API documentation for libcolor

Contents

1 Overview

A library for easily manipulating pixels

Author: Kyle Alexander Buan
Version: 1.0
License: GPL-3
Flags: `-lcolor`

2 color.eh — Header for easily manipulating pixels

```
use "color.eh"
```

2.1 Description

2.2 Types

```
type Color = {
  r: Int = 0,
  g: Int = 0,
  b: Int = 0
}
```

A structure that defines a single pixel with colors made up of red, green, and blue components ranging from 0 (0x00) to 255 (0xFF) each.

2.3 Functions

```
def Color.abs(): Int;
```

Will return the distance of the 3D vector (r, g, b) from the center (0, 0, 0) in the 3D cartesian plane. This function is NOT made for accuracy but for speed. Therefore, it returns values from 0 to 195,075 instead of 0 to 0 to 441.

```
def Color.sub(c: Color): Color;
```

Will subtract two colors, returns `new Color(this.r-c.r, this.g-c.g, this.b-c.b)`

```
def Color.toint(): Int;
```

Will convert a Color to Int, suitable for using in `Graphics.set_color()` Int is in the form of `0xRRGGBB`.

```
def Color.luminosity(): Color;
```

Converts a Color into greyscale.

```
def Color.closest_index(pal: List): Int;
```

Will return the index of the closest color in the palette as compared to this.

```
def Color.closest(pal: List): Color;
```

Will return the closest Color in the palette as compared to this.

```
def Image.get_pix_real(x: Int, y: Int): Color;
```

Will return a pixel from this Image coordinates (x, y). If coordinates is bigger than image, will return an error.

```
def get_image_x(i: Image): Int;
```

Returns width of image.

```
def get_image_y(i: Image): Int;
```

Returns height of image.

```
def List.highest(): Int;
```

Returns the index of the highest number in the list.

```
def Color.correct(): Color;
```

Will check if the Color contains valid values, and correct them accordingly.

```
def Color.add(c: Color): Color;
```

Add two colors together, returning `new Color(this.r+c.r, this.g+c.g, this.b+c.b)`.

```
def Color.add_int(i: Int): Color;
```

Add a greyscale color to this, returning `new Color(this.r+i, this.g+i, this.b+i)`.

```
def Color.frac(a: Int, b: Int): Color;
```

Multiply a color by a/b, returning `new Color(this.r*a/b, this.g*a/b, this.b*a/b)`.

```
def Image.get_pix(x: Int, y: Int, x_s: Int, y_s: Int): Color;
```

Will return a pixel from this Image coordinates (x, y). If coordinates is bigger than image, will return `new Color(0, 0, 0)`.

```
def Color.tohtml(): String;
```

Convert color to HTML format, returns a string "#RRGGBB"

```
def Int.tocolor(): Int;
```

Convert an Int to color. Int should be in 0xRRGGBB format.