necessary instructions on installation of non-standard dependencies used (if there are any),
- results obtained,
- list of references.

The report should be submitted in pdf format.

The code should be well-structured and documented. Please place the functions in the respective *.py files with at least one comment explaining what the function does. The main script may be in either *.py or *.ipynb file. Please do not introduce magic numbers or numerical constants in the middle of the code – all the constants should be defined in the beginning of the file, e.g.

HOURS_IN_DAY = 24.
MINUTES_IN_HOUR = 60, etc.

The report and the code should be unified in a single zip archive, and emailed to i.ostanin@utwente.nl. Please do not use Canvas for submissions. **In python module, the report does not have to be defended or presented –** you send a project and get the final grade.

3. What are the deadlines?

In the term 2022 the submissions will be accepted till **1.12.2022**. Those who submitted their projects before the deadline will receive their grades within one week. After that date, one can still submit the final project in a self -study mode, but the grade will be received in the end of the next term (May. 2023).

If you have any further questions regarding PiE python module, please e-mail me to (i.ostanin@utwente.nl).

Good luck!
Igor Ostanin.