

# NGN\_GRAPHICS.H

```
bool Init(
    std::string window_name,          // Nombre en la ventana
    uint32_t native_width,            // Resolución Nativa del juego
    uint32_t native_height,
    int8_t full_scr = NGN_SCR_WINDOW, // Pantalla completa?
    bool bilinear_filter = false,     // Filtro bilinear activado?
    bool sync = true                  // VSYNC activo?
);
```

Inicia el modo gráfico con los parámetros especificados. Devuelve FALSE en caso de error.

```
ngn->graphics->Init("nSDL Demo", 1280, 720);
```

NGN_SCR_WINDOW	0	Modo ventana
NGN_SCR_WINDOW_FULL	1	Modo ventana a pantalla completa (recomendado para Linux)
NGN_SCR_FULLSCREEN	2	Modo pantalla completa (recomendado para Windows)

## **void Update();**

Intenta actualizar la escena a 60fps. Llamar una vez por frame.

```
ngn->graphics->Update();
```

```
void SetViewportClip(
    int32_t x,    // Posición X
    int32_t y,    // Posición Y
    int32_t w,    // Ancho
    int32_t h     // Alto
);
```

Define el área visible de la pantalla.

```
ngn->graphics->SetViewportClip(100, 100, 250, 250);
```

## **void ShowMouse(bool visible);**

Muestra u oculta el cursor del ratón.

```
ngn->graphics->ShowMouse(false);
```

```
void OpenViewport(
    uint8_t id,          // ID del VIEWPORT (0-7)
    int32_t pos_x,       // Posición del viewport
    int32_t pos_y,
    uint32_t width,      // Ancho del viewport
    uint32_t height,     // Alto del viewport
    // Resolución del render en el viewport
    uint32_t h_res = NGN_DEFAULT_VALUE,
    uint32_t v_res = NGN_DEFAULT_VALUE
);
```

Abre uno de los 8 viewports disponibles con los parámetros facilitados.

```
ngn->graphics->OpenViewport(3, 480, 240, 320, 240, 160, 120);
```

```
void CloseViewport(uint8_t id);
```

Cierra el viewport con el ID facilitado.

```
ngn->graphics->CloseViewport(3);
```

```
void SelectViewport(uint8_t id);
```

Selecciona el viewport con el ID facilitado. A partir de ese momento todos los comandos de ngn->render se enviarán a ese viewport.

```
ngn->graphics->SelectViewport(3);
```

```
void ViewportPosition(uint8_t id, int32_t x, int32_t y);
void ViewportPosition(uint8_t id, Vector2I32 position);
```

Posiciona en las coordenadas facilitadas el viewport solicitado. El punto de referencia se sitúa en la coordenada superior izquierda.

```
ngn->graphics->ViewportPosition(3, 100, 200);
```

```
void DefaultViewport();
```

Devuelve el destino del render a la pantalla principal.

```
ngn->graphics->DefaultViewport();
```

### ***bool full\_screen***

Indica si el renderer está en modo pantalla completa o no.

### ***bool vsync***

Indica si el sincronismo vertical está activo.

### ***int32\_t native\_w***

### ***int32\_t native\_h***

Almacena la resolución nativa del juego.

### ***std::string window\_caption***

Guarda el texto del título de la ventana.

### ***SDL\_Rect cliparea***

Almacena los valores del área de recorte.

### ***bool force\_redraw***

Indica si debido a un cambio de modo de pantalla, debe redibujarse la escena.