Final Project Information

MATH 261 Computational Geometry

The final project is your opportunity to learn about some aspect of computational geometry that we didn't study during the course or to go deeper into a topic that we did study—and to learn about many other topics from your classmates!

For your project, you will need to explore a topic, write a paper, and give a short presentation to the class. You will need to do something to make the topic your own, rather than copying information verbatim from whatever source(s) you use. This could involve producing some examples, proving theorems, implementing an algorithm, applying algorithms to data, or making a research paper understandable to your classmates.

This will be a group project, completed in groups of two or three students.

Timeline

- **January 16-20:** Consider possible topics and discuss with potential group members.
- **January 21:** Complete the Project Planning Survey. This will ask for possible topics and who you do (or don't) want to work with.
- **January 22:** Project topics and groups finalized.
- **January 23-29**: Afternoon class time will be devoted to final projects. Discuss progress and questions with the professor.
- **January 30, 8:00am:** Projects due. Self and Peer Evaluation due. Group presentations about what you studied and discovered.

Paper

Your paper should explain what you have learned or discovered as part of your final project. Assume that the audience for your paper consists of other students in this course. Write so that your classmates could read and understand your paper.

Your paper must be typed (with proper mathematical typesetting) and submitted on Moodle by start of the scheduled final exam period (Thursday, January 30, 8:00am).

The title of your paper should be brief and should describe the content of your paper. An abstract not exceeding 200 words that summarizes the principal concepts and conclusions of your work must appear following your title, but before the content of your paper.

Your paper should have a bibliography with at least three references. All illustrations must be of professional quality. If you borrow graphics from other sources, you must give credit to those sources.

A typical length for the paper is 2000 words for a two-author paper, or 3000 words for a three-author paper. However, the length of the paper may be less if your project involves writing a substantial amount of code. Furthermore, the quality of the paper is more important then the length of the paper.

For some guidelines about writing a math paper, consult *Guidelines for Good Mathematical Writing* by Francis Su[1] and *How to Write Mathematical Papers* by Bruce Berndt.[2]

[1] Francis Edward Su, *Guidelines for Good Mathematical Writing*, https://scholarship.claremont.edu/cgi/viewcontent.cgi?article=2154&context=hmc_fac_pub

[2] Bruce Berndt, *How to Write Mathematical Papers*, https://alozano.clas.uconn.edu/wp-content/uploads/sites/490/2020/08/berndt.pdf.

Presentation

During the scheduled final exam period, you will give a short presentation about your project. While the presentation does not have to be as detailed or as technical as your paper, it should give your audience a clear idea of what you have done and what you have learned. The length of the talk should be 3–4 minutes per person.

Your presentation should involve a few slides, prepared using the technology of your choice. Make sure your slides are legible, with figures clearly labeled. Slides with pictures and concise text tend to be more informative than those filled with equations or code. Include references in your slides as appropriate.

Grading

To earn a high score on this project, your work should exhibit the following characteristics.

- Your paper explains your topic with clarity, precision, and attention to detail.
- Mathematical terminology (especially terminology from class) is used correctly and consistently.
- Mathematical notation is properly typeset. For example, write x^2 and *not* x^2.
- Examples and diagrams are used as appropriate to illustrate important concepts, especially technical concepts.
- Your writing quality is of high quality, including proper grammar, proper sentence structure, and organization using paragraphs and section headings.
- It's OK to focus on applications, but the paper can't just be about a subject such as physics or biology. Rather, show how computational geometry connects to such an application; that is, show how computational geometry can lead to scientific insights.
- Your paper should distill insights from multiple sources. It should not be primarily a re-write of a single research article.
- Illustrations should be professional quality, not hand-drawn (unless you can draw with professional quality). It's OK to borrow diagrams from other figures if you cite your sources.
- If your project involves writing code, then your code must be of high quality: use proper coding constructs and data structures as necessary, well organized, comments to explain what your code does. Provide sample input/output and diagrams as appropriate to demonstrate that your code works as expected.
- Your presentation is clear, demonstrating that you have thought about how to communicate your work to the class.

Topic Ideas

Some possible topics for the final project appear below. This list is not intended to be exhaustive—feel free to come up with other ideas as well!

1. Research and implement a computational geometry algorithm. Discuss implementation choices that you made and difficulties that you overcame. Always acknowledge parts of code or ideas that you found elsewhere. After implementing the algorithm, demonstrate that it works.

2. Create a visual demonstration of an algorithm or proof. An interactive demonstration that allows the user to supply input or try out multiple examples would be ideal.

3. Study data structures used for storing geometric objects such as graphs, surfaces, or polyhedra. Implement a demonstration. One possible starting place is http://www.sccg.sk/~samuelcik/dgs/geom_structures.pdf.

4. Investigate and implement image morphing. One possible starting place is http://andrew.gibiansky.com/blog/image-processing/image-morphing/.

5. Investigate and implement triangulated image abstraction. One possible starting place is https://puckey.studio/projects/delaunay-raster.

6. Go into depth with some sort of art gallery problem. Joseph O'Rourke has a book on this topic: http://cs.smith.edu/~jorourke/books/ArtGalleryTheorems/art.html

7. Investigate motion planning or collision detection algorithms. Implement an algorithm or apply existing algorithms to some real-world scenario.

8. Investigate algorithms and applications of mesh construction. Implement an algorithm or apply existing algorithms to data. One possible starting place is http://persson.berkeley.edu/distmesh/persson04mesh.pdf.

9. Investigate applications of Voronoi diagrams and create a demonstration. One possible starting place is http://www.voronoi.com/wiki/index.php?title=Voronoi_Applications.

10. Research some aspect of the Fold-and-Cut Problem. Erik Demaine's website is a good place to start: http://erikdemaine.org/foldcut/

11. Heesch's problem: How many times can a polygon (or other shape) be surrounded by copies of itself? What is known about Heesch numbers?

12. Explore some aspect of tiling. For example, what is Conway's criterion for planar tilings? What are aperiodic tilings and what is known about them? What symmetries are possible with tilings? What is known about tilings in higher dimensions?

13. Investigate an open problem! The text gives many open problems. You can produce a quality project even without solving the problem.