

1) Nota previa: Si no puede responder a), b) o ninguna de las dos, pruebe empezar por la c) o c')

a) ¿A qué se refiere el modelado por *Level-Sets* (conjuntos de nivel)?

b) ¿Se puede renderizar con una API como OpenGL, o hay que convertirlo a otro tipo de modelado? (explique)

c) Suponiendo que los puntos no están organizados como vóxeles (capas de píxeles) sino que son puntos dispersos y ya hay hecha una tetraedrización del volumen total (los puntos son vértices de tetraedros que rellenan el volumen); describa el método para encontrar los triángulos que definen una superficie de nivel; explique cuáles son las aristas relevantes y calcule por donde pasa la superficie, de ese nivel, en esa arista.

c') Sólo si no entendió c) en 3D, hágalo en 2D, pero es más o menos lo mismo: Suponiendo que los puntos no están organizados como píxeles en una cuadrícula sino que son puntos dispersos en el plano y ya hay hecha una triangulación de la superficie plana total; describa el método para encontrar los segmentos que definen una curva de nivel; explique cuáles son las aristas relevantes y calcule por donde pasa la curva, de ese nivel, en esa arista.

2) Describa las variables que utiliza el modelo de iluminación de Phong:

$$\mathbf{I} = K_a \mathbf{I}_{ag} + \sum_j [K_a \mathbf{I}_{aj} + K_d \mathbf{I}_{dj} \mathbf{n} \cdot \mathbf{l}_j + K_s \mathbf{I}_{sj} (\mathbf{n} \cdot \mathbf{h}_j)^q] + K_e$$

¿Que representa cada una? ¿Cómo se multiplican? ¿Qué resultado define la ecuación y a qué se asigna?

Nota: Puede utilizar $\mathbf{n} \cdot \mathbf{h}_j$ o $\mathbf{v} \cdot \mathbf{r}_j$ o definir un coseno (para ésta y/u otras componentes), como prefiera.

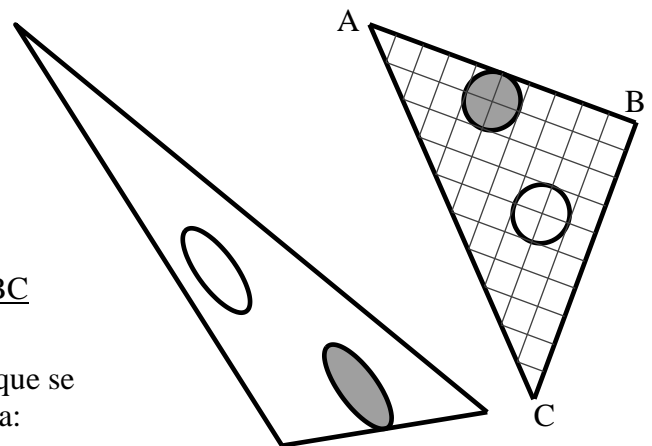
3) ¿Cómo se simula el *fog* o neblina en OpenGL? (No hacen falta ecuaciones sino la descripción y, si puede ser, una explicación de la linearización de la distancia)

¿Para qué puede servir, además de simular neblina?

4) El triángulo ABC de la figura derecha sufrió una transformación afín cuyo resultado se muestra a la izquierda.

En el nuevo triángulo transformado, calcule las coordenadas baricéntricas de los centros de ambas elipses, la rellena y la hueca, respecto de los puntos A, B y C transformados.

Muestre en un gráfico (a mano alzada), sobre el triángulo ABC transformado, que los valores calculados son los correctos.



5) Se muestra la Projection Matrix (o matriz de proyección) que se aplica a una escena, ya ubicada en el espacio del ojo. Defina:

a) ¿Qué tipo de transformación representa?

b) ¿Para qué sirve (toda Projection Matrix)?

c) ¿Cuál es el efecto que produce sobre la escena?

d) ¿Cuáles son las imágenes de los puntos $P_0 = \{2, 3, -3\}$ y $P_1 = \{4, 2, -4\}$?

Nota: Asuma que ambos puntos caben holgadamente en el viewport y están entre z_{near} y z_{far} .

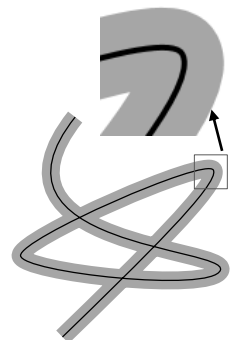
e) ¿Cuál es la imagen de (cada punto de) la línea $(1-\alpha)P_0 + \alpha P_1$

f) ¿Cuál es la imagen del punto $\{1, 1, 1\}$? (Hay algo raro y tiene que ver con el sistema de referencia visual)

$$\begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & -1 & -3 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

6) Rasterizar una curva ancha no es tarea sencilla. Se trata de lograr que la curva, de ancho $2a$, rellene a píxeles a cada lado “perpendicularmente” a la curva; pero hay que evitar que queden huecos de un lado y píxeles pintados dos veces del otro. (También se podría correr la curva a hacia cada lado y rasterizar las dos, tapar los bordes y “pintar” el interior; con las dificultades que eso tenga; pero, dado que aún no estudiamos curvas, créanme que eso se hace en tal y cual caso, pero no en general).

Analice las ventajas y desventajas de utilizar DDA o subdivisiones. Si quiere puede inventar o agregar algún mecanismo para rellenar o evitar la duplicación de píxeles, señalando también sus limitaciones y potenciales problemas. Aún con las desventajas que haya enumerado; explique cuál considera el método más adecuado de utilizar.



(NOTA: La calificación es global)