

Informe del Laboratorio de Computación Gráfica

Curso: Computación Gráfica

Universidad Nacional de San Agustín

E.P. de Ciencia de la Computación

Integrantes:

- Anagabriela Pilar Jimenez Lopez
- Sennayda Rimache Choquehuanca
- Merisabel Ruelas Quenaya
- Katherine Nikole Bejar Roman

Generación de Puntos Aleatorios

La generación de puntos aleatorios se maneja en la clase [DotCloudGenerator](#), definida en DotCloud.h.

- **GetSphericalDots:** Este método es responsable de generar una nube de puntos distribuidos de manera que parezcan formar una esfera. Esto implica generar puntos aleatorios dentro de un espacio tridimensional y luego proyectarlos sobre la superficie de una esfera, o generarlos directamente en coordenadas esféricas y luego convertirlas a coordenadas cartesianas.

Triangulación de Delaunay

La Triangulación de Delaunay se implementa en la clase [DelaunayTriangulation](#), definida en Triangulation.h. Este proceso se inicia con la llamada al método [GetTriangulationResult](#), que toma un vector de punteros a objetos Vector3D (los puntos a triangulizar) y devuelve un vector de tuplas, cada una representando un triángulo mediante los índices de sus vértices.

- **Construcción del Casco Inicial:** El método [BuildInitialHull](#) construye un casco inicial utilizando un número fijo de puntos auxiliares. Este casco sirve como punto de partida para la triangulación.
- **Inserción de Puntos:** [InsertDot](#) inserta puntos uno por uno en la estructura de datos de la triangulación, ajustando la triangulación existente según sea necesario para mantener las propiedades de Delaunay.

- Eliminación de Triángulos Extra: **RemoveExtraTriangles** elimina triángulos que no forman parte de la triangulación final, como aquellos que incluyen puntos auxiliares.
- División de Triángulos y Optimización Local: **SplitTriangle** divide triángulos para insertar nuevos puntos, y **DoLocalOptimization** realiza optimizaciones locales para asegurar que la triangulación cumpla con las propiedades de Delaunay, posiblemente intercambiando diagonales de cuadriláteros formados por pares de triángulos adyacentes.

Visualización

La visualización de la nube de puntos y su triangulación se maneja en la clase `Visualization`, definida en `Visualization.h`. El método **ReconstructIn3D** toma los puntos y la malla de triangulación como entrada y reconstruye la estructura en 3D para su visualización. Esto implicaría renderizar los puntos y las aristas de los triángulos en un espacio 3D, utilizando una biblioteca gráfica.

Flujo General del Programa

1. **Generación o Lectura de Puntos:** El programa principal (`Main.cpp`) solicita al usuario que elija entre generar puntos aleatoriamente o leerlos desde un archivo. Utiliza `DotCloudGenerator` o `DotCloudReader` según la elección.
2. **Triangulación de Delaunay:** Con los puntos generados o leídos, el programa realiza la Triangulación de Delaunay utilizando la clase `DelaunayTriangulation`.
3. **Visualización:** Finalmente, los puntos y la malla de triangulación se pasan a la clase `Visualization` para su reconstrucción y visualización en 3D.