Project - Delaunay Triangulation

Now that you are familiar with visualization tools and geometry algorithms, it is time to do some serious stuff.

You are going to build a *Delaunay triangulation*, a triangle mesh that presents nice properties and is used extensively as a support in both computational mechanics and computer graphics.

You will also build a *Voronoï diagram* on a set of points. As you will see, it is quite simple to obtain the Voronoï cells starting from the Delaunay triangulation, as the two structures are dual to each other.

Input. You are given only a set of 2D vertices. Variable `nodeData` is defined in a `.json` file.

General guidelines. We do not give you a starter code on purpose to let you build your code from scratch and encourage you to be creative. You are free to define other input files, as many canvases as you like, interactive code, fancy buttons, . . . Try to impress us and make your website fun and intuitive.

To help you get started, we make a solution code for homework 2 available to you on the course website. As always, plagiarism between students is strongly forbidden and will be checked, but you are allowed to reuse bits of code that are published by the teaching staff.

Deliverables. On top of your codes, we want you to deliver a *project report*. This report can either be included on your website, or separately in a Markdown file. It is up to you to define the content of your report: make it interesting! Here are some suggestions:

- Make a pedagogical presentation of the algorithms for your colleagues that have not followed the course,
- Explain the visualization techniques that you have used that we have not seen in the homeworks,
- Show us how you accelerated your codes, compare to state-of-the-art algorithms,
- Show a nice practical application of the structures you have built.

We will test your code using custom input data, so please **explain in your report how we can run your code on our inputs** (using a special input file name, a button to upload the file, . . .).

Grading. The grading structure is as follows (total of 20 points):

- The **basic tasks** (Delaunay triangulation + Voronoï diagram), if done perfectly with a robust (but not necessarily fast) code and a good report, will earn you up to **12 points**.
- **Speeding up** your algorithms, achieving optimal complexity and proving this complexity with numerical experiments will award you up to **6 points**.

- A **pretty website**, with nice aesthetics, colors, animations (fairly easy to do in canvas: look it up!) will earn you up to **6 points**.
- If your code is **robust** and able to handle pathological cases, you can earn up to **3 points**. Think about the following cases: colinear points, concyclic points, points for which you don't know the bounds in advance. Show us how you handle these edge cases.

Oral defense. An oral defense of your project will be organized during the exam session. The practical details of this session will be communicated later on.