



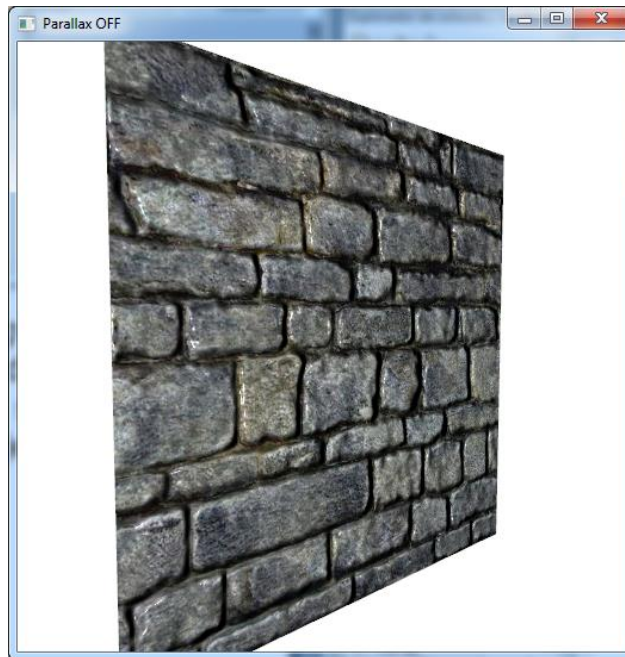
Práctica 8

Objetivo

En esta práctica se implementarán las técnicas de *bump mapping* y *parallax mapping*.

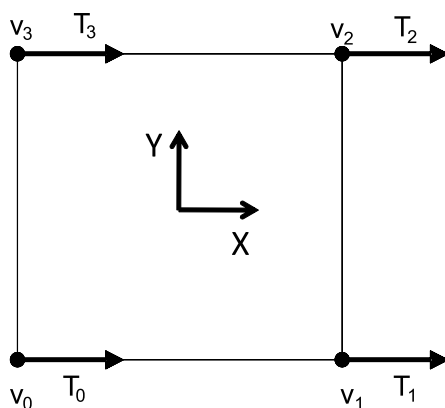
Primera parte

El objetivo de la primera parte de la práctica es implementar la técnica de *bump mapping* estudiada en clase.



El programa que se te proporciona (p8) se encarga de cargar las tres texturas mostradas en teoría (color base, mapa de normales y mapa de brillos), y establecer el valor de las tres variables tipo `sampler2D` correspondientes. Dichas variables están declaradas en el shader de fragmento, para poder acceder a las texturas (almacenadas en los ficheros `rocas-color.png`, `rocas-normales.png` y `rocas-gloss.png`, respectivamente). La aplicación carga los shaders desde los ficheros **p8.vert** y **p8.frag**.

Tendrás que probar tu implementación de bump mapping sobre el objeto de tipo **Rect** que se dibuja por defecto. Para ello deberás crear un nuevo atributo en el que pasar la tangente a cada vértice, tal y como muestra la siguiente figura. El polígono está en el plano XY, y las normales de todos los vértices apuntan hacia las Z positivas.

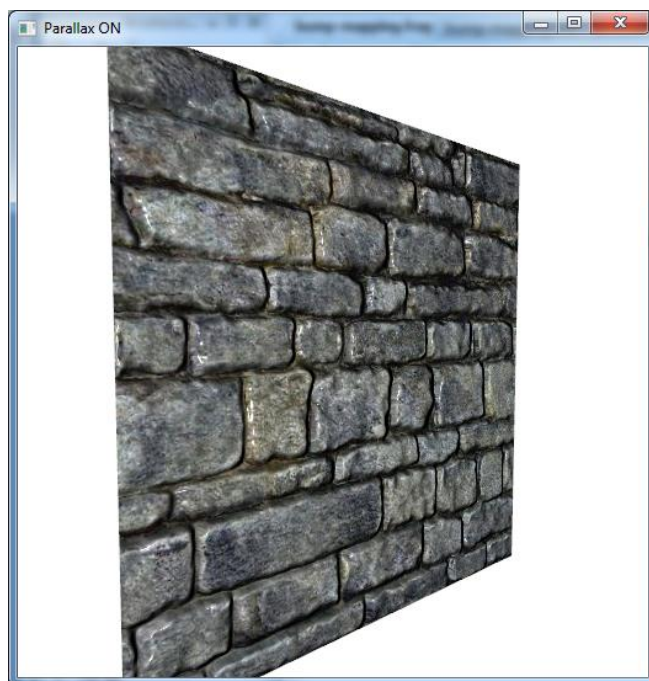


Para añadir las tangentes a la malla del modelo tienes que usar el método **Mesh::addTangents** en la función **setup** (para este objeto, como todas las tangentes son iguales, también podrías usar el método **Mesh::setTangent**, que crea un atributo constante para todos los vértices).

La aplicación también define una fuente de luz que se mueve en círculos justo delante de la pared. Basándote en el shader que desarrollaste para la práctica 5 (el de cálculo de iluminación por pixel), deberás tener en cuenta la posición de dicha luz (asume que es puntual).

Segunda parte.

La segunda tarea de la práctica consiste en implementar la técnica del *parallax mapping*.



Amplía la primera parte (sobre el mismo proyecto) añadiendo el efecto de parallax mapping tal y como se describe en las transparencias. Para poder comparar entre bump mapping normal y bump mapping + parallax mapping, puedes usar el botón correspondiente (si no ves la interfaz de usuario, pulsa Ctrl+G para mostrarla). Dicho botón está conectado con la variable uniform de tipo bool llamada **useParallax**, que



está definida en el shader de fragmento. Ten en cuenta dicha variable a la hora de implementar tu shader. La imagen `rocas-mapa-alturas.png` contiene el mapa de alturas de la pared. Tendrás que cargar la textura y asociarla correctamente a un nuevo sampler. Añade dos `FloatSliders` a tu aplicación para poder ajustar los valores de la escala y el *bias* del parallax mapping. Así, podrás probar qué valores generan mejores resultados. Puedes empezar por los valores de 0.02 para la escala y 0.01 para el *bias*.

Ampliación.

Añade algún objeto más complejo a la escena (p.e., un cilindro). Recuerda que deberás definir la tangente por vértice en las mallas a las que quieras aplicar el efecto. Otra ampliación puede ser definir el patrón de las normales procedualmente en vez de mediante una textura. En ese caso deberías calcular la normal perturbada mediante una función en el shader de fragmento.