Project Logistics

Timeline

- Commitment to Project Topic (with Instructor’s consent)
  by Monday December 2nd
  (recommendation : aim for before Thanksgiving)

- Check-in with instructor for progress report
  Friday December 6th

- Final deliverables
  by Thursday December 19th (last day of exams)

Instructor Availability

- Office hours for next 2 weeks:
  - **Thursday Nov 20 : 1:30-2:15**
  - **Friday Nov 21 : 2:30-3:15 (class meets 4pm)**
  - **Mon Nov 25 : 4:15-5:00 (no class)**
    Tue/Wed Nov 26/27 : 1:30-2:15 (no classes)
  - Other times by appointment (flexible)
  - Availability Dec 2-6 to be announced - Away Dec 9-13

Project proposal

- Ideal format : 1-page (or more if needed) write-up
- Communicate via email, or follow-up on office hours (ideally, both)
- Feel free to discuss during office hours first, and submit proposal write-up afterwards

Project deliverables

- Mandatory : Written report
- Most projects : Code executable, or in-person demo, visual products (gallery of images/screenshots/videos)
- Optional : Poster or PowerPoint/Keynote presentation

Project types

- “*Implement a Paper*”

- Replicate technique, and adequate set of visual products

- Ok to use existing code as reference, but if comprehensive implementation is provided, you need to add to it substantially

- Credit for rich visualizations, or nice rendering of results

Project types

- *“Improve on a Paper/Method”*
- Enrich the feature set demonstrated, e.g.
  - Implement more complex material models
  - Add collisions, if none exist
  - Add dynamics, if none exist
  - Implement a different type of time integration
  - Use different data structures (grids vs. meshes, etc)

Project types

- *“Improve on a Paper/Method”*
- Make method faster
  - Create parallel variant of algorithm
  - Port to GPU, or explore GPU accelerations
  - Explore matrix-free methods
  - Use a more effective numerical solver
  - Use more efficient data structures

Project types

- *“Use a Paper/Method for an innovative application”*
- Explore parallel to AI/Machine Learning applications
    - Create training data for a ML algorithm using simulation
    - Use ML-based techniques for collision detection
    - Implement a “differentiable” simulator as part of a neural net
    - Re-write solvers using ML/DL APIs (TensorFlow/PyTorch)
- Demonstrate parallels between simulation and your own project(s)

Project types

- “*Make the visual results of a Paper/Method prettier*”
  - Interface the results of the algorithm with a powerful renderer (e.g. RenderMan, Embree)
  - Write a plug-in for a simulation algorithm to be run inside Maya/Houdini/Unity

Project types

- *“Help enrich the documentation/infrastructure for future class offerings”*
  - Work with instructor to turn notes into textbook chapters (FEM notes created that way)
  - Help create a basic fluids simulation framework
  - Help create examples of collision processing through 3rd party level-set libraries

Project types

- *Special note for fluids-related projects*
  - Instructor can provide basic framework
  - Linux-only (for now)
  - Recommend coordinating into a fluids-task-force group, for reuse of advice/tools