

# その他のツール

(Ver.1.0)

2014 年 10 月

株式会社 アイディール

## 目次

eyemMiscChangeRectToRect2 .....	1
機 能    EyemRect 矩形構造体から EyemRect2 矩形構造体への変換 .....	1
eyemMiscChangeRect2ToRect .....	2
機 能    EyemRect2 矩形構造体から EyemRect 矩形構造体への変換 .....	2
eyemMiscBilinearGrayLevel .....	3
機 能    線形補間による指定位置の画像濃度取得 .....	3
eyemMiscPlotPointsOfLine .....	4
機 能    ブレゼンハムのアルゴリズムによる線分描画点の取得 .....	4

---

## eyemMiscChangeRectToRect2

---

**機 能** EyemRect 矩形構造体から EyemRect2 矩形構造体への変換

**形 式**

```
#include "eyemLib.h"

int      eyedMiscChangeRectToRect2( EyemRect *tpSrc, EyemRect2 *tpDst );
```

**解 説** EyemRect 矩形構造体(始点, サイズ)のデータを, EyemRect2 矩形構造体(始点, 終点)のデータに変換します.

**引 数**

*tpSrc	EyemRect矩形構造体(始点, サイズ)のデータです.
*tpDst	EyemRect2矩形構造体(始点, 終点)のデータが格納されます.

**戻り値** エラー報告です.

FUNC_OK	正常終了
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

**留意事項** 特にありません.

---

## eyemMiscChangeRect2ToRect

---

**機 能** EyemRect2 矩形構造体から EyemRect 矩形構造体への変換

**形 式**

```
#include "eyemLib.h"

int      eyedMiscChangeRect2ToRect ( EyemRect2 *tpSrc, EyemRect *tpDst );
```

**解 説** EyemRect2 矩形構造体(始点, 終点)のデータを, EyemRect 矩形構造体(始点, サイズ)のデータに変換します.

**引 数**

*tpSrc	EyemRect2矩形構造体(始点, 終点)のデータです.
*tpDst	EyemRect矩形構造体(始点, サイズ)のデータが格納されます.

**戻り値** エラー報告です.

FUNC_OK	正常終了
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

**留意事項** 特にありません.

---

## eyemMiscBilinearGrayLevel

---

機 能	線形補間による指定位置の画像濃度取得		
形 式	#include “eyemLib.h”  double   eyemMiscBilinearGrayLevel( EyemImage *tpImage, double dPosX, double dPosY );		
解 説	指定された画像位置の濃度を, 線形補間により取得します.		
引 数	*tpImage	濃度取得対象となるグレイ画像の情報です.	
	dPosX	指定位置のX座標です.	
	dPosY	指定位置のY座標です.	
戻り値	線型補間による画像濃度です (0～255). なお, 指定位置が画面からはみ出していたらゼロを返します.		
留意事項	特にありません.		

---

## eyemMiscPlotPointsOfLine

---

**機 能**                    ブレゼンハムのアルゴリズムによる線分描画点の取得

**形 式**                    `#include "eyemLib.h"`  
`int        eyedMiscPlotPointsOfLine( EyemOcsIXY *tpSt, EyemOcsIXY *tpEd,`  
`EyemOcsIXY taPlotPt[] );`

**解 説**                    ブレゼンハムのアルゴリズムにより, 線分描画点を取得します.

**引 数**                    `*tpSt`                    線分の始点(整数点)です.  
                         `*tpEd`                    線分の終点(整数点)です.  
                         `taPlotPt[]`                線分描画点(整数点)が格納されます. 以下の計算で得られる要素数以上の配列を指定してください.  
                         要素数 :  $\max(|tpSt \rightarrow iX - tpEd \rightarrow iX|, |tpSt \rightarrow iY - tpEd \rightarrow iY|) + 1$

**戻り値**                    エラー報告です.

1以上	(正常終了)描画点数
FUNC_ILLEGAL_ARGUMENT	パラメータが不正

**留意事項**                    特にありません.

## 改訂履歴

Version No.	内 容
1.0	• 新規発行