

# NGN\_RENDER.H

```
void Texture(  
    NGN_Texture* texture,  
    float position_x = DEFAULT_VALUE,  
    float position_y = DEFAULT_VALUE  
);
```

Dibuja una textura con el formato NGN\_Texture en la escena.

```
ngn->render->Texture(bg, 0, 0);
```

```
void Sprite(  
    NGN_Sprite* sprite,  
    float position_x = DEFAULT_VALUE,  
    float position_y = DEFAULT_VALUE  
);
```

Dibuja un Sprite con el formato NGN\_Sprite en la escena. Si no se especifica una coordenada de dibujado, se usaran las internas del sprite.

```
ngn->render->Sprite(ball);  
ngn->render->Sprite(ball, 127, 96);
```

```
void TiledBg(NGN_TiledBg* bg);
```

Dibuja un fondo compuesto de tiles con el formato NGN\_TiledBg en la escena.

```
ngn->render->TiledBg(clouds);
```

```
void TextLayer(  
    NGN_TextLayer* layer,  
    float position_x = DEFAULT_VALUE,  
    float position_y = DEFAULT_VALUE  
);
```

Dibuja una capa de texto con el formato NGN\_TextLayer en la escena.

```
ngn->render->TextLayer(textbox);
```

```
void Canvas(  
    NGN_Canvas* canvas,  
    float position_x = DEFAULT_VALUE,  
    float position_y = DEFAULT_VALUE  
);
```

Dibuja una capa de dibujado con el formato NGN\_Canvas en la escena.

```
ngn->render->Canvas(mydraw);
```

```
void Viewports();
```

Dibuja el contenido de todos los viewports habilitados en la escena.

```
ngn->render->Viewports();
```

```
void RenderToTexture(NGN_Texture* texture);
```

Selecciona y fija una textura como destino del render.

```
ngn->render->RenderToTexture(surface);
```

```
void RenderToScreen();
```

Fija la pantalla (o el VIEWPORT) como destino del render.

```
ngn->render->RenderToScreen();
```