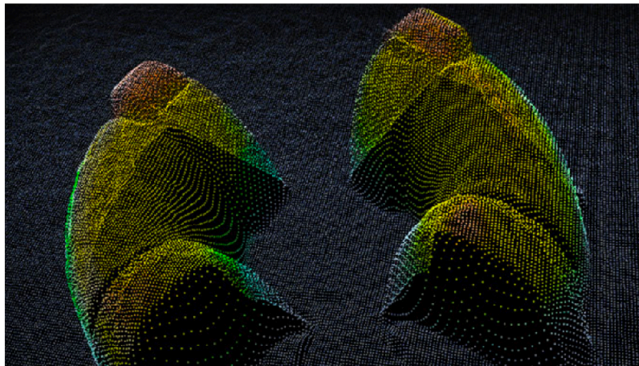


1. Objetivos

- Diseñar e implementar TADs

2.- Desarrollo del taller: Compilación con g++

Para la visualización en pantalla de objetos en tres dimensiones se utilizan muchas técnicas, una de ellas se conoce como point rendering o dibujado de puntos. Así, los objetos están representados con una estructura llamada nube de puntos, que básicamente es un conjunto de puntos tridimensionales ($p = [p_x, p_y, p_z]$) que, vistos desde una distancia específica, dan la impresión de representar una figura geométrica. Un ejemplo se puede ver en la siguiente imagen, donde se intenta representar el perfil de un terreno a partir de puntos de diferentes colores:



<https://metrology.news/time-of-flight-camera-generates-3d-point-cloud/>

De esta forma, cada objeto en la escena puede representarse con una nube de puntos, con sus coordenadas tridimensionales. El color de cada punto se representa con la combinación de 3 colores: rojo, verde y azul, los cuales se expresan en niveles de 0 a 255, cada uno. El objeto en la escena también tiene un nombre asociado, que permite identificar lo que la nube de puntos representa. Y para facilitar el dibujo de los puntos en pantalla, se requiere: la información del punto centroide de la nube de puntos (calculado como el punto promedio de todos los puntos que conforman el objeto) y un indicador de visualización que puede tomar tres valores: completa, cuando el objeto cabe completamente en la escena; parcial, cuando el objeto sólo puede visualizarse parcialmente en la escena y nula, cuando el objeto no puede visualizarse dentro de la escena.

La escena corresponde al espacio que alberga todos los objetos, la cual se visualizará en la pantalla. Esta escena tiene unos límites definidos en los tres ejes de coordenadas, que facilita identificar el indicador de visualización de cada objeto: visualización completa se logra si todos los puntos del objeto están dentro de los límites de la escena; se obtiene visualización parcial cuando sólo algunos puntos están dentro de los límites de la escena (y por tanto otros puntos están por fuera de los límites); y la visualización nula implica que todos los puntos del objeto están fuera de los límites de la escena. Los objetos en la escena deben ubicarse u organizarse de acuerdo a la cercanía al observador de la escena: del objeto más lejano al más cercano. De esta forma, al extraer los objetos se garantiza que aquellos más cercanos se dibujarán completamente (de últimos), encima de otros más lejanos (que ya se han dibujado primero).

Dentro de la representación computacional de los objetos en una escena tridimensional, se han identificado inicialmente dos operaciones importantes:

1. Actualizar el indicador de visualización para un objeto dado, revisando todos los puntos de su nube con respecto a los límites de la escena a la que pertenece.
2. Crear un nuevo objeto a partir de la unión de las nubes de puntos de dos objetos específicos en la escena. Si hay un punto con la misma coordenada en los dos objetos, se usa sólo el punto del objeto ubicado más adelante (más cercano) en la escena. Es necesario calcular el centroide del objeto, así como asignar su indicador de visualización. Finalmente, el nuevo objeto debe quedar ubicado adelante (más cerca) de todos los objetos en la escena, para que pueda ser el último en dibujarse.

2.- Diseño

Diseñe el sistema y el (los) TAD(s) solicitado(s). Utilice la plantilla de especificación de TADs vista en clase para el diseño. Recuerde que diseñar es un proceso previo a la implementación, por lo que no debería contener ninguna referencia a lenguajes de programación (es decir, si escribe encabezados o código fuente, el punto no será evaluado y tendrá una calificación de cero). Para simplicidad del diseño, no es necesario incluir los métodos obtener y fijar (get/set) del estado de cada TAD.

3. Diagrama de relación

Incluya el diagrama de relación entre TAD's.

4.- Operación 1: Actualizar indicador de visualización.

Dado el (los) TAD(s) ya diseñado(s), escriba la implementación en C++ del algoritmo que permite actualizar el indicador de visualización de un objeto dado. La implementación deberá tener en cuenta:

- la definición apropiada de los prototipos de los métodos/funciones (i.e. recibir/retornar los datos suficientes y necesarios para su correcta ejecución),
- el NO uso de salidas/entradas por pantalla/teclado (i.e. paso/retorno correcto de valores y/o objetos),
- el correcto uso del diseño definido en el punto anterior, y
- la escritura de todo el código que pueda llegar a necesitar que no esté incluido en la STL.

5.- Operación 2: Unión de dos objetos.

Dado el (los) TAD(s) ya diseñado(s), escriba la implementación en C++ del algoritmo que permite crear un nuevo objeto a partir de la unión de las nubes de puntos de dos objetos dados en la escena. Así como en el punto anterior, la implementación deberá tener en cuenta:

- la definición apropiada de los prototipos de los métodos/funciones (i.e. recibir/retornar los datos suficientes y necesarios para su correcta ejecución),
- el NO uso de salidas/entradas por pantalla/teclado (i.e. paso/retorno correcto de valores y/o objetos),
- el correcto uso del diseño definido en el punto anterior, y
- la escritura de todo el código que pueda llegar a necesitar que no esté incluido en la STL.