USB メモリの中身は ColorCMY.pdf ColorCMY.png ColorCMY.st ColorHSB.pdf ColorHSB.st ColorRGB.pdf ColorRGB.png ColorRGB.st PreLoad.st

PreLoad.st の中身を確認しておく この中身は前回、手で作ったパッケージを自動で作ってくれるもの

ということで、 PreLoad.st を File in すると KSU の中に KSU-Template が自動で作られる

この状態で、 ColorRGB.st を File in する。 実際に作っていく物は ColorRGB.png

赤:	0.25	] ——	-0			
<b>緑</b> :	0.75	]			0	
青:	0.5	]		_0		

これを作って、 ColorCMY の方は ColorRGB と似ているのでハッカソン風に自分で 直してみる。

ColorRGB, Class Ø Definition Smalltalk.KSU defineClass: #ColorRGB superclass: #{UI.ApplicationModel} indexedType: #none private: false instanceVariableNames: 'redGauge greenGauge blueGauge redField greenField blueField ' classInstanceVariableNames: " imports: " category: 'KSU-Template'

それぞれ、スライドバーと Field に対応している redGauge

```
redGauge

ifNil:

[redGauge := 0.5 asValue.

redGauge compute: [:aValue | self updateColorRed: aValue]].

^redGauge
```

redField

redField ifNil: [redField := (self valueString: self redGauge value) asValue]. ^redField

example1 を Inspect It すると



こんなグレーの画面が開く(RGB がそれぞれ 0.5)





## Window spec から Edit いろいろ置くので Grid を表示しておく UI Painter Tool on: 加法混色 rrange Grid Look Special Help On





Menu Bar を Enable にする

Edit Tools Layout A	UI Painter Tool on: 加法混色 rrange Grid Look Special Help
« < > > < ~ ``	
🗂 Main Window : 加法	Basics Details Position/Size Color Dr
	Label: Supplied by Application
	String: 加法混色
	Lookup key:
	Catalog:
	Menu Bar
	Menu: #menuBar
	Tool Bar
	Menu:
	Apply Cancel Prev Next





それぞれ、 String: に赤:、緑:、青: ID に redLabel, greenLabel, blueLabel とつ ける

A     String:     赤:       String:     赤:       Lookup key:	Main Window : 加法 — A redLabel — A greenLabel	Basics Deta	Label
Lookup key: Catalog: ID: #redLabel	A DideLaber	String:	赤:
Catalog: ID: #redLabel		Lookup key:	
ID: #redLabel		Catalog:	
Label Is Image		ID:	#redLabel
			iage
Apply Cancel Prev		Apply Can	cel Prev Nex

	<ul> <li>✓ 公 ふ ✓ □</li> <li>Main Window : 加法</li> <li>A redLabel</li> <li>A greenLabel</li> <li>A blueLabel</li> <li>abc redField</li> <li>abc greenField</li> <li>blueField</li> </ul>	Aspect: Menu: ID: Type: Format:	ook Special Help
--	---	---	------------------

Details タブの Read Only にチェックを入れておく(変更されないようにする)

🧳 🕗 🎡 🍄 🕒 🛅 Main Window : 加法	Basics Details Position Elv-by Help V
A redLabel	Input Field
A greenLabel	
A blueLabel	Font: System (Widget Text)
	Align:
abc, InputField3	Size:
	Sordered
	Opaque
	🗹 Can Tab
	🗹 Read Only
	Initially Disabled
	Initially Invisible
	Apply Cancel Prev Nex

背景色は Color タブで、色を選択して Backgrnd



この時点ではこんな感じ



次は、 Slider web 83つ置く



Aspect と ID は同じような感じで redGauge とする Start, Stop はそれぞれスライダーの上限値と下限値のこと Step は刻み幅

Main Window :加法 A redLabel A greenLabel A blueLabel BlueLabel BlueField BlueField Slider1 Slider3	Aspect: ID: Start: Stop: Step:	Details     Position     Fly-by Help     V       Slider       #redGauge       0       1       0.01
	Apply Ca	Prev Next

ここまで来たら Install 🇳

example1 を Do it するとこんな感じで動く

赤:	0.79	ī ———		0
		_	_	
緑:	0.67	]		
青:	0.21	]		
		<b>_</b>		

中身を見ていく 初めて開かれたときにウィンドウを初期化する(グレーにする) postOpenWith: aBuilder

super postOpenWith: aBuilder. self updateColor

実際にバックグラウンドを塗る

updateColor

self builder ifNotNil: [:aBuilder I aBuilder window ifNotNil: [:aWindow I aWindow background: self color; display]]

コレで丸めてしまう valueString: aValue

^(aValue roundTo: 0.01) printString

スライダーをいじると呼ばれて、中身を書き換える

updateColorRed: aValue

InputState default altDown ifTrue: [self greenGauge value = aValue ifFalse: [self greenGauge value:

aValue].

self blueGauge value = aValue ifFalse: [self blueGauge value:

aValue]]. self redField value: (self valueString: aValue). self updateColor

これで、 RGB は終わり

CMY をやる ColorCMY.st を File in

こんなものを作っていく

シアン :	0.75	/5%
マゼンタ:	0.25	25%
1-0-:	0.5	0\$

Window Spec を開く

Spin Button と Percent Done Bar を使う



今回はバーとボタンが同じ値を見ている RGB と違って Type を Number にしている

Edit Tools Layout Ar 《 《 《 《 《 《 《 Main Window : 滅法 ····································	range Grid	Look Special Details Posit Spin Bu #cyanGauge #cyanButton Number 0 0.01	Help ion Fly-by Help V • itton
	Apply Ca	ncel	Prev Next

Selection Backgrnd(o が無いけど仕様と思うしかないと…) も色を付けておかないと チカチカする





& 42 & 44 🗳	968 🍕	
Main Window : 減法 <sub>[1973]</sub> cyanButton <sub>[1973]</sub> magentaButton	Basics Details Position Color Label	
	Label: 📃 Supplied by Appli	cation
	String: シアン :	
	Lookup key:	
<u>A</u> cyanLabel <u>A</u> magentaLabel	Catalog:	
A yellowLabel	ID: #cyanLabel	
	📃 Label Is Image	
	Apply Cancel	Prev Ne:

cyan うんたら以外も cyan 部分を magenta や yellow と変更する。

こんな感じで動くようになった

	減法混色	
シアン :	0.69	69%
マゼンタ:	0.15	15%
イエロー:	0.5	5 <mark>0%</mark>

## 最後に HSB



こんな部品はないので自前でつくる HSB は Model だけじゃなくて Controller と View がある

Definition をみると Gauge が3つだけ

Smalltalk.KSU defineClass: #ColorHSB superclass: #{UI.ApplicationModel} indexedType: #none private: false instanceVariableNames: 'hueGauge saturationGauge brightnessGauge ' classInstanceVariableNames: " imports: " category: 'KSU-Template'

interface opening を見ると、それぞれの View がある

WindowSpec を Edit

View Holder この様に設置して



それぞれのラベルを設定して





で、最後に Install

この状態で、 example1 を実行すると



この様に動く

で、変な線が出ていて見栄えが良くないので、 Bordered のチェックボックスを外すと消える





これにておしまい。