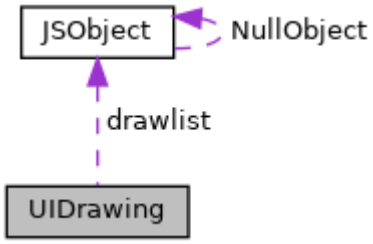
UIDrawing Class Reference

`#include <gui.h>`

Collaboration diagram for UIDrawing:



[legend]

Public Types

enum	{ FNT_NORMAL =0x0, FNT_ITALIC =0x1, FNT_BOLD =0x700 }
	ImageType { IMG_BMP, IMG_GIF, IMG_JPEG, IMG_PAM,
enum	 IMG_PBM, IMG_PNG, IMG_RAW }

Public Member Functions

	UIDrawing ()
	~UIDrawing ()
void	reset ()
void	clear (unsigned rgb)
void	color (unsigned rgb)
void	linewidth (int w)
void	line (int x, int y, int x2, int y2)
void	pixel (int x, int y)
void	rect (int x, int y, int w, int h)
void	rectf (int x, int y, int w, int h)
void	trif (int x, int y, int x2, int y2, int x3, int y3)
void	font (const char *name, int size, unsigned style=FNT_NORMAL)
void	font (const std::string &name, int size, unsigned style=FNT_NORMAL)
void	text (const char *text, int x, int y)
void	text (const std::string &text, int x, int y)
void	image (ImageType type, const void *data, unsigned size, int x, int y, int w=0, int h=0)
void	image (const char *filename, int x, int y)
void	image (const std::string &filename, int x, int y)

Data Fields

vfihtml::JSObject drawlist

Detailed Description

class for creating a drawing list

Member Enumeration Documentation

? anonymous enum

anonymous enum

font style, values can be combined by or-ing them together

Enumerator

FNT_NORMAL	normal font
FNT_ITALIC	italic style
FNT_BOLD	bold style

? ImageType

enum ImageType

image type

Enumerator

IMG_BMP	BMP image
IMG_GIF	GIF image
IMG_JPEG	JPEG image
IMG_PAM	PAM image
IMG_PBM	PBM image
IMG_PNG	PNG image
IMG_RAW	raw image data, the format is inferred from the size information of the data.

Constructor & Destructor Documentation

? UIDrawing()

UIDrawing ( )

constructor

? ~UIDrawing()

~UIDrawing ( )

destructor

Member Function Documentation

? clear()

void clear (unsigned *rgb*)

clear canvas

Parameters

[in] rgb 24-bit color value to be used for clearing the canvas

? color()

void color (unsigned *rgb*)

set current color for successive drawing commands

Parameters

[in] rgb 24-bit color value

? font() [1/2]

```
void font ( const char *  name,
            int           size,
            unsigned      style = FNT_NORMAL
          )
```

set the current font

Parameters

[in] name font name

[in] size nominal font height

[in] style font style

? font() [2/2]

```
void font ( const std::string &  name,
            int                  size,
            unsigned             style = FNT_NORMAL
          )
```

set the current font

Parameters

[in] name font name
[in] size nominal font height
[in] style font style

? image() [1/3]

```
void image ( const char *  filename,
             int           x,
             int           y
           )
```

draw image data to screen

Parameters

[in] filename image file name
[in] x x-coordinate
[in] y y-coordinate

? image() [2/3]

```
void image ( const std::string &  filename,
             int                  x,
             int                  y
           )
```

draw image data to screen

Parameters

[in] filename image file name

[in] x x-coordinate
[in] y y-coordinate

? image() [3/3]

```
void image ( ImageType  type,
             const void *  data,
             unsigned  size,
             int  x,
             int  y,
             int  w = 0,
             int  h = 0
           )
```

draw image data to screen

Parameters
[in] type image type
[in] data image data (format depends on type)
[in] size number of bytes in data
[in] x x-coordinate
[in] y y-coordinate
[in] w width of the image, only used for IMG_RAW
[in] h height of the image, only used for IMG_RAW

In case of IMG_RAW the number of bytes per pixel is determined from size and w*h. Depending on the number of bytes per pixel the image format is considered to be as follows:

- 1: Grayscale image
- 2: Grayscale image with alpha
- 3: RGB image
- 4: RGB image with alpha

? line()

```
void line ( int  x,
            int  y,
            int  x2,
            int  y2
          )
```

draw line between the given points

Parameters

[in] x x-coodinate first point
[in] y y-coodinate first point
[in] x2 x-coodinate second point
[in] y2 y-coodinate second point

? linewidth()

void linewidth (int *w*)

set line width for line and rect

Parameters

[in] w line width in pixels, 0 is default

? pixel()

void pixel (int *x*,
int *y*
)

draw single pixel pixel

Parameters

[in] x x-coodinate
[in] y y-coodinate

? rect()

void rect (int *x*,
int *y*,
int *w*,
int *h*
)

draw rectangle

Parameters

[in] x x-coodinate
[in] y x-coodinate
[in] w width
[in] h height

? rectf()

```
void rectf ( int x,
             int y,
             int w,
             int h
           )
```

draw filled rectangle

Parameters

[in] x x-coodinate
[in] y x-coodinate
[in] w width
[in] h height

? reset()

```
void reset ( )
```

reset drawing

? text() [1/2]

```
void text ( const char * text,
            int          x,
            int          y
          )
```

draw text using current font and color

Parameters

[in] text text
[in] x x-coodinate first point
[in] y y-coodinate first point

Coordinates refer to the starting point of the base line, e.g. drawing an 'A', x/y refers to the leftmost bottom pixel of 'A'.

? text() [2/2]

void text (const std::string & *text*,
int *x*,
int *y*
)

draw text using current font and color

Parameters
[in] text text
[in] x x-coodinate first point
[in] y y-coodinate first point

Coordinates refer to the starting point of the base line, e.g. drawing an 'A', x/y refers to the leftmost bottom pixel of 'A'.

? trif()

void trif (int *x*,
int *y*,
int *x2*,
int *y2*,
int *x3*,
int *y3*
)

draw filled triangle given by three points

Parameters

[in] x x-coodinate first point
[in] y y-coodinate first point
[in] x2 x-coodinate second point
[in] y2 y-coodinate second point
[in] x3 x-coodinate third point
[in] y3 y-coodinate third point

Field Documentation

? drawlist

vfihtml::JSObject drawlist

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

- guiprt/src/html/gui.h