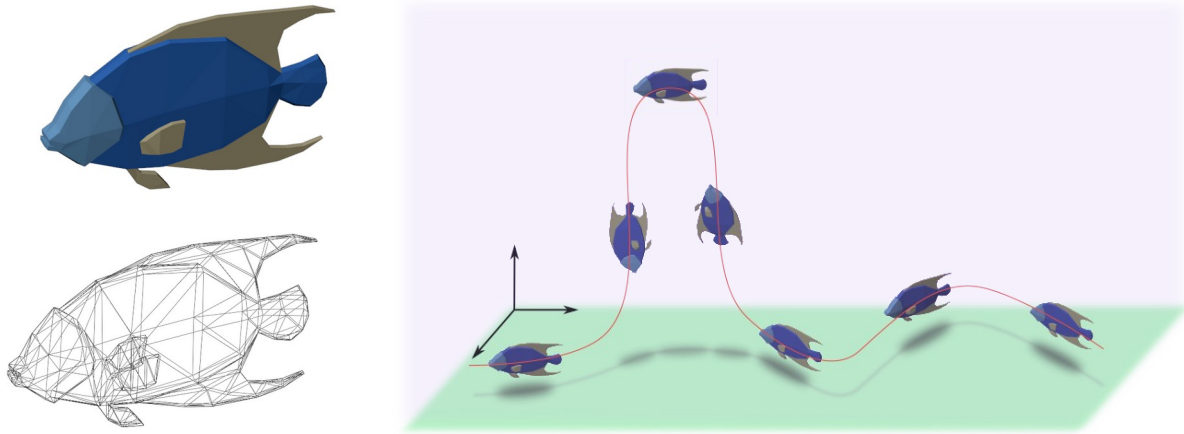
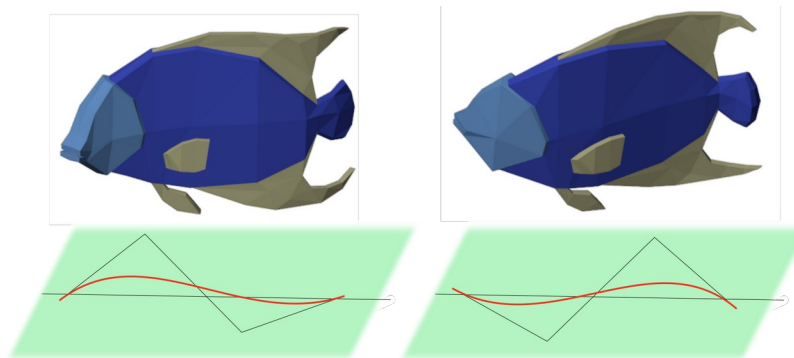


Computación Gráfica 2021 – Parcial – 10/11/2021 – Ejercicio 1

Se quiere animar el movimiento de un pez a lo largo de una trayectoria en el agua. La trayectoria está definida por una sucesión de curvas de Bezier de grado 3. El pez debe desplazarse siguiendo esa trayectoria



Además, el pez debe deformarse (doblar hacia un lado y hacia otro según una par de curvas) como se muestran en las figura de la izquierda para simular el movimiento que hace su cuerpo para avanzar.



Suponga que debe programar la función que dibuja el pez, donde los argumentos son:

- la malla del pez: vértices/triángulos
- el tiempo t en segundos para el cual se quiere dibujar el pez
- dos curvas planas (2D, 4 ptos de control cada una) representando las deformaciones máximas del cuerpo del pez. A medida que avanza el tiempo el pez debe oscilar suavemente entre estas dos.
- un vector de puntos de control representando la spline del trayecto: del 0 al 3 corresponden a la primera curva de bezier, del 3 al 6 a la 2da, del 6 al 9 a la 3ra, y así sucesivamente. Los puntos son 3D (la trayectoria es en el espacio R^3).
- si su método requiere algún argumento adicional incluya una breve justificación para el mismo.

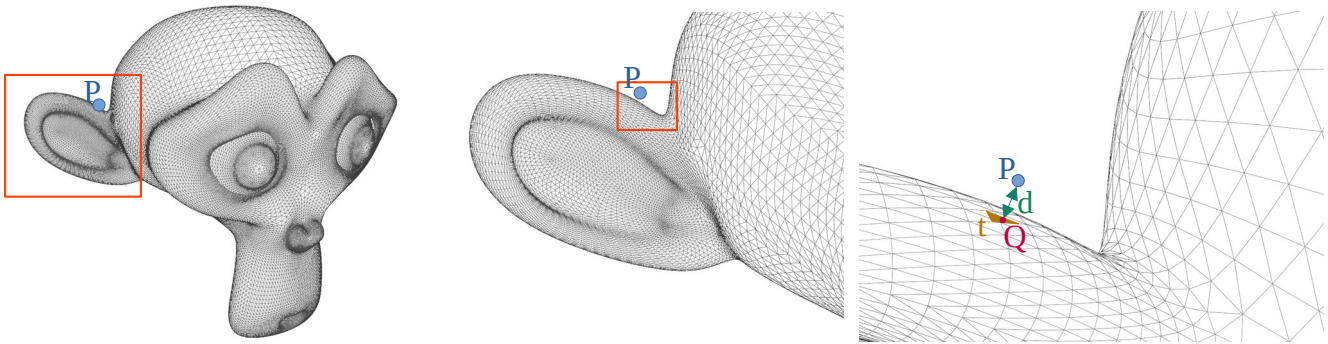
Describa qué debería hacer esta función (si quiere puede usar un pseudocódigo inventado, no hace falta respetar sintaxis OpenGL). Si realiza transformaciones que puedan representarse con una matriz, describa cómo armaría o calcularía la matriz. Si hay transformaciones que no pueden representarse con una matriz, indique por qué, y cómo la aplicaría. Indique cómo evaluaría la spline para cada t (qué tramo usaría, con qué parámetro, etc. teniendo en cuenta que el pez debería moverse a velocidad constante).

Computación Gráfica 2021 – Parcial – 10/11/2021 – Ejercicio 2

Se tiene una malla compuesta por muchos triángulos (millones), y se quiere programar una función para encontrar el punto de la superficie más cercano a otro punto dado P . La función debe determinar:

- Las coordenadas del punto (Q) de la malla más cercano a P (notar que el punto más cercano no es necesariamente un vértice, sino que puede estar en el interior de un triángulo).
- A qué distancia (d) está y a qué triángulo (t) pertenece.
- Si el punto P está en el interior o exterior de la malla (suponga que la malla representa una superficie cerrada)

¿Cómo lo resolvería? Incluya las ecuaciones necesarias, la descripción del algoritmo, señale si hay pasos que podrían realizarse una sola vez en caso de tener que hacer muchas consultas (invocar a la función para muchos P s diferentes), si será necesaria información adicional sobre la malla además de nodos y elementos (por ej: normales o vecindades como en el TP de superficies), si hay casos particulares donde la salida deba ser diferente o un error, etc.

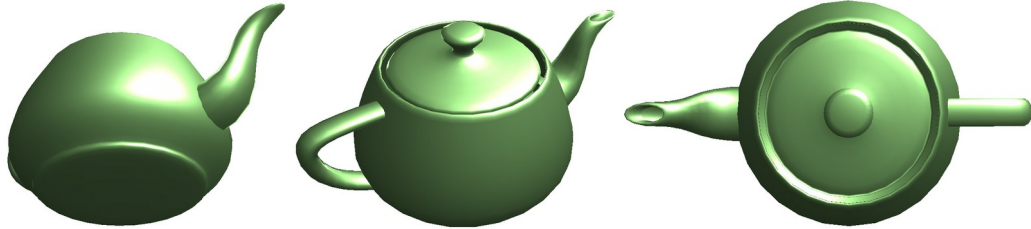
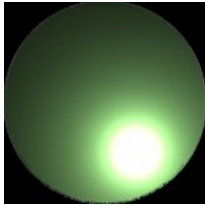


Computación Gráfica 2021 – Parcial – 10/11/2021 – Ejercicio 3

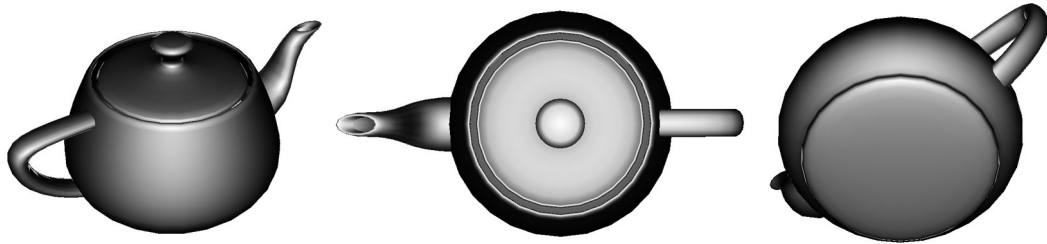
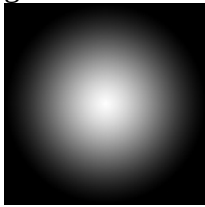
Para hacer un mapeo ambiental con OpenGL podemos usar el método de generación de coordenadas de textura GL_SPHERE_MAP. Este método genera coordenadas dentro de un círculo de radio 0.5, centrado en $s=t=0.5$. Cada punto del círculo en la textura se corresponde con una dirección (no hace falta detenerse en la fórmula exacta). El uso habitual es representar en la textura un ambiente lejano, y simular así el reflejo de ese ambiente sobre un objeto. Las coordenadas generadas s y t eligen entonces un punto dentro del círculo (la imagen) según la dirección de un rayo reflejado (r). Es decir, que se pueden ver como funciones $s(r)$ y $t(r)$, con r unitario. Este rayo es el reflejo de uno que sale desde el ojo hacia el vértice, según la normal del vértice; por lo que su dirección depende de las coordenadas y de la normal del vértice.

Usando ciertas texturas, el resultado puede parecerse mucho al del cálculo de iluminación por el modelo de phong. Las siguientes imágenes se generaron con el TP de texturas, usando este mapeo y las texturas “hilit.png” y “glare.png” (sin iluminación, modo de aplicación Replace).

hilit:



glare:



1) Explique por qué se da esto. ¿Por qué el resultado de la aplicación de textura puede aproximar al modelo de iluminación en ciertos casos/configuraciones? ¿En cuales casos/configuraciones?

Ayudas:

- Recuerde que el rayo reflejado depende de la coordenadas del vértice y de la normal. Considerando que el cálculo del rayo reflejado se da en el espacio del ojo. ¿Qué coordenadas tendrán los vértices?
- Imagine que la textura representa la solución para un función $f(r)$. Intente pensar cada componente del modelo de phong como una función. ¿De qué depende cada una? (intente reducir las variables independientes al mínimo). ¿Pueden estas funciones aproximarse con una textura ambiental?

2) ¿A cual de los modelos de “sombreado” se parece más este resultado? ¿Por qué?