

# NGN\_TEXTURE.H

```
NGN_Texture(  
    NGN_TextureData* texture,  
        // Objeto de la clase TextureData  
    int32_t position_x = DEFAULT_VALUE,  
        // Posición X inicial (oculto por defecto)  
    int32_t position_y = DEFAULT_VALUE,  
        // Posición Y inicial (oculto por defecto)  
    uint32_t textura_width = DEFAULT_VALUE,  
        // Ancho de la Textura (por defecto, el de la textura)  
    uint32_t textura_height = DEFAULT_VALUE,  
        // Altura de la Textura (por defecto, la de la textura)  
);
```

Crea una nueva textura, usando los datos previamente cargados.

```
NGN_Texture* clouds = new NGN_Texture(clouds_gfx, 0, 0);
```

```
NGN_Texture(  
    uint32_t texture_width,  
        // Ancho de la textura  
    uint32_t texture_height,  
        // Altura de la textura  
    int32_t position_x = NGN_DEFAULT_VALUE,  
        // Posicion X inicial (oculto por defecto)  
    int32_t position_y = NGN_DEFAULT_VALUE  
        // Posicion Y inicial (oculto por defecto)  
);
```

Crea una nueva textura en blanco con el tamaño especificado.

```
NGN_Texture* clouds = new NGN_Texture(1280, 720, 0, 0);
```

```
void Position(float position_x, float position_y);  
void Position(Vector2 pos);
```

Posiciona la textura en la coordenada dada.

```
clouds->Position(1200, 900);
```

```
void Translate(float speed_x, float speed_y);  
void Translate(Vector2 spd);
```

Mueve la textura en la dirección y velocidades dadas.

```
clouds->Translate(5.0f, 0.0f);
```

**void Size(float w, float h);**

Cambia el tamaño de la textura.

```
clouds->Size(64, 48);
```

**void Scale(float w, float h);**

**void Scale(float scale);**

Escala la textura, según el factor dado. Según la sobrecarga usada, escalara los ejes en conjunto o por separado. La escala por defecto es 1.0f.

```
clouds->Scale(1.5f);  
clouds->Scale(2.0f, 0.75f);
```

**void Rotate(double degrees);**

Rota la textura cada frame el número de unidades dado, en grados.

```
clouds->Rotate(1.2f);
```

**void SetCenter(float x, float y);**

Especifica, en coordenadas relativas y desde el centro real de la textura, donde se ubicará el centro de rotación de la textura.

```
clouds->SetCenter(-10, -5);
```

**void ClearContent ();**

Borra el contenido de una textura, si esta se ha creado en blanco.

```
clouds->ClearContent();
```

***Vector2 position***

***Vector2 screen***

Posición de la textura (global o en pantalla).

***float width***

***float height***

Tamaño de la textura.

***bool visible***

Indica si la textura es o no visible.

### ***int32\_t alpha***

Nivel de transparencia de la textura, entre 0 y 255.

### ***SDL\_BlendMode blend\_mode***

Modo de mezcla de color de la textura. Los modos disponibles son: *NGN\_BLENDMODE\_NONE*, *NGN\_BLENDMODE\_ALPHA*, *NGN\_BLENDMODE\_ADDITIVE* y *NGN\_BLENDMODE\_MODULATE*. El valor por defecto de esta propiedad es *NGN\_BLENDMODE\_ALPHA*.

### ***double rotation***

Rotación de la textura, en grados.

### ***bool flip\_h***

### ***bool flip\_v***

Volteado vertical y horizontal de la textura.