GRRLIB

4.4.1

Generated by Doxygen 1.10.0

Chapter 1

GRRLIB Documentation

Welcome to the GRRLIB documentation. A complete list of functions is available from this page.

1.1 Introduction

GRRLIB is a C/C++ 2D/3D graphics library for Wii application developers. It is essentially a wrapper which presents a friendly interface to the Nintendo GX core.

1.2 Links

Code: https://github.com/GRRLIB/GRRLIB
Discussions: http://grrlib.santo.fr/forum
Chat: #GRRLIB on EFnet

1.3 Credits

Project Leader : NoNameNo
Documentation : Crayon, BlueChip
Lead Coder : NoNameNo
Support Coders : Crayon, Xane, DragonMinded, BlueChip
Advisors : RedShade, Jespa

1.4 Licence

See the LICENCE file for licence rights and limitations (MIT).

Chapter 2

LICENCE

Chapter 3

Changelog

All notable changes to this project will be documented in this file.

Unreleased

- Fixed compatibility issues with devkitPPC release 39.
- Fixed linking problem in examples with FreeType 2.13.2 (pkgrel 1 and 2).
- Added `GRRLIB_LoadTTFFromFile()` to load a TTF from a file.
- Added `GRRLIB_Ellipse()` to draw an ellipse.
- Changed `GRRLIB_PrintfTTF()` and `GRRLIB_PrintfTTFW()` so they use the alpha channel from the `color` parameter.
- Changed function arguments types in a few functions.
- Fixed documentation for `GRRLIB_Camera3dSettings()`, `GRRLIB_Screen2Texture()` and `GRRLIB_CompoEnd()`.

4.4.1 - 2021-03-05

- Patched widescreen on Wii U.

4.4.0 - 2020-01-12

- Removed libpng, zlib, libfreetype and libjpeg from the project. These libraries should now be installed in devkitPro with pacman.
- GRRLIB and pngu are now installed into the *portlibs* folder instead of the *libogc* folder.
- Removed `uint` declaration. The `u32` type should be used instead.
- The `GRRLIB_CreateEmptyTexture()` function is not inline anymore.

4.3.2 - 2012-08-12

- libpng was updated to version 1.5.12
- zlib was updated to version 1.2.7
- libjpeg was updated to version 8d
- FreeType updated to version 2.4.10
- `GRRLIB_ClampVar8()` was removed from GRRLIB
- Removed warnings from `GRRLIB_LoadBMF()`

4.3.1 - 2010-10-22

- libpng was updated to version 1.4.4
- FreeType updated to version 2.4.3
- GRRLIB is compatible with libogc 1.8.4

4.3.0 - 2010-06-28

- libpng was updated to version 1.4.2
- libjpeg was updated to version 8b
- zlib was updated to version 1.2.5
- FreeType 2.3.12 support (first support)
- 3D primitive drawing: torus, sphere, cube, cylinder and cone
- Light functions (diffuse, specular, spot) + ambient
- Split functions for rotation/translation/scaling 3d object.
- Here is the list a new added samples demo code:
  - 3d_light1 -> Simple Diffuse light sample code
  - 3d_light2 -> Simple Lights and `GRRLIB_ObjectViewInv()` sample
  - 3d_light3 -> A little Specular light sample code
  - 3d_light4 -> Spot Light Sample Code
  - 3D_sample5 -> Simple demo of rotation/translation/scaling 3d object.
  - ttf -> TrueType Font demo

4.2.0 - 2009-12-16

- First support to 3D functions
- GRRLIB_CompoStart() and GRRLIB_CompoEnd() for real GX compositing with transparency support
- GRRLIB_Screen2Texture() is now fully optimized
- USB_Gecko output facilities
- GRRLIB_Compose() was deleted since it was not fully using GX
- GRRLIB_GetColor() was deleted, the RGBA macro should be used instead
- Lot of new sample code:
  - 3D_CubedTileDemo (How to use dynamic texturing)
  - 3D_sample1 (A simple rotating flat cube)
  - 3D_sample2 (A simple rotating textured cube)
  - 3D_sample3 (A textured cube and compositing)
  - 3D_sample4 (A complex object rotating)
  - basic_drawing (How to use some basic GRRLIB functions)
  - bitmap_fx (Effects ShowRoom)
  - blending (How to use blending mode)
  - compositing (A simple compositing how to)
  - funsin (A gradient sinusoid dancing)
  - particle (A nice particle sample code)
  - template (Use this as a basis for your project)
  - TileDemo (This will show you how to use tiles/tileset and map)
  - unlimited2d (A faky technic for unlimited sprites)
  - unlimited3d (Same as above but with 3D)

4.1.1 - 2009-11-24

- Fully compatible with devkitPro release 18 and 19 (code and examples)
- libpng was updated to version 1.2.40
- Support for MS-Windows Bitmap format uncompressed (1-bit, 4-bit, 8-bit, 24-bit and 32-bit)
- New function called GRRLIB_DrawPart() to draw a specific part of a texture
- Extra parameters to GRRLIB_Screen2Texture()
- Video is now initialized even without a SD card
- Fixed a problem with scaling images
- Documentation improvement

4.1.0 - 2009-10-05

- Completely new file structure with sub-folders
- Installer for vendor libraries (jpeg, png, pngu)
- Makefile for GRRLIB
- Many (all suitable) functions are now inlined
- Support for the recent changes to libogc
- Alpha compositor function added
- Correct use of pointers (no more struct passing)
- Speed improvement, bug fixing and more...

4.0.0 - 2009-03-05

- Changed color format to RGBA for ALL GRRLib functions to fit to `GXColor` format and use `GX_Color1u32`
- `GRRLIB_LoadTexture()` now auto detect PNG or JPEG
- GRRLib introduce a new texture structure (easier to handle texture width, height, etc ...)
- Add `GRRLIB_InitTileSet()` to initialize a tile set
- `GRRLIB_DrawImg()` recoded for simpler use
- `GRRLIB_DrawTile()` recoded for simpler use
- `InitVideo()` and `GRRLIB_Start()` merge into `GRRLIB_Init()`.
- Add `GRRLIB_PtInRect()`, `GRRLIB_RectInRect()` and `GRRLIB_RectOnRect()` to detect hot zone
- `GRRLIB_GetPixelFromtexImg()` and `GRRLIB_SetPixelTotexImg()` to directly read/write in texture
- `GRRLIB_CreateEmptyTexture()` and `GRRLIB_FlushTex()`
- New Bitmap FX
- Add `GRRLIB_Exit()` to free the memory allocated by GRRLIB

Chapter 4

Topic Index

4.1 Topics

Here is a list of all topics with brief descriptions:

Chapter 5

Data Structure Index

5.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 6

File Index

6.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 7

Topic Documentation

7.1 Everything in GRRLIB

This is the complete list of functions, structures, defines, typedefs, enumerations and variables you may want to used to make your homebrew with GRRLIB.

Data Structures

- struct GRRLIB_drawSettings

  Structure to hold the current drawing settings.
- struct GRRLIB_texImg

  Structure to hold the texture information.
- struct GRRLIB_bytemapChar

  Structure to hold the bytemap character information.
- struct GRRLIB_bytemapFont

  Structure to hold the bytemap font information.
- struct GRRLIB_Font

  Structure to hold the TTF information.

Macros

- #define **GRRLIB_VER_STRING** "4.4.1"

  Version information for GRRLIB.
- #define **R**(c) (((c) >>24) &0xFF)

  Extract red component of colour.
- #define **G**(c) (((c) >>16) &0xFF)

  Extract green component of colour.
- #define **B**(c) (((c) >> 8) &0xFF)

  Extract blue component of colour.
- #define **A**(c) ((c) &0xFF)

  Extract alpha component of colour.
- #define RGBA(r, g, b, a)

  Build an RGB pixel from components.
- #define **GRRLIB_BLEND_NONE** (GRRLIB_BLEND_ALPHA)

Alias for GRRLIB_BLEND_ALPHA.

- #define **GRRLIB_BLEND_LIGHT** (GRRLIB_BLEND_ADD)

  Alias for GRRLIB_BLEND_ADD.
- #define **GRRLIB_BLEND_SHADE** (GRRLIB_BLEND_MULTI)

  Alias for GRRLIB_BLEND_MULTI.
- #define **GRR_EXTERN** extern
- #define **GRR_INIT**(v)
- #define **GRR_INITS**(...)
- #define **INLINE** static inline

Typedefs

- typedef enum GRRLIB_blendMode **GRRLIB_blendMode**

  GRRLIB Blending Modes.
- typedef struct GRRLIB_drawSettings **GRRLIB_drawSettings**

  Structure to hold the current drawing settings.
- typedef struct GRRLIB_texImg **GRRLIB_texImg**

  Structure to hold the texture information.
- typedef struct GRRLIB_bytemapChar **GRRLIB_bytemapChar**

  Structure to hold the bytemap character information.
- typedef struct GRRLIB_bytemapFont **GRRLIB_bytemapFont**

  Structure to hold the bytemap font information.
- typedef struct GRRLIB_Font **GRRLIB_ttfFont**

  Structure to hold the TTF information.

Enumerations

- enum GRRLIB_blendMode {
  GRRLIB_BLEND_ALPHA = 0 , GRRLIB_BLEND_ADD = 1 , GRRLIB_BLEND_SCREEN = 2 ,
  GRRLIB_BLEND_MULTI = 3 ,
  GRRLIB_BLEND_INV = 4 }

  GRRLIB Blending Modes.

Functions

- GRR_EXTERN void *xfb[2] **GRR_INITS** (NULL, NULL)
- GRR_EXTERN u32 fb **GRR_INIT** (0)
- INLINE void **GRRLIB_ClipReset** (void)

  Reset the clipping to normal.
- INLINE void GRRLIB_ClipDrawing (const u32 x, const u32 y, const u32 width, const u32 height)

  Clip the drawing area to an rectangle.
- INLINE bool GRRLIB_PtInRect (const int hotx, const int hoty, const int hotw, const int hoth, const int wpadx, const int wpady)

  Determine whether the specified point lies within the specified rectangle.
- INLINE bool GRRLIB_RectInRect (const int rect1x, const int rect1y, const int rect1w, const int rect1h, const int rect2x, const int rect2y, const int rect2w, const int rect2h)

  Determine whether a specified rectangle lies within another rectangle.
- INLINE bool GRRLIB_RectOnRect (const int rect1x, const int rect1y, const int rect1w, const int rect1h, const int rect2x, const int rect2y, const int rect2w, const int rect2h)

  Determine whether a part of a specified rectangle lies on another rectangle.

- INLINE void GRRLIB_NPlot (const guVector v[ ], const u32 color[ ], const u16 n)

  Draw an array of points.
- INLINE void GRRLIB_NGone (const guVector v[ ], const u32 color[ ], const u16 n)

  Draw a polygon.
- INLINE void GRRLIB_NGoneFilled (const guVector v[ ], const u32 color[ ], const u16 n)

  Draw a filled polygon.
- INLINE void GRRLIB_GXEngine (const guVector v[ ], const u32 color[ ], const u16 n, const u8 fmt)

  Draws a vector.
- INLINE void GRRLIB_FillScreen (const u32 color)

  Clear screen with a specific color.
- INLINE void GRRLIB_Plot (const f32 x, const f32 y, const u32 color)

  Draw a dot.
- INLINE void GRRLIB_Line (const f32 x1, const f32 y1, const f32 x2, const f32 y2, const u32 color)

  Draw a line.
- INLINE void GRRLIB_Rectangle (const f32 x, const f32 y, const f32 width, const f32 height, const u32 color, const bool filled)

  Draw a rectangle.
- INLINE void GRRLIB_SetHandle (GRRLIB_texImg *tex, const int x, const int y)

  Set a texture's X and Y handles.
- INLINE void GRRLIB_SetMidHandle (GRRLIB_texImg *tex, const bool enabled)

  Center a texture's handles.
- INLINE u32 GRRLIB_GetPixelFromtexImg (const int x, const int y, const GRRLIB_texImg *tex)

  Return the color value of a pixel from a GRRLIB_texImg.
- INLINE void GRRLIB_SetPixelTotexImg (const int x, const int y, GRRLIB_texImg *tex, const u32 color)

  Set the color value of a pixel to a GRRLIB_texImg.
- INLINE u32 GRRLIB_GetPixelFromFB (int x, int y)

  Reads a pixel directly from the FrontBuffer.
- INLINE void GRRLIB_SetPixelToFB (int x, int y, u32 pokeColor)

  Writes a pixel directly from the FrontBuffer.
- INLINE void GRRLIB_SetBlend (const GRRLIB_blendMode blendmode)

  Set a blending mode.
- INLINE GRRLIB_blendMode GRRLIB_GetBlend (void)

  Get the current blending mode.
- INLINE void GRRLIB_SetAntiAliasing (const bool aa)

  Turn anti-aliasing on/off.
- INLINE bool GRRLIB_GetAntiAliasing (void)

  Get current anti-aliasing setting.
- INLINE void GRRLIB_ClearTex (GRRLIB_texImg *tex)

  Clear a texture to transparent black.
- INLINE void GRRLIB_FlushTex (GRRLIB_texImg *tex)

  Write the contents of a texture in the data cache down to main memory.
- INLINE void GRRLIB_FreeTexture (GRRLIB_texImg *tex)

  Free memory allocated for texture.
- GRRLIB_bytemapFont * GRRLIB_LoadBMF (const u8 my_bmf[ ])

  Load a ByteMap font structure from a buffer.
- void GRRLIB_FreeBMF (GRRLIB_bytemapFont *bmf)

  Free memory allocated by ByteMap fonts.
- void GRRLIB_InitTileSet (GRRLIB_texImg *tex, const u32 tilew, const u32 tileh, const u32 tilestart)

  Initialize a tile set.
- void GRRLIB_BMFX_FlipH (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest)

  Flip texture horizontal.

- void GRRLIB_BMFX_FlipV (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest)

  Flip texture vertical.
- void GRRLIB_BMFX_Grayscale (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest)

  Change a texture to gray scale.
- void GRRLIB_BMFX_Sepia (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest)

  Change a texture to sepia (old photo style).
- void GRRLIB_BMFX_Invert (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest)

  Invert colors of the texture.
- void GRRLIB_BMFX_Blur (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest, const u32 factor)

  A texture effect (Blur).
- void GRRLIB_BMFX_Scatter (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest, const u32 factor)

  A texture effect (Scatter).
- void GRRLIB_BMFX_Pixelate (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest, const u32 factor)

  A texture effect (Pixelate).
- int GRRLIB_Init (void)

  Initialize GRRLIB.
- void GRRLIB_Exit (void)

  Call this before exiting your application.
- void GRRLIB_Ellipse (const f32 x, const f32 y, const f32 radiusX, const f32 radiusY, const u32 color, const u8 filled)

  Draw an ellipse.
- void GRRLIB_Circle (const f32 x, const f32 y, const f32 radius, const u32 color, const u8 filled)

  Draw a circle.
- int GRRLIB_LoadFile (const char *filename, u8 **data)

  Load a file to memory.
- GRRLIB_texImg * GRRLIB_LoadTextureFromFile (const char *filename)

  Load a texture from a file.
- GRRLIB_ttfFont * GRRLIB_LoadTTFFromFile (const char *filename)

  Load a TTF from a file.
- bool GRRLIB_ScrShot (const char *filename)

  Make a PNG screenshot.
- void GRRLIB_Printf (const f32 xpos, const f32 ypos, const GRRLIB_texImg *tex, const u32 color, const f32 zoom, const char *text,...)

  Print formatted output.
- void GRRLIB_PrintBMF (const f32 xpos, const f32 ypos, const GRRLIB_bytemapFont *bmf, const char *text,...)

  Print formatted output with a ByteMap font.
- void GRRLIB_DrawImg (const f32 xpos, const f32 ypos, const GRRLIB_texImg *tex, const f32 degrees, const f32 scaleX, const f32 scaleY, const u32 color)

  Draw a texture.
- void GRRLIB_DrawImgQuad (const guVector pos[4], GRRLIB_texImg *tex, const u32 color)

  Draw a textured quad.
- void GRRLIB_DrawTile (const f32 xpos, const f32 ypos, const GRRLIB_texImg *tex, const f32 degrees, const f32 scaleX, const f32 scaleY, const u32 color, const int frame)

  Draw a tile.
- void GRRLIB_DrawPart (const f32 xpos, const f32 ypos, const f32 partx, const f32 party, const f32 partw, const f32 parth, const GRRLIB_texImg *tex, const f32 degrees, const f32 scaleX, const f32 scaleY, const u32 color)

  Draw a part of a texture.
- void GRRLIB_DrawTileQuad (const guVector pos[4], GRRLIB_texImg *tex, const u32 color, const int frame)

  Draw a tile in a quad.
- void **GRRLIB_Render** (void)

Call this function after drawing.

- void GRRLIB_Screen2Texture (u16 posx, u16 posy, GRRLIB_texImg *tex, bool clear)

  Make a snapshot of the screen in a texture WITHOUT ALPHA LAYER.
- void GRRLIB_CompoStart (void)

  Start GX compositing process.
- void GRRLIB_CompoEnd (u16 posx, u16 posy, GRRLIB_texImg *tex)

  End GX compositing process (Make a snapshot of the screen in a texture WITH ALPHA LAYER).
- GRRLIB_texImg * GRRLIB_CreateEmptyTexture (const u32 width, const u32 height)

  Create an empty texture.
- GRRLIB_texImg * GRRLIB_LoadTexture (const u8 *my_img)

  Load a texture from a buffer.
- GRRLIB_texImg * GRRLIB_LoadTexturePNG (const u8 *my_png)

  Load a texture from a buffer.
- GRRLIB_texImg * GRRLIB_LoadTextureJPG (const u8 *my_jpg)

  Load a texture from a buffer.
- GRRLIB_texImg * GRRLIB_LoadTextureJPGEx (const u8 *my_jpg, const u32 my_size)

  Load a texture from a buffer.
- GRRLIB_texImg * GRRLIB_LoadTextureBMP (const u8 *my_bmp)

  Load a texture from a buffer.
- bool GRRLIB_GeckoInit (void)

  Initialize USB Gecko.
- void GRRLIB_GeckoPrintf (const char *text,...)

  Print Gecko.
- void GRRLIB_SetBackgroundColour (u8 r, u8 g, u8 b, u8 a)

  Set the background parameter when screen is cleared.
- void GRRLIB_Camera3dSettings (f32 posx, f32 posy, f32 posz, f32 upx, f32 upy, f32 upz, f32 lookx, f32 looky, f32 lookz)

  Set the camera parameter (contributed by chris_c aka DaShAmAn).
- void GRRLIB_3dMode (f32 minDist, f32 maxDist, f32 fov, bool texturemode, bool normalmode)

  Set up the position matrix (contributed by chris_c aka DaShAmAn).
- void **GRRLIB_2dMode** (void)

  Go back to 2D mode (contributed by chris_c aka DaShAmAn).
- void **GRRLIB_ObjectViewBegin** (void)

  Init the object matrix to draw object.
- void GRRLIB_ObjectViewScale (f32 scalx, f32 scaly, f32 scalz)

  Scale the object matrix to draw object.
- void GRRLIB_ObjectViewRotate (f32 angx, f32 angy, f32 angz)

  Rotate the object matrix to draw object.
- void GRRLIB_ObjectViewTrans (f32 posx, f32 posy, f32 posz)

  Translate the object matrix to draw object.
- void **GRRLIB_ObjectViewEnd** (void)

  Concat the object and the view matrix and calculate the inverse normal matrix.
- void GRRLIB_ObjectView (f32 posx, f32 posy, f32 posz, f32 angx, f32 angy, f32 angz, f32 scalx, f32 scaly, f32 scalz)

  Set the view matrix to draw object (in this order scale, rotate AND trans).
- void GRRLIB_ObjectViewInv (f32 posx, f32 posy, f32 posz, f32 angx, f32 angy, f32 angz, f32 scalx, f32 scaly, f32 scalz)

  Set the view matrix to draw object (in this order scale, trans AND rotate).
- void GRRLIB_SetTexture (GRRLIB_texImg *tex, bool rep)

  Set the texture to an object (contributed by chris_c aka DaShAmAn).
- void GRRLIB_DrawTorus (f32 r, f32 R, int nsides, int rings, bool filled, u32 col)

Draw a torus (with normal).

- void GRRLIB_DrawSphere (f32 r, int lats, int longs, bool filled, u32 col)

  Draw a sphere (with normal).
- void GRRLIB_DrawCube (f32 size, bool filled, u32 col)

  Draw a cube (with normal).
- void GRRLIB_DrawCylinder (f32 r, f32 h, u16 d, bool filled, u32 col)

  Draw a cylinder (with normal).
- void GRRLIB_DrawCone (f32 r, f32 h, u16 d, bool filled, u32 col)

  Draw a cone (with normal).
- void GRRLIB_DrawTessPanel (f32 w, f32 wstep, f32 h, f32 hstep, bool filled, u32 col)

  Draw a Tesselated panel (with normal).
- void GRRLIB_SetLightAmbient (u32 ambientcolor)

  Set ambient color.
- void GRRLIB_SetLightDiff (u8 num, guVector pos, f32 distattn, f32 brightness, u32 lightcolor)

  Set diffuse light parameters.
- void GRRLIB_SetLightSpec (u8 num, guVector dir, f32 shininess, u32 lightcolor, u32 speccolor)

  Set specular light parameters.
- void GRRLIB_SetLightSpot (u8 num, guVector pos, guVector lookat, f32 angAttn0, f32 angAttn1, f32 ang↩ Attn2, f32 distAttn0, f32 distAttn1, f32 distAttn2, u32 lightcolor)

  Set Spot light parameters.
- void **GRRLIB_SetLightOff** (void)

  Set all lights off, like at init.
- GRRLIB_ttfFont ∗ GRRLIB_LoadTTF (const u8 ∗file_base, s32 file_size)

  Load a TTF from a buffer.
- void GRRLIB_FreeTTF (GRRLIB_ttfFont ∗myFont)

  Free memory allocated by TTF fonts.
- void GRRLIB_PrintfTTF (int x, int y, GRRLIB_ttfFont ∗myFont, const char ∗string, unsigned int fontSize, const u32 color)

  Print function for TTF font.
- void GRRLIB_PrintfTTFW (int x, int y, GRRLIB_ttfFont ∗myFont, const wchar_t ∗string, unsigned int fontSize, const u32 color)

  Print function for TTF font.
- u32 GRRLIB_WidthTTF (GRRLIB_ttfFont ∗myFont, const char ∗, unsigned int)

  Get the width of a text in pixel.
- u32 GRRLIB_WidthTTFW (GRRLIB_ttfFont ∗myFont, const wchar_t ∗, unsigned int)

  Get the width of a text in pixel.

Variables

- GRR_EXTERN GXRModeObj ∗ **rmode**

  Video mode.

7.1.1 Detailed Description

This is the complete list of functions, structures, defines, typedefs, enumerations and variables you may want to used to make your homebrew with GRRLIB.

You simply need to include grrlib.h in your project to have access to all of these.

7.1.2 Macro Definition Documentation

7.1.2.1 RGBA

```
#define RGBA(
               r,
               g,
               b,
               a )
```

Value:

```
( (u32)( ( ((u32)(r))         «24) |  \
((((u32)(g)) &0xFF) «16) |  \
((((u32)(b)) &0xFF) « 8) |  \
( ((u32)(a)) &0xFF       ) ) )
```

Build an RGB pixel from components.

Parameters

r	Red component.
g	Green component.
b	Blue component.
a	Alpha component.

7.1.3 Enumeration Type Documentation

7.1.3.1 GRRLIB_blendMode

```
enum GRRLIB_blendMode
```

GRRLIB Blending Modes.

Enumerator

GRRLIB_BLEND_ALPHA	Alpha Blending.
GRRLIB_BLEND_ADD	Additive Blending.
GRRLIB_BLEND_SCREEN	Alpha Light Blending.
GRRLIB_BLEND_MULTI	Multiply Blending.
GRRLIB_BLEND_INV	Invert Color Blending.

7.1.4 Function Documentation

7.1.4.1 GRRLIB_3dMode()

```
void GRRLIB_3dMode (
             f32 minDist,
             f32 maxDist,
             f32 fov,
```

```
            bool texturemode,
            bool normalmode )
```

Set up the position matrix (contributed by chris_c aka DaShAmAn).

Parameters

minDist	Minimal distance for the camera.
maxDist	Maximal distance for the camera.
fov	Field of view for the camera.
texturemode	False, GX won't need texture coordinate, True, GX will need texture coordinate.
normalmode	False, GX won't need normal coordinate, True, GX will need normal coordinate.

7.1.4.2 GRRLIB_BMFX_Blur()

```
void GRRLIB_BMFX_Blur (
            const GRRLIB_texImg * texsrc,
            GRRLIB_texImg * texdest,
            const u32 factor )
```

A texture effect (Blur).

See also

GRRLIB_FlushTex

Parameters

texsrc	The texture source.
texdest	The texture destination.
factor	The blur factor.

7.1.4.3 GRRLIB_BMFX_FlipH()

```
void GRRLIB_BMFX_FlipH (
            const GRRLIB_texImg * texsrc,
            GRRLIB_texImg * texdest )
```

Flip texture horizontal.

See also

GRRLIB_FlushTex

Parameters

texsrc	The texture source.
texdest	The texture destination.

7.1.4.4 GRRLIB_BMFX_FlipV()

```
void GRRLIB_BMFX_FlipV (
            const GRRLIB_texImg * texsrc,
            GRRLIB_texImg * texdest )
```

Flip texture vertical.

See also

GRRLIB_FlushTex

Parameters

texsrc	The texture source.
texdest	The texture destination.

7.1.4.5 GRRLIB_BMFX_Grayscale()

```
void GRRLIB_BMFX_Grayscale (
            const GRRLIB_texImg * texsrc,
            GRRLIB_texImg * texdest )
```

Change a texture to gray scale.

See also

GRRLIB_FlushTex

Parameters

texsrc	The texture source.
texdest	The texture grayscaled destination.

7.1.4.6 GRRLIB_BMFX_Invert()

```
void GRRLIB_BMFX_Invert (
            const GRRLIB_texImg * texsrc,
            GRRLIB_texImg * texdest )
```

Invert colors of the texture.

See also

GRRLIB_FlushTex

Parameters

texsrc	The texture source.
texdest	The texture destination.

7.1.4.7 GRRLIB_BMFX_Pixelate()

```
void GRRLIB_BMFX_Pixelate (
            const GRRLIB_texImg * texsrc,
            GRRLIB_texImg * texdest,
            const u32 factor )
```

A texture effect (Pixelate).

See also

GRRLIB_FlushTex

Parameters

texsrc	The texture source.
texdest	The texture destination.
factor	The factor level of the effect.

7.1.4.8 GRRLIB_BMFX_Scatter()

```
void GRRLIB_BMFX_Scatter (
            const GRRLIB_texImg * texsrc,
            GRRLIB_texImg * texdest,
            const u32 factor )
```

A texture effect (Scatter).

See also

GRRLIB_FlushTex

Parameters

texsrc	The texture source.
texdest	The texture destination.
factor	The factor level of the effect.

7.1.4.9 GRRLIB_BMFX_Sepia()

```
void GRRLIB_BMFX_Sepia (
            const GRRLIB_texImg * texsrc,
            GRRLIB_texImg * texdest )
```

Change a texture to sepia (old photo style).

See also

GRRLIB_FlushTex

Parameters

texsrc	The texture source.
texdest	The texture destination.

Author

elisherer

7.1.4.10 GRRLIB_Camera3dSettings()

```
void GRRLIB_Camera3dSettings (
            f32 posx,
            f32 posy,
            f32 posz,
            f32 upx,
            f32 upy,
            f32 upz,
            f32 lookx,
            f32 looky,
            f32 lookz )
```

Set the camera parameter (contributed by chris_c aka DaShAmAn).

Parameters

posx	x position of the camera.
posy	y position of the camera.
posz	z position of the camera.
upx	x up position of the camera.
upy	y up position of the camera.
upz	z up position of the camera.
lookx	x position of the target.
looky	y position of the target.
lookz	z position of the target.

7.1.4.11 GRRLIB_Circle()

```
void GRRLIB_Circle (
            const f32 x,
            const f32 y,
            const f32 radius,
            const u32 color,
            const u8 filled )
```

Draw a circle.

Parameters

x	Specifies the x-coordinate of the circle.

Parameters

y	Specifies the y-coordinate of the circle.
radius	The radius of the circle.
color	The color of the circle in RGBA format.
filled	Set to `true` to fill the circle.

7.1.4.12 GRRLIB_ClearTex()

```
INLINE void GRRLIB_ClearTex (
            GRRLIB_texImg * tex )
```

Clear a texture to transparent black.

Parameters

tex	Texture to clear.

7.1.4.13 GRRLIB_ClipDrawing()

```
INLINE void GRRLIB_ClipDrawing (
            const u32 x,
            const u32 y,
            const u32 width,
            const u32 height )
```

Clip the drawing area to an rectangle.

Parameters

x	The x-coordinate of the rectangle.
y	The y-coordinate of the rectangle.
width	The width of the rectangle.
height	The height of the rectangle.

7.1.4.14 GRRLIB_CompoEnd()

```
void GRRLIB_CompoEnd (
            u16 posx,
            u16 posy,
            GRRLIB_texImg * tex )
```

End GX compositing process (Make a snapshot of the screen in a texture WITH ALPHA LAYER).

EFB is cleared after this function.

See also

GRRLIB_CompoStart

Parameters

posx	Top left corner of the grabbed part.
posy	Top left corner of the grabbed part.
tex	A pointer to a texture representing the screen.

7.1.4.15 GRRLIB_CompoStart()

```
void GRRLIB_CompoStart (
            void  )
```

Start GX compositing process.

See also

GRRLIB_CompoEnd

7.1.4.16 GRRLIB_CreateEmptyTexture()

```
GRRLIB_texImg * GRRLIB_CreateEmptyTexture (
            const u32 width,
            const u32 height )
```

Create an empty texture.

Parameters

width	Width of the new texture to create.
height	Height of the new texture to create.

Returns

A GRRLIB_texImg structure newly created.

7.1.4.17 GRRLIB_DrawCone()

```
void GRRLIB_DrawCone (
            f32 r,
            f32 h,
            u16 d,
            bool filled,
            u32 col )
```

Draw a cone (with normal).

Parameters

r	Radius of the cone.

Parameters

h	High of the cone.
d	Density of slice.
filled	Wired or not.
col	Color of the cone.

7.1.4.18 GRRLIB_DrawCube()

```
void GRRLIB_DrawCube (
            f32 size,
            bool filled,
            u32 col )
```

Draw a cube (with normal).

Parameters

size	Size of the cube edge.
filled	Wired or not.
col	Color of the cube.

7.1.4.19 GRRLIB_DrawCylinder()

```
void GRRLIB_DrawCylinder (
            f32 r,
            f32 h,
            u16 d,
            bool filled,
            u32 col )
```

Draw a cylinder (with normal).

Parameters

r	Radius of the cylinder.
h	High of the cylinder.
d	Density of slice.
filled	Wired or not.
col	Color of the cylinder.

7.1.4.20 GRRLIB_DrawImg()

```
void GRRLIB_DrawImg (
            const f32 xpos,
            const f32 ypos,
            const GRRLIB_texImg * tex,
```

```
            const f32 degrees,
            const f32 scaleX,
            const f32 scaleY,
            const u32 color )
```

Draw a texture.

Parameters

xpos	Specifies the x-coordinate of the upper-left corner.
ypos	Specifies the y-coordinate of the upper-left corner.
tex	The texture to draw.
degrees	Angle of rotation.
scaleX	Specifies the x-coordinate scale. -1 could be used for flipping the texture horizontally.
scaleY	Specifies the y-coordinate scale. -1 could be used for flipping the texture vertically.
color	Color in RGBA format.

7.1.4.21 GRRLIB_DrawImgQuad()

```
void GRRLIB_DrawImgQuad (
            const guVector pos[4],
            GRRLIB_texImg * tex,
            const u32 color )
```

Draw a textured quad.

Parameters

pos	Vector array of the 4 points.
tex	The texture to draw.
color	Color in RGBA format.

7.1.4.22 GRRLIB_DrawPart()

```
void GRRLIB_DrawPart (
            const f32 xpos,
            const f32 ypos,
            const f32 partx,
            const f32 party,
            const f32 partw,
            const f32 parth,
            const GRRLIB_texImg * tex,
            const f32 degrees,
            const f32 scaleX,
            const f32 scaleY,
            const u32 color )
```

Draw a part of a texture.

Parameters

xpos	Specifies the x-coordinate of the upper-left corner.
ypos	Specifies the y-coordinate of the upper-left corner.
partx	Specifies the x-coordinate of the upper-left corner in the texture.
party	Specifies the y-coordinate of the upper-left corner in the texture.
partw	Specifies the width in the texture.
parth	Specifies the height in the texture.
tex	The texture containing the tile to draw.
degrees	Angle of rotation.
scaleX	Specifies the x-coordinate scale. -1 could be used for flipping the texture horizontally.
scaleY	Specifies the y-coordinate scale. -1 could be used for flipping the texture vertically.
color	Color in RGBA format.

7.1.4.23 GRRLIB_DrawSphere()

```
void GRRLIB_DrawSphere (
            f32 r,
            int lats,
            int longs,
            bool filled,
            u32 col )
```

Draw a sphere (with normal).

Parameters

r	Radius of the sphere.
lats	Number of latitudes.
longs	Number of longitudes.
filled	Wired or not.
col	Color of the sphere.

7.1.4.24 GRRLIB_DrawTessPanel()

```
void GRRLIB_DrawTessPanel (
            f32 w,
            f32 wstep,
            f32 h,
            f32 hstep,
            bool filled,
            u32 col )
```

Draw a Tesselated panel (with normal).

Parameters

w	Width of the panel.
wstep	Size of width slices.

Parameters

h	Height of the panel.
hstep	Size the de height slices.
filled	Wired or not.
col	Color in RGBA format.

7.1.4.25 GRRLIB_DrawTile()

```
void GRRLIB_DrawTile (
            const f32 xpos,
            const f32 ypos,
            const GRRLIB_texImg * tex,
            const f32 degrees,
            const f32 scaleX,
            const f32 scaleY,
            const u32 color,
            const int frame )
```

Draw a tile.

Parameters

xpos	Specifies the x-coordinate of the upper-left corner.
ypos	Specifies the y-coordinate of the upper-left corner.
tex	The texture containing the tile to draw.
degrees	Angle of rotation.
scaleX	Specifies the x-coordinate scale. -1 could be used for flipping the texture horizontally.
scaleY	Specifies the y-coordinate scale. -1 could be used for flipping the texture vertically.
color	Color in RGBA format.
frame	Specifies the frame to draw.

7.1.4.26 GRRLIB_DrawTileQuad()

```
void GRRLIB_DrawTileQuad (
            const guVector pos[4],
            GRRLIB_texImg * tex,
            const u32 color,
            const int frame )
```

Draw a tile in a quad.

Parameters

pos	Vector array of the 4 points.
tex	The texture to draw.
color	Color in RGBA format.
frame	Specifies the frame to draw.

7.1.4.27 GRRLIB_DrawTorus()

```
void GRRLIB_DrawTorus (
            f32 r,
            f32 R,
            int nsides,
            int rings,
            bool filled,
            u32 col )
```

Draw a torus (with normal).

Parameters

r	Radius of the ring.
R	Radius of the torus.
nsides	Number of faces per ring.
rings	Number of rings.
filled	Wired or not.
col	Color of the torus.

7.1.4.28 GRRLIB_Ellipse()

```
void GRRLIB_Ellipse (
            const f32 x,
            const f32 y,
            const f32 radiusX,
            const f32 radiusY,
            const u32 color,
            const u8 filled )
```

Draw an ellipse.

Author

Dark_Link

Parameters

x	Specifies the x-coordinate of the ellipse.
y	Specifies the y-coordinate of the ellipse.
radiusX	The X radius of the ellipse.
radiusY	The Y radius of the ellipse.
color	The color of the ellipse in RGBA format.
filled	Set to `true` to fill the ellipse.

7.1.4.29 GRRLIB_Exit()

```
void GRRLIB_Exit (
            void  )
```

Call this before exiting your application.

Ensure this function is only ever called once and only if the setup function has been called.

7.1.4.30 GRRLIB_FillScreen()

```
INLINE void GRRLIB_FillScreen (
             const u32 color )
```

Clear screen with a specific color.

Parameters

color	The color to use to fill the screen.

7.1.4.31 GRRLIB_FlushTex()

```
INLINE void GRRLIB_FlushTex (
             GRRLIB_texImg * tex )
```

Write the contents of a texture in the data cache down to main memory.

For performance the CPU holds a data cache where modifications are stored before they get written down to main memory.

Parameters

tex	The texture to flush.

7.1.4.32 GRRLIB_FreeBMF()

```
void GRRLIB_FreeBMF (
             GRRLIB_bytemapFont * bmf )
```

Free memory allocated by ByteMap fonts.

If *bmf* is a null pointer, the function does nothing.

Note

This function does not change the value of *bmf* itself, hence it still points to the same (now invalid) location.

Parameters

bmf	A GRRLIB_bytemapFont structure.

7.1.4.33 GRRLIB_FreeTexture()

```
INLINE void GRRLIB_FreeTexture (
             GRRLIB_texImg * tex )
```

Free memory allocated for texture.

If *tex* is a null pointer, the function does nothing.

Note

This function does not change the value of *tex* itself, hence it still points to the same (now invalid) location.

Parameters

tex	A GRRLIB_texImg structure.

7.1.4.34 GRRLIB_FreeTTF()

```
void GRRLIB_FreeTTF (
             GRRLIB_ttfFont * myFont )
```

Free memory allocated by TTF fonts.

If *myFont* is a null pointer, the function does nothing.

Note

This function does not change the value of *myFont* itself, hence it still points to the same (now invalid) location.

Parameters

myFont	A TTF.

7.1.4.35 GRRLIB_GeckoInit()

```
bool GRRLIB_GeckoInit (
             void  )
```

Initialize USB Gecko.

Returns

Returns `true` if everything worked, `false` if problems occurred.

7.1.4.36 GRRLIB_GeckoPrintf()

```
void GRRLIB_GeckoPrintf (
             const char * text,
              ...  )
```

Print Gecko.

Parameters

text	Text to print.
...	Optional arguments.

7.1.4.37 GRRLIB_GetAntiAliasing()

```
INLINE bool GRRLIB_GetAntiAliasing (
            void  )
```

Get current anti-aliasing setting.

Returns

Returns `true` if anti-aliasing is enabled.

7.1.4.38 GRRLIB_GetBlend()

```
INLINE GRRLIB_blendMode GRRLIB_GetBlend (
            void  )
```

Get the current blending mode.

Returns

The current blending mode.

7.1.4.39 GRRLIB_GetPixelFromFB()

```
INLINE u32 GRRLIB_GetPixelFromFB (
            int x,
            int y )
```

Reads a pixel directly from the FrontBuffer.

Parameters

x	The x-coordinate within the FB.
y	The y-coordinate within the FB.

Returns

The color of a pixel in RGBA format.

7.1.4.40 GRRLIB_GetPixelFromtexImg()

```
INLINE u32 GRRLIB_GetPixelFromtexImg (
```

```
            const int x,
            const int y,
            const GRRLIB_texImg * tex )
```

Return the color value of a pixel from a GRRLIB_texImg.

Parameters

x	Specifies the x-coordinate of the pixel in the texture.
y	Specifies the y-coordinate of the pixel in the texture.
tex	The texture to get the color from.

Returns

The color of a pixel in RGBA format.

7.1.4.41 GRRLIB_GXEngine()

```
INLINE void GRRLIB_GXEngine (
            const guVector v[],
            const u32 color[],
            const u16 n,
            const u8 fmt )
```

Draws a vector.

Parameters

v	The vector to draw.
color	The color of the vector in RGBA format.
n	Number of points in the vector. The maximum is 65536.
fmt	Type of primitive.

7.1.4.42 GRRLIB_Init()

```
int GRRLIB_Init (
            void  )
```

Initialize GRRLIB.

Call this once at the beginning your code.

Returns

A integer representing a code:

- 0 : The operation completed successfully.
- -1 : Not enough memory is available to initialize GRRLIB.
- -2 : Failed to add the fat device driver to the devoptab.
- -3 : Failed to initialize the font engine.

See also

GRRLIB_Exit

7.1.4.43 GRRLIB_InitTileSet()

```
void GRRLIB_InitTileSet (
            GRRLIB_texImg * tex,
            const u32 tilew,
            const u32 tileh,
            const u32 tilestart )
```

Initialize a tile set.

Parameters

tex	The texture to initialize.
tilew	Width of the tile.
tileh	Height of the tile.
tilestart	Offset for starting position (Used in fonts).

7.1.4.44 GRRLIB_Line()

```
INLINE void GRRLIB_Line (
            const f32 x1,
            const f32 y1,
            const f32 x2,
            const f32 y2,
            const u32 color )
```

Draw a line.

Parameters

x1	Starting point for line for the x coordinate.
y1	Starting point for line for the y coordinate.
x2	Ending point for line for the x coordinate.
y2	Ending point for line for the x coordinate.
color	Line color in RGBA format.

Author

JESPA

7.1.4.45 GRRLIB_LoadBMF()

```
GRRLIB_bytemapFont * GRRLIB_LoadBMF (
            const u8 my_bmf[] )
```

Load a ByteMap font structure from a buffer.

File format version 1.1 is used, more information could be found at https://bmf.php5.cz/?page=format

Parameters

my_bmf	The ByteMap font buffer to load.

Returns

A GRRLIB_bytemapFont structure filled with BMF information.

See also

GRRLIB_FreeBMF

7.1.4.46 GRRLIB_LoadFile()

```
int GRRLIB_LoadFile (
            const char * filename,
            u8 ** data )
```

Load a file to memory.

Parameters

filename	Name of the file to be loaded.
data	Pointer-to-your-pointer. Ie. { u8 *data; GRRLIB_LoadFile("file", &data); }. It is your responsibility to free the memory allocated by this function.

Returns

A integer representing a code:

- 0 : EmptyFile.
- -1 : FileNotFound.
- -2 : OutOfMemory.
- -3 : FileReadError.
- >0 : FileLength.

7.1.4.47 GRRLIB_LoadTexture()

```
GRRLIB_texImg * GRRLIB_LoadTexture (
            const u8 * my_img )
```

Load a texture from a buffer.

Parameters

my_img	The JPEG, PNG or Bitmap buffer to load.

Returns

A GRRLIB_texImg structure filled with image information.

7.1.4.48 GRRLIB_LoadTextureBMP()

```
GRRLIB_texImg * GRRLIB_LoadTextureBMP (
            const u8 * my_bmp )
```

Load a texture from a buffer.

It only works for the MS-Windows standard format uncompressed (1-bit, 4-bit, 8-bit, 24-bit and 32-bit).

Parameters

my_bmp	The Bitmap buffer to load.

Returns

A GRRLIB_texImg structure filled with image information.

7.1.4.49 GRRLIB_LoadTextureFromFile()

```
GRRLIB_texImg * GRRLIB_LoadTextureFromFile (
            const char * filename )
```

Load a texture from a file.

Parameters

filename	The JPEG, PNG or Bitmap filename to load.

Returns

A GRRLIB_texImg structure filled with image information. If an error occurs NULL will be returned.

7.1.4.50 GRRLIB_LoadTextureJPG()

```
GRRLIB_texImg * GRRLIB_LoadTextureJPG (
            const u8 * my_jpg )
```

Load a texture from a buffer.

Take care to have the JPG finish with 0xFF 0xD9!

Parameters

my_jpg	The JPEG buffer to load.

Returns

A GRRLIB_texImg structure filled with image information.

7.1.4.51 GRRLIB_LoadTextureJPGEx()

```
GRRLIB_texImg * GRRLIB_LoadTextureJPGEx (
            const u8 * my_jpg,
            const u32 my_size )
```

Load a texture from a buffer.

Author

DrTwox

Parameters

my_jpg	The JPEG buffer to load.
my_size	Size of the JPEG buffer to load.

Returns

A GRRLIB_texImg structure filled with image information.

7.1.4.52 GRRLIB_LoadTexturePNG()

```
GRRLIB_texImg * GRRLIB_LoadTexturePNG (
            const u8 * my_png )
```

Load a texture from a buffer.

Parameters

my_png	the PNG buffer to load.

Returns

A GRRLIB_texImg structure filled with image information. If image size is not correct, the texture will be completely transparent.

7.1.4.53 GRRLIB_LoadTTF()

```
GRRLIB_ttfFont * GRRLIB_LoadTTF (
            const u8 * file_base,
            s32 file_size )
```

Load a TTF from a buffer.

Parameters

file_base	Buffer with TTF data. You must not deallocate the memory before calling GRRLIB_FreeTTF.
file_size	Size of the TTF buffer.

Returns

A handle to a given TTF font object or NULL if it fails to load the font.

See also

GRRLIB_FreeTTF

7.1.4.54 GRRLIB_LoadTTFFromFile()

```
GRRLIB_ttfFont * GRRLIB_LoadTTFFromFile (
            const char * filename )
```

Load a TTF from a file.

Parameters

filename	The TTF filename to load.

Returns

A GRRLIB_ttfFont structure filled with font information. If an error occurs NULL will be returned.

7.1.4.55 GRRLIB_NGone()

```
INLINE void GRRLIB_NGone (
            const guVector v[],
            const u32 color[],
            const u16 n )
```

Draw a polygon.

Parameters

v	The vector containing the coordinates of the polygon.
color	The color of the filled polygon in RGBA format.
n	Number of points in the vector. The maximum is 65536.

7.1.4.56 GRRLIB_NGoneFilled()

```
INLINE void GRRLIB_NGoneFilled (
            const guVector v[],
```

```
            const u32 color[],
            const u16 n )
```

Draw a filled polygon.

Parameters

v	The vector containing the coordinates of the polygon.
color	The color of the filled polygon in RGBA format.
n	Number of points in the vector. The maximum is 65536.

7.1.4.57 GRRLIB_NPlot()

```
INLINE void GRRLIB_NPlot (
            const guVector v[],
            const u32 color[],
            const u16 n )
```

Draw an array of points.

Parameters

v	Array containing the points.
color	The color of the points in RGBA format.
n	Number of points in the vector array. The maximum is 65536.

7.1.4.58 GRRLIB_ObjectView()

```
void GRRLIB_ObjectView (
            f32 posx,
            f32 posy,
            f32 posz,
            f32 angx,
            f32 angy,
            f32 angz,
            f32 scalx,
            f32 scaly,
            f32 scalz )
```

Set the view matrix to draw object (in this order scale, rotate AND trans).

Parameters

posx	x position of the object.
posy	y position of the object.
posz	z position of the object.
angx	x rotation angle of the object.
angy	y rotation angle of the object.
angz	z rotation angle of the object.
scalx	x scale of the object.
scaly	y scale of the object.
scalz	z scale of the object.

7.1.4.59 GRRLIB_ObjectViewInv()

```
void GRRLIB_ObjectViewInv (
             f32 posx,
             f32 posy,
             f32 posz,
             f32 angx,
             f32 angy,
             f32 angz,
             f32 scalx,
             f32 scaly,
             f32 scalz )
```

Set the view matrix to draw object (in this order scale, trans AND rotate).

Parameters

posx	x position of the object.
posy	y position of the object.
posz	z position of the object.
angx	x rotation angle of the object.
angy	y rotation angle of the object.
angz	z rotation angle of the object.
scalx	x scale of the object.
scaly	y scale of the object.
scalz	z scale of the object.

7.1.4.60 GRRLIB_ObjectViewRotate()

```
void GRRLIB_ObjectViewRotate (
             f32 angx,
             f32 angy,
             f32 angz )
```

Rotate the object matrix to draw object.

Parameters

angx	x rotation angle of the object.
angy	y rotation angle of the object.
angz	z rotation angle of the object.

7.1.4.61 GRRLIB_ObjectViewScale()

```
void GRRLIB_ObjectViewScale (
             f32 scalx,
             f32 scaly,
             f32 scalz )
```

Scale the object matrix to draw object.

Parameters

scalx	x scale of the object.
scaly	y scale of the object.
scalz	z scale of the object.

7.1.4.62 GRRLIB_ObjectViewTrans()

```
void GRRLIB_ObjectViewTrans (
            f32 posx,
            f32 posy,
            f32 posz )
```

Translate the object matrix to draw object.

Parameters

posx	x position of the object.
posy	y position of the object.
posz	z position of the object.

7.1.4.63 GRRLIB_Plot()

```
INLINE void GRRLIB_Plot (
            const f32 x,
            const f32 y,
            const u32 color )
```

Draw a dot.

Parameters

x	Specifies the x-coordinate of the dot.
y	Specifies the y-coordinate of the dot.
color	The color of the dot in RGBA format.

Author

Jespa

7.1.4.64 GRRLIB_PrintBMF()

```
void GRRLIB_PrintBMF (
            const f32 xpos,
            const f32 ypos,
            const GRRLIB_bytemapFont * bmf,
```

```
            const char * text,
             ...  )
```

Print formatted output with a ByteMap font.

This function could be slow, it should be used with GRRLIB_CompoStart and GRRLIB_CompoEnd.

Parameters

xpos	Specifies the x-coordinate of the upper-left corner of the text.
ypos	Specifies the y-coordinate of the upper-left corner of the text.
bmf	The ByteMap font to use.
text	Text to draw.
...	Optional arguments.

7.1.4.65 GRRLIB_Printf()

```
void GRRLIB_Printf (
            const f32 xpos,
            const f32 ypos,
            const GRRLIB_texImg * tex,
            const u32 color,
            const f32 zoom,
            const char * text,
             ...  )
```

Print formatted output.

Parameters

xpos	Specifies the x-coordinate of the upper-left corner of the text.
ypos	Specifies the y-coordinate of the upper-left corner of the text.
tex	The texture containing the character set.
color	Text color in RGBA format. The alpha channel is used to change the opacity of the text.
zoom	This is a factor by which the text size will be increase or decrease.
text	Text to draw.
...	Optional arguments.

7.1.4.66 GRRLIB_PrintfTTF()

```
void GRRLIB_PrintfTTF (
            int x,
            int y,
            GRRLIB_ttfFont * myFont,
            const char * string,
            unsigned int fontSize,
            const u32 color )
```

Print function for TTF font.

Parameters

x	Specifies the x-coordinate of the upper-left corner of the text.
y	Specifies the y-coordinate of the upper-left corner of the text.
myFont	A TTF.
string	Text to draw.
fontSize	Size of the font.
color	Text color in RGBA format.

7.1.4.67 GRRLIB_PrintfTTFW()

```
void GRRLIB_PrintfTTFW (
            int x,
            int y,
            GRRLIB_ttfFont * myFont,
            const wchar_t * utf32,
            unsigned int fontSize,
            const u32 color )
```

Print function for TTF font.

Author

wplaat and DrTwox

Parameters

x	Specifies the x-coordinate of the upper-left corner of the text.
y	Specifies the y-coordinate of the upper-left corner of the text.
myFont	A TTF.
utf32	Text to draw.
fontSize	Size of the font.
color	Text color in RGBA format.

7.1.4.68 GRRLIB_PtInRect()

```
INLINE bool GRRLIB_PtInRect (
            const int hotx,
            const int hoty,
            const int hotw,
            const int hoth,
            const int wpadx,
            const int wpady )
```

Determine whether the specified point lies within the specified rectangle.

Parameters

hotx	Specifies the x-coordinate of the upper-left corner of the rectangle.

Parameters

hoty	Specifies the y-coordinate of the upper-left corner of the rectangle.
hotw	The width of the rectangle.
hoth	The height of the rectangle.
wpadx	Specifies the x-coordinate of the point.
wpady	Specifies the y-coordinate of the point.

Returns

If the specified point lies within the rectangle, the return value is `true` otherwise it's `false`.

7.1.4.69 GRRLIB_Rectangle()

```
INLINE void GRRLIB_Rectangle (
            const f32 x,
            const f32 y,
            const f32 width,
            const f32 height,
            const u32 color,
            const bool filled )
```

Draw a rectangle.

Parameters

x	Specifies the x-coordinate of the upper-left corner of the rectangle.
y	Specifies the y-coordinate of the upper-left corner of the rectangle.
width	The width of the rectangle.
height	The height of the rectangle.
color	The color of the rectangle in RGBA format.
filled	Set to `true` to fill the rectangle.

7.1.4.70 GRRLIB_RectInRect()

```
INLINE bool GRRLIB_RectInRect (
            const int rect1x,
            const int rect1y,
            const int rect1w,
            const int rect1h,
            const int rect2x,
            const int rect2y,
            const int rect2w,
            const int rect2h )
```

Determine whether a specified rectangle lies within another rectangle.

Parameters

rect1x	Specifies the x-coordinate of the upper-left corner of the rectangle.
rect1y	Specifies the y-coordinate of the upper-left corner of the rectangle.
rect1w	Specifies the width of the rectangle.
rect1h	Specifies the height of the rectangle.
rect2x	Specifies the x-coordinate of the upper-left corner of the rectangle.
rect2y	Specifies the y-coordinate of the upper-left corner of the rectangle.
rect2w	Specifies the width of the rectangle.
rect2h	Specifies the height of the rectangle.

Returns

If the specified rectangle lies within the other rectangle, the return value is `true` otherwise it's `false`.

7.1.4.71 GRRLIB_RectOnRect()

```
INLINE bool GRRLIB_RectOnRect (
             const int rect1x,
             const int rect1y,
             const int rect1w,
             const int rect1h,
             const int rect2x,
             const int rect2y,
             const int rect2w,
             const int rect2h )
```

Determine whether a part of a specified rectangle lies on another rectangle.

Parameters

rect1x	Specifies the x-coordinate of the upper-left corner of the first rectangle.
rect1y	Specifies the y-coordinate of the upper-left corner of the first rectangle.
rect1w	Specifies the width of the first rectangle.
rect1h	Specifies the height of the first rectangle.
rect2x	Specifies the x-coordinate of the upper-left corner of the second rectangle.
rect2y	Specifies the y-coordinate of the upper-left corner of the second rectangle.
rect2w	Specifies the width of the second rectangle.
rect2h	Specifies the height of the second rectangle.

Returns

If the specified rectangle lies on the other rectangle, the return value is `true` otherwise it's `false`.

7.1.4.72 GRRLIB_Screen2Texture()

```
void GRRLIB_Screen2Texture (
             u16 posx,
```

```
            u16 posy,
            GRRLIB_texImg * tex,
            bool clear )
```

Make a snapshot of the screen in a texture WITHOUT ALPHA LAYER.

Parameters

posx	Top left corner of the grabbed part.
posy	Top left corner of the grabbed part.
tex	A pointer to a texture representing the screen.
clear	When this flag is set to `true`, the screen is cleared after copy.

7.1.4.73 GRRLIB_ScrShot()

```
bool GRRLIB_ScrShot (
            const char * filename )
```

Make a PNG screenshot.

It should be called after drawing stuff on the screen, but before GRRLIB_Render(). libfat is required to use the function.

Parameters

filename	Name of the file to write.

Returns

Returns `true` if everything worked, `false` if problems occurred.

7.1.4.74 GRRLIB_SetAntiAliasing()

```
INLINE void GRRLIB_SetAntiAliasing (
            const bool aa )
```

Turn anti-aliasing on/off.

Parameters

aa	Set to `true` to enable anti-aliasing (Default: Enabled).

7.1.4.75 GRRLIB_SetBackgroundColour()

```
void GRRLIB_SetBackgroundColour (
            u8 r,
            u8 g,
```

```
            u8 b,
            u8 a )
```

Set the background parameter when screen is cleared.

Parameters

r	Red component.
g	Green component.
b	Blue component.
a	Alpha component.

7.1.4.76 GRRLIB_SetBlend()

```
INLINE void GRRLIB_SetBlend (
            const GRRLIB_blendMode blendmode )
```

Set a blending mode.

Parameters

blendmode	The blending mode to use (Default: GRRLIB_BLEND_ALPHA).

7.1.4.77 GRRLIB_SetHandle()

```
INLINE void GRRLIB_SetHandle (
            GRRLIB_texImg * tex,
            const int x,
            const int y )
```

Set a texture's X and Y handles.

For example, it could be used for the rotation of a texture.

Parameters

tex	The texture to set the handle on.
x	The x-coordinate of the handle.
y	The y-coordinate of the handle.

7.1.4.78 GRRLIB_SetLightAmbient()

```
void GRRLIB_SetLightAmbient (
            u32 ambientcolor )
```

Set ambient color.

When no diffuse light is shining on a object, the color is equal to ambient color.

Parameters

ambientcolor	Ambient color in RGBA format.

7.1.4.79 GRRLIB_SetLightDiff()

```
void GRRLIB_SetLightDiff (
            u8 num,
            guVector pos,
            f32 distattn,
            f32 brightness,
            u32 lightcolor )
```

Set diffuse light parameters.

Parameters

num	Number of the light. It's a number from 0 to 7.
pos	Position of the diffuse light (x/y/z).
distattn	Distance attenuation.
brightness	Brightness of the light. The value should be between 0 and 1.
lightcolor	Color of the light in RGBA format.

7.1.4.80 GRRLIB_SetLightSpec()

```
void GRRLIB_SetLightSpec (
            u8 num,
            guVector dir,
            f32 shininess,
            u32 lightcolor,
            u32 speccolor )
```

Set specular light parameters.

Parameters

num	Number of the light. It's a number from 0 to 7.
dir	Direction of the specular ray (x/y/z).
shininess	Shininess of the specular. (between 4 and 254)
lightcolor	Color of the light in RGBA format.
speccolor	Specular color in RGBA format.

7.1.4.81 GRRLIB_SetLightSpot()

```
void GRRLIB_SetLightSpot (
            u8 num,
            guVector pos,
```

```
            guVector lookat,
            f32 angAttn0,
            f32 angAttn1,
            f32 angAttn2,
            f32 distAttn0,
            f32 distAttn1,
            f32 distAttn2,
            u32 lightcolor )
```

Set Spot light parameters.

Parameters

num	Number of the light. It's a number from 0 to 7.
pos	Position of the spot light (x/y/z).
lookat	Where spot light look at (x/y/z).
angAttn0	cone attenuation factor 0.
angAttn1	cone attenuation factor 1.
angAttn2	cone attenuation factor 2.
distAttn0	Distance attenuation factor 0.
distAttn1	Distance attenuation factor 1.
distAttn2	Distance attenuation factor 2.
lightcolor	Color of the light in RGBA format.

7.1.4.82 GRRLIB_SetMidHandle()

```
INLINE void GRRLIB_SetMidHandle (
            GRRLIB_texImg * tex,
            const bool enabled )
```

Center a texture's handles.

For example, it could be used for the rotation of a texture.

Parameters

tex	The texture to center.
enabled	

7.1.4.83 GRRLIB_SetPixelToFB()

```
INLINE void GRRLIB_SetPixelToFB (
            int x,
            int y,
            u32 pokeColor )
```

Writes a pixel directly from the FrontBuffer.

Parameters

x	The x-coordinate within the FB.

Parameters

y	The y-coordinate within the FB.
pokeColor	The color of the pixel in RGBA format.

7.1.4.84 GRRLIB_SetPixelTotexImg()

```
INLINE void GRRLIB_SetPixelTotexImg (
              const int x,
              const int y,
              GRRLIB_texImg * tex,
              const u32 color )
```

Set the color value of a pixel to a GRRLIB_texImg.

See also

GRRLIB_FlushTex

Parameters

x	Specifies the x-coordinate of the pixel in the texture.
y	Specifies the y-coordinate of the pixel in the texture.
tex	The texture to set the color to.
color	The color of the pixel in RGBA format.

7.1.4.85 GRRLIB_SetTexture()

```
void GRRLIB_SetTexture (
              GRRLIB_texImg * tex,
              bool rep )
```

Set the texture to an object (contributed by chris_c aka DaShAmAn).

Parameters

tex	Pointer to an image texture (GRRLIB_texImg format).
rep	Texture Repeat Mode, `true` will repeat it, `false` won't.

7.1.4.86 GRRLIB_WidthTTF()

```
u32 GRRLIB_WidthTTF (
              GRRLIB_ttfFont * myFont,
              const char * string,
              unsigned int fontSize )
```

Get the width of a text in pixel.

Parameters

myFont	A TTF.
string	The text to check.
fontSize	The size of the font.

Returns

The width of a text in pixel.

7.1.4.87 GRRLIB_WidthTTFW()

```
u32 GRRLIB_WidthTTFW (
            GRRLIB_ttfFont * myFont,
            const wchar_t * utf32,
            unsigned int fontSize )
```

Get the width of a text in pixel.

Parameters

myFont	A TTF.
utf32	The text to check.
fontSize	The size of the font.

Returns

The width of a text in pixel.

Chapter 8

Data Structure Documentation

8.1 GRRLIB_bytemapChar Struct Reference

Structure to hold the bytemap character information.

`#include <grrlib.h>`

Data Fields

- u8 width
  Character width.
- u8 height
  Character height.
- s8 relx
  Horizontal offset relative to cursor (-128 to 127).
- s8 rely
  Vertical offset relative to cursor (-128 to 127).
- u8 **kerning**
  Kerning (Horizontal cursor shift after drawing the character).
- u8 * data
  Character data (uncompressed, 8 bits per pixel).

8.1.1 Detailed Description

Structure to hold the bytemap character information.

8.1.2 Field Documentation

8.1.2.1 data

`u8* data`

Character data (uncompressed, 8 bits per pixel).

8.1.2.2 height

```
u8 height
```

Character height.

8.1.2.3 relx

```
s8 relx
```

Horizontal offset relative to cursor (-128 to 127).

8.1.2.4 rely

```
s8 rely
```

Vertical offset relative to cursor (-128 to 127).

8.1.2.5 width

```
u8 width
```

Character width.

The documentation for this struct was generated from the following file:

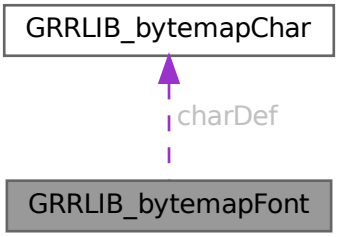
- grrlib.h

8.2 GRRLIB_bytemapFont Struct Reference

Structure to hold the bytemap font information.

`#include <grrlib.h>`

Collaboration diagram for GRRLIB_bytemapFont:



Data Fields

- char * name
  Font name.
- u32 * palette
  Font palette.
- u16 nbChar
  Number of characters in font.
- u8 version
  Version.
- s8 **tracking**
  Tracking (Add-space after each char) (-128 to 127).
- GRRLIB_bytemapChar **charDef** [256]
  Array of bitmap characters.

8.2.1 Detailed Description

Structure to hold the bytemap font information.

8.2.2 Field Documentation

8.2.2.1 name

`char* name`

Font name.

8.2.2.2 nbChar

```
u16 nbChar
```

Number of characters in font.

8.2.2.3 palette

```
u32* palette
```

Font palette.

8.2.2.4 version

```
u8 version
```

Version.

The documentation for this struct was generated from the following file:

- grrlib.h

8.3 GRRLIB_drawSettings Struct Reference

Structure to hold the current drawing settings.

```
#include <grrlib.h>
```

Data Fields

- bool **antialias**

  AntiAlias is enabled when set to true.
- GRRLIB_blendMode blend

  Blending Mode.
- int lights

  Active lights.

8.3.1 Detailed Description

Structure to hold the current drawing settings.

8.3.2 Field Documentation

8.3.2.1 blend

```
GRRLIB_blendMode blend
```

Blending Mode.

8.3.2.2 lights

```
int lights
```

Active lights.

The documentation for this struct was generated from the following file:

- grrlib.h

8.4 GRRLIB_Font Struct Reference

Structure to hold the TTF information.

```
#include <grrlib.h>
```

Data Fields

- void * **face**
  A TTF face object.
- bool **kerning**
  true whenever a face object contains kerning data that can be accessed with FT_Get_Kerning.

8.4.1 Detailed Description

Structure to hold the TTF information.

The documentation for this struct was generated from the following file:

- grrlib.h

8.5 GRRLIB_texImg Struct Reference

Structure to hold the texture information.

`#include <grrlib.h>`

Data Fields

- u32 w
  The width of the texture in pixels.
- u32 **h**
  The height of the texture in pixels.
- int **handlex**
  Texture handle x.
- int **handley**
  Texture handle y.
- int **offsetx**
  Texture offset x.
- int **offsety**
  Texture offset y.
- bool tiledtex
  Texture is tiled if set to true.
- u32 tilew
  The width of one tile in pixels.
- u32 tileh
  The height of one tile in pixels.
- u32 nbtilew
  Number of tiles for the x axis.
- u32 nbtileh
  Number of tiles for the y axis.
- u32 tilestart
  Offset to tile starting position.
- f32 **ofnormaltexx**
  Offset of normalized texture on x.
- f32 **ofnormaltexy**
  Offset of normalized texture on y.
- void * **data**
  Pointer to the texture data.

8.5.1 Detailed Description

Structure to hold the texture information.

8.5.2 Field Documentation

8.5.2.1 nbtileh

`u32 nbtileh`

Number of tiles for the y axis.

8.5.2.2 nbtilew

```
u32 nbtilew
```

Number of tiles for the x axis.

8.5.2.3 tiledtex

```
bool tiledtex
```

Texture is tiled if set to true.

8.5.2.4 tileh

```
u32 tileh
```

The height of one tile in pixels.

8.5.2.5 tilestart

```
u32 tilestart
```

Offset to tile starting position.

8.5.2.6 tilew

```
u32 tilew
```

The width of one tile in pixels.

8.5.2.7 w

```
u32 w
```

The width of the texture in pixels.

The documentation for this struct was generated from the following file:

- grrlib.h

Chapter 9

File Documentation

9.1 grrlib.h File Reference

GRRLIB user include file.

```
#include <gccore.h>
#include "grrlib/GRRLIB__lib.h"
#include "grrlib/GRRLIB__inline.h"
```

Data Structures

- struct GRRLIB_drawSettings

  Structure to hold the current drawing settings.
- struct GRRLIB_texImg

  Structure to hold the texture information.
- struct GRRLIB_bytemapChar

  Structure to hold the bytemap character information.
- struct GRRLIB_bytemapFont

  Structure to hold the bytemap font information.
- struct GRRLIB_Font

  Structure to hold the TTF information.

Macros

- #define **GRRLIB_VER_STRING** "4.4.1"

  Version information for GRRLIB.
- #define **R**(c) (((c) >>24) &0xFF)

  Extract red component of colour.
- #define **G**(c) (((c) >>16) &0xFF)

  Extract green component of colour.
- #define **B**(c) (((c) >> 8) &0xFF)

  Extract blue component of colour.
- #define **A**(c) ((c) &0xFF)

  Extract alpha component of colour.

- #define RGBA(r, g, b, a)

  Build an RGB pixel from components.
- #define **GRRLIB_BLEND_NONE** (GRRLIB_BLEND_ALPHA)

  Alias for GRRLIB_BLEND_ALPHA.
- #define **GRRLIB_BLEND_LIGHT** (GRRLIB_BLEND_ADD)

  Alias for GRRLIB_BLEND_ADD.
- #define **GRRLIB_BLEND_SHADE** (GRRLIB_BLEND_MULTI)

  Alias for GRRLIB_BLEND_MULTI.
- #define **GRR_EXTERN** extern
- #define **GRR_INIT**(v)
- #define **GRR_INITS**(...)
- #define **INLINE** static inline

Typedefs

- typedef enum GRRLIB_blendMode **GRRLIB_blendMode**

  GRRLIB Blending Modes.
- typedef struct GRRLIB_drawSettings **GRRLIB_drawSettings**

  Structure to hold the current drawing settings.
- typedef struct GRRLIB_texImg **GRRLIB_texImg**

  Structure to hold the texture information.
- typedef struct GRRLIB_bytemapChar **GRRLIB_bytemapChar**

  Structure to hold the bytemap character information.
- typedef struct GRRLIB_bytemapFont **GRRLIB_bytemapFont**

  Structure to hold the bytemap font information.
- typedef struct GRRLIB_Font **GRRLIB_ttfFont**

  Structure to hold the TTF information.

Enumerations

- enum GRRLIB_blendMode {
  GRRLIB_BLEND_ALPHA = 0 , GRRLIB_BLEND_ADD = 1 , GRRLIB_BLEND_SCREEN = 2 ,
  GRRLIB_BLEND_MULTI = 3 ,
  GRRLIB_BLEND_INV = 4 }

  GRRLIB Blending Modes.

Functions

- GRR_EXTERN void ∗xfb[2] **GRR_INITS** (NULL, NULL)
- GRR_EXTERN u32 fb **GRR_INIT** (0)

Variables

- GRR_EXTERN GXRModeObj ∗ **rmode**

  Video mode.

9.1.1 Detailed Description

GRRLIB user include file.

9.2 GRRLIB__inline.h File Reference

GRRLIB inline function prototypes.

```
#include <grrlib/GRRLIB_clipping.h>
#include <grrlib/GRRLIB_collision.h>
#include <grrlib/GRRLIB_fbComplex.h>
#include <grrlib/GRRLIB_fbGX.h>
#include <grrlib/GRRLIB_fbSimple.h>
#include <grrlib/GRRLIB_handle.h>
#include <grrlib/GRRLIB_pixel.h>
#include <grrlib/GRRLIB_settings.h>
#include <grrlib/GRRLIB_texSetup.h>
```

Functions

- INLINE void **GRRLIB_ClipReset** (void)

  Reset the clipping to normal.
- INLINE void GRRLIB_ClipDrawing (const u32 x, const u32 y, const u32 width, const u32 height)

  Clip the drawing area to an rectangle.
- INLINE bool GRRLIB_PtInRect (const int hotx, const int hoty, const int hotw, const int hoth, const int wpadx, const int wpady)

  Determine whether the specified point lies within the specified rectangle.
- INLINE bool GRRLIB_RectInRect (const int rect1x, const int rect1y, const int rect1w, const int rect1h, const int rect2x, const int rect2y, const int rect2w, const int rect2h)

  Determine whether a specified rectangle lies within another rectangle.
- INLINE bool GRRLIB_RectOnRect (const int rect1x, const int rect1y, const int rect1w, const int rect1h, const int rect2x, const int rect2y, const int rect2w, const int rect2h)

  Determine whether a part of a specified rectangle lies on another rectangle.
- INLINE void GRRLIB_NPlot (const guVector v[ ], const u32 color[ ], const u16 n)

  Draw an array of points.
- INLINE void GRRLIB_NGone (const guVector v[ ], const u32 color[ ], const u16 n)

  Draw a polygon.
- INLINE void GRRLIB_NGoneFilled (const guVector v[ ], const u32 color[ ], const u16 n)

  Draw a filled polygon.
- INLINE void GRRLIB_GXEngine (const guVector v[ ], const u32 color[ ], const u16 n, const u8 fmt)

  Draws a vector.
- INLINE void GRRLIB_FillScreen (const u32 color)

  Clear screen with a specific color.
- INLINE void GRRLIB_Plot (const f32 x, const f32 y, const u32 color)

  Draw a dot.
- INLINE void GRRLIB_Line (const f32 x1, const f32 y1, const f32 x2, const f32 y2, const u32 color)

  Draw a line.
- INLINE void GRRLIB_Rectangle (const f32 x, const f32 y, const f32 width, const f32 height, const u32 color, const bool filled)

  Draw a rectangle.
- INLINE void GRRLIB_SetHandle (GRRLIB_texImg *tex, const int x, const int y)

  Set a texture's X and Y handles.
- INLINE void GRRLIB_SetMidHandle (GRRLIB_texImg *tex, const bool enabled)

  Center a texture's handles.
- INLINE u32 GRRLIB_GetPixelFromtexImg (const int x, const int y, const GRRLIB_texImg *tex)

Return the color value of a pixel from a GRRLIB_texImg.

- INLINE void GRRLIB_SetPixelTotexImg (const int x, const int y, GRRLIB_texImg *tex, const u32 color)

  Set the color value of a pixel to a GRRLIB_texImg.
- INLINE u32 GRRLIB_GetPixelFromFB (int x, int y)

  Reads a pixel directly from the FrontBuffer.
- INLINE void GRRLIB_SetPixelToFB (int x, int y, u32 pokeColor)

  Writes a pixel directly from the FrontBuffer.
- INLINE void GRRLIB_SetBlend (const GRRLIB_blendMode blendmode)

  Set a blending mode.
- INLINE GRRLIB_blendMode GRRLIB_GetBlend (void)

  Get the current blending mode.
- INLINE void GRRLIB_SetAntiAliasing (const bool aa)

  Turn anti-aliasing on/off.
- INLINE bool GRRLIB_GetAntiAliasing (void)

  Get current anti-aliasing setting.
- INLINE void GRRLIB_ClearTex (GRRLIB_texImg *tex)

  Clear a texture to transparent black.
- INLINE void GRRLIB_FlushTex (GRRLIB_texImg *tex)

  Write the contents of a texture in the data cache down to main memory.
- INLINE void GRRLIB_FreeTexture (GRRLIB_texImg *tex)

  Free memory allocated for texture.

9.2.1 Detailed Description

GRRLIB inline function prototypes.

Do not include GRRLIB__inline.h directly, include only GRRLIB.h.

9.3 GRRLIB__lib.h File Reference

GRRLIB library function prototypes.

Functions

- GRRLIB_bytemapFont * GRRLIB_LoadBMF (const u8 my_bmf[])

  Load a ByteMap font structure from a buffer.
- void GRRLIB_FreeBMF (GRRLIB_bytemapFont *bmf)

  Free memory allocated by ByteMap fonts.
- void GRRLIB_InitTileSet (GRRLIB_texImg *tex, const u32 tilew, const u32 tileh, const u32 tilestart)

  Initialize a tile set.
- void GRRLIB_BMFX_FlipH (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest)

  Flip texture horizontal.
- void GRRLIB_BMFX_FlipV (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest)

  Flip texture vertical.
- void GRRLIB_BMFX_Grayscale (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest)

  Change a texture to gray scale.
- void GRRLIB_BMFX_Sepia (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest)

Change a texture to sepia (old photo style).

- void GRRLIB_BMFX_Invert (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest)

  Invert colors of the texture.
- void GRRLIB_BMFX_Blur (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest, const u32 factor)

  A texture effect (Blur).
- void GRRLIB_BMFX_Scatter (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest, const u32 factor)

  A texture effect (Scatter).
- void GRRLIB_BMFX_Pixelate (const GRRLIB_texImg *texsrc, GRRLIB_texImg *texdest, const u32 factor)

  A texture effect (Pixelate).
- int GRRLIB_Init (void)

  Initialize GRRLIB.
- void GRRLIB_Exit (void)

  Call this before exiting your application.
- void GRRLIB_Ellipse (const f32 x, const f32 y, const f32 radiusX, const f32 radiusY, const u32 color, const u8 filled)

  Draw an ellipse.
- void GRRLIB_Circle (const f32 x, const f32 y, const f32 radius, const u32 color, const u8 filled)

  Draw a circle.
- int GRRLIB_LoadFile (const char *filename, u8 **data)

  Load a file to memory.
- GRRLIB_texImg * GRRLIB_LoadTextureFromFile (const char *filename)

  Load a texture from a file.
- GRRLIB_ttfFont * GRRLIB_LoadTTFFromFile (const char *filename)

  Load a TTF from a file.
- bool GRRLIB_ScrShot (const char *filename)

  Make a PNG screenshot.
- void GRRLIB_Printf (const f32 xpos, const f32 ypos, const GRRLIB_texImg *tex, const u32 color, const f32 zoom, const char *text,...)

  Print formatted output.
- void GRRLIB_PrintBMF (const f32 xpos, const f32 ypos, const GRRLIB_bytemapFont *bmf, const char *text,...)

  Print formatted output with a ByteMap font.
- void GRRLIB_DrawImg (const f32 xpos, const f32 ypos, const GRRLIB_texImg *tex, const f32 degrees, const f32 scaleX, const f32 scaleY, const u32 color)

  Draw a texture.
- void GRRLIB_DrawImgQuad (const guVector pos[4], GRRLIB_texImg *tex, const u32 color)

  Draw a textured quad.
- void GRRLIB_DrawTile (const f32 xpos, const f32 ypos, const GRRLIB_texImg *tex, const f32 degrees, const f32 scaleX, const f32 scaleY, const u32 color, const int frame)

  Draw a tile.
- void GRRLIB_DrawPart (const f32 xpos, const f32 ypos, const f32 partx, const f32 party, const f32 partw, const f32 parth, const GRRLIB_texImg *tex, const f32 degrees, const f32 scaleX, const f32 scaleY, const u32 color)

  Draw a part of a texture.
- void GRRLIB_DrawTileQuad (const guVector pos[4], GRRLIB_texImg *tex, const u32 color, const int frame)

  Draw a tile in a quad.
- void **GRRLIB_Render** (void)

  Call this function after drawing.
- void GRRLIB_Screen2Texture (u16 posx, u16 posy, GRRLIB_texImg *tex, bool clear)

  Make a snapshot of the screen in a texture WITHOUT ALPHA LAYER.
- void GRRLIB_CompoStart (void)

  Start GX compositing process.

- void GRRLIB_CompoEnd (u16 posx, u16 posy, GRRLIB_texImg *tex)

  End GX compositing process (Make a snapshot of the screen in a texture WITH ALPHA LAYER).
- GRRLIB_texImg * GRRLIB_CreateEmptyTexture (const u32 width, const u32 height)

  Create an empty texture.
- GRRLIB_texImg * GRRLIB_LoadTexture (const u8 *my_img)

  Load a texture from a buffer.
- GRRLIB_texImg * GRRLIB_LoadTexturePNG (const u8 *my_png)

  Load a texture from a buffer.
- GRRLIB_texImg * GRRLIB_LoadTextureJPG (const u8 *my_jpg)

  Load a texture from a buffer.
- GRRLIB_texImg * GRRLIB_LoadTextureJPGEx (const u8 *my_jpg, const u32 my_size)

  Load a texture from a buffer.
- GRRLIB_texImg * GRRLIB_LoadTextureBMP (const u8 *my_bmp)

  Load a texture from a buffer.
- bool GRRLIB_GeckoInit (void)

  Initialize USB Gecko.
- void GRRLIB_GeckoPrintf (const char *text,...)

  Print Gecko.
- void GRRLIB_SetBackgroundColour (u8 r, u8 g, u8 b, u8 a)

  Set the background parameter when screen is cleared.
- void GRRLIB_Camera3dSettings (f32 posx, f32 posy, f32 posz, f32 upx, f32 upy, f32 upz, f32 lookx, f32 looky, f32 lookz)

  Set the camera parameter (contributed by chris_c aka DaShAmAn).
- void GRRLIB_3dMode (f32 minDist, f32 maxDist, f32 fov, bool texturemode, bool normalmode)

  Set up the position matrix (contributed by chris_c aka DaShAmAn).
- void **GRRLIB_2dMode** (void)

  Go back to 2D mode (contributed by chris_c aka DaShAmAn).
- void **GRRLIB_ObjectViewBegin** (void)

  Init the object matrix to draw object.
- void GRRLIB_ObjectViewScale (f32 scalx, f32 scaly, f32 scalz)

  Scale the object matrix to draw object.
- void GRRLIB_ObjectViewRotate (f32 angx, f32 angy, f32 angz)

  Rotate the object matrix to draw object.
- void GRRLIB_ObjectViewTrans (f32 posx, f32 posy, f32 posz)

  Translate the object matrix to draw object.
- void **GRRLIB_ObjectViewEnd** (void)

  Concat the object and the view matrix and calculate the inverse normal matrix.
- void GRRLIB_ObjectView (f32 posx, f32 posy, f32 posz, f32 angx, f32 angy, f32 angz, f32 scalx, f32 scaly, f32 scalz)

  Set the view matrix to draw object (in this order scale, rotate AND trans).
- void GRRLIB_ObjectViewInv (f32 posx, f32 posy, f32 posz, f32 angx, f32 angy, f32 angz, f32 scalx, f32 scaly, f32 scalz)

  Set the view matrix to draw object (in this order scale, trans AND rotate).
- void GRRLIB_SetTexture (GRRLIB_texImg *tex, bool rep)

  Set the texture to an object (contributed by chris_c aka DaShAmAn).
- void GRRLIB_DrawTorus (f32 r, f32 R, int nsides, int rings, bool filled, u32 col)

  Draw a torus (with normal).
- void GRRLIB_DrawSphere (f32 r, int lats, int longs, bool filled, u32 col)

  Draw a sphere (with normal).
- void GRRLIB_DrawCube (f32 size, bool filled, u32 col)

  Draw a cube (with normal).

- void GRRLIB_DrawCylinder (f32 r, f32 h, u16 d, bool filled, u32 col)

  Draw a cylinder (with normal).
- void GRRLIB_DrawCone (f32 r, f32 h, u16 d, bool filled, u32 col)

  Draw a cone (with normal).
- void GRRLIB_DrawTessPanel (f32 w, f32 wstep, f32 h, f32 hstep, bool filled, u32 col)

  Draw a Tesselated panel (with normal).
- void GRRLIB_SetLightAmbient (u32 ambientcolor)

  Set ambient color.
- void GRRLIB_SetLightDiff (u8 num, guVector pos, f32 distattn, f32 brightness, u32 lightcolor)

  Set diffuse light parameters.
- void GRRLIB_SetLightSpec (u8 num, guVector dir, f32 shininess, u32 lightcolor, u32 speccolor)

  Set specular light parameters.
- void GRRLIB_SetLightSpot (u8 num, guVector pos, guVector lookat, f32 angAttn0, f32 angAttn1, f32 angAttn2, f32 distAttn0, f32 distAttn1, f32 distAttn2, u32 lightcolor)

  Set Spot light parameters.
- void **GRRLIB_SetLightOff** (void)

  Set all lights off, like at init.
- GRRLIB_ttfFont * GRRLIB_LoadTTF (const u8 *file_base, s32 file_size)

  Load a TTF from a buffer.
- void GRRLIB_FreeTTF (GRRLIB_ttfFont *myFont)

  Free memory allocated by TTF fonts.
- void GRRLIB_PrintfTTF (int x, int y, GRRLIB_ttfFont *myFont, const char *string, unsigned int fontSize, const u32 color)

  Print function for TTF font.
- void GRRLIB_PrintfTTFW (int x, int y, GRRLIB_ttfFont *myFont, const wchar_t *string, unsigned int fontSize, const u32 color)

  Print function for TTF font.
- u32 GRRLIB_WidthTTF (GRRLIB_ttfFont *myFont, const char *, unsigned int)

  Get the width of a text in pixel.
- u32 GRRLIB_WidthTTFW (GRRLIB_ttfFont *myFont, const wchar_t *, unsigned int)

  Get the width of a text in pixel.

9.3.1 Detailed Description

GRRLIB library function prototypes.

Do not include GRRLIB__lib.h directly, include only GRRLIB.h.

Chapter 10

Examples

10.1 template/source/main.c

This example shows the minimum code required to use GRRLIB.

This example shows the minimum code required to use GRRLIB.It could be used as a template to start a new project. More elaborate examples can be found inside the *examples* folder.

```
/*===============================================
        GRRLIB (GX Version)
        - Template Code -

        Minimum Code To Use GRRLIB
===============================================*/
#include <grrlib.h>

#include <stdlib.h>
#include <wiiuse/wpad.h>

int main(int argc, char **argv) {
    // Initialise the Graphics & Video subsystem
    GRRLIB_Init();

    // Initialise the Wiimotes
    WPAD_Init();

    // Loop forever
    while(1) {

        WPAD_ScanPads();  // Scan the Wiimotes

        // If [HOME] was pressed on the first Wiimote, break out of the loop
        if (WPAD_ButtonsDown(0) & WPAD_BUTTON_HOME)  break;

        // ---------------------------------------------------------------------
        // Place your drawing code here
        // ---------------------------------------------------------------------

        GRRLIB_Render();  // Render the frame buffer to the TV
    }

    GRRLIB_Exit(); // Be a good boy, clear the memory allocated by GRRLIB

    exit(0);  // Use exit() to exit a program, do not use 'return' from main()
}
```

Index