

Project announcement

SE271 Object-oriented Programming (2017)

Prof. Min-gyu Cho

Project schedule (subject to change)

- 10/6 (today): Announcement of candidate topics
- 11/10: Topic suggestion
- **11/15: Accouchement of detailed requirements for N body simulation**
 - For those who are working on your own projects, refer these documents for the expected scope/complexity/user interfaces
 - For anyone, it is free to extend the scope of term projects
- 11/17: Project design submission due
- 12/11-13: Project presentation (tentative)
- 12/13: Project due
- Note: You will have a few programming assignments for the rest of this term

Project schedule

- 10/6 (today): Announcement of candidate topics
- 11/10: Topic suggestion
- 11/15: Accouchement of detailed requirements for N body simulation
 - For those who are working on your own projects, refer these documents for the expected scope/complexity/user interfaces
 - For anyone, it is free to extend the scope of term proejcts
- 11/17: Project design submission due
- 12/11-13: Project presentation (tentative)
- 12/13: Project due
- Note: You will have a few more programming assignments

N-body simulation

- Problem definition:
 - Simulate N particles interact with each other by gravity (or collision)
- What you have to implement
 - Implement a class for particles
 - Implement behaviors that simulate gravity, collision and other forces
 - Enable dynamic adjustment of parameters such as amount of forces, on/off of gravity, manual force
- See also: https://en.wikipedia.org/wiki/N-body_simulation

Reference: Meeting room management

- Problem definition:
 - Manage meetings; room #, attendance, time
 - Manage meeting schedule of a particular room
 - Manage meeting schedule of a particular person
- What you have to implement
 - Design classes that represent meeting, person and room
 - Read standard inputs for commands, and process them, e.g.,
 - Add/modify person
 - Add/modify meeting
 - Add/remove attendee
 - List meetings (for each person/room)
 - Manage conflicts; one meeting at a time in a certain room, one person can attend only one meeting at a time, etc.

3. Automatic and safe unit conversion library (TBD)

- Problem definition:
 - Provide easy-to-use library that guaranteed physical/chemical unit conversion; w/o such tools, it is very easy to go wrong
- What you have to implement
 - Implement classes that guarantees unit conversions, that means generate error/exception when something goes wrong
 - Start with SI units, such as meter, kilogram, and second
 - Provide automatic conversion of prefix, e.g., kilo, milli

Assessment criteria

- Correctness
- Performance (if applicable): time and memory
- C++ standard library usage (avoid reinventing the wheel)
- Coding style
- Object design
- Final documentation



ANY QUESTIONS?