

Triangulation de Delaunay et Diagramme de Voronoï

— TP —

Les points du plan sont repérés par leurs coordonnées supposées entières et appartenant à $[0, x_{max}] \times [0, y_{max}]$. On pourra, par exemple, fixer $x_{max} = 1024$ et $y_{max} = 1024$.

Il est conseillé d'utiliser le squelette de code fourni en java.

- 1° Écrivez une fonction calculant le centre et le rayon du cercle circonscrit à un triangle
- 2° Écrivez une fonction vérifiant si une arête de la triangulation est légale
- 3° Écrivez la fonction de Flip¹.
- 4° Écrivez l'algorithme permettant de transformer une triangulation à une triangulation de Delaunay².
- 5° Dessinez le résultat ; Au lieu de les dessiner vous même, vous pouvez aussi choisir de sortir le résultat dans un fichier au format `.PLY`³ afin de les visualiser dans un logiciel approprié, par exemple Meshlab (<http://www.meshlab.net/>).
- 6° Calculez, à partir d'une triangulation de Delaunay, les points du diagramme de Voronoï
- 7° Dessinez le diagramme de Voronoï correspondant⁴.
- 8° Implémenter une fonction de génération aléatoire de points, et refaites une triangulation de Delaunay pour un grand nombre de points.

1. Pensez à la mise à jour les références des demi-arêtes vers les faces et les sommets, même chose pour les références vers les demi-arêtes des faces et des sommets.

2. La fonction `makeMesh` fournie permet de générer une triangulation via l'algorithme incrémental.

3. format `.PLY` : <http://paulbourke.net/dataformats/ply/> ou [https://en.wikipedia.org/wiki/PLY_\(file_format\)](https://en.wikipedia.org/wiki/PLY_(file_format))

4. Attention aux triangles du bord de la triangulation.