PNGwriter Quick Reference Manual

Version 0.3.7 (12/ X / 2003)

http://pngwriter.sourceforge.net/

Introduction

This is the PNGwriter Quick Reference Manual. It is a summary of the functions that PNGwriter provides. For a more detailed description, see the pngwriter.h header file.

General Notes

*It is important to remember that all functions that accept an argument of type "const char *" will also accept "char *".*

It is also important to remember that whenever a function has a colour coeffiecient as its argument, that argument can be either an int from 0 to 65535 or a double from 0.0 to 1.0.

Constructor

The constructor requires the width and the height of the image, the background colour for the image and the filename of the file (a pointer or simply "myfile.png").

```
pngwriter();
pngwriter(const pngwriter &rhs);
pngwriter(int width, int height, int backgroundcolour, char * filename);
pngwriter(int width, int height, double backgroundcolour, char * filename);
pngwriter(int width, int height, int backgroundcolour, const char *
filename);
pngwriter(int width, int height, double backgroundcolour, const char *
filename);
```

Assignment Operator

PNGwriter overloads the assignment operator =.

```
pngwriter & operator = (const pngwriter & rhs);
```

Plot

The pixels are numbered starting from (1, 1) and go to (width, height). If the colour coefficients are of type int, they go from 0 to 65535. If they are of type double, they go from 0.0 to 1.0.

```
void plot(int x, int y, int red, int green, int blue);
void plot(int x, int y, double red, double green, double blue);
```

Plot HSV

With this function a pixel at coordinates (x, y) can be set to the desired colour, but with the colour coefficients given in the Hue, Saturation, Value colourspace.

```
void plotHSV(int x, int y, double hue, double saturation, double value);
void plotHSV(int x, int y, int hue, int saturation, int value);
```

Read

With this function we find out what colour the pixel (x, y) is. If "colour" is 1, it will return the red coefficient, if it is set to 2, the green one, and if set to 3, the blue colour coefficient will be returned, and this returned value will be of type int and be between 0 and 65535.

```
int read(int x, int y, int colour);
```

Read, Average

Same as the above, only that the average of the three colour coefficients is returned.

```
int read(int x, int y);
```

dRead

Same as Read, but the returned value will be of type double and be between 0.0 and 1.0.

```
double dread(int x, int y, int colour);
```

dRead, Average

Same as the above, only that the average of the three colour coefficients is returned.

```
double dread(int x, int y);
```

Read HSV

With this function we find out what colour the pixel (x, y) is, but in the Hue, Saturation, Value colourspace. If "colour" is 1, it will return the Hue coefficient, if it is set to 2, the Saturation one, and if it set to 3, the Value colour coefficient will be returned, and this returned value will be of type int and be between 0 and 65535.

```
int readHSV(int x, int y, int colour);
```

dRead HSV

Same as the above, but the returned value will be of type double and be between 0.0 and 1.0.

```
double dreadHSV(int x, int y, int colour);
```

Clear

The whole image is set to black.

```
void clear(void);
```

Close

Close the instance of the class, and write the image to disk.

```
void close(void);
```

Rename

To rename the file once an instance of pngwriter has been created. If the argument is a long unsigned int, for example 77, the filename will be changed to 0000000077.png

```
void pngwriter_rename(char * newname);
void pngwriter_rename(const char * newname);
void pngwriter_rename(long unsigned int index);
```

Figures

These functions draw basic shapes. Available in both int and double version.

```
void line(int xfrom, int yfrom, int xto, int yto, int red, int green,int
blue);
void line(int xfrom, int yfrom, int xto, int yto, double red, double
green,double blue);
void square(int xfrom, int yfrom, int xto, int yto, int red, int green,int
blue);
void square(int xfrom, int yfrom, int xto, int yto, double red, double
green,double blue);
void filledsquare(int xfrom, int yfrom, int xto, int yto, int red, int
green,int blue);
void filledsquare(int xfrom, int yfrom, int xto, int yto, double red, double
green,double blue);
void circle(int xcentre, int ycentre, int radius, int red, int green, int
blue);
void circle(int xcentre, int ycentre, int radius, double red, double green,
double blue);
void filledcircle(int xcentre, int ycentre, int radius, int red, int green,
int blue);
void filledcircle(int xcentre, int ycentre, int radius, double red, double
green, double blue);
```

Bezier Curve

(After Frenchman Pierre Bézier from Regie Renault) A collection of formulae for describing curved lines and surfaces, first used in 1972 to model automobile surfaces. (from the The Free On-line Dictionary of Computing) See http://www.moshplant.com/direct-or/bezier/ *for one of many available descriptions of bezier curves. There are four points used to define the curve: the two endpoints of the curve are called the anchor points, while the other points, which define the actual curvature, are called handles or control points. Moving the handles lets you modify the shape of the curve.*

```
void bezier(  int startPtX, int startPtY,
              int startControlX, int startControlY,
              int endPtX, int endPtY,
              int endControlX, int endControlY,
              double red, double green, double blue);

void bezier(  int startPtX, int startPtY,
              int startControlX, int startControlY,
              int endPtX, int endPtY,
```

```
int endControlX, int endControlY,
int red, int green, int blue);
```

Read From File

Open the existing PNG image, and copy it into this instance of the class. It is important to mention that not all colour types and bit depths are supported. Try and make sure that your PNG image is of bit depth 8 or 16.

```
void readfromfile(char * name);
void readfromfile(const char * name);
```

Get Height

When you open a PNG with readfromfile() you can find out its height with this function.

```
int getheight(void);
```

Get Width

When you open a PNG with readfromfile() you can find out its width with this function.

```
int getwidth(void);
```

Write PNG

Writes the PNG image to disk. You can still change the PNGwriter instance after this.

```
void write_png(void);
```

Set Compression Level

Set the compression level that will be used for the image. -1 is default, 0 is none, 9 is best compression.

```
void setcompressionlevel(int level);
```

Get Bit Depth

When you open a PNG with readfromfile() you can find out its bit depth with this function.

```
int getbitdepth(void);
```

Get Colour Type

When you open a PNG with readfromfile() you can find out its colour type.

```
int getcolortype(void);
```

Set Gamma Coeff

Set the image's gamma (file gamma) coefficient. The default value of 0.5 should be fine.

```
void setgamma(double gamma);
```

Get Gamma Coeff

Get the image's gamma coefficient. This is experimental.

```
double getgamma(void);
```

Set Text

Sets the text information in the PNG header. If it is not called, the default is used.

```
void settext(char * title, char * author, char * description, char *
software);
void settext(const char * title, const char * author, const char *
description, const char * software);
```

Version Number

Returns the PNGwriter version number.

```
double version(void);
```