Soutien ACPI - Color

Generated by Doxygen 1.8.16

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

Class Documentation

2.1 Color Class Reference

Public Member Functions

- Color & fromHue (float hue)
- Color & fromHSV (float hue, float saturation, float value)
- void toHSV (float &hue, float &saturation, float &value) const

2.1.1 Member Function Documentation

2.1.1.1 fromHSV()

```
Color & Color::fromHSV (
            float hue,
            float saturation,
            float value )
```

Modifie la couleur selon des données HSV

- hue la teinte de la couleur noté en dégrée de 0.0f à 360.0f
- saturation la saturation de la coueleur noté de 0.0f (blanc) à 1.0f (coloré)
- value la valeur de la couleur (aussi dit son intensité) noté de 0.0f (noir) à 1.0f (coloré)

Returns

une référence de l'objet actuel coloré selon les valeurs HSV

2.1.1.2 fromHue()

```
Color & Color::fromHue (
            float hue )
```

Modifie la couleur selon une teinte

- hue la teinte de la couleur en degrée de 0.0f à 360.0°f

Returns

Une référence de l'objet courant coloré selon la teinte

2.1.1.3 toHSV()

```
void Color::toHSV (
            float & hue,
            float & saturation,
            float & value ) const
```

Transforme la couleur actuelle en ces valeurs HSV

- hue référence de la teinte de la couleur noté en dégrée de 0.0f à 360.0f
- saturation référence de la saturation de la coueleur noté de 0.0f (blanc) à 1.0f (coloré)
- value référence de la valeur de la couleur (aussi dit son intensité) noté de 0.0f (noir) à 1.0f (coloré)

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

- classes/Color.h
- classes/Color.cpp

2.2 Kernel Class Reference

Public Member Functions

- Kernel (const vect2Df &vect)
- void ApplyKernelToImage (const colorVector &input, colorVector &output, unsigned width, unsigned height)

Protected Member Functions

- void ApplyKernelToPixel (const colorVector &input, colorVector &output, unsigned x, unsigned y)

2.2.1 Constructor & Destructor Documentation

2.2.1.1 Kernel()

```
Kernel::Kernel (
            const vect2Df & vect )
```

crée un kernel à partir d'un vecteur2D de float

- vect un vecteur2D de float

2.2.2 Member Function Documentation

2.2.2.1 ApplyKernelToImage()

```
void Kernel::ApplyKernelToImage (
            const colorVector & input,
            colorVector & output,
            unsigned width,
            unsigned height )
```

Applique le kernel à une image

- input un vecteur de couleurs représentant une image d'entrée
- output un vecteur de couleur représentant une image de sortie
- width la longeur de l'image
- height la hauteur de l'image

2.2.2.2 ApplyKernelToPixel()

```
void Kernel::ApplyKernelToPixel (
            const colorVector & input,
            colorVector & output,
            unsigned x,
            unsigned y )  [protected]
```

Applique le kernel à un pixel de coordonnée x,y

- input un vecteur de couleurs représentant une image d'entrée
- output un vecteur de couleur représentant une image de sortie
- x la coordonnée x dans l'image ou appliquer le kernel
- y la coordonnée y dans l'image ou appliquer le kernel

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

- classes/Editor.h
- classes/Editor.cpp

2.3 Asemco::PNG Class Reference

Public Member Functions

- **PNG** (const char *filename)
- void reserve (unsigned int height, unsigned int width)
- const std::vector< unsigned char > & **getBytes** () const
- unsigned int **getWidth** () const
- unsigned int **getHeight** () const
- unsigned long **getSize** () const
- void colorsToBytes (const colorVector &colors)
- void bytesToColor (colorVector &colors) const
- void **loadFromFile** (const char *filename)
- void **saveToFile** (const char *filename)

Protected Attributes

- std::vector< unsigned char > **_bytes**
- unsigned int **_width**
- unsigned int **_height**
- unsigned long **_size**

2.3.1 Member Function Documentation

2.3.1.1 bytesToColor()

```
void Asemco::PNG::bytesToColor (
            colorVector & colors ) const
```

transforme tous les bytes de l'image en vecteur de couleur

- pixels un vecteur de couleur (colorVector, voir Color.h)

2.3.1.2 colorsToBytes()

```
void Asemco::PNG::colorsToBytes (
            const colorVector & colors )
```

remplaces les bytes de l'image selon un vecteur de couleur

- pixels un vecteur de couleur (colorVector, voir Color.h)

2.3.1.3 reserve()

```
void Asemco::PNG::reserve (
            unsigned int height,
            unsigned int width )
```

réserve la place mémoire nécessaire pour stocker l'image prévue

- height la hauteur de l'image (en pixels)
- width la largeur de l'image (en pixels)

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

- classes/PNG.h
- classes/PNG.cpp

Chapter 3

Example Documentation

3.1 C:/Users/killi/CLionProjects/Soutien-ACPI-Color/tools/AsMath.h

normalise 2 nombres selon le nombre le plus grand.

float f1 = 4.0f; float f2 = 2.0f; normalize(f1, f2); // f1 -> 1.0 et f2 -> 0.5

- num1 une référence d'un float strictement positif
- num2 une référence d'un float strictement positif

```
#pragma once
#include <cmath>
#include <random>
namespace Asemco
{
    // définie une méthode préprocesseur qui effectue un aléatoire bornée.
    #define randRange(n_min, n_max) rand() % (n_max - n_min + 1) + n_min
    inline void normalize(float & num1, float & num2)
    {
        if (num1 > num2)
        {
            num2 = (1.0f / num1)*num2;
            num1 = 1.0f;
        }
        else
        {
            num1 = (1.0f / num2)*num1;
            num2 = 1.0f;
        }
    }
    inline float f3min(float f1, float f2, float f3)
    {
        float tmp = (float)fmin(f1, f2);
        return (float)fmin(tmp, f3);
    }
    inline float f3max(float f1, float f2, float f3)
    {
        float tmp = (float)fmax(f1, f2);
        return (float)fmax(tmp, f3);
    }
    inline float ufmodf(float fnum, float fmod)
    {
        float r = fmodf(fnum, fmod);
        return r = (r < 0) ? (fmod - r * -1.0f) : r;
    }
    inline unsigned char floatToChar(float f){
        if(f < 0)     return 0;
        if( f > 255 ) return 255;
        return (char)f;
    }
}
```

Index