

N3lib

NightFox's 3D lib.

Funciones de sistema

```
void N3_Init3D(u8 screen, u8 mode);
```

Inicializa el modo 3D en la pantalla seleccionada. De seleccionarse la pantalla inferior para el 3D, las referencias de las funciones 2D quedaran invertidas, usando 0 para referirse a la pantalla inferior y 1 a la pantalla superior.

```
void N3_Init3dEngine(void);
```

Inicializa el OpenGL y bancos de memoria necesarios para las texturas.

```
void N3_3dLayer(u8 layer);
```

Establece la prioridad de la capa de dibujado del 3D sobre las demas capas 2D. Los valores validos de 0 (por defecto) a 3.

```
void N3_DmaMemCopy(void* destination, const void* source, u32 size);
```

Copia un bloque de memoria usando, si es posible, el canal DMA.
Usalo solo para copiar bloques de RAM a VRAM y viceversa.

Funciones de carga de archivos

```
void N3_InitBuffers(void);
```

Inicializa los buffers en RAM para el almacenamiento de texturas y paletas.
Esta funcion debe ser usada 1 SOLA VEZ antes de cargar ningun archivo.

```
void N3_ResetBuffers(void);
```

Vacia todos los buffers (libera la RAM usada) e inicializa todas las estructuras de controls de estos buffers.

```
void N3_RootPath(const char* path);
```

Define el ROOT del sistema de archivos.
Si se especifica "NITROFS" como root, se usara NitroFS como sistema de archivos.
En caso contrario, se usara FAT con la ruta especificada.
Esta ruta no puede tener mas de 32 caracteres de longitud.
Esta funcion NO INICIALIZA el sistema de archivos. Usala si ya has iniciado el sistema de archivos desde otra funcion o libreria.

```
void N3_InitFilesystem(const char* path);
```

Define el ROOT del sistema de archivos.
Si se especifica "NITROFS" como root, se usara NitroFS como sistema de archivos.
En caso contrario, se usara FAT con la ruta especificada.
Esta ruta no puede tener mas de 32 caracteres de longitud.
Esta funcion INICIALIZA el sistema de archivos. Usala si no has iniciado el sistema de archivos desde otra funcion o libreria.

```
void N3_LoadTexture(const char* file, u16 id, u16 width, u16 height);
```

Carga una textura en RAM desde el sistema de archivos.
Debes especificar la ruta, slot en RAM donde cargarla y las medidas en pixeles de la textura.

```
void N3_UnloadTexture(u16 id);
```

Borra de la RAM la textura cargada en el SLOT indicado.

```
void N3_LoadPal(const char* file, u16 id);
```

Carga una paleta en RAM desde el sistema de archivos.
Debes especificar la ruta y el slot en RAM donde cargarla.

```
void N3_UnloadPal(u16 id);
```

Borra de la RAM la paleta cargada en el SLOT indicado.

Funciones de dibujo de poligonos

```
void N3_SolidQuad(    s16 x1, s16 y1, s16 z1,    Coordenadas Vertice 1
                     s16 x2, s16 y2, s16 z2,    Coordenadas Vertice 2
                     s16 x3, s16 y3, s16 z3,    Coordenadas Vertice 3
                     s16 x4, s16 y4, s16 z4,    Coordenadas Vertice 4
                     u8 r, u8 g, u8 b          Color RGB (8 bits canal)
                     );
```

Dibuja un cuadrado solido con las coordenadas y el color especificado.

```
void N3_TexturedQuad( s16 x1, s16 y1, s16 z1,    Coordenadas Vertice 1
                     s16 x2, s16 y2, s16 z2,    Coordenadas Vertice 2
                     s16 x3, s16 y3, s16 z3,    Coordenadas Vertice 3
                     s16 x4, s16 y4, s16 z4,    Coordenadas Vertice 4
                     u16 tex, u8 pal          Textura y paleta
                     );
```

Dibuja un cuadrado con las coordenadas y la textura y paleta especificados.

Funciones de manejo de la VRAM

```
void N3_VramTexture(u16 ram, u16 vram, bool keepframes);
```

Copia una textura cargada en RAM a la VRAM.
Debes especificar el SLOT de origen en RAM, el de destino en la VRAM y si se copiaran o no todos los frames (false) o solo el primero (true).

```
void N3_FreeTexture(u16 vram);
```

Borra la textura indicada de la VRAM y marca el slot como disponible.
Si es necesario, desfragmenta la VRAM.

```
void N3_VramTextureDefrag(void);
```

Reordena la VRAM usada por las texturas.

```
void N3_VramPalette(u8 ram, u8 vram);
```

Copia una Paleta cargada en RAM a la VRAM.
Debes especificar el SLOT de origen en RAM, el de destino en la VRAM.

```
void N3_TextureFrame(u8 texture, u8 frame);
```

Cambia el frame mostrar de la textura indicada.
Debes indicar el SLOT de la textura en VRAM y el numero de frame.

```
void N3_UpdateTexturesVram(void);
```

Actualiza, si es necesario, las texturas en VRAM, en caso de que estas sean animadas y hayan cambiado de frame.
Utiliza esta funcion despues del VBLANK.

Funciones de debug

```
void N3_Error(u16 code, const char* text, u32 value);
```

Maneja los codigos de error de la libreria y detiene la ejecucion.

- 101 - Fuera de rango
- 102 - El ID o SLOT estan ya en uso
- 103 - Tamaño Ilegal
- 104 - No hay suficiente memoria RAM
- 105 - Fichero no encontrado
- 106 - El ID o SLOT esta vacio, sin datos
- 107 - No hay suficiente memoria RAM
- 108 - Slot o ID no esta en uso.