

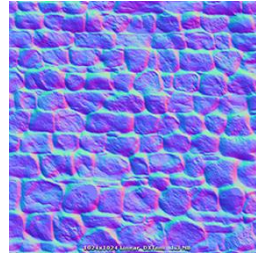
Parte obligatoria

Adrián David Morillas Marco y Guillermo Meléndez Morales
Diseño y Desarrollo de Videojuegos + Ingeniería de Computadoras

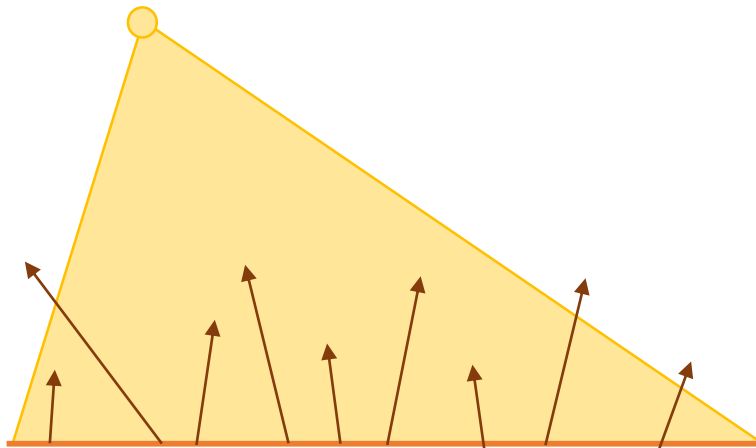
Bump Mapping

“Técnica que, gracias al uso de la luz, permite simular relieve en una textura, dándole sensación de profundidad o rugosidad a un objeto que en origen no era más que un color plano. “

Para emplear el bump mapping es necesario tener un mapa de normales como el que se nos proporciona en la práctica:



El mapa de normales hace que la reflexión de la luz sea diferente en cada pixel, es decir:



El inconveniente que tiene el mapa de normales proporcionales, es que es proporcionado en coordenadas de las tangentes, por lo que se deberá crear una matriz de cambio de base, que nos traslade ese mapa de normales a coordenadas de la cámara. Nuestra base en el espacio tangencial está formado por la normal, por la tangente y por la bitangente de la cual carecemos el valor inicialmente. Al formar una base eso tres vectores, forman todos entre sí 90 grados, por lo que mediante el producto vectorial $N \times T$ obtendremos la bitangente. Conocidos nuestros tres vectores de la base realizaremos la matriz de cambio de base:

$$TBN = \begin{pmatrix} Tangente \\ Binormal \\ Normal \end{pmatrix} = \begin{pmatrix} T_x & T_y & T_z \\ B_x & B_y & B_z \\ N_x & N_y & N_z \end{pmatrix}$$

Esta matriz la realizaremos en el shader de vértices, pues será allí donde recibiremos el vector tangente y el vector normal:

```
in vec3 inNormal;
```

```
in vec3 inTangent;
```

```
out mat3 TBN;    // transferimos nuestra matriz al shader de fragmentos donde realizaremos el cambio  
                //de base
```

```
vec3 Tangente = normalize (vec3 (inTangent));
```

```
vec3 Normal = normalize (vec3 (inNormal));
```

```
vec3 Binormal = normalize (vec3 (cross (Normal, Tangente)));
```

```
TBN [0].x = Tangente.x;
```

```
TBN [0].y = Binormal.x;
```

```
TBN [0].z = Normal.x;
```

```
TBN [1].x = Tangente.y;
```

```
TBN [1].y = Binormal.y;
```

```
TBN [1].z = Normal.z;
```

```
TBN [2].x = Tangente.z;
```

```
TBN [2].y = Binormal.z;
```

```
TBN [2].z = Normal.z;
```

A continuación nos dirigimos al shader de fragmentos y empezamos a realizar los cálculos necesarios para realizar el cambio de base. Lo primero que haremos será colocar como entrada la matriz TBN y el mapa de normales:

```
in mat3 TBN;
```

```
uniform sampler2D normalTex;
```

A continuación en el main (), extraemos las normales del mapa de normales:

```
vec3 coordlocal = 2.0* Normal.rgb - vec3(1.0);
```

Para realizar el cambio de base, pasaremos primero a coordenadas del objeto (TBN) y posteriormente pasaremos a coordenadas de la cámara (modelView):

```
N = normalize TBN*coordlocal*mat3(modelView);    //hacemos un cambio de base
```

Obteniendo finalmente el bump mapping:

