Final Project

Specific dates for project deliverables will be posted on the course webpage.

Overview

In this project, you can work individually or as part of a team (up to 3 members) to tackle specific problems in one of the following categories:

Category 1: Pedagogical Aids

The goal of this project category is to develop an interactive application (preferably web-based) to illustrate concepts learned in class. It is NOT sufficient to just implement the algorithm, your implementation must also highlight (visually) the important points of what the algorithm is doing.

Note that there is a great deal of code that is publicly available on the web. You are welcome to use this (e.g. classes that draw graphs, or support the creation of balanced binary trees, etc), BUT you must reference where it came from, and clearly indicate WHAT functionality you have added to the original code.

Example project ideas:

- An application showing the running of the "merge-sort" type of convex hull construction algorithm. The application should show every edge that is tested as the algorithm runs, including, for example, which edges are tested when finding the common lower tangent of two sub hulls.
- An application showing Kirkpatrick's triangular decomposition data structure for point location. The application should have a side by side picture. On the right, the "decomposition tree", whose root node represents the whole enclosing triangle, and on the left, the original set of points, and the set of triangles which the query point is being tested against.
- http://alexbeutel.com/webgl/voronoi.html
- http://thomasdiewald.com/javascript/webgl/PixelVoronoi_KdTree_v05/
- http://www.sw-engineering-candies.com/blog-1/n-body-simulation-of-a-growing-water-melon-in-a-box

Category 2: Work on Open Problems in the Field

This can take the form of a paper (a) describing the current best algorithms for these problems, the best bounds on the running time of these algorithms, and (b) a description of your approaches to these problems.

YOU DO NOT have to improve on the best known bounds for these algorithms, but you do have to carefully document what approaches you tried, why they worked/failed. The expected result of this would be a roughly ten page paper (with diagrams and references), or an equivalent length webpage/blog talking about things you tried and the results obtained.

Example project ideas:

- The Freeze-Tag problem
- Edge Coloring Geometric Graphs
- Thrackles
- Other unsolved problems in the Open Problems Project

Category 3: Applications of Computational Geometry in other fields

One of the claims you would hear repeatedly in class is that Computational Geometry is useful in many different fields. Here is your chance to help prove that statement:

For this project category you can tackle problems that are in line with your own interests. For example:

- The application of computational geometry in an artistic forum (e.g. Interactive Voronoi Art)
- The application of computational geometry in a games
- Traffic flow optimization
- Matching 3D objects scanned with a Kinect to those in existing databases (3D object similarity is an active area of research)

Your implementation should be accompanied with a document/webpage describing the problem that you are trying to solve, a summary of existing approaches, and a description of your technique. You should also provide details about datasets that you are using and any assumptions you are making.

Final document/webpage guidelines

Your final document or webpage, should include:

- Names of people in your team.
- A specific description of the problem that you are trying to solve, in two sentences or less. This may be an answer to a question like: What does your application do? What is your visualization trying to show? What open problem were you trying to address?
- If you are doing a pedagogical application:
  - Background information about the problem and approach. Does not need to be long (one or two paragraphs is ok), but state any assumptions that are necessary for your approach to work.
  - Describe the pseudo-code of the algorithm that you are creating and highlight what parts of that pseudocode you are animating.
  - Discuss any interesting choices that you made in the implementation of the algorithm.
- If you are working on an open research problem:
  - Background information about the problem. What is the current best known algorithm? What are the proven theoretical bounds on the problem?
  - What approaches have you tried? Can you visualize the results? Show sketches of where proof attempts have failed, or example point sets where your (mostly good) algorithm has a bad case?
- If you have an application domain problem:
  - Background information about the problem.
  - What are your specific inputs and outputs (show a visualization, if possible).

    - What are the needs for the problem domain -- is it important that you find the optimal solution? why or why not?
    - Are there related projects by others that tackle the same problem. How?
    - Describe your algorithm and show example results.
    - Give a complexity analysis of your algorithm, and also show measured running time for different sizes of inputs
    - If your project is more artistic or a game:
        - What is the goal?
        - Describe input/outputs and give pseudocode of your approach.

Logistics

- By the end of week 2 you should email the instructor indicating if you are working individually or as part of a team. The email should include the names of all team member. Be sure to copy all member of the team on the email. Your team should begin brainstorming the project category and specific project that they would like to focus on.
- At the end of week 5, each team will submit a one page project summary. The summary should include:
    - The names of the team members
    - A project title
    - A description of the project
    - A list of references you will be using (does not need to be exhaustive as this list will probably grow as you get deeper into the project)
    - A detailed timelime (i.e. what must be accomplished each week for the project to be successful
    - A plan for dividing the duties amongst the team members
- During week 8, each team will give a 10-minute presentation in class describing their project, the work that have accomplished thus far, and plans for completing the project.
- During week 15, each team will give a 10-minute final presentation/demonstration in class describing the outcome of their project.
- Final project document/webpage will be due on Friday of week 15.

Grading

- One page project summary (5 points)
- Week 8 presentation (15 points)
- Week 15 presentation (30 points)
- Final project document/webpage (50 points)