

Tabla de Contenido

1. Introducción	1
1.1. Justificación	3
1.2. Objetivos de la Memoria	4
1.3. Metodología	5
1.4. Contenido de la Memoria	5
2. Marco teórico	7
2.1. Conceptos geométricos y mallas	7
2.1.1. Mallas de polígonos	7
2.1.2. Triangulación de Delaunay	7
2.1.3. Teselación de Voronoi	9
2.1.4. Dualidad Delaunay-Voronoi	9
2.2. Programación en GPU	10
2.2.1. CUDA	10
2.2.2. OpenGL y lenguaje de Shaders	13
2.2.3. Interoperabilidad entre CUDA y OpenGL	16
3. Análisis y diseño de las librerías CLEAP y TIUQUE	18
3.1. Introducción	18
3.2. Estructuras de datos	19
3.2.1. Vértices	19
3.2.2. Aristas	19
3.2.3. Triángulos	20
3.2.4. Mallas	20
3.3. Algoritmos en CLEAP	20
3.3.1. Creación de estructuras y fill-in de datos	20
3.3.2. Transformación de Delaunay	25
3.4. Funcionalidades de TIUQUE	28
4. Diseño e implementación del cálculo del Diagrama de Voronoi	31
4.1. Integración	31
4.2. Estructuras	31
4.2.1. Circuncentros	31
4.2.2. Aristas de Voronoi	32
4.2.3. Índices-Aristas	33
4.2.4. Relación aristas de Voronoi	33

4.2.5. Polígonos	33
4.3. Algoritmo	34
4.3.1. Identificación de Ciruncuncentros	34
4.3.2. Indexación de aristas	34
4.3.3. Proyección de puntos medios	35
4.3.4. Relación ciruncuncentro-arista	35
4.3.5. Half-edges y Reconocimiento de polígonos	36
4.4. Visualización de Voronoi	40
5. Pruebas y Resultados	42
5.1. Cambios de interfaz	42
5.2. Definición de pruebas	42
5.3. Uso de memoria	43
5.3.1. CLEAP: Delaunay vs Delaunay-Voronoi	44
5.3.2. Delaunay-Voronoi: CLEAP vs Qhull	44
5.4. Tiempos de ejecución	46
5.4.1. CLEAP: Delaunay vs Delaunay-Voronoi	46
5.4.2. Delaunay-Voronoi: CLEAP vs Qhull	46
5.5. Representaciones y correctitud	48
6. Conclusiones	56
Bibliografía	57
Anexos	58
Anexo A Capítulo 2	58
Anexo B Capítulo 4	60
Anexo C Capítulo 5	63