

Tarea11

Normal Mapping

Previamente utilizaste un Phong shader para un objeto. Ahora crea otro objeto como al que le aplicaste el Phong Shader para hacer una comparación más sencilla.

Practicamente, en el Phong Shader utilizaste la normal de los atributos dentro del archivo .nfg para la iluminación y se consiguió un objeto con una superficie suave (la normal tomada de los atributos es interpolada para obtener la normal de cada fragmento, pero la diferencia entre la normal de fragmento a fragmento es muy pequeña para ser percibida, por lo tanto vemos una superficie suave).

En esta práctica usaras la normal obtenida de NormalMap.tga que dará la normal para cada fragmento, por lo que se verán asperezas.

Instrucciones:

1. Leer:

https://docs.gameloft.org/3d-training/#Normal_Mapping

2. Cuando hagas el Sample de NormalMap.tga (extrae solo los valores xyz, una normal es un vec3), los valores obtenidos serán entre [0,1], pero los valores para la normal deben de ser entre [-1,1]. Por lo tanto necesitamos hacer un ajuste:

normalT * 2.0-1.0

Donde **normalT** es la normal obtenida de NormalMap.tga. Esta expresada en espacio de coordenadas de la tangente (llamado también espacio textura). El espacio de la tangente es la tangente hacia tu objeto en un fragmento en particular.

3. Porque el cálculo de la iluminación es realizado normalmente en el world space, necesitas transformar **normalT** desde el tangent space al world space

Para esto necesitas calcular la matriz de transformación llamada **matriz TBN**(tangente, binormal, normal – el orden importa).

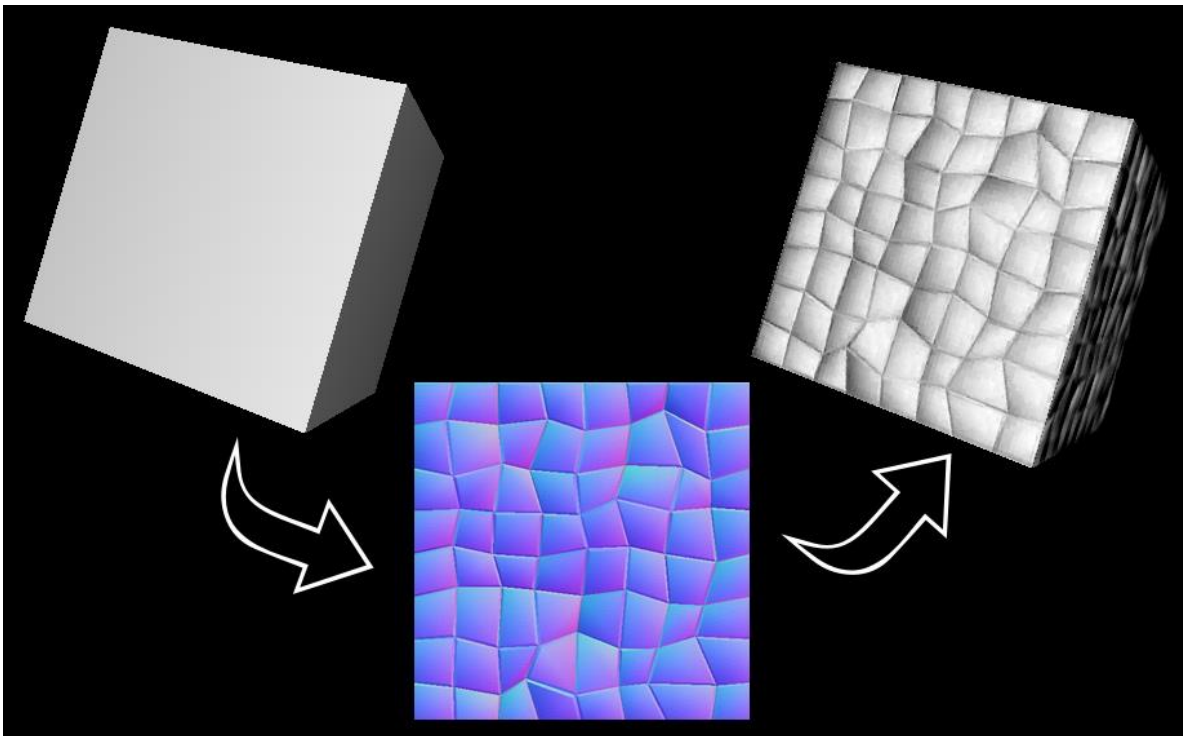
Tangente, binormal (también llamada bitangente) y la normal son ejes que caracterizan el tangent space. Ellos tienen cierta orientación (por ejemplo rotación) al contrario de los ejes de la world space. Si lo expresamos en las coordenadas de world space, podremos usarlos para transformar cualquier vector de tangent space a world space.

La matriz TBN en efecto es solo una matriz de rotación.

4. En el vertex shader usaras los atributos `a_normL`, `a_binormalL`, `a_tangentL`, los transformaras a world space y los mandaras al fragment shader como varyings

```
mat3 TBN = mat3( normalize(v_tangentW), normalize(v_binormalW),  
normalize(v_normW) );
```

5. Transforma tu normalT usando TBN, así obtendrás una normal en coordenadas de world space.
No olvides normalizar.
6. Usar la normal calculada para realizar la iluminación (como lo hiciste en el phong shader, pero con esta nueva normal que obtuviste).



Resultado:

